

ZÁPADOČESKÁ UNIVERZITA V PLZNI

FAKULTA ELEKTROTECHNICKÁ

Katedra výkonové elektroniky a strojů

BAKALÁŘSKÁ PRÁCE

Jednoduché paritní kódy pro korekci chyb

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Tomáš NEJEDLO**
Osobní číslo: **E18B0131P**
Studijní program: **B2644 Aplikovaná elektrotechnika**
Studijní obor: **Aplikovaná elektrotechnika**
Téma práce: **Jednoduché paritní kódy pro korekci chyb**
Zadávací katedra: **Katedra výkonové elektroniky a strojů**

Zásady pro vypracování

1. Realizujte základní kodér paritního kódu o nejméně třech dimenzích.
2. Realizujte základní dekodér paritního kódu o nejméně třech dimenzích.
3. Analyzujte detekční a korekční schopnosti navrženého základního kódu pro různé informační poměry.

Rozsah bakalářské práce: **30 – 40 stran**
Rozsah grafických prací: **podle doporučení vedoucího**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. DUDÁČEK, L., VEŘTÁT, I. Multidimensional parity check codes with short block lengths. In 2016 24th Telecommunications Forum (TELFOR 2016) : Proceedings of Papers. Piscataway: IEEE, 2016. s. 258-261. ISBN: 978-1-5090-4086-5.

Vedoucí bakalářské práce: **Ing. Ivo Veřtát, Ph.D.**
Katedra elektroniky a informačních technologií

Datum zadání bakalářské práce: **9. října 2020**
Termín odevzdání bakalářské práce: **27. května 2021**



Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan



Prof. Ing. Václav Kůs, CSc.
vedoucí katedry

Abstrakt

Předkládaná bakalářská práce je zaměřena na testování multidimenzionálních paritních kódů a seznamuje čtenáře s různými metodami kódování, které se využívají pro zabezpečení kódových slov. Cílem této práce bylo studium metod kódování a realizace kodéru a dekodéru alespoň o 3 dimenzích v programovacím jazyce MATLAB.

Klíčová slova

Paritní kódy, Kódování, Dekódování, CubeSat, Multidimenzionální paritní kódy, Matlab, Samoopravné kódy

Abstract

The bachelor thesis presents multidimensional cross parity check codes and demonstrates different coding methods which are used for securing code words. The goal of this bachelor thesis had been study of coding methods and realize channel encoder and decoder with at least 3 dimensions in MATLAB.

Key words

Multidimensional Cross Parity Check Codes, Decoding, Coding, CubeSat, Parity codes, Matlab, Correcting codes

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....

podpis

V Plzni dne 27.5.2021

Tomáš Nejedlo

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Ivu Veřtátovi Ph.D. za cenné profesionální rady, připomínky a metodické vedení práce. Dále bych chtěl poděkovat své rodině za podmínky a zázemí, které mi byly poskytnuty ke studiu.

Obsah

ÚVOD	10
1 SDĚLOVACÍ SYSTÉM	11
1.1 ZDROJ A PŘÍJEMCE INFORMACE	11
1.2 KODÉR ZDROJE	12
1.3 KODÉR KANÁLU	13
1.4 MODULÁTOR	13
1.5 DEMODULÁTOR	15
1.6 DEKODÉR KANÁLU	15
2 CUBESAT.....	15
2.1 ÚVODNÍ INFORMACE A PARAMETRY	15
2.2 DRUHY KOMUNIKACE.....	16
2.3 MODULACE A KÓDOVÁNÍ	17
3 KÓDOVÁNÍ.....	18
3.1 POUŽITÉ POJMY	18
3.2 PARITNÍ BITY	19
3.3 HAMMINGOVY KÓDY.....	19
3.3.1 <i>Hammingův kód (7,4)</i>	19
3.3.2 <i>Dekódování</i>	21
3.4 BCH KÓDY.....	21
3.4.1 <i>Kódování</i>	22
3.4.2 <i>Dekódování</i>	23
3.5 REED-SOLOMONOVY KÓDY	24
3.5.1 <i>Kódování</i>	24
3.5.2 <i>Dekódování</i>	24
4 MULTIDIMENZIONÁLNÍ PARITNÍ KÓDY	26
4.1 DVOUROZMĚRNÉ PARITNÍ KÓDY	26
4.2 VÍCEROZMĚRNÉ PARITNÍ KÓDY	27
4.3 KÓDOVÁNÍ	30
4.4 DEKÓDOVÁNÍ.....	31
5 PRAKTICKÁ ČÁST	33
5.1 VYTVOŘENÍ BLOKU DAT	34
5.2 KÓDOVÁNÍ	34
5.3 MODULACE A DEMODULACE	35
5.4 DEKÓDOVÁNÍ	36
5.4.1 <i>Dekódování bez váhových koeficientů</i>	36
5.4.2 <i>Dekódování s váhovými koeficienty</i>	37
5.4.3 <i>Dekódování při kombinaci obou metod</i>	37
5.5 CHYBY DEKÓDOVÁNÍ	38
5.5.1 <i>Charaktery chyb</i>	38
5.5.2 <i>Nedetkování chyb</i>	41

5.6	MĚŘENÍ A VÝPOČTY	41
5.6.1	Bitová chybovost algoritmů	41
5.6.2	Kódový zisk	44
5.6.3	Chybovost rámců	45
5.6.4	Nerozpoznaná chybovost rámců	48
5.7	SHRnutí VÝSLEDKŮ	51
5.7.1	Výsledky bitové chybovosti	51
5.7.2	Výsledky rámcové chybovosti	51
5.7.3	Výsledky nerozpoznané chybovosti rámců	51
6	ZÁVĚR	52
	SEZNAM POUŽITÝCH OBRÁZKŮ	53
	SEZNAM POUŽITÝCH TABULEK	54
	SEZNAM SYMBOLŮ A ZKRATEK	55
	SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	57
	PŘÍLOHY	1

Úvod

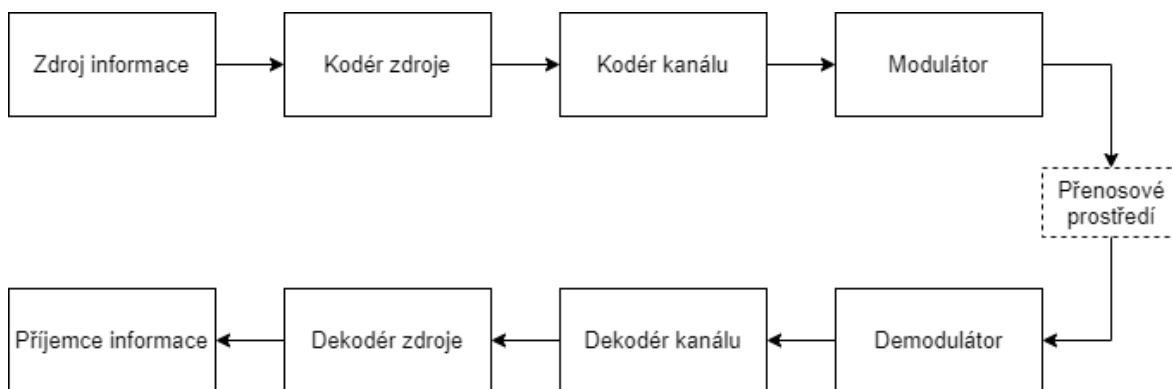
Předkládaná práce je zaměřena na paritní kódy a testování multidimenzionálních paritních kódů, které by se mohly uplatnit v satelitech malých rozměrů CubeSat.

Text bakalářské práce je rozdělen do pěti částí; první se zabývá sdělovacím systémem a jednotlivými funkčními bloky, které jsou při přenosu dat a informací využívány. Druhá část představuje CubeSaty a uvádí o nich základní informace a jejich parametry, jsou zde popsány jednotlivé možnosti jejich komunikace a přehled využívaných modulací. Třetí část se zabývá kódovacími metodami, které se u CubeSatů uplatňují a jsou zde vysvětleny základní postupy kódování a dekodování. Čtvrtá část se zabývá multidimenzionálními paritními kódy, možnostmi a principy kódování a dekodování. Pátá část je věnována praktické části, ve které jsou popsány použité principy a metody vytvoření bloku dat, modulací, kódování, dekodování, na což navazují změřené výsledky a jejich analýza.

Cílem bakalářské práce bylo studium kódovacích a dekodovacích metod, realizace kodéru a dekodéru kanálu v programovém prostředí MATLAB a testování bitové chybovosti pro AWGN kanál a jednotlivé metody dekodování.

1 Sdělovací systém

Přenos informací po staletí spočíval v předávání informací z jednoho místa na jiné. Současný sdělovací systémy využívají zpravidla digitální přenosy, které se zabezpečují proti chybovým kódováním (FEC). Sdělovací systém lze rozdělit na přijímací a vysílací část a přenosové prostředí. Sdělovací kanál přenosového prostředí může být řešen metalickým spojem, opticky či rádiově. Sdělovací systém lze rozdělit do několika funkčních bloků (viz. Obrázek 1)



Obrázek 1: Obecné schéma sdělovacího systému.

1.1 Zdroj a příjemce informace

Prvním blokem v obecném schématu sdělovacího systému pro vysílací část je zdroj informace. Zdroj informace může být zvukový či obrazový signál, číselná data či jakákoliv jiná informace. V případě zvukového signálu se v tomto bloku nachází elektroakustický měnič, který převádí mechanické a akustické kmity na elektrický signál, zatímco obrazový signál je převeden na elektrický signál pomocí optoelektrického měniče. Tato informace je vyjádřena formou zprávy, ať už spojité či diskrétní, a tato zpráva je v následujících blocích zpracována do podoby, která je vhodná pro přenos.

Naopak posledním blokem je příjemce informace. Tato část má na starosti převod signálu pro naše smyslové orgány, protože ve většině případech přenášíme obrazovou či zvukovou informaci. V případě zvukového signálu dojde za pomoci elektroakustických měničů k převodu elektrické informace na akustickou, což nám umožní danou informaci slyšet.

1.2 Kodér zdroje

Hlavní úlohou kodéru zdroje je zdrojové kódování, které odstraní irelevantní a redundantní informace, které nejsou využité příjemcem. Díky tomu dochází ke snížení přenosové rychlosti a zmenšení energetických nároků potřebných pro přenos, ať už se jedná o finanční či hardwarovou úsporu. Zdrojové kódování dělíme na ztrátové a bezztrátové. Ztrátové kódování se používá u komprese zvukové a obrazové informace, bezztrátové dekódování u přenosu datových souborů.

Další funkcí kodéru zdroje může být převod analogového signálu na signál digitální z důvodu konverze do číslicové podoby, přičemž v některých případech je tato funkce řešena předřadným A/D převodníkem. Konverze do číslicové podoby probíhá ve 3 krocích: vzorkováním, kvantováním a kódováním.

Vzorkování má za následek diskretizaci v čase, jelikož po procesu vzorkování je signál definován pouze v diskrétních časových okamžicích a spojitý v amplitudě. Ideální vzorkování se dá provést Diracovými impulsy, ale tato metoda je v praxi prakticky nerealizovatelná, neboť šířka vzorků by musela být nekonečně malá. Druhou možností je vzorkování 1. druhu, které odpovídá pulsně amplitudové modulaci, nicméně i tato možnost je prakticky nerealizovatelná, neboť digitální systém může přenést pouze jednu hodnotu amplitudy, a ne celý interval, který nabývá různých hodnot. Z tohoto důvodu se v praxi využívá vzorkování 2. druhu, ve kterém se šířka pulsů rovná šířce vzorkovací periody. Je nutné nezapomenout na vzorkovací teorém a podmínku, kterou je potřeba dodržet pro zpětnou rekonstrukci signálu, aby nedošlo k tzv. aliasingu.

Druhý krok v digitalizaci je kvantování, čímž dojde k diskretizaci v amplitudě. Po tomto kroku je signál diskrétní v čase i amplitudě, a proto se mohou hodnoty signálu zaokrouhlit na předem definované kvantizační hladiny. Ve středu mezi kvantizačními hladinami se nachází rozhodovací úrovně, podle kterých se rozhodne, zdali má proběhnout zaokrouhlení na vyšší či nižší kvantizační hladinu. Díky tomu dochází k chybám, a to buď ke kladné kvantizační chybě, pokud proběhne zaokrouhlení na vyšší kvantizační hladinu, než byla ve skutečnosti, nebo záporné kvantizační chybě. Projevy této chyby se podobají šumu, a proto se tato chyba nazývá kvantizační šum.

Třetím a posledním krokem je kódování. Ke každé kvantizační hladině je přiřazeno vhodné kódové slovo, nejčastěji ve dvojkové soustavě tvořené logickými nulami a jedničkami. Pro

celkový počet kvantizačních hladin N a počet bitů b platí vztah

$$N = 2^b \quad (1.1)$$

Kódování lze provést například pulsně kódovou modulací (PCM). Na výstupu kodéru zdroje je číslo kódované v binárním kódu, a to je posláno do kodéru kanálu.

1.3 Kodér kanálu

V tomto bloku proběhne kanálové kódování, které zabezpečí kanál proti chybám a rušení při přenosu, jelikož na rádiové prostředí působí rušení a při přenosu by se mohla jednotlivá kódová slova poškodit, což by vedlo na straně příjemce k výsledku, kdy by byly přijaté bity po demodulaci chybně interpretovány. Z tohoto důvodu je datové slovo zvětšeno o redundantní informaci, aby dekodér na straně příjemce byl schopen rozpoznat, jestli signál obdržel v pořádku či je poškozen. Toho může být docíleno zabezpečením bloku dat například v podobě sudé a liché parity, ačkoliv to má za následek nárůst přenosové rychlosti. Tento způsob má pouze omezené detekční schopnosti a nemá schopnost korekce kódových slov. Jako korekční kódy se používají Reed-Solomonovy kódy, konvoluční kódy, LDPC kódy a jiné.

1.4 Modulátor

Modulátor zajišťuje převod elektrického signálu na jiný typ, jehož vlastnosti jsou optimalizovány pro přenosovou cestu. Druhou funkcí modulátoru je přizpůsobení parametrů signálu a umísťuje například rozhlasové stanice do vhodného frekvenčního pásma, které je přiděleno rozhlasové stanici, a umožní tím sdílet kanál více uživatelům. Nejčastěji se realizuje přenos v přeloženém pásmu, kdy je signál namodulován na nosný signál do vyšší kmitočtové polohy.

Modulace lze dělit podle různých kritérií. Jedno z dělení rozděluje modulace na modulaci s nosnými vlnami, které jsou kmitočtově přeloženy do vyššího pásma a modulace v základním pásmu.

Modulace s nosnými vlnami přivádí do modulátoru nosný harmonický signál a rozlišujeme je na analogové a digitální.

U analogových modulací ovlivňujeme amplitudu, fázi či frekvenci nosného signálu analogovým modulačním signálem a podle toho rozlišujeme analogovou amplitudovou

modulaci (AM), frekvenční analogovou modulaci (FM) a fázovou analogovou modulaci (PM).

Digitální modulace s nosnými vlnami jsou řešeny klíčováním amplitudy, fáze či frekvence nosného signálu digitálním modulačním signálem. Jedná se o klíčování amplitudovým posuvem (ASK), klíčování frekvenčním posuvem (FSK) a klíčování fázovým posuvem (PSK). Hlavní rozdíl je v řídicím signálu, kterým je datový bitový signál. Pro digitální modulace je možné využívat modulátor, který používá 2 techniky dohromady a může tedy ovlivňovat např. fázi nosné i amplitudu zároveň. V tomto případě mluvíme o tzv. QAM modulaci.

Modulace v základním pásmu mohou být buď diskrétní kódované, nebo diskrétní nekódované. U těchto modulací již není použit nosný signál a dochází k přenosu v základním kmitočtovém pásmu.

Jednou z diskrétních nekódovaných modulací je pulsně amplitudová modulace (PAM), která odpovídá vzorkování 1. druhu, kdy pulsy po určitý krátký čas kopírují původní signál. Druhou z těchto modulací je Pulsně šířková modulace (PWM), která je rozsáhle využívaná v oblasti řízení elektrických pohonů. U této modulace se v určitých časových okamžicích nastavuje šířka pulsu na výstupu PWM tak, aby amplituda byla úměrná šířce pulsu, kdy v místech nejvyšší amplitudy nastává maximální šířka pulsu. Další z těchto modulací je pulsně polohová modulace, která neurčuje šířkou pulsu okamžitou hodnotu amplitudy, ale určuje polohu pulsu, jelikož jeho šířka je pořád stejná.

Druhou skupinou modulací v základním pásmu jsou modulace diskrétní kódované. Mezi ně patří především pulsně kódová modulace a další její modifikované verze. Pulsně kódová modulace je proces, při kterém konvertujeme analogový signál do číslicové podoby vzorkováním, kvantováním a kódováním

V případě, že přenášíme signál v základním pásmu a přenášíme bitové slovo po metalickém spoji, tak je vypuštěn de/modulátor. Nedochozí v tomto případě ke kmitočtové transpozici a aplikují se linkové kódy. Ty zajišťují potlačení stejnosměrné složky ve spektru, přičemž v případě neměnné binární posloupnosti dochází po několika neměnných bitech podle různých pravidel k převodu bitu do jeho inverzní polohy, aby se usnadnila synchronizace časových základů přijímače a vysílače a dochází tím k optimalizaci šířky frekvenčního

pásma.

1.5 Demodulátor

Demodulátor je zařízení, které slouží k demodulaci přijatého signálu. V případě využití modulace s nosnou vlnou odstraní demodulátor nosnou složku a upravuje přijatý signál do podoby vhodné pro zpracování dalšími bloky. Proto je jeho hlavním úkolem odstranit nosnou složku a převést zpět na modulační signál.

1.6 Dekodér kanálu

Dekodér kanálu slouží k dekódování přijatého signálu na základě zvolené metody kódování a pravidel, které zavedl kodér kanálu. Jeho funkcí je obnovení původního signálu a zjistit, jestli je signál nepoškozen anebo došlo k jeho chybné demodulaci. V dekodéru kanálu se například provádí kontroly součtu parity, které slouží k ověření správnosti signálu při použití paritních kódů. V případě, že došlo k chybné demodulaci, můžeme vyžádat opětovné zaslání signálu. V případě použití detekčních kódů (např. Reed-Solomonovy kódy, konvoluční kódy, LDPC kódy a jiné) je proveden v tomto bloku pokus o odstranění chybovosti přenosu.

