



Univerzitet Union
Računarski fakultet

Diplomski rad

Tehnike interlivinga u zaštitnom kodovanju za kanale sa
paketima grešaka

Mentor:

Prof. dr Desimir Vučić

Student:

Relja Vukajlović

Broj indeksa:

RM 03/2014

Beograd, 2020.

Sadržaj

1.	Uvod.....	4
2.	Opšti principi interlivinga	6
3.	Blok interliving	8
4.	Slučajni (<i>random</i>) interliving.....	11
5.	Konvolucioni interliving	14
6.	Linearni interliving	17
7.	Interliving i turbo kodovi	19
7.1.	Performanse turbo kodova	21
8.	Zaključak.....	24
9.	Literatura.....	25

Apstrakt diplomskog rada

Student: Relja Vukajlović RM 03/14

Smer: Računarske mreže i komunikacije

Mentor: Prof. dr Desimir Vučić

Naziv teme: Tehnike interlivinga u zaštitnom kodovanju za kanale sa paketima grešaka.

Apstrakt

Realni kanali su često nestacionarni i u nima se javljaju usamljene greške u dužim vremenskim intervalima, a povremeno se pojavljuju i paketi grešaka.

Tehnike interlivinga, u kombinaciji sa kodovima koji otkrivaju i ispravljaju pojedinačne greške, omogućuju da efikasno zaštiti informacija u kanalima sa paketima grešaka. Interliver permutuje pozicije u sekvenci simbola raspoređujući ih u različit vremenski redosled. Na taj način se postiže osnovni cilj interlivera da „randomizuje“ sekvencu podataka. Interliveri pretvaraju oblike dugih sekvenčnih grešaka u više slučajnih oblika grešaka, distribuirajući tako greške na više kodnih reči.

U diplomskom radu su opisana četiri najpoznatija tipa interlivera i to: blok interliver, „slučajni“ (*random*) interliver, konvolucioni interliver i linearни interliver. Opisuje se i primena tehnika interlivinga kod turbo kodova.

Kroz primere su ilustrovani principi korekcije paketa grešaka primenom različitih tehnika interlivinga.

Beograd, 22.09.2020.

1. Uvod

U toku prenosa podataka kroz kanal, greške mogu biti prouzrokovane različitim faktorima, uključujući oštećenje usled šuma, ograničeni propusni opseg kanala i druge tipove smetnji I zobičenja između kanala i izvora.

Pri prenosu informacija kroz kanal možemo govoriti o pouzdanosti sistema, tj. o tome koliko se informacija može pouzdano prenositi kroz jedan kanal. Pouzdan prenos možemo definisati kao prenos uz kontrolisanu – dopuštenu – verovatnoću greške. Dopuštena (tolerisana) verovatnoća greške varira za više redova veličina u zavisnosti od informacija koje se prenose (npr. od 10^{-2} do 10^{-3} u ekstremno teškim uslovima, ispod 10^{-6} do 10^{-8} i manje u nekim specijalnim telekomunikacionim sistemima).[1]

Dobro je poznato da su razvijeni mnogi zaštitni kodovi za ispravljanje i kontrolu grešaka (*ECC – error correction (control) codes*) kako bi se ispravile greške i osigurala tačnost podataka.

Da bi se greške nastale pri prenosu otkrile i ispravile, mora se u prenošene poruke uneti izvesna redundansa, što rezultira u onim što je poznato kao kodovi za ispravljanje slučajnih grešaka.[10]

Međutim, kodovi za kontrolu grešaka nisu efikasni u borbi protiv niza (paketa/*burst*) grešaka.

U analizi kanala potrebno je pomenuti kanal sa impulsnim šumom (*burst of noise*). Impulsne smetnje će izazvati niz uskcesivnih pogrešnih bita, tačnije rečeno, verovatnoća greške po bitu će u intervalu njihovog delovanja biti velika, u najgorem slučaju 0.5 za binarni kanal. Na taj način se može smatrati da se u kanalu pojавio paket grešaka (*burst of errors*). Filozofski gledano, Gausov beli šum je najnezgodnija smetnja i izaziva statistički nezavisne greške, pa problem koncentrisanja grešaka treba da bude lakše rešiv od problema usamljenih grešaka, pri istoj srednjoj verovatnoći greške po bitu. Međutim realni kanali su često nestacionarni u smislu da se na njima u dužim periodima javljaju usamljene greške, a povremeno i paketi grešaka.

Sada dolazimo do pojma interlivinga (eng. *Interleaving – pletenje, upredanje*) i postupka koji koristimo za “borbu” protiv paketa grešaka.

Interliving je postupak za preuređivanje simbola koda kako bi se proširio/rasporedio niz grešaka na više kodnih reči koje ECC mogu ispraviti. Pretvaranjem niza grešaka u nasumične greške, interliving tako postaje efikasno sredstvo za borbu protiv paketa (*burst*) grešaka.

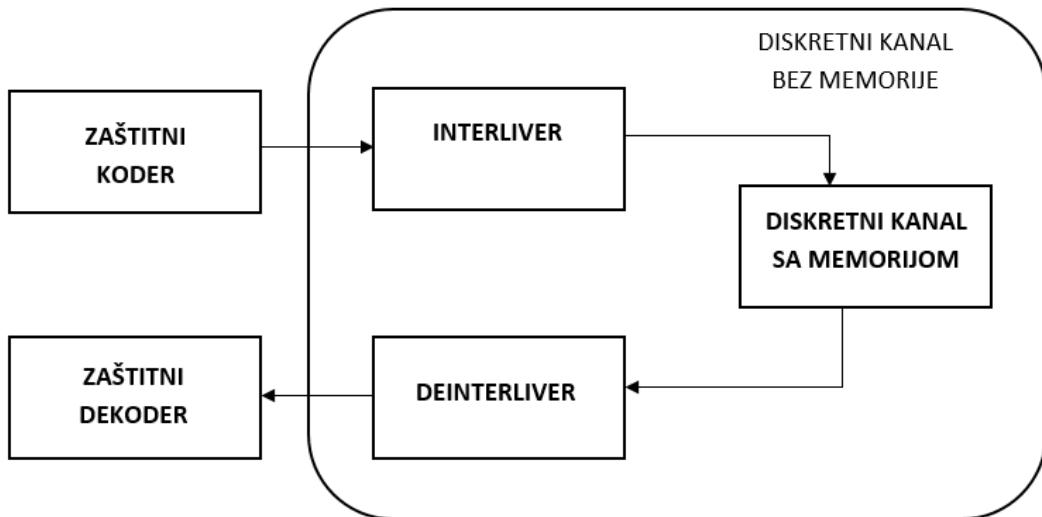
Interliving je tehnika koja se široko koristi u digitalnim komunikacijama i sistemima za memorisanje. Interlever uzima dati niz simbola i permutuje njihove položaje, raspoređujući ih u

različit, vremenski redosled. Osnovni cilj interlivera je randomizacija sekvence podataka. Kada se koriste protiv paketa grešaka, interliveri su dizajnirani da pretvore obrasce grešaka koji sadrže duge sekvence serijskih pogrešnih podataka u više kodnih reči koje ECC kodovi mogu uspešno da isprave. Paketi grešaka su karakteristični za neke kanale, poput bežičnog kanala, a javljaju se i u spojenim (kaskadnim) kodovima (*code concatenation*), gde unutrašnji dekoder, preopterećen greškama, može proslediti niz grešaka na spoljni dekoder.

