

**ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU**



**SOFTVERSKA IMPLEMENTACIJA E MODELA ZA PRORAČUN  
KVALITETA GOVORNE VEZE**

– Diplomski rad –

Kandidat:

Marko Jovanović 2010/0361

Mentor:

doc. dr Zoran Čiča

Beograd, Mart 2017.

# SADRŽAJ

<b>SADRŽAJ</b> .....	<b>2</b>
<b>1. UVOD</b> .....	<b>3</b>
<b>2. KODERI</b> .....	<b>4</b>
2.1. G.723.1 .....	4
2.2. G.729.A .....	5
2.3. GSM-EFR .....	5
2.4. POREĐENJE KODERA .....	6
<b>3. O PROGRAMSKOM JEZIKU APLIKACIJE</b> .....	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>4. OPIS APLIKACIJE</b> .....	<b>9</b>
4.1. IMPLEMENTACIJA E MODELA.....	9
4.2. KORISNIČKI SADRŽAJ .....	13
<b>5. PRIKAZ REZULTATA</b> .....	<b>15</b>
<b>6. ZAKLJUČAK</b> .....	<b>18</b>
<b>LITERATURA</b> .....	<b>20</b>

# 1. UVOD

Opšte je poznato da postoje dva načina za prenos podataka: jedan baziran na komutaciji kola, a drugi na komutaciji paketa. Pošto je komutacija paketa postala dominantna tehnologija, i prenos govornog signala, koji je uglavnom vezan za prenos komutacijom kola, počinje sve više da prelazi na prenos komutacijom paketa.

Da bi se govorni signal preneo putem paketske mreže, neophodno je da bude smešten u tzv. pakete. Pre nego što se paketizuje, signal mora biti digitalizovan. Proces digitalizacije obavlja se u koderima. U zavisnosti od tipa koderaz razlikuje se i kvalitet govorne veze između korisnika. Neki koderi zahtevaju manji propusni opseg nauštrb kvaliteta govornog signala i obratno. U ovoj tezi će biti pokriveni sledeći koderi: G.723.1+VAD, G.729.A+VAD i GSM-EFR.

Dva najpoznatija modela za opis kvaliteta govornog signala jesu MOS (*Mean Opinion Score*) i E model. MOS model predstavlja srednju ocenu korisnika za govornu vezu u određenim (jednakim) uslovima i ona se kreće između 1 (najniži kvalitet) i 5 (najviši kvalitet). S druge strane, E model vrši predikciju kvaliteta govornog signala, ako su poznate vrednosti parametara govorne veze te je pogodan pri planiranju i projektovanju telekomunikacionih mreža, naročito onih koje se tiču telefonije. U ovom radu biće izložen princip E modela.

Dakle, ova disertacija ima za cilj da prikaže rad jedne aplikacije koja po E modelu vrši proračun ocene R za kodere gore navedenih tipova. Drugi cilj teze je da pokaže primenu *Python* programskog jezika u realizaciji E modela, kao i grafičkom prikazu rezultata proračuna aplikacije.

## 2. KODERI

U ovome poglavlju biće opisani koderi, tačnije kodeci, koji su zastupljeni u realizovanoj aplikaciji. Kodek (*codec*) predstavlja uređaj koji kodira i dekodira signal, kompresuje i dekompresuje podatke, pomaže ili obavlja analogno-digitalnu, odnosno digitalno-analognu konverziju. Naziv je nastao kao skraćenica za *coder/decoder*. Dakle, kodek za razliku od kodera ima dvojak u ulogu.

### 2.1. G.723.1

Ovaj kodek predstavlja standard za digitalne komunikacije koje koriste 16-bitnu PCM (*Pulse Code Modulation*) modulaciju. G.723.1 koder ima veoma dobre mogućnosti kompresije i može da kompresuje originalni signal i do 12 puta. Postoji u dve verzije: u jednoj koder radi brzinom od 5.3 kbps, a u drugoj je neznatno brži – 6.3 kbps. Zbog takvih brzina, njegova primena je ograničena na prenos govornog signala pa se najviše upotrebljava za tehnologiju VoIP (*Voice over Internet Protocol*). Muzički tonovi poput DTMF (*Dual Tone Multy-Frequency*) se ne koriste jer ne postoji garancija za njihov pouzdan prenos. Stoga se poseže za drugim metodama. Jedna od metoda je upotreba G.711 kodeka, koji je robusniji i ima bolji audio kvalitet od pomenutog kodeka, zbog čega ga u praksi sve više zamenjuje. G.723.1 kodek je intelektualna svojina kompanije *Sipro Lab Telecom*.

Princip rada je takav da se govorni signal kompresuje u okvire dužine trajanja 30 ms prenoseći tako 20 bajtova brzinom od 5.3 kbps, odnosno 24 bajtova brzinom od 6.3 kbps. Pošto algoritam koristi *look-ahead* princip, čija je dužina trajanja 7.5 ms, to znači da će sada kašnjenje koje unosi koder iznositi 37.5 ms. Kako se u koderu troši vreme i na prenos paketa, njihovu obradu, signalizaciju i sl, ukupno kašnjenje kodera iznosiće 67.5 ms. Inženjerska praksa nalaže da se nikada ne računa „u knap“, tj. da se ostavi neka rezerva u slučaju nepredviđenih okolnosti. Tako je u ovoj aplikaciji uzet proračun u kojem kašnjenje kodera dostiže 70 ms, čime se obezbeđuje dobar kvalitet govorne veze čak i u najgorim uslovima.

G.723.1 koder u okviru od 30 ms sadrži 240 odabiraka, što podrazumeva učestanost odabiranja od 8 kHz. Važi za hibridni govorni koder čija je kompleksnost algoritma velika, što je još jedan razlog da ga potisne jednostavni G.711. Osim što izuzetno dobro vrši kompresiju, još jedna vrlo dobra karakteristika ostavlja G.723.1 i dalje konkurentnim. U slučaju višekanalnih operacija ima mogućnost rada i u režimu *full-duplex* i u režimu *half-duplex*, što može biti korisno u niskoprotočnim multimedijalnim sistemima sa integrisanim DSP (*Digital Signal Processing*) procesorima.

