UNIVERZITET U BANJOJ LUCI
ELEKTROTEHNIČKI FAKULTET

Dino Kosić

ADAPTIVNO-PREDIKTIVNI KONTROLER NA BAZI NEURALNIH MREŽA

Doktorska disertacija

Banja Luka, 2017.

UNIVERSITY OF BANJA LUKA
FACULTY OF ELECTRICAL ENGINEERING

Dino Kosić

NEURAL NETWORK BASED ADAPTIVE-PREDICTIVE CONTROLLER

Doctoral dissertation

Banja Luka, 2017.

Informacije o mentoru i disertaciji

Mentor: dr Milorad Božić, redovni profesor, Elektrotehnički fakultet, Univerzitet u Banjoj Luci

Naslov doktorske disertacije: Adaptivno-prediktivni kontroler na bazi neuralnih mreža

Rezime: Disertacija predstavlja rezultat istraživanja primjene vještačkih neuralnih mreža u različitim strategijama automatskog upravljanja, te rada na razvoju brzog postupka obučavanja neuralne mreže i/ili takve strukture koja bi omogućila brže obučavanje, a sa svrhom primjene u adaptivnom upravljanju. Prva poglavlja predstavljaju teoretski uvod u glavna polja istraživanja: automatsko upravljanje, naročito adaptivne i prediktivne varijante istog; te vještačke neuralne mreže, sa posebnim naglaskom na mreže sa funkcijama radijalne baze. Na ovim temeljima je razvijena kasnije prikazana nova arhitektura neuralne mreže sa Gausovim aktivacionim funkcijama - FCRBFN, koja se obučava neiterativnim postupkom (u jednom koraku). Izvršeni su eksperimenti poređenja sposobnosti takve mreže po pitanju brzine i preciznosti obučavanja (regresije), kao i generalizacije, sa nekoliko standarno korištenih aproksimatora/klasifikatora i pokazano je da nova mreža daje bolje rezultate. Takva mreža je dalje korištena u prediktivnom upravljanju, gdje je takođe pokazala odlične eksperimentalne rezultate. Konačno, razvijena je i posebna strategija adaptivno-prediktivnog upravljanja sa FCRBFN, koja je u poređenju sa radom standardnog PID, te aktuelnog *podacima-vođenog* kontrolera pokazala odlične rezultate u primjeni na standardnom laboratorijskom sistemu sa tri rezervoara, kako bez šuma mjerenja, tako i u prisustvu šuma.

Ključne riječi: adaptivno upravljanje, prediktivno upravljanje, neuralne mreže

Naučna oblast: Tehnološke nauke

Naučno polje: Automatizacija, robotika, kontrolni inženjering

Klasifikaciona oznaka: T 125

Information about mentor and dissertation

Mentor: dr Milorad Božić, full-time professor, Faculty of electrical engineering, University of Banja Luka

Title of doctoral dissertation: Neural network based adaptive-predictive controller

Abstract: The dissertation is the result of a research in application of artifical neural networks in different strategies of automatic control, and of a work related to the development of a fast training method and/or such network structure that would enable faster training, all towards usage in adaptive control. The initial chapters give theoretical introduction to the main fields of research: automatic control, especially its adaptive and predictive variants; and artificial neural networks, with strong accent on radial basis function networks. Based on that, a novel architecture of a neural network with Gaussian activation functions, trained with non-iterative (single step) method was developed and it is presented as FCRBFN. To test that network's abilities in regard of its training speed and precision, as well as generalisation precision, and to compare them with those of various standard approximators/classifiers, several experiments were conducted and they have shown better results by FCRBFN. That network was then used in predictive control and experiments again showed excellent results. Finally, a novel strategy of adaptive-predictive control using FCRBFN was developed and tested. Compared to the standard PID controller, as well as modern *data-driven* controller, it produced excellent results in experiments on a standard laboratory three tank system, with and without measurement noise.

Keywords: adaptive control, predictive control, neural networks

Scientific area: Technological sciences

Scientific field: Automation, robotics, control engineering

Classification code: T 125

Tati i mami

Sadržaj

Tabele

Slike

Zahvalnice

Ako bih krenuo da navodim samo imena svih ljudi kojima dugujem nemjerljivu zahvalnost za pomoć, podršku, strpljenje, a ponajviše vjeru u mene, date u toku puta koji me je doveo do izrade ove disertacije, trebalo bi mi mnogo više prostora od ovog koji je predviđen za te stvari, a čak i kad bih se odlučio na to, nijedan redoslijed ne bi bio pošten.

Zato se nadam da svako ko zaslužuje da bude na tom spisku, bio porodica, prijatelj ili nastavnik, zna da, i bez nabrajanja, pamtim i zahvalan sam za sve što su učinili za mene.

Banja Luka, Bosna i Hercegovina | Dino Kosić
mart, 2017.

1. Uvod

Od postanka čovjek je imao tendenciju da utiče na okolinu i procese oko sebe da bi ostvario neku korist, tako da se može reći da je upravljanje, iako je kao zasebna naučna disciplina relativno mlado, zapravo staro koliko i čovjek. Do industrijske revolucije upravljanje je uglavnom bilo izvođeno direktno od strane čovjeka, a tad je počeo razvoj naprava koje bi smanjile ili potpuno isključile potrebu za ljudskom intervencijom u upravljanju procesima. Tako je započeo razvoj automatskog upravljanja. Kako se automatsko upravljanje uključivalo u sve više poslova sa daljim razvojem industrije, tako se pojavila i potreba za formalizacijom koja je rezultovala u razvoju nauke o automatskom upravljanju.

Prvi upravljački sistemi - *kontroleri*, su bili potpuno mehanički i njihova realizacija je zavisila prevashodno od samog objekta upravljanja, tj. za svaki sistem se posebno projektovao kontroler. Nauka o automatskom upravljanju je, između ostalog, dala i neke opštije principe projektovanja kontrolera koji su mogli biti primijenjeni na više različitih sistema koji pokazuju slične karakteristike. Postavljeni su matematički temelji za takvo projektovanje, kao i za utvrđivanje karakteristika samih procesa. U skladu sa kompleksnošću prirodnih procesa, osnovne varijante kontrolera su omogućavale postizanje željenih performansi sistema za relativno mali skup radnih uslova. Logična potreba da kontroleri postanu *otporniji* na promjene uslova, koje se u praksi prirodno dešavaju, dovela je do razvoja adaptivnih kontrolera, odnosno kontrolera koji imaju mogućnost promjene svoje strukture ili svojih parametara, u svrhu prilagođavanja promjenama radnih uslova cijelog sistema.

Još jedna od naprednih tehnika u upravljanju je prediktivno upravljanje na bazi modela. Ideja kod ovakvog načina upravljanja je da se na osnovu matematičkog modela samog procesa utvrdi buduće ponašanje tog procesa, a odatle da se izvede zakon upravljanja, najčešće optimalan u skladu sa nekom funkcijom cilja. Kao i kod adaptivnog upravljanja, i za ovaj metod razvijeno je više različitih pristupa.

Kako je pokazano u Poglavlju 2, za razvoj uspješnog kontrolera jako je važno poznavanje samog procesa kojim se upravlja. Najpoznatiji način formalizacije tog znanja je matematičko modelovanje. Dat je poseban pregled strategija upravljanja koje su bazirane upravo na modelu sistema (adaptivno i prediktivno upravljanje). Ovo poglavlje sadrži i opis linearnih i nelinearnih sistema, načina ispitivanja stabilnosti svake od tih vrsta sistema, te načine upravljanja njima.

Paralelno sa željom da upravlja, čovjek je oduvijek imao i želju da saznaje, kako o svijetu oko sebe, tako i o samom sebi. U toj želji za samospoznajom, čovjek se bavio i pojmovima učenja i inteligencije. Kao i kod mnogih drugih nauka, postojala je potreba dokazivanja znanja kroz reprodukciju prirodnih procesa, te je tako započet razvoj vještačkih sistema koji bi imitirali učenje (ili imali sposobnost učenja), pa čak i inteligenciju, što je dovelo do razvoja nauke o vještačkoj inteligenciji. Iako ni danas ne postoji jedinstvena definicija same inteligencije, pa ni sistema koji bi se smatrao inteligentnim, i uz konstantne polemike o moralnosti pokušaja stvaranja takvih sistema, nauka o vještačkoj inteligenciji se stalno razvija i daje doprinose. Jedan od sistema koji je nastao kao pokušaj imitacije ljudskog mozga su vještačke neuralne mreže. Po nastanku su izazvale visoka očekivanja, ali već rani radovi nastali kao rezultat istraživanja vještačkih neuralnih mreža su pokazali velike nedostatke u odnosu na inspiraciju iz prirode, što je tako drastično uticalo na prvobitnu popularnost da je skoro ugušilo dalji razvoj. Nakon početnog zatišja, naučnici su se vratili proučavanju i razvoju vještačkih neuralnih mreža, i iako to nije dovelo do sistema koji bi imitirali ljudski mozak, pokazano je dosta korisnih svojstava takvih mreža. Jedno od njih je sposobnost aproksimacije funkcija, što je posebno zanimljivo u teoriji automatskog upravljanja. Naime, iako se nauka stalno razvija, a s njom i znanje o raznim procesima, i dalje je mnogo procesa o kojima se ne zna dovoljno da bi se napravio dobar matematički model, a česta je i situacija da postoji znanje, ali je sam proces takav da se dobija vrlo kompleksan model. U tim situacijama se pribjegava modelovanju tipa crne kutije (eng. *black-box modelling*) kod kojeg je važno samo utvrditi način na koji promjena ulaza sistema utiče na promjenu izlaza sistema, bez saznanja o fizikalnim i ostalim aspektima procesa koji dovodi do tog uticaja. U Poglavlju 3 je, pored istorijskog pregleda, te pregleda različitih struktura vještačkih neuralnih mreža, pokazana i sposobnost neuralnih mreža da se koriste za takvo modelovanje, kao i načini da mreže steknu takvo znanje, tj. da budu obučene. Načini obučavanja su uglavnom iterativni metodi, koji su uprkos predstavljenim poboljšanjima po pitanju brzine i sigurnosti konvergencije, i dalje relativno spori i računski kompleksni. Nakon toga je u Poglavlju 4 dat pregled jedne posebne klase vještačkih neuralnih mreža, mreža sa funkcijama radijalne baze, koje pokazuju određene prednosti upravo pri aproksimaciji funkcija.

Korištenje vještačkih neuralnih mreža u upravljanju je prvi put predstavljeno u [1]. Otad se izučavanjem upravljanja pomoću neuralnih mreža bavio veliki broj istraživača. Početkom XXI vijeka može se reći da su neuralne mreže postale sastavni dio teorije upravljanja kao prirodno proširenje adaptivnog upravljanja sistemima koji su nelinearni preko svojih podesivih parametara. Stanje upravljanja pomoću neuralnih mreža u tom periodu je detaljno opisano u [2]. Pregled pionirskih radova u upravljanju pomoću neuralnih mreža je dat u [3] i [4], gdje su istaknuti mnogi problemi sa kojima se treba izboriti upravljanje u zatvorenoj petlji. Primjena neuralnih mreža u zatvorenoj petlji je fundamentalno drugačija od primjena u otvorenoj petlji (npr. klasifikacije ili obrade slika). Osnovna strategija za podešavanje osnovne višeslojne neuralne mreže je propagacija greške unazad. Osnovni problemi koje treba riješiti pri upravljanju pomoću neuralnih mreža u zatvorenoj petlji uključuju inicijalizaciju težina koja obezbjeđuje stabilnost, određivanje gradijenta za uspješno podešavanje propagacijom greške unazad, razmatranje početnog off-line obučavanja, efikasnija propagacija greške unazad i sl. Otad su razvijeni mnogi pristupi rješavanju ovih problema.

Inicijalni radovi na primjeni neuralnih mreža u upravljanju su bili vezani za identifikaciju sistema i za indirektno upravljanje na bazi identifikacije. U primjenama u zatvorenoj petlji neophodno je pokazati stabilnost greške identifikacije, kao i ograničenost težina veza same mreže. Ovi dokazi, kao i garancija performanse i robustnosti nisu bili dio prvih radova. Nesigurnost oko inicijalizacije težina je dovela do potrebe za početnim off-line podešavanjem koje može dati značajne strukturalne informacije, a što je formalizovano u [5]. Modeli na bazi neuralnih mreža se mogu izvesti na osnovu mjerenih podataka sa ulaza i izlaza sistema. Naročito je povoljno to što se, uglavnom, u procesu već nalazi kontroler (u industriji je to najčešće i dalje neki tradicionalni kontroler, npr. P/PI/PID kontroler) koji stabilizuje objekat upravljanja i daje neko osnovno, iako ponekad ne baš efikasno, upravljanje. Podaci koji se prikupljaju sa takvog sistema koji ima linearno upravljanje se pokazuju kao bolji za obučavanje neuralne mreže jer su izlazi mnogo sličniji onima koji će se postići sa boljim upravljanjem, nego izlazima u otvorenoj sprezi [6, 7, 8, 9, 10, 11, 12].

Dalji radovi na primjeni neuralnih mreža u upravljanju su se pozabavili problemima strukture sistema sa zatvorenom petljom i stabilnosti. U [13] i [14] su predstavljeni važni koraci u određivanju osobina sistema sa neuralnim mrežama, poput minimalnosti i jedinstvenosti idealnih težina veza, kao i opservabilnosti dinamičkih neuralnih mreža. U [15] i [16] je dat naglasak na pronalaženju gradijenata potrebnih za podešavanje neuralne mreže propagacijom greške unazad. Te mreže, kad se uključi i dinamika procesa postaju rekurentne, pa gradijenti zadovoljavaju diferencne ili diferencijalne jednačine sistema, pa ih je teško pronaći. U [17] je pokazano da je poznavanje Jakobijana procesa često dovoljno za garantovanje odgovarajućih performansi u zatvorenoj petlji.

Nastavak razvoja primjene neuralnih mreža u upravljanju je doveo do kombinovanja sa drugim pristupima u upravljanju. U sprezi sa dinamičkom inverzijom, u [18, 19, 20] su neuralne mreže korištene za upravljanje letjelicama i raketama. Linearizacija u povratnoj petlji pomoću neuralnih mreža je korištena u [21, 22]. Neuralne mreže su korištene i sa koračanjem unazad (eng. *backstepping*) u [23, 24, 25, 26].

Međutim, kako je rečeno ranije, aproksimacione sposobnosti neuralnih mreža su osnova njihove primjene u upravljanju. Na osnovu ovoga i analize dinamike greške, predstavljene su razne modifikacije algoritma propagacije greške unazad koje garantuju stabilnost u zatvorenoj petlji, kao i ograničenost težina. Ovo je srodno uslovima dodanim u adaptivno upravljanje koji čine algoritme robustnim po pitanju visokofrekventne nemodelovane dinamike. U [27] su korištene RBF mreže u upravljanju i pokazano je kako izabrati aktivacione funkcije mreže. U [28] i [29] je korišten projekcioni metod za korekciju težina. U [30] je korištena propagacija greške unazad sa *e-mod* uslovom, predstavljenim u [31]. Ovakve neuralne mreže su linearne po pitanju nepoznatih parametara, a u [32] je pokazano da takve mreže imaju ograničenje preciznosti aproksimacije do reda $L^{-\frac{2}{n}}$, gdje je L broj skrivenih slojeva, a n broj ulaza. Ovo ograničenje prevazilaze neuralne mreže koje su nelinearne po pitanju nepoznatih parametara, a koje su prvi put korištene u [33]. Dinamičke/rekurentne neuralne mreže su u različitim varijantama korištene u upravljanju u [34, 35, 36, 37].

Mane korištenja neuralnih mreža su u algoritmima obučavanja. Najčešće korištene varijante algoritama obučavanja su zasnovane na iterativnim, gradijentim metodama koje su i računski i vremenski zahtjevne, uz prilično striktne uslove za konvergenciju. Ukoliko se neuralna mreža koristi kao unaprijed obučena, tj.

samo za modelovanje u prediktivnom upravljanju, pa se samo obučavanje vrši prije uključivanja u upravljačku petlju, ovaj problem i nije tako značajan, jer se smatra da je off-line priprema kontrolera jednokratan proces, te je njegovo trajanje zanemarivo u odnosu na trajanje korištenja u upravljanju. Sama arhitektura sistema koja omogućava prilagodbu parametara neuralne mreže tokom rada je jednostavna, jer neuralna mreža je inherentno spremna za dodatni trening ukoliko se njena struktura ne mijenja. Stoga se problem adaptacije ovakvog kontrolera svodi na brzinu dodatnog obučavanja. Iako su razvijeni mnogi brzi metodi za prevazilaženje problema sa visokim brojem iteracija (samim tim i većim trajanjem obučavanja), te sa sigurnošću konvergencije, kao i mnoge varijacije arhitektura neuralnih mreža koje se koriste kao adaptivno-prediktivni kontroleri, i dalje su važna istraživanja koja će eliminisati te probleme [38, 39, 40, 41, 42, 43].

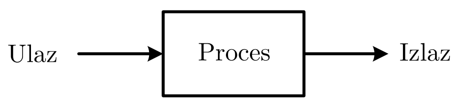
Istraživanja takve arhitekture neuralne mreže koja omogućava brzo i efikasno obučavanje, naročito ako bi se eliminisala potreba za iterativnim postupkom koji bi bio zamijenjen postupkom koji se izvršava u jednom koraku, što bi povećalo sposobnost adaptacije, ali i smanjilo vrijeme preprocesiranja, mogu značajno da doprinesu na polju adaptivno-prediktivnog upravljanja. Štaviše, takva mogućnost adaptacije, koja bi vršila doobučavanje neuralne mreže u svakom koraku dodavanjem novog znanja u istom koraku kada se ono i stekne, bi omogućila da se i samo modelovanje sistema obavlja u realnom radu, tako da bi se off-line treniranje značajno smanjilo ili potpuno eliminisalo.

Sve ovo je poslužilo kao motivacija za istraživanje koje je rezultovalo u doprinosu predstavljenom u ovom radu u Poglavljima 5, 6 i 7 gdje je dokazana radna hipoteza: *Za upravljanje procesima moguće je projektovati adaptivno-prediktivni kontroler na bazi višeslojne neuralne mreže sa takvom strukturom koja obezbjeđuje brzu i efikasnu adaptaciju dodavanjem novostečenog znanja u istom koraku kad se novo znanje i stekne, bez iterativnih postupaka i bez neophodnosti za poznavanjem modela objekta upravljanja.*

2. Automatsko upravljanje

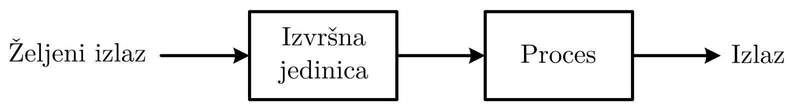
Upravljanje je naučna disciplina koja je bavi razumijevanjem i rukovođenjem segmentima okoline, koji se najčešće nazivaju *sistemima*, radi postizanja unapred definisanih ciljeva [44, 45, 46, 47, 48, 49, 50, 51]. Razumijevanje i rukovođenje su komplementarni ciljevi jer je za uspješno upravljanje potrebno da sistemi budu shvaćeni i modelovani, mada je često potrebno upravljati i sistemima koji se slabo razumiju (npr. hemijski procesi).

Upravljački sistem je skup međusobno povezanih komponenata koje formiraju konfiguraciju koja će proizvesti željeni odgovor sistema. Osnova za analizu sistema je preuzeta iz teorije linearnih sistema koja podrazumijeva uzročno-posljednične odnose komponenti sistema. U skladu s tim, komponenta ili proces kojim se upravlja može se predstaviti blok-dijagramom kao na Slici 2.1. Odnos ulaz-izlaz predstavlja uzročno-posljedičnu prirodu procesa, koji zauzvrat predstavlja obradu (procesiranje) ulaznog signala koja rezultuje izlaznim signalom.



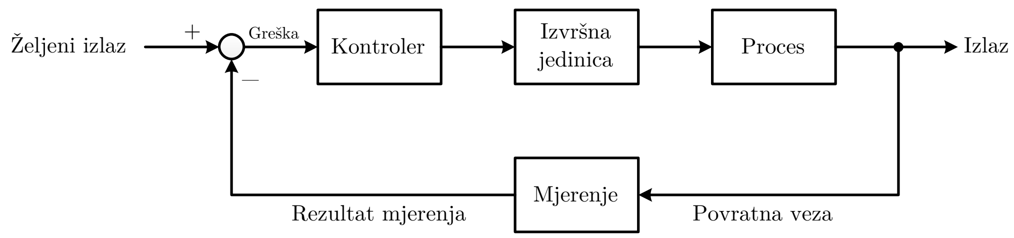
Slika 2.1 *Blok-dijagram procesa kojim se upravlja*

Upravljački sistem sa otvorenom petljom (eng. *open-loop*) je sistem bez povratne sprege i koristi kontroler ili izvršnu jedinicu (upravljački aktuator) za postizanje željenog odziva, kako je prikazano na Slici 2.2.



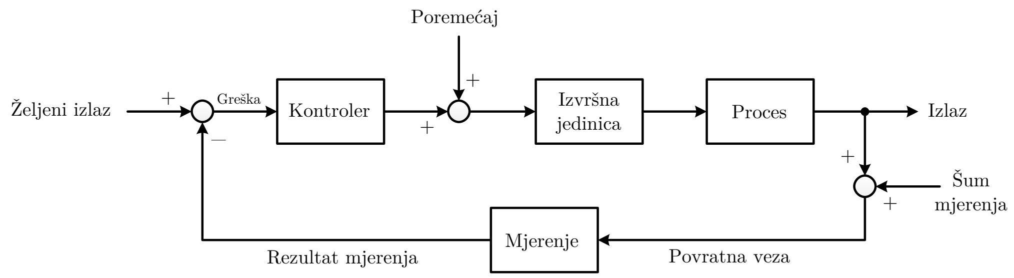
Slika 2.2 *Blok-dijagram upravljačkog sistema sa otvorenom petljom*

Za razliku od sistema sa otvorenom petljom, sistem sa zatvorenom petljom (eng. *closed-loop*) koristi dodatno mjerenje stvarnog izlaza za poređenje sa željenom vrijednošću izlaza. Mjerenje izlaza daje mjereni signal (eng. *feedback signal*). Jednostavan sistem sa zatvorenom petljom (povratnom spregom) je prikazan na Slici 2.3. Upravljački sistemi sa zatvorenom povratnom spregom su upravljački sistemi koji teže da održe unapred definisan odnos sistemskih promjenljivih poređenjem tih promjenljivih i korištenjem njihove razlike kao sredstva upravljanja, a u svrhu stalnog smanjivanja te razlike. Koncept povratne sprege je osnova za analizu i sintezu sistema upravljanja.



Slika 2.3 *Blok-dijagram upravljačkog sistema sa zatvorenom petljom*

Upravljanje u zatvorenoj petlji ima mnoge prednosti u odnosu na upravljanje u otvorenoj petlji, poput sposobnosti smanjivanja uticaja spoljašnjih uticaja (smetnji), te šuma mjerenja koji su neizbježni u realnim primjenama i na koje se mora obratiti pažnja u praktičnom razvoju sistema upravljanja. Smetnje i šum mjerenja se u blok-dijagramu prikazuju kao ulazi, kako je prikazano na Slici 2.4.



Slika 2.4 *Blok-dijagram upravljačkog sistema sa zatvorenom petljom u prisustvu spoljašnjih smetnji i šuma mjerenja*

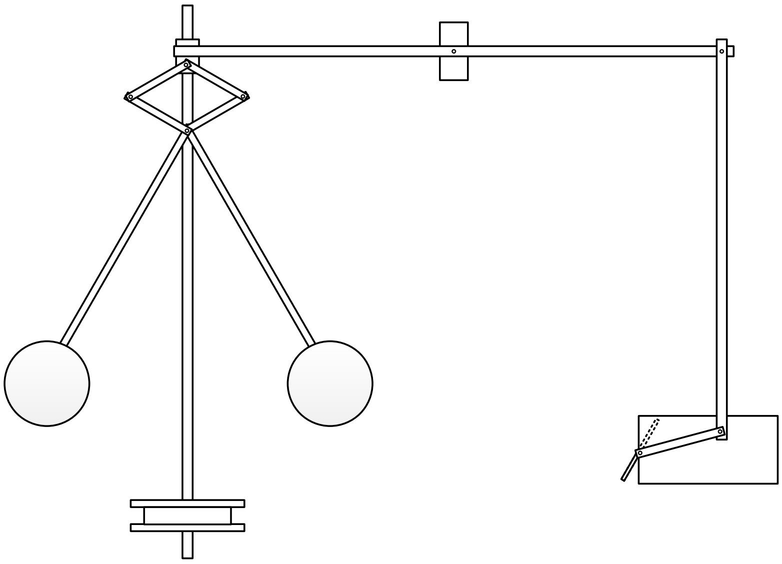
2.1 Kratka istorija automatskog upravljanja

Jedna od prvih primjena upravljanja sa zatvorenom petljom je regulacioni mehanizam sa plovkom za upravljanje vodenim satovima u antičkoj Grčkoj [52]. Sličan sistem Grci su koristili i za održavanje nivoa ulja u uljanim lampama. Heron iz Aleksandrije je u prvom vijeku nove ere objavio knjigu naziva *Pneumatica* u kojoj je iznio nekoliko primjera mehanizama sa plovkom.

Prvi upravljački mehanizam sa povratnom petljom u modernoj Evropi je Drebelov temperaturni regulator. Papin je izumio prvi regulator pritiska za parne bojlere, sigurnosni mehanizam sličan ventilu ekspres-lonca.

Kao prvi automatski sistem upravljanja sa povratnom petljom korišten u industrijskom procesu se univerzalno smatra *centrifugalni regulator* (regulator sa kuglicama, eng. *centrifugal flyball governor*) koji je 1788. godine izumio Džejms Vat za kontrolu brzine parne mašine. Potpuno mehanički uređaj čiji je šematski prikaz dat na Slici 2.5 je mjerio brzinu izlazne osovine i koristio kretanje kuglica za podešavanje ventila za paru, a samim tim i količinu pare koja je dolazila u mašinu. Kad se brzina okretanja poveća, kuglice se podižu i povlače sistem veza koji zatvara ventil i usporava rad mašine.

U narednom vijeku su razvijani različiti sistemi automatskog upravljanja kroz intuitivno i inovativno razmišljanje onih koji su se bavili razvojem. Pokušaji



Slika 2.5 *Vatov centrifugalni regulator*

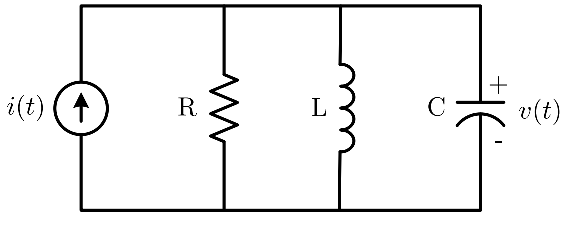
da se na taj način dođe do preciznijih sistema su doveli do otkrića prelaznih oscilacija u odzivu sistema, te nestabilnih sistema. Zbog toga se pojavila potreba za razvojem teorije automatskog upravljanja. Sredinom XIX vijeka Maksvel je formulisao matematičku teoriju u upravljanju korištenjem diferencijalnih jednačina [53]. U istom periodu je Višnjegradski formulisao matematičku teoriju regulatora [54].

U periodu između dva Svjetska rata značajan doprinos dali su Bode [55], Nikvist i Blek [56] radeći za Belovu laboratoriju. Tokom Drugog svjetskog rata je dodatno do izražaja došla potreba za upravljanjem i teorijom koja stoji iza njega, tako da je u tom periodu automatsko upravljanje i postalo zasebna inženjerska disciplina.

2.2 Matematički modeli sistema

Za razumijevanje kompleksnih sistema i upravljanje njima potrebno je doći do kvantitativnih matematičkih modela tih sistema koji opisuju veze između promjenljivih u tom sistemu. S obzirom da su razmatrani sistemi po prirodi dinamički, uobičajen način opisivanja je pomoću diferencijalnih jednačina.

Tipičan primjer je RLC električno kolo, prikazano na Slici 2.6.



Slika 2.6 *RLC električno kolo*

Na osnovu Kirhofovih zakona dobija se

$$\frac{v(t)}{R} + C\frac{dv(t)}{dt} + \frac{1}{L}\int_0^t v(t)dt = i(t). \tag{2.1}$$

Ako je struja strujnog izvora konstantna, onda napon na kondenzatoru ima oblik

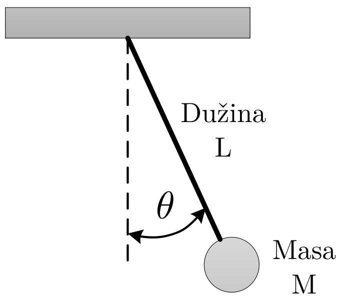
$$v(t) = Ke^{-\alpha t}\sin(\beta t + \theta), \tag{2.2}$$

gdje su K, α, β i θ konstante koje zavise od vrijednosti parametara u kolu.

Do ovakvog rješenja je moguće doći klasičnim metodama ili primjenom Laplasove transformacije.

U praksi, zbog kompleksnosti realnog sistema, ali i nepoznavanja svih faktora koji utiču na njegovo ponašanje, neophodno je uvesti pretpostavke o radu samog sistema. Većina realnih sistema nije linearna, što znatno otežava proces modelovanja, kao i rješavanja diferencijalnih jednačina koje se dobiju pri modelovanju, ali su ti sistemi uglavnom linearni u nekom opsegu vrijednosti njihovih promjenljivih. Poznato je da je sistem linearan ako za njega važi princip superpozicije, te homogenosti.

Tipičan primjer je klatno sa kuglicom, prikazano na Slici 2.7.



Slika 2.7 *Klatno*

Moment sile kuglice je

$$T = MgL\sin\theta, \tag{2.3}$$

gdje je g gravitaciono ubrzanje Zemlje.

Zavisnost momenta od ugla je očito nelinearna, međutim može se pokazati da je u okolini ravnotežne tačke ova zavisnost skoro pa linearna.

Proces linearizacije sistema se zasniva na razvoju funkcije $f(x)$ u Tejlorov red u okolini neke tačke x_0 prema

$$f(x) = f(x_0) + \left.\frac{df}{dx}\right|_{x=x_0}\frac{x - x_0}{1!} + \left.\frac{d^2f}{dx^2}\right|_{x=x_0}\frac{(x-x_0)^2}{2!} + \cdots. \tag{2.4}$$

Tangenta u okolini tačke x_0 je dobra aproksimacija krive na malom opsegu $(x - x_0)$, tako da se zanemarivanjem članova sume iz (2.4) koji sadrže izvode višeg reda dobija

$$f(x) \approx f(x_0) + \left.\frac{df}{dx}\right|_{x=x_0}(x - x_0). \tag{2.5}$$

Ako se ovo primijeni na moment sile kuglice u okolini ravnotežnog stanja $\theta_0 = 0°$, dobija se

$$T = T_0 + MgL \left. \frac{\partial \sin\theta}{\partial\theta} \right|_{\theta=\theta_0} (\theta - \theta_0). \tag{2.6}$$

Kako je moment sile za θ_0 jednak nuli, tj. $T_0 = 0$, dobija se

$$T = MgL \cos\theta_0 (\theta - \theta_0) = MgL\theta. \tag{2.7}$$

Ovo je prihvatljiva aprkosimacija jer je za njihanje od $\pm 30°$ greška manja od 5%.

U skladu sa ovim, modelovanje sistema se izvodi u nekoliko koraka:

1. Definišu se sistem i njegove komponente,
2. Pomoću osnovnih principa fizike i prihvatljivih pretpostavki utvrđuju se zakonitosti i vrši linearizacija,
3. Na osnovu prethodnog formulišu se diferencijalne jednačine koje predstavljaju matematički model,
4. Korištenjem matematičkih alata, poput Laplasove transformacije, dolazi se do rješenja koje opisuje ponašanje sistema.

Ukoliko se koristi Laplasova transformacija, te odredi odnos Laplasovih transformacija izlaza i ulaza sistema, dobija se *funkcija prenosa sistema*. Za RLC kolo opisano jednačinom (2.1) se na taj način dobija

$$H(s) = \frac{V(s)}{I(s)} = \frac{RLs}{RLCs^2 + Ls + R}. \tag{2.8}$$

Funkcija prenosa sistema se može smatrati načinom modelovanja sistema u smislu da daje informaciju o vezi ulaza i izlaza, ali ne daje nikakav opis unutrašnje strukture i ponašanja sistema. Često se u predstavljanju sistema blok-dijagramima (poput onih sa Slika 2.1, 2.2, 2.3 i 2.4) unutar samog bloka koji opisuje određeni sistem upisuje upravo prenosna funkcija tog sistema.

Stotinjak godina nakon uvođenja diferencijalnih jednačina za opisivanje ponašanja sistema, razvijen je način modelovanja sistema kod kojeg se jedna ili više komponenata mijenja tokom vremena - *model u prostoru stanja*. Stanje sistema je skup promjenljivih čije vrijednosti zajedno sa vrijednostima ulaznih signala, a na osnovu jednačina koje opisuju dinamiku, određuju buduće stanje sistema i njegovih izlaza. Uobičajeno je da se stanja sistema označavaju sa $x_k(t)$, a da se jednačine u prostoru stanja zapisuju u matričnom obliku

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \tag{2.9}$$

gdje je $\boldsymbol{x}$ vektor promjenljivih stanja, $\dot{\boldsymbol{x}}$ vektor prvih izvoda promjenljivih stanja, a $\boldsymbol{u}$ vektor ulaznih signala sistema. Matrica $\boldsymbol{A}$ se naziva *matrica sistema*. Karakteristične (sopstvene) vrijednosti te matrice predstavljaju *polove* sistema, a karakteristični polinom te matrice $p(\lambda) = \det(\boldsymbol{A} - \lambda\boldsymbol{I})$, gdje je $\boldsymbol{I}$ jedinična matrica istog reda kao matrica $\boldsymbol{A}$, je ujedno i karakteristični polinom sistema. Polovi sistema određuju dinamičke karakteristike sistema, o čemu će biti riječi u nastavku.

Jednačina izlaza u prostoru stanja je data sa

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}, \tag{2.10}$$

gdje je $\boldsymbol{y}$ vektor izlaznih signala sistema.

Primjenom Laplasove transformacije na jednačine u prostoru stanja, lako se dolazi do prenosne funkcije sistema

$$H(s) = \boldsymbol{C}\left(s\boldsymbol{I} - \boldsymbol{A}\right)^{-1}\boldsymbol{B} + \boldsymbol{D}, \tag{2.11}$$

gdje je $\boldsymbol{I}$ jedinična matrica istog reda kao matrica $\boldsymbol{A}$. Iz (2.11) se vidi da je nazivnik prenosne funkcije sistema karakteristični polinom tog sistema.

2.3 Linearni sistemi

Na početku poglavlja je rečeno da upravljački sistem služi za dobijanje željenog izlaza sistema. U skladu s tim, specifikacija dizajna upravljačkog sistema se najčešće zadaje na osnovu željenog vremenskog odziva, odnosno ponašanja izlaza sistema tokom vremena i to u prelaznom režimu (efekat koji prolazi tokom vremena), te u ustaljenom režimu (uglavnom u vidu preciznosti). Rijetko su te specifikacije striktno određene jer bez detaljnog poznavanja ponašanja sistema nije moguće ni znati da li je takve zahtjeve moguće ispuniti. Zato se specifikacije daju kao spisak željenih performansi, uz dovoljno prostora za kompromis u postizanju što bližih rezultata.

Za potrebe upoznavanja ponašanja sistema pri različitim signalima na ulazu, pri čemu treba voditi računa i o tome da stvarni ulazni signal nekad i nije unapred poznat, koriste se tehnike proučavanja odziva sistema na standardne testne signale na osnovu kojih se može doći do zaključka o odzivu sistema na proizvoljne signale. Pored toga, često su stvarni signali slični tim standardnim testnim ili su neka njihova kombinacija.

Prvi od tih često korištenih testnih signala je *jedinični impulsni signal*. Taj signal je baziran na pravougaonoj funkciji

$$f_\epsilon(t) = \begin{cases} \frac{1}{\epsilon}, & -\frac{\epsilon}{2} < t < \frac{\epsilon}{2} \\ 0, & \text{inače} \end{cases} \tag{2.12}$$

gdje je $\epsilon > 0$. Kako se ϵ približava nuli, funkcija $f_\epsilon(t)$ se približava jediničnoj impulsnoj funkciji $\delta(t)$, koja se često naziva i Dirakova funkcija, koja ima sljedeće osobine

$$\int_{-\infty}^{\infty} \delta(t)dt = 1 \tag{2.13}$$

i

$$\int_{-\infty}^{\infty} \delta(t-a)g(t)dt = g(a). \tag{2.14}$$

Neka je prenosna funkcija sistema $G(s)$, a Laplasova transformacija ulaznog signala $R(s)$. Tada je izlaz sistema određen sa

$$y(t) = \mathscr{L}^{-1}\left\{G(s)R(s)\right\} = \int_{-\infty}^{t} g(t-\tau)r(\tau)d\tau. \tag{2.15}$$

Ako je na ulazu impulsni signal, odnosno ako je $r(t) = \delta(t)$ tada je

$$y(t) = \int_{-\infty}^{t} g(t-\tau)\delta(\tau)d\tau = g(t), \tag{2.16}$$

odakle se može zaključiti da je inverzna Laplasova transformacija prenosne funkcije sistema zapravo impulsni odziv sistema. Iz (2.15) je jasno da se na osnovu impulsnog odziva može odrediti odziv na proizvoljan ulazni signal.

Ostali standardni testni signali su oblika

$$r(t) = \begin{cases} t^n, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{2.17}$$

pa je Laplasova transformacija oblika

$$R(s) = \frac{n!}{s^{n+1}}. \tag{2.18}$$

Najčešće korišten je odskočni signal, često nazivan i Hevisajdova funkcija sa oznakom $h(t)$, za koji važi $n = 0$. Kada je $n = 1$ riječ je o funkciji rampe, a ponekad se koristi i parabolička funkcija ($n = 2$).

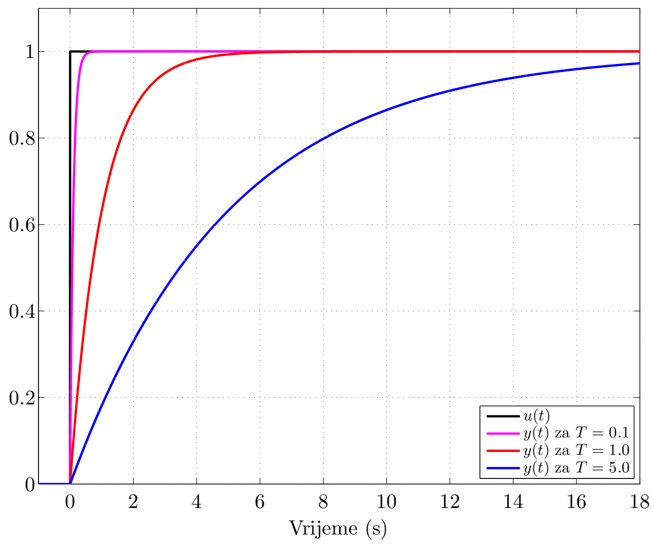
Sistem prvog reda ima prenosnu funkciju

$$G(s) = \frac{K}{Ts+1}. \tag{2.19}$$

Odskočni odziv takvog sistema je dat sa

$$y(t) = K\left(1 - e^{-\frac{t}{T}}\right). \tag{2.20}$$

Vrijednost odziva u ustaljenom režimu je K.



Slika 2.8 *Odskočni odziv sistema prvog reda za različite vremenske konstante*

Na Slici 2.8 prikazan je odskočni odziv sistema prvog reda za 3 različite vremenske konstante. Sa Slike 2.8 se vidi da se povećanjem vremenske konstante

T smanjuje brzina odziva, tj. potrebno je više vremena da se dostigne vrijednost odziva u ustaljenom režimu. Pol sistema je $s = -\frac{1}{T}$, pa je jasno da je pol bliži imaginarnoj osi za veće T. Polovi koji su bliži imaginarnoj osi se često nazivaju spori, a ukoliko su dovoljno sporiji od drugih (najmanje 4 puta bliže imaginarnoj osi), onda se nazivaju i dominantni. Ukoliko sistem ima više polova, uticaj onih koji su nedominantni na brzinu se može zanemariti, ali pri tom treba voditi računa da se ne može zanemariti njihov uticaj na vrijednost odziva u ustaljenom režimu.

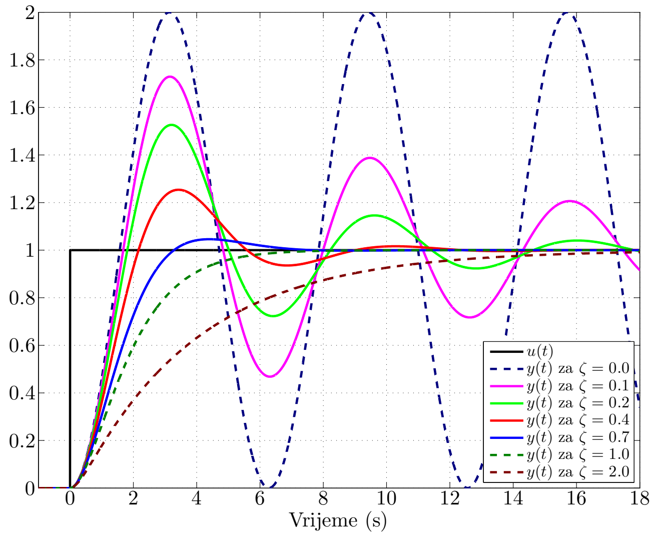
Sistem drugog reda sa kompleksnim polovima ima prenosnu funkciju

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \tag{2.21}$$

gdje je ω_n prirodna frekvencija sistema, a ζ faktor relativnog prigušenja. Odskočni odziv takvog sistema je dat sa

$$y(t) = 1 - \frac{1}{\beta} e^{-\zeta\omega_n t} \sin(\omega_n \beta t + \theta), \tag{2.22}$$

gdje je $\beta = \sqrt{1-\zeta^2}$, $\theta = \arccos\zeta$ i $0 < \zeta < 1$. Vrijednost odziva u ustaljenom režimu je 1.



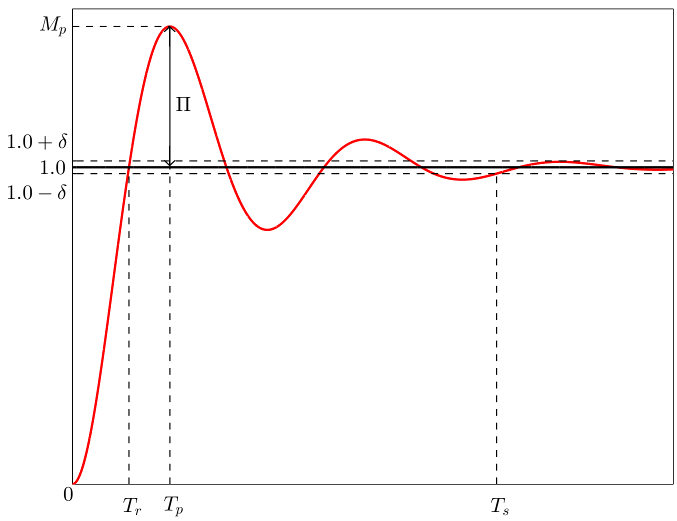
Slika 2.9 *Odskočni odziv sistema drugog reda sa kompleksnim polovima za različite vrijednosti faktora relativnog prigušenja*

Na Slici 2.9 prikazan je odskočni odziv sistema drugog reda sa kompleksnim polovima za različite vrijednosti faktora relativnog prigušenja. Isprekidanom linijom su označeni odzivi za neregularne vrijednosti faktora relativnog prigušenja (0.0, 1.0 i 2.0). U prvom slučaju sistem ima čisto imaginarne polove, te nema prigušenja oscilacija, odnosno odziv je periodičan. U drugom slučaju sistem ima dvostruki realan pol u -1. U poređenju sa prethodno opisanim odzivom sistema prvog reda sa istim polom ($T = 1.0$), vidljivo je da je odziv sporiji, tj. da se dodavanjem još jednog pola koji nije nedominantan usporava odziv sistema. U trećem slučaju sistem ima dva različita realna pola od kojih nijedan nije dominantan, pa je rezultujući odziv sporiji nego sa pojedinačnim polovima.

Standardno mjere performanse koje se mogu dobiti iz odskočnog odziva su:

- *vrijeme porasta* T_r (eng. *rise time*) koje opisuje brzinu odziva sistema, a radi se o vremenu potrebnom da odziv dostigne vrijednost odziva u ustaljenom režimu (ponekad se koristi i vrijeme potrebno odzivu da dostigne od 10% do 90% svoje konačne vrijednosti).
- *vrijeme maksimuma* T_p (eng. *peak time*) koje predstavlja vrijeme kad odziv ima maksimalnu vrijednost. Ukoliko je odziv negativan, onda se radi o minimumu. Dobija se iz
$$T_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}}. \tag{2.23}$$
- *preskok* (kod negativnog odziva *podbačaj*) definisan kao razlika između maksimalne (minimalne) vrijednosti odziva M_p i vrijednosti odziva u ustaljenom režimu. Ponekad se daje i u vidu procenta. Određuje se na osnovu
$$\Pi = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}. \tag{2.24}$$
- *vrijeme smirenja* T_s (eng. *settling time*) koje predstavlja vrijeme potrebno da se odziv *smiri* unutar određenog opsega $[1-\delta, 1+\delta]$ (najčešće $\pm 2\%$ oko vrijednosti odziva u ustaljenom režimu). Obično se uzima da je
$$T_s = 4T_d = \frac{4}{\zeta\omega_n}, \tag{2.25}$$
gdje je sa T_d označena *dominantna vremenska konstanta sistema.*

Sve navedene vrijednosti su prikazane na Slici 2.10.



Slika 2.10 *Standardne mjere performanse na odskočnom odzivu sistema drugog reda sa kompleksnim polovima*

Razlog za razmatranje sistema prvog i drugog reda je što mnogi realni sistemi imaju ili dominantne realne polove ili dominantan par konjugovano-kompleksnih polova što, uz uvažavanje rečenog za nedominantne polove, omogućava da se izvrši jako dobra aprokimacija sistema upravo sistemom prvog, odnosno drugog reda.

2.3.1 Stabilnost linearnih sistema

Pri razmatranju analize i sinteze sistema upravljanja sa povratnom petljom, stabilnost je jedna od najvažnijih stavki. Stabilan sistem rezultuje u ograničenoj vrijednosti izlaza kad je na odgovarajućem ulazu signal ograničene vrijednosti. Ovo je poznato kao princip stabilnosti *ograničen ulaz - ograničen izlaz*.

Iako postoje izuzetne situacije kada je od interesa dobiti nestabilan sistem, većina realnih situacija kao zahtjev ima stabilan sistem. Mnogi fizički sistemi su inherentno nestabilni u otvorenoj petlji, a neki su čak i dizajnirani da budu takvi, npr. moderni borbeni avioni. Bez aktivne povratne petlje takvi sistemi su nefunkcionalni. Dakle, prvi način stabilizacije sistema je povratna petlja.

Pored karakterizacije stabilnosti u smislu da neki sistem ili jeste ili nije stabilan (u tom slučaju riječ je o *apsolutnoj stabilnosti*), može se govoriti i o stepenu stabilnosti, tj. o *relativnoj stabilnosti*. Pioniri razvoja letjelica su bili upoznati sa tim pojmom - što je stabilnija letjetlica, teže je manevrisati njom. Zato su borbeni avioni manje stabilni od komercijalnih.

Stabilnost sistema je direktno povezana sa lokacijom polova sistema u s-ravni. Ovo slijedi iz osobine Laplasove transformacije (tzv. pomijeranje u frekvenciji): Ako je $F(s)$ transformacija signala $f(t)$, onda je $F(s-a)$ transformacija signala $e^{at}f(t)$. Prema tome, ako je pol u lijevoj poluravni, u funkciji prenosa sistema postoji član oblika $F(s+a)$, gdje je $a > 0$, pa će inverzna Laplasova transformacija imati faktor oblika e^{-at} koji se smanjuje prolaskom vremena. U skladu s tim, postojanje pola u lijevoj poluravni za posljedicu ima odziv koji se smanjuje tokom vremena. Analogno tome, postojanje pola u desnoj poluravni znači da će se odziv povećavati tokom vremena. Polovi sa imaginarne ose rezultuju u neutralnom odzivu, tj. nema faktora koji slabi odziv, niti faktora koji pojačava odziv tokom vremena. Odavdje se može zaključiti da stabilni sistemi imaju polove u lijevoj poluravni s-ravni.

Razvijeno je nekoliko metoda za ispitivanje stabilnosti sistema koji se mogu svrstati u dvije glavne grupe: metodi zasnovani na položaju polova u s-ravni i metodi zasnovani na frekvencijskim karakteristikama.

Jedan od najpoznatijih kriterijuma iz prve grupe je Raus-Hurvicov kriterijum stabilnosti. Krajem XIX vijeka, nezavisno jedan od drugog, Hurvic [57] i Raus [58] su objavili radove na temu stabilnosti linearnih sistema. Pomenuti kriterijum se zasniva na određivanju broja nula polinoma u desnoj poluravni bez određivanja samih nula. Koeficijenti karakterističnog polinoma sistema datog sa

$$f(s) = a_n s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_2 s^2 + a_1 s + a_0 \tag{2.26}$$

se slažu u tzv. Rausovu tabelu. Prva dva reda su ispunjena koeficijentima, a elementi ostalih redova se izračunavaju na osnovu elemenata iz prethodna dva reda. Broj redova u tabeli je za jedan veći od stepena karakterističnog polinoma, odnosno od reda sistema. U prvi red se upisuju koeficijenti počev od koeficijenta uz najveći stepen u polinomu (a_n), pa koeficijent uz stepen za 2 manji od najvećeg (a_{n-2}) i tako dalje. Drugi red se popunjava analogno tome, pri čemu se počinje od koeficijenta uz stepen za 1 manji od najvećeg (a_{n-1}). Element u proizvoljnom redu poslije drugog se računa po formuli

$$el = -\frac{D}{el_{-1}}, \tag{2.27}$$

gdje je D determinanta reda 2 u čijoj prvoj koloni su elementi prve kolone iz dva reda koji prethode redu elementa koji se određuje, a druga kolona iz kolone desno od kolone elementa koji se određuje, a iz dva reda koji prethode redu elementa koji se određuje; a el_{-1} element iz prve kolone reda koji prethodi redu elementa koji se određuje. Uobičajeno je da se redovi označavaju redom s^n, s^{n-1}, s^{n-2}, $\ldots$, s^0. U skladu sa prethodno opisanim, Rausova tabela je data u nastavku.

$$\begin{array}{l|lllll}
s^n & a_n & a_{n-2} & a_{n-4} & \cdots \\
s^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} & \cdots \\
s^{n-2} & b_1 & b_2 & b_3 & \cdots \\
s^{n-3} & c_1 & c_2 & c_3 & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
s^2 & * & * & & \cdots \\
s^1 & * & & & \cdots \\
s^0 & * & & & \cdots
\end{array}$$

gdje je

$$b_1 = -\frac{1}{a_{n-1}} \begin{vmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{vmatrix},$$

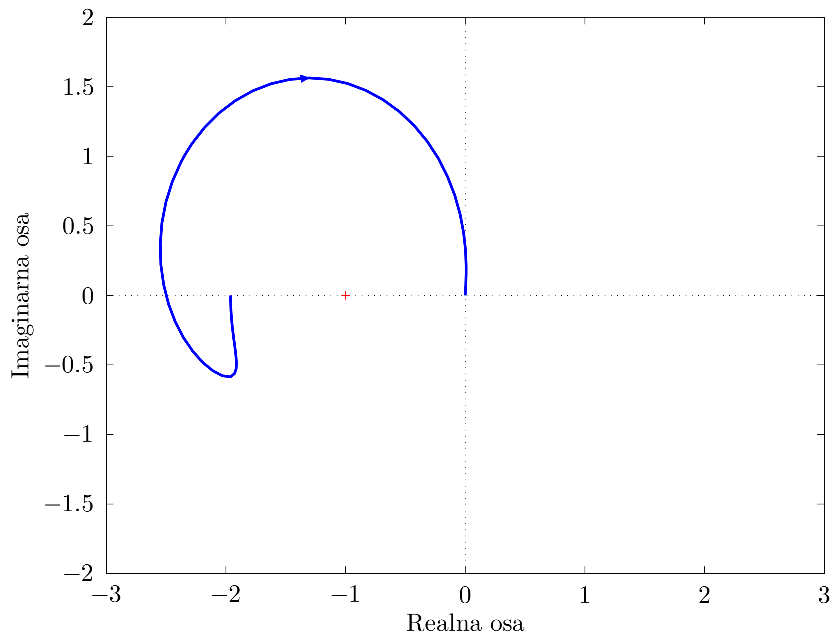
$$b_2 = -\frac{1}{a_{n-1}} \begin{vmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{vmatrix},$$

$$c_1 = -\frac{1}{b_1} \begin{vmatrix} a_{n-1} & a_{n-3} \\ b_1 & b_2 \end{vmatrix}$$

i tako dok se ne popuni prva kolona posljednjeg reda.

Prema Raus-Hurvicovom kriterijumu, broj korijena polinoma $f(s)$ sa pozitivnim realnim dijelom jednak je broju promjena znaka u prvoj koloni Rausove tabele. Stabilan sistem, dakle, ima karakteristični polinom za koji Rausova tabela nema promjena znaka u prvoj koloni.