Uopšteno, interliveri podataka mogu se klasifikovati u:

- Blok interliving
- Slučajni (*random*) interliving
- Konvolucioni interliving
- Linearni interliving

U bloku interlevera, podaci se prvo zapisuju u formatu reda u permutacijskoj matrici, a zatim čitaju u formatu kolone. Pseudo-slučajni interliver je varijacija blok-interlivinga gde se podaci čuvaju u registru na pozicijama koje su slučajno određene. Konvolucioni interliving karakteriše pomeranje podataka, obično primenjeno na fiksni i kumulativni način. Linearni interleveri su blok interleveri kod kojih se položaji podataka menjaju sledeći linearни zakon.[5]



Slika 1: Blok-šema sistema koji koristi interliving.

2. Opšti principi interlivinga

Interliving je postupak preuređivanja redosleda niza podataka na deterministički način, pri čemu je preuređivanje tipa “jedan-na-jedan”. Deinterliving je inverzni postupak ovog procesa koji vraća primljeni niz u svoj originalni redosled.

Postupak interlivinga, veličine N , prikazan je na slici 2.



Slika 2: Interliving.

Zbog jednostavnosti, prepostavlja se da je niz podataka na ulazu u interlivera I binarni, dat kao:

$$\mathbf{c} = (c_1, c_2, c_3, \dots, c_N)$$

gde $c_i \in \{0, 1\}$, $1 \leq i \leq N$. Interlever permutuje sekvecu \mathbf{c} u binarnu sekvencu:

$$\tilde{\mathbf{c}} = (\tilde{c}_1, \tilde{c}_2, \tilde{c}_3, \dots, \tilde{c}_N)$$

gde $\tilde{c}_i \in \{0, 1\}$, $1 \leq i \leq N$. Sekvenca $\tilde{\mathbf{c}}$ ima sve elemente iz \mathbf{c} ali u drugom redosledu. Ako ulaznu sekvencu \mathbf{c} i izlaznu sekvencu $\tilde{\mathbf{c}}$ posmatramo kao skup parova od N elemenata, postoji korespondencija “jedan-na-edan” $\mathbf{c} \rightarrow \tilde{\mathbf{c}}$ između svakog elementa \mathbf{c} i svakog elementa $\tilde{\mathbf{c}}$.

Na primer, neka se skup A definiše kao:

$$A = \{1, 2, 3, \dots, N\}$$

tj. elementi skupa A odgovaraju indeksima originalne (ulazne) sekvenice.

Interlivier se u posmatranom slučaju može definisati jedan-na-jedan funkcijom mapiranja (preslikavanja) indeksa:

$$\pi(A \rightarrow A): j = \pi(i), \quad i, j \in A$$

gde je i indeks elemenata u originalnoj (ulaznoj) sekvenici \mathbf{c} , a j indeks elemenata u interliving (izlaznoj) sekvenici $\tilde{\mathbf{c}}$. Funkcija mapiranja može biti predstavljen vektorom interlivinga:

$$\pi_N = (\pi(1), \pi(2), \pi(3), \dots, \pi(N))$$

Na primer, razmotrimo pseudo-slučajni interliving veličine $N=8$. Ulazna sekvenca je predstavljena je kao:

$$\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8)$$

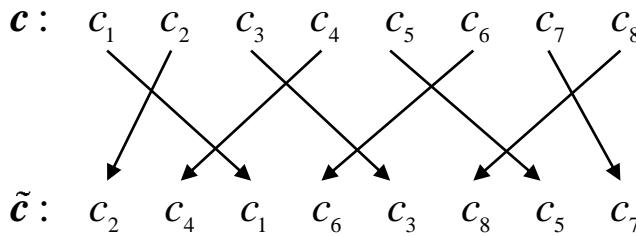
Interliving sekvenca je ima oblik:

$$\begin{aligned}\tilde{\mathbf{c}} &= (\tilde{c}_1, \tilde{c}_2, \tilde{c}_3, \tilde{c}_4, \tilde{c}_5, \tilde{c}_6, \tilde{c}_7, \tilde{c}_8) \\ &= (c_2, c_4, c_1, c_6, c_3, c_8, c_5, c_7)\end{aligned}$$

U ovom slučaju se interliving vektor može predstaviti kao:

$$\begin{aligned}\pi_8 &= (\pi(1), \pi(2), \pi(3), \pi(4), \pi(5), \pi(6), \pi(7), \pi(8)) \\ &= (3, 1, 5, 2, 7, 4, 8, 6)\end{aligned}$$

Ovo interliving preslikavanje (mapiranje) je ilustrovano na slici 3.



Slika 3: Interliving preslikavanje.

Pseudo-slučajni interliving prestavlja osnovnu tehniku interlivinga. Međutim, u praksi se treba izabrati određena struktura interlivinga koja može imati bolje performanse od pseudo-slučajnog uniformnog interlivinga (U pseudo-slučajnom uniformnom interliveru periode L, bilo koja ulazna sekvenca sa l "1" može se preuređiti u bilo koju drugu izlaznu sekvencu sa takođe l "1" sa jednakom verovatnoćom). Dalje će se razmatrati četiri tipa često korišćenih tehnika interlivinga i to: blok interliving, slučajni intrliving, konvolucioni interliving i linearни interliving.

3. Blok interliving

Blok interliver se po pravilu koristi s kodovima koji imaju fiksnu dužinu kodne reči, i to obično s klasičnim algebarskim blok kodovima, tj. kodovima koji se konstruišu koristeći metode diskretnе matematike. Pre primene interlivinga, vrši se „klasično“ kodovanje nekim blok kodom kojim se otkrivaju i ispravljaju pojedinačne greške. Međutim, formirane kodne reči se ne šalju sukcesivno na kanal, već se čuvaju u memoriji u interliveru. Kada se u memoriju upiše određen broj kodnih reči (recimo l – stepen interlivinga), tada se iz memorije šalju u kanal najpre samo prvi biti ovih reči, za njima drugi biti itd. Na prijemu se sačeka da stigne ceo blok od l kodnih reči, čiji se biti onda – u normalnom redosledu – prosleđuju dekoderu. Može se zamisliti da se, analogno dvodimenzionalnim proverama na parnost, i ovde formira dvodimenzionalni blok kodnih reči dužine n , one ovde već uključuju kontrolne bite prema sledećoj šemi [1]:

$$\begin{array}{cccc} X_{11} & X_{12} & \dots & X_{1n} \\ X_{21} & X_{22} & \dots & X_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ X_{l1} & X_{l2} & \dots & X_{ln} \end{array}$$

gde prvi indeks označava redni broj kodne reči a drugi redni broj bita u kodnoj reči. Sada se na kanal šalju biti sledećim redosledom

$$X_{11} X_{21} \dots X_{l1} X_{12} X_{22} \dots X_{l2} \dots X_{1n} X_{2n} \dots X_{ln}.$$

Na prijemu se, u deinterliveru, opet formira ceo blok kodnih reči i vrši se ispravljanje pojedinačnih grešaka. Kada se pojavi paket grešaka on će pogoditi sukcesivne bite u kanalu. Međutim, ti biti pripadaju različitim kodnim rečima i ako je paket grešaka kraći od stepena interlivinga u svakoj reči će se pojaviti najviše po jedna greška. Znači, kod kojih ispravlja jednu pojedinačnu grešku u kodnoj reči, uz interliving stepena l će moći da ispravi paket grešaka čija dužina ne prelazi l bita. Ako je paket duži od l bita ili ako se pojavi još neka pojedinačna greška u kodnoj reči koja je već zahvaćena paketom, interliving neće biti uspešan. Ceo proces je najlakše zamisliti tako da se kodne reči na predaji u bafer upisuju horizontalno, a da se sadržaj bafera iščitava vertikalno u kanal. Na prijemu se prispeli biti upisuju vertikalno u bafer, a kodne reči se iščitavaju horizontalno i daju dekoderu. Ovim postupkom biti na predaji se iz kodnih reči „raščešljavaju“, njima se superponiraju paketi grešaka u kodnom kanalu, dok se na prijemu pri „učešljavanju“ bita u kodne reči, paketi grešaka „raščešljavaju“ po kodnim rečima.

