

Mehko prediktivno funkcijsko vodenje modela laboratorijske naprave polhelikopter

Martin Porenta

Mentor: as. dr. Goran Andonovski

Univerza v Ljubljani, Fakulteta za elektrotehniko

Tržaška cesta 25, 1000 Ljubljana

porenta.martin@gmail.com

Fuzzy predictive functional control based on model of Helio-crane pilot plant

In this paper design procedure and results of fuzzy predictive functional control are presented. Whole procedure was performed in two steps. In the first step fuzzy model of the system is developed. In the second step, fuzzy model is used for predictive functional control. For identification of fuzzy model we used online evolving cloud based algorithm. For model identification we defined four different excitation signals and for each signal we determined fuzzy model of the system. With validation process the quality of each model was determined. Predictive functional control was conducted with different parameters and for each parameter we observed result of the predictive control. We also experimented with different fuzzy models and observed how quality of the model effects result of closed loop control.

Kratek pregled prispevka

V članku smo predstavili postopek načrtovanja in rezultate vodenja z mehkim prediktivnim regulatorjem. Načrtovanje vodenja smo izvedli v dveh korakih: v prvem koraku smo z identifikacijo določili mehki model, ki smo ga v drugem koraku uporabili za vodenje. Za identifikacijo smo uporabili samorazvijajoči se algoritem na osnovi oblakov. Identifikacijo smo izvajali pri štirih različnih vzbujačnih signalih in za vsak tip vzbujanja določili mehki model. Kvaliteto določenih modelov smo ocenili s postopkom validacije, kjer smo sistem in model vzbujali s testnim signalom, izmerili odzive in izračunali koren srednje kvadratne napake (RMSE) med odzivom modela in sistema. Prediktivno funkcijsko vodenje smo izvajali pri različnih vrednostih parametrov in opazovali vplive na rezultat vodenja. Prav tako smo vodenje izvedli z različni modeli in opazovali, kako kvaliteta mehkega modela vpliva na rezultat vodenja.

1 Uvod

V praksi imamo dostikrat opravka z izrazito nelinearnimi sistemi. V tem primeru ne moremo s klasičnimi algoritmi (npr. PID regulatorji) zagotoviti dobrih rezultatov vodenja čez celotno delovno območje sistema. Možen pristop k rešitvi omenjenega problema je prediktivni funkcijski regulator na osnovi mehkega modela. Pri tem pristopu obravnavani nelinearni sistem modeliramo z mehkim modelom in na podlagi mehkega modela izvedemo prediktivno vodenje sistema.

Načrtovanja vodenja z mehkim prediktivnim funkcijskim regulatorjem smo se lotili v dveh delih. V prvem delu bomo predstavili teoretične osnove mehkih modelov in postopek identifikacije mehkega modela. V drugem bomo predstavili delovanje prediktivnega funkcijskega regulatorja in predstavili rezultate vodenja na osnovi mehkega modela iz prvega dela.

2 Identifikacija mehkega modela

V tem razdelku bodo na kratko predstavljene teoretične osnove mehke identifikacije, rezultati identifikacije in rezultati validacije določenih modelov.

Identifikacijo izvajamo v več korakih. Najprej je potrebno z ustreznim signalom vzbujati sistem in zajeti ustrezne meritve. Nato na meritvah izvajamo algoritem mehke identifikacije in določimo mehke modele. Na koncu sledi še validacija določenih mehkih modelov.

2.1 Teoretične osnove mehke identifikacije

Pri mehkiem modeliranju lahko nelinearni sistem opišemo z množico lokalnih linearnih modelov, ki jih definiramo z naslednjo enačbo:

$$y_{mi}(k) = b_{1i}u(k) + a_{1i}y_{mi}(k-1) + a_{2i}y_{mi}(k-2) + r_i \quad (1)$$

kjer je y_{mi} izhod i -tega lokalnega linearnega modela, u vhodni signal in $a_{1i}, a_{2i}, b_{1i}, r_i$ parametri lokalnega linearnega modela. Ker je

največja zakasnitev v enačbi (1) enaka 2, pomeni da imamo opravka z modelom drugega reda.