Iz druge grupe se izdvaja Nikvistov kriterijum stabilnosti [59]. Ovaj kriterijum je zasnovan na Košijevoj teoremi vezanoj za crtanje konture u kompleksnoj s-ravni i broj obilazaka takve konture oko kritične tačke $(-1, 0)$. Pomenuta kontura je skup tačaka u kompleksnoj ravni dobijenih iz prenosne funkcije sistema u otvorenoj petlji $W(s)$ kada važi $s = j\omega$, gdje se ω mijenja od 0 do ∞. Ta kontura predstavlja amplitudsko-faznu frekvencijsku karakteristiku poznatiju kao Nikvistov dijagram, prikazan na Slici 2.11. Nikvistov kriterijum glasi: Da bi sistem sa povratnom petljom bio stabilan, potrebno je i dovoljno da se, pri promjeni učestanosti ω od 0 do ∞, vektor, čiji se početak nalazi u tački $(-1, j0)$, a vrh na amplitudsko-faznoj frekvencijskoj karakteristici $W(j\omega)$ sistema u otvorenoj petlji, obrne suprotno od kretanja kazaljke na časovniku za ugao od $P\pi$, gdje je P broj polova funkcije povratnog prenosa sistema $W(s)$, koji imaju pozitivne realne dijelove, tj. da amplitudsko-fazna frekvencijska karakteristika $W(j\omega)$ obuhvati kritičnu tačku $(-1, j0)$ u pozitivnom smjeru $P/2$ puta.



Slika 2.11 *Primjer Nikvistovog dijagrama*

2.3.2 Upravljanje linearnim sistemima

U prethodnim potpoglavljima je pokazano da i stabilnost i performanse sistema zavise od položaja polova u s-ravni. S obzirom na to da je cilj upravljačkog mehanizma da proizvede takav ulazni signal za sistem, pa da sistem ima željene performanse (uobičajeno date kroz brzinu odziva, veličinu preskoka i grešku u ustaljenom stanju), logičan pristup je podešavanje polova spregnutog sistema (najčešće u formi kao na Slici 2.3). Jedan od metoda sa takvim pristupom je metod zasnovan na *geometrijskom mjestu korijena* [60] (eng. *root locus method*). Geometrijsko mjesto korijena je skup tačaka u s-ravni koje predstavljaju korijene karakterističnog polinoma za različite vrijednosti jednog parametra (najčešće nekog pojačanja K). Takav grafički prikaz daje uvid i u relativnu stabilnost sistema (za koje vrijednosti parametra su svi polovi u lijevoj poluravni, te kolika promjena parametra može biti, pa da sistem ostane stabilan), kao i u performanse samog sistema (polovi koji su bliži imaginarnoj osi više utiču na vrijeme odziva, kompleksni polovi znače da će postojati preskok i oscilacije u odzivu i sl.), a ujedno daje uvid u to kakav kontroler treba dodati sistemu, pa da spregnuti sistem ima željene polove.

Geometrijsko mjesto korijena sistema sa n polova i m nula ima n grana koje polaze od polova sistema u otvorenoj sprezi, od čega $n - m$ grana *odlazi* u beskonačnost, a m grana završava u nulama sistema u otvorenoj sprezi. Grane koje odlaze u beskonačnost imaju asimptote koje zajedno čine pramen polupravih čiji je centar određen direktno položajem polova i nula sistema u otvorenoj sprezi, pri čemu date asimptote dijele krug sa centrom u njihovom centru na jednake dijelove. Neke od grana mogu da imaju zajedničke tačke i te tačke se nazivaju *tačke razdvajanja*. One se takođe određuju na osnovu polova i nula sistema u otvorenoj sprezi. Za svaki pol moguće je odrediti i vektor promjene, odnosno ugao tangente grane koja prolazi kroz taj pol.

Projektovanje kontrolera metodom geometrijskog mjesta korijena se svodi na podesan izbor parametara kaskadnog kompenzatora (povezuje se u kaskadu, odnosno serijski) čija je prenosna funkcija

$$G_c(s) = K\frac{s+a}{s+b}, \tag{2.28}$$

tako da grane geometrijskog mjesta korijena kompenzovanog sistema prolaze kroz željene polove. Ukoliko za parametre kompenzatora važi $|a| < |b|$ kompenzator se naziva diferencijalnim, ovo je zbog činjenice da tačka koja je dalja od imaginarne ose manje utiče na dinamiku procesa, pa se može reći da je dominantno množenje sa s u odnosu na dijeljenje, što odgovara Laplasovoj transformaciji operatora diferenciranja. Analogno tome, kada je $|a| > |b|$, kompenzator se naziva integralnim.

Postoje dva pristupa u određivanju parametara K, a i b. Prvi i matematički jednostavniji je *metod skraćivanja polova*, kod kojeg se a bira tako da bude jednako jednom od polova sistema u otvorenoj sprezi (otud naziv skraćivanje), a onda se K i b računaju na osnovu željenog karakterističnog polinoma. Izbor pola koji će se skratiti se vrši na osnovu analize geometrijskog mjesta korijena kojom se utvrđuje kako grane geometrijskog mjesta korijena treba da budu pomjerene, pa da prolaze kroz željene polove. Drugi princip ne vrši direktno skraćivanje pola, već se parametar a bira tako da bude blizu nekog od polova, tako da grana geometrijskog mjesta korijena koja potiče iz tog pola završi u toj nuli, čime se efektivno poništava uticaj tog pola. Ovaj metod, iako matematički kompleksniji, ne zavisi od subjektivne procjene koji pol treba biti skraćen, već ima potpuno definisan postupak i formule za određivanje svih parametara. S praktične strane gledišta, jako je teško realizovati savršeno skraćivanje polova, pa je to još jedan razlog za korištenje drugog pristupa.

Postoji i sličan pristup projektovanju kaskadnog kompenzatora kad se ne vrši direktno podešavanja polova kompenzovanog sistema, već kad je traženo upravljanje dato u vidu parametara vidljivih na frekvencijskim karakteristikama.

Međutim, u upravljanju industrijskim procesima najzastupljenija vrsta kontrolera je PID kontroler. Akronim je nastao od riječi Proporcionalno, Integralno i Diferencijalno, odnosno atributa koji opisuju vrstu djelovanja tri dijela ovog kontrolera. U skladu sa rečenim, prenosna funkcija takvog kontrolera je

$$G_c(s) = K_p + \frac{K_I}{s} + K_D s, \tag{2.29}$$

gdje je K_p pojačanje proporcinalnog dejstva, K_I pojačanje integralnog dejstva, a K_D pojačanje diferencijalnog dejstva kontrolera. U praktičnoj izvedbi, diferencijalni član zapravo ima prenosnu funkciju oblika

$$G_d(s) = \frac{K_D s}{\tau_d s + 1}, \tag{2.30}$$

gdje je τ_d uobičajeno mnogo manje od vremenske konstante samog procesa.

Ako je $K_D = 0$, radi se od PI kontroleru, a ako je $K_I = 0$, radi se od PD kontroleru.

Proporcionalno dejstvo je dobilo ime po činjenici da je proporcionalno ulaznom signalu kontrolera (najčešće je to signal greške). Što je veće pojačanje K_p, to je sistem osjetljiviji na grešku, ali preveliko pojačanje može da uzrokuje nestabilnost sistema, o čemu će biti riječi u nastavku. Proporcionalno dejstvo uglavnom uzrokuje grešku u ustaljenom režimu.

Integralno dejstvo je proporcionalno sa sumom trenutnih grešaka tokom vremena i ono ubrzava odziv sistema, te eliminiše grešku u ustaljenom režimu, ali može da uzrokuje preskok i pojavu oscilacija u odskočnom odzivu.

Diferencijalno dejstvo je proporcionalno sa brzinom promjene greške, pa na osnovu predviđanja ponašanja sistema popravlja vrijeme smirenja i stabilnost sistema. Prema [61] oko 25% kontrolera sadrži diferencijalno dejstvo. Ovo je posljedica ranije pomenute praktične izvedbe.

Činjenica da PID kontroleri imaju izuzetno veliku primjenu u industriji je posljedica toga da imaju dobre performanse u širokom opsegu radnih uslova, kao i toga da su funkcionalno jednostavni što omogućava inženjerima lakše projektovanje i rad. Projektovanje se svodi na određivanje tri parametra samog kontrolera K_p, K_I i K_D. Ta procedura se često naziva *podešavanje PID kontrolera* (eng. *PID tuning*). Postoji mnogo načina za takvo podešavanje, a među najpoznatijim su *ručno podešavanje* (pojačanja PID kontrolera se određuju metodom pokušaja i promašaja, bilo u simulaciji, bilo na stvarnom sistemu, bez mnogo analitičkog istraživanja, najčešće bazirano na iskustvu i posmatranju) i *Zigler-Nikolsov metod* [62] (više analitički pristup koji ima nekoliko varijacija).

Jedan od načina ručnog podešavanja je da se postavi $K_I = 0$ i $K_D = 0$, a da se K_p polako povećava dok se sistem ne dovede na granicu stabilnosti, tj. do pojave neprigušenih oscilacija u odzivu. Nakon toga se K_p smanji tako da u odzivu amplituda oscilacija smanji na četvrtinu (eng. *quarter amplitude decay*). Neko empirijsko pravilo je da se za te potrebe K_p smanji na pola svoje vrijednosti za koju je sistem na granici stabilnosti, i odatle vrši fino podešavanje. Nakon toga se K_I i K_D ručno mijenjaju dok se ne postigne željeni odziv.

Dva pristupa u Zigler-Nikolsovom metodu su pristup sa zatvorenom petljom i pristup sa otvorenom petljom.

Pristup sa zatvorenom petljom razmatra odziv sistema na odskočni ulazni signal, pri čemu je PID kontroler već u petlji. Slično ručnom podešavanju, inicijalno se isključuju integralno i diferencijalno dejstvo, a proporcionalno dejstvo se povećava dok odziv ne postane periodičan. Pojačanje koje sistem dovodi do granice stabilnosti se naziva *krajnje pojačanje* (eng. *ultimate gain*) i označava sa K_U. Period neprigušenih oscilacija T_U se naziva *krajnji period* (eng. *ultimate period*). Nakon određivanja ovih vrijednosti, na osnovu formula se računaju vrijednosti pojačanja PID kontrolera (postoje formule i za P, odnosno PI i PD kontrolere). Sve formule su date u tabeli 2.1. Ovaj metod je razvijen da pruži najbolje uklanjanje smetnji, ali proizvodi veći preskok nego ručno podešavanje, pa se ne koristi tamo gdje su zahtjevi za preskok stroži.

Pristup sa otvorenom petljom koristi krivu reakcije koja se dobije kad se kontoler isključi, a na ulaz objekta upravljanja dovede odskočni signal. Ovo je uobičajeno korišten pristup primjenjivan u procesnom upravljanju. Izmjereni odziv predstavlja

Tabela 2.1 *Zigler-Nikolsov metod sa zatvorenom petljom*

Tip kontrolera	K_p	K_I	K_D
P	$0.50K_U$	-	-
PI	$0.45K_U$	$\frac{0.54K_U}{T_U}$	-
PD	$0.80K_U$	-	$0.10K_U T_U$
PID	$0.60K_U$	$\frac{1.20K_U}{T_U}$	$0.075K_U T_U$

krivu reakcije. Ako kriva reakcije ima oblik sličan odskočnom odzivu sistema prvog reda sa transportnim kašnjenjem ili ako je objekat upravljanja linearan i letargičan, može se doći do prihvatljivo dobrog PID kontolera na osnovu pristupa sa otvorenom petljom. Kriva reakcije je karakterisana transportnim kašnjenjem T_d i brzinom reakcije R (koeficijent pravca prave koja najbolje aproksimira krivu reakcije za vrijeme prelaznog režima). Nakon određivanja ovih vrijednosti, na osnovu formula se računaju vrijednosti pojačanja PID kontrolera (postoje formule i za P, odnosno PI i PD kontrolere). Sve formule su date u tabeli 2.2.

Tabela 2.2 *Zigler-Nikolsov metod sa otvorenom petljom*

Tip kontrolera	K_p	K_I	K_D
P	$\frac{1.0}{RT_d}$	-	-
PI	$\frac{0.9}{RT_d}$	$\frac{0.27}{RT_d^2}$	-
PID	$\frac{1.2}{RT_d}$	$\frac{0.60}{RT_d^2}$	$\frac{0.6}{R}$

Nijedan od tri opisana metoda neće dovesti do željenih performansi za baš sve situacije, ali svakako predstavljaju dobre početne pretpostavke za vrijednosti pojačanja PID regulatora koje se daljim iterativnim postupkom mogu dovesti do vrijednosti koje će omogućiti ispunjavanje traženih zahtjeva.

2.4 Nelinearni sistemi

Iako je, kako je istaknuto u prethodnom izlaganju, često moguće, a i vrlo korisno, nelinearne sisteme linearizovati u okolini neke radne tačke, ponekad to nije dovoljno [63, 64]. Postoje dva glavna ograničenja linearizacije. Kao prvo, linearizacija je aproksimacija modela u okolini radne tačke, što znači da se na osnovu takvog modela može predvidjeti samo lokalno ponašanje sistema u okolini te tačke, ne i ponašanje u nekoj tački udaljenoj od te tačke, a naročito ne globalno ponašanje na cijelom prostoru stanja. Drugo, dinamika nelinearnih sistema je mnogo bogatija od dinamike linearnih sistema. Postoje tzv. *isključivo nelinearni fenomeni* koji se dešavaju samo u prisustvu nelinearnosti, te se nikako ne mogu opisati ili predvidjeti pomoću linearnih modela. Neki primjeri takvih fenomena su:

- *Konačno vrijeme bježanja.* Stanje nestabilnog linearnog sistema teži u beskonačnost kako vrijeme teži u beskonačnost, za razliku od nelinarnih sistema kod kojih stanje može da teži u beskonačnost za konačno vrijeme.
- *Više izolovanih ravnotežnih tačaka.* Linearan sistem može da ima samo jednu izolovanu ravnotežnu tačku, pa samim tim može da ima jednu radnu tačku ustaljenog stanja kojoj konvergiraju stanja sistema bez obzira na početno stanje. Nelinearan sistem može da ima više ravnotežnih tačaka, pa u zavisnosti od početnog stanja, stanja sistema mogu da konvergiraju u svaku od tih ravnotežnih tačaka.
- *Ograničeni ciklusi.* Da bi linearan vremenski invarijantan sistem oscilovao mora da ima par konjugovano-kompleksnih polova na imaginarnoj osi, što je nerobustan uslov koji se teško održava u prisustvu smetnji. Čak i kad se postigne održavanje, amplituda oscilacija zavisi od početnog stanja. Postoje nelinearni sistemi koji osciluju sa fiksnom amplitudom i fiksnom frekvencijom, bez obzira na počeno stanje. Ova vrsta oscilacija je poznata kao ograničeni ciklus.
- *Podharmonijske, harmonijske i skoro-periodične oscilacije.* Stabilan linearan sistem na čiji ulaz se dovede periodičan signal na izlazu daje periodičan signal iste frekvencije kao kod ulaznog. Nelinearni sistemi sa periodičnom eksitacijom daju izlaze čija frekvencija je dio ili umnožak frekvencije ulaznog signala, a moguće je da se dobije i skoro-periodičan signal (npr. suma signala sa frekvencijama koje nisu umnožak iste).
- *Haos.* Nelinearan sistem može imati komplikovanije ustaljeno stanje koje nije ni ravnotežno, ni periodično, ni skoro-periodično. Takvo stanje se uobičajeno naziva haos i manifestuje se u haotičnim kretanjima sa karakteristikama neodređenosti (slučajnosti), uprkos determinističkoj prirodi sistema.
- *Više različitih režima ponašanja.* Nije neuobičajeno da jedan nelinearan sistem pokazuje više različitih režima ponašanja, a sve u zavisnosti od početnog stanja ili promjene ulaza.

S obzirom na potrebu analize nelinearnih sistema, neophodan je i matematički alat za modelovanje takvog sistema. Najčešća varijanta tog modela je proširenje (2.9) za nelinearne sisteme.

$$\dot{\boldsymbol{x}} = f\left(t, \boldsymbol{x}, \boldsymbol{u}\right), \tag{2.31}$$

gdje je, kao i ranije, $\boldsymbol{x}$ vektor promjenljivih stanja, $\dot{\boldsymbol{x}}$ vektor prvih izvoda promjenljivih stanja, a $\boldsymbol{u}$ vektor ulaznih signala sistema. Slično je i

$$\boldsymbol{y} = h\left(t, \boldsymbol{x}, \boldsymbol{u}\right), \tag{2.32}$$

gdje je $\boldsymbol{y}$ vektor izlaznih signala sistema.

Često se sreće i varijanta

$$\dot{\boldsymbol{x}} = f\left(t, \boldsymbol{x}\right), \tag{2.33}$$

gdje se ne navodi eksplicitno ulaz sistema, što ne znači da nije prisutan. Moguće je da je ulaz funkcija nekog od stanja ili jedno od stanja sistema.

U slučaju kad f i h ne zavise od eksplicitno od neke nezavisne promjenljive, riječ je o *autonomnim* sistemima. Ukoliko je ta nezavisna promjenljiva vrijeme t, tj. zavisnost izvoda stanja i izlaza sistema od stanja i ulaza sistema se ne mijenja tokom vremena, riječ je o *vremenski invarijantnim* sistemima. U suprotnom, riječ je o *neautonomnim*, odnosno *vremenski promjenljivim sistemima.*

Važan koncept kod jednačina stanja je koncept ravnotežnih tačaka. Tačka $\boldsymbol{x} = \boldsymbol{x}^*$ je ravnotežna tačka sistema opisanog sa (2.33) ako ima osobinu da kad je početno stanje sistema upravo u toj tački, stanje ostane u toj tački u svim budućim trenucima. Kod vremenski invarijantnih sistema ravnotežne tačke su realna rješenja jednačine (sistema jednačina)

$$f(\boldsymbol{x}) = 0. \tag{2.34}$$

Ravnotežna tačka može biti izolovana, tj. u njenoj okolini nema drugih ravnotežnih tačaka, a moguće je i da ravnotežne tačke čine kontinualnu oblast.

2.4.1 Stabilnost nelinearnih sistema

Kod nelinearnih sistema postoji više vrsta stabilnosti, ali se najčešće proučava stabilnost ravnotežnih tačaka. Temelje ove teorije postavio je ruski matematičar i inženjer Ljapunov, pa ta teorija i nosi njegovo ime [65, 66, 67, 68].

Neka je dat autonoman sistem

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}), \tag{2.35}$$

gdje je $f : D \to \mathbb{R}^n$ lokalno Lipšicovo preslikavanje iz domena $D \subset \mathbb{R}^n$ na $\mathbb{R}^n$. Neka je $\boldsymbol{x}^*$ ravnotežna tačka sistema (2.35), odnosno neka je $f(\boldsymbol{x}^*) = 0$. Bez gubitka opštosti može se uzeti da je $\boldsymbol{x}^* = 0$, jer je moguće uvesti smjenu pomijeranjem osnovne promjenljive $\boldsymbol{x}$ za $\boldsymbol{x}^*$, odnosno $y = \boldsymbol{x} - \boldsymbol{x}^*$.

Ravnotežna tačka $\boldsymbol{x}$ je stabilna ako za svako $\varepsilon > 0$ postoji $\delta = \delta(\varepsilon) > 0$ takvo da važi

$$\|\boldsymbol{x}(0)\| < \delta \Rightarrow \|\boldsymbol{x}(t)\| < \varepsilon, \forall t \geq 0. \tag{2.36}$$

Ravnotežna tačka $\boldsymbol{x}$ je asimptotski stabilna ako se može izabrati δ takvo da

$$\|\boldsymbol{x}(0)\| < \delta \Rightarrow \lim_{t\to\infty} \boldsymbol{x}(t) = 0. \tag{2.37}$$

Ravnotežna tačka $\boldsymbol{x}$ je nestabilna ako nije stabilna.

Teorema 2.1. ***(Teorema Ljapunova)*** *Neka je $\boldsymbol{x} = 0$ ravnotežno stanje sistema (2.35) i neka je $D \subset \mathbb{R}^n$ domen koji sadrži $\boldsymbol{x}$. Neka je $V : D \to \mathbb{R}$ neprekidna diferencijabilna funkcija takva da je*

$$V(0) = 0 \;\; i \;\; V(\boldsymbol{x}) > 0, \forall \boldsymbol{x} \in D \backslash \{0\}, \tag{2.38}$$

$$\dot{V}(\boldsymbol{x}) \leq 0, \forall \boldsymbol{x} \in D. \tag{2.39}$$

Tada je $\boldsymbol{x}$ stabilna ravnotežna tačka. Ako važi

$$\dot{V}(\boldsymbol{x}) < 0, \forall \boldsymbol{x} \in D \backslash \{0\}, \tag{2.40}$$

ravnotežna tačka $\boldsymbol{x}$ je asimptotski stabilna.

Dokaz. Za neko $\varepsilon > 0$ se bira $r \in (0, \varepsilon]$ tako da

$$B_r = \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}\| \leq r\} \subset D.$$

Neka je $\alpha = \min_{\|\boldsymbol{x}\|=r} V(\boldsymbol{x})$. Tada je, na osnovu (2.38), $\alpha > 0$. Za neko $\beta \in (0, \alpha)$ definiše se

$$\Omega_\beta = \{\boldsymbol{x} \in B_r \mid V(\boldsymbol{x}) \leq \beta\}.$$

Tada je Ω_β u unutrašnjosti B_r što se može pokazati kontradikcijom. Skup Ω_β ima osobinu da bilo koja putanja koja počinje u Ω_β u trenutku $t = 0$ ostaje u Ω_β za svako $t \geq 0$. Ovo slijedi iz (2.39). S obzirom da je skup Ω_β kompaktan, slijedi da sistem (2.35) ima jedinstveno rješenje za svako $t \geq 0$ kad god $\boldsymbol{x}(0) \in \Omega_\beta$. Kako je $V(\boldsymbol{x})$ neprekidna i $V(0) = 0$, postoji $\delta > 0$ takvo da

$$\|\boldsymbol{x}\| < \delta \Rightarrow V(\boldsymbol{x}) < \beta.$$

Tada je

$$B_\delta \subset \Omega_\beta \subset B_r$$

i

$$\boldsymbol{x}(0) \in B_\delta \Rightarrow \boldsymbol{x}(0) \in \Omega_\beta \Rightarrow \boldsymbol{x}(t) \in \Omega_\beta \Rightarrow \boldsymbol{x}(t) \in B_r.$$

Na osnovu toga je

$$\|\boldsymbol{x}(0)\| < \delta \Rightarrow \|\boldsymbol{x}(t)\| < r \leq \varepsilon, \forall t \geq 0$$

što pokazuje da je ravnotežna tačka stabilna.

Neka važi i (2.40). Za dokazivanje asimptotske stabilnosti treba pokazati da $\boldsymbol{x}(t) \to 0$ kad $t \to 0$, odnosno da za svako $a > 0$ postoji $T > 0$ takvo da $\|\boldsymbol{x}(t)\| < a$ za svako $t > T$. Ponavljanjem prethodnih argumenata dolazi se do zaključka da za svako $a > 0$ postoji $b > 0$ takvo da je $\Omega_b \subset B_a$. Dakle, dovoljno je pokazati da $V(\boldsymbol{x}(t)) \to 0$ kad $t \to \infty$. Kako je $V(\boldsymbol{x}(t))$ monotono opadajuća i ograničena odozdo nulom, važi

$$V(\boldsymbol{x}(t)) \to c \geq 0 \text{ za } t \to \infty.$$

Za dokazivanje da je $c = 0$ koristi se kontradikcija. Zbog neprekidnosti $V(\boldsymbol{x})$ postoji $d > 0$ takvo da je $B_d \subset \Omega_c$. Ograničenje $V(\boldsymbol{x}(t)) \to c > 0$ implicira da putanja $\boldsymbol{x}(t)$ leži izvan B_d za svako $t \geq 0$. Neka je $\gamma = -\max_{d \leq \|V(\boldsymbol{x})\| \leq r} \dot{V}(\boldsymbol{x})$, što sigurno postoji jer neprekidna funkcija $\dot{V}(\boldsymbol{x})$ ima maksimum na kompaktnom skupu $\{d \leq \|V(\boldsymbol{x})\| \leq r\}$. Iz (2.40) slijedi $\gamma > 0$, a odatle

$$V(\boldsymbol{x}(t)) = V(\boldsymbol{x}(0)) + \int_0^t \dot{V}(\boldsymbol{x}(\tau))d\tau \leq V(\boldsymbol{x}(0)) - \gamma\tau.$$

Kako desna strana nejednakosti sigurno mora da postane negativna, javlja se kontradikcija se pretpostavkom da je $c > 0$. ∎

Funkcija $V(\boldsymbol{x})$ koja zadovoljava (2.38) i (2.39) se naziva Ljapunovljeva funkcija. Površ $V(\boldsymbol{x}) = c$, za neko $c > 0$, se naziva Ljapunovljeva površ. Korištenjem Ljapunovljevih funkcija teorema postaje intuitivno jasna. Iz uslova $\dot{V} \leq 0$ slijedi da kad putanja uđe u jednu Ljapunovljevu površ, nikad ne izađe iz nje. Ako je još i $\dot{V} < 0$, onda putanja ulazi u površi sa sve manjim c, što teži ishodištu kako se vrijeme povećava.

Funkcija $V(\boldsymbol{x})$ koja zadovoljava uslove $V(0) = 0$ i $V(\boldsymbol{x}) > 0$, za $\boldsymbol{x} \neq 0$, se naziva pozitivno određenom, a ako zadovoljava slabiji uslov $V(\boldsymbol{x}) \geq 0$, za $\boldsymbol{x} \neq 0$, onda se naziva pozitivno poluodređenom. Funkcija $V(\boldsymbol{x})$ je negativno određena, odnosno negativno poluodređena, ako je funkcija $-V(\boldsymbol{x})$ pozitivno određena, odnosno pozitivno poluodređena, respektivno. Ukoliko funkcija nije nijedno od navedenog, onda je neodređena. U skladu sa tim, teorema Ljapunova se može parafrazirati tako da glasi: *Ishodište je stabilno ako postoji neprekidna diferencijabilna pozitivno određena funkcija $V(\boldsymbol{x})$ takva da je $\dot{V}(\boldsymbol{x})$ negativno poluodređena, i asimptotski je stabilno ako je $\dot{V}(\boldsymbol{x})$ negativno određena.*

Ne postoji sistematičan metod za pronalaženje Ljapunovljevih funkcija. Ponekad je prirodan kandidat funkcija energije, a ponekad se svodi na metod pokušaja i promašaja. Treba uzeti u obzir i sljedeće - uslovi teoreme su dovoljni, ne i neophodni. Ako neka kandidatska Ljapunovljeva funkcija ne ispunjava uslove stabilnosti ili asimptotske stabilnosti, ne znači da ishodište nije stabilno ili asimptotski stabilno, već samo da pokušaj ne vodi ka odgovoru, te da je potrebno dalje ispitivanje.

Neka je dat neautonoman sistem

$$\dot{\boldsymbol{x}} = f\left(t, \boldsymbol{x}\right), \tag{2.41}$$

gdje je $f : [0, \infty) \times D \to \mathbb{R}^n$ dio-po-dio neprekidno u vremenu, te lokalno Lipšicovo preslikavanje iz domena $[0, \infty) \times D$ i $D \subset \mathbb{R}^n$ na $\mathbb{R}^n$. Ishodište je ravnotežna tačka sistema (2.41) u trenutku $t = 0$ ako je

$$f\left(t, 0\right) = 0, \forall t \geq 0. \tag{2.42}$$

Slično autonomnim sistemima, ravnotežna tačka u ishodištu može biti transformacija ishodišta u proizvoljnu tačku.

Principi razmatranja stabilnosti i asimptotske stabilnosti ravnotežnih tačaka neautonomnih sistema su praktično isti kao i oni kod autonomnih sistema. Razlika je u tome što rješenje autonomnog sistema zavisi samo od $(t - t_0)$, dok kod neautonomnog može da zavisi i od t i od t_0.

Ravnotežna tačka $\boldsymbol{x} = 0$ sistema (2.41) je:

- stabilna ako za svako $\varepsilon > 0$ postoji $\delta = \delta(\varepsilon, t_0) > 0$ takvo da važi

$$\|\boldsymbol{x}(t_0)\| < \delta \Rightarrow \|\boldsymbol{x}(t)\| < \varepsilon, \forall t \geq t_0 \geq 0, \tag{2.43}$$

- uniformno stabilna ako za svako $\varepsilon > 0$ postoji $\delta = \delta(\varepsilon) > 0$, nezavisno od t_0 takvo da važi (2.43),
- nestabilna ako nije stabilna,
- asimptotski stabilna ako je stabilna i ako postoji konstanta $c = c(t_0)$ takva da $\boldsymbol{x}(t) \to 0$ kad $t \to \infty$ za sve $\|\boldsymbol{x}(t_0)\| < c$,
- uniformno asimptotski stabilna ako je uniformno stabilna i ako postoji konstanta c nezavisna od t_0 takva da $\boldsymbol{x}(t) \to 0$ kad $t \to \infty$ za sve $\|\boldsymbol{x}(t_0)\| < c$, uniformno u t_0, odnosno da za svako $\eta > 0$ postoji $T = T(\eta) > 0$ takvo da

$$\|\boldsymbol{x}(t)\| < \eta, \forall t \geq t_0 + T(\eta), \forall \|\boldsymbol{x}(t_0)\| < c, \tag{2.44}$$

- globalno uniformno asimptotski stabilna ako je uniformno stabilna, ako se može izabrati $\delta(\varepsilon)$ tako da važi $\lim_{\varepsilon\to\infty}\delta(\varepsilon)=\infty$ i ako za svaki par pozitivnih brojeva η i c postoji $T=T(\eta,c)>0$ takvo da

$$\|\boldsymbol{x}(t)\|<\eta, \forall t\geq t_0+T(\eta,c), \forall\|\boldsymbol{x}(t_0)\|<c. \tag{2.45}$$

Poseban slučaj uniformne asimptotske stabilnosti je eksponencijalna stabilnost. Ravnotežna tačka $\boldsymbol{x}=0$ sistema (2.41) je eksponencijalno stabilna ako postoje pozitivne konstante c, k i λ takve da važi

$$\|\boldsymbol{x}(t)\|<k\|\boldsymbol{x}(t_0)\|e^{-\lambda(t-t_0)}, \forall\|\boldsymbol{x}(t_0)\|<c \tag{2.46}$$

i globalno eksponencijalno stabilna ako je (2.46) ispunjeno za bilo koje početno stanje $\boldsymbol{x}(t_0)$.

Teorema Ljapunova se može proširiti i na neautonomne sisteme.

Teorema 2.2. *Neka je $\boldsymbol{x}=0$ ravnotežno stanje sistema* (2.41) *i neka je $D\subset\mathbb{R}^n$ domen koji sadrži $\boldsymbol{x}$. Neka je $V:[0,\infty)\times D\to\mathbb{R}$ neprekidna diferencijabilna funkcija takva da je*

$$W_1(\boldsymbol{x})\leq V(t,\boldsymbol{x})\leq W_2(\boldsymbol{x}), \tag{2.47}$$

$$\frac{\partial V}{\partial t}+\frac{\partial V}{\partial \boldsymbol{x}}f(t,\boldsymbol{x})\leq 0 \tag{2.48}$$

za sve $t\geq 0$ i $\boldsymbol{x}\in D$, gdje su $W_1(\boldsymbol{x})$ i $W_2(\boldsymbol{x})$ neprekidne pozitivno određene funkcije na D. Tada je $\boldsymbol{x}=0$ uniformno stabilna ravnotežna tačka.

2.4.2 Upravljanje nelinearnim sistemima

Kompleksnost upravljanja nelinearnim sistemima predstavlja izazov naučnicima za razvijanje sistematskih procedura dizajna kontrolera kojim će se ispuniti upravljački ciljevi i specifikacije zadataka. Jasno je da ne treba očekivati da se jedna konkretna procedura može primijeniti na sve nelinearne sisteme, niti da se kompletan dizajn kontrolera može zasnivati na jednom alatu. To znači da inženjeri koji realizuju upravljanje treba da odaberu pogodan alat za problem na kojem rade. Postoji mnogo takvih alata, a u nastavku će biti predstavljeni samo neki: upravljanje u kliznom režimu (eng. *sliding mode control*), Ljapunovljev redizajn i koračanje unazad (eng. *backstepping*).

Upravljanje u kliznom režimu je nelinearan metod upravljanja kojim se mijenja dinamika nelinearnog sistema primjenom upravljačkog signala sa prekidima čime se sistem primorava na *klizanje* po granicama normalnog ponašanja sistema. Zakon upravljanja se bazira na povratnoj sprezi po stanju, nije neprekidna funkcija, tj. mijenja se iz jedne u drugu neprekidnu strukturu u zavisnosti od trenutnog položaja sistema u prostoru stanja. Te višestruke upravljačke strukture se realizuju tako da putanja uvijek vodi prema susjednom regionu sa drugom strukturom, pa zato konačna putanja ne leži u samo jednoj upravljačkoj strukturi, već kliže po granicama više upravljačkih struktura. Geometrijsko mjesto tih granica se naziva klizna (hiper)površ. Upravljanje u kliznom režimu se mora primjenjivati sa više opreza nego drugi oblici nelinearnog upravljanja koji imaju umjerenije upravljačko djelovanje. Naime, zbog kašnjenja aktuatora i drugih nesavršenosti, ovakvo upravljanje može dovesti do podrhtavanja, gubitka energije, kao i oštećenja objekta upravljanja.

Ljapunovljev redizajn je tehnika dizajna povratne sprege po stanju za stabilizaciju sistema zasnovan na poznavanju Ljapunovljeve funkcije V. Za sistem

$$\dot{\boldsymbol{x}} = f(t, \boldsymbol{x}) + G(t, \boldsymbol{x})[\boldsymbol{u} + \delta(t, \boldsymbol{x}, \boldsymbol{u})], \tag{2.49}$$

gdje je $\boldsymbol{x} \in \mathbb{R}^n$ vektor stanja, a $\boldsymbol{u} \in \mathbb{R}^p$ vektor ulaza. Funkcije f, G i δ su definisane za $(t, \boldsymbol{x}, \boldsymbol{u}) \in [0, \infty] \times D \times \mathbb{R}^p$, gdje je $D \subset \mathbb{R}^n$ domen koji sadrži ishodište. Nominalan model sistema je

$$\dot{\boldsymbol{x}} = f(t, \boldsymbol{x}) + G(t, \boldsymbol{x})\boldsymbol{u}, \tag{2.50}$$

dok je δ nepoznata funkcija koja uključuje razne neodređenosti, poput greške ili pojednostavljenja modela i sl. Na osnovu poznavanja Ljapunovljeve funkcije V je moguće dizajnirati upravljački zakon

$$\boldsymbol{u} = \psi(t, \boldsymbol{x}) + v, \tag{2.51}$$

kojim se stabilizuje dati sistem u prisustvu neodređenosti, a tehnika određivanja v se naziva Ljapunovljev redizajn.

Koračanje unazad je tehnika dizajna stabilizujućeg kontrolera za specijalne klase nelinearnih sistema koji se sastoje od podsistema koji se razvijaju od jednostavnih podsistema (ne mogu se dalje redukovati) koji mogu biti stabilizovani nekom drugom metodom. Zbog ove rekurzivne strukture, inženjer može započeti proces dizajna od poznatog stabilnog stanja i praviti korake unazad dizajnom novih kontrolera koji progresivno stabilizuju svaki vanjski podsistem. Proces se završava kad se postigne konačno vanjsko upravljanje. Matematički model pomenute klase sistema se dat sa

$$\begin{aligned}
\dot{\boldsymbol{x}} \quad &= f_x(\boldsymbol{x}) + g_x(\boldsymbol{x}) z_1 \\
\dot{z_1} \quad &= f_1(\boldsymbol{x}, z_1) + g_1(\boldsymbol{x}, z_1) z_2 \\
\dot{z_2} \quad &= f_2(\boldsymbol{x}, z_1, z_2) + g_2(\boldsymbol{x}, z_1, z_2) z_3 \\
&\vdots \\
\dot{z_i} \quad &= f_i(\boldsymbol{x}, z_1, z_2, \ldots, z_{i-1}, z_i) + g_i(\boldsymbol{x}, z_1, z_2, \ldots, z_{i-1}, z_i) z_{i+1} \\
&\vdots \\
\dot{z_{k-1}} \quad &= f_{k-1}(\boldsymbol{x}, z_1, z_2, \ldots, z_{k-1}) + g_{k-1}(\boldsymbol{x}, z_1, z_2, \ldots, z_{k-1}) z_k \\
\dot{z_k} \quad &= f_k(\boldsymbol{x}, z_1, z_2, \ldots, z_k) + g_k(\boldsymbol{x}, z_1, z_2, \ldots, z_k) u
\end{aligned} \tag{2.52}$$

gdje je

- $\boldsymbol{x} \in \mathbb{R}^n$, $n \geq 1$;
- $z_1, z_2, \ldots, z_k$ su skalarne funkcije;
- u je ulazni skalar;
- $f_x, f_1, f_2, \ldots, f_k$ iščezavaju u ishodištu, tj. $f_i(0, 0, \ldots, 0) = 0$;
- $g_x, g_1, g_2, \ldots, g_k$ ne prolaze kroz nulu na domenu od interesa.

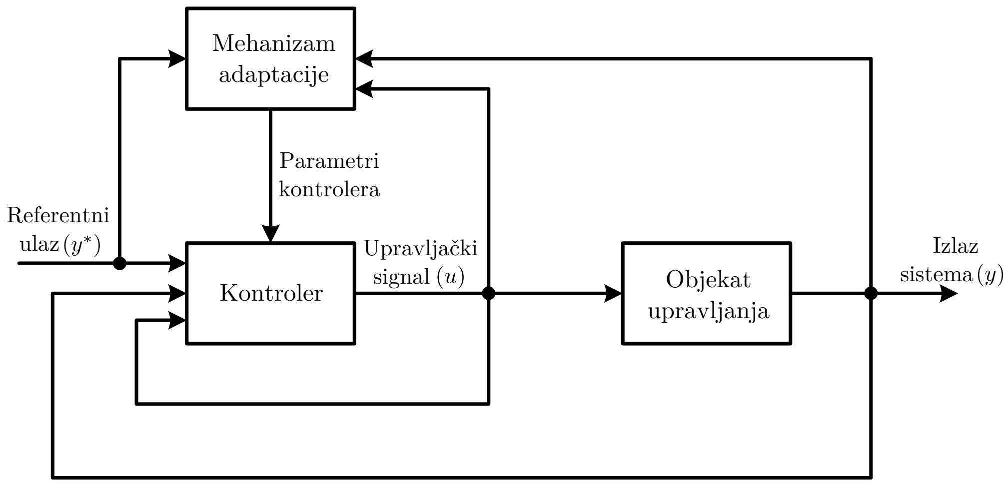
Pretpostavka je da je podsistem

$$\dot{\boldsymbol{x}} = f_x(\boldsymbol{x}) + g_x(\boldsymbol{x}) u_x(\boldsymbol{x}) \tag{2.53}$$

stabilizovan poznatim upravljačkim signalom $u_x(\boldsymbol{x})$ i da je poznata Ljapunovljeva funkcija V_x za taj stabilan podsistem. Proces koračanja unazad se svodi na stabilizaciju podsistema $\boldsymbol{x}$ pomoću z_1, pa se nakon toga određuje način kako da pomoću z_2 upravljati sa z_1 tako da se postigne ta stabilizacija i tako do z_k.

2.5 Adaptivno upravljanje

U svakodnevnom izražavanju, adaptacija je proces promjene ponašanja zarad prilagođenja novim okolnostima. Ne postoji usaglašena definicija, ali intuitivno, adaptivni kontroler [69, 70, 71, 72] je kontroler sa promjenljivim parametrima i sa mehanizmom za promjenu tih parametara. Sistem sa adaptivnim upravljanjem se može posmatrati kao sistem sa dvije petlje. Jedna petlja je standardna povratna petlja kojom se izlaz procesa dovodi na ulaz kontrolera. Druga petlja je petlja za prilagođavanje parametara kontrolera. Blok-dijagram sistema sa adaptivnim upravljanjem je prikazan na Slici 2.12. Petlja za prilagođavanje je uobičajeno sporija od standardne povratne petlje.



Slika 2.12 *Blok-dijagram sistema sa adaptivnim upravljanjem*

U nastavku će biti opisana 3 tipa adaptivnih sistema: sistemi sa raspoređivanjem po pojačanju (eng. *Gain Scheduling - GS*), sistemi sa referentnim modelom [73, 74] (eng. *Model-Reference Adaptive Systems - MRAS*) i sistemi sa samopodešavajućim regulatorima [75, 76] (eng. *Self-Tuning Regulators*).

2.5.1 Sistemi sa raspoređivanjem po pojačanju

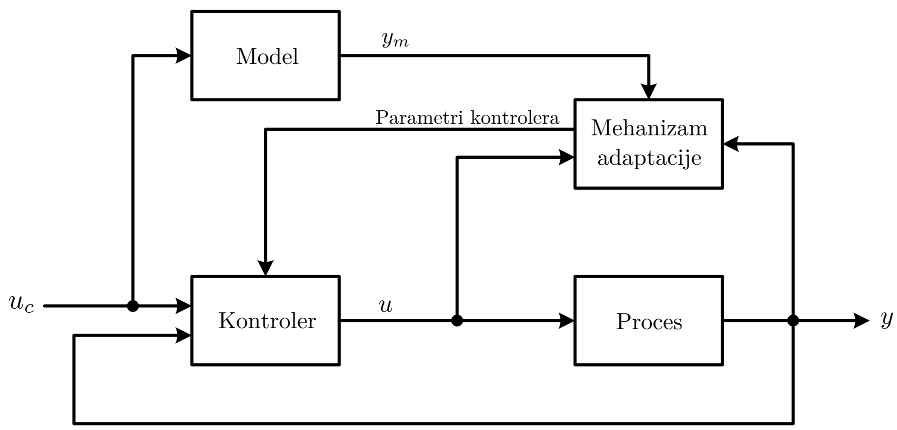
U mnogim slučajevima moguće je utvrditi mjerljive vrijednosti koje imaju dobru korelaciju sa promjenama u dinamici procesa. Te vrijednosti se mogu koristiti za promjenu parametara kontrolera. Ovaj pristup se naziva raspoređivanje po pojačanju, jer je ta šema prvobitno korištena za mjerenje pojačanja, nakon čega bi se mijenjali, odnosno raspoređivali, parametri kontrolera da kompenzuju promjene u pojačanju procesa. Sam proces promjene parametara se zasniva na mapiranju parametara procesa na parametre kontrolera, a može se implementirati kao funkcija ili kao uporedna tabela (eng. *lookup table*).

Koncept raspoređivanja po pojačanju vodi porijeklo iz razvoja sistema za upravljanje letom, pri čemu su brzina i visina aviona korištene kao vrijednosti za raspoređivanje. U upravljanju industrijskim procesima, često se stopa produktivnosti može koristiti kao veličina za raspoređivanje.

Istorijski, bilo je rasprava o tome da li je ovaj način upravljanja adaptivan ili ne, ali u skladu sa neformalnom definicijom datom ranije, definitivno se može smatrati adaptivnim.

2.5.2 Sistemi sa referentnim modelom

Adaptivno upravljanje na bazi referentnog modela je prvobitno predloženo kao rješenje problema u kojem su sprecifikacije performansi date kroz referentni model. Model daje informaciju kakav bi u idealnom slučaju trebao da bude izlaz sistema na zadati ulaz. Blok-dijagram sistema je dat na Slici 2.13. Prva petlja, kako je ranije opisano je standardna povratna petlja, dok druga petlja služi za podešavanje parametara kontrolera tako da greška, odnosno razlika između stvarnog izlaza procesa (y) i izlaza modela (y_m) bude što manja.



Slika 2.13 *Blok-dijagram sistema sa adaptivnim upravljanjem na bazi referentnog modela - MRAS*

Kao i prethodni pristup, MRAS je nastao u razvoju upravljanja letom, a referentni model je opisivao željeni odziv aviona na pokrete upravljača.

Ključni problem kod MRAS je određivanje mehanizma adaptacije tako da se dobije stabilan sistem koji smanjuje grešku na nulu. Taj problem nije trivijalan. U originalnoj varijanti MRAS korišten je sljedeći princip, poznat kao MIT pravilo[1]:

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta}, \tag{2.54}$$

gdje je $e = y - y_m$ greška u odnosu na model, a θ parametar kontrolera. Vrijednost $\frac{\partial e}{\partial \theta}$ predstavlja osjetljivost greške na promjenu parametra θ. Parametar γ određuje

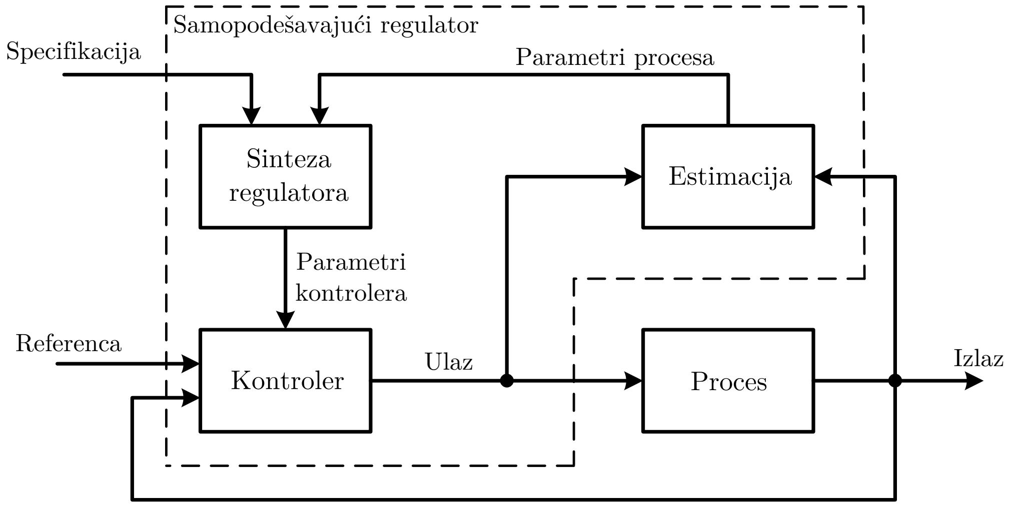
[1]Originalno MRAS pristup je razvijen na univerzitetu MIT (Massachusetts Institute of Technology) oko 1960. godine.

stopu prilagođenja. U praksi je neophodno vršiti aproksimacije za određivanje osjetljivosti. MIT pravilo se može smatrati gradijentnim pristupom minimizaciji kvadrata greške.

2.5.3 Sistemi sa samopodešavajućim regulatorom

Prethodno opisane adaptivne šeme spadaju u *direktne* metode, jer mehanizam adaptacije daje pravila za izmjenu parametara kontrolera. Drugačija šema se dobija kad se prilagođavaju procjene parametara procesa, a parametri kontrolera dobijaju kao rješenja problema sinteze regulatora na osnovu tih procjena. Blok-dijagram takvog sistema je dat na Slici 2.14. Kao i ranije, u sistemu postoje dvije petlje, standardna povratna petlja i petlja za podešavanje parametara regulatora, koja se ovdje sastoji od rekurzivnog estimatora parametara procesa i sistema za sintezu kontrolera. Ovako konstruisan kontroler se naziva samopodešavajući regulator.

S obzirom da se za sintezu regulatora koriste procjene parametara procesa, u zavisnosti od stepena sigurnosti tih procjena (nekad je moguće mjerenjima utvrditi tačnost procjena) moguće su različite specifikacije načina sinteze regulatora.



Slika 2.14 *Blok-dijagram sistema sa samopodešavajućim regulatorom - STR*

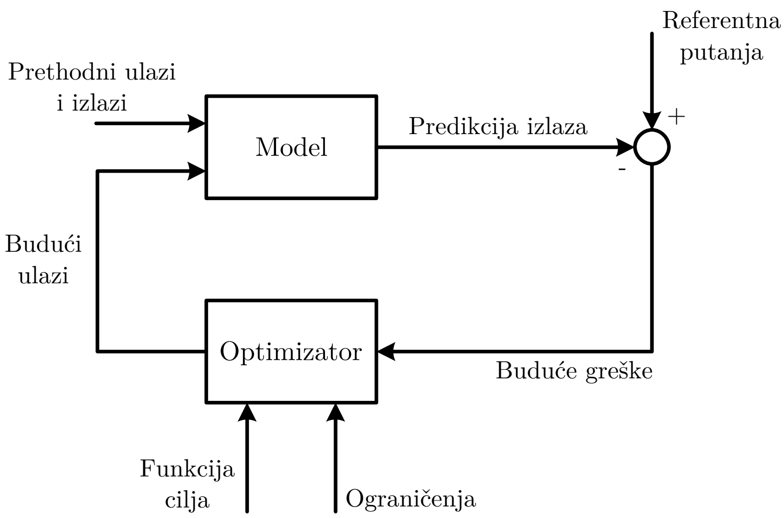
2.6 Prediktivno upravljanje

Prediktivno upravljanje na bazi modela (eng. *Model Predictive Control - MPC*) [77, 78, 79, 80, 81, 82, 83] je nastalo u sedamdesetim godinama XX vijeka i otad se znatno razvilo. Ne radi se o jedinstvenoj strategiji upravljanja, već o širokom opsegu metoda, a ideje koje se u manjem ili većem stepenu pojavljuju u svim familijama prediktivnog upravljanja su:

- eksplicitno korištenje modela procesa za predviđanje izlaza procesa u budućnosti (horizont),
- računanje upravljačke sekvence minimizacijom neke funkcije cilja,

- strategija povlačenja, po kojoj se u svakom odmjerku vremena horizont pomijera u budućnost primjenom prvog upravljačkog signala iz prethodno izračunate sekvence.

Osnovna struktura MPC sistema je data na Slici 2.15.



Slika 2.15 *Osnovna struktura MPC*

Različite varijante MPC algoritama se razlikuju u modelu koji koriste za opisivanje procesa, te šumova i funkcija cilja koje treba minimizovati.

Prednosti MPC pristupa u odnosu na druge metode su brojne, a među njima se ističu:

- može se koristiti za upravljanje mnogim vrstama procesa, od onih sa jednostavnom dinamikom, do onih sa kompleksnom, poput sistema sa velikim kašnjenjima, te neminimalno faznim i nestabilnim sistemima,
- lako se prilagođava multivarijabilnim sistemima,
- interno kompenzuje *mrtva* vremena,
- uvodi direktno upravljanje na prirodan način da kompenzuje mjerljive poremećaje,
- rezultujući kontroler je linearan i relativno jednostavan za implementaciju,
- jednostavno se proširuje da uključi ograničenja,
- vrlo je korisno znati buduće vrijednosti izlaza procesa,
- radi se o potpuno otvorenoj metodologiji zasnovanoj na bazičnim principima što omogućava proširenja i poboljšanja u budućnosti.

Očekivano, postoje i neke mane. Jedna od njih je to što je, uprkos činjenici da je rezultujući zakon upravljanja jednostavan i lak za implementaciju, njegovo izvođenje kompleksnije od onog za standardne PID regulatore. Najčešće je potrebno to izvođenje izvršavati u svakom odmjerku vremena. Kad se uključe i ograničenja, ovaj proračun postaje još kompleksniji. Iako su današnji računari na visokom nivou računarske moći, pa se ovaj problem može smatrati manjim, treba uzeti u obzir da su u industrijskim procesima u upotrebi ipak nešto manje razvijeni računari, a i to da isti moraju da obavljaju i druge važne aktivnosti (komunikacija, skladištenje podataka, interakcija sa operaterom i sl.). Druga, znatno veća mana, je neophodnost postojanja odgovarajućeg modela procesa. S obzirom da se razvoj kontrolera bazira na modelu, očigledno je da će netačnost modela znatno uticati na kvalitet kontrolera.

2.6.1 Modeli za predikciju

Model je kamen temeljac za MPC i zato je neophodno da bude najbolji moguć, što znači da bi trebalo da potpuno obuhvati dinamiku procesa, te da omogući izračunavanje predikcija, a bilo bi poželjno i da bude intuitivan i da omogući teoretsku analizu. Model može da bude razdvojen na dva dijela: model procesa i model smetnji. Model smetnji opisuje ponašanje koje nije obuhvaćeno modelom procesa: uticaj nemjerljivih ulaza, šuma i greške modela.

2.6.1.1 Model procesa

Pored ranije opisanih načina modelovanja (funkcija prenosa i model sistema u prostoru stanja), često korišteni načini modelovanja u MPC su bazirani na impulsnom, odnosno odskočnom odzivu sistema. Impulsni odziv sistema je odziv na jedinični impulsni signal (često nazivan Dirakov impuls), dok je odskočni odziv sistema odziv na jedinični odskočni signal (često nazivan Hevisajdov signal).

Ako je poznat impulsni odziv sistema, moguće je izračunati odziv sistema na proizvoljan ulaz $u(t)$ prema

$$y(t) = \sum_{i=1}^{\infty} g_i u(t-i), \tag{2.55}$$

gdje je g_i i-ti odmjerak impulsnog odziva sistema.

Ako je poznat odskočni odziv sistema, moguće je izračunati odziv sistema na proizvoljan ulaz $u(t)$ prema

$$y(t) = \sum_{i=1}^{\infty} h_i \Delta u(t-i), \tag{2.56}$$

gdje je h_i i-ti odmjerak odskočnog odziva sistema, a $\Delta u(t) = u(t) - u(t-1)$.

Postoji i veza između impulsnog i odskočnog odziva

$$g_i = h_i - h_{i-1}, \tag{2.57}$$

odnosno

$$h_i = \sum_{j=1}^{i} g_j. \tag{2.58}$$

Prednost u odnosu na druge vrste modela je to što nikakve prethodne informacije o procesu nisu potrebne, tako da je postupak identifikacije pojednostavljen, a istovremeno se kompleksna dinamika lako opisuje. Mane su to što su ovakvi modeli primjenjivi samo na stabilne sisteme bez integratora, te prilično zauzeće memorije, čak i kad se pamti samo N odmjeraka odziva, jer je taj broj uobičajeno oko 40, 50.