2 CubeSat

2.1 Úvodní informace a parametry

CubeSat byl prvně vytvořen v roce 1999 Kalifornskou technickou univerzitou a Stanfordskou univerzitou, aby studentům zjednodušil přístup do vesmíru. Od té doby se pikosatelity a nanosatelity CubeSat rozšířily po celém světě a nevyužívají je pouze vzdělávací instituce a univerzity pro studijní účely, ale i soukromé firmy a vládní organizace. To všechno z důvodu cenově dostupného vypuštění těchto pikosatelitů a možností jejich vypuštění na většině raketových nosičů.

Satelity CubeSat jsou malé krychle se stranou o délce 10 cm, kdy dosahují hmotností až 1,33 kg. Jejich rozměry mohou být i větší a značí se 1U, 2U, 3U či 6U v závislosti na počtu jednotlivých bloků, kdy např. 2U CubeSat by měl rozměry 10 cm x 10 cm x 20 cm. Pro výrobu nosné konstrukce těchto satelitů se převážně používá hliník 7075 nebo 6061, který má maximální pevnost v tahu přes 280 MPa, v modifikovaných verzích dosahuje jeho pevnost v tahu až 500 MPa. Všechny CubeSaty bývají tvořeny pouze jednoduchými mikrokontrolery z důvodu náchylnosti k ionizujícímu záření a chybí jim dostatečný

výpočetní výkon. Dále musí pikosatelity CubeSat splňovat několik dalších omezení, aby se zabránilo poškození satelitu a rakety při vzletu, např. je zakázána veškerá pyrotechnika a nebezpečné materiály, veškerá elektronika musí být během startu vypnutá a další.

Pro komunikaci jsou použity všesměrové antény umístěné na satelitu, které mají nízký zisk a jsou do nich dodávány nízké úrovně signálu z výkonových zesilovačů, protože satelity CubeSat mají k dispozici pouze málo elektrické energie. CubeSat je vybaven několika solárními panely, které se nachází na jeho těle a případně na výklopných křídlech. V závislosti na ploše solárních panelů generují energii od jednotek Wattů až do několika málo desítek. Většina satelitů CubeSat používá výkonu pro navázání spojení se zemí do 1 W. Pozemní stanice smí využívat mnohem větší vysílací výkon, který se pohybuje okolo 100 W k odeslání dat ze země na CubeSat.

Vypuštění nákladu na nízké oběžné dráhy je nejlevnější a současné rakety na ně mohou vynést desítky tuny nákladu. Z toho důvodu se malé satelity CubeSat se pohybují především na LEO oběžných drahách, což jsou nízké oběžné dráhy se vzdáleností do 2000 km nad zemským povrchem. Oběžná doba satelitů okolo země závisí na výšce satelitu a odpovídá přibližně 90 až 120 minutám. Největší výhodou LEO oběžných drah je malé zpoždění signálu, které je potřeba především pro systémy, které zprostředkovávají komunikaci v reálném čase. Další z výhod je krátká komunikační vzdálenost, kdy mobilní zařízení nemohou disponovat dostatečně vysokým výkonem a krátká komunikační vzdálenost znamená i menší ztráty úrovně signálu. Nevýhodou nízkých oběžných drah je malé pokrytí území jedním satelitem, kdy pro systémy s globálním pokrytím je zapotřebí několik desítek satelitů. [1]

2.2 Druhy komunikace

CubeSaty mohou využívat různé komunikační technologie s různými možnostmi navázání spojení se zemským povrchem. Mezi základní možnosti patří klasické radiové vysílání a experimentálně se zkouší komunikace pomocí laseru. Klasická radiová komunikace s využitím satelitů CubeSat je rychlostně velmi omezená a nepřesahuje většinou rychlost 9,6 kb/s. To je většinou dáno omezeným výkonem vysílání satelitu CubeSat jeho všesměrovou anténou s nízkým ziskem a malou šířkou sdělovacího kanálu, který tomuto vysílání bývá přidělen. Nicméně v nedávné době bylo testování spojení CubeSatu ve výšce 450 km se zemí laserem, kdy se dosáhlo přenosové rychlosti až 100 Mb/s s bitovou chybovostí 10^{-6} . Ovšem

spojení laserem i přes výhodu vysokorychlostního spojení se zemí se nedá zcela využít, protože dochází v atmosféře k absorpčním a rozptylovým ztrátám, atmosférickým turbulencím způsobenými změnami teplot a tlaku či ztrátou divergence paprsku. Proto je spojení laserem pouze ve fázi výzkumu. V současné době jsou proto pro komunikaci CubeSatu se zemí široce používány vysokofrekvenční kanály. [2]

2.3 Modulace a kódování

Pro modulaci a kódování CubeSatu lze využít spoustu druhů modulací a protichybového kódování a tím je hledán jistý kompromis v závislosti na parametrech s ohledem na danou misi. Mezi tyto parametry patří frekvenční pásmo, rychlost přenosu dat, v jakých periodách je schopen CubeSat komunikovat či požadovaná chybovost přenosu dat.

Většina CubeSat projektů je využívána především na vědecké a studijní účely, ve kterých se nekladou velké nároky na rychlost přenosu dat, a proto se používají především jednoduché modulační metody. Značné zastoupení mezi modulacemi má modifikované klíčování fázovým posuvem (PSK), ať už kvadrurní (QPSK) či binární (BPSK), M-PSK či asymetrická M-PSK, kde $M \in \{4, 8, 16, 32\}$, avšak u těchto modulací vyššího řádu dochází k nelineárnímu zkreslení výkonovým zesilovačem.

Značně rozšířené je i klíčování minimálním posuvem (MSK). Tato modulace je formou klíčování kontinuálním fázovým kmitočtovým posunem, kdy kódování bity probíhá střídavě mezi kvadrurními složkami a každý bit je kódován jako poloviční sinusoid. U této modulace je rozdíl mezi vyšší a nižší frekvencí shodný s poloviční bitovou rychlostí. Zároveň je maximální odchylka rovna polovině maximální frekvence, a to má za následek ortogonalitu křivek pro 0 a 1. V praxi se používá v podobě GMSK modulace. Kromě těchto modulací se také často používá tzv. AFSK modulace, což je modifikace klíčování frekvenčním posuvem, ve které nabývají bity pouze hodnot 0 či 1 v závislosti na frekvenci nosné složky.

Jednou z nich je binární klíčování frekvenčním posuvem (BFSK), což je modifikace klíčování frekvenčním posuvem, ve které nabývají bity pouze hodnot 0 či 1 v závislosti na frekvenci nosné složky.

V bezdrátových aplikacích či v satelitní technice je pro přenos informací využívána dopředná chybová korekce (dále jen FEC), kdy přijímač provádí kontroly poslaných dat na

základě nadbytečných bitů, díky čemuž mohou být chyby způsobené šumem zjištěny a následně opraveny, aniž by musela být data opětovně poslána. To platí pouze pro malý výskyt chyb v přenesené informaci. Jestliže se v přenášené informaci objeví vysoké množství chyb, jejich odstranění selhává a je zapotřebí zaslat danou informaci znovu. Hlavní výhodou FEC je možnost vysílání informací z jednoho zdroje k mnoha přijímacím cílům. [2]

Samoopravné kódy lze rozdělit do dvou hlavních skupin, konvolučních a blokových kódů. Blokové kódy, jak již název vypovídá, fungují v jednotlivých blocích určité délky a mezi hlavní vlastnosti blokových kódů patří linearita a cykličnost. Nejznámější blokové kódy jsou Hammingovy kódy, BCH kódy či Reed-Solomonovy kódy. Konvoluční kódy mají bitové sekvence o libovolné velikosti a hodnota výstupních bitů závisí nejen na vstupních bitech, ale i bitech uložených v paměti. Jednotlivé metody kódování jsou popsány v následující samostatné kapitole.

3 Kódování

Pod procesem kódování spočívá přidání redundantních bitů k přenášené informaci, aby mohl dekodér na straně kanálu detekovat chyby, které nastanou při přenosu a případně tyto chyby opravit. Dále se budu zabývat pouze blokovými kódy z důvodu zaměření práce na multidimenzionální paritní kódy, které rovněž patří do skupiny blokových kódů.

3.1 Použité pojmy

- **Binární kódy**

Binární kódy jsou kódy, které jsou tvořeny pouze čísly z dvojkové soustavy, kterými jsou čísla 0 a 1. Pokud přenášená informace obsahuje 8 bitů, dá se využít převod $8 \text{ bit} = 1 \text{ byte}$.

- **Blokové kódy**

Blokové kódy jsou kódy, které jsou uspořádány do bloků připomínajících matice, jejichž kódová slova mají totožnou délku a zabezpečení všech bitů je provedeno shodným počtem paritních bitů, jejichž počet je pevně daný. Kódovaná zpráva se poté může skládat z několika bloků o stejných parametrech.

- **Cyklické kódy**

Cyklické kódy jsou lineární kódy, jejichž kódová slova jsou uzavřena cyklickým posunem. Pokud u nich dojde k cyklické záměně prvků kódového slova, tak opět vzniká kódové slovo. Každé kódové slovo těchto kódů lze zapsat polynomem nad Galoisovým tělesem, přičemž tyto polynomy jsou násobky generujícího polynomu.

3.2 Paritní bity

Během přenosu informací sdělovacím systémem dochází vlivem šumu k poškození informace a přijatá informace by neodpovídala informaci odeslané. Proto se využívá mnoho způsobů a metod kódování, které mají za cíl detekovat a opravit poškozené bity. Mnoho metod proto používá paritní bity, které slouží k zabezpečení zprávy vhodnou redundantní informací. Paritní bity jsou známé již po několik desetiletí a slouží k identifikaci chyb v binární zprávě, díky nimž jsme schopni za určitých okolností zjistit, zdali byla informace přijata nepoškozená.

Paritní bity fungují na jednoduchém principu, kdy za bitovou zprávu je vložen další bit, který je nastaven do 1 či 0. O tom rozhoduje kódování sudou a lichou paritou, což závisí na konkrétních obvodových technologiích. Při použití sudé parity sečteme počet bitů s logickou 1 a dopočítáme paritní bit tak, aby byl počet jedniček sudý. V případě použití liché parity musí být součet všech jedniček lichý.

3.3 Hammingovy kódy

Hammingovy kódy jsou binární blokové kódy pojmenované po Richardu Hammingovi, který je objevil v roce 1940 a jedná se o zcela první samoopravné kódy. Tyto kódy jsou definovány minimální Hammingovou vzdáleností $d_{min} = 3$, která udává počet pozic, ve kterých se dva řetězce o stejné délce mohou lišit. Při této minimální Hammingově vzdálenosti dokáže kód detekovat dvě chyby anebo provést opravu právě jedné chyby.

Definice Hammingova kódu je následující: „Binární kód se nazývá Hammingův, jestliže má kontrolní matici, jejíž sloupce jsou všechna nenulová slova dané délky a žádné z nich se neopakuje“ [3]

3.3.1 Hammingův kód (7,4)

Tento typ Hammingova kódu patří mezi nejjednodušší a jsou na něm vysvětleny jeho

základní principy. Tento kód je tvořen 3 kontrolními bity m a 4 informačními bity k a tento kód je schopen opravit jednu chybu t . Tyto parametry se určují těmito vztahy:

- $d_{min} \geq 2t + 1$ minimální Hammingova vzdálenost
- $n = 2^m - 1$ délka kódového slova
- $k = n - m$ počet informačních bitů

Kontrolní bity se nachází na pozicích rovnající se násobkům dvěma, což je v tomto případě na 1., 2. a 4. pozici. Na ostatních pozicích, což jsou pozice 3, 5, 6 a 7, budou zobrazeny informační bity. Nejprve se vytvoří generující matice. „Generující matice se nazývá generující, pokud každý řádek je kódovým slovem, každé kódové slovo je lineární kombinací více řádků a řádky jsou lineárně nezávislé“ [4]

Na základě této formulace vytvoříme generující matici:

$$G_H = \begin{pmatrix} p_{11} & p_{21} & 1 & p_{31} & 0 & 0 & 0 \\ p_{12} & p_{22} & 0 & p_{32} & 1 & 0 & 0 \\ p_{13} & p_{23} & 0 & p_{33} & 0 & 1 & 0 \\ p_{14} & p_{24} & 0 & p_{34} & 0 & 0 & 1 \end{pmatrix} \quad (3.1)$$

Nyní je zapotřebí dopočítat jednotlivé paritní bity. Pro výpočet jednotlivých paritních bitů se použije následující soustava rovnic:

$$p_1 = x_3 \oplus x_5 \oplus x_7 \quad (3.2)$$

$$p_2 = x_3 \oplus x_6 \oplus x_7 \quad (3.3)$$

$$p_3 = x_5 \oplus x_6 \oplus x_7 \quad (3.4)$$

Každá z těchto rovnic obsahuje jediný paritní bit a tři informační bity. Výpočet parity, který zde probíhá obvyklým způsobem, se dá vyjádřit matematickou operací XOR, jejíž hodnota je 1, pokud obě hodnoty nabývají odlišných hodnot. Následným dosazením získáme úplný tvar generující matice

$$G_H = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \quad (3.5)$$

Nyní je potřeba zjistit kontrolní matici. Kontrolní matice představuje homogenní soustavu lineárních rovnic. Pokud kontrolní matici vynásobíme vektorem daného kódového slova a

výsledkem je nulový vektor, tak kódové slovo je nepoškozeno. [4]

Kontrolní matici získáme z následujících lineárních rovnic o 7 neznámých:

$$b_1 \oplus b_3 \oplus b_5 \oplus b_7 = 0 \quad (3.6)$$

$$b_2 \oplus b_3 \oplus b_6 \oplus b_7 = 0 \quad (3.7)$$

$$b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 0 \quad (3.8)$$

Na základě těchto rovnic se nyní vytvoří kontrolní matice pro Hammingův kód (7,4)

$$H_H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (3.9)$$

3.3.2 Dekódování

Dekódování a oprava chyb se zajišťuje výpočtem tzv. syndromu z kontrolní matice. Kontrolní matici H_H je třeba vynásobit zakódovanou zprávou a tím se vypočítá hodnota syndromu.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} \quad (3.10)$$

Pokud je výsledný vektor syndromu nulový, informace byla přenesena úspěšně a bez chyb. V případě nenulového syndromu se porovná jeho hodnota s jednotlivými sloupci kontrolní matice. Poloha daného sloupce nám určí, ve kterém bitu se nachází chyba a může být opravena. Tento kód je ovšem schopen opravit pouze jednu chybu. Pokud by došlo k vícenásobné chybě, syndrom bude nenulový, ale opravením dané pozice by došlo k vytvoření nové chyby. Vícenásobné chyby je tedy Hammingův kód (7,4) schopen pouze detekovat. [4]

3.4 BCH kódy

BCH kódy byly vynalezeny v letech 1959 a 1960 pány Hocquenghemem, Bosem a Ray-Chaudhurim, kteří vytvořili jejich konstrukci. Jsou to cyklické blokové kódy. Binární BCH kód existují pro každé celé číslo $m \geq 3$ a mají následující parametry:

- m počet kontrolních bitů
- t počet chyb, které chceme opravit
- $n = 2^m - 1$ maximální délka kódového slova
- $k \geq n - mt$ počet informačních bitů v kódovém slově
- $d_{\min} \geq 2t + 1$ minimální Hammingova vzdálenost [5]

BCH kódy jsou definovány generujícím polynomem. Jeho sestavení vychází z Bosého a Chaudhuriho teorému (dále je BC teorém):

„Při sestavování vytvářecího mnohočlenu $g(x)$ vycházíme z Bosého a Chaudhuriho teorému. Jestliže se tedy mezi kořeny vytvářecího mnohočlenu cyklického kódu (n, k) nachází určitý počet $(2t)$ prvků Galoisova tělesa jdoucích velikostí svých exponentů po sobě, pak minimální Hammingova vzdálenost d_{\min} není menší než $d = 2t + 1$.“ [5]

Pro zjištění generujícího polynomu BCH kódu se provede rozklad na minimální polynom nad daným Galoisovým tělesem GF. Kód určený pouze jediným minimálním polynomem o Hammingově vzdálenosti $d_{\min} = 3$ by byl schopen dle BC Teorému opravit jedinou chybu. Proto požadujeme delší Hammingovu vzdálenost a generující polynom se volí jako součin několika minimálních polynomů. Výsledný generující polynom je poté dán součinem jednotlivých minimálních polynomů:

$$g(x) = m_1(x) \cdot m_2(x) \cdot \dots \cdot m_n(x) \quad (3.11)$$

3.4.1 Kódování

Kódování BCH kódů se provádí využitím děličky cyklickým kódem. Polynom nezabezpečené zprávy je vynásoben polynomem x^{n-k} a následně vydělen generujícím polynomem. Z tohoto výrazu získáme polynom podílu a polynomu zbytku. Tento vztah lze vyjádřit rovnicí:

$$\frac{z(x) \cdot x^{(n-k)}}{g(x)} = m(x) + \frac{r(x)}{g(x)} \quad (3.12)$$

Tuto rovnici lze zjednodušit vynásobením $g(x)$ a získáme upravený vztah:

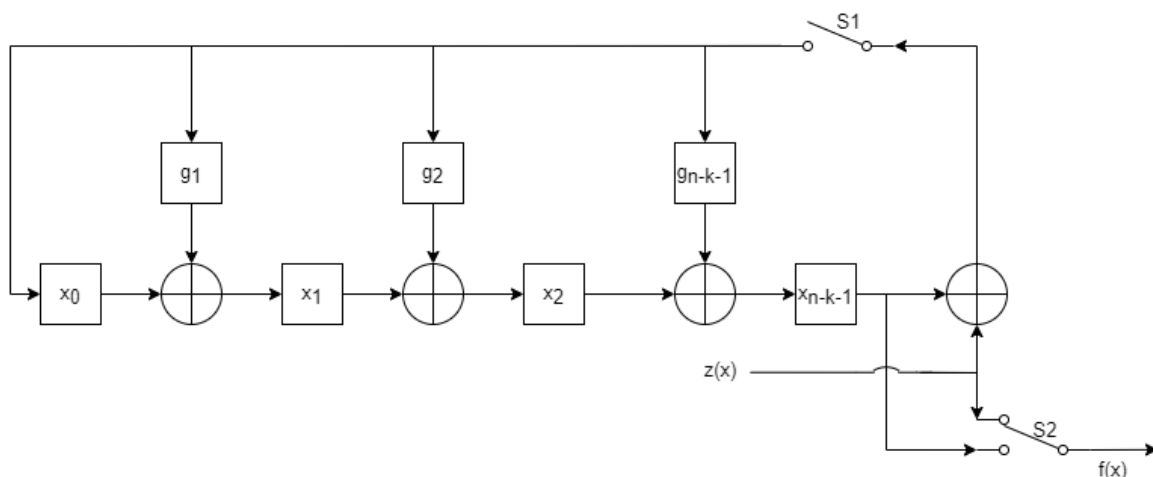
$$z(x) \cdot x^{(n-k)} = m(x) \cdot g(x) + r(x) \quad (3.13)$$

