

UNIVERZITET U SARAJEVU  
ELEKTROTEHNIČKI FAKULTET  
ODSJEK ZA RAČUNARSTVO I INFORMATIKU  
SPECIJALNA POGLAVLJA SISTEMSKOG SOFTVERA

Identifikacija objekta drugog reda sa transportnim kašnjenjem  
bazirana na prinudnim oscilacijama

Studenti:

Ibradžić Samir  
Ljubović Lejla  
Mehić Adis  
Musić Lejla

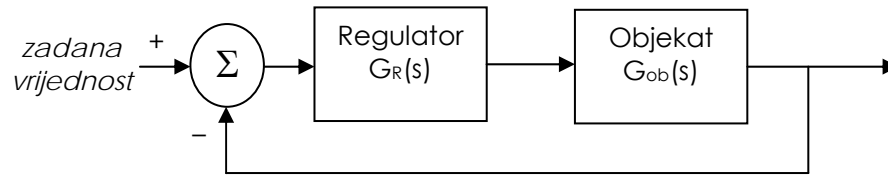
Sarajevo, juni 2005. godine

# I Sadržaj

|     |  |                                     |
|-----|--|-------------------------------------|
| I   | Sadržaj.....   | 2                                   |
| II  | Projektni zadatak.....                                 | 3                                   |
| III | Standardni regulacioni sistem – teoretske osnove ..... | 4                                   |
| IV  | Realizacija projektnog zadatka .....                   | 7                                   |
|     | Simulink.....  | 7                                   |
|     | Matlab.....  | 8                                   |
|     | InduSoft Web Studio.....                               | <b>Error! Bookmark not defined.</b> |
| A   | Kod Matlab M-fileova.....                              | 33                                  |
| B   | Varijable i tagovi .....                               | 37                                  |

## II Projektni zadatak

Koristeći IWS HMI paket za interfejs prema korisniku i MATLAB 7/SIMULINK 6, paket za simulaciju i modeliranje, polazeći od blok strukture standardnog feedback regulacionog sistema, prema sljedećoj shemi:



, i uz pretpostavku da je regulator standardni PID regulator sa prenosnom funkcijom:

$$G_R(s) = K_R \left( 1 + \frac{1}{T_i s} + T_d s \right)$$

realizovati aplikaciju „Identifikacija objekta drugog reda sa transportnim kašnjenjem bazirana na prinudnim oscilacijama“.

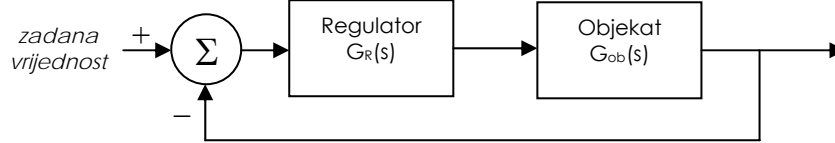
$K_R$ ,  $T_i$  i  $T_d$  su podesivi parametri regulatora, čija će se vrijednost zadavati sa ekrana IWS-InduSofta, kao i zadana vrijednost.

Zadana vrijednost, parametri podešenja regulatora kao i izlazne veličine iz regulacione konture će se prikazivati na ekranima IWS-a, te trendirati u real time i historijskim trend ekranima.

Model dinamičkog objekta i regulatora će se realizovati u SIMULINK-u. Komunikacija između MATLAB-a i IWS-a će se odvijati preko OPC klijent/server protokola.

### III Standardni regulacioni sistem – teoretske osnove

Neka imamo standardni regulacioni sistem, prema sljedećoj shemi:



Za regulator ćemo pretpostaviti da je standardni PID regulator sa prenosnom funkcijom:

$$G_R(s) = K_R \left( 1 + \frac{1}{T_i s} + T_d s \right)$$

gdje su  $K_R$ ,  $T_i$  i  $T_d$  podesivi parametri regulatora.

Pretpostavićemo da je dinamika objekta opisana prenosnom funkcijom drugog reda sa eventualnom čistim transportnim kašnjenjem, odnosno prenosnom funkcijom oblika:

$$G_{ob}(s) = \frac{K_{ob}}{T_{ob}^2 s^2 + 2 T_{ob} \xi_{ob} s + 1} e^{-T_{ob} s}$$

Prenosne funkcije velikog broja standardnih industrijskih objekata mogu se sa zadovoljavajućom tačnošću opisati ovakvom prenosnom funkcijom. Ovdje su  $K_{ob}$ ,  $T_{ob}$ ,  $\xi_{ob}$  i  $T_{ob}$  parametri objekta (respektivno pojačanje, vremenska konstanta, faktor prigušenja i iznos čistog kašnjenja) koji će možda biti fiksni, a možda će se i oni zadavati iz InduSofta. Uglavnom, tokom izvođenja eksperimenta ovi parametri se ne mijenjaju, tako da u svakom slučaju smatramo da su sa aspekta eksperimenta oni fiksni. Cilj eksperimenata je upravo da se eksperimentalno odrede vrijednosti ovih parametara.

Poznato je da za ovakvu regulacionu konturu i proizvoljne prenosne funkcije regulatora  $G_R(s)$  i objekta  $G_{ob}(s)$  prenosna funkcija čitave regulacione konture  $W(s)$  iznosi:

$$W(s) = \frac{G_R(s) G_{ob}(s)}{1 + G_R(s) G_{ob}(s)}$$

Kako prenosna funkcija regulatora zavisi od upravljivih parametara  $K_R$ ,  $T_i$  i  $T_d$ , možemo tvrditi da je prenosna funkcija regulatora  $G_R(s) = G_R(s, K_R, T_i, T_d)$ . Samim tim, i prenosna funkcija  $W(s)$  također ovisi od ovih parametara:

$$W(s) = W(s, K_R, T_i, T_d) = \frac{G_R(s, K_R, T_i, T_d) G_{ob}(s)}{1 + G_R(s, K_R, T_i, T_d) G_{ob}(s)}$$

Pretpostavimo sada da smo, eksperimentišući, pronašli takve vrijednosti parametara regulatora  $K_R = K_R'$ ,  $T_i = T_i'$  i  $T_d = T_d'$  za koje se pri promjeni zadane vrijednosti izlaz iz procesa ne smiruje, već počinje oscilirati neprigušenim oscilacijama kružne frekvencije  $\omega = \omega'$ . Na osnovu poznatih osobina Laplaceove transformacije slijedi da je to moguće jedino ukoliko prenosna funkcija  $W(s)$  ima pol u tački  $s = j\omega'$ , odnosno ukoliko je  $W(j\omega') = \infty$ , iz čega dalje slijedi:

$$1 + G_R(j\omega', K_R', T_i', T_d') G_{ob}(j\omega') = 0$$

odnosno:

$$G_R(j\omega', K_R', T_i', T_d') = -\frac{1}{G_{ob}(j\omega')}$$

Uvrštavajući izraze za  $G_R(s, K_R, T_i, T_d)$  i  $G_{ob}(s)$  dobijamo:

$$K_R' \left( 1 + \frac{1}{T_i' j\omega'} + T_d' j\omega' \right) = - \frac{T_{ob}^2 (j\omega')^2 + 2 T_{ob} \xi_{ob} j\omega' + 1}{K_{ob}} e^{T_{ob} j\omega'}$$

odnosno:

$$K_{ob} K_R' \left( 1 + \frac{1}{T_i' j\omega'} + T_d' j\omega' \right) = - \left( 2 T_{ob} \xi_{ob} j\omega' + 1 - T_{ob}^2 \omega'^2 \right) e^{T_{ob} j\omega'}$$

Uzmemo li kvadrat apsolutne vrijednosti obje strane ove jednačine, dobijamo:

$$K_{ob}^2 K_R'^2 \left[ 1 + \left( T_d' \omega' - \frac{1}{T_i' \omega'} \right)^2 \right] = 4 T_{ob}^2 \xi_{ob}^2 \omega'^2 + (1 - T_{ob}^2 \omega'^2)^2$$

Uz malo preuređivanja, ova jednačina se može napisati u obliku:

$$K_{ob}^2 K_R'^2 \left[ 1 + \left( T_d' \omega' - \frac{1}{T_i' \omega'} \right)^2 \right] - 2 T_{ob}^2 (2 \xi_{ob}^2 - 1) \omega'^2 - T_{ob}^4 \omega'^4 = 1$$

Uvedemo li smjenu  $K_{ob}^2 = \alpha$ ,  $T_{ob}^2 (2 \xi_{ob}^2 - 1) = \beta$  i  $T_{ob}^4 = \gamma$ , ova jednačina postaje:

$$K_R'^2 \left[ 1 + \left( T_d' \omega' - \frac{1}{T_i' \omega'} \right)^2 \right] \alpha - 2 \omega'^2 \beta - \omega'^4 \gamma = 1$$

odakle vidimo da je ova jednačina linearna po  $\alpha$ ,  $\beta$  i  $\gamma$ . Ukoliko sada ponovimo eksperiment još dva puta, i pored četvorke  $(K_R', T_i', T_d', \omega')$  pronađemo još dvije četvorke vrijednosti  $(K_R'', T_i'', T_d'', \omega'')$  i  $(K_R''', T_i''', T_d''', \omega''')$  za koje se postižu ustaljene oscilacije, možemo i za ove vrijednosti formirati analogne jednačine, što nas na kraju dovodi do sistema tri linearne jednačine sa tri nepoznate  $\alpha$ ,  $\beta$  i  $\gamma$ :

$$K_R'^2 \left[ 1 + \left( T_d' \omega' - \frac{1}{T_i' \omega'} \right)^2 \right] \alpha - 2 \omega'^2 \beta - \omega'^4 \gamma = 1$$

$$K_R''^2 \left[ 1 + \left( T_d'' \omega'' - \frac{1}{T_i'' \omega''} \right)^2 \right] \alpha - 2 \omega''^2 \beta - \omega''^4 \gamma = 1$$

$$K_R'''^2 \left[ 1 + \left( T_d''' \omega''' - \frac{1}{T_i''' \omega'''} \right)^2 \right] \alpha - 2 \omega'''^2 \beta - \omega'''^4 \gamma = 1$$

Kada iz ovog sistema odredimo  $\alpha$ ,  $\beta$  i  $\gamma$ , lako nalazimo nepoznate  $K_{ob}$ ,  $T_{ob}$  i  $\xi_{ob}$ . Doduše, njihov znak se ne može odrediti, jer je izgubljen kvadriranjem. Međutim, ukoliko se zna da je sam objekat stabilan (što je naščešće slučaj), sve ove veličine moraju biti pozitivne, pa je i taj problem otklonjen. Također se može desiti da se za neku od veličina  $K_{ob}$ ,  $T_{ob}$  i  $\xi_{ob}$  dobije vrijednost koja nije realna (npr. ako se za  $\alpha$  dobije negativna vrijednost,  $T_{ob}$  sigurno neće biti realan). Ovakva situacija ukazuje na nekorektnost modela, odnosno da se identificirani objekat ne može zadovoljavajuće dobro opisati pretpostavljenim oblikom prenosne funkcije.

Ostaje još da se odredi nepoznata vrijednost  $\tau_{ob}$ . Nju možemo odrediti iz relacije:

$$K_{ob} K_R' \left( 1 + \frac{1}{T_i' j\omega'} + T_d' j\omega' \right) = - \left( 2 T_{ob} \xi_{ob} j\omega' + 1 - T_{ob}^2 \omega'^2 \right) e^{T_{ob} j\omega'}$$

koja nakon rješavanja po  $\tau_{ob}$  daje:

$$\tau_{ob} = \frac{1}{\omega'} \arg \left( \pi + \frac{1 + \frac{1}{T_i' j\omega'} + T_d' j\omega'}{2 T_{ob} \xi_{ob} j\omega' + 1 - T_{ob}^2 \omega'^2} \right)$$

U ovoj relaciji se sve može izračunati, jer smo prethodno odredili  $K_{ob}$ ,  $T_{ob}$  i  $\xi_{ob}$ . Međutim, slična relacija za  $T_{ob}$  se može formirati i na osnovu četvorki  $(K_R'', T_i'', T_d'', \omega')$  i  $(K_R''', T_i''', T_d''', \omega'')$  umjesto četvorke  $(K_R', T_i', T_d', \omega)$ . Tako, možemo dobiti tri vrijednosti za  $T_{ob}$ . Ukoliko je model korektan, ove tri vrijednosti moraju biti identične. Naravno, zbog neizbježnih grešaka u mjerenju, dopušteno je imati izvjesna odstupanja u izračunatim vrijednostima za  $T_{ob}$ , ali u svakom slučaju bitna odstupanja za vrijednosti  $T_{ob}$  dobijene iz rezultata tri eksperimenta ukazuju da nešto nije u redu sa pretpostavljenim modelom.

Da rezimiramo, identifikacija pretpostavljenog objekta može se odrediti na osnovu tri eksperimenta u kojima se nalaze takve vrijednosti parametara regulatora za koje se na izlazu dobija neprigušen oscilatorni odziv. Ulazni parametri za identifikaciju su kritične vrijednosti parametara regulatora iz sva tri eksperimenta, kao i kružna frekvencija  $\omega$  uspostavljenih oscilacija za sva tri slučaja. Frekvencija  $\omega$  se može odrediti bilo odoka sa snimljenog grafikona odziva, mjereći period oscilacija  $T$  i računajući  $\omega = 2\pi/T$ . Alternativno, moguće je napraviti MATLAB program koji će postprocesirati izlaz iz sistema i automatski odrediti traženu kružnu frekvenciju  $\omega$ .

## IV Realizacija projektnog zadatka

U realizaciji projektnog zadatka „Identifikacija objekta drugog reda sa transportnim kašnjenjem bazirana na prinudnim oscilacijama“ korišteni su IWS HMI paket za interfejs prema korisniku i MATLAB 7/SIMULINK 6, paket za simulaciju i modeliranje.

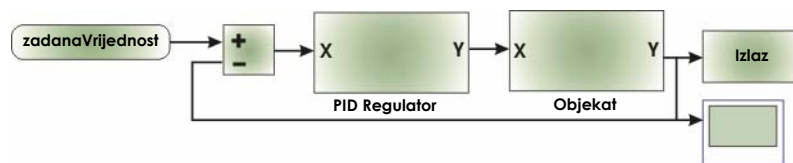
### Simulink

Simulink model “model.mdl” predstavlja model koji uključuje regulacionu konturu sa regulatorom sa prenosnom funkcijom:

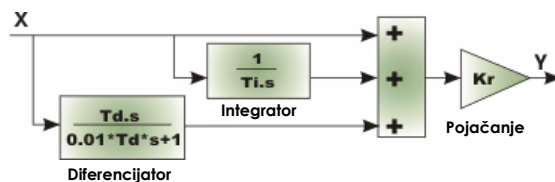
$$G_R(s) = K_R \left( 1 + \frac{1}{T_i s} + T_d s \right)$$

i objektom sa prenosnom funkcijom:

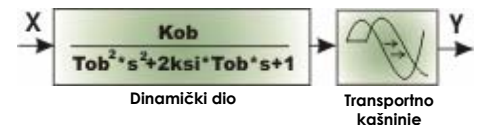
$$G_{ob}(s) = \frac{K_{ob}}{T_{ob}^2 s^2 + 2 T_{ob} \xi_{ob} s + 1} e^{-T_{ob} s}$$



Slika 4.1: Model



Slika 4.2: PID Regulator



Slika 4.3: Objekat

Da bi model ispravno radio u MATLAB okruženju trebaju biti definirane promjenljive “Kr”, “Ti”, “Td” koje opisuju parametre regulatora  $K_R$ ,  $T_i$  i  $T_d$  respektivno, zatim “Kob”, “Tob”, “ksi” i “tau” koje opisuju parametre objekta  $K_{ob}$ ,  $T_{ob}$ ,  $\xi_{ob}$  i  $T_{ob}$  respektivno, kao i promjenljiva “zadanaVrijednost” koja opisuje zadanu vrijednost na kojoj treba održavati reguliranu veličinu.

