

Univerza v Ljubljani

Fakulteta za elektrotehniko

Uroš Petković

**Analiza mehkega  
samorazvijajočega se sistema na  
osnovi Gaussovega rojenja**

Magistrsko delo

Magistrski študijski program druge stopnje Elektrotehnika

Mentor: Prof. dr. Igor Škrjanc

Ljubljana, 2020



## Zahvala

Zahvaljujem se mentorju, prof. dr. Igorju Škrjancu, za pomoč pri nastanku magistrskega dela.

Zahvaljujem se tudi vsem profesoricam, profesorjem, asistentkam, asistentom in ostalim pedagoškim delavcem, ki so me spremljali tekom izobraževanja. Posebna zahvala prof. dr. Tomažu Vrtovcu ter dr. Robertu Korezu in izr. prof. dr. Vitomirju Štrucu ter prof. dr. Petru Peeru.

Zahvala tudi staršem za podporo med študijem.



## Povzetek

V magistrskem delu smo predstavili področje samorazvijajočih se sistemov, pri čemer smo se osredotočili na sisteme z mehko logiko. Osrednji del naloge je predstavitev in analiza obstoječega samorazvijajočega sistema eGAUSS+. Sistem temelji na rojenju glede na vrednost pripadnostne funkcije, ki je določena z Gaussovo funkcijo. Roji sistema se združujejo glede na primerjavo prostornin prostora, ki ga pokrivajo. Metodo smo modificirali in ji dodali dodatni vhodni parameter za omejitev velikosti rojev. Analizirali smo vpliv posameznih vhodnih parametrov sistema na samo rojenje. Pokazali smo, da dodani parameter bistveno izboljša ponovljivost nenadzorovanega rojenja. Metoda je uporabna za primer identifikacije vhodno-izhodnega dinamičnega sistema in za primer nadzorovanega razvrščanja. Preizkusili smo sistem na problemu vhodno-izhodne identifikacije dveh nelinearnih dinamičnih sistemov. Določili smo tudi interval zaupanja napovedi in raziskali vpliv predhodnega filtriranja regresorjev na rojenje in identifikacijo. Delovanje sistema smo ocenili še na primeru nadzorovanega razvrščanja in rezultate primerjali s preostalimi razvrščevalniki.

**Ključne besede:** samorazvijajoči se sistemi, tok podatkov, mehki sistemi



## Abstract

In this thesis, we presented the field of evolving systems with an emphasis on ones with fuzzy logic. The main focus of this work is a recently presented evolving eGAUSS+ system. The system is based on Gaussian clustering, where clusters are merged depending on the comparison between the sum of volumes of two clusters. The method was modified and a new input parameter for cluster volume control was added. We analysed the input parameters of the system, and evaluated their impact on clustering performance. We showed that the added input parameter significantly improves the repeatability of unsupervised clustering. This method can be used for input-output identification and supervised classification. The performance of input-output identification was evaluated on two different, nonlinear dynamical systems. We determined the confidence interval of model output and examined the effect of filtering of regressors on clustering and identification as well. The method performance was evaluated for supervised classification and the obtained results were compared with other classifiers.

**Key words:** evolving systems, data streams, fuzzy systems



# Vsebina

|          |                                                |           |
|----------|------------------------------------------------|-----------|
| <b>1</b> | <b>Uvod</b>                                    | <b>1</b>  |
| <b>2</b> | <b>Tok podatkov</b>                            | <b>3</b>  |
| <b>3</b> | <b>Samorazvijajoči se sistemi</b>              | <b>7</b>  |
| 3.1      | Mehki sistemi . . . . .                        | 9         |
| 3.1.1    | Model Takagi-Sugeno . . . . .                  | 9         |
| 3.2      | Samorazvijajoči se mehanizmi . . . . .         | 10        |
| 3.2.1    | Dodajanje rojev . . . . .                      | 11        |
| 3.2.2    | Delitev rojev . . . . .                        | 11        |
| 3.2.3    | Odstranjevanje rojev . . . . .                 | 12        |
| 3.2.4    | Združevanje rojev . . . . .                    | 13        |
| 3.2.5    | Pregled mehanizmov za združevanje . . . . .    | 13        |
| 3.3      | Pregled samorazvijajočih se sistemov . . . . . | 14        |
| <b>4</b> | <b>Metoda</b>                                  | <b>17</b> |
| 4.1      | Predstavitev rojev . . . . .                   | 17        |

---

|          |                                                                  |           |
|----------|------------------------------------------------------------------|-----------|
| 4.2      | Rekurzivno računanje parametrov roja . . . . .                   | 19        |
| 4.3      | Samorazvijajoče se Gaussovo rojenje . . . . .                    | 20        |
| 4.3.1    | Dodajanje rojev . . . . .                                        | 20        |
| 4.3.2    | Združevanje rojev . . . . .                                      | 21        |
| 4.4      | Identifikacija vhodno-izhodnega modela . . . . .                 | 23        |
| 4.4.1    | Določitev parametrov lokalnega modela . . . . .                  | 24        |
| 4.4.2    | Določitev izhoda mehkega modela . . . . .                        | 26        |
| 4.4.3    | Modeliranje dinamičnih sistemov . . . . .                        | 26        |
| 4.4.4    | Določitev intervala zaupanja . . . . .                           | 27        |
| 4.4.5    | Identifikacija s predhodnim filtriranjem . . . . .               | 28        |
| 4.5      | Nadzorovano razvrščanje . . . . .                                | 29        |
| <b>5</b> | <b>Eksperimenti in rezultati</b>                                 | <b>31</b> |
| 5.1      | Rojenje z dvodimenzionalnimi podatki . . . . .                   | 31        |
| 5.1.1    | Predstavitev podatkov . . . . .                                  | 31        |
| 5.1.2    | Rezultati . . . . .                                              | 32        |
| 5.2      | Identifikacija Hammersteinovega modela . . . . .                 | 38        |
| 5.2.1    | Predstavitev sistema . . . . .                                   | 38        |
| 5.2.2    | Rezultati . . . . .                                              | 39        |
| 5.3      | Identifikacija Hammersteinovega modela s predhodnim filtriranjem | 48        |
| 5.3.1    | Predstavitev sistema . . . . .                                   | 48        |

---

|          |                                              |           |
|----------|----------------------------------------------|-----------|
| 5.3.2    | Rezultati . . . . .                          | 49        |
| 5.4      | Identifikacija naprave Helio-crane . . . . . | 58        |
| 5.4.1    | Predstavitev podatkov . . . . .              | 58        |
| 5.4.2    | Rezultati . . . . .                          | 58        |
| 5.5      | Razvrščanje . . . . .                        | 66        |
| 5.5.1    | Razvrščanje bolezni dojk . . . . .           | 67        |
| 5.5.2    | Razvrščanje perunik . . . . .                | 67        |
| <b>6</b> | <b>Zaključek</b>                             | <b>69</b> |
|          | <b>Literatura</b>                            | <b>71</b> |
| <b>A</b> | <b>Grafični vmesnik za rojenje</b>           | <b>79</b> |



## Seznam slik

|     |                                                                    |    |
|-----|--------------------------------------------------------------------|----|
| 2.1 | Vpliv spremembe koncepta . . . . .                                 | 4  |
| 2.2 | Tipi sprememb koncepta . . . . .                                   | 5  |
| 3.1 | Delitev samorazvijajočih se sistemov . . . . .                     | 8  |
| 3.2 | Mehanizem dodajanja rojev . . . . .                                | 11 |
| 3.3 | Mehanizem delitve rojev . . . . .                                  | 12 |
| 4.1 | Prikaz predstavitve roja . . . . .                                 | 19 |
| 4.2 | Prikaz izračuna intervala zaupanja . . . . .                       | 27 |
| 4.3 | Prikaz nadzorovanega razvrščanja . . . . .                         | 29 |
| 5.1 | Prostor $Z$ . . . . .                                              | 32 |
| 5.2 | Naključno prihajanje vzorcev. . . . .                              | 33 |
| 5.3 | Vizualizacija rojenja umetnih dvodimenzionalnih podatkov . . . . . | 36 |
| 5.4 | Prostor regresorjev s statično karakteristiko . . . . .            | 40 |
| 5.5 | Vzbujanje APRBS . . . . .                                          | 41 |
| 5.6 | Prikaz rojenja Hammersteinovega modela . . . . .                   | 43 |

---

|      |                                                                    |    |
|------|--------------------------------------------------------------------|----|
| 5.7  | Validacija simulacijskega modela . . . . .                         | 44 |
| 5.8  | Validacija predikcijskega modela . . . . .                         | 45 |
| 5.9  | Validacija predikcijskega modela za deset korakov predikcije . . . | 46 |
| 5.10 | Filtriran vhodno-izhodni prostor . . . . .                         | 49 |
| 5.11 | Signala vzbujanja . . . . .                                        | 50 |
| 5.12 | Primerjava rojenja s predhodnim filtriranjem . . . . .             | 51 |
| 5.13 | Validacija simulacijskega modela . . . . .                         | 53 |
| 5.14 | Validacija predikcijskega modela . . . . .                         | 54 |
| 5.15 | Validacija predikcijskega modela za deset korakov predikcije . . . | 55 |
| 5.16 | Shema modela Helio-crane . . . . .                                 | 58 |
| 5.17 | Filtriran vhodno-izhodni prostor . . . . .                         | 59 |
| 5.18 | Signala vzbujanja APRBS . . . . .                                  | 60 |
| 5.19 | Prikaz rojev . . . . .                                             | 61 |
| 5.20 | Validacija simulacijskega modela . . . . .                         | 62 |
| 5.21 | Validacija predikcijskega modela . . . . .                         | 63 |
| 5.22 | Validacija predikcijskega modela za deset korakov predikcije . . . | 64 |
| A.1  | Grafični vmesnik . . . . .                                         | 79 |
| A.2  | Signala vzbujanja APRBS . . . . .                                  | 80 |
| A.3  | Signala vzbujanja APRBS . . . . .                                  | 80 |

## Seznam tabel

|      |                                                                                                                                                                                                                                             |    |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 5.1  | Rezultati rojenja dvodimenzionalnih umetnih podatkov z vhodnimi parametri. Prikazani čas umiritve $k_s$ , število rojev ob umiritvi $c_s$ in največje število rojev med rojenjem $c_{max}$ . Vsak poskus je bil ponovljen 100-krat. . . . . | 37 |
| 5.2  | Uporabljeni parametri za rojenje. . . . .                                                                                                                                                                                                   | 41 |
| 5.3  | Primerjava parametrov lokalnih modelov . . . . .                                                                                                                                                                                            | 42 |
| 5.4  | Napaka $RMSE$ za modela brez šuma in modela s šumom. $RMSE$ računa napako med modelom in sistemom brez dodanega šuma. $RMSE_n$ računa napako med modelom in sistemom z dodanim šumom. . . . .                                               | 42 |
| 5.5  | Analiza ocene napake pri enokoračni napovedi. . . . .                                                                                                                                                                                       | 48 |
| 5.6  | Anliza ocene napake pri desetkoračni napovedi. . . . .                                                                                                                                                                                      | 48 |
| 5.7  | Uporabljeni parametri za rojenje. . . . .                                                                                                                                                                                                   | 50 |
| 5.8  | Primerjava parametrov lokalnih modelov . . . . .                                                                                                                                                                                            | 52 |
| 5.9  | Napaka $RMSE$ za model brez šuma in model s šumom. $RMSE$ računa napako med modelom in sistemom brez dodanega šuma. $RMSE_n$ računa napako med modelom in sistemom z dodanim šumom. . . . .                                                 | 56 |
| 5.10 | Analiza ocene napake pri enokoračni napovedi. . . . .                                                                                                                                                                                       | 57 |

---

|      |                                                                                                                                                                                                      |    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 5.11 | Anliza ocene napake pri desetkoračni napovedi. . . . .                                                                                                                                               | 57 |
| 5.12 | Uporabljeni parametri za rojenje. . . . .                                                                                                                                                            | 60 |
| 5.13 | Napaka $RMSE$ za model brez šuma in model s šumom. $RMSE$<br>računa napako med modelom in sistemom brez dodanega šuma.<br>$RMSE_n$ računa napako med modelom in sistemom z dodanim<br>šumom. . . . . | 65 |
| 5.14 | Analiza ocene napake pri enokoračni napovedi. . . . .                                                                                                                                                | 66 |
| 5.15 | Anliza ocene napake pri desetkoračni napovedi. . . . .                                                                                                                                               | 66 |
| 5.16 | Uporabljeni parametri za rojenje. . . . .                                                                                                                                                            | 67 |
| 5.17 | Uspešnost razvrščanja bolezni dojk. . . . .                                                                                                                                                          | 68 |
| 5.18 | Uporabljeni parametri za rojenje. . . . .                                                                                                                                                            | 68 |
| 5.19 | Uspešnost razvrščanja perunik. . . . .                                                                                                                                                               | 68 |

## Seznam uporabljenih simbolov

V pričujočem zaključnem delu so uporabljene naslednje veličine in simboli:

| Veličina / oznaka              |                            | Enota |        |
|--------------------------------|----------------------------|-------|--------|
| Ime                            | Simbol                     | Ime   | Simbol |
| pričakovana vrednost           | $\mu$                      | -     | -      |
| kovariančna matrika            | $\Sigma$                   | -     | -      |
| standardni odklon              | $\sigma$                   | -     | -      |
| normalna/Gaussova porazdelitev | $\mathcal{N}(\mu, \sigma)$ | -     | -      |
| vzorec                         | $\mathbf{z}$               | -     | -      |
| matrika z vzorci               | $\mathbf{Z}$               | -     | -      |
| vhod                           | $\mathbf{x}$               | -     | -      |
| izhod                          | $\mathbf{y}$               | -     | -      |

Pri čimer so matrike zapisane z velikimi črkami in s poudarjeno pisavo. Vektorji so zapisani z malimi črkami in s poudarjeno pisavo. Oglati oklepaji ob vektorju ponazarjajo časovno komponento.



# 1 Uvod

V zadnjem času se je z naraščanjem števila merilnih naprav in izboljšanimi komunikacijskimi potmi število vseh zajetih podatkov izjemno povečalo. Ocena za leto 2012 znaša, da je bilo zajetih in obdelanih 2,8 ZB<sup>1</sup> podatkov. Za leto 2020 se napoveduje, da se bo število povečalo na 40 ZB, kar bi znašalo približno 5 TB podatkov na zemljana v enem letu [1]. Pomemben del podatkov prihaja v obliki podatkovnega toka, kjer zaradi narave aplikacije podatki prihajajo in se obdelujejo sproti ter nepretrgoma, shranjevanje podatkov in njihova kasnejša celostna obravnava pa zaradi številčnosti le teh ni sprejemljiva. Sprotna analiza podatkov nam lahko predstavlja pomembno informacijo o procesu in nam omogoča, da sprejemamo odločitve v realnem času. Tako je prišlo v zadnjem času do pospešenega razvoja metod, ki so sposobni samorazvijajočega se modeliranja, kjer model s sprotno identifikacijo izboljšuje prilaganje svojega odziva merjenim podatkom, pri čemer se lahko spreminjajo njegova struktura in vrednosti parametrov.

Načrtovanje samorazvijajočega se sistema predstavlja velik izziv, saj obstajajo različni kriteriji, katerim je težko popolnoma ugoditi. Od sistema bi želeli: da bi potreboval čim manj vhodnih parametrov za svoje delovanje, da omogoča modeliranje s čim manjšim poznavanjem modeliranega procesa, da lahko učinkovito spreminja ter prilagaja svojo strukturo skozi čas, da je potrebna čim manjša količina shranjenih podatkov in da je računska zahtevnost dovolj majhna za delovanje v realnem času.

Cilj naloge je predstaviti področje samorazvijajočih se sistemov in njihovih mehanizmov za delovanje, pri čemer se bomo osredotočili na sisteme, ki delujejo v skladu z mehko logiko. Glavni cilj naloge je predstavitev obstoječega samoraz-

---

<sup>1</sup>1 zetabajt (ZB) je  $10^{21}$  bajtov

vijajočega sistema [2], ki ga bomo analizirali in podali možne izboljšave. Želimo pokazati, da je sistem mogoče splošno uporabljati; osredotočili se bomo na problem vhodno-izhodne identifikacije, hkrati pa bomo pokazali možnost uporabe metode tudi na primeru razvrščanja.

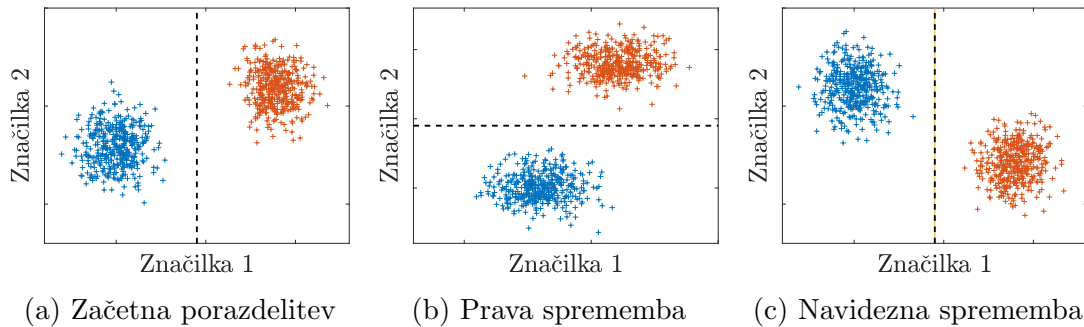
Magistrsko nalogo sestavlja 6 poglavij. V poglavju 1 podamo motivacijo za delo in predstavimo cilje, ki smo si jih zadali v sklopu naloge. V poglavju 2 predstavimo tok podatkov in njegove lastnosti. V poglavju 3 se spoznamo z mehko logiko, samorazvijajočimi se sistemi in mehanizmi, čemur sledi pregled metod mehanizmov za združevanje in pregled samorazvijajočih se sistemov. V poglavju 4 predstavimo metodo, njeno uporabo za vhodno-izhodno identifikacijo z določitvijo intervala zaupanja ter možnim predhodnim filtriranjem podatkov; poglavje zaključimo z uporabo metode za nadzorovano razvrščanje. V poglavju 5 predstavimo eksperimente in dosežene rezultate s predstavljeno metodo, pri čemer smo uporabili dvodimenzionalne umetne podatke, dva nelinearna dinamična sistema in dve pogosto uporabljeni podatkovni zbirki. Delo se konča s poglavjem 6, ki vsebuje zaključek in sklepne ugotovitve.

## 2 Tok podatkov

Tok podatkov (ang. *data streams*) predstavlja vhodne podatke, ki prihajajo z visoko frekvenco. Zaradi tega predstavlja njihova analiza, kjer želimo izluščiti željeno informacijo in jo pripraviti primerno za nadaljnjo uporabo, velik izziv. Glavne značilnosti toka podatkov in izzivi pri njihovi obdelavi so [3, 4]:

- podatki niso znani v naprej, ampak prihajajo sekvenčno v vzorcih,
- vzorci prihajajo skokovito v različnih časovnih intervalih,
- sistem nima nadzora nad vrstnim redom, v katerem prihajajo vzorci,
- načeloma so glede velikosti neomejeni, torej je nemogoče shraniti vse podatke,
- vzorec je dostopen omejeno časovno obdobje, v specifičnih primerih je to samo enkrat, in je nato izbrisan,
- oteženo je nadzorovano učenje, saj so izhodi sistema v določenih primerih dostopni šele čez daljše časovno obdobje,
- statistične lastnosti vzorcev se lahko časovno spreminjajo (nestacionarni procesi).

Tok podatkov je lahko deterministični ali stohastični proces, pri čemer predstavlja modeliranje stohastičnega procesa s samorazvijajočim se sistemom bistveno večji izziv zaradi prisotne naključnosti v naravi procesa. V primeru stohastičnega procesa je potrebno modelirati stohastično spremenljivko  $\mathbf{x}$ , ki ni odvisna



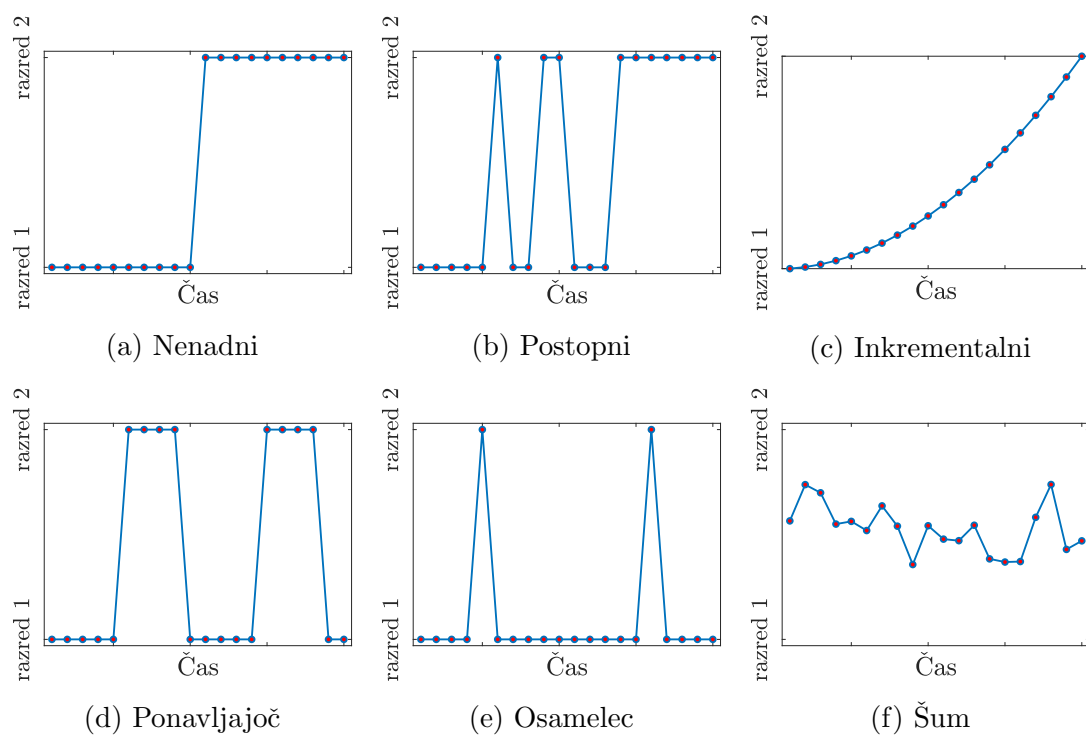
Slika 2.1: Vpliv spremembe koncepta.

samo od eksperimenta, ampak tudi časa. Proces lahko tako opišemo s porazdelitvijo  $p(\mathbf{x}, k)$ , kjer stohastičen proces vidimo kot neskončno število naključnih spremenljivk, vsaka za en časovni trenutek  $k$ . Če se statistične lastnosti procesa ne spreminjajo s časom, torej velja  $p(\mathbf{x}, k_1) = p(\mathbf{x}, k_2)$  za vse  $k_1$  in  $k_2$ , je proces *stacionaren*, drugače *nestacionaren* [5].

Spremembi statističnih lastnosti vzorcev pravimo tudi *sprememba koncepta* (ang. *concept drift*). V primeru klasifikacije poznamo dva različna vpliva, ki ju ima sprememba koncepta (slika 2.1). Ko sprememba koncepta vpliva na odločitveno funkcijo, govorimo o *pravi spremembi koncepta* (ang. *real concept drift*) in predstavlja grožnjo za uspešno delovanje razvrščevalnika. V primeru, da sprememba koncepta ne vpliva na odločitveno funkcijo, govorimo o *navidezni spremembi koncepta*; navidezna sprememba ne vpliva na samo kvaliteto razvrščevalnika, še zmeraj pa je zaželeno, da jo zaznamo [3].

Razlikujemo tudi med različnimi tipi sprememb koncepta (slika 2.2). Vpliv *postopne* in *inkrementalne* spremembe je mogoče odpraviti z adaptivno metodo s področja strojnega učenja, ki sicer ne omogoča spremembe strukture modela, medtem ko je za *nenadni* in *ponavljajoči se* tip koncepta potrebna poleg prilagoditve parametrov tudi sprememba strukture modela, kar omogočajo samorazvijajoči se algoritmi. *Osamelec* in *šum* sta primera anomalij, katera mora sistem filtrirati in prezreti pri gradnji modela, saj je sprememba, ki jo vnašata, naključna [6].

Modeliranje nestacionarnih procesov predstavlja prav poseben izziv. Dober model zgrajen iz toka podatkov bi moral biti robusten na šum in hkrati občutljiv na spremembo koncepta.



Slika 2.2: Tipi sprememb koncepta.



### 3 Samorazvijajoči se sistemi

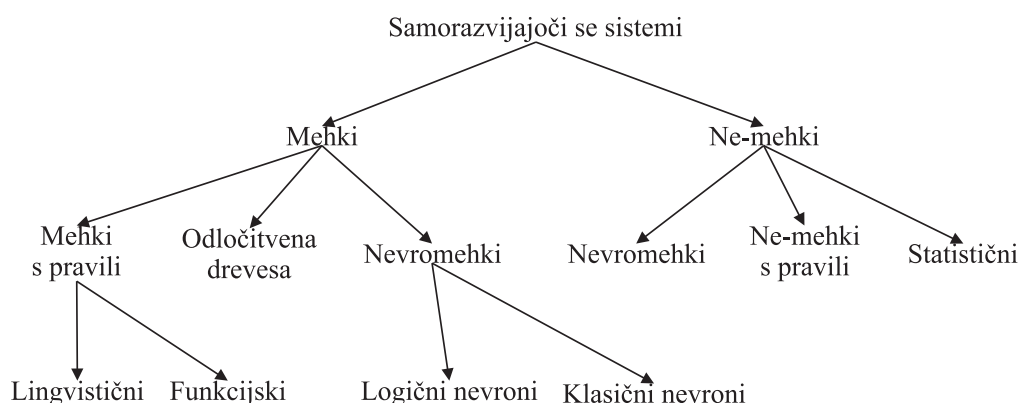
Modeliranja sistemov se v splošnem lahko lotimo s pristopom z *nizom podatkov* (ang. *batch adaptation*) ali s pristopom po *posameznih podatkih* (tudi *trenutno učenje*), ki pride v poštev v primeru toka podatkov. Pri prilagoditvi parametrov po posameznih podatkih je potrebno razlikovati med *iterativnimi algoritmi* in *rekurzivnimi optimizacijskimi algoritmi*. Pri iterativnih algoritmih je potreben večkratni tek skozi podatke in samo na ta način lahko zagotovimo konvergenco. Rekurzivni algoritmi so tisti, ki se uporabljajo v primeru toka podatkov, saj so prilagojeni sekvenčnemu prihajanju podatkov in večkratni tek skozi isto podatkovno bazo ne vnaša dodatne informacije v model.

Rekurzivni algoritmi so tudi del *samorazvijajočih se inteligentnih sistemov* (ang. *evolving intelligent systems*) ali *samorazvijajočih se sistemov* (ang. *evolving systems*). Slednji izraz bomo uporabljali tudi v tem delu. Njihova glavna značilnost je, da so sposobni posodabljanja svojih parametrov in strukture na podlagi sporoti sprejetih podatkov, kar jih razlikuje od adaptivnih sistemov (ang. *adaptive systems*) in sistemov strojnega učenja (ang. *machine learning systems*). Z možnostjo posodabljanja parametrov in strukture se lahko hitro prilagodijo na spremembe v okolju ali opazovanem sistemu, kar je ključnega pomena pri nelinearnih prostorsko-časovnih interakcijah, ki so prisotne v realnih sistemih. Njihovo uporabo lahko zasledimo na področju adaptivnega vodenja, aktivnega učenja, pametnih senzorjev, bioinformatike, sprotnega nadzora in pametnih senzorjev [7].

