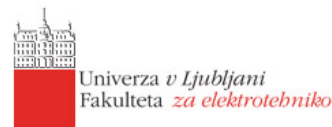


Razvoj nelinearnega modela in prediktivno funkcijskega regulatorja

Umetni inteligentni sistemi za podporo odločanju

Jernej Sabadin



Mentorja: Prof. Dr. Igor Škrjanc, Univ. Dipl. Inž. El., doc. dr. Dejan Dovžan

Predmet: Inteligentni sistemi za podporo odločanju

Datum: 31. Julij 2023

Contents

1	Opis sistema	1
2	Grob opis nevronskega omrežja	2
3	Model nevrona	2
4	Opis triplastnega nevronskega omrežja	2
4.1	Učenje triplastnega perceptrona	2
5	Uporabljeno nevronske omrežje	3
6	Nelinearni model izračunan s pomočjo ANN	4
7	Takagi Sugeno mehki model	5
7.1	Mehko rojenje s c-središči (FCM)	5
8	Identifikacija Takagi Sugeno mehkega modela	6
9	Primerjava modelov	7
10	Pretvorba T-S modela v prostor stanj	7
11	Prediktivno funkcijski regulator (PFC)	7
12	Sledenje referenci	8
13	Dušenje vhodnih motenj	8
14	Reference	9

Ključne besede: Umetne nevronske mreže (ang. Artificial Neural Networks - ANN), Takagi Sugeno mehki model, Rojenje po metodi Gustafson-Kessel (GK), Prediktivno funkcijski regulator (PFC)

1 Opis sistema

Sistem, ki ga identificiramo sestavlja tog kovinski drog, nameščen na vrtljivo os, podobno kot pri nihalu. Ta drog se lahko prosto nihajoče giblje v navpični ravnini, pri čemer je njegovo gibanje omejeno na kot 100 stopinj med tlemi in drogom.

Posebnost te naprave pa je, da je na enem koncu droga nameščen motor z lahkim plastičnim propelerjem. Ta propeler je postavljen pravokotno na drog, kar pomeni, da se s spreminjanjem potiska, ki ga proizvaja propeler, drog dvigne ali spusti. Ko motor ni vključen, je del droga z motorjem v najnižjem položaju.

Motor ima zmožnost vrtenja le v eno smer, zato se smer potiska ne spreminja glede na položaj motorja. S tem sistemom se lahko drog nadzorovano dvigne ali spusti, kar ustvarja možnost preciznega krmiljenja in upravljanja naprave heliocrane.

Sistem matematično opišemo z naslednjo diferencialno enačbo 2. reda:

$$J\ddot{\phi} = T(\phi) - f\dot{\phi} \quad (1)$$

Kjer je ϕ naklon droga, T je vsota vseh navorov v sistemu, J je vztrajnostni moment in f je dušilni faktor.

Statična karakteristika nelinearnega sistema je podana spodaj.

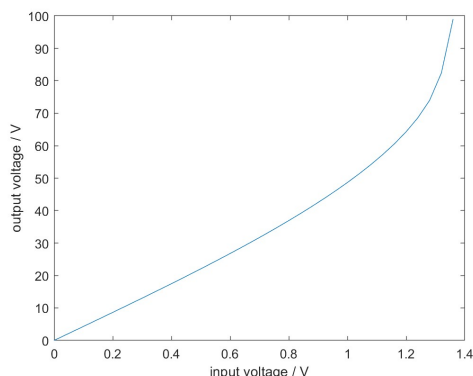


Fig. 1: Statična karakteristika

2 Grob opis nevronskega omrežja

”Umetno nevronske omrežje je umetna oblika človekovih možganov. Ti so se sposobni naučiti novih stvari in se prilagajati na spreminjajoče se okolje. Na primer dojenček prepozna mater po vonju in glasu. Brati zmoremo pisave drugih ljudi. V slabi vidljivosti prepoznamo predmete po njihovi obliki. Zaradi svoje sposobnosti nadziranja telesa, razmišljanja, vizualizacije, sanjanja, domišljanja in učenja so možgani bolj zmogljivi kot najnaprednejši računalnik.

Možgani so sestavljeni iz nevronov, njihove povezave pa tvorijo nevronske omrežje. Biološki nevron je sestavljen iz celičnega telesa, aksona in dendrita. Nevron sprejema signale od drugih nevronov prek dendritov. Ko moč signala preseže določen prag, ta nevron sproži lasten signal, ki se prenese na naslednje nevrone.” [3]

3 Model nevrona

”Gradnik nevronskega omrežja je nevron. Nevron prejme več vhodov in ima en sam izhod. Priključeni vhodi so pomnoženi z utežmi. Uteženi vhodi se seštejejo s pragom nevrona. Prag nevrona predstavlja vrednost, pri kateri se aktivira izhod. Utežena vsota vstopa v aktivacijsko funkcijo, ki definira, kako se ta pretvori v izhod.