Enačbo lokalnega linearnega modela (1) v vektorski obliki zapišemo kot:

$$f_i(k) = \underline{\theta}_i \underline{\psi}_i(k) \quad (2)$$

kjer $\underline{\psi}_i(k)$ predstavlja regresor lokalnega linearnega modela v trenutku k in $\underline{\theta}_i$ vektor parametrov lokalnega linearnega modela [1]. Definicijo regresorja prikazuje enačba (3):

$$\underline{\psi}_i = [y_{mi}(k-1), y_{mi}(k-2), u(k), 1] \quad (3)$$

Izhod celotnega modela v trenutku k izračunamo s seštevanjem prispevkov lokalnih linearnih modelov, kot to opisuje naslednja enačba:

$$y_m = \sum_{i=1}^c \beta_i(x_f) \underline{\theta}_i^T \underline{\psi}_i(k) = \underline{\beta}(x_f) \underline{\theta}^T \underline{\psi}(k) \quad (4)$$

kjer $\beta_i(x_f)$ predstavlja pripadnostno funkcijo regresorja x_f k i -temu lokalnemu modelu. Regresor x_f predstavlja regresor rojenja in ga uporabimo za izračun pripadnostnih funkcij vhodnega podatka k posameznem roju. Tako imamo dva regresorja: x_f za izračun pripadnostnih funkcij in $\underline{\psi}_i$ za izračun izhoda modela [1],[2].

Pripadnostne funkcije določimo tako, da v izmerjenih podatkih poiščemo roje, ki so definirani s centrom in varianco. Vsakemu roju pripada en lokalni linearni model (enačba (1)). Za iskanje rojev smo uporabili »samorazvijajoči se algoritem na osnovi oblakov« (ang. Online evolving cloud based algorithm). Delovanje algoritma je bolj podrobno razloženo v članku [2].

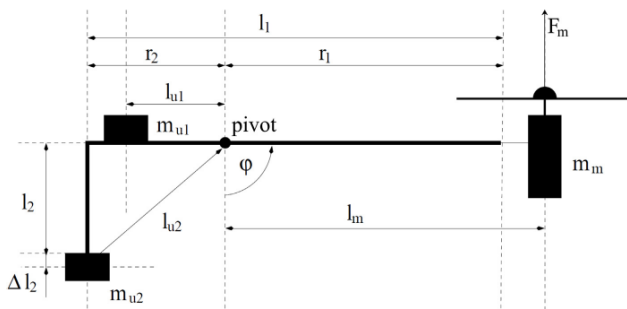
Preden začnemo izvajati algoritem, moramo definirati

- maksimalno število rojev c_{max} ,
- maksimalno lokalno gostoto γ_{max} ,
- regresor rojenja: x_f ,
- regresor lokalnih linearnih modelov: $\underline{\psi}$.

Po končanem izvajanju nam algoritem vrne centre in variance rojev, ki jih uporabimo za izračun pripadnostnih funkcij $\beta(x_f)$. Nato z metodo rekurzivnih uteženih najmanjših kvadratov (rWLS), kjer za uteži izberemo pripadnostne funkcije $\beta(x_f)$, določimo parametre lokalnih linearnih modelov (definirani z enačbo (1)).

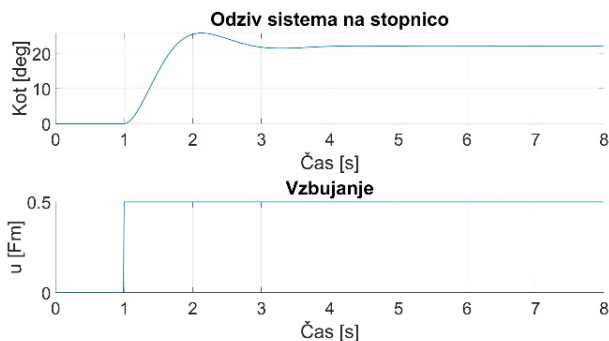
2.2 Lastnosti obravnavanega sistema

Identifikacijo in vodenje smo izvajali na modelu laboratorijske naprave polhelikopter. Shematski prikaz sistema prikazuje slika 1.



