Modeliranje dinamičnih sistemov z umetnimi nevronskimi mrežami in sorodnimi metodami

Juš Kocijan

Založba Univerze v Novi Gorici

2015

Modeliranje dinamičnih sistemov z umetnimi nevronskimi mrežami in sorodnimi metodami
Juš Kocijan

2. izdaja

Recenzenta: prof. dr. Aleš Belič in prof. dr. Igor Škrjanc

Jezikovne korekture: prof. dr. Gregor Kocijan

Založnik: Založba Univerze v Novi Gorici, Vipavska 13, Rožna dolina, Nova Gorica

Leto izida: 2015

ISBN 978-961-6311-92-2 (pdf)

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

681.5.033:004.032.26(075.8)(0.034.2)

KOCIJAN, Juš
Modeliranje dinamičnih sistemov z umetnimi nevronskimi mrežami in sorodnimi metodami [Elektronski vir] / Juš Kocijan. - 2. izd. - El. knjiga. - V Novi Gorici : Založba Univerze, 2015

ISBN 978-961-6311-92-2 (pdf)

281954560

To delo je objavljeno pod licenco Creative Commons. Avtor/ji besedila dovoli/jo reproduciranje, distribuiranje, dajanje v najem in priobčitev javnosti samega izvirnega avtorskega dela pod pogoji: priznanje avtorstva in nekomercialno.
Celotno besedilo licence je na voljo na spletu, na naslovu http://creativecommons.si/licence

Copyright © 2015 Založba Univerze v Novi Gorici. Vse pravice pridržane. Razmnoževanje in fotokopiranje dela v celoti ali po delih brez predhodnega dovoljenja Založbe Univerze v Novi Gorici je prepovedano.

Predgovor

Knjigo Modeliranje dinamičnih sistemov z umetnimi nevronskimi mrežami in sorodnimi metodami je mogoče uporabljati kot učbenik za področje, ki ga obravnava, ali kot uvodno strokovno literaturo, ki je lahko odskočna deska za strokovno bolj poglobljeno literaturo s tega področja.

Namenjena je študentom, predvsem podiplomske ravni, ki imajo zadosti predznanja o dinamičnih sistemih, in strokovnjakom z ustreznim predznanjem za samoizobraževanje in seznanjanje z opisanimi koncepti in pogledi na njih.

Knjiga ni pripravljena kot podroben teoretično zasnovan opis področja, marveč je pregled področja identifikacije dinamičnih sistemov z nevronskimi mrežami in sorodnimi metodami z vidika teorije dinamičnih sistemov in predvsem njene uporabe. Delo naj bi bralca seznanilo s stališčem o tej problematiki, ki jo različne stroke, čeprav sorodne, obravnavajo zelo različno.

Gradiva za knjigo sem zbiral kot podlago za predavanja pri predmetu Izbrana poglavja iz teorije avtomatskega vodenja na podiplomskem študiju Fakultete za elektrotehniko Univerze v Ljubljani. Za njeno končno obliko se zahvaljujem Miru Štrublju, ki je izrisal veliko slik, dr. Gregorju Gregorčiču, ki je dovolil uporabo nekaterih slik iz njegove doktorske disertacije, in vsem drugim, ki so neposredno ali posredno vplivali na nastajanje tega dela.

Ljubljana, v jeseni 2007 *Juš Kocijan*

Kazalo

1 Uvod v umetne nevronske mreže **1**

1.1 Kratek časovni pregled razvoja področja . . . 1

1.2 O umetnih nevronskih mrežah . . . 2

1.3 Večnivojski perceptron . . . 5

1.4 Mreža radialnih baznih funkcij . . . 7

1.5 Nevronska mreža za opis nelinearnega dinamičnega sistema . . . 9

2 Identifikacija linearnih dinamičnih sistemov **13**

2.1 Povzetek o identifikaciji linearnih dinamičnih sistemov . . . 13

2.2 Primer identifikacije mehatronskega sistema . . . 16

3 Identifikacija nelinearnih dinamičnih sistemov **27**

3.1 Splošno o identifikaciji nelinearnih sistemov . . . 27

3.2 Primer identifikacije pH procesa . . . 35

4 Vodenje procesov z umetnimi nevronskimi mrežami **41**

4.1 Principi uporabe za vodenje . . . 41

4.2 Prediktivno vodenje . . . 44

5 Mreže lokalnih modelov in zliti večmodelni sistemi **51**

5.1 Hitrostna linearizacija . . . 54

5.2 Zlivani večmodelni sistemi . . . 58

6 Načrtovanje vodenja z razvrščanjem ojačenj **63**

6.1 Načrtovanje z uporabo hitrostne linearizacije . . . 63

6.2 Primer načrtovanja vodenja: enosegmentni robotski manipulator . . . 66

6.3 Primer načrtovanja vodenja enote za pripravo plina . . . 67

7 Identifikacija nelinearnih sistemov z Gaussovimi procesi **73**

7.1 Gaussovi procesi 73

7.2 Identifikacija dinamičnih sistemov z GP 77

7.3 Primer identifikacije pH procesa 81

7.4 Načrtovanje vodenja 83

1

Uvod v umetne nevronske mreže

Umetne nevronske mreže (angl. artificial neural networks)[1] so pojem in način reševanja raznih problemov, ki se pojavljajo že več kot pol stoletja. V knjigi jih bomo imenovali kar nevronske mreže. Njihova uporaba je zelo raznovrstna tako po strokovnih področjih (tehnika, računalništvo, naravoslovje, družboslovje itd.) in po problemskih področjih (regresija, klasifikacija, rojenje, logično povezovanje itd.).

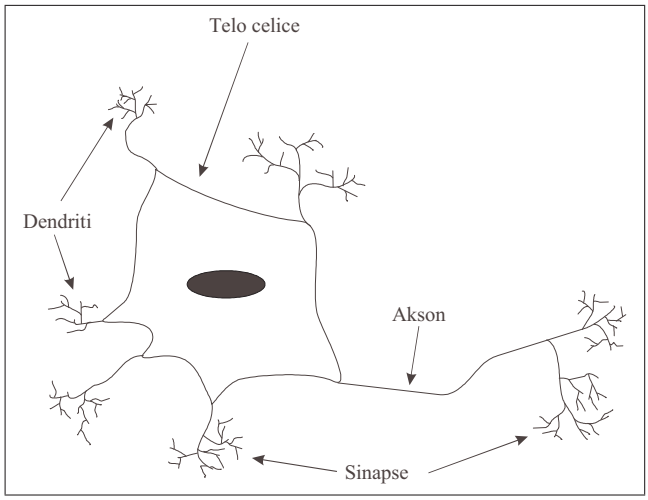
Nevronske mreže se pogosto uporabljajo za eksperimentalno modeliranje dinamičnih sistemov, predvsem kompleksnih in nelinearnih sistemov, in posledično tudi pri načrtovanju avtomatskega vodenja. V knjigi bomo osvetlili predvsem problematiko s tega področja.

Pri opisu nevronskih mrež se je potrebno zavedati, da gre za zgoščen pregled, ki ni mišljen, da bi bil popoln, marveč je njegov namen dati vpogled v področje najpogosteje uporabljenih nevronskih mrež za eksperimentalno modeliranje nelinearnih dinamičnih sistemov in nekaterih posebnosti, ki pri tem nastopajo.

Nevronske mreže se razvrščajo kot ena, če ne kar najbolj znana, izmed metod računske inteligence. Razvrstitev in povezovanje z metodami tako imenovane umetne inteligence izvirata iz začetkov nevronskih mrež. Umetne nevronske mreže izhajajo iz ideje o ponazoritvi biološkega nevrona, ki ga prikazuje slika 1.1, z umetnim in na ta način narediti osnovni element umetnih možganov.

Ta miselni vzorec je bil pozneje presežen z dognanjem, da nevronske mreže niso nič drugega kot neke osnovne matematične funkcije, s katerimi ponazorimo izbrano nelinearno funkcijo.

V nadaljevanju poglavja sledi kratek časovni pregled razvoja nevronskih mrež. Temu bo sledila delitev nevronskih mrež po raznih kriterijih. Opisali bomo dve vrsti najpogosteje uporabljenih nevronskih mrež, in sicer večnivojski perceptron in mreže radialnih baznih funkcij. Nato bomo opisali še idejo, kako uporabimo nevronske mreže za modeliranje dinamičnih sistemov.



Slika 1.1: Biološki vzor umetnega nevrona

1.1 Kratek časovni pregled razvoja področja

V tem podpoglavju bomo povzeli glavne mejnike v časovnem razvoju področja. Namen je prikazati razvoj miselnosti, ki se je z dognanji spreminjala. Pregled smo naredili na podlagi [13] s sistemskega vidika. Gre za celosten pogled na mrežo in njene lastnosti z vidika vhodnih in izhodnih podatkov.

1943 Tega leta sta McCulloch in Pitts [16] predstavila modele bioloških nevronov, ki so bili temeljni elementi vezij za reševanje računskih nalog. Iz tistega časa izvira mišljenje, da gre za element, ki bo nadomestil funkcije biološkega nevrona.

1949 Hebb, ki je bil po stroki psiholog, je izdal knjigo [8], v kateri je opisal učenje nevronov kot nastavljanje uteži na povezavah med enotami, to je nevroni. Različna in veliko pozneje razvita pravila določanja uteži pri nevronskih mrežah, ki jih lahko predstavimo kot enote z medsebojnimi povezavami, so v bistvu izpeljanke iz te metode.

1959 Rosenblatt [25] je opisal enonivojsko nevronsko mrežo, sestavljeno iz elementov, ki jim je dal ime »per-

ceptron« (dojemalnik). Nevronske mreže iz teh elementov, čeprav drugačne, so še danes med najbolj običajno uporabljenimi nevronskimi mrežami. Temeljni elementi nevronske mreže so preklopne funkcije, mreža pa ima, kot smo že omenili, samo en nivo, kar pomeni, da gre vhodni signal iz posameznega vhoda do posameznega izhoda samo skozi en nevron. Ta enostavna struktura precej omejuje zmožnosti nevronske mreže.

1960 Widrow in Hoff [29] sta predstavila nevronsko mrežo iz linearnih elementov, imenovanih Adaline (angl. ADAptive LINear Element). To mrežo lahko električno predstavimo kot vhode, utežene z upori in povezane na seštevalnik. V bistvu je nevron linearna funkcija in mreža Adaline aproksimira z uteženo vsoto linearnih funkcij. Avtorja sta tudi izpeljala prvo analitično metodo nastavljanja uteži, ki sta jo imenovala pravilo delta. V bistvu gre za metodo najmanjših kvadratov.

1963 Widrow in Smith [30] sta uporabila nevronsko mrežo Adaline za stabilizacijo invertiranega nihala.

1969 Minsky in Papert sta v knjigi Perceptrons [17] pokazala na omejitve (enonivojskih) nevronskih mrež, ki uporabljajo perceptron. Na primer, z njimi ni mogoče razvrščati elementov, ki niso linearno ločljivi. To pomeni, da z enonivojsko mrežo ne moremo izvesti logične funkcije »ekskluzivni ali« (XOR). Avtorja sta v svoji knijigi pokazala tudi, da lahko omejitve enonivojskih mrež presežemo z dvonivojsko mrežo, toda nista pokazala, kako lahko nastavimo uteži večnivojske mreže. Objava te knjige je pomenila konec obdobja enonivojskih nevronskih mrež. Izsledki, objavljeni v knjigi [17], so povzročili, da je do približno leta 1982 nastopilo zatišje v raziskavah na področju nevronskih mrež.

1986 Rumelhart in sodelavci [26] so objavili postopek za nastavljanje - učenje uteži za večnivojsko mrežo. Ta način nastavljanja so poimenovali vzvratno učenje (angl. backpropagation), ker so jo interpretirali kot razširjanje napake iz izhodov nevronske mreže proti vhodom. S tem delom so omogočili, da so z večnivojskim perceptronom lahko razvrščali tudi funkcije, ki niso linearno ločljive oziroma lahko z njimi izvedemo poljubno nelinearno preslikavo. S tem dognanjem so povzročili razcvet raziskav in uporabe umetnih nevronskih mrež.

pozna 80. leta Umetne nevronske mreže so začeli uporabljati za avtomatsko vodenje, npr. [24].

1990 Narendra in Parthasarathy [18] sta uporabljala nevronske mreže za identifikacijo in za vodenje dinamičnih sistemov.

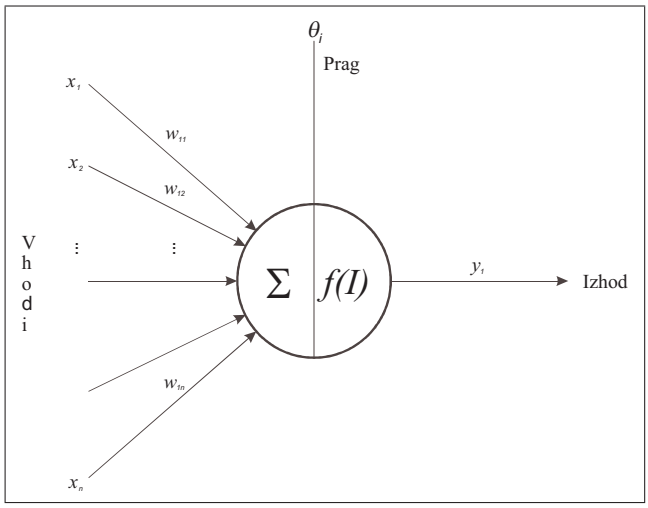
1995 Sjöberg in sodelavci so v članku [27] pokazali, da lahko na nevronske mreže s sistemskega vidika gledamo kot na identifikacijo z nelinearno regresijo, s tem so nevronskim mrežam odvzeli »bliščc« metode umetne inteligence. Te in podobne metode so strokovnjaki preimenovali v metode računske inteligence, da ne bi prišlo do nepotrebnih nesporazumov.

1.2 O umetnih nevronskih mrežah

Temeljni gradnik nevronske mreže je nevron. Opišemo ga s funkcijo

$$y_i = f(\sum_j w_{ij} x_j + \theta_i), \tag{1.1}$$

kjer so x_j vhodi v nevron, ki so uteženi z vrednostmi w_{ij}, in skupaj s predvrednostmi (angl. bias) θ_i preko aktivacijske (tudi pragovne) funkcije $f(\cdot)$ določajo izhode nevrona y_i. Umetni nevron grafično prikazuje slika 1.2



Slika 1.2: Prikaz temeljnega elementa nevronske mreže - nevrona

To formalno obliko nevrona sta uveljavila že McCulloch in Pitts [16] in oblika se je v svoji osnovi ohranila do danes. Aktivacijska funkcija ima lahko različne oblike. Prav tako so tudi nevroni lahko povezani med seboj na različne načine. Nevronske mreže so se razvijale v veliko smeri in delimo jih na različne načine. Najpogostejši kriteriji delitev so: način določanja parametrov, topologija ali namen.

Način določanja parametrov

S parametri nevronske mreže so največkrat mišljene uteži in predvrednosti, ki jih moramo določiti z nekim optimizacijskim postopkom. Načini določanja se imenujejo učenje. Ločimo tri načine učenja: nadzorovano (angl. supervised), nenadzorovano (angl. self-organising map) in ocenjevano ali stopenjsko (angl. reinforcement learning).

Nadzorovano učenje pomeni, da imamo množico vhodnih podatkov in ciljno množico izhodnih podatkov ter, glede

na izbrano kriterijsko funkcijo, učimo nevronsko mrežo preslikavo med danimi vhodnimi in izhodnimi podatki. Prvi postopki so bili namenjeni enostopenjskim nevronskim mrežam. To sta: pravilo učenja perceptrona (Hebbovo pravilo)[8]:

$$\Delta w_{ij} = \gamma y_i x_j \ ,$$

kjer je Δw_{ij} sprememba uteži in γ konstanta, ki določa hitrost učenja, in
delta pravilo (Widrow-Hoffovo pravilo)[29]:

$$\Delta w_{ij} = \gamma x (d_i - y_i) x_j \ ,$$

kjer je $(d_i - y_i)$ razlika med želeno in dejansko vrednostjo izhoda. Za to pravilo se izkaže, da je v bistvu optimizacija po metodi najmanjših kvadratov.

Najstarejši postopek učenja za večnivojske mreže je metoda povratnega učenja (angl. backpropagation). To je v bistvu delta pravilo, posplošeno za nelinearne probleme. Podrobnejši pogled na metodo pokaže, da je to gradientna optimizacijska metoda prvega reda. Metoda povratnega učenja se uporablja za izračunavanje gradienta kriterijske funkcije. Uporabna je za vse vrste nelinearnih sistemov, vendar se glede na sistem razlikuje v računski obremenitvi. Da bi izboljšali njene pomanjkljivosti, ki sta predvsem počasnost in možnost, da optimizacija obstane v lokalnem minimumu, so razvili nekatere izboljšane različice, kot sta [13]:

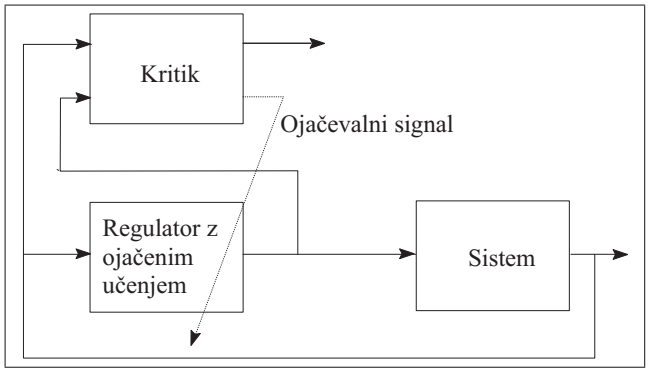
- učenje z momentom in
- učenje s prilagodljivo hitrostjo.

Gradientne metode prvega reda so znane kot neučinkovite in zato se danes za reševanje nelinearnih optimizacijskih problemov uporabljajo v glavnem Newtonove optimizacijske metode, ki spadajo med gradientne metode drugega reda. Med najpogosteje uporabljanimi sta:

- Gauss-Newtonova modifikacija in
- Levenberg-Marquardtova modifikacija.

Samoorganizirane nevronske mreže se učijo nenadzorovano in v nasprotju z nadzorovanim učenjem imajo samo-organizirane mreže na voljo samo vhodne podatke. Uporabljajo se v glavnem pri razpoznavanju vzorcev za rojenje, kvantizacijo vektorjev, redukcijo velikega števila podatkov, izločanje značilnosti. Tipičen primer je Kohonenova mreža, imenovana tudi samoorganizirana preslikava (angl. Self-Organising Map - SOM)[7].

Ocenjevano ali tudi stopenjsko učenje je v bistvu aproksimacija dinamičnega programiranja za učenje regulatorja. To pomeni, da ima zelo specifično uporabo. Učenje je sestavljeno iz dveh stopenj, in sicer ocenjevanja in popravljanja parametrov. Shematsko je prikazano na sliki 1.3.



Slika 1.3: Bločna shema stopenjskega ali ocenjevanega učenja

S tem ko se razvijajo nove optimizacijske metode, se uporabljajo tudi za učenje nevronskih mrež. V zadnjem času so popularni predvsem stohastični optimizacijski postopki, kot so iskanje parametrov in strukture z evolucijskimi načini (genetski algoritmi in genetsko programiranje).

Topologija nevronskih mrež

Po topologiji delimo nevronske mreže na tiste s popolno, lokalno in nivojsko povezanostjo. Nevronska mreža s popolno povezanostjo je na primer Hopfieldova nevronska mreža.

Hopfieldova nevronska mreža ima za aktivacijsko funkcijo preklopno funkcijo ali linearno funkcijo z nasičenjem. Njena značilnost je povratna zanka iz izhodov na vhode. Uporabljamo jo kot asociativni pomnilnik in je zanimiva predvsem s teoretičnega vidika, manj je uporabna v praksi. Izvedenka so Boltzmanovi stroji.

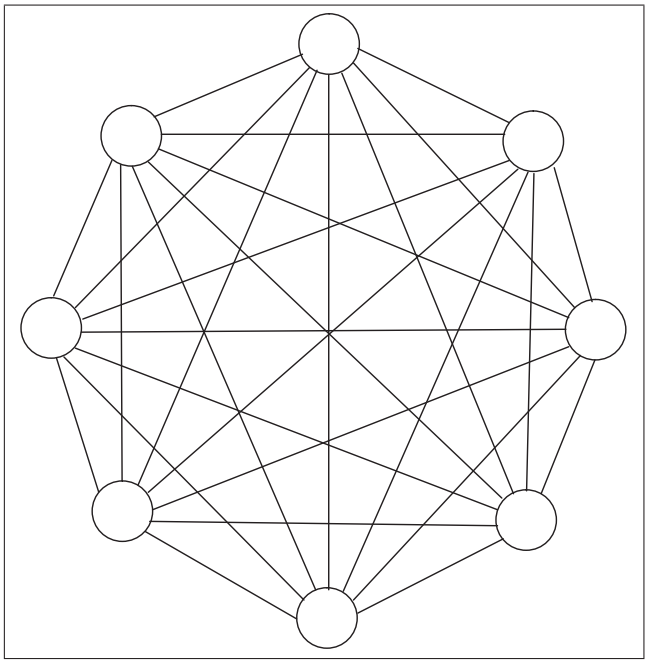
Boltzmanovi stroji so Hopfielova mreža s skritimi nivoji. Njihove parametre, to je uteži, določamo s stohastičnim pravilom spreminjanja uteži, ki ga imenujemo »simulirano ohlajanje« (angl. simulated annealing).

Primer nevronske mreže z lokalno povezanostjo je Kohonenova nevronska mreža.

Kohonenova mreža je primer mreže, ki jo uporabljamo predvsem za rojenje (angl. clustering); njena glavna značilnost je, da ohranja topološko povezavo (preslikavo) med podatki.

Primer nevronske mreže z nivojsko povezanostjo je večnivojski perceptron, o katerem bomo podrobneje spregovorili pozneje.

Druga možna delitev je na vnaprejšnje (angl. feedforward) in povratne (angl. recurrent) nevronske mreže (slika 1.4). Ta delitev je narejena glede na pretok signalov po mreži. Med vnaprejšnje bi spadale predvsem tiste z nivojsko povezanostjo, med povratne pa one s popolno in



Slika 1.4: Primer topologije povratne mreže

lokalno povezanostjo. Povratne mreže se od vnaprejšnjih, pri katerih signali tečejo izključno od vhoda proti izhodu, razlikujejo po tem, da vsebuje mreža tudi povratne zanke iz izhodov. V glavnem gre za samoorganizirajoče mreže.

Namen

Nevronske mreže po namenu lahko razdelimo bolj na grobo ali bolj na fino. Najbolj na grobo bi lahko razdelili nevronske mreže na tiste, ki se uporabljajo za klasifikacijo, in na tiste, ki se uporabljajo za regresijo.

Klasifikacija ali razporejanje je postopek, v katerem moramo podatke razporediti v končno število množic. To pomeni, da delamo preslikavo v kvantizirani izhodni prostor. Regresija je postopek, v katerem lahko izhodne vrednosti zavzamejo kakršno koli vrednost. Pri regresiji delamo vhodno-izhodno preslikavo v zvezni izhodni prostor.

V tej knjigi je poudarek na uporabi nevronskih mrež in podobnih pristopov za modeliranje dinamičnih sistemov in uporabo teh modelov za načrtovanje avtomatskega vodenja. Zato se ne bomo ukvarjali s klasifikacijo in njim namenjenimi nevronskimi mrežami, marveč se bomo izključno posvetili regresiji. To pomeni, da se bomo ukvarjali z nadzorovanim učenjem v glavnem vnaprejšnjih mrež.

Pomembnejše lastnosti nevronskih mrež

Nevronska mreža je nelinearna preslikava iz vhodnega prostora $\mathcal{R}^n$ v izhodni prostor $\mathcal{R}^m$ podatkov

$$f : \mathcal{R}^n \to \mathcal{R}^n \tag{1.2}$$

pri vhodno-izhodnih podatkih $(\mathbf{x}, y)$ tako, da velja $y = f(\mathbf{x})$. Nevronske mreže so univerzalni aproksimatorji [9]. To pomeni, da lahko vedno najdemo nevronsko mrežo ustreznih dimenzij s katero lahko aproksimiramo izbrano nelinearno funkcijo poljubno natančno.

V literaturi velikokrat najdemo interpretacijo, da ima nevronska mreža sposobnost učenja na primerih. To v bistvu pomeni, da parametre mreže dobimo z optimizacijo glede na izbrano kriterijsko funkcijo, npr.

$$E(k) = \frac{1}{2}\sum_{j=1}^{n}(d_j(k) - y_j(k))^2 \tag{1.3}$$

in z izbranim optimizacijskim algoritmom (učnim pravilom).

Tako naučena nevronska mreža lahko opravlja naučeno preslikavo, kar je velikokrat interpretirano kot sposobnost posploševanja.

Skupna definicija nevronskih mrež ne obstaja. Obstajajo le skupne značilnosti, ki družijo različne modele. Te skupne značilnosti nevronskih mrež so:

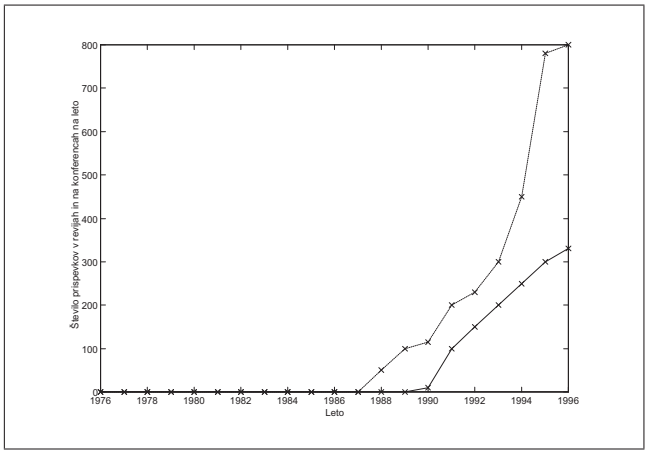
- zgradba iz velikega števila preprostih elementov, ki so medsebojno povezani,
- v splošnem spremenljiva topologija, kar pomeni, da se povezave spreminjajo oziroma spreminjajo se vrednosti uteži na povezavah ali pa se spreminja število elementov mreže,
- zgradba omogoča vzporedno obdelavo informacij,
- informacije se obdelujejo glede na stanje mreže in vhode vanjo.

Veliko več, kot je skupnih značilnosti, je lastnosti, ki nevronske mreže med seboj ločujejo, in sicer:

- metodologija postopkov,
- tipi mrež,
- učna pravila,
- področja uporabe in
- druge lastnosti.

Umetne nevronske mreže danes

Nevronske mreže že dolgo niso več le raziskovalno področje, marveč tudi uporabna tehnologija. Če je bilo to področje od leta 1985 magnet za raziskave (slika 1.5 prikazuje število objav s področja vodenja dinamičnih sistemov v prejšnjem stoletju), je dandanes to tehnologija, ki jo uporabljamo na mnogih področjih. Tudi pri nas je nekaj



Slika 1.5: Rast števila člankov - umetne nevronske mreže v regulacijski zanki (vir: G.W. Ng: Application of Neural Networks to Adaptive Control of Nonlinear Systems)[20]

učbenikov, ki opisujejo nevronske mreže (npr. [2], [3], [10], [11], [15], [23], [28], [31]), in poleg tega še več znanstvenih in strokovnih člankov. Slovenski avtorji so napisali tudi nekaj znanstvenih monografij s tega področja, kot sta na primer [5] in [12].

Naj naštejemo nekaj področij in uporab nevronskih mrež:

- aeronavtika (avtomatski pilot, zaznavanje napak itd.),
- avtomobilska industrija (avtomatsko vodenje itd.),
- bančništvo (razpoznavanje dokumentov itd.),
- elektronika (vodenje, razpoznavanje vzorcev itd.),
- govor (razpoznavanje itd.),
- kosovna industrija (vodenje, napovedovanje itd.),
- medicina (analize signalov itd.),
- računovodstvo (razne analize itd.),
- robotika (vodenje, zaznavanje itd.),
- telekomunikacije (kompresija podatkov itd.),
- transport (sistemi za diagnozo itd.),
- vojaška industrija (obdelava signalov radarjev, sonarjev itd.),
- zabava (animacija, specialni efekti itd.),
- zavarovalništvo (optimizacija vrednosti itd.),
- drugo.

Ali so umetne nevronske mreže izpolnile pričakovanja, ki so jih imeli raziskovalci v zgodnjih obdobjih razvoja?

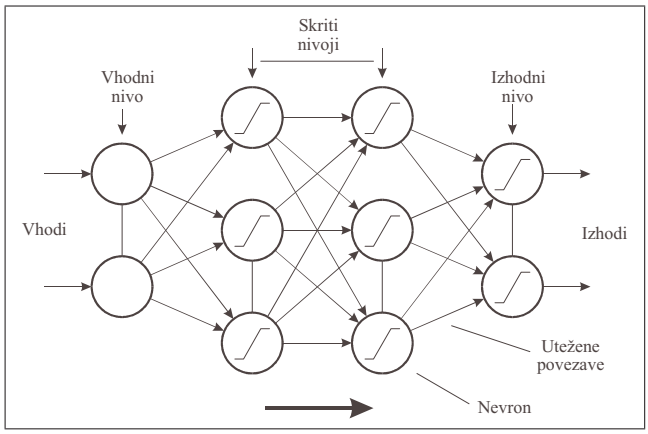
V zgodnjih obdobjih so raziskovalci želeli predvsem razumeti delovanje bioloških sistemov z uporabo matematičnih modelov. Obstajala je zelo močna vez med biološkimi raziskavami in področjem nevronskih mrež. Pričakovanja v tistem času so bila velika. Nevronska mreža naj bi bil zametek umetne inteligence. Kot smo že povedali, se je izkazalo, da temu ni tako. Po eno strani je bilo to razočaranje, po drugi strani smo dobili tehnologijo, ki je zelo primerna tam, kjer potrebujemo učinkovito metodologijo za izdelavo nelinearnih preslikav [14].

Eno izmed področij, na katerem so nevronske mreže zelo uporabne, je tudi modeliranje dinamičnih sistemov in njihova uporaba za načrtovanje avtomatskega vodenja. V glavnem nevronske mreže uporabljamo za identifikacijo nelinearnih dinamičnih sistemov, imamo pa tudi izvedbe avtomatskega vodenja (temu bomo posvetili posebno poglavje).

Obsežni literaturi s področja nevronskih mrež sledi tudi programska oprema, ki jo lahko uporabljamo za delo z nevronskimi mrežami. Med bolj znanimi spada programska oprema na podlagi programskega paketa Matlab (npr. [19] in [21]).

1.3 Večnivojski perceptron

Večnivojski perceptron, ki ga shematično prikazuje slika 1.6, zapišemo z enačbo

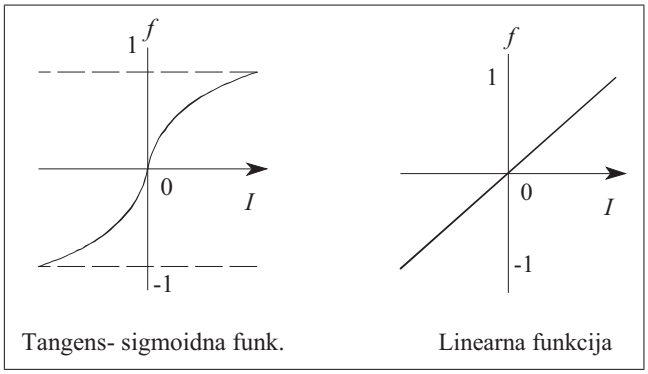


Slika 1.6: Večnivojski perceptron

$$y_i = f_i(\sum_j w_{ij} f_n(\sum_k w_{jk} f_m(\sum_l w_{kl} x_l + w_{0l}^1) + w_{0k}^2) + w_{0j}^3). \tag{1.4}$$

Večnivojski se imenuje zato, ker je sestavljen iz vsaj enega skritega nivoja (aktivacijski funkciji f_n, f_m) in iz izhodnega nivoja f_i. Večnivojski perceptron je univerzalni aproksimator [9]. To velja takrat, kadar ima en ali več skritih nivojev in zaradi tega se vedno več uporablja le en skriti nivo.

Najpogostejše aktivacijske funkcije (primera na sliki 1.7) so:



Slika 1.7: Dva primera aktivacijskih funkcij

- sigmoidna,
- hiperbolični tangens,
- linearna,
- Gaussova,
- funkcija signum.

V skritem nivoju je aktivacijska funkcija pogosto sigmoidna

$$f(I) = \frac{1}{1 + e^{-aI}} \ , \tag{1.5}$$

medtem ko je najpogostejša izhodna funkcija linearna (slika 1.7).

Večnivojski perceptron je tipičen primer nevronske mreže, ki ga uporabljamo za regresijo. Poglejmo si dva primera, kako lahko aproksimiramo nelinearni funkciji z večnivojskim perceptronom.

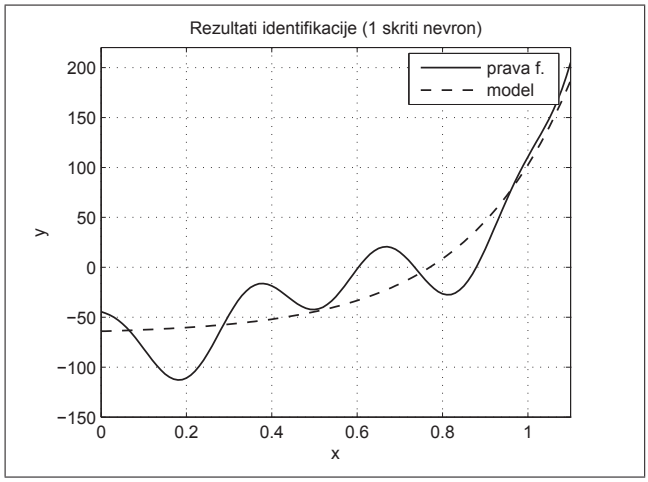
Primer aproksimacije funkcije ene spremenljivke

Funkcijo

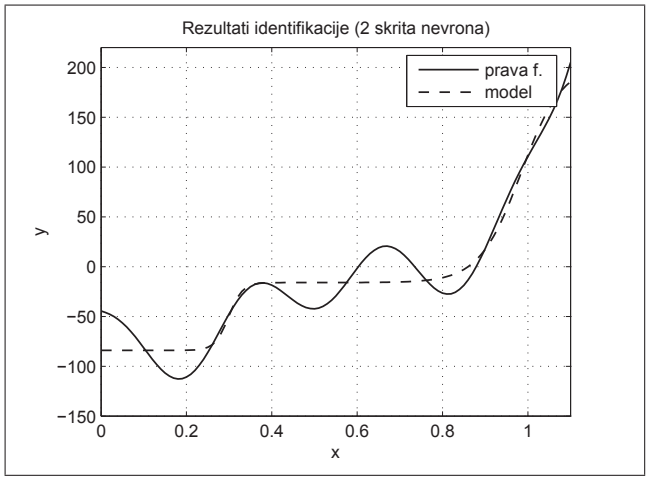
$$\begin{aligned} y &= 30\left[100(x-0.6)^2(x-0.1)(x-0.8) - 5e^{-5x}\right. \\ &+ \left. 0.9\sin(20x + \frac{\pi}{4})\right] \end{aligned} \tag{1.6}$$

bomo aproksimirali z večnivojskim perceptronom z enim skritim nivojem. Aktivacijska funkcija skritega nivoja je sigmoidna funkcija, medtem ko je izhodnega linearna funkcija.

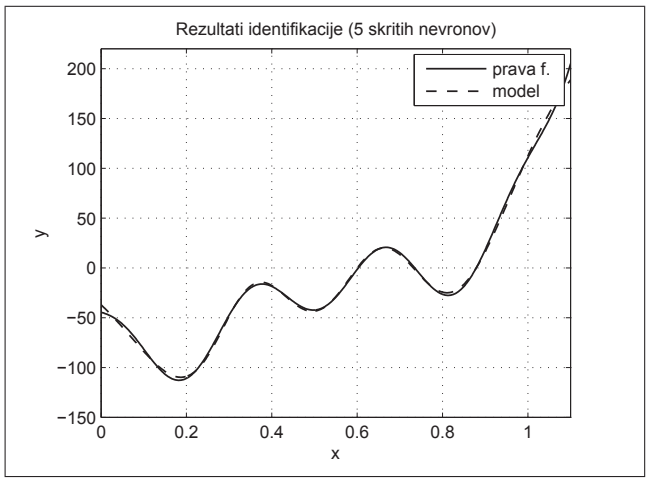
Funkcijo smo aproksimirali v več korakih tako, da smo najprej optimirali uteži mreže z enim nevronom v skritem nivoju, potem z dvema in tako smo povečevali število nevronov skritega nivoja, dokler nismo dosegli želene stopnje aproksimacije. Za učenje - optimizacijo smo uporabili Levenberg-Marquardtovo modifikacijo Newtonove optimizacijske metode ali kratko Levenberg-Marquardtovo učno metodo. Rezultate optimizacije prikazujejo slike 1.8 do 1.10. Na koncu smo dobili nevronsko mrežo s petimi nevroni v skritem nivoju.



Slika 1.8: Večnivojski perceptron z enim nevronom v skritem nivoju



Slika 1.9: Večnivojski perceptron z dvema nevronoma v skritem nivoju



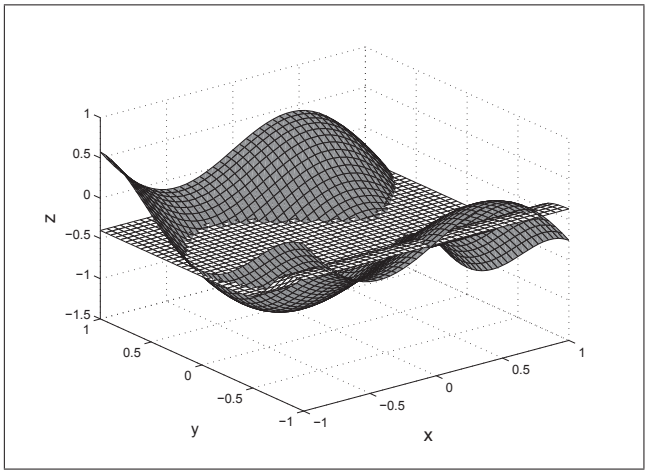
Slika 1.10: Večnivojski perceptron s petimi nevroni v skritem nivoju

Primer aproksimacije funkcije dveh spremenljivk

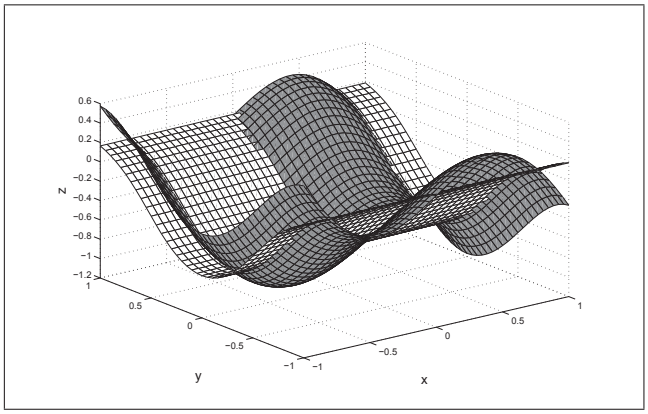
Funkcijo

$$z(x, y) = x\cos(2x) + y\sin(2y) - 0.75 \tag{1.7}$$

bomo zopet aproksimirali z večnivojskim perceptronom z enim skritim nivojem. Tudi tokrat bo aktivacijska funkcija skritega nivoja sigmoidna funkcija, izhodnega pa linearna. Tudi tokrat smo stopnjevali strukturo skritega nivoja od enega do petih nevronov. Rezultate prikazujejo slike 1.11 do 1.13.



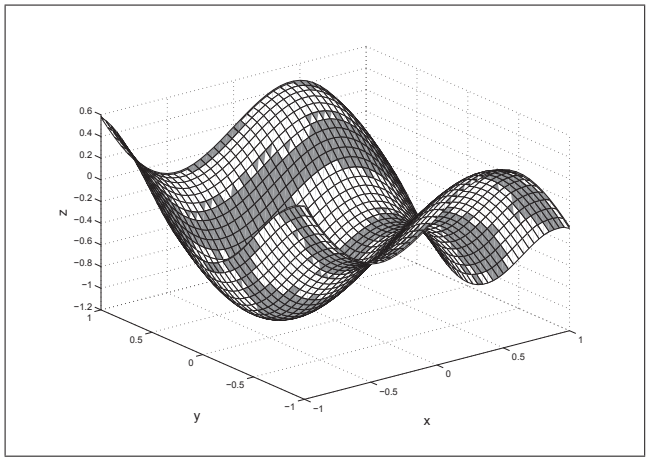
Slika 1.11: Večnivojski perceptron z enim nevronom v skritem nivoju (ciljna površina je temno sive barve, naučena površina je bele barve)



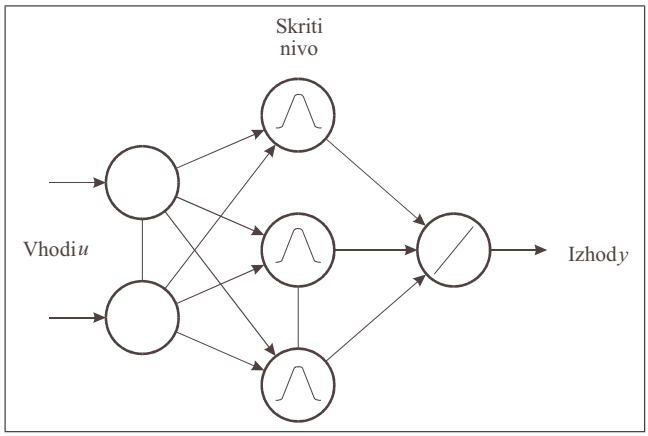
Slika 1.12: Večnivojski perceptron z dvema nevronoma v skritem nivoju (ciljna površina je temno sive barve, naučena površina je bele barve)

1.4 Mreža radialnih baznih funkcij

Mreža radialnih baznih funkcij (RBF angl. radial basis function network) je druga najpogosteje uporabljena nevronska mreža. Mreža vsebuje nelinearne pretvorbe vhodnih podatkov, ki z uteženo vsoto tvorijo izhod mreže, prikazane na sliki 1.14. Nevronska mreža izvede aproksimacijsko funkcijo $f : \mathcal{R}^m \to \mathcal{R}$, ki je matematično opisana kot:



Slika 1.13: Večnivojski perceptron s petimi nevroni v skritem nivoju (ciljna površina je temno sive barve, naučena površina je bele barve)



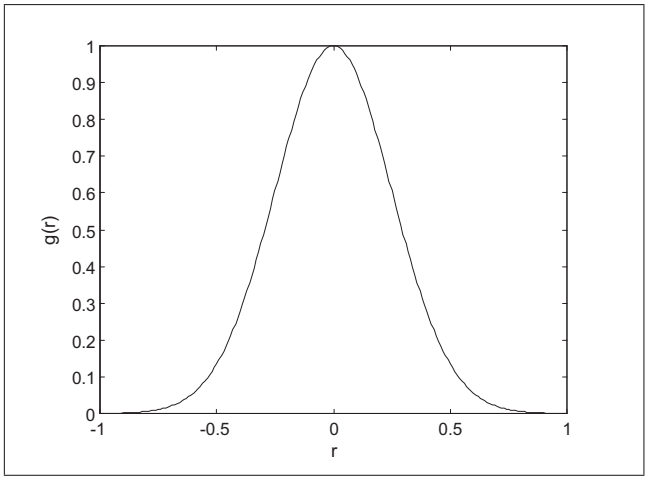
Slika 1.14: Mreža radialnih baznih funkcij

$$y = f(\mathbf{x}) = \sum_{i=1}^{N} w_i g_i(\mathbf{x}, \gamma_i). \tag{1.8}$$

Funkcija g se imenuje bazna funkcija in je v primeru te nevronske mreže neka radialna funkcija. Možne radialne funkcije so:

- $g(r) = r$, linearna radialna funkcija,
- $g(r) = r^2$, kvadratna funkcija,
- $g(r) = exp(\frac{-r^2}{\rho^2})$, Gaussova funkcija (slika 1.15),
- $g(r) = [1 + exp(\frac{r^2}{\rho^2}]^{-1})$, logistična funkcija,
- $g(r) = r^2 \log(r)$, razni zlepki (angl. spline function),

- $g(r) = (r^2 + \rho^2)^{\frac{1}{2}}$, večkvadratna funkcija,
- drugo.

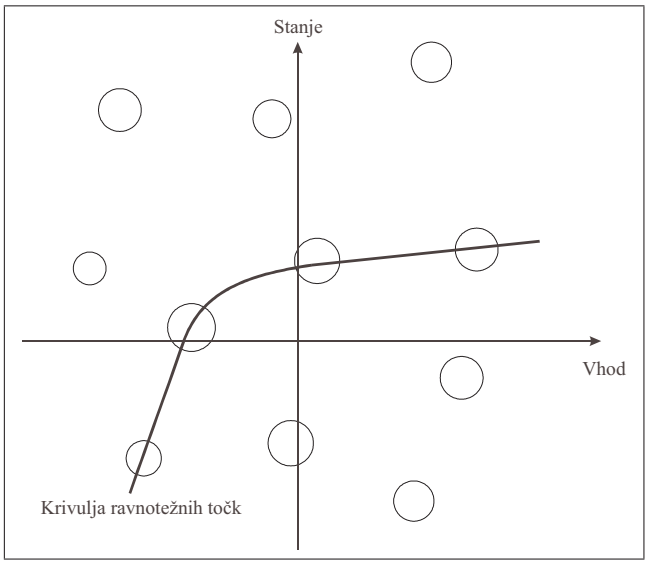


Slika 1.15: Gaussova bazna funkcija

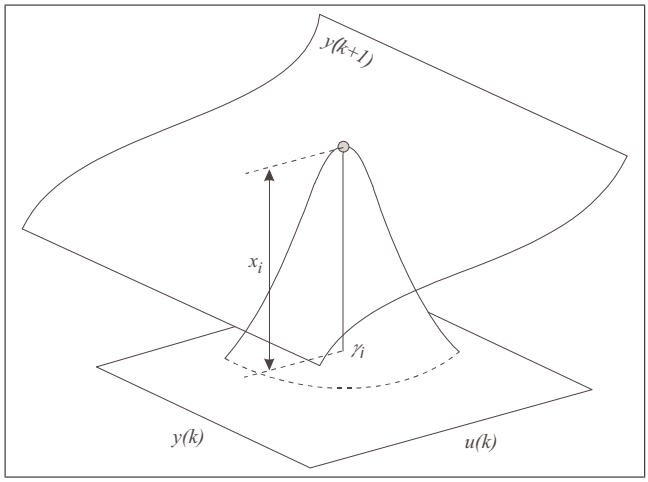
Za več vhodov bi bila na primer Gaussova bazna funkcija videti taka:

$$g_i = \exp\left[-\frac{1}{2}\left(\frac{(x_1-\gamma_{i1})^2}{\rho_{i1}^2} + \frac{(x_2-\gamma_{i2})^2}{\rho_{i2}^2} + \dots + \frac{(x_n-\gamma_{in})^2}{\rho_{in}^2}\right)\right]. \quad (1.9)$$

Radialne funkcije so pogosto usrediščene v N točk vhodnih podatkov, npr. $\gamma_i = \mathbf{x}_i \quad i = 1 \dots n \leq N$. Lahko pa je njihov položaj in izbira radijev ρ_i del optimizacijskega postopka oziroma učenja. Primer porazdelitve in ponazoritev, kako je postavljena radialna funkcija, prikazujeta iz različnih perspektiv sliki 1.16 in 1.17.



Slika 1.16: Porazdelitev centrov mreže RBF



Slika 1.17: Posamezna RBF glede na nelinearnost

Pomembna lastnost mreže radialnih baznih funkcij je njena stabilnost, ki je odvisna od izbire radialnih funkcij. Ker imajo npr. Gaussove bazne funkcije lastnost, da se bližajo vrednosti nič, ko se oddaljujejo od središča, pomeni, da je tudi izhod nevronske mreže vedno omejen, če imajo uteži končne vrednosti. Ta BIBO stabilnost (angl. bounded input bounded output) je tako pomembna, da so bazne funkcije zaradi tega v glavnem Gaussove funkcije. Tudi mreže radialnih baznih funkcij so univerzalni aproksimatorji [22].

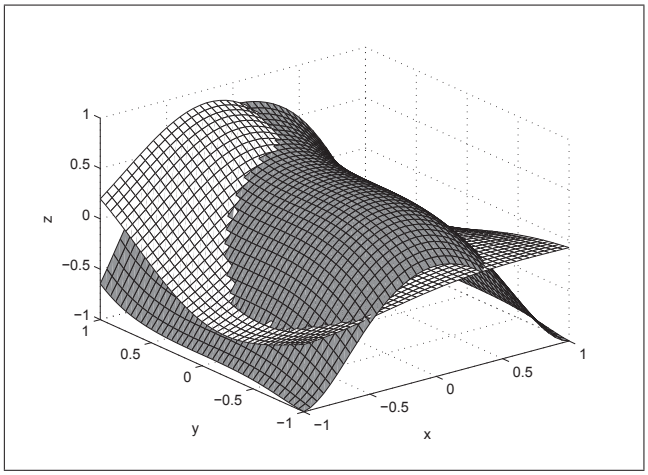
Ker se uteži w_i spreminjajo linearno glede na izhod, jim ni tako težko najti optimalne vrednosti glede na kriterij najmanjših kvadratov napake. Mreža radialnih baznih funkcij vsebuje dodatne parametre γ_i in ρ, katerih vrednosti so v splošnem neznane. Če optimiziramo tudi vrednosti teh parametrov, potem imamo opravka z nelinearnim optimizacijskim problemom, kot je bil to primer z večnivojskim perceptronom. Zelo uspešen optimizacijski pristop v tem primeru je metoda ortogonalnih najmanjših kvadratov (angl. ortogonal least square - OLS)[4].

Primer funkcije dveh spremenljivk

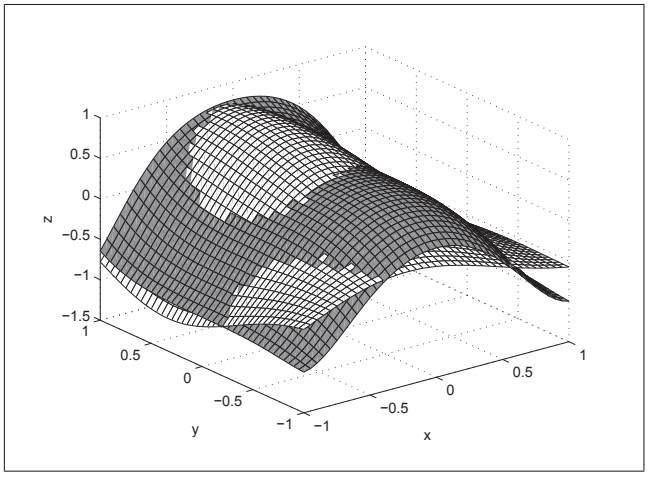
Funkcijo

$$z(x, y) = \sin(0.4x^3 - 1.6y^2 + 0.5) \quad (1.10)$$

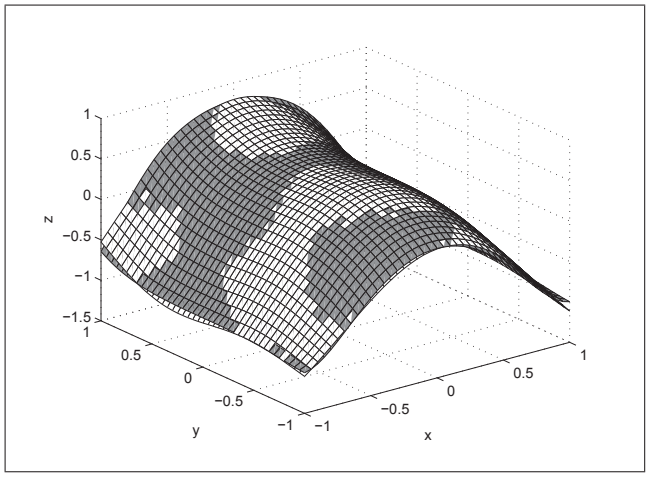
bomo aproksimirali z mrežo baznih radialnih funkcij. Bazna funkcija bo Gaussova funkcija. Optimirali bomo uteži mreže z različnim številom baznih funkcij širine $\rho = 0.1$, ki jih bomo povečevali, dokler ne bomo dovolj dobro aproksimirali dano funkcijo $z(x, y)$. Rezultati so prikazani na slikah od 1.18 do 1.21.



