wusaustria

Austrian Cooperation
Eastern Europe

Internet-Mediated Process Control Laboratory

Milan Matijević, Vladimir Cvjetković, Vesna Ranković, Miladin Stefanović

- the publishing of this book was financed by Austrian Cooperation through WUS Austria -
- within the eLearning project No.: eLTF 8093-019-01/2006

- this copy is not for sale -

Upravljanje laboratorijskim procesima posredstvom Interneta

Milan Matijević, Vladimir Cvjetković, Vesna Ranković, Miladin Stefanović

- objavljivanje ovog laboratorijskog praktikuma omogućili su Austrian Cooperation i WUS Austria u okviru eLearning projekta eLTF 8093-019-01/2006

- besplatan primerak -

Mašinski fakultet u Kragujevcu
Jun, 2007.

Autori:

dr Milan Matijević, *vanredni profesor Mašinskog fakulteta u Kragujevcu*
dr Vladimir Cvjetković, *docent, Institut za Fiziku, PMF u Kragujevcu*
dr Vesna Ranković, *docent Mašinskog fakulteta u Kragujevcu*
dr Miladin Stefanović, *docent Mašinskog fakulteta u Kragujevcu*

UPRAVLJANJE LABORATORIJSKIM PROCESIMA POSREDSTVOM INTERNETA
I izdanje

ISBN 978-86-86663-12-2

Recenzenti:

dr Ilija Nikolić, *redovni profesor Mašinskog fakulteta u Kragujevcu*
dr Jasna Radulović, *vanredni profesor Mašinskog fakulteta u Kragujevcu*

Izdavač:

Mašinski fakultet u Kragujevcu
Sestre Janjić 6, 34000 Kragujevac

Za izdavača:

dr Miroslav Babić, *dekan Mašinskog fakulteta u Kragujevcu*

Korice:

dr Miladin Stefanović, dipl.ing.

Tiraž:

50 primeraka

Štampa:

InterPrint, Jurija Gagarina 12, 34000 Kragujevac

Odlukom Nastavno-naučnog veća Mašinskog fakulteta u Kragujevcu br. 01-1312/1 od 14.06.2007. godine, ova knjiga je odobrena da se štampa kao pomoćni univerzitetski udžbenik.

NAPOMENA: Reprodukovanje (fotokopiranje ili umnožavanje na bilo koji način) ili ponovno objavljivanje ove knjige – u celini ili u delovima – nije dozvoljeno bez prethodne izričite saglasnosti i pismenog odobrenja autora.

Predgovor

Ova knjiga je nastala kao rezultat projekta eLTF 8093-019/2006, koji je podržala i finansirala WUS Austria. Njegova namena je da podrži razvoj Web Laboratorije Univerziteta u Kragujevcu (http://weblab.kg.ac.yu), i nastavu iz više predmeta, koji se predaju na Univerzitetu u Kragujevcu. Želja autora je da ova publikacija bude od koristi studentima tehničkih fakulteta i fakulteta prirodnih nauka u okviru različitih predmetnih oblasti poput modeliranja dinamičkih sistema, informacionih sistema, upravljanja tehničkih sistema, mernih sistema i tehnika, itd. Naša namera je da nastavimo sa unapređivanjem sadržaja ove knjige, i nastavnih metoda koje će pomoći studentima da što kvalitetnije i brže savladaju predmetnu oblast. Zato, najiskrenije, molimo čitaoce za povratne informacije, komentare i sugestije, koje će nam pomoći u pripremi sledećeg izdanja, i unapređenju Internet servisa.

Autori duguju određenu zahvalnost mr Miroslavu Ravliću, dipl.el.ing. koji je doprineo u izgradnji eksperimentalne aparature i softverskog rešenja za akviziciju signala i upravljanje u programskom jeziku Delphi.

Autori se zahvaljuju i recenzentima Prof. dr Iliji Nikoliću i Prof. dr Jasni Radulović koji su uložili napor da rukopis pažljivo pročitaju i svojim sugestijama doprinesu kvalitetu ovog izdanja.

U Kragujevcu, jun 2007. godine *Autori*

Sadržaj

PREDGOVOR..... 3
SADRŽAJ..... 5
1 UVODNA RAZMATRANJA 7
1.1 PRIMENA ICT TEHNOLOGIJA U EDUKACIJI..... 7
1.2 KLASIFIKACIJA EKSPERIMENTALNIH OKRUŽENJA..... 8
1.3. WEB - BAZIRANE LABORATORIJE -TELELABORATORIJE..... 11
1.4 ZAHTEVI PRI KREIRANJU TELEEKSPERIMENTA..... 12
2 LABORATORIJSKI MODELI SPREGNUTIH REZERVOARA..... 15
2.1 AKTUELNI LABORATORIJSKI MODEL I EKSPERIMENTALNA APARATURA 21
2.2 MODELIRANJE SISTEMA SA JEDNIM REZERVOAROM..... 22
3 EDUKATIVNI CILJEVI I ZADACI LABORATORIJSKE VEŽBE..... 29
3.1. PRIMER PROJEKTNOG ZADATKA..... 30
4 PRIMERI REŠAVANJA PROBLEMA SA IZVODIMA IZ TEORIJE 35
4.1 IDENTIFIKACIJA KARAKTERISTIKA SISTEMA..... 35
4.2 KONVENCIONALNE METODE UPRAVLJANJA..... 36
4.3. FAZI UPRAVLJANJE..... 42
4.3.1 Primena fazi upravljanja..... 42
4.3.2 Fazi skupovi i fazi logika – osnovni pojmovi..... 43
4.3.3 Osnovna struktura fazi kontrolera..... 50
4.3.4 Sinteza fazi PI kontrolera..... 51
4.3.5 Ciljevi i mogući zahtevi laboratorijske vežbe..... 54
5 AKVIZICIJA SIGNALA I UPRAVLJANJE U REALNOM VREMENU U FUNKCIJI IZVRŠENJA EKSPERIMENTA 56
5.1 SISTEMI ZA AKVIZICIJU I UPRAVLJANJE – PRINCIPI PROGRAMIRANJA..... 56
5.2 OSNOVE KORIŠĆENJA PROGRAMSKOG PAKETA LABVIEW..... 60

5.3 SOFTVERSKA APLIKACIJA I IZVRŠENJE EKSPERIMENTA..... 71
6 KORIŠĆENJE LABORATORIJSKE VEŽBE PUTEM INTERNETA..... 72
6.1 STRUKTURA WEB LABORATORIJE..... 72
DODATAK A – SINTEZA FAZI KONTROLERA U PROGRAMSKOM PAKETU MATLAB..... 74
DODATAK B – LISTING DELPHI PROGRAMA ZA AKVIZICIJU SIGNALA I UPRAVLJANJE NA LOKALNOM RAČUNARU..... 78
DODATAK C – OPŠTA UPUTSTVA ZA PISANJE RADA SA ELEMENTIMA MATEMATIČKOG TEHNIČKOG PRAVOPISA..... 83
LITERATURA..... 86

1. Uvodna razmatranja

1.1 PRIMENA ICT TEHNOLOGIJA U EDUKACIJI

Razvoj ICT tehnologija u značajnoj meri utiče na sve oblasti ljudskog delovanja i rada. Sama edukacija nije nikakav izuzetak. U procesu edukacije, ICT tehnologija i njena primena utiču na različite aspekte nastavnog procesa, počev od podučavanja (odnosno načina, sredstava i mogućih vidova transfera znanja od predavača ka studentu), preko procesa učenja (odnosno načina na koji student prihvata informacije i znanje u procesu edukacije uz eventualno korišćenje ICT), pa sve do didaktike i globalnog modela komunikacije između predavača i studenata sa jedne strane i studenata međusobno sa druge strane.

Različite tehnologije koje se koriste u procesu edukacije generisaće, pak, nove probleme, koje sa sobom, uostalom, nosi svaka novina i nova tehnologija. Otuđenje studenata od okoline i njihovo odvajanje od zajedničkog nastavnog procesa je jedan od najčešće pominjanih problema.

U svakom slučaju, primena ICT u procesu edukacije dobijaće na sve većem značaju. Danas je moguće kreirati sisteme podučavanja u kojima ICT tehnologija ima značajno mesto, pri čemu je poželjno i moguće ostvariti:

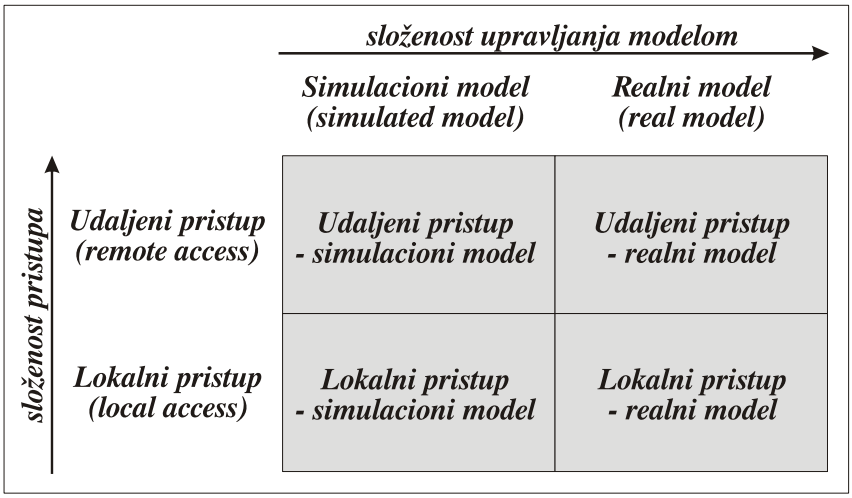
- bolju interakciju između ljudi i mašina,
- prirodan i interaktivan grafički korisnički interfejs,
- visok nivo interaktivnosti.

U ciju projektovanja tehničkih sistema ili jednostavnog razumevanja fizičkih zakona koji opisuju njihovo ponašanje, naučnici i inženjeri često koriste računare za proračun i grafičku prezentaciju različitih veličina. Ponekad je različit broj merenih i potrebnih veličina međusobno povezan tako da sve veličine zajedno predstavljaju jednu sliku realnog sistema. Razumevanje ovih relacija je često veoma važan korak u uspešnom učenju i savlađivanju nastavnog programa u brojnim inženjerskim i naučnim disciplinama. Ukoliko je studentima omogućeno da sve izmerene veličine imaju grafički prikazane vrednosti, oni mogu jednostavnim menjanjem pojedinih parametara da prate promenu drugih parametara, odnosno da opserviraju međusobnu zavisnost pojedinih parametara, kao i da prate ponašanje sistema u celini. Dinamičko i grafičko prezentovanje podataka nije samo od izuzetne važnosti za

inženjersku edukaciju, već je izuzetno značajno za sticanje iskustava u analizi, sintezi i projektovanju. Različiti interaktivni alati koji mogu biti smešteni na Internetu, predstavljaju značajan resurs koji omogućava razvijanje inženjerskih sposobnosti i znanja kod studenata. Svaki koncept grafičke vizuelizacije i interaktivnosti demistifikuje apstraktne matematičke koncepte, naročito ako su prezentovani kroz primere i kroz praktične laboratorijske primere i vežbe. Postojanje interaktivnih softverskih alata i različitih eksperimentalnih okruženja, prema tome, pružaju značajne koristi u kvalitetnoj i uspešnoj inženjerskoj, ali i svakoj drugoj edukaciji.

1.2 KLASIFIKACIJA EKSPERIMENTALNIH OKRUŽENJA

Pod samim pojmom eksperimentalnog okruženja mogu se svrstati različiti modaliteti i sistemi eksperimenata. Suštinski postoji veliki broj različitih klasifikacija eksperimentalnih okruženja.



Slika 1. ***Klasifikacija eksperimentalnih okruženja***

Međutim, za potrebe ove analize mogu se definisati dva jasna kriterijuma koja omogućavaju preciznu klasifikaciju ovih sistema, posmatrajući iz ugla korisnika sistema (studenta) i to:

1. Prvi kriterijum predstavlja način pristupa resursima u svrhu ostvarivanja eksperimenta. Prema ovom prvom kriterijumu postoje dva osnovna načina pristupa resursima i to:
    a. udaljeni pristup (*remote access*) korišćenjem Internet infrastrukture i
    b. lokalni pristup (*local access*) gde nije neophodan pristup Internetu da bi se ostvarila konekcija sa laboratorijskim eksperimentom.
2. Drugi kriterijum predstavalja priroda fizičkog sistema. Prema ovom kriterijumu moguće je napraviti sledeću podelu:

a. simulacioni modeli (*simulated models*), odnosno sistemi kod kojih ne postoji stvarna laboratorijska oprema kojom se upravlja, već egzistira simulacioni softver koji simulira ponašanje realnog objekta i
b. relani modeli (*real plants*), odnosno sistemi gde postoje realni, fizički objekti kojim se upravlja.

Kombinovanjem ova dva kriterijuma, može se doći do četiri vrste okruženja, koja su po svojoj prirodi veoma različita, ali donekle konvergiraju u ostvarivanju pojedinih edukacijskih ciljeva (slika 1):

- *Lokalni pristup – realni model* predstavlja tradicionalni, klasični, pristup laboratorijskim vežbama. U ovom okruženju kompjuter je povezan na realni, stvarni, laboratorijski eksperiment i pomoću odgovarajućeg, pratećeg, hardvera (AD i DA konvertora) i softvera vrše se upotrebna merenja i akvizicija podataka. U ovom slučaju student, koristeći računar, upravlja realnim laboratorijskim eksperimentom, prema unapred utvrđenoj proceduri.
- *Lokalni pristup – simulacioni model,* predstavlja takvo okruženje koje je u celosti bazirano na različitim softverskim komponentama. Koristeći grafički korisnički interfejs korisnik upravlja softverskim paketom koji sumulira ponašanje realnog fizičkog objekta. Znači, ovde se vrši upravljanje virtualnim, softverskim objektom upravljanja koji predstavlja simulaciju realnog fizičkog objekta ili laboratorijskog entiteta. Karakteristika lokalnog pristupa je da samo jedan student može da pristupa simulacionom softveru i da sprovodi predviđenu proceduru, tako da se ovaj pristup često karakteriše kao mono-korisnička virtualna laboratorija (*mono-user virtual laboratory*).
- *Udaljeni pristup – simulacioni model,* je vrsta eksperimenta koji je u mnogome sličan sa prethodno opisanim okruženjem. Korisnici u ovom slučaju pristupaju softveru koji ima zadatak da simulira ponašanje realnog sistema, preko Interneta. Osnovna razlika ovog i prethodnog sistema je u tome što u ovom slučaju nekoliko korisnika može simultano da koristi isti model, zato se često ovaj sistem i naziva više-korisnička virtualna laboratorija (*multi-users virtual laboratory* ili *WBS*).
- *Udaljeni pristup – realni model,* predstavlja pristup stvarnoj, realnoj, laboratorijskoj opremi preko Interneta. Koristeći eksperimentalni interfejs korisnici sa udaljene lokacije, pristupaju stvarnoj laboratorijskoj opremi i vrše potrebne eksperimente. Ovaj pristup je najkomplikovaniji i često se označava kao udaljene laboratorije (*remote laboratory)*, telelaboratorije (*telelaboratory*), ili teleupravljanje (*teleoperatio*) preko WWW.

Udaljeni pristup virutalnom ili realnom eksperimentu preko Interneta je zahtevniji i komplikovaniji od lokalnog pristupa laboratorijskim eksperimentima. Najkomplikovaniji, najzahtevniji pristup, koji ima najveći edukativni i svaki drugi potencijal je pristup u kome se ostvaruje udaljen pristup realnom laboratorijskom eksperimentu, odnosno realnom objektu. Ovaj pristup nije ograničen samo na edukativnu primenu, naprotiv, ovaj pristup je značajan i za industrijsku primenu, pošto može omogućiti realnu primenu i pristup realnoj opremi u cilju ostvarivanja

određenih inženjerskih i istraživačkih zadataka. Ovaj pristup omogućuje istraživačkim centrima iz industrije da na neki način "iznajmljuju" laboratorijsku opremu i da sprovode potrebne eksperimente na opremi koja može biti relativno skupa.

Da bi ove laboratorije koje se često nazivaju i telelaboratorije uspešno funkcionisale potrebno je ostvariti potreban broj zahteva i to:

- Jednostavnost instalacije i korišćenja – Eksperimentalno okruženje mora biti jednostavno za instalaciju i potencijalno korišćenje. Zbog činjenice da je predavač odsutan u momentu izvođenja vežbi, potrebno je imati detaljno objašnjenje fizičke prirode eksperimenta, kao i sa jedne strane jednostavno, a sa druge dovoljno detaljno i precizno korišćenje laboratorijske opreme.
- Pristup korišćenju Interent infrastrukture – Internet infrastuktura je mrežna infrastruktura koja je dovoljno raširena i prihvaćena i zasniva se na poznatim protokolima uz svoje poznate karakteristike (jednakost servisa, nezavisnost softverske i hardverske platforme i druge) i prema tome predstavlja idealno okruženje za upravljanje laboratorijskim eksperimentom. Mogućnost da se korišćenjem Interneta, koristeći standardan softver (*browser*) pristupi laboratorijskoj vežbi predstavlja jedan od glavnih preduslova u formiranju laboratorija kojima se upravlja na daljinu. S tim u vezi značajna je činjenica da studenti mogu da koriste *open-soruce* software tj. besplatni softver koji im omogućava da pristupe laboratorijskim vežbama i da upravljaju njima.
- Interaktivnost i realnost – predstavljaju karakteristike ovih sistema koje imaju zadatak da omoguće da studenti aktivno učestvuju u procesu sprovođenja laboratorijskog eksperimenta. Potrebno je da laboratorijska vežba omogućava studentu da aktivno učestvuje i da mu predstavi realne rezultate i izlaze vežbe. Postojanje odgovarajućih web/IP kamera omogućava postizanje visokog stepena interaktivnosti i osećaja postojanja realnog sistema u procesu sprovođenja laboratorijske vežbe.
- Dostupnost – je karakteristika koja delimično proističe iz prethodnih karakteristika. Korišćenje Internet infrastrukture omogućava da se pristupi laboratorijskoj vežbi sa bilo kog mesta, tome treba dodati i potrebu da laboratorijske vežbe treba da budu dostupne sve vreme. Jedino ograničenje može biti situacija kada neki drugi korisnik već koristi laboratorijsku opremu, u tom slučaju potencijalni korisnik može u realnom vremenu preko video zapisa pratiti odvijanje laboratorijske vežbe sve dok ne bude u situaciji da sam sprovede potrebni eksperiment.

Ovo su najznačajnije karakteristike pri definisanju potrebnih karakteristika laboratorija kod kojih se ostvaruje udaljeni pristup realnoj laboratorijskoj opremi. Sem ovih karakteristika postoje i druge kao što su: mogućnost *download*-a eksperimentalnih rezultata, funkcionisanje klijent modula na svim raspoloživim platformama, dokumentacija svih rezultata, mogućnost lakog upravljanja izmenama i distribucija novih verzija, modularna i otvorena arhitektura sistema i drugo.

1.3 WEB - BAZIRANE LABORATORIJE –TELELABORATORIJE

Laoboratorijski eksperimenti imaju značajnu ulogu i inženjerskoj i tehničkoj edukaciji. Kao što je bilo prezentovano u pojedinim slučajevima pojedini instrumenti u virutalnim laboratorijama mogu biti zamenjeni odgovarajućim simulacionim softverom. Iako je ovaj pristup značajan i koristan i pored svojih mogućnosti simaulacije i vizuelizacije, on jednostavno ne može da zameni pristup u kome se upravlja realnom, stvarnom laboratorijskom opremom i gde se sprovode realni eksperimenti.

Jedan od značajnih segmenata inženjerske edukacije je omogućavanje studentima da rade sa realnom opremom, da mere stvarnim instrumentima i da rešavaju realne probleme. Osnovna ideja web-baziranih telelaboratorija je sprovođenje stvarnih eksperimenata, u realnom vremenu, na realnoj opremi ali preko Interneta. Postoji veliki broj različitih primena ovog koncepta i različitih načina upravljanja realnom laboratorijskom opremom. Ono što je zajedničko za sve pristupe je da se udaljeni korisnik (student) povezuje na Internet koristeći određeni web server koji je povezan sa kompjuterom koji je zadužen za upravljanje (i/ili nadzor) laboratorijskom opremom.

Ovakav pristup gde se realnom laboratorijskom opremom upravlja preko Interneta ima veliki broj prednosti:

- Laboratorija je dostupna sve vreme (24/7). Vreme kojim student raspolaže je personalizovano i slobodno, tj. studenti mogu samostalno da planiraju svoje obaveze.
- Studenti ne moraju da fizički budu prisutni u laboratoriji da bi vršili eksperiment. Ovo je naročito značajno jer omogućuje ostvarivanje koncepta *distant learninga*-a i mogućnost permanentnog obrazovanja ljudi koji su dislocirani iz unverzitetskih i edukativnih centara.
- Optimalno korišćenje resursa. Povećanjem vremena dostupnosti laboratorijske opreme i mogućnosti punog pristupa laboratoriji bez obzira na pristupnu lokaciju vodi boljem korišćenju laboratorijskih resursa. A pri svemu ovome, studenti racionalnije koriste postojeće resurse uz niže troškove laboratorije i niže fiksne troškove.
- Bolja pripremljenost za eksperiment. Pristupi laboratorijskoj opremi omogućavaju studentima da se bolje upoznaju sa teorijskim sadržajem i da posmatraju rezultate i postupke koji čine njihove kolege. Sem toga studenti mogu samostalno da probaju eksperiment i da ga ponove onoliko puta koliko je to njima potrebno.
- Primenom ovog pristupa omogućava se koncept permanentnog obrazovanja, odnosno stalnog usavršavanja za zaposlene, a što je još važnije omogućava se da studenti i ljudi sa specijalnim potrebama budu ravnopravni u edukativnom procesu.
- Rezultati u implementaciji i korišćenju web baziranih laboratorijskih vežbi pokazale su da one imaju značajan motivacioni i edukativni efekat i da poboljšavaju organizaciju i kvalitet nastave. Kroz analizu povratnih

informacija od studenata pokazalo se da je ovakav pristup visoko vrednovan u inženjerskoj edukaciji.

Velika pogodnost je što web laboratorija može biti deo integrisanog (vertikalno) eLearning okruženja koje studentima pruža i teorijski i praktični deo, kao i deo za samovrednovanje. Sa druge strane moguće je ostvariti (horizontalno) povezivanje laboratorija i stvaranje Univerzitetskih mreža web baziranih laboratorija. Sve napred nabrojano ide u prilog dinamičkom *eLearning* okruženju i velikim mogućnostima koje može da pruži čak i jeftina laboratorijska oprema ako je dobro organizovana.

Naravno i pored brojnih prednosti ovakav način definisanja i izvođenja laboratorijskih eksperimenata ima i svojih mana, kao na primer:

- Nepostojanje direktnog fizičkog kontakta sa eksperimentom. Fizičko odvajanje studenta od eksperimenta može dovesti do toga da se izgubi osećaj realne dimenzije eksperimenta. Da bi se prevazišao ovaj problem preporučljivo je korišćenje multimedijalnih alata, video *streaming*-a, zvuka u cilju postizanja bolje prezentacije samog eksperimenta.
- Pojedini eksperimenti po svojoj suštini nisu kompatibilni sa ovakvim načinom sprovođenja. U slučaju kada je reč o sporim procesima ili o postupcima gde je potrebno preduzeti čitav niz koraka radi ostvarivanja određene procedure (na primer hemijski eksperimenti) upravljanje eksperimentom preko Interneta ne može se realizovati u dovoljno kvalitetnoj i potrebnoj meri. Jedan od načina je eventualno kreiranje virtualne laboratorije gde će softver simulirati deo ili ceo proces.
- Potreban je zadovoljavajući nivo mrežne infrastrukture da bi se pojedini elementi kvalitetno sproveli.
- Potrebno je da dođe do promena u načinu rada i razmišljanja kod predavača i studenata.

Uprkos navedenim nedostacima web bazirane telelaboratorije imaju dominatnijih prednosti koje ih čine izuzetno važnim alatom u nastavnom procesu. Jasno je da će dalji razvoj informacione infrastrukture, kao i napredak u oblasti softverskog inženjerstva, elektronike i razvoja edukacijskih laboratorijskih *set-up*-ova doprineti bržem razvoju ovog koncepta laboratorija.

1.4 ZAHTEVI PRI KREIRANJU TELEEKSPERIMENTA

Uspešna implementacija sistema koji će imati zadatak da zameni klasično laboratorijsko okruženje sa telelaboratorijom ima više zahteva među kojima su najvažniji:

1. Zahtevi koji se tiču laboratorije:
   - lako razumevanje i korišćenje,
   - prilagođavanje tradicionalnih didaktičkih materijala novom kontekstu,
   - *on-line* supervizija laboratorijskog okruženja,
   - odgovarajući serveri i sistem zaštite,

- fleksibilan algoritam upravljanja sistemom,
- održavanje sistema i upravljanje izmenama,
- otvorena i modularna arhitektura,
- prilagodljivost fizičkog sistema i
- postojanje parametara kvaliteta servisa;

2. Zahtevi koji se tiču klijent okruženja:
   - multi-platformski klijent softver,
   - laka instalacija klijent softvera,
   - sigurnost klijenta i
   - besplatan klijent softver.
3. Zahtevi koji se tiču eksperimenta:
   - lak pristup biblioteci prethodnih algoritama ili prethodnih stanja,
   - čuvanje eksperimentalnih podataka i
   - *download* eksperimentalnih podataka.

Suštinski ovi zahtevi se mogu posmatrati iz tri ugla: zahtevi koji se tiču laboratorije u užem smislu (softvera koji upravlja laboratorijskom opremom i same opreme), zahteva koji se tiču klijent modula softvera i zahtevi koji se tiču eksperimenta u užem smislu.

Zahtevi koji se tiču laboratorije u užem smislu usmereni su ka tome da sam laboratorijski eksperiment, koji se odvija bez prisustva predavača, mora biti intuitivan i lak za korišćenje i opremljen potrebnim materijalom za razumevanje. Sem toga upravljanje samom laboratorijskom opremom mora biti pouzdano i dovoljno kvalitetno pri čemu relevantni parametri moraju ukazivati na potreban nivo servisa (brzina odziva sistema i slično). Što se tiče serverske softverske komponente, koja upravlja samim eksperimentom, ona mora biti fleksibilna, modularna i jednostavna za nadgledanje i održavanje.

Zahtevi koji se tiču klijenta usmereni su ka tome da se omogući korišćenje laboratorijskih vežbi nezavisno od hardverskog/softverskog sistema na kojem se nalazi korisnik. Ostali zahtevi se svode na laku instalaciju i sigurnost klijenta. Pošto je jedan od osnovnih zahteva da se sistemu pristupa preko Interneta, klasični web čitač (*browser)* ispunjava gotovo sve navede zahteve. Uz to web *brower*-i mogu biti besplatni za korišćenje.

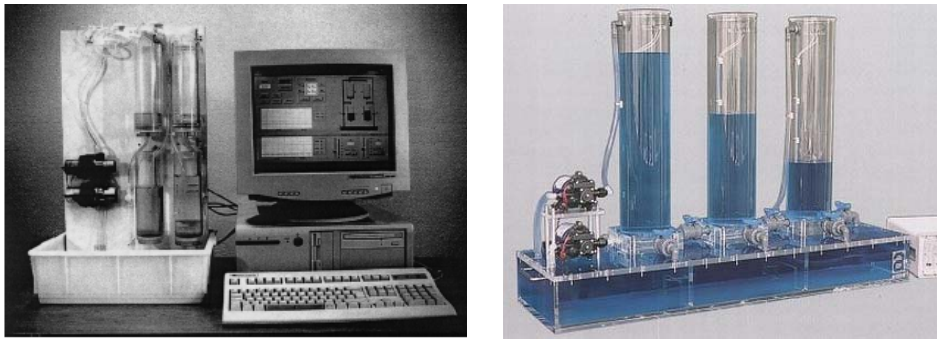
Poslednja grupa zahteva se tiče eksperimenta u užem smislu, odnosno rezultata eksperimenta. Potrebno je omogućiti da u slučaju da postoje zakoni upravljanja kao što je to slučaj kod vežbi čiji je predmet automatsko upravljanje, pojedini ili svi algoritmi upravljanja budu smešteni u posebnoj biblioteci kao i, eventualno, parametri koji karakterišu izabrana stanja sistema. Sem toga, potrebno je sa jedne strane obezbediti čuvanje eksperimentalnih podataka, a sa druge *download* eksperimentalnih rezultata za svakog pojedinačnog korisnika. Pojedinačni *download* treba da omogući kreiranje raznovrsnih tehničkih izveštaja, dok čuvanje eksperimentalnih podataka u sistemu omogućava predavačima i studentima da

porede različite eksperimentalne rezultate i da formiraju svojevrsnu bazu podataka koja se tiče ostvarenih eksperimenata.

Naravno, sem ovih navedenih zahteva, mogu postojeći i brojni drugi zahtevi koji bi bili karakteristični za različite uslove i situacije implementacije ovih laboratorija. Navedeni zahtevi možda najpotpunije opisuju zahteve koji se tiču inženjerskih eksperimenata i inženjerske edukacije dok bi kod nekih drugih naučnih i istraživačkih oblasti verovatno postojali i neki drugi specifični zahtevi.

