

UNIVERZA V MARIBORU  
FAKULTETA ZA ELEKTROTEHNIKO,  
RAČUNALNIŠTVO IN INFORMATIKO

Maj Fontana Korošec

# **GENERATOR IN RAZČLENJEVALNIK OMS LPWAN OKVIRJEV**

Projekt na študijskem programu Telekomunikacije UN

Maribor, september 2024

# **GENERATOR IN RAZČLENJEVALNIK OMS LPWAN OKVIRJEV**

**Projekt na študijskem programu Telekomunikacije UN**

Študent:	Maj Fontana Korošec
Študijski program:	univerzitetni študijski program Telekomunikacije
Mentor:	doc. dr. Iztok Kramberger, univ. dipl. inž. el.

# Generator in razčlenjevalnik OMS LPWAN okvirjev

**Ključne besede:** OMS LPWAN, RSC kodiranje, AWGN

## **Povzetek**

*Zaključni projekt opisuje zasnovo generatorja in razčlenjevalnika okvirjev niznega profila komunikacijskega standarda OMS LPWAN, vključno z vsemi potrebnimi tehnikami kodiranja. S simulacijo kanala aditivnega belega Gaussovega šuma je analizirana uspešnost korekcijskega kodiranja za vse implementirane komunikacijske profile.*

# Generator and parser of OMS LPWAN frames

**Keywords:** OMS LPWAN, RSC coding, AWGN

## **Abstract**

*The final project describes the design of a generator and parser for OMS LPWAN burst mode frames, including all the necessary coding techniques. The performance of forward error correction coding for all implemented communication profiles is analyzed using a simulation of an additive white Gaussian noise channel.*

## Kazalo vsebine

1	UVOD .....	9
2	TEHNOLOŠKI PREGLED .....	10
2.1	Aditivni beli Gaussov šum .....	10
2.2	Konvolucijsko kodiranje .....	11
2.3	Ciklična redundanca .....	16
2.4	Modulacija z minimalnim skokom .....	17
2.5	Korelacijsko iskanje sinhronizacijske besede .....	18
2.6	Standard OMS LPWAN .....	20
2.6.1	Komunikacijski profili .....	22
2.6.2	Format podokvirjev .....	22
2.6.3	Korekcijsko kodiranje .....	27
2.6.4	Modulacija .....	28
3	ZASNOVA IN IZVEDBA .....	30
3.1	Programsko okolje .....	31
3.2	Predstavitev podatkov .....	31
3.3	Kodiranje podatkov .....	33
3.4	Generator okvirjev .....	35
3.5	Razčlenjevalnik okvirjev .....	37
4	ANALIZA IN REZULTATI .....	41
4.1	Simulacija prenosnega kanala .....	42
4.2	Zaznavanje podokvirjev .....	44
4.2.1	Navzdolnja povezava .....	44
4.2.2	Navzgornja povezava .....	46

4.3	Dekodiranje glave .....	52
4.4	Dekodiranje vsebine .....	56
4.4.1	Navzdolnja povezava .....	56
4.4.2	Navzgornja povezava .....	62
5	SKLEP .....	67
6	VIRI IN LITERATURA.....	68

## Kazalo slik

Slika 1: Primer konvolucijskega kodirnika.....	12
Slika 2: Prikaz postopka Viterbi dekodiranja konvolucijske kode.....	14
Slika 3: Rekurzivni sistematski konvolucijski kodirnik stopnje 1/2 z omejitveno dolžino 3 .....	15
Slika 4: Shema kodirnika s ciklično redundanco .....	16
Slika 5: Časovni potek modulacije z minimalnim skokom .....	17
Slika 6: Modulacija z minimalnim skokom v konstelacijskem diagramu .....	18
Slika 7: Enačba za izračun časovno-diskretne korelacije .....	19
Slika 8: Korekcijski faktor za normalizacijo križne korelacije .....	20
Slika 9: Deli M-Bus standarda .....	21
Slika 10: Razdelitev standarda OMS .....	21
Slika 11: Časovni potek oddaje treh okvirjev v načinu večkratne oddaje navzgornje povezave .....	22
Slika 12: Format podokvirja navzdolnje povezave .....	23
Slika 13: Format podokvirja navzgornje povezave .....	23
Slika 14: Format kodirane glave.....	24
Slika 15: Format kodirane dolžine .....	24
Slika 16: Format kodirane vsebine v načinu enojne oddaje. Po vrstnem redu: kodiranje stopnje 7/8, stopnje 1/2 in stopnje 1/3.....	25
Slika 17: Format kodirane vsebine v načinu večkratne oddaje. Po vrstnem redu: podokvir 1, podokvir 2 in podokvir 3 .....	25
Slika 18: Tok podatkov in struktur pri generiranju podokvirjev niznega načina .....	26
Slika 19: Diagram RSC kodirnika v niznem profilu .....	27
Slika 20: Enačba premešalnika RSC kodirnika v niznem profilu .....	28
Slika 21: konstelacijski diagram diferenčno kodirane GMSK navzgornje povezave .....	29
Slika 22: Blok diagram generatorja okvirjev .....	30
Slika 23: Blokovni diagram razčlenjevalnika okvirjev iz zaporedja neznanih vzorcev ....	31
Slika 24: Definicija strukture s skupino polj - primer za podokvir navzgornje povezave	32

Slika 25: Razred za enostaven dostop do struktur in vrednosti - primer za kodirano vsebino .....	33
Slika 26: Optimizacija funkcije s knjižnico Numba - primer za dotProduct .....	34
Slika 27: Potek generacije okvirja navzgornje povezave .....	36
Slika 28: Potek generacije okvirja navzdolnje povezave.....	37
Slika 29: Potek razčlenjevanja okvirja navzgornje povezave.....	39
Slika 30: Potek razčlenjevanja okvirja navzdolnje povezave .....	40
Slika 31: Preverjanje s testnimi vektorji - primer za SingleBurstRate78Test.....	41
Slika 32: Implementacija dekodirnika diferenčno kodirane modulacije z minimalnim skokom.....	43
Slika 33: Enačba za deviacijo AWGN šuma .....	43
Slika 34: Verjetnost zgrešitve sinhronizacijske besede pri navzdolnji povezavi.....	45
Slika 35: Frekvenca lažne zaznave sinhronizacijske besede pri navzdolnji povezavi .....	46
Slika 36: Verjetnost zgrešitve sredinske sinhronizacijske besede pri navzgornji povezavi .....	48
Slika 37: Frekvenca lažne zaznave sredinske sinhronizacijske besede pri navzgornji povezavi .....	49
Slika 38: Verjetnost zgrešitve podokvirja navzgornje povezave pri zaznavi s sinhronizacijsko besedo in kodirano dolžino .....	50
Slika 39: Frekvenca lažne zaznave podokvirja navzgornje povezave pri zaznavi s sinhronizacijsko besedo in kodirano dolžino .....	51
Slika 40: Verjetnost napačno sprejete kodirane glave pri navzdolnjem prenosu .....	54
Slika 41: Verjetnost napačno sprejete kodirane glave pri navzgornjem prenosu.....	55
Slika 42: Pogostost napake v načinu enkratne oddaje navzdolnje povezave z 255-zložno vsebino .....	58
Slika 43: Pogostost napake v načinu enkratne oddaje navzdolnje povezave s 5-zložno vsebino.....	59
Slika 44: Pogostost napake v načinu večkratne oddaje navzdolnje povezave z 255-zložno vsebino .....	60



Slika 45: Pogostost napake v načinu večkratne oddaje navzdolnje povezave s 5-zložno vsebino .....	61
Slika 46: Pogostost napake v načinu enkratne oddaje navzgornje povezave z 255-zložno vsebino .....	63
Slika 47: Pogostost napake v načinu enkratne oddaje navzgornje povezave s 5-zložno vsebino .....	64
Slika 48: Pogostost napake v načinu večkratne oddaje navzgornje povezave z 255-zložno vsebino .....	65
Slika 49: Pogostost napake v načinu večkratne oddaje navzgornje povezave s 5-zložno vsebino .....	66

## Kazalo tabel

Tabela 1: generatorski polinomi RSC kodirnika v niznem profilu .....	28
Tabela 2: Luknjalne maske RSC kodirnika v niznem profilu .....	28
Tabela 3: Ustvarjeni razredi za kodirne tehnike v OMS LPWAN .....	35
Tabela 4: Verjetnost lažno veljavne glave .....	52
Tabela 5: Mejne vrednosti razmerja energija bita/spektralna gostota šuma za pravilno dekodiranje vsakega drugega okvirja pri navzdolnji povezavi z 255-zložno vsebino .....	57
Tabela 6: Mejne vrednosti razmerja energija bita/spektralna gostota šuma za pravilno dekodiranje vsakega drugega okvirja pri navzgornji povezavi z 255-zložno vsebino .....	62

## Seznam okrajšav

AMI	Napredna merilna infrastruktura (angl. advanced metering infrastructure)
AWGN	Aditivni beli Gaussov šum (angl. additive white Gaussian noise)
BER	Pogostost napake (angl. bit-error rate)
CPFSK	Modulacija s frekvenčnim skokom z zvezno fazo (angl. continuous-phase frequency-shift keying)
CRC	Ciklično preverjanje redundance (angl. cyclic redundancy check)
$E_b/N_0$	Energija bita/spektralna gostota šuma (angl. energy per bit to noise power spectral density ratio)
I/Q	sofazna in kvadratura komponenta (angl. in-phase and quadrature components)
LPWAN	Nizkoenergijsko prostrano omrežje (angl. low-power wide-area network)
M-Bus	Merilno vodilo (angl. Meter Bus)
MSB	Najbolj pomemben bit (angl. most significant bit)
MSK	Modulacija z minimalnim skokom (angl. minimum-shift keying)
OMS	Odprt merilni sistem (angl. open metering system)
RMS	Koren povprečnega kvadrata (angl. root mean square)
SDR	Programsko definiran radio (angl. software defined radio)
SNR	Razmerje signal/šum (angl. signal-to-noise ratio)
UART	Asinhrona serijska komunikacija (angl. universal asynchronous receiver-transmitter)
wM-Bus	Brezžično merilno vodilo (angl. Wireless Meter Bus)
XOR	Ekskluzivni ali (angl. exclusive or)

# 1 UVOD

Napredna merilna infrastruktura (angl. Advanced Metering Infrastructure oz. AMI) predstavlja logičen korak naprej za meritve energentov. Omogoča elektronski dostop do ažurnih merilnih podatkov brez zaposlenih na terenu in brez potrebe po poseganju v privatnost domov. Odstrani tudi možnost človeške napake pri branju števcov, zaradi svoje avtomatizirane narave pa se lahko izvaja veliko bolj pogosto in konsistentno, in lajša analizo porabe tako iz strani ponudnikov energentov, kot porabnikov. OMS (open metering system, slov. odprt merilni sistem) je odprt standard, ki si prizadeva doseči popolno interoperabilnost med AMI napravami različnih proizvajalcev in s tem pohitriti in poceniti prehod merilnih sistemov na AMI, ki je zaradi števila naprav potrebnih v takšnem sistemu že sam po sebi izredno velik infrastrukturni in ekonomski izziv.

Cilj projekta je bil v celoti implementirati generator in razčlenjevalnik okvirjev fizičnega sloja novega komunikacijskega protokola OMS LPWAN (OMS low-power wide-area network, slov. nizkoenergijsko prostrano omrežje OMS), ta pa nato uporabiti kot osnovo za analizo robustnosti tega protokola proti naključnemu belemu šumu.

Pridobljeni rezultati lahko služijo kot neodvisna referenca pri ekonomski evaluaciji prehoda iz obstoječih tehnologij (na primer wM-BUS), ali pri načrtovanju kompatibilnih radijskih sprejemnikov. Z integracijo programsko definiranega radia lahko ustvarjena programska oprema deluje kot OMS LPWAN oddajnik ali sprejemnik brez namenske strojne opreme, kar omogoča cenovno testiranje različnih konfiguracij v laboratoriju ali na terenu.

## 2 TEHNOLOŠKI PREGLED

### 2.1 Aditivni beli Gaussov šum

Šum je komponenta signala, ki ne vsebuje koristne informacije. V mnogih sistemih je šum neželen, in je posledica omejene možnosti izolacije signala od šuma v prenosnem mediju, še posebej v radijskih komunikacijah, kjer je medij skoraj vedno deljen med mnogimi uporabniki in povsem izpostavljen naravnim virom šuma.

Šum v signalu je aditiven, če je signalu prištet. To se zgodi ko se šum in signal srečata v linearnem sistemu. Šum je bel, če ima enako jakost pri vseh frekvencah, kar se kaže v popolni neodvisnosti verjetnostnih porazdelitev vrednosti v posameznih časovnih oknih. Šum je Gaussov, če verjetnostna porazdelitev vrednosti šuma sledi Gaussovi distribuciji v vsakem časovnem oknu.

Aditivni beli Gaussov šum (angl. additive white Gaussian noise oz. AWGN) je model kanala, v katerem se signalu prišteje beli Gaussov šum. Model dobro opisuje vpliv šuma naključnih naravnih virov, kot so na primer toplotno sevanje in toplotne vibracije elektronov v sprejemniku, in predstavlja model šuma ozadja, saj je tovrstni šum neodvisen od signala in zmerom prisoten [1].

Pri analizi komunikacijskega kanala se pogosto navaja povprečna moč šuma relativna na povprečno moč signala, in sicer z razmerjem signal/šum (angl. signal-to-noise ratio oz. SNR) v decibelih. Energija bita/spektralna gostota šuma (angl. energy per bit to noise power spectral density ratio oz.  $E_b/N_0$ ) je alternativna mera, katere prednost je, da direktno predstavlja šumnost posameznih bitov signala, ne glede na to, kako so ti biti predstavljeni v moduliranem signalu. Zato je posebej primerna pri analizi zmogljivosti korekcijskih kodiranj (angl. forward error correction oz. FEC), kjer izbira modulacije vpliva na šumnost simbolov, ne pa neposredno na zmogljivost kodiranja.

## 2.2 Konvolucijsko kodiranje

Pri sprejemanju šumnega digitalnega signala se ne moremo izogniti popačitvi prejetih simbolov. Če lahko vsak bit sporočila kodiramo z več simboli razpršenimi v času, tako da je vpliv šuma na te simbole naključen, potem je napaka simbolov v povprečju enake magnitude v vse smeri prostora simbolov, in se vsota teh napak z večanjem števila simbolov približuje ničli. Zaradi tega bo večja tudi možnost za pravilno dekodiranje, vendar za ceno nižje efektivne prenosne hitrosti.

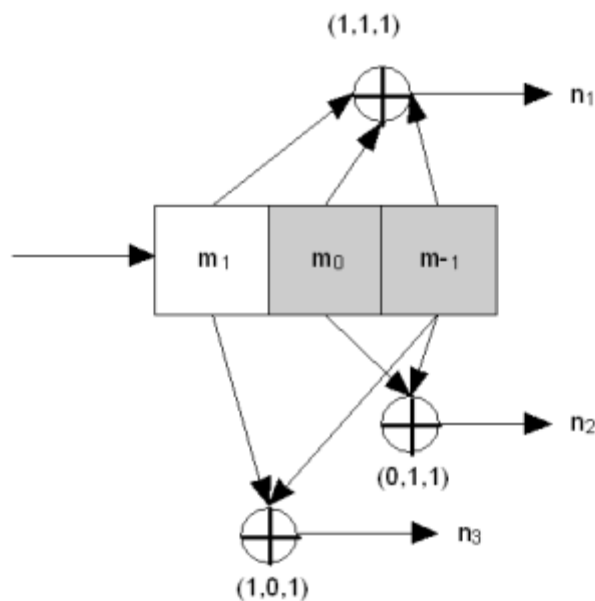
Konvolucijsko kodiranje je tip FEC kodiranja ki uporablja konvolucijo v binarni končni množici, da sporočilu doda dodatne bite oz. redundanco, s katero zviša možnost pravilnega dekodiranja v šumnih pogojih. V tem kontekstu je konvolucija filter z impulznim odzivom končne dolžine, vrednosti katerega definirajo koeficienti konvolucijskega polinoma. Drugače povedano, je preslikava, kjer je vsak bit izhoda skalarni produkt med koeficienti obrnjenega konvolucijskega polinoma in biti drsečega okna nad sporočilom. V binarni končni množici se vsi izračuni izvajajo po modulu 2 in ni prenosov. Skalarni produkt dveh zaporedij bitov je tako lahko implementiran kot vrsta vrat in (angl. and gate oz. AND gate), ki predstavljajo produkt, izhode katerih pa pošljemo v kaskado vrat ekskluzivni ali (angl. exclusive or gate oz. XOR gate), ki predstavljajo vsoto. Pri konvolucijskem kodiranju konvolucijskemu polinomu pravimo generatorski polinom, konvolucija pa je namesto z drsečim oknom pogosto prikazana in implementirana s pomičnim registrom, tako v strojnih kot v programskih implementacijah.

Za konvolucijsko kodiranje uporabimo enega ali več generatorskih polinomov, rezultatom posameznih konvolucij pa pravimo emisije ali paritete. Stopnja konvolucijskega kodirnika je razmerje med številoma vhodnih in izhodnih bitov, oziroma  $1/M$ , kjer je  $M$  število emisij. V vsakem časovnem trenutku je izhod konvolucijskega kodirnika neka množica bitov emisij (angl. emissions), ki je odvisna od vhodnega bita. Pomični register konvolucijskega kodirnika predstavlja stanje kodirnika in ima  $K-1$  bitov, kjer je  $K$  omejitvena dolžina (angl. constraint length) [2] [3]. Generatorski polinomi kodirnika imajo  $K$  členov. Iz vsakega od  $2^K$   $K$ -bitnih

stanj obstaja  $2^K$  prehodov v naslednjo stanje, od katerih pa sta v konvolucijskem kodirniku možna samo dva, saj je vhod 1-biten. Ta omejitev dovoljenih prehodov je ključna lastnost za korekcijsko zmogljivost konvolucijskega kodiranja.

Posebne pomena sta začetno in končno stanje konvolucijskega kodirnika, saj z njima omejimo možne prehode pri dekodiranju začetka in konca kodiranega sporočila. Da dosežemo neko vnaprej določeno končno stanje moramo kodirnik terminirati z vnaprej znano sekvenco vhodnih bitov (po navadi z ničlami) dolžine  $K-1$ . Pri tem nastane  $K-1$  dodatnih bitov emisije, ki jim pravimo repni biti (angl. tail bits).

Na sliki (Slika 1) je prikazan konvolucijski kodirnik stopnje  $1/3$  z omejitveno dolžino 3 implementiran s pomičnim registrom. Senčena kvadrata predstavljata bita pomičnega registra oz. stanja kodirnika. Bel kvadrat predstavlja vhodni bit.  $n_1$ ,  $n_2$  in  $n_3$  so emisije, števila v oklepajih pa koeficienti generatorskih polinomov.



Slika 1: Primer konvolucijskega kodirnika stopnje  $1/3$  z omejitveno dolžino 3 [4]

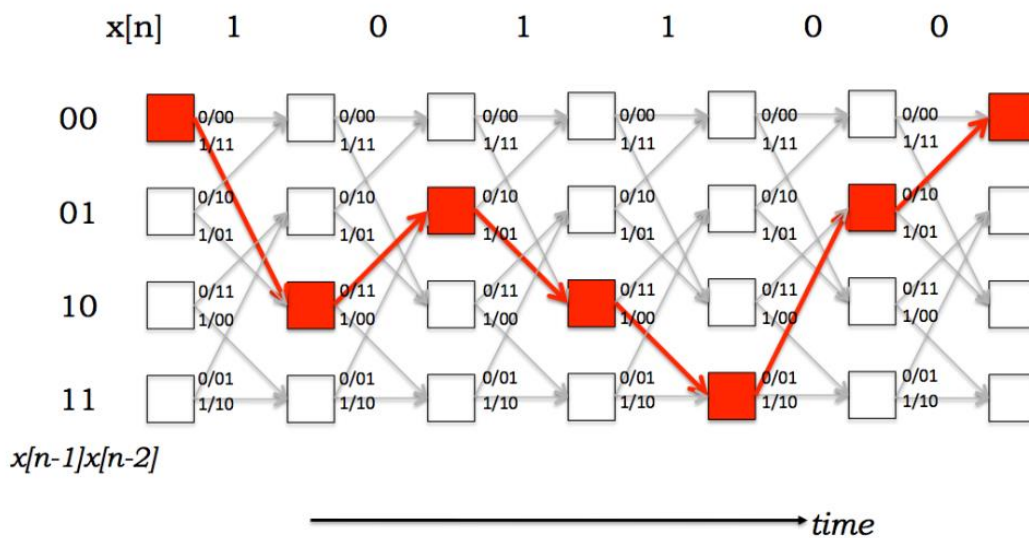
Dekodiranje konvolucijske kode se običajno izvaja z Viterbi dekodirnikom. Ta stopa po bitih emisij po posameznih časovnih trenutkih, v vsakem od katerih sta iz vsakega stanja možna dva prehoda, ki imata dve različni emisiji. S primerjavo teh emisij z dejansko emisijo, kakršna je bila prejeta na sprejemniku, dekodeer določi kateri prehod se je verjetneje zgodil glede na to, emisija katerega prehoda se bolje ujema s prejetjo. Dekoder v vsakem časovnem trenutku vsakemu stanju pripiše strošek poti, ki predstavlja najmanjšo možno vsoto napake zbrano na poti iz začetnega stanja v to stanje. Nato izračuna stroške prehodov v naslednja možna stanja in te stroške prišteje k stroškom poti. V vsako novo stanje vodita dve poti, od katerih pa ima ena manjši strošek poti in je zato bolj verjetna. Začetno in končno stanje sta po navadi vnaprej določena. Ko pride dekodeer do konca emisije sledi optimalni poti iz končnega stanja nazaj v začetno stanje, in pri tem generira pripadajočo zaporedje vhodnih bitov, ki so rezultat dekodiranja.