Slika 1: Shematski prikaz sistema polhelikopter

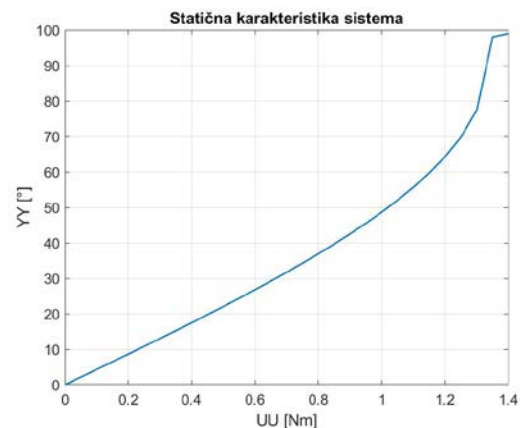
Kot lahko vidimo iz slike 1, ima sistem eno prostorsko stopnjo (vrti se okoli ene osi), na eni strani ima protiutež, na drugi strani pa motorček s propelerjem. Sistem kot vhod sprejme silo vzgona propelerja F_m in na izhodu vrne kot odmika. Slika 2 prikazuje odziv sistema na stopnico.



Slika 2: Odziv sistema na stopico

Iz slike 2 je razvidno, da imamo opravka s sistemom drugega reda (ali več). Statično

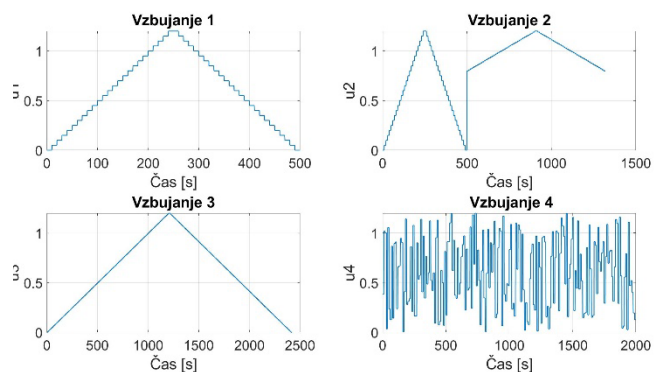
karakteristiko obravnavanega sistema prikazuje slika 3. Iz statične karakteristike na sliki 3 je razvidno, da je pri nižjem vzbujanju sistem dokaj linearen, nato pa z večanjem amplitude vzbujanja nelinearnost sistema narašča. Identifikacijo in ostale eksperimente smo izvajali z vhodnimi signali v območju od 0 do 1.2 Nm.



Slika 3: Statična karakteristika sistema

2.3 Izbira vzbujanja sistema

Zanimalo nas je, kako vzbujanje sistema vpliva na kvaliteto določenega modela. V ta namen definiramo štiri različne vzbujalne signale. Definirane signale prikazuje slika 4.



Slika 4: Vzbujalni signali za identifikacijo sistema

Prvi vzbujalni signal je definiran kot zaporedje stopnic velikosti 0.05 čez celotno delovno območje sistema. Drugi signal je podoben prvemu, le da mu dodamo stopnice velikosti 0.01 v območju, kjer je sistem najbolj nelinearen. Tretji signal je definiran kot stopnice

velikosti 0.01 čez celotno območje. Četrti signal sestavlja zaporedje stopnic konstante dolžine z naključnimi amplitudami.

2.4 Rezultati identifikacije mehkega modela

Algoritem mehke identifikacije smo poganjali pri naslednjih parametrih,

- $x_f(k) = [y(k-1), y(k-2), u(k-1)]$,
- maksimalno število rojev: $c_{max} = 1000$,
- red modela: $n = 2$,
- $\psi = [x_f, 1]$ in
- $\gamma_{max} = 0.1 - 0.9$.

