



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA



Ispit iz predmeta „Algoritamske heuristike”

Profesor : Dautović Staniša

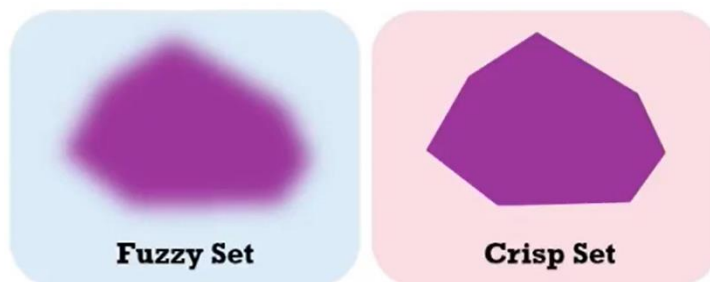
Student : Milin Ivan E1-79/2023

Zadatak 2 :

Intervalna Fazi logika (Type 2 Fuzzy Logic) i Fazi logički PID kontroleri (FLC) tipa 1 i tipa 2

Zadatak 2a)

• Uvod :



1. Poređenje između fuzzy i crisp set [2]

Pored teorije verovatnoće, postoje i drugi načini kojima se može posmatrati neizvesnost ili nepreciznost u budućim procenama, jedan od takvih načina je fazi logika i fazi skupovi.

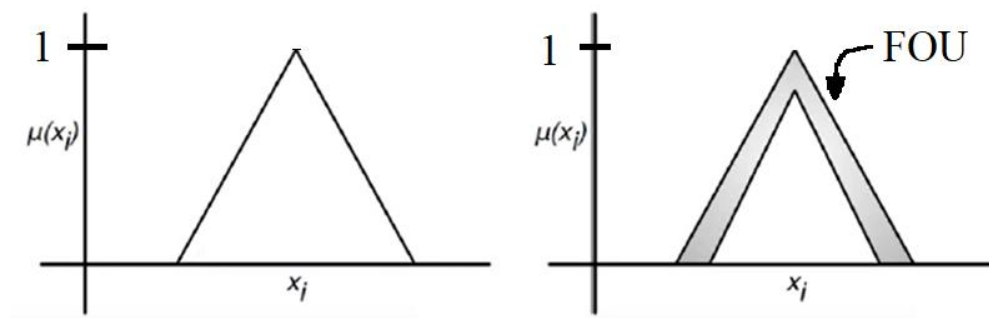
U klasičnoj teoriji skupova neki element x iz skupa X može, a i ne mora pripadati nekom skupu A , dok u klasičnoj logici neki iskaz može biti samo istinit ili samo lažan, ne postoji treća varijanta. U svakodnevnom govoru često se srećemo sa rečima čije značenje je dvosmisleno ili granice te reči nisu jasno definisane. U mekom računarstvu (eng. *soft computing*) takav termin se naziva lingvistička promenljiva.

Na slici 1. možemo videti da za razliku od *Fuzzy* seta, *Crisp* set ima jasno definisane oštre granice između roze i ljubičaste oblasti, u pogledu digitalne elektronike to može biti dve različite diskretne oblasti (tačno ili netačno, 0 ili 1, nizak ili visok napon)

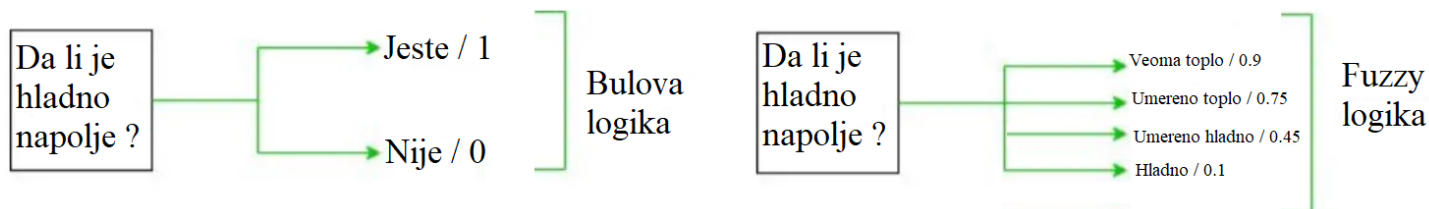
Dok kod *Fuzzy* seta te granice nisu jasno definisane između dve oblasti, postoji tolerancija na nepreciznost. U ovom slučaju postoji prelazna oblast koja nije diskretna (odgovor ne može biti samo tačan ili netačan, već postoji skup beskonačnih vrednosti, ili u pogledu napona sve vrednosti između visokog i niskog, ili svi realni brojevi između 0 i 1).

Teorija Fuzzy skupova uvodi nov pojam koji se zove funkcija pripadnosti (eng. *membership function*) koja pokazuje koliko x iz skupa X ispunjava uslov pripadnosti skupu A . U fuzzy logici ta funkcija može biti npr. trougaona, trapezoidna, Gausova, itd... Vrednost te funkcije u klasičnoj teoriji može imati samo vrednost 0 ili 1, dok kod fuzzy skupova ta vrednost može biti bilo koji realni broj između 0 i 1. [10]

• Fuzzy set tip 1 i tip 2 :



2. Fuzzy set tip 1 i fuzzy set tip 2 [2]



3. Razlika između Bulove i Fuzzy logike

Fuzzy logika (nejasno, maglovito) je računarska tehnika koja imitira način na koji ljudi razmišljaju, oni mogu donositi odluke na osnovu nejasnih informacija i jezički nepreciznih/dvosmislenih informacija. Neke informacije čak treba uzeti i sa rezervom ako su dobijene od nepouzdatih izvora, to dodatno usložnjava čitav proces. Jedan takav primer je određivanje da li je temperatura napolju visoka ili niska bez poznavanja konkretne vrednosti **temperature** (hladno, umereno hladno, umereno toplo, toplo) ili ukoliko se opisuje **brzina** (jedva da ide, mala, velika, vrlo velika). Ukoliko na osnovu takvog opisa treba da se obučemo, nekada ti opisi nisu dovoljni.

