ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU

APLIKACIJA ZA PRIKAZ ZAVISNOSTI OCENE KVALITETA GOVORNOG SIGNALA OD KAŠNJENJA KROZ MREŽU

– Diplomski rad –

Kandidat:

Miroslav Vučevski 2008/474

Mentor:

doc. dr Zoran Čiča

Beograd, Oktobar 2014.

SADRŽAJ

SADRŽAJ 2
1. UVOD 3
2. E MODEL OCENJIVANJA 4
3. OPIS APLIKACIJE I KODA 7
4. PRIKAZ RADA APLIKACIJE 13
5. ZAKLJUČAK 17
LITERATURA 18
A. KOD APLIKACIJE 19

1. UVOD

Danas su dominantne mreže bazirane na paketskoj komutaciji. Da bi se govorni signal uopšte preneo preko paketske mreže potrebno je da se on smesti u pakete, a pre samog smeštanja u pakete potrebno je obaviti digitalizaciju dotičnog govornog signala. Digitalizaciju govornog signala vršimo koderima koji su preporučeni ITU-T preporukama kao što su koderi G.711, G.726, G.729, ali i drugim koderima. Za razliku od komutacije kola gde su jednom telefonskom razgovoru dodeljeni resursi koje koristi isključivo dotični razgovor, u mrežama koje se baziraju na komutaciji paketa resurse dele svi korisnici. Iz navedenog razloga lakše je garantovati kvalitet servisa u mrežama sa komutacijom kola, dok se u mrežama sa komutacijom pakete mogu obezbediti statističke granacije.

Na kvalitet govornog signala tj. razgovora utiče dosta parametara kao što su kašnjenje, varijacija kašnjenja, tip kodera i dr. Bilo je poželjno razviti analitički model koji bi uključio sve relevantne parametre i obezbedio dobru ocenu kvaliteta govorne veze. ITU-T G.107 preporuka definiše E model koji se može koristiti za predikciju kvaliteta govornog signala ako su poznate vrednosti parametara govorne veze. E model vrši proračun ocene R kvaliteta govorne veze, pri čemu E model uzima sve relevantne parametre koji utiču na kvalitet govorne veze. Iz navedenog sledi da se E model može koristiti u planiranju i projektovanju telekomunikacionih mreža kojima se između ostalog prenosi i paketizovani govorni signal.

Dalje u radu, prezentovan je E model i parametri relevatni za izračunavanje R ocene kvaliteta. Takođe je prezentovana aplikacija za izračunavanje R ocene kvaliteta po E modelu. Objašnjen je način realizacije kao i sam rad aplikacije. Aplikacija omogućuje grafički prikaz R ocene kvaliteta govorne veze u zavisnosti od jednosmernog kašnjenja pri propagaciji kroz mrežu. Aplikacija nam omogućuje da kvalitetnije predstavimo i analiziramo kvalitet veze u zavisnosti od relevantnih parametera. Za proračun i prikaz korišćen je programski alat *MATLAB*.

2. E MODEL OCENJIVANJA

ITU-T preporuka G.107 definiše model koji se može koristiti za predikciju kvaliteta govornog signala ako su poznate vrednosti parametara govorne veze i samim tim se E model može koristiti u planiranju i projektovanju telekomunikacionih mreža namenjenih, između ostalog, i prenosu telefonskih razgovora. E model vrši proračun ocene R kvaliteta govorne veze, pri čemu E model uzima u obzir sve relevantne parametre koji utiču na kvalitet govorne veze. Ocena R kvaliteta veze se izračunava po sledećem obrascu:

$$R = R_0 - I_s - I_d - I_e + A \tag{2.1}$$

R_0 predstavlja osnovni 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. I_d predstavlja negativan uticaj kašnjenja i eha. I_e predstavlja negativan uticaj kompresora govornog signala (kodera), kao i gubitaka paketa u mreži, dok parametar *A* predstavlja subjektivna iščekivanja korisnika i jedini je faktor koji može pozitivno uticati na ocenu veze. Važno je uočiti da je E model aditivan model, odnosno, svi uticaji su aditivnog karaktera na konačnu ocenu što ide u prilog jednostavnosti E modela. Na ITU-T sajtu se može naći onlajn kalkulator [1] i po uzoru na taj kalkulator je realizovana i aplikacija sa proračun ocene R po E modelu, uz dodatak da je u aplikaciji dodata grafička predstava ocene R kvaliteta govorne veze u zavisnosti od kašnjenja. U aplikaciji je omogućen unos relevantnih parametara govorne veze i zatim je moguće izvršiti proračun R ocene kvaliteta govorne veze. Ocena R se kreće u granicama od 0 – 100, pri čemu se ne preporučuje projektovati govorne veze koje imaju R ocenu nižu od 50. Lokalna ISDN veza ima najbolji kvalitet i R ocena iznosi 94. Ocena R = 94 se dobija u slučaju korišćenja difolt vrednosti koje su definisane ITU-T G.107 preporukom.

Parametar R_0 se računa po formuli:

$$R_0 = 15 - 1.5(SLR - N_0) \tag{2.2}$$

gde parametar *SLR* (*Send Loudnes Rating*) predstavlja gubitak glasnosti govornog signala na predajnoj strani tj. gubitak glasnosti govornog signala od korisnika (njegovih usta) do električnog interfejsa. Pod električnim interfejsom se podrazumeva mesto gde je govor digitalizovan. ParametarN_0 zbirni uticaj šumova na linijama u slučaju analognog prenosa, šuma okoline na predaji, šuma okoline i svih drugih izvora šumova.

Takođe jedan od parametara koji su definisani u formuli za R ocenu kvaliteta govorne veze je I_s i izračunava se na sledeći način:

$$I_s = I_{olr} + I_{st} + I_q \tag{2.3}$$

gde $\boldsymbol{I_{olr}}$ predstavlja uticaj neoptimalnsti glasnosti signala govora, $\boldsymbol{I_{st}}$ predstavlja uticaj neoptimalnosti lokalnog eha, $\boldsymbol{I_q}$ predstavlja uticaj šuma kvantizacije A/D konverzije. Precizan proračun ova tri parametra dat je u ITU-T G,107 preporuci.

Dalje, parametar I_d se proračunava:

$$I_d = I_{dte} + I_{dle} + I_{dd} \tag{2.4}$$

I_{dte} predstavlja uticaj eha govornika, I_{dle} predstavlja utiaceha slušaoca i I_{dd} predstavlja uticaj ukupnog kašnjenja u jednom smeru. Takođe i za tri prethodno navedena parametra dat je proračun u ITU-T G.107 preporuci.

Kašnjenje u jednom smeru u slučaju paketskih mreža zavisi od mnogo činilaca poput obrade govora na predajnoj strani, vremena paketizacije na predajnoj strani, vremena čekanja za slanje paketa ako se i drugi paketi šalju iz izvorišnog čvora, vremena propagacije kroz linkove, vremena obrade u mrežnim čvorovima, vremena čekanja u izlaznim čvorovima i dr. U proračun se tipično uzima maksimalno kašnjenje kroz paketsku mrežu, kao najgori slučaj. Većina kodera uzima odsečak govornog signala na obradu pa je otuda kašnjenje koje koder unosi $2T_0 + T_{la}$, gde je T_0 trajanje jednog odsečka, dok je T_{la} je *look - ahead* vreme. $2T_0$ se dobija iz razloga što je potrebno uzeti jedan odsečak (T_0) i zatim taj isti odsečak obraditi (T_0), pri čemu neki koderi moraju imati uvid i u naredni odsečak (T_{la}). Ukupno kašnjenje koje koder unosi se još naziva i akumulirano kašnjenje. Algoritamsko kašnjenje predstavlja vreme obrade odsečka uvećano za vreme dela narednog odsečka koje je neophodno za proces obrade ($T_0 + T_{la}$).

Parametar I_e se odnosi na degradaciju kvaliteta koju unosi sam koder u uslovima kada nema gubitaka paketa. Naravno pri postojanju gubitaka paketa dolazi do dodatne degradacije kvaliteta. U preporuci ITU-T G.107 se za slučaj postojanja gubitaka paketa definiše vrednost I_{e-eff} koja se koristi u izračunavanju R ocene u (2.2) umesto I_e i izraz za izračunavanje I_{e-eff} je:

$$I_{e-eff} = I_e + (95 - I_e)\frac{P_{pl}}{B_{pl} + {P_{pl}}/{Burst_R}} \qquad 2.5$$

gde je P_{pl} verovatnoća gubitaka paketa, B_{pl} faktor otpornosti kodera na gubitke paketa, a $Burst_R$ predstavlja koeficijent sporadičnosti (gde vrednost 1 odgovara slučaju da su gubici potpuno slučajni, a vrednosti veće od 1 označavaju gubici koji se dešavaju u grupama gde što veći broj predstavlja da se gubici dešavaju većim tj. dužim burstovima).

Parametar *A* jedini može pozitivno uticati na konačnu ocenu R. Korisnici daju bolje ocene kada su im očekivanja u pogeldu govorne veze lošija. U Tabeli 1 je moguće videti neke vrednosti paramtra I_e kao i algoritamskog kašnjenja za različite tipove kodera.