Predikcija za trenutak $t+k$ izračunata u trenutku t na osnovu impulsnog odziva je data sa

$$\hat{y}\left(t+k|\,t\right)=\sum_{i=1}^{N} g_i u\left(t+k-i|\,t\right), \tag{2.59}$$

a na osnovu odskočnog sa

$$\hat{y}\left(t+k|\,t\right)=\sum_{i=1}^{N} h_i \Delta u\left(t+k-i|\,t\right). \tag{2.60}$$

2.6.1.2 Model smetnje

Najčešće korišten model je upravljani autoregresivni sa integrisanim pomijeranjem prosjeka (eng. *Controlled Auto-Regressive and Integrated Moving Average - CARIMA*) u kojem se smetnje, odnosno razlike između izmjerenog izlaza i izlaza izračunatog na osnovu modela, računaju preko

$$n(t)=\frac{C\left(z^{-1}\right)e(t)}{D\left(z^{-1}\right)}, \tag{2.61}$$

gdje je $D\left(z^{-1}\right)$ polinom koji eksplicitno sadrži $\Delta = 1 - z^{-1}$, $e(t)$ je bijeli šum sa nultom srednjom vrijednošću, a polinom $C\left(z^{-1}\right)$ je uobičajeno jednak 1.

Ovakav model se smatra pogodnim za dvije vrste smetnji: kad se slučajne promjene dešavaju u slučajnim trenucima vremena, te Braunovo kretanje.

Korištenjem Diofantske jednačine

$$1=E_k\left(z^{-1}\right)D\left(z^{-1}\right)+z^{-k}F_k\left(z^{-1}\right) \tag{2.62}$$

dobija se

$$n(t)=E_k\left(z^{-1}\right)e(t)+z^{-k}\frac{F_k\left(z^{-1}\right)}{D\left(z^{-1}\right)}e(t), \tag{2.63}$$

odnosno

$$n(t+k)=E_k\left(z^{-1}\right)e(t+k)+F_k\left(z^{-1}\right)n(t), \tag{2.64}$$

pa je predikcija

$$\hat{n}\left(t+k|\,t\right)=F_k\left(z^{-1}\right)n(t). \tag{2.65}$$

2.6.2 Funkcija cilja

Uobičajeno je da se traži da budući odziv (y) prati određeni referentni signal (w) i da se istovremeno promjena upravljačkog signala (Δu) potrebna da se to postigne drži što manjom. Uopšteni izraz za takvu funkciju cilja je

$$\begin{aligned} J(N_1,N_2,N_u)= & \sum_{j=N_1}^{N_2}\delta(j)\left[\hat{y}\left(t+j|\,t\right)-w(t+j)\right]^2+ \\ & \sum_{j=1}^{N_u}\lambda(j)\left[\Delta u(t+j-1)\right]^2. \end{aligned} \tag{2.66}$$

U nekim varijantama MPC drugi sabirak koji se odnosi na promjene upravljačkog signala se izostavlja, dok se kod nekih ne razmatraju promjene, već direktno vrijednosti upravljačkog signala.

Parametri N_1 i N_2 predstavljaju minimalan i maksimalan horizont funkcije cilja i prilično su intuitivni jer označavaju granice vremenskih instanci u kojima je poželjno praćenje reference. Ako se postavi velika vrijednost N_1, onda se smatra da nije važno da li postoji greška u početnim trenucima. Ako postoji mrtvo vrijeme d, nema potrebe da N_1 bude manje od d, jer svakako ne postoji odziv za to vrijeme. Ako je proces neminimalno fazni, onda će ovaj parametar omogućiti eliminaciju prvih odmjeraka inverznog odziva.

Parametar N_u je upravljački horizont.

Koeficijenti $\delta(j)$ i $\lambda(j)$ su sekvence koje se odnose na buduće ponašanje. Uobičajeno su konstantne ili eksponencijalne sekvence. Neka je, npr. $\delta(j)$ dato sa

$$\delta(j) = \alpha^{N_2-j} \tag{2.67}$$

i ako α ima vrijednost između 0 i 1, onda se strožije kažnjavaju greške koje su dalje od vremenskog trenutka t, što rezultuje u glatkijim promjenama upravljanja, dok za $\alpha > 1$ veći značaj dobijaju greške u trenucima bližim trenutku t, što rezultuje u strožijem upravljanju.

U praksi svi procesi uključuju i određena ograničenja, poput minimalnog ili maksimalnog nivoa tečnosti u rezervoaru, ugla okrenutosti ventila itd. Ovo čini uvođenje ograničenja i u funkciju cilja. Neki algoritmi MPC uključuju ograničenja interno, dok ih neki uključuju naknadno. Uobičajena su ograničenja vrijednosti upravljačkog signala, kao i njegove promjene, te vrijednosti izlaznog signala. Uvođenje ograničenja čini minimizaciju funkcije cilja kompleksnijom, često toliko da se rješenje ne može dobiti eksplicitno, kao kod problema minimizacije bez ograničenja.

2.6.3 Izvođenje zakona upravljanja

Za određivanje upravljačkih vrijednosti $u\left(t+k|\, t\right)$ neophodno je minimizovati funkciju J iz (2.66). Da bi se ovo postiglo, računaju se procjene vrijednosti izlaza $\hat{y}\left(t+k|\, t\right)$ kao funkcije prethodnih vrijednosti ulaza i izlaza, te budućih vrijednosti upravljačkog signala dobijenih na osnovu modela. Ovaj zadatak je kvadratičan ako je model linearan i ako nisu uključena ograničenja, i u tom slučaju postoji analitičko rješenje. U suprotnom, koristi se neki iterativni optimizacioni metod. Koji god metod da se koristi, problem nije jednostavan jer je u pitanju $N_2 - N_1 + 1$ nezavisnih promjenljivih, što je uobičajeno veliki broj (10 do 30). Za smanjivanje reda problema, često se uvode određene pretpostavke i aproksimacije u zakon upravljanja. Sve ove različitosti definišu pojedine MPC algoritme. U nastavku je dat pregled nekih od najpoznatijih algoritama.

2.6.3.1 Upravljanje na bazi dinamičke matrice

Upravljanje na bazi dinamičke matrice (eng. *Dynamic Matrix Control - DMC*) koristi model zasnovan na odskočnom odzivu sistema ograničen na N odmjeraka, čime se pretpostavlja da je proces stabilan i da nema integratora. Po pitanju smetnji, njihova vrijednost se uzima da je jednaka kao u trenutku t duž cijelog horizonta, odnosno jednaka razlici stvarnog izlaza y_m i procjene izlaza $\hat{y}\left(t|\,t\right)$.

$$\hat{n}\left(t+k|\,t\right) = \hat{n}\left(t|\,t\right) = y_m(t) - \hat{y}\left(t|\,t\right), \tag{2.68}$$

pa je procjena izlaza

$$\hat{y}\left(t+k|\,t\right) = \sum_{i=1}^{k} h_i \Delta u(t+k-i) + \sum_{i=k+1}^{N} h_i \Delta u(t+k-i) + \hat{n}\left(t+k|\,t\right), \tag{2.69}$$

gdje prvi sabirak sadrži buduće upravljačke akcije koje treba izračunati, drugi sadrži poznate prethodne vrijednosti upravljačkog signala, a posljednji predstavlja smetnje.

Jedna od karakteristika ovog metoda je uključivanje ograničenja, što ga čini vrlo popularnim u industriji. Uvode se tako što se u minimizaciju uključuju nejednakosti tipa

$$\sum_{i=1}^{N} C_{yi}^{j} \hat{y}\left(t+k|\,t\right) + C_{ui}^{j} u(t+k-i) + c^{j} \leq 0, \tag{2.70}$$

gdje je $j = 1, 2, \ldots, N_c$, a N_c broj ograničenja.

Optimizacija (koja je numerička u slučaju korištenja ograničenja) se izvodi u svakom odmjerku.

Nepogodnosti ovog metoda su ranije pomenuta veličina modela, te nemogućnost rada sa nestabilnim procesima.

2.6.3.2 Algoritamsko upravljanje na bazi modela

Algoritamsko upravljanje na bazi modela (eng. *Model Algorithmic Control - MAC*) je slično prethodnom metodu. Koristi se model na bazi impulsnog odziva, ne odskočnog, a ne koristi se ni horizont za upravljanje. Uvodi se referentna putanja koja se mijenja od stvarnog izlaza do definisane tačke kao sistem prvog reda sa određenom vremenskom konstantom. Smetnje se mogu posmatrati kao kod DMC [77].

2.6.3.3 Prediktivno funkcionalno upravljanje

Prediktivno funkcionalno upravljanje (eng. *Predictive Functional Control - PFC*) se koristi kod brzih procesa. Koristi model u prostoru stanja i dozvoljava rad i sa nelinarnim, te sa nestabilnim linearnim sistemima. PFC ima dvije specifičnosti: podudarajuće tačke (eng. *coincidence points*) i funkcije baze (eng. *basis functions*).

Koncept podudarajućih tačaka se koristi za pojednostavljivanje proračuna tako što se razmatra samo podskup tačaka iz predikcionog horizonta. Željeni i predviđeni odziv treba da se podudaraju samo u tim tačkama, ne na cijelom horizontu.

Upravljački signal se predstavlja kao linearna kombinacija određenih unapred odabranih funkcija baza

$$u(t+k)=\sum_{i=1}^{n}\mu_i(t)B_i(k), \tag{2.71}$$

pri čemu se najčešće koriste polinomske funkcije baze $B_0=1$, $B_1=k$, $B_2=k^2$, itd.

Funkcija cilja koja se minimizuje je data sa

$$J=\sum_{j=1}^{n_H}\left[\hat{y}(t+j)-w(t+j)\right]^2. \tag{2.72}$$

2.6.3.4 Samoadaptivno upravljanje s produženom predikcijom

Samoadaptivno upravljanje s produženom predikcijom (eng. *Extended Prediction Self Adaptive Control - EPSAC*) se implementira drugačije od prethodno opisanih metoda. Proces se modeluje jednačinom

$$A\left(z^{-1}\right)y(t)=B\left(z^{-1}\right)u(t-d)+v(t), \tag{2.73}$$

gdje je d kašnjenje, a $v(t)$ smetnja.

Karakteristika ovog metoda je to što je zakon upravljanja jednostavan, jer se smatra da će upravljački signal ostati konstantan od trenutka t, odnosno da je $\Delta u(t+k)=0$ za $k>0$. Ukratko, upravljački horizont je skraćen na 1, a samim tim proračun sveden na računanje jedne vrijednosti - $u(t)$. Koristi se funkcija cilja

$$J=\sum_{k=d}^{N}\gamma(k)\left[P\left(z^{-1}\right)\hat{y}\left(t+k\right|t)-w(t+k)\right]^2, \tag{2.74}$$

pa se upravljački signal može izračunati analitički (prednost u odnosu na prethodne metode)

$$u(t)=\frac{\sum\limits_{k=d}^{N}g_k\gamma(k)\left[w(t+k)-P\left(z^{-1}\right)\hat{y}\left(t+k\right|t)\right]}{\sum\limits_{k=d}^{N}\gamma(k)g_k^2}. \tag{2.75}$$

2.6.3.5 Adaptivno upravljanje s proširenim horizontom

Adaptivno upravljanje s proširenim horizontom (eng. *Extended Horizon Adaptive Control - EHAC*) koristi modelovenje funkcijom prenosa bez razmatranja smetnji

$$A\left(z^{-1}\right)y(t)=B\left(z^{-1}\right)u(t-d). \tag{2.76}$$

Minimizira se razlika između izlaza modela i reference u trenutku $t+N$, odnosno $\hat{y}\left(t+N|\,t\right)-w(t+N)$, gdje je $N\geq d$. Rješenje je jedinstveno samo za $N=d$, pa se razmatraju različite strategije za nalaženje rješenja. Jedna je korištenje pristupa iz EPSAC

$$\Delta u(t+k-1)=0, 1<k\leq N-d, \tag{2.77}$$

a druga je minimizacija upravljačkog opterećenja

$$J=\sum_{k=0}^{N-d}u^2(t+k). \tag{2.78}$$

Postoji i inkrementalna varijanta EHAC kod koje se minimizuje

$$J=\sum_{k=0}^{N-d}\Delta u^2(t+k). \tag{2.79}$$

Za ovakvu formulaciju se koristi prediktor sa N koraka

$$\hat{y}\left(t+N|\,t\right)=y(t)+F\left(z^{-1}\right)\Delta y(t)+E\left(z^{-1}\right)B\left(z^{-1}\right)\Delta u(t+N-d), \tag{2.80}$$

gdje su $E\left(z^{-1}\right)$ i $F\left(z^{-1}\right)$ polinomi koji zadovoljavaju jednačinu

$$1-z^{-1}=A\left(z^{-1}\right)E\left(z^{-1}\right)\left(1-z^{-1}\right)+z^{-N}F\left(z^{-1}\right)\left(1-z^{-1}\right), \tag{2.81}$$

pri čemu je stepen polinoma E jednak $N-1$.

Kao kod EPSAC, postoji analitičko rješenje problema

$$u(t)=u(t-1)+\frac{\alpha_0\left(w(t+N)-\hat{y}\left(t+N|\,t\right)\right)}{\sum\limits_{k=0}^{N-d}\alpha_k^2}, \tag{2.82}$$

gdje je α_k koeficijent koji mijenja faktor koji stoji uz Δu u (2.80).

Zakon upravljanja tako zavisi samo od parametara procesa i lako može biti pretvoren u samopodešavajući dodavanjem on-line identifikacije. Jedini parametar za podešavanje je horizont predikcije N što pojednostavljuje korištenje, ali i ograničava slobodu u razvoju.

2.6.3.6 Uopšteno prediktivno upravljanje

Uopšteno (generalizovano) prediktivno upravljanje (eng. *Generalized Predictive Control - GPC*) koristi CARIMA model za predikciju

$$A\left(z^{-1}\right)y(t)=B\left(z^{-1}\right)z^{-d}u(t-1)+C\left(z^{-1}\right)\frac{e(t)}{\Delta}, \tag{2.83}$$

gdje su nemjerljive smetnje opisane bijelim šumom obojenim sa $C\left(z^{-1}\right)$.

Izvođenje optimalne predikcije se svodi na rješavanje Diofantske jednačine, za šta postoje efikasni rekurzivni algoritmi.

GPC koristi kvadratičnu funkciju cilja (2.66).

3. Vještačke neuralne mreže

Prvi talas interesovanja za vještačke neuralne mreže [84, 85, 86, 87, 88, 89, 90, 91, 92, 93] (uobičajeno nazivane skraćeno *neuralne mreže*) se pojavio nakon predstavljanja pojednostavljenih modela neurona u [94]. Ti neuroni su predstavljeni kao modeli bioloških neurona i kao konceptualne komponente za električna kola koja bi mogla izvršavati zadatke izračunavanja. Nedostaci modela peceptrona su pokazani u [95], što je dovelo do privremenog napuštanja istraživanja i preusmjeravanja finansiranja istraživanja na druga polja, uz rijetke izuzetke naučnika poput T. Kohonena [96] ili K. Fukušime [97]. U ranim osamdesetim godinama XX vijeka, nakon postizanja nekoliko važnih teoretskih rezultata (ponajviše pronalaskom propagacije greške unazad) obnavlja se interes za istraživanje neuralnih mreža, što se ogleda u povećanom broju naučnika koji su se okrenuli istraživanju tog polja, prilivom značajnih finansijskih sredstava u ta istraživanja, te formiranju brojnih naučnih konferencija i časopisa vezanih za neuralne mreže. Danas mnogi univerziteti imaju grupe za neuralne mreže unutar svojih odjeljenja za psihologiju, fiziku, biologiju ili računarske nauke.

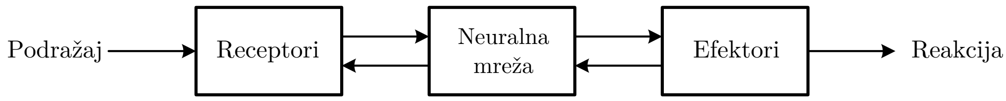
Neuralne mreže se vrlo adekvatno mogu opisati kao modeli sa osobinama poput sposobnosti prilagođavanja, odnosno učenja, generalizacije ili grupisanja, odnosno organizacije podataka, i čiji se rad zasniva na paralelnom procesiranju. Međutim, ove osobine posjeduju i drugi modeli. Intrigantno pitanje je do koje se mjere neuralni pristup pokazuje bolji od tih modela za određene aplikacije. Još uvijek nije dat univerzalan odgovor na to pitanje. Motivacija za rad sa neuralnim mrežama, od samog nastanka ideje, je u činjenici da mozak (najčešće se ovo odnosi na ljudski, ali nije ograničeno na to) izvršava razne operacije na potpuno drugačiji način od konvencionalnog digitalnog računara. Mozak je vrlo kompleksan, nelinearan i paralelan računar (sistem za procesiranje informacija). Ima sposobnost organizacije svojih strukturalnih činilaca - *neurona*, tako da izvodi procesiranja (npr. prepoznavanje uzoraka, upravljanje kretanjem i sl.) mnogo brže od najbržih dostupnih računara. Naravno, još uvijek se jako malo zna, čak i na najnižem nivou, ćelijskom, o biološkim sistemima, tako da su modeli vještačkih neuralnih sistema zapravo značajno pojednostavljenje njihovih bioloških uzora. Iako ne postižu ni sposobnosti, ni performanse bioloških sistema, neuralne mreže su aktuelan predmet istraživanja ponajviše zbog važne osobine preuzete iz bioloških sistema - *sposobnost korisnog procesiranja na osnovu naučenog*. Za postizanje dobrih performansi, u neuralnim mrežama se koristi veliki broj međusobno povezanih jednostavnih ćelija za procesiranje koje su nazvane, u skladu sa biološkim sistemom koji modeluju, neuroni (otud i ime samih mreža). U skladu s ovim, može se dati definicija neuralne mreže kako slijedi.

Definicija 3.1. Neuralna mreža je masivno paralelno distribuiran procesor sačinjen od jednostavnih procesnih jedinica koje imaju prirodnu sklonost čuvanju iskustvenog znanja i činjenju istog dostupnim za korištenje. Slične su mozgu na dva načina: znanje se stiče od okruženja učenjem i snage međuneuronskih veza, poznatih kao **sinaptičke težine**, se koriste za čuvanje stečenog znanja.

Procedura koja se koristi u procesu učenja se naziva *algoritam obučavanja*. Funkcija algoritma obučavanja je modifikacija sinaptičkih težina mreže na takav način da postigne željeni cilj. Ova modifikacija je tradicionalan metod dizajniranja neuralnih mreža. Međutim, moguće je da neuralna mreža sama modifikuje svoju topologiju, što je motivisano činjenicom da neuroni u mozgu mogu da odumru, kao i da se mogu razviti nove sinaptičke veze.

3.1 Ljudski mozak

Ljudski nervni sistem se može posmatrati kao sistem sa tri podsistema [93], kao na Slici 3.1. Centralni u sistemu je mozak koji kontinualno prima informacije, obrađuje ih i donosi odgovarajuće odluke. Strelice koje idu slijeva nadesno označavaju kretanje signala nosilaca informacija unaprijed kroz sistem (eng. *feed-forward*). Strelice koje idu zdesna nalijevo ukazuju na postojanje povratnih veza u sistemu (eng. *feedback*). Receptori pretvaraju stimuluse iz ljudskog tijela ili iz okoline u električne impulse koji prenose informacije do mozga. Efektori pretvaraju električne impulse koje generiše mozak u odgovarajuće odzive.

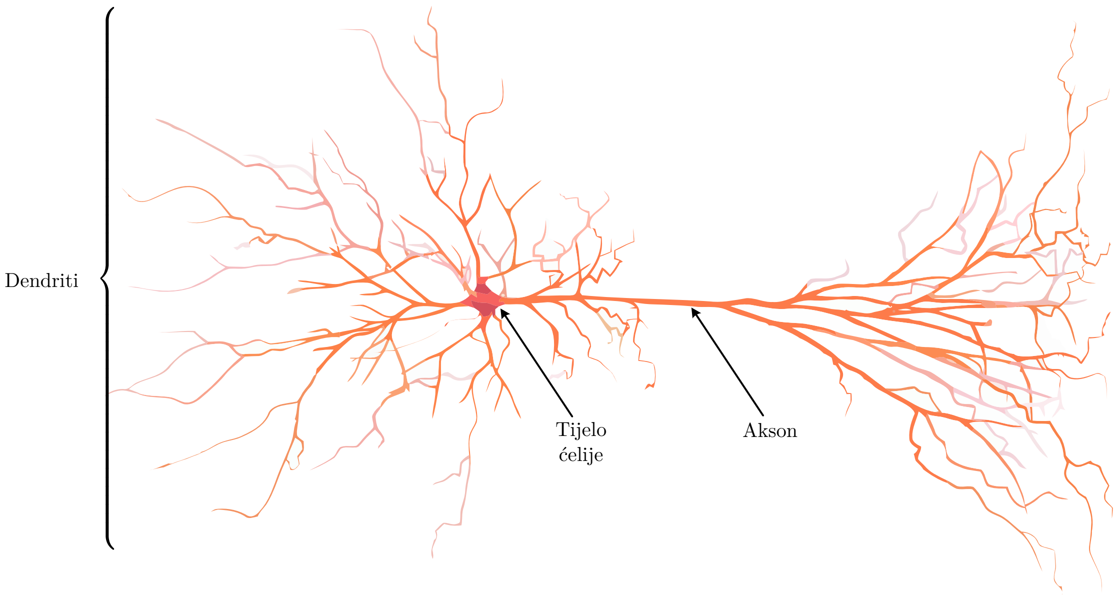


Slika 3.1 *Blok-dijagram ljudskog nervnog sistema*

Borba za razumijevanjem rada mozga je olakšana pionirskim radom [98], koji je uveo ideju neurona kao strukturnog činioca mozga. Zanimljivo je da su neuroni 5 do 6 redova veličine sporiji od silikonskih logičkih kola - događaji u silikonskim čipovima se dese u nanosekundama (10^{-9} s), pa čak i brže, a neuralni događaji se obave u milisekundama (10^{-3} s). Međutim, mozak nadoknađuje relativno malu brzinu rada neurona time što ima zapanjujuće veliki broj neurona koji su masovno međusobno povezani. Procijenjeno je da u ljudskom korteksu ima oko 10 milijardi neurona sa oko 60 biliona sinaptičkih veza među njima [99].

Sinapse su elementarne strukturne i funkcionalne jedinice koje obezbjeđuju interakciju između neurona. Najčešća varijanta su hemijske sinapse koje funkcionišu na sljedeći način: Presinaptički proces oslobađa emitujuću supstancu koja se difuzijom prostire kroz sinaptičke čvorove između neurona i tamo djeluje na postsinaptički proces. Tako sinapsa pretvara presinaptički električni signal u hemijski i onda nazad u postsinaptički električni. U skladu s tim, sinapsa se posmatra kao jednostavna veza koja može da uzrokuje pojačanje (eng. *excitation*) ili slabljenje (eng. *inhibition*), ali ne oboje, na neuron prijemnik. Mozak se mijenja na dva načina: stvaranjem novih, odnosno odumiranjem starih sinaptičkih veza ili modifikacijom postojećih.

Tipičan neuron se sastoji od tri komponente: tijela ćelije (some), aksona i dendrita. Jedan neuron ima jedan akson i uobičajeno veći broj dendrita. Akson je glatke površine, mnogo duži od dendrita i ima manji broj grana i to samo na jednom od krajeva, dok je dendrit razgranat na skoro cijeloj svojoj dužini. Glavna razlika je da akson služi za prenos impulsa od tijela ćelije, a dendriti prenose impulse do tijela ćelije. Neuroni imaju različite oblike i veličine u zavisnosti od položaja u mozgu. Najčešći oblik je *piramidalna ćelija*, kao na Slici 3.2. Piramidalna ćelija može primiti preko 10000 sinaptičkih kontakata, i emitovati prema hiljadama drugih ćelija.



Slika 3.2 *Neuron - piramidalna ćelija*

Većina neurona daje izlaz u vidu niza kratkih naponskih impulsa. Ovi impulsi, poznatiji kao *akcioni potencijali* ili *udari*, nastaju u ili blizu tijela ćelije i propagiraju duž neurona konstantnom brzinom. Akson karakterišu velika otpornost i kapacitivnost, tako da se može posmatrati kao RC transmisioni vod, pa se *jednačina kabla* (3.1) uobičajeno koristi za opisivanje propagacije signala kroz akson.

$$\frac{\partial V}{\partial t} = \frac{\partial^2 V}{\partial x^2} \tag{3.1}$$

Rješavanjem ove jednačine se dobija

$$V(x) = V_0 e^{-x}. \tag{3.2}$$

Na osnovu (3.2) se zaključuje da se napon koji se dovede na jedan kraj aksona eksponencijalno smanjuje sa udaljenošću, tj. tokom propagacije kroz akson, dostižući zanemarljivo male vrijednosti dok stigne na drugi kraj aksona. Zato se ovaj model može smatrati adekvatnim samo za kratke aksone. Akcioni potencijali, kao način komunikacije između neurona, pružaju način zaobilaženja ovog transmisionog problema.

3.2 Vještački neuron

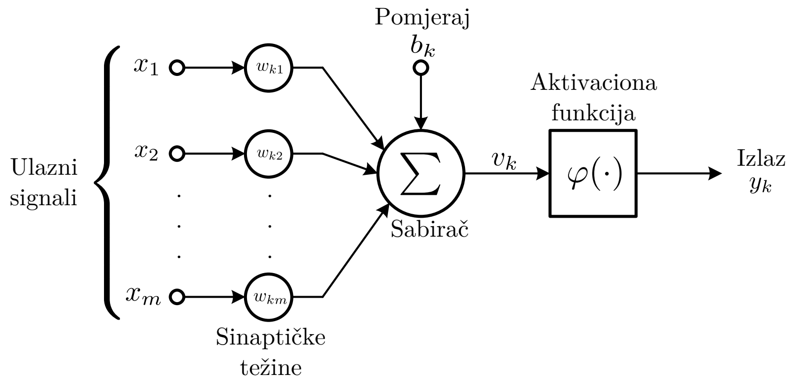
Vještački neuron je jedinica za procesiranje informacija koja je fundamentalna za rad neuralne mreže. Na Slici 3.3 je prikazan model neurona, na kojem se mogu identifikovati tri osnovna elementa:

1. Skup sinapsi, od kojih svaku karakteriše *težina* ili *snaga*. Ovo znači da se signal x_j na ulazu sinapse j povezane sa neuronom k množi sa težinom w_{kj}. Za razliku od sinapsi mozga, sinaptičke težine vještačkih neurona mogu biti u opsegu koji obuhvata i negativne vrijednosti;

2. Sabirač (sumator) za sabiranje ulaznih signala, pomnoženih odgovarajućim težinama. S obzirom da se formira linearna kombinacija ulaznih signala, ovakva operacija dati element čini *linearnim kombinatorom*;

3. Aktivaciona funkcija za ograničavanje amplitude izlaza neurona. Aktivaciona funkcija se takođe naziva i funkcija prigušenja (eng. *squashing function*), jer prigušuje pomenutu amplitudu. Uobičajeno je da se radi o normalizovanom opsegu izlaza, bilo $[0, 1]$, bilo $[-1, 1]$.

Model neurona često uključuje i vanjski pomjeraj (eng. *bias*), označen sa b_k, koji služi za povećanje ili smanjenje ukupnog ulaza aktivacione funkcije.



Slika 3.3 *Model vještačkog neurona*

Matematički, model neurona se može opisati parom jednačina:

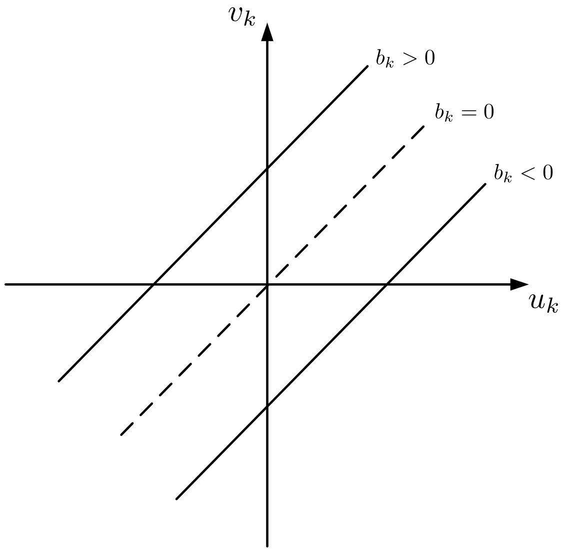
$$u_k = \sum_{j=1}^{m} w_{kj} x_j, \tag{3.3}$$

$$y_k = \varphi(u_k + b_k), \tag{3.4}$$

gdje su $x_1, x_2, \ldots, x_m$ ulazni signali; $w_{k1}, w_{k2}, \ldots, w_{km}$ sinaptičke težine k-tog neurona; u_k izlaz linearnog kombinatora; b_k pomjeraj; $\varphi(\cdot)$ aktivaciona funkcija; i y_k izlazni signal neurona. Korištenje pomjeraja ima efekat primjene afine transformacije (eng. *affine transformation*) na signal u_k koja daje signal v_k.

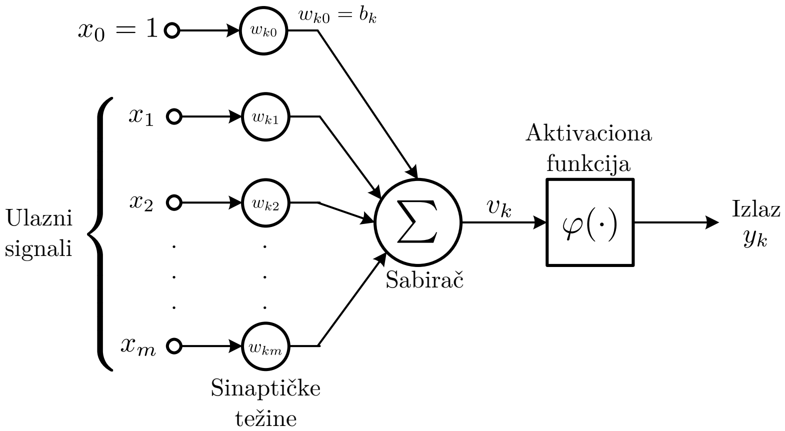
$$v_k = u_k + b_k \tag{3.5}$$

Konkretno, u zavisnosti od znaka pomjeraja, odnos između indukovanog lokalnog polja (aktivacionog potencijala) v_k i izlaza linearnog kombinatora u_k je prikazana na Slici 3.4.



Slika 3.4 *Uticaj pomjeraja na indukovano lokalno polje*

Model sa Slike 3.3 se može transformisati tako da se pomjeraj posmatra kao težina dodatne sinapse na čiji ulaz je doveden signal fiksne amplitude 1, kako je prikazano na Slici 3.5.



Slika 3.5 *Alternativni model vještačkog neurona*

3.2.1 Tipovi aktivacionih funkcija

Kako je ranije rečeno, aktivaciona funkcija, označena sa $\varphi(v)$, definiše izlaz neurona u zavisnosti od indukovanog lokalnog polja v. Ovdje su identifikovana tri osnovna tipa aktivacionih funkcija:

1. *Funkcija praga*. Ova funkcija prikazana na Slici 3.6a, matematički je opisana sa
$$\varphi(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases} \tag{3.6}$$
Ova funkcija se često naziva i Hevisajdova. Neuron sa ovakvom aktivacionom funkcijom se često naziva i *MekKaloh-Pits model*, po naučnicima koji su

predstavili takav model u svom pionirskom radu. Taj model je poznat po svojoj sve-ili-ništa osobini, u skladu sa aktivacionom funkcijom.

2. *Dio-po-dio linearna funkcija.* Ova funkcija prikazana na Slici 3.6b, matematički je opisana sa

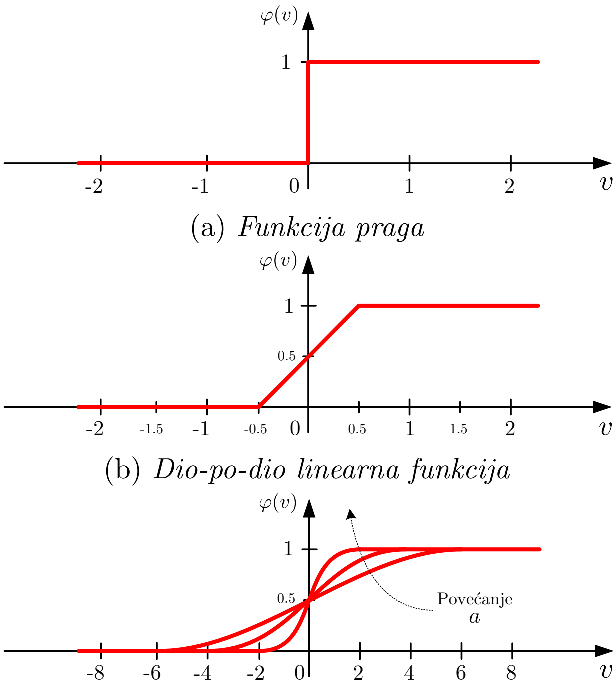
$$\varphi(v) = \begin{cases} 1, & v \geq \frac{1}{2} \\ v, & -\frac{1}{2} < v < \frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases} \tag{3.7}$$

Ovdje se za pojačanje u linearnoj oblasti uzima 1. Ovaj oblik funkcije se može posmatrati kao aproksimacija nelinearnog pojačavača. Funkcija zadržava vrijednost linearnog kombinatora dok god isti nije u zasićenju, inače se svodi na funkciju praga.

3. *Sigmoidalna funkcija.* Ovo je najčešće korištena aktivaciona funkcija u konstruisanju neuralnih mreža. Definisana je kao striktno rastuća funkcija koja ispoljava određenu ravnotežu između linearnog i nelinearnog ponašanja. Primjer sigmoidalne funkcije je logistička funkcija, definisana sa

$$\varphi(v) = \frac{1}{1 + e^{-av}}, \tag{3.8}$$

gdje je a parametar strmine (eng. *slope*). Promjena tog parametra je ilustrovana na Slici 3.6c. Kako se parametar nagiba približava beskonačnosti, logistička funkcija postaje funkcija praga. Za razliku od funkcije praga sa dvije diskretne vrijednosti 0 i 1, logistička funkcija preslikava na cijeli interval $[0, 1]$, a takođe je i diferencijabilna.



(c) *Sigmoidalna funkcija za različite vrijednosti parametra nagiba*

Slika 3.6 *Osnovni tipovi aktivacionih funkcija*

Aktivacione funkcije iz (3.6), (3.7) i (3.8) preslikavaju na interval $[0, 1]$. Ponekad je poželjno da aktivaciona funkcija preslikava na interval $[-1, 1]$. Tad se formiraju neparne funkcije, simetrične u odnosu na ishodište.

U tom slučaju funkcija praga je data sa

$$\varphi(v) = \begin{cases} 1, & v > 0 \\ 0, & v = 0 \\ -1, & v < 0 \end{cases} \tag{3.9}$$

Ovakva funkcija se često naziva i funkcija znaka (*signum*).

Odgovarajući oblik sigmoidalne funkcije je hiperbolička tangens funkcija, definisana sa

$$\varphi(v) = \tanh(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}}. \tag{3.10}$$

3.3 Topologije neuralnih mreža

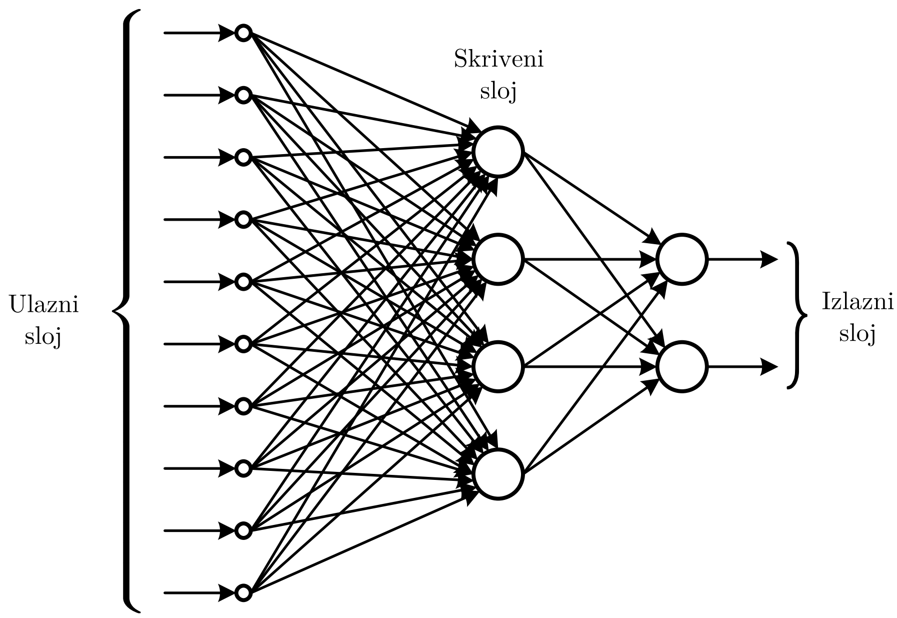
U zavisnosti od načina povezivanja neurona unutar mreže, te toka podataka kroz mrežu, razlikuju se dva tipa topologije neuralnih mreža: neuralne mreže bez povratnih veza (eng. *feedforward*) i rekurentne neuralne mreže. Uobičajeno je da su neuroni organizovani u slojeve. Neuroni čiji su ulazi povezani direktno sa ulazima mreže pripadaju *ulaznom sloju* mreže. Oni čiji su izlazi povezani direktno sa izlazima mreže pripadaju *izlaznom sloju* mreže. Treba naglasiti da je moguće da postoji samo jedan sloj koji je onda istovremeno i ulazni i izlazni. Ukoliko postoje neuroni koji nisu ni iz ulaznog, ni iz izlaznog sloja, onda oni pripadaju jednom od proizvoljnog broja skrivenih (unutrašnjih) slojeva mreže. Često se unutrašnji slojevi numerišu posmatrano od ulaznog prema izlaznom sloju. Konkretno, prvi unutrašnji sloj čine neuroni čiji ulazi su povezani sa izlazima neurona ulaznog sloja, itd.

3.3.1 Neuralne mreže bez povratnih veza

Kod neuralnih mreža bez povratnih veza tok podataka od ulaznog do izlaznog sloja je isključivo unaprijed, tj. nema veza između izlaza neurona jednog sloja i ulaza neurona istog, niti nekog od prethodnih slojeva. Na Slici 3.7 je prikazana jedna takva mreža.

Često se ovakve mreže označavaju na osnovu broja neurona u svakom od slojeva. Ako mreža ima 4 sloja (ulazni, dva skrivena i izlazni) koji imaju redom m, h_1, h_2 i q neurona, tada se ta mreža označava sa $m - h_1 - h_2 - q$. U skladu s tim, mreža sa Slike 3.7 je $10 - 4 - 2$ mreža.

Ukoliko su svi izlazi neurona jednog sloja (izuzev izlaznog) povezani sa svim ulazima neurona sljedećeg sloja, tada se takva mreža naziva potpuno povezanom (eng. *fully connected*). Takva je mreža sa Slike 3.7. Ukoliko neka od veza nedostaje, mreža je djelimično povezana (eng. *partially connected*).



Slika 3.7 *Neuralna mreža sa jednim skrivenim slojem bez povratnih veza*

3.3.2 Rekurentne neuralne mreže

Za razliku od prethodno opisanih mreža, rekurentne mreže imaju bar jednu povratnu vezu (eng. *feedback*), tj. vezu izlaza neurona jednog sloja sa ulazima neurona istog ili nekog od prethodnih slojeva. Ukoliko je vezan izlaz neurona sa ulazom upravo tog neurona, radi se o samopovratnoj petlji (eng. *self-feedback loop*). Povratne veze uglavnom uključuju i posebne elemente koji se nazivaju elementi za jedinično kašnjenje (eng. *unit-delay elements*), koji se označavaju sa z^{-1}. Ti elementi rezultuju u nelinearnom dinamičkom ponašanju mreže.

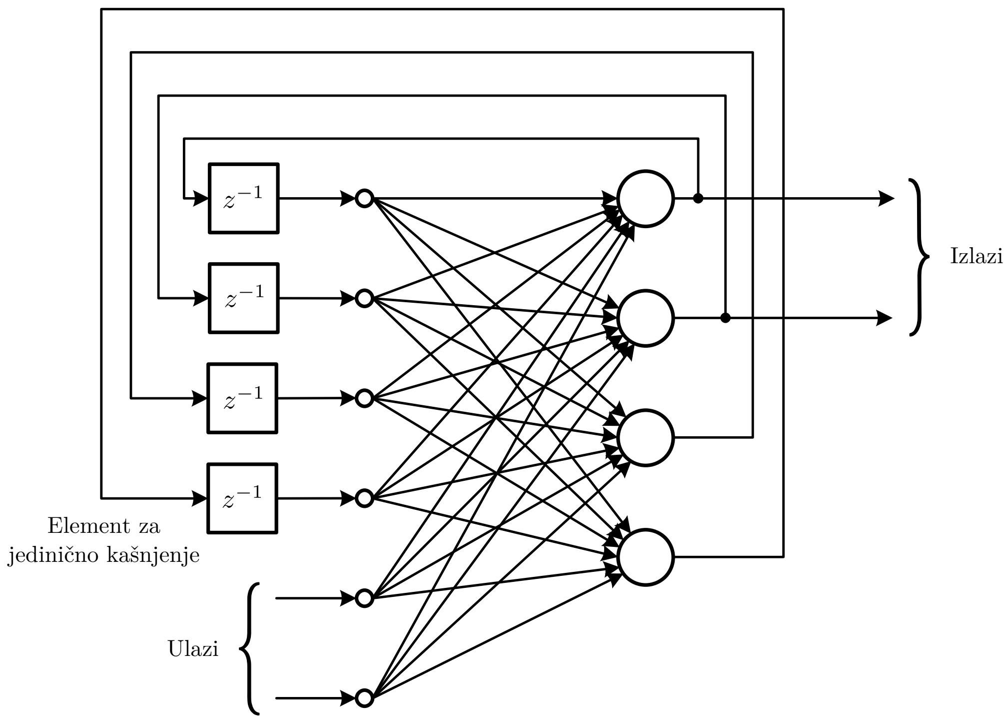
Na Slici 3.8 je prikazana jedna višeslojna rekurentna mreža sa povratnim vezama koje potiču i iz skrivenog i iz izlaznog sloja, bez samopovratnih petlji.

3.4 Obučavanje neuralnih mreža

Neuralne mreže treba da budu podešene tako da primjena određenog skupa ulaznih vrijednosti rezultuje željenim skupom izlaznih vrijednosti. Postoje različiti načini da se podese težine sinaptičkih veza, ali najčešći je obučavanje (trening), za vrijeme kojeg se mreži zadaju određeni uzorci, te se težine mijenjaju u skladu sa nekim pravilom učenja.

Obučavanje se može podijeliti na dvije vrste (paradigme):

1. *Nadgledano* ili *asocijativno učenje* kod kojeg se mreža obučava tako što joj se zadaju uzorci ulaza i odgovarajućih izlaza. Ovi uzorci se mreži zadaju od strane vanjskog učitelja, ili od strane samog sistema koji sadrži mrežu (*samonadgledano učenje*).
2. *Nenadgledano učenje* ili *Samoorganizacija* u kojem mreža sama prepoznaje uzorke u ulaznim signalima.



Slika 3.8 *Rekurentna neuralna mreža*

Obje paradigme rezultuju u podešavanju težina veza na osnovu nekog pravila. Skoro sva pravila se mogu smatrati varijacijom Hebovog pravila obučavanja [100]. Osnovna ideja je da, ako su jedinice j i k aktivne istovremeno, njihova međusobna veza treba da bude pojačana. Ako j prima signal od k, najjednostavnija verzija Hebovog pravila preporučuje da se modifikuje težina w_{jk} za

$$\Delta w_{jk} = \gamma y_j y_k, \tag{3.11}$$

gdje je γ pozitivna konstanta koja predstavlja brzinu obučavanja (eng. *learning rate*).

Drugo često pravilo ne koristi samu aktivaciju k-te jedinice, već razliku između stvarne i željene aktivacije, za korekciju težine

$$\Delta w_{jk} = \gamma y_j (d_k - y_k), \tag{3.12}$$

gdje je d_k željena aktivacija koju zadaje učitelj. Ovo se često naziva *delta pravilo*.

3.5 Perceptron i Adaline

U ovom potpoglavlju opisane su neuralne mreže sa jednim slojem, kao i neki klasični pristupi neuralnom računanju i problemima obučavanja, i to u vidu pregleda dva istorijski značajna modela: Perceptron [101, 102] i Adaline [103].

3.5.1 Jednoslojne mreže sa funkcijama praga

Jednoslojna mreža bez povratnih veza se sastoji od jednog ili više izlaznih neurona o, od kojih je svaki povezan sa svakim od ulaza i vezom težine w_{io}. U najjednostavnijem slučaju, mreža ima samo dva ulaza i jedan izlaz (u tom slučaju može da se izostavi indeks o). Ulaz neurona je težinska suma ulaza plus pomjeraj. Izlaz mreže se formira na osnovu aktivacione funkcije izlaznog neurona prema

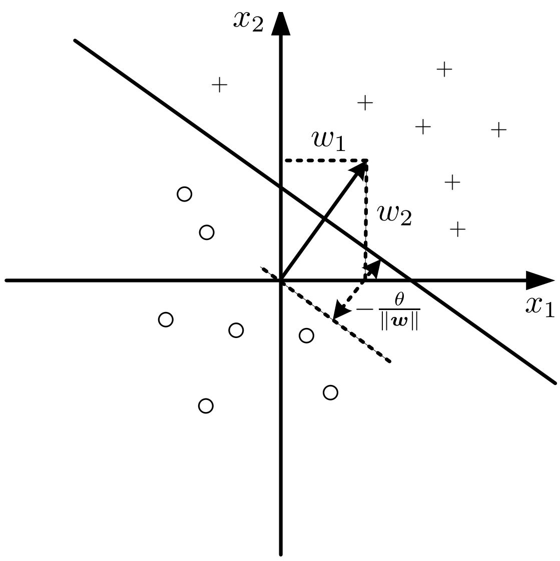
$$y = \mathcal{F}\left(\sum_{i=1}^{2} w_i x_i + \theta\right). \tag{3.13}$$

Aktivaciona funkcija $\mathcal{F}$ može biti linearna ili nelinearna. Neka je, u ovom slučaju, aktivaciona funkcija jedna varijanta funkcije praga

$$\mathcal{F} = \begin{cases} 1, & v \geq 0 \\ -1, & v < 0 \end{cases} \tag{3.14}$$

Iz (3.14) je jasno da je izlaz mreže 1 ili -1. Ovakva mreža se može koristiti za zadatak *klasifikacije*: može da odluči kojoj od dvije klase ulazni uzorak pripada. Ako je ukupan ulaz pozitivan, uzorku će biti dodijeljena klasa 1, a ako je ukupan ulaz negativan, uzorku će biti dodijeljena klasa -1. Ono što razdvaja dvije klase u ovom slučaju je prava linija, definisana jednačinom

$$w_1 x_1 + w_2 x_2 + \theta = 0. \tag{3.15}$$



Slika 3.9 *Linearno razdvajanje dvije klase*

Dakle, jednoslojna mreža je *linearno diskriminantna*, a ovakve dvije klase se nazivaju *linearno separabilne*. Geometrijska reprezentacija linearne jednoslojne neuralne mreže sa funkcijama praga je data na Slici 3.9. Jednačina (3.15) se može zapisati (kad je $w_2 \neq 0$, što je očekivano, jer bi inače funkcija zavisila samo od jednog ulaza, što je trivijalno) i u obliku

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{\theta}{w_2}, \tag{3.16}$$

odakle se vidi da težine veza određuju nagib prave, a uz pomjeraj određuju i udaljenost prave od ishodišta.

U nastavku su opisana dva načina obučavanja (određivanja težina i pomjeraja) ovakvih mreža: pravilo obučavanja perceptrona i delta pravilo. Oba metoda su iterativne procedure koje podešavaju težine na osnovu uzorka za obučavanje. Kako je opisano u Potpoglavlju 3.4, kod oba ova metoda nove vrijednosti težina se računaju dodavanjem korekcije na staru vrijednost. Isto važi i za pomjeraj.

$$w_i(t+1) = w_i(t) + \Delta w_i(t) \tag{3.17}$$

$$\theta_i(t+1) = \theta_i(t) + \Delta\theta_i(t) \tag{3.18}$$

3.5.2 Pravilo obučavanja perceptrona i teorema konvergencije

Neka je dat skup obučavajućih uzoraka koji se sastoje od ulaznog vektora $\boldsymbol{x}$ i željenih izlaza $d(\boldsymbol{x})$. Za zadatak klasifikacije $d(\boldsymbol{x})$ je uobičajeno 1 ili -1. Pravilo obučavanja perceptrona je jednostavno i može se napisati kako slijedi:

1. Težinama veza se daju proizvoljne početne vrijednosti;
2. Bira se ulazni vektor $\boldsymbol{x}$ iz skupa obučavajućih uzoraka koji se dovodi na ulaz mreže, te se računa izlaz mreže y;
3. Ako je $y \neq d(\boldsymbol{x})$ (perceptron daje pogrešan odgovor), modifikuju se sve težine w_i u skladu sa $\Delta w_i = d(\boldsymbol{x})x_i$;
4. Povratak na 2. dok se ne prođu svi uzorci.

Treba primijetiti da je procedura slična Hebovom pravilu; jedina razlika je to što nema korekcija težina ako je izlaz mreže korektan. Takođe, potrebno je modifikovati i pomjeraj θ. Ako se pomjeraj posmatra kao težina veze w_0 *lažnog* ulaza koji ima konstantu vrijednost 1, kako je opisano u Potpoglavlju 3.2, onda se u skladu sa prethodno opisanom procedurom pomjeraj modifikuje u skladu sa

$$\Delta\theta = \begin{cases} 0, & \text{perceptron daje tačan odgovor} \\ d(\boldsymbol{x}), & \text{inače} \end{cases} \tag{3.19}$$

Za pravilo obučavanja perceptrona postoji i sljedeća teorema konvergencije:
Teorema 3.1. *Ako postoji vektor težina veza $\boldsymbol{w}^*$ koji omogućava transformaciju $y = d(\boldsymbol{x})$, pravilo obučavanja perceptrona će konvergirati nekom rješenju (koje može, a ne mora biti $\boldsymbol{w}^*$) u konačnom broju koraka za bilo koju početnu vrijednost težina.*

Dokaz. S obzirom na činjenicu da je aktivaciona funkcija neurona zapravo funkcija znaka, skaliranjem svih težina nekim pozitivnim brojem neće se promijeniti znak težinske sume svih ulaza, pa samim tim ni rezultat klasifikacije. Zato se može uzeti da je $\|\boldsymbol{w}^*\| = 1$, gdje je $\|\cdot\|$ Euklidska norma vektora ,tj. $\|\boldsymbol{a}\| = \sqrt{\sum_{i=1}^{n} {a_i}^2}$, gdje je $\boldsymbol{a} = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}$. Dalje, kako je $\boldsymbol{w}^*$ takav vektor težina veza da mreža sa takvim težinama vrši dobru klasifikaciju, tada je $|\boldsymbol{w}^* \cdot \boldsymbol{x}| > \delta$, $\delta > 0$, za svako $\boldsymbol{x}$, gdje je $\cdot$ skalarni proizvod vektora. Neka je $\boldsymbol{w}$ početna vrijednost vektora težina veza i neka sa takvim težinama mreža ne vrši korektnu klasifikaciju za ulazni vektor $\boldsymbol{x}$. Tada je prva modifikacija, u skladu sa pravilom obučavanja perceptrona, $\boldsymbol{w'} = \boldsymbol{w} + \Delta\boldsymbol{w} = \boldsymbol{w} + d(\boldsymbol{x})\boldsymbol{x}$. Odatle slijedi

$$\boldsymbol{w'} \cdot \boldsymbol{w}^* = \boldsymbol{w} \cdot \boldsymbol{w}^* + d(\boldsymbol{x})\boldsymbol{w}^* \cdot \boldsymbol{x}$$

Kako je klasifikacija pogrešna, sigurno je $d(\boldsymbol{x}) = -\operatorname{sgn}(\boldsymbol{w} \cdot \boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w}^* \cdot \boldsymbol{x})$. Tada je

$$\boldsymbol{w'} \cdot \boldsymbol{w}^* = \boldsymbol{w} \cdot \boldsymbol{w}^* + \operatorname{sgn}(\boldsymbol{w}^* \cdot \boldsymbol{x})\boldsymbol{w}^* \cdot \boldsymbol{x} = \boldsymbol{w} \cdot \boldsymbol{w}^* + |\boldsymbol{w}^* \cdot \boldsymbol{x}| > \boldsymbol{w} \cdot \boldsymbol{w}^* + \delta$$

Takođe važi i sljedeće

$$\|\boldsymbol{w'}\|^2 = \|\boldsymbol{w} + d(\boldsymbol{x})\boldsymbol{x}\|^2 = \|\boldsymbol{w}\|^2 + 2d(\boldsymbol{x})\boldsymbol{w} \cdot \boldsymbol{x} + \|\boldsymbol{x}\|^2 < \|\boldsymbol{w}\|^2 + \|\boldsymbol{x}\|^2 = \|\boldsymbol{w}\|^2 + M$$

Ukoliko se radi o t modifikacija, važi

$$\boldsymbol{w(t)} \cdot \boldsymbol{w}^* > \boldsymbol{w} \cdot \boldsymbol{w}^* + t\delta$$

$$\|\boldsymbol{w'}\|^2 < \|\boldsymbol{w}\|^2 + tM$$

Neka je α ugao između $\boldsymbol{w}$ i $\boldsymbol{w}^*$. Tada je, na osnovu skalarnog proizvoda

$$\cos(\alpha(t)) = \frac{\boldsymbol{w(t)} \cdot \boldsymbol{w}^*}{\|\boldsymbol{w(t)}\|^2} > \frac{\boldsymbol{w} \cdot \boldsymbol{w}^* + t\delta}{\sqrt{\|\boldsymbol{w}\|^2 + tM}}$$

Ako bi bilo moguće da $t \to \infty$, onda bi bilo i $\cos(\alpha(t)) \to \infty$, što je nemoguće, pa slijedi da je broj modifikacija t ograničen, tj. konačan. Ukoliko je početna vrijednost težina $\boldsymbol{w} = 0$, onda je

$$t_{max} = \frac{M}{\delta^2}$$

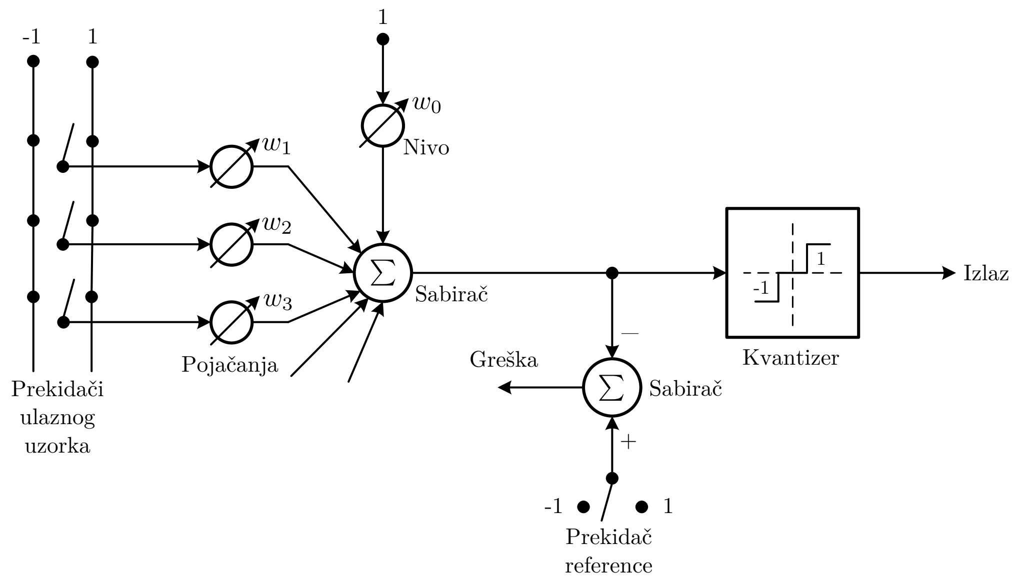
■

3.5.3 Adaptivni linearni element - Adaline

Važna generalizacija algoritma perceptrona je procedura zasnovana na metodi najmanjih kvadrata, poznata i kao delta pravilo. Glavna funkcionalna razlika u odnosu na pravilo obučavanja perceptrona je način korištenja izlaza mreže pri obučavanju. Takvo obučavanje je primijenjeno na adaptivni linearni element, poznat i po akronimu Adaline[1]. U jednostavnoj fizičkoj implementaciji, prikazanoj na Slici 3.10, uređaj se sastoji od skupa podesivih otpornika povezanih na kolo koje može da sabere struje nastale pod uticajem ulaznih napona. Obično je centralni blok, sabirač, praćen kvantizerom koji kao izlaz daje 1 ili -1 u zavisnosti od polariteta sume.