Treba još reći da se ovaj koder često koristi zajedno sa tehnologijom VAD (*Voice Activation Detection*), koja omogućava izuzetno usku širinu opsega željenog signala. Pomenuta tehnologija se naročito upotrebljava kada je u pitanju prenos govora preko Internet protokola, tj. VoIP. Tada ova softverska aplikacija ima ulogu detektovanja tišine o čemu informiše protokol PVP (*Packet Voice Protocol*). Na osnovu toga se reguliše rad kodera sprečavajući gomilanje tzv. nečujnih paketa (*silent*

*packets*), što podrazumeva da se propusni opseg manje opterećuje. Zbog svega navedenog, pojam VAD poznat je i pod izrazom „suzbijanje tišine“ (*silence suppression*).

## 2.2. G.729.A

Reč je o kodeku sličnih karakteristika kao kodek G.723.1. On takođe koristi 16-bitnu PCM modulaciju i nešto je većeg protoka – 8 kbps. Slabije kompresuje u odnosu na prethodno opisani kodek kompresujući original u razmeri 8:1. Zbog takvih karakteristika, idealan je u slučajevima kada je potrebno očuvati propusni opseg, te je njegova primena široka u tehnologijama kao što su VoIP (posebno za konferencijske pozive), ADSL (*Asymmetric Digital Subscriber Line*), satelitske veze, itd. Ovaj kodek takođe koristi VAD. Na taj način dodatno vrši očuvavanje propusnog opsega. Ovaj tip kodeka produkt je organizacija: *France Telecom*, *Mitsubishi Electric Corporation* i *Nippon Telegraph and Telephone Corporation (NTT)*.

Princip rada je takođe sličan kao u slučaju koder G.723.1. Kreiraju se okviri, koji su ovde kraći u odnosu na dotični koder i njihova dužina trajanja iznosi 10 ms. Kada se uračuna i *look-ahead* kašnjenje, dolazi se do ukupnog kašnjenja koder od 15 ms. Pored ovih kašnjenja, treba voditi računa i o kašnjenjima koja se tiču procesa unutar koder, pa tako ukupno kašnjenje broji 25 ms. Dosledno inženjerskoj praksi, i ovde je vođeno računa o dodatnoj margini. Stoga je u aplikaciji račun izvršen sa ukupnim kašnjenjem koder od 28 ms.

G.729.A koder u okviru od 10 ms sadrži 80 odabiraka, što podrazumeva učestanost odabiranja od 8 kHz. Važi za hibridni govorni koder čija je kompleksnost algoritma visoka, ali ipak niža od osnovnog G.729 koder (s kojim je kompatibilan). To znači da ova varijacija zahteva manje snage od osnovne verzije, a to može dovesti do slabijeg kvaliteta govorne veze.

U toku jednog okvira kodek prenese 10 bajtova, a svaki okvir sadrži linearne predikcione koeficijente na osnovu kojih dekodek vrši reprodukciju govora. Ovi koeficijenti deo su algoritma koji se zove CS-ACELP (*Conjugate-Structure Algebraic Code-Excited Linear Prediction*). Zahvaljujući tom algoritmu, smanjena je kompleksnost u odnosu na kodek G.723.1, u kojem su implementirani algoritmi ACELP (*Algebraic Code-Excited Linear Prediction*) i MPC-MLQ (*Multipulse LPC with Maximum Likelihood Quantization*) za slučajeve protoka 5.3 kbps i 6.3 kbps, respektivno.

## 2.3. GSM-EFR

Kodek GSM-EFR (*GSM – Enhanced Full Rate*) je konstruisan kako bi poboljšao nedostatke svog prethodnika, GSM-FR (*GSM – Full Rate*). Tu se pre svega misli na loš kvalitet zvuka koji je prisutan u *Full Rate* verziji. Za bolji kvalitet GSM-EFR kodeka zadužen je dobar algoritam, koji spada u kategoriju visoke složenosti, zbog čega je utrošak energije veći za oko 5% u odnosu na stariji model. Kao i kodeci opisani u poglavljima 2.1. i 2.2, i ovaj kodek koristi PCM modulaciju s tim što ona može biti u ovom slučaju 13-bitna ili 8-bitna.

Brzina protoka ovog kodeka iznosi 12.2 kbps, što je otprilike dvostruko više u odnosu na kodek G.723.1 i za oko 50% više nego protok kodeka G.729.A. Dužina trajanja jednog okvira iznosi 20 ms u sklopu kojeg se prenese 244 bita. Kašnjenje usled algoritma iznosi 5 ms, a ukupno kašnjenje u kome su uključeni svi procesi unutar koder dostiže 40 ms. Računajući rezervnu marginu, u aplikaciji je urađen proračun sa ukupnim kašnjenjem koder od 43 ms.

GSM-EFR koder ima frekvenciju odabiranja od 8 kHz, pri kojoj se prenese 8000 odbiraka. Podržava *full-duplex* i *half-duplex* režime i ima mogućnost višekanalne implementacije. Strukturu koderu čini pet podmodula: pretprocesorski podmodul, podmodul za kvantizaciju, generator interpoliranih parametara, podmodul za adaptivnu pretragu i podmodul za ažuriranje memorijskog filtra. Prva tri podmodula se koriste za obradu okvira i izvršavaju se jedanput po okviru, dok se preostala dva podmodula upotrebljavaju za procesuiranje podokvira i trebalo bi ih ponavljati četiri puta po okviru. Struktura dekoderu sačinjena je od svega dva podmodula: podmodul za oporavak u okvirima (trebalo bi da se izvršava jednom po okviru) i podmodul za oporavak u podokvirima, koji ima i funkciju sintetizacije govora i završnog filtriranja (trebalo bi da se ponavlja četiri puta po okviru).

Zbog svih navedenih dobrih karakteristika, njegova primena je široka. Zapravo, GSM-EFR kodek je trenutno najkorišćeniji kodek na svetu! Upotrebljava se u tehnologijama kao što su GSM (*Global System for Mobile communication*), GPRS (*General Packet Radio Service*), EDGE (*Enhanced Data Rates for GSM Evolution*) i VoIP. Koristi se i za konferencijske audio i video pozive, multimedijalne plejere i servere, zatim u sistemima sa DSP procesorima i aplikacijama namenjenih PDA (*Personal Data Assistant*) uređajima, kao DTMF detektor, identifikator poziva i u još mnoge svrhe.