Simulacija se može pokrenuti iz menija SIMULINK-a, ili direktno iz MATLAB okruženja komandom: `sim('model', trajanje_simulacije);`

Rezultati simulacije se akviziraju u vektorsku promjenljivu “Izlaz”. Vektorska promjenljiva “Vrijeme” sadrži trenutke u kojima su akvizirani podaci, tako da se rezultati simulacije mogu grafički prikazati prostim navođenjem komande: `plot(Vrijeme, Izlaz);`

Kao primjer, sljedeća sekvenca producira oscilatorni odziv (postignute su kritične vrijednosti parametara, kakve se upravo koriste za identifikaciju):

```
zadanaVrijednost = 1;
Kob = 0.1667;      % Parametri objekta
Tob = 0.4082;      % Parametri objekta
ksi = 1.0206;      % Parametri objekta
```

```

tau = 2;           % Parametri objekta
Kr = 2;           % Parametri regulatora
Ti = 0.48;        % Parametri regulatora
Td = 0;           % Parametri regulatora
sim ('model', 100);
plot (Vrijeme, Izlaz);

```

Slični oscilatorni odzivi dobijaju se recimo i za parametre regulatora (uz isti objekat)  $K_r = 4$ ,  $T_i = 0.945$ ,  $T_d = 0$  ili  $K_r = 6$ ,  $T_i = 1.82$ ,  $T_d = 0$ .

S druge strane, parametri regulatora  $K_r = 4.41$ ,  $T_i = 2.774$ ,  $T_d = 0.694$ , odgovaraju dobrom podešenju regulatora, za koje se dobija odziv koji se brzo smiruje.

## Matlab

Cjelokupna interakcija sa razvojnim okruženjem Matlaba je izvršena preko niza predefinisanih skripti i funkcija, tako da krajnji korisnik nema nikakvu potrebu za poznavanje rada sa razvojnim okruženjem Matlaba.

Skripte i funkcije korištene u ovom projektnom zadatku:

| Naziv m fajla      | Opis  |
|--------------------|---|
| azuriraj_ulaze     | Ažuriranje ulaznih vrijednosti u Matlab modelu  |
| inicijalizacija    | Inicijalizacija servera, reset Matlab-ovog OPC modula i ažuriranje ulaznih vrijednosti u Matlab modelu                              |
| kontrolaMatlab     | Kontrola event-a sa OPC servera radi kontrole aktivnosti Matlab-a   |
| kontrolaPodesenje  | Kontrola event-a sa OPC servera radi određivanja parametara za dobro podešenje regulatora, Ziegler-Nichols-ovom metodom, u Matlab-u |
| kontrolaRjesavanja | Kontrola event-a sa OPC servera radi pokretanja proračuna rješavanja zadatka u Matlab-u   |
| kontrolaSimulacija | Kontrola event-a sa OPC servera radi pokretanja simulacionog modela i aproksimacija simulacije u Matlab-u...                        |
| period             | Proračun perioda oscilacije trenutne simulacije   |
| podesenje          | Proračun parametara za dobro podešenje regulatora metodom Ziegler-Nichols-a   |
| proracun           | Proračunavanje rezultatskih parametara iz tri eksperimenta  |



# PRAVILA PRISTUPA APLIKACIJI

Da bi se zaštitio integritet aplikacije „**Identifikacija objekta drugog reda sa transportnim kašnjenjem bazirana na prinudnim oscilacijama**“ i njenih podataka, te pristup istima ograničio, kreirane su sigurnosne grupe korisnika aplikacije, koje uvode restrikcije na prava izvršavanja određenih operacija nad aplikacijom u razvojnom i izvršnom modu. Kreirane su sljedeće sigurnosne grupe korisnika:

1. **Guest**, koji ima pravo pristupa isključivo izvršnom dijelu aplikacije i to operacijama pokretanja i zatvaranja aplikacije (i njenih ekrana) sa sigurnosnim nivoom **0 – 2**,
2. **Korisnik**, koji ima pravo pristupa isključivo izvršnom dijelu aplikacije i svim operacijama omogućenim za taj mod rada (pokretanje i zatvaranje aplikacije i ekrana, upisivanje vrijednosti tagova u bazu tagova itd.) sa sigurnosnim nivoom **0 – 4**,
3. **Programer**, koji ima pravo pristupa izvršnom dijelu aplikacije i svim njenim operacijama sa sigurnosnim nivoom **0 – 6**, te pravo pristupa razvojnom dijelu aplikacije i operacija kreiranja i mijenjanja tagova, kreiranja i mijenjanja ekrana i kreiranja i mijenjanja radnih listova sa sigurnosnim nivoom **0 – 6**,
4. **Admin**, koji ima pravo pristupa izvršnom dijelu aplikacije i svim njenim operacijama sa sigurnosnim nivoom **0 – 7**, te pravo pristupa razvojnom dijelu aplikacije i svim njenim operacijama (to su operacije dozvoljene grupi korisnika **Programer**, te još operacije podešavanja projektnih postavki, podešavanja drivera i izvora podataka i operacije mrežnog konfigurisanja) sa sigurnosnim nivoom **0 – 7**.

U zavisnosti od toga kako se korisnik logira pri pokretanju aplikacije (tj. kojoj sigurnosnoj grupi pripada) i u zavisnosti od sigurnosnih nivoa podešenih za pojedine ekrane i elemente ekrana, određuje se kojim ekranima (i/ili njihovim dijelovima) korisnik ima pristup.

## GLAVNI EKRAN

Pokretanjem aplikacije „*Identifikacija objekta drugog reda sa transportnim kašnjenjem bazirana na prinudnim oscilacijama*“ otvara se ekran prikazan na slici 1.

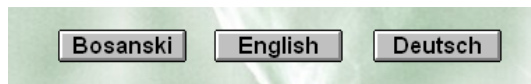


## Slika 1: Glavni ekran.scr

Pristup ovom ekranu imaju svi korisnici aplikacije (bez obzira na sigurnosnu grupu kojoj pripadaju).

Na ovom ekranu se nalazi šesnaest tastera, čijim se izborom otvaraju novi ekrani i/ili izvršavaju određene predefinisane akcije.

Sama aplikacija „*Identifikacija objekta drugog reda sa transportnim kašnjenjem bazirana na prinudnim oscilacijama*“ je trojezična aplikacija, što ostavlja korisniku mogućnost da za jezik aplikacije izabere između bosanskog, engleskog i njemačkog jezika, što se čini odabirom odgovarajućeg tastera, a što je prikazano na slici 2.



Slika 2: Odabir jezika aplikacije

Odabirom tastera *Pokreni Matlab...* pokreće se predefinisana akcija pokretanja skripte *pokreniMatlab.cmd*, koja pronalazi putanju do programa Matlab, te istu pokreće i odgovarajuće vrijednosti kontrolnih, programskih i predefinisanih parametara postavlja na početne vrijednosti.

Sam taster *Pokreni Matlab...* se nakon pokretanja programa Matlab mijenja u taster *Matlab pokrenut!*.

U slučaju da program Matlab nije pokrenut ispod tastera *Pokreni Matlab...* nalazi se taster *Matlab ugašen!*, koji nema funkciju dok je program Matlab ugašen.

U slučaju da je program Matlab pokrenut taster *Matlab ugašen!* se mijenja u taster *Ugasi Matlab!*, čijim se izborom pokreće predefinisana akcija za zatvaranje programa Matlab.

Sigurnosni nivo tastera *Pokreni Matlab...* i *Matlab ugašen!* je 3, što znači da im korisnik logiran kao *Guest* ne može pristupiti.

Odabirom tastera *Logiranje* pokreće se novi ekran na kojem je moguće odabrati i registrovati trenutnog korisnika aplikacije, a što će detaljnije biti opisano u nastavku.

Odabirom tastera *Pomoć* se pokreće dokument u .pdf formatu u kojem je opisano korištenje aplikacije „*Identifikacija objekta drugog reda sa transportnim kašnjenjem bazirana na prinudnim oscilacijama*“.

Odabirom jednog od tastera: *Struktura stabla ekrana*, *Sistemska datum i vrijeme*, *Model objekta*, *Parametri objekta*, *Simulacija*, *Podešenje regulatora*, *Historijski trend*, *ODBC* pokreću se odgovarajući ekrani, što će u nastavku biti detaljnije opisano. Sigurnosni nivo tastera *Struktura stabla ekrana* i *ODBC* je 3, što znači da im korisnik logiran kao *Guest* ne može pristupiti, a sigurnosni nivo tastera *Sistemska datum i vrijeme* je 7, što znači da mu može pristupiti samo korisnik logiran kao *Admin*.

U prostoru prikazanom na slici 3. nalazi se korisničko ime trenutno logiranog korisnika aplikacije.



Slika 3 : Trenutni korisnik

Odabirom tastera *Izlaz* aplikacija „*Identifikacija objekta drugog reda sa transportnim kašnjenjem bazirana na prinudnim oscilacijama*“ se zatvara i završava rad sa istom, te se pritom izvršava operacija kojom se poništava operacija logiranja trenutnog korisnika.

## LOGIRANJE

Nakon odabira tastera *Logiranje* na glavnom ekranu aplikacije otvara se ekran *Logiranje* (prikazan na slici 4.) na kojem je omogućeno logiranje, odnosno prijava korisnika.



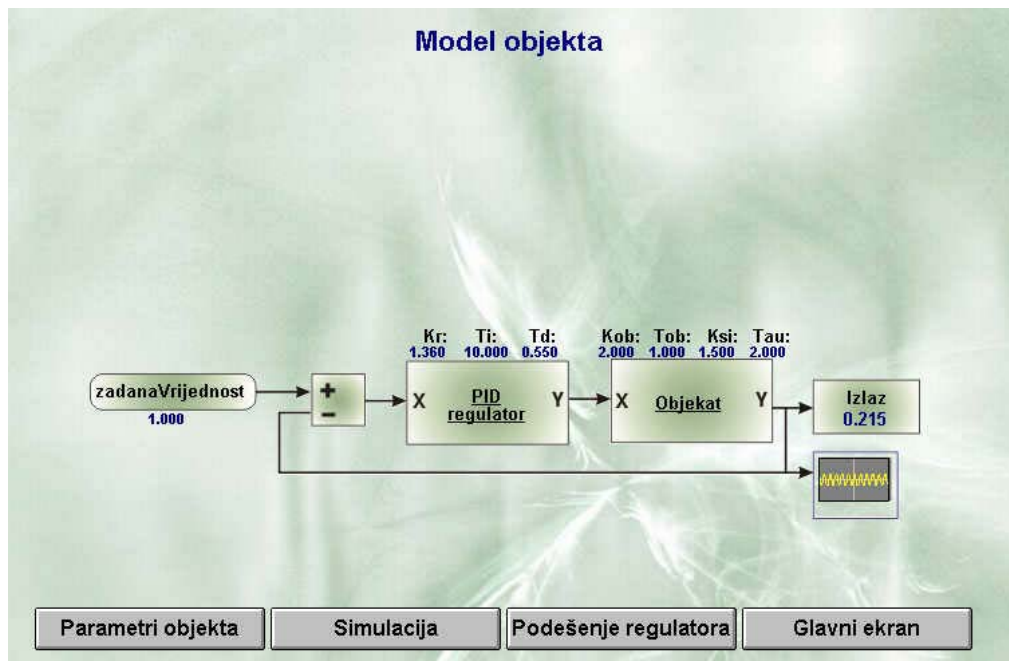
Slika 4 : Logiranje.scr

Pristup ovom ekranu imaju svi korisnici aplikacije (bez obzira na sigurnosnu grupu kojoj pripadaju).

Da bi se aplikacija mogla koristiti, potrebno se je prvo logirati, odnosno unijeti *Korisničko ime* i *Šifru* u za to predviđena polja na ekranu. Neregistrovani korisnici aplikaciji mogu pristupiti i kao *Guest* (*Gost*), ali sa vrlo ograničenim pravima pristupa aplikaciji. Nakon unosa *Korisničkog imena* i *Šifre*, odabirom tastera *OK* potvrđuju se uneseni podaci, zatvara se ekran *Logiranje* i vraća se na ekran *Glavni*. Odabirom tastera *Odustani* se odustaje od logiranja (poništavaju se uneseni podaci), zatvara se ekran *Logiranje* i vraća se na ekran *Glavni*.

## MODEL OBJEKTA

Nakon odabira tastera *Model objekta* na glavnom ekranu aplikacije otvara se ekran *Model* (prikazan na slici 5.) na kojem je prikazan model objekta čiju identifikaciju vršimo. Na ovom ekranu omogućeno je praćenje parametara regulatora, parametara objekta, kao i zadane vrijedosti te izlaza iz regulacione konture.

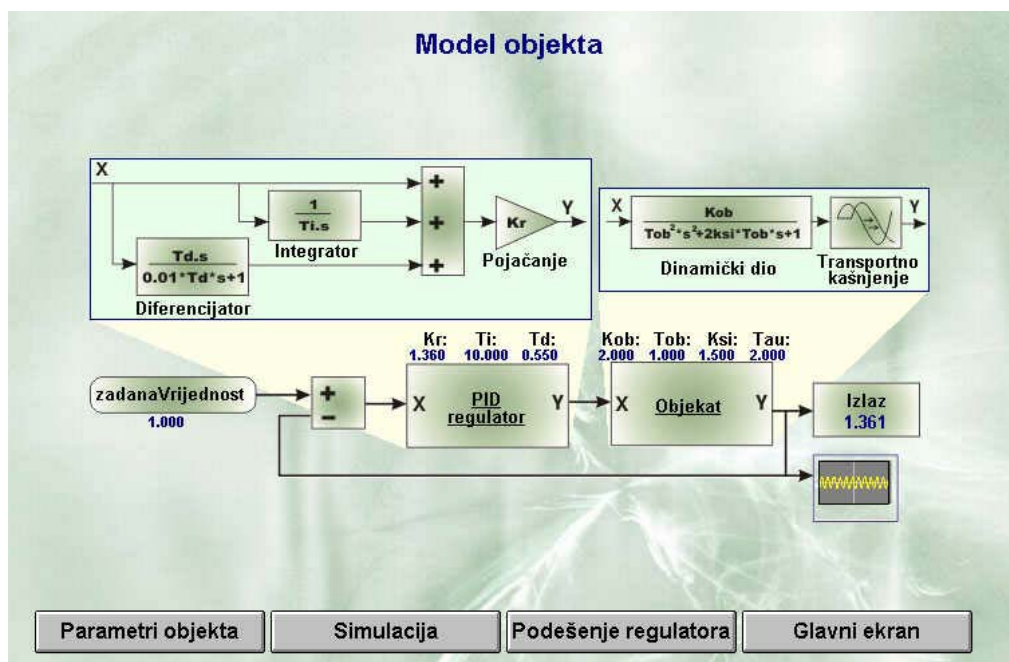


Slika 5: Model.scr

Pristup ovom ekranu imaju svi korisnici aplikacije (bez obzira na sigurnosnu grupu kojoj pripadaju).