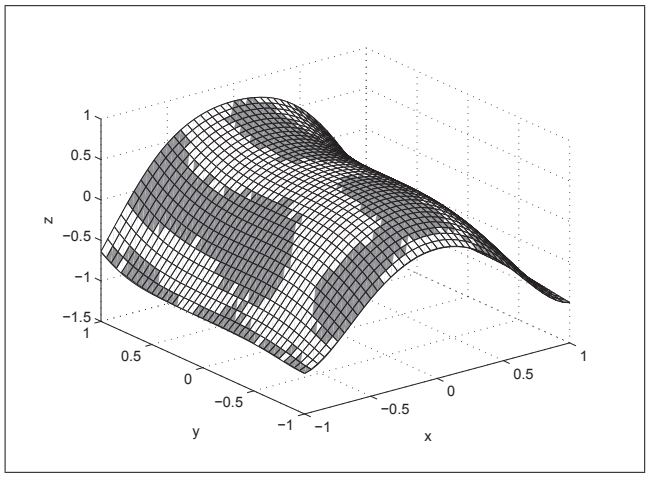
Slika 1.18: Mreža s tremi radialnimi baznimi funkcijami (ciljna površina je temno sive barve, naučena površina je bele barve)



Slika 1.19: Mreža s petimi radialnimi baznimi funkcijami (ciljna površina je temno sive barve, naučena površina je bele barve)



Slika 1.20: Mreža z enajstimi radialnimi baznimi funkcijami (ciljna površina je temno sive barve, naučena površina je bele barve)



Slika 1.21: Mreža z enaintridesetimi radialnimi baznimi funkcijami (ciljna površina je temno sive barve, naučena površina je bele barve)

1.5 Nevronska mreža za opis nelinearnega dinamičnega sistema

Nevronske mreže, kot smo jih obravnavali, opisujejo statične nelinearne funkcije. Njihove uteži optimiramo tako, da dobimo najbolj ustrezno povezavo med vhodnimi in izhodnimi podatki. Ko damo na vhod nove podatke, mreža, ki ponazarja preslikavo, poišče ustrezne izhodne vrednosti.

Pri dinamičnih sistemih imamo opravka z nelinearnostjo, ki na izhodu daje vrednost, ki ni odvisna le od vrednosti na vhodu, marveč tudi od stanj sistema, ki so ponavadi pretekle vrednosti vhodnega in izhodnega signala oziroma njunih odvodov (odvisno od tega, ali imamo opravka z diskretnim ali zveznim sistemom).

Običajno ponazarjanje dinamičnih sistemov je tako, da spomin (ali odvajanje) izvedemo zunaj aproksimatorja, v našem primeru nevronske mreže, in potem na vhod pripeljemo tudi zakasnjene (ali odvajane) vrednosti vhodov in izhodov. Pri dinamičnih sistemih po navadi gledamo vhodno-izhodno obnašanje prek signalov oziroma časovnih vrst podatkov za razliko od statičnih sistemov, pri katerih se ukvarjamo s posameznimi vhodnimi in izhodnimi vrednostmi. Ker gre za iste vrste nevronskih mrež in druge sorodne metode, je rešitev ta, da mreža opravi svojo funkcijo v vsakem časovnem koraku posebej. To pomeni, da v izbranem časovnem trenutku optimiramo nevronsko mrežo tako, da pri vrednostih vhodov, ki so sestavljeni iz vzorca vhodnega signala in izbranega števila vzorcev vhodnega in izhodnega signala iz predhodnih trenutkov (zakasnjenih vzorcev), daje vrednost napovedi odziva dinamičnega sistema v naslednjem trenutku. To lahko ponazorimo z enačbo (1.11)

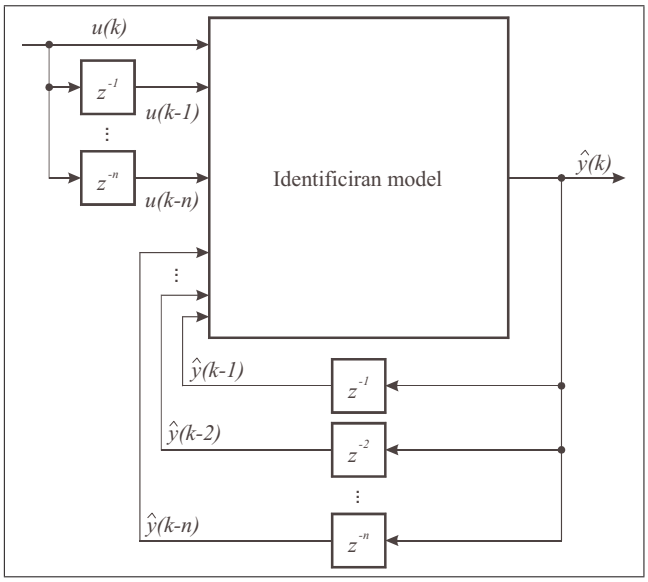
$$\hat{y}(k+1) = f(y(k), y(k-1), \ldots, u(k), \ldots), \qquad (1.11)$$

kjer je s simbolom ˆ označena napoved in je k zaporedno število vzorca. Tak način se imenuje enokoračna predikcija. Podrobneje bomo o tem govorili v naslednjih poglavjih.

Ko preverjamo obnašanje dinamičnega sistema, naredimo to z večkoračno predikcijo oziroma simulacijo, to pomeni, da preizkušamo obnašanje mreže s celotnim signalom pri čemer neznane vrednosti izhodnega signala zamenjamo z njihovimi napovedmi. Idejo o izvedbi simulacije nevronske mreže, ki opisuje dinamični sistem, prikazuje slika 1.22 in jo lahko ponazorimo z enačbo

$$\hat{y}(k+1) = f(\hat{y}(k), \hat{y}(k-1), \ldots, u(k), \ldots). \tag{1.12}$$

Ko nevronsko mrežo optimiramo, imamo opravka z nadzorovanim učenjem vnaprejšnje mreže. Model dinamičnega sistema, ponazorjen s to nevronsko mrežo, pa je zaradi povratnih zvez povratna mreža.



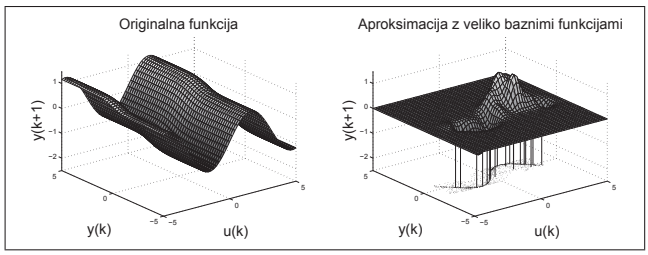
Slika 1.22: Prikaz simulacije identificiranega nelinearnega dinamičnega sistema

Primer aproksimacije dinamičnega sistema prvega reda z mrežo RBF

Z mrežo RBF želimo ponazoriti dinamični sistem, ki ga opisuje diferenčna enačba:

$$y(k+1) = 0.2\tanh(y(k)) + \sin(u(k)). \tag{1.13}$$

Vhodni in izhodni signal, ki ju uporabimo pri učenju nevronske mreže, sta vsak posebej sestavljena iz 2207 vzorcev. Dinamični sistem bomo modelirali z nevronsko mrežo, ki bo vsebovala 20 Gaussovih baznih funkcij z $\rho = 0.3$ naključno razmeščenih v ravnini v področju, v katerem so vrednosti vhodnega in izhodnega signala. Več o tem, kako identificiramo dinamične sisteme z nevronsko mrežo, bo govora v naslednjih poglavjih. Poglejmo si, kako je nevronska mreža ponazorila nelinearnost $y(k+1)$ pri vhodih $y(k)$ in $u(k)$. Rezultate prikazuje slika 1.23. Na prvi pogled je videti ponazoritev, ki je prikazana na desni strani slike 1.23, zelo slaba, vendar se pri podrobnejšem ogledu izkaže, da je dovolj dobra v področju, v katerem so bili na voljo podatki za učenje nevronske mreže. Primer zelo nazorno pokaže, kako pomembno je, da se zavedamo omejenih možnosti nevronskih mrež, obenem pa nakazuje kompleksnost identifikacije dinamičnih sistemov z nevronskimi mrežami.



Slika 1.23: Nelinearni dinamični sistem (leva slika) in njegova ponazoritev z mrežo RBF (desna slika)[6]

Literatura

[1] Artificial Neural Network, [Elektronski vir]. http://en.wikipedia.org/wiki/Artificial_neural_network

[2] J. Balič (2004): Inteligentni obdelovalni sistemi, Fakulteta za strojništvo, Maribor.

[3] A. Dobnikar (1990): Nevronske mreže: teorija in aplikacije, Didakta, Radovljica.

[4] A. P. Englebrecht (2002): Computational intelligence, An Introduction, Wiley and Sons, Chichester.

[5] I. Grabec, W. Sachse (1997): Synergetics of measurement, prediction and control, Springer Series in Synergetics, Springer-Verlag Berlin and Heidelberg GmbH & Co.

[6] G. Gregorčič (2004): Data-based modelling of nonlinear systems for control, doktorska disertacija, University College Cork, National University of Ireland, Cork.

[7] S. Haykin (1994): Neural Networks: A Comprehensive Foundation, Macmillian Publishing Company, Inc.

[8] D. O. Hebb (1949): The Organization of Behaviour, Wiley, New York.

[9] K. Hornik (1989): Multilayer Feedforward Networks are Universal Approximators, Neural Networks, Vol. 2, 359-366.

[10] P. Kokol, Š. Hleb Babič, V. Podgorelec, M. Zorman (2001): Inteligentni sistemi, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor.

[11] I. Kononenko (2005): Strojno učenje, Fakulteta za računalništvo in informatiko, Ljubljana.

[12] I. Kononenko, M. Kukar (2007): Machine learning and data mining: Introduction to principles and algorithms, Horwood Publishing, Chichester.

[13] B. J. A. Kröse and P. P. van der Smagt (1994): An Introduction to Neural Networks, Seventh edition, The University of Amsterdam, [Elektronski vir]. http://staff.science.uva.nl/~krose/publications/index_en.html

[14] D. J. Mackay (2003): Information theory, inference and learning algorithms, Cambridge University Press.

[15] D. Matko, S. Strmčnik, B. Zupančič, G. Mušič (1995): Računalniško vodenje procesov, Poglavje 4.3, Založba FER, Ljubljana.

[16] W. S. McCulloch, W. Pitts (1943): A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics, 5, 115-133.

[17] M. Minsky, S. Papert (1969): Perceptrons: An Introduction to Computational Geometry, The MIT press.

[18] K. S. Narendra, K. Parthasarathy (1990): Identification and Control of Dynamical Systems using Neural Networks, IEEE Transactions on Neural Networks, Vol. 1, No. 1, 4-27.

[19] Neural Network Toolbox for Matlab http://www.mathworks.com/products/neuralnet/

[20] G. W. Ng (1997): Application of Neural Networks to Adaptive Control of Nonlinear Systems, UMIST Control Systems Centre Series, Vol. 4, Research Studies Press, Manchester.

[21] The NNSYSID Toolbox for use with Matlab http://www.iau.dtu.dk/research/control/nnsysid.html

[22] J. Park , I. W. Sandberg (1991): Universal approximation using radial-basis-function networks, Neural Computation, Vol. 3, No. 2, 246-257.

[23] B. Potočnik (2007): Osnove razpoznavanja vzorcev z nevronskimi mrežami, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor.

[24] D. Psaltis, A. Sideris, A. A. Yamamura (1988): A multilayer neural network controller, IEEE Control Systems Magazine, Vol. 8, No. 2, 17-21.

[25] F. Rosenblatt (1959): Principles of Neurodynamics, Spartan Books, New York.

[26] D. E. Rumelhart, J. L. McClelland (1986): Parallel Distributed Processing: Explorations in the Microstructure of Cognition, The MIT press.

[27] J. Sjöberg et al. (1995): Non-linear Black-Box Modeling in System Identification: a Unified Overview, Automatica, Vol. 31, No. 12, 1691-1724.

[28] R. Šafarič, A. Rojko (2005): Inteligentne regulacijske tehnike v mehatroniki, [Elektronski vir]. http://www.ro.feri.uni-mb.si/predmeti/int_reg/vsebina.html

[29] B. Widrow, M. E. Hoff (1960): Adaptive switching circuits. In 1960 IRE WESCON Convention Record, 96-104, DUNNO.

[30] B. Widrow, F. W. Smith (1963): Pattern Recognizing Control Systems, Computer and Information Sciences (COINS) Symposium Proceedings, Spartan Books, 351-357.

[31] M. Zorman, V. Podgorelec, M. Lenič, P. Povalej, P. Kokol, A. Tapajner (2003): Inteligentni sistemi in profesionalni vsakdan, Center za interdisciplinarne in multidisciplinarne raziskave in študije, Univerza v Mariboru.

2

Identifikacija linearnih dinamičnih sistemov

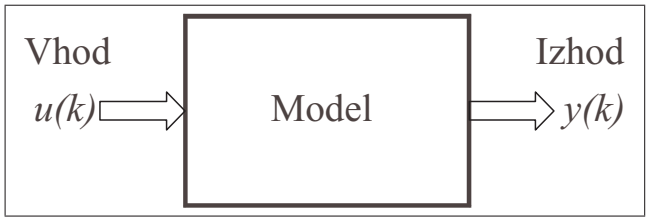
2.1 Povzetek o identifikaciji linearnih dinamičnih sistemov

Namen tega poglavja je povzeti temelje identifikacij linearnih sistemov zato, da bi lahko znanje nadgradili oziroma uporabili za razlago pristopov k identifikaciji nelinearnih sistemov. Podrobno o teoriji in praksi identifikacije linearnih dinamičnih sistemov najdemo npr. v delih [10], [2], [7], [3] idr.

Identifikacija je eksperimentalna določitev časovnega obnašanja procesa z uporabo merjenih signalov. Časovno obnašanje je definirano znotraj nekega razreda matematičnih modelov in ponazarja identificirani proces tako, da so razlike med procesom in njegovim matematičnim modelom čim manjše [10].

Poglavitna naloga pri identifikaciji, ki je oblika eksperimentalne analize sistemov, je najti primerno strukturo modela, ki določa razred modelov, v katerem model iščemo. Optimizacija parametrov te strukture je naslednji korak. Temeljno pravilo pri identifikaciji je, da ne identificiramo delov procesa, ki jih že poznamo. Predznanje o procesu, naj bo fizikalni vpogled ali znanje, pridobljeno z izkušnjami, je potrebno izkoristiti. Ena izmed delitev matematičnih modelov, ki to nakazuje, je po barvni kodi naslednja:

- model »bele škatle« (angl. white-box model), tudi teoretični model, ki ga dobimo iz fizikalnega, kemijskega ipd. vpogleda v dogajanje v procesu;
- model »sive škatle« (angl. grey-box model), tudi kombinirani model, pri katerem imamo nekaj predznanja, ki je lahko v obliki
  - celotne ali delne strukture modela, parametre pa je potrebno oceniti iz meritev;
  - poznavanja kombinacij merjenih signalov ali njihovih funkcijskih povezanosti, s katerimi si lahko pomagamo pri modeliranju;
- model »črne škatle« (angl. black-box model), tudi eksperimentalni model, pri katerem identificiramo strukturo in parametre, saj o procesu nimamo predznanj. Do modela procesa pridemo samo z analizo vhodnih in izhodnih signalov procesa (slika 2.1).



Slika 2.1: Proces kot dinamični sistem z vhodnimi in izhodnimi signali

Teorija in praksa identifikacij dinamičnih sistemov je skupek rezultatov znanstvenih dognanj in inženirske prakse. Na tem področju je poleg omenjene literature na voljo tudi precej programske opreme za identifikacijo.

Identifikacijske metode lahko delimo na veliko načinov. Poglejmo si le nekatere od delitev glede na različne vidike [10]:

- Razred matematičnih modelov:
  - neparametrični modeli,
  - parametrični modeli.
- Razred uporabljenih signalov:
  - zvezni, diskretni,
  - deterministični, naključni, psevdonaključni.
- Pogrešek med procesom in modelom:
  - izhodni,
  - vhodni,
  - posplošeni.
- Sočasnost meritev in vrednotenja:
  - nesprotne,
  - sprotne.
- Postopek obdelave podatkov:
  - nerekurziven -
    - direktni,
    - iterativni,

  - rekurziven.
- Struktura modelov:
  - linearni modeli,
  - nelinearni modeli.

Celoten postopek identifikacije je opisan npr. v literaturi [10], [2], [7]. Osnovne korake identifikacije lahko povzamemo kot:

1. določitev uporabe oziroma namena modela,
2. zbiranje predznanj,
3. načrtovanje meritev,
4. izvedba meritev,
5. identifikacija modela in
6. preskus identificiranega modela.

Celoten postopek je zelo iterativen in interaktiven. Za inženirsko prakso identifikacij so posebno pomembni koraki, v katerih določimo oziroma izberemo uporabo modela, meritve podatkov in vrednotenje modela.

Ko eksperimentalno modeliramo - identificiramo neki sistem, je težavnostna stopnja problema zelo odvisna od tega, ali imamo opravka s statičnim (enačba (2.1)) ali dinamičnim (enačbi (2.2) in (2.3)) sistemom. Dinamika sistema bistveno poveča zahtevnost problema.

- Statični sistem:

$$F[u(t), y(t)] = 0. \tag{2.1}$$

- Dinamični sistem:

$$\begin{aligned} &F[t, u(t), \dot{u}(t), \ddot{u}(t), \ldots, u^{(m)}(t), \\ &y(t), \dot{y}(t), \ddot{y}(t), \ldots, y^{(n)}(t)] = 0. \end{aligned} \tag{2.2}$$

$$\begin{aligned} &F[k, u(k), u(k-1), u(k-2), \ldots, u(k-m), \\ &y(k), y(k-1), y(k-2), \ldots, y(k-n)] = 0. \end{aligned} \tag{2.3}$$

Metode identifikacije pogosto delimo po razredu modelov. Razdelimo jih na:

- neparametrične in
- parametrične modele.

Neparametrični so najpogosteje modeli linearnih sistemov. Vhodno-izhodno obnašanje procesa opisujejo v obliki tabel vrednosti ali krivulje. Tipični predstavniki so:

- frekvenčni odzivi (Bodejevi diagrami),
- utežne funkcije, odzivi na stopnico,
- Fourierova analiza, analiza frekvenčnega odziva, korelacijska analiza, spektralna analiza.

Parametrične modele uporabljamo za modeliranje tako linearnih kot nelinearnih dinamičnih sistemov. To so modeli z eksplicitno izraženimi parametri. Najpogosteje so to modeli, zapisani v obliki:

- diferenčnih enačb,
- diferencialnih enačb,
- prenosnih funkcij,
- zapisa v prostoru stanj.

Pri identifikaciji linearnega modela sistema moramo za take modele določiti red linearnega modela; ta je določen z redom enačb, ki opisujejo sistem in regresorje. Regresorji so tiste veličine, od katerih je funkcijsko odvisna napoved modela. Parametre določimo z optimizacijo, ki ima najpogosteje za kriterij vsoto kvadratov najmanjše napake (metode najmanjših kvadratov), čeprav se uporabljajo tudi drugi kriteriji, npr. največja verjetnost (metoda največje podobnosti).

Linearne sisteme lahko prikažemo z naslednjo enačbo v z prostoru [9]:

$$A(z^{-1})y(z) = \frac{B(z^{-1})}{F(z^{-1})}u(z) + \frac{C(z^{-1})}{D(z^{-1})}v(z), \tag{2.4}$$

kjer so A, B, C, D, F polinomi kompleksne spremenljivke, z, $y(z)$, $u(y)$, $v(z)$ pa transformi diskretnega izhodnega signala, vhodnega signala in signala šuma. V odvisnosti od tega, katere regresorje smo izbrali za opis sistema, so nekateri od teh polinomov enaki 0 ali 1.

Metode identifikacije se razlikujejo po uporabljenih regresorjih in/ali po kriteriju za optimizacijo, pri čemer bomo v nadaljevanju za optimizacijo predpostavili metodo najmanjših kvadratov:

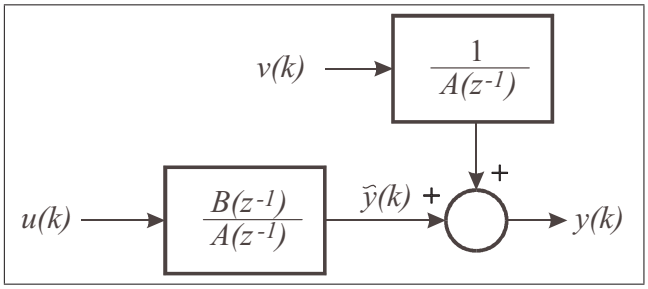
- metoda končnega impulznega odziva (angl. finite impulse response - FIR) ($A = F = D = 1, C = 0$), to pomeni, da so regresorji samo zakasnjeni vhodni signali,
- metoda najmanjših kvadratov s posplošenim pogreškom (angl. autoregressive model with exogenous input - ARX) ($F = C = D = 1$), to pomeni, da so regresorji zakasnjeni vhodni in izhodni signali,
- metoda z izhodnim pogreškom (angl. output error method - OE) ($A = C = D = 1$), to pomeni, da so regresorji zakasnjene vrednosti vhodnega signala in ocen izhodnega signala (predikcij iz preteklosti),

- metoda razširjenih najmanjših kvadratov (angl. autoregressive and moving average model with exogenous input - ARMAX) ($F = D = 1$), kjer so regresorji zakasnjeni vhodni, izhodni in šumni signali,
- metoda Box - Jenkins (BJ) ($A = 1$), ki ima za regresorje zakasnjene vhodne signale, zakasnjene ocene izhodnega signala, napako napovedi in napako simulirane napovedi (ko smo za napoved uporabljali ocene izhodnega signala),
- možna je tudi predstavitev v prostoru stanj, kjer linearni sistem zapišemo v obliki : $\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \mathbf{B}\mathbf{u}(k-1)$,
- drugi možni regresorji.

Poleg opisanih metod obstaja še večje število njihovih različic. Opisane metode se razlikujejo predvsem po tem, kako v model vključujejo šum, ki se ga pri realnih sistemih ne moremo izogniti. Poglejmo si nekaj izbranih modelov linearnih dinamičnih sistemov (slike 2.2, 2.3 in 2.4):

- metoda najmanjših kvadratov s posplošenim pogreškom (ARX)

$$\begin{aligned} y(k) &= a_1 y(k-1) + a_2 y(k-2) \\ &+ b_1 u(k-1) + b_2 u(k-2) \\ &+ v(k); \end{aligned} \tag{2.5}$$



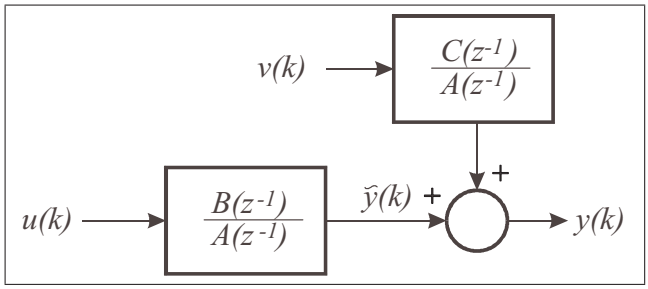
Slika 2.2: Metoda najmanjših kvadratov s posplošenim pogreškom (ARX)

- metoda najmanjših kvadratov z modelom šuma (ARMAX)

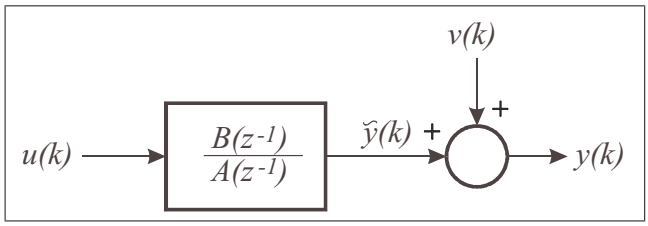
$$\begin{aligned} y(k) &= a_1 y(k-1) + a_2 y(k-2) \\ &+ b_1 u(k-1) + b_2 u(k-2) \\ &+ v(k) + c_1 v(k-1) + c_2 v(k-2); \end{aligned} \tag{2.6}$$

- metoda z izhodnim pogreškom (OE)

$$\begin{aligned} y(k) &= a_1[y(k-1) - v(k-1)] \\ &+ a_2[y(k-2) - v(k-2)] \\ &+ b_1 u(k-1) + b_2 u(k-2) \\ &+ v(k). \end{aligned} \tag{2.7}$$



Slika 2.3: Metoda najmanjših kvadratov z modelom šuma (ARMAX)



Slika 2.4: Metoda z izhodnim pogreškom (OE)

Vsaka izmed metod ima tudi svoje rekurzivne različice za sprotno identifikacijo [10], [2], [7], [3].

Primer identifikacije z metodo najmanjših kvadratov s posplošenim pogreškom

Na enostavnem primeru identifikacije sistema prvega reda si poglejmo delovanje metode najmanjših kvadratov s posplošenim pogreškom. Sistem, ki bi ga radi identificirali, opisuje diferenčna enačba:

$$y(k) = 0.9512y(k-1) + 0.09754u(k-1) \tag{2.8}$$

oziroma prenosna funkcija

$$H(z) = \frac{0.09754z^{-1}}{1 - 0.9512z^{-1}} \tag{2.9}$$

in ni moten s šumom.

Gre za sistem prvega reda in napoved bo odvisna od ene zakasnjene vrednosti izhoda in ene zakasnjene vrednosti vhoda. Regresorja sta torej : $y(k-1), u(k-1)$. Strukturo modela postavimo z enačbo za enokoračno napoved (napoved vrednosti na podlagi znanih vrednosti regresorjev v prejšnjih trentukih):

$$y(k) = -a_1 y(k-1) + b_1 u(k-1). \tag{2.10}$$

Če vstavimo namesto regresorjev $(y(k-1), u(k-1))$ in predikcij $(y(k))$ merjene vrednosti signalov, lahko zapišemo enačbo (2.10) v matrični obliki:

$$\begin{aligned} \mathbf{y} &= \boldsymbol{\Psi\theta} \\ \begin{bmatrix} y(2) \\ y(3) \\ \vdots \end{bmatrix} &= \begin{bmatrix} -y(1) & u(1) \\ -y(2) & u(2) \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \end{aligned} \tag{2.11}$$

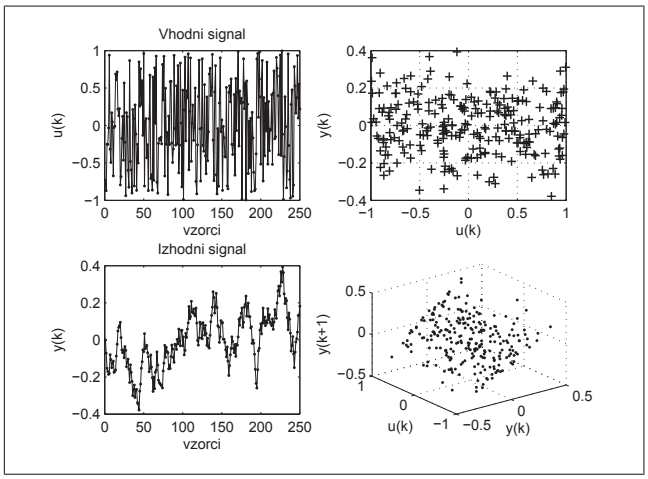
Vedeti moramo, da je vrstni red vrstic lahko tudi zamenjan, saj optimiramo najboljšo predikcijo glede na posamezne vrednosti signala in ne na celoten signal.

Predoločen sistem enačb rešimo po metodi najmanjših kvadratov:

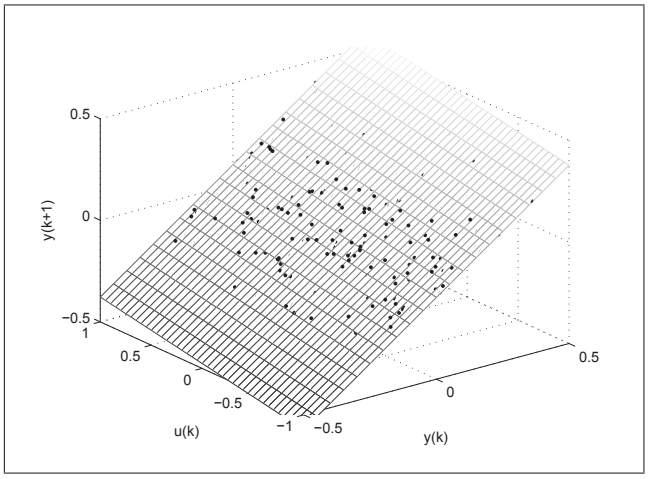
$$\hat{\boldsymbol{\theta}} = [\boldsymbol{\Psi}^T\boldsymbol{\Psi}]^{-1}\boldsymbol{\Psi}^T\mathbf{y}. \tag{2.12}$$

Parametre smo sicer optimirali za enokoračno predikcijo, ker je to lastnost metod na podlagi modela s posplošenim pogreškom, vendar jo moramo preskusiti s simulacijo (večkoračno predikcijo), da bi ugotovili, ali model res zadovoljivo povzame dinamiko sistema.

Na slikah 2.5 in 2.6 vidimo uporabljene podatke in rezultat identifikacije parametrov.



Slika 2.5: Vhodni in izhodni signal (sliki levo), pokritost področja z merjenimi podatki (sliki desno)



Slika 2.6: Ravnina, ki prikazuje identificiran model in podatke za identifikacijo

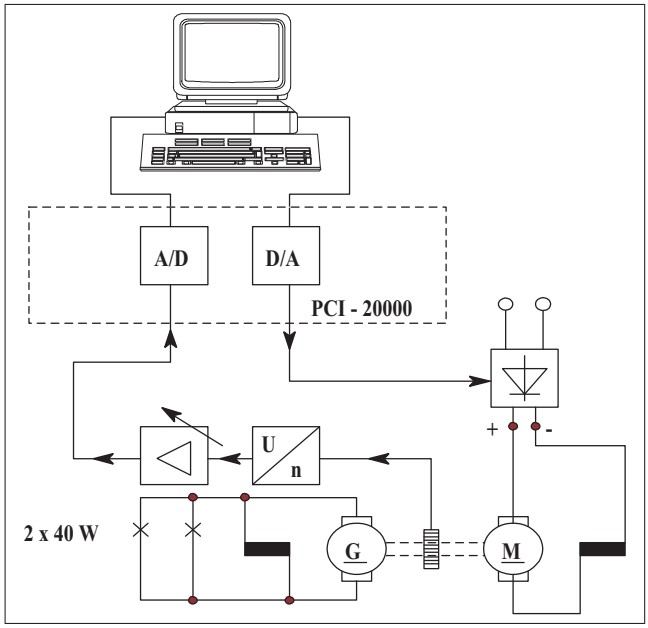
V tem primeru smo se osredotočili izključno na metodo identifikacije. Celoten postopek identifikacije bomo prikazali v naslednjem primeru.

2.2 Primer identifikacije mehatronskega sistema

Namen tega primera je prikazati celoten proces identifikacije linearnega dinamičnega sistema v praksi in v njem osvetliti vprašanja, ki se pri tem pojavljajo. Primer smo vzeli iz [4], kjer je v nekoliko podrobnejši različici.

Opis procesa

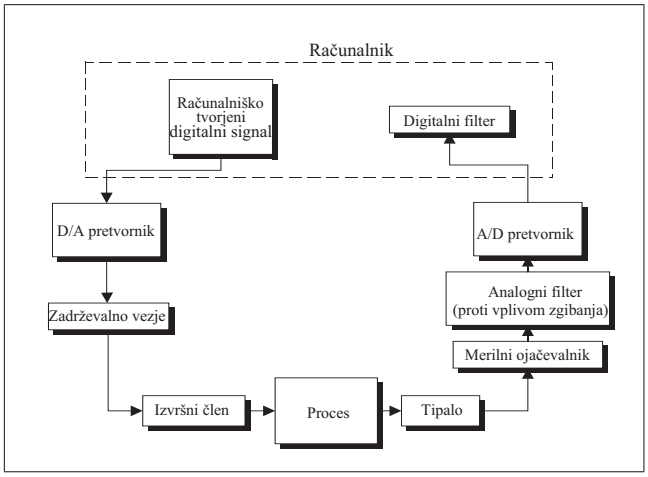
Dinamični proces, ki bi ga radi identificirali, je elektromehanska modelna naprava motor-generator proizvajalca ELWE [1]. To je naprava, namenjena laboratorijskemu eksperimentalnemu delu pri načrtovanju vodenja. Shemo naprave skupaj z opremo za zajemanje podatkov prikazuje slika 2.7.



Slika 2.7: Modelna naprava motor-generator

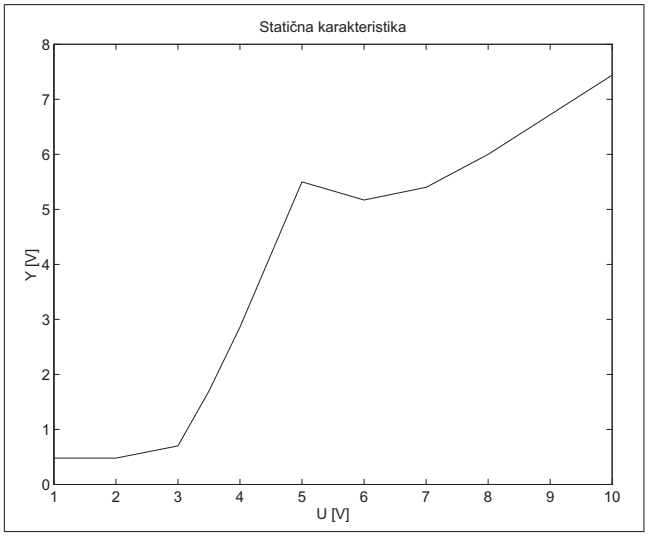
Enosmerni motor moči 100 W s serijskim vzbujanjem je mehansko povezan s tuje vzbujanim enosmernim generatorjem. Breme generatorja sta dve 40 W žarnici (220 V). Zanima nas model procesa, ki ima kot vhod napetost na sponkah motorja in kot izhod število vrtljajev oziroma s stališča meritev model sistema, ki ima kot vhod napetost na sponkah motorja u in za izhod napetost iz merilnega pretvornika vrtljajev y. Da lahko sistem vodimo s procesnim računalnikom, imamo na vhodu sistema tiristorski pretvornik. Merilnik števila vrtljajev daje na izhodu 1 V napetosti pri 1000 vrtljajih na minuto, ki jo peljemo še prek merilnega ojačevalnika. Tako sta vhodni in izhodni signal v območju 0 do 10 V. Tiristorski pretvornik in ojačevalnik sta del spremljevalne opreme sistema motor-generator istega proizvajalca.

Del merilne opreme je tudi pretvornik PCI-20000 proizvajalca Burr-Brown, ki med drugim vsebuje digitalno analogne in analogno digitalne pretvornike, potrebne za zajemanje in dovajanje signalov pri identifikaciji. Bločno shemo splošnega merilnega sistema za identifikacijo prikazuje slika 2.8. Izmerjeno statično karakteristiko procesa



Slika 2.8: Shema merilnega sistema

prikazuje slika 2.9.



Slika 2.9: Izmerjena statična karakteristika

Medtem ko statični model opisuje le ojačenje med vhodno in izhodno spremenljivko v ustaljenem stanju, dinamični model opisuje poleg ojačenja v ustaljenem stanju tudi prehodne pojave izhodne spremenljivke, ki nastopijo pri spremembi vhodne spremenljivke. Zaradi poenostavitve bomo dinamični model procesa imenovali kar model procesa.

Vrtljaji sistema so odvisni od toka skozi žarnici, ki se znatno spremeni, ko žarnici zagorita; to se na statični karakteristiki vidi (slika 2.9) kot padec napetosti. Delovna točka, v kateri iščemo linearizirani model, je 8.5 V. Statično karakteristiko moramo izmeriti zato, da bi lahko na njeni podlagi proces statično linearizirali, kar pa, kot vidimo iz slike 2.9, v našem primeru ni potrebno, ker je statična karakteristika okoli delovne točke, ki nas zanima (8.5 V), dokaj linearna. Če bi želeli model, ki opisuje sistem v celotnem delovnem področju, bi morali iskati nelinearni model (v naravi v glavnem ni linearnih procesov) ali pa se zadovoljiti z linearnim slabe točnosti. Če bi želeli identificirati nestabilen sistem, bi se morali tega lotiti nekoliko drugače [10].

Čas vzorčenja, vhodni signal, predhodna obdelava

Izbira vhodnega signala

Različne identifikacijske metode zahtevajo različne vhodne signale. Skupno vsem metodam je, da morajo biti vhodni signali frekvenčno dovolj bogati; to se lahko izraža v velikem številu poskusov ali v razgibanosti posameznega vhodnega signala [10]. Dober model dobimo le v frekvenčnem področju, ki ga vzbujamo. Obstaja veliko primernih in manj primernih vhodnih signalov. Ni dovolj, da uporabimo le eno obliko signalov, marveč moramo vhodni signal sestaviti iz za vsako frekvenčno območje ugodnih signalov. Podrobnejši napotki so v [10]. Naj omenimo še to, da je izbira vhodnega signala izredno pomembna pri postopku identifikacije. Po navadi se posname odziv procesa na več vhodnih signalih in se potem v nadaljnjem postopku odločimo za najbolj primerno kombinacijo vhodnega signala in odziva.

V našem primeru smo za identifikacijo izbrali psevdonaključni binarni signal (PNBS) kot vhodni signal. Ta signal je približek belega šuma. Beli šum [10] je signal s statistično neodvisnimi vrednostmi, ki ima spektralno močnostno gostoto enako konstanti. Zato je zelo primeren za vzbujanje procesov (proces je vzbujan v celotnem frekvenčnem področju), vendar praktično neizvedljiv, saj je njegova srednja moč neskončna [10].

PNBS je enostaven za tvorjenje in ima pri omejeni amplitudi zelo veliko močnostno gostoto. PNBS ima vrednost amplitude $+a$ in $-a$. Menjave predznaka nastopajo le v časovno diskretnih trenutkih $k\lambda, k = 1, 2, \ldots$, kjer je λ dolžina časovnega intervala, ki ga imenujemo tudi takt. Položaj ničel spektra je odvisen od takta λ. λ bomo torej določili glede na zanimivo področje za identifikacijo, to se pravi glede na ocenjeno dinamiko procesa. Zanimivo področje ugotavljamo iterativno toliko časa, dokler se nismo prepričali, da vhodni signal vzbuja celotno za nas zanimivo frekvenčno področje.

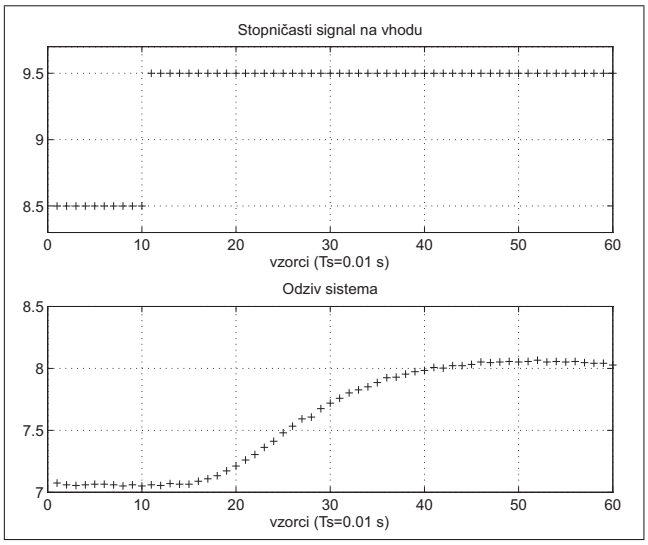
Določanje amplitude a vhodnega signala naj bo kompromis med čim večjo vrednostjo zaradi razmerja signal/šum na eni strani ter varnostjo in ekonomičnostjo delovanja in nelinearnostjo sistema na drugi strani. V našem primeru je nelinearnost zelo vpliven dejavnik, kar vidimo iz statične karakteristike (slika 2.9). Izbrali smo amplitudo $a = \pm 1$ V okoli delovne točke (8.5 V), ker smo želeli ostati v kolikor toliko linearnem območju.

Kadar na izbiro vhodnega signala zaradi različnih dejavnikov (npr. nezmožnost spreminjanja) ne moremo vplivati, moramo vsaj ugotoviti, v katerem frekvenčnem področju ta vhodni signal vzbuja sistem in (posledično) v katerem področju bo naš model veljaven.

Za vrednotenje modela smo izbrali vlak pravokotnih impulzov širine 1 sekunda in amplitude ± 1, ki zadovoljivo vzbuja proces v nekoliko ožjem frekvenčnem področju.

Izbira časa vzorčenja

Temeljno pravilo pri izbiri časa vzorčenja [10] je, naj bo ta 10% časa izravnave procesa pri odzivu na stopnico. Odziv na stopnico amplitude 1 V v izbranem linearnem območju prikazuje slika 2.10. Iz slike določimo čas izravnave, ki je



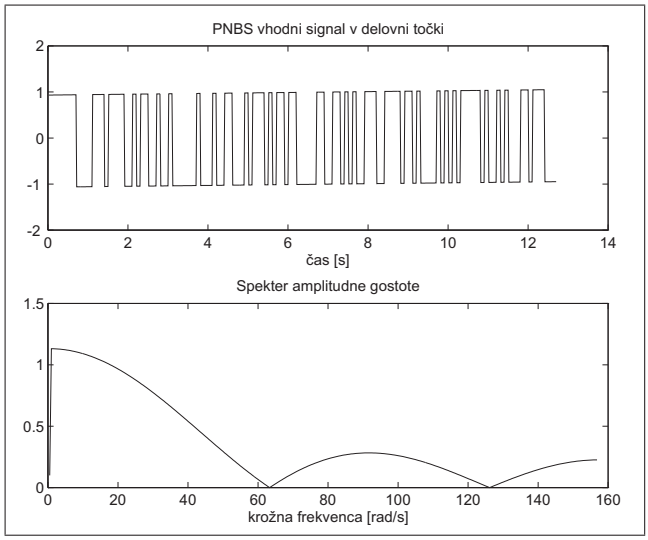
Slika 2.10: Odziv na stopničasto spremembo

v našem primeru ≈ 0.2 s, in od tod izberemo čas vzorčenja $T_s = 0.02$ s. Čas vzorčenja določa frekvenčno področje, v katerem lahko opazujemo signal (Shannonov teorem):

$$\begin{aligned} f_{max} &= \frac{1}{2T_s} = 25Hz \\ \omega_{max} &= 2\pi f_{max} = 157 rad/s. \end{aligned} \tag{2.13}$$

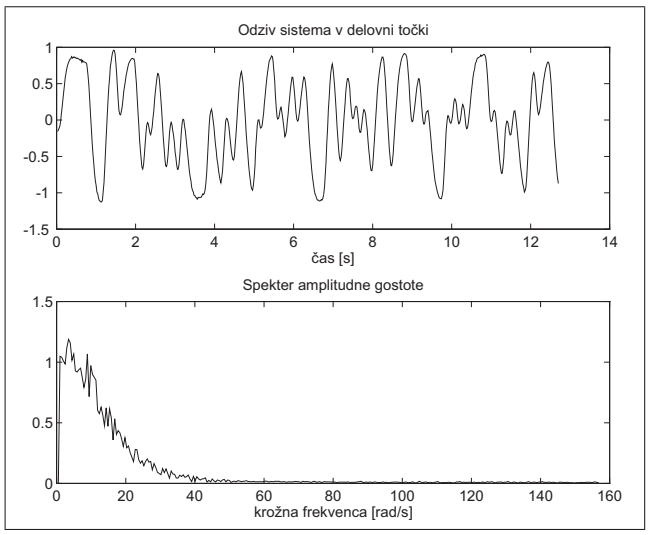
Vhodni signal PNBS smo tvorili z digitalnim registrom sedme stopnje, in ker smo se odločili za takt $\lambda = 5T_s = 0.1$ s, dobimo 635 vzorcev. Celotna meritev je bila torej dolga 12.7 s. Prva ničla močnostnega spektra vhodnega signala je bila pri $\omega = 62.8$ rad/s. Najbolj uporabno področje je do tiste frekvence, ko močnostni spekter pade na polovico, kar je v našem primeru pri 31.4 rad/s, saj le na tem frekvenčnem področju zadovoljivo vzbuja proces.

Vhodni signal in njegov spekter amplitudne gostote (amplitudni spekter aperiodičnega signala) prikazuje slika 2.11. Skok spektra amplitudne gostote vhodnega signala pri vrednosti 0 rad/s na sliki 2.11 nastopi zato, ker smo njegovo enosmerno komponento ($\omega = 0$) izločili. O tem bomo govorili malo pozneje. Nismo izbrali le enega vhodnega



Slika 2.11: Vhodni signal in njegova gostota amplitudnega spektra

signala, marveč je bil opisani signal izbran med desetimi po lastnostih podobnimi signali (PNBS signali različnih dolžin in taktov, kar pomeni različne spektre amplitudne gostote), za katere smo opravili meritve in šele na podlagi vrednotenja izbrali najprimernejši vhodni signal in ustrezno meritev. Število vzorcev naj bo čim večje, kajti čim več je vzorcev, tem boljša je identifikacija. Vhodnemu signalu, ki smo ga tvorili z računalnikom, dodamo prednapetost, tako da bo njegova srednja vrednost enaka delovni točki 8.5 V. Odziv sistema v delovni točki z izločeno enosmerno komponento (delovna točka) prikazuje slika 2.12.



Slika 2.12: Odziv in njegova gostota amplitudnega spektra

Pred analogno digitalnim pretvornikom lahko izvedemo analogno filtriranje (slika 2.8) in na ta način preprečimo zgibanje frekvenc [10]. V našem primeru je bil za to dober kar merilni ojačevalnik, ki ima frekvenčno karakteristiko

nizkopasovnega filtra. Celotnega vpliva zgibanja zaradi širokega pasovnega področja ojačevalnika nismo mogli izločiti. Toda tega vpliva ne moremo ločiti od motilnega šuma.

Pred postopkom identifikacije je treba posnete signale primerno obdelati. Odstraniti moramo enosmerno komponento, ki nastane zaradi delovanja v delovni točki (posledici sta skoka spektrov amplitudne gostote na slikah 2.11 in 2.12) in po potrebi zmanjšati vplive šuma predvsem za parametrično identifikacijo (o njej bomo govorili pozneje), da ne bi po nepotrebnem modelirali dinamike, ki nastopi zaradi motenj, oziroma da ne bi dobili pristranskih rezultatov. To lahko naredimo s filtriranjem, pri čemer moramo upoštevati izbrano metodo identifikacije. Zato se bomo k vprašanju filtriranja vrnili, ko bomo imeli metodo identifikacije izbrano.

Identificiramo lahko neparametrične modele, pri katerih ne potrebujemo vnaprejšnjih predpostavk o strukturi modela (razen linearnosti), ali parametrične modele, pri katerih moramo določiti strukturo vnaprej. Pri identifikaciji se ti modeli dopolnjujejo, in kot bomo videli v našem primeru, lahko na podlagi enega modela sklepamo na informacijo za identifikacijo drugega. Neparametrične modele običajno uporabimo za vrednotenje parametričnih.

Izbira strukture modela

Izbira strukture modela je pri identifikaciji parametričnega modela zelo pomembna, saj lahko z metodami za ocenjevanje parametrov določimo le vrednosti parametrov. Kakšen model procesa bomo izbrali, je odvisno od namena modela. V našem primeru želimo linearni parametrični model, ki bo čim bolj natančno opisoval sistem okoli delovne točke, vendar naj ne bi bil prevelikega reda. Če ima model velik red, sicer bolje opisuje dinamiko, vendar je tudi bolj kompleksen in ima pole in ničle, ki se med seboj približno krajšajo.

Izbirali smo med metodami, ki so opisane v [10] in smo jih že omenili. Pri identifikaciji linearnih sistemov se moramo odločiti le za red modela. Prvo oceno reda modela lahko dobimo z analizo informacijske matrike ali pa z identifikacijo neparametričnega modela - frekvenčnim odzivom.

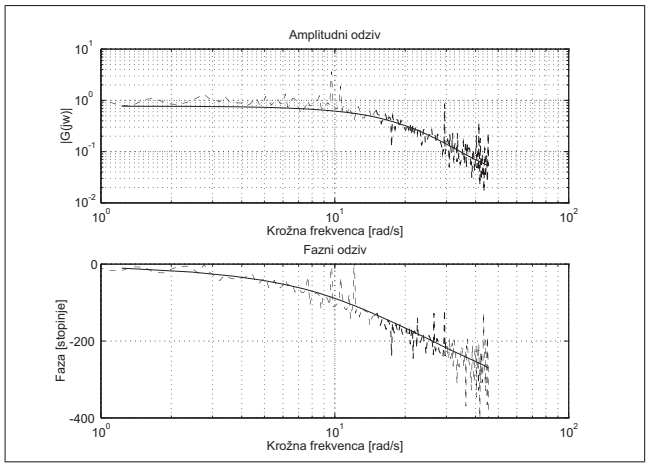
Z informacijsko matriko [10] si lahko pomagamo tako: Vrednost determinante informacijske matrike $\det(\frac{[\mathbf{\Psi}^T\mathbf{\Psi}]^{-1}}{N})$, kjer sta $\mathbf{\Psi}$ matrika podatkov (podrobneje opisana pozneje z enačbama (2.15) in (2.16)) in N dimenzija matrike podatkov, naredi preskok pri redu informacijske matrike, in to večjemu od reda modela, ki že zadosti dobro opisuje proces. Vrednosti determinante informacijske matrike so bile v našem primeru: za prvi red 0.3194, za drugi 0.0012, za tretji $4.6501 \cdot 10^{-7}$, za četrti red $1.4037 \cdot 10^{-10}$ in tako naprej. Največji padec vrednosti je med drugim in tretjim redom, torej je identificirani proces drugega reda.

Drugi način za določitev reda modela je neparametrična identifikacija. Glede na to, da smo si izbrali aperiodični vhodni signal, lahko določimo frekvenčni odziv z več metodami: s Fourierovo analizo, korelacijsko analizo ali spektralno analizo. Vse metode morajo načelno dati enake rezultate. Vse zahtevajo aperiodični vhodni signal, medtem ko korelacijska analiza zahteva kot vhod beli šum (PNBS je približek belega šuma).

Fourierova analiza je sicer enostavna metoda, vendar zelo občutljiva na šum. Fourierove transforme izračunamo z diskretno ali hitro Fourierovo transformacijo.

Frekvenčni odziv lahko izračunamo tudi s spektralno analizo. Tako kot daje Fourierova analiza v prisotnosti šuma zelo slabe rezultate, je tudi varianca ocene korelacijskih funkcij pri velikih premikih τ velika. Rešitev obeh problemov je glajenje npr. s Hammingovim oknom [10].

Glajenje je postopek filtriranja s spremenljivimi utežmi (frekvenčno ali časovno okno), s katerimi lahko izločimo šum. Zelo pazljivi moramo biti pri izbiri širine okna za glajenje, saj mora biti to ozko v primerjavi s časom opazovanja in dovolj široko, da ohrani bistveno informacijo. Podrobnosti o glajenju najdemo v [10]. Slika 2.13 prikazuje amplitudni in fazni frekvenčni odziv v zanimivem frekvenčnem področju brez glajenja, ki smo ga dobili s Fourierovo analizo (odziv bi bil enak, če bi vzeli spektralno analizo brez glajenja) in z glajenjem s časovnim Hammingovim oknom velikosti 35 vzorcev, ki smo ga dobili s spektralno analizo.



Slika 2.13: Amplitudni in fazni odziv brez glajenja (črtkana krivulja) in z glajenjem (polna krivulja)

Zavedati se moramo, da lahko z glajenjem zabrišemo konice v frekvenčnem odzivu. Za to, da dobimo prvo oceno reda sistema, bo dobljeni odziv zadovoljiv. Iz frekvenčnega odziva, prikazanega na sliki 2.13, smo na podlagi strmine padanja amplitudnega odziva in poteka faznega odziva (pri čemer lahko sklepamo tudi na mrtvi čas) potrdili odločitev, da proces lahko modeliramo kot sistem drugega reda.

Pri določanju reda modela procesa se moramo zavedati, da je to iterativni postopek in da končno odločitev lahko

sprejmemo šele v fazi vrednotenja ter da je do sedaj ugotovljena vrednost uporabna le kot začetna vrednost.