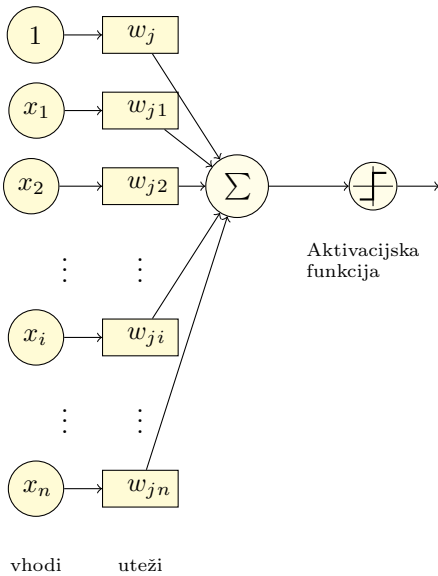


Fig. 2: Slika prikazuje model nevrona

Utež med i -tim in j -tim nevronom predstavimo z w_{ji} . Vsoto vhodov v j -ti nevron in njenega praga w_j zapišemo z izrazom:

$$z_j = \left(\sum_{i=1}^n w_{ji} x_i \right) - w_j \quad (2)$$

Utežena vsota vstopa v nelinearen element oz. aktivacijsko funkcijo. Izhod iz te definiramo kot:

$$y_j = f(z_j) \quad (3)$$

Aktivacijska funkcija je podobna aktivnostim v naših možganih, kjer se različni nevroni sprožijo z različnimi dražljaji. Najpogostejša v praksi je sigmoidalna funkcija.” [3]

4 Opis triplastnega nevronskega omrežja

”V nadaljevanju opišemo triplastno nevronske omrežje, ki ga opišemo s pomočjo knjige Razpoznavanje vzorcev Nikole Pavšiča [4].

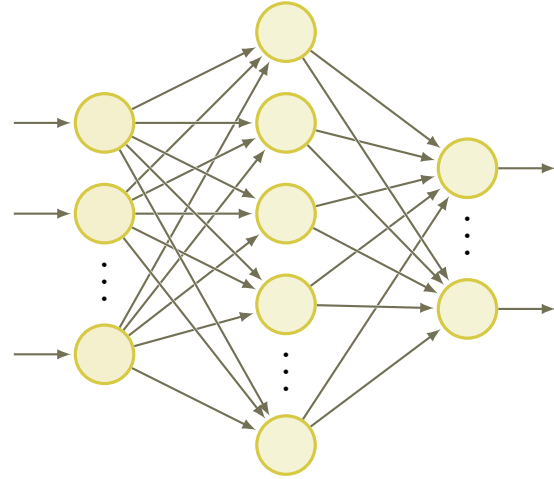


Fig. 3: Shema nevronskega omrežja

Izrazimo j -ti nevron v l -ti plasti z

$$x_j^{(l)} = f \left(\sum_{i=1}^{n_{l-1}+1} w_{ji}^{(l)} x_i^{(l-1)} \right) \quad (4)$$

kjer upoštevamo, da so vhodi $x_{n_{l-1}+1}^{(l-1)} = 1$. Števila nevronov v posamezni plasti predstavimo z n_l . Število $x_i^{(l-1)}$ pa predstavlja i -ti nevron v plasti $l-1$.

Za aktivacijsko funkcijo $f(z)$ uporabimo sigmoidalno funkcijo definirano z:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

Značilke $x_i^{(0)}$ vzorca, ki ga razvrščamo, razvrstimo v razred vzorcev C_i , če za izhode nevronov zadnje plasti velja

$$x_i^{(2)} > x_j^{(2)} \quad \text{za} \quad j = 1, 2, \dots, M; \quad j \neq i, \quad (6)$$

pri čemer je M število vseh razredov.

4.1 Učenje triplastnega perceptrona

V postopku vzvratnega učenja (angl. back-propagation training algorithm) minimiziramo razliko med želenim poznanim $x^{(4)}$ in dejanskim t izhodom. Uteži se nastavijo na take vrednosti, da minimizirajo napako.

Postopek učenja temelji na gradientni metodi. Uteži se skozi postopek učenja popravljajo, dokler se ne ustalijo na določeni vrednosti. Nove vrednosti uteži izračunamo s pomočjo delnih odvodov funkcije napake. To storimo

za vsako plast in vsak učni vzorec. V $k + 1$ koraku popravimo uteži po naslednjem izrazu:

$$w_{ji}(k+1)^{(l)} = w_{ji}^{(l)}(k) - \beta \frac{\partial e(w(k))}{\partial w_{ji}^{(l)}(k)} = w_{ji}^{(l)}(k) + \Delta w_{ji}^{(l)}(k), \quad (7)$$

kjer je $e(w(k))$ napaka nevronskega omrežja v k -ti ponovitvi postopka:

$$e(w(k)) = \frac{1}{2} \sum_{p=1}^{n_2} \left(t_p(k) - x_p^{(2)}(k) \right)^2 \quad (8)$$

Pri tem je t želen odziv omrežja. Označimo ga z vektorjem ničel in eno enko, ki ponazarja dani razred.

Faktor β , ki je manjši od ena, določa korak premika k minimumu funkcije napake. Če ga nastavimo na vrednost 1, se bomo proti minimumu premikali prehitro in ga bomo preskočili. Če pa izberemo premajhnega, se računska zahtevnost učenja poveča.

Učenje nevronskega omrežja poteka tako s pomočjo delnih odvodov napake.” [4]

Za izhodno plast pišemo:

$$\begin{aligned} \frac{\partial e(w(k))}{\partial w_{ji}^{(2)}(k)} &= \frac{\partial e(w(k))}{\partial x_j^{(2)}(k)} \cdot \frac{\partial x_j^{(2)}(k)}{\partial z_j^{(2)}(k)} \cdot \frac{\partial z_j^{(2)}(k)}{\partial w_{ji}^{(2)}(k)} \\ &= \left(t_j(k) - x_j^{(2)}(k) \right) \left(1 - x_j^{(2)}(k) \right) x_j^{(2)}(k) x_i^{(1)}(k) \\ &= -d_j^{(2)}(k) x_i^{(1)}(k) \end{aligned} \quad (9)$$

kjer je

$$d_j^{(2)}(k) = \left(t_j(k) - x_j^{(2)}(k) \right) \left(1 - x_j^{(2)}(k) \right) x_j^{(2)}(k) \quad (10)$$