Ukupno Hamingovo rastojanje je ostalo isto, i redundansa je ostala ista, samo je ono sada tako iskorišćeno da ispravi koncentrisane greške, ali ne i veći broj grešaka po bitu, nego što je ukupno moglo da se ispravi sa svim kodnim rečima. Nedostatak interlivinga je dodatno kašnjenje

na predaji i prijemu, jer se mora formirati blok od $l*n$ bita pre slanja kodnih reči, odnosno, pre ispravljanja grešaka.

Kao što je prethodno objašnjeno, blok interliveri sastoje se od matričnog niza veličine $M_I \times N_I$ gde se nalaze podaci koji se obično pišu po redovima (*row-wise*), a zatim isčitavaju po kolonama (*column-wise*). Popunjavanje svih pozicija u matrici je potrebno, a to rezultira kašnjenjem od $M_I \times N_I$ intervala.

Blok interliving je predstavljen u sledećem primeru kao deo šeme turbo kodova. Koristi se blok interlever veličine $M_I \times N_I = 4 \times 4$. Sekvence ulaznih bitova sastoje se od 16 bitova poruke. Blok interlever veličine $M_I \times N_I$, simbole bilo kog oblika paketa grešaka dužine manje od M_I razdvaja za najmanje N_I simbola. Ako se, na primer, paket od tri uzastopne greške u sledećem nizu, upisuje po kolonama u 4x4 deinterliver (1, 5, 9, 13, **2**, **6**, **10**, 14, 3, 7, 11, 15, 4, 8, 12, 16), onda će ove greške biti razdvojene u najmanje 4 intervala.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Sekvenca na izlazu iz deinterlivera bi u ovom slučaju izgledala ovako:

$$(1, \mathbf{2}, 3, 4, 5, \mathbf{6}, 7, 8, 9, \mathbf{10}, 11, 12, 13, 14, 15, 16)$$

što potvrđuje da su greške odvojene za četiri pozicije (intervala).

U datom primeru koda za kontrolu grešaka, blok interliver je odabran da ima broj redova koji bi idealno trebalo da bude veći od najdužeg paketa očekivanih grešaka, a u praksi, najmanje toliko kolika je duzina najočekivanijih paketa grešaka. Drugi parametar blok interlivera je broj kolona u matrici, N_I , koji je uglavnom odabran da bude jednak ili veći od bloka ili dekodirane dužine koda koji se koristi. Na ovaj način paket od N_I grešaka će proizvesti samo jednu grešku po kodnom vektoru (kodna reč). Za kodove za ispravljanje grešaka koji su sposobni da isprave bilo koji oblik grešaka veličine t ili manje, vrednost N_I može se podesiti da bude veća od očekivane dužine paketa grešaka podeljene sa t .

Ako blok interliver formatira ulazni niz u matricu od m redova i n kolona, tako da je veličina interlivinga $N = m \times n$, funkcija interlivinga za blok interliving se može predstaviti kao [8]:

$$\pi(i) = [(i - 1) \bmod n] \times m + \lfloor (i - 1)/n \rfloor + 1, i \in A$$

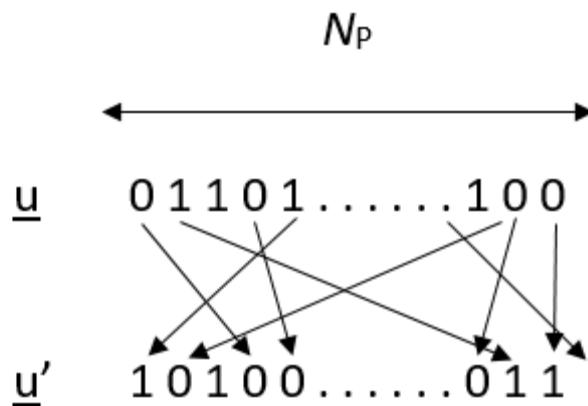
gde $\lfloor x \rfloor$ označava celobrojnu vrednost od x .

U teoriji zaštitnog kodovanja, broj redova u interliving matrici se takođe naziva stepen (ili dubina) interlivinga, a broj kolona se naziva raspon interlivinga (*interleaver span*).

4. Slučajni (*random*) interliving

Kod ovog tipa interlivinga kodovana sekvenca se takođe deli u blokove, ali njihove dužine u opštem slučaju ne moraju biti jednake umnošcima dužine kodne reči. Unutar jednog bloka dužine N_p interliver vrsi "slučajno" premutovanje bita, pri čemu se slučajnost odnosi samo na prvobitni izbor permutacije. Ova permutacija se ne menja tokom daljeg rada kodera, tj. periodično se ponavlja na narednim blokovima, a tačan zakon funkcionisanja interlivera poznat je deinterlivеру tako da u njegovom funkcionisanju nema nikakve slučajnosti.

Princip rada ovog tipa interlivera prikazan je na slici 4. Ovakav tip interlivera ima veliki uticaj na performanse turbo kodova, o čemu će biti reči u zasebnoj celini o turbo kodovima. Poseban problem predstavlja određivanje parametara slučajnog interlivera ako je poznata njegova perioda ali ne i struktura permutacije. Iako naizgled veoma težak, pokazuje se da ovaj problem ima izuzetno jednostavno rešenje čak i za velike periode interlivinga.



Slika 4: Princip rada slučajnog interlivera.

Slučajni interliveri su konstruisani kao blok interliveri gde se položaji podataka određuju nasumično. Za konstrukciju ovih interlivera može se koristiti pseudo-slučajni generator. Zahtevi za memorijom slučajnog interlivera su veličine $M_I \times N_I$ simbola, ali i pošto i postoji praktična potreba da postoje dva interlivinga, jedan za pisanje (popunjavanje) i drugi za čitanje (praznjnenje), stvarni zahtev za memorijom je tada $2 M_I \times N_I$ simbola.

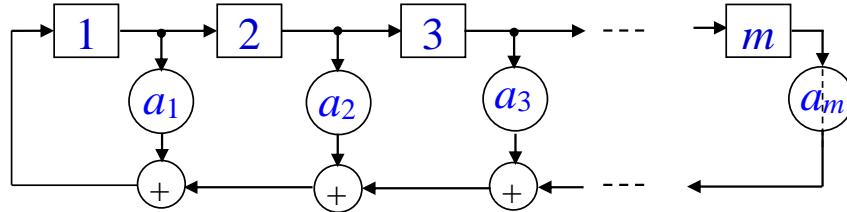
Vektor slučajnog interlivinga $\pi(i)$, $i \in 1, 2, \dots, N$, se može generisati u skladu sa sledećim algoritmom [8] koji zahteva N koraka:

Korak 1: Bira se slučajno ceo broj i_1 iz skupa $A = \{1, 2, \dots, N\}$, uniformne raspodele između 1 i N , sa verovatnoćom $p(i_1) = 1/N$. Izabrani ceo broj i_1 se postavlja da bude $\pi(1)$.

Korak k : ($k > 1$) Bira se slučajno ceo broj i_k iz skupa $A_k = \{i \in A, i \neq i_1, i_2, \dots, i_{k-1}\}$, uniformne raspodele, sa verovatnoćom $p(i_k) = 1/(N-k+1)$. Izabrani ceo broj i_k se postavlja da bude $\pi(k)$.

Kada je $k = N$, poslednji ceo broj i_N se postavlja da bude $\pi(N)$.

Kada je veličina intrlivinga $N = 2^m - 1$, pseudo-slučajni intrliver se može generisati pomeračkim registrom sa m stanja i linearom povratnom vezom kao što je prikazano na slici 5.