Eden prvih člankov s področja samorazvijajočih se sistemov je bil objavljen leta 1991 [8], kjer je predstavljen primer sistema zgrajenega z nevronske mreže, vendar v tistem obdobju ni pritegnil velikega zanimanja. S tehnološkim razvojem in potrebo po analizi toka podatkov se samo področje začne bolj pospešeno

razvijati okoli leta 2000. V tem obdobju se je tudi dokončno izoblikoval izraz *samorazvijajoči se sistem* kot sistem, ki vključuje rekurzivni optimizacijski pristop pri obdelavi podatkov in ima sposobnost stalnega spreminjanja, rasti, posodabljanja in krčenja strukture, ko je to potrebno [9]. Začne se tudi razlikovati med samorazvijajočimi se sistemi in *evolucijskimi sistemi* (ang. *evolutionary system*), kamor spadajo genetski algoritmi in genetsko programiranje [10].

V zadnjih dvajsetih letih so se izoblikovali različni tipi samorazvijajočih se sistemov, ki so prikazani na sliki 3.1 in jih delimo na mehke (ang. *fuzzy systems*) in ne-mehke (ang. *non-fuzzy systems*) sisteme [4]. Sistemi se razlikujejo tudi glede na gradnike; gradniki sistema so lahko umetni nevroni (ang. *artificial neurons*), mehka pravila (ang. *fuzzy rules*), podatkovni roji (ang. *data clusters*) ali poddrevesa (ang. *sub-trees*).



Slika 3.1: Delitev samorazvijajočih se sistemov.

Samorazvijajoči se sistemi se učijo nadzorovano ali nenadzorovano in so razviti za probleme rojenja, regresije, identifikacije in razvrščanja. Obširni pregled metod je objavljen v [4], kjer so zbrane predvsem metode, ki bazirajo na mehkih pravilih in hibridnih nevromehkih konceptih. V tem delu se bomo osredotočili predvsem na mehke modele zgrajene iz podatkovnih rojev, kjer je podatkovni roj predstavljen s centrom in področjem vpliva oz. mejo roja.

### 3.1 Mehki sistemi

Mehko logiko je vpeljal Lotfi Aliasker Zadeh leta 1965 [11] kot razširitev klasične (Boolove) logike, ki sicer pripiše izjavi ali spremenljivki vrednost 0, če je ta napačna, ali vrednost 1, če je pravilna; po drugi strani dovoljuje mehka logika prireditve katerekoli vrednosti na intervalu  $[0, 1]$ . Predstavitev mehkih sistemov tako poteka v obliki pravil ČE-POTEM (ang. *IF-THEN*), kjer posamezno pravilo sestavljata pogojni in posledični del. Dokazano je bilo, da lahko z mehki modeli aproksimiramo katerokoli realno zvezno funkcijo do poljubne natančnosti [12]. Slednje dosežemo s povečanjem števila pravil oz. s povečanjem *granulacije* prostora.

Na osnovi mehke logike se je razvilo več modelov. Eden prvih je bil predstavljen model Mamdani [13], ki mu je sledil model Takagi-Sugeno [14]. Leta 2010 je bil predstavljen model AnYa [15, 16, 17] in predstavlja nov neparametrični pristop pri gradnji samorazvijajočih se sistemov, kjer je model zgrajen na podlagi podatkovnih oblakov. V tem delu se bomo osredotočili izključno na model Takagi-Sugeno, ki se najpogosteje uporablja na področju samorazvijajočih se sistemov [18].

#### 3.1.1 Model Takagi-Sugeno

Oblika modela Takagi-Sugeno, ki sta jo avtorja predlagala leta 1985 [14], je predstavljena z naslednjimi pravili:

$$\mathcal{R}^i : \text{ČE } ((\mathbf{x}(1) = A_{i1}) \text{ IN } \dots \text{ IN } (\mathbf{x}(p) = A_{ip})) \text{ POTEM } (y_i(\mathbf{x}) = f_i(\mathbf{x})), \quad (3.1)$$

kjer je indeks  $i = 1, \dots, M$  ter  $M$  predstavlja število pravil,  $\mathbf{x}$  je  $p$ -dimenzionalni vhodni vektor,  $y_i$  izhod modela (posledični del) in  $A_{ij}$  je mehka množica za spremenljivko  $\mathbf{x}(j)$  v pravilu  $\mathcal{R}^i$ . Funkcija  $f_i(\mathbf{x})$  je načeloma splošna, vendar velja omeniti predvsem dve. Če za funkcijo  $f_i$  izberemo skalar  $s_i$ , potem dobimo mehki model Takagi-Sugeno *ničtega reda*, saj je ničti red v Taylorjevi vrsti funkcije konstanta; v kolikor je funkcija  $f_i$  linearna funkcija vhodov, govorimo o modelu *prvega reda*, kjer je izhod enak:

$$y_i(\mathbf{x}) = \omega_{i0} + \omega_{i1}\mathbf{x}(1) + \dots + \omega_{ip}\mathbf{x}(p). \quad (3.2)$$

Za vsako pravilo  $\mathcal{R}^i$  je definirana tudi pripadnostna funkcija  $\gamma^i(\mathbf{x})$ , pri čemer velja:

$$\sum_{i=1}^M \gamma_i(\mathbf{x}) = 1 \quad \text{za vse } \mathbf{x}. \quad (3.3)$$

Slednje lahko zagotovimo tudi z ustreznim normiranjem. Tako lahko izhod iz mehkega Takagi-Sugeno modela izračunamo kot:

$$\hat{y} = \frac{\sum_{i=1}^M f_i(\mathbf{x}) \gamma_i(\mathbf{x})}{\sum_{i=1}^M \gamma_i(\mathbf{x})}. \quad (3.4)$$

Pripadnostna funkcija je v splošnem multivariabilna funkcija vhoda in vsebuje še parametre, ki definirajo obliko in pozicijo roja. Primer pripadnostne funkcije je Gaussova funkcija, kjer pozicija roja definirana s središčem  $\boldsymbol{\mu}$  in oblika roja s kovariančno matriko  $\boldsymbol{\Sigma}$ .

Pripadnostna funkcija je tista, ki v model vnaša nelinearnost, medtem ko posamezni lokalni modeli v primeru Takagi-Sugeno modela prvega reda vnašajo linearnost, kar omogoča interpretabilnost modelov na nekem lokalnem območju. Slednje je tudi razlog, zakaj je model Takagi-Sugeno tako priljubljen, sploh v primeru dinamičnih sistemov, kjer je lokalni model predstavljen z diferenčno enačbo, katere parametri določajo dinamiko sistema.

## 3.2 Samorazvijajoči se mehanizmi

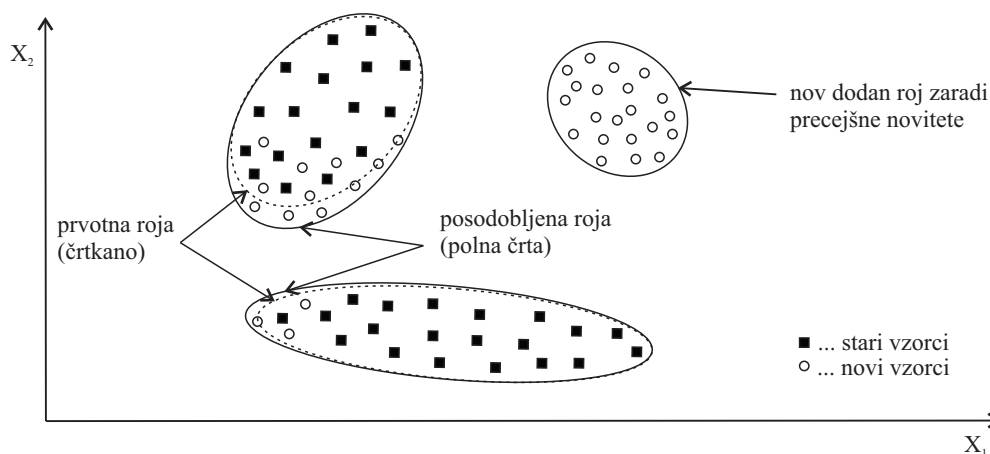
Samorazvijajoči se sistemi morajo biti sposobni prilagajanja parametrov lokalnih modelov glede na spremembe v sistemu. Slednje se lahko doseže z rekurzivno metodo (uteženih) najmanjših kvadratov.

Največji izziv pri gradnji samorazvijajočega se sistema je zagotoviti primerno spreminjanje strukture sistema. V ta namen so se razvili mehanizmi, ki omogočajo dodajanje rojev, delitev rojev, odstranjevanje rojev in združevanje rojev. V tem sklopu bomo opisali posamezne mehanizme, pri čemer bo mehanizem za združevanje rojev predstavljen podrobneje.

### 3.2.1 Dodajanje rojev

Dodajanje rojev je najbolj osnovni samorazvijajoči se mehanizem. Dodajanje je potrebno, ko se v toku podatkov pojavi nov vzorec, ki ga trenutni mehki model ne opisuje. Če je odstopanje novega vzorca manjše, ni nujno potrebno dodati novega roja, saj je v nekaterih primerih dovolj, da zgolj posodobimo parametre obstoječih rojev. Slednje smo prikazali na sliki 3.2.

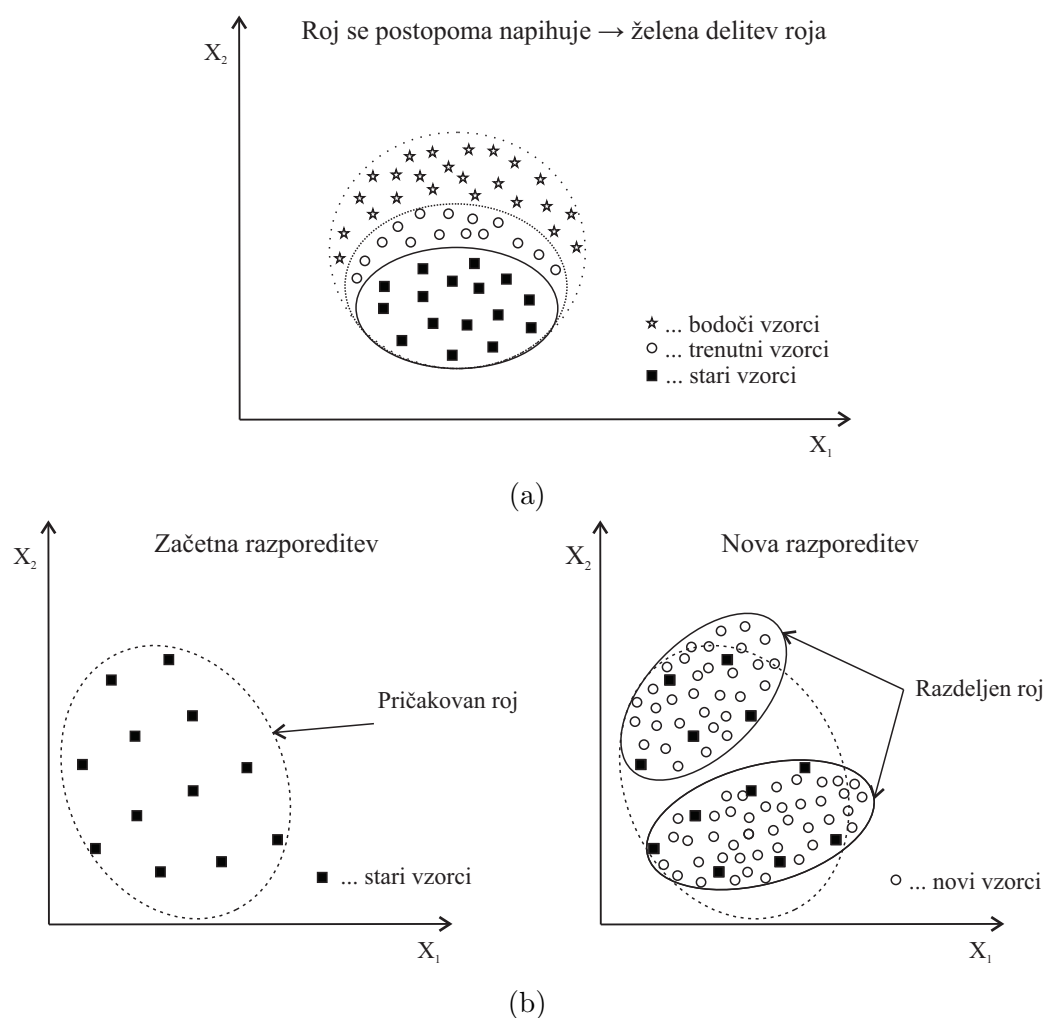
Izoblikovali so se trije kriteriji, na podlagi katerih lahko dodamo nov roj: 1) glede na napako modela na izhodu, 2) glede na razdaljo trenutnega vzorca do središča roja in 3) glede na vrednost pripadnostnih funkcij do obstoječih rojev, pri čemer je potrebno definirati prag, nad/pod katerim stopi mehanizem dodajanja v veljavo [19].



Slika 3.2: Primer dodajanja rojev.

### 3.2.2 Delitev rojev

Delitev rojev je mehanizem, ki omogoča bolj precizno delitev prostora. Delitev je potrebna takrat, ko na nekem območju, ki ga že pokriva nek roj, zaznamo, da je napaka modela tam prevelika oz. narašča. Eden izmed razlogov je lezenje vhodnih podatkov, kar pripelje do napihovanja roja (slika 3.3a). V tem primeru bi želeli roj razdeliti, ohraniti samo aktivni del in odstraniti vpliv starih vzorcev. Drugi razlog je tako imenovani efekt delaminacije (slika 3.3b), ki nastopi ponavadi



Slika 3.3: Primer delitve rojev - (a) v primeru lezenja roja, (b) v primeru efekta delaminacije

takrat, ko se že naučen samorazvijajoči se model uporabi za neko novo aplikacijo in so parametri še zmeraj naravnani na prejšnjo [4].

### 3.2.3 Odstranjevanje rojev

Roje odstranjujemo, da bi zmanjšali računski čas pri delovanju modela, pri čemer z odstranitvijo ne smemo (bistveno) vplivati na napako modela. Taki roji so lahko odstranjeni, saj ne pripomorejo pri opisovanju modeliranega sistema. Mehanizmi za odstranjevanje temeljijo na starosti pravila, številčnosti vzorcev, ki vplivajo na

pravilo, kombinaciji starosti pravila ter aktivacij pravila in na podlagi najmanjše dovoljene razdalje med središči rojev [4].

#### 3.2.4 Združevanje rojev

Združevanje rojev je potrebno, ko pride do precejšnega prekrivanja med roji. Glavni razlog za prekrivanja pogosto tiči v toku podatkov, ko se vzorci obravnavajo zaporedoma in pride do vrzeli na področju, saj vzorci ne pokrivajo področja povsem homogeno. Slednje je prisotno predvsem na začetku učenja, ko je informacije o procesu malo, in je zaželen večja granulacija prostora. Z združevanjem rojev dosežemo bolj kompakten in interpretabilen sistem.

#### 3.2.5 Pregled mehanizmov za združevanje

Predlagane so bile že mnoge metode z različnimi kriteriji za združevanje rojev, vendar se izkaže, da ima vsaka metoda svoje omejitve. V [20] je tako predlagan pristop, kjer se roja združita na podlagi dveh geometrijskih meril (roja morata imeti dovolj podobno orientacijo in središči rojev morata biti dovolj skupaj), ki izluščita primerne roje za združitev. Ker slednji merili sami po sebi ne zagotavljata dovolj širše uspešnosti, jima sledi še odločevalni algoritem, ki temelji na podlagi podobnosti med pripadnostnima funkcijama obeh rojev. Metoda se izkaže za računsko kompleksno in ni primerna za tok podatkov višjih dimenzij.

Metoda FLEXFIS+ [21] uporablja za združevanje rojev kriterij prekrivanja rojev v vseh dimenzijah, kjer se prekrivanje določi na podlagi pripadnostnih funkcij rojev. Če je prekrivanje dovolj veliko, pride do združitve rojev. Zopet se izkaže, da je mera podobnosti med pripadnostnimi funkcijami računsko prezahtevna za podatke višjih dimenzij.

Predlagana je bila tudi metoda GS-EFS [22], kjer so avtorji predlagali dva potrebna kriterija za združevanje rojev, ki so predstavljeni s hiperelipsoidi. Pri prvem kriteriju se izračuna prekrivanje med rojema na osnovi razdalje Bhattacharyya [23, 24], pri drugem pa se upošteva velikost kota med hiperravninama, ki definirata izhod lokalnega modela z največjo lastno vrednostjo. V kolikor je

prekrivanje dovolj veliko in kot dovolj majhen, pride do združitve. Izkaže se, da je računanje prekrivanja časovno zahtevno.

V metodi eFuMo [19] sta implementirana dva načina za združevanje rojev: nadzorovan in nenadzorovan. Nenadzorovan način temelji na normalizirani razdalji med središči rojev in združuje roje, ki so blizu skupaj. Nadzorovan način uporablja tri pogoje, ki bazirajo na lokalnih modelih: primerjava kotov, primerjava korelacije in razmerje razdalj med roji, pri čemer morata biti izpolnjena vsaj dva pogoja za združitev. Metoda vnaša veliko različnih pogojev in dodatnih parametrov, hkrati pa je še zmeraj počasna.

V delu [2], ki je bil tudi motivacija za to nalogo, je predlagan način združevanja na podlagi prostornine rojev, pri čemer se upošteva razmerje prostornin vsote dveh posamičnih rojev in njunega združenega roja. Prednost tega načina združevanja je predvsem majhna računska zahtevnost in malo število parametrov.

### 3.3 Pregled samorazvijajočih se sistemov

Obširen pregled samorazvijajočih se sistemov lahko bralec najde v [4, 25], v tem razdelku bomo omenili zgolj nekatere.

Eden prvih samorazvijajočih se sistemov, ki temeljijo na rojenju, je bil predstavljen leta 2004 in predstavlja samorazvijajoči se pristop Takagi-Sugeno [26]. Osnova modela so roji, kjer se za vsak roj določi pravilo v obliki lokalnega mehkega modela prvega reda. Uporablja se samo mehanizem dodajanja rojev, ki temelji na primerjavi izračuna potenciala med roji ter novim vzorcem in oddaljenostja vzorca od središča rojev. Parametri sistema se rekurzivno posodabljaajo v skladu z uteženimi ter neuteženimi najmanjšimi kvadrati [27]. Kasneje je bilo predstavljenih več podobnih pristopov (npr. [28, 29, 30]), ki se najbolj razlikujejo v načinu rekurzivnega rojenja [25].

Tudi metoda eFuMo [19] je osnovana na mehkem modelu Takagi-Sugeno in rojenju, kjer rojenje temelji na metodi Gustafson-Kessel [31]. Uporabljeni so mehanizmi za dodajanje rojev, odstranjevanje rojev, deljenje rojev in združevanje

rojev, ki je že bil predstavljen v poglavju 3.2.5. Dodajanje rojev poteka na podlagi Mahalanobisove razdalje, odstranjevanje rojev na podlagi številčnosti vzorcev v roju ter njihove starosti, in deljenje na osnovi relativne napake modela. Z večjim številom mehanizmov postane sistem sicer bolj splošen, vendar se tudi bistveno poveča število parametrov.

Predlagana je bila tudi metoda eCauchy [32], ki je prav tako osnovana na mehkem modelu Takagi-Sugeno in rojenju osnovanem na Cauchyevi porazdelitvi. Rojenje je zastavljeno zelo splošno, prilagaja se ga z različnimi izbirami norm notranjega produkta; v posebnem primeru je metoda enaka metodi rojenja s c-možnimi roji (ang. *Possibilistic C-Means clustering* ali PCM) [33]. Metoda predvideva tudi vse osnovne samorazvijajoče se mehanizme - dodajanje rojev, odstranjevanje rojev, združevanje rojev in delitev rojev, pri čemer je število parametrov manjše kot pri metodi eFuMo. Identifikacija vhodno-izhodnega modela poteka na podlagi metode regresije na osnovi glavnih komponent (ang. *principal component regression* ali PCR).

V [2] je objavljena metoda eGAUSS+, ki prinaša nov pristop predvsem pri združevanju rojev na podlagi volumnov le-teh. Metoda temelji na mehkem rojenju, kjer se uporablja Gaussova funkcija. Uporabljata se samo mehanizem za dodajanje rojev in mehanizem za združevanje rojev, pri čemer ima sistem zgolj tri parametra, ki jih moramo določiti pred začetkom. Tudi pri tej metodi identifikacija vhodno-izhodnega modela poteka na podlagi metode regresije na osnovi glavnih komponent. Metoda je bila tudi motivacija za to magistrsko delo in bo podrobneje predstavljena v poglavju 4, kjer so ji bile dodane tudi manjše modifikacije.



## 4 Metoda

V poglavju bomo opisali samorazvijajoči se sistem, ki temelji na delu [2] in je sestavljen iz Gaussovega rojenja ter mehanizma za združevanje rojev na podlagi primerjave prostornine rojev. V prvem delu tega poglavja bomo predstavili Gaussovo rojenje, čemur bo sledil opis, kako metodo uporabljati za potrebe vhodno-izhodne identifikacije in razvrščanja.

### 4.1 Predstavitev rojev

Gaussovo rojenje je osnovano na funkciji gostote verjetnosti *Gaussove porazdelitve* oz. *normalne porazdelitve*. Slednja je za posamezno naključno spremenljivko  $x$  definirana kot:

$$f_u(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (4.1)$$

kjer  $\mu$  predstavlja pričakovano vrednost in  $\sigma^2$  varianco naključne spremenljivke  $x$ . V primeru večih naključnih spremenljivk, ki jih lahko združimo v vektor  $\mathbf{x} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(m)]^T$  in imajo pričakovano vrednost  $\boldsymbol{\mu} = [\boldsymbol{\mu}(1), \boldsymbol{\mu}(2), \dots, \boldsymbol{\mu}(m)]^T$  ter kovariančno matriko  $\boldsymbol{\Sigma}$  velikosti  $m \times m$ , Gaussovo funkcijo zapišemo kot:

$$f_m(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{|2\pi\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}. \quad (4.2)$$

V tem delu bomo uporabljali zgolj nenormalizirano obliko funkcijo gostote verjetnosti Gaussove porazdelitve večih spremenljivk:

$$\gamma[k] = e^{-d^2[k]}, \quad (4.3)$$

ki nam bo služila kot mera za pripadnost oz. "tipičnost" (ang. *typicality*) vzroca  $z[k]$  v časovnem trenutku  $k$  k posameznemu roju, pri čemer:

$$d^2[k] = \frac{1}{2} (\mathbf{z}[k] - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{z}[k] - \boldsymbol{\mu}) \quad (4.4)$$

predstavlja Mahalanobisovo razdaljo od središča roja  $\boldsymbol{\mu}$ . Središče  $\boldsymbol{\mu}$  je definirano preko povprečne vrednosti vzorcev  $\mathbf{z}$  v roju:

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{z}(i), \quad (4.5)$$

kjer  $n$  in  $\mathbf{z}(i)$  predstavljata število vzorcev in  $i$ -ti vzorec roja dolžine  $m$ .  $\boldsymbol{\Sigma}$  predstavlja kovariančno matriko opazovanega roja in je definirana kot:

$$\boldsymbol{\Sigma} = \frac{1}{n-1} (\mathbf{Z} - \mathbf{E}\mathbf{M})^T (\mathbf{Z} - \mathbf{E}\mathbf{M}), \quad (4.6)$$

kjer  $\mathbf{Z}$  predstavlja matriko vseh vzorcev v roju velikosti  $n \times m$ ,  $\mathbf{Z} = [\mathbf{z}(1), \dots, \mathbf{z}(n)]^T$ ,  $\mathbf{M}$  predstavlja diagonalno matriko velikosti  $m \times m$ ,  $\mathbf{M} = \text{diag}(\boldsymbol{\mu}(1), \dots, \boldsymbol{\mu}(m))$ , in  $\mathbf{E}$  predstavlja matriko velikosti  $n \times m$ , kjer so vsi elementi enaki 1.

Kovariančno matriko  $\boldsymbol{\Sigma}$  lahko zapišemo tudi s singularnim razcepom:

$$\boldsymbol{\Sigma} = \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^T. \quad (4.7)$$

Tako lahko Mahalanobisovo razdaljo (enačba 4.4) zapišemo v obliki vsote:

$$d^2[k] = \sum_{i=1}^m \frac{1}{\lambda_i} (\mathbf{z}[k] - \boldsymbol{\mu})^T \mathbf{p}_i \cdot \mathbf{p}_i^T (\mathbf{z}[k] - \boldsymbol{\mu}), \quad (4.8)$$

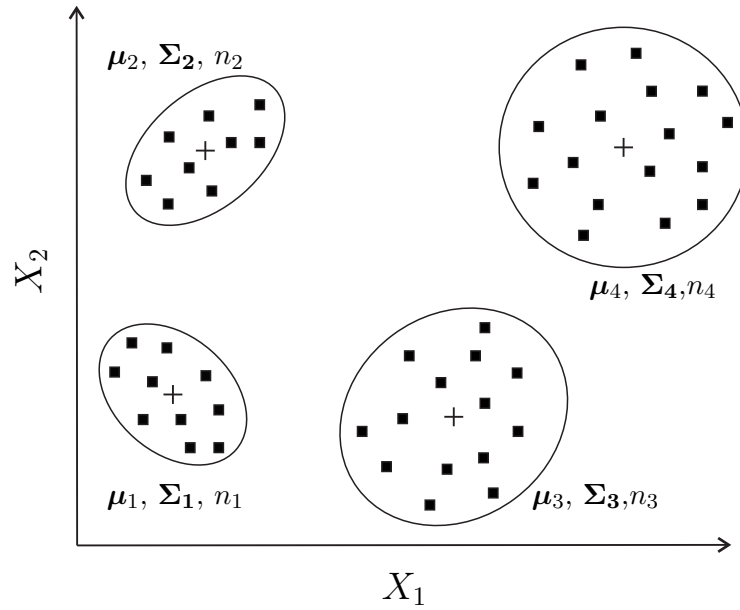
kjer  $\mathbf{p}_i$  predstavlja  $i$ -ti lastni vektor kovariančne matrike  $\boldsymbol{\Sigma}$  z lastno vrednostjo  $\lambda_i$ . Iz takšnega zapisa lahko vidimo, da je Mahalanobisova razdalja normirana vsota razdalj vzorca  $\mathbf{z}$  do središča roja v smeri lastnih vektorjev, ki je utežena s pripadajočo lastno vrednostjo lastnega vektorja. Mahalanobisova razdalja nam omogoča zapis poljubno zavrtenih hiperelipsoidnih rojev, kjer obliko ter rotacijo roja določa kovariančna matrika  $\boldsymbol{\Sigma}$  in kjer je središče roja določeno z vektorjem  $\boldsymbol{\mu}$ .