Koder	**Protok[kb/s]**	I_e	**Algoritamsko kašnjenje [ms]**
G.711	64	0	0.125
G.726	40	2	0.125
G.726	32	7	0.125
G.726	24	25	0.125
G.726	16	50	0.125
G.728	16	7	0.625
G.728	12.8	20	0.625
G.729	8	10	10
G.723.1	6.3	15	30
G.723.1	5.3	19	30
GSM 06.10	13	20	20
GSM 06.20	5.6	23	20
GSM 06.30	12.2	5	20

Tabela1 – Prikaz karakteristika poznatih kodera

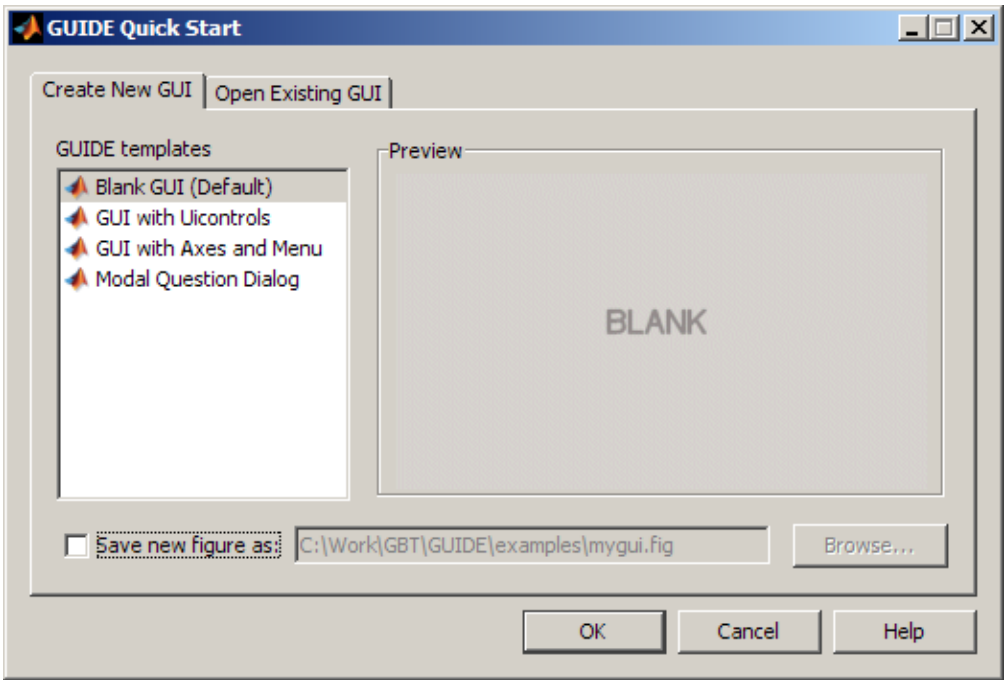
Napomena: najveći deo ovog poglavlja je preuzet iz materijala za predavanja predmeta “Komutacioni sistemi” [2].

3. OPIS APLIKACIJE I KODA

Aplikacija je izrađena da vrši proračun R ocene kvaliteta govorne veze po E modelu. Napravljena je pomoću programskog paketa *Matlab*. *Matlab* nudi dve opcije za kreiranje GUI-ja (*Graphical User Interface*). Prva opcija za kreiranje programa jeste GUIDE (*Graphical User Interface Development Environment*). GUIDE je sam po sebi jedna vrsta GUI-ja u okviru *Matlab*-a koja je tu sa ciljem da pojednostavi korisniku kreiranje željenog GUI-ja i sa par jednostavnih klikova omogući ubacivanje, izmenu, pozicioniranje i sl. željenih elemenata u okviru aplikacije. Druga opcija je da se GUI kreira programski tj. klasično, da se pišu linije koda i da se tako formira željeni GUI i svi elementi u samoj aplikaciji.

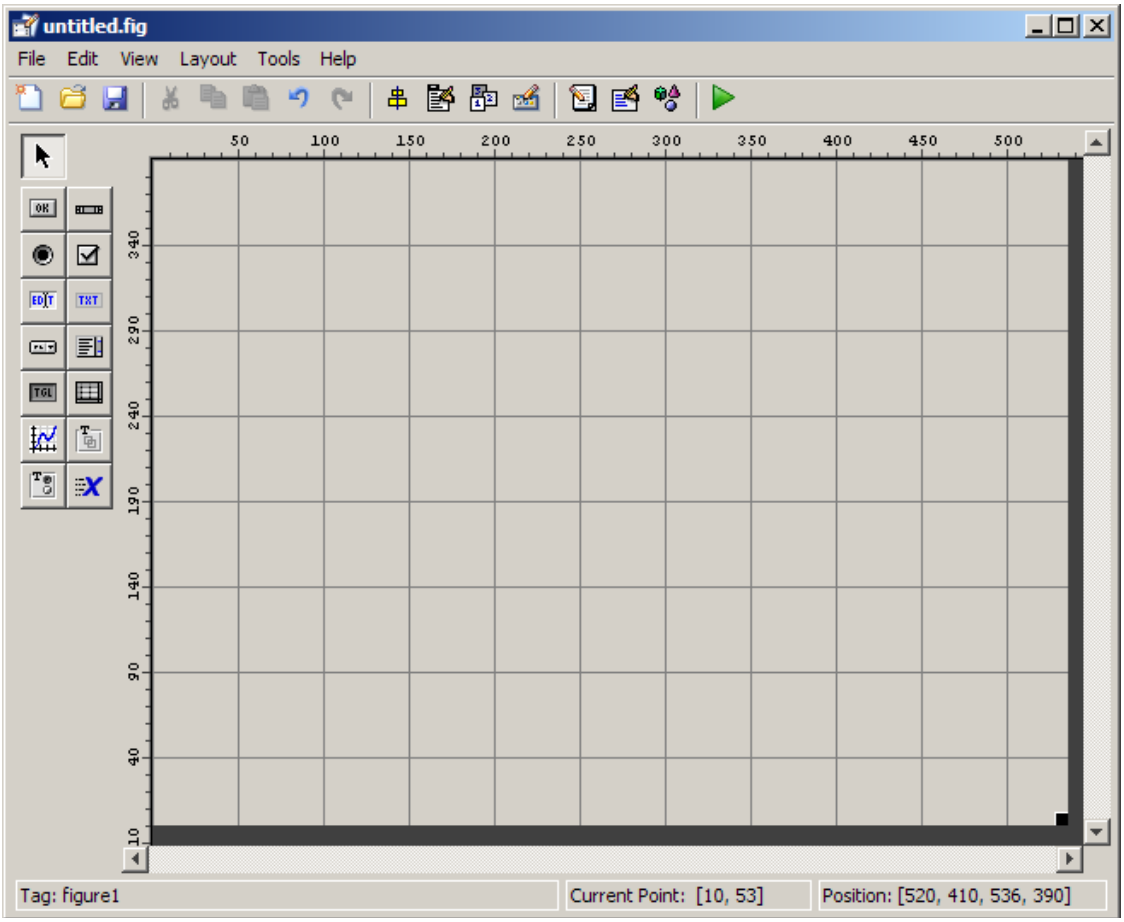
Odlučili smo se za prvu opciju, kreiraćemo aplikaciju korišćenjem GUIDE-a. GUIDE omogućava korisniku momentalan uvid u grafički izgled same aplikacije i samim tim skraćuje vreme potrebno za izradu aplikacije u odnosu na vreme koje bi bilo potrebno kada bi koristili programsko kreiranje aplikacije. Korišćenjem programskog kreiranja bi morali da definišemo koordinate glavnog prozora aplikacije, pa zatim i koordinate svakog elementa pojedinačno i morali bi konstatno da pokrećemo kod kako bi dobili uvid u to gde se nalaze elementi aplikacije.

GUIDE se može pokrenuti unosom pomoću tastature fraze*guide* u glavnom prozoru *Matlab*-a ili klikom na ikonicu GUIDE u *toolbar*-u glavnog prozora *Matlab*-a. Nakon toga se otvara GUIDE *quick start* prozor prikazan na slici 3.1. GUIDE *quick start* nam omogućava sledeće tri opcije: otvaranje novog GUI projekta, nekog predefinisanog projekta ili ranije sačuvanog projekta od strane korisnika.