Slika 5: Pomerački registar sa m stanja i linearom povratnom vezom.

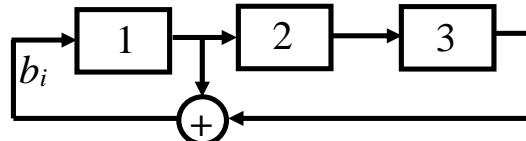
Povratne veze u pomeračkom registru se mogu definisati primitivnim polinomom stepena m (generišući polinom):

$$1 + a_1D + a_2D^2 + \dots + a_mD^m$$

Ako početno stanje pomeračkog registra nije nulto, pomerački registar će proći kroz svih $2^m - 1$ stanja ciklično. Prema tome, stanje m -stopenog pomeračkog registra može predstavljati funkciju interlivinga.

Primer [9]

Na primer, za interliver veličine $N = 7$, pseudo-slučajni interliver može se generisati pomoću pomeračkog registra sa 3 stanja sa primitivnim polinomom $1 + D + D^3$. Odgovarajući pomerački registar je dat na slici 6.



Slika 6: Pomerački registar sa primitivnim generišućim polinomom $1+D+D^3$.

Prepostavimo da je početno stanje 010 a ulazna sekvenca je $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, c_6, c_7)$. Ako je početno stanje pomeračkog registra 010, sva stanja pomeračkog stanja su određena u sledećoj tabeli:

	$(1)_i = b_{i-1}$	$(2)_i = (1)_{i-1}$	$(3)_i = s^{(1)}_{i-1}$	$b_i = (1)_i \oplus (3)_i$
1	0	1	0	0
2	0	0	1	1
3	1	0	0	1
4	1	1	0	1
5	1	1	1	0
6	0	1	1	1
7	1	0	1	0
1	0	1	0	0

Ako izlaz interlivera odgovara brojevima koji predstavljaju sucesivna stanja pomeračkog registra u oktoalnoj formi, tada se interliving vektor može predstaviti kao:

$$\begin{aligned}\boldsymbol{\pi}_7 &= (\pi(1), \pi(2), \pi(3), \pi(4), \pi(5), \pi(6), \pi(7)) \\ &= (2, 1, 4, 6, 7, 3, 5)\end{aligned}$$

pa interliving sekvenca generisana pseudo-slučajnim interlivingom ima oblik:

$$\begin{aligned}\tilde{\mathbf{c}} &= (\tilde{c}_1, \tilde{c}_2, \tilde{c}_3, \tilde{c}_4, \tilde{c}_5, \tilde{c}_6, \tilde{c}_7) \\ &= (c_2, c_1, c_6, c_3, c_7, c_4, c_5)\end{aligned}$$

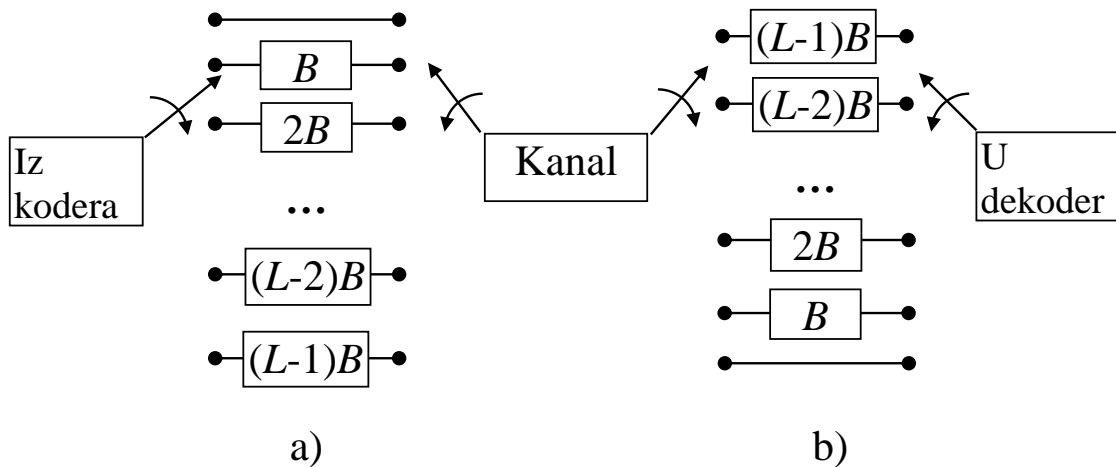
U šemi turbo kodovanja, interliver igra veoma važnu ulogu. Generalno, BER (**Bit Error Rate – broj grešaka po bitu**) performanse se poboljšavaju ako se poveća dužina interlivinga koja je deo šeme. U turbo kodu se mogu koristiti blok ili slučajni interliving. Pokazalo se da se blok interliveri bolje ponašaju od slučajnih interlivera ako je veličina matrice $M_I \times N_I$ u interliveru mala, a slučajni interliveri imaju bolje performanse od blok interlivera kada je veličina $M_I \times N_I$ matrice interlivera srednja ili velika. BER performanse kod turbo kodova sa velikim slučajnim interliverima su značajno bolje nego kod turbo kodova sa blok interliverima iste veličine. Međutim, što je veći interliver veće je i kašnjenje u sistemu. Ponekad, u zavisnosti od primene, kašnjenje prouzrokovano turbo kodom, ili tačnije njegovim interliverom, može biti neprihvatljivo za određenu stvar, i tako uprkos njihovim impresivnim BER performansama, turbo kodovi sa velikim slučajnim interliverom se prakticno ne mogu koristiti. To je slučaj, na primer, u audio aplikacijama, gde se ponekad ne može toleristi kašnjenje turbo kodova. Ako je kašnjenje prihvatljivo u određenoj aplikaciji, veliki slučajni interliveri omogućavaju da BER performanse turbo kodova budu blizu Šenonove granice. Može se zaključiti da obe porodice turbo kodova, oni konstruisani pomoću malih blok interlivera i oni napravljeni sa daleko većim slučajnim interliverima, mogu biti iskorišćene u praksi, u zavisnosti od primene. Takođe se pokazalo da su kvadratni blok interliveri bolji od pravougaonih, i da su interliveri sa neparnim dimenzijama bolji od onih sa parnim. Stoga, najbolji izbor za blok interliver se dobija korišćenjem $M_I = N_I$, pri čemu su M_I i N_I neparni brojevi.

5. Konvolucioni interliving

Konvolucioni interliveri sastoje se od ulaznog i izlaznog komutatora i skupa od L pomeračkih registara (os čega je jedan trivijalan jer ne unosi nikakvo kašnjenje). Informaciona sekvenca deli se u blokove od L bita koji se zatim, pomoću komutatora, uskcesivno ciklično prosleđuju na ulaze registara tako da se na ulaz svakog od registara šalje po jedan bit. Na taj način si i -ti bit u ulaznom informacionom bloku šalje na ulaz i -tog pomeračkog registra koji se sastoji od $(i-1)B$ celija za kašnjenje, pri čemu parametar B može biti bilo koji prirodni broj. Izlazni komutator od bita koji se u datom trenutku pojave na izlazima pomeračkih registara formira izlazni blok od L bita.

Deinterliver obavlja inverznu operaciju, što se postiže obrnutim redosledom postavljanja pomeračkih registara na izlaz ulaznog komutatora. Tada se i -ti bit sekvence na ulazu interlivera šalje na ulaz i -tog pomeračkog registra koji se sastoji od $(L-i)B$ celija za kašnjenje. Izlazni komutator zatim od izlaza pomeračkih registara formira bloki od L bita koji su identični informacionim, ali zakašnjenim za $(L-1)LB$ pozicija.

Blok šema odgovarajućeg konvolucionog interlivera i deinterlivera prikazana je na slici 7.



Slika 7: Blok-šema konvolucionog interlivera (a) i konvolucionog deinterlivera (b).