2.
Laboratorijski modeli spregnutih rezervoara

Laboratorijski modeli spregnutih rezervoara su česti u inženjerskoj edukaciji zbog svog interdisciplinarnog sadržaja i edukativnog potencijala. U inženjerskoj praksi ovakvi sistemi se mogu naći u procesnoj industriji (pravljenje papira, postrojenja za tretman vode, petrohemijska industrija, itd.). Na slici 2 su prikazana neka rešenja celovitih eksperimentalnih instalacija u eksperimentalnoj primeni.



Slika 2. ***Primeri eksperimentalnih instalacija sa laboratorijskim modelom spregnutih rezervoara***

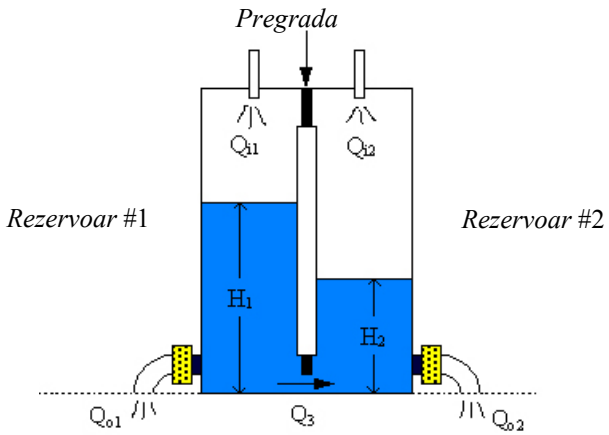
Moguća su različita rešenja oblika i sprezanja rezervoara, i to sa različitom ulogom pumpi i ventila u sistemu, tako da se dobijaju modeli različitih karakteristika. Na slici 3 su demonstrirana takva rešenja profesionalnih proizvođača laboratorijske opreme. Prokomentarišimo svrhu i specifičnosti različitih rešenja na slici3.

Rešenje na slici 3a) je proizvod firme KentRidge Instruments [6]. Sastoji se od dva mala izdužena rezervoara od pleksi-stakla okačena iznad velikog rezervoara (služi za skupljanje vode). Iz donjeg rezervoara se voda pumpa sa dve nezavisne pumpe na vrh svakog od dva izdužena rezervoara. Promena nivoa je vizuelno saglediva i može se očitati. Svaki od izduženih rezervoara, pri dnu ima ispust kroz koji može oticati voda, a koji je priključen na crevo koje odvodi vodu u sabirni rezervoar ispod. Protok vode koji se vraća u sabirni rezervoar je aproksimativno proporcionalan visini vode u rezervoaru iz koga voda ističe, pošto plastična cev na ispustu deluje kao pseudo-linearan hidraulički otpor.

a) b) c) d) e)

Slika 3. ***Ponuda profesionalnih proizvođača laboratorijske opreme***

Ako je potrebno, ovaj otpor može biti povećan zavrtanjem plastične slavine (ručnog ventila). Nivo vode u svakom rezervoaru je meren sa sondom kapacitativnog tipa koja je povezana sa elektronskim kolom u kutiji zaklonjenoj iza, iz koje se obezbeđuje priključak sa izlaznim signalom koji je proporcionalan nivou fluida u rezervoaru. Radi se od DC naponskom signalu u opsegu od 0 do 10 V. Nulti nivo signala se vezuje za tačku kada je rezervoar prazan (tj. nivo je oko 20 mm, v. 4), dok je rezervoar potpuno pun za nivo od 300 mm. Promenom pozicije pregrade između rezervoara, kao što je to pokazano na slici 4, se utiče na isticanje vode između dva rezervoara.



Slika 4. ***Funkcionalna šema laboratorijskog modela sa slike 3a)***

Matematički model dinamičkog sistema sa slike 4 je dat sledećim sistemom nelinearnih diferencijalnih jednačina:

$$\begin{aligned} A_1 \frac{dH_1}{dt} &= Q_{i1} - \alpha_1 \sqrt{H_1} - \alpha_3 \operatorname{sgn}(H_1 - H_2) \sqrt{|H_1 - H_2|} \\ A_2 \frac{dH_1}{dt} &= Q_{i2} - \alpha_2 \sqrt{H_2} - \alpha_3 \operatorname{sgn}(H_2 - H_1) \sqrt{|H_1 - H_2|} \end{aligned} \qquad (1)$$

gde su H_1 i H_2 visine izražene u milimetrima, Q_{i1} i Q_{i2} ulazni protoci koje obezbeđuje rad pumpi, A_1 i A_2 površine poprečnih preseka rezervoara i to su

$$A_1 = 3688.095\,\mathrm{mm}^2,\ A_2 = 3539.513\,\mathrm{mm}^2$$

dok su α_1, α_2 i α_3 koeficijenti isticanja, koji su takođe dati u katalogu proizvoda

$$\alpha_1 = 2020.70\,\mathrm{mm}^{5/2}/\mathrm{s},\ \alpha_2 = 2199.36\,\mathrm{mm}^{5/2}/\mathrm{s},\ \alpha_3 = 1144.30\,\mathrm{mm}^{5/2}/\mathrm{s}$$

Rad motora/pumpi opisuje sledeći linearni model

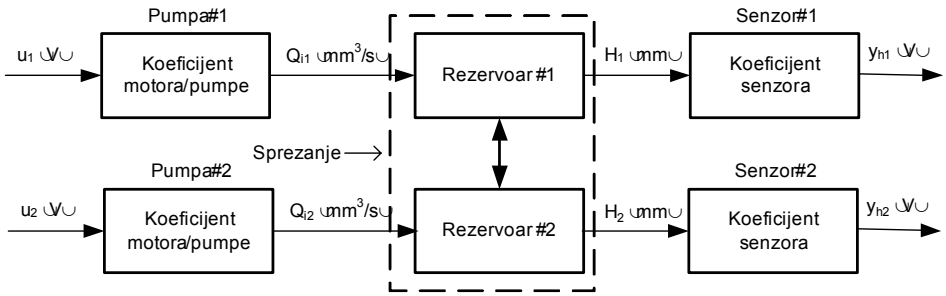
$$Q_{i1} = k_{i1}u_1 + c_{i1},\ Q_{i2} = k_{i2}u_2 + c_{i2}. \tag{2}$$

gde su Q_{i1} i Q_{i2} protoci pumpi, u_1 i u_2 su ulazni naponi DC motora pumpe, i $k_{i1} = 6760.8\,\mathrm{mm}^3/\mathrm{sV},\ k_{i2} = 6810.7\,\mathrm{mm}^3/\mathrm{sV},\ c_{i1} = 53\,\mathrm{mm}^3/\mathrm{s},\ c_{i2} = -1307.7\,\mathrm{mm}^3/\mathrm{s}.$

Senzori nivoa su takođe linearni i dati modelom

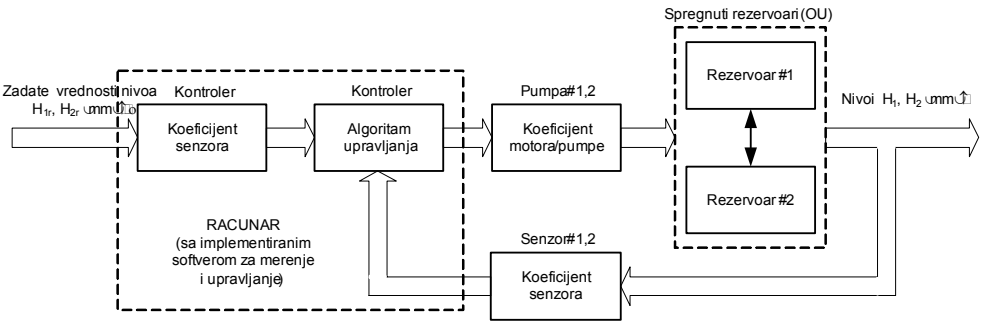
$$y_{h1} = k_{h1}H_1 + c_{h1},\ y_{h2} = k_{h2}H_2 + c_{h2}. \tag{3}$$

gde su H_1 i H_2 visine u [mm], y_{h1} i y_{h2} su signali senzora visine u [V], dok su $k_{h1} = 0.0406\,\mathrm{V/mm},\ k_{h2} = 0.04012\,\mathrm{V/mm},\ c_{h1} = 0.3568\,\mathrm{V},\ c_{h2} = 0.0378\,\mathrm{V}$ vrednosti koje su date u katalogu proizvođača laboratorijske opreme (slika 3a)).



Slika 5. ***Strukturni blok dijagram laboratorijskog modela sa slike 3a)***

Opšta struktura sistema upravljanja zasnovana na ovakvom laboratorijskom modelu je data na slici 6.



Slika 6. ***Strukturni blok dijagram eksperimentalne instalacije***

Primenjeni senzori i pumpe pokazuju najjednostavnije moguće dinamičko ponašanje (videti (2) i (3)), i korisniku eksperimenta su unapred raspoloživi podaci koji opisuju rad pumpi (2) i senzora (3). Pokretanje pumpi je u konkretnom slučaju predviđeno

da se realizuje upravljačkim PWM (širinski modulisanim) signalom, što je svakako važan podatak za samostalno pravljenje softvera za akviziciju podataka i upravljanje, ali ne i za projektovanje algoritma upravljanja. U postupku projektovanja proizvoljnog algoritma upravljanja, dinamika objekta upravljanja se svodi na dinamiku spregnutih rezervoara. Za male varijacije ulaznog protoka Q_{i1} i nivoa H_1 i H_2 ponašanje sistema je moguće opisati sledećim funkcijama prenosa

$$\frac{H_1(s)}{Q_{i1}(s)} = \frac{K}{Ts+1}, \quad \frac{H_2(s)}{Q_{i1}(s)} = \frac{K(s+\beta)}{s^2+2\zeta\omega_n s+\omega_n^2}. \tag{4}$$

tj. funkcijama prenosa prvog i drugog reda, čiji je oblik tipičan za teoretska razmatranja na kursevima iz teorije sistema i automatskog upravljanja [9, 10, 11].

Slika 7. ***Proizvod firme Quanser***

Šta bi se dogodilo, ako bi strukturi na slici 4 bio dodat i treći, četvrti ili peti rezervoar u započetom nizu?

Proces koji je ograničen na jedan rezervoar se može aproksimirati tzv. sistemom prvog reda. Pridodavanjem rezervoara kao na slici 4, red sistema raste za jedan po pridodatom rezervoaru. Rad sistema postaje složeniji funkcijom drugih pumpi čiji rad može biti posmatran kao poremećaj ili kao upravljački signal multivarijabilnog sistema. Razumevajući suštinu fizičkih analogija, odnosno da se rad mnogih procesa može modelirati tipičnim modelima prvog, drugog i višeg reda, edukativni kapacitet vežbe sa rezervoarima, gde su procesi očigledni, postaje jasan.

Razmotrimo sada specifičnosti laboratorijskog modela na slici 3d). Isti model kao na slici 3d), ali od drugog proizvođača laboratorijske opreme je dat i na slici 7. Osnovna razlika u odnosu na laboratorijski model sa slike 3a) je u načinu sprezanja rezervoara. Pumpa iz sabirnog rezervoara može da pumpa vodu u rezervoare iznad, a isticanjem iz gornjeg rezervoara #1 se puni donji rezervoar #2. Ručnim ventilima se reguliše da li će pumpa pumpati vodi samo u gornji rezervoar ili u oba, kao i protok isticanja iz oba rezervoara. Pumpa koja se koristi je zupčasta pumpa koja se sastoji od 12 V DC motora koji okreće radijalno postavljene lopatice. Dakle, upravljački signal je jednosmerni naponski signal, opsega od 0 V do 12 V. Merenje nivoa u svakom rezervoaru se realizuje putem davača pritiska na dnu rezervoara, koji proporcionalno visini vode u rezervoaru na svom izlazu daje naponski signal u opsegu od 0 V do 5 V. Visina rezervoara #1 i rezervoara #2 je 30 cm, njihov unutrašnji prečnik je 4.445 cm (a površina poprečnog preseka rezervoara je 15.5179 cm^2), osetljivost senzora je 5 cm/V, a konstanta pumpe je K_m = 4.6 (cm^3/s)/V. Ručni ventili koji regulišu isticanje iz rezervoara #1 i rezervoara #2 mogu biti podešeni u pozicije da je površina otvora isticanja 0.079173 cm^2 (mala), 0.17813919 cm^2 (srednja), ili 0.242467 cm^2 (velika). Kada pumpa uliva vodu samo u gornji rezervoar #1, upravljački cilj može biti regulacija nivoa H_1 u rezervoaru #1, ili nivoa H_2 u

donjem rezervoaru #2. Za male varijacije ulaznog protoka i nivoa H_1 i H_2 ponašanje sistema je moguće opisati funkcijama prenosa sledećeg oblika

$$\frac{H_1(s)}{U(s)} = \frac{K_1}{T_1 s + 1}, \quad \frac{H_2(s)}{H_1(s)} = \frac{K_2}{T_2 s + 1}, \quad \frac{H_2(s)}{U(s)} = \frac{K_1}{T_1 s + 1} \frac{K_2}{T_2 s + 1}. \tag{5}$$

gde je $u(t)$ naponski ulaz pumpe, i ujedno ulaz sistema. Svaka sledeći rezervoar u kome se događa akumulacija tečnosti povećava red sistema za po jedan.

Gore navedeni parametri laboratorijskih modela sa slika 3a) i 3d) mogu poslužiti studentima kao osnova za vežbu u smislu simulacije rada sistema.

Razliku između laboratorijskih modela 3d) i 3e) čini samo dugo crevo koje povezuje pumpu i gornji rezervoar #1. Na ovaj način se uvodi značajno vremensko kašnjenje u sistem. Ako je crevo dužine L i površine poprečnog preseka A, pumpa produkuje protok $Q(t)$ i τ vreme koje je potrebno da fluid pređe put sa početka na kraj creva, nabrojane veličine povezuje sledeća relacija

$$L = \int_0^{\tau} \frac{Q(t)}{A} dt \tag{6}$$

a u slučaju konstantnog protoka $Q(t) = Q$, moguće je jednostavno izračunati čisto vreme kašnjenja koje se unosi u sistem transportom fluida

$$\tau = \frac{LA}{Q} \tag{7}$$

Međutim, protok pumpe je upravljačka veličina i menja se vremenom, a prema tome čisto vremensko kašnjenje sistema $\tau(t)$. Ako je $Q_p(t)$ protok pumpe, $Q_{in}(t)$ ulazni protok rezervoara, ove dve veličine su povezane sledećom relacijom

$$Q_{in}(t) = Q_p(t - \tau(t)), \quad Q_{in}(s) = Q_p(s) e^{-s\tau} \; (\tau \neq const.). \tag{8}$$

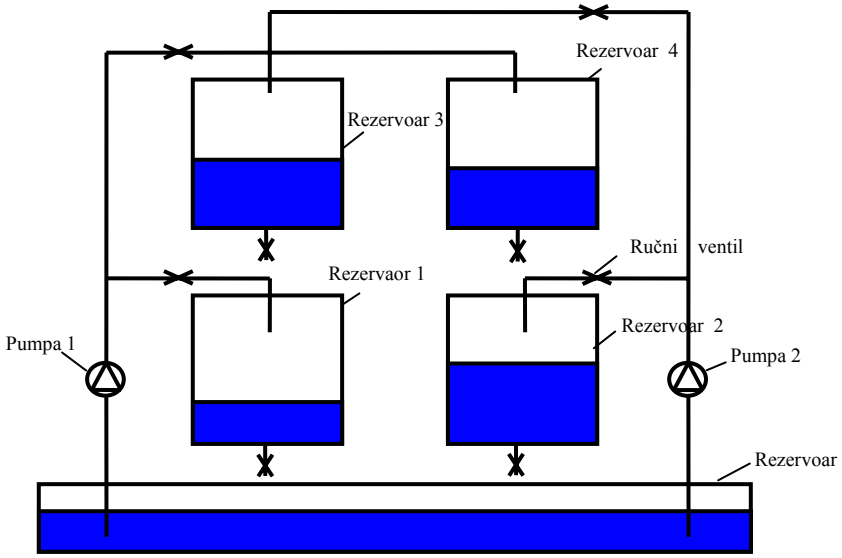
Upravljanje sistemima sa značajnim, i to još promenljivim, čistim vremenskim kašnjenjem τ je poseban izazov.

Laboratorijski model na slici 3b) u odnosu na model sa slike 3d), unosi dodatne nelinearnosti u sistem, i približava model realnim primerima iz inženjerske prakse. Površina poprečnog preseka rezervoara je promenljiva, te jednačina akumulacije tečnosti u rezervoaru postaje složenija

$$\frac{dV(t)}{dt} = \frac{dA(h(t))h(t)}{dt} = Q_{in}(t) - Q_{out}(h(t)) \tag{9}$$

Promena zapremine tečnosti u rezervoaru jednaka je razlici ulaznog i izlaznog protoka. Naravno, funkcija površine poprečnog preseka rezervoara $A(h)$ je poznata.

Laboratorijski model sa slike 3c) je najsloženiji, može biti posmatran kao proširenje laboratorijskog modela sa slike 3d) i bliže je objašnjen na slici 8.



Slika 8. ***Struktura laboratorijskog modela sa slike 3c)***

Ručni ventili, obeleženi na slici 8 imaju zadatak da prilikom prepodešavanja promene parametre sistema, ili čak i njegov način funkcionisanja. Pumpe upravljane jednosmernim naponskim signalima pumpaju vodu iz sabirnog rezervoara u četiri gornja rezervoara, pri čemu voda iz gornja dva rezervoara otiče u rezervoare ispod. Svaka od pumpi utiče na nivoe vode u donjim rezervoarima, a nivo ove interakcije može biti podešavan ručnim ventilima. Ulazni protok od rezervoara iznad u donje rezervoare može biti posmatran kao poremećaj. Model ovakvog sistema može biti predstavljen sledećim sistemom nelinearnih diferencijalnih jednačina

$$\begin{aligned}
\frac{dH_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gH_1} + \frac{a_3}{A_1}\sqrt{2gH_3} + \frac{\gamma_1 Q_{p1}}{A_1} \\
\frac{dH_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gH_2} + \frac{a_4}{A_2}\sqrt{2gH_4} + \frac{\gamma_2 Q_{p2}}{A_2} \\
\frac{dH_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gH_3} + \frac{(1-\gamma_2) Q_{p2}}{A_3} \\
\frac{dH_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gH_4} + \frac{(1-\gamma_1) Q_{p1}}{A_4}
\end{aligned} \qquad (10)$$

gde su H_i – visina nivoa tečnosti u i-tom rezervoaru, A_i – površina poprečnog preseka i-tog rezervoara, a_i – koeficijenti proporcionalni površini otvora za isticanje tečnosti i-tog rezervoara, Q_{pj} – protok j-te pumpe, a koeficijenat γ_j odgovara delu protoka koji odlazi donjem rezervoaru od j-te pumpe (shodno Bernulijevoj jednačini). Linearizacijom (10), dobija se

$$\frac{d\mathbf{x}}{dt}=\begin{bmatrix} -\frac{1}{T_1} & 0 & 0 & 0 \\ 0 & -\frac{1}{T_2} & 0 & 0 \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix}\mathbf{x}+\begin{bmatrix} \frac{\gamma_1}{A_1} & 0 \\ 0 & \frac{\gamma_2}{A_2} \\ 0 & \frac{1-\gamma_2}{A_2} \\ \frac{1-\gamma_2}{A_2} & 0 \end{bmatrix}\mathbf{u};\quad \mathbf{x}=\begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{bmatrix}, \mathbf{u}=\begin{bmatrix} Q_{p1} \\ Q_{p2} \end{bmatrix}. \qquad (11)$$

gde je $T_i=\frac{A_i}{a_i}\sqrt{\frac{2H_i(0)}{g}}$. Linearizacija je izvršena u okolini stanja $H_i(0)$. Procedura modeliranja i analiza na osnovu modela biće detaljnije objašnjena u delovima praktikuma koji slede (poglavlje 2.3).

2.1. AKTUELNI LABORATORIJSKI MODEL I EKSPERIMENTALNA APARATURA

Na Mašinskom fakultetu u Kragujevcu se koristi laboratorijski model spregnutih rezervoara koji je proizvela firma „Majk Elektronik" iz Mladenovca realizujući nešto fleksibilniji koncept od objašnjenog na slici 8. Napravljena je fleksibilna eksperimentalna aparatura modularnog tipa koja može imati karakteristike svakog od fizičkih modela sa slike 3 (izuzev modela sa slike 3b)). Ostavljena je i mogućnost da izlazni protok rezervoara: 1) ne bude kontrolisan, 2) bude kotrolisan ručnim ventilima, 3) bude kontrolisan električnim on-off ventilima. Ipak, sliku 8 možemo prihvatiti kao funkcionalnu šemu spregnutih rezervoara i u ovom slučaju.

Slika 9. ***Eksperimentalna aparatura***

Eksperimentalna aparatura, prikazana na slici 9, je postavljena u prostoriji C85 Centra za primenjenu automatiku Mašinskog fakulteta u Kragujevcu, koja treba da čini integralni deo WEB Laboratorije Univerziteta u Kragujevcu, a koja je trenutno u proceduri formalnog osnivanja (www.weblab.kg.ac.yu). Modularna struktura laboratorijskog modela sa slike 9 odgovara slučaju kada izlazni protok spregnutih rezervoara nije kontrolisan, a dodatno transportno kašnjenje nije uvedeno (kao što je to moglo biti učinjeno kao na slici 3d))

Aparatura se sastoji od laboratorijskog modela spregnutih rezervoara, AD/DA interfejsa, web kamere i PC računara.

Svaki od rezervoara je unutrašnjih dimenzija 40x40x300 mm. Davač nivoa H je zapravo davač hidrostatičkog pritiska u rezervoaru ($p=\rho gH$) koji na svom izlazu daje naponski signal u opsegu od (1-5) V. Deo studentskog zadatka je crtanje statičke karakteristike i kalibracija ovog davača, iako se po specifikaciji radi o mernom pretvaraču linearnog tipa koeficijenta pojačanja k_h =0.1 V/cm. Sabirni rezervoar je dimenzija 14x37x10 cm i može da prihvati zapreminu od 5 l tečnosti.

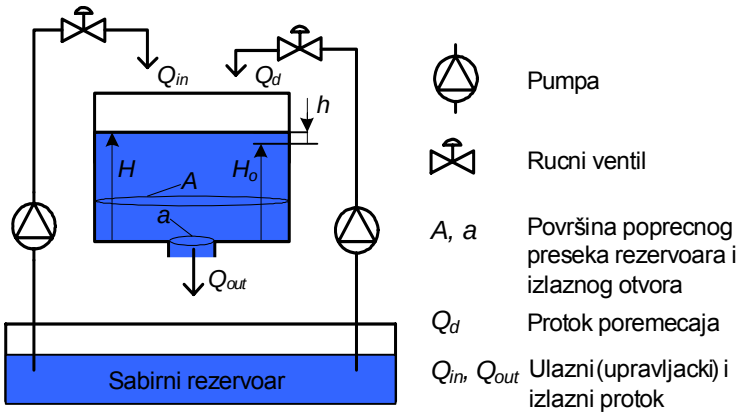
Korišćene pumpe su identične, ulaznog jednosmernog napona u opsegu od 0 V do 12 V, i izlaznog protoka od 0 do 20 cm^3/s. Pumpe su centrifugalnog tipa i prema nominalnoj specifikaciji su nelinearne, pa je deo studentskog zadatka i crtanje statičke karatkeristike i kalibracija pumpi.

Slika 10. ***Korišćeni modul za akviziciju signala i upravljanje NI USB 6009***

Korišćeni AD/DA interfejs, tj. modul za akviziciju signala i upravljanje je prikazan na slici 10. DA sekcija NI USB 6009 ima 2 kanala. D/A konvertori su 12-to bitni opsega od 0 V do 5 V. AD sekcija NI USB 6009 ima 8 kanala, odnosno 4 diferencijalnog tipa. A/D konvertori su 14-bitni. Diferencijalana veza ulaza omogućava merenje signala u opsegu od ±20 V, a asimetrična u opsegu od ±10 V. Detaljniji opis AD/DA interfejsa je dat u [36].

2.2. MODELIRANJE SISTEMA SA JEDNIM REZERVOAROM

Principi modeliranja sistema (laboratorijskog modela) sa jednim rezervoarom i sa spregnutim rezervoarima su isti, tako da će ovde, jednostavnosti radi, biti objašnjeno modeliranje sistema sa jednim rezervoarom. Primer pokazan na slici 11 je dovoljno reprezentativan, i lako proširiv tzv. složenije slučajeve.



Slika 11. Funkcionalna šema sistema sa jednim rezervoarom

Model dinamičkog ponašanja sistema proističe iz jednačine kontinuiteta ili zakona o održanju mase. Promena zapremine u rezervoaru odgovara razlici ulaznog i izlaznog protoka

$$\frac{dV}{dt} = A\frac{dH}{dt} = \left(Q_{in} + Q_d\right) - Q_{out} \tag{12}$$

Ulazni protoci Q_{in} i Q_d zavise od pumpi koje pobuđuje upravljački jednosmerni napon ($Q_{in} = f\left(u_{in}\right), Q_d = f\left(u_d\right)$), i čiji rad u najjednostavnijem slučaju može biti modeliran linearnom funkcijom $Q_{in} = k_m u_{in}$, $Q_d = k_m u_{in}$. U slučaju kada bi se radilo o sistemu spregnutih rezervoara, protok Q_d, koji u slučaju slike 11 znači planski postavljen poremećaj u okviru eksperimenta, bi bio protok koji se dobija isticanjem iz drugih rezervoara. Pošto nećemo u napred prihvatiti da su naše pumpe linearne sa poznatim koeficijentom linearnosti, generisani uticaji na objekat upravljanja neće biti naponski signali koji utiču na pumpe, već jednostavno protoci tih pumpi.

Izlazni protok Q_{out} zavisi od visine nivoa tečnosti u rezervoaru *H*, od površine poprečnog preseka izlaznog otvora *a*, razlike pritisaka iznad nivoa tečnosti i ispod otvora oticanja (izlazne cevi), i gubitaka energije pri isticanju tečnosti kroz male otvore (koeficijenta protoka μ). Na osnovu Bernulijeve jednačine (važi za neviskozni i nestišljivi fluid u stacionarnom strujanju), i situacije na slici 11 se dobija brzina isticanja tečnosti

$$p_a + \rho g H = p_a + \frac{\rho v^2}{2} \;\Rightarrow\; v = \sqrt{2gH} \tag{13}$$

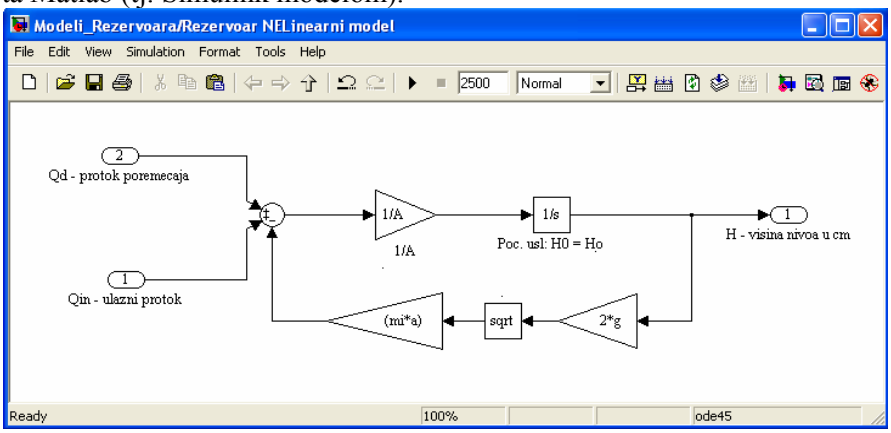
Izlazni protok Q_{out} je proizvod površine poprečnog preseka otvora (a) i brzine isticanja (13), korigovane koeficijentom protoka (μ) usled lokalnih gubitaka

$$Q_{out} = \mu a\sqrt{2gH} \tag{14}$$

Model punjenja rezervoara, prema slici 12 je dat sledećom nelinearnom diferencijalnom jednačinom

$$A\frac{dH}{dt} = (Q_{in} + Q_d) - \mu a\sqrt{2gH} \Rightarrow \frac{dH}{dt} = \frac{(Q_{in} + Q_d)}{A} - \frac{\mu a\sqrt{2g}}{A}\sqrt{H} \qquad (15)$$

Jednačina (15) predstavlja matematički model sistema sa slike 11, koji je na slici 12 predstavljen simulacionim blok dijagramom u modulu Simulink programskog paketa Matlab (tj. Simulink modelom).



Slika 12. ***Simulink model rezervoara***

Vezujući se za oznake sa slike 11 i parametre realizovanih rezervoara sa slike 9, oslanjajući se na numeričke vrednosti sistematizivane u tabeli 1, teorijska razmatranja će biti delom propraćena primerima simlacije u programskom paketu Matlabu.

