

KATEDRA ZA INFORMATIKU U INŽINJERSTVU



ZAVRŠNI RAD

Sa temom:

PROGRAMSKI PAKET LABVIEW U RAČUNARSKI PODRŽANOM SISTEMU MERENJA I UPRAVLJANJA

SADRŽAJ

1.	Uvod.....	3
2.	LabVIEW	4
2.1.	Uvod u LabVIEW	4
2.2.	Pogled na LabVIEW	4
2.3.	Osnovni koncepti LabVIEW-a	5
2.3.1.	Front panel toolbar.....	6
2.3.2.	Paleta alata	8
2.4.	Primer za promenu opsega termometra.....	9
2.5.	Paleta kontrola	10
3.	KONTROLE I INDIKATORI.....	11
3.1.	Numeričke kontrole i indikatori	11
3.2.	Bulova kontrola i indikatori	11
3.3.	Konfigurisanje kontrola i indikatora	11
3.4.	Paleta funkcija	12
4.	BLOK DIJAGRAM.....	13
4.1.	Tehnike povezivanja (Wiring techniques).....	13
4.2.	“Trake saveta”	14
4.3.	Istezanje provodnika.....	15
4.4.	Obeležavanje i brisanje provodnika	15
4.5.	“Loši provodnici”	16
5.	STARTOVANJE VI-a	17
5.1.	Dokumentovanje VI-a	17
6.	KRATAK PREGLED	19
7.	NI6008 KARTICA.....	20
7.1.	Ulazno/izlazni konektor.....	22
7.1.1.	Analogni konektor	22
7.1.2.	Digitalni konektor	23
7.2.	Povezivanje kartice sa PC-jem i objektom upravljanja.....	24
8.	OPIS SISTEMA	25
9.	KREIRANJE TASKOVA.....	26
10.	SNIMANJE PROCESA.....	29
11.	IDENTIFIKACIJA PROCESA	32
11.1.	IDENTIFIKACIJA PRVOG SENZORA.....	34
11.2.	IDENTIFIKACIJA POSLEDNJEG SENZORA.....	36
12.	UPRAVLJANJE PROCESOM	37
13.	PERIODA ODABIRANJA	38
14.	PODEŠAVANJE PARAMETARA PID KONTROLERA.....	42
15.	ZAKLJUČAK.....	46
16.	LITERATURA	47

1. Uvod

Zadatak ovog završnog rada je da uočimo razliku u upravljanju temperaturom na prvom i poslednjem senzoru korišćenjem softverskog alata LabVIEW i određivanje koeficijenata za podešavanje PID kontrolera. Podešavanje koeficijenata PID kontrolera vršeno je korišćenjem obrazaca iz knjiga "Računarski podržano merenje i upravljanje" i "Handbook of PI and PID Controller tuning rules 1".

Laboratorijski setup se sastoji od mašine PT400 koja je preko NI6008 kartice povezan sa računarom. Korisnički interfejs kojim se upravlja napravljen je u LabVIEW. Korišćenjem softvera MATLAB biće izvršena identifikacija objekta upravljanja, a zatim na osnovu ove identifikacije podesićemo PID kontroler. Nakon toga biće napravljen upravljački algoritam koji će pokazati kako se sistem ponaša u povratnoj sprezi pri normalnom radu, kao i pri dejstvu poremećaja.

2. LabVIEW

2.1. Uvod u LabVIEW

LabVIEW (skraćenica od Laboratory Virtual Instrumentation Engineering Workbench) je platforma i razvojno okruženje za vizuelne programske jezike iz National Instruments-a. U grafičkom jeziku se zove "G". Originalno izdata za Apple Macintosh 1986. god, Lab View se obično koristi za prikupljanje podataka, kontrolu instrumenata i industrijske automatike na različitim platformama uključujući Microsoft Windows, različite tipove UNIX-a, Linux-a, i Mac OS-a. Najnovija verzija LabView-a je verzija 8.6.1, izdata februara 2009. god.

LabVIEW je programski paket posebno izrađen tako da dozvoljava brzu ugradnju kompjuterskog prikupljanja podataka i sistema za njihovu analizu koji mogu biti široko prilagođeni potrebama korisnika paketa. Programski paket se odlikuje velikim mogućnostima ali se znatno razlikuje od bilo kog programskog paketa sa kojim se programer može sresti u svom dosadašnjem radu. Da bi početak savladavanja programskog paketa bio što jednostavniji i brži, postoji nekoliko ograničenja i pretpostavki koji se postavljaju pred buduće korisnike :

- Poznavanje koncepta kompjuterskog sistema kao što su *file*, *čuvanje fajlova i direktorijuma*, *štampanje* itd.
- Poznavanje GUT (*Graphical User Interface*) kao i korišćenje miša i tastature
- Naglasak je dat na tehnike prikupljanja podataka a ne na "lepom" programiranju (važno je dobro obaviti posao prikupljanja podataka a ne elegantnost obrade)

2.2. Pogled na LabVIEW

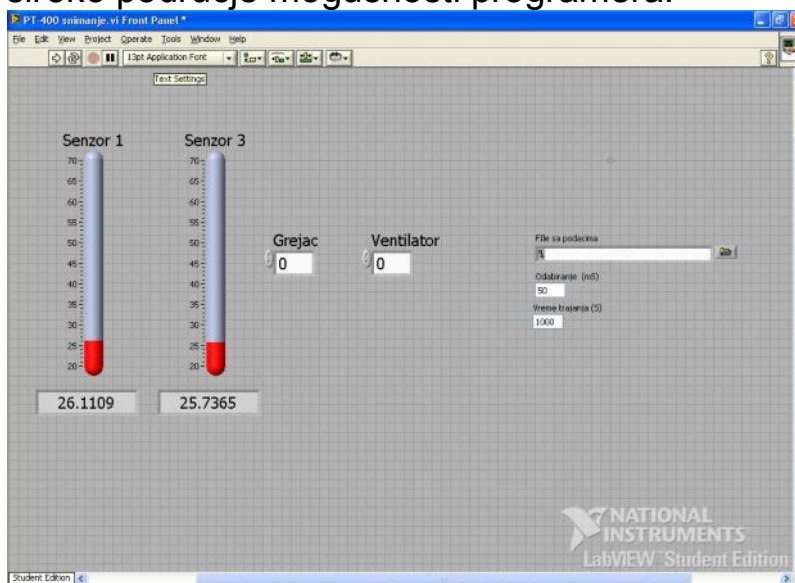
LabVIEW je u potpunosti grafički program koji izgleda poput crteža elektornske šeme sa jedne strane i stila električnih mernih uređaja iz pedesetih godina dvadesetog veka. To čini concept blok dijagrama (block diagram) i prednje strane (front pannel). LabVIEW je hijerarhijski program, u smislu da se bilo koji virtuelni instrument (bilo koja potpuna funkcionalna celina naziva se virtuelni instrument i uvek se označava sa VI) koji se porgramira može brzo pretvoriti u modul koji može biti potprogram drugog VI. To je u potpunosti analogno konceptu programiranja kod konvencionalnih programskih jezika.

LabVIEW je dizajniran da se može proširivati. Mogu da se dodaju moduli pomoću različitih sredstava. Proizvođač merne kartice ili nekog mernog instrumenta može programirati LabVIEW driver koji se ponaša kao *VI* koji predstavlja mernu karticu i njenu funkcionalnost u LabVIEW okruženju. Korisnik može takođe sam napisati LabVIEW modul i koristiti ga kao *VI* u drugim programima ili može napisati modul koji se slaže sa LabVIEW u drugim jezicima kao što su C i C++. Oni su poznati kao *podVI* i ne razlikuju se od *VI* osim što je interfejs određen za drugi nivo. PodVI u C ili C++ su vrlo korisni ukoliko je potrebno izvesti vrlo složeni numerički postupak nad podacima koji nisu pokriveni u standardnoj LabVIEW rutini.

2.3. Osnovni koncepti LabVIEW-a

Kao što je pomenuto ranije postoje dva izgleda bilo kog LabVIEW-a virtuelnog instrumenta. To su *block diagram* i *front panel*.

Front pannel je ono što korisnik vidi pri radu sa mernim sistemom. On sadrži kontrole i indikatore. LabVIEW ima vrlo bogat *menu* jednih i drugih (čak se može dizajnirati i sopstvena kontrola ili indicator) i to omogućava široko područje mogućnosti programera.



Slika 1 - Izgled front panel-a

Kao što je pomenuto ranije postoje dva izgleda bilo kog LabVIEW-a virtuelnog instrumenta. To su *block diagram* i *front panel*.

Front panel je ono što korisnik vidi pri radu sa mernim sistemom. On sadrži kontrole i indikatore. LabVIEW ima vrlo bogat *menu* jednih i drugih (čak se može dizajnirati i sopstvena kontrola ili indicator) i to omogućava široko područje mogućnosti programera.

2.3.1. Front panel toolbar

Front panel sadrži toolbar (paletu sa alatima) komandnih dugmića i status indikatora koje koristimo za startovanje i otklanjanje grešaka (debugiranje) VI-a.

Front Panel paleta:



- Dugme za pokretanje – pokreće VI



- Dugme za kontinuirano pokretanje – pokreće VI stalno iznova (korisno za debugiranje)



- Dugme za zaustavljanje – prekida VI izvršenje



- Pauziraj / nastavi dugme – pauzira (odnosno nastavlja) VI izvršenje



- Font ring – služi za podešavanje opcija u fontu, uključujući tip fonta, veličinu i stil.



- Alignment ring – podešava opcije poravnjavanja, uključujući verikalno, gornje ivice, levo itd. , za dva ili više objekta



- Distribution ring – podešava opcije distribucije, uključujući praznine, kompresije itd. , za dva ili više objekta



- Dugme menja izgled da bi pokazalo da je VI pokrenut



- Kopisti Operatin tool da menjna vrednost gornje i donjne granice. Prvo, istaknuti staru vrednost, ili dvoklikom na vrednost koju želimo da menjamo, ili tako što ćemo kliknuti i prevući preko vrednosti sa Labeling tool-om.

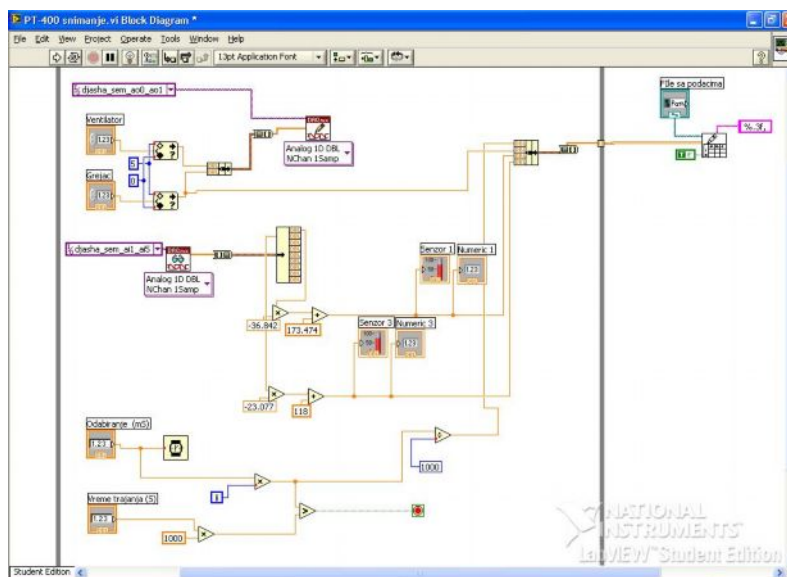


- Menja period ažuriranja kontorle slajdova, prikazano na sledećej ilustraciji, postavljanjem Operatin tool-a na klizač i prevlačenjem na novu lokaciju

Kontrole mogu imati mnoštvo oblika. Mnogi od tih oblika su virtuelne „slike“ realnih kontrola koje se koriste kod stvarnih instrumenata kao rotirajuće dugme na promer. Ostali imaju digitalni koncept. Sve kontrole imaju neku vrstu vizualne informacije na osnovu koje korisnik uvek zna u kojem se stanju nalaze. To jako pomaže jer se ne mora pri programiranju eksplicitno programirati i stanje pojedine kontrole. Drugo vrlo korisno svojstvo kontrola je to da se može odrediti njihova reakcija ukoliko je postavljena ulazna vrednost nestabilana. U specifičnom slučaju, ukoliko ulazna kontrola može poprimiti samo celobrojne vrednosti u opsegu od 0 do 10, može se specificirati što se treba dogoditi ukoliko se unese ulazna vrednost 3,5 ili -1 ili «nula» (zero) kao tekstualna naredba. Programski rešiti tu vrstu problema zahtijeva previše programerskog vremena tako da se ovom opcijom štedi na vremenu.

Indikatori imaju mnoštvo različitih oblika. Neki od njih su kopija stvarnih indikatora (instrumenata i sl.) a neki su dizajnirani prikladnije za pozadinu računara. Koncept indikatora uključuje grafove (graphs) i dijagrame (charts) što je druga važna ušteda u vremenu jer se ti pokazivački elementi ne moraju dizajnirati posebno.

Blok dijagram predstavlja pozadinu njegove prednje ploče koja prikazuje kako su sve kontrole i indikatori povezani međusobno kao i skrivene module gde se odvija sva programska obrada podataka. Izgleda na neki način poput elektroničkog šematskog dijagrama i barem je konceptualno ožičen na isti način. Takav prikaz često izgleda vrlo složen kao u elektroničkoj šemi stvarnog elektroničkog mernog instrumenta. Stoga je jedno od najvažnijih svojstava u LabVIEW programiranju odrediti tajming i redosled operacija. U konvencionalnom programskom jeziku to je ostvareno redosledom naredbi i korišćenjem različitih programskih petlji (FOR, WHILE itd.). LabVIEW radi na potpuno isti način, jedino što je način na koji se određuje redosled izvršenja naredbi prefinjeniji. Koncept korišćen u LabVIEW-u je *dataflow* (radnja se na nekom elementu izvršava kad su svi njegovi ulazi na raspolaganju). To znači paralelnost u izvršenju radnje (ili barem pseudoparalelizam). Uobičajeno izvršavanje radnji je s leva na desno jer su ulazi obično na levom delu elementa a izlazi na desnom delu. Međutim, to nije zahtev već samo stvar dogovora. Programske petlje i redosled kojim se odvijaju naredbe odvija se pomoću struktura koje izgledaju poput knjiga s brojevima stranica ili kao odsecci filmske trake.



