

Univerza v Ljubljani
Fakulteta za elektrotehniko

Dejan Dovžan

**REKURZIVNA MEHKA IDENTIFIKACIJA V
VODENJU IN NADZORU PROCESOV**

DOKTORSKA DISERTACIJA

Mentor: prof. dr. Igor Škrjanc

V Ljubljani, 2013

Zahvala

Doktorska Disertacija predstavlja več kot le neko zaključno poročilo raziskav, ki jih je raziskovalec opravil. Je delo, ki označuje konec nekega obdobja. Zame je bilo to eno lepših obdobj v življenju. Zato bi se na tem mestu rad zahvalil vodjem laboratorija LAMS in LMSV ter mojim sodelavcem za ustvarjanje prijetne delovne klime ter za pomoč in nasvete pri raziskavah. Dvomim, da bi si človek lahko izbral boljši kolektiv, kot je kolektiv katerega del sem bil tudi sam. Dečki - "kapo dol". Bilo mi je v veselje delati z vami.

Hvala tudi mentorju prof. dr. Igorju Škrjancu za koristne nasvete in usmerjanje pri raziskavah, ter nenazadnje za zagotavljanje možnosti vključevanja v različne projekte, zaradi katerih je bilo moje delo zanimivo in dinamično.

Seveda pa se moram zahvaliti tudi moji celotni družini še posebej staršem, Mojci in Janezu, ki sta me podpirala pri študiju in na drugih področjih. Po svojih najboljših močeh sta se trudila, da bi mi olajšala ustvarjanje mojega samostojnega življenja in nove družine, ter me s tem obvarovala pred stresom in marsikaterimi težavami, s katerimi se danes srečujejo mladi. Za to sem jima iskreno hvaležen.

Zahvala gre tudi ženi Tamari za njeno podporo, ker je vedno verjela vame in kot pravi star slovenski pregovor, ker je držala tri vogale hiše. Pa še pri četrtem sem ji le pomagal. Hvala sinčku Jakobu za vse prelepe trenutke. Opravičujem se jima za prenekatere izgubljene vikende in popoldneve, ko sta razumevajoče prenašala moje sedenje za računalnikom.

Povzetek

Doktorska disertacija obravnava problematiko sprotne identifikacije mehkega modela. Razviti sta *c*-means in Gustafson-Kessel metoda za sprotno rojenje. Metodi sprotnega rojenja sta uporabljeni v kombinaciji z rekurzivnimi mehкими najmanjšimi kvadrati. Skupaj tvorijo mehanizem za sprotno identifikacijo in sprotno adaptacijo mehkega modela. Omenjen koncept rekurzivne mehke identifikacije je nadalje razvit v samorazvijajočo eFuMo metodo identifikacije. Tu poleg sprotne identifikacije uporabljamo tudi mehanizme za dodajanje novih rojev, odstranjevanje nepomembnih rojev, deljenje rojev in združevanje podobnih rojev. Podan je pregled mehanizmov, ki jih uporabljajo druge metode. Predstavljena je ideja predlaganih mehanizmov, ki so implementirani v predlagani eFuMo metodi.

V disertaciji sta podana dva možna primera uporabe predlagane metode. Prvi primer se navezuje na mehki prediktivni regulator (FPFC), kjer smo vpeljali mehki adaptivni model. Začetni mehki model se s pomočjo razvite metode prilagaja na novo dinamiko v procesu. S tem se izboljša kvaliteta regulacije. Predlagani adaptivni mehki prediktivni funkcionalni regulator (AFPFC) je bil testiran na primeru vodenja semi-saržnega reaktorja. Eksperiment je bil izveden na simulacijskem modelu z dodanim šumom, ki dobro opisuje dejanski proces. Z uporabo AFPFC smo izboljšali sledenje referenci in zmanjšali število preklapov med hladilnim in grelnim medijem. AFPFC regulacija je bila primerjana z regulacijo z FPFC.

V drugem primeru je bila predlagana eFuMo metoda uporabljena kot ključni element v sistemu za zaznavanje napak na procesu čistilne naprave. S pomočjo sistema smo poskušali zaznati izpade senzorjev na signalih, povezanih s koncentracijo kisika, in oceniti signale za čas izpada le-tega. Predlagani sistem za detekcijo ni razvit do potankosti. Razvita je bila le osnovna ideja, ki nam je omogočala testiranje eFuMo metode. Podatki, na katerih je bil sistem testiran, so bili pridobljeni iz realne naprave.

Oba primera kažeta na potencialno uporabnost metode v realnem svetu. Poleg omenjenih aplikacij je bila metoda testirana tudi na nekaterih tipičnih simulacijskih primerih, kot je na primer Mackey-Glass časovna vrsta. Dobljeni rezultati so bili primerjani z nekaterimi bolj znanimi in uveljavljenimi metodami identifikacije mehkega modela.

Abstract

The PhD thesis deals with the recursive identification of the Takagi-Sugeno fuzzy model. The two methods for recursive clustering are proposed: the recursive fuzzy c -means clustering and the recursive Gustafson-Kessel clustering. The methods are then used in combination with the recursive fuzzy least squares method. The combination gives us the method for recursive identification of fuzzy model. Since both, c -means and Gustafson-Kessel clustering methods assume a fixed number of cluster, we then implemented the evolving mechanisms for adding, removing, splitting and merging of clusters. This gives us an evolving fuzzy model method (eFuMo method), which is able to change the number of clusters if necessary. The evolving mechanisms of other established methods are described and compared to proposed evolving mechanisms.

After establishing the concepts and algorithms of eFuMo, we also give two examples of method's potential use. The first example is from a control area. We took the algorithm of the fuzzy predictive functional controller (FPFC) and tried to expend it with an adaptive fuzzy model. We proposed an adaptive fuzzy predictive functional controller (AFPFC). The recursive fuzzy model identification method proposed in the first chapter was used to adapt the fuzzy model to the process dynamics. The control algorithm was tested on a semi-batch reactor model with the added noise. The simulation model is a very accurate representation of the real plant. The AFPFC controller was compared to the FPFC controller. We showed that AFPFC controller can improve the reference following and lower the number of switches between the cooling and heating media.

In the second example we propose a basic fault detection system and soft sensor for a waste water treatment process, where the eFuMo method plays an important part in adapting the process model. The system was constructed for monitoring the oxygen concentration in the plant. It is able to detect the sensor fallouts and to estimate the sensor output during the fallout. The data, on which the system was tested, were collected from a real plant.

Both presented examples gave a very promising results that show the potential use of the method in a real world application. Beside the mentioned examples, the method was also tested some benchmark examples like Mackey-Glass timeseries prediction. The obtained results were compared to results, obtained with more established methods for on-line fuzzy model identification.

Vsebina

Seznam slik	xiii
Seznam tabel	xvii
1. Uvod	1
2. Mehki model	3
2.1 Delovanje mehkega Takagi-Sugeno modela	5
3. Identifikacija mehkega Takagi-Sugeno modela	7
3.1 Identifikacija rojev	7
3.1.1 Mehko c -means in Gustafson-Kessel rojenje	8
3.1.2 Identifikacija parametrov linearnih podmodelov	11
4. Rekurzivna identifikacija mehkega modela	13
4.1 Rekurzivna mehka c -means in Gustafson-Kessel identifikacija rojev	17
4.1.1 Primerjava sprotne identifikacije in klasične c -means identifikacije rojev	19
4.1.2 Rekurzivno računanje mehke kovariančne matrike	23
4.2 Rekurzivna identifikacija lokalnih linearnih modelov	27
4.3 O parametrih mehke rekurzivne identifikacije	32
4.4 Primerjava metode z nekaterimi drugimi samorazvijajočimi metodami	36

5. Vodenje z adaptivnim mehkim modelom	43
5.1 Mehki prediktivno funkcijski regulator z adaptivnim mehkim modelom	44
5.2 Vodenje semi-saržnega reaktorja z AFPFC	46
5.2.1 Opis procesa in regulacijskega problema	47
5.2.2 Vodenje procesa	50
5.2.3 Komentar	61
6. Samorazvijajoča mehka identifikacija	63
6.1 Dodajanje rojev	64
6.1.1 Razdalja po posameznih komponentah	67
6.1.2 Evklidska razdalja	73
6.1.3 Mahalanobisova razdalja	73
6.1.4 Dodatni varnostni pogoji	75
6.1.5 Inicializacija novega roja	76
6.2 Odstranjevanje rojev	77
6.3 Združevanje rojev	80
6.4 Deljenje rojev	84
6.5 Delovanje eFuMo metode	86
6.5.1 Primerjava eFuMo metode z drugimi	90
7. Zaznavanje napak na primeru čistilne naprave	99
7.1 Opis procesa in problema	102
7.2 Spremljanje koncentracije kisika	103
8. Sklep	119
9. Doprinos k znanosti	121

Literatura

123

Seznam slik

2.1	Shema mehkega modela.	4
2.2	Koncept mehkega modela.	6
4.1	Prikaz centrov v podatkovnem prostoru.	21
4.2	Časovni prikaz pozicij centrov rojev pri rekurzivnem rojenju. . . .	22
4.3	Prikaz vpliva različnih faktorjev pozabljanja na izračun kova- riančne matrike.	25
4.4	Različne postavitve rojev.	29
4.5	Absolutna napaka.	36
4.6	Pozicije centrov po x osi.	36
4.7	Pozicije centrov po y osi.	36
4.8	Pozicije centrov po z osi.	36
4.9	Vpliv faktorja pozabljanja na variacije v oceni parametrov.	37
4.10	Vpliv faktorja pozabljanja na hitrost prilagajanja modela.	37
4.11	Vpliv faktorja mehkosti.	37
4.12	Vpliv faktorja prekrivanja.	37
4.13	Predikcija MG časovne vrste	40
4.14	Predikcija MG časovne vrste pri pošumljenem signalu	41
5.1	Sehma semi-saržnega reaktorja.	48
5.2	Koeficient prevajanja toplote glikola in specifična toplota glikola. .	50
5.3	Ojačanje procesa.	52
5.4	Časovna konstanta procesa.	52

5.5	Odziv procesa pri vodenju z AFPFC in FPFC.	55
5.6	Izhod regulatorja in razlika med dejanskim vhodom v proces in izhodom regulatorja.	55
5.7	Pozicija diskretnega ventila za vroč glikol in pozicija mešalnega ventila.	56
5.8	Odziv procesa na dodajanje novega materiala.	57
5.9	Izhod regulatorja in razlika med dejanskim in želenim vhodom v proces ob dodanem novem materialu.	57
5.10	Pozicija ventila za vroč glikol in mešalnega ventila na intervalu dodajanja.	58
5.11	Odziv procesa po dodanem novem materialu.	59
5.12	Izhod regulatorja in razlika med dejanskim in želenim vhodom v proces po dodanem novem materialu.	59
5.13	Pozicija ventila za vroč glikol in mešalnega ventila po intervalu dodajanja.	60
5.14	Izhod mehkega modela in procesa pri AFPFC in napaka modela. . .	61
5.15	Izhod mehkega modela in procesa pri FPFC in napaka modela. . .	62
6.1	Shema samorazvijajoče metode (zgornji nivo).	64
6.2	Območje dodajanja, kjer vse komponente vektorja razdalje zadoščajo pogoju dodajanja.	69
6.3	Območje dodajanja, kjer prva komponenta vektorja razdalje zadošča pogoju dodajanja.	70
6.4	Območje dodajanja, kjer ena od komponent vektorja razdalje zadošča pogoju dodajanja	71
6.5	Območje dodajanja, kjer komponente vektorja razdalje skupno zadoščajo pogoju dodajanja.	72
6.6	Območje dodajanja, kjer Evklidova razdalja izpolnjuje pogoj dodajanja.	74

6.7	Območje dodajanja, kjer Mahalanobisova razdalja izpolnjuje pogoj dodajanja.	75
6.8	Izvajanje eFuMo metode (CDL blok).	87
6.9	Rekurzivno rojenje.	89
6.10	Rekurzivna identifikacija lokalnih modelov.	89
6.11	Primerjava med predikcijo in dejanskim izhodom procesa.	93
6.12	Napaka med realnim izhodom in predikcijo.	94
6.13	Pripadnostne funkcije rojev.	95
6.14	Roji in njihova območja.	96
7.1	Shema čistilne naprave.	103
7.2	Shema sistema za zaznavanje napak in mehkega senzorja.	104
7.3	Shema učenja mehkih modelov.	106
7.4	Odprtosti ventila.	108
7.5	Prvi eksperiment: signal pretoka zraka in signal koncentracije kisika.	109
7.6	Prvi eksperiment: signal pretoka zraka in signal koncentracije kisika	109
7.7	Prvi eksperiment: alarm signala pretoka zraka in signala koncentracije kisika.	110
7.8	Drugi eksperiment: signal pretoka zraka in signal koncentracije kisika.	111
7.9	Drugi eksperiment: signal pretoka zraka in signal koncentracije kisika.	111
7.10	Drugi eksperiment: alarm signala pretoka zraka in signala koncentracije kisika.	112
7.11	Prvi eksperiment: Število rojev za mehki model odprtost ventila - pretok zraka in mehki model pretok zraka - koncentracija kisika. .	113
7.12	Drugi eksperiment: Število rojev za mehki model odprtost ventila - pretok zraka in mehki model pretok zraka - koncentracija kisika.	114
7.13	Interval kalibracije senzorja koncentracije kisika.	116

7.14 Mehki senzor z uporabo detekcije samokalibracije na podlagi vari- anc signalov in na podlagi korelacijskega koeficienta.	118
--	-----

Seznam tabel

4.1	Pozicije rojev skozi iteracije pri časovno naključnem zaporedju vzorcev.	20
4.2	Pozicije rojev skozi iteracije pri časovnem zaporedju vzorcev, kjer najprej nastopijo vzorci prvega in nato drugega roja.	20
4.3	Primerjava predlagane rekurzivne FCM in GK metode z ostalimi.	39
5.1	Začetne vrednosti mehke <i>c</i> -means metode.	52
5.2	Začetne vrednosti in parametri procesa semi-saržnega reaktorja. .	53
5.3	Parametri AFPFC in preklopne logike.	54
6.1	Rezultati predikcije MG vrste z eFuMo metodo.	91
6.2	Glavne nastavitve eFuMo.	92
6.3	Primerjava rezultatov predikcije dinamičnega nelinearnega sistema.	93
6.4	Glavne nastavitve eFuMo za primer identifikacije nelinearnega dinamičnega sistema.	97
7.1	Napaka modela glede na realen signal.	108

1. Uvod

V današnjem času so zahteve po kvalitetni in učinkoviti proizvodnji zelo narasle. Zaradi nižanja stroškov proizvodnje in zmanjšanja človeškega vpliva na proizvodnjo se že od približno 1975. leta dalje, ko so se pojavili prvi poceni in dostopni mikroprocesorji, poskuša nadomestiti človeka z računalniškimi sistemi. Ti sistemi izvajajo monotone naloge namesto njega in s tem zagotavljajo ponovljivost kvalitete izdelka ter prispevajo k večji produktivnosti same proizvodnje. V veliki meri pa se uporabljajo tudi kot ekspertni sistemi, ki pomagajo operatorjem oziroma nadzornikom proizvodnje linije pri sprejemanju odločitev.

Sprva so mikroprocesorji prevzeli večji del regulacije samih procesov. Danes praktično ni mogoče najti resnega proizvajalca, ki nima celotne proizvodnje avtomatizirane. Sam nadzor nad regulacijskimi sistemi je izvajal človek. Najprej z obhodom po celotni proizvodnji in preverjanjem vrednosti na lokalnih merilnikih, kasneje pa je to delo opravljal in ga opravlja še danes v eni nadzorni sobi, kjer so zbrani vsi podatki (meritve) o procesih na enem mestu. Hitri in poceni senzorji ter oprema za zajem podatkov nam v današnjem času omogoča zajem velike količine meritev, ki kažejo, kaj se trenutno dogaja s procesom. Tudi sprotno spremljanje podatkov ni ravno zanimivo in dinamično delo, hkrati pa so kriteriji za določitev napake od človeka do človeka drugačni. Danes se lahko na enem mestu spremlja toliko procesov in zgenerira toliko podatkov, da jih en sam človek zelo težko obdelava in določi mesto napake v procesu. Zato je bilo za namen obdelave in analize teh podatkov razvito veliko statističnih metod [84, 44], ki pomagajo operaterju pri obdelavi podatkov in opozarjajo na možne alarme in napake v procesu.

Nadzor nad delovanjem procesov je nujen, saj kljub tehnološkemu napredku, kateremu smo priča zadnjih nekaj desetletij, ni mogoče proizvesti popolnoma zanesljivih senzorjev niti regulatorjev. Zato je potrebno za učinkovito proizvodnjo

processe avtomatizirati in hkrati implementirati metode, ki nadzorujejo delovanje teh procesov in sporočajo stanja procesov nadzorniku oziroma operaterju.

Za izgradnjo učinkovitega vodenja in sistema za spremljanje procesa največkrat potrebujemo nek model, ki zadovoljivo dobro opisuje sam proces. Model lahko dobimo s pomočjo matematičnega modeliranja ali pa s pomočjo identifikacije.

Realni procesi so pogostokrat nelinearni in časovno spremenljivi. Pri takih procesih je smiselno uporabiti tudi nelinearni model procesa, ki se je sposoben prilagoditi trenutnim razmeram v procesu. Tako se izognemo napačnim alarmom pri sistemu detekcije, izboljšamo pa lahko tudi regulacijo. Za izgradnjo takega modela ponavadi uporabimo identifikacijske postopke, ki so sposobni identificirati nelinearni model, hkrati pa nudijo možnost prilagajanja modela trenutnim razmeram.

Zelo preprost pristop za opis nelinearnega procesa je mehki Takagi-Sugeno model. Prednost tega modela je, da je zelo intuitiven, lahko razumljiv in preprost za uporabo. Zaradi tega je tudi pogostokrat uporabljen v vodenju, v sistemih za detekcijo napak in tudi na drugih področjih.

V doktorski disertaciji bomo predstavili rekurzivno metodo učenja mehkega Takagi-Sugeno modela. Metoda je izpeljana iz mehke *c*-means identifikacije. Predstavljena metoda nam omogoča možnost adaptacije mehkega modela pri časovno spremenljivih procesih. Glavni motiv pri razvoju metode je bil njena preprostost in učinkovitost. Delovanje metode bomo prikazali na primerih vodenja in na primeru detekcije napak v procesu.

Disertacijo lahko v grobem razdelimo na štiri tematske sklope. V prvem sklopu je predstavljena razvita mehka rekurzivna identifikacija. Drugi sklop je posvečen uporabi razvite metode v vodenju. V tretjem sklopu rekurzivni identifikaciji dodamo samorazvijajoče algoritme in tako tvorimo samorazvijajočo metodo. Četrty sklop pa govori o možni uporabi predlagane metode za spremljanje procesov in odkrivanje napak.

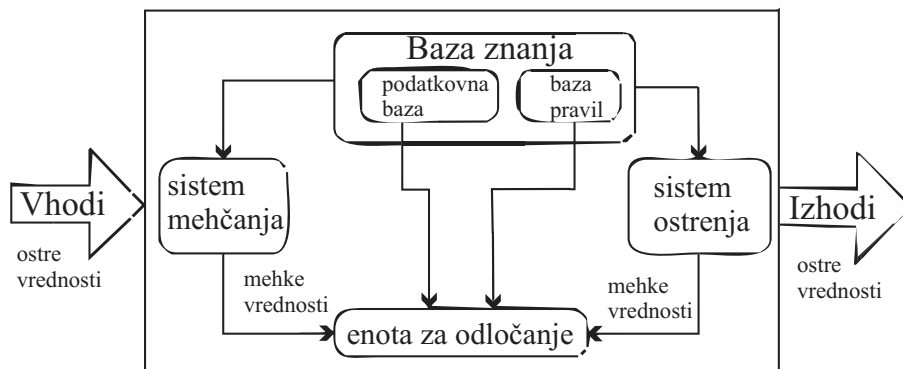
2. Mehki model

Realni procesi so večinoma nelinearni. Gradnja njihovih matematičnih modelov je ponavadi zahtevna in dolgotrajna. Dobljeni modeli so lahko visokih redov in težko razumljivi. Linearna aproksimacija takih procesov je zadovoljiva le v nekem ozkem območju delovne točke, za katero so bili modeli zgrajeni. Z več modelnim pristopom (*multi-model approach*) lahko te težave dokaj učinkovito rešimo [21]. Pri tem pristopu nelinearni model zgradimo z množico linearnih modelov, ki v okolici svoje delovne točke zadovoljivo dobro opišejo nelinearni proces. Ti linearni modeli so ponavadi nizkih redov in dobro razumljivi. Te ideje se poslužujejo tudi mehki modeli [159], katerih koncept temelji na teoriji mehke logike [188], [187]. Za modeliranje procesov so v literaturi največkrat uporabljeni Takagi-Sugeno (TS) mehki modeli, ki so uporabno in učinkovito orodje za modeliranje in vodenje kompleksnih sistemov. Ideja mehkih TS modelov je razdeliti nelinearni sistem na mehka področja/roje (*clusters*). Na njih so definirani linearni modeli. Ti aproksimirajo nelinearni sistem [83] na določenem mehkem področju z linearnim modelom. Z mehkim zlitjem izhodov posameznih linearnih modelov aproksimiramo obnašanje nelinearnega sistema. S tem pristopom lahko nelinearni sistem aproksimiramo s poljubno natančnostjo [78].

Na splošno je mehki model največkrat predstavljen kot na shemi na sliki 2.1. Model je sestavljen iz petih blokov ([18],[129],[153]):

sistem mehčanja (*Fuzzification interface/fuzzifier*): ostre vrednosti vhodov zmehta tako, da jim pripiše pripadnost k določeni lingvistični vrednosti ali roju. To stori na podlagi pripadnostnih funkcij.

podatkovna baza (*Fuzzy inference engine/decision-making unit*): v podatkovni bazi so shranjene pripadnostne funkcije.



Slika 2.1: Shema mehkega modela.

baza pravil (*Fuzzy inference engine/decision-making unit*): v bazi pravil imamo shranjena *if-then* pravila.

enota za odločanje (*Fuzzy inference engine/decision-making unit*): izvaja proces sklepanja in operacij nad pravili, da dobimo mehki izhod modela.

sistem ostrenja (*Defuzzification interface/defuzzifier*): sistem ostrenja pretvori mehke vrednosti nazaj v ostre.

Proces določanja izhoda modela se prične z mehčanjem ostrih vhodov. Na podlagi primerjave vhodov s pripadnostnimi funkcijami na permisni strani določimo pripadnost trenutne vhodne slike k posameznim rojem oziroma lingvističnim spremenljivkam. Z določenim *T-norm* operatorjem združimo pripadnosti na permisni strani, da dobimo moč proženja posameznega pravila. Nato izračunamo posledično (consequent) stran pravila glede na moč proženja. Te izhode posameznih pravil združimo in tako dobimo oster izhod mehkega modela.

Mehke modele običajno razdelimo na tri tipe: Mamdani, Takagi-Sugeno in Tsukamoto (singleton mehki model) [64], [153].

V Mamdani modelu so vhodi in izhodi interpretirani kot lingvistične spremenljivke. Predstavljeni so v obliki pravil kot na primer:

$$R^i : \text{Če } x_1 = A_1^i \text{ in } x_2 = A_2^i \dots \text{ in } x_n = A_n^i, \text{ potem } y^i = B^i, \quad (2.1)$$

kjer so x vhodi, A_j^i so lingvistične spremenljivke (npr.: majhno, veliko, srednje ...), y^i je lokani izhod in B^i je izhodna lingvistična spremenljivka. Izhod se ponavadi izračuna z *min-max* normo [64].

Tsukamoto modeli, imenovani tudi singleton ali Takagi-Sugeno model ničtega reda, so predstavljeni v obliki:

$$R^i : \text{ Če } X = A^i, \text{ potem } y^i = k^i, \quad (2.2)$$

kjer vhodnim podatkom (X) priredimo pripadnosti k posameznim mehkim področjem (A^i). Vsako mehko področje ima definiran svoj izhod v obliki konstante (k^i). Izhod celotnega sistema pa je kombinacija posameznih lokalnih izhodov. Način, kako se izračuna izhod, je odvisen od izbire metode ostrenja.

Takagi-Sugeno model lahko predstavimo v obliki:

$$R^i : \text{ Če } X = A^i, \text{ potem } y^i = f^i(X), \quad (2.3)$$

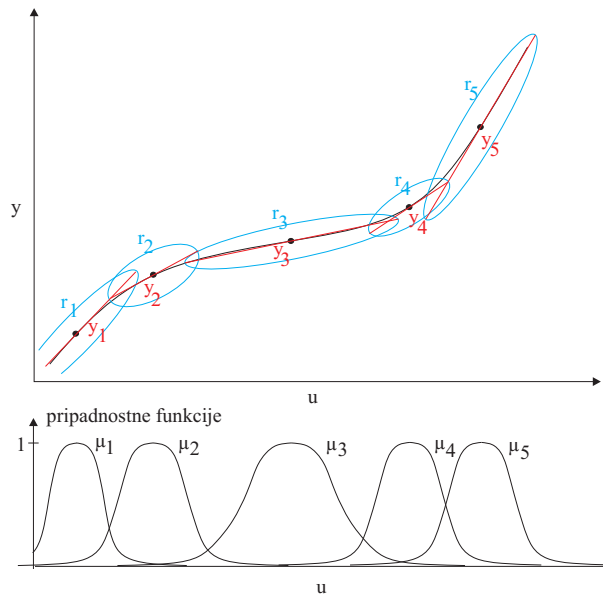
kjer vhodnim podatkom (X) priredimo pripadnosti k posameznim mehkim področjem. Izhod na področju se izračuna kot linearna kombinacija vhodov v obliki ARX modela (en. 2.4). Izhod celotnega sistema pa je ponovno kombinacija posameznih lokalnih izhodov y^i .

$$f^i(X) = a_1^i x_1 + a_2^i x_2 + \dots + a_n^i x_n + r_i. \quad (2.4)$$

Na tem mestu velja omeniti, da to niso edini mehki modeli, ki obstajajo. Vsi mehki modeli so grajeni na podobnem principu, razlikujejo pa se v predvsem v delu ostrenja. Pri vodenju in detekciji napak se najpogosteje za izgradnjo nelinearnega modela procesa uporablja mehki Takagi-Sugeno model. Zato si bomo v nadaljevanju pogledali delovanje tega modela bolj podrobno.

2.1 Delovanje mehkega Takagi-Sugeno modela

Poenostavljen koncept mehkega modela je predstavljen na sliki 2.2. Predpostavimo, da imamo neko nelinearno funkcijo, katere vrednost y je odvisna od spremenljivke u . Ideja mehkega modela je, da vhodno-izhodni prostor u - y deli na mehka območja (npr.: r_1, r_2, r_3, r_4 in r_5), kjer lahko obnašanje oziroma vrednost funkcije y opišemo z lokalnimi linearnimi funkcijami (y_1, y_2, y_3, y_4 in y_5). Mehka območja so v večini primerov opisana s pripadnostnimi funkcijami na oseh vhodnega območja ($\mu_1, \mu_2, \mu_3, \mu_4, \mu_5$). Načeloma so pripadnostne funkcije določene tudi za izhodno os, vendar mehki model, uporabljen v tej disertaciji, pri računanju



Slika 2.2: Koncept mehkega modela.

izhoda teh pripadnosti ne upošteva. Za računanje izhoda oziroma vrednosti funkcije y (sl. 2.2) je potrebno vходу oziroma spremenljivki u določiti pripadnosti $(\mu_1(u), \mu_2(u), \mu_3(u), \mu_4(u), \mu_5(u))$. To naredimo tako, da pogledamo vrednosti pripadnostnih funkcij glede na vrednost spremenljivke u . Vrednost pripadnosti nato normiramo:

$$\beta_i(u) = \frac{\mu_i(u)}{\sum_{j=1}^5 \mu_j(u)} \quad i = 1, \dots, 5. \quad (2.5)$$

Nato izračunamo izhode linearnih funkcij (modelov) za določen u : $y_1(u)$, $y_2(u)$, $y_3(u)$, $y_4(u)$ in $y_5(u)$. Izhod mehkega modela in s tem aproksimacijo nelinearne funkcije dobimo z uteženo vsoto izhodov lokalnih linearnih funkcij (modelov):

$$\hat{y}(u) = \sum_{i=1}^5 \beta_i(u) y_i(u). \quad (2.6)$$

3. Identifikacija mehkega Takagi-Sugeno modela

Za identifikacijo Takagi-Sugeno (TS) mehkega modela je potrebno identificirati mehka področja in pripadajoče linearne modele [159]. Mehka področja se lahko določijo z rojenjem na vhodno-izhodnem prostoru, parametri linearnih podmodelov pa so v veliki večini identificirani z metodo uteženih najmanjših kvadratov [32].

TS model lahko identificiramo na dva načina: z *off-line* identifikacijo, kjer imamo prisotne vse podatke (npr.: [99], [96], [153], [89]) ali z *on-line* oziroma sprotno identifikacijo, kjer se model gradi sprotno s prihajanjem novih podatkov (npr.: [6], [92], [65]).

Preden se osredotočimo na rekurzivno mehko identifikacijo, si pogledjmo *off-line* postopek identifikacije mehkega modela in algoritem, ki je tudi naša osnova za nadaljno izgradnjo rekurzivne/sprotne identifikacije.

3.1 Identifikacija rojev

Pri identifikaciji rojev delimo vhodno-izhodni prostor na področja, za katera lahko identificiramo lokalne linearne modele, ki na teh področjih dovolj dobro opisujejo obnašanje procesa. Mehka območja običajno določimo z rojenjem. Za deljenje vhodno-izhodnega prostora obstajajo različne metode, kot na primer: LOLIMOT [131], SUHICLUST [155], *mountain clustering* [144] ali njeno izboljšano različico imenovano rojenje z odštevanjem (*subtractive clustering*) [32], ICLA [174], *fuzzy c-means* (FCM) [20], *Gustafson-Kessel* (GK) [42], *Gath-Geva* [67], *possibilistic c-means* (PCM) [184] ...

V nadaljevanju bomo podali mehko *c-means* (FCM) metodo rojenja, ki nam

služi kot osnova za izpeljavo rekurzivne mehke identifikacije. Glavni razlog za izbiro te metode je v tem, da je metoda preprosta in razumljiva. Centri rojev so povprečja vhodno-izhodnih podatkov, za razliko od rojenja z odštevanjem (*subtractive clustering*), kjer so centri rojev lahko le določeni pretekli vzorci. Hkrati je lahko razširljiva v GK metodo rojenja. Izkaže se tudi, da sta za rekurzivno uporabo FCM in GK metodi bolj primerni kot *Gath-Geva* metoda rojenja, ker sta numerično bolj prizanesljivi.

3.1.1 Mehko c -means in Gustafson-Kessel rojenje

Mehko c -means rojenje [20] je osnova za našo izpeljavo rekurzivne metode. Za identifikacijo mehkih področij moramo najprej urediti vhodno-izhodne podatke v matriko. Vzemimo $z + 1$ signalov, kjer je z število vhodov (x) v naš proces in kjer imamo en izhod (y). Za vsak trenutek k lahko te podatke uredimo v $z + 1$ dimenzionalni vektor $\mathbf{x}(\mathbf{k})^T = [x_1(k), \dots, x_z(k), y(k)]$, $\mathbf{x}(\mathbf{k}) \in \mathbb{R}^{z+1}$, kjer je $x_z(k)$ vzorec z -tega vhoda v trenutku k . Nabor n meritev oziroma vzorcev uredimo v matriko $\mathbf{X} = \{\mathbf{x}(\mathbf{k}) \mid k = 1, 2, \dots, n\}$, $\mathbf{X} \in \mathbb{R}^{n \times z+1}$. Cilj rojenja je razdeliti matriko $\mathbf{X}$ v c particij oziroma rojev.

$$\mathbf{X} = \begin{bmatrix} x_1(1) & x_2(1) & \dots & x_z(1) & x_{z+1}(1) \\ x_1(2) & x_2(2) & \dots & x_z(2) & x_{z+1}(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1(n) & x_2(n) & \dots & x_z(n) & x_{z+1}(n) \end{bmatrix}. \quad (3.1)$$

Mehko razdeljeno matriko $\mathbf{X}$ predstavlja množica mehkih podmnožic $\{\mathbf{A}_i \mid 1 \leq i \leq c\}$. Te so definirane s pripadnostnimi funkcijami, ki so implicitno vsebovane v pripadnostni matriki $\mathbf{U} = [\mu_i(k)] \in \mathbb{R}^{c \times n}$. V i -ti vrsti matrike $\mathbf{U}$ so vrednosti pripadnostne funkcije k -tega vzorca k i -temu roju $\mathbf{A}_i$. Matrika $\mathbf{U}$ izpolnjuje naslednje pogoje: pripadnosti so realna števila iz intervala $\mu_i(k) \in [0, 1]$, $1 \leq i \leq c$, $1 \leq k \leq n$; pripadnost vzorca k uniji rojev je enaka ena $\sum_{i=1}^c \mu_i(k) = 1$, $1 \leq k \leq n$; noben roj ni prazen in ne vsebuje vseh podatkov $0 < \sum_{k=1}^n \mu_i(k) < n$, $1 \leq i \leq c$. To pomeni, da matrika $\mathbf{U}$ pripada množici, ki je definirana kot:

$$\mathbf{M} = \{\mathbf{U} \in \mathbb{R}^{c \times n} \mid \mu_i(k) \in [0, 1], \forall i, k; \sum_{i=1}^c \mu_i(k) = 1, \forall k; 0 < \sum_{k=1}^n \mu_i(k) < n, \forall i\}. \quad (3.2)$$

Algoritem mehkega deljenja vhodno-izhodnega prostora temelji na minimiranju kriterijske funkcije:

$$J(\mathbf{X}, \mathbf{U}, \mathbf{V}) = \sum_{i=1}^c \sum_{k=1}^n (\mu_i(k))^\eta d^2(\mathbf{x}(k), \mathbf{v}_i) \quad (3.3)$$

pri upoštevanju omejitve:

$$\sum_{i=1}^c \mu_i(k) = 1 \quad \forall k. \quad (3.4)$$

Matrika $\mathbf{V}$ v enačbi 3.3 predstavlja centre mehkih območij $\mathbf{v}_i \in \mathbb{R}^{z+1}$; η je faktor mehкости območij, ki na splošno lahko obsega vrednosti med ena in neskončno, običajna vrednost tega parametra pa je $\eta = 2$; d^2 je razdalja med centri rojev in vzorci. Katero razdaljo bomo uporabili, je odvisno od problema, ki ga rešujemo. V mehkem c -means algoritmu je uporabljena Evklidska norma (en. 3.5), Gustafson-Kessel pa predvideva Mahalanobisovo normo (en. 3.6).

$$d_{ik}^2 = (\mathbf{x}(k) - \mathbf{v}_i)^T (\mathbf{x}(k) - \mathbf{v}_i), \quad 1 \leq i \leq c, \quad 1 \leq k \leq n. \quad (3.5)$$

$$D_{ik}^2 = (\mathbf{x}(k) - \mathbf{v}_i)^T \mathbf{A}_i (\mathbf{x}(k) - \mathbf{v}_i), \quad 1 \leq i \leq c, \quad 1 \leq k \leq n. \quad (3.6)$$

Slednja pri izračunu razdalje upošteva še obliko roja, ki jo določimo iz mehke kovariančne matrike $\mathbf{F}_i$ (en. 3.7). Ta matrika nam opisuje porazdelitev vzorcev okoli posameznega centra roja.

$$\mathbf{A}_i = [\rho_i \det(\mathbf{F}_i)]^{1/p} \mathbf{F}_i^{-1} \quad (3.7)$$

V enačbi 3.7 ρ_i označuje volumen roja. V literaturi najdemo različne vrednosti za ta parameter. V naših primerih bomo uporabljali vrednost $\rho_i = 1$. Parameter p označuje število vrstic v mehki kovariančni matriki (za naš primer $p = z + 1$). Velikokrat se za matriko $\mathbf{A}_i$ uporabi kar inverzno vrednost mehke kovariančne matrike.

Mehki c -means algoritem rojenja predpostavlja konstantno število rojev. Za določanje števila rojev lahko uporabimo funkcije za merjenje veljavnosti rojev ali pa jih iterativno dodajamo oziroma odstranjujemo glede na odstopanje modela.

Enačbo za pozicije centrov rojev in pripadnostnih funkcij dobimo z minimizacijo kriterijske funkcije 3.3 ob upoštevanju pogoja 3.4. Enačba za pozicijo centrov je sledeča:

$$\mathbf{v}_i = \frac{\sum_{k=1}^n (\mu_{ik})^\eta \mathbf{x}_k}{\sum_{k=1}^n (\mu_{ik})^\eta}, \quad 1 \leq i \leq c \quad (3.8)$$

in za pripadnost:

$$\mu_{ik} = \frac{1}{\left(\frac{d_{ik}}{\sum_{j=1}^c d_{jk}} \right)^{\frac{2}{\eta-1}}}. \quad (3.9)$$

Mehka kovariančna matrika [20] se računa po spodnji enačbi:

$$\mathbf{F}_i = \frac{\sum_{k=1}^n \mu_{ik}^\eta (\mathbf{x}(k) - \mathbf{v}_i) (\mathbf{x}(k) - \mathbf{v}_i)^T}{\sum_{k=1}^n \mu_{ik}^\eta}, \quad 1 \leq i \leq c. \quad (3.10)$$

Opisani algoritem rojenja je iterativni. Opišemo ga lahko z naslednjimi koraki:

- **Inicializacija:** Izberemo število rojev c , faktor mehkosti η (običajno $\eta = 2$) in napako, razliko $\epsilon_{end} > 0$, ki določa kdaj se bo izvedla zadnja iteracija. Inicializiramo matriko pripadnosti: $U \in M$ (naključno) in epoch $r = 0$,

- **Zanka:**

$$r = r + 1$$

, izračun centrov rojev:

$$\mathbf{v}_i^{(r)} = \frac{\sum_{k=1}^n \left(\mu_{ik}^{(r)} \right)^\eta \mathbf{x}(k)}{\sum_{k=1}^n \left(\mu_{ik}^{(r)} \right)^\eta}, \quad 1 \leq i \leq c, \quad (3.11)$$

izračun kovariančne matrike in matrike $\mathbf{A}_i$:

$$\mathbf{F}_i^{(r)} = \frac{\sum_{k=1}^n \left(\mu_{ik}^{(r)} \right)^\eta \left(\mathbf{x}(k) - \mathbf{v}_i^{(r)} \right) \left(\mathbf{x}(k) - \mathbf{v}_i^{(r)} \right)^T}{\sum_{k=1}^n \left(\mu_{ik}^{(r)} \right)^\eta}, \quad (3.12)$$

$$\mathbf{A}_i^{(r)} = \left(\rho_i \det \left(\mathbf{F}_i^{(r)} \right) \right)^{1/p} \mathbf{F}_i^{-1(r)}, \quad 1 \leq i \leq c, \quad (3.13)$$

izračun razdalje vzorcev do centrov rojev:

$$d_{ik}^2 = \left(\mathbf{x}(k) - \mathbf{v}_i^{(r)} \right)^T \mathbf{A}_i^{(r)} \left(\mathbf{x}(k) - \mathbf{v}_i^{(r)} \right), \quad (3.14)$$

$$1 \leq i \leq c, \quad 1 \leq k \leq n,$$

osvežitev matrike pripadnosti $\mathbf{U}$ z novimi pripadnostmi:

$$\mu_{ik}^{(r+1)} = \frac{1}{\left(\frac{d_{ik}}{\sum_{j=1}^c d_{jk}}\right)^{\frac{2}{\eta-1}}}, \quad (3.15)$$

- **dokler ni** $\|\mathbf{U}^{(r+1)} - \mathbf{U}^{(r)}\| < \epsilon_{end}$.

Algoritem se ustavi, ko je razlika med matriko pripadnosti iz prejšnjega epoha in trenutnega manjša od ϵ_{end} . Ta pogoj je lahko vezan tudi na spremembo kriterijske funkcije 3.3.

3.1.2 Identifikacija parametrov linearnih podmodelov

V večini primerov se pri identifikaciji lokalnih linearnih modelov uporablja metoda mehkih najmanjših kvadratov. Za razliko od običajnih najmanjših kvadratov je potrebno pri metodi mehkih najmanjših kvadratov upoštevati proženje rojev pri posameznem vzorcu. Proženje nam podaja matrika $\mathbf{U}$, ki jo dobimo ob koncu postopka rojenja (en. 3.19). To matriko uporabimo za utežno matriko. Za vsak lokalni model lahko zapišemo sledečo kriterijsko funkcijo: [6]:

$$J_i = (\mathbf{y}_i - \mathbf{X}_i \theta_i)^T (\mathbf{y}_i - \mathbf{X}_i \theta_i) \quad i = 1, \dots, c, \quad (3.16)$$

kjer je θ_i vektor parametrov lokalnega modela i -tega roja, $\mathbf{y}_i$ vektor izhodov za i -ti model - zadnji stolpec matrike $\mathbf{X}$ iz enačbe 3.1 pomnožen s proženji roja:

$$\mathbf{y}_i = \begin{bmatrix} \mu_{i1} x_{z+1}(1) \\ \mu_{i2} x_{z+1}(2) \\ \vdots \\ \mu_{in} x_{z+1}(n) \end{bmatrix} \quad (3.17)$$

in $\mathbf{X}_i$ je matrika vhodov za i -ti lokalni model:

$$\mathbf{U} = \begin{bmatrix} \mu_{i1} x_1(1) & \cdots & \mu_{i1} x_z(1) \\ \vdots & \ddots & \vdots \\ \mu_{in} x_1(n) & \cdots & \mu_{in} x_z(n) \end{bmatrix}, \quad (3.18)$$

$$\mathbf{U} = \begin{bmatrix} \mu_{11} & \cdots & \mu_{1n} \\ \vdots & \ddots & \vdots \\ \mu_{c1} & \cdots & \mu_{cn} \end{bmatrix}. \quad (3.19)$$

V zgornjih enačbah je c število mehkih področij oziroma rojev in n število meritev. Če minimiziramo optimizacijski problem, dobimo enačbo za oceno parametrov lokalnih modelov [54]:

$$\theta_i = [\mathbf{X}_i^T \mathbf{X}_i]^{-1} \mathbf{X}_i^T \mathbf{y}_i \quad i = 1, \dots, c. \quad (3.20)$$

4. Rekurzivna identifikacija mehkega modela

Veliko nelinearnih procesov je odvisnih od zunanjih ali drugih nemerljivih vplivov. Tem procesom se nelinearnost in dinamika spreminjata. Zato je za potrebe vodenja in nadzora takšnih procesov potrebno uporabiti sprotno identifikacijo, ki je sposobna mehki model prilagajati na spremenjene pogoje. Uporaba *off-line* metod za sprotno prilagajanje modela namreč ni primerna [6]. Zaradi vse večjega števila podatkov je potreba po razvijanju algoritmov sprotne identifikacije in rojenja vse večja [10]. Sprotna identifikacija, predstavljena v disertaciji, je bila razvita za sistem spremljanja procesa čiščenja odpadnih voda [166] in elektro obločne peči [115], [116], [114]. Za slednji proces se je kasneje izkazalo, da ni dovolj dosegljivih podatkov za uporabo rekurzivne identifikacije, zato se je uporabilo klasično modeliranje na podlagi fizikalnih zakonitosti.

Področje sprotne identifikacije nelinearnih sistemov lahko uvrstimo v področje samorazvijajočih se sistemov (*evolving system*). Področje se je začelo razvijati okoli leta 1991 z objavo članka [138] in metode RAN, čeprav je bilo osnovno pravilo za učenje adaptivnih nevronske mreže razvito že okoli leta 1970 [180]. Pravilo je temeljilo na metodi padajočega gradienta (*gradient decent*) in verižnem pravilu (*chain rule*). Vendar pa takrat to delo ni imelo večjega odmeva v strokovni javnosti, deloma tudi zato, ker nevronske mreže takrat še niso bile tako dobro razvite in uveljavljene. Od leta 1991 naprej je bilo na tem področju razvitih veliko metod. Na trenutno živahnost tega področja pa nakazujejo tudi nedavni izidi knjig kot so *Evolving Connectionist Systems* [93], *Evolving Intelligent Systems* [11] in *Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications* [118] ter začetek izhajanja revije *Evolving systems* [10], ki pokriva to področje.

Kot že rečeno, je bilo v zadnjih letih razvito veliko metod za rekurzivno identifikacijo nelinearnih sistemov. Nekateri pristopi temeljijo na mehki logiki (eTS [6], exTS [15, 5], simple_TS [9], +eTS [8], FLEXFIS [122], FLEXFIS+ [120]),

drugi pa na kombinaciji nevronske mreže in mehke logike (EFuNN [91, 90], DENFIS [92], ANFIS [153], GANFIS [17], SOFNN [106], SAFIS [150], SCFNN [111], NFCN [110], D-FNN [181], GD-FNN [182], SONFIN [85], NeuroFAST [165], RAN [138], ESOM [47], Neural gas [63], ENFM [156], GAP-RBF [77] in druge).

Metode, ki uporabljajo kombinacijo mehke logike in nevronske mreže, temeljijo na funkcionalni ekvivalenci mehkega modela z nevronskimi mrežami [81]. Mehki model oziroma mehki inferenčni sistem realizirajo z nevronskimi mrežami. Večina teh metod uporablja pet plasti nevronske mreže (SAFIS, SOFNN, D-FNN, EFuNN, ANFIS). V prvi plasti je vsakemu vhodu pripisana pripadnost k posameznemu roju. V drugi plasti se pripadnosti iz prve plasti pomnožijo za vsak roj tako, da dobimo pripadnost trenutne vhodne slike k posameznemu roju. Iz tega dobimo pripadnost trenutnega stanja na vhodu k posameznemu pravilu mehkega modela. V tretjem sloju se ta pripadnost normalizira. V četrtem sloju se izračunajo izhodi lokalnih linearnih modelov, ki pripadajo posameznim rojem oziroma pravilom. V petem pa se izračuna celotni izhod modela. Nekatere metode temeljijo tudi na šest stopenjskih oziroma štiri stopenjskih (SCFNN, GD-FNN) nevronske mreže. SONFIN ima poleg običajnih pet plasti pred prvo še eno plast, kjer ni nobenega računanja. Uteži povezav na naslednjo plast so ena. Plast služi za to, da vhode, ki niso korelirani z izhodom, izloči iz modela. Pri mrežah s štirimi plastmi je plast, kjer se stopnje proženja pravil normalizirajo, izpuščena.

Večina neuro-mehkih metod temelji na RBF (*radial basis function*) nevronske mreže oziroma na njihovi posplošeni obliki GRBF (GANFIS). Zaradi uporabe RBF mreže so vse širine Gausovih pripadnostnih funkcij, ki se nanašajo na isti vhod, enake [182]. Zato nekateri predlagajo uporabo EBF (*ellipsoidal basis functions*) nevronske mreže [182], [106] (GD-FNN, SOFNN).

Na neuro-mehki logiki so temeljile predvsem začetne metode, saj so s pomočjo prilagojenih algoritmov učenja nevronske mreže lahko sprotno adaptirale parametre mreže [180] in s tem posledično tudi parametre mehkega modela. Te metode se poslužujejo nadzorovanega načina učenja. Parametre delijo na linearne in nelinearne. Nelinearni parametri so povezani z delitvijo vhodno izhodnega prostora (centri rojev, širina pripadnostnih funkcij ...) in so v angleški literaturi imenovani kot *premise*, linearni pa so parametri lokalnih linearnih modelov, v angleški

literaturi imenovani kot *consequence*. Za identifikacijo rojev se pri teh metodah uporablja predvsem metoda padajočega gradienta (*gradient decent*), za identifikacijo linearnih parametrov pa Widrow-Hoff LMS (*least mean square*) algoritem ali pa rekurzivni najmanjši kvadrati. V zadnjem času se pojavlja kombinacija nadzorovanega in nenadzorovanega učenja. Rojenje se opravi z nenadzorovanim učenjem, identifikacija linearnih parametrov pa z metodo najmanjših kvadratov oziroma z metodo uteženih najmanjših kvadratov (npr.: DENFIS, NFCN). Takakega način učenja se poslužujejo tudi metode, ki temeljijo samo na mehki logiki (npr.: eTS). V ta namen so bile razvite metode rekurzivnega rojenja (npr.: ECM [92]). Metode rojenja so večinoma izpeljane iz njihovih *off-line* različic. Metoda eTS uporablja rekurzivno različico rojenja z odštevanjem [4] (*subtractive clustering* [32]), ENFM pa rekurzivno različico Gath-Geva rojenja. Redko se za deljenje vhodnega prostora uporablja algoritem, ki temelji na mehkiem adaptivnem ART (*fuzzy adaptive resonance theory*) konceptu (NeuroFAST). Ta prav tako spada med nenadzorovano učenje.