Identifikacijo smo izvajali tako, da smo z vsemi štirimi signali vzbujali sistem, zajeli meritve in na njih izvajali algoritem uteženih najmanjših kvadratov, kjer za uteži uporabimo pripadnostne funkcije vhodnega regresorja k določenim rojem. Parameter γ_{max} smo spreminjali od 0.1 do 0.9 in pri vsaki vrednosti določili mehki model sistema. Iz med devetih določenih modelov izberemo najboljšega, tako da izračunamo koren srednje kvadratne napake (RMSE) med izhodom sistema in modela. RMSE računamo po enačbi (5)

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(y_m - y_p)^2}{n}} \quad (5)$$

Kjer so y_m izhod modela, y_p izhod sistema in n število vzorcev v meritvah.

Opisani eksperiment ponovimo za vsako vrsto vzbujanja (slika 4), tako na koncu dobimo 4 različne modele pri optimalnih vrednostih parametra γ_{max} . Rezultati eksperimenta so povzeti v tabeli 1.

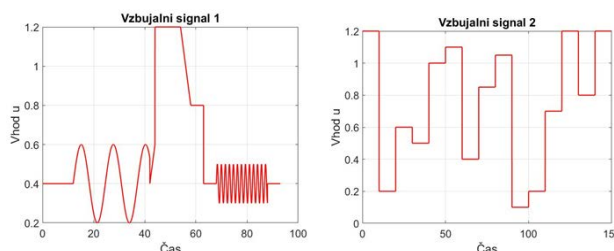
Tabela 1: Rezultati identifikacije za najboljši model pri posameznem vzbujalnem signalu

Tip signal	γ_{max}	Št. rojev	RMSE
Stopnice: 0.05	0.1	26	1.9446
Stopnice: 0.05 + 0.01	0.1	26	0.2623
Stopnice: 0.01	0.1	19	0.8340
Naključne stopnice	0.1	33	0.1112

Štiri določene modele iz tabele 1 uporabimo za validacijo in določimo najboljšega.

2.5 Validacija določenih modelov

Validacijo modelov izvedemo tako, da sistem in model vzbujamo s testnim signalom (različnim kot pri identifikaciji) in izračunamo koren srednje kvadratne napake (RMSE) med odzivom modela in odzivom sistema. Za validacijo uporabimo dva različna testna signala, ki sta prikazana na sliki 5.



Slika 5: Prvi testni signal za validacijo modelov

Rezultati validacije za vse 4 modele, so povzeti v tabeli 2.

Tabela 2: Rezultati validacije

Tip vzbujanja	Št. rojev	RMSE (signal 1)	RMSE (signal 2)
Stopnice: 0.05	26	1.3865	1.9446
Stopnice: 0.05 + 0.01	26	0.1467	0.2623
Stopnice: 0.01	19	0.5097	0.8340
Naključne stopnice	33	0.1000	0.1112

2.6 Komentar rezultatov identifikacije

Iz rezultatov v tabeli 2, lahko ugotovimo da izbira vzbujanja močno vpliva na rezultat identifikacije, saj je četrti model (pridobljen z naključnimi stopnicami) veliko boljši kot prvi model. Ugotovimo, da je najboljše vzbujanje številka štiri, naključne stopnice.

Če pogledamo rezultat validacije za najboljši model (zadnja vrstica v tabeli 2), lahko vidimo, da je algoritem našel 33 rojev, kar je veliko več, kot pa bi jih bilo dejansko potrebno. Manjše število rojev pomeni hitrejši čas računanja izhoda

modela, kar si definitivno pri samem vodenju želimo.

Problem prevelikega števila najdenih rojev rešimo tako, da vhodne meritve (regresor x_f) normaliziramo na interval od 0 do 1. Tako dosežemo, da imajo vse vhodne meritve enako težo in algoritem za iskanje rojev bolje deluje. Celoten postopek identifikacije in validacije iz poglavij 2.4 in 2.5 ponovimo na normaliziranih meritvah. Za vzbujanje uporabimo le naključne stopnice, za katere se izkaže, da so najbolj optimalno vzbujanje za tovrstni problem (tabela 2).