Slika 2 - Blok šema

2.3.2. Paleta alata

LabVIEW koristi plutajuće palete alata koje možete da koristite za uređivanje i otklanjanje grešaka VI. Vi koristite <Tab> taster kroz najčešće korišćene alatke na paleti. Ako zatvorite paletu alata idite na **Windows»Show Tools Palette** da bi ste prikazali paletu. Donja slika prikazuje ilustraciju palete alata.



- Operativni alat



- Pozicioniranje alata – selektovanje objekata, resajzovanje itd.



- Alat za izmenu teksta

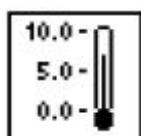
-  - Alat za povezivanje objekata provodnicima u blok dijagramu
-  - Alat za skrolovanje
-  - Alat za prelomnu tačku
-  - Sonda alat – postavlja sonde na provodnicima
-  - Alat za uzimanje boje
-  - Alat sa bojama

2.4. Primer za promenu opsega termometra

1. Otvaramo novi *Front panel* izabirom opcije **File » New**. Za Windows I Unix, ako smo zatvorili sve VI, selektujemo **New VI** iz LabVIEW dialog box-a.

Napomena: Ako paleta *Controls* nije vidljiva, opcijom **Windows » Show Controls Palette** je činimo vidljivom.

2. Odaberemo indicator temperature opcijom **Controls » Numeric**, i postavimo ga na front panel.



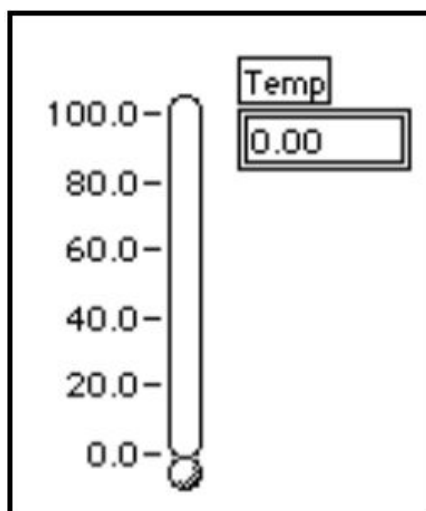
Indikator temperature

3. Kucamo **Temp** unutar datog tekst box-a i potvrđujemo sa **Enter** dugmetom na toolbar-u.

Napomena: Ako kliknemo van tekst box-a, label će nestati. On ponovo može biti prikazan selektovanjem **Show»Label**.

4. Promeniti opseg temperature na termometru od 0.0 do 100.0 .

(Po default-u je skaliran do 10.0, ali jednostavnim dvoklikom na njega i unošenjem vrednosti 100.0 umesto date se vrši ova promena opsega).



Slika 3 - Izgled termometra posle promene opsega

2.5. Paleta kontrola

Paleta kontrola se sastoji iz graficke plivajuće palete koja se automamtski otvara kada startujeom LAbVIEW. Koristimom ovu paletu da postavimo kontrole i indikatore na front panel. Svaka ikonica „najvišeg nivoa“ sadrži podpalete. Ako paleta kontrola nije vidljiva, možemo je otvoriti opcijom **Windows » Show Controls Palette** iz Front panel menia. Takođe se može pojaviti na slobodnomm prostoru u front panel-u da pristupi privremenoj kopiji palete kontrola.

Sledeća ilustracija prikazuje “top level” palete kontrola :

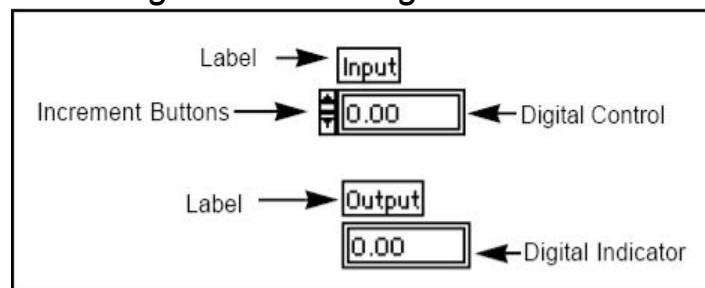


Slika 4 - Paleta kontrola

3. KONTROLE I INDIKATORI

3.1. Numeričke kontrole i indikatori

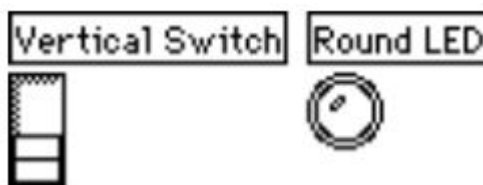
Numeričke kontrole koristimo za unos brojčane veličine, dok numerički indikatori ispisuju brojčanu vrednost. Dva najčešće korišćena numerička objekta su *digital control* i *digital indicator*.



Slika 5 - Digitalna kontrola i digitalni indikator

3.2. Bulova kontrola i indikatori

Bulovu kontrolu i indikatore koristimo da unos i ispis Bulove (*true/false*) vrednosti. Ovi objekti simuliraju prekidače, dugmiće i LED diode. Dva najčešće korišćena Bulova objekta su *vertikalni prekidač* i *kružna LED dioda*.

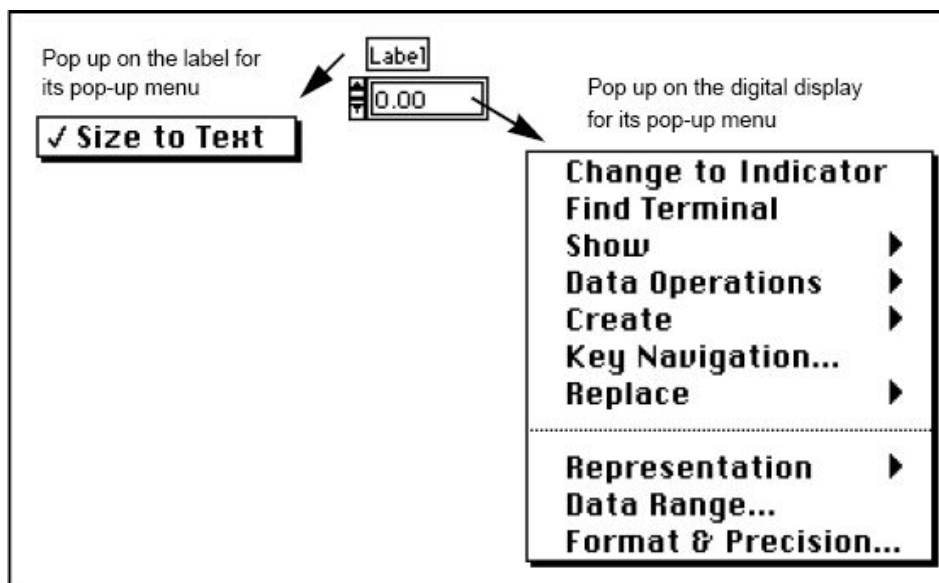


Slika 6 - Vertikalni prekidač (levo) i kružna LED dioda (desno)

3.3. Konfigurisanje kontrola i indikatora

Možemo podesiti skoro sve kontrole i indikatore pomoću opcija iz njihovih pop-up menia. Najjednostavniji način za pristup pop-up meniju je klikom na objekat pop-up meni alatku () na mjestu koji objekat koji ima pop-up meni.

Sledeća slika ilustruje dati metod za digitalnu kontrolu :

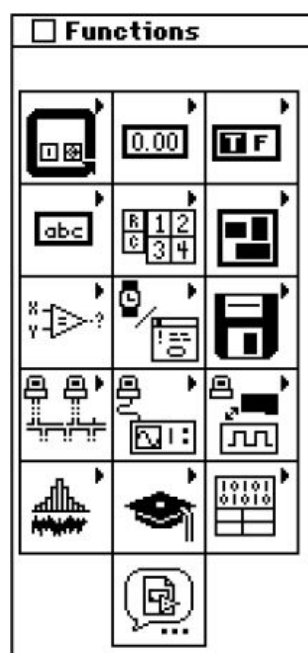


Slika 7 - Pop-up meni

3.4. Paleta funkcija

Paleta funkcija sadrži grafičke plivajuće palete koje se automatski otvaraju kada uključimo blok dijagram. Koristimo ovu paletu da postavimo čvorove na blok dijagram VI-a. Svaka "top level" ikonica sadrži podpalete. Ako paleta funkcija nije vidljiva, opcijom **Windows » Show Functions Palette** iz blok dijagrama je aktiviramo.

Sledeća ilustracija prikazuje "top level" palete funkcija :



Slika 8 - Paleta funkcija

4. BLOK DIJAGRAM

1. Otvaramo blok dijagram opcijom **Windows » Show Diagram**. Potom selektujemo objekte blok dijagrama iz palete funkcija. Za svaki objekat koji želimo da ubacimo, selektujemo ikonicu i onda objekat iz "top level" palete ili odgovarajuće podpalete. Kada pozicioniramo kursor miša na blok dijagram, LabVIEW prikazuje pregled objekta. Ovo je prilika da stavimo objekat na mesto koje želimo na blok dijagramu.

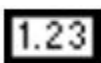
Napomena: Ako paleta funkcija nije vidljiva, opcijom **Windows » Show Functions Palette** je uključujemo.



- Opcija simulira čitanje napona iz plug-in podataka sa ploče



- Funkcija množenja (**Function » Numeric**)



- Numerička konstanta (**Function » Numeric**)



- String konstanta (**Function » String**)

4.1. Tehnike povezivanja (Wiring techniques)

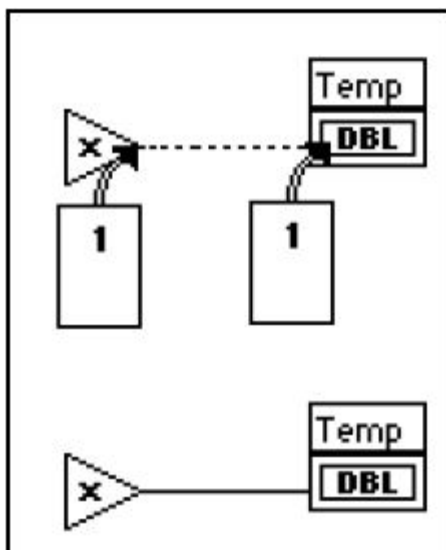
U ilustraciji povezivanja u ovom odeljku, strelice na kraju kursora pokazuju gde da kliknemo, a broj kod strelice koliko puta moramo da kliknemo.

"Hot spot" alata je vrh odmotan na segmentu (Hot Spot ).

Da bi povezali provodnikom jedan terminal sa drugim, aktiviramo alat za povezivanje na prvom terminal, prevučemo ga do drugog terminal, i potom kliknemo na drugi terminal. Nije bitno sa kog terminal krećemo.

Kada je alat za povezivanje iznad terminal, oblast oko terminal trepće, i ukazuje da klikom povežete žicu za dati terminal. Ne držite dugme miša svo vreme povezivanja dva terminal. Žicu možemo savijati

pomeranjem miša normalno na dati pravac. Za menjanje pravca žice koristimo *Space* na tastaturi.

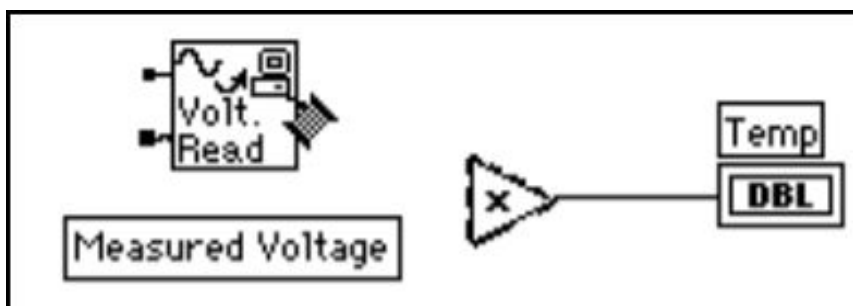


Slika 9 - Povezivanje dva terminala

4.2. “Trake saveta”

Kada prelazimo preko terminala alatom za povezivanje, pojavljuje se tzv. “traka saveta”. Ona se sastoji od malog banera, teksta žute boje, koji ispisuje ime terminala. One nam pomažu da povežemo terminale.

Sledeća ilustracija traku saveta (Measured voltage) koja se pojavljuje kada se alat za povezivanje nalazi iznad izlaza Demo Voltage čitača VI-a.



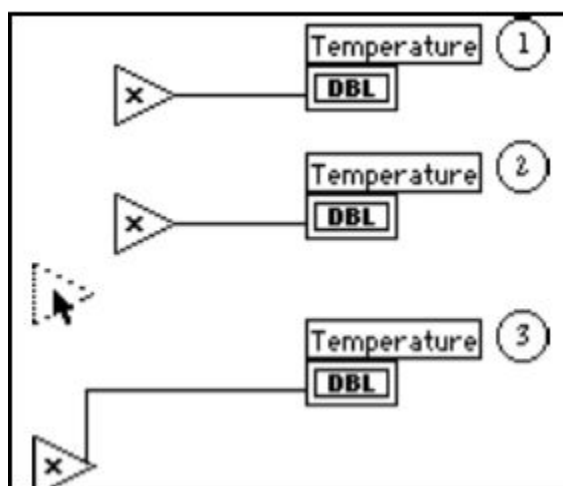
Slika 10 - Primer “trake saveta”

4.3. Istezanje provodnika

Objekte povezane provodnicima možemo premeštati pojedinačno ili grupno prevlačenjem datih objekata na novu lokaciju pomiču alata za pozicioniranje.