Na sliki (Slika 2) je prikazan postopek Viterbi dekodiranja konvolucijske kode stopnje 1/1 z omejitveno dolžino 3 prikazan v Trellis grafu. Stolpci predstavljajo posamezne časovne trenutke, številke nad njimi pa bite emisije. Vrstice predstavljajo različna stanja. Sive puščice predstavljajo možne prehode, rdeče puščice pa pot iz začetnega stanja (00) v končno stanje (00) z najnižjim stroškom.

Dekodiranje z mehкими biti poteka tako, da za računanje odstopanja pričakovanih emisij od prejetih ne uporabimo Hammingove razdalje od binarnih vrednosti prejetih simbolov, pač pa Evklidsko razdaljo od decimalnih vrednosti simbolov (na primer med 0 in 1), kjer vmesne vrednosti izražajo negotovost demodulatorja o tem ali je prejel bit 0 ali 1 [3]. Vrednost 0.5 lahko na primer izraža popolno negotovost, saj je enako oddaljena tako od 0 kot od 1, in zato tudi ne vpliva na razlike med stroški prehodov. Pogosteje se bita 0 in 1 predstavita z vrednostma -1 in +1, kjer popolno negotovost izraža vrednost 0.

Pri rekurzivnem konvolucijskem kodiranju vhod v konvolucijski kodirnik ni vhodno zaporedje bitov, temveč povratna veza (angl. feedback), ki je skalarni produkt koeficientov polinoma povratne veze (angl. feedback polynomial) in bitov stanja kodirnika, najbolj pomemben bit (angl. most significant bit oz. MSB) pa je zmnožen z dejanskim vhodnim

bitom. Ker je v binarni končni možici konvolucija koeficientov enaka produktu polinomov, dodajanje (ki je v tem primeru ekvivalentno odštevanju) zmnožkov nazaj k polinomu preko povratne veze pa enako postopku dolgega deljenja, se rekurzija občasno označi kot deljenje s polinomom povratne veze (na primer  $1/g_0$ , kjer je  $g_0$  polinom). Na sliki (Slika 3) je prikazan rekurzivni konvolucijski kodirnik. Povratna veza je izhod iz levega seštevalnika, vhodi vanj pa so določeno s polinomom povratne veze.

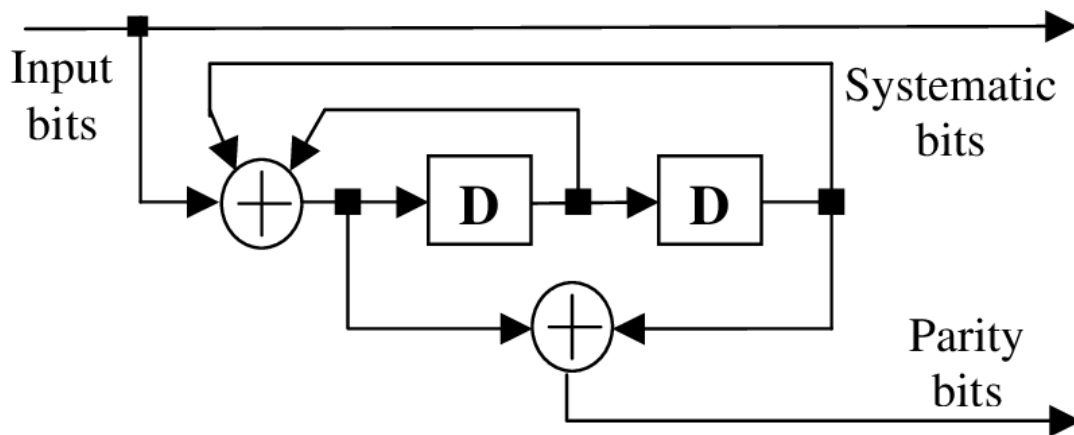


Slika 2: Prikaz postopka Viterbi dekodiranja konvolucijske kodestopnje 1/1 z omejitveno dolžino 3 [5]

Sistematsko kodiranje je kodiranje pri katerem se poleg emisij v sporočilu nahaja tudi originalno zaporedje bitov (sistematski izhod, angl. systematic output), kar računsko poenostavi dekodiranje v nešumnih pogojih, in hitro analizo sporočila, ko pravilnost vsebine ni pomembna. Sistematsko kodiranje se lahko po navadi implementira enostavno kot priključitev originalnega sporočila kodiranemu sporočilu.



Emisija konvolucije je enake velikosti kot vhodno zaporedje bitov (brez upoštevanja repnih bitov). To pomeni, da se za konvolucijsko kodiranje stopnje  $1/2$  efektivna bitna hitrost kanala razpolovi. Če se želi doseči kakšno vmesno stopnjo, na primer  $3/4$ , se lahko to doseže tako, da se od generiranih bitov emisije pošlje samo en del, na primer vsak drugi bit. Temu postopku se reče luknjanje (angl. puncturing), ponavljajoči se bitni maski, ki določa kateri biti se ohranijo pa luknjalna maska (angl. puncturing mask) ali luknjalni vzorec (angl. puncturing pattern).



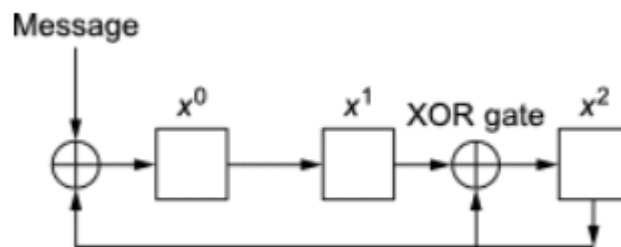
Slika 3: Rekurzivni sistematski konvolucijski kodirnik stopnje  $1/2$  z omejitveno dolžino 3 [6]

Informacija vsakega vhodnega bita se v emisiji nahaja v sosednjih bitih. Ker se napake pri prenosu pogosto dogajajo v kratkih zaporedjih (angl. burst errors) je informacijo koristno še bolj razpršiti. To se lahko stori s premešavanjem (scrambling), ki bite premeša v vnaprej znano psevdo-naključno zaporedje.

## 2.3 Ciklična redundanca

Ciklična redundanca je koda, ki se najpogosteje uporablja za odkrivanje napak v podatkih, postopek česar imenujemo ciklično preverjanje redundance (angl. cyclic redundancy check oz. CRC), tako kot konvolucijsko kodiranje pa se lahko uporabi za odpravljanje napak. Pri CRC kodiranju se iz podatkov izračuna pariteta fiksne dolžine, ki se preko kanala pošlje skupaj s podatki. Zaradi fiksne dolžine, ki je po navadi veliko manjša od dolžine podatkov ima CRC po navadi veliko slabšo zmožnost odprave napak v primerjavi z ostalimi kodirnimi tehnikami zasnovanimi za ta namen, je pa računsko veliko enostavnejše tako za kodiranje, kot za dekodiranje, še posebej z namenskimi digitalnimi vezji.

Za izračun se tako kot pri konvolucijskem kodiranju vhodne bite pošilja v pomični register. Izhod registra, ki predstavlja MSB polinoma stanja se zmnoži z zaporedjem bitov, tudi tukaj imenovanim generatorski polinom, rezultirajoče zaporedje bitov pa se nato odšteje od stanja ko v register pride nov bit. Vsi izračuni se tudi tukaj izvajajo po modulu 2, zato je odštevanje ekvivalentno seštevanju, in se lahko implementira z vrati XOR. Ponavljanje tovrstnega odštevanja je tako kot rekurzija v konvolucijskem kodirniku ekvivalentna postopku dolgega deljenja. Dolžino CRC kodirnika določa dolžina generatorskega polinoma.  $n$ -bitni CRC kodirnik ima polinom dolžine  $n+1$ .

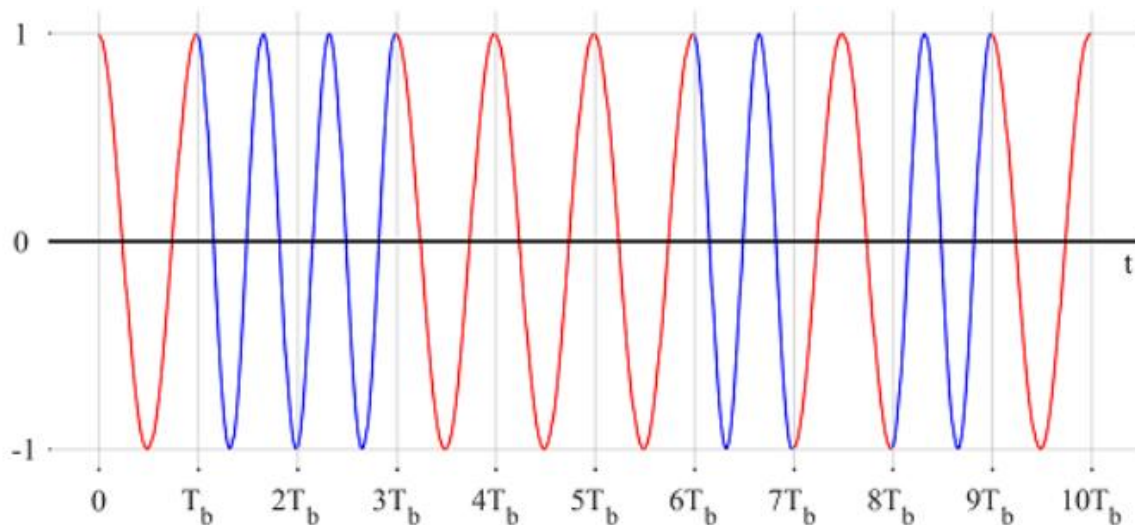


Slika 4: Shema kodirnika s ciklično redundanco [7]

Rezultat CRC kodiranja je pariteta, ki se na koncu nahaja v pomikalnem registru. Posebna lastnost CRC kodiranega sporočila je, da ponovni izračun paritete na sporočilu in na izračuni pariteti proizvede pariteto 0. Tako je z istim vezjem enostavno izvajati tako kodiranje kot preverjanje.

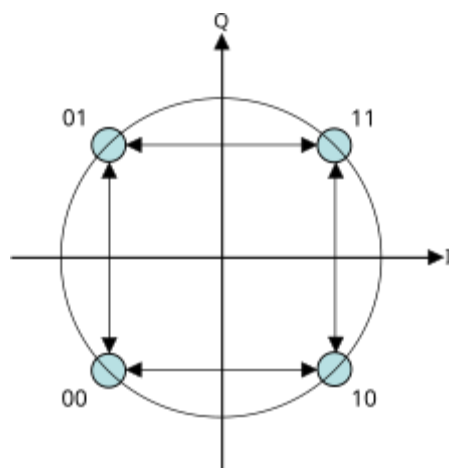
## 2.4 Modulacija z minimalnim skokom

Modulacija z minimalnim skokom (angl. minimum-shift keying oz. MSK) je tip modulacije s frekvenčnim skokom z zvezno fazo (angl. continuous-phase frequency-shift keying oz. CPFSK), pri kateri se v intervalu enega simbola faza spremeni za  $\pm\pi/2$ . Zaradi tega jo lahko predstavimo v konstelacijskem diagramu s štirimi simboli na enotinem krogu, kar je razvidno iz slike (Slika 6), in spominja na ortogonalno kvadraturno fazno modulacijo, saj se v obeh primerih prehodi dogajajo samo med sosednjimi simboli, a se za razliko od nje pri MSK med simboli faza zvezno spreminja.



Slika 5: Časovni potek modulacije z minimalnim skokom [8]

Gaussova MSK (angl. Gaussian MSK oz. GMSK) je poseben primer MSK, pri katerem je vhodni bitni tok pred modulacijo filtriran z Gaussovim filtrom, kar zmanjša zaseden del frekvenčnega spektra, a zviša med-simbolno interferenco.



Slika 6: Modulacija z minimalnim skokom v konstelacijskem diagramu [9]

## 2.5 Korelacijsko iskanje sinhronizacijske besede

V komunikacijskih sistemih kjer se sporočila prenašajo v nizih (angl. bursts) mora sprejemnik določiti kdaj je na kanalu tišina, in kdaj se prične sporočilo. V sistemih s konstanto jakostjo signala in brez pomenljivega šuma je moč zaznati spremembo iz enega simbola v drugega, kot na primer začetni bit v asinhroni serijski komunikaciji (angl. universal asynchronous receiver-transmitter oz. UART). V radijskih komunikacijah, kjer pa je jakost tako šuma kot signala zelo spremenljiva, ta pristop ne deluje. Rešitev je oddati nenaključno (vnaprej znano) zaporedje simbolov, ki ga lahko sprejemnik poišče v šumnem signalu. Daljše kot je to zaporedje, manj je podobno zaporedju naključnih vzorcev šuma, in ju je zato lažje ločiti pri enakem SNR.

Takšnemu zaporedju pravimo sinhronizacijska beseda (angl. syncword), podobnost med dvema zaporedjema pa lahko merimo z njuno križno korelacijo. Sprejemnik lahko torej neprestano računa križno korelacijo med znano sinhronizacijsko besedo shranjeno na primer v pomnilniku in prejetim signalom. Ko je korelacija (podobnost) dovolj visoka sprejemnik smatra, da je prejel sinhronizacijsko besedo in nadaljuje z branjem sporočila.

Sama vrednost korelacije pa je še vedno odvisna od SNR, zato je zanjo potrebno izbrati pragovno vrednost (angl. threshold), nad katero se smatra, da je bila zaznana sinhronizacijska beseda. Nižje pragovne vrednosti povečajo verjetnost, da bo pri nekem SNR zaznana sinhronizacijska beseda, a tudi povečajo verjetnost lažnih zaznav, ki odvisno od zasnove sprejemnika povečajo porabo energije, lahko pa celo povzročijo izgubo dejanskih nizov.

Slika (

Slika 7) prikazuje enačbo za izračun časovno-diskretne korelacije v otipku j normalizirane s faktorjem  $1/N$ , kjer je  $N$  dolžina krajšega zaporedja, na primer sinhronizacijske besede. Iz enačbe je razvidno, da se izvaja skalarni produkt na drsečem oknu nad zaporedjem  $x_2$ . Korelacija se od konvolucije razlikuje predvsem v tem, da obe zaporedji množimo v originalnem vrstnem redu, načeloma pa tudi v tem, da jo v kontekstu iskanja sinhronizacijske besede ne izvajamo v končnem polju, in imajo zato vrednosti korelacije lahko poljubne decimalne vrednosti.

$$r_{12}[j] = \frac{1}{N} \sum_{n=0}^{N-1} x_1[n]x_2[n+j]$$

Slika 7: Enačba za izračun časovno-diskretne korelacije [10]

Magnituda navadne križne korelacije je odvisna od amplitud obeh zaporedij, kar pa ni ugodno ko so amplitude zelo spremenljive. Temu se lahko izognemo če korelacijo normaliziramo s korekcijskim faktorjem, ki je osnovan na korenu povprečnega kvadrata (angl. root mean square oz. RMS) vrednosti zaporedij.

$$\rho_{12}[j] = \frac{r_{12}[j]}{\frac{1}{N} \left[ \sum_{n=0}^{N-1} x_1^2[n] \sum_{n=0}^{N-1} x_2^2[n] \right]^{1/2}},$$

Slika 8: Korekcijski faktor za normalizacijo križne korelacije [10]

## 2.6 Standard OMS LPWAN

M-Bus (Meter Bus) je evropski standard EN13757 za napredno merilno infrastrukturo (angl. advanced metering infrastructure oz. AMI), ki je komunikacijski sistem za avtomatizirano digitalno zbiranje meritev energentov z namenom nadaljnjega procesiranja in shrambe. wM-Bus (Wireless Meter Bus) je četrti del standarda oz. EN13757-4, ki opisuje komunikacijo med pametnimi merilniki in komunikacijskim prehodom (angl. gateway), ki zbira merilne podatke iz mnogih merilnikov.

OMS združenja OMS Group je odprt komunikacijski standard, osnovan na M-Bus, katerega namen je zagotoviti interoperabilnost AMI sistemov različnih proizvajalcev. OMS LPWAN je priloga Q k temu standardu, ki opisuje nov profil za brezžično komunikacijo z izboljšanimi prenosnimi lastnostmi in poenostavljenim povezavnim slojem v primerjavi z obstoječimi profili wM-Bus. Razdelitev standarda OMS je prikazana na sliki (Slika 10), kjer senčena polja predstavljajo OMS LPWAN.

EN13757	Purpose
1	Basic data communication between meters and collectors
2	Physical layer requirements for wired M-Bus
3	Application layer
4	Physical and Data Link layers for wireless M-Bus
5	Relaying and routing for range enhancement
6	Local bus for short distance wired links

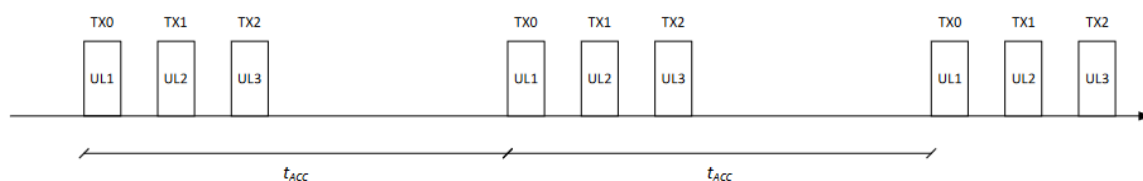
Slika 9: Deli M-Bus standarda [11]

OSI Model		OMS LPWAN Layer Model	
Application Presentation		APL [OMS-S2], clause 8 <sup>a</sup>	
Session Transport		TPL [OMS-S2], clause 7 <sup>a</sup>	
		AFL [OMS-S2], clause 6 <sup>a</sup>	
Data Link	Logical Link Control (LLC) <sup>b</sup>	Wireless M-Bus Frame Format C	
	Medium Access Control (MAC)	Wireless M-Bus MAC	
Physical		Burst Mode	Splitting Mode
<sup>a</sup> optional upper protocol layer <sup>b</sup> The LLC may optionally contain the ELL. When using Frame Format C the ELL can be avoided.			

Slika 10: Razdelitev standarda OMS

### 2.6.1 Komunikacijski profili

Nizni profil (angl. burst mode) je eden izmed dveh profilov komunikacije definiranih za fizični sloj OMS LPWAN, in za razliko od razpršenega profila (angl. splitting mode) ne uporablja patentiranih tehnologij. Protokolna podatkovna enota v niznem profilu so okvirji, njihov format pa se razlikuje med navzgorjo povezavo (od metra do prehoda, angl. uplink) in navzdoljno povezavo (od prehoda do metra, angl. downlink). Format se v obeh scenarijih dalje deli na dva načina oddaje (angl. burst modes) in različne tipe oddaje (angl. burst types). Okvir protokola, v katerem se prenese od 5 do 255 zlogov vsebine (angl. payload), je v načinu enojne oddaje (angl. single-burst mode) sestavljen iz enega podokvirja, ki se po kanalu pošlje kot en niz simbolov (angl. burst). V načinu večkratne oddaje (angl. multi-burst mode) je sestavljen iz treh podokvirjev, ki se pošljejo kot trije nizi ločeni s premori.



Slika 11: Časovni potek oddaje treh okvirjev v načinu večkratne oddaje navzgornje povezave

### 2.6.2 Format podokvirjev

Začetek podokvirjev navzdolnje povezave (Slika 12) označuje sinhronizacijska beseda, kateri takoj sledi kodirana glava, nato pa kodirana vsebina. V podokvirjih navzgornje povezave (Slika 13) se glava skupaj s sredinsko sinhronizacijsko besedo (angl. midamble) nahaja na sredini podokvirja, med dvema polovicama kodirane vsebine, sinhronizacijski besedi na začetku podokvirja pa namesto glave sledi kodirana dolžina. Pred vsakim okvirjem obeh povezav se nahaja še preambula, ki je namenjena sinhronizaciji sprejemne ure.



