TURBO DEKODOVANJE KORIŠĆENJEM MAP ALGORITMA SA DOVOĐENJEM TRELISA OBA KODERA U ISTO KRAJNJE STANJE

Milan Sečujski
Fakultet tehničkih nauka, Novi Sad

Sadržaj *– U radu je prikazana simulacija turbo-dekodovanja korišćenjem MAP algoritma, pri čemu je korišćen ciklični par-nepar interliver dimenzija 420 bita, koji omogućuje dovođenje trelisa oba konvoluciona kodera u isto stanje. Konvolucioni koderi imali su memoriju dužine ν=4. Dekodovanje je obuhvatalo N=8 iteracija, i pri bitskom odnosu signal-šum E_b/N_0=2 dB postignuta je bitska verovatnoća greške od $6{,}1\times10^{-5}$, za 200 registrovanih grešaka. Simulacija je izvedena na osnovu opisa datog u [2], pri čemu je ukazano i na određene nekorektne pretpostavke od kojih su autori pošli. Simulacija je u potpunosti realizovana u programskom jeziku C++.*

1. UVOD

1.1. MAP algoritam

MAP algoritam predstavlja jedan od algoritama za dekodovanje linearnih kodova korišćenjem trelisa. Ovaj algoritam minimizuje bitsku (simbolsku) verovatnoću greške. Za svaki preneti simbol c_t generiše se gruba procena simbola, ali se na osnovu primljene sekvence **r** daje i mera pouzdanosti donesene odluke, i to u obliku:

$$\Lambda(c_t)=\log\frac{P_r\{c_t=1\,|\,\mathbf{r}\}}{P_r\{c_t=0\,|\,\mathbf{r}\}}$$

za $1\le t\le\tau$, gde je τ dužina primljene sekvence. Ova vrednost poredi se s pragom postavljenim na nulu da bi se dobila gruba procena c_t, na sledeći način:

$$c_t=\begin{cases}1, & \Lambda(c_t)>0\\ 0, & \Lambda(c_t)\le 0.\end{cases}$$

Vrednost $\Lambda(c_t)$ predstavlja precizniju informaciju pridruženu gruboj proceni c_t, i kao takva mogla bi se iskoristiti u sledećem stepenu dekodovanja, bilo da je u pitanju stepen kaskadnog dekodera ili naredna iteracija u postupku iterativnog dekodovanja.

MAP algoritam za dekodovanje može se koristiti u slučaju kada je kodovana sekvenca napadnuta šumom nulte srednje vrednosti, Gaussove raspodele trenutnih vrednosti, i varijanse σ^2. Prelasci kodera iz stanja l' u stanje l odvijaju se u skladu sa verovatnoćama prelazaka:

$$p_t(l\,|\,l')=P_r\{S_t=l\,|\,S_{t-1}=l'\};\ 0\le l,l'\le M_s-1,$$

gde je M_s ukupan broj stanja kodera, dok su izlazi kodera određeni verovatnoćama:

$$q_t(x_t\,|\,l,l')=P_r\{x_t\,|\,S_{t-1}=l',S_t=l\};\ 0\le l,l'\le M_s-1.$$

Imajući u vidu da su gustine raspodele verovatnoća pojedinih vrednosti oštećene binarne sekvence posle prolaska kroz kanal sa Gaussovim šumom varijanse σ^2 date sa:

$$P_r\{r_t\,|\,x_t=-1\}=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(r_t+1)^2}{2\sigma^2}},$$

$$P_r\{r_t\,|\,x_t=+1\}=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(r_t-1)^2}{2\sigma^2}},$$

aposteriorne verovatnoće pojedinih poslatih simbola (bita) mogu se odrediti na sledeći način:

$$P_r\{c_t=0\,|\,\mathbf{r}^\tau\}=\sum_{(l,l')\in B_t^0}P_r\{S_{t-1}=l',S_t=l\,|\,\mathbf{r}^\tau\}$$

$$P_r\{c_t=1\,|\,\mathbf{r}^\tau\}=\sum_{(l,l')\in B_t^1}P_r\{S_{t-1}=l',S_t=l\,|\,\mathbf{r}^\tau\},$$