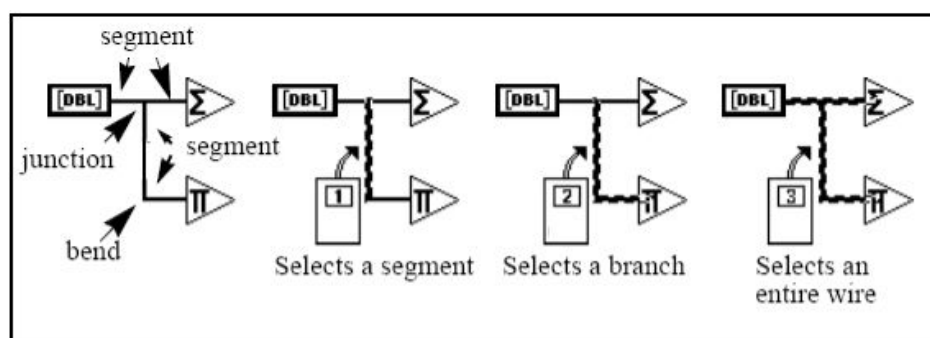
- Alat za pozicioniranje



Slika 11 - Primer premeštanja terminala

4.4. Obeležavanje i brisanje provodnika

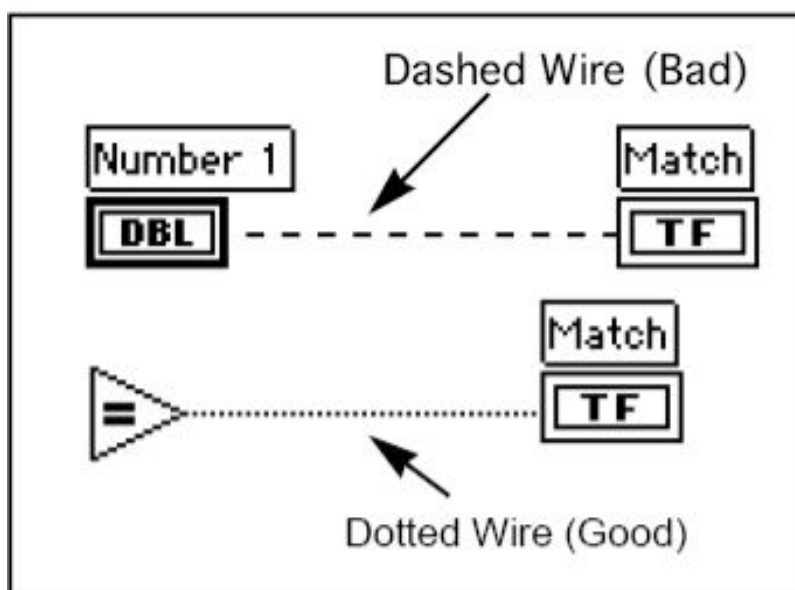
Slučajno može doći do pogrešnog povezivanja čvorova. Ukoliko se to dogodi, obeležimo provodnik koji želimo da izbrišemo i pritisnemo *Delete* na tastaturi. Segment provodnika je horizontalan ili vertikalan deo provodnika. Mesto gde se tri ili četiri provodnika spajaju zove se čvor. Provodnik sadrži delove provodnika od čvora do čvora, od jednog do drugog terminala. Segment provodnika se selektuje jednostavnim klikom na njega.



Slika 12 - Selektovanje dela, grane i celog provodnika

4.5. "Loši provodnici"

Isprekidan provodnik predstavlja loš provodnik. Ovo se može desiti iz mnogo razloga, kao npr. povezivanje dve kontrole ili povezivanje dva terminala gde se tipovi podataka ne podudaraju (npr. numeričke i tačno/netačno). Ove provodnike možemo jednostavno izbrisati klikom na njih i protiskom na *Delete*. Opcijom **Edit » Remove Bad Wires** brišemo sve "loše provodnike".



Slika 13 - "Dobar" i "loš" provodnik

5. STARTOVANJE VI-a

1. Za Windows i Macintosh, front panel se aktivira klikom bilo gde na njemu. Za Unix se pak aktivira opcijom **Windows » ShowPanel**.

2. Startujemo VI klikom na dugme za startovanje ().

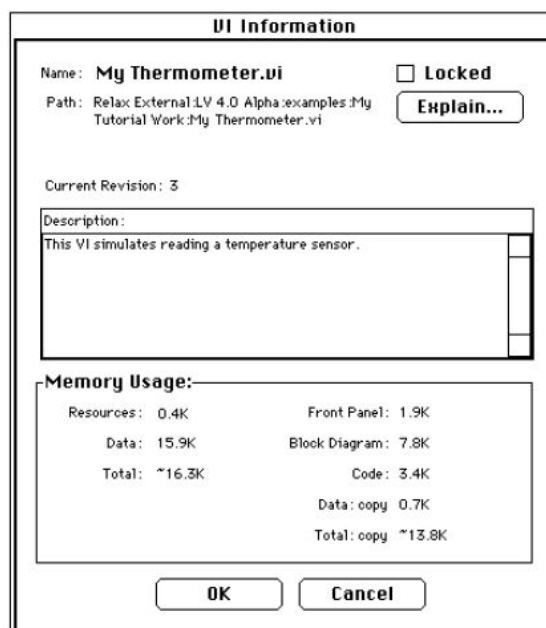
Primetićemo da moramo da svaki put moramo da isključimo VI pre ponovnog startovanja. Ukoliko želimo da bez prekida startujemo VI, startujemo dugme za neprekidno pokretanje.

3. Startujemo dugme za neprekidno pokretanje (). Ako želimo da prekinemo neprekidno startovanje, jednostavnim klikom na ovo dugme to omogućavamo.

Napomena: Dugme za neprekidno startovanje nije omiljeni metod za ponavljajući kod blok dijagrama, već bi trebalo da se koristi *Looping* struktura.

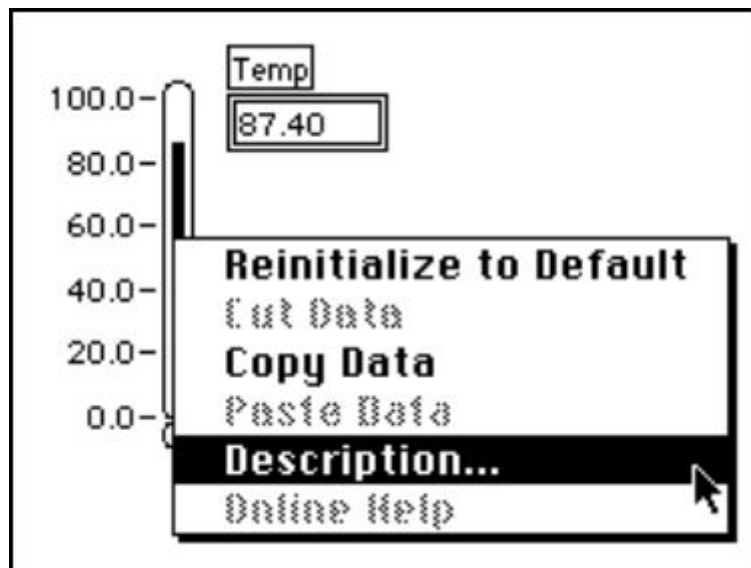
5.1. Dokumentovanje VI-a

1. VI možemo dokumentovati opcijom **Windows » Show VI Info**. Opis VI-a kucamo u VI dijalog informacija. Klikom na OK potvrđujemo.



Slika 14 - VI information

Opis objekta možemo videti desnim klikom na objekat i odabirom opcije **Description** (u *blok dijagram-u*) i **Data Operations » Description** (u *front panel-u*).



Slika 15 - Opis objekta

Napomena: Opis se ne može menjati dok je VI radi.

6. KRATAK PREGLED

Virtuelni instrumenti (VI) imaju tri glavna dela:

- a. *Front panel*
- b. *Blok dijagram*
- c. *Icon/connector*

- *Front panel* specificira ulaze i izlaze VI-a.
- *Blok dijagram* sadrži izvršni kod koji kreiramo koristeći čvorove, terminale i provodnike.
- Uz pomoć *icin/connector* možemo koristiti VI i pod VI u blok dijagramima drugih VI-a.

Paleta alati sadrži grafičke lebdeće palete. U front panel- u ovu paletu možemo koristiti za kreiranje, editovanje i debugiranje VI-a.

Alat za pozicioniranje koristimo za selektovanje, pozicioniranje i “resize-ovanje” objekata.

Alat za označavanje koristimo za kreiranje “nalepnica” i upisivanje teksta u njih.

Alat za povezivanje koristimo da povežemo međusobno željene objekte u blok dijagramu.

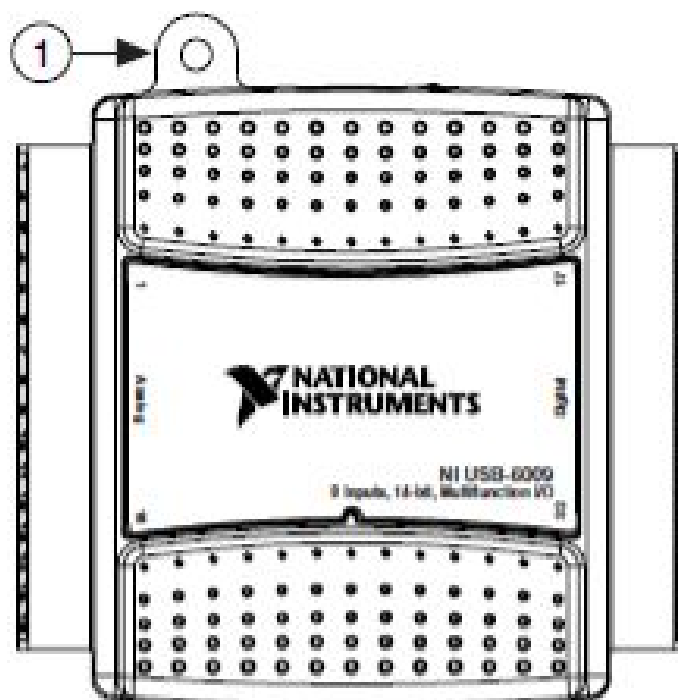
Alat za bojenje koristimo za promenu boje prednje strane i pozadine prozora, kontrola, indikatora itd.

Front panel i blok dijagram sadrži paletu sa alatima kojima se nalazi dugme za startovanje zajedno sa ostalim dugmićima i kontrolama za izvršavanje VI-a. U front panel-u postavljamo kontrole i indikatore da označimo ulaze i izlaze VI-a. Koristimo ***paletu kontrola*** da dodamo kontrole i indikatore. Na blok dijagramu razvijamo dijagram izvora povezivanjem čvorova i terminala ***alatom za povezivanje***. Koristimo ***paletu funkcija*** da postavimo čvorove na blok dijagram.

Možemo modifikovati skoro sve LabVIEW objekte kroz njihove pop-up menije, kojima pristupamo *Object pop-up menu* alatom.

7. NI6008 KARTICA

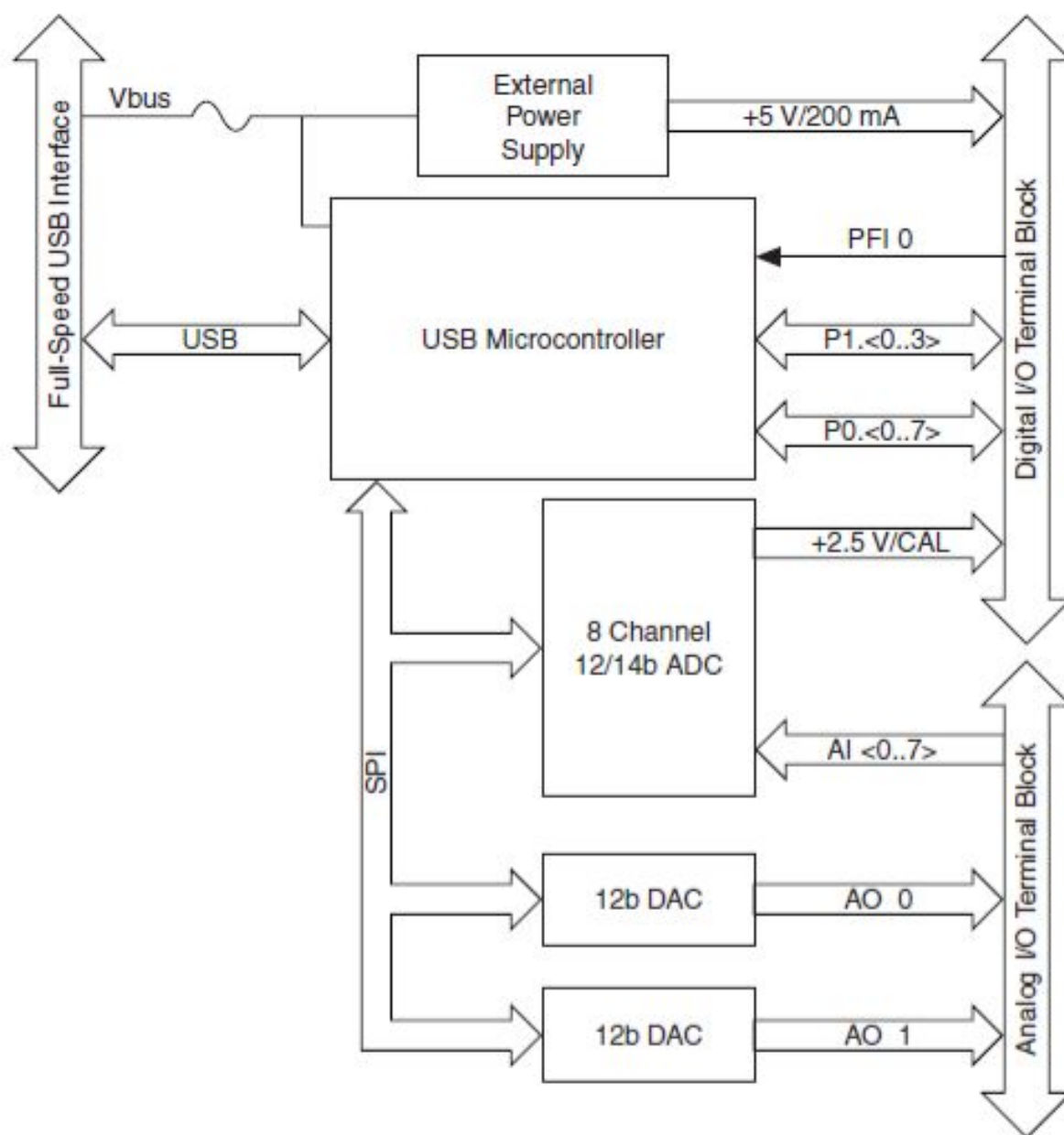
National Instruments 6008 obezbeđuje osnovne funkcije za prikupljanje podataka za aplikacije kao što su jednostavne evidencije podataka, prenosiva merenja i akademske labaratorijske eksperimente. Ona je pristupačna i za studentsku upotrebu i dovoljno jaka za sofisticiranije merne aplikacije. NI USB-6008 kartica obezbeđuje vezu sa osam analognih ulaza (AI), dva analogna izlaza (AO), 12 digitalnih izlazno/ulaznih kanala i 32-bitni brojač sa punom brzinom USB interfejsa.