Določitev zakasnitve (mrtvega časa) je, razen kadar imamo opravka s sistemom prvega reda, lahko dokaj kompleksna. Inženirski pristop k temu problemu je, da ocenjujemo parametre pri različnih časih zakasnitve in vzamemo tisti rezultat, pri katerem se model najbolj ujema s podatki. V našem primeru smo po opisanem postopku prišli do zakasnitve 0.04 s. Dobljeno zakasnitev smo preverili pri faznem odzivu.

Izbor metode za ocenjevanje parametrov ni odvisen le od procesa, marveč tudi od oblike motenj oziroma šuma. Vedno moramo identificirati več modelov in se potem na podlagi preskusa veljavnosti (vrednotenja) modela odločiti za tistega, ki najbolj ustreza. Zelo težko določimo vnaprej, katero izmed metod za ocenjevanje parametrov - parametrično identifikacijo bomo uporabili. Po navadi preskusimo več metod in se tudi v tem primeru odločimo glede na vrednotenje modelov. Izbirali smo med metodami, ki so razložene v knjigah [10, 2].

Odločili smo se za najenostavnejšo, to je nerekurzivno metodo najmanjših kvadratov z modelom posplošenega pogreška [10]. Ta metoda daje, razen pri posebni obliki šumnega filtra, pristranske rezultate. Pristranskost lahko zmanjšamo z ustreznim filtriranjem, pri čemer iterativno poiščemo čim boljši filter. Takšni postopki so glede pristranskosti podobni drugim metodam ocenjevanja parametrov (metodi razširjenih najmanjših kvadratov, metodi največje podobnosti).

Metoda najmanjših kvadratov v bistvu pomeni reševanje predoločenega sistema enačb. V primeru ocenjevanja parametrov neznanega procesa, motenega s šumom, ocenimo parametre tako, da uporabimo matematični model procesa in minimiziramo kvadrat razlike med izhodom procesa in modela.

Enačba motenega procesa v prostoru z se glasi

$$y(z) = \frac{B(z^{-1})}{A(z^{-1})} z^{-d} u(z) + n(z), \tag{2.14}$$

kjer so: $A(z^{-1})$ - imenovalec prenosne funkcije v prostoru z, $B(z^{-1})$ - števec prenosne funkcije v prostoru z, y - izhodni signal, u - vhodni signal in n - motnja,

oziroma v časovnem prostoru

$$\begin{aligned} y(k) &= a_1 y(k-1) - a_2 y(k-2) - \ldots - a_n y(k-n) \\ &+ b_1 u(k-d-1) + \ldots + b_n u(k-d-n) \\ &+ a_1 n(k-1) - a_2 n(k-2) - \ldots - a_n n(k-n). \end{aligned} \tag{2.15}$$

Vektorski zapis te enačbe je

$$y(k) = \boldsymbol{\psi}^T(k)\boldsymbol{\theta}, \tag{2.16}$$

kjer je

$$\boldsymbol{\theta} = [a_1, \ldots, a_n, b_1, \ldots, b_n] \tag{2.17}$$

vektor parametrov. Oceno parametrov izračunamo v smislu najmanjših kvadratov (izpeljava in pogoji so podrobno opisani v [10]) po enačbi

$$\hat{\boldsymbol{\theta}} = [\boldsymbol{\Psi}^T \boldsymbol{\Psi}]^{-1} \boldsymbol{\Psi}^T \mathbf{y}, \tag{2.18}$$

kjer je $\mathbf{y}$ vektor izhodnega signala in je matrika $\boldsymbol{\Psi}$ sestavljena iz vektorjev $\boldsymbol{\psi}$ za ustrezne vrednosti elementov vektorja $\mathbf{y}$.

Filtriranje je pripomoček, s katerim lahko dosežemo, da so rezultati čim bolj nepristranski. Proces, ki ga opisuje enačba (2.14), lahko zapišemo tudi

$$A(z^{-1})y(z) = B(z^{-1})z^{-d}u(z) + A(z^{-1})n(z). \tag{2.19}$$

Metoda najmanjših kvadratov daje nepristranske rezultate, če je $A(z^{-1})n(z) = v(z)$ beli šum, to pomeni, če dobimo $n(z)$ iz $v(z)$ prek filtra $\frac{1}{A(z^{-1})}$. To seveda v praksi in tudi v našem primeru ni res, zato je potrebno filtriranje. Če se odločimo za filtriranje, moramo filtrirati z istim filtrom vhodni in izhodni signal.

Enačbo (2.19) delimo s filtrom $F(z^{-1})$ in dobimo

$$A(z^{-1})\frac{y(z)}{F(z^{-1})} = B(z^{-1})z^{-d}\frac{u(z)}{F(z^{-1})} + \frac{A(z^{-1})}{F(z^{-1})}n(z). \tag{2.20}$$

Nepristranske rezultate identifikacije dobimo, če je napaka enačbe (2.20) oziroma njen zadnji člen beli šum.

Vzemimo, da je $n(z)$ beli šum. V tem primeru dobimo nepristranske rezultate, če velja $A(z^{-1}) = F(z^{-1})$, to pomeni, če filtriramo z neznanim imenovalcem prenosne funkcije neznanega procesa, ki ga identificiramo.

V praksi seveda $n(z)$ ni beli šum (je pa lahko bliže belemu šumu kot pasovno omejenemu z $\frac{1}{A(z^{-1})}$). Zato uporabimo obliko filtra (npr. Butterworthov filter), ki mora biti takšen, da bo zadnji člen enačbe (2.20) čim bolj bel, kar pa preverjamo iterativno z vrednotenjem rezultatov.

Po omenjenem postopku smo identificirali modele različnih redov in se po vrednotenju, katerega elemente opisuje naslednje poglavje, odločili za model drugega reda, pri čemer smo imeli na voljo dve možnosti, ki ju bomo še naprej vrednotili, da bi izbrali tisto, ki bo primernejša za naš namen.

Pri izbranem drugem redu dobimo, če ne filtriramo vhodnega in izhodnega signala, prenosno funkcijo modela procesa

$$\hat{G}_{pn}(z) = \frac{0.0281z^{-1} + 0.0165z^{-2}}{1 - 1.6379z^{-1} + 0.6890z^{-2}} z^{-2}. \tag{2.21}$$

To je prenosna funkcija drugega reda v z prostoru s stabilnimi poli in fazno neminimalno ničlo ter dodatno zakasnitvijo (dva časovna vzorca, kar je 0.04 sekunde).

Po iterativnem postopku določevanja filtra, s katerim naj bi čim bolj izpolnili pogoj za nepristranskost, smo se odločili za Butterworthov filter drugega reda in lomno frekvenco 0.061 rad/s in se po vrednotenju odločili za model

drugega reda, ki ga opisuje prenosna funkcija

$$\hat{G}_{pf}(z) = \frac{0.0590z^{-1} - 0.0272z^{-2}}{1 - 1.7558z^{-1} + 0.7923z^{-2}} z^{-2}. \qquad (2.22)$$

Pri odločitvi, katerega izmed obeh modelov bomo izbrali, si pomagamo z vrednotenjem. Glavne elemente vrednotenja opisuje naslednji razdelek.

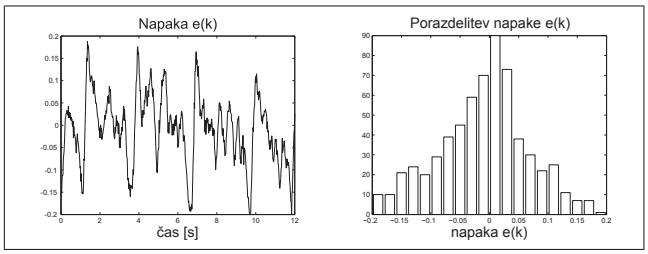
Preskus veljavnosti modela

Tisti del postopka identifikacije, ki pove, kako dober je model in kaj moramo spremeniti (red, metoda, vhodni signal, čas vzorčenja), je najpomembnejši del identifikacije. To je preskus veljavnosti modela ali tudi vrednotenje modela, ki se izvaja **iterativno** in pri katerem upoštevamo svoja opazovanja in zaključke in se ne zanašamo le na delne rezultate, ki jih daje računalnik. Kateri model sistema bomo izbrali, ni odvisno le od izpolnjevanja zahtev, ampak tudi od namena modela. Za preskus veljavnosti ločimo več postopkov. Postopke bomo prikazali z rezultati vrednotenja izbranih modelov iz enačb (2.21) in (2.22).

Preskus pogreška modela

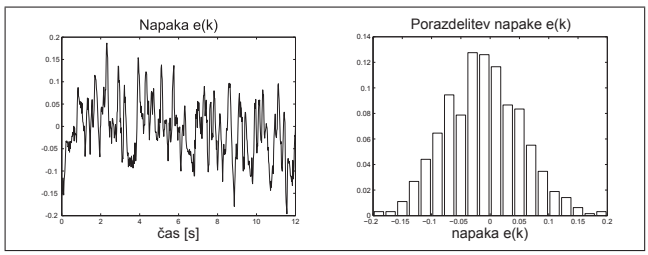
Pri metodi najmanjših kvadratov naj bi bila napaka modela enaka belemu šumu [10]. Zato lahko uporabljamo naslednje preskuse:

- Srednja vrednost signala napake $e(t)$ mora biti nič. Iz levih delov slik 2.14 in 2.15 vidimo, da napaka ustreza temu preskusu, vendar bolj pri modelu, dobljenemu iz filtriranih signalov.

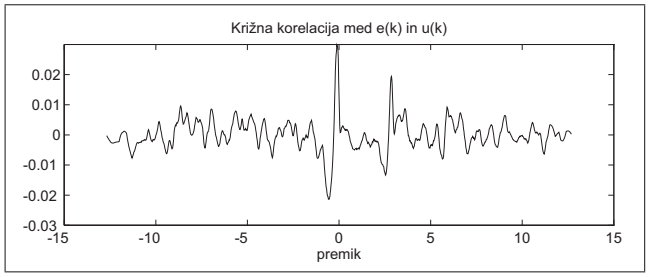


Slika 2.14: Napaka $e(k)$ med odzivoma prenosne funkcije $\hat{G}_{pn}$ in procesa, uporabljenega za identifikacijo (leva slika), ter porazdelitev signala napake $e(k)$ (desna slika)

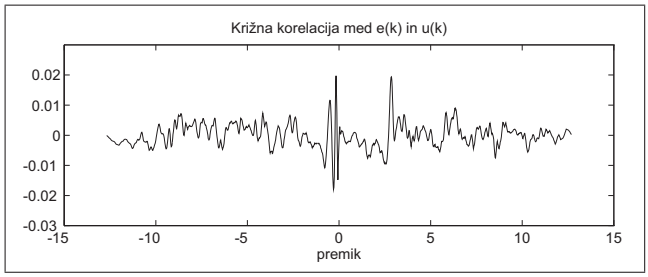
- Porazdelitev signala napake $e(k)$ mora biti simetrična, kar lahko potrdimo na podlagi desnih delov slik 2.11 in 2.14.
- Signal napake $e(k)$ mora biti neodvisen od preteklih vhodov ($\phi_{eu}(\tau) = 0$) za $\tau < 0$. Pri obratovanju v odprti zanki mora biti napaka neodvisna od vseh vhodov. Sliki 2.16 in 2.17 prikazujeta križno korelacijo med napako in vhodnim signalom in vidimo, da je njuna vrednost povsod majhna.



Slika 2.15: Napaka $e(k)$ med odzivoma prenosne funkcije $\hat{G}_{pf}$ in procesa, uporabljenega za identifikacijo (leva slika), ter porazdelitev signala napake $e(k)$ (desna slika)



Slika 2.16: Križna korelacija med napako $e(k)$ in vhodnim signalom $u(k)$ za prenosno funkcijo $\hat{G}_{pn}$



Slika 2.17: Križna korelacija med napako $e(k)$ in vhodnim signalom $u(k)$ za prenosno funkcijo $\hat{G}_{pf}$

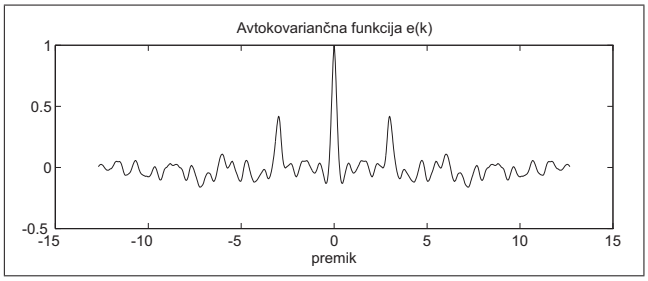
- Avtokovariančna funkcija signala napake $e(k)$ mora biti delta impulz. Iz slik 2.18 in 2.19 vidimo, da ima avtokovariančna funkcija največjo vrednost prav pri premiku τ=0, pri preostalih premikih pa je majhna. Od tod lahko sklepamo na podobnost (ne enakost) z delta impulzom.

Pri vseh omenjenih preskusih lahko na podlagi slik ugotovimo, da obe prenosni funkciji izpolnjujeta pogoje, vendar jih bolj izpolnjuje prenosna funkcija, dobljena z identifikacijo filtriranih signalov.

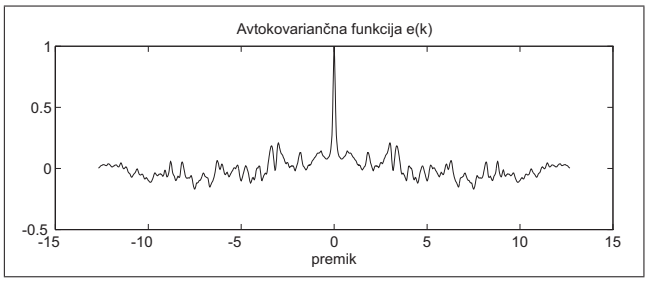
Konsistenca vhodno-izhodnega obnašanja

Model vrednotimo tako, da preskusimo, kako se odziva na vhodne signale, pri katerih poznamo odziv procesa. Pri tem imamo dve možnosti:

- vrednotenje opravimo na podatkih, ki smo jih uporabili za identifikacijo;



Slika 2.18: Avtokovariančna funkcija signala napake $e(k)$ za prenosno funkcijo $\hat{G}_{pn}$



Slika 2.19: Avtokovariančna funkcija signala napake $e(k)$ za prenosno funkcijo $\hat{G}_{pf}$

- vrednotenje opravimo na podatkih, ki jih nismo uporabili za identifikacijo, vendar v isti delovni točki. Taka vrsta vrednotenja pove veliko več o veljavnosti modela kot vrednotenje na podatkih, ki smo jih že uporabili za identifikacijo.

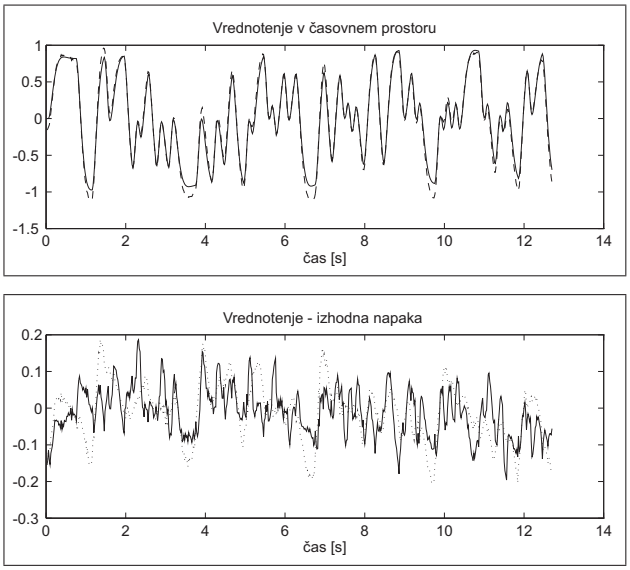
Sliki 2.20 in 2.21 prikazujeta rezultate vhodno-izhodnega obnašanja za obe vrsti podatkov za obe izbrani prenosni funkciji. Slika 2.20 prikazuje primerjavo časovnih odzivov obeh modelov na vhodni signal, uporabljen za identifikacijo in časovni potek napake med merjenim odzivom in odzivom modela $\hat{G}_{pn}$ ter med merjenim odzivom in odzivom modela $\hat{G}_{pf}$ za podatke, ki smo jih uporabili za identifikacijo. Slika 2.21 pa prikazuje primerjavo časovnih odzivov obeh modelov na vlak pravokotnih impulzov in časovni potek napake med merjenim odzivom in odzivom modela $\hat{G}_{pn}$ ter med merjenim odzivom in odzivom modela $\hat{G}_{pf}$ za podatke, ki jih nismo uporabili za identifikacijo. Iz slik lahko vidimo, da so napake pri modelu $\hat{G}_{pf}$ v povprečju manjše kot pri modelu $\hat{G}_{pn}$.

Tudi za podatke, ki niso bili uporabljeni za identifikacijo, morajo veljati vsi preskusi iz prejšnjega podpoglavja, v kar bi se lahko prepričali, če bi izrisali ustrezne diagrame.

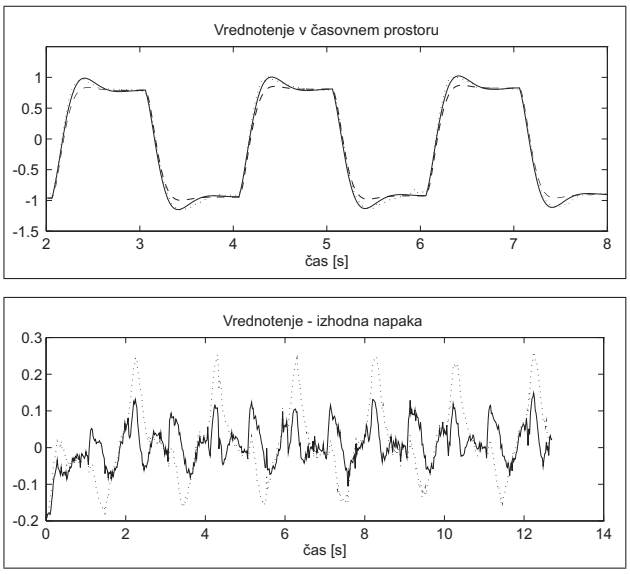
Iz odzivov na slikah 2.20 in 2.21 lahko vidimo, da prenosna funkcija, dobljena iz filtriranih signalov, bolje opisuje obnašanje procesa.

Konsistenca frekvenčnega obnašanja

Ujemanje med frekvenčnim odzivom, ki smo ga dobili s



Slika 2.20: Signal za identifikacijo



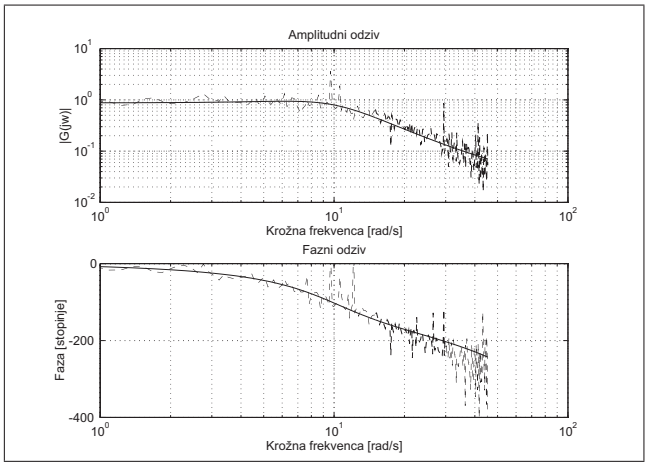
Slika 2.21: Signal za vrednotenje

Fourierovo analizo, in frekvenčnim odzivom prenosne funkcije $\hat{G}_{pf}$ prikazuje slika 2.22. Ujemanju časovnih odzivov mora slediti tudi ujemanje frekvenčnega odziva in narobe.

Na sliki 2.22 lahko vidimo, da je fazni kot z naraščajočo frekvenco absolutno gledano vedno večji. To je posledica mrtvega časa, kar se ujema tudi pri modelu, ki ima zakasnitev 0.04 sekunde.

Smiselnost parametrov, redukcija modela, kovariančna matrika, napake parametrov, uporabnost modela

Smiselnost diskretnega modela zveznega procesa lahko med drugim preskusimo s tem, da pogledamo število polov na



Slika 2.22: Preskus frekvenčnega obnašanja prenosne funkcije $\hat{G}_{pf}$ (odziv $\hat{G}_{pf}$ - polna krivulja, primerjalni frekvenčni odziv - črtkana krivulja)

negativni realni osi, ki ne sme biti liho. V našem primeru nimamo nobenega na negativni realni osi. Prav tako se mora model s procesom ujemati po stabilnosti.

Standardno deviacijo dobimo s korenjenjem diagonalnih elementov kovariančne matrike napak parametrov [10]. Tako dobimo kvantitativno mero zaupanja ocen parametrov. Večja je kovarianca, večje so deviacije, manjša je zanesljivost in veljavnost modela.

Kovariančno matriko izračunamo z enačbo

$$\mathrm{cov}[\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}] = \sigma_e^2 E\{[\boldsymbol{\Psi}^T \boldsymbol{\Psi}]^{-1}\}, \tag{2.23}$$

kjer je σ_e^2 varianca napake modela [10]. Za prenosno funkcijo $\hat{G}_{pn}$ so deviacije posameznih parametrov naslednje:

$$\begin{aligned} a1 &= -1.6379; \sigma = 0.0117; \\ a2 &= 0.6890; \sigma = 0.0112; \\ b1 &= 0.0281; \sigma = 0.0013; \\ b2 &= 0.0165; \sigma = 0.0017; \end{aligned}$$

za prenosno funkcijo $\hat{G}_{pf}$ pa dobimo deviacije:

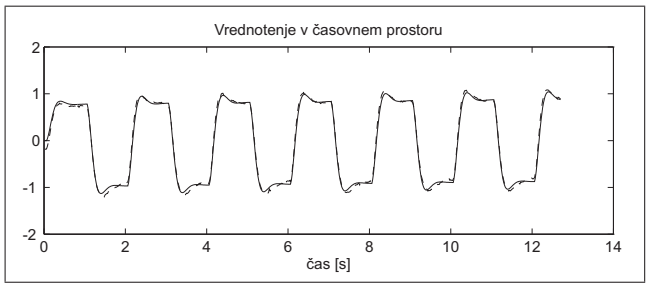
$$\begin{aligned} a1 &= -1.7558; \sigma = 0.0040; \\ a2 &= 0.7923; \sigma = 0.0035; \\ b1 &= 0.0590; \sigma = 0.0014; \\ b2 &= -0.0272; \sigma = 0.0019. \end{aligned}$$

Lahko opazimo, da so deviacije parametrov modela drugega reda, dobljenega iz filtriranih signalov, v splošnem manjše kot v primeru modela, dobljenega iz nefiltriranih signalov. Iz vseh prikazanih rezultatov vrednotenja lahko ugotovimo, da model, dobljen iz filtriranih signalov, natančneje opisuje obnašanje procesa, ki smo ga identificirali. Še več podrobnosti o vrednotenju za ta primer je v delu [4].

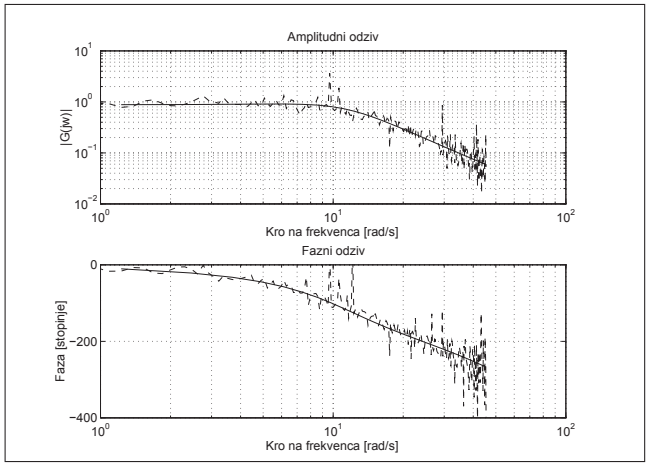
Vemo [9], da ne moremo sklepati na lego polov v zveznem prostoru iz lege polov v diskretnem prostoru. Običajno se ničle v neskončnosti v ravnini s preslikajo v pole na negativni realni osi oziroma v bližino koordinatnega izhodišča v ravnini z. Iz vsega tega bi lahko sklepali, da zvezni proces, ki ga identificiramo, pri predpostavki, da je drugega reda, nima ničel v končnosti. To potrjuje tudi slika 2.22, pri kateri lahko vidimo, da ima amplitudni odziv pri visokih frekvencah naklon 40 dB/dekado. Identificirani model drugega reda ima ničlo v končnosti ($s = -37.2989$), zato se njegov odziv pri visokih frekvencah najbolje ne ujema s frekvenčnim odzivom procesa. Če ponovimo postopek identifikacije z diskretnim modelom, ki ima ničlo pri z=0, dobimo naslednji model

$$\hat{G}_p(z) = \frac{0.0390 z^{-2}}{1 - 1.7144 z^{-1} + 0.7589 z^{-2}} z^{-2}. \tag{2.24}$$

Tako dobljeni frekvenčni odziv (slika 2.23) se zelo dobro ujema. Ujemanje časovnega odziva (slika 2.24) ni tako dobro, vendar je dobljen rezultat za večino praktičnih izvedb vodenja še sprejemljiv.



Slika 2.23: Del časovnega poteka odziva procesa (črtkana krivulja) in modela $\hat{G}_p$ (polna krivulja) na vlak pravokotnih impulzov



Slika 2.24: Preskus frekvenčnega obnašanja prenosne funkcije $\hat{G}_p$ (odziv $\hat{G}_p$ - polna krivulja, primerjalni frekvenčni odziv - črtkana krivulja)

Če bi želeli natančnejši odgovor o modelu, predvsem o njegovem obnašanju pri visokih frekvencah, bi morali izvesti dodatne poskuse oziroma ponoviti postopek identifikacije.

Zahtevana natančnost modela in uporabnost modela določene natančnosti je odvisna predvsem od namena njegove uporabe. Zato je ta ocena prepuščena uporabniku. Modele lahko uporabljamo za različne namene, kot so načrtovanje regulacijskih algoritmov, poenostavljanje kompleksnih sistemov, načrtovanje simulatorjev, načrtovanje algoritmov za odkrivanje napak itd. V našem primeru smo želeli prikazati postopek identifikacije s praktične strani in dobiti linearni model, ki bo opisoval dinamiko modelne naprave okoli delovne točke.

Identificirani model je glede na rezultate vrednotenja dovolj natančen za npr. načrtovanje regulacijskih algoritmov kot tudi za simulacijo dinamičnega obnašanja sistema motor-generator, pri čemer je treba upoštevati vrednost standardnih deviacij parametrov. Zadnjo potrditev modela bi dala njegova uporaba pri načrtovanju regulacijskega algoritma ali uporaba kot simulator.

Literatura

[1] ELWE (1988): ELWE Educational Systems, Experimental Manual U9, ELWE, Cremlingen.

[2] R. Isermann (1987): Identifikacija dinamičnih sistemov, Založba Fakultete za elektrotehniko, Ljubljana.

[3] R. Isermann (1971): Experimentelle Analyse der Dynamik von Regelsystemen, Identifikation I, Hochschultaschenbücher Verlag, Mannheim.

[4] J. Kocijan (1998): Praktikum identifikacij dinamičnih sistemov, ZAFER, Ljubljana, poglavje 7.1.

[5] J. Kocijan, D. Matko (1996): Primer identifikacije modela sistema v delovni točki, Elektrotehniški vestnik, Vol. 63, No. 3, 159-165.

[6] J. Kocijan, D. Matko (1996): Primer vrednotenja modela dinamičnega sistema, Elektrotehniški vestnik, Vol. 63, No. 4/5, 256-261.

[7] L. Ljung (1987): System Identification, Theory for the user, Prentice-Hall, Englewood Cliffs, NJ.

[8] L. Ljung (1992): System Identification Toolbox, User's Guide, The MathWorks, Inc., Natick, MA.

[9] D. Matko (1989): Diskretni regulacijski sistemi, Založba FER, Ljubljana.

[10] D. Matko (1998): Identifikacije, Založba Fakultete za elektrotehniko, Ljubljana, druga izdaja.

[11] J. Sjöberg et al. (1995): Non-linear Black-Box Modeling in System Identification: a Unified Overview, Automatica, Vol. 31, No. 12, 1691-1724.

3

Identifikacija nelinearnih dinamičnih sistemov

3.1 Splošno o identifikaciji nelinearnih sistemov

Kot smo povedal že v prvem poglavju, so raziskovalci v devetdesetih letih dvajsetega stoletja prišli do spoznanja, da je učenje nevronskih mrež, da bi se obnašale kot nelinearni dinamični sistemi, pravzaprav identifikacija nelinearnih dinamičnih sistemov. S tem spoznanjem je bilo mogoče vključiti in upoštevati dotedanja znanja s tega področja [2]. V pričujočem poglavju si bomo pogledali temeljne smernice za identifikacijo nelinearnih dinamičnih sistemov, kot jih podrobneje razlaga [4] in [5].

Poglavitni problem identifikacije sistemov je, kot smo omenili že v prejšnjem poglavju, kako najti primerno strukturo modela, s katero lahko zadosti dobro opišemo obnašanje nelinearnega sistema, ki ga identificiramo. Poiskati vrednosti parametrov za izbrano strukturo je v večini primerov manjši problem. Temeljno pravilo, ki se ga moramo vedno držati, je da ne identificiramo delov procesa, katerih model že poznamo. To pomeni, da moramo uporabiti vse znanje o procesu, ki ga imamo že od prej.

Tudi pri nelinearnih sistemih se uporablja barvna koda; s to označujemo ozadje znanja, ki ga uporabljamo za modeliranje: model bele škatle, kadar gre za teoretično modeliranje, model črne škatle, kadar gre za eksperimentalno modeliranje, in model sive škatle, kadar gre za različne kombinacije obeh načinov.

Eksperimentalni modeli, ki se uporabljajo za identifikacijo nelinearnih dinamičnih sistemov, so: umetne nevronske mreže, mehki modeli oziroma mreže lokalnih modelov, modeli na podlagi Gaussovih procesov, valčni modeli in mnogi drugi [3].

Če pristopamo k problemu identifikacije na ustaljen statistični način, je potrebno dodatno uskladiti tudi izrazoslovje. Nevronske mreže in druge podobne metode so med razvojem pridobile tudi izrazoslovje, ki je specifično za te in podobne metode. Naj naštejemo nekaj najpogosteje uporabljenih pojmov in različnih izrazov, ki se uporabljajo za njihov opis.

- Izrazi (izraz iz sistemske teorije = izraz na področju nevronskih mrež):
  - oceniti, identificirati = naučiti,
  - vrednotiti = posplošiti,
  - struktura modela = mreža,
  - podatki za identifikacijo = učna množica,
  - podatki za vrednotenje = testna množica,
  - preparametriziran model[1] = preveč naučen model.

Praktični vidiki identifikacije nelinearnih sistemov

Postopek identifikacije ne sme in ne more biti popolnoma avtomatiziran, saj je v njem preveč subjektivnih odločitev. Za uspešno delo poleg znanja o postopku identifikacije potrebujemo tudi primerno programsko opremo (npr. [7], [8]) in predvsem vhodno-izhodne podatke, ki nosijo dovolj informacije o dinamičnem obnašanju sistema, ki ga identificiramo.

Identifikacija nelinearnih dinamičnih sistemov je bistveno obširnejši problem, kot je identifikacija linearnih dinamičnih sistemov. Pojem nelinearnosti zajema neskončno veliko različnih oblik nelinearnosti in izkušnje za določeno vrsto nelinearnih dinamičnih sistemov v splošnem niso uporabne za druge vrste. Zato je priporočljivo, da si praktične izkušnje za modeliranje izbranega nelinearnega procesa naberemo na podobnih primerih, ki jih računalniško simuliramo. To je eden izmed načinov uporabe predhodnega znanja o procesu, ki ga identificiramo.

Problem identifikacije nelinearnih sistemov lahko opišemo na naslednji način: Posamezne vzorce izhodne spremenljivke iz sistema, ki ga želimo modelirati, lahko predstavimo kot neko funkcijo zakasnjenih vzorcev vhodnega in izhodnega signala, ki je motena s šumom:

$$y(k) = g(y(k-1), y(k-2), \ldots, u(k-1), u(k-2), \ldots) + n(k). \tag{3.1}$$

[1] Nevronske mreže so ponavadi vedno preparametriziran model, tudi če niso preveč naučene.

Identifikacija pomeni iskanje funkcije $g(\cdot)$

$$\hat{y}(k|\boldsymbol{\theta}) = g(\boldsymbol{\psi}(k), \boldsymbol{\theta}), \tag{3.2}$$

kjer sta:

$\boldsymbol{\psi}(k)$= $\boldsymbol{\psi}(u(k-1), u(k-2), \ldots, y(k-1), y(k-2), \ldots)$ vektor regresorjev,

$\boldsymbol{\theta}$ vektor parametrov.

Problem lahko razdelimo na dva podproblema:

- na izbiro vektorja regresorjev $\boldsymbol{\psi}(k)$ in
- na izbiro nelinearne preslikave $g(\boldsymbol{\psi})$.

Regresorji

Pri linearnih sistemih, pri katerih je struktura linearna, lahko sistem, ki je opisan v obliki

$$A(z^{-1})y(z) = \frac{B(z^{-1})}{F(z^{-1})}u(z) + \frac{C(z^{-1})}{D(z^{-1})}v(z), \tag{3.3}$$

zapišemo zelo enostavno kot

$$\hat{y}(k) = \boldsymbol{\psi}^T(k)\boldsymbol{\theta}. \tag{3.4}$$

Kakor smo opisali že v prejšnjem poglavju, se metode identifikacije linearnih sistemov razlikujejo po uporabljenih regresorjih. Podobno velja tudi za nelinearne sisteme in metode imenujemo po različnih uporabljenih regresorjih $\boldsymbol{\psi}$, s katerimi ponazorimo nelinearnost

$$\hat{y}(k) = g(\boldsymbol{\psi}(k), \boldsymbol{\theta}). \tag{3.5}$$

Regresorji $\boldsymbol{\psi}$ pri različnih nelinearnih modelih:

- metoda končnega impulznega odziva (angl. nonlinear finite impulse response - NFIR), kar pomeni, da so regresorji samo zakasnjeni vhodni signali ($u(k-i)$);
- metoda najmanjših kvadratov s posplošenim pogreškom (angl. nonlinear autoregressive model with exogenous input - NARX), kar pomeni, da so regresorji zakasnjeni vhodni in izhodni signali ($u(k-i), y(k-i)$);
- metoda z izhodnim pogreškom (angl. nonlinear output error method - NOE), kar pomeni, da so regresorji zakasnjene vrednosti vhodnega signala in ocen izhodnega signala (predikcij iz preteklosti)($u(k-i)$, $\hat{y}(k-i)$);
- metoda razširjenih najmanjših kvadratov (angl. nonlinear autoregressive and moving average model with exogenous input - NARMAX), kjer so regresorji zakasnjeni vhodni, izhodni in šumni signali ($u(k-i)$, $y(k-i), \varepsilon(k-i) = y(k-i) - \hat{y}(k-i)$);
- metoda Box - Jenkins (NBJ), ki ima za regresorje zakasnjene vhodne signale, zakasnjene ocene izhodnega signala, napako napovedi in napako simulirane napovedi (ko smo za napoved uporabljali ocene izhodnega signala)($u(k-i), \hat{y}(k-i), \varepsilon(k-i), \varepsilon_u(k-i) = y(k-i) - \hat{y}_u(k-i)$);
- možna je tudi predstavitev v prostoru stanj, kjer nelinearni sistem zapišemo v obliki : $\mathbf{x}(k) = \mathbf{F}(\mathbf{x}(k-1)\mathbf{u}(k-1))$,
- drugi možni regresorji.

Nelinearne preslikave

Nelinearne preslikave največkrat prikažemo kot uteženo vsoto baznih funkcij $g_k(\boldsymbol{\psi})$:

$$g(\boldsymbol{\psi}(k), \boldsymbol{\theta}) = \sum_k \alpha_k g_k(\boldsymbol{\psi}), \tag{3.6}$$

čeprav bomo kasneje pri modeliranju z Gaussovimi procesi videli, da so mogoče tudi druge oblike ponazoritev.

Znan skalarni primer ponazoritve z uteženo vsoto je na primer razvoj funkcije s Fourierovo vrsto. Tipični primeri nelinearnih preslikav, ponazorjenih z uteženo vsoto, so:

- valčne funkcije,
- B zlepki (angl. splines),
- UMETNE NEVRONSKE MREŽE,
  - večnivojski perceptron,
  - mreže radialnih baznih funkcij (angl. radial basis function network),
  - drugo,
- MEHKI MODELI.

Poglejmo si zapise nekaterih znanih baznih funkcij:

- nevronske mreže s sigmoidno aktivacijsko funkcijo

  $$g_k(\boldsymbol{\psi}) = \sigma(\beta_k \boldsymbol{\psi} + \gamma_k),$$

  kjer sta β širina in γ položaj;
- mreže z radialnimi baznimi funkcijami

  $$g_k(\boldsymbol{\psi}) = r(\beta_k(\boldsymbol{\psi} - \gamma_k));$$

- mehki modeli

$$g_k(\boldsymbol{\psi}) = \sum_j y_j(\prod \mu_A(\boldsymbol{\psi}));$$

- povratne mreže

$$\boldsymbol{\psi}(k) = g(\boldsymbol{\psi}(k-i), \boldsymbol{\theta})$$

- in mnoge druge.

Napotki za izbiro strukture

Najprej pričnemo s sistematično izbiro regresorjev od najbolj enostavne možne kombinacije do bolj kompleksnih:

- $u(k)$ - statična nelinearnost,
- $u(k-i)$ - NFIR,
- $u(k-i), y(k-i)$ - NARX,
- drugo.

Ko določamo red modela oziroma število zakasnitev vzorcev, je koristno, da se zavedamo Takensovega teorema (njegova oblika za vzbujane sisteme je v [10]), ki pove, kako iz vzorčenih vhodnih in izhodnih signalov rekonstruiramo model zveznega dinamičnega sistema. Zadostni pogoj za rekonstrukcijo sistema je:

$$n \geq 2p + 1, \tag{3.7}$$

kjer je n red diskretnega modela in p red originalnega zveznega sistema. V praksi se izkaže, da je marsikdaj dovolj model manjšega reda, kar je odvisno od sistema:

$$p \leq n \leq 2p + 1. \tag{3.8}$$

Ko smo red modela izbrali, se lotimo izbire baznih funkcij. Možnih je več načinov izbire in to objektivnih in subjektivnih. Konkretnih pravil ni, saj so vse bolj znane bazne funkcije, ki smo jih našteli, sposobne poljubno natančne ponazoritve nelinearnosti. Nekateri napotki, ki bi jih lahko upoštevali, so [27]:

- radialne bazne funkcije izberemo, kadar imamo opravka z majhnim številom regresorjev (npr. valčne funkcije za največ 3 regresorje),
- slemenaste (angl. ridge) bazne funkcije izberemo, kadar imamo opravka z večjim številom regresorjev (npr. sigmoidne nevronske mreže),
- mehke modele uporabimo, kadar obstaja vnaprejšnje hevristično znanje.

Ko izberemo bazne funkcije in s tem nelinearno strukturo modela, moramo določiti še njegove parametre, ki so uteži posameznih baznih funkcij in odvisno od izbrane strukture še nekateri drugi parametri. To lahko storimo s katero koli znano in ustrezno deterministično ali stohastično metodo optimizacije. Med determinističnimi metodami se dobro obnesejo Gauss-Newtonovi algoritmi, medtem ko so gradientne metode prvega reda v splošnem časovno potratne. Možna je nesprotna optimizacija parametrov ali pa sprotna tako imenovana rekurzivna identifikacija. Kriteriji optimizacije so po navadi odvisni od pogreška modela.

Pogrešek modela

Optimizacijski kriteriji so različni. Odvisni so od namena modela in od optimizacijske metode in narobe. Kriterijska funkcija, s katero je optimizacijski kriterij zapisan, je kvantitativno izraženo merilo kvalitete modela. Po navadi želimo doseči čim manjšo vrednost kriterijske funkcije, ki je največkrat funkcija pogreška modela. Pogosto uporabljen primer je kriterijska funkcija, odvisna od kvadrata pogreška modela:

$$\begin{aligned} \bar{V}(\boldsymbol{\theta}) &= \mathbb{E} \parallel y(k) - g(\boldsymbol{\psi}(k), \boldsymbol{\theta}) \parallel^2 \\ &= \sigma_n^2 + \mathbb{E} \parallel g_0(\boldsymbol{\psi}(k)) - g(\boldsymbol{\psi}(k), \boldsymbol{\theta}) \parallel^2, \end{aligned} \tag{3.9}$$

kjer je $g_0(\boldsymbol{\psi}(k))$ originalni sistem in σ_n^2 varianca šuma.

Kadar optimiziramo model glede na kvadrat pogreška, so v splošnem možni trije viri napak, ki kvarijo kvaliteto modela:

- šum,
- pristranskost, ki jo lahko prikažemo kot

$$V = \mathbb{E} \parallel g_0(\boldsymbol{\psi}(k)) - g(\boldsymbol{\psi}(k), \hat{\boldsymbol{\theta}}(m))) \parallel^2,$$

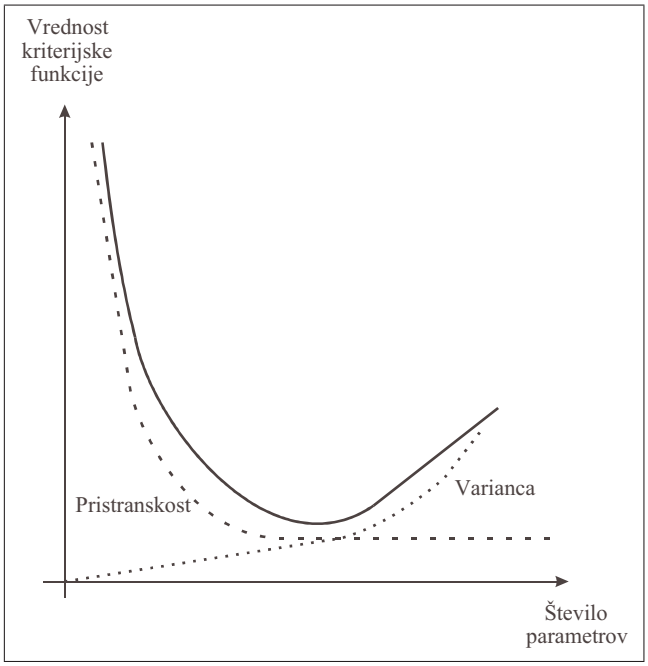
  kjer je $\hat{\boldsymbol{\theta}}(m)$ vektor ocenjenih vrednosti parametrov izbrane dolžine m,
- in varianca vrednosti parametrov, ki je odvisna od variance šuma meritev σ_n^2

$$V \approx \sigma_n^2 \frac{dim\,\{\boldsymbol{\theta}\}}{N}.$$

Ne moremo identificirati boljšega modela, kot to dopušča vsebina informacije v merjenih vhodno-izhodnih podatkih. Varianca vrednosti parametrov in pristranskost sta med seboj povezana, kot prikazuje slika 3.1.

Nasveti za izvedbo identifikacije

Glede na kompleksnost problema identifikacije nelinearnih dinamičnih sistemov je težko predpisati natančen postopek. V splošnem ima postopek enake korake kot za identifikacijo linearnih sistemov, ki smo ga navedli v prejšnjem poglavju. Lahko pa naštejemo nekaj nasvetov, ki bi pripomogli k učinkovitejšemu delu.



Slika 3.1: Prikaz poteka razmerja med pristranskostjo in varianco

1. Oglejte si vhodno-izhodne merjene podatke. Iz njih lahko sklepamo, ali je sistem zelo nelinearen in kakšne so njegove časovne konstante (kot se pogosto uporablja žargon iz identifikacije linearnih dinamičnih sistemov).

2. Najprej poskusite enostavne stvari, to je enostavne strukture in dimenzije modelov: najprej linearni modeli in majhno število regresorjev, baznih funkcij in parametrov za ocenjevanje.

3. Poiščite fizikalno ozadje dinamičnega procesa, saj lahko iz njega dobite idejo za izbiro regresorjev.

4. Podatke razdelite na tiste za identifikacijo in tiste za vrednotenje modela.

5. Podatke normalizirajte (različne vrste podatkov transformirajte v en velikostni razred) in usrediščite.

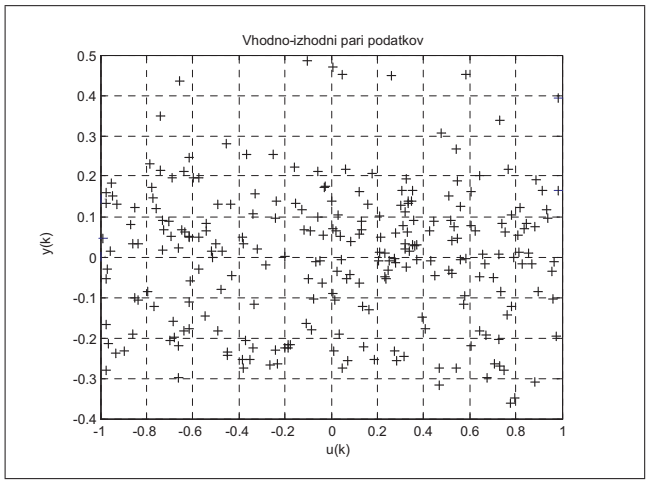
6. Ves čas optimizacije spremljajte razmerje pristranskost/varianca (angl. bias/variance), ki nakazuje nespametno večanje števila parametrov.

7. Kadar sistem modelirate z nevronsko mrežo uporabite regularizacijo. Podrobno je opisana v [5].

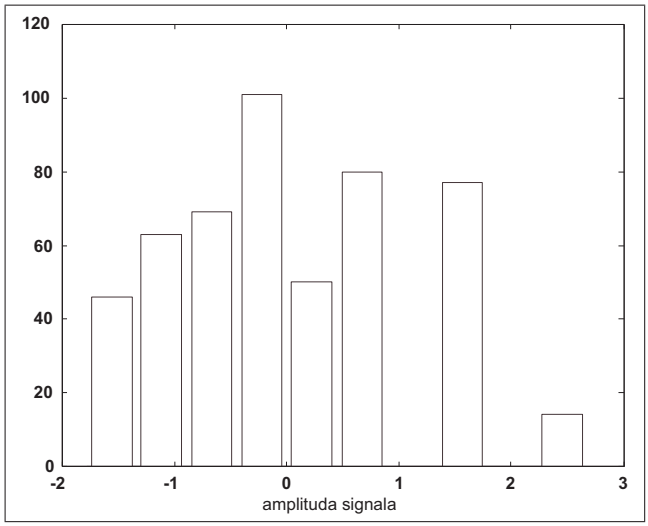
8. Raziščite značaj učinkovitosti števila parametrov, saj naj bi bil model čim bolj enostaven, toda ne preenostaven.

9. Ko izbirate čas vzorčenja, naj velja enako pravilo kot pri linearnih sistemih (čeprav ga je bistveno težje upoštevati): čas vzorčenja naj bo tak, da bo z njim zajeta vsa zanimiva dinamika procesa.

10. Če imate možnost izbire vhodnega signala, izberite takega, da bo imel vhodno-izhodno porazdelitev podatkov in amplitudno porazdelitev čim bolj bogato v delovnem področju, ki je omejeno s fizikalnimi omejitvami procesa. Primer porazdelitev prikazujeta sliki 3.2 in 3.3.



Slika 3.2: Primer vhodno-izhodne porazdelitve podatkov



Slika 3.3: Primer amplitudne porazdelitve vhodnega signala

Preskus veljavnosti modela

Verifikacija in vrednotenje modela je, kot smo že povedali, eden najpomembnejših korakov pri modeliranju. Nelinearnost dinamičnih sistemov postopkov vrednotenja prav nič ne olajša.

Kot smo videli in je opisano z enačbo (3.2), je modeliranje prirejeno v glavnem za enokoračno napoved. Model dinamičnega sistema je ustrezen skladno s svojim namenom v glavnem tedaj, kadar omogoča napovedovanje tudi

za daljši horizont in za simulacijo. Tipičen primer takega namena sta uporaba modela za načrtovanje vodenja ali za diagnosticiranje delovanja sistema. To pomeni, da je treba model po verifikaciji preizkusiti za enokoračno napoved, vendar tudi za simulacijsko delovanje.

- Enokoračna predikcija:

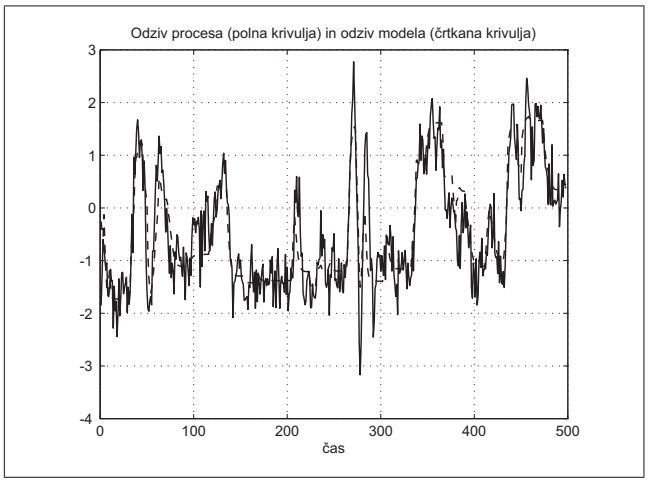
$$\hat{y}(k+1) = g(y(k), y(k-1), \ldots, u(k), u(k-1), \ldots).$$

- Simulacija:

$$\hat{y}(k+1) = g(\hat{y}(k), \hat{y}(k-1), \ldots, u(k), u(k-1), \ldots).$$

Kako simuliramo model prikazuje slika 1.22

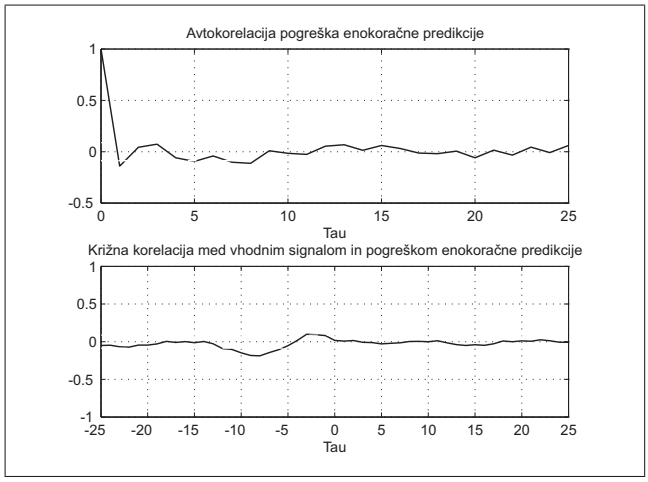
Analizo konsistence vhodno-izhodnega obnašanja z enokoračno predikcijo in simulacijo (slika 3.4) se praviloma izvede na podatkih, ki so bili uporabljeni za identifikacijo, in na drugih podatkih, namenjenih posebej za vrednotenje (testna množica).



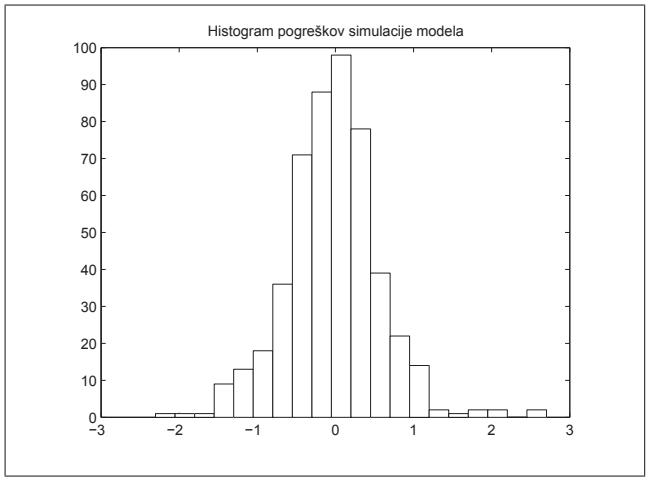
Slika 3.4: Preskus konsistence vhodno-izhodnega obnašanja

Poleg naštetih opisov vhodno-izhodnega obnašanja lahko uporabimo tudi za nelinearne sisteme, tako kot smo uporabili za linearne sisteme, različne oblike statističnih preizkusov. Med te sodijo razne korelacije, s katerimi preskušamo, ali je pogrešek napovedi neodvisen od vseh vhodnih in izhodnih signalov. Primer je preskus pogreška modela enokoračne predikcije z avto- in križnokorelacijo ter analiza histograma pogreška identifikacije in vrednotenja (primera na sliki 3.5 in 3.6).