pri čemu se sumiranja vrše po onim granama trelisa, odnosno, promenama stanja (l,l') koje su izazvane odgovarajućim ulaznim simbolom (0 ili 1). Ukoliko se definišu združene verovatnoće:

$$\sigma_t(l,l')=P_r\{S_{t-1}=l',S_t=l,\mathbf{r}^\tau\},\ l,l'=0,1,....M_s-1$$

logaritamski odnos se može prevesti u oblik:

$$\Lambda(c_t)=\log\frac{\sum_{(l,l')\in B_t^1}\sigma_t(l',l)}{\sum_{(l,l')\in B_t^0}\sigma_t(l',l)}=\log\frac{\sum_{(l,l')\in B_t^1}\alpha_{t-1}(l')\gamma_t^1(l',l)\beta_t\,(l)}{\sum_{(l,l')\in B_t^0}\alpha_{t-1}(l')\gamma_t^0(l',l)\beta_t\,(l)}$$

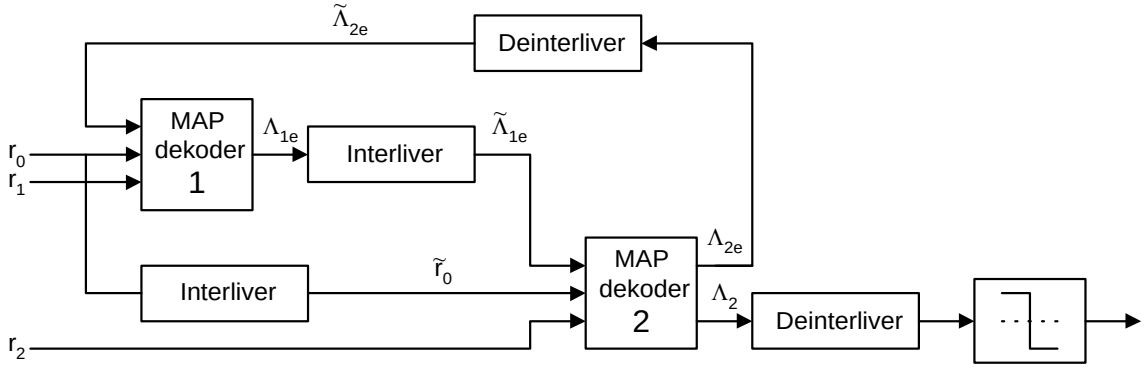
pri čemu:

$$\alpha_t(l)=P_r\{S_t=l,\,\mathbf{r}^\tau\},\ \ \beta_t(l)=P_r\{\mathbf{r}_{t+1}^\tau\,|\,S_t=l\},$$

$$\gamma_t(l,l')=P_r\{c_t=i,S_t=l,\mathbf{r}^\tau\,|\,S_{t-1}=l'\},\ i=0,1.$$

U [1] je detaljno objašnjen rekurzivan postupak na osnovu kog se, u okviru jednog prolaska kroz trelis udesno i jednog ulevo, mogu odrediti koeficijenti potrebni za nalaženje odgovarajućih združenih verovatnoća, pa ujedno i samih vrednosti $\Lambda(c_t)$, za $1\le t\le\tau$. Na osnovu ovih vrednosti se, kao što je rečeno, poređenjem s nulom može dobiti gruba procena poslate sekvence, a mogu se i kao takve proslediti sledećem stepenu u postupku dekodovanja.

MAP algoritam se, za razliku od Viterbijevog, može primenjivati samo u slučaju kada su poslate sekvence konačne dužine. Drugim rečima, pri slanju sekvence čija dužina nije unapred ograničena, neophodno je izdeliti je na blokove određene dužine.



Slika 1. *Turbo dekodovanje korišćenjem MAP algoritma*

2. Turbo dekodovanje korišćenjem MAP algoritma

Iterativno turbo dekodovanje korišćenjem MAP algoritma obavlja se uz pomoć dva MAP dekodera koji su serijski povezani preko interlivera identičnog onom koji je korišćen pri turbo kodovanju. Blok šema iterativnog turbo dekodovanja korišćenjem MAP algoritma prikazana je na Slici 1, a čitav postupak detaljno je opisan u [1].