Ovaj kodek je razvila kompanija *Nokia* u saradnji sa kanadskim Univerzitetom Šerbruk. Godine 1995. ETSI (*European Telecommunications Standards Institute*) je imenovao GSM-EFR kodek za industrijski standard tehnologije GSM.

## 2.4. Poređenje koderu

Može se primetiti da koderi opisani u prethodna tri poglavlja imaju sličnosti, ali svaki poseduje neku specifičnost koja ga odvaja od konkurenata. Sva tri koderu obavljaju funkciju vokoderu, tj. koderu govora (*voice coder*). Treba napomenuti da rade u realnom vremenu i svi koriste PCM modulaciju, o kojoj je bilo reči u predstavljanju svakog koderu pojedinačno. Dotični koderi su dostupni na svim poznatim operativnim sistemima, poput *Google Android*, *Apple iOS*, *MacOS*, *Unix*, *Linux*, *Microsoft Windows*, *Symbian*, *Micrium μCOS* itd.

Tabela 2.3.1. Uporedni prikaz karakteristika koderu

TIP KODERA	PROTOK [kbps]	KODNI ALGORITAM	UKUPNO KAŠNJENJE KODERA [ms]	KOMPRESIONI RAZMER
G.723.1	5.3/6.3	ACELP / MPC-MLQ	67.5	12:1
G.729.A	8.0	CS-ACELP	25	8:1
GSM-EFR	12.2	ACELP	40	4:1

Složenost jednog koderu definiše njegov kodni algoritam. Na osnovu tabele se može uočiti da je nivo kompleksnosti sva tri koderu približno jednak, s tim što je koder G.729.A nešto jednostavniji zbog konjugovane strukture algoritma. U svakom slučaju, svi ovi koderi spadaju u kategoriju visokokompleksnih vokoderu. Takođe, tabela preko datih parametara može da ukaže u kakve svrhe je najbolje iskoristiti svaki od koderu. Jasno je da je u slučaju netolerisanja velikih

kašnjenja najbolje upotrebiti koder G.729.A, dok za potrebe visoke kompresije idealan izbor predstavlja koder G.723.1. Kada je od značaja bitski protok i kada je neophodno nešto više od klasične govorne veze, nema sumnje da se tada treba opredeliti za koder GSM-EFR. Sve ove pretpostavke na osnovu parametara kodera biće proverene u aplikaciji za proračun kvaliteta govorne veze po modelu E, čiji će rezultati biti šrikazani u glavi 5.

## 3.0 PROGRAMSKOM JEZIKU APLIKACIJE

Razvojno okruženje korišćeno za ovu aplikaciju je *Canopy*. Ono ujedno predstavlja i kompajler za programski jezik Pajton (*Python*), koji je iskorišćen u pravljenju aplikacije za računanje R ocene po E modelu.

Tvorac ovog programskog jezika je Gvido van Rosum (*Guido van Rossum*), holandski programer i magistar računarskih nauka, nekada zaposleni u čuvenoj kompaniji *Google*. Svom izumu nadenuo je ovakav naziv inspirisan britanskom humorističkom serijom Monti Pajton (*Monty Python*), koja je svoj vrhunac doživljavala '70-tih, tačno dekadu pre nego što se Gvido upustio u razvijanje novog programskog jezika. Naime, radeći jedan projekat s ciljem da unapredi zaštitu od hakera, sačuvao je svoj rad baš pod nazivom *Python*. Taj naziv ostao je do današnjeg dana, kada ovaj programski jezik postaje jedan od najmoćnijih alata u oblasti računarskih i telekomunikacionih mreža.

Decembra 1989. počinje implementacija Pajtona, a od tada je izdato više njegovih verzija. Poslednja dostupna edicija do trenutka pisanja ovog rada jeste 3.6, mada je aplikacija za računanje R ocene po E modelu rađena u verziji 2.7. Glavna razlika između izdanja 2.x i 3.x jeste to što su u novijim verzijama već uključeni paketi kao što su *NumPy*, *SciPy*, *Flask* i drugi, pa ih nije potrebno posebno pozivati u samome kodu. Biblioteka je bogatija i za *Tkinter*, koji predstavlja standard za kreiranje grafičkog korisničkog interfejsa za Pajton aplikacije.

Danas Pajton predstavlja jedan od najpreglednijih *open source* programskih jezika visokog nivoa opšte namene. Podržava imperativni, objektno-orijentisani, funkcionalni i proceduralni stil pisanja koda. Za razliku od većine drugih jezika, ovde nije neophodno stavljati znak ; da bi se označio kraj naredbe, već je dovoljno samo preći u nov red. Dostupan je za sve poznatije operativne sisteme i to potpuno besplatno.

Srž filozofije ovog programskog paketa opisana je kroz aforizme u dokumentu *The Zen of Python*. Neki od njih su:

- Lepo je bolje nego ružno.
- Eksplicitno je bolje nego implicitno.
- Jednostavno je bolje nego kompleksno.
- Kompleksno je bolje nego komplikovano.
- Treba uzeti u obzir čitljivost.



## 4. OPIS APLIKACIJE

Ova glava je podeljena na dva dela. U prvom delu biće reči o softverskoj implementaciji za računanje ocene  $R$  po modelu E i strukturi njenog koda. Drugi deo baviće se korisničkim sadržajem; kako je napisan kod za interakciju sa korisnikom i kratko uputstvo za samoga korisnika.