Mehke metode se razlikujejo po različnih tipih mehkih modelov. Večinoma se uporabljata Takagi-Sugeno mehki model prvega reda (ANFIS, SONFIN, D-FNN, GD-FNN, DENFIS, eTS, NeuroFAST, SOFNN ...) ali pa Takagi-Sugeno mehki model ničtega reda (SCFNN, SAFIS, GAP-RBF, EFuNN). Nekatere metode uporabljajo tudi posplošen mehki model, ki je kombinacija Mamdani modela in modela Takagi-Sugeno prvega reda (GANFIS).

Metode se razlikujejo tudi po sposobnostih prilagajanja. Nekatere (npr.: GANFIS, ANFIS) potrebujejo začetno strukturo mreže oziroma mehkega modela, ki ga lahko dobimo z *off-line* rojenjem. Število pravil se ne spreminja. Glede na vhodne in izhodne podatke se prilagajajo le parametri (adaptivne metode). Nekatere metode imajo vgrajene mehanizme dodajanja rojev (npr.: RAN, SONFIN, SCFNN, NeuroFAST, DENFIS, eTS, FLEXFIS), ne pa tudi odstranjevanja identificiranih rojev, ki niso več pomembni (inkrementalne metode). Samorazvijajoče metode pa imajo poleg mehanizmov za dodajanje rojev vgrajene tudi mehanizme za odstranjevanje rojev (ESOM, SAFIS, SOFNN, GAP-RBF, Growing Neural Gas (GNG), EFuNN, D-FNN, GD-FNN, ENFM, simpl.eTS, exTS, +eTS), nekatere celo za združevanje podobnih rojev (npr.: ENFM, SOFNN).

Pogoji za dodajanje pravil (novih centrov oziroma nevronov) so različni. Pri

GNG se doda nevron na vsakih n -vzorcev (n je konstanta, ki jo določi uporabnik). Največkrat pa je pogoj vezan na razdaljo trenutnega vzorca do centrov rojev (ESOM, DENFIS, FLEXFIS). Pri nadzorovanem rojenju je pogoj ponavadi vezan na napako med izhodom modela in procesa, v kombinaciji s pogojem, vezanim na razdaljo (EFuNN, D-FNN, GD-FNN, SAFIS, SCFNN) ali na ϵ -kompletnost oziroma v angleški literaturi imenovan kot ϵ -completeness (RAN, SCFNN, SONFIN, eTS). Slednji pogoj določa minimalno dopustno vrednost proženja najbližjega pravila (nenormirano) pri trenutnem vhodu.

Mehanizmi za odstranjevanje pravil temeljijo na starosti pravila (exTS, GNG, ESOM), velikosti podporne množice roja (+eTS), doprinosu pravila k izhodni napaki (SAFIS, GAP-RBF, D-FNN, GD-FNN), kombinaciji starosti in skupnega števila aktivacij (EFuNN) ali pa na minimalni dopustni razdalji med centri rojev (ENFM).

Deljenje vhodnega prostora na podlagi padajočega gradienta je počasno, ker je metoda podvržena lokalnim minimumom [153] [92]. Bolje je, če se za deljenje uporabljajo rekurzivne metode rojenja [6].

Za potrebe adaptiranja mehkega modela predlagamo rekurzivno metodo rojenja, ki temelji na mehki c -means metodi. Lahko pa jo razširimo tudi na Gustafson-Kessel metodo rojenja. Ker metoda predvideva fiksno število rojev, sta računski čas in potreba po spominu vnaprej določena. Zaradi uvedbe faktorja pozabljanja je predlagana metoda sposobna mehki model prilagoditi na spremenjeno delovanje procesa. Pri metodah, ki dodajajo roje, se pogosto opazi prehodne pojave na predikciji izhoda, ko je dodan nov roj. Na izhodu se lahko pojavljajo skoki (eTS). To se ponavadi zgodi, če začetni parametri novega roja niso dobro inicializirani, oziroma če je začetna kovariančna matrika najmanjših kvadratov prevelika. Predlagana metoda pri predikciji izhoda sistema ne povzroča večjih skokov, saj je prilagajanje zvezno (na hitrost bomo lahko vplivali s faktorji pozabljanja).

4.1 Rekurzivna mehka c -means in Gustafson-Kessel identifikacija rojev

Za izpeljavo rekurzivne identifikacije moramo pozicije centrov in mehko kovariančno matriko računati brez iteracij. Ideja pri izpeljavi temelji na dejstvu, da imamo nek začetni model, dobljen s klasičnim mehkim c -means rojenjem oziroma GK rojenjem. Torej imamo v trenutku k nek začetni model, ki je bil zgrajen z iteracijskim postopkom. Od tega trenutka naprej vsak nov podatek oziroma vzorec, ki ga dobimo, obravnavamo kot novo iteracijo. Za naslednji vzorec, ki pride v algoritem za rekurzivno identifikacijo, lahko enačbo centrov 3.11 zapišemo v obliki:

$$\mathbf{v}_i^{(r+1)} = \frac{\sum_{k=1}^{n+1} \left(\mu_{ik}^{(r+1)} \right)^\eta \mathbf{x}(k)}{\sum_{k=1}^{n+1} \left(\mu_{ik}^{(r+1)} \right)^\eta}, \quad 1 \leq i \leq c. \quad (4.1)$$

Pripadnosti $\mu_{ik}^{(r+1)}$ izračunamo pred adaptacijo centrov:

$$\mu_{ik}^{(r+1)} = \frac{1}{\left(\frac{d_{ik}}{\sum_{j=1}^c d_{jk}} \right)^{\frac{2}{\eta-1}}}, \quad (4.2)$$

kjer je razdalja izračunana glede na prejšnjo pozicijo centrov:

$$d_{ik}^2 = \left(\mathbf{x}(k) - \mathbf{v}_i^{(r)} \right)^T \mathbf{A}_i^{(r)} \left(\mathbf{x}(k) - \mathbf{v}_i^{(r)} \right), \quad 1 \leq i \leq c, \quad 1 \leq k \leq n+1. \quad (4.3)$$

Enačbo 4.1 lahko preuredimo v obliko:

$$\mathbf{v}_i^{(r+1)} = \mathbf{v}_i^{(r)} + \Delta \mathbf{v}_i^{(r+1)}. \quad (4.4)$$

Najprej v enačbi 4.1 ločimo stari del od novega:

$$\mathbf{v}_i^{(r+1)} = \frac{\sum_{k=1}^n \left(\mu_{ik}^{(r+1)} \right)^\eta \mathbf{x}(k) + \left(\mu_{ik+1}^{(r+1)} \right)^\eta \mathbf{x}(k+1)}{\sum_{k=1}^n \left(\mu_{ik}^{(r+1)} \right)^\eta + \left(\mu_{ik+1}^{(r+1)} \right)^\eta}. \quad (4.5)$$

Prvi člen ulomka množimo in delimo z vsoto pripadnosti in dobimo enačbo:

$$\mathbf{v}_i^{(r+1)} = \frac{\frac{\sum_{k=1}^n (\mu_{ik}^{(r+1)})^\eta \mathbf{x}(k)}{\sum_{k=1}^n (\mu_{ik}^{(r+1)})^\eta}}{\frac{\sum_{k=1}^{n+1} (\mu_{ik}^{(r+1)})^\eta}{\sum_{k=1}^n (\mu_{ik}^{(r+1)})^\eta}} + \frac{(\mu_{ik+1}^{(r+1)})^\eta \mathbf{x}(r+1)}{\sum_{k=1}^{n+1} (\mu_{ik}^{(r+1)})^\eta}. \quad (4.6)$$

Glede na to, da adaptiramo že prej identificirane roje, ne pričakujemo večje spremembe v pozicijah centrov in s tem tudi ne večje razlike v spremembi pripadnosti med računanjem $\mu_{ik}^{(r)}$ in $\mu_{ik}^{(r+1)}$ za novi vzorec. Zato predpostavimo, da bo razmerje, ki ga definira zgornji ulomek prvega člena v enačbi 4.6, približno enako razmerju v enačbi 3.11:

$$\frac{\sum_{k=1}^n (\mu_{ik}^{(r+1)})^\eta \mathbf{x}(k)}{\sum_{k=1}^n (\mu_{ik}^{(r+1)})^\eta} \approx \frac{\sum_{k=1}^n (\mu_{ik}^{(r)})^\eta \mathbf{x}(k)}{\sum_{k=1}^n (\mu_{ik}^{(r)})^\eta} = \mathbf{v}_i^{(r)}. \quad (4.7)$$

Z upoštevanjem enačbe 4.7 dobimo iz enačbe 4.6:

$$\mathbf{v}_i^{(r+1)} = \frac{\mathbf{v}_i^{(r)} \sum_{k=1}^n (\mu_{ik}^{(r+1)})^\eta}{\sum_{k=1}^{n+1} (\mu_{ik}^{(r+1)})^\eta} + \frac{(\mu_{ik+1}^{(r+1)})^\eta \mathbf{x}(k+1)}{\sum_{k=1}^n (\mu_{ik}^{(r+1)})^\eta + (\mu_{ik+1}^{(r+1)})^\eta}. \quad (4.8)$$

Zgornjo enačbo prevedemo v obliko:

$$\mathbf{v}_i^{(r+1)} = \mathbf{v}_i^{(r)} - \frac{\mathbf{v}_i^{(r)} (\mu_{ik+1}^{(r+1)})^\eta}{\sum_{k=1}^{n+1} (\mu_{ik}^{(r+1)})^\eta} + \frac{(\mu_{ik+1}^{(r+1)})^\eta \mathbf{x}(k+1)}{\sum_{k=1}^n (\mu_{ik}^{(r+1)})^\eta + (\mu_{ik+1}^{(r+1)})^\eta}. \quad (4.9)$$

Upoštevač obliko iz enačbe 4.4 lahko zapišemo spremembo pozicije centrov kot:

$$\Delta \mathbf{v}_i^{(r+1)} = \frac{(\mu_{ik+1}^{(r+1)})^\eta (\mathbf{x}(k+1) - \mathbf{v}_i^{(r)})}{\sum_{k=1}^n (\mu_{ik}^{(r+1)})^\eta + (\mu_{ik+1}^{(r+1)})^\eta}. \quad (4.10)$$

Vidimo, da spremembo pozicije centrov v enačbi 4.10 ni mogoče rekurzivno računati, saj je potrebno pri vsakem koraku za stare vzorce izračunati nove pripadnosti rojem. Zato moramo na tem mestu ponovno narediti aproksimacijo tako, da za vsoto pripadnosti $(\mu_{ik}^{(r+1)})^\eta$ starih vzorcev vzamemo stare pripadnosti in jih utežimo s pozabljanjem γ_v ($0 < \gamma_v \leq 1$):

$$s_i^{(r)} = \sum_{k=1}^n (\mu_{ik}^{(r)})^\eta, \quad (4.11)$$

$$s_i^{(r+1)} = \gamma_v s_i^{(r)} + \left(\mu_{ik+1}^{(r+1)} \right)^\eta. \quad (4.12)$$

Enačbo za računanje spremembe pozicije centrov i -tega roja zapišemo kot:

$$\Delta \mathbf{v}_i^{(r+1)} = \frac{\left(\mu_{ik+1}^{(r+1)} \right)^\eta \left(\mathbf{x}(k+1) - \mathbf{v}_i^{(r)} \right)}{s_i^{(r+1)}}. \quad (4.13)$$

4.1.1 Primerjava sprotne identifikacije in klasične c -means identifikacije rojev

Z enačbami, izpeljanimi v prejšnjem poglavju, smo dobili osnovo za sprotno prilagajanje centrov rojev po c -means metodi. Čeprav smo pri izpeljavi predvidevali, da izhajamo iz že poznanega mehkega modela, se izkaže, da lahko s to metodo začnemo tudi graditi mehki model oziroma deliti vhodno-izhodni prostor.

Rekurzivno metodo smo primerjali s klasično FCM metodo na primeru dveh dvodimenzionalnih rojev. Oba roja sta bila generirana z matlabovo funkcijo *randn*, ki naključno generira števila iz normalne porazdelitve z varianco ena. Center prvega roja je bil v točki $[0 \ 0]$, center drugega pa v točki $[5 \ 5]$. Roja sta imela po 1000 vzorcev. Povprečenje generiranih podatkov je dalo centre rojev v točki $[-0,0365 \ 0,0329]$ in $[5,0383 \ 5,0273]$. Časovno gledano najprej nastopi prvi roj in nato drugi. Na tem mestu velja poudariti, da smo smatrali, da je ena iteracija *off-line* metode enaka enemu preletu skozi podatke rekurzivne metode.

S testom smo skušali ugotoviti, kako dobro se centri rekurzivne metode skladajo s centri klasične FCM metode. Pri rekurzivni identifikaciji je načeloma inicializacija začetnih pozicij centrov naključna. V našem primeru smo zaradi ponovljivosti rezultatov izbrali dve fiksni točki, ki sta relativno blizu: $\mathbf{v}_1(0) = [-2 \ -2]^T$ in $\mathbf{v}_2(0) = [-1,9 \ -1,9]^T$. Tabela 4.1 prikazuje pozicije rojev skozi iteracije pri naključno pomešanem časovnem zaporedju rojev, tabela 4.2 pa pri časovnem zaporedju, kjer nastopijo najprej vzorci prvega in nato drugega roja.

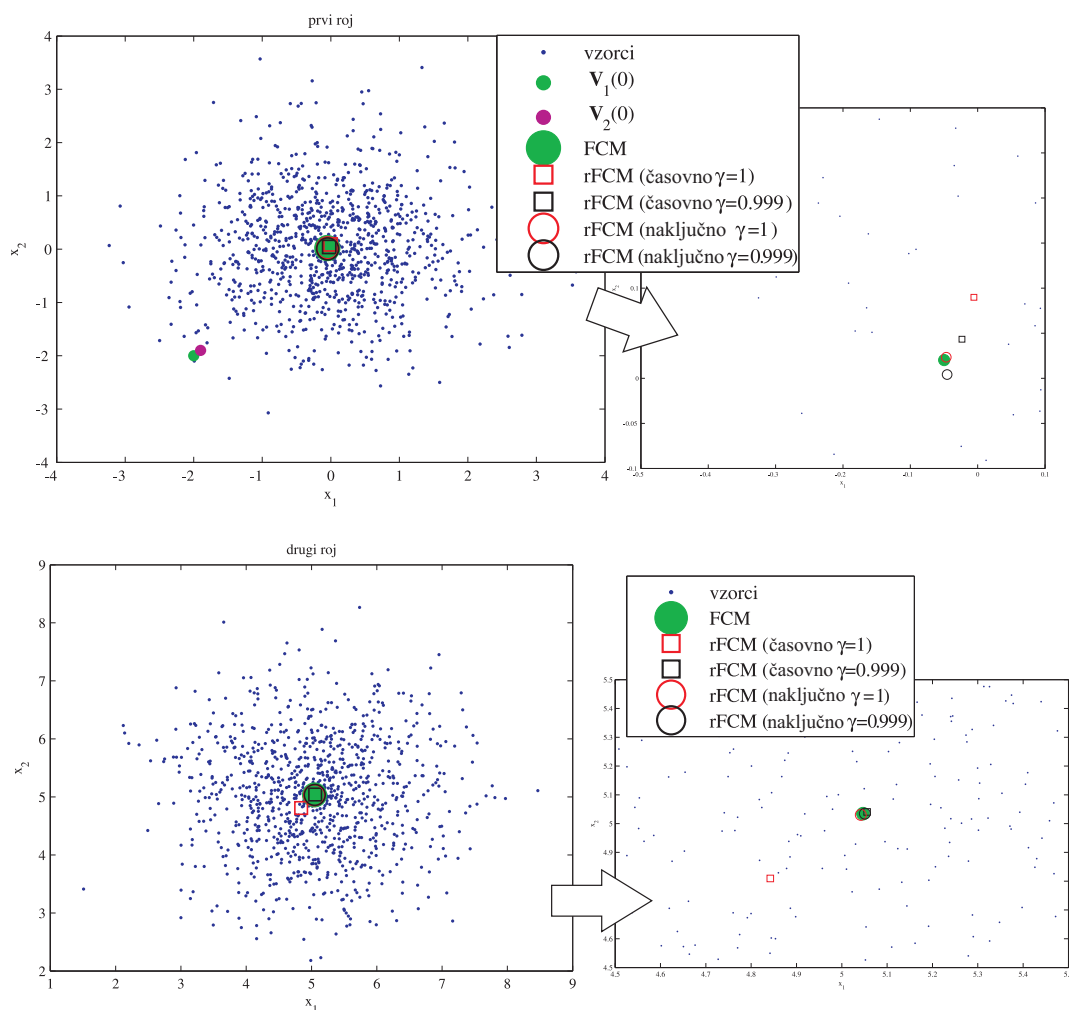
Na sliki 4.1 so prikazani začetni centri rojev in končni centri rojev. Vidimo, da se pri časovno naključnem zaporedju vzorcev centri klasične FCM metode precej dobro ujemajo s centri, dobljenimi z izpeljano rekurzivno verzijo, kjer ni nastavljenega pozabljanja. V primeru, ko nimamo časovno naključno premešanih vzorcev, izpeljana metoda deluje boljše, če imamo nastavljeno pozabljanje. V našem primeru je bil faktor pozabljanja nastavljen tako, da je bilo okno pozablja-

Iteracija		1	2	3	4	5	6	7	8	9	10
FCM	v_{11}	2,52	2,636	3,02	4,083	4,937	5,04	5,046	5,047	5,047	5,047
	v_{12}	2,582	2,677	3,043	4,08	4,923	5,027	5,034	5,034	5,034	5,034
	v_{21}	2,485	2,36	1,976	0,912	0,059	-0,043	-0,049	-0,05	-0,05	-0,05
	v_{22}	2,461	2,379	2,012	0,97	0,129	0,027	0,021	0,02	0,02	0,02
rFCM, $\gamma_v = 1$	v_{11}	-0,022	-0,035	-0,04	-0,042	-0,044	-0,045	-0,045	-0,046	-0,046	-0,047
	v_{12}	0,047	0,034	0,03	0,028	0,026	0,025	0,025	0,024	0,024	0,023
	v_{21}	5,005	5,025	5,032	5,035	5,037	5,039	5,04	5,041	5,041	5,042
	v_{22}	4,999	5,016	5,022	5,025	5,027	5,028	5,029	5,029	5,03	5,03
rFCM, $\gamma_v = 0,999$	v_{11}	-0,036	-0,044	-0,045	-0,045	-0,045	-0,045	-0,045	-0,045	-0,045	-0,045
	v_{12}	0,013	0,005	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004
	v_{21}	5,036	5,049	5,051	5,051	5,051	5,051	5,051	5,051	5,051	5,051
	v_{22}	5,023	5,033	5,034	5,035	5,035	5,035	5,035	5,035	5,035	5,035

Tabela 4.1: Pozicije rojev skozi iteracije pri časovno naključnem zaporedju vzorcev.

Iteracija		1	2	3	4	5	6	7	8	9	10
FCM	v_{11}	2,52	2,636	3,02	4,083	4,937	5,04	5,046	5,047	5,047	5,047
	v_{12}	2,582	2,677	3,043	4,08	4,923	5,027	5,034	5,034	5,034	5,034
	v_{21}	2,485	2,36	1,976	0,912	0,059	-0,043	-0,049	-0,05	-0,05	-0,05
	v_{22}	2,461	2,379	2,012	0,97	0,129	0,027	0,021	0,02	0,02	0,02
rFCM, $\gamma_v = 1$	v_{11}	0,669	0,237	0,124	0,074	0,046	0,028	0,016	0,007	0	-0,006
	v_{12}	1,096	0,451	0,283	0,209	0,167	0,14	0,122	0,108	0,098	0,09
	v_{21}	3,243	4,073	4,392	4,553	4,649	4,713	4,759	4,794	4,821	4,842
	v_{22}	3,062	3,968	4,316	4,492	4,598	4,668	4,719	4,756	4,786	4,81
rFCM, $\gamma_v = 0,999$	v_{11}	0,922	0,04	-0,016	-0,023	-0,023	-0,023	-0,023	-0,023	-0,023	-0,023
	v_{12}	1,344	0,137	0,055	0,045	0,044	0,043	0,043	0,043	0,043	0,043
	v_{21}	4,333	4,963	5,042	5,054	5,055	5,056	5,056	5,056	5,056	5,056
	v_{22}	4,237	4,937	5,025	5,038	5,04	5,041	5,041	5,041	5,041	5,041

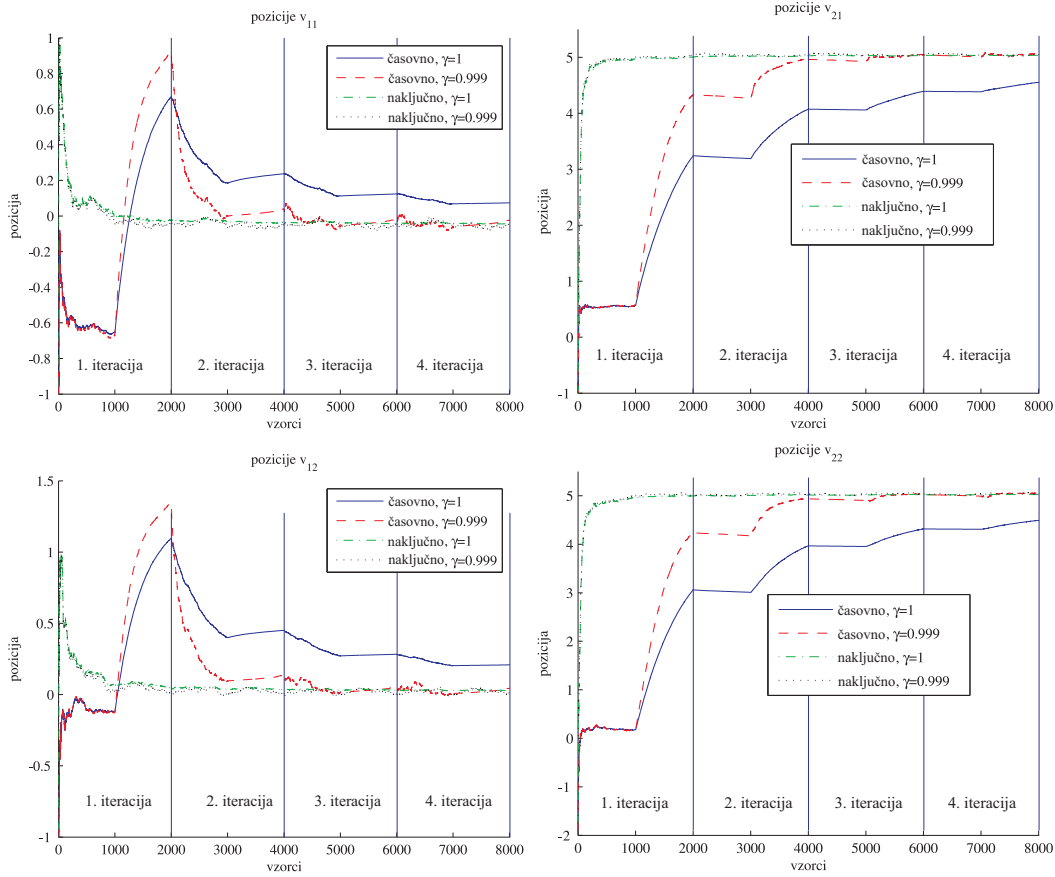
Tabela 4.2: Pozicije rojev skozi iteracije pri časovnem zaporedju vzorcev, kjer najprej nastopijo vzorci prvega in nato drugega roja.



Slika 4.1: Prikaz centrov v podatkovnem prostoru. Zgoraj je prvi roj spodaj drugi roj.

nja približno 2000 vzorcev. Na sliki 4.2 lahko vidimo časovne poteke posameznih komponent centrov rojev.

Pri časovno naključnem zaporedju se centra obeh rojev precej hitro postavita na svoje pozicije. Pri časovno urejenem zaporedju se oba centra postavita najprej v prvi roj. Ko nastopijo podatki za drugi roj, se enden izmed njih (tisti, ki je bližje novim vzorcem) začne pomikati proti centru drugega roja. V primeru, da ima drugi roj precej več vzorcev kot prvi, bi se tudi preostali center začel pomikati proti drugemu roju. Temu se lahko izognemo s selektivnim prilagajanjem centrov tako, da popravimo pozicije le tistim centrom/rojem, ki imajo proženje večje od



Slika 4.2: Časovni prikaz pozicij centrov rojev pri rekurzivnem rojenju.

določenega pragu oziroma tistim, ki imajo največje proženje [150],[110].

Iz časovnih potekov pozicij centrov lahko vidimo, da se le-ti premikajo z neko časovno konstanto. Enačbo 4.9 lahko obravnavamo kot diferenčno enačbo prvega reda, kjer je vhod $\mathbf{x}(k+1)$. Če opustimo pisanje iteracij, jo lahko zapišemo kot:

$$\mathbf{v}_i(k+1) = \left(1 - \frac{\mu_{ik+1}^\eta}{s_i(k+1)}\right) \mathbf{v}_i(k) + \frac{\mu_{ik+1}^\eta}{s_i(k+1)} \mathbf{x}(k+1). \quad (4.14)$$

Člen $\left(1 - \frac{\mu_{ik+1}^\eta}{s_i(k+1)}\right)$ nam definira pol sistema. Pripadnost trenutnega vzorca i -temu roju μ_{ik+1}^η je vedno manjša ali enaka ena oziroma večja ali enaka nič. Vsota pripadnosti s_i je vedno večja od trenutne pripadnosti, razen pri prvem vzorcu (ker jo ponavadi inicializiramo na ena). Zgoraj omenjeni člen bo vedno manjši od ena. Kar pomeni, da bo pol znotraj enotine krožnice. Iz tega lahko sklepamo, da je enačba za adaptacijo centrov stabilna. Ko gre število vzorcev v neskončnost, gre razmerje med trenutno pripadnostjo in vsoto pripadnosti proti nič. Pol potuje

proti enotini krožnici. Enačba pridobiva integrirni značaj, hkrati pa novi vzorci izgubljajo vpliv na pozicijo centra, saj je vpliv vzorca skaliran z istim razmerjem $\frac{\mu_{ik+1}^\eta}{s_i(k+1)}$, ki povzroči potovanje pola enačbe proti enotini krožnici. To pomeni, da se centri ustalijo. To velja, če je faktor pozabljanja za vsoto preteklih pripadnosti nastavljen na ena. Iz enačbe 4.14 lahko vidimo, da lahko s pomočjo faktorja pozabljanja vplivamo na samo hitrost prilagajanja centrov. Na začetno hitrost prilagajanja pa lahko vplivamo z inicializacijo začetnih vrednosti vsote pripadnosti s_i .

Na tem mestu velja opozoriti, da rekurzivna metoda ne zagotavlja enakih pozicij centrov kot *off-line* metoda. V praksi se izkaže, da so odstopanja med pozicijami majhna, predvsem v primeru, ko centre rekurzivne metode inicializiramo blizu dejanskim centrom in pri časovno naključnem zaporedju vzorcev.

4.1.2 Rekurzivno računanje mehke kovariančne matrike

Roji so definirani s centri in s pripadnostnimi funkcijami, ki opisujejo porazdeljenost vzorcev/podatkov okoli centrov. Načeloma bi lahko uporabili pripadnostno funkcijo, definirano z enačbo 4.2, vendar se v praksi izkaže, da te funkcije niso najboljše za uporabo v Takagi-Sugeno mehkem modelu, saj so prehodi med lokalnimi modeli nekoliko trši. Zato se največkrat uporabljajo radialne 4.15 ali pa eliptične 4.17 (ponavadi Gaussove) pripadnostne funkcije [128], [2], ki zagotavljajo mehkejši prehode med lokalnimi modeli.

$$\mu_{ik} = e^{-\frac{d_{ik}^2}{2\eta\sigma^2}}, \quad (4.15)$$

$$d_{ik}^2 = (\mathbf{x}_k - \mathbf{v}_{i_k})^T (\mathbf{x}_k - \mathbf{v}_{i_k}), \quad (4.16)$$

$$\mu_i = e^{-\frac{D_i^2}{2\eta}}, \quad (4.17)$$

$$D_i^2 = (\mathbf{x} - \mathbf{v}_i)^T \mathbf{F}_i^{-1} (\mathbf{x} - \mathbf{v}_i). \quad (4.18)$$

Za določitev teh pripadnostnih funkcij moramo rekurzivno računati tudi porazdelitev podatkov okoli centrov. Porazdelitev podatkov okoli centrov opisuje mehka kovariančna matrika [2]. Pri izpeljavi rekurzivne enačbe zanjo postopamo podobno kot pri izpeljavi za centre. Najprej zapišemo enačbo za naslednjo itera-

cijo:

$$\mathbf{F}_i^{(r+1)} = \frac{\sum_{k=1}^{n+1} \left(\mu_{ik}^{(r+1)} \right)^\eta \left(\mathbf{x}(k) - \mathbf{v}_i^{(r+1)} \right) \left(\mathbf{x}(k) - \mathbf{v}_i^{(r+1)} \right)^T}{\sum_{k=1}^{n+1} \left(\mu_{ik}^{(r+1)} \right)^\eta}. \quad (4.19)$$

Enačbo 4.19 lahko razdelimo na dva dela:

$$\begin{aligned} \mathbf{F}_i^{(r+1)} &= \frac{\sum_{k=1}^n \left(\mu_{ik}^{(r+1)} \right)^\eta \left(\mathbf{x}(k) - \mathbf{v}_i^{(r+1)} \right) \left(\mathbf{x}(k) - \mathbf{v}_i^{(r+1)} \right)^T}{\sum_{k=1}^{n+1} \left(\mu_{ik}^{(r+1)} \right)^\eta} + \\ &+ \frac{\left(\mu_{ik+1}^{(r+1)} \right)^\eta \left(\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)} \right) \left(\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)} \right)^T}{\sum_{k=1}^{n+1} \left(\mu_{ik}^{(r+1)} \right)^\eta}. \end{aligned} \quad (4.20)$$

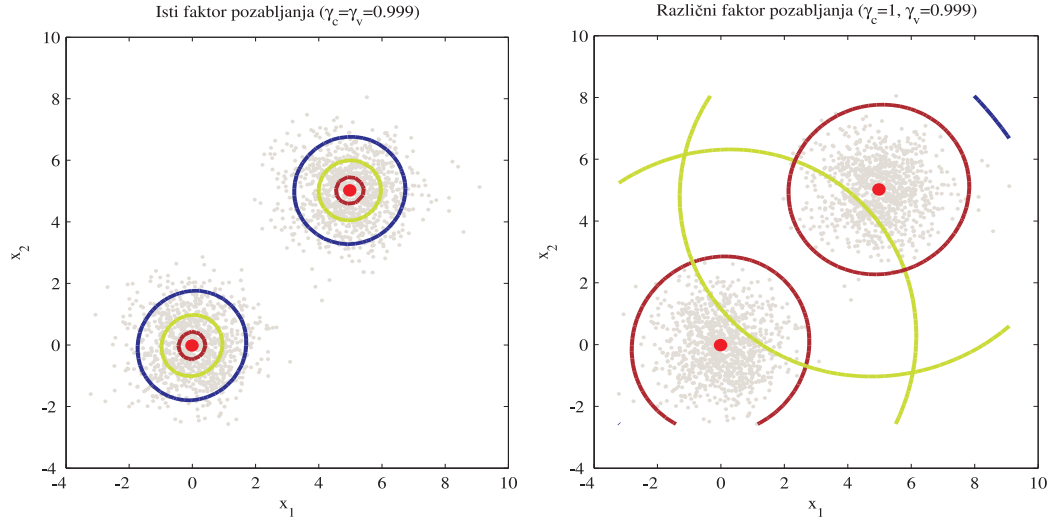
Imenovalec in števec prvega člena enačbe 4.20 delimo z $\sum_{k=1}^n \left(\mu_{ik}^{(r+1)} \right)^\eta$. Če naredimo podobne predpostavke in aproksimacije kot pri izpeljavi centrov ter upoštevamo enačbi 4.11 in 4.12, dobimo:

$$\mathbf{F}_i^{(r+1)} = \gamma_c \frac{s_i^{(r)}}{s_i^{(r+1)}} \mathbf{F}_i^{(r)} + \frac{\left(\mu_{ik+1}^{(r+1)} \right)^\eta}{s_i^{(r+1)}} \left(\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)} \right) \left(\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)} \right)^T. \quad (4.21)$$

Podobno kot pri enačbi 4.12, smo tudi tukaj dodali faktor pozabljanja γ_c . Običajno je faktor pozabljanja γ_c nastavljen na isto vrednost kot γ_v . Če je γ_c večji kot γ_v , dobimo večje mehke kovariančne matrike, ki ponavadi ne odražajo dejanske porazdelitve vzorcev okoli centrov. Slika 4.3 prikazuje primer, ko sta faktorja pozabljanja različna. Mehke kovariančne matrike so bile izračunane za primer, predstavljenem v razdelku 4.1.1. Iz slike 4.3 je razvidno, da pri istih faktorjih pozabljanja ($\gamma_v = \gamma_c = 0,999$) dobimo precej dober opis porazdelitve podatkov okoli centrov (slika 4.3 levo). Prav tako dobimo podoben rezultat, če oba faktorja pozabljanja nastavimo na ena (torej brez pozabljanja). V primeru, ko sta faktorja pozabljanja različna ($\gamma_v = 0,999$, $\gamma_c = 1$), vidimo da je kovariančna matrika precej prevelika (slika 4.3 desno).

Za računanje Mahalanobisove norme potrebujemo tudi inverz mehke kovariančne matrike in determinanto. Inverz mehke kovariančne matrike izpeljemo s pomočjo Woodbury matrix identity lemme (en. 4.22) [74],[3]:

$$(\mathbf{A} + \mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}. \quad (4.22)$$



Slika 4.3: Prikaz vpliva različnih faktorjev pozabljanja na izračun kovariančne matrike. Narisane so vrednosti pripadnosti 0,9-rjava, 0,6-zelena in 0,2-modra.

Za naš primer vzamemo:

$$\begin{aligned}
 \mathbf{A} &= \gamma_c \frac{s_i^{(r)}}{s_i^{(r+1)}} \mathbf{F}_i^{(r)}, \\
 \mathbf{U} &= \frac{\left(\mu_{ik+1}^{(r+1)}\right)^\eta}{s_i^{(r+1)}} \left(\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}\right), \\
 \mathbf{V} &= \left(\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}\right)^T, \\
 \mathbf{C} &= 1.
 \end{aligned} \tag{4.23}$$

Dobimo izraz za inverz matrike $\mathbf{F}_i^{(r+1)}$:

$$\begin{aligned}
 \left[\mathbf{F}_i^{(r+1)}\right]^{-1} &= \left[\gamma_c \frac{s_i^{(r)}}{s_i^{(r+1)}} \mathbf{F}_i^{(r)}\right]^{-1} - \left[\gamma_c \frac{s_i^{(r)}}{s_i^{(r+1)}} \mathbf{F}_i^{(r)}\right]^{-1} \frac{\left(\mu_{ik+1}^{(r+1)}\right)^\eta}{s_i^{(r+1)}} \left(\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}\right) \cdot \\
 &\cdot \left[1^{-1} + \left(\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}\right)^T \left[\gamma_c \frac{s_i^{(r)}}{s_i^{(r+1)}} \mathbf{F}_i^{(r)}\right]^{-1} \frac{\left(\mu_{ik+1}^{(r+1)}\right)^\eta}{s_i^{(r+1)}} \left(\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}\right)\right]^{-1} \cdot \\
 &\cdot \left(\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}\right)^T \left[\gamma_c \frac{s_i^{(r)}}{s_i^{(r+1)}} \mathbf{F}_i^{(r)}\right]^{-1},
 \end{aligned} \tag{4.24}$$

ki ga lahko poenostavimo. Najprej obrnemo inverze in skalar damo v ulomek:

$$\begin{aligned} \left[\mathbf{F}_i^{(r+1)} \right]^{-1} &= \frac{1}{\gamma_c} \frac{s_i^{(r+1)}}{s_i^{(r)}} \left[\mathbf{F}_i^{(r)} \right]^{-1} - \\ &= \frac{\frac{1}{\gamma_c} \frac{s_i^{(r+1)}}{s_i^{(r)}} \left[\mathbf{F}_i^{(r)} \right]^{-1} \frac{(\mu_{ik+1}^{(r+1)})^\eta}{s_i^{(r+1)}} (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}) (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)})^T \frac{1}{\gamma_c} \frac{s_i^{(r+1)}}{s_i^{(r)}} \left[\mathbf{F}_i^{(r)} \right]^{-1}}{1 + (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)})^T \frac{1}{\gamma_c} \frac{s_i^{(r+1)}}{s_i^{(r)}} \left[\mathbf{F}_i^{(r)} \right]^{-1} \frac{(\mu_{ik+1}^{(r+1)})^\eta}{s_i^{(r+1)}} (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)})} \end{aligned} \quad (4.25)$$

S krajšanjem dobimo slednjo enačbo za rekurzivno računanje inverza mehke kovariančne matrike:

$$\left[\mathbf{F}_i^{(r+1)} \right]^{-1} = \frac{1}{\gamma_c} \frac{s_i^{(r+1)}}{s_i^{(r)}} \left[\left[\mathbf{F}_i^{(r)} \right]^{-1} - \frac{\left[\mathbf{F}_i^{(r)} \right]^{-1} (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}) (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)})^T \left[\mathbf{F}_i^{(r)} \right]^{-1}}{\gamma_c \frac{s_i^{(r)}}{(\mu_{ik+1}^{(r+1)})^\eta} + (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)})^T \left[\mathbf{F}_i^{(r)} \right]^{-1} (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)})} \right]. \quad (4.26)$$

Naslednji korak je izpeljava rekurzivnega računanja determinante za mehko kovariančno matriko. Pri tem upoštevamo lemo o determinanti matrike:

$$\det(\mathbf{A} + \mathbf{u}\mathbf{v}^T) = (1 + \mathbf{v}^T \mathbf{A}^{-1} \mathbf{u}) \det(\mathbf{A}). \quad (4.27)$$

Nastavimo enačbo za determinanto mehke kovariančne matrike:

$$\det(\mathbf{F}_i^{(r+1)}) = \det \left(\gamma_c \frac{s_i^{(r)}}{s_i^{(r+1)}} \mathbf{F}_i^{(r)} + \frac{(\mu_{ik+1}^{(r+1)})^\eta}{s_i^{(r+1)}} (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}) (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)})^T \right), \quad (4.28)$$

z upoštevanjem:

$$\begin{aligned} \mathbf{A} &= \gamma_c \frac{s_i^{(r)}}{s_i^{(r+1)}} \mathbf{F}_i^{(r)}, \\ \mathbf{u} &= \frac{(\mu_{ik+1}^{(r+1)})^\eta}{s_i^{(r+1)}} (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}), \\ \mathbf{v} &= (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}) \end{aligned} \quad (4.29)$$

in enačbe 4.27 dobimo:

$$\det(\mathbf{F}_i^{(r+1)}) = \det \left(\gamma_c \frac{s_i^{(r)}}{s_i^{(r+1)}} \mathbf{F}_i^{(r)} \right) \left(1 + (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)})^T \frac{1}{\gamma_c} \frac{s_i^{(r+1)}}{s_i^{(r)}} \left[\mathbf{F}_i^{(r)} \right]^{-1} \frac{(\mu_{ik+1}^{(r+1)})^\eta}{s_i^{(r+1)}} (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}) \right). \quad (4.30)$$

Če preuredimo zgornjo enačbo in upoštevamo lastnost determinante matrike $\det(c\mathbf{A}_{n \times n}) = c^n \det(\mathbf{A}_{n \times n})$, lahko zapišemo enačbo za rekurzivno računanje determinante v naslednji obliki:

$$\det(\mathbf{F}_i^{(r+1)}) = \left(\gamma_c \frac{s_i^{(r)}}{s_i^{(r+1)}} \right)^{z+1} \det(\mathbf{F}_i^{(r)}) \left(1 + \frac{1}{\gamma_c} \frac{(\mu_{ik+1}^{(r+1)})^\eta}{s_i^{(r)}} (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)})^T \left[\mathbf{F}_i^{(r)} \right]^{-1} (\mathbf{x}(k+1) - \mathbf{v}_i^{(r+1)}) \right), \quad (4.31)$$

kjer je $z + 1$ število vrstic oziroma stolpcev matrike $\mathbf{F}_i$.

4.2 Rekurzivna identifikacija lokalnih linearnih modelov

Centri in mehka kovariančna matrika definirajo linearna področja, na katerih veljajo linearni modeli, ki jih identificiramo s pomočjo najmanjših kvadratov. Načeloma bi lahko dobili parametre teh modelov iz mehke kovariančne matrike s pomočjo dekompozicije matrike na singularne vrednosti. Vendar parametri dobljeni na ta način, ne dajo najboljših rezultatov. Zato za oceno uporabimo rekurzivne najmanjše kvadrate. Pri tem je potrebno upoštevati pripadnosti vzorcev k rojem in temu primerno skalirati vhode in izhode za vsak roj. To storimo s pripadnostmi. Za potrebe identifikacije je potrebno najprej izračunati pripadnost vhodne slike k posameznim rojem. Pripadnosti vhodne slike bomo označevali z β_i . Lahko vzamemo pripadnosti, ki smo jih izračunali pri rojenju ali pa uporabimo druge pripadnostne funkcije. V našem primeru smo izbrali Gaussove pripadnostne funkcije, saj se izkaže, da dobimo nekoliko boljše rezultate kot z uporabo pripadnostnih funkcij, ki so uporabljene za rojenje. Pri Gaussovih pripadnostnih funkcijah lahko preprosto uporabimo podatke o centrih in mehki kovariančni matriki, ki smo jih dobili z rojenjem. Uporabimo lahko radialne funkcije, kjer izračunamo pripadnost vsake komponente vhodnega vektroja μ_{ik} . S tem napravimo projekcijo roja na posamezne komponente. Z množenjem pripadnosti po komponentah dobimo pripadnost vhodnega vektroja k roju β_i :

$$\mu_{ik} = e^{-\frac{(x_k - v_{ik})^2}{2\eta_m \mathbf{F}_{ik,k}}} \quad k = 1, 2, \dots, z \quad i = 1, 2, \dots, c, \quad (4.32)$$

$$\beta_i = \prod_{k=1}^z \mu_{ik}, \quad (4.33)$$

kjer je η_m faktor prekrivanja in $\mathbf{F}_{ik,k}$ diagonalni člen mehke kovariančne matrike. Dobljene pripadnosti je na koncu priporočljivo še normirati, čeprav v literaturi najdemo obe različici, tako normirano kot nenormirano. Normirana različica je uporabljena večkrat:

$$\beta_i = \frac{\beta_i}{\sum_{k=1}^c \beta_k} \quad i = 1, 2, \dots, c. \quad (4.34)$$

Bolj natančno vrednost pripadnosti dobimo z uporabo elipsoidnih pripadnostnih funkcij:

$$\beta_i = e^{\frac{-D_{ik}^2}{2\eta_m}} \quad i = 1, 2, \dots, c, \quad (4.35)$$

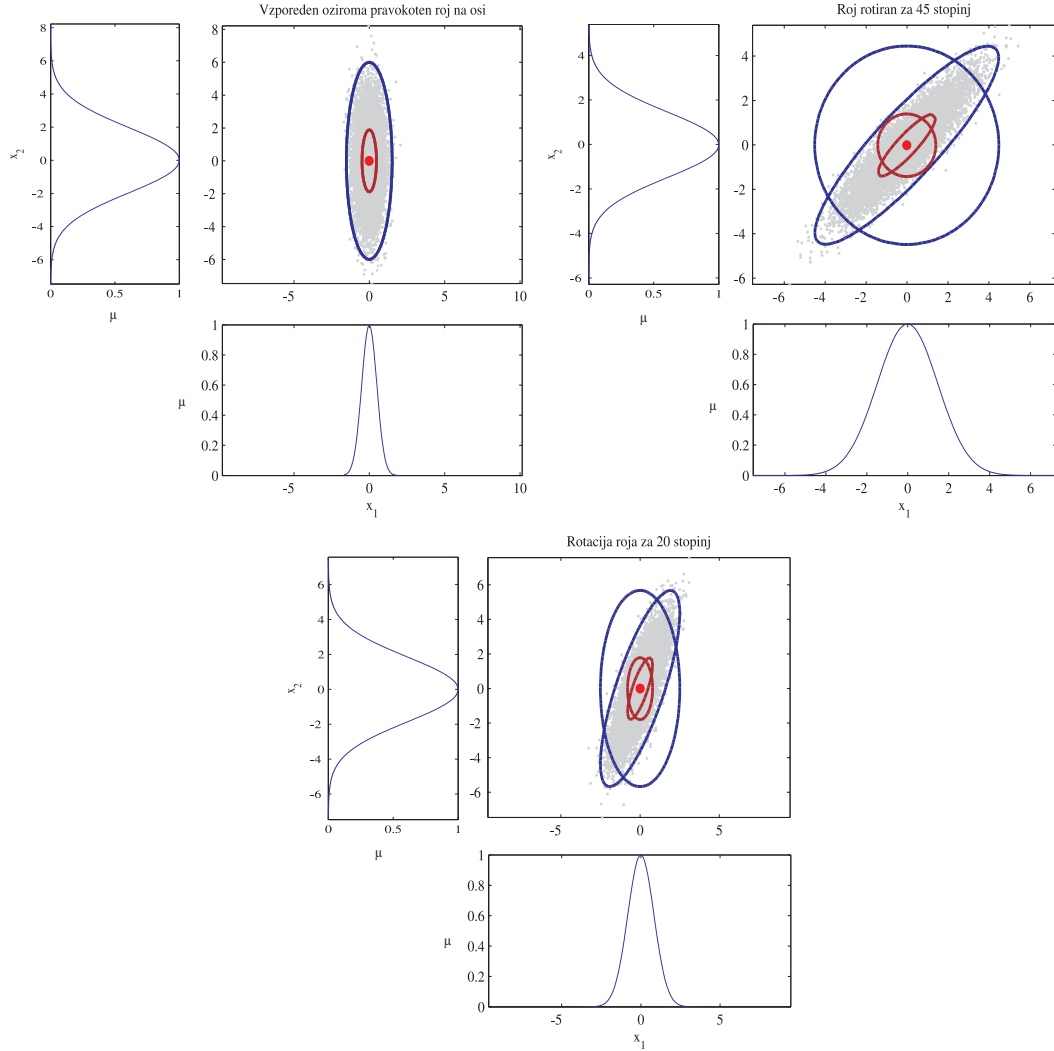
kjer je D_{ik} Mahalanobisova razdalja (en. 3.6 in en. 3.7) oziroma razdalja računana po enačbi 4.36.

$$D_{ik}^2 = (\mathbf{x}(k) - \mathbf{v}_i)^T \mathbf{F}_i^{-1} (\mathbf{x}(k) - \mathbf{v}_i). \quad (4.36)$$

Pri tem je potrebno upoštevati, da za razliko od rojenja tu ne računamo pripadnosti za vhodno-izhodnem prostoru, ampak le na vhodnem. Dolžina vektorja $\mathbf{x}(k)$ je z in ne $z+1$, kot pri rojenju. Velikost mehke kovariančne matrike je $(z+1) \times (z+1)$. Problem različnih dimenzij vektorja in matrike lahko rešimo na dva načina. Pri prvem poleg rojenja na vhodno-izhodnem prostoru lahko vzporedno delamo še rojenje na vhodnem prostoru. Tako dobimo vhodno mehko kovariančno matriko, ki jo potem uporabimo za računanje pripadnosti pri identifikaciji parametrov lokalnih modelov. S tem se poveča računski čas metode in potreben spomin. Pri drugem načinu vhodnemu vektorju dodamo še eno komponento, ki predstavlja izhod procesa. Vrednost te komponente je enaka zadnji komponenti centra roja k kateremu računamo pripadnost. Enak rezultat dobimo, če odvezamemo vektorju centrov zadnjo komponento, ki predstavlja pozicijo centra na izhodnem prostoru, in zadnji stolpec ter zadnjo vrsto vhodno-izhodne mehke kovariančne matrike. Na tem mestu je smiselno poudariti, da vhodna mehka kovariančna matrika v praksi ni vedno podmatrika vhodno-izhodne mehke kovariančne matrike, kot bi lahko logično sklepali. Razlog tiči v tem, da so pripadnosti, računanane samo na vhodnem prostoru malenkost različne od pripadnosti na vhodno-izhodnem prostoru.