Hiperelipsoidna oblika roja omogoča tudi enostaven izračun prostornine roja, saj jo dobimo kot:

$$V = \frac{2\pi^{m/2}}{m\Gamma(m/2)} \prod_{i=1}^m \lambda_i, \quad (4.9)$$

kjer  $\Gamma(\cdot)$  predstavlja funkcijo gama.

Vsak  $j$ -ti roj lahko tako predstavimo s povprečno vrednostjo vzorcev v roju, njihovo kovariančno matriko ter številom vzorcev v roju oz. kot trojček  $(\mu_j, \Sigma_j, n_j)$ . Primer predstavitve je predstavljen na sliki 4.1.



Slika 4.1: Prikaz predstavitve roja.

## 4.2 Rekurzivno računanje parametrov roja

Ko razvijamo samorazvijajočo se metodo primerno za tok podatkov, se moramo zavedati, da vseh prejšnjih vzorcev ne shranjujemo; tako imamo v trenutku  $k$  največkrat na voljo samo vzorec  $\mathbf{z}[k]$ , kar nam onemogoča da bi uporabili enačbi 4.5 in 4.6 za izračun središča roja  $\mu$  in njegove kovariančne matrike  $\Sigma$ . Slednji zato izračunamo na rekurziven način.

Središče in kovariančno matriko  $j$ -tega roja z  $n_j$  vzorci zapišimo kot  $\mu_j^{n_j}$  in  $\Sigma_j^{n_j}$ . Ko dodamo vzorec  $\mathbf{z}[k]$  v  $j$ -ti roj, se najprej lotimo posodabljanja parametrov z izračunom razlike med trenutnim vzorcem in središčem roja:

$$\mathbf{e}_j[k] = \mathbf{z}[k] - \mu_j^{n_j}. \quad (4.10)$$

Nato posodobimo središče roja:

$$\boldsymbol{\mu}_j^{n_j+1} = \boldsymbol{\mu}_j^{n_j} + \frac{1}{n_j + 1} \mathbf{e}_j[k]. \quad (4.11)$$

Zatem posodobimo nenormalizirano kovariančno matriko roja:

$$\mathbf{S}_j^{n_j+1} = \mathbf{S}_j^{n_j} + \mathbf{e}_j[k] \left( \mathbf{z}[k] - \boldsymbol{\mu}_j^{n_j+1} \right)^T. \quad (4.12)$$

Posodobljeno kovariančno matriko dobimo kot:

$$\boldsymbol{\Sigma}_j^{n_j+1} = \frac{1}{n_j} \mathbf{S}_j^{n_j+1}. \quad (4.13)$$

Ko je v roju samo vzorec  $\mathbf{z}[k]$ , ima slednji središče, kovariančno matriko in število vzorcev enako:

$$\boldsymbol{\mu} = \mathbf{z}[k], \quad \boldsymbol{\Sigma} = \mathbf{0}, \quad n = 1. \quad (4.14)$$

### 4.3 Samorazvijajoče se Gaussovo rojenje

Metoda poteka v dveh korakih. Ko prispe novi vzorec  $\mathbf{z}[k]$ , se najprej preveri, če je izpolnjen pogoj, da se vzorec doda k enemu od obstoječih rojev, ki so trenutno v sistemu, drugače se doda nov roj. V drugem koraku poteka združevanje rojev, v kolikor je pogoj za združevanje izpolnjen.

#### 4.3.1 Dodajanje rojev

Ko prispe nov vzorec, se sprva izračuna pripadnost vzorca do vseh obstoječih rojev:

$$\gamma_i[k] = e^{-d_i^2[k]}, \quad \text{za vsak } i \in \{1, \dots, c\}, \quad (4.15)$$

kjer  $c$  predstavlja število rojev, razdalja  $d_i^2[k]$  pa je odvisna od števila vzorcev  $n_i$  v  $i$ -tem roju:

$$d_i^2[k] = \begin{cases} (\mathbf{z}[k] - \boldsymbol{\mu}_i)^T (\mathbf{z}[k] - \boldsymbol{\mu}_i), & n_i < N_{max} \\ (\mathbf{z}[k] - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{z}[k] - \boldsymbol{\mu}_i), & n_i \geq N_{max} \end{cases}. \quad (4.16)$$

Če je število vzorcev manjše od parametra  $N_{max}$ , razdaljo računamo kot Evklidsko razdaljo zaradi numerične nestabilnosti inverza  $\Sigma_i^{-1}$ , drugače se pripadnost računa z Mahalanobisovo razdaljo. Če je največja pripadnost vzorca posameznemu roju enaka  $\gamma_j[k]$ :

$$\gamma_j[k] = \max_i \gamma_i, \text{ za vsak } i \in \{1, \dots, c\}, \quad (4.17)$$

$$j = \operatorname{argmax}_i \gamma_i \quad (4.18)$$

in velja:

$$\gamma_j[k] > \Gamma_{min}, \quad (4.19)$$

kjer  $\Gamma_{min}$  predstavlja prag za najmanjšo potrebno pripadnost, potem je prispeli vzorec pripisan roju  $j$ , ki se mu posodobijo parametri v skladu z enačbami 4.10 - 4.13. V kolikor je pripadnost manjša:

$$\gamma_j[k] \leq \Gamma_{min}, \quad (4.20)$$

se doda nov roj z enim vzorcem, kjer se parametri inicializirajo v skladu z enačbo 4.14. Psevdokoda Gaussovega rojenja je zapisana v sklopu algoritma 1.

### 4.3.2 Združevanje rojev

Potem ko se vzorec  $\mathbf{z}[k]$  pridruži obstoječemu roju ali so doda nov roj, sistem preveri za vse pare rojev ali je prišlo do takšne spremembe parametrov, ki ustrezajo pogojem za združevanje rojev. Dva roja,  $i$ -ti in  $j$ -ti roj, se združita na podlagi vrednosti  $\kappa_{ij}$ :

$$\kappa_{ij} = \frac{V_{ij}}{V_i + V_j}, \quad (4.21)$$

kjer je  $V_{ij}$  prostornina združenega roja,  $V_i$  prostornina  $i$ -tega roja in  $V_j$  prostornina  $j$ -tega roja, pri čemer so prostornine izračunane v skladu z enačbo 4.9. Vrednost  $\kappa$  predstavlja oceno prekrivanja dveh rojev, kjer se z upadanjem vrednosti, prekrivanje povečuje.

Da izračunamo prekrivanje, je potrebno prvo določiti parametre združenega roja. Število vzorcev v združenem roju in središče združenega roja dobimo kot:

$$n_{ij} = n_i + n_j, \quad (4.22)$$

---

**Algoritem 1** Psevdokoda Gaussovega rojenja
 

---

Parametri:  
 $\Gamma_{min}, N_{max}$   
 Inicializacija:  
 $c \leftarrow 1, \boldsymbol{\mu}_1 \leftarrow \mathbf{z}(1), \boldsymbol{\Sigma}_1 \leftarrow \mathbf{0}, k \leftarrow 1$   
 $\boldsymbol{\mu}_{global} \leftarrow \mathbf{z}(1), \boldsymbol{\Sigma}_{global} \leftarrow \mathbf{0}$

- 1: **repeat**
- 2:    $k \leftarrow k + 1$
- 3:   Posodbi  $\boldsymbol{\mu}_{global}$  in  $\boldsymbol{\Sigma}_{global}$  // enačbe 4.10 - 4.13
- 4:   **for**  $i = 1 : c$
- 5:     **if**  $n_i < N_{max}$  // enačba 4.16
- 6:        $d_i^2[k]$  je Evklidska razdalja
- 7:     **else**
- 8:        $d_i^2[k]$  je Mahalanobisova razdalja razdalja
- 9:     **end if**
- 10:    Izračunaj pripadnosti  $\gamma_i[k] = e^{-d_i^2[k]}$  // enačba 4.15
- 11:   **end for**
- 12:   Izberi roj z največjo pripadnostjo:  $j = \operatorname{argmax}_i \gamma_i$  // enačba 4.18
- 13:   **if**  $\gamma_j[k] > \Gamma_{min}$
- 14:     Pripiši  $z[k]$  roju  $j$  in posodobi parametre // enačbe 4.10 - 4.13
- 15:   **else**
- 16:     Dodaj in inicializiraj nov roj // enačba 4.14
- 17:      $c \leftarrow c + 1$
- 18:   **end if**
- 19:   Mehanizem za združevanje rojev
- 20: **until**  $k > N$

---

$$\boldsymbol{\mu}_{ij} = \frac{n_i \boldsymbol{\mu}_i + n_j \boldsymbol{\mu}_j}{n_{ij}}. \quad (4.23)$$

Kovariančno matriko združenega roja dobimo kot:

$$\boldsymbol{\Sigma}_{ij} = \frac{1}{n_{ij} - 1} (\mathbf{Z}_i^T \mathbf{Z}_i + \mathbf{Z}_j^T \mathbf{Z}_j - \mathbf{M}_{ij}^T \mathbf{E}_{ij}^T \mathbf{E}_{ij} \mathbf{M}_{ij}), \quad (4.24)$$

kjer

$$\mathbf{Z}_i^T \mathbf{Z}_i = (n_i - 1) \boldsymbol{\Sigma}_i + \mathbf{M}_i^T \mathbf{E}_i^T \mathbf{E}_i \mathbf{M}_i, \quad (4.25)$$

$$\mathbf{Z}_j^T \mathbf{Z}_j = (n_j - 1) \boldsymbol{\Sigma}_j + \mathbf{M}_j^T \mathbf{E}_j^T \mathbf{E}_j \mathbf{M}_j, \quad (4.26)$$

kjer  $\mathbf{M}_k$  predstavlja diagonalno matriko velikosti  $m \times m$ ,  $\mathbf{M}_k = \text{diag}(\boldsymbol{\mu}_k(1), \dots, \boldsymbol{\mu}_k(m))$ , in  $\mathbf{E}_k$  predstavlja matriko velikosti  $n_k \times m$ , kjer so vsi elementi enaki 1.

Roja se v delu [2] združita, v kolikor je prekrivanje dveh rojev  $\kappa_{ij}$  manjše od v naprej določenega parametra  $\kappa_{join}$ :

$$\kappa_{ij} < \kappa_{join}, \quad (4.27)$$

pri čemer najprej združimo roja, kjer je prekrivanje največje:

$$\kappa_{i^*,j^*} = \min_{i,j} \kappa_{i,j}, \text{ za vsak } i, j \in \{1, \dots, c\}, \text{ pri čemer } i \neq j, \quad (4.28)$$

kjer sta  $i^*$  in  $j^*$  enaka:

$$(i^*, j^*) = \underset{i,j}{\operatorname{argmin}} \kappa_{i,j}. \quad (4.29)$$

Izkazalo se je, da zgolj ob upoštevanju pogoja v enačbi 4.27, rojenje ni najbolj stabilno, saj prihaja do bistvenih sprememb v delovanju sistema pri manjših spremembah vhodnih parametrov sistema ali vzbujačnega signala. Slednje je bilo opazno zlasti pri sistemih z višje dimenzionalnim vhodom v obliki prepogostega združevanja rojev, ko so se roji, ko je njihovo število že konvergiralo k določeni vrednosti, nenadoma združili v bistveno večje roje. Zato smo mehanizmu za združevanje dodali dodaten pogoj, s katerim smo preprečili takšen tip združevanj:

$$(\kappa_{ij} < \kappa_{join}) \wedge \left( \frac{V_{ij}}{V_{global}} < v_{max} \right), \quad (4.30)$$

kjer je  $V_{global}$  prostornina vseh vzorcev, v kolikor bi bili združeni v samo en roj, in  $v_{max}$  relativna največja dopustna prostornina glede na vse vzorce.

Združevanje poteka toliko časa, dokler je pogoj za združevanje v enačbi 4.30 izpolnjen. Parametre združenega roja določimo v skladu z enačbami 4.22 - 4.24. Psevdokoda Gaussovega rojenja je zapisana v sklopu algoritma 2.

## 4.4 Identifikacija vhodno-izhodnega modela

Identifikacija vhodno-izhodnega modela poteka tako, da sprva določimo lokalne modele, katerih izhode nato združimo z mehko logiko. Ker uporabljamo mehki

---

**Algoritem 2** Združevanje rojev
 

---

Parametri:

 $\kappa_{join}, v_{max}$ 

```

1: while 1
2:   Izračunaj  $n_{ij}, \mu_{ij}, \Sigma_{ij} \forall i, j \in \{1, \dots, c\}, i \neq j$  // enačbe 4.22 - 4.24
3:   Izračunaj  $V_{ij} \forall i, j \in \{1, \dots, c\}, i \neq j$  // enačba 4.9
4:   Izračunaj prekrivanja  $\kappa_{ij}$  // enačba 4.21
5:   Izberi roja  $(i^*, j^*)$  z največjim prekrivanjem  $\kappa_{ij}$  // enačba 4.29
6:   if  $\kappa_{i^*, j^*} < \kappa_{join}$  and  $\frac{V_{ij}}{V_{global}} < v_{max}$ 
7:     Združi roja  $i^*, j^*$  // enačbe 4.22 - 4.24
8:      $c \leftarrow c - 1$ 
9:   else
10:    break // Končaj z združevanjem
11:   end if
12: end while

```

---

model Takagi-Sugeno prvega reda, bo izhod lokalnega modela določala linearna funkcija vhodov (enačba 3.2). Parametre lokalnega modela se lahko določi ali z metodo rekurzivnih najmanjših kvadratov ali z metodo regresije na osnovi glavnih komponent (ang. *principal component regression - PCR*) [32], ki jo bomo uporabili v tem delu. Opisana metoda je primerna zgolj za sisteme z enima izhodom.

#### 4.4.1 Določitev parametrov lokalnega modela

Metoda PCR je poseben primer metode totalnih najmanjših kvadratov (ang. *total least squares - TLS*) in se jo v literaturi imenuje tudi kot metoda ortogonalne regresije [34]. Od metode najmanjših kvadratov se razlikuje po tem, da ne dopušča zgolj možnosti napake na izhodu  $\tilde{\mathbf{y}} = \mathbf{y} + \Delta\mathbf{y}$ , ampak tudi napako na vходу  $\tilde{\mathbf{X}} + \Delta\mathbf{X}$ , kar pomeni, da se ne išče rešitve minimuma  $\|\mathbf{X}\boldsymbol{\omega} - \mathbf{y}\|_2$  v smeri osi  $\mathbf{y}$  med utežmi  $\boldsymbol{\omega}$ , kakor v primeru metode najmanjših kvadratov.

Metoda PCR izhaja iz metode glavnih komponent (ang. *principal component analysis - PCA*), ki temelji na obravnavi kovariančne matrike  $\Sigma_j$  vhodno-izhodnih

podatkov  $j$ -tega roja:

$$\mathbf{Z}_j = [\mathbf{z}_1^T, \dots, \mathbf{z}_{n_j}^T], \quad (4.31)$$

kjer je  $\mathbf{Z}_j$  sestavljena iz  $n_j$  vzorcev  $\mathbf{z}$ :

$$\mathbf{z}^T = [\mathbf{x}^T \ y]. \quad (4.32)$$

Vektor  $\mathbf{z}$  je dolžine  $m$  in je sestavljen iz vhodnega vektorja  $\mathbf{x}$  in skalarja  $y$ , ki predstavlja izhod.

Z metodo PCA dobimo rotacijo osi izhodiščnega koordinatnega sistema v nov ortogonalni sistem, ki rezultira v največji kovarianci podatkov glede na novo izbrane osi [5]. Lastni vektorji kovariančne matrike predstavljajo glavne komponente oz. osi novega ortogonalnega sistema, medtem ko pripadajoče lastne vrednosti predstavljajo varianco podatkov v smeri pripadajočega lastnega vektorja. Primerna numerična metoda za iskanje lastnih vrednosti in vektorjev je singularni razcep (enačba 4.7).

Ideja je, da lastni vektor  $\mathbf{z}$  z najmanjšo lastno vrednostjo  $\mathbf{p}_m^j$  definira hiper-ravnino, ki predstavlja vhodno-izhodni model prvega reda, pri čemer gre hiper-ravnina skozi središčno točko roja  $\boldsymbol{\mu}_j$ :

$$(\mathbf{z} - \boldsymbol{\mu}_j)^T \mathbf{p}_m^j = 0. \quad (4.33)$$

Minimizacija napake tako poteka v smeri  $\mathbf{p}_m^j$ , lastna vrednost  $\lambda_m^j$  pa predstavlja varianco napake:

$$\sigma_j^2 = \lambda_m^j. \quad (4.34)$$

Upoštevajoč enačbi 4.32 in 4.33 lahko dobimo eksplicitno enačbo izhoda lokalnega modela:

$$\tilde{y}_j = \boldsymbol{\mu}_j(m) + \delta y_j, \quad (4.35)$$

kjer je:

$$\delta y_j = \frac{(\boldsymbol{\mu}_j(1) - \mathbf{x}(1))\mathbf{p}_m^j(1) + \dots + (\boldsymbol{\mu}_j(m-1) - \mathbf{x}(m-1))\mathbf{p}_m^j(m-1)}{\mathbf{p}_m^j(m)}. \quad (4.36)$$

Metoda PCR je tako preprosta ter hitra in v primeru opisanega sistema zahteva minimalno količino dodatnih izračunov, saj samorazvijajoči se sistem izračuna glavne komponente že pri rojenju, ko je potreben singularni razcep za izračun inverza kovariančne matrike rojev (enačba 4.16).

#### 4.4.2 Določitev izhoda mehkega modela

Izhod mehkega modela lahko določimo v skladu z enačbo 3.4:

$$\tilde{y}[k] = \frac{\sum_{j=1}^c \gamma_j[k] \tilde{y}_j[k]}{\sum_{j=1}^c \gamma_j[k]}, \quad (4.37)$$

kjer  $y_j$  predstavlja izhod  $j$ -tega lokalnega roja iz enačbe 4.35 in  $\gamma_j[k]$  pripadnost vzorca  $\tilde{z}$  temu roju. Ker izhoda  $y$  v časovnem trenutku  $k$  ne poznamo, lahko pripadnosti  $\gamma_j[k]$  določimo z vektorjem:

$$\tilde{z}^T = [\mathbf{x} \quad \tilde{y}_j], \quad (4.38)$$

kjer za izračun pripadnosti roju, uporabimo kar vrednost modela  $\tilde{y}_j$   $j$ -tega roja. Pripadnosti tako izračunamo kot:

$$\gamma_j[k] = e^{-\frac{1}{2}(\tilde{z} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\tilde{z} - \boldsymbol{\mu}_j)}. \quad (4.39)$$

Ker so v  $\tilde{z}$  prisotni regresorji, ki niso direktno merjeni, lahko pride do napačnih izračunov pripadnosti glede na realno stanje. Da bi se temu ognili, lahko za izračun pripadnosti uporabimo zgolj regresorje, ki so direktno merjeni, pri čemer tudi kovariančno matriko  $\boldsymbol{\Sigma}$  v enačbi 4.39 sestavimo zgolj iz pripadajočih regresorjev.

#### 4.4.3 Modeliranje dinamičnih sistemov

V kolikor modeliramo dinamični sistem:

$$y = f(\mathbf{x}[k], k), \quad (4.40)$$

ki je  $p$ -tega reda in je izhod sistema odvisen od  $p$  prejšnjih stanj sistema  $y$  in vzbujačnega signala<sup>1</sup>  $u$ :

$$\mathbf{x}[k] = [u[k], u[k-1], \dots, u[k-p], y[k-1], \dots, y[k-p]], \quad (4.41)$$

predstavlja izhod mehkega modela  $\tilde{y}[k]$  iz enačbe 4.37 enokoračno napoved.

<sup>1</sup>Vzbujanje je lahko tudi večvhodno, vendar bomo zaradi preglednosti predpostavili, da imamo enovhodni sistem.

Če bi želeli dobiti večkoračno napoved, moramo neznane regresorje  $y$  v enačbi 4.41 zamenjati z napovedmi modela  $\tilde{y}$ . Za dvokoračno napoved bi bila zamenjava:

$$\mathbf{x}[k+1] = [u[k+1], u[k], \dots, u[k-p+1], \tilde{y}[k], y[k], \dots, y[k-p+1]]. \quad (4.42)$$

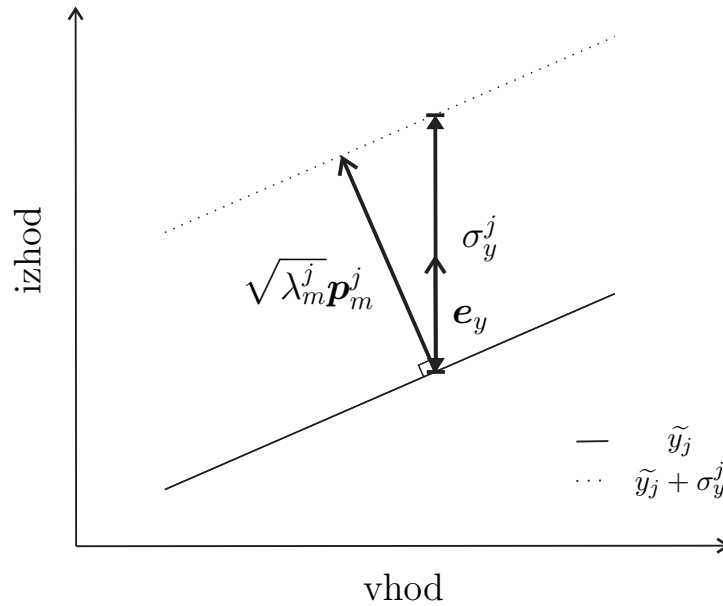
Do želene večkoračne napovedi tako pridemo z iteracijami, kjer v vsaki iteraciji zamenjamo neznan regresor  $y$  z napovedjo  $\tilde{y}$  iz prejšnje iteracije.

#### 4.4.4 Določitev intervala zaupanja

V skladu s PCR metodo in enačbama 4.33 ter 4.34 lahko interval zaupanja v smeri izhodne osi določimo s projekcijo vektorja  $\mathbf{p}_m^j$  na izhodno os, kot je prikazano na sliki 4.2. Slednje z enačbo zapišemo kot

$$\sigma_y^j = \frac{\sqrt{\lambda_m^j}}{\mathbf{p}_m^j \mathbf{e}_y}. \quad (4.43)$$

kjer je  $\lambda_m^j$  najmanjša lastna vrednost,  $\mathbf{p}_m^j$  pripadajoči lastni vektor in  $\mathbf{e}_y$  enotski vektor, ki kaže v smeri izhoda  $y$ .



Slika 4.2: Prikaz izračuna intervala zaupanja.

Varianco izhoda mehkega modela nato dobimo iz lokalnih modelov in pripadnosti vzorca kot:

$$\sigma_y^2 = \sum_{i=1}^c (\gamma_i \sigma_y^i)^2. \quad (4.44)$$

Če bi želeli imeti oceno za večkoračno napoved, bi slednjo lahko določili eksperimentalno z vzbujanjem v okolici središč rojev, s čimer bi dobili ocene za  $\sigma_y^j$ , ki bi jih uporabili v enačbi 4.44, vendar slednje ne poteka s sprotnim učenjem.

#### 4.4.5 Identifikacija s predhodnim filtriranjem

Rojenje dinamičnih sistemov lahko izvajamo tudi s predhodnim filtriranjem, kjer vhodno spremenljivko  $u$  in izhodno spremenljivko  $y$  filtriramo s filtrom prvega reda. Vzorec je tako sestavljen kot:

$$\mathbf{z}_f[k] = [u_f[k] \ y_f[k]]^T, \quad (4.45)$$

kjer  $u_f$  in  $y_f$  predstavljata filtrirano spremenljivko vhoda in izhoda. Digitalni filter prvega reda dobimo z diskretizacijo zveznega sistema  $G_z(s)$ :

$$G_z(s) = \frac{1}{T_f s + 1}, \quad (4.46)$$

kjer  $T_f$  predstavlja časovno konstanto filtra, ki jo določimo glede na mejno frekvenco opazovanega sistema. Diskretizacijo lahko izvedemo s Tustinovim pravilom:

$$H(z) = G_z(s)|_{s=\frac{2}{T_s} \frac{z-1}{z+1}}, \quad (4.47)$$

kjer  $T_s$  predstavlja vzorčni čas. Tako dobimo filtrirana signala  $u_f$  in  $y_f$  kot:

$$\begin{aligned} y_f[k] &= \frac{T_s}{2T_f + T_s} \left( y[k] + y[k-1] - y_f[k-1] \left( 1 - \frac{2T_f}{T_s} \right) \right), \\ u_f[k] &= \frac{T_s}{2T_f + T_s} \left( u[k] + u[k-1] - u_f[k-1] \left( 1 - \frac{2T_f}{T_s} \right) \right). \end{aligned} \quad (4.48)$$

Rojenje poteka v dvodimenzionalnem filtriranem prostoru, ki predstavlja aproksimacijo statične karakteristike procesa.

Hkrati z rojenjem se formirajo tudi roji v nefiltriranem prostoru, ki je sestavljen iz vzorcev, kot so zapisani v enačbi 4.32, pri čemer je vzorec  $\mathbf{z}[k]$  razporejen

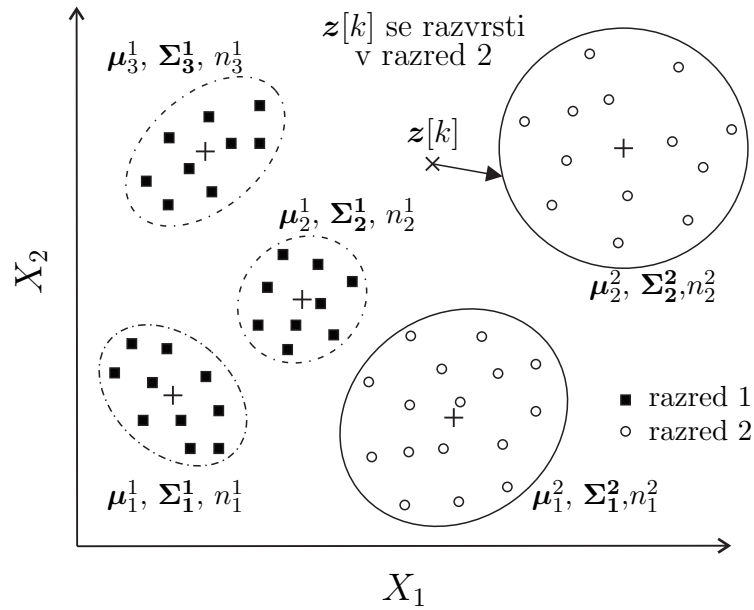
v roje glede na pripadnost vzorca  $z_f[k]$ . V nefiltriranem prostoru poteka nato računanje mehkega modela z metodo PCR.

## 4.5 Nadzorovano razvrščanje

S predlagano metodo lahko razvrščamo vzorce na nadzorovan način, kar pomeni, da je razred pri učenju znan. Tako rojenje poteka ločeno za vsak razred. Vzorec  $z[k]$  razvrstimo v tisti razred, kamor spada roj, pri katerem ima vzorec največjo pripadnost:

$$k = \underset{i,j}{\operatorname{argmax}} \gamma_i^j, \text{ za vsak } j \in \{1, \dots, r\}, i \in \{1, \dots, c_j\}, \quad (4.49)$$

kjer je  $r$  število razredov,  $c_j$  število rojev  $j$ -tega razreda in  $\gamma_i^j$  pripadnost vzorca  $i$ -temu roju  $j$ -tega razreda. Primer razvrščanja je prikazan na sliki 4.3.