Vrijednosti  $K_r$ ,  $T_i$  i  $T_d$  predstavljaju parametre regulatora: njegovo pojačanje, integralno vrijeme i derivativno vrijeme respektivno, dok vrijednosti  $zadanaVrijednost$ ,  $K_{ob}$ ,  $T_{ob}$ ,  $K_{si}$  i  $\tau_{au}$  označavaju parametre objekta, a predstavljaju respektivno zadanu vrijednost, pojačanje, vremensku konstantu, faktor prigušenja i iznos čistog kašnjenja.

Detaljnija struktura PID regulatora, kao i objekta se može dobiti odabirom bloka koji predstavlja regulator odnosno objekat, kao što je prikazano na slici 6.



Slika 6: Detaljniji prikaz blokova PID regulatora i objekta

Mali osciloskop na dnu ekrana (slika 7.) predstavlja trend objekat koji omogućava prikazivanje izlazne veličine, a klikom na



Slika 7: Prikaz izlaza

njega otvara se ekran *Simulacija* na kojem se može vršiti pokretanje simulacije i njeno praćenje.

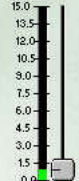

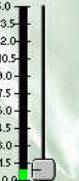
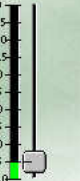

Odabirom nekog od tastera *Parametri objekta*, *Podešenje regulatora*, *Simulacija* ili *Glavni ekran* zatvara se ekran *Model* i otvara se neki od ekrana *ParametriObjekta*, *PodešenjeRegulatora*, *Simulacija* ili *Glavni* respektivno.

## PARAMETRI OBJEKTA

Nakon odabira tastera *Parametri objekta* na glavnom ekranu aplikacije otvara se ekran *Model* (prikazan na slici 8.) na kojem je omogućeno postavljanje vrijednosti *Zadana vrijednost*, *Pojaćanje (Kob)*, *Vremenska konstanta (Tob)*, *Faktor prigušenja (Ksi)*, te *Iznos čistog kašnjenja (Tau)* pomoću klizača ili direktnim unosom vrijednosti u polja ispod klizača.

Pristup ovom ekranu imaju svi korisnici aplikacije (bez obzira na sigurnosnu grupu kojoj pripadaju).

**Postavljanje parametara objekta i upoređivanje sa rješenjem**

| Zadana vrijednost:   | Pojaćanje (Kob):   | Vremenska konstanta (Tob):   | Faktor prigušenja (Ksi):  | Iznos čistog kašnjenja (Tau):  |
|--|--|--|---|--|
|  |  |  |  |  |
| 1.0000   | 2.0000   | 1.0000   | 1.5000  | 2.0000   |

Parametri objekta identificirani nakon izvršenih pokusa:

| Pojaćanje (Kob): | Vremenska konstanta (Tob): | Faktor prigušenja (Ksi): | Iznos čistog kašnjenja (Tau): |
|------------------|----------------------------|--------------------------|-------------------------------|
| 1.972            | 0.986                      | 1.495                    | 2.032                         |

Model objektaSimulacijaPodešenje regulatoraGlavni ekran

Slika 8: ParametriObjekta.scr

Parametri objekta identificirani nakon izvršenih pokusa (*Kob*, *Tob*, *Ksi* i *Tau*) predstavljaju vrijednosti parametara objekta koji se dobiju kao rezultat proračuna navedenih parametara na osnovu provedena tri eksperimenta u kojima se traže takve vrijednosti parametara regulatora (*Kr*, *Ti* i *Td*) za koje se na izlazu iz regulacione konture dobija oscilatorni odziv. Ovaj postupak će kasnije biti detaljnije objašnjen.

Prikaz ovih parametara nam omogućava da uporedimo postavljene vrijednosti sa vrijednostima koje smo dobili kao rezultat proračuna, da bi utvrdili ispravnost metode identifikacije objekta i njene rezultate.

Sigurnosni nivo klizača i polja za unos vrijednosti parametara objekta je 3, što znači da im korisnik logiran kao *Guest* ne može pristupiti.

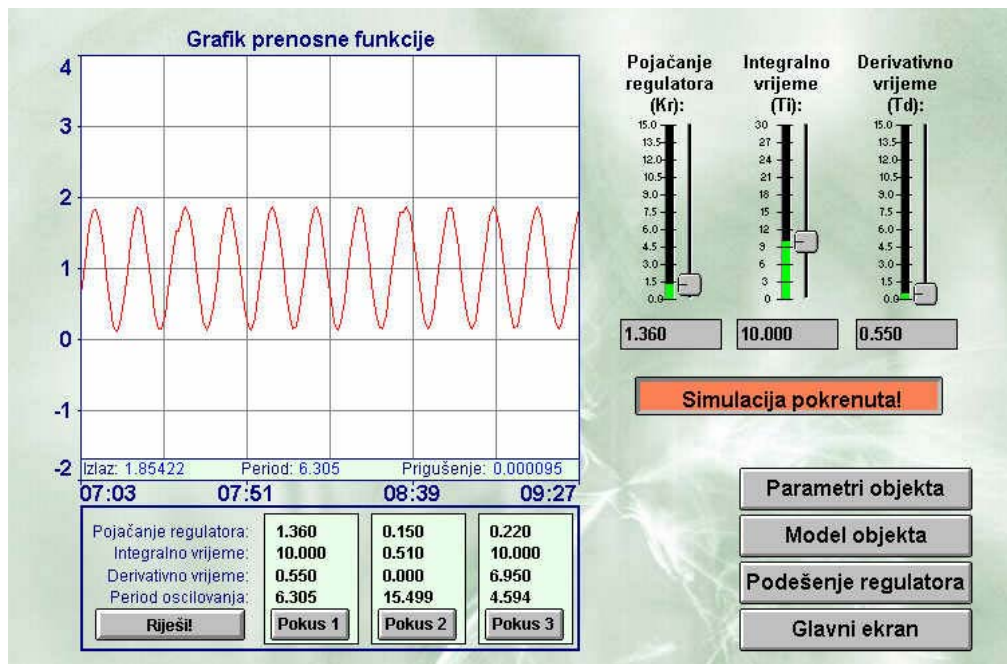
Odabirom nekog od tastera *Model objekta*, *Simulacija*, *Podešenje regulatora* ili *Glavni ekran* zatvara se ekran *Parametri objekta* i otvara se neki od ekrana *Model*, *Simulacija*, *PodešenjeRegulatora* ili *Glavni* respektivno.



# SIMULACIJA

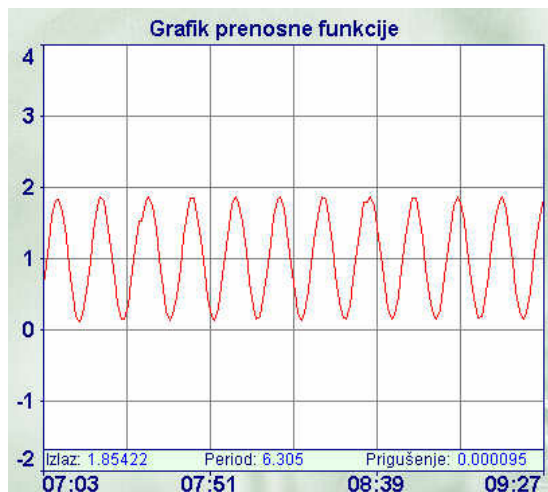
Nakon odabira tastera *Simulacija* na glavnom ekranu aplikacije otvara se ekran *Simulacija* (prikazan na slici 9.) na kojem je autorizovanom korisniku omogućeno da izvrši eksperimente (simulacije) potrebne za proces identifikacije parametara objekta.

Pristup ovom ekranu imaju svi korisnici aplikacije (bez obzira na sigurnosnu grupu kojoj pripadaju).

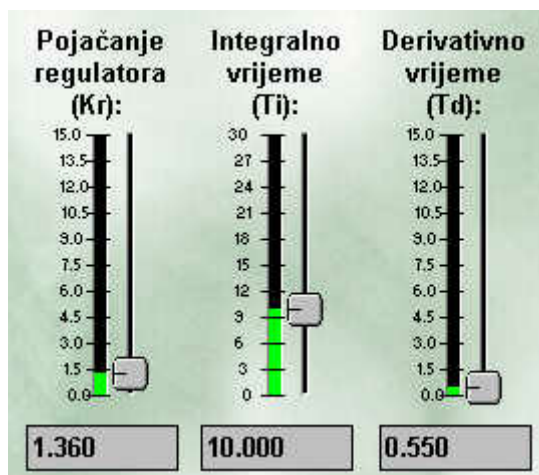


Slika 9: Simulacija.scr

Grafik prenosne funkcije, prikazan na slici 10., predstavlja trend objekat koji prikazuje izlaznu veličinu iz modela objekta. Osim iscrtavanja izlazne veličine, na trendu grafiku se prikazuju i sljedeće vrijednosti: *Izlaz*, *Period* i *Prigušenje*. *Izlaz* predstavlja numerički oblik izlazne veličine (ispisuje vrijednosti izlazne veličine u vremenu), *Period* predstavlja period oscilovanja izlazne veličine (za provođenje eksperimenata za identifikaciju objekta i za podešenje regulatora potrebno je podesiti parametre regulatora tako da izlazna veličina ima oscilatorni odziv), a *Prigušenje* predstavlja faktor prigušenja oscilacija izlazne veličine. Pozitivne vrijednosti ovog parametra odgovaraju prigušenom oscilatornom odzivu (amplituda oscilacija se konstantno smanjuje), a negativne vrijednosti odgovaraju divergentnom oscilatornom odzivu (amplituda se konstantno povećava). Sam iznos parametra određuje stepen prigušenja/divergencije. Ukoliko je vrijednost ovog parametra 0, oscilatorni odziv je neprigušen (to znači da je amplituda oscilacija konstantnog iznosa).



Slika 10: Grafik prenosne funkcije



Slika 11: Parametri regulatora

Polja za unos prikazana na slici 11. omogućavaju korisniku da direktno ili pomoću klizača postavi parametre regulatora: pojačanje regulatora ( $K_r$ ), integralno vrijeme ( $T_i$ ) i derivatno vrijeme ( $T_d$ ). Navedena tri parametra su predstavljena tagovima  $iUlazKr$ ,  $iUlazTi$  i  $iUlazTd$ . Vrijednosti navedena tri taga (koji su serverskog tipa) se proslijeđuju u program Matlab, gdje se vrši proračunavanje parametara za dobro podešenje regulatora. Proslijeđivanje vrijednosti ovih parametara se vrši odabirom tastera *Pokretanje simulacije* koji pokreće predefinisanu akciju pokretanja MatLab skript fajla *kontrolaSimulacija.m* koji pokreće simulaciju modela objekta u SimuLinku za postavljene parametre objekta i regulatora. Sam taster *Pokretanje simulacije* se mijenja u taster *Simulacija pokrenuta!*, sve dok se simulacija ne završi ili dok je korisnik sam ne prekine ponovnim pritiskom na isti taster. Kao rezultat simulacije na trendu objektu dobijemo prikaz izlazne veličine, te vrijednosti perioda oscilovanja i faktora prigušenja oscilacija. Sigurnosni nivo klizača i polja za unos vrijednosti parametara regulatora je 3, što znači da im korisnik logiran kao *Guest* ne može pristupiti.

Da bi izvršili identifikaciju parametara objekta, potrebno je podesiti parametre regulatora tako da se kao izlazna veličina dobije oscilatorni odziv, pri čemu nije važno da li se kao oscilatorni odziv dobija neprigušen, prigušen ili divergentan oscilatorni odziv. Ovo se može utvrditi vizualnom inspekcijom izlazne veličine na trend objektu. Nakon što na trend objektu dobijemo prikaz izlazne veličine (tj. nakon što kao izlaz dobijemo oscilatorni odziv), da bi se izvršio proračun parametara za identifikaciju objekta, potrebno je odabrati taster *Pokus 1* (prikazano na slici 12.), koji omogućava da se zapamte vrijednosti  $K_r$ ,  $T_i$ ,  $T_d$  i *Period* (odnosi se na one vrijednosti parametara regulatora, pri kojoj se kao izlaz dobija oscilatorni odziv, te na period oscilovanja) u tagove  $iUlazEksperiment[0].iUlazKr$ ,  $iUlazEksperiment[0].iUlazTi$ ,  $iUlazEksperiment[0].iUlazTd$  i  $iUlazEksperiment[0].ilzlazPeriod$ .

Kako je za proces identifikacije parametara objekta nužno provesti tri eksperimenta, potrebno je ponoviti gore opisanu proceduru još dva puta (smještanje vrijednosti regulatora se u drugom eksperimentu vrši odabirom tastera *Pokus 2* u iste tagove navedene gore – ali sa indeksom niza 1, a u trećem odabirom tastera *Pokus 3* u navedene tagove za indeksom niza 2), pri čemu je nužno da skupovi vrijednosti parametara regulatora budu različiti za sva tri slučaja, kao što je prikazano na slici 12. Nakon izvršenih eksperimenata, odabirom tastera *Riješi!* (prikazan na slici 12.) pokreće se predefinisana akcija pokretanja Matlab skript fajla *kontrolaRjesavanja.m* koji učitava vrijednosti parametara regulatora i perioda oscilovanja za tri navedena eksperimenta putem OPC servera, te pokreće Matlab funkciju *proracun.m*, koja vrši proračun parametara objekta na osnovu gore spomenutih parametara. Taster *Riješi!* se mijenja u taster *Rješavam...*, sve dok se ne identificiraju vrijednosti parametara. Sam postupak računanja (matematski model procesa identifikacije objekta) se može pogledati u dokumentaciji priloženoj uz aplikaciju, te u spomenutoj Matlab funkciji *proracun.m*. Sigurnosni nivo tastera *Pokus 1*, *Pokus 2*, *Pokus 3* i *Riješi!* je 3, što znači da im korisnik logiran kao *Guest* ne može pristupiti.

|                       |                |                |                |
|-----------------------|----------------|----------------|----------------|
| Pojačanje regulatora: | 1.360          | 0.150          | 0.220          |
| Integralno vrijeme:   | 10.000         | 0.510          | 10.000         |
| Derivativno vrijeme:  | 0.550          | 0.000          | 6.950          |
| Period oscilovanja:   | 6.305          | 15.499         | 4.594          |
| <b>Riješi!</b>        | <b>Pokus 1</b> | <b>Pokus 2</b> | <b>Pokus 3</b> |

Slika 12: Eksperimenti za identifikaciju objekta

Nakon što se izračunaju vrijednosti parametara objekta (objekat se identificira), iste se ponovo, putem OPC servera, šalju u IWS gdje se prikazuju na ekranu *ParametriObjekta*, kao na slici 13. (rezultati identifikacije su naznačeni crvenim pravougaonikom).