Funkcionisanje ovog tipa interlivera se može opisati izrazom

$$\pi(i) = i + [(i-1) \bmod L] * LB, i=1,2,\dots,L,$$

pri čemu $\pi(i)$ označava poziciju i -tog bita ulazne sekvence u sekvenci na izlazu interlivera. Kao što je već rečeno, ova vrsta interlivera unosi kašnjenje od $(L-1)LB$ bita a ukupan broj memorijskih celija u pomeračkim registrima iznosi $(L-1)LB/2$. Stoga je realizacija ovakvih interlivera jednostavnija, a uneto kašnjenje i memorijski zahtevi približno dva puta manji nego kod matričnog blok interlivera koji imaju uporedive performanse.

Primer:

Sekvenca bita b_1, b_2, \dots, b_{21} dovodi se na ulaz konvolucionog interlivera sa parametrima $L=3$ i $B=2$. U interliveru postoji jedan trivijalni registar, koji ne unosi kašnjenje i po jedan registar sa dve i četiri memoriske celije.

Ulagani komutator šalje prvi bit na liniju na kojoj nema kašnjenja, drugi bit na liniju koja unosi kašnjenje od $B=2$ bitska intervala a treći bit na liniju koja unosi kašnjenje od $2B=4$ bitska intervala. Ako se pretpostavi da su pomerački registri na početku rada resetovani, sekvene bita na izlazima prikazane su u tabeli 1, pri čemu su biti koji se prvi pojavljaju na izlaznom komutatoru prikazani na krajnjem desnom kraju tabele.

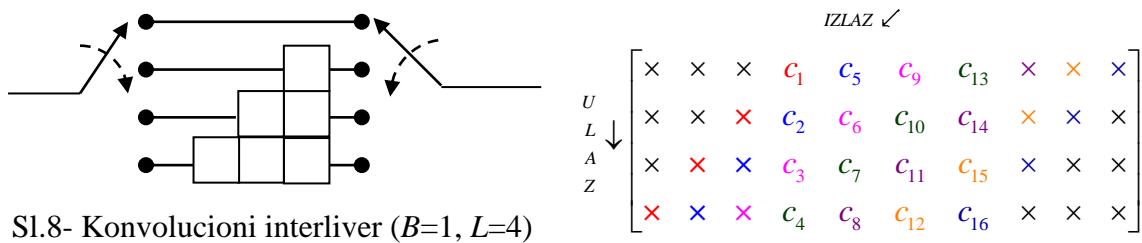
Tokom jedanaest sukcesivnih ciklusa na izlaz interlivera emituje se sekvenca

$$\begin{aligned} c = (& b_1, 0, 0, b_4, 0, 0, b_7, b_2, 0, b_{10}, b_5, 0, b_{13}, b_8, b_3, b_{16}, b_{11}, b_6, \\ & \dots, b_{19}, b_{14}, b_9, 0, b_{17}, b_{12}, 0, b_{20}, b_{15}, 0, 0, b_{18}, 0, 0, b_{21}) \end{aligned}$$

$i \backslash t$	11	10	9	8	7	6	5	4	3	2	1
1	0	0	0	0	19	16	13	10	7	4	1
2	0	0	20	17	14	11	8	5	2	0	0
3	21	18	15	12	9	6	3	0	0	0	0

Tabela 1: Uz objašnjenje rada konvolucionog interlivera, dato u primeru

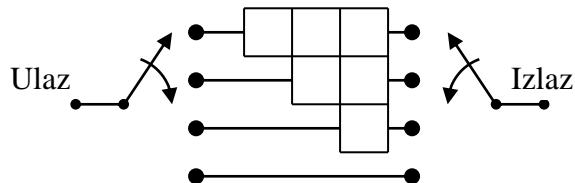
U jednostavnijem slučaju, za $B=1$, konvolucioni interliver i deinterliver pogodno je predstaviti preko matrica. Na primer, za slučaj konvolucionog interlivera sa $B=1$ i $L=4$ (prikazan na slici 8) i sekvencu bita c_1, c_2, \dots, c_{16} , popunjavaju se bitima kolone matrice sa $L=4$ vrste, a zatim se biti iščitavaju dijagonalno [9].



Dakle, na izlazu interlivera se u ovom slučaju dobija sekvenca:

$c_1 \times \times \times c_5 \quad c_2 \times \times c_9 \quad c_6 \quad c_3 \times c_{13} \quad c_{10} \quad c_7 \quad c_4 \times c_{14} \quad c_{11} \quad c_8 \times \times c_{15} \quad c_{12} \times \times \times c_{16}$

U slučaju deinterlivera postupak je sličan, popunjavaju ju se ulaznim bitima kolone matrice a izlaz deinerlivera se iščitava po dijagonalama u drugom smeru. Na primer, ako je na ulazu konvolucionog deinterlivera ($B=1, L=4$) sekvenca **1xxx00xx010x1011x110xx00xxx0**, dobijanje izlaza deinterlivera ilustruje sledeća matrica:

Sl.9-Konvolucioni deinterliviver ($B=1, L=4$)

$$\begin{matrix} & & & & & \text{IZLAZ} & \searrow \\ & & & & & & \\ \text{ULAZ} \downarrow & \left[\begin{array}{ccccccc} 1 & 0 & 0 & 1 & \times & \times & \times \\ \times & 0 & 1 & 0 & 1 & \times & \times \\ \times & \times & 0 & 1 & 1 & 0 & \times \\ \times & \times & \times & 1 & 0 & 0 & 0 \end{array} \right] \end{matrix}$$

Dakle, na izlazu deinterlivera se dobija sekvenca **1001 0110 0010 1100**.

6. Linearni interliving

Još jedna vrsta interlivinga koja se takođe koristi u šemama turbo kodova jeste linearni interliving. Jedna zanimljiva karakteristika ovog interlivinga je ta što ima matematički izraz za generisanje interliving permutacije, čime se izbegava potreba za skladištenjem kompletne strukture interlivera, koja je obično u obliku velike alokacije memorije, što je slučaj kod blok i slučajnih interlivera.

Generalno, turbo kodovi imaju impresivne BER performanse u takozvanom „vodopad“ regionu, gde kriva P_{be} (verovatnoća greške po bitu) u odnosu na E_b/N_0 (*Signal to Noise Ratio, SNR*, odnos signal-šum) strmo opada. Postoji i još jedan karakterističan region krive koja oslikava BER performanse turbo kodova, poznat kao podna regija (*error floor region*), slika 10 [2]. Ovaj podni region nastaje usled degradacije BER performansi koje uzrokuje relativno mala minimalna distanca turbo koda. Ovaj podni region je takođe posledica minimalne udaljenosti svakog od sastavnih kodova, tako da što je manja minimalna distanca sastavnih kodova, veća je verovatnoća greške (BER) na kojoj efekat podne regije počinje da se pojavljuje. Pored toga, vrsta i veličina interlivera igraju važnu ulogu u određivanju minimalnog rastojanja turbo kodova.

Jedno od rešenja za smanjenje efekta podne regije je upotreba višestrukih turbo kodova (*MTC – multiple turbo codes*). Ovi kodovi se sastoje od modifikacije klasične turbo šeme, uglavnom uključujući jedan interliver i dva sastavna koda. U opštoj MTC strukturi, postoji $J_{MTC} > 2$ sastavnih konvolucionih kodova i $J_{MTC} - 1$ interlivera, i upotreba linearnih interlivera u MTC šemi može biti veoma efikasna.