Druge oblike statističnih preskusov so ocene povprečnega pogreška posploševanja (angl. estimating average generalisation error). Tukaj gre za ocenjevanje pogreška z raznimi kriterijskimi funkcijami, kot sta srednja kvadratična vrednost ali Akaikejeva kriterijska funkcija. Podrobneje o vrednotenjih s statističnini preskusi najdemo v [6].



Slika 3.5: Preskus pogreška modela

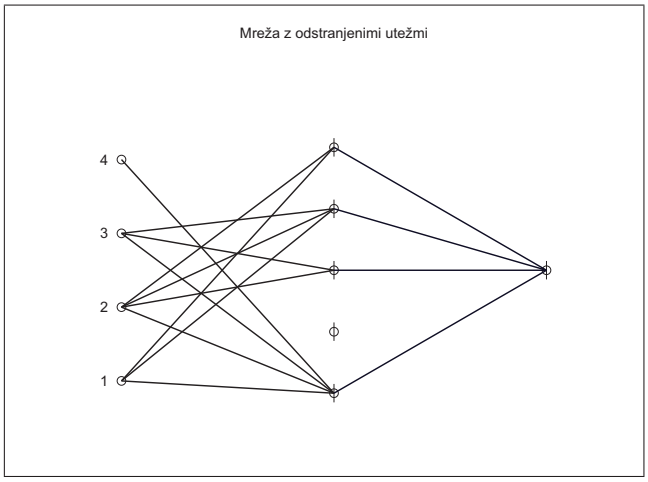


Slika 3.6: Histogram pogreškov enokoračne predikcije

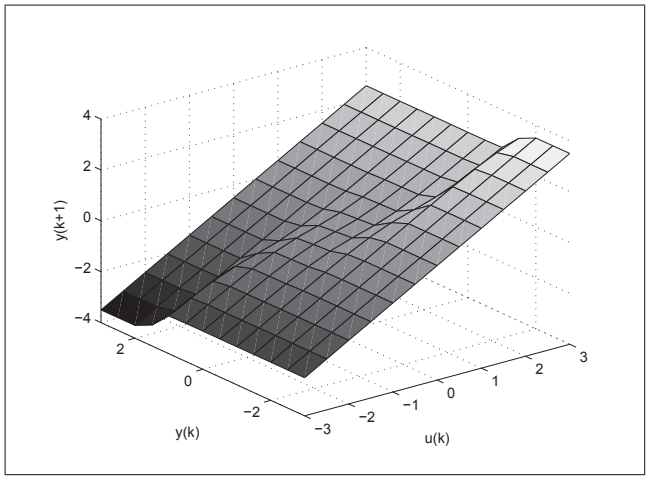
Del vrednotenja je tudi redukcija ali obrezovanje modela. Primer nevronske mreže po opravljenem obrezovanju prikazuje slika 3.7. Postopki za redukcijo so odvisni od uporabljene strukture in načina modeliranja. Več o teh postopkih je napisano v literaturi, ki opisuje posamezne postopke (npr. specifične nevronske mreže), in v razni programski opremi. S temi postopki se ne bomo ukvarjali podrobno; več o tem lahko preberemo npr. v delu [1].

Glavni preskus modela je ne glede na njegovo strukturo in način modeliranja vedno preskus modela glede na njegov namen. Model je dovolj dober takrat, ko je uporaben za namen, za katerega je bil izdelan.

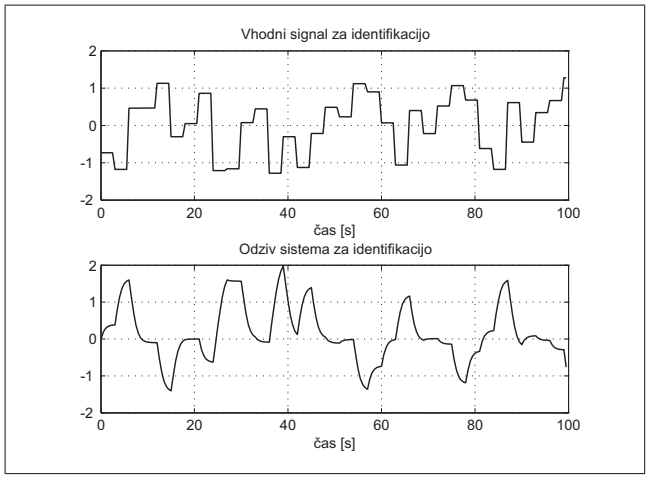
To je bil le pregled modeliranja. Za boljši vpogled je na voljo literatura, ki opisuje identifikacijo z nevronskimi mrežami podrobneje, npr. [5],[6].



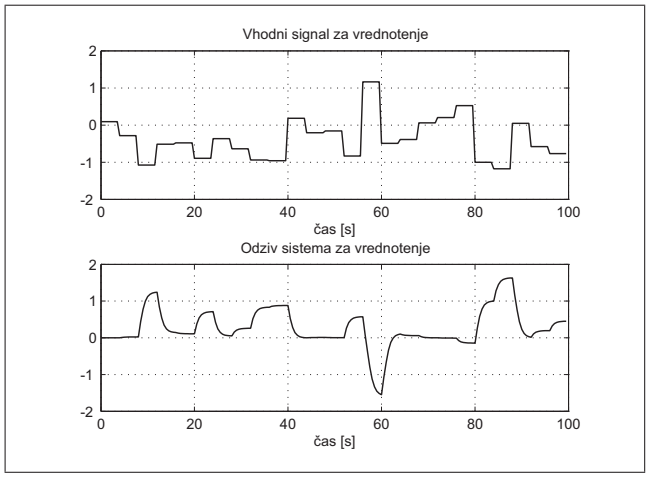
Slika 3.7: Primer reducirane (obrezane) strukture umetne nevronske mreže



Slika 3.8: Prikaz nelinearnosti sistema



Slika 3.9: Identifikacijski signal in odziv



Slika 3.10: Signal za vrednotenje in odziv

Prikaz identifikacije z metodo posplošenega pogreška (NARX)

Postopek identifikacije bomo prikazali na primeru identifikacije dinamičnega nelinearnega sistema prvega reda z umetno nevronsko mrežo po metodi posplošenega pogreška.

Matematični model procesa je opisan z naslednjo nelinearno diferenčno enačbo:

$$y(k) = y(k-1) - 0.5\tanh(y(k-1) + u^3(k-1)), \quad (3.10)$$

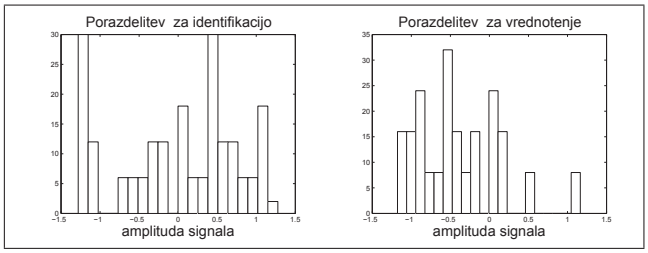
kjer sta u vhodni in y izhodni signal iz sistema. To je ena izmed diskretnih oblik zveznega sistema

$$\dot{y} = -\tanh(y + u^3) \quad (3.11)$$

pri času vzorčenja 0.5s.

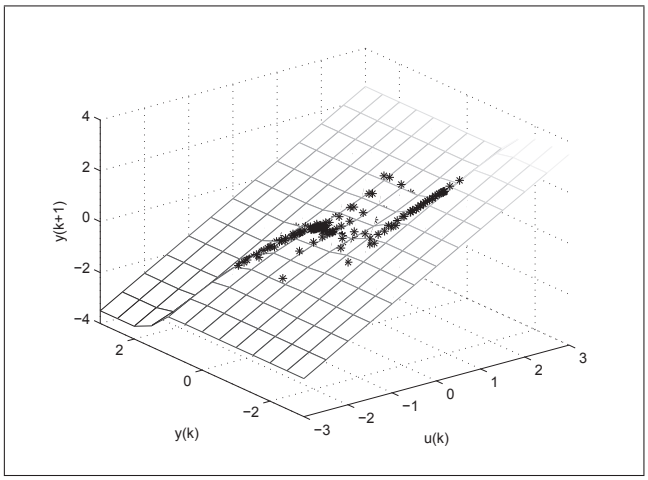
Ker gre za sistem prvega reda, lahko nelinearnost prikažemo v prostoru (to je prikazano na sliki 3.8). Sistem smo vzbujali z naključnim signalom spremenljive amplitude na področju med -1.2 in +1.2. Vzbujevalni signal in odziv procesa prikazuje slika 3.9. Signal za vrednotenje mora biti drugačen od tistega za identifikacijo, da bi se lahko prepričali o ustreznosti modela. Izbrali smo prav tako naključno spreminjajoči se signal na enakem amplitudnem področju, kot je bil identifikacijski signal. Vhodni signal za vrednotenje in odziv procesa sta prikazana na sliki 3.10. Čeprav v našem primeru ni bilo tako, pa je pogosto, da signal, ki ga imamo, razdelimo na dva dela, in sicer v del za identifikacijo (učna množica) in del za vrednotenje (testna množica). Dolžina signalov je lahko poljubna, vendar je smiselno, da je daljši identifikacijski signal, da bi bilo v njem vsebovano čim več informacije o dinamičnem obnašanju procesa.

Histograma amplitudnih porazdelitev izbranih vhodnih signalov sta prikazana na sliki 3.11. Lahko vidimo, da amplitudna porazdelitev ni popolna v celotnem izbranem področju, toda signal za identifikacijo ima boljšo amplitudno porazdelitev. Kako so porazdeljeni identifikacijski vhodno-izhodni vzorci po področju nelinearnosti procesa, prikazuje slika 3.12. Če bi želeli gostejšo porazdelitev vzorcev, bi potrebovali več vzorcev, to pomeni tudi daljši signal. Zelo pomembno je, da si ogledamo, kakšne vhodno-



Slika 3.11: Histograma amplitudnih porazdelitev vhodnih signalov za identifikacijo in vrednotenje

izhodne podatke imamo na voljo za identifikacijo in za vrednotenje, da lahko dovolj zgodaj prepoznamo, kaj lahko dosežemo z identifikacijo modela. Tam, kjer ni podatkov za identifikacijo, model ne more povzeti obnašanja procesa.

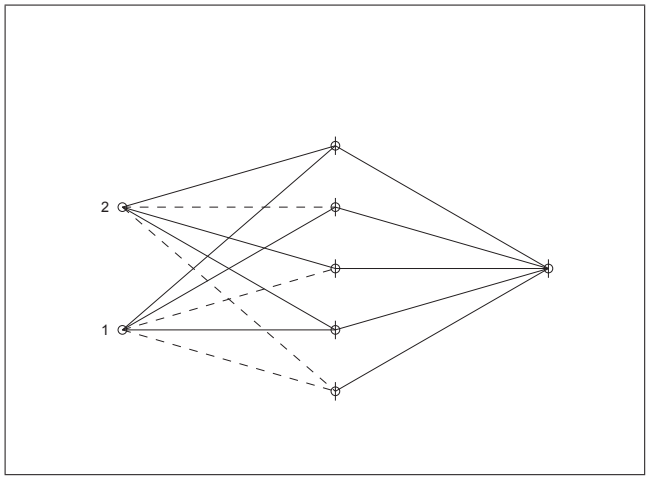


Slika 3.12: Vhodno-izhodna porazdelitev podatkov glede na nelinearnost sistema, ki ga identificiramo

V naslednjem koraku bomo izbrali strukturo nevronske mreže in regresorje ter določili parametre mreže.

Pri identifikaciji smo uporabljali programsko opremo NNSYSID Toolbox za Matlab [8]. Odločili smo se za nevronsko mrežo z večnivojskim perceptronom, vendar bi lahko izbrali tudi katero drugo nevronsko mrežo, ki je univerzalni aproksimator ali bi lahko ustrezno ponazorila ta sistem. Izbrana regresorja sta bila $y(k-1)$ in $u(k-1)$. Za strukturo smo izbrali strukturo NARX oziroma metodo s posplošenim pogreškom. Po iterativnem postopku identifikacije in vrednotenja smo izbrali nevronsko mrežo z enim skritim nivojem in petimi nevroni v njem. Shematsko je struktura mreže prikazana na sliki 3.13.

Parametre (uteži) mreže smo določili z Levenberg-Marquardtovo metodo, ki smo jo izbrali zaradi njene visoke učinkovitosti pri optimizaciji parametrov nelinearnih optimizacijskih problemov. Vrednosti parametrov, kot jih opisuje enačba (1.4), lahko zberemo v matriki/vektorju skritega nivoja $\mathbf{W}_1 = [w_{jk}|w_{0k}]$ in izhodnega nivoja $\mathbf{W}_2 = [w_{ij}|w_{0j}]$. Po končani optimizaciji, ko je bila napaka tako majhna, da se uteži niso več opazno spreminjale, smo dobili naslednje rezultate:



Slika 3.13: Struktura mreže

$$\mathbf{W}_1 = \begin{bmatrix} -0.5588 & -2.0621 & -1.9530 \\ 0.5155 & 0.0499 & -0.8670 \\ -1.5149 & 0.3190 & 0.4768 \\ 0.3366 & -1.2029 & 1.8379 \\ 0.8411 & 1.3841 & 1.7123 \end{bmatrix},$$

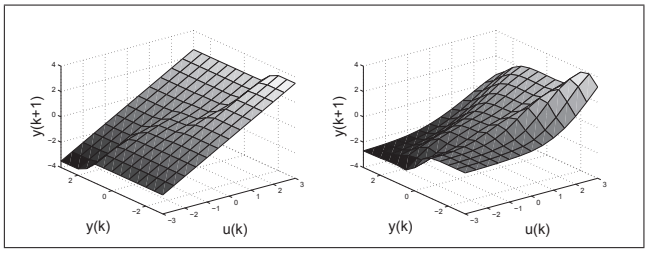
$$\mathbf{W}_2 = \begin{bmatrix} 1.2054 & 1.7784 & 0.0810 & 1.1704 & 1.4048 & -0.0580 \end{bmatrix}.$$

Matrika uteži W_2 ima en element več, kot je povezav med skritim in izhodnim nivojem, ker vsebuje na zadnjem mestu še vrednosti izhodnega praga oziroma predvrednosti (angl. offset) w_{0j}.

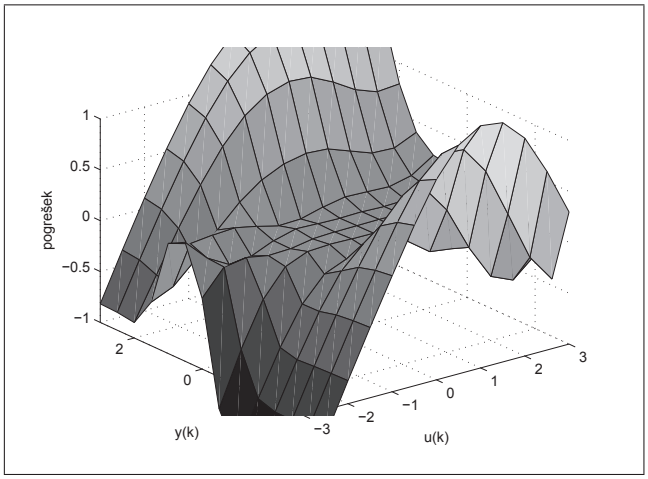
Oglejmo si vrednotenje modela, ki se v postopku identifikacije izvaja vsakič, ko dobimo nov model in želimo ovrednotiti njegovo ustreznost. Postopek identifikacije izvajamo toliko časa, dokler model ne ustreza svojemu končnemu namenu. Prikazali bomo samo rezultate za končni izbrani model. V našem primeru je bil namen modela ilustracija postopka identifikacije.

Na sliki 3.14 vidimo, kako je na podlagi identifikacijskih podatkov nevronska mreža ponazorila nelinearnost sistema. Napako med nelinearnostjo procesa in nelinearnostjo modela si lahko ogledamo na sliki 3.15.

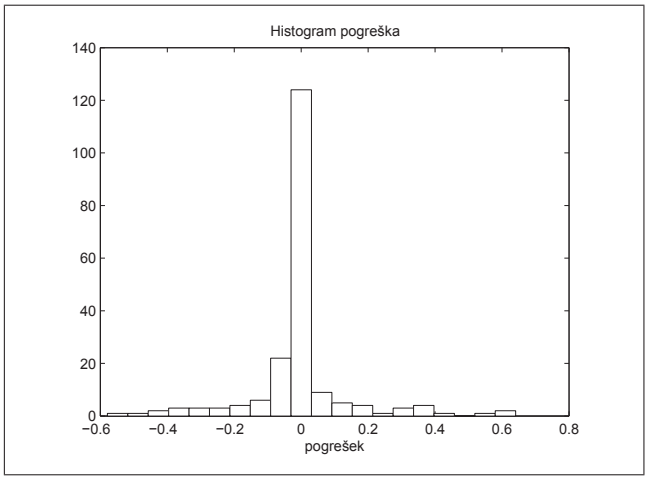
Če je struktura modela ustrezna, se bo načelno po optimizaciji odziv modela razlikoval od odziva procesa le za beli šum. Amplitudna porazdelitev pogreška simulacije pri uporabljenem signalu za vrednotenje, ki bi morala biti normalna, je prikazana na sliki 3.16. Vidimo, da prikazano porazdelitev lahko pojmujemo kot precej ozko Gaussovo krivuljo. Drugi statistični orodji, s katerima lahko potrdimo statistične lastnosti pogreška, je avtokorelacija pogreška (slika 3.17), ki prejšnjo ugotovitev potrjuje, in križna korelacija med pogreškom in vhodnim signalom (slika



Slika 3.14: Enokoračna predikcija: sistem (levo), model (desno)



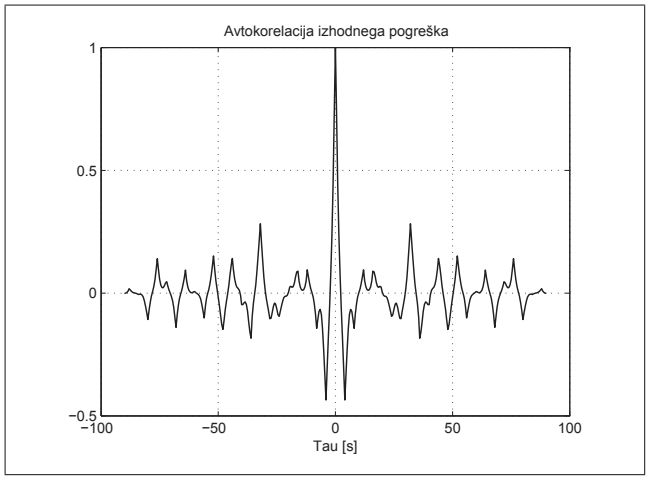
Slika 3.15: Vrednotenje pogreška odziva pri enokoračni predikciji



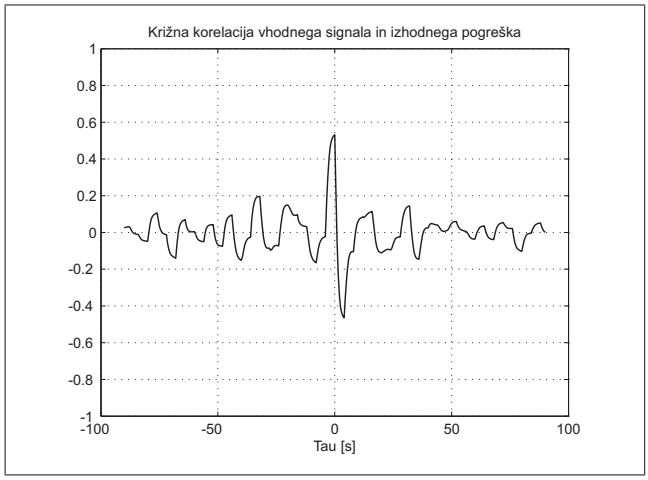
Slika 3.16: Porazdelitev pogreška odziva na signal za vrednotenje s simulacijo

3.18), ki govori o nizki korelaciji med pogreškom in vhodnim signalom.

Med bolj informativnimi je običajno kvalitativna ocena vhodno-izhodnega obnašanja, ki jo izvajalec identifikacije izvede s primerjavo odzivov procesa in modela na signal za vrednotenje. Iz slike 3.19, ki prikazuje tako primerjavo, lahko ugotovimo precejšnje ujemanje obeh odzivov.

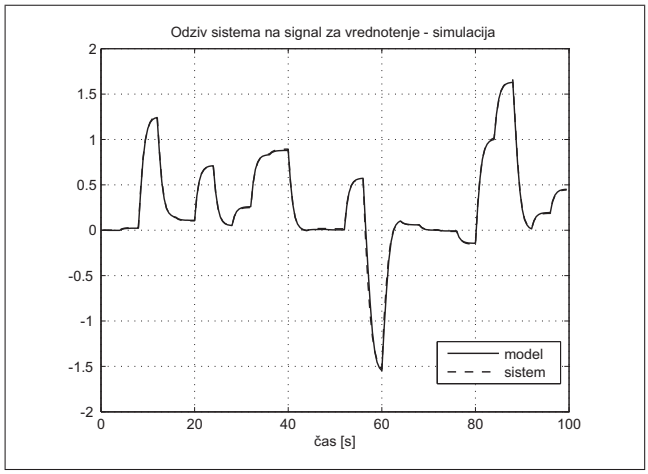


Slika 3.17: Avtokorelacija izhodnega pogreška



Slika 3.18: Križna korelacija pogreška z vhodnim signalom

Naš namen, ki je bil ilustracija postopka identifikacije ne-



Slika 3.19: Preskus konsistence vhodno-izhodnega obnašanja na signalu za vrednotenje

linearnega procesa, je bil zadovoljivo dosežen z dobljenim

modelom, zato smo zelo iterativen postopek zaključili.

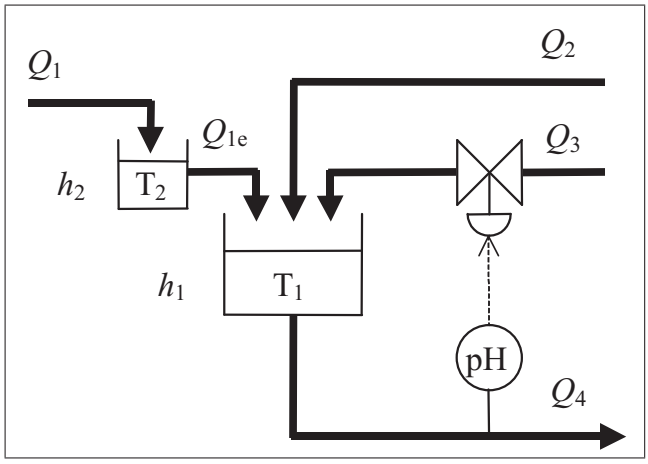
3.2 Primer identifikacije pH procesa

V ilustrativnem primeru v prejšnjem podpoglavju smo prikazali postopek identifikacije. Njegov prikaz je bil precej olajšan zato, ker smo identificirali sistem prvega reda, ki mu lahko nelinearnost prikažemo v prostoru. Za sisteme višjega reda je tako vrednotenje zelo oteženo. V primeru identifikacije pH procesa bomo prikazali identifikacijo sistema višjega reda na simuliranem procesu, ki odraža resnični kemijski proces.

Poenostavljeno shemo pH procesa, ki je v literaturi pogosto kot primerjalni primer za različne metode modeliranja in načrtovanja vodenja, vidimo na sliki 3.20. Namen tega procesa je kemijska nevtralizacija vhodne tekočine. Proces ima tri vhodne pretoke, in sicer kisline (Q_1), vhodne tekočine (Q_2) in baze (Q_3), ki se zmešajo v posodi T_1. Pred mešanjem gre kislina v posodo T_2, kar pomeni dodatno dinamiko v procesu. Pretoka kisline in baze sta regulirana z avtomatskimi regulacijskimi ventili, medtem ko je vhodna tekočina samo merjena z merilnikom pretoka - rotametrom. Izhodna veličina, ki jo merimo, je kislost - pH (pH) mešanice. Ker je tipalo vrednosti pH postavljeno na izhodni odtok iz posode T_1, imamo v meritvi kislosti tudi nekoliko mrtvega časa. V našem primeru je vhodna veličina v proces pretok baze (in ne vrednost pretoka vhodne tekočine). Bolj podroben opis procesa je v [9].

Dinamični model procesa pH nevtralizacije je bil izpeljan iz ravnotežnih kemijskih enačb. Izpeljani model vsebuje tudi dinamiko merilnikov in ventilov ter dinamiko hidravlike izhodnega pretoka. Predpostavke pri modeliranju so bile idealno mešanje tekočin, njihova konstanta gostota in popolna topnost ionov. Model pH procesa, opisan s sistemom enačb (3.13) in podatki iz tabele 3.1, smo namesto izvajanja meritev na nam nedostopnem procesu uporabili za pridobivanje podatkov za identifikacijo modela. Uporabljeni model vsebuje razne nelinearnosti, med njimi je vsekakor treba omeniti implicitno računano in zelo nelinearno titracijsko krivuljo, ki je tipičen element vseh pH nevtralizacijskih procesov:

$$
\begin{aligned}
\dot{x} &= f(x) + g(x)Q_3 + p(x)Q_2 \\
c(x,y) &= 0 \\
f(x) &= \left[\frac{q_1}{A_1x_3}(W_{a1}-x_1)\frac{q_1}{A_1x_3}(W_{b1}-x_2)\right. \\
&\quad \left.\frac{1}{A_1}(q_1 - C_{v4}(h_1+z)^n)\right]^T \\
g(x) &= \left[\frac{1}{A_1x_3}(W_{a3}-x_1)\frac{1}{A_1x_3}(W_{b3}-x_2)\frac{1}{A_1}\right]^T
\end{aligned}
\tag{3.12}
$$



Slika 3.20: Shema procesa pH nevtralizacije

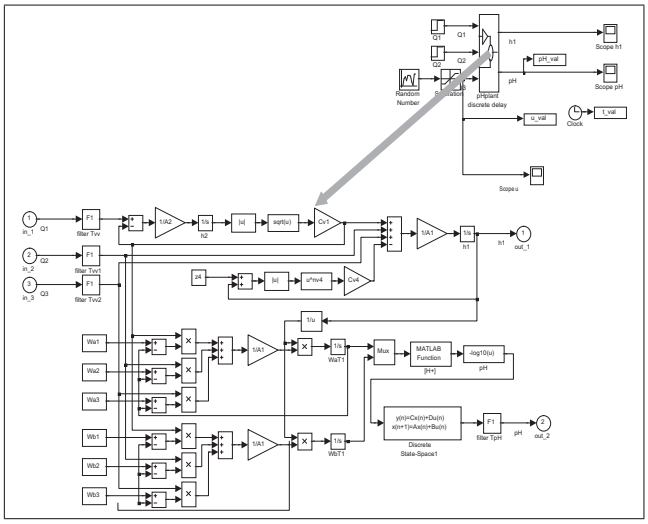
$$
\begin{aligned}
p(x) &= \left[\frac{1}{A_1x_3}(W_{a2}-x_1)\frac{1}{A_1x_3}(W_{b2}-x_2)\frac{1}{A_1}\right]^T \\
c(x,y) &= x_1 + 10^{y-14} - 10^{-y} \\
&\quad + x_2\frac{1+2\cdot 10^{y-pK_2}}{1+10^{pK_1-y}+10^{y-pK_2}} \\
pH &= y
\end{aligned}
\tag{3.13}
$$

Tabela 3.1: Parametri pH procesa v delovni točki [9]

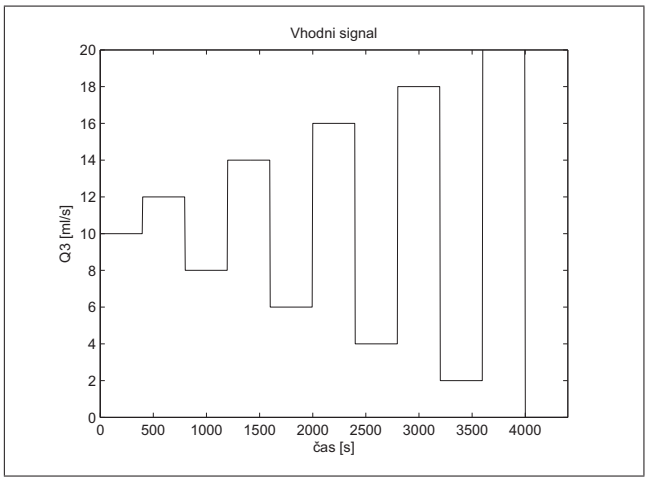
$\lvert q_1\rvert$=0.003 M HNO_3	q_1=16.6 ml/s
$\lvert q_2\rvert$=0.03 M $NaHCO_3$	q_2=0.55 ml/s
$\lvert q_3\rvert$=0.003 M NaOH,	q_3=15.6 ml/s
0.0005 M $NaHCO_3$	q_{1e}=16.6 ml/s
A_1=207 cm^2	q_4=32.8 ml/s
A_2=42 cm^2	$W_{a1}=3\cdot 10^{-3}$ M
z=11.5 cm	W_{b1}=0 M
p=0.607	W_{a2}=-0.03 M
$K_{a1} = 4.47\cdot 10^7$	W_{b2}=0.03 M
$K_{a2} = 5.62\cdot 10^{-11}$	$W_{a3} = 3.05\cdot 10^3$ M
$K_w = 1.00\cdot 10^{14}$	$W_{b3} = 5.00\cdot 10^5$ M
Δt=15 s	h_1=14 cm
Δt_c=1 s	h_2=3 cm
τ_{pH}=15 s	$W_{a4} = 4.32\cdot 10^{-4}$ M
τ_h=15 s	$W_{b4} = 5.28\cdot 10^{-4}$ M
τ_v=6 s	pH=7.0
θ=10 s	

Slika 3.21 prikazuje simulacijsko shemo za pH nevtralizacijski proces. Nelinearnost procesa lahko prikažemo tudi tako, da prikažemo odzive procesa na enak vhodni signal v različnih področjih. Tak primer je vlak pravokotnih impulzov z naraščajočo amplitudo, ki ga prikazuje slika 3.22. Odziv našega nelinearnega sistema na ta vhodni signal prikazuje slika 3.23.

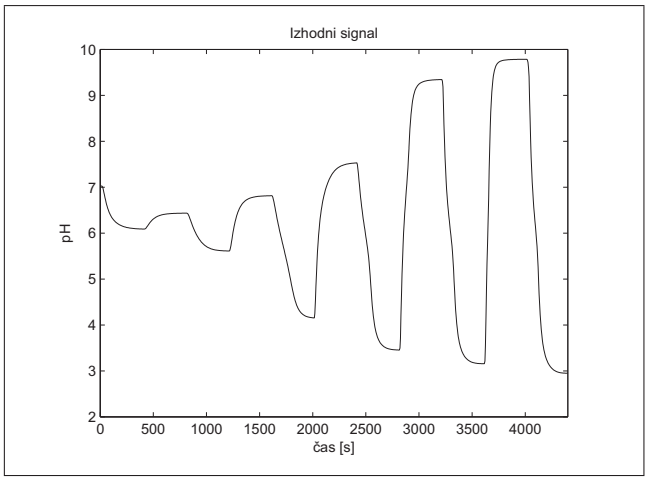
Izbrani identifikacijski signal in signal za vrednotenje ter odzive procesa nanj prikazuje slika 3.24. Kot lahko vidimo iz primerjave signalov iz slike 3.24 in porazdelitve amplitud na slikah 3.25 in 3.26, je identifikacijski signal bolj



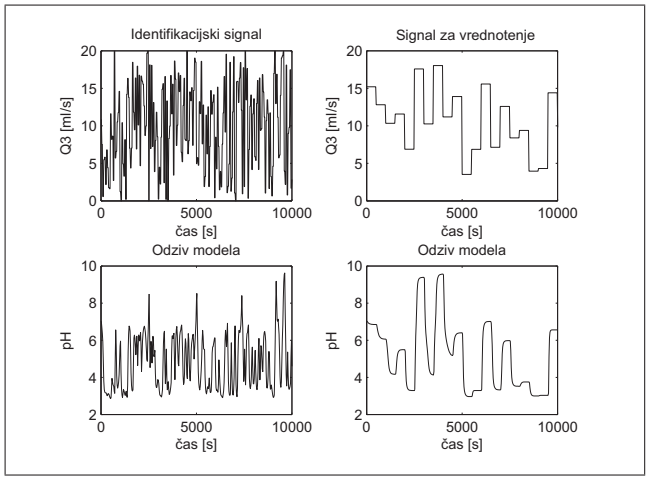
Slika 3.21: Simulink simulacijska shema pH nevtralizacijskega procesa



Slika 3.22: Vhodni signal za prikaz nelinearnosti sistema

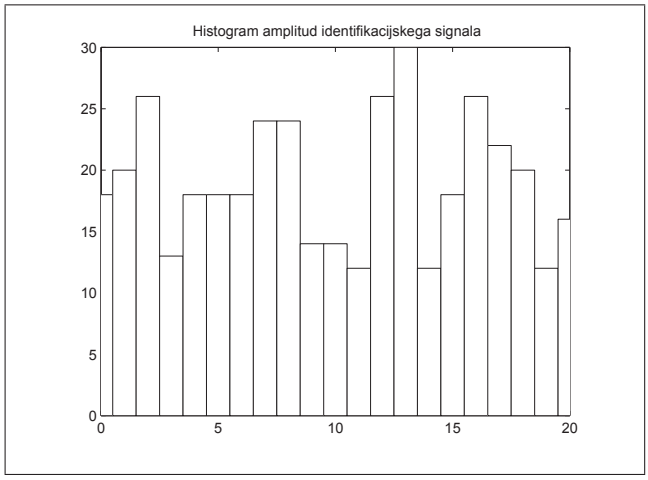


Slika 3.23: Odziv nelinearnega sistema ($T_s = 25$ sek)



Slika 3.24: Identifikacijski signal in signal za vrednotenje in njuna odziva

dinamičen in ima bogatejšo porazdelitev amplitud. To v splošnem pomeni tudi, da odziv za identifikacijo nosi večjo količino informacije o dinamiki procesa.

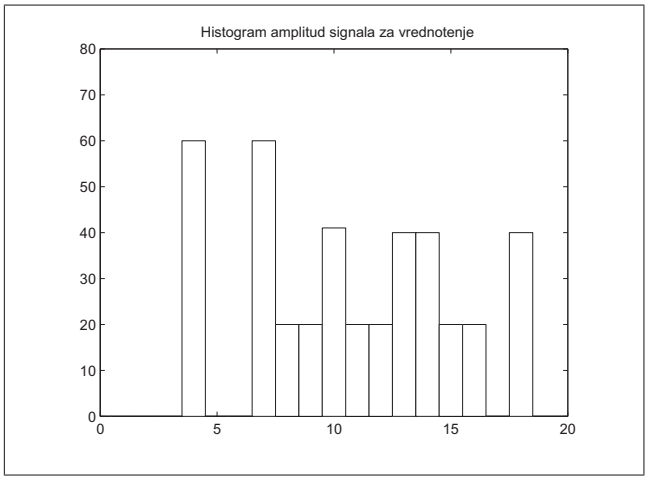


Slika 3.25: Histogram amplitud vhodnega signala za identifikacijo

Tako kot v ilustrativnem primeru v prejšnjem podpoglavju smo tudi tokrat uporabili enako programsko podporo in izbrali nevronsko mrežo z večnivojskim perceptronom za identifikacijo modela.

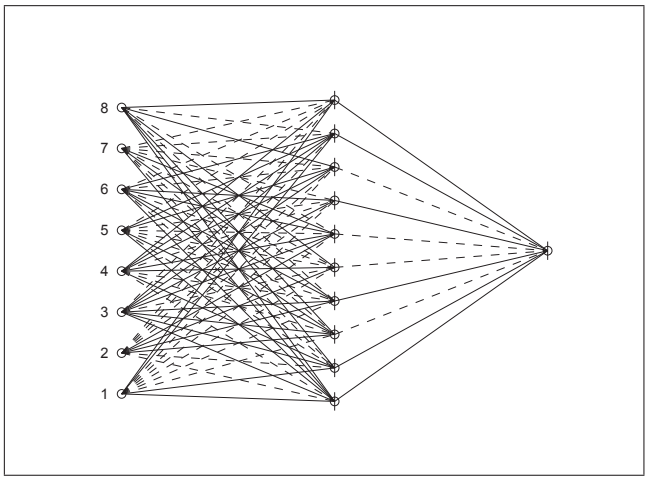
Z iterativnim postopkom smo določili regresorje, ki so dali najboljši model, vrednoten s kriterijsko funkcijo pri najugodnejšem razmerju med kvaliteto modela in čim manjšim številom regresorjev, ki določa kompleksnost modela. Odločili smo se za model z naslednjimi regresorji: $y(k-1), y(k-2), y(k-3), y(k-4), u(k-1), u(k-2), u(k-3), u(k-4)$.

Za strukturo smo izbrali NARX (metoda s posplošenim pogreškom) in v iterativnem postopku en skrit nivo in v njem 10 nevronov. Pripomniti moramo, da število skritih



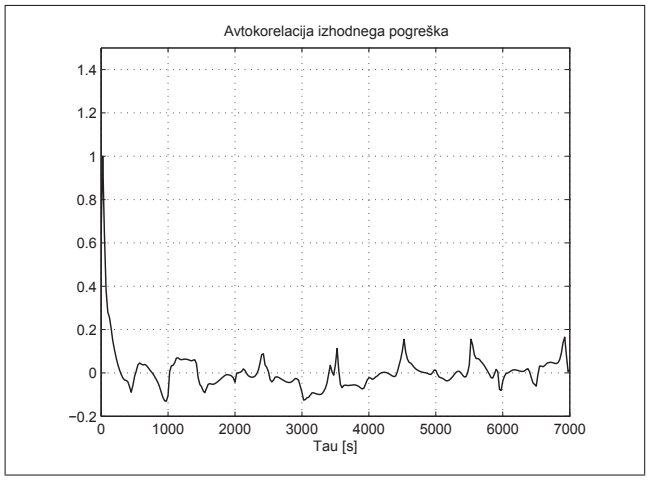
Slika 3.26: Histogram amplitud vhodnega signala za vrednotenje

nivojev (običajno je eden), število nevronov in regresorje ne izbiramo zaporedno, marveč hkrati. Gre za cikličen postopek spreminjanja posameznih elementov in vrednotenja kriterijske funkcije: za NARX model je to vsota kvadratov pogreška med odzivom procesa in modela. Parametre smo optimirali z Levenberg-Marquardt in po končani optimizaciji uporabili tudi v programsko opremo vgrajeno metodo za obrezovanje mreže, s katero smo izločili odvečne uteži, ki pa jih je bilo zelo malo. Shema strukture mreže je na sliki 3.27.

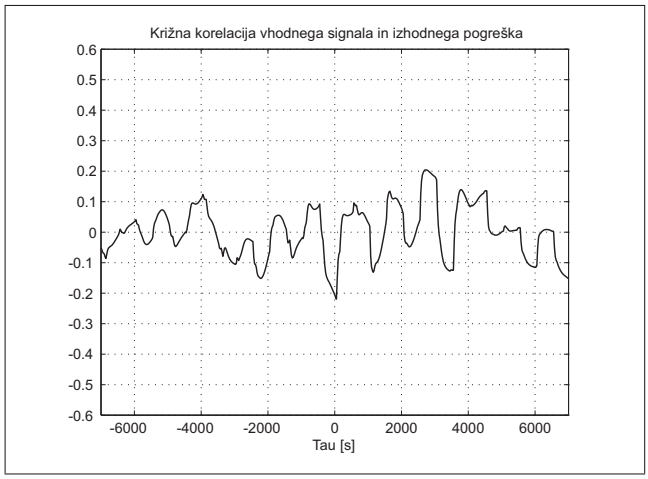


Slika 3.27: Struktura mreže

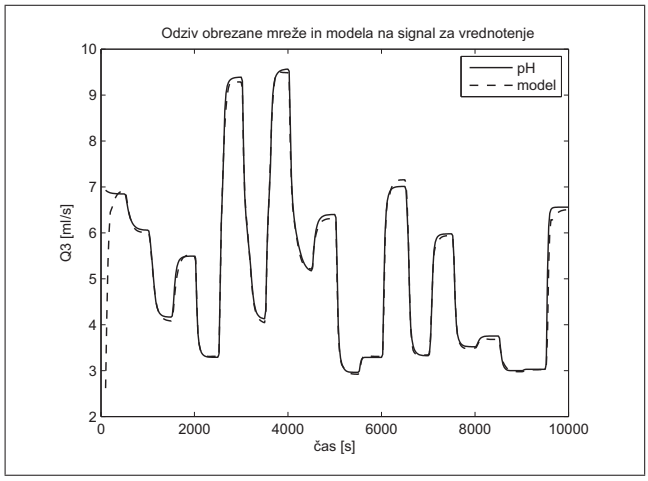
Na slikah 3.28 in 3.29 sta prikazana avtokorelacija pogreška med odzivoma procesa in modela na signal za vrednotenje dobljenim s simulacijo in križna korelacija med pogreškom odzivov in vhodnim signalom. Model vrednotimo predvsem s simulacijo, čeprav je bil v bistvu identificiran za enokoračno predikcijo. Iz prikaza simulacije odzivov procesa in modela na signal za vrednotenje na sliki 3.30 lahko potrdimo, kar sta prikazali že statistični orodji na slikah 3.28 in 3.29. To je relativno dobro ponazarjanje dinamičnega obnašanja procesa pH nevtralizacije z nevronsko mrežo.



Slika 3.28: Avtokorelacija izhodnega pogreška



Slika 3.29: Križna korelacija vhodnega signala in izhodnega pogreška



Slika 3.30: Preskus konsistence vhodno-izhodnega obnašanja na signalu za vrednotenje

Literatura

[1] A. P. Englebrecht (2002): Computational intelligence, An Introduction, Wiley and Sons, Chichester.

[2] M. Gevers (2006): A personal view of the development of system identification: A 30-year journey through an exciting field, IEEE Control System Magazine, Vol. 26, No. 6, 93-105.

[3] G. B. Giannakis, E. Serpedin (2001): A bibliography on nonlinear systems identification, Signal Processing, Vol. 81, No. 3, 533–580.

[4] J. Sjöberg et al. (1995): Non-linear Black-Box Modeling in System Identification: a Unified Overview, Automatica, Vol. 31, No. 12, 1691-1724.

[5] J. Sjöberg (1995): Non-Linear System Identification with Neural Networks, Ph.D. Dissertation, Linköping University, Linköping.
`http://www.control.isy.liu.se/research/reports/Ph.D.Thesis/PhD381.ps.gz`

[6] M. Nørgaard, O. Ravn, N. K. Poulsen, L. K. Hansen (2000): Neural networks for modelling and control of dynamic systems, Springer, London.

[7] Neural Network Toolbox for Matlab
`http://www.mathworks.com/products/neuralnet/`

[8] The NNSYSID Toolbox for use with Matlab
`http://www.iau.dtu.dk/research/control/nnsysid.html`
in
M. Nørgaard (1995): Neural Network Based System Identification Toolbox User's Guide, Technical Report 95-E-773, Institute of Automation, Technical University of Denmark, Lyngby.

[9] M. A. Henson and D. E. Seborg (1994): Adaptive Nonlinear Control of a pH Neutralization Process, IEEE Trans. Control System Technology, Vol. 2, No. 3, 169-183.

[10] J. Stark, D. S. Broomhead, M. E. Davies, J. Huke (2003): Delay embeddings of forced systems: II Stochastic forcing, Journal of Nonlinear Science, Vol. 13, No. 6, 519–577.

4

Vodenje procesov z umetnimi nevronskimi mrežami

4.1 Principi uporabe za vodenje

Nevronske mreže so v svojem bistvu eksperimentalno dobljeni transformatorji vhodnih v izhodne podatke, iz katerih ni vidno fizikalno ali drugačno ozadje teh preslikav. Kot smo že ugotovili, gre za predstavnike modelov v obliki »črne škatle«. To moramo upoštevati tudi, ko razmišljamo o njihovi vlogi pri načrtovanju avtomatskega vodenja dinamičnih sistemov. Njihova vloga je mnogovrstna, čeprav ne tako mnogostranska, kot je vloga teoretičnih modelov dinamičnih sistemov. Nevronske mreže se uporabljajo kot modeli celotnih sistemov ali podsistemov, kot zamenjava določenih nelinearnosti, za vodenje pa kot model sistema ali njegovega inverza, kjer je potreben za delovanje regulatorja, kot je primer pri prediktivnem in adaptivnem vodenju [6].

Sistematična razdelitev raznovrstnih uporab nevronskih mrež pri načrtovanju in izvedbi avtomatskega vodenja je narejena v preglednem članku [1]; v njem so tudi reference na literaturo, v kateri so posamezni načini uporabe opisani. Ne bomo opisovali podrobnosti, marveč naj samo naštejemo načine uporabe, kot so opredeljeni v [1]. Shematsko je delitev načinov uporabe prikazana na sliki 4.1.

Uporabo delimo: na uporabo nevronskih mrež kot pomoč pri načrtovanju vodenja in na uporabo nevronskih mrež v vlogi ali kot del regulatorja.

Nevronska mreža samo kot pomoč

Vloge nevronske mreže so naslednje:

- nevronska mreža kot pomoč modeliranju:
  - temelji na posrednem cilju;
    - nevronska mreža identificira neznane dele v modelu (slika 4.2),
      - uporaba za vodenje s povratnozančno linearizacijo (slika 4.3),
      - uporaba za vodenje s trenutno linearizacijo (slika 4.4),
    - nevronska mreža napoveduje regulirano veličino,
      - nevronska mreža model za regulatorje, ki ne rabijo modela procesa, temveč le napoved regulirane veličine,
  - temelji na cilju vodenja;
    - nevronska mreža je model regulirane veličine (slika 4.5),
    - nevronska mreža je model referenčnega sistema,
    - nevronska mreža je model kriterijske funkcije,
      - nevronska mreža pomaga pri optimiranju klasičnih regulatorjev,
- nevronska mreža kot nadzorna pomoč:
  - kombinacije signalov;
- nevronska mreža kot pomoč izvedbi vodenja:
  - nevronska mreža preslika neznanke v regulatorju;
    - spremenljivi parametri regulatorja, modelirani z nevronsko mrežo,
    - vodenje z inverznim modelom dela procesa,
    - prediktivno vodenje (nevronska mreža predstavlja model procesa),
  - nevronska mreža predstavlja rešitev implicitnega zakona vodenja (slika 4.6);
    - nevronska mreža nadomešča reševanje računsko intenzivnih operacij (npr. reševanje Riccatijeve enačbe).

Nevronska mreža kot regulator

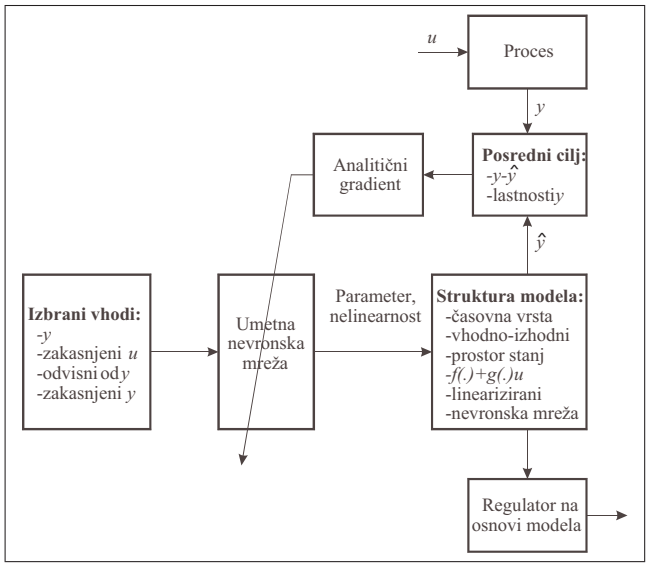
Vloge nevronske mreže so naslednje:

- učenje nevronska mreža na podlagi u:
  - nevronska mreža oponaša človeka;

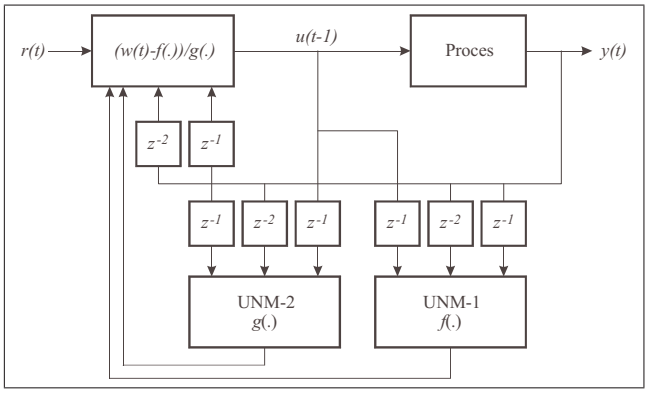
UNM v sistemih vodenja
- UNM samo kot pomoč
  - Pomoč modeliranju
    - Temelji na posrednem cilju
      - Identificira neznanke v modelu
      - Napoveduje regulirno veličino
      - Model regulirane veličine
      - Model kriterijske funkcije
    - Temelji na cilju vodenja
  - Pomoč izvedbi vodenja
    - Preslika neznanke v regulatorju
    - Reši implicitni zakon vodenja
  - Nadzorna pomoč
- UNM kot regulator
  - Učenje na osnovi u
    - Oponaša človeka
    - Oponaša regulator
    - Uporablja odprtozančne podatke
  - Učenje na osnovi cilja
    - Učenje s simulacijo na modelu
    - Učenje na procesu
      - Učenje na osnovi gradienta
        - Analitični gradient
          - Odvodi spremenljivk iz modela
          - Ad-hoc odvodi spremenljivk
          - Pretvorba napake
        - Numerični gradient
        - Pristop z iskanjem
        - Iskanje z upoštevanjem značilnosti
      - Učenje brez gradienta

Slika 4.1: Razdelitev načinov uporabe umetnih nevronskih mrež (UNM) za vodenje [1]

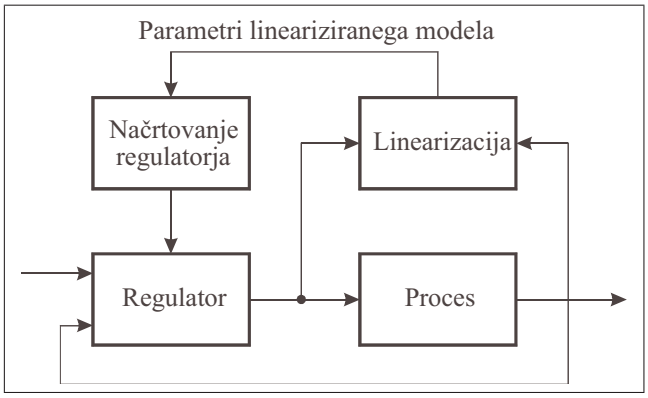
    - preslikava operaterjevih ukrepanj,
  - nevronska mreža oponaša drug regulator;
    - preslikava lastnosti konvencionalnih regulatorjev (slika 4.7),
  - nevronska mreža uporablja odprtozančne podatke (slika 4.8);
    - adaptivno vodenje z inverznim modelom,
- učenje na podlagi cilja vodenja:
  - učenje nevronske mreže s simulacijo na modelu;
  - učenje na procesu;
    - učenje na podlagi gradienta kriterijske funkcije (numerično, analitično),
    - učenje brez gradienta kriterijske funkcije (iskanje).