Preamble	Sync	Coded Header	Data
$B_{PRE}$ bits	$B_{SYNC}$ bits	$B_{CH}$ bits	$L_D$ bytes

Slika 12: Format podokvirja navzdolnje povezave

Preamble	Sync	CL	Data A	Midamble	Coded Header	Data B
$B_{PRE}$ bits	$B_{SYNC}$ bits	$B_{CL}$ bits	$L_{DA}$ bytes	$B_{MID}$ bits	$B_{CH}$ bits	$L_{DB}$ bytes

Slika 13: Format podokvirja navzgornje povezave

Kodirana glava vsebuje 5 parametrov, in poleg časa prejema podokvirja relativnega na čase prejema ostalih podokvirjev okvirja enolično definira format, in s tem postopek razčlenjevanja. Parametri glave so verzija, dolžina vsebine, vhodna časovna vrednost (angl. timing input value), način oddaje in tip oddaje. Verzija obravnavanega protokola je 0. Parametri kodirane glave so kodirani z 8-bitnim cikličnim kodirnikom, vse skupaj pa nato še s konvolucijskim kodirnikom. Generatorski polinom cikličnega kodirnika je  $107_h$ .

Kodirana dolžina vsebuje dolžino prve polovice kodirane vsebine, in s tem določi položaj sredinske sinhronizacijske besede in kodirane glave, ki tej polovici sledita. Ta je kodirana s 15-bitnim cikličnim kodirnikom z generatorskim polinomom  $C617_h$ .

Vsak podokvir okvirja vsebuje enako kodirano glavo, različni podokvirji pa poleg polnil vsebujejo različne dele kodirane vsebine okvirja oziroma različne emisije konvolucijskega kodirnika. V načinu enojne oddaje je format določen s tipom oddaje, s formatom pa je določena tudi efektivna stopnja konvolucijskega kodirnika. V tem načinu so možni trije tipi oddaje: kodiranje stopnje 7/8 (angl. FEC rate 7/8), kodiranje stopnje 1/2 (angl. FEC rate 1/2)

in kodiranje stopnje 1/3 (angl. FEC rate 1/3). V načinu večkratne oddaje tip oddaje ne označuje formata kodirane vsebine. Namesto tega so določeni trije formati, eden za vsakega od treh podokvirjev.

Name	Size (bits)	Description / range
Version	2	00 <sub>b</sub> for initial version
PHY Payload length	8	Length, $L_P$ , in bytes from 5 to 255
Timing Input Value	7	Input value for timing from 0 to 127.
Burst mode	1	0 Single-burst 1 Multi-burst
Burst type	2	Type of burst – see subclause Q.2.4.3.1.9.6 and Q.2.4.3.2.4.
Coded header CRC	8	Checksum of the above fields
FEC parity CH1	28	FEC parity 1 for coded header
FEC parity CH2	28	FEC parity 2 for coded header
FEC tail CH1	6	FEC tail 1 for coded header
FEC tail CH2	6	FEC tail 2 for coded header
<b>Total length of uplink coded header</b>	<b>96</b>	

Slika 14: Format kodirane glave

$L_{DA}$	<b>CRC15(<math>L_{DA}</math>)</b>
9 bits	15 bits

Slika 15: Format kodirane dolžine

PHY Payload	7/8-padding	FEC parity 3A	FEC tail 0	Padding
$B_P$ bits	$B_{pad78}$ bits	$\frac{B_{FEC}}{7}$ bits	6 bits	2 bits

PHY Payload	FEC parity 1	FEC tail 1	Padding
$B_P$ bits	$B_{FEC}$ bits	6 bits	2 bits

PHY Payload	FEC parity 1	FEC tail 1	Padding	FEC parity 2	FEC tail 2	Padding
$B_P$ bits	$B_{FEC}$ bits	6 bits	2 bits	$B_{FEC}$ bits	6 bits	2 bits

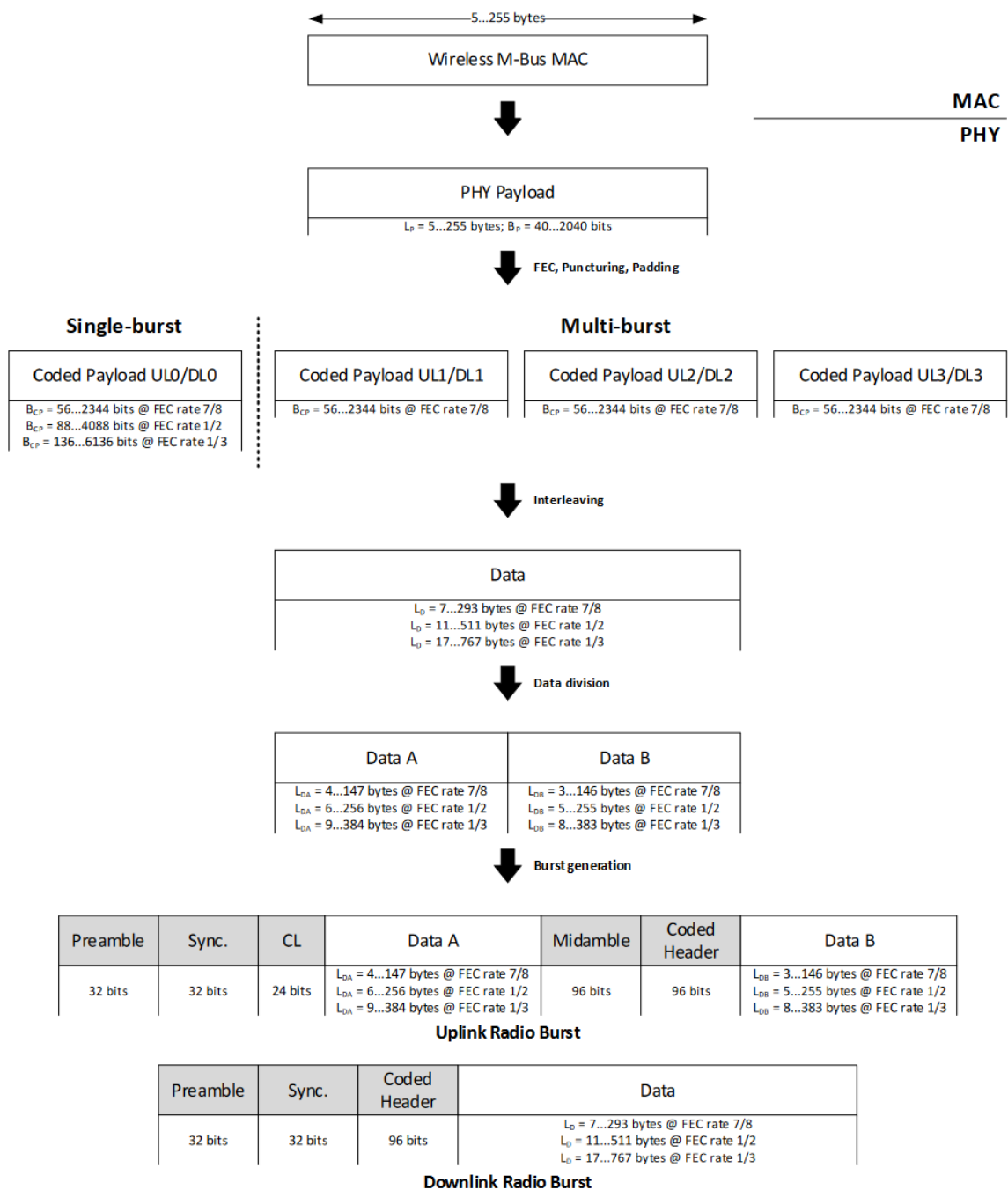
Slika 16: Format kodirane vsebine v načinu enojne oddaje. Po vrstnem redu: kodiranje stopnje 7/8, stopnje 1/2 in stopnje 1/3

PHY Payload	7/8-padding	FEC parity 3A	FEC tail 0	Padding
$B_P$ bits	$B_{pad78}$ bits	$\frac{B_{FEC}}{7}$ bits	6 bits	2 bits

FEC parity 1	FEC parity 3B	FEC tail 1	Padding
$B_{FEC}$ bits	$\frac{B_{FEC}}{7}$ bits	6 bits	2 bits

FEC parity 2	FEC parity 3C	FEC tail 2	Padding
$B_{FEC}$ bits	$\frac{B_{FEC}}{7}$ bits	6 bits	2 bits

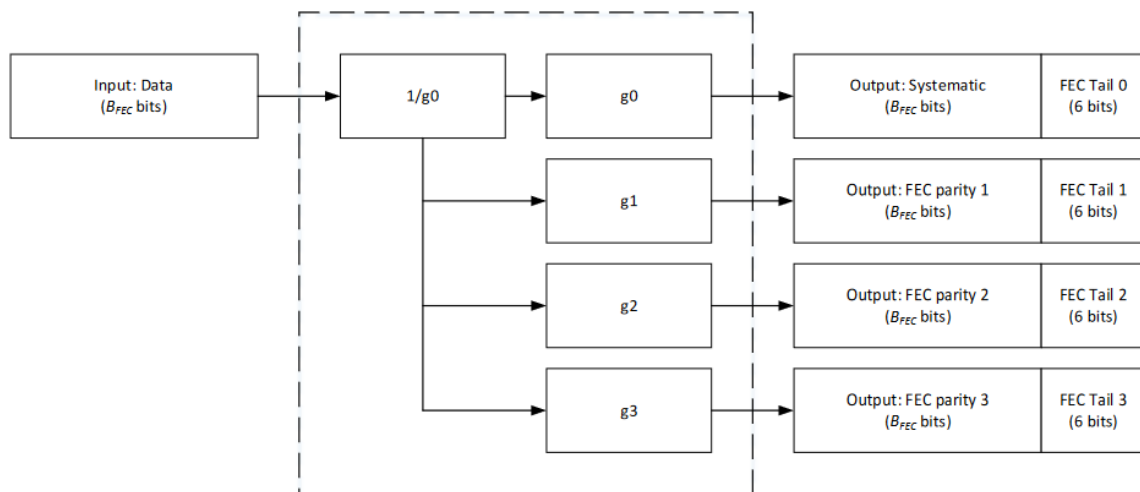
Slika 17: Format kodirane vsebine v načinu večkratne oddaje. Po vrstnem redu: podokvir 1, podokvir 2 in podokvir 3



Slika 18: Tok podatkov in struktur pri generiranju podokvirjev niznega načina

### 2.6.3 Korekcijsko kodiranje

Tako vsebina kot glava sta kodirani z istim rekurzivnim sistematskim konvolucijskim (angl. recursive systematic convolutional oz. RSC) kodirnikom omejitvene dolžine 7 in stopnje 1/4, ki iz vhodnega sporočila ustvari 4 emisije, ena izmed katerih je preluknjana s tremi različnimi maskami. Kodiranje se vedno začne in terminira v stanju 0, repni biti pa so v formatih navedeni ločeno od emisij. Za povratno vezo kodirnika je uporabljen generatorski polinom  $g_0$ . Sistematska emisija je dosežena tako, da za emisijo 0 prav tako uporabimo polinom  $g_0$ . Ta emisija vsebuje sistematski izhod, kateremu pa sledijo repni biti emisije 0. Emisije 3A, 3B in 3C so preluknjane različice emisije 3. Luknjanje določajo tri maske dolžine 7.



Slika 19: Diagram RSC kodirnika v niznem profilu

Zaporedje bitov kodirane vsebine je pred vstavitvijo v podokvir še premešano s psevdonaključnim premešalnikom oz. prepletevalnikom (angl. interleaver).

Tabela 1: generatorski polinomi RSC kodirnika v niznem profilu

Polinom	Vrednost
<b>g0</b>	4D <sub>h</sub>
<b>g1</b>	73 <sub>h</sub>
<b>g2</b>	67 <sub>h</sub>
<b>g3</b>	5D <sub>h</sub>

Tabela 2: Luknjalne maske RSC kodirnika v niznem profilu

Maska	Vrednost
<b>P<sub>3A</sub></b>	1000000 <sub>b</sub>
<b>P<sub>3B</sub></b>	0100000 <sub>b</sub>
<b>P<sub>3C</sub></b>	0010000 <sub>b</sub>

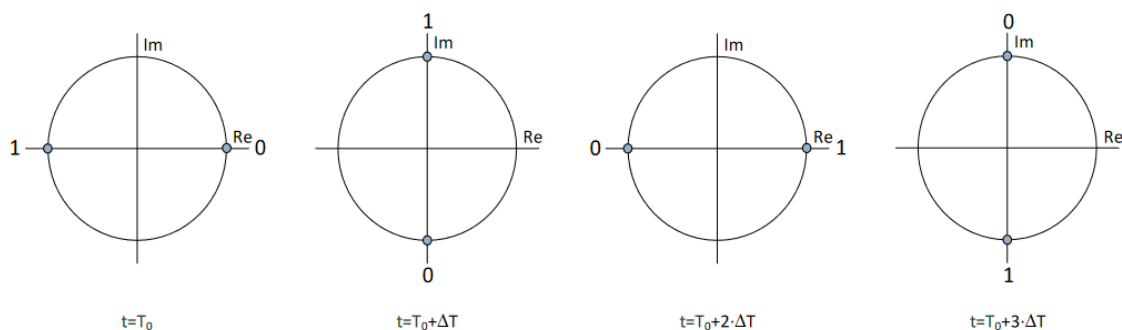
$$s(i) = (188527 \cdot i) \text{ modulo } B_{CP} \text{ for } i \in \{0,1,2, \dots B_{CP} - 1\}$$

Slika 20: Enačba premešalnika RSC kodirnika v niznem profilu

#### 2.6.4 Modulacija

Pri navzdolnji povezavi se uporablja frekvenčno-skočna modulacija (angl. frequency shift keying oz. FSK), kjer nižja frekvenca predstavlja bit 0, višja pa bit 1. Pri navzgornji povezavi pa se uporablja GMSK, pri čemer se zaporedje bitov pred modulacijo še diferenčno kodira.

Tako lahko demodulator originalni signal prebere direktno iz faze simbolov, ali še enostavneje iz realne in imaginarne komponente posebej, saj se diferenčni naravi GMSK in diferenčnega kodiranja izničita, in so originalni biti predstavljeni z absolutno fazo po vrtečem se vzorcu, ki se ponavlja vsake štiri simbolne intervale. Na sliki (Slika 21) so štiri konstelacijski diagrami, ki predstavljajo ta vzorec. Številke predstavljajo vrednosti originalnih bitov. [12]

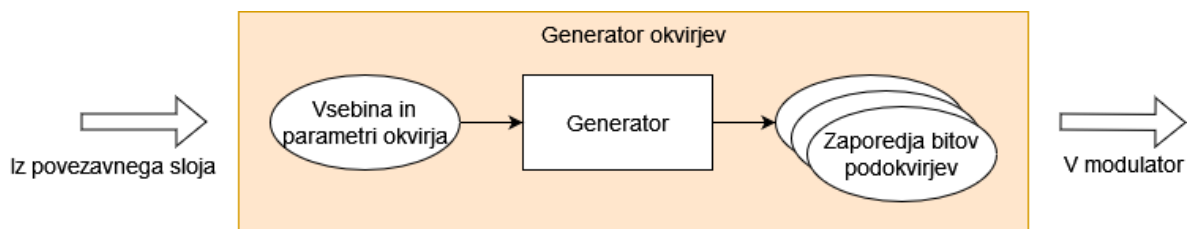


Slika 21: konstelacijski diagram diferenčno kodirane GMSK navzgorne povezave

### 3 ZASNOVA IN IZVEDBA

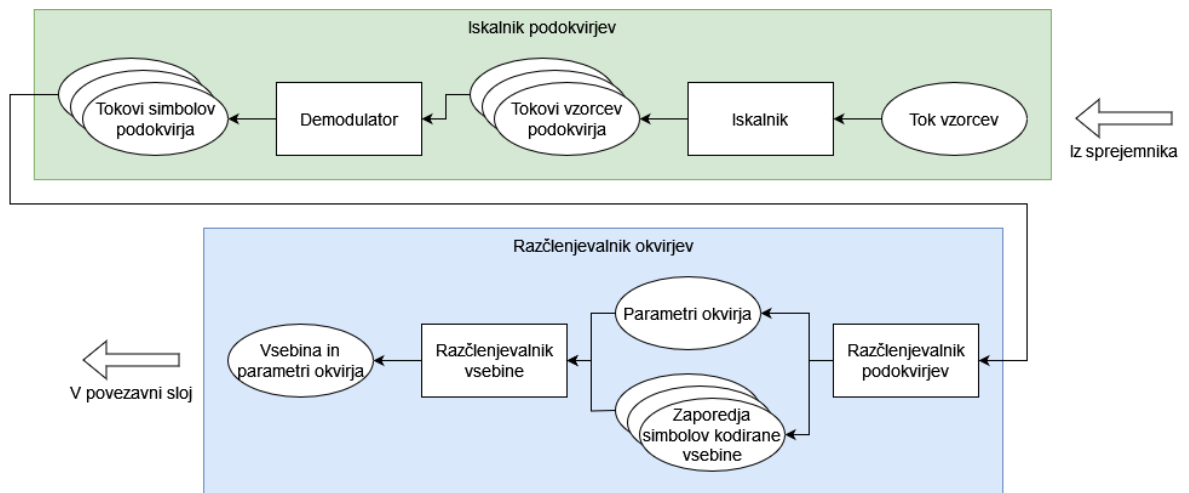
Da lahko sporočilo pošljemo preko komunikacijskega kanala, moramo tvoriti zaporedje simbolov, ki enoumno definira potek stanja oddajnika po času, in s tem tudi potek signala, ki ga oddajnik odda. Ko sprejemnik prejme signal, ga pretvori v zaporedje simbolov, ki znotraj določenih omejitev prav tako enoumno definira časovni potek signala, kakršnega je prejel sprejemnik.

Cilj tega dela projekta je bil zasnovati in implementirati programski vmesnik, ki v celoti poskrbi za postopek generacije in razčlenjevanja OMS LPWAN okvirjev, tako da je edina vloga zunanje kode pri procesiranju okvirjev preko tega vmesnika dovajati vhodne podatke in brati izhodne podatke. Pri generaciji okvirja so vhodni podatki binarna vsebina in parametri protokola fizičnega sloja, izhodni podatek pa je zaporedje bitov ustrezno za modulacijo. Pri razčlenjevanju je vhodni podatek tok zaporedja vzorcev iz sprejemnika, izhod pa binarna vsebina in parametri protokola fizičnega sloja, takrat ko je v toku vzorcev bil uspešno najden in dekodiran okvir. Na sliki (Slika 23) je razvidno, da je pri razčlenjanju tokov vzorcev z neznano vsebino potrebno pred razčlenjanjem okvirjev potrebno prvo najti podokvirje, zato je razčlenjanje razdeljeno na dva sklopa. Zaradi spremenljive dolžine podokvirjev in njihovega razmaka po času je potrebno razčlenjevanje početi v več časovno zakasnenih korakih.



Slika 22: Blok diagram generatorja okvirjev





Slika 23: Blokovni diagram razčlenjevalnika okvirjev iz zaporedja neznanih vzorcev

### 3.1 Programsko okolje

Vsa programska koda je napisana v jeziku Python. Jezik je bil izbran zaradi njegove ustreznosti za hitro prototipiranje in dostop do širokega nabora knjižnic za procesiranje podatkov. Za vektorske operacije na zaporedjih bitov in simbolov je bila uporabljena knjižnica NumPy, za pretvorbo med bitno in številsko predstavitev knjižnica bitstring, za izris grafov knjižnica Matplotlib, ter za optimizacijo časovno kritičnih rutin knjižnica Numba.

### 3.2 Predstavitev podatkov

Marsikateri podatek vsebovan v okvirju ima dve pomembni predstavitvi, do katerih je potrebno dostopati med generacijo in razčlenjevanjem: predstavitev z vrednostjo, najpogosteje številom, ter predstavitev z zaporedjem bitov. Predstavitev z vrednostjo je potrebna za podatke, od katerih je odvisna oblika okvirja, na primer podatek o dolžini vsebine, ali podatek o uporabljenem tipu FEC. Vsak podatek pa se hrani tudi kot zaporedje

bitov, saj se v takšni obliki nahaja v podokvirju. Vsak podokvir ima pripadajoče zaporedje bitov, bodisi namenjeno v oddajnik, ali kakršno je bilo prejeto iz sprejemnika.