Slika 4.3: Prikaz nadzorovanega razvrščanja.



## 5 Eksperimenti in rezultati

V tem sklopu bomo predstavili eksperimente ter rezultate, ki smo jih dobili z metodo opisano v poglavju 4. Metodo smo preizkusili za potrebe rojenja, identifikacije vhodno-izhodnega modela ter razvrščanja, pri čemer smo uporabili različen nabor podatkov.

### 5.1 Rojenje z dvodimenzionalnimi podatki

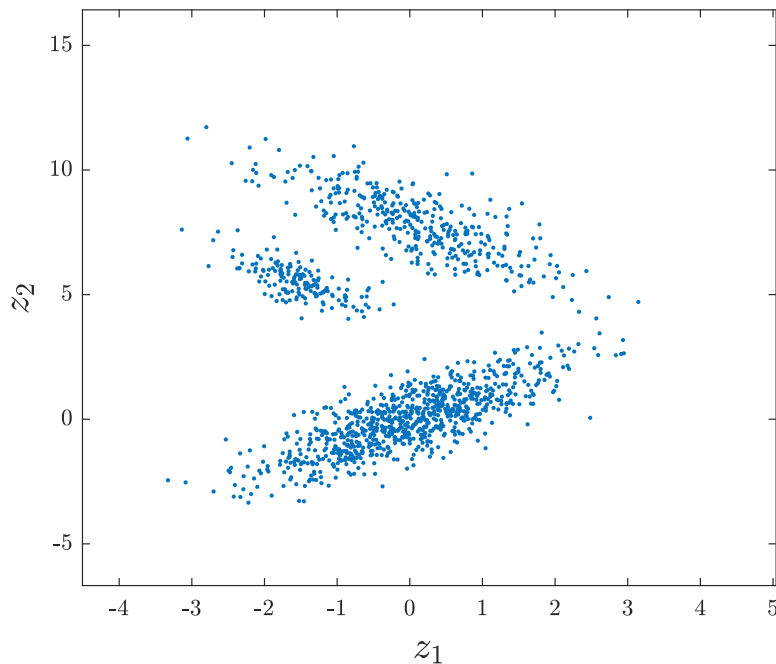
Rojenje smo izvedli na dvodimenzionalnem umetnem sistemu. S tem enostavnim primerom želimo pokazati delovanje metode. Z različnimi izbirami vhodnih parametrov  $\Gamma_{min}$ ,  $\kappa_{join}$  in  $v_{max}$  smo analizirali, kako posamezni parametri vplivajo na samorazvijajoči se sistem.

#### 5.1.1 Predstavitev podatkov

Dvodimenzionalne podatke smo ustvarili z naključnim generatorjem, ki generira naključne vrednosti v skladu z Gaussovo porazdelitvijo  $\mathcal{N}(\mu, \sigma)$ . Ustvarili smo tri procese: prvi proces  $\mathbf{Z}_1$  je generiran kot  $z_{11} = \mathcal{N}(0, 1)$  in  $z_{12} = z_{11} + \mathcal{N}(0, 0.8)$ , drugi proces  $\mathbf{Z}_2$  je generiran kot  $z_{21} = \mathcal{N}(0, 1)$  in  $z_{22} = -z_{21} + \mathcal{N}(8, 0.85)$  in tretji proces  $\mathbf{Z}_3$  je generiran kot  $z_{31} = \mathcal{N}(-1.5, 0.5)$  in  $z_{32} = -z_{31} + \mathcal{N}(4, 0.5)$ . Delež  $\mathbf{Z}_1$  je  $\frac{5}{8}$ , delež  $\mathbf{Z}_2$  je  $\frac{2}{8}$  in delež  $\mathbf{Z}_3$  je  $\frac{1}{8}$ . Dvodimenzionalni prostor je prikazan na sliki 5.1. Vrstni red prihajanja vzorcev je naključen glede na proces in njegov delež, kar je prikazano na sliki 5.2.

Vzorec  $\mathbf{z}$  je v časovnem trenutku  $k$  definiran kot:

$$\mathbf{z}[k] = [z_1[k], z_2[k]]. \quad (5.1)$$

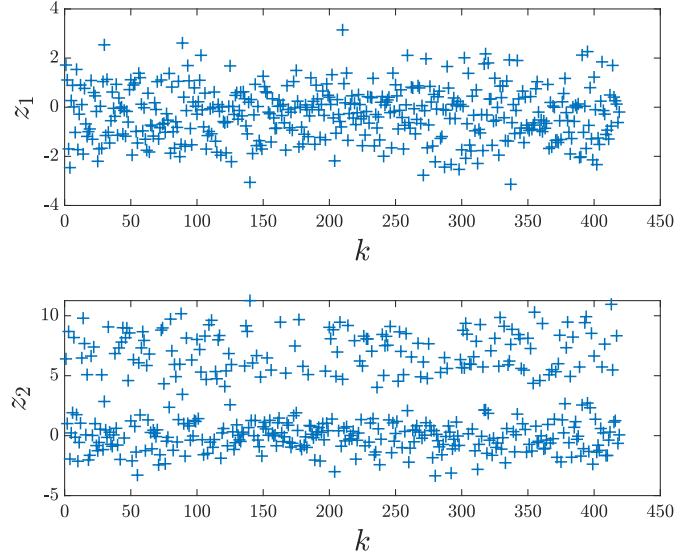


Slika 5.1: Prostor  $Z$ .

### 5.1.2 Rezultati

Naredili smo več poskusov z različnimi vhodnimi parametri  $\Gamma_{min}$ ,  $\kappa_{join}$  in  $v_{max}$ . Vsak poskus smo ponovili 100-krat z nanovo generiranimi podatki. Vsi uporabljeni parametri so vidni iz tabele 5.1. Rojenje smo prekinili, ko po 1000-ih vzorcih ni prišlo do spremembe števila rojev. Parametra  $N_{max}$ , ki služi numerični stabilnosti sistema, nismo spreminjali in je bil konstanten pri vseh poskusih  $N_{max} = 7$ .

V tabeli 5.1 so prikazani rezultati rojenja. Hitrost rojenja smo merili s parametrom  $k_s$ , ki predstavlja čas umiritve rojenja, pri čemer gre za čas ob zadnji spremembi števila rojev; umiritev je stanje, ko se število rojev ne spremeni več kot 1000 iteracij. Granulacijo prostora  $c_{max}$  smo merili z najvišjo vrednostjo števila rojev med rojenjem. Indikator kvalitete rojenja je končno število rojev  $c_s$ . V tabeli so podane povprečne vrednosti posameznih metrik in standardni odklon.

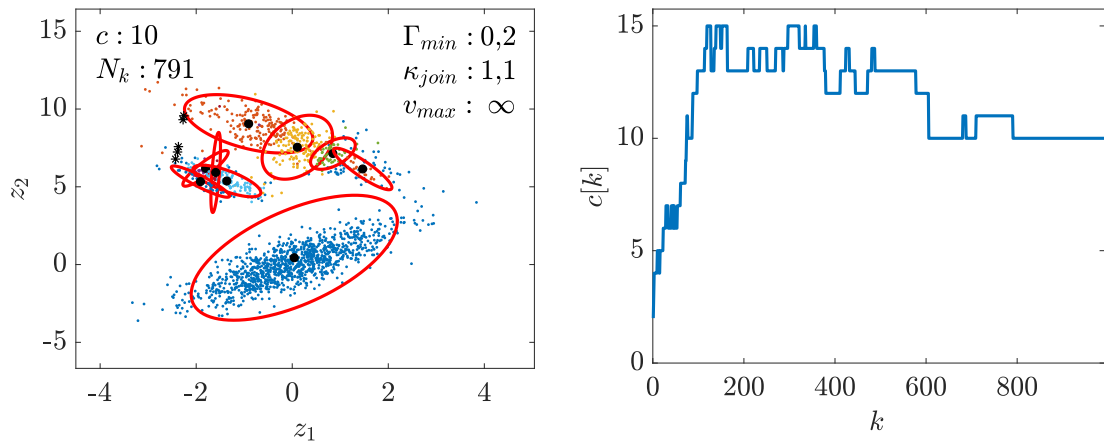
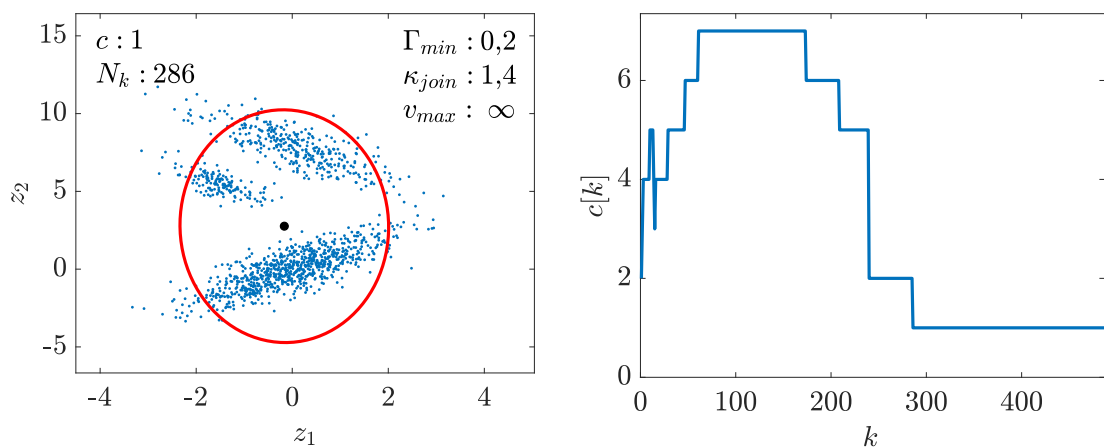
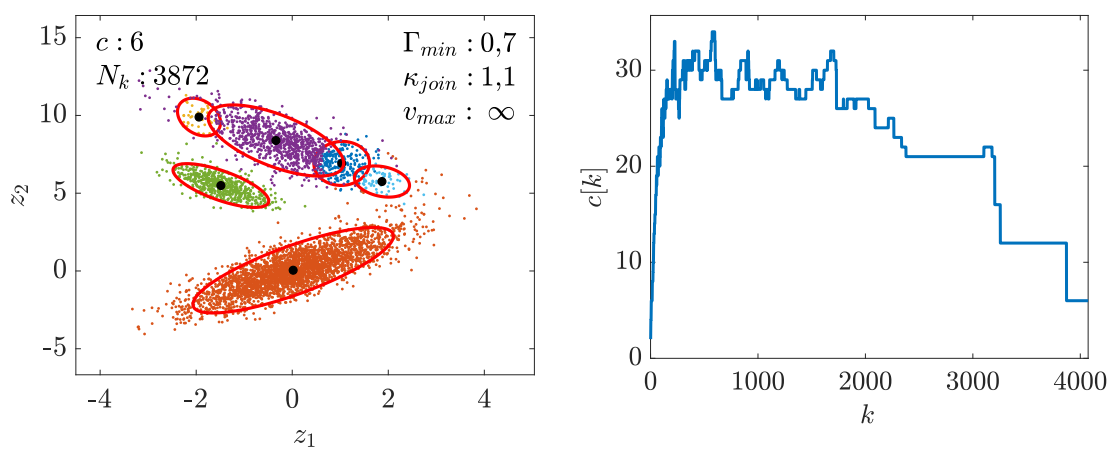


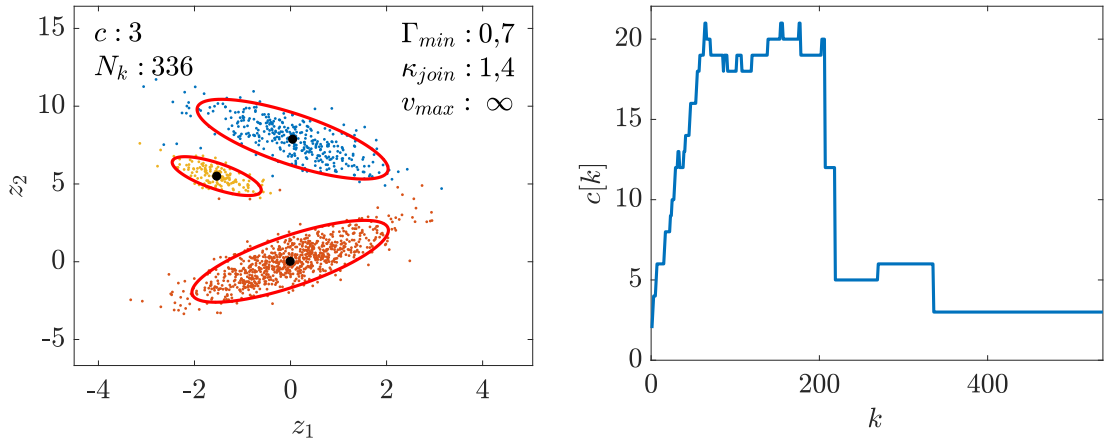
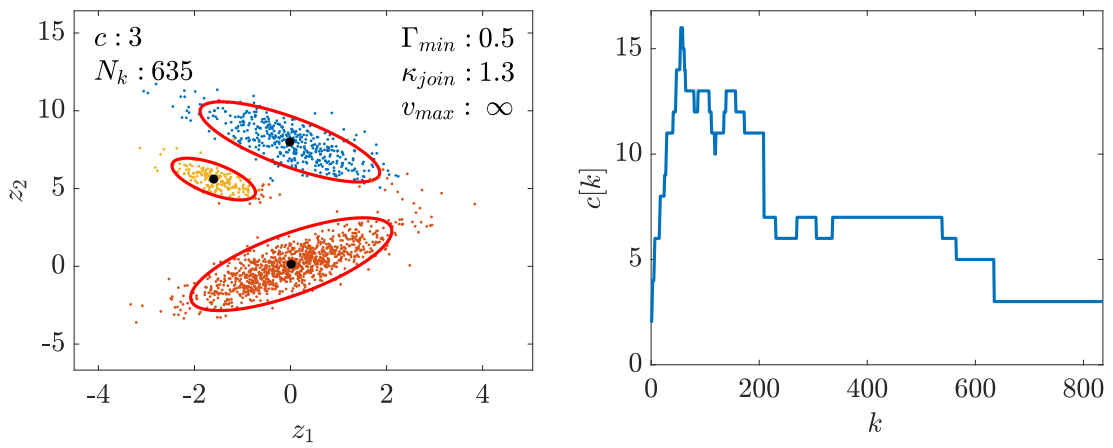
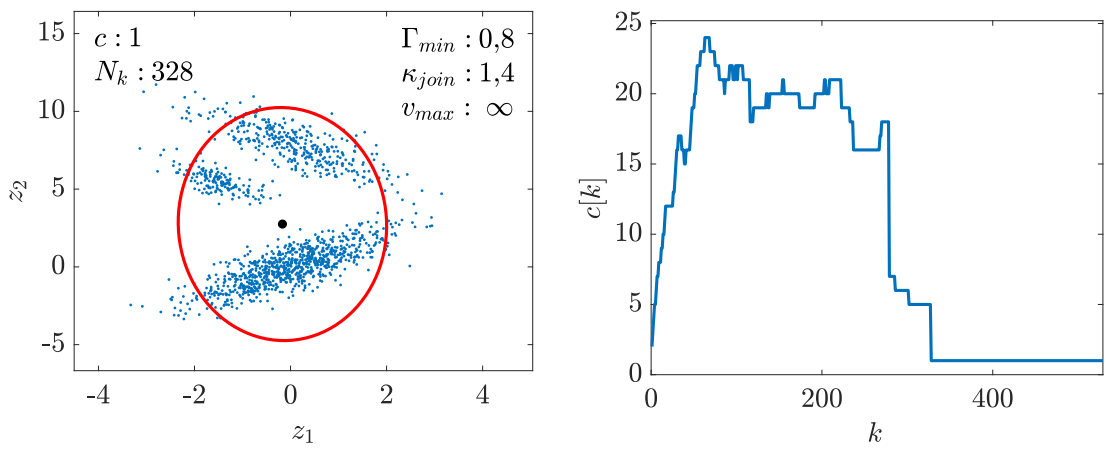
Slika 5.2: Prikaz prihajanja vzorcev.

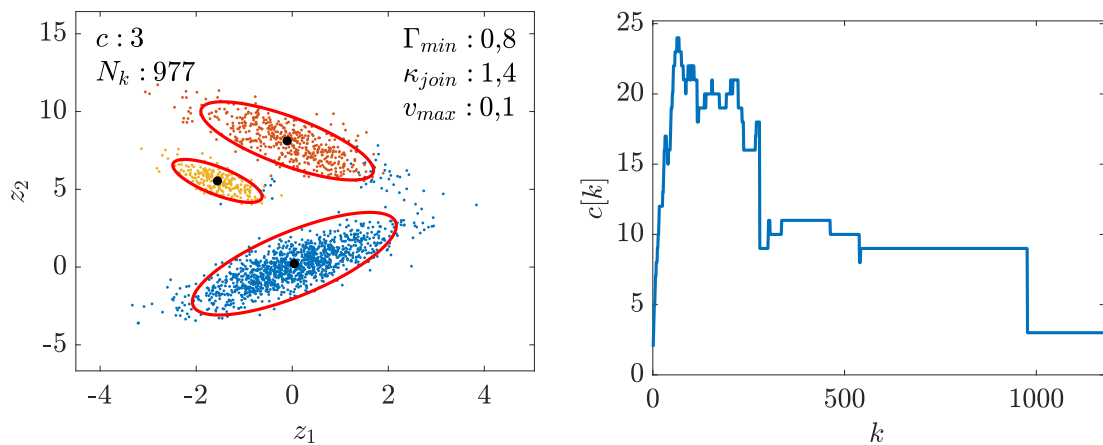
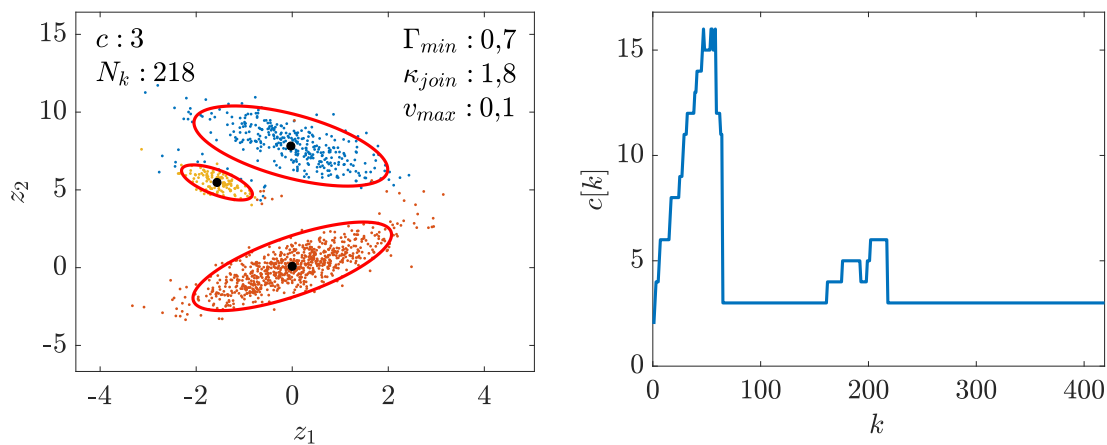
Kvaliteto rojenja lahko ocenimo tudi iz slike 5.3, kjer je prikazana tudi časovna odvisnost spreminjanja števila rojev  $c[k]$ . Za vsak poskus, ki je viden na sliki 5.3, so bili uporabljeni enaki podatki. Na slikah so vidne tudi vrednosti opazovanih metrik.

Iz simulacij rojenja z različnimi parametri ugotavljamo, da je opisano rojenje kompleksen proces, kjer stežka z zagotovostjo napovemo, kakšen bo odziv sistema na spremembo posameznega vhodnega parametra, vseeno pa lahko podamo neke osnovne relacije, kako parametri vplivajo na sistem.

Iz tabele 5.1 in slike 5.3, kjer je prikazana časovno odvisnost števila rojev  $c[k]$  med rojenjem, vidimo, da  $\Gamma_{min}$  vpliva predvsem v začetni fazi rojenja. V primeru, da je izbira parametra premajhna, pride do združevanja vzorcev v roje, ki glede na ustvarjene podatke, ne spadajo v enak roj, kar pokvari kasnejše rojenje (poskus b). Ugotavljamo, da je bolje izbrati višji  $\Gamma_{min}$ , s čimer dobimo večjo granulacijo prostora v začetni fazi rojenja. Iz tabele 5.1 vidimo, da manjši kot je  $\Gamma_{min}$ , manjše je največje število rojev med rojenjem  $c_{max}$  (poskusa a in b). Parameter  $\Gamma_{min}$  ne vpliva bistveno v zadnji fazi rojenja, kar lahko sklepamo iz časa umiritve  $k_s$ , saj ne zaznavamo bistvene odvisnosti od parametra  $\Gamma_{min}$ . Ugotavljamo tudi, da končno število rojev  $c_s$  ni bistveno odvisno od izbire  $\Gamma_{min}$ .

(a)  $\Gamma_{min} : 0, 2, \kappa_{join} : 1, 1, v_{max} : \infty$ .(b)  $\Gamma_{min} : 0, 2, \kappa_{join} : 1, 4, v_{max} : \infty$ .(c)  $\Gamma_{min} : 0, 7, \kappa_{join} : 1, 1, v_{max} : \infty$ .

(d)  $\Gamma_{min} : 0,7$ ,  $\kappa_{join} : 1,4$ ,  $v_{max} : \infty$ .(e)  $\Gamma_{min} : 0,5$ ,  $\kappa_{join} : 1,3$ ,  $v_{max} : \infty$ .(f)  $\Gamma_{min} : 0,8$ ,  $\kappa_{join} : 1,4$ ,  $v_{max} : \infty$ .

(g)  $\Gamma_{min} : 0,8$ ,  $\kappa_{join} : 1,4$ ,  $v_{max} : 0,1$ .(h)  $\Gamma_{min} : 0,7$ ,  $\kappa_{join} : 1,8$ ,  $v_{max} : 0,1$ .

Slika 5.3: Levo - vizualizacija rojenja umetnih dvodimenzionalnih podatkov ob koncu rojenja; prikazana so središča rojev (polni črni krogi) ter njihov meja rojev z oddaljenostjo  $2\sigma$ . Središča rojev, ki ne ustrezajo pogoju  $N_{max}$  za numerično stabilnost, so prikazana z \*. Vzorci, ki pripadajo istemu roju, so obarvani z enako barvo. Desno - časovna odvisnost spreminjanja števila rojev  $c[k]$ .

Tabela 5.1: Rezultati rojenja dvodimenzionalnih umetnih podatkov z vhodnimi parametri. Prikazani čas umiritve  $k_s$ , število rojev ob umiritvi  $c_s$  in največje število rojev med rojenjem  $c_{max}$ . Vsak poskus je bil ponovljen 100-krat.

| Poskus | $\Gamma_{min}$ | $\kappa_{join}$ | $v_{max}$ | $N_{max}$ | $c_s$ | $\sigma_{c_s}$ | $c_{max}$ | $\sigma_{c_{max}}$ | $k_s$ | $\sigma_{k_s}$ |
|--------|----------------|-----------------|-----------|-----------|-------|----------------|-----------|--------------------|-------|----------------|
| (a)    | 0,2            | 1,1             | inf       | 7         | 6,49  | 2,20           | 16,10     | 3,44               | 1630  | 750            |
| (b)    | 0,2            | 1,4             | inf       | 7         | 2,62  | 0,89           | 9,46      | 2,11               | 320   | 300            |
| (c)    | 0,7            | 1,1             | inf       | 7         | 11,46 | 6,35           | 39,88     | 4,90               | 3880  | 1340           |
| (d)    | 0,7            | 1,4             | inf       | 7         | 1,95  | 1,11           | 20,72     | 4,59               | 440   | 370            |
| (e)    | 0,5            | 1,3             | inf       | 7         | 3,27  | 1,65           | 16,38     | 3,09               | 640   | 390            |
| (f)    | 0,8            | 1,4             | inf       | 7         | 1,36  | 0,84           | 28,50     | 5,15               | 320   | 250            |
| (g)    | 0,8            | 1,4             | 0,1       | 7         | 3,71  | 1,37           | 27,78     | 6,04               | 1330  | 1360           |
| (h)    | 0,7            | 1,8             | 0,1       | 7         | 3,07  | 0,29           | 15,01     | 0,56               | 450   | 530            |

Parameter  $\kappa_{join}$  bistveno vpliva na največje število rojev med rojenjem  $c_{max}$ , ki je višji ob nižjem  $\kappa_{join}$  (poskus c). Vidimo tudi, da rojenje počasneje konvergira h končnemu številu rojev pri nižjem  $\kappa_{join}$  (poskusa a in c). Pri nižjem  $\kappa_{join}$  je končno število rojev načeloma višje (poskusa a in c). Parameter ne sme imeti prevelikega vpliva v prvi fazi rojenja (poskus f), ko je vzorcev malo. Vpliv parametra  $\kappa_{join}$  zmanjšamo, če zmanjšamo  $\Gamma_{min}$ .

Kljub temu, da so podatki, kjer je potekalo rojenje, skoraj idealni, saj so ustvarjeni z normalno distribucijo, kar rezultira v elipsoidnih rojih, je postopek rojenja težko popolnoma kontrolirati zgolj s parametroma  $\Gamma_{min}$  in  $\kappa_{join}$ , ki se uporabljata v delu [2], kar je razvidno tudi na podlagi standardnih odklonov metrik, ki so prikazani v tabeli 5.1. S poskusi b, d in e lahko dosežemo željen rezultat, da bi bilo končno število rojev enako tri, vendar je ob visokem standardnem odklonu  $\sigma_{c_s}$  slednje težko ponovljivo.

Da dosežemo boljšo kontrolo nad rojenjem, smo dodali parameter  $v_{max}$ , ki omogoča boljšo kontrolo predvsem v zadnji fazi rojenja, saj preprečuje, da bi se vzorci združili v prevelike roje. Slednje lahko vidimo s primerjavo poskusov f in g, kjer sicer dosežemo podoben vrh števila rojev  $c_{max}$ , končno število rojev je pa v poskusu g višje in bolj ustrezno. Z ustrezno izbiro parametrov lahko tako dosežemo hitro konvergenco števila rojev in preprečimo, da število rojev ne naraste preveč. Z  $v_{max}$  dobimo daleč najboljšo ponovljivost rojenja glede na

končno število rojev. To se zgodi v poskusu h, kjer število rojev v povprečju naraste na 15, rojenje pa se umiri pri treh rojih po 450. vzorcu.

Iz tabele 5.1 opazimo, da je ponovljivost rojenja z uporabo parametra  $v_{max}$  vidna iz standardnega odklona končnega števila rojev  $c_s$  in največjega števila rojev  $c_{max}$ . Nasprotno sicer velja za čas umiritve  $k_s$ , kjer je negotovost večja, kar je posledica manjših novonastalih lokalnih rojev v zadnji fazi rojenja, ko so glavni roji že izoblikovani. Novonastali roji se sicer hitro združijo z enim izmed večjih rojev in ne vplivajo bistveno na kvaliteto rojenja. Takšen primer dinamike rojenja je viden iz grafa časovne odvisnosti spreminjanja rojev  $c[k]$  na sliki 5.3h.