Linearni interliver dužine L_I može se opisati sledećim pravilom permutacije [5]:

$$\begin{pmatrix} 0 & 1 & \dots & L_I - 1 \\ \pi\{0\} & \pi\{1\} & & \pi\{L_I - 1\} \end{pmatrix}$$

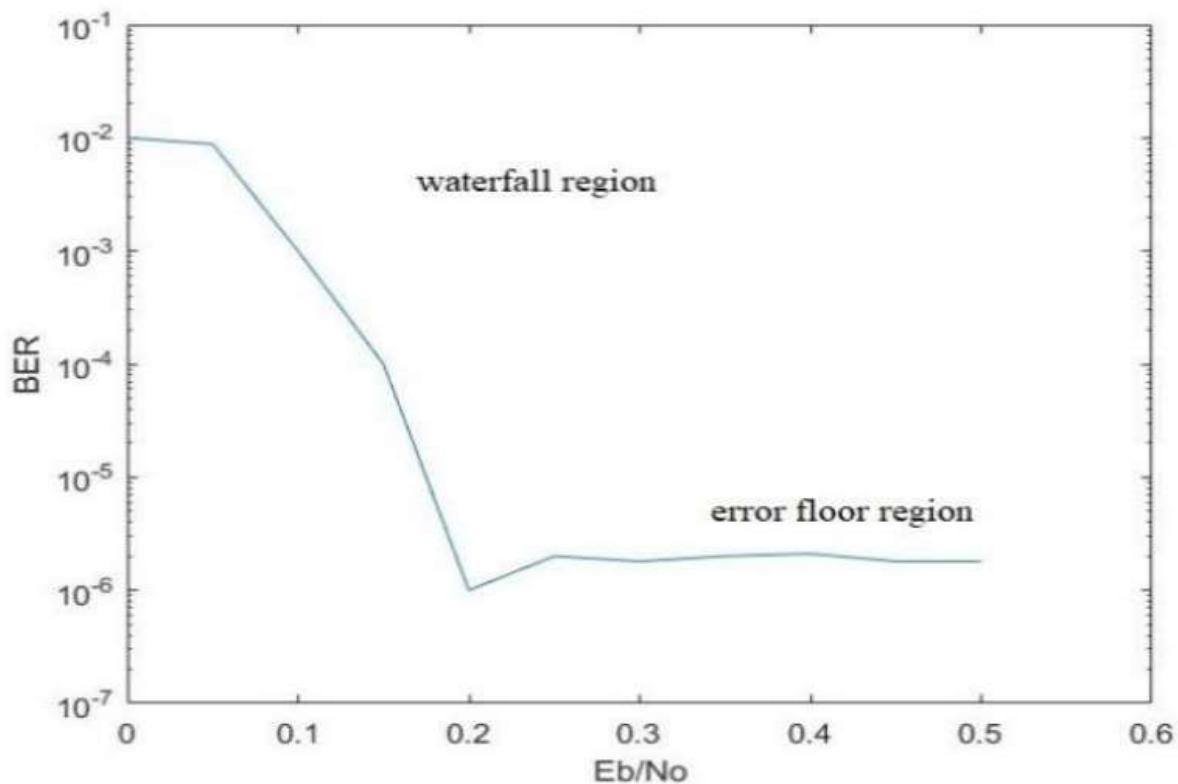
gde je

$$\pi(i) = (ip_{MTC} + s_{MTC}) \bmod L_I$$

U ovom izrazu p_{MTC} , $0 \leq p_{MTC} \leq L_I - 1$, je parametar koji se naziva ugaoni koeficijent, a s_{MTC} , $0 \leq s_{MTC} \leq L_I - 1$, je parametar koji se naziva linearni pomeraj. Neophodno je da najveći zajednički delilac (NZD) između p_{MTC} i L_I bude 1, matematički rečeno, NZD (p_{MTC}, L_I) = 1. Dakle, p_{MTC} i L_I treba da budu uzajamno prosti brojevi.

Uočeno je da dok se minimalno rastojanje tradicionalnog turbo koda, sastavljenog od dva sastavna koda i jednog interlivera, logaritamski povećava sa veličinom interlivera, minimalno rastojanje jednog MTC ispoljava veći porast koji je reda $L_I^{(J_{MTC}-2)/J_{MTC}}$. Stoga, čini se da su

linearni interliveri lako konstruisani interliveri koji takođe pružaju turbo kodove sa poboljšanim svojstvima minimalnog rastojanja, uporedivo i čak i bolje od onih turbo kodova sa drugim vrstama interlivera, kao što su DRP (*Dithread relatively prime*) interliveri ili S-random interliveri.



Slika 8: Verovatnoća greške po bitu naspram odnosa signal - šum. Prikaz "vodopad" regije i "podne" regije.

7. Interliving i turbo kodovi

Turbo kodovi su među najsavremenijim FEC (*Forward Error Correction*) tehnikama za ispravljanje grešaka. Uz pomoć ovih kodova, kapacitet kanala je prešao na novi nivo. 3G i 4G mobilne komunikacije koriste turbo kodove. Oni se takođe koriste u satelitski komunikacijama u svemiru, u aplikacijama gde se informacije pouzdano šalju kroz kanal ili u komunikaciji pod uticajem kašnjenja i oštećenjem podataka usled šuma.

Od 1948. godine istraživači su pokušavali da dizajniraju kodove približne kapacitetu kanala. Problem je ukoliko bi kod bio previše nasumičan, ne bi imao nikakvu dekodovanu strukturu, što prouzrokuje veliku složenost u procesu dekodiranja. Dakle, izazov je dizajnirati kod koji ima dovoljno nasumičnosti zajedno sa dovoljno strukture, koja se može iskoristiti za dekodovanje koda. Turbo kodovi su klasa kodova koji su slučajni (nasumični) zbog inherentne strukture preplitanja, uklopljene u strukturu paralelnog spajanja. Postoji dovoljno strukture u kodu koja omogućava efikasno dekodiranje ovih kodova za ispravljanje grešaka.[2]

Pojava turbo kodova približava nas Šenonovoj granici (dakle komunikaciji sa maksimalno mogućom brzinom u blizini kapaciteta kanala, sa nultom greškom prenosa). Front oblast je danas dominantno tzv (*Network coding*) mrežno kodovanje, odnosno teorija simultanog komuniciranja više čvorova u jednoj mreži različitim brzinama u prisustvu šumova i interferencija.

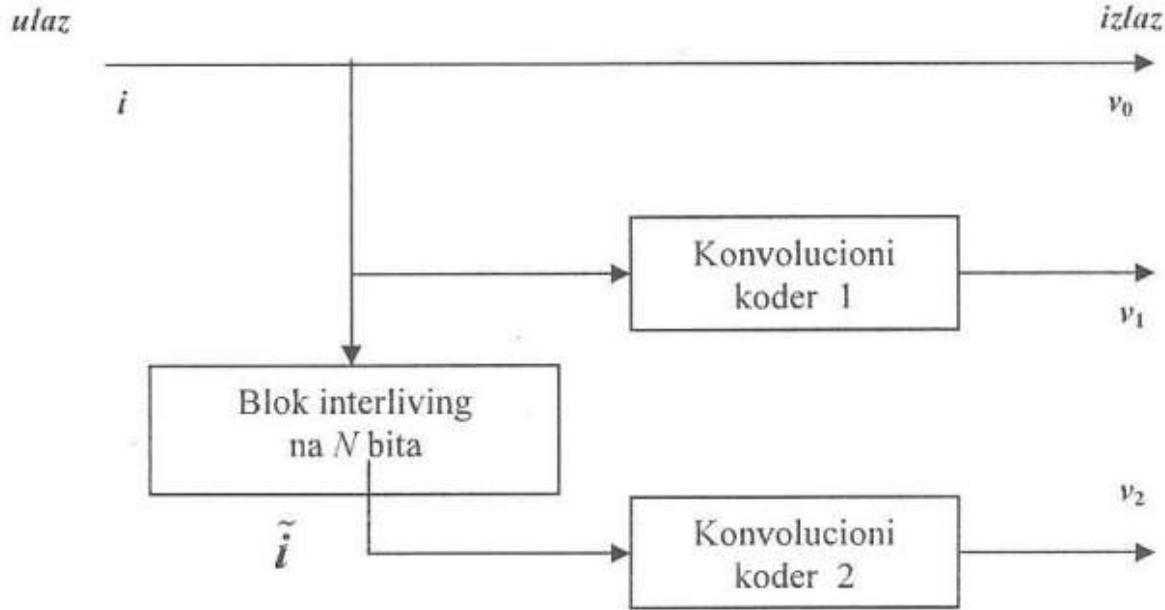
Grupa istraživača koji su radili u Francuskoj 1993. razvili su turbo kodove. Početni rezultati što se tiče energetske efikasnosti pokazali su približavanje na 0.5dB od Šenonovog kapaciteta. U početku skeptično dočekan, ovaj rezultat je ipak bio potvrđen s tim da su karakteristike koda s vremenom poboljšane. Do kraja 90-ih turbo kod je bio nadaleko poznat i implementiran u različite sisteme. Sad se nalazi u NASA standardima za svemirsку komunikaciju, digitalni video prenos i oba (cdma2000 i UMTS) standarda za mobilne uređaje treće generacije.[4] [6]