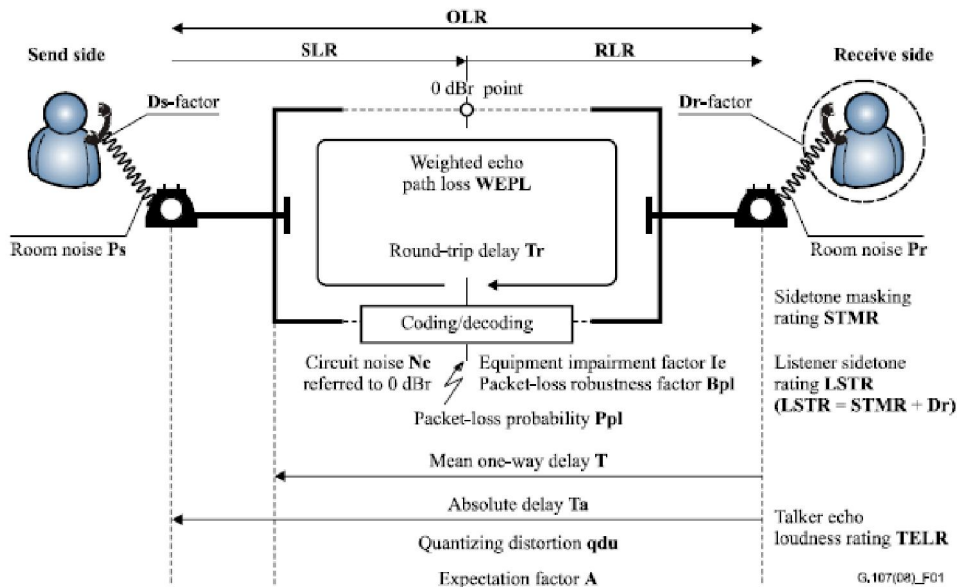
### 4.1. Implementacija E modela

E model je produkt posebne grupe organizacije ETSI poznate pod imenom „Kvalitet prenosa govora od usta do uva“. Ovaj model uzima u obzir sve relevantne parametre koji utiču na kvalitet govorne veze. Obrazac po kome se računa ocena  $R$  glasi:

$$R = R_0 - I_s - I_d - I_e + A \quad (4.1.1)$$

$R_0$  predstavlja odnos signal/šum, pri čemu se u izvore šuma ubrajaju šum linije kod analognog prenosa i šum okoline.  $I_s$  predstavlja negativan uticaj činilaca koji se javljaju uporedo sa korisnim signalom govora, kao što su kvantizacioni šum kod A/D konverzije, neoptimalna glasnost signala govora, negativni uticaj lokalnog eha i dr.  $I_d$  podrazumeva negativan uticaj kašnjenja i eha. Parametar  $I_e$  opisuje negativan uticaj kodera, kao i gubitka paketa u mreži.  $A$  predstavlja subjektivna očekivanja korisnika i jedini je faktor koji može pozitivno da utiče na ocenu veze. Taj poslednji član je u ovoj softverskoj implementaciji zanemaren, tj. ima vrednost jednaku nuli jer je svrha aplikacije prikaz objektivnih rezultata. Formula (4.1.1) je korigovana za potrebe rada ove softverske implementacije, pošto se pomenuta formula odnosi na slučaj kada nema gubitka paketa. Aplikacija razmatra scenarije kada su gubici paketa 0.1%, 0.5%, 1% i 2%. Zato se koristi efektivna vrednost  $I_e$  člana,  $I_{e-eff}$ .

Aplikacija analizira zavisnost ocene  $R$  od kašnjenja  $T$ . Možda na prvi pogled deluje kao da ocena  $R$  nije funkcija od  $T$ , međutim ovo kašnjenje figuriše u izrazima za  $I_s$  i  $I_d$ . Da bi čitalac imao jasniju predstavu o svim ovim parametrima, prikazana je jedna govorna veza na slici 4.1.1.



Slika 4.1.1. Uticaj parametara pri govornoj vezi [3]

Dakle, kašnjenje je sadržano u okviru  $STMR_0$  faktora, koji pripada članu  $I_s$ . Takođe, uticaj kašnjenja se javlja i u parametrima  $TERV$  i  $I_{die}$  člana  $I_d$ . Osim toga, kašnjenje  $T$  indirektno utiče i na faktore  $T_a$  i  $T_r$ , koji predstavljaju apsolutno i *round-trip* kašnjenje, respektivno.

Svi pomenuti parametri, i još mnogi drugi, dostupni su u preporuci ITU-T G.107. Pored ove preporuke, za svrhu rada aplikacije potrebna je korišćeni su i podaci iz preporuke G.113. Implementacija E modela u samo programsko okruženje svodi se na modeliranje rada koda parametrima iz preporuka. Njihov opis i *default* vrednosti su prikazani na slici 4.1.1.

```

1 # -*- coding: utf-8 -*-
2 #Parametri iz ITU-T preporuka, neophodni za racun R ocene
3 SLR=8.0 #Send Loudness Rating
4 RLR=2.0 #Receive Loudness Rating
5 STMR=15.0 #SideTone Masking Rating
6 LSTR=18.0 #Listener SideTone Rating
7 Ds=3.0 #D-Value of telephone, send side
8 Dr=3.0 #D-Value of telephone, receive side
9 TELR=65.0 #Talker Echo Loudness Rating
10 WEPL=110.0 #Weighted Echo Loudness Rating
11 mT=100.0 #Minimum perceivable delay
12 qdu=1.0 #Number of quantization distortion units
13 Nc=-70.0 #Circuit Noise referred to 0 dBr-point
14 Nfor=-64.0 #Noise floor at the receive side
15 Pr=35.0 #Room noise at the receive side
16 Ps=35.0 #Room noise at the send side
17 sT=1.0 #Delay sensitivity
18 Ppl=[0.1, 0.5, 1.0, 2.0] #Packet-loss rate
19 A=0.0 #Advantage factor

```

Slika 4.1.2. Implementacija parametara iz ITU-T preporuka u programskom okruženju Pajton

Može se uočiti da su sve vrednosti zapisane u decimalnom formatu. Razlog tome je što Pajton cele brojeve vidi kao *integer* tip, pa bi npr. u slučaju deljenja bio ispisan pogrešan rezultat jer

je deljenje sa dotičnim tipom brojeva celobrojno, pri čemu se ostatak zanemaruje. Ukoliko se u nekoj matematičkoj operaciji koriste mešovito *integer* i *float* tipovi, krajnji rezultat biće tipa *float*, tj. realni broj prikazan u decimalnom zapisu.