Svaki MAP dekoder pojedinačno na ulazu dobija dve sekvence bita, kao i informaciju o do tada izračunatim verovatnoćama određenih simbola, koje smatra njihovim apriornim verovatnoćama. Pri prvoj iteraciji prvog MAP dekodera smatra se da je ta verovatnoća za svaki simbol jednaka 1/2, dok se kod svake sledeće iteracije apriorna verovatnoća pojedinih simbola određuje na osnovu procene koju je o njima dao prethodni dekoder. Na taj način MAP dekoderi razmenjuju informacije među sobom. Jednostavnosti radi, na Slici 1 nisu uzeta u obzir kašnjenja koja pojedini dekoderi unose.

Ulaz prvog MAP dekodera čine sekvenca r_0 (nastala propuštanjem informacione sekvence kroz kanal) i kodovana sekvenca r_1, (takođe propuštena kroz kanal). Ulaz drugog MAP dekodera čine interlivovana sekvenca $\tilde{r}_0$ i sekvenca r_2 dobijena kodovanjem interlivovane informacione sekvence, a zatim propuštanjem kroz kanal. Iz toga se može zaključiti da će interlivovana procena prvog dekodera $\tilde{\Lambda}_{1e}$ biti u korelaciji sa ulaznom sekvencom $\tilde{r}_0$ drugog dekodera, kao i da će deinterlivovana procena $\tilde{\Lambda}_{2e}$ drugog dekodera biti u korelaciji sa ulaznom sekvencom r_0 prvog dekodera.

Za izračunavanje apriornih verovatnoća simbola ne koristi se direktno dobijena procena verovatnoće simbola koju je dao prethodni stepen, jer u toj proceni učestvuje i informacija o apriornoj verovatnoći koju je prethodni stepen u prethodnoj iteraciji dobio, kao i podatak o tome u kolikoj meri je pomenuti simbol bio ugrožen šumom. Zbog toga je iz $\Lambda(c_t)$ potrebno eliminisati te uticaje, pa se na osnovu tako dobijenih $\Lambda_e(c_t)$ apriorne verovatnoće za sledeći stepen dobijaju u obliku:

$$p_t(1) = \frac{e^{\Lambda_e(c_t)}}{1+e^{\Lambda_e(c_t)}},\ p_t(0) = \frac{1}{1+e^{\Lambda_e(c_t)}}.$$

Kompleksnost dekodovanja turbo koda linearno raste sa brojem stanja trelisa takvog koda, a on eksponencijalno raste sa dimenzijama interlivera, tako da je turbo dekodovanje korišćenjem MAP algoritma praktično moguće ostvariti samo u slučaju interlivera relativno malih dimenzija.

2. SIMULACIJA

2.1. Dovođenje trelisa oba kodera u isto krajnje stanje

Ideja o dovođenju trelisa oba kodera u isto krajnje stanje korišćenjem naročito projektovanog interlivera prvi put je objavljena u [2]. Dovođenje oba trelisa u isto krajnje stanje značajno je zato što bi tada bilo moguće i njihovo dovođenje u nulto stanje jedinstvenom povorkom bita dužine ν, gde je ν dužina memorije konvolucionih kodera koji se koriste. Ideja koju su A. S. Barbulescu i S. S. Pietrobon izneli u svom radu jednostavna je i korektno objašnjena na primeru konvolucionog kodera memorije ν=2, ali je uopštavanje na proizvoljnu vrednost ν u istom radu izvedeno nekorektno. Autori navode da je, u opštem slučaju, kod konvolucionog kodera dužine memorije ν potrebno uzeti interliver sa brojem kolona jednakim parnom umnošku ν+1, što nije tačno. Naime, polinom u imeniocu prenosne funkcije kodera mora biti primitivan nad GF(2). U tom slučaju, cifra koja u k-tom trenutku uđe u koder uticaće na stanja kodera u određenim trenucima koji će se kasnije ponavljati s periodom 2^{ν}-1, a ne ν+1.