Nyní lze zabezpečený blok informace vyjádřit finálním vztahem

$$f(x) = z(x) \cdot x^{(n-k)} + r(x) \quad (3.14)$$

$$f(x) = m(x) \cdot g(x) + r(x) + r(x) = m(x) \cdot g(x) \quad (3.15)$$

Zabezpečení informace BCH kódem lze popsat pomocí posuvného registru se zpětnými vazbami a sčítačkami modulo 2. Počet paměťových buněk posuvného registru je roven rozdílu délky kódované informace a počtu informačních bitů v dané informaci. Polohy zpětných vazeb a sčítaček jsou určeny z generujícího polynomu. [6]



Obrázek 2: Schéma BCH kodéru. [5]

3.4.2 Dekódování

Dekódování má za cíl vytvořit přijatou zprávu shodnou s odeslanou a v případě odlišností objevit chyby a opravit je. Dekódování lze rozdělit do několika kroků. Nejprve se vytvoří syndromové rovnice, díky kterým se provádí kontrola mezi odeslanou a přijatou informací. V případě chyb budou některé z kořenů těchto rovnic nenulové a tím se zjistí poloha a počet chyb. Následně se opraví polynom s chybnou informací a dojde k vypočtení jeho správných hodnot. Po jejich vypočtení je již známa pozice chyb a nastane jejich opravení do správné hodnoty. [5]

Tyto principy dekodování používá spousta algoritmů. Mezi ty neznámější patří:

- Maticové dekodování
- Petersonův algoritmus
- Sugyamův algoritmus
- Berlekampův algoritmus

3.5 Reed-Solomonovy kódy

Reed-Solomonovy (dále jen RS) kódy jsou cyklické blokové kódy, které spadají do podskupiny BCH kódů a byly vyvinuty v 60. letech 20. století. RS kódy bývají označovány parametry n a k , kdy parametr k představuje celé číslo vyjadřující počet informačních bitů přenášené zprávy a n značí délku zabezpečené přenášené zprávy. RS dekodér dokáže detekovat stejné množství chybně přijatých bitů, jako je použito paritních bitů a opravit t chybných symbolů, kdy t je vyjádřeno rovnicí 3.16

$$2t = n - k \quad (3.16)$$

3.5.1 Kódování

Kódování probíhá obdobným způsobem jako u BCH kódů, kdy se nejprve vytvoří generující polynom nad Galoisovým tělesem. Generující polynom se skládá z $2t$ členů a je tvořen kódovými slovy ve tvaru

$$g(x) = (x - \alpha^0) \cdot (x - \alpha^1) \dots \cdot (x - \alpha^{2t-1}) = \prod_{i=0}^{2t-1} (x - \alpha^i) \quad (3.17)$$

kde α jsou primitivní prvky z množiny Galoisova tělesa a polynom nad Galoisovým tělesem dosahuje stupně $2t - 1$ nebo menšího. Pro samotný kódovací proces je zpráva převedena do vektorového tvaru

$$f(x) = f_0 + f_1x + \dots + f_{2t-1}x^{2t-1} = \sum_{i=0}^{2t-1} f_i x^i, \quad (3.18)$$

kdy jednotlivé koeficienty představují symboly Galoisova tělesa. Následný proces zakódování probíhá stejně jako u BCH kódů. Polynom zprávy je vynásoben polynomem x^{n-k} a následně vydělen generujícím polynomem. Z tohoto výrazu získáme polynom podílu a polynomu zbytku. Tento vztah je vyjádřen rovnicí 3.19. [7]

$$\frac{f(x) \cdot x^{(n-k)}}{g(x)} = m(x) + \frac{r(x)}{g(x)} \quad (3.19)$$

3.5.2 Dekódování

Dekódování má za cíl obnovit původní informaci, která může být vlivem rušení poškozená.

Původní zakódovaná zpráva byla odeslána ve tvaru

$$f(x) = f_0 + f_1x + f_2x^2 + \dots + f_{n-1}x^{n-1} \quad (3.20)$$

ovšem při přenosu dojde ke vzniku e chyby, která může být popsána polynomem

$$e(x) = e_0 + e_1x + e_2x^2 + \dots + e_{n-1}x^{n-1} \quad (3.21)$$

Přijátá zpráva je tedy součtem polynomů odeslané zprávy a chyby

$$v(x) = f(x) + e(x) = v_0 + v_1x + v_2x^2 + \dots + v_{n-1}x^{n-1} \quad (3.22)$$

kde kořeny $v(x)$ představují jednotlivé přijaté symboly. Následně se provádí výpočet jednotlivých syndromů. Syndromy se vypočítávají ze vztahu

$$S_j = v(a^j) = \sum_{i=0}^{n-1} v_i(a^j)^i \text{ pro } j = 1, 2, \dots, n-1 \quad (3.23)$$

kde α jsou primitivní prvky. Pokud by přijatá informace byla shodná s odeslanou informací, zakódovaná zpráva by byla beze zbytku dělitelná generujícím polynomem a výsledný syndrom by byl nulový. Pro zjednodušení se využívá úprava, kdy

$$X = \alpha^i \quad (3.24)$$

$$Y = e_i \quad (3.25)$$

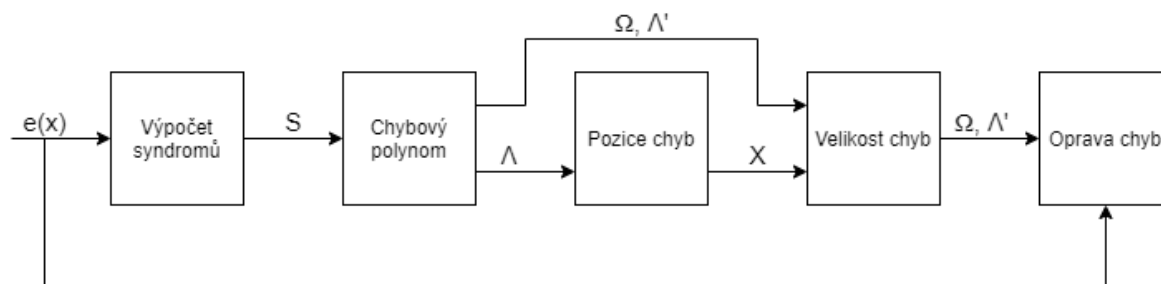
Hodnoty X se nazývají pozice chyb a hodnoty Y definují hodnoty chyb. Následně je vypočítán lokalizační člen. Tento polynom je určen vztahem 3.26 [8]

$$\Lambda = \prod_{l=1}^v (1 - xX_l) \quad (3.26)$$

Nyní zbývá nalézt jeho kořeny a vypočítat mnohočlen vyčíslení chyb, určit chybové symboly a následně provést patřičné opravy. Toho lze docílit následujícími algoritmy a metodami:

- Berlekamp - Masseyho metoda
- Euklidovský algoritmus
- Chienovo vyhledávání
- Forneyův algoritmus

- Petersonův algoritmus



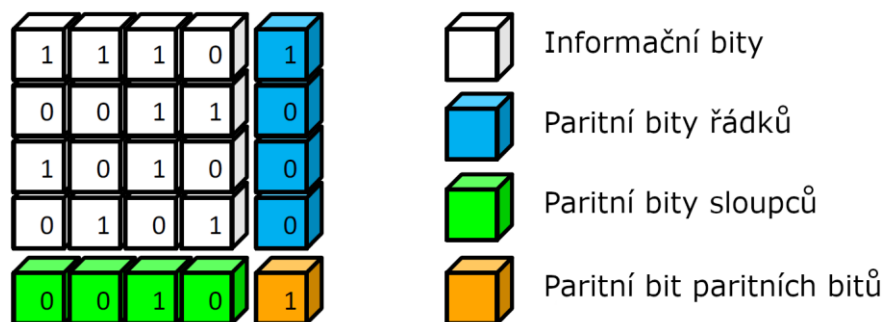
Obrázek 3: Obecné schéma dekódování RS kódů. [7]

4 Multidimenzionální paritní kódy

Multidimenzionální paritní kódy (MDPC) spadají do skupiny blokových kódů a tato metoda může slibovat dobré výsledky dekódování a chybovosti a mohla by být používána pro komunikaci CubeSatu se zemí. Využití křížové parity (2D paritní kód) je metoda známá po mnoho let pro její jednoduché kódování a dekódování, kdy se používala jako jednoduchý detekční a korekční kód pro zabezpečení záznamu na magnetických nosičích. Nedostatečný výpočetní výkon v době jejich používání zabránil jejich rozšíření do vyšších dimenzí a postupně se FEC kódování začalo ubírat jiným směrem (Reed-Solomonovy kódy, BCH kódy, konvoluční kódy). Paritní kódy se však v posledních letech s nárůstem výpočetních výkonů procesorů vrátily v podobě LDPC kódů do moderních sdělovacích systémů. Rozšíření multidimenzionálních paritních kódů do více dimenzí by mohlo nabídnout podobně dobré detekční a korekční schopnosti.

4.1 Dvourozměrné paritní kódy

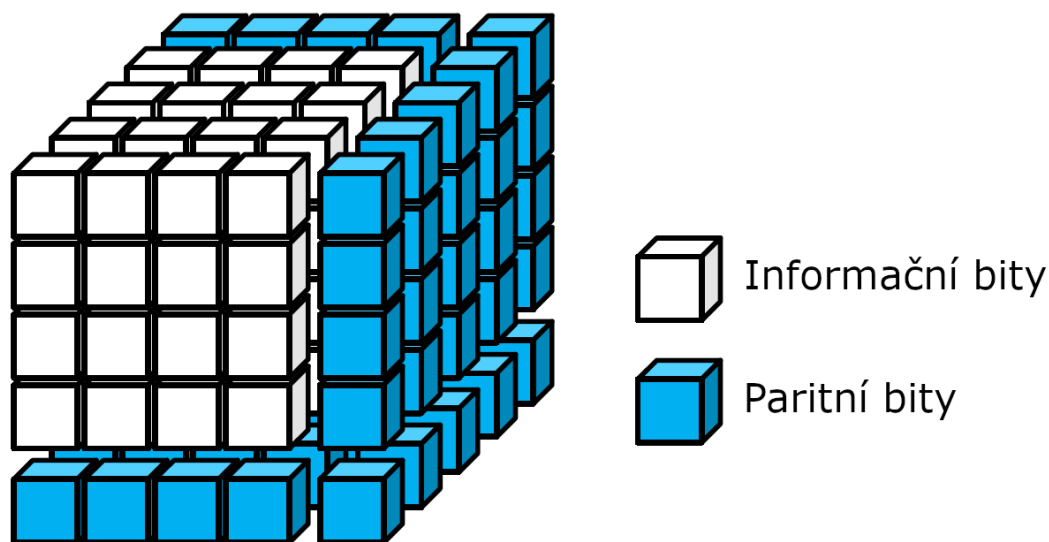
Jak lze vidět na obrázku 4, blok informačních bitů je umístěn do dvourozměrné matice, kdy poslední řádek a sloupec jsou tvořeny paritními bity, které kontrolují počet jedniček a nul v daném řádku či sloupci. Po určení paritních bitů je určen paritní bit paritních bitů, přičemž v tomto příkladu je použito kódování sudou paritou. Nicméně nevýhoda této možnosti nastává při sudém počtu chyb v daném řádku či sloupci, kdy paritní bit zůstane ve stejné pozici jako při odeslání a kontrola parity selže pouze v jedné dimenzi.



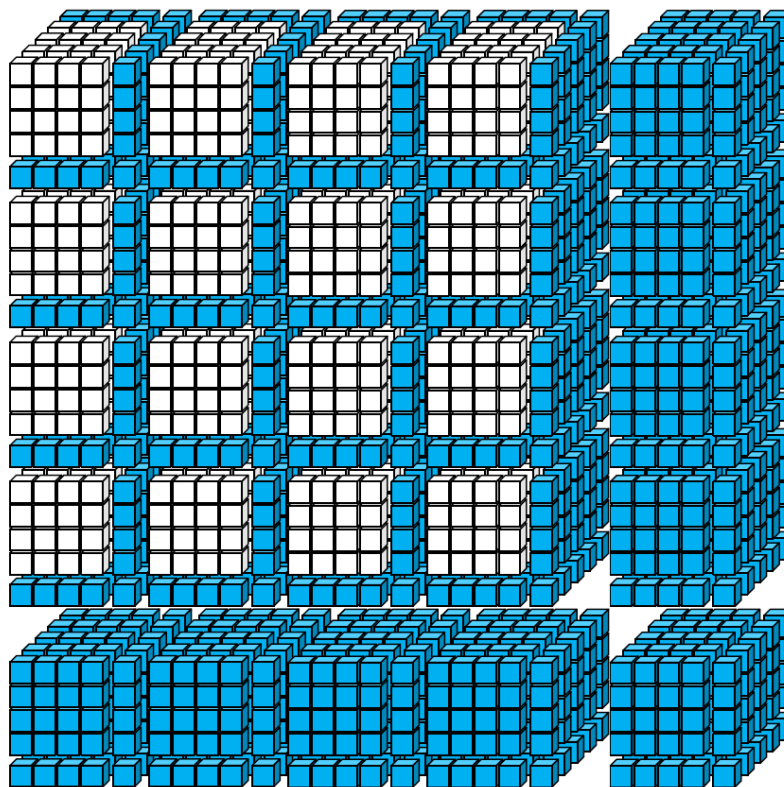
Obrázek 4: Výpočet paritních bitů pro kódování ve 2D. [9]

4.2 Vícerozměrné paritní kódy

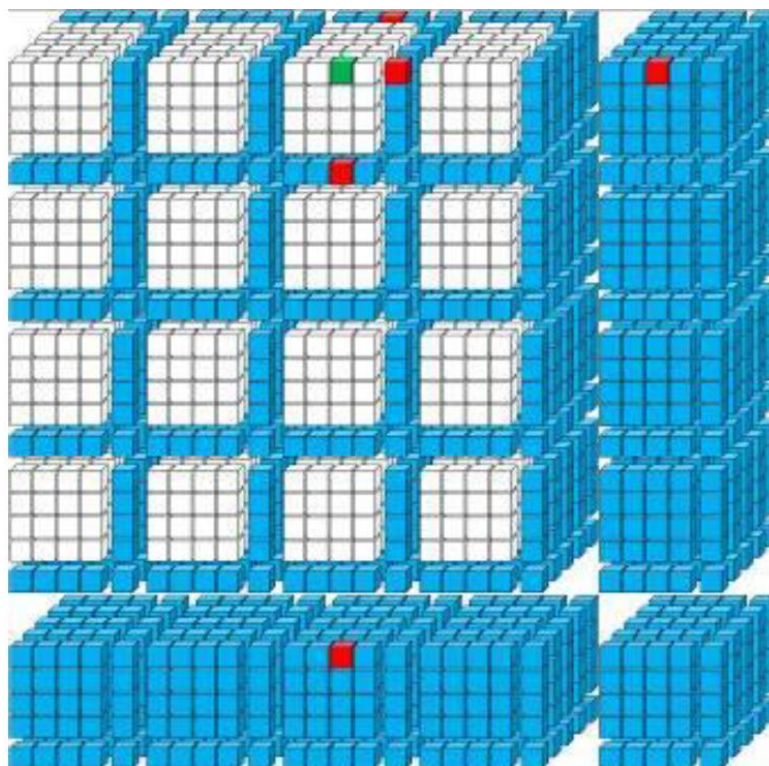
Citlivost náhodných chyb můžeme omezit rozšířením paritních kódů do obecně m -dimenzionálního prostoru. Multidimenzionální paritní kódy lze sestavit jednotlivým skládáním bloků paritních a informačních bitů do vícerozměrné krychle. Ve třetí dimenzi jsou za dvourozměrné paritní bloky postupně vrstveny další bloky o stejné velikosti čímž vznikne krychle. Další dimenze lze tvořit obdobným vrstvením dalšími bloky. V případě páté dimenze by bylo poskládáno x paritních krychlí do každého řádku a sloupce. [9]



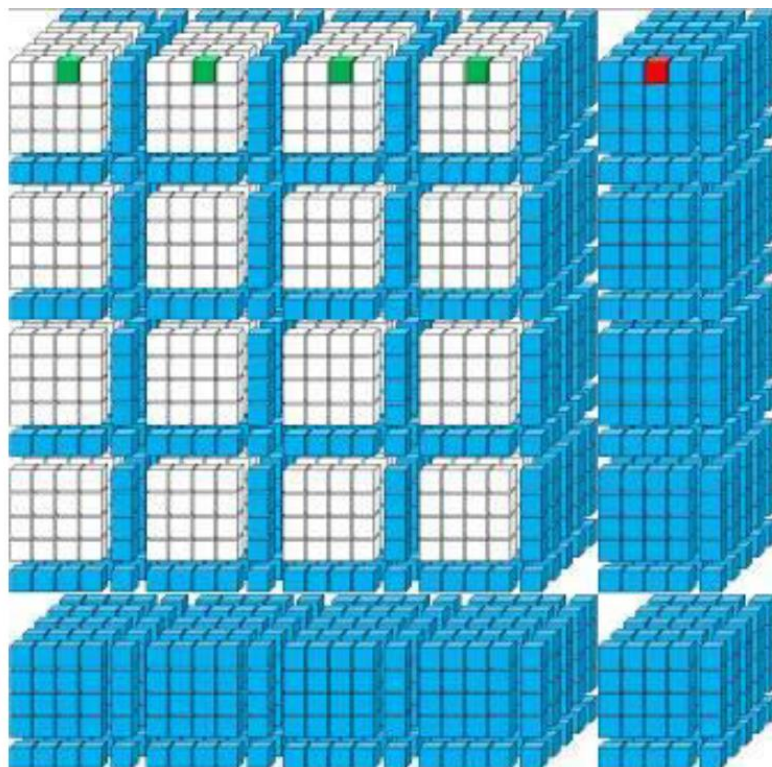
Obrázek 5: Princip zabezpečení 3D bloku dat. [9]



Obrázek 6: Krychle pro určení křížové parity v 5D. [9]



Obrázek 7: Paritní bity kontrolující informační bit v 5D. [9]



Obrázek 8: Informační bity kontrolované paritním bitem v 5D. [9]

Multidimenzionální paritní kódy lze zkráceně označovat ve tvaru MDPC(x D/ y L), kdy x představuje počet dimenzí a y délku jednoho kódového slova. Dále lze říct, že každý paritní bit kontroluje y informačních bitů a x paritních bitů kontroluje právě jeden informační bit. Z takto zadaných údajů lze zjistit počet informačních bitů na vstupu do kodéru, který je dán vztahem

$$k = y^x \quad (4.1)$$

Další rovnice ukazuje celkový počet bitů zabezpečených dat na výstupu z kodéru

$$n = (y + 1)^x \quad (4.2)$$

Z těchto vztahů lze následně vypočítat kódový poměr (Code Rate), který odpovídá vztahu

$$R = \frac{k}{n} = \frac{y^k}{(y + 1)^x} \quad (4.3)$$

Z těchto rovnic lze vidět, že se vzrůstajícím počtem dimenzí, které poskytují zabezpečení kódu, klesá kódový poměr. Kódový poměr může být zvýšen využitím delších slov s větším

počtem informačních bitů, ovšem tím dojde ke zvýšení nároků na výkon a zpomalení výpočetního cyklu, protože se zvětšuje celková délka zabezpečeného bloku dat. [10] V tabulce 1 lze vidět, jak s rostoucím počtem dimenzí a klesajícím počtem informačních bitů klesá kódový poměr.