## 5.2 Identifikacija Hammersteinovega modela

Identifikacijo vhodno-izhodnega modela smo izvedli na Hammersteinovem modelu. Gre za umeten in izrazito nelinearen sistem prvega reda, kjer je potrebno rojenje v trodimenzionalnem prostoru. Sistem smo tudi linearizirali in primerjali lokalne modele z lineariziranimi. Identifikacijo smo izvedli tudi na modelu z dodanim šumom na izhodu in rezultate validirali. Identifikacijo smo ovrednotili tudi za primer, kjer smo pripadnosti (enačba 4.39) določili zgolj na podlagi vrednosti vhoda. Identifikacijo vhodno-izhodnega modela smo nato izvedli še s predhodnim filtriranjem opisanem v poglavju 4.4.5.

### 5.2.1 Predstavitev sistema

Izhod Hammersteinovega modela je podan z diferenčno enačbo:

$$y[k] = \frac{1}{10} \arctan(u[k-1]) + \frac{9}{10} (y[k-1]), \quad (5.2)$$

torej je izhod funkcija dveh spremenljivk:  $y[k] = f(u[k-1], y[k-1])$ . Izhod Hammersteinovega modela z dodanim Gaussovim šumom zapišemo kot:

$$y[k] = \frac{1}{10} \arctan(u[k-1]) + \frac{9}{10} (y[k-1]) + n[k], \quad (5.3)$$

pri čemer je  $n[k] \in \mathcal{N}(0, \sigma_n)$ .

Lineariziran model v delovni točki  $(u_0, y_0)$  zapišemo z enačbo:

$$\Delta y[k] = b_1 \Delta u[k-1] + a_1 \Delta y[k-1], \quad (5.4)$$

kjer sta  $\Delta u[k] = u[k] - u_0$  in  $\Delta y[k] = y[k] - y_0$  odmika od delovne točke. Parametra  $a_1$  in  $b_1$  dobimo s parcialnim odvajanjem enačbe 5.2:

$$\begin{aligned} b_1 &= \left. \frac{\partial f(u[k-1], y[k-1])}{\partial u[k-1]} \right|_{(u_0, y_0)} = \frac{\frac{1}{10}}{1 + u_0^2}, \\ a_1 &= \left. \frac{\partial f(u[k-1], y[k-1])}{\partial y[k-1]} \right|_{(u_0, y_0)} = \frac{9}{10}. \end{aligned} \quad (5.5)$$

Glede na to da gre za sistem prvega reda, smo regresorski vektor sestavili kot:

$$\mathbf{z}[k] = [u[k-1], y[k-1], y[k]]. \quad (5.6)$$

Vizualizacija regresorskega prostora s statično karakteristiko je prikazana na sliki 5.4.

### 5.2.2 Rezultati

V tem razdelku bomo prikazali rezultate identifikacije vhodno-izhodnega modela za Hammersteinov model brez dodanega šuma in z dodanim šumom. Pri izračunu pripadnosti pri določitvi izhoda modela (enačba 4.39) je bilo uporabljeno različno število regresorjev. *Model a* sestavljajo regresorji:

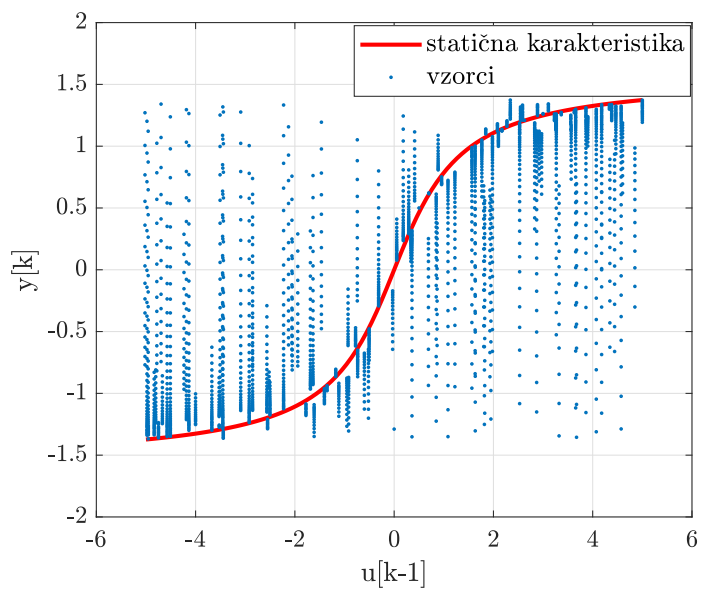
$$\tilde{\mathbf{z}}^T = [u[k-1], y[k-1], \tilde{y}[k]], \quad (5.7)$$

kjer je  $\tilde{y}[k]$  izhod modela v trenutku  $k$ . Pri *modelu b* je regresorski vektor sestavljen zgolj iz vrednosti vhoda:

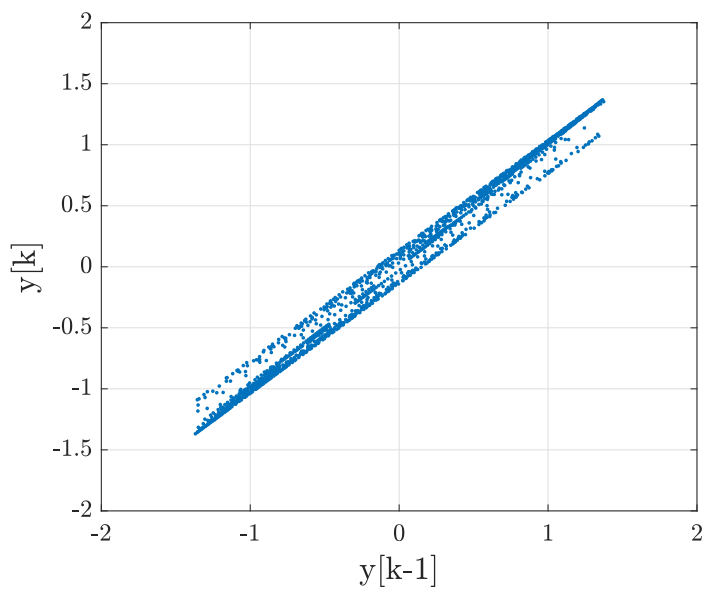
$$\tilde{\mathbf{z}}^T = [u[k-1]]. \quad (5.8)$$

Sistem smo vzbujali z amplitudno moduliranim psevdonaključnim binarnim signalom (APRBS) na intervalu  $[-5, 5]$ . Število vseh vzorcev je bilo 20030. Izsek signala je prikazan na sliki 5.5. Z vzbujalnim signalom APRBS smo dosegli bolj poln regresorski prostor in široko frekvenčno pasovno širino.

Do primernih parametrov smo prišli na podlagi znanja iz poglavja 5.1.2 in s poskušanjem. V tabeli 5.2 so prikazani uporabljeni vhodni parametri. Dodani



(a) Prostor  $u[k-1]-y[k-1]$  s statično karakteristiko.



(b) Prostor  $y[k-1]-y[k]$

Slika 5.4: Prikaz prostora  $u[k-1]-y[k]$  s statično karakteristiko (a) in prostora  $y[k-1]-y[k]$  (b).



Tabela 5.3: Primerjava parametrov  $b_1$  in  $a_1$  lokalnega identifikacijskega (indeks  $m$ ) modela z lineariziranim modelom (indeks  $l$ );  $\mu(1)$  je središče roja glede na vhod  $u$ .

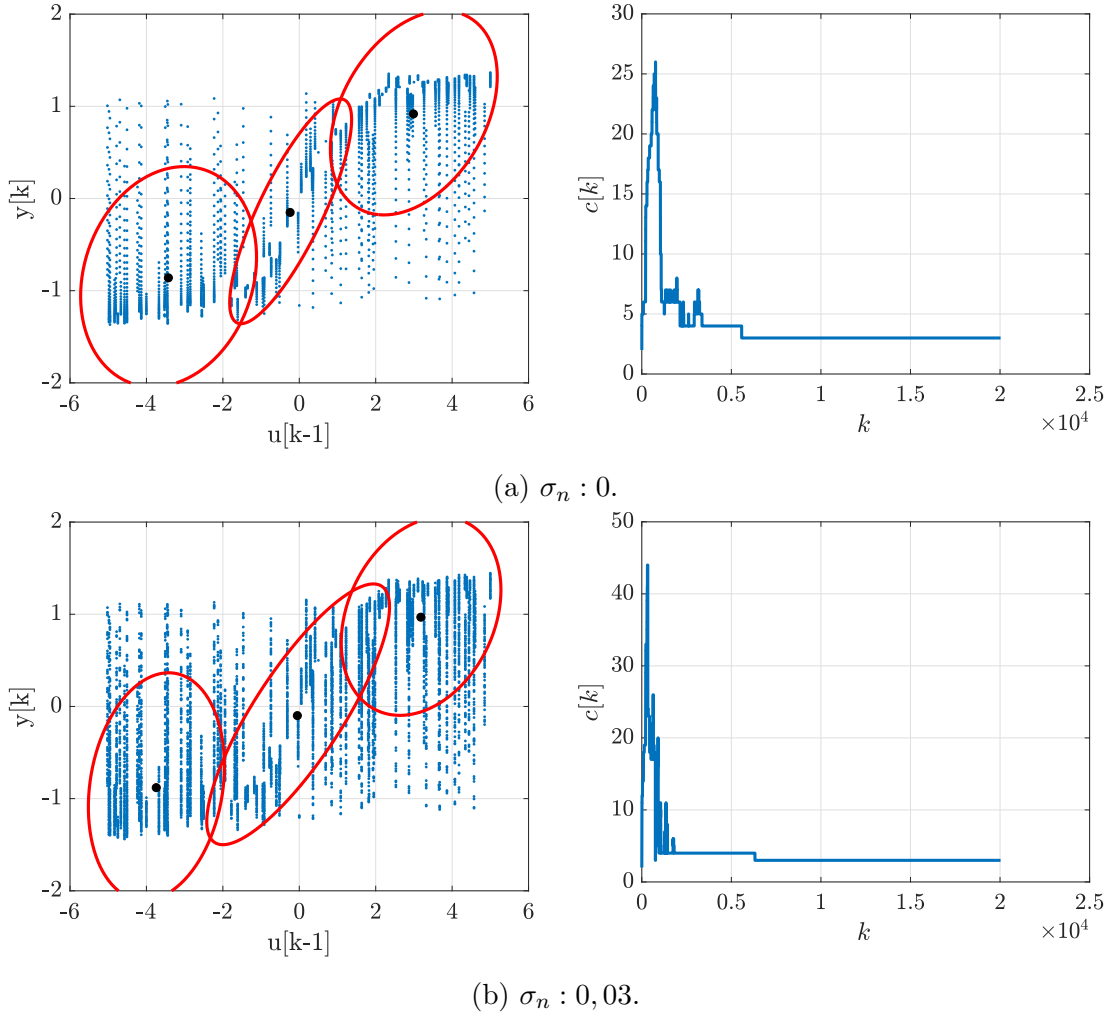
| $\sigma_n$ | $\mu(1)$ | $b_1^l$ | $b_1^m$ | $a_1^l$ | $a_1^m$ |
|------------|----------|---------|---------|---------|---------|
| 0          | -3,4202  | 0,0079  | 0,0118  | 0,9000  | 0,9008  |
|            | -0,2256  | 0,0952  | 0,0534  | 0,9000  | 0,9246  |
|            | 2,9973   | 0,0100  | 0,0129  | 0,9000  | 0,9042  |
| 0,03       | -3,7397  | 0,0067  | 0,0101  | 0,9000  | 0,8999  |
|            | -0,0297  | 0,0999  | 0,0664  | 0,9000  | 0,9124  |
|            | 3,1877   | 0,0090  | 0,0170  | 0,9000  | 0,9002  |

Tabela 5.4: Napaka  $RMSE$  za modela brez šuma in modela s šumom.  $RMSE$  računa napako med modelom in sistemom brez dodanega šuma.  $RMSE_n$  računa napako med modelom in sistemom z dodanim šumom.

| model | $\sigma_n$ | metrika  | simulacija |        | 1-koračna |        | 10-koračna |        |
|-------|------------|----------|------------|--------|-----------|--------|------------|--------|
|       |            |          | učna       | testna | učna      | testna | učna       | testna |
| (a)   | 0          | $RMSE$   | 0,0614     | 0,0667 | 0,0066    | 0,0061 | 0,0421     | 0,0453 |
|       | 0,03       | $RMSE$   | 0,0583     | 0,0968 | 0,0277    | 0,0287 | 0,0431     | 0,0651 |
|       |            | $RMSE_n$ | 0,0657     | 0,1008 | 0,0405    | 0,0416 | 0,0527     | 0,0712 |
| (b)   | 0          | $RMSE$   | 0,0544     | 0,0666 | 0,0063    | 0,0061 | 0,0391     | 0,0453 |
|       | 0,03       | $RMSE$   | 0,0592     | 0,0953 | 0,0281    | 0,0285 | 0,0438     | 0,0645 |
|       |            | $RMSE_n$ | 0,0664     | 0,0994 | 0,0414    | 0,0412 | 0,0530     | 0,0708 |

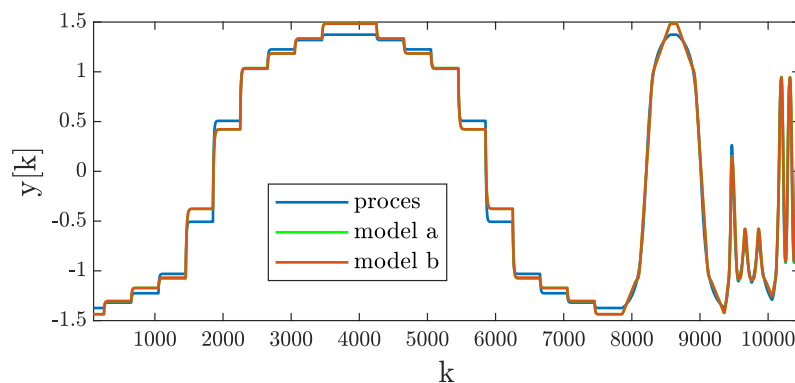
Parameter  $b_1$  ni konstanten, kar privede do večjih relativnih odstopanj kot pri parametru  $a_1$ . Zopet ugotovimo, da šum ne vpliva na kvaliteto rojenja in določitve parametrov lokalnih modelov.

Model a in model b smo ovrednotili z učnim signalom (slika 5.5) in testnim signalom (slika 5.7c). Vrednotenje je potekalo za primer simulacije, enokoračne napovedi in desetkoračne napovedi. Napako  $RMSE$  smo prikazali v tabeli 5.4. V primeru modela s šumom smo je izračunali glede na sistem, kjer je dodan šum ( $RMSE_n$ ) in kjer je šum odvzet ( $RMSE$ ). Kvalitativna ocena simulacije je vidna na sliki 5.7, izsek kvalitativne ocene enokoračne napovedi je viden na sliki 5.8 in izsek kvalitativne ocene desetkoračne napovedi je viden na sliki 5.9.

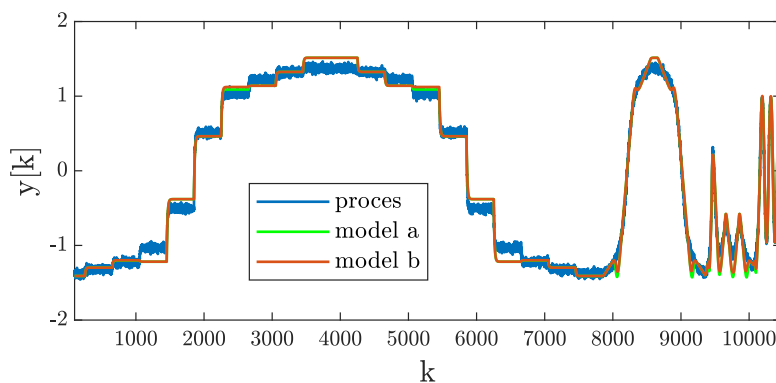


Slika 5.6: Prikaz rojev z robom  $2\sigma$  ob koncu rojenja (levo) in časovna odvisnost spreminjanja števila rojev  $c[k]$  (desno) za model brez šuma (a) in s šumom (b).

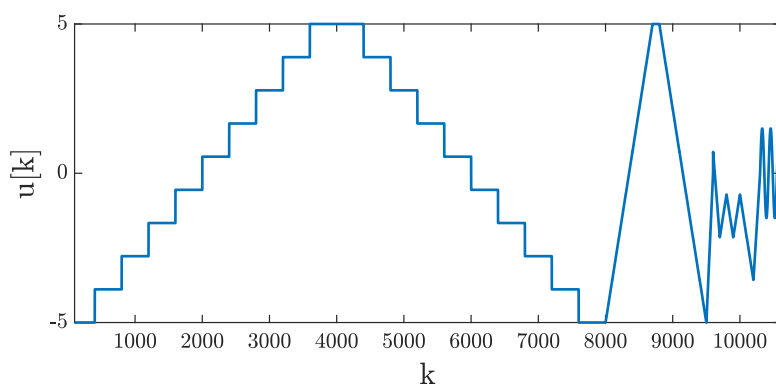
Iz simulacijske slike 5.7 vidimo, da sta model a in model b primerljiva za primer brez šuma in za primer z dodanim šumom, kar je razvidno tudi iz napake *RMSE* (tabela 5.4). Pojavlja se sicer napaka v stopničastem delu, kjer so odstopanja v ojačanju med modeloma in sistemom. Opazimo, da je v primeru modela s šumom več oscilacij, ki se pojavljajo na mejah med roji, pri čemer so oscilacije bolj izrazite pri modelu a, ki uporablja pošumljene regresorje pri določitvi pripadnosti. Napaka *RMSE* iz tabele 5.4 je sprejemljiva za primer simulacije. V primeru modela brez šuma se ne razlikuje bistveno med učno in testno množico podatkov. Zanimivo je, da na učni množici model z dodanim šumom dosega nižji *RMSE*;



(a) Simulacija za modela brez šuma:  $\sigma_n = 0$ .

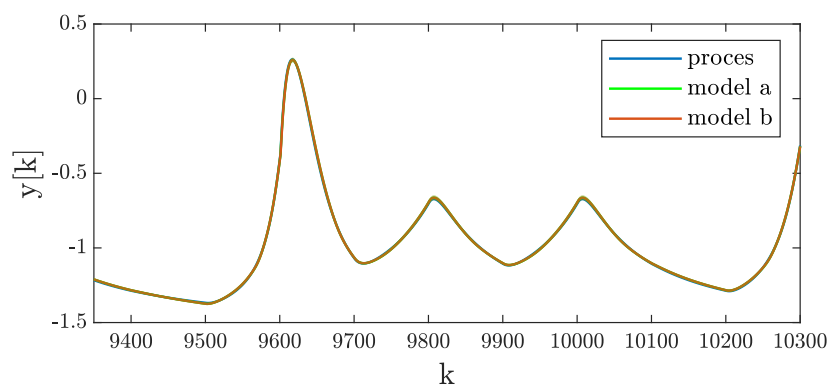


(b) Simulacija za modela s šumom:  $\sigma_n = 0,03$ .

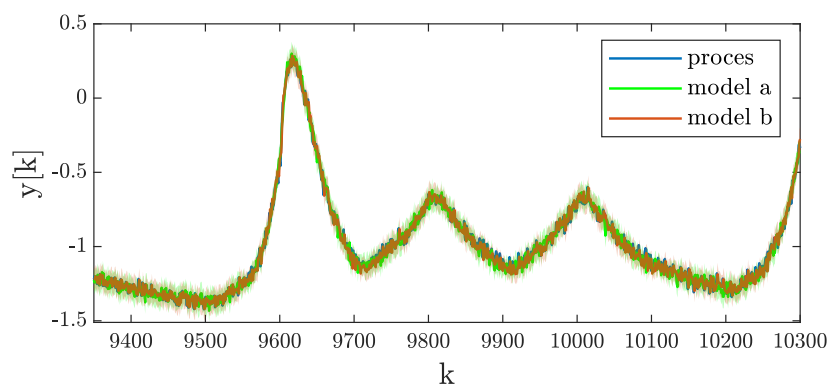


(c) Testni signal za validacijo.

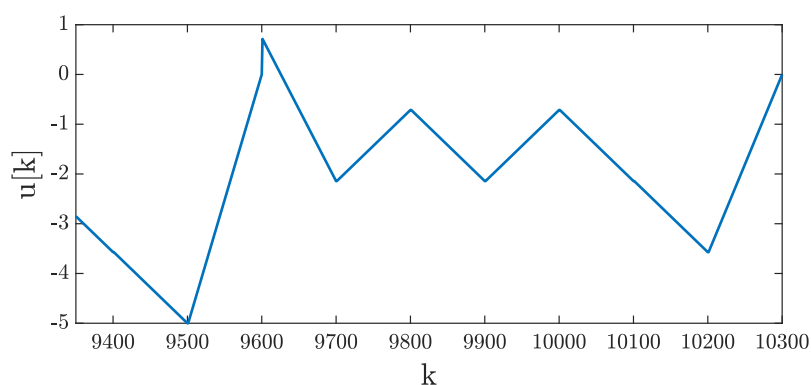
Slika 5.7: Validacija simulacijskih modelov a in b za model brez šuma (slika a) in model s šumom (slika b). Testni signal za validacijo (slika c).



(a) Izsek enokoračne predikcije za modela brez šuma:  $\sigma_n = 0$ . Prikazan tudi interval zaupanja  $2\sigma$ .

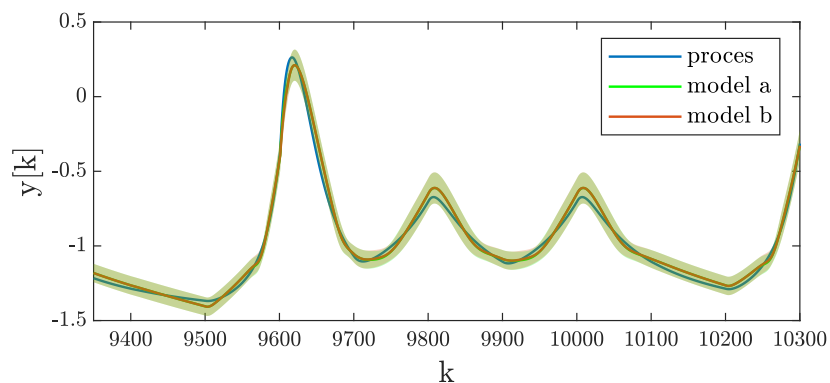


(b) Izsek enokoračne predikcije za modela s šumom:  $\sigma_n = 0,03$ . Prikazan tudi interval zaupanja  $2\sigma$ .

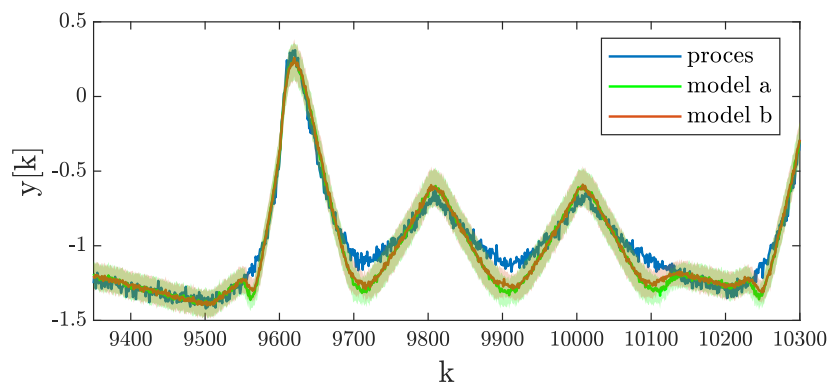


(c) Izsek testnega signala za validacijo.

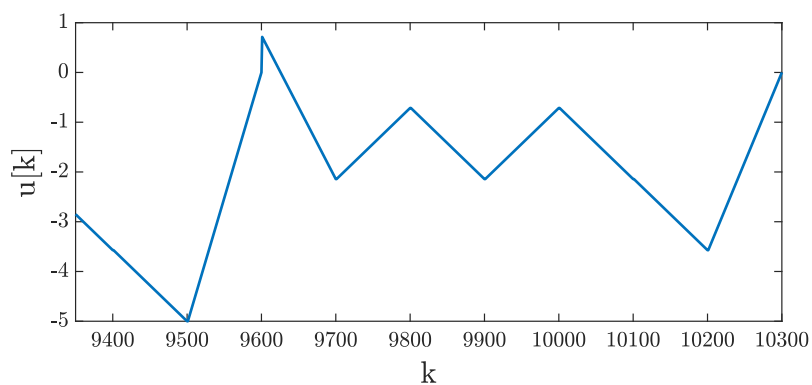
Slika 5.8: Validacija predikcijskih modelov brez šuma (a) in s šumom (b). Testni signal za validacijo (c). Prikazani so izseki signalov, dodan je tudi interval zaupanja  $2\sigma$ .



(a) Izsek predikcijskega modela za deset korakov predikcije za modela brez šuma:  $\sigma_n = 0$ . Prikazan tudi interval zaupanja  $2\sigma$ .



(b) Izsek predikcijskega modela za deset korakov predikcije za modela s šumom:  $\sigma_n = 0,03$ . Prikazan tudi interval zaupanja  $2\sigma$ .



(c) Izsek testnega signala za validacijo.

Slika 5.9: Validacija predikcijskega modela za deset korakov predikcije brez šuma (a) in s šumom (b). Testni signal za validacijo (c). Prikazani so izseki signalov, dodan je tudi interval zaupanja  $2\sigma$ .

drugače je na testni množici, kjer se modelu brez šuma  $RMSE$  ne poveča bistveno, pri modelu s šumom pa je povečanje znatno.

Iz slike 5.8 in tabele 5.4 vidimo, da je enokoračna napoved ustrezna za vse modele, pri čemer ni opaziti razlike v delovanju med modeloma a in b. Napaka  $RMSE$  je pri enokoračni napovedi majhna in je pri modelih s prisotnim šumom v velikem delu posledica zgolj tega. Rezultati med testno in učno množico so povsem primerljivi. Napaka  $RMSE$  med modelom in izmerjenim izhodom procesa brez šuma je manjša od velikosti šuma, saj metoda PCR delno odpravi normalni šum. Ker je normalni šum še zmeraj prisoten pri regresorju  $y[k - 1]$ , šum ni v celoti odpravljen.

Iz tabele 5.4 vidimo, da se rezultati deskoračne napovedi bližajo simulacijskim rezultatom. Na podlagi vrednosti  $RMSE$  bi dejali, da ni bistvene razlike v delovanju med modeloma a in b, vendar lahko vidimo iz slike 5.9, da so pri modelu a prisotne večje oscilacije v primeru z dodanim šumom. Zanimivo je, da dodani šum ne vpliva bistveno na napako  $RMSE$  na učni množici, kar bi kazalo na to, da je glavni izvor napake sama napaka modela in ne šum. Izkaže se, da je na testni množici odstopanje vseeno večje in da šum vpliva na kakovost modela, kar je očitno tudi iz slik 5.9a in 5.9b, kjer se napaka pri modelu z dodanim šumom bistveno poveča, model brez šuma pa dosega povsem ustrezne rezultate.