Računanje pripadnosti z radialnimi funkcijami je zelo podobno računanju z eliptičnimi, le da namesto celotne mehke kovariančne matrike upoštevamo le njene diagonalne člene. Problem takega računanja je, da ne upoštevamo rotacije roja v prostoru [2]. Pravilne pripadnosti dobimo v primeru, ko je roj vzporeden oziroma pravokoten na osi. Razlika med pripadnostmi na je prikazana na sliki 4.4. Viden je roj podatkov (sive pike) ter elipse (izočrte), ki označujejo pripadnost 0,63 (rdeča barva) in pripadnost 0,01 (modra barva). Pri vsakem prikazanem roju so levo in spodaj prikazane še vrednosti pripadnostne funkcije 4.32. Izočrte, ki se dobro skladajo z roji, so izračunane z elipsoidnimi pripadnostnimi funkcijami (en. 4.36), tiste ki se slabše skladajo pa z radialnimi (en. 4.32, 4.33). Iste izočrte dobimo le pri rojih, ki so vzporedni oziroma pravokotni na osi. Kljub temu pa pri mehkem Takagi-Sugeno modelu ni opaziti bistvene razlike v natančnosti modela pri uporabi elipsoidnih pripadnostnih funkcij in eliptičnih pripadnostnih funkcij.

S pomočjo enačbe 4.37 lahko ustrezno razdelimo vrednosti vhodov in izhodov med



Slika 4.4: Različne postavitve rojev. Vzporeden oziroma pravokoten roj na osi (zgoraj levo), roj obrnjen za 45 stopinj glede na x_1 os (zgoraj desno) in roj, obrnjen za 20 stopinj glede na x_1 os (spodaj).

roje. Vsakemu roju pripada vrednost vhoda in izhoda, pomnožena s pripadnostjo.

$$\begin{aligned}\psi_i(k+1) &= \beta_i [x_1(k+1), \dots, x_z(k+1), 1]^T, \\ y_i(k+1) &= \beta_i x_{z+1}(k+1) \quad i = 1, 2, \dots, c.\end{aligned}\tag{4.37}$$

Nato s pomočjo rekurzivnih najmanjših kvadratov (en. 4.38) ocenimo parametre

lokalnih modelov za vsak roj:

$$\begin{aligned}\mathbf{P}_i(k+1) &= \frac{1}{\lambda_r} \left(\mathbf{P}_i(k) - \frac{\mathbf{P}_i(k)\psi_i(k+1)\psi_i^T(k+1)\mathbf{P}_i(k)}{\lambda_r + \psi_i^T(k+1)\mathbf{P}_i(k)\psi_i(k+1)} \right), \\ \theta_i(k+1) &= \theta_i(k) + \mathbf{P}_i(k)\psi_i(k+1) (y_i(k+1) - \psi_i^T(k+1)\theta_i(k)).\end{aligned}\quad (4.38)$$

V literaturi najdemo več modifikacij mehkih najmanjših kvadratov. V [6] je izpeljan globalni in lokalni pristop za oceno parametrov.

Lokalne najmanjše kvadrate izpeljemo s pomočjo minimizacije kriterijske funkcije:

$$J_L = \sum_{i=1}^c (\mathbf{y} - \mathbf{X}^T \theta_i)^T \mathbf{\Lambda}_i (\mathbf{y} - \mathbf{X}^T \theta_i), \quad (4.39)$$

kjer je $\mathbf{\Lambda}_i$ diagonalna matrika, ki vsebuje pripadnosti vhodnih vektorjev v matriki $\mathbf{X}$, $\mathbf{y}$ pa je vektor izhodov. Predpostavljamo, da lokalni modeli niso močno povezani z utežno matriko $\mathbf{\Lambda}_i$, zato lahko zapišemo minimizacijsko funkcijo za vsak roj ločeno:

$$\begin{aligned}J_L &= \sum_{i=1}^c J_{Li}, \\ J_{Li} &= (\mathbf{y} - \mathbf{X}^T \theta_i)^T \mathbf{\Lambda}_i (\mathbf{y} - \mathbf{X}^T \theta_i).\end{aligned}\quad (4.40)$$

Rešitev dobimo z uporabo uteženega psevdo-inverza [80],[143]:

$$\theta_i = (\mathbf{X}^T \mathbf{\Lambda}_i \mathbf{X})^{-1} \mathbf{X}^T \mathbf{\Lambda}_i \mathbf{y} \quad i = 1, \dots, c. \quad (4.41)$$

To lahko zapišemo v rekurzivni obliki in dodamo faktor pozabljanja:

$$\begin{aligned}\mathbf{x}_e(k+1) &= [x_1(k+1), \dots, x_z(k+1), 1]^T, \\ y(k+1) &= x_{z+1}(k+1), \\ \mathbf{P}_i(k+1) &= \frac{1}{\lambda_r} \left(\mathbf{P}_i(k) - \frac{\beta_i \mathbf{P}_i(k) \mathbf{x}_e(k+1) \mathbf{x}_e^T(k+1) \mathbf{P}_i(k)}{\lambda_r + \beta_i \mathbf{x}_e^T(k+1) \mathbf{P}_i(k) \mathbf{x}_e(k+1)} \right), \\ \theta_i(k+1) &= \theta_i(k) + \mathbf{P}_i(k) \beta_i \mathbf{x}_e(k+1) (y(k+1) - \mathbf{x}_e^T(k+1) \theta_i(k)).\end{aligned}\quad (4.42)$$

Globalni najmanjši kvadrati minimizirajo funkcijo:

$$J_G = \sum_{k=1}^n (y(k) - \Psi(k)^T \mathbf{\Theta})^2, \quad (4.43)$$

kjer je $\Psi(k) = [\beta_1 \mathbf{x}_e^T, \beta_2 \mathbf{x}_e^T, \dots, \beta_c \mathbf{x}_e^T]^T$ in $\Theta = [\theta_1^T, \dots, \theta_c^T]^T$. Iz minimizacije sledi enačba za rekurzivno estimacijo parametrov Θ [159], [32]:

$$\begin{aligned} y(k+1) &= x_{z+1}(k+1), \\ \mathbf{P}(k+1) &= \frac{1}{\lambda_r} \left(\mathbf{P}(k) - \frac{\mathbf{P}(k)\Psi(k+1)\Psi^T(k+1)\mathbf{P}(k)}{\lambda_r + \Psi^T(k+1)\mathbf{P}(k)\Psi(k+1)} \right), \\ \Theta(k+1) &= \Theta(k) + \mathbf{P}(k)\Psi(k+1) (y(k+1) - \Psi^T(k+1)\Theta(k)). \end{aligned} \quad (4.44)$$

Podrobno izpeljavo obeh algoritmov si lahko ogledamo v [6].

Kriterijska funkcija 4.43 je globalno optimalna, vendar ne zagotavlja ustreznega lokalnega obnašanja podmodelov [80]. Lokalno ustrezne podmodele dobimo z minimizacijo kriterijske funkcije 4.39 [80], [143]. Obstajajo še nekatere druge izpeljanke najmanjših kvadratov [156], [132], [45].

Podobno kot pri rojenju tudi pri identifikaciji parametrov lokalnih modelov uvedemo faktor pozabljanja λ_r . Ta preprečuje, da bi se kovariančna matrika $\mathbf{P}$ oziroma $\mathbf{P}_i$ zmanjšala in s tem ustavila prilagajanje. Pozorni moramo biti na tako imenovani *burst* efekt. Pri tem algoritmu zaradi slabega vzbujanja najprej “zaspi” (kovariančna matrika se močno poveča), ko pa pride do zadostnega vzbujanja pa algoritem “eksplodira” [101], [54]. Temu se lahko izognemo na več načinov: z uvedbo mrtve cone, s prilagajanjem faktorja pozabljanja glede na vzbujanje, uvedemo lahko algoritem, ki zagotavlja konstantno sled matrike ... Več o metodah zagotavljanja pozabljanja in preprečevanja *burst* efekta si lahko preberemo v [16], [101], [54], [133].

Pri identifikaciji dinamičnih sistemov si je smiselno pomagati tudi s pristopom pomožnih spremenljivk (*instrumental variable*) [22], saj pri prilagajanju parametrov upošteva tudi vrednosti izhoda modela:

$$\begin{aligned} y(k+1) &= x_{z+1}(k+1), \\ \mathbf{P}(k+1) &= \frac{1}{\lambda_r} \left(\mathbf{P}(k) - \frac{\mathbf{P}(k)\Psi_m(k+1)\Psi_m^T(k+1)\mathbf{P}(k)}{\lambda_r + \Psi_m^T(k+1)\mathbf{P}(k)\Psi_m(k+1)} \right), \\ \Theta(k+1) &= \Theta(k) + \mathbf{P}(k)\Psi_m(k+1) (y(k+1) - \Psi_m^T(k+1)\Theta(k)). \end{aligned} \quad (4.45)$$

Zakasnjeni izhodi procesa v vektorju Ψ so zamenjani z zakasnjenimi izhodi modela (vektor Ψ_m).

4.3 O parametrih mehke rekurzivne identifikacije

Zgoraj izpeljano metodo lahko zapišemo v sledečih korakih:

- inicializacija parametrov $(s_i(0), v_i(0), F_i(0), P_i(0), \theta_i(0), c, \gamma_c, \gamma_v, \lambda_r, \eta, \eta_m)$,

ZANKA:

- izračun pripadnosti na vhodno-izhodnem prostoru (en. 4.2),
- izračun nove vsote preteklih pripadnosti (en. 4.12),
- adaptacija centrov (en. 4.13, in en. 4.4),
- adaptacija mehke kovariančne matrike (en. 4.21),
- izračun inverza mehke kovariančne matrike (en. 4.26) in adaptacija determinante (en. 4.31),
- izračun pripadnosti na vhodnem prostoru (en. 4.32, en. 4.33 in en. 4.34 ali en. 4.35 in en. 4.34),
- adaptacija parametrov linearnih lokalnih modelov (en. 4.38 ali en. 4.42 ali en. 4.44)
- čakanje na nov vzorec in ponovitev korakov od ZANKA naprej.

V primeru, da je metoda uporabljena za adaptacijo že zgrajenega mehkega modela, je potrebno nastaviti le faktorje pozabljanja. V nasprotnem primeru je potrebno nastaviti šest parametrov $(c, \gamma_c, \gamma_v, \lambda_r, \eta, \eta_m)$ in pet inicializacijskih vrednosti $(s_i(0), \mathbf{v}_i(0), \mathbf{F}_i(0), \mathbf{P}_i(0), \theta_i(0))$.

Problem mehkih metod identifikacije se kaže tudi v “mehkosti” pravil za nastavljanje parametrov. Tako za razvito metodo kot tudi za druge metode ni enolično določenih vrednosti ali pravil za nastavljanje parametrov metode. Obstajajo le priporočila. Ta so običajno podana na podlagi izkušenj in opazovanj razvijalcev metod in ne zagotavljajo optimalnega rezultata.

Začetna vrednost kovaraiančne matrike najmanjših kvadratov se običajno inicializira kot enotina matrika pomnožena z neko konstanto:

$$\mathbf{P} = \alpha \mathbf{I}_{(z+1) \times (z+1)}, \quad (4.46)$$

kjer je $\alpha \gg 1$. Vrednost α variira nekje od 10 do 1000 [16],[101],[54],[133]. Začetne vrednosti parametrov lokalnih modelov θ postavimo na nič, razen v primeru, ko imamo opravka s časovnim dodajanjem rojev. Takrat za prvi roj nastavimo parametre θ na nič, ostale pa nastavimo glede na pripadnost novega roja k ostalim (uteženo povprečje).

Predstavljena metoda v osnovi predpostavlja fiksno, vnaprej določeno število rojev. Za določitev števila rojev ni točnih pravil. Pri tem je uporabnik prepuščen svoji lastni intuiciji. Načeloma velja, da z večjim številom rojev lahko dosežemo večjo natančnost aproksimacije. Vendar je potrebno biti pri tem pozoren, saj preveliko število rojev lahko povzroči tako imenovani *overfitting* oziroma preveliko prilagajanje modela na učno množico. Takrat začnemo modelirati šum. Napaka na učni množici podatkov se sicer zmanjša, na validacijski množici pa se poveča. Število rojev se da dovolj dobro oceniti s pomočjo *scatter* točkovnih diagramov. Lahko pa uporabimo tudi druge, bolj matematične metode.

Za določanje začetnih vrednosti centrov imamo več možnosti. Lahko jih porazdelimo enakomerno ali naključno po prostoru ali pa jih dodajamo na podlagi časovne zakasnitve d . Pri slednji možnosti za prvi center vzamemo prvi podatkovni vektor ($\mathbf{x}(1)$), za naslednji center podatkovni vektor, ki je prisoten po preteku zakasnitve ($\mathbf{x}(1+d)$) in tako naprej, dokler ne inicializiramo vseh rojev. Tak način inicializacije je lahko uporaben predvsem pri identifikaciji nelinearnega procesa, kjer imamo vnaprej določene delovne točke.

Mehko kovariančno matriko lahko inicializiramo z diagonalno matriko, ki odraža razpon vhodno-izhodnega prostora. Ocenimo maksimalno in minimalno vrednost določenega vhoda in izhoda ter ju med seboj odštejemo (en. 4.47). Razpon delimo s številom rojev, da dobimo vplivno cono oziroma polmer roja (en. 4.48). Varianco roja v eni dimenziji določimo tako, da je proženje izven vplivne cone roja manjše od $\epsilon_\beta = 0,01$.

$$d_{max_j} = \max(x_j) - \min(x_j), \quad j = 1, \dots, z + 1, \quad (4.47)$$

$$d_{vpliv_j} = \frac{d_{max_j}}{2c}, \quad j = 1, \dots, z + 1, \quad (4.48)$$

$$\begin{aligned}
\epsilon_\beta &= e^{-\frac{d_{vpliv_j}^2}{2\eta_m\sigma_j^2}}, \\
\ln(\epsilon_\beta) &= -\frac{d_{vpliv_j}^2}{2\eta_m\sigma_j^2}, \\
\sigma_j^2 &= -\frac{d_{vpliv_j}^2}{2\eta_m\ln(\epsilon_\beta)} \quad j = 1, \dots, z + 1.
\end{aligned} \tag{4.49}$$

Z σ_j^2 inicializiramo mehko kovariančno matriko:

$$\mathbf{F}_i = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{z+1}^2 \end{bmatrix}. \tag{4.50}$$

Vsoto preteklih lastnih vrednosti običajno inicializiramo na ena ($s_i(0) = 1$). Z začetno vrednostjo $s_i(0)$ vplivamo na začetno prilagajanje roja. Če hočemo upočasniti začetno prilagajanje centrov in mehke kovariančne matrike, lahko izberemo tudi višjo vrednost, do približno 200. Možna je tudi različna inicializacija $s_i(0)$ za mehko kovariančno matriko in za pozicije centrov. S tem dosežemo različno hitrost prilagajanja centrov in mehke kovariančne matrike.

Faktor mehkosti običajno nastavimo na vrednost $\eta = 2$, faktor prekrivanja pa od 0,25 do 1 ($\eta_m = [0,25 \ 1]$).

Za faktorje pozabljanja smo že ugotovili, da morata biti γ_c in γ_v enaka. Od faktorja pozabljanja je odvisna sposobnost prilagajanja modela. Večje kot je pozabljanje, hitrejšje je prilagajanje modela. Z večanjem pozabljanja se večajo tudi vplivi *outlier*-jev na oceno parametrov. Za določitev ustrezne vrednosti faktorja pozabljanja lahko uporabimo enačbo iz [16]:

$$\lambda = 1 - \frac{2}{N}, \tag{4.51}$$

kjer je λ faktor pozabljanja, N pa število vzorcev, ki vplivajo na oceno parametrov modela.

Vpliv nekaterih parametrov metode si oglejmo na primeru statične nelinearne funkcije z dvema vhodom in enim izhodom:

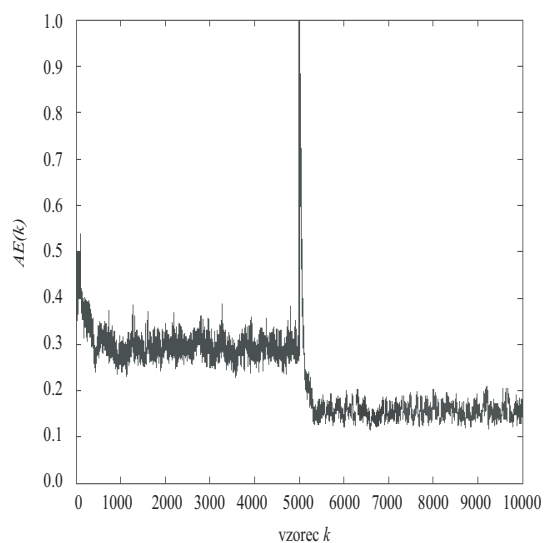
$$\begin{aligned}
z(x(k), y(k)) &= \frac{-10y(k)}{2 + x(k)^2} + 1^2 \tanh(x(k)) \quad k \leq 5000, \\
z(x(k), y(k)) &= \frac{-10y(k)}{5 + x(k)^2} + 2,5^2 \tanh(y(k)) \quad k > 5000.
\end{aligned} \tag{4.52}$$

Celoten eksperiment je trajal 10000 vzorcev. Vhoda $x(k)$ in $y(k)$ sta zavzemala naključne vrednosti iz intervala $[-0,5 \ 2,5]$. Točnost modela je bila evalvirana v vsakem koraku k z absolutno estimacijsko napako definirano kot:

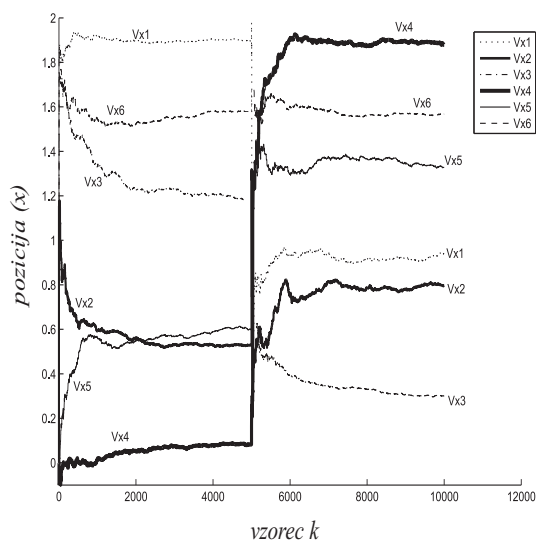
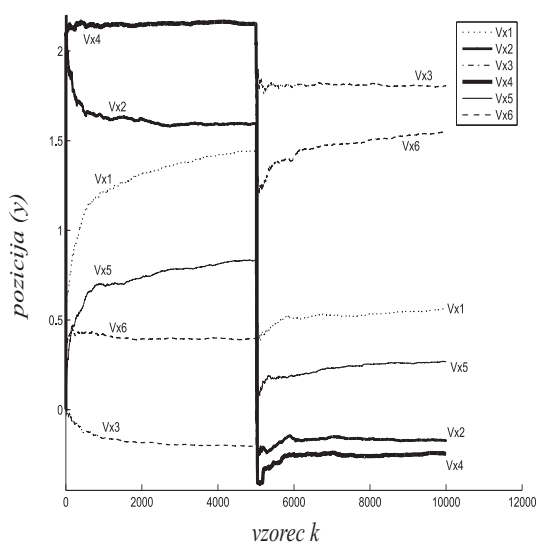
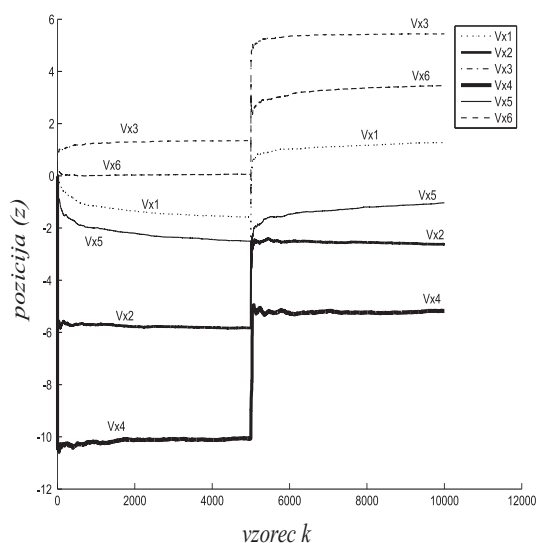
$$AE(k) = \frac{1}{300} \sum_{j=1}^{300} |z(x(j), y(j)) - y_m(x(j), y(j))|, \quad (4.53)$$

kjer je y_m izhod modela, $x(j)$ in $y(j)$ pa sta zopet naključno generirana vhoda z vrednostmi iz intervala $[-0,5 \ 2,5]$. Izbrani so bili sledeči parametri metode: faktor mehkosti $\eta = 2$, faktor prekrivanja $\eta_m = 0,25$, računanje pripadnosti za identifikacijo lokalnih modelov z radialnimi funkcijami, rojenje z rekurzivno c -means metodo, najmanjši kvadrati podani z enačbo 4.38, podatkovni vektor $\mathbf{x}(k) = [x(k) \ y(k) \ z(k)]^T$ in faktorji potabljanja $\gamma_v = \gamma_c = \lambda_r = 0,9996$, kar pomeni, da imamo približno 5000 vzorcev spomina. Z namenom pohitritve prilagajanja mehkega modela smo implementirali še ponastavljanje kovariančne matrike $\mathbf{P}_i$ [16]. Algoritem ponastavi kovariančno matriko, ko je razlika med izhodom modela in dejanskim izhodom večja kot 1,5. Na sliki 4.5 je prikazana trenutna absolutna napaka, na sliki 4.6 potovanje centrov po x osi, na sliki 4.7 po y osi in na sliki 4.8 po z osi vhodno-izhodnega prostora.

Vpliv faktorjev pozabljanja je prikazan na sliki 4.10. Pri tem eksperimentu smo opustili ponastavljanje kovariančne matrike. Absolutna trenutna napaka je prikazana za faktorje pozabljanja 1, 0,9998, 0,9996, 0,999, 0,995, 0,99 in 0,95 kar ustreza vplivnemu številu vzorcev ∞ , 10000, 5000, 2000, 400, 200 in 40. Obkroženo območje b na sliki prikazuje absolutno napako pri pozabljanju od 1 do 0,999, območje a pa od 0,995 do 0,95. Iz slike je razvidno hitrejše prilagajanje modela pri večjem pozabljanju. Na sliki 4.9 so prikazane variacije v oceni pozicije drugega centra po y komponenti. Opazimo, da se pri večanju pozabljanja večja variacija ocene parametrov. Eksperiment ponovimo še za različne faktorje prekrivanja in mehkosti. Ponovno uvedemo resetiranje kovariančne matrike in faktor pozabljanja 0,9998. Slika 4.11 prikazuje vpliv faktorja mehkosti na hitrost prilagajanja modela in na natančnost modela. Faktor prekrivanja je bil nastavljen na 0,25. Slika 4.12 prikazuje vpliv faktorja prekrivanja. Faktor mehkosti je bil nastavljen na 2. Opazimo lahko, da faktorja nimata bistvenega vpliva na hitrost prilagajanja, je pa od njiju odvisna natančnost modela.

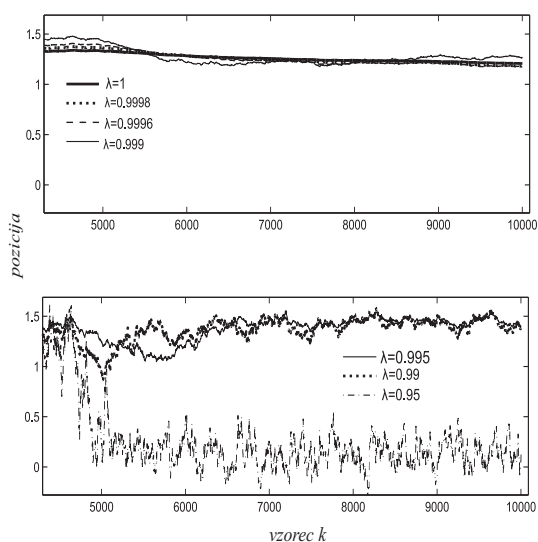


Slika 4.5: Absolutna napaka.

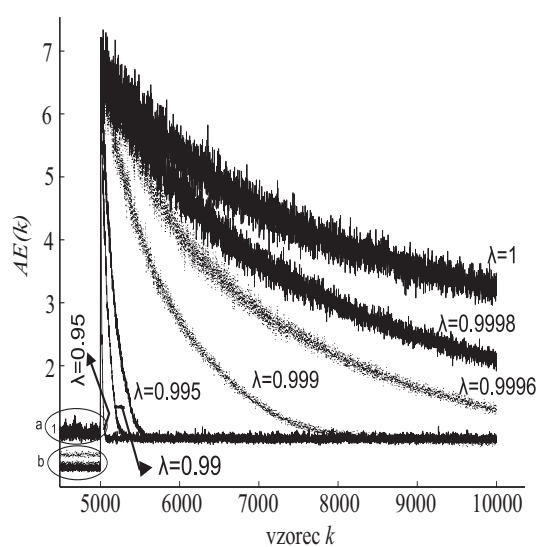
Slika 4.6: Pozicije centrov po x osi.Slika 4.7: Pozicije centrov po y osi.Slika 4.8: Pozicije centrov po z osi.

4.4 Primerjava metode z nekaterimi drugimi samorazvijajočimi metodami

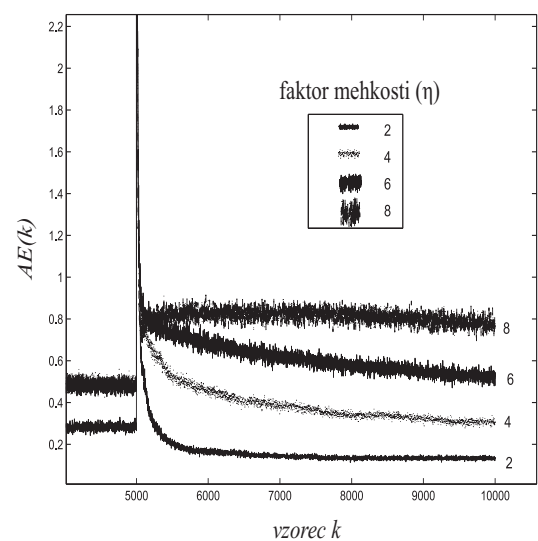
Razvita metoda je bila primerjana z nekaterimi ostalimi objavljenimi metodami. Tipičen testni problem, na katerem so metode učenja primerjane, je



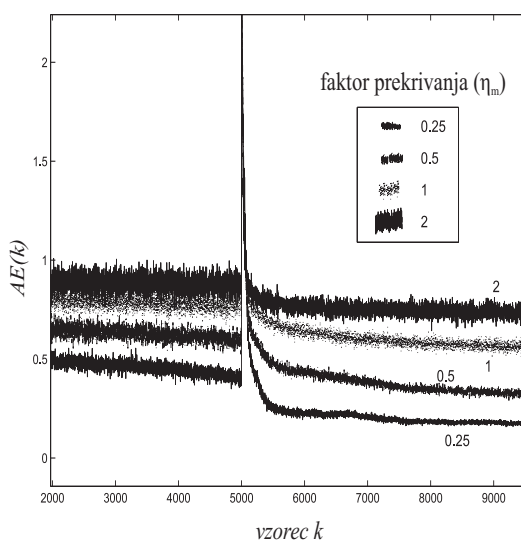
Slika 4.9: Vpliv faktorja pozabljanja na variacije v oceni parametrov.



Slika 4.10: Vpliv faktorja pozabljanja na hitrost prilagajanja modela.



Slika 4.11: Vpliv faktorja mehkosti.



Slika 4.12: Vpliv faktorja prekrivanja.

Mackey–Glass časovna vrsta [127]. Kaotična časovna vrsta je generirana z enačbo 4.54:

$$x(t) = \frac{0,2x(t - \tau)}{1 + x^{10}(t - \tau)} - 0,1x(t). \quad (4.54)$$

Cilj eksperimenta je, da iz preteklih vzorcev x ocenimo prihodnjo vrednost vrste. Začetno stanje vrste definiramo kot $x(0) = 1,2$, časovna zakasnitev pa je izbrana

kot $\tau = 17$. Predikcijo delamo na podlagi trenutne vrednosti x ter zakasnenih vrednosti x za 6, 12 in 18 vzorcev. Ocenjujemo x za 85-ti korak vnaprej.

$$\begin{aligned} Izhod : & [x(t + 85)], \\ Vhod : & [x(t - 18)x(t - 12)x(t - 6)x(t)]. \end{aligned} \quad (4.55)$$

Točnost modela opredelimo z nedimenzionalnim indeksom napake (en. 4.56) (*non-dimensional error index* (NDEI) [82]):

$$\begin{aligned} NDIE &= \frac{1}{\sigma^2} \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \\ \sigma^2 &= \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2}, \\ \bar{y} &= \frac{1}{n} \sum_{i=1}^n y_i. \end{aligned} \quad (4.56)$$

Za učenje modela vzamemo 3000 vzorcev iz intervala $t \in [201, 3200]$ in 500 vzorcev iz intervala $t \in [5001, 5500]$ za evalvacijo naučenega modela [92]. V nekateri člankih vzorce pred učenjem normirajo [92], v našem primeru tega nismo storili. Rezultati so podani v tabeli 4.3. Cenilke drugih metod so povzete po člankih v katerih so objavljene. Za računanje pripadnosti smo uporabili radialne pripadnostne funkcije.

Predikcija časovne vrste z modelom s 100 roji je prikazana na sliki 4.13, skupaj z napako predikcije.

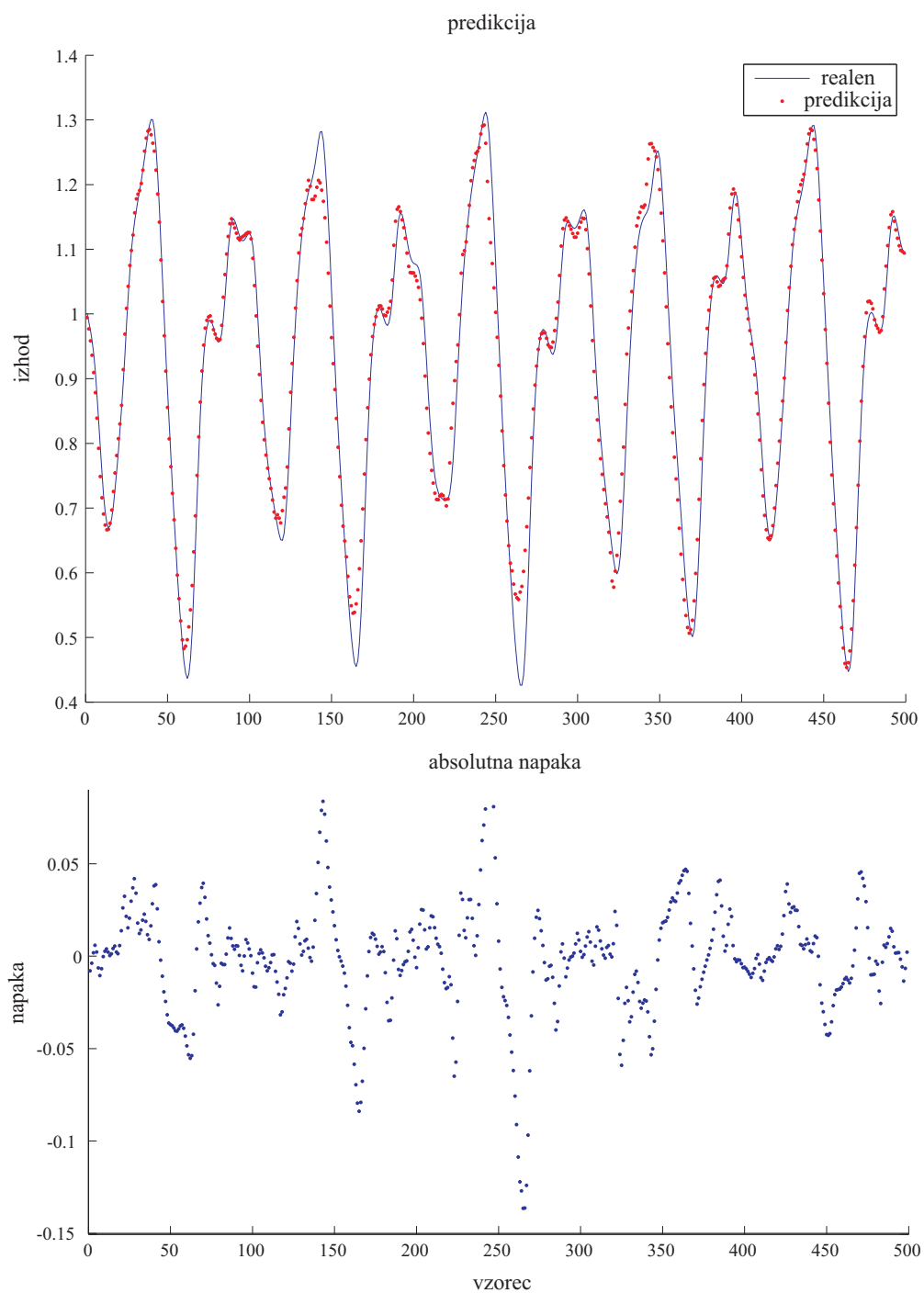
Za testiranje robustnosti smo vrsti dodali Gaussov šum z varianco 0,01. V tem primeru je bila NDEI cenilka za 100 rojev 0,4809. Izhod in napaka modela sta prikazana na sliki 4.14.

V literaturi najdemo tudi primerjavo metod na šest koračni predikciji [137]. Za šest-koračno predikcijo je rGK metoda imela NDIE indeks 0,0859 z 10-imi roji. Več o primerjavi predlagane metode z drugimi najdemo v [50] in [49].

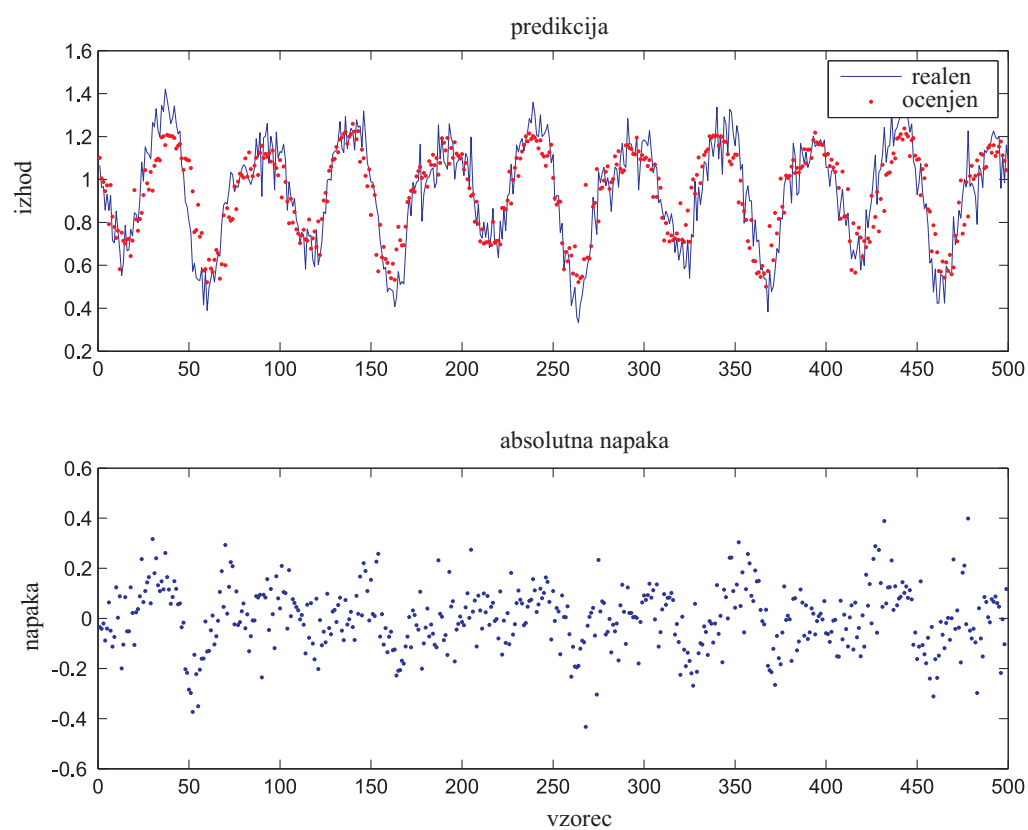
Predlagana metoda daje primerljive rezultate z že obstoječimi uveljavljenimi metodami. Dobro deluje z majhnim številom rojev in pri prisotnosti šuma. Metoda je preprosta in lahko razumljiva. Tudi nastavljanje parametrov je lažje (vsaj po našem mnenju) kot pri nekaterih ostalih metodah.

Metoda	Pravila/roji	NDEI
DENFIS	58	0,276
DENFIS	27	0,404
exTs	10	0,331
exTs	9	0,361
eTs	9	0,372
eTS	113	0,0954
eTs+	8	0,438
eTs+	10	0,392
RAN	113	0,375
ESOM	114	0,32
EFuNN	193	0,401
DENFIS	883	0,033
ESOM	1000	0,044
Neural gas	1000	0,062
EFuNN	1125	0,094
rFCM	3	0,4849
rFCM	10	0,4562
rFCM	58	0,3085
rFCM	100	0,1250
rGK	3	0,4667
rGK	10	0,3787
rGK	58	0,2113
rGK	100	0,1166

Tabela 4.3: Primerjava predlagane rekurzivne FCM in GK metode z ostalimi.



Slika 4.13: Predikcija MG časovne vrste



Slika 4.14: Predikcija MG časovne vrste pri pošumljenem signalu

5. Vodenje z adaptivnim mehkim modelom

V zadnjem desetletju se je pojavilo veliko idej in algoritmov na področju adaptivnih mehkih regulatorjev. Na splošno adaptivne mehke regulatorje lahko delimo na direktne in indirektne. Direktni adaptivni mehki regulatorji skušajo z mehko logiko aproksimirati idealni regulacijski zakon za določeno območje ([31, 104, 24, 151, 102, 162, 161, 97, 177, 154, 7]). Indirektni pristopi skušajo z mehko logiko aproksimirati obnašanje procesa. Na podlagi dobljenega modela spreminjajo parametre mehkega regulatorja ([69, 175, 140, 176, 28, 58]). Pristopi večinoma prilagajajo le parametre linearnih podmodelov, ne pa tudi pripadnostne funkcije oziroma mehkih področij. Ta so vnaprej določena in fiksna. Malo adaptivnih mehkih regulatorjev prilagaja tako strukturo (pripadnostne funkcije) kot tudi parametre linearnih podmodelov. Direktn pristop, kjer je uporabljen mehki regulator in se adaptirajo tako pripadnostne funkcije kot tudi parametri regulacijskega zakona, je predstavljen v [7]. Za adaptacijo je uporabljena eTS metoda. V [139] je predlagan direktn algoritem z mehkim regulacijskim zakonom. V pristopu so uporabljene trikotne pripadnostne funkcije in mehki TS model ničtega reda. Algoritem dodaja in prilagaja pripadnostne funkcije glede na sistemsko napako in definiran prag le-te ter na podlagi ϵ_0 -kompletnosti posameznega pravila. Direktn pristop, ki prav tako uporablja trikotne pripadnostne funkcije, je predstavljen v [149]. Pristop prilagaja tako parametre linearnih modelov (*consequents*) kot tudi pripadnostne funkcije. Pripadnostne funkcije skuša pozicionirati tako, da je izpolnjen določen kriterij kvalitete regulacije skozi celotno območje. Za mero kvalitete regulacije uporabljajo kvadratno napako. Indirektni pristop je predstavljen v [140]. Pri tem pristopu uporabljajo Gausove pripadnostne funkcije, adaptacija centrov in širine funkcij pa je izvedena s pristopom padajočega gradienta (*gradient decent*). Parametri podmodelov so adaptirani z metodo najmanjših kvadratov. Na podlagi TS mehkega modela nato načrtujejo regulator s povratno linearizacijo (*feedback linearization control*).

5.1 Mehki prediktivno funkcijski regulator z adaptivnim mehkim modelom

Mehki prediktivno funkcijski regulator z adaptivnim mehkim modelom (AFPFC) lahko uvrstimo v skupino pristopov vodenja, ki jo imenujemo prediktivno vodenje na podlagi modela (*MPC-model based control*). Strategija MPC temelji na uporabi dinamičnega modela procesa za predikcijo izhoda procesa na določenem horizontu [26]. Cilj pristopa je določiti tak izhod regulatorja, da minimizira kriterijsko funkcijo 5.1:

$$J(u, k) = \sum_{j=N_1}^{N_2} (y_m(k+j) - y_r(k+j))^2 + \lambda \sum_{j=1}^{N_u} u^2(k+j), \quad (5.1)$$

kjer je $y_m(k+j)$ j -koračna predikcija izhoda procesa, $y_r(k+j)$ referenčni izhod procesa in $u(k+j)$ regulirni signal. N_1 in N_2 označujeta minimalni in maksimalni horizon za predikcijo izhoda procesa, N_u pa horizon za regulacijski signal. λ je utež s katero določimo relativno pomembnost izhoda in regulirnega signala.

MPC je ime za skupek regulacijskih algoritmov, ki se v splošnem razlikujejo po uporabljenem modelu, kriterijski funkciji in šumu, ki ga želijo minimizirati. Nekateri najbolj znani so MAC (Model Algorithmic control), EPSAC (Extended Prediction Self Adaptive Control), EHAC (Extended Horizon Adaptive Control), GMV (Generalized Predictive Control) in PFC (predictive functional control). Algoritmi so bolj podrobno predstavljeni v [26]. Chamacho in Bordons kot prednosti MPC pristopov navajata privlačnost za operaterje zaradi intuitivnih konceptov, enostavno uglaševanje parametrov, možnost vodenja procesov z enostavno in kompleksno dinamiko, možnost regulacije multivariabilnih sistemov, dobro vodenje sistemov z mrtvim časom, enostavni regulacijski zakoni za implementacijo, možnost vključevanja raznih omejitev v postopek načrtovanja regulatorjev.

Prediktivno funkcijski regulator (PFC) je vpeljal Richalet [147]. Motiviran s potrebami petrokemije in podobne industrije je Jacques Richelet med poznimi 60. in zgodnjimi 70. leti 20. stoletja začel z razvojem in promocijo nove prediktivne ideje vodenja. Njegov algoritem je doživel veliko uspešnih aplikacij v industriji [145, 147, 142] in raznih akademskih študijah [48, 51, 100, 171, 107, 23, 191]. Večina ga smatra za očeta modelno prediktivnega vodenja [146]. Uspeh prediktivno funkcijskega regulatorja lahko pripišemo njegovi enostavnosti, razumljivosti

[145, 147] in dobremu delovanju. Algoritem je bil sprva razvit za linearne sisteme z enim vhodom in enim izhodom. Izkaže se, da je za preprost PFC algoritem [51],[48] možno dokazati, da je to dejansko PI regulator oziroma PI regulator s Smithovim prediktorjem v primeru, ko upoštevamo mrtvi čas procesa. Kasneje je bil PFC algoritem razširjen za vodenje nelinearnih sistemov [34, 53] in multi-variabilnih sistemov [170, 134].

Za regulacijo nelinearnih sistemov je zelo priročen prediktivno funkcijski regulator z mehkim modelom (FPFC-*fuzzy predictive functional control*) [172, 107]. Pristop temelji na ideji, da lahko nelinearen proces (z gladko nelinearnostjo) aproksimiramo s setom linearnih modelov v obliki Takagi-Sugeno mehkega modela [159, 178].

PFC algoritem je izpeljan na predpostavki, da lahko dinamiko procesa aproksimiramo z linearnim modelom prvega reda. V podrobno izpeljavo se na tem mestu ne bomo spuščali, lahko si jo ogledamo v [51]. V trenutni delovni točki lahko tudi nelinearni proces aproksimiramo z linearnim modelom prvega reda. Slednjega pri FPFC dobimo iz mehkega modela. Zapišemo lahko diferenčno enačbo globalnega linearnega modela:

$$y_m(k+1) = \tilde{a}_m y_m(k) + \tilde{b}_m u(k) + \tilde{r}_m, \quad (5.2)$$

kjer je $y_m(k+1)$ izhod mehkega modela, ki ga dobimo z uteženo vsoto izhodov linearnih podmodelov mehkega modela 5.3:

$$y_m(k+1) = \sum_{i=1}^c \beta_i y_{m,i}. \quad (5.3)$$

Pri uporabi mehkega Takagi-Sugeno modela prvega reda lahko izhod linearnega podmodela zapišemo kot:

$$y_{m,i}(k+1) = a_{m,i} y_{m,i}(k) + b_{m,i} u(k) + r_{m,i} \quad i = 1, 2, \dots, c. \quad (5.4)$$

Z upoštevanjem enačbe 5.3 in 5.4 lahko zapišemo enačbe za $\tilde{a}_m$, $\tilde{b}_m$, in $\tilde{r}_m$:

$$\tilde{a}_m = \beta_1 a_{m,1} + \dots + \beta_c a_{m,c}, \quad (5.5)$$

$$\tilde{b}_m = \beta_1 b_{m,1} + \dots + \beta_c b_{m,c}, \quad (5.6)$$

$$\tilde{r}_m = \beta_1 r_{m,1} + \dots + \beta_c r_{m,c}. \quad (5.7)$$

S tem smo dobili globalni linearni model. Za izpeljavo regulacijskega zakona je potrebno podati še referenčno trajektorijo odziva procesa. Ta je podana v obliki referenčnega modela (en. 5.8), ki zagotavlja sledenje referenci w :

$$y_r(k+1) = a_r y_r(k) + (1 - a_r) w(k). \quad (5.8)$$

Regulacijski zakon je dobljen z minimizacijo kriterijske funkcije v enačbi 5.1. PFC predvideva horizon ujemanja H ($N_1 = N_2 = H$) in utežni faktor $\lambda = 0$. V horizonu ujenanja naj predikcija izhoda procesa sovpadala z referenčnim odzivom. S predpostavko konstantnega regulirnega signala (*constant future control*) in padajočega horizona (*receding horizon*) dobimo regulacijski zakon v obliki:

$$u(k) = \frac{(1 - a_r^H)(w(k) - y_p(k))}{\frac{\tilde{b}_m}{1 - \tilde{a}_m}(1 - \tilde{a}_m^H)} + \frac{y_m(k)}{\frac{\tilde{b}_m}{1 - \tilde{a}_m}} - \frac{\tilde{r}_m}{\tilde{b}_m}. \quad (5.9)$$

Podrobna izpeljava algoritma je izpeljana v [171]. O stabilnosti in integralni naravi FPFC algoritma si lahko preberemo v [169].