Na ovaj način bi mogao da isprojektuje regulator brzine koji bi imao 4 koraka, gde bi brzina obrtanja bila opisana rečima, a sama kontrola zavisila od izgovorene reči. Ovakav pristup se može nazvati „račun pomoću reči“, (eng. *CW – Computing with Words*) kod koga je u cilju da se vrednost fizičke veličine izrazi običnim rečima i poznavanje nekog sistema pretoči u upravljački algoritam. Takođe sa Fuzzy logikom moguće je predviđati kvarove nekog sistema, senzori vode računa npr. o radnoj temperaturi motora i ukoliko dođe do povećanja temperature sistem predlaže akcije (smanjenje brzine) koje treba preduzeti kako ne bi došlo do pregorevanja. tj. kvara. U bankarstvu u slučaju donošenja odluke da li je klijent pogodan ili ne prilikom davanja kredita, takve odluke banka može donositi na osnovu iskustva u poslovanju sa klijentima sličnih osobina sa kojima je sarađivala u prošlosti i time može da eskivira kašnjenje isplate rate kredita. [6]

Profesor Lofti Zadeh je prvi uveo pojam fuzzy set tip-1(1965) zasnovano na teoriji mogućnost, on je time hteo da kaže da aproksimativno rezonovanje nudi prirodniji oblik rezonovanja nalik načinu na koji čovek rezonuje stvari, tada čovek odlučuje na osnovu svog iskustva ili intuicije. Primenom prvog tipa shvatio je da taj tip ima određena ograničenja u samoj praksi da bi deset godina kasnije uveo i pojam fuzzy set tip-2.

Ograničenja su bila u načinu na koji je set tip-1 upravljao sa nesigurnošću podataka, tj. da modeluje i minimizira efekat neizvostnosti što je poboljšano u fuzzy set tip-2.

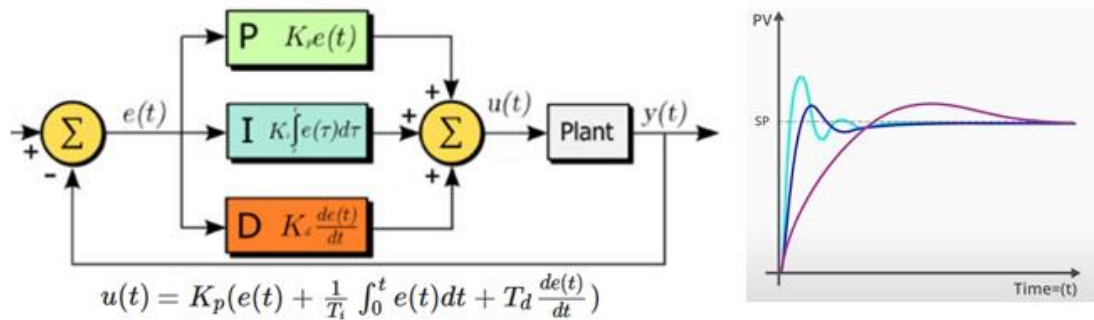
Npr. fuzzy set tip-1 uspešno je realizovan u nekoliko optimizacionih problema poput prepoznavanja šablona sa slike ili klasifikacije podataka, dok sa fuzzy set tip-2 je moguće dobiti bolje rezultate.

Ovo je moguće izvesti uz veće nivoe neizvesnosti i veće prisustvo šuma stoga set tip-2 se može takođe koristiti u prepoznavanju šablona koji su još nejasniji kao i u kontroli/upravljanju sistema i donošenju odluka. [3]

Kod Fuzzy set tip-1 *membership* $\mu_i(X_i)$ funkcija ima jedinstvenu vrednost X_i (*membership value*) i ona modeluje jedinstven nivo članstva (*degree of membership*), dok je kod Fuzzy set tip-2 *membership* $\mu_i(X_i)$ funkcija ima više vrednosti X_i u nekom opsegu. Kod Fuzzy set tip-2 struktura funkcije pripadnosti je trodimenzionalna umesto dvodimenzionalna kao što je kod Fuzzy set tip-1. [12]

Na slici 2, desni grafik prikazuje *Interval type-2 membership function* koje je definisana gornjom i donjom *membership* funkcijom, gornja *membership* funkcija je ekvivalent tradicionalne *type-1 membership* funkcije dok je donja *membership* funkcija jednaka ili manja od gornje *membership* funkcije za sve moguće ulazne podatke X_i , taj opseg se naziva oblast nesigurnosti (eng. FOU – *footprint of uncertainty*) [4].

• **PID kontroler :**



4. Blok šema, formula po kojoj se ponaša PID kontroler i primer odziva[5]

Svaki sistem automatskog upravljanja ponaša se po određenom zakonu, najčešće je taj zakon predstavljen pomoću matematičke formule po kojoj sistem treba da se ponaša i u zavisnosti od nje regulator pomoću povratne sprege reguliše stanje sistema generišući upravljačke signale po potrebi, ukoliko dođe do odstupanja stvarne vrednosti sistema i očekivane vrednosti.

Na slici 4. može se videti kako u zavisnosti od samih vrednosti K_p , K_i i K_d odziv sistema može da se menja, može se videti kako se odziv posle izvesnog vremena smiruje, tj. ne osciluje što je i zapravo suština PID regulatora.

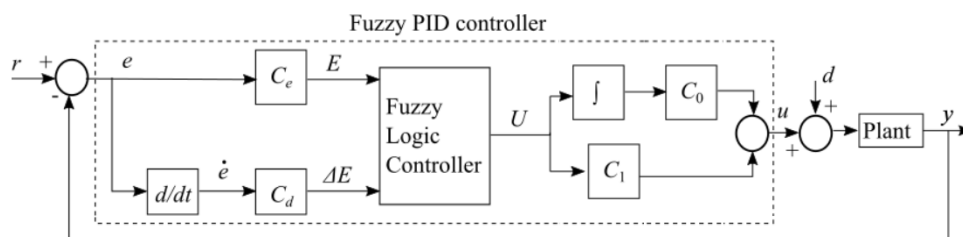
Najčešća podela regulatora po načinu generisanja kontrolnog signala je *proporcionalni, integralni i diferencijalni*.

1. Proporcionalni regulator [$u(t) = K_p * e(t)$] – pojačava signal greške $e(t)$ konstantnom K_p
2. Integralni regulator [$u(t) = K_i * \int_0^t e(t) dt$, $K_i = \frac{1}{T_i}$] – uvođenjem integralnog regulatora menja se inernost sistema, tj. sistem sporije reaguje na nagle promene unutar sistema.
3. Diferencijalni regulator [$u(t) = K_d * \frac{de(t)}{dt}$] – postavljanjem samo diferencijalnog regulatora u sistem nema poente. U ustaljenom stanju vrednost signala $u(t)$ bila bi nula, jer bi bila proporcionalna brzini promene $e(t)$. Kombinacija ovog regulatora sa P ili sa I regulatorom imalo bi smisla. Sa D regulatorom se prati dinamika sistema, prati se veličina promene greške.