Krajnje rezultate treba ubaciti na vremensku osu i iscrtati grafik zavisnosti ocene R od kašnjenja T. Vremenska osa u ovoj softverskoj implementaciji je ograničena na 1000 odbiraka (gde svaki odbirak označava jedan milisekund), iako se mogla realizovati i na više. Međutim, za tako nešto nema potrebe jer ocena degradira ispod minimalno dozvoljene vrednosti ( $R = 50$ ) dosta pre 1000. podeoka. Za realizaciju vremenske ose i računanje vrednosti u vremenskim trenucima, tj. odbircima, upotrebljena je **for** petlja. Kao njena početna stanica, uzeta je vrednost kumulativnog kašnjenja kodera, *Tkod*. Inače, ova vrednost predstavlja jedan od ukupno tri faktora koji se razlikuju za kodere G.723.1, G.729 i GSM-EFR. Preostala dva su  $I_e$  (koji opisuje negativan uticaj kodera, kao i gubitka paketa u mreži) i *Bpl* (faktor robusnosti gubitka paketa). Svi ostali parametri imaju iste vrednosti za sva tri kodera. Iz tog razloga će u ovom radu biti posmatran samo deo koda za koder G.723.1+VAD, kako ne bi bilo ponavljanja.

```

71     #Delay impairment factor (Id): Id=Idte+Idle+Idd
72     if T<1:
73         Idte=0.0
74     else:
75         pomTERV=(1+T/10.0)/(1+T/150.0); POM_TERV=math.log10(pomTERV)
76         TERV=TELR-40*POM_TERV+6*(math.exp(-0.3*(T**2)))
77         Re=80+2.5*(TERV-14)
78         Roe=-1.5*(No-RLR)
79         Idte=((Roe-Re)/2+math.sqrt(((Roe-Re)**2)/4+100)-1)*(1-math.exp(-T))
80         #ovde nije potrebno razmatrati slucajeve za STMR, posto on spada u opseg [9, 20]dB
81
82         Rle=10.5*(WEPL+7)*((Tr+1)**(-0.25))
83         Idle=(Ro-Rle)/2+math.sqrt(((Ro-Rle)**2)/4+169.0)
84         if Ta<=mT:
85             Idd=0.0
86         else:
87             X=(math.log10(Ta/mT))/(math.log10(2))
88             Idd=25.0*((1+X**6)**(1/6.0)-3*(1+(X/3.0)**6)**(1/6.0)+2)
89
90     Id=Idte+Idle+Idd

```

Slika 4.1.3. Deo koda za računanje *Id* faktora

Slika 4.1.3. prikazuje deo koda u kome se izračunava doprinos negativnog uticaja kašnjenja i eha. Za potrebe složenijih matematičkih operacija, neophodno je uključiti biblioteku **math**, što je i učinjeno na početku programskog koda naredbom **import math**. U liniji koda br. 79 jasno se vidi da operacije korenovanja i stepenovanja prirodnog broja e koriste ovu biblioteku. S druge strane, logaritmovanje za osnovu e ne zahteva upotrebu pomenute biblioteke. Međutim, u proračunima se radi sa osnovom 10, pa je i za ovaj slučaj iskorišćena dotična biblioteka (linije koda br. 75 i br. 87). Stepenovanje realnih brojeva vrši se upotrebljavajući operator **\*\***. Preporuka G.107 razlikuje slučajeve kada je kašnjenje ispod i iznad 1 ms, na osnovu kojih faktor  $I_{dte}$  dobija različit izraz za obračunavanje vrednosti. Razgraničavanje tih scenarija implementirano je **if** konstrukcijom, koja obuhvata linije koda od 71 do 80. Komentar u liniji br. 80 odnosi se na vrednost parametra *STMR*, koja direktno utiče na izraze drugih članova. Kako je u ovoj tezi vrednost pomenutog parametra 18 dB, onda nije potrebno razmatrati slučajeve kada je *STMR* ispod 9 dB ili iznad 20 dB. Faktor  $I_{dd}$  zavisi od odnosa između parametara  $T_a$  i  $mT$ , pa je i ovde na sličan način iskorišćena **if** petlja.

Dakle, slična je priča i za izračunavanje vrednosti preostalih članova modifikovane formule (4.1.1). Modifikovane, jer se koristi efektivna vrednost paramtera  $I_e$ . Kada se svi članovi srede, potrebno ih je ubaciti u pomenutu formulu i na taj način dobiti adekvatne vrednosti za ocenu R u

svakom vremenskom trenutku. Takođe, potrebno je prikazati četiri ocene za kvalitet govorne veze. Svaka od tih ocena odgovara verovatnoći gubitka paketa ( $Ppl$ ), gde ovaj parametar uzima vrednosti 0.1%, 0.5%, 1% i 2%. Stoga je potrebno na grafiku iscrtati četiri krive. Za potrebe rada sa graficima Pajton koristi biblioteku **matplotlib.pyplot**, koja je narednom **import matplotlib.pyplot as plt** ubačena neposredno nakon što je to isto učinjeno sa bibliotekom **math**. Pošto iscrtavanje grafika zahteva rad sa listama, pre same **for** petlje definisane su prazne liste  $listT$  i  $R1$ ,  $R2$ ,  $R3$  i  $R4$ . Lista  $listT$  predstavlja vrednosti na  $x$ -osi, dok preostale četiri liste predstavljaju vrednosti na  $y$ -osi za svaku od četiri krive, odnosno za svaki od slučajeva  $Ppl$  parametra. Kako se prolazi kroz **for** petlju, svaka vrednost promenljivih  $R_i$  ( $T$ ) i  $T$  se ubacuje u odgovarajuću listu. Tako popunjene liste formiraju grafički prikaz zavisnosti ove dve veličine. Proširenje liste novim članovima obavlja se naredbom **append**.

```

89
90     Id=Idte+Idle+Idd
91
92     #Equipment impairment factor - effective (Ie_eff)
93     Ie=15.0
94     Ie_eff=[]
95     Bpl=16.1
96     #Ie i Bpl su definisane preporukom G.113, tabela I.3)
97     for j in range(4): Ie_eff.append(Ie+(95-Ie)*Ppl[j]/(Ppl[j]/BurstR+Bpl))
98
99
100    #sada treba konacna ocena R
101    R1.append(Ro-Is-Id-Ie_eff[0]+A)
102    R2.append(Ro-Is-Id-Ie_eff[1]+A)
103    R3.append(Ro-Is-Id-Ie_eff[2]+A)
104    R4.append(Ro-Is-Id-Ie_eff[3]+A)
105
106    listT.append(T-Tkod)
107