Za vodenje časovno spremenljivih nelinearnih procesov lahko uporabimo adaptivni FPFC. Adaptivnost lahko dosežemo z uporabo predlagane mehke rekurzivne identifikacije. S tem zagotovimo prilagajanje mehkega modela novim razmeram v procesu. Primer delovanja AFPFC si bomo ogledali na primeru regulacije semi-saržnega reaktorja.

5.2 Vodenje semi-saržnega reaktorja z AFPFC

Semi-saržni reaktor je pomemben del procesov v kemijski in farmacevtski industriji. Problem vodenja semi-saržnega reaktorja v večini primerov pomeni problem vodenja temperature zmesi v njem. Pri tem probleme vnaša hibridna narava procesa, saj imamo tako zvezne kot diskretne vhode, različne mešanice v reaktorju, različni načini delovanja reaktorja, dodajanje novih snovi v reaktor med samim delovanjem ... Običajno naletimo na semi-saržni reaktor, kjer je najprej dodana začetna mešanica snovi, ko pa proces segrevanja pride v določeno stanje, v reaktor dodamo dodatne primesi. S tem rasteta volumen in masa zmesi v reaktorju, spremeni se površina in s tem prehajanje toplote ter sama toplotna kapacitivnost zmesi. Semi-saržni reaktor predstavlja časovno spremenljiv proces

z neznanimi parametri. Cilj regulacije je zagotoviti čimbolj natančno sledenje vnaprej podani referenčni temperaturi brez večjih prevzponov ([168]).

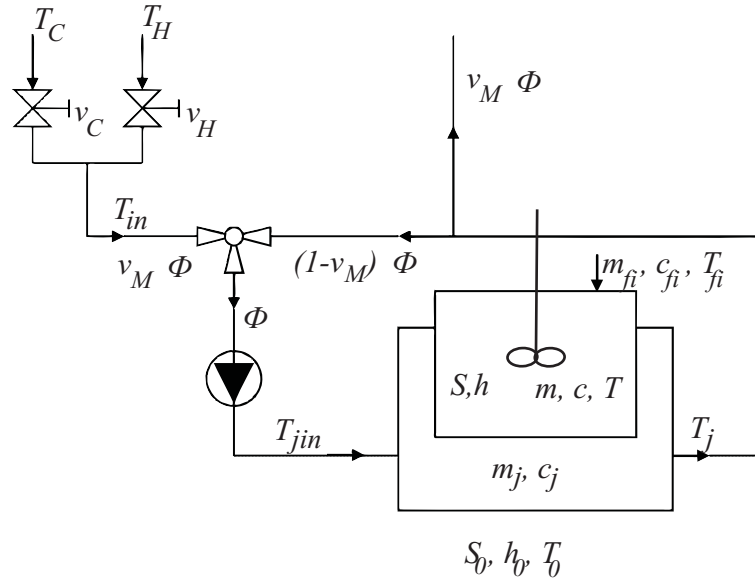
Regulacija semi-saržnega reaktorja je ena izmed bolj obravnavanih tem v procesni avtomatiki. V [86] so obravnavani obetavni koncepti vodenja, predlagani do leta 1986. Od takrat do danes je bilo objavljenih veliko strategij vodenja semi-saržnega reaktorja. Obetavnejši pristopi so koncepti adaptivnega vodenja ([30, 117]) in pristopi optimalnega vodenja ([29, 40, 124]). Najobetavnejši, tako z aplikativnega vidika kot tudi z vidika kvalitete vodenja pa so po našem mnenju pristopi prediktivnega vodenja ([36, 34, 46, 61, 75, 103, 113]). Koncepti MPC strategije vodenja so lahko razumljivi in preprosti za implementacijo. Uspešno so bili uporabljeni za reševanje realnih problemov vodenja v industriji ([19, 41, 35, 142, 66]). Največkrat uporabljena pristopa sta splošni prediktivni regulator GPC (*generalized predictive control*) [37] in prediktivno funkcijski regulator ([145]). Študija adaptivnega prediktivnega regulatorja na primeru semi-saržnega reaktorja je predstavljena v [88] in [167]. Za časovno spremenljive nelinearne procese, kot sta semi-saržni reaktor in pH proces [49], je smiselno uporabiti regulator, ki je sposoben upoštevati te spremembe.

5.2.1 Opis procesa in regulacijskega problema

Mehki prediktivno funkcijski regulator z adaptivnim mehkim modelom smo preizkusili na že omenjenem semi-saržnem procesu, katerega shema je predstavljena na sliki 5.1. Dinamično gledano je proces hibridne narave, kjer imamo poleg zveznih vhodov in izhodov tudi diskretne vhode. Cilj regulacije je zagotoviti čim bolj natančno sledenje notranje temperature reaktorja (temperature zmesi) referenčni temperaturi, ki je vnaprej podana z receptom.

Notranjost reaktorja (temperatura T), kjer se nahaja zmes, je ogrevana oziroma hlajena preko medija, ki teče skozi plašč reaktorja (temperatura T_j). Temperatura medija in zmesi reaktorja sta tudi merjeni veličini in predstavljata izhod procesa.

Medij v plašču je mešanica med svežim glikolom, ki priteče v sistem skozi ON/OFF ventile in povratnim glikolom, ki je že v sistemu. Temperatura svežega glikola je odvisna od tega, kateri ON/OFF ventil je odprt. Če odpremo ventil



Slika 5.1: Sehma semi-saržnega reaktorja.

za vroč glikol ($v_H = 1$), bo ventil za mrzel glikol zaprt ($v_C = 0$). Temperatura glikola bo v tem primeru $T_{in} = T_H = 130\text{ }^{\circ}\text{C}$. Obratno velja za ventil, ki prepušča mrzel glikol ($v_C = 1$ in $v_H = 0$). Temperatura svežega glikola je v tem primeru $T_{in} = T_C = -25\text{ }^{\circ}\text{C}$. Ventila v_C in v_H predstavljata diskretna vhoda procesa. Razmerje med svežim in povratnim glikolom določamo z odprtostjo mešalnega ventila v_m , ki predstavlja zvezni vhod v naš proces. Z njim določamo temperaturo medija na vhodu v plašč (T_{jin}), ki pa je ne merimo. Ocenjujemo jo iz temperature svežega in povratnega glikola ter odprtosti mešalnega ventila (en. 5.16). Odprtost mešalnega ventila je v mejah $[0\text{ } 1]$. Časovne konstante ventilov smo zanemarili, saj so precej manjše od ostalih časovnih konstant.

Obnašanje reaktorja lahko opišemo z diferencialnimi enačbami 5.10, 5.11, 5.13, 5.14, 5.15 in algebrajičnima enačbama 5.12, 5.16.

$$m_j c_j \frac{dT_j}{dt} = v_m \Phi c_j T_{in} + (1 - v_m) \Phi c_j T_j - \Phi c_j T_j - hS(T_j - T) - h_0 S_0(T_j - T_0) \quad (5.10)$$

$$mc \frac{dT}{dt} = hS(T_j - (T + \Delta T \delta(t - T_i))) \quad (5.11)$$

$$\Delta T = \frac{m_{fi} c_{fi} (T_{fi} - T)}{mc + m_{fi} c_{fi}} \quad (5.12)$$

V zgornjih enačbah m_j označuje maso glikola v plašču reaktorja, c_j specifično toplotno kapacitivnost glikola v plašču, Φ masni pretok skozi plašč, h koeficient prevajanja toplote med plaščem in jedrom reaktorja in S površino prevajanja med njima. Označbi h_0 in S_0 označujeta faktor in površino prevajanja toplote med plaščem in okolico. Temperaturo okolice smo v našem primeru smatrali kot konstanto $T_0 = 17^\circ\text{C}$. To je v večini primerov ustrezna predpostavka, saj so običajno prostori, kjer se reaktorji nahajajo, klimatizirani na določeno temperaturo.

Med izvajanjem recepta se v nekem trenutku dodajo nove sestavine v reaktor. To povzroči spremembo v masi (m) zmesi v reaktorju, spremembo specifične toplotne kapacitivnosti (c), temperature v jedru reaktorja (T) in površine prevajanja (S). Te spremembe so precej hitrejše od dinamike procesa, zato jih lahko modeliramo z nezveznimi skoki (Dirakovimi impulzi $\delta(t - t_i)$), ki se pojavijo v času t_i . Sprememba temperature T je modelirana z enačbo 5.11. Skok ΔT je podan z enačbo 5.12, kjer je m_{fi} masa dodatka, c_{fi} specifična toplotna kapacitivnost dodatka in T_{fi} temperatura dodatka v času dodajanja. Sprememba mase v reaktorju je podana z enačbo 5.13. Enačba 5.14 opisuje spremembo specifične toplotne kapacitivnosti zmesi, kjer je $\Delta c_i = (c_{fi} - c) \frac{m_{fi}}{m + m_{fi}}$.

$$\frac{dm}{dt} = m_{fi} \delta(t - t_i) \quad (5.13)$$

$$\frac{dc}{dt} = \Delta c_i \delta(t - t_i) \quad (5.14)$$

Sprememba površine prevajanja S je podana z enačbo 5.15:

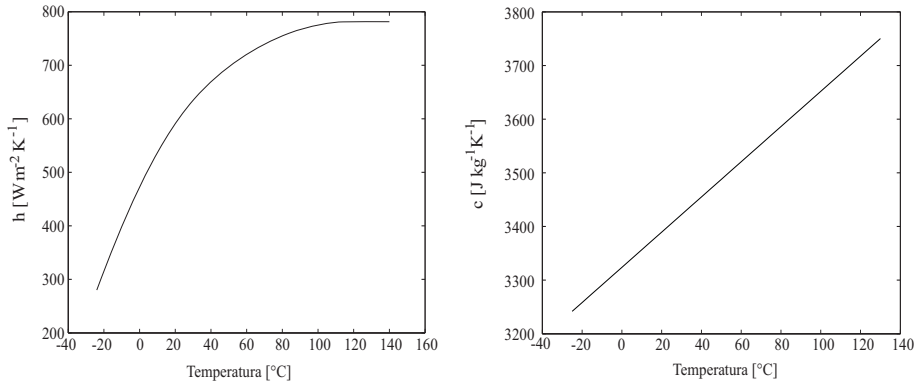
$$\frac{dS}{dt} = S_{fi} \delta(t - t_i), \quad (5.15)$$

kjer je S_{fi} sprememba površine v času t_i zaradi nove dodane snovi, ki je približno ocenjena kot $S_{fi} = S \frac{m_{fi}}{m}$.

Temperatura jedra reaktorja je regulirana posredno s temperaturo plašča (T_{jin}). Ta je odvisna od pozicije mešalnega ventila in temperature svežega glikola (en. 5.16):

$$T_{jin} = v_m T_{in} + (1 - v_m) T_j. \quad (5.16)$$

Z uporabo glikola za ogrevanje in hlajenje reaktorja dobimo nelinearno karakteristiko procesa. Koeficient prevajanja toplote je nelinearna funkcija, odvisna



Slika 5.2: Koeficient prevajanja toplote glikola (levi graf) in specifična toplota glikola (desni graf).

od temperature glikola v plašču. Podobno je s specifično toplotno kapaciteto. Karakteristika obeh je podana na sliki 5.2. Karakteristika procesa je podana na sliki 5.3 (ojačanje) in sliki 5.4 (časovna konstanta).

5.2.2 Vodenje procesa

Vodenje semi-saržnega reaktorja temelji na indirektni regulirni veličini T_{jin} . Iz nje izračunamo pozicije regulacijskih elementov: pozicija mešalnega ventila (v_m) in odprtost ali zaprtost diskretnih ON/OFF ventilov (v_C in v_H). Odprtost mešalnega ventila izračunamo iz preurejene enačbe 5.16:

$$v_m = \frac{T_{jin} - T_j}{T_{in} - T_j}. \quad (5.17)$$

Za določitev odprtosti diskretnih ventilov je potrebno uvesti logiko, ki izbira med mrzlim in vročim svežim glikolom. Če je referenčna temperatura višja kot temperatura okolice T_0 , se logika preklapljanja glasi:

$$\begin{aligned} \text{če je } e(k) < -\delta_d, \text{ potem je } T_{in}(k) &= T_C \quad (v_C = 1, v_H = 0), \\ \text{če je } e(k) > -\delta_u, \text{ potem je } T_{in}(k) &= T_H \quad (v_C = 0, v_H = 1), \\ \text{drugače } T_{in}(k) &= T_{in}(k-1), \end{aligned} \quad (5.18)$$

kjer sta δ_d in δ_u pragova preklapljanja in $e(k)$ napaka med referenčno in dejansko temperaturo. Ko je referenčna temperatura nad temperaturo okolice, se logika

preklapljanja glasi:

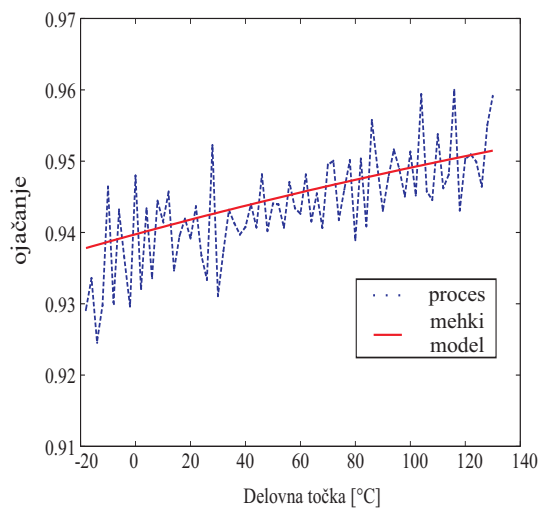
$$\begin{aligned} \text{če je } e(k) > \delta_d, \text{ potem je } T_{in}(k) &= T_H \quad (v_C = 0, v_H = 1), \\ \text{če je } e(k) < \delta_u, \text{ potem je } T_{in}(k) &= T_C \quad (v_C = 1, v_H = 0), \\ \text{drugače } T_{in}(k) &= T_{in}(k-1). \end{aligned} \quad (5.19)$$

Logiko preklapljanja smo ločili na dva scenarija glede na vrednost referenčne temperature v primerjavi s temperaturo okolice. Ideja za to je naslednja. Ko je referenčna temperatura nad temperaturo okolice, poteka hlajenje v okolici referenčne temperature preko prehajanja toplote iz reaktorja v okolico. Ker je vroč glikol potreben za ogrevanje, pustimo odprt ventil za vroč glikol. Obratno razmišljanje velja za primer, ko je referenčna temperatura pod temperaturo okolice. Takrat za gretje okoli reference uporabimo prehajanje toplote iz okolice na reaktor. Ker je za hlajenje potreben mrzel glikol, pustimo odprt ventil za mrzel glikol. S tem lahko zmanjšamo preklope med ventiloma v_C in v_H in povečamo njuno življensko dobo.

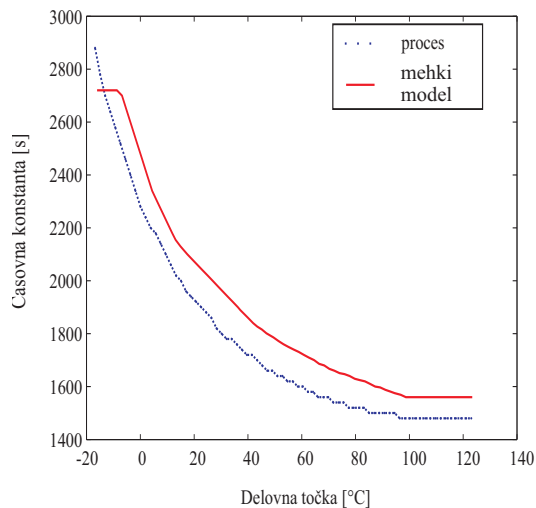
Ena izmed zahtev, ki jo je potrebno v praksi upoštevati, je omejitev regulacijskega signala. Razlika med temperaturo plašča in temperaturo jedra reaktorja ne sme preseči $60 \text{ } ^\circ\text{C}$.

Začetni Takagi-Sugeno mehki model smo zgradili z uporabo rekurzivne *c*-means metode z radialnimi pripadnostnimi funkcijami. Metoda je bila uporabljena tudi za kasnejšo adaptacijo modela. Uporaba GK algoritma je predstavljena v [49] na primeru vodenja pH procesa. Nastavitve metode za začetno identifikacijo so podane v tabeli 5.1. Model je bil identificiran v različnih delovnih točkah s psevdo naključnim binarnim signalom. Karakteristika zgrajenega mehkega modela je zelo podobna karakteristiki procesa. Na sliki 5.3 je prikazana primerjava med ojačanjem procesa in modela, na sliki 5.4 pa primerjava med časovnimi konstantami v različnih delovnih točkah. Časovne konstante in ojačanje v delovnih točkah smo določili iz odziva na stopnico. Glede na to, da je bil našemu procesu dodan tudi šum, je tudi potek ojačanja na sliki 5.3 precej pošumljen. Na temperaturo plašča (T_j) je bil dodan Gaussov šum z varianco 0,03, na temperaturo jedra reaktorja (T) je bil dodan Gaussov šum z varianco 0,08. To je povzročilo nihanje izhodne temperature reaktorja T v mejah $\pm 0,4 \text{ } ^\circ\text{C}$ okoli nepošumljene vrednosti.

Parameter	oznaka	vrednost
LSE kovariančna matrika	P_i	$100I_{3 \times 3}$
Mehka kovariančna matrika	F_i	$1,5I_{3 \times 3}$
Faktor pozabljanja	λ_r	1
Faktor pozabljanja	γ_c	1
Faktor pozabljanja	γ_v	1
Faktor mehkosti	η	2
Faktor prekrivanja	η_m	0,3
Vsota pripadnosti	s_i	1
Parametri podmodelov	θ_i	0
Število rojev	c	16

Tabela 5.1: Začetne vrednosti mehke c -means metode.

Slika 5.3: Ojačanje procesa.



Slika 5.4: Časovna konstanta procesa.

veličina	oznaka	vrednost
masa v jedru reaktorja	m	600 kg
površina prevajanja toplote	S	2 m ²
specifična toplotna kapacitivnost zmesi	c	4200 Jkg ⁻¹ K ⁻¹
koeficijent prevajanja toplote	h	Sl. 5.2 desno
masa glikola v plašču	m_j	200 kg
specifična toplotna kapacitivnost glikola	c_j	Sl. 5.2 levo
površina prevajanja toplote	S_0	4 m ²
koeficijent prevajanja toplote	h_0	84 Wm ⁻² K ⁻¹
temperatura v plašču	$T_j(0)$	17 °C
temperatura v jedru	$T(0)$	17 °C
pozicija ventila v_H	v_H	1 - odprt
pozicija ventila v_C	v_C	0- zaprt
masni pretok	Φ_m	1,6 kgs ⁻¹
temperatura vročega glikola	T_H	130 °C
temperatura mrzlega glikola	T_C	-25 °C

Tabela 5.2: Začetne vrednosti in parametri procesa semi-saržnega reaktorja.

Začetne vrednosti opisanega modela reaktorja so podane v tabeli 5.2. Čas vzorčenja je bil $T_s = 20$ s. Med delovanjem reaktorja je bila vanj dodana zmes s temperaturo $T_{fi} = 17$ °C, maso $m_{fi} = 1000$ kg in specifično toplotno kapaciteto $c_{fi} = 4000$ Jkg⁻¹K⁻¹. Rojenje je bilo izvedeno na vhodno-izhodnem prostoru s podatkovnim vektorjem $\mathbf{x}(k) = [T(k-1) \ T_{in}(k-1) \ T(k)]^T$. Regresijski vektor za najmanjše kvadrate je bil v obliki $\psi_i(k+1) = \beta_i(k)[1 \ T(k-1) \ T_{in}(k-1)]^T, i = 1, \dots, c$ z izhodom i -tega lokalnega modela $y_i = \beta_i(k)T(k)$. Identifikacija parametrov lokalnih modelov je bila izvedena z enačbo 4.38. Za preprečitev eksplodiranja kovariančne matrike najmanjših kvadratov v primeru, ko je vzbujanje slabo, je bila uvedena tudi mrtva cona v vrednosti 0,01 [16].

Narava procesa in motnje nam omogočata, da lahko detektiramo trenutek dodajanja novih snovi v reaktor (motnjo). Le-ta je detektiran, ko se ob konstantni referenčni vrednosti temperature izhod procesa T občutno spremeni. Takrat se ponovno nastavijo kovariančne matrike $\mathbf{P}_i$ na njihovo začetno vrednost. Vsote pripadnosti s_i in mehke kovariančne matrike $\mathbf{F}_i$ se ponastavijo s spremembo faktorja pozabljanja za en vzorec na vrednost $\gamma_v = \gamma_v = 0,0001$ [16]. S tem zagotovimo hitrejšo prilagoditev mehkega modela na spremembo procesa.

Parametri AFPFC regulatorja se navezujejo na parametre mehkega modela. Podani so v tabeli 5.3, kjer je $\tilde{a}_m$ parameter modela (en. 5.2). Nastavitve za horizon in referenčno hitrost odziva v tabeli 5.3, gledano z vidika časovne konstante

Parameter	oznaka	vrednost
Referenčna hitrost odziva	a_r	$\tilde{a}_m^5$
Horizon	H	$\lceil -0,5(\log(a_r))^{-1} \rceil$
Prag za preklap	δ_d	$1 \text{ } ^\circ C$
Prag za preklap	δ_u	$0,5 \text{ } ^\circ C$

Tabela 5.3: Parametri AFPFC in preklapne logike.

procesa pomenijo, da je horizon enak:

$$H = \lceil \frac{T_r}{2T_s} \rceil \quad (5.20)$$

in zelena hitrost referenčnega odziva

$$T_r = \frac{1}{5}T_m, \quad (5.21)$$

kjer je T_m časovna konstanta modela. Hitrost referenčnega odziva v diskretnem prostoru pri podani zeleni časovni konstanti T_r dobimo z enačbo 5.22:

$$a_r = e^{-\frac{T_s}{T_r}}. \quad (5.22)$$

Z upoštevanjem enačbe 5.21 lahko zapišemo:

$$a_r = e^{-\frac{5T_s}{T_m}} = e^{-\frac{T_s}{T_m}^5} = a_m^5. \quad (5.23)$$

Podobno izpeljemo enačbo za horizon. Horizon je odvisen od razmerja zelene časovne konstante in časa vzorčenja. To razmerje je že vsebovano v enačbi 5.22. Z logaritmiranjem enačbe 5.22 pridemo do razmerja, ki pa je potrebno ustrezno matematično obdelati:

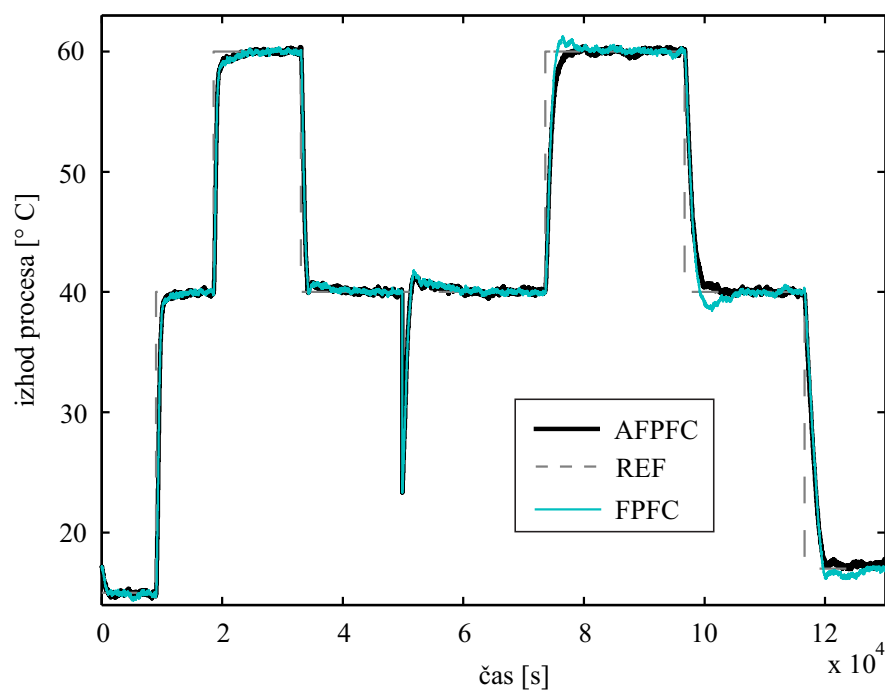
$$\log a_r = -\frac{T_s}{T_r} \quad (5.24)$$

Če zgornjo enačbo pomnožimo z minus dva, jo obrnemo in zaokrožimo, dobimo enačbo za horizon:

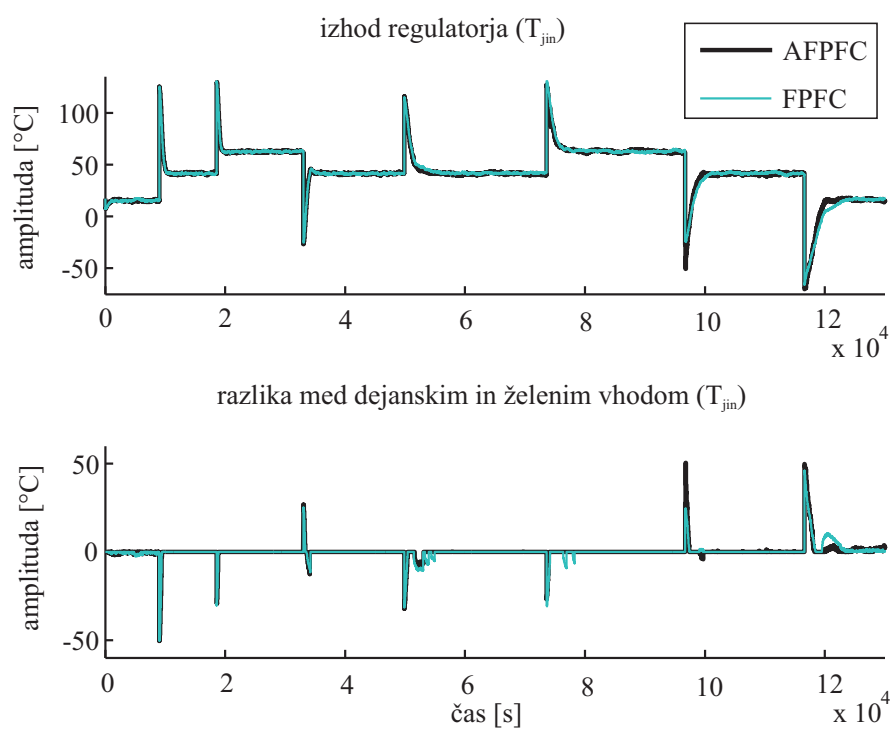
$$\lceil \frac{T_r}{2T_s} \rceil = \lceil (-2 \log a_r)^{-1} \rceil = H. \quad (5.25)$$

Več o nastavitvah prediktivno funkcijskega regulatorja najdemo v [171] in [51].

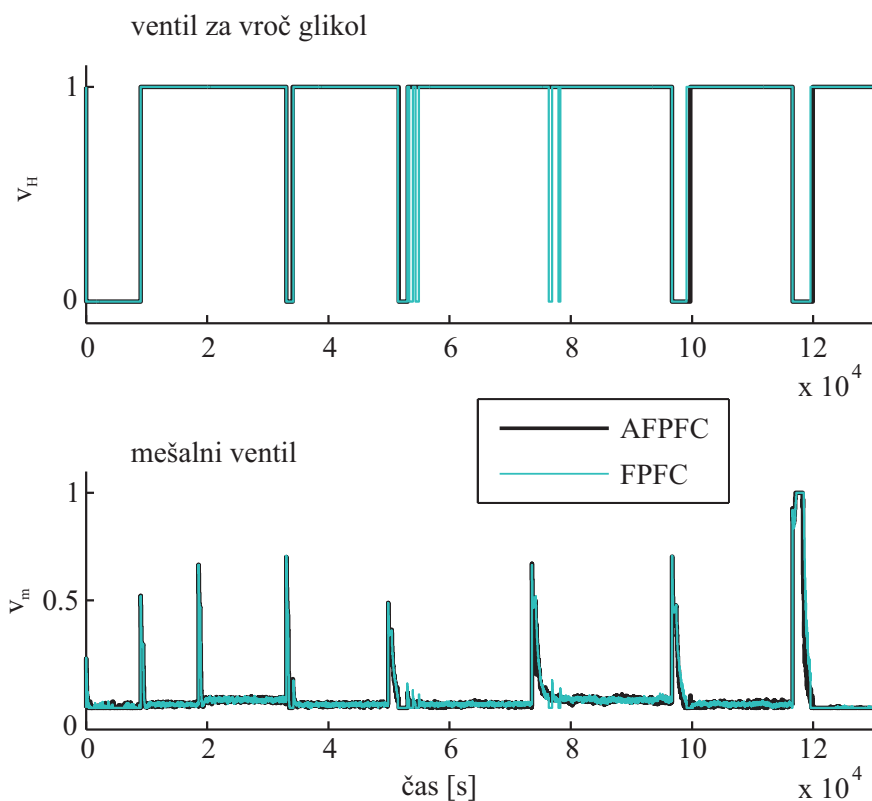
Odzive predlaganega AFPFC-ja smo primerjali z odzivi klasičnega FPFC-ja. Na sliki 5.5 so prikazani odzivi procesa skozi celoten eksperiment. Izhod obeh regulatorjev (zelena vrednost T_{jin}) je prikazan na zgornjem grafu slike 5.6. Na spodnjem grafu slike 5.6 je prikazana razlika med dejanskim vhomom v proces in izhodom regulatorja. Do razlik pride zaradi upoštevanja omejitve maksimalne razlike med temperaturo plašča in temperaturo jedra reaktorja. Pozicija



Slika 5.5: Odziv procesa pri vodenju z AFPFC in FPFC.



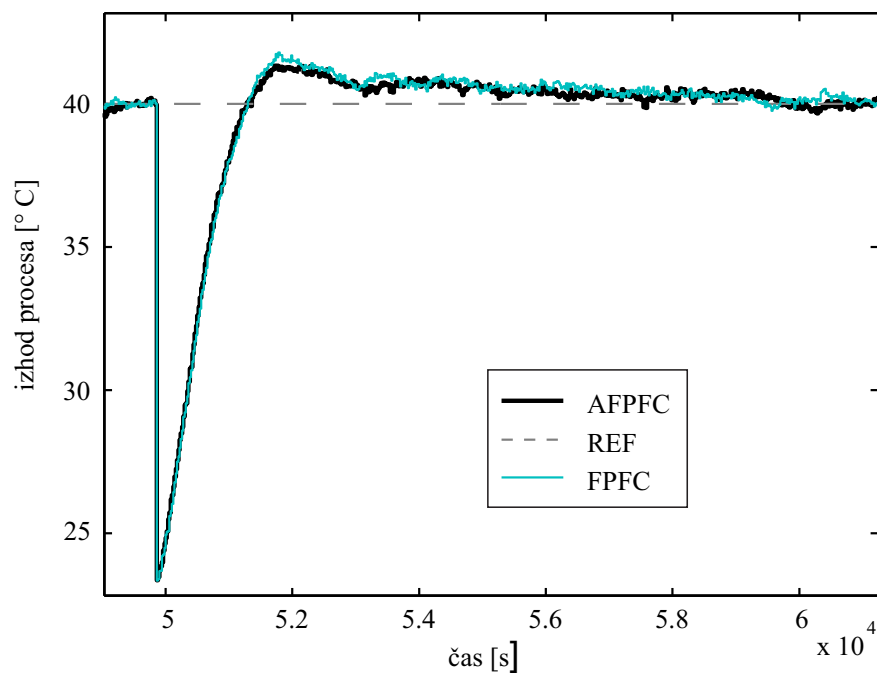
Slika 5.6: Izhod regulatorja (zgornji graf) in razlika med dejanskim vhodom v proces in izhodom regulatorja (spodnji graf).



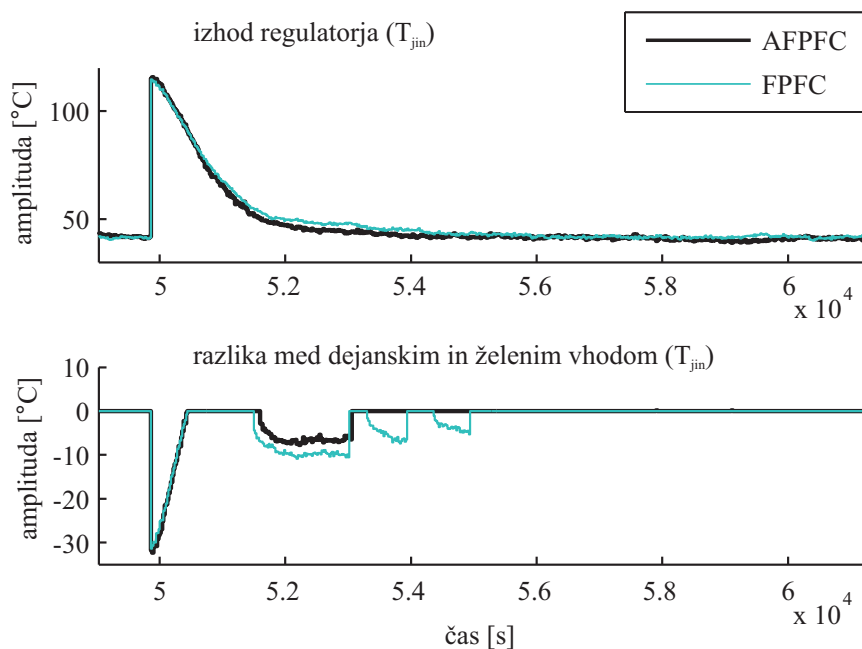
Slika 5.7: Pozicija diskretnega ventila za vroč glikol (zgornji graf) in pozicija mešalnega ventila (spodnji graf).

mešalnega ventila je prikazana na sliki 5.7 spodaj, prižiganje in ugašanje ventila za vroč glikol pa na zgornjem grafu. Zaradi precej dobrega začetnega mehkega modela ni večje razlike v kvaliteti regulacije s FPFC in AFPFC pred dodajanjem novega materiala. Z novim dodatkom se spremeni dinamika procesa. Opazimo lahko da ima proces, voden s FPFC rahel prevzpon pri odzivu na stopničasto spremembo reference. Pri procesu, vodenem z AFPFC, se mehki model adaptira na novo dinamiko in prevzpon ni opazen. Ker pri AFPFC prevzpona ni, je posledično tudi preklpov med ventilom za vroč in mrzel glikol manj.

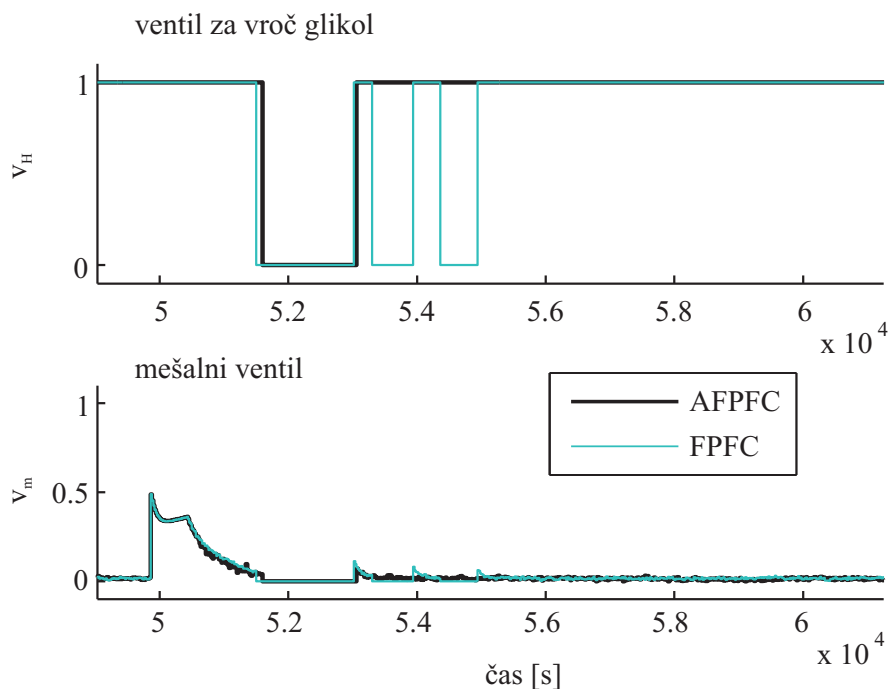
Podroben prikaz intervala dodajanja novega materiala v reaktor je prikazan na slikah 5.8, 5.9 in 5.10. Na sliki 5.8 je prikazana temperatura v jedru reaktorja. Slika 5.9 zgoraj prikazuje izhod regulatorja, spodaj pa razliko med dejanskim in želenim vhom v proces. Pozicija mešalnega ventila in ventila za vroč glikol je prikazana na sliki 5.10. Hitrost izreguliranja motnje je pri obeh regulatorjih podobna. Pri AFPFC je prevzpon malenkost manjši. Pri slednjem lahko opazimo



Slika 5.8: Odziv procesa na dodajanje novega materiala.



Slika 5.9: Izhod regulatorja (zgornji graf) in razlika med dejanskim in želenim vhodom v proces ob dodanem novem materialu (spodnji graf).



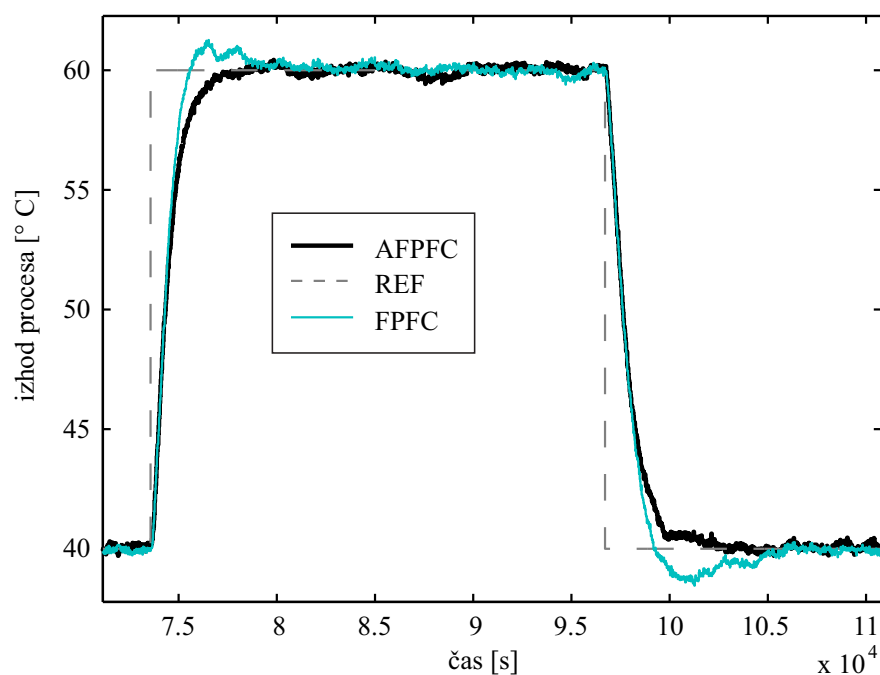
Slika 5.10: Pozicija ventila za vroč glikol (zgornji graf) in mešalnega ventila (spodnji graf) na intervalu dodajanja.

tudi manjše število preklopov med vročim in hladnim glikolom.

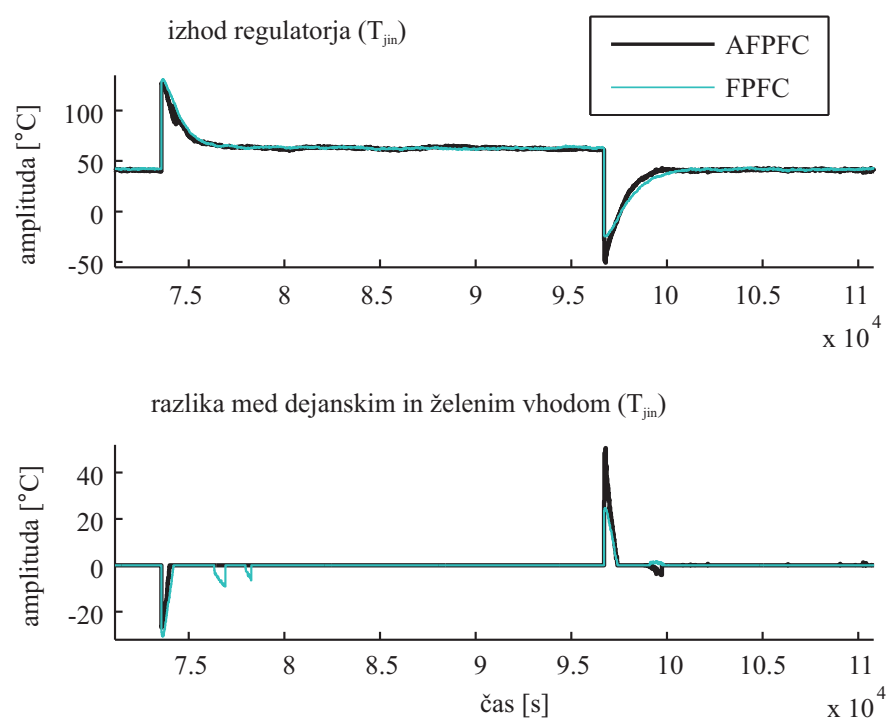
Na slikah 5.11, 5.12 in 5.13 je prikazan odziv na stopničasto spremembo reference po dodajanju. AFPFC adaptira mehki model na nove razmere v procesu. Vodenje procesa po dodajanju novih snovi se nadaljuje brez prevzopnov. To je velikokrat pomemben faktor pri vodenju, saj prevzponi v temperaturi vplivajo na samo kvaliteto izdelka. Z zmanjšanjem prevzponov zmanjšamo tudi število preklopov med mrzlim in vročim glikolom, kar podaljšuje življenjsko dobo ventilov.

Vrednosti SSE (*sum of squared error*) cenilke potrjujejo vizualna opažanja. Za izračun cenilke smo opravili deset eksperimentov. Pri vsakem smo izračunali vrednost SSE med referenco in izhodom procesa. Povprečna vrednost SSE za AFPFC regulator je bila $7,69e4$, kar je malenkost boljše od FPFC cenilke ($7,82e4$). Skoraj za polovico pa se zmanjša tudi število preklopov. Pri AFPFC je bilo opravljenih 10 preklopov med vročim in mrzlim glikolom pri FPFC pa 18.

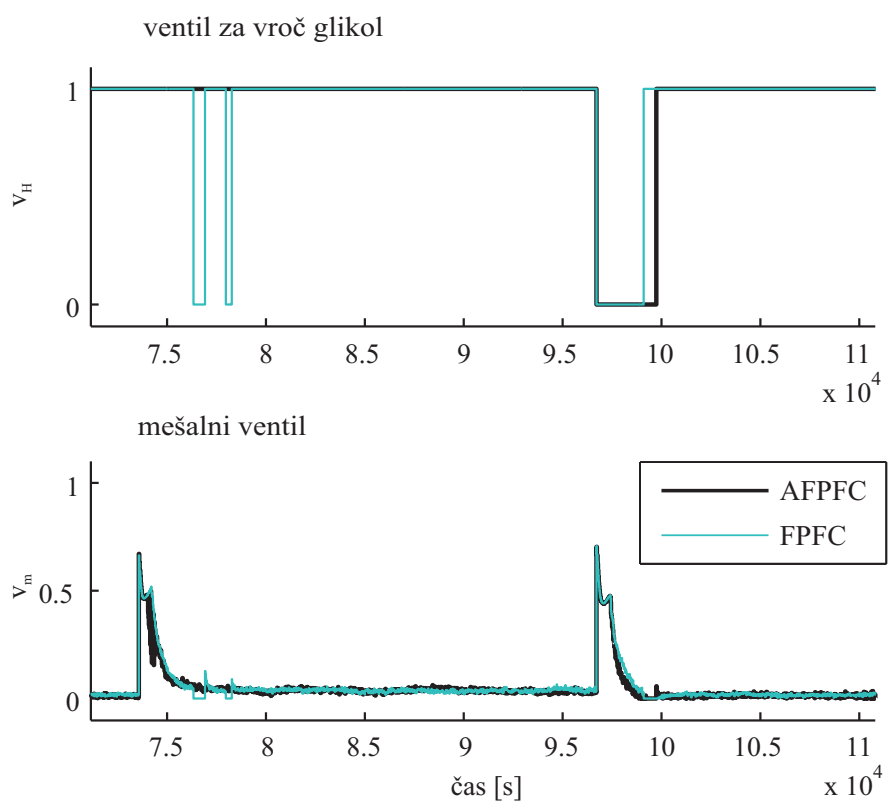
Največja razlika med AFPFC in FPFC se kaže v SSE vrednosti med izhodom mehkega modela in procesa. Ta je bila, kot pričakovano precej boljše pri adap-



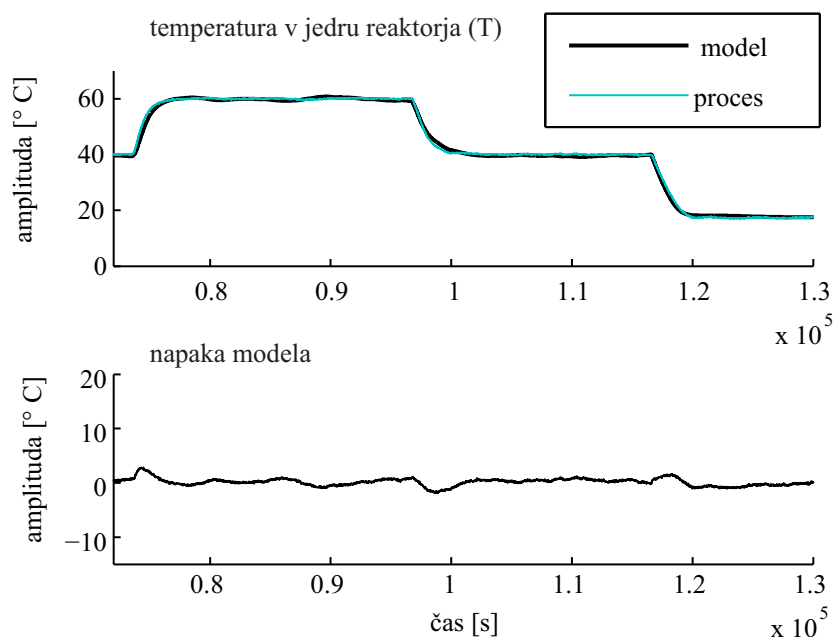
Slika 5.11: Odziv procesa po dodanem novem materialu.



Slika 5.12: Izhod regulatorja (zgornji graf) in razlika med dejanskim in želenim vhom v proces (spodnji graf) po dodanem novem materialu.



Slika 5.13: Pozicija ventila za vroč glikol (zgornji graf) in mešalnega ventila (spodnji graf) po intervalu dodajanja.