**Postavljanje parametara objekta i upoređivanje sa rješenjem**

| Zadana vrijednost:  | Pojačanje (Kob):   | Vremenska konstanta (Tob):  | Faktor prigušenja (Ksi):  | Iznos čistog kašnjenja (Tau):   |
|---|--|---|---|---|
| 15.0<br>13.5<br>12.0<br>10.5<br>9.0<br>7.5<br>6.0<br>4.5<br>3.0<br>1.5<br>0.0 | 30<br>27<br>24<br>21<br>18<br>15<br>12<br>9<br>6<br>3<br>0 | 15.0<br>13.5<br>12.0<br>10.5<br>9.0<br>7.5<br>6.0<br>4.5<br>3.0<br>1.5<br>0.0 | 15.0<br>13.5<br>12.0<br>10.5<br>9.0<br>7.5<br>6.0<br>4.5<br>3.0<br>1.5<br>0.0 | 15.0<br>13.5<br>12.0<br>10.5<br>9.0<br>7.5<br>6.0<br>4.5<br>3.0<br>1.5<br>0.0 |
| 1.0000  | 2.0000   | 1.0000  | 1.5000  | 2.0000  |

**Parametri objekta identificirani nakon izvršenih pokusa:**

| Pojačanje (Kob): | Vremenska konstanta (Tob): | Faktor prigušenja (Ksi): | Iznos čistog kašnjenja (Tau): |
|------------------|----------------------------|--------------------------|-------------------------------|
| 1.972            | 0.986                      | 1.495                    | 2.032                         |

Model objekta    Simulacija    Podešenje regulatora    Glavni ekran

Slika 13: Identificirani parametri objekta

Za objekat se smatra da je dobro identificiran, ako vrijednosti parametara objekta koje se dobiju procesom identifikacije objekta ne odstupaju puno od vrijednosti istih parametara



postavljenih na početku procesa identifikacije. To se može utvrditi vizualnom inspekcijom, tj. poređenjem identificiranih vrijednosti sa postavljenim, kako je prikazano na slici 13. Odstupanja od  $\pm 5\%$  do  $\pm 10\%$  se smatraju tolerantnim odstupanjima dobijenih vrijednosti u odnosu na postavljene vrijednosti parametara pri procesu identifikacije, što je kako vidimo u našem slučaju zadovoljeno (odstupanja od vrijednosti su kako slijedi: za  $K_{ob}$  je  $-1.4\%$ , za  $T_{ob}$  je  $-1.4\%$ , za  $K_{si}$  je  $-0.3333\%$ , te za  $\tau$  je  $+1.6\%$ ).

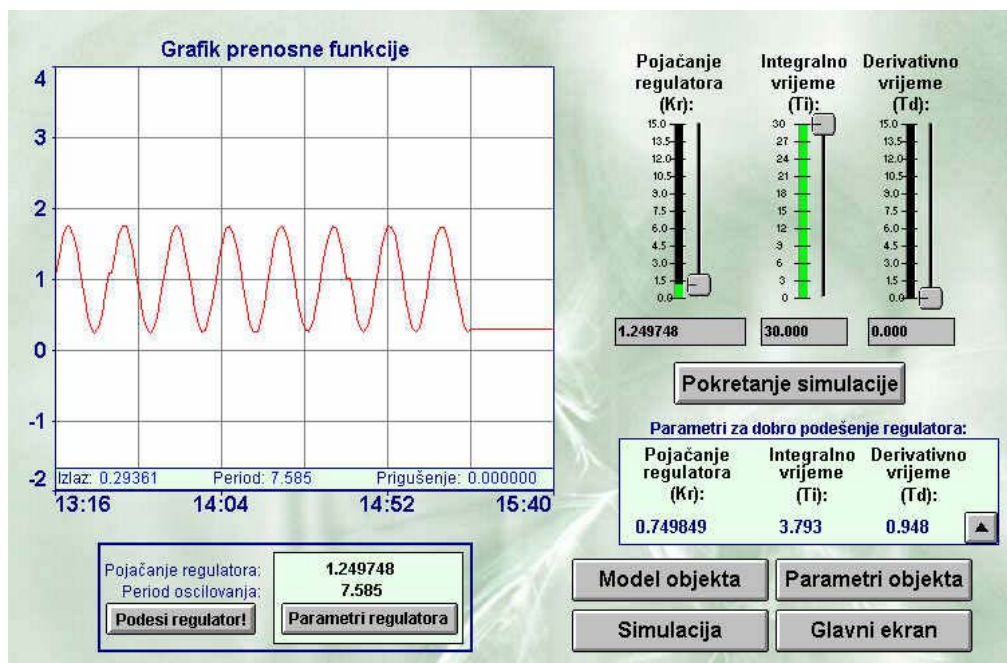
Odabirom nekog od tastera *Parametri objekta*, *Model objekta*, *Podešenje regulatora* ili *Glavni ekran*, zatvara se ekran *Simulacija* i otvara jedan od ekrana *ParametriObjekta*, *Model*, *PodesenjeRegulatora* ili *Glavni* respektivno (slika 14).



Slika 14: Tasteri za pristup ekranima aplikacije

## PODEŠENJE REGULATORA

Nakon odabira tastera *Podešenje regulatora* na glavnom ekranu aplikacije otvara se ekran *PodesenjeRegulatora* (prikazan na slici 15.) na kojem je autorizovanom korisniku omogućeno da izvrši podešenje regulatora.

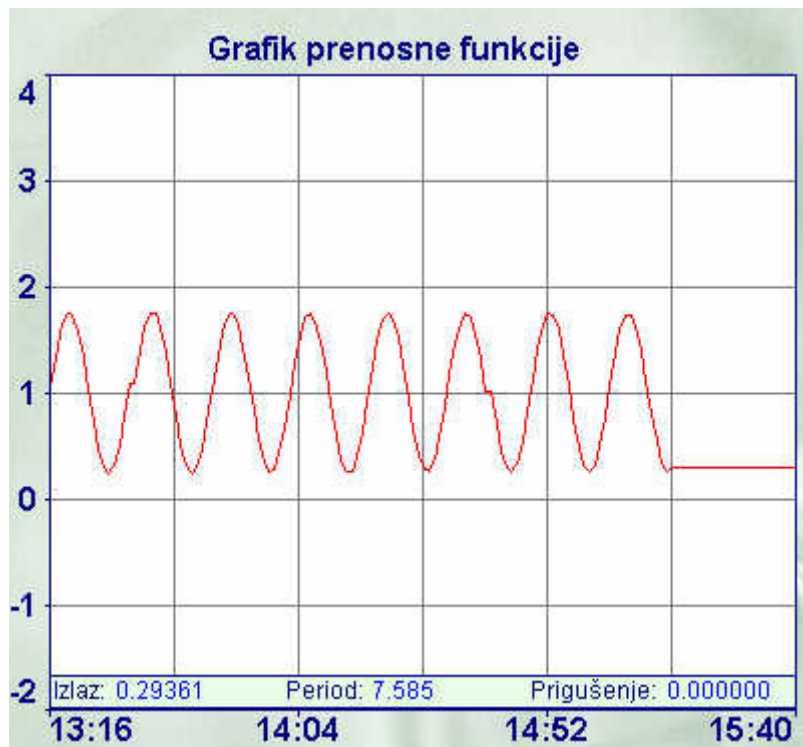


Slika 15: PodesenjeRegulatora.scr

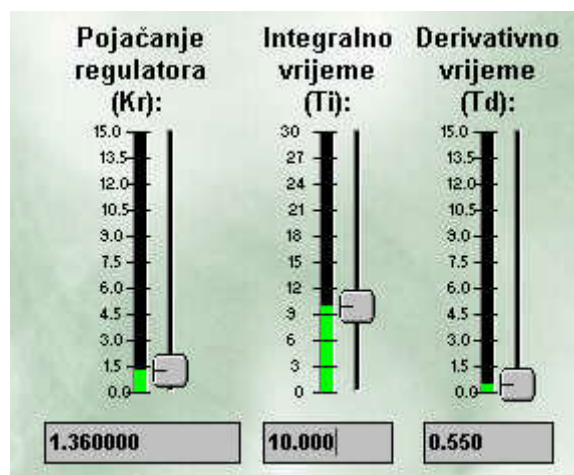
Pristup ovom ekranu imaju svi korisnici aplikacije (bez obzira na sigurnosnu grupu kojoj pripadaju).

Grafik prenosne funkcije, prikazan na slici 16., predstavlja trend objekat koji prikazuje izlaznu veličinu iz modela objekta. Osim iscrtavanja izlazne veličine, na trendu grafiku se prikazuju i sljedeće vrijednosti: *Izlaz*, *Period* i *Prigušenje*. *Izlaz* predstavlja numerički oblik

izlazne veličine (ispisuje vrijednosti izlazne veličine u vremenu), *Period* predstavlja period oscilovanja izlazne veličine (za provođenje eksperimenata za identifikaciju objekta i za podešenje regulatora potrebno je podesiti parametre regulatora tako da izlazna veličina ima oscilatorni odziv), a *Prigušenje* predstavlja faktor prigušenja oscilacija izlazne veličine. Pozitivne vrijednosti ovog parametra odgovaraju prigušenom oscilatornom odzivu (amplituda oscilacija se konstantno smanjuje), a negativne vrijednosti odgovaraju divergentnom oscilatornom odzivu (amplituda se konstantno povećava). Sam iznos parametra određuje stepen prigušenja/divergencije. Ukoliko je vrijednost ovog parametra 0, oscilatorni odziv je neprigušen (to znači da je amplituda oscilacija konstantnog iznosa).



Slika 16: Grafik prenosne funkcije



Slika 17: Parametri regulatora

Polja za unos prikazana na slici 17. omogućavaju korisniku da direktno ili pomoću klizača postavi parametre regulatora: pojačanje regulatora ( $K_r$ ), integralno vrijeme ( $T_i$ ) i derivatno vrijeme ( $T_d$ ). Navedena tri parametra su predstavljena tagovima *iUlazKr*, *iUlazTi* i *iUlazTd*. Vrijednosti navedena tri taga (koji su serverskog tipa) se proslijeđuju u program Matlab, gdje se vrši proračunavanje parametara za dobro podešenje regulatora. Proslijeđivanje vrijednosti ovih parametara se vrši odabirom tastera *Pokretanje simulacije* koji pokreće predefinisanu akciju pokretanja Matlab skript fajla *kontrolaSimulacija.m* koji pokreće

simulaciju modela objekta u SimuLinku za postavljene parametre objekta i regulatora. Sam taster *Pokretanje simulacije* se mijenja u taster *Simulacija pokrenuta!*, sve dok se simulacija ne završi ili dok je korisnik sam ne prekine ponovnim pritiskom na isti taster. Kao rezultat simulacije na trendu objektu dobijemo prikaz izlazne veličine, te vrijednosti perioda oscilovanja i faktora prigušenja oscilacija. Sigurnosni nivo klizača i polja za unos vrijednosti parametara regulatora je 3, što znači da im korisnik logiran kao *Guest* ne može pristupiti.

Da bi odredili parametre za dobro podešenje regulatora, potrebno je podesiti parametre regulatora na sljedeći način. Potrebno je podesiti vrijednost  $T_i$  (integralno vrijeme) na maksimalnu dozvoljenu vrijednost (u našem slučaju, to je vrijednost 30), te vrijednosti  $T_d$  (derivativno vrijeme) na minimalnu dozvoljenu vrijednost (to je vrijednost 0). Nakon toga, potrebno je mijenjati vrijednost parametra  $K_r$  (pojačanje regulatora) i pokretati simulaciju, sve dok se kao izlazna veličina ne dobije neprigušeni oscilatorni odziv. Kao precizniji način utvrđivanja da je odziv neprigušen (pored vizualnog), moguće je pratiti vrijednost faktora prigušenja (ukoliko je ona jednaka 0, odziv je neprigušen). Ovo je prikazano na slici 16.

Nakon što na trend objektu dobijemo prikaz izlazne veličine (tj. nakon što kao izlaz dobijemo neprigušeni oscilatorni odziv), da bi se izvršio proračun parametara za dobro podešenje regulatora, potrebno je odabrati taster *Parametri regulatora* (prikazano na slici 18.), koji omogućava da se zapamte vrijednosti  $K_r$  (odnosi se na onu vrijednost pojačanja regulatora, pri kojoj se kao izlaz dobija neprigušen oscilatorni odziv) i *Period*, koji predstavlja period oscilovanja. Nakon što se navedene vrijednosti smjeste u tagove *iParametriPodesenja.iUlazKr* i *iParametriPodesenja.ilzlazPeriod*, odabirom tastera *Podesi regulator!* pokreće se predefinisana akcija pokretanja Matlab skript fajla *kontrolaPodesenje.m* koji učitava prethodno navedene vrijednosti putem OPC servera, te pokreće Matlab funkciju *podesenje.m*, koja vrši proračun parametara regulatora, prema sljedećim formulama:

$$\begin{aligned} K_r &= 0.6 * K_{r1} \\ T_i &= 0.5 * T \\ T_d &= 0.125 * T \end{aligned}$$

gdje  $K_r$ ,  $T_i$  i  $T_d$  predstavljaju respektivno pojačanje regulatora, integralno vrijeme i derivativno vrijeme,  $K_{r1}$  – vrijednost pojačanja regulatora pri kojoj je dobiven neprigušeni oscilatorni odziv, a  $T$  – period oscilovanja za koje je dobiven takav odziv izlazne veličine. Sam taster *Podesi regulator!* se mijenja u taster *Podešavam...*, sve dok se ne proračunaju vrijednosti parametara. Sigurnosni nivo tastera *Parametri regulatora* i *Podesi regulator!* je 3, što znači da im korisnik logiran kao *Guest* ne može pristupiti.

Ove formule predstavljaju matematski model **Ziegler – Nichols**-ovog eksperimenta za određivanje parametara za dobro podešenje regulatora. Brojčane vrijednosti u formulama su navedeni autori dobili eksperimentalnim putem.