Tabulka 1: Kódový poměr v závislosti na počtu dimenzí a počtu informačních bitů. [9]

		x – Počet dimenzí				
		3	4	5	6	7
y – počet informačních bitů kódového slova	1	0,125	0,063	0,031	0,016	0,008
	2	0,296	0,198	0,132	0,088	0,059
	3	0,422	0,316	0,237	0,178	0,133
	4	0,512	0,410	0,328	0,262	0,210
	5	0,579	0,482	0,402	0,335	0,279
	6	0,630	0,540	0,463	0,397	0,340
	7	0,670	0,586	0,513	0,449	0,393
	8	0,702	0,624	0,555	0,493	0,438
	9	0,729	0,656	0,590	0,531	0,478

4.3 Kódování

Proces kódování má za cíl zabezpečit informační bity vhodným paritním bitem, aby bylo dodrženo předem určené pravidlo sudé či liché parity. V případě shodného počtu informačních bitů v každé dimenzi platí vztahy z kapitoly 4.2. Jestliže bity nejsou vrstveny do krychlí a v každé dimenzi se nachází rozdílný počet informačních bitů, délka kódového slova je dána vícerozměrným vektorem a různé informační bity mohou být chráněny odlišným počtem paritních bitů.

Jedna z možností kódování spočívá ve využití blokového uspořádání informačních bitů. Ty jsou umístěny do bloků v závislosti na počtu dimenzí. Následně je postupně od nejvyšší dimenze po nejnižší provedeno modulo 2 informačních bitů pro určení správné hodnoty paritního bitu, např. u třírozměrného bloku bitů se nejprve určí paritní bity v zadní stěně krychle, která je tvořena pouze paritními bity, a až poté se určí zbylé paritní bity.

Druhá možnost také využívá modulo 2, ovšem vektor obsahující informaci je vynásoben s generující maticí, čímž vznikne kódové slovo. Generující matice může být vytvořena

následujícím algoritmem, kde r značí vícedimenzionální vektor, N je počet dimenzí kódu a symbol \otimes značí Kroneckerův produkt. Funkce *eye* vytvoří jednotkovou matici o zvolených rozměrech a funkce *ones* matici vyplněnou 1 na všech pozicích. [11]

```

function generate_G( $r$ )
{
     $G = [\text{eye}(r_1) \mid \text{ones}(r_1, 1)]$ ;
    for( $n = 2$ ;  $n \leq N$ ;  $n++$ )
    {
         $G_{tmp} = [\text{eye}(r_n) \mid \text{ones}(r_n, 1)]$ ;
         $G = G_{tmp} \otimes G$ ;
    }
    return  $G$ ;
}

```

Obrázek 9: Funkce pro vytvoření generující matice. [11]

4.4 Dekódování

U dvourozměrné matice je dekodování jednoduché. V případě, že došlo k chybě v paritním bitu ve sloupci i řádku zároveň, předpokládaná chyba je detekována a daný bit je invertován. Pokud ovšem nastala chyba pouze v řádku či sloupci, chyba bude detekována, ale nelze zjistit její přesná poloha.

V případě využití vyššího počtu dimenzí kontroluje danou pozici několik paritních bitů, kdy každá pozice je kontrolována tolika paritními bity, kolik je použito dimenzí. Paritní bit může tedy detekovat chybu pouze v jedné či ve všech dimenzích. Z tohoto důvodu se pro každý bit vypočítá pravděpodobnost chyby (dále jen FDM), která může v tomto případě nabývat hodnoty 0 pro nejnižší pravděpodobnost výskytu chyby pro daný bit, který by měl být ve správné pozici, až po hodnotu shodnou s počtem dimenzí, kdy má daný bit největší pravděpodobnost chyby a bude následně invertován. Proto se nejprve vypočítá FDM pro celý blok informací a zjistí se, které bity jsou s největší pravděpodobností poškozené a invertované. Celý proces začne probíhat iterativně, dokud nejsou hodnoty FDM ideálně nulové. Nicméně mohou nastat bloky dat, ve kterých se dekodování nepovede ukončit, a proto je zapotřebí vždy určit limitní množství iterací, ve kterých dochází k opravě poškozených bitů.

Pro testování dekódování se v této bakalářské práci vychází z těchto 2 metod:

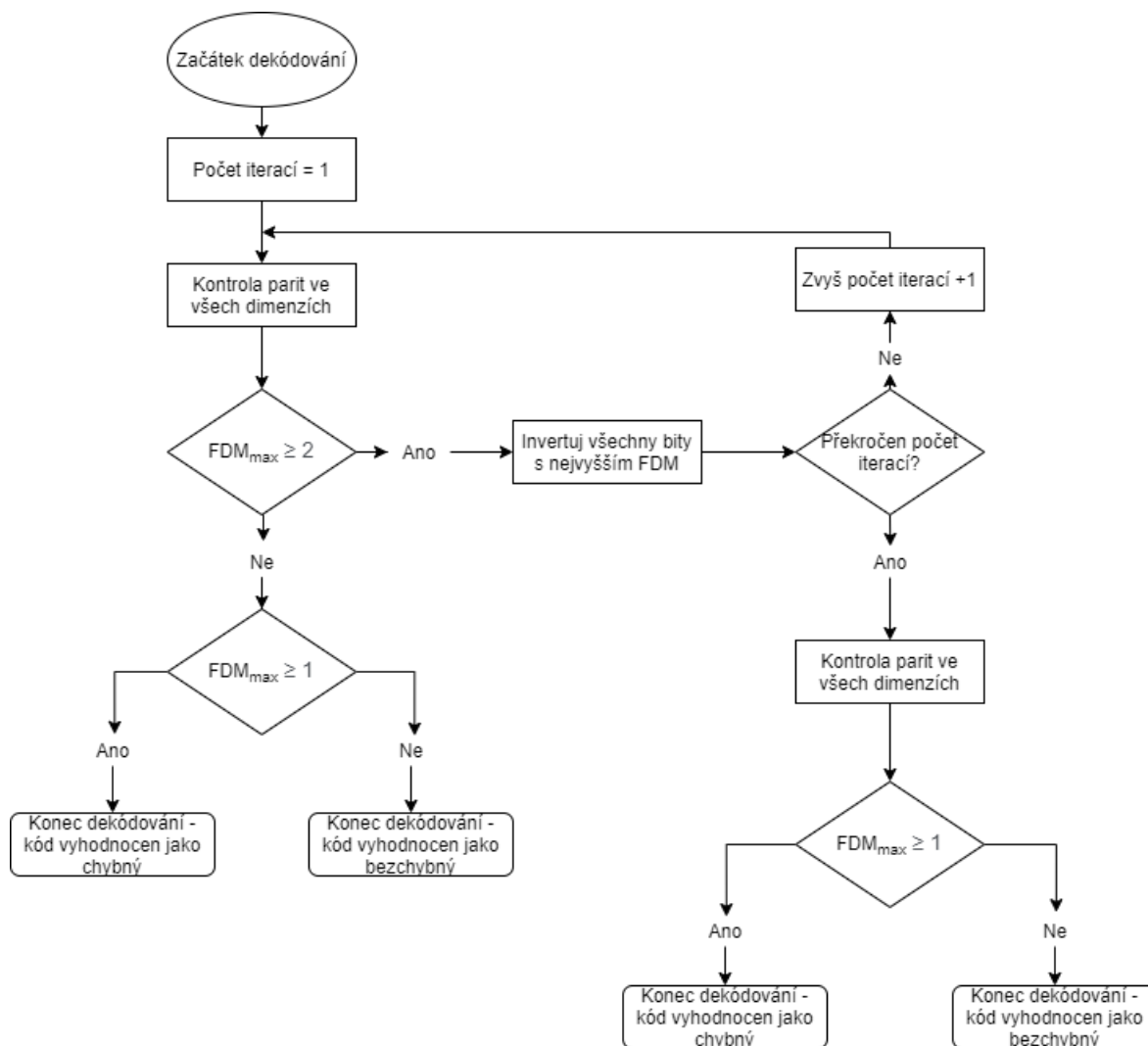
- **Bez váhového koeficientu**

V tomto případě je váhový koeficient stejný pro všechny bity a váha všech bitů je neměnná a konstantní. V každé iteraci je vypočítána FDM všech bitů a všechny bity s nejvyšší hodnotou FDM jsou invertovány. Po invertování bitů se opět vypočítají nové hodnoty FDM a invertují bity s nejvyšší hodnotou. Dekódování končí při překročení limitu iterací či po dosažení nízkých či nulových hodnot FDM, kdy je pravděpodobnost výskytu chyby nízká či nepravděpodobná.

- **S váhovým koeficientem**

V této metodě se vypočítá hodnota FDM a ta je vynásobena váhovým koeficientem na dané pozici a invertován je pouze první bit s nejvyšší hodnotou FDM. Po invertování bitu proběhne opět výpočet FDM vynásoben váhovým koeficientem a pokud je součet bloku FDM nižší než součet FDM před opravou, oprava je přínosná a bit zůstane invertován a postoupí se k dalšímu bitu. Jestliže je součet bloku hodnot FDM vynásobených váhovým koeficientem po opravě vyšší než před invertováním bitu, bit je nastaven zpět do původní hodnoty a pozice daného bitu je vynásobena koeficientem 0,9 a v dalších iteraci bude danému bitu kladena nižší důležitost. Proces dekódování končí, když se překročí limit iterací nebo po dosažení nízkých hodnot FDM, kdy je nízká pravděpodobnost výskytu chyby. [11]

Dekódování v obou případech vychází z výpočtů FDM a je vyhodnoceno jako bezchybné, pokud maximální hodnota FDM je rovna 0. Jestliže se na některé pozici nachází hodnota $FDM \geq 1$ nebo je překročen limit iterací, dekódování je vyhodnoceno jako chybné.



Obrázek 10: Vývojový diagram použití iterativní metody. [10]

5 Praktická část

Praktickou částí bakalářské práce bylo vypracovat kodér a dekodér v programu MATLAB alespoň o 3 dimenzích a následně analyzovat zjištěné výsledky. Nejprve byla vytvořena hlavní funkce s testovací smyčkou, ve které testování vychází z poměru chybně přenesených bitů k celkovému počtu přenesených bitů informace pro různou míru přidaného šumu a tím i snížení kvality přenášeného signálu. V té jsou nejprve zjištěny hodnoty pro vykreslení křivky nekódované BPSK modulace v závislosti na poměru energie jednoho bitu ku spektrální výkonové hustotě šumu (E_b/N_0) a její chybovosti (dále jen BER). Začátkem této funkce jsem si určil počet dimenzí multidimenzionální matice a délku strany bloku dat. Následně jsem si nadefinoval jednotlivé proměnné určující chybovost demodulátoru či všech tří metod dekódování – bitovou chybovost (BER), rámcovou chybovost (FER) a

nerozpoznanou chybovost rámců (UFER).

Rámcová chybovost (FER) odpovídá počtu chybně přenesených a dekodovaných bloků na výstupu dekodéru vůči celkovému počtu přenesených dat.

Nerozpoznaná chybovost rámců (UFER) odpovídá množství rámců, u kterých bylo dekodování ukončeno s tím, že kontrola parity ve všech směrech sedí, ale dekodovaný blok neodpovídá zakódovanému bloku dat.

5.1 Vytvoření bloku dat

Prvním krokem v každém výpočetním cyklu je vytvoření bloku dat. Blok dat je vytvořen z počtu dimenzí a počtu informačních bitů v každé straně, kdy každá strana má délku počet informačních bitů + 1 paritní bit. Daný blok dat je vytvořen MATLAB funkcemi *logical* a *randi*. Funkce *logical* vytváří blok logických hodnot, ve kterém jsou všechny nenulové prvky převedeny na logickou 1 značící pravdivou hodnotu a nulové prvky na logickou 0 značící nepravdivou hodnotu. Funkce *randi* vytváří náhodný blok celých čísel ve zvoleném rozsahu. V každém cyklu je vytvořen nový blok náhodných dat funkcí *bp_Gendata.m*, která je volána každým cyklem.

5.2 Kódování

Po vytvoření bloku dat následovalo jeho kódování, které je implementováno ve funkci *bp_CrossParityBlock*. Nejprve bylo vytvořeno pomocné pole o délce odpovídající počtu dimenzí a na každé pozici tohoto pole se nachází vnořené pole o délce odpovídající délce strany bloku dat. Poté jsou nastaveny nejprve všechny paritní bity do 0 a následně je provedeno jejich nastavení do správné hodnoty. Toho je docíleno nastavením paritního bitu, který se nachází na poslední pozici každé strany bloku dat. V této bakalářské práci je prováděna kontrola lichou paritou, kdy součet jedniček v každé straně musí být lichý. Určení parity probíhá postupně skrz všechny dimenze od nejvyšší po nejnižší.

y1				y2			
0	1	1	1	0	1	1	1
0	0	0	1	0	0	0	1
1	1	1	0	1	1	1	0
1	1	0	1	0	1	1	1

Obrázek 11: Blok dat před nastavením (y1) a po nastavení (y2) paritního bitu.

5.3 Modulace a demodulace

Nejprve je generován signál BPSK modulace. Toho je docíleno funkcí *bp_Psk*, ve které se generuje dvoustavově modulovaný signál pro logickou 0 a 1. Následně je simulován přenos dat funkcí *bp_TransmitDataPsk*. Do této funkce vstupují vypočítané hodnoty pro BPSK modulaci (s_0 , s_1), multidimenzionální zakódovaná matice y_2 a šum odpovídající poměru výkonu signálu ku šumu snr . Tyto hodnoty vstupují do cyklu, který postupně bere jednotlivé bity bloku dat a je k nim přidán šum v závislosti na logické 0 či 1 a hodnotě snr . Toho je dosaženo funkcí *awgn*, která přidává do signálu Gaussovský bílý šum. Poté je volána funkce *bp_PskDem* pro demodulaci, ve které je signál demodulován zpět na jednotlivé bity, avšak s tím rozdílem, že některé bity se nenachází ve správné hodnotě kvůli přidanému šumu.

```

1  function [y3]=bp_TransmitDataPsk(y2,snr,s0,s1)
2  y3=y2;
3  b=single(zeros(length(s0)));
4  c1=numel(y2);
5
6  for i=1:c1
7      switch y2(ind2sub(size(y2),i))
8          case (0),
9              b=awgn(s0,snr,0);
10             [y3(ind2sub(size(y2),i))]=bp_PskDem(b);
11          case (1),
12             b=awgn(s1,snr,0);
13             [y3(ind2sub(size(y2),i))]=bp_PskDem(b);
14         end;
15     end;

```

Obrázek 12: Funkce pro přenos dat s využitím BPSK modulace a demodulace.

5.4 Dekódování

Dekódovací proces je rozdělen do 3. samostatných funkcí, ve kterých jsou využity různé možnosti dekódování a má za cíl objevit poškozené bity v bloku dat a invertovat je do jejich správné polohy. V případě úspěšného dekódování se dekódovaný blok dat shoduje se zakódovanými daty. Do všech dekódovacích algoritmů jsou odeslány parametry s přenášenými daty, počtem dimenzí a počtem informačních bitů v jedné straně multidimenzionální matice. Výstupem této funkce jsou dekódované vícerozměrné matice, počet nenulových prvků pomocné matice značící chybu parity a počet iterací, po kterých bylo dekódování ukončeno.

5.4.1 Dekódování bez váhových koeficientů

Tento dekódovací algoritmus je implementován ve funkci *bp_Decoding.m* a jedná se o nejrychlejší z uvedených metod dekódování, jelikož opravuje všechny bity na pozicích s maximální hodnotou pomocné matice, ve kterých selhala kontrola parity. Nejprve jsou vytvořeny dvě pomocné matice e a b o stejných rozměrech, jako je dekódovaná matice a jsou naplněny 0. K tomu jsou využity funkce *ones* a *zeros*, kdy je nejprve vytvořena matice jedniček o rozměrech dekódované matice a následně jsou všechny hodnoty nastaveny do nuly. Následný proces dekódování je tvořen dvěma cykly. První cyklus slouží k zabránění velkému počtu opakování v případě neopravitelných chyb a druhý postupně od nejvyšší dimenze po nejnižší kontroluje hodnotu paritního bitu v každém řádku a sloupci. Pokud hodnota parity v některé straně nesedí, v pomocné matici b dojde k inkrementaci o 1 na dané pozici. Po provedení kontroly parity v nejvyšší dimenzi jsou hodnoty přičteny do pomocné matice e a matice b je vynulována a cyklus pokračuje kontrolou parity v dimenzi, která je o jednu nižší. Po průchodu nejnižší dimenzí zůstávají uloženy pozice, ve kterých selhala kontrola parity v matici e a funkcí *max* je zjištěna její nejvyšší hodnota, která může dosahovat stejné hodnoty, jako je počet dimenzí. Následně jsou invertovány bity na všech pozicích, ve kterých dosahuje pomocná matice maxima. V dalším cyklu se opět provede kontrola parity a bity se invertují na pozicích, ve kterých dosahuje pomocná matice maxima. Dekódování končí, pokud dojde k překročení přednastaveného počtu iterací v prvním cyklu anebo když je maximální hodnota pomocné matice příliš nízká a pravděpodobnost chyby na každé pozici je nízká či zcela nulová. Pokud je maximální hodnota pomocné matice nulová, byly opraveny všechny bity a kontrola parity sedí napříč celou maticí.