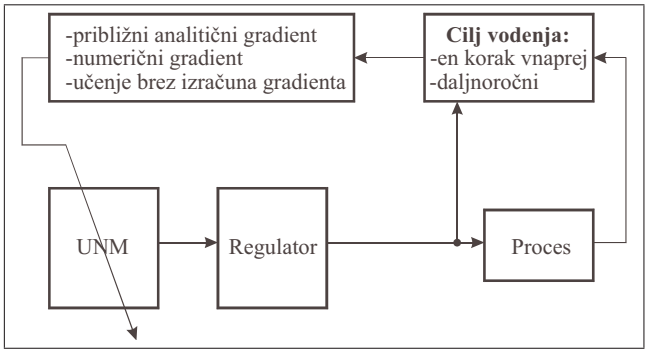
Slika 4.2: Nevronska mreža identificira neznane dele v modelu [1]



Slika 4.3: Vodenje s povratnozančno linearizacijo



Slika 4.4: Vodenje s trenutno linearizacijo

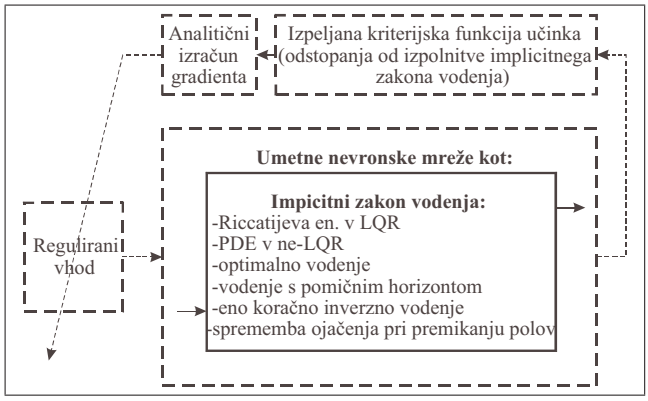


Slika 4.5: Nevronska mreža je model regulirane veličine [1]

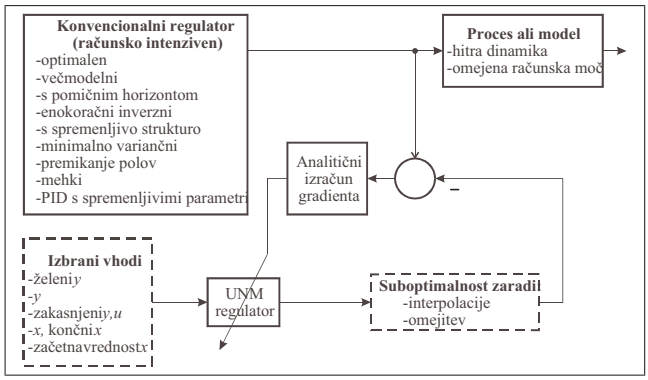
Največkrat uporabljeni načini vodenja

Pri pregledu literature, ki opisuje uporabo nevronskih mrež za vodenje dinamičnih sistemov, zasledimo, da se najpogosteje pojavljata predvsem dva načina uporabe, in sicer:

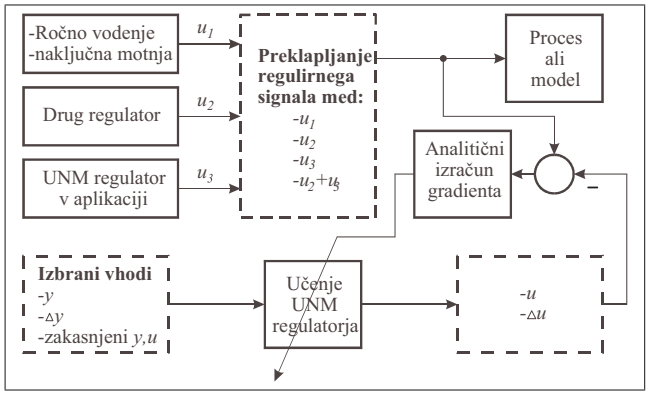
- razne oblike prediktivnega vodenja (ki so precej uporabne v praksi) in
- razne oblike adaptivnega vodenja (ki so nekoliko manj uporabne v praksi).



Slika 4.6: Nevronska mreža predstavlja rešitev implicitnega zakona vodenja [1]



Slika 4.7: Preslikava lastnosti konvencionalnih regulatorjev [1]



Slika 4.8: Nevronska mreža uporablja odprtozančne podatke [1]

Zaradi njegove uporabnosti predvsem za modele »črne škatle« si bomo v naslednjem podpoglavju podrobneje ogledali prediktivno vodenje in kako bi uporabili nevronske mreže pri njegovem načrtovanju.

4.2 Prediktivno vodenje

Osnovni koncept prediktivnega vodenja

Opis osnovnega koncepta smo povzeli po [4]. Prediktivno vodenje (angl. Model-Based Predictive Control - MPC, MBPC) je ena redkih naprednejših metod vodenja, ki se je uveljavila v industrijski praksi, predvsem na področju procesnega vodenja. Vzroke moramo iskati v enostavnem, intuitivno razumljivem principu metode, z enostavnim vključevanjem omejitev regulirnih in reguliranih veličin, enostavnem načrtovanju in uglaševanju vodenja tako univariabilnih kot multivariabilnih, linearnih in nelinearnih procesov. Prediktivno vodenje temelji na eksplicitni uporabi modela procesa za napovedovanje (predikcijo) izhoda procesa v prihodnosti, regulirni signal pa je določen na podlagi minimizacije kriterijske funkcije razlike med napovedanim potekom izhodnega signala in potekom referenčne trajektorije za določen horizont v prihodnosti. Področje prediktivnega vodenja vključuje veliko različnih algoritmov s podobnimi principi vodenja in različnimi uporabljenimi modeli in mehanizmi minimizacije kriterijske funkcije, izpeljanimi iz konkretne oblike zapisa modela (Model Algorithmic Control - MAC, Dynamic Matrix Control - DMC, Generalized Predictive Control - GPC, Predictive Functional Control - PFC, Unified Predictive Control - UPC itd.), medtem ko je splošna oznaka pri uporabi nelinearnih modelov nelinearno prediktivno vodenje (angl. Nonlinear Model-Based Predictive Control - NMPC).

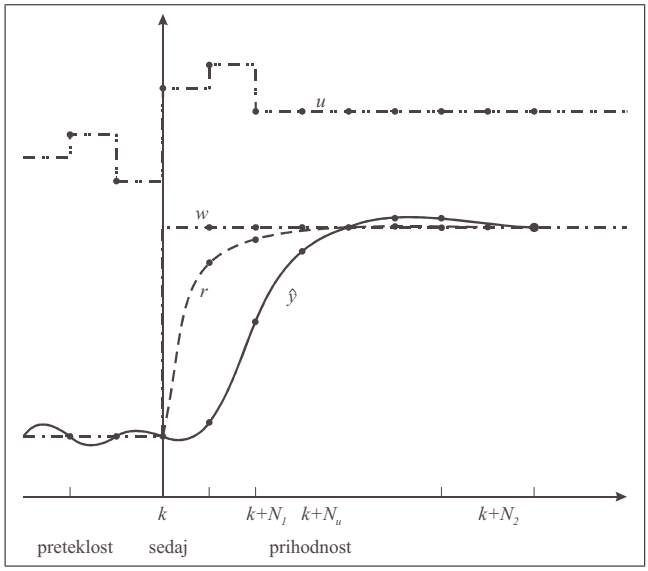
S stališča načrtovanja vodenja so poglavitne prednosti uporabe metod prediktivnega vodenja [2]:

- primerne so za vodenje procesov z zahtevnejšo dinamiko,
- primerne so za procese z mrtvim časom in minimalno fazo,
- splošen koncept omogoča vodenje tako univariabilnih kot multivariabilnih procesov,
- omogočajo predkompenzacijo (angl. feedforward) merljivih motenj,
- omejitve procesa zajamemo v postopku načrtovanja,
- pri vodenju lahko upoštevamo omejitve v velikosti reguliranega, regulirnega signala in omejitve v hitrosti sprememb regulirnega signala,
- precejšnja prostost pri načrtovanju, pri čemer gledamo na parametre načrtovanja kot na specifikacije,
- prediktivni regulator lahko ob poznavanju prihodnjega poteka referenčne trajektorije vnaprej tvori ustrezen regulirni signal,
- metode ne vsebujejo eksplicitnega odvajanja signalov (zato šum meritev ne povzroča problemov),
- metode ne vsebujejo eksplicitnega integriranja (ni problema integralskega pobega),
- princip je lahko razumljiv.

Največji slabosti uporabe prediktivnih metod sta:

- potrebujemo dober model vodenega procesa, saj je kakovost vodenja neposredno odvisna od kakovosti modela,
- računska zahtevnost metod lahko postane problematična pri vodenju hitrejših procesov.

Temeljni principi prediktivnega vodenja so razvidni v naslednjih korakih (slika 4.9):



Slika 4.9: Temeljni koncept prediktivnega vodenja

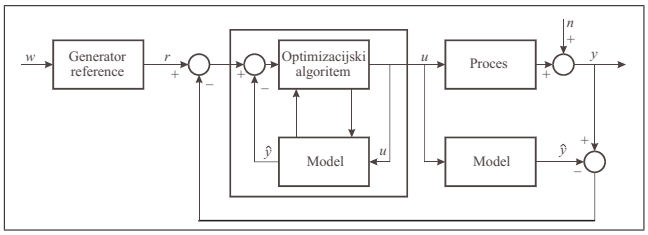
- Napoved (predikcija) izhodnega signala procesa na podlagi modela procesa. Ob vsakem časovnem trenutku k izračunamo potek izhodnega signala $y(k+j)$ za horizont v prihodnosti $j = (N_1, \ldots, N_2)$, kjer z N_1 in N_2 označimo spodnjo in zgornjo vrednost predikcijskega horizonta, ki določa horizont ujemanja (angl. coincidence horizon), znotraj katerega želimo doseči ujemanje izhodnega signala s predpisanim obnašanjem. Napovedane vrednosti izhodnega signala procesa, izračunane z modelom procesa, označene kot $\hat{y}(k+j \mid k)$, predstavljajo j-koračno napoved modela. Napovedane vrednosti so odvisne od regulirnega scenarija v prihodnosti $u(k+j \mid k); j = 0, \ldots, N_u - 1$, ki ga nameravamo uporabiti od trenutka k naprej.
- Tvorjenje referenčne trajektorije. Z definiranjem referenčne trajektorije (angl. setpoint trajectory) $r(k+j \mid k); j = N_1, \ldots, N_2$ določimo želen časovni potek procesa od trenutne vrednosti $y(k)$ do predpisane referenčne vrednosti $w(k)$.

- Določitev prihodnjih regulirnih signalov. Vektor prihodnjih regulirnih signalov $u(k+j \mid k); j = 0, \ldots, N_u - 1; N_u \leq N_2$ izračunamo z minimizacijo ustrezne kriterijske funkcije, prek katere minimiziramo napovedano napako med $r(k+j \mid k)$ in $\hat{y}(k+j \mid k)$. Določanje prihodnjih signalov temelji na uporabi odprtozančnega kriterija optimalnosti na določenem intervalu v prihodnosti.

- Uporaba prvega elementa vektorja regulirnih signalov za vodenje procesa. Za vodenje uporabimo samo prvi element $u(k \mid k)$ optimalnega vektorja regulirnih signalov $u(k+j \mid k); j = 0, \ldots, N_u - 1$.

V naslednjem časovnem trenutku, ko imamo na voljo nov merjeni izhod procesa, ponovimo celotni postopek. Ta princip se imenuje strategija pomičnega horizonta (angl. receding horizon strategy). Za vsak korak je predlaganih veliko različnih rešitev, po katerih se posamezne metode prediktivnega vodenja razlikujejo med seboj.

Osnovno shemo zaprtozančnega prediktivnega sistema vodenja prikazuje slika 4.10. Medtem ko že osnovni model lahko vključuje tudi model vpliva merljivih motenj, je povratna zanka vključena v algoritem z namenom, da bi odpravili napake enokoračne predikcije modela, nastale zaradi nemodelirane dinamike in drugih (nemerljivih) motenj v procesu. Napako izračunamo kot razliko izhoda procesa in modela v času vzorčenja:



Slika 4.10: Skupni koncept predlaganih principov prediktivnega vodenja

$$e(k) = y(k) - \hat{y}(k). \tag{4.1}$$

Pri napovedovanju izhoda procesa v prihodnosti napako $e(k)$ prištejemo predikciji modela $\hat{y}(k+j \mid k)$, za vsak $j = N_1, \ldots, N_2$. S tem ko predpostavimo konstantno napako $e(k)$ za celoten predikcijski horizont, kompenziramo napake modela v ustaljenem stanju in konstantne motnje v procesu. Prediktivni algoritem na podlagi odstopanj med vrednostmi izhoda modela in meritev oceni prihodnji vpliv nemerljivih motenj, ki delujejo na proces. V idealnem primeru ujemanja modela s procesom in odsotnosti motenj je povratna zanka nedelujoča, vodenje je primer odprtozančnega optimalnega vodenja.

Za napoved izhodnega signala procesa je bistvena uporaba modela procesa, s katerim je mogoče izračunati napoved izhodnega signala več korakov v prihodnost. Načelno lahko uporabimo poljuben linearni ali nelinearni model. Mednje spadajo poleg teoretičnih tudi eksperimentalni modeli. Primeri takih modelov procesov, ki se uporabljajo za načrtovanje prediktivnega vodenja, so: nevronske mreže, mehki modeli, modeli na podlagi Gaussovih procesov itd. Tako dobljeni modeli so normalno zapisani v časovno diskretni obliki.

Pri uporabi nelinearnih modelov lahko različne prediktivne algoritme z nelinearnim modelom delimo glede na način reševanja nelinearnega regulacijskega problema:

- Neposredni nelinearni način zapiše regulacijski problem v obliki problema nelinearnega programiranja in ga rešuje z iterativnimi optimizacijskimi metodami. Tak pristop izhaja direktno iz ideje o uporabi nelinearnega modela pri predikciji, vendar je zaradi reševanja nekonveksnih optimizacijskih problemov z dodatnimi omejitvami računsko zahteven.

- Linearizacijski načini pa zaradi poenostavitev optimizacijskega problema in uporabe linearnih prediktivnih algoritmov vodenja proces linearizirajo. Znane in od nelinearnega zapisa modela odvisne so uporabe povratnozančne linearizacije ali inverzne nelinearne preslikave, sprotne linearizacije modela v delovni točki, uporabe množice lokalnih linearnih modelov, tvorjenje odzivov na stopnico v posameznih delovnih točkah itd.

Vse metode prediktivnega vodenja predpostavljajo poznavanje poteka referenčnega signala v prihodnosti, toda ta predpostavka ni vedno izpolnjena. Običajno predpišemo, na kakšen način naj se izhod procesa približuje referenčnemu signalu, kar določimo s tvorjenjem referenčne trajektorije $r(k+j \mid k); j = N_1, \ldots, N_2$. Referenčno trajektorijo si lahko predstavljamo kot interno referenco prediktivnega regulatorja, ki določa želeno zaprtozančno obnašanje in na podlagi katere določimo vektor prihodnjih regulirnih signalov. Pri tvorjenju referenčne trajektorije glede na poznavanje referenčnega signala ločimo dve situaciji:

1. Referenčni signal $w(k+j)$ je znan vnaprej za vse $j = 1, \ldots, N_2$. Zaradi principa prediktivnega vodenja, ki napoveduje obnašanje v prihodnosti, prediktivni regulator povzroči ustrezno regulirno akcijo še pred spremembo referenčnega signala. Tako kompenziramo mrtvi čas in velike časovne zakasnitve procesa. Referenčni signal je lahko vnaprej poznan pri robotskih sistemih, sledilnih sistemih, šaržnih procesih itd.

2. Referenčni signal $w(k+j)$ ni znan vnaprej. Kot najboljšo možno napoved referenčnega signala vzamemo njegovo trenutno vrednost: $w(k+j) = w(k)$.

Glede na začetno točko (inicializacijo) referenčne trajektorije ločimo dve situaciji:

1. Uporabimo trenutno merjeno vrednost izhodnega signala $r(k) = y(k)$. Na ta način vnesemo v sistem dodatno povratno zanko, katere vpliv na celoten zaprtozančni sistem težko ovrednotimo, saj ni rezultat optimizacije kriterijske funkcije. V nekaterih primerih lahko tako celo destabiliziramo sistem.

2. Uporabimo trenutno vrednost referenčne trajektorije $r(k)$.

Odprtozančno optimalna določitev odziva sistema v določenem intervalu prihodnosti je glavna lastnost prediktivnega vodenja. V tej točki zasledimo podobnost z linearno kvadratnim (angl. linear quadratic - LQ) regulatorjem [3], vendar ta uporablja neskončno dolg interval prihodnosti (predikcijski horizont). Izbira ustrezne kriterijske funkcije je prvi korak pri določanju vektorja prihodnjih regulirnih signalov $u(k+j \mid k); j = 0, \ldots, N_u - 1$. Pri tem težimo k izbiri kriterijskih funkcij, ki so enostavno izračunljive in katerih minimum lahko poiščemo analitično ali z optimizacijskimi algoritmi. Ker imamo z referenčno trajektorijo že določeno želeno obnašanje sistema v prihodnosti, je logična izbira kriterijskih funkcij razlike med napovedanim odzivom procesa in referenčno trajektorijo, npr.:

$$J = \sum_{j=N_1}^{N_2} (r(k+j) - \hat{y}(k+j))^2. \tag{4.2}$$

Parametra N_1 in N_2 določata horizont ujemanja, v katerem želimo, da se napovedani izhod procesa čim bolj ujema z referenčno trajektorijo. Z večanjem parametra N_1 izpuščamo vpliv napak vodenja v bližnji prihodnosti, kar posebno pri fazno minimalnih sistemih ali sistemih z mrtvim časom pelje k bolj umirjenemu in gladkemu vodenju. Razširjeno obliko kriterijske funkcije opisuje enačba:

$$J = \sum_{j=N_1}^{N_2} \alpha_j (r(k+j) - \hat{y}(k+j))^2. \tag{4.3}$$

Z vektorjem uteži $\boldsymbol{\alpha} = [\alpha_{N_1}, \ldots, \alpha_{N_2}]$ je mogoče dodatno vplivanje na pomembnost napak ob posameznih časovnih trenutkih v horizontu ujemanja. Pogosta je tudi vključitev mere za varianco regulirnega signala v kriterijsko funkcijo, prek katere (za ceno povečanja odstopanj med napovedanim izhodom procesa in referenčno trajektorijo) skušamo zmanjšati spreminjanje regulirnega signala:

$$J = \sum_{j=N_1}^{N_2} (r(k+j) - \hat{y}(k+j))^2 + \sum_{j=0}^{N_u} \beta (\Delta u(k+j))^2. \tag{4.4}$$

Zgornja kriterijska funkcija vsebuje vektor sprememb regulirnega signala $u(k)$. Pri številnih metodah uporabljamo določanje optimalnega vektorja sprememb regulirnih signalov $\Delta u(k+j \mid k); j = 0, \ldots, N_u - 1$ in ne absolutnih vrednosti $u(k+j|k); j = 0, \ldots, N_u - 1$.

Napovedani signal v prihodnosti $\hat{y}(k+j|k); j = 1, \ldots, N_2$ je odvisen od predvidenega vektorja regulirnih signalov v prihodnosti $u(k+j|k); j = 0, \ldots, N_u - 1; N_u \leq N_2$. V splošnem so elementi vektorja poljubni in medsebojno neodvisni, kar z večanjem velikosti N_u izjemno poveča računsko zahtevnost in časovno potratnost optimizacije. Regulirni signal lahko tako postane tudi bogat z neželenimi visokimi frekvencami. V praksi se vedno odločimo za strukturiranje vektorja regulirnih signalov z uvedbo relacij med elementi vektorja. Taka odločitev tudi poveča robustnost prediktivnega vodenja. Zaradi principa pomičnega horizonta in uporabe le prvega elementa vektorja regulirnih signalov za vodenje pa dejanski regulirni signal zaradi uvedbe strukturiranja ni omejen. Najpogosteje uporabljene tehnike strukturiranja regulirnega signala [5] so:

- Uvedba regulirnega horizonta po prehodnem pojavu predpostavlja konstanten regulirni signal ob predpostavki o konstantnem referenčnem signalu v prihodnosti. Regulirni horizont $N_u; N_u \leq N_2$ predstavlja časovni trenutek, od katerega naprej bo regulirni signal ostal konstanten. S tem zmanjšamo število spremenljivk optimizacije (elementov vektorja regulirnih signalov). Najenostavnejši in v praksi pogosto uporabljen limitni primer je $N_u = 1$, ki daje dobre rezultate tudi v primeru spreminjajočega se referenčnega signala.

- Tehnika grupiranja, ob predpostavki $N_u = N_2$, razdeli celoten predikcijski horizont na določeno število segmentov, znotraj katerih predpostavi konstanten regulirni signal.

- Razvoj po baznih funkcijah strukturira regulirni signal kot linearno kombinacijo neodvisnih, vnaprej določenih baznih funkcij (linarnih, polinomskih itd.). Izbira baznih funkcij je odvisna od procesa in referenčnega signala. Izračun regulirnega signala se tako skrči na izračun parametrov izbranih baznih funkcij.

Po določitvi oblike kriterijske funkcije in strukture vektorja prihodnjih vhodnih signalov sledi določitev vrednosti elementov tega vektorja. Pri uporabi linearnih modelov, brez uvedb omejitev, rešitev poiščemo z določitvijo minimuma konveksnega optimizacijskega problema. Rešitev pogosto lahko določimo analitično (izhod procesa je linearno odvisen od preteklih vrednosti vhodov in izhodov sistema) ali pa z enokoračnimi optimizacijskimi metodami, npr. metodo najmanjših kvadratov. Uvedba omejitev pri minimizaciji kriterijske funkcije z uporabo linearnih modelov zahteva iskanje minimuma kriterijske funkcije z uporabo iterativnih optimizacijskih algoritmov, optimizacijski problem pa ostane konveksen. Uporaba nelinearnih modelov v splošnem prinese nekonveksnost optimizacijskega problema, z uporabo časovno in računsko zahtevnih iterativnih algoritmov pa konvergenca k pravi rešitvi ni več zagotovljena znotraj predpisanega časa oziroma števila iteracij optimizacije. Dodatna vključitev omejitev procesnih spremenljivk v optimizacijo pri izbranem optimizacijskem algoritmu lahko privede celo do nedoločljivosti minimuma kriterijske funkcije ob upoštevanju omejitev (npr. pri nepredvideno velikih motnjah). Obe možnosti sta za izvedbo

nelinearnega prediktivnega vodenja na realnih procesih nesprejemljivi. V literaturi je predlaganih več izhodov ob nastopu take situacije, in to od vodenja v proces predhodne nespremenjene vrednosti regulirne veličine do vodenja drugega elementa optimalnega vektorja regulirnih signalov, določenega v predhodnem koraku, itd. Taki optimizacijski postopki (angl. interior point methods) [5] pazijo na upoštevanje časovnih in rezultatskih omejitev v vsakem koraku optimizacije in dajo vsaj približno rešitev, če je potrebno zaradi časovne omejitve optimizacijo predčasno prekiniti ali pa optimizacijski postopek ne najde rešitve optimizacijskega problema.

Zanimivo možnost predstavlja »mehčanje« omejitev, pri kateri privzamemo, da omejitve sicer predstavljajo meje, ki naj jih ne bi smeli prestopiti, toda prek strukture kriterijske funkcije uvedemo mehanizem, ki to ob izrednih razmerah dovoli. Pri tem je potrebno definirati dve množici omejitev:

1. omejitve, ki jih zaradi fizičnih, varnostnih in podobnih razlogov ne moremo prekoračiti (npr. območje in hitrost spremembe regulirne veličine sta omejeni z uporabljenim izvršnim sistemom) in jih ne mehčamo,
2. druge omejitve, ki jih lahko mehčamo (npr. v primeru izredne situacije lahko prekoračimo meje, ki določajo sprejemljivo kakovost izdelka).

Mehčanje omejitev izvedemo z razširitvijo uporabljene kriterijske funkcije z dodatno funkcijo mehkih omejitev, ki je neničelna samo v primeru, ko so omejitve prekoračene. Na ta način optimizacijski algoritem omejitve prekorači le v nastopu izredne situacije (npr. ob nepredvideno velikih motnjah v procesu) [5].

Za vodenje uporabimo samo prvi element $u(k|k)$ optimalnega vektorja regulirnih signalov $u(k+j|k); j = 0, \ldots, N_u - 1$. Zaradi strategije pomičnega horizonta in končne dolžine predikcijskega horizonta odziv regulacijskega sistema v splošnem ni enak optimalnemu odprtozančnemu odzivu, na podlagi katerega je bil določen optimalni vektor regulirnih signalov.

Ilustrativni primer vodenja s prediktivnim regulatorjem

Pogledali si bomo pogledali primer vodenja s prediktivnim regulatorjem, ki za napovedovanje uporablja nevronsko mrežo.

Vodenje načrtujemo za proces, ki ga matematično opišemo z nelinearno diferenčno enačbo (3.10):

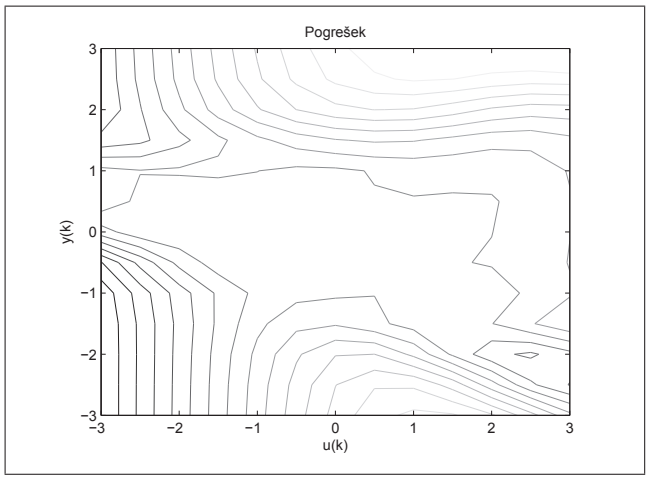
$$y(k) = y(k-1) - 0.5\tanh(y(k-1) + u^3(k-1)),$$

kjer sta u vhodni in y izhodni signal iz sistema. Postopek načrtovanja lahko strnemo v tri korake:

1. korak - Identifikacija procesa v delovnem področju: Ker gre za enak proces, kot smo ga z nevronsko mrežo modelirali že v podpoglavju 3.1, naj samo povzamemo ključne lastnosti:

- regresorja: $y(k-1)$, $u(k-1)$,
- struktura: NARX (metoda s posplošenim pogreškom),
- en skrit nivo s petimi nevroni večnivojskega perceptrona,
- metoda optimizacije: Levenberg-Marquardt.

Vrednotenje modela smo prav tako prikazali že v podpoglavju 3.1. Tokrat si poglejmo le izohipse pogreška enokoračne predikcije na sliki 4.11, ki razkrivajo delovno področje, v katerem je model uporaben.



Slika 4.11: Vrednotenje pogreška odziva (enokoračna predikcija)

2. korak - Načrtovanje prediktivnega vodenja:

- Izbrali smo prediktivno funkcijsko vodenje (angl. Predictive Functional Control - PFC) kot eno najenostavnejših oblik [5]. Značilnost večine PFC metod je skrčenje horizonta ujemanja na nekaj točk. V primeru ene same točke horizont ujemanja (tudi predikcijski horizont) imenujemo točka ujemanja (angl. coincidence point). Podrobnejši opis uporabljene metode PFC najdemo npr. v delih [5] in [4].
- Kriterijska funkcija pri PFC, ki smo jo uporabili, je:

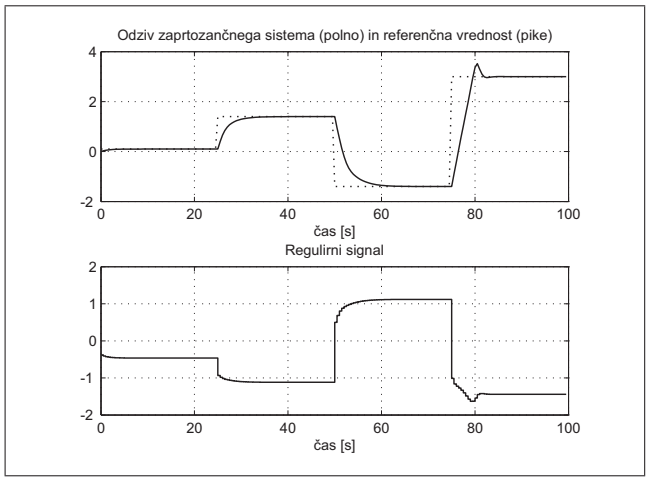
$$J = \min_{\mathbf{U}(k)} \left[r(k+P) - \hat{y}(k+P)\right]^2. \qquad (4.5)$$

- Kot točko ujemanja smo s poskušanjem izbrali vrednost osmih vzorcev vnaprej. Referenčna trajektorija je bila izbrana tako, da se vrednost reguliranega signala približuje referenčni vrednosti eksponentno s časovno konstanto 10 vzorcev.

- Nekatere pomembne lastnosti, na katere moramo biti pozorni pri načrtovanju prediktivnega vodenja:
  - računska zahtevnost,
  - robustnost zaprtozančnega vodenja na odstopanja med procesom, ki ga vodimo, in njegovim modelom,
  - upoštevanje omejitev raznih veličin, s katerimi imamo opravka.

3. korak - Vrednotenje načrtovanega vodenja:

Odziv zaprtozančnega vodenja in ustrezni regulirni signal prikazuje slika 4.12. Iz nje lahko vidimo, da deluje prediktivno vodenje v skladu z zastavljeno referenčno trajektorijo, dokler deluje v področju, kjer model zadosti dobro opisuje proces. Pri zadnji stopničasti spremembi pa zaprtozančni sistem zaide izven področja dobrega opisovanja modela, kar ima za posledico odziv, ki ni v skladu s postavljenimi zahtevami. Temu se lahko ognemo tako, da omejimo delovanje na področje, kjer je model dovolj dober.



Slika 4.12: Odziv prediktivnega vodenja na podlagi modela z nevronsko mrežo

Literatura

[1] M. Agarwal (1997): A systematic classification of neural-network-based control, IEEE Control Systems Magazine, Vol. 17, No. 2, 75-93.

[2] K. Kavšek-Biasizzo (1998): Prediktivno vodenje nelinearnih sistemov, doktorska disertacija, Univerza v Ljubljani, Fakulteta za elektrotehniko, Ljubljana.

[3] W. S. Levine (1996): The Control Handbook, CRC Press, Boca Raton, Fl.

[4] B. Likar (2004): Prediktivno vodenje nelinearnih sistemov na osnovi Gaussovih procesov, magistrsko delo, Univerza v Ljubljani, Fakulteta za elektrotehniko, Ljubljana.

[5] J. M. Maciejowski (2002): Predictive control with constraints, Pearson Education Limited, Harlow.

[6] M. Nørgaard, O. Ravn, N. K. Poulsen, L. K. Hansen (2000): Neural networks for modelling and control of dynamic systems, Springer, London.

[7] M. Nørgaard (1995): Neural Network Based Control System Design Toolkit User's Guide, Technical Report 96-E-830, Institute of Automation, Technical University of Denmark,Lyngby. `http://www.iau.dtu.dk/research/control/nnctrl.html`

5

Mreže lokalnih modelov in zliti večmodelni sistemi

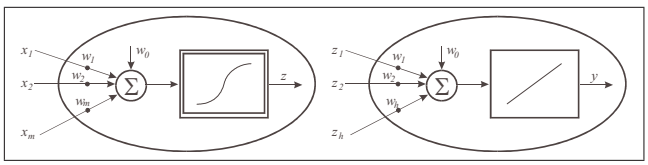
Večmodelni sistemi so skupen pojem za različne načine modeliranja in tudi načrtovanja vodenja za nelinearne sisteme, ki jih sestavimo iz manj kompleksnih podsistemov. Glede na to, iz katerega okolja izhajajo taki sistemi, se imenujejo različno. Nekateri najbolj znani večmodelni sistemi so: mreže lokalnih modelov, mehki Takagi-Sugeno modeli, zliti večmodelni sistemi, večmodelni sistemi s preklapljanjem, Markovske mešanice ekspertov in drugi. Pregled večmodelnih sistemov za modeliranje najdemo v knjigi [15], nekatera kasnejša dognanja pa v reviji, posvečeni tej problematiki [7].

Ker smo v prejšnjih poglavjih obravnavali nevronske mreže, tudi sedaj izhajajmo iz te smeri. Z vidika nevronskih mrež lahko gledamo na večmodelne sisteme kot na mrežo lokalnih modelov, ki jo bomo okvirno opisali v naslednjih poglavjih. Bistvo je enako tudi za druge oblike večmodelnih sistemov.

Nivojske mreže

Kot smo videli, nevronske mreže lahko razdelimo v tiste s slemenasto in v tiste z radialno strukturo. Tipičen predstavnik nevronske mreže s slemenasto strukturo je večnivojski perceptron (slika 5.1), pri katerem nelinearnost modeliranega sistema aproksimiramo s slemenastimi baznimi funkcijami:

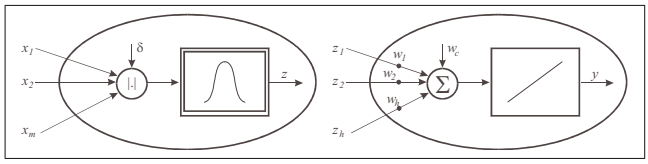
$$z_i = f_i \left(\sum_{j=1}^{m} w_{ij} x_j + w_{0j} \right). \tag{5.1}$$



Slika 5.1: Večnivojski perceptron

Predstavnik nevronskih mrež z radialno strukturo so mreže radialnih baznih funkcij (slika 5.2), pri katerih nelinearnost modeliranega sistema aproksimiramo z različno uteženimi centri, to je s točkami, razsutimi po prostoru regresorjev:

$$z_i = \exp \left(-\frac{1}{2} \sum_{j=1}^{m} \frac{(x_j - \gamma_{ij})^2}{\rho_{ij}^2} \right). \tag{5.2}$$

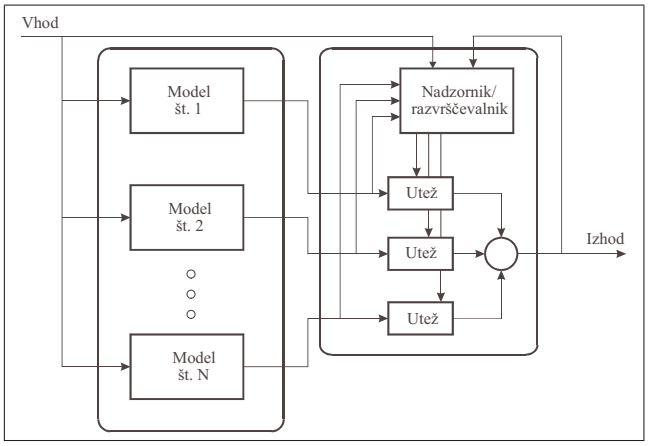


Slika 5.2: Mreža radialnih baznih funkcij

Ti centri so bolj ali manj enakomerno razporejeni po celotnem zanimivem prostoru (slika 1.16). Tipično so povezani z občutnim zlivanjem tudi v delovnih točkah nelinearnega sistema, pri katerem imamo običajno precej merjenih podatkov, če delamo identifikacijo sistema.

Mreže lokalnih modelov in zliti večmodelni sistemi

Za centre radialnih baznih funkcij po navadi izberemo posamezne točke vhodnih regresorjev, ki predstavljajo centre množic merjenih podatkov. Te množice podatkov lahko nadomestimo kar z lokalnimi modeli, dobljenimi iz teh podatkov, in na tak način občutno zmanjšamo količino podatkov za modeliranje, ne vplivamo pa na količino informacije. Namesto da bi uteževali centre podatkov tako, da bi dobili čim boljše ujemanje modelirane nelinearnosti, utežujemo raje lokalne podmodele in jih združujemo v globalni model modelirane nelinearnosti. To strukturo imenujemo mreža lokalnih modelov (angl. Local Model Network - LMN). Optimizacijo parametrov, po navadi so to uteži, izvajamo običajno na podoben način kot pri drugih nevronskih mrežah. Lokalne modele, ki jih lahko dobimo na poljuben način združujemo v globalen model na različne načine. Utežujemo lahko izhode posameznih lokalnih modelov ali pa, če imajo lokalni modeli enako strukturo, parametre lokalnih modelov v enotni strukturi. Enostaven princip mreže lokalnih modelov prikazuje slika 5.3.

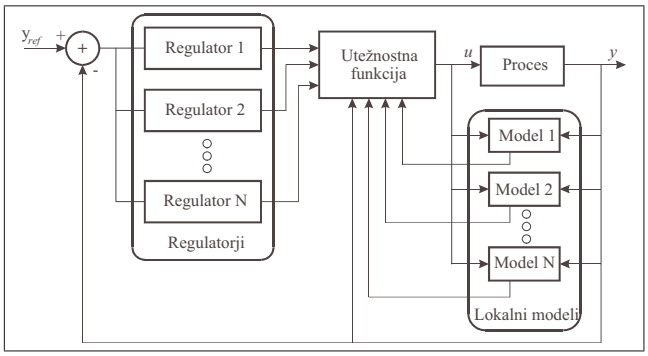


Slika 5.3: Princip mreže lokalnih modelov z utežmi na izhodih lokalnih modelov

Dinamični sistemi

Če ta princip, ki smo ga opisali, dobro deluje za opis statičnih sistemov, moramo biti pri modeliranju dinamičnih sistemov pozorni na kar nekaj dejstev.

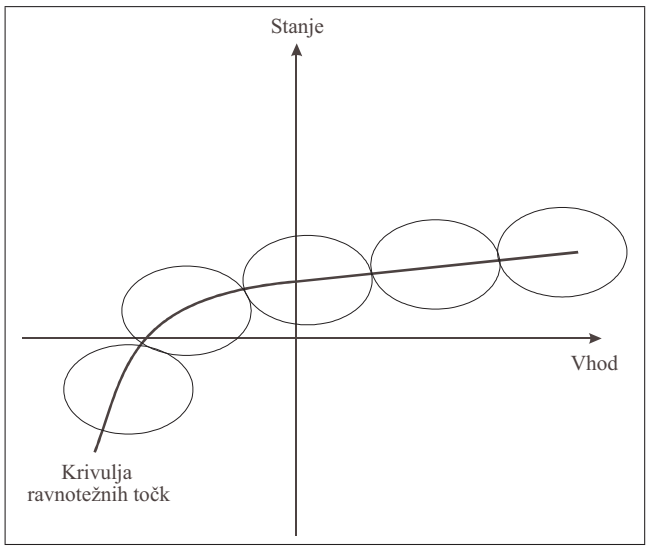
Tipičen pristop k modeliranju in tudi k načrtovanju vodenja dinamičnih sistemov je tako imenovani princip »deli in vladaj«. To pomeni, da problem (nelinearni dinamični sistem, ki ga modeliramo, ali globalni nelinearni regulator, ki ga načrtujemo) razdelimo največkrat na linearne podprobleme. Linearne zato, ker so linearne metode zelo uveljavljene. Nato rešimo posamezne linearne podprobleme, ki so veljavni samo v majhnem področju in jih povežemo v globalno nelinearno rešitev. Slika 5.4 prikazuje možno izvedbo vodenja na podlagi mreže lokalnih modelov.



Slika 5.4: Blokovna shema, ki ponazarja zaprtozančni sistem z adaptivnim regulatorjem v obliki zlitih lokalnih regulatorjev, katerih parametri se spreminjajo glede na spremembo lokalnih modelov procesa.

Linearizacija nelinearnega sistema je osnova opisanega načina modeliranja. Običajni način linearizacije je Taylorjeva linearizacija prvega reda okoli delovne točke ali pa identifikacija okoli te točke. Zavedati se moramo, da je tak pristop veljaven samo v bližnji okolici izbrane delovne točke.

Standardna metoda za modeliranje z mrežami lokalnih modelov je obravnava lineariziranih modelov okoli reprezentativnega števila ravnotežnih točk. Ravnotežne točke (slika 5.5) so tiste, h katerim teži nelinearni dinamični sistem. V časovni domeni predstavljajo te točke velikokrat konstantna ustaljena stanja, v katerih se lahko znajde nelinearni sistem. Delovne točke so običajno, ne pa nujno,



Slika 5.5: Model nelinearnega sistema kot družina linearnih sistemov (linearizacije samo v ravnotežnih točkah)

ravnotežne točke.

Pojavi se vprašanje, kako razdelimo celotno nelinearno področje delovanja nelinearnega sistema na podpodročja. Načinov je več. Sistematično so opisani v [15], opis rešitev pa najdemo tudi v [2], [18], [14].

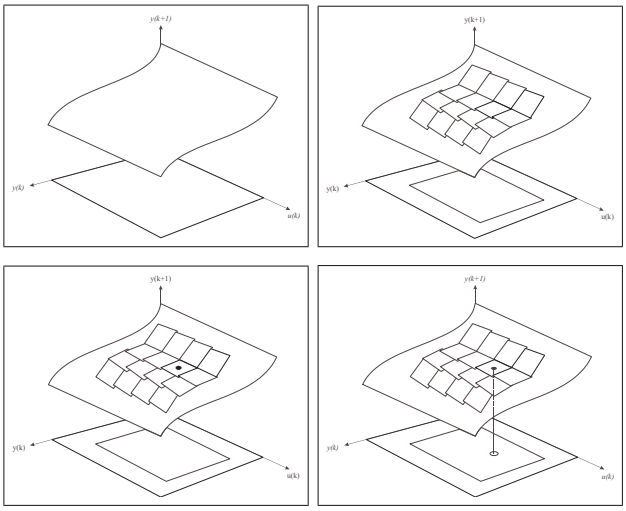
Možen pristop je, da enakomerno razdelimo področje delovanja oziroma nelinearnost, ki jo modeliramo. Gre za pristop, ki je uporaben za probleme manjše kompleksnosti in ga ilustrirata sliki 5.6 in 5.7.

Možen je tudi pristop, da področje delovanja razdelimo na delovna podpodročja glede na režime delovanja nelinearnega sistema. To je še posebej uporabno takrat, kadar modeliramo npr. industrijski dinamični proces, ki ima različne režime delovanja in pri vsakem kaže različno dinamično obnašanje. Tudi tak pristop je uporaben samo za omejeno število aplikacij in ga ilustrirata sliki 5.8 in 5.9.

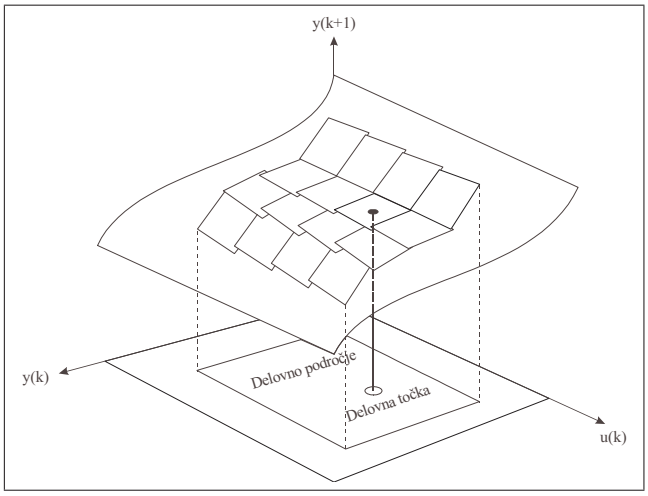
Različne načine modeliranja mrež lokalnih modelov najdemo opisane poleg v omenjenih delih še v [1], [4], s programsko opremo opisano v [5] in uporabo za vodenje v [6], [17], [19] in mnogih drugih.

Problemi, na katere moramo biti pozorni

Ko delamo linearizacijo nelinearnega sistema s Taylorjevo linearizacijo okoli delovne točke iz znanega nelinearnega



Slika 5.6: Modeliranje nelinearnega dinamičnega sistema z enakomerno razporejenimi lokalnimi linearnimi modeli

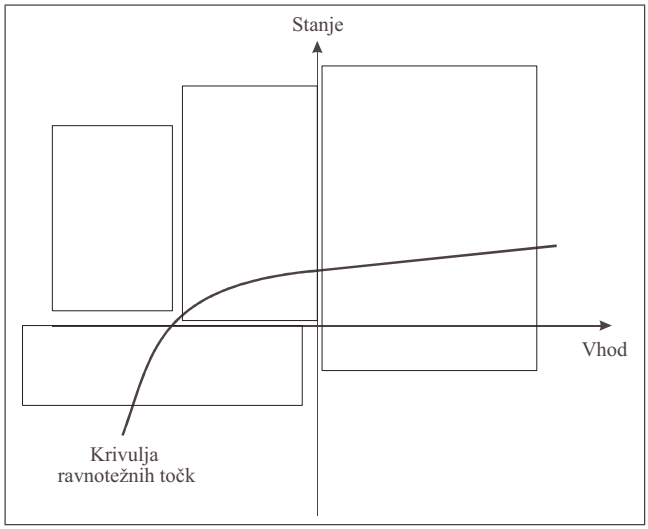


Slika 5.7: Modeliranje nelinearnega dinamičnega sistema z enakomerno razporejenimi lokalnimi linearnimi modeli - podrobnost iz slike 5.6

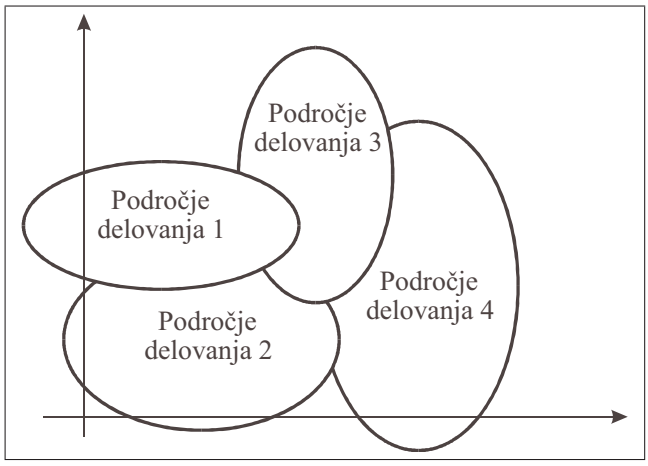
sistema, ne dobimo linearnega sistema, marveč afin sistem, ki je linearen v parametrih.

$$
\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{F}(\mathbf{x},\mathbf{u}) \text{ lineariziramo v} \\
\dot{\mathbf{x}} &= \mathbf{F}(\mathbf{x}_0,\mathbf{u}_0) + \nabla_x\mathbf{F}(\mathbf{x},\mathbf{u})(\mathbf{x}-\mathbf{x}_0) + \\
&\quad \nabla_u\mathbf{F}(\mathbf{x},\mathbf{u})(\mathbf{u}-\mathbf{u}_0) + \text{elementi višjega reda.}
\end{aligned}
$$

Enosmerni element, ki določa delovno točko sistema ($\mathbf{F}(\mathbf{x}_0,\mathbf{u}_0)$), pomeni, da pogoj superpozicije ni izpolnjen, ker sistem ni linearen (tudi če ga preoblikujemo, da je videti kot linearen). Ta enosmerni element je lahko zelo velik; to pomeni, da ne moremo reči, da gre za linearen sistem z majhno enosmerno motnjo. Ko se pomikamo od ene do druge delovne točke, se enosmerni element spreminja. To pomeni, da je povezan z dinamiko sistema, torej



Slika 5.8: Razdelitev delovnega področja nelinearnega procesa na delovna podpodročja glede na podobne lastnosti



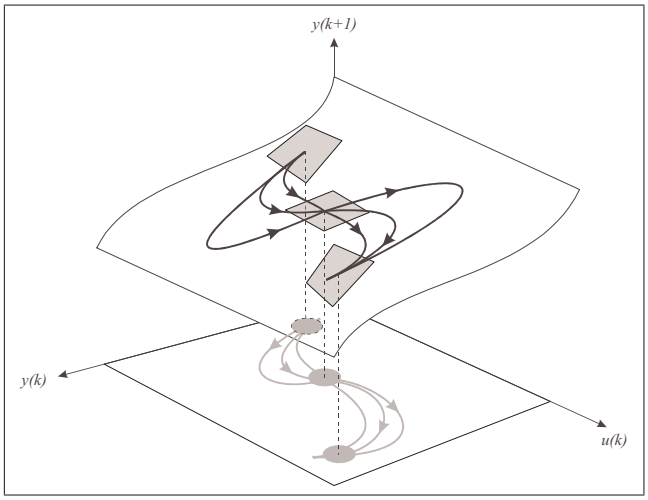
Slika 5.9: Drugačna razdelitev delovnega področja nelinearnega procesa

ga ne moremo obravnavati kot zunanjo motnjo ali celo zanemariti.

To je le eden izmed problemov, ki nastanejo pri modeliranju nelinearnih dinamičnih sistemov z mrežo lokalnih modelov.

Naslednji je problem opisovanja dinamike sistema v neravnotežnih področjih [8],[16], ki izhaja iz »težnje« stabilnih dinamičnih sistemov k ravnotežnim področjem. Sistem je zato le krajši čas na območjih stran od ravnotežne krivulje (slika 5.10) in za opis teh področij imamo na voljo le malo merjenih podatkov. Problem ne bi bil tako velik, če bi v praksi sisteme lahko poljubno vzbujali in tako dobili informacije za opis sistema na celotnem delovnem območju, vendar smo zaradi narave procesa in drugih omejitev pri načrtovanju vzbujevalnega signala običajno omejeni.

Naslednji problem je optimizacija parametrov, ki določa izbrano dinamiko in ki naj jo opisuje LMN. Na voljo imamo



Slika 5.10: Prehajanje od enega do drugega lokalnega modela

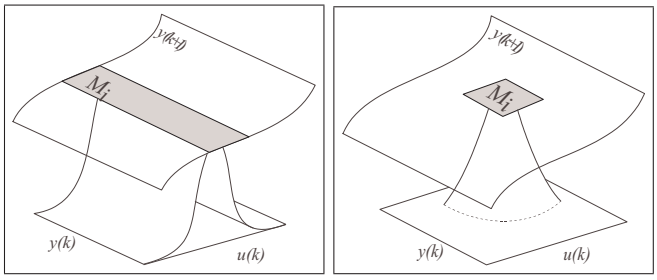
dva pristopa [16]:

- Prva možnost je opisovanje globalne dinamike neznanega sistema. V tem primeru optimiziramo vse parametre lokalnih modelov, ki sestavljajo LMN, skupaj in naenkrat, pri čemer uporabimo vse učne podatke, ki jih imamo na voljo.

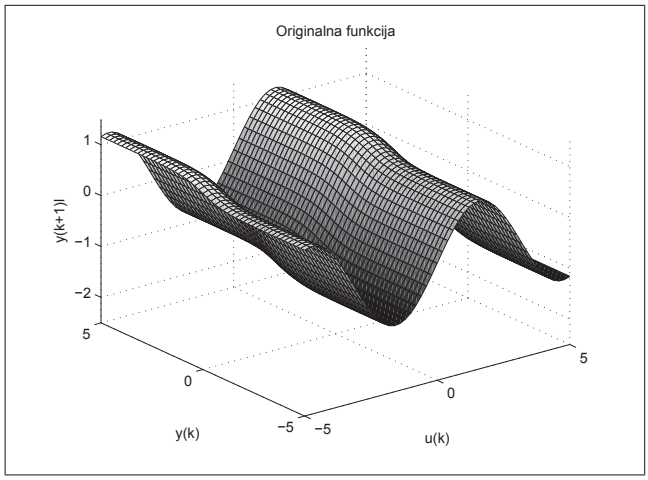
- Druga možnost je opisovanje lokalne dinamike, pri kateri za optimizacijo parametrov posameznih lokalnih modelov LMN uporabimo samo podatke, ki ustrezno področje opisujejo.

Postopek optimizacije globalne dinamike daje navadno globalno boljši opis sistema, ker se prosti parametri lokalnih modelov neravnotežnih področij lahko prilagodijo tako, da povečajo veljavnost lokalnih modelov prek širšega območja. Posledica boljšega globalnega obnašanja je izguba informacije o lokalni dinamiki, saj je parametri lokalnih modelov ne odražajo več [16], [8], [9]. Če nasprotno lokalne modele optimiziramo na drugi način, le-ti predstavljajo lokalno dinamiko in so s tem bolj transparentni ter bolj uporabni za analizo in načrtovanje vodenja, a so hkrati veljavni samo za področje, v katerem so bili identificirani (to je ponavadi majhno). Posledično lahko nekatera področja delovanja sistema ostanejo neopisana, kar nas pripelje na problem opisovanja področij stran od ravnotežne krivulje.

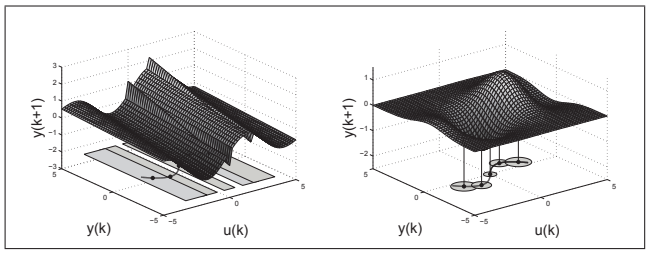
Problemu neopisanosti nekaterih področij se lahko izognemo tudi z uporabo manjšega vektorja razvrstilnih spremenljivk [8]. Ta način poveča doseg veljavnosti lokalnih modelov (slika 5.11), vendar se pojavi problem zlivanja lokalnih modelov na področjih »daleč« stran od ravnotežne krivulje. Zlivanje lahko postane ne-gladko ali celo nezvezno [3] (sliki 5.12 in 5.13), kar lahko pripelje tudi do nestabilnosti pri vodenju [3].