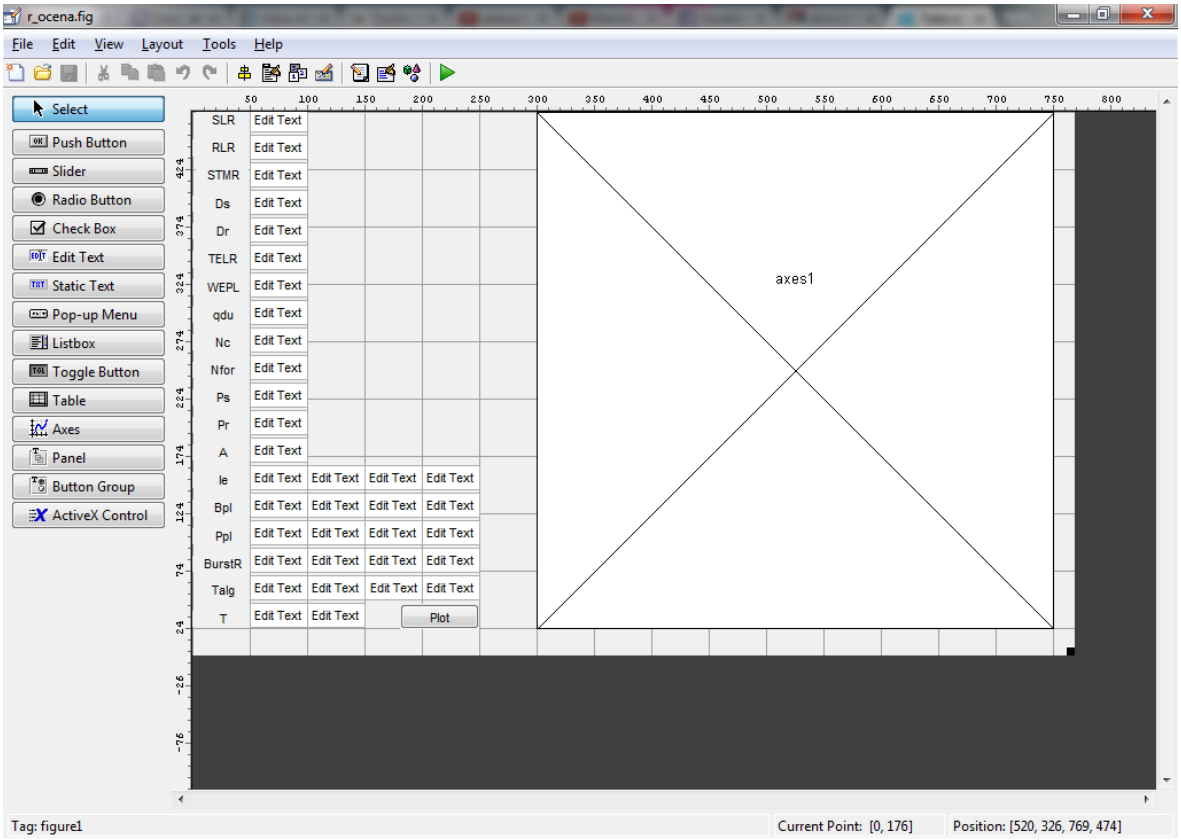
Slika 3.1 – GUIDE *quick start* prozor

Nakon izbora opcije *Blank GUI* otvara se novi prozor kao na slici 3.2 koji predstavlja prazan GUI. Promena veličine aktivnog dela GUI-ja se obavlja klikom na na donju ili gornju ivicu aktivnog dela ili klikom na donji desni ugao u kome se može primeiti mali crni kvadratić. Približavanjem kursora na navedene ivice ili ugao pojaviće se strelice koje nam označavaju da možemo da kliknemo i promenimo dimenzije aktivnog dela GUI-ja. Sa leve strane prozora koji smo dobili, kao na slici 3.2, se vidi lista elemenata koje možemo da ubacimo u naš GUI. Ubacivanje komponenata je jako jednostavno. Potrebno je samo kliknuti na željeni elementi zatim kliknuti u aktivnom delu prozora gde želimo da nam se dotični element nalazi.



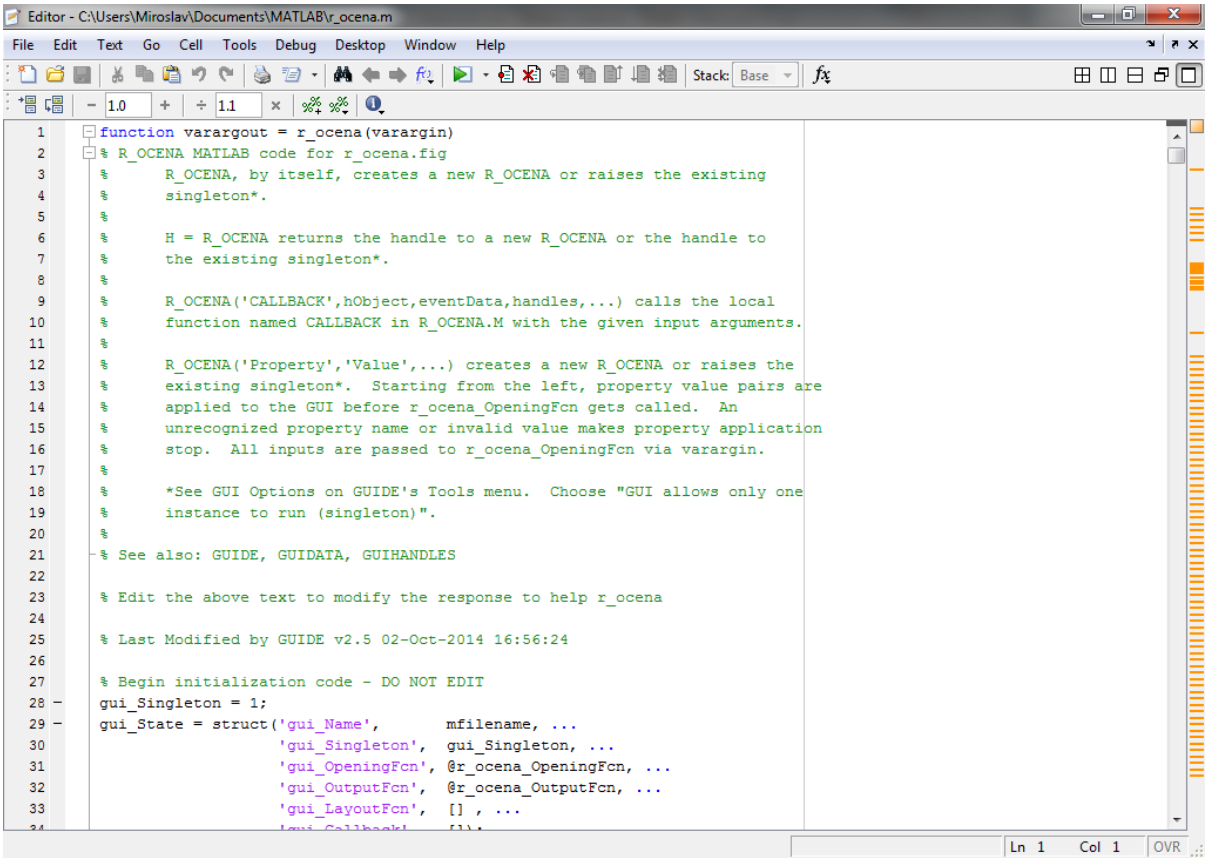
Slika 3.2 – *Blank* GUI prozor

Pri kreiranju ove aplikacije korišćene su četiri različite komponente. Te komponete su: *static text*, *edit text*, *push button* i *axes*. Komponente su kreirane kao strukture koje u sebi sadrže različite informacije. Za nas će biti najbitniji delovi strukture koju su obeleženi sa *String*, *Value* i *Tag*. *Static text* kao što i samo ime kaže je statična komponenta - jednom uneti podaci pri kreiranju GUI-ja se kasnije više ne mogu menjati. U našem slučaju smo *static text* komponetu koristili za obeležavanje *edit text* polja (praktično labele *edit text* polja) kako bi korisnicima bilo jasnije u koje polje je potrebno uneti podatke za određenu promenljivu potrebnu za proračun R ocene.



Slika 3.3 – GUIDE sa unetim komponentama

Na slici 3.3 se može videti GUIDE sa svim unetim komponentama potrebnim za ovu aplikaciju. Može se primetiti da aplikacija veoma podseća na kalkulator R ocene kvaliteta govorne veze koji se može naći na ITU-T sajtu [1]. Razlika je u tome što aplikacija realizovana u ovoj tezi ima dodatne mogućnosti. Postoji mogućnost iscrtavanja grafika R ocene kvaliteta u zavisnosti od vremenskog kašnjenja paketa pri prenosu kroz paketsku mrežu. Takođe se može primetiti da postoje po četiri polja za parametre I_e, P_{pl}, B_{pl}, $Burst_R$ i T_{alg} (vremensko kašnjenje koje unosi koder) i dva polja za parametar T (vremensko kašnjenje usled propagacije paketa kroz paketsku mrežu).

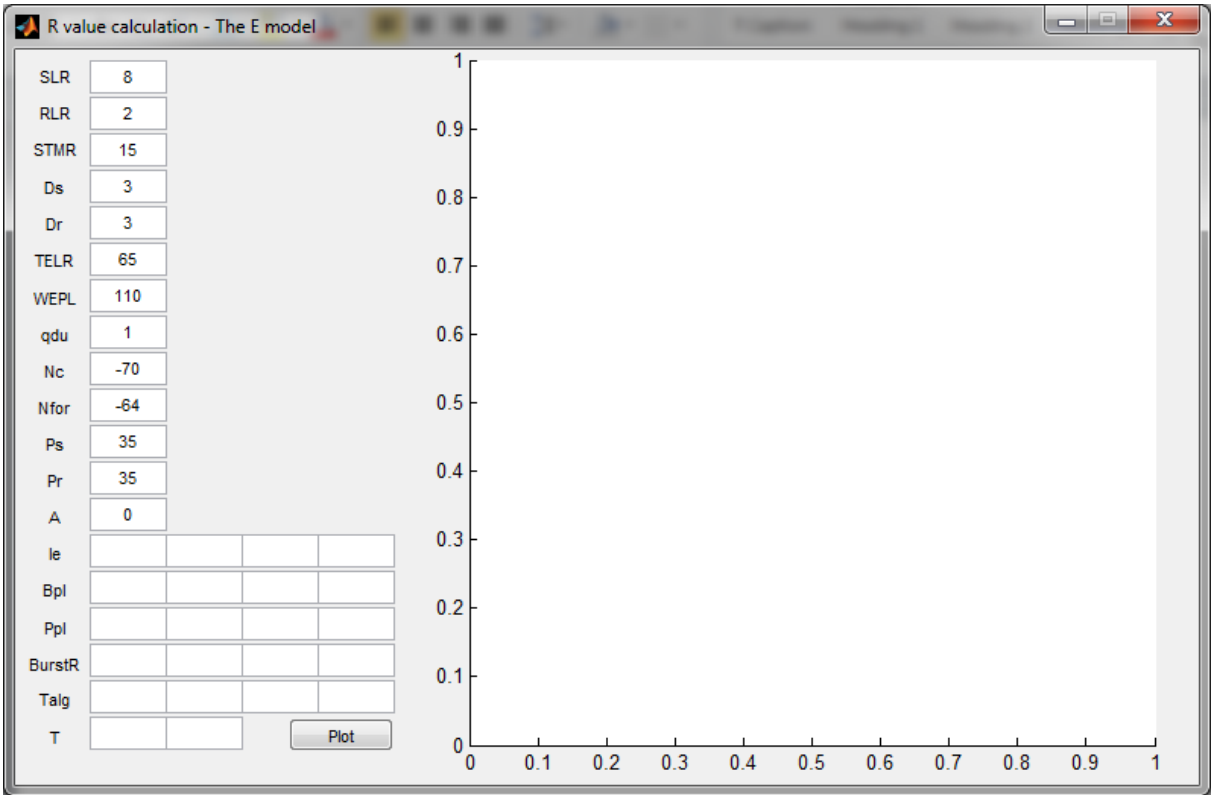


Slika 3.4 Editor sa kodom “iza” GUI-ja

Dalje, kada završimo sa unosom svih komponenti u GUIDE-u pokrećemo konfiguraciju sa klikom na zelenu (*play*) strelicu koja označava pokretanje konfiguracije (*Run Figure*) ili to možemo obaviti pritiskom *Ctrl + T* tastera na tastaturi. Nakon pokretanja nam se otvaraju dva nova prozora, a to su *Editor* u kome se nalazi kod ‘’iza’’ GUI-ja i otvara se sam GUI kao što je prikazano na slikama 3.4 i 3.5, respektivno.