Treba naglasiti da je u primjeru prikazana izvedba sa jednim izlazom, ali se korištenjem više takvih jedinica može dobiti sistem sa više paralelnih izlaza.

[1] ADALINE je prvobitno uzeto od ADAptivni LInearni NEuron, ali je zbog gubitka popularnosti neurona promijenjeno na ADAptivni LINearni Element. Naravno, obje varijante su izvedene od zapisa na engleskom jeziku, ali kako su odgovarajuća slova ista, izabran je navedeni zapis.



Slika 3.10 *Adaline*

Neka su ulazne provodnosti označene sa w_i, $i = 0, 1, 2, \ldots, n$, a ulazni i izlazni signali sa x_i i y respektivno. Tada je izlaz centralnog bloka definisan sa

$$y = \sum_{i=0}^{n} w_i x_i = \sum_{i=1}^{n} w_i x_i + \theta, \tag{3.20}$$

gdje je $\theta \equiv w_0$. Svrha ovakvog uređaja je da se dobije željeni izlaz $y = d^p$ kada je skup vrijednosti x_i^p, $i = 0, 1, 2, \ldots, n$ primijenjen na ulaze. Problem je odrediti koeficijente w_i, $i = 0, 1, 2, \ldots, n$ na takav način da je odziv dobar za veliki broj proizvoljno izabranih skupova signala. Ako tačno mapiranje nije moguće, prosječna greška treba da bude minimizovana, na primjer u smislu najmanjih kvadrata. Operacija adaptacije znači da postoji mehanizam na osnovu kojeg se podešavaju w_i, obično iterativni, da se postignu dobre vrijednosti. Za Adaline, uvedeno je delta pravilo koje je opisano u nastavku.

3.5.4 Mreže sa linearnim aktivacionim funkcijama i delta pravilo

Za jednoslojnu mrežu sa jednim izlaznim neuronom sa linearnom funkcijom, izlaz je jednostavno definisan sa

$$y = \sum_{j} w_j x_j + \theta, \tag{3.21}$$

Takva jednostavna mreža može da predstavi linearnu zavisnost vrijednosti izlazne jedinice od vrijednosti ulaznih jedinica. Dodavanjem funkcije praga na izlaznu vrijednost (kao kod Adaline-a) moguće je konstruisati klasifikator, ali ovdje je dat fokus linearnoj zavisnosti i korištenju mreže za zadatak *aproksimacije funkcije*. U visokodimenzionalnom prostoru ulaza mreža predstavlja hiper-površ.

Neka je potrebno obučiti mrežu za aproksimaciju hiper-površi na osnovu obučavajućih uzoraka koji se sastoje od ulaznih vrijednosti x^p i željenih izlaznih vrijednosti d^p. Za svaki zadati uzorak, izlaz mreže y^p se razlikuje od željene vrijednosti za $(d^p - y^p)$. Delta pravilo koristi funkciju koštanja ili funkciju greške, zasnovane na toj razlici, za podešavanje težina veza.

Funkcija greške, kako se vidi iz imena *najmanji kvadrati*, je suma kvadrata grešaka, tj. ukupna greška E je definisana sa

$$E = \sum_p E^p = \frac{1}{2} \sum_p \left(d^p - y^p\right)^2, \tag{3.22}$$

gdje indeks p uzima vrijednosti tako da pokrije sve ulazne uzorke. Metoda najmanjih kvadrata je procedura koja nalazi vrijednosti težina veza tako da funkcija greške bude minimalizovana pomoću *gradijentnog spuštanja*. Ideja je da se težine mijenjaju proporcionalno negativnoj vrijednosti izvoda greške izračunate za svaki od uzoraka, po svakoj od težina

$$\Delta_p w_j = -\gamma \frac{\partial E^p}{\partial w_j}, \tag{3.23}$$

gdje je γ konstanta proporcionalnosti. Za izvod važi

$$\frac{\partial E^p}{\partial w_j} = \frac{\partial E^p}{\partial y^p} \frac{\partial y^p}{\partial w_j}. \tag{3.24}$$

Iz (3.21) važi

$$\frac{\partial y^p}{\partial w_j} = x_j, \tag{3.25}$$

a iz (3.22)

$$\frac{\partial E^p}{\partial y^p} = -\left(d^p - y^p\right), \tag{3.26}$$

pa je, konačno,

$$\Delta_p w_j = \gamma \delta^p x_j, \tag{3.27}$$

gdje je $\delta^p = (d^p - y^p)$.

Delta pravilo modifikuje težine za željene i stvarne izlaze bilo kog polariteta, kao i za obje vrste signala - kontinualne i binarne. Ova osobina otvara vrata mnoštvu novih primjena.

3.6 Propagacija greške unazad

Kako je opisano u prethodnom potpoglavlju, jednoslojne mreže su veoma ograničene po pitanju zadataka koje mogu obaviti (npr. klasifikacija je moguća samo za linearno separabilne klase). U ovom potpoglavlju dat je pregled višeslojnih neuralnih mreža bez povratnih veza.

Pri pokazivanju ograničenja modela perceptrona u [95], autori su pokazali da dvoslojna mreža može da prevaziđe mnoga od tih ograničenja, ali nisu predstavili rješenje problema podešavanja težina veza od ulaznog do skrivenog sloja. Odgovor na to pitanje je predstavljen u [104], ali slična rješenja su objavljena i ranije [105, 106, 107].

Glavna ideja ovog rješenja je da su greške za neurone u skrivenom sloju određene propagacijom grešaka neurona izlaznog sloja unazad. Zbog toga se metoda i zove *pravilo obučavanja propagacijom greške unazad*. Propagacija greške unazad se može smatrati uopštenjem delta pravila za višeslojne mreže sa nelinearnim aktivacionim funkcijama[2].

3.6.1 Višeslojne mreže bez povratnih veza

Struktura višeslojnih mreža bez povratnih veza je opisana u Potpoglavlju 3.3. Uobičajeno je da ulazni neuroni imaju aktivacionu funkciju koja samo preslikava (propušta) ulaz na izlaz.

Iako se propagacija greške unazad može primijeniti na mreže sa proizvoljnim brojem slojeva, pokazano je [108, 109, 110] da je samo jedan skriveni sloj neurona dovoljan za aproksimaciju proizvoljne funkcije sa konačnim brojem prekida, do proizvoljne preciznosti, pod uslovom da su aktivacione funkcije neurona skrivenog sloja nelinearne. U većini aplikacija koriste se mreže bez povratnih veza sa jednim skrivenim slojem, sa sigmoidalnim aktivacionim funkcijama.

3.6.2 Uopšteno delta pravilo

Aktivaciona funkcija je diferencijabilna funkcija ukupnog ulaza neurona k-tog sloja

$$y_k^p = \mathcal{F}\left(s_k^p\right), \tag{3.28}$$

gdje je

$$s_k^p = \sum_j w_{jk} y_j^p + \theta_k. \tag{3.29}$$

Generalizacija delta pravila iz (3.23) je data sa

$$\Delta_p w_{jk} = -\gamma \frac{\partial E^p}{\partial w_{jk}}. \tag{3.30}$$

Greška E^p je definisana kao ukupna srednjekvadratna greška za uzorak p u neuronima izlaznog sloja, u skladu sa (3.22)

$$E = \sum_p E^p = \frac{1}{2} \sum_p \left(d_o^p - y_o^p\right)^2, \tag{3.31}$$

gdje je d_o^p željeni izlaz neurona o za uzorak p. Analogno delta pravilu za jednoslojne mreže, važi

$$\frac{\partial E^p}{\partial w_{jk}} = \frac{\partial E^p}{\partial s_k^p} \frac{\partial s_k^p}{\partial w_{jk}}, \tag{3.32}$$

[2]Ako bi se koristile linearne funkcije kod višeslojnih mreža, ne bi bilo nikakvog unapređenja u odnosu na jednoslojne, jer bi izlaz i dalje bio linearna kombinacija ulaza

te

$$\frac{\partial s_k^p}{\partial w_{jk}} = y_j^p. \tag{3.33}$$

Ako se definiše

$$\delta_k^p = -\frac{\partial E^p}{\partial s_k^p}, \tag{3.34}$$

dobija se pravilo za korekciju ekvivalentno delta pravilu za jednoslojne mreže, koje predstavlja gradijentno spuštanje po površi greške

$$\Delta_p w_{jk} = \gamma \delta_k^p y_j^p. \tag{3.35}$$

U nastavku je pokazano da je moguće jednostavno rekurzivno računanje δ faktora na osnovu propagacije greške unazad kroz mrežu.

Za računanje δ_k^p primjenjuje se i ranije primijenjeno pravilo ulančavanja izvoda u vidu proizvoda dva faktora, pri čemu su u ovom slučaju ta dva faktora izvodi koji pokazuju promjenu greške u funkciji izlaza datog neurona, te promjenu tog izlaza u funkciji ulaza tog neurona. To se može zapisati kao

$$\delta_k^p = -\frac{\partial E^p}{\partial s_k^p} = -\frac{\partial E^p}{\partial y_k^p}\frac{\partial y_k^p}{\partial s_k^p}. \tag{3.36}$$

Iz (3.28) se dobija

$$\frac{\partial y_k^p}{\partial s_k^p} = \mathcal{F}'\left(s_k^p\right), \tag{3.37}$$

što je jednostavno izvod aktivacione funkcije $\mathcal{F}$ k-tog neurona, izračunat kad je ulaz tog neurona s_k^p.

Za računanje prvog faktora iz (3.36) razmatraju se dva slučaja. Prvi je kad se radi o neuronima izlaznog sloja. Tada je iz definicije greške

$$\frac{\partial E^p}{\partial y_k^p} = -\left(d_k^p - y_k^p\right), \tag{3.38}$$

što je rezultat koji se dobija i kod standardnog delta pravila. Ako se ovo i (3.37) uvrsti u (3.36), dobija se

$$\delta_k^p = \left(d_k^p - y_k^p\right)\mathcal{F}'\left(s_k^p\right), \tag{3.39}$$

za svaki od izlaznih neurona.

Drugi slučaj podrazumijeva neurone skrivenih slojeva za koje nije očigledan doprinos grešci izlaza mreže. Međutim, mjera greške se može napisati kao funkcija ukupnih ulaza od skrivenog do izlaznog sloja $E^p = E^p\left(s_1^p, s_2^p, \ldots, s_j^p, \ldots\right)$ i može se koristiti pravilo ulančavanja izvoda

$$\begin{aligned} \frac{\partial E^p}{\partial y_k^p} &= \sum_{o=1}^{N_o} \frac{\partial E^p}{\partial s_o^p}\frac{\partial s_o^p}{\partial y_k^p} = \sum_{o=1}^{N_o} \frac{\partial E^p}{\partial s_o^p}\frac{\partial}{\partial y_k^p}\sum_{j=1}^{N_k} w_{ko} y_j^p \\ &= \sum_{o=1}^{N_o} \frac{\partial E^p}{\partial s_o^p} w_{ko} = -\sum_{o=1}^{N_o} \delta_o^p w_{ko}. \end{aligned} \tag{3.40}$$

Uvrštavanjem ovoga u (3.36) dobija se

$$\delta_k^p = \mathcal{F}'\left(s_k^p\right) \sum_{o=1}^{N_o} \delta_o^p w_{ko}. \tag{3.41}$$

Jednačine (3.39) i (3.41) određuju rekurzivnu proceduru za računanje δ faktora za sve neurone u mreži, koji se dalje koriste za računanje promjena težina na osnovu (3.30). Ova procedura čini uopšteno delta pravilo za mreže bez povratnih veza sa nelinearnim aktivacionim funkcijama.

Ovdje se može vidjeti i pogodnost izbora sigmoidalne funkcije za aktivacionu funkciju. Iz (3.39) se vidi da korekcije težina zavise od izvoda aktivacione funckije (jer *delta* faktor zavisi od istog). Kako je rečeno u Potpoglavlju 3.2, najčešće korištena varijanta sigmoidalne funkcije je logistička funkcija

$$y^p = \mathcal{F}\left(s^p\right) = \frac{1}{1 + e^{-s^p}}. \tag{3.42}$$

U jednačini (3.8) je u eksponentu bio i parametar nagiba. U nastavku će biti pokazano zašto u (3.42) ne figuriše taj parametar.

Za izvod jednačine (3.42) važi

$$\mathcal{F}'\left(s^p\right) = \frac{e^{-s^p}}{\left(1 + e^{-s^p}\right)^2} = \frac{1}{1 + e^{-s^p}} \frac{e^{-s^p}}{1 + e^{-s^p}} = y^p\left(1 - y^p\right). \tag{3.43}$$

Ova karakteristika značajno olakšava/ubrzava računanje izvoda sigmoidalne funkcije, samim tim i cijeli proces obučavanja propagacijom greške unazad.

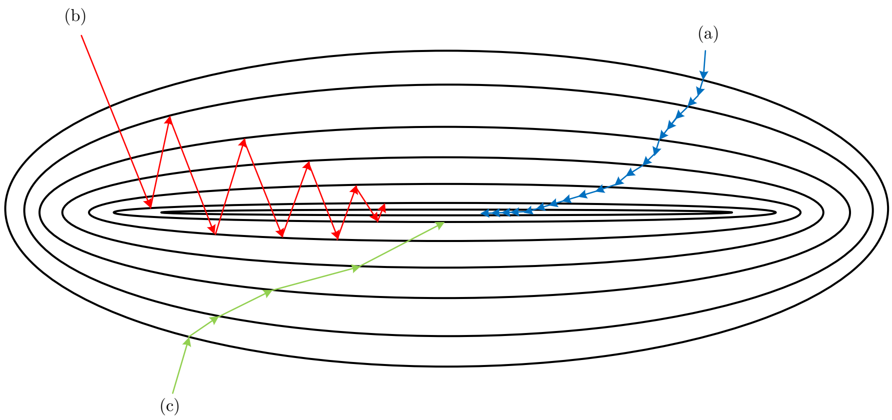
3.6.3 Brzina obučavanja i momentum član

Kod osnovnog algoritma gradijentnog spuštanja, koriste se infinitezimalni koraci prema rješenju. Kako je ovo nepraktično za realne potrebe, uvodi se i konstanta proporcionalnosti - brzina obučavanja γ. Izbor vrijednosti brzine obučavanja nema jasnu matematičku formulu. Što je veća vrijednost, veća će biti i brzina konvergencije, ali i vjerovatnoća pojavljivanja oscilacija oko stvarnog rješenja. Jedan način da se izbjegnu oscilacije za velike vrijednosti γ je učiniti vrijednost promjene težine veze zavisnom od prethodne vrijednosti promjene težine dodavanjem *momentum člana*

$$\Delta w_{jk}(t+1) = \gamma \delta_k^p y_j^p + \alpha \Delta w_{jk}(t), \tag{3.44}$$

gdje t označava broj iteracije, a α je konstanta koja određuje uticaj prethodne vrijednosti promjene.

Uloga momentum člana je prikazana na Slici 3.11. Kada se ne koristi momentum član, a mala je brzina obučavanja, potrebno je mnogo vremena za dostizanje minimuma, dok se kod velike brzine obučavanja minimum ni ne dostiže zbog pojave oscilacija. Dodavanjem momentum člana, oscilacije se eliminišu, a vrijeme dostizanja minimuma skraćuje.



Slika 3.11 *Spuštanje u prostoru težina. (a) mala brzina obučavanja; (b) velika brzina obučavanja (postoje oscilacije); i (c) velika brzina obučavanja i dodat momentum član*

3.6.4 Nedostaci propagacije greške unazad

Uprkos prvobitnom uspjehu obučavanja propagacijom greške unazad, pokazalo se da postoje određeni aspekti tog algoritma koji čine da se ne može garantovati univerzalna primjenljivost. Najproblematičniji je dugotrajnost samog procesa. Ovo može biti posljedica neoptimalnog izbora brzine obučavanja i momentum člana. Razvijeni su mnogi napredni algoritmi bazirani na propagaciji greške unazad, koji imaju neku optimalnu metodu adaptacije brzine obučavanja, što je pokazano u nastavku. Većina direktnih neuspjeha u obučavanju pak dolazi iz jednog od dva izvora: paraliza mreže i zaglavljivanje u lokalnom minimumu.

Kako se mreža obučava, težine veza mogu da dostignu jako velike vrijednosti, pa samim tim ukupni ulaz skrivenih ili izlaznih neurona može dostići jako velike vrijednosti (po apsolutnoj vrijednosti), što će za posljedicu imati to da mreža sa sigmoidalnim aktivacionim funkcijama daje izlaz jako blizu 0 ili jako blizu 1. Kako se pri obučavanju takve mreže koristi (3.43), δ faktor, a samim tim i korekcija težine, imaju zanemarivo male vrijednosti. Tim se dalje promjene težine praktično zaustavljaju i taj efekat se naziva *paraliza mreže.*

Hiperpovrš greške kompleksne mreže je puna tzv. brda i dolina. S obzirom da se koristi tehnika gradijentnog spuštanja po toj hiperpovrši, obučavanje može da zaglavi u nekom od lokalnih minimuma, iako postoji neka mnogo dublja dolina u okolini. Probabilistički metodi mogu pomoći u izbjegavanju ovakvih zamki, ali oni imaju tendenciju da budu spori. Druga mogućnost je povećanje broja skrivenih neurona. Ovo povećava dimenzionalnost prostora greške, smanjuje vjerovatnoću zaglavljivanja, ali ne postoji jasna formula za izbor tog broja, mada se čini da postoji neka gornja granica broja skrivenih neurona, čijim prelaskom sistem ponovo povećava mogućnost upadanja u lokalni minimum.

3.6.5 Napredni algoritmi

Mnogi istraživači su razvili poboljšanja i proširenja osnovnog algoritma propagacije greške unazad. Ne postoje jasni dokazi da su te tehnike spremne da istisnu osnovni algoritam iz upotrebe, ali je pregled nekih od njih dat zbog njihovog potencijala.

Možda je najočiglednije poboljšanje zamjena prilično primitivnog metoda spuštanja po najvećoj strmini metodom minimizacije skupa putanja, poput minimizacije uparenih gradijenata [111] (eng. *conjugate gradient minimisation*). Minimizacija duž putanje $\mathbf{u}$ dovodi funkciju f do mjesta gdje joj je gradijent okomit na tu putanju. Umjesto praćenja gradijenta u svakom koraku, skup od n putanja se konstruiše tako da sve budu međusobno povezane i tako da minimizacija duž jedne od tih putanja $\mathbf{u}_j$ ne pokvari minimizaciju duž neke od ranijih putanja $\mathbf{u}_i$. Na ovaj način jedna minimizacija duž putanje $\mathbf{u}_i$ je dovoljna, pa da n minimizacija u sistemu sa n stepeni slobode dovede sistem do minimuma (pod uslovom da je kvadratičan).

Neka je funkcija koja se minimizuje aproksimirana razvojem u Tejlorov red

$$\begin{aligned} f(\boldsymbol{x}) &= f(\boldsymbol{p}) + \sum_i \left.\frac{\partial f}{\partial x_i}\right|_{\boldsymbol{p}} x_i + \frac{1}{2}\sum_{i,j} \left.\frac{\partial^2 f}{\partial x_i \partial x_j}\right|_{\boldsymbol{p}} x_i x_j + \cdots \\ &\approx \frac{1}{2}\boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^{\mathsf{T}}\boldsymbol{x} + c, \end{aligned} \tag{3.45}$$

gdje je $\boldsymbol{A}$ Hesijan funkcije f u $\boldsymbol{p}$, tj. $[\boldsymbol{A}]_{ij} = \left.\frac{\partial^2 f}{\partial x_i \partial x_j}\right|_{\boldsymbol{p}}$, i to je pozitivno određena simetrična matrica; $\boldsymbol{b}$ je negativan gradijent funkcije f u $\boldsymbol{p}$, tj. $\boldsymbol{b} = -\left.\nabla f\right|_{\boldsymbol{p}}$ i $c = f(\boldsymbol{p})$. Promjena $\boldsymbol{x}$ dovodi do promjene gradijenta

$$\delta\left(\nabla f\right) = \boldsymbol{A}\left(\delta \boldsymbol{x}\right). \tag{3.46}$$

Neka je f minimizovana duž putanje $\boldsymbol{u}_i$ do tačke kad je gradijent $-\boldsymbol{g}_{i+1}$ funkcije f okomit na $\boldsymbol{u}_i$, odnosno kad je

$$\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{g}_{i+1} = 0, \tag{3.47}$$

i da se traži nova putanja $\boldsymbol{u}_{i+1}$. Za garanciju da kretanje duž nove putanje neće narušiti minimizaciju duž prethodne putanje potrebno je da novi gradijent funkcije f ostane okomit na prethodnu putanju, odnosno

$$\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{g}_{i+2} = 0. \tag{3.48}$$

Oduzimanjem (3.48) od (3.47) se dobija

$$0 = \boldsymbol{u}_i^{\mathsf{T}}\left(\boldsymbol{g}_{i+1} - \boldsymbol{g}_{i+2}\right) = \boldsymbol{u}_i^{\mathsf{T}}\delta\left(\nabla f\right) = \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{A}\left(\delta \boldsymbol{u}_{i+1}\right). \tag{3.49}$$

Ako za dva vektora $\boldsymbol{u}_i$ i $\boldsymbol{u}_{i+1}$ važi (3.49), kaže se da su *upareni.*

Neka se polazi od početne tačke $\boldsymbol{p}_0$. Prva putanja minimizacije se uzima da je $\boldsymbol{u}_0 = -\boldsymbol{g}_0 = \nabla f\left(\boldsymbol{p}_0\right)$, čime se dolazi do nove tačke $\boldsymbol{p}_1$. Dalje putanje se računaju prema

$$\boldsymbol{u}_{i+1} = \boldsymbol{g}_{i+1} + \gamma_i \boldsymbol{u}_i, \tag{3.50}$$

pri čemu se γ_i bira tako da nova putanja bude uparena sa prethodnom, te da su uzastopni gradijenti okomiti, npr.

$$\gamma_i = \frac{\boldsymbol{g}_{i+1}^{\mathsf{T}}\boldsymbol{g}_{i+1}}{\boldsymbol{g}_i^{\mathsf{T}}\boldsymbol{g}_i}. \tag{3.51}$$

Dalje se računa nova tačka $\boldsymbol{p}_{i+2} = \boldsymbol{p}_{i+1} + \lambda_{i+1}\boldsymbol{u}_i$, gdje se λ_{i+1} bira tako da se minimizuje $f\left(\boldsymbol{p}_{i+2}\right)$. Ovo nije trivijalan problem [112], ali se pokazuje da postoji metoda minimizacije sa super-linearnom konvergencijom[3].

Za kvadratične sisteme sa n stepeni slobode je potrebno n iteracija, ali se ne minimizuju uvijek kvadratični sistemi, a prisutna je i greška zaokruživanja, tako da se n putanja mora preći nekoliko puta, ali i za to postoje poboljšanja [113]. Glavna posljedica je da algoritam ima kompleksnost $O(n)$, što je značajno bolje nego kod linearne konvergencije metode najstrmijeg spuštanja.

Neka od poboljšanja algoritma propagacije greške unazad su bazirana na nezavisno podesivim brzinama obučavanja za svaku od težina.

U [114] je pokazano da je, za mreže bez povratnih veza i skrivenih slojeva, za inkrementalnu proceduru za traženje optimalnih vrijednosti matrice W težine veza potrebna promjena težina u skladu sa

$$\Delta W(t+1) = \gamma(t+1)\left(\boldsymbol{d}(t+1) - W(t)\boldsymbol{x}(t+1)\right)\boldsymbol{x}(t+1), \tag{3.52}$$

gdje γ nije konstantna, već promjenljiva $(N_i + 1) \times (N_i + 1)$ matrica koja zavisi od ulaznog vektora.

U [115] su takođe pokazane prednosti korištenja nezavisnih brzina obučavanja za sve težine. Brzina obučavanja se mijenja nakon svakog uzorka prema

$$\gamma_{jk}(t+1) = \begin{cases} u\gamma_{jk}(t), & \text{ako su } \frac{\partial E(t+1)}{\partial w_{jk}} \text{ i } \frac{\partial E(t)}{\partial w_{jk}} \text{ istog znaka} \\ d\gamma_{jk}(t), & \text{inače} \end{cases} \tag{3.53}$$

gdje su u i d pozitivne konstante čije su vrijednosti malo iznad i ispod 1, respektivno. Ovo je zasnovano na ideji da se umanji brzina obučavanja u slučaju pojave oscilacija.

3.7 Rekurentne neuralne mreže

Iako dodavanjem ciklusa (povratnih veza) neuralna mreža ne dobija bolju mogućnost aproksimacije, moguće je da se smanji kompleksnost i veličina mreže za rješavanje istog problema koji je rješavan mrežom bez povratnih veza. Važno je pitanje šta je zapravo od interesa da rekurentna mreža nauči. Štaviše, čak nije ni jednoznačno određeno kako propagirati vrijednosti sa ulaza prema izlazima - do beskonačnosti ili do dostizanja stabilne tačke (atraktora). U nastavku će biti opisane mreže koje su bazirane na atraktorima, tj. propagira se signal sa ulaza do izlaza do dostizanja atraktora, a nakon toga se vrši adaptacija težina. Opisane su i rekurentne mreže kod kojih se pravilo obučavanja primjenjuje nakon svake propagacije (signal propagira preko svake veze jednom), pa se u takvim mrežama povratne veze mogu posmatrati kao dodatni ulazi mreže koji nisu zadati, već ih izračunava sama mreža.

[3] Za metod se kaže da konvergira linearno ako za grešku važi $E_{i+1} = cE_i$, uz $c < 1$. Ako važi $E_{i+1} = c\left(E_i\right)^m$, uz $m > 1$, konvergencija je super-linearna.

3.7.1 Uopšteno delta pravilo u rekurentnim mrežama

Algoritam propagacije greške unazad se lako može koristiti za obučavanje rekurentnih mreža. Prije razmatranja uopštenog slučaja, dat je pregled mreža kod kojih su izlazne vrijednosti povratnim vezama dovedene do dodatnog skupa ulaza (*Džordanova mreža*) [116] i mreže gdje su vrijednosti izlaza neurona skrivenog sloja povratnim vezama dovedene na dodatni skup ulaza (*Elmanova mreža*) [117].

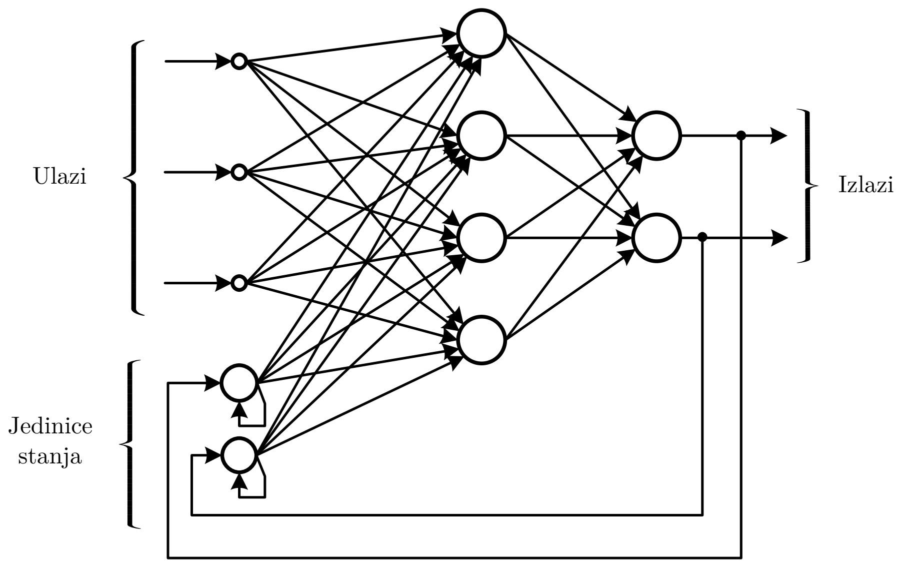
Tipična upotreba takvih mreža je data u nastavku. Neka je potrebno konstruisati mrežu koja treba da generiše upravljački signal u zavisnosti od vanjskog ulaza koji je dat u vidu vremenskog niza $\boldsymbol{x}(t), \boldsymbol{x}(t-1), \boldsymbol{x}(t-2), \ldots$ Sa mrežama bez povratnih veza postoje dva moguća pristupa za rješavanje tog problema:

1. konstruisati mrežu sa n ulaza koji predstavljaju posljednjih n vrijednosti pomenute serije, čime se postiže da je *vremenski prozor* ulaznog niza ulaz mreže;
2. konstruisati mrežu čiji su ulazi prvi, drugi, i viši izvodi signala iz vremenske serije, čije računanje i nije jednostavan zadatak.

Mana je, očigledno, ta što je dimenzionalnost ulaza pomnožena sa n, što dovodi do jako velike mreže, koja se sporo i teško obučava. Pomenute rekurentne mreže pružaju rješenje ovog problema. Zbog postojanja povratnih veza nije neophodno da vremenski prozor bude ulaz mreže, već mreža treba da nauči uticaj prethodnih vremenskih koraka.

3.7.1.1 Džordanova mreža

Jedna od najstarijih rekurentnih mreža je Džordanova mreža. Tipičan primjer topologije Džordanove mreže je dat na Slici 3.12.

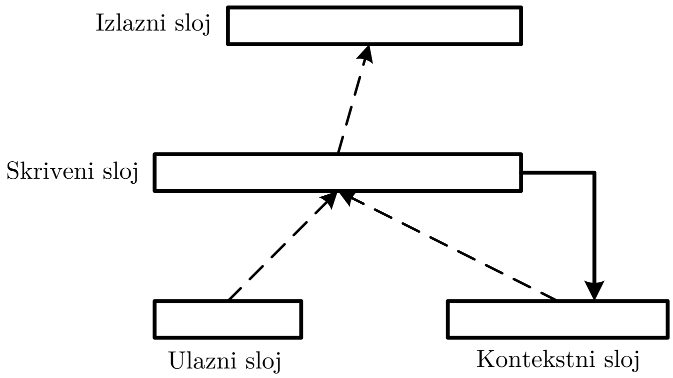


Slika 3.12 *Džordanova mreža*

Kod ove mreže vrijednosti aktivacionih funkcija neurona izlaznog sloja se povratnim vezama dovode do ulaznog sloja preko skupa dodatnih ulaznih neurona koji se nazivaju *jedinice stanja*. Postoji onoliko jedinica stanja koliko ima neurona u izlaznom sloju. Pomenute povratne veze imaju fiksnu težinu 1, a obučavanje se odnosi samo na promjene direktnih veza, tako da se sva pravila obučavanja višeslojnih mreža mogu primijeniti i za Džordanove mreže.

3.7.1.2 Elmanova mreža

Kod Elmanove mreže uvodi se skup *kontekstnih jedinica* koji predstavlja skup dodatnih ulaznih neurona koji dobijaju ulaz povratnim vezama od neurona skrivenog sloja. Topologija ovakve mreže prikazana je na Slici 3.13. Kao i kod Džordanove mreže, težine povratnih veza su fiksirane na 1.



Slika 3.13 *Elmanova mreža*

Obučavanje se obavlja na sljedeći način:

1. izlaz kontekstnih jedinica se postavlja na 0, iteracija $t = 1$;
2. na ulaz se dovodi uzorak $\boldsymbol{x}^t$, propagacija unaprijed se obavlja jednom;
3. primjenjuje se pravilo propagacije greške unazad;
4. $t \leftarrow t + 1$; skok na 2.

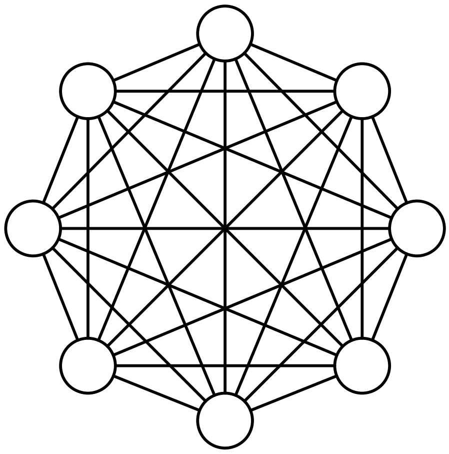
Na ovaj način je vrijednost izlaza kontekstnih jedinica u iteraciji t jednaka vrijednosti izlaza neurona skrivenog sloja u iteraciji $t - 1$.

3.7.2 Hopfildova mreža

Hopfildova mreža [118] se sastoji od skupa od N međusobno povezanih neurona, kao na Slici 3.14, za koje se izračunava vrijednost aktivacione funkcije asinhrono i nezavisno od drugih neurona. Svi neuroni su istovremeno i ulazni i izlazni. Aktivacione vrijednosti su binarne. Originalno su izabrane vrijednosti 0 i 1, ali, kako je opisano u nastavku, pokazalo se da postoje određene prednosti korištenja vrijednosti 1 i -1.

Stanje sistema je dato vektorom izlaza $\boldsymbol{y} = (y_k)$. Ukupan ulaz $s_k(t+1)$ k-tog neurona u iteraciji $t + 1$ je težinska suma

$$s_k(t+1) = \sum_{j \neq k} y_j(t) w_{jk} + \theta_k. \tag{3.54}$$



Slika 3.14 *Hopfildova mreža*

Jednostavna funkcija praga se koristi kao aktivaciona funkcija svakog neurona

$$y_k(t+1) = \begin{cases} 1, & s_k(t+1) > U_k \\ -1, & s_k(t+1) < U_k \\ y_k(t+1), & s_k(t+1) = U_k \end{cases} \tag{3.55}$$

odnosno, $y_k(t+1) = \operatorname{sgn}(y_k(t+1) - U_k)$. Uobičajeno se uzima $U_k = 0$.

Za k-ti neuron Hopfildove mreže se kaže da je *stabilan* u iteraciji t ako je

$$y_k(t) = \operatorname{sgn}(s_k(t-1)). \tag{3.56}$$

Stanje mreže je stabilno ako su svi neuroni stabilni. Uzorak $\boldsymbol{x}^p$ je stabilan ako dovođenjem tog uzorka na ulaz mreže, mreža bude stabilna.

Ako se uvede dodatni uslov $w_{jk} = w_{kj}$, ponašanje sistema se može opisati energetskom funkcijom

$$\varepsilon = -\frac{1}{2}\sum_{j \neq k}\sum y_j y_k w_{jk} - \sum_k \theta_k y_k. \tag{3.57}$$

Teorema 3.2. *Hopfildova mreža za čije veze važi $w_{jk} = w_{kj}$ i sa aktivacionim funkcijama datim sa* (3.55) *ima stabilne granične tačke.*

Dokaz. Iz (3.57) se vidi da je data funkcija ograničena odozdo jer su y_k ograničeni odozdo, a w_{jk} i θ_k konstante. Takođe, ε je monotono opadajuća kad se desi promjena stanja, jer je

$$\Delta\varepsilon = -\Delta y_k \left(\sum_{j \neq k} y_j w_{jk} + \theta_k \right) \tag{3.58}$$

uvijek negativno. ■

Prednost korištenja $1/-1$ modela umjesto $1/0$ modela je u simetriji stanja mreže, pa kad je uzorak $\boldsymbol{x}$ stabilan, onda je i njegov inverz stabilan. Takođe, uzorak i njegov inverz imaju istu energiju.

3.7.2.1 Hopfildova mreža kao asocijativna memorija

Primarna primjena Hopfildove mreže je asocijativna memorija. U tom slučaju se težine veza između neurona moraju podesiti tako da stanja sistema koja odgovaraju uzorcima koji se pamte u mreži budu stabilna. Ova stanja se mogu vidjeti kao *padovi* u energetskom prostoru. Kad se na ulaz mreže dovede zašumljen ili nepotpun testni uzorak, mreža će nadoknaditi nekorektne ili nedostajuće podatke iterirajući prema stabilnom stanju koje je u nekom smislu *blizu* uzorka na ulazu.

Za pamćenje P uzoraka može se koristiti Hebovo pravilo

$$w_{jk} = \begin{cases} \sum\limits_{p=1}^{P} x_j^p x_k^p, & j \neq k \\ 0, & j = k \end{cases} \tag{3.59}$$

Međutim, čini se da mreža vrlo brzo ulazi u zasićenje i da oko $0.15N$ uzoraka može biti zapamćeno prije nego što greške pamćenja postanu prevelike.

Postoje dva problema sa čuvanjem prevelikog broja uzoraka:

1. sačuvani uzorci postaju nestabilni;
2. pojavljuju se lažna stabilna stanja koja ne odgovaraju stabilnim uzorcima.

Prvi problem se može riješiti sljedećim algoritmom [119]:

Neka je data početna vrijednost matrice težina veza $W = [w_{jk}]$. Za svaki uzorak $\boldsymbol{x}^p$ koji treba pohraniti i za svaki element x_k^p tog uzorka definiše se korekcija ϵ_k prema

$$\epsilon_k = \begin{cases} 0, & \text{ako je } y_k \text{ stabilno za uzorak } \boldsymbol{x}^p \text{ na ulazu} \\ 1, & \text{inače} \end{cases} \tag{3.60}$$

Težina w_{jk} se modifikuje za $\Delta w_{jk} = y_j y_k (\epsilon_j + \epsilon_k)$ ako je $j \neq k$. Proceduru treba ponavljati dok svi uzorci ne postanu stabilni.

U praksi se pokazuje da ovaj algoritam obično konvergira.

Drugi problem se rješava primjenom Hebovog pravila u obrnutom smjeru za lažno stabilno stanje, i to sa malom brzinom obučavanja [120].

3.8 Samoorganizujuće mreže

Ranije u ovom poglavlju opisane su razne mreže koje su obučavane da obave mapiranje $F : \mathbb{R}^n \to \mathbb{R}^m$ pružanjem mreži primjera tog mapiranja - $(\boldsymbol{x}^p, \boldsymbol{d}^p)$ gdje je $F(\boldsymbol{x}^p) = \boldsymbol{d}^p$. Međutim, postoje i problemi gdje podaci koji sadrže parove ulaza i željenih izlaza nisu dostupni, već su date samo informacije o skupu ulaznih uzoraka $\boldsymbol{x}^p$. U takvim situacijama informacije od interesa se moraju naći unutar samih uzoraka za obučavanje. Neki od primjera takvih problema su:

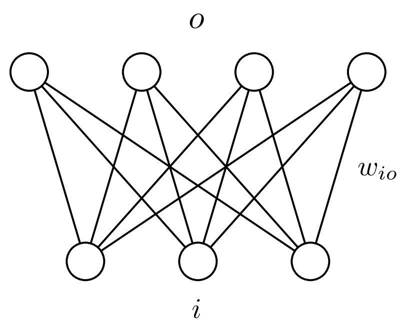
- *grupisanje (klasterizacija)*: ulazni podaci se grupišu u klastere, a obučavanje treba da nađe karakteristične klastere za te ulazne podatke. Izlaz sistema treba da bude oznaka (labela) klastera kojem pripada ulazni podatak (diskretan izlaz);

- *kvantizacija vektora*: kontinualan prostor treba da bude diskretizovan, a obučavanje treba da nađe optimalnu diskretizaciju ulaznog signala;
- *izdvajanje svojstava*: sistem treba da izdvoji određena svojstva iz ulaznog signala. Ovo je jedna varijanta problema *redukcije dimenzionalnosti.*

Kako je opisano, pri obučavanju ovih mreža ne postoji učitelj, tj. radi se o nenadgledanom obučavanju. Jedan od osnovnih oblika takvog obučavanja je *kompetitivno učenje* [121]. Mreža sa sličnim pristupom, ali sa drugačijim neposrednim osobinama je Kohonenova mreža [122]. Među ostalim samoorganizujućim mrežama ističe se ART [123].

3.8.1 Kompetitivno učenje

Kompetitivno učenje je procedura obučavanja koja dijeli skup ulaznih uzoraka u grupe koje su karakteristične za ulazne podatke. Mreži koja se obučava na ovaj način se daju samo ulazni podaci. Primjer jednostavne mreže za kompetitivno učenje je dat na Slici 3.15. Svi izlazni neuroni o su povezani sa svim ulaznim neuronima i vezama čije su težine w_{io}. Kad se na ulaz dovede uzorak $\boldsymbol{x}$ samo jedan od izlaza (*pobjednik*) biva aktiviran. Kod pravilno obučene mreže svi $\boldsymbol{x}$ iz jednog klastera će imati istog pobjednika. Postoje dva metoda određivanja pobjednika i odgovarajuća pravila obučavanja.



Slika 3.15 *Mreža za kompetitivno učenje*

3.8.1.1 Izbor pobjednika: skalarni proizvod

Za početak, pretpostavka je da su i ulazni vektor $\boldsymbol{x}$ i vektori težina $\boldsymbol{w}_o$ normalizovani na jediničnu dužinu. Svaki izlaz o se računa kao skalarni proizvod vektora težina i ulaza

$$y_o = \sum_i w_{io} x_i = \boldsymbol{w}_o^{\mathrm{T}} \boldsymbol{x}. \tag{3.61}$$

Zatim se bira neuron sa maksimalnom aktivacijom. Neka je to k-ti neuron. Tada se na izlaz tog neurona postavlja 1, a na ostale izlaze 0, odnosno $y_k = 1$ i $y_{o \neq k} = 0$. Ovo je kompetitivni aspekt mreže i zato se za izlazni sloj kaže da je tipa *pobjednik-uzima-sve* (eng. *winner-takes-all*). Ovo se najčešće implementira softverski, ali postoje i varijante mreže koja sama bira pobjednika, npr. MAXNET [124].

Nakon što je pobjednik k izabran, težine se koriguju u skladu sa

$$\boldsymbol{w}_k(t+1) = \frac{\boldsymbol{w}_k(t) + \gamma\left(\boldsymbol{x}(t) - \boldsymbol{w}_k(t)\right)}{\left\|\boldsymbol{w}_k(t) + \gamma\left(\boldsymbol{x}(t) - \boldsymbol{w}_k(t)\right)\right\|}, \tag{3.62}$$

gdje imenilac u razlomku osigurava da svi težinski vektori budu normalizovani. Takođe, treba primijetiti da se samo težine veza pobjednika koriguju.

Korekcija težina opisana sa (3.62) zapravo predstavlja rotaciju vektora težina $\boldsymbol{w}_k$ prema ulaznom vektoru $\boldsymbol{x}$. Svaki put kad se neki vektor dovede na ulaz mreže, vektor težina najbliži tom vektoru se bira i dodatno rotira prema tom ulazu. Posljedica je da se vektori težina rotiraju prema oblastima gdje se nalazi mnogo ulaznih vektora - klasterima ulaza.

3.8.1.2 Izbor pobjednika: Euklidova udaljenost

U slučaju kad ulazni vektori i vektori težina nisu normalizovani, prethodno opisana procedura nije primjenljiva. Za taj slučaj, pobjednički neuron k se bira kad je njegov vektor težina $\boldsymbol{w}_k$ najbliži ulaznom vektoru $\boldsymbol{x}$ razmatrajući Euklidovu udaljenost između njih.

$$k : \|\boldsymbol{w}_k - \boldsymbol{x}\| \leq \|\boldsymbol{w}_o - \boldsymbol{x}\|, \forall o. \tag{3.63}$$

Kad su vektori normalizovani, ovo se svodi na prethodni slučaj.

Umjesto rotacije vektora težina prema ulazu, kako je rađeno u (3.62), u ovom slučaju promjena vektora težina se vrši tako da dođe do pomijeranja prema ulaznom vektoru

$$\boldsymbol{w}_k(t+1) = \boldsymbol{w}_k(t) + \gamma\left(\boldsymbol{x}(t) - \boldsymbol{w}_k(t)\right). \tag{3.64}$$

I ovdje se koriguju samo težine veza prema neuronu pobjedniku.

Treba naglasiti da je za obje opisane varijante izbora pobjednika važna inicijalizacija težina. Nije teško zamisliti da će se potpuno nasumičnom inicijalizacijom težina, naročito u visoko-dimenzionalnom prostoru, pojaviti vektori koji nikad neće biti izabrani kao pobjednici, pa samim tim se nikad neće ni koristiti, ni pomijerati. Zato je uobičajeno da se vektori težina inicijalizuju upravo vektorima ulaza odabranih nasumično iz obučavajućeg skupa.

3.8.2 Kohonenova mreža

Kohonenova mreža se može opisati kao proširenje mreža koje se obučavaju kompetitivnim učenjem, iako bi to hronološki bilo netačno. Pored toga, Kohonenova mreža ima drugačije primjene.

Kod Kohonenove mreže neuroni izlaznog sloja su poredani u nekom smislu, često u dvodimenzionalnu mrežu ili niz, mada ovo zavisi od primjene. Poredak, koji je najčešće odabran od strane korisnika, ali postoje i varijante kod kojih se to radi automatski [125, 126], određuje koji izlazni neuroni su susjedni.

Kako se obučavajući uzorci dovode na ulaze mreže, težine veza do izlaznih neurona se podešavaju tako da se poredak koji postoji u ulaznom prostoru zadrži u izlaznom. Ovo znači da se obučavajući uzorci koji su blizu jedni drugima po nekom kriterijumu udaljenosti koji se koristi za određivanje pobjednika moraju mapirati na izlazne neurone koji su takođe blizu jedni drugima, odnosno susjedne neurone. Odatle se vidi da ako su ulazi uniformno raspoređeni u $\mathbb{R}^N$ i ako poredak mora biti zadržan, dimenzija izlaznog prostora mora biti bar N. Za mapiranje, koje predstavlja diskretizaciju ulaznog prostora, se kaže da *održava topologiju*.

Uobičajeno je da su obučavajući uzorci izabrani slučajno iz $\mathbb{R}^N$. U iteraciji t, uzorak $\boldsymbol{x}(t)$ se generiše i dovodi na ulaz mreže. Kao i u prethodnom potpoglavlju, bira se neuron pobjednik k. Za razliku od ranije, podešavaju se težine neurona pobjednika, ali i njegovih susjeda, prema pravilu

$$\boldsymbol{w}_o(t+1) = \boldsymbol{w}_o(t) + \gamma g(o,k)\left(\boldsymbol{x}(t) - \boldsymbol{w}_o(t)\right), \forall o. \tag{3.65}$$

Ovdje je $g(o,k)$ opadajuća funkcija udaljenosti neurona o i k takva da je $g(k,k) = 1$. Npr. može se koristiti Gausova funkcija $g(o,k) = e^{-(o-k)^2}$. Zbog ovakve kolektivne obučavajuće šeme, ulazni signali koji su bliski će se mapirati na susjedne neurone. Na taj način će se topologija koja postoji unutar ulaznih signala održati u mapiranju.

Ako se dimenzionalnost izlaza postavi na manje od N, neuroni mreže *savijaju* ulazni prostor.

Sposobnost održavanja topologije kod ovakvih mreža ima mnoge paralele u mozgu. Mozak je na mnogim mjestim organizovan tako da su senzorna okruženja predstavljena u obliku dvodimenzionalnih mapa. Npr. u vizuelnom sistemu postoji nekoliko topografskih mapiranja vidljivog prostora na površinu vizuelnog korteksa. Postoje i organizovana mapiranja površine tijela na korteks i u motornim i somatosenzornim područjima, te tonotopska mapiranja frekvencije u auditornom korteksu. Korištenje topografskih reprezentacija, gdje su neki važni aspekti senzora povezani sa fizičkom lokacijom ćelije na površini, su tako česti da to očigledno ima važnu funkciju u obradi informacija.

3.8.3 Adaptivna rezonantna teorija

Slijedi opis mreže koja se obučava bez nadgledanja, ali je, za razliku od prethodnih primjera, rekurentna.

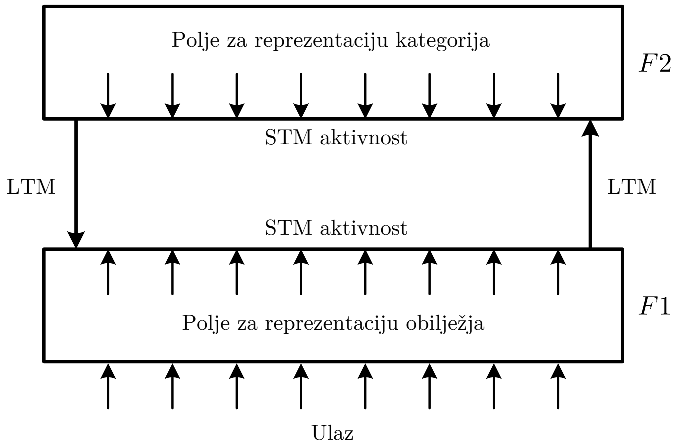
U [127] je predstavljen model za objašnjavanje određenih bioloških fenomena. Taj model ima tri krucijalne osobine:

1. *normalizacija* cjelokupne aktivnosti mreže. Biološki sistemi su najčešće vrlo prilagodljivi velikim promjenama u svom okruženju. Npr. ljudsko oko može da se prilagodi velikim varijacijama u intenzitetu svjetlosti;

2. *pojačanje kontrasta* u ulaznim uzorcima. Biti svjestan suptilnih razlika u ulaznim uzorcima može da znači mnogo u preživljavanju. Npr. razlikovanje predatora koji se sakriva od onog koji se odmara je vrlo značajno;

3. *kratkoročno pamćenje* kontrastno-pojačanih uzoraka. Prije nego što se ulazni uzorak prepozna, mora biti pohranjen u kratkoročnu memoriju (eng. *short-term memory - STM*). Dugoročno pamćenje (eng. *long-term memory - LTM*) je zaduženo za mehanizam odziva (npr. klasifikaciju), dok se STM koristi za postepene promjene u LTM.