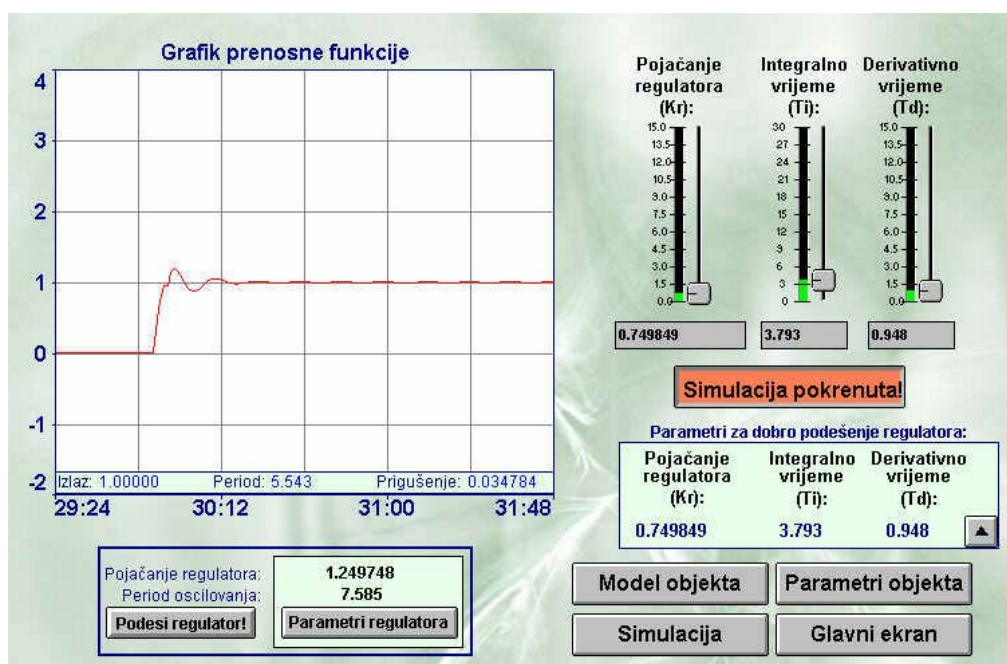
Slika 18: Podešenje regulatora

Nakon što se izračunaju vrijednosti parametara regulatora, iste se ponovo, putem OPC servera, šalju u IWS gdje se prikazuju na ekranu, kao na slici 19.



Slika 19: Parametri za dobro podešenje regulatora

Odabirom tastera sa oznakom 'gore' (naznačen crvenim krugom na slici 19.) vrijednosti parametara za dobro podešenje regulatora se kopiraju u polja za unos vrijednosti parametara regulatora (prikazana na slici 17.), nakon čega je moguće ponovo odabrati taster *Pokretanje simulacije* radi vizualne provjere da izračunati parametri regulatori zaista predstavljaju vrijednosti za dobro podešenje regulatora. Sigurnosni nivo navedenog tastera je 3, što znači da mu korisnik logiran kao *Guest* ne može pristupiti. Regulator je dobro podešen, ukoliko njegova izlazna veličina odgovara parametru *ZadanaVrijednost* objekta (koji je u našem primjeru jednak 1). To je prikazano na slici 20.



Slika 20: Vizualizacija dobrog podešenja regulatora

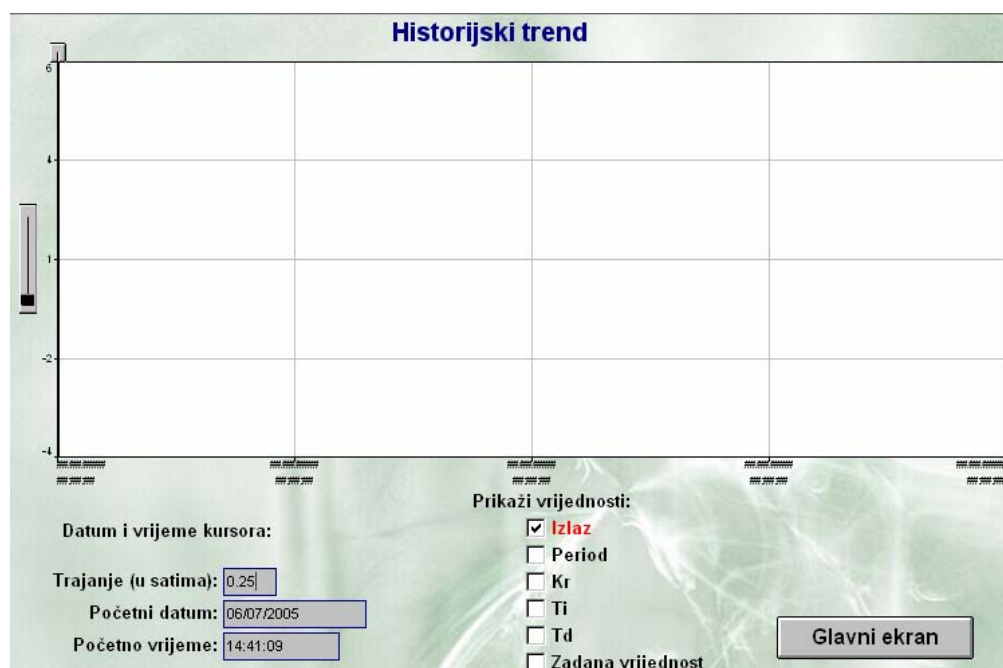
Odabirom nekog od tastera *Model objekta*, *Parametri objekta*, *Simulacija* ili *Glavni ekran*, zatvara se ekran *Podešenje regulatora* i otvara jedan od ekrana *Model*, *ParametriObjekta*, *Simulacija* ili *Glavni* respektivno (slika 21.).



Slika 21: Tasteri za pristup ekranima aplikacije

## HISTORIJSKI TREND

Nakon odabira tastera *Historijski trend* na glavnom ekranu aplikacije otvara se ekran *Historijski trend* na kojem korisnik može da vidi historijsko trendiranje zadane vrijednosti, parametara podešenja regulatora kao i izlazne veličine iz regulacione konture kroz pripadajuće im grafike (slika 22.).

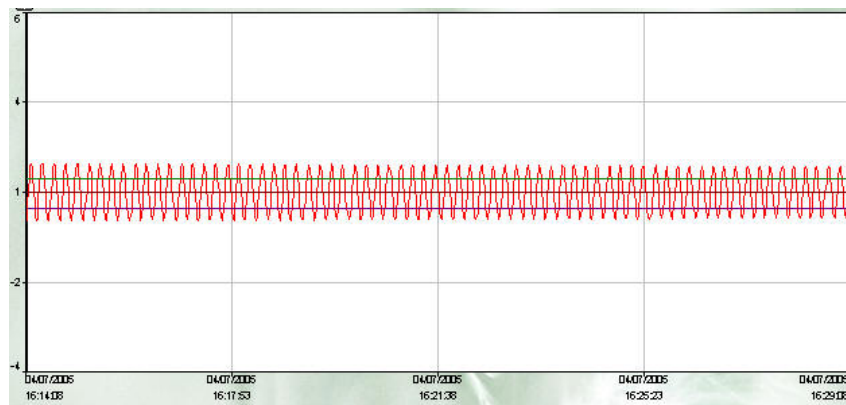


Slika 22: Historijski trend

Pristup ovom ekranu imaju svi korisnici aplikacije (bez obzira na sigurnosnu grupu kojoj pripadaju).

Zadana vrijednost, parametri podešenja regulatora kao i izlazne veličine iz regulacione konture se prikazuju u trend objektu prikazanom na slici 23. Vrijednosti koje se prikazuju su označene različitim bojama: *Izlaz* – crvena, *Period* – plava, *Pojačanje regulatora (Kr)* – zelena, *Integralno vrijeme (Ti)* – maslinasto zelena, *Derivativno vrijeme (Td)* – ljubičasta, *Zadana vrijednost* – braon.





Slika 23: Trend objekt

Odabir vrijednosti koje će biti prikazane se vrši označavanjem kvadratića pored vrijednosti koju želimo prikazati što je prikazano na slici 24.



Slika 24: Odabir vrijednosti koje će biti prikazane

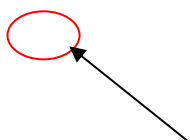
Vrijednosti trendiranih veličina se učitavaju iz .hst fajla koji se generiše na dnevnoj bazi (svaki dan se kreira novi .hst fajl). U ovom fajlu se trendiraju sve prikazane veličine, ali pored njih i *Pojačanje (Kob)*, *Vremenska konstanta (Tob)*, *Faktor prigušenja (Ksi)*, *Iznos čistog kašnjenja (Tau)*, i izlazni *Faktor prigušenja*, ali se zbog ograničenja razvojnog alata aplikacije iste ne prikazuju na trend objektu.

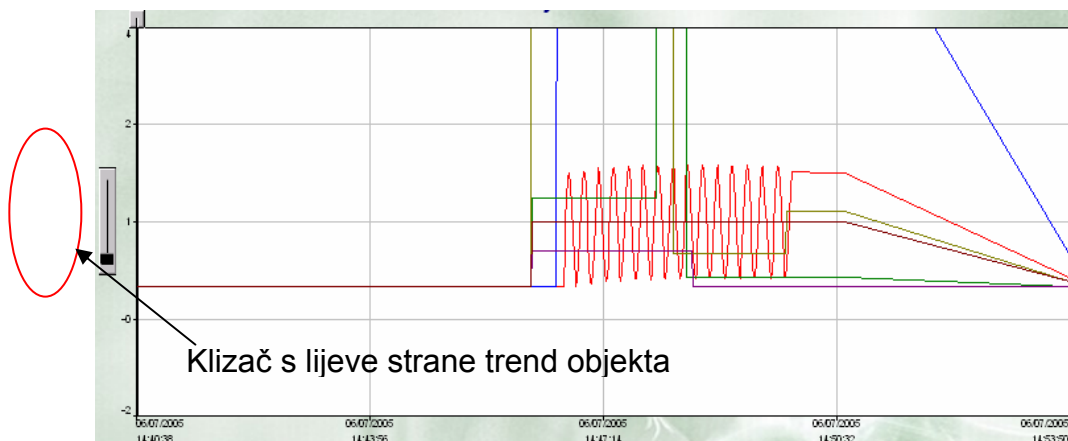
Izbor perioda za koji se želi prikazati historijski trend se vrši unosom vrijednosti u polja: *Trajanje (u satima)*, *Početni datum* i *Početno vrijeme*. Pri startovanju ekrana u navedenim poljima se nalaze tekuće sistemsko vrijeme i datum i trajanje u iznosu od 0.25 sati (slika 25.).

|                      |            |
|----------------------|------------|
| Trajanje (u satima): | 0.25       |
| Početni datum:       | 04/07/2005 |
| Početno vrijeme:     | 16:14:08   |

Slika 25: Izbor perioda za koji se želi prikazati historijski trend

Datum i vrijeme kursora predstavljaju datum i vrijeme trenutne pozicije kursora na trendu, koja se dobije presjekom klizača na vrhu trend objekta (označeno crvenim krugom) na slici 26. i vremenskom linijom trenda (x-osa na trend objektu), dok sam kursor predstavlja presjek gore navedenog klizača i bilo koje trendirane veličine. Svaki presjek predstavlja posebnu kursor liniju, za svaku od prikazanih trendiranih vrijednosti. Klizač na vrhu trend objekta se također koristi za pozicioniranje na dati vremenski trenutak i određivanje vrijednosti trendiranih veličina u tom vremenskom trenutku.





Slika 26: Klizač na vrhu trend ekrana

Klizač s lijeve strane trend objekta se koristi za skaliranje opsega vrijednosti izlazne veličine, to jest opsega y-ose. Opseg vrijednosti y-ose je od 0 do 30. Granice opsega y-ose su određene prema vrijednostima trendiranih veličina, uzeta je vrijednost najšireg opsega.

Odabirom tastera *Glavni ekran* zatvara se ekran *Historijski trend* i otvara ekran *Glavni*.

## ODBC

### NAPOMENA:

Opis koji slijedi je bio uradjen za raniju verziju ove aplikacije koja se izvršava korektno na PC-jevima sa Operativnim sistemom Win2K i WinXP SP1. Medjutim zbog poznatog problema u radu ODBC komunikacije izmedju IWS-a i Access baze podataka kod rada sa WindowsXP i SP2 ( za SQL callove SELECT, NEXT I DELETE), a aplikacija je verificirana na takvoj mašini, morala se modificirati da može raditi i u ovom okruženju.

Zbog toga, za korektan rad aplikacije, potrebno je formirati DSN sa slijedećim imenom( vidjeti iz headera ODBC konfiguracione tabele):

SPSSGrupa07Konekcija4

Koja povezuje Access bazu podataka pod imenom:

ParametriObjekta

na stazi :

C:\SPSS\Projects05\Grupa07

Slijedi nastavak opisa ranijeg rješenja, ali manje više je relevantan i za novo rješenje izuzev što su ODBC callovi : SELECT, NEXT , DELETE, itd , implementirani preko ODBC funkcija iz biblioteke IWS-a.

Nakon odabira tastera *ODBC* na glavnom ekranu aplikacije otvara se ekran *IdentifikacijaObjektaODBC* (prikazan na slici 27.) na kojem je, korištenjem ODBC protokola, omogućeno povezivanje aplikacije sa MS Access relacionom bazom podataka, te manipulacija nad podacima u bazi podataka.

**Konekcija sa bazom podataka putem ODBC-a**

Kolone u bazi:

| Postavljeni parametri objekta:                                    | Identificirani parametri objekta:                                 |
|---|---|
| Redni broj sloga u tabeli (ID): <input type="text" value="1"/>    |   |
| Zadana vrijednost: <input type="text" value="1.0000"/>            | Pojaćanje (Kob): <input type="text" value="0.0000"/>              |
| Pojaćanje (Kob): <input type="text" value="1.0000"/>              | Vremenska konstanta (Tob): <input type="text" value="0.0000"/>    |
| Vremenska konstanta (Tob): <input type="text" value="1.0000"/>    | Faktor prigušenja (Ksi): <input type="text" value="0.0000"/>      |
| Faktor prigušenja (Ksi): <input type="text" value="1.0000"/>      | Iznos čistog kašnjenja (Tau): <input type="text" value="0.0000"/> |
| Iznos čistog kašnjenja (Tau): <input type="text" value="1.0000"/> |   |

SQL komande:

**STATUS ODBC KONEKCIJE**

Kod greške: 0 Komanda: Nije unesena!

Opis greške: Transakcija izvršena bez greske!

Slika 27: IdentifikacijaObjektaODBC.scr

Pristup ovom ekranu imaju svi korisnici aplikacije osim grupe korisnika *Guest* (kojima je pristup zabranjen tako što je na glavnom ekranu aplikacije na tasteru *ODBC* postavljen sigurnosni nivo 3).