Primerjavo rezultatov validacije novega modela s »starim« modelom (najboljši model iz tabele 2) prikazuje tabela 3.

Tabela 3: Primerjava rezultatov validacije novega in starega modela

Model	Št. rojev	RMSE (signal 1)	RMSE (signal 2)
»Novi« model	3	0.9756	1.3222
»Stari« model	33	0.1000	0.1112

Iz rezultatov v tabeli 3 lahko vidimo, da je novi model slabši kot stari model, ampak imamo pri novem modelu le 3 roje, kar pomeni veliko manj računanja in hitrejše delovanje algoritma za prediktivno vodenje. Razporeditev centrov rojev novega modela na statični karakteristiki prikazuje slika 6.



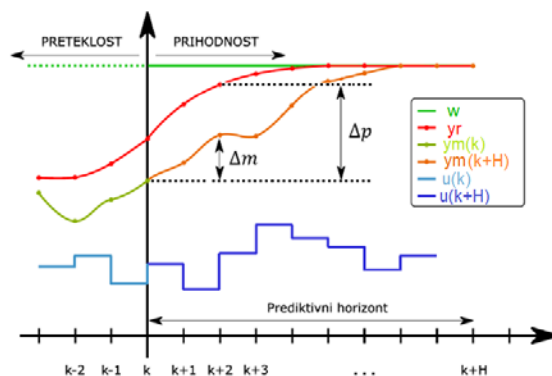
Slika 6: Razporeditev centrov rojev

Oba modela iz tabele 3 smo uporabili za izvedbo prediktivnega regulatorja, tako bomo lahko primerjali, kako kvaliteta modela vpliva na rezultat vodenja.

3 Mehko prediktivno vodenje sistema

3.1 Princip delovanja prediktivnega funkcijskega regulatorja

Osnovna ideja prediktivnega funkcijskega vodenja (PFC) je, da z uporabo modela napovemo, kako se bo sistem odzival v prihodnosti in na podlagi te napovedi izračunamo ustrezn regulirni signal [3]. Princip delovanja PFC prikazuje slika 7.



Slika 7: Princip delovanja PFC regulatorja

Horizont ujemanja H nam predstavlja točko v prihodnosti, kjer želimo doseči ujemanje odziva sistema in referenčnega signala, horizont predpišemo sami.

Preden začnemo izvajati algoritem vodenja, moramo definirati modelno referenčno trajektorijo y_r , s katero predpišemo, kako naj se odziv procesa obnaša od sedanjega trenutka do horizonta ujemanja. Referenčno trajektorijo lahko predpišemo z modelom prvega reda, ki ima enosmerno ojačenje 1, kot prikazuje enačba (6).

$$G_r(z) = \frac{Y_r(z)}{W(z)} = \frac{1-a_r}{z-a_r} \quad (6)$$

Ob vsakem diskretnem časovnem trenutku izračunamo napoved izhodnega signala $y_m(k+H)$ in napoved referenčne trajektorije $y_r(k+H)$ na horizontu ujemanja H . Cilj pristopa je določiti tak regulirni signal, da se ujematata razlika med napovedjo referenčne vrednosti in trenutno

vrednostjo izhoda (Δp) in razlika med napovedjo izhoda modela (Δm) [4],[5].

$$y_r(k+H) - y_p(k) = y_m(k+H) - y_m(k) \quad (7)$$

$$\Delta p = \Delta m \quad (8)$$

3.2 Regulirni zakon

V tem članku smo uporabili regulacijski zakon zapisan v prostoru stanj, ki ga izpeljemo analitično iz zveze (8). Regulirni zakon v prostoru stanj podaja enačba (9).

$$u(k) = \frac{y_r(k+H) - \tilde{c}_{m1} \tilde{A}_m^H x_{m1}(k) - y_{m2}(k+H) + e(k)}{\tilde{c}_{m1} (\tilde{A}_m^H - I) (\tilde{A}_m - I)^{-1} \tilde{B}_{m1}} \quad (9)$$

Kjer so $\tilde{A}_m(k)$, $\tilde{B}_{m1}(k)$, $\tilde{c}_{m1} \tilde{B}_{m2}(k)$, \tilde{c}_{m2} matrike prostora stanj mehkega modela, $y_r(k+H)$ predikcija referenčne trajektorije, $y_{m2}(k+H)$ predikcija izhoda modela zaradi prostega člena in $e(k)$ pogrešek med modelom in izhodom procesa.