Popravki uteži $\Delta w_{ji}^{(2)}$ so tako definirani kot:

$$\Delta w_{ji}^{(2)} = \beta d_j^{(2)}(k) x_i^{(1)}(k) \quad (11)$$

popravljeni uteži v plasti $l = 1$ pa so definirani kot:

$$\Delta w_{ji}^{(1)} = \beta d_j^{(1)}(k) x_i^{(0)}(k) \quad (12)$$

za $i = 1, 2, \dots, (n_{l-1} + 1)$, $j = 1, 2, \dots, n_l$ in $l = 1$, kjer je:

$$d_j^{(1)}(k) = \left(1 - x_j^{(1)}(k) \right) x_j^{(1)}(k) \sum_{p=1}^{n_2} d_p^{(2)}(k) w_{pj}^{(2)}(k) \quad (13)$$

Postopek učenja poteka tako, da za vhodne označene podatke izračunamo izhode. Glede na pričakovane izhode in dejanske izračunamo napako ter na podlagi nje popravimo vse uteži. To ponovimo tolikokrat, dokler se uteži ne umirijo.

5 Uporabljeno nevronske omrežje

Umetno nevronske omrežje, ki ga bomo uporabili ima naslednjo strukturo.

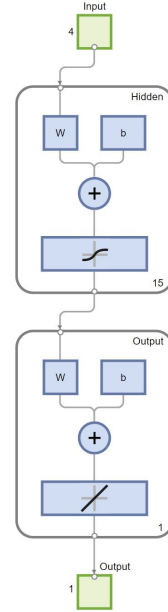


Fig. 4: Model nevronskega omrežja z eno skrito plastjo

Za nelinearni dinamični sistem velja, da so trenutne vrednosti izhodov odvisne tudi od preteklih vrednosti vhodnih in izhodnih signalov, zvezo pa podaja nelinearna funkcija $f(\cdot)$.

$$\hat{y}(k) = f(u(k-1), \dots, u(k-m), y(k-1), \dots, y(k-m)) \quad (14)$$

pri čemer u predstavlja vhodni signal, y izhodni signal ter m red sistema brez zakasnitev, v našem primeru 2.

Enačba (14) ustreza NARX modelu, ki ga zapišemo tudi z spodnjo enačbo:

$$\hat{y}(k) = f(\varphi(k)) \quad (15)$$

kjer so v regresijskem vektorju $\varphi(k)$ shranjeni zakasneni vhodi in izhodi modela. Za NARX model velja:

$$\varphi(k) = [u(k-1) \ \dots \ u(k-m) \ y(k-1) \ \dots \ y(k-m)]^T \quad (16)$$

Opisujemo dva pristopa s katerima naučimo in preizkusimo nevronske omrežje:

• Učenje

V fazi učenja pripeljemo na vhod nevronskega omrežja zakasnjene vrednosti vhodov in izhodov. Nevronske omrežje bo na podlagi vhodnega vektorja $[u(k-1), u(k-2), y(k-1), y(k-2)]^T$ v vsakem časovnem trenutku k posebej optimiralo svoje uteži, ki so parametri nelinearne preslikave $f(\cdot)$, tako da bo $\hat{y}(k)$ najbližje izmerjeni vrednosti $y(k)$. Pristopu računanja $\hat{y}(k) = f(u(k-1), \dots, u(k-m), y(k-1), \dots, y(k-m))$.

1), $u(k-2), y(k-1), y(k-2)$) pravimo enokoračna predikcija.

Naše nevronske omrežje ima 4 nevrone v vhodni plasti, ki ustrezajo vektorju $\varphi(k) = [u(k-1), u(k-2), y(k-1), y(k-2)]^T$ in en nevron v izhodni plasti, ki ustreza $\hat{y}(k)$. V vmesni plasti pa imamo 15 nevronov.

Nelinearna aktivacijska funkcija v prikriti plasti je tangent hiperbolična funkcija, ali 'tansig'. Matematično jo opišemo z naslednjo enačbo:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (17)$$

V izhodni plasti ANN uporabimo linearno aktivacijsko funkcijo.

Za optimizacijo uteži uporabimo

Levenberg-Marquardtovo metodo, ki je nelinearna lokalna optimizacijska metoda za reševanje problema nelinearnih najmanjših kvadratov. Je iterativna metoda, ki se pogosto uporablja pri učenju v nevronskih omrežjih.

Levenberg-Marquardtova metoda kombinira dve optimizacijski metodi: metodo največjega spusta (ki je bolj robustna in stabilna, a konvergira počasi) in Gauss-Newtonovo metodo (ki hitro konvergira v bližini optimalne rešitve, vendar ni tako stabilna). Levenberg-Marquardtova metoda poskuša doseči ravnovesje med obema pristopoma, s čimer zagotavlja stabilnost in hitro konvergenco.

• Simulacija

Preverjanje obnašanja dinamičnega sistema opravimo z večkratno predikcijo oziroma z simulacijo, kar pomeni, da neznane vrednosti izhodnega signala zamenjamo z njihovimi napovedmi. Kar ponazorimo z enačbo

$$\hat{y}(k+1) = f(\hat{y}(k), \hat{y}(k-1), \dots, u(k), u(k-1), \dots) \quad (18)$$

6 Nelinearni model izračunan s pomočjo ANN

Za identifikacijski vhodni signal izberemo APRBS.

APRBS, ali Amplitude-Modulated Pseudo Random Binary Sequence, je signal, ki je še posebej koristen za identifikacijo nelinearnih sistemov. Gre za obliko PRBS signala, ki vključuje dodatno amplitudno modulacijo, kar pomeni, da se amplituda signala v času spreminja na pseudo-naključen način.