Slika 5.11: Pomen vektorja razvrstilnih spremenljivk



Slika 5.12: Originalni sistem [2]



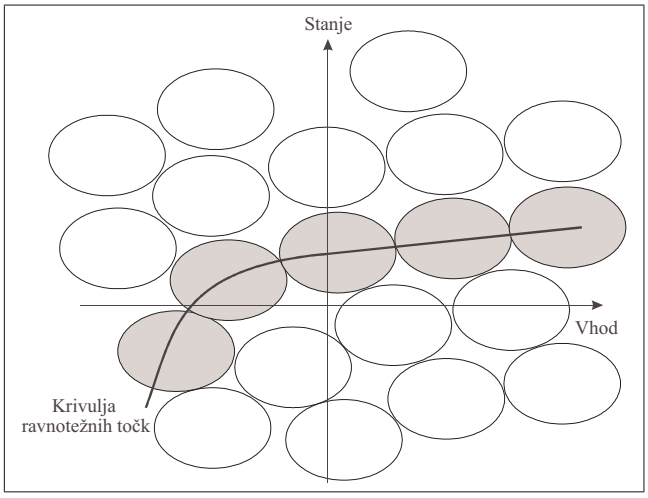
Slika 5.13: Model z reduciranim in popolnim vektorjem razvrstilnih spremenljivk [2]

5.1 Hitrostna linearizacija

V prejšnjem podpoglavju smo omenili, da Taylorjeva linearizacija ni dovolj dobro orodje, s katerim bi dobili ustrezen linearni podmodel. Ustreznejša je njena izpeljanka hitrostna linearizacija.

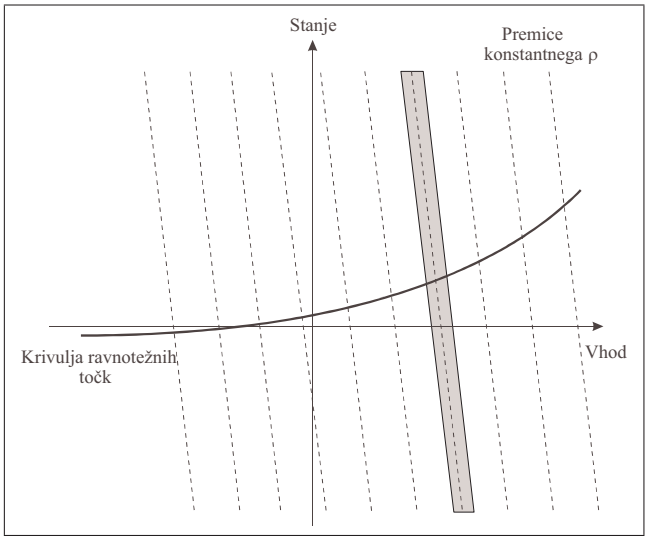
Hitrostna linearizacija (angl. velocity-based linearisation) je posplošitev običajne linearizacije okoli delovne točke: lokalni linearni sistem pripada vsaki delovni točki celotnega sistema (slika 5.14), ne samo ravnotežnim točkam. Teorija hitrostne linearizacije je opisana v člankih [10] in [11], demonstracijsko programsko opremo pa najdemo v [13].

Hitrostna linearizacija v kateri koli točki, ki lahko leži poljubno daleč od ravnotežja, je glede na vrednost raz-



Slika 5.14: Lokalni modeli

vrstilnega vektorja enaka hitrostni linearizaciji v eni od ravnotežnih točk (določeni z isto vrednostjo razvrstilnega vektorja). Grafično je to dejstvo prikazano na sliki 5.15.



Slika 5.15: Prikaz učinka hitrostne linearizacije zunaj ravnotežja

Hitrostna linearizacija

Poglejmo si, kako znan nelinearni sistem opišemo s hitrostno linearizacijo.

Nelinearni dinamični sistem je opisan s sistemom nelinearnih diferencialnih enačb

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{F}(\mathbf{x},\mathbf{u}) \\ \dot{\mathbf{y}} &= \mathbf{G}(\mathbf{x},\mathbf{u}), \end{aligned} \tag{5.3}$$

kjer so: $\mathbf{x} \in \mathcal{R}^n$, $\mathbf{u} \in \mathcal{R}^m$, $\mathbf{y}$ je vektor izhodov in $\mathbf{F}, \mathbf{G}$ sta matriki funkcijskih odvisnosti.

Ta sistem lahko ekvivalentno opišemo v obliki, pri kateri smo ločili linearni in nelinearni del:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{f}(\boldsymbol{\rho}) \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} + \mathbf{g}(\boldsymbol{\rho}) \\ \boldsymbol{\rho} &= \boldsymbol{\rho}(\mathbf{x},\mathbf{u}), \text{ z } \nabla_{\mathbf{x}}\boldsymbol{\rho}, \nabla_u\boldsymbol{\rho} \text{ konstantnima,} \end{aligned} \tag{5.4}$$

kjer so $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ ustrezno velike konstantne matrike. Nelinearni funkciji $\mathbf{f}(\cdot)$ in $\mathbf{g}(\cdot)$ sta odvisni od vektorja razvrstilnih spremenljivk $\boldsymbol{\rho} = \boldsymbol{\rho}(\mathbf{x},\mathbf{u}) \in \mathcal{R}^q$, $q \leq m+n$, ki izraža nelinearno odvisnost dinamike sistema od njegovega stanja in vhoda s konstantnima $\nabla_x\boldsymbol{\rho}$, $\nabla_u\boldsymbol{\rho}$. Če ta sistem odvajamo, dobimo

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{w} \\ \dot{\mathbf{w}} &= (\mathbf{A} + \nabla\mathbf{f}(\boldsymbol{\rho})\nabla_{\mathbf{x}}\boldsymbol{\rho})\mathbf{w} + (\mathbf{B} + \nabla\mathbf{f}(\boldsymbol{\rho})\nabla_{\mathbf{u}}\boldsymbol{\rho})\dot{\mathbf{u}} \\ \dot{\mathbf{y}} &= (\mathbf{C} + \nabla\mathbf{g}(\boldsymbol{\rho})\nabla_{\mathbf{x}}\boldsymbol{\rho})\mathbf{w} + (\mathbf{D} + \nabla\mathbf{g}(\boldsymbol{\rho})\nabla_{\mathbf{u}}\boldsymbol{\rho})\dot{\mathbf{u}}. \end{aligned}$$

Če zapis »zamrznemo« pri eni vrednosti razvrstilnega vektorja $\boldsymbol{\rho}_1$, dobimo zapis na podlagi hitrostne linearizacije v tej točki, ki predstavlja linearni dinamični sistem:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{w} \\ \dot{\mathbf{w}} &= (\mathbf{A} + \nabla\mathbf{f}(\boldsymbol{\rho}_1)\nabla_{\mathbf{x}}\boldsymbol{\rho})\mathbf{w} + (\mathbf{B} + \nabla\mathbf{f}(\boldsymbol{\rho}_1)\nabla_{\mathbf{u}}\boldsymbol{\rho})\dot{\mathbf{u}} \\ \dot{\mathbf{y}} &= (\mathbf{C} + \nabla\mathbf{g}(\boldsymbol{\rho}_1)\nabla_{\mathbf{x}}\boldsymbol{\rho})\mathbf{w} + (\mathbf{D} + \nabla\mathbf{g}(\boldsymbol{\rho}_1)\nabla_{\mathbf{u}}\boldsymbol{\rho})\dot{\mathbf{u}}. \end{aligned}$$

Osnovne značilnosti hitrostne linearizacije lahko strnemo v naslednjih ugotovitvah:

- Hitrostna linearizacija je povezana z vsako delovno točko (tako ravnotežno kot neravnotežno).
- Lastnosti modela, dobljenega s hitrostno realizacijo, so popolnoma točna preslikava lastnosti nelinearnega sistema v neposredni bližini relevantne delovne točke.
- Družina modelov, dobljenih s hitrostno linearizacijo okoli *vseh* delovnih točk, nima izgube informacije glede na nelinearno dinamiko in je alternativna predstavitev nelinearnega sistema.

Hitrostna linearizacija razreši marsikatero od pomanjkljivosti običajne teorije linearizacije okoli ravnotežnih točk v zvezi z analizo in načrtovanjem:

- Ne vsebuje omejitve na delovanje v okolici ravnotežnih točk.
- Predstavlja direktno povezavo med rešitvijo, dobljeno v obliki hitrostne linearizacije, in lokalno rešitvijo nelinearnega sistema.
- Rešitve, dobljene s hitrostno linearizacijo, so lahko sestavljene v globalno rešitev nelinearnega sistema.
- Vnaprejšnje poznavanje ravnotežnih točk ni potrebno.

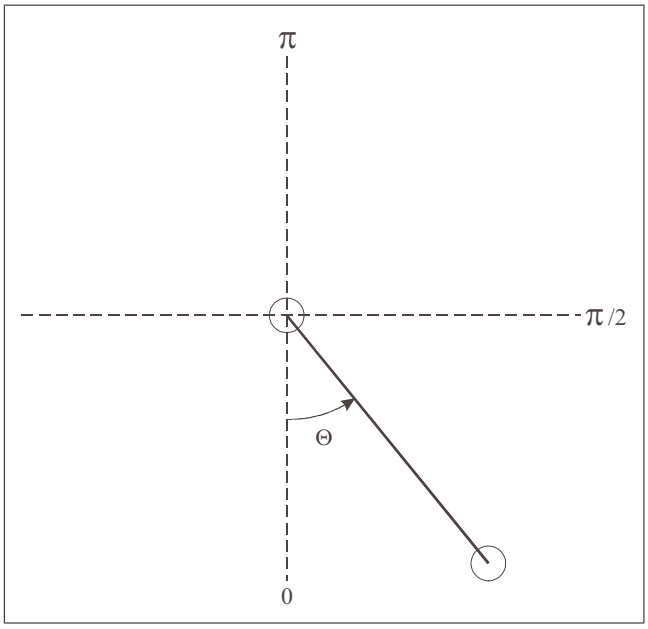
Potrebno je pripomniti, da metoda vpelje nove probleme, kot je na primer odvod vhodnega signala, ki pa so relativno enostavno rešljivi tako pri analizi kot pri načrtovanju sistemov.

Primer modeliranja sistema

Poglejmo si, kakšna je hitrostna linearizacija za dušeno nihalo [12] (slika 5.16), ki ga lahko opišemo z enačbo

$$\ddot{\theta} = Q\dot{\theta} - Q\sin\theta + bF, \tag{5.5}$$

kjer sta $Q = 29.46$ in $b = 1.21$, $\theta \in [0, \pi]$. V prostoru stanj



Slika 5.16: Shema nihala

v obliki, kjer sta ločena linearni in nelinearni del, je zapis sistema naslednji:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & -Q \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix} u \\ &\quad + \begin{bmatrix} 0 \\ -Q\sin(x_1) \end{bmatrix} \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \end{aligned} \tag{5.6}$$

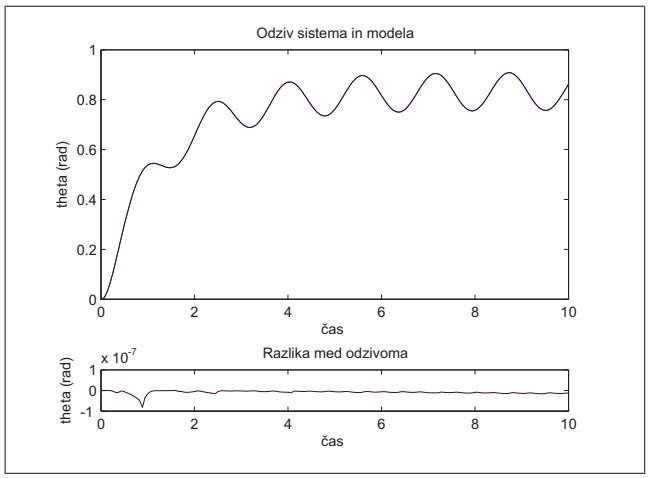
pri čemer smo vzeli θ za stanje x_1, $\dot{\theta}$ za stanje x_2 in F za vhod u.

Prikazani sistem enačb se v obliki hitrostne linearizacije glasi

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \\ \begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -Q\cos(x_1) & -Q \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \dot{u} \\ \dot{y} &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}. \end{aligned} \tag{5.7}$$

Linearizacijo npr. v točki (x_{1_0}, x_{2_0}, u_0) dobimo tako, da »zamrznemo« enačbo v tej točki. Konkretno je v našem primeru dovolj, da vstavimo v enačbo samo vrednost x_{1_0}. Tako bi dobili iz sistema enačb (5.7) v izbrani točki popolnoma linearni sistem in ne afinega, kot bi ga dobili iz sistema (5.6).

Ko simuliramo model v obliki hitrostne linearizacije, da bi ga za namene analize primerjali z originalnim sistemom, moramo $\dot{y}$ integrirati, $\dot{u}$ pa lahko dobimo prek filtra, ki predstavlja izvedbo odvajanja. Kako se znebimo odvajanja, kar pride v poštev predvsem pri načrtovanju, bo govora v naslednjem poglavju. Slika 5.17 prikazuje ujemanje obeh zapisov sistema. Do odstopanja pride zaradi aproksimacije odvoda in drugih numeričnih razlogov.



Slika 5.17: Odziv modela in sistema, ki se zelo ujemata, kar je razvidno tudi iz spodnje slike, ki prikazuje razliko med odzivoma.

* * *

Hitrostno linearizacijo lahko uporabimo tudi za diskretne sisteme, pri čemer za odvode in integrale ne smemo uporabiti nobenih aproksimacij, saj se napaka akumulira zaradi integriranja izhodnega signala.

Predstavljajmo si, da imamo nelinearni dinamični sistem, ki je predstavljen z vzorčenimi podatki v obliki diskretnega sistema:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{F}\left(\mathbf{x}(k), \mathbf{u}(k)\right) \\ \mathbf{y}(k) &= \mathbf{G}\left(\mathbf{x}(k), \mathbf{u}(k)\right), \end{aligned} \tag{5.8}$$

kjer je bil čas vzorčenja T izbran dovolj majhen, da je opis (5.8) zajel vso zanimivo nelinearno dinamiko originalnega sistema. Ta sistem želimo opisati z linearnim modelom s spremenljivimi parametri.

Da bi lahko uporabili hitrostno linearizacijo, diskretni sistem (5.8) ponazorimo v zveznem prostoru v obliki zapisa zveznih signalov z zakasnitvami:

$$\begin{aligned} \mathbf{x}(t+T) &= \mathbf{F}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \\ \mathbf{y}(t) &= \mathbf{G}\left(\mathbf{x}(t), \mathbf{u}(t)\right), \end{aligned} \tag{5.9}$$

kjer so vrednosti stanj $\mathbf{x}$, izhodov $\mathbf{y}$ in vhodov $\mathbf{u}$ diskretnega in zveznega sistema v časih vzorčenja $t = kT$, $k = 1, 2, \ldots$ enake.

Sistem (5.9) lahko izrazimo v obliki razširjene lokalne linearne ekvivalence (angl. extended local linear equivalence, ELLE) [10]:

$$\begin{aligned} \mathbf{x}(t+T) &= \tilde{\mathbf{A}}\mathbf{x}(t) + \tilde{\mathbf{B}}\mathbf{u}(t) + \mathbf{f}(\boldsymbol{\rho}), \\ y(t) &= \tilde{\mathbf{C}}\mathbf{x}(t) + \tilde{\mathbf{D}}\mathbf{u}(t) + \mathbf{g}(\boldsymbol{\rho}), \end{aligned} \tag{5.10}$$

kjer sta $\mathbf{x}(t) \in \mathcal{R}^n$, $\mathbf{u}(t) \in \mathcal{R}^m$ in so $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \tilde{\mathbf{D}}$ ustrezno velike konstantne matrike. Nelinearni funkciji $\mathbf{f}(\cdot)$ in $\mathbf{g}(\cdot)$ sta odvisni od vektorja razvrstilnih spremenljivk $\boldsymbol{\rho} = \boldsymbol{\rho}(\mathbf{x}(t), \mathbf{u}(t)) \in \mathcal{R}^q$, $q \leq m+n$, ki izraža nelinearno odvisnost dinamike sistema od njegovega stanja in vhoda s konstantnima $\nabla_x \boldsymbol{\rho}$, $\nabla_u \boldsymbol{\rho}$ [10].

Prvo enačbo iz (5.10) lahko relativno glede na delovno točko $(\mathbf{x}, \mathbf{u}) = (\mathbf{x}_0, \mathbf{u}_0)$ zapišemo z uporabo

$$\delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_0, \tag{5.11}$$

$$\delta\mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}_0, \tag{5.12}$$

kot:

$$\mathbf{x}(t+T) = \tilde{\mathbf{A}}\left(\mathbf{x}_0 + \delta\mathbf{x}(t)\right) + \tilde{\mathbf{B}}\left(\mathbf{u}_0 + \delta\mathbf{u}(t)\right) + \mathbf{f}(\boldsymbol{\rho}). \tag{5.13}$$

Pri predpostavki o lokalni linearnosti v bližini delovne točke $(\mathbf{x}_0, \mathbf{u}_0)$ sledi:

$$\begin{aligned} \mathbf{x}(t+T) - \mathbf{x}_0 + \mathbf{x}_0 &= \tilde{\mathbf{A}}\mathbf{x}_0 + \tilde{\mathbf{A}}\delta\mathbf{x}(t) + \tilde{\mathbf{B}}\mathbf{u}_0 + \tilde{\mathbf{B}}\delta\mathbf{u}(t) \\ &+ \mathbf{f}_0 + \mathbf{f}_{x0}\delta\mathbf{x}(t) + \mathbf{f}_{u0}\delta\mathbf{u}(t), \end{aligned} \tag{5.14}$$

$$\begin{aligned} \delta\mathbf{x}(t+T) + \mathbf{x}_0 &= \left(\tilde{\mathbf{A}}\mathbf{x}_0 + \tilde{\mathbf{B}}\mathbf{u}_0 + \mathbf{f}_0\right) \\ &+ \left(\tilde{\mathbf{A}} + \mathbf{f}_{x0}\right)\delta\mathbf{x}(t) + \Big(\tilde{\mathbf{B}} \\ &+ \mathbf{f}_{x0}\Big)\,\delta\mathbf{u}(t), \end{aligned} \tag{5.15}$$

$$\begin{aligned} \delta\mathbf{x}(t+T) &= (\mathbf{F}_0 - \mathbf{x}_0) + \left(\tilde{\mathbf{A}} + \mathbf{f}_{x0}\right)\delta\mathbf{x}(t) \\ &+ \left(\tilde{\mathbf{B}} + \mathbf{f}_{u0}\right)\delta\mathbf{u}(t), \end{aligned} \tag{5.16}$$

kjer so $\mathbf{F}_0 = \mathbf{F}(\mathbf{x}_0, \mathbf{u}_0)$, $\mathbf{f}_0 = \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0)$, $\mathbf{f}_{x0} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_0, \mathbf{u}_0)$ in $\mathbf{f}_{u0} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{x}_0, \mathbf{u}_0)$.

Z odvajanjem enačbe (5.16) po času dobimo:

$$\dot{\mathbf{x}}(t+T) = \left(\tilde{\mathbf{A}} + \mathbf{f}_{x0}\right)\dot{\mathbf{x}}(t) + \left(\tilde{\mathbf{B}} + \mathbf{f}_{u0}\right)\dot{\mathbf{u}}(t) \tag{5.17}$$

$$\dot{\mathbf{x}}(t+T) = \mathbf{A}(\boldsymbol{\rho})\dot{\mathbf{x}}(t) + \mathbf{B}(\boldsymbol{\rho})\dot{\mathbf{u}}(t), \tag{5.18}$$

kjer sta $\mathbf{A}(\boldsymbol{\rho}) = (\tilde{\mathbf{A}} + \mathbf{f}_{x0})$ in $\mathbf{B}(\boldsymbol{\rho}) = (\tilde{\mathbf{B}} + \mathbf{f}_{u0})$. Pri istih pogojih lahko na enak način drugo enačbo iz (5.10) zapišemo kot:

$$\dot{y}(t) = \mathbf{C}(\boldsymbol{\rho})\dot{\mathbf{x}}(t) + \mathbf{D}(\boldsymbol{\rho})\dot{\mathbf{u}}(t), \tag{5.19}$$

kjer sta $\mathbf{C}(\boldsymbol{\rho}) = (\tilde{\mathbf{C}} + \mathbf{g}_{x0})$ in $\mathbf{D}(\boldsymbol{\rho}) = (\tilde{\mathbf{D}} + \mathbf{g}_{u0})$, z $\mathbf{g}_{x0} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\mathbf{x}_0, \mathbf{u}_0)$ in $\mathbf{g}_{u0} = \frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\mathbf{x}_0, \mathbf{u}_0)$.

Na opisani način lahko predstavimo delovanje sistema v kateri koli točki, določeni z vektorjem razvrstilnih spremenljivk $\boldsymbol{\rho}$, z vektorjem stanj $\mathbf{x}(t)$ in vhodom $\mathbf{u}(t)$. Sistem (5.10) tako lahko zapišemo:

$$\begin{aligned} \dot{\mathbf{x}}(t+T) &= \mathbf{A}(\boldsymbol{\rho})\dot{\mathbf{x}}(t) + \mathbf{B}(\boldsymbol{\rho})\dot{\mathbf{u}}(t) \\ \dot{\mathbf{y}}(t) &= \mathbf{C}(\boldsymbol{\rho})\dot{\mathbf{x}}(t) + \mathbf{D}(\boldsymbol{\rho})\dot{\mathbf{u}}(t), \end{aligned} \tag{5.20}$$

kjer so matrike

$$\begin{aligned} \mathbf{A}(\boldsymbol{\rho}) &= \tilde{\mathbf{A}} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{B}(\boldsymbol{\rho}) &= \tilde{\mathbf{B}} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{C}(\boldsymbol{\rho}) &= \tilde{\mathbf{C}} + \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{D}(\boldsymbol{\rho}) &= \tilde{\mathbf{D}} + \frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\mathbf{x}(t), \mathbf{u}(t)), \end{aligned} \tag{5.21}$$

odvisne od vektorja razvrstilnih spremenljivk $\boldsymbol{\rho}$.

Poglejmo si primer modeliranja diskretnega sistema s hitrostno linearizacijo.

Primer modeliranja sistema

Zanima nas hitrostna linearizacija za nelinearni diskretni sistem [17], opisan z enačbo

$$y(k+1) = \frac{y(k)}{1 + y^2(k)} + u(k)^3 \tag{5.22}$$

in s časom vzorčenja $T = 1$ s. V prostoru stanj lahko sistem (5.22) zapišemo:

$$\begin{aligned} x(k+1) &= \frac{x(k)}{1 + x^2(k)} + u(k)^3 \\ y(k) &= x(k). \end{aligned} \tag{5.23}$$

Ponazoritev sistema (5.23) v zveznem prostoru je

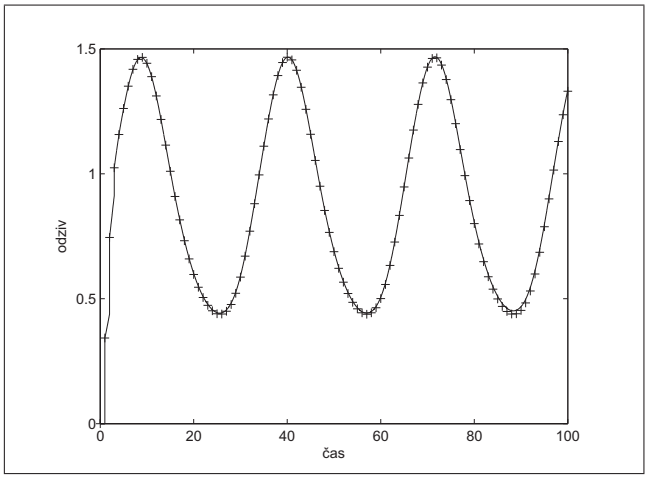
$$\begin{aligned} x(t+T) &= \frac{x(t)}{1 + x^2(t)} + u(t)^3 \\ y(t) &= x(t), \end{aligned} \tag{5.24}$$

kar ni ekvivalenten sistem, ima pa enak odziv v trenutkih vzorčenja. Če bi sistem (5.24) diskretizirali s časom vzorčenja T, bi dobili sistem (5.22).

Sistem (5.24) v obliki hitrostne linearizacije se glasi

$$\begin{aligned} \dot{x}(t+T) &= (1 - x(t)^2)/(1 + x(t)^2)^2\dot{x}(t) + 3u(t)^2\dot{u}(t) \\ \dot{y}(t) &= \dot{x}(t). \end{aligned} \tag{5.25}$$

Slika 5.18 prikazuje ujemanje odzivov sistemov (5.22) in (5.25). Odstopanje med sistemoma je nekoliko večje zaradi večjih aproksimacij, ki pa so neprimerno manjše, kot če bi skušali odvajanje in integriranje originalnega sistema nadomestiti z diskretnimi ekvivalenti ali inkrementalnim diskretnim modelom.



Slika 5.18: Odziv diskretnega sistema in zveznega modela v obliki hitrostne linearizacije

5.2 Zlivani večmodelni sistemi

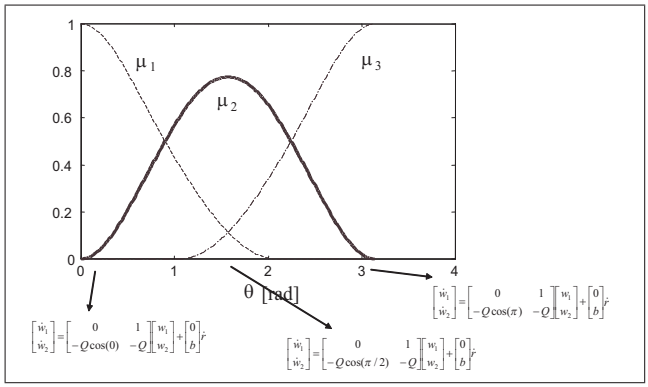
Metoda hitrostne linearizacije, kot smo jo opisali, je uporaben pripomoček za linearizacijo nelinearnih sistemov z neskončno veliko družino linearizirani podmodelov. Za prakso pa bi radi imeli tudi končno število parametriziranih modelov. Hitrostna linearizacija za primer, ko nelinearni sistem modeliramo z zlivanjem končnega števila linearnih podmodelov, je podrobno opisana v [12].

Izberemo manjše število reprezentativnih članov družine in jih interpoliramo med seboj, da nadomestimo manjkajoče: (približna) končna parametrizacija, kar je v bistvu nova oblika zlivane večmodelne predstavitve. V nasprotju z običajnimi zlitimi sistemi (mreže lokalnih modelov) imajo sistemi na podlagi hitrostne linearizacije naslednje prednosti:

- Uporabljajo linearne lokalne modele (prave linearne, ne linearne v parametrih).
- Obdržijo neposredno zvezo med dinamiko lokalnih modelov in dinamiko celotnega zlitega sistema:
  - obnašanje in lastnosti celotnega sistema so lokalno aproksimirani z obnašanjem in lastnostmi utežene linearne kombinacije lokalnih modelov;
  - še bolje, obnašanje celotnega sistema je lokalno aproksimirano z uteženo kombinacijo obnašanja in lastnostmi lokalnih modelov.

Primer modeliranja dušenega nihala

Aproksimirali bomo nelinearni sistem (5.5) in družino hitrostnih modelov z zlivanjem »lokalnih« modelov pri kotih 0, $\pi/2$ in π. Pri tem bomo za zlivanje uporabili Gaussove utežnostne funkcije, označene z μ, kot jih prikazuje slika 5.19.



Slika 5.19: Utežnostne funkcije

Model na podlagi hitrostne linearizacije v vmesni delovni točki aproksimiranega sistema dobimo z zlivanjem hitrostno lineariziranih modelov pri delovnih točkah, ki ustrezajo kotom 0, $\pi/2$ in π.

Sistem	utežene kombinacije
$=$	sistem hitrostne realizacije
$\approx$	nelinearni sistem(lokalno glede na ustrezno delovno točko)

1.

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{w}\\ \dot{\mathbf{w}} &= (\mathbf{A}+\nabla\mathbf{f}(\rho_1)\nabla_{\mathbf{x}}\rho)\mathbf{w}+(\mathbf{B}+\nabla\mathbf{f}(\rho_1)\nabla_u\rho)\dot{u},\end{aligned}$$

2.

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{w}\\ \dot{\mathbf{w}} &= (\mathbf{A}+\nabla\mathbf{f}(\rho_2)\nabla_{\mathbf{x}}\rho)\mathbf{w}+(\mathbf{B}+\nabla\mathbf{f}(\rho_2)\nabla_u\rho)\dot{u},\end{aligned}$$

3.

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{w}\\ \dot{\mathbf{w}} &= (\mathbf{A}+\nabla\mathbf{f}(\rho_3)\nabla_{\mathbf{x}}\rho)\mathbf{w}+(\mathbf{B}+\nabla\mathbf{f}(\rho_3)\nabla_u\rho)\dot{u}.\end{aligned}$$

Utežena kombinacija rešitev:

$$\tilde{\mathbf{w}} = \sum_{i=1}^{3} w_i\mu_i(\rho).$$

Lahko opazimo, kako malo lineariziranih sistemov (lokalnih modelov) smo vzeli. Dovolj so samo trije lokalni modeli, da pokrijejo celotno delovno področje [0, π]:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{w}\\ \dot{\mathbf{w}} &= \sum_{i=1}^{3}[(\mathbf{A}+\nabla\mathbf{f}(\rho_i)\nabla_{\mathbf{x}}\rho)\mathbf{w}\end{aligned}$$

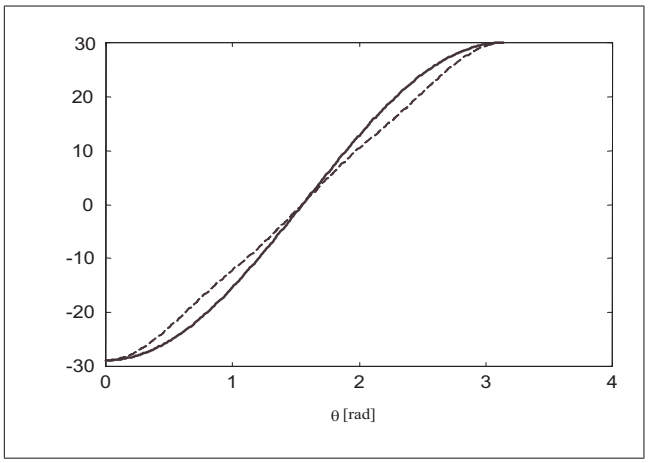
$$+ \quad (\mathbf{B} + \nabla \mathbf{f}(\rho_i)\nabla_u \rho)\dot{u}]\, \mu_i(\rho). \tag{5.26}$$

Pokažemo lahko, da velja:

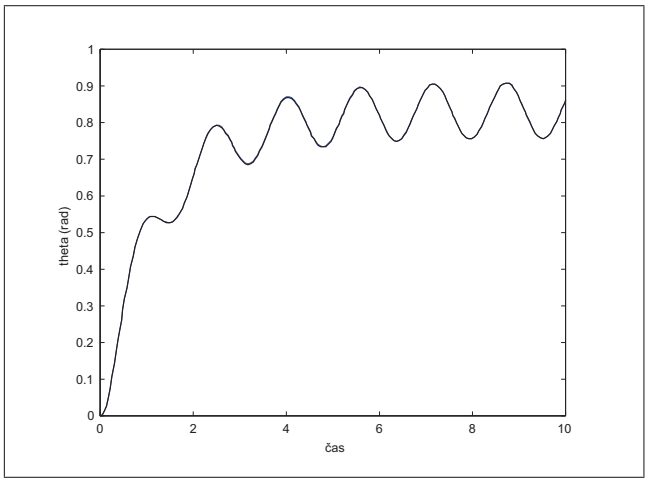
$$\begin{aligned} \tilde{\mathbf{w}} &\approx \mathbf{w}, \text{ rešitev utežene kombinacije} \\ &\approx \text{lokalna rešitev nelinearnega sistema.} \end{aligned}$$

To je neposredna povezava med dinamiko lokalnih modelov in dinamiko nelinearnega zlitega sistema ter posledica uporabe hitrostne realizacije in NE velja za običajne mreže lokalnih modelov.

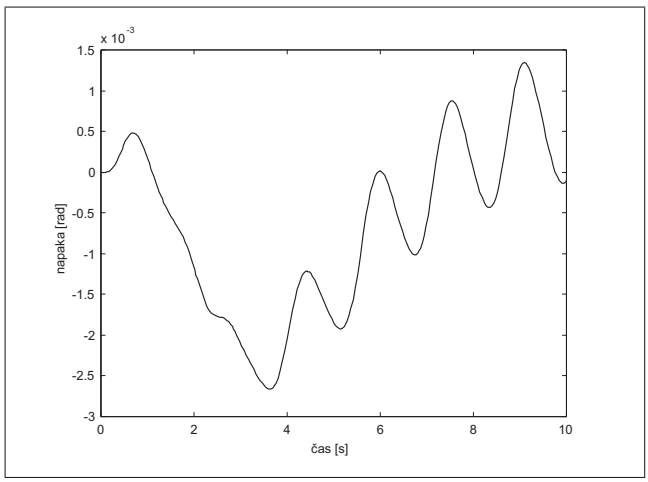
Ujemanje odzivov originalnega in zlivanega sistema prikazujejo slike 5.20, 5.21 in 5.22.



Slika 5.20: Ujemanje nelinearnosti zlivanega modela ($\dot{w}_2$) na podlagi hitrostne linearizacije in nelinearnega sistema: polna krivulja – nelinearni sistem, črtkana krivulja – zlivani sistem



Slika 5.21: Odziv nelinearnega in zlivanega sistema na določen vhodni signal – delovno območje je okoli $\pi/4$ rad, kar je, kar se tiče zlivanja, najbolj zoprno območje

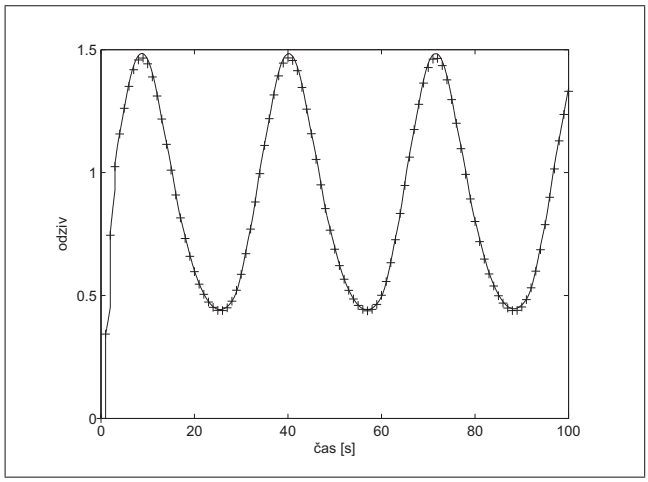


Slika 5.22: Razlika med izhodom nelinearnega in zlitega sistema

Primer modeliranja diskretnega sistema

Enako kot v prejšnjem primeru lahko postopamo tudi pri modeliranju diskretnega sistema. Diskretni sistem (5.22) smo v njegovi zvezni obliki (5.24) aproksimirali s sedmimi enakomerno porazdeljenimi lokalnimi modeli, ki smo jih zlivali z Gaussovimi utežnostnimi funkcijami.

Ujemanje odzivov originalnega in zlivanega sistema prikazuje slika 5.23.



Slika 5.23: Odziv nelinearnega diskretnega sistema (5.22) in odziv zlivanega sistema (5.26)

Literatura

[1] R. Babuška (1998): Fuzzy Modeling for Control, Norwell, MA: Kluwer.

[2] G. Gregorčič (2004): Data-based modelling of nonlinear systems for control, Ph.D. Thesis, University College Cork, National University of Ireland.

[3] G. Gregorčič, G. Lightbody (2003): From multiple model networks to the Gaussian processes prior model, Proceedings in IFAC ICONS conference, Faro, 149–154.

[4] T. A. Johansen, B. A. Foss (1993): Constructing NARMAX models using ARMAX models, Int. J. Control, Vol. 58, 1125–1153.

[5] T. A. Johansen, B. A. Foss (1998): ORBIT - Operating Regime Based Modeling and Identification Toolkit, Control Engineering Practice, Vol. 6, 1277-1286.

[6] T. A. Johansen, K. Hunt and H. Fritz (1998): A Software Environment for Gain Scheduled Local Controller Network Design, IEEE Control Systems Maganzine, 48-60.

[7] T. A. Johansen, B. A. Foss [Eds.] (1999): Int. J. Control, Special Issue: Multiple Model Approaches to Modelling and Control, Vol. 72, No. 7/8.

[8] T. A. Johansen, R. Shorten and R. Murray-Smith (2000): On the interpretation and identification of dynamic Takagi-Sugeno fuzzy models, IEEE Trans. on Fuzzy Systems, Vol. 8, No. 3, 297-313.

[9] T. A. Johansen,R. Babuška (2003): Multiobjective Identification of Takagi–Sugeno Fuzzy Models, IEEE Trans. Fuzzy Systems, Vol. 11, No. 6, 847-860.

[10] D. J. Leith, W. E. Leithead (1998): Gain-Scheduled & Nonlinear Systems: Dynamic Analysis by Velocity-Based Linearisation Families. Int. J. Control, Vol. 70, 289-317.

[11] D. J. Leith, W. E. Leithead (1998): Gain-Scheduled Controller Design: An Analytic Framework Directly Incorporating Non-Equilibrium Plant Dynamics, Int. J. Control, Vol. 70, 249-269.

[12] D. J. Leith, W. E. Leithead (1999): Analytic framework for blended multiple model systems using linear local models, Int. J. Control, Vol. 72, 605-619.

[13] D. J. Leith, W. E. Leithead (2007): Velocity-Based Gain-Scheduling Demo, [Elektronski vir], `http://www.hamilton.ie/doug/demof.htm`

[14] R. Li, Z. Zhao, X.-B. Li (2005): General Model-Set Design Methods for Multiple-Model Approach, IEEE Trans. Auto. Control, Vol. 50, No. 9, 1260-1276.

[15] R. Murray-Smith and T. A. Johansen [Eds.] (1997): Multiple Model Approaches to Modelling and Control, Taylor & Francis, London.

[16] R. Murray-Smith, T. A. Johansen, R. Shorten (1999): On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures, Proceedings in European Control Conference, Karslruhe, BA-14.

[17] K. S. Narendra, J. Balakrishnan, M. K. Ciliz (1995): Adaptation and learning using multiple models switching and tuning, IEEE Control System Magazine, Vol. 15, No. 3, 37–51.

[18] O. Nelles (2001): Nonlinear System Identification, From Classical Approaches to Neural Networks and Fuzzy Models, Springer Verlag, Berlin.

[19] T. Takagi and M. Sugeno (1985): Fuzzy identification of systems and its application to modeling and control, IEEE Trans. Syst., Man, Cybern., Vol. 15, 116–132.

6

Načrtovanje vodenja z razvrščanjem ojačenj

Dinamični sistemi povsod okoli nas so nelinearni. Žal pa so orodja za njihovo analizo in načrtovanje še relativno nerazvita. Po drugi strani pa imamo za linearne dinamične sisteme dobro razvita in znana orodja. To pomeni, da je način »deli in vladaj«, pri katerem nelinearni problem razdelimo na množico linearnih, privlačen.

Vodenje z razvrščanjem ojačenj (angl. gain-scheduling)[11] je tipična metoda, k načrtovanju katere pristopamo z načinom »deli in vladaj« in ga uporabljamo na veliko področjih avtomatskega vodenja: od letalstva do procesne tehnologije. Metoda načrtovanja vodenja z razvrščanjem ojačenj je sestavljena iz naslednjih korakov:

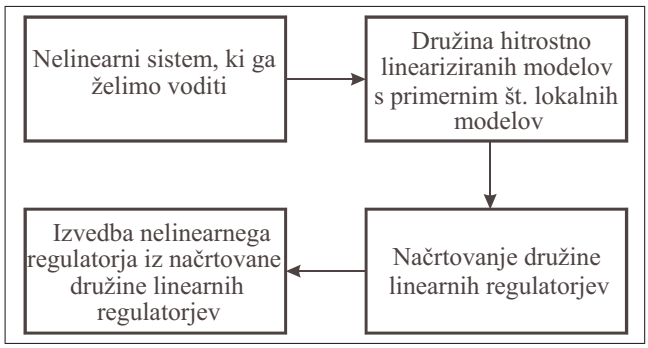
1. linearizacija nelinearnega procesa okoli izbranega števila ravnotežnih točk;
2. načrtovanje linearnih regulatorjev za vsakega od linearnih podmodelov;
3. združevanje linearnih regulatorjev v globalni nelinearni regulator s tem, da so vrednosti parametrov regulatorjev ali signali razvrščani z razvrstilnimi spremenljivkami.

Običajni regulator z razvrščanjem ojačenj je omejen na delovanje v bližini ravnotežnih točk, saj je bil načrtovan glede na tovrstne modele. Razvrstilne spremenljivke, s katerimi so razvrščani parametri ali signali linearnih regulatorjev, morajo biti počasne. To pomeni, da je njihova hitrost spremembe počasnejša od hitrosti sprememb izhodnega signala zaprtozančnega sistema. Teorija, ki bi podprla ta zelo široko uporabljani pristop, je relativno slabo razvita. Podrobnejši pregled te problematike je v delu [9].

6.1 Načrtovanje z uporabo hitrostne linearizacije

Hitrostna linearizacija, ki smo jo opisali v prejšnjem poglavju, reši precej pomanjkljivosti običajnega vodenja z razvrščanjem ojačenj. Predvsem posploši načrtovanje vodenja tako, da vsebuje informacijo o dinamiki procesa tudi v neravnotežnih delovnih točkah. Kot smo omenili v prejšnjem poglavju, je tovrsten zapis veljaven po celotnem področju in ni omejen samo na področje v bližini ravnotežnih točk. To pomeni, da lahko obvladamo tudi hitre in obsežne prehode med še tako oddaljenimi področji delovanja sistema. Načrtovanje vodenja z razvrščanjem ojačenj na podlagi hitrostne linearizacije je naslednje (slika 6.1):

1. določevanje družine hitrostno lineariziranih sistemov za nelinearni sistem;
2. načrtovanje linearnih regulatorjev za vsakega od linearnih podmodelov;
3. tvorjenje nelinearnega regulatorja iz družine hitrostno lineariziranih linearnih regulatorjev, načrtovanih v prejšnji točki.



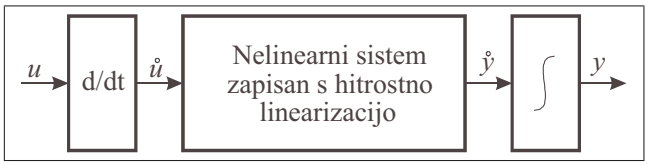
Slika 6.1: Postopek načrtovanja vodenja z razvrščanjem ojačenj (gain-scheduling) na podlagi hitrostne linearizacije

- Regulator, načrtovan s hitrostno linearizacijo, dobro deluje tudi pri velikih spremembah referenčne veličine in na veliki oddaljenosti od ravnotežnega stanja.
- Za tak regulator ni nobene zahteve po počasni spremembi katere koli veličine. To pomeni, da podpira tudi globalno invertiranje dinamike, kar bi lahko poenostavljeno opisali kot neposredno krajšanje polov in ničel reguliranega procesa v delovnih točkah.
- Prednost načrtovalnega postopka je, da lahko uporabimo znanje načrtovanja linearnih sistemov.

Podrobneje o načrtovanju vodenja z razvrščanjem ojačenj na podlagi hitrostne linearizacije najdemo v delih [6], [7] in si delovanje ogledamo v demonstracijski programski opremi [10]. Primeri uporabe so npr. [2], [3], [4].

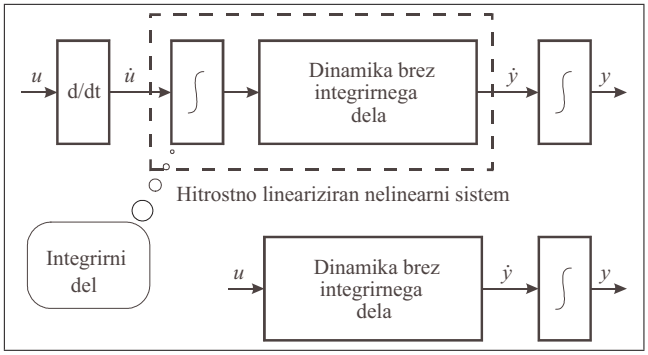
Izvedba regulatorjev, načrtovanih na podlagi hitrostne linearizacije

Oblika zapisa sistema na podlagi hitrostne linearizacije (slika 6.2) vsebuje tudi odvod vhodnega signala namesto vhodnega signala in odvod izhodnega signala namesto izhodnega signala. Odvod vhodnih signalov pri izvedbah vodenja ni zaželen, vendar se njegovi izvedbi lahko izognemo.



Slika 6.2: Sistem v obliki hitrostne linearizacije

Lahko upoštevamo dejstvo, da vsebujejo skoraj vsi regulatorji tudi integrirni del (za odpravljanje napak v ustaljenem stanju). Če pa ima regulator integrirni del, potem lahko sistem priredimo, kot prikazuje slika 6.3.



Slika 6.3: Prikaz, kako se izognemo odvodu vhodnega signala

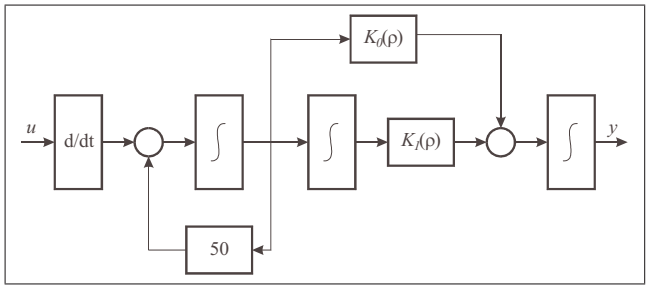
Primer PI regulatorja

Ilustrirajmo izvedbo za izogibanje odvoda signalov na enostavnem proporcionalno integrirnem (PI) regulatorju:

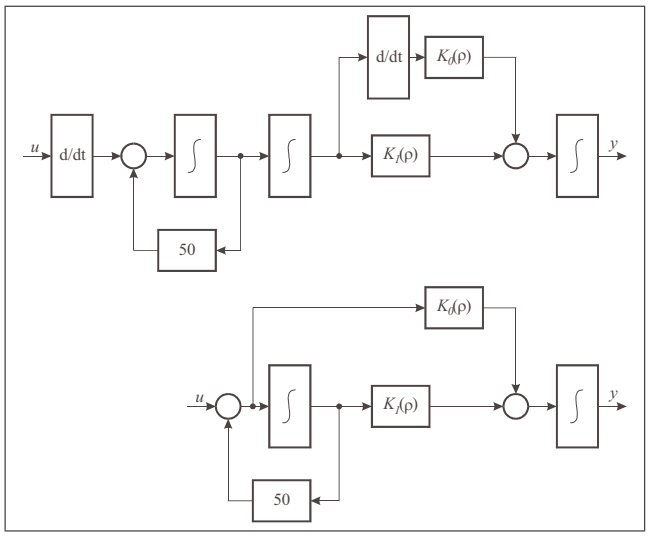
$$K(s,\rho) = (K_0(\rho) + \frac{K_1}{s})\frac{1}{(s+50)}. \qquad (6.1)$$

Slika 6.4 prikazuje osnovno in za izvedbo neželeno obliko, slika 6.5 pa, kako se odvoda izognemo.

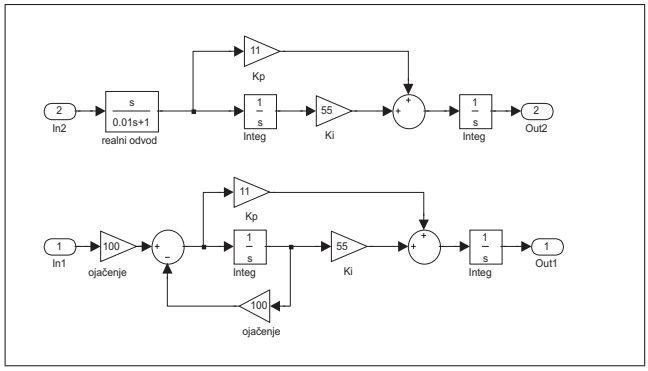
Konkreten primer izvedbe v obliki simulacijske sheme paketa Matlab/Simulink za vrednosti $K_0 = 11$ in $K_1 = 55$ prikazuje slika 6.6.



Slika 6.4: PI regulator v osnovni in za izvedbo neželeni hitrostni obliki



Slika 6.5: Končna izvedba PI regulatorja



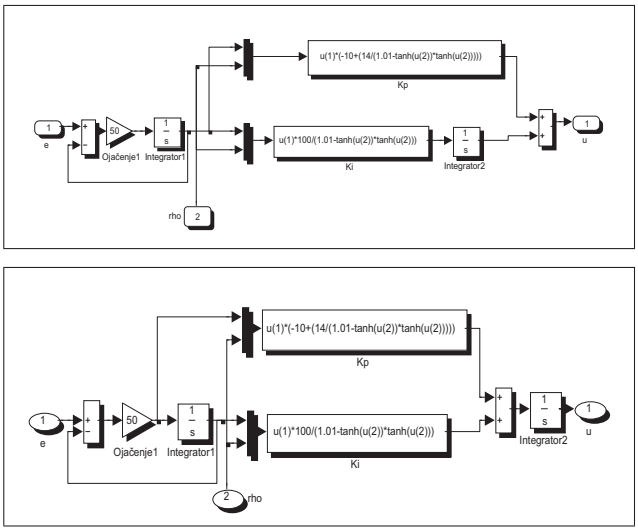
Slika 6.6: Primer izvedbe regulatorja $K(s,\rho_1) = (11 + \frac{55}{s})\frac{1}{(0.01s+1)}$ v Simulinku

Primer dveh različno izvedenih regulatorjev z razvrščanjem ojačenj

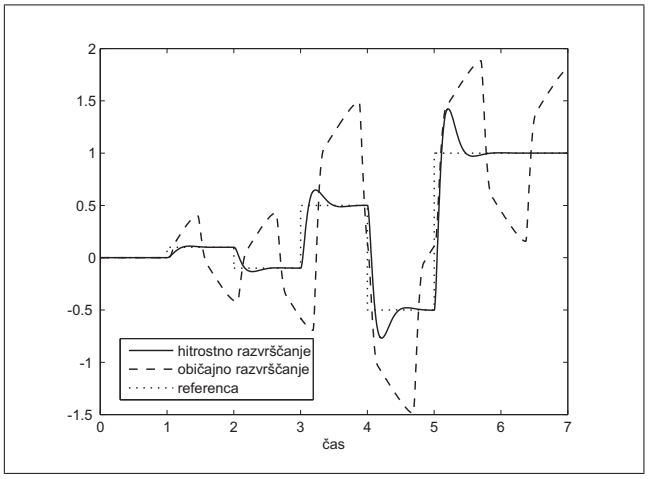
Oglejmo si načrtovanje regulatorja v obliki hitrostne linearizacije in v obliki običajnega vodenja z razvrščanjem ojačenj za nelinearni sistem:

$$\dot{y} = \tanh(u - 10y) + 0.01(u - 10y). \qquad (6.2)$$

Regulatorja, ki imata nelinearnost parametrov regulatorja enaki, imata drugačni izvedbi. Simulacijski shemi prikazuje slika 6.7, odzive zaprtozančnega sistema pa slika 6.8. Iz odzivov lahko vidimo, da običajno vodenje z razvrščanjem ojačenj ne more slediti velikim spremembam referenčnega signala.



Slika 6.7: Simulink shema za regulator v klasični izvedbi (zgoraj), regulator v hitrostni izvedbi (spodaj)



Slika 6.8: Zaprtozančni odzivi različnih izvedb

* * *

Doslej smo obravnavali družine hitrostno lineariziranih modelov z neskončno velikim številom članov, ki so ponazorili nelinearni sistem. V praksi se pogosto srečamo z željo, da bi nelinearni sistem aproksimirali z modelom, sestavljenim iz končnega števila linearnih podmodelov.