Matrike prostora stanj mehkega modela določimo tako, da parametre lokalnih linearnih modelov pomnožimo s pripadnostnimi funkcijami β (za trenutni vhodni regresor), nato jih zapišemo v matrični obliki. Pretvorba mehkega modela v prostor stanj in izpeljava regulirnega zakona sta podrobneje opisani v članku [6].

3.3 Implementacija PFC vodenja

Vodenje smo izvajali s časom vzorčenja $T_s = 0.01$ s. Referenčni model smo izbrali kot sistem prvega reda z enosmernim ojačanjem 1 in časovno konstanto $T_r = 0.1$ s, kar je 1/5 časovne konstante sistema. Prenosno funkcijo referenčnega modela prikazuje enačba (10).

$$a_r = e^{-\frac{T_s}{T_r}} = 0.9512$$

$$G_r(z) = \frac{Y_r(z)}{W(z)} = \frac{1-a_r}{z-a_r} = \frac{0.04877}{z-0.9512} \quad (10)$$

Horizont ujemanja določimo po pravilu (11) [3].

$$n < H < \frac{T_r}{2 * T_s} \quad (11)$$

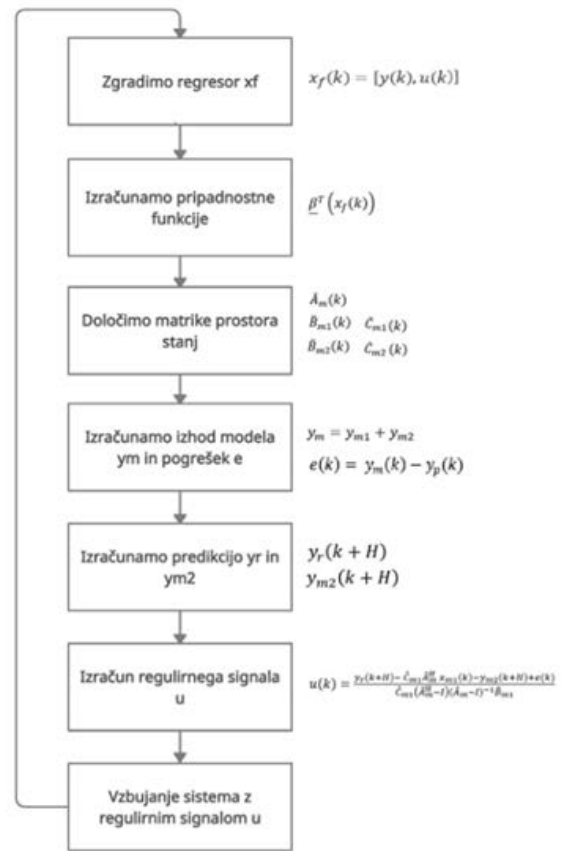
n ...red sistema

T_r ...časovna konstanta referenčnega modela

T_s ...čas vzorčenja

Iz enačbe (11) je razvidno, da mora biti vrednost H večja od reda sistema ($n = 2$) in manjša od 10, pri začetnih eksperimentih smo H nastavili na 5.

PFC algoritem izvajamo iterativno, eno iteracijo algoritma prikazuje slika 8.