Slika 16 - Izgled kartice NI6008

Za kurseve sa praktičnim eksperimentima National Instruments je razvio USB-6008 komplet za učenike koji pored kartice sadrži kopiju LabVIEW studentsko izdanje.

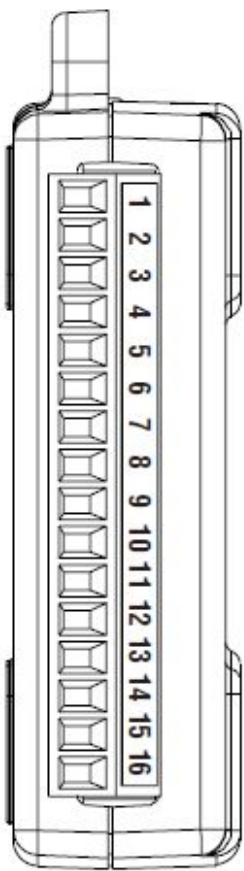
Sledeći blok dijagram će pokazati ključne funkcionalne komponente kartice NI6008.



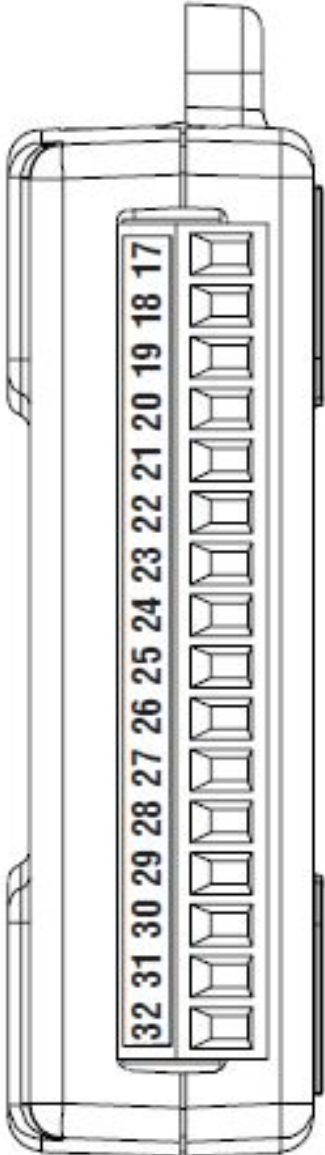
Slika 17 - **Blok dijagram uređaja**

7.1. Ulazno/izlazni konektor

7.1.1. Analogni konektor

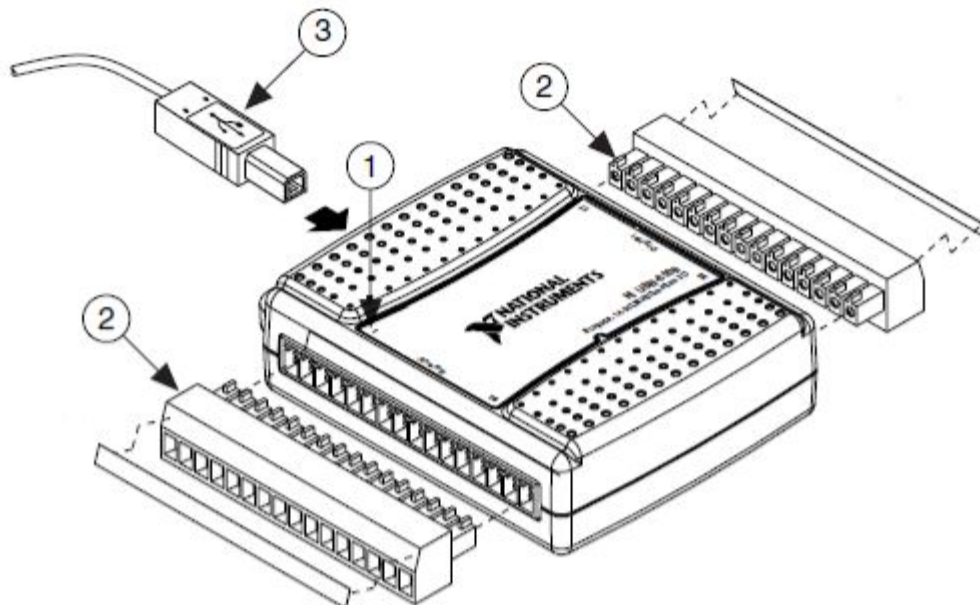
Module	Terminal	Signal, Single-Ended Mode	Signal, Differential Mode
	1	GND	GND
	2	AI 0	AI 0+
	3	AI 4	AI 0–
	4	GND	GND
	5	AI 1	AI 1+
	6	AI 5	AI 1–
	7	GND	GND
	8	AI 2	AI 2+
	9	AI 6	AI 2–
	10	GND	GND
	11	AI 3	AI 3+
	12	AI 7	AI 3–
	13	GND	GND
	14	AO 0	AO 0
	15	AO 1	AO 1
	16	GND	GND

7.1.2. Digitalni konektor

Module	Terminal	Signal
	17	P0.0
	18	P0.1
	19	P0.2
	20	P0.3
	21	P0.4
	22	P0.5
	23	P0.6
	24	P0.7
	25	P1.0
	26	P1.1
	27	P1.2
	28	P1.3
	29	PFI 0
	30	+2.5 V
	31	+5 V
	32	GND

7.2. Povezivanje kartice sa PC-jem i objektom upravljanja

Na sledećoj slici možemo videti karticu sa konektorima. Sa konektorom 3 (USB kablom) ostvarujemo konekciju između računara i kartice. Konektorima, označenim brojem 2 na slici, ostvarujemo analogno/digitalnu konekciju sa objektom upravljanja.



Slika 18 - Šema kartice sa konektorima za povezivanje



Slika 19 - Izgled kartice NI 6008 sa konektorima za povezivanje

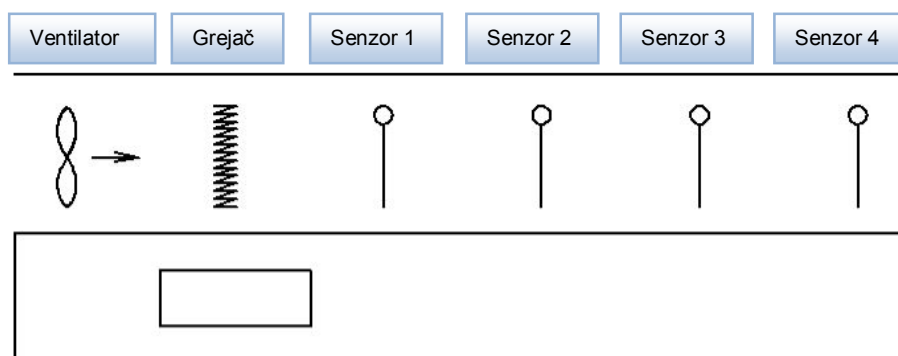
8. OPIS SISTEMA

Laboratorijski setup na kome radim završni rad, sastoji se od mašine PT400, kartice NI6008 koja omogućava vezu mašine sa računarom, i računara na kome se nalazi korisnički interfejs uz pomoć kojeg upravljamo ovim procesom. Interfejs koji koristimo za ovaj setup je LabVIEW i MATLAB. Uz pomoć LabVIEW softvera radimo kompletno upravljanje procesom, a u MATLAB-u vršimo identifikaciju procesa, uz pomoć koje ćemo podešavati PID kontroler.



Slika 20 – **Model sistema**

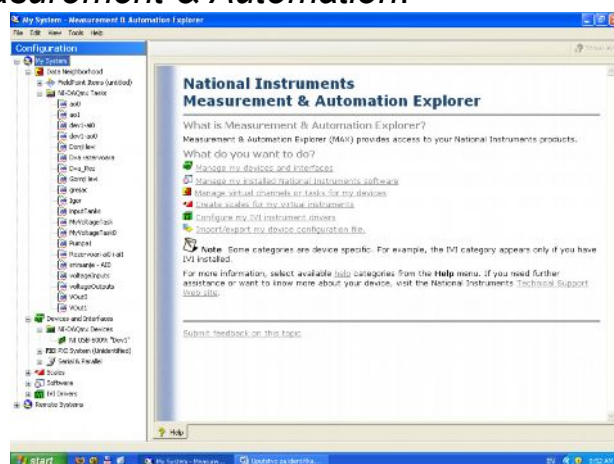
Mašina PT400 je laboratorijski model sušare. Ona se sastoji od komore u kojoj se nalaze ventilator, grejač i senzori na različitim lokacijama. Zbog temperaturnog procesa na različitim lokacijama različita je i temperatura. Koriste se različite vrste senzora. Postoje senzori sa bržim odzivom ali generalno manjom robusnošću. Kao i senzori sa sporijim odzivom i većom robusnošću. Pošto je moja tema upravljanje temperaturom na prvom i poslednjem senzoru, na prvo i poslednje mesto u komori sam postavio brze senzore.



Slika 21 – **Model mašine PT400**

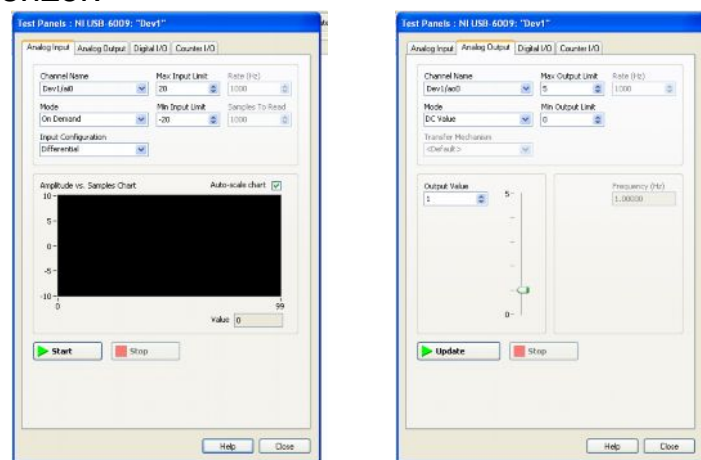
9. KREIRANJE TASKOVA

Otvoriti *Measurement & Automation*.



Slika 22 - *Measurement & Automation*

U prozoru sa strane kliknuti na *My System/Data Neighborhood/Devices and Interfaces/NI-DAQmx Devices*. U opcijama u gornjem delu ekrana kliknuti na *Test Panels*. Tada se otvara prozor kao na sledećim slikama. U kartici *Analog Input* se testira na kojim kanalima se nalazi koji senzor.



Slika 23 - *Test Panels*

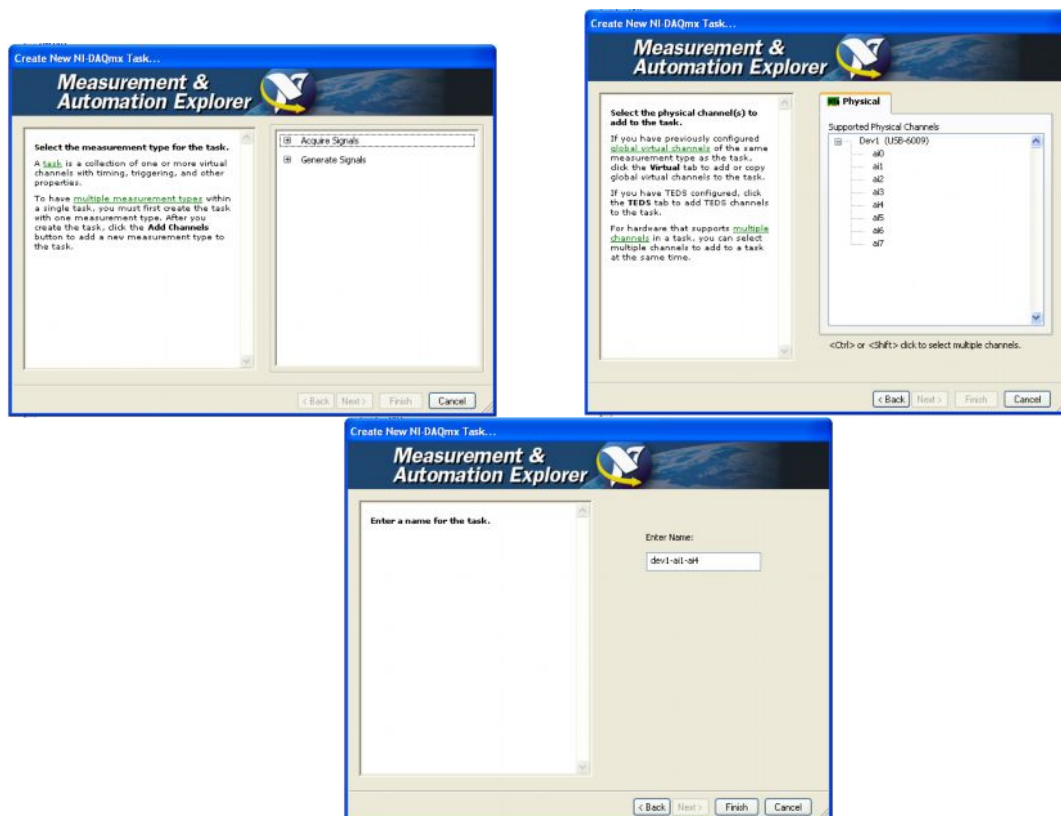
Prvo treba podesiti limit na *Limit* na +10 i -10, a *Input configuration* na *RSE*. U polju *Chanel Name* nalazi se lista od 8 mogućih ulaza. Korisnik sada treba da obeleži prvi ulaz i klikne na *Start*. Ovim će pokrenuti prijem podataka. Sada treba da testira koji se to senzor nalazi na odabranom kanalu. Kada se nađe koji se senzor nalazi na datom kanalu, to se zabeleži, pređe na sledeće kanal, i ceo proces se ponavlja dok se ne pronađu svi kanali.