Uvođenjem D regulatora povećava se stabilnost i brzina reakcija na promene u sistemu.

Kombinovanjem sva tri regulatora dobija se PID regulator koji kombinuje dobre osobine svakog individualnog regulatora i time omogućuje dobijanje veće stabilnosti sistema i veće brzine reagovanja, kao i smanjenje trajanja prelaznog procesa. [5]

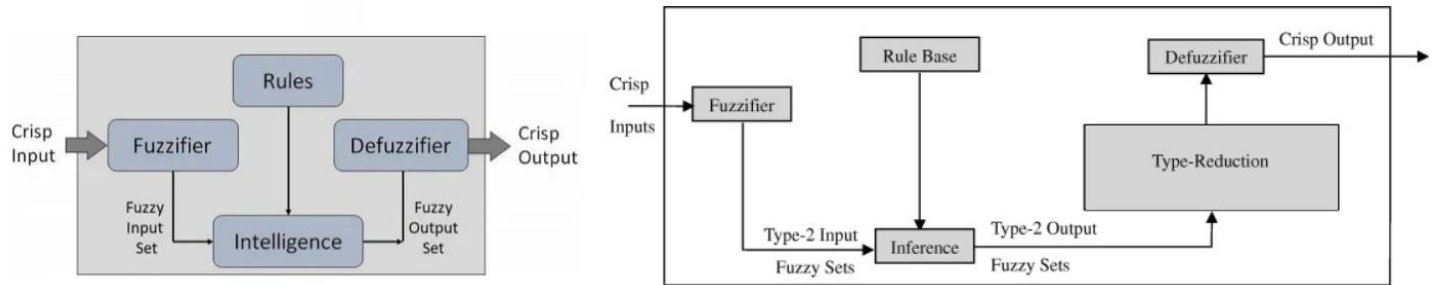
• **Fazi logički PID kontroleri (FLC):**



5. Blok šema Fuzzy PID regulatora [9]

Klasnični PID regulatori su najpopularniji u industriji zbog svoje jednostavnosti rada i niske cene. Fazi logika se koristi kako bi se poboljšale njihove osobine i sposobnost prevođenja kontrolne akcije u bazu pravila. [7]

Fazifikacioni modul dodeljuje numeričke vrednosti iz intervala $[0,1]$ vrednostima koje pripadaju skupu članstva, te vrednosti se zovu i nivoi članstva (*degree of membership*). Na slici 3. pored svakog opisa temperature dodeljana je i numerička vrednost kojom se kvantifikuje lingvistički opis date temperature, to se postiže pomoću funkcije članstva (*membership function*) i time se omogućava grafički prikaz fuzzy seta.



6. Arhitektura Fuzzy logic kontrolera tipa 1 i tipa 2 [7] [8]

Na slici 6. prikazana je arhitektura fuzzy kontrolera tipa 1 i tipa 2.

Arhitektura tipa 1 sastoji se od :

1. *Fuzzifier* – ovaj modul transformiše ulazne vrednosti u fuzzy set.
2. *Rules* – modul koji predstavlja lookup tabelu IF-THEN pravila.
3. *Intelligence* – modul simulira način ljudskog rezonovanja informacija tako što donosi zaključke na osnovu ulaznih podataka i podataka iz lookup tabele.
4. *Defuzzifier* – ovaj modul transformiše fuzzy set obrađen u *Intelligence* modulu u *crisp* vrednost (tačno/netačno, jeste/nije, 0/1).

Kao što se može videti na slici 6. *Fuzzy logic* kontroler tipa 2 ima još jedan dodatni modul „*Type-reduction*“. Njegova uloga je da redukuje fuzzy set tipa 2 na fuzzy set tipa 1, ovo se najčešće radi pomoću iterativnog Karnik-Mender algoritma koji je računarski vrlo zahtevan zbog čega performanse čitavog sistema mogu značajno da opadnu, pogotovo kod industrijskih embedded sistema koji imaju ograničene računске i memorijske mogućnosti. Alternativne TR metode ne mogu obuhvatiti sve karakteristike koje KM algoritam može. [11].

Jedan od takvih je Takagi-Sugeno-Kang(TSK) za koji postoji i softver otvorenog koda (eng. *open-source*) kako bi i šira populacija ljudi mogla sa njim da se upozna putem softvera (*type-2-fuzzy-logic-systems-matlab-toolbox* [1]). Tim softverom moguće je izvršiti razne analize kojima se može proučavati uticaj *Type-reduction* metoda IT2-FLS. Softver nam takođe pruža već ugrađene defazifikacione metode, a imamo mogućnost i da sami kreiramo svoju metodu.

Naučnici Mendel i Wu predložili su metodu aproksimacije kojom bi se izbegao Karnik-Mender algoritam zbog čega ne bi bile ugrožene performanse čitavog sistema.

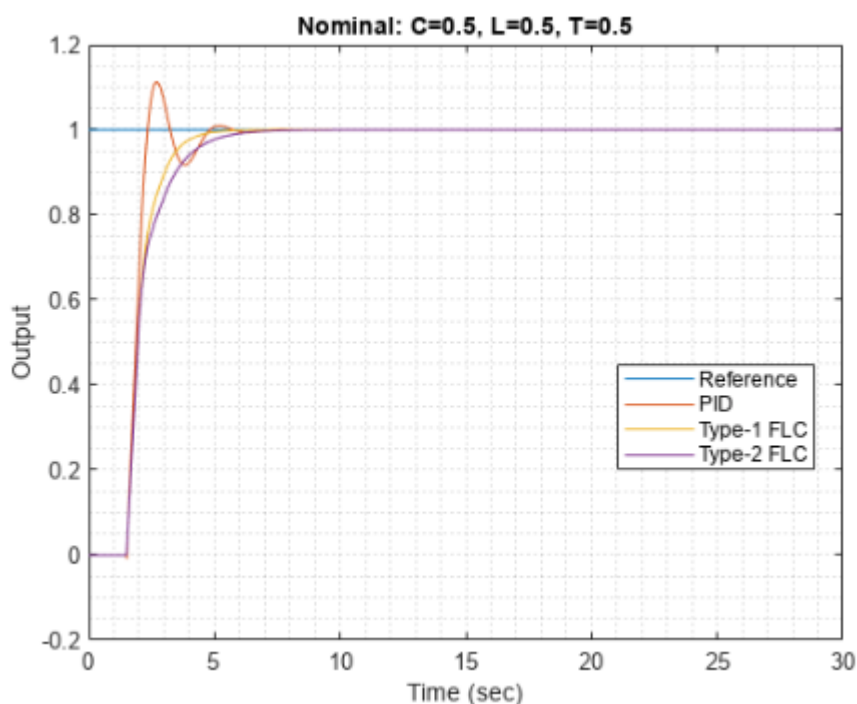
Metoda aproksimacije računa spoljne i unutrašnje granice seta za *type-reduced* seta, ova dva seta se mogu koristiti ne samo za procenu nesigurnosti sadržane u IT2-FLC već se mogu koristiti i za direktno izvođenje defazifikovanog izlaza pod određenim uslovima.