Na slici 9. je dat prikaz jednog primera implementacije turbo kodera, sa prepostavkom da su korišćeni sistematski konvolucioni kodovi. Ista informaciona sekvenca se dovodi na ulaz oba kodera, pri čemu je pre dovođenja na ulaz jednog od njih ona propuštena kroz interliver. Primjenjeni kod je konvolucioni, pa generisana sekvenca može biti beskonačna, tj. može odgovarati ukupnom broju informacionih bita koje treba preneti. Interliving je uveden da koderi ne bi štitili identične nizove informacionih bita, odnosno za „raščеšljavanje“ paketa grešaka.

Kod turbo kodova se primenjuje interliving na bloku fiksne dužine (blok interliving). Kako je reč o RSC kodovima to znači da se pre kodovanja bloka određenog broja informacionih bita, a može ih biti i više hiljada, utiskuju neinformacioni biti (najčešće sve nule) koji kodere treba da dovedu u određeno stanje. Ovaj proces se u engleskoj literaturi naziva i „tail biting“ i on osigurava da se proces dekodovanja svakog narednog bloka dešava nezavisno od prethodnog. Zaključujemo

da se turbo koder ponaša kao linearни blok koder koji generiše veoma dugačke kodne reči, što i jeste jedan od ciljeva pri konstrukciji zaštitnih kodova.[1]

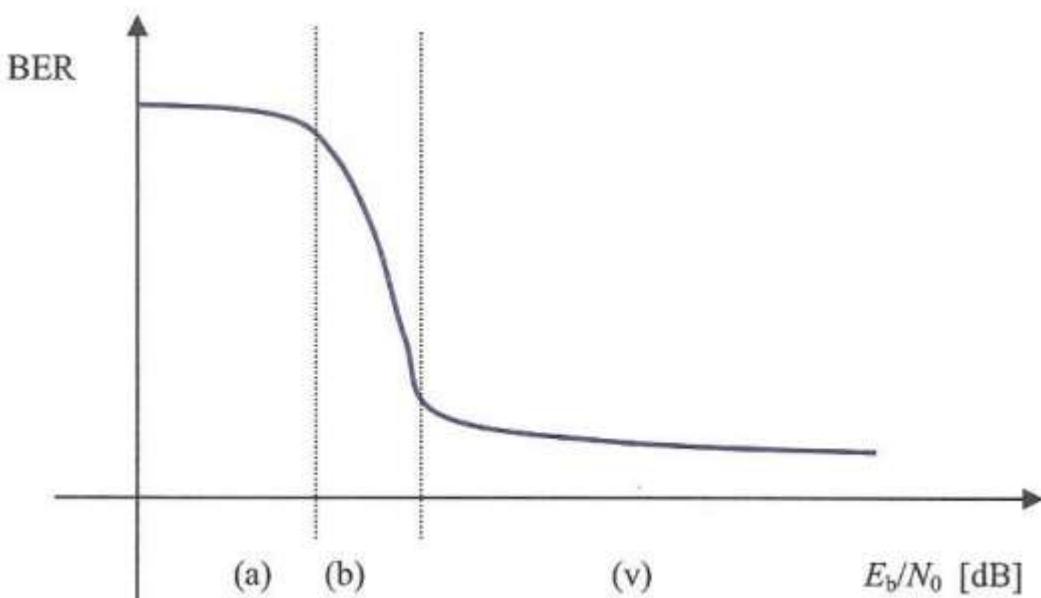
Dužina informacionog niza na ulazu označena sa N je zapravo perioda interlivera, odnosno jedan ciklus posle koga se permutacije interlivera periodično ponavljaju i primenjuju na sledeću informacioni niz (sledeću kodnu reč).



Slika 9: Jedna moguća konfiguracija kodera za turbo kod.

7.1. Performanse turbo kodova

Ponašanje verovatnoće greške po bitu kod turbo kodova šematski je prikazano na slici 10. Posmatrajući apscisu (gde je predstavljen odnos E_b/N_0) možemo uočiti tri oblasti. Pri najmanjim vrednostima odnosa signal-šum – oblast obeležena sa (a) – BER ima skoro konstantnu veliku vrednost (*nonconvergence region*), zatim ova vrednost počinje naglo da opada do srednjih ili malih vrednosti (b) (*waterfall region*), da bi se u trećoj oblasti (v) ovaj nagib naglo smanjio (*error floor region*). Pokazalo se da verovatnoća greške po bloku (FER) zavisi od samog koda, dok BER zavisi od izabranog načina kodovanja tj. od načina preslikavanja informacionih reči u kodne reči, a za to je zadužen interliver.