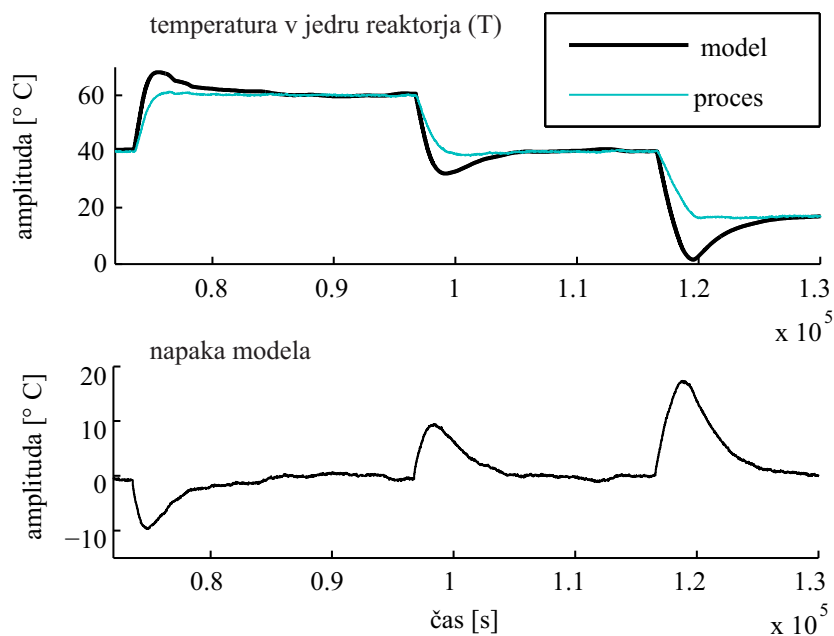


Slika 5.14: Izhod mehkega modela in procesa pri AFPFC (zgornji graf) in napaka modela (spodnji graf).

tivnem modelu. Vrednost SSE je bila računana od trenutka, ko oba regulatorja izničita vpliv motnje, pa do konca eksperimenta. Pri AFPFC je bila vrednost cenilke $1,47e3$, pri FPFC pa $6,68e4$. Ujemanje mehkega modela po dodajanju novih snovi je prikazano na sliki 5.14 za adaptivni model in sliki 5.15 za neadaptivni model.

5.2.3 Komentar

Z uporabo rekurzivnega rojenja smo zgradili adaptivni mehki prediktivno funkcijski regulator. Mehki model se je sposoben prilagoditi novim razmeram v procesu. S tem se izboljša regulacija procesa. Čeravno AFPFC daje malenkost boljše rezultate kot FPFC, pa je njegova praktična uporaba precej vprašljiva. Problem AFPFC je prav njegova adaptivna narava. Ta sicer izboljša njegovo delovanje, hkrati pa je potrebno, podobno kot pri klasičnih adaptivnih regulatorjih [16], zagotoviti nadzorni sistem, ki preverja točnost modela, pravilnost identificiranih parametrov modela in vzbujanje procesa. To posledično pomeni dražji razvoj takega algoritma. Tu se pojavi vprašanje upravičenosti vpeljave dražjega algoritma,



Slika 5.15: Izhod mehkega modela in procesa pri FPFC (zgornji graf) in napaka modela (spodnji graf).

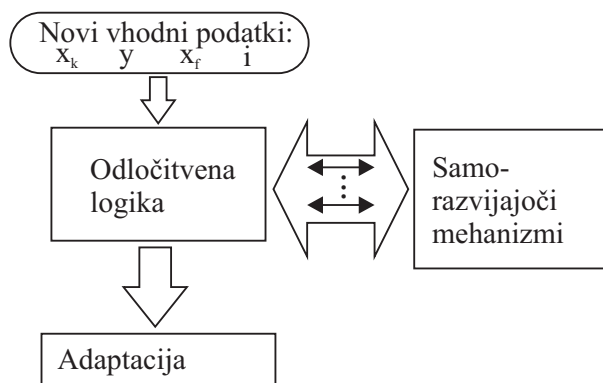
saj pri večini industrijskih procesov spremembe parametrov procesa zadovoljivo dobro obvladujemo z izgradnjo cenejših, klasičnih regulatorjev, katerih parametri so robustno nastavljeni.

6. Samorazvijajoča mehka identifikacija

Čeprav je predlagana samorazvijajoča metoda uporabljena tudi za avtomatsko nastavljanje regulatorja [189], se uporabnost samorazvijajočih metod mehke identifikacije kaže predvsem za namene detekcije napak v procesih [135, 121, 12]. Z dodatnimi mehanizmi, ki izvajajo različne operacije nad mehkim modelom, lahko rekurzivno metodo mehke identifikacije nadgradimo v samorazvijajočo. Samorazvijajoče metode morajo prilagajati število rojev in njihovo obliko ter parametre lokalnih linearnih modelov. Za prilagajanje oblike rojev in parametrov podmodelov lahko uporabimo predstavljeno rekurzivno mehko identifikacijo. Če temu dodamo še mehanizme za dodajanje, odstranjevanje, deljenje in združevanje rojev, dobimo samorazvijajočo metodo identifikacije mehkega modela. Ideje, na katerih ti mehanizmi temeljijo so različne, skupno jim je le to, da temeljijo na izkušnjah razvijalcev in nimajo stroge matematične utemeljitve. V nadaljevanju bodo predstavljeni mehanizmi, ki smo jih implementirali za doseg samorazvijajoče metode.

Predlagano samorazvijajočo identifikacijo mehkega modela (*eFuMo* - *evolving Fuzzy Model*) lahko v grobem opišemo s tremi bloki. Shema na sliki 6.1.

Shema je sestavljena iz treh blokov. Prvi in osrednji blok algoritma je CDL blok (*central decision logic*). Ta vsebuje logiko, ki kliče blok za adaptacijo in blok, ki vsebuje samorazvijajoče algoritme. Pri tem upošteva uporabniško določeno izbiro samorazvijajočih algoritmov. Uporabnik namreč lahko sam izbere, kateri samorazvijajoči mehanizmi bodo aktivni pri identifikaciji mehkega modela ter čas mirovanja (zakasnitev) samorazvijajočih algoritmov (n iz en. 6.1). V adaptacijskem bloku sta implementirana algoritma za rekurzivno rojenje in rekurzivno identifikacijo mehkega modela (poglavje 4.). Za uporabo rekurzivne mehke identifikacije v samorazvijajočem algoritmu smo dodali omejitev pri prilagajanju pozicij rojev. V praksi se velikokrat zgodi, da v nekem časovnem intervalu vzorci



Slika 6.1: Shema samorazvijajoče metode (zgornji nivo).

po večini pripadajo enemu roju. To bi pomenilo, da centri rojev konvergirajo na območje tega roja. Zato smo dodali omejitev (*betacut* postopek), kjer prilagajamo samo tiste roje, katerih trenutno proženje (normirano) je večje od nekega vnaprej določenega pragu. Ta prag je ponavadi nekje okoli 0,3.

V samorazvijajočem bloku so implementirani algoritmi in pogoji za dodajanje, odzemanje, deljenje in združevanje rojev. Ti mehanizmi in njihovo delovanje so podrobneje razloženi v naslednjih poglavjih.

6.1 Dodajanje rojev

Dodajanje rojev je eden pomembnejših mehanizmov pri izgradnji samorazvijajoče metode identifikacije. Začnemo z enim rojem in nato po potrebi dodajamo nove. Pri tem moramo pravilno inicializirati parametre novega roja. Pomembni so tudi pogoji dodajanja novih rojev. V literaturi najdemo pogoje, vezane na izhodno napako modela, razdaljo novega vzorca do obstoječih rojev in pogoj ϵ -kompletnosti oziroma pogoj, vezan na pripadnost trenutnega vzorca k obstoječim rojem.

V [72] (DFKNN) je dodajanje novega roja odvisno od razdalje trenutnega vzorca do centrov rojev (Evklidova razdalja) ter spremembe variance, ki jo nov vzorec prinese k najbližjemu roju. Za dodajanje novega roja morata biti razdalja in sprememba variance večji od določenega pragu. Poleg tega se spremlja tudi število zaporednih vzorcev, ki zadostijo zgoraj opisanemu kriteriju. Če je to število večje od določenega pragu potem se doda nov roj.

V [39] mora biti trenutni vzorec dlje od polovice minimalne razdalje med obstoječimi centri rojev, hkrati pa mora biti njegova pripadnost k obstoječim rojem manjša od določenega pragu.

Pri DENFIS-u [92] dodajanje temelji na posplošeni evklidski razdalji. Če je trenutni vzorec znotraj radija vsaj enega izmed rojev, se mehki model ne prilagodi, niti se ne doda nov roj. Če je razdalja večja, se za vzorec izračunajo vsote razdalje do vsakega centra roja in njegovega radija. Če je minimalna vsota večja od dvakratne vrednosti pragu, se doda nov roj, v nasprotnem primeru se parametri mehkega modela adaptirajo. Prag je maksimalen dopustni radij roja.

D-FNN [181] in GD-FNN [182] delujeta na pooben način. Dodajanje je pogojeno z izhodno napako modela in z razdaljo trenutnega vzorca do obstoječih centrov. Če je samo napaka presegla določeno vrednost, se prilagodijo parametri lokalnih modelov. Če je samo razdalja prestopila pragovno vrednost, se prilagodijo vrednosti pripadnostnih funkcij. Če sta napaka in razdalja večji od pragu, se doda nov roj. Pragova se s časom manjšata. S tem naj bi model, identificiran s to metodo, najprej generalno opisal proces, z zmanjšanjem pragu pa njegovo bolj podrobno obnašanje. Razlika med obema metodama je v načinu manjšanja pragov. Na podoben način deluje tudi metoda RAN [138], le da ima ta fiksne pragove.

NeuralGas [63] spremlja akumulirano napako skozi določen časovni interval. Če ta napaka preseže določen prag, se doda nov roj. Podoben princip najdemo tudi pri NeuroFAST metodi [165]. GAP-RBF [77] in SAFIS [150] dodajata nov roj glede na izhodno napako modela in razdaljo. Hkrati pa računata doprinos trenutnega vzorca k zmanjšanju napake v primeru, da bi bil ta izbran za nov center. Če so izpolnjeni vsi trije pogoji (napaka izhoda večja od pragu, razdalja večja od pragu in zadovoljiv doprinos k zmanjšanju napake), se doda nov roj.

Pri EFuNN [91, 90] je pogoj vezan na občutljivost, ki je funkcija normiranih razdalj. NFCN [110], ENFM [156], SONFIN [85], SCFNN [111], SOFNN [106] delujejo po principu ϵ -kompletnosti, kar pomeni, da maksimalna pripadnost trenutnega vzorca k rojem ne sme biti manjša od določene vrednosti. SOFNN in SCFNN imata poleg ϵ -kompletnosti implementiran tudi dodatni pogoj. Ta je pri obeh vezan na izhodno napako modela. SCFNN predvideva poleg pogoja na izhodno napako modela tudi pogoj vezan na spremembo napake modela.

Pri eTS [6] metodi je pogoj dodajanja vezan na potencial novega vzorca. Vzorec se sprejme za center novega roja, če je od najbližjega obstoječega roja oddaljen za vnaprej definiran prag in če je njegov potencial večji od potencialov obstoječih rojev. Če pogoj oddaljenosti ni izpolnjen, se najbližji center premakne na pozicijo novega vzorca. Če je izpolnjen pogoj oddaljenosti, potencial kandidata pa je nižji od potencialov centrov, se prilagodijo le parametri lokalnih modelov.

Dodajanje centrov pri FLEXFIS-u [122] je vezano na razdaljo novega vzorca do rojev. Nov roj se doda, če je najmanjša razdalja večja od definiranega koeficienta bližine.

Dodajanje je priporočljivo vezati na več pogojev. Podobno kot pri NEURO-Fast metodi smo se odločili uvesti zakasnitev (čas mirovanja) samorazvijajočih mehanizmov. To je določen interval, med katerim se ne izvaja mehanizmov dodajanja, odstranjevanja, združevanja in deljenja rojev. Adaptacija centrov in parametrov lokalnih modelov poteka ves ta čas.

Če eden od samorazvijajočih mehanizmov poseže v strukturo mehkega modela, se izvajanje le-teh ustavi za določen čas. Po preteku tega časa se ponovno pričnejo izvajati. Modelu je tako dana možnost, da ustrezno prilagodi novo strukturo. Uporabnik vnaprej določi zakasnitevž/čas mirovanja glede na vrsto in število podatkov. Pri določanju zakasnitve si lahko pomagamo s χ^2 porazdelitvijo. Po Chochranovem teoremu [38] ima vsota kvadratov spremenljivk z normalno porazdelitvijo χ^2 porazdelitev, kar v našem primeru pomeni, da ima varianca χ^2 . Iz tega teorema lahko izpeljemo število vzorcev, ki je potrebno za oceno variance podatkov na določeno natančnost. Enostavneje pa lahko določimo število vzorcev, če si pomagamo s trditvijo Kennya in Keepinga [95], ki pravi: Če iz množice vzorcev z varianco σ^2 vzamemo n -vzorcev, je pričakovana varianca $\hat{\sigma}^2$ teh vzorcev:

$$\hat{\sigma}^2 = \frac{n-1}{n}\sigma^2. \quad (6.1)$$

Če vzamemo zakasnitev 100 vzorcev (ob predpostavki, da bodo vzorci v intervalu zakasnitve pripadali istemu roju), se ocenjena varianca približa na 1 % dejanski varianci.

Pogoje za dodajanje rojev smo vezali na razdaljo trenutnega vzorca do obstoječih rojev. Problem vezanja pogojev na pripadnost je pri naši metodi predvsem v tem, da s pomočjo rojenja dobimo normirane vrednosti pripadnosti. To

pomeni, da pri prisotnosti enega roja vsak vzorec sto odstotno pripada le temu. Iz tega sledi, da se nov roj ne bi nikoli dodal. Lahko bi uporabili nenormirane pripadnosti, ki jih dobimo z Gaussovimi funkcijami (en. 4.35, en. 4.32). V tem primeru se izkaže, da metoda generira veliko več rojev, kot jih je dejansko potrebnih.

V naslednjih podpoglavjih bomo predstavili nekaj predlaganih razdalj, ki se lahko uporabijo pri pogojih za dodajanje rojev.

6.1.1 Razdalja po posameznih komponentah

Prva razdalja, ki se kaže kot logična izbira, je razdalja med posameznimi komponentami podatkovnega vektorja (x_j - j -ta komponenta podatkovnega vektorja) in i -tega centra roja (v_{ij} - j -ta komponenta centra i -tega roja):

$$d_{ij} = |x_j - v_{ij}|, \quad j = 1, 2, \dots, z + 1, \quad (6.2)$$

$$\mathbf{d}_i = [d_{i1} \ d_{i2} \ \dots \ d_{iz+1}]^T, \quad (6.3)$$

pri tem je i odvisen od izbire uporabnika. Lahko opazujemo samo najbližji roj (i je indeks najbližjega roja) ali pa vse obstoječe roje ($i = 1, \dots, c$).

Za določitev pogojev dodajanja razdalje pred zlaganjem v vektor $\mathbf{d}_i$ normiramo z diagonalnimi členi mehke kovariančne matrike i -tega roja ($\mathbf{F}_{ijj}$):

$$d_{ij} = \frac{d_{ij}}{k_n \sqrt{\mathbf{F}_{ijj}}}, \quad (6.4)$$

kjer je k_n konstanta, ki jo izbere uporabnik. Priporočena izbira je med 1,8 in 2,5 (običajno 2). Zgornja enačba pomeni, da smo razdalje po komponentah normirali s k_n -kratno vrednost standardne deviacije. S tem smo upoštevali velikost posameznega roja. Pri določanju faktorja k_n lahko upoštevamo tudi faktor prekrivanja η_m , če je le-ta različen od ena ($k_n = \sqrt{\eta_m} k_n$).

Imamo več možnosti oblikovanja pogojev dodajanja:

- Vse komponente vektorja razdalje $\mathbf{d}_i$ morajo zadoščati pogoju:

$$d_{ij} > 1, \quad j = 1, \dots, z + 1, \quad (6.5)$$

kjer je i odvisen od tega, ali gledamo razdaljo do najbližjega roja ali razdalje do vseh rojev.

- Le točno določene komponente morajo zadoščati pogoju:

$$d_{ij} > 1, \quad (6.6)$$

kjer je i odvisen od tega, ali gledamo razdaljo do najbližjega roja ali razdalje do vseh rojev. Vrednosti j določi uporabnik.

- Le določeno število komponent mora zadoščati pogoju, vendar ni pomembno katere:

$$l_{ij} = d_{ij} > 1, \quad j = 1, \dots, z + 1, \quad (6.7)$$

$$L_i = \sum_{j=1}^{z+1} l_{ji}, \quad (6.8)$$

$$L_i > n_{komp}, \quad (6.9)$$

kjer je i odvisen od tega, ali gledamo razdaljo do najbližjega roja ali razdalje do vseh rojev. Število komponent, ki morajo zadoščati pogoju, je označeno z n_{komp} .

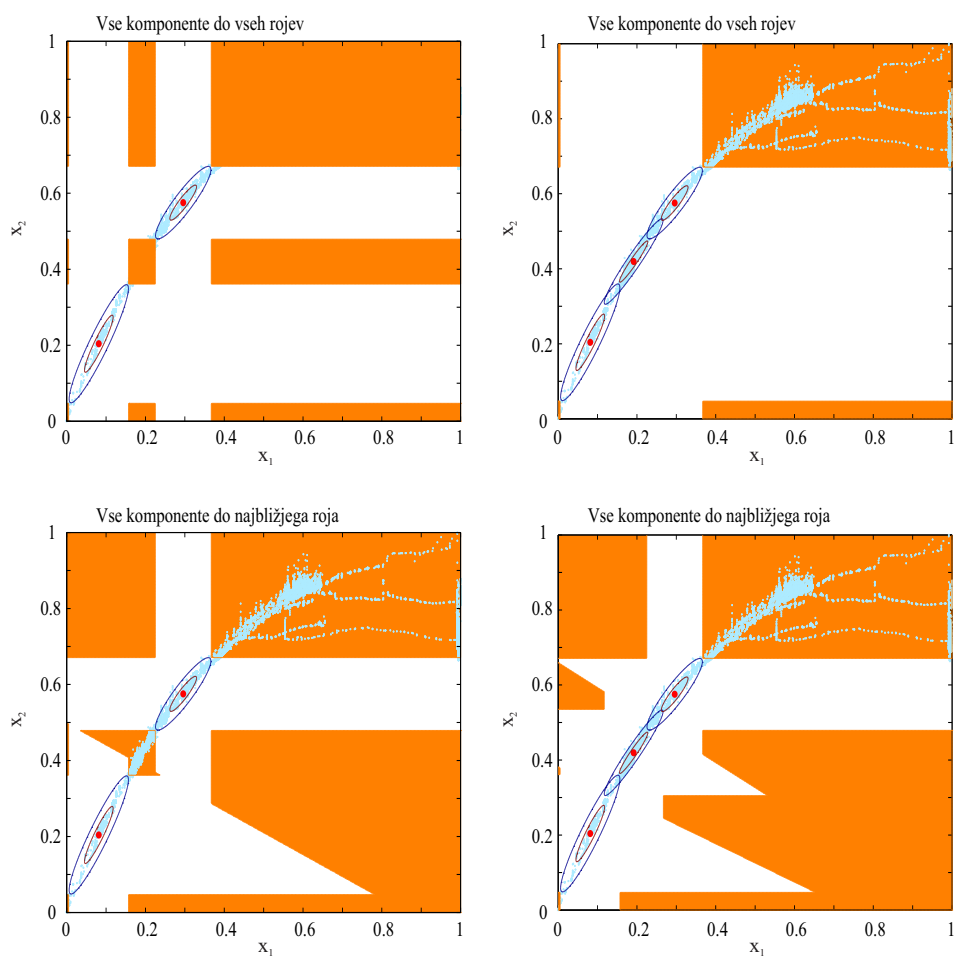
- Komponente skupaj morajo zadoščati pogoju:

$$L_i = \sum_{j=1}^{z+1} d_{ji}, \quad (6.10)$$

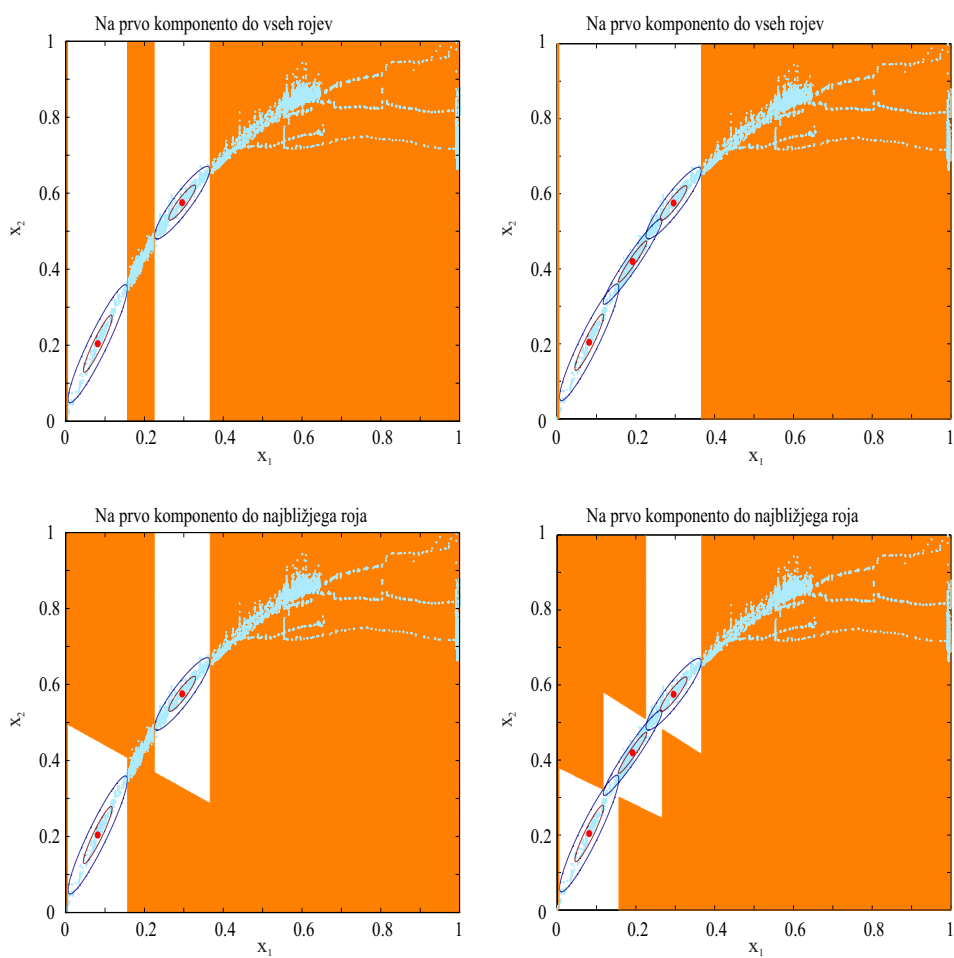
$$L_i > k(z + 1), \quad (6.11)$$

kjer je i odvisen od tega, ali gledamo razdaljo do najbližjega roja ali razdalje do vseh rojev, k je faktor skaliranja, ki je običajno ena.

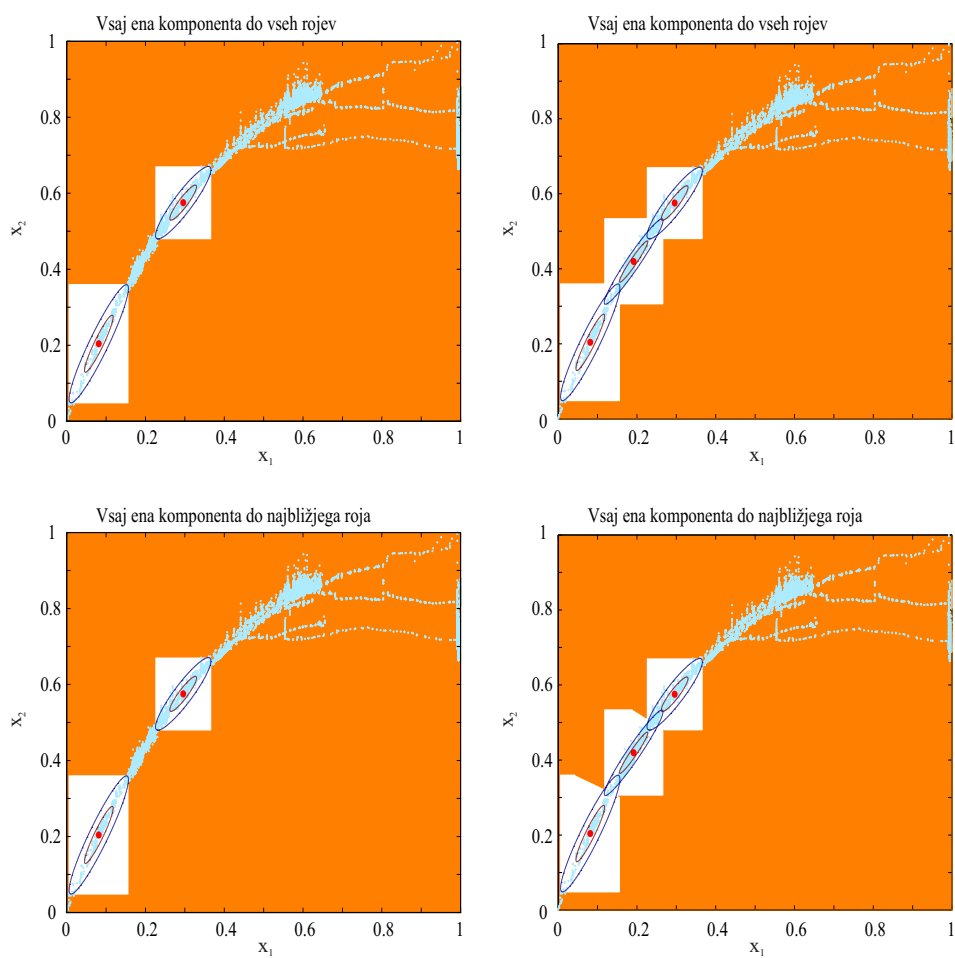
Na slikah 6.2, 6.3, 6.4 in 6.5 je prikazano območje dodajanja pri upoštevanju teh pogojev. Faktor normiranja k_n je nastavljen na 2. Če izberemo večji faktor skaliranja, se območja okoli centrov kjer ni mogoče dodati roja, povečajo. Za ponazoritev območij dodajanja smo za lažjo predstavlo predstavili na dvodimenzionalnem primeru. Območja, kjer se novi roji lahko dodajo so obarvana. Za najbližji roj smo vzeli tistega, ki je imel najmanjšo Evklidsko razdaljo do trenutnega vzorca. Na slikah so prikazane tudi ovojnice rojev 0,63 in 0,1. Na 6.5 je prikazan pogoj kjer morajo komponente skupaj zadoščati pogoju dodajanja (en. 6.11). Prikazana je vsota L_i manjša od ena, večja od ena in manjša od dva ter večja od dva. Območje kjer se roj lahko doda je v območju kjer je vsota večja od dva.



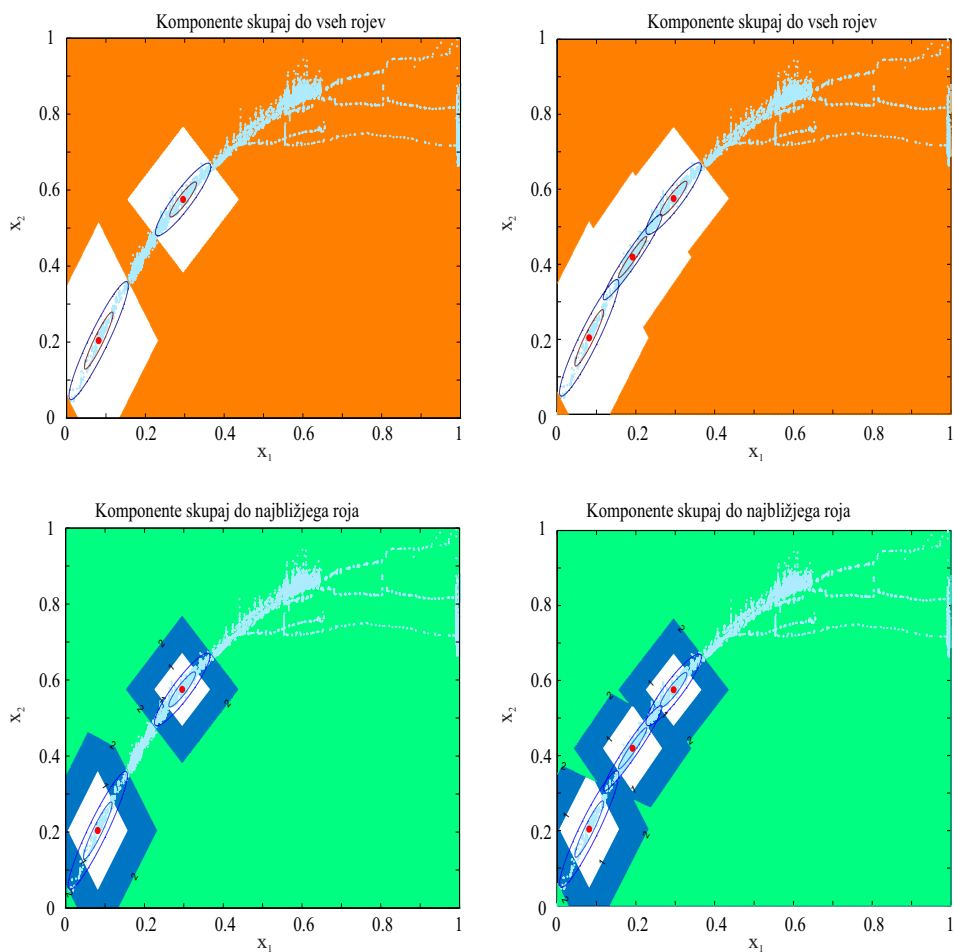
Slika 6.2: Območje dodajanja, kjer vse komponente vektorja razdalje zadoščajo pogoju dodajanja (levo za dva roja, desno za tri, spodaj za najbližji roj, zgoraj do vseh obstoječih rojev).



Slika 6.3: Območje dodajanja, kjer prva komponenta vektorja razdalje zadošča pogoju dodajanja (levo za dva roja, desno za tri, spodaj za najbližji roj, zgoraj do vseh obstoječih rojev).



Slika 6.4: Območje dodajanja, kjer ena od komponent vektorja razdalje zadošča pogoju dodajanja (levo za dva roja, desno za tri, spodaj za najbližji roj, zgoraj do vseh obstoječih rojev).



Slika 6.5: Območje dodajanja, kjer komponente vektorja razdalje skupno zadoščajo pogoju dodajanja (levo za dva roja, desno za tri, spodaj za najbližji roj, zgoraj do vseh obstoječih rojev).

6.1.2 Evklidska razdalja

Pogoj za dodajanje rojev lahko vežemo tudi na evklidsko razdaljo, ki jo normiramo z diagonalnimi elementi mehke kovariančne matrike roja. V vektor zberemo diagonalne člene mehke kovariančne matrike i -tega roja (en. 6.12), kjer je i odvisen od tega, ali gledamo razdalje do vseh rojev ali samo razdaljo do najbližjega:

$$\mathbf{s}_{i_{norm}} = [\sqrt{\mathbf{F}_{i11}} \quad \sqrt{\mathbf{F}_{i22}} \quad \dots \quad \sqrt{\mathbf{F}_{imm}}]^T, \quad (6.12)$$

kjer je m število komponent mehke kovariančne matrike ($z + 1$). Normirano razdaljo zapišemo kot:

$$d_{i_{norm}} = \frac{((\mathbf{x}(k) - \mathbf{v}_i)^T (\mathbf{x}(k) - \mathbf{v}_i))^{0,5}}{k_n (\mathbf{s}_{i_{norm}}^T \mathbf{s}_{i_{norm}})^{0,5}}. \quad (6.13)$$

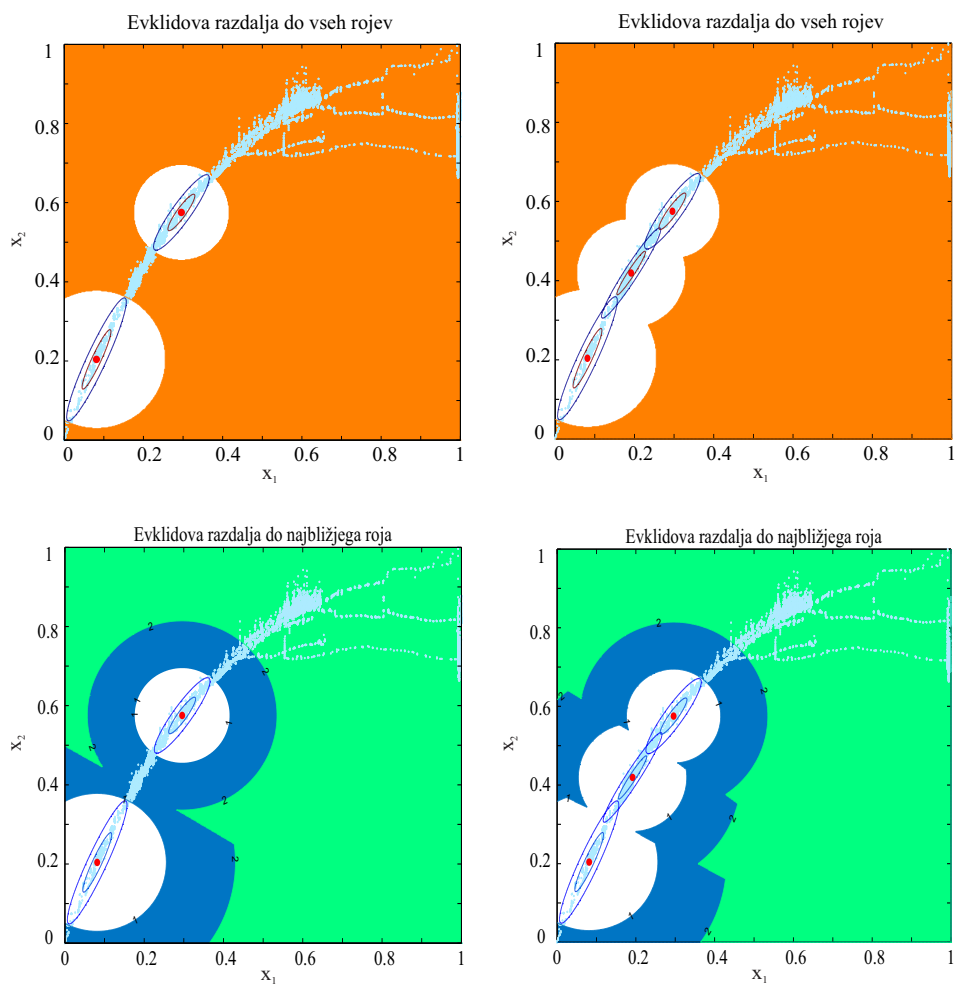
Nov roj se doda, če je razdalja $d_{i_{norm}}$ večja od ena. Ta pogoj lahko pogoj preverjamo za najbližji roj ali pa za vse obstoječe roje. Bistvene razlike ni. Na sliki 6.6 je prikazana meja za dodajanje pri tem pogoju. Na zgornjih grafih slike 6.6 je prikazano območje, kjer vsi roji zadoščajo pogoju, na spodnjih pa pogoju zadošča le najbližji roj. Tu je predstavljen pas, kjer je vrednost normirane razdalje manjša od ena, med ena in dva in večja od dva. Vrednost faktorja k_n je 2, kar je običajna vrednost. Obarvane površine predstavljajo področje, kjer se lahko dodajo novi roji. Na slikah so prikazane tudi ovojnice rojev 0,63 in 0,1.

6.1.3 Mahalanobisova razdalja

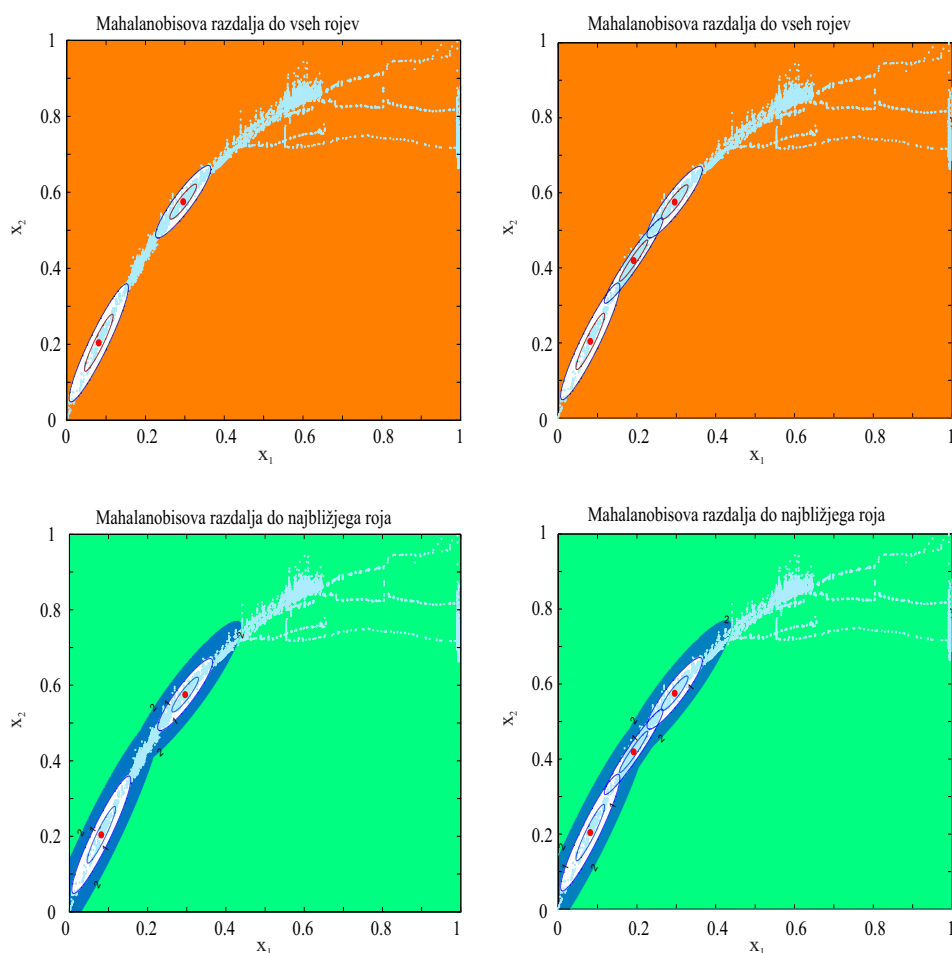
Z Mahalanobisovo razdaljo lahko najbolj natančno opišemo obliko roja. Podobno kot pri evklidovi razdalji tvorimo normalizacijski vektor $\mathbf{s}_{i_{norm}}$ (en. 6.12). Normirano razdaljo zapišemo kot:

$$d_{i_{norm}} = \frac{((\mathbf{x}(k) - \mathbf{v}_i)^T \mathbf{F}_i^{-1} (\mathbf{x}(k) - \mathbf{v}_i))^{0,5}}{k_n (\mathbf{s}_{i_{norm}}^T \mathbf{F}_i^{-1} \mathbf{s}_{i_{norm}})^{0,5}} \quad (6.14)$$

Vrednost faktorja k_n je običajno nekje med 2 in 3. Vzorec sprejmemo kot nov center, če je $d_{i_{norm}}$ večja od 1. Na sliki 6.7 je prikazano območje dodajanja za ta pogoj. Pri tem pogoju vidimo, da se območje možnega dodajanja zelo dobro prilega obliki rojev. Na zgornjih grafih slike 6.7 je prikazano območje, kjer vsi roji zadoščajo pogoju, na spodnjih pa pogoju zadošča le najbližji roj. Tu je predstavljen pas, kjer je vrednost normirane razdalje manjša od ena, med ena



Slika 6.6: Območje dodajanja, kjer Evklidova razdalja izpolnjuje pogoj dodajanja (levo za dva roja, desno za tri, spodaj za najbližji roj, zgoraj do vseh obstoječih rojev).



Slika 6.7: Območje dodajanja, kjer Mahalanobisova razdalja izpolnjuje pogoje dodajanja (levo za dva roja, desno za tri, spodaj za najbližji roj, zgoraj do vseh obstoječih rojev).

in dva in večja od dva. Vrednost faktorja k_n je 2, kar je običajna vrednost. Obarvane površine predstavljajo področje, kjer se lahko dodajo novi roji. Na slikah so prikazane tudi ovojnice rojev 0,63 in 0,1.

6.1.4 Dodatni varnostni pogoji

Iz slik, ki prikazujejo območja dodajanja rojev vidimo, da pri vseh možnih pogojih razen pri Mahalanobisovi razdalji dobimo precejšnje varnostno območje okoli roja, kjer centra ni možno dodati. Kljub temu se lahko pojavi, predvsem pri obdelavi realnih podatkov, kakšen vzorec, ki je izjemek (outlier). Na podlagi takega

vzorca se nov roj ne bi smel dodati. Zato uvedemo dodaten pogoj pri dodajanju novega roja. Ta je vezan na število zaporednih vzorcev, ki izpadejo iz roja. S tem pogojem sicer zakasnimo dodajanje novega roja, vendar pa preprečimo nepotrebno dodajanje rojev na podlagi izjemkov. Dodatni pogoj se glasi:

Nov roj dodamo v primeru, ko N -zaporednih vzorcev, ki pripadajo istemu roju, izpolnjuje izbran pogoj dodajanja na podlagi razdalje. Vzorec vedno pripada najbližjemu roju.

Če upoštevamo dodatni pogoj se verjetnost dodajanja napačnega roja zmanjša na $P(x)^N$, kjer je $P(x)$ verjetnost brez upoštevanja dodatnega pogoja. Število vzorcev N je ponavadi med 5 in 10.

Naslednji varnostni ukrep, ki ga lahko uvedemo, je spremenljiv faktor k_n . Ko je nov roj ustvarjen, je ta večji za ustvarjen roj. S številom vzorcev, ki pripadajo posameznemu roju, pada do običajne vrednosti. Ta pogoj še ni bil implementiran v predlagani metodi.

6.1.5 Inicializacija novega roja

Ko ugotovimo, da nov vzorec ustreza pogojem za dodajanje, ga sprejmemo kot začetni center roja. Potrebna je še inicializacija obsega roja (mehke kovariančne matrike) in inicializacija parametrov lokalnega modela. V literaturi najdemo več načinov inicializacije teh parametrov: pripadnostne funkcije in s tem velikost novega roja inicializirajo glede na oddaljenost do najbližjega roja (DENFIS [92], D-FNN [181]), fiksno z začetno vrednostjo, ki jo poda uporabnik (SCFNN [111], SONFIN [85]), kot povprečje vseh obstoječih pripadnostnih funkcij (exTS [15, 5], ENFM [156]), jih prevzamejo od najbližjega roja ali pa jih postavijo na nič (FLEXFIS [122]).

Najbolj smiselna inicializacija je glede na oddaljenost do najbližjega roja, saj s tem pokrijemo območje med novim in najbližjim centrom. Mehko kovariančno matriko novega roja inicializiramo s pomočjo enačbe 4.49 kot diagonalno matriko, kjer so diagonalni členi izračunani na podlagi želenega proženja v posamezni dimenziji vhodno-izhodnega prostora, ki ga poda uporabnik (ϵ_β):

$$\sigma_j^2 = -\frac{d_j^2}{2\eta_m \ln(\epsilon_\beta)}, \quad (6.15)$$

kjer je d_j razdalja med j -to komponento centra novega roja in njemu najbližjega roja. Pri takšnem računanju mehke kovariančne matrike moramo biti pozorni na roja, ki imata centra na isti premici, vzporedni z eno od osi prostora. Takrat bo ena od razdalj d_j enaka nič. V tem primeru vzamemo diagonalni člen iz mehke kovariančne matrike najbližjega roja. Pri inicializaciji moramo vedeti, da z uporabo take diagonalne mehke kovariančne matrike pripadnost (en. 4.17) celotnega vektorja vzorca ne bo ϵ_β , saj se pripadnosti po komponentah množijo. Če želimo, da bo pripadnost celotnega vektorja enaka želeni moramo diagonalno kovariančno matriko napihniti. Pomnožimo jo s številom njenih vrstic oziroma stolpcev. V praksi tega ni potrebno početi.

Začetni parametri linearnega lokalnega modela novega roja v [6] so nastavljeni na vrednost nič za lokalne mehke najmanjše kvadrate in na uteženo povprečje parametrov obstoječih lokalnih modelov za globalne mehke najmanjše kvadrate. Uteži so pripadnosti novega roja k ostalim. V [92] so za začetne vrednosti uporabljeni parametri lokalnega modela najbližjega roja.

Parametre linearnega modela je smiselno inicializirati z uteženim povprečjem (en. 6.16) kot predlaga [6]. Pri tem je dobro upoštevati ne le pripadnosti novega roja, ampak tudi točnost parametrov obstoječih lokalnih modelov.

$$\theta_{i+1_j} = \frac{\sum_{i=1}^c \omega_{i_j} \theta_{i_j}}{\sum_{i=1}^c \omega_{i_j}} \quad j = 1, \dots, z + 1. \quad (6.16)$$

V utežeh upoštevamo pripadnost in varianco posameznega parametra:

$$\omega_{i_j} = \beta_i \frac{1}{\sigma_{P_{i_j j}}^2}, \quad (6.17)$$

kjer je $\sigma_{P_{i_j j}}^2$ diagonalni člen kovariančne matrike najmanjših kvadratov.

Ko dodamo nov roj, lahko povečamo kovariančno matriko najmanjših kvadratov ostalih lokalnih modelov. V [6] je predlagano množenje kovariančnih matrik s faktorjem $\rho = \frac{c^2+1}{c^2}$, kjer je c število rojev pred dodajanjem novega roja.

6.2 Odstranjevanje rojev

Mehanizem odstranjevanja rojev služi za odstranjevanje starih rojev, ki niso več aktualni. V našem primeru ta mehanizem ni tako pomemben. Predlagan mehki

rekurzivni algoritem rojenja ima vgrajene faktorje pozabljanja, ki zagotavljajo osveževanje pozicij rojev in njihove mehke kovariančne matrike. Kljub temu se lahko zgodi, da se ustvari nov roj v nekem delu vhodno-izhodnega prostora, kjer je prisotnih malo vzorcev. Ti vzorci so običajno posledica napak v meritvah. Načeloma lahko pustimo, da ta roj konvergira k pravim podatkom. V tem primeru generiramo več rojev, kot jih je mogoče potrebno za opis vhodno-izhodnega prostora. Da se taki roji hitro izbrišejo, smo v predlagani eFuMo metodi implementirali tudi mehanizme za odstranjevanje rojev.

V [39] je roj odstranjen, če mu v določenem časovnem intervalu ne pripada noben vzorec. Časovni interval poda uporabnik. Težava nastane, če se identificirani proces dolgo časa zadržuje v enem samem območju. Roji, ki opisujejo ostala območja se v tem primeru lahko izbrišejo.

V [181], [182], [77], [150] in [106] (D-FNN, GD-FNN, GAP-RBF, SAFIS in SOFNN) je odstranjevanje rojev vezano na napako modela. V [181] (D-FNN) uveden *error reduction ratio*, kjer izračuna doprinos posameznega roja k celotni napaki modela. Če roj bistveno ne vpliva na zmanjšanje napake, ga odstranijo. Podoben pogoj je predlagan v [182] (GD-FNN). Poleg *error reduction ratio* uvedejo še občutljivostni indeks *sensitivity index*. Roje odstranjujejo na podlagi vrednosti teh dveh spremenljivk. V [150] (SAFIS) je izpeljana formula za oceno spremembe napake, v kolikor določen roj odstranimo iz strukture. Če je ta vrednost manjša od določenega pragu, se roj izbriše. Pri SOFNN metodi [106] odstranjevanje rojev temelji na konceptu *optimal brain surgeon approach* [73, 108]. S pomočjo drugih odvodov je izračunana občutljivost napake na spremembo parametrov lokalnih modelov. Če je ta sprememba dovolj majhna, je roj odstranjen. Ta način je zelo podoben načinu odstranjevanja v [181] in [182].

V [63] (NeuralGas) so odstranjeni tisti roji, ki so bili generirani pred $t - a_{max}$. t je trenutni čas, a_{max} pa podan prag.