Dakle, spoljašnji i unutrašnji set imaju potencijal da eliminišu usko grlo u računanju kod IT2-FLC kontrolera.

U sustini, IT2-FLC je ili samo proporcionalno-diferencijalan (PD) ili samo proporcionalno-integralan (PI) tip kontrolera, u zavisnosti da li je izlaz Fuzzy kontrolnih pravila. Ako je izlaz kontrolni signal najčešće je onda IT2-FLC realizovan kao PD kontroler, a ako je izlaz promena kontrolnog signala tada je IT2-FLC realizovan kao PI kontroler.

IT2 Fuzzy PID kontroler se najčešće pored informacija o grešci (eng. *error*) i izvoda greške (eng. *error derivative*) sastoji i od 3-D baze pravila.

Takav IT2 Fuzzy PID kontroler je složeniji za razviti jer bi takva baza pravila bila sve kompleksija kako bi se povećavao broj kvantizacije svake dimenzije pravila. U ovom slučaju broj kontrolnih pravila povećava se kubično sa povećanjem broja kvantovanja. [8]



7. Poređenje odziva kod različitih tipova [9]

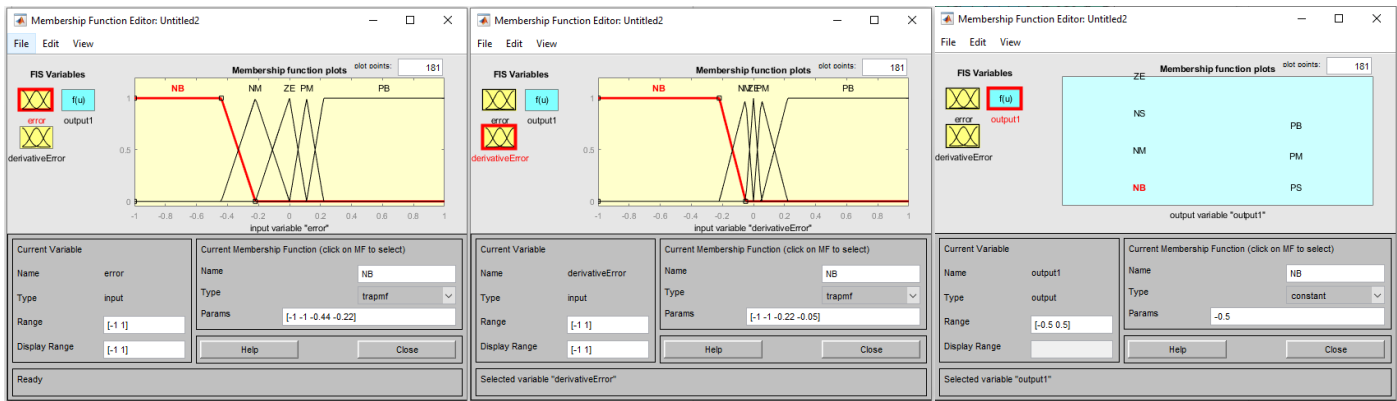
Na slici 7. prikazana je razlika u odzivu između različitih implementacija PID kontrolera.

Sa grafika se može videti :

1. Konvencionalni PID kontroler radi bolje u pogledu vremena porasta i integrala apsolutne greške (što predstavlja integral iznad jediničnog odziva)
2. U pogledu vremena porasta i vremena potrebnog da se oscilacije smire FLC tip-1 ima bolje performanse od FLC tip-2.

Robusnost konvencionalnog PID kontrolera može se popraviti korišćenjem različitih metoda kao što su predviđanje ili konfiguracija sistema sa više PID kontrolera. Sa druge strane performanse FLS tip-2 mogu se korigovati korišćenjem drugačijih vrednosti za bazu pravila, broj pravila i FOU (*footprint of uncertainty*) čime se može korigovati vreme odziva na jedinični odziv. [9]

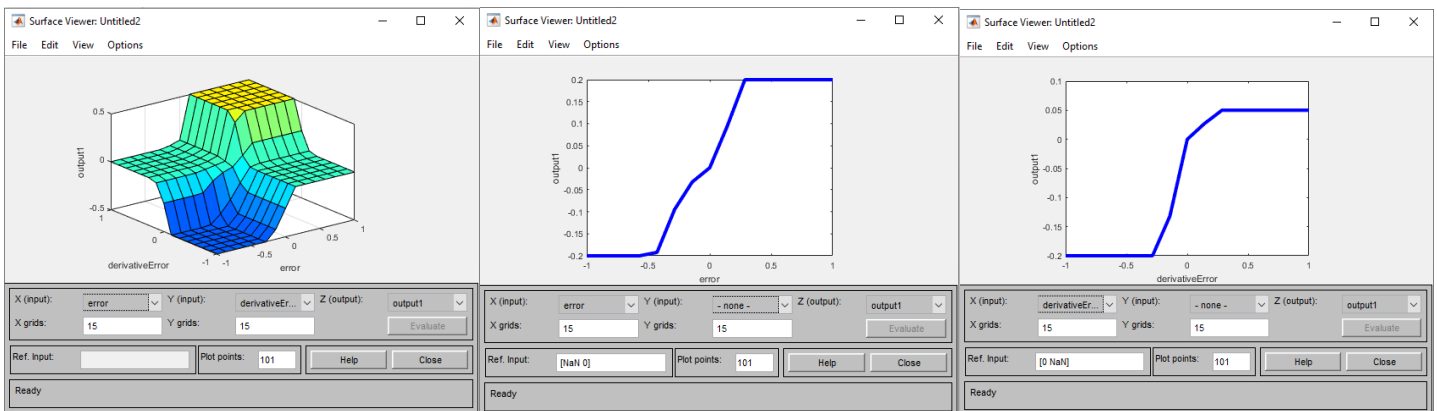
Zadatak 2b)