5.4.2 Dekódování s váhovými koeficienty

Tento dekódovací proces je implementován ve funkci *bp_DecodHard.m* a je výpočetně nejnáročnější z algoritmů uvedených v této práci. V této metodě se vytvoří 3 pomocné matice e , $e1$ a b o rozměrech dekódovací matice a jsou naplněny 0. Dále je vytvořena matice *hard_prob* naplněná 1 charakterizující váhové koeficienty. Následný dekódovací proces je tvořen 3 cykly. První slouží k určení limitu počtu iterací v případě neopravitelného bloku dat. Druhý cyklus porovnává přínos opravy a třetí slouží ke kontrole parity. Po vypočtení hodnot kontrolní matice e shodným způsobem jako u dekódování bez váhových koeficientů nastane její vynásobení po prvcích s maticí *hard_prob*. Tato hodnota je uložena do matice $e1$ a je objevena pozice první maximální hodnoty této matice a bit na dané pozici je invertován. Následně jsou opět zjištěny hodnoty kontrolní matice e vynásobené po prvcích pravdivostní maticí. Nyní se vypočítá součet kontrolní matice $e1$, ve které jsou uloženy hodnoty před změnou a kontrolní matice e značící chybné hodnoty po invertování bitu. Pokud je celkový součet kontrolní matice po invertování bitu nižší než před invertováním, změna je přínosná a bit zůstane invertován. V opačném případě se dá předpokládat, že bit je nastaven do správné hodnoty a je invertován zpět do původní hodnoty a tato pozice je v matici znázorňující váhové koeficienty vynásobena koeficientem 0,9. Tím se na této pozici nebude nacházet v dalším cyklu maximum a postoupí se k dalšímu bitu. Dekódování končí po dosažení definovaného počtu iterací nebo pokud je maximální hodnota v kontrolní matici příliš nízká či nulová, kdy není žádný bit poškozen nebo pravděpodobnost chyby je příliš nízká.

5.4.3 Dekódování při kombinaci obou metod

Jak už vypovídá název, tato metoda využívá kombinaci metod popsaných výše a nachází se ve funkci *bp_DecodingMix.m*. Tato metoda vychází z algoritmu s váhovými koeficienty, kdy se nejprve vytvoří kontrolní matice e , $e1$, b a matice váhových koeficientů *hard_prob*. Rozdíl nastává v dekódování, kdy je nejprve použita první metoda pro invertování všech prvků na pozicích, na kterých se nachází maxima kontrolní matice, které jsou rovny s počtem dimenzí. Následně jsou po 5 prvcích zároveň invertovány bity na pozicích, ve kterých dosahuje maximum hodnoty $dimenze - 1$ za předpokladu, že se jedná o matici s vyšším počtem dimenzí než 3. Po invertování bitů splňujících tyto dvě podmínky probíhá dekódování jednotlivě po maximálních prvcích jako v algoritmu s váhovými koeficienty. Jediný rozdíl nastává po 100 cyklech, kdy jsou nastaveny všechny prvky matice s váhovými

koeficienty do hodnoty 1, čímž budou prvky, u kterých nebylo invertování bitu přínosem opět testovány. Dekódování končí, stejně jako u předešlých algoritmů, po dosažení definovaného počtu iterací nebo za předpokladu, že je maximální hodnota v kontrolní matici příliš nízká či nulová, kdy není žádný bit poškozen s vysokou pravděpodobností či vůbec.

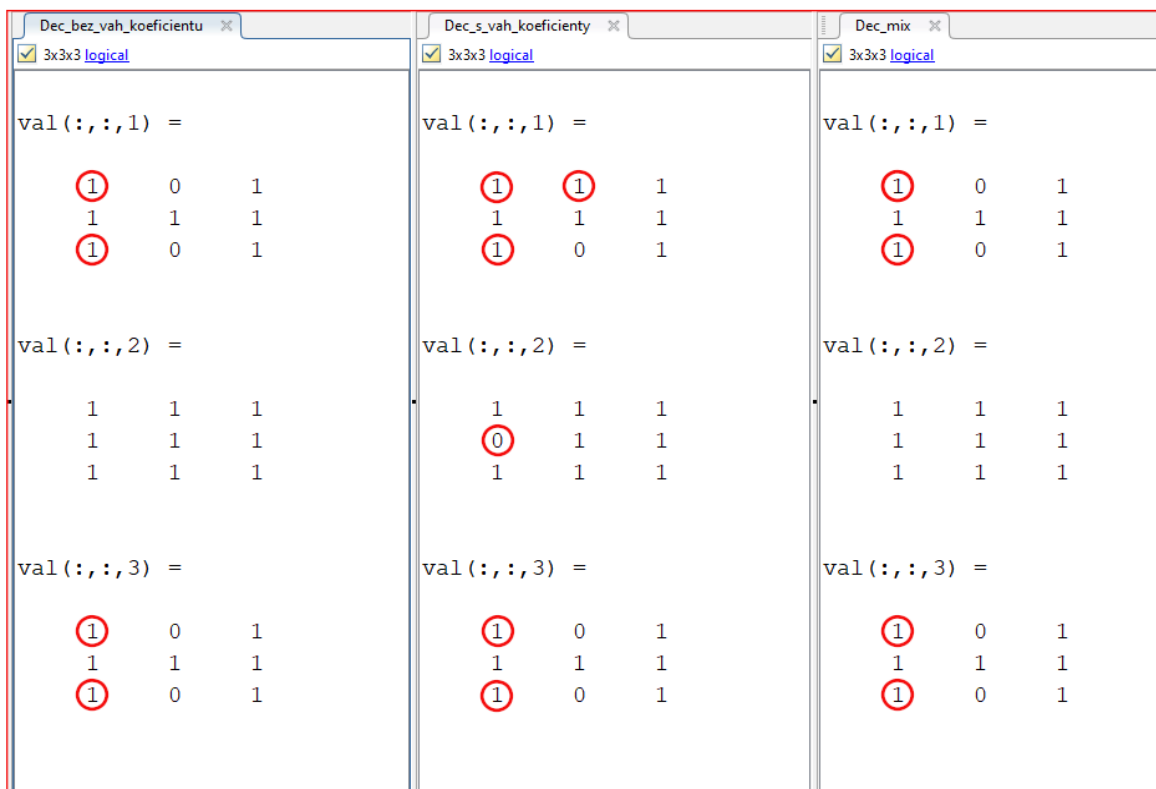
5.5 Chyby dekodování

5.5.1 Charaktery chyb

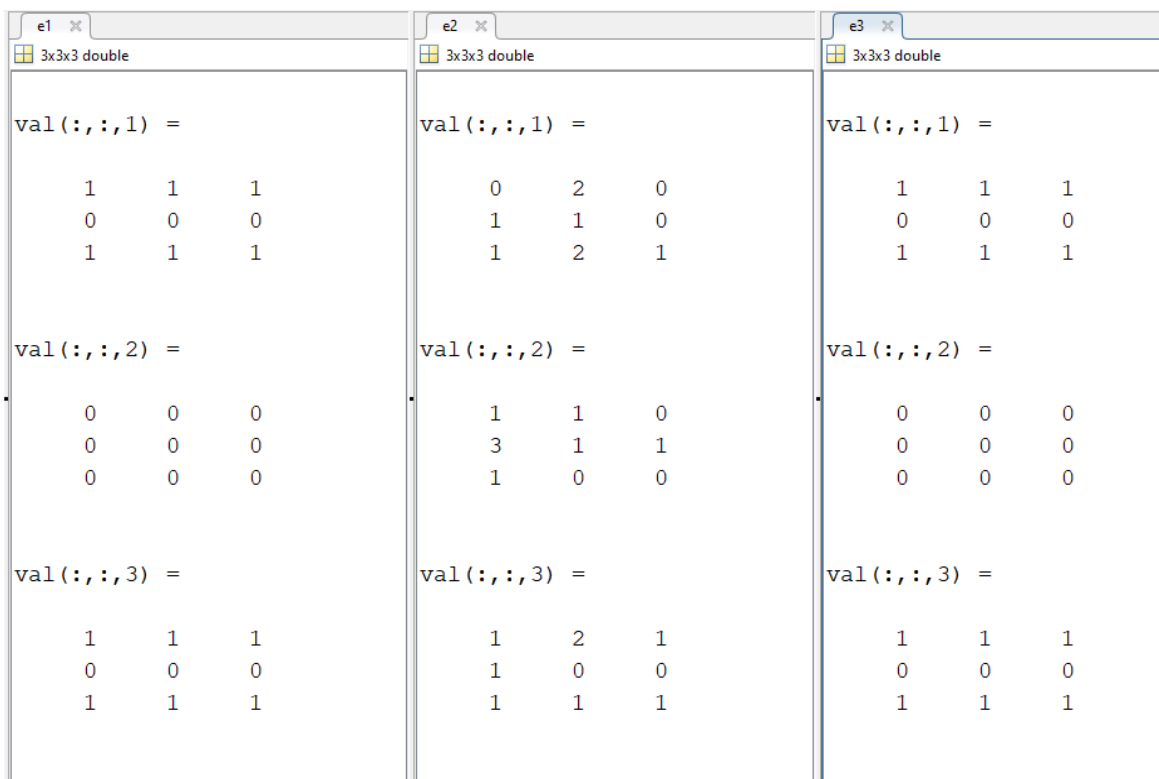
Dekódovací algoritmy nejsou bezchybné a jejich chybovost je se vzrůstajícím počtem dimenzí nižší. Nejnižší dimenze prováděného měření byla 3. a jedná je o jedinou dimenzi, kterou si lze prostorově představit, a proto se v této části zabývám pouze touto dimenzí. Na obrázku 13 je znázorněn jeden z přenášených bloků dat s poškozenými bity, kde y_2 je zakódovaná matice, y_3 je matice poškozená šumem a kontrolní matice e znázorňuje chybu při kontrole parity u poškozené matice před začátkem dekodování.

y2	y3	e
<pre> 3x3x3 logical val(:, :, 1) = 0 0 1 1 1 1 0 0 1 val(:, :, 2) = 1 1 1 1 1 1 1 1 1 val(:, :, 3) = 0 0 1 1 1 1 0 0 1 </pre>	<pre> 3x3x3 logical val(:, :, 1) = 0 1 1 1 1 1 1 0 1 val(:, :, 2) = 1 1 1 0 1 1 1 1 1 val(:, :, 3) = 1 0 1 1 1 1 1 0 1 </pre>	<pre> 3x3x3 double val(:, :, 1) = 3 3 1 2 1 0 2 2 1 val(:, :, 2) = 2 1 0 3 1 1 1 0 0 val(:, :, 3) = 2 2 1 1 0 0 1 1 1 </pre>

Obrázek 13: Blok dat po zakódování a po přidání šumu.



Obrázek 14: Dekódované bloky dat pro jednotlivé algoritmy po první iteraci.



Obrázek 15: Hodnoty kontrolních matic pro jednotlivé dekódovací algoritmy po první iteraci.

Na obrázku 14 je zobrazen průběžný výsledek dekódování po první iteraci pro jednotlivé algoritmy s vyznačenými invertovanými bity oproti zakódované matici a na obrázku 15 jsou zobrazeny hodnoty kontrolní matice algoritmů po první iteraci. Pro algoritmus bez váhových koeficientů (levá část výsledků) a algoritmus s kombinovanou metodou (pravá část výsledků) je maximální hodnota kontrolních matic již po první iteraci příliš nízká a dekódovací algoritmy skončí z důvodu nízké pravděpodobnosti chyby na všech pozicích. Došlo tedy k invertování pouze bitů na pozicích, ve kterých nabývá kontrolní matice e na obrázku 13 maximální hodnoty. Naopak druhý algoritmus využívá váhové koeficienty a opravuje v každém kroku pouze jediný bit za předpokladu, že to pomůže výsledné paritě. V tomto případě došlo k invertování bitu na pozici, která lze zapsat v MATLABu jako $Dec_s_vah_koeficienty(1,1,1)$, ačkoliv byl tento bit ve správné poloze. V další iteraci by došlo k invertování bitu na pozici, která je shodná s maximální hodnotou kontrolní matice na obrázku 15 na pozici $e2(2,1,2)$, a proto by byl invertován bit na pozici $Dec_s_vah_koeficienty(2,1,2)$, pokud by tato změna bylo prospěšná výsledné paritě. Takto by dekódovací algoritmus pokračoval, dokud by nenastaly nízké maximální hodnoty kontrolní matice toho algoritmu jako u předchozích algoritmů.

Tento blok dat je jedním z bloků, ve kterém selhaly všechny tři algoritmy dekódování. Dekódovací algoritmy mají vcelku vysokou úspěšnost a většinu rámců jsou schopny správně dekódovat, nicméně mohou se objevit určité vzory chyby, které nejsou vždy schopny opravit a tyto chyby budou ukázány na obrázku 13.

- **Chyba přes všechny dimenze s průsečíkem v daném bodě**

První ze zásadních problémů nastává u chyby ve všech směrech s průsečíkem v jednom bodě, kdy je daný bit považován za chybný i přes jeho správnou hodnotu. V tomto případě nabývá kontrolní matice maximální hodnoty 3 a tento bit je považován za poškozený s vysokou pravděpodobností. Tato chyba se nachází na pozici, která lze v MATLABu vyjádřit jako $y3(1, 1, 1)$.

- **Návazné dvojnásobné chyby**

Druhý zásadní problém při dekódování je způsoben dvojicí chyb v jednom směru, kdy tato dvojice má společný poškozený bit s jinou dvojicí poškozených bitů. Tyto dvě návazné dvojice lze vyjádřit jako $y3(1, :, 3)$ a $y3(3, 1, :)$, kdy obě dvojnásobné chyby jsou propojeny

na pozici $y_3(3, 1, 3)$. Hlavní problém sudého počtu chyb (v tomto případě 2) v jednom směru spočívá ve správnosti kontrolní parity v tomto směru, ačkoliv nastalo poškození bitů.

5.5.2 Nedetekování chyb

Pro zjištění nedetekovatelných chyb se vychází z výpočtů provedených na maticích uvedených výše. První z nedetekovatelných chyb nastává v případě 3. a více propojených dvojnásobných chyb. V tomto případě algoritmy bez váhových koeficientů a kombinovaný algoritmus selžou téměř ve všech případech. Tyto algoritmy selhávají i v případě chyby přes všechny dimenze, jestliže se daný bit nachází ve správné hodnotě, ačkoliv kontrolní matice nabývá maximální hodnoty. Naopak oba algoritmy jsou schopny detekovat návazné dvojnásobné chyby a úspěšně je dekodovat s vysokou úspěšností. Oproti tomu algoritmus s váhovými koeficienty je schopen v některých případech úspěšně dekodovat chyby přes všechny dimenze i návazné dvojnásobné chyby ve větším počtu. Vše ovšem závisí na míře poškození informačních a paritních bitů a na konkrétních pozicích maximálních hodnot, které nabývá kontrolní matice. Tento algoritmus opravuje všechny prvky postupně po jednotlivých bitech a po každé inverzi bitu kontroluje přínos opravy a zdali se snížil počet chyb při kontrole parity. Jestliže první maxima kontrolní matice obsahují skutečně poškozené bity a tyto bity mají spojitost s dalšími chybami, může to vést k úspěšnému dekodování. Nicméně pokud první maximum kontrolní matice nabývá hodnoty 3 (viz. Obrázek 13) a daný bit je ve správné pozici, tak se nepovede dekodování provést úspěšně a dekodování skončí ve většině případech s chybou.

5.6 Měření a výpočty

Měření a výpočty probíhaly pro různé kombinace počtu dimenzí a délky bloků dat s cílem zaměřit se na bitovou chybovost všech algoritmů v rozsahu $10^{-4} - 10^{-6}$. Toho se dosáhlo různými hodnotami poměru energie jednoho bitu ku spektrální výkonové hustotě šumu (E_b/N_0). Tato hodnota je přepočítána na poměr výkonu signálu ku šumu (SNR), na jejíž základě je přidáno odpovídající množství Gaussova bílého šumu (AGWN).

5.6.1 Bitová chybovost algoritmů

Jak bylo zmíněno výše, každá kombinace různého počtu dimenzí a délky bloku disponuje vlastními hodnotami E_b/N_0 , aby se dosáhlo chybovosti v okolo hodnot 10^{-6} . Z toho důvodu byly zvoleny hodnoty uvedené v tabulce 2.

Tabulka 2: Hodnoty E_b/N_0 pro různé rozměry bloků dat.

