



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Илија Каменко

**Аутоматско подешавање параметара
ПИД регулатора
засновано на расплинutoј логици
са применом у аутоматизацији
стамбено-пословних објеката**

ДОКТОРСКА ДИСЕРТАЦИЈА

Нови Сад, 2023. година



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:		
Идентификациони број, ИБР:		
Тип документације, ТД:	Монографска документација	
Тип записа, ТЗ:	Текст, штампани материјал	
Врста рада, ВР:	Докторска дисертација	
Аутор, АУ:	Илија Каменко	
Ментор, МН:	др Велимир Чонградац, редовни професор	
Наслов рада, НР:	Аутоматско подешавање параметара ПИД регулатора засновано на расплинutoј логици са применом у аутоматизацији стамбено-пословних објеката	
Језик публикације, ЈП:	Српски	
Језик извода, ЈИ:	Српски/Енглески	
Земља публикавања, ЗП:	Република Србија	
Уже географско подручје, УГП:	АП Војводина	
Година, ГО:	2023	
Издавач, ИЗ:	Ауторски репринт	
Место и адреса, МА:	Факултет техничких наука, 21000 Нови Сад, Трг Доситеја Обрадовића 6	
Физички опис рада, ФО: (поглавља/страна/ цитата/табела/слика/графика/прилога)	9/147/107/32/73/0/3	
Научна област, НО:	Електротехничко и рачунарско инжењерство	
Научна дисциплина, НД:	Аутоматика и управљање системима	
Предметна одредница/Кључне речи, ПО:	ПИД, расплинута (фази) логика, експертско знање, аутоматско подешавање параметара, стамбено-пословни објекти, модел у реалном времену	
УДК		
Чува се, ЧУ:	Библиотека Факултета техничких наука, Универзитет Нови Сад	
Важна напомена, ВН:		
Извод, ИЗ:	У дисертацији је представљено једно решење аутоматског подешавања ПИД (пропорционално-интегрално-диференцијалног) регулатора засновано на експертском знању имплементираном у расплинutom (фази) систему закључивања. Предложена метода је верификована рачунарским симулацијама коришћењем широког спектра различитих модела процеса. Након успешне верификације предложена метода је имплементирана на реалном контролеру који се обично користи у аутоматизацији стамбено-пословних објеката. Тестирање је извршено на експерименталној поставци у лабораторији.	
Датум прихватања теме, ДП:	22.09.2022	
Датум одбране, ДО:		
Чланови комисије, КО:	Председник:	Др Зоран Јеличић, редовни професор
	Члан:	Др Филип Кулић, редовни професор
	Члан:	Др Томислав Шекара, редовни професор
	Члан:	Др Владимир Бугарски, доцент
	Члан:	Др Срђан Вукмировић, редовни професор
	Члан, ментор:	др Велимир Чонградац, редовни професор
		Потпис ментора



KEY WORDS DOCUMENTATION

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	Monographic documentation
Type of record, TR :	Text, printed material
Contents code, CC :	PhD thesis
Author, AU :	Ilija Kamenko
Mentor, MN :	Velimir Čongradac, PhD, full professor
Title, TI :	Auto-tuning of PID controller based on fuzzy logic with application in building automation
Language of text, LT :	Serbian
Language of abstract, LA :	Serbian/English
Country of publication, CP :	Republic of Serbia
Locality of publication, LP :	AP of Vojvodina
Publication year, PY :	2023
Publisher, PB :	Autor reprint
Publication place, PP :	Faculty of Technical Sciences, 21000 Novi Sad, Trg Dositeja Obradovića 6
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	9/147/107/32/73/0/3
Scientific field, SF :	Electrical and Computer Engineering
Scientific discipline, SD :	Automation and Control Systems
Subject/Key words, S/KW :	PID, fuzzy logic, expert knowledge, auto-tuning, building automation, real-time model, Hardware-in-loop
UC	
Holding data, HD :	Library of the Faculty of Technical Sciences, University of Novi Sad
Note, N :	
Abstract, AB :	The dissertation presents a solution for automatic tuning of the PID (proportional-integral-differential) controller based on expert knowledge implemented in a fuzzy inference system. The proposed method was verified by computer simulations using a wide range of different process models. After successful verification, the proposed method was implemented on a real controller commonly used in the automation of residential and commercial buildings. The testing was done on an experimental setup in the laboratory.
Accepted by the Scientific Board on, ASB :	2022/09/22
Defended on, DE :	
Defended Board, DB :	
President:	Zoran Jeličić, PhD, full professor
Member:	Filip Kulić, PhD, full professor
Member:	Tomislav Šekara, PhD, full professor
Member:	Vladimir Bugarski, PhD, assistant professor
Member:	Srđan Vukmirović, PhD, full professor
Member, Mentor:	Velimir Čongradac, PhD, full professor
	Mentor's sign