8. Unos vrednosti ulaznih i izlazne varijable u GUI

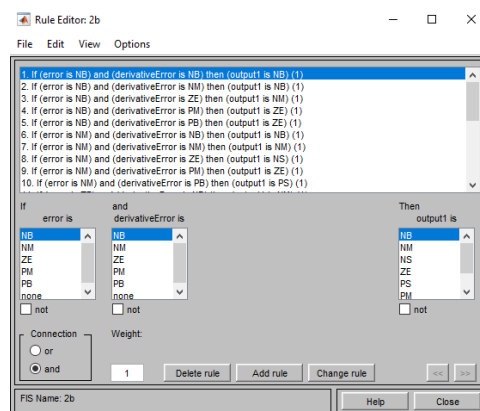
Na slici 8. prikazan je GUI (eng. *Graphical user interface*) prozor u kojem su kreirane dve ulazne varijable greška (*error*) i izvod greške (*derivativeError*) kao i izlazna varijabla (*output1*) i kojima su dodeljene vrednosti kao što je zadato u zadatku.

Ovaj prozor je moguće otvoriti pozivom naredbe *fuzzy* iz matlab konzole, zatim pri vrhu prozora je potrebno stisnuti opciju *File->Import->From File* i odabrati fajl *zadatak2b.fis*.



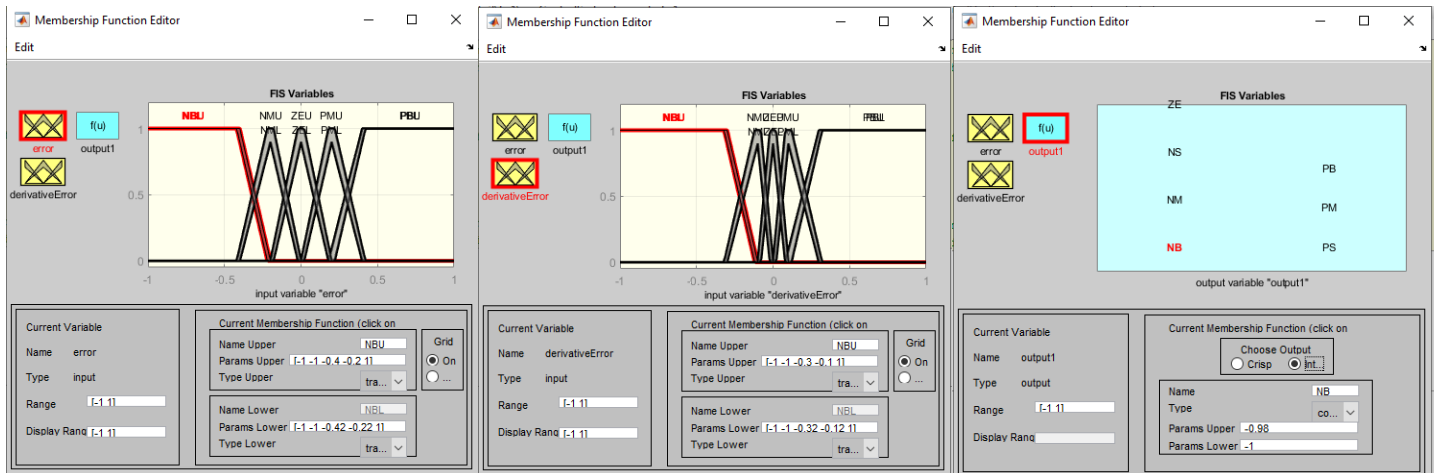
9. Površina projektovanog IT1-FLS

Na slici 9. u sklopu *Surface Viewer GUI* prikazana je iz više perspektiva površina (eng. *surface*) sistema.



Kao što je prikazano na slici iznad, na vrhu prozora pritiskom na opciju *Edit->Rules* postavljena su pravila koja su takođe definisana u tekstu zadatka.

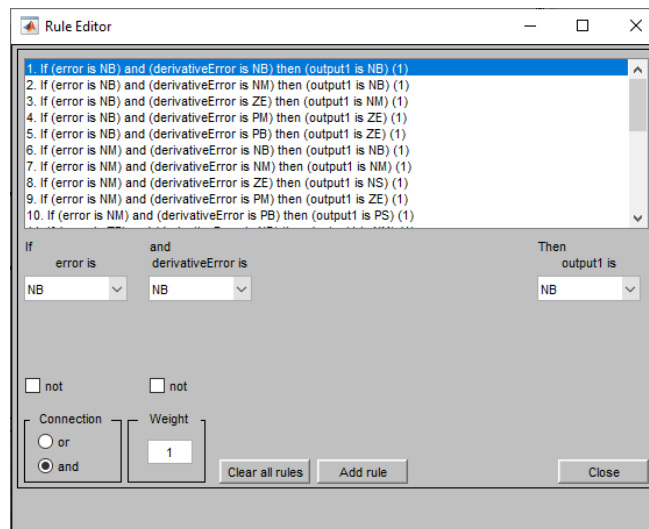
Zadatak 2c)



10. Unos vrednosti ulaznih i izlazne varijable

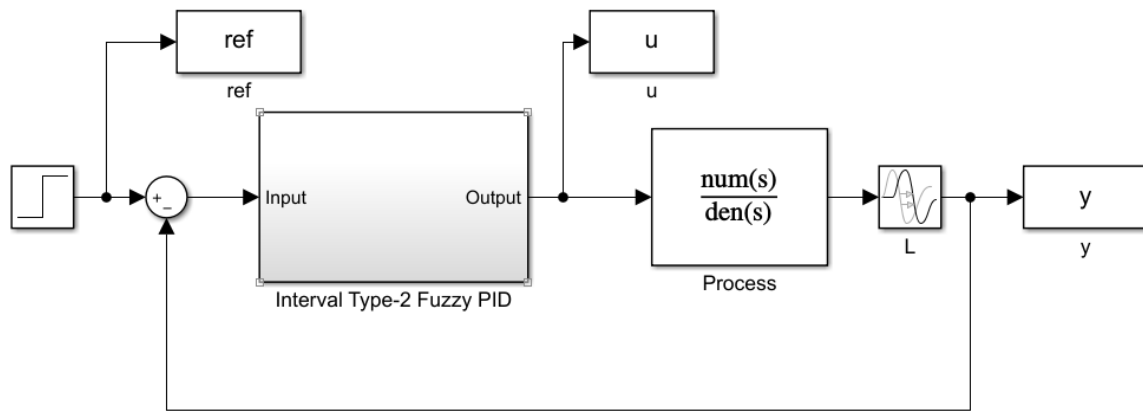
Na slici 10. prikazan je GUI (eng. *Graphical user interface*) prozor u kojem su kreirane dve ulazne varijable greška (*error*) i izvod greške (*derivativeError*) kao i izlazna varijabla (*output1*) i kojima su dodeljene vrednosti kao što je zadato u zadatku.