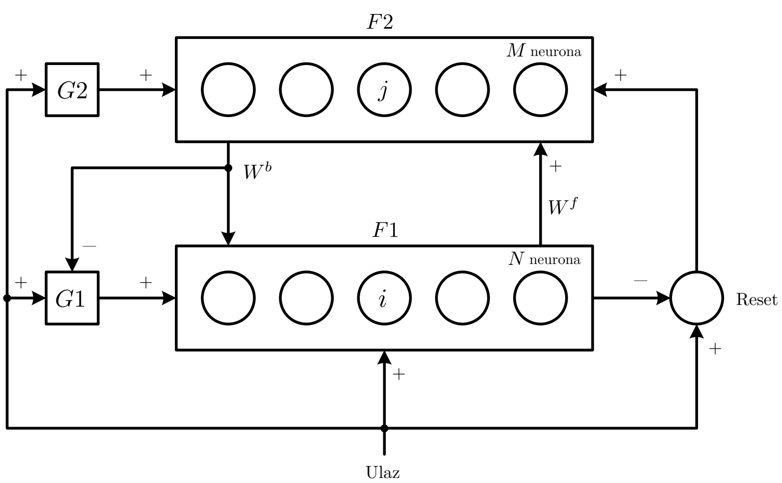
Sistem se sastoji od 2 sloja, $F1$ i $F2$, koji su međusobno povezani preko LTM, kao na Slici 3.16. Ulazni uzorak se prima preko $F1$, dok se klasifikacija odvija u $F2$. Kako je rečeno ranije, uzorak se ne klasifikuje direktno. Prvo se dešava karakterizacija ekstrakcijom obilježja, čime se aktivira polje za reprezentaciju obilježja. Očekivanja, sadržana u LTM vezama, prevode ulazni uzorak u kategorizaciju u polju za reprezentaciju kategorija. Klasifikacija se poredi sa očekivanjem mreže, pa ako postoji poklapanje, očekivanja se pojačavaju, inače se klasifikacija odbacuje.



Slika 3.16 *ART arhitektura*

3.8.3.1 ART1: Pojednostavljeni model

Pojednostavljeni model ART1 se sastoji od dva sloja binarnih neurona (sa izlaznim vrijednostima 0 i 1) koji se nazivaju sloj poređenja ($F1$) i sloj prepoznavanja ($F2$), kao na Slici 3.17. Svaki neuron iz $F1$ je povezan sa svim neuronima iz $F2$ preko direktne dugoročne memorije W^f, i obratno, preko povratne dugoročne memorije W^b. Ostali moduli su pojačanja $G1$ i $G2$, te modul za resetovanje.



Slika 3.17 *ART1 neuralna mreža*

Svaki neuron u modulu za poređenje ima tri ulaza: komponenta ulaznog uzorka, komponenta povratnog uzorka i pojačanje $G1$. Neuron daje izlaz 1 ako i samo ako su bar dva od ova tri ulaza na visokom nivou (pravilo dvije trećine).

Neuroni u sloju za prepoznavanje izračunavaju skalarni proizvod svojih težina veza i uzorka poslatog preko tih veza. Pobjednički neuron onda inhibira ostale neurone preko lateralne inhibicije.

Pojačanje $G2$ je logičko 'ili' svih elemenata ulaznog uzorka $\boldsymbol{x}$.

Pojačanje $G1$ je jednako pojačanju $G2$, osim kad povratni uzorak iz $F2$ sadrži bar jedno 1, tada se postavlja na 0.

Konačno, signal za resetovanje se šalje aktivnim neuronima u $F2$ ako se ulazni vektor $\boldsymbol{x}$ i izlaz $F1$ razlikuju za više od nekog nivoa, nazvanog nivo budnosti (eng. *vigilance*).

Sistem radi u sljedećim koracima:

1. inicijalizacija:
$$\begin{aligned} w_{ji}^b(0) &= 1 \\ w_{ji}^f(0) &= \tfrac{1}{1+N} \end{aligned}$$
gdje je N broj neurona u $F1$, M je broj neurona u $F2$, $0 \leq i < N$, $0 \leq j < M$. Takođe, bira se prag budnosti ρ takav da je $0 \leq \rho \leq 1$;

2. na ulaz se dovodi novi uzorak $\boldsymbol{x}$;

3. izračunavaju se aktivacione vrijednosti $\boldsymbol{y}'$ neurona u $F2$
$$y_i' = \sum_{j=1}^{N} w_{ij}^f(t) x_i; \tag{3.66}$$

4. bira se pobjednički neuron k ($0 \leq k < M$);

5. provjerava se budnost. Ako je
$$\frac{\boldsymbol{w}_k^b(t) \cdot \boldsymbol{x}}{\boldsymbol{x} \cdot \boldsymbol{x}} > \rho, \tag{3.67}$$
gdje je $\cdot$ skalarni proizvod, ide se na korak 7, inače se ide na korak 6. Treba naglasiti da je $\boldsymbol{w}_k^b(t) \cdot \boldsymbol{x}$ zapravo $\boldsymbol{x}^* \cdot \boldsymbol{x}$, što će biti veliko ako su $\boldsymbol{x}^*$ i $\boldsymbol{x}$ bliski;

6. neuronu k se onemogućuje dalja aktivnost i ide se na korak 3;

7. vrši se korekcija težina ($0 \leq l < N$)
$$\begin{aligned} w_{kl}^b(t+1) &= w_{kl}^b(t) x_l \\ w_{kl}^f(t+1) &= \frac{w_{kl}^f(t) x_l}{\frac{1}{2} + \sum_{i=1}^{N} w_{ki}^f(t) x_i} \end{aligned}$$

8. omogućava se rad svih neurona u $F2$ i ide se na korak 2.

3.8.3.2 ART1: Originalni model

Originalna ART1 mreža ima ugrađen algoritam za klasterizaciju na principu *praćenja vođe* (eng. *follow-the-leader clustering algorithm*) [128]. Ovaj algoritam pokušava da uklopi svaki novi ulazni uzorak u neku od postojećih klasa. Ako ne postoji poklapanje, npr. udaljenost između novog uzorka i svake od postojećih klasa je veća od nekog praga, kreira se nova klasa koja sadrži dati uzorak. Novitet ovakvog pristupa je to da se mreža prilagođava novim uzorcima bez narušavanja prethodne memorije. U većini neuralnih mreža, poput onih obučenih algoritmom propagacije greške unazad, svi uzorci se uče sekvencijalno, pa učenje novog uzorka može da pokvari težine veza za sve prethodno naučene uzorke. Mijenjanjem strukture mreže umjesto težina veza, ART1 prevazilazi ovaj problem.

Svaki neuron k, bilo u $F1$, bilo u $F2$, prima ulaz s_k i generiše izlaz y_k.

Za uvođenje normalizacije u model, uzima se $I = \sum s_k$, na osnovu čega se uvodi relativna vrijednost ulaza $\Theta_k = s_k I^{-1}$.

Model je takav da promjena odziva y_k:

- zavisi inhibitorno od svih ostalih ulaza i osjetljivosti samog neurona, odnosno okolina svakog neurona ima negativan uticaj na taj neuron $-y_k \sum\limits_{l \neq k} s_l$;
- ima uticaj pojačanja na ulaz neurona Bs_k;
- ima uticaj slabljenja za normalizaciju $-y_k s_k$;
- ima *raspad* Ay_k.

A i B su konstante. Diferencijalna jednačina za neurone oba sloja je

$$\frac{dy_k}{dt} = -Ay_k + (B - y_k)\, s_k - y_k \sum_{l \neq k} s_l, \tag{3.68}$$

gdje je $0 \leq y_k(0) \leq B$, jer inhibitorni efekat ulaza nikad ne može biti veći od pojačanja.

U ravnotežnom stanju, kad je $\frac{dy_k}{dt} = 0$, dobija se

$$y_k(A + I) = Bs_k. \tag{3.69}$$

Uvođenjem relativne vrijednosti ulaza u (3.69), dobija se

$$y_k = \Theta_k \frac{BI}{(A + I)}. \tag{3.70}$$

S obzirom na uslov poslije (3.68), važi

$$\frac{BI}{(A + I)} \leq B, \tag{3.71}$$

pa ukupni izlaz nikad nije veći od B, odnosno izlaz je normalizovan.

Da bi $F2$ bolje reagovao na razlike u izlaznim vrijednostima neurona iz $F1$ (ili obratno), primjenjuje se pojačanje kontrasta. Za te potrebe, (3.68) nije zadovoljavajuća. Za pojačanje kontrasta odbacuju se svi jednaki dijelovi u $F1$ i $F2$. Ovo je moguće uraditi dodavanjem još jednog inhibitorskog ulaza proporcionalnog ulazima ostalih neurona sa faktorom C

$$\frac{dy_k}{dt} = -Ay_k + (B - y_k)\, s_k - (y_k + C) \sum_{l \neq k} s_l, \tag{3.72}$$

Ako se postavi $B = (n-1)C$, gdje je n broj neurona, u ravnotežnom stanju se dobija

$$y_k = \frac{nCI}{(A+I)}\left(\Theta_k - \frac{1}{n}\right). \tag{3.73}$$

Sada, kad se zada takav ulaz da svi s_k budu jednaki, dobija se da su svi y_k jednaki 0, tj. efekat dodavanja C je pojačavanje razlika. Ako se postavi $B < (n-1)C$ doći će i do još većeg odbacivanja ulaza.

4. Vještačke neuralne mreže sa funkcijama radijalne baze

U ovom poglavlju dat je pregled pristupa rješavanju problema aproksimacije funkcije u visokodimenzionalnom prostoru pomoću neuralnih mreža. U skladu s tim, obučavanje ovakvih mreža je ekvivalentno pronalaženju hiperpovrši u visokodimenzinalnom prostoru koja najbolje odgovara obučavajućim podacima (najčešće datih u vidu tačaka), pri čemu se "najbolje odgovaranje" mjeri u nekom statističkom smislu. Analogno tome, generalizacija je ekvivalentna korištenju takve hiperpovrši za interpolaciju testnih podataka. Takav pristup je motivacija za rad sa funkcijama radijalne baze koje su pokazale dobre osobine u istraživanjima tradicionalne stroge interpolacije u visokodimenzionalnom prostoru. U kontekstu neuralnih mreža, skriveni neuroni sadrže skup funkcija koje čine proizvoljnu bazu za ulazne uzorke (vektore), i ove funkcije se zovu funkcije radijalne baze (eng. *Radial Basis Function - RBF*).

Konstrukcija mreže sa funkcijama radijalne baze, u svom osnovnom obliku, uključuje tri sloja sa potpuno različitim ulogama. Ulazni sloj je sačinjen od ulaznih jedinica koje povezuju mrežu sa njenom okolinom. Drugi sloj, jedini skriveni sloj mreže, služi za primjenu nelinearne transformacije iz ulaznog prostora u skriveni prostor, koji je u većini primjena visokodimenzionalan. Izlazni sloj je linearan, i odgovoran je za generisanje konačnog odziva mreže na uzorak doveden na ulaz mreže. Matematičko opravdanje za uvođenje nelinearne transformacije popraćene linearnom transformacijom može se naći u ranim radovima, poput [129], gdje je dato tvrđenje da je problem klasifikacije uzoraka prebačen u visokodimenzionalan prostor vjerovatnije linearno separabilan nego u niskodimenzionalnom prostoru. Druga važna stavka je činjenica da je dimenzionalnost skrivenog prostora direktno povezana sa kapacitetom mreže za aproksimaciju glatkog mapiranja ulaza na izlaz [130, 131] - što je veća dimenzionalnost, aproksimacija je preciznija.

4.1 Problem interpolacije sa funkcijama radijalne baze

Problem interpolacije se svodi na sljedeće: Neka je dat skup od N različitih tačaka $\{\boldsymbol{x}_i \in \mathbb{R}^m \mid i = 1, 2, \ldots, N\}$ i njima odgovarajući skup od N realnih brojeva $\{d_i \in \mathbb{R} \mid i = 1, 2, \ldots, N\}$. Potrebno je pronaći funkciju $F : \mathbb{R}^m \to \mathbb{R}$ koja ispunjava uslov interpolacije dat sa

$$F(\boldsymbol{x}_i) = d_i, \; i = 1, 2, \ldots, N. \tag{4.1}$$

Ovo je stroga interpolacija jer je zahtjev da interpolaciona površ (u ovom slučaju funkcija F) prolazi kroz sve tačke iz obučavajućeg skupa.

Tehnika zasnovana na korištenju funkcija radijalne baze se svodi na izbor funkcije F u obliku

$$F(\boldsymbol{x}) = \sum_{i=1}^{N} w_i \varphi\left(\|\boldsymbol{x} - \boldsymbol{x}_i\|\right), \tag{4.2}$$

gdje je $\{\varphi\left(\|\boldsymbol{x} - \boldsymbol{x}_i\|\right) \mid i = 1, 2, \ldots, N\}$ skup od N proizvoljnih (obično nelinearnih) funkcija, poznatih kao funkcije radijalne baze, a $\|\cdot\|$ norma vektora, najčešće Euklidova. Pomenute funkcije radijalnih baza su dobile ime u skladu sa tim što su to zapravo funkcije udaljenosti (radijusa) od određene tačke. Te tačke, u ovom slučaju tačke iz obučavajućeg skupa $\boldsymbol{x}_i \in \mathbb{R}^m \mid i = 1, 2, \ldots, N$, predstavljaju *centre* funkcija radijalnih baza, jer su to funkcije udaljenosti od baš tih tačaka.

Uvrštavanjem uslova interpolacije iz (4.1) u (4.2) dobija se sistem linearnih jednačina po nepoznatim koeficijentima (težinama) w_i

$$\begin{bmatrix} \varphi_{1,1} & \varphi_{1,2} & \cdots & \varphi_{1,N} \\ \varphi_{2,1} & \varphi_{2,2} & \cdots & \varphi_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{N,1} & \varphi_{N,2} & \cdots & \varphi_{N,N} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \tag{4.3}$$

gdje je

$$\varphi_{j,i} = \varphi\left(\|\boldsymbol{x}_j - \boldsymbol{x}_i\|\right) \mid i = 1, 2, \ldots, N; j = 1, 2, \ldots, N. \tag{4.4}$$

Neka je $\boldsymbol{d} = \begin{bmatrix} d_1 & d_2 & \cdots & d_N \end{bmatrix}^{\mathrm{T}}$ vektor željenih odziva (izlaza) i neka je $\boldsymbol{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_N \end{bmatrix}^{\mathrm{T}}$ vektor težina, gdje je N, kao i ranije, broj tačaka u obučavajućem skupu. Neka je $\boldsymbol{\Phi}$ $N \times N$ matrica čiji su elementi $\varphi_{j,i}$

$$\boldsymbol{\Phi} = \left[\varphi_{j,i}\right]_{N \times N}. \tag{4.5}$$

Matrica $\boldsymbol{\Phi}$ se često naziva i *matrica interpolacije.*

Jednačina (4.3) se može napisati i u matričnom obliku

$$\boldsymbol{\Phi w} = \boldsymbol{d}. \tag{4.6}$$

Pod pretpostavkom da je $\boldsymbol{\Phi}$ nesingularna matrica, te samim tim postoji i njen inverz $\boldsymbol{\Phi}^{-1}$, moguće je riješiti sistem iz (4.6).

$$\boldsymbol{w} = \boldsymbol{\Phi}^{-1}\boldsymbol{d} \tag{4.7}$$

Na pitanje koji je uslov potreban da bi matrica $\boldsymbol{\Phi}$ bila regularna, za veliki skup funkcija radijalne baze, odgovor daje sljedeća teorema [132], ovdje data bez dokaza.

Teorema 4.1. *Neka je dat skup od N različitih tačaka $\{\boldsymbol{x}_i \in \mathbb{R}^m \mid i = 1, 2, \ldots, N\}$. Onda je matrica interpolacije $\boldsymbol{\Phi}$, čiji su elementi $\varphi_{j,i} = \varphi\left(\|\boldsymbol{x}_j - \boldsymbol{x}_i\|\right)$, regularna.*

Funkcije koje su obuhvaćene ovom teoremom, a koje se često koriste u istraživanju RBF neuralnih mreža, su:

1. *Multikvadrična*: $\varphi(r) = \sqrt{r^2 + c^2}$, $c > 0$;
2. *Inverzna multikvadrična*: $\varphi(r) = \frac{1}{\sqrt{r^2+c^2}}$, $c > 0$;
3. *Gausova*: $\varphi(r) = e^{-\frac{r^2}{2\sigma^2}}$.

Inverzna multikvadrična i Gausova funkcija imaju zajedničke osobine: obje su lokalizovane u smislu da $\varphi(r) \to 0$ kad $r \to \infty$, i kod obje je interpolaciona matrica $\mathbf{\Phi}$ pozitivno određena. Ono što je vrlo zanimljivo je da funkcije radijalne baze koje rastu do beskonačnosti, poput multikvadrične, mogu da se koriste za aproksimaciju glatkog mapiranja ulaza na izlaz sa većom preciznošću od onih sa pozitivno određenom interpolacionom matricom.

4.2 Nadgledano učenje kao loše postavljen problem rekonstrukcije hiperpovrši

Prethodno opisana procedura stroge interpolacije često nije dobra strategija za obučavanje RBF mreža, naročito za određene klase zadataka, zbog loše generalizacije novih podataka iz sljedećeg razloga: Kada je broj podataka za obučavanje mnogo veći od broja stepeni slobode fizičkog procesa koji se aproksimira, a procedura nalaže korištenje onoliko funkcija radijalne baze koliko ima podataka u obučavajućem skupu, problem je preodređen. Posljedica je da mreža može da bude obučena za zavaravajuću količinu varijacija koje su možda posljedica neregularnosti ili šuma, a ne prirode same funkcije, što dalje dovodi do umanjenih sposobnosti generalizacije.

Da bi se dobro shvatio problem "prepristajanja"(eng. *overfitting*) i način kako ga riješiti, prvo je potrebno vratiti se razmišljanju da je dizajniranje neuralne mreže obučene za izvođenje izlaznog uzorka na osnovu ulaznog ekvivalentno učenju hiperpovrši (multidimenzionalno mapiranje) koja definiše izlaz preko ulaza. Drugim riječima, obučavanje se posmatra kao problem rekonstrukcije hiperpovrši na osnovu skupa podataka koji može biti prorijeđen.

Problem od interesa, prema terminologiji uvedenoj u primijenjenoj matematici, može biti dobro ili loše postavljen [133, 134]. Neka su domen $\mathbb{X}$ i kodomen $\mathbb{Y}$ metrički prostori koji su povezani fiksnim, ali nepoznatim preslikavanjem (mapiranjem) f. Problem rekonstrukcije preslikavanja f je dobro postavljen (eng. *well-posed*) ako su ispunjena tri uslova:

1. *Rješenje postoji.* Za svaki ulazni vektor $\boldsymbol{x} \in \mathbb{X}$ postoji izlaz $y = f(\boldsymbol{x})$ takav da je $y \in \mathbb{Y}$.
2. *Rješenje je jedinstveno.* Za svaki par ulaznih vektora $\boldsymbol{x}, \boldsymbol{t} \in \mathbb{X}$ važi $f(\boldsymbol{x}) = f(\boldsymbol{t})$ ako i samo ako je $\boldsymbol{x} = \boldsymbol{t}$.
3. *Ponašanje rješenja se mijenja kontinualno sa promjenom početnih uslova.* Preslikavanje je neprekidno, odnosno za svako $\varepsilon > 0$ postoji $\delta = \boldsymbol{\delta}(\varepsilon)$ takvo da iz $\rho_x(\boldsymbol{x}, \boldsymbol{t}) < \delta$ slijedi $\rho_y(f(\boldsymbol{x}), f(\boldsymbol{t})) < \varepsilon$, gdje je $\rho(\cdot, \cdot)$ udaljenost između argumenata u odgovarajućem prostoru. Ova osobina se često naziva i

stabilnost.

Ako bilo koji od ovih uslova nije ispunjen, za problem se kaže da je loše postavljen (eng. *ill-posed*). U osnovi, loše postavljen problem znači da vrlo veliki skupovi podataka mogu da sadrže iznenađujuće malo informacija o traženom rješenju.

Pored ocjene postavljenosti problema, uvodi se i pojam međusobne inverznosti dva problema, gdje se jedan naziva *direktnim*, a drugi *inverznim*, a formulacija jednog od njih zahtijeva (potpuno ili djelimično) poznavanje drugog.

U kontekstu problema interpolacije, može se reći da je fizički fenomen odgovoran za generisanje obučavajućih podataka (npr. govor, slika, signali sa raznih senzora i sl.) dobro postavljen direktan problem. Međutim, obučavanje na osnovu takvih fizičkih formi podataka, posmatrano kao problem rekonstrukcije hiperpovrši, je loše postavljen inverzan problem. Intuitivno je jasno da nijedan od tri uslova nije garantovano ispunjen, ponajviše zbog neizbježne pojave šuma, ali i zbog nedovoljne količine korisnih informacija u podacima za obučavanje.

Loše postavljeni problemi se mogu prevesti u dobro postavljene pomoću regularizacije, kako je opisano u nastavku.

4.3 Teorija regularizacije

U [135] predložen je novi metod za rješavanje loše postavljenih problema nazvan *regularizacija*. U kontekstu problema rekonstrukcije hiperpovrši, osnovna ideja regularizacije je stabilizacija rješenja pomoću nekog dodatnog nenegativnog funkcionala u koji su ugrađene ranije informacije o rješenju. Najčešći oblik tih informacija uključuje pretpostavku da je preslikavanje ulaza na izlaz *glatka* funkcija, u smislu da slični ulazi daju slične izlaze.

Neka je skup parova ulaz-izlaz (obučavajućih uzoraka) dostupnih za aproksimaciju označen sa:

$$\begin{array}{lll} \text{Ulazni signali:} & \boldsymbol{x}_i \in \mathbb{R}^m, & i = 1, 2, \ldots, N \\ \text{Izlazni signali:} & d_i \in \mathbb{R}, & i = 1, 2, \ldots, N \end{array}$$

Pretpostavka da je izlaz jednodimenzionalan ne ograničava opštu primjenljivost teorije regularizacije na bilo koji način.

Neka je aproksimirajuća funkcija označena sa $F(\boldsymbol{x})$, gdje je radi jednostavnosti prikaza izostavljen vektor težina mreže $\boldsymbol{w}$ iz liste argumenata.

Teorija regularizacije uključuje dva člana:

1. *Član standardne greške.* Ovaj član, označen sa $\mathscr{E}_s(F)$, je mjera standardne greške (udaljenosti) između željenog odziva d_i i stvarnog odziva y_i za sve obučavajuće uzorke ($i = 1, 2, \ldots, N$). Konkretno, definiše se sa

$$\mathscr{E}_s(F) = \frac{1}{2} \sum_{i=1}^{N} (d_i - y_i)^2 = \frac{1}{2} \sum_{i=1}^{N} [d_i - F(\boldsymbol{x}_i)]^2 . \tag{4.8}$$

Ovaj član se još naziva i *srednjekvadratna greška.*

2. *Regularizacioni član.* Ovaj član, označen sa $\mathscr{E}_c(F)$, zavisi od geometrijskih osobina aproksimacione funkcije $F(\boldsymbol{x})$. Konkretno, zapisuje se u obliku

$$\mathscr{E}_c(F) = \frac{1}{2} \|\mathbf{D}F\|^2 , \tag{4.9}$$

gdje je $\mathbf{D}$ *linearni diferencijalni operator*, a $\|\cdot\|$ neka norma. Prethodne informacije o obliku rješenja su ugrađene u taj operator što čini izbor operatora zavisnim od samog problema. Ovaj operator se takođe naziva i *stabilizator* jer stabilizuje rješenje problema regularizacije, čineći ga glatkim, čime se ispunjava treći uslov dobro postavljenih problema o neprekidnosti rješenja. Treba primijetiti da je glatka funkcija neprekidna, dok obratno nije obavezno tačno.

Analitički pristup za rješavanje situacije opisane sa (4.9) se bazira na konceptu *prostora funkcija*, koji se poziva na *normirane prostore funkcija*. U takvom prostoru sa mnogo dimenzija, vektori su neprekidne funkcije, čime se uvodi veza između matrica i linearnih diferencijabilnih operatora, pa se analiza linearnih sistema može primijeniti za analizu linearnih diferencijalnih jednačina.

U skladu s tim, $\|\cdot\|$ iz (4.9) predstavlja normu u prostoru funkcija kojem pripada $\mathbf{D}F(\boldsymbol{x})$. Uobičajeno je da se koristi prostor $\mathbb{L}_2$ koji se sastoji od svih funkcija realnog argumenta $f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^m$, takve da je $\|f(\boldsymbol{x})\|^2$ (L)-integrabilna [136].

Vrijednost koju treba minimizovati prema teoriji regularizacije je

$$\mathscr{E}(F) = \mathscr{E}_s(F) + \lambda \mathscr{E}_c(F) = \frac{1}{2} \sum_{i=1}^{N} \left[d_i - F\left(\boldsymbol{x}_i\right)\right]^2 + \frac{1}{2} \lambda \|\mathbf{D}F\|^2 , \tag{4.10}$$

gdje je λ pozitivan realan broj koji se naziva *regularizacioni parametar*. Funkcional $\mathscr{E}(F)$, koji se naziva *funkcional Tikonova*, mapira funkcije iz nekog odgovarajućeg prostora funkcija na realnu krivu. Minimizator funkcionala $\mathscr{E}(F)$, tj. rješenje regularizacionog problema se označava sa $F_\lambda(\boldsymbol{x})$.

Regularizacioni parametar λ se može posmatrati kao indikator da li je dati skup podataka dovoljan za određivanje rješenja $F_\lambda(\boldsymbol{x})$. Konkretno, granični slučaj $\lambda \to 0$ implicira da problem nije ograničen i da je rješenje $F_\lambda(\boldsymbol{x})$ potpuno određeno datim primjerima. Drugi granični slučaj $\lambda \to \infty$ implicira da je ograničenje da funkcija bude glatka nametnuto diferencijalnim operatorom $\mathbf{D}$ dovoljno da se odredi rješenje $F_\lambda(\boldsymbol{x})$, što dalje znači da su primjeri nepouzdani. Na taj način regularizacioni član $\mathscr{E}_c(F)$ predstavlja *funkciju penalizacije kompleksnosti modela* čiji se uticaj na konačno rješenje određuje regularizacionim parametrom λ.

4.3.1 Frešeov diferencijal funkcionala Tikonova

Suština regularizacije je pronalaženje funkcije $F_\lambda(\boldsymbol{x})$ koja minimizira funkcional Tikonova $\mathscr{E}(F)$ zadat sa (4.10). Za ostvarivanje minimizacije funkcionala $\mathscr{E}(F)$ potrebno je pravilo za određivanje diferencijala tog funkcionala. Ovo se može obaviti korištenjem *Frešeovog diferencijala* [137].

Iz elementarne matematičke analize je poznato da je tangenta na krivu prava linija koja daje najbolju aproksimaciju krive u okolini tačke dodira tangente sa krivom. Slično tome, Frešeov diferencijal funkcionala se može interpretirati kao najbolja lokalna linearna aproksimacija. Frešeov diferencijal funkcionala $\mathscr{E}(F)$ se definiše sa [138]

$$d\mathscr{E}(F,h) = \left[\frac{d}{d\beta}\mathscr{E}(F+\beta h)\right]_{\beta=0}, \tag{4.11}$$

gdje je $h(\boldsymbol{x})$ fiksna funkcija vektora $\boldsymbol{x}$.

Potreban uslov da funkcija $F(\boldsymbol{x})$ bude lokalni ekstrem funkcionala $\mathscr{E}(F)$ je taj da Frešeov diferencijal $d\mathscr{E}(F,h)$ mora da bude 0 u $F(\boldsymbol{x})$ za sve $(h \in \mathscr{H})$, odnosno

$$d\mathscr{E}(F,h) = d\mathscr{E}_s(F,h) + \lambda d\mathscr{E}_c(F,h) = 0, \tag{4.12}$$

gdje su $d\mathscr{E}_s(F,h)$ i $d\mathscr{E}_c(F,h)$ Frešeovi diferencijali funkcionala $\mathscr{E}_s(F)$ i $\mathscr{E}_c(F)$ respektivno.

Pri računanju Frešeovog diferencijala člana standardne greške $\mathscr{E}_s(F)$ iz (4.8), dobija se

$$\begin{aligned} d\mathscr{E}_s(F,h) &= \left[\tfrac{d}{d\beta}\mathscr{E}_s(F+\beta h)\right]_{\beta=0} \\ &= \left[\tfrac{1}{2}\tfrac{d}{d\beta}\sum_{i=1}^{N}\left[d_i - F\left(\boldsymbol{x}_i\right) - \beta h\left(\boldsymbol{x}_i\right)\right]^2\right]_{\beta=0} \\ &= -\sum_{i=1}^{N}\left[d_i - F\left(\boldsymbol{x}_i\right) - \beta h\left(\boldsymbol{x}_i\right)\right] h\left(\boldsymbol{x}_i\right)\Big|_{\beta=0} \\ &= -\sum_{i=1}^{N}\left[d_i - F\left(\boldsymbol{x}_i\right)\right] h\left(\boldsymbol{x}_i\right) \end{aligned} \tag{4.13}$$

Za dalje računanje Frešeovog diferencijala može da se iskoristi *Risova teorema reprezentacije* [139].

Teorema 4.2. *Neka je f ograničen linearan funkcional u* Hilbertovom prostoru *(zatvoren prostor u kom je definisan skalarni proizvod vektora) označenom sa $\mathscr{H}$. Postoji $h_0 \in \mathscr{H}$ takav da je*

$$f = (h, h_0)_{\mathscr{H}}, \text{ za sve } h \in \mathscr{H},$$

gdje je $(\cdot,\cdot)_{\mathscr{H}}$ skalarni proizvod dvije funkcije u prostoru $\mathscr{H}$. Pored toga, važi i

$$\|f\|_{\widetilde{\mathscr{H}}} = \|h_o\|_{\mathscr{H}},$$

gdje je $\widetilde{\mathscr{H}}$ konjugat Hilbertovog prostora $\mathscr{H}$.

U skladu sa Risovom teoremom reprezentacije, jednačina (4.13) se može zapisati kao

$$d\mathscr{E}_s(F,h) = -\left(h, \sum_{i=1}^{N}\left(d_i - F\left(x_i\right)\right)\delta_{\boldsymbol{x}_i}\right)_{\widetilde{\mathscr{H}}}, \tag{4.14}$$

gdje $\delta_{\boldsymbol{x}_i}$ označava *Dirakovu delta distribuciju* od $\boldsymbol{x}$ sa centrom u $\boldsymbol{x}_i$, odnosno

$$\delta_{\boldsymbol{x}_i} = \delta\left(\boldsymbol{x} - \boldsymbol{x}_i\right). \tag{4.15}$$

Isto se može primijeniti i na računanje Frešeovog diferencijala regularizacionog člana $\mathscr{E}_c(F)$ iz (4.9).

$$\begin{aligned} d\mathscr{E}_c(F,h) &= \tfrac{d}{d\beta}\mathscr{E}_c(F+\beta h)\Big|_{\beta=0} \\ &= \frac{1}{2}\frac{d}{d\beta}\int\limits_{\mathbb{R}^m} (\mathbf{D}[F+\beta h])^2\, d\boldsymbol{x}\Bigg|_{\beta=0} \\ &= \int\limits_{\mathbb{R}^m} \mathbf{D}[F+\beta h]\mathbf{D}h d\boldsymbol{x}\Bigg|_{\beta=0} \\ &= \int\limits_{\mathbb{R}^m} \mathbf{D}F\mathbf{D}h d\boldsymbol{x} \\ &= (\mathbf{D}h, \mathbf{D}F)_{\mathscr{H}}. \end{aligned} \tag{4.16}$$

4.3.2 Ojler-Lagranžova jednačina

Na osnovu lineranog diferencijalnog operatora $\mathbf{D}$ moguće je pronaći jednoznačno određen *vezani operator*, označen sa $\widetilde{\mathbf{D}}$, takav da za bilo koji par funkcija $u(\boldsymbol{x})$ i $v(\boldsymbol{x})$ koje su diferencijabilne na dovoljno velikom opsegu i koje ispunjavaju podobne ograničavajuće uslove, važi

$$\int\limits_{\mathbb{R}^m} u(\boldsymbol{x})\mathbf{D}v(\boldsymbol{x})d\boldsymbol{x} = \int\limits_{\mathbb{R}^m} v(\boldsymbol{x})\widetilde{\mathbf{D}}u(\boldsymbol{x})d\boldsymbol{x} \tag{4.17}$$

Jednačina (4.17) se naziva *Grinov identitet*. Ako se $\mathbf{D}$ posmatra kao matrica, vezani operator $\widetilde{\mathbf{D}}$ ima sličnu ulogu kao transponovana matrica.

Posmatranjem jednačina (4.16) i (4.17) moguće je postaviti sljedeće veze

$$\begin{aligned} u(\boldsymbol{x}) &= \mathbf{D}F(\boldsymbol{x}), \\ \mathbf{D}v(\boldsymbol{x}) &= \mathbf{D}h(\boldsymbol{x}), \end{aligned}$$

pa se Grinov identitet iz (4.17) može napisati u sljedećem obliku

$$d\mathscr{E}_c(F,h) = \int\limits_{\mathbb{R}^m} h(\boldsymbol{x})\widetilde{\mathbf{D}}\mathbf{D}F(\boldsymbol{x})d\boldsymbol{x} = \left(h, \widetilde{\mathbf{D}}\mathbf{D}F\right)_{\mathscr{H}}. \tag{4.18}$$

Povratkom na (4.12) i uvrštavanjem (4.14) i (4.18), dobija se novi izraz za Frešeov diferencijal

$$d\mathscr{E}(F,h) = \left(h, \left[\widetilde{\mathbf{D}}\mathbf{D}F - \frac{1}{\lambda}\sum_{i=1}^{N}\left(d_i - F\left(x_i\right)\right)\delta_{\boldsymbol{x}_i}\right]\right)_{\mathscr{H}}. \tag{4.19}$$

S obzirom da je regularizacioni parametar λ iz intervala $(0,\infty)$, Frešeov diferencijal je 0 za svako $h(\boldsymbol{x})$ u prostoru $\mathscr{H}$ ako i samo ako je ispunjen sljedeći uslov

$$\widetilde{\mathbf{D}}\mathbf{D}F_\lambda - \frac{1}{\lambda}\sum_{i=1}^{N}\left(d_i - F\left(x_i\right)\right)\delta_{\boldsymbol{x}_i} = 0, \tag{4.20}$$

odnosno

$$\widetilde{\mathbf{D}}\mathbf{D}F_\lambda(\boldsymbol{x}) = \frac{1}{\lambda}\sum_{i=1}^{N}\left(d_i - F\left(x_i\right)\right)\delta\left(\boldsymbol{x} - \boldsymbol{x}_i\right), \tag{4.21}$$

Jednačina (4.21) je *Ojler-Lagranžova jednačina* za funkcional Tikonova $d\mathscr{E}(F)$. Ona definiše potreban uslov da funkcional Tikonova $d\mathscr{E}(F)$ ima ekstrem u $F_\lambda(\boldsymbol{x})$ [140].

4.3.3 Grinova funkcija

Jednačina (4.21) predstavlja parcijalnu diferencijalnu jednačinu za aproksimaciju funkcije F. Za rješenje ove jednačine je poznato da sadrži integralnu transformaciju desne strane jednačine.

Neka $G\left(\boldsymbol{x}, \boldsymbol{\xi}\right)$ označava funkciju kod koje je vektor $\boldsymbol{x}$ parametar, a vektor $\boldsymbol{\xi}$ argument. Za dati linearni diferencijalni operator $\mathbf{L}$ zadaje se funkcija $G\left(\boldsymbol{x}, \boldsymbol{\xi}\right)$ koja ispunjava sljedeće uslove:

1. Za fiksno $\boldsymbol{\xi}$, $G\left(\boldsymbol{x}, \boldsymbol{\xi}\right)$ je funkcija od $\boldsymbol{x}$ koja ispunjava unapred definisane ograničavajuće uslove;
2. Osim u tački $\boldsymbol{x} = \boldsymbol{\xi}$, izvodi funkcije $G\left(\boldsymbol{x}, \boldsymbol{\xi}\right)$ po $\boldsymbol{x}$ su svi neprekidni, a broj tih izvoda je određen redom operatora $\mathbf{L}$;
3. Ako se $G\left(\boldsymbol{x}, \boldsymbol{\xi}\right)$ posmatra kao funkcija od $\boldsymbol{x}$, ona zadovoljava parcijalnu diferencijalnu jednačinu
$$\mathbf{L}G\left(\boldsymbol{x}, \boldsymbol{\xi}\right) = 0, \tag{4.22}$$
svugdje osim u tački $\boldsymbol{x} = \boldsymbol{\xi}$, gdje ima singularitet. Ovo znači da funkcija $G\left(\boldsymbol{x}, \boldsymbol{\xi}\right)$ zadovoljava parcijalnu diferencijalnu jednačinu
$$\mathbf{L}G\left(\boldsymbol{x}, \boldsymbol{\xi}\right) = \delta\left(\boldsymbol{x} - \boldsymbol{\xi}\right), \tag{4.23}$$
gdje je, kako je ranije definisano, $\delta\left(\boldsymbol{x} - \boldsymbol{\xi}\right)$ Dirakova delta funkcija pomjerena u tačku $\boldsymbol{x} = \boldsymbol{\xi}$.

Ovako definisana funkcija $G\left(\boldsymbol{x}, \boldsymbol{\xi}\right)$ se naziva *Grinova funkcija* za diferencijalni operator $\mathbf{L}$ [141]. Grinova funkcija za linearni diferencijalni operator ima sličnu ulogu kao inverzna matrica za matričnu jednačinu.

Neka je $\varphi(\boldsymbol{x})$ neprekidna ili dio-po-dio neprekidna funkcija vektora $\boldsymbol{x} \in \mathbb{R}^m$. Tada je funkcija

$$F(\boldsymbol{x}) = \int\limits_{\mathbb{R}^m} G\left(\boldsymbol{x}, \boldsymbol{\xi}\right)\varphi\left(\boldsymbol{\xi}\right)d\boldsymbol{\xi} \tag{4.24}$$

rješenje diferencijalne jednačine

$$\mathbf{L}F(\boldsymbol{x}) = \varphi(\boldsymbol{x}). \tag{4.25}$$

Naime, ako se na (4.24) primijeni linearni diferencijalni operator $\mathbf{L}$, dobija se

$$\mathbf{L}F(\boldsymbol{x}) = \mathbf{L}\int\limits_{\mathbb{R}^m} G\left(\boldsymbol{x}, \boldsymbol{\xi}\right)\varphi\left(\boldsymbol{\xi}\right)d\boldsymbol{\xi} = \int\limits_{\mathbb{R}^m} \mathbf{L}G\left(\boldsymbol{x}, \boldsymbol{\xi}\right)\varphi\left(\boldsymbol{\xi}\right)d\boldsymbol{\xi} \tag{4.26}$$

Diferencijalni operator vrši diferenciranje samo po $\boldsymbol{x}$, tj. tretira $\boldsymbol{\xi}$ kao konstantu, pa se na osnovu (4.23) dobija

$$\mathbf{L}F(\boldsymbol{x}) = \int\limits_{\mathbb{R}^m} \delta\left(\boldsymbol{x} - \boldsymbol{\xi}\right) \varphi\left(\boldsymbol{\xi}\right) d\boldsymbol{\xi} \tag{4.27}$$

Konačno, na osnovu osobine *prosijavanja* Dirakove delta funkcije, odnosno

$$\int\limits_{\mathbb{R}^m} \delta\left(\boldsymbol{x} - \boldsymbol{\xi}\right) \varphi\left(\boldsymbol{\xi}\right) d\boldsymbol{\xi} = \varphi(\boldsymbol{x}), \tag{4.28}$$

dobija se (4.24).

4.3.4 Rješenje problema regularizacije

Za rješenje Ojler-Lagranžove jednačine (4.21) postavlja se

$$\mathbf{L} = \widetilde{\mathbf{D}}\mathbf{D} \tag{4.29}$$

i

$$\varphi\left(\boldsymbol{\xi}\right) = \frac{1}{\lambda} \sum_{i=1}^{N} \left[d_i - F\left(x_i\right)\right] \delta\left(\boldsymbol{\xi} - \boldsymbol{x}_i\right). \tag{4.30}$$

Tada je, na osnovu (4.24)

$$\begin{aligned} F_\lambda(\boldsymbol{x}) &= \int\limits_{\mathbb{R}^m} G\left(\boldsymbol{x}, \boldsymbol{\xi}\right) \left\{ \frac{1}{\lambda} \sum_{i=1}^{N} \left[d_i - F\left(\boldsymbol{x}_i\right)\right] \delta\left(\boldsymbol{\xi} - \boldsymbol{x}_i\right) \right\} d\boldsymbol{\xi} \\ &= \frac{1}{\lambda} \sum_{i=1}^{N} \left[d_i - F\left(\boldsymbol{x}_i\right)\right] \int\limits_{\mathbb{R}^m} G\left(\boldsymbol{x}, \boldsymbol{\xi}\right) \delta\left(\boldsymbol{\xi} - \boldsymbol{x}_i\right) d\boldsymbol{\xi}. \end{aligned} \tag{4.31}$$

Konačno, korištenjem osobine *prosijavanja* Dirakove delta funkcije iz (4.28) dobija se rješenje Ojler-Lagranžove jednačine

$$F_\lambda(\boldsymbol{x}) = \frac{1}{\lambda} \sum_{i=1}^{N} \left[d_i - F\left(\boldsymbol{x}_i\right)\right] G\left(\boldsymbol{x}, \boldsymbol{\xi}\right). \tag{4.32}$$

Dobijena jednačina (4.32) pokazuje da je minimizaciono rješenje $F_\lambda(\boldsymbol{x})$ regularizacionog problema linearna superpozicija N Grinovih funkcija. Vektori $\boldsymbol{x}_i$ predstavljaju *centre širenja*, a težine $\frac{[d_i - F(\boldsymbol{x}_i)]}{\lambda}$ predstavljaju *koeficijente širenja*. Drugim riječima, rješenje regularizacionog problema leži u N-dimenzionalnom potprostoru prostora glatkih funkcija, a skup Grinovih funkcija $\{G\left(\boldsymbol{x}, \boldsymbol{\xi}\right)\}$ sa centrima u $\boldsymbol{x}_i$, $i = 1, 2, \ldots, N$ predstavlja bazu tog potprostora.

4.3.5 Određivanje koeficijenata širenja

Sljedeći problem koji treba riješiti je određivanje nepoznatih koeficijenata širenja. Neka je

$$w_i = \frac{1}{\lambda} \left[d_i - F\left(\boldsymbol{x}_i\right)\right], i = 1, 2, \ldots, N. \tag{4.33}$$

Tada se (4.32) može zapisati kao

$$F_\lambda(\boldsymbol{x}) = \sum_{i=1}^{N} w_i G(\boldsymbol{x}, \boldsymbol{\xi}) . \tag{4.34}$$

Izračunavanjem vrijednosti jednačine (4.34) za $\boldsymbol{x}_j, j = 1, 2, \ldots, N$, dobija se

$$F_\lambda(\boldsymbol{x}_j) = \sum_{i=1}^{N} w_i G(\boldsymbol{x}_j, \boldsymbol{\xi}) . \tag{4.35}$$

Ako se uvedu sljedeće oznake:

$$\boldsymbol{F}_\lambda(\boldsymbol{x}) = \begin{bmatrix} F_\lambda(\boldsymbol{x}_1) & F_\lambda(\boldsymbol{x}_2) & \cdots & F_\lambda(\boldsymbol{x}_N) \end{bmatrix}^{\mathrm{T}}, \tag{4.36}$$

$$\boldsymbol{d} = \begin{bmatrix} d_1 & d_2 & \cdots & d_N \end{bmatrix}^{\mathrm{T}}, \tag{4.37}$$

$$\mathbf{G} = \begin{bmatrix} G(\boldsymbol{x}_1, \boldsymbol{x}_1) & G(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & G(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ G(\boldsymbol{x}_2, \boldsymbol{x}_1) & G(\boldsymbol{x}_2, \boldsymbol{x}_2) & \cdots & G(\boldsymbol{x}_2, \boldsymbol{x}_N) \\ \vdots & \vdots & & \vdots \\ G(\boldsymbol{x}_N, \boldsymbol{x}_1) & G(\boldsymbol{x}_N, \boldsymbol{x}_2) & \cdots & G(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}, \tag{4.38}$$

$$\boldsymbol{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_N \end{bmatrix}^{\mathrm{T}}, \tag{4.39}$$

onda se jednačine (4.33) i (4.35) mogu zapisati u matričnoj formi kako slijedi, respektivno.

$$\boldsymbol{w} = \frac{1}{\lambda}(\boldsymbol{d} - \boldsymbol{F}_\lambda) \tag{4.40}$$

$$\boldsymbol{F}_\lambda = \mathbf{G}\boldsymbol{w} \tag{4.41}$$

Iz jednačina (4.40) i (4.41) se dobija

$$(\mathbf{G} + \lambda\mathbf{I})\, \boldsymbol{w} = \boldsymbol{d} \tag{4.42}$$

gdje je $\mathbf{I}$ jedinična $N \times N$ matrica. Matrica $\mathbf{G}$ se naziva Grinova matrica.

Diferencijalni operator $\boldsymbol{L}$ je *samovezani operator*, u smislu da je njegov vezani operator upravo on sam. Odatle slijedi da je pridružena Grinova funkcija $G(\boldsymbol{x}, \boldsymbol{x}_i)$ *simetrična* funkcija, odnosno da je

$$G(\boldsymbol{x}_i, \boldsymbol{x}_j) = G(\boldsymbol{x}_j, \boldsymbol{x}_i)\,, \text{za sve } i \text{ i } j \tag{4.43}$$

Jednačina (4.43) pokazuje da se pozicije dviju tačaka $\boldsymbol{x}$ i $\boldsymbol{\xi}$ mogu međusobno mijenjati bez uticaja na Grinovu funkciju. Analogno tome, Grinova matrica $\mathbf{G}$ je simetrična matrica, odnosno

$$\mathbf{G}^{\mathrm{T}} = \mathbf{G}. \tag{4.44}$$

U Potpoglavlju 4.1 je opisan problem interpolacije u kontekstu interpolacione matrice $\mathbf{\Phi}$. Može se primijetiti da Grinova matrica $\mathbf{G}$ u teoriji regularizacije igra sličnu ulogu kao interpolaciona matrica $\mathbf{\Phi}$ u teoriji interpolacije. Obje matrice su istih dimenzija. U skladu s tim, može se tvrditi da je matrica $\mathbf{G}$, za određene klase Grinovih funkcija, pozitivno određena ako su tačke $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ različite. Klase Grinovih funkcija za koje ovo važi, kako je navedeno ranije, su inverzna multikvadrična i Gausova funkcija. U praksi je uvijek moguće izabrati dovoljno veliko λ da se obezbijedi da je $\mathbf{G}+\lambda\mathbf{I}$ pozitivno određena, a samim tim i invertibilna. To povlači da sistem jednačina (4.42) ima jedinstveno rješenje

$$\boldsymbol{w} = (\mathbf{G} + \lambda\mathbf{I})^{-1}\, \boldsymbol{d}. \tag{4.45}$$

Kao zaključak može se reći da je regularizacija ekvivalentna širenju rješenja u skladu sa skupom Grinovih funkcija čija karakterizacija zavisi samo od oblika usvojenog za stabilizator $\mathbf{D}$ i pridružene ograničavajuće uslove, te da je broj korištenih Grinovih funkcija jednak broj uzoraka u obučavajućem skupu.

Treba naglasiti da ovakvo rješenje nije potpuno za sve klase Grinovih funkcija zbog postojanja funkcija iz *nula-prostora* operatora $\mathbf{D}$ (prostor svih funkcija koje operator $\mathbf{D}$ preslikava u 0) na koje član $\|\mathbf{D}F\|^2$ nema efekat. Međutim, ovo ne važi za Grinove funkcije sa oblikom zvona (eng. *bell-shaped*), poput Gausove ili inverzne multikvadrične.

S obzirom da karakterizacija Grinove funkcije zavisi samo od oblika stabilizatora $\mathbf{D}$, ako je izabrani stabilizator invarijantan na translaciju i na rotaciju, onda Grinova funkcija $G(\boldsymbol{x}, \boldsymbol{x}_i)$ zavisi samo od Euklidske norme razlike vektora $\boldsymbol{x}$ i $\boldsymbol{x}_i$, odnosno

$$G(\boldsymbol{x}, \boldsymbol{x}_i) = G(\|\boldsymbol{x} - \boldsymbol{x}_i\|). \tag{4.46}$$

Takva Grinova funkcija je zapravo funkcija radijalne baze, pa se rješenje problema regularizacije može zapisati kao

$$F_\lambda(\boldsymbol{x}) = \sum_{i=1}^{N} w_i G(\|\boldsymbol{x} - \boldsymbol{x}_i\|). \tag{4.47}$$

Jednačina (4.47) predstavlja rješenje stroge interpolacije, ali je, za razliku od (4.2), to rješenje regularizovano. Za regularizacioni parametar $\lambda = 0$ ta dva rješenja se svode na isto.

4.3.6 Gausove funkcije više promjenljivih

Primjer Grinove funkcije $G(\boldsymbol{x}, \boldsymbol{x}_i)$ čiji je linearni diferencijalni operator $\mathbf{D}$ invarijantan i na translaciju i na rotaciju je Gausova funkcija više promjenljivih definisana sa

$$G(\boldsymbol{x}, \boldsymbol{x}_i) = e^{-\frac{1}{2\sigma_i^2}\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}, \tag{4.48}$$

gdje $\boldsymbol{x}_i$ označava centar funkcije, a σ_i njenu širinu (varijansu). Samovezani operator $\mathbf{L} = \widetilde{\mathbf{D}}\mathbf{D}$ koji određuje Grinovu funkciju iz (4.48) je dat sa

$$\mathbf{L} = \sum_{n=1}^{\infty} (-1)^n \alpha_n \nabla^{2n}, \tag{4.49}$$

gdje je

$$\alpha_n = \frac{\sigma_i^{2n}}{n!2^n} \tag{4.50}$$

i

$$\nabla^a = \frac{\partial^a}{\partial x_1^a} + \frac{\partial^a}{\partial x_2^a} + \cdots + \frac{\partial^a}{\partial x_m^a}. \tag{4.51}$$

Jednačina (4.51) predstavlja *iterirani Laplasov operator.*

Zbog činjenice da suma u (4.49) ide do beskonačnosti, operator $\mathbf{L}$ se naziva *pseudo-diferencijalni operator.*

Iz $\mathbf{L} = \widetilde{\mathbf{D}}\mathbf{D}$ i (4.49) dobija se

$$\begin{aligned} \mathbf{D} &= \sum_n \sqrt{\alpha_n} \left(\tfrac{\partial}{\partial x_1} + \tfrac{\partial}{\partial x_2} + \cdots + \tfrac{\partial}{\partial x_m} \right)^n \\ &= \sum_{a+b+\cdots+k=n} \sqrt{\alpha_n} \tfrac{\partial^n}{\partial x_1^a \partial x_2^b \cdots \partial x_m^k} \end{aligned} \tag{4.52}$$

i

$$\begin{aligned} \widetilde{\mathbf{D}} &= \sum_n (-1)^n \sqrt{\alpha_n} \left(\tfrac{\partial}{\partial x_1} + \tfrac{\partial}{\partial x_2} + \cdots + \tfrac{\partial}{\partial x_m} \right)^n \\ &= \sum_{a+b+\cdots+k=n} (-1)^n \sqrt{\alpha_n} \tfrac{\partial^n}{\partial x_1^a \partial x_2^b \cdots \partial x_m^k}. \end{aligned} \tag{4.53}$$

Kad se prethodne jednačine uvrste u (4.47) dobija se

$$F_\lambda(\boldsymbol{x}) = \sum_{i=1}^{N} w_i e^{-\frac{1}{2\sigma_i^2}\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}. \tag{4.54}$$

U (4.54) svaki Gausov sabirak u sumi ima različitu širinu σ_i. Često se, radi pojednostavljenja, a uprkos određenim ograničenjima koja ipak ne utiču na osobinu univerzalnog aproksimatora, postavlja $\sigma_i = \sigma$ za sve i.

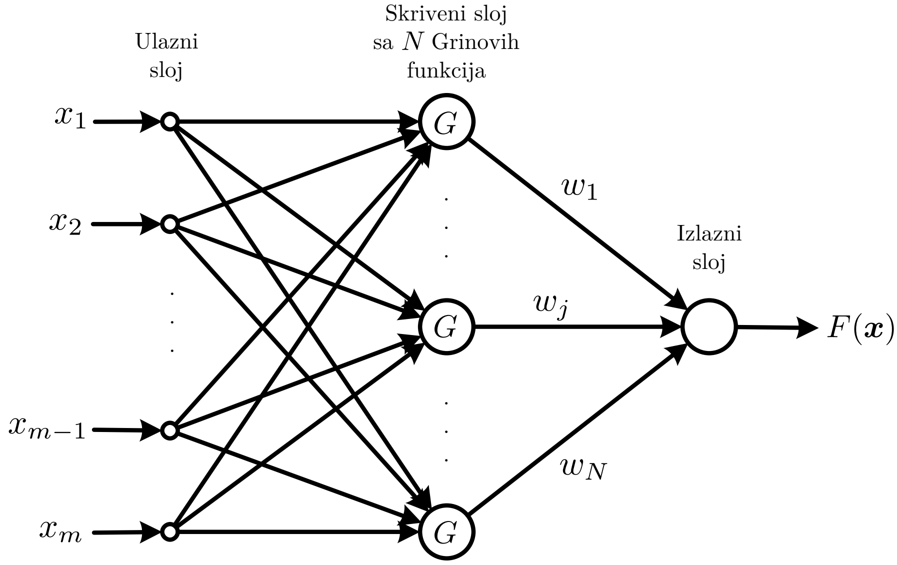
4.4 Regularizacione mreže

Neuralna mreža sa tri sloja: ulazni sa m neurona za ulazne vektore dimenzije m, skriveni sa N neurona čije su aktivacione funkcije Grinove funkcije $G(\boldsymbol{x}, \boldsymbol{x}_i)$ sa centrom u $\boldsymbol{x}_i$ i izlazni sa jednim neuronom (ovo se potpuno analogno može proširiti na proizvoljan broj); koja je potpuno povezana, pri čemu su veze između neurona ulaznog sloja i neurona skrivenog sloja *direktne* (težina 1), a veze između neurona skrivenog sloja i izlaza imaju težine w_i, prikazana na Slici 4.1, se naziva *regularizaciona mreža.*

Težine izlaznog sloja su nepoznati koeficijenti širenja za Grinove funkcije $G(\boldsymbol{x}, \boldsymbol{x}_i)$ i regularizacioni parametar λ koji se određuju pomoću (4.45). Ovo znači da se podrazumijeva da je matrica $\mathbf{G}$ pozitivno određena, kao što je u slučaju korištenja Gausovih funkcija.