U tabu *Analog Output* se nalaze izlazi sa kartice. Sada treba pronaći na kom izlaznom kanalu se nalaze koji uređaji. Postupak je

skoro isti. Odabere se kanal i klikne na *Update*. Tada će se vrednost napona koja je odabrana u polju *Output Value* poslati na dati izlaz. Tada treba pronaći koji je izlaz pokrenut i to pribeležiti. Postupak ponoviti i sa drugim kanalom. Na kraju testiranja obavezno vratiti 0 na oba kanala.

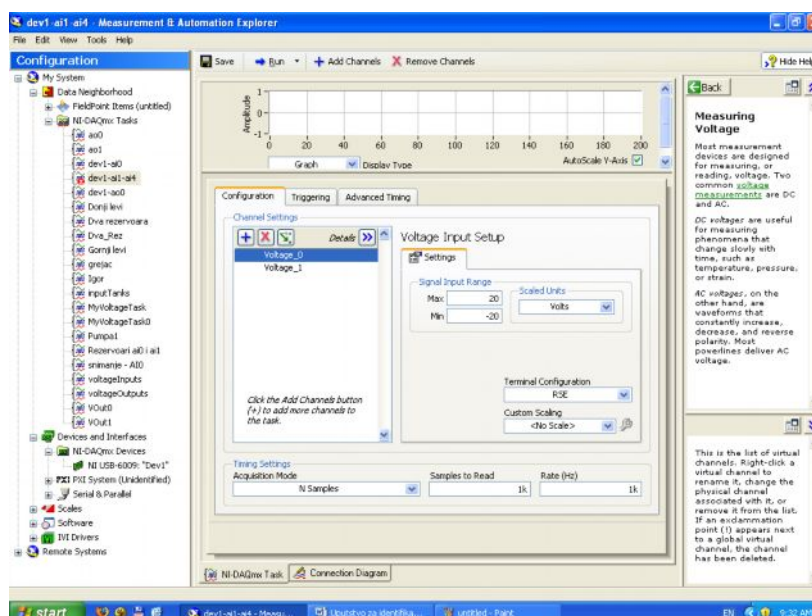
Kada je poznato na kojem kanalu se nalazi koji uređaj treba napraviti grupe kanala preko kojih će se signali primati ili slati – *Tasks*. To se nalazi u *My System/Data Neighborhood/NI-DAQmx Tasks*.

U gornjem delu ekrana nalazi se dugme *Create New NI-DAQmx Task*. Klikom na njega otvara se niz prozra kao na sledećem slikama. Ulazni signali se grupišu odvojeno od izlaznih signala. Klikom na *Acquire Signals/Analog Input/Voltage* dolazi se do sledećeg ekrana na kojem treba izabrati koji će se kanali koristiti. Ovde treba izabrati samo kanale koji će se koristiti. Kada se odaberu kanali ide se dalje. Na sledećem kanalu se da je naziv tasku. Klikom na *Finish* zatvaraju se ovi prozori.



Slika 24 - *Create New NI-DAQmx Task*

Sada treba još samo dodatno podesiti napravljeni task. U polju *Signal Input Range* treba namestiti maksimum i minimum 10 i -10 volti, *Terminal Configuration* treba da bude *RSE*, a *Acquisition Mode* treba da bude *1 Sample (On Demand)*. Klikom na *Save* podešavanja su snimljena i može se preće na podešavanje izlaza.



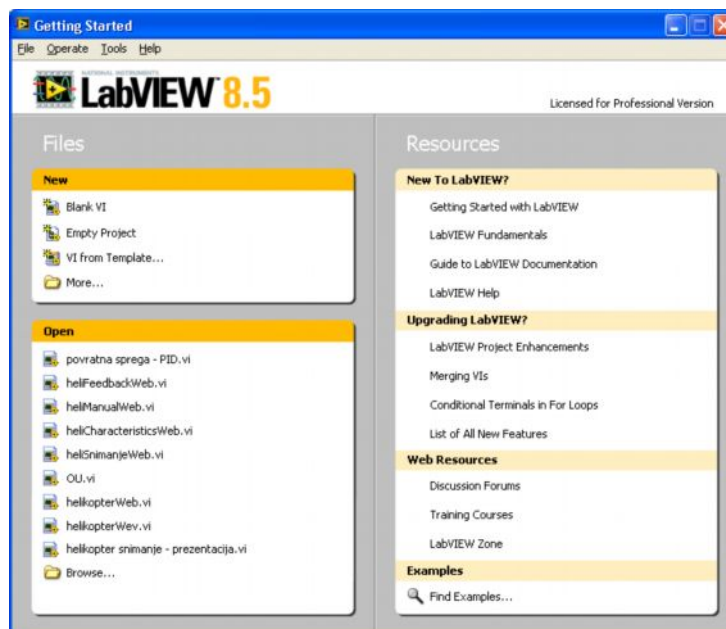
Slika 25 - Task

Podešavanja za izlaze se rade na gotovo isti način kao za ulaze. Jedino se *Generate Signals/Analog Output/Voltage*. Dalje je princip isti. U polju *Signal Output Range* treba namestiti maksimum i minimum na 5 i 0 volti, *Terminal* treba da bude *RSE*, a *Generation Mode* treba da bude *1 Sample (On Demand)*.

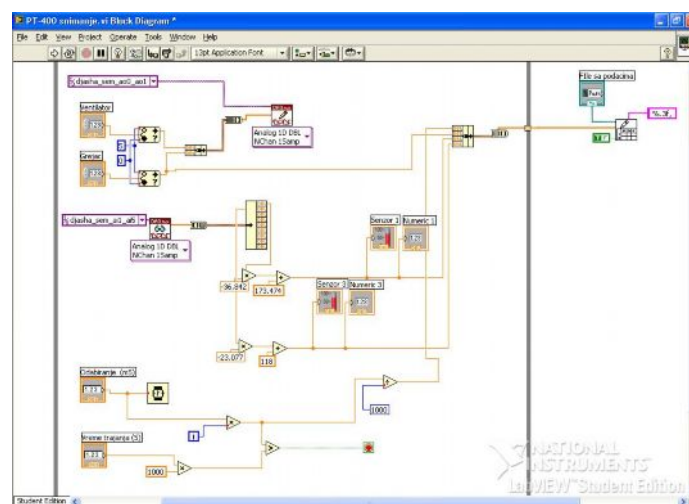
Time je proces kreiranja taskova gotov.

10. SNIMANJE PROCESA

Snimanje procesa vrši se u LabVIEW softveru.



Slika 26 - Izgled LabVIEW-a



Slika 27 - Blok šema

Prvi korak u snimanju procesa je kalibracija senzora. Sa senzora se dobija signal u voltima. Ta informacija nije u mnogome korisna i zato se pretvara u fizicki parametar koji je nama potreban (u ovom slučaju temperatura na prvom i poslednjem senzoru). To se radi tako što se pravi zavisnost između napone na sensorima i temperature očitane na

tom mestu pomoću termometra koji se uzima kao reper. Ako pretpostavimo da je zavisnost napona i temperature linearna onda dobijamo jednačinu kalibracije koja glasi:

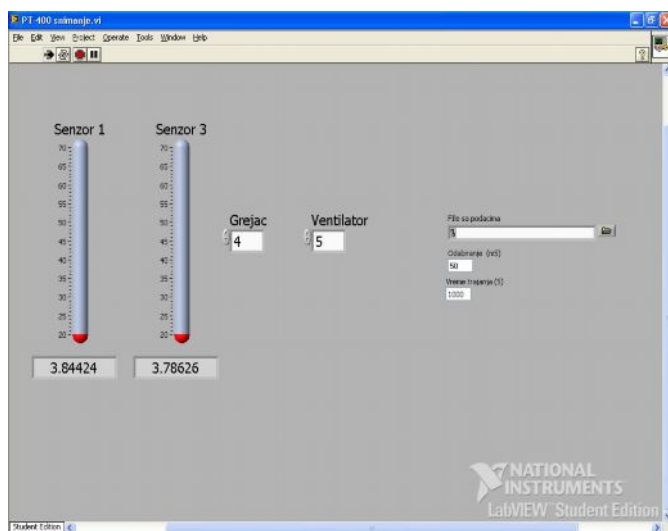
Za prvi senzor:

$$\text{Temperatura} = -36.842 * \text{Napon} + 173.474$$

Za poslednji senzor:

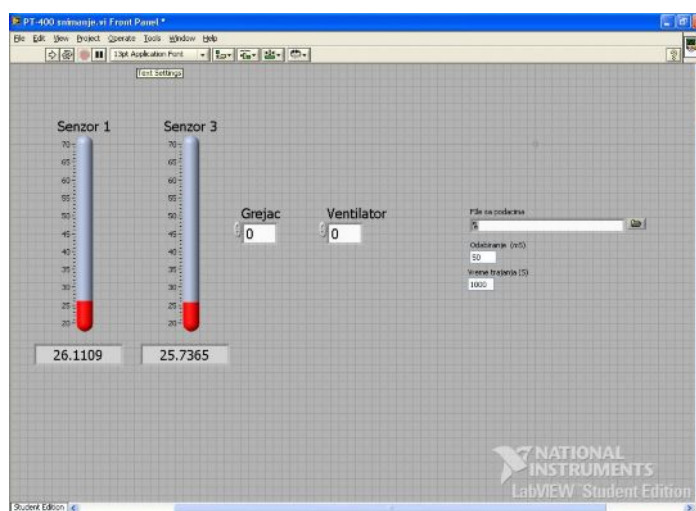
$$\text{Temperatura} = -23.072 * \text{Napon} + 118.000$$

Na kraju je dovoljno da ove vrednosti ubacimo u LabVIEW, i program će pokazivati stvarnu temperaturu na senzorima.

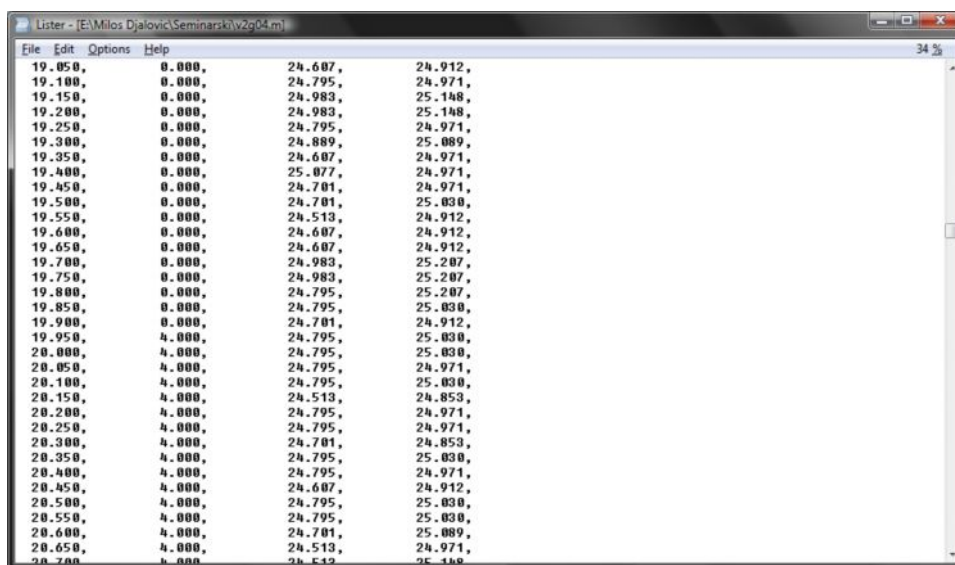


Slika 28 - Izgled Front Panela pre kalibracije

Slika 29 - Izgled Front Panela posle kalibracije



Potom je vršeno snimanje odziva u LabVIEW-u. Snimanje procesa se vrši korišćenjem fajla *PT-400 snimanje.vi*. Na slici 27 dat je blok dijagram tog programa. Posle podešavanja periode odabiranja i vremena trajanja procesa startuje se program. Snimanje odziva vrši se pri konstantnom naponu na ventilatoru od 2 V i naponu na grejaču koji je prvih 20 sekundi snimanja bio 0 V a preostalo vreme snimanja promenjen na 4V. Posle isteka datog vremena rezultati su snimljeni u fajl *v2g04.m*, koji je nazvan tako da bi bilo očigledno koji su parametri korišćeni prilikom snimanja procesa.