Ovaj prozor je moguće otvoriti pozivom naredbe **fuzzyt2** iz matlab konzole, zatim pri vrhu prozora je potrebno stisnuti opciju **File->Load->From File** i odabrati fajl **zadatak2c.t2fis**.



Kao što je prikazano na slici iznad, na vrhu prozora pritiskom na opciju **Edit->Rules** postavljena su pravila koja su takođe definisana u tekstu zadatka.

Zadatak 2d)

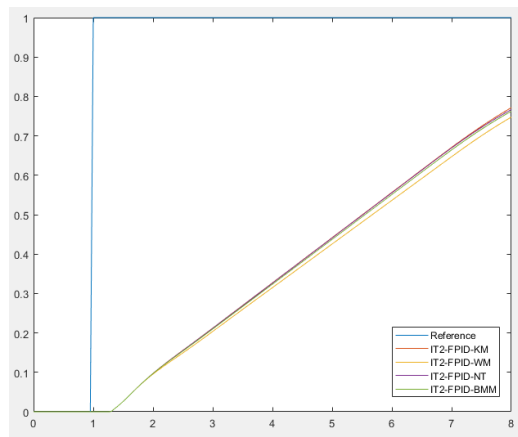


11.FLC PID kontroler toolbox-a

Na slici 11. prikazana je implementacija *Type-2 Fuzzy* PID kontrolera u Simulink alatu koji je obezbeđen uz sam softver.

Kako bi kod mogao da se pokrene potrebno je sa leve strane matlab-a u delu *Current Folder* pritisnuti desni klik miša na dva foldera iznad u odnosu na folder u kom se nalazi **runControlSystemExample.m** i odabrati opciju *Add to Path->Selected folders and Subfolder*, zatim u matlab kodu **runControlSystemExample.m** potrebno je upisati u funkciji **readt2fis** ime fajla **zadatak2c.t2fis** sa ekstenzijom **.t2fis**, kako bi kod mogao da pokrene izgenerisani fajl iz zadatka **2c)**

Nakon obavljenih navedenih koraka potrebno je pritisnuti dugme *Run* nakon čega se otvara prozor u kom se iscertava grafik kao sa slike 12, na slici je može videti jedinični ulazni signal koji predstavlja pobudu sistema i izlazni odzivni signal iz FLC PID kontrolera.



12 Jedinični ulaz i izlazni signal FLC PID kontrolera



Na slici 14. data ja izmenjena šema FLC PID kontrolera, duplim klikom na blok koji se zove *Interval Type-1 Fuzzy PID* pojaviće se blok šema kao sa slike 14, potrebno je bilo ukloniti blok *Interval Type-2 Fuzzy PID* i umesto njega postaviti modul koji se zove *Fuzzy Logic Controller* u koji je potrebno upisati ime fajla **zadatak2b.fis**.

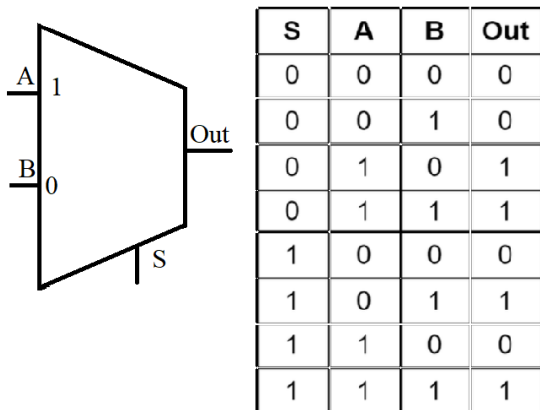
Nakon svega toga potrebno je pokrenuti fajl **runControlSystemExample_fis1.m** i otvara se prozor u kom se iscrtava grafik kao sa slike 15.



9

Zadatak 3 : Soft computing - Veštačke neuralne

Zadatak 3a)



16. Simbol multipleksera i njegova tablica istinitosti

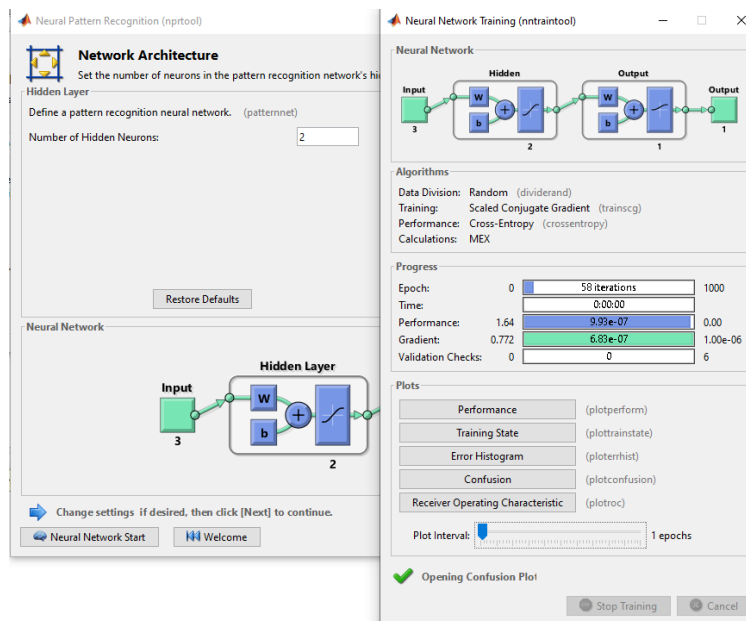
```

zadatak3a.m
1  %Tablica istinitosti
2  %Port: S A B
3  s = [ 0 0 0;
4        0 0 1;
5        0 1 0;
6        0 1 1;
7        1 0 0;
8        1 0 1;
9        1 1 0;
10       1 1 1];
11 %Port: Out
12 t = [0; 0; 1; 1; 0; 1; 0; 1;];
13
14 %Uvecavamo trening skup
15 ss_MUX = [s; s; s; s; s];
16 tt_MUX = [t; t; t; t; t];
17 ss_MUX = logical(ss_MUX);
18 tt_MUX = logical(tt_MUX);

```

17. Matlab kod multipleksera 2 na 1

Na slici 16. prikazan je simbol i tablica istinitosti multipleksera 2 na 1. Ta tablica pretočena je u matlab kod koji se nalazi u fajlu **zadatak3a.m**, koji je prikazan na slici 17, zatim je potrebno u matlabu pozvati naredbu *nstart* i odabrati opciju *Pattern Recognition app*.