Za enostavno predstavitev podokvirjev z zaporedji bitov je bil ustvarjen Python razred, ki omogoča hierarhični opis struktur kot skupine polj, kjer so posamezna polja lahko zaporedja bitov, ali pa podskupine polj. Glavna motivacija za izdelavo tega razreda je bila potreba po delu z velikosti polj, ki niso celoštevilski večkratniki enega zloga. Ustvarjen je bil tudi Python modul z definicijami vseh struktur vsebovanih v okvirjih niznega profila.

```
class BurstModeUplink(FieldGroup):

    PREAMBLE = 0x66666666
    SYNCWORD = 0x8153884C
    MIDAMBLE = 0xDF46428F20B9BD70DF46428F

    def __init__(self):
        with self.defineFields():
            self.preamble = Field(32, uint=BurstModeUplink.PREAMBLE)
            self.syncword = Field(32, uint=BurstModeUplink.SYNCWORD)
            self.coded_length = CodedLength()
            self.data_a = Field()
            self.midamble = Field(96, uint=BurstModeUplink.MIDAMBLE)
            self.coded_header = CodedHeader()
            self.data_b = Field()
```

Slika 24: Definicija strukture s skupino polj - primer za podokvir navzgorne povezave

Pri procesiranju okvirjev je prisotnih veliko podatkov, potrebnih v obeh predstavitev, ki se nanašajo na različne dele strukture okvirja. Za lažje delo s temi podatki je bil ustvarjen razred, v katerem so poleg skupin polj z uporabo podrazredov hierarhično shranjene tudi vrednosti. Ta razred olajša prenos razčlenjenih podatkov med različnimi fazami razčlenitve, olajša pa tudi ponovno uporabo že generiranih struktur, ki se v podokvirjih ponovijo, na primer kodirana glava okvirja.

```

class CodedPayload:
    def __init__(self):
        self.single_burst_fec_7_8_struct: Optional[structs.CodedPayloadSingleBurstFec78] = None
        self.single_burst_fec_1_2_struct: Optional[structs.CodedPayloadSingleBurstFec12] = None
        self.single_burst_fec_1_3_struct: Optional[structs.CodedPayloadSingleBurstFec13] = None
        self.multi_burst_1_struct: Optional[structs.CodedPayloadMultiBurst1] = None
        self.multi_burst_2_struct: Optional[structs.CodedPayloadMultiBurst2] = None
        self.multi_burst_3_struct: Optional[structs.CodedPayloadMultiBurst3] = None

        self.phy_payload = None

        self.bits_payload = None
        self.bits_padding_7_8 = None
        self.bits_fec = None
        self.bits_coded_payload = None
        self.fec_parities = coding.CommonFecEncodingSchemeParities()

```

Slika 25: Razred za enostaven dostop do struktur in vrednosti - primer za kodirano vsebino

Ker implementiran Viterbi dekodeer podpira dekodiranje mehkih bitov, se pri razčlenjevanju zaporedje prejetih simbolov predstavi s kodiranjem brez vračanja v ničelno vrednost (angl. non-return-to-zero oz. NRZ) s podatkovnim tipom plavajoče vejice. Tako število -1 predstavlja bit 0, število +1 pa bit 1. To število predstavlja amplitudo simbola, kot jo je določil demodulator, kar je informacija, ki bi se izgubila, če bi se zaporedje shranilo v binarni obliki. NRZ je tudi glavna oblika, v kateri se hranijo vsa zaporedja bitov v programski kodi, saj to omogoča poenostavitev enačb pri dekodiranju FEC. Razred za delo s polji omogoča enostavno pretvorbo med zaporedji bitov in NRZ števili.

### 3.3 Kodiranje podatkov

Od vseh kodiranj uporabljenih v OMS LPWAN okvirjih je za implementacijo najbolj zahteven konvolucijski kodirnik. Zaradi specifičnosti zasnove konvolucijskega kodirnika, in da bi bilo v prihodnosti omogočeno testiranje alternativnih zasnov, je bil izdelan splošno-namenski

kodirnik, ki podpira poljubno število in velikost generatorskih polinomov ter opsijski rekurzivni polinom. Viterbi dekodier ima podporo za mehke bite, ter omogoča generiranje iskalne tabele za hitrejše dekodiranje. Da bi se lahko v smiselnem času testiralo kodiranje in dekodiranje velikega števila okvirjev, je ta kodirnik bil napisan z uporabo zank namesto z vektorskimi operacijami. Ta oblika je primerna za optimizacijo s knjižnico Numba.

```
@numba.njit(**NUMBA_PARAMS)
def dotProduct(a: int, b: int, size: int) -> int:
    mul = a & b
    dot = 0
    for _ in range(size):
        dot ^= (mul & 0b1)
        mul >>= 1
    return dot
```

Slika 26: Optimizacija funkcije s knjižnico Numba - primer za dotProduct

Pri dekodiranju niso nikoli prisotne vse emisije in vsi repni biti skupnega kodirnika stopnje 1/4, ki se uporablja pri kodiranju. Ker jih dekodirnik kljub vsemu pričakuje, se manjkajoči biti v NRZ kodiranju zapolnijo z ničlami. Enako velja za manjkajoče bite preluknjane emisije 3.

Na osnovi implementiranega konvolucijskega kodirnika je bil zgrajen tudi CRC kodirnik kot sistematični rekurzivni konvolucijski kodirnik stopnje 1/1, kjer sta polinom povratne veze in konvolucijski polinom enaka. Repni biti predstavljajo CRC pariteto.

Za vsako kodirne tehniko uporabljeno v OMS LPWAN burst mode je bil ustvarjen razred za kodiranje in dekodiranje podatkov, ki inicializira ustrezne kodirnike s konstantami iz standarda.

Naziv razreda	Metode
<b>CommonEncodingScheme</b>	encode, decode
<b>CommonInterleavingScheme</b>	interleave, deinterleave
<b>Precoder</b>	encode, decode
<b>CodedLengthCrc</b>	encode, decode, parity, check
<b>CodedHeaderCrc</b>	encode, decode, parity, check

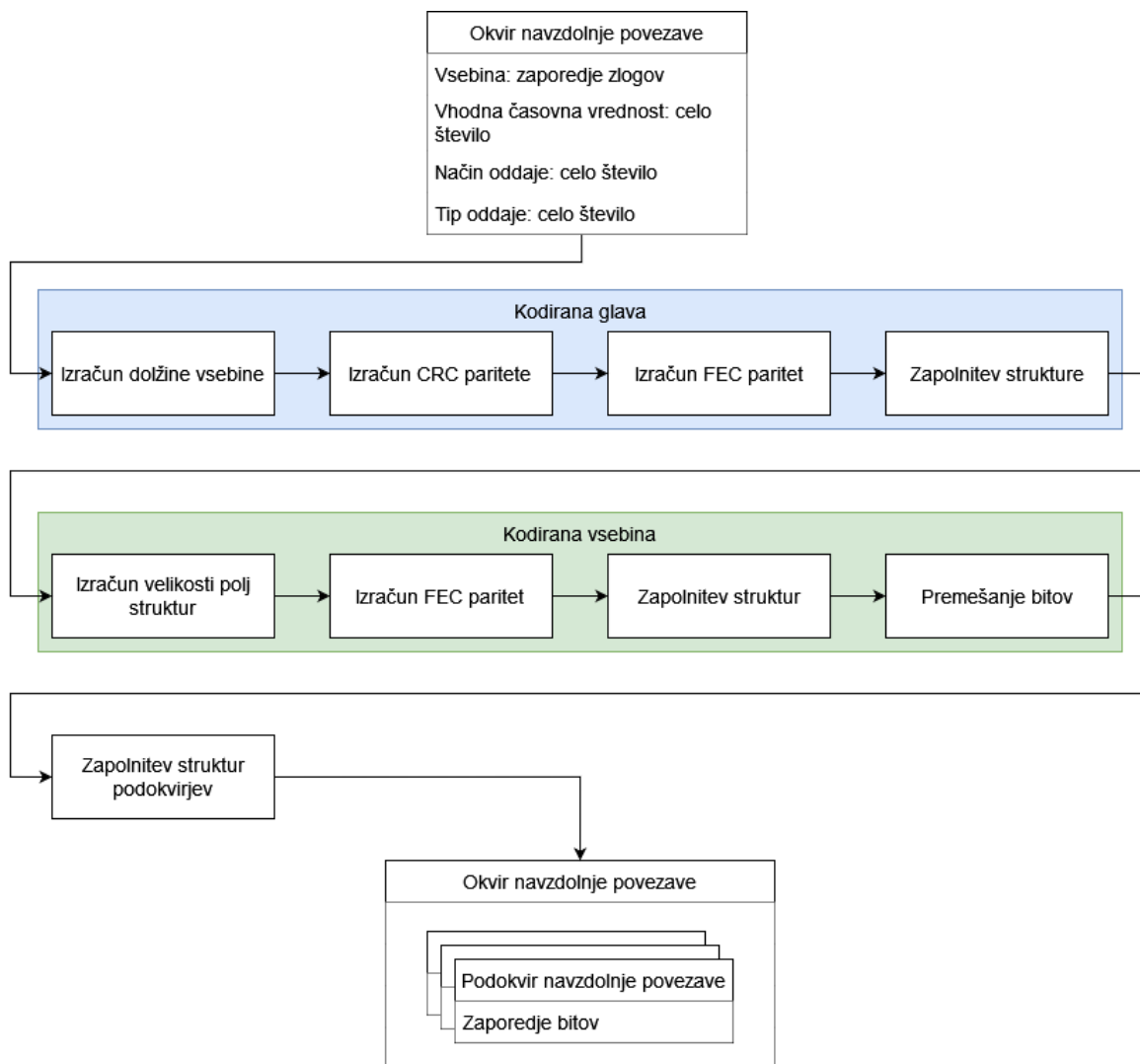
Tabela 3: Ustvarjeni razredi za kodirne tehnike v OMS LPWAN

### 3.4 Generator okvirjev

Razred generatorja sestavlja vrsta metod, ki omogočajo, da se okvir sestavi po korakih, kar izboljša berljivost kode. Ustvarjena je bila metoda `generateFrame`, ki kliče vse ostale metode po vrsti, da se lahko z enim klicem ustvari celoten okvir. Posamezne metode izračunajo in zapolnijo različne dele strukture (na primer kodirano glavo), hkrati pa izračunajo vrednosti, kot so dolžine polj, in jih shranijo v objekt okvirja, da jih lahko uporabijo tudi druge metode. Generator se uporabi tako, da se ustvari prazen objekt okvirja, v katerega se zapišejo binarna vsebina ter parametri glave, nato pa se ta objekt poda v `generateFrame`. Ko je generacija končana se v objektu okvirja nahaja eno ali več bitnih zaporedij, ki jih lahko pošljemo na oddajnik ali v simulacijo.



Slika 27: Potek generacije okvirja navzgorne povezave



Slika 28: Potek generacije okvirja navzdolnje povezave

### 3.5 Razčlenjevalnik okvirjev

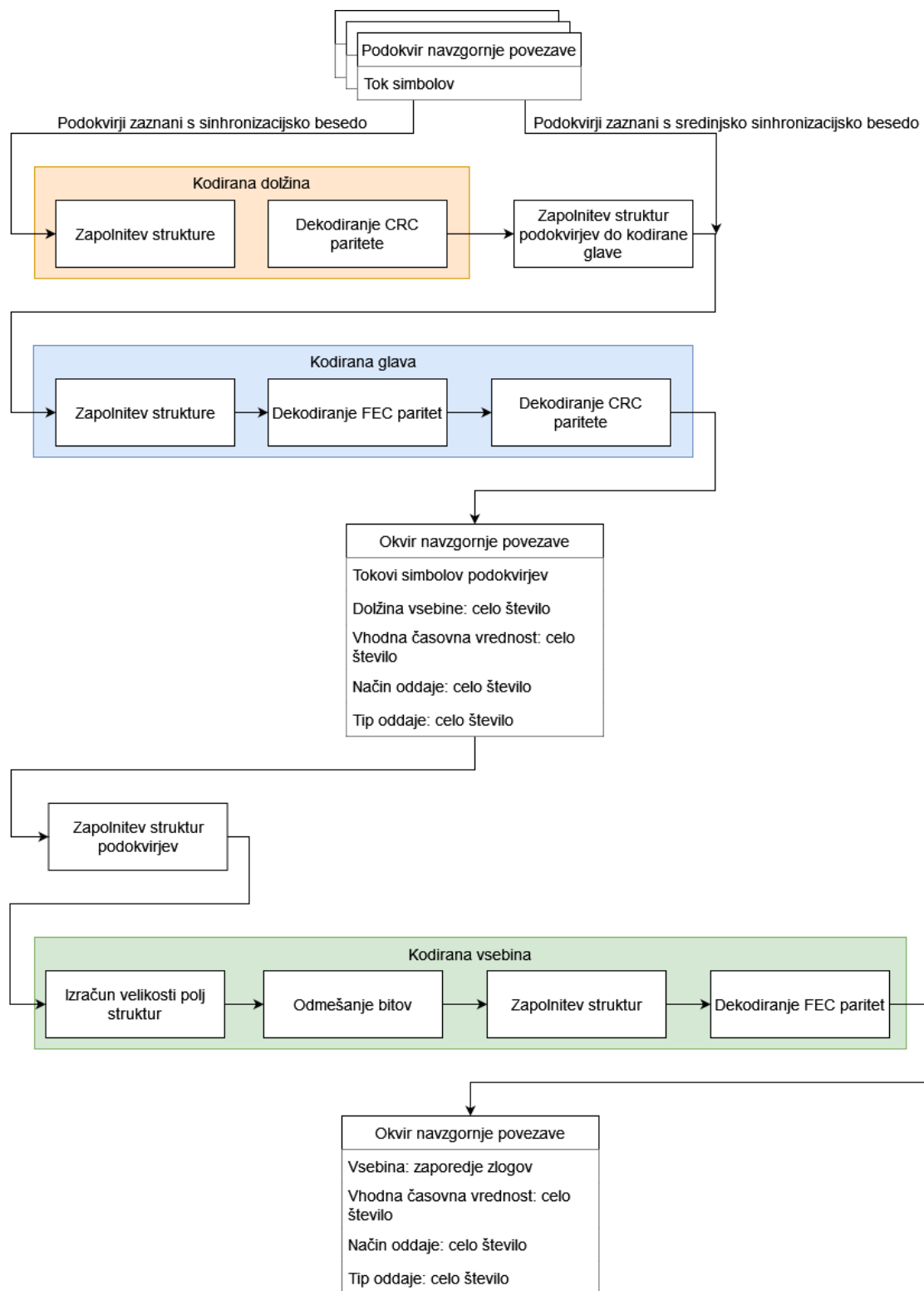
Tako kot generacija se razčlenjanje prične s zapolnitvijo objekta z vhodnimi podatki. V tem primeru se objekt podokvirja zapolni z zaporedjem mehkih bitov iz demodulatorja. Razčlenjevanje se ne more pričeti z objektom okvirja, saj okvir vsebuje več podokvirjev,

medtem ko sprejemnik prejema le en podokvir naenkrat, pri čemer ni vnaprej znano kateri podokvirji sodijo v isti okvir.

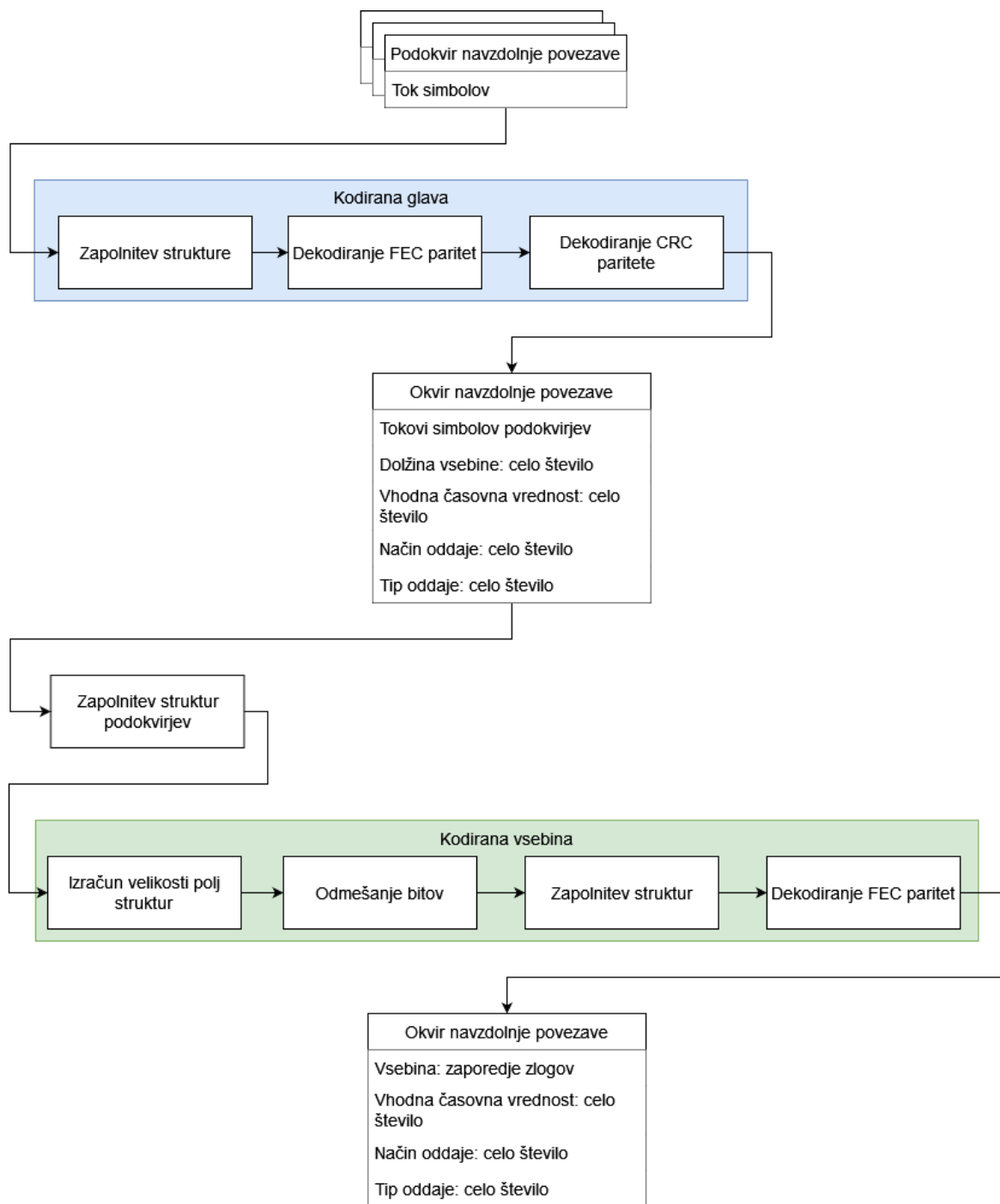
Za popoln sprejemnik bi bila potrebna dodatna programska koda, ki uporabi podatek o času prejema različnih podokvirjev, da jih združi v okvir. To pomeni tudi, da se morajo podokvirji v pomnilniku hraniti dlje časa, dokler niso prejeti vsi podokvirji okvirja, ali dokler ne poteče časovno okno sprejema okvirja. V tem projektu časovni parametri in časovne zahteve niso bili implementirani, zato je združevanje podokvirjev v okvirje v trenutni implementaciji potrebno početi z zunanjo kodo.

Razčlenjevanje tako kot generacija poteka v korakih, kar pa je v tem primeru nujno, saj je treba prvo dekodirati glavo, da se ugotovi velikost in oblika preostalega dela podokvirja. Metoda razčlenjevalnika `parseBurst` vrne podokvir z dekodiranimi parametri glave in neprocesirano kodirano vsebino. Podokvirje se nato zbere v objekt okvirja in pošlje v metodo `parseFrame`, ki v objekt okvirja zapiše dekodirano binarno vsebino.





Slika 29: Potek razčlenjevanja okvirja navzgornje povezave



Slika 30: Potek razčlenjevanja okvirja navzdolnje povezave

## 4 ANALIZA IN REZULTATI

Glavni cilj tega razdelka je bil preveriti, da je implementacija protokola OMS LPWAN ustrezna, in preučiti kako različne stopnje šuma vplivajo na možnost uspešnega sprejema podatkov na strani sprejemnika. Izbran je bil šum tipa AWGN, saj je ta v naravi vseprisoten in se prišteje k šumu iz ostalih virov. V realnem okolju je potrebno upoštevati tudi vire interference, kot so na primer večpotje, ter v urbanih okoljih, za katera je OMS LPWAN namenjen, ostale oddajnike na istem radijskem kanalu. Ker pa je AWGN najbolj univerzalen, predstavlja tak kanal uporabno referenco, in je v tem projektu bil edini upoštevan šum.