Ključna lastnost APRBS signalov je, da vključujejo širok spekter frekvenčnih komponent, kar pomeni, da z njim "raziskujemo" sistem pri različnih frekvencah. Poleg tega, ker je APRBS amplitudno moduliran PRBS, lahko sistem "raziskujemo" v različnih obratovalnih točkah ali amplitudah.

Za nelinearne sisteme, ki se lahko obnašajo drugače pri različnih obratovalnih pogojih, je ta lastnost zelo koristna. Omogoča, da se sistem preuči in identificira v

širokem območju obratovalnih pogojev, kar zagotavlja boljše razumevanje celotne dinamike sistema.

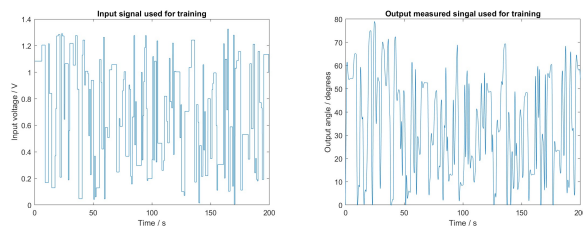


Fig. 5: Vhodni signal za identifikacijo (levo) in odziv naprave (desno)

Po učenju ANN simuliramo naše omrežje z levim signalom na sliki 5.

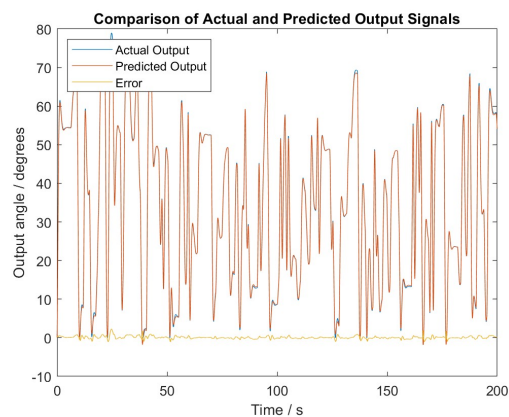


Fig. 6: Simuliran in dejanski odziv na vhodni učni signal. MAE (Mean of absolute error) znaša 0.25

Naučeno nevronske omrežje simuliramo tudi na testnem signalu

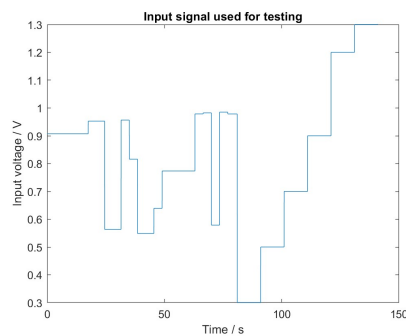


Fig. 7: Testni signal

in prikazemo simuliran in merjen odziv

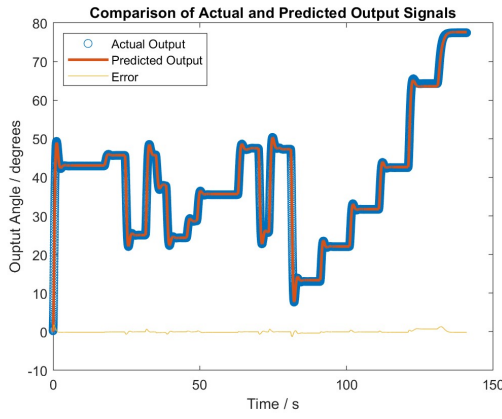


Fig. 8: Simuliran in dejanski odziv na vhodni testni signal. MAE (Mean of absolute error) znaša 0.19

7 Takagi Sugeno mehki model

Takagi-Sugeno mehki model je pristop za modeliranje nelinearnih sistemov, ki uporablja tehniko fuzzy logike. Sistem je sestavljen iz več pravil v obliki "IF-THEN", ki skupaj opisujejo dinamiko sistema.

Vsako pravilo ima svojo pripadajočo funkcijo pripadnosti, ki izraža, kako močno je dan podatek povezan s tem pravilom. Potem se izhod vsakega pravila uteži glede na to stopnjo pripadnosti in vsota teh uteženih izhodov je končni izhod sistema.

Za namene našega modela uporabimo naslednji set mehkih pravil:

$$\begin{aligned} &\text{ČE } y(k-2) \text{ v } A_2^j \text{ IN } y(k-1) \text{ v } A_1^j \\ &\text{POTEM} \\ &y(k) = f_j(u(k-2), u(k-1), y(k-2), y(k-1)). \end{aligned} \quad (19)$$

ČE deli mehkih pravil opisujejo regije v prostoru vhodnih spremenljivk. A_i^j predstavljajo mehke sete, ki jih opišemo z funkcijami pripadnosti (slika 10). POTEM deli pa opisujejo utežene linearne modele.

Takagi Sugeno mehki model za modeliranje nelinearnih dinamičnih sistemov sestavlja več lokalnih linearnih modelov, le ti pa so veljavni na določenih področjih. Linearne modele povežemo med seboj z normiranimi pripadnostnimi funkcijami Φ_i .

$$\Phi_i(u) = \frac{\mu_i(\underline{u})}{\sum_{j=1}^M \mu_j(\underline{u})} \quad (20)$$

Kjer je \underline{u} vhodni vektor sestavljen iz $[y(k-2), y(k-1)]$, $\mu_i(\underline{u})$ pa dvodimenzionalna funkcija pripadnosti, ki jo dobimo kot t-normo produkta mehkih funkcij in predstavlja rezultat konjunkcije IN v mehkih pravilih. $\mu_i(\underline{u})$ ustreza i -temu pravilu in meri kako dobro dan vektor u ustreza temu pravilu. Vsako pravilo potem prispeva k končnemu izhodu y , sorazmerno z njegovo stopnjo pripadnosti. Takagi-Sugeno model je torej oblike:

$$\hat{y} = \sum_{i=1}^M \hat{y}_i \Phi_i(\underline{u}) \quad (21)$$

kjer je \hat{y}_i izhod i -tega linearnega modela in M število vseh pravil. $(\hat{y}_i(k) = -a_{i1}y(k-1) - \dots - a_{im}y(k-m) + b_{i1}u(k-1) + \dots + b_{im}u(k-m) + C_i)$.