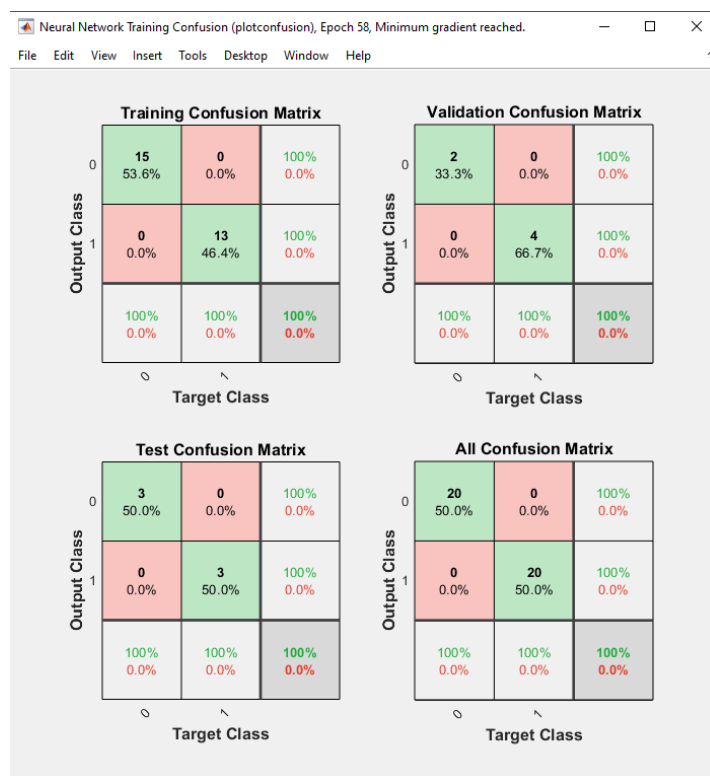


18. Treniranje mreže i rezultati nakon testiranja

Na slici 18. prikazana su dva prozora, u prozoru *Neural Pattern Recognition* podešeno je da mreža ima dva skrivena sloja mreže, eksperimentalno je provereno da ukoliko mreža ima manje od dva sloja rezultati nakon testiranja nisu 100% tačni, te bi mreža morala još nekoliko puta da se trenira.

U prozoru *Neural Network Training* može se videti da je mreža istrenirana u 58 iteracija.

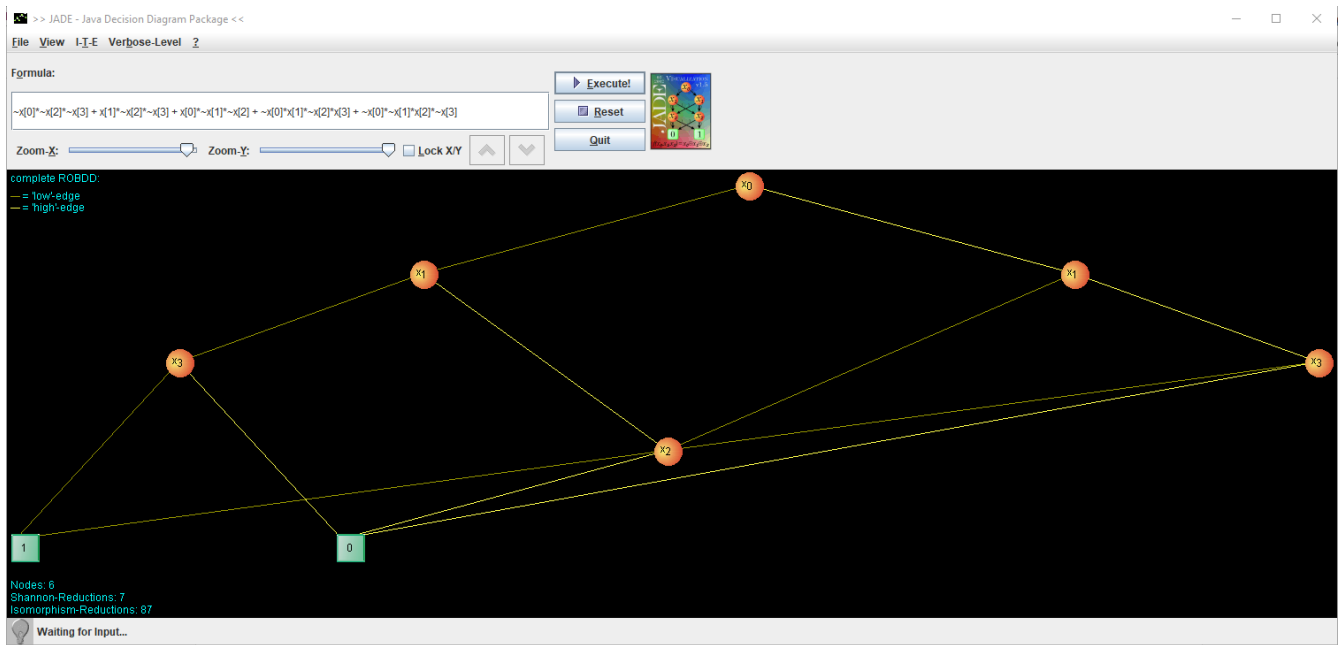
Na slici 19. prikazani su u vidu matrice rezultati treniranja i testiranja mreže, za treniranje je upotrebljeno 28 uzoraka, od toga 15 je iskorišćeno za slučaj 0,0 a 13 za slučaj 1,1. Dok je za validaciju iskorišćeno 6 uzoraka, od toga 2 za slučaj 0,0 a 4 za slučaj 1,1 i za svaku kombinaciju dobijeno je 100%.