Посвећено успомени на драгог оца...

РЕЗИМЕ

У дисертацији је представљена нова метода за аутоматско подешавање ПИД (пропорционално-интегрално-диференцијалног) регулатора заснована на експертском знању имплементираном у расплинута (фази) систем закључивања.

Предложена метода итеративно покушава да побољша перформансе система у затвореној спрези. Као мере перформанси предложена метода користи карактеристике одскочног одзива (време успона, прескок и време смирења). Параметри ПИД регулатора у првој итерацији могу се израчунати на основу једноставног експеримента у отвореној спрези или је могуће користити постојеће параметре. У свакој узастопној итерацији се врши прорачун карактеристика одскочног одзива. Релативне промене изражене у процентима вредности у првој итерацији се затим израчунавају и претварају у лингвистичке вредности. Користећи базу од 29 правила, фази експертски систем израчунава фази вредности које се користе након дефазификације као фактори множења за тренутне ПИД параметре. Да би се постигао баланс између агресивног и робусног одзива у затвореној спрези, као и између споријег и бржег, фази експертски систем може да ради у три различита режима рада: за убрзавање система, за смањење прескока и за уравнотежено смањење времена успона и прескока.

Верификација рачунарским симулацијама је извршена коришћењем широког спектра различитих модела процеса који се најчешће налазе у проблемима аутоматизације стамбено-пословних објеката.

Како би се извршила верификација у реалним експерименталним условима предложена метода је имплементирана на реалном контролеру који се обично користи у аутоматизацији стамбено-пословних објеката. Верификација је извршена на експерименталној НПЛ (Hardware-in-loop) поставци у оквиру које је на посебном контролеру реализован модел топлотне подстанице система даљинског грејања који се извршава у реалном времену.

Кључне речи: ПИД, расплинута (фази) логика, експертско знање, аутоматско подешавање параметара, стамбено-пословни објекти, модел у реалном времену

ABSTRACT

The dissertation presents a new method for automatic tuning of the PID (proportional-integral-differential) controller based on expert knowledge implemented in the fuzzy inference system.

The proposed method iteratively tries to improve the performance of the closed-loop system. As performance measures, the proposed scheme uses the characteristics of the step response (rise time, overshoot, and settling time). PID parameters in the first iteration can be calculated based on the basic open-loop step response experiment or it is possible to use current parameters. In each successive iteration, step response characteristics are measured. Relative changes expressed in the percentage of value in the first iteration are then calculated and converted into linguistic values. Using a rule base of 29 fuzzy rules, the fuzzy expert system computes fuzzy values that are used after defuzzification as multiplying factors for current PID parameters. To achieve a balance between the aggressive and robust closed-loop response, as well as between the slower and the faster one, the fuzzy expert system works in three operating modes: the one for speeding up the system, the one for reducing the overshoot, and the one for a balanced reduction of rise time and overshoot.

Verification by computer simulations was performed using a wide range of different process models that are most often found in problems of automation of residential and commercial buildings.

To perform the verification in real experimental conditions, the proposed method was implemented on a real controller commonly used in the automation of residential and commercial buildings. The verification was performed on an experimental HIL (Hardware-in-loop) setup, in which a real time model of heating substation of district heating system was implemented on a separate controller.

Keywords: PID, fuzzy logic, expert knowledge, auto-tuning, building automation, real-time model, Hardware-in-loop

ЗАХВАЛНИЦА

Захваљујем се свом ментору, проф. др Велимиру Чонградцу, на знању и саветима које ми је пренео током докторских студија и израде докторске дисертације.