Matlab za svaku komponentu, koja je bila ubačena u procesu kreiranja GUI-ja u GUIDE-u, kreira po dve funkcije. Jedna od te dve funkcije služi za kreiranje samog polja te komponente u GUI-ju. Tačnije obezbeđuje da komponenta koju smo ubacili u GUIDE-u bude kasnije grafički prikazana u GUI-ju definišući poziciju date komponente, kao i njen izgled (boju, oblik i sl.) i naziva se *tag_CreateFcn* gde *tag* predstavlja vrednost *tag* polja u strukturi za datu komponentu. Druga funkcija je funkcija koja se poziva, ako ima potrebe za tim, prilikom izvršavanja koda koja je dodeljena datoj komponenti i u koju smeštamo kod koji želimo pritom da se izvrši. Ova funkcija je obeležena sa *tag_Callback.*

Pored funkcija koje *Matlab* stvara za svaku komponentu pojedninačno, prave se još tri glavne funkcije. To su funkcije:*ime_fajla* (istoimena funkcija kao i *.m fajl koji smo sačuvali i pomoću koga ćemo kasnije pokretati GUI), *ime_fajla_OpeningFcn* u kojoj se nalazi kod koji se izvršava pri inicijalizaciji GUI-ja, a pre samog grafičkog prikaza i *ime_fajla_OutputFcn* u kojoj je definisano gde se ispisuju izlazni podaci, ako ih ima, tj. na komandnu liniju glavnog prozora *Matlab*-a.



Slika 3.5 – Realizovani GUI

Povezivanje komponente koja se nalazi u GUI-ju sa kodom koji treba da se izvrši ''iza'' nje ostvaruje se funkcijom *tag_Callback* funkcijom, gde *tag* predsavlje vrednost *tag* polja u strukturi date komponente.*Tag* polju date komponente pristupamo i menjamo desnim klikom miša na željenu komponentu koja je prethodno ubačena u GUIDE-u i zatim levim klikom na opciju *Property Inspector*. Ovim pristupamo detaljnom izgledu strukture date komponente GUI-ja. Menjanje vrednosti *tag* polja nam omogućava jednostavnije snalaženje i izradu samog koda.

Klikom na *Plotpush button* poziva se *pushbutton1_Callback* funkcija kojom se izvršava kod i iscrtavaju se potrebni grafici na *axes* komponenti GUI-ja.*Pushbutton1_Callback* funkcija predstavlja glavnu funkciju u kojoj se izračunava sve potrebno za dobijanje krajnjeg rezultata. Pokretanjem *pushbutton1_Callback* funkcije prvo uzimamo sve vrednsti koje su upisane u *edit text* polja i smeštamo ih u odgovarajuće promenljive. Naredbom *get(handles.tag, 'String')* uzimamo vrednost *string* polja komponente koja je obeležena sa *tag*. Pošto sada imamo vrednost polja kao *string* moramo je konvertovati u vrednost pogodnu za rad. To činimo naradbom *str2double* koja konvertuje string koji reprezentovan*ASCII* karakterima u *double.* Naravno, svaku od ovih vrednosti *string* polja smeštamo u odgovarajuću promenljivu.

Dalje imamo par provera vrednosti polja. Prva proverava da nisu vrednosti *T* polja nekim slučajem zamenjene tj. da minimalna vrednost nije stavljena u desno polje, a maksimalna u levo. Druga proverava da li su sva četiri polja I_e, P_{pl}, B_{pl}, $Burst_R$ i T_{alg} popunjena. Zatim se gleda koja ima najviše popunjenih polja i uzima se kao referentna. To znači sledeće, ako je maksimalan broj popunjenih polja *n* (gde je maksimalna vrednost *n* = 4) toliko treba i grafika da bude iscrtano. Ako neko od polja ima manje popunjenih polja od referentne vrednosti*n* tada se uzima vrednost poslednjeg popunjenog polja i kopira se u naredne, sve dok se ne popuni *n* polja. Kada smo to obavili prelazi se na proračun R ocene kvaliteta veze za zadate vrednosti parametara. Kao što je već

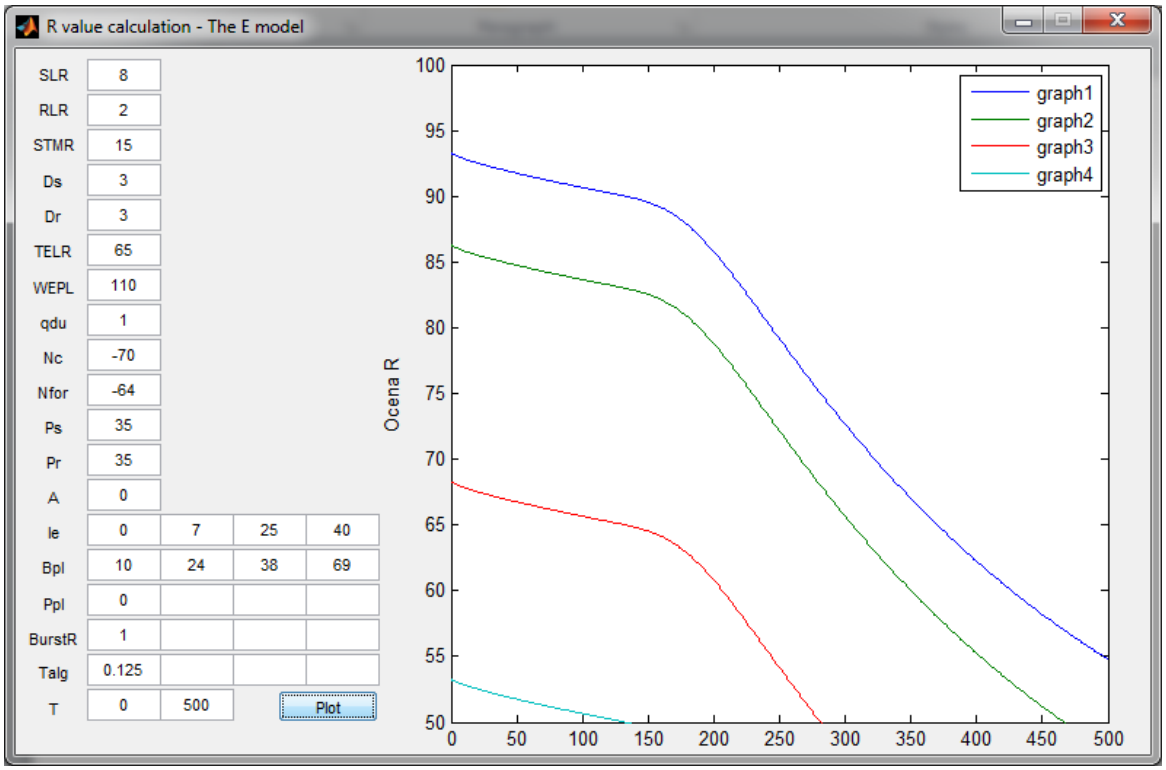
rečeno R ocenu kvaliteta se računa po ITU-T G.107 standardu i po formulama koje taj standard propisuje.

4. PRIKAZ RADA APLIKACIJE

Pri pokretanju kreirane *Matlab* funkcije dobijamo prozor kao na slici 3.5. Na levoj strani GUI-ja se nalaze parametri koji su relevantni za izračunavanje R ocene kvaliteta, dok se sa desne strane nalazi grafik na kome se iscrtavaju željene krive. Pri dnu GUI-ja se nalazi *Plot* taster kojim se inicijalizuje proračun i iscrtavanje grafika nakon unosa parametara.

Kao što se može videti u nekim poljima se već nalaze vrednosti. Te vrednosti predstavljaju difolt vrednosti koje se odnose na lokalnu ISDN vezu i definisane ITU-T G.107 preporukom. Lokalna ISDN veza se smatra najkvalitetnijom vezom. Preostala su I_e, P_{pl}, B_{pl}, $Burst_R$, T_{alg} i T polja koja su ostavljena korisniku da ih popuni.