Hitrostna linearizacija omogoča tudi aproksimacijo modela procesa z interpolacijo majhnega števila lineariziranih lokalnih modelov. Poleg tega obstaja neposredna zveza med rešitvami teh »lokalnih« modelov in rešitvijo nelinearnega zlitega sistema.

Na tej podlagi lahko načrtujemo vodenje z razvrščanjem ojačenj po naslednjem postopku:

- Načrtovanje linearnega regulatorja za vsako izmed linearizacij procesa (lokalnih modelov) pri katerem dobimo majhno število linearnih regulatorjev.
- Uporaba iste vrste funkcij za zlivanje kot pri aproksimiranem modelu procesa za zlivanje regulatorjev družine na podlagi hitrostne linearizacije.
- Pod pogojem, da je model procesa dovolj točen za načrtovanje vodenja in regulator zadosti robusten, je družina regulatorjev primerna tudi za vodenje procesa. Ta pogoj velja za vse postopke načrtovanja, ki uporabljajo model procesa. Pri tem velja pripomniti, da je za uspešno načrtovanje vodenja proces potrebno aproksimirati z več kot minimalnim številom lokalnih modelov, ki zadovoljivo opisujejo nelinearno dinamiko. Pri načrtovanju vodenja namreč do neke mere ponazarjamo inverzno dinamiko procesa.
- Izvedba nelinearnega regulatorja iz načrtovane družine regulatorjev na podlagi hitrostne linearizacije.

Povzetek ugotovitev v zvezi z načrtovanjem vodenja na podlagi hitrostne linearizacije:

- Pri načrtovanju regulatorja z razvrščanjem ojačenj moramo narediti nelinearni regulator s točno določeno družino elementov, dobljenih na podlagi hitrostne linearizacije, kar je relativno enostavno.
- Nobene potrebe ni po odvajanju zašumljenih vhodnih signalov.
- Lahko dobimo nelinearni regulator, ki je neposredno povezan z družino linearnih regulatorjev, dobljenih na podlagi hitrostne linearizacije.
- Družina hitrostno lineariziranih modelov procesa ima neskončno število elementov, vendar želimo velikokrat delati z manjšo in končno množico elementov. Načrtujemo lahko regulatorje za majhno število lineariziranih modelov procesa in jih nato zlijemo z utežnostnimi funkcijami, da bi dobili en celoten nelinearen regulator za celoten proces.
- Pri izvedbi regulatorjev, dobljenih s hitrostno linearizacijo, moramo paziti, da ohranimo pridobljene prednosti.
- Zlivanje (interpolacija) je odvisno od lastnosti procesa.
- Običajno izvedeni regulatorji imajo večja odstopanja od želenega obnašanja v oddaljenosti od ravnotežnih točk.

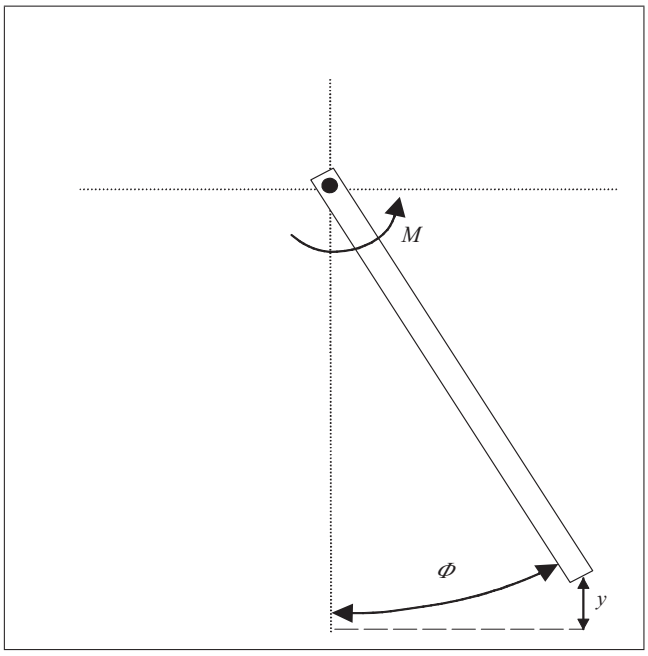
- Neposredna zveza med lastnostmi hitrostno lineariziranih modelov in lastnostmi zlitega nelinearnega modela/regulatorja zelo olajša analizo in načrtovanje. To je lastnost uporabe hitrostne realizacije in ne velja za običajne oblike zlitih sistemov (mrež lokalnih modelov, mehkih modelov ipd.).
- Ne smemo pozabiti, da je uspešna izvedba vodenja odvisna od pravilne izbire spremenljivke za razvrščanje in števila funkcij za zlivanje.

6.2 Primer načrtovanja vodenja: enosegmentni robotski manipulator

Namen tega primera, ki je podrobneje opisan v [1], je ilustrirati postopek načrtovanja vodenja z razvrščanjem ojačenj na podlagi končnega števila lineariziranih podmodelov in prikazati pomembnost pravilne izvedbe.

Prikazali bomo postopek načrtovanja vodenja in ga ovrednotili z računalniško simulacijo na primeru enosegmentnega robotskega manipulatorja. En segment robotskega manipulatorja aproksimiramo s homogeno palico, kar prikazuje slika 6.9:

$$\begin{aligned} M &= J\ddot{\Phi} + mg\sin\Phi\frac{r}{2} + k_v\dot{\Phi} \\ y &= r(1-\cos\Phi). \end{aligned} \tag{6.3}$$



Slika 6.9: Shema enosegmentnega robotskega manipulatorja

To je nelinearni sistem, ki naj ne bi predstavljal večjih problemov pri načrtovanju vodenja (ne glede na metodo).

Postopek načrtovanja je opisan v naslednjih korakih:

- Proces predstavimo z lokalnimi modeli (točke $r = 0, 0.2$ in 0.7, kar je $\Phi = 0, 36$ in 72 kotnih stopinj).
- Za lokalne modele načrtujemo lokalne PI regulatorje z zahtevo, da je zaprtozančni odziv brez prenihaja in je na vseh področjih enaka hitrost zaprtozančnega odziva.
- Lokalne regulatorje primerno povežemo v nelinearni regulator.
- Postavimo tri funkcije za zlivanje v obliki Gaussove zvončaste krivulje.
- Izhodna spremenljivka y (vertikalno odstopanje konice palice) je spremenljivka za razvrščanje.
- Dodan je pol za slabljenje visoko frekvenčnih motenj in odstopanj ter za pomoč pri izvedbi regulatorja.
- Primerjali bomo dve različni izvedbi zlivanega regulatorja.

Običajna izvedba regulatorja z razvrščanjem ojačenj

Običajna izvedba zlivanega regulatorja je naslednja:

$$\begin{aligned} \dot{\mathbf{x}}_{c_i} &= \sum_i \mathbf{F}_i(\mathbf{x}_c, e)\mu_i(\boldsymbol{\rho}) \\ u_i &= \sum_i \mathbf{G}_i(\mathbf{x}_c, e)\mu_i(\boldsymbol{\rho}) \\ \boldsymbol{\rho}(\mathbf{x}_c, e) &= \nabla_x\boldsymbol{\rho}\mathbf{x}_c + \nabla_e\boldsymbol{\rho}e, \end{aligned} \tag{6.4}$$

kjer sta ponavadi

$$\mathbf{F}_i(\mathbf{x}_c, e) = \alpha_i + \mathbf{A}_i\mathbf{x}_c + \mathbf{B}_i e, \tag{6.5}$$

$$\mathbf{G}_i(\mathbf{x}_c, e) = \beta_i + \mathbf{C}_i\mathbf{x}_c + \mathbf{D}_i e \tag{6.6}$$

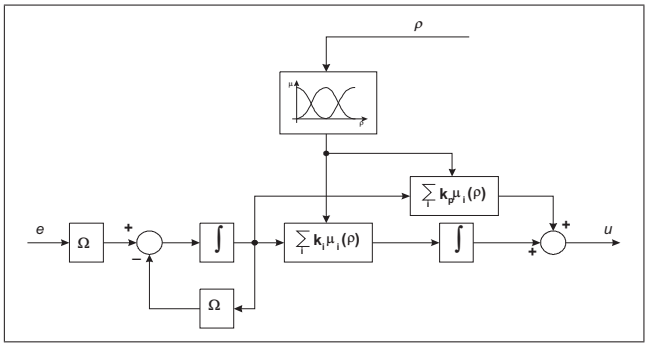
$$\tag{6.7}$$

s konstantnimi $\alpha_i, \mathbf{A}_i, \mathbf{B}_i, \beta_i, \mathbf{C}_i, \mathbf{D}_i$, ki predstavljajo linearizacijo nelinearnega sistema v delovni točki.

Primer običajne izvedbe nelinearnega PI regulatorja prikazuje bločna shema na sliki 6.10. Pomanjkljivosti običajne izvedbe lahko strnemo v dve ugotovitvi:

- Napovedljivo obnašanje samo v bližini ravnotežnih točk.
- Zahteva po počasnem spreminjanju razvrstilne spremenljivke.

Razvrstilna spremenljivka je izhod iz procesa, kar pomeni, da se razvrstilna spremenljivka NE spreminja počasi. Izvedba z zlivanjem regulirnih signalov namesto zlivanja parametrov ima poleg omenjenih še dodatne slabosti, saj se zaradi vzporedno vezanih dinamičnih regulatorjev lahko poveča red nelinearnega regulatorja proporcionalno številu aktivnih regulatorjev, kar vpliva na zaprtozančno dinamiko. Izjema je, kadar imajo lokalni regulatorji enake pole, kar je dejstvo tudi v našem primeru in se zaradi tega zaprtozančna sistema z obema vrstama običajno izvedenih regulatorjev obnašata enako.



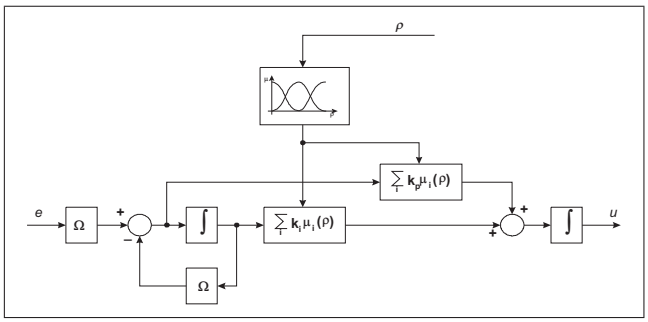
Slika 6.10: Običajna izvedba zlivanega PI regulatorja

Izvedba regulatorja z razvrščanjem ojačenj na podlagi hitrostne linearizacije

Izvedba zlivanega regulatorja na podlagi hitrostne linearizacije je naslednja:

$$\begin{aligned}\dot{\mathbf{x}}_c &= \mathbf{w}_c \\ \dot{\mathbf{w}}_c &= \left(\sum_i \mathbf{A}_i \mu_i(\boldsymbol{\rho})\right) \mathbf{w}_c + \left(\sum_i \mathbf{B}_i \mu_i(\boldsymbol{\rho})\right) \dot{e} \\ \dot{u} &= \left(\sum_i \mathbf{C}_i \mu_i(\boldsymbol{\rho})\right) \boldsymbol{w}_c + \left(\sum_i \mathbf{D}_i \mu_i(\boldsymbol{\rho})\right) \dot{e}.\end{aligned} \tag{6.8}$$

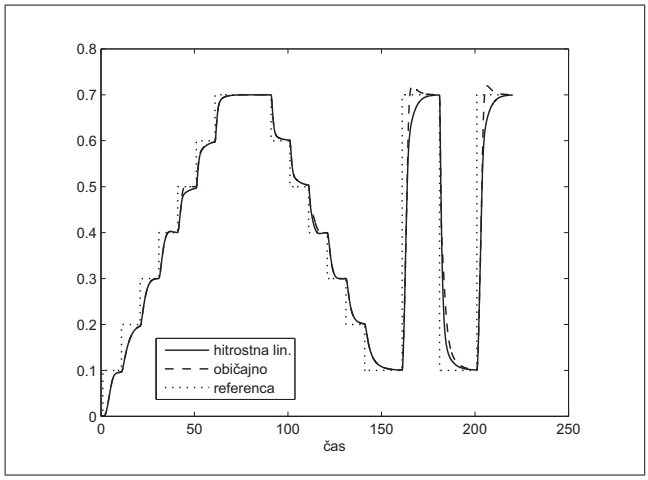
Primer izvedbe nelinearnega PI regulatorja na podlagi hitrostne linearizacije prikazuje bločna shema na sliki 6.11. Pomanjkljivost izvedbe s hitrostno linearizacijo je, da moramo biti pri njej pozorni, da jo izvedemo pravilno.



Slika 6.11: Izvedba zlivanega regulatorja na podlagi hitrostne linearizacije

Filter na vhodu regulatorja, ki smo ga izkoristili, da se izognemo neposrednemu odvajanju vhodnega signala, vsebuje dodatni pol Ω za slabljenje visokih frekvenc, ki omejuje tudi vsebnost šuma v signalu.

Primerjavo odzivov zaprtozančnih odzivov z različno izvedenimi regulatorji prikazuje slika 6.12, na kateri lahko vidimo, da se zaprtozančni sistem z običajno izvedenim regulatorjem ne obnaša, kot bi se moral pri velikih spremembah referenčne vrednosti, pri katerih sistem hitro prehaja z enega področja nelinearnosti na drugega.



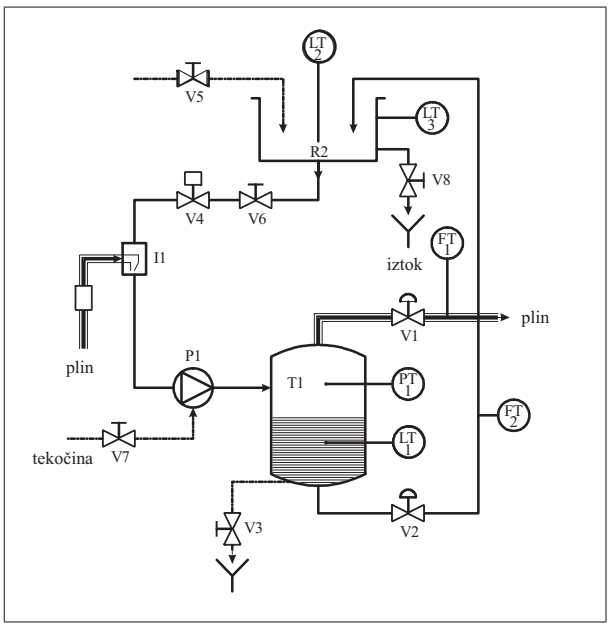
Slika 6.12: Rezultati simulacije

6.3 Primer načrtovanja vodenja enote za pripravo plina

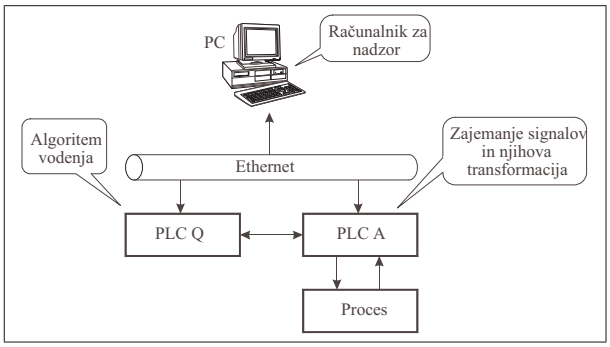
Namen tega primera je prikazati praktično izvedeno vodenje z razvrščanjem ojačenj na način, kot bi bil blizu inženirju, ki običajno načrtuje procesno vodenje. Naloga je bila: načrtovati vodenje tlaka za polindustrijsko procesno napravo za pripravo plina, ki je v procesnem laboratoriju Odseka za sisteme in vodenje na Inštitutu Jožefa Stefana. Njena shema je prikazana na sliki 6.13. Podrobneje je problem opisan v delu [5]. Postavljene zahteve za vodenje so naslednje:

- Obnašanje zaprtozančnega odziva mora biti čim bolj enako po celotnem delovnem področju.
- Vodenje naj bo izvedeno s programirljivimi logičnimi krmilniki (angl. Programmable Logic Controller - PLC) kot tipičnimi predstavniki industrijske strojne opreme za vodenje.
- Izbrani algoritem vodenja mora biti razumljiv inženirju, ki običajno dela s PLC-ji.
- Zaradi enostavnosti naj bo razvrščanje ojačenj izvedeno z zlivanjem proporcionalno integrirnih (PI) regulatorjev in zlivanje s trikotnimi utežnimi (pripadnostnimi) funkcijami.
- Regulacija tlaka v tanku pri različnih nivojih tekočine v njem, tako da bosta hitrost odziva in oblika enaka po celem področju delovanja.
- Rešujemo nelinearni problem, ker se dinamika procesa spreminja s spremembo nivoja tekočine.

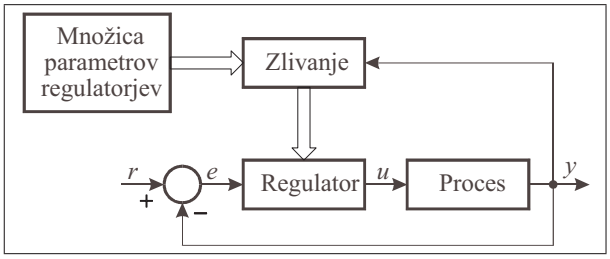
Strukturo vodenja iz izvedbenega stališča prikazuje shema na sliki 6.14, iz sistemskega stališča bločna shema na sliki 6.15, iz izvedbenega stališča pa shema na sliki 6.16.



Slika 6.13: Shema ločevalnika plin-tekočina



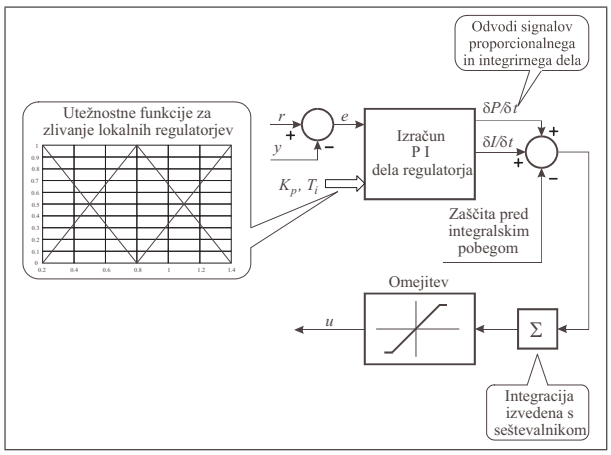
Slika 6.14: Postavitev sistema vodenja za enoto za pripravo plina



Slika 6.15: Blokovna shema zaprtozančnega sistema

Postopek načrtovanja je bil enak, kot smo ga opisali že v prejšnjih podpoglavjih. Nekatere specifičnosti konkretnega načrtovanja lahko opišemo v naslednjih točkah:

- Določili smo primerno število ravnotežnih točk, v katere smo postavili središča utežnostnih funkcij tako, da smo naredili kompromis med dobrim zaprtozančnim odzivom (kar zahteva čim več funkcij) in zaradi enostavne strojne opreme enostavno izvedbo (kar zahteva čim manj funkcij).



Slika 6.16: Izvedbena shema zlivanega regulatorja z razvrščanjem ojačenj za PLC opremo

- Uglaševanje lokalnih regulatorjev v ravnotežnih točkah smo izvedli kar z uglaševanjem po inženirskih pravilih na podlagi odzivov na zelo majhno stopnico s pozitivno in z negativno spremembo amplitude okoli ravnotežne točke.

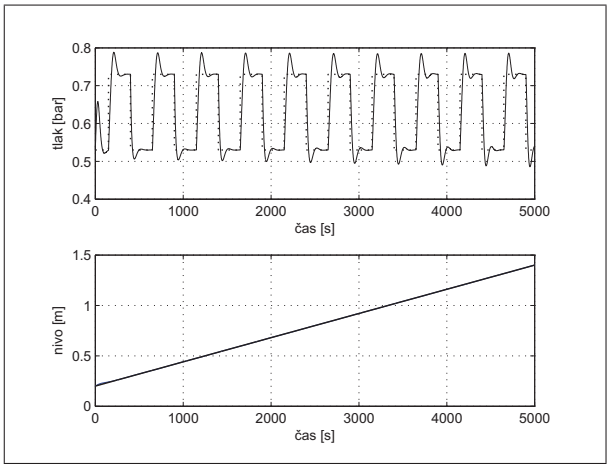
- Celotni postopek načrtovanja je iterativen in interaktiven in ga ponavljamo, dokler ne zadovoljimo postavljenih zahtev. Postopek bi bil lahko bolj premočrten, če bi uporabili matematični model procesa, kar pa bi zahtevalo dodaten vložek v teoretično modeliranje naprave, ki se ga inženirji v praksi nemalokrat ognejo.

- Razvrstilna spremenljivka je bila glede na fizikalno ozadje višina tekočine, ki je počasnejša od zaprtozančnega odziva; to pomeni, da izvedba ni v nobenem primeru kritična. V resnici bi morali delati z vektorjem razvrstilnih spremeljivk, vendar je analiza, ki jo najdemo v [5] pokazala, da lahko vektor razvrstilnih spremeljivk poenostavimo z dominantno spremeljivko, ki je višina tekočine v napravi.

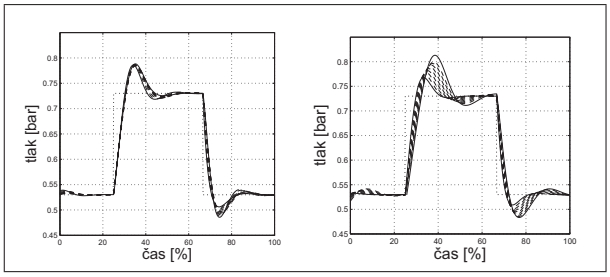
Simulacijski rezultati, ki so prikazani na slikah 6.17 in 6.18, kažejo, da ima zaprtozančni sistem različno obnašanje pri stopnicah navzgor in navzdol zaradi nelinearnosti ventila. Drugače pa lahko ugotovimo, da smo dosegli zadovoljiv zaprtozančni odziv prek celotnega zanimivega področja s tremi zlivanimi regulatorji.

Naslednji korak je bil preizkusiti načrtovani regulator s PLC-jem, priključenim na računalnik, ki je simuliral dinamiko procesa (angl. “Hardware-in-the-loop”- HIL simulation), kar prikazuje shema na sliki 6.19. Rezultati so vidni na sliki 6.20.

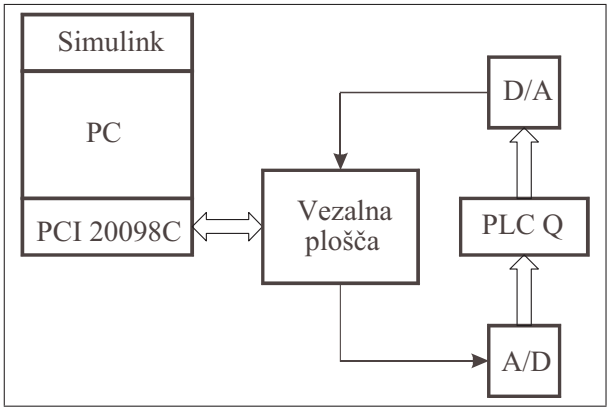
Na koncu sledi še preskus regulatorja na napravi. Na sliki 6.21 vidimo, da regulator zadovoljivo regulira napravo. Razlike med merjenimi in simulacijskimi odzivi pa so vidni



Slika 6.17: Simulacijski rezultati

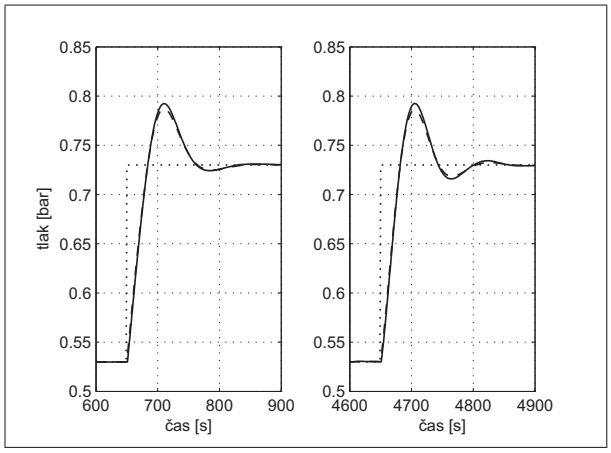


Slika 6.18: Simulacijski rezultati – primerjava normiranih signalov (vsi signali na eni sliki): zaprtozančni odziv z zlivanim regulatorjem (leva slika), zaprtozančni odziv z enim samim PI regulatorjem

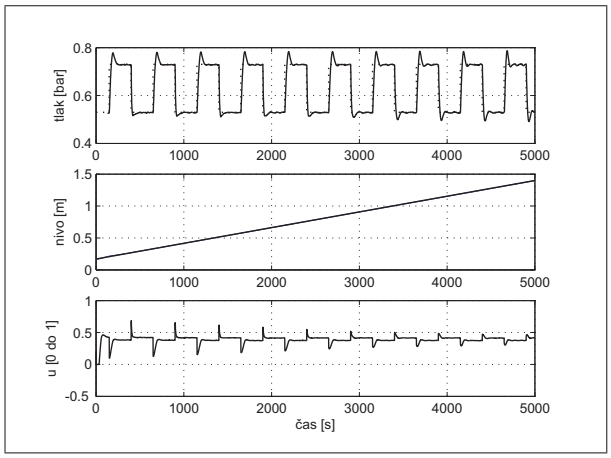


Slika 6.19: Izvedba

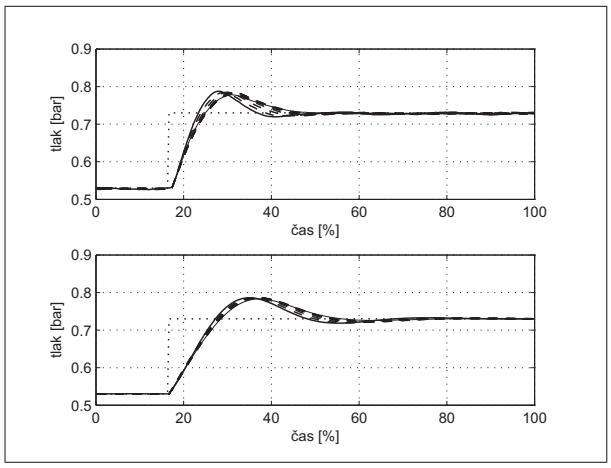
na sliki 6.22. Do razlike v hitrosti odzivov pride zato, ker matematični model, ki smo ga uporabili za simulacijo, ni bil dovolj dobra ponazoritev dinamike enote za pripravo plina. Ne glede na opažena odstopanja lahko ugotovimo, da smo z enostavnim algoritmom in industrijsko strojno opremo uspeli doseči podobno obnašanje po celotnem področju delovanja nelinearnega sistema.



Slika 6.20: Primerjava zaprtozančnih odzivov na robu delovnega področja. Polna črta predstavlja odziv HIL, prekinjena pa odziv računalniške simulacije.



Slika 6.21: Meritve zaprtozančnega odziva, spreminjanje spremenljivke za razvrščanje in regulirni signal



Slika 6.22: Pogled od blizu: primerjava normiranih signalov med meritvami na napravi (zgornja slika) in računalniško simulacijo (spodnja slika)

Literatura

[1] B. Čokan, J. Kocijan (2001): Some realisation issue of fuzzy gain-scheduling controllers : a robotic manipulator case study : [predstavljeno na 6th On-line World Conference on Soft Computing in Industrial Applications, September 10-24, 2001 na svetovnem spletu]. V: Roy, R. (ur.). Soft computing and industry : recent applications. New York: Springer, 2002, 191-199.

[2] J. Kocijan, D. J. Murray-Smith (1999): Robust nonlinear control for ship steering. V: Mastorakis, N. E. (ur.). Progress in simulation, modeling, analysis and synthesis of modern electrical and electronic devices and systems, World Scientific and Engineering Society Press, 235-240.

[3] J. Kocijan, D. Vrančić (2000): Multiple blended controller design for bilinear systems. V: Rudas, I. J. (ur.), Tar, J. K. (ur.). IFAC Symposium on Artificial Intelligence in Real Time Control, AIRTC-2000, October 2-4, 2000, Budapest, Hungary. Preprints. Budapest: Budapest Polytechnic, 187-192.

[4] J. Kocijan, N. Hvala, S. Strmčnik (2000): Multimodel control of wastewater treatment reactor. V: Mastorakis, N. (ur.). System and control : theory and applications, (Electrical and computer engineering series). World Scientific and Engineering Society, 2000, 49-54.

[5] J. Kocijan, G. Žunič, S. Strmčnik, D. Vrančić (2002): Fuzzy gain-scheduling control of a gas-liquid separation plant implemented on a PLC. Int. J. Control, Vol. 75, 1082-1091.

[6] D. J. Leith, W. E. Leithead (1998): Gain-Scheduled & Nonlinear Systems: Dynamic Analysis by Velocity-Based Linearisation Families. Int. J. Control, Vol. 70, 289-317.

[7] D. J. Leith, W. E. Leithead (1998): Gain-Scheduled Controller Design: An Analytic Framework Directly Incorporating Non-Equilibrium Plant Dynamics. Int. J. Control, Vol. 70, 249-269.

[8] D. J. Leith, W. E. Leithead (1999): Analytic framework for blended multiple model systems using linear local models, Int. J. Control, Vol. 72, 605-619.

[9] D. J. Leith, W. E. Leithead (2000): Survey of Gain-Scheduling Analysis and Design, Int. J. Control, Vol. 73, 1001-1025.

[10] D. J. Leith, W. E. Leithead (2007): Velocity-Based Gain-Scheduling Demo, [Elektronski vir], http://www.hamilton.ie/doug/demof.htm

[11] W. J. Rough, J. S. Shamma (2000): Research on gain scheduling, Automatica, Vol. 36, 1401-1425.

7

Identifikacija nelinearnih sistemov z Gaussovimi procesi

7.1 Gaussovi procesi

Gaussovi procesi so naključni procesi. Naključni proces je posplošitev naključne spremenljivke (npr. vektor, skalar) na neki od neodvisnih spremenljivk odvisen prostor. Če je vrednost naključne spremenljivke v vsaki točki tega prostora porazdeljena po Gaussovi (normalni) porazdelitvi, takemu procesu rečemo *Gaussov proces* (GP) [17]. Drugače: če je vhod v proces vektor neodvisnih spremenljivk $\mathbf{x}$, je ta proces Gaussov, če je porazdelitev vrednosti funkcije $f(\mathbf{x})$ za vsak vhodni vektor $\mathbf{x}$ Gaussova.

Model na podlagi Gaussovih procesov (krajše GP model) običajno imenujemo neparametrični verjetnostni model [25]. Namesto za modeliranje bolj običajne omejitve na neki razred (parametriziranih) funkcij s tem načinom *a priori* dopuščamo opis neznanega sistema z neskončno množico funkcij. Pri tem dopuščamo večjo verjetnost funkcij, za katere menimo, da se pri opisu sistema bolj verjetno pojavljajo npr. gladke, stacionarne, periodične.

Podrobneje so osnove modeliranja z modelom na podlagi Gaussovih procesov opisane v [29] ali začetnih poglavjih v [25], bolj podrobno razlago najdemo v [25, 24].

Velja opozoriti, da vhod in izhod v GP model nista signala, marveč je vhod v model vektor z (vzorčenimi) vrednostmi neodvisnih spremenljivk $\mathbf{x}$, izhod iz GP modela pa verjetnostna porazdelitev izhodne vrednosti $f(\mathbf{x})$ pri danem vhodnem vektorju; to izhaja iz Bayesovega načina modeliranja [22], ki predstavlja bistveno razliko od drugih načinov. Poglejmo si podrobnosti tega načina modeliranja.

Za poljubni nabor N vhodnih vektorjev $\mathbf{x}_i$, $i = 1, \ldots, N$ je GP določen z vektorjem srednjih vrednosti $\mathbf{m} = [m_1(\mathbf{x}_1) \ldots m_N(\mathbf{x}_n)]^T$ in kovariančno matriko $\mathbf{K}$,

$$\mathbf{K} = \begin{bmatrix} K_{11} & \ldots & K_{1N} \\ \vdots & \ddots & \vdots \\ K_{N1} & \ldots & K_{NN} \end{bmatrix}, \tag{7.1}$$

kjer so:

$$m_i(\mathbf{x}_i) = \mathbb{E}[f(\mathbf{x}_i)] \tag{7.2}$$

in so elementi kovariančne matrike K_{ij} določeni kot:

$$\begin{aligned} K_{ij} &= \mathrm{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)) \\ &= \mathbb{E}[(f(\mathbf{x}_i) - m(\mathbf{x}_i)(f(\mathbf{x}_j) - m(\mathbf{x}_j)], \end{aligned} \tag{7.3}$$

običajno dobljeni z neko *kovariančno funkcijo* $C(\mathbf{x}_i, \mathbf{x}_j)$:

$$\mathrm{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)) = C(\mathbf{x}_i, \mathbf{x}_j). \tag{7.4}$$

Če je porazdelitev neke množice spremenljivk Gaussova, je Gaussova tudi porazdelitev katere koli podmnožice elementov te množice, kar imenujemo zahteva po konsistenci (angl. consistency requirement). To za delovanje GP modela pomembno lastnost dosežemo, če so elementi kovariančne matrike GP-ja dobljeni s kovariančno funkcijo [25].

Kovariančna funkcija

Vrednost kovariančne funkcije $C(\mathbf{x}_i, \mathbf{x}_j)$ izraža korelacijo med vrednostima izhodov $f(\mathbf{x}_i)$ in $f(\mathbf{x}_j)$ na podlagi vrednosti vhodnih vektorjev $\mathbf{x}_i$ in $\mathbf{x}_j$. Kovariančna funkcija je lahko katere koli oblike, če za poljubni nabor N vhodnih vektorjev $\mathbf{x}_i, i = 1, \ldots, N$ tvori ne-negativno definitno kovariančno matriko $\mathbf{K}$. Kovariančne funkcije so lahko stacionarne, nestacionarne, periodične itd., podrobneje so naštete v [25]. Kovariančne funkcije, ki določa obliko neznane funkcije $f(\mathbf{x})$, navadno ne poznamo vnaprej, lahko pa iz znanja o splošnih lastnostih funkcije $f(\mathbf{x})$ sklepamo na obliko kovariančne funkcije.

Največ se uporablja Gaussova kovariančna funkcija, ki izraža dve pogosti lastnosti procesov:

- gladkost, ki pove, da se bo izhod procesa z majhno spremembo vhoda relativno malo spremenil (korelacija je večja za dva izhoda, čigar vhoda ležita blizu skupaj), in
- stacionarnost, pri kateri je kovarianca med dvema vhodnima vektorjema odvisna samo od njune medsebojne razdalje in ne tudi od njune absolutne lege v prostoru.

Uporabimo jo pogosto, ko nimamo vnaprejšnjega znanja o strukturi procesa. Pri njej je kovarianca med dvema izhodoma $y_i = f(\mathbf{x}_i)$ in $y_j = f(\mathbf{x}_j)$:

$$K_{ij} = C(\mathbf{x}_i, \mathbf{x}_j) = v_1 \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_i^d - x_j^d)^2\right]; \quad (7.5)$$

D je dolžina vhodnega vektorja $\mathbf{x}$ oz. število neodvisnih spremenljivk sistema. Parametri v_1 in w_d, $d = 1, \ldots, D$ so poljubno določljivi parametri kovariančne funkcije. Imenujemo jih *hiperparametri*[1] [24, 21], da poudarimo, da so to parametri sicer neparametričnega modela[2], ki določajo obliko neznane funkcije $f(\mathbf{x})$. Parameter v_1 govori o velikosti kovariance, parametri w_d pa odražajo pomembnost posamezne komponente vhodnega vektorja; večji je parameter w_d, bolj vplivna je sprememba komponente vektorja x^d na vrednost izhoda. Da dana kovariančna funkcija tvori pozitivno definitno kovariančno matriko, morajo biti vsi parametri Gaussove kovariančne funkcije večji od nič.

Modeliranje

Najlaže predstavimo delovanje GP modela na primeru. Vzemimo, da bi radi opisali neki sistem

$$y = f(\mathbf{x}) + v, \quad (7.6)$$

kjer je v beli Gaussov šum z varianco v_0, $v \sim \mathcal{N}(0, v_0)$. Na podlagi N-tih vhodno-izhodnih vzorcev, tj. parov vektorjev $(\mathbf{x}_i, y_i)$, zbranih v množici $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, želimo določiti neznano vrednost izhoda y^* pri vrednostih vhodnega vektorja $\mathbf{x}^*$. V nadaljevanju v kontekstu GP modela $N \times D$ matriko $\mathbf{X}$ in $N \times 1$ vektor $\mathbf{y}$ označimo kot *učno množico*, saj jih uporabljamo za učenje (angl. *training*) GP modela. Posamezni vhodno/izhodni par $(\mathbf{x}_i, y_i)$ iz te množice imenujemo tudi učni vektor oz. učna točka. Par $(\mathbf{x}^*, y^*)$ označimo kot preizkusno oz. testno množico ali tudi kot *testni vhod/izhod*.

Učni izhodi y_i, $i = 1, \ldots, N$ predstavljajo vrednosti naključnih spremenljivk, izhajajočih iz Gaussovega procesa. Predpostavimo, da je izhod sistema gladek in da je sistem stacionaren ter za tvorjenje kovariančne matrike $\mathbf{K}$ uporabimo Gaussovo kovariančno funkcijo (7.5) z neznanimi parametri na začetku.

Dobimo: $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$, kjer so elementi kovariančne matrike $K_{ij} = \Sigma_{ij} + v_0\delta_{ij}$. Σ_{ij} so elementi kovariančne matrike, dobljeni s kovariančno funkcijo (7.5), $v_0\delta_{ij}$ pa opisuje vpliv šuma na izhodu procesa, kjer je δ_{ij} Kroneckerjev operator. Ker smo predpostavili beli šum, so njegove vrednosti korelirane samo same s sabo.

[1] Neal [21] je pokazal, da je vnaprejšnja nevronska mreža z enim skritim nivojem, v katerem je neskončno število nevronov, enaka GP modelu. Hiperparametri določajo distribucijo (sicer neskončnega števila) parametrov te nevronske mreže.

[2] Model je neparametričen, saj za napovedovanje poleg hiperparametrov in kovariančne funkcije potrebujemo še informacijo o obnašanju sistema v obliki vhodno/izhodnih podatkov, uporabljenih pri modeliranju.

Ker je (zaenkrat še neznani) izhod y^* udejanjenje istega procesa kot učni izhodi $\mathbf{y}$, lahko zapišemo [4]: $\mathbf{y}_{N+1} = \begin{bmatrix} \mathbf{y} \\ y^* \end{bmatrix} \sim \mathcal{N}(0, \mathbf{K}_{N+1})$. Skupno kovariančno matriko $\mathbf{K}_{N+1}$ vektorja $\mathbf{y}_{N+1}$ lahko razbijemo:

$$\mathbf{K}_{N+1} = \begin{bmatrix} \begin{bmatrix} \mathbf{K} \end{bmatrix} & \begin{bmatrix} \mathbf{k}(\mathbf{x}) \end{bmatrix} \\ \begin{bmatrix} \mathbf{k}(\mathbf{x})^T \end{bmatrix} & \begin{bmatrix} k(\mathbf{x}) \end{bmatrix} \end{bmatrix}. \quad (7.7)$$

Matrika $\mathbf{K}$ je kovariančna matrika učnih podatkov, $\mathbf{k}(\mathbf{x}^*)$ je vektor kovarianc med učnimi izhodi in testnim izhodom, $k(\mathbf{x}^*)$ pa avtokovarianca testnega izhoda.

Po Bayesovem načinu [22] lahko verjetnostno porazdelitev vrednosti izhoda y^* razdelimo na dva dela: na del, ki določa verjetnost učnih izhodov glede na učne vhode (angl. *marginal part*): $p(\mathbf{y}|\mathbf{X}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$, in na pogojni del (angl. *conditional part*), ki glede na prvi del in vhod $\mathbf{x}^*$ napoveduje verjetnostno porazdelitev izhoda y^*. Formalno zapisano je izračun porazdelitve izhodne verjetnosti odziva y^* [17]:

$$p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}) = \int p(y^*|\mathbf{x}^*, \boldsymbol{\theta}, \mathbf{y}, \mathbf{X}) p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X}) d\boldsymbol{\theta}. \quad (7.8)$$

Običajno je ta integral analitično neizračunljiv, imamo pa na voljo dve alternativi [17]. Prva, bolj pogosta, je aproksimacija integrala z uporabo najbolj verjetnih vrednosti neznanih hiperparametrov $\boldsymbol{\theta}_{\mathrm{MP}}$:

$$p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}) \approx p(y^*|\mathbf{x}^*, \boldsymbol{\theta}_{\mathrm{MP}}, \mathbf{y}, \mathbf{X}). \quad (7.9)$$

Uporabimo tiste vrednosti hiperparametrov $\boldsymbol{\theta}_{\mathrm{MP}}$, pri katerih je verjetnost učnih izhodov $\mathbf{y}$ glede na vrednosti učnih vhodov $\mathbf{X}$ in kovariančno funkcijo $C(.,.)$ največja. Dobimo jih z metodo največje podobnosti (angl. *maximum likelihood method* – ML). Da se izognemo optimizaciji z omejitvami, za optimizacijo uporabimo logaritem porazdelitve učnih podatkov (angl. *log-marginal likelihood*):

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) &= \log(p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})) \\ &= -\frac{1}{2}\log(|\mathbf{K}|) - \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y} - \frac{N}{2}\log(2\pi), \end{aligned} \quad (7.10)$$

kjer je $\boldsymbol{\theta}$ vektor parametrov, $\boldsymbol{\theta} = [w_1 \ldots w_D \; v_1 \; v_0]^T$ in $\mathbf{K}$ kovariančna matrika za učne podatke $\mathcal{D}$. Če je optimizacija izvedena z metodo konjugiranih gradientov (ali katero drugo gradientno metodo), je potreben še izračun odvodov po vseh hiperparametrih:

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \theta_i} = -\frac{1}{2}\mathrm{sled}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta_i}\right) + \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta_i}\mathbf{K}^{-1}\mathbf{y}. \quad (7.11)$$

Ob vsakem koraku optimizacije je potrebno izračunati inverz kovariančne matrike $\mathbf{K}^{-1}$, kar je računsko zahtevno za velike N.

Druga možnost za aproksimacijo integrala (7.8) je numerična integracija nad celotno porazdelitvijo hiperparametrov (MCMC metode, [17]), dobljeno z optimizacijo verjetnosti učnih podatkov (7.10).

Obstaja še ena možnost izbire hiperparametrov modela, tj. navzkrižno vrednotenje (angl. *cross-validation*) [25]. Vrednosti hiperparametrov iščemo kot običajno, le da učne podatke razbijemo na n delov. Za učenje uporabimo $n-1$ delov, za vrednotenje pa tistega, ki ostane. Postopek ponovimo n-krat, vsakič z drugimi podatki za vrednotenje. Ekstremni primer je navzkrižno vrednotenje z izpuščanjem (angl. *leave-one-out - LOO cross-validation*). Največji problem tega postopka je računska zahtevnost, saj moramo naučiti n modelov [25].

Napovedovanje

Skupna porazdelitev $p(\mathbf{y}_{N+1})$ je Gaussova; torej je Gaussova tudi pogojna porazdelitev $p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y}_{N+1})}{p(\mathbf{y}|\mathbf{X})}$. Po poenostavitvi [17, 29] kot napovedan izhod sistema (7.6) dobimo Gaussovo porazdelitev:

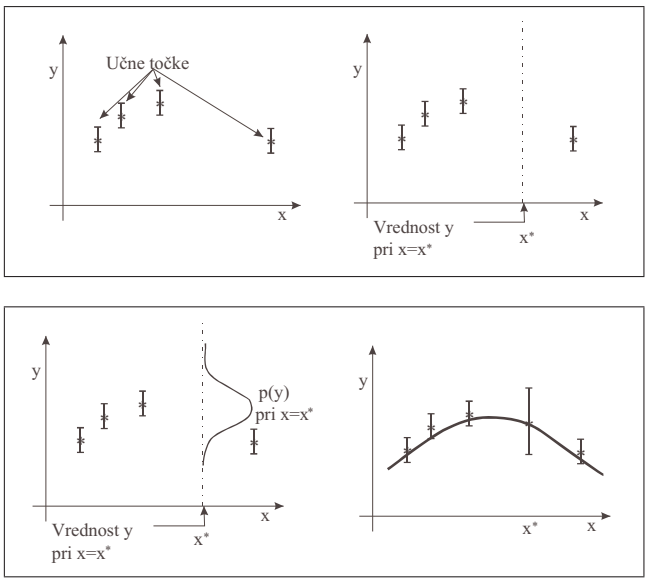
$$p(y^*|\mathbf{x}^*, \mathbf{X}, \mathbf{y}) = \mathcal{N}\left(\mu(\mathbf{x}^*), \sigma^2(\mathbf{x}^*)\right) \tag{7.12}$$

s srednjo vrednostjo $\mu(\mathbf{x}^*)$ in varianco $\sigma^2(\mathbf{x}^*)$:

$$\mu(\mathbf{x}^*) = \mathbf{k}(\mathbf{x}^*)^T\mathbf{K}^{-1}\mathbf{y} \tag{7.13}$$

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}^*) + v_0, \tag{7.14}$$

kjer je $\mathbf{k}(\mathbf{x}^*) = [C(\mathbf{x}_1, \mathbf{x}^*) \ldots C(x_N, \mathbf{x}^*)]$ že omenjeni $N \times 1$ kovariančni vektor med testnim izhodom in učnimi izhodi ter $k(\mathbf{x}^*) = C(\mathbf{x}^*, \mathbf{x}^*)$ avtokovarianca testnega izhoda. Ilustracijo opisanega prikazuje slika 7.1.



Slika 7.1: Princip modeliranja z Gaussovimi procesi

Interpretacija

GP model je sestavljen iz dveh delov:

- iz parov vhodno/izhodnih učnih podatkov (točk) $\mathcal{D}$, ki predstavljajo obnašanje neznanega sistema, in
- kovariančne funkcije $C(.,.)$ z znanimi oz. optimiziranimi hiperparametri $\boldsymbol{\theta}$, ki povedo, v kakšnem razmerju so podatki $\mathcal{D}$.

Ker GP model potrebuje informacijo o neznani funkciji v obliki učnih vhodov in izhodov tudi po učenju, je model neparametričen. Hiperparametri namreč prek kovariančne funkcije samo povedo, kako se učna informacija uporabi za napovedovanje, ni pa v njih spravljena informacija o opisovani funkciji/sistemu.

Na vektor $\mathbf{k}(\mathbf{x}^*)^T\mathbf{K}^{-1}$ v izrazu za srednjo vrednost napovedanega izhoda (7.13) lahko gledamo kot na vektor uteži, ki določa utežitev posameznih učnih izhodov y_i v $\mathbf{y}$ glede na razdaljo med učnimi in testnim vhodnim vektorjem. Ta linearna kombinacija učnih izhodov (angl. *linear predictor*) se lahko razume kot glajenje v GP modelu vsebovane informacije o neznanem sistemu (učni podatki). Še drugače si lahko napoved $\mu(\mathbf{x}^*)$ predstavljamo kot linearno kombinacijo N jedrnih (angl. *kernel*) funkcij, usredišćenih v učnih točkah: $y^* = \sum_{i=1}^{N} \alpha_i\, C(\mathbf{x}^*, \mathbf{x}_i)$. Izhod iz sistema je en vzorec iz dobljene normalne porazdelitve (7.12).

Majhna varianca $\sigma^2(\mathbf{x}^*)$ napovedane porazdelitve izhoda pomeni večje *zaupanje* v napoved in narobe. Če si ogledamo izraz za varianco, vidimo, da je sestavljen iz dveh delov [25]. Od prvega dela $k(\mathbf{x}^*)$, ki predstavlja a priori varianco GP, je odštet izraz $\mathbf{k}(\mathbf{x}^*)^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}^*)$. Ta predstavlja zmanjšanje a priori variance GP pri $\mathbf{x}^*$ zaradi učnih podatkov in se veča z večjo kovarianco med učnimi in testnim vhodom. Preprosteje: »bližje« ko je testni vhod že znanim (učnim) vhodom, večje je zaupanje GP modela v točnost napovedi. Prav varianca, odvisna tudi od lege testnega vhoda glede na učne, je ena izmed glavnih prednosti GP modela pred drugačnimi modeli.

Primerjava z drugimi vrstami modelov

Splošno o lastnostih GP modela

GP model se od večine modelov razlikuje po tem, da je *neparametričen*, saj informacije o neznanem sistemu ne vsebuje samo v (hiper)parametrih kovariančne funkcije, marveč je obnašanje sistema opisano tudi z učnimi podatki $\mathcal{D}$, ki jih model vsebuje.

Dobre lastnosti (prednosti) GP modela so:

- mera negotovosti napovedi izhoda, dana z varianco in odvisna od medsebojne kovariance (lege) učnih vhodnih vektorjev in testnega vhodnega vektorja;
- možnost vključevanja različnih vrst predznanja, npr. linearnih lokalnih modelov, histereze, predznanja o šumu, statične karakteristike itd.;
- relativna enostavnost uporabe;

- majhno število (hiper)parametrov, ki hkrati izražajo tudi vpliv posameznih vhodnih komponent in šuma.

Poleg naštetih prednosti ima GP model tudi slabosti, pri katerih bi poudarili:

- neparametričnost GP modela, ki omejuje možnosti uporabe, in
- veliko računsko zahtevnost učenja, ko je neznani sistem opisan z velikim številom učnih podatkov.

Računsko zahtevnost lahko zmanjšamo na več načinov. Ena možnost je zgoščevanje učne informacije npr. v lokalne modele. Druga možnost so pohitritve, ki nastanejo z uporabo manjše podmnožice učne množice, npr. [26], ali z aproksimacijo inverza kovariančne matrike, ki je računsko najbolj zahtevna [30]. Te in še nekatere druge metode so opisane v [23, 25] in tamkajšnih referencah.

Primerjava

Pojavlja se vprašanje o razlikah med GP modelom in drugimi modeli, dobljenimi z eksperimentalnim modeliranjem, pri modeliranju neznanih funkcij oz. sistemov.

Primerjavo z metodami, pri katerih določimo najprej strukturo z npr. teoretičnim modeliranjem in potem optimiramo parametre s poljubno optimizacijsko metodo, ni povsem ustrezna.

Lažje je GP model primerjati z umetnimi nevronskimi mrežami, ki se uporabljajo za opis nelinearnih relacij. Izkazalo se je, da imajo nevronske mreže poleg problema z določevanjem primerne strukture še dva večja problema:

- netransparentnost (angl. *the lack of transparency*) – struktura nevronske mreže ne odraža strukture neznanega sistema, in
- »prekletstvo dimenzije« (angl. *the curse of dimensionality*), ki se izraža na dva sorodna načina. Z naraščanjem dimenzije vhodnega prostora se namreč povečujeta:
  - potreba po podatkih, ki narašča eksponentno z dimenzijo, in
  - število gradnikov (nevronov) nevronske mreže,

  kar ima za posledico večjo računsko zahtevnost in večje možnosti končanja optimizacije v nekem lokalnem minimumu.

Prednosti GP modela pred nevronskimi mrežami so: dana mera zaupanja v napoved pri GP modelu, boljše delovanje GP modela pri manjšem številu podatkov in zmanjšanje problema »prekletstva dimenzije«. Več o povezavi/primerjavi med GP modelom in nevronskimi mrežami najdemo v [18, 21, 11].

Modeli na podlagi mehke logike in mreže lokalnih modelov zmanjšajo opisane probleme nevronskih mrež, vendar ima v določenih primerih GP model pred njimi potencialne prednosti. Predvsem to velja, kadar imamo opravka z majhnim ali pomanjkljivim številom podatkov za identifikacijo.

Modeli, kot sta mreža radialnih baznih funkcij in metoda podpornih vektorjev (angl. *Support Vector Machines* – SVM), so tesno povezani z GP modelom, saj so tako kot GP model metode, ki uporabljajo jedrne funkcije (angl. *kernel functions*). Glavna prednost GP modela pred njimi je zaupanje v napoved GP modela in uporaba pogojne verjetnosti za določanje parametrov (Bayesov način), slabost pa računska zahtevnost. Metoda relevantnih vektorjev (angl. *Relevance Vector Machines* – RVM) je posebna oblika GP modela [25].