Time (s)	Voltage (V)	Temperature 1 (°C)	Temperature 2 (°C)
19.050,	0.000,	24.607,	24.912,
19.100,	0.000,	24.795,	24.971,
19.150,	0.000,	24.983,	25.148,
19.200,	0.000,	24.983,	25.148,
19.250,	0.000,	24.795,	24.971,
19.300,	0.000,	24.889,	25.089,
19.350,	0.000,	24.607,	24.971,
19.400,	0.000,	25.077,	24.971,
19.450,	0.000,	24.701,	24.971,
19.500,	0.000,	24.701,	25.030,
19.550,	0.000,	24.513,	24.912,
19.600,	0.000,	24.607,	24.912,
19.650,	0.000,	24.607,	24.912,
19.700,	0.000,	24.983,	25.207,
19.750,	0.000,	24.983,	25.207,
19.800,	0.000,	24.795,	25.207,
19.850,	0.000,	24.795,	25.030,
19.900,	0.000,	24.701,	24.912,
19.950,	4.000,	24.795,	25.030,
20.000,	4.000,	24.795,	25.030,
20.050,	4.000,	24.795,	24.971,
20.100,	4.000,	24.795,	25.030,
20.150,	4.000,	24.513,	24.853,
20.200,	4.000,	24.795,	24.971,
20.250,	4.000,	24.795,	24.971,
20.300,	4.000,	24.701,	24.853,
20.350,	4.000,	24.795,	25.030,
20.400,	4.000,	24.795,	24.971,
20.450,	4.000,	24.607,	24.912,
20.500,	4.000,	24.795,	25.030,
20.550,	4.000,	24.795,	25.030,
20.600,	4.000,	24.701,	25.089,
20.650,	4.000,	24.513,	24.971,
20.700,	4.000,	24.513,	25.148,

Slika 30 - Rezultati snimanja

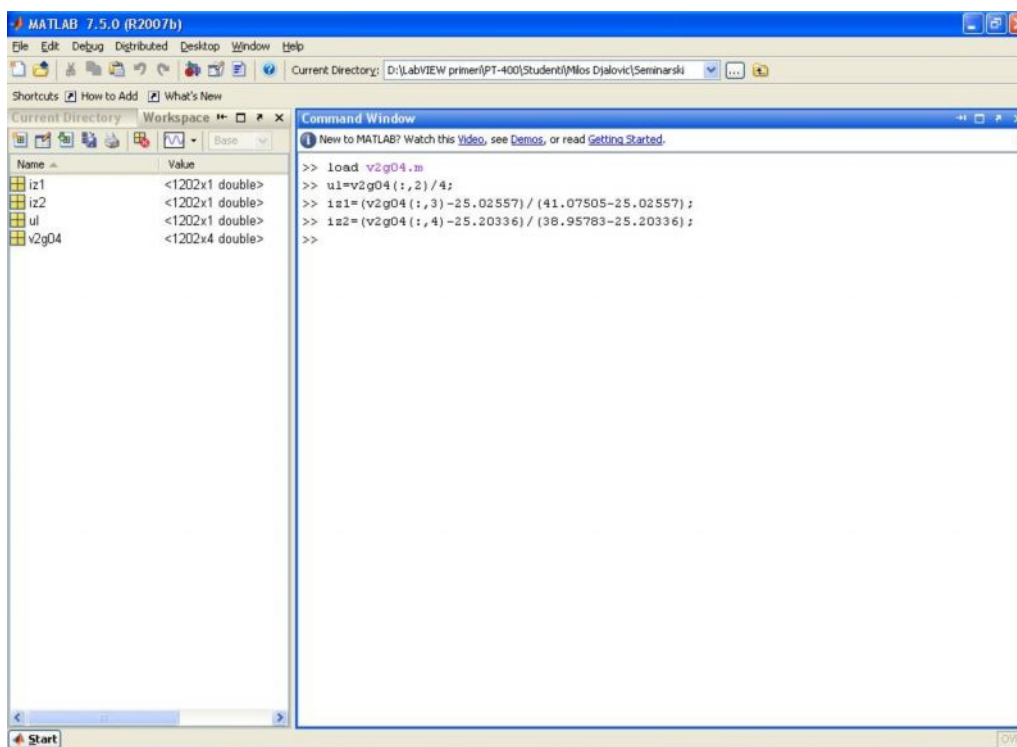
U prvoj koloni snimljenog fajla nalazi se vreme snimanja, u drugoj koloni napon na grejaču, u trećoj koloni temperatura na prvom senzoru i četvrtoj temperatura na poslednjem senzoru.

11. IDENTIFIKACIJA PROCESA

Identifikacija podrazumeva eksperimentalne prilaze modeliranja procesa, objekata upravljanja ili dinamičkih sistema u opštem slučaju. Ona danas predstavlja posebno razvijenu naučnu i stručnu oblast koja se bavi gradnjom matematičkih modela sistema polazeći od eksperimentalnih podataka merenja i opservacija.

Po završetku snimanja procesa u LabVIEW-u prelazimo na sledeći korak, identifikaciju procesa. Potrebno je da se snimljeni podaci prvo ubace u MATLAB. Izgled MATLAB okruženja sa ubačenim podacima prikazan je na slici ispod.

MATLAB je numeričko računarsko okruženje i četvrta generacija programskog jezika. Omogućava jednostavne matrične manipulacije, štampanje funkcija i podataka, implementaciju algoritama i kreiranje korisničkog interfejsa.



Slika 31 - MATLAB

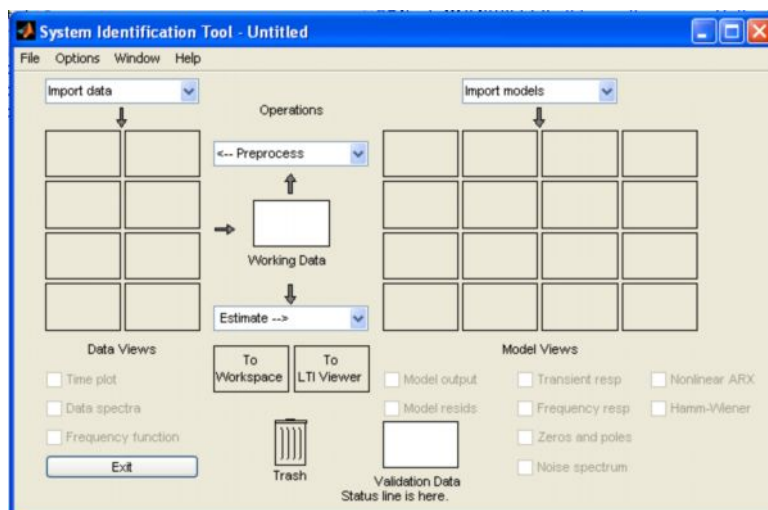
U glavnom delu ekrana prvo treba ukucati naredbe za ubacivanje podataka, a onda te podatke treba razvrstati po promenljivima. Pošto *ident toolbox* radi najbolje sa vrednostima koje su normalizovane, mi sve ulaze i izlaze pretvaramo u funkcije koje su ograničene između nule i jedinice. To se radi na sledeći način:

```
>> load V2G04.m;  
>> ul = v2g04(:,2) / 4;  
>> iz1 = (v2g04(:,3) - 25,02557) / (41,07505 - 25,02557);  
>> iz2 = (v2g04(:,4) - 25,20336) / (38,95783 - 25,20336);  
>> ident
```

Procedura koja se vidi naziva se normalizacija i ona je obavezna pre ubacivanja podataka u *ident toolbox*.

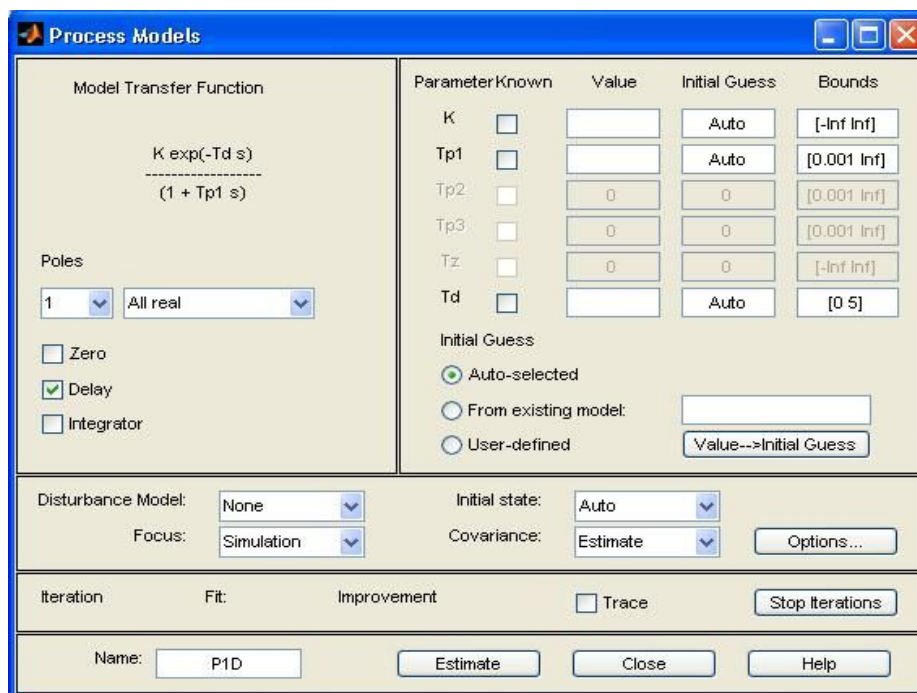
11.1. IDENTIFIKACIJA PRVOG SENZORA

Sledeći korak je identifikacija. Ona se radi u modulu matlaba koji se zove *ident toolbox*.



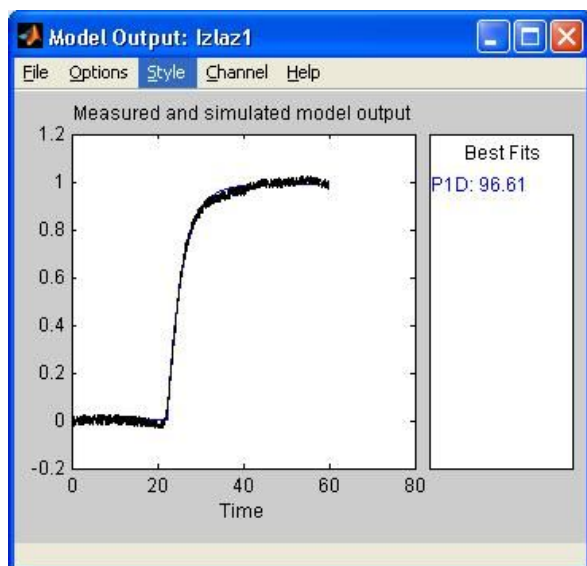
Slika 32 - *Ident Toolbox*

Ovde se vrši nekoliko pretpostavki modela (u našem slučaju za jedan i dva pola) odabiranjem datih u opcijama i klikom na *estimate*.

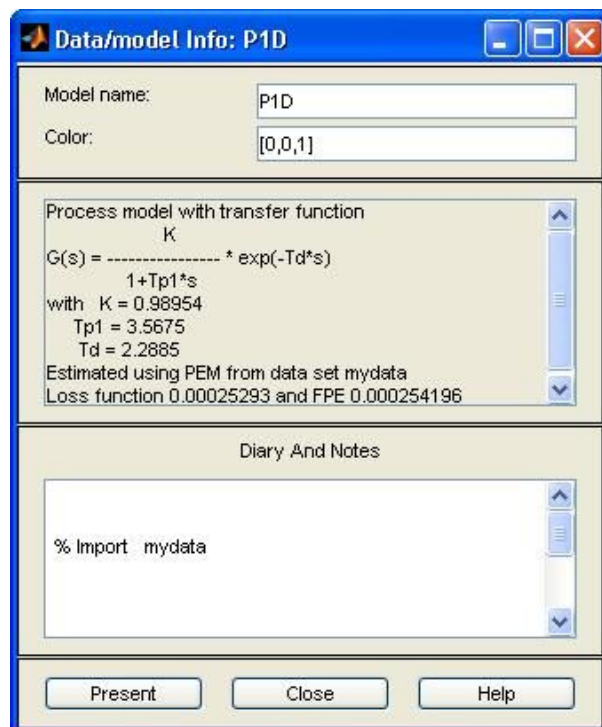


Slika 33 - *Modeli funkcije prenosa*

Rezultati nam ukazuju da modeli prvog i drugog reda imaju gotovo istu preciznost (96.61 %), i iz tog razloga biramo model prvog reda i koristimo njegove parametre.



Slika 34 - Pretpostavljeni i realni modeli



Slika 35 - Parametri modela

Funkcija prenosa :

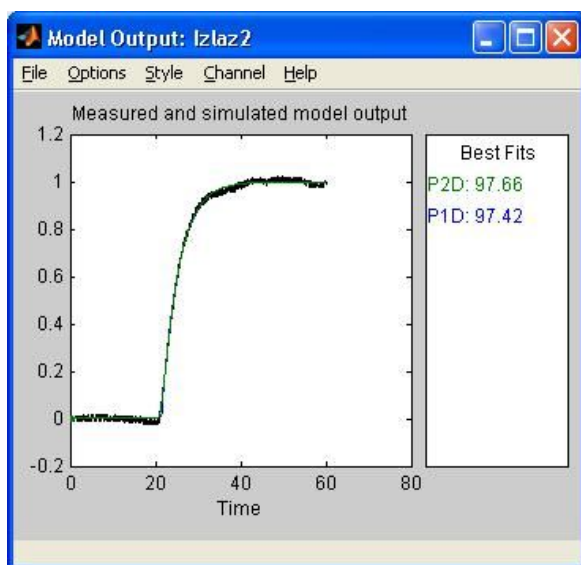
$$G(s) = \frac{K}{1 + T_{p1} \cdot s} \cdot e^{(-T_d \cdot s)}$$

Parametri modela :

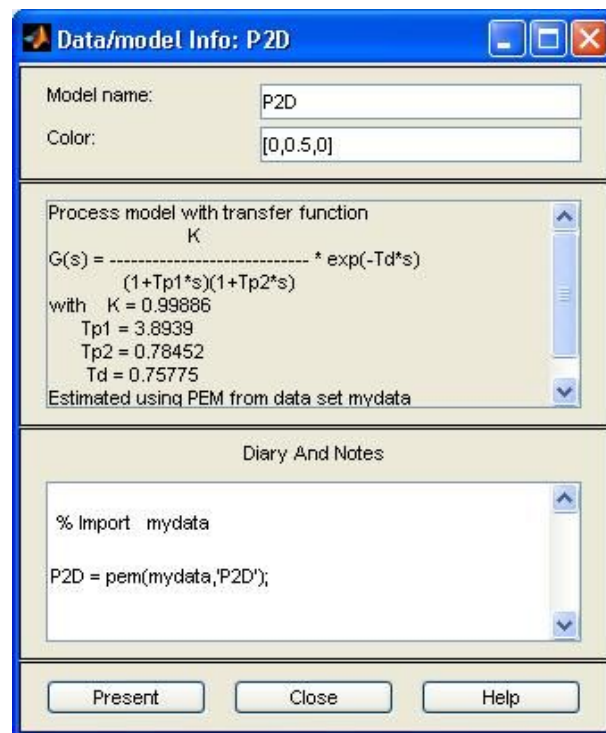
- K = 0.98954
- Tp1 = 3.5675
- Td = 2.2885

11.2. IDENTIFIKACIJA POSLEDNJEG SENZORA

Proces identifikacije poslednjeg senzora je sličan procesu identifikacije prvog senzora. Rezultati identifikacije nam ukazuju da je preciznost modela prvog reda (97,42 %), a preciznost modela drugog reda (97,66 %). Iako je ova razlika vrlo mala, skoro zanemarljiva, odlučujemo da izaberemo model drugog reda, tek da bi pokazali drugačije metode podešavanja PID kontrolera.