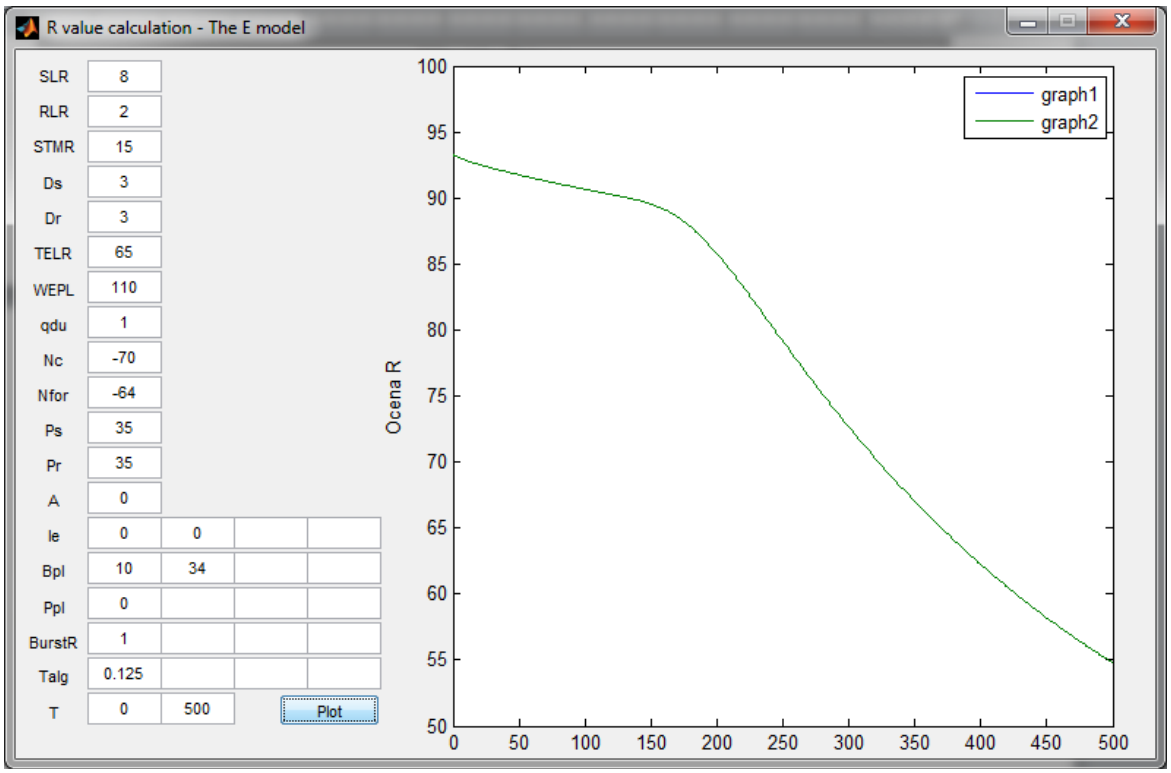
Vrednosti parametara I_e i T_{alg} direktno predstavljaju o kom tačno koderu se radi, dok parametri P_{pl}, B_{pl} i $Burst_R$ predstavljaju osobine mreže koje se odnose na gubitke paketa pri prenosu. Parametar T se odnosi na kašnjenje paketa pri prenosu kroz mrežu i u zavisnosti od njega iscrtavamo grafik koji se nalazi s desne strane GUI-ja tj. vrednosti polja parametra T predstavljaju minimalnu i maksimalnu vrednost x – ose. Kao što je već rečeno do sada, nakon unosa svih parametara pritiskom na taster *Plot* vrši se proračun R ocene u zavisnosti od kašnjenja kroz mrežu i iscrtava se grafik.



Slika 4.1 – Primer 1

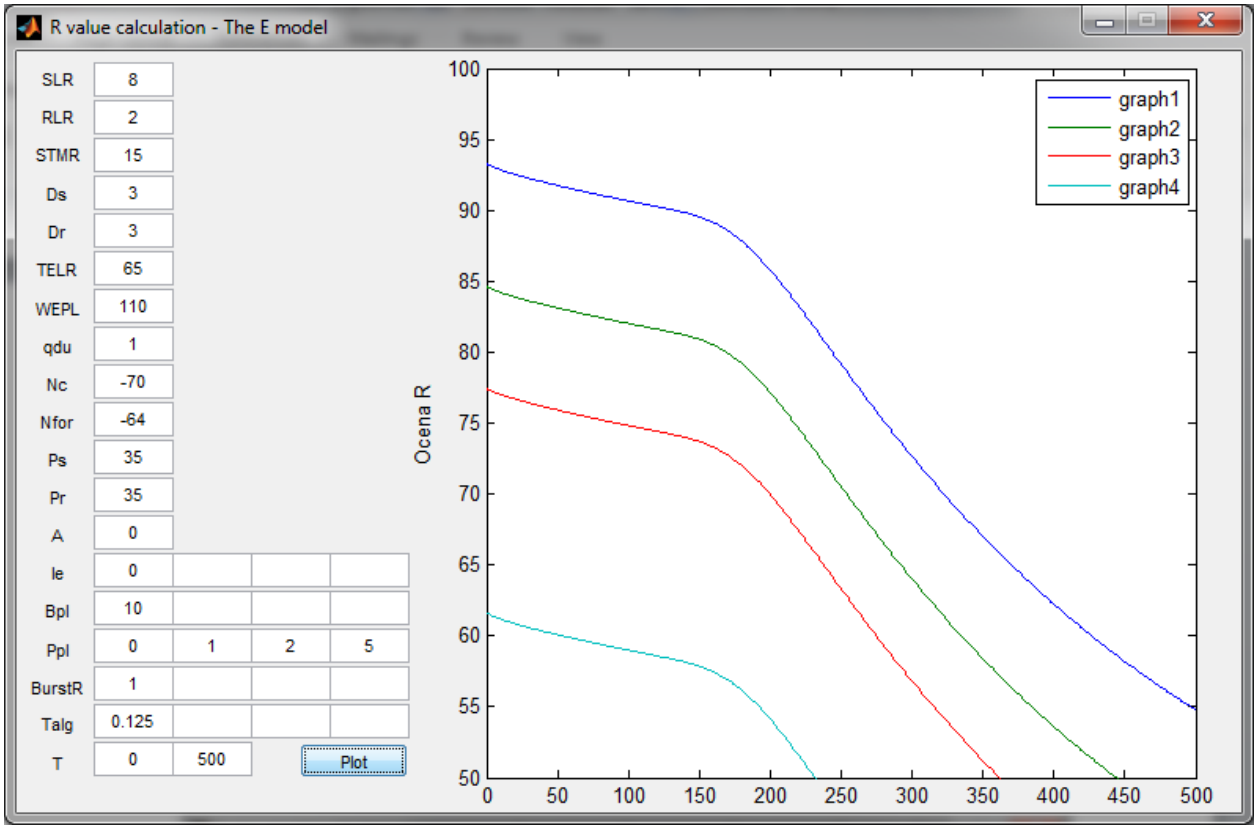
Unećemo sada vrednosti parametara za neke kodere i različite vrednosti parametara mreže i videti kako oni utiču na kvalitet veze. Možemo na slici 4.1 videti četiri primera. Uzeli smo pretpostavku da u mreži nema gubitaka paketa i upoređujemo uticajsamih kodera. *Graph1* predstavlja lokalnu ISDN vezu u kojoj se koristi koder G.711, dok *graph2*, *graph3* i *graph4* predstavljaju koder G.726 koji radi na brzinama 40 Kb/s, 24 kb/s i 16 kb/s respektivno. Algoritamsko kašnjenje oba kodera je isto i iznosi 0.125 ms. Može se videti da su za veće vrednosti I_e (degradacija kvaliteta koju unosi sam koder u uslovima kada nema gubitaka paketa) dobijeni lošiji rezultati što je i očekivano.

Na sledećem primeru 2, prikazanom na slici 4.2 dobijamo sledeći rezultat. Dve dobijene krive, tačnije moglo bi se reći jedna jer se dobijaju identnične krive koje se preklapaju, su dobijene za različite vrednosti parametra B_{pl}. To nam govori da dati parametar ne utiče na kvalitet što je i logično jer nema gubitaka paketa u mreži ($P_{pl} = 0$) i ako se pogleda formula (2.5) može se videti da parametar B_{pl} nema uticaj što je i potvrđeno radom realizovane aplikacije. Isto važi i za primer 1, jer je i tamo podešeno da nema gubitaka paketa. *Graph1* i *graph2* predstavljaju R ocenu kvaliteta kodera G.711 i G.711 sa PLC – om, respektivno.