Pred simulacijo je bila implementacija preverjena tudi na testnih vektorjih priloženih v OMS LPWAN specifikaciji. Napisani testi preverjajo vhodna in izhodna bitna zaporedja v obe smeri preslikav, pri generaciji in razčlenjevanju pa tudi vmesna bitna zaporedja struktur pred različnimi kodiranj.

```
class SingleBurstRate78Test(unittest.TestCase):
    def test_generator(self):
        frame = DownlinkFrame()
        frame.coded_payload.phy_payload = self.vector.phy_payload.bytes
        frame.coded_header.timing_input_value = self.vector.timing_input_value.uint
        frame.coded_header.burst_mode = BURST_MODE_SINGLE_BURST
        frame.coded_header.burst_type = BURST_TYPE_DOWNLINK_SINGLE_BURST_FEC_RATE_7_8
        self.generator.generateFrame(frame)

        burst = frame.downlink_0
        coded_payload = frame.coded_payload
        coded_header = frame.coded_header

        self.assertEqual(coded_header.struct.version.getBits(), self.vector.version)
        self.assertEqual(coded_header.struct.phy_payload_length.getBits(), self.vector.phy_payload_length)
        self.assertEqual(coded_header.struct.timing_input_value.getBits(), self.vector.timing_input_value)
        self.assertEqual(coded_header.struct.burst_mode.getBits(), self.vector.burst_mode)
        self.assertEqual(coded_header.struct.burst_type.getBits(), self.vector.burst_type)

        self.assertEqual(coded_payload.single_burst_fec_7_8_struct.phy_payload.getBits(), self.vector.phy_payload)
        self.assertEqual(coded_payload.single_burst_fec_7_8_struct.getBits(), self.vector.coded_payload)

        self.assertEqual(burst.struct.preamble.getBits(), self.vector.preamble)
        self.assertEqual(burst.struct.syncword.getBits(), self.vector.sync)
        self.assertEqual(burst.struct.coded_header.getBits(), self.vector.coded_header)
        self.assertEqual(burst.struct.data.getBits(), self.vector.data)

        self.assertEqual(burst.struct.getBits(), self.vector.radio_burst)
```

Slika 31: Preverjanje s testnimi vektorji - primer za SingleBurstRate78Test

Za namene analize so bili celotni procesi generiranja, zaznave in razčlenjevanja razdeljeni v manjše funkcije, da so se lahko različni koraki testirali posebej, in da so testne funkcije bile manj računsko zahtevne. Da je lahko bilo zbranih več veljavnih meritev pri analizi FEC kodiranja, sta se zaznavanje podokvirjev in razčlenjevanje testirala ločeno. Pravzaprav so s simulacijo bili tako ločeno analizirani trije koraki razčlenjevanja toka vzorcev: iskanje (zaznavanje) podokvirjev, dekodiranje kodirane glave in dekodiranje kodirane vsebine. Razen kjer je navedeno drugače, je za vsako točko v grafu bilo simuliranih 5,000 testnih primerov. Parametri, ki v grafih niso navedeni (na primer vrednost vhodne časovne vrednosti, ali binarna vsebina), so bili za vsak testni primer naključno generirani.

Lažne zaznave so bile simulirane tako, da se je zaznavanje ali pa dekodiranje izvedlo na zaporedju vzorcev AWGN šuma brez prisotnosti kakršnegakoli signala. Uporabljena je bila amplituda šuma, ki bi pripadala vrednosti  $E_b/N_0 = 0$  dB pri ustrezni modulačijski tehniki (FSK ali MSK), a načeloma ta vrednost ni pomembna, saj je korelacija normalizirana.

#### 4.1 Simulacija prenosnega kanala

Za simulacijo frekvenčne modulacije (kakršna se uporablja pri navzdolnji povezavi) v osnovnem pasu, so se zaporedja bitov kodirala z NRZ kodiranjem, nato pa so bili prišteti realni vzorci AWGN šuma.

Pri simulaciji navzgorne povezave se je namesto GMSK modulacije uporabila MSK modulacija. Tako je v moduliranem signalu edini vir šuma AWGN kanal in ne tudi med-simbolna interferenca. Pri nadaljnjem testiranju bi MSK modulator in demodulator bilo potrebno zamenjati z GMSK različico.

Štiri simbole MSK se je predstavilo s kompleksnimi simboli, katerih realni in imaginarni del predstavljata sofazno in kvadraturno (angl. in-phase and quadrature oz. I/Q) komponento signala, tem pa so se dodali kompleksni vzorci AWGN šuma. Izbran postopek demodulacije vrača originalne simbole diferenčno kodirane MSK modulacije in zato ni bilo potrebno

posebej izvajati diferenčnega dekodiranja. Programsko kodo demodulatorja prikazuje slika (Slika 32).

```
class PrecodedMskDemodulator:
    def __init__(self, initial_phase, zero_clockwise):
        self.initial_phase = initial_phase
        self.phase = self.initial_phase
        if zero_clockwise:
            self.phase_shift_coefficient = 1j
        else:
            self.phase_shift_coefficient = -1j

    def clear(self):
        self.phase = self.initial_phase

    def demodulate(self, modulated):
        phase_offset = self.phase * self.phase_shift_coefficient
        phase = phase_offset * numpy.insert(numpy.cumprod(numpy.full(len(modulated) - 1, -self.phase_shift_coefficient)), 0, 1)
        nrz = numpy.real(modulated / phase)
        self.phase = phase[-1]
        return nrz
```

Slika 32: Implementacija dekodirnika diferenčno kodirane modulacije z minimalnim skokom

Iskanje sinhronizacijskih besed s križno korelacijo se izvaja neposredno na moduliranih simbolih, in se zato pri MSK pričakuje višja verjetnost zaznave zaradi dodatne fazne informacije.

Šumni signal AWGN kanala se je simuliral z generacijo naključnih otipkov po normalni distribuciji z uporabo funkcije `numpy.random.normal`, ki zahteva dva parametra: sredinsko vrednost in standardni odklon (angl. standard deviation) vrednosti. Slika (Slika 33) podaja uporabljeno enačbo za določitev ustreznega odklona za izbrano vrednost  $E_b/N_0$ . Za simulacijo kompleksnega šuma sta se posebej generirali dve naključni zaporedji za realne in imaginarne komponente, vsaka s polovično deviacijo [13].

$$dev = \sqrt{\frac{1}{2(E_b/N_{0_{linear}})}}$$

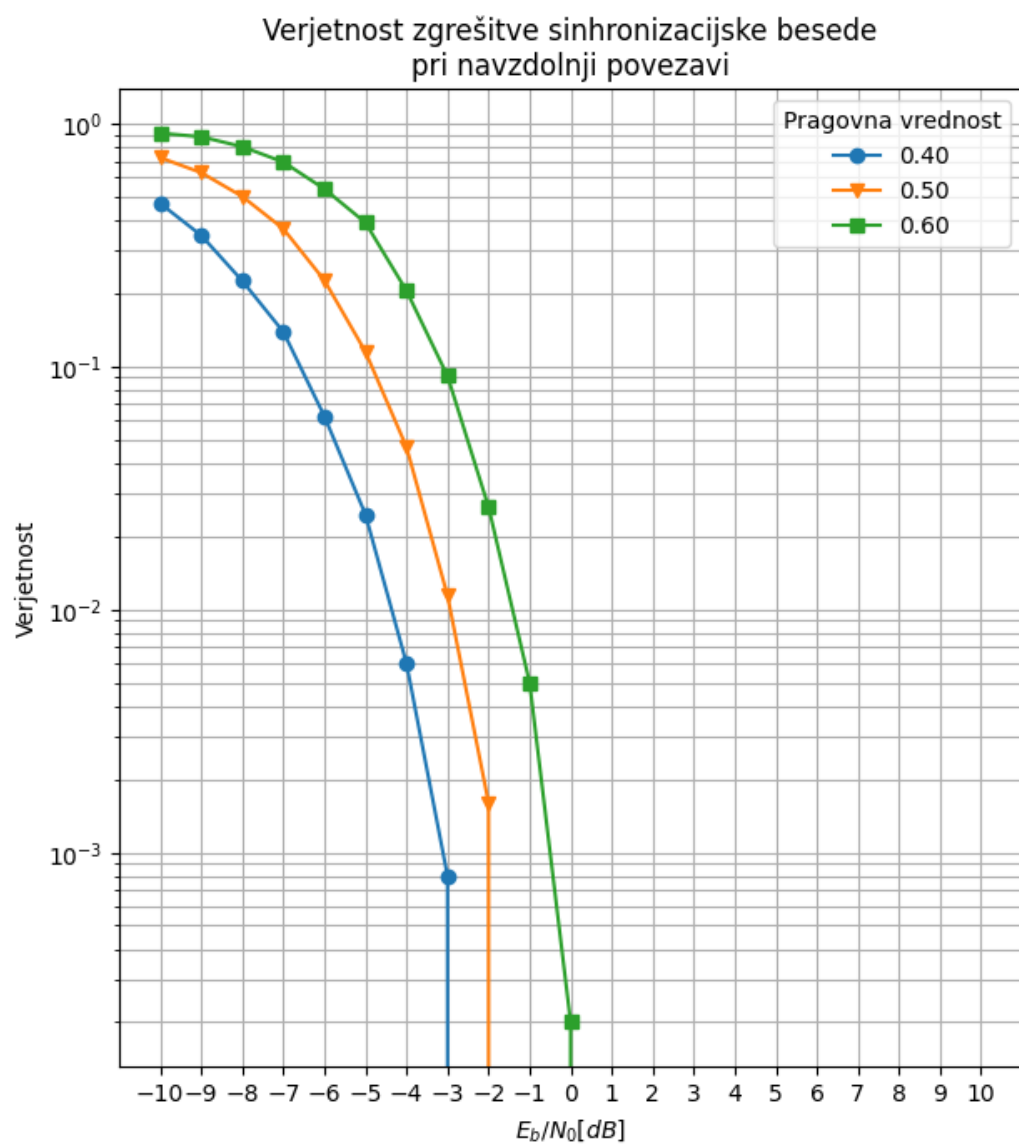
Slika 33: Enačba za deviacijo AWGN šuma [14]

## 4.2 Zaznavanje podokvirjev

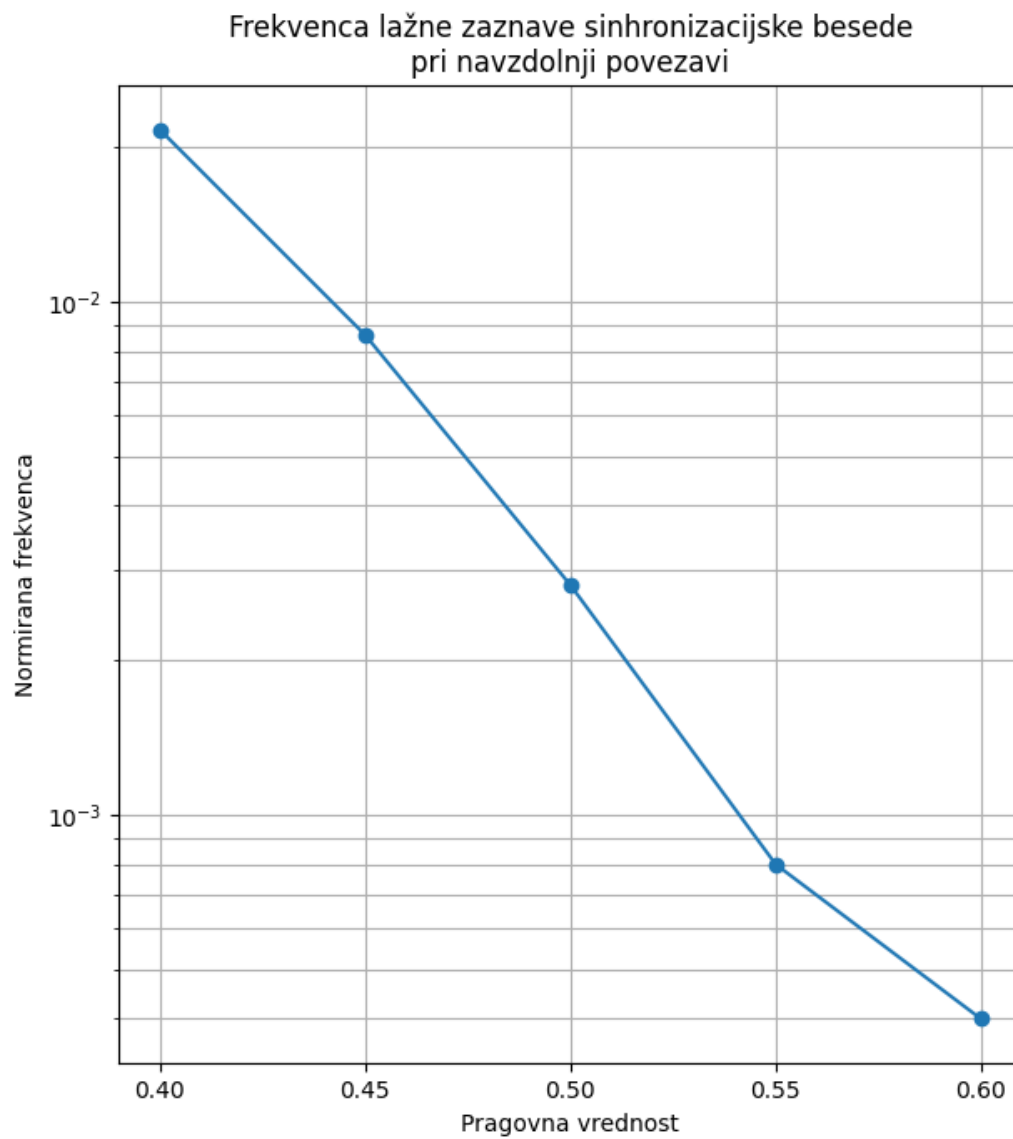
### 4.2.1 Navzdolnja povezava

Edini uporabljen mehanizem za zaznavanje podokvirjev v zaporedju otpikov v navzdolnji povezavi je korelacijsko zaznavanje sinhronizacijske besede. Pogoji za uspešno zaznan podokvir je torej, da je vrednost normalizirane korelacije v iskalniku sinhronizacijske besede višja od pragovne vrednosti. Na grafu (Slika 34) je prikazana verjetnost, da pri neki vrednosti  $E_b/N_0$  ta pogoj ni izpolnjen in je podokvir izgubljen. Vidimo, da je pri višjih vrednostih pragovne vrednosti zaznavanje manj občutljivo, saj je pri enakih vrednostih  $E_b/N_0$  višja verjetnost, da je podokvir zgrešen. Nad 0 dB je bil pri testiranih pragovnih vrednostih zaznan vsak od 5,000 podokvirjev.

Na grafu (Slika 35) je prikazana normirana frekvenca (med 0 in 1, kjer 1 predstavlja frekvenco vzorčenja v osnovnem pasu) lažne zaznave sinhronizacijske besede za različne pragovne vrednosti. Iz tega grafa je razvidno, da nižje pragovne vrednosti privedejo do večjega števila lažnih zaznav. Pri pragovni vrednosti 0.4 pride do lažne zaznave v povprečju vsakih 50 vzorcev, pri vrednosti 0.6 pa vsake 2,500 vzorcev.



Slika 34: Verjetnost zgrešitve sinhronizacijske besede pri navzdolnji povezavi



Slika 35: Frekvenca lažne zaznave sinhronizacijske besede pri navzdolnji povezavi

#### 4.2.2 Navzgornja povezava

Pri navzgornji povezavi sta uporabljena dva mehanizma za zaznavanje podokvirjev: zaznavanje sinhronizacijske besede ter zaznavanje sredinske sinhronizacijske besede.

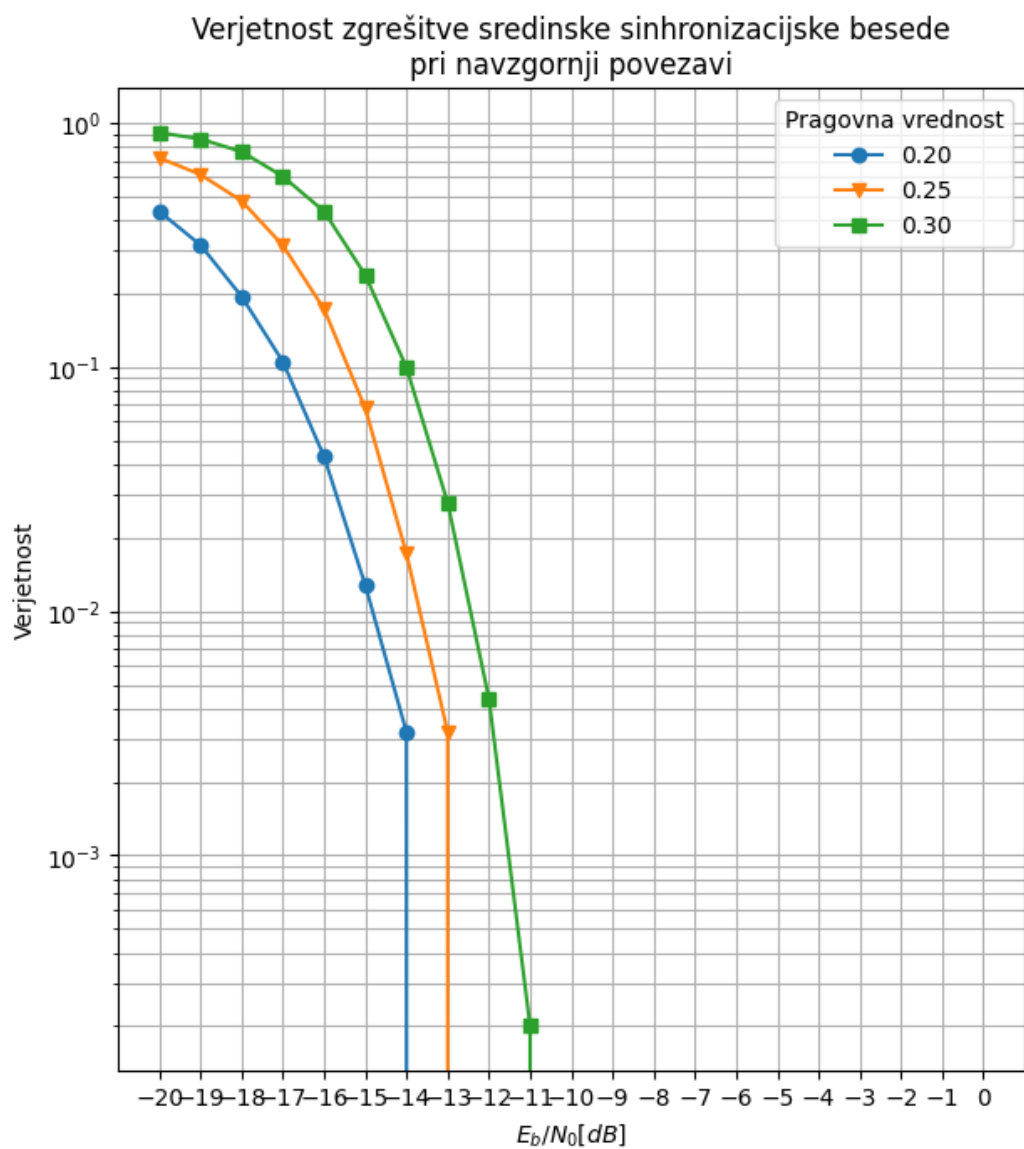


Zaznavanje sredinske sinhronizacijske besede poteka tako kot pri zaznavanju sinhronizacijske besede navzdolnje povezave zgolj s križno korelacijo, in je pogoj za zaznan podokvir enak. Iz grafov (Slika 36) in (Slika 37) pa lahko razberemo, da je občutljivost v tem scenariju veliko višja pri nižjih pragovnih vrednostih (tokrat med 0.2 in 0.3) in s podobno frekvenco lažnih zaznav. To je za pričakovati, saj je sredinska sinhronizacijska beseda veliko daljša od tiste, ki se nahaja na začetku. Nad  $-11 \text{ dB } E_b/N_0$  ni bil zgrešen noben izmed 5,000 podokvirjev pri nobeni izmed testiranih pragovnih vrednosti.