Slika 36 - *Pretpostavljeni i realni modeli*



Slika 37 - *Parametri modela*

Funkcija prenosa :

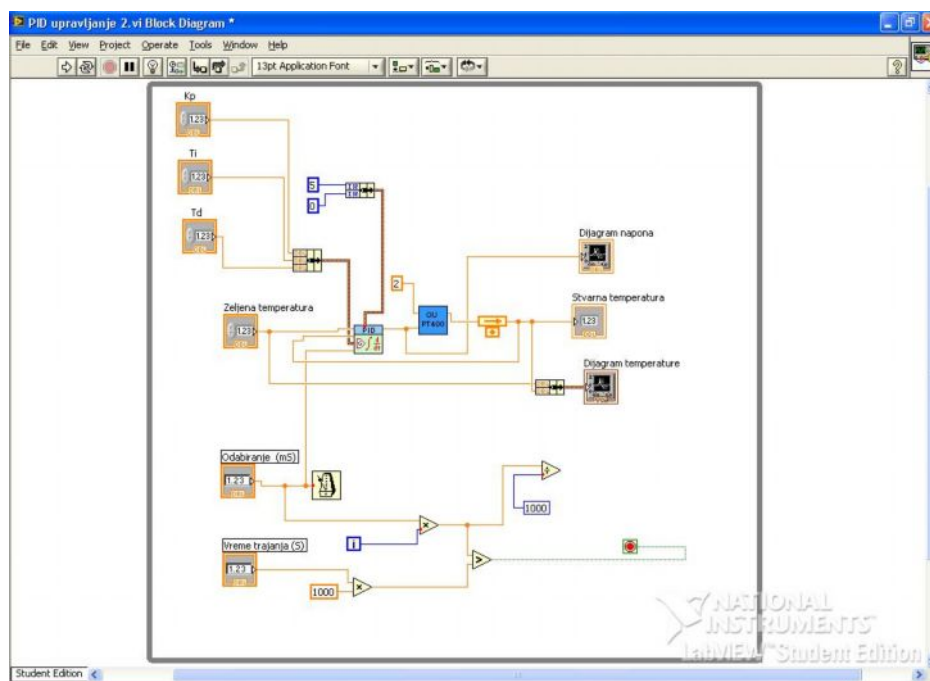
$$G(s) = \frac{K}{(1 + T_{p1} \cdot s) \cdot (1 + T_{p2} \cdot s)} \cdot e^{(-T_d \cdot s)}$$

Parametri modela :

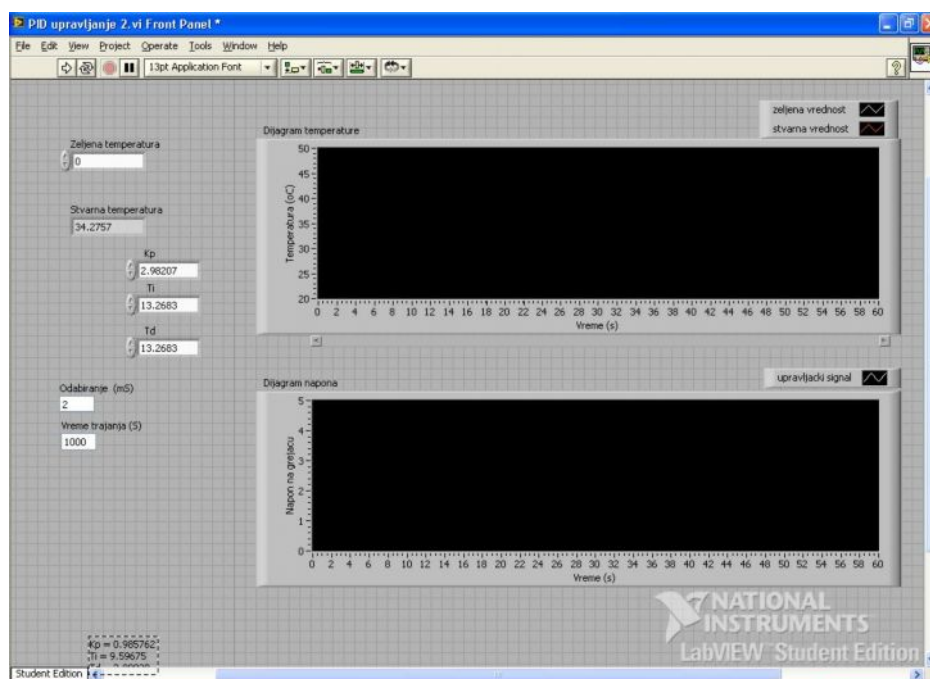
- K = 0.99886
- Tp1 = 3.8939
- Tp2 = 0.78452
- Td = 0.75775

12. UPRAVLJANJE PROCESOM

Upravljanje procesom radi se u LabVIEW. U ovom programu napravljen je upravljački algoritam koji sadrži PID kontroler.



Slika 38 - Blok šema



Slika 39 - Radno okruženje

13. PERIODA ODABIRANJA

Perioda odabiranja je osnovni parameter diskretizacije signala ili sistema. Otuda se ona bira saglasno modelu ili raspoloživim informacijama o kontinualnom signalu ili (i) sistemu. U tom smislu je definisana Šenonova teorema odabiranja koja nameće osnovne uslove za izbor periode odabiranja (T):

$$T \leq \frac{\pi}{\omega_0}$$

Dopuštajući da pretek faze može biti smanjen za 5° do 15° , pretpostavljajući da je “antialiasing” filter oblika i sumirajući navedene negativne uticaje izvodi se opšta formula:

$$T \cdot \omega_1 \approx 0.05 \div 0.14$$

Gde je ω_1 (rad/s) presečna učestanost kontinualnog dela sistema

Gore navedena relacije se može prihvatiti kao pravilo za određivanje periode odabiranja. Ili, ako bi pretpostavili da “antialiasing” filter neće biti implementiran u sistem, prethodno pravilo se redukuje na:

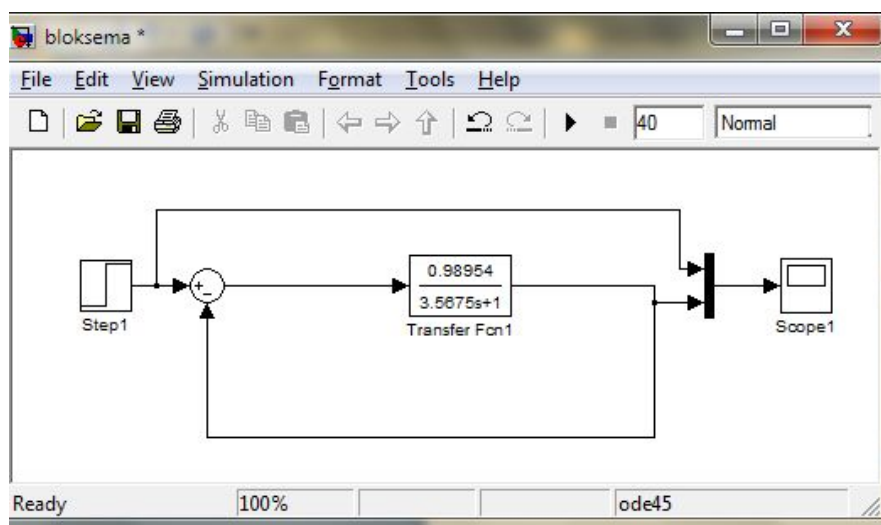
$$T \cdot \omega_1 \approx 0.15 \div 0.5$$

Dakle manja perioda odabiranja doprinosi sprečavanju efekta transponovanja frekvencije i tačnijoj rekonstrukciji diskretizovanog signala, zatim, povećanju rezerve stabilnosti digitalnog sistema upravljanja sa zatvorenim povratnom spregom. Takođe se može pokazati da smanjenje periode odabiranja ubrzava odziv sistema i podiže tačnost rada digitalnog sistema u stacionarnom stanju, generalno poboljšava dinamičke karakteristike sistema, i povećava efikasnost sistema u otklanjanju i potiskivanju uticaja različitih tipova poremećaja.

Međutim, smanjivanje periode odabiranja povećava propusni opseg sistema, samim tim i osetljivost sistema na šum i greške modeliranja, a može dovesti u pitanje i mogućnost fiičke realizacije sistema. Previše mala perioda odabiranja može ugroziti i zahtev za radom u realnom vremenu. Današnji komercijalni digitalni kontroleri najčešće imaju fiksnu periodu odabiranja reda veličine 200 ms.

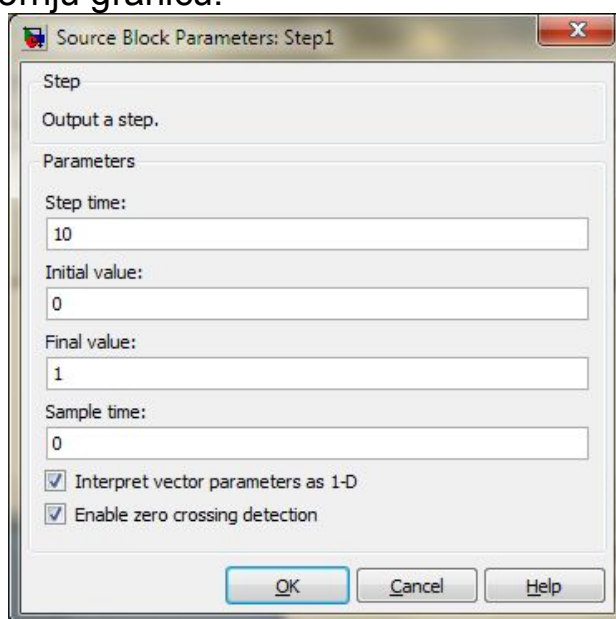
Sada pristupamo praktičnom određivanju maksimalne periode odabiranja. Jedan od načina za određivanje dat je sledećim konceptom.

Na sledećoj slici data je blok šema sistema sa povratnom spregom bez vremenskog kašnjenja. Sastoji se od bloka odskočnog signala, bloka prenosne funkcije sistema i osciloskopa. Signal sa ulaza se šalje na blok za sabiranje, kao i na jedan od dva ulaza bloka za spajanje signala koji se vodi na osciloskop. Signal sa izlaza bloka prenosne funkcije se vraća povratnom spregom na sabirač (u ovom slučaju se oduzima) od ulaznog signala. Korigovani signal se ponovo šalje u prenosnu funkciju i vodi na drugi ulaz bloka za spajanje signala.



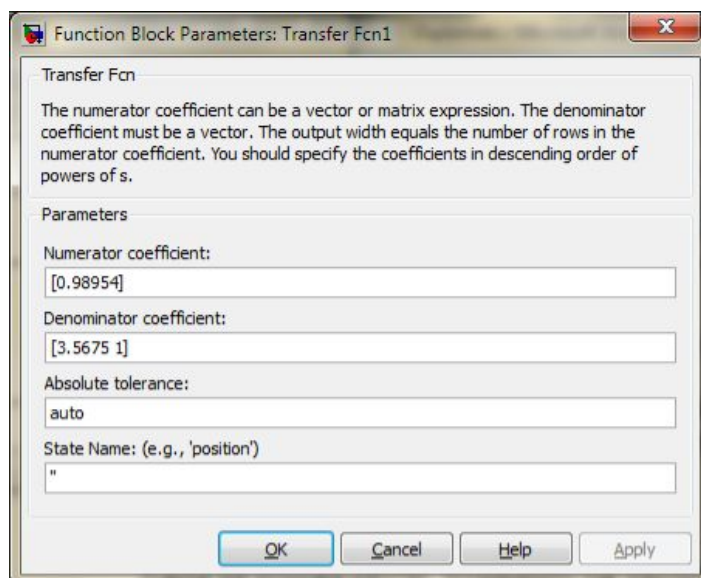
Slika 40 - Šema sistema sa povratnom spregom

U bloku *Step1* zadajemo trenutak u kojem se dešava promena i definišemo donju i gornju granicu.



Slika 41 - Step blok

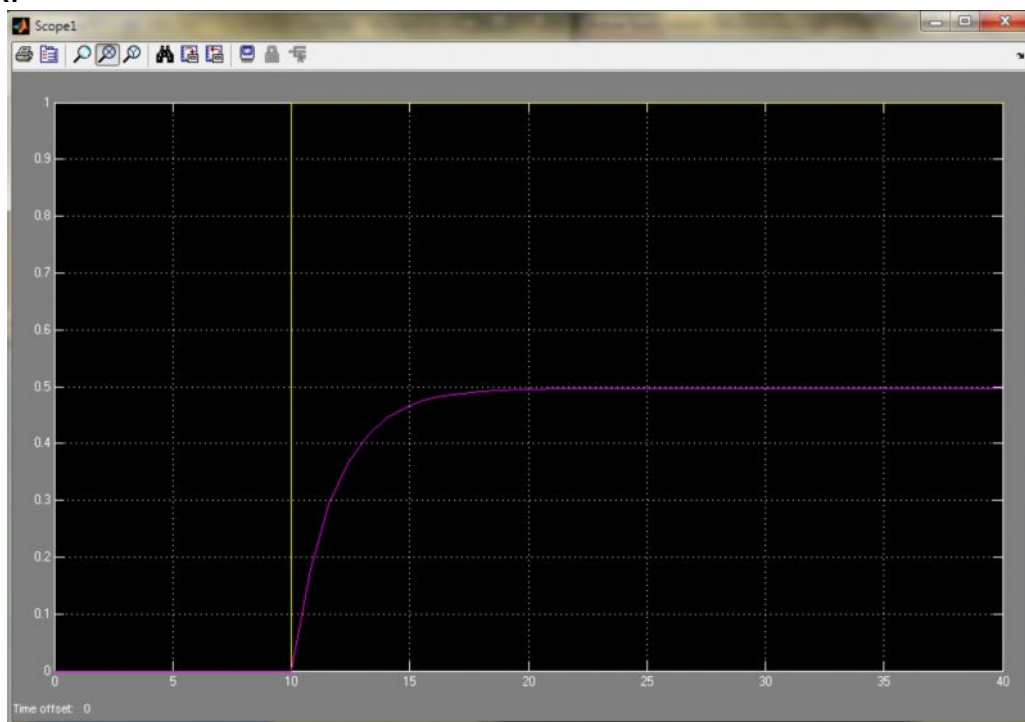
U bloku *Transfer function* zadajemo funkciju prenosa sistema (u ovom slučaju sistema drugog reda) preko proporcionalnog, integralnog i diferencijalnog koeficijenata.