Naziv	*Oznaka*	*Vrednost*	*Jedinica*
Maksimalna visina	H_{max}	30	cm
Radna tačka	H_0	50% H_{max} = 15	cm
Površina poprečnog preseka rezervoara	A	16	cm^2
Površina poprečnog preseka izlaznog otvora	a	0.16	cm^2
Koeficijent protoka	μ	0.635	-

Tabela 1. ***Parametri objekta upravljanja*** **(*v.* slike 11)**

Rad sa Simulink modelom sa slike 12 je moguć tek po učitavanju potrebnih podataka u radnom prostoru (w*orkspace*) programskog paketa Matlab. To se može učiniti startovanjem programa *Podaci.m* čiji listing sledi. Detaljnije o korišćenju programskog paketa Matlab moguće je saznati iz [15, 16].

```
%-------------------------------------------------
%Polazni podaci (Fajl podaci.m)
%-------------------------------------------------
mi = 0.635; A = 16; a = 0.16;
Hmax = 30; Ho = 15; g = 981;
```

Model dat jednačinom (15) je nelinearan. Generalno, sa stanovišta i analize i sinteze sistema je poželjno raspolagati linearnim modelima (zbog korišćenja mogućnosti koje nudi vrlo dobro sistematizovana teorija linearnih sistema). Zato se pribegava postupku linearizacije nelinearnog modela u okolini radne tačke. Radna tačka je definisana nominalnim radnim uslovima (uslovima okruženja, koje u ovom slučaju definiše nominalna vrednost poremećaja $Q_d(t) = Q_{d0} \equiv 0$) i nominalnim vrednostima upravljanja (u našem slučaju SISO sistema, $Q_{in}(t) = Q_{in0}$) [10]. Naravno, nominalne vrednosti uticajnih veličina (upravljanja i poremećaja) imaju za rezultat nominalni odziv sistema (u našem slučaju SISO sistema, $H(t) = H_0$). Nominalne veličine (koje definišu radnu tačku (Q_{in0}, Q_{d0}, H_0)) imaju karakter stacionarnih veličina, te na osnovu matematičkog modela (15), možemo uspostaviti vezu

$$Q_{in0} = \mu a\sqrt{2g}\sqrt{H_0} \;\Leftrightarrow\; H_0 = \frac{Q_{in0}^2}{2g\mu^2 a^2}. \tag{16}$$

uz uvažavanje pretpostavke da je $Q_{d0} \equiv 0$.

Radno područije, odnosno kretanje sistema u okolini radne tačke (Q_{in0}, Q_{d0}, H_0), $Q_{d0} \equiv 0$, shodno slici 11, je definisano sledećim opsegom relevantnih signala

$$Q_{in}(t) = Q_{in0} + q_{in}(t),\; Q_d(t) = Q_{d0} + q_d(t), H(t) = H_0 + h(t). \tag{17}$$

Sada, podsećanja radi, navedimo i relaciju

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \ldots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$$

koja predstavlja razvoj funkcije $f(x)$ u Tejlorov red.

Zamenom (17) u (15), te razvojem u Tejlorov red u odnosu na definisanu radnu tačku (Q_{in0}, Q_{d0}, H_0), $Q_{d0} \equiv 0$, uvažavanjem relacije (16) i zanemarivanjem viših članova reda, dobija se

$$\frac{dH}{dt} = \frac{1}{A}(Q_{in} - Q_{in0}) + \frac{1}{A}(Q_d - Q_{d0}) - \frac{\mu a}{A}\sqrt{\frac{g}{2H_0}}(H - H_0) \tag{18}$$

ili, na osnovu relacije (17), dobija se sledeći oblik linearizovanog modela

$$\frac{dh(t)}{dt} = \frac{1}{A}q_{in}(t) + \frac{1}{A}q_d(t) - \frac{\mu a}{A}\sqrt{\frac{g}{2H_0}}h(t) \tag{19}$$

Relaciju (19) je dalje pogodno transformisati u

$$T_l\frac{dh(t)}{dt} + h(t) = K_l q_{in}(t) + K_l q_d(t), \quad T_l = \frac{A}{\mu a}\sqrt{\frac{2H_0}{g}}, K_l = \frac{1}{\mu a}\sqrt{\frac{2H_0}{g}}. \tag{20}$$

Primenom Laplasove transformacije nad relacijom (20) dobijamo model u sledećoj formi

$$h(s) = \begin{bmatrix} \dfrac{K_l}{T_l s + 1} & \dfrac{K_l}{T_l s + 1} \end{bmatrix} \begin{Bmatrix} q_{in}(s) \\ q_d(s) \end{Bmatrix} \tag{21}$$

gde matrica funkcija prenosa povezuje kompleksne likove ulaznih i izlaznih promenljivih.

Dakle, shodno strukturi sistema na slici 11, matematički model sistema je dat relacijom (15). S druge strane, mogućnost primene linearne teorije leži u pretpostavci da sistem (u nominalnom režimu rada) funkcioniše u uskom radnom području unutar koga je moguća aproksimacija njegovog ponašanja linearnim modelom [10]. Linearni model sistema predstavljaju skupa relacije (17) i (20), ili (17) i (21). Ovde ćemo demonstrirati mogućnosti i ograničenja linearnog modela u odnosu na originalni nelinearni model.

Važna karakteristika sistema je njegova statička karakteristika. Statičku karakteristiku sistema čini geometrijsko mesto tačaka koje opisuju vrednosti ulaza i izlaza sistema u stacionarnom stanju (prelazni procesi koji prate promene stacionarnih stanja se zanemaruju, a razmatraju se samo promene stacionarnih stanja). U konkretnom slučaju sistema, karakter uticaja poremećaja Q_d je isti kao i karakter ulaza Q_{in} u sistem, i nepotrebno je pratiti skup stacionarnih stanja tipa (Q_{in}, Q_d, H). Skup stacionarnih stanja tipa (Q_{in}, H) daje statičku karakteristiku koja je sasvim adekvatna za analizu posmatranog sistema.

Ako u jednačini (15)

$$\frac{dH(t)}{dt} = \frac{\left(Q_{in}(t) + Q_d(t)\right)}{A} - \frac{\mu a\sqrt{2g}}{A}\sqrt{H(t)}$$

pretpostavimo da je $Q_d(t) \equiv 0$ i $dH(t)/dt \equiv 0$, dobijamo statičku karakteristiku

$$H = \frac{Q_{in}^{\ 2}}{2g\mu^2 a^2}. \tag{22}$$

Ako bi sada respektovali linearni model ((20) i (17)), uzimajući da je $dh(t)/dt \equiv 0$ i $q_d(t) \equiv 0$, iz (20) se dobija

$$h = K_l q_{in} \ , K_l = \frac{1}{\mu a}\sqrt{\frac{2H_0}{g}}. \tag{23}$$

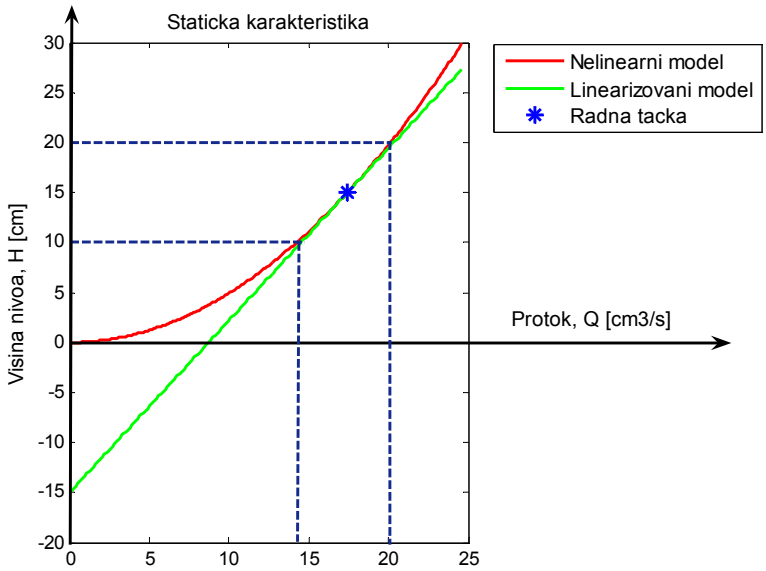
a na osnovu (17) se dolazi do statičke karakteristike sistema dobijene na osnovu linearizovanog modela

$$H = H_0 + K_l\left(Q_{in} - Q_{in0}\right). \tag{24}$$

Slaganje statičkih karakteristika (22) i (24), kao i modela (15) i ((20) i (17)) ilustrovaćemo konkretnim rezultatom simulacije. Na osnovu vrednosti u tabeli 1 i izraza (16), izračunavamo da je za stacionarnu vrednost visine nivoa H_0= 15 cm potrebna stacionarna vrednost ulaznog protoka $Q_{in0} = 17.43\ \mathrm{cm}^3/\mathrm{s}$. Po aktiviranju gore pokazanog programa *Podaci.m* u programskom paketu Matlab i sledećih linija koda

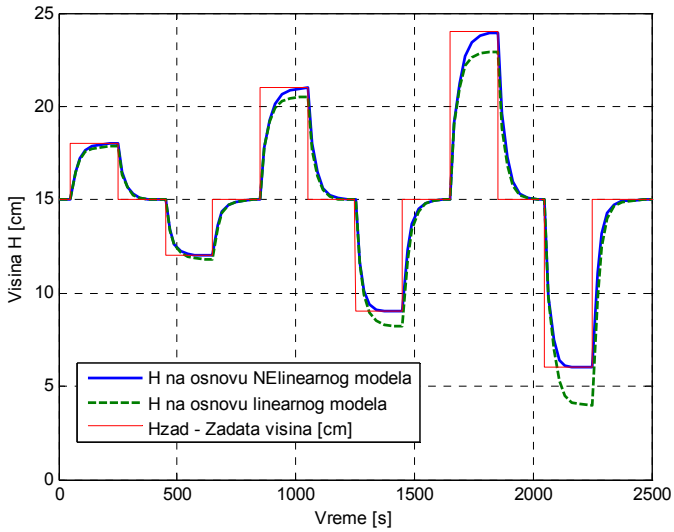
```
%-------------------------------------------------
%Program za crtanje statičke karakteristike sistema
%-------------------------------------------------
% Definisanje radne tačke (Qo,Ho)
Ho = 15; Qo=mi*a*sqrt(2*g)*sqrt(Ho);

Qmax=mi*a*sqrt(2*g)*sqrt(Hmax); % max stac. protok
Q=0:0.1:Qmax;   % Vrednosti protoka, definisanje niza

% Visina nivoa, na osnovu NELINEARNOG modela
HNL=(Q.^2)/(2*g*(mi^2)*(a^2));

% Visina nivoa, na osnovu LINEARNOG modela
Kl=sqrt(2*Ho/g)/(mi*a);
HL=Ho+Kl*(Q-Qo);

figure (1)
plot(Q,HNL,'r',Q,HL,'g',Qo,Ho,'*');
xlabel('Protok, Q [cm3/s]');
ylabel('Visina nivoa, H [cm]');
title('Staticka karakteristika');
grid;
legend('Nelinearni model','Linearizovani model','Radna tacka',-1);
```

dobijamo kao rezultat na slici 13.

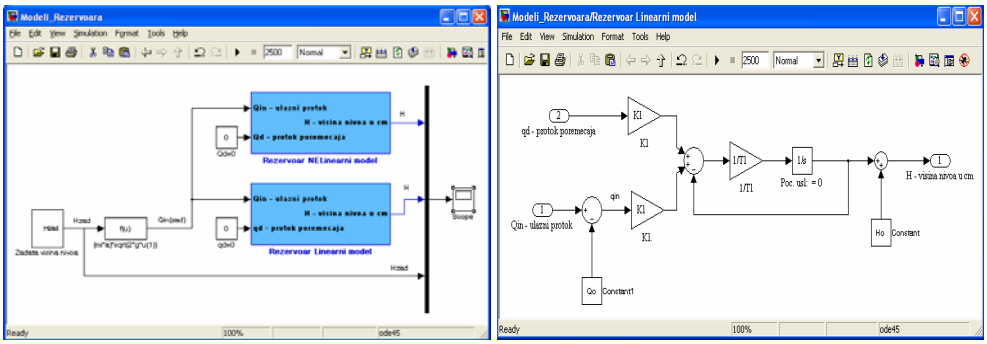


Slika 13. ***Statička karakteristika***

Slaganje statičkih karakteristika nelinearnog (realnog) i linearnog (aproksimativnog) modela je sasvim dobro u okolini radne tačke (±5 cm od H_0=15 cm, prema slici 13). Međutim, treba zapaziti da je prema statičkoj karakteristici dobijenoj na osnovu linearizovanog modela moguće dobiti i apsurdne rezultate (negativnu visinu nivoa za pozitivne vrednosti ulaznog protoka, v. sliku 13).



Slika 14. ***Poređenje odziva linearnog i nelinearnog modela u odnosu na zadatu vrednost nivoa „ feedforwad“ upravljanog sistema***



Slika 15. ***Simulink modeli: a) procesa verifikacije linearnog modela rezervoara, b) linearnog modela rezervoara, c) na slici 12, nelinearnog modela rezervoara***

Ograničene mogućnosti korišćenja linearnog modela nelinearnog objekta upravljanja su ilustrovane i na slici 14. Simulacija je izvršena kreiranjem Simulink modela datih na slici 15. Udaljavanjem od radne tačke efekti greške modeliranja linearnog modela su uočljiviji (slika 14).

3.
Edukativni ciljevi i zadaci laboratorsijske vežbe

Za inženjere različitih profila je jako važno da razumeju kako funkcioniše sistem spregnutih rezervoara, kako primeniti tehnike merenja i odgovarajuće hardverske komponente, i kako upravljački problem može biti rešavan i implementiran kroz pisanje upravljačkog softvera. Problemi rezervoara, regulacije protoka i nivoa, su česti: u sistemima vodosnabdevanja i kanalizacije, biosistemima (neki problemi medicinske fiziologije), u brojnim sistemima procesne industrije. Recimo, u procesnoj industriji (industriji pravljenja papira, petro-hemijskoj industriji, itd.) problemi transporta fluida, skladištenja i čuvanja fluida u rezervoaru, i pretakanja u drugi rezervoar su čest slučaj. Zatim, problemi hemijskog tretmana ili mešanja fluida u rezervoarima su takođe vrlo česti. U svim tim slučajevima, nivo fluida u rezervoarima mora biti kontrolisan, kao i protok između rezervoara. Često su rezervoari tako spregnuti tako da su njihovi nivoi u međusobnoj interakciji, i da moraju biti skupa kontrolisani. Upravljanje nivoa i protoka u rezervoarima je jedno od centralnih pitanja hemijskih inženjerskih sistema.

Laboratorijska aparatura predstavljena na slikama 9 i 11 omogućava da širok krug eksperimenata različitog nivoa složenosti može biti sproveden. Procesi koji se dešavaju su sami po sebi očigledni i lako shvatljivi i za neukog posmatrača. Očigledan je i mogući cilj: upotrebiti neku od upravljačkih tehnika da bi se došlo do kontrolera za regulaciju visine nivoa tečnosti u više spregnutih rezervoara i/ili ostvariti da kontrolisani nivoi prate zadate trajektorije. Funkcionalne celine upotrebljenog hardvera su takođe sasvim jasne.

Opšti ciljevi vežbe su da studenti

- budu upoznati fenomenologijom sistema sa povratnom spregom – ključnim idejama teorije sistema, komponentama sistema i terminologijom
- steknu iskustvo u kalibraciji senzora, izvođenju merenja i primene za to odgovarajućeg softvera
- testiraju stečena znanja iz oblasti modeliranja i identifikacije procesa

- analiziraju statičke i dinamičke osobine procesa i sistema sa povratnom spregom, i ocene kvalitet i glavne parametre sistema upravljanja
- demonstriraju performanse i ograničenja sistema sa otvorenom povratnom spregom (ručno upravljanje) kada postoji potreba za promenom zadate vrednosti ili otklanjanju efekata dejstva poremećaja
- demonstriraju nestabilan rad sistema sa povratnom spregom, i to kao uticaj pojave čistog vremenskog kašnjenja u sistemu (uvođenje transportnog kašnjenja je ilustrovano na slici 3e)
- demonstriraju uticaj promene pojačanja (P regulatora) na dinamičke i statičke osobine sistema
- demonstriraju uticaj uvođenja integralnog dejstva (transformacija P regulatora u PI) u smislu eliminacije greške stacionarnog stanja sistema upravljanja sa povratnom spregom
- steknu iskustvo u sintezi i podešavanju različitih zakona upravljanja (uobičajene metode: PI+Feedforward upravljanje, PID algoritmi, metoda podešavanja polova, MPC algoritmi, povratna sprega po stanju, itd.)
- steknu iskustvo u implementaciji algoritama upravljanja kroz pisanje odgovarajućeg softvera za rad sistema u realnom vremenu
- steknu iskustva u primeni procesnih računara (PLC, HMI, itd) i pripadajućeg softvera za implementaciju određenih upravljačkih strategija.

S obzirom da je predviđeno izvršenje vežbe putem Interneta, mogući su i drugi pridruženi studentski zadaci sa aspekta kreiranja funkcija nadzorno upravljačkih sistema zasnovanih na Internet tehnologijama (pisanje softvera, kreiranje baze podataka, funkcije video nadzora, čuvanja, slanja i prikaza podataka, itd.). Dakle, hardverske mogućnosti kompletne laboratorijske vežbe omogućavaju izradu zadataka osmišljenih da student stiče i unapređuje veštine programiranja i znanja iz drugih referentnih oblasti.

U poglavlju koje tretira principe fazi upravljanja, posle teorijskih uvoda i pokazanih primera, u kontekstu razmatrane teorije i datog laboratorijskog modela su predloženi zadaci i ciljevi laboratorijske vežbe.

3.1. PRIMER PROJEKTNOG ZADATKA

Proteklih godina je na Mašinskom fakultetu u Kragujevcu, u sklopu različitih studijskih profila je bio izdavan projektni zadatak sledećeg naslova i sadržine

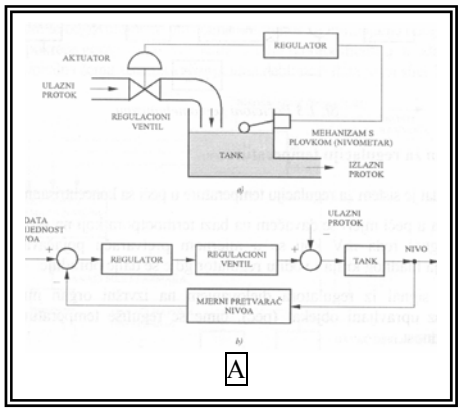
RAČUNARSKA PODRŠKA PRI SINTEZI KONTROLERA I SIMULACIJI RADA SISTEMA

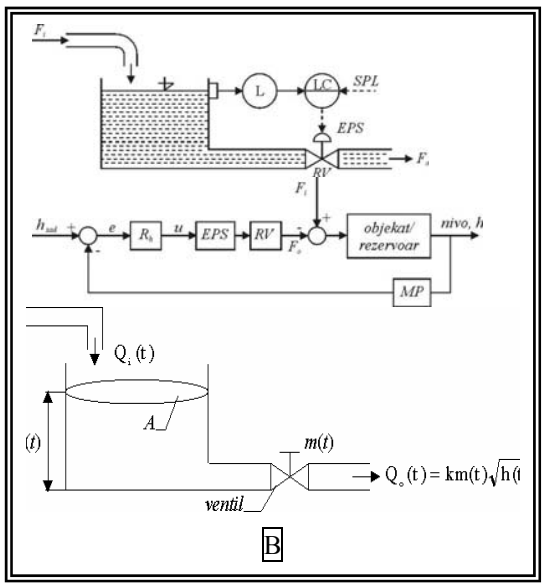
P R O J E K T N I Z A D A T A K

Slika 16. ***Standardne eksperimentalne aparature u laboratorijama širom sveta***

Proces punjenja rezervoara i isticanja tečnosti je nelinearan, jednostavan za modeliranje, čest u tehničkoj praksi, i ilustrativan u pogledu inženjerske edukacije (modeliranje dinamičkih sistema, regulacija nivoa, primena linearnog zakona upravljanja na nelinearnom objektu, itd.). Pri tome je moguće osmisliti različite koncepte regulacije nivoa.

Primer:





Date su dve koncepcije regulacije nivoa: A i B (a mogu biti osmišljene i druge). Primer je prepoznatljiv sa predavanja i obrađivan je kroz predložene udžbenike i drugu literaturu. Studenti se ohrabruju da sami predlože koncept regulacije tečnosti u rezervoaru.

Slede zahtevi zadatka:

Objekat upravljanja (OU) je rezervoar sa tečnošću. Izlazna promenljiva je visina nivoa tečnosti u rezervoaru. Na nivo tečnosti u rezervoaru utiču ulazni protok i protok isticanja. Data je sloboda izbora samog koncepta funkcionisanja sistema, odnosno izbora upravljačke promenljive i poremećaja. Praktično treba dati rešenje za regulaciju nivoa tečnosti u rezervoaru, i simulacijom na digitalnom računaru verifikovati kvalitet tog rešenja. Dozvoljena su sva pojednostavljenja elemenata objekta u širem smislu, izuzev samog procesa akumulacije vode u rezervoaru.

1. Struktura sistema – *Definisati sistem u celosti, opisati ga i dati njegov kompletan matematički model*. Ova tačka sadrži: Funkcionalnu šemu sistema i opis rada. Detaljan i realan (nelinearni) matematički model OU u širem smislu. Statičku karakteristiku OU. Linearizovani matematički model OU u okolini radne tačke (radna tačka, definisana sa x% od max zadate visine tečnosti u rezervoaru, je različita za svakog studenta). Strukturni blok dijagram sistema i u kontinualnoj i u digitalnoj verziji. Proveru validnosti linearizovanog modela (u programskom paketu Matlab, programskom modulu Simulink, napraviti nelinearan (realan) i linearizovani model (i kontinualni i diskretni). Isti ulaz, stepenastog tipa (sekvanca pravougaonih signala dovoljne širine da prelazni proces uvek bude doveden do kraja (u oba smera) i rastuće amplitude), se dovodi na sva tri modela i prati se izlaz (visina nivoa). Ocenjuje se validnost opsega linearizovanog modela – njegovo slaganje i neslaganje sa nelinearnim modelom u različitim tačkama statičke karakteristike). Koristeći programski paket Matlab, dati alternativne predstave i karakteristike modela OU (predstaviti polovi i nule, model u prostoru stanja, frekvencijske karakteristike OU). Parametre rezervoara i komponenti sistema je moguće preuzeti od proizvođača profesionalne opreme za inženjersku edukaciju: 1) http://www.quanser.com/, 2) http://www.fbk.com/, 3) http://www.tecquip.com/, 4) http://www.kri.com.sg/, ili ih usvojiti u skladu sa „sopstvenim inženjerskim razlozima“, usvajanjem racionalnih pretpostavki. U strukturi sistema predvideti poremećajni ulazni protok.

2. **Identifikacija OU** – *U okolini radne tačke identifikovati adekvatan matematički model*. Nelinearni Simulink model OU je osnova za dalje simulacije (taj model tretiramo kao realni OU) – i praktično nad tim modelom se vrši "eksperiment" i sprovodi parametarska identifikacija. Ovakva vežba pruža mogućnost da se ispitaju efekti uticaja izbora reda modela, pobude, periode odabiranja, identifikacione metode na kvalitet dobijenog modela (ARX, na primer). Ovde jednostavno treba opisati postupak identifikacije modela korišćenjem programskog paketa Matlab i izvršiti njegovu verifikaciju.

3. Sinteza algoritma upravljanja – Odabrati strukturu i podesiti parametre PID zakona upravljanja. Kreirati i kontinualni i digitalni PID algoritam.

4. Analiza sistema – Analizirati dinamičke osobine dobijenog sistema sa povratnom spregom (vidi pod 3.) primenom programskog paketa Matlab i njegovih alata ltview i sisotool. (Dati i analizirati parametre dinamičkog odziva u okolini izabrane radne tačke, frekvencijske karakteristike sistema, raspored polova i nula, itd.). Podsećanja radi: zakon upravljanja je u prethodnoj tački definisan na osnovu linearnog modela OU, dok se u proverama efikasnosti sistema putem simulacije dinamičkih odziva koristi realan, nelinearan, model OU. S druge strane, u analizi sistema (u ovoj tački) se koristi linearna teorija, te se uzima u obzir sistema sa povratnom spregom koja obuhvata kreirane PID zakone upravljanja iz prethodne tačke, i linearni model OU.

5. Rezultati simulacije – *Simulirati rad sistema upravljanja u svim njegovim realnim aspektima* (realan OU nije linearan te u svrhu simulacije treba koristiti nelinearni model OU, u digitalnoj realizaciji postoji šum kvantovanja, itd). Testirati simulacijom valjanost zakona upravljanja, kontinualnog i digitalnog, i dati sledeća

dva dijagrama: a) odziv od strane zadate vrednosti (poremećaj = 0) b) odziv od strane poremećaja (referentni nivo je konstantan).

Eventualno: c) Propratiti simulaciju animacijom, d) Ispitati osetljivost sistema na promene parametara OU (nacrtati odgovarajuće indekse performanse i analizirati njihovu promenu), e) Na datom OU primeniti fuzzy zakon upravljanja i uporediti njegovu efikasnost u odnosu na PID zakon upravljanja, f) Primeniti neki od naprednih zakona upravljanja po izboru.

6. Zaključak

7. Literatura (i eventualno **Prilog)**

4.
Primeri rešavanja problema i izvodi iz teorije

Metodologija učenja na problemu (problem based learning) bazirana na datoj eksperimentalnoj aparaturi u opsegu fundamentalnih interdisciplinarnih teorijskih sadržaja obuhvata

A. Koncepte modeliranja i analize sistema

- Modeliranje sistema
- Identifikacija sistema
- Linearizacija modela
- Konverzije modela (dati model u kompleksnom, frekvencijskom i vremenskom domenu, ili model u formi funkcije prenosa, frekvencijskih karakteristika ili model u formi prostora stanja)
- Ispitivanje stabilnosti i drugih performansi sistema na osnovu modela

B. Algoritme upravljanja u širem smislu (sa njihovim pridruženim funkcijama)

- Upravljanje u otvorenoj i zatvorenoj povratnoj sprezi
- PID upravljanje i procedure podešavanja PID regulatora
- P upravljanje sa pridruženim kompenzatorima
- Sinteza kontrolera u frekvencijskom domenu
- Povratna sprega po stanju, sinteza kontrolera zasnovana na modelu u prostoru stanja i koncepti optimalnog upravljanja (LQR kontroler itd.)
- Primena opservera ili estimatora stanja (Kalmanov filter, itd.)
- Sinteza kontrolera metodom podešavanja polova
- Prediktivno upravljanje (MPC algoritmi)

- Fazi upravljanje, itd.

C. Faktore ograničenja u realnom funkcionisanju sistema

- Prisustvo mernog šuma
- Delovanje poremećaja na rad sistema
- Efekti nemodelirane dinamike (ili neodređanosti modela)
- Efekti zasićenja aktuatora (wind-up)

Većina nabrojanih tema je teorijski objašnjena u [10]. U daljem tekstu, na primeru sistema sa slici 12, kao što je to započeto u poglavlju 2.3, biće objašenjena primena nekih teorijskih koncepata.

4.1. IDENTIFIKACIJA KARAKTERISTIKA SISTEMA

Identifikacija sistema podrazumeva generisanje modela sistema na osnovu eksperimentalnih podataka. Postoje brojne tehnike identifikacije sistema i njegovih karakteristika. Prema cilju identifikacije razlikujemo strukturnu i parametarsku identifikaciju, identifikaciju statičkih i dinamičkih karakteristika, itd. Strukturna identifikacija se najčešće svodi na pravilnu pretpostavku strukture modela. Recimo, za slučaj sistema sa slike 12 je opravdano pretpostaviti sledeći statički i dinamički model

$$H_0 = K_{st} Q_{in0}^2, \ \frac{h(s)}{q_{in}(s)} = \frac{K}{Ts+1}. \tag{25}$$

gde se nepoznati parametri K_{st}, K, T određuju u postupku parametarske identifikacije. Oblik modela (25) proizilazi iz razmatranja datih u poglavlju 2.3. Linearni dinamički model je moguć samo u okolini radne tačke, tako da ne treba gubiti iz vida da su apsolutne veličine stanja procesa zbir relativnih i nomialnih veličina stanja. Nominalna stanja određuju radnu tačku.

$$H(t) = H_0 + h(t), \ Q_{in}(t) = Q_{in0} + q_{in}(t).$$

Pretpostavimo sada da raspolažemo setom ulazno izlaznih podataka $(Q_{in}(t), H(t))$, koji je nastao kao posledica pobude $Q_{in}(t) = Q_{in0} + 1(t)$, gde je $1(t)$ jedinična odskočna funkcija. Pretpostavimo i da je prethodno identifikovana statička karakteristika procesa, tako da sekvence $(Q_{in}(t), H(t))$ možemo normalizovati na $(q_{in}(t), h(t)) = (Q_{in}(t) - Q_{in0}, H(t) - H_0)$. Jedinični odskočni odziv dinamičkog modela (25) je dat relacijom

$$\hat{h}(t) = L^{-1}\left\{\frac{1}{s}\frac{K}{Ts+1}\right\} = K(1 - e^{-\frac{t}{T}}). \tag{26}$$

te parametre modela K i T treba odredititi tako da odziv modela $\hat{h}(t)$ najbolje odgovara eksperimentalnom odzivu $h(t)$. Jedna od mogućnosti da se to uradi jeste

primena metode najmanjih kvadrata (pravi izbor parametara daje minimalnu sumu kvadrata greške $\int (h(t) - \hat{h}(t))^2 dt$), a kako je to moguće ostvariti u MATLABu je ilustrovano sledećim programom.

```
%-----------------------------------------------
%Program za identifikaciju parametara modela I reda: K/(Ts+1)
%-----------------------------------------------
% Definisanje radne tačke (Qo,Ho)
Ho = 15; Qo = 17.43;

% exp ulazno izlazna sekvenca (Q,H) je smestena u fajlu ident1.m
% prva kolona podataka je protok (ulaz), a druga nivo (izlaz)
load ident1.m;             % očitavanje podataka iz fajla
id=ident1;
q=ident1(:,1)-Qo;          % normalizovana ulazna sekvenca (je
                           %jedinični odskočni signal)
h=ident1(:,2)-Ho;          % normalizovana izlazna sekvenca
% ako je perioda odabiranja 0.1s, onda je vremenski niz
for i=1: length(h)
t(i)=i*0.1;                % Ts=0.1
end

% ogranicenje resenja: K,T=>0
% Kriterijum: minimizacija sume kvadrata odstupanja
k0=[2.5,10];         % vector početih parametara K, T pri num. izrač.
f2=inline(' k(1)*(1-exp(-t/k(2)))','k','t');
[k,sumakv]=lsqcurvefit(f2,k0,t,h,[0 0], [inf inf]);
K=k(1);              % identifikovano pojačanje K
T=k(2);              % identifikovana vremenska konstanta T
S1=sumakv;           % ostvarena suma "najmanjih kvadrata"
```

Pokazana metodologija može biti korišćena za modele proizvoljne složenosti. Pri tome treba voditi računa da tačnost numeričkih izračunavanja može da zavisi od počeno izabranog vektora nepoznatih parametara k0.