MS Access relaciona baza podataka *IdentifikacijaObjekta.mdb* ima jednu tabelu pod nazivom *ParametriObjekta* koja sadrži redom sljedeće kolone: *ID*, *ZadanaVrijednost*, *Kob*, *Tob*, *Ksi*, *Tau*, *IdentificiraniKob*, *IdentificiraniTob*, *IdentificiraniKsi* i *IdentificiraniTau*. *ID* predstavlja redni broj sloga u tabeli (primarni ključ tabele), *ZadanaVrijednost*, *Kob*, *Tob*, *Ksi* i *Tau* predstavljaju postavljene parametre objekta, a *IdentificiraniKob*, *IdentificiraniTob*, *IdentificiraniKsi* i *IdentificiraniTau* predstavljaju identificirane parametre objekta za objekat definisan postavljenim parametrima objekta. Navedene kolone direktno korespondiraju sljedećim tagovima: *dbIDParametri*, *ilnicijalizacijaKob*, *ilnicijalizacijaTob*, *ilnicijalizacijaKsi*, *ilnicijalizacijaTau*, *ildentificiraniParametriObjekta.ilnicijalizacijaKob*, *ildentificiraniParametriObjekta.ilnicijalizacijaTob*, *ildentificiraniParametriObjekta.ilnicijalizacijaKsi* i *ildentificiraniParametriObjekta.ilnicijalizacijaTau*.

Slika 28. prikazuje polja za unos u kojima je moguće direktno unijeti vrijednosti tagova *dbIDParametri*, *ilnicijalizacijaKob*, *ilnicijalizacijaTob*, *ilnicijalizacijaKsi*, *ilnicijalizacijaTau*, a vrijednosti preostala četiri taga se dobivaju nakon izvršenih eksperimenata (one predstavljaju krajnje rezultate procesa identifikacije objekta) za dati objekat, i ne mogu se direktno unositi. Tag *dbIDParametri* predstavlja tag koji prima cjelobrojne vrijednosti (pri čemu se samo prirodni brojevi prihvataju kao ispravne vrijednosti), dok svi ostali tagovi čije se vrijednosti bilježe u bazu podataka primaju realne vrijednosti. Opsezi vrijednosti za tagove sa realnim vrijednostima su sljedeći:

- *ilnicijalizacijaZadanaVrijednost*: 0–15
- *ilnicijalizacijaKob*, *ildentificiraniParametriObjekta.ilnicijalizacijaKob*: 0–30
- *ilnicijalizacijaTob*, *ildentificiraniParametriObjekta.ilnicijalizacijaTob*: 0– 5
- *ilnicijalizacijaKsi*, *ildentificiraniParametriObjekta.ilnicijalizacijaKsi*: 0–15
- *ilnicijalizacijaTau*, *ildentificiraniParametriObjekta.ilnicijalizacijaTau*: 0–15.



Kolone u bazi:

Redni broj sloga u tabeli (ID):

| Postavljeni parametri objekta:                                    | Identificirani parametri objekta:                                 |
|---|---|
| Zadana vrijednost: <input type="text" value="1.0000"/>            | Pojaćanje (Kob): <input type="text" value="0.0000"/>              |
| Pojaćanje (Kob): <input type="text" value="1.0000"/>              | Vremenska konstanta (Tob): <input type="text" value="0.0000"/>    |
| Vremenska konstanta (Tob): <input type="text" value="1.0000"/>    | Faktor prigušenja (Ksi): <input type="text" value="0.0000"/>      |
| Faktor prigušenja (Ksi): <input type="text" value="1.0000"/>      | Iznos čistog kašnjenja (Tau): <input type="text" value="0.0000"/> |
| Iznos čistog kašnjenja (Tau): <input type="text" value="1.0000"/> |   |

Slika 28: Unos vrijednosti kolona u bazi podataka

Odabirom tastera *Pokreni MS Access...* pokreće se predefinisana akcija pokretanja skripte *pokreniMSAccess.cmd*, koja pronalazi putanju do programa MS Access, te pokreće navedeni program i u okviru njega bazu podataka aplikacije.

Za uspostavljanje konekcije sa bazom podataka i vršenje manipulacije nad podacima u bazi, u okviru aplikacije su kreirana tri ODBC radna lista

Prvi ODBC radni list pod nazivom *Select, Insert i Delete po koloni ID*, čiji je izvor podataka *SPSSGrupa07Konekcija1*, omogućava izvršavanje uslovnih SQL komandi SELECT, INSERT i DELETE. Uslov  $ID = \{dbIDParametri\}$ , naveden u polju *Condition* u zaglavlju prvog ODBC radnog lista definiše da se SQL komande SELECT, INSERT i DELETE izvršavaju nad samo jednim redom tabele u bazi i to onim čiji je redni broj u koloni *ID* u bazi jednak vrijednosti taga *dbIDParametri* unesenoj u polju za unos na ekranu *IdentifikacijaObjektaODBC*. Tako komanda SELECT definisana sa triggerom *dbSelect* na ovom ODBC radnom listu vrši iščitavanje vrijednosti kolona jednog reda u tabeli čija vrijednost u *ID* koloni tabele odgovara vrijednosti taga *dbIDParametri*, zatim učitavanje tih vrijednosti u korespodentne tagove i njihovo prikazivanje na ekranu. Komanda INSERT definisana sa triggerom *dbInsert* vrši upisivanje novog reda u tabeli sa vrijednostima koje su unesene na ekranu i rednim brojem sloga koji je unesen u polje za unos *ID*. Komanda DELETE briše jedan red iz tabele (briše sve njemu pripadajuće vrijednosti kolona, uključujući i *ID*), čiji redni broj sloga odgovara vrijednosti taga *dbIDParametri*.

Drugi ODBC radni list pod nazivom *SelectAll i Next*, čiji je izvor podataka *SPSSGrupa07Konekcija2*, omogućava izvršavanje bezuslovnih SQL komandi SELECT i NEXT. Komanda SELECT definisana sa triggerom *dbSelectAll* vrši iščitavanje vrijednosti kolona svih redova u tabeli, zatim učitavanje vrijednosti kolona prvog reda tabele u korespodentne tagove i njihovo prikazivanje na ekranu. Komanda NEXT definisana sa triggerom *dbNext* vrši iščitavanje vrijednosti kolona jednog reda u tabeli čija vrijednost u *ID* koloni tabele odgovara vrijednosti narednog sloga u tabeli u odnosu na trenutni, zatim učitavanje vrijednosti kolona tog reda tabele u korespodentne tagove i njihovo prikazivanje na ekranu. Ovdje treba naglasiti da se "naredni" slog određuje u odnosu na zadnji slog iščitavan isključivo SQL komandom SELECT bez uslova. Komanda SELECT sa uslovom i njen rezultat se, u ovom slučaju, potpuno ignorišu. Na primjer, neka tabela sadrži pet redova, i neka je izvršena komanda SELECT bez uslova uspješno. Kao rezultat, na ekranu dobijamo vrijednosti prvog reda u tabeli. Ukoliko nakon ove komande primijenimo komandu NEXT, i ona se uspješno izvrši, kao rezultat ćemo dobiti prikaz vrijednosti kolona drugog reda u tabeli. Ako ponovimo istu komandu, dobićemo kao rezultat vrijednosti trećeg reda tabele. Izvršimo li sad komandu SELECT sa uslovom i kao vrijednost *ID* kolone navedemo onu koja odgovara prvom redu tabele, i komanda se uspješno izvrši, kao rezultat ćemo ponovo dobiti prikaz vrijednosti kolona prvog reda. Ako nakon ove komande primijenimo komandu NEXT,

kao rezultat ove komande (u slučaju uspješnog izvršavanja) ćemo dobiti vrijednosti četvrtog reda tabele, a ne drugog (kao što bismo mogli očekivati, jer se trenutno nalazimo na prvom redu tabele i njen sljedbenik je drugi red), zbog toga što je posljednji red na kojem smo stali prilikom izvršavanja komande SELECT bez uslova bio treći, čiji je sljedbenik četvrti red. Rezultat komande SELECT sa uslovom se u potpunosti ignorisao. Ako nakon ispisa vrijednosti četvrtog reda ponovimo komandu NEXT, na ekranu ćemo dobiti ispis vrijednosti petog reda. Naredna komanda NEXT (ukoliko bismo je primijenili) će biti neuspješna (jer smo došli do kraja tabele) i na ekranu će se ispisati poruka o grešci. Svaka naredna komanda NEXT će biti neuspješna, čak i ako ponovimo komandu SELECT sa uslovom i pozicioniramo se na neki od pet redova, sve dok ponovo ne izvršimo komandu SELECT bez uslova i ne vratimo se na početak tabele. Kako vidimo komanda NEXT se ne može izvršavati uspješno, ukoliko na istom ODBC radnom listu, na kojem je definisana, nije definisana i komanda SELECT bez uslova. Navedeni zaključak predstavlja jedini razlog definisanja komande SELECT bez uslova, koja bi se, da nije navedenog problema, mogla u potpunosti izbjeći (kako SELECT bez uslova, iako iščitava sve redove tabele, na ekranu prikazuje samo prvi, moguće ju je zaobići korištenjem komande SELECT sa uslovom navodeći kao vrijednost kolone *ID* onu koja odgovara prvom redu tabele).

Treći ODBC radni list pod nazivom *Update po koloni ID*, čiji je izvor podataka *SPSSGrupa07Konekcija3*, omogućava izvršavanje uslovne SQL komande UPDATE. Uslov *ID = {dbIDParametri}*, naveden u polju *Condition* u zaglavlju trećeg ODBC radnog lista definiše da se SQL komanda UPDATE izvršava nad samo jednim redom tabele u bazi i to onim čiji je redni broj u koloni *ID* u bazi jednak vrijednosti taga *dbIDParametri* unesenoj u polju za unos na ekranu *IdentifikacijaObjektaODBC*. Tako komanda UPDATE definisana sa triggerom *dbUpdate* na ovom ODBC radnom listu vrši ažuriranje vrijednosti kolona jednog reda u tabeli čija vrijednost u *ID* koloni tabele odgovara vrijednosti taga *dbIDParametri*. Ažuriranje vrijednosti kolona podrazumijeva prepisivanje trenutnih vrijednosti kolona datog reda u tabeli sa novim vrijednostima kolona unesenih na ekranu (ukoliko su se neke od vrijednosti kolona promijenile, stare će biti zamijenjene novim, a ukoliko nije došlo do promjene vrijednosti, komanda UPDATE će samo preko starih vrijednosti prepisati nove, koje su iste kao i stare – efektivno promjene neće biti). Za razliku od prethodna dva ODBC radna lista, na kojima su u tijelu radnog lista bile uključene sve kolone tabele, kod ovog radnog lista izbačena je kolona *ID*. Naime, komanda UPDATE sa uslovom bi pokušala da preko stare prepíše, također, i novu vrijednost kolone *ID* skupa sa vrijednostima ostalih kolona, a kako je polje *ID* primarni ključ tabele, ono se ne može ažurirati (nakon unosa sloga u tabelu polje nije moguće uopće mijenjati, osim obrisati čitav red iz tabele), čak ni ako to ažuriranje podrazumijeva prepisivanje iste vrijednosti ove kolone preko već postojeće. Ovo je ujedno i razlog zbog čega je ova komanda morala biti izdvojena u poseban radni list i nije mogla biti pripojena prvom ODBC radnom listu.

Pošto komanda NEXT za svoje izvršavanje zahtijeva da se prije nje uspješno izvrši komanda SELECT bez uslova, kreiran je MATH radni list pod nazivom *Selektovanje svih slogova iz baze podataka* koji se pokreće pri otvaranju ekrana *IdentifikacijaObjektaODBC* i koji izvršava komandu SELECT bez uslova. Na taj način je obezbjeđeno da se komanda NEXT može izvršavati odmah nakon otvaranja ekrana, kao i ostale komande, pod uslovom da se komanda SELECT bez uslova uspješno izvrši. Ovo ujedno omogućava da se polja za unos popune inicijalnim vrijednostima – vrijednostima iz baze – pri otvaranju ekrana.

Odabirom nekog od tastera prikazanih na slici 29. pokreće se predefinisana akcija izvršavanja neke od SQL komandi za manipulaciju nad gore navedenim podacima u bazi podataka. Koja se SQL komanda izvršava zavisi od toga koji je taster od ponuđenih odabran.



Slika 29: Izvršavanje SQL komandi za manipulaciju nad podacima u bazi podataka

Odabirom tastera *Izlistaj* pokreće se SQL komanda SELECT koja iščitava slog u bazi podataka čiji redni broj odgovara vrijednosti unesenoj u polje za unos taga *dbIDParametri* (ovdje je riječ o uslovnoj SELECT komandi, koja kao rezultat vraća samo jedan red tabele), te vrijednosti iščitane iz baze podataka učitava u aplikacione tagove koji korespondiraju sa kolonama u bazi podataka i prikazuje ih u poljima za unos navedenih tagova. Ukoliko je operacija uspješno izvršena, kao rezultat se dobiju navedene vrijednosti i poruka da je transakcija uspješno izvršena. Ukoliko operacija iz nekog razloga nije izvršena, na ekranu se prikazuje poruka o grešci koja se desila (kod greške i opis greške koja se desila).

Odabirom tastera *Sljedeći* pokreće se SQL komanda NEXT koja iščitava naredni slog u bazi podataka u odnosu na zadnji slog iščitao bezuslovnom komandom SELECT. Ukoliko se uspješno izvrši, kao rezultat se dobiju prikaz vrijednosti datog reda tabele u poljima za unos na ekranu. U slučaju greške, na ekranu se ispisuje poruka o grešci koja se desila (njen kod i opis). Kako je komanda NEXT vezana za komandu SELECT bez uslova (nakon poruke o grešci više se ne bi mogla izvršavati do narednog izvršavanja komande SELECT bez uslova), napravljen je mehanizam kojim se, nakon što se javi poruka o grešci pri izvršavanju komande NEXT (kod greške je 5), pri narednom odabiru tastera *Izlistaj* ponovo izvrši MATH radni list *Selektovanje svih slogova iz baze podataka* koji izvršava komandu SELECT bez uslova. Nakon ovoga komanda NEXT ponovo postaje omogućena. Navedena komanda će se izvršiti bez obzira da li je prilikom odabira tastera *Izlistaj* u polje za unos *ID* unesena validna vrijednost reda u tabeli ili ne. Ukoliko jeste, tada će se prvo izvršiti komanda SELECT sa uslovom i prikazat će se nakratko vrijednosti kolona pripadnog reda u tabeli, nakon čega će se izvršiti komanda SELECT bez uslova i prikazat će se vrijednosti kolona prvog reda u tabeli. Ukoliko nije, odmah će se izvršiti komanda SELECT bez uslova.

Odabirom tastera *Ubaci* pokreće se SQL komanda INSERT koja upisuje slog u bazu podataka čiji redni broj odgovara vrijednosti unesenoj u polje za unos taga *dbIDParametri* uvećanoj za 1 (ovdje je riječ o uslovnoj INSERT komandi, koja kao rezultat upisuje samo jedan red u tabelu), a vrijednosti ostalih kolona odgovaraju vrijednostima korespondentnih tagova unesenih na ekranu u poljima za unos. Inkrementirana vrijednost taga *dbIDParametri* se odmah prikazuje na ekranu. Samo uvećavanje vrijednosti ovog polja se dešava prilikom odabira tastera INSERT. Na ovaj način je obezbijeđeno da korisnik ne pokuša slučajno unijeti više od jednog sloga u bazu sa istom vrijednošću kolone *ID*. Ukoliko je operacija uspješno izvršena, kao rezultat se navedene vrijednosti kolona upisuju u bazu podataka i prikazuje se poruka da je transakcija uspješno izvršena. Ukoliko operacija iz nekog razloga nije izvršena, na ekranu se prikazuje poruka o grešci koja se desila (kod greške i opis greške koja se desila).