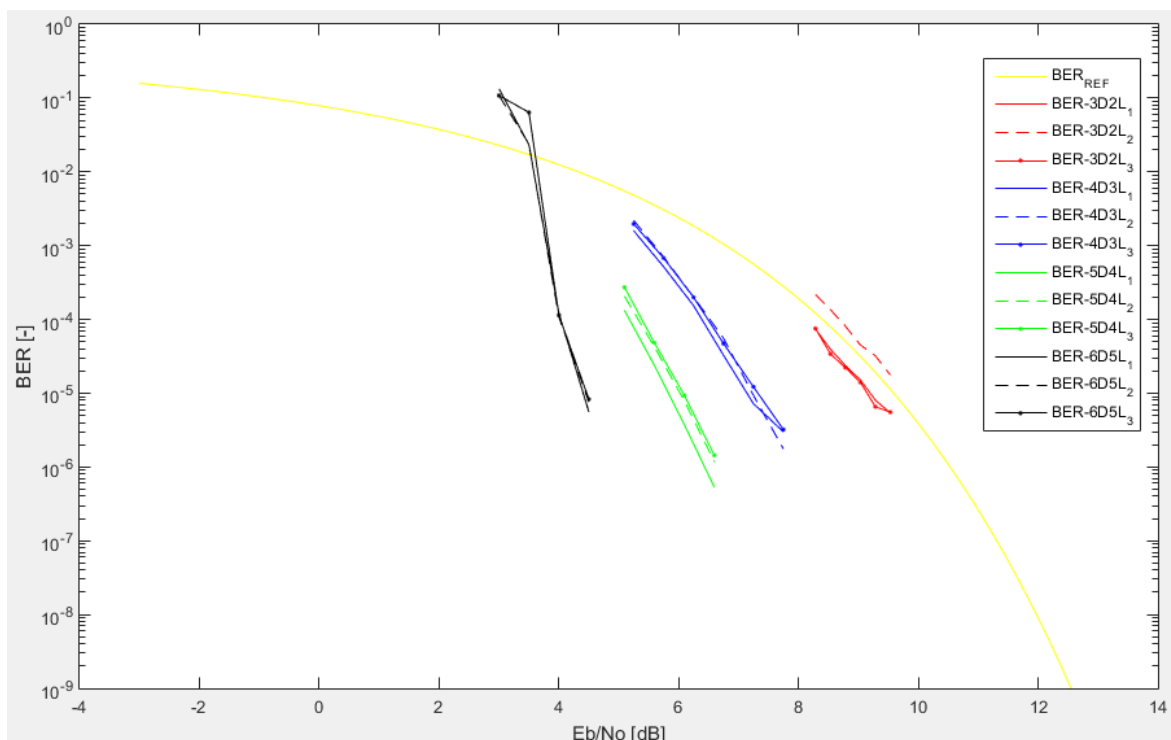
Eb/N ₀								
3D2L	3	3,25	3,5	3,75	4	4,25	---	---
4D3L	0,25	0,75	1,25	1,75	2,25	2,75	---	---
5D4L	0,25	0,75	1,25	1,75	---	---	---	---
6D5L	-1,75	-1,25	-0,75	-0,25	---	---	---	---
3D4L	2,95	3,25	3,5	3,75	4	4,25	4,55	4,85
4D5L	1,25	1,5	1,75	2	2,25	2,5	2,8	3,1

Na základě těchto hodnot se přidala odpovídající hodnota Gaussova bílého šumu, kdy pro vyšší hodnoty E_b/N_0 bylo přidáno nižší množství AGWN. To lze vidět na následujících grafech (obrázky 14 a 15), jelikož pro shodné hodnoty E_b/N_0 klesá chybovost s rostoucím počtem dimenzí, ovšem pro nižší hodnoty E_b/N_0 přepočtené na SNR je chybovost výrazně vyšší.

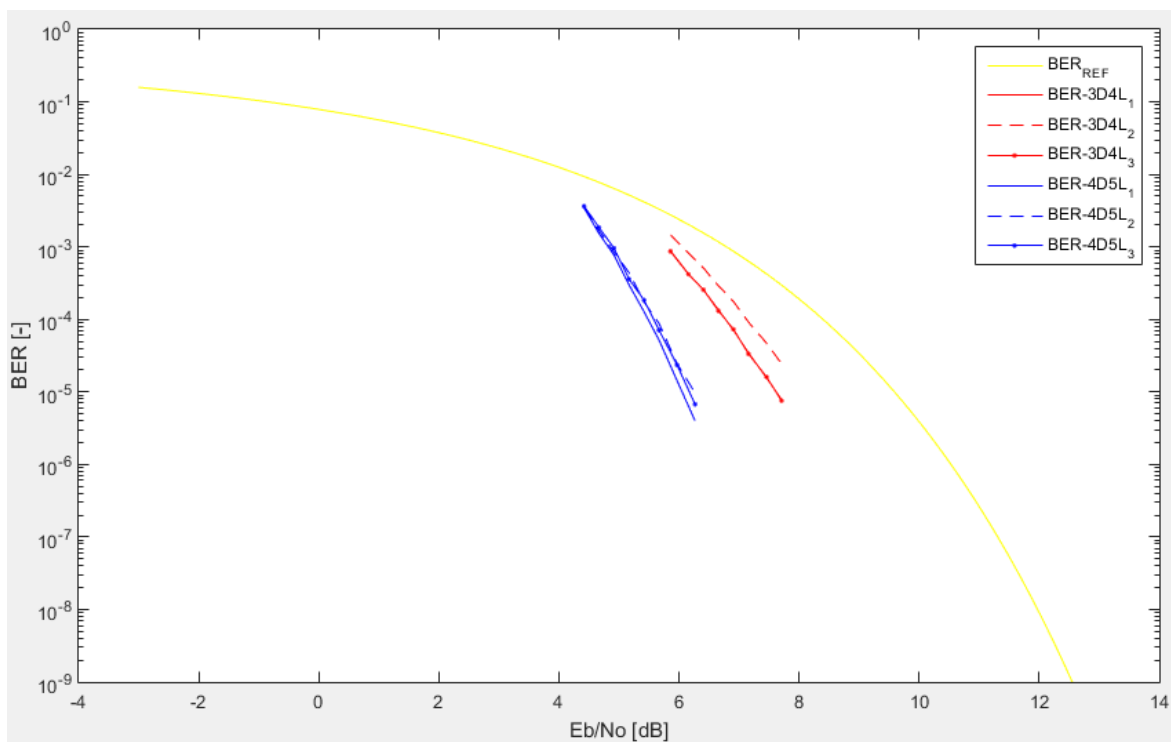
Změřené a vypočtené hodnoty jsou rozděleny do 2 grafů, kdy v každém grafu jsou zobrazeny kombinace bloků dat o podobném kódovém poměru. Hodnoty na ose x následujících grafů jsou vypočteny z kódového poměru a hodnot E_b/N_0 vztahem

$$x_i = EbNo[i] + 10 \cdot \log_{10} \frac{1}{R} \quad (5.1)$$

kde R představuje kódový poměr. Osa y je poté tvořena hodnotami chybovostí všech tří algoritmů pro různé hodnoty E_b/N_0 uvedených v tabulce 2. Jednotlivé kombinace dimenzí a délky bloků dat jsou zobrazeny vůči chybovosti demodulátoru (BER_{REF}) a jsou popsány v legendě, přičemž poslední hodnota každé kombinace určuje metodu dekódování. Tyto metody jsou popsány v kapitole 5.4, přičemž v případě 1 se jedná o metodu bez váhových koeficientů, 2 značí metodu s váhovými koeficienty a 3 kombinovanou metodu.



Obrázek 16: Bitová chybovost algoritmů pro různé hodnoty E_b/N_0 a kódový poměr blízký hodnotě $1/3$.



Obrázek 17: Bitová chybovost algoritmů pro různé hodnoty E_b/N_0 a kódový poměr blízký hodnotě $1/2$.

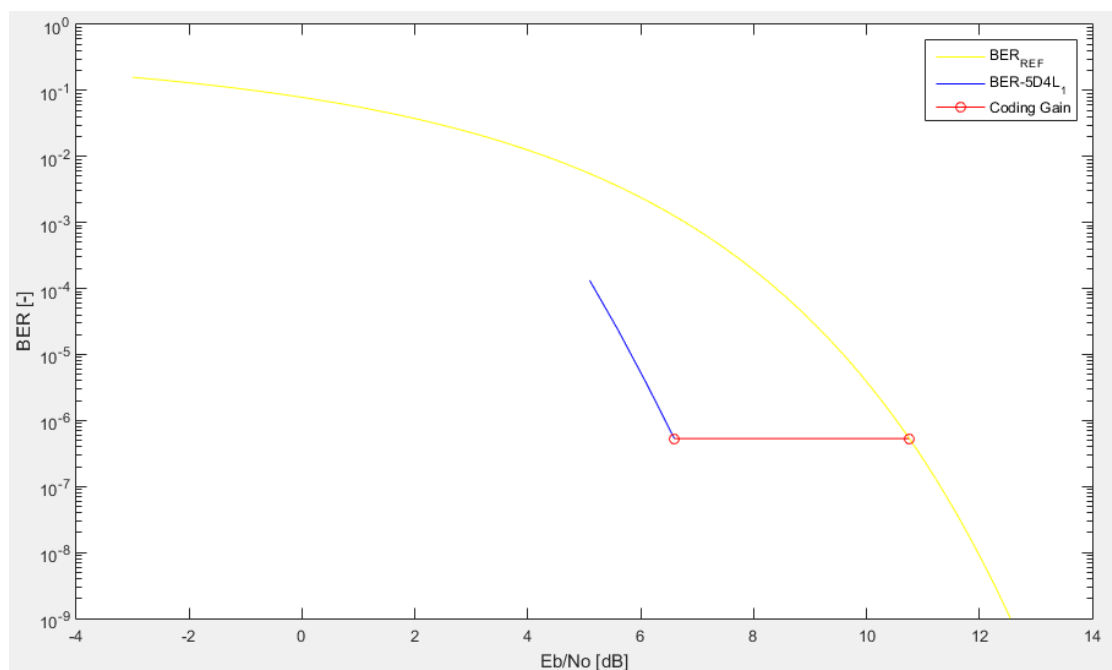
V tabulce 3 jsou číselně znázorněny hodnoty nejnižší chybovosti pro různé kombinace počtu dimenzí a délky bloku dat pro zvolené hodnoty E_b/N_0 .

Tabulka 3: Nejnižší změřená chybovost pro zvolené hodnoty E_b/N_0 .

	3D2L	4D3L	5D4L	6D5L	3D4L	4D5L
E_b/N_0	4,25	2,75	1,75	-0,25	4,85	3,1
$BER1_{MIN}$	$5,36 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	$5,3 \cdot 10^{-7}$	$5,57 \cdot 10^{-6}$	$7,84 \cdot 10^{-6}$	$3,99 \cdot 10^{-6}$
$BER2_{MIN}$	$1,76 \cdot 10^{-5}$	$1,76 \cdot 10^{-6}$	$1,18 \cdot 10^{-6}$	$7,59 \cdot 10^{-6}$	$2,45 \cdot 10^{-5}$	$9,69 \cdot 10^{-6}$
$BER3_{MIN}$	$5,53 \cdot 10^{-6}$	$3,21 \cdot 10^{-6}$	$1,44 \cdot 10^{-6}$	$8,29 \cdot 10^{-6}$	$7,57 \cdot 10^{-6}$	$6,74 \cdot 10^{-6}$

5.6.2 Kódový zisk

Kódový zisk (Coding gain) vyjadřuje míru zlepšení chybovosti při použití kódování a dekódování oproti chybovosti demodulátoru BER_{REF} . Princip zjištění kódového zisku je znázorněn na obrázku 16. Jedná se o spojení chybovosti dekódovacího algoritmu a chybovosti demodulátoru v konstantní hodnotě BER, která v případě prvního algoritmu u bloku 5D4L nabývá hodnoty $5,3 \cdot 10^{-7}$.



Obrázek 18: Kódový zisk.

Funkce značící chybovost demodulátoru je definována v MATLABu jako

$$x = db1 = 20 * \log_{10}(\operatorname{erfinv}(1 - 2 * ber1));$$

$$y = ber1 = \operatorname{logspace}(-9, -0.8, 1001);$$

Jelikož jsou známy obě souřadnice bodu pro nejnižší chybovost algoritmu pro danou metodu dekódování, výpočet hodnoty x , která leží na průsečíku křivek pro chybovost demodulátoru a kódový zisk, se provede dosazením hodnoty $5,3 \cdot 10^{-7}$ za $ber1$. Rozdíl těchto dvou souřadnic rovnoběžných s osou x značí kódový zisk. Výsledné hodnoty kódového zisku pro zveřejněné metody dekódování a rozměry bloků jsou uvedeny v tabulce 4.

Tabulka 4: Kódový zisk pro nejnižší chybovost jednotlivých dekódovacích algoritmů a rozměry bloků dat.

	3D2L	4D3L	5D4L	6D5L	3D4L	4D5L
1. algoritmus	0,3298	2,356	4,159	5,345	1,99	3,720
2. algoritmus	-0,2098	2,569	3,869	5,21	1,453	3,335
3. algoritmus	0,3158	2,329	3,8	5,171	2,005	3,497

5.6.3 Chybovost rámců

Další ze změřených chybovostí je počet přenesených a dekódovaných dat přepočtených na rámcová chybovost (FER), která udává množství chybně přenesených bloků dat na výstupu dekodéru vůči celkovému počtu bloků přenesených dat. Vypočtené hodnoty pro měřené bloky dat jsou uvedeny pro lepší přehlednost v tabulkách 6, 7 a 8, přičemž hodnoty E_b/N_0 , pro které bylo přidáno odpovídající množství Gaussova bílého šumu, jsou uvedeny na začátku kapitoly 5.6.1 v tabulce 2.

Tabulka 5: Rámcová chybovost dekódování bez váhových koeficientů.

	3D2L	4D3L	5D4L	6D5L	3D4L	4D5L
E_b/N_0 [1]	$4,46 \cdot 10^{-4}$	$3,33 \cdot 10^{-2}$	$4,37 \cdot 10^{-2}$	1	$2,08 \cdot 10^{-2}$	0,25
E_b/N_0 [2]	$2,39 \cdot 10^{-4}$	$1,2 \cdot 10^{-2}$	$7,96 \cdot 10^{-3}$	1	$1,09 \cdot 10^{-2}$	0,13
E_b/N_0 [3]	$1,43 \cdot 10^{-4}$	$3,81 \cdot 10^{-3}$	$1,4 \cdot 10^{-3}$	0,24	$6,57 \cdot 10^{-3}$	$7,61 \cdot 10^{-2}$
E_b/N_0 [4]	$9,73 \cdot 10^{-5}$	$9,73 \cdot 10^{-4}$	$1,94 \cdot 10^{-4}$	$1,53 \cdot 10^{-2}$	$3,67 \cdot 10^{-3}$	$3,25 \cdot 10^{-2}$
E_b/N_0 [5]	$5,21 \cdot 10^{-5}$	$2,13 \cdot 10^{-4}$	---	---	$1,92 \cdot 10^{-3}$	$1,54 \cdot 10^{-2}$
E_b/N_0 [6]	$3,51 \cdot 10^{-5}$	$8,92 \cdot 10^{-5}$	---	---	$9,63 \cdot 10^{-4}$	$6,92 \cdot 10^{-3}$
E_b/N_0 [7]	---	---	---	---	$4,48 \cdot 10^{-4}$	$2,21 \cdot 10^{-3}$
E_b/N_0 [8]	---	---	---	---	$2,09 \cdot 10^{-4}$	$5,51 \cdot 10^{-4}$

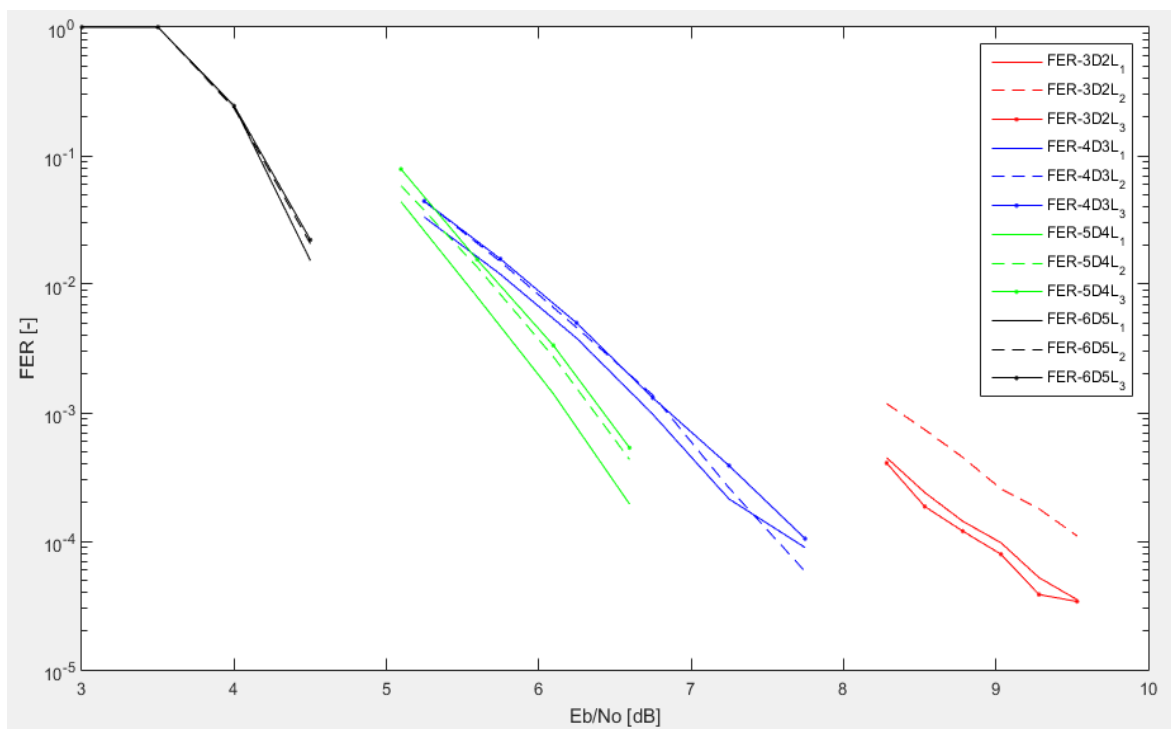
Tabulka 6: Rámcová chybovost dekódování s váhovými koeficienty.