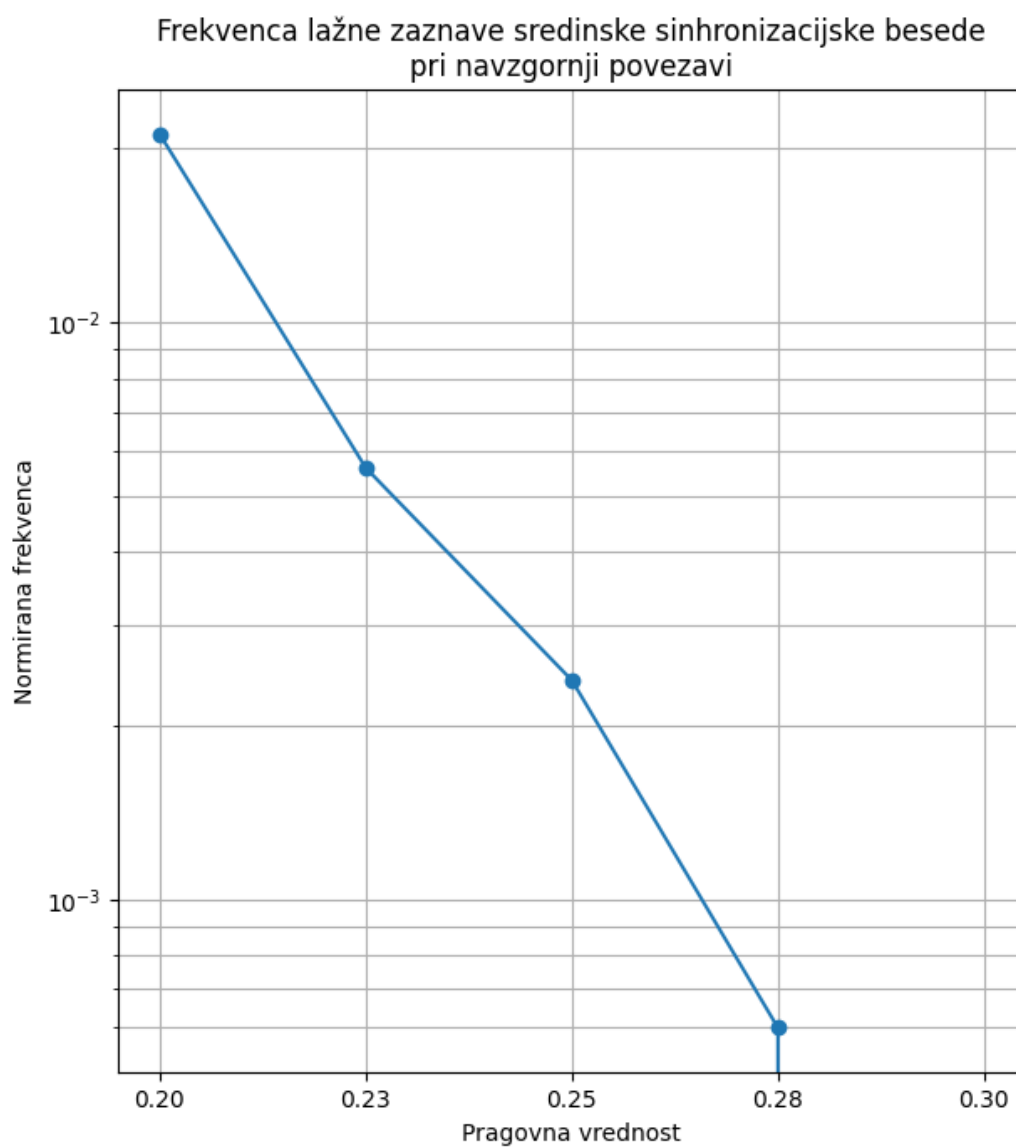
Drugi mehanizem za zaznavo okvirjev pri navzgornji povezavi je podobno kot pri navzdolnji povezavi zaznavanje začetne sinhronizacijske besede. Sinhronizacijski besedi pa sledi kodirana dolžina, ki jo mora sprejemnik potrebno pravilno dekodirati, da najde kodirano glavo. Pogoj za zaznavo je tako poleg dosežene pragovne vrednosti še pravilno dekodirana kodirana dolžina. Dovoljene vrednosti dolžine so tudi navzdol in navzgor omejene, zato kodirana dolžina pripomore tudi k zavračanju določenega deleža lažnih zaznav. Vredno pa je omeniti, da omejitve kodirane dolžine niso izrecno podane v standardu, čeprav jih je moč izračunati iz ostalih omejitev.

Iz grafa (Slika 38) vidimo, da je zaznavanje spet manj občutljivo, a zaradi MSK modulacije vseeno bolj kot pri navzdolnji povezavi, razlike med občutljivostmi različnih pragov pa so veliko manjše. Vsak od 5,000 podokvirjev je uspešno zaznan nad  $E_b/N_0$  vrednosti  $-5 \text{ dB}$ . Pri vrednostih  $E_b/N_0$  višjih od  $4e-3$  pride do obrnitve občutljivosti različnih pragovnih vrednosti, kar je verjetno posledica omejene natančnosti simulacije pri majhnih vrednostih.

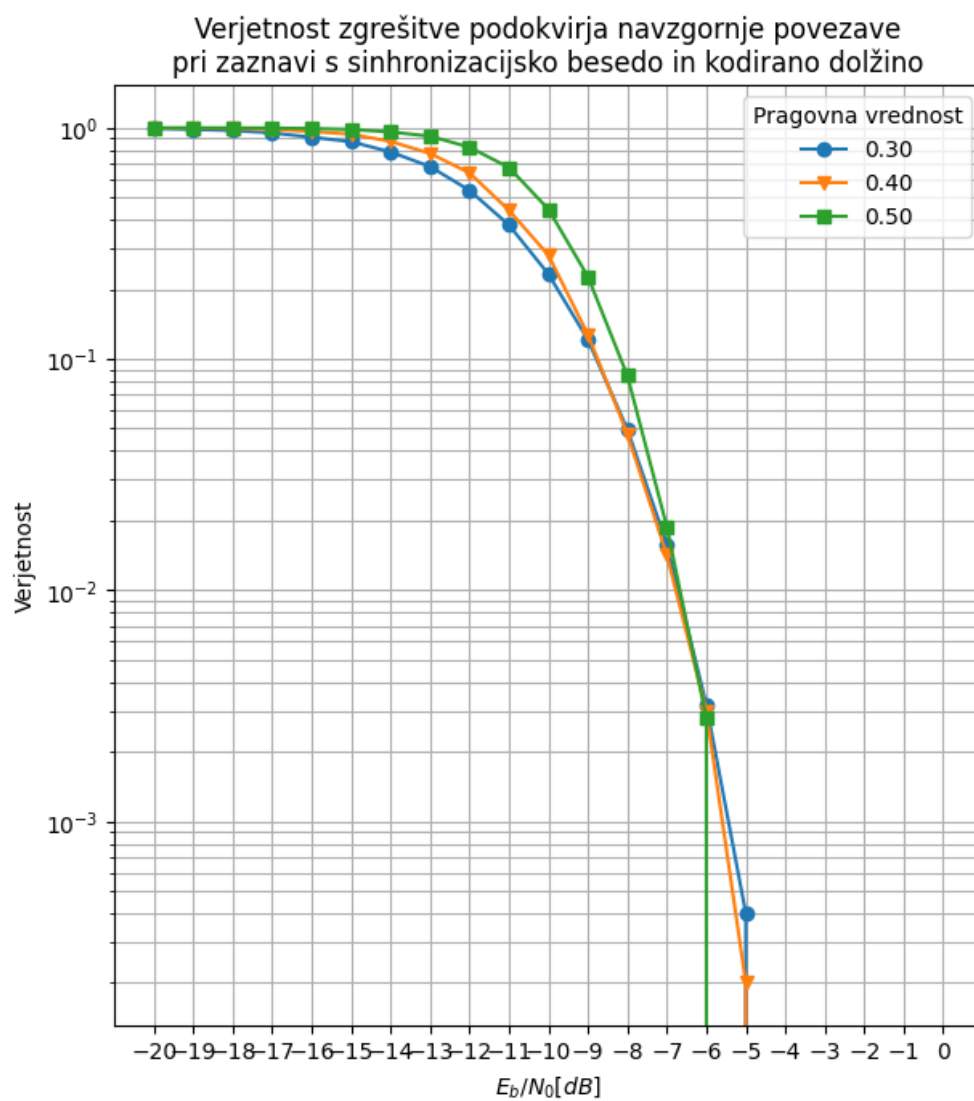
Graf (Slika 39) kaže, da imajo izbrane pragovne vrednosti nekoliko slabšo karakteristiko zavračanja lažnih podokvirjev. Izbrane bi lahko bile višje pragovne vrednosti za ceno slabšega uspeha pri zaznavanju. Vseeno pa so vrednosti dovolj podobne, da lahko napravimo primerjavo med pragovno vrednostjo 0.25 pri zaznavi sredinske sinhronizacijske besede in vrednostjo 0.4 pri zaznavi začetne sinhronizacijske besede navzgornje povezave. Zaznava z začetno sinhronizacijsko besedo je manj občutljiva za približno 7 dB.



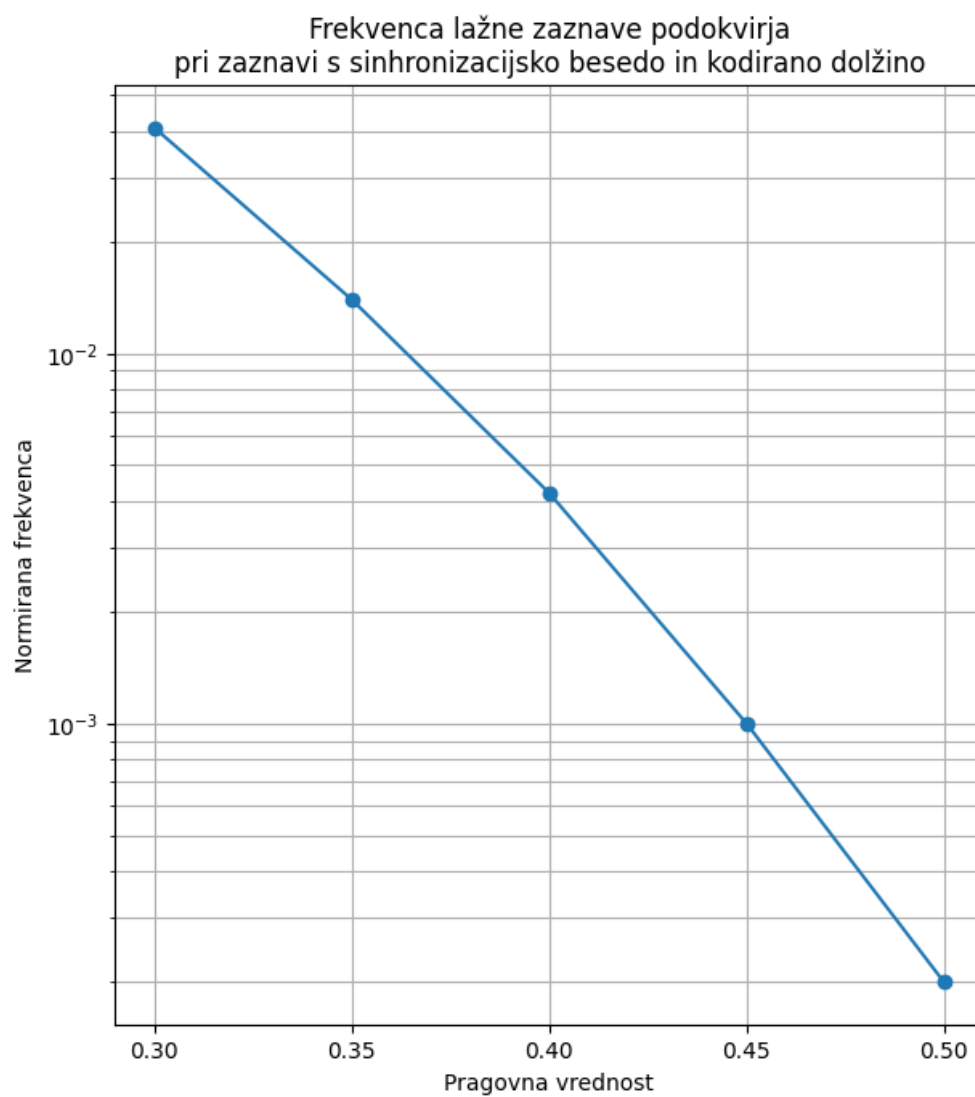
Slika 36: Verjetnost zgrešitve sredinske sinhronizacijske besede pri navzgornji povezavi



Slika 37: Frekvenca lažne zaznave sredinske sinhronizacijske besede pri navzgornji povezavi



Slika 38: Verjetnost zgrešitve podokvirja navzgornje povezave pri zaznavi s sinhronizacijsko besedo in kodirano dolžino



Slika 39: Frekvenca lažne zaznave podokvirja navzgorne povezave pri zaznavi s sinhronizacijsko besedo in kodirano dolžino

### 4.3 Dekodiranje glave

Ko je zaznan podokvir, se dekodira glava z vsebovanimi FEC in CRC paritetami. Če je glava pravilno dekodirana, pride v poštev še FEC kodiranje vsebine. Če je glava dekodirana napačno, potem bo premešana kodirana vsebina praktično naključna in zavržena v podatkovnem sloju. Upoštevati pa je potrebno tudi verjetnost lažno veljavne kodirane glave, ki poleg verjetnosti lažne zaznave podokvirja določa pogostost lažnih podokvirjev, ki jih sprejemnik v celoti dekodira, kar je slabo za energijsko učinkovitost, na šibkejših mikrokrmilnikih pa lahko privede tudi do izgube dejanskih podokvirjev.

Verjetnost lažno veljavne glave je bila analizirana s 100,000 testi in zaokrožena celo število procentov. Verjetnosti iz tabele (Tabela 4) kažejo, da pri navzgornji povezavi manjša možnost za zavrnitev lažne glave. To do neke mere znotraj pričakovanj, saj ima glava navzdolnje povezave več rezerviranih (neveljavnih) vrednosti parametrov.

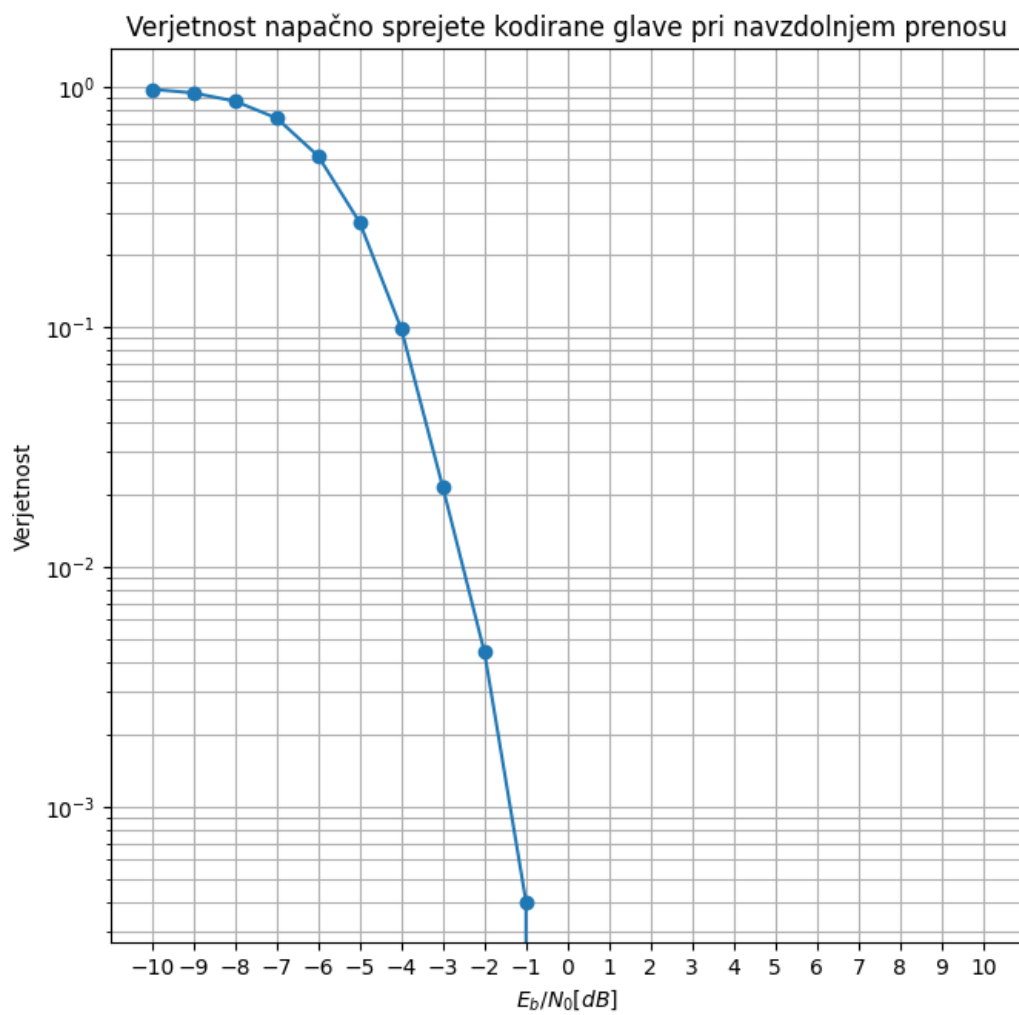
Razberemo lahko, da bo pri navzdolnji povezavi bil sprejet vsak osmi ali deveti lažni podokvir, pri navzgornji pa vsak peti ali šesti. Kodirana glava torej nima velikega vpliva pri zavračanju lažnih podokvirjev. To bi se lahko izboljšalo tako, da se bi CRC namesto za odpravljanje napak uporabil za detekcijo napak, a se bi s tem verjetno zmanjšalo tudi število pravilno dekodiranih glav.

Tabela 4: Verjetnost lažno veljavne glave

Smer povezave	Verjetnost
Navzdolnja	12 %
Navzgornja	18 %

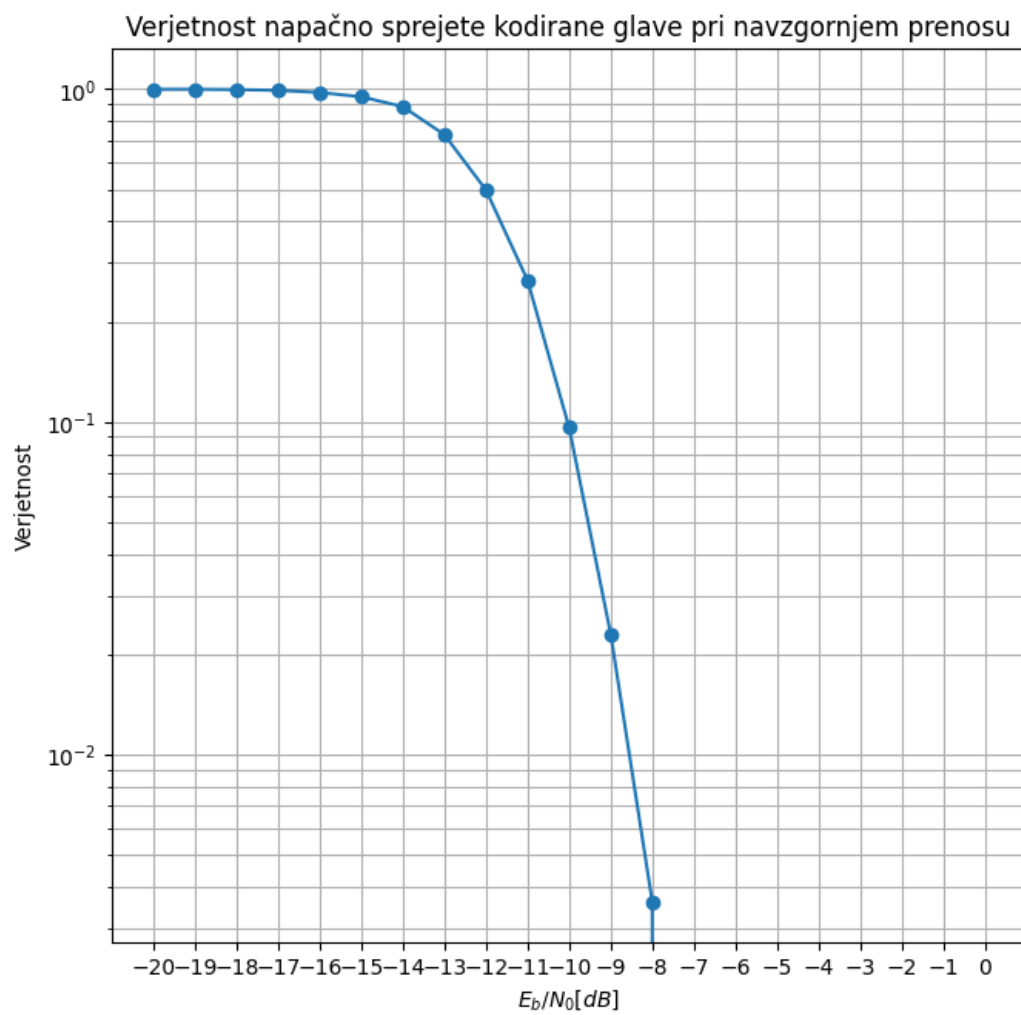
Na grafih (Slika 40) in (Slika 41) lahko vidimo uspešnost FEC dekodiranja glave predstavljeno z verjetnostjo napačnega sprejema v odvisnosti od  $E_b/N_0$ . Pri tem je glava uspešno dekodirana le takrat, ko se izhodni parametri povsem ujemajo z vhodnimi.

Pri navzdolnjem prenosu je pri -2.5 dB napačen 1 % prejetih glav, pri  $E_b/N_0$  višji od -1 dB pa je bila pravilno dekodirana vsaka od 5,000 glav. Pri navzgornjem prenosu je napačen 1 % prejetih glav pri -8.5 dB, od -8 dB višje pa so bile vse glave pravilno dekodirane. Tukaj lahko ocenimo odstopanje zmogljivosti FEC kodiranja med navzgornjim in navzdolnjim prenosom približno 6 dB. Ta razlika ne more biti rezultat kodiranja, saj je v obeh primerih uporabljen enak FEC kodirnik. Ta razlika je verjetno posledica I/Q predstavitve simbolov, saj je bilo glede na [13] deviacijo potrebno pri generiranju kompleksnega šuma razpoloviti.