Primera radi, za ν=4 neophodno je uzeti interliver sa brojem kolona jednakim parnom umnošku 2^4-1=15, a ne 4+1=5, jer u suprotnom iterativni postupak turbo dekodovanja uopšte neće konvergirati. Autori navode da su izvršili uspešnu simulaciju sa interliverom dimenzija 420, pri čemu ne navode koliko je taj interliver imao kolona, a koliko vrsta. S obzirom da broj kolona mora biti paran umnožak 15, a broj vrsta uzajamno prost sa brojem kolona, jedine dimenzije interlivera s kojim bi opisana simulacija bila moguća su 60 kolona i 7 vrsta, tako da je simulacija koja je bila zadatak ovog rada izvedena korišćenjem tog interlivera.

U pomenutoj simulaciji je, kao što je navedeno, korišćen ciklični par-nepar interliver dimenzija 420 bita, koji omogućuje dovođenje trelisa oba konvoluciona kodera u isto stanje. Dekodovanje je obuhvatalo N=8 iteracija na sekvencama dužine 15000 blokova (6300000 informacionih bita), što je približno dvostruka dužina u odnosu na onu korišćenu u [2]. Ponašanje dekodera je analizirano na bitskim odnosima signal-šum u opsegu 0.5 – 2 dB.

2.2. Generisanje slučajnih brojeva

Kvalitetan generator slučajnih brojeva bio je neophodan kako za generisanje informacionih sekvenci, tako i za generisanje vrednosti odmeraka Gaussovog šuma. Pri tome je

bilo veoma bitno da generisane vrednosti u oba slučaja budu međusobno nekorelisane. Korišćen je generator slučajnih vrednosti sa uniformnom raspodelom, a po potrebi su one transformisane u vrednosti sa Gaussovom raspodelom.

Svaka u nizu pseudoslučajnih vrednosti uniformne raspodele može se generisati na osnovu prethodne dve, korišćenjem relacije:

$$n_k = \mathrm{Frac}(n_{k-1} + 33n_{k-2}),$$

gde je Frac funkcija koja vraća razlomljeni deo realnog broja. Kod ovako generisanih slučajnih brojeva dva uzastopna odmerka korelisana su u relativno maloj meri, ali su tri uzastopna odmerka u deterministčkoj vezi. Zbog ovoga je uvedena sledeća modifikacija navedenog rešenja. Kada se na opisan način generiše n_k, ono se ne uzima kao sledeća generisana slučajna vrednost, već se smešta u odgovarajuću tabelu, a broj koji je do tada bio u tabeli na tom mestu proglašava se za sledeću generisanu slučajnu vrednost. Na kojem će se mestu u tabeli izvršiti ova zamena, određuje generator slučajnih vrednosti koji može biti i slabijeg kvaliteta, tako da je u tu svrhu korišćen generator slučajnih brojeva već ugrađen u Visual C++.

2.3. Kontrola prekoračenja opsega

Pri korišćenju MAP dekodera, kao i turbo dekodera zasnovanog na MAP algoritmu, postoji opasnost od prekoračenja opsega. Kod MAP dekodera, koeficijenti $\alpha_t(l)$ i $\beta_t(l)$ se određuju rekurzivno, tako da u svakom narednom čvoru trelisa razlike između najvećih i najmanjih vrednosti tih koeficijenata veoma brzo rastu. Treba, međutim, primetiti da, pošto je $\Lambda(c_t)$ dato u obliku količnika, da se vrednosti $\alpha_t(l)$ i $\beta_t(l)$ u svakom čvoru trelisa mogu normalizovati, što će sprečiti pojavu prekoračenja pri MAP dekodovanju. Normalizacija se izvodi deljenjem svih koeficijenata $\alpha_t(l)$, odnosno, $\beta_t(l)$ faktorom

$$F(l) = e^{\frac{\log Min_t + \log Max_t}{2}}$$

gde su Min_t i Max_t minimalna i maksimalna vrednost koeficijenta $\alpha_t(l)$, odnosno, $\beta_t(l)$, u čvoru trelisa t.