Iz perspektive aproksimacije regularizaciona mreža ima 3 poželjne osobine:

1. Regularizaciona mreža je *univerzalni aproksimator* u smislu da može aproksimirati proizvoljno dobro bilo koju funkciju više promjenljivih na kompaktnom potprostoru prostora $\mathbb{R}^m$ uz dovoljan broj skrivenih neurona.



Slika 4.1 *Regularizaciona mreža*

2. Regularizaciona mreža daje najbolju aproksimaciju, tj. za proizvoljnu nelinearnu funkciju f uvijek postoji izbor koeficijenata koji aproksimiraju f bolje od svih ostalih izbora.

3. Rješenje dobijeno pomoću regularizacione mreže je optimalno u smislu minimizacije funkcionala koji mjeri koliko rješenje odstupa od stvarne vrijednosti predstavljene obučavajućim podacima.

4.5 Neuralne mreže sa uopštenim funkcijama radijalne baze

S obzirom da se prethodno opisana mreža formira tako da postoji 1-na-1 poklapanje između centara Grinovih funkcija i obučavajućih ulaznih podataka, za jako veliki broj obučavajućih podataka N regularizaciona mreža može postati računski vrlo skupa. Tačnije, računanje linearnih težina mreže, odnosno koeficijenata širenja, zahtijeva invertovanja $N \times N$ matrice, što raste polinomski sa N, približno sa N^3. Dalje, veće matrice imaju i veću vjerovatnoću da budu *loše uslovljene* (određeno odnosom najveće i najmanje sopstvene vrijednosti matrice; kod dobro uslovljenih matrica taj odnos je manji). Za prevazilaženje ovih računskih problema, kompleksnost mreže treba da bude smanjena, a to zahtjeva aproksimaciju regularizacionog rješenja.

Pristup koji se koristi za to uključuje traženje suboptimalnog rješenja u niže-dimenzionalnom prostoru koje aproksimira regularizaciono rješenje, a zasnovano je na standardnoj tehnici koja se naziva *Galerkinov metod.* U skladu sa tim metodom, aproksimirano rješenje $F^*(\boldsymbol{x})$ se širi na konačnoj bazi, što se može zapisati sa

$$F^*(\boldsymbol{x}) = \sum_{i=1}^{m_1} w_i \varphi_i(\boldsymbol{x}), \tag{4.55}$$

gdje je $\{\varphi_i(\boldsymbol{x}) | i = 1, 2, \ldots, m_1\}$ novi skup baznih funkcija za koje se pretpostavlja da su linearno nezavisne bez gubitka opštosti. Uobičajeno je da je broj baznih funkcija manji od broja obučavajućih podataka, tj. $m_1 \leq N$, a w_i su nove težine. Uzima se

$$\varphi_i(\boldsymbol{x}) = G\left(\|\boldsymbol{x} - \boldsymbol{t}_i\|\right), i = 1, 2, \ldots, m_1, \tag{4.56}$$

gdje je skup centara $\{\boldsymbol{t}_i | i = 1, 2, \ldots, m_1\}$ nepoznat i treba ga odrediti. Ovaj konkretan izbor baznih funkcija jedini garantuje da se u slučaju da je $m_1 = N$ i $\boldsymbol{t}_i = \boldsymbol{x}_i, i = 1, 2, \ldots, N$ dobije standardno regularizaciono rješenje.

Na osnovu (4.55) i (4.56) se dobija

$$F^*(\boldsymbol{x}) = \sum_{i=1}^{m_1} w_i G\left(\|\boldsymbol{x} - \boldsymbol{t}_i\|\right). \tag{4.57}$$

Problem koji treba riješiti je određivanje novog skupa težina w_i tako da se minimizuje nova funkcija koštanja $\mathscr{E}(F^*)$ definisana sa

$$\mathscr{E}(F^*) = \sum_{i=1}^{N} \left(d_i - \sum_{i=1}^{m_1} w_i G\left(\|\boldsymbol{x} - \boldsymbol{t}_i\|\right) \right)^2 + \lambda \|\mathbf{D}F^*\|^2. \tag{4.58}$$

Prvi član na desnoj strani se može posmatrati kao kvadrat Euklidove norme $\|\boldsymbol{d} - \mathbf{G}\boldsymbol{w}\|^2$, gdje je

$$\boldsymbol{d} = \begin{bmatrix} d_1 & d_2 & \cdots & d_N \end{bmatrix}^{\mathrm{T}}, \tag{4.59}$$

$$\mathbf{G} = \begin{bmatrix} G(\boldsymbol{x}_1, \boldsymbol{t}_1) & G(\boldsymbol{x}_1, \boldsymbol{t}_2) & \cdots & G(\boldsymbol{x}_1, \boldsymbol{t}_{m_1}) \\ G(\boldsymbol{x}_2, \boldsymbol{t}_1) & G(\boldsymbol{x}_2, \boldsymbol{t}_2) & \cdots & G(\boldsymbol{x}_2, \boldsymbol{t}_{m_1}) \\ \vdots & \vdots & & \vdots \\ G(\boldsymbol{x}_N, \boldsymbol{t}_1) & G(\boldsymbol{x}_N, \boldsymbol{t}_2) & \cdots & G(\boldsymbol{x}_N, \boldsymbol{t}_{m_1}) \end{bmatrix}, \tag{4.60}$$

$$\boldsymbol{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_{m_1} \end{bmatrix}^{\mathrm{T}}. \tag{4.61}$$

Vektor željenih rješenja $\boldsymbol{d}$ je N-dimenzionalan, kao i ranije, ali matrica Grinovih funkcija $\mathbf{G}$ i vektor težina $\boldsymbol{w}$ imaju drugačije dimenzije: matrica $\mathbf{G}$ je $N \times m_1$ matrica i više nije simetrična, a vektor $\boldsymbol{w}$ je dimenzije m_1.

Iz (4.57) se vidi da je aproksimirajuća funkcija F^* linearna kombinacija Grinovih funkcija za stabilizator $\mathbf{D}$, pa se u skladu s tim, drugi član na desnoj strani jednačine (4.58) može izraziti kao

$$\begin{aligned} \|\mathbf{D}F^*\|^2 &= (\mathbf{D}F^*, \mathbf{D}F^*)_{\mathscr{H}} \\ &= \left[\sum_{i=1}^{m_1} w_i G(\boldsymbol{x}, \boldsymbol{t}_i), \widetilde{\mathbf{D}}\mathbf{D} \sum_{i=1}^{m_1} w_i G(\boldsymbol{x}, \boldsymbol{t}_i) \right]_{\mathscr{H}} \\ &= \left[\sum_{i=1}^{m_1} w_i G(\boldsymbol{x}, \boldsymbol{t}_i), \sum_{i=1}^{m_1} w_i \delta(\boldsymbol{t}_i) \right]_{\mathscr{H}} \\ &= \sum_{j=1}^{m_1} \sum_{i=1}^{m_1} w_j w_i G(\boldsymbol{t}_j, \boldsymbol{t}_i) \\ &= \boldsymbol{w}^{\mathrm{T}} \mathbf{G}_0 \boldsymbol{w} \end{aligned} \tag{4.62}$$

gdje je matrica $\mathbf{G}_0$ simetrična $m_1 \times m_1$ matrica definisana sa

$$\mathbf{G}_0 = \begin{bmatrix} G(\boldsymbol{t}_1, \boldsymbol{t}_1) & G(\boldsymbol{t}_1, \boldsymbol{t}_2) & \cdots & G(\boldsymbol{t}_1, \boldsymbol{t}_{m_1}) \\ G(\boldsymbol{t}_2, \boldsymbol{t}_1) & G(\boldsymbol{x}_2, \boldsymbol{t}_2) & \cdots & G(\boldsymbol{t}_2, \boldsymbol{t}_{m_1}) \\ \vdots & \vdots & & \vdots \\ G(\boldsymbol{t}_{m_1}, \boldsymbol{t}_1) & G(\boldsymbol{t}_{m_1}, \boldsymbol{t}_2) & \cdots & G(\boldsymbol{t}_{m_1}, \boldsymbol{t}_{m_1}) \end{bmatrix} \tag{4.63}$$

Konačno, problem minimizacije iz (4.58) se svodi na

$$\left(\mathbf{G}^{\mathsf{T}}\mathbf{G} + \lambda\mathbf{G}_0\right)\boldsymbol{w} = \mathbf{G}^{\mathsf{T}}\boldsymbol{d}. \tag{4.64}$$

Kad regularizacioni parametar λ teži 0, vektor težina $\boldsymbol{w}$ teži pseudoinverznom rješenju preodređenog problema najmanjih kvadrata, datom sa

$$\boldsymbol{w} = \mathbf{G_{inv}}\boldsymbol{d}, \tag{4.65}$$

gdje je $\mathbf{G_{inv}}$ pseudoinverzna matrica matrice $\mathbf{G}$ dat sa

$$\mathbf{G_{inv}} = \left(\mathbf{G}^{\mathsf{T}}\mathbf{G}\right)^{-1}\mathbf{G}^{\mathsf{T}}. \tag{4.66}$$

Norma koja se koristi u (4.57) je uobičajeno Euklidova norma. Međutim, kad individualni elementi ulaznog vektora $\boldsymbol{x}$ pripadaju različitim klasama, podobnije je koristiti opštiju *težinsku normu*, čija je kvadratna forma definisana sa

$$\|\boldsymbol{x}\|_C^2 = (\mathbf{C}\boldsymbol{x})^{\mathsf{T}}(\mathbf{C}\boldsymbol{x}) = \boldsymbol{x}^{\mathsf{T}}\mathbf{C}^{\mathsf{T}}\mathbf{C}\boldsymbol{x}, \tag{4.67}$$

gdje je $\mathbf{C}$ $m \times m$ *matrica težinske norme*, a m dimenzija ulaznog vektora $\boldsymbol{x}$.

Na osnovu ove definicije težinske norme, (4.57) dobija opštiji oblik

$$F^*(\boldsymbol{x}) = \sum_{i=1}^{m_1} w_i G\left(\|\boldsymbol{x} - \boldsymbol{t}_i\|_C\right). \tag{4.68}$$

Težinska norma slijedi direktno iz generalizacije m dimenzionalnog Laplasijana iz definicije pseudodiferencijalnog operatora $\mathbf{D}$ iz (4.52). Gausova funkcija radijalne baze $G\left(\|\boldsymbol{x} - \boldsymbol{t}_i\|_C\right)$ sa centrom u $\boldsymbol{t}_i$ i sa matricom težinske norme $\mathbf{C}$ se može zapisati kao

$$G\left(\|\boldsymbol{x} - \boldsymbol{t}_i\|_C\right) = e^{-(\boldsymbol{x}-\boldsymbol{t}_i)^{\mathsf{T}}\mathbf{C}^{\mathsf{T}}\mathbf{C}(\boldsymbol{x}-\boldsymbol{t}_i)} = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{t}_i)^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{t}_i)}, \tag{4.69}$$

gdje je inverzna matrica $\boldsymbol{\Sigma}^{-1}$ definisana sa

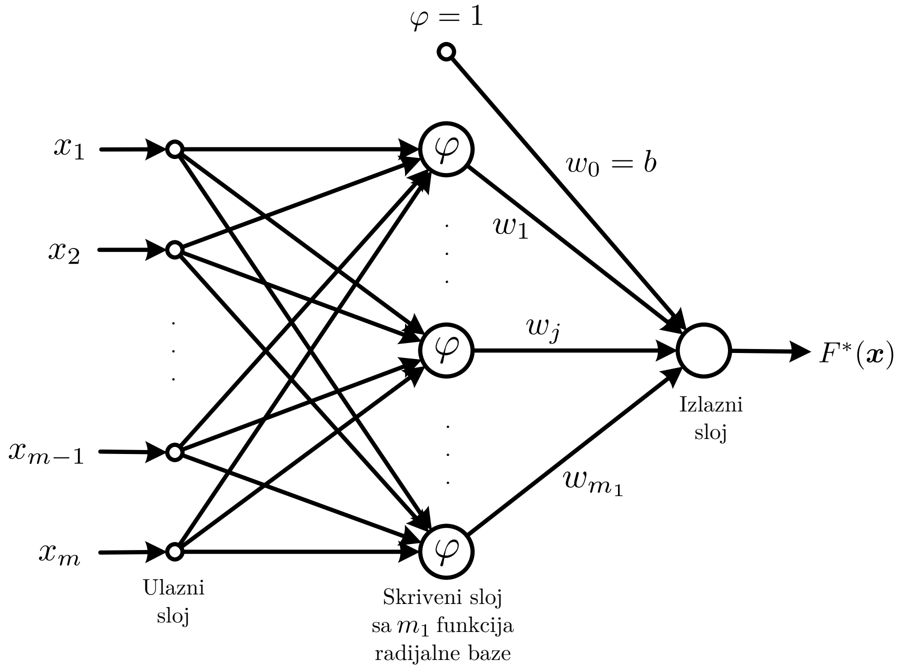
$$\frac{1}{2}\boldsymbol{\Sigma}^{-1} = \mathbf{C}^{\mathsf{T}}\mathbf{C}. \tag{4.70}$$

Jednačina (4.69) predstavlja Gausovu funkciju više promjenljivih sa vektorom srednjih vrijednosti $\boldsymbol{t}_i$ i matricom kovarijansi $\boldsymbol{\Sigma}$ i kao takva predstavlja uopštenje jednačine (4.48).

Rješenje problema aproksimacije iz (4.58) predstavlja okvir za *neuralne mreže sa uopštenim funkcijama radijalne baze*, čija je struktura prikazana na Slici 4.2. Kod ove mreže ostavljena je i mogućnost pomjeraja (promjenljiva nezavisna od ulaza) u izlaznoj jedinici.

Strukturno gledano, mreže sa Slika 4.1 i 4.2 su slične. Međutim, razlikuju se u dvije važne stavke:

1. Broj neurona u skrivenom sloju mreže sa uopštenim RBF je m_1 koje je uobičajeno manje od N (broja neurona u skrivenom sloju regularizacione mreže, odnosno broja obučavajućih podataka).
2. Kod regularizacione mreže su nepoznate samo težine veza, dok su kod mreže sa uopštenim RBF pored toga nepoznati i centri aktivacionih funkcija, kao i matrica težinske norme.



Slika 4.2 *Neuralna mreža sa uopštenim funkcijama radijalne baze*

4.6 Aproksimacione sposobnosti RBF mreža

Neka je $G : \mathbb{R}^m \to \mathbb{R}$ integrabilna ograničena neprekidna funkcija takva da je

$$\int\limits_{\mathbb{R}^m} G(\boldsymbol{x})d\boldsymbol{x} \neq 0. \tag{4.71}$$

Neka je $\mathscr{S}_G$ familija RBF mreža koje se sastoje od funkcija $G : \mathbb{R}^m \to \mathbb{R}$ predstavljenih sa

$$F(\boldsymbol{x}) = \sum_{i=1}^{m_1} w_i G\left(\frac{\boldsymbol{x} - \boldsymbol{t}_i}{\sigma}\right), \tag{4.72}$$

gdje je $\sigma > 0, w_i \in \mathbb{R}, \boldsymbol{t}_i \in \mathbb{R}^m, i = 1, 2, \ldots, m_1$.

Na osnovu uvedenih pojmova daje se *Teorema o univerzalnoj aproksimaciji za RBF mreže* [142].

Teorema 4.3. *Za bilo koju neprekidnu funkciju $f(\boldsymbol{x})$ postoji RBF mreža sa skupom centara $[\boldsymbol{t}_i]_{i=1}^{m_1}$ i zajednička širina $\sigma > 0$ takva da je funkcija $F(\boldsymbol{x})$ realizovana pomoću te RBF mreže bliska funkciji $f(\boldsymbol{x})$ u L_p normi, $p \in [1, \infty)$.*

Pored osobine univerzalnog aproksimatora RBF mreža, tu je i pitanje stepena aproksimacije koji postižu ove mreže. Kompleksnost klase aproksimacionih funkcija raste eksponencijalno u odnosu $\frac{m}{s}$, gdje je m dimenzionalnost ulaza, a *s indeks glatkosti* koji mjeri broj ograničenja nametnutih aproksimacionoj funkciji u datoj klasi. Belmanovo *prokletstvo dimenzionalnosti* kaže da, nezavisno od korištene tehnike aproksimacije, ako je indeks glatkosti s konstantan, broj parametara potrebnih da aproksimaciona funkcija zadrži traženu preciznost se povećava eksponencijalno sa dimenzionalnošću ulaza m. Jedini način na koji se može postići stepen konvergencije nezavisan od dimenzionalnosti ulaza m je da se indeks glatkosti s povećava sa brojem parametara u aproksimacionoj funkciji da se kompenzuje kompleksnost. Odavde se može izvući zaključak da prostor aproksimacionih funkcija postaje ograničeniji povećanjem dimenzionalnosti ulaza m. Ovo važi i za RBF mreže i za višeslojne perceptrone.

Diskusija o aproksimaciji ne može biti kompletna bez razmatranja važne stavke - *greške generalizacije*. Greška generalizacije ima dvije komponente: *greška aproksimacije* - rezultat ograničene sposobnosti mreže da predstavi željenu funkciju, dobija se pri testiranju mreže podacima kojima je mreža i obučena; i *greška estimacije* - rezultat ograničene količine informacija u obučavajućim podacima, dobija se pri testiranju mreže podacima koji nisu učestvovali u obučavanju. Korištenjem ovog oblika dekompozicije, izvedena je granica greške generalizacije RBF mreže sa Gausovim funkcijama, izražena preko veličine skrivenog sloja i veličine obučavajućeg skupa.

Neka je G klasa RBF mreža sa Gausovim funkcijama sa m ulaznih neurona i m_1 neurona u skrivenom sloju. Neka je $f(\boldsymbol{x})$ funkcija koja se aproksimira. Neka je obučavajući skup $\mathscr{T} = [\boldsymbol{x}_i, \boldsymbol{d}_i]_{i=1}^{N}$ dobijen slučajnim uzorkovanjem funkcije $f(\boldsymbol{x})$. Tada je, za bilo koju vrijednost parametra povjerenja $\delta \in (0, 1]$, greška generalizacije koju pravi mreža odozgo ograničena sa

$$O\left(\frac{1}{m_1}\right) + O\left(\frac{mm_1}{N}\log(m_1 N) + \frac{1}{N}\log\sqrt{\frac{1}{\delta}}\right) \tag{4.73}$$

sa vjerovatnoćom većom od $1 - \delta$.

4.7 Poređenje RBF mreža i višeslojnih perceptrona

RBF mreže i višeslojni perceptroni su primjeri nelinearnih slojevitih neuralnih mreža bez povratnih veza. Obje vrste su univerzalni aproksimatori, tako da nije iznenađenje da postoji mreža jedne vrste koja može tačno da oponaša mrežu druge vrste. Međutim, ove dvije vrste mreža se razlikuju u nekoliko važnih stavki:

1. RBF mreža ima jedan skriveni sloj, dok višeslojni perceptron može da ima jedan ili više skrivenih slojeva.
2. Neuroni skrivenog ili izlaznog sloja višeslojnog perceptrona uobičajeno imaju isti model (istu aktivacionu funkciju), dok se neuroni skrivenog sloja RBF mreže razlikuju i služe različitim svrhama od onih u izlaznom sloju.
3. Skriveni sloj RBF mreže je nelinearan, a izlazni linearan, dok su kod višeslojnog perceptrona i skriveni i izlazni sloj uobičajeno nelinearni.
4. Argumenti aktivacionih funkcija skrivenih neurona RBF mreže su Euklidske udaljenosti između ulaznog vektora i centra te funkcije, dok su kod višeslojnog perceptrona to skalarni proizvodi ulaznog vektora i vektora sinaptičkih težina konkretnog neurona.
5. Višeslojni perceptroni realizuju globalnu aproksimaciju nelinearnih funkcija, dok RBF mreže realizuju lokalnu aproksimaciju nelinearnih funkcija. Ovo

znači da će višeslojni perceptron vjerovatno imati manje parametara nego RBF mreža pri istim zahtjevima za preciznost.

Posebno je važno istaći način na koji se vrši izračunavanje sinaptičkih težina, odnosno obučavanje kod ovih mreža. Kod višeslojnog perceptrona to je uobičajeno iterativni proces (npr. propagacija greške unazad), dok se kod RBF mreža problem svodi na rješavanje linearnog sistema jednačina. Međutim, kod RBF mreža ostaje i problem izbora centara aktivacionih funkcija, o čemu će biti riječi u nastavku.

4.8 Strategije obučavanja

Proces obučavanja RBF mreža se može posmatrati kao proces koji se odvija u dvije faze na dvije različite vremenske skale. Dok aktivacione funkcije skrivenog sloja polako evoluiraju u skladu sa nekom nelinearnom strategijom optimizacije, težine izlaznog sloja se prilagođavaju tim promjenama brzo korištenjem linearne strategije optimizacije. S obzirom da slojevi imaju različite svrhe, sasvim je opravdano razdvojiti optimizaciju po slojevima, pa čak i vremenskim skalama.

Razvijene su različite strategije učenja u zavisnosti od načina izbora centara funkcija radijalnih baza.

4.8.1 Slučajno izabrani fiksni centri

Najjednostavniji pristup je da se pretpostave fiksne funkcije radijalne baze za aktivacione funkcije skrivenih neurona. Lokacije centara se mogu izabrati slučajno iz obučavajućeg skupa podataka. Ovo se smatra smislenim pristupom, pod uslovom da su obučavajući podaci raspoređeni reprezentativno za problem koji se rješava. Za same funkcije radijalne baze se dalje može koristiti *izotropna* Gausova funkcija čija je standardna devijacija fiksirana u skladu sa rasporedom centara. Konkretno, ona se definiše sa

$$G\left(\|\boldsymbol{x}-\boldsymbol{t}_i\|^2\right) = e^{-\frac{m_1}{d_{\max}^2}\|\boldsymbol{x}-\boldsymbol{t}_i\|^2}, i = 1,2,\ldots,m_1, \tag{4.74}$$

gdje je m_1 broj centara, a $d_{\max}$ najveća udaljenost između centara. Time je standardna devijacija (širina) svih Gausovih funkcija fiksirana na

$$\sigma = \frac{d_{\max}}{\sqrt{2m_1}}. \tag{4.75}$$

Ovo obezbjeđuje da pojedinačne funkcije radijalne baze nisu ni previše oštre, ni previše zaravnjene, što su dva granična slučaja koje treba izbjegavati.

Umjesto fiksiranja širine, moguće je na osnovu obučavajućih podataka odrediti oblasti gdje ima manje, odnosno više podataka, pa za funkcije koje imaju centre u tim oblastima birati veće, odnosno manje širine, respektivno.

Kad su funkcije radijalnih baza fiksirane, jedino obučavanje koje treba obaviti je određivanje sinaptičkih težina, a to se može izvesti pomoću (4.65). U ovom slučaju, elementi matrice $\mathbf{G}$ su g_{ji}, dati sa

$$g_{ji} = e^{-\frac{m_1}{d_{\max}^2}\|\boldsymbol{x}_j-\boldsymbol{t}_i\|^2}, i = 1,2,\ldots,m_1, j = 1,2,\ldots,N. \tag{4.76}$$

4.8.2 Samoorganizujući izbor centara

Glavni problem prethodno opisanog metoda fiksnih centara je činjenica da bi mogao da zahtijeva veliki skup obučavajućih podataka za zadovoljavajući nivo performansi. Jedan od načina da se prevaziđe to ograničenje je da se koristi *hibridni proces obučavanja* [143], koji se sastoji od dvije različite faze:

- faza *samoorganizujućeg učenja*, čija je svrha procjena podobnih lokacija centara funkcija radijalnih baza skrivenog sloja, i
- faza *nadgledanog učenja*, koja dovršava dizajn mreže estimacijom linearnih sinaptičkih težina izlaznog sloja.

Iako postoji više načina za implementaciju ovih faza, preporučuje se iterativni pristup.

Za samoorganizujući proces je potreban neki algoritam grupisanja (eng. *clustering*) koji grupiše dati skup podataka u podgrupe tako da svaka bude koliko god je moguće homogena. Jedan takav algoritam je *k-means clustering* algoritam [144], koji postavlja centre funkcija radijalnih baza samo u one oblasti ulaznog prostora $\mathscr{H}$ gdje su prisutne značajne količine podataka. Neka je m_1 broj funkcija radijalne baze, čija vrijednost se najčešće određuje na osnovu dovoljnog broja eksperimenata. Neka je $[\boldsymbol{t}_k(n)]_{k=1}^{m_1}$ skup centara funkcija radijalne baze u n-toj iteraciji algoritma. k-means clustering algoritam se odvija kako slijedi:

1. *Inicijalizacija.* Biraju se slučajne vrijednosti za početne centre $\boldsymbol{t}_k(0)$ uz jedno ograničenje - da te vrijednosti budu različite. Često je poželjno da Euklidska udaljenost centara bude mala.
2. *Uzorkovanje.* Kao ulazni uzorak se iz ulaznog prostora $\mathscr{H}$ uzima vektor $\boldsymbol{x}$ sa određenom vjerovatnoćom i uvodi u algoritam u n-toj iteraciji.
3. *Pronalaženje sličnosti.* Neka je $k(\boldsymbol{x})$ indeks najsličnijeg (pobjedničkog) centra za ulazni vektor $\boldsymbol{x}$. Taj indeks se u n-toj iteraciji određuje na osnovu kriterijum najmanje Euklidske udaljenosti
$$k(\boldsymbol{x}) = \arg\min_k \|\boldsymbol{x}(n) - \boldsymbol{t}_k(n)\|, k = 1, 2, \ldots, m_1. \tag{4.77}$$
4. *Postavljanje novih vrijednosti.* Podešavaju se centri funkcija radijalnih baza prema pravilu
$$\boldsymbol{t}_k(n+1) = \begin{cases} \boldsymbol{t}_k(n) + \eta\left[\boldsymbol{x}(n) - \boldsymbol{t}_k(n)\right], & k = k(\boldsymbol{x}) \\ \boldsymbol{t}_k(n), & \text{inače} \end{cases} \tag{4.78}$$
gdje je η *parametar brzine obučavanja* za koji važi $0 < \eta < 1$.
5. *Razmatranje završetka algoritma.* n se uvećava za 1 i prelazi se na korak 2. Procedura se zaustavlja kad nema primjetnih promjena centara $\boldsymbol{t}_k$.

Ovaj algoritam je poseban slučaj obučavajućeg procesa samoorganizujućih mreža, opisan u Potpoglavlju 3.8.

Nedostatak ovog algoritma je to što može da postigne lokalno optimalno rješenje koje je zavisno do početnog izbora centara. Mnogo računskih resursa može biti potrošeno ako neki od početnih centara ostanu u regionima ulaznog prostora $\mathscr{H}$ u kojem nema mnogo podataka, zbog čega nikad ne dobiju priliku da promijene lokaciju. Mogući rezultat je nepotrebno velika mreža. Za prevazilaženje ovog problema razvijene su varijacije k-means algoritma, poput *poboljšanog k-means algoritma* [145].

Nakon identifikovanja pojedinačnih centara Gausovih funkcija radijalne baze i njihove zajedničke širine, prelazi se na drugu fazu. Jednostavan metod za estimaciju težina izlaznog sloja je npr. metod najmanjih kvadrata.

4.8.3 Nadgledani izbor centara

U ovom pristupu centri funkcija radijalnih baza i svi ostali slobodni parametri mreže prolaze nadgledani proces obučavanja. Ovo znači da RBF mreža ima najopštiji mogući oblik. Prirodan kandidat za takav proces je *učenje korekcijom greške*, što se najpogodnije implementira korištenjem procedure gradijentnog spuštanja koja predstavlja uopštenje metode najmanjih kvadrata, a o čemu je bilo riječi ranije.

Prvi korak u razvoju takve procedure obučavanja je definisanje trenutne vrijednosti greške

$$\mathscr{E} = \frac{1}{2}\sum_{j=1}^{N} e_j^2, \tag{4.79}$$

gdje je N veličina obučavajućeg skupa, a e_j signal greške definisan sa

$$e_j = d_j - F^*(\boldsymbol{x}_j) = d_j - \sum_{i=1}^{m} w_i G\left(\|\boldsymbol{x}_j - \boldsymbol{t}_i\|_{C_i}\right). \tag{4.80}$$

Zahtjev je da se nađu slobodni parametri w_i, $\boldsymbol{t}_i$ i $\boldsymbol{\Sigma}_i^{-1}$ (povezan sa matricom težinske norme $\mathbf{C}_i$ na osnovu (4.70)) tako da se minimuzuje $\mathscr{E}$.

Linearne težine se u n-toj iteraciji podešavaju na osnovu

$$\frac{\partial \mathscr{E}}{\partial w_i(n)} = -\sum_{j=1}^{N} e_j(n) G\left(\|\boldsymbol{x}_j - \boldsymbol{t}_i(n)\|_{C_i}\right), \tag{4.81}$$

$$w_i(n+1) = w_i(n) - \eta_1 \frac{\partial \mathscr{E}}{\partial w_i(n)}. \tag{4.82}$$

Pozicije centara se u n-toj iteraciji podešavaju na osnovu

$$\frac{\partial \mathscr{E}}{\partial \boldsymbol{t}_i(n)} = 2w_i(n)\sum_{j=1}^{N} e_j(n) G'\left(\|\boldsymbol{x}_j - \boldsymbol{t}_i(n)\|_{C_i}\right) \boldsymbol{\Sigma}_i^{-1}\left[\boldsymbol{x}_j - \boldsymbol{t}_i(n)\right], \tag{4.83}$$

$$\boldsymbol{t}_i(n+1) = \boldsymbol{t}_i(n) - \eta_2 \frac{\partial \mathscr{E}}{\partial \boldsymbol{t}_i(n)}, \tag{4.84}$$

gdje je $G'(\cdot)$ prvi izvod Grinove funkcije $G(\cdot)$ po njenom argumentu.

Širine se u n-toj iteraciji podešavaju na osnovu

$$\frac{\partial \mathscr{E}}{\partial \boldsymbol{\Sigma}_i^{-1}(n)} = -w_i(n) \sum_{j=1}^{N} e_j(n) G' \left(\|\boldsymbol{x}_j - \boldsymbol{t}_i(n)\|_{C_i} \right) \mathbf{Q}_{ji}(n), \tag{4.85}$$

$$\mathbf{Q}_{ji}(n) = [\boldsymbol{x}_j - \boldsymbol{t}_i(n)] [\boldsymbol{x}_j - \boldsymbol{t}_i(n)]^{\mathtt{T}}, \tag{4.86}$$

$$\boldsymbol{\Sigma}_i^{-1}(n+1) = \boldsymbol{\Sigma}_i^{-1}(n) - \eta_3 \frac{\partial \mathscr{E}}{\partial \boldsymbol{\Sigma}_i^{-1}(n)}. \tag{4.87}$$

U [146] je za zadatak iz [147] izvršeno poređenje performansi RBF mreža sa fiksnim centrima sa RBF mrežama kod kojih se položaj centara uči nedgledanim obučavanjem, uz dodatno poređenje sa višeslojnim perceptronima, i eksperimenti su pokazali sljedeće:

- Generalizacija vršena pomoću RBF mreža sa fiksnim centrima nije ni izbliza dobra kao pomoću višeslojnih perceptrona obučenih algoritmom propagacije greške unazad.
- Neuralne mreže sa uopštenim funkcijama radijalne baze sa pozicijom centara određenom nadgledanim učenjem su znatno premašile performanse višeslojih perceptrona.

5. Brza grupisana vještačka neuralna mreža sa Gausovim aktivacionim funkcijama

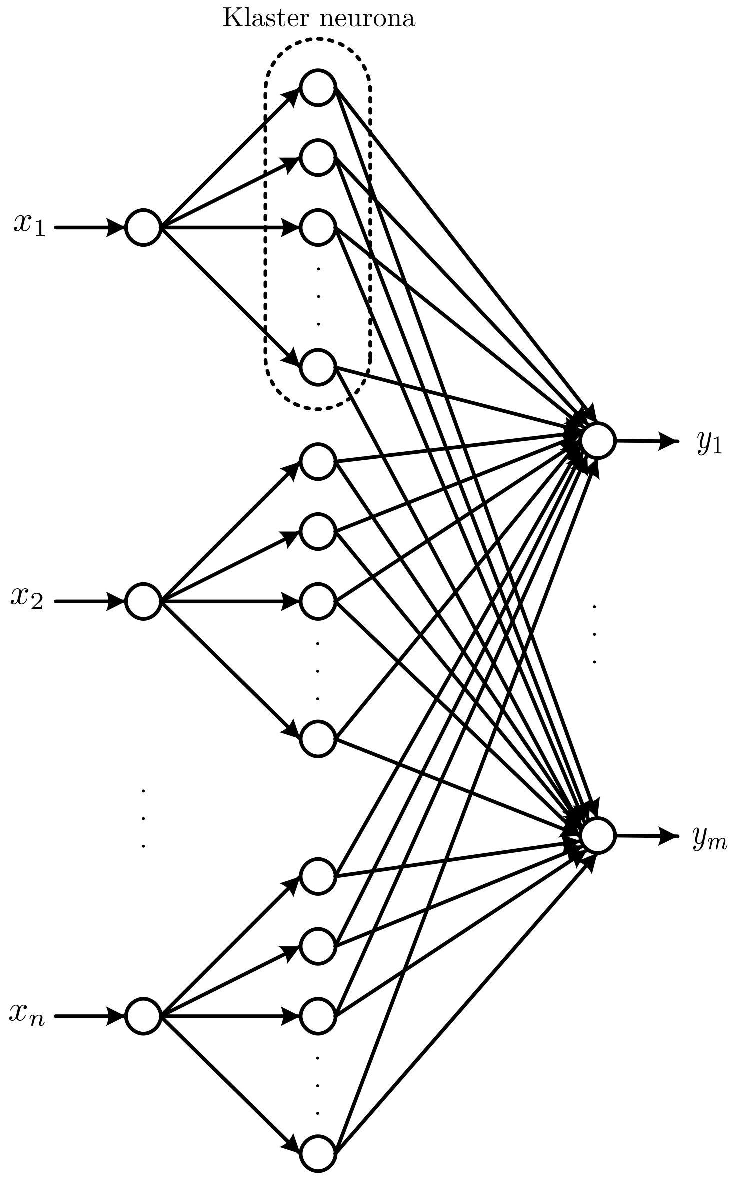
U ovom poglavlju opisana je nova vještačka neuralna mreža sa tri sloja: ulaznim, skrivenim i izlaznim. Glavna karakteristika nove mreže je to da postoje veze između svakog neurona ulaznog sloja sa samo određenim brojem neurona skrivenog sloja i da su svi neuroni skrivenog sloja povezani sa samo jednim neuronom ulaznog sloja, za razliku od potpuno povezanih mreža gdje su svi neuroni ulaznog sloja povezani sa svim neuronim skrivenog sloja. Svi neuroni skrivenog sloja koji su povezani sa istim neuronom ulaznog sloja čine jednu grupu neurona nazvanu **klaster neurona** (eng. *Neuron Cluster*). Svi neuroni ulaznog i izlaznog sloja imaju aktivacionu funkciju $a(x) = x$. Aktivaciona funkcija neurona skrivenog sloja je funkcija radijalne baze, i predlaže se korištenje Gausove funkcije $a(x) = \mathrm{e}^{-\left(\frac{x-c}{\sigma}\right)^2}$. Broj neurona u svakom klasteru može da varira, kao i centri c, te širine σ aktivacionih funkcija svakog neurona. Težine svih veza između ulaznog i skrivenog sloja su jednake 1. Težine veza između skrivenog i izlaznog sloja se određuju jednim korištenjem metoda najmanjih kvadrata (eng. *Least Squares method - LS*) za rješavanje sistema linearnih jednačina za svaki od izlaza, što čini brzinu procesa obučavanja vrlo velikom. Zbog svega navedenog, takva vještačka neuralna mreža je nazvana **brza grupisana vještačka neuralna mreža sa Gausovim aktivacionim funkcijama** (eng. *Fast Clustered Radial Basis Function Network - FCRBFN*) [148]. Struktura FCRBFN mreže je prikazana na Slici 5.1. Proces obučavanja ovakve mreže je opisan u nastavku.

5.1 Određivanje težina

Neka mreža ima n ulaza i m izlaza. Ovo znači da postoji n klastera neurona, kako je opisano ranije, jedan za svaki ulaz. Dalje, neka postoji n_i neurona u i-tom klasteru neurona ($i = 1, 2, \ldots, n$). Odavdje se može zaključiti da je ukupan broj neurona u skrivenom sloju dat sa $n_{tn} = \sum_{i=1}^{n} n_i$.

Neka je aktivaciona funkcija j-tog neurona u i-tom klasteru neurona definisana pomoću svog centra $c_{i,j}$ i svoje širine $\sigma_{i,j}$ i neka je težina veze između j-tog neurona u i-tom klasteru neurona i k-tog neurona u izlaznom sloju označena sa $w_{i,j,k}$.

Neka je ulaz mreže dat u vidu n-dimenzionog vektora $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$



Slika 5.1 *Struktura brze grupisane vještačke neuralne mreže sa Gausovim aktivacionim funkcijama*

i neka je izlaz mreže dat u vidu m-dimenzionog vektora $\mathbf{y} = \left[\begin{array}{cccc} y_1 & y_2 & \cdots & y_m \end{array}\right]$, gdje je y_k $(k = 1, 2, \ldots, m)$ izlaz k-tog neurona izlaznog sloja određen sa

$$y_k = \sum_{i=1}^{n} \sum_{j=1}^{n_i} w_{i,j,k} \mathrm{e}^{-\left(\frac{x_i - c_{i,j}}{\sigma_{i,j}}\right)^2}. \tag{5.1}$$

Dalje, neka postoji n_p poznatih parova ulaznih i željenih izlaznih vektora $\left(\begin{array}{cc} {}^{(l)}\mathbf{x} & {}^{(l)}\mathbf{y} \end{array}\right)$ $(l = 1, 2, \ldots, n_p)$ za proces obučavanja. Ako se to primijeni na (5.1), formira se sistem od n_p linearnih jednačina za svaki od m izlaza. Promjenljive u ovom sistemu su težine veza između jednog od neurona u izlaznom sloju i svih neurona u skrivenom sloju. Ovaj sistem sistema linearnih jednačina se može zapisati u obliku

$$\mathbf{\Phi W} = \mathbf{Y}. \tag{5.2}$$

$\mathbf{\Phi}$ je $n_p \times n_{tn}$ matrica $\mathbf{\Phi} = [\varphi_{a,b}]_{n_p \times n_{tn}}$ gdje je

$$\varphi_{a,b} = \mathrm{e}^{-\left(\frac{^{(a)}x_i - c_{i,j}}{\sigma_{i,j}}\right)^2}, \tag{5.3}$$

uz $(i = 1, 2, \ldots, n; j = 1, 2, \ldots, n_i; b = j + \sum_{t=1}^{i-1} n_t)$.

$\mathbf{W}$ je $n_{tn} \times m$ matrica $\mathbf{W} = [w_{a,b}]_{n_{tn} \times m}$ gdje je

$$w_{a,b} = w_{i,j,b}, \tag{5.4}$$

uz $(i = 1, 2, \ldots, n; j = 1, 2, \ldots, n_i; a = j + \sum_{t=1}^{i-1} n_t)$.

$\mathbf{Y}$ je $n_p \times m$ matrica $\mathbf{Y} = [y_{a,b}]_{n_p \times m}$ gdje je

$$y_{a,b} = {}^{(a)}y_b. \tag{5.5}$$

Ovaj sistem sistema linearnih jednačina, ako je rješiv, ima rješenje dato sa

$$\mathbf{W} = \mathbf{\Phi_{inv}} \mathbf{Y}, \tag{5.6}$$

gdje je $\mathbf{\Phi_{inv}}$ pseudo-inverz matrice $\mathbf{\Phi}$ izračunat pomoću

$$\mathbf{\Phi_{inv}} = \left(\mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^{\mathrm{T}}. \tag{5.7}$$

Ako je $n_p = n_{tn}$ onda je dati sistem kvadratni i (5.7) može biti zapisana u skraćenom obliku, korištenjem standardnog inverza

$$\mathbf{\Phi_{inv}} = \mathbf{\Phi}^{-1}. \tag{5.8}$$

Ovo je ujedno i rješenje problema interpolacije funkcije definisane tačkama $\left({}^{(\mathbf{l})}\mathbf{x} \quad {}^{(\mathbf{l})}\mathbf{y} \right)$ $(l = 1, 2, \ldots\ldots, n_p)$ korištenjem funkcija radijalne baze pomoću metode najmanjih kvadrata.

Relativno je lako odrediti da li je sistem rješiv na osnovu Kroneker-Kapelijeve teoreme, tj. računanjem ranga matrica $\mathbf{\Phi}$ i $\left[\; \mathbf{\Phi} \quad | \quad \mathbf{Y} \; \right]$. Za razliku od toga, iterativni metodi su zavisni od početne pretpostavke rješenja, kao i od strukture mreže, što otežava utvrđivanje razloga mogućeg neuspjeha u obučavanju mreže.

5.2 Dodavanje novog znanja

Ovakva vrsta mreže dozvoljava da se novo znanje doda nakon početnog obučavanja bez potrebe za računanjem inverza iz (5.7), što je dio procesa obučavanja koji zahtijeva najviše vremena. To dodavanje se vrši korištenjem rekurzivne metode najmanjih kvadrata (eng. *Recursive Least Squares method - RLS*) čiji je skraćeni opis dat u nastavku.

Dodavanje novog para $\left({}^{(*)}\mathbf{x} \quad {}^{(*)}\mathbf{y} \right)$ u obučavajući skup implicira dodavanje novog reda u matricu $\mathbf{\Phi}$. Neka je novi red označen sa ϕ.

Matrica $\mathbf{P}$ je data sa $\mathbf{P} = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}$. Nakon obučavanja matrica $\mathbf{P}$ ima vrijednost $\mathbf{P_t}$, a matrica $\mathbf{W}$ ima vrijednost $\mathbf{W_t}$. Da bi mreža bila obučena i za novi par, nova vrijednost matrice $\mathbf{P}$, označena sa $\mathbf{P_{t+1}}$, se računa na osnovu

$$\mathbf{P_{t+1}} = \mathbf{P_t} + \frac{\mathbf{P_t}\phi^{\mathrm{T}}\phi\mathbf{P_t}}{1 - \phi\mathbf{P_t}\phi^{\mathrm{T}}}. \tag{5.9}$$

Matrica greške $\mathbf{E}$ je data sa

$$\mathbf{E} = {}^{(*)}\mathbf{y} - \phi^{\mathrm{T}}\mathbf{W_t}. \tag{5.10}$$

Matrica korekcije $\mathbf{K}$ je data sa

$$\mathbf{K} = \mathbf{P_{t+1}}\phi. \tag{5.11}$$

Konačno, nova vrijednost matrice težina $\mathbf{W_{t+1}}$ se računa pomoću

$$\mathbf{W_{t+1}} = \mathbf{W_t} + \mathbf{KE}. \tag{5.12}$$

Ova mogućnost FCRBFN mreže da usvoji novo znanje u jednoj iteraciji, i to za kratko vrijeme, kvalifikuje takvu mrežu za korištenje u vidu adaptivnog kontrolera.

5.3 O izboru parametara klastera neurona

Ne postoji očigledan način za određivanje vrijednosti parametara centra i širine aktivacione funkcije svakog od neurona u klasteru neurona, niti postoji precizna formula za broj neurona u svakom od klastera. Tokom rada na ovom istraživanju, izvršen je veliki broj testova i provjera koji su rezultovali u preporukama koje su date u nastavku.

Neka je sa r_i označen opseg obučavajućih vrijednosti i-tog ulaza mreže ${}^{(l)}x_i$ ($l = 1, 2, \ldots, n_p$, gdje je n_p, kao i ranije, broj parova za obučavanje), definisan na osnovu minimalne i maksimalne vrijednosti tog ulaza pri obučavanju.

$$r_i = \left[\min_l {}^{(l)}x_i, \max_l {}^{(l)}x_i\right] \tag{5.13}$$

Neka je sa wr_i označena širina opsega r_i

$$wr_i = \max_l {}^{(l)}x_i - \min_l {}^{(l)}x_i \tag{5.14}$$

Neka je sa ex_i označeno dozvoljeno proširenje opsega r_i (uobičajeno izraženo u procentima), koje predstavlja koliko se opseg povećava u odnosu na njegovu širinu, tako da je prošireni opseg dat sa

$$er_i = \left[\min_l {}^{(l)}x_i - ex_i wr_i, \max_l {}^{(l)}x_i + ex_i wr_i\right]. \tag{5.15}$$

Ovo proširenje se procjenjuje u skladu sa očekivanim vrijednostima datog ulaza pri uobičajenom korištenju mreže.

Širina proširenog opsega je

$$wer_i = \max er_i - \min er_i. \tag{5.16}$$

Kao i ranije, neka postoji n_i neurona u i-tom klasteru neurona mreže. Centri $c_{i,j}$ $(j = 1, 2, \ldots, n_i)$ aktivacionih funkcija neurona u i-tom klasteru neurona se pozicioniraju ekvidistantno u proširenom opsegu er_i.

Širine $\sigma_{i,j}$ $(j = 1, 2, \ldots, n_i)$ aktivacionih funkcija neurona u i-tom klasteru neurona se biraju da budu jednake za sve neurone u tom klasteru.

Neka je sa sr_i označen odnos pokrivanja, koji predstavlja odnos širine $\sigma_{i,j}$ tih aktivacionih funkcija i udaljenosti dc_i između dva susjedna centra tih aktivacionih funkcija. Kako su centri raspoređeni ekvidistantno, pomenuta udaljenost je data sa

$$dc_i = c_{i,2} - c_{i,1} = \frac{wer_i}{n_i - 1}. \tag{5.17}$$

Odatle, širina $\sigma_{i,j}$ je data sa

$$\sigma_{i,j} = sr_i dc_i. \tag{5.18}$$

Vrijednosti parametara n_i, ex_i i sr_i se biraju eksperimentalno u skladu sa konkretnim problemom. U Potpoglavlju 5.5 detaljno je opisan jedan takav eksperiment.

5.4 Regresija i generalizacija

Za testiranje predložene FCRBFN mreže, izvršeno je poređenje sa standardnom neuralnom mrežom bez povratnih veza sa jednim skrivenim slojem koja je obučena algoritmom propagacije greške unazad (u daljem tekstu BP), ekstremnom mašinom za učenje (eng. *Extreme Learning Machine* - ELM) i mašinom sa potpornim vektorima (eng. *Support Vector Machines* - SVM). Svi sistemi su primijenjeni na rješavanje zadatka regresije na četiri problema opisana u Tabeli 5.1 (California housing, Bank domains, Machine CPU i Abalone) [42]. Podaci za ove probleme se mogu naći na adresi `http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html`.

Sve simulacije su vršene u MATLAB R2013a programskom okruženju na računaru sa 8-jezgrenim procesorom takta 2.4 GHz i 8 GB memorije RAM tipa.

Tabela 5.1 *Specifikacija skupova podataka za regresiju*

Skup podataka	**Broj atributa**	**Broj mjerenja**	**Za obučavanje**	**Za testiranje**
California housing	8	20460	8000	12460
Bank domains	8	8192	4500	3692
Machine CPU	6	209	100	109
Abalone	8	4177	2000	2177

Za BP i ELM broj neurona u skrivenom sloju je 30 i aktivaciona funkcija je sigmoidalna $a(x) = \frac{1}{1+e^{-x}}$. Za SVM, parametar koštanja je izabran da bude $C = 2^1 = 2$, a parametar jezgra je izabran da bude $\gamma = 2^6 = 64$. Ove vrijednosti su izabrane u skladu sa [42].

Kod FCRBFN su korištena 2 neurona po klasteru neurona, sa dozvoljenim proširenjem opsega ex_i od 50% širine opsega r_i i odnosom pokrivanja od $sr_i = 100\%$. Ove vrijednosti su dobijene kao rezultat većeg broja eksperimenata.

Za svaki zadatak, ulazni argumenti su normalizovani na opseg $[-1, 1]$, a izlazni atribut na opseg $[0, 1]$. Rezultati su usrednjeni na osnovu 30 testova. Prosječna trajanja obučavanja i testiranja, te srednje vrijednosti i standardne devijacije srednjekvadratnih grešaka obučavanja i testiranja za sve aplikacije su prikazani u Tabelama 5.2, 5.3, 5.4 i 5.5 (jedna tabela za svaki zadatak, sa dva reda po aplikaciji koja rješava problem, jedan za obučavajuće primjere, jedan za testne primjere).

Kao što se može vidjeti iz rezultata, SVM ima najbolju preciznost regresije na dva problema regresije pri obučavanju (California housing, Abalone) i najbolju generalizaciju na jednom problemu generalizacije pri testiranju (California housing), ELM ima najbolju preciznost regresije na jednom problemu regresije pri obučavanju (Machine CPU), dok FCRBFN ima najbolju preciznost na jednom problemu regresije pri obučavanju (Bank domains) i najbolju generalizaciju na tri problema generalizacije pri testiranju (Bank domains, Machine CPU, Abalone). Treba primijetiti i to da FCRBFN ima drugi najbolji rezultat u svim primjenama gdje nema najbolji rezultat. SVM i FCRBFN imaju najbolju stabilnost u svim primjenama, sa nultom standardnom devijacijom srednjekvadratne greške. Što se tiče vremena obučavanja i testiranja, FCRBFN je najbolja u svim kategorijama (tamo gdje nije najbrža, ima bolju stabilnost).

Tabela 5.2 *Srednja vrijednost i standardna devijacija srednjekvadratne greške i vremena izvršavanja svih aplikacija za obučavajuće i testne primjere - California housing*

Model	**Srednjekvadratna greška**($\times 10^{-1}$)		**Vrijeme izvršavanja** ($\times 10^{-3} s$)	
	Srednja vrijednost	**Standardna devijacija**	**Srednja vrijednost**	**Standardna devijacija**
SVM	1.1544	0.0000	2268.8381	13.2966
	1.2274	0.0000	787.5148	3.8949
ELM	1.2656	0.0112	31.7202	18.2298
	1.3338	0.0126	21.8401	28.0801
BP	1.6924	0.2549	337.4112	6.9884
	1.7376	0.2453	31.5232	0.4704
FCRBFN	1.2442	0.0000	11.6447	0.1974
	1.3106	0.0000	34.6456	0.2542

Tabela 5.3 *Srednja vrijednost i standardna devijacija srednjekvadratne greške i vremena izvršavanja svih aplikacija za obučavajuće i testne primjere - Bank domains*

Model	**Srednjekvadratna greška**($\times 10^{-2}$)		**Vrijeme izvršavanja** ($\times 10^{-3} s$)	
	Srednja vrijednost	**Standardna devijacija**	**Srednja vrijednost**	**Standardna devijacija**
SVM	7.0212	0.0000	415.4703	2.2944
	9.6518	0.0000	135.8010	0.7799
ELM	5.3435	0.2288	21.8401	25.3471
	5.5173	0.2426	3.1200	12.3493
BP	15.6450	1.9371	239.8690	11.2740
	15.8480	1.9163	25.7729	0.4551
FCRBFN	4.2987	0.0000	7.7812	2.3495
	4.6345	0.0000	10.4522	0.1094

Tabela 5.4 *Srednja vrijednost i standardna devijacija srednjekvadratne greške i vremena izvršavanja svih aplikacija za obučavajuće i testne primjere - Machine CPU*

Model	**Srednjekvadratna greška**($\times 10^{-2}$)		**Vrijeme izvršavanja** ($\times 10^{-3} s$)	
	Srednja vrijednost	**Standardna devijacija**	**Srednja vrijednost**	**Standardna devijacija**
SVM	4.8831 9.4833	0.0000 0.0000	0.2131 0.0860	0.0685 0.0549
ELM	1.6294 10.4860	0.1137 4.2943	1.5600 1.7012	8.4005 3.1547
BP	6.3354 11.757	1.4313 3.2363	97.1831 22.5431	2.4645 0.2108
FCRBFN	3.1299 4.5981	0.0000 0.0000	0.1652 0.1489	0.2127 0.0746

Tabela 5.5 *Srednja vrijednost i standardna devijacija srednjekvadratne greške i vremena izvršavanja svih aplikacija za obučavajuće i testne primjere - Abalone*

Model	**Srednjekvadratna greška**($\times 10^{-2}$)		**Vrijeme izvršavanja** ($\times 10^{-3} s$)	
	Srednja vrijednost	**Standardna devijacija**	**Srednja vrijednost**	**Standardna devijacija**
SVM	6.6005 7.9311	0.0000 0.0000	99.8450 19.3354	1.0303 0.3534
ELM	6.8346 7.7433	0.0310 0.0275	8.8400 2.0800	16.4521 7.7826
BP	10.4430 11.0490	2.3409 2.2384	153.5502 24.8397	4.2737 0.3970
FCRBFN	6.7921 7.6782	0.0000 0.0000	2.7118 4.9341	0.2241 0.1273

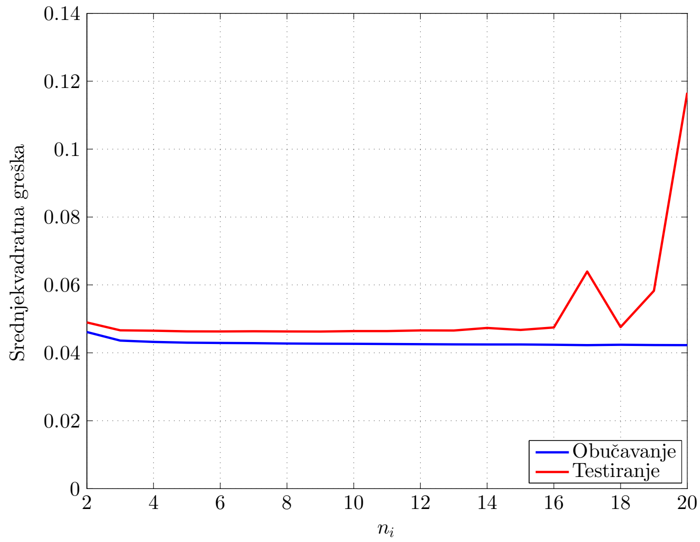
Ukratko, FCRBFN sa mnogo jednostavnijom strukturom i kompaktnijom mrežom od onih sa kojima je poređena postiže dobru preciznost regresije, generalizaciju, performanse, stabilnost i ponovljivost na većini funkcija, pa se može smatrati efektivnim alatom za mašinsko učenje.