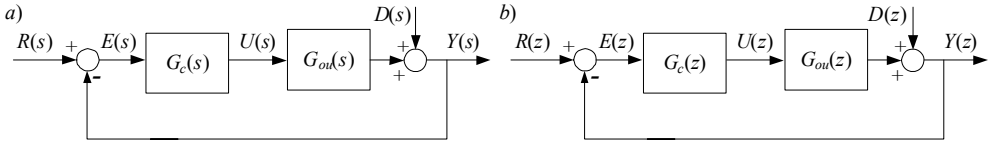
Postupak identifikacije je moguće sprovesti i na druge načine, korišćenjem diskretnih modela (ARX, ARMAX, itd.), Ident Toolboxa programskog paketa MATLAB, i zatim konverzijom modela u pogodnu formu.

Po pravilu, parametarska identifikacija modela se sprovodi na 2/3 eksperimentalnih podataka, dok se dobijeni model verifikuje u poređenju sa preostalom trećinom eksperimentalnih podataka koja nije korišćena za dobijanje modela u procesu identifikacije.

4.2. KONVENCIONALNE METODE UPRAVLJANJA

U [10] je sistematizovana posebna glava udžbenika o algoritmima upravljanja, koja može biti korisna i studentima poslediplomskih studija. Pored konvencionalnih algoritama upravljanja, dato je i kraće poglavlje o fazi upravljanju. U okviru

sledećeg poglavlja, ovde će fazi upravljanju biti posvećena još veća pažnja, kao jednom od često korišćenih nekonvencionalnih upravljačkih koncepata. Kada je reč o konvencionalnim upravljačkim metodama, uglavnom se radi o linearnim algoritmima čija je sinteza zasnovana na linearnom modelu upravljanog procesa. Sa stanovišta sinteze algoritma upravljanja SISO sistema, uobičajeno je da se kontinualni ili digitalni sistem upravljanja predstavlja strukturnim blok dijagramom kao na slici 17, mada, shodno slici 6 i iznetom u [10], struktura algoritma upravljanja može biti i složenija.



Slika 17. ***Kontinualni (a)) i digitalni (b)) sistem upravljanja sa povratnom spregom***

Pošto se o linearnosti procesa može samo uslovno govoriti (u okolini radne tačke), jako je važno razumeti principe primene linearnog zakona upravljanja na nelinearnom objektu upravljanja (tj. suštinsko razumevanje poglavlja 2.3 je jako važno).

Najbitniji predstavnik konvencionalnih algoritama upravljanja jesu PID algoritmi. Radi se o heurističkom algoritmu upravljanja koji, shodno strukturi na slici 17, u svojoj idealizovanoj formi, može biti predstavljen sledećom funkcijom prenosa

$$G_c(s) = \frac{U(s)}{E(s)} = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) = K_p + \frac{K_i}{s} + K_d s \tag{27}$$

ili u diskretnoj formi

$$G_c(z) = \frac{U(z)}{E(z)} = K_p\left(1 + \frac{T_s/T_i}{1 - z^{-1}} + \frac{T_d}{T_s}(1 - z^{-1})\right) \tag{28}$$

gde je T_s perioda odabiranja, a K_p, K_i i K_d ili K_p, T_i i T_d parametri zakona upravljanja. Bez obzira na brojne modifikacije (neke su date u [10]) suština tumačenja PID kontrolera ostaje ista. Radi se o vrlo popularnom industrijskom kontroleru, dokazane upotrebne vrednosti, sa samo tri podešljiva parametra. Signal greške (razlika između željene i postignute vrednosti izlazne promenljive procesa) se obrađuje tako da se superponiraju ishodi uticaja P, I i D dejstva.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{29}$$

Proporcionalano dejstvo operiše sa signalom greške u sadašnjosti, integralno sa signalom greške u prošlosti, a diferencijalno sa predikcijom signala greške (estimiranom vrednošću signala greške u budućnosti). Sledeća tabela ilustruje kakav doprinos po performanse sistema pruža upotreba P, I i D dejstva

	Vreme uspona t_u	*Preskok* Π	*Vreme smirenja* t_s	*Greška stacinarnog stanja* $e(\infty)$
P dejstvo $\nearrow$	Smanjuje se	Povećava se	Malo utiče	Smanjuje se
I dejstvo $\nearrow$	Smanjuje se	Povećava se	Povećava se	Eliminiše se
D dejstvo $\nearrow$	Malo utiče	Povećava se	Smanjuje se	Malo utiče

Tabela 2. ***Uticaj P, I, D dejstva na performanse sistema***

Međutim, uticaji P, I i D dejstva su još složeniji nego što je to iskazano u tabeli 1, te se čitalac upućuje na [10] i širu literaturu. Podešavanje PID regulatora može teći u „online" i „offline" režimu. U „online" režimu podešavanja parametara se po pravilu isključuje I i D dejstvo (K_i, K_d = 0) i povećava se P dejstvo ($K_p \nearrow$) sve dok posmatrani izlaz sistema ne počne da osciluje. Zatim se povećava I dejstvo sve dok se oscilacije potpuno ne priguše. Na kraju se povećava D dejstvo sve dok sistem odgovarajućom brzinom ne postigne referentnu vrednost izlaza (obično dolazi do blagog preskoka, i ako je takva performansa sistema prihvatljiva, sistem se naknadno ne prepodešava). Ipak treba istaći da ne postoji univerzalna i istovremeno efikasna metoda podešavanja PID regulatora jer su objekti upravljanja jednostavno različiti, te i scenario „online" podešavanja nije uvek. Uvek kada je to moguće, pribegava se „offline" podešavanju parametara regulatora, a zatim se neophodne korekcije sprovode u „online" režimu. Sem opštih metoda sinteze linearnih regulatora, postoje razne inženjerske metode podešavanja PID regulatora. Najstarija pravila podešavanja parametara PID kontrolera su Zigler – Nikolsova pravila i detaljno su opisana u [10]. Ukoliko je objekat upravljanja dat modelom tipa

$$G_{ou}(s) = \frac{Ke^{-\tau s}}{Ts+1}. \tag{30}$$

Tada se prema Ziegler-Nikolsovoj proceduri (ali ne prema metodi graničnog ciklusa) predlažu sledeće vrednosti parametara regulatora

Tip kontrolera	*Predložene vrednosti parametara*
P	$K_p = T/\tau$
PI	$K_p = 0{,}9T/\tau$, $T_i = \tau/0{,}3$
PID	$K_p = 1{,}2T/\tau$, $T_i = 2\tau$, $T_d = 0{,}5\tau$

Tabela 3. ***Zigler-Nikolsova pravila podešavanja PID kontrolera prema eksperimentu u otvorenoj povratnoj sprezi***

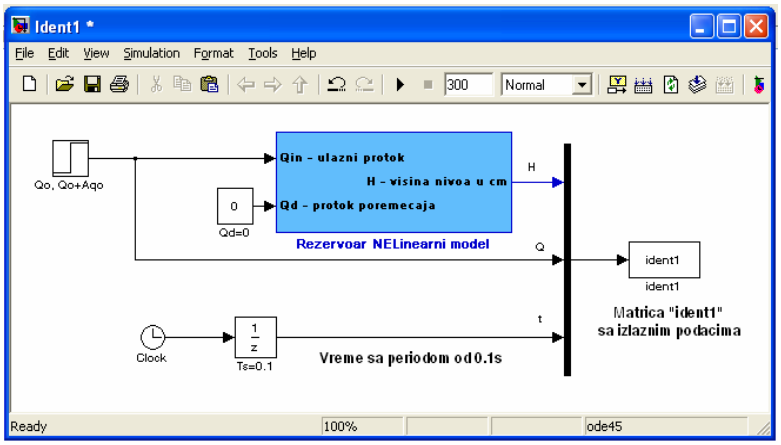
Upravljani proces sa slike 11 je saglasno matematičkom modelu (20) i izabranim vrednostima radne tačke (v. slike 13-15) dat relacijom

$$G_{ou}(s) = \frac{h(s)}{q_{in}(s)} = \frac{1.721}{27.54s+1}.$$

ali ga je moguće modelirati i modelom strukture (30), ukoliko takvu strukturu modela pretpostavimo, a onda identifikujemo nepoznate parametre. Da bi smo obezbedili ponovljivost rezultata, to ćemo ilustrovati na simulacionom modelu OU

datom na slici 12. Neka je nelinearni model sa slike 12 testiran odskočnim signalom ulaznog protoka amplitude A_{qo} (vrednost protoka se menja od nominalne vrednosti Qo do $Qo+A_{qo}1(t)$). Sa pretpostavkom dinamičkog modela (30), odziv sistema je dat relacijom

$$H_0 + \hat{h}(t) = H_0 + L^{-1}\left\{\frac{A_{q0}}{s}\frac{Ke^{-s\tau}}{Ts+1}\right\} = H_0 + A_{q0}K(1-e^{-\frac{t-\tau}{T}}). \qquad (31)$$



Slika 18. Simulink model u cilju dobijanja ulazno izlaznih podataka za identifikaciju

Koristi se sličan Simulink model kao na slici 15, odnosno model na slike 18. Ulazno izlazni podaci, i vreme uzorkovano sa periodom odabiranja 0.1s, smeštaju se u matricu *ident1*, a procedura identifikacije je objašnjena programskim rešenjem ispod.

```
%-----------------------------------------------------
%Program za identifikaciju parametara modela (30)
%-----------------------------------------------------
t=ident1(:,3);              % vremenska sekvenca, Ts=0.1
h=ident1(:,1)-Ho;           % normalizovana izlazna sekvenca
q=ident1(:,2)-Qo;           % normalizovana ulazna sekvenca

% ogranicenje resenja: K,T,Tau=>0
k0=[2.5,10,1];              % vektor početih parametara K, T, Tau
f2=inline(' k(1)*(1-exp(-(t-k(3))/k(2)))','k','t');
[k,sumakv]=lsqcurvefit(f2,k0,t,h,[0 0 0], [inf inf inf]);
K=k(1)/Aq0;          % identifikovano pojačanje K
T=k(2);              % identifikovana vremenska konstanta T
Tau=k(3);            % identifikovano vremensko kašnjenje Tau
S1=sumakv;           % ostvarena suma "najmanjih kvadrata"
```

S obzirom na red vremenske konstante procesa T, korektan model bi se dobio i pretpostavkom fiksne vrednosti vremenskog kašenjenja od recimo τ=1 s, dok bi se preostali parametri modela računali saglasno

```
k0=[2.5,10];        % vektor početih parametara K, T,
f2=inline(' k(1)*(1-exp(-(t-1)/k(2)))','k','t');
[k,sumakv]=lsqcurvefit(f2,k0,t,h,[0 0], [inf inf]);
K=k(1)/Aq0;         % identifikovano pojačanje K
T=k(2);             % identifikovana vremenska konstanta T
```

Naredna tabela pruža pregled rezultata parametarske identifikacije modela (30) u slučaju procesa prikazanog na slici 11, a za poznate parametre iz tabele 1. Slično kao na slici 15, poređenjem izlaza nelinearnog modela i identifikovanog linearnog modela je moguće potvrditi korektnost izračunatih vrednosti iz tabele 4. Variranjem vrednosti vektora **k**0 mogu se ilustrovati i numerički aspekti predloženog načina parametarske identifikacije linearnog modela. Variranje parametara modela u zavisnosti od amplitude ulaznog signala je već komentarisano kroz rezultat na slici 14. Praktično, greška modeliranja uvek postoji, i može se diskutovati kada je ona prihvatljiva, a kada ne.

Amplituda A_{qo}	*Identifikovani parametri modela* (30)			*Identifikovani paremtri uz pretpostavku da je* τ=1 s	
1	K=1.76	T=28.5	$\tau = 0$	K=1.76	T=27.45
3	K=1.86	T=30.73	$\tau = 0$	K=1.86	T=29.67
5	K=1.96	T=33.59	$\tau = 0.05$	K=1.96	T=31.9

Tabela 4. ***Ishodi parametarske identifikacije***

Identifikovani model

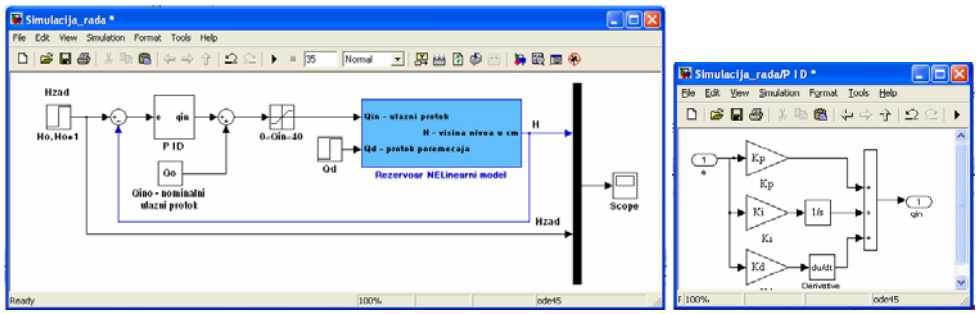
$$G_{ou}(s) = \frac{h(s)}{q_{in}(s)} = \frac{1.76e^{-s}}{27.45s + 1}.$$

pruža mogućnost primene Zigler-Nikolsovih pravila podešavanja PID kontrolera iz tabele 3.

Tip kontrolera	*Predložene vrednosti parametara*
P	$K_p = T/\tau = 27.45$
PI	$K_p = 0,9T/\tau = 24.71$, $T_i = \tau/0,3 = 3.33$
PID	$K_p = 1,2T/\tau = 32.94$, $T_i = 2\tau = 2$, $T_d = 0.5\tau = 0.5$

Tabela 5. ***Parametri PID kontrolera***

Dobijeni PID kontroler se testira simulacijom, i pri tome se želi ispitati efikasnost dobijenog kontrolera u što realnijim uslovima (nelinearna dinamika procesa, protok pumpe kao upravljačka promenljiva ima ograničene vrednosti od nulte do maksimalne (40 cm^3/s), postoji dejstvo poremećaja na sistem, itd.).

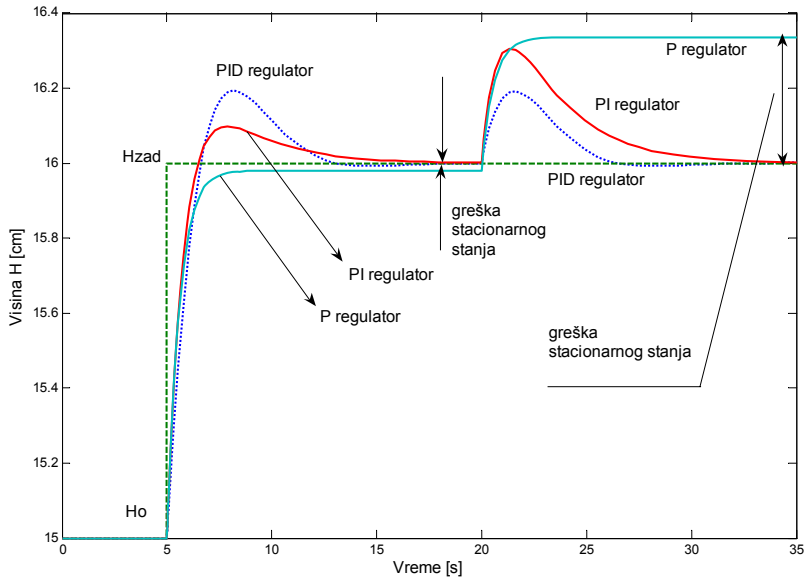


Slika 19. ***Simulacija rada sistema sa PID kontrolerom u uslovima promene zadate vrednosti nivoa $H_{zad}(t)=H_0+1(t-5)$ i poremećaja $Q_d(t)=10\cdot1(t-20)$***

Posebno, na slici 19, treba obratiti pažnju na implementaciju linearnog zakona upravljanja na nelinearnom objektu: izlaz PID algoritma $q_{in}(t)$ je aditiv na nominalnu upravljačku promenljivu Q_{in0} koja određuje radnu tačku

$$Q_{in}(t)=Q_{in0}+q_{in}(t),\ q_{in}(t)=K_p e(t)+K_i\int_0^t e(\tau)d\tau+K_d\frac{de(t)}{dt},\ e(t)=H_{zad}(t)-H(t).$$

Funkcija kontrolera je generisanje upravljačke promenljive $Q_{in}(t)$, pri čemu je jasno da je u našem razmatranju dinamika aktuatora (pumpe) zanemarena (što nema implikacije principijelnog karaktera). Rezultati simulacije su očekivani. Posebno je interesantno uporediti brzinu sistema sa i bez povratne sprege (slike 20 i 14). Treba naglasiti da primena poznatih pravila podešavanja regulatora retko kad daje konačno rešenje, već je često samo polazna tačka za dodatna prepodešavanja regulatora.



Slika 20. ***Rezultati simulacije na osnovu Simulink modela sa slike19.***

Analitičko projektovanje PID regulatora može biti sprovedeno bilo kojom metodom projektovanja linearnih regulatora uz ograničenje da se radi o regulatoru poznate strukture. Uvek se nastoji da zahtevane performanse sistema budu postignute kontrolerom što jednostavnije strukture sa što manje podešljivih parametara koji po mogućstvu imaju konkretna fizička značenja (da bi bilo omogućeno jednostavno preopodešavanje kontrolera). U sintezi linearnih regulatora, programski paket Matlab i njegovi alati Sisotool i Ltiview pružaju moćnu podršku. Pomenuti alati imaju izuzetno kvalitetan korisnički interfejs, te uz pretpostavku da se relevantna teorija poznaje, objašnjavanje njihovog korišćenja nije potrebno.

```
sisotool
ltiview
% putem opcije Import u meniju File selektuju se funkicije prenosa
% primer definicije funkcije prenosa u Matlabu, gore pomenute:
G1=tf(1.76, [27.45 1], 'inputdelay', 1);
G2=tf(1.721, [27.54 1]);
```

4.3. FAZI UPRAVLJANJE

4.3.1. Primena fazi upravljanja

Preko 90% kontrolera koji se danas primenjuju u industriji su PID kontroleri. U teoriji automatskog upravljanja postoji više procedura za podešavanje parametara standardnih PID regulatora. Parametri se mogu podešavati: Ziegler-Nichols-ovom metodom, ručno, podešavanjem polova, optimizacijom, itd. Primena linearnih regulatora daje zadovoljavajuće performanse kada je objekat linearan ili kada je nelinearnost blago izražena, pri čemu je brzina promene parametara mala. Različite varijante adaptivnog upravljanja su implementirane za upravljanje nelinearnim objektima čija je brzina promene parametara velika.

Fazi kontroleri se uspešno primenjuju za upravljanje objektima koje je nemoguće matematički modelirati ili je model izrazito nelinearan i izložen dejstvu jakih poremećaja. Često su iskusni opearteri, koji imaju praktična znanja stečena više-godišnjim iskustvom, upravljali složenim procesima. Delovanje operatera bazirano je na pravilima oblika:

Ako *su tolike vrednosti posmatranih veličina procesa* **tada**

je upravljačko dejstvo toliko

(Npr. **Ako** *je temperature niska i pada* **tada** *povećaj protok vode)*

Primena teorije fazi skupova i fazi logike u matematičkoj interpretaciji empirijskog znanja omogućila je da se u kompjuterski proces vođenja uključi ekspertsko znanje najboljih operatera.

Pored toga što su korišćeni za upravljanje složenim procesima koji se ne mogu matematički modelirati i nelinearnim objektima, brojne su primene fazi kontrolera u upravljanju objektima kod kojih bi zadovoljavajuće performanse mogle da se dobiju

i primenom konvencionalnih upravljačkih algoritama. Kao razlog primene u ovakvim slučajevima često se navodi jednostavnija sinteza fazi kontrolera.
Za uspešnu sintezu konvencionalnih kontrolera od velikog značaja je tačnost modela. Sam model nikada ne može biti toliko precizan da u potpunosti odražava realan objekat. Postoje objektivne nemogućnosti da se sagledaju sve karakteristike i svi mogući faktori koji utiču na ponašanje objekta. Identifikaciju linearnih i nelinearnih sistema, tj. dobijanje matematičkih modela dinamičkih sistema na osnovu ulazno/izlaznih podataka, moguće je ostvariti primenom fazi sistema.

4.3.2. Fazi skupovi i fazi logika – osnovni pojmovi

Definicija fazi skupa. Lingvistička promenljiva.

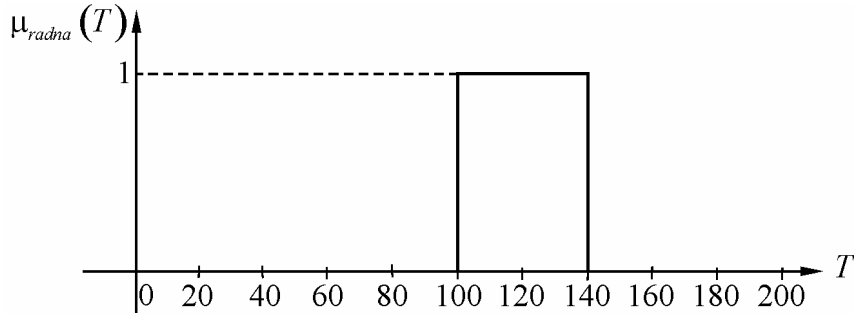
U klasičnoj Bulovoj logici, pripadnost elementa $x \in \boldsymbol{X}$ skupu $\boldsymbol{A}$ opisuje se karakterističnom funkcijom koja se definiše kao:

$$\mu_A(x) = \begin{cases} 1 \ \textit{ako je} \ x \in \boldsymbol{A} \\ 0 \ \textit{ako je} \ x \notin \boldsymbol{A} \end{cases}.$$

Element x ili pripada ili ne pripada posmatranom skupu $\boldsymbol{A}$. $\boldsymbol{X}$ je univerzalni ili referentni skup koji sadrži sve objekte povezane sa datim kontekstom.
Neka je radna temperatura nekog procesa od $100°C$ do $140°C$. U klasičnoj teoriji skupova pojam *radna temperatura* može da se definiše kao:

$$\mu_{radna}(T) = \begin{cases} 1, & T \in [100,140] \\ 0, & T \notin [100,140] \end{cases}.$$

Karakteristična funkcija skupa *radna* prikazana je na slici 21.



Slika 21. ***Karakteristična funkcija skupa radna temperatura***

Ovakva stroga definicija pripadnosti elementa skupu dovodi do zaključka da temperature od $99.8°C$ ili $140.2°C$ nisu radne temperature jer ne pripadaju definisanom skupu.

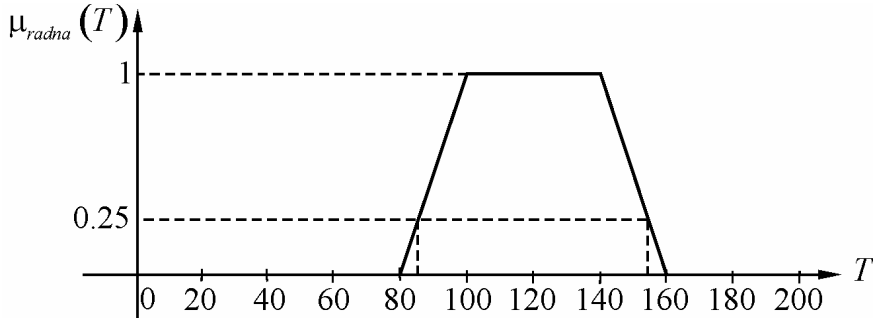
Fazi skupovi nemaju jasno izražene granice pa neki elementi mogu delimično pripadati skupovima. Stepen pripadnosti elementa skupu može da bude bilo koji broj iz intervala $[0,1]$. U razmatranom primeru teperatura $99.8°C$ je bliska temperaturi

$100°C$ pa možemo da kažemo da pripada skupu *radna temperatura* sa stepenom pripadnosti 0.99, kao i temperatura $140.2°C$. Temperature $85°C$ i $155°C$ malo pripadaju skupu *radna temperatura,* sa stepenom pripadnosti 0.25. Funkcija pripadnosti fazi skupa mogla bi da se definiše kao:

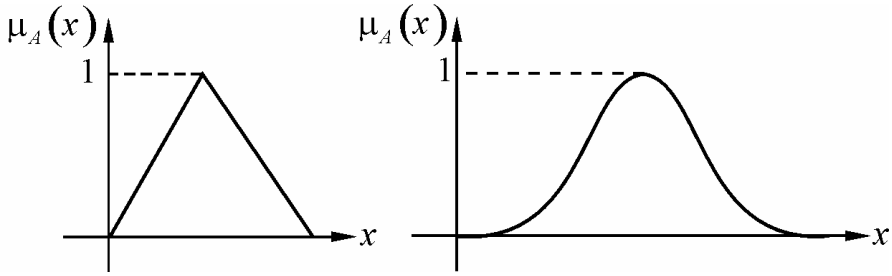
$$\mu_{radna}(T) = \begin{cases} 0.05T - 4, & T \in [80,100] \\ 1, & T \in [100,140] \\ -0.05T + 8, & T \in [140,160] \end{cases}.$$

Grafički prikaz funkcije pripadnosti skupa *radna temperatura* dat je na slici 22.

Za predstavljanje fazi skupova, pored trapeznog oblika, često se koriste trougaona i Gausova funkcija, slika 23.



Slika 22. ***Funkcija pripadnosti fazi skupa radna temperatura***



Slika 23. ***Tipični oblici funkcija pripadnosti fazi skupova***

U opštem slučaju, fazi skup $\boldsymbol{A}$ u univerzalnom skupu $\boldsymbol{X}$ određen je funkcijom pripadnosti $\mu_A(x)$ koja svakom elementu iz $\boldsymbol{X}$ dodeljuje realan broj iz intervala $[0,1]$. Fazi skup se predstavlja uređenim parom:

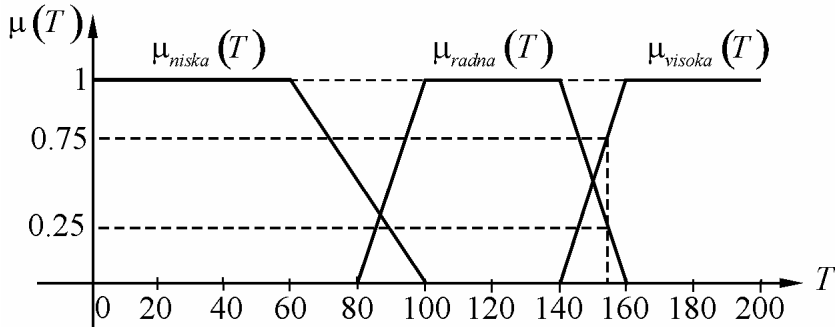
$$\boldsymbol{A} = \{(x, \mu_{\boldsymbol{A}}(x)) | x \in \boldsymbol{X}\}.$$

U primeru sa slike 22 mi smo posmatrali temperaturu u intervalu $[0°C, 200°C]$. Nad univerzalnim skupom definisali smo jedan fazi skup, *radna temperatura.* Na isti način moguće je definisati fazi skupove *niska temperatura* i *visoka temperatura* i predstaviti ih odgovarajućim funkcijama pripadnosti, (slika 24):

$$\mu_{niska}(T) = \begin{cases} 1, & T \in [0,60] \\ -0.025T + 2.5, & T \in [60,100] \end{cases}$$

$$\mu_{visoka}(T) = \begin{cases} 0.05T - 7, & T \in [140,160] \\ 1, & T \in [160,200] \end{cases}$$

Sa slike 24 se vidi da $155°C$ pripada fazi skupu *radna temperatura* sa stepenom pripadnosti 0.25 i fazi skupu *visoka temperatura* sa stepenom pripadnosti 0.75.
Fazi skupovi *niska temperatura, radna temperatura i visoka temperatura* određuju vrednosti lingvističke promenljive temperatura. Fazi particioniranje promenljive temperatura moguće je ostvariti izborom većeg broja lingvističkih vrednosti., npr. *veoma niska temperatura*, *niska temperatura*, *radna temperatura*, *visoka temperatura*, *veoma visoka temperatura*. Svakako da je potrebno svaku lingvističku vrednost, tj. primarni fazi skup predstaviti odgovarajućom funkcijom pripadnosti.