Odabirom tastera *Obriši* pokreće se SQL komanda DELETE koja briše jedan slog u bazi podataka čiji redni broj odgovara vrijednosti unesenoj u polje za unos taga *dbIDParametri* (ovdje je riječ o uslovnoj DELETE komandi, koja kao rezultat briše samo jedan red tabele. Ukoliko je operacija uspješno izvršena, kao rezultat se obriše navedeni slog (brišu se sve vrijednosti) iz baze podataka i prikazuje se poruka da je transakcija uspješno izvršena. Ukoliko operacija iz nekog razloga nije izvršena, na ekranu se prikazuje poruka o grešci koja se desila (kod greške i opis greške koja se desila).

Odabirom tastera *Ažuriraj* pokreće se SQL komanda UPDATE koja ažurira jedan slog u bazi podataka čiji redni broj odgovara vrijednosti unesenoj u polje za unos taga *dbIDParametri* (ovdje je riječ o uslovnoj UPDATE komandi, koja kao rezultat ažurira samo

jedan red tabele), pri čemu se vrši ažuriranje vrijednosti svih kolona, osim kolone *ID*. Ukoliko je operacija uspješno izvršena, kao rezultat se ažuriraju navedene vrijednosti i prikazuje se poruka da je transakcija uspješno izvršena. Ukoliko operacija iz nekog razloga nije izvršena, na ekranu se prikazuje poruka o grešci koja se desila (kod greške i opis greške koja se desila).

Na slici 30. je prikazan displej statusa ODBC konekcije. U okviru njega, prikazuju se tri podatka: kod greške koja se desila prilikom pokušaja izvršavanja transakcije nad bazom podataka (vrijednost je jednaka 0, ako greške nije bilo; svaka cjelobrojna vrijednost iznad 1 indicira da se neka greška desila), SQL komandu koja se pokušala izvršiti u trenutku pojavljivanja greške, te opis greške koja se desila (prikazuje se u kojoj se SQL funkciji greška desila), u zavisnosti od koda greške koja se desila.

Ukoliko greške nije bilo, navedeni podaci će biti ispisani žutom bojom; ukoliko se greška desila, podaci će biti ispisani crvenom bojom.



Slika 30: Status ODBC konekcije

Polje za ispis *Komanda* ispisuje naziv posljednje SQL komande koja se pokušala izvršiti. Ovo polje se ažurira svaki put kad se odabere jedan od tastera *Izlistaj*, *Sljedeći*, *Ubaci*, *Obriši* i *Ažuriraj*, tj. pritiskom na navedene tastere u polje *Komanda* se upisuje jedna od vrijednosti SELECT, NEXT, INSERT, DELETE ili UPDATE respektivno. Ovo polje se, također, skupa sa poljem *Opis greške* ažurira pri izvršavanju svake SQL komande, tako što se vrijednosti tih polja iščitavaju iz fajla *ODBCGreske.dat* sa hard diska u skladu sa odgovarajućim kodom greške i prikazuju na spomenutom displeju. Da bi se ova funkcionalnost zadovoljila, kreirana je receptura pod nazivom *ODBCGreske*, koja smješta podatke u i iščitava podatke iz fajla *ODBCGreske.dat*. Podaci predstavljaju dva taga niza stringova: *Komanda* i *OpisGreske*, koji sadrže sve SQL komande koje se mogu izvršiti i sve greške koje se mogu javiti pri izvršenju tih komandi (u skladu sa kodom greške koja se desila). Kako se može desiti ukupno 14 različitih grešaka povezanih sa pet navedenih SQL komandi plus poruka o tome da nema greške, oba navedena niza stringova imaju po 15 elemenata: 14 za greške i 1 za slučaj da se greška nije desila. Za svrhu učitavanja date recepture, kreiran je MATH radni list pod nazivom *ODBC status transakcija*, koji omogućava učitavanje podataka zapisanih u recepturi u korespondentne tagove u aplikaciji: *Komanda* i *OpisGreske*. Ovaj MATH radni list se izvršava prilikom pokretanja ekrana *IdentifikacijaObjektaODBC*.

Odabirom tastera *Glavni ekran* zatvara se ekran *IdentifikacijaObjektaODBC* i otvara ekran *Glavni*.

## SISTEMSKI DATUM I VRIJEME

Nakon odabira tastera *Sistemska datum i vrijeme* na glavnom ekranu aplikacije otvara se ekran *Datum/Vrijeme* na kojem je autorizovanom korisniku omogućeno postavljanje sistemskog datuma i vremena, te prikaz navedenih veličina (slika 31.).



Slika 31: DatumIvrijeme.scr

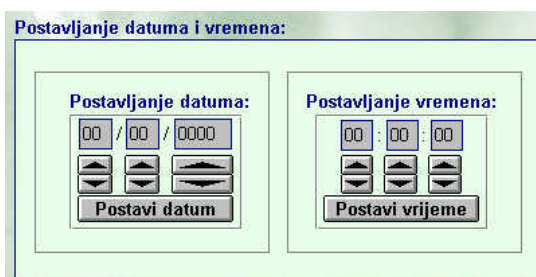
Pristup ovom ekranu imaju samo korisnici aplikacije iz grupe korisnika *Admin* (ostalim korisnicima je pristup zabranjen tako što je na glavnom ekranu aplikacije na tasteru *Sistemski datum i vrijeme* postavljen sigurnosni nivo 7).

Polja prikazana na slici 32. služe za prikaz sistemskog datuma i vremena. Pri pokretanju ekrana, navedena polja sadrže inicijalne vrijednosti koje predstavljaju tekući sistemski datum i vrijeme. Također se, pri pokretanju ekrana, tekući sistemski datum i vrijeme pamte u aplikacione tagove *SistemskiDatum* i *SistemskoVrijeme* respektivno.



Slika 32: Prikaz datuma i vremena

Polja prikazana na slici 33. služe za postavljanje sistemskog datuma i vremena. Pošto ova operacija mijenja datum i vrijeme sistema (računara) na kojem se aplikacija izvršava, ova operacija je dozvoljena samo autorizovanom korisniku (samo korisnicima iz sigurnosne grupe *Admin*).



Slika 33: Postavljanje datuma i vremena

Postavljanje datuma i vremena se može izvršiti direktnim unošenjem vrijednosti u polja za unos prikazana na gornjoj slici, ili odabiranjem tastera 'gore/dole' ispod polja za unos (tasteri 'gore' uvećavaju trenutnu vrijednost za 1, a tasteri 'dole' smanjuju trenutnu vrijednost za 1). Format datuma je: *DD/MM/YYYY* (npr. 08/05/2005), gdje D predstavlja oznaku za dan, M za mjesec, a Y za godinu. Znak "/" se koristi kao razdjelnik (delimiter). Format za vrijeme je *HH:MM:SS* (npr. 15:22:35), gdje H predstavlja oznaku za sat, M za minutu, a S za sekundu. Znak ":" se koristi kao razdjelnik (delimiter).

Opsezi vrijednosti za navedene veličine su sljedeći:

- dan: 1 – 31
- mjesec: 1 – 12
- godina: 1970 – 2100
- sat: 00 – 23
- minuta: 00 – 59
- sekunda: 00 – 59.

Nakon izvršenog unosa navedenih vrijednosti, odabirom *tastera Postavi datum* za datum, odnosno *Postavi vrijeme* za vrijeme, vrši se postavljanje sistemskog datuma/vremena. Promjene se odmah apliciraju na sistemski datum i vrijeme.

Kod unosa datuma se vrši provjera ispravnosti datuma. To podrazumijeva da se vrši kontrola sljedećih vrijednosti:

- ukoliko je uneseni mjesec april (MM = 04), juni (MM = 06), septembar (MM = 09) ili novembar (MM = 11), koji imaju po 30 dana, a korisnik unese kao dan 31 (DD = 31), dan se koriguje u 30 (DD = 30)
- ukoliko je uneseni mjesec februar (MM = 02) i godina nije prestupna, i korisnik unese kao dan 29, 30 ili 31 (DD = 29 ili DD = 30 ili DD = 31), dan se koriguje u 28 (DD = 28)
- ukoliko je uneseni mjesec februar (MM = 02) i godina je prestupna, i korisnik unese kao dan 30 ili 31 (DD = 30 ili DD = 31), dan se koriguje u 29 (DD = 29).

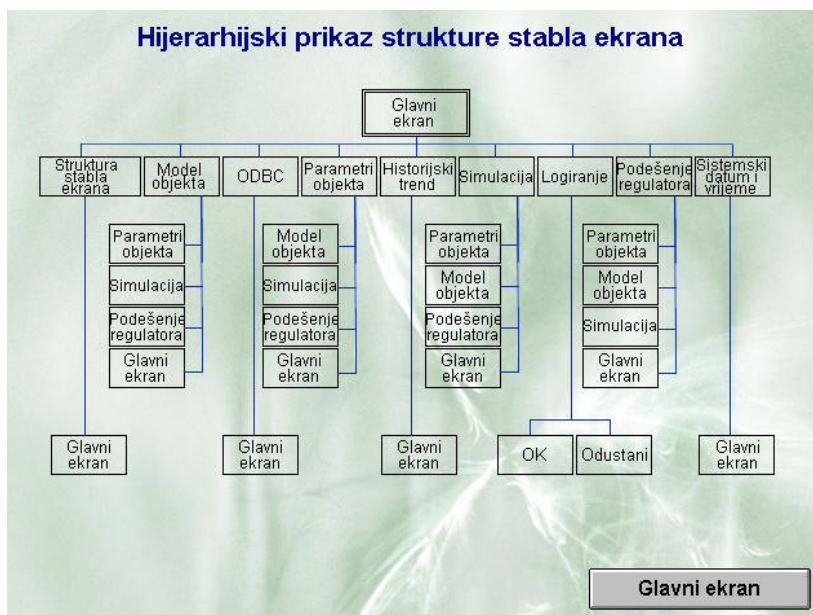
Ukoliko se neispravne vrijednosti unose direktno u polja ta unos, njihova korekcija se vrši prilikom pokušaja postavljanja sistemskog datuma, tj. prilikom odabira dugmeta *Postavi datum*. Ukoliko se vrijednosti za datum unose odabiranjem tastera 'gore/dole', korekcija neispravnih vrijednosti se vrši odmah, tj. korisniku se ne dozvoljava da unese neispravne vrijednosti. Navedene ispravke se vrše tako što se za njihovo izvršenje poziva MATH radni list pod nazivom *Provjera ispravnosti datuma*.

Odabirom tastera *Glavni ekran* zatvara se ekran *Datum/Vrijeme* i otvara ekran *Glavni*. Osim toga, izvršava se operacija postavljanja sistemskog datuma i vremena na vrijednosti navedenih veličina koje su bile u trenutku otvaranja ekrana *Datum/Vrijeme* (ove vrijednosti su prilikom otvaranja ekrana bile spašene u tagove *SistemskiDatum* i *SistemskoVrijeme*, i iz tih tagova se iščitavaju i postavljaju).

## STRUKTURA STABLA EKRANA

Nakon odabira tastera *Struktura stabla ekrana* na glavnom ekranu aplikacije otvara se ekran *Struktura ekrana* na kojem se nalazi hijerarhijski prikaz strukture stabla ekrana sa naznačenim tasterom za selekciju svakog nižeg nivoa (slika 34.).





Slika 34: Hijerarhijski prikaz strukture stabla ekrana

Pristup ovom ekranu imaju svi korisnici aplikacije (bez obzira na sigurnosnu grupu kojoj pripadaju).

Odabirom tastera *Glavni ekran* zatvara se ekran *Struktura ekrana* i otvara ekran *Glavni*.

## TCP/IP

Za potrebe demonstriranja rada aplikacije na više računara u mreži u multiple klijent – server konfiguraciji, omogućen je razmjena OPC tagova između računara u mreži korištenjem TCP/IP protokola.

Da bi se aplikacija mogla u potpunosti izvršavati na više računara u mreži, potrebno je da svaki računar u mreži ima vlastitu instalaciju IWS softverskog paketa, kao i instalaciju softverskog paketa Matlab 7.0, razvijenu aplikaciju i konačno, konfigurisani TCP/IP radni list (ovo vrijedi za TCP/IP klijente u mreži). Na računaru koji će biti TCP/IP server dovoljno je samo pokrenuti *TCP/IP Server* (postaviti mu startup opciju na *Automatic*).

Putem TCP/IP protokola se razmjenjuju sljedeći tagovi: *ilnicijalizacijaZadanaVrijednost*, *ilnicijalizacijaKob*, *ilnicijalizacijaTob*, *ilnicijalizacijaKsi*, *ilnicijalizacijaTau*, *iUlazKr*, *iUlazTi*, *iUlazTd* i *ilzlazlzlaz*.

## WEB TANKI KLIJENTI

Za potrebe demonstriranja rada aplikacije sa Web tankim klijentima (Web thin client) u mreži sa Web browser pristupom, omogućen je prikaz ekrana aplikacije kao .html fajlova u Web browseru.

Iz navedenog razloga, kreirani su .html fajlovi iz ekrana aplikacije (po jedan .html fajl za svaki ekran). Da bi se aplikacija mogla u potpunosti izvršavati u mreži sa Web pristupom, potrebno je na računaru koji će biti Web server u mreži podesiti u projektnim postavkama adresu tog računara, zatim zaustaviti MS IIS Web servis i pokrenuti *NTWebServer.exe* koji se nalazi u folderu Web ove aplikacije.