```

Slika 4.1.4. Rad sa listama

Na slici 4.1.4. prikazan je deo koda neophodan za adekvatno formiranje grafičkog prikaza rezultata proračuna govorne veze po E modelu. Uočljivo je da i parametri  $Ie\_eff$  i  $Ppl$  u osnovi predstavljaju liste u ovom kodu. Parametar  $Ie\_eff$  definiše se poput promenljivih  $R_i$  ( $T$ ) i  $T$  prvobitno kao prazna lista, ali za razliku od njih to je urađeno unutar **for** petlje. Njeno popunjavanje vrši se unutar dvostruke **for** petlje, pošto je parametar  $Ie\_eff$  direktno srazmeran parametru  $Ppl$ . Zbog toga je potrebno ubaciti brojač  $j$ , koji indeksira članove  $Ppl$  liste, tj. parametra verovatnoće gubitka paketa. Lista ovog parametra nije prazna na svom početku, kao što je to bio slučaj za ostale liste, već su u njoj upisane vrednosti koje označavaju procenat verovatnoće gubitka paketa. Ta lista je definisana na samome početku pre svih petlji.

```

107
108     plt.plot(listT,R1,listT,R2,listT,R3,listT,R4)
109     plt.title('Zavisnost R ocene od kasnjenja')
110     plt.xlabel('T [ms]')
111     plt.ylabel('R (T)')
112     plt.show()
113

```

Slika 4.1.5. Kod za iscrtavanje grafika

Kao što je već rečeno, da bi se prikazao grafik potrebno je pozvati biblioteku **matplotlib.pyplot**. Slika 4.1.5. prikazuje deo koda koji koristi pomenutu biblioteku za rad sa grafikom. Linija br. 108 definiše vrednosti koje će se javiti na grafiku, gde (*listT*, *Ri*) odgovara matematičkom paru ( $x, y_i$ ). Primećuje se da se to obavlja navodeći sve uparene promenljive, ne odvajajući ih nekim posebnim karakterom. To može biti manji problem za programera, jer je uočljivost slaba u slučaju velikog broja krivih. Naredbe u linijama koda 109, 110 i 111 definišu kako treba da izgleda naslov grafika, odnosno šta treba da se ispiše na apscisi i ordinati.

## 4.2. Korisnički sadržaj

Softverska implementacija za proračun kvaliteta govorne veze po modelu E podržava interakciju sa korisničkim svetom. Ona daje mogućnost da korisnik izabere tip kodera za koji treba izvršiti proračun. Takođe, korisnik zadaje vrednost parametra *Burst Ratio* koji predstavlja odnos prosečne dužine sekvence posmatranog slučajnog uzorka, tzv. bursta i prosečne dužine burstova očekivanih u mreži kao „slučajni“ gubitak. Za gubitak paketa kaže se da je nezavisan, tj. slučajan ako je  $BurstR = 1$ . U suprotnom ga smatramo zavisnim.

```

23
24 tip_kodera=raw_input('Odaberite tip koder: \n')
25 '1 - G.723.1+VAD \n'
26 '2 - G.729.A+VAD \n'
27 '3 - GSM-EFR \n')
28

```

Slika 4.2.1. Odabir željenog koder

Slika 4.2.1. prikazuje deo koda, čija je namena da postavi pitanje korisniku za koji koder treba da se izvrši proračun zavisnosti ocene R od kašnjenja. To se u programskom jeziku Pajton radi koristeći funkciju **raw\_input( )**. Zbog bolje preglednosti, autor aplikacije se odlučio da svaki tip koder smesti u zaseban red. To odvajanje je izvršeno zahvaljujući terminatoru **\n**. Nakon upita, korisnik treba da pritisne taster 1, 2 ili 3. Ukoliko pritisne neko drugo dugme na tastaturi, biće obavešten o grešci i zbog čega je do nje došlo. Vrednost koju unese korisnik, upisuje se u promenljivu *tip\_kodera*. Pajton ovakve vrednosti vidi kao tip *string*. Iz tog razloga, potrebno je izvršiti konverziju tipova kada korisnik bude unosio vrednost parametra *BurstR*, što je i urađeno u kodu prikazanom na slici 4.2.2.

```

28
29 if tip_kodera=='1':
30     BurstR=raw_input('Unesite vrednost parametra Burst Ratio u skupu realnih brojeva [1..8]: ')
31     BurstR=float(BurstR)
32     Tkod=70
33     listT=[ ]
34     R1=[]; R2=[]; R3=[]; R4=[];

```

Slika 4.2.2. Parametri koje unosi korisnik

Pomenuta konverzija vrši se funkcijom **float( )**. Na taj način se uneti karakteri, tzv. stringovi, pretvaraju u realne brojeve sa kojima se kasnije mogu obavljati matematičke operacije. Mogla je da se uradi konverzija i za promenljivu *tip\_kodera*, pa da uslov **if** petlje glasi bez apostrofa (koji označavaju string). Onda bi se poredila sa samom vrednošću broja. Osim konverzije u promenljive tipa *float*, moguće je konvertovati i u promenljive tipa *integer*.

Dakle, korisnik nakon odabira tipa koder dobija novo pitanje u kojem se zahteva da unese vrednost parametra *BurstR*. Obično se za njegovu vrednost uzima vrednost jednaka jedinici ili približno jednaka tom broju, pošto se za takav slučaj smatra da je gubitak paketa slučajan. Nakon ovog unosa korisnik nema više interakcija sa aplikacijom i dobija prikaz zavisnosti ocene R od

kašnjenja T po modelu E u vidu grafika. Kao što se vidi, upotreba ovog softvera je vrlo intuitivna i nadasve jednostavna. Čak i ako negde pogreši, biće mu objašnjeno kako treba da postupi sledeći put pri pokretanju programa.