Slika 24. ***Grafički prikaz lingvističke promenljive teperatura***

Lingvistička pravila.

Odluke koje donosi čovek koji upravlja nekim procesom, zasnivaju se na pravilima sledećeg oblika:

Ako x je A_1 *i* y je B_1 tada z je C_1

takođe: *ako* x je A_1 *i* y je B_2 tada z je C_2

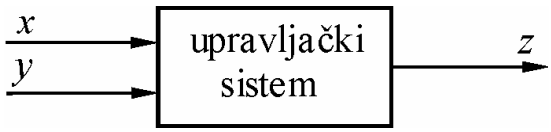
...

takođe: *ako* x je A_i *i* y je B_j tada z je C_k

...

takođe: *ako* x je A_l *i* y je B_m tada z je C_n

Broj ulaznih i izlaznih promenljivih zavisi od vrste procesa. Mi smo uzeli da upravljačko dejstvo operatera zavisi od dve promenljive procesa x i y, slika 25.



Slika 25. ***Sistem upravljanja sa dva ulaza i jednim izlazom***

A_i, i=1, 2…,*l*; B_j, j=1, 2…,*m*; i C_k, k=1, 2…,*n*; su lingvističke vrednosti promenljivih *x*, *y* i *z*, respektivno. Maksimalan broj lingvističkih pravila je $p = l \cdot m$.

Lingvistička ako–tada pravila imaju dva dela: premisa, izražena *ako* <fazi predlog>; posledica, izražena: *tada* <fazi predlog>. Složen fazi predlog formira se korišćenjem: *i*, *ili*, *nije* itd. Veznik *i* se predstavlja operatorom preseka, dok se veznici *ili* i *takođe* predstavljaju operatorom unije.

Primenom teorije fazi skupova i fazi logike moguće je matematički interpretirati fazi pravila. Za konkretne vrednosti ulaza x^* i y^* i na osnovu pravila može da se odredi vrednost upravljačkog dejstva z^*.

Predpostavimo da smo fazi particioniranje ulaznih promenljivih i izlazne promenljive ostvarili izborom trougaonih funkcija pripadnosti, pri čemu je $l = 3$, $m = 2$, $n = 3$, slika 26.

Neka su fazi pravila za ovaj sistem oblika:

Ako *x* je A_1 *i* *y* je B_1 tada *z* je C_1

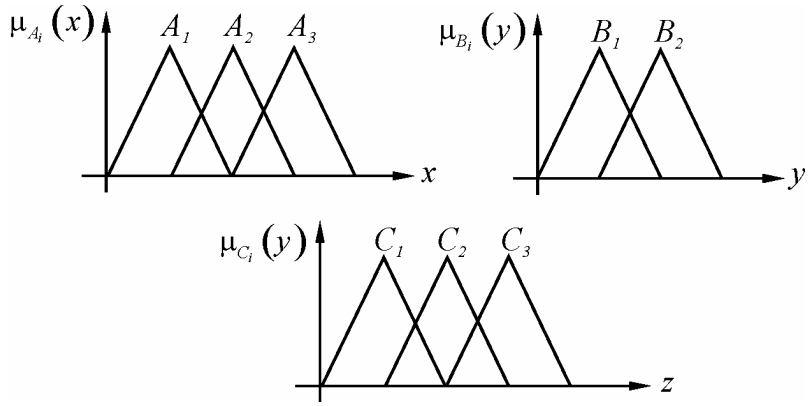
takođe: *ako* *x* je A_1 *i* *y* je B_2 tada *z* je C_2

takođe: *ako* *x* je A_2 *i* *y* je B_1 tada *z* je C_2

takođe: *ako* *x* je A_2 *i* *y* je B_2 tada *z* je C_3

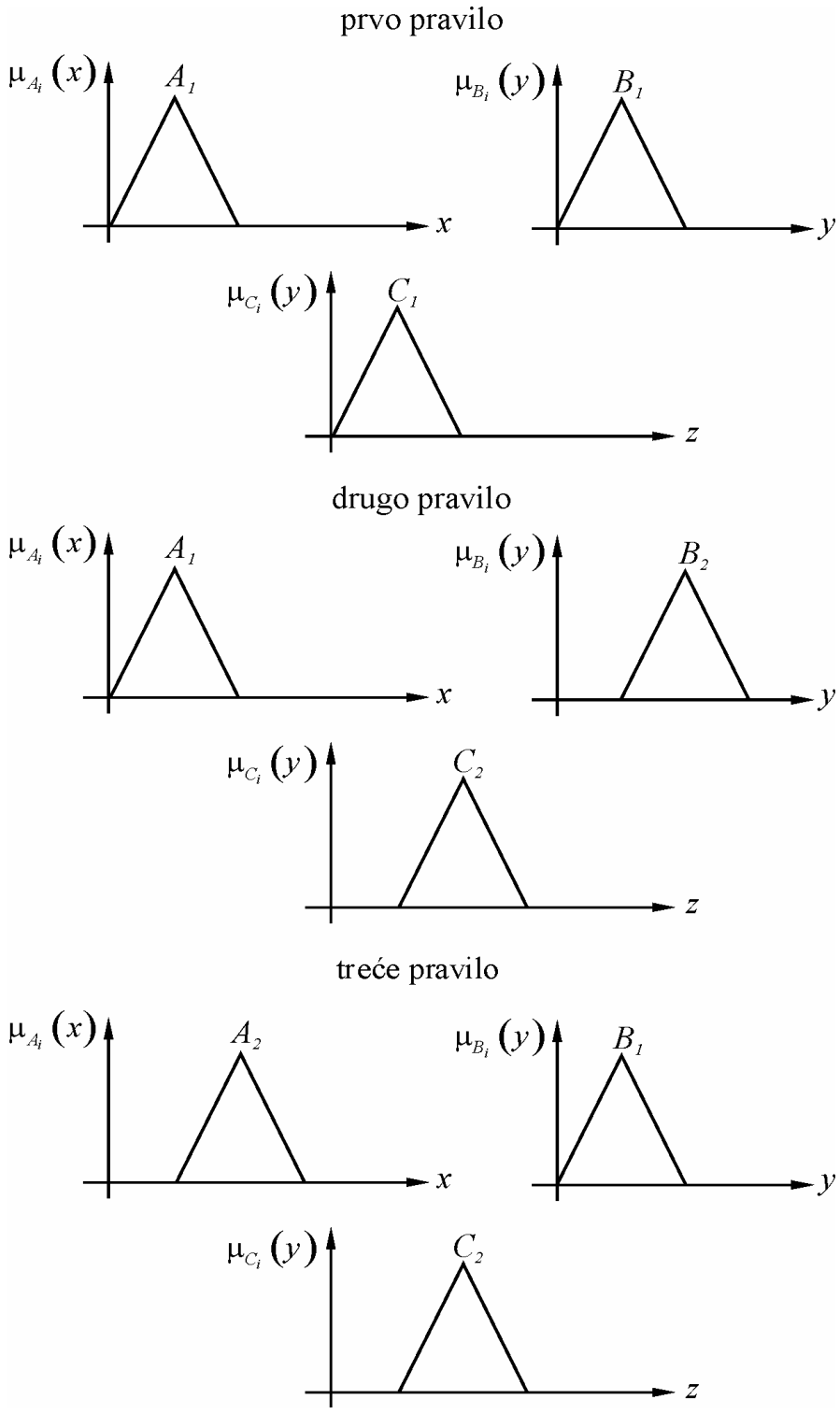
takođe: *ako* *x* je A_3 *i* *y* je B_1 tada *z* je C_2

takođe: *ako* *x* je A_3 *i* *y* je B_2 tada *z* je C_3

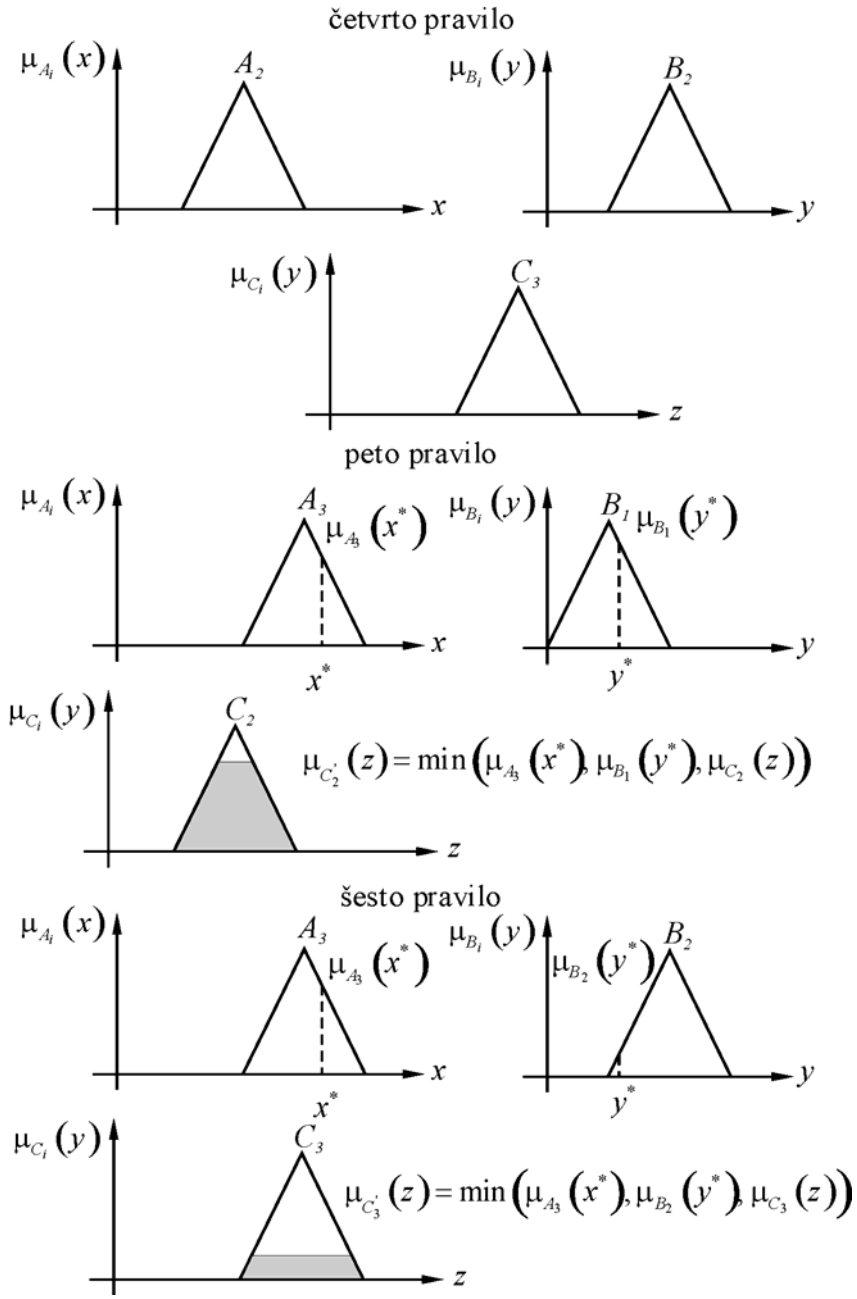


Slika 26. ***Fazi particioniranje lingvističkih promenljivih x, y i z***

Svako fazi pravilo može grafički da se prikaže. Na slici 27. dat je grafički prikaz prva tri pravila pravila. Grafički prikaz četvrtog, petog i šestog pravila dat je na slici 28.



Slika 27. ***Grafički prikaz prva tri pravila***



Slika 28. ***Grafički prikaz četvrtog, petog i šestog pravila***

Za $x = x^*$ i $y = y^*$, slika 28, vidimo da x^* pripada fazi skupu A_3 sa stepenom pripadnosti $\mu_{A_3}(x^*)$, a da y^* pripada fazi skupovima B_1 i B_2 sa stepenima pripadnosti $\mu_{B_1}(y^*)$ i $\mu_{B_2}(y^*)$. Određivanje stepena pripadosti numeričkih vrednosti fazi skupovima naziva se ***fazifikacija.*** Fazifikacijom se numeričkim

vrednostima pridružuje odgovarajuća lingvistička promenljiva. Kako x^* ne pripada fazi skupovima A_1 i A_2 aktivna su samo poslednja dva pravila iz baze pravila, slika 26. Funkcija pripadnosti zaključka $\mu_{C'}$ najčešće se računa korišćenjem jednog od sledećih načina:

$$\mu_{C'}(z) = \max\left\{\min\left(\mu_{A_3}(x^*), \mu_{B_1}(y^*), \mu_{C_2}(z)\right), \min\left(\mu_{A_3}(x^*), \mu_{B_2}(y^*), \mu_{C_2}(z)\right)\right\} \quad (32)$$

ili:

$$\mu_{C'}(z) = \max\left\{\min\left(\mu_{A_3}(x^*), \mu_{B_1}(y^*)\right)\cdot \mu_{C_2}(z), \min\left(\mu_{A_3}(x^*), \mu_{B_2}(y^*)\right)\cdot \mu_{C_2}(z)\right\} \quad (33)$$

Ukoliko koristimo (1) tada funkcija pripadnosti zaključka $\mu_{C'}(z)$ izgleda kao na slici 29. Na slike 30 prikazana je funkcija pripadnosti zaključka kada se računa prema (2). Proces određivanja brojne vrednosti zaključka na osnovu funkcije pripadnosti naziva se ***defazifikacija***.

Najčešći postupci defazifikacije su:

- ***metod maksimuma***

Prema metodu maksimuma kao precizna, defazifikovana vrednost bira se ili prva $\left(z_L^*\right)$, ili zadnja $\left(z_D^*\right)$ tačka u kojoj funkcija pripadnosti dostiže maksimalnu vrednost, slika 29. Prema metodu srednje vrednosti maksimuma defazifikovana vrednost $\left(z^*\right)$ se dobija kao srednja vrednost svih lokalnih vrednosti u kojima funkcija pripadnosti dostiže maksimalnu vrednost:

$$z^* = \sum_{i=1}^{j} \frac{z_i}{m}$$

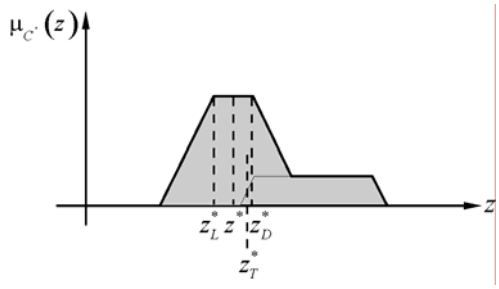
gde je:

z_i -vrednost promenljive za koju funkcija pripadnosti ima maksimalnu vrednost

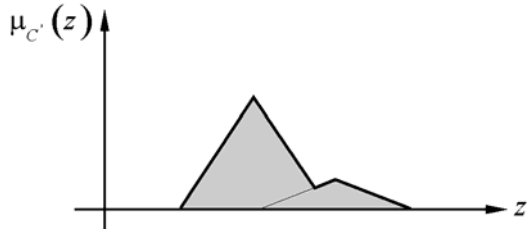
m- broj takvih vrednosti.

- ***metod težišta***

Kod metoda težišta defazifikovana vrednost se dobija određivanjem težišta površine fazi skupa $\boldsymbol{C}'$ (slika 29).



Slika 29. ***Funkcija pripadnosti zaključka određena prema (32) sa prikazom različitih metoda defazifikacije***

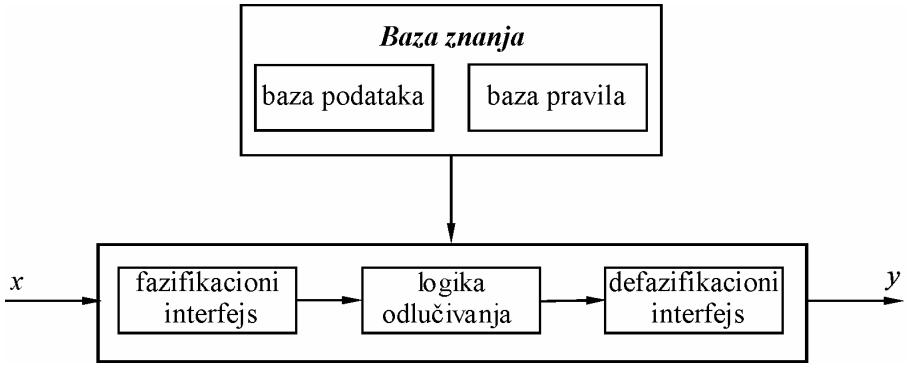


Slika 30. ***Funkcija pripadnosti zaključka određena prema (33)***

4.3.3. Osnovna struktura fazi kontrolera

Osnovna konfiguracija fazi logičkog kontrolera (slika 31) se sastoji iz:

- fazifikacionog interfejsa
- baze znanja
- logike odlučivanja
- defazifikacionog interfejsa



Slika 31. ***Osnovna struktura fazi kontrolera (Legenda:*** x ***je ulaz fazi kontrolera,*** y ***je upravljački signal)***

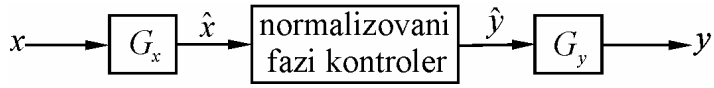
Projektovanje baze podataka podrazumeva fazi particioniranje ulazno/izlaznog prostora čime se određuje broj lingvističkih vrednosti promenljivih x i y. Od broja lingvističkih vrednosti zavisi broj fazi upravljačkih pravila. Optimalno fazi particioniranje se ostvaruje metodom *probe i greške*.

Projektovanje baze pravila najčešće se obavlja na osnovu ekspertskog iskustva i poznavanja teorije upravljanja. Optimalno definisanje fazi upravljačkih pravila se ostvaruje probom.

Osnovni zadatak modula logika odlučivanja je određivanje funkcije pripadnosti fazi zaključka, tj. upravljačkog dejstva. Pri projektovanju ovog modula potrebno je izvršiti izbor načina određivanja fazi vrednosti izlaza kontrolera. Najčešće se koriste jed. (1) ili (2).

Često se pre fazifikacije i posle defazifikacije obavlja ***normalizacija*** ulazno/izlaznih promenljivih. Normalizacijom se domeni ulazno/izlaznih promenljivih, preslikavaju

u normalizovani zatvoreni interval $[-1,+1]$. Normalizacija zahteva poznavanje opsega ulazno/izlaznih signala kontrolera, $[-x_m, x_m]$ i $[-y_m, y_m]$. Ovakvi kontroleri u literaturi se nazivaju ***normalizovani*** fazi kontroleri, slika 32.



Slika 32. ***Normalizovani fazi kontroler***

Na slici 32. $\hat{x}$ i $\hat{y}$ su normalizovane vrednosti ulaza i upravljačkog signala. G_x i G_y su normalizujuća pojačanja:

$$G_x = \frac{1}{x_m}, \quad G_y = y_m .$$

4.3.4. Sinteza fazi PI kontrolera

Za regulaciju nivoa tečnosti u rezervoaru u prethodnom poglavlju korišćen je PI regulator. U ovom poglavlju biće objašnjen postupak projektovanja fazi PI regulatora. Fazi PI regulator je nelinearan. Osnovni izvori nelinearnosti su: algoritam fazifikacije, fazi upravljačka pravila, primenjeni mehanizam zaključivanja, algoritam defazifikacije. Normalizujuća pojačanja nisu izvor nelinearnosti ali se njihovom promenom, može nelinearnost potencirati ili potiskivati.

Zakon upravljanja PI regulatora je:

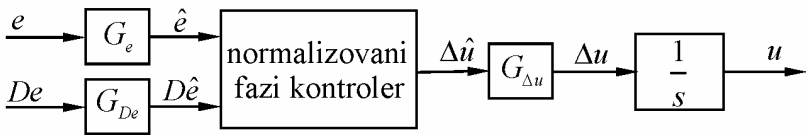
$$u(t) = K_p \left[e + \frac{1}{T_i} \int_0^t e(\tau)\, d\tau \right] \tag{34}$$

ili:

$$u(t) = \int_0^t \left[K_p \frac{de(\tau)}{d\tau} + \frac{K_p}{T_i} e(\tau) \right] d\tau = \int_0^t \Delta u(\tau)\, d\tau \tag{35}$$

gde je: $\Delta u(\tau) = K_p \dfrac{de(\tau)}{d\tau} + \dfrac{K_p}{T_i} e(\tau)$.

Fazi PI kontroler se realizuje kao inkrementalni kontroler, slika 33. Ulazi u fazi kontroler su e i $\dot{e} = De$, a izlaz je promena upravljačkog signala Δu. PI zakon upravljanja može da se dobije integracijom Δu. Baza pravila koja se primenjuje kod fazi PI kontrolera je ista kao i baza pravila fazi PD kontrolera. Za upravljanje procesima koji nemaju astatizam fazi PD regulator, kao i konvencionalni PD, ne eliminiše grešku stacionarnog stanja.



Slika 33. Fazi PI kontroler

Proces projektovanja fazi PI kontrolera se sastoji iz sledećih koraka:

1. korak

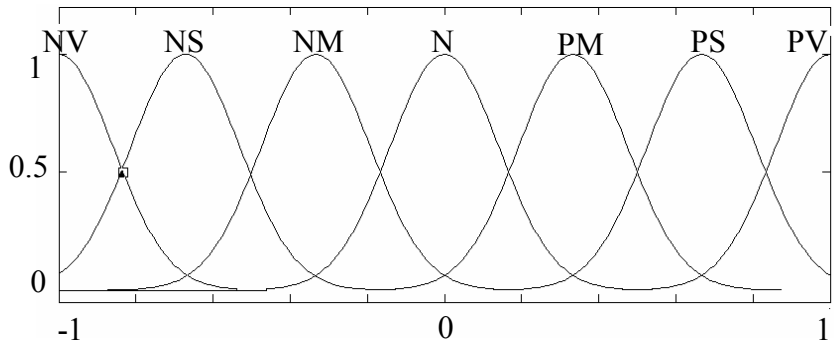
Poznavajući domene ulaznih promenljivih $[-e_m, e_m]$ i $[-De_m, De_m]$ i izlazne promenljive $[-\Delta u_m, \Delta u_m]$ određuju se normalizujuća pojačanja:

$$G_e = \frac{1}{e_m},\ G_{De} = \frac{1}{De_m} \text{ i } G_{\Delta u} = \Delta u_m .$$

2. korak

Projektuje se baza znanja i baza pravila. Fazi particioniranje $\hat{e}$, $D\hat{e}$ i $\Delta\hat{u}$ moguće je ostvariti izborom različitog broja i oblika primarnih fazi skupova. Na slici 34. prikazan je mogući način fazi particioniranja ulazno/izlaznog prostora. Izabrane su Gausove funkcije pripadnosti. Broj lingvističkih vrednosti za sve tri promenljive je 7. Broj i raspored funkcija pripadnosti fazi skupova ima veliki uticaj na performanse kontrolera.

Baza pravila fazi PD kontrolera data je u tabeli 1.



Slika 34. ***Fazi particioniranje ulazno/izlaznog prostora Gausovim funkcijama pripadnosti***

3. korak

Bira se način određivanja funkcije pripadnosti fazi zaključka, jed. (32) ili (33).

Lingvističke vrednosti za $\hat{e}$, $D\hat{e}$ i $\Delta\hat{u}$:

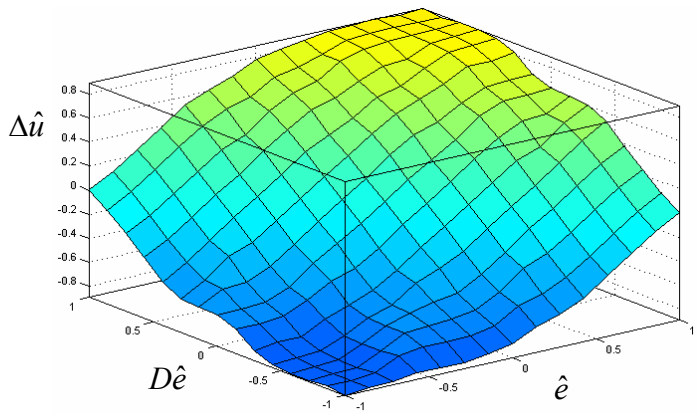
NV-negativno veliko PM-pozitivno malo
NS-negativno srednje PS-pozitivno srednje
NM-negativno malo PV-pozitivno veliko
N-nula

$\hat{e}$ \ $D\hat{e}$	NV	NS	NM	N	PM	PS	PV
NV	NV	NV	NV	NS	NS	NM	N
NS	NV	NV	NV	NS	NM	N	PM
NM	NV	NS	NS	NM	N	PM	PS
N	NV	NS	NM	N	PM	PS	PV
PM	NS	NM	N	PM	PS	PV	PV
PS	NM	N	PM	PS	PV	PV	PV
PV	N	PM	PS	PS	PV	PV	PV

Tabela 6. ***Baza pravila fazi PD regulatora***

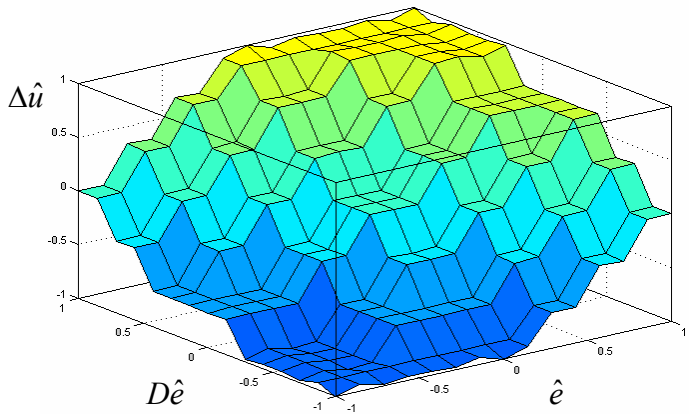
4. korak

Bira se metod defazifikacije. U literaturi se mogu naći izvesna upoređenja različitih algoritama defazifikacije u smislu njihovog uticaja na performanse kontrolera. Može da se zaključi da metod težišta daje najbolje rezultate, pa je najčešće primenjivan. Izbor metoda defazifikacije utiče na brzinu rada kontrolera. Metod težišta je numerički najnepogodniji za implementaciju. Na slici 35. prikazana je površ Δu kada se jed. (32) koristi za određivanje funkcije pripadnosti fazi zaključka i kada se metod težišta koristi kao metod defazifikacije.



Slika 35. ***Površ $\Delta\hat{u}$ za Gausove funkcije pripadnosti primarnih fazi skupova i metod težišta kao metod defazifikacije***

Na slici 36. prikazana je površ $\Delta\hat{u}$ kada se jed. (32) koristi za određivanje funkcije pripadnosti fazi zaključka i kada se metod srednje vrednosti maksimuma primeni kao metod defazifikacije.
Metode levog, desnog i srednjeg maksimuma su pogodni za implementaciju, ali u slučaju primene ovih metoda male promene ulaznih promenljivih izazivaju velike skokovite promene upravljačkog signala kao što se vidi na slici 36.



Slika 36. ***Površ $\Delta\hat{u}$ za Gausove funkcije pripadnosti primarnih fazi skupova i metod srednje vrednosti maksimuma kao metod defazifikacije***

Pri projektovanju fazi kontrolera ne postoji opšta procedura za fazi particioniranje ulaznog i izlaznog prostora, izbor funkcija pripadnosti primarnih fazi skupova, izbor normalizujućih pojačanja, definisanje baze pravila, izbor metoda defazifikacije. Zadovoljavajući izbor podrazumeva metodu "probe i greške" i nema garancije da ostvaren zadovoljavajući rezultat predstavlja optimalno rešenje.

4.3.5. Ciljevi i mogući zahtevi laboratorsijske vežbe

Cilj ove vežbe je projektovanje fazi PI regulatora za regulaciju nivoa tečnosti u rezervoaru.

Zadaci za studente:

1. Prema postupku koji je objašnjen u ovom poglavlju izvršiti sintezu fazi PI regulatora za regulaciju nivoa tečnosti u rezervoaru. U dodatku A prikazan je način sinteze fazi kontrolera u MATLAB-u.
2. Menjati normalizujuća pojačanja i doneti zaključke o njihovom uticaju na performanse sistema.
3. Ispitati uticaj promene broja lingvističih vrednosti ulaznih promenljivih i izlazne promenljive na performanse sistema. Uzeti da je broj fazi skupova za sve promenljive 5 i definisati novu bazu pravila fazi regulatora.
4. Ispitati uticaj promene oblika i rasporeda funkcija pripadnosti.
5. Koristiti jed. (32) i (33) za određivanje funkcije pripadnosti zaključka

6. Primeniti različite metode defazifikacije. Kako utiču različite metode defazifikacije na rad kontrolera i na ponašanje sistema?
7. Ispitati robustnost fazi regulatora na manje i veće promene parametara procesa.