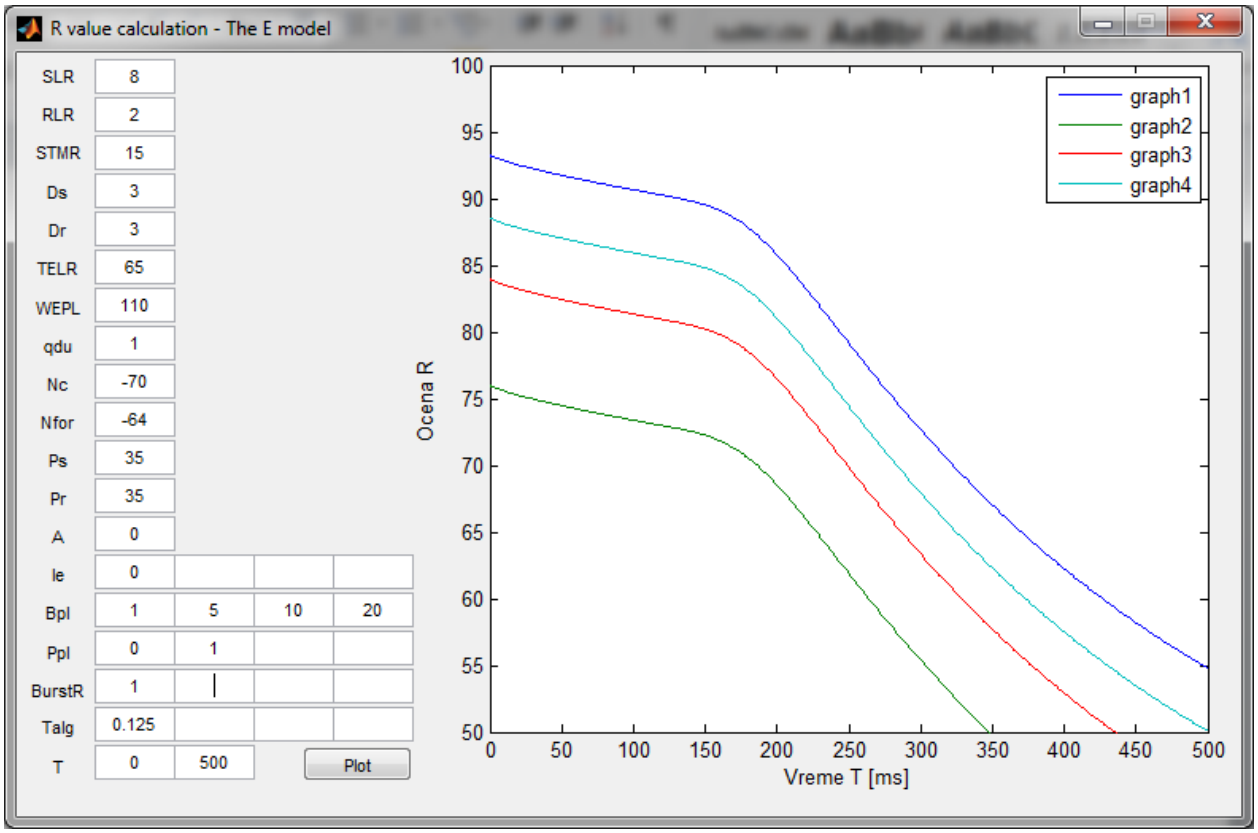


Slika 4.2 – Primer 2

Na primeru 3, slici 4.3 prikazan je uticaj P_{pl} (verovatnoća gubitaka paketa) na R ocenu kvaliteta veze. Dobijeni rezultati su očekivani, za povećanje verovatoće gubitaka paketa smanjuje se i ocena R. Sa slike se može primetiti da se za jako male promene vrednosti datog parametra ocena R drastično menja vrednost.



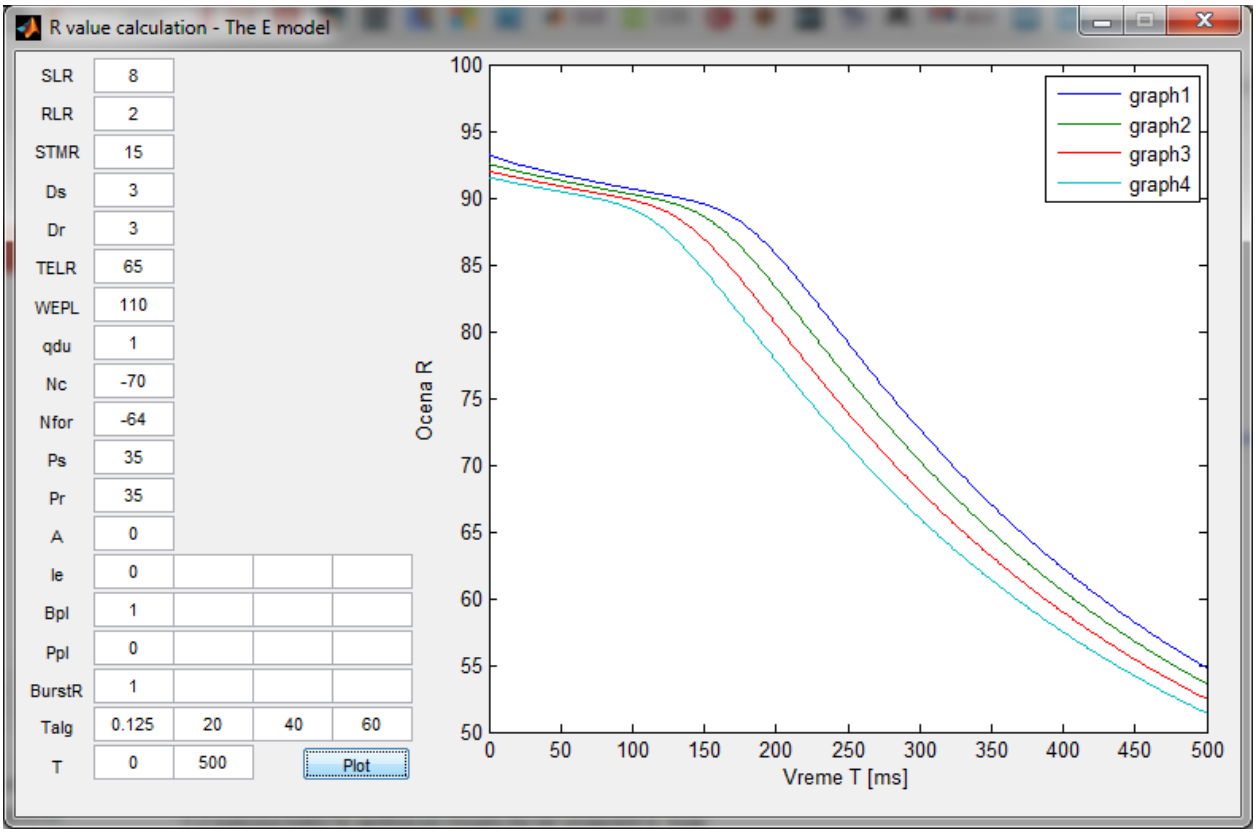
Slika 4.3 – Primer 3



Slika 4.4 – Primer 4

Na slici 4.4 može se videti četvrti primer na kome je prikazan uticaj parametra B_{pl} na ocenu kvaliteta R. Vidimo da povećanje datog parametra pozitivno utiče na ocenu kvaliteta R. Što je bilo i očekivano, jer ako ponovo pogledamo formulu (2.5) vidimo da se parametar B_{pl} nalazi ispod razlomačke crte i njegovim rastom smanjuje se uticaj pramatra I_{e-eff} na konačnu ocenu kvaliteta R tj. koder je otporniji na gubitak paketa, koji se dešava kada P_{pl} nije jednako nuli.

Na primeru 5, slici 4.5 možemo videti kako algoritamski kašnjenje kodera T_{alg} utiče na ocenu R. Dobijene krive su jako bliske. Preimetimo da za male vrednosti kašnjenja kroz paketsku mrežu T, do 150 ms, T_{alg} ne utiče toliko značajno na R ocenu kvaliteta, dok dalje na grafiku, za veće vrednosti T, razlika između krivih postaje primetnija. Vidi se da se za veće vrednosti T_{alg} dobijaju manje vrednosti ocene R i da kriva kreće ranije da opada.



Slika 4.4 – Primer 5

Verifikacija aplikacije obavljena je upoređivanjem vrednosti R ocene kvaliteta koja se dobija korišćenjem kalkulatora koji se može naći na sajtu ITU-T organizacije sa vrednostima R ocene koje su dobijale korišćenjem ove aplikacije. Takođe, sa priloženih primera se može primetiti da su dobijeni očekivani rezultati.

5. ZAKLJUČAK

U radu je realizovana aplikacija za izračunavanje R ocene kvaliteta veze. Takođe, u aplikacija omogućava iscrtavanje krive R ocene u zavisnosti od propagacionog kašnjenja paketa kroz mrežu. Naravno, grafik i sam proračun uzimaju u obzir relevantne parametre koje korisnik može da podešava. Aplikacija je namenjena za ispiticanje uticaja kašnjenja na R ocenu kvaliteta veze što je bilo pokazano i na primerima u prethodnom poglavlju. Takođe, pored uticaja kašnjenja jako je interesantno posmatrati zavisnost R ocene od verovatnoće paketskih grešaka, kao i uticaj otpornosti kodera na greške.

Ova aplikacija je pogodna za ispitivanje parametara paketske mreže i može nam pomoći u planiranju paketske mreže preko koje bi se prenosio paketizovani govorni signal. To jeste i suština R ocene kvaliteta, da se na lak i jednostavan način simulira paketska mreža i tako ispita da li data mreže zadovoljava određene kriterijume i da li je pogodna za puštanje u rad.

Aplikacija bi mogla biti unapređena dodavanjem polja za unos vrednosti onim parametrima koji imaju za sada samo jedno polje i tako omogući iscrtavanje različitih grafika za različitie vrednosti svakog parametra tj. šire ispitivanje uticaja parametara koji utiču na kvalitet govornog signala. Mogla bi se napraviti predefinisana baza parametara za svaki koder i omogućiti iscrtavanje ocene R sa samo par klikova. Potencijalno bi dobro dodati opciju sličnoj *Data Cursor* opciji koja postoji u *Figure* prozoru koji se javlja pri klasičnom plotovanju u *Matlab*-u. *Data Cursor* opcija bi nam omogućila jako precizna i brza očitavanja vrednosti R ocene kvaliteta sa grafika.