## A Kod Matlab M-fileova

```
%
% azuriraj_ulaze.m
%

Kr = PojacanjeRegulatora.value
Ti = IntegralnoVrijeme.value
Td = DerivativnoVrijeme.value
zadanaVrijednost = ZadanaVrijednost.value
Kob = Pojacanje.value
Tob = VremenskaKonstanta.value
ksi = FaktorPrigusenja.value
tau = IznosCistogKasnjenja.value

%
% inicijalizacija.m
%

opcreset
clear mpcopcPlantStep
clear mpcopcMPCStep
try
    Server = opcda('localhost','Studio.Scada.OPC');
    connect(Server);
catch
    error('Server nije aktivan!')
end

% kreiranje grupa i "OPC Item" objekata...

Citanje = addgroup(Server,'Citanje');
PojacanjeRegulatora = additem(Citanje,'iUlazKr');
IntegralnoVrijeme = additem(Citanje,'iUlazTi');
DerivativnoVrijeme = additem(Citanje,'iUlazTd');
ZadanaVrijednost = additem(Citanje,'ilnicijalizacijaZadanaVrijednost');
%Kob
    Pojacanje = additem(Citanje,'ilnicijalizacijaKob');
%Tob
    VremenskaKonstanta = additem(Citanje,'ilnicijalizacijaTob');
%ksi
    FaktorPrigusenja = additem(Citanje,'ilnicijalizacijaKsi');
%tau
    IznosCistogKasnjenja = additem(Citanje,'ilnicijalizacijaTau');

KontrolaMatlab = addgroup(Server,'KontrolaMatlab');
MatlabPokrenut = additem(KontrolaMatlab,'kMatPokrenut');

KontrolaSimulacija = addgroup(Server,'KontrolaSimulacija');
ModelAktivan = additem(KontrolaSimulacija,'kModelAktivan');

KontrolaRjesavanje = addgroup(Server,'KontrolaRjesavanje');
TraziRjesenje = additem(KontrolaRjesavanje,'kRjesavanje');

KontrolaPodesavanje = addgroup(Server,'KontrolaPodesavanje');
Podesavanje = additem(KontrolaPodesavanje,'kPodesenje');

Pisanje = addgroup(Server,'Pisanje');
PrenosnaFunkcija = additem(Pisanje,'ilzlazzlaz');
TrenutniPeriod = additem(Pisanje,'ilzlazPeriod');
TrenutniFaktorPrigusenja = additem(Pisanje,'ilzlazFaktorPrigusenja');

Klase = addgroup(Server,'Klase');
UlazKr0 = additem(Klase,'iUlazEksperiment[0].iUlazKr');
set(UlazKr0,'DataType','double');
UlazKr1 = additem(Klase,'iUlazEksperiment[1].iUlazKr');
set(UlazKr1,'DataType','double');
UlazKr2 = additem(Klase,'iUlazEksperiment[2].iUlazKr');
set(UlazKr2,'DataType','double');
UlazTi0 = additem(Klase,'iUlazEksperiment[0].iUlazTi');
set(UlazTi0,'DataType','double');
UlazTi1 = additem(Klase,'iUlazEksperiment[1].iUlazTi');
set(UlazTi1,'DataType','double');
UlazTi2 = additem(Klase,'iUlazEksperiment[2].iUlazTi');
```

```

set(UlazTi2,'DataType','double');
UlazTd0 = additem(Klase,'iUlazEksperiment[0].iUlazTd');
set(UlazTd0,'DataType','double');
UlazTd1 = additem(Klase,'iUlazEksperiment[1].iUlazTd');
set(UlazTd1,'DataType','double');
UlazTd2 = additem(Klase,'iUlazEksperiment[2].iUlazTd');
set(UlazTd2,'DataType','double');
IzlazPeriod0 = additem(Klase,'iUlazEksperiment[0].ilzazPeriod');
set(IzlazPeriod0,'DataType','double');
IzlazPeriod1 = additem(Klase,'iUlazEksperiment[1].ilzazPeriod');
set(IzlazPeriod1,'DataType','double');
IzlazPeriod2 = additem(Klase,'iUlazEksperiment[2].ilzazPeriod');
set(IzlazPeriod2,'DataType','double');

KlasaIdentifikacija = addgroup(Server,'KlasaIdentifikacija');
InicijalizacijaKob = additem(KlasaIdentifikacija,'iIdentificiraniParametriObjekta.iInicijalizacijaKob');
set(InicijalizacijaKob,'DataType','double');
InicijalizacijaTob = additem(KlasaIdentifikacija,'iIdentificiraniParametriObjekta.iInicijalizacijaTob');
set(InicijalizacijaTob,'DataType','double');
InicijalizacijaKsi = additem(KlasaIdentifikacija,'iIdentificiraniParametriObjekta.iInicijalizacijaKsi');
set(InicijalizacijaKsi,'DataType','double');
InicijalizacijaTau = additem(KlasaIdentifikacija,'iIdentificiraniParametriObjekta.iInicijalizacijaTau');
set(InicijalizacijaTau,'DataType','double');

KlasaParametriPodesenja = addgroup(Server,'ParametriPodesenja');
UlazKr3 = additem(KlasaParametriPodesenja,'iParametriPodesenja.iUlazKr');
set(UlazKr3,'DataType','double');
IzlazPeriod3 = additem(KlasaParametriPodesenja,'iParametriPodesenja.ilzazPeriod');
set(IzlazPeriod3,'DataType','double');

KlasaPodesenje = addgroup(Server,'KlasaPodesenje');
UlazKr4 = additem(KlasaPodesenje,'iPodesenjeRegulatora.iUlazKr');
set(UlazKr4,'DataType','double');
UlazTi4 = additem(KlasaPodesenje,'iPodesenjeRegulatora.iUlazTi');
set(UlazTi4,'DataType','double');
UlazTd4 = additem(KlasaPodesenje,'iPodesenjeRegulatora.iUlazTd');
set(UlazTd4,'DataType','double');

%
% dodjeljivanje tipa & default vrijednosti OPC Item varijablama...
%

set(PojacanjeRegulatora,'DataType','double');
write(PojacanjeRegulatora, 1.36);
set(IntegralnoVrijeme,'DataType','double');
write(IntegralnoVrijeme, 10);
set(DerivativnoVrijeme,'DataType','double');
write(DerivativnoVrijeme, 0.55);
set(ZadanaVrijednost,'DataType','double');
write(ZadanaVrijednost, 1);
set(Pojacanje,'DataType','double');
write(Pojacanje, 2);
set(VremenskaKonstanta,'DataType','double');
write(VremenskaKonstanta, 1);
set(FaktorPrigusenja,'DataType','double');
write(FaktorPrigusenja, 1.5);
set(IznosCistogKasnjenja,'DataType','double');
write(IznosCistogKasnjenja, 2);

set(PrenosnaFunkcija,'DataType','double');
set(TrenutniPeriod,'DataType','double');
set(TrenutniFaktorPrigusenja,'DataType','double');
set(MatlabPokrenut,'DataType','logical');
set(ModelAktivan,'DataType','logical');
set(TraziRjesenje,'DataType','logical');
set(Podesavanje,'DataType','logical');

%
% pokretanje m fajlova u MatLabu Event-om sa OPC servera:
%

Citanje.DataChangeFcn = 'azuriraj_ulaze'
KontrolaMatlab.DataChangeFcn = 'kontrolaMatlab'
KontrolaSimulacija.DataChangeFcn = 'kontrolaSimulacija'
KontrolaRjesavanje.DataChangeFcn = 'kontrolaRjesavanja'
KontrolaPodesavanje.DataChangeFcn = 'kontrolaPodesenje'

```

```
% Neke globalne varijable:  
trajanjeSimulacije = 1000
```

```
write(MatlabPokrenut, 1);
```

---

```
%  
% kontrolaMatlab.m  
%
```

```
if (MatlabPokrenut.Value == 0)  
    clc;  
    display('Izlazim iz matlaba!');  
    pause(0.1);  
    exit;  
end;
```

---

```
%  
% kontrolaPodesenje.m  
%
```

```
if(Podesavanje.value == 1)  
    Podaci = [UlazKr3.value, IzlazPeriod3.value];  
    [Kr, Ti, Td] = podesenje (Podaci);  
    write(UlazKr4, Kr);  
    write(UlazTi4, Ti);  
    write(UlazTd4, Td);  
    write(Podesavanje, 0);  
end;
```

---

```
%  
% kontrolaRjesavanja.m  
%
```

```
if(TraziRjesenje.value == 1)  
    Podaci = [UlazKr0.value, UlazTi0.value, UlazTd0.value, IzlazPeriod0.value;  
              UlazKr1.value, UlazTi1.value, UlazTd1.value, IzlazPeriod1.value;  
              UlazKr2.value, UlazTi2.value, UlazTd2.value, IzlazPeriod2.value];  
    [Kob, Tob, Ksi, Tau] = proracun (Podaci);  
    write(InicijalizacijaKob, Kob);  
    write(InicijalizacijaTob, Tob);  
    write(InicijalizacijaKsi, Ksi);  
    write(InicijalizacijaTau, Tau);  
    write(TraziRjesenje, 0);  
end;
```

---

```
%  
% kontrolaSimulacija.m  
%
```

```
clc  
display('Kontrola simulacije je aktivirana!');
```

```
if (ModelAktivan.Value == 1)  
    sim('model', trajanjeSimulacije);  
    [PeriodOscilovanja, Sigma] = period (tout, Izlaz);  
    write(TrenutniPeriod, PeriodOscilovanja);  
    write(TrenutniFaktorPrigusenja, Sigma);  
    for i=1:50:(100*trajanjeSimulacije+1)  
        pause(1);  
        write(PrenosnaFunkcija, Izlaz (i));  
        if (ModelAktivan.value == 0)  
            break;  
        end;  
    end;  
    write(ModelAktivan, 0);  
end;  
display('Simulacija završena!');
```

---

```

%
% period.m
%

function [PeriodOscilovanja, FaktorPrigusenja] = period(vrijeme,izlaz);

n=size(izlaz);
stacionarna=mean(izlaz);
presjeci=[]; vrhovi=[]; razmjere=[];

for i=1:n-1,
    if (izlaz(i)-stacionarna)*(izlaz(i+1)-stacionarna)<=0,
        presjeci=[presjeci,(vrijeme(i)+vrijeme(i+1))/2];
        end;
    if i>1 & (izlaz(i-1)<izlaz(i)) & (izlaz(i)>izlaz(i+1)),
        vrhovi=[vrhovi,izlaz(i)-stacionarna];
        end
    end;

poluperiodi=diff(presjeci);

for i=1:length(vrhovi)-1,
    razmjere=[razmjere,vrhovi(i+1)/vrhovi(i)];
end;

PeriodOscilovanja=2*mean(poluperiodi);
FaktorPrigusenja=-log(mean(razmjere))/PeriodOscilovanja;

```

---

```

%
% podesenje.m
%

function [Kr, Ti, Td] = podesenje(podaci);

Kr1=podaci(1,1);
T=podaci(1,2);

Kr = 0.6*Kr1;
Ti = 0.5*T;
Td = 0.125*T;

% Ako nije pogodno da parametar Td ima vrijednost vecu od 0 (u slucaju da objekat
% ima puno smetnji, sto nije slucaj za testirani objekat), parametre regulatora
% treba postaviti na sljedece vrijednosti: Kr = 0.4*Kr1; Ti = 0.8*T; Td = 0;

```

---

```

%
% proracun.m
%

function [Kob, Tob, Ksi, Tau] = proracun(podaci);

Kr1=podaci(1,1); Ti1=podaci(1,2); Td1=podaci(1,3); w1=2*pi/podaci(1,4);
Kr2=podaci(2,1); Ti2=podaci(2,2); Td2=podaci(2,3); w2=2*pi/podaci(2,4);
Kr3=podaci(3,1); Ti3=podaci(3,2); Td3=podaci(3,3); w3=2*pi/podaci(3,4);

matrica=[Kr1^2*(1+(Td1*w1-1/(Ti1*w1))^2),-2*w1^2,-w1^4;
    Kr2^2*(1+(Td2*w2-1/(Ti2*w2))^2),-2*w2^2,-w2^4;
    Kr3^2*(1+(Td3*w3-1/(Ti3*w3))^2),-2*w3^2,-w3^4];

rjesenja=matrica\ [1;1;1];

alfa=rjesenja(1); beta=rjesenja(2); gama=rjesenja(3);

Kob=sqrt(alfa);
Tob=sqrt(sqrt(gama));
Ksi=sqrt((1+beta/Tob^2)/2);

j=sqrt(-1);
tau1=(pi+angle((1+1/(Ti1*j*w1)+Td1*j*w1)/(2*Tob*Ksi*j*w1+1-Tob^2*w1^2)))/w1;
tau2=(pi+angle((1+1/(Ti2*j*w2)+Td2*j*w2)/(2*Tob*Ksi*j*w2+1-Tob^2*w2^2)))/w2;
tau3=(pi+angle((1+1/(Ti3*j*w3)+Td3*j*w3)/(2*Tob*Ksi*j*w3+1-Tob^2*w3^2)))/w3;
Tau=mean([tau1,tau2,tau3]);

```

## B Varijable i tagovi

| Matlab okruženje         |   |
|--------------------------|---|
| Varijabla                | Opis                                    |
| Citanje                  |   |
| PojacanjeRegulatora      | Pojačanje regulatora (Kr)               |
| IntegralnoVrijeme        | Integralno vrijeme (Ti)                 |
| DerivativnoVrijeme       | Derivativno vrijeme (Td)                |
| ZadanaVrijednost         | Zadana vrijednost (zadanaVrijednost)    |
| Pojacanje                | Pojačanje (Kob)                         |
| VremenskaKonstanta       | Vremenska konstanta (Tob)               |
| FaktorPrigušenja         | Faktor prigušenja (Ksi)                 |
| IznosCistogKasnjenja     | Iznos čistog kašnjenja (Tau)            |
| KontrolaMatlab           |   |
| MatlabPokrenut           | Da li je Matlab pokrenut ili ne         |
| KontrolaSimulacija       |   |
| ModelAktivan             | Da li je Simulink model pokrenut ili ne |
| KontrolaRjesavanje       |   |
| TraziRjesenje            |   |
| KontrolaPodesavanje      |   |
| Podesavanje              |   |
| Pisanje                  |   |
| PrenosnaFunkcija         |   |
| TrenutniPeriod           |   |
| TrenutniFaktorPrigušenja |   |
| Klase                    |   |
| UlazKr0                  |   |
| UlazKr1                  |   |
| UlazKr2                  |   |
| UlazTi0                  |   |
| UlazTi1                  |   |
| UlazTi2                  |   |
| UlazTd0                  |   |
| UlazTd1                  |   |
| UlazTd2                  |   |
| IzlazPeriod0             |   |
| IzlazPeriod1             |   |
| IzlazPeriod2             |   |
| KlasaIdentifikacija      |   |
| KlasaIdentifikacija      |   |
| InicijalizacijaKob       |   |
| InicijalizacijaTob       |   |
| InicijalizacijaKsi       |   |
| InicijalizacijaTau       |   |
| KlasaParametriPodesenja  |   |
| UlazKr3                  |   |
| IzlazPeriod3             |   |
| KlasaPodesenje           |   |
| UlazKr4                  |   |
| UlazTi4                  |   |
| UlazTd4                  |   |
| Simuink                  |   |
| Varijabla                | Opis                                    |
| Kr                       | Pojačanje regulatora                    |
| Ti                       | Integralno vrijeme                      |
| Td                       | Derivativno vrijeme                     |
| zadanaVrijednost         | Zadana Vrijednost                       |
| Kob                      | Pojačanje                               |
| Tob                      | Vremenska konstanta                     |
| ksi                      | Faktor prigušenja                       |
| tau                      | Iznos čistog kašnjenja                  |
| Izlaz                    | Izlaz                                   |
| InduSoft Web Studio      |   |
| Tag                      | Opis                                    |
|                          |   |
|                          |   |
|                          |   |
|                          |   |
|                          |   |
|                          |   |