5.
Akvizicija signala i upravljanje u realnom vremenu

Eksperimentalna aparatura je opisana u poglavlju 2.1. Za akviziciju signala i upravljanje se koristi PC računar sa NI USB 6009 modulom u funkciji AD/DA interfejsa. Poznavanje korišćenog AD/DA interfejsa (NI USB 6009) je neophodno za sitezu softvera koji implementira željeni algoritam upravljanja, akviziciju signala od interesa, i odgovarajući korisnički interfejs koji može uključiti i vizuelizaciju prikupljenih signala. NI USB 6009 je detaljnije opisan u [21]. Radi se o proizvodu globalne firme National Instruments koja obezbeđuje i softversku podršku za svoj proizvod (programski paket LabView, ali i druge softverske alate), kao i biblioteke programskih funkcija koje omogućavaju programiranje u različitim programskim jezicima. I jedan i drugi princip programiranja će biti predstavljeni u ovoj glavi.

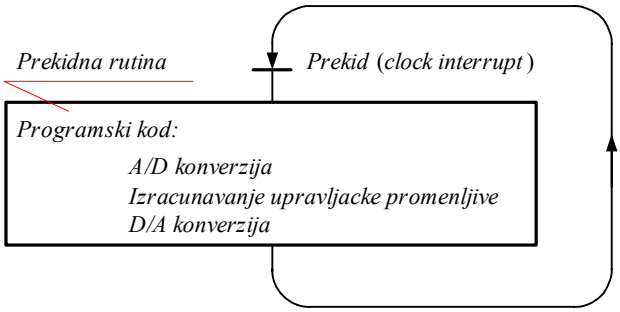
5.1. SISTEMI ZA AKVIZICIJU I UPRAVLJANJE – PRINCIPI PROGRAMIRANJA

Bez obzira koji se računarski hardver, softverski pristup i/ili alat koristi za prikupljanje (akviziciju) signala i upravljanje, postoje tri ključna pitanja koja treba rešiti pri realizaciji real-time softverske aplikacije

1. Kako se setuje željena izlazna vrednost signala D/A konvertora?
2. Kako se očitava signal sa ulaza A/D konvertora?
3. Kako se setuje željena perioda odabiranja? Ili, kako implementirati periodične „task“ – ove kontrolera u skladu sa specificiranom periodom odabiranja? (Detaljniju teorijsku podlogu o računarskim aspektima merenja i upravljanja potražiti u [10].)

Implementacija digitalnog algoritma upravljanja je ilustrovana na slici 37 [10]. U trenucima odabiranja, koji su definisani periodom odabiranja T_s, izvršava se A/D konverzija signala izmerenih upravljanih vrednosti dovedenih na ulaz računara, i D/A konverzija upravljačkih signala. Između trenutaka odabiranja izvršava se

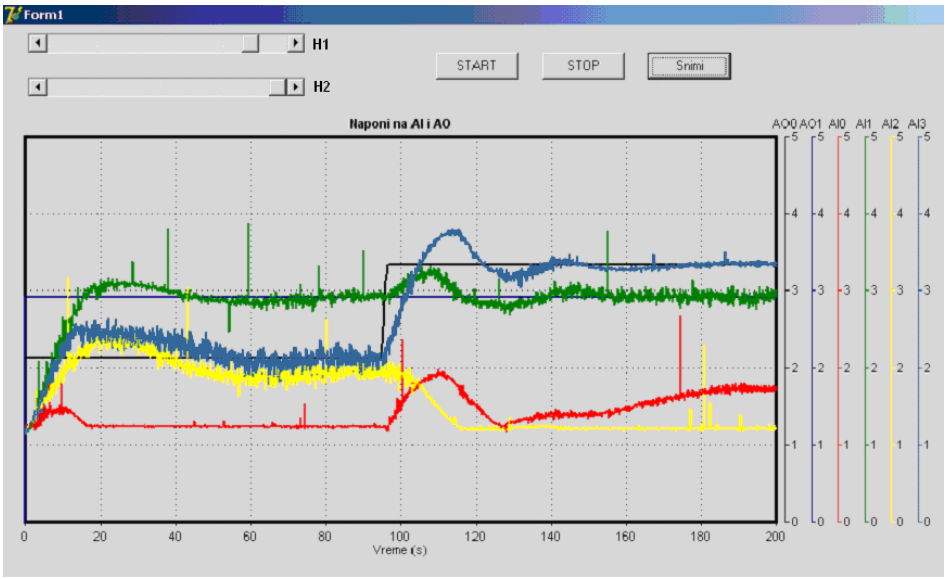
algoritam upravljanja prema kome se sračunavaju vrednosti upravljačkih signala koje će biti prosleđene na odgovarajući D/A konvertor. Dakle, tačno na svakih T_s sekundi se pokreće prekidna rutina u okviru koje se očitavaju signali A/D konvertora, sračunavaju upravljačke promenljive, šalju na ulaze D/A konvertora, pri čemu se D/A konverzija izvršava u narednom trenutku odabiranja. Pisanjem upravljačkog softvera ostvaruje se veza između senzora, aktuatora, i operatera (ljudi, korisnika), pri čemu algoritmu upravljanja mogu biti pridružene i dopunske funkcije filtriranja signala, nadzora i dijagnostike sistema, itd..



Slika 37. Implementacija digitalnog algoritma upravljanja

Na slici 38 je predstavljen korisnički interfejs softverske aplikacije za merenje i upravljanje nivoom donjih rezervoara eksperimentalne aparature opisane u poglavlju 2.1, u slučaju kada su svi ručni ventili potpuno otvoreni (radom pumpi se ostvaruje punjenje sva četiri rezervoara, a reguliše željeni nivo u donja dva). Radi se o DELPHI programu koji se realizuje na lokalnom računaru koji radi pod operativnim sistemom Windows XP. Listing programa je dat u dodatku B, dok su ključni delovi programa objašnjeni u ovom poglavlju. Principi programiranja su isti bez obzira na platformu na kojoj se radi. Implementiran je PI zakon upravljanja u slučaju regulacije nivoa u oba donja rezervoara. Na grafiku korisničkog interfejsa se on-line beleže signali nivoa u oba gornja rezervoara (AI0 i AI2), oba donja rezervoara (AO1 i AI3 koje je putem istog interfejsa moguće zadati i menjati i u toku eksperimenta), i upravljački signali za obe pumpe (AO0 i AO1). Eksperiment se pokreće pritiskom na dugme „START”, zaustavlja pritiskom na dugme „STOP”, a nizovi prikupljenih signala se po pritisku na dugme „SNIMI” upisuju u tekstualnu datoteku. Perioda odabiranja je T_s =100 ms. Perioda odabiranja, parametri algoritma upravljanja i sam algoritam upravljanja menjaju se izmenom koda programa. Saglasno karakteru korišćenog računarskog hardvera (NI USB 6009), tj. softverskoj podršci koju pruža National Instruments kao proizvođač, program koristi NI DAQmx 8.0 biblioteku sa drajverima. Sve funkcije iz te biblioteke prepoznaju se u listingu programa po tome što im naziv počinje slovima DAQmx.

Na slici 38. je uočljiv šum na signalima merenih visina nivoa u rezervoarima usled talasanja površine tečnosti, a koje nastaje usled ulivanja tečnosti u rezervoar.



Slika 38. ***Softverska aplikacija i izvršenje eksperimenta***

Sledi komentar ključnih linija koda koji reguliše procese A/D i D/A konverzije i setovanje željene periode odabiranja.

```
// Definicija globalnih promenljivih
Var
// Promenljive koji sluze za komunikaciju sa hardverom
  AIdata : Array[0..31] of Double;
  AOdata :  Array[0..31] of Double;
// Bafer u koje se pakuju podaci iz eksperimenta
  Rs : Array[0..10000, 0..1] of Double;      // Referentne vrednosti
  AIs : Array[0..10000, 0..7] of Double;     // Analogni ulazi
  AOs : Array[0..10000, 0..1] of Double;     // Analogni izlazi
// Perioda odabiranja, proteklo vreme, pocetno vreme, sledeci trenutak
// uzorkovanja i krajnje vreme
  dt, t, t0, ts, tend : double;
                                              // Redni broj periode odabiranje
  SampleNo : LongInt;
                                             // Flag da se omogucava akvizicija
  DoACQ : Boolean;
                                             // Promenljive regulatora
  r1, r2, ki1, ki2, kp1, kp2, e1, e2, i1, i2, u1, u2 : Double;
```

Sledeći segment programa objašnjava kako se podatak šalje na D/A konvertor:

```
// Upisivanje jednog uzorka na kanale DA0 i DA1 u opsegu 0V do 5V
// Vrednosti koje se upisuju se nalaze u globalnom nizu Aodata
procedure TForm1.WriteAO;
var
  hdl  : Integer;
  p : Pointer;
  ret : LongInt;
  sampsPerChanWriten : LongInt;
begin
  ret := DAQmxCreateTask('Analog Output', @hdl);
```

```
  ret := DAQmxCreateAOVoltageChan(hdl, 'Dev1/ao0:Dev1/ao1', '', 0.0, 5.0,
DAQmx Val VoltageUnits1 Volts, '');
  p := Nil;
  ret := DAQmxWriteAnalogF64(hdl, 1, -1, 10, DAQmx_Val_GroupByScanNumber,
@AOdata, @sampsPerChanWriten, p);
  ret := DAQmxWaitUntilTaskDone(hdl,10);
  ret := DAQmxStopTask(hdl);
  ret := DAQmxClearTask(hdl);
end;
```

Sledeći segment programa objašnjava kako se signal preuzima sa A/D konvertora:

```
// Citanje jednog uzorka sa kanala AD0 do AD7. Ocitane vrednosti su u opsegu
// -10V do 10V
// Vrednosti koje se citaju smestaju se u globalni niz AIdata
procedure TForm1.ReadAI;
var
  hdl  : Integer;
  p : Pointer;
  ret : LongInt;
  sampsPerChanRed : LongInt;
begin
  ret := DAQmxCreateTask('Analog Input', @hdl);
  ret := DAQmxCreateAIVoltageChan(hdl, 'Dev1/ai0:Dev1/ai7', '',
DAQmx Val InputTermCfg RSE,  -10.0, 10.0, DAQmx Val VoltageUnits1 Volts, '');
  p := nil;
  ret := DAQmxReadAnalogF64(hdl, 1, 10, DAQmx_Val_GroupByScanNumber, @AIdata,
10, @sampsPerChanRed, p);
  ret := DAQmxWaitUntilTaskDone(hdl,10);
  ret := DAQmxStopTask(hdl);
  ret := DAQmxClearTask(hdl);
end;
```

Sledeći segment programa objašnjava kako se setuje perioda odabiranja T_s=100 ms i realizuje algoritam upravljanja. Perioda tajmera je 10 ms, tj. svakih 10 ms proverava se da li je nastupio trenutak za akviziciju podataka. Ako jeste, računa se kada nastupa trenutak za sledeću akviziciju, prikupljaju se podaci, izvršava regulator i izdaje upravljanje.

```
procedure TForm1.Timer1Timer(Sender: TObject);
var
  i : LongInt;
begin
// Perioda tajmera je dosta kraca od periode uzorkovanja. Kod svakog
// okidanja tajmera proverava se da li je dosao trenutak za uzorkovanje.
  if Now < ts then Exit;
  Inc(SampleNo);                // Inkrementiraj brojac prikupljenih uzoraka
  ts := t0+dt*SampleNo/(24*3600);  // Izracunaj sledeci trenutak uzorkovanja
  t := (ts-t0)*(24*3600);    // Izracunaj vreme trajanja eksperimenta u sek.

        // Prekini eksperiment ako je isteklo vreme ili se rucno zaustavlja
  if (t > tend) Or Not DoACQ then
    begin
      ResetAO;
      Timer1.Enabled := False;
```

```
    Exit;
   end;
                                                          // Procitaj ulaze
  ReadAI;
                                                               // Regulator
  kp1 := 2;
  ki1 := 0.5;
  kp2 := 2;
  ki2 := 0.5;
                                                    // Referentne vrednosti
  r1 := 1.2+ScrollBar1.Position/200;
  r2 := 1.2+ScrollBar2.Position/200;
                                                          // Signali greske
  e1 := r1 - AIdata[1];
  e2 := r2 - AIdata[3];
                                             // Integeratori
  i1 := i1+dt*ki1*e1;
  i2 := i2+dt*ki2*e2;
                                                               // Antiwindup
  if kp1*e1+i1 > 5 then i1 := 5-kp1*e1;
  if kp1*e1+i1 < 0 then i1 := -kp1*e1;

  if kp2*e2+i2 > 5 then i2 := 5-kp2*e2;
  if kp2*e2+i2 < 0 then i2 := -kp2*e2;
                                                 // Kompletiranje upravljanja
  u1 := kp1*e1+i1;
  u2 := kp2*e2+i2;
          // Upisi vrednosti u niz koji ce biti prosledjen na DA konvertore
  AOdata[0] := u1;
  AOdata[1] := u2;
                                                    // Upis na DA konvertore
  WriteAO;
                             // Nacrtaj podatke iz tekuce periode odabiranja
  AI0.DataDraw(Image.Canvas, t, AIdata[0]);
  AI1.DataDraw(Image.Canvas, t, AIdata[1]);
  AI2.DataDraw(Image.Canvas, t, AIdata[2]);
  AI3.DataDraw(Image.Canvas, t, AIdata[3]);
  AO0.DataDraw(Image.Canvas, t, r1{AOdata[0]});
  AO1.DataDraw(Image.Canvas, t, r2{AOdata[0]})
                       // Sacuvaj podatke iz tekuce periode odabiranja
  Rs[SampleNo, 0]  := r1;
  Rs[SampleNo, 1]  := r2;
  for i := 0 to 7 do AIs[SampleNo, i] := AIdata[i];
  for i := 0 to 1 do AOs[SampleNo, i] := AOdata[i];
end;
```

5.2. OSNOVE KORIŠĆENJA PROGRAMSKOG PAKETA LABVIEW

LabVIEW (LV) predstavlja izuzetno moćno razvojno okruženje za vizuelno programiranje kompletnih aplikacija za razne vrste merenja, upravljanja, analize rezultata merenja, generisanje raznih vrsta mernih izveštaja kao i za razne druge funkcije koje su u neposrednoj ili posrednoj vezi sa merenjima.

LV podržava izuzetno širok spektar programabilne merne opreme i uređaja za koje se mogu razvijati merne i upravljačke aplikacije. Firma National Instruments sa sedištem u Austin-u, Texas (USA) već 30 godina (2006) razvija merne i druge

programabilne uređaje vrhunskog kvaliteta i 20 godina razvija softver za merenja LV. U osnovi LV-a se nalazi nekoliko osnovnih koncepata koji daju osnovnu snagu jedinstvenom spoju mernog hardvera i softvera za podršku merenjima. Osnovni koncept je tzv. Virtuelni Instrument (Virtual Instrument VI). VI predstavlja softverski instrument koji ima određene funkcionalnosti i sa kojim se upravlja preko upravljačke forme – front panel koji sa jedne strane predstavlja softverski analog prednje ploče klasičnih hardverskih instrumenata, a sa druge strane to je forma sa veoma sličnim konceptom kao što su forme u raznim Windows aplikacijama. To znači da se VI instrumenti kreiraju kao softverske tvorevine, i da se samim tim veoma lako menjaju, modifikuju prema potrebama, za razliku od klasičnih instrumenata čija se funkcionalnost daleko teže ili uopšte ne može menjati. Svrha virtuelnog instrumenta može biti upravljanje nekim hardverskim uređajem, ali ne nužno, to može biti i nešto drugo, kao na pr. obrada rezultata merenja, analiza merenja, grafički prikaz merenja ili nešto slično. Takođe postoji i veoma jednostavna mogućnost pristupa VI preko WEB-a, pri čemu je sasvim svejedno da li se sa VI radi preko WEB-a ili lokalno.

Sledeći moćan koncept je NI DAQ biblioteka koju čini veliki skup driver-a za razne programabilne uređaje kako firme NI tako i raznih drugih proizvođača, kao i funkcija i klasa koje se koriste za kreiranje VI. NI DAQ postoji u verziji Traditional za tzv. legacy instruments – instrumente starije generacije, i u verziji NI DAQmx za instrumente novije generecije.

NI DAQ je moćan koncept iz više razloga. Osim što može da se koristi iz LV-a, omogućava korišćenje funkcija i klasa direktno iz savremenih moćnih programskih jezika kao što su ANSI C, zatim .NET jezici C#, Visual Basic, C++. NI DAQ može da se koristi sa raznim operativnim sistemima kao što su Windows, Linux, Mac OS. To sve omogućava da se za razvoj softvera za merenje i upravljanje odabere najpovoljniji pristup.

Funkcije i klase softverske biblioteke NI DAQ omogućavaju kreiranje VI koji mogu na potpuno isti način da rade sa različitim hardverskim uređajima koji podržavaju potrebne funkcionalnosti. To omogućava da se isti VI koristi sa drugim hardverom jednostavnom zamenom driver-a za taj hardverski uređaj. To znači da se sa hardverski različitim uređajima radi na potpuno isti način, jer sloj softvera na kome se nalazi specifični driver omogućava tu hardversku nezavisnost. Ovakav koncept je potpuno analogan konceptu kod savremenih operativnih sistema (OS) gde se primenom driver-a za hardverske uređaje postiže nezavisnost OS od hardvera na kome radi. Teoretski, bilo koji OS može da radi na bilo kom računaru, sa bilo kojim periferijskim uređajima. Naravno, izuzeci postoje, ali kao specijalni slučajevi koji ne ugrožavaju sam konsept. Slično tome, postoji i nezavisnost LV-a od OS-a i hardvera.

Osnovni elementi LabVIEW-a

U radu sa razvojnim okruženjem LV postoje dva osnovna pogleda Front Panel (FP) – forma, korisnički interfejs i Block Diagram (BD) na kome se vrši povezivanje

elemenata. Svakom VI odgovara jedan FP i jedan BD. BD sadrži grafički kod koji se još naziva i G jezik, jezik za grafičko programiranje. U grafičkom programiranju se koriste tri osnovne vrste elemenata i to su:

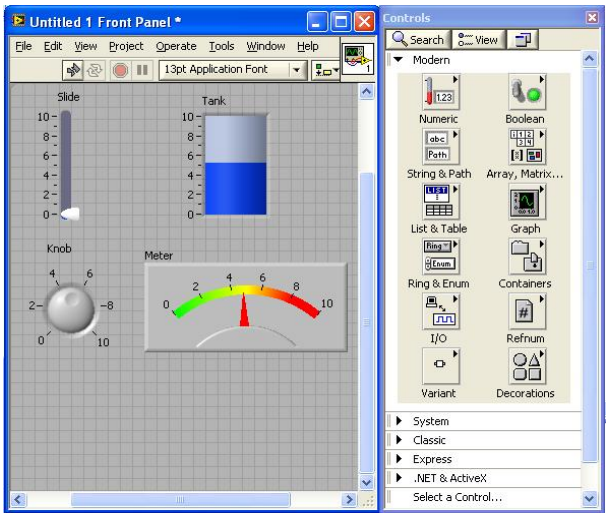
- Kontrole - izvori signala, izlazni elementi
- Indikatori – prijemnici signala, ulazni elementi
- Funkcije poseduju ulaze i izlaze – ulazno izlazni elementi

Ovi elementi su na BD povezani sa vezama – wires (žice) koje u stvari predstavljaju tokove podataka strogo određenih tipova. Sa svakom vezom se može povezati samo jedan izlaz sa jednim ili više ulaza istog tipa. Ukoliko se pokuša povezivanje dva ulaza, dva izlaza ili ulaza i izlaza različitih tipova podataka, dobija se isprekidana veza sa znakom x na sredini. Stavljanjem strelice na tu liniju dobija se tool tip – pravougaonik sa tekstualnim opisom greške. Greška se dobija i kada se sa jednim ulazom poveže više izlaza.

FP može sa sadrži kontrole i indikatore, dok se funkcije mogu nalaziti samo na BD.

Kada je aktivan FP, tj. kada se prikazuje FP, onda se može prikazati i paleta sa kontrolama.

Na slici 39 je prikazana jednostavna forma – FP sa paletom kontrola sa desne strane. FP je prikazana u tzv. design modu kada se vidi mreža, i kada VI ne radi već se može menjati – editovati. Sa desne strane se nalazi paleta sa kontrolama koja sadrži više različitih grupa kontrola: Modern, System, Classic, Express, .NET & ActiveX, kao i mogućnost selektovanja pojedinačnih kontrola.



Slika 39. ***Jednostavna forma i paleta kontrola***

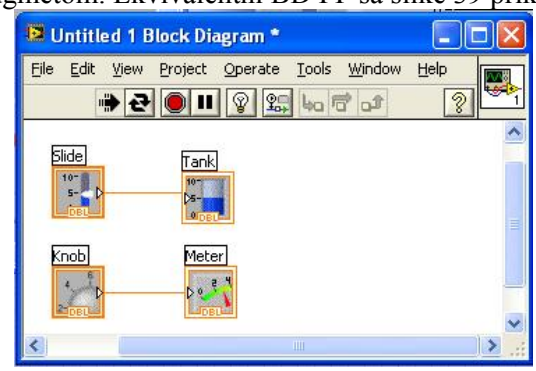
Na slici 39 otvorena je grupa „Modern“, i vidi se niz ikona koje predstavljaju podgrupe. Na slici 40 prikazan je sadržaj podgrupe „Numeric“ kojoj pripadaju kontrole na FP na slici 39

Slika 40. ***Numeričke kontrole***

Primer za jednostavan FP VI je dat na slici 39. Mreža ukazuje da se radi o design modu. U izvršni mod se prelazi pritiskom na strelicu u toolbar-u koja je posebno prikazana na slici 41.

- Slika 41 ***Dugmad za startovanje i zaustavljanje aplikacije***

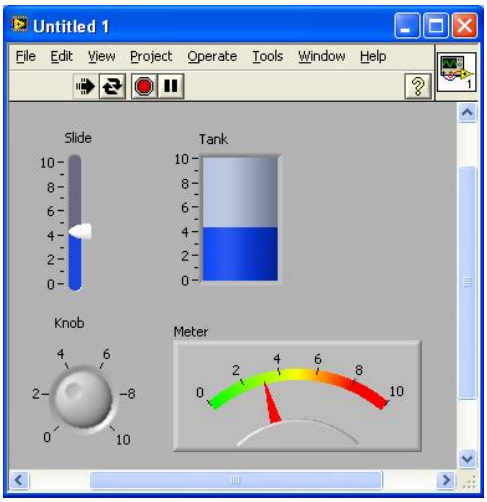
Aplikacija se startuje pritiskom na strelicu, i ukoliko nije definisana petlja, tj. ponavljanje, odmah posle izvršavanja aplikacija se završava i ponovo se prelazi u design mod. Pritiskom na dve strelice koje idu u krug, aplikacija se automatski ponavlja, tj. nalazi se u petlji, koja se prekida sa crvenim dugmetom, a pauzira sa krajnjim desnim dugmetom. Ekvivalentni BD FP sa slike 39 prikazan je na slici 42.



Slika 42. ***BD za FP sa slici 39.***

Na BD-u su kontrole prikazane sa ikonama i povezane su sa vezama. „Slide“ i „Knob“ su kontrole, dok su „Tank“ i „Meter“ indikatori. Na slici 43 je prikazan primer kada se VI nalazi u izvršnom modu. Treba uočiti da su položaji kontrola i odgovarajućih indikatora za koje su vezani usaglašeni, tj. da pokazuju iste vrednosti.

Merenje sa LabVIEW-om



Slika 43. ***VI u izvršnom modu***

Jednostavan VI prethodno prikazan nije povezan ni sa kakvim mernim uređajem – hardverom. U eksperimentu sa rezervoarima se mere nivoi tečnosti u četiri rezervoara, upravlja se radom dve pumpe i zadaju se nivoi u rezervoarima koje treba dinamički održavati pomoću kontrolera. Za to se koriste odgovarajuće kontrole i funkcije koje omogućavaju povezivanje sa sistemom za akviziciju i upravljanje (AD/DA interfejsom), jer se pomoću sistema za akviziciju vrši merenje nivoa rezervoara kao i upravljanje pumpama.

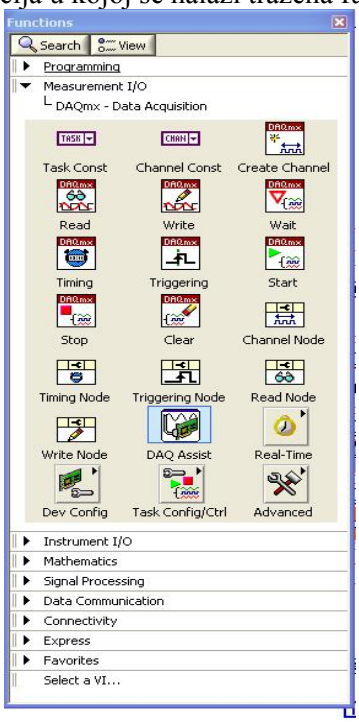
Za merenje nivoa vode u rezervoarima koriste se davači – merni konvertori koji hidrostatički pritisak vode prevode u električni napon. Veza izmerenog napona i visine vodenog stuba je linearna, i data je sa:

$$u = k\rho gh = k_1 h$$

u je izmereni napon sa davača za pritisak, a k je konstanta proporcionalnosti dok je $k_1 = k\rho g$ nova konstanta. Vrednost za k_1 se dobija na osnovu merenja da za h = 30 cm izmereni napon u iznosi 3 V. Odatke sledi da je k_1=3V/30 cm=0.1 V/cm ili u jedinicama SI k_1=10 V/m. Određivanjem vrednosti za k_1 vrši se kalibracija davača za pritisak čime se na osnovu izmerenog napona može izračunati visina vode u rezervoaru.

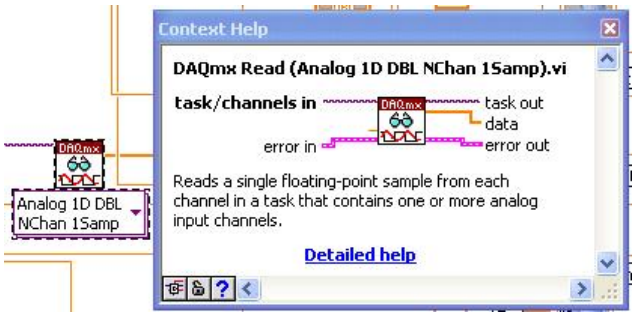
Vrednost napona se meri analognom sekcijom sistema za akviziciju NI USB 6009. Analogna sekcija ima 4 nezavisna ulazna kanala. Da bi izmerene vrednosti bile dostupne u VI neophodno je povezati hardver – sistem za akviziciju sa VI. To se postiže korišćenjem kontrole „Task“ koja se nalazi malo dublje u hijerarhijskoj strukturi kontrola u LB – „**Controls/Modern/IO/DAQmx Name Controls/Task“**. „Task“ je termin NI vezan za DAQmx koji predstavlja grupu virtuelnih kanala u ovom slučaju 4 virtuelna kanala koji odgovaraju fizičkim kanalima sistema za akviziciju. Izraz virtuelni kanal je takođe termin NI koji označava fizički kanal sa odgovarajućim karakteristikama kao što su opseg, jedinice. „Task“ kontrola se selektovanjem u combo box-u vezuje za odgovarajući task koji treba da bude

unapred kreiran. Ukoliko nije, može se kreirati iz same kontrole. Da bi se kreirao novi task, potrebno je odabrati odgovarajuću sekciju sistema za akviziciju i kanale u toj sekciji. Za merenje nivoa rezervoara potreban je task sa sva 4 kanala ulazne analogne sekcije. „Task“ kontrola nije dovoljna za merenje, već je neophodno koristiti još i neke funkcije. Funkcije se koriste za transformisanje podataka iz jednog oblika u drugi. Za dobijanje izmerenih vrednosti iz kontrole „Task“ koristi se funkcija „Read“ koja se nalazi u paleti funkcija „**Functions/Measurement IO/DAQmx Data Acquisition/Read“**. Na slici 44 prikazana je paleta sa funkcijama sa otvorenom grupom funkcija u kojoj se nalazi tražena funkcija „Read“.



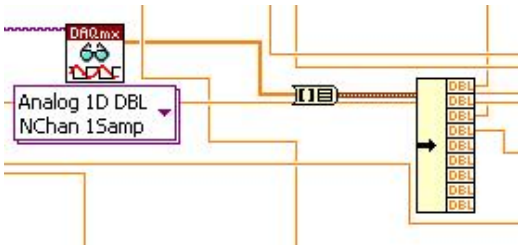
Slika 44. ***Paleta sa funkcijama***

Paleta sa funkcijama je vidljiva isključivo u pogledu na BD, dok je paleta sa kontrolama vidljiva isključivo u pogledu na FP. Na slici 45 je prikazana funkcija „Read“ zajedno sa prozorom „Context help“ koji prikazuje ulaze i izlaze ove funkcije.