Slika 42 - *Transfer function blok*

Po završetku ovih podešavanja blokova, odabiramo vreme trajanja procesa (u ovom slučaju 40 sekundi) i startujemo proces klikom na dugme *play*.

Kada se proces završi, dvolikom na osciloskop dobijamo sledeći grafik:



Slika 43 - *Odziv sistema*

Uzimamo 63% visine signala od trenutka kada je različit od nule do trenutka kada dostiže svoj maksimum, i tu vrednost delimo sa 10. Na taj način dobijamo maksimalnu periodu odabiranja u sekundama. Za ovaj slučaj ona iznosi $0,25s = 250ms$.

14. PODEŠAVANJE PARAMETARA PID KONTROLERA

Za podešavanje parametara PID kontrolera korišćena je knjiga "Handbook of PI and PID Controller tuning rules 1".

Knjiga "Handbook of PI and PID Controller tuning rules 1" sveobuhvatno sadrži pravila za podešavanje predloženih kontrolera u poslednjih sedam decenija. Pravila su pažljivo kategorizovana i data je primena podataka o svakom od njih. U knjizi se govori o arhitekturi kontrolera i problemima procesa modeliranja, kao i efikasnost robusnosti petlje kompenzovane PI ili PID kontrolerima.

Koristeći obrasce iz gore navedene knjige dobijeni su sledeći koeficijenti za podešavanje PID kontrolera za upravljenje prvim senzorom.

Jednačine po kojima se računaju koeficijenti:

$$K_c = \frac{1}{K_m} \cdot \left(1.35 \cdot \frac{T_m}{\tau_m} + 0.25 \right)$$

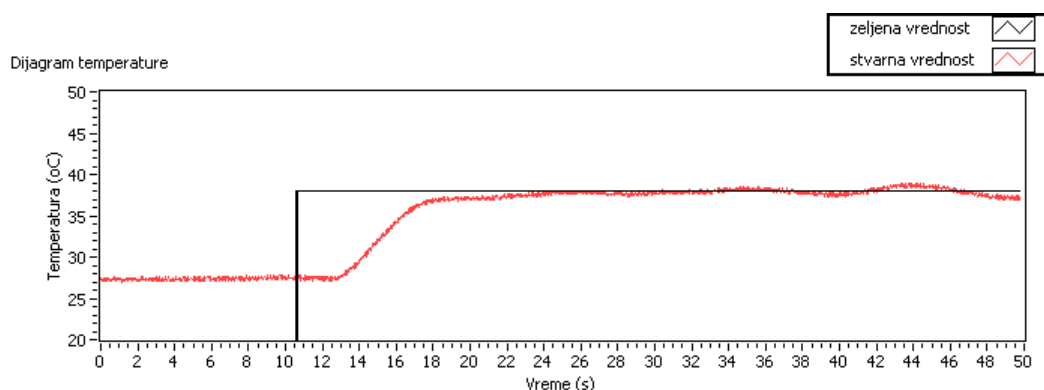
$$T_i = T_m \cdot \left(\frac{2.5 \cdot \frac{\tau_m}{T_m} + 0.46 \cdot \left(\frac{\tau_m}{T_m} \right)^2}{1 + 0.61 \cdot \frac{\tau_m}{T_m}} \right)$$

$$T_d = 0.365 \cdot T_m \cdot \left(\frac{\tau_m}{T_m} \right)^{0.950}$$

Koeficijenti dobijeni posle računanja:

- $K_c = 2.379378$
- $T_i = 4.5975012$
- $T_d = 0.8540522$

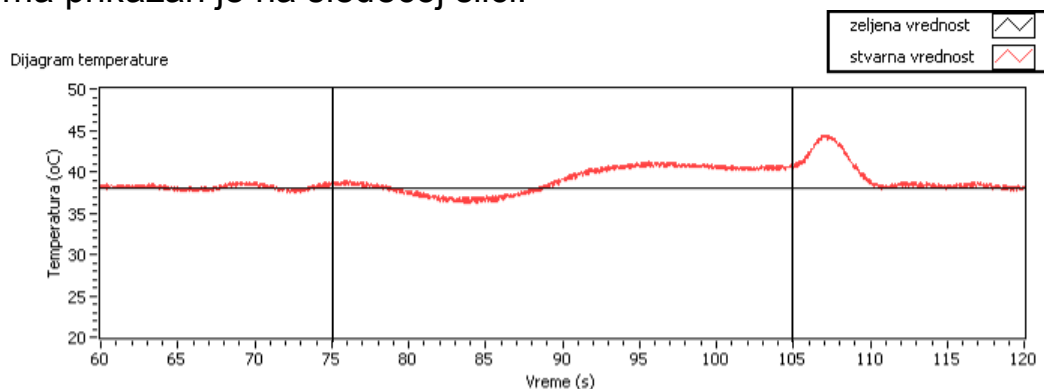
Odziv sistema sa prvim senzorom na promenu temperature sa 28°C na 35°C.



Slika 44 - Grafik promene temperature na prvom senzoru

Primećujemo da kada u 10 sekundi procesa promenimo temperaturu sa 28°C na 35°C, prvom senzoru je potrebno 15 sekundi da dostigne temperaturu od 35.

Vrlo bitna stvar kod upravljanja procesom je ponašanje sistema prilikom dejstva poremećaja. Mi unosimo poremećaj u sistem tako što zatvaramo kraj tunela i držimo ga zatvorenim određeno vreme. Odziv sistema prikazan je na sledećoj slici:



Slika 45 - Uticaj poremećaja na održavanje temperature

U ovom procesu tunel je zatvoren u 75 sekundi a otvoren u 105 sekundi. Primećujemo da prilikom zatvaranja tunela temperatura se povećava za par stepeni i ta temperatura se održava dok je tunel zatvoren. Takođe možemo primetiti da temperatura naglo skače prilikom otvaranja tunela. To se dešava zbog toga što se topao vazduh, dok je tunel zatvoren, nagomilava oko grejača i nema cirkulacije vazduha oko senzora. Nakon otvaranja tunela topao oblak vazduha sa grejača kreće i prelazi preko senzora i naglo povećava temperaturu. Nakon tog toplog talasa PID kontroleru je potrebno samo par sekundi (3-4 sekunde) da

vрати температуру u tunelu na željenu температуру (u ovom slučaju na 38°C).

Za podešavanje PID kontrolera, prilikom upravljanja poslednjim senzorom, nemožemo da koristimo iste obrazce kao za upravljanje prvim sezorom. To je zato što smo sistem sa poslenjim senzorom pretpostavili kao sistem drugog reda. Zato i jednačine podešavanja PID-a moraju da se odnose na sistem drugog reda. Te jednačine pronađene su u knjizi "Handbook of PI and PID Controller tuning rules 1"

Jednačine po kojima se računaju koeficijenti:

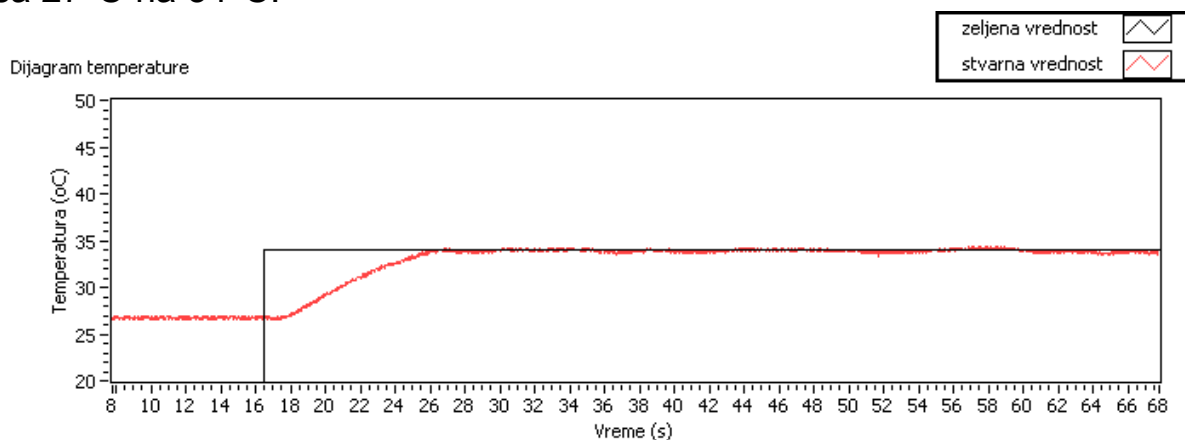
$$K_c = \frac{5.5030}{K_m} \cdot \left[\left(10 \cdot \frac{\tau_m}{T_{m1}} \right)^{(-1,1445 + 0,1100 \cdot \ln(10 \cdot \frac{\tau_m}{T_{m1}}))} \cdot \left(10 \cdot \frac{T_{m2}}{T_{m1}} \right)^{(0,1648 + 0,1709 \cdot \ln(10 \cdot \frac{T_{m2}}{T_{m1}}) - 0,2142 \cdot \ln(10 \cdot \frac{\tau_m}{T_{m1}}))} \right]$$

$$T_i = T_d = 1,8681 \cdot T_{m1} \cdot \left[\left(10 \cdot \frac{\tau_m}{T_{m1}} \right)^{(0,6212 - 0,0760 \cdot \ln(10 \cdot \frac{\tau_m}{T_{m1}}))} \right]$$

Koeficijenti dobijeni nakon računanja:

- $K_c = 2,982078232$
- $T_i = 13,26831474$
- $T_d = 13,26831474$

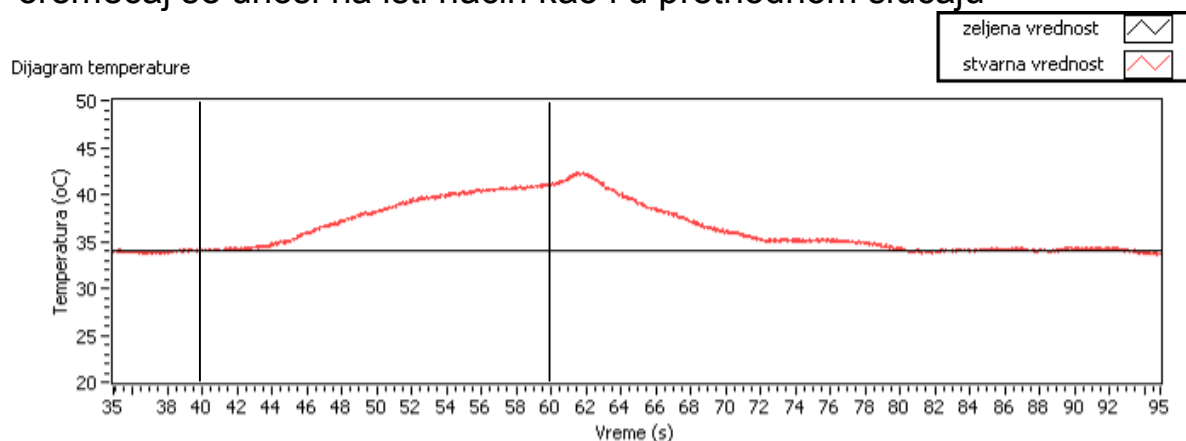
Odziv sistema sa poslednjim senzorom na promenu temperetaure sa 27°C na 34°C.



Slika 46 - Grafik promene temperature na poslednjem senzoru

Možemo primetiti, sa gore navedene slike, da je poslednjem senzoru potrebno desetak sekundi da promeni temperaturu sa 27°C na 34°C, i da je pravilno održava.

I ovde proveravamo kako se sistem ponaša na uticaj poremećaja. Poremećaj se unosi na isti način kao i u prethodnom slučaju



Slika 47 - Uticaj poremećaja na održavanje temperature

U ovom procesu tunel je zatvoren u 40-oj sekundi a otvoren u 60 sekundi. Ako sada uporedimo ovaj grafik sa gornjom slikom, očigledno je da je proces sličan ali da su parametri malo drugačiji. Poremećaj na sistem inicijalno deluje dosta sporije. To je zato što treba mnogo više vremena da se konvekcijom toplota prenese od grejača na drugi kraj tunela. Poremećaj je ekstremniji zato što poslednji sensor normalno nije mogao da se zagreje onoliko koliko je prvi mogao, pa je ovo zatvaranje tunela omogućilo da se temperatura popne više nego obično. Kada je tunel otvoren došlo je do naglog skoka temperature. Međutim taj skok je manji nego kod prvog senzora. To je zato što je uticaj akumulirane toplote na prvom senzoru veći nego na poslednjem zbog disipacije toplote duž tunela. Po otvaranju tunela PID kontroleru potrebno je 20-tak sekundi da se vrati u normalu, što je više nego prilikom upravljanja sistema prvim senzorom.

15. ZAKLJUČAK

U ovom radu prikazan je uspešan process identifikacije sistema i podešavanje PID kontrolera. Pokazano je da prvi senzor brže reaguje na upravljanje od poslednjeg senzora. Takođe, kada se deluje poremećajem, sistem sa poslednjim senzorom je mnogo više podložan uticaju tog poremećaja od sistema sa prvim senzorom.

16. LITERATURA

- [1] M. Matijević, G. Jakupović, J. Car, *Računarski podržano merenje i upravljanje*, Mašinski fakultet, Kragujevac, 2005
- [2] Aidan O'Dwyer , *Handbook of PI and PID Controller tuning rules*, Imperial College Press, London, 2006
- [3] www.wikipedia.org , internet stranica
- [4] Milić R. Stojić, *Digitalni sistemi upravljanja*, Elektrotehnički fakultet, Beograd, 2004
- [5] www.ni.com , internet stranica
- [6] <http://home.hit.no/> , internet stranica