	3D2L	4D3L	5D4L	6D5L	3D4L	4D5L
Eb/N ₀ [1]	$1,18 \cdot 10^{-3}$	$4,43 \cdot 10^{-2}$	$5,85 \cdot 10^{-2}$	1	$2,91 \cdot 10^{-2}$	0,28
Eb/N ₀ [2]	$7,38 \cdot 10^{-4}$	$1,49 \cdot 10^{-2}$	$1,36 \cdot 10^{-2}$	1	$1,74 \cdot 10^{-2}$	0,15
Eb/N ₀ [3]	$4,49 \cdot 10^{-4}$	$4,6 \cdot 10^{-3}$	$2,72 \cdot 10^{-3}$	0,24	$1,09 \cdot 10^{-2}$	$8,82 \cdot 10^{-2}$
Eb/N ₀ [4]	$2,56 \cdot 10^{-4}$	$1,37 \cdot 10^{-3}$	$4,32 \cdot 10^{-4}$	$1,53 \cdot 10^{-2}$	$6,38 \cdot 10^{-3}$	$4,64 \cdot 10^{-2}$
Eb/N ₀ [5]	$1,79 \cdot 10^{-4}$	$2,59 \cdot 10^{-4}$	---	---	$3,99 \cdot 10^{-3}$	$2,05 \cdot 10^{-2}$
Eb/N ₀ [6]	$1,1 \cdot 10^{-4}$	$5,82 \cdot 10^{-5}$	---	---	$2,2 \cdot 10^{-3}$	$9,61 \cdot 10^{-3}$
Eb/N ₀ [7]	---	---	---	---	$1,17 \cdot 10^{-3}$	$3,22 \cdot 10^{-3}$
Eb/N ₀ [8]	---	---	---	---	$6,37 \cdot 10^{-4}$	$1,36 \cdot 10^{-3}$

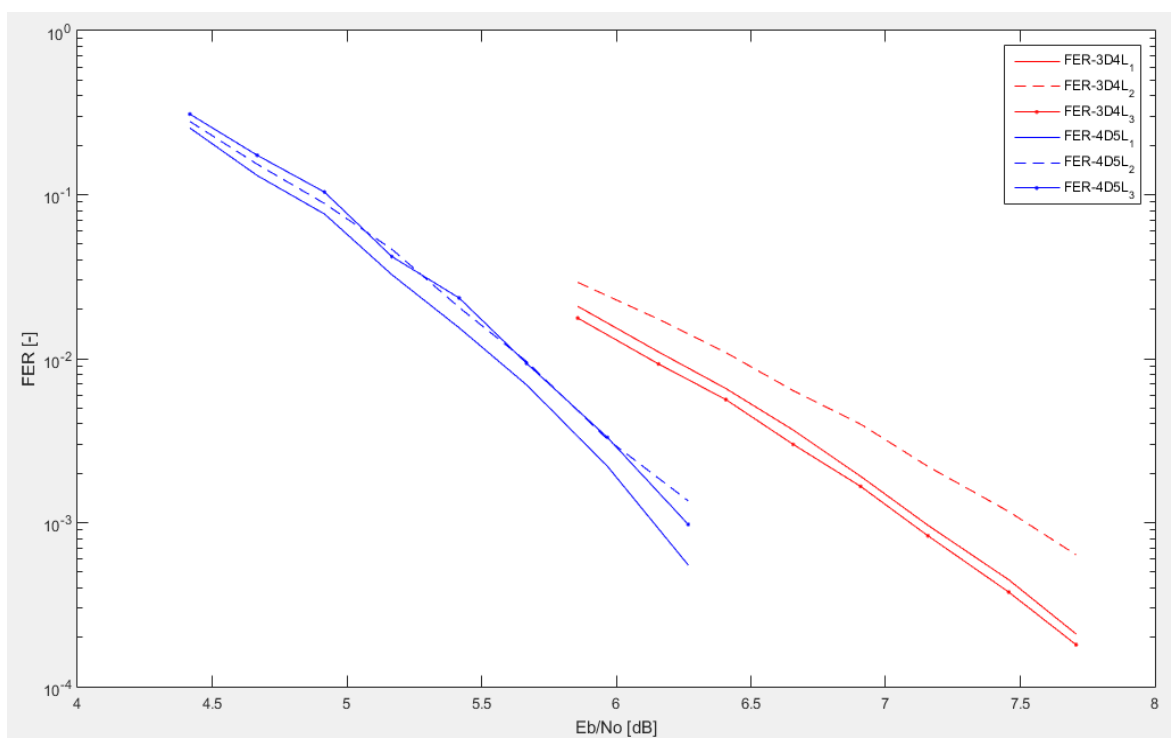
Tabulka 7: Rámcová chybovost kombinované metody dekódování.

	3D2L	4D3L	5D4L	6D5L	3D4L	4D5L
Eb/N ₀ [1]	$4,06 \cdot 10^{-4}$	$4,43 \cdot 10^{-2}$	$7,86 \cdot 10^{-2}$	1	$1,77 \cdot 10^{-2}$	0,31
Eb/N ₀ [2]	$1,86 \cdot 10^{-4}$	$1,57 \cdot 10^{-2}$	$1,55 \cdot 10^{-2}$	1	$9,27 \cdot 10^{-3}$	0,17
Eb/N ₀ [3]	$1,19 \cdot 10^{-4}$	$4,99 \cdot 10^{-3}$	$3,34 \cdot 10^{-3}$	0,24	$5,62 \cdot 10^{-3}$	0,1
Eb/N ₀ [4]	$7,92 \cdot 10^{-5}$	$1,31 \cdot 10^{-3}$	$5,33 \cdot 10^{-4}$	$1,53 \cdot 10^{-2}$	$2,99 \cdot 10^{-3}$	$4,19 \cdot 10^{-2}$
Eb/N ₀ [5]	$3,85 \cdot 10^{-5}$	$3,88 \cdot 10^{-4}$	---	---	$1,66 \cdot 10^{-3}$	$2,34 \cdot 10^{-2}$
Eb/N ₀ [6]	$3,39 \cdot 10^{-5}$	$1,05 \cdot 10^{-4}$	---	---	$8,31 \cdot 10^{-4}$	$9,36 \cdot 10^{-3}$
Eb/N ₀ [7]	---	---	---	---	$3,77 \cdot 10^{-4}$	$3,31 \cdot 10^{-3}$
Eb/N ₀ [8]	---	---	---	---	$1,8 \cdot 10^{-4}$	$9,76 \cdot 10^{-4}$

Na následujících obrázcích 19 a 20 je zobrazena rámcová chybovost jednotlivých dekódovacích algoritmů pro zvolené hodnoty Eb/N₀ z tabulky 2, přičemž v případě 1 se jedná o metodu bez váhových koeficientů, 2 značí metodu s váhovými koeficienty a 3 kombinovanou metodu. Tyto hodnoty jsou rozděleny do grafů v závislosti na podobnosti kódového poměru. Na obrázku 17 lze vidět, že blok dat 6D5L nabývá v části grafu maximální hodnoty 10⁰, což značí, že žádný rámeček bloku dat se nepovedlo úspěšně dekódovat.



Obrázek 19: Rámcová chybovost pro různé hodnoty E_b/N_0 a kódový poměr blízký hodnotě 1/3.



Obrázek 20: Rámcová chybovost pro různé hodnoty E_b/N_0 a kódový poměr blízký hodnotě 1/2.

5.6.4 Nerozpoznaná chybovost rámců

Poslední z měřených chybovostí je nerozpoznaná chybovost rámců (UFER). Ta udává množství rámců, u kterých bylo dekódování ukončeno s tím, že paritní bity se nachází ve správné pozici, ovšem blok dat není shodný se zaslaným zakódovaným blokem dat. Vypočtené hodnoty pro měřené bloky dat jsou opět uvedeny pro lepší přehlednost ve třech tabulkách níže pro hodnoty E_b/N_0 uvedené v tabulce 2.

Tabulka 8: Nerozpoznaná chybovost rámců pro dekódování bez váhových koeficientů.

	3D2L	4D3L	5D4L	6D5L	3D4L	4D5L
E_b/N_0 [1]	$5,41 \cdot 10^{-5}$	$1,18 \cdot 10^{-2}$	0	0	$3,39 \cdot 10^{-3}$	0,14
E_b/N_0 [2]	$1,99 \cdot 10^{-5}$	$3,67 \cdot 10^{-3}$	0	0	$1,52 \cdot 10^{-3}$	$6,17 \cdot 10^{-2}$
E_b/N_0 [3]	$1,66 \cdot 10^{-5}$	$8,89 \cdot 10^{-4}$	0	$4,39 \cdot 10^{-3}$	$9,45 \cdot 10^{-4}$	$3,04 \cdot 10^{-2}$
E_b/N_0 [4]	$4,53 \cdot 10^{-6}$	$1,09 \cdot 10^{-4}$	0	0	$3,84 \cdot 10^{-4}$	$1,14 \cdot 10^{-2}$
E_b/N_0 [5]	$2,26 \cdot 10^{-6}$	$2,72 \cdot 10^{-5}$	---	---	$2 \cdot 10^{-4}$	$4,35 \cdot 10^{-3}$
E_b/N_0 [6]	$1,13 \cdot 10^{-6}$	$1,16 \cdot 10^{-5}$	---	---	$9,92 \cdot 10^{-5}$	$1,47 \cdot 10^{-3}$
E_b/N_0 [7]	---	---	---	---	$3,35 \cdot 10^{-5}$	$2,54 \cdot 10^{-4}$
E_b/N_0 [8]	---	---	---	---	$1,68 \cdot 10^{-5}$	$1,27 \cdot 10^{-4}$

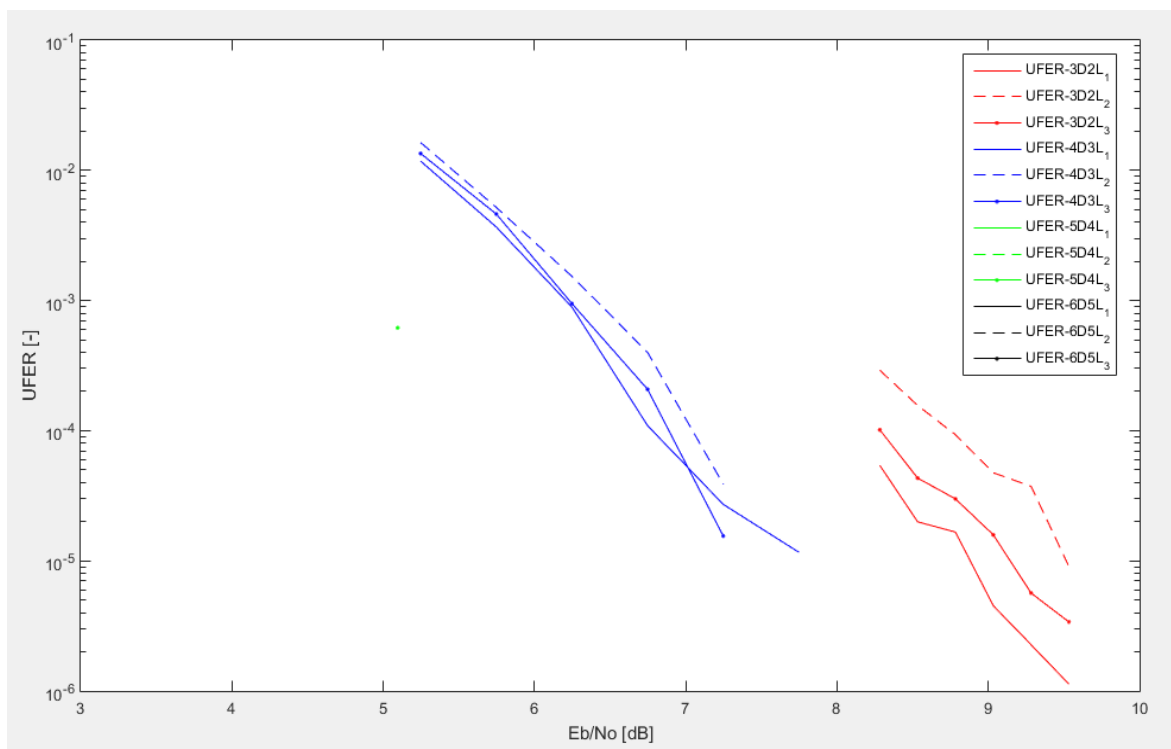
Tabulka 9: Nerozpoznaná chybovost rámců pro dekódování s váhovými koeficienty.

	3D2L	4D3L	5D4L	6D5L	3D4L	4D5L
E_b/N_0 [1]	$2,91 \cdot 10^{-4}$	$1,63 \cdot 10^{-2}$	$9,24 \cdot 10^{-4}$	0	$1,25 \cdot 10^{-2}$	$6,78 \cdot 10^{-2}$
E_b/N_0 [2]	$1,56 \cdot 10^{-4}$	$5,19 \cdot 10^{-3}$	0	0	$6,46 \cdot 10^{-3}$	$4,8 \cdot 10^{-2}$
E_b/N_0 [3]	$9,31 \cdot 10^{-5}$	$1,53 \cdot 10^{-3}$	0	0	$4 \cdot 10^{-3}$	$2,31 \cdot 10^{-2}$
E_b/N_0 [4]	$4,75 \cdot 10^{-5}$	$3,99 \cdot 10^{-4}$	0	0	$2,12 \cdot 10^{-3}$	$1,46 \cdot 10^{-2}$
E_b/N_0 [5]	$3,74 \cdot 10^{-5}$	$3,88 \cdot 10^{-5}$	---	---	$1,2 \cdot 10^{-3}$	$5,72 \cdot 10^{-3}$
E_b/N_0 [6]	$9,06 \cdot 10^{-6}$	0	---	---	$5,99 \cdot 10^{-4}$	$3,55 \cdot 10^{-3}$
E_b/N_0 [7]	---	---	---	---	$2,35 \cdot 10^{-4}$	$7,64 \cdot 10^{-4}$
E_b/N_0 [8]	---	---	---	---	$1,13 \cdot 10^{-4}$	$1,7 \cdot 10^{-4}$

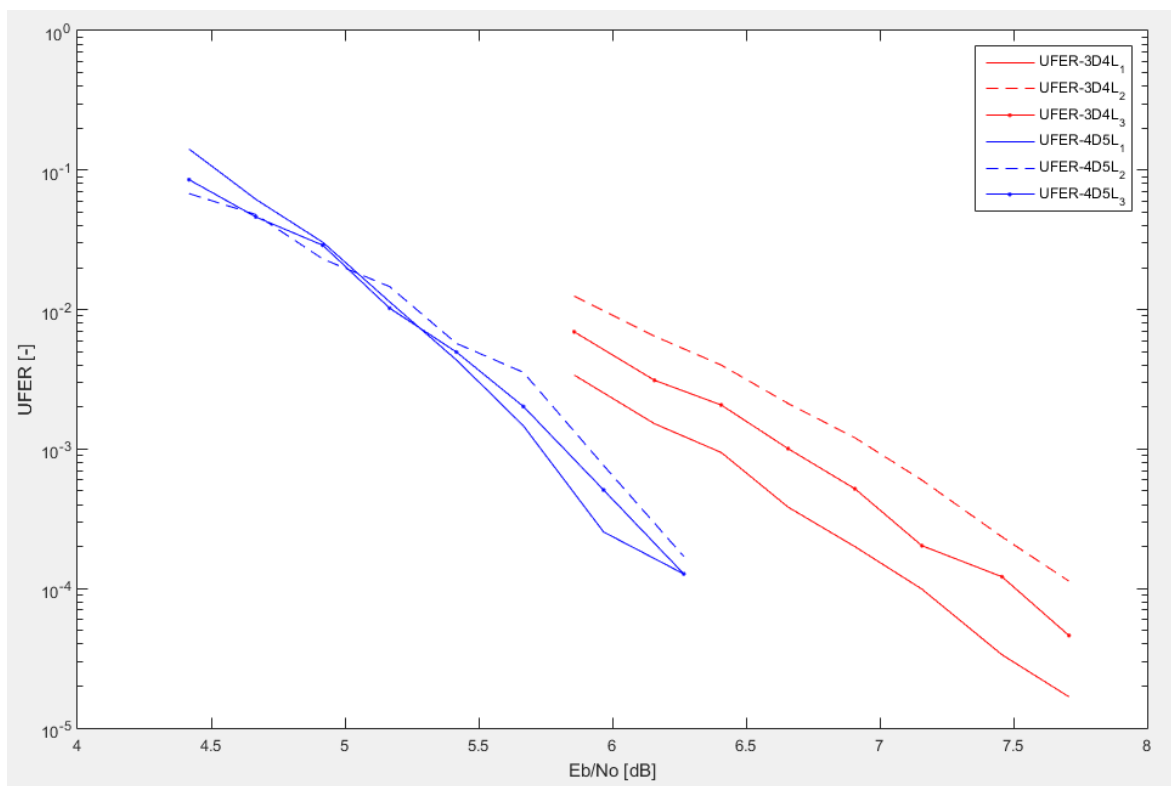
Tabulka 10: Nerozpoznaná chybovost rámců pro kombinované metody dekódování.

	3D2L	4D3L	5D4L	6D5L	3D4L	4D5L
Eb/N ₀ [1]	$1,01 \cdot 10^{-4}$	$1,35 \cdot 10^{-2}$	$6,16 \cdot 10^{-4}$	0	$6,93 \cdot 10^{-3}$	$8,54 \cdot 10^{-2}$
Eb/N ₀ [2]	$4,32 \cdot 10^{-5}$	$4,61 \cdot 10^{-3}$	0	0	$3,12 \cdot 10^{-3}$	$4,61 \cdot 10^{-2}$
Eb/N ₀ [3]	$2,99 \cdot 10^{-5}$	$9,47 \cdot 10^{-4}$	0	0	$2,07 \cdot 10^{-3}$	$2,89 \cdot 10^{-2}$
Eb/N ₀ [4]	$1,58 \cdot 10^{-5}$	$2,08 \cdot 10^{-4}$	0	0	$2,99 \cdot 10^{-3}$	$1,02 \cdot 10^{-2}$
Eb/N ₀ [5]	$5,66 \cdot 10^{-6}$	$1,55 \cdot 10^{-5}$	---	---	$1 \cdot 10^{-3}$	$4,96 \cdot 10^{-3}$
Eb/N ₀ [6]	$3,39 \cdot 10^{-6}$	0	---	---	$5,19 \cdot 10^{-4}$	$2,02 \cdot 10^{-3}$
Eb/N ₀ [7]	---	---	---	---	$2,03 \cdot 10^{-4}$	$5,09 \cdot 10^{-4}$
Eb/N ₀ [8]	---	---	---	---	$4,61 \cdot 10^{-5}$	$1,27 \cdot 10^{-4}$

Na následujících obrázcích 19 a 20 je zobrazena nerozpoznaná chybovost rámců jednotlivých dekódovacích algoritmů pro zvolené hodnoty Eb/N₀ z tabulky 2, přičemž v případě 1 se opět jedná o metodu bez váhových koeficientů, 2 značí metodu s váhovými koeficienty a 3 kombinovanou metodu. Na obrázku 19 zobrazující nerozpoznanou chybovost rámců pro kódový poměr v blízkosti 1/3 jsou zobrazeny křivky pouze pro kombinace 3D2L a 4D3L a chybí na nich křivky pro data o velikosti 5D4L a 6D5L. To proto, že pro tuto velikost bloků téměř nedochází k nerozpoznané chybovosti rámců. Pro blok 5D4L nastala chyba maximálně v jednom bodě, čímž nelze vykreslit křivku a pro blok 6D5L se nepovedlo žádný blok dat pro hodnoty $Eb/N_0 = -1,75$ a $Eb/N_0 = -1,25$ úspěšně dekódovat z důvodu vysoké míry přidaného šumu, a proto v těchto hodnotách nemůže tato chyba ani nastat.