## 5. PRIKAZ REZULTATA

Realizovana aplikacija se može pokrenuti na više načina. Autor ovog rada preferira njeno izvršavanje putem razvojnog okruženja *Canopy*. Nakon što pokrene dotično okruženje, korisnik treba da odabere odgovarajući fajl sa ekstenzijom **.py**. Zatim se otvara taj fajl i korisnik ima uvid u kod softverske implementacije. Sama aplikacija pokreće se klikom na dugme u obliku zelenog trougla, poznatog kao *Run*. Ovo dugme smešteno je tik ispod jezička *Window*. Drugi način je preko komandne linije, odnosno terminala. Potrebno je u njemu zadati putanju do foldera u kome se nalazi fajl. Aplikacija se pokreće tako što se eksplicitno ukuca u konzoli njen pun naziv u formatu *ime\_aplikacije.py*. U ovom slučaju korisnik nema uvid u kod aplikacije, pa ukoliko ga zanima da vidi kako je pisan softver, potrebno je da fajl otvori u nekom tekstualnom editoru. Preporuka je *Notepad++*. Još jedna razlika između ova dva načina je ta što pre pokretanja aplikacije putem terminala, korisnik mora da instalira programski jezik *Python* sa svim neophodnim bibliotečkim paketima, dok su oni već integrisani u razvojnom okruženju *Canopy*. Postoje još mnogi načini za izvršavanje aplikacije ali se oni uglavnom svode na prvu opisanu metodu, samo što je razvojno okruženje različito. Na slici 5.1. prikazan je interaktivni deo u kome se korisnik opredeljuje za tip kodera i unosi vrednost *BurstR* parametra.

```
Welcome to Canopy's interactive data-analysis environment!
Type '?' for more information.
Python 2.7.11 | 32-bit | (default, Jun 11 2016, 11:34:14) [MSC v.1500 32 bit (Intel)]
Type "copyright", "credits" or "license" for more information.

IPython 4.1.2 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

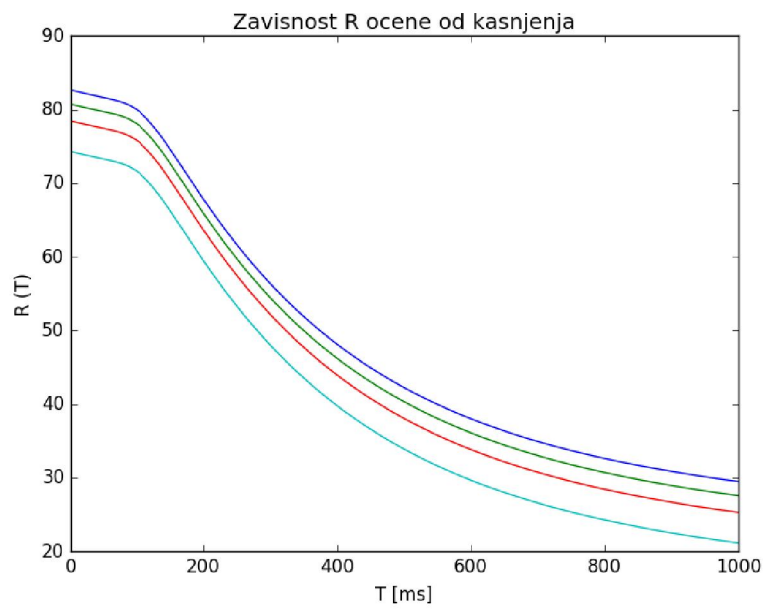
In [1]: %run "C:\Users\Markoni\2010_0361.py"

Odaberite tip kodera:
1 - G.723.1+VAD
2 - G.729.A+VAD
3 - GSM-EFR
3

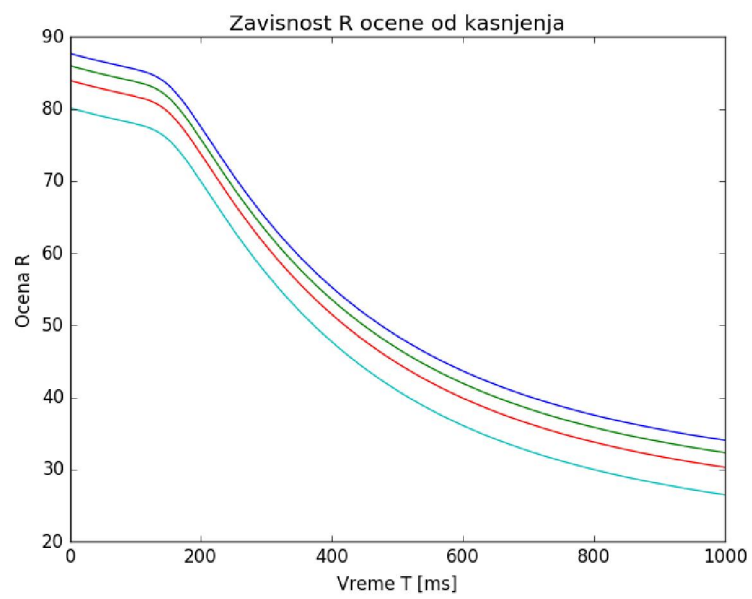
Unesite vrednost parametra Burst Ratio u skupu realnih brojeva [1..8]: 1
```

**Slika 5.1. Korisnik unosi vrednosti parametara u aplikaciji, pri čemu se koristi razvojno okruženje *Canopy***

U nastavku ove glave biće prikazani rezultati proračuna i njihova analiza. Kako bi upoređivanje kodera bilo fer, za svaki od njih izvršen je proračun sa parametrom  $BurstR = 1$ .

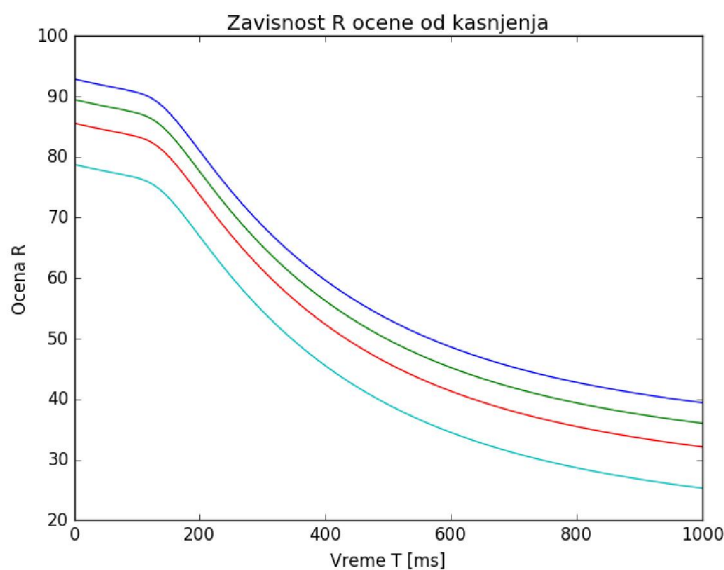


Slika 5.2. G.723.1+VAD



Slika 5.3. G.729.A+VAD





Slika 5.4. GSM-EFR

Na slikama 5.2, 5.3. i 5.4. grafički su prikazane ocene R po modelu E u vremenskim periodima do 1000 ms. Normalno je da su najbolje ocene dobila ona merenja kod kojih je verovatnoća gubitka paketa najmanja. One su označene plavom bojom. Analogno tome, najgori kvalitet govorne veze javlja se u slučajevima kada je  $Ppl = 2\%$ . Zanimljivo je to što je GSM-EFR koder najosetljiviji po pitanju gubitka paketa, tj. najveća razlika u oceni između najniže i najviše stope gubitka paketa javlja se baš kod ovog koder.