LITERATURA

[1] R value calculator - the E model,

www.itu.int/ITU-T/studygroups/com12/emodelv1/calcul.php

[2] Materijali sa predavanja predmeta Komutacioni sistemi

[3] *MATLAB Creating Graphical User Interfaces*, The MathWorks Inc., http://www.mathworks.com/help/pdf_doc/matlab/buildgui.pd

A. KOD APLIKACIJE

```
functionvarargout = r_ocena(varargin)
% R_OCENA MATLAB code for r_ocena.fig
%      R_OCENA, by itself, creates a new R_OCENA or raises the existing
%      singleton*.
%
%      H = R_OCENA returns the handle to a new R_OCENA or the handle to
%      the existing singleton*.
%
%      R_OCENA('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in R_OCENA.M with the given input arguments.
%
%      R_OCENA('Property','Value',...) creates a new R_OCENA or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before r_ocena_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to r_ocena_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help r_ocena

% Last Modified by GUIDE v2.5 02-Oct-2014 16:56:24

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @r_ocena_OpeningFcn, ...
'gui_OutputFcn',  @r_ocena_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
ifnargin&&ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end

ifnargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before r_ocena is made visible.
```

```
functionr_ocena_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to r_ocena (see VARARGIN)


% Postavljanjedifoltvrednosti u polja GUI-ja
set(handles.SLR,'String','8');
set(handles.RLR,'String','2');
set(handles.STMR,'String','15');
set(handles.Ds,'String','3');
set(handles.Dr,'String','3');
set(handles.TELR,'String','65');
set(handles.WEPL,'String','110');
set(handles.Nc,'String','-70');
set(handles.Nfor,'String','-64');
set(handles.Ps,'String','35');
set(handles.Pr,'String','35');
set(handles.qdu,'String','1');
set(handles.A,'String','0');
% set(handles.SLR,'String','');
% set(handles.RLR,'String','');
% set(handles.STMR,'String','');
% set(handles.Ds,'String','');
% set(handles.Dr,'String','');
% set(handles.TELR,'String','');
% set(handles.WEPL,'String','');
% set(handles.Nc,'String','');
% set(handles.Nfor,'String','');
% set(handles.Ps,'String','');
% set(handles.Pr,'String','');
% set(handles.qdu,'String','');
% set(handles.A,'String','');
set(handles.Ie1,'String','');
set(handles.Ie2,'String','');
set(handles.Ie3,'String','');
set(handles.Ie4,'String','');
set(handles.Bpl1,'String','');
set(handles.Bpl2,'String','');
set(handles.Bpl3,'String','');
set(handles.Bpl4,'String','');
set(handles.Ppl1,'String','');
set(handles.Ppl2,'String','');
set(handles.Ppl3,'String','');
set(handles.Ppl4,'String','');
set(handles.BurstR1,'String','');
set(handles.BurstR2,'String','');
set(handles.BurstR3,'String','');
set(handles.BurstR4,'String','');
set(handles.Talg1,'String','');
set(handles.Talg2,'String','');
set(handles.Talg3,'String','');
set(handles.Talg4,'String','');
set(handles.Tmin,'String','');
set(handles.Tmax,'String','');
```

```
% Choose default command line output for r_ocena
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes r_ocena wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
functionvarargout = r_ocena_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



functionSLR_Callback(hObject, eventdata, handles)
% hObject    handle to SLR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of SLR as text
%        str2double(get(hObject,'String')) returns contents of SLR as a double


% --- Executes during object creation, after setting all properties.
functionSLR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SLR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



functionRLR_Callback(hObject, eventdata, handles)
% hObject    handle to RLR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RLR as text
%        str2double(get(hObject,'String')) returns contents of RLR as a double
```

```
% --- Executes during object creation, after setting all properties.
functionRLR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RLR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc
SLR = str2double(get(handles.SLR,'String'));
RLR = str2double(get(handles.RLR,'String'));
STMR = str2double(get(handles.STMR,'String'));
Ds = str2double(get(handles.Ds,'String'));
Dr = str2double(get(handles.Dr,'String'));
TELR = str2double(get(handles.TELR,'String'));
WEPL = str2double(get(handles.WEPL,'String'));
qdu = str2double(get(handles.qdu,'String'));
Nc = str2double(get(handles.Nc,'String'));
Nfor = str2double(get(handles.Nfor,'String'));
Ps = str2double(get(handles.Ps,'String'));
Pr = str2double(get(handles.Pr,'String'));
A = str2double(get(handles.A,'String'));
LSTR = STMR+Dr;

Tmin = str2double(get(handles.Tmin,'String'));
Tmax = str2double(get(handles.Tmax,'String'));
% Provera vrednosti T polja, da nisuslucajnozamenjene
ifTmin>Tmax
      a = Tmin;
      Tmin = Tmax;
      Tmax = a;
end

testIe = [str2double(get(handles.Ie1,'String'))
str2double(get(handles.Ie2,'String')) str2double(get(handles.Ie3,'String'))
str2double(get(handles.Ie4,'String'))];
testBpl = [str2double(get(handles.Bpl1,'String'))
str2double(get(handles.Bpl2,'String')) str2double(get(handles.Bpl3,'String'))
str2double(get(handles.Bpl4,'String'))];
testPpl = [str2double(get(handles.Ppl1,'String'))
str2double(get(handles.Ppl2,'String')) str2double(get(handles.Ppl3,'String'))
str2double(get(handles.Ppl4,'String'))];
testBurstR = [str2double(get(handles.BurstR1,'String'))
str2double(get(handles.BurstR2,'String'))
str2double(get(handles.BurstR3,'String'))
str2double(get(handles.BurstR4,'String'))];
```

```
testTalg = [str2double(get(handles.Talg1,'String'))
str2double(get(handles.Talg2,'String')) str2double(get(handles.Talg3,'String'))
str2double(get(handles.Talg4,'String'))];

j1=1;
j2=1;
j3=1;
j4=1;
j5=1;

fori = 1:4
if ~isnan(testIe(i)) && j1 <= 4
Ie(j1) = testIe(i);
            j1=j1+1;
end
if ~isnan(testBpl(i)) && j2 <= 4
Bpl(j2) = testBpl(i);
            j2=j2+1;
end
if ~isnan(testPpl(i)) && j3 <= 4
Ppl(j3) = testPpl(i);
            j3=j3+1;
end
if ~isnan(testBurstR(i)) && j4 <= 4
BurstR(j4) = testBurstR(i);
            j4=j4+1;
end
if ~isnan(testTalg(i)) && j5 <= 4
Talg(j5) = testTalg(i);
            j5=j5+1;
end
end

C = max([numel(Ie) numel(Bpl) numel(Ppl) numel(BurstR) numel(Talg)]);

ifnumel(Ie) < C
      fori = (numel(Ie)+1):C
            Ie(i) = Ie(i-1);
      end
end
ifnumel(Bpl) < C
      fori = (numel(Bpl)+1):C
            Bpl(i) = Bpl(i-1);
      end
end
ifnumel(Ppl) < C
      fori = (numel(Ppl)+1):C
            Ppl(i) = Ppl(i-1);
      end
end
ifnumel(BurstR) < C
      fori = (numel(BurstR)+1):C
            BurstR(i) = BurstR(i-1);
      end
end
ifnumel(Talg) < C
      fori = (numel(Talg)+1):C
            Talg(i) = Talg(i-1);
```

```
      end
end


for j=1:C


forei = Tmin:1:Tmax
            T=i+Talg(j);
            Tr=2*T;
            Ta=T;

%Izracunavanje R0 faktora

Nfo=Nfor+RLR;
        Pre=Pr+10*log10(1+10^((10-LSTR)/10));
        Nor=RLR-121+Pre+0.008*(Pre-35)^2;
        OLR=RLR+SLR;
Nos=Ps-SLR-Ds-100+0.004*(Ps-OLR-Ds-14)^2;
        No=10*log10(10^(Nc/10)+10^(Nos/10)+10^(Nor/10)+10^(Nfo/10));
        Ro=15-1.5*(SLR+No);



%IzracunavanjeparametraIs

        Q=37-15*log10(qdu);
        G=1.07+0.258*Q+0.0602*Q^2;
        Z=46/30-G/40;
        Y=(Ro-100)/15+46/8.4-G/9;
Iq=15*log10(1+10^Y+10^Z);
STMRo=-10*log10(10^(-STMR/10)+exp(-T/4)*10^(-TELR/10));
        Ist=12*((1+((STMRo-13)/6)^8)^(1/8))-28*((1+((STMRo+1)/19.4)^35)^(1/35))-
13*((1+((STMRo-3)/33)^13)^(1/13))+29;
Xolr=OLR+0.2*(64+No-RLR);
Iolr=20*((1+(Xolr/8)^8)^(1/8)-Xolr/8);
        Is=Iolr+Ist+Iq;



%Izracunavanjeparametra Id

        TERV=TELR-40*log10((1+T/10)/(1+T/150))+6*exp(-0.3*T^2);

if STMR<9

                TERV=TERV+Ist/2;

end

        Re=80+2.5*(TERV-14);
        Roe=-1.5*(No-RLR);
Idte=((Roe-Re)/2+sqrt((Roe-Re)^2/4+100)-1)*(1-exp(-T));

if STMR>20
```

```
Idte=sqrt(Idte^2+Ist^2);
end

if T<1
Idte=0;
end

Rle = 10.5*(WEPL+7)*(Tr+1)^(-0.25);
        Idle = (Ro-Rle)/2+sqrt((Ro-Rle)^2/4+169);

if Ta<=100
Idd=0;
else
                X=log10(Ta/100)/log10(2);
Idd=25*((1+X^6)^(1/6)-3*(1+(X/3)^6)^(1/6)+2);
end

        Id=Idte+Idle+Idd;


ifBpl(j)==0 &&BurstR(j) == 0
Ieeff=Ie(j);
else
Ieeff=Ie(j)+(95-Ie(j))*(Ppl(j)/((Ppl(j)/BurstR(j))+Bpl(j)));
end
R(j,i+1-Tmin)=Ro-Is-Id-Ieeff+A;

end



end

% Iscrtavanjegrafika
switchnumel(R(:,1))
case 1
t = Tmin:1:Tmax;
      plot(handles.axes1,t,R)
      xlabel('Vreme T [ms]')
      ylabel('Ocena R')
      axis([TminTmax 50 100])
      legend('graph1')
case 2
t = Tmin:1:Tmax;
      plot(handles.axes1,t,R)
      xlabel('Vreme T [ms]')
      ylabel('Ocena R')
      axis([TminTmax 50 100])
      legend('graph1','graph2')
case 3
t = Tmin:1:Tmax;
      plot(handles.axes1,t,R)
      xlabel('Vreme T [ms]')
      ylabel('Ocena R')
      axis([TminTmax 50 100])
      legend('graph1','graph2','graph3')
case 4
```

```
t = Tmin:1:Tmax;
      plot(handles.axes1,t,R)
      xlabel('Vreme T [ms]')
      ylabel('Ocena R')
      axis([TminTmax 50 100])
      legend('graph1','graph2','graph3','graph4')
end



functionSTMR_Callback(hObject, eventdata, handles)
% hObject    handle to STMR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of STMR as text
%        str2double(get(hObject,'String')) returns contents of STMR as a double


% --- Executes during object creation, after setting all properties.
functionSTMR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to STMR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



functionDs_Callback(hObject, eventdata, handles)
% hObject    handle to Ds (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Ds as text
%        str2double(get(hObject,'String')) returns contents of Ds as a double


% --- Executes during object creation, after setting all properties.
functionDs_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Ds (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
functionDr_Callback(hObject, eventdata, handles)
% hObject    handle to Dr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Dr as text
%        str2double(get(hObject,'String')) returns contents of Dr as a double


% --- Executes during object creation, after setting all properties.
functionDr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Dr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



functionTELR_Callback(hObject, eventdata, handles)
% hObject    handle to TELR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TELR as text
%        str2double(get(hObject,'String')) returns contents of TELR as a double


% --- Executes during object creation, after setting all properties.
functionTELR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TELR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



functionWEPL_Callback(hObject, eventdata, handles)
% hObject    handle to WEPL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of WEPL as text
%        str2double(get(hObject,'String')) returns contents of WEPL as a double


% --- Executes during object creation, after setting all properties.
functionWEPL_CreateFcn(hObject, eventdata, handles)
% hObject    handle to WEPL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



functionTmin_Callback(hObject, eventdata, handles)
% hObject    handle to Tmin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Tmin as text
%        str2double(get(hObject,'String')) returns contents of Tmin as a double


% --- Executes during object creation, after setting all properties.