Za pripadnostne funkcije si izberemo normirane Gaussove funkcije. V splošnem večdimenzionalna funkcija $\mu_i(\underline{u})$ pa je potem oblike:

$$\mu_i(\underline{u}) = \prod_{j=1}^p e^{-\frac{(u_j - c_{ij})^2}{2\sigma_{ij}^2}} \quad (22)$$

kjer predstavljajo c_{ij} središča i -tega pravila in σ_{ij} standardne deviacije i -tega pravila, p pa dimenzijo vektorja \underline{u} , ki je v našem primeru 2.

Ko imamo znane vrednosti središč c_{ij} in standardnih deviacij σ_{ij} , določimo parametre linearnih modelov po metodi lokalnega ocenjevanja parametrov. Omenjena metoda obravnava podatke znotraj določenega področja povsem ločeno in zanemari interakcije z ostalimi modeli. Parametre θ_i vsakega izmed M modelov določimo posamično s pomočjo metode uteženih najmanjših kvadratov, kjer je utež enaka veljavnostni funkciji

$$I_i = \sum_{k=1}^N \Phi_i(\underline{u}(k)) e^2(k) \rightarrow \min_{\theta_i} \quad (23)$$

vektor parametrov $\theta_i = [a_{i1}, \dots, a_{im}, b_{i1}, \dots, b_{im}, C_i]^T$ i -tega modela poiščemo z reševanjem matrične enačbe

$$\theta_i = (\Psi^T Q_i \Psi)^{-1} \Psi^T Q_i Y \quad (24)$$

kjer je matrika Q_i diagonalna matrika z elementi $\Phi_i(u(k))$ za $k = 1, 2, \dots, N$ po diagonali, kjer je N dolžina vhodnega signala. Matrika Ψ pa predstavlja regresijsko matriko oblike

$$\underline{\Psi}^T = [-y(k-1), \dots, -y(k-m), u(k-1), \dots, u(k-m), 1] \quad (25)$$

kjer je $y(k)$ merjeni izhodni vektor in $u(k)$ vhodni vektor. Konstanta 1 je dodana za oceno odmika C linearnega modela.

Glavni problem Takagi sugeno metode je delitev vhodno-izhodnih podatkov na področja. V ta namen uporabimo algoritem mehkega rojenja z c -središči in pa GK algoritem rojenja, s katerima dobimo centre c_{ij} in standardne deviacije σ_{ij} vsakega področja.

7.1 Mehko rojenje s c -središči (FCM)

Fuzzy C-Means (FCM) je algoritem za razvrščanje podatkov v skupine (roje), kjer ima vsak podatek določeno stopnjo pripadnosti vsaki skupini. Ta pristop je drugačen od tradicionalnih metod razvrščanja, kjer vsak podatek pripada samo eni skupini. Stopnja pripadnosti vzorca vsakemu roju se nahaja med 0 (ničelna pripadnost) in 1 (popolna pripadnost). FCM predpostavlja, da so skupine podatkov v obliki sferičnih rojev. Algoritem se, da tudi nadgraditi, da temu ne bi bilo tako, to pa se doseže z upoštevanjem mehke kovariančne matrike in matrike notranjega produkta, ki pomaga določiti, kako daleč je vsaka točka podatkov od centra določenga roja, ob upoštevanju oblike in velikosti roja. Tako se potem

roji lahko pojavljajo v različnih oblikah in velikostih ter ne le kot hiperkroglice

FCM algoritem sledi naslednji psevdokodi [2]:

1. Inicializacija števila rojev c
2. Izbira uteži η
3. Izbira tolerančnega praga za zaključek iteracije min_{impro} in izbira maksimalnega števila iteracij r_{max}
4. Inicializacija matrike pripadnosti $U^0 \in M$ z naključnimi vrednostmi
5. **ponavlja**
6. Izračun središč rojev: $\underline{v}_i^{(r)} = \frac{\sum_{k=1}^n (\mu_{ik}^{(r)})^\eta \underline{x}(k)}{\sum_{k=1}^n (\eta_{ik}^{(r)})^\eta}$
7. Izračun razdalj:
 $d_{ik}^2 = (\underline{x}(k) - \underline{v}_i^{(r)})^T (\underline{x}(k) - \underline{v}_i^{(r)}), i = 1, \dots, c,$
 $k = 1, \dots, n$
8. Izračun matrike pripadnosti:
 $\mu_{ik}^{(r)} = \frac{\frac{1}{d_{ik}^w}}{\sum_{j=1}^c \frac{1}{d_{ij}^w}}, w = \frac{2}{\mu - 1}$
9. **dokler** $\|\underline{U}^{(r)} - \underline{U}^{(r-1)}\|_\infty < min_{impro}$
ali $r \geq r_{max}$

\underline{x} predstavlja točke v prostoru, ki jih poskušamo razdeliti v skupine. V našem primeru bodo to pari vrednosti vhodnega in izhodnega signala.

Koraki (6-9) predstavljajo zanko, v kateri algoritem iterativno izboljšuje skupine.