Preverili smo tudi, kakšna je ustreznost metode določitve ocene napake opisane v poglavju 4.4.4. V tabeli 5.5 smo prikazali rezultate za enokoračno napoved. Večje relativno odstopanje je sicer pri modelu, kjer šum ni bil prisoten; v primeru prisotnega šuma pa se metoda izkaže za ustrezno. V tabeli 5.6 smo prikazali rezultate za deskoračno napoved, kjer so odstopanja tolikšna, da je opisan pristop neustrezen. Do enakih zaključkov pridemo tudi na podlagi slik 5.8 in 5.9, kjer se izmerjen izhod procesa v primeru deskoračne napovedi nahaja prepogosto zunaj intervala zaupanja.

Tabela 5.5: Analiza ocene napake pri enokoračni napovedi.

| model | standardni<br>odklon napake | model: $\sigma_n = 0$ |        | model: $\sigma_n = 0,03$ |        |
|-------|-----------------------------|-----------------------|--------|--------------------------|--------|
|       |                             | učna                  | testna | učna                     | testna |
| (a)   | dejanski                    | 0,0042                | 0,0046 | 0,039                    | 0,040  |
|       | ocena                       | 0,0031                | 0,0037 | 0,038                    | 0,038  |
| (b)   | dejanski                    | 0,0035                | 0,0039 | 0,039                    | 0,040  |
|       | ocena                       | 0,0030                | 0,0032 | 0,038                    | 0,037  |

Tabela 5.6: Analiza ocene napake pri desetkoračni napovedi.

| model | standardni<br>odklon napake | model: $\sigma_n = 0$ |        | model: $\sigma_n = 0,03$ |        |
|-------|-----------------------------|-----------------------|--------|--------------------------|--------|
|       |                             | učna                  | testna | učna                     | testna |
| (a)   | dejanski                    | 0,050                 | 0,042  | 0,059                    | 0,051  |
|       | ocena                       | 0,015                 | 0,017  | 0,036                    | 0,035  |
| (b)   | dejanski                    | 0,046                 | 0,035  | 0,056                    | 0,049  |
|       | ocena                       | 0,015                 | 0,015  | 0,036                    | 0,035  |

### 5.3 Identifikacija Hammersteinovega modela s predhodnim filtriranjem

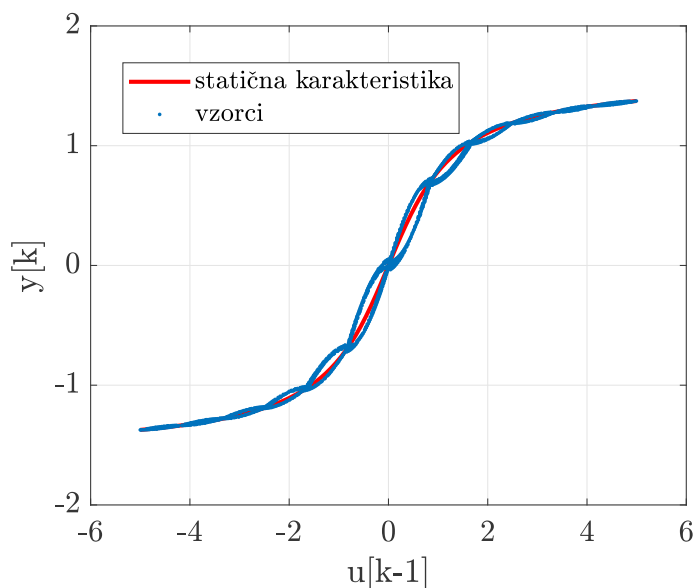
Identifikacijo Hammersteinovega modela smo ponovili tudi na način s predhodnim filtriranjem, ki je opisan v poglavju 4.4.5.

#### 5.3.1 Predstavitev sistema

Za samo identifikacijo je bilo potrebno določiti lomno frekvenco Hammersteinovega modela. Slednje smo naredili eksperimentalno. Lomna frekvenca je enaka  $f_l = 0,02$  Hz, če je čas vzorčenja  $T_s = 1$  s. Časovno konstanto filtra smo tako določili kot:

$$T_f = \frac{3}{2\pi f_l}. \quad (5.9)$$

Vhod in izhod sistema smo filtrirali v skladu s filtrom opisanim z enačbo 4.48. Filtriran vhodno-izhodni prostor s statično karakteristiko je prikazana na sliki 5.10.



Slika 5.10: Filtriran vhodno-izhodni prostor.

### 5.3.2 Rezultati

V tem razdelku bomo prikazali rezultate vhodno-izhodne identifikacije za Hammersteinov model brez dodanega šuma in z dodanim šumom. Sistem smo vzbujali s stopničastim signalom na intervalu  $[-5, 5]$ , ki je prikazan na sliki 5.11. Število vseh vzorcev je bilo 15560. Vzbujalni signal smo spremnili, da bi dosegli boljše prekrivanje s statično karakteristiko (slika 5.10).

Do primernih parametrov smo prišli s poskušanjem in na podlagi predhodnih identifikacij. Zaradi slabših rezultatov pri manjšem številu končnih rojev, smo izbrali take parametre, ki so število rojev povečali. V tabeli 5.7 so prikazani uporabljeni vhodni parametri. Tokrat je bilo potrebno pri modelu s šumom spremeniti večino parametrov. Dodani šum  $\sigma_n$  je enak kot pri nefiltriranem pristopu. Izbira regresorjev za določitev pripadnosti pri *modelu a* in *modelu b* je tudi enaka kot pri nefiltriranem pristopu.

Prikaz rojev ob koncu rojenja za model brez šuma in model s šumom je prikazana na sliki 5.12. Pri modelu brez šuma je končno število rojev 6, pri modelu s šumom pa 8. Ker je rojev več, so slednji manjši (velja tudi za nefiltriran pro-

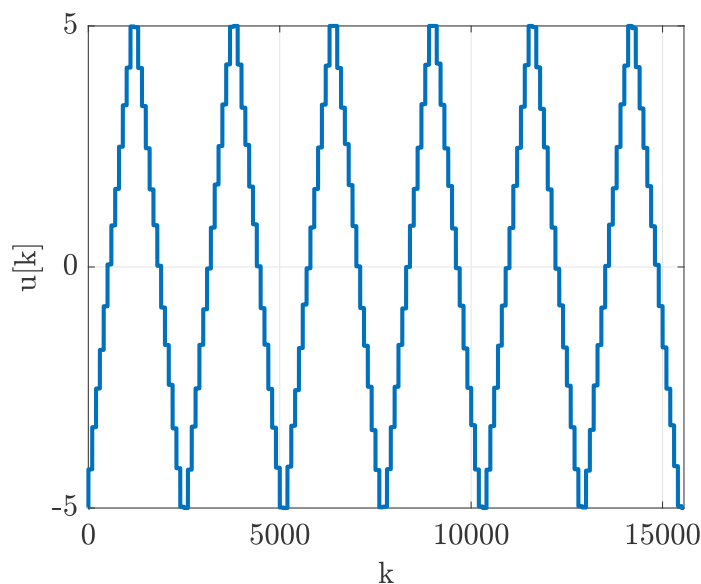
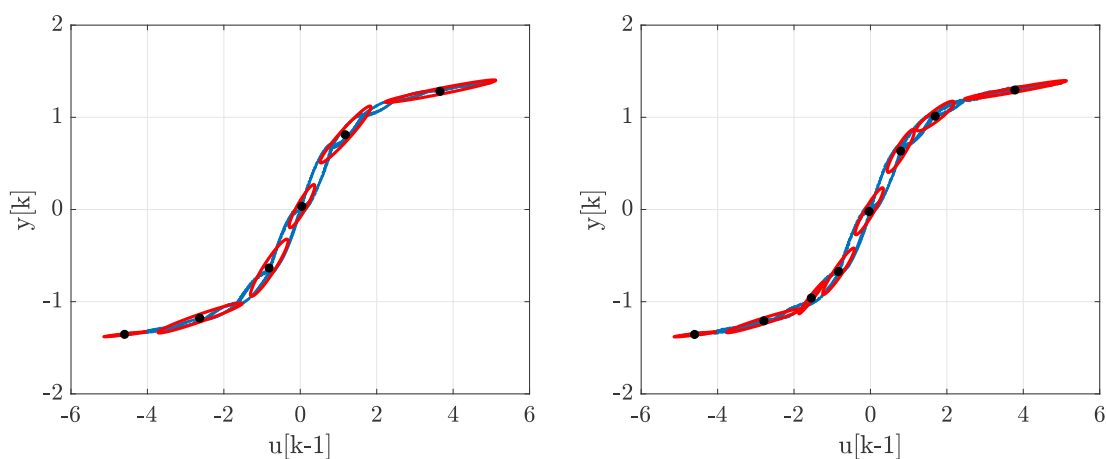
Slika 5.11: Signala vzbujanja  $u[k]$ .

Tabela 5.7: Uporabljeni parametri za rojenje.

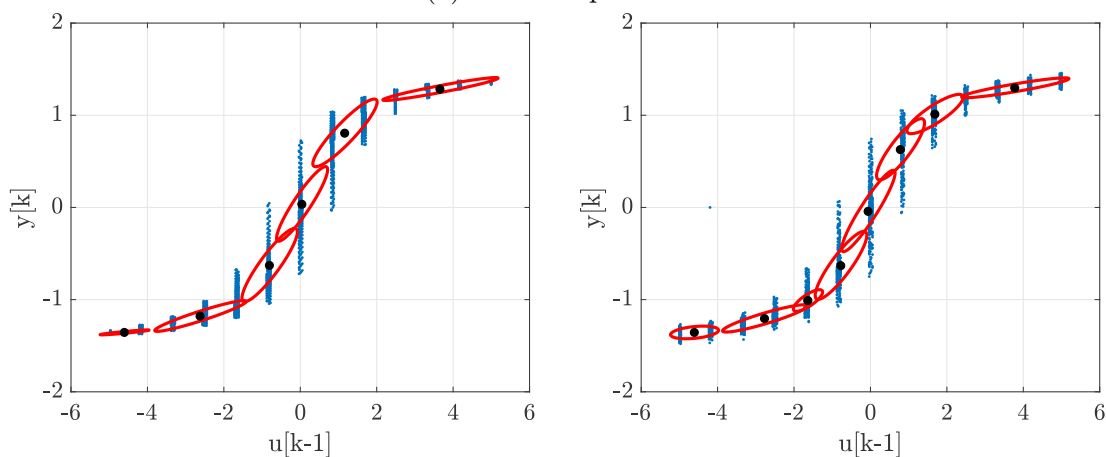
| $\sigma_n$ | $\Gamma_{min}$ | $\kappa_{join}$ | $v_{max}$ | $N_{max}$ | $f_l$ |
|------------|----------------|-----------------|-----------|-----------|-------|
| 0          | 0,55           | 1,6             | 0,001     | 10        | 0,02  |
| 0,03       | 0,50           | 1,6             | 0,0003    | 10        | 0,02  |

stor). Na manjšo velikost rojev vpliva tudi manj gost regresorski prostor zaradi drugačne izbire vzbujačnega signala. Iz slike 5.12 vidimo, da do zadnje spremembe števila rojev pride bistveno hitreje v primeru brez šuma. V primeru šuma vidimo, da se število rojev načeloma ustali, vendar se občasno pojavijo novi, ki se sicer kmalu združijo z enim od obstoječih rojev. Zopet pride do večjega števila rojev v primeru šuma.

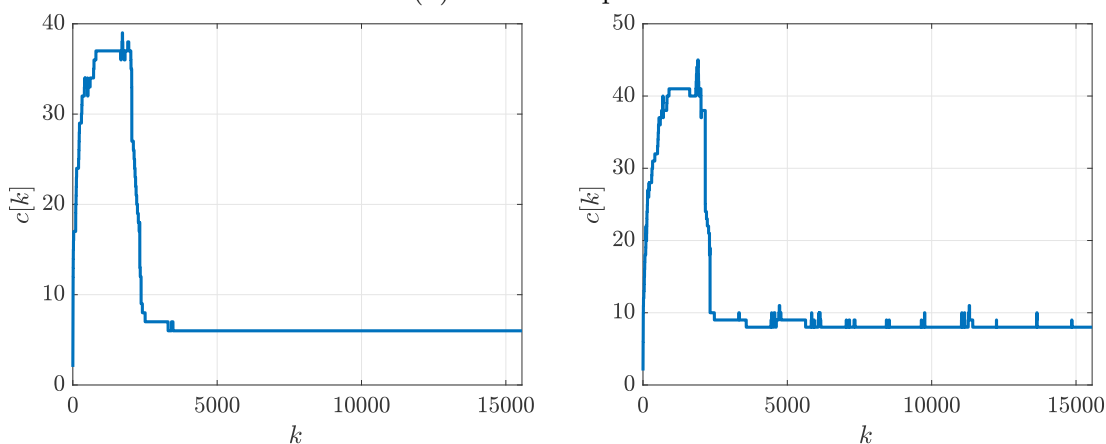
V tabeli 5.8 je prikazana primerjava parametrov  $b_1$  in  $a_1$  lokalnega identifikacijskega modela z lineariziranim modelom. Parametri lokalnih modelov se slabše ujemajo s parametri lineariziranih modelov kot pri nefiltrirani metodi; očitna sprememba je predvsem pri parametru  $a_1$ , kjer so odstopanja nekoliko večja. Iz tabele 5.8 lahko sicer opazimo, da je pri rojih, ki se nahajajo na sredini prostora, kjer je ojačanje največje, ujemanje parametrov z lineariziranim modelom najboljše. Pri modelu s šumom se pojavi celo roj ( $\mu(1) = -4,6033$ ), kjer je



(a) Filtriran prostor.



(b) Nefiltriran prostor.

(c) Časovna odvisnost spreminjanja števila rojev  $c[k]$ 

Slika 5.12: Primerjava rojev z robom  $2\sigma$  med modelom brez šuma (levi grafi) in modelom s šumom (desni grafi). Primerjava podana za filtriran prostor (a), za nefiltriran prostor (b) in za časovna odvisnost spreminjanja števila rojev  $c[k]$  (c).

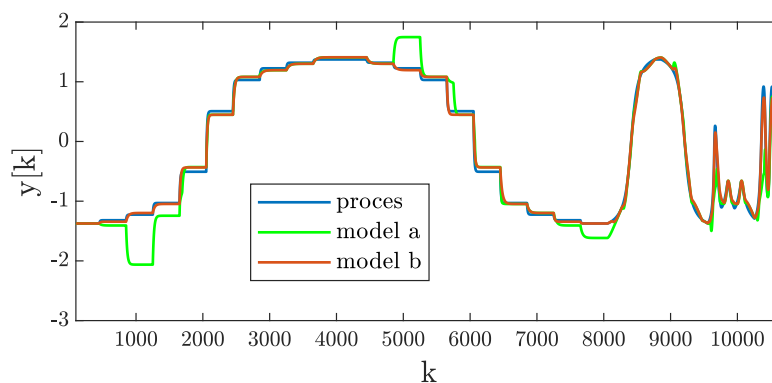
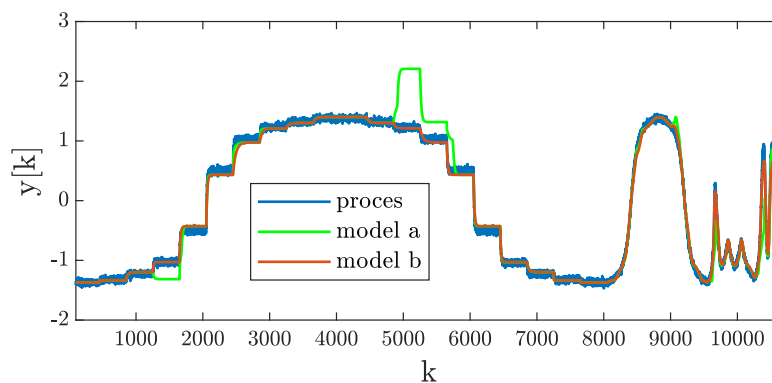
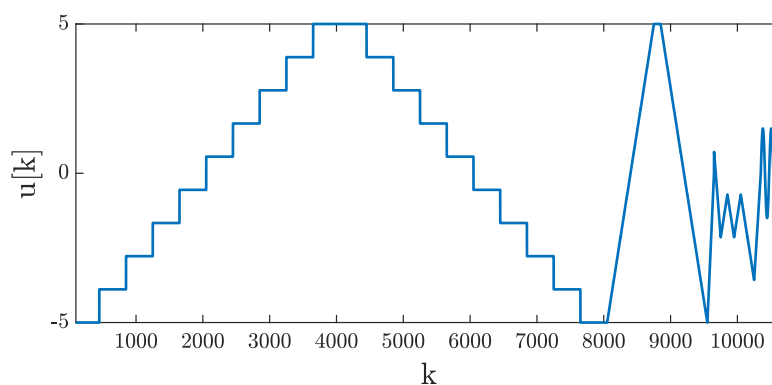
Tabela 5.8: Primerjava parametrov  $b_1$  in  $a_1$  lokalnega identifikacijskega (indeks  $m$ ) modela z lineariziranim modelom (indeks  $l$ );  $\mu(1)$  je središče roja glede na vhod  $u$ .

| $\sigma_n$ | $\mu(1)$ | $b_1^l$ | $b_1^m$ | $a_1^l$ | $a_1^m$ |
|------------|----------|---------|---------|---------|---------|
| 0          | -4,5983  | 0,0045  | 0,0052  | 0,9000  | 0,8893  |
|            | -2,6112  | 0,0127  | 0,0060  | 0,9000  | 0,9680  |
|            | -0,8011  | 0,0609  | 0,0417  | 0,9000  | 0,9433  |
|            | 0,0475   | 0,0997  | 0,0832  | 0,9000  | 0,9003  |
|            | 1,1699   | 0,0422  | 0,0283  | 0,9000  | 0,9526  |
|            | 3,6692   | 0,0069  | 0,0024  | 0,9000  | 0,9748  |
| 0,03       | -4,6033  | 0,0045  | 0,0379  | 0,9000  | 0,0089  |
|            | -2,7621  | 0,0116  | 0,0069  | 0,9000  | 0,9627  |
|            | -1,6337  | 0,0273  | 0,0345  | 0,9000  | 0,9048  |
|            | -0,7703  | 0,0628  | 0,0462  | 0,9000  | 0,9432  |
|            | -0,0546  | 0,0997  | 0,0931  | 0,9000  | 0,8845  |
|            | 0,7879   | 0,0617  | 0,0477  | 0,9000  | 0,9405  |
|            | 1,6922   | 0,0259  | 0,0143  | 0,9000  | 0,9773  |
|            | 3,7898   | 0,0065  | 0,0027  | 0,9000  | 0,9699  |

lokalni model povsem zgrešen. Vzrok za zgrešeno napoved je verjetno pozicija v območju nasičenja, kjer je zaradi manjše dinamike večja korelacija med regresorji, kar lahko vidimo na podlagi oblike in velikosti roja na sliki 5.12a.

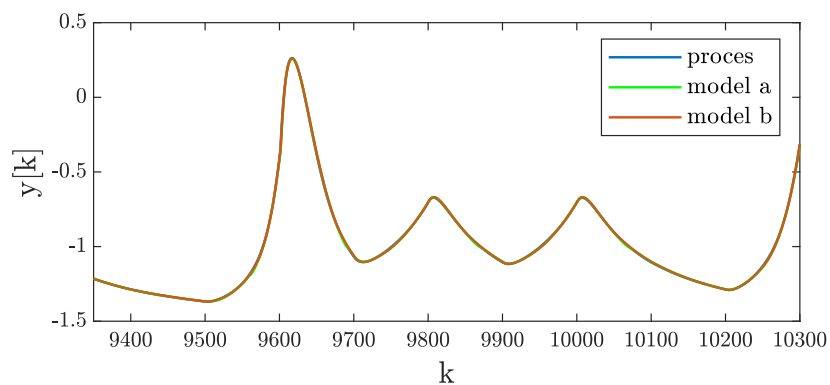
Model a in model b smo ovrednotili na enak način kot v primeru brez filtriranja. Vrednotenje je potekalo za primer simulacije, enokoračne napovedi in desetkoračne napovedi na učnem in testnem signalu. Napako  $RMSE$  smo prikazali v tabeli 5.9. V primeru modela s šumom smo je izračunali glede na sistem, kjer je dodan šum ( $RMSE_n$ ) in kjer je šum odvzet ( $RMSE$ ). Kvalitativna ocena simulacije je vidna na sliki 5.13, izsek kvalitativne ocene enokoračne napovedi je viden na sliki 5.14 in izsek kvalitativne ocene desetkoračne napovedi je viden na sliki 5.15.

Iz simulacijske slike 5.13 lahko vidimo, da je model b, kjer pripadnost določimo zgolj na podlagi vrednosti vhoda, bistveno bolj primeren od modela a. Zaradi uporabe zgolj izhoda modela pri določevanju pripadnosti vzorcev se vzorci ne raz-

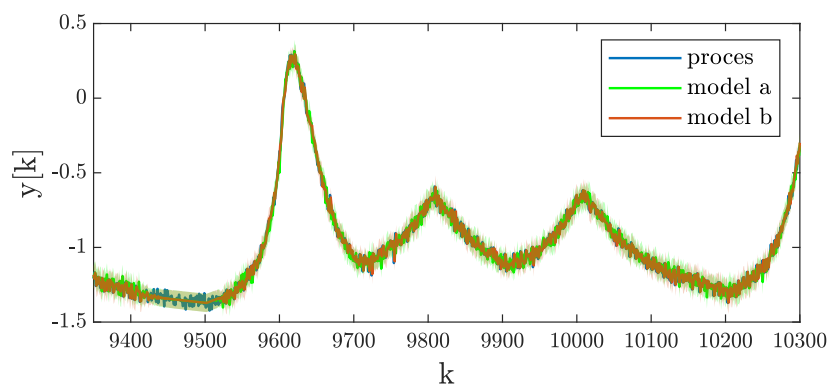
(a) Simulacija za modela brez šuma:  $\sigma_n = 0$ .(b) Simulacija za modela s šumom:  $\sigma_n = 0,03$ .

(c) Testni signal za validacijo.

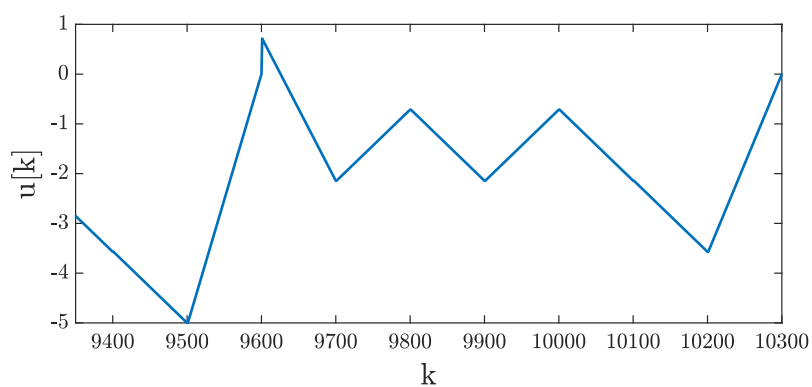
Slika 5.13: Validacija simulacijskih modelov a in b za model brez šuma (slika a) in model s šumom (slika b). Testni signal za validacijo (slika c).



(a) Izsek enokoračne predikcije za modela brez šuma:  $\sigma_n = 0$ . Prikazan tudi interval zaupanja  $2\sigma$ .

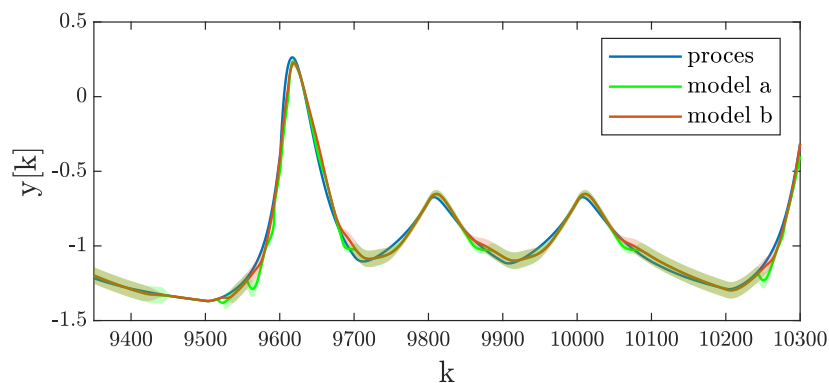


(b) Izsek enokoračne predikcije za modela s šumom:  $\sigma_n = 0,03$ . Prikazan tudi interval zaupanja  $2\sigma$ .

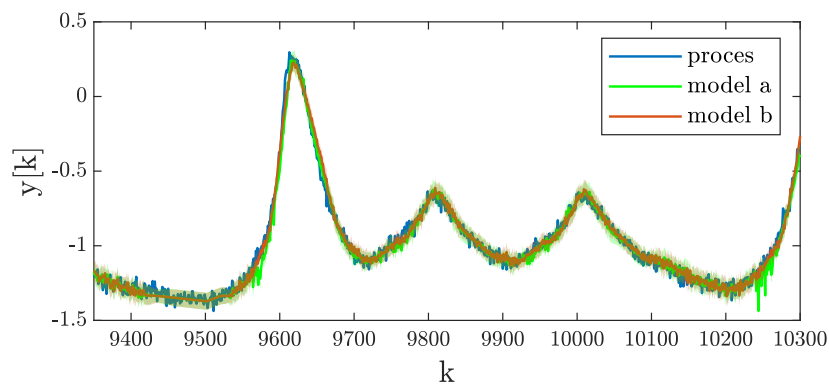


(c) Izsek testnega signala za validacijo.

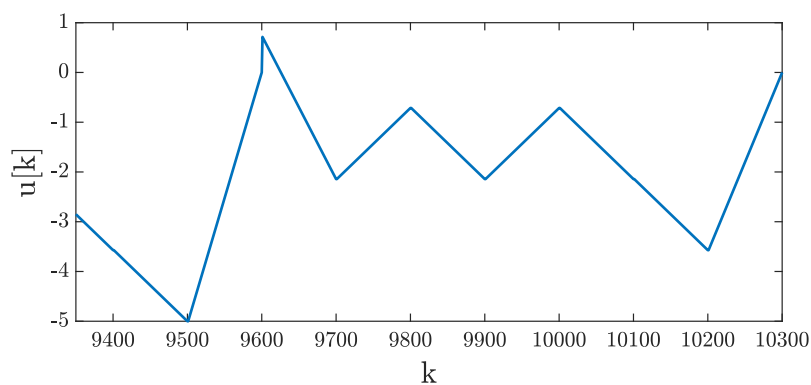
Slika 5.14: Validacija predikcijskega modela brez šuma (a) in model s šumom (b). Prikazani so izseki signalov.



(a) Izsek predikcijskega modela za deset korakov predikcije za model brez šuma:  $\sigma_n = 0$ . Prikazan tudi interval zaupanja  $2\sigma$ .



(b) Izsek predikcijskega modela za deset korakov predikcije za model s šumom:  $\sigma_n = 0,03$ . Prikazan tudi interval zaupanja  $2\sigma$ .



(c) Izsek testnega signala za validacijo.

Slika 5.15: Validacija predikcijskega modela za deset korakov predikcije brez šuma (a) in s šumom (b). Testni signal za validacijo (c). Prikazani so izseki signalov, dodan je tudi interval zaupanja  $2\sigma$ .