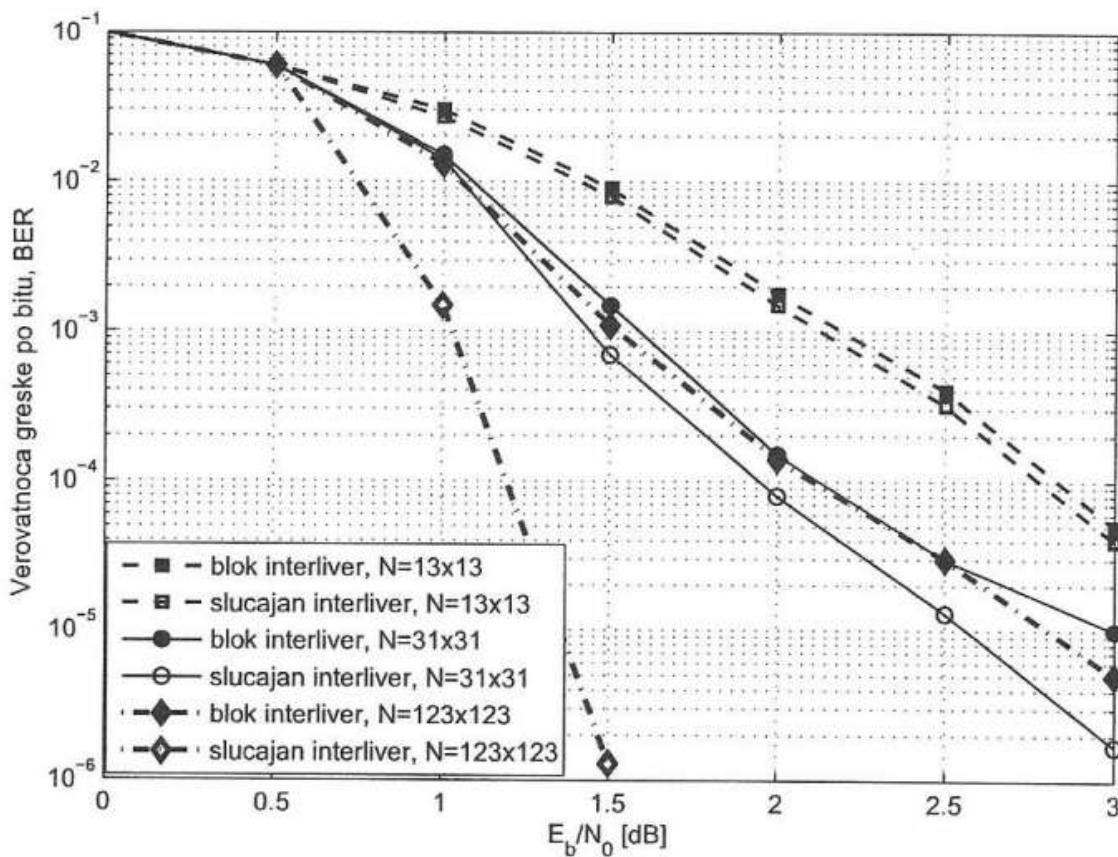


Slika 10: Šematski prikaz promene verovatnoće (količnika) greške po bitu kod turbo kodova.

Problem uticaja izbora interlivera rešen je usrednjavanjem po skupu interlivera iste dužine (*uniform interleaver*). Ovaj postupak primenjivan je još od strane Šenona u Teoriji informacija pri dokazivanju II Šenonove teoreme. Objašnjena je i posebna uloga tipa komponentnog kodera i pokazano je da su na nižim odnosima signal-šum rekurzivni sistematski konvolucioni kodovi (RSC), tj. oni koji imaju povratne sprege u koderu, bolji od nerekurzivnih konvolucionih kodova i da se u toj oblasti gornja granica BER menja obrnuto сразмерно povećanju dužine interlivera u koderu, ostvarujući dobitak interlivera (*interleaving performance gain*), kada je dužina interlivera znatno veća od memorije koda, što nije slučaj kod nerekurzivnih kodova. Nerekurzivni kodovi su nešto bolji od RSC pri većim odnosima signal-šum. Pri visokim odnosima signal-šum, i u okviru asimptotskog ponašanja, na BER utiču prvenstveno prvi nekoliko komponenata spektra rastojanja koje su prouzrokovane ulaznim sekvencama male težine. U tom domenu na performanse

sistema utiče struktura interlivera (naravno i dužina), tako da se može govoriti o **interliveru podešenom prema kodu** (*code matched interleaver*), kojim se može smanjiti *error floor*.[1]

Da bi se pokazala zavisnost performansi od tipa i dužine kodera, izvedene su simulacije kojima je određena verovatnoća greške po bitu za blok interliver (formiran po matričnom principu) i slučajni interliver iste dužine. Rezultati prikazani na slici 11. pokazuju da se performanse poboljšavaju s povećanjem dužine interlivera. Za velike dužine interlivera, primena slučajnog (random) interlivera rezultuje znatno boljim performansama. Iz ovog razloga se kada god nije poznat interliver podešen prema kodu, u turbo kodovima obično primenjuje slučajni interliver.



Slika 11: Zavisnost performansi turbo koda od vrste i dužine interlivera.

Simulacije su pokazale da se u *waterfall* regionu pri povećanju dužine interlivera smanjuje i FER, dok je poznato da u *error floor* regionu nema dobitaka interlivera. Nema ni „gubitaka“ jer se snaga turbo dekodovanja povećava s povećanjem dužine rama i to je više nego dovoljno da efikasno zaštitи duge ramove, iako se u njima pojavljuje veći broj grešaka. Treba imati u vidu da su turbo kodovi ustvari blok kodovi – imaju kodne reči konstantne dužine. Na taj način se za procenu

njihovih performansi koriste spektri rastojanja blok kodova i neke njihove modifikacije, na primer, uzima se u obzir Hemingova težina informacionog dela kodne reči.

Pored, uobičajenog, paralelnog vezivanja kodera u okviru turbo kodovanja, što je do sada ovde jedino i razmatrano, moguće je kodere vezivati i „serijski“, pa čak i hibridno. Pokazuje se da serijsko vezivanje može da ima prednost pod određenim uslovima.[1]

Osim veličine interlivera na kašnjenje u turbo dekoderu značajno utiče i broj iteracija. Zbog karakteristike iterativnog dekodovanja, tj. razmene informacija izmedju procesora, stvaranja redundanse, kašnjenje se povećava, iako ovo doprinosi boljim karakteristikama dekodera. Nakon svake iteracije estimacija podataka je bolja, iako svaka sledeća iteracija tome manje doprinosi.[4] Jedan od praktičnih načina da se smanji kašnjenje u turbo dekoderu je da se posle svake iteracije vrši istovremeno i tvrdo odlučivanje i ako se ustanovi da nema grešaka, tada se dalje iteracije mogu prekinuti. Ovo se može postići dodavanjem CRC bita informacionim bitima pre ulaska u turbo koder.

8. Zaključak

U prvom delu diplomskog rada bavili smo se tehnikama intelivinga i njihovim primenama, prikazan je opšti princip rada interlivera, detaljno su opisane četiri vrste interlivinga i kroz primere objašnjeni načini funkcionisanja svakog od njih. Uloga interlivninga je da omogući kodovima za ispravljanje grešaka da pravilno obavljaju svoju funkciju. Interliving se bavi paketima grešaka (pod paketom grešaka se podrazumeva niz od 2 ili više uzastopna bita koji su pogrešni), koje na različite načine rastavlja i deli u kodove sa pojedinačnom greškom koju kodovi za ispravljanje grešaka mogu uspešno da detektuju i otklone.

Drugi deo rada bavi se pojmom turbo kodova i ulogom interlivera u njima. Opisan je položaj interlivera i njegov uticaj na performanse turbo kodova, njihova široka primena u svetu (i van njega), od mobilnih aplikacija do satelitskih sistema. Turbo kodovi su i dalje intenzivnog proučavanja i unapređivanja.

9. Literatura

- [1] Dušan Drajić, Predrag Ivaniš, "Uvod u teoriju informacija i kodovanje", Akadembska misao, Beograd, 2009.
- [2] M. Rajani Devi, K. Ramanjaneyulu, B. T. Krishna, "A survey of interleaving techniques in turbo codes", International Journal of Engineering & Technology, 7(4), 2018.
- [3] Emilia Kasper, "Turbo codes".
- [4] Turbo kodovi (PDF).
- [5] J. C. Moreira and P. G. Farrell "ESSENTIALS OF ERROR-CONTROL CODING", John Wiley, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2009.
- [6] Mohanad Babiker, Othman Omran Khalifa, Asiya Hassan Abdullah Hashim, Momoh J. E. Salami, Muhammed Zaharadeen Ahmed, "Performance of Turbo Code in CDMA under AWGN Channel, Sience & Engineering Research Support soCiety (SERSC), 2017.
- [7] Vidya T. Ramachandran, "Turbo Codes", School of Engineering, The University of Kansas, 2008.
- [8] Branka Vučetić, Jinhong Yuan, "Turbo codes: principles and applications", Kluwer Academic Publishers, MA, United States, 2000.
- [9] Desimir Vučić, Lekcije/predavanja-Kodovanje i teorija informacija, Računarski fakultet, Beograd, 2020.
- [10] Yun Q. Shi, Xi Min Zhang, Zhi-Cheng Ni, Nirwan Ansari, "Interleaving for Combating Bursts of Errors, IEEE Circuits and Systems Magazine, 2004.