1. Izračun središč rojev: Ta korak določa središče vsakega roja.
2. Izračun razdalj: Ta korak izračuna razdalje med posameznimi vzorci in središči rojev.
3. Izračun matrike pripadnosti: Na podlagi izračunanih razdalj se posodobi matrika pripadnosti.
4. Algoritem se ponavlja, dokler ne doseže zgoraj navedenih pogojev za ustavitev.

Mehko kovariančno matriko izračunamo s pomočjo izraza

$$\underline{\Sigma}_i = \frac{\sum_{k=1}^n (\mu_{ik}^{(r-1)})^\eta (\underline{x} - \underline{v}_i)(\underline{x} - \underline{v}_i)^T}{\sum_{k=1}^n (\eta_{ik}^{(r-1)})^\eta}, i = 1, \dots, c \quad (26)$$

8 Identifikacija Takagi Sugeno mehkega modela

Vhodni signal, ki ga uporabimo za identifikacijo je sestavljen iz stopnic, ki so v območju višjih napetosti, ki ustreza "večji" nelinearnosti v statični karakteristiki porazdeljene bolj skupaj. Na koncu vsake stopnice pa imamo APRBS signal, da zajamemo boljše tudi višje frekvence signala.

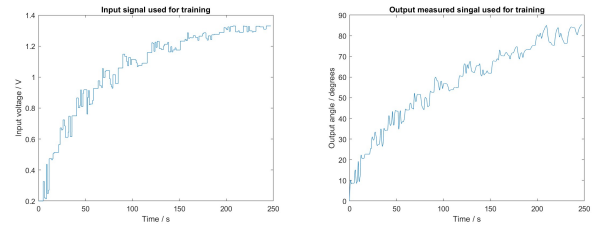


Fig. 9: Vhodni signal za identifikacijo (levo) in odziv naprave (desno)

Z obliko vhodnega signala dosežemo, da bo FCM algoritem lažje našel področja tudi v delih večje nelinearnosti. Spodaj prikažemo pripadnostne funkcije, ki nastopajo v premisah mehkih pravil v enačbi 19 in ki jih potem združimo s t-normo produkta. Centre in standardne deviacije poiščemo z FCM metodo rojenja.

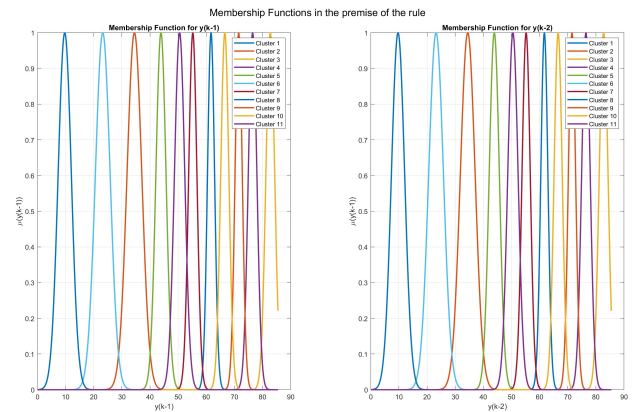


Fig. 10: Pripadnostne funkcije, katerih centre in standardne deviacije pridobimo z metodo FCM

Algoritem FCM preizkusimo pri številu rojev $c = 11$ in faktorju uteži $\eta = 2$.

Spodaj prikažemo tudi izgled večdimenzionalne pripadnostne funkcije, ki sledi pravilom iz enačbe 19.

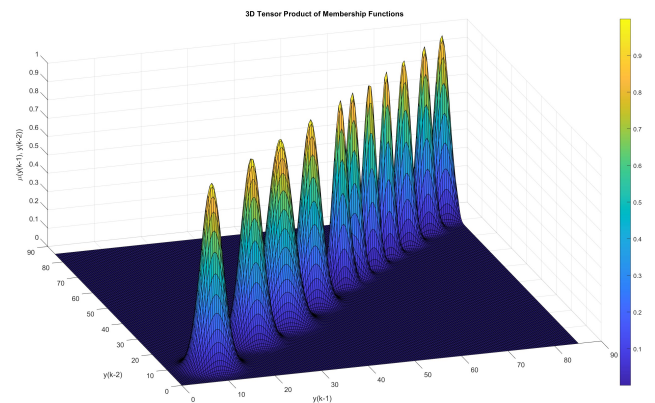


Fig. 11: Več dimenzionalna pripadnostna funkcija, ki sledi pravilom iz enačbe 19

Po izračunu parametrov linearnih modelov po metodi

uteženih najmanjših kvadratov, simuliramo odziv na učni signal in dobimo naslednje rezultate:

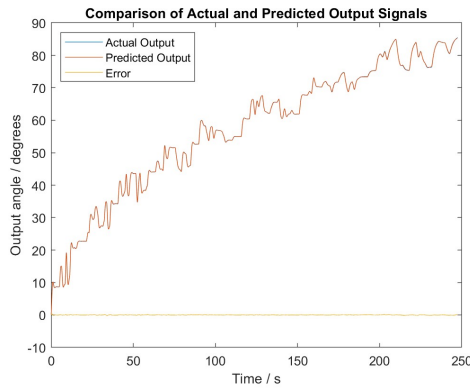


Fig. 12: Simuliran in dejanski odziv na vhodni učni signal. MAE znaša 0.06

Simuliran odziv na testni vhodni signal iz slike 7 prikažemo na spodnji sliki

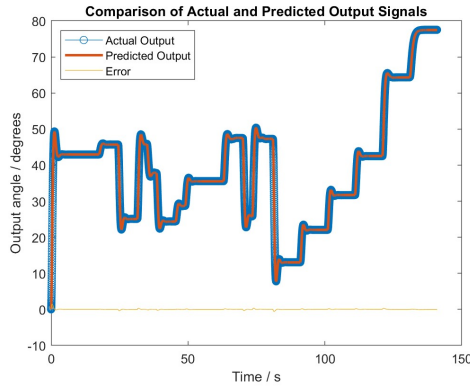


Fig. 13: Simuliran in dejanski odziv na vhodni testni signal. MAE(Mean of absolute error) znaša 0.06.