Slika 40: Verjetnost napačno sprejete kodirane glave pri navzdolnjem prenosu





Slika 41: Verjetnost napačno sprejete kodirane glave pri navzgorjem prenosu

## 4.4 Dekodiranje vsebine

Pri analizi FEC kodiranja vsebine je bilo izvedeno dekodiranje neposredno na šumnem signalu okvirja, brez izvajanja detekcije. Neveljavnih vrednosti vsebine v fizičnem sloju ni, zato se je upošteval vsak dekodiran okvir. Izmerjeno je bila normirana frekvenca napačno prejetih bitov oz. pogostost napak (angl. bit-error rate oz. BER). Tudi najmanjša količina napake okvari sporočilo, zato je želja doseči dovolj nizko vrednost BER, da je večji del sporočil popolnoma brez napake. Najdaljša možna dolžina vsebine je 255 zlogov oziroma 2,040 bitov, kar pomeni, da je pri BER  $2.45 \times 10^{-4}$  v povprečju okvarjen vsak drugi takšen okvir. Ekvivalentna vrednost za vsebino najkrajše dolžine (5 zlogov) je  $1.25 \times 10^{-2}$ . Pri vsakem profilu in načinu sta izrisana grafa tako za 255-zložno vsebino kot za 5-zložno vsebino.

V grafih načina enojne oddaje so izrisane krivulje za vse tri možne stopnje kodiranja: 7/8, 1/2 in 1/3. V grafih načina večkratne oddaje različne krivulje predstavljajo različna števila prejetih podokvirjev okvirja. To so podokvirji, ki jih sprejemnik uspešno zaznal, in katerih glavo je potem pravilno dekodiral. Možna števila prejetih podokvirjev so 1, 2 in 3 katerim ustrezajo efektivne stopnje kodiranja 7/8, 7/16 in 7/24.

### 4.4.1 Navzdolnja povezava

V grafih (Slika 42) in (Slika 43) vidimo pri BER  $1 \times 10^{-3}$  med dvema dolžinama vsebine za približno 1 dB boljšo korekcijo pri stopnji 1/3 in za približno 0.5 dB boljšo korekcijo pri stopnji 7/8. Tovrstna razlika se kaže v vseh grafih tega razdelka in je za pričakovati, ko manjši delež kodirane vsebine sestavlja dejanska vsebina, saj vsebuje kodirana vsebina polnila, ki so lahko okvarjena brez posledice pri dekodiranju, prav tako pa je količina repnih bitov, ki predstavljajo dodatno redundanco, vedno enaka neodvisno od dolžine vsebine, in zato v primeru krajše vsebine predstavlja večji delež kodirane vsebine.

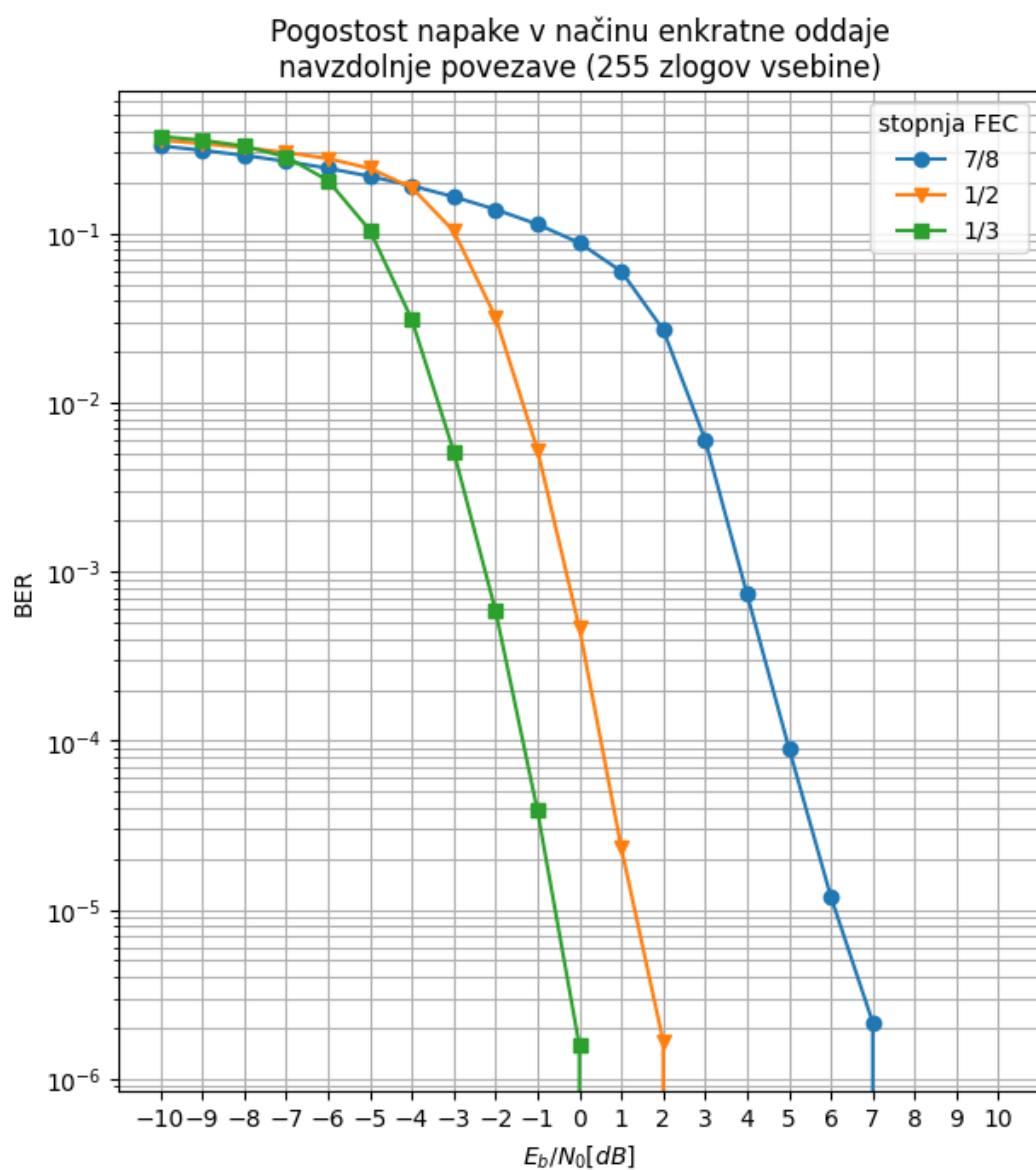
Vidina je očitna izboljšava v zmogljivosti korekcije z manjšanjem stopnje kodirnika, kar je pričakovano. Pri BER  $1e-4$  ima stopnja 1/2 za 4.5 dB boljšo zmogljivost, stopnja 1/3 pa za približno 6.5 dB. BER nižjih stopenj do določene moči šuma upadajo počasneje, nato pa bolj strmo, zato so razlike v zmogljivosti pri višjih vrednosti  $E_b/N_0$  večje.

Na grafu (Slika 44) so pri BER  $1e-4$  podobne razlike med stopnjami. Stopnja 1/3 je za približno 0.5 dB manj zmogljiva od stopnje 7/24 3 prejetih podokvirjev večkratne oddaje. Podobna razlika 0.5 dB velja med stopnjama 1/2 in 7/16. BER krivulja enega prejetega podokvirja večkratne oddaje ima v primerjavi s podokvirjem enkratne oddaje kodiranja stopnje 7/8 zakasnen padec. To gre verjetno pripisati različnim karakteristikam generatorskih polinomom, ki so v kombinaciji s polinomom povratne vezave uporabljeni v podokvirjih večkratne oddaje, saj krivulja kaže njihov povprečni BER.

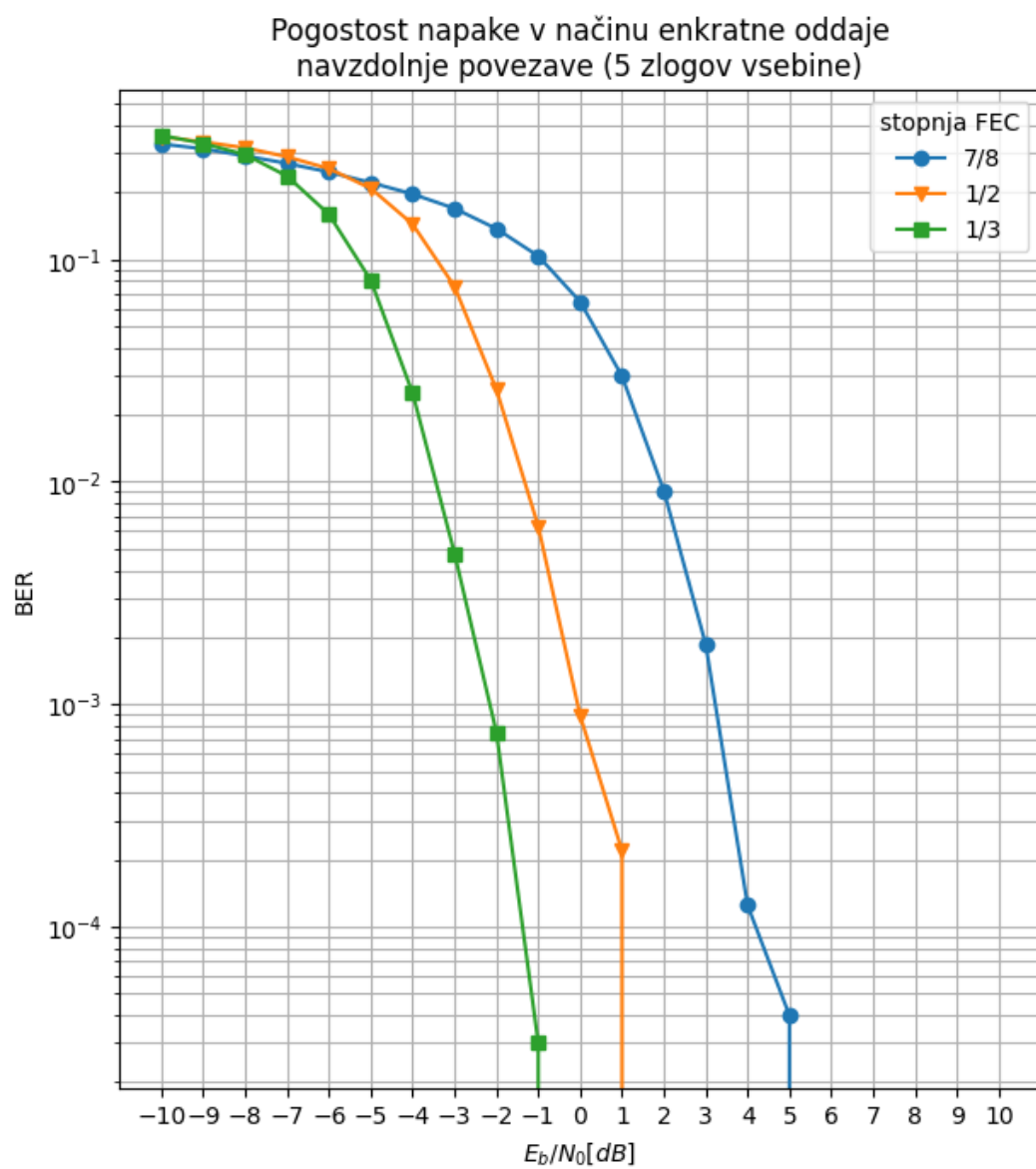
Tabela (Tabela 5) uporabi omenjeno mejo BER ( $2.45e-4$ ) da navede mejne vrednosti  $E_b/N_0$  za pravilno dekodiranje vsaj vsakega drugega okvirja z 255-zložno vsebino.

Tabela 5: Mejne vrednosti razmerja energija bita/spektralna gostota šuma za pravilno dekodiranje vsakega drugega okvirja pri navzdolnji povezavi z 255-zložno vsebino

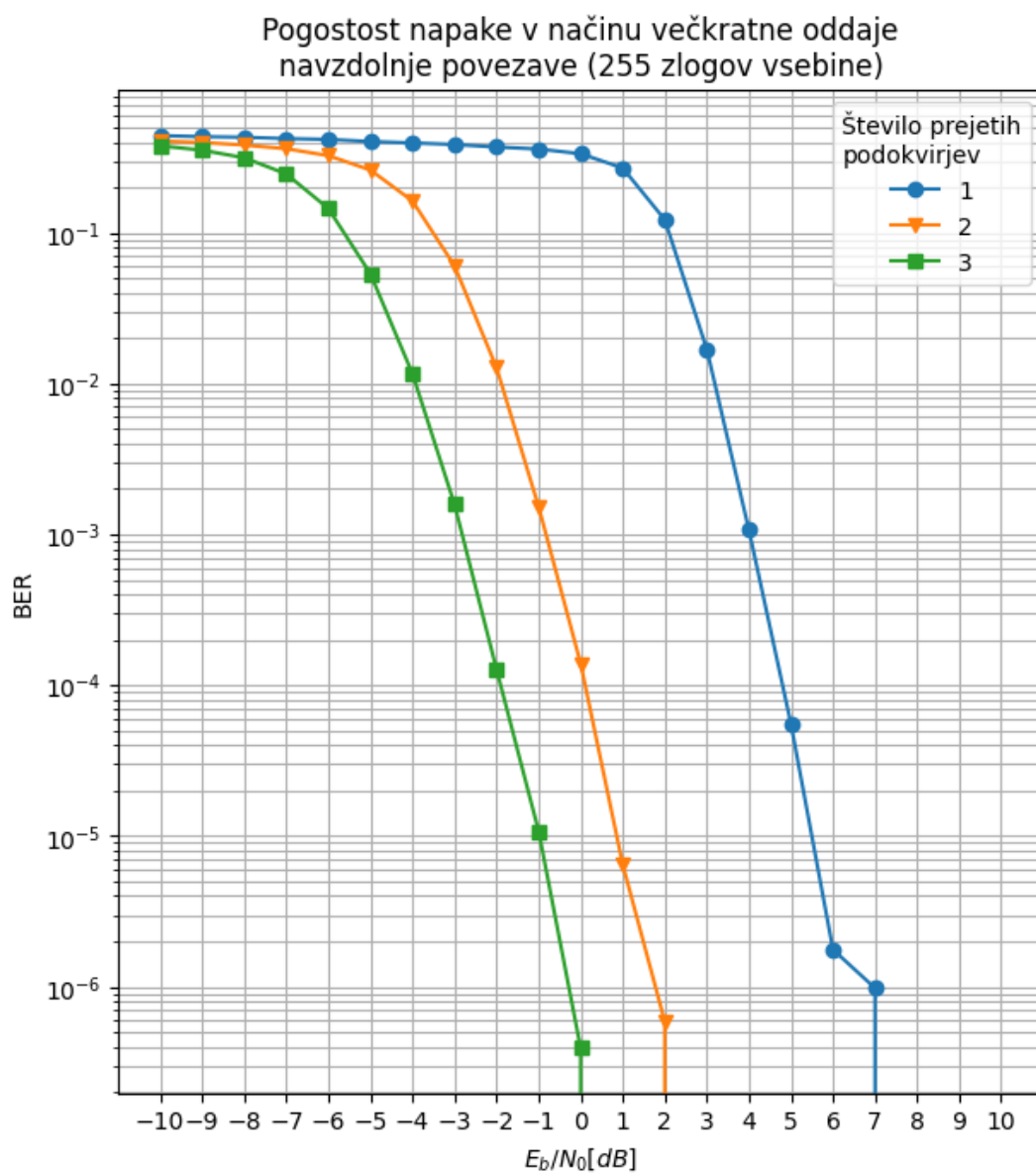
Scenarij	Mejna vrednost $E_b/N_0$ za pravilno dekodiranje vsakega drugega okvirja v dB
Enkratna oddaja, stopnja 7/8	4.4
Enkratna oddaja, stopnja 1/2	1
Enkratna oddaja, stopnja 1/3	-1.8
Večkratna oddaja, 1 prejet podokvir	4.4
Večkratna oddaja, 2 prejeta podokvirja	-0.4
Večkratna oddaja, 3 prejeti podokvirji	-2.4



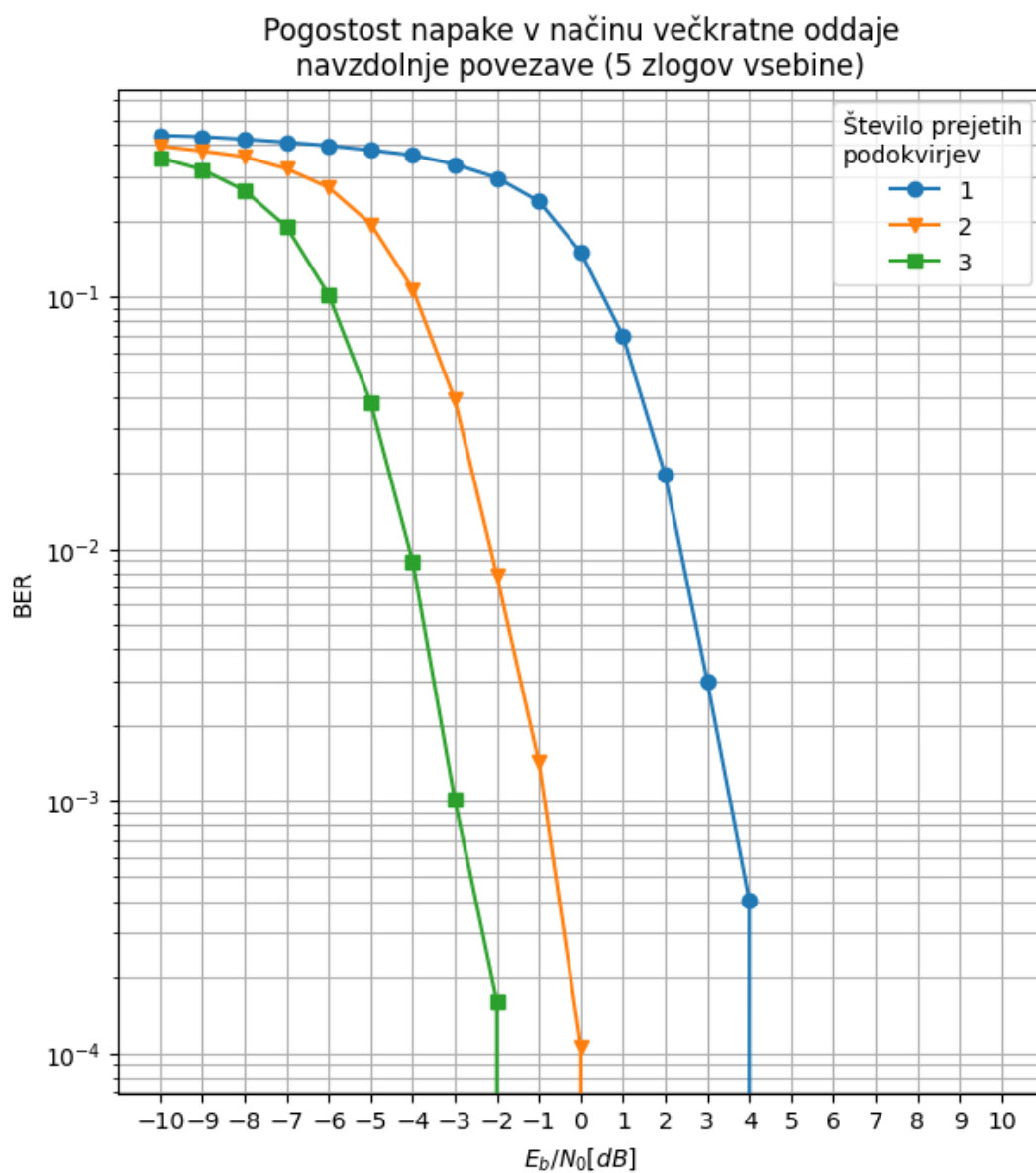
Slika 42: Pogostost napake v načinu enkratne oddaje navzdolnje povezave z 255-zložno vsebino



Slika 43: Pogostost napake v načinu enkratne oddaje navzdolnje povezave s 5-zložno vsebino



Slika 44: Pogostost napake v načinu večkratne oddaje navzdolnje povezave z 255-zložno vsebino



Slika 45: Pogostost napake v načinu večkratne oddaje navzdolnje povezave s 5-zložno vsebino

#### 4.4.2 Navzgornja povezava

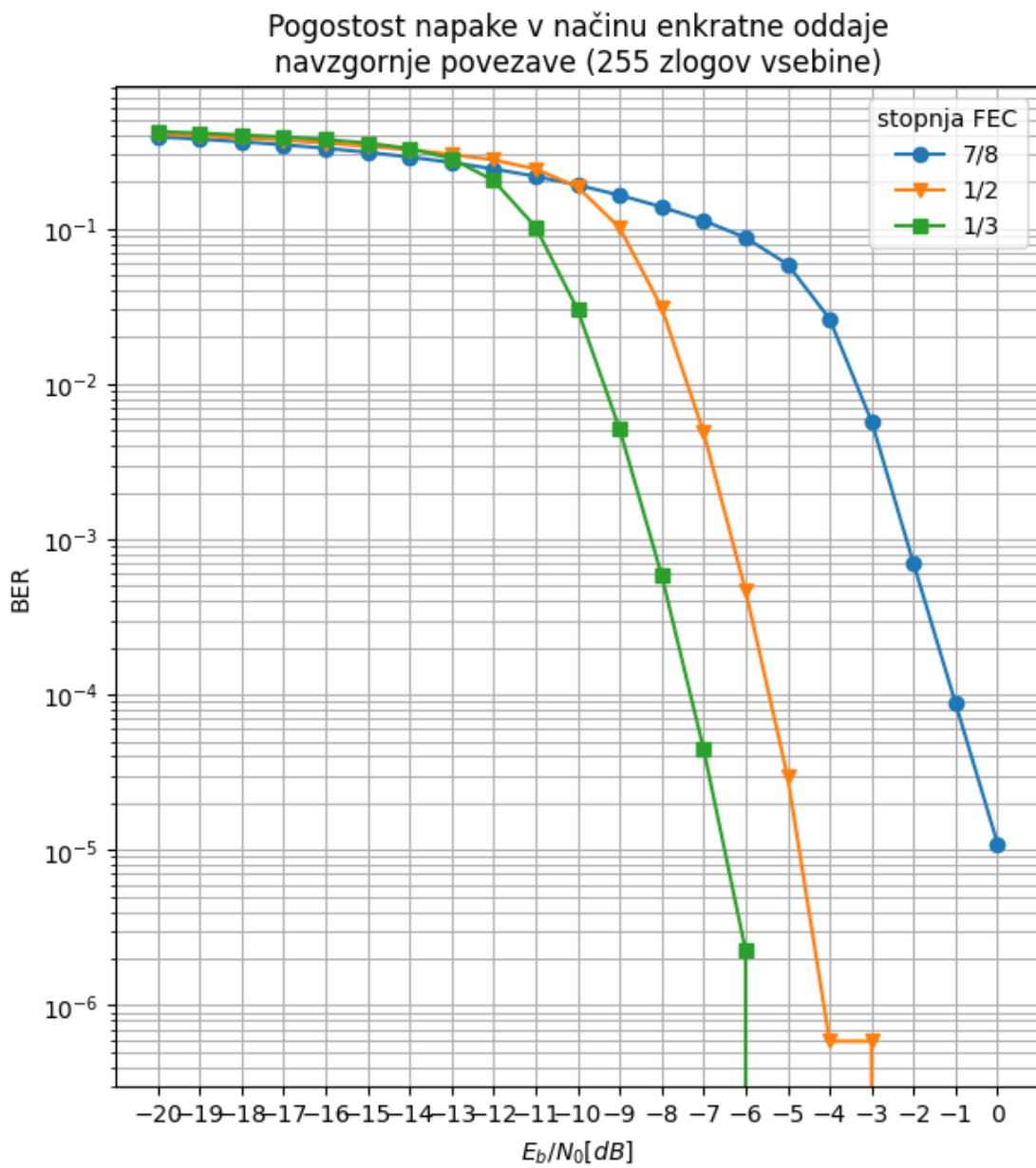
Kot pričakovano kažejo BER krivulje pri navzgornji povezavi enake karakteristike kot pri navzdolnji povezavi, saj je uporabljeno enako kodiranje. Edina opazna razlika je za 6 dB boljša zmogljivost pri navzgornji povezavi, kar se ujema z razliko opaženo pri dekodiranju glave.