Slika 8: Iteracija vodenja sistema s PFC

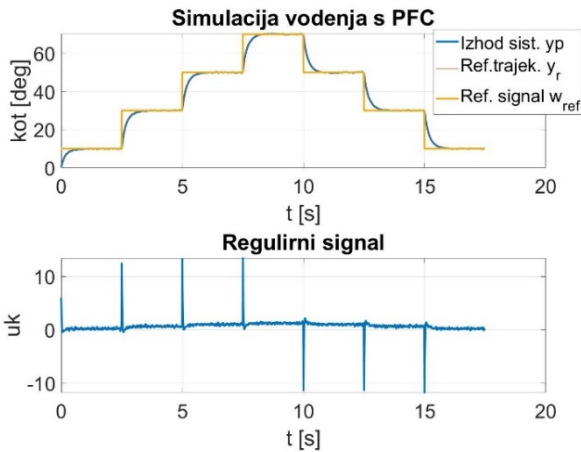
3.4 Rezultati vodenja s PFC regulatorjem

Za vodenje smo uporabili »novi« model iz tabele 3 (poglavje 2.6), to je model s 3. roji oz. lokalnimi modeli. Prvi poskus vodenja smo izvajali pri naslednjih parametrih:

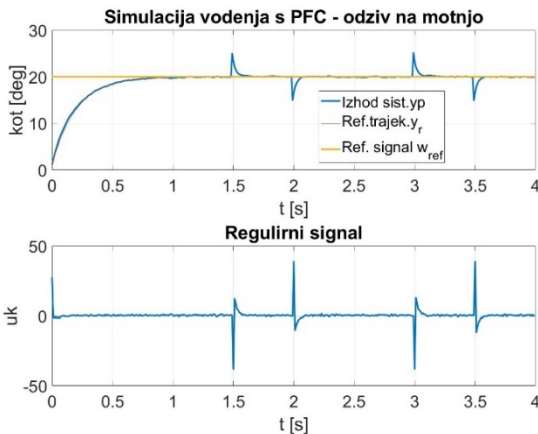
- Horizont ujemanja $H = 5$
- Časovna konstanta referenčnega modela $T_r = 0.1$ s,
- Čas vzorčenja $T_s = 0.01$ s

Za referenčni signal smo si izbrali stopničasti signal, rezultati vodenja so prikazani na sliki 9.

Preskusili smo tudi, kako dobro regulator odpravlja motnje. Eksperiment smo izvedli tako, da izberemo konstanten referenčni signal, nato ob nekem trenutku na izhod sistema prištejemo in odštejemo motnjo velikosti 5° . Obnašanje sistema ob nastopu motnje prikazuje slika 10.



Slika 9: Rezultati vodenja s PFC regulatorjem



Slika 10: Vpliv motnje na vodenje s PFC regulatorjem

Iz slike 10 lahko vidimo, da regulator PFC dobro izniči vpliv motnje, brez kakšnih dodatnih oscilacij izhoda sistema.

3.5 Vpliv izbire horizonta ujemanja na kvaliteto vodenja

Zanimalo nas je, kako izbira horizonta vpliva na kvaliteto vodenja. Zato smo izvedli

eksperiment, kjer smo horizont H spreminjali od 2 do 10 in opazovali, kako se kvaliteta vodenja spreminja. Kvaliteto vodenja smo določili tako, da smo računali napako RMSE med referenčno trajektorijo y_r in odzivom sistema y_p . RMSE računamo po naslednji enačbi:

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(y_r - y_p)^2}{n}} \quad (12)$$

Rezultati eksperimenta so prikazani v tabeli 4.

Tabela 4: Ocene RMSE pri različnih vrednostih horizonta ujemanja H

Vrednost H	Ocena RMSE
2	55.496
3	4.5784
4	0.077268
5	0.086989
6	0.10438
7	0.12757
8	0.1547
9	0.18448
10	0.21604

Kot lahko vidimo iz tabele 4, najboljše rezultate dobimo pri horizontu 4 in 5, potem pa z višanjem vrednosti horizonta kvaliteta vodenja pada. Prav tako lahko opazimo, da imamo pri $H = 2$ zelo slabo delovanje, kar je skladno s pravilom (11).

3.6 Vpliv kvalitete modela na kvaliteto vodenja

Preveriti smo želeli, kako sama kvaliteta modela vpliva na rezultat vodenja. Zato smo poleg modela s tremi roji («novi» model iz tabele 3) izvedli še vodenje z modelom s 33 roji («stari» model iz tabele 3).