Još jedno mesto osetljivo na prekoračenje jesu vrednosti koje jedan dekoder prosleđuje drugom. Pošto je u svakoj narednoj iteraciji svaki dekoder „sve sigurniji“ u to koju vrednost ima određeni simbol u informacionoj sekvenci, logaritamski odnosi $\Lambda(c_t)$ koji predstavljaju meru te pouzdanosti se iz iteracije u iteraciju uvećavaju po apsolutnoj vrednosti, te mogu dovesti do prekoračenja pri određivanju apriornih verovatnoća. Ovo se može izbeći ukoliko se $\Lambda(c_t)$ fiksiraju na unapred zadatu pozitivnu ili negativnu vrednost u slučaju da postoji opasnost od prekoračenja.

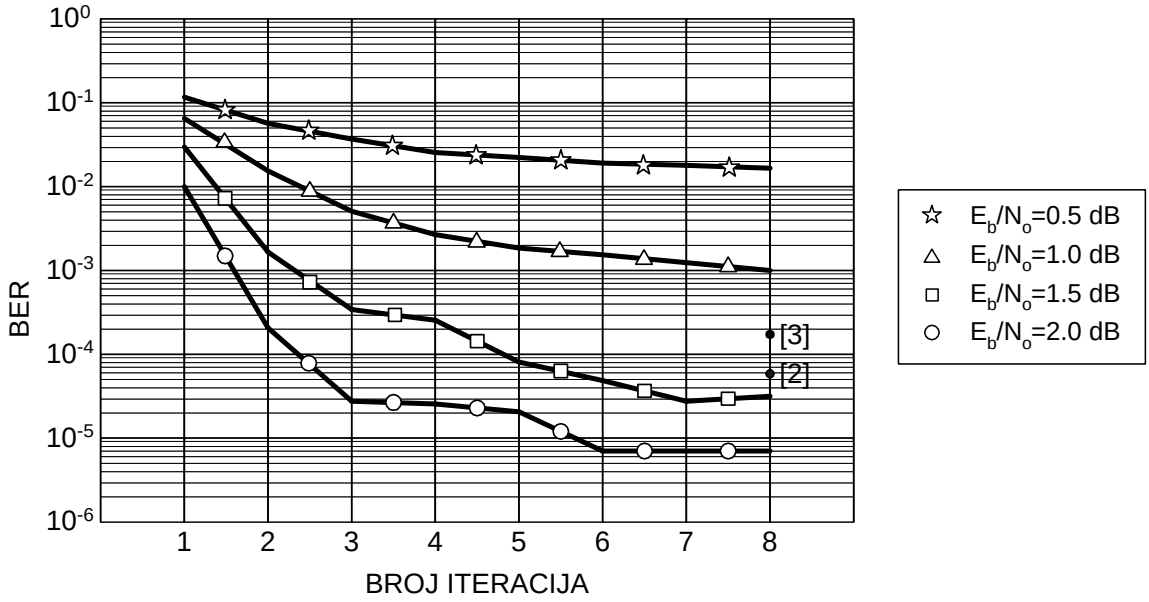
3. REZULTATI

Prilikom simulacije za kodovanje je korišćen konvolucioni koder (33/31) dužine memorije ν=4, koji u pogledu spektra rastojanja predstavlja najbolji izbor pri visokim odnosima signal-šum [1]. Za interliving je korišćen ciklični par-nepar interliver 7×60 opisan u [2], koji obezbeđuje da oba trelisa na kraju bloka završe u istom stanju, te da se mogu simultano dovesti u nulto stanje. Izvedeno je N=8 iteracija nad 15000 blokova informacionih bita dužine 420, pri čemu je E_b/N_o varirano u opsegu 0.5 do 2.0 dB u koracima od po 0.5 dB. Rezultati simulacije prikazani su u tabeli 1**,** i grafički prikazani na Slici 2. Može se zapaziti da je dobijen znatno bolji rezultat u odnosu na simulaciju opisanu u [2], gde je pri E_b/N_o=2 dB dobijena bitska verovatnoća greške $6{,}1\times10^{-5}$, posebno označena na grafiku na Slici 2. U ovoj simulaciji je pri istim uslovima dobijena bitska verovatnoća greške za red veličine niža. Poboljšanje je još izrazitije u odnosu na rezultat $1{,}55\times10^{-4}$ dobijen u [3], gde je pri simulaciji korišćen blok interliver dimenzija 20×20. Ovaj rezultat je takođe prikazan na grafiku na Slici 2.