5.5 Analiza osjetljivosti na izbor parametara klastera neurona

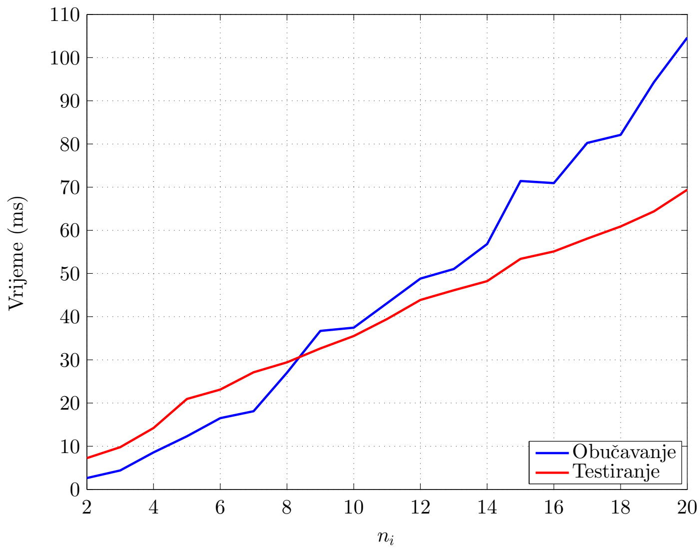
Da bi se pokazala osjetljivost FCRBFN na izbor parametara klastera neurona: broj neurona po klasteru n_i, maksimalno dozvoljeno proširenje opsega ex_i i odnos pokrivanja sr_i; sprovedeno je nekoliko eksperimenata, svi na skupu podataka *Bank domains* iz prethodnog potpoglavlja.

U prvom eksperimentu parametri ex_i i sr_i su postavljeni na fiksne vrijednosti 50% i 100% respektivno, a parametar n_i je variran od 2 do 20. Rezultati tog eksperimenta su prikazani na Slikama 5.2 i 5.3.

Kao što se može vidjeti sa tih slika, broj neurona po klasteru ne utiče značajno na srednjekvadratnu grešku u obučavanju, što je i očekivano. Ovo znači da je za sve rezultujuće arhitekture FCRBFN sistem (5.2) saglasan. Međutim, očigledno je i da povećanje broja neurona po klasteru negativno utiče na srednjekvadratnu grešku u testiranju. To je efekat prefitovanja. Očekivano, povećanje broja neurona po klasteru rezultuje u povećanju vremena izvršavanja i pri obučavanju i pri testiranju, kao posljedica povećanja matrica u (5.2).



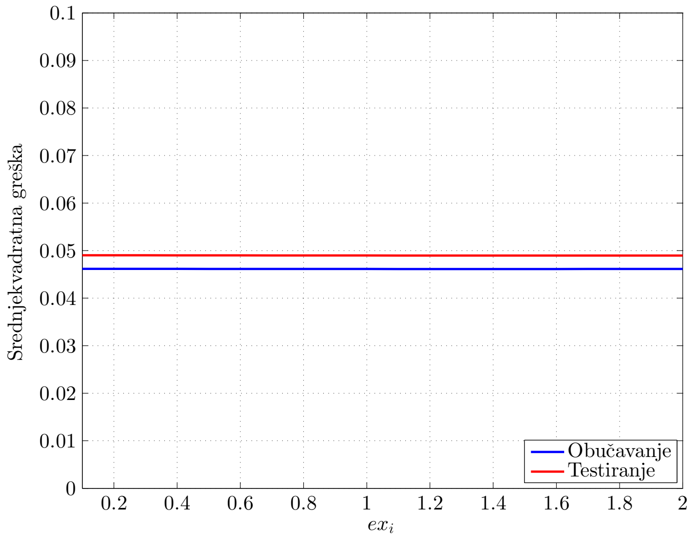
Slika 5.2 *Osjetljivost srednjekvadratne greške na promjenu parametra* n_i



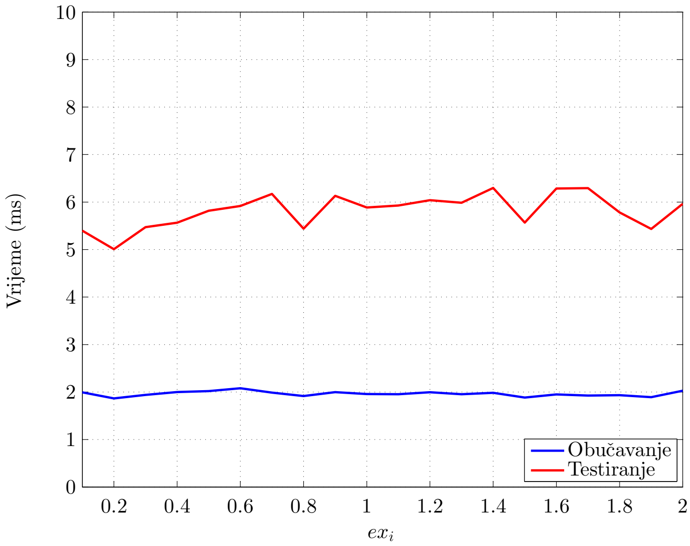
Slika 5.3 *Osjetljivost vremena izvršavanja na promjenu parametra* n_i

Drugi eksperiment je sproveden da bi se testirala osjetljivost na promjenu dozvoljenog proširenja opsega ex_i koje je mijenjano u opsegu $[10\%, 200\%]$, dok su parametri n_i i sr_i bili fiksirani na 2 i 100% respektivno. Rezultati tog eksperimenta su prikazani na Slikama 5.4 i 5.5.

Ovaj rezultat pokazuje da promjena tog parametra ne utiče značajno na bilo koji od indikatora. Ovo je očekivan rezultat jer su ulazne vrijednosti pri obučavanju dobro raspoređene po određenom opsegu, pa s obzirom na činjenicu da su ulazne vrijednosti pri testiranju iz istog opsega, proširenje opsega nema značajan efekat.



Slika 5.4 *Osjetljivost srednjekvadratne greške na promjenu parametra* ex_i

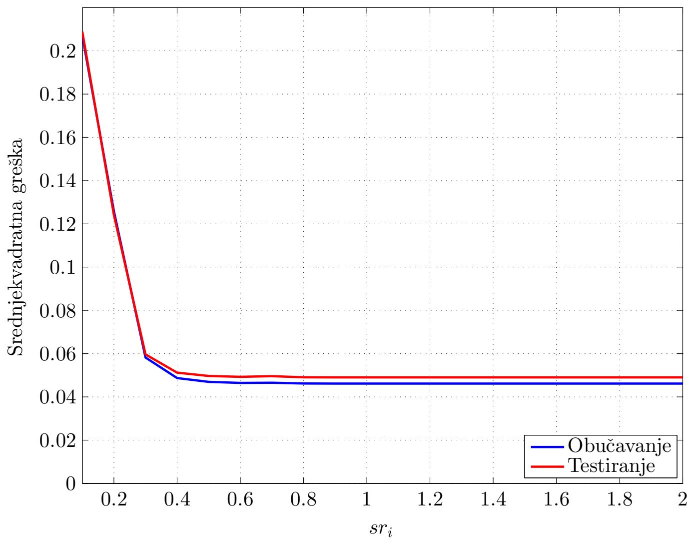


Slika 5.5 *Osjetljivost vremena izvršavanja na promjenu parametra* ex_i

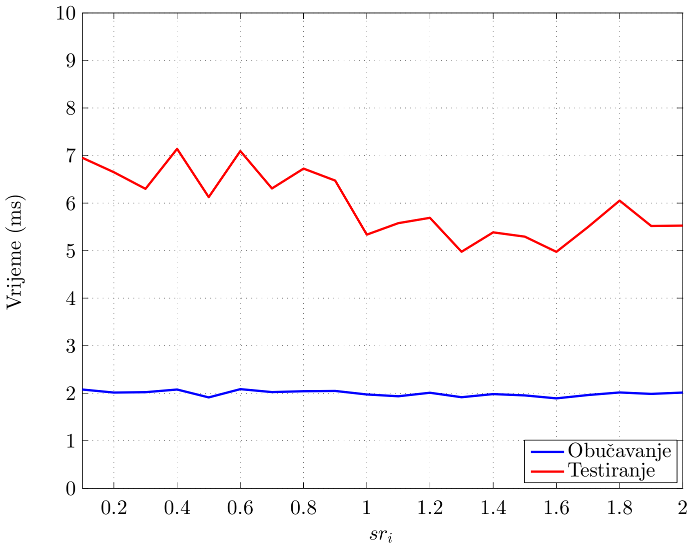
U trećem eksperimentu je provjeravana osjetljivost na promjenu parametra sr_i, koji je za te potrebe variran u opsegu $[10\%, 200\%]$, dok su parametri n_i i ex_i fiksirani na 2 i 50% respektivno. Rezultati tog eksperimenta su prikazani na Slikama 5.6 i 5.7.

Posljednji test pokazuje da povećanje parametra sr_i rezultuje u poboljšanju srednjekvadratne greške, a ne utiče značajno na brzinu izvršavanja. Prva posljedica je zbog činjenice da veći odnos znači bolje pokrivanje prostora između centara tako da dvije susjedne aktivacione funkcije (funkcije čiji su centri najmanje udaljeni) imaju veće preklapanje što povećava sposobnost regresije.

Ova analiza pokazuje da je izbor parametara klastera neurona prilično intuitivan. Pored toga, svi parametri mogu biti izabrani u relativno velikim opsezima bez značajnog uticaja na performanse mreže, što smanjuje vrijeme potrebno za pronalaženje pogodne konfiguracije mreže.



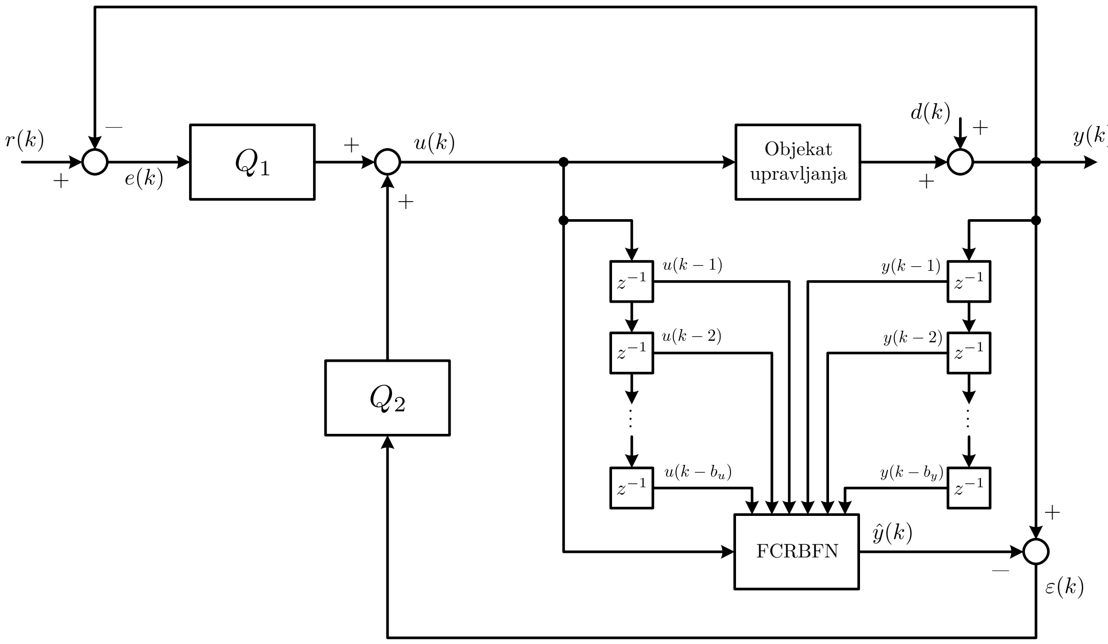
Slika 5.6 *Osjetljivost srednjekvadratne greške na promjenu parametra* sr_i



Slika 5.7 *Osjetljivost vremena izvršavanja na promjenu parametra* sr_i

6. Prediktivno upravljanje na bazi modela sa FCRBFN

Na Slici 6.1 je prikazan funkcionalni blok-dijagram sistema sa prediktivnim upravljanjem na bazi modela, pri čemu je za modelovanje objekta upravljanja, odnosno kao prediktor korištena FCRBFN. U prediktor je ugrađen i mehanizam adaptacije koji vrši dodavanje novog znanja kako je opisano u Potpoglavlju 5.2.



Slika 6.1 *Blok-dijagram sistema sa FCRBFN za prediktivno upravljanje na bazi modela*

U spregnutom sistemu postoje dva kontrolera Q_1 i Q_2 (u literaturi se sreću različite strukture, ali se sve mogu svesti na ovakvu). Prvi kontroler služi za postizanje nulte greške u ustaljenom stanju, i u skladu s tim najčešće se bira tako da uključuje integralno dejstvo. Drugi kontroler služi za podešavanje uticaja greške predikcije $\varepsilon(k) = y(k) - \hat{y}(k)$ na konačni upravljački signal $u(k)$, koji se dobija iz

$$u(k) = Q_1(z)e(k) + Q_2(z)\varepsilon(k), \tag{6.1}$$

gdje je $e(k) = r(k) - y(k)$ greška praćenja referentnog ulaza $r(k)$.

Prema Teoremi 4.3 datoj u Potpoglavlju 4.6 RBF mreže su univerzalni aproksimatori, odnosno postoji takva mreža koja će aproksimirati proizvoljnu funkciju do određene tačnosti. Na ovom se temelji očekivanje da, za povoljno izabranu strukturu mreže i dovoljno obučavajućih parova, prediktor sa FCRBFN može uspješno da vrši predikciju izlaznog signala sistema. U skladu sa ovim, tokom rada kontrolera *prikuplja* se dovoljno znanja da greška predikcije teži nuli. S obzirom na korištenje RLS za adaptaciju, odnosno doobučavanje mreže, a prema jednačinama (5.9), (5.10), (5.11) i (5.12), ukoliko greška predikcije teži nuli, onda i matrica greške **E** teži nula-matrici. Odatle slijedi da se, praktično, isključuje mehanizam adaptacije, jer je tad

$$\mathbf{W_{t+1}} = \mathbf{W_t}. \tag{6.2}$$

Ovo znači da, kad se postigne nulta greška predikcije, povratna grana sa kontrolerom Q_2 prestaje da doprinosi i da se dalje upravljanje izvodi isključivo na osnovu greške praćenja reference.

Treba naglasiti da, ako je objekat upravljanja nestabilan, ponekad treba uvesti još jedan kontroler (ili izvršiti transformaciju ovakve strukture) u kaskadu sa objektom upravljanja, tako da se izvrši stabilizacija sistema.

Kontroleri Q_1 i Q_2 uglavnom imaju funkcije prenosa niskog reda, a često su i sa fiksnim parametrima. Dodatno poboljšanje performansi se može postići tako da parametri bar jednog od kontrolera budu promjenljivi, tj. da se vrši njihova adaptacija na osnovu prediktora u toku samog rada sistema [149, 150].

6.1 Eksperimentalni rezultati

Model sistema je dat sa

$$y(k) = 0.5 \sin y(k-1) + 3u(k) + \frac{\sin u(k) y(k-1)}{1 + y^2(k-1)} + d(k). \tag{6.3}$$

Referentni signal je opisan sa

$$r(t) = \begin{cases} 0.6 & t \leq 100 \\ 1.0 & 100 < t \leq 210 \\ 2.0 & 210 < t \leq 400 \\ 1.0 & 400 < t \leq 610 \\ 2.0 & 610 < t \leq 700 \end{cases}$$

Poremećaj je opisan sa

$$d(t) = \begin{cases} 0.0 & t \leq 150 \\ 0.4 & 150 < t \leq 300 \\ 0.0 & 300 < t \leq 450 \\ 0.4 & 450 < t \leq 550 \\ 0.0 & 550 < t \leq 650 \\ 0.4 & 650 < t \leq 700 \end{cases}$$

FCRBFN ima 3 ulaza: prethodna i trenutna vrijednost upravljačkog signala, te prethodna vrijednost izlaznog signala. Svaki klaster neurona u skrivenom sloju se sastoji od 2 neurona, sa dozvoljenim proširenjem opsega ex_i od 50% širine opsega r_i, dok je odnos pokrivanja $sr_i = 1$. Ove vrijednosti su dobijene eksperimentalno.

Korišteni kontroleri su čisto integralnog tipa

$$Q_2(z) = Q_1(z) = \frac{0.05}{z-1}. \tag{6.4}$$

Kriterijum za izbor kontrolera sa fiksnim parametrima je bio jednostavnost, a eksperimentalno je pokazano da su performanse zadovoljavajuće.

Sa ovakvim kontrolerima zakon upravljanja je dat sa

$$U(z) = Q_1(z)E(z) + Q_2(z)\varepsilon(z) \tag{6.5}$$

$$U(z) = \frac{0.05}{z-1}E(z) + \frac{0.05}{z-1}\varepsilon(z) \tag{6.6}$$

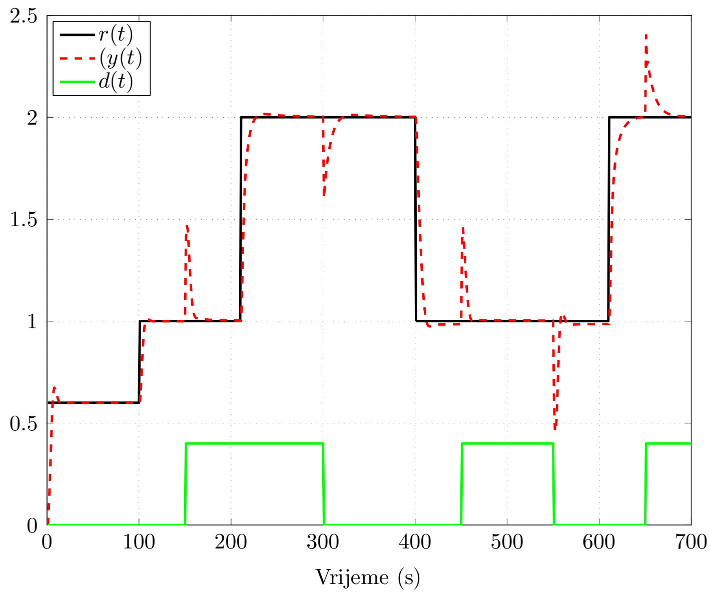
$$(z-1)U(z) = 0.05E(z) + 0.05\varepsilon(z) \tag{6.7}$$

$$zU(z) = U(z) + 0.05E(z) + 0.05\varepsilon(z) \tag{6.8}$$

$$u(k+1) = u(k) + 0.05e(k) + 0.05\varepsilon(k) \tag{6.9}$$

$$u(k+1) = u(k) + 0.05[r(k) - y(k)] + 0.05[y(k) - \hat{y}(k)] \tag{6.10}$$

Rezultati eksperimenta su prikazani na Slici 6.2.



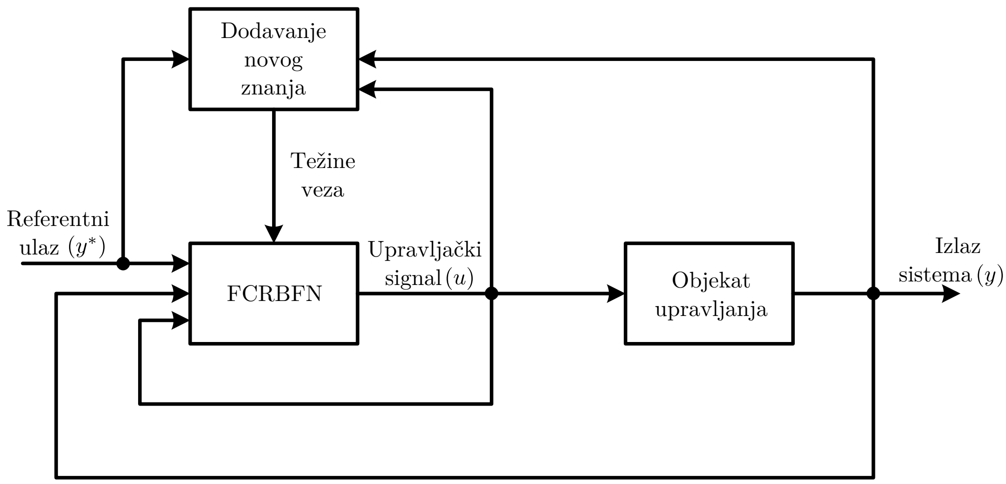
Slika 6.2 *Rezultati eksperimenta za prediktivno upravljanje na bazi modela*

Ovakvi rezultati pokazuju da se FCRBFN može uspješno koristiti kao prediktor u MPC strukturi i da je takva struktura robustna na prisustvo konstantnog poremećaja na izlazu sistema.

7. Adaptivno-prediktivni kontroler sa FCRBFN

Na Slici 7.1 je prikazan opšti funkcionalni blok-dijagram sistema sa adaptivnim upravljanjem. U ovom poglavlju kao adaptivno-prediktivni kontroler je korištena FCRBFN. U skladu s tim, kontroler na Slici 7.1 je konkretna FCRBFN koja koristi svoju sposobnost generalizacije za predikciju potrebnog upravljačkog signala da bi se postigla željena referentna vrijednost. Mehanizam adaptacije je blok koji vrši dodavanje novog znanja kako je opisano u Potpoglavlju 5.2. Otud i naziv *adaptivno-prediktivni kontroler na bazi neuralnih mreža*.

Treba naglasiti da u ovakvoj strukturi ne postoji izdvojen prediktor, tj. funkcionalni blok koji služi za predikciju izlaza, pomoću kojeg bi se, na osnovu greške predikcije, a u skladu sa određenom funkcijom cilja, u izdvojenom kontroleru (optimizatoru) vršilo generisanje upravljačkog signala. Struktura ovog kontrolera objedinjuje sve te funkcionalnosti u jednu neuralnu mrežu čiji su ulazi: vektor referentnih signala koje je od interesa dobiti i na izlazu objekta upravljanja, vektor stvarnih izlaza objekta upravljanja, te vektor upravljačkih signala. U takvoj postavci, mreža ne modeluje objekat upravljanja čime bi služila za predikciju izlaza sistema, već sadrži znanje o tome kakav upravljački signal rezultuje u kakvom izlazu sistema. Ovo znači da se prediktivne osobine takvog kontrolera ne koriste indirektno za određivanje zakona upravljanja, već direktno kroz predikciju potrebnog zakona upravljanja, ako je željeni izlaz dat.

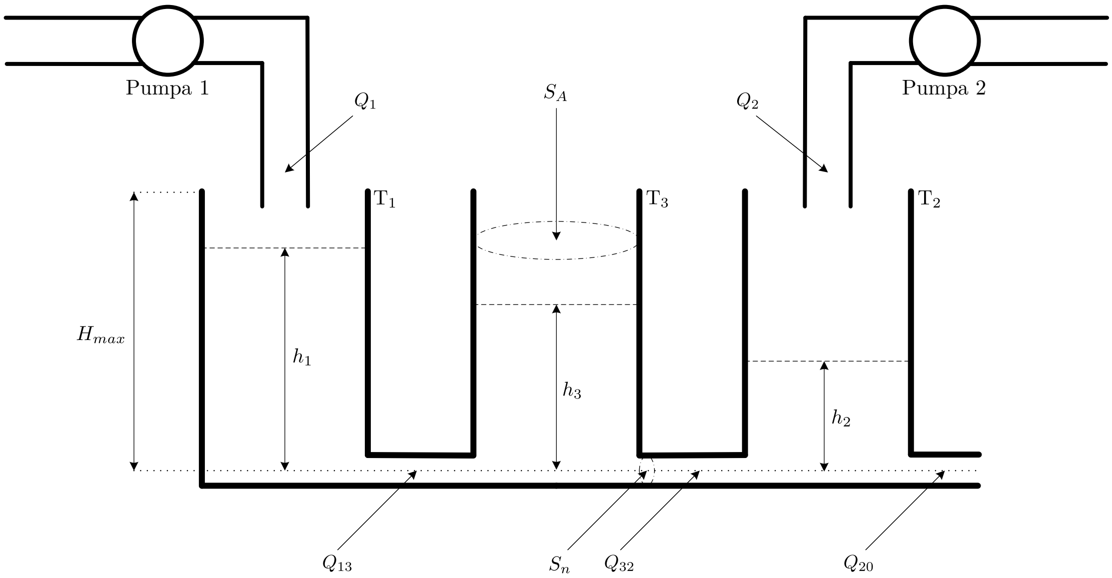


Slika 7.1 *Blok-dijagram sistema sa FCRBFN za adaptivno-prediktivno upravljanje*

Za dokazivanje mogućnosti ovakvog kontrolera, isti je primijenjen na *sistem sa* *tri rezervoara*. Matematički model tog sistema se u ovom slučaju koristi samo kao generator ulazno/izlaznih podataka koje će kontroleri koristiti, a nikakve informacije o modelu nisu korištene pri dizajnu samih kontrolera. FCRBFN se obučava sa jako malo informacija prije puštanja u rad (eng. *off-line*), većina podešavanja se izvodi pristupom adaptivnog upravljanja za vrijeme samog eksperimenta (eng. *on-line*).

7.1 Model sistema sa tri rezervoara

Principijelna struktura sistema sa tri rezervoara je prikazana na Slici 7.2. Sastoji se od 3 cilindrična rezervoara T_1, T_2 i T_3 čiji su svi poprečni presjeci jednake površine S_A. Ovi rezervoari su povezani serijski cijevima poprečnog presjeka površine S_n. Jedan izlazni ventil poprečnog presjeka površine S_n je vezan za rezervoar T_2. Tečnost koja teče kroz sistem (najčešće destilovana voda) se skladišti u centralni rezervoar iz kojeg se snabdijevaju pumpe 1 i 2, koje dalje ubacuju vodu u rezervoare T_1 i T_2 respektivno, što sistem čini zatvorenim i omogućava da tečnost cirkuliše. Uobičajeni zadatak sistema je da dostigne i održava nivoe tečnosti u rezervoarima T_1 i T_2 (ili svim rezervoarima) na zadatoj vrijednosti tako što se mijenja protok tečnosti iz pumpi.



Slika 7.2 *Šematski prikaz sistema sa tri rezervoara*

Definišu se sljedeće promjenljive i parametri: α_i - koeficijent izlaznog toka i-tog rezervoara (bez dimenzije, realan broj iz opsega $[0, 1]$); h_1, h_2, h_3 - nivoi tečnosti u odgovarajućim rezervoarima (m); Q_{13} - protok tečnosti iz rezervoara 1 u rezervoar 3 ($\frac{m^3}{s}$); Q_{32} - protok tečnosti iz rezervoara 3 u rezervoar 2 ($\frac{m^3}{s}$); Q_{20} - protok tečnosti iz rezervoara 2 u centralni rezervoar ($\frac{m^3}{s}$); Q_1, Q_2 - protok tečnosti koju dovode pumpe ($\frac{m^3}{s}$); S_A - površina poprečnog presjeka cilindara (m^2); S_n - površina poprečnog presjeka povezne cijevi (m^2); g - ubrzanje Zemljine gravitacije.

Dinamika sistema sa tri rezervoara je opisana u nastavku [151].

$$S_A \frac{dh_1}{dt} = Q_1(t) - Q_{13}(t) \tag{7.1}$$

$$S_A \frac{dh_3}{dt} = Q_{13}(t) - Q_{32}(t) \tag{7.2}$$

$$S_A \frac{dh_2}{dt} = Q_2(t) + Q_{32}(t) - Q_{20}(t) \tag{7.3}$$

$$Q_{13}(t) = \alpha_1 S_n \mathrm{sgn}\left(h_1(t) - h_3(t)\right) \sqrt{2g\left|h_1(t) - h_3(t)\right|} \tag{7.4}$$

$$Q_{32}(t) = \alpha_3 S_n \mathrm{sgn}\left(h_3(t) - h_2(t)\right) \sqrt{2g\left|h_3(t) - h_2(t)\right|} \tag{7.5}$$

$$Q_{20}(t) = \alpha_2 S_n \mathrm{sgn}\left(h_2(t)\right) \sqrt{2g\left|h_2(t)\right|} \tag{7.6}$$

7.2 Eksperimentalni rezultati

Parametri sistema sa tri rezervoara koji su korišteni u eksperimentima su dati u Tabeli 7.1.

Tabela 7.1 *Parametri sistema sa tri rezervoara*

S_A	S_n	H_{max}	Q_{max}	α_1	α_2	α_3
$0.0154m^2$	$5 \cdot 10^{-5}m^2$	$0.6m$	$0.0001\frac{m^3}{s}$	0.22	0.28	0.27

Ulazni upravljački signali su protoci tečnosti iz pumpi Q_1 i Q_2, izlazni signali su nivoi tečnosti h_1 i h_2.

Početni uslovi su $h_1(0) = 0$, $h_2(0) = 0$, $h_3(0) = 0$, $Q_1(0) = 0$ i $Q_2(0) = 0$.

Željeni nivoi tečnosti su definisani sa

$$h_1^*(t) = \begin{cases} 0.15 & t \leq 500 \\ 0.3 & 500 < t \leq 1000 \\ 0.15 & 1000 < t \leq 1500 \end{cases}$$

$$h_2^*(t) = \begin{cases} 0.2 & t \leq 300 \\ 0.4 & 300 < t \leq 700 \\ 0.2 & 700 < t \leq 1200 \\ 0.05 & 1200 < t \leq 1500 \end{cases}$$

FCRBFN ima 4 ulaza: 2 za trenutna (prethodna) stanja i 2 za željena (buduća) stanja nivoa tečnosti h_1 i h_2; i 2 izlaza: protoke tečnosti iz pumpi Q_1 i Q_2. Ovo je urađeno kako bi mreža bila korištena za predikciju vrijednosti Q_1 i Q_2 koje su potrebne za promjenu stanja nivoa tečnosti h_1 i h_2 od trenutnih na željena (referentna). Svaki klaster neurona u skrivenom sloju se sastoji od 2 neurona, sa dozvoljenim proširenjem opsega ex_i od 50% širine opsega r_i, dok je odnos pokrivanja $sr_i = 1$. Ove vrijednosti su dobijene eksperimentalno.

Za verifikaciju efektivnosti i efikasnosti predloženog pristupa izvršeno je poređenje sa PID upravljačkom metodom, te metodom pod nazivom *podacima vođena adaptivna upravljačka metoda bez modela* (PFDL-MFAC). PID kontroler za sistem sa tri rezervoara je proučavan u [152], kako je predstavljeno u nastavku.

$$\begin{bmatrix} u_1(t+1) \\ u_2(t+1) \end{bmatrix} = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} + K \times E \tag{7.7}$$

$$K = \begin{bmatrix} K_{P11} & K_{I11} & K_{D11} & K_{P12} & K_{I12} & K_{D12} \\ K_{P11} & K_{I11} & K_{D11} & K_{P12} & K_{I12} & K_{D12} \end{bmatrix} \tag{7.8}$$

$$E = \begin{bmatrix} e_1(k) - e_1(k-1) \\ e_1(k) \\ e_1(k) - 2e_1(k-1) + e_1(k-2) \\ e_2(k) - e_2(k-1) \\ e_2(k) \\ e_2(k) - 2e_2(k-1) + e_2(k-2) \end{bmatrix} \tag{7.9}$$

gdje su $u_1(t)$ i $u_2(t)$ upravljački signali, a $e_1(t)$ i $e_2(t)$ greške praćenja reference za rezervoare 1 i 2 respektivno. K_{Pij}, K_{Iij} i K_{Dij} su parametri PID kontrolera.

Glavni problem kod PID upravljanja je pronalaženje odgovarajućih vrijednosti PID parametara. U [152] je korištena hibridna strategija, tj. kombinacija genetičkog algoritma [153] i *big bang-big crunch* [154] algoritma za pronalaženje optimalnih parametara PID kontrolera. Za eksperimente su izabrani sljedeći parametri [155]

$$\begin{aligned} K_{P11} &= 0.00512; & K_{I11} &= 0.00063; & K_{D11} &= 0.00721; \\ K_{P12} &= 0.00131; & K_{I12} &= 0.00005; & K_{D12} &= 0.00130; \\ K_{P21} &= 0.00267; & K_{I21} &= 0.00027; & K_{D21} &= 0.00242; \\ K_{P22} &= 0.00985; & K_{I22} &= 0.00101; & K_{D22} &= 0.00642. \end{aligned}$$

PFDL-MFAC kontroler za sistem sa tri rezervoara je proučavan u [155], kako je dato u nastavku.

Neka je sa $\mathbf{u}(k)$ označen vektor trenutnih vrijednosti upravljačkih signala (ulaz kontrolera). Neka je sa

$$\overline{\mathrm{U}}(k-1) = \begin{bmatrix} \mathbf{u}(k-1) & \mathbf{u}(k-2) & \cdots & \mathbf{u}(k-L) \end{bmatrix}$$

označena memorijska matrica upravljačkih signala (posljednjih L stanja upravljačkih signala). Neka je sa

$$\mathbf{\Phi}(k) = \begin{bmatrix} \mathbf{\Phi}_1(k) & \mathbf{\Phi}_2(k) & \cdots & \mathbf{\Phi}_L(k) \end{bmatrix}$$

označena matrica trenutnih pseudo-parcijalnih izvoda (eng. *Pseudo-Partial Derivate - PPD*) koja se sastoji od L matrica $\mathbf{\Phi}_i(k)$ $(i = 1, \ldots, L)$ vezanih za trenutno i $L-1$ prethodnih stanja. Neka $\mathbf{y}(k)$ označava vektor trenutnih izlaza sistema, a ${}^*\mathbf{y}(k+1)$ vektor željenih izlaza sistema.

Neka $\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$ označava promjenu upravljačkih signala u odnosu na prethodni korak. Neka $\Delta\overline{\mathrm{U}}(k-1) = \overline{\mathrm{U}}(k-1) - \overline{\mathrm{U}}(k-2)$ označava promjenu memorijske matrice upravljačkih signala u odnosu na prethodni korak. Neka $\Delta\mathbf{y}(k) = \mathbf{y}(k) - \mathbf{y}(k-1)$ označava promjenu izlaza sistema u odnosu na prethodni korak.

Tada se PFDL-MFAC šema konstruiše na sljedeći način:

$$\mathbf{\Phi}_c = \frac{\eta \left(\Delta\mathbf{y}(k) - \mathbf{\Phi}(k-1)\Delta\overline{\mathrm{U}}(k-1) \right) \Delta\overline{\mathrm{U}}^{\mathrm{T}}(k-1)}{\mu + \|\Delta\overline{\mathrm{U}}^{\mathrm{T}}(k-1)\|^2} \tag{7.10}$$

$$\mathbf{\Phi}(k) = \mathbf{\Phi}(k-1) + \mathbf{\Phi}_c \tag{7.11}$$

$$\mathbf{u}_{c1} = \frac{\rho_1 \mathbf{\Phi}_1^{\mathsf{T}}(k)\left({}^{*}\mathbf{y}(k+1) - \mathbf{y}(k)\right)}{\lambda + \|\mathbf{\Phi}_1(k)\|^2} \tag{7.12}$$

$$\mathbf{u}_{c2} = \frac{\mathbf{\Phi}_1^{\mathsf{T}}(k) \sum_{i=2}^{L} \rho_i \mathbf{\Phi}_i(k) \Delta\mathbf{u}(k-i+1)}{\lambda + \|\mathbf{\Phi}_1(k)\|^2} \tag{7.13}$$

$$\mathbf{u}(k) = \mathbf{u}(k-1) + \mathbf{u}_{c1} - \mathbf{u}_{c2} \tag{7.14}$$

gdje je $\eta \in (0,1), \rho_1, \ldots, \rho_L$ niz veličina koraka, a λ i μ težinske konstante.

Po izračunavanju novih vrijednosti PPD matrice ona mora ostati dijagonalno dominantna što se osigurava vraćanjem svih vrijednosti koje ne ispunjavaju taj uslov na njihove početne vrijednosti.

U skladu sa [155], parametri PFDL-MFAC kontrolera[1] su $L = 2$, $\rho_1 = \rho_2 = 0.1$, $\lambda = \eta = 1$, $\mu = 0.0001$ i $\mathbf{\Phi}(1) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$.

Izvršena su dva eksperimenta: bez šuma i sa šumom mjerenja.

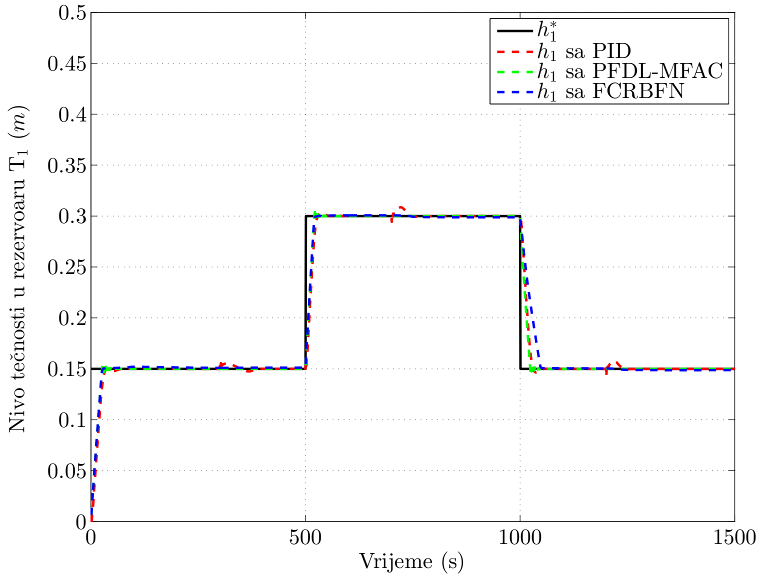
7.2.1 Eksperiment bez šuma mjerenja

Za potrebe ovog eksperimenta, razmatrana je situacija u kojoj nema šuma mjerenja. Uporedni rezultati eksperimenata sa FCRBFN, PID i PFDL-MFAC kontrolerima su prikazani na Slikama 7.3 i 7.4, pri čemu Slika 7.3a prikazuje nivo tečnosti u rezervoaru 1, Slika 7.3b prikazuje nivo tečnosti u rezervoaru 2, Slika 7.4a prikazuje protok tečnosti kroz pumpu 1, a Slika 7.4b prikazuje protok tečnosti kroz pumpu 2.

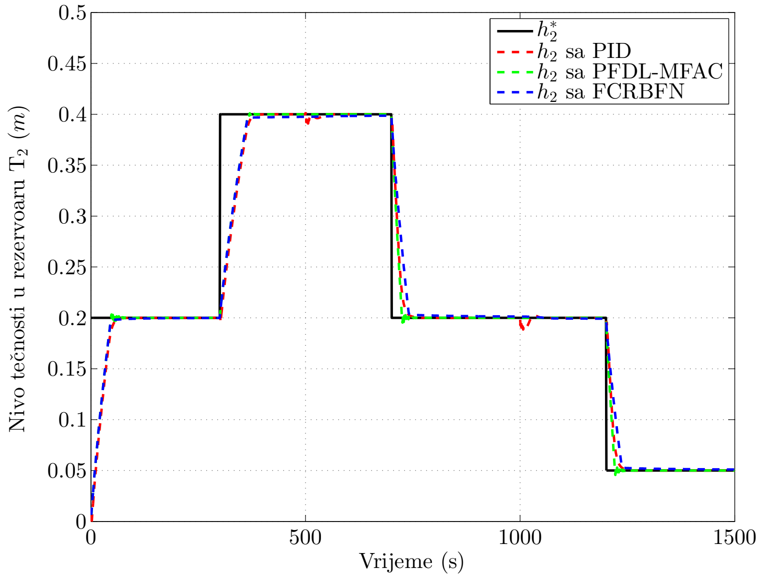
Iz ovih rezultata je očigledno da sva tri upravljačka metoda izvršavaju zadatak uspješno. Međutim, treba primijetiti i sljedeće:

- Algoritam PID kontrolera ima jasnu prednost po pitanju matematičke kompleksnosti i u odnosu na PFDL-MFAC kontroler, i u odnosu na FCRBFN kontroler, čije su procedure adaptacije podjednako kompleksne;
- Procedura podešavanja PID parametara oduzima mnogo više vremena i u odnosu na PFDL-MFAC kontroler, i u odnosu na FCRBFN kontroler, jer je potreban veliki broj eksperimenata za to;
- Podešavanja PID kontrolera se vrši off-line, i zahtijeva proceduru prikupljanja ulazno-izlaznih podataka, dok se PFDL-MFAC i FCRBFN kontroleri baziraju na on-line adaptivnom upravljačkom pristupu;

[1] Citirani rad zapravo predlaže da vrijednosti parametara ρ_1 i ρ_2 budu 0.5, ali eksperimenti sprovedeni za potrebe ovog rada pokazuju bolje performanse kontrolera za navedene vrijednosti.



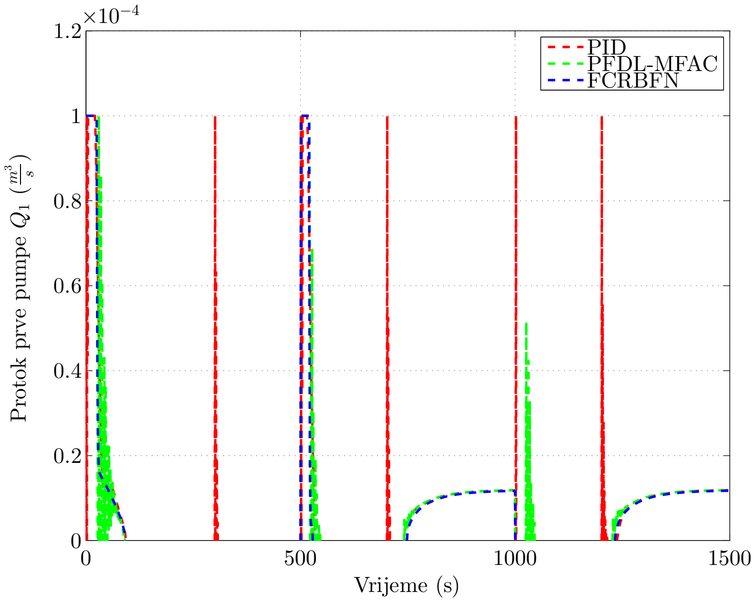
(a)



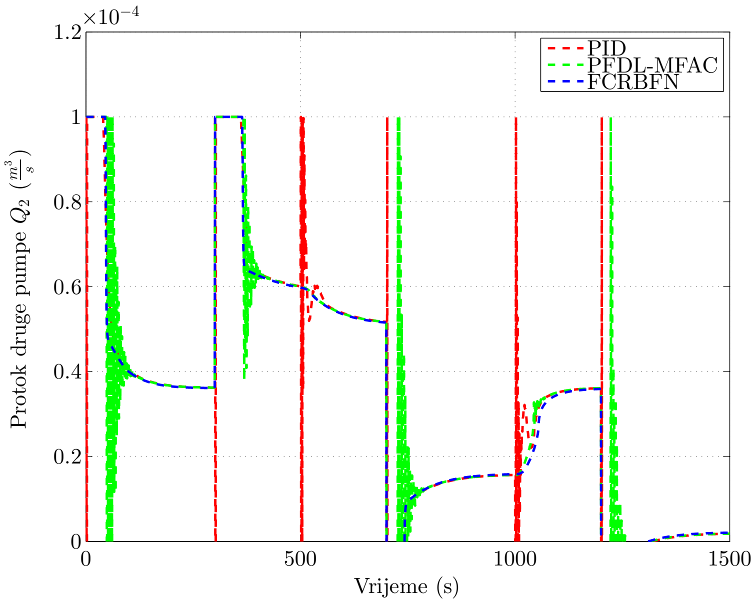
(b)

Slika 7.3 *Rezultati eksperimenata na sistemu sa tri rezervoara; (a) Nivo tečnosti u rezervoaru* T_1*; (b) Nivo tečnosti u rezervoaru* T_2

- PID kontroler ne uspijeva da kompenzuje smetnje jednako uspješno kao FCRBNF i PFDL-MFAC kontroleri u trenucima vremena kad se jedan od referentnih ulaza naglo mijenja;
- Parametri PFDL-MFAC kontrolera nisu intuitivni kao parametri FCRBFN kontrolera što čini analizu osjetljivosti na izbor tih parametara mnogo težom, uz dodatno otežavanje zbog činjenice da parametara ima mnogo, a među njima neki su i matrice.



(a)



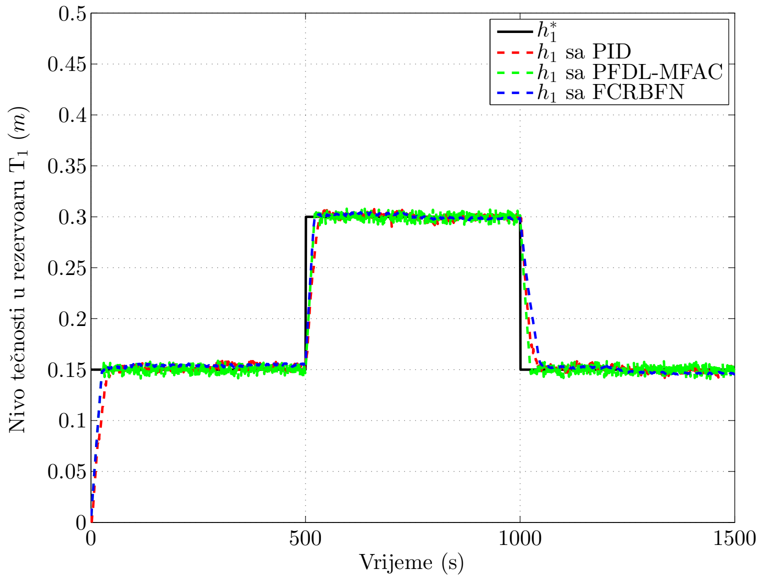
(b)

Slika 7.4 *Rezultati eksperimenata na sistemu sa tri rezervoara; (a) Protok tečnosti kroz prvu pumpu* Q_1*; (b) Protok tečnosti kroz drugu pumpu* Q_2

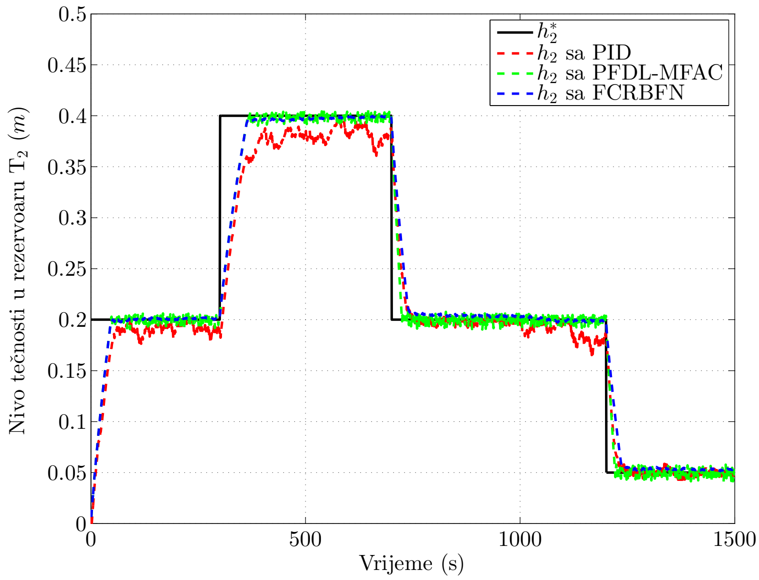
- Upravljački izlazi koje daje PFDL-MFAC kontroler imaju tendenciju oscilovanja što je nepoželjno za aktuatore i eksperimenti pokazuju da ovo rezultuje u preskoku ili u oscilacijama oko referentne vrijednosti ako ista traje dovoljno dugo (ove oscilacije se mogu prigušiti promjenom parametara ρ_1 i ρ_2, ali to dovodi do preskoka, tako da je potrebno pronaći kompromis između te dvije posljedice, što produžava vrijeme izbora parametara), a takođe i rezultuje u većem vremenu smirenja nego kod FCRBFN kontrolera.

7.2.2 Eksperiment kad je prisutan šum mjerenja

Pri izvođenju ovog eksperimenta, razmatrana je situacija u kojoj je prisutan šum mjerenja. Konkretno riječ je o šumu pseudoslučajnog karaktera koji unosi grešku mjerenja u opsegu $[-3mm, 3mm]$. Uporedni rezultati eksperimenata sa FCRBFN, PID i PFDL-MFAC kontrolerima su prikazani na Slikama 7.5 i 7.6. Iz ovih rezultata se vidi da FCRBFN i PFDL-MFAC kontroleri izvršavaju zadatak uspješno, dok PID kontroler unosi značajnu grešku praćenja reference.

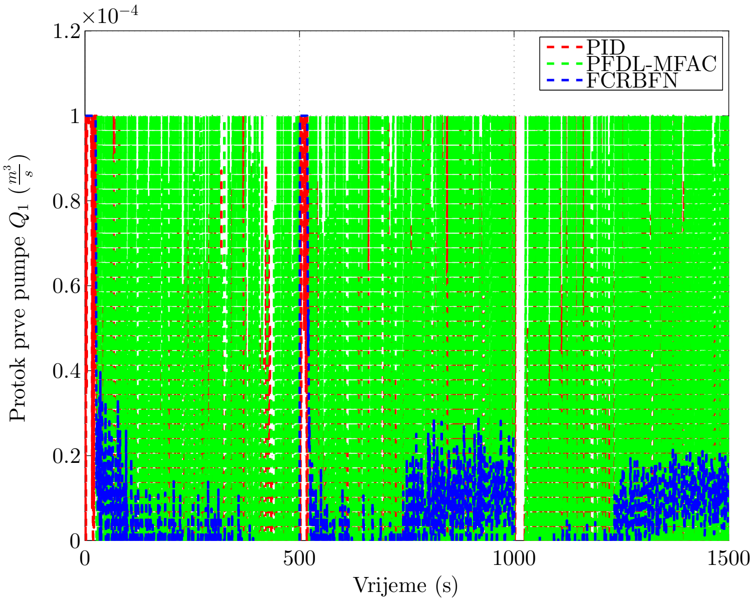


(a)

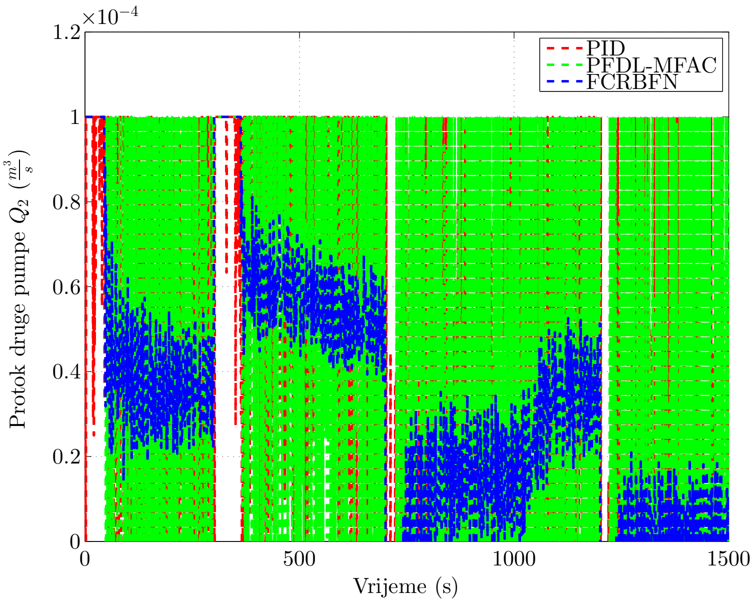


(b)

Slika 7.5 *Rezultati eksperimenata na sistemu sa tri rezervoara kad je prisutan šum mjerenja; (a) Nivo tečnosti u rezervoaru* T_1*; (b) Nivo tečnosti u rezervoaru* T_2



(a)



(b)

Slika 7.6 *Rezultati eksperimenata na sistemu sa tri rezervoara kad je prisutan šum mjerenja; (a) Protok tečnosti kroz prvu pumpu Q_1; (b) Protok tečnosti kroz drugu pumpu Q_2*

Sve opservacije iz prethodnog eksperimenta važe i za ovaj, ali je jasno da jedino FCRBFN kontroler eliminiše vibracije nastale kao posljedica šuma mjerenja. Ovu osobinu FCRBFN kontroler inherentno dobija zbog korištenja LMS i RLS metoda za obučavanje.

Prema svemu prikazanom kroz rezultate ova dva eksperimenta, može se reći da je FCRBFN kontroler po većini kriterijuma ili jednak ili bolji od ostala dva ispitana kontrolera.

8. Zaključak

O značaju upravljanja bilo je riječi u uvodnim poglavljima rada. Stoga ne čudi da postoji obimna literatura na temu upravljanja, pa i za svaku od podoblasti ponaosob. U uvodnom poglavlju je dat i pregled nekih od pravaca u kojima se nauka o upravljanju razvija. U današnjem svijetu nauke logična je i simbioza naizgled raznorodnih naučnih disciplina. Jedan od takvih primjera je i primjena vještačkih neuralnih mreža u upravljanju. Uvodno poglavlje daje i pregled ideja kojima se istraživači danas vode na tom polju. Činjenica da se i dalje mnogo radi na razvoju, kako vještačkih neuralnih mreža, tako i upravljanja i njihove sprege, poslužila je kao motivacija za istraživanje koje je dovelo do ove disertacije.