Slika 45. ***Funkcija Read***

Ulaz „task/channels in“ se povezuje sa kontrolom „Task“, dok se na izlazu data pri svakom merenju dobija niz od 4 izmerene vrednosti. Da bi se iz niza dobile pojedinačne izmerene vrednosti na svakom kanalu, niz (array) se konvertuje u cluster. I „cluster“ i „array“ su kolekcije vrednosti, ali je array uvek kolekcija istog tipa, dok „cluster“ može biti kolekcija vrednosti različitih tipova. Funkcija „Array to Cluster“ konvertuje niz u klaster, i nalazi se u „**Functions/Programming/ Array/Array to Cluster“,** dok se klaster grana u pojedinačne vrednosti sa funkcijom „Unbundle“ koja se nalazi u „**Functions/Programming/Cluster & Variant/Unboundle“.** Na slici 46 prikazane su s leva na desno funkcije „read“, „Array to Cluster“ i „Unboundle“ koja ima 8 izlaza, od kojih svaki odgovara po jednom analognom ulaznom kanalu.

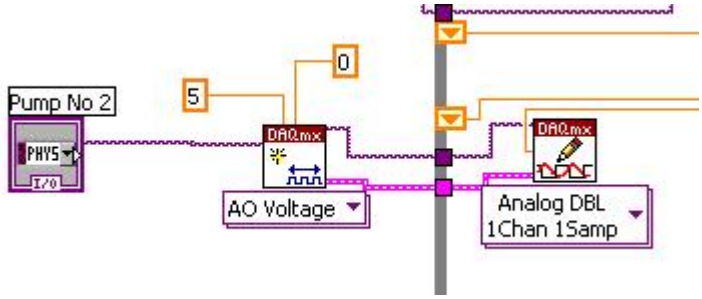


Slika 46. ***Dobijanje pojedinačnih izmerenih vrednosti sa analognih ulaza***

Na svakom od 4 gornja izlaza funkcije „Unbundle“ dobija se po jedna vrednost napona izmerena sa odgovarajućeg analognog kanala i predstavlja trenutnu vrednost analognog napona na davačima za pritisak. Ove vrednosti se dalje obrađuju u bloku kojim se implementira kontroler o čemu će kasnije biti više reči.

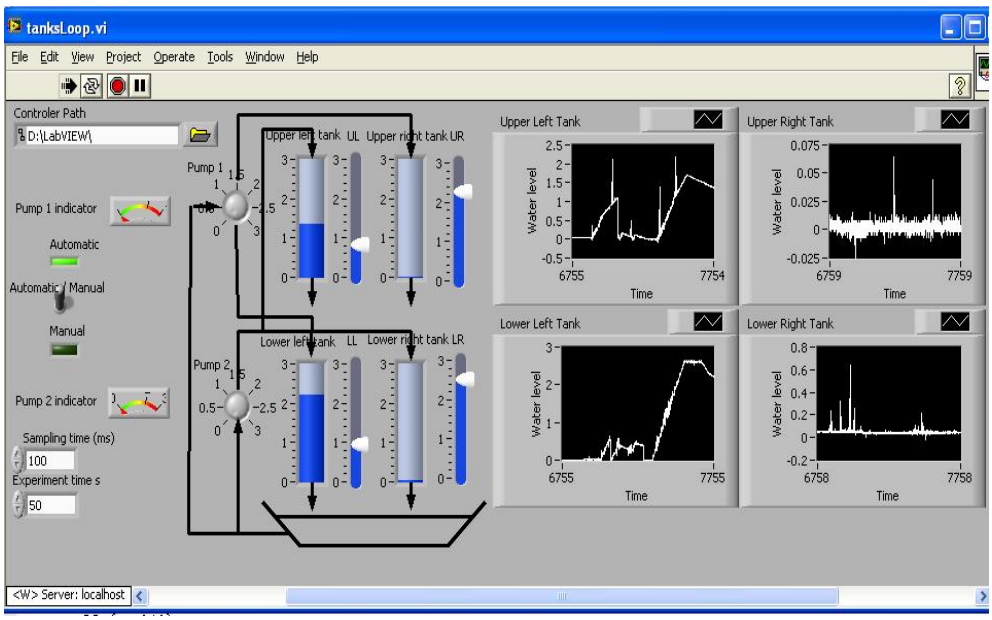
Za upravljanje sa električnim pumpama je potrebno generisati analogni napon određene vrednosti. Slično prethodno pomenutim kontrolama koje se koriste za merenje napona, koriste se odgovarajuće kontrole i funkcije. Kontrola koja se koristi za generisanje analognog napona je „Physical Channel“ koja se nalazi u „**Controls/Modern/IO/DAQmx Name Controls/Physical Chan**“**.** Potpuno ista kontrola se može koristiti i za merenje napona, s tim što se preko padajućeg menija vezuje za odgovarajući analogni ulaz. Razlog da ova kontrola nije iskorišćena za merenje je taj da je bilo potrebno istovremeno meriti više napona a ne samo jedan, pa je upotrebljena kontrola „Task“. Iako je potrebno upravljati sa dva analogna napona, suštinska razlika u odnosu na merenje je da za merenje iako postoji 8 ulaznih kanala, fizički postoji samo jedan A/D konvertor koji se koristi za merenje vrednosti analognih napona na svim kanalima, dok fizički postoje dva odvojena D/A konvertora koji se kontrolišu sa po jednom kontrolom „Physical Channel“. Na slici 47 prikazana je kontrola „Physical Channel“ koja je označena sa „Pump No 2“ jer je povezana sa analognim izlazom koji kontroliše pumpu broj 2, zajedno sa odgovarajućim funkcijama koje su neophodne za generisanje napona. Izlaz sa kontrole se vodi na ulaz funkcije „**DAQmx Create Virtual Channel“** koja generiše virtuelni kanal. Kreirani virtueni kanal se dalje vodi na ulaz sledeće funkcije „Write“ koja se koristi za upisivanje željene vrednosti na na analogni izlaz. Obe funkcije su u istoj grupi „**Functions/Measurement IO/DAQmx Data Acquisition“.** Na slici 47

sleva na desno su prikazane kontrola „Physical Cannel“, funkcija „Create Virtual Channel“ i funkcija „Write“.



Slika 47 ***Kontrola „Physical Cannel“, i funkcije „Create Virtual Channel“ i „Write“***

Na ulaz funkcije „Write“ dovodi se vrednost koju treba upisati na analogni izlazni kanal. Ovavrednost se dobija sa kontrolera u slučaju automatskog upravljanja ili sa numeričke kontrole u slučaju manuelnog upravljanja. Manuelno upravljanje sistemom rezervoara se vrši podešavanjem analognog napona kojim se upravlja sa pumpama koje pune rezervoare sa vodom. FP VI za upravljanje sistemom rezervoara prikazan je na slici 48. Eksperiment na slici 48. je urađen u otvorenoj povratnoj sprezi (manuelno upravljanje) u trenutku kada je ručnim ventilima sprečeno oticanje vode u desne rezervoare.



Slika 48. ***FP VI za upravljanje sistemom rezervoara***

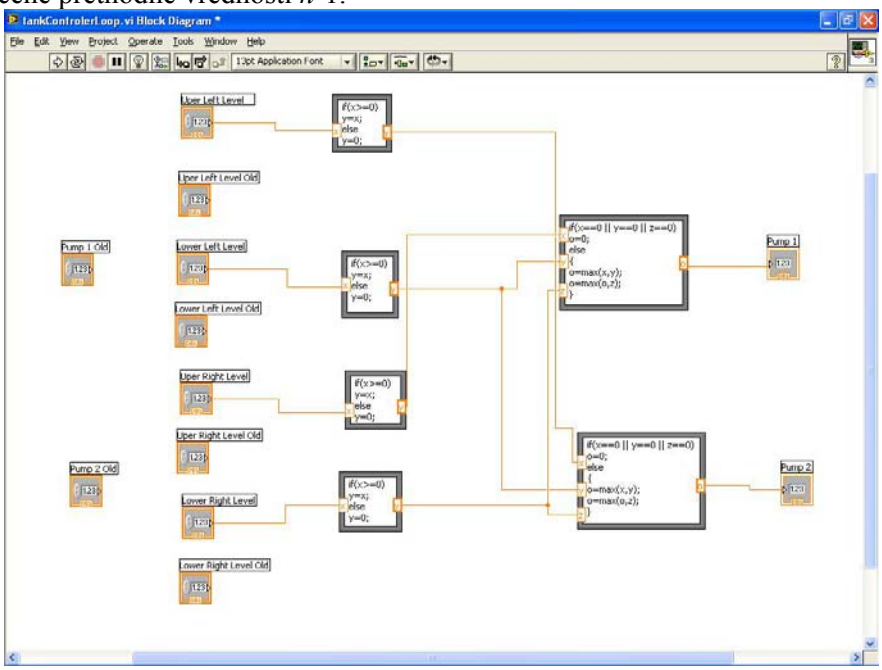
Preklopnik Automatic/Manual na slici 48 služi za izbor manuelnog ili automatskog upravljanja. Ispod i iznad preklopnika se nalaze i svetlosni indikatori trenutnog stanja. Kada je preklopnik u donjem položaju – Manual, onda se okretanjem kontrola „Pump 1“ i „Pump 2“ može kontinualno upravljati odgovarajućim

pumpama. Indikatori u vidu rezervoara pokazuju trenutne vrednosti nivoa u rezervoarima koji su na eksperimentalnoj aparaturi prostorno raspoređeni na isti način. Crnim linijama sa strelicama su predstavljene putanje tečnosti – vode kroz eksperimentalni sistem. Sa slike se vidi da pumpe pune dijagonalne rezervoare, kao i da se gornji rezervoari prazne oticanjem tečnosti u donje rezervoare. Tečnost iz donjih rezervoara se uliva u sabirni rezervoar odakle se pumpama rezervoari ponovo pune, što znači da je tok vode u sistemu zatvoren. Usled isparavanja vode i kvašenja zidova suda, ipak je neophodno povremeno dolivati vodu da bi se obezbedila minimalna količina za odvijanje eksperimenta, a treba voditi računa i o tome da pumpe nikada ne smeju biti suve u toku rada. FP VI na slici 48 omogućava manuelno upravljanje i bez neposredne vizuelne veze sa stvarnim rezervoarima, jer su rezervoari indikatori kalibrisani tako da proporcionalno prikazuju stvarni nivo vode u odgovarajućem rezervoaru. To takođe omogućava i realizovanje vežbe putem interneta kada se isti prikaz kao na slici 48 ima i u prozoru browser-a. To znači da za remote verziju ovog eksperimenta nije neophodna WEB kamera za vizuelizaciju procesa. Kontrole sa belim strelicama služe za zadavanje nivoa koje treba održavati u automatskom režimu rada i nisu bitne u manuelnom režimu upravljanja. Sa desne strane se nalaze 4 grafička prikaza promene nivoa tečnosti u odgovarajućim rezervoarima tokom vremena. Dva text box-a u donjem levom delu služe za zadavanje periode odabiranja – „Sampling time“, tj. intervala vremena između dva uzastopna merenja, kao i podešavanje trajanja eksperimenta – Experiment time u s. Ukoliko je potrebno ranije prekinuti eksperiment, to se jednostavno postiže pritiskom na crveno dugme, ili na pauzu u „tool“ baru.

Kontroler za automatsko upravljanje sistemom rezervoara

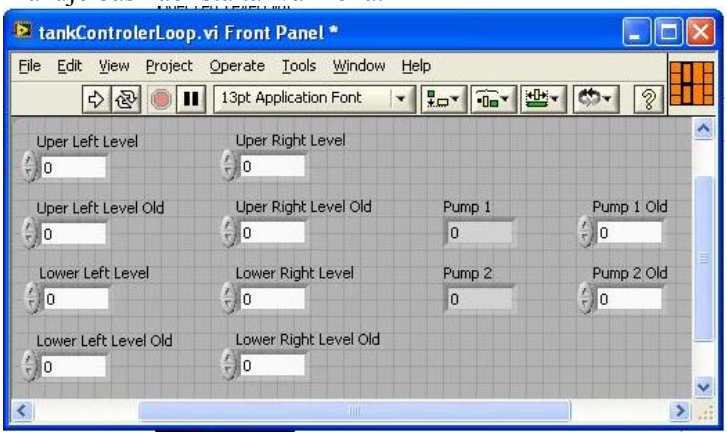
Upravljanje bilo kojim sistemom može se vršiti u otvorenoj i zatvorenoj povratnoj sprezi. Kako samo upravljanje u zatvorenoj povratnoj sprezi može da omogući adekvatno upravljanje u prisustvu poremećaja koji se ne mogu predvideti, kontroler za sistem rezervoara u ovom eksperimentu se realizuje u zatvorenoj povratnoj sprezi. Osnovna uloga kontrolera je da omogući održavanje nivoa tečnosti na zadatom nivou u rezervoarima koristeći dve pumpe kojima se rezervoari pune. Za regulisanje nivoa tečnosti u rezervoarima mogu se koristiti razne vrste kontrolera, tako da je shodno tome VI za ovaj instrument dizajniran tako da kontroler koji će se koristiti nije fiksiran, već je predviđena implementacija kontrolera kao posebnog VI, tj. sub VI koji predstavlja izmenljivi modul u okviru VI za upravljanje eksperimentom. VI koji predstavlja kontroler za ovaj eksperiment mora da ima standardizovan interfejs kojim se povezuje sa ostatkom sistema. Bilo koji VI koji ima zahtevani interfejs se može dinamički povezati sa ostatkom sistema koji obezbeđuje radno okruženje za kontroler. Pod dinamičkim povezivanjem se podrazumeva izmena modula kontrolera bez potrebe restartovanja razvojnog i izvršnog okruženja LabVIEW. Odabrani, prethodno implementiran kontroler u vidu VI sa zahtevanim interfejsom se učitava kao VI file iz FP VI preko kontrole za izbor VI pod nazivom „Controler path“ u gornjem levom delu slike 48. Osnovni zadatak radnog okruženja je da dovede sve potrebne signale za rad kontrolera, kao i da prihvati signale koje kontroler generiše, a koji se koriste za upravljanje pumpama.

Za eksperiment sa rezervoarima signali koji se dovode na kontroler su trenutne vrednosti nivoa tečnosti u svim rezervoarima, zatim zahtevane vrednosti koje treba održavati na svim rezervoarima, prethodne vrednosti nivoa tečnosti iz prethodnog merenja, kao i prethodne vrednosti signala upravljanja. Trenutne vrednosti signala se mogu obeležiti sa n, dok su prethodne vrednosti u tom slučaju n-1. Broj signala koji se dovodi na kontroler se može smanjiti ako se dovode samo razlike signala, tj. signali greške $e_i(\underline{n})$ i $e_i(n\text{-}1)$ pri čemu je $e_i(n)$=željeni nivo$_i$ (n)– mereni nivo$_i$(n) indeks i označava rezervoare i=1, 2, 3, 4. Takođe se dovode i $u_j(n\text{-}1)$, prethodni signali upravljanja, j=1, 2. Kontroler treba da generiše dva signala $u_j(n)$ kojima se upravlja pumpama. To znači da kontroler treba da ima 4 ulaza $e_i(n)$, 4 ulaza $e_i(n\text{-}1)$, 2 ulaza $u_j(n\text{-}1)$, ukupno 10 ulaza i dva izlaza $u_j(n)$. Interfejs za povezivanje VI sa drugim VI se definiše analogno definisanju prototipa funkcije kod programskih jezika, C na pr. Prototip treba da odredi tačan tip podataka za svaki ulaz i za svaki izlaz. Prototip ili interfejs VI se definiše kao „connector VI“. Za svaki ulaz VI mora da postoji odgovarajuća ulazna kontrola, dok za svaki izlaz treba da postoji po jedan indikator. Te kontrole i indikatori su neophodni elemenat interfejsa, i ukoliko ne postoji potreba za tim kontrolama, moraju se kreirati za potrebe povezivanja sa ostatkom sistema. To je i urađeno kod kontrolera za nivo tečnosti, gde se kao ulazne kontrole koriste text box-ovi, dok se kao indikatori koriste numerički indikatori. Na slici 49 je prikazan BD za jednu verziju jednostavnog kontrolera. Ikone ulaznih kontrola su prikazane sa leve strane, dok su sa desne strane indikatori. Sa Old su označene prethodne vrednosti n-1.



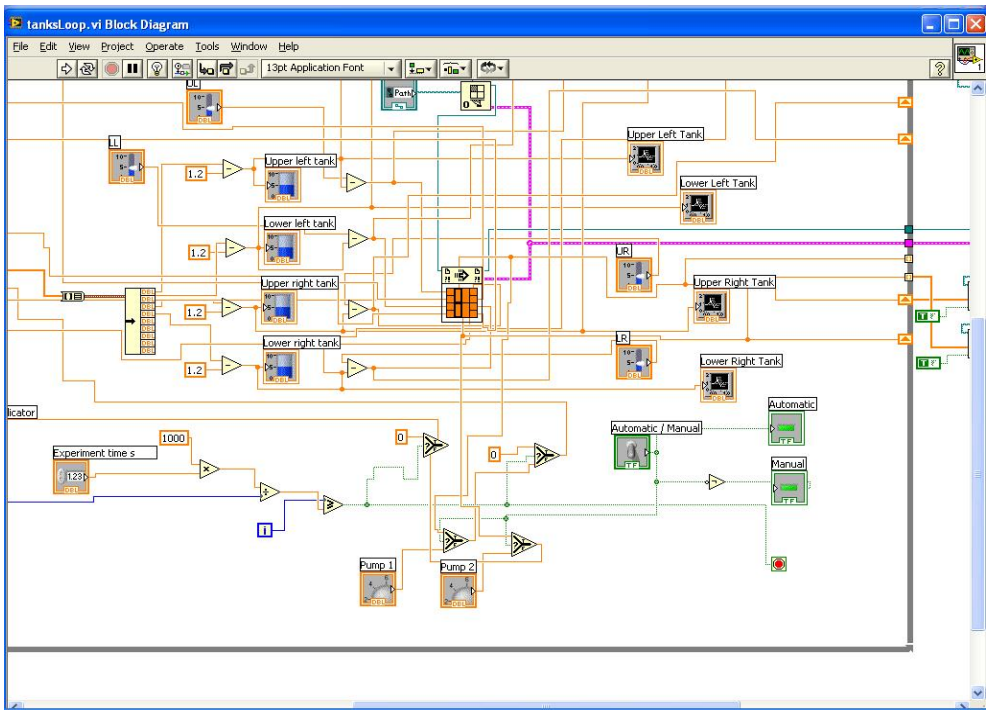
Slika 49. ***Jednostavan kontroler sa ulazima i izlazima VI (kontroler je ad hoc studentski pokušaj da se napravi inteligentan on-off kontroler radi ispravnog punjenja sva 4 rezervoara – Ideja je da se pokaže jednostavnost implementacije proizvoljnog kontrolera)***

Pravougaonici su funkcije koje omogućavaju da se unutar njih napiše kod sa sintaksom sasvim sličnom kao kod C jezika. Na slici 50 je prikazan izgled FP VI kontrolera. Vide se kontrole koje služe za povezivanje sa ostatkom sistema. U gornjem desnom uglu je umesto ikone prikazan izgled „connector-a“ koji vizuelno predstavlja interefejs VI kojim se povezuje sa nekim drugim VI. Svako polje na konektoru predstavlja po jedan signal. Kada se ovakav modul učita u neki drugi VI, na BD se prikazuje baš kao ista takva ikona.



Slika 50. ***Izgled FP VI kontrolera***

Ulazne kontrole na slici 50 su bele, dok su dva indikatora siva. Na slici 51 prikazan je deo BD VI čiji je FP prikazan na slici 48.



Slika 51. ***Prikaz dela BD VI sa slike 48***

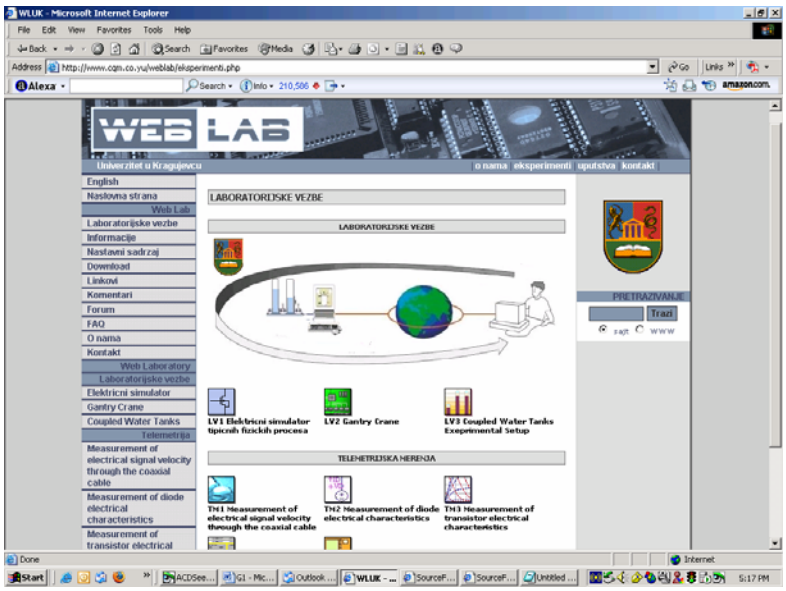
Sa slike 51 se vidi da je struktura grafičkog programa relativno složena, ali treba obratiti pažnju na narandžastu ikonu u *središnjem* delu slike. Ta ikona označava učitani kontroler koji je definisan kao sub VI preko standardnog interfejsa. Korisnik ovakvog sistema koji želi da testira kontroler koji je dizajnirao i želi da ispita ponašanje kontrolera na nekom realnom sistemu, to može da uradi tako što će da implementirani kontroler sa definisanim interefejsom jednostavno učitati u sistem kao file. Povezivanje sa ostatkom sistema okruženja se vrši automatski i korisnik o tome uopšte ne vodi računa. Radno okruženje za kontroler se jednom implementira i može se koristiti sa raznim kontrolerima koji imaju potreban standardni interfejs. Za povezivanje kontrolera sa ostatkom sistema dovoljno je učitati odgovarajući vi file preko kontrole „Controler path“ u gornjem levom uglu slike 48.

5.3. SOFTVERSKA APLIKACIJA I IZVRŠENJE EKSPERIMENTA

Aplikacija za upravljanje eksperimentom se sastoji iz pomenuta dva modula. Okruženje za merenje i upravljanje čiji je FP prikazan na slici 48 se povezuje sa kontrolerom preko odgovarajućeg definisanog interefejsa čiji je FP prikazan na slici 50. Korisnik razvija svoj kontroler kao VI, učitava ga, i povezuje sa sistemom za merenje i upravljanje. Korisnik ne mora nista da zna o tehničkim detaljima eksperimentalne aparature ili o karakteristikama sistema za akviziciju, već se jednostavno skoncentriše samo na dizajn kontrolera. Kao što je već navedeno, aparaturom se može upravljati manuelno ili automatski preko učitanog kontrolera. Ponašanje i efikasnost kontrolera se može pratiti vizuelno direktno na aparaturi, zatim preko indikatora nivoa u vidu rezervoara prikazanih na slici 48 i preko dijagrama koji beleže promene nivoa rezervoara u toku vremena. Osim toga, za vreme trajanja eksperimenta sve izmerene vrednosti nivoa tečnosti u rezervoarima kao i upravljački naponi se beleže u 6 fajlova, 4 fajla za nivoe svakog od 4 rezervoara i dva fajla za upravljačke signale svake pumpe. Sadržaj ovih fajlova sa numeričkim vrednostima se može koristiti za egzaktnu analizu ponašanja kontrolera posle izvršenja eksperimenta. Vrednosti referentnih (željenih) nivoa tečnosti u rezervoarima, i vreme su promenljive koje se takođe snimaju.

6.
Korišćenje laboratorijske vežbe putem Interneta

Osnovna ideja da se omogući pristup i da se laboratorijske vežbe mogu raditi preko Interneta je povezana sa idejom da se omogući što širi i nesmetan pristup studenata univerzitetskim laboratorijskim resursima koji su ograničeni. Na ovaj način se rešava problem pristupa laboratorijskim uređajima i omogućava se efikasnije iskorišćavanje postojećih resursa omogućavanjem daljinskog pristupa u vreme koje odgovara studentu korisniku.



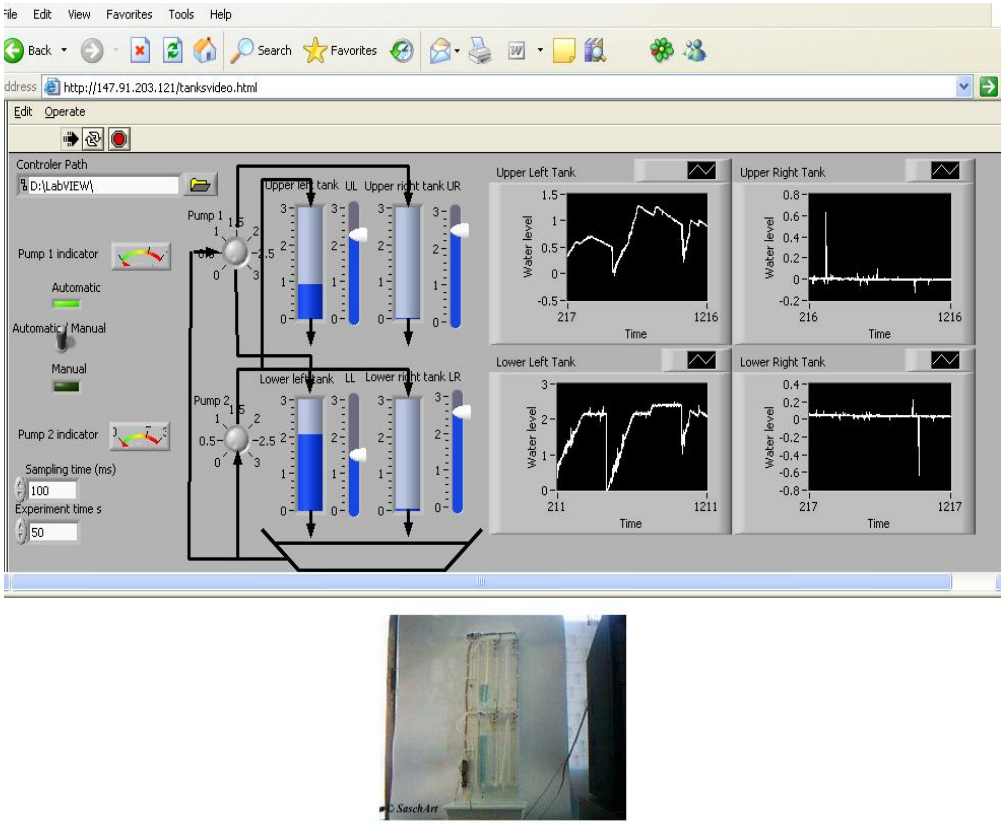
Slika 52. ***Portal Web Laboratorije Univerziteta u Kragujevcu***

Web Laboratorija Univerziteta u Kragujevcu (koja je u proceduri formalnog osnivanja) ima 6 laboratorijskih vežbi koje se mogu sprovoditi putem Interneta. Razvijeni sistem vežbi preko Interneta može da vodi u praktičnu realizaciju e learning-a kao posebne forme distant learning-a koja sa sve većom dostupnošću Interneta i sve bržim i jeftinijim pristupom postaje izuzetno atraktivna mogućnost koja se već dosta koristi. Ipak, laboratorijski eksperimenti kao obavezni deo

obrazovanja na tehničkim fakultetima i fakultetima prirodnih nauka predstavljaju svojevrsnu teškoću zbog nesrazmerno složenije problematike on line pristupa putem Interneta od pristupa klasičnim e learning sadržajima.

6.1. STRUKTURA WEB LABORATORIJE

Struktura dela WEB laboratorije, vezano za eksperiment sa spregnutim rezervoarima, za koji se koristi softversko rešenje implementirano u LabVIEW, je izuzetno jednostavna. Naime radi se o tome da je podrška za WEB pristup VI već ugrađena u LabVIEW okruženje i treba je samo aktivirati. Aktiviranje WEB pristupa je jednostavno i sastoji se od startovanja ugrađenog WEB servera, nekih jednostavnih podešavanja i podešavanja kojima se omogućava WEB pristup svakom pojedinačnom VI. Na slici 53 je predstavljen izgled prozora Internet Explorer-a u kome se vidi FP VI potpuno isto kao i na slici 48.