9 Primerjava modelov

T-S modeli so vrsta fuzzy modelov, ki uporabljajo skupino linearnih modelov, ki so uteženi z fuzzy logiko. Ta pristop je preprost in intuitiven. Prav tako so T-S modeli analitični, kar pomeni, da lahko preučimo, kako vplivi vhodnih spremenljivk vplivajo na izhod.

ANN so močne in prilagodljive metode za modeliranje nelinearnih dinamičnih sistemov. ANN lahko modelira kompleksne nelinearne odvisnosti. Vendar pa so ANN pogosto "črne škatle", kar pomeni, da je težko razumeti, kako posamezne vhodne spremenljivke vplivajo na izhod. Prav tako je za uspešno uporabo ANN potrebno ustrezno nastaviti številne hiperparametre, kot so število plasti, število nevronov v vsaki plasti, hitrost učenja itd.

Primerjava med dvema metodama je odvisna od specifičnega problema. Če je pomembno razumeti, kako vhodne spremenljivke vplivajo na izhod, so T-S modeli boljša izbira. Če pa je na voljo veliko podatkov in je glavni cilj napovedna natančnost, je ANN boljša izbira.

Modela primerjamo med sabo in vidimo, da je T-S model boljši na testni zbirki. Sicer, bi se lahko ANN omrežje odrezalo bolje z dodajanjem večjega števila nevronov in spreminjanjem hiperparametrov. Priporočljiva je tudi normalizacija podatkov.

10 Pretvorba T-S modela v prostor stanj

Za namen prediktivnega vodenja je potrebno T-S model pretvoriti v prostor stanj.

vektor parametrov i -tega linearnega modela zapišemo kot $\theta_i = [a_{i1}, \dots, a_{im}, b_{i1}, \dots, b_{im}, C_i]^T$. Diferenčna enačba kateri ustreza pa je

$$\hat{y}_i(k) = -a_{i1}y(k-1) - \dots - a_{im}y(k-m) \quad (27)$$

$$+ b_{i1}u(k-1) + \dots + b_{im}u(k-m) + C_i \quad (28)$$

Za naš primer kjer je red sistema $m = 2$ pišemo

$$\hat{y}_i(k) = -a_{i1}y(k-1) - a_{i2}y(k-2) + b_{i1}u(k-1) + b_{i2}u(k-2) + C_i \quad (29)$$

i -ti linearni model zapišemo v SOKO (Spoznativna observatorska kanonična oblika) obliki kjer dodamo matriko R_i .

$$x_i(k+1) = \begin{bmatrix} 0 & -a_{i2} \\ 1 & -a_{i1} \end{bmatrix} \begin{bmatrix} x_{i1}(k) \\ x_{i2}(k) \end{bmatrix} + \begin{bmatrix} b_{i2} \\ b_{i1} \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ C_i \end{bmatrix} \quad (30)$$

$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i1}(k) \\ x_{i2}(k) \end{bmatrix} \quad (31)$$

oziroma

$$x_i(k+1) = A_i x_i(k) + B_i u(k) + R_i \quad (32)$$

$$y_i(k) = C_i x_i(k) \quad (33)$$

Linearne modele pa nato združimo v T-S model po enačbi

$$A_m = \beta_1 A_1 + \dots + \beta_c A_c \quad (34)$$

$$B_m = \beta_1 B_1 + \dots + \beta_c B_c \quad (35)$$

$$C_m = \beta_1 C_1 + \dots + \beta_c C_c \quad (36)$$

$$R_m = \beta_1 R_1 + \dots + \beta_c R_c \quad (37)$$

$$x(k+1) = A_m x(k) + B_m u(k) + R_m \quad (38)$$

$$y(k) = C_m x(k) \quad (39)$$

Kjer parametri β_i predstavljajo normirane pripadnostne funkcije.

11 Prediktivno funkcijski regulator (PFC)

Prediktivno vodenje (Model Predictive Control - MPC) je tehnika krmiljenja, ki uporablja model sistema za napovedovanje prihodnjih izhodov sistema. Na podlagi teh napovedi se izračuna optimalna zaporedja vhodnih signalov, ki minimizirajo napako med napovedjo in

želeno vrednostjo, ob upoštevanju morebitnih omejitev sistema.

PFC algoritem je dejansko PI regulator oz. PI regulator z Smithovim prediktorjem v primeru, ko upoštevamo mrtvi čas procesa. PFC deluje dobro tudi, ko nelinearen proces opišemo z množico lokalnih linearnih modelov v obliki Takagi Sugeno mehkega modela.

[1] PFC uporablja model procesa za predikcijo izhoda procesa na določenem horizontu H . Določiti je potrebno takšen regulirni signal u , da se ujemata razliki med napovedjo referenčne vrednosti $y_r(k+H)$ in trenutno vrednostjo izhoda procesa $y_p(k)$ ter napovedjo izhoda modela $y_m(k+H)$ in $y_m(k)$. Torej

$$y_r(k+H) - y_p(k) = y_m(k+H) - y_m(k) \quad (40)$$

Po izpeljavi dobimo naslednji regulirni zakon zapisan v prostoru stanj:

$$u(k) = G(w(k) - y_p(k)) + G_0^{-1}y_m(k) + \quad (41)$$

$$\dots - G_0^{-1}C_m A_m^H x_m(k) - B_m^{-1}R_m \quad (42)$$

kjer je

$$G = G_0^{-1}(I - A_r^H) \quad (43)$$

in G_0

$$G_0 = C_m(A_m^H - I)(A_m - I)^{-1}B_m \quad (44)$$

Matrika A_r pripada referenčnemu modelu, ki ima ojačanje 1, $w(k)$ pa je predpisana referenčna vrednost (ang. setpoint trajectory).