Tabela 5.9: Napaka  $RMSE$  za model brez šuma in model s šumom.  $RMSE$  računa napako med modelom in sistemom brez dodanega šuma.  $RMSE_n$  računa napako med modelom in sistemom z dodanim šumom.

| model | $\sigma_n$ | metrika  | simulacija |        | 1-koračna |        | 10-koračna |        |
|-------|------------|----------|------------|--------|-----------|--------|------------|--------|
|       |            |          | učna       | testna | učna      | testna | učna       | testna |
| (a)   | 0          | $RMSE$   | 0,1629     | 0,2425 | 0,0042    | 0,0046 | 0,0492     | 0,0416 |
|       | 0,03       | $RMSE$   | 0,1734     | 0,2328 | 0,0266    | 0,0268 | 0,0496     | 0,0408 |
|       |            | $RMSE_n$ | 0,1754     | 0,2350 | 0,0404    | 0,0401 | 0,0574     | 0,0506 |
| (b)   | 0          | $RMSE$   | 0,0523     | 0,0575 | 0,0035    | 0,0039 | 0,0456     | 0,0347 |
|       | 0,03       | $RMSE$   | 0,0515     | 0,0655 | 0,0267    | 0,0269 | 0,0477     | 0,0384 |
|       |            | $RMSE_n$ | 0,0595     | 0,0718 | 0,0408    | 0,0400 | 0,0570     | 0,0484 |

vrstijo pravilno. Slednje je opazno predvsem v področju nasičenja, kjer so lokalni modeli slabše določeni, kar je razvidno iz tabele 5.8. V področju večjega ojačanja sistema je prileganje modela a boljše. Zopet opazimo oscilacije na mejah med roji, ki so tudi tokrat bolj izrazite pri modelu a. Tudi iz napaka  $RMSE$  iz tabele 5.9 vidimo, da dosega model a bistveno slabše rezultate. Model b dosega celo boljše rezultate napram nefiltriranem pristopu, kar bi pripisali večjemu številu lokalnih modelov. Zopet ni zaznati vpliva šuma na kvaliteto modela. Na testni množici dosega modela a in b za primer z dodanim šumom in primer brez

Enokoračna napoved se zopet izkaže za ustrezno za vse modele, kar vidimo iz slike 5.14 in tabele 5.9; model b dosega malo boljše rezultate zgolj v primeru brez dodanega šuma, v primeru z dodanim šumom sta modela a in b primerljiva. Napaka  $RMSE$  je pri enokoračni napovedi majhna in je pri modelu s prisotnim šumom v velikem delu posledica zgolj tega. Rezultati med testno in učno množico so povsem primerljivi. Ponovno je napaka  $RMSE$  med modelom in referenčnim sistemom brez šuma manjša od velikosti šuma, kar pomeni, da metoda PCR zopet zgolj delno odpravi normalni šum, saj je še zmeraj prisoten vpliv normalnega šuma pri regresorju  $y[k-1]$ . Napaka  $RMSE$  je manjša kot pri nefiltrirani metodi, kar bi pripisali večjemu številu rojev, čeprav slednji ne ustrezajo tako dobro lineariziranemu modelu.

Iz slike 5.15 in tabele 5.9 vidimo, da obstaja znatna razlika pri desetkoračni napovedi med modeloma a in b za primer z dodanim šumom in primer brez

Tabela 5.10: Analiza ocene napake pri enokoračni napovedi.

| model | standardni<br>odklon napake | model: $\sigma_n = 0$ |        | model: $\sigma_n = 0,03$ |        |
|-------|-----------------------------|-----------------------|--------|--------------------------|--------|
|       |                             | učna                  | testna | učna                     | testna |
| (a)   | dejanski                    | 0,0042                | 0,0046 | 0,039                    | 0,040  |
|       | ocena                       | 0,0031                | 0,0037 | 0,038                    | 0,038  |
| (b)   | dejanski                    | 0,0035                | 0,0039 | 0,039                    | 0,040  |
|       | ocena                       | 0,0030                | 0,0032 | 0,038                    | 0,037  |

Tabela 5.11: Analiza ocene napake pri desetkoračni napovedi.

| model | standardni<br>odklon napake | model: $\sigma_n = 0$ |        | model: $\sigma_n = 0,03$ |        |
|-------|-----------------------------|-----------------------|--------|--------------------------|--------|
|       |                             | učna                  | testna | učna                     | testna |
| (a)   | dejanski                    | 0,049                 | 0,042  | 0,059                    | 0,051  |
|       | ocena                       | 0,015                 | 0,016  | 0,038                    | 0,037  |
| (b)   | dejanski                    | 0,045                 | 0,035  | 0,056                    | 0,049  |
|       | ocena                       | 0,015                 | 0,015  | 0,037                    | 0,036  |

dodanega šuma. Zopet so prisotne večje oscilacije pri prehodu med roji pri modelu a, kar pripomore k nekoliko večji vrednosti napake  $RMSE$ . Zanimivo je, da dodan šum v primeru desetkoračne napovedi pri modelu a ne vpliva na kvaliteto modela, saj so si vrednosti  $RMSE$  ne glede na šum podobne na obeh množicah. Pri modelu b je drugače in je vrednost napake  $RMSE$  pri dodanem šumu večja. Pri desetkoračni napovedi je napaka  $RMSE$  na testnem signalu manjša v primerjavi z nefiltriranim modelom, pri čemer ima model b skoraj dvakrat manjšo napako  $RMSE$  v primeru dodanega šuma. Napake  $RMSE$  ne moremo primerjati na učnem signalu, saj signala nista enaka.

Zopet smo preverili, kakšna je ustreznost metode določitve ocene napake opisane v poglavju 4.4.4. V tabeli 5.10 smo prikazali rezultate za enokoračno napoved. Podamo lahko enake zaključke kot pri nefiltriranim pristopu. Metoda je ustrezna, kar je vidno tudi iz slike 5.14. V tabeli 5.11 smo prikazali rezultate za desetkoračno napoved, kjer so odstopanja tolikšna, da je opisan pristop zopet neustrezen. Slednje je vidno tudi na sliki 5.15, saj se izhod procesa, ki ga identificiramo, nahaja zunaj intervala zaupanja.

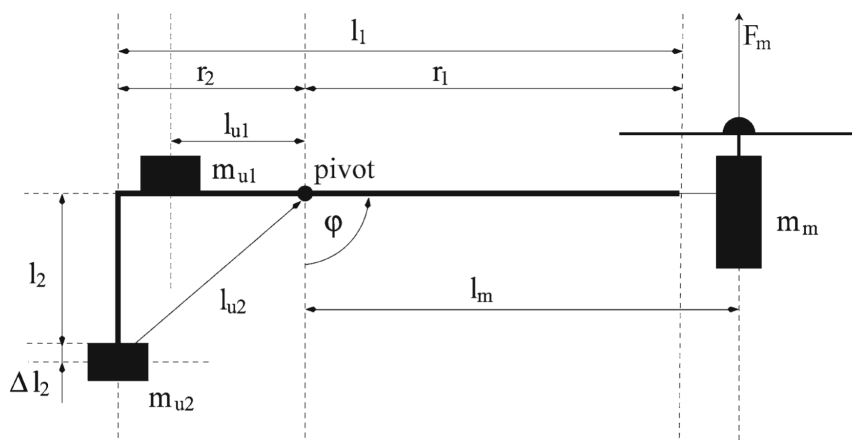
## 5.4 Identifikacija naprave Helio-crane

Vhodno-izhodno identifikacijo smo izvedli tudi na procesu Helio-crane, ki smo ga simulirali z natančnim modelom [35] in se uporablja v sklopu laboratorijskih vaj pri predmetu Inteligentni sistemi za podporo odločanju. Identifikacijo smo izvedli tudi na podatkih, ki smo jim dodali šum na izhodu, in rezultate validirali.

### 5.4.1 Predstavitev podatkov

Model Helio-crane je vrteči sistem, čigar shemo lahko vidimo na sliki 5.16. Iz [35] sledi, da gre za nelinearni sistem drugega reda, kjer diferencialna enačba ne vsebuje odvodov vzbujanja  $u$ . Tako je smiselna izbira regresorskega vektorja:

$$z[k] = [u[k-2], y[k-2], y[k-1], y[k]]. \quad (5.10)$$

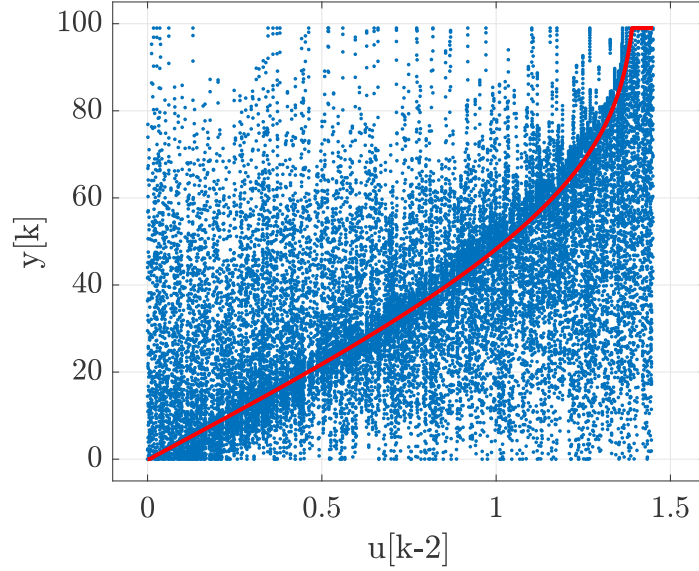


Slika 5.16: Shema modela Helio-crane. Slika je povzeta po [35].

Filtriran vhodno-izhodni prostor s statično karakteristiko je prikazana na sliki 5.17. Uporabili smo čas vzorčenja  $T_s = 0,1$  s.

### 5.4.2 Rezultati

V tem razdelku bomo prikazali rezultate vhodno-izhodne identifikacije za sistem Helio-crane brez dodanega šuma in z dodanim šumom. Pri izračunu pripadnosti



Slika 5.17: Filtriran vhodno-izhodni prostor s statično karakteristiko.

(enačba 4.39) pri določitvi izhoda modela smo uporabili različno število regresorjev. *Model a* sestavljajo regresorji:

$$\tilde{\mathbf{z}}^T = [u[k-2], y[k-2], y[k-1], \tilde{y}[k]], \quad (5.11)$$

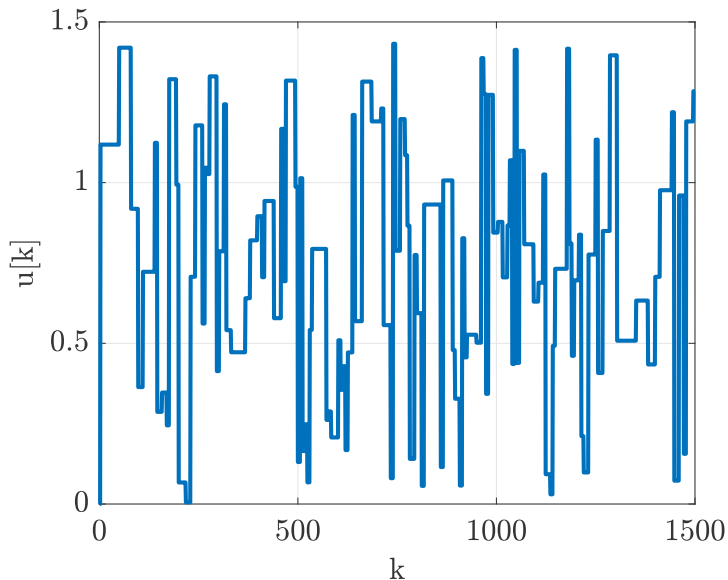
kjer je  $\tilde{y}[k]$  izhod modela v trenutku  $k$ . Pri *modelu b* je regresorski vektor sestavljen zgolj iz vrednosti vhoda:

$$\tilde{\mathbf{z}}^T = [u[k-2]]. \quad (5.12)$$

Sistem smo vzbujali s signalom APRBS na intervalu  $[0, 1.45]$ ; izsek signala je prikazan na sliki 5.18; število vseh vzorcev je bilo 30600.

Do primernih parametrov smo prišli s poizkušanjem in na podlagi predhodnih identifikacij. V tabeli 5.12 so prikazani uporabljeni vhodni parametri. Modela se razlikujeta zgolj v parametru  $v_{max}$ . Dodan je bil šum z normalno porazdelitvijo s standardnim odklonom  $\sigma_n = 0,5$ .

Prikaz rojev ob koncu rojenja za model brez šuma in model s šumom je prikazana na sliki 5.19a. Pri modelu brez šuma je končno število rojev 4, pri modelu s šumom pa 3. Opazimo večja prekrivanja rojev v vhodno-izhodnem prostoru,



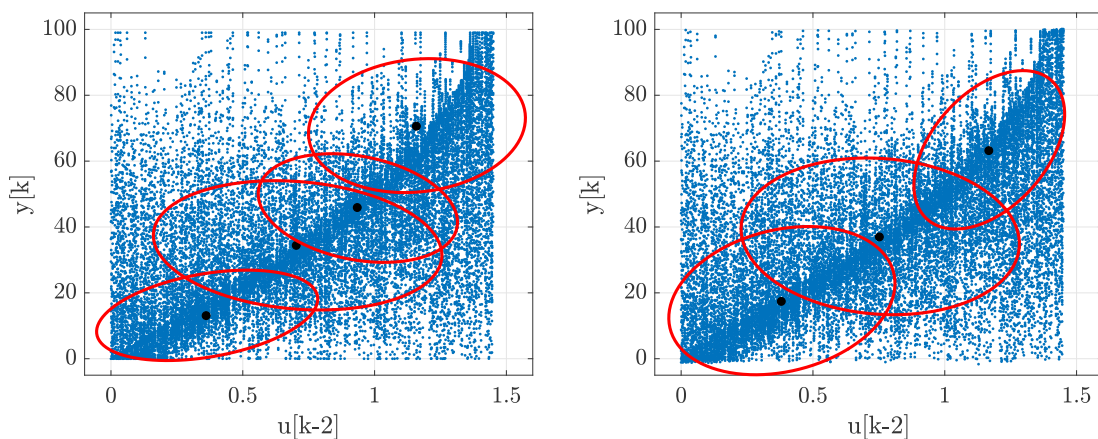
Slika 5.18: Izsek signala vzbujanja APRBS.

Tabela 5.12: Uporabljeni parametri za rojenje.

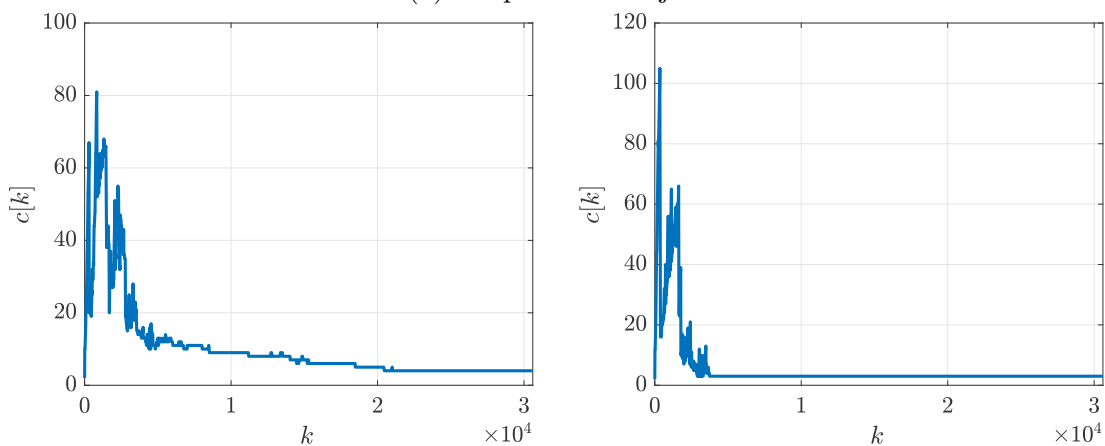
| $\sigma_n$ | $\Gamma_{min}$ | $\kappa_{join}$ | $v_{max}$ | $N_{max}$ |
|------------|----------------|-----------------|-----------|-----------|
| 0          | 0,40           | 1,70            | 0,05      | 14        |
| 0,5        | 0,40           | 1,70            | 0,45      | 14        |

vendar velja poudariti, da poteka rojenje tokrat v 4-dimenzionalnem prostoru. Iz slike 5.19b vidimo, da se število rojev bistveno hitreje ustali pri modelu z dodatnim šumom, kar bi lahko pripisali bolj polnemu prostoru zaradi šuma, kar privede do hitrejšega in večjega prekrivanja rojev.

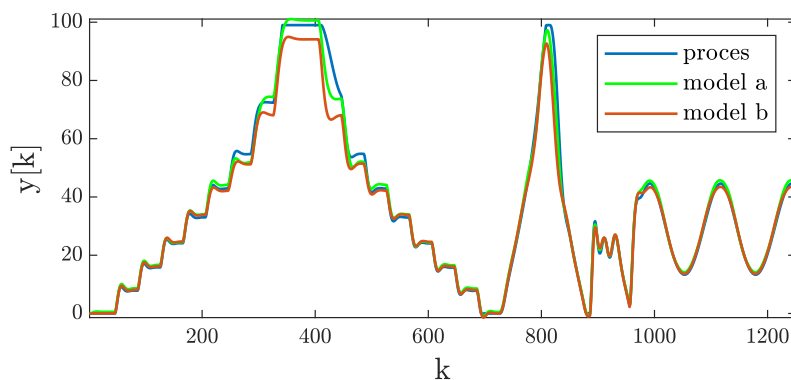
Model a in model b smo ovrednotili na enak način kot v primeru modela Hammerstein. Vrednotenje je potekalo za primer simulacije, enokoračne napovedi in desetkoračne napovedi na učnem in testnem signalu. Testni signal je viden na sliki 5.20c. Napako  $RMSE$  smo prikazali v tabeli 5.13. V primeru modela s šumom smo jo izračunali za primer, kjer je šum dodan ( $RMSE_n$ ) in kjer je šum odvzet ( $RMSE$ ). Kvalitativna ocena simulacije je vidna na sliki 5.20, kvalitativna ocena enokoračne napovedi je vidna na sliki 5.21 in kvalitativna ocena desetkoračne napovedi je vidna na sliki 5.22.



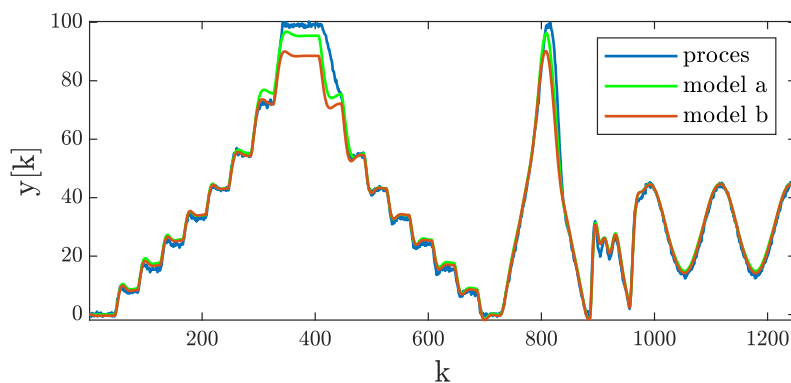
(a) Razporeditev rojev.

(b) Časovna odvisnost spreminjanja števila rojev  $c[k]$ 

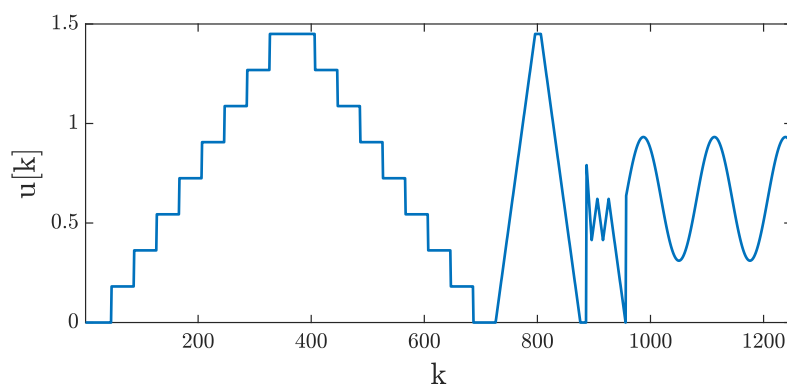
Slika 5.19: Primerjava rojev (a) z robom  $2\sigma$  med modelom brez šuma (levi grafi) in modelom s šumom (desni grafi); primerjava podana še za časovno odvisnost spreminjanja števila rojev  $c[k]$  (b).



(a) Simulacija za modela brez šuma:  $\sigma_n = 0$ .

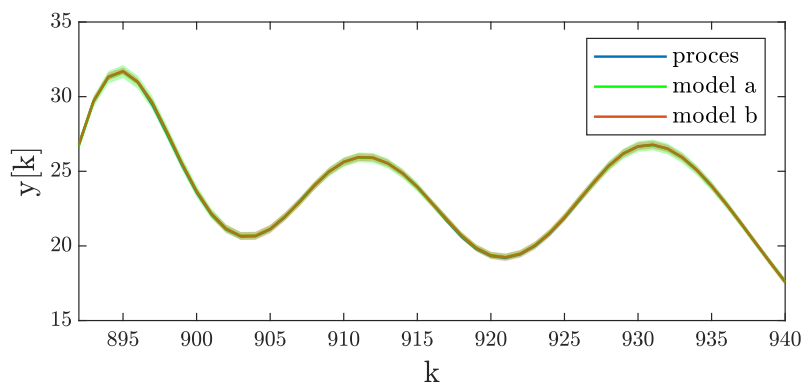


(b) Simulacija za modela s šumom:  $\sigma_n = 0,5$ .

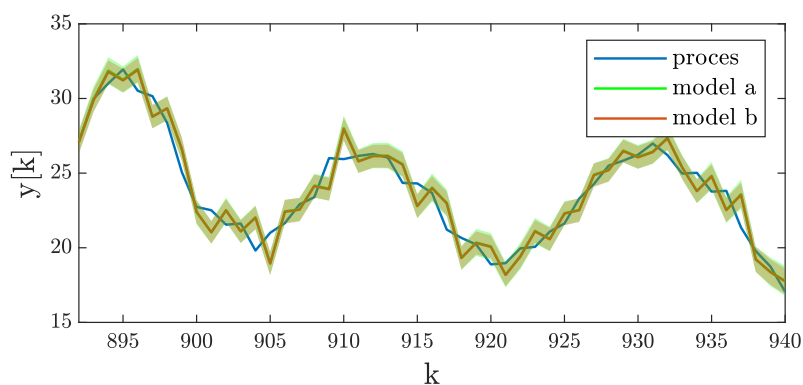


(c) Testni signal za validacijo.

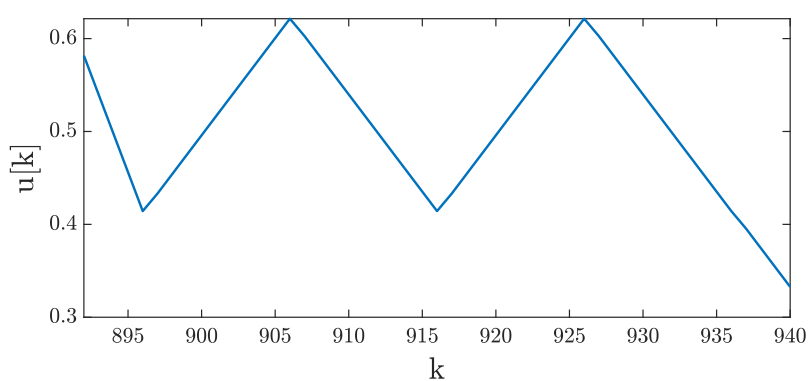
Slika 5.20: Validacija simulacijskih modelov a in b za model brez šuma (slika a) in model s šumom (slika b). Testni signal za validacijo (slika c).



(a) Izsek enokoračne predikcije za modela brez šuma:  $\sigma_n = 0$ . Prikazan tudi interval zaupanja  $2\sigma$ .

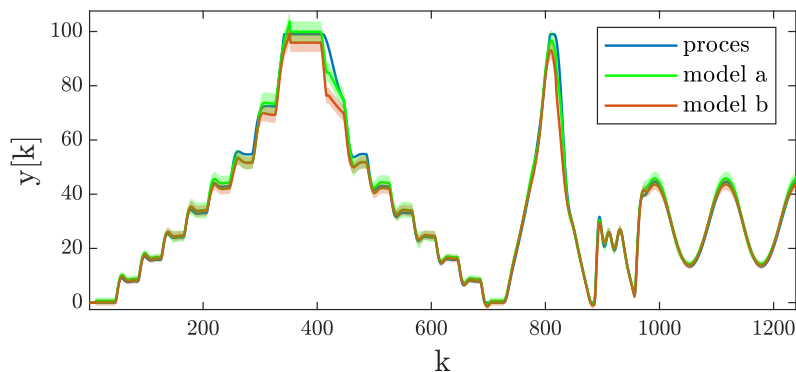


(b) Izsek enokoračne predikcije za modela s šumom:  $\sigma_n = 0,5$ . Prikazan tudi interval zaupanja  $2\sigma$ .

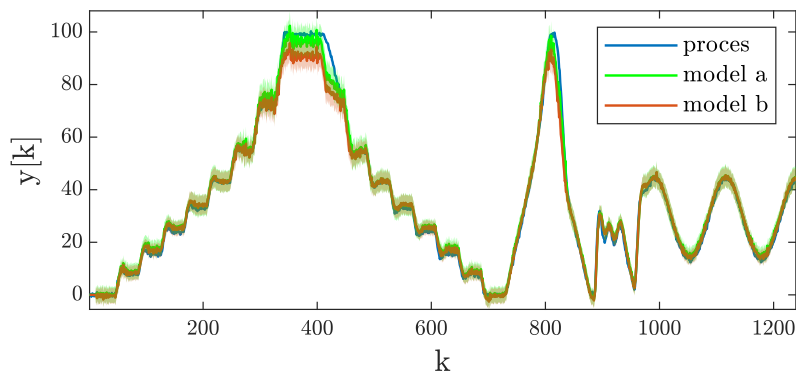


(c) Izsek testnega signala za validacijo.

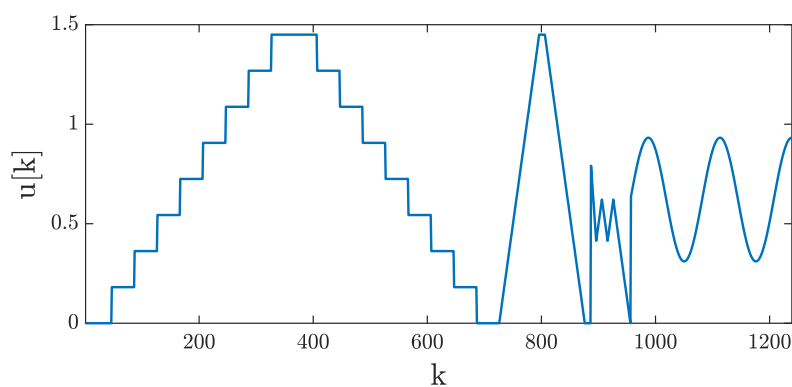
Slika 5.21: Validacija predikcijskih modelov brez šuma (a) in s šumom (b). Testni signal za validacijo (c). Prikazani so izseki signalov, dodan je tudi interval zaupanja  $2\sigma$ .