functionTmin_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Tmin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



functionTmax_Callback(hObject, eventdata, handles)
% hObject    handle to Tmax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Tmax as text
%        str2double(get(hObject,'String')) returns contents of Tmax as a double


% --- Executes during object creation, after setting all properties.
functionTmax_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Tmax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Ie2_Callback(hObject, eventdata, handles)
% hObject    handle to Ie2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Ie2 as text
%        str2double(get(hObject,'String')) returns contents of Ie2 as a double


% --- Executes during object creation, after setting all properties.
function Ie2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Ie2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



functionqdu_Callback(hObject, eventdata, handles)
% hObject    handle to qdu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of qdu as text
%        str2double(get(hObject,'String')) returns contents of qdu as a double


% --- Executes during object creation, after setting all properties.
functionqdu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to qdu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function Ie1_Callback(hObject, eventdata, handles)
% hObject    handle to Ie1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Ie1 as text
%        str2double(get(hObject,'String')) returns contents of Ie1 as a double


% --- Executes during object creation, after setting all properties.
function Ie1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Ie1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Bpl1_Callback(hObject, eventdata, handles)
% hObject    handle to Bpl1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Bpl1 as text
%        str2double(get(hObject,'String')) returns contents of Bpl1 as a double


% --- Executes during object creation, after setting all properties.
function Bpl1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Bpl1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Ppl1_Callback(hObject, eventdata, handles)
% hObject    handle to Ppl1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of Ppl1 as text
%        str2double(get(hObject,'String')) returns contents of Ppl1 as a double


% --- Executes during object creation, after setting all properties.
function Ppl1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Ppl1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function BurstR1_Callback(hObject, eventdata, handles)
% hObject    handle to BurstR1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of BurstR1 as text
%        str2double(get(hObject,'String')) returns contents of BurstR1 as a
double


% --- Executes during object creation, after setting all properties.
function BurstR1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to BurstR1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



functionNc_Callback(hObject, eventdata, handles)
% hObject    handle to Nc (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Nc as text
%        str2double(get(hObject,'String')) returns contents of Nc as a double


% --- Executes during object creation, after setting all properties.
```

```
functionNc_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Nc (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



functionNfor_Callback(hObject, eventdata, handles)
% hObject    handle to Nfor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Nfor as text
%        str2double(get(hObject,'String')) returns contents of Nfor as a double


% --- Executes during object creation, after setting all properties.
functionNfor_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Nfor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



functionPs_Callback(hObject, eventdata, handles)
% hObject    handle to Ps (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Ps as text
%        str2double(get(hObject,'String')) returns contents of Ps as a double


% --- Executes during object creation, after setting all properties.
functionPs_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Ps (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



functionPr_Callback(hObject, eventdata, handles)
% hObject    handle to Pr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Pr as text
%        str2double(get(hObject,'String')) returns contents of Pr as a double


% --- Executes during object creation, after setting all properties.
functionPr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Pr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



functionA_Callback(hObject, eventdata, handles)
% hObject    handle to A (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of A as text
%        str2double(get(hObject,'String')) returns contents of A as a double


% --- Executes during object creation, after setting all properties.
functionA_CreateFcn(hObject, eventdata, handles)
% hObject    handle to A (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function Ie3_Callback(hObject, eventdata, handles)
% hObject    handle to Ie3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Ie3 as text
%        str2double(get(hObject,'String')) returns contents of Ie3 as a double


% --- Executes during object creation, after setting all properties.
function Ie3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Ie3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Bpl2_Callback(hObject, eventdata, handles)
% hObject    handle to Bpl2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Bpl2 as text
%        str2double(get(hObject,'String')) returns contents of Bpl2 as a double


% --- Executes during object creation, after setting all properties.
function Bpl2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Bpl2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Bpl3_Callback(hObject, eventdata, handles)
% hObject    handle to Bpl3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Bpl3 as text
%        str2double(get(hObject,'String')) returns contents of Bpl3 as a double
```

```
% --- Executes during object creation, after setting all properties.
function Bpl3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Bpl3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Bpl4_Callback(hObject, eventdata, handles)
% hObject    handle to Bpl4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Bpl4 as text
%        str2double(get(hObject,'String')) returns contents of Bpl4 as a double


% --- Executes during object creation, after setting all properties.
function Bpl4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Bpl4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Ppl2_Callback(hObject, eventdata, handles)
% hObject    handle to Ppl2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Ppl2 as text
%        str2double(get(hObject,'String')) returns contents of Ppl2 as a double


% --- Executes during object creation, after setting all properties.
function Ppl2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Ppl2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Ppl3_Callback(hObject, eventdata, handles)
% hObject    handle to Ppl3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Ppl3 as text
%        str2double(get(hObject,'String')) returns contents of Ppl3 as a double


% --- Executes during object creation, after setting all properties.
function Ppl3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Ppl3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Ppl4_Callback(hObject, eventdata, handles)
% hObject    handle to Ppl4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Ppl4 as text
%        str2double(get(hObject,'String')) returns contents of Ppl4 as a double


% --- Executes during object creation, after setting all properties.
function Ppl4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Ppl4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function BurstR2_Callback(hObject, eventdata, handles)
% hObject    handle to BurstR2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of BurstR2 as text
%        str2double(get(hObject,'String')) returns contents of BurstR2 as a
double


% --- Executes during object creation, after setting all properties.
function BurstR2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to BurstR2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function BurstR3_Callback(hObject, eventdata, handles)
% hObject    handle to BurstR3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of BurstR3 as text
%        str2double(get(hObject,'String')) returns contents of BurstR3 as a
double


% --- Executes during object creation, after setting all properties.
function BurstR3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to BurstR3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function BurstR4_Callback(hObject, eventdata, handles)
% hObject    handle to BurstR4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of BurstR4 as text
%        str2double(get(hObject,'String')) returns contents of BurstR4 as a
double


% --- Executes during object creation, after setting all properties.
function BurstR4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to BurstR4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Ie4_Callback(hObject, eventdata, handles)
% hObject    handle to Ie4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Ie4 as text
%        str2double(get(hObject,'String')) returns contents of Ie4 as a double


% --- Executes during object creation, after setting all properties.
function Ie4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Ie4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Talg1_Callback(hObject, eventdata, handles)
% hObject    handle to Talg1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Talg1 as text
%        str2double(get(hObject,'String')) returns contents of Talg1 as a double


% --- Executes during object creation, after setting all properties.
function Talg1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Talg1 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Talg2_Callback(hObject, eventdata, handles)
% hObject    handle to Talg2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Talg2 as text
%        str2double(get(hObject,'String')) returns contents of Talg2 as a double


% --- Executes during object creation, after setting all properties.
function Talg2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Talg2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function Talg3_Callback(hObject, eventdata, handles)
% hObject    handle to Talg3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Talg3 as text
%        str2double(get(hObject,'String')) returns contents of Talg3 as a double


% --- Executes during object creation, after setting all properties.
function Talg3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Talg3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
```

```
end



function Talg4_Callback(hObject, eventdata, handles)
% hObject    handle to Talg4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Talg4 as text
%        str2double(get(hObject,'String')) returns contents of Talg4 as a double


% --- Executes during object creation, after setting all properties.
function Talg4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Talg4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```