V [15, 5] (exTS) temelji odstranjevanje na podporni množici in starosti roja. Podporna množica roja je definirana kot število vzorcev, ki pripadajo roju. Vzorec vedno pripada najbližjemu roju. Starost je definirana kot razmerje med akumuliranim časom prihajanja vzorcev k roju in trenutnim časom. Roji se brišejo glede na razmerje podporne množice roja proti vsem vzorcem in glede na starost. Isti pogoji so uvedeni v +eTS [8]. Pri +eTS metodi so poleg omenjenega pogoja

dodali še pogoj *utility*-ja. Tu se vsota vseh proženj roja normira s časom kreacije roja. Pogoje odstranjevanja rojev vežejo na odstopanje zgoraj omenjenih vrednosti od povprečne vrednosti, meja dovoljenega odstopanja pa je standardna deviacija. Pri majhnem številu rojev postavitev meje na standardno deviacijo na najboljša rešitev. Pri +eTS je bil uveden tudi minimalni pogoj, ki ga mora roj izpolnjevati za svoj obstoj. To pomeni, da morajo od trenutka nastanka novega roja v intervalu desetih vzorcev njemu pripasti vsaj trije vzorci.

Pri DFKNN metodi [72] je brisanje pogojeno z minimalnim številom vzorcev, ki morajo pripadati roju. Če ima roj premajhno podporno množico, se izbriše. Minimalno število vzorcev določi uporabnik sam. Drugi pogoj je vezan na časovni interval. Če v določenem časovnem intervalu po kreaciji novega roja ni nobenega vzorca, ki bi pripadal temu roju, se ta roj izbriše.

Pri EFuNN [91, 90] je brisanje roja odvisno od starosti roja in vsote proženj. Starost rojev je definirana s številom vzorcev od kreacije roja do trenutnega vzorca. Če je starost večja od določenega pragu in je vsota proženj roja manjša od starosti roja, pomnožene z uporabniško definirano konstanto (iz področja $[0 \ 1]$), je roj izbrisan.

V predlagani eFuMo metodi smo implementirali minimalni pogoj za obstoj roja, predlaganega pri +eTS metodi in pogoj brisanja, ki temelji na razmerju med podporno množico (N_{p_i}) in starostjo roja (a_i). Starost roja definiramo kot:

$$a_i = k - k_i, \quad (6.18)$$

kjer je k trenutni vzorec, k_i pa število vzorcev do trenutka, ko je bil nov i -ti roj ustvarjen.

Minimalni pogoj za obstoj roja lahko zapišemo v obliki:

$$\check{C}e N_{trh} \geq N_{p_i}(a_{trh}), \text{ potem odstrani roj,} \quad (6.19)$$

kjer je N_{trh} minimalno število pripadajočih vzorcev (podporne množice), a_{trh} prag starosti roja in $N_{p_i}(a_{trh})$ vrednost podporne množice roja, ko ta doseže prag starosti a_{trh} . Prag starosti in minimalno število pripadajočih vzorcev poda uporabnik.

Pogoj med razmerjem podporne množice in starostjo roja se glasi:

$$\frac{N_{p_i}}{a_i} > \epsilon_{Na} \frac{1}{c} \sum_{i=1}^c \frac{N_{p_i}}{a_i}, \quad (6.20)$$

kjer je ϵ_{Na} uporabniško določena konstanta, manjša od ena. Pogoj je uporaben predvsem, ko imamo opravka z večimi roji. Vsi roji, ki izpolnjujejo pogoj, se obdržijo v strukturi mehkega modela. Razlika med implementiranim pogojem in pogojem v eTS+ je v tem, da pri našem pogoju za mejo brisanja upoštevamo procent povprečne vrednosti kvocienta $\frac{N_{p_i}}{a_i}$, medtem ko v eTS+ za mejo upoštevajo varianco kvocienta. Slednje je precej nesmiselno, saj imamo v večini primerov opravka z majhnim številom rojev. Računanje variance na majhnem številu vrednosti pa je nesmiselno.

6.3 Združevanje rojev

V tem razdelku, bomo predstavili mehanizem združevanja rojev, ki je implementiran v predlagani metodi. Metoda integrira tako nenadzorovani, kot tudi "nadzorovani" način združevanja.

V DKFNN se spremlja pozicija centrov rojev. Če gresta poziciji proti istemu področju jih je potrebno združiti. Uporabljena mera podobnosti je predstavljena v [62]. Temelji na pripadnosti vzorcev k rojem in je podobna korelaciji med proženji rojev. Združevanje na podlagi korelacije preteklih proženj je omejeno tudi v [91] (EFuNN). Sam mehanizem ni opisan. V članku je predlagano združevanje na podlagi maksimalnega radija. Združimo lahko sosednje roje, katerih pa skupni radij ne sme preseči definiranega pragu maksimalnega radija. Pri ENFM sta dva roja združena, če je pripadost enega v drugem in obratno večja od v naprej definiranega pragu. V [106] (SOFNN) združijo dva roja, če sta poziciji obeh centrov isti, kar pa je v praksi skoraj nemogoče. Omenjajo pa tudi možnost uporabe mere podobnosti iz [179]. V FLEXFIS+ predlagajo računanje presečišča pripadnostnih funkcij v vsaki dimenziji. Iz tega določijo indeks prekrivanja. Če je indeks prekrivanja večji od določenega pragu, se roja združita. Poleg tega predlaga tudi sistem združevanja glede na mero podobnosti med roji. Pri tem upošteva tudi parametre lokalnih modelov. Parametre lokalnih modelov zloži v vektro in primerja kot med njimi. Če je kot majhen in podobnost med rojema

visoka sta roja združena. Definira tudi svoj indeks podobnosti.

Pri predlagani metodi smo uporabili mehanizem združevanja, ki temelji na normirani razdalji med centri. Tako kot pri dodajanju smo uporabili Mahalanobisovo razdaljo (možna je uporaba tudi drugih razdalj):

$$d_{ik}^2 = (\mathbf{v}_i - \mathbf{v}_k)^T F_i^{-1} (\mathbf{v}_i - \mathbf{v}_k), \quad i, k = 1, \dots, c \quad i \neq k. \quad (6.21)$$

Razdaljo normiramo s pomočjo normalizacijskega vektorja $\mathbf{s}_{i_{norm}}$ (en. 6.12):

$$d_{norm_{ik}} = \sqrt{\frac{d_{ik}^2}{2\mathbf{s}_{i_{norm}}^T F_i^{-1} \mathbf{s}_{i_{norm}}}} \quad (6.22)$$

Roja združimo, če sta obe normirani razdalji $d_{norm_{ik}}$ in $d_{norm_{ki}}$ manjši od definirane pragu k_d :

$$d_{norm_{ik}} < k_d \quad \text{in} \quad d_{norm_{ki}} < k_d \quad (6.23)$$

Prag k_d lahko določimo kot:

$$k_d = \sqrt{-\ln(\epsilon_\beta)}, \quad (6.24)$$

kjer je ϵ_β uporabniško določena konstanta, običajno nastavljena na 0,9.

Ko je z enim rojem možno združiti več rojev, z njim združimo tistega, ki da manjšo sled mehke kovariančne matrike skupnega roja.

Parametri združenega roja so običajno inicializirani z uteženim povprečjem [156], redko z običajnim povprečjem [91]. Center združenega roja določimo kot:

$$\mathbf{v}_{new} = \frac{N_{p_i} \mathbf{v}_i + N_{p_k} \mathbf{v}_k}{N_{p_i} + N_{p_k}}, \quad (6.25)$$

kjer sta i in k indeksa rojev, ki ju združimo, N_p pa število pripadajočih vzorcev rojema (podporna množica). Po istem postopku izračunamo vsoto preteklih proženj (s) in trenutek kreacije roja. Za združeni roj izračunamo število pripadajočih vzorcev:

$$N_{p_{new}} = \frac{N_{p_i} s_i + N_{p_k} s_k}{s_i + s_k}. \quad (6.26)$$

Parametre linearnega podmodela izračunamo z uteženim povprečjem:

$$\theta_{new_j} = \frac{\omega_{i_j} \theta_{i_j} + \omega_{k_j} \theta_{k_j}}{\omega_{i_j} + \omega_{k_j}} \quad j = 1, \dots, z + 1, \quad (6.27)$$

kjer je ω utež. V uteži upoštevamo število pripadajočih vzorcev N_p in varianco posameznega parametra.

Za inicializacijo mehke kovariančne matrike upoštevamo enačbo, izpeljano v [156]. Izpeljana enačba temelji na ideji, ki je predstavljena v [141] in uporabljena tudi v [120]:

$$\begin{aligned} \mathbf{F}_{new} = & \frac{1}{(N_{p_i} + N_{p_k})^3} ((N_{p_i}^3 + 2N_{p_i}^2 N_{p_k} + N_{p_i} N_{p_k}^2) \mathbf{F}_i + \\ & + (N_{p_k}^3 + 2N_{p_k}^2 N_{p_i} + N_{p_k} N_{p_i}^2) \mathbf{F}_k + \\ & + (N_{p_i}^2 N_{p_k} + N_{p_i} N_{p_k}^2) (\mathbf{v}_i - \mathbf{v}_k)(\mathbf{v}_i - \mathbf{v}_k)^T). \end{aligned} \quad (6.28)$$

Za “nadzorovano” učenje upoštevamo parametre lokalnih modelov, podobno kot pri FLEXFIS+ metodi [120]. eFuMo upošteva napako predikcije modela in ima tri mehanizme za detekcijo rojev, ki so ustrezni za združevanje: kotni pogoj, korelacijski pogoj in pogoj, vezan na razmerje razdalj. Združena sta lahko le dva roja, ki ustrezata enemu od pogojev. Poleg tega mora biti razlika med predikcijo njenih lokalnih modelov pod določenim pragom. Združita se lahko le roja, ki imata podporno množico od zadnje spremembe števila rojev večjo od podanega pragu.

Korelacijski pogoj je vezan na korelacijski koeficient med proženji rojev. Spremlja se produkt $\beta_{ij}(k) = \beta_{ij}(k-1) + \beta_i(k)\beta_j(k)$, $\beta_{ii}(k) = \beta_{ii}(k-1) + \beta_i(k)\beta_i(k)$ iz katerega se izračuna korelacijski koeficient:

$$C_{ij}(k) = \frac{\beta_{ij}}{\beta_{ii}^{0,5} \beta_{jj}^{0,5}}, \quad (6.29)$$

Če koeficient $C_{ij}(k)$ preseže določeno vrednost (običajno 0,9), se lahko roja i in j združita. Za pravilno računanje Pearsonovega korelacijskega koeficienta bi bilo potrebno upoštevati še povprečne vrednosti proženj. Ker nam pogoj služi le za detekcijo možnih parov rojev za združitev, smo v našem primeru centriranje proženj z njihovimi povprečnimi vrednostmi izpustili.

Pogoj, vezan na razmerje razdalj detektira možne pare rojev za združitev na podlagi razmerja razdalj:

$$\begin{aligned} d_{ik} = & \sqrt{(\mathbf{v}_i - \mathbf{v}_k)^T \det(F_i)^{\frac{1}{p}} F_i^{-1} (\mathbf{v}_i - \mathbf{v}_k)}, \\ K_d = & 1 - \frac{\min(d_{ik}, d_{ki})}{\max(d_{ik}, d_{ki})}. \end{aligned} \quad (6.30)$$

Če je razmerje K_d manjše od določenega pragu (običajno 0,05) in korelacijski koeficient dosega vsaj polovico korelacijskega pragu, se lahko roja i in j združita.

Kotni pogoj določi pare rojev na podlagi kotov njihovih parametrov. Najprej se določi vektor maksimalnih absolutnih vrednosti parametrov:

$$\theta_{max} = \left[\max_{1 \leq j \leq c} |\theta_{j1}|, \dots, \max_{1 \leq j \leq c} |\theta_{jr}| \right] \quad (6.31)$$

Nato se parametri lokanih modelov normirajo s tem vektorjem. Na podlagi normiranih parametrov se izračunajo koti med njimi:

$$\alpha_{ij_k} = |\arctan(\theta_{ni_k}) - \arctan(\theta_{nj_k})|, \quad (6.32)$$

kjer je k indeks parametra, θ_{ni_k} k -ti normiran parameter i -tega lokalnega modela ($\theta_{ni_k} = \frac{\theta_{i_k}}{\theta_{max_k}}$). Če so koti α_{ij_k} , $k = 1, \dots, r$, kjer je r število parametrov lokalnega modela manjši od določenega pragu (običajno 2 stopinji), in korelacijski koeficient dosega vsaj polovico korelacijskega pragu, se lahko roja i in j združita.

Zgornji pogoji se izračunajo le za roje, pri katerih je podporna množica od zadnje spremembe števila rojev večja od določene vrednosti, ki jo poda uporabnik. Ko so identificirani možni pari za združitev, algoritem preveri, če sta si roja v paru sosednja (če med njima ni nobenega roja). Če si roja nista sosednja, ju izključi iz nadaljne obdelave.

Nato na preostalih parih izračunamo odstopanje obeh lokalnih modelov:

$$\begin{aligned} \mathbf{x}_1 &= [1, \bar{u}_1, \dots, \bar{u}_z]^T, \\ e_1 &= |\theta_i^T \mathbf{x}_1 - \theta_j^T \mathbf{x}_1|, \\ e_2 &= \sum_{r=2}^z |\theta_{i_r}(x_{1_r} + 2\sigma_{u_{r-1}}) - \theta_{j_r}(x_{1_r} + 2\sigma_{u_{r-1}})| + |\theta_{i_1} - \theta_{j_1}|, \\ e_3 &= \sum_{r=2}^z |\theta_{i_r}(x_{1_r} - 2\sigma_{u_{r-1}}) - \theta_{j_r}(x_{1_r} - 2\sigma_{u_{r-1}})| + |\theta_{i_1} - \theta_{j_1}|, \\ e &= \frac{1}{(2 \cdot 1,7 \cdot 3) \sigma_y} \sum_{r=1}^3 e_r, \end{aligned} \quad (6.33)$$

kjer je $\bar{u}$ povprečna vrednost vhodne spremenljivke, σ_u njena standardna deviacija, σ_y standardna deviacija izhoda procesa, z število vhodov, j in i indeksa rojev, θ_i vektor parametrov lokalnega modela i -tega roja in θ_{i_r} r -ti parameter tega lokalnega modela. V zadnji enačbi 6.33 imenovalec sestavlja produkt števil $2 \cdot 1,7 \cdot 3$. Številka 3 je v imenovalcu zaradi računanja povprečja treh napak. Številki 2 in 1,7 sta v imenovalcu zaradi normiranja z izhodno varianco. Če rekurzivno

računano izhodno varianco pomnožimo z 1,7, z njo zajamemo približno zgornjo polovico signala (od povprečne vrednosti do maksimalne vrednosti). Da bi dobili celotno območje izhodnega signala, to vrednost pomnožimo še z 2. Variance in povprečne vrednosti spremenljivk se računajo rekurzivno po enačbi:

$$\begin{aligned}\sigma_x^2(k) &= \frac{1}{k} ((k-1)(\sigma^2(k-1) + \bar{x}(k-1)^2) + x(k)^2) - \frac{1}{k^2} ((k-1)\bar{x}(k-1) + x(k))^2, \\ \bar{x}(k) &= \frac{1}{k} ((k-1)\bar{x}(k-1) + x(k)).\end{aligned}\tag{6.34}$$

Par, kjer se lokalna modela najmanj razlikujeta (najmanjša vrednost e), se združi. Razlika mora biti manjša od vnaprej določene vrednosti. Center novega roja se postavi na sredino zgornje in spodnje meje obeh rojev:

$$\begin{aligned}\mathbf{d}_1 &= \mathbf{v}_i - \mathbf{v}_j, \\ \mathbf{v}'_i &= \mathbf{v}_i + \text{sign}(\mathbf{d}_1)\mathbf{s}_{i\text{norm}}, \\ \mathbf{v}'_j &= \mathbf{v}_j - \text{sign}(\mathbf{d}_1)\mathbf{s}_{j\text{norm}}, \\ \mathbf{d}_2 &= \frac{\mathbf{v}'_i - \mathbf{v}'_j}{2}, \\ \mathbf{v}_{\text{new}} &= \mathbf{v}_j + \mathbf{d}_2.\end{aligned}\tag{6.35}$$

Mehka kovariančna matrika se inicializira kot vsota obeh mehkih matrik. Podporni množici se tudi seštejeta. Parametri lokalnih modelov se inicializirajo kot povprečna vrednost parametrov obeh lokalnih modelov. Čas kreiranja novega roja se nastavi na čas kreiranja starejšega roja. Vsota preteklih pripadnosti se inicializira na večjo vrednost obeh rojev.

6.4 Deljenje rojev

Deljenje rojev je običajno namenjeno fini nastavitvi mehkega modela, ki dodaja roje v področje, kjer je napaka višja kot določena meja. Koncept deljenja je uporabljen v [126] in [1]. V prvem primeru je uporabljena Chernoffnova mera, v drugem pa mera zvestobe (*fidelity measure*). V [119] trdijo, da je deljenje na podlagi teh pristopov slabo in počasno. Zato je v [119] predlagan skupen scenarij za deljenje in združevanje rojev, ki temelji na kombinirani BIC (*Bayesian information criterion*) vrednosti. Vrednost, izračunana s pomočjo Gaussove gostotne

funkcije in faktorja prekrivanja rojev. Roji, primerni za deljenje so najdeni s pomočjo poskušanja. Pri NeuroFAST metodi [165] so roji deljeni glede na povprečno kvadratno napako (MSE). Algoritem preveri napako na vsakih P korakov in deli roj z največjo napako se deli.

Mehanizem deljenja pri predlagani eFuMo metodi temelji na relativni napaki modela, ki jo roj zbere v določenem obdobju. Napaka je izračunana za vsak vzorec, ki ne zadostuje pogoju dodajanja roja na podlagi razdalje. Najprej se izračuna relativna napaka modela:

$$e(k) = \frac{|y_m(k) - y(k)|}{3,4\sigma_y}, \quad (6.36)$$

kjer je y dejanski izhod procesa, y_m izhod modela in σ_y standardna deviacija izhoda procesa. Varianca izhoda je množena s faktorjem 3,4. S tem dobimo približno celotno območje izhodnih vrednosti. Napaka je nato razdeljena med roje glede na njihovo proženje:

$$e_{sum_i}(k) = e_{sum_i}(k-1) + \beta_i e(k), \quad (6.37)$$

kjer je β_i proženje i -tega roja.

Pri preverjanju pogoja cepljenja algoritem poišče roj z največjo napako. Če je njegova napaka normirana z N (število vzorcev uporabljenih za izračun napake) večja od definirane pragu, in če je roj od zadnje spremembe števila rojev zbral dovolj podporne množice (običajno 20 vzorcev), se le-ta lahko deli. Prag napake, pri kateri se roj lahko deli, je definiran z maksimalnim (e_{max}) in minimalnim pragom (e_{min}) ter konstanto upadanja (T), podobno kot v [181]:

$$e_{trh} = \max(e_{max} \exp(-N/T), e_{min}). \quad (6.38)$$

Pozicija cepljenega roja je izračunana s pomočjo diagonalnih elementov mehke kovariančne matrike (vektor $\mathbf{s}_{i_{norm}}$):

$$\begin{aligned} \mathbf{v}_{i1} &= \mathbf{v}_i + 0,5\mathbf{s}_{i_{norm}}, \\ \mathbf{v}_{i1} &= \mathbf{v}_i - 0,5\mathbf{s}_{i_{norm}}, \end{aligned} \quad (6.39)$$

kjer je i indeks roja, ki se deli. Nove pozicije centrov lahko izračunamo tudi na podlagi SVD (*singular value decomposition*) [155]. Mehka kovariančna matrika, podpora in vsota preteklih pripadnosti, so inicializirane kot polovica teh vrednosti originalnega roja. Oba roja obdržita čas kreiranja na isti vrednosti kot v originalnem roju.

6.5 Delovanje eFuMo metode

V prejšnjih poglavjih smo opisali mehanizme, ki so implementirani v eFuMo metodi. Potek same metode kontrolira CDL blok. Shema bloka je predstavljena na sliki 6.8. Vhodi v eFuMo algoritem so regresijski vektor ($\mathbf{x}_k$), vektor za rojenje ($\mathbf{x}_f$), izhod identificiranega procesa (y) in številka zaporednega vzorca (i). Algoritem najprej preveri, če so izpolnjeni pogoji za klicanje samorazvijajočih algoritmov (blok 1), torej če je od zadnje spremembe v številu rojev preteklo dovolj vzorcev (čas mirovanja). Če pogoj ni izpolnjen, se izvajanje samorazvijajočih algoritmov preskoči.

Samorazvijajoči algoritmi se izvajajo glede na njihovo prioriteto. Najvišjo prioriteto ima dodajanje rojev, sledi odstranjevanje, nadzorovano združevanje, nenadzorovano združevanje in deljenje rojev. Uporabnik določi, kateri samorazvijajoči algoritmi se bodo izvajali. Pred izvajanjem CDL blok vedno preveri ali je algoritem omogočen ali ne (bloki 3, 9, 12, 15, 18). V primeru, da je kateri izmed samorazvijajočih mehanizmov spremenil število rojev, se klicanje naslednjih samorazvijajočih mehanizmov prekine, izvajanje metode pa se nadaljuje z adaptacijo oziroma preverjanjem minimalnega pogoja za obstoj roja (bloki 6, 11, 14, 17, 22). Pred tem se ponastavijo nekateri parametri (blok 32), ki služijo za nadzor rojev, kot so: podporna množica rojev in vsota pripadnosti od zadnje spremembe števila rojev, napaka, uporabljena pri algoritmu deljenja, spremenljivke uporabljene za izračun korelacijskih koeficientov ...

CDL najprej kliče algoritem za dodajanje rojev, če so pogoji za klic izpolnjeni. Trenutno število rojev c mora biti manjše od števila dopustnih rojev c_{trh} , ki ga določi uporabnik (blok 2). Če ta pogoj ni izpolnjen, se algoritem dodajanja preskoči. Uporabnik lahko izbira med dvema načinoma dodajanja roja. Pri prvem se doda dopustno število rojev ob prvem klicu eFuMo algoritma. Pri drugem načinu se pri začetnem klicu metode doda prvi roj, katerega center je v prvem podatkovnem vzorcu (vektor rojenja), ostali roji pa se dodajo kasneje glede na pogoje dodajanja (blok 5), ki so opisani v poglavju 6.1. Če trenutni vektor rojenja izpolnjuje pogoj za dodajanje glede na razdaljo in ne pogoja zaporednih vzorcev (poglavje 6.1.4), si CDL blok to zapomni (tmp vrednost v bloku 7). Ta informacija se kasneje uporabi pri računanju napake modela za deljenje rojev



(blok 19).

Mehanizmu dodajanja sledi mehanizem odstranjevanja rojev (blok 10). Mehanizem preverja pogoj 6.20 in se izvede, če je število rojev večje od ena (blok 8).

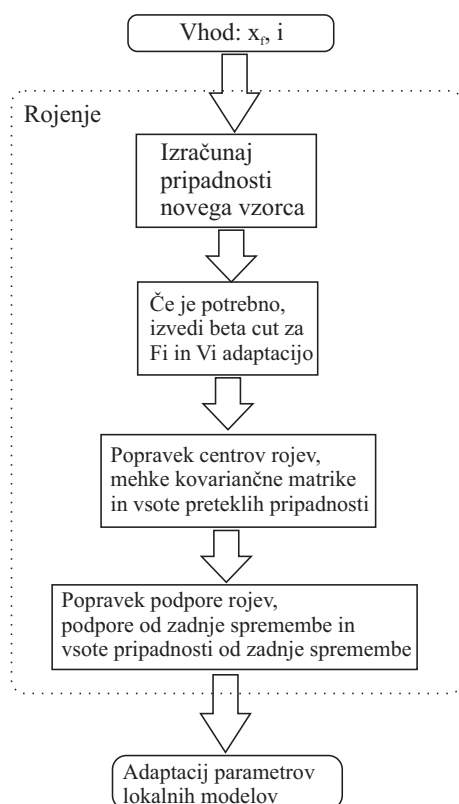
Sledi klic mehanizma nadzorovanega združevanja (blok 13). Mehanizem preveri izbrane pogoje za nadzorovano združevanje rojev (poglavje 6.3, drugi del). Če noben roj ni bil združen, CDL blok nadaljuje s klicem nenadzorovanega združevanja rojev (blok 16), ki preveri pogoj 6.23 in po potrebi združi roje.

Nazadnje sledi klic mehanizma za deljenje rojev. Klic se izvede, če je število obstoječih rojev manjše od dopustnega števila (blok 18). Najprej se osveži napaka predikcije modela (en. 6.36), ki se porazdeli med roje (en. 6.37). Napaka se osveži za tiste vzorce, ki pripadajo trenutni strukturi mehkega modela oziroma za vzorce, ki ne izpolnjujejo pogoja dodajanja (blok 19). Sledi preverjanje roja z maksimalno napako. Če ta presega prag (en. 6.38) se roj deli.

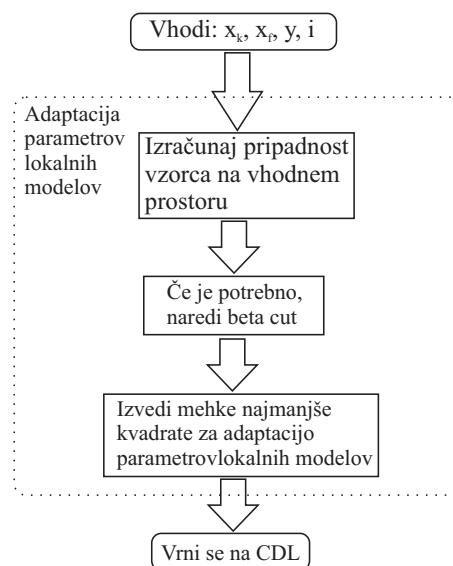
Mehanizem za preverjanje minimalnega pogoja za obstoj roja (en. 6.19) se izvaja neodvisno od časa mirovanja samorazvijajočih mehanizmov. Izvede se, ko najmlajši roj doseže mejno starost (*age_trh*, blok 23).

Po preverjanju pogojev za dodajanje, združevanje, deljenje in odstranjevanje rojev eFuMo algoritem nadaljuje z izračunom območja izhoda procesa (en. 6.34), ki je potreben za izračun normirane napake pri mehanizmu deljenja (blok 26). Sledi klic adaptacijskega mehanizma (blok 27) ter izračun povprečnih vrednosti in varianc vhodov (en. 6.34), ki so potrebni za izvršitev nadzorovanega združevanja rojev (blok 29). Sledi čakanje na naslednji podatkovni vzorec.

Adaptacijski algoritem (blok 27) obsega rekurzivno rojenje in rekurzivno identifikacijo parametrov linearnih lokalnih modelov. Potek rekurzivnega rojenja je prikazan na sliki 6.9, potek identifikacije lokalnih modelov pa na sliki 6.10. Najprej se izvede rekurzivno rojenje (sl. 6.9). Vhoda v rojenje sta vektor rojenja ($\mathbf{x}_f$) in zaporedna številka vzorca (i). Algoritem izračuna pripadnosti oziroma proženja rojev za trenutni vektor rojenja (en. 4.2). Po potrebi preveri pripadnosti in tiste, ki so manjše od določenega pragu postavi na nič. Ostale ponovno normira. Sledi adaptacija centrov rojev (en. 4.4 in en. 4.10), mehke kovariančne matrike (en. 4.21) in njenega inverza (en. 4.26), adaptacija determinante (en.



Slika 6.9: Rekurzivno rojenje.



Slika 6.10: Rekurzivna identifikacija lokalnih modelov.

4.31) in vsote preteklih pripadnosti (en. 4.12). Roju z najvišjim proženjem se poveča podporna množica za en vzorec. Izvajanje algoritma se nadaljuje z rekurzivno identifikacijo parametrov lokalnih linearnih modelov (sl. 6.10).

Vhodi v algoritem rekurzivne identifikacije parametrov lokalnih modelov so: vektor rojenja samo za vhodni prostor ($\mathbf{x}_f$), regresijski vektor ($\mathbf{x}_k$), izhod modeliranega procesa (y) in zaporedna številka vzorca (i). Na podlagi vektorja rojenja se izračunajo vhodne pripadnosti podatkovnega vzorca (en. 4.32 in en. 4.33 ali en. 4.35), ki se nato normirajo (en. 4.34). Pri identifikaciji parametrov lokalnih modelov dobimo boljše rezultate, če ne opravimo *betacut* postopka. Sledi adaptacija parametrov lokalnih modelov po eni izmed metod, predstavljenih v poglavju 4.2.

6.5.1 Primerjava eFuMo metode z drugimi

Napovedovanje MG časovne vrste

Predlagana eFuMo metoda daje nekoliko boljše rezultate kot rFCM in rGK identifikacija mehkega modela. Primerjavo eFuMo metode smo naredili na primeru MG časovne vrste, ki je opisana v poglavju 4.4. Dobljeni rezultati so predstavljeni v tabeli 6.1. Z nastavljanjem različnih faktorjev dobimo različno število rojev. Prva vrstica tabele ($eFuMo_1$), prikazuje rezultate dobljene z privzetimi nastavitvami: začetna vrednost vsote preteklih pripadnosti ($s_i(0)$) je bila 1, prag za *betacut* postopek 0,2; število zaporednih vzorcev, ki morajo izpolnjevati pogoj dodajanja na razdaljo 5, začetna kovariančna matrika najmanjših kvadratov je bila inicializirana kot $\mathbf{P}_i = 10\mathbf{I}$, izbrana metoda za identifikacijo lokalnih modelov je bila podana z enačbo 4.38, pripadnosti za identifikacijo parametrov lokalnih modelov so bile računane z enačbami 4.32, 4.33 in 4.34; za rojenje je bila uporabljena Mahalanobisova razdalja (en. 3.6 in 3.7), faktorji pozabljanja so bili nastavljeni na 1, pri mehanizmu deljenja je bila maksimalna dopustna napaka 0,5, minimalna 0,05 in konstanta upadanja 1000, faktor normiranja razdalje pri pogoju dodajanja (k_n) je bil nastavljen na 2. V ostalih primerih ($eFuMo_2$, $eFuMo_3$, $eFuMo_4$, $eFuMo_5$) je bila začetna vrednost $s_i(0)$ nastavljena na 100, začetna kovariančna matrika najmanjših kvadratov kot $\mathbf{P}_i = 1000\mathbf{I}$, število zaporednih vzorcev na 1, za identifikacijo lokalnih modelov je bila uporabljena metoda pod

eFuMo	Pravila/roji	NDEI
$eFuMo_1$	2	0,4911
$eFuMo_2$	41	0,1386
$eFuMo_3$	85	0,1098
$eFuMo_4$	88	0,1003
$eFuMo_5$	88	0,0981

Tabela 6.1: Rezultati predikcije MG vrste z eFuMo metodo.

enačbo 4.44. Pripadnosti so se računale po enačbah 4.36, 4.35 in 4.34. Faktorji pozabljanja so bili enaki kot pri $eFuMo_1$, razen pri $eFuMo_5$, kjer so bili nastavljeni na vrednost 0,9999. Prag za *betacut* postopek je bil enak kot pri $eFuMo_1$, razen pri $eFuMo_2$, kjer je bil nastavljen na vrednost 0,3. Pri mehanizmu deljenja pri $eFuMo_3$, $eFuMo_4$ in $eFuMo_5$ je bila maksimalna dopustna napaka 0,01 in minimalna 0,001, konstanta upadanja pa 100. Razdalja uporabljena za rojenje je bila pri $eFuMo_2$ enaka kot pri $eFuMo_1$, pri $eFuMo_3$, $eFuMo_4$ in $eFuMo_5$ pa je bila uporabljena razdalja pod enačbo 4.36. Faktor normiranja razdalje pri pogoju dodajanja k_n je bil pri $eFuMo_2$ enak 2, pri $eFuMo_3$ 1,7, pri $eFuMo_4$ in $eFuMo_5$ pa 1,4. V vseh primerih, razen v $eFuMo_1$, je bilo vklopljeno osveževanje parametrov adaptacije. Osveževanje se zgodi, če se spremeni število rojev v strukturi. Takrat se rojem, ki imajo večji s_i (vsota preteklih pripadnosti) od 300, le-ta zmanjša za 10 procentov. Vsem lokalnim modelom se osveži kovariančna matrika najmanjših kvadratov kot je opisano v poglavju 6.1.5. Nastavitve so pregledneje predstavljene v tabeli 6.2.

Rezultati kažejo, da se z uporabo samorazvijajočih mehanizmov učinkovitost metode izboljša. Z eFuMo metodo smo lahko z manjšim številom rojev bolje aproksimirali MG vrsto, kot z rFCM in rGK metodo.

Predikcija nelinearnega procesa

Pri primerjavi metod učenja mehkih modelov se v literaturi pojavi tudi primer predikcije nelinearnega sistema [163], [164], [150]. eFuMo metodo smo preizkusili na primeru aproksimacije procesa, ki jo podaja enačba 6.40.

$$y(k+1) = \frac{y(k)}{(1+y(k)^2)} + u(k)^3 \quad (6.40)$$

eFuMo	$s_i(0)$	Prag <i>betacut</i>	N_{out}	$\mathbf{P}_i(0)$	LSE	računanje pripadnosti za LSE
$eFuMo_1$	1	0,2	5	10 I	en. 4.38	en. 4.32, 4.33 in 4.34
$eFuMo_2$	100	0,3	1	1000 I	en. 4.44	en. 4.36, 4.35 in 4.34
$eFuMo_3$	100	0,2	1	1000 I	en. 4.44	en. 4.36, 4.35 in 4.34
$eFuMo_4$	100	0,2	1	1000 I	en. 4.44	en. 4.36, 4.35 in 4.34
$eFuMo_5$	100	0,2	1	1000 I	en. 4.44	en. 4.36, 4.35 in 4.34
eFuMo	napaka min/max/T	razdalja za rojenje	pozabljanje	faktor k_n	osveževanje	čas mirovanja
$eFuMo_1$	0,5/0,05/1000	en. 3.6 in 3.7	1	2	NE	5
$eFuMo_2$	0,5/0,05/1000	en. 4.36	1	2	DA	5
$eFuMo_3$	0,01/0,001/100	en. 4.36	1	1,7	DA	5
$eFuMo_4$	0,01/0,001/100	en. 4.36	1	1,4	DA	5
$eFuMo_5$	0,01/0,001/100	en. 4.36	0,9999	1,4	DA	5

Tabela 6.2: Glavne nastavitve eFuMo.

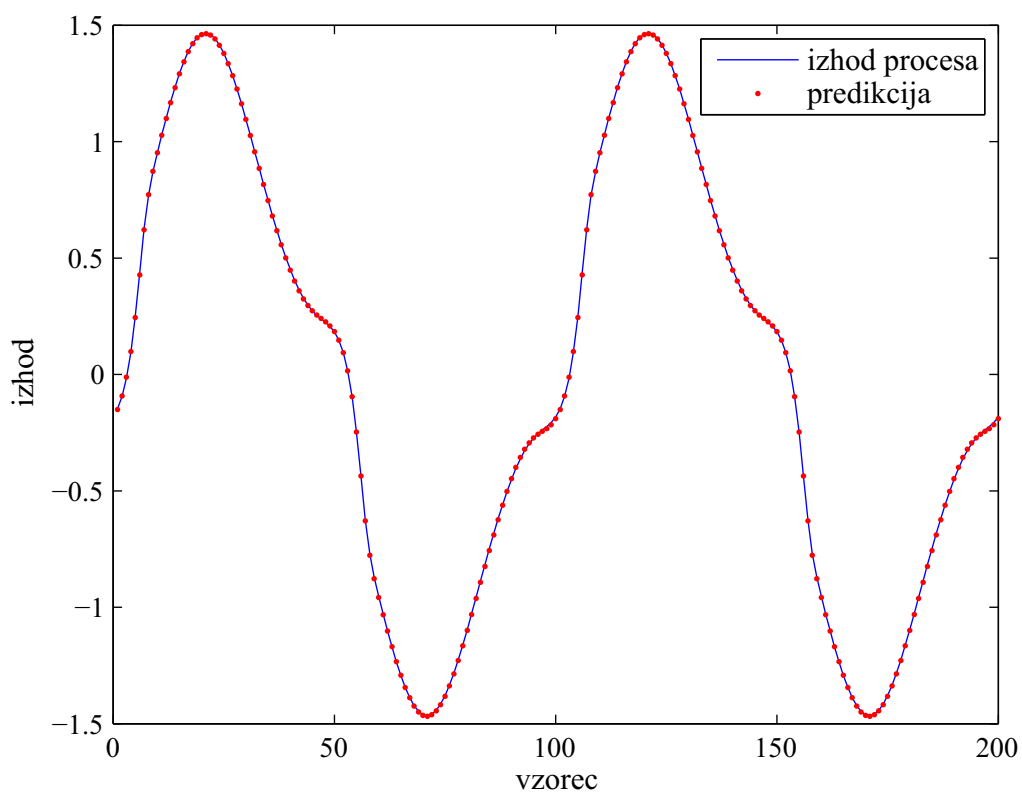
Vhod v proces je definiran kot $u(k) = \sin(2\pi \frac{k}{100})$. Učna množica je vsebovala 50000 vzorcev, validacijska pa 200. Začetni pogoj za izhod procesa je bil $y(0) = 0$ in za vhod v proces $u(0) = 0$. Kot predlagano v [150], je bil uporabljen regresor $\mathbf{x}_k = [u(k-1) \ y(k-1)]$ in vektor rojenja $\mathbf{x}_f = [u(k-1) \ y(k-1) \ y(k)]$.

Rezultati za predikcijo izhoda dinamičnega nelinearnega sistema so podani v tabeli 6.3. Rezultati primerjanih metod so povzeti po člankih [150] in [164]. Iz tabele je razvidno, da s predlagano metodo dosegamo primerljive rezultate že uveljavljenih metod. Na sliki 6.11 vidimo primerjavo predikcije izhoda in dejanskega izhoda za primer $eFuMo_3$. Predikcija se zelo dobro ujema z dejanskim izhodom. Napaka med izhodom procesa in njegovo predikcijo je prikazana na sliki 6.12. Pripadnostne funkcije in območja rojev so prikazane na slikah 6.13 in 6.14.

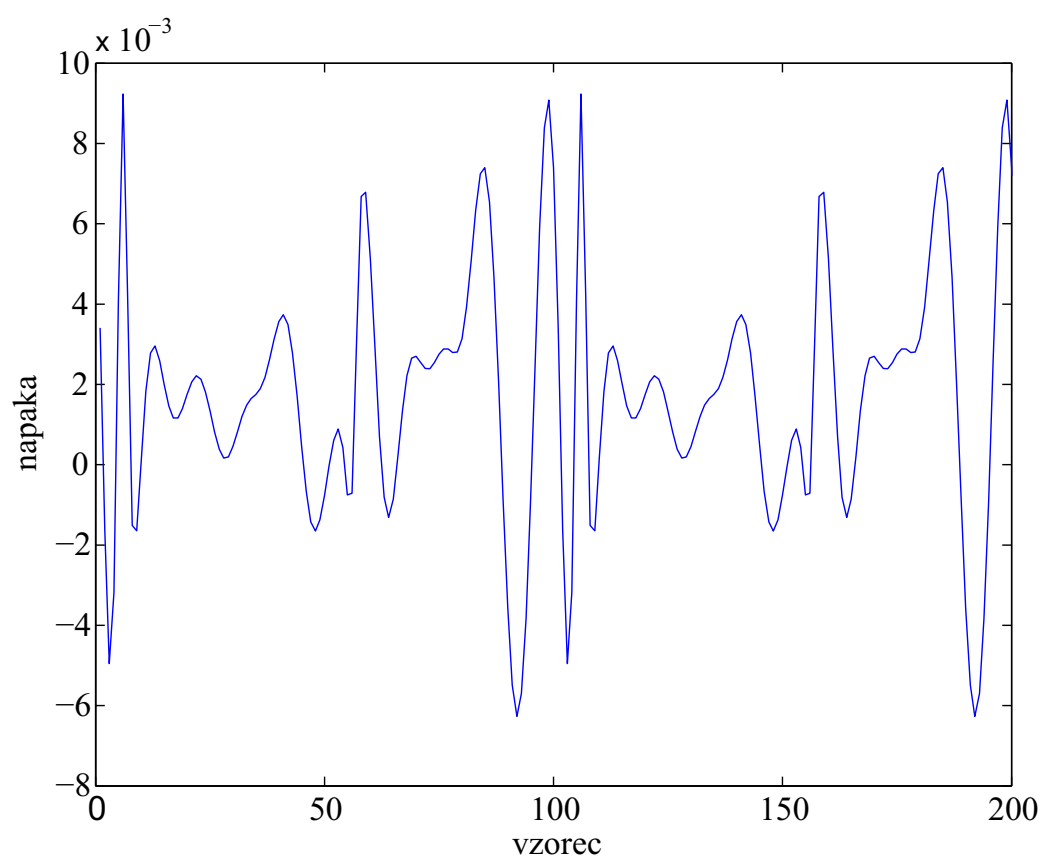
Rezultati so precej odvisni od tega nastavitve metode. Pri različnih nastavitvah dobimo različne natančnosti modelov. Predlagana eFuMo metoda nudi uporabniku veliko fleksibilnosti pri izbiri pogojev in parametrov. Tako lahko dosežemo boljše natančnosti modelov, vendar je lahko iskanje najboljših parametrov dolgotrajen proces. Pomembnejše nastavitve v katerih se razlikujejo primeri $eFuMo_1$, $eFuMo_2$ in $eFuMo_3$ so podani v tabeli 6.4.

Metoda	Pravila/roji	RMSE
SAFIS	8	0,0116
MRAN [87]	10	0,0129
RANEKF [125]	11	0,0184
simpl.eTS	18	0,0122
eTS	19	0,0082
SONFIN	10	0,013
SAFIN	6	0,017
SAFIN	8	0,011
SAFIN	13	0,007
SAFIN	14	0,006
<i>eFuMo</i> ₁	2	0,0522
<i>eFuMo</i> ₂	7	0,0194
<i>eFuMo</i> ₃	12	0,0035

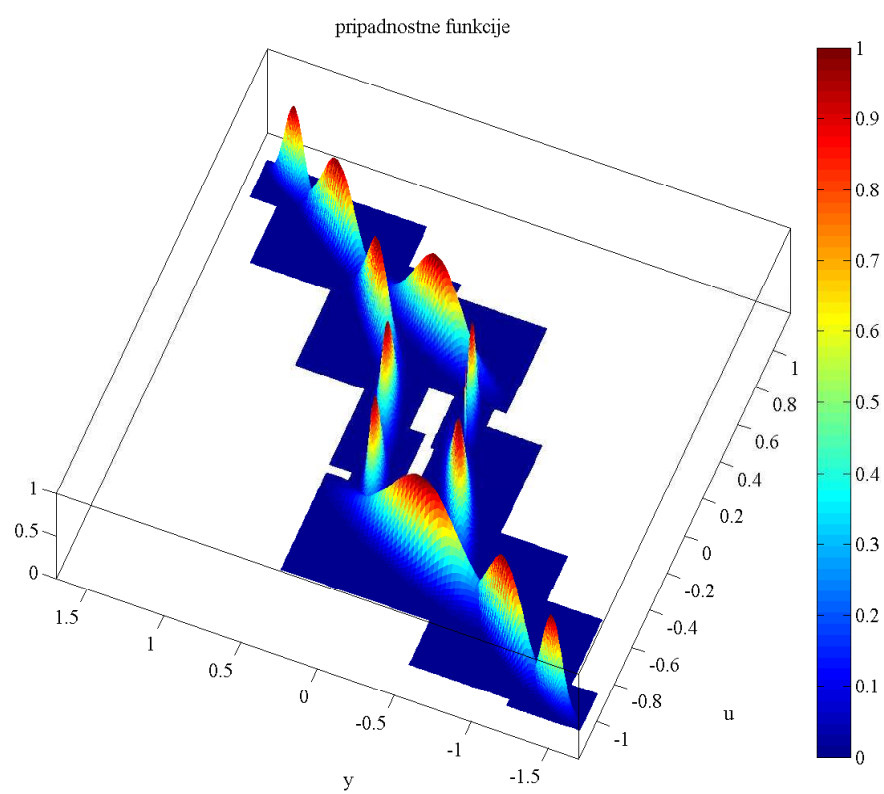
Tabela 6.3: Primerjava rezultatov predikcije dinamičnega nelinearnega sistema.



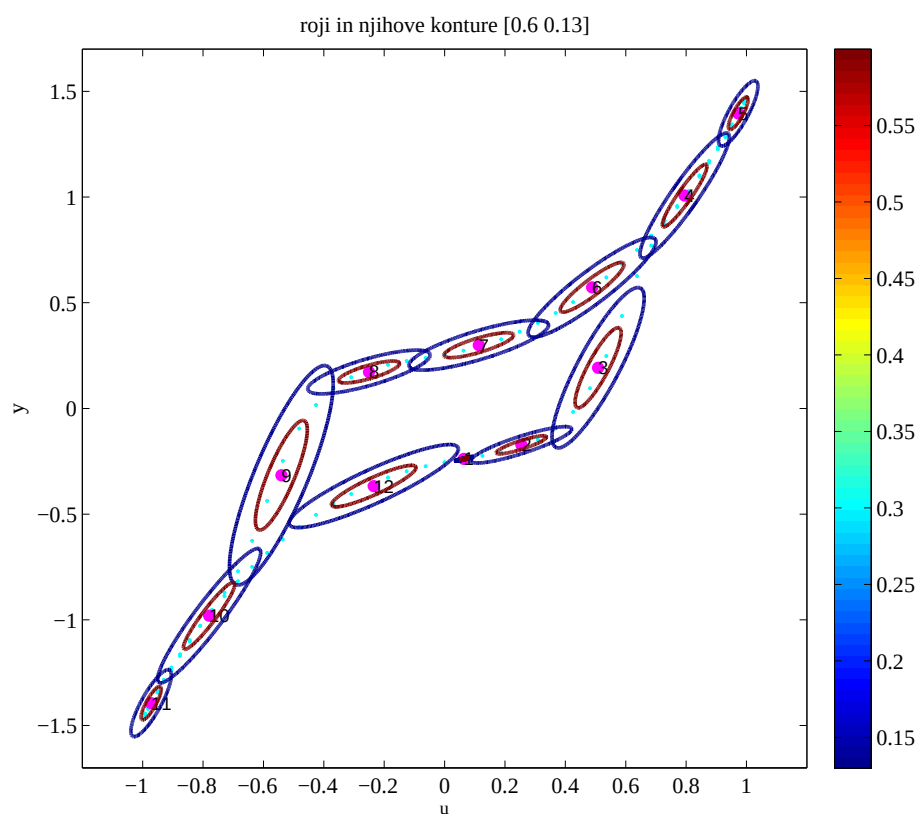
Slika 6.11: Primerjava med predikcijo in dejanskim izhodom procesa.



Slika 6.12: Napaka med realnim izhodom in predikcijo.



Slika 6.13: Pripadnostne funkcije rojev.



Slika 6.14: Roji in njihova območja.

eFuMo	$s_i(0)$	Prag <i>betacut</i>	N_{out}	$\mathbf{P}_i(0)$	LSE	računanje pri- padnosti za LSE
$eFuMo_1$	1	0,2	5	10 I	en. 4.38	en. 4.32, 4.33 in 4.34
$eFuMo_2$	100	0,3	1	1000 I	en. 4.44	en. 4.36, 4.35 in 4.34
$eFuMo_3$	100	0,2	1	10000 I	en. 4.44	en. 3.6 in 3.7, 4.35 in 4.34
eFuMo	napaka min/max/T	razdalja za rojenje	pozabljanje	faktor k_n	osveževanje	čas mirovanja
$eFuMo_1$	0,5/0,05/1000	en. 3.6 in 3.7	1	2	NE	2
$eFuMo_2$	0,05/0,01/100	en. 3.6 in 3.7	0,9999	1,7	DA	20
$eFuMo_3$	0,05/0,01/100	en. 4.36	0,9999	2	DA	2

Tabela 6.4: Glavne nastavitve eFuMo za primer identifikacije nelinearnega dinamičnega sistema.

7. Zaznavanje napak na primeru čistilne naprave

Pri povečanih zahtevah za učinkovitost proizvodnje in kvalitete izdelkov igra pomembno vlogo zaznavanje napak in spremljanje procesa. Napake lahko zaznavamo s pomočjo modela procesa, statističnih metod ali umetne inteligence [79].