(a) Predikcijski model za deset korakov predikcije za modela brez šuma:  $\sigma_n = 0$ . Prikazan tudi interval zaupanja  $2\sigma$ .



(b) Predikcijski model za deset korakov predikcije za modela s šumom:  $\sigma_n = 0,5$ . Prikazan tudi interval zaupanja  $2\sigma$ .



(c) Testni signal za validacijo.

Slika 5.22: Validacija predikcijskega modela za deset korakov predikcije brez šuma (a) in s šumom (b). Testni signal za validacijo (c). Prikazani so izseki signalov, dodan je tudi interval zaupanja  $2\sigma$ .

Tabela 5.13: Napaka  $RMSE$  za model brez šuma in model s šumom.  $RMSE$  računa napako med modelom in sistemom brez dodanega šuma.  $RMSE_n$  računa napako med modelom in sistemom z dodanim šumom.

| model | $\sigma_n$ | metrika  | simulacija |        | 1-koračna |        | 10-koračna |        |
|-------|------------|----------|------------|--------|-----------|--------|------------|--------|
|       |            |          | učna       | testna | učna      | testna | učna       | testna |
| (a)   | 0          | $RMSE$   | 2,01       | 2,50   | 0,23      | 0,11   | 1,95       | 1,88   |
|       | 0,5        | $RMSE$   | 3,13       | 3,01   | 0,98      | 0,96   | 2,70       | 2,48   |
|       |            | $RMSE_n$ | 3,17       | 3,06   | 1,09      | 1,09   | 2,76       | 2,54   |
| (b)   | 0          | $RMSE$   | 6,10       | 4,44   | 0,38      | 0,17   | 4,85       | 3,48   |
|       | 0,5        | $RMSE$   | 4,61       | 4,47   | 1,00      | 0,97   | 3,77       | 3,69   |
|       |            | $RMSE_n$ | 4,64       | 4,50   | 1,11      | 1,09   | 3,79       | 3,72   |

Iz simulacijske slike 5.13 lahko vidimo, da se model a, kjer so uporabljeni vsi regresorji pri izračunu pripadnosti, izkaže za bolj ustreznega tako v primeru brez šuma kot v primeru z dodanim šumom. Za oba modela velja, da je ujemanje z izhodom modeliranega procesa dobro zlasti pri nižjih vrednostih vzbujačnega signala; pri višjih, kjer je nelinearnost bolj izrazita, so odstopanja večja. Oscilacije pri prehodih med modeli niso tako izrazite kot v primeru Hammersteinovega modela. Napaka  $RMSE$  iz tabele 5.13 se nam zdi ustrezna glede na izhodno območje procesa, pri čemer tudi tu vidimo, da model a dosega boljše rezultate. Vpliv šuma na kvaliteto modela tako ni bistven.

Enokoračna napoved se tudi v tem primeru izkaže za ustrezno, kar vidimo iz slike 5.21 in tabele 5.13, pri čemer ni zaznati razlike med modeloma a in b. Napaka  $RMSE$  je pri enokoračni napovedi majhna in je pri modelu s prisotnim šumom v velikem delu posledica zgolj tega; velja upoštevati, da je sistem drugega reda in se pri napovedi uporablja dva pošumljena zakasnjena izhoda. Rezultati med testno in učno množico so primerljivi pri modelu z dodanim šumom; model brez dodanega šuma dosega celo boljše rezultate.

Iz slike 5.22 in tabele 5.13 vidimo, da dosega model a boljše rezultate napram modelu b. Vpliv dodanga šuma se zmanjša napram enokoračni napovedi, pri čemer dosega model brez dodanega šuma še zmeraj nekoliko boljše rezultate. Očitno napaka samega modela izrazito prevladuje nad napako, ki je posledica šuma.

Tabela 5.14: Analiza ocene napake pri enokoračni napovedi.

| model | standardni<br>odklon napake | model: $\sigma_n = 0$ |        | model: $\sigma_n = 0,5$ |        |
|-------|-----------------------------|-----------------------|--------|-------------------------|--------|
|       |                             | učna                  | testna | učna                    | testna |
| (a)   | dejanski                    | 0,23                  | 0,11   | 1,09                    | 1,09   |
|       | ocena                       | 0,20                  | 0,21   | 1,02                    | 1,02   |
| (b)   | dejanski                    | 0,38                  | 0,17   | 1,11                    | 1,09   |
|       | ocena                       | 0,18                  | 0,18   | 0,92                    | 0,91   |

Tabela 5.15: Anliza ocene napake pri desetakoračni napovedi.

| model | standardni<br>odklon napake | model: $\sigma_n = 0$ |        | model: $\sigma_n = 0,5$ |        |
|-------|-----------------------------|-----------------------|--------|-------------------------|--------|
|       |                             | učna                  | testna | učna                    | testna |
| (a)   | dejanski                    | 1,95                  | 1,86   | 2,71                    | 2,49   |
|       | ocena                       | 1,28                  | 1,18   | 1,98                    | 1,94   |
| (b)   | dejanski                    | 4,14                  | 3,30   | 3,74                    | 3,68   |
|       | ocena                       | 0,99                  | 0,99   | 1,80                    | 1,77   |

Tudi za primer Helio-crane smo preverili, kakšna je ustreznost metode določitve ocene napake opisane v poglavju 4.4.4. V tabeli 5.14 smo prikazali rezultate za enokoračno napoved. Podamo lahko enake zaključke kot pri modelu Hammerstein. Določitev intervala zaupanja v primeru enokoračne napovedi je ustrezna. V tabeli 5.15 smo prikazali rezultate za desetakoračno napoved, kjer so odstopanja zopet tolikšna, da je opisan pristop neustrezen.

## 5.5 Razvrščanje

Predstavljeno samorazvijajočo se metodo smo preizkusili tudi v primeru razvrščanja. Uporabili smo podatkovno zbirko [36] Univerze Wisconsin, ki vsebuje meritve benignih ter malignih tvorbo na dojkah, in podatkovno zbirko [37], ki vsebuje meritve treh različnih vrst rož roda perunik (ang. *iris*). Obe zbirki sta znani kot merilo za vrednotenje kvalitete razvrščevalnikov.

Za oceno razvrščevalnika smo uporabili metodo z navzkrižnim preverjanjem s  $K$ -timi pregibi (ang. *K-fold cross validation*), kjer množico podatkov razdelimo

Tabela 5.16: Uporabljeni parametri za rojenje.

| $\Gamma_{min}$ | $\kappa_{join}$ | $v_{max}$ | $N_{max}$ |
|----------------|-----------------|-----------|-----------|
| 0,50           | 1,70            | 0,1       | 14        |

na  $K$  podmnožic. Eno podmnožico uporabimo za preizkus razvrščevalnika, ki smo ga naučili s preostalimi  $K - 1$  podmnožic vzorcev. Slednje ponovimo  $K$ -krat, pri čemer vsako od  $K$  podmnožic enkrat obravnavamo na opisan način; končno kvaliteto razvrščevalnika predstavlja povprečje vseh pregibov.

Razvrščevalnik smo učili in uporabili na način, kot je opisan v poglavju 4.5.

### 5.5.1 Razvrščanje boleznij dojk

Podatkovna zbirka [36] obsega 699 vzorcev dojk z 9 značilkami; vzorci pripadajo dvema razredoma. Benignih vzorcev je 458, malignih vzorcev je 241. Pred razvrščanjem smo vrednosti značilk normirali na interval  $[0, 1]$ .

Parametri, ki smo jih uporabili pri rojenju, so prikazani v tabeli 5.16. Za oba razreda smo uporabili enake parametre. Kakovost razvrščevalnika smo ocenili z navzkrižnim preverjanjem z 10-timi pregibi. Uspešnost razvrščanja in primerjava z ostalimi metodami so vidni v tabeli 5.17.

Glede na pridobljene rezultate se nam zdi kakovost našega razvrščevalnika zadovoljiva. Velja omeniti, da so bili razvrščevalniki, s katerimi se primerjamo, posebej prilagojeni za problem razvrščanja boleznij dojk, tako da ni bilo pričakovati, da bi dosegali primerljive rezultate.

### 5.5.2 Razvrščanje perunika

Podatkovna zbirka [37] obsega 150 vzorcev perunika s 4 značilkami; vzorci pripadajo trem razredom. Vsak razred ima 50 vzorcev. Pred razvrščanjem smo vrednosti značilk normirali na interval  $[0, 1]$ .

Parametri, ki smo jih uporabili pri rojenju, so prikazani v tabeli 5.18. Za oba

Tabela 5.17: Uspešnost razvrščanja bolezni dojk.

| Razvrščevalnik   | Uspešnost (%) |
|------------------|---------------|
| Akay [38]        | 99,51         |
| Übeyli [39]      | 99,54         |
| Dheeba [40]      | 93,67         |
| Mert [41]        | 90,49         |
| Nahato [42]      | 98,60         |
| Abdel-Zaher [43] | 99,68         |
| opisana metoda   | 94,43         |

Tabela 5.18: Uporabljeni parametri za rojenje.

| $\Gamma_{min}$ | $\kappa_{join}$ | $v_{max}$ | $N_{max}$ |
|----------------|-----------------|-----------|-----------|
| 0,50           | 2,00            | 0,1       | 7         |

razreda smo uporabili enake parametre. Kakovost razvrščevalnika smo ocenili z navzkrižnim preverjanjem s tremi pregibi. Rezultati pravilnega razvrščanja in primerjava z ostalimi metodami so vidni v tabeli 5.17. Rezultati ostalih metod so bili pridobljeni v [44]; avtor je rezultate ocenil brez navzkrižnega preverjanja, vendar je razdelil zbirko v enakem razmerju.

Glede na pridobljene rezultate se nam zdi kakovost našega razvrščevalnika dobra, saj se primerjamo s splošno uveljavljenimi razvrščevalniki, ki se na področju strojnega učenja uporabljajo za številne probleme.

Tabela 5.19: Uspešnost razvrščanja perunik.

| Razvrščevalnik                  | Uspešnost (%) |
|---------------------------------|---------------|
| BaggingClassifier [45]          | 96,08         |
| GradientBoostingClassifier [46] | 96,08         |
| LogisticRegression [47]         | 100,00        |
| RandomForestClassifier [48]     | 96,08         |
| RidgeClassifier [49]            | 88,24         |
| SVC [50]                        | 98,04         |
| opisana metoda                  | 95,21         |

## 6 Zaključek

V magistrskem delu smo predstavili področje samorazvijajočih se sistemov in njihovih mehanizmov za delovanje, pri čemer smo se osredotočili na sisteme z mehko logiko. Osrednji del naloge je bila predstavitev obstoječega samorazvijajočega sistema eGAUSS+. Sistem temelji na dodajanju rojev glede na vrednost pripadnostne funkcije, ki je določena z Gaussovo funkcijo. Roji se združujejo glede na njihovo prostornino. Metodo smo modificirali in ji dodali dodatni vhodni parameter  $v_{max}$  za omejitev velikosti rojev, ki bistveno izboljša ponovljivost nenadzorovanega rojenja. Sistem smo uporabili na problemih rojenja, identifikacije vhodno-izhodnega modela in razvrščanja.

Metoda se je izkazala za uspešno pri identifikaciji vhodno-izhodnega modela naprave Helio-crane in Hammersteinovega modela, kjer dosegamo dobre rezultate pri simulaciji, enokoračni napovedi in desetkoračni napovedi. Predhodno filtriranje podatkov se je izkazalo za manj ustrezno, saj dosegamo boljšo porazdelitev rojev brez filtriranja, za primerljive rezultate pa potrebujemo večje število rojev. Ocenjujemo tudi, da bi se rojenje ob predhodnem filtriranju še poslabšalo, če bi modelirali sistem višjega reda z večjim številom regresorjev. Prednost metode je tudi računsko enostavna zmožnost določitve intervala zaupanja pri enokoračni napovedi. V nesprotnem načinu smo poskusili določiti tudi interval zaupanja pri desetkoračni napovedi, vendar se je slednje izkazalo za neuspešno.

Metoda je zelo prilagodljiva, vendar slednje hkrati vnaša dodatno zahtevnost tako pri izbiri vhodnih parametrov kot tudi pri izbiri regresorjev. Kot je razvidno iz eksperimentov, potrebujejo različni sistemi različne vhodne parametre za doseganje primernih rezultatov. Za pomembno se je izkazala tudi primerna izbira regresorjev pri izračunu pripadnostih. Identifikacija Hammersteinovega modela

tako dosega boljše rezultate, ko pri izračunu pripadnosti uporabimo zgolj vhodni signal, medtem ko za napravo Helio-crane velja povsem nasprotno in je delovanje boljše z uporabo vseh regresejev. Izkaže se, da je za primerno izbiro parametrov in regresorjev potrebno dobro poznavanje modeliranega sistema, kar dosežemo s predhodnim eksperimentiranjem na učni množici podatkov.

Čeprav je bila metoda razvita predvsem za problem vhodno-izhodne identifikacije, smo ugotovili, da dosega dobre rezultate tudi pri razvrščanju. Na primeru razvrščanja perunik smo pokazali, da je metoda primerljiva s splošno uveljavljenimi razvrščevalniki. Na primeru razvrščanja bolezni dojk se je izkazalo, da metoda ni primerljiva z razvrščevalniki, ki so bili razviti izključno za omenjen problem. Glavna prednost naše metode je, da se jo lahko uporablja tudi v sprotnem načinu, kar ne velja za omenjene razvrščevalnike.

Možnosti nadaljnjega dela za izboljšavo metode gotovo obstajajo. Zanimivo bi bilo ugotoviti primernost delovanja sistema za potrebe vodenja dinamičnih sistemov. Metodo bi lahko izboljšali tudi z določitvijo intervala zaupanja v sprotnem načinu pri večkorakni napovedi.

V sklopu magistrskega dela je bil v programskem okolju MATLAB izdelan tudi grafični vmesnik za dvodimenzionalno rojenje, ki omogoča spremljanje rojenja v realnem času in je predstavljen v prilogi A.

## Literatura

- [1] J. Gantz in D. Reinsel, “The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east,” *IDC iView: IDC Analyze the future*, vol. 2007, no. 2012, str. 1–16, 2012.
- [2] I. Škrjanc, “Cluster-volume based merging concept for incrementally evolving fuzzy gaussian clustering-egauss+,” *IEEE Transactions on Fuzzy Systems*, 2019.
- [3] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak in F. Herrera, “A survey on data preprocessing for data stream mining: Current status and future directions,” *Neurocomputing*, vol. 239, str. 39–57, 2017.
- [4] I. Škrjanc, J. A. Iglesias, A. Sanchis, D. Leite, E. Lughofer in F. Gomide, “Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: a survey,” *Information Sciences*, vol. 490, str. 344–368, 2019.
- [5] I. Škrjanc, *Inteligentni sistemi za podporo odločanju*. 2016.
- [6] D. Brzeziński, “Block-based and online ensembles for concept-drifting data streams,” 2015.
- [7] G. Andonovski, *Samorazvijajoči se sistemi pri spremljanju in vodenju procesov*. PhD thesis, Univerza v Ljubljani, Fakulteta za elektrotehniko, 2018.
- [8] J. Platt, “A resource-allocating network for function interpolation,” *Neural computation*, vol. 3, no. 2, str. 213–225, 1991.
- [9] P. Angelov in N. Kasabov, “Evolving computational intelligence systems,” v *Proceedings of the 1st international workshop on genetic fuzzy systems*, str. 76–82, 2005.

- 
- [10] N. K. Kasabov, “Evolving connectionist systems: Methods and applications in bioinformatics, brain study and intelligent machines,” 2002.
  - [11] L. A. Zadeh, “Fuzzy sets,” *Information and control*, vol. 8, no. 3, str. 338–353, 1965.
  - [12] L.-X. Wang in J. M. Mendel, “Fuzzy basis functions, universal approximation, and orthogonal least-squares learning,” *IEEE transactions on Neural Networks*, vol. 3, no. 5, str. 807–814, 1992.
  - [13] E. H. Mamdani in S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *International journal of man-machine studies*, vol. 7, no. 1, str. 1–13, 1975.
  - [14] T. Takagi in M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE transactions on systems, man, and cybernetics*, no. 1, str. 116–132, 1985.
  - [15] P. Angelov in R. Yager, “A simple fuzzy rule-based system through vector membership and kernel-based granulation,” v *2010 5th IEEE International Conference Intelligent Systems*, str. 349–354, IEEE, 2010.
  - [16] P. Angelov in R. Yager, “Simplified fuzzy rule-based systems using non-parametric antecedents and relative data density,” v *2011 IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS)*, str. 62–69, IEEE, 2011.
  - [17] P. Angelov in R. Yager, “A new type of simplified fuzzy rule-based system,” *International Journal of General Systems*, vol. 41, no. 2, str. 163–185, 2012.
  - [18] E. Lughofer, *Evolving fuzzy systems-methodologies, advanced concepts and applications*, vol. 53. Springer, 2011.
  - [19] D. Dovžan, V. Logar in I. Škrjanc, “Implementation of an evolving fuzzy model (efumo) in a monitoring system for a waste-water treatment process,” *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, str. 1761–1776, 2014.
  - [20] U. Kaymak in R. Babuska, “Compatible cluster merging for fuzzy modeling,” v *Proceedings of 1995 IEEE International Conference on Fuzzy Systems.*, vol. 2, str. 897–904, IEEE, 1995.

- 
- [21] E. Lughofer, J.-L. Bouchot in A. Shaker, “On-line elimination of local redundancies in evolving fuzzy systems,” *Evolving systems*, vol. 2, no. 3, str. 165–187, 2011.
  - [22] E. Lughofer, C. Cernuda, S. Kindermann in M. Pratama, “Generalized smart evolving fuzzy systems,” *Evolving systems*, vol. 6, no. 4, str. 269–292, 2015.
  - [23] A. Bhattacharyya, “On a measure of divergence between two statistical populations defined by their probability distributions,” *Bull. Calcutta Math. Soc.*, vol. 35, str. 99–109, 1943.
  - [24] A. Djouadi, O. Snorrason in F. Garber, “The quality of training sample estimates of the bhattacharyya coefficient,” *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 12, no. 1, str. 92–97, 1990.
  - [25] D. Leite, I. Škrjanc in F. Gomide, “An overview on evolving systems and learning from stream data,” *Evolving Systems*, str. 1–18, 2020.
  - [26] P. P. Angelov in D. P. Filev, “An approach to online identification of takagi-sugeno fuzzy models,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, str. 484–498, 2004.
  - [27] S. L. Chiu, “Fuzzy model identification based on cluster estimation,” *Journal of Intelligent & fuzzy systems*, vol. 2, no. 3, str. 267–278, 1994.
  - [28] E. D. Lughofer, “Flexfis: A robust incremental learning approach for evolving takagi-sugeno fuzzy models,” *IEEE Transactions on fuzzy systems*, vol. 16, no. 6, str. 1393–1410, 2008.
  - [29] E. Lima, M. Hell, R. Ballini in F. Gomide, “Evolving fuzzy modeling using participatory learning,” *Evolving intelligent systems: methodology and applications*, str. 67–86, 2010.
  - [30] G. Klančar in I. Škrjanc, “Evolving principal component clustering with a low run-time complexity for lrf data mapping,” *Applied soft computing*, vol. 35, str. 349–358, 2015.
  - [31] D. Dovžan in I. Škrjanc, “Recursive clustering based on a gustafson-kessel algorithm,” *Evolving systems*, vol. 2, no. 1, str. 15–24, 2011.

- 
- [32] I. Škrjanc, S. Blažič, E. Lughofer in D. Dovžan, “Inner matrix norms in evolving cauchy possibilistic clustering for classification and regression from data streams,” *Information Sciences*, vol. 478, str. 540–563, 2019.
- [33] R. Krishnapuram in J. M. Keller, “A possibilistic approach to clustering,” *IEEE transactions on fuzzy systems*, vol. 1, no. 2, str. 98–110, 1993.
- [34] R. Grča, *Strukturiran problem totalnih najmanjših kvadratov: delo diplom-skega seminarja*. Univerza v Ljubljani, Fakulteta za matematiko in fiziko, 2014.
- [35] A. Zdešar, O. Cerman, D. Dovžan, P. Hušek in I. Škrjanc, “Fuzzy control of a helio-crane,” *Journal of intelligent & robotic systems*, vol. 72, no. 3-4, str. 497–515, 2013.
- [36] W. H. Wolberg in O. L. Mangasarian, “Multisurface method of pattern separation for medical diagnosis applied to breast cytology,” *Proceedings of the national academy of sciences*, vol. 87, no. 23, str. 9193–9196, 1990.
- [37] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of eugenics*, vol. 7, no. 2, str. 179–188, 1936.
- [38] M. F. Akay, “Support vector machines combined with feature selection for breast cancer diagnosis,” *Expert systems with applications*, vol. 36, no. 2, str. 3240–3247, 2009.
- [39] E. D. Übeyli, “Implementing automated diagnostic systems for breast cancer detection,” *Expert systems with Applications*, vol. 33, no. 4, str. 1054–1062, 2007.
- [40] J. Dheeba, N. A. Singh in S. T. Selvi, “Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach,” *Journal of biomedical informatics*, vol. 49, str. 45–52, 2014.
- [41] A. Mert, N. Kılıç, E. Bilgili in A. Akan, “Breast cancer detection with reduced feature set,” *Computational and mathematical methods in medicine*, vol. 2015, 2015.

- 
- [42] K. B. Nahato, K. N. Harichandran in K. Arputharaj, “Knowledge mining from clinical datasets using rough sets and backpropagation neural network,” *Computational and mathematical methods in medicine*, vol. 2015, 2015.
- [43] A. M. Abdel-Zaher in A. M. Eldeib, “Breast cancer classification using deep belief networks,” *Expert Systems with Applications*, vol. 46, str. 139–144, 2016.
- [44] P. Barbiero in A. Tonda, “Fundamental flowers: evolutionary discovery of coresets for classification,” v *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, str. 550–564, Springer, 2019.
- [45] L. Breiman, “Pasting small votes for classification in large databases and on-line,” *Machine learning*, vol. 36, no. 1-2, str. 85–103, 1999.
- [46] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Annals of statistics*, str. 1189–1232, 2001.
- [47] D. R. Cox, “The regression analysis of binary sequences,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, str. 215–232, 1958.
- [48] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, str. 5–32, 2001.
- [49] A. N. Tikhonov, “On the stability of inverse problems,” v *Dokl. Akad. Nauk SSSR*, vol. 39, str. 195–198, 1943.
- [50] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt in B. Scholkopf, “Support vector machines,” *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, str. 18–28, 1998.

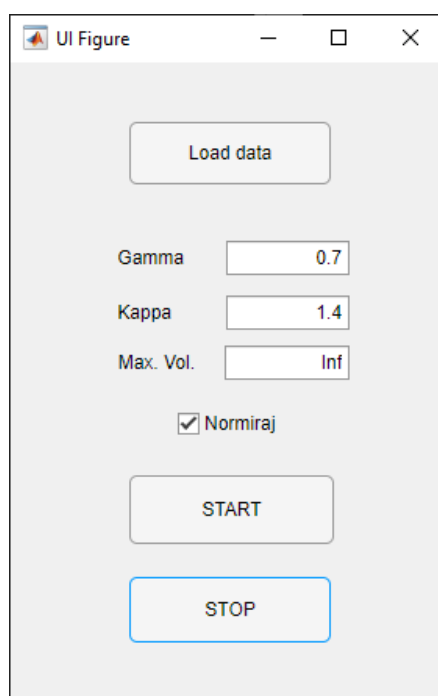


# Dodatek

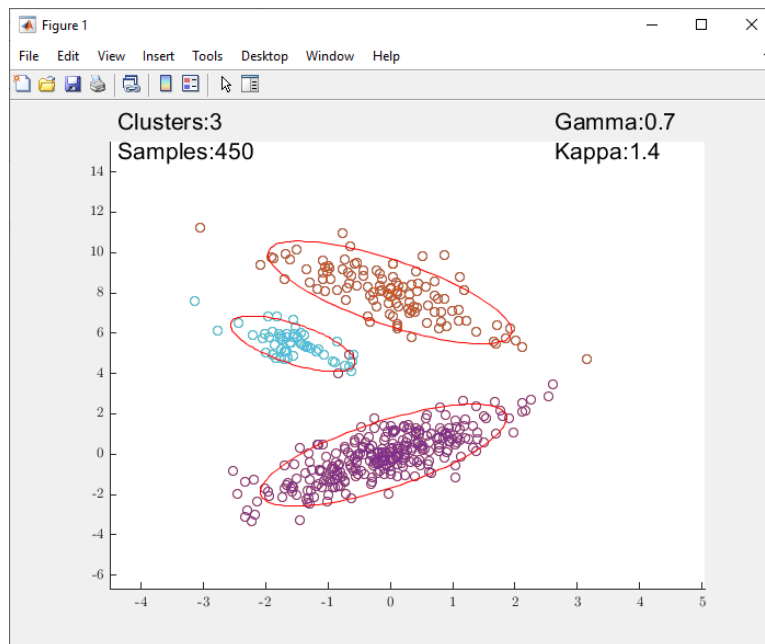


## A Grafični vmesnik za rojenje

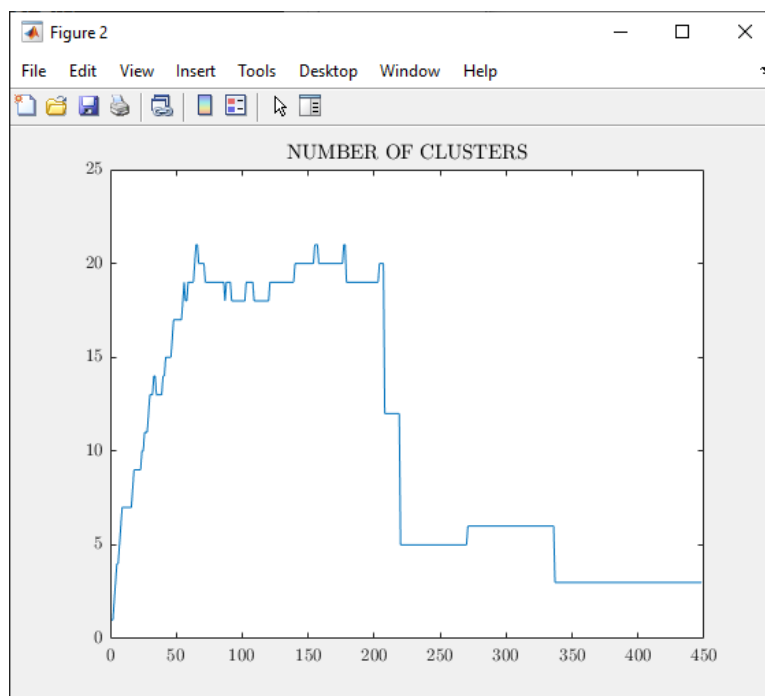
V sklopu magistrskega dela je bil v programskem okolju MATLAB izdelan tudi grafični vmesnik za rojenje dvodimenzionalnih podatkov, ki omogoča spremljanje rojenja v realnem času. Vmesnik omogoča nastavljanje vhodnih parametrov (slika A.1), spremljanje razporeditve prostora (slika A.2) in spremljanje dinamike števila rojev (A.3).



Slika A.1: Grafični vmesnik.



Slika A.2: Okno za opazovanje razporeditve prostora.



Slika A.3: Okno za opazovanje dinamike števila rojev.