Poboljšanje u pogledu bitske verovatnoće greške najverovatnije se javilo kao posledica toga što je implementirana varijanta MAP algoritma opisana u [1] savremenija u odnosu na pojednostavljeni Bahlov algoritam dekodovanja opisan u [3] a korišćen u [2]. S druge strane, postoje indikacije i da je određeni uticaj na brzinu konvergencije turbo dekodovanja imao i način sprečavanja prekoračenja opsega korišćen u ovom radu.

E_b/N_o [dB]	σ	ukupan broj grešaka na 6300000 bita							
		bitska verovatnoća greške $\times10^{-6}$							
		posle 1 it.	posle 2 it.	posle 3 it.	posle 4 it.	posle 5 it.	posle 6 it.	posle 7 it.	posle 8 it.
0.5	1.162	676960	357646	221609	163795	135385	118197	107035	**100273**
		107454	56769	35176	25999	21490	18761	16990	**15916**
1.0	1.097	407333	91500	31514	17571	11858	9244	7597	**6431**
		64656	14524	5002	2789	1882	1467	1206	**1021**
1.5	1.035	186547	10901	2074	892	519	308	169	**204**
		29611	1730	329	142	82,4	48,9	26,8	**32,4**
2.0	0.978	60663	675	116	100	72	43	40	**43**
		9629	107	18,4	15,9	11,4	6,8	6,8	**6,8**

Tabela 1**.** *Rezultati simulacije*



Slika 2. *Grafički prikaz rezultata simulacije*

4. ZAKLJUČAK

MAP algoritam, kao *soft-input/soft-output* dekoder pogodan za iterativno dekodovanje, predstavlja vrlo dobro rešenje za dekodovanje turbo-kodova, pogotovo u uslovima niskog bitskog odnosa signal-šum. U radu je prikazana simulacija turbo-dekodovanja korišćenjem MAP algoritma, pri čemu je, u N=8 iteracija, a pri bitskom odnosu signal-šum E_b/N_0=2 dB postignuta bitska verovatnoća greške od $6{,}1\times10^{-5}$, za 200 registrovanih grešaka. U okviru rada izvedena je simulacija MAP dekodovanja u uslovima navedenim u [2], a u skladu sa varijantom MAP dekodovanja opisanom u [1], pri čemu su dobijeni bolji rezultati nego u [2].

LITERATURA

[1] B. Vučetić, J. Yuan: „TURBO CODES - PRINCIPLES AND APPLICATIONS“, Kluwer Academic Publishers, 2000,

[2] A. S. Barbulescu, S. S. Pietrobon: „TERMINATING THE TRELLIS OF TURBO-CODES IN THE SAME STATE“, Electronics Letters, Vol 31, pp. 22-23, 1995,

[3] S. S. Pietrobon, A. S. Barbulescu: „A SIMPLIFICATION OF THE MODIFIED BAHL DECODING ALGORITHM FOR SYSTEMATIC CONVOLUTIONAL CODES“, International Symposium on Information Theory and its Applications, Sydney, Australia, pp. 1073-1077, 1994.

Abstract – This paper presents a turbo decoding scheme based on the MAP algorithm, with trellises of both turbo-codes terminating in the same state. The simulation described in [2] was carried out following the instructions given by the authors, and at $E_b/N_0 = 2$dB bit error-rate of $6{,}1\times10^{-5}$ was achieved, measured after 200 errors were detected, which is better than reported in [2]. Some observations on incorrect assumptions made by the authors of [2] were also given.

A TURBO-DECODING SCHEME BASED ON MAP ALGORITHM WITH TRELLISES OF BOTH TURBO CODES TERMINATING IN THE SAME STATE

Milan Sečujski