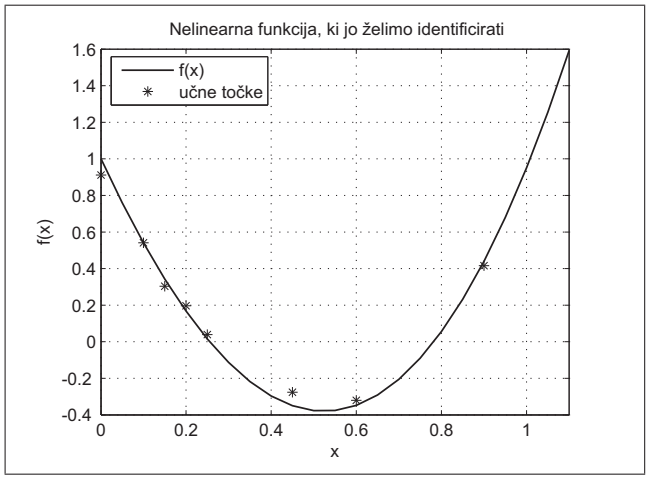
Povezava med GP modeli in še nekaterimi drugimi modeli je podrobneje opisana v [25, 18] ter tamkajšnjih referencah.

Primer identifikacije statične nelinearnosti z GP

Ilustrirajmo uporabo GP modela na primeru. Želimo identificirati nelinearno funkcijo $f(x)$, odvisno od neodvisne spremenljivke x:

$$f(x) = 4x^2 + x - 6\sin(x) + 1 + v \qquad (7.15)$$

na intervalu $x \in [0, 1.2]$. Varianca Gaussovega šuma v na izhodu je $\sigma^2 = 0.0025$. Nelinearna funkcija je predstavljena z osmimi neenakomerno porazdeljenimi učnimi pari (točkami), ki predstavljajo vhodno/izhodno relacijo $x/f(x)$. Funkcijo in učne točke lahko vidimo na sliki 7.2.

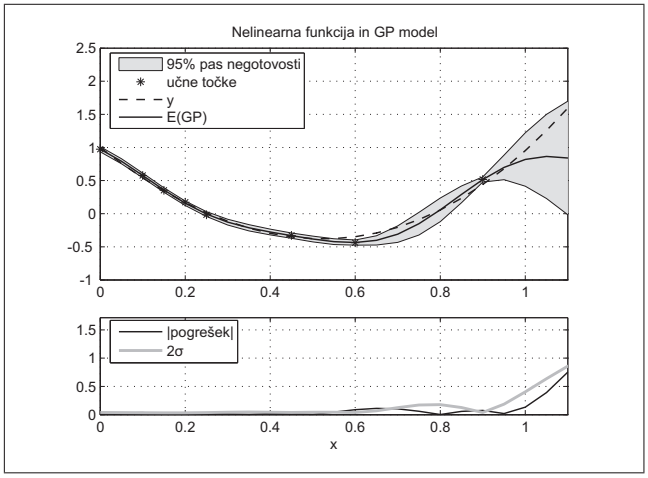


Slika 7.2: Nelinearna funkcija

Za identifikacijo izberemo Gaussovo kovariančno funkcijo (7.5), in to zaradi samo enega vhoda poenostavljeno v:

$$C(x_i, x_j) = v_1 \exp\left[-\frac{1}{2}w(x_i - x_j)^2\right] + v_0\delta_{ij}. \qquad (7.16)$$

Z optimizacijo določimo tri hiperparametre, ki dobijo vrednosti $v_1 = 13.8$, $w = 4.2$ ter $v_0 = 0.0065$. Rezultati identifikacije so prikazani na sliki 7.3. Lahko opazimo, da model slabo opisuje neznano funkcijo na področju, ki ni opisano z učnimi točkami ($x > 1$), prav tako je predikcija slabša na redkeje (to je z malo točkami) opisanih področjih $x > 0.7$. Dobra lastnost GP modela je, da nas na slabše opisano področje opozori povečana varianca (negotovost) na sliki 7.3, kar je vidno predvsem pri $x > 1$. Manj opazna (zaradi manjšega šuma) je druga lastnost GP modela, to je glajenje vsebovane učne informacije, pri kateri model vsebovane (učne) pošumljene vzorce zgladi za napoved novega izhoda.



Slika 7.3: Izhod modela (polna krivulja) z negotovostjo (siva krivulja) in nelinearne funkcije (zvezdice)

7.2 Identifikacija dinamičnih sistemov z GP

Strnjene dobre lastnosti GP modela pri uporabi za identifikacijo dinamičnih sistemov so zbrane v naslednjih točkah:

- Napoved je dana v obliki Gaussove porazdelitve, pri kateri na varianco lahko gledamo kot na mero zaupanja v napoved. Ta je odvisna od kvalitete in lege učnih podatkov v prostoru.[3]
- Število hiperparametrov, ki jih moramo v postopku identifikacije optimirati, je relativno majhno.
- Model je precej robusten, saj deluje relativno dobro tudi z malo podatki za učenje (npr. v neravnotežnem področju).
- Možno je vključevanje različnih vrst predznanja, kot so znanje o statični karakteristiki, lokalnih modelih itd.
- Prekletstvo dimenzije ne narašča eksponentno, ampak s tretjo potenco [24].
- Model ne uporablja razvrstilnih spremenljivk; prav tako ni potrebna delitev operacijskega področja na podpodročja (v primerjavi z npr. mrežami lokalnih modelov).

Seveda ima GP model tudi nekaj omejitev:

- je neparametričen model in kot tak neuporaben, če potrebujemo parametričnega (npr. za nekatere postopke načrtovanja vodenja);
- uporaba modela je računsko zahtevna v primeru velike učne množice, predvsem to velja v fazi optimizacije hiperparametrov.

Kdaj torej uporabiti GP model za identifikacijo dinamičnih sistemov namesto bolj uveljavljenih metod?

- Informacija o sistemu je dana v obliki vhodno/izhodnih podatkov.
- Podatki, ki so na voljo, so slabi, vsebujejo veliko šuma, napake meritev, manjkajoče in neenakomerno razporejene podatke.
- Potrebujemo mero negotovosti v napoved.
- Na voljo imamo relativno majhno (vendar ne premajhno) število podatkov glede na število regresorjev.

Procedura identifikacije

V tem podpoglavju bomo najprej orisali postopek identifikacije dinamičnih sistemov z GP modelom [3]. Pri identifikaciji model dobimo iz meritev, seveda pa si pri tem lahko pomagamo z znanjem o sistemu, če ga imamo. V grobem je identifikacija z GP modelom sestavljena iz naslednjih faz:

1. definicija namena modela;
2. postavitev GP modela;
3. načrtovanje eksperimenta z uporabo a priori znanja ali predhodnih meritev;
4. eksperiment in obdelava signalov, da pridobimo podatke za učenje in vrednotenje;

[3] Dana »mera zaupanja« v napovedan izhod ni izključno domena GP modelov. Kako jo je mogoče dobiti za mehke modele, je opisano npr. v [15]. V splošnem je to lastnost Bayesovega načina modeliranja.

5. učenje GP modela, tj. optimizacija hiperparametrov, in
6. vrednotenje GP modela.

V vsaki od faz se lahko odločimo ali bomo nadaljevali z naslednjo fazo identifikacije ali pa se bomo vrnili na katero izmed prejšnjih. Celoten postopek identifikacije je redko opravljen v enem koraku, po navadi je iterativen. Identifikacija je končana, ko se na podlagi vrednotenja odločimo, da je model zadosti dober za svoj namen.

V nadaljevanju bomo najprej na kratko predstavili posamezne faze identifikacije. Večji del faz identifikacijskega postopka je enak kot pri drugih identifikacijskih postopkih [19, 16], ki smo jih opisali v poglavju 3. Poudarili bomo faze identifikacijskega postopka, ki se zaradi lastnosti GP modela bolj razlikujejo od istih faz pri drugih vrstah modelov.

Definicija namena modela

Namen identifikacije je dobiti model nekega procesa. Ta model ni sam sebi namen, marveč naj bi bil namenjen neki nalogi. Ta naloga določa kriterije, ki jih mora model izpolniti pri vrednotenju, da ga sprejmemo kot zadovoljiv model. Model uporabljamo za simulacijo, napovedovanje odziva, načrtovanje vodenja, analizo ali zaznavanje napak ipd.

Postavitev modela

Naslednja faza v identifikaciji je postavitev modela. Na podlagi apriori znanja o procesu, morebitnih predhodnih meritev ali *rezultatov vrednotenja* postavimo GP model. Postavitev obsega:

- izbiro primerne kovariančne funkcije,
- izbiro regresorjev in
- odločitev o morebitnem vključevanju predznanja.

Pri dinamičnih sistemih z izbiro regresorjev hkrati določimo tudi red modela.

Načrtovanje eksperimenta

Naslednja faza v postopku identifikacije je načrtovanje eksperimenta. Tega načrtujemo na podlagi znanja o procesu, dobljenega iz predznanja ali predhodnih meritev, in glede na postavljeni GP model.

Postopek načrtovanja je soroden kot pri drugih metodah:

1. izbira vhodov in izhodov iz procesa, pri čemer moramo paziti, da merimo vse vplivne veličine;
2. izbira primernega časa vzorčenja;
3. izbira primernega vzbujevalnega signala, da je proces opisan na celotnem področju delovanja, ki nas zanima. Lastnost GP modela je namreč, da model dobro interpolira na področju, kjer imamo učne podatke, in slabo ekstrapolira.

Pri načrtovanju moramo upoštevati omejitve, kot so:

- motnje merjenih signalov;
- omejitev vhodnega signala v proces zaradi fizičnih ali varnostnih razlogov in omejitve izvršnih členov;
- omejen čas, ki je na voljo za izvedbo identifikacijskega postopka (recimo pri merjenjih v industrijskem okolju).

Eksperiment in obdelava podatkov

Eksperiment izvedemo, kot smo ga začrtali v prejšnjem razdelku. Rezultat eksperimenta so izmerjeni vhodni in izhodni signali procesa, ki predstavljajo njegovo obnašanje.

Vhod v GP niso signali, marveč vzorci, ki določajo obnašanje sistema/procesa pri določenih vrednostih regresorjev. Te vzorce dobimo z vzorčenjem vhodnih in izhodnih signalov procesa, pri čemer čas vzorčenja določimo že v fazi načrtovanja eksperimenta.

Primer možne izbire vhodno/izhodnega učnega para (z indeksom i) za proces z vhodnim signalom u in izhodnim signalom y, ki bi ga radi opisali z na primer sistemom drugega reda:

- učni izhod i: izhod iz procesa v času $t = kT$: $y_i = y(k)$;
- ustrezni učni vhod: $\mathbf{x}_i = [y(k-1)\ y(k-2)\ u(k-1)\ u(k-2)]$,

kjer je T čas vzorčenja vhodnih in izhodnih signalov. Testne vhode sestavljamo enako kot učne.

Dobra lastnost GP modela je, da se pomembnost posameznih regresorjev odraža skozi vrednost ustreznih hiperparametrov (angl. *automatic relevance detection*, ARD). Za Gaussovo kovariančno funkcijo (7.5) tako npr. velja, da večja ko je vrednost hiperparametra w_d, bolj pomemben je ustrezni regresor d. To lastnost lahko uporabimo samo, če dobljene učne vzorce pred uporabo za učenje **normaliziramo** tako, da so vrednosti vzorcev po posameznih regresorjih v enakem velikostnem razredu.

V primeru manjkajočih vrednosti regresorjev v posameznem učnem vzorcu lahko te vrednosti, ali nadomestimo npr. z nekim (primernim) povprečjem prejšnjih in poznejših vzorcev (GP model tako ali tako povpreči informacijo) ali pa tak vzorec zavržemo ali zamenjamo z drugim, če imamo na voljo dovolj podatkov.

Učenje modela

Učenje GP modela je enako, naj gre za GP model, ki opisuje statične, ali za GP model, ki opisuje dinamične sisteme. V postopku učenja optimiziramo hiperparametre $\boldsymbol{\theta}$. Ti niso znani vnaprej, marveč jih je potrebno določiti iz učnih podatkov.

Običajno je uporabljena metoda največje podobnosti, saj kljub preprostosti daje dobre rezultate. V splošnem lahko za določitev najbolj verjetnih vrednosti hiperparametrov uporabimo katerokoli optimizacijsko metodo [6].

Zaradi možnosti obtičanja postopka optimizacije v lokalnem minimumu učenje navadno ponovimo večkrat za različne začetne vrednosti hiperparametrov in vrednotimo tako dobljene modele.

Vrednotenje modela

Čeprav je vrednotenje zelo pomemben korak v identifikacijskem postopku, ki pove kako dober je dobljen model, mu je dostikrat namenjeno premalo pozornosti. Z vrednotenjem preverimo ujemanje matematičnega modela in obravnavanega sistema [20].

Kvaliteto modela lahko merimo na več načinov, najbolj pomembni vidiki so:

- verodostojnost modela (angl. *model plausibility*), pri tem nas zanima skladnost modela s predhodnim znanjem;
- (ne)pravilnost modela (angl. *model falseness*), pri čemer nas navadno zanima ujemanje izhoda modela in procesa, ter
- ustreznost modela (angl. *model purposiveness*), pri tem preverimo, ali je model ustrezen za namenjeno nalogo.

Pregled metod vrednotenja najdemo npr. v [20, 9].

Zelo pogosto pri vrednotenju uporabljamo rezultate simulacije ali enokoračne predikcije modela, ki ju primerjamo z obnašanjem procesa, zato bo v naslednjem razdelku ilustrirana še simulacija GP modela.

Simulacija dinamičnih GP modelov

V tem podpoglavju bomo pokazali, kako »običajni« GP model uporabimo za simulacijo.

Zaradi svoje oblike se GP model uporablja kot diskretni vhodno/izhodni dinamični model, ki uporablja diskretne korake, najbolj tipično kot NARX model (angl. *Nonlinear AutoRegressive with eXogenous input model*) [27]. Tako v koraku k kot vhodni vektor v GP model $\mathbf{x}_k$ nastopajo s časom T vzorčene pretekle vrednost vhodov v sistem u in izhodov iz sistema y:

$$\mathbf{x}_k = [y(k-1)\dots y(k-L)\ \ u(k-1)\dots u(k-L)]. \tag{7.17}$$

Na splošno imamo za simulacijo, tj. večkoračno predikcijo, dve možnosti:

- *direktno metodo*, pri kateri naučimo več modelov, za vsak želeni predikcijski horizont posebej, ali
- *iterativno metodo*, pri kateri naredimo model za enokoračno predikcijo, ki jo ponavljamo iterativno.

Problem direktne metode je, da moramo vnaprej izbrati predikcijski horizont, in če ga spremenimo, moramo model učiti znova. Drug problem te metode je, da za močno nelinearne sisteme in velik horizont potrebuje na voljo veliko število podatkov [6].

Nasprotno je model za enokoračno predikcijo mogoče narediti relativno enostavno, z iteracijo pa je mogoče dobiti model za kateri koli predikcijski horizont hočemo [6]. To metodo uporabljamo npr. pri simulaciji sistemov z nevronskimi mrežami. Problem iterativne metode simulacije je akumulacija napake, ko se pomikamo naprej po času. Ena možnost rešitve problema je eliminacija sistematičnih napak, ki nastanejo zaradi zaporednih enokoračnih napovedi [28, 10]. Z uporabo GP modela pa nasprotno dobimo tudi možnost, da napake ne odpravljamo, marveč jo z uporabo variance napovedi raje ovrednotimo [14, 6].

Postopek simulacije

Predpostavimo, da poznamo zgodovino obnašanja dinamičnega sistema reda L do koraka k. Potem v koraku $k+1$ poznamo celoten vhodni vektor GP modela:

$$\mathbf{x}_{k+1} = [y(k)\dots y(k-L+1)\ \ u(k)\dots u(k-L+1)]. \tag{7.18}$$

V koraku $k+2$ kot nov vhodni vektor v GP model nastopa:

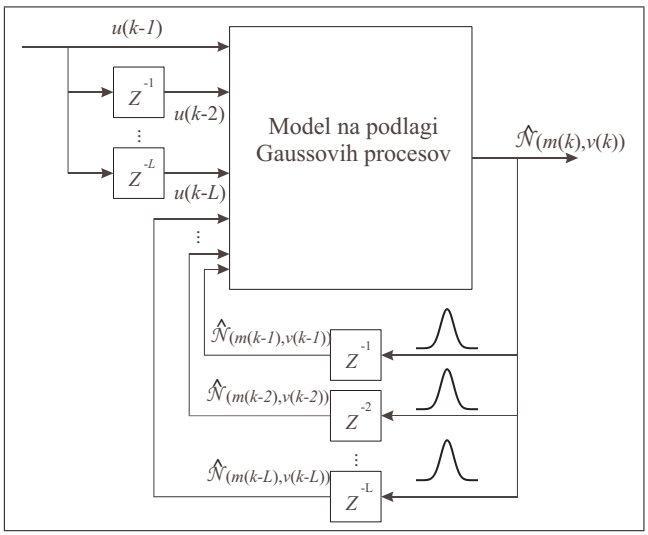
$$\begin{aligned}\mathbf{x}_{k+2} \quad = \quad & [y(k+1)\ y(k)\dots y(k-L+2)\ \ u(k+1)\dots \\ & u(k-L+2)]. \end{aligned} \tag{7.19}$$

Tu nastopi problem, saj ne poznamo vrednosti izhoda v času $k+1$. Postopamo podobno kot pri nevronskih mrežah

– namesto prave vrednosti izhoda iz sistema $y(k+1)$ uporabimo njen približek $\hat{y}(k+1) = f(\mathbf{x}_k)$, izračunan z GP modelom. Enako postopamo v vseh nadaljnjih korakih.

Imamo dve možnosti simulacije:

- *simulacija brez propagacije negotovosti* ali »naivna« metoda, pri kateri podobno kot pri nevronskih mrežah za vrednost izhoda uporabimo samo srednjo vrednost napovedi, in
- *simulacija s propagacijo negotovosti* ali »eksaktna« metoda, pri kateri uporabimo celotno napovedano porazdelitev. Postopek je podrobno opisan v [6], skrajšana inačica pa je v [14, 5] in [1]. Tudi pri tej ne moremo izračunati natančnih porazdelitev na izhodu modela, zato se moramo zopet zateči k aproksimaciji. Imamo dve možnosti aproksimacije [6, 18]:
  1. deterministična aproksimacija (analitična aproksimacija integrala), pri kateri izhodno *ne-Gaussovo porazdelitev* aproksimiramo z Gaussovo z enako srednjo vrednostjo in varianco (slika 7.4), in
  2. numerične Monte-Carlo (MCMC) metode.



Slika 7.4: Simulacijska shema za identificirani GP model dinamičnega procesa

Kdaj torej uporabiti prvo ali drugo metodo?

Metoda s propagacijo negotovosti daje natančnejšo mero zaupanja v napoved, a je hkrati računsko bolj zahtevna tako za analitično kot za numerično propagacijo. Na drugi strani je metoda brez propagacije negotovosti hitra, enostavna in še vedno daje neko informativno mero zaupanja. Ta je sicer preoptimistična, saj se akumulirana napaka iz preteklih korakov ne upošteva pri napovedi.

Propagacija variance ne daje samo natančnejšega napovedovanja izhoda, marveč spremeni tudi srednjo vrednost napovedane porazdelitve. Odvisno od nelinearnosti sistema so te razlike lahko večje ali manjše. Ker gre tako pri »naivni« kot pri »eksaktni« simulaciji za približek, lahko samo domnevamo, za koliko so rezultati slednje natančnejši. Odstopanje modela pa moramo že tako ali tako preveriti z vrednotenjem, ki pokaže kakovost napovedovanja modela.

»Naivno« metodo torej uporabimo, ko nas zanima hitrost in enostavnost, »eksaktno« pa, ko nas zanima natančnejša mera zaupanja v napoved in čas trajanja simulacije ni preveč omejen.

Primer identifikacije dinamičnega procesa

Opisano metodo ilustrirajmo z identifikacijo nelinearnega dinamičnega sistema prvega reda (3.10), ki ga bomo modelirali kot model prvega reda:

$$y(k+1) = f(y(k), u(k)). \tag{7.20}$$

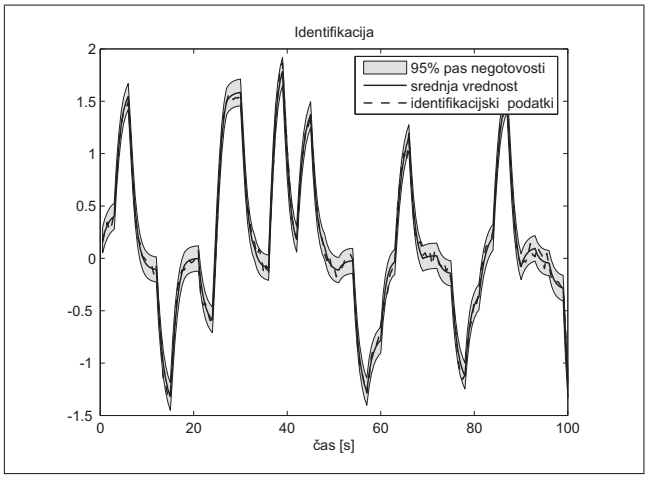
Funkcija f je GP, in sicer imamo dvodimenzionalni regresijski model $D = 2$. To pomeni, da identificiramo hiperparametre: v_0, v_1, w_1, w_2.

Vhodni in izhodni signali

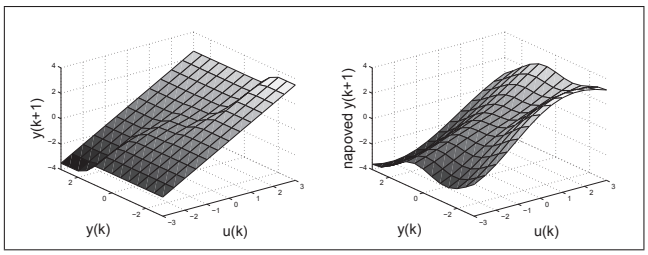
Poglavitne značilnosti izbranih vhodnih in izhodnih signalov so naslednje:

- Vhodni signal je definiran na področju [-1.3, 1.3].
- Izbrani čas vzorčenja glede na dinamiko sistema je 0.5 s.
- Vhodni signal za identifikacijo (enak kot v poglavju 3):
  - tvorjen z generatorjem naključnih števil,
  - število podatkov določa dimenzijo kovariančne matrike,
  - velikost kovariančne matrike določa računsko obremenitev.
- Vhodni signal za vrednotenje (enak kot v poglavju 3):
  - tvorjen z generatorjem naključnih števil, vendar z drugim vzorčenjem in amplitudo na področju [-1.2,1.2],
  - drugi signal za vrednotenje z amplitudo zunaj identificiranega področja, in sicer [-1.3,1.5].

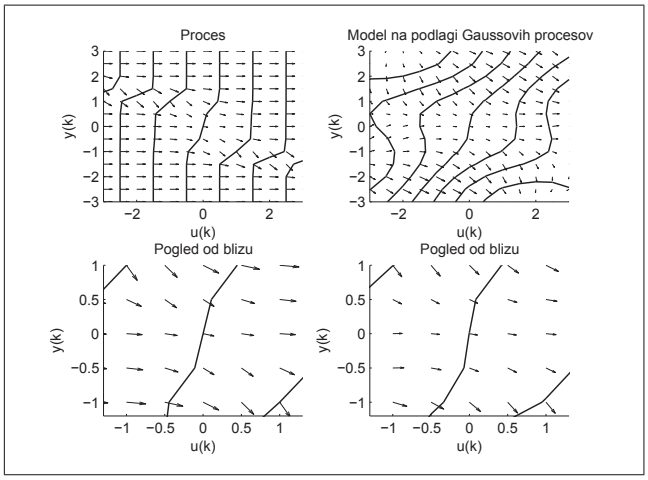
Odziv na vhodni signal za identifikacijo prikazuje slika 7.5, primerjava modela z originalom pa je prikazana na slikah 7.6 in 7.7 .



Slika 7.5: Odziv na signal za identifikacijo



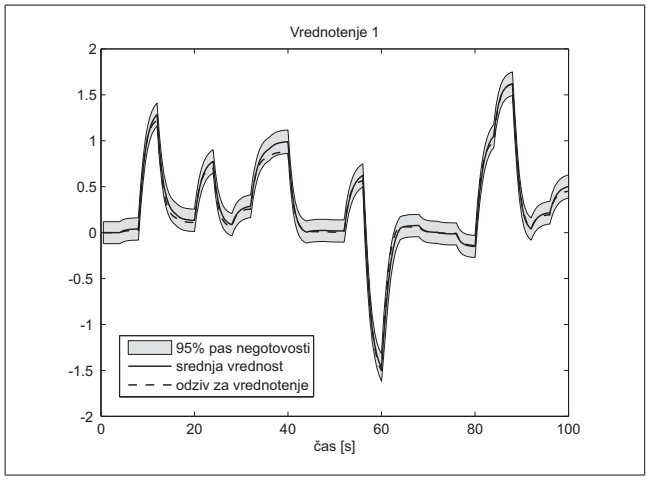
Slika 7.6: Nelinearnost sistema y(k+1)=f(u(k),y(k)) (leva slika) in GP modela (desna slika)



Slika 7.7: Primerjava izohips in gradientov

Vrednotenje prikazujejo slike 7.8, 7.9, 7.10 in 7.11.

Odzive lahko kvantitativno vrednotimo z različnimi cenilkami. Zelo pogosti sta povprečna vrednost absolutne napake in še posebej povprečna vrednost kvadrata napake, ki je posebej primerna za vse metode, pri katerih optimiramo z metodo najmanjših kvadratov. Primer cenilke, ki upošteva celotno porazdelitev, je logaritem napovedane gostote napake (angl. log-predictive density error, LD), ki bolj uteži napako tistih napovedi, pri katerih model z



Slika 7.8: Odziv na prvi signal za vrednotenje

manjšo varianco izkazuje večje zaupanje v napoved.

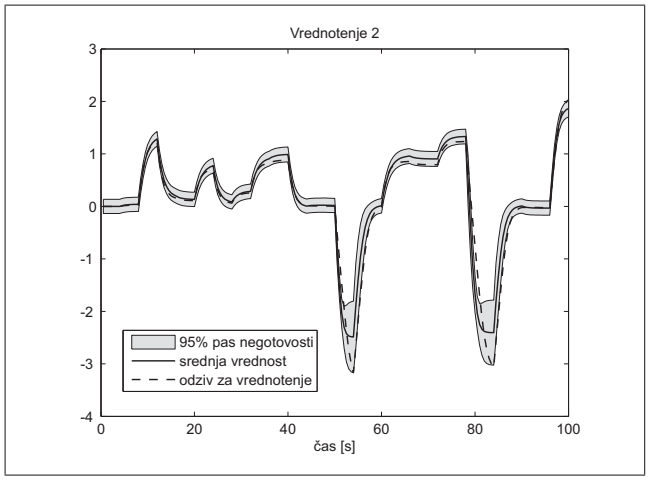
Omenjene cenilke so zbrane v naslednjih enačbah in rezultatih za naš primer:

$$AE = \frac{i=1}{N} \sum |\hat{y}_i - y_i| = 0.028,$$

$$SE = \frac{i=1}{N} \sum (\hat{y}_i - y_i)^2 = 0.0016,$$

$$LD = \frac{1}{2N} \sum_i (\log(2\pi) + \log(\sigma_i^2) + \frac{(\hat{y}_i - y_i)^2}{\sigma_i^2}) = -1.3842,$$

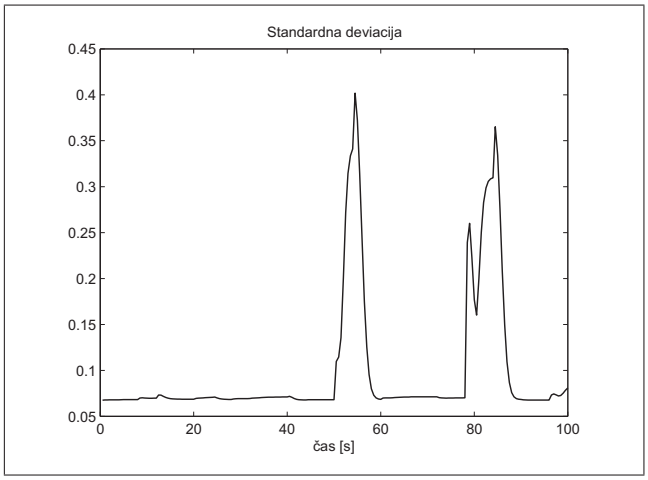
kjer so:
AE – povprečna vrednost absolutne napake
SE – povprečna vrednost kvadrata napake
LD – logaritem gostote napake



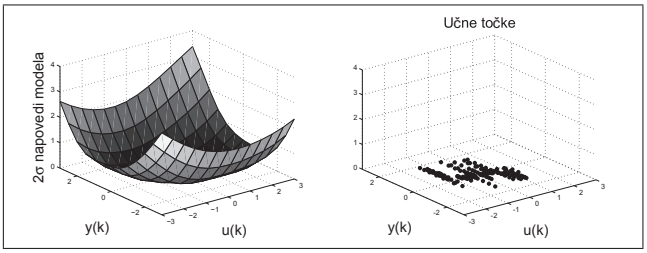
Slika 7.9: Odziv na drugi signal za vrednotenje

7.3 Primer identifikacije pH procesa

Oglejmo si še primer identifikacije pH procesa (2.20) [11], ki ga bomo modelirali z modelom višjega reda. Upora-



Slika 7.10: Standardna deviacija odziva na drugi signal za vrednotenje



Slika 7.11: Površina negotovosti (leva slika) $y(k+1) = f(u(k), y(k))$ za GP model in položaj učnih podatkov (desna slika)

bili smo ga že za prikaz modeliranja z nevronsko mrežo v poglavju 3.

Identifikacija

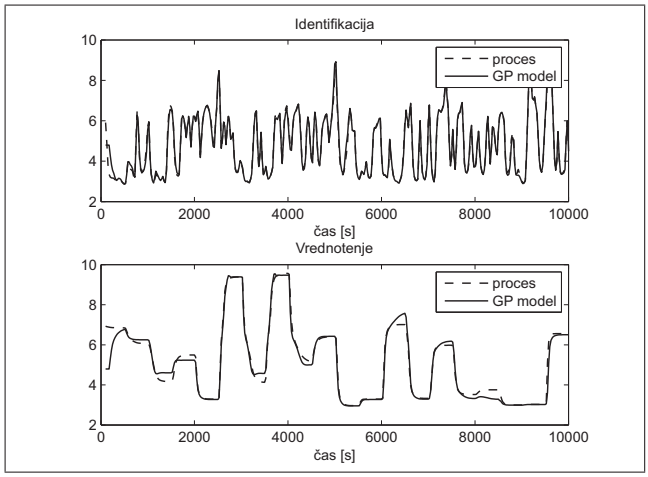
Z iterativnim postopkom smo izbrali naslednji model:

- Kovariančna funkcija:

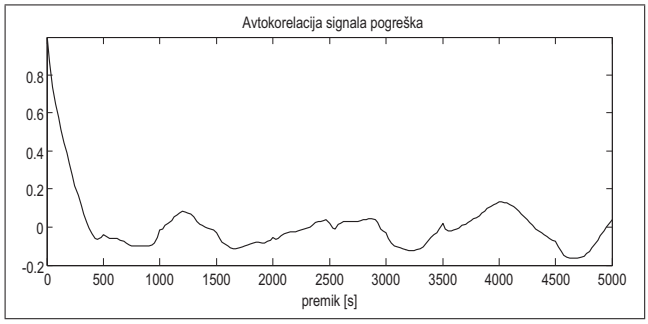
$$C(\mathbf{x}^p, \mathbf{x}^q) \;=\; v_1 \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_d^p - x_d^q)^2\right].$$

- Regresorji: $y(k-1), ..., y(k-4), u(k-1), ..., u(k-4)$, kar pomeni deset hiperparametrov za optimizacijo.
- Iz tega sledi večja računska obremenitev (po izkušnji približno 110-krat večja kot pri sistemu 1. reda).
- Uporabljena optimizacijska metoda: Polak-Ribierova metoda konjugiranih gradientov.

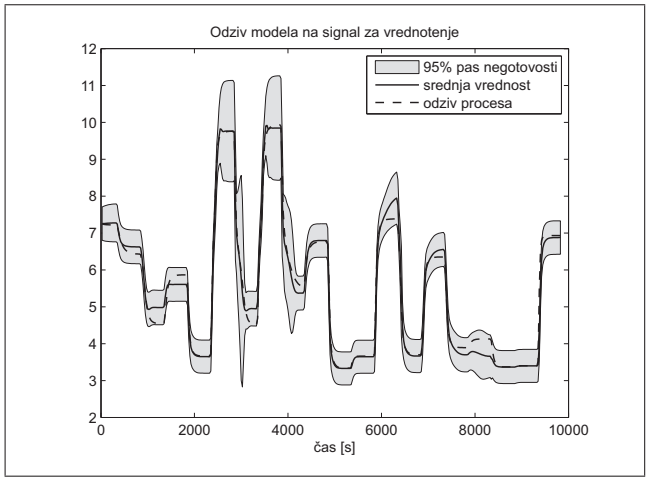
Vrednotenje modela prikazujejo slike 7.12 do 7.14. Vrednosti kriterijskih funkcij za vrednotenje sta $AE = 0.1494$, $SE = 0.0512$ in $LD = 27.87$.



Slika 7.12: Odziv procesa in srednja vrednost napovedi modela



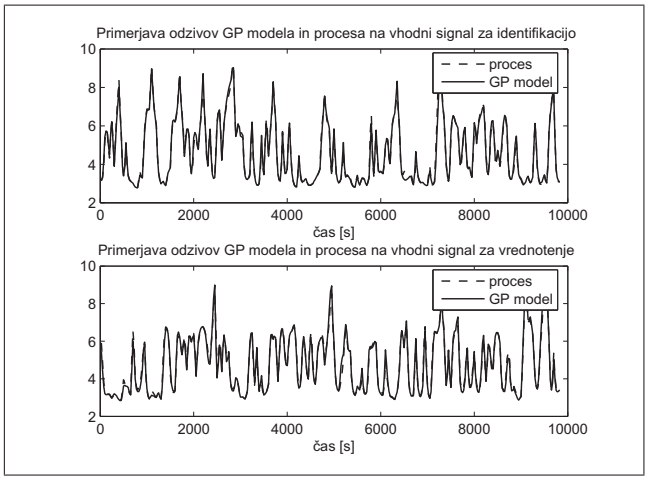
Slika 7.13: Avtokorelacija pogreška modela



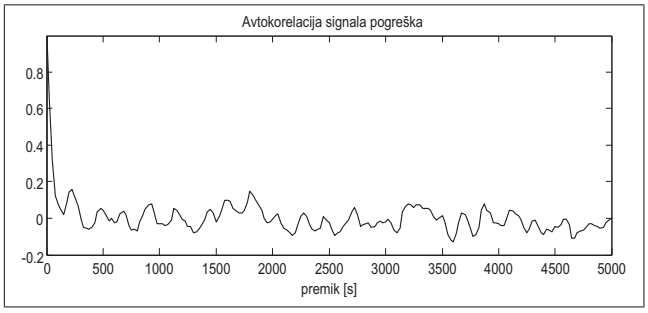
Slika 7.14: Odziv na signal za vrednotenje skupaj s pasom negotovosti

Na sliki 7.14 lahko vidimo področja z večjo varianco, ki pomenijo slabšo kvaliteto predikcije. To je posledica tega, da smo sistem vrednotili s signalom, ki se dinamično precej razlikuje od signala, ki smo ga uporabili za identifikacijo. Pomeni, da se pri vrednotenju model odziva zunaj ali na robu področja, kjer je bil identificiran. Zato sistem vre-

dnotimo še s signalom, ki vzbuja sistem v področju, kjer je bil proces identificiran (slike 7.15 in 7.16), pri čemer dobimo boljše rezultate vrednotenja.



Slika 7.15: Vrednotenje z drugim vhodnim signalom



Slika 7.16: Avtokorelacija pogreška modela

7.4 Načrtovanje vodenja

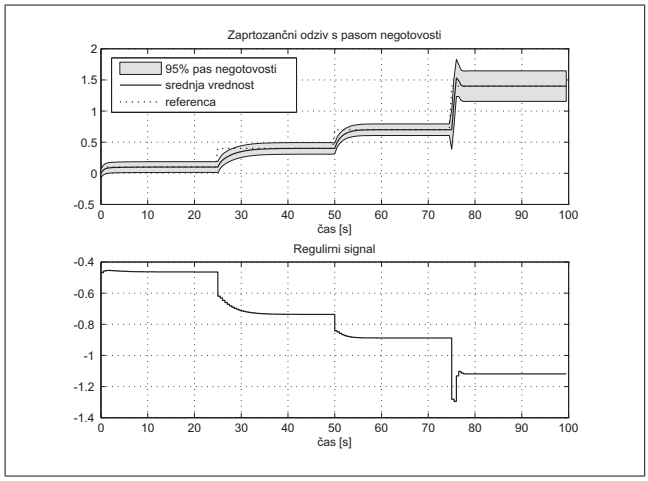
GP model je neparametrični verjetnostni model. Metode načrtovanja vodenja, ki bi lahko uporabljale tovrstni model, so bolj ali manj iste kot metode, opisane pri uporabi umetnih nevronskih mrež za načrtovanje vodenja. Ilustrirali bomo primer prediktivnega vodenja z uporabo GP modela na procesu prvega reda (3.10) [12] in na pH procesu [13] na enak način, kot smo to naredili v poglavju 4.

Uporabili smo prediktivno funkcijsko vodenje s kriterijsko funkcijo

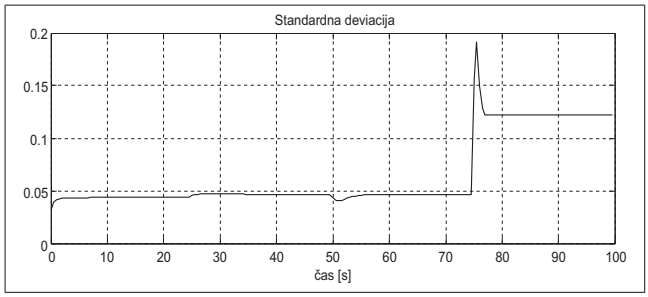
$$J = \min_{\mathbf{U}(k)} \left[r(k+P) - \hat{y}(k+P) \right]^2 .$$

Uporabljeni predikcijski horizont je dolg osem vzorcev, regulirni horizont pa en vzorec. Pri optimizaciji regulirnega signala bi lahko upoštevali omejitve amplitud in hitrosti na vhodu, izhodu in stanjih, vendar smo upoštevali le omejitev variance var $\hat{y}(k+P) \leq k_v$, ki implicitno upošteva tudi druge omejitve, saj tam, kjer je proces omejen, ne moremo identificirati dobrega modela oziroma ga sploh ne moremo identificirati, na kar kaže velika vrednost variance.

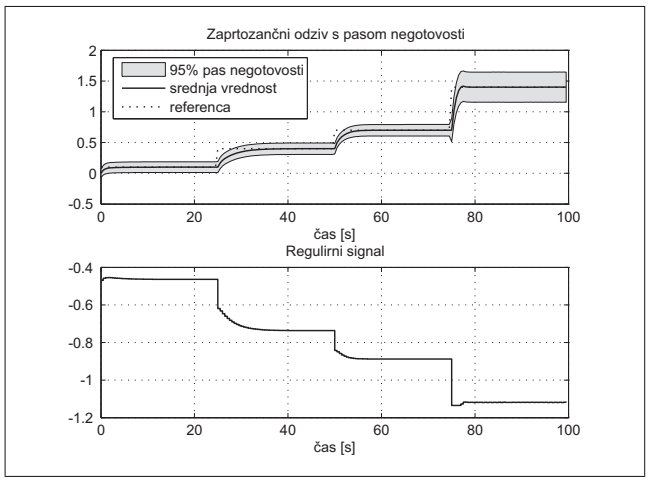
Rezultate sledilnega vodenja brez upoštevanja omejitve prikazujeta sliki 7.17 in 7.18, z omejitvijo pa sliki 7.19 in 7.20.



Slika 7.17: Rezultati vodenja – primer brez omejitev

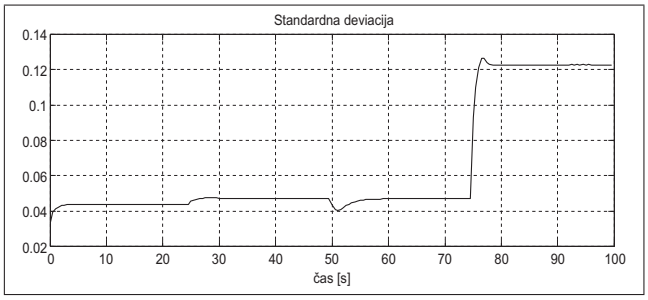


Slika 7.18: Standardna deviacija



Slika 7.19: Rezultati vodenja – primer z omejitvijo variance na vrednost 0.13^2

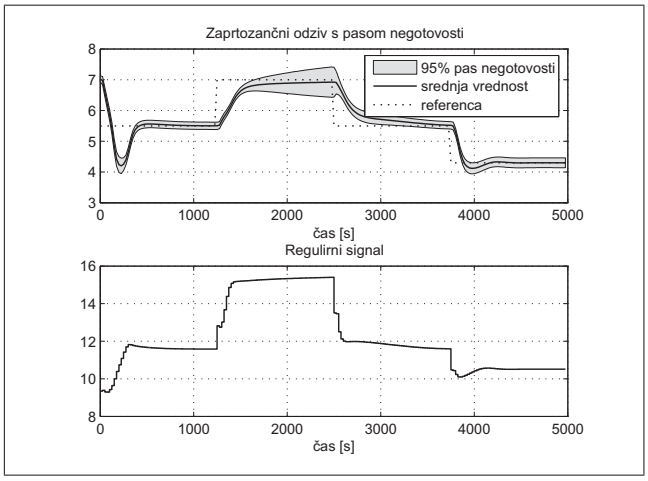
Popolnoma enak princip vodenja smo uporabili tudi na pH procesu. Rezultate sledilnega vodenja brez upoštevanja



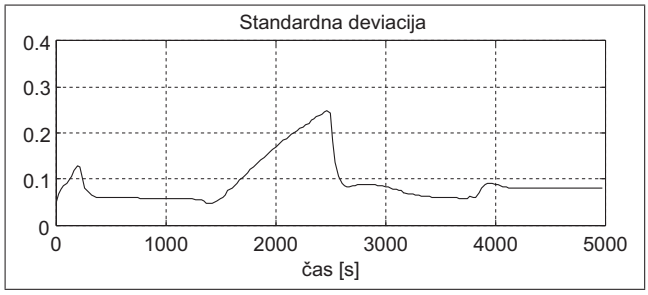
Slika 7.20: Standardna deviacija

omejitve prikazujeta sliki 7.21 in 7.22, z omejitvijo pa sliki 7.23 in 7.24.

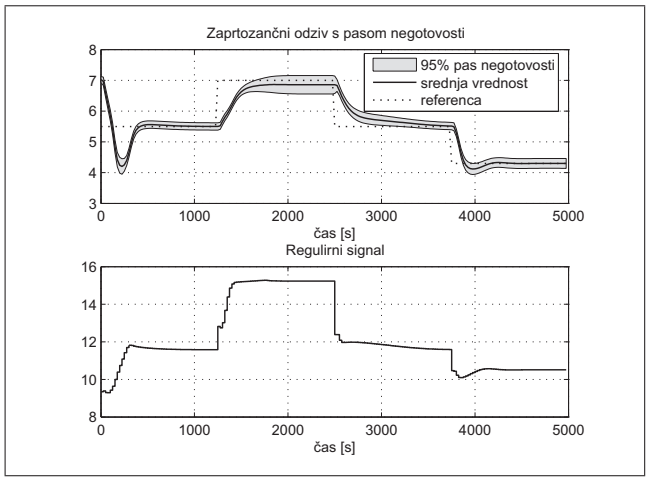
Enostaven sklep, ki ga lahko naredimo na podlagi prikazanih rezultatov, je, da prediktivno vodenje lahko s pridom izkorišča informacijo o negotovosti modela. Prediktivni regulator proces ne pusti v področje, kjer je deviacija modela večja od predpisane vrednosti, in na ta način vzdržuje varnost vodenja.



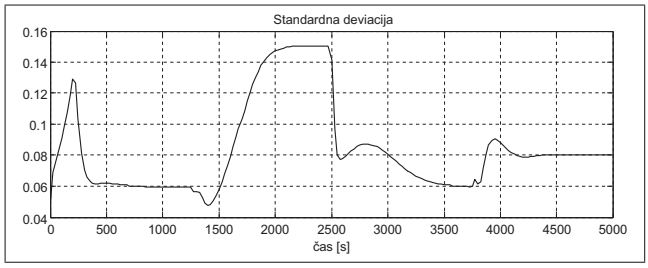
Slika 7.21: pH proces: Rezultati vodenja – primer brez omejitev



Slika 7.22: Standardna deviacija



Slika 7.23: pH proces: Rezultati vodenja – primer z omejitvijo variance na vrednost 0.15^2



Slika 7.24: Standardna deviacija

Literatura

[1] K. Ažman (2004): Identifikacija dinamičnih sistemov z Gaussovimi procesi z vključenimi lokalnimi modeli, magistrsko delo, Univerza v Ljubljani, Ljubljana.

[2] K. Ažman, J. Kocijan (2005): An example of Gaussian process model identification, L. Budin, S. Ribarić, (ur.), Proceedings of 28th International conference MIPRO, CIS – Inteligent Systems, Opatija, 79-84.

[3] K. Ažman, J. Kocijan (2006): Gaussian process model validation: biotechnological case studies, I. Troch, F. Breitenecker, (ur.) Proceedings of the 5th Vienna Symposium on Mathematical Modeling – MathMod, Dunaj.

[4] M. N. Gibbs (1997): Bayesian Gaussian processes for regression and classification, doktorska disertacija, Cambridge University, Cambridge.

[5] A. Girard, C.E. Rasmussen, R. Murray-Smith (2002): Gaussian process priors with uncertain inputs: multiple-step-ahead prediction, delovno poročilo DCS TR-2002-119, University of Glasgow, Glasgow.

[6] A. Girard (2004): Approximate methods for propagation of uncertainty with Gaussian process models, doktorska disertacija, University of Glasgow, Glasgow.

[7] G. Gregorčič, G. Lightbody (2002): Gaussian processes for modelling of dynamic non-linear systems, Proceedings in Irish Signals and Systems Conference, Cork.

[8] M. A. Henson and D. E. Seborg (1994): Adaptive Nonlinear Control of a pH Neutralization Process, IEEE Trans. Control System Technology, Vol. 2, No. 3, 169-183.

[9] N. Hvala, S. Strmčnik, D. Šel, S. Milanič, B. Banko (2005): Influence of model validation on proper selection of process models — an industrial case study, Computers and Chemical Engineering, Vol. 29, 1507-1522.

[10] K. Judd, M. Small (2000): Towards long-term prediction, *Physica D*, Vol. 136, No. 1-2, 31-44.

[11] J. Kocijan, B. Banko, B. Likar, A. Girard, R. Murray-Smith, C.E. Rasmussen (2003): A case based comparison of identification with neural networks and Gaussian process models, Proceedings in IFAC ICONS conference, Faro, 137-142.

[12] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, B. Likar (2003): Predictive control with Gaussian process models. V: B. Zajc (ur.), M. Tkalčič (ur.). Proceedings the IEEE Region 8 EUROCON 2003: computer as a tool, Vol. A, Ljubljana, 352-356.

[13] J. Kocijan, R. Murray-Smith (2005): Nonlinear predictive control with a Gaussian process model. V: R. Murray-Smith (ur.), R. Shorten(ur.). Switching and learning in feedback systems, Lecture notes in computer science, Vol. 3355, Springer, Heidelberg, 185-200.

[14] J. Kocijan, A. Girard, B. Banko, R. Murray-Smith (2005): Dynamic systems identification with Gaussian processes, Mathematical and Computer Modelling of Dynamic Systems, Vol. 11, No. 4, 411-424.

[15] I. Škrjanc, S. Blažič, O. Agamenonni (2005): Identification of dynamical systems with a robust interval fuzzy model, Automatica, Vol. 41, 327-332.

[16] L. Ljung (1999): System identification – theory for the user, Prentice Hall, New Jersey, druga izdaja.

[17] D. J. C. MacKay (1998): Introduction to Gaussian processes, C. M. Bishop (ur.), Neural networks and machine learning, NATO ASI Series – F 168, Springer-Verlag, Berlin, 133-166.

[18] D. J. C. MacKay (2003): Information theory, inference and learning algorithms, poglavje Gaussian Processes, Cambridge University Press, Cambridge, 535-548.

[19] D. Matko (1998): Identifikacije, Založba Fakultete za elektrotehniko, Ljubljana, druga izdaja.

[20] D. J. Murray-Smith (1998): Methods for the external validation of continuous system simulation models: a review, Mathematical and Computer Modelling of Dynamical Systems, Vol. 4, 5-31.

[21] R. M. Neal (1996): Bayesian learning for neural networks, Lecture Notes in Statistics, Vol. 118, Springer-Verlag, New York.

[22] V. Peterka (1981): Bayesian system identification, In Trends and Progress in System Identification, P. Eykhoff (ur.), Pergamon Press, Oxford, 239-304.

[23] J. Quiñonero-Candela, C. E. Rasmussen (2005): Analysis of some methods for reduced rank Gaussian process regression, R. Murray-Smith, (ur.), R. Shorten, (ur.), Switching and Learning in Feedback Systems,

Lecture Notes in Computer Science, Vol. 3355, Springer, Heidelberg, 98-127.

[24] C. E. Rasmussen (1996): Evaluation of Gaussian processes and other methods for nonlinear regresion, doktorska disertacija, University of Toronto.

[25] C. E. Rasmussen, C. K. I. Williams (2006): Gaussian Processes for machine learning, The MIT Press, Cambridge, MA.

[26] M. Seeger, C. K. I. Williams, N. D. Lawrence (2003): Fast forward selection to speed up sparse Gaussian process regression, C.M. Bishop, (ur.), B.J. Frey, (ur.), Proceedings of the Ninth International Workshop on AI and Statistics, Key West, FL.

[27] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, A. Juditsky (1995): Nonlinear black-box modeling in system identification: a unified overwiev, Automatica, Vol. 31, No. 12, 1691-1724.

[28] M. Small, K. Judd (1999): Variable prediction steps and long term prediction, delovno poročilo, University of Western Australia, Dept. of Mathematics and Statistics.

[29] C. K. I. Williams (1998): Prediction with Gaussian processes: from linear regression and beyond, M. I. Jordan, (ur.), Learning in graphical models, Vol. 89 iz *Nato Science Series D*, Springer, Berlin, 599-621.

[30] C. K. I. Williams, M. Seeger (2001): Using the Nyström method to speed up kernel machines, T. K. Leen, (ur.), T. G. Diettrich, (ur.),V. Tresp, (ur.), Proceedings of Advances in Neural Information Processing Systems conference, Vol. 13, The MIT Press, MA, 682-688.

Stvarno kazalo

Adaline, 2
adaptivno vodenje, 41, 52
aktivacijska funkcija, 2, 6

Boltzmanovi stroji, 3

delta pravilo, 3
dinamični sistem, 14

eksprimentalni model, 13
enokoračna napoved, 31, 79

Fourierova analiza, 19

Gaussovi procesi, 73
GP model, 73, 75, 77

hitrostna linearizacija, 54–56, 63
Hopfieldova nevronska mreža, 3

identifikacija, 13, 14

Kohonenova mreža, 3

linearni model, 14

mehki modeli, 28, 29
metoda najmanjših kvadratov s posplošenim pogreškom (ARX), 15
metoda najmanjših kvadratov z modelom šuma (ARMAX), 15
metoda z izhodnim pogreškom (OE), 15
mreža radialnih baznih funkcij, 7, 8
mreže lokalnih modelov, 51
mreže radialnih baznih funkcij, 28, 51

nelinearne preslikave, 28
nelinearni sistemi, 27
nevronske mreže, 1, 5, 28, 41

perceptron, 2
prediktivno funkcijsko vodenje, 47, 83
prediktivno vodenje, 44, 83

radialne bazne funkcije, 29
regresorji, 14, 28

samoorganizirane nevronske mreže, 3
simulacija, 31, 79
simulirano ohlajanje, 3
slemenaste bazne funkcije, 29
statični sistem, 14

Takensov teorem, 29
teoretični model, 13

univerzalni aproksimator, 5

večmodelni sistemi, 51
večnivojski perceptron, 5, 6, 51
vodenje z razvrščanjem ojačenj, 63, 64, 66
vzvratno učenje, 2

zlivani večmodelni sistemi, 58