Sva tri koder daju izuzetno dobre rezultate sve dok je kašnjenje do 100 ms. Taj period odličnih ocena u slučaju koder G.729.A se proteže i do 150 ms. Ovaj koder zajedno sa koderom GSM-EFR postiže dobre rezultate i u pogledu na duža kašnjenja, pa tako oni daju prelaznu ocenu čak i kada je ukupno kašnjenje oko 350 ms. Naročito tolerantan na kašnjenja je G.729.A, što se i moglo očekivati s obzirom da je njegovo kumulativno kašnjenje koder najmanje.

Generalno gledano, koder G.729.A dobija ubedljivo najviše ocene. Koliko je dominantan svedoči i njegova najniža ocena. Naime, kada je verovatnoća gubitka paketa 2%, ovaj koder ima neznatno slabiji kvalitet govorne veze od koder G.723.1 sa verovatnoćom gubitka paketa 0.1%. Razlog zbog čega postiže visoke ocene dotični koder leži u tome što su njegovi parametri dobri u svim značajnim segmentima. Ima dobar bitski protok, ali ne i najbolji; ima dobar kompresioni razmer, ali ne i najbolji. Kada se na to dodaju i vrline poput najmanjeg kašnjenja i najmanje kompleksnosti od ova tri koder, jasno je zašto koder G.729.A ima najvišu ocenu. Žargonski rečeno, on je jedan „*all-around* koder“.

Posmatrajući pojedinačne rezultate, najbolji skor ima koder GSM-EFR. On jedini ima ocenu koja prelazi  $R = 90$ . Čak i scenario sa verovatnoćom gubitka paketa od 0.5% daje rezultate blizu ove visoke ocene. Problem nastaje sa povećanjem  $Ppl$  parametra. Zbog toga se ne preporučuje upotreba ovog koder kada je mogućnost izubljenog paketa veća od 1.5%, jer je tada bolje iskoristiti koder G.729.A, koji daje približno iste rezultate za isti  $Ppl$  parametar ali je zato manje kompleksnosti.

Uočljivo je da na ovom testu koder G.723.1 nije „pobedio“ ni u jednom segmentu svoje konkurente. Međutim, postoji jedan deo u kome je ovaj koder bez premca, a to je kompresija. Definitivno najviše može da suzbije original i to je njegova glavna (verovatno i jedina) prednost, baš kao što je i bilo napomenuto u poglavljima 2.1. i 2.4. Ovim merenjima je i pokazano da su predikcije za sva tri koder na osnovu tabele 2.4.1 bile dobre.

## 6. ZAKLJUČAK

Softverska implementacija za proračun kvaliteta govorne veze po E modelu je dobar metod za efikasno spoznavanje zavisnosti ocene  $R$  od kašnjenja  $T$ . Ona daje brzu procenu inženjeru, na osnovu koje on donosi odluke o tipu koda, načinu implementacije, promeni pojedinih parametara i sl. Aplikacija je jednostavna za upotrebu, nije sistemski zahtevna i lako se može izmeniti kod u slučaju željenih promena.

Takođe, na osnovu ovog proračuna moguće je dobiti i MOS ocenu, pošto postoji direktna veza između nje i E modela. Međutim, taj segment nije implementiran u ovoj verziji aplikacije, pa bi to mogla da bude ideja za neko njeno eventualno ažuriranje. Osim toga, kao *upgrade* aplikacije moglo bi poslužiti i proširenje liste koda za koje se vrši proračun. S obzirom da postoji tendencija objavljivanja novih izdanja programskog jezika Pajton koja uključuju sve bogatiji program grafičkog korisničkog interfejsa, eventualna nova verzija ove aplikacije bi mogla to iskoristiti i uvesti korisničku interakciju na jedan lepši i zanimljiviji nivo

## LITERATURA

- [1] Z. Čiča, „Paketizacija govornog signala“. Dostupno na:  
[http://telekomunikacije.etf.rs/predmeti/te4ks/docs/KS/KS\\_08.pdf](http://telekomunikacije.etf.rs/predmeti/te4ks/docs/KS/KS_08.pdf)
- [2] M. Bjelica, „Programski jezik Python“, 2016. Dostupno na:  
[http://www.etf.bg.ac.rs/etf\\_files/udzbenici/python.pdf](http://www.etf.bg.ac.rs/etf_files/udzbenici/python.pdf)
- [3] ITU-T Rec. G.107, June 2015.
- [4] ITU-T Rec. G.113, Nov. 2007.
- [5] ITU-T Rec. G.113, Amendment 1, Mar. 2009.
- [6] C. Severance, „Python for Informatics“, Oct. 2013.
- [7] M. Lutz „Learning Python“, 5th Edition, June 2013.
- [8] Cisco Systems, „Internetworking Technologies Handbook“, 4th Edition.
- [9] V. K. Garg, „Wireless Network Evolution 2G to 3G“, 2007.
- [10] International Engineering Consortium, „Annual Review of Communications“, Volume 59.
- [11] <https://www.vocal.com/speech-coders/>
- [12] <http://searchnetworking.techtarget.com/definition/codec>
- [13] <http://www.webopedia.com/TERM/C/codec.html>
- [14] [http://www.radio-electronics.com/info/cellulartelecomms/gsm\\_technical/audio-codecs-vocoders-amr-celp.php](http://www.radio-electronics.com/info/cellulartelecomms/gsm_technical/audio-codecs-vocoders-amr-celp.php)
- [15] [http://www.etsi.org/deliver/etsi\\_i\\_ets/300700\\_300799/300723/01\\_30\\_115/ets\\_300723e01v.pdf](http://www.etsi.org/deliver/etsi_i_ets/300700_300799/300723/01_30_115/ets_300723e01v.pdf)
- [16] <http://www.signallogic.com/index.pl?page=gsmefr>
- [17] <http://www.voiceage.com/EFR-GSM.html>
- [18] [https://en.wikipedia.org/wiki/Main\\_Page](https://en.wikipedia.org/wiki/Main_Page)