Učinkovit sistem mora napako zaznati (*fault detection*), najti mesto in vzrok napake (*fault diagnosis/fault isolation*) ter ustrezno ukrepati, da zagotovi nemoteno delovanje procesa (*fault management*) [79].

Napake lahko delimo v tri skupine: nenadne napake (*abrupt faults*), ki imajo obliko stopnice, počasne napake v obliki lezenja (*incipient fault*) in alternirajoče napake (*intermittent fault*) v obliki naključnih impulzov.

Metode za zaznavanje napak večinoma temeljijo na modelu procesa (glede na izhodno napako [68, 135] ali glede na identificirane parametre [57]), na observatorju stanj, na modelu signalov ali na analizi glavnih komponent [109]. Zelo pogosta je tudi uporaba preverjanja limitnih vrednosti signala (*limit checking*). Da bi zmanjšali število lažnih alarmov, se večina teh metod uporablja v povezavi z drugimi statističnimi pristopi [14, 79], kot so test hipotez [158, 173], t-test [192], Run-Sum in CUSUM test [148, 94], F-test ali podobnostni test (*likely-hood test*) [136].

Mehki (*soft* ali *software*) senzorji so nekakšen diagnostični sistem za nadzor senzorjev, ki zazna napako na senzorju in v času njegovega izpada oceni vrednost izhoda [60]. Mehki senzorji se uporabljajo tudi za ocenjevanje nemerljivih spremenjivk v procesu, ki jih ocenjujejo na podlagi drugih signalov, ki so tako ali drugače povezani z nemerjenim signalom [152, 43].

Eden prvih primerov uporabe mehkega senzorja je predstavljen v [160], kjer so uporabljeni linearni modeli za spremljanje procesov v rafineriji. Primer uporabe

statičnih fizikalnih odvisnosti med veličinami je uporabljen v [56] za spremljanje degradacije fenola in spremljanje optične gostote v aerobnih in anaerobnih bioreaktorjih. V [43] je uporabljen dinamični matematični model za spremljanje kristalizacije sladkorja. Model je zgrajen na podlagi fizikalnih in kemičnih relacij med merjenimi in ocenjevanimi spremenljivkami. Z njim ocenjujejo lastnosti materiala, ki sproti niso merljive.

Primer mehkega senzorja na podlagi statističnih metod najdemo v [183], kjer je za gradnjo modela uporabljeno SVM učenje (Suport Vector Machine), ki temelji na SLT teoriji (Statistical Learning Theory). Za izbiro modelov je uporabljeno Bayesovo odločitveno pravilo. Senzor je zgrajen za oceno temperature, pri kateri zamrzne dizelsko gorivo. Enak problem je obravnavan v [123] s to razliko, da je tu uporabljen mehki senzor, zgrajen z uporabo neuro-mehkega pristopa.

V [59] je predlagan mehki senzor na podlagi nelinearnega dinamičnega modela, ki je zgrajen z nevronske mreže. Z njim je izboljšana regulacija procesa debutanizacije. Mehki senzor je uporabljen za oceno koncentracije bencina in butana. Predikcijo izhoda računajo za tri korake. S senzorjem so premagali veliko časovno zakasnitev meritev, ki jo uvaža kromatograf.

Statični model iz nevronske mreže za oceno NO_x koncentracije v dimnih plinih je uporabljen v [70], kjer je pokazana tudi prednost pred mehkim senzorjem z linearnim modelom. V [55] je obravnavan problem zaznavanja napačnih vrednosti izhoda senzorja (outliers) v procesu litografskega tiskanja. Mehki senzor je bil zgrajen na podlagi mehke logike in nevronske mreže. Nevronska mreža je uporabljena za model procesa, medtem ko je ekspertni mehki model uporabljen za zaznavanje napačnih vrednosti senzorjev. Z izločitvijo teh vrednosti se zmanjša možnost, da se model uči na napačnih podatkih.

Metoda eTS je uporabljena v [13] za izgradnjo mehkega adaptivnega senzorja, ki je sposoben samokalibracije. Na podlagi starosti centrov mehki senzor zaznava lezenje (*shift*) na izhodu senzorja.

V [190] je predstavljen sistem za spremljanje obrabe ležajev. Na podlagi Bayesovega estimatorja (poznanega tudi pod imenom *particle filtering*) in modela obrabe ocenjujejo prehitro obrabo ležaja.

Eden izmed ciljev doktorske disertacije je uporaba razvite rekurzivne

Gustafson-Kessel metode za izgradnjo adaptivnega mehkega senzorja. Zgrajeni senzor bo preizkušen na procesu čiščenja odpadnih voda, kjer pogosto prihaja do izpadov fizičnih senzorjev [186]. Proces čiščenja voda je nelinearen proces [157] s časovno spremenljivimi parametri [186]. Zato je za gradnjo mehkega senzorja smiselno izbrati adaptivni mehki model.

V procesu čiščenja odpadnih voda se mehki senzorji večinoma uporabljajo za ocenjevanje spremenljivk, ki niso merjene sprotno. V [71] je predstavljen mehki senzor, zgrajen za ocenjevanje vsebnosti fosforja. V članku je narejena primerjava med statičnim in FIR modelom ter med različnimi metodami identificiranja parametrov (PLS - *Partial Least Squares*, MLR - *Multiple Linear Regression* in PCR - *Principal Component Regression*). V [25, 186, 185] se ukvarjajo z estimacijo respiracije (dihanja) mikroorganizmov in oceno funkcije vnosa kisika. Ti dve spremenljivki ključno vplivata na kvaliteto regulacije raztopljenega kisika. Ocenjujeta se iz meritev raztopljenega kisika in pretoka zraka. V [25] za estimacijo stanja respiracije in identifikacijo parametrov funkcije vnosa kisika predlagajo nelinearni algoritem, ki temelji na Lunembergerjevem algoritmu. Predlagani algoritem omogoča hkratno estimacijo stanj in identifikacijo parametrov v nelinearnih procesih. Za oceno parametrov vnosa kisika v [185] predlagajo LS-SVM (*Least Squares Support Vector Machine*) algoritem, v [186] uporabo GDLS (*Generalized Damped Least Squares*) algoritma, v [76] in [112] pa Kalmanov filter.

V člankih [98, 27, 33] je obravnavan problem on-line estimacije amoniaka in dušikovega oksida. Za oceno teh vrednosti so uporabljene sprotne meritve za raztopljeni kisik, pH in oksidacijski redox potencial. V [98] je predlagan model zgrajen na polinomskih nevronskih mrežah z GMDH (*Group Method of Data Gathering*) metodo. Za oceno parametrov mreže uporabljajo LSE metodo. Predlagana je tudi uporaba kompenzatorja, ki temelji na izkustvenih pravilih. Z uporabo kompenzatorja povečajo točnost ocene. Enak problem rešujejo v [27] z uporabo observatorja oziroma Kalmanovega filtra.

V [33] je predlagana uporaba hibridnih nevronskih mrež (nevronska mreža v kombinaciji s PCA) za oceno koncentracije celotnega dušika (Kjeldahlov dušik) v sistemu. Mehki senzor zgradijo z večimi pristopi in jih med seboj primerjajo. Uporabijo: multivariabilno regresijo (*Multivariate Regression*), regresijo z glavnimi komponentami (*Principal Component Regression*), nevronske mreže in

predlagane hibridne nevronske mreže.

Celotni sistem za nadzor čistilnih naprav je predstavljen v [105]. Mehki senzorji so uporabljani za nadzor sprotnih senzorjev in za oceno nemerjenih spremenljivk. Sproti je merjena temperatura, pretok, redox potencial, pH, prevodnost in motnost. Preko relacij ocenjujejo vsebnost fosforja, dušika, potrebo po kisiku (COD - *Chemical Oxygen Demand* in BOD - *Biochemical Oxygen Demand*), koncentracijo fosfatov in nekatere druge vrednosti, ki so pomembne za kvalitetno regulacijo čistilne naprave. Modeli za ocenjevanje teh vrednosti so zgrajeni z nevronskimi mrežami z ARX strukturo lokalnih modelov.

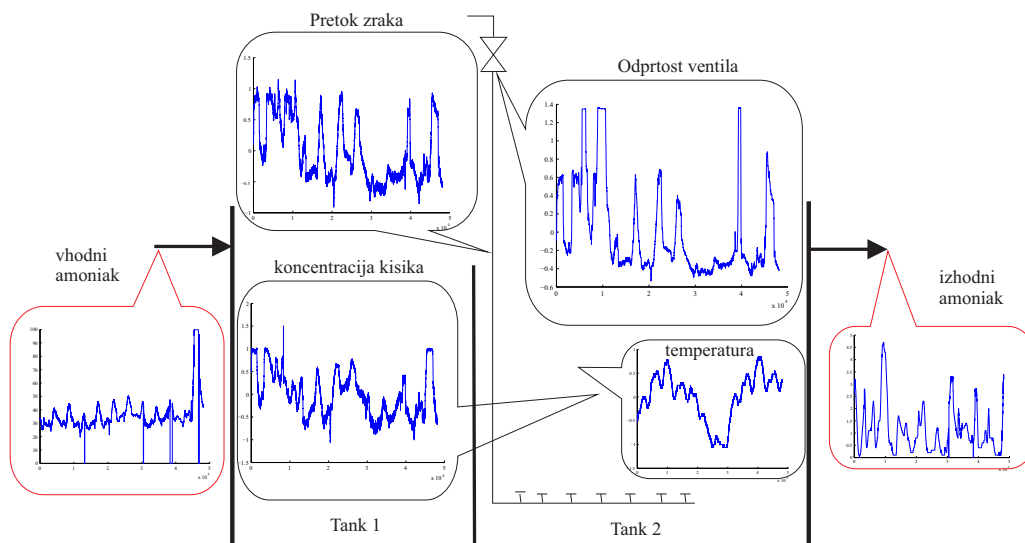
Spremljanje sprotnih senzorjev temelji na adaptivnem PCA modelu. Adaptivnost PCA modela je bila dosežena s pristopom drsečega okna. Za vsak senzor se vodi mera za veljavnost senzorja (*Sensor Validity Index*). Ko ta preseže določeno vrednost nadzorni sistem javi napako. V primeru napake se izhod senzorja oceni na podlagi modela senzorja (mehkega senzorja).

7.1 Opis procesa in problema

Z razvito samorazvijajočo metodo identificiranja mehkega modela (eFuMo) smo poskušali zgraditi sistem za odkrivanje napak na procesu čiščenja odpadnih voda. Ker so parametri tega procesa sezonsko pogojeni, je smiselna uporaba mehkega modela, ki se je sposoben prilagajati trenutnim razmeram. S takim modelom lahko zmanjšamo število napačnih alarmov. Sam koncept in izvedba sistema je v času pisanja te disertacije še v razvoju, zato so tu predstavljeni le rezultati začetne študije. Razvoj sistema in njegovo testiranje je bilo izvedeno na realnih podatkih pridobljenih iz čistilne naprave.

Shema procesa je predstavljena na sliki 7.1. Vhod v proces je koncentracija amoniaka v odpadni vodi, ki pride v čistilno napravo. Izhod procesa je koncentracija amoniaka v prečiščeni vodi. Na izhodno koncentracijo amoniaka vplivamo s koncentracijo kisika v drugem tanku, ki jo nastavljamo s pomočjo pretoka zraka. Le-ta je odvisen od odprtosti ventila. Podrobnejši opis procesa najdemo v [52] in [166].

Med delovanjem procesa se pogostokrat zgodi, da eden izmed senzorjev izpade.



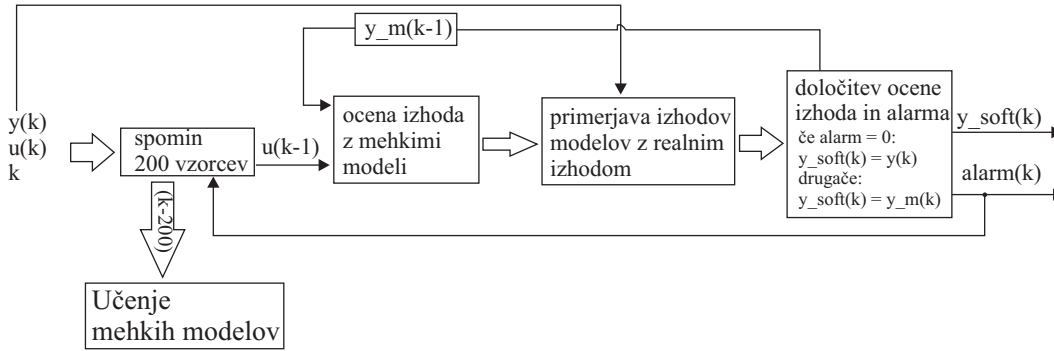
Slika 7.1: Shema čistilne naprave.

Zato želimo zgraditi mehki senzor, ki bo v primeru izpada senzorja sposoben opozoriti operaterja na izpad in nadomestiti izhod senzorja z ocenjenim izходом. Zato je potrebno zgraditi modele, ki opisujejo odnose med spremenljivkami v procesu. Izkaže se, da lahko sistem spremljanja procesa ločimo na dva dela. Prvi del nadzoruje spremenljivke procesa, povezane s koncentracijo kisika, drugi del pa nadzoruje amoniak.

7.2 Spremljanje koncentracije kisika

Za izgradnjo sistema za spremljanje koncentracije kisika v drugem tanku potrebujemo dva mehka modela. Prvi mehki model opisuje odnos med odprtostjo ventila in pretokom zraka, kjer je vhod v model odprtost ventila, izhod pa pretok zraka. Drugi mehki model opisuje odnos med pretokom zraka in koncentracijo kisika, kjer sta vhoda v model pretok zraka in temperatura, izhod pa koncentracija kisika. V obeh primerih smo izbrali mehki model z ARX lokalnimi modeli prvega reda. Glede na rezultate, ki smo jih dobili z metodami za izbiro relevantnih vhodov [130], so bili to najbolj primerni vhodi.

Sistem za zaznavanje napak lahko predstavimo s shemo na sliki 7.2. Naslednji



Slika 7.2: Shema sistema za zaznavanje napak in mehkega senzorja.

opis velja za sistem spremljanja relacije med odprostjo ventila in pretokom zraka. Vhodi v sistem za zaznavanje napak so: številka zaporednega vzorca, vhod spremljanega procesa in izhod spremljanega procesa. Vrednosti se najprej shranijo v vmesni pomnilnik, ki je v našem primeru sposoben pomniti 200 vzorcev. Sledi računanje izhoda modela. Izračunajo se ocenjeni izhodi procesa ter njihove standardne deviacije (en. 7.1). Izhode procesa računamo na podlagi treh modelov, ki opisujejo isto relacijo.

$$\sigma_{y_m} = \sum_{i=1}^c \beta_i \sqrt{\mathbf{F}_{i_{z+1,z+1}}} \quad (7.1)$$

Prvi model je popolnoma samorazvijajoč, drugi je adaptivni in tretji je statični v smislu, da se parametri ne prilagajajo procesu. Vse tri aproksimacije (y_{m1} , y_{m2} , y_{m3}) se primerjajo z dejanskim izhodom procesa (y). Izračuna se absolutna napaka za vsak model:

$$e_{abs_i} = |y - y_{m_i}|, \quad i = 1, 2, 3. \quad (7.2)$$

Sledi računanje dopustne napake, na podlagi katere se določi alarm:

$$e_{max_i} = 0,9M(k-1) + 0,1\sigma_{y_{m_i}} + e_{m_i} \quad (7.3)$$

kjer je $M(k-1)$ filtrirana dopustna meja (en. 7.4) in e_{m_i} spremljana napaka modela.

$$M(k) = 0,9M(k-1) + 0,1(\sigma_{y_{m_i}} + e_{m_i}) \quad (7.4)$$

Absolutne napake izhodov modelov e_{abs} primerjamo z dopustnimi napakami:

$$e_{abs_i} < 1,4e_{max_i}, \quad i = 1, 2, 3. \quad (7.5)$$

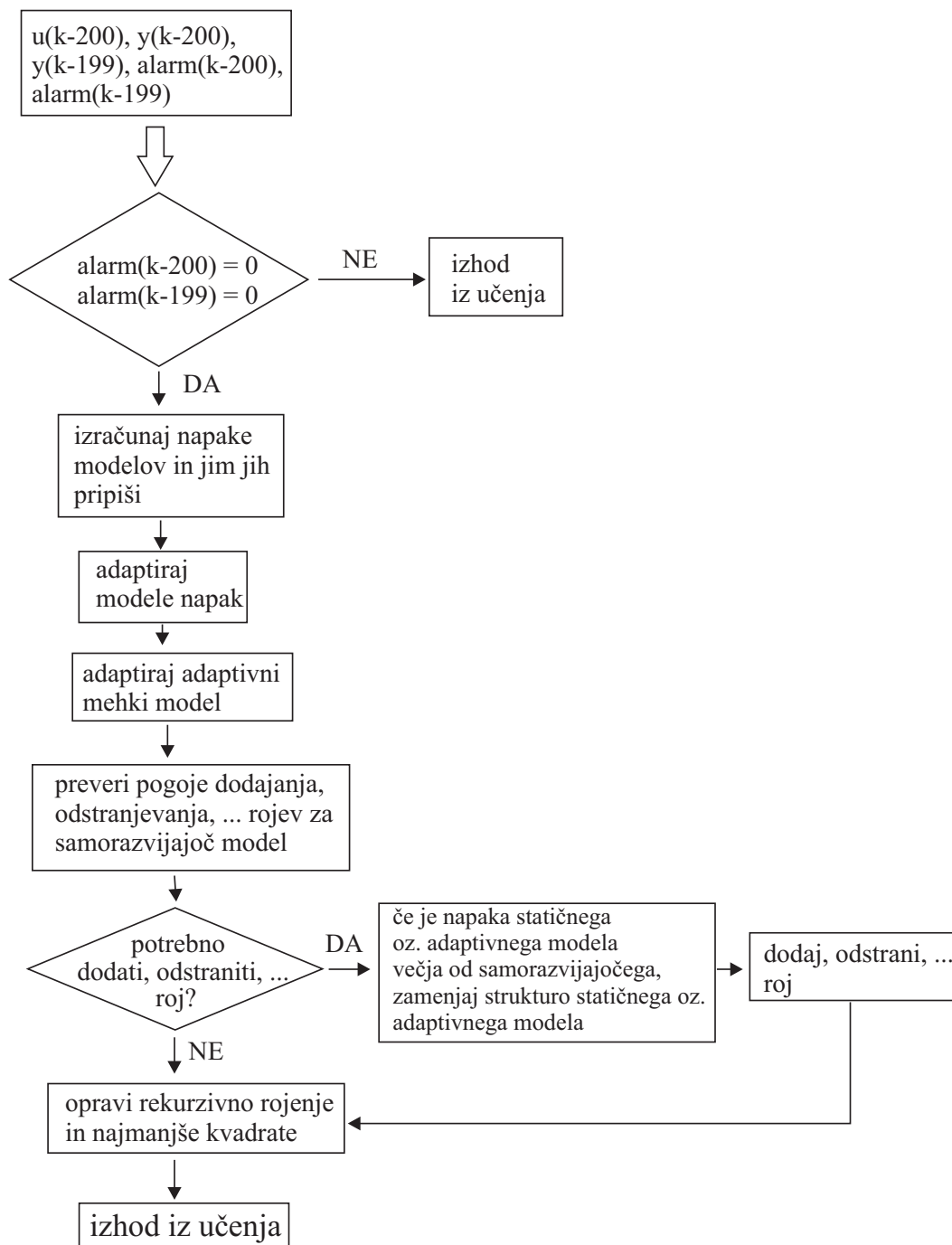
Če zgornji izraz (en. 7.5) velja vsaj za eno od ocen izhoda, potem sistem za zaznavanje napak indikator alarma postavi na nič in kot izhod mehkega senzorja vrne realen signal senzorja. Če izraz velja le za eno od ocen, se ta ocena shrani za potrebe predikcije v naslednjem koraku. Nova meja (en. 7.4) se izračuna na podlagi vrednosti $\sigma_{y_{m_i}}$ in e_{m_i} za to oceno izhoda. Če izraz velja za več ocen se isti postopek opravi za oceno, ki ima najmanjšo napako $0,4e_{abs_i} + 0,6e_{m_i}$.

Glede na to, da imamo večinoma opravka s hitrimi napakami (izpadi senzorjev), smo dodali še dodaten pogoj, ki nam pomaga pri hitrejšem odkrivanju napak. Pogoj spremlja odvod izhodnega signala. Če njegov odvod preseže vrednost 0,22, se prav tako sproži alarm.

Zaznan alarm sproži algoritem za oceno izhoda senzorja. Algoritem poišče model z najmanjšo napako e_m . Če pred zaznanim alarmom ni bilo alarma se izhod modela za $(k - 1)$ vzorec inicializira na vrednost dejanskega izhoda senzorja v spominu. V nasprotnem primeru, se za predikcijo izhoda senzorja uporabi prejšnja vrednost izhoda modela. Nato se na podlagi izbranega mehkega modela izračuna ocenjen izhod senzorja. Mehki senzor vrne ocenjeni izhod senzorja. Ko alarm ugasne, je izhod mehkega senzorja kombinacija med dejanskim izhodom procesa in ocenjenim izhodom. Za prvih 30 vzorcev po alarmu, je izhod mehkega senzorja povprečje med dejanskim in ocenjenim izhodom senzorja. S tem dosežemo gladek prehod na realni signal.

Sistem za zaznavanje napak na signalu koncentracije kisika je zgrajen na enak način. Zaznavanje napake na signalu koncentracije kisika odvisno tudi od napake na signalu pretoka zraka. Zato v času alarma na signalu pretoka zraka, v sistem za detekcijo napake na signalu koncentracije kisika vodimo izhod mehkega senzorja za signal pretoka zraka.

Učenje modelov zaostaja za velikost spomina (200 vzorcev) (sl. 7.3). Zakasnitev je bila dodana, če bi imeli opravka s počasnim lezenjem izhoda, ki bi bil zaznan kot napaka z zakasnitvijo. V tem primeru bi lahko naknadno zaznali točen trenutek začetka napake in postavili alarm za prejšnje vrednosti signalov. Tako bi preprečili učenje modelov na napačnih meritvah. Ta mehanizem trenutno še ni impelmentiran. Učenje modelov poteka z eFuMo metodo. Pri učenju adaptivnega modela so samorazvijajoči mehanizmi onemogočeni. Najprej se za vsak model izračuna absolutna napaka predikcije e_{abs_i} , ki se pripiše mode-



Slika 7.3: Shema učenja mehkih modelov.

lom $e_{m_i}(k) = 0,98e_{m_i}(k-1) + 0,02e_{abs_i}$. Sledi adaptacija adaptivnega modela in učenje samorazvijajočega modela, pri čemer algoritem zaznava spremembe v številu rojev. Preden se število rojev spremeni, algoritem preveri napake vseh treh modelov (e_{m_i}). Če je napaka samorazvijajočega modela manjša od napak adaptivnega in statičnega modela, se strukturi adaptivnega in statičnega modela zamenjata s strukturo samorazvijajočega modela.

Učenje mehkih modelov, ki opisujejo relacijo med odprtostjo ventila in pretokom zraka, se izvede na vzorcih, kjer ni bilo zaznane napake na signalu pretoka zraka. Učenje mehkih modelov, ki opisujejo relacijo med pretokom zraka in koncentracijo kisika, se izvede na vzorcih, kjer ni bilo zaznane napake na signalu pretoka zraka ter signalu koncentracije kisika.

Na vsakih 1000 vzorcev se izračuna tudi napaka simulacijskega modela (če so alarmi za vse vzorce shranjeni v spominu nič). Ta se prišteje napaki modela: $e_{m_i}(k) = 0,6e_{m_i}(k-1) + 0,4e_{sim_i}$. Vsak model (samorazvijajoč, adaptivni in statični) ima tudi model napake, ki izboljša oceno izhoda procesa. Modeli napake so Takagi-Sugeno modeli ničtega reda, kjer je vhodno-izhodni prostor definiran z vhodom in izhodom mehkega modela in napako med izhodom modela in dejanskim izhodom.

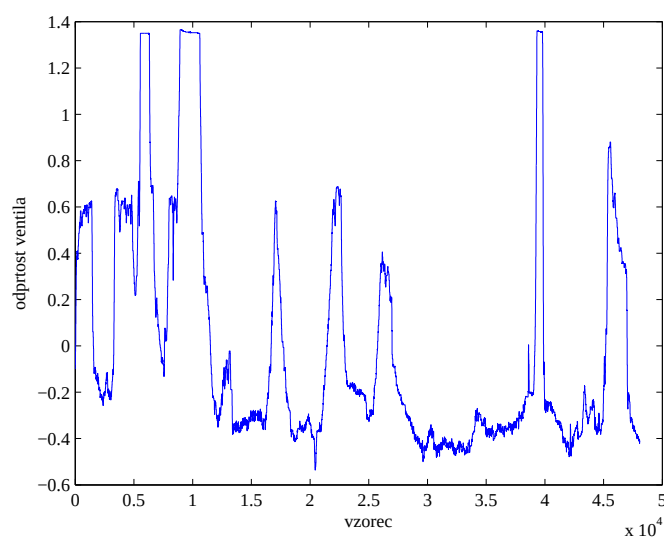
Testiranje sistema smo opravili na realnih signalih. Zaradi boljšega učenja so bili signali normirani z njihovo varianco in povprečno vrednostjo. Simulirani sta bili napaki na dveh odsekih signala. V enem primeru se je napaka najprej pojavila na signalu pretoka zraka in nato na signalu koncentracije kisika, v drugem primeru je bilo obratno. Rezultate podaja tabela 7.1.

Na sliki 7.4 je prikazana odprtost ventila. Izhod procesa in mehkega senzorja za prvi primer je prikazan na sliki 7.5. S črno barvo je prikazan realen signal, z modro realen signal z dodano napako in z rdečo izhod mehkega senzorja. Mesto napake podrobneje prikazuje slika 7.6. Proženje alarma za prvi primer je prikazano na sliki 7.7, kjer zgornji graf prikazuje alarm za signal pretoka zraka, spodnji pa za signal koncentracije kisika.

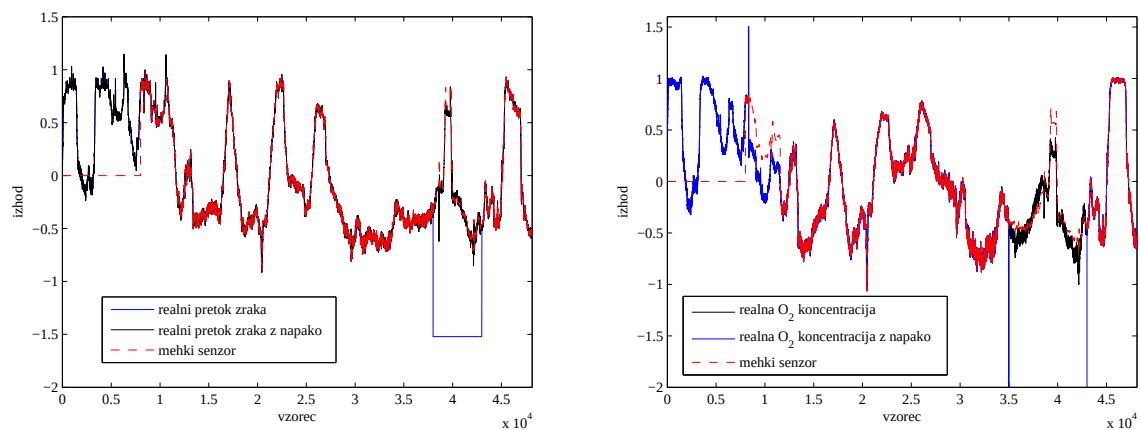
Izhod procesa in mehkega senzorja za drugi primer je prikazan na sliki 7.8. S črno barvo je prikazan realen signal, z modro realen signal z dodano napako in z rdečo izhod mehkega senzorja. Mesto napake podrobneje prikazuje slika

	Primer 1		Primer 2	
napaka	pretok zraka	koncentracija O_2	pretok zraka	koncentracija O_2
NDEI	0,121	0,261	0,050	0,233
min. absolutna	7,126e-6	1,507e-5	3,698e-6	2,141e-5
max. absolutna	0,576	0,673	0,118	0,277
povprečna absolutna	0,031	0,107	0,025	0,093
širina signala	1,69	1,427	1,486	0,918
min. relativna	4,218e-6	1,056e-5	2,488e-6	2,333e-5
max. relativna	0,341	0,472	0,079	0,301
povprečna relativna	0,018	0,075	0,017	0,101
vzorec napake [10^3]	38 – 45	35 – 43	25 – 35	30 – 36

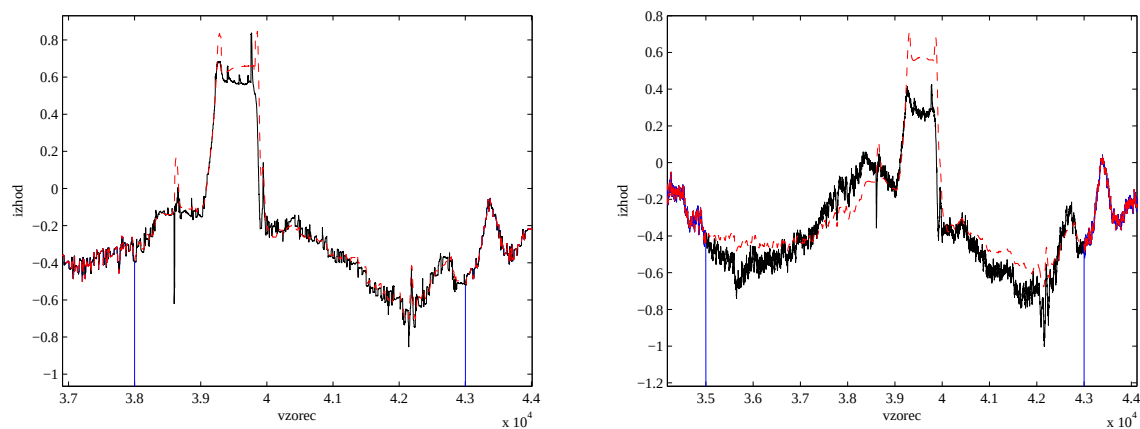
Tabela 7.1: Napaka modela glede na realen signal.



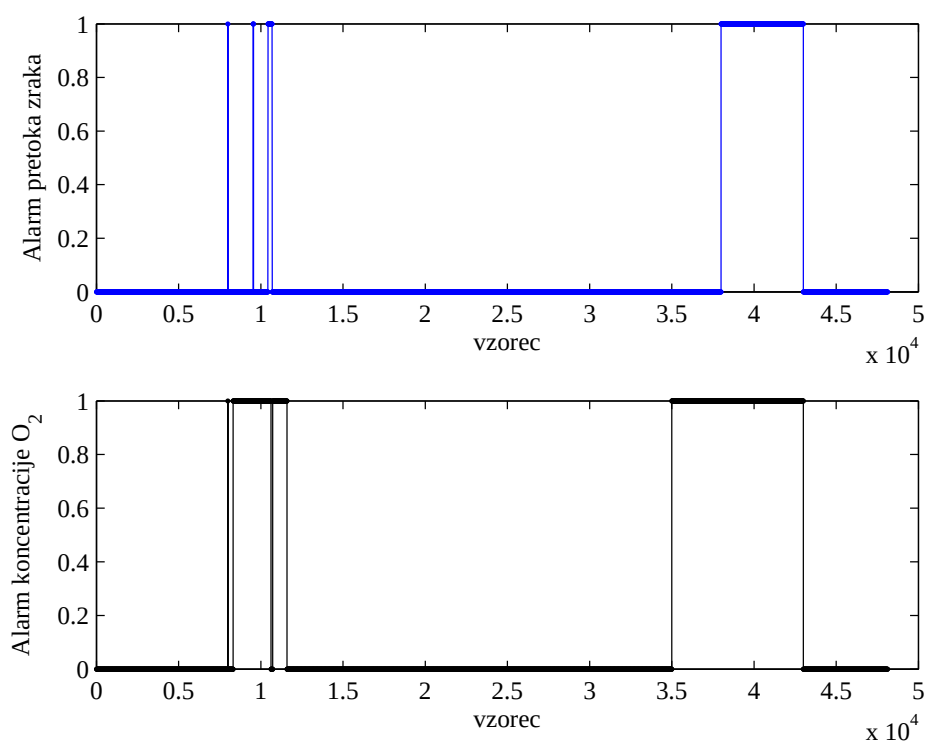
Slika 7.4: Odprtosti ventila.



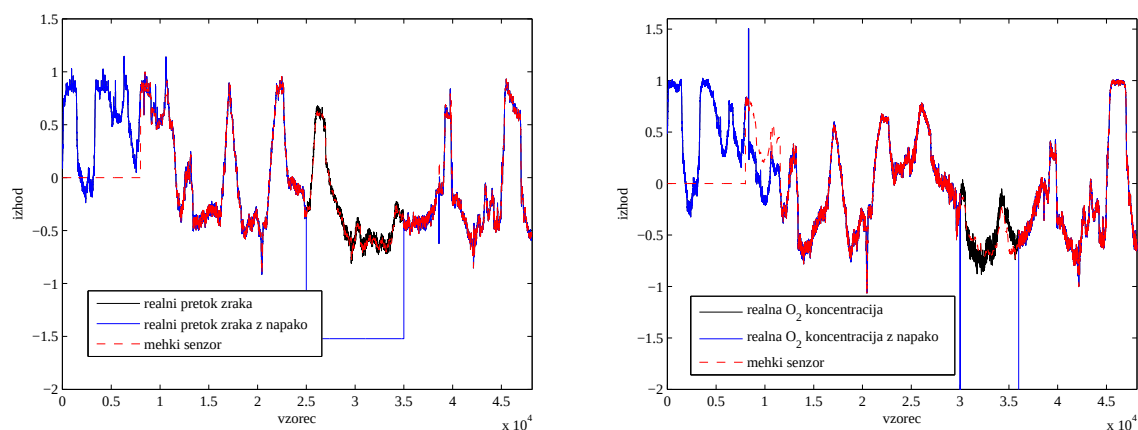
Slika 7.5: Prvi eksperiment: signal pretoka zraka (levo) in signal koncentracije kisika (desno).



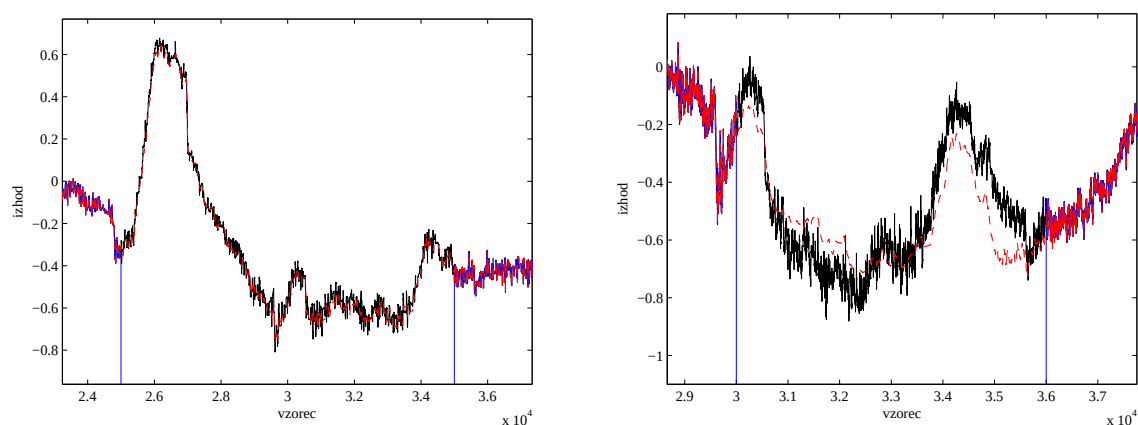
Slika 7.6: Prvi eksperiment: signal pretoka zraka (levo) in signal koncentracije kisika (desno).



Slika 7.7: Prvi eksperiment: alarm signala pretoka zraka (zgoraj) in signala koncentracije kisika (spodaj).



Slika 7.8: Drugi eksperiment: signal pretoka zraka (levo) in signal koncentracije kisika (desno).

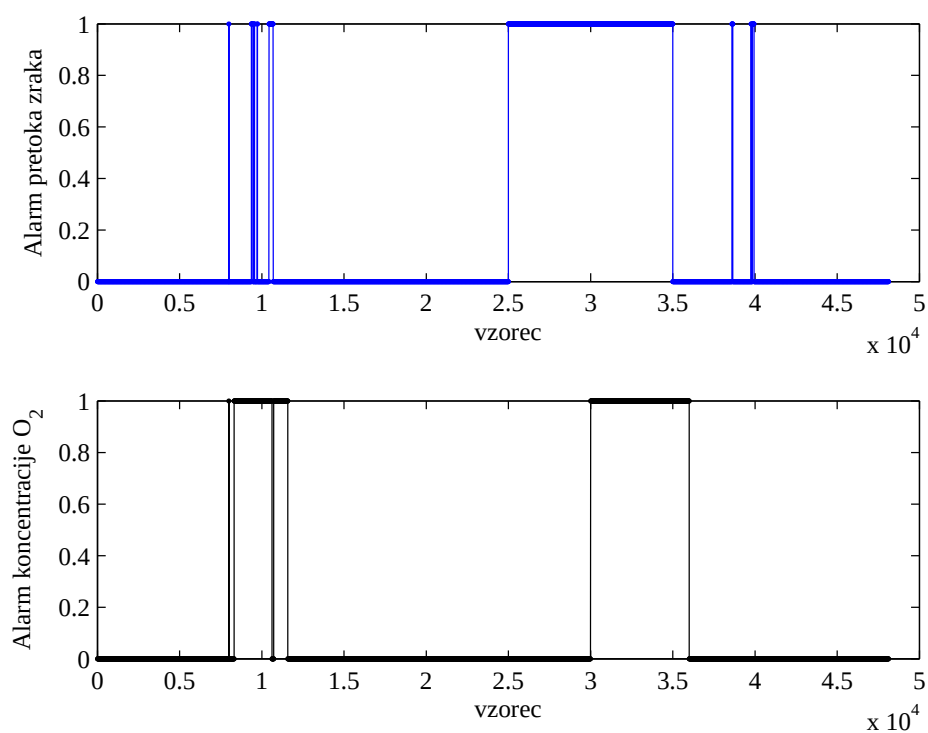


Slika 7.9: Drugi eksperiment: signal pretoka zraka (levo) in signal koncentracije kisika (desno).

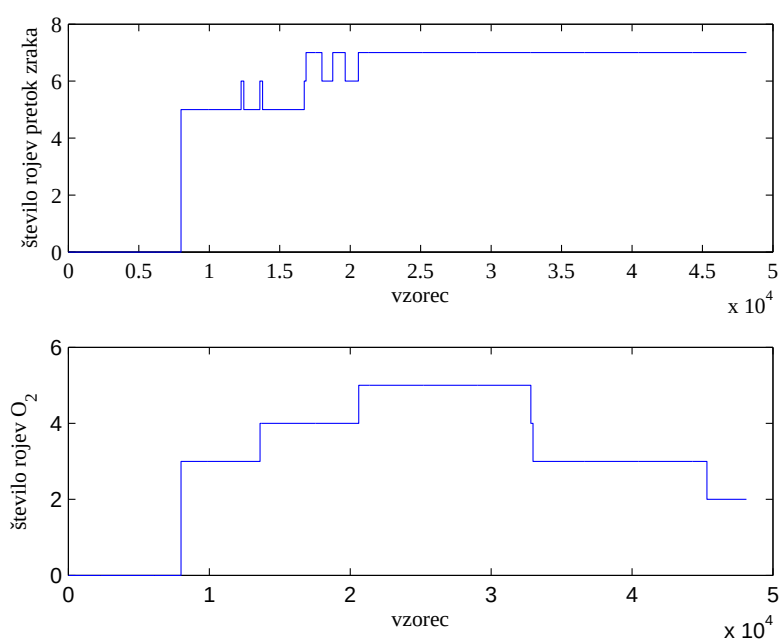
7.9. Proženje alarma za drugi primer je prikazano na sliki 7.10, kjer zgornji graf prikazuje alarm za signal pretoka zraka, spodnji pa za signal koncentracije kisika.

Število rojev samorazvijajočega mehkega modela skozi eksperimentn je za prvi primer prikazano na sliki 7.11, za drugi primer pa na sliki 6.14.

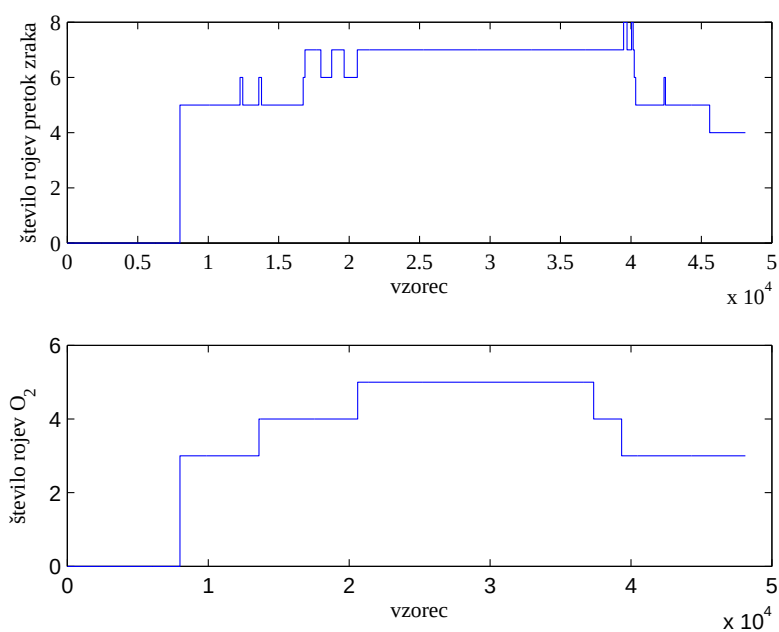
Iz prikazanih rezultatov lahko sklepamo, da je ideja, na kateri sloni predstavljen sistem za zaznavanje napak, uporabna za zaznavanje izpadov realnih senzorjev v procesu čiščenja odpadnih voda (za kisik). Poleg simuliranih napak je algoritem zaznal tudi nekaj napak, ki niso bile namensko dodane. Iz potekov



Slika 7.10: Drugi eksperiment: alarm signala pretoka zraka (zgoraj) in signala koncentracije kisika (spodaj)



Slika 7.11: Prvi eksperiment: Število rojev za mehki model odprtost ventila - pretok zraka (zgoraj) in mehki model pretok zraka - koncentracija kisika (spodaj).

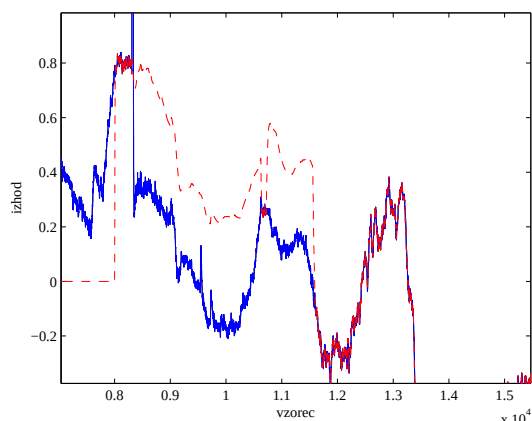


Slika 7.12: Drugi eksperiment: Število rojev za mehki model odprtost ventila - pretok zraka (zgoraj) in mehki model pretok zraka - koncentracija kisika (spodaj).

signalov se zdi, da so napake upravičeno zaznane. Povzročili so jih predvsem nenadni skoki v izhodnih signalih. Čeprav se izhod mehkega senzorja ne prilega popolnoma dejanskemu izhodu procesa v času napake, je vseeno boljše voditi sistem na prikazanem ocenjenem signalu kot na signalu izpadlega senzorja. Pri identifikaciji relacije med odprtostjo ventila in pretokom zraka je bila precejšnja dilema, kaj storiti, ko se ventil hipno popolnoma odpre. Tu se zgodi, da pretok zraka najprej upade in šele nato naraste. V našem primeru tega nismo obravnavali kot napako v procesu, saj je to lastnost dotičnega sistema. Ta pojav vnese dodatno dinamiko v samo relacijo, ki jo z izbranim mehkim modelom ni bilo mogoče dobro opisati. To se kaže pri precejšnjem odstopanju ocenjenega signala od realnega na sliki 7.6, kjer je odstopanje precej večje kot na sliki 7.9.

Prvih 8000 vzorcev je bilo uporabljenih za začetno učenje modela (uporabljena eFuMo metoda), zato v tem času ni izhoda mehkega senzorja (izhod je 0). Preklop na spremljanje procesa je bil narejen ročno, vendar bi z implementacijo algoritmov, ki so sposobni določiti kvaliteto modela, lahko to storili tudi avtomatsko (lahko na podlagi spremljanja varianc izhodov modela, procesa in napake). Naš cilj je bil testirati primernost eFuMo metode za uporabo pri spremljanju procesa čiščenja odpadnih voda, zato so bili implementirani le najnujnejši algoritmi, ki omogočajo delovanje sistema za zaznavanje napak. Za realno uporabo bi bilo potrebno uvesti tudi algoritme, ki poleg zagotavljanja točnosti modela nadzorujejo tudi vzbujanje procesa in po potrebi ustavijo učenje modela. Pred uporabo ideje na realni napravi bi bilo potrebno preizkusiti delovanje na napakah, ki niso takojšnje (lezenja in podobno) ter implementirati logiko za obnašanje sistema, ko se stori samokalibracija senzorjev ...

Na podlagi zbranih podatkov lahko sklepamo, da se je na senzorju za merjenje koncentracije kisika poleg napake zgodila tudi samokalibracija senzorja. Interval prikazuje slika 7.13. Sistem je zaznal napako in pravilno ocenil pričakovan izhod senzorja z mehkim modelom. Če pozorno pogledamo realen signal in izhod mehkega senzorja na sliki 7.13 vidimo, da sta si signala zelo podobna. Signala sta enake oblike, razlikujeta se le za določeno konstantno vrednost. Čeprav je oblika signalov enaka sistem javlja napako, saj je njihova enosmerna komponenta precej različna. Posledično to pomeni, da je tudi napaka velika in v tem primeru očitno večja od dopustne. Vidimo tudi, da okoli vzorca 10600 realen signal pride v dopu-



Slika 7.13: Interval kalibracije senzorja koncentracije kisika (modra - realni signal, rdeča - mehki senzor).

stno območje napak, vendar ga kmalu zapusti (okoli vzorca 10750). Zato se alarm začasno ugasne. Okoli vzorca 11550 očitno odstopanje med realnim signalom in modelom preide v dopustno območje napake za dlje časa. To da modelu možnost, da se adaptira na nove razmere v procesu. Izhod modela se približa izhodu realnega senzorja, kar povzroči, da se alarm ugasne. Lahko se zgodi, da je enosmerna komponenta tako velika, da po samokalibraciji odstopanje med realnim signalom in ocenjenim ne preide več v dopustno območje napake. V tem primeru bi sistem kljub enaki obliki ocenjenega in realnega signala javljal napako. Za rešitev tega problema obstaja več možnosti. V disertaciji predlagamo dva možna načina, ki temeljita na predpostavki, da po samokalibraciji senzorja izravnava njegov signal (odpravi *offset* senzorja). To pomeni, da oblika in varianca signala ostane enaka po samokalibraciji.

Prva rešitev temelji na podlagi varianc signalov. Sklepali smo, da sta varianci realnega in ocenjenega signala, kljub različni enosmerni vrednosti signalov, približno enaki, varianca napake pa manjša od variance obeh signalov (ob predpostavki, da imamo model, ki dobro opisuje proces). Pri drugi predlagani rešitvi pa se naslanjamo na Pearsonov korelacijski koeficijent med ocenjenim in realnim signalom senzorja.