Kroz Poglavlje 2 dat je pregled osnovnih principa opisivanja, ispitivanja stabilnosti, te upravljanja linearnim i nelinearnim sistemima, kao i naprednijih tehnika upravljanja poput adaptivnog u Potpoglavlju 2.5, te prediktivnog u Potpoglavlju 2.6. Poglavlje 3 je posvećeno vještačkim neuralnim mrežama, a Poglavlje 4 jednoj posebnoj klasi vještačkih neuralnih mreža - RBF mrežama. Ova tri poglavlja zajedno služe za postavljanje temelja za jednu vrstu kontrolera koji spreže RBF mreže sa adaptivnim i prediktivnim upravljanjem za koji je ideja nastala kroz istraživanje aktuelnih tehnologija opisanih u uvodnom poglavlju.

U poglavlju 5 predstavljena je nova arhitektura neuralne mreže koja se obučava bez iterativnog metoda, kako je pokazano u Potpoglavlju 5.2, a pokazuje dobre aproksimacione sposobnosti, kako je pokazano u Potpoglavlju 5.4, što je čini izuzetno pogodnom za korištenje i u adaptivnom i u prediktivnom upravljanju, kako je pokazano na primjerima različitih strategija upravljanja u Poglavljima 6 i 7. Ona takođe omogućava da grupa neurona koji su povezani sa jednim ulazom imaju centre raspoređene nezavisno od vrijednosti drugih ulaza, što umnogom pojednostavljuje rješavanje problema obučavanja koje imaju standardne RBF mreže. U Potpoglavlju 5.5 je pokazano da je izbor parametara FCRBFN mreže intuitivan, a i da je mreža prilično robustna po pitanju promjene tih parametara. Ovim je dokazana postavljena radna hipoteza, što je i najveći doprinos ove disertacije.

Za ilustraciju sposobnosti FCRBFN mreže da obavlja ulogu prediktora u prediktivnom upravljanju na bazi modela sproveden je simulacioni eksperiment na nelinearnom sistemu koji se često koristi u literaturi, a sve u prisustvu poremećaja tipa odskočne funkcije. Rezultati simulacija dati u Poglavlju 6 pokazuju da se FCRBFN mreža može uspješno koristiti kao prediktor u takvoj strukturi, te da je cijela struktura vrlo robustna.

Za pokazivanje mogućnosti kontrolera sa FCRBFN mrežom kao jedinstvene upravljačke jedinice koja ima i adaptivna i prediktivna svojstva sprovedena su dva eksperimenta na standardnom testnom uređaju - sistemu sa tri rezervoara. Kako je pomenuti sistem prisutan u mnogim laboratorijama, uključujući i Laboratoriju za Automatiku Elektrotehničkog fakulteta Univerziteta u Banjoj Luci, a često se koristi i u naučnim radovima za testiranje različitih metoda upravljanja, može se reći da čini dobru platformu za testiranje. Izvedena su dva eksperimenta sa istom postavkom i istim zahtjevima, jedan bez, drugi sa šumom mjerenja, kako je predstavljeno u Poglavlju 7. Rezultati pokazuju da kontroler sa FCRBFN strukturom postiže bolje rezultate od druga dva kontrolera sa kojima su izvršeni testovi pod istim uslovima (najčešće korišten industrijski kontroler - PID, te PFDL-MFAC), naročito u prisustvu šuma mjerenja, što je i realna situacija. Time je dokaz radne hipoteze upotpunjen i eksperimentalnim dokazima.

Dalji rad bi mogao da se kreće u pravcu proširenja sposobnosti adaptacije mreže i na strukturu, te na parametre aktivacionih funkcija neurona. Ovo je vrlo izazovno po pitanju istraživanja postojećih mehanizama adaptacije, te razvoja novih, a koji bi bili podjednako efikasni kao ovako predloženo rješenje. Posebno bi bilo zanimljivo razmotriti korištenje drugih metoda iz oblasti vještačke inteligencije u rješavanju takvog optimizacionog problema (evolutivni algoritmi, inteligencija roja i sl.).

Drugi mogući pravac daljeg istraživanja je izvođenje matematičkih dokaza stabilnosti i nulte greške ustaljenog režima za određene klase nelinearnih i linearnih sistema sa i bez vanjskih poremećaja, upravljanih kontrolerom sa FCRBFN.

Bibliografija

[1] P. J. Werbos, "Neural networks for control and system identification," *Proceedings of 28th IEEE Conference on Decision and Control*, 1989.

[2] K. S. Narendra and F. L. Lewis, "Introduction to the special issue on neural network feedback control," *Automatica*, vol. 37, pp. 1147–1148, 2001.

[3] P. J. Werbos, "A menu of designs for reinforcement learning over time," *Neural Networks for Control*, pp. 67–95, 1991.

[4] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," *Handbook of Intelligent Control*, 1992.

[5] M. Kawato, "Computational schemes and neural network models for formation and control of multijoint arm trajectory," *Neural Networks for Control*, pp. 197–228, 1991.

[6] A. Draeger, S. Engell, and H. Ranke, "Model predictive control using neural networks," *IEEE Control Systems*, vol. 15, no. 5, pp. 61–66, 1995.

[7] J. Thibault and B. P. A. Grandjean, "Neural networks in process control: A survey," *Advanced Control of Chemical Processes*, no. 8, pp. 251–260, 1992.

[8] M. J. Willis, G. A. Montague, C. DiMassimo, M. T. Than, and A. J. Morris, "Artificial neural network based predictive control," *Advanced Control of Chemical Processes*, no. 8, pp. 261–266, 1992.

[9] J. MacMurray and D. Himmelblau, "Identification of packed distillation column for control via artificial neural network," *Proceedings of the American Control Conference*, pp. 1455–1459, 1992.

[10] M. M. Božić, J. Igić, and I. Krčmar, "Adaptivno prediktivno upravljanje nelinearnim sistemima," *Zbornik radova XLVI Konferencije za ETRAN*, vol. 1, pp. 197–200, 2002.

[11] M. Hagan, H. Demuth, and O. De Jesus, "An introduction to the use of neural networks in control systems," *International Journal of Robust and Nonlinear Control*, vol. 12, no. 11, pp. 959–985, 2002.

[12] S. S. Ge, C. Yang, and T. H. Lee, "Adaptive predictive control using neural network for a class of pure-feedback systems in discrete time," *IEEE Transactions on Neural Networks*, vol. 19, no. 9, pp. 1599–1614, 2008.

[13] H. J. Sussmann, "Uniqueness of the weights for minimal feedforward nets with a given input-output map," *Neural Networks*, vol. 5, pp. 589–593, 1992.

[14] F. Albertini and E. D. Sontag, "For neural nets, function determines form," *Proceedings of 31th IEEE Conference on Decision and Control*, pp. 26–31, 1992.

[15] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, pp. 4–27, 1990.

[16] K. S. Narendra and K. Parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 252–262, 1991.

[17] N. Sadegh, "A perceptron network for functional identification and control of nonlinear systems," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 982–988, 1993.

[18] J. Leitner, A. J. Calise, and J. V. R. Prasad, "Analysis of adaptive neural networks for helicopter flight control," *Journal of Guidance Control and Dynamics*, vol. 20, no. 5, pp. 972–979, 1997.

[19] J. McFarland, A. J. Calise, and J. V. R. Prasad, "Adaptive nonlinear control of agile anti-air missiles using neural networks," *IEEE Transactions on Control Systems Technologies*, vol. 8, no. 5, pp. 749–756, 2000.

[20] A. J. Calise, N. Hovakimyan, and H. Lee, "Adaptive output feedback control of nonlinear systems using neural networks," *Automatica*, vol. 37, no. 8, pp. 1201–1211, 2001.

[21] A. Yesildirek and F. L. Lewis, "Feedback linearization using neural networks," *Automatica*, vol. 31, no. 11, pp. 1659–1664, 1995.

[22] T. Zhang, C. C. Hang, and S. S. Ge, "Robust adaptive controller for general nonlinear systems using multilayer neural networks." preprint, 2001.

[23] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems.* Taylor and Francis, 1999.

[24] G. Arslan and T. Basar, "Disturbance attenuating controller design for strict-feedback systems with structurally unknown dynamics," *Automatica*, vol. 37, no. 8, pp. 1175–1188, 2001.

[25] J. Q. Gong and B. Yao, "Neural network adaptive robust control of nonlinear systems in semi-strict feedback form," *Automatica*, vol. 37, no. 8, pp. 1149–1160, 2001.

[26] J. Wang and J. Huang, "Neural network enhanced output regulation in nonlinear systems," *Automatica*, vol. 37, no. 8, pp. 1189–1200, 2001.

[27] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 837–863, 1992.

[28] M. M. Polycarpou and P. A. Ioannou, "Identification and control using neural network models: design and stability analysis," 1991.

[29] M. M. Polycarpou and P. A. Ioannou, "Neural networks as on-line approximators of nonlinear systems," *Proceedings of 31th IEEE Conference on*

Decision and Control, pp. 7–12, 1992.

[30] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 703–715, 1995.

[31] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Transactions on Automatic Control*, vol. AC-32, no. 2, pp. 134–145, 1987.

[32] K. S. Narendra and A. M. Annaswamy, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 930–945, 1993.

[33] F.-C. Chen and H. K. Khalil, "Adaptive control of nonlinear systems using neural networks," *International Journal of Control*, vol. 55, no. 6, pp. 1299–1317, 1992.

[34] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 3, pp. 400–412, 1994.

[35] A. S. Poznyak, W. Yu, E. N. Sanchez, and J. P. Perez, "Nonlinear adaptive trajectory tracking using dynamic neural networks," *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1402–1411, 1999.

[36] G. A. Rovithakis, "Performance of a neural adaptive tracking controller for multi-input nonlinear dynamical systems," *IEEE Transactions on Systems, Man and Cybernetics Part A*, vol. 30, no. 6, pp. 720–730, 2000.

[37] Y. Zhang and J. Wang, "Recurrent neural networks for nonlinear output regulation," *Automatica*, vol. 37, no. 8, pp. 1161–1173, 2001.

[38] M. R. Azizi-Sadjadi and L. Ren-Jean, "Fast learning process of multilayer neural networks using recursive least squares method," *IEEE Transactions on Signal Processing*, vol. 40, pp. 446–450, 1992.

[39] E. Castillo, B. Guijarro-Berdinas, O. Fontenla-Romero, and A. Alonso-Betanzos, "A very fast learning method for neural networks based on sensitivity analysis," *Journal of Machine Learning Research*, vol. 7, pp. 1159–1182, 2006.

[40] T. W. S. Chow, J. Y.-F. Yam, and S.-Y. Cho, "Fast training algorithm for feedforward neural networks: application to crowd estimation at underground stations," *Artificial Intelligence in Engineering*, vol. 13, pp. 301–307, 1999.

[41] M. Jiang, G. Gielen, B. Zhang, and Z. Luo, "Fast learning algorithms for feedforward neural networks," *Applied Intelligence*, vol. 18, pp. 37–54, 2003.

[42] G. Li, P. Niu, H. Wang, and Y. Liu, "Least square fast learning network for modeling the combustion efficiency of a 300wm coal-fired boiler," *Neural Networks*, vol. 51, pp. 57–66, 2014.

[43] M. Y. Mashor, "Hybrid training algorithm for rbf network," *International Journal of the Computer, the Internet and Management*, vol. 8, no. 2, pp. 50–66, 2000.

[44] K. J. Åström and B. Wittenmark, *Computer Controlled Systems: Theory and Design.* Prentice Hall, 1984.

[45] M. R. Stojić, *Kontinualni sistemi automatskog upravljanja.* Naučna knjiga, 1985.

[46] R. C. Dorf and R. H. Bishop, *Modern Control Systems.* Prentice Hall, 2011.

[47] W. Bolton, *Control Systems.* Elsevier, 2002.

[48] M. Popović, *Signali i sistemi.* Akademska misao, 2006.

[49] M. J. Stojčić, *Osnove automatskog upravljanja.* Mašinski fakultet Banja Luka, 2012.

[50] M. J. Stojčić, *Sinteza linearnih sistema automatskog upravljanja.* Mašinski fakultet Banja Luka, 2009.

[51] M. M. Božić and P. S. Marić, *Osnove sistema automatskog upravljanja.* Elektrotehnički fakultet Banja Luka, 2008.

[52] O. Mayr, *The origins of feedback control.* MIT Press, 1970.

[53] J. C. Maxwell, "On governors," *Proceedings of the Rozal Society of London*, no. 16, pp. 270–283, 1868.

[54] I. A. Vyshnegradskii, "On controllers of direct action," *Izv. SPB Praktiche-skogo Tekhnologicheskogo Inst.*, 1877.

[55] H. W. Bode, "Feedback — the history of an idea," *Selected Papers on Mathematical Trends in Control Theory*, pp. 106–123, 1964.

[56] H. S. Black, "Inventing the negative feedback amplifier," *IEEE Spectrum*, pp. 55–60, 1977.

[57] A. Hurwitz, "On the conditions under which an equation has only roots with negative real parts," *Mathematische Annalen*, no. 46, pp. 273–284, 1895.

[58] E. J. Routh, *Dynamics of a System of Rigid Bodies.* Macmillan, 1892.

[59] H. Nyquist, "Regeneration theory," *Bell Systems Technical Journal*, pp. 126–147, 1932.

[60] W. R. Evans, "Graphical analysis of control systems," *Transactions of the American Institute of Electrical Engineers*, vol. 67, no. 1, pp. 547–551, 1948.

[61] K. H. Ang, G. C. Y. Chong, and Y. Li, "Pid control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, 2005.

[62] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of the American Society of Mechanical Engineers*, vol. 64, pp. 759–768, 1942.

[63] H. K. Khalil, *Nonlinear Systems.* Prentice Hall, 2002.

[64] R. Marino and P. Tomei, *Nonlinear Control Design: Geometric, Adaptive and Robust.* Prentice Hall, 1995.

[65] A. M. Lyapunov, "The general problem of the stability of motion," 1892.

[66] R. E. Kalman and J. F. Bertram, "Control system analysis and design via the second method of lyapunov," *Journal of Basic Engineering*, vol. 88, pp. 371–394, 1960.

[67] J. P. LaSalle and S. Lefschetz, *Stability by Lyapunov's Second Method with Applications*. Academic Press, 1961.

[68] P. S. Parks, "Lyapunov's method in automatic control theory," *Control*, vol. 5, no. 3, pp. 102–105, 1962.

[69] K. J. Åström and B. Wittenmark, *Adaptive control*. Prentice Hall, 1994.

[70] P. Ioannou and J. Sun, *Robust Adaptive control*. Dover Publications, 2012.

[71] S. Sastry and M. Bodson, *Adaptive control: Stability, Convergence and Robustness*. Dover Publications, 2011.

[72] E. Mishkin and L. Braun, *Adaptive Control Systems*. McGraw-Hill, 1961.

[73] H. P. Whitaker, J. Yamron, and A. Kezer, *Design of Model Reference Adaptive Control for Aircraft*. MIT Instrumentation Laboratory, 1958.

[74] H. Butler, *Model reference adaptive control: from theory to practice*. Prentice Hall, 1992.

[75] K. J. Åström, "Design principles for self-tuning regulators," *Methods and Applications in Adaptive Control*, vol. 24, pp. 1–20, 1980.

[76] B. Wittenmark, *Self-tuning regulators*. Wiley Encyclopedia of Electrical and Electronics Engineering, 2001.

[77] E. F. Camacho and C. Bordons, *Model Predictive Control*. Springer, 1999.

[78] D. Bao-Cang, *Modern Predictive Control*. CRC Press, 2010.

[79] J. C. Allwright, *Advances in Model-Based Predictive Control*. Oxford University Press, 1994.

[80] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control, part I. The basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[81] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control, part II. Extensions and interpretations," *Automatica*, vol. 23, no. 2, pp. 149–160, 1987.

[82] P. Albertos and R. Ortega, "On generalized predictive control: Two alternative formulations," *Automatica*, vol. 25, pp. 753–755, 1989.

[83] T. Alvarez and C. Prada, "Handling infeasibility in predictive control," *Computers and Chemical Engineering*, vol. 21, pp. 577–582, 1997.

[84] L. P. J. Veelenturf, *Analysis and Applications of Artificial Neural Networks*. Prentice Hall, 1995.

[85] A. Konar, *Artificial Intelligence and Soft Computing Behavioral and Cognitive Modeling of the Human Brain*. CRC Press, 2000.

[86] C. Fyfe, *Artificial Neural Networks*. University of Paisley, 1996.

[87] B. Kröse and P. van der Smagt, *An Introduction to Neural Networks.* University of Amsterdam, 1996.

[88] A. P. Engelbrecht, *Computational Intelligence: An Introduction.* Wiley, 2002.

[89] N. Kazabov, *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering.* MIT Press, 1998.

[90] S. Haykin, *Neural Networks - A Comprehensive Foundation.* Prentice Hall, 1999.

[91] J. A. Freeman and D. M. Skapura, *Neural Networks Algorithms, Applications,and Programming Techniques.* Addison-Wesley, 1991.

[92] L. Jain and A. M. Finelli, *Recent Advances in Artificial Neural Networks: Design and Applications.* CRC Press, 2000.

[93] M. A. Arbib, *The Handbook of Brain Theory and Neural Networks.* MIT Press, 2003.

[94] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[95] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry.* MIT Press, 1969.

[96] T. Kohonen, *Associative Memory: A System-Theoretical Approach.* Springer, 1977.

[97] K. Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biological Cybernetics*, vol. 20, pp. 121–136, 1975.

[98] S. R. Cajal, *Histologie du système nerveux de l'homme et des vertébrés.* Maloine, 1909.

[99] G. M. Shepherd, *The Synaptic organization of the brain.* Oxford University Press, 1990.

[100] D. O. Hebb, *The Organization of Behaviour.* Wiley, 1949.

[101] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[102] F. Rosenblatt, *Principles of Neurodynamics.* Spartan Books, 1962.

[103] B. Widrow and M. E. Hoff Jr., "Adaptive switching circuits," *IRE WESCON Convention Record*, no. 4, pp. 96–104, 1960.

[104] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations*, ch. 8, pp. 318–362, MIT Press, 1986.

[105] P. J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," 1974.

[106] D. B. Parker, *Learning-logic.* MIT Press, 1985.

[107] Y. LeCun, "Une procédure d'apprentissage pour réseau à seuil asymétrique," *Proceedings of Cognitiva*, no. 85, pp. 599–604, 1985.

[108] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[109] G. Cybenko, "Approximations by superpositions of sigmoidal functions," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[110] E. J. Hartman, J. D. Keeler, and J. M. Kowalski, "Layered neural networks with gaussian hidden units as universal approximations," *Neural Computation*, vol. 2, no. 2, pp. 210–215, 1990.

[111] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.

[112] W. H. Press, B. P. Flannery, S. P. Teukolsky, and W. T. Vetterling, *Numerical recipes - the art of scientific computing*. Cambridge University Press, 1986.

[113] M. J. D. Powell, "Restart procedures for the conjugate gradient method," *Mathematical Programming*, vol. 12, pp. 241–254, 1977.

[114] R. van den Boomgaard and A. W. M. Smeulders, "Self learning image processing using a priori knowledge of spatial patterns," *Intelligent Autonomous Systems 2*, pp. 305–314, 1989.

[115] F. M. Silva and L. B. Almeida, "Speeding-up backpropagation," *Advanced Neural Computers*, pp. 151–158, 1990.

[116] M. I. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machines," *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, pp. 531–546, 1986.

[117] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179–211, 1990.

[118] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Science*, vol. 79, pp. 2554–2558, 1982.

[119] A. D. Bruce, A. Canning, B. Forrest, E. Gardner, and D. J. Wallace, "Learning and memory properties in fully connected networks," *Proceedings of the Conference on Neural Networks for Computing*, pp. 65–70, 1986.

[120] J. J. Hopfield, D. I. Feinstein, and R. G. Palmer, "'Unlearning' has a stabilizing effect in collective memories," *Nature*, vol. 304, pp. 158–159, 1983.

[121] D. E. Kohonen, T.Rumelhart and D. Zipser, "Feature discovery by competitive learning," *Cognitive Science*, vol. 9, no. 1, pp. 75–112, 1985.

[122] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.

[123] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for

a self-organizing neural pattern recognition machine," *Computer Vision, Graphics, and Image Processing*, vol. 73, pp. 54–115, 1987.

[124] R. P. Lippmann, B. Gold, and M. L. Malpass, *A comparison of Hamming and Hopfield neural nets for pattern classification.* MIT, 1987.

[125] T. Martinetz and K. Schulten, "A 'neural-gas' network learns topologies," in *Artificial Neural Networks*, pp. 397–402, Elsevier Science, 1991.

[126] B. Fritzke, "Let it grow – self-organizing feature maps with problem dependent cell structure," in *Artificial Neural Networks*, pp. 403–408, Elsevier Science, 1991.

[127] S. Grossberg, "Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors," *Biological Cybernetics*, vol. 23, no. 3, pp. 121–134, 1976.

[128] J. A. Hartigan, *Clustering Algorithms.* Wiley, 1975.

[129] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Transactions on Electronic Computers*, vol. 14, pp. 326–334, 1965.

[130] H. N. Mhaskar, "Neural networks for optimal approximation of smooth and analytic functions," *Neural Computation*, vol. 8, no. 1, pp. 164–177, 1996.

[131] P. Niyogi and F. Girosi, "On the relationship between generalization error, hypothesis complexity and sample complexity for radial basis functions," *Neural Computation*, vol. 8, no. 4, pp. 819–842, 1996.

[132] C. A. Micchelli, "Interpolation of scattered data: Distance matrices and conditionally positive definite functions," 1986.

[133] J. Hadamard, "Sur les problèmes aux dérivées partielles et leur signification physique," *Princeton University Bulletin*, pp. 49–52, 1902.

[134] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems.* Winston, 1977.

[135] A. N. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Soviet Mathematics*, vol. 4, pp. 1035–1038, 1963.

[136] H. Lebesgue, *Leçons sur l'intégration et la recherche des fonctions primitives.* Gauthier-Villars, 1904.

[137] M. Fréchet, "Sur les ensembles de fonctions et les opérations linéaires," *Comptes rendus de l'Académie des sciences*, vol. 144, pp. 1414–1416, 1907.

[138] C. N. Dorny, *A Vector Space Approach to Models and Optimization.* Wiley, 1975.

[139] F. Riesz, "Sur une espèce de géométrie analytique des systèmes de fonctions sommables," *Comptes rendus de l'Académie des sciences*, vol. 144, pp. 1409–1411, 1907.

[140] L. Debnath and P. Mikusiński, *Introduction to Hilbert spaces with applications.* Academic Press, 1990.

[141] R. Courant and D. Hilbert, *Methods of Mathematical Physics, vol. 2.* Interscience, 1962.

[142] J. Park and I. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991.

[143] J. Moody and C. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, pp. 281–294, 1989.

[144] R. O. Duda and P. E. Hart, *Pattern classification and scene analysis.* Wiley, 1973.

[145] S. Chen, "Nonlinear time series modelling and prediction using gaussian rbf networks with enhanced clustering and rls learning," *Electronics Letters*, vol. 31, no. 2, pp. 117–118, 1995.

[146] D. Wettschereck and T. G. Dietterich, "Improving the performance of radial basis function networks by learning center locations," *Advances in Neural Information Processing Systems*, vol. 4, pp. 1133–1140, 1992.

[147] T. J. Sejnowski and C. R. Rosenberg, "Parallel networks that learn to pronounce english text," *Complex Systems*, vol. 1, pp. 145–168, 1987.

[148] D. Kosic, "Fast clustered radial basis function network as an adaptive predictive controller," *Neural Networks*, vol. 63, pp. 79–86, 2015.

[149] Y.-N. Wang and X.-F. Yuan, "SVM approximate-based internal model control strategy," *Acta Automatica Sinica*, vol. 34, no. 2, pp. 172–179, 2008.

[150] M. Bozic, I. Krcmar, and J. Igic, "An adaptive neural imc design of nonlinear dynamic processes," *Infoteh-Jahorina*, vol. 15, pp. 752–757, 2016.

[151] *DTS200 Laboratory Setup Three-Tank-System.* amira GmbH, 2002.

[152] S. Iplikci, "A support vector machine based control application to the experimental three-tank system," *ISA Transactions*, vol. 49, no. 3, pp. 376–386, 2010.

[153] W.-D. Chang, "A multi-crossover genetic approach to multivariable PID controllers tuning," *Expert Systems with Applications*, vol. 33, pp. 620–626, 2007.

[154] O. K. Erol and I. Eksin, "A new optimization method: Big bang—big crunch," *Advances in Engineering Software*, vol. 37, pp. 106–111, 2006.

[155] Z. Hou and S. Jin, "Data-driven model-free adaptive control for a class of mimo nonlinear discrete-time systems," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2173–2188, 2011.

Biografija autora

Dino Kosić je rođen 22.11.1983. u Banjoj Luci, gdje je završio i osnovnu školu 1998. godine, te Gimnaziju 2002. godine. Školovanje je nastavio na Elektrotehničkom fakultetu Univerziteta u Banjoj Luci, smjer Računarska tehnika i automatika. Diplomirao je 2007. godine kao najbolji student. Iste godine je zaposlen kao saradnik na Elektrotehničkom fakultetu, prvo u naučnom zvanju asistenta, a od 2012. u zvanju višeg asistenta.

Образац -3

УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ

ИЗВЈЕШТАЈ
о оцјени урађене докторске дисертације

I ПОДАЦИ О КОМИСИЈИ

Комисију је именовао Сенат Универзитета у Бањој Луци 27.10.2016. године,
Састав Комисија за оцјену урађене докторске дисертације:

1. Др Петар Марић, редовни професор; педсједник;
   Ужа научна област Аутоматика и роботика,
   Универзитет у Бањој Луци, Електротехнички факултет
2. Др Милорад Божић, редовни професор; ментор-члан;
   Ужа научна област: Аутоматика и роботика, Вјештачка интелигенција,
   Универзитет у Бањој Луци, Електротехнички факултет
3. Др Бранко Ковачевић, редовни професор; члан;
   Ужа научна област Аутоматика,
   Универзитет у Беогарду, Електротехнички факултет
4. Др Жељко Ђуровић, редовни професор; члан;
   Ужа научна област Аутоматика,
   Универзитет у Беогарду, Електротехнички факултет
5. Др Игор Крчмар, доцент; члан;
   Ужа научна област Аутоматика и роботика,
   Универзитет у Бањој Луци, Електротехнички факултет

1) Навести датум и орган који је именовао комисију;
2) Навести састав комисије са назнаком имена и презимена сваког члана, научно-наставног звања, назива уже научне области за коју је изабран у звање и назива универзитета/факултета/института на којем је члан комисије запослен.

II ПОДАЦИ О КАНДИДАТУ

Дино (Баслија) Косић;
Рођен 22.11.1983, у Бањој Луци, Југославија;
Претходни циклус студија је завршио на Универзитету у Бањој Луци, Електротехнички факултет, студијски програм Рачунарска техника и аутоматика, и стекао звање дипломирани инжињер електротехнике – мастер рачунарства и информатике;
Академско звање мастера је стекао на Електротехничком факултету, одбраном дипломског рада 26.06.2007. године под називом Информатичка платформа за подршку берзанским трансакцијама, рачунарска техника и аутоматика;
Уписан је на докторске студије 2009. на студијском програму Информационо-комуникационе технологије.

1) Име, име једног родитеља, презиме;

2) Датум рођења, општина, држава;
3) Назив универзитета и факултета и назив студијског програма академских студија II циклуса, односно послиједипломских магистарских студија и стечено стручно/научно звање;
4) Факултет, назив магистарске тезе, научна област и датум одбране магистарског рада;
5) Научна област из које је стечено научно звање магистра наука/академско звање мастера;
6) Година уписа на докторске студије и назив студијског програма.

III УВОДНИ ДИО ОЦЈЕНЕ ДОКТОРСКЕ ДИСЕРТАЦИЈЕ

Наслов дисертације је Адаптивно-предиктивни контролер на бази неуралних мрежа
Дисертацију је прихватио Сенат Универзитета у Бањој Луци, 12.10.2015.
Садржај:

Обим: 121 страна (укупно 132 стране са свим додатним странама)
Број табела: 8
Број слика, шема, графикона: 49
Број цитиране литературе: 155

1) Наслов докторске дисертације;
2) Вријеме и орган који је прихватио тему докторске дисертације
3) Садржај докторске дисертације са страничењем;
4) Истаћи основне податке о докторској дисертацији: обим, број табела, слика, шема, графикона, број цитиране литературе и навести поглавља.

IV УВОД И ПРЕГЛЕД ЛИТЕРАТУРЕ

Адаптивно управљање је од седамдесетих година прошлог вијека непрекидно у фокусу истраживања у области аутоматског управљања. Паралелно с тим расте његов практичан значај кроз многобројне реализације, прије свега у индустрији, али и у другим областима. Несумњиво ће адаптивно управљање бити предмет интензивних истраживања и у наредном периоду, због његовог својства да подразумијева аутоматско подешавање параметара регулатора у реалном времену тако да се задржавају задате перформансе управљања када је динамичко понашање објекта управљања непознато и/или се мијења [1,2,3,4].

До сада развијене методе адаптивног управљања се по правилу сврставају у једну од сљедеће три групе: распоређивање појачања (енгл. *gain scheduling*), адаптивно управљање на основу референтног модела (енгл. *model-reference adaptive control*) [5,6,7], и самоподешавајући регулатори (енгл. *self-tuning regulators*).

Предиктивно управљање на бази модела (енгл. *Model Predictive Control* - MPC) представља фамилију управљачких метода које експлицитно користе модел за предвиђање излаза процеса у будућем времену. Различити MPC алгоритми се међусобно разликују искључиво по моделу кориштеном за представљање процеса, те функцији циља коју треба минимизовати.

Предности MPC метода су:

- довољно је ограничено познавање управљања, јер су концепти врло интуитивни, а истовремено је подешавање контролера релативно лако;
- може се користити за јако разнолике примјене, од једноставних до

комплексних система, укључујући оне са великим транспортним кашњењем, те неминимално фазне и нестабилне системе;

- лако се проширује на већи број промјенљивих;
- компензује мртво вријеме и мјерљиве поремећаје;
- резултује у контролеру са једноставним линеарним законом управљања;
- лако укључује ограничења система итд.

Најочигледнија мана је проблем рачунања, нарочито код адаптивних контролера, а посебно ако се узимају у обзир и ограничења. Иако су нумеричке перформансе данашњих контролера јако добре, то не важи обавезно за многе контролере у процесној индустрији, а поред тога, добар дио процесног времена рачунар који обавља и управљачке активности мора да посвети и осталим активностима (комуникација, приказивање и меморисање стања процеса и сл.). Осим тога, највећа мана је потреба за познавањем доброг модела процеса. Иако је процес синтезе контролера независан од модела, тачност резултујућег контролера зависи од тачности модела, тј. његовог слагања са стварним процесом [8,9,10,11,12].

Како је за неке процесе јако тешко или чак немогуће одредити физички заснован модел, како због комплексности унутрашњих процеса, тако и због недостатка знања о критичним параметрима модела, истраживања су оријентисана у смјеру тражења адекватних метода идентификације процеса, типа „црна кутија" (енгл. *black-box model*). Обећавајуће резултате, али и одређена ограничења, показују истраживања на бази неуралних мрежа.

Помоћу неуралних мрежа се могу добити адекватни модели на основу одговарајућих мјерених података о улазима и излазима процеса [13,14,15,16,17,18,19].

Мане кориштења неуралних мрежа су у неопходним алгоритмима за обучавање. Најчешће кориштене варијанте алгоритама обучавања су засноване на итеративним, градијентим методама које су рачунски и временски захтјевне, уз прилично стриктне услове за конвергенцију. Уколико се неурална мрежа користи само за моделовање у предиктивном управљању, прије укључивања у управљачку петљу, овај проблем и није тако значајан. Насупрот томе, за потребе адаптивног управљања тежиште је на адаптацији параметара контролера у реалном времену, па су наведена ограничења алгоритама обучавања веома значајна. Сама архитектура система која омогућава подешавање параметара неуралне мреже током рада је једноставна, јер неурална мрежа је инхерентно спремна за ажурирање параметара уколико се њена структура не мијења. Стога, проблем адаптације оваквог контролера се своди на брзину додатног обучавања [20,21,22,23,24,25,26,27].

Иако су развијени многи методи са бржом и сигурном конвергенцијом, као и многе варијације архитектура неуралних мрежа које се користе као адаптивно-предиктивни контролери, и даље су важна истраживања која ће умањити или елиминисати та ограничења [28,29,30,31,32,33,34,35,36].

Један од кључних истраживачких изазова у области адаптивног управљања је повећање способности адаптације. Реализација адаптивно-предиктивног управљања помоћу неуралних мрежа код којих је краће вријеме процесирања значајан је допринос у том правцу.

Истраживања такве архитектуре неуралне мреже која омогућава брзо и ефикасно обучавање, нарочито ако се може замијенити поступком који се извршава у једном кораку, тако да се елиминише потреба за итеративним поступком, доприноси већој способноси адаптације контролера. Неурална мрежа са таквим својствима би представљала допринос у фази препроцесирања, као и у фази адаптивног управљања у реалном времену. На тај начин би била унапређење како за управљање објектима непознате динамике тако и управљање објектима са промјенљивом динамиком.

Преглед литературе

[1] K. J. Åström and B. Wittenmark, *Adaptive control*, Prentice Hall, 1994.

[2] P. Ioannou and J. Sun, *Robust Adaptive control*, Dover Publications, 2012.

[3] S. Sastry and M. Bodson, *Adaptive control: Stability, Convergence and Robustness*. Dover Publications, 2011.

[4] E. Mishkin and L. Braun, *Adaptive Control Systems*. McGraw-Hill, 1961.

[5] Y.D. Landau, *Adaptive control- The Model reference approach*. New York: Marcel Dekker, 1979.

[6] E. Trulsson and L. Ljung, *Adaptive control based on explicit criterion minimization*, Automatica 21, pp. 385-399, 1985.

[7] J.J. Craig, Adaptive Control of Mechanical Manipulators, Addison Wesley, 1988.

[8] M. M. Božić, J. Igić, I. Krčmar, "Adaptivno prediktivno upravljanje nelinearnim sistemima," *Zbornik radova XLVI Konferencije za ETRAN*, svez. 1, str. 197–200, 2002.

[9] M. Hagan, H. Demuth, and O. De Jesus, "An introduction to the use of neural networks in control systems," *International Journal of Robust and Nonlinear Control*, vol. 12, no. 11, pp. 959–985, 2002.

[10] S. S. Ge, C. Yang, and T. H. Lee, "Adaptive predictive control using neural network for a class of pure-feedback systems in discrete time," *IEEE Transactions on Neural Networks*, vol. 19, no. 9, pp. 1599–1614, 2008.

[11] H. J. Sussmann, "Uniqueness of the weights for minimal feedforward nets with a given input-output map," *Neural Networks*, vol. 5, pp. 589–593, 1992.

[12] F. Albertini and E. D. Sontag, "For neural nets, function determines form," *Proceedings of 31th IEEE Conference on Decision and Control*, pp. 26–31, 1992.

[13] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, pp. 4–27, 1990.

[14] K. S. Narendra and K. Parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 252–262, 1991.

[15] N. Sadegh, "A perceptron network for functional identification and control of nonlinear systems," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 982–988, 1993.

[16] J. Leitner, A. J. Calise, and J. V. R. Prasad, "Analysis of adaptive neural networks for helicopter flight control," *Journal of Guidance Control and Dynamics*, vol. 20, no. 5, pp. 972–979, 1997.

[17] J. McFarland, A. J. Calise, and J. V. R. Prasad, "Adaptive nonlinear control of agile anti-air missiles using neural networks," *IEEE Transactions on Control Systems Technologies*, vol. 8, no. 5, pp. 749–756, 2000.

[18] A. J. Calise, N. Hovakimyan, and H. Lee, "Adaptive output feedback control of nonlinear systems using neural networks," *Automatica*, vol. 37, no. 8, pp. 1201–1211, 2001.

[19] A. Yesildirek and F. L. Lewis, "Feedback linearization using neural networks," *Automatica*, vol. 31, no. 11, pp. 1659–1664, 1995.

[20] T. Zhang, C. C. Hang, and S. S. Ge, "Robust adaptive controller for general nonlinear

systems using multilayer neural networks." preprint, 2001.

[21] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor and Francis, 1999.

[22] G. Arslan and T. Basar, "Disturbance attenuating controller design for strict-feedback systems with structurally unknown dynamics," *Automatica*, vol. 37, no. 8, pp. 1175–1188, 2001.

[23] J. Q. Gong and B. Yao, "Neural network adaptive robust control of nonlinear systems in semi-strict feedback form," *Automatica*, vol. 37, no. 8, pp. 1149– 1160, 2001.

[24] J. Wang and J. Huang, "Neural network enhanced output regulation in nonlinear systems," *Automatica*, vol. 37, no. 8, pp. 1189–1200, 2001.

[25] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 837–863, 1992.

[26] M. M. Polycarpou and P. A. Ioannou, "Identification and control using neural network models: design and stability analysis," 1991.

[27] M. M. Polycarpou and P. A. Ioannou, "Neural networks as on-line approximators of nonlinear systems," *Proceedings of 31th IEEE Conference on Decision and Control*, pp. 7–12, 1992.

[28] M. Jiang, G. Gielen, B. Zhang, and Z. Luo, "Fast learning algorithms for feedforward neural networks," *Applied Intelligence*, vol. 18, pp. 37–54, 2003.

[29] G. Li, P. Niu, H. Wang, and Y. Liu, "Least square fast learning network for modeling the combustion efficiency of a 300wm coal-fired boiler," *Neural Networks*, vol. 51, pp. 57–66, 2014.

[30] E. Castillo, B. Guijarro-Berdinas, O. Fontenla-Romero, and A. AlonsoBetanzos, "A very fast learning method for neural networks based on sensitivity analysis," *Journal of Machine Learning Research*, vol. 7, pp. 1159–1182, 2006.

[31] D. Bao-Cang, *Modern Predictive Control*. CRC Press, 2010.

[32] Y.-N. Wang and X.-F. Yuan, "SVM approximate-based internal model control strategy," *Acta Automatica Sinica*, vol. 34, no. 2, pp. 172–179, 2008. P. Ioannou and J. Sun, *Robust Adaptive control*. Dover Publications, 2012.

[33] S. Sastry and M. Bodson, *Adaptive control: Stability, Convergence and Robustness*. Dover Publications, 2011.

[34] Z. Hou and S. Jin, "Data-driven model-free adaptive control for a class of mimo nonlinear discrete-time systems," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2173–2188, 2011.

[35] D. Kosic, "Fast clustered radial basis function network as an adaptive predictive controller," *Neural Networks*, vol. 63, pp. 79–86, 2015.

[36] M. Bozic, I. Krcmar, and J. Igic, "An adaptive neural IMC design of nonlinear dynamic processes," *Infoteh-Jahorina*, vol. 15, pp. 752–757, 2016.

1) Укратко истаћи разлог због којих су истраживања предузета и представити проблем, предмет, циљеве и хипотезе;

2) На основу прегледа литературе сажето приказати резултате претходних истраживања у вези проблема који је истраживан (водити рачуна да обухвата најновија и најзначајнија сазнања из те области код нас и у свијету);

3) Навести допринос тезе у рјешавању изучаваног предмета истраживања;
4) Навести очекиване научне и прагматичне доприносе дисертације.

V МАТЕРИЈАЛ И МЕТОД РАДА

У току реализације докторске дисертације, између осталог, коршћене су сљедеће основне научне методе истраживања:

- Прикупљање, анализа и систематизација доступне литературе;
- Пројектовање и анализа алгоритама управљања;
- Математичко извођење доказа радне хипотезе;
- Статистичка анализа експерименталних резултата.

Резултати истраживања и побољшања постигнута предложеним алгоритмима су потврђени путем симулација рада адаптивног система аутоматкосг управљања, за шта је коришћен програмски пакет MATLAB.
Надаље, побољшања су потврђена резултатима експеримента проведеног на реалном процесу. Предложени алгоритам је имплементриран на контролеру за адаптивно управљање системом три резервоара. За верификацију побољшања, на истом контролеру су такође имплементирани ПИД закон управљања и управљање на основу подацима вођеног адаптивног управљања без познавања модела. Дати су упоредни резултати за сва три реализована контролера, који потврђују предности предложеног адаптивно-предиктивног контролера на бази неуралних мрежа. Извршени су тестови на стандардним скуповима података кориштеним у научним радовима објављеним у међународним научним часописима и дато је поређење са стандардно кориштеним варијантама вјештачких неуралних мрежа, односно машина за учење: потпуно повезана трослојна вјештачка неурална мрежа без повратних веза обучена алгоритмом пропагације грешке уназад, екстремна машина за учење (енг. *Extreme Learning Machine*), те машина са потпорним векторима (енг. *Support Vector Machine*).

1) Објаснити материјал који је обрађиван, критеријуме који су узети у обзир за избор материјала;
2) Дати кратак увид у примијењени метод истраживања при чему је важно оцијенити сљедеће:
   1. Да ли су примијењене методе истраживања адекватне, довољно тачне и савремене, имајући у виду достигнућа на том пољу у свјетским нивоима;
   2. Да ли је дошло до промјене у односу на план истраживања који је дат приликом пријаве докторске тезе, ако јесте зашто;
   3. Да ли испитивани параметри дају довољно елемената или је требало испитивати још неке, за поуздано истраживање;
   4. Да ли је статистичка обрада података адекватна.

VI РЕЗУЛТАТИ И НАУЧНИ ДОПРИНОС ИСТРАЖИВАЊА

На основу текста дисертације Комисија је мишљења да су постигнути сљедећи резултати и научни допринос:

Брза груписана вјештачка неурална мрежа са Гаусовим активационим функцијама. Представљен је потпуно нова архитектура вјештаких неуралних мрежа са Гаусовим активационим функцијама (енгл. *Fast Clustered Radial Basis Function Network* – FCRBFN). Ради се о трослојној архитектури без повратних веза, у којој нису сви неурони улазног слоја повезани са неуронима скривеног слоја, већ је један неурон улазног слоја повезан само са групом неурона у скривеном слоју, а са којима нису повезани други неурони улазног слоја, док су сви неурони скривеног слоја повезани са свим неуронима излазног слоја. Група неурона скривеног слоја која је повезана само са једним од неурона улазног слоја се назива кластер неурона.

Параметри Гаусових активационих функција (центри и ширине) се могу бирати независно за сваки од кластера, а у складу са опсегом вриједности улаза неурона улазног слоја са којим је одређени кластер повезан. Тежине веза између улазног и скривеног слоја су све једнаке 1, тако да се обучавање овакве мреже своди на одређивање тежина веза између скривеног и излазног слоја, што је неитеративна метода и завршава се у једном кораку. Показано је да се то обучавање обавља рјешавањем једног система линеарних једначина за сваки од излаза мреже. Дат је потпун опис процедуре обучавања овакве мреже, као и брзе процедуре додавања новог знања помоћу методе најмањих квадрата. Дата је и препорука избора параметара мреже. За верификацију брзине обучавања, као и способности регресије и генерализације, извршени су тестови на стандардним скуповима података коришћеним у научним радовима објављеним у међународним научним часописима и дато је поређење са стандардно коришћеним варијантама вјештачких неуралних мрежа, односно машина за учење: потпуно повезана трослојна вјештачка неурална мрежа без повратних веза обучена алгоритмом пропагације грешке уназад, екстремна машина за учење, те машина са потпорним векторима. Резултати извршених тестирања показују да представљена мрежа даје најбоље резултате у времену и стабилности извршавања, те је према перформансама увијек прва или друга по питању прецизности било обучавања, било тестирања. Такође је показано да је избор параметара мреже интуитиван и да је мрежа робусна на тај избор.

Стратегије предиктивног и адаптивно-предиктивног управљања на бази FCRBFN. Описане су двије стратегије управљања на бази FCRBFN: Предиктивно управљање на бази модела са FCRBFN и адаптивно-предиктивно управљање на бази FCRBFN. У првој стратегији FCRBFN је коришћена као предиктор, односно за моделовање објекта управљања, а у стандардној управљачкој структури за предиктивно управљање на бази модела. Извршени експерименти са нелинеарним моделом (који се често користи у литератури) уз присуство константог поремећаја на излазу система показују да предложена структура успјешно рјешава задатак управљања. У другој стратегији предложена је структура управљачког система гдје се као адаптивно-предиктивни контролер користи FCRBFN. У таквој поставци, мрежа не моделује објекат управљања чиме би служила за предикцију излаза система, већ садржи знање о томе какав управљачки сигнал резултује у каквом излазу система, ако је жељени излаз дат. Ово значи да се предиктивне особине таквог контролера не користе индиректно за одређивање закона управљања, већ директно кроз предикцију потребног закона управљања. За доказивање могућности овакког контролера, исти је примијењен на лабораторијски систем са три резервоара, присутан у многим лабораторијама свијета, укључујући и Лабораторију за Аутоматику Електротехничког факултета Универзитета у Бањој Луци, а често се користи и у научним радовима објављеним у међународним научним часописима за тестирање различитих метода управљања. Изведена су два експеримента са истом поставком и истим захтјевима, први без, други са шумом мјерења. Исти експерименти су поновљени коришћењем још двије врсте контролера: најчешће коришћен индустријски контролер - ПИД контролер, те PFDL-MFAC (енгл. *Partial Form Dynamic Linearization based Model Free Adaptive Control*) контролер. Резултати првог експеримента показују да предложени контролер даје боље резултате по питању праћења референтног сигнала, а нарочито по питању оптерећења актуатора. Резултати другог теста показују још боље резултате предложеног контролера у односу на друга два.

1) Укратко навести резултате до којих је кандидат дошао;
2) Оцијенити да ли су добијени резултати јасно приказани, правилно, логично и јасно тумачени, упоређујући са резултатима других аутора и да ли је кандидат при томе испољавао довољно критичности;
3) Посебно је важно истаћи до којих нових сазнања се дошло у истраживању, који је њихов теоријски и практични допринос, као и који нови истраживачки задаци се на основу њих могу утврдити или назирати.

VII ЗАКЉУЧАК И ПРИЈЕДЛОГ

Докторска дисертација Дине Косића под називом „Адаптивно-предиктивни контролер на бази неуралних мрежа" садржи веома значајне резултате и научне доприносе у области адаптивног управљања. У дисертацији је дат систематичан и критички преглед стања у области. Главни научни доприноси се односе на оригинално рјешење за брзо груписане вјештачке неуралне мреже са Гаусовим активационим функцијама и на предложене стратегије предиктивног и адаптивно-предиктивног управљања на бази FCRBFN. Предложена побољшања су потврђена симулационим примјерима и експериментално у којим су извршена поређења са индустријским ПИД контролером и PFDL-MFAC контролером. Резултати поређења су презентовани одговарајућим табелама и дијаграмима.
Додатна потврда успјешно проведених истраживањав је и један публикован рад у научном часопису са SCI листе са фактором утицаја 2.076.
У закључку овог Извештаја, а имајући у виду све напријед наведено, Комисија има част да Наставно-научном вијећу Електротехничког факултета Универзитета у Бањој Луци предложи **да прихвати докторску дисертацију "Адаптивно-предиктивни контролер на бази неуралних мрежа" и да кандидату Дини Косићу одобри усмену јавну одбрану.**

1) Навести најзначајније чињенице што тези даје научну вриједност, ако исте постоје дати позитивну вриједност самој тези;
2) На основу укупне оцјене дисертације комисија предлаже:
- да се докторска дисертација прихвати, а кандидату одобри одбрана,
- да се докторска дисертација враћа кандидату на дораду (да се допуни или измијени) или
- да се докторска дисертација одбија.

Датум: 30.12.2016.

ПОТПИС ЧЛАНОВА КОМИСИЈЕ

1. Проф. др Петар Марић, предсједник;
2. Проф. др Милорад Божић, ментор-члан;
3. Проф. др Бранко Ковачевић, члан
4. Проф. др Жељко Ђуровић, члан
5. Доц. др Игор Крчмар, члан

ИЗДВОЈЕНО МИШЉЕЊЕ: Члан комисије који не жели да потпише извјештај јер се не слаже са мишљењем већине чланова комисије, дужан је да унесе у извјештај образложење, односно разлог због којих не жели да потпише извјештај.

Прилог 2.

УНИВЕРЗИТЕТУ У БАЊОЈ ЛУЦИ
ПОДАЦИ О АУТОРУ ОДБРАЊЕНЕ ДОКТОРСКЕ ДИСЕРТАЦИЈЕ

Име и презиме аутора дисертације
Дино Косић
Датум, мјесто и држава рођења аутора
22.11.1983, Бања Лука, Босна и Херцеговина
Назив завршеног факултета/Академије аутора и година дипломирања
Електротехнички факултет у Бањој Луци, 2007.
Датум одбране мастер / магистарског рада аутора
Наслов мастер / магистарског рада аутора
Академска титула коју је аутор стекао одбраном мастер/магистарског рада
Академска титула коју је аутор стекао одбраном докторске дисертације
Доктор техничких наука у области електротехнике – 480 ECTS
Назив факултета/Академије на коме је докторска дисертација одбрањена
Електротехнички факултет у Бањој Луци
Назив докторске дисертације и датум одбране
„Адаптивно-предиктивни контролер на бази неуралних мрежа“, 31.03.2017. године
Научна област дисертације према CERIF шифрарнику
Т 125
Имена ментора и чланова комисије за одбрану докторске дисертације

1. проф. др Петар Марић, предсједник
2. проф. др Милорад Божић, предсједник,
3. проф. др Бранко Ковачевић, члан,
4. проф. др Жељко Ђуровић, члан,
5. доц. др Игор Крчмар, члан.

У Бањој Луци, дана 26.04.2017. године

Декан

Проф. др Бранко Блануша

Прилог 3.

Изјава 1

ИЗЈАВА О АУТОРСТВУ

Изјављујем
да је докторска дисертација

Наслов рада АДАПТИВНО-ПРЕДИКТИВНИ КОНТРОЛЕР НА БАЗИ НЕУРАЛНИХ МРЕЖА

Наслов рада на енглеском језику NEURAL NETWORK BASED ADAPTIVE-PREDICTIVE CONTROLLER

- результат сопственог истраживачког рада,
- да докторска дисертација, у цјелини или у дијеловима, није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

У Бањој Луци 31.03.2017.

Потпис докторанта

Дино Косић

Изјава 2

**Изјава којом се овлашћује Универзитет у Бањој Луци
да докторску дисертацију учини јавно доступном**

Овлашћујем Универзитет у Бањој Луци да моју докторску дисертацију под насловом

АДАПТИВНО-ПРЕДИКТИВНИ КОНТРОЛЕР НА БАЗИ НЕУРАЛНИХ МРЕЖА

која је моје ауторско дјело, учини јавно доступном.

Докторску дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у дигитални репозиторијум Универзитета у Бањој Луци могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (*Creative Commons*) за коју сам се одлучио/ла.

1. Ауторство
2. Ауторство – некомерцијално
3. Ауторство – некомерцијално – без прераде
4. Ауторство – некомерцијално – дијелити под истим условима
5. Ауторство – без прераде
6. Ауторство – дијелити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

У Бањој Луци 31.03.2017.

Потпис докторанта

Дино Лозић

ТИПОВИ ЛИЦЕНЦНИ КРЕАТИВНЕ ЗАЈЕДНИЦЕ

Ауторство (CC BY)

Дозвољавате умножавање, дистрибуцију и јавно саопштавање дјела, и прераде, ако се наведе име аутора, на начин одређен од аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

Ауторство - некомерцијално (CC BY-NC)

Дозвољавате умножавање, дистрибуцију и јавно саопштавање дјела, и прераде, ако се наведе име аутора, на начин одређен од аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дјела.

Ауторство - некомерцијално - без прерада (CC BY-NC-ND)

Дозвољавате умножавање, дистрибуцију и јавно саопштавање дјела, без промјена, преобликовања или употребе дјела у свом дијелу, ако се наведе име аутора, на начин одређен од аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дјела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дјела.

Ауторство - некомерцијално - дијелити под истим условима (CC BY-NC-SA)

Дозвољавате умножавање, дистрибуцију и јавно саопштавање дијела, и прераде, ако се наведе име аутора, на начин одређен од аутора или даваоца лиценце, и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дјела и прерада.

Ауторство - без прерада (CC BY-ND)

Дозвољавате умножавање, дистрибуцију и јавно саопштавање дјела, без промјена, преобликовања или употребе дјела у свом дјелу, ако се наведе име аутора, на начин одређен од аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дјела.

Ауторство - дијелити под истим условима (CC BY-SA)

Дозвољавате умножавање, дистрибуцију и јавно саопштавање дјела, и прераде, ако се наведе име аутора, на начин одређен од аутора или даваоца лиценце, и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дјела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.

Напомена: Овај текст није саставни дио изјаве аутора.

Више информација на линку: *http://creativecommons.org.rs/*

Изјава 3

Изјава о идентичности штампане и електронске верзије докторске дисертације

Име и презиме аутора ДИНО КОСИЋ

Наслов рада АДАПТИВНО-ПРЕДИКТИВНИ КОНТРОЛЕР НА БАЗИ НЕУРАЛНИХ МРЕЖА

Ментор ПРОФ. ДР МИЛОРАД БОЖИЋ

Изјављујем да је штампана верзија моје докторске дисертације идентична електронској верзији коју сам предао/ла за дигитални репозиторијум Универзитета у Бањој Луци.

У Бањој Луци 31.03.2017.

Потпис докторанта

Дино Косић