Obrázek 21: Nerozpoznaná chybovost rámců pro různé hodnoty E_b/N_0 a kódový poměr blízky hodnotě 1/3.



Obrázek 22: Nerozpoznaná chybovost rámců pro různé hodnoty E_b/N_0 a kódový poměr blízky hodnotě 1/2.

5.7 Shrnutí výsledků

5.7.1 Výsledky bitové chybovosti

Bitová chybovost vyjadřuje chybovost dekódovacích algoritmů a klesá se vzrůstajícími hodnotami E_b/N_0 . Pro shodné hodnoty E_b/N_0 bitová chybovost klesá s narůstající délkou zabezpečeného bloku dat a s klesajícím kódovým poměrem. Nejnižší změřené bitové chybovosti pro zvolené hodnoty E_b/N_0 dosahuje blok dat 5D4L u algoritmu bez váhových koeficientů, jehož chybovost je $BER = 5,3 \cdot 10^{-7}$. Tento blok dat má kódový poměr $R = 0,328$ a délku kódového slova $n = 3125$. Při srovnání jednotlivých algoritmů nastává nejnižší bitová chybovost u většiny testovaných kombinací právě pro algoritmus bez váhových koeficientů.

Kódový zisk vyjadřuje zlepšení chybovosti dekódovacích algoritmů oproti chybovosti demodulátoru a při hranici bitové chybovosti 10^{-5} narůstá s klesajícím kódovým poměrem a narůstající délkou zabezpečeného bloku dat. Nejvyšší kódový zisk vykazoval kód 6D5L, který disponuje kódovým poměrem $R = 0,335$ s velikostí zabezpečeného bloku $n = 46656$.

5.7.2 Výsledky rámcové chybovosti

Rámcová chybovost vyjadřuje počet bloků dat na výstupu dekodéru k celkovému počtu přenesených bloků a klesá s klesající hodnotou kódového poměru a s narůstající délkou zabezpečeného bloku dat. Nicméně nejnižší rámcová chybovost nastává pro blok dat o velikosti 3D2L díky vysokým hodnotám E_b/N_0 , díky kterým bylo přidáno nižší množství odpovídajícího šumu oproti ostatním kombinacím. Kód 3D2L má kódový poměr $R = 0,296$ a velikost zabezpečeného bloku $n = 27$.

5.7.3 Výsledky nerozpoznané chybovosti rámců

Nerozpoznaná chybovost rámců udává množství rámců, u kolika bylo dekódování ukončení s tím, že kontrola parity sedí, ale dekódovaný blok dat neodpovídá odeslanému bloku dat. Chybovost rámců klesá narůstající délkou zabezpečeného bloku dat a s klesajícím kódovým poměrem, přičemž u robustnějších bloků, které jsou zabezpečeny dostatečným počtem paritních bitů, k této chybovosti téměř nedochází. Nejnižší rámcová chybovost nastává pro bloky dat 5D4L s kódovým poměrem $R = 0,328$ a 6D5L s kódovým poměrem $R = 0,335$. Nejlepších výsledků dosahuje algoritmus bez váhových koeficientů, u kterého dosahuje nerozpoznaná rámcová chybovost nejnižších hodnot.

6 Závěr

Tato bakalářská práce je zaměřena na multidimenzionální paritní kódy, které zabezpečují informační bity do vícerozměrného bloku dat a slibují perspektivní metodu kódování a dekódování. V této bakalářské práci jsem zrealizoval kodéry a dekodéry pro multidimenzionální paritní kódy a otestoval na nich bitovou chybovost, rámcovou chybovost a nerozpoznanou chybovost rámců až do 6. dimenze s různými délkami bloku dat.

Cílem této bakalářské práce bylo studium paritních kódů, vytvoření kodéru a dekodéru paritních kódů pro více dimenzí a analyzovat tyto kódy a výsledky. Bitová chybovost, rámcová chybovost a nerozpoznaná chybovost rámců multidimenzionálních paritních kódů klesá s klesajícím kódovým poměrem a se vzrůstající velikostí zabezpečeného bloku dat. Kódový zisk při hranici bitové chybovosti 10^{-5} narůstá s klesajícím kódovým poměrem a narůstající délkou zabezpečeného bloku dat.

Při testování provedených měření jsem provedl analýzu charakteristických chyb u kódu 3D2L, které není dekodér schopen detekovat a opravit, a tyto charakteristické chyby jsem v této práci popsal. Poté jsem provedl analýzu dekódovacích algoritmů uvedených v této práci a nejlepších výsledků dosahuje algoritmus bez váhových koeficientů, který ve většině případů dává nejvyšší kódový zisk, nejnižší bitovou chybovost, nejnižší chybovost rámců i nejnižší nerozpoznanou chybovost rámců. Zároveň je tento dekódovací algoritmus nejrychlejší z uvedených algoritmů.

Z provedených měření představuje kód 5D4L přijatelný kompromis mezi rychlostí dekódování, přijatelnou délkou bloku a kódovým ziskem pro jednodušší hardware a tím i jeho vhodnost pro CubeSaty. Tento kód zároveň disponuje nízkou chybovostí a neobsahuje téměř žádnou nerozpoznanou chybovost rámců.

Výsledky této bakalářské práce lze v budoucnu ještě rozšířit a nalézt způsoby, jak analyzovat fungování dekódovacích algoritmů i při vyšších počtech dimenzí a delších blocích dat, aby je bylo možné ladit a optimalizovat. Výkonost uvedených dekódovacích algoritmů by mohla být výrazně zlepšena, pokud by demodulátor obdržel na vstupu pravděpodobnost z demodulátoru s měkkým rozhodováním (tzv. soft decision demodulation) namísto tvrdého rozhodování.

Seznam použitých obrázků

OBRÁZEK 1: OBECNÉ SCHÉMA SDĚLOVACÍHO SYSTÉMU.....	11
OBRÁZEK 2: SCHÉMA BCH KODÉRU	23
OBRÁZEK 3: OBECNÉ SCHÉMA DEKÓDOVÁNÍ RS KÓDŮ	26
OBRÁZEK 4: VÝPOČET PARITNÍCH BITŮ PRO KÓDOVÁNÍ VE 2D	27
OBRÁZEK 5: PRINCIP ZABEZPEČENÍ 3D BLOKU DAT.....	27
OBRÁZEK 6: KRYCHLE PRO URČENÍ KŘÍŽOVÉ PARITY V 5D	28
OBRÁZEK 7: PARITNÍ BITY KONTROLUJÍCÍ INFORMAČNÍ BIT V 5D.....	28
OBRÁZEK 8: INFORMAČNÍ BITY KONTROLOVANÉ PARITNÍM BITEM V 5D.....	29
OBRÁZEK 9: FUNKCE PRO VYTVOŘENÍ GENERUJÍCÍ MATICE.....	31
OBRÁZEK 10: VÝVOJOVÝ DIAGRAM POUŽITÍ ITERATIVNÍ METODY.....	33
OBRÁZEK 11: BLOK DAT PŘED NASTAVENÍM (Y1) A PO NASTAVENÍ (Y2) PARITNÍHO BITU. ..	35
OBRÁZEK 12: FUNKCE PRO PŘENOS DAT S VYUŽITÍM BPSK MODULACE A DEMODULACE. ...	35
OBRÁZEK 13: BLOK DAT PO ZAKÓDOVÁNÍ A PO PŘIDÁNÍ ŠUMU.	38
OBRÁZEK 14: DEKÓDOVANÉ BLOKY DAT PRO JEDNOTLIVÉ ALGORITMY PO PRVNÍ ITERACI. .	39
OBRÁZEK 15: HODNOTY KONTROLNÍCH MATIC PRO JEDNOTLIVÉ DEKÓDOVACÍ ALGORITMY PO PRVNÍ ITERACI.....	39
OBRÁZEK 16: BITOVÁ CHYBOVOST ALGORITMŮ PRO RŮZNÉ HODNOTY E_b/N_0 A KÓDOVÝ POMĚR BLÍZKÝ HODNOTĚ $1/3$	43
OBRÁZEK 17: BITOVÁ CHYBOVOST ALGORITMŮ PRO RŮZNÉ HODNOTY E_b/N_0 A KÓDOVÝ POMĚR BLÍZKÝ HODNOTĚ $1/2$	43
OBRÁZEK 18: KÓDOVÝ ZISK.....	44
OBRÁZEK 19: RÁMCOVÁ CHYBOVOST PRO RŮZNÉ HODNOTY E_b/N_0 A KÓDOVÝ POMĚR BLÍZKÝ HODNOTĚ $1/3$	47
OBRÁZEK 20: RÁMCOVÁ CHYBOVOST PRO RŮZNÉ HODNOTY E_b/N_0 A KÓDOVÝ POMĚR BLÍZKÝ HODNOTĚ $1/2$	47
OBRÁZEK 21: NEROZPOZNANÁ CHYBOVOST RÁMČŮ PRO RŮZNÉ HODNOTY E_b/N_0 A KÓDOVÝ POMĚR BLÍZKÝ HODNOTĚ $1/3$	50
OBRÁZEK 22: NEROZPOZNANÁ CHYBOVOST RÁMČŮ PRO RŮZNÉ HODNOTY E_b/N_0 A KÓDOVÝ POMĚR BLÍZKÝ HODNOTĚ $1/2$	50

Seznam použitých tabulek

TABULKA 1: KÓDOVÝ POMĚR V ZÁVISLOSTI NA POČTU DIMENZÍ A POČTU INFORMAČNÍCH BITŮ	30
TABULKA 2: HODNOTY E_b/N_0 PRO RŮZNÉ ROZMĚRY BLOKŮ DAT.	42
TABULKA 3: NEJNIŽŠÍ ZMĚŘENÁ CHYBOVOST PRO ZVOLENÉ HODNOTY E_b/N_0	44
TABULKA 4: KÓDOVÝ ZISK PRO NEJNIŽŠÍ CHYBOVOST JEDNOTLIVÝCH DEKÓDOVACÍCH ALGORITMŮ A ROZMĚRY BLOKŮ DAT.	45
TABULKA 5: RÁMCOVÁ CHYBOVOST DEKÓDOVÁNÍ BEZ VÁHOVÝCH KOEFICIENTŮ.	45
TABULKA 6: RÁMCOVÁ CHYBOVOST DEKÓDOVÁNÍ S VÁHOVÝMI KOEFICIENTY.	46
TABULKA 7: RÁMCOVÁ CHYBOVOST KOMBINOVANÉ METODY DEKÓDOVÁNÍ.....	46
TABULKA 8: NEROZPOZNANÁ CHYBOVOST RÁMCŮ PRO DEKÓDOVÁNÍ BEZ VÁHOVÝCH KOEFICIENTŮ.....	48
TABULKA 9: NEROZPOZNANÁ CHYBOVOST RÁMCŮ PRO DEKÓDOVÁNÍ S VÁHOVÝMI KOEFICIENTY.....	48
TABULKA 10: NEROZPOZNANÁ CHYBOVOST RÁMCŮ PRO KOMBINOVANÉ METODY DEKÓDOVÁNÍ.	49

Seznam symbolů a zkratk

MATLAB	Matrix laboratory
A/D	Analogově digitální
PCM.....	Pulse-code modulation
AM.....	Amplitude modulation
PM	Phase modulation
FM	Frequency modulation
ASK	Amplitude-shift keying
FSK.....	Frequency-shift keying
PSK.....	Phase-shift keying
PAM	Pulse-amplitude modulation
QAM.....	Quadrature-amplitude modulation
PWM.....	Pulse-width Modulation
LEO	Low Earth Orbit
QPSK.....	Quadrature phase-shift keying
BPSK	Binary phase-shift keying
M-PSK.....	Multiple phase-shift keying
MSK	Minimum-shift keying
BFSK	Binary frequency-shift keying
BER	Bit-error rate
FER.....	Frame error rate
UFER	Unrecognized frame error rate
FEC.....	Forward error correction
EEC.....	Error correction code
FDM	Failed dimension marker
E_b/N_0	Energy per bit to noise power spectral density ratio
SNR	Signal to Noise Ratio
AWGN.....	Additive white Gaussian noise
BCH.....	Bose–Chaudhuri–Hocquenghem codes
RS	Reed–Solomon codes
MDPC.....	Multidimensional Cross Parity Check Codes
e, e1, b.....	Pomocné matice
p1, p2, p3	Paritní bity

x_3, x_5, x_6, x_7	Informační bity
s_1, s_2, s_3	Syndromy
$g(x)$	Generující polynom
$m(x)$	Minimální polynom
$z(x)$	Polynom nezabezpečené zprávy
$r(x)$	Polynom zbytku
$f(x)$	Polynom zabezpečené zprávy
$v(x)$	Polynom přijaté zprávy
\prod	Celkový součin
\sum	Celková suma
\oplus	Modulo
\otimes	Kroneckerův produkt
MPa.....	Megapascal
W	Watt
km	Kilometr

Seznam literatury a informačních zdrojů

- [1] CubeSat. Product Specification. [online]. Verze 13. [Cit. 20.2.2021]. Dostupné z: https://www.cubesat.org/s/cds_rev13_final2.pdf
- [2] SAEED, Nasir, ELZANATY, Ahmed, ALMORAD, Heba, DAHROUJ, Hayssam, AL-NAFFOURI, Y. Tareq a ALOUINI, Mohamed-Slim, *CubeSat Communications: Recent Advances and Future Challenges*. IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1839-1862, thirdquarter 2020, doi: 10.1109/COMST.2020.2990499.
- [3] MATOUŠEK, Radomil. *Metody kódování*. [Cit. 26.2.2021]. Dostupné z: <http://www.uai.fme.vutbr.cz/~matousek/TIK/TIKv19.pdf>
- [4] VLČEK, Karel. *Kompresa a kódová zabezpečení v multimediálních komunikacích*. 2. vydání. Praha: BEN – technická literatura, 2004, s. 85-93. ISBN 80-7300-134-9.
- [5] ČÍKA, Petr. Protichybové zabezpečení BCH kódem. *Elektrorevue* [online]. Brno, 2006. ISSN 1213-1539. Dostupné z: <http://www.elektrorevue.cz/clanky/06015/index.html>
- [6] NĚMEC, Karel. Protichybové kódové zabezpečení s Bose-Chaudhury-Hocquenhémovými kódy. *Elektrorevue* [online]. Brno, 2005. ISSN 1213-1539. Dostupné z: <http://www.elektrorevue.cz/clanky/05015/index.html>
- [7] ČÍKA, Petr, KOTON Jaroslav a KŘIVÁNEK Vítězslav. *Samoopravné Reed-Solomonovy kódy*. Brno, 2006. ISSN 1214-9675. Dostupné z: <http://access.fel.cvut.cz/view.php?cislocclanku=2006080002>
- [8] SLANINA, Martin a ŘÍČNÝ Václav. Reed-Solomonovy kódy v časové a ve frekvenční oblasti. *Elektrorevue* [online]. Brno, 2006. ISSN 1213-1539. Dostupné z: <http://www.elektrorevue.cz/clanky/06011/index.html>
- [9] DUDÁČEK, Luděk a VEŘTÁT, Ivo. *Multidimensional Cross Parity Check Codes as a Promising Solution to CubeSat Low Data Rate Downlinks*. 2019 International Conference on Applied Electronics (AE), Plzeň, ISBN: 978-80-261-0812-2
- [10] SHEA, M. John a WONG, F. Tan. *Multidimensional codes*. Encyclopedia of Telecommunications. Wiley 2003.
- [11] DUDÁČEK, Luděk a VEŘTÁT, Ivo. *Multidimensional parity check codes with short block length*. 2016 24th Telecommunications Forum (TELFOR 2016), ISBN: 978-1-5090-4086-5
- [12] GROVER, Pulkit, WOYACH, Kristen, SAHAI, Anant. *Towards a Communication-Theoretic Understanding of System-Level Power Consumption*. In IEEE Journal on Selected Areas in Communications, vol. 29, no. 8, pp. 1744-1755, September 2011, doi: 10.1109/JSAC.2011.110922
- [13] The MathWorks, Inc. 1994. *MathWorks* [online]. ©1994-2021. Dostupné z: <https://uk.mathworks.com/>

Přílohy

3D2L_bp_BerFerValue.mat – Výsledky pro rozměry multidimenzionální matice 3D2L

3D4L_bp_BerFerValue.mat – Výsledky pro rozměry multidimenzionální matice 3D4L

4D3L_bp_BerFerValue.mat – Výsledky pro rozměry multidimenzionální matice 4D3L

4D5L_bp_BerFerValue.mat – Výsledky pro rozměry multidimenzionální matice 4D5L

5D4L_bp_BerFerValue.mat – Výsledky pro rozměry multidimenzionální matice 5D4L

6D5L_bp_BerFerValue.mat – Výsledky pro rozměry multidimenzionální matice 6D5L

bp_CreateWriteLock.m – Funkce umožňující čtení a zápis při spolupráci více MATLAB instancí na stejných výpočtech

bp_CrossParityBlock.m – Funkce pro zabezpečení bloku dat paritními bity

bp_DecodHard.m – Funkce obsahující dekódovací algoritmus s váhovými koeficienty

bp_Decoding.m – Funkce obsahující dekódovací algoritmus bez váhových koeficientů

bp_DecodingMix.m – Funkce obsahující kombinovaný dekódovací algoritmus

bp_DeleteWriteLock.m – Funkce pro smazání souboru, který slouží k ověření, zdali může MATLAB číst či zapisovat ze souboru při spolupráci více instancí na stejných výpočtech

bp_drawOut.m – Funkce, která sloužila na průběžné vykreslování bitové chybovosti během výpočtů

bp_drawOut_final.m – Funkce, ve které jsou dopočítány chybové hodnoty BER, FER a UFER a která vykresluje zvolenou chybovost bloků dat jednotlivých dekódovacích algoritmů o blízkém kódovém poměru

bp_Gendata.m – Funkce pro generování multidimenzionální matice o zvolené velikosti

bp_main_5d4l.m – Hlavní funkce použitá pro multidimenzionální matice 5D4L

bp_Psk.m – Funkce generující dvoustavově modulovaný signál BPSK

bp_PskDem.m – Funkce pro demodulaci signálu po přidání AWGN

bp_TransmitDataPsk.m – Funkce pro simulaci AWGN