12 Sledenje referenci

Horizont napovedi izberemo 2 vzorca, pri čemer je čas med vzorcema $T_s = 0.01s$. Matrike v prostoru stanj, ki pripadajo referenčnemu modelu, dobimo preko pretvorbe prenosne funkcije zveznega sistema $G_r(s) = \frac{3}{s+3}$ z ZOH metodo v diskretno obliko in potem v prostor stanj.

Spodaj prikažemo rezultate sledenja referenci

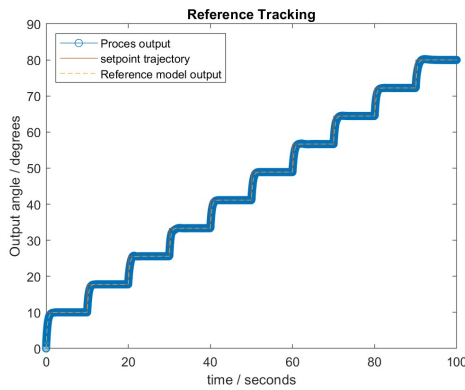


Fig. 14: Sledenje referenci

regulirni signal:

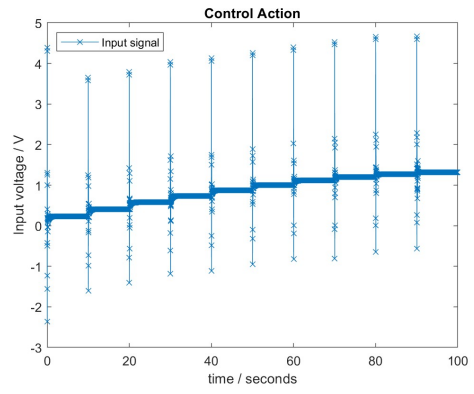


Fig. 15: Regulirni signal $u(k)$

Opazno je hitro in natančno sledenje referenčnemu signalu.

Napaka v ustaljenem stanju je zanemarljivo majhna. Opazno je tudi natančno sledenje v področju velike nelinearnosti. Vhodni signal se prav tako zelo hitro umiri na konstantno vrednost, kar je zaželeno. V nasprotnem bi lahko prišlo do težav kot so:

- Obraba opreme: Hitre spremembe vhodnega signala lahko povzročijo hitrejšo obrabo in potencialno poškodujejo sistemsko opremo.
- Neefektivnost: Sistem, ki se nenehno prilagaja skokovitim spremembam, morda ne deluje tako učinkovito kot bi lahko. To lahko pripelje do večje porabe energije.

13 Dušenje vhodnih motenj

Krmilnik preizkusimo pri dušenju vhodnih motenj amplitude 0.5. To storimo pri konstantni referenci 20, 40 in 60 stopinj.

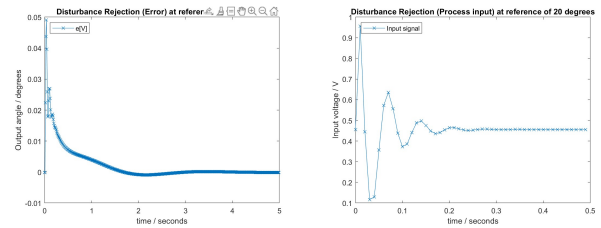


Fig. 16: Razlika med procesom in referenčno vrednostjo 20 stopinj levo in regulirni signal $u(k)$ desno

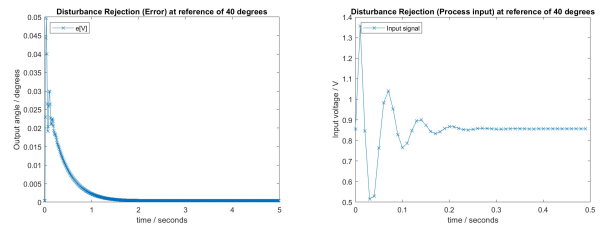


Fig. 17: Razlika med procesom in referenčno vrednostjo 40 stopinj levo in regulirni signal $u(k)$ desno

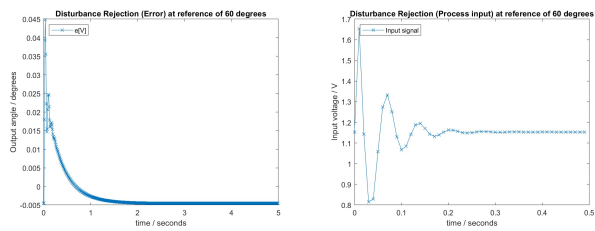


Fig. 18: Razlika med procesom in referenčno vrednostjo 60 stopinj levo in regulirni signal $u(k)$ desno

14 Reference

- 1 I. Škrjanc, Knjiga - Inteligentni sistemi za podporo odločanju, FE 2023
- 2 I. Škrjanc, Inteligentni sistemi za podporo odločanju, gradivo za predavanja, FE 2023
- 3 J. Sabadin, Diplomsko delo
- 4 N. Pavešić, 'Razpoznavanje vzorcev: Uvod v analizo in razumevanje vidnih in slušnih signalov', 3. izd., Ljubljana: Založba FE in FRI, 2012.