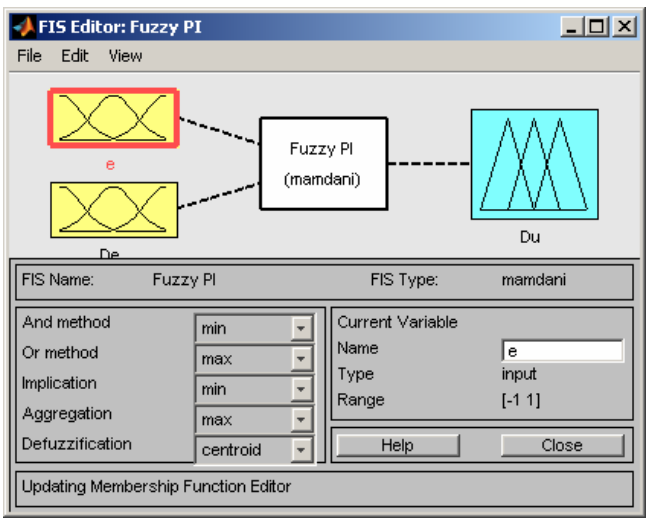


Slika 53 ***VI preko WEB-a, u prozoru Internet Explore-a***

Upravljanje eksperimentom, tj. sa VI je potpuno isto preko interneta kao i sa lokalnog računara. Da bi interaktivni pristup VI bio moguć, neophodno je na klijentski računar instalirati LabVIEW run time engine. To je besplatni dodatak koji se može download-ovati sa zvaničnog NI site-a i koristiti za daljinski pristup VI.

Dodatak A
Sinteza fazi kontrolera u programskom paketu MATLAB

Fuzzy Logic Toolbox u okviru programskog paketa MATLAB omogućava sintezu fazi kontrolera. Fuzzy Logic Toolbox se pokreće naredbom *fuzzy,* slika 1. Sastoji se od pet osnovnih prozora: FIS editora, editora funkcija pripadnosti, editora pravila, prozora za prikaz pravila i prozora za prikaz upravljačke površi.



Slika A1. ***FIS editor***

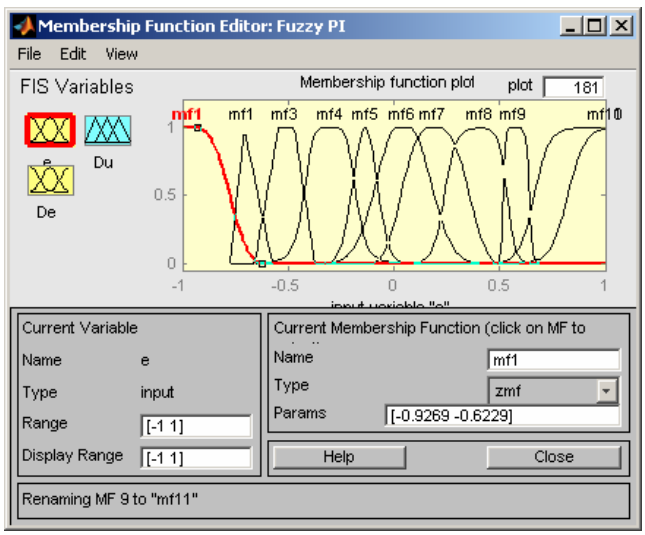
U FIS editoru se vrši izbor: operatora preseka (**and method**); operatora unije (**or method**); načina određivanja funkcija pripadnosti zaključka (**implication:** *min*-jed. (1), *prod*-jed. (2)); metoda defazifikacije (**defuzzification:** *centroid*-metod težišta, *som*-metod levog maksimuma (prva tačka u kojoj funkcija pripadnosti zaključka dostiže maksimalnu vrednost), *mom*-metod srednje vrednosti maksimuma, lom-metod desnog maksimuma maksimuma (zadnja tačka u kojoj funkcija pripadnosti zaključka dostiže maksimalnu vrednost)). Takođe, u FIS editoru moguće je promeniti nazive ulazima i izlazima tako što se u prozoru *Name* definiše novi naziv lingvističkoj promenljivoj.

Korišćenjem menija *Edit* moguće je:

- dodati promenljive - *Add Variable*... ulazne – *Input* i izlazne – *Output*
- izbrisati izabranu promenljivu- *Remove Selected Variable*
- aktivirati editor funkcija pripadnosti – *Membership Functions*…
- aktivirati editor pravila – *Rules*…

Ako se iz menija *View* izabere *Rules* otvara se prozor za grafički prikaz pravila, a ako se izabere *Surface* može se videti upravljačka površ.

Editor funkcija pripadnosti prikazan je na slici A2. On omogućava definisanje broja lingvističkih vrednosti za svaku promenljivu i izbor oblika funkcija pripadnosti izborom menija *Edit* i *Add MFs*.... Na slici A2 prikazano je 11 standardnih funkcija. Neka funkcija pripadnosti može da se izbriše ako se izabere Edit i *Remove Selected MF*. Prethodno treba da bude selektovana.



Slika A2. ***Editor funkcija pripadnosti***

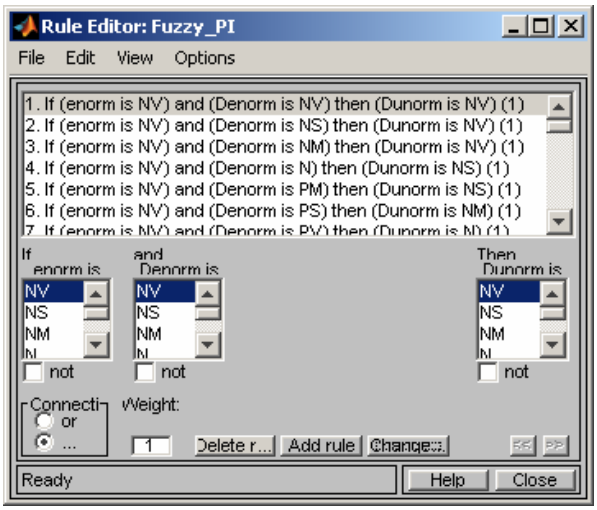
Domen ulazno/izlaznih promenljivih podešava se u prozoru *Range*, dok domen koji želimo da bude prikazan u ovom editoru podešavamo sa *Display Range.* U prozorima *Name* i *Params* definišemo ime i parametre izabranoj funkciji pripadnosti. Promenu oblika ostvarujemo izborom *Type.*

U editoru pravila moguće je automatski generisati pravila, slika A3. Izabere se veznik složenog fazi predloga, *and* ili *or.* Na primer pravilo:

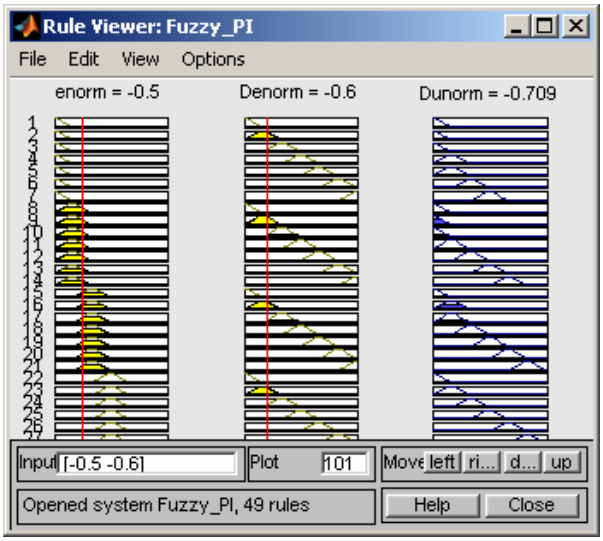
ako *enorm* je *NV* i *Denorm* je *N* tada *Dunorm* je *NS*

se generiše tako što se izaberu odgovarajuće lingvističke vrednosti: *NV* za *enorm*, *N* za *Denorm* i *NS* za *Dunorm* i aktivira se *Add rule.* Ako je potrebno da se izmeni neko pravilo onda se ono prvo izabere, zatim se izvrše potrebne izmene (dodele se nove lingvističke vrednosti) a onda se aktivira *Change*.... Pravilo se briše aktiviranjem *Delete rule.*

Pravila se mogu pisati na Engleskom, Nemačkom i Francuskom jeziku (*Options/Language)*. Ona mogu biti prikazana opisno, pomoću matematičko-logičkih simbola ili indeksno (*Options/Format: Verbose*, *Symbolic, Indexed)*.



Slika A3. ***Editor pravila***



Slika A4. ***Prozor za prikaz pravila***

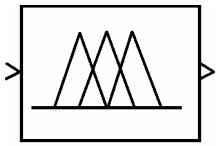
Ako u editoru pravila izaberemo *View* a onda *Rules* dobićemo grafički prikaz definisanih pravila, slika A4. Svaki red predstavlja po jedno pravilo. Prve dve kolone se odnose na ulazne promenljive, a poslednja na izlaznu. Ulazne vrednosti mogu se zadati preko tastature u polju *Input.* U gornjem desnom uglu dobija se defazifikovana vrednost izlaza.

Izgled upravljačke površi moguće je dobiti izborom *View/Surface.*

Da bi se fazi kontroler, čija je sinteza ostvarena u Fuzzy Logic Toolbox-u koristio u Simulinku potrebno je FIS sačuvati na disku ili ga eksportovati u radni prostor MATLAB-a. Ako je FIS sačuvan na disku tada se u radni prostor učitava sa:

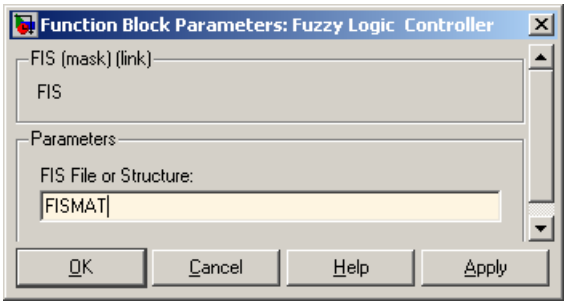
FISMAT = readfis (‘ime_sis.fis’)

Blok za fazi kontroler u Simulinku izgleda kao na slici A5.

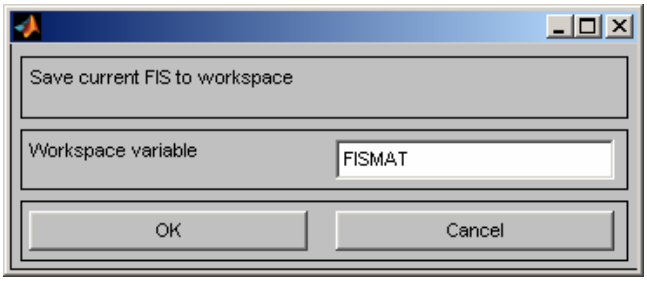


Slika A5. ***Blok za fazi kontroler u Simulinku***

Dva puta se klikne na blok i dobija se prozor kao na slici A6 u okviru koga se unosi FISMAT.



Slika A6.



Slika A7. ***Eksportovanje FIS-a u radni prostor***

Ako se FIS eksportuje u radni prostor onda se pojavljuje prozor kao na slici A7. U polju *Workspace variable* unosi se odgovarajuće ime.

Dodatak B

Listing Delphi programa za akviziciju i upravljanje na lokalnom računaru

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Dijag, nidaqmx1, NIDAQmxCAPI_TLB, ExtCtrls;

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    ScrollBar1: TScrollBar;
    ScrollBar2: TScrollBar;
    Label1: TLabel;
    Label2: TLabel;
    Image: TImage;
    SaveDialog1: TSaveDialog;
    StartButton: TButton;
    StopButton: TButton;
    SnimiButton: TButton;
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure ReadAI;
    procedure ResetAO;
    procedure WriteAO;
    procedure Timer1Timer(Sender: TObject);
    procedure StartButtonClick(Sender: TObject);
    procedure StopButtonClick(Sender: TObject);
    procedure SnimiButtonClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation
```

```
var
// Promenljive koji sluze za komunikaciju sa hardverom
  Aldata : Array[0..31] of Double;
  AOdata : Array[0..31] of Double;
// Bafer u koje se pakuju podaci iz eksperimenta
  Rs : Array[0..10000, 0..1] of Double;       // Referentne vrednosti
  Als : Array[0..10000, 0..7] of Double;          // Analogni ulazi
  AOs : Array[0..10000, 0..1] of Double;          // Analogni izlazi
// Perioda odabiranja, proteklo vreme, pocetno vreme, sledeci
// trenutak uzorkovanja i krajnje vreme
  dt, t, t0, ts, tend : double;
// Redni broj periode odabiranje
  SampleNo : LongInt;
// Flag da se omogucava akvizicija
  DoACQ : Boolean;
                                // Promenljive regulatora
  r1, r2, ki1, ki2, kp1, kp2, e1, e2, i1, i2, u1, u2 : Double;

{$R *.dfm}

procedure TForm1.FormCreate(Sender: TObject);
begin
  dt := 0.1;                                  // Perioda odabiranja u sekundama
  tend := 200;                                // Vreme trajanja eksperimenta
  DefineGraphs(Image, tend);                 // Definisi izgled grafika
  ResetAO;                                    // Postavi sve analogne izlaze na 0V
end;

procedure TForm1.FormDestroy(Sender: TObject);
begin
  ResetAO;                                    // Na kraju programa upisi 0V na sve DA konvertore
end;

// Procedura cita odjednom svih 7 kanala AD konvertora i stavlja
// vrednosti u niz Aldata
// Vrednosti se ocitavaju u Voltima. Postavljen je ulazni opseg
// napona je +/-10V.
procedure TForm1.ReadAI;
var
  hdl  : Integer;
  p : Pointer;
  ret : LongInt;
  sampsPerChanRead : LongInt;
begin
  ret := DAQmxCreateTask('Analog Input', @hdl);
  ret := DAQmxCreateAIVoltageChan(hdl, 'Dev1/ai0:Dev1/ai7', '', DAQmx_Val_InputTermCfg_RSE,  -
10.0, 10.0, DAQmx_Val_VoltageUnits1_Volts, '');
  p := nil;
  ret := DAQmxReadAnalogF64(hdl, 1, 10, DAQmx_Val_GroupByScanNumber, @Aldata, 10,
@sampsPerChanRead, p);
  ret := DAQmxWaitUntilTaskDone(hdl,10);
  ret := DAQmxStopTask(hdl);
  ret := DAQmxClearTask(hdl);
end;

// Procedura upisuje vrednosti iz niza AOdata na oba DA konvertora
// Vrednosti koje se upisuju su u Voltima. Opseg Da konvertora je
// 0..5V
procedure TForm1.WriteAO;
var
  hdl  : Integer;
  p : Pointer;
```

```
  ret : LongInt;
  sampsPerChanWriten : LongInt;
begin
  ret := DAQmxCreateTask('Analog Output', @hdl);
  ret := DAQmxCreateAOVoltageChan(hdl, 'Dev1/ao0:Dev1/ao1', '', 0.0, 5.0,
DAQmx_Val_VoltageUnits1_Volts, '');
  p := Nil;
  ret := DAQmxWriteAnalogF64(hdl, 1, -1, 10, DAQmx_Val_GroupByScanNumber, @AOdata,
@sampsPerChanWriten, p);
  ret := DAQmxWaitUntilTaskDone(hdl,10);
  ret := DAQmxStopTask(hdl);
  ret := DAQmxClearTask(hdl);
end;

// Procedura postavlja izlaze oba DA konvertora na 0V
procedure TForm1.ResetAO;
begin
  AOdata[0] := 0;
  AOdata[1] := 0;
  WriteAO;
end;

// Procedura snima prikupljene podatke iz eksperimenta na disk
procedure TForm1.SnimiButtonClick(Sender: TObject);
var
  i : LongInt;
  f : TextFile;

  function PadSpacesLeft(s : String; len : Integer) : String;
  begin
    Result := Copy(s, 1, len);
    while Length(Result) < len do Result := ' '+Result;
  end;

begin
  DoACQ := False;                          // Zaustavi eksperiment, ako je jos u toku
                                           // Odredi ime datoteke u koju ce se snimiti podaci
  if Not SaveDialog1.Execute then Exit;
  AssignFile(f, SaveDialog1.FileName); Rewrite(f);
                                           // Upisi zaglavlje
  WriteLn(f, 't       r1      r2      u1      u2      gornji l. donji l.  gornji d. donji d.');
                                           // Upisi podatke
  for i := 0 to SampleNo-1 do
    WriteLn(f, PadSpacesLeft(FloatToStrF(i*dt, ffFixed, 18, 3), 10),
            PadSpacesLeft(FloatToStrF(Rs[i,0],ffFixed,18,3), 10),
            PadSpacesLeft(FloatToStrF(Rs[i,1],ffFixed,18,3), 10),
            PadSpacesLeft(FloatToStrF(AOs[i,0],ffFixed,18,3),10),
            PadSpacesLeft(FloatToStrF(AOs[i,1],ffFixed,18,3),10),
            PadSpacesLeft(FloatToStrF(Als[i,0],ffFixed,18,3),10),
            PadSpacesLeft(FloatToStrF(Als[i,1],ffFixed,18,3),10),
            PadSpacesLeft(FloatToStrF(Als[i,2],ffFixed,18,3),10),
            PadSpacesLeft(FloatToStrF(Als[i,3],ffFixed,18,3),10));

  CloseFile(f);
end;

// Procedura za rucno zaustavljanje eksperimetna pre krajnjeg vremena
procedure TForm1.StopButtonClick(Sender: TObject);
begin
  DoACQ := False;
end;

                                         // Procedura za pokretanje eksperimenta
```

```
procedure TForm1.StartButtonClick(Sender: TObject);
begin
  DefineGraphs(Image, tend);            // Obrisi grafike
  ResetAO;                              // Postavi 0V na sve DA konvertore
  SampleNo := 0;                        // Obrisi brojac prikupljenih podataka
                                        // Resetuj integratore
  i1 := 0;
  i2 := 0;
                                        // Pokreni eksperiment
  t0 := Now;                            // Pokupi trenutno vreme (u danima)
  ts := t0;                     // Prvo uzorkovanje ce se napraviti odmah
  DoACQ := True;                        // Pokreni akviziciju
  Timer1.Enabled := True;
  Timer1Timer(Sender);
end;

procedure TForm1.Timer1Timer(Sender: TObject);
var
  i : LongInt;
begin
// Perioda tajmera je dosta kraca od periode uzorkovanja. Kod svakog
// okidanja tajmera proverava se da li je dosao trenutak za uzorkovanje.
  if Now < ts then Exit;
  Inc(SampleNo);                        // Inkrementiraj brojac prikupljenih uzoraka
                                        // Izracunaj sledeci trenutak uzorkovanja
  ts := t0+dt*SampleNo/(24*3600);
  t := (ts-t0)*(24*3600);         //Izracunaj vreme trajanja eksperimenta u s.
                                  // Prekini eksperiment ako je isteklo vreme ili se rucno zaustavlja
  if (t > tend) Or Not DoACQ then
    begin
      ResetAO;
      Timer1.Enabled := False;
      Exit;
    end;
                                        // Procitaj ulaze
  ReadAI;
                                        // Regulator
  kp1 := 2;
  ki1 := 0.5;
  kp2 := 2;
  ki2 := 0.5;
                                        // Referentne vrednosti
  r1 := 1.2+ScrollBar1.Position/200;
  r2 := 1.2+ScrollBar2.Position/200;
                                        // Signali greske
  e1 := r1 - AIdata[1];
  e2 := r2 - AIdata[3];
                                        // Integeratori
  i1 := i1+dt*ki1*e1;
  i2 := i2+dt*ki2*e2;
                                        // Antiwindup
  if kp1*e1+i1 > 5 then i1 := 5-kp1*e1;
  if kp1*e1+i1 < 0 then i1 := -kp1*e1;

  if kp2*e2+i2 > 5 then i2 := 5-kp2*e2;
  if kp2*e2+i2 < 0 then i2 := -kp2*e2;
                                        // Kompletiranje upravljanja
  u1 := kp1*e1+i1;
  u2 := kp2*e2+i2;
  // Upisi vrednosti u niz koji ce biti prosledjen na DA konvertore
  AOdata[0] := u1;
  AOdata[1] := u2;
                                        // Upis na DA konvertore
```

```
  WriteAO;
                                        // Nacrtaj podatke iz tekuce periode odabiranja
  AI0.DataDraw(Image.Canvas, t, AIdata[0]);
  AI1.DataDraw(Image.Canvas, t, AIdata[1]);
  AI2.DataDraw(Image.Canvas, t, AIdata[2]);
  AI3.DataDraw(Image.Canvas, t, AIdata[3]);
  AO0.DataDraw(Image.Canvas, t, r1{AOdata[0]});
  AO1.DataDraw(Image.Canvas, t, r2{AOdata[0]});
                                        // Sacuvaj podatke iz tekuce periode odabiranja
  Rs[SampleNo, 0]  := r1;
  Rs[SampleNo, 1]  := r2;
  for i := 0 to 7 do AIs[SampleNo, i] := AIdata[i];
  for i := 0 to 1 do AOs[SampleNo, i] := AOdata[i];
end;

end.
```

Dodatak C

Opšta uputstva za pisanje rada sa elementima matematičko tehničkog pravopisa

Cilj pisanja izveštaja i seminarskih radova je proširivanje praktičnog i teorijskog znanja studenata iz određene naučne oblasti, kao i priprema za pisanje diplomskog rada i samostalnih stručnih i naučnih radova.

Seminarski kao i diplomski rad sadrži: naslovnu stranu, sadržaj, uvod, glavni deo koji se sastoji iz više poglavlja, zaključak i spisak literature. Stručni i naučni radovi su prošireni abstraktom i ključnim rečima i nemaju naslovnu stranu i sadržaj

Naslovna strana sadrži osnovne podatke kao što su: naziv univerziteta i fakulteta, naziv predmeta u okviru koga je rađen rad, naslov rada, ime i prezime kandidata i broj indeksa, ime i prezime mentora, mesto i datum izrade.

U **sadržaju** se navode naslovi i podnaslovi poglavlja kao i brojevi strana na kojima se nalaze.

U **abstraktu** se u nekoliko rečenica koncizno navodi problem koji je razmatran u radu, kao i metode koje su korišćene za njegovo rešavanje. U par rečenica komentarišu se dobijeni rezultati.

U prvom delu **uvoda** potrebno je da se ukaže na predmet rasprave, relevantnu oblast i postojeće stanje u oblasti. Analizira se referentna literatura. Izvori literature mogu da budu: knjige, monografije, radovi iz domaćih i međunarodnih časopisa ili iz zbornika sa konferencija i savetovanja, dostupne Internet publikacije. Na kraju uvoda definiše se zadatak istraživanja sa nagoveštajem doprinosa rada. Ukratko se izlaže struktura rada po poglavljima.

U **glavnom telu rada** se detaljno i precizno definiše suština problema. Tema rada se obrađuje kroz prikaz najvažnijih teorijskih i praktičnih aspekata. Tekst treba da bude organizovan u više celina koje se označavaju naslovima i podnaslovima. Treba izložiti razvoj same ideje od koncepta do rešenja. Teorijska razmatranja se verifikuju putem rezultata simulacije, eksperimenta, statističkog pregleda, uz objašnjenje

organizacije simulacije/eksperimenta. Rezultati imaju vrednost za čitaoca jedino ukoliko su uslovi ispitivanja opisani sa dovoljno detalja. Jasno i sistematično predstaviti dobijene rezultate.

U **zaključku** se daje ocena dobijenih rezultata u kontekstu postavke problema iz uvoda. Navode se nedostaci i ograničenja predloženog rešenja, kao i predlozi kako se istraživanje može poboljšati i dalje razvijati.

Na kraju rada literatura se navodi na sledeći način:

Rad iz časopisa:

[1] K. S. Tang, K. F. Man, G. Chen, S. Kwong, "Fuzzy contoller design: methods classification and parameters choice", *IEEE Transactions on Industrial Electronics,* Vol. 48, No. 4, pp. 757-765, 2001.

Knjiga:

[2] W. J. Palm, *Control Systems Engineering*, John Wiley & Sons, New York, 1986.

Rad iz zbornika radova, saopšten na konferenciji:

[3] J. Đorđević, T. Borozan, B. Nikolić, "Softversko okruženje za simulaciju računara," *Zbornik XLV Konferencije za ETRAN,* str. 86-89, 2001.

Operatore i oznake veličina koje ne uzimaju brojne vrednosti pišu se običnim slovima, a oznake veličina koje mogu uzimati brojne vrednosti – kurzivom (italic). Brojevi se pišu uspravnim fontom.

Na primer, $\sin x$, $\ln x(t)$, d*f(t)*/d*t*.

Jedinice fizičkih veličina pišu se uspravnim fontom, pri čemu se ostavlja jedno prazno mesto između brojne vrednosti i odgovarajućih jedinica.

Na primer: h=45 cm, *U*=12 V, itd.

Jednačine se pišu sa numeracijom uz desnu ivicu, kao

$$u = K_p e + K_D \frac{\mathrm{d}e}{\mathrm{d}t} \tag{1}$$

Matrice i vektori pišu se uspravnim, bold fontom.

Na primer.

$$\mathbf{a} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \ \mathbf{b} = \begin{Bmatrix} b_{11} \\ b_{21} \end{Bmatrix}.$$

U radu se literature navidi u uglastim zagradama po redosledu citiranja.

Na primer:

U [1] je razmatrana ...

Slike i tabele treba da budu numerisane i opisane. Iznad tabele treba da stoji natpis, a ispod slike objašnjenje.

Na primer:

Tabela 1. *Rezultati eksperimentalnih merenja*. Slika 1. *Odziv sistema.*

Font kojim se piše tekst ispod slika i tabela treba da se razlikuje od fonta osnovnog teksta (obično je font teksta ispod ilustracija za jedan broj manji ili je pak italic).

Literatura

1. S.D. Bencomo, "Control learning: Present and future", *Annual Reviews in Control*, vol. 28, pp. 115-136, Elsevier Science, 2004.
2. M. Piasecka, *Online Experiments and Learning Management Systems*, *Co-op Work Report,* NRC – National Research Council of Canada, Fredericton, NB, 2004.
3. http://www.learnet.de/
4. *Making Sense of Learning Specifications & Standards: A Decision Maker's Guide to their Adoption*, the MASIE Center, 2002 USA
5. P. Horacek, „Laboratory experiments for control theory courses: A survey", *Annual Reviews in Control,* Elsevier Science, 2000.
6. http://vlab.ee.nus.edu.sg/vlab/control/index.html
7. KW Lim, UNSW & KK Sin, "Teaching with the *KRi* Coupled-Tank Control Apparatus Model PP-100"**,** *KentRidge Instruments Pte Ltd, 1995.*

   http://www.kri.com.sg/ct104.pdf, http://www.kri.com.sg/ct101.pdf
8. F.J. Doyle III et al. „Innovative control education using a 4-tank experiment and the WWW", *Proc. of IFAC/IEEE Symposium on Advances in Control Education ACE 2000*, Australia, 2000
9. K.H. Johansson, "The Quadruple-Tank Process: A Multivariable Laboratory Process with an Adjustable Zero", *IEEE Trans. on Control Systems Technology*, vol. 8, No. 3, May 2000.
10. M. Matijević, G. Jakupović, J. Car, *Računarski podržano merenje i upravljanje*, Univerzitet u Kragujevcu, 2005
11. M.R. Stojić, *Kontinualni sistemi automatskog upravljana*, Naučna knjiga, Beograd, 1988.
12. K. Astrom and B. Wittenmark, *Computer Controlled Systems, Theory and Design*, Third Edition, Prentice Hall, New Jersey, 1997.
13. http://www.quanser.com/
14. http://www.fbk.com/
15. http://www.tecquip.com

16. D. Lazić, M. Ristanović, *Uvod u Matlab*, Mašinski fakultet, Univerzitet u Beogradu, 2005

17. L. Ćalasan, M. Petkovska, *MATLAB i dodatni moduli Control System Toolbox i Simulink*, Mikro knjiga, Beograd, 1996.

18. K. Ogata, *Modern Control Engineering*, Prentice Hall, 1997.

19. Lj. Draganović, *Projektovanje SAU*, LOLA Institut, Beograd, 2000,

20. E. Laubwald, *Coupled Tanks Systems I,* http://www.control-systems-principles.co.uk

21. G. Franklin, J.D. Powell and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Chapter 2, Prentice Hall, 1999.

22. M. Stefanović, M. Matijević, M. Ravlić, V. Cvjetković, *Laboratorijska merenja i upravljanje eksperimentima putem Interneta*, Univerzitet u Kragujevcu, 2007

23. Robert E. King, *Computational Intelligence in Control Engineering*, Marcel Dekker Inc, 1999.

24. Kevin M. Passino, Stephen Yurkovich, *Fuzzy control*, Addison Wesley Longman Inc, 1998

25. Toshinori Munakata, *Fundamentals of the New Artificial Intelligence*, Springer, 1998.

26. http://www.dcc.ttu.ee/Automaatika/LAS/ISS0021/Fuzzy_control_2006.pdf

27. http://www.ro.feri.uni-mb.si/predmeti/int_reg/Welcome.html

28. http://www.etse.urv.es/~aoller/fuzzy/fuzzy_logic.htm

29. *Introduction to LabView - Interactive Training CD*, National Instruments Corporation, 2004.

30. *LabView 8*, Evaluation CD, National Instruments Corporation, 2005.

31. D. Stanković, "Elementi tehničko-matematičkog pravopisa", uputstvo za studente, Elektrotehnički fakultet Univerziteta u Beogradu, 2005.

32. http://vlab.ee.nus.edu.sg/vlab/control/index.html

33. http://www.control-systems-principles.co.uk

34. http://www.control.lth.se/~ictools/, http://www.control.lth.se/education/processes/welcome.html

35. http://control.ee.ethz.ch/~hybrid/tts/

36. http://sine.ni.com/nips/cds/view/p/lang/en/nid/14605