Pri prvi predlagani rešitvi ob nastopu alarma začnemo z rekurzivnim računanjem variance ocenjenega in realnega signala ter variance odstopanja med

njima. Variance računamo po enačbi 6.34. Če je alarm prisoten vsaj 300 vzorcev, začnemo z preverjanjem vrednosti varianc. Algoritem ugasne alarm, ki je bil zaznan na podlagi dopustnega odstopanja realnega in ocenjenega izhoda, ko vrednost variance napake (σ_e) pade pod vrednost variance realnega (σ_y) in ocenjenega izhoda ($\sigma_{\hat{y}}$):

$$\text{Če je } \sigma_e < \sigma_y \text{ in } \sigma_e < \sigma_{\hat{y}}, \text{ potem } alarm = 0. \quad (7.6)$$

Predlagani algoritem za detekcijo samokalibracije senzorja se izključi, če je bil alarm določen na podlagi dopustnega odstopanja realnega in ocenjenega izhoda za 100 zaporednih vzorcev 0, oziroma če je presežen čas njegovega delovanja. Čas njegovega delovanja je bil v našem primeru 700 vzorcev od trenutka, ko je vrednost variance napake padla pod vrednosti variance realnega in ocenjenega izhoda.

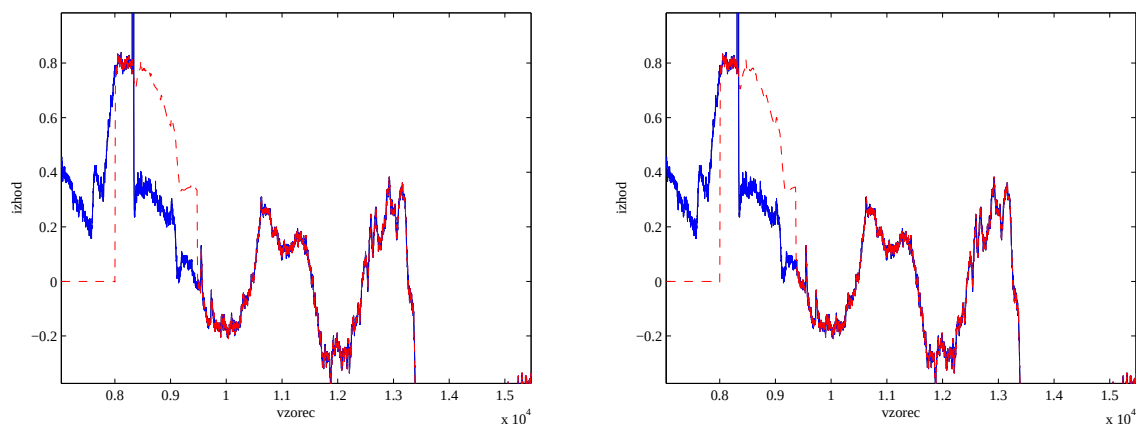
Pri drugi rešitvi pričnemo ob nastopu alarma z računanjem povprečnih vrednosti realnega ($\bar{y}$) in ocenjenega izhoda ($\bar{\hat{y}}$). Povprečna vrednost za oba signala se izračuna rekurzivno po enačbi 6.34 za prvih 500 vzorcev in se nato zamrzne. Ko imamo ocenjeno povprečno vrednost, pričnemo z računanjem produktov, ki so potrebni za določitev korelacijskega koeficienta:

$$\begin{aligned} y_{2\Sigma}(k) &= y_{2\Sigma}(k-1) + (\bar{y} - y(k))^2, \\ \hat{y}_{2\Sigma}(k) &= \hat{y}_{2\Sigma}(k-1) + (\bar{\hat{y}} - \hat{y}(k))^2, \\ y\hat{y}_{2\Sigma}(k) &= y_{2\Sigma}(k-1) + (\bar{y} - y(k))(\bar{\hat{y}} - \hat{y}(k)). \end{aligned} \quad (7.7)$$

Po preteklih 800 vzorcih od nastopa alarma pričnemo s preverjanjem korelacijskega koeficienta:

$$C_{y\hat{y}} = \frac{y\hat{y}_{2\Sigma}(k)}{\sqrt{\hat{y}_{2\Sigma}(k)}\sqrt{y_{2\Sigma}(k)}}. \quad (7.8)$$

Algoritem ugasne alarm, ki je bil zaznan na podlagi dopustnega odstopanja realnega in ocenjenega izhoda, ko vrednost korelacijskega koeficienta preseže določeno pragovno vrednost, v našem primeru 0,9. Predlagani algoritem za detekcijo samokalibracije senzorja se izključi, če je bil alarm določen na podlagi dopustnega odstopanja med realnim in ocenjenim izhodom za 100 zaporednih vzorcev 0, oziroma če je presežen čas njegovega delovanja. Čas njegovega delovanja je bil v našem primeru 700 vzorcev od trenutka, ko je bil na nič popravljen prvi alarm.



Slika 7.14: Mehki senzor z uporabo detekcije samokalibracije na podlagi varianc signalov (levo) in na podlagi korelacijskega koeficienta (desno).

Z obema rešitvama dosežemo, da se alarm zaznan zaradi samokalibracije senzorja ugasne prej. Na sliki 7.14 je prikazan isti odsek signala kot na sliki 7.13. Vendar sta bila v primeru na sliki 7.14 uporabljena algoritma za odpravo alarma pri samokalibraciji senzorja. Če primerjamo rezultate na sliki 7.13 in 7.14 vidimo, da lahko z uporabo predlaganih postopkov odkrijemo trenutek samokalibracije (ob predpostavki, da je naš model točen).

Obe predlagani rešitvi se dobro obneseta. Rešitev na podlagi varianc signalov je bolj praktična, saj uporabniku ni potrebno določiti nobenega pragu, kot mora to storiti pri uporabi rešitve na podlagi korelacijskega koeficienta. Predlagani rešitvi prav tako ne vplivata na zaznavanje napake in izhod mehkega senzorja, ko senzor izpade, saj v tem primeru niso izpolnjeni pogoji za ugasnitev alarma. Velja omeniti, da sta bili rešitvi preizkušeni le za primer samokalibracije senzorja in izpada signala. Kaj se zgodi, če so prisotne napake drugih vrst, še ni bilo testirano!

8. Sklep

V doktorski disertaciji smo predstavili idejo za sprotno identifikacijo mehkega Takagi-Sugeno modela. Ideja je bila razvita na podlagi mehkega c -means rojenja v kombinaciji z mehкими najmanjšimi kvadrati. Rojenje je bilo uporabljeno za sprotno deljenje vhodno-izhodnega prostora, mehki najmanjši kvadrati pa za sprotno identifikacijo parametrov lokalnih linearnih modelov. Rdeča nit disertacije je sprotno deljenje (rojenje) vhodno-izhodnega prostora procesa. Iz razvitega osnovnega rekurzivnega mehkega c -means rojenja smo izpeljali enačbe za rekurzivno Gustafson-Kessel rojenje. Predstavljen je bil primer uporabe za vodenje semi-saržnega reaktorja. Iz rekurzivne metode rojenja smo nadaljevali razvoj v samorazvijajočo metodo identifikacije mehkega modela, ki smo jo poimenovali *evolving Fuzzy Model method - eFuMo*. Njena uporaba je bila predstavljena na primeru sistema za zaznavanje napak na čistilni napravi. Metoda je bila preizkušena tako na realnih kot tudi na simulacijskih primerih. V obeh primerih so bili rezultati zelo pozitivni, njena natančnost pa se lahko kosa tudi z bolj uveljavljenimi metodami. Z nadgradnjo rekurzivne metode v samorazvijajočo se je nabor parametrov metode povečal, vendar pa so se izboljšali tudi rezultati aproksimacije.

Mehke rekurzivne in samorazvijajoče metode so uporabne predvsem pri modeliranju nelinearnih procesov, kjer se parametri med delovanjem zaradi zunanjih vplivov spreminjajo. Prav tako so uporabne pri modeliranju relacij iz velikega nabora podatkov, kjer bi imeli lahko z uporabo *off-line* metod težave s spominom. Na splošno metode za *on-line* identifikacijo mehkih modelov dosegajo malenkost slabše rezultate kot *off-line metode* in imajo več nastavitvenih parametrov, od katerih je odvisna njihova natančnost. Rekurzivne metode potrebujejo tudi večje število vzorcev za učenje modela. Prednost rekurzivnih metod je v sprotni adaptaciji modela, ki lahko doprinese k izboljšanju regulacije pri vodenju oziroma

do manj napačnih alarmov pri sistemih za zaznavanje napak (seveda ob predpostavki, da se taki mehki modeli uporabljajo pri vodenju oziroma sistemu detekcije napak).

9. Doprinos k znanosti

Glavne izvirne prispevke k znanosti lahko povzamemo v naslednjih točkah:

1. **Razvoj rekurzivne c -means metode rojenja**

Na podlagi mehkega c -means algoritma smo razvili njegovo rekurzivno različico, ki jo je možno uporabiti za sprotno rojenje.

2. **Razvoj rekurzivne Gustafson-Kessel metode rojenja.**

Mehko c -means rojenje smo nadgradili v rekurzivno Gustafson-Kessel rojenje. Pri tem smo morali izpeljati rekurzivno obliko računanja mehke kovariančne matrike, njenega inverza in determinante.

3. **Vpeljava adaptivnega mehkega modela v mehki prediktivno funkcijski regulator**

S pomočjo rekurzivnega rojenja in mehkih najmanjših kvadratov smo zgradili adaptivni Takagi-Sugeno mehki model, ki smo ga vpeljali v mehki prediktivno funkcijski regulator. S tem smo dobili adaptivni mehki prediktivno funkcijski regulator, ki se je sposoben prilagajati spremembam parametrov nelinearnega procesa in s tem izboljšati njegovo regulacijo.

4. **Nadgradnja rekurzivne metode mehke identifikacije v samorazvijajočo metodo**

Razviti rekurzivni mehki identifikaciji smo dodali algoritme dodajanja, odzemanja, združevanja in deljenja rojev. S tem smo dobili samorazvijajočo metodo, ki je glede na podatke sposobna prilagajati število rojev.

5. **Uporaba razvite metode v sistemu za zaznavanje napak in mehkem senzorju na primeru procesa čiščenja odpadnih voda**

Z uporabo predlagane samorazvijajoče metode smo zgradili prototip mehkega senzorja in sistema za detekcijo napak za proces čiščenja odpadnih

voda. Sistem je bil zgrajen in preizkušen na realnih podatkih, pridobljenih iz eksperimentalne postavitve procesa. Sama ideja pa je po potrebi lahko razširljiva na drugo čistilno napravo.

Literatura

- [1] Declercq A, Piater J. Online learning of gaussian mixture models—a two-level approach. *Proceedings of the 3rd international conference on computer vision theory and applications VISAPP*, str. 605–611, Funchal, Portugal, 2008.
- [2] Dickerson Julie A, Bart Kosko. Fuzzy function approximation with ellipsoidal rules. *IEEE Transactions on systems, man and cybernetics-Part B: cybernetics*, Vol. 26, No. 4, str. 542–560, 1996.
- [3] Woodbury Max A. Inverting modified matrices. Memorandum Rept 42, Statistical Research Group, Princeton University, Princeton, NJ, 1950.
- [4] P Angelov. An approach for fuzzy rule-base adaptation using on-line clustering. *Integration of Methods and Hybrid Systems*, Vol. 35, No. 3, str. 275–298, 2004.
- [5] P. Angelov, Xiaowei Zhou. Evolving fuzzy systems from data streams in real-time. *Evolving Fuzzy Systems, 2006 International Symposium on*, str. 29 –35, 2006.
- [6] P P Angelov, D P Filev. An approach to on-line identification of Takagi-Sugeno fuzzy models. *IEEE Trans. Syst. Man Cyber. Part B*, Vol. 34, No. 1, str. 484–497, 2004.
- [7] Plamen Angelov. A fuzzy controller with evolving structure. *Information Sciences*, Vol. 161, No. 1-2, str. 21–35, 2004.
- [8] Plamen Angelov. *EVOLVING INTELLIGENT SYSTEMS: Methodology and Applications*, poglavje Evolving Takagi-Sugeno Fuzzy Systems From Streaming Data (eTS+), str. 21 – 50. Wiley, New Jersey, 2010.

-
- [9] Plamen Angelov, Dimitar Filev. *Simpl_eTS: A Simplified Method for Learning Evolving Takagi-Sugeno Fuzzy Models*. *Proceedings of The 2005 IEEE International Conference on Fuzzy Systems*, str. 1068 – 1073, Reno, Nevada, USA, 22-25 May 2005.
 - [10] Plamen Angelov, Dimitar Filev, Nikola Kasabov. Editorial. *Evolving Systems*, Vol. 1, No. 1, str. 1–2, 2010.
 - [11] Plamen Angelov, Dimitar P Filev, Nikola Kasabov. *EVOLVING INTELLIGENT SYSTEMS: Methodology and Applications*. Wiley, New Jersey, 2010.
 - [12] Plamen Angelov, Veniero Giglio, Carlos Guardiola, Edwin Lughofer, Jose Manuel Lujan. An approach to model-based fault detection in industrial measurement systems with application to engine test benches. *Measurement Science and Technology*, Vol. 17, str. 1809–1818, 2006.
 - [13] Plamen Angelov, Arthur Kordon. Adaptive Inferential Sensors Based on Evolving Fuzzy Models. *IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics*, Vol. 40, No. 2, str. 529 – 539, 2010.
 - [14] Đani Juričič, Mina Žele. Robust detection of sensor faults by means of a statistical test. *Automatica*, Vol. 38, str. 737 – 742, 2002.
 - [15] M.M. Asif, P. Angelov, H. Ahmed. An approach to real-time color-based object tracking. *Evolving Fuzzy Systems, 2006 International Symposium on*, str. 86 –91, sept 2006.
 - [16] K J Åstrom, B Wittenmark. *Adaptive control*. Addison-Wesley Publishing Company, USA, 2 edition, 1995.
 - [17] M F Azeem, H Hanmandlu, N Ahmad. Structure identification of generalized adaptive neuro-fuzzy inference systems. *IEEE Trans. on Fuzzy Syst.*, Vol. 11, No. 5, str. 666–681, 2003.
 - [18] Robert Babuška. Fuzzy systems, modeling and identification. tutorial.
 - [19] B W Bequette. Nonlinear control of chemical processes: A review. *Ind. Eng. Chem. Res.*, Vol. 30, str. 1391–1413, 1991.

-
- [20] J C Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, New York, 1981.
- [21] Z Binder. About a multimodel control methodology, algorithm, multi-processors, implementation and application. *8th IFAC World Congress*, str. 981–986, Kyoto, 1991. IFAC.
- [22] Sašo Blažič, Igor Škrjanc, Samo Gerksič, Gregor Dolanc, Stanko Strmčnik, Mincho B. Hadjiski, Anna Stathaki. Online fuzzy identification for an intelligent controller based on a simple platform. *Engineering Applications of Artificial Intelligence*, Vol. 22, str. 628–638, 2009.
- [23] H. Bouhenchir, M. Cabassud, M.V. Le Lann. Predictive functional control for the temperature control of a chemical batch reactor. *Computers & Chemical Engineering*, Vol. 30, No. 6-7, str. 1141 – 1154, 2006.
- [24] A Boulkroune, M Tadjine, M M Saad, M Farza. Fuzzy adaptive controller for MIMO nonlinear systemy with known and unknown control direction. *Fuzzy Sets and Syst.*, Vol. 161, str. 797–820, 2010.
- [25] Gregor Brdavič, Nadja Hvala, Juš Kocijan, Dani Jurič ič. Nelinearni algoritem za estimacijo stanj in identifikacijo parametrov šaržnega biološkega procesa. *Elektrotehniški vestnik*, Vol. 68, str. 57 – 63, 2001.
- [26] E. F. Camacho, C. Bordons. *Model Predictive Control*. Springer-Verlag, London, 2004.
- [27] D Cecil, M Kozłowska. Software sensors are a real alternative to true sensors. *Environmental Modelling and Software*, Vol. 25, str. 622 – 625, 2010.
- [28] P T Chan, A B Rad, J Wang. Indirect adaptive fuzzy sliding mode control: Part II: parameter projection and supervisory control. *Fuzzy Sets and Syst.*, Vol. 122, str. 31–43, 2001.
- [29] J S Chang, W Y Hseih. Optimization and control of the semi-batch reactors. *Ind. Eng. Chem. Res.*, Vol. 33, str. 545–556, 1991.
- [30] L Chen, G Bastin, V Van Breusegam. A case study of adaptive nonlinear regulation of fed-batch reactors. *Automatica*, Vol. 31, str. 55–65, 1995.

-
- [31] Q Chen, S Tan, Y Han, Z Wang. Adaptive fuzzy scheme for efficient, fast valving control. *Control Eng. Practice.*, Vol. 5, No. 6, str. 811–821, 1997.
 - [32] S L Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intellegent and Fuzzy Systems*, Vol. 2, str. 267–278, 1994.
 - [33] Dong-Jin Choi, Heekyung Park. A hybrid artificial neural network as a software sensor for optimal control of a wastewater treatment process. *Water Research*, Vol. 35, No. 16, str. 3959 – 3967, 2001.
 - [34] D W Clarke. *Advances in model-based predictive control*. Oxford Sci., Oxford, U.K., 1994.
 - [35] D W Clarke, C Mohtadi. Properties of generalized predictive control. *Automatica*, Vol. 25, No. 6, str. 859–875, 1989.
 - [36] D W Clarke, C Mohtadi, P S Tuffs. Generalized predictive control-Part 1; Part 2. *Automatica*, Vol. 23, No. 2, str. 137–160, 1987.
 - [37] D.W. Clarke. Application of generalized predictive control to industrial processes. *Control Systems Magazine, IEEE*, Vol. 8, No. 2, str. 49 –55, apr 1988.
 - [38] W G Cochran. The distribution of quadratic forms in a normal system, with applications to the analysis of covariance. *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 30, No. 2, str. 178–191, 1934.
 - [39] Fernando Crespoa, Richard Weberb. A methodology for dynamic data mining based on fuzzy clustering. *Fuzzy Sets and Systems*, Vol. 150, str. 267–284, 2005.
 - [40] J E Cuthrell, L T Biegler. Simultaneous optimization and solution methods for semi-batch reactor control. *Comput. Chem. Eng.*, Vol. 13, str. 49–62, 1989.
 - [41] C R Cutler, B L Ramaker. *Dynamic matrix control-A computer control algorithm*. ACC, San Francisco, 1980.
 - [42] Gustafson D, Kessel W. Fuzzy clustering with fuzzy covariance matrix. *Proc. IEEE CDC*, str. 761–766, San Diego, CA, USA, 1979.

-
- [43] Cedric Damour, Michel Benne, Brigitte Grondin-Perez, Jean-Pierre Chabriet. Soft-sensor for industrial sugar crystallization: On-line mass of crystals, concentration and purity measurement. *Control Engineering Practice*, Vol. 18, str. 839 – 844, 2010.
- [44] M Daszykowski, B Walczak, D L Massart. Projection methods in chemistry. *Chemometrics and Intelligent Laboratory Systems*, Vol. 65, str. 97–112, 2003.
- [45] José de Jesús Rubio. *EVOLVING INTELLIGENT SYSTEMS Methodology and Applications*, poglavje STABILITY ANALYSIS FOR AN ONLINE EVOLVING NEURO-FUZZY RECURRENT NETWORK, str. 173–199. John Wiley & Sons, Hoboken, New Jersey, 2010.
- [46] R A J De Vries, H B Verbruggen. *Advances in model-based predictive control*, poglavje Multivariable unified predictive control, str. 84–102. Oxford University Press, Oxford, 1994.
- [47] D Deng, N Kasabov. Evolving self-organizing maps for on-line learning, data analysis and modeling. *Proc. IJCNN 2000 Neural Networks Neural Computing: New Challenges Perspectives New Millenium*, Volume 6, str. 3–8, New York, 1991. IEEE.
- [48] Dejan Dovžan, Igor Škrjanc. Control of mineral wool thickness using predictive functional control. *Robotics and Computer-Integrated Manufacturing*, Vol. , No. 0, str. –, 2011.
- [49] Dejan Dovžan, Igor Škrjanc. Recursive clustering based on a gustafson–kessel algorithm. *Evolving Systems*, Vol. 2, str. 15–24, 2011.
- [50] Dejan Dovžan, Igor Škrjanc. Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes. *ISA Transactions*, Vol. 50, No. 2, str. 159 – 169, 2011.
- [51] Dejan Dovžan. *Študija samonastavljivih algoritmov za prediktivno funkcijski regulator*. Bachelor thesis, Faculty of electrical engineering, Ljubljana, sept. 2008.

-
- [52] D. Dovžan, V. Logar, N. Hvala, I. Škrjanc:. Monitoring and sensor fault detection in a waste-water treatment process based on fuzzy model. *Journal of Electrical Engineering and Computer Science*, Vol. 78, No. 3, str. 142–146, 2011.
- [53] F J Doyle III, T A Ogunnaike, R K Pearson. Nonlinear model-based control using second-order Volterra models. *Automatica*, Vol. 31, No. 5, str. 697–714, 1995.
- [54] Matko Drago. *Identifikacije*. Založba FE in FRI, Tržaška, Ljubljana, 1998.
- [55] C Englund, A Verikas. A hybrid approach to outlier detection in the offset lithographic printing process. *Engineering Applications of Artificial Intelligence*, Vol. 18, str. 759 – 768, 2005.
- [56] Heiko Feitkenhauer, Ulrich Meyer. Software sensors based on titrimetric techniques for the monitoring and control of aerobic and anaerobic bioreactors. *Biochemical Engineering Journal*, Vol. 17, str. 147 – 151, 2004.
- [57] D Filbert, L Metzger. Quality test of systems by parameter estimation. *Proceedings of 9th IMEKO-Congress*, Berlin, Germany, May 1982.
- [58] Martin Fischer, Oliver Nelles, Rolf Isermann. Adaptive predictive control of a heat exchanger based on a fuzzy model. *Control Engineering Practice*, Vol. 6, No. 2, str. 259 – 269, 1998.
- [59] L Fortuna, S Graziani, MG Xibilia. Soft sensors for product quality monitoring in debutanizer distillation columns. *Control Engineering Practice*, Vol. 13, str. 499 – 508, 2005.
- [60] Luigi Fortuna, Salvatore Graziani, Alessandro Rizzo, Maria G Xibilia. *Soft Sensors for Monitoring and Control of Industrial Processes*. Springer-Verlag, London, 2007.
- [61] B A Foss, T A Johansen, A V Sorensen. Nonlinear predictive control using local models-applied to a batch fermentation process. *Contrl Eng. Part 3*, Vol. 36, str. 389–396, 1995.

-
- [62] H Frigui, R Krishnapuram. A robust algorithm for automatic extraction of an unknown number of clusters from noisy data. *Pattern Recognition Letters*, Vol. 17, str. 1223–1232, 1996.
- [63] B Fritzke. A growing neural gas network learns topologies. *Adv. Neural Inform. Processing Syst.*, Vol. 7, str. 845–865, 1995.
- [64] Robert Fuller. *Neural Fuzzy Systems*. Abo Akademi University, Abo, 1995.
- [65] T M McGinnty G Leng, G Prasad. An on-line algorithm for creating self-organizing fuzzy neural networks. *Neural Networks*, Vol. 17, No. 10, str. 1477–1493, 2004.
- [66] Carlos E. García, David M. Prett, Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, Vol. 25, No. 3, str. 335 – 348, 1989.
- [67] I Gath, B Geva. Unsupervised optimal fuzzy clustering. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, Vol. 11, No. 7, str. 773–781, 1989.
- [68] J Gertler. *Fault detection and diagnosis in engineering systems*. Marcel Dekker, New York, 1998.
- [69] N Golea, A Golea, K Benmahammed. Stable indirect fuzzy adaptive control. *Fuzzy Sets and Syst.*, Vol. 137, str. 353–366, 2003.
- [70] S Graziani, N Pitrone, MG Xibilia, N Barbalace. Improving monitoring of NOx emissions in refineries. *Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference*, Volume 1, str. 594 – 597, Como, Italy, 18-20 May 2004.
- [71] S Graziani, N Pitrone, MG Xibilia, N Barbalace. Robust BOD_5 Soft Sensor Design Using Local Learning. *Proceedings of the 2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, Volume 3, str. 584 – 589, Taiyuan, China, 22-24 October 2010.
- [72] Laurent Hartert, Moamar Sayed Mouchaweh, Patrice Billaudel. A semi-supervised dynamic version of fuzzy k-nearest neighbours to monitor evolving systems. *Evolving Systems*, Vol. 1, str. 3–15, 2010.

-
- [73] B Hassibi, D G Stork. Second-order derivatives for network pruning: Optimal brain surgeon. *Advances in Neural Information Processing*, Vol. 4, str. 164–171, 1993.
- [74] H. V. Henderson, S. R. Searle. On deriving the inverse of a sum of matrices. *SIAM Review*, Vol. 23, No. 1, str. pp. 53–60, 1981.
- [75] M A Henson, D E Seborg. Input-Output linearization of general processes. *AIChE J.*, Vol. 36, str. 17–53, 1995.
- [76] U Holmberg, G Olsson, B Andersson. Simultaneous do control and respiration estimation. *Wat. Sci. Tech.*, Vol. 21, str. 1185–1195, 1989.
- [77] Guang-Bin Huang, P Saratchandran, Narashiman Sundararajan. A recursive growing and pruning rbf (gap-rbf) algorithm for function approximations. *Proceedings of The Fourth International Conference on Control and Automation (ICCA'03)*, str. 10 – 12, Montreal, Canada, June 2003.
- [78] C L Hwang, L J Chang. Fuzzy neural-based control for nonlinear time-varying delay systems. *IEEE Trans. Syst. Man Cyber. part B*, Vol. 37, No. 6, , 2007.
- [79] Rolf Isermann. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer-Verlag, Heidelberg, Germany, 2006.
- [80] Yen J, Wang L, Gillespie C W. Improving the interpretability of tsf fuzzy models by combining global and local learning. *Trans. Fuzzy Syst.*, Vol. 6, str. 530–537, Nov. 1998.
- [81] J-S R Jang, C-T Sun. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Trans. Neural Networks*, Vol. 4, str. 1139–1150, 1993.
- [82] Pan Jeng-Shyang, Chen Shyi-Ming, Nguyen Ngoc-Thanh. *Computational Collective Intelligence. Technologies and Applications*, Volume 6421 of *Lecture Notes in Computer Science, Vol. 6421*. Springer, Berlin, 1st edition, 2010.

-
- [83] T A Johanson, R Murray-Smith. *Operating regime approach to nonlinear modeling and control*. Taylor Francis, 1981.
- [84] R A Johnson, D W Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, New Jersey, 1992.
- [85] C F Juang, C T Lin. An on-line self-constructing neural fuzzy inference network and its applications. *IEEE Trans. on Fuzzy Syst.*, Vol. 6, No. 1, str. 12–32, 1998.
- [86] M R Juba, J W Hamer. Progress and challenges in batch process control. *Chem. Process Control*, Vol. CPC II, str. 139–183, 1986.
- [87] V. Kadiramanathan, M. Niranjan. A function estimation approach to sequential learning with neural networks. *Neural Comput.*, Vol. 5, No. 6, str. 954–975, 1993.
- [88] G Karer, I Škrjanc, B Zupančič. Self-adaptive predictive functional control of the temperature in an exothermic batch reactor. *Chem. eng. process*, Vol. 47, str. 2379–2385, 2008.
- [89] N Kasabov. Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems. *Fuzzy Sets and Systems*, Vol. 82, No. 2, str. 135–149, 1996.
- [90] N Kasabov. Evolving fuzzy neural networks-Algorithms, applications and biological motivation. *Methodologies for the Conceptation, Design and Application of Soft Computing*, str. 271–274, Japan, 1998.
- [91] N K Kasabov. Evolving fuzzy neural networks for supervised/unsupervised on-line knowledge-based learning. *IEEE Trans. Syst. Man Cyber. Part B*, Vol. 31, No. 6, str. 902–918, 2001.
- [92] N K Kasabov, Q Song. DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and its application for time-series prediction. *IEEE Trans. on Fuzzy Syst.*, Vol. 10, No. 2, str. 144–154, 2002.
- [93] Nikola Kasabov. *Evolving Connectionist Systems*. Springer-Verlag, London, 2007.

-
- [94] Kenneth W Kemp. The use of cumulative sums for sampling inspection schemes. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 11, No. 1, str. 16 – 31, 1962.
- [95] J F Kenny, E S Keeping. *Mathematics of Statistics Pt. 2*. Van Nostrand, Princeton NJ, 2 edition, 1951.
- [96] E Kim, Minkee Park, S Ji, Migon Park. A new approach to fuzzy modeling. *IEEE Trans. on Fuzzy Syst.*, Vol. 5, No. 3, str. 328–337, 1997.
- [97] S W Kim, E T Kim, M Park. A new adaptive controller using parallel structure of fuzzy controller and its application. *Fuzzy Sets and Syst.*, Vol. 81, str. 205—226, 1996.
- [98] Yejin Kim, Hyeon Bae, Kyungmin Poo, Jongrack Kim, Taesup Moon, Sungshin Kim, Changwon Kim. Soft sensor using pnn model and rule base for wastewater treatment plant. Jun Wang, Zhang Yi, Jacek Zurada, Bao-Liang Lu, Hujun Yin, uredniki, *Advances in Neural Networks - ISNN 2006*, Volume 3973 of *Lecture Notes in Computer Science*, str. 1261–1269. Springer Berlin / Heidelberg, 2006.
- [99] D Kukolj, E Levi. Identification of Complex Systems Based on Neural and Takagi–Sugeno Fuzzy Model. *IEEE Trans. Syst. Man Cyber. part B*, Vol. 34, No. 1, str. 272–282, 2004.
- [100] H.-B. Kuntze, A. Jacubasch, J. Richalet, Ch. Arber. On the predictive functional control of an elastic industrial robot. *Decision and Control, 1986 25th IEEE Conference on*, Volume 25, str. 1877 –1881, dec. 1986.
- [101] Ljung L. *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, NJ, 1987.
- [102] S Labiod, T M Guerra. Adaptive fuzzy control of class of SISO nonaffine nonlinear systems. *Fuzzy Sets and Syst.*, Vol. 158, str. 1126–1137, 2007.
- [103] N Lakshmanan, Y Arkun. Estiamtion and control of batch process using multiple models. *Int. J. Control*, Vol. 72, No. 7/8, str. 659–675, 1999.

-
- [104] D Lee, J S Lee, T Kang. Adaptive fuzzy control of the molten steel level in a strip-casting process. *Control Eng. Practice.*, Vol. 4, No. 11, str. 1511–1520, 1996.
- [105] Min Woo Lee, Sung Hun Hong, Hyeoksun Choi, Ji-Hoon Kim, Dae Sung Lee, Jong Moon Park. Real-time remote monitoring of small-scaled biological wastewater treatment plants by a multivariate statistical process control and neural network-based software sensors. *Process Biochemistry*, Vol. 43, No. 10, str. 1107 – 1113, 2008.
- [106] Gang Leng, Girijesh Prasad, Thomas Martin McGinnity. An on-line algorithm for creating self-organizing fuzzy neural networks. *Neural Networks*, Vol. 17, str. 1477 – 1493, 2004.
- [107] M Lepetič, I Škrjanc, H G Chiacchiarini, D Matko. Predictive functional control based on fuzzy model: magnetic suspension system case study. *Engineering Applications of Artificial Intelligence*, Vol. 16, str. 425–430, 2003.
- [108] C S Leung, K W Wong, P F Sum, L W Chan. A pruning method for the recursive least squared algorithm. *Neural Networks*, Vol. 14, str. 147–174, 2001.
- [109] Bao Lin, Bodil Recke, Jøgen K H Knudsen, Sten Bay Jørgensen. A systematic approach for soft sensor development. *Computers and Chemical Engineering*, Vol. 31, str. 419 – 425, 2007.
- [110] C T Lin. A neural fuzzy control system with structure and parameter learning. *Fuzzy Sets and Syst.*, Vol. 70, str. 183–212, 1995.
- [111] F J Lin, C H Lin, P H Shen. Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive. *IEEE Trans. on Fuzzy Syst.*, Vol. 9, No. 5, str. 751–759, 2001.
- [112] C F Lindberg. *Control and Estimation Strategies applied to the activated sludge process*. Doktorska disertacija, Uppsala University, Stockholm, Sweden, 1997.

-
- [113] C Loebelin, J D Perkin, B Srinivasan, D Bonvin. Economic performance analysis in the design of online batch optimization system. *J. Process Control*, , No. 9, str. 61–78, 1999.
- [114] V. Logar, D. Dovžan, I. Škrjanc. Mathematical modeling and experimental validation of an electric arc furnace. *ISIJ International*, Vol. 51, No. 3, str. 382–391, 2012.
- [115] V. Logar, D. Dovžan, I. Škrjanc. Modeling and validation of an electric arc furnace: Part 1, heat and mass transfer. *ISIJ International*, Vol. 52, No. 3, str. 402–412, 2012.
- [116] V. Logar, D. Dovžan, I. Škrjanc. Modeling and validation of an electric arc furnace: Part 2, thermo-chemistry. *ISIJ International*, Vol. 52, No. 3, str. 413–423, 2012.
- [117] Z Louleh, M Cabassud, M V Le Lann. A new strategy for temperature control of batch reactors: Experimental application. *Chem. Eng. J.*, Vol. 75, str. 11–20, 1999.
- [118] Edwin Lughofer. Evolving Fuzzy Systems-Methodologies, Advanced Concepts and Applications. Volume 266 of *Studies in Fuzziness and Soft Computing Series*, str. 476. Springer-Verlag, Berlin Heidelberg, 1 edition, January 2011.
- [119] Edwin Lughofer. A dynamic split-and-merge approach for evolving cluster models. *Evolving Systems*, Vol. DOI: 10.1007/s12530-012-9046-5, , 2012.
- [120] Edwin Lughofer, Jean-Luc Bouchot, Ammar Shaker. On-line elimination of local redundancies in evolving fuzzy systems. *Evolving Systems*, , No. 2, str. 165–187, 2011.
- [121] Edwin Lughofer, Carlos Guardiola. On-line fault detection with data-driven evolving fuzzy models. *Control and Intelligent Systems*, Vol. 36, No. 4, str. 307–317, 2008.
- [122] Edwin Lughofer, Erich Peter Klement. Flexfis: A variant for incremental learning of takagi-sugeno fuzzy systems. *Proceedings of The 2005 IEEE*

- International Conference on Fuzzy Systems*, str. 915 – 920, Reno, Nevada, USA, 22-25 May 2005.
- [123] JX Luo, HH Shao. Developing soft sensors using hybrid soft computing methodology: a neurofuzzy system based on rough set theory and genetic algorithms. *Soft Computing*, Vol. 10, str. 54 – 60, 2006.
- [124] R Luus, O N Okongwu. Towards practical optimal control of batch reactors. *Chem. Eng. J.*, Vol. 75, str. 1–9, 1999.
- [125] L.Yingwei, N. Sundararajan, P. Saratchandran. A sequential learning scheme for function approximation using minimal radial basis function (rbf) neural networks. *Neural Comput.*, Vol. 9, str. 461–478, 1997.
- [126] Hall P. M., Hicks Y. Tehnično poročilo.
- [127] M C Mackey, L Glass. Oscilations and chaos in physiological control systems. *Science*, Vol. 197, No. 4300, str. 287–289, 1977.
- [128] M.W. Mak, C.K. Li, X. Li. Maximum likelihood estimation of elliptical basis function parameters with application to speaker verification. *Signal Processing Proceedings, 1998. ICSP '98. 1998 Fourth International Conference on*, Volume 2, str. 1287 –1290 vol.2, 1998.
- [129] Kamyar Mehran. Takagi-sugeno fuzzy modeling for process control. tutorial.
- [130] Dejan Gradišar Miha Glavan, Stanko Strmčnik. Metodologije za določitev relevantnih krmilnih vhodov. Projektni rezultat RRP3.1-D03, IJS, junij 2012.
- [131] O. Nelles, R. Isermann. Basis function networks for interpolation of local linear models. *Decision and Control, 1996., Proceedings of the 35th IEEE*, Volume 1, str. 470 –475 vol.1, dec 1996.
- [132] Nelles O. Local linear model tree for on-line identification of time invariant nonlinear dynamic systems. *Proceedings of Int Conf on Artificial Neural Networks (ICANN)*, str. 115–120, Bochum, Germany, 1996.

-
- [133] Nelles O. *Nonlinear system identification*. Springer, London, 2001.
- [134] S. Oblak, I. Skrjanc. Multivariable fuzzy predictive functional control of a mimo nonlinear system. *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, str. 1029 –1034, june 2005.
- [135] P Angelov and V Giglio and C Guardiola and E Lughofer and J M Luján. An approach to model-based fault detection in industrial measurement systems with application to engine test benches. *Measurement Science And Technology*, Vol. 17, str. 1809 – 1818, 2006).
- [136] E Page. Continous inspection schemes. *Biometrika*, Vol. 41, str. 100 – 115, 1954.
- [137] R. P. Paiva, A. Dourado. Structure and parameter learning of neuro-fuzzy systems: a methodology and a comparative study. *Journal of Intelligent and Fuzzy Systems*, Vol. 11, str. 147–161, 2001.
- [138] J Patt. A resource allocating network for function interpolation. *Neural Computat.*, Vol. 3, No. 2, str. 213–225, 1991.
- [139] P A Phan, T J Gale. Direct adaptive fuzzy control with a self-structuring algorithm. *Fuzzy Sets and Syst.*, Vol. 159, str. 871–899, 2008.
- [140] R Qi, M A Brdys. Stable indirect adaptive control based on discrete-time T-S fuzzy model. *Fuzzy Sets and Syst.*, Vol. 159, str. 900–925, 2008.
- [141] S Qin, W Li, H Yue. Recursive pca for adaptive process monitoring. *Journal of Process Control*, Vol. 10, str. 471–486, 200.
- [142] S.Joe Qin, Thomas A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, Vol. 11, No. 7, str. 733 – 764, 2003.
- [143] Babuska R. *Fuzzy modeling and identification*. Ph.d. thesis, Univ. of Delft, Delft, The Netherlands, 1996.

-
- [144] Yager R R, Filev D P. Learning of fuzzy rules by mountain clustering. *Proc. SPIE Conf. Applicat. Fuzzy Logic Technol*, str. 246–254, Boston MA, 1993.
- [145] J Richalet. Industrial application of model based predictive control. *Automatica*, Vol. 29, No. 5, str. 1251–1274, 1993.
- [146] J Richalet, D O'Donovan. *Predictive Functional Control: Principles and Industrial Applications*. Springer-Verlag, London, 2009.
- [147] J Richalet, A Rault, J L Testud, J Papon. Model predictive heuristic control: Application to industrial processes. *Automatica*, Vol. 14, No. 5, str. 413–428, 1978.
- [148] S Roberts. A comparison of some control chart procedures. *Technometrics*, Vol. 8, str. 411 – 430, 1966.
- [149] I Rojas, H Pomares, J Gonzalez, L J Herrera, A Guillen, F Rojas, O Valenzuela. Adaptive fuzzy controller: Application to the control of the temperature of a dynamic room in real time. *Fuzzy Sets and Syst.*, Vol. 157, str. 2241–2258, 2006.
- [150] H J Rong, N Sundararajan, G B Huang, P Saratchandran. Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction. *Fuzzy Sets and Syst.*, Vol. 157, No. 9, str. 1260–1275, 2006.
- [151] M Shahrokhi S Salehi. Adaptive fuzzy approach for H^∞ temperature tracking control of continuous stirred tank reactors. *Control Engineering Practice*, Vol. 16, str. 1101–1108, 2008.
- [152] Åsa Jansson, Jonas Röttorp, Magnus Rahmberg. Developement of a software sensor for phosphorus in municipal wastewater. *Jurnal of chemometrics*, Vol. 16, str. 542 – 547, 2002.
- [153] J Shing, R Jang. ANFIS : Adaptive-Network-Based Fuzzy Inference System. *IEEE Trans. Syst. Man Cyber.*, Vol. 23, No. 3, str. 665–685, 1993.
- [154] Y P Singh. A modified self-organizing controller for real-time process control applications. *Fuzzy Sets and Syst.*, Vol. 96, str. 147–159, 1998.

-
- [155] I. Skrjanc, B. Hartmann, O. Banfer, O. Nelles, A. Sodja, L. Teslic. Supervised hierarchical clustering in fuzzy model identification. *Fuzzy Systems, IEEE Transactions on*, Vol. PP, No. 99, str. 1, 2011.
- [156] Hossein Soleimani-B, Caro Lucas, Babak N Araabi. Recursive gath-geva clustering as a basis for evolving neuro-fuzzy modeling. *Evolving Systems*, Vol. 1, No. 1, str. 59 – 71, 2010.
- [157] Oscar A Z Sotomayor, Song Won Park, Claudio Garcia. Software sensor for on-line estimation of the microbial activity in activated sludge systems. *ISA Transactions*, Vol. 41, No. 2, str. 127 – 143, 2002.
- [158] H Stark, J Woods. *Probability, random processes and estimation theory for engineers*. Prentice Hall, Englewood Cliffs, New Jersey, 1994.
- [159] T Takagi, M Sugeno. A fuzzy identification of systems and its applications to modelling and control. *IEEE Trans. on Syst., Man and Cyber.*, Vol. 15, No. 1, str. 116–132, 1985.
- [160] MT Tham, AJ Morris, GA Montague. Soft-sensing: a solution to the problem of measurement delays. *Chemical Engineering Research and Design*, Vol. 67, str. 547 – 554, 1994.
- [161] S Tong, X He, Y Li, H Zang. Adaptive fuzzy backstepping robust control for uncertain nonlinear systems based on small-gain approach. *Fuzzy Sets and Syst.*, Vol. 161, str. 771–796, 2010.
- [162] S Tong, Y Li, P Shi. Fuzzy adaptive backstepping robust control for siso nonlinear system with dynamic uncertainties. *Information Science*, Vol. 179, str. 1319–1332, 2009.
- [163] Sau Wai Tung, Chai Quek, Cuntai Guan. SaFIN a self-adaptive fuzzy inference network. *Jurnal of Intelligent Robotic Systems*, Vol. 49, str. 279–292, 2007.
- [164] Sau Wai Tung, Chai Quek, Cuntai Guan. Safin: A self-adaptive fuzzy inference network. *Neural Networks, IEEE Transactions on*, Vol. 22, No. 12, str. 1928 –1940, dec. 2011.

-
- [165] S G Tzafestas, K C Zikidis. NeuroFAST: On-line neuro-fuzzy ART-based structure and parameter learning TSK model. *IEEE Trans. Syst. Man Cyber. Part B*, Vol. 31, No. 5, str. 797–802, 2001.
- [166] Darko Vrecko, Nadja Hvala, Marjeta Stražar. The application of model predictive control of ammonia nitrogen in an activated sludge process. *Water Sci Technol.*, Vol. 64, No. 4, str. 1115 – 1121, 2011.
- [167] I Škrjanc. Adaptive supervisory predictive control of a hybrid fed-batch reactor with slow actuator. *Ind. Eng. Chem. Res.*, Vol. 46, No. 24, str. 8050–8057, 2007.
- [168] I Škrjanc. Self-adaptive supervisory predictive functional control of a hybrid semi-batch reactor with constraints. *Chem. Eng. J.*, Vol. 136, No. 2/3, str. 312–319, 2008.
- [169] I Škrjanc, S Blažič. Predictive functional control based on fuzzy model: Design and stability study. *Journal of Intelligent and Robotic Systems*, Vol. 43, str. 283–299, 2005.
- [170] I Škrjanc, S Blažič, S Oblak, J Richalet. An approach to predictive control of multivariable time-delayed plant: Stability and design issues. *ISA Transactions*, Vol. 43, No. 4, str. 585–595, 2004.
- [171] I Škrjanc, D Matko. Predictive functional control based on fuzzy model for heat-exchanger pilot plant. *IEEE Trans. on Fuzzy Syst.*, Vol. 8, No. 6, str. 705–712, 2000.
- [172] I Škrjanc, D Matko. Fuzzy predictive functional control in the state space domain. *Journal of Intelligent and Robotic Systems*, Vol. 31, No. 1–3, str. 283–297, 2001.
- [173] A Wald. *Sequential analysis*. J. Wiley, New York, 1947.
- [174] D Wang, Xiao-Jun Zeng, J A Keane. An incremental construction learning algorithm for identification of t-s fuzzy systems. *Proc. FUZZ 2008 (IEEE International Conference on Fuzzy Systems 2008)*, str. 1660–1666, Hong Kong, 2008. IEEE.

-
- [175] J Wang, S S Ge, T H Lee. Adaptive fuzzy sliding mode control of a class of nonlinear systems. *Proc. Third Asian Control Conf.*, str. 599–604, 2000.
- [176] J Wang, A B Rad, P T Chan. Indirect adaptive fuzzy sliding mode control: Part I: fuzzy switching. *Fuzzy Sets and Syst.*, Vol. 122, str. 21–30, 2001.
- [177] L X Wang. Stable adaptive fuzzy control of nonlinear systems. *IEEE Transactions on Fuzzy Syst.*, Vol. 1, str. 146–155, 1993.
- [178] L X Wang. *Adaptive fuzzy systems and control: Design and Stability Analysis*. Prentice-Hall, Engelwood Cliffs, NJ, 1994.
- [179] W J Wang. New similarity measures on fuzzy sets and on elements. *Fuzzy Sets and Systems*, Vol. 85, str. 305–309, 1997.
- [180] P Werbos. *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. Ph.d. dissertation, Harvard University, Cambridge, 1974.
- [181] S Wu, M J Er. Dynamic fuzzy neural networks—A novel approach to function approximation. *IEEE Trans. Syst. Man Cyber. Part B*, Vol. 30, No. 2, str. 358–364, 2000.
- [182] S Wu, M J Er, Y Gao. A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks. *IEEE Trans. on Fuzzy Syst.*, Vol. 9, No. 4, str. 578–594, 2001.
- [183] W Yan, H Shao, X Wang. Soft sensing modeling based on support vector machine and Bayesian model selection. *Computers and Chemical Engineering*, Vol. 28, str. 1489 – 1498, 2004.
- [184] Miin-Shen Yang, Chien-Yo Lai. A robust automatic merging possibilistic clustering method. *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, Vol. 19, No. 1, str. 26–41, February 2011.
- [185] Hongtao Ye, Fei Luo, Yuge Xu. Dissolved oxygen generic model control of wastewater treatment process based on immune optimization least squares support vector machine. *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, str. 5082 –5086, 2010.

-
- [186] Chang Kyoo Yoo, In-Beum Lee. Soft sensor and adaptive model-based dissolved oxygen control for biological wastewater treatment processes. *Environmental Engineering Science*, Vol. 21, No. 3, str. 331 – 340, 2004.
- [187] L A Zadeh. Fuzzy sets. *Information and Control*, Vol. 3, No. 8, str. 338–353, 1965.
- [188] L A Zadeh. *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers*. World Scientific Publishing Co, Singapore, 1996.
- [189] Andrej Zdešar, Otta Cerman, Dejan Dovžan, Petr Husek, Igor Škrjanc. Fuzzy control of a helio-crane: Comparison of two control approaches. *Journal of Intelligent and Robotic Systems*, Vol. 0, No. 0, str. 0, 2012.
- [190] B Zhang, C Sconyers, C Byington, R Patrick, M Orchard, G Vachtsevanos. A probabilistic fault detection approach: Application to bearing fault detection. *IEEE Transactions on Industrial Electronics*, Vol. PP, No. 99, str. 1, 2010.
- [191] Lihui Zhou, Tiankun Wang, Pu Han, Daping Xu. Study and application of predictive functional control in networked control system. *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, june 2008.
- [192] R Zurmühl. *Praktische Mathematik*. Springer, Berlin, 5th edition, 1965.

Izjava

Izjavljam, da sem doktorsko delo izdelal samostojno pod vodstvom mentorja prof. dr. Igorja Škrjanca, dipl. inž. el. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.