Tako kot za navzdolnjo povezavo, je za navzgornjo povezavo podana tabela (Tabela 6), ki navaja mejne vrednosti  $E_b/N_0$  za pravilno dekodiranje vsaj vsakega drugega okvirja z 255-zložno vsebino.

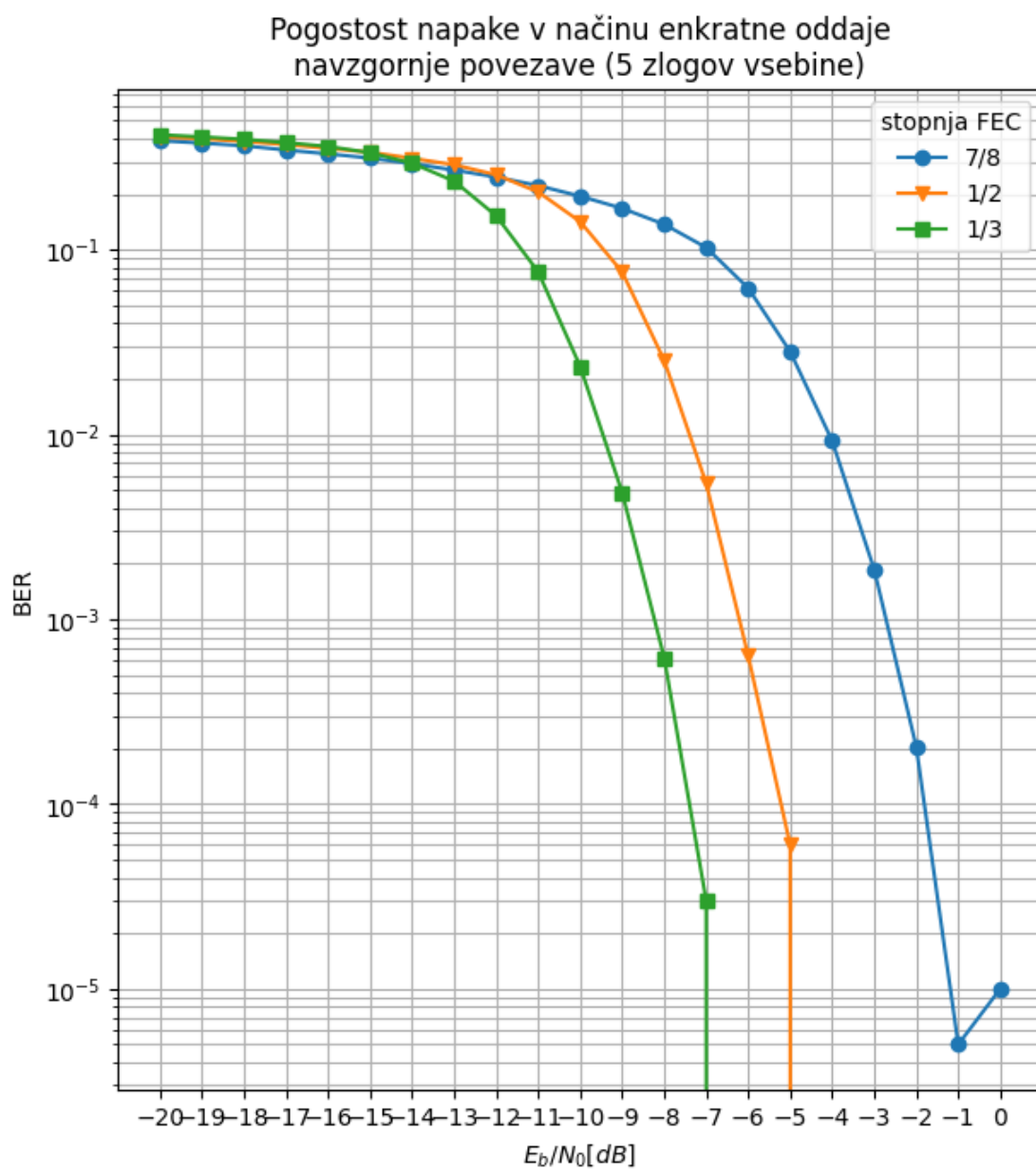
Tabela 6: Mejne vrednosti razmerja energija bita/spektralna gostota šuma za pravilno dekodiranje vsakega drugega okvirja pri navzgornji povezavi z 255-zložno vsebino

Scenarij	Mejna vrednost $E_b/N_0$ za pravilno dekodiranje vsakega drugega okvirja v dB
Enkratna oddaja, stopnja 7/8	-1.6
Enkratna oddaja, stopnja 1/2	-6
Enkratna oddaja, stopnja 1/3	-7.8
Večkratna oddaja, 1 prejet podokvir	-1.6
Večkratna oddaja, 2 prejeta podokvirja	-6.4
Večkratna oddaja, 3 prejeti podokvirji	-8.4

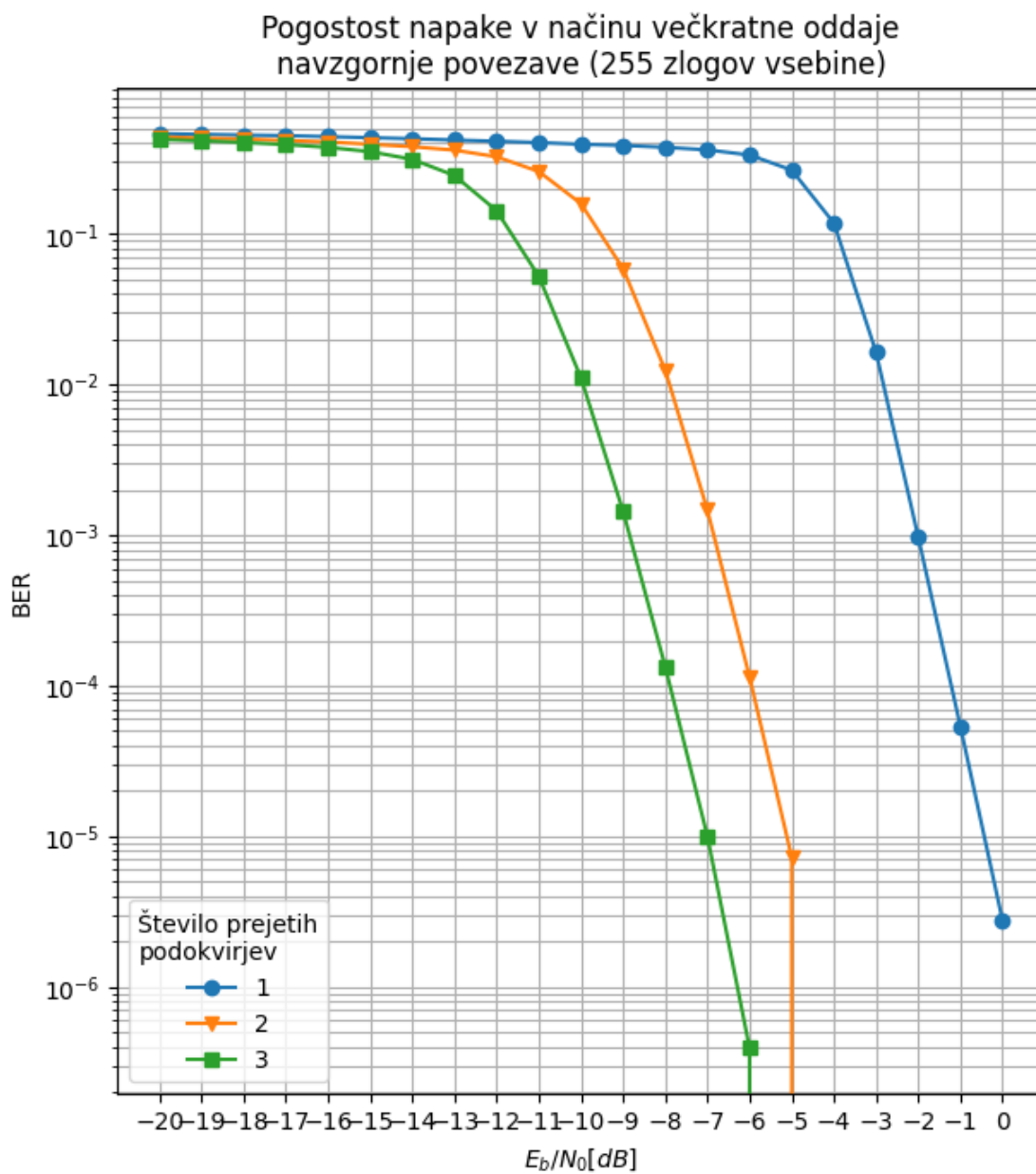




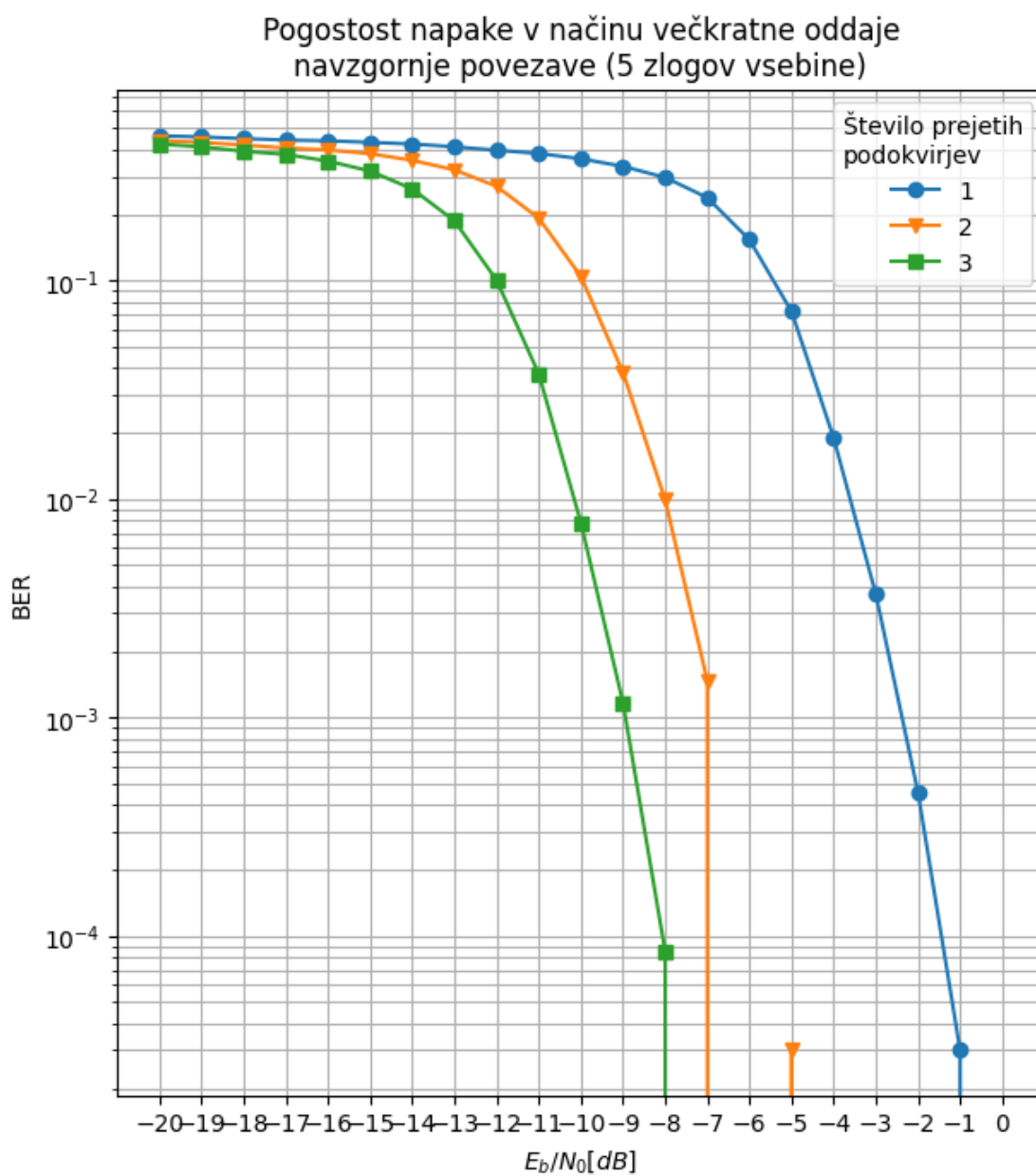
Slika 46: Pogostost napake v načinu enkratne oddaje navzgorne povezave z 255-zložno vsebino



Slika 47: Pogostost napake v načinu enkratne oddaje navzgorne povezave s 5-zložno vsebino



Slika 48: Pogostost napake v načinu večkratne oddaje navzgornje povezave z 255-zložno vsebino



Slika 49: Pogostost napake v načinu večkratne oddaje navzgorne povezave s 5-zložno vsebino

## 5 SKLEP

Implementirani so bili vsi podprofili niznega profila OMS LPWAN. Generator in razčlenjevalnik sta prestala vse testne vektorje iz OMS LPWAN specifikacije. Kodirnik in pomožni razredi so poleg testnih vektorjev prestali tudi mnogo naključnih preizkusov tekom simulacije.

Hitrost simulacij, ki so trajale od nekaj sekund do nekaj minut, je bila primerna za hitro testiranje različnih parametrov. Uspešno je bila opravljena simulacija vsakega implementiranega koraka generacije in razčlenjevanja. Iskalnik sinhronizacijskih besed je uspešno našel podokvirje v šumnih pogojih, z zmerno količino lažnih zaznav. Razlike med BER krivuljami ustrezajo pričakovanjem, krivulje pa v vseh primerih kažejo uspešno korekcijo napak. Za višjo stopnjo zaupanja v rezultate FEC bi bilo potrebno raziskati alternativne metode verifikacije.

Programski vmesnik je enostaven, zasnovane podatkovne strukture pa so tekom razvoja, testiranja in simulacije omogočale ustrezno stopnjo dostopa do različnih predstavitev podatkov. Vmesnik je primeren za prihodnjo integracijo z zunanjo programsko strojno opremo.

Nadaljnji razvoj je možen pri implementaciji časovnega razporejevalnika oddajnika in časovno-odvisnih funkcij sprejemnika. Možna je tudi implementacija podatkovnega sloja za bolj celovito testiranje prenosnega kanala. Z integracijo z zunanjo programsko opremo se lahko prenosne lastnosti protokola testirajo v za to namenskih okoljih. Z integracijo z SDR je možno generator in razčlenjevalnik uporabiti kot referenco za testiranje ostalih implementacij OMS LPWAN protokola.

## 6 VIRI IN LITERATURA

- [1] Chaudhari, Q. "Additive White Gaussian Noise (AWGN)" [Elektronski]. Available: <https://wirelesspi.com/additive-white-gaussian-noise-awgn/>. [Poskus dostopa 19. 9. 2024].
- [2] The Wikimedia Foundation, Inc. "Convolutional code" [Elektronski]. Available: [https://en.wikipedia.org/wiki/Convolutional\\_code](https://en.wikipedia.org/wiki/Convolutional_code). [Poskus dostopa 19. 9. 2024].
- [3] Proakis, J.G. and Salehi, M. (2001). *Digital Communications*.
- [4] The Wikimedia Foundation, Inc. "Additive White Gaussian Noise (AWGN)" [Elektronski]. Available: [https://commons.wikimedia.org/wiki/File:Convolutional\\_encoder\\_non-recursive.png](https://commons.wikimedia.org/wiki/File:Convolutional_encoder_non-recursive.png). [Poskus dostopa 19. 9. 2024].
- [5] Massachusetts Institute of Technology. "Viterbi Decoding of Convolutional Codes" [Elektronski]. Available: <https://web.mit.edu/6.02/www/f2011/handouts/8.pdf>. [Poskus dostopa 19. 9. 2024].
- [6] Balta, Horia & Trifina, Lucian & Rusinaru, Anca. (1997). THE EFFECT OF PUNCTURING ON THE CONVOLUTIONAL TURBO-CODES PERFORMANCES.
- [7] Peterson, L.L. & Davie, B.S. (2021). *Computer Networks: A Systems Approach*.
- [8] Chaudhari, Q. "Minimum Shift Keying (MSK) - A Tutorial" [Elektronski]. Available: <https://www.dsprelated.com/showarticle/1016.php>. [Poskus dostopa 19. 9. 2024].

- [9] The Wikimedia Foundation, Inc. "Additive White Gaussian Noise (AWGN)" [Elektronski]. Available: [https://en.wikipedia.org/wiki/Minimum-shift\\_keying](https://en.wikipedia.org/wiki/Minimum-shift_keying). [Poskus dostopa 19. 9. 2024].
- [10] Kačič, Zdravko. (1994). Digitalno procesiranje signalov.
- [11] Mohan, V. "An Introduction to Wireless M-Bus" [Elektronski]. Available: <https://pages.silabs.com/rs/634-SLU-379/images/introduction-to-wireless-mbus.pdf>. [Poskus dostopa 19. 9. 2024].
- [12] OMS-Group e. V. "Open Metering System Specification Vol.2 Annex Q" [Elektronski]. Available: [https://oms-group.org/fileadmin/files/download4all/omsSpezifikationen/Generation5/OMS-Spec\\_Vol2\\_AnnexQ\\_A101.pdf](https://oms-group.org/fileadmin/files/download4all/omsSpezifikationen/Generation5/OMS-Spec_Vol2_AnnexQ_A101.pdf). [Poskus dostopa 19. 9. 2024].
- [13] Jason R. "Adding white noise to complex signal (complex envelope)" [Elektronski]. Available: <https://dsp.stackexchange.com/questions/16216/adding-white-noise-to-complex-signal-complex-envelope>. [Poskus dostopa 19. 9. 2024].
- [14] Doering, E. "Utility subVI: Convert Eb/No to standard deviation" [Elektronski]. Available: <https://www.youtube.com/watch?v=5Dmt-27e0nw>. [Poskus dostopa 19. 9. 2024].