19. Rezultati treniranja i testiranja mreže

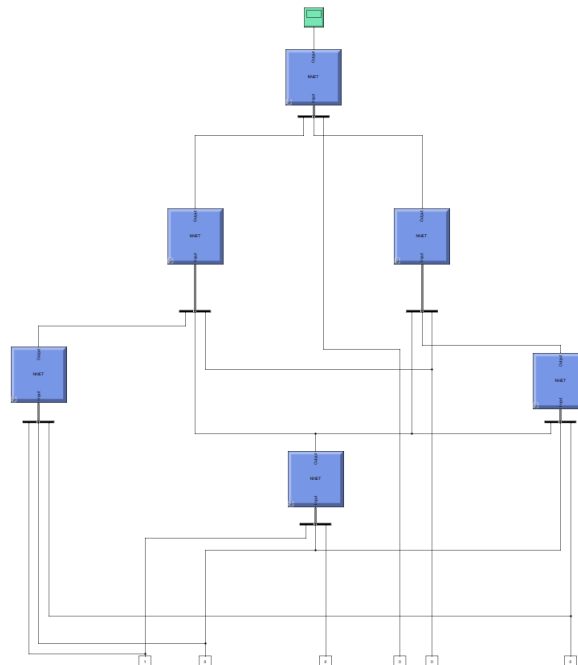
Nakon što je mreža istrenirana izgenerisan je Simulink fajl **mux2na1.slx** u kom se nalazi istrenirana mreža koja je takođe proverena da ispunjava očekivanu funkcionalnost.

Zadatak 3b)



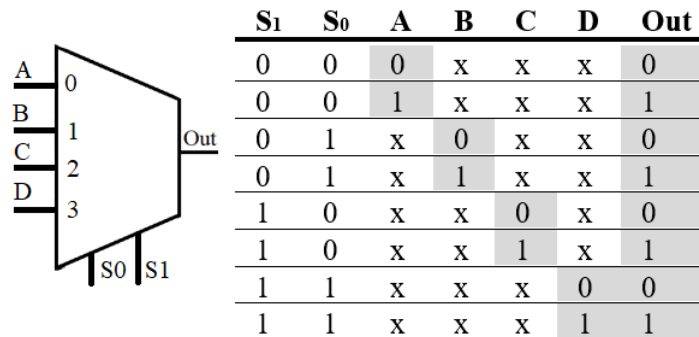
20. Graf minimizovane BF pomoću JADE alata

Na slici 20. prikazan je graf Bulove funkcije koja je minimizovana pomoću JADE alata, nakon toga je pokrenut Simulink i na mesto svakog čvora postavljeno je kolo iz fajla **mux2na1.slx** i sve to je sačuvano u fajl **zadatak3b.slx**, što se može videti na slici 21.



21. Graf sa slike 20. predstavljen u Simulink-u

Zadatak 3c)



Na slici iznad data je blok šema multipleksera 4 na 1 i prateća istinitosna tablica, tablica je pretočena u matlab kod koji se nalazi u fajlu **zadatak3c.m**, zatim je potrebno u matlabu pozvati naredbu *nnstart* i odabrati opciju *Pattern Recognition app*.



22. Treniranje mreže i rezultati nakon testiranja

23. Rezultati treniranja i testiranja mreže

Na slici 22. prikazan je prozor *Neural Network Training* u kom se može videti da je mreža sačinjena od 4 skrivena sloja i da je istrenirana u 46 iteracija.

Eksperimentalno je utvrđeno da ukoliko mreža ima manje od četiri skrivena sloja rezultati nakon testiranja nisu 100% tačni, te bi mreža morala još nekoliko puta da se trenira.

Na slici 23. prikazani su u vidu matrice rezultati treniranja i testiranja mreže, za treniranje je upotrebljeno 134 uzoraka, od toga 64 je iskorišćeno za slučaj 0,0, a 70 za slučaj 1,1. Dok je za validaciju iskorišćeno 29 uzoraka, od toga 20 za slučaj 0,0, a 9 za slučaj 1,1 i za svaku kombinaciju dobijeno je 100%.

Nakon što je mreža istrenirana, izgenerisan je simulink fajl **mux4na1.slx** u kom se nalazi istrenirana mreža koja je takođe proverena da ispunjava očekivanu funkcionalnost.

Literatura :

- [1] „An Open Source Matlab/Simulink Toolbox for Interval Type-2 Fuzzy Logic Systems“, A. Taskin, T. Kumbasar
- [2] „Type 2 Fuzzy Set (Part 1)“, <https://www.youtube.com/watch?v=BuVqDOA1fzY>, pristupio 13.5.2024.
- [3] „Comparative Study of Type-1 and Interval Type-2 Fuzzy Logic Systems in Parameter Adaptation for the Fuzzy Discrete Mycorrhiza Optimization Algorithm“, <https://www.mdpi.com/2227-7390/11/11/2501>, pristupio 13.5.2024.
- [4] „Type-2 Fuzzy Inference Systems“, <https://www.mathworks.com/help/fuzzy/type-2-fuzzy-inference-systems.html>, pristupio 13.5.2024.
- [5] „Osnove PID regulatora“, <https://www.automatika.rs/baza-znanja/teorija-upravljanja/osnove-pid-regulatora.html>, pristupio 13.5.2024.
- [6] https://www.ucg.ac.me/skladiste/blog_8554/objava_30673/fajlovi/algupr.pdf, pristupio 14.5.2024.
- [7] „FUZZY PID CONTROLLERS“, <https://medium.com/k-r-i-s-s/fuzzy-pid-controllers-26e1e1d800c5>, pristupio 14.5.2024.
- [8] <https://www.sciencedirect.com/science/article/pii/S2090447913001287>, pristupio 14.5.2024.
- [9] „Fuzzy PID Control with Type-2 FIS“, <https://www.mathworks.com/help/fuzzy/fuzzy-pid-control-with-type-2-fis.html#FuzzyPIDControlWithType2FISExample-2>, pristupio 14.5.2024.
- [10] <https://scindeks-clanci.ceon.rs/data/pdf/0042-8469/2010/0042-84691003125P.pdf>, , pristupio 15.5.2024.
- [11] „An open source Matlab/Simulink Toolbox for interval type-2 Fuzzy logic systems“, A. Taskin, T. Kumbasar
- [12] „Fazi logički sistemi FLS-T” za podršku sistema daljinskog nadzora, detekciji i selekciji kvarova“, S. D. Milić, Lj. Čičkarić, <https://zenodo.org/record/6475464/files/R%20A1%2001.pdf>, pristupio 18.5.2024.