Kvaliteto vodenja za posamezni model smo ocenili z napako RMSE. Vodenje smo izvajali pri izbranih parametrih:

- Horizont ujemanja $H = 4$
- Časovna konstanta referenčnega modela $T_r = 0.1$ s,
- Čas vzorčenja $T_s = 0.01$ s

Rezultati eksperimenta so prikazani v tabeli 5.

Tabela 5: Primerjava rezultatov vodenja za oba modela

Model	Št. rojev	H	RMSE (validacija)	RMSE (vodenje)
»Stari model«	33	4	0.1112	0.0511
»Novi model«	3	4	1.3222	0.077268

Kot lahko vidimo iz tabele 5, pri vodenju oba modela delujeta enako dobro, stari model je le malenkost boljši od novega, čeprav ima veliko boljšo oceno pri validaciji. Zaključimo lahko, da kvaliteta modela ne vpliva veliko na sam rezultat vodenja.

4 Zaključek

Rezultati eksperimenta iz poglavja 3.6 so pokazali, da sama kvaliteta modela ne vpliva na rezultat vodenja, saj se RMSE ocena za vodenje med novim in starim modelom ne razlikuje veliko. Kar posledično pomeni, da je novi model (s 3 roji) boljši kot stari model (s 33 roji), saj je manjše število rojev lažje interpretirati (razumeti njihov položaj in vpliv na vodenje) in izvajanje samega algoritma vodenja je veliko učinkovitejše, saj je zaradi manjšega števila potrebno veliko manj računanja.

Poleg dobrih lastnosti mehkega prediktivnega funkcijskega regulatorja je tudi nekaj slabih. Prediktivno funkcijsko vodenje temelji na podlagi modela sistema, zaradi tega je načrtovanje vodenja zahtevno in zamudno. Mehki model pridobimo z identifikacijo na realnem sistemu, torej mora realni sistem že obstajati. Pri identifikaciji je tudi pomembna izbira ustreznega vzbujačnega signala, kar pa zahteva nekaj eksperimentiranja na samem sistemu, kar je zopet zamudno.

Po naših ugotovitvah bi se za mehko prediktivo vodenje odločili v primeru, ko imamo opravka z izrazito nelinearnim sistemom in

želimo izvesti vodenje čez celotno delovno območje sistema. Če pa imamo opravka z linearnim sistemom ali pa z nelinearnim sistemom, katerega želimo voditi okoli izbrane delovne točke, mehko prediktivno funkcijsko vodenje ni tako primerno. V tem primeru lahko dobre rezultate pridobimo tudi s preprostejšimi strukturami, kot je PID regulator.

5 Literatura

- [1] A. Zdešar, D. Dovžan, in I. Škrjanc, „Self-tuning of 2 DOF control based on evolving fuzzy model“, *Appl. Soft Comput.*, let. 2014, št. 19, str. 403–418.
- [2] G. Andonovski, G. Mušič, S. Blažič, in I. Škrjanc, „On-line Evolving Cloud-based Model Identification for Production Control“, *IFAC-Pap.*, let. 2016, št. 49(5), str. 79–84.
- [3] D. Dovžan in I. Škrjanc, „Predictive functional control based on an adaptive fuzzy model of a hybrid semi-batch reactor“, *Control Eng. Pract.*, let. 2010, št. 18(8), str. 979–989.
- [4] I. Škrjanc in D. Matko, „Predictive functional control based on fuzzy model for heat-exchanger pilot plant“, *IEEE Trans. Fuzzy Syst.*, let. 2000, št. 8(6), str. 705–712.
- [5] I. Škrjanc, *Inteligentni sistemi za podporo odločanju*, let. 2016. Ljubljana.
- [6] I. Škrjanc in D. Matko, „Fuzzy Predictive Functional Control in the State Space Domain“, *J. Intell. Robot. Syst.*, let. 2001, št. 31, str. 283–297.