Исто тако се захваљујем и свом саветнику, проф. др Филипу Кулићу, на чију иницијативу сам и остао на Катедри као сарадник и започео докторске судије.

Велику захвалност дугујем проф. др Зорану Јеличићу на подршци и разумевању као и помоћи током трајања докторских студија.

Хвала колегама др Драгану Матићу, доц. др Владимиру Бугарском и мр Перици Николићу, са којим сам делио канцеларију дуги низ година, на сарадњи и веома инспиративним разговорима из којих су се родиле многе идеје.

Хвала свим колегама са Катедре.

Захваљујем се мами и сестри које су ме свесрдно подржавале и биле уз мене у току мог школовања.

Ипак, највећу захвалност дугујем мојој породици, супрузи Валентини и сину Николи који су били уз мене и подржавали ме у сваком погледу.

САДРЖАЈ

1 Увод	1
2 Полазне основе истраживања	3
2.1 Предмет и циљ истраживања	3
2.2 Стање у научној области	3
3 Основе ПИД регулатора	6
3.1 Повратна спрега	6
3.2 ПИД закон управљања.....	7
3.3 Модификовани ПИД.....	9
3.3.1 Филтрирање диференцијалног дејства.....	9
3.3.2 ПИД са два степена слободе	9
3.3.3 ПИД са <i>anti-windup</i> дејством	10
3.4 Дискретни ПИД.....	11
3.5 Историјски преглед развоја.....	12
4 Основе фази система	15
4.1 Основе фази логике.....	15
4.1.1 Фази скупови	15
4.1.2 Функција припадности	17
4.1.3 Фази операције	20
4.1.4 Лингвистичке променљиве	20
4.2 Структура фази система	21
4.2.1 Фазификација	21
4.2.2 База правила.....	22
4.2.3 Апроксимативно закључивање	22
4.2.4 Дефазификација	24
4.3 Сугено фази систем закључивања	25
5 Аутоматско подешавање параметара.....	26
5.1 Перформансе и робусност	27
5.1.1 Мере перформанси.....	27
5.1.2 Мере робусности	28
5.2 Методе базиране на моделу	29
5.2.1 Идентификација процеса.....	29
5.2.1.1 Апроксимација моделом	30
5.2.1.2 Карактеризација у фреквенцијском домену	32
5.2.2 Методе подешавања.....	33
5.2.2.1 Циглер-Николсова метода.....	33
5.2.2.2 Острем-Хеглундова метода	33
5.3 Методе базиране на правилима	34
5.3.1 Методе засноване на фази логици	35

6 Предложена метода за аутоматско подешавање параметара	37
6.1 Структура методе	37
6.2 Скуп процеса задатих функцијама преноса.....	38
6.3 Синтеза експертског знања	39
6.4 Фази систем закључивања (Мамдани)	40
6.4.1 Фазификација	40
6.4.2 Дефазификација	41
6.4.3 База правила.....	41
6.5 Фази систем закључивања (Сугено).....	42
6.6 Режији рада.....	43
6.7 Опис поступка	45
7 Симулациони резултати.....	46
7.1 Имплементација предложене методе у симулационом софтверу	46
7.1.1 Имплементација почетног подешавања параметара	46
7.1.2 Имплементација аутоматског подешавања параметара.....	47
7.2 Симулациони модели.....	48
7.2.1 Симулациони модел почетног подешавања параметара.....	49
7.2.2 Симулациони модел аутоматског подешавања параметара	50
7.3 Резултати и анализа	51
7.3.1 Предложена метода у режиму рада Убрзање	51
7.3.1.1 Пример: Процес са доминантном вредношћу временске константе	51
7.3.1.2 Пример: Процес са доминантном вредношћу временског кашњења ...	52
7.3.1.3 Пример: Процес са равномерним вредностима временске константе и временског кашњења.....	53
7.3.1.4 Пример: Процес са интегратором	54
7.3.1.5 Анализа над целим скупом процеса.....	56
7.3.2 Предложена метода у режиму рада Смањење прескока	57
7.3.2.1 Пример: Процес са доминантном вредношћу временске константе	57
7.3.2.2 Пример: Процес са доминантном вредношћу временског кашњења ...	58
7.3.2.3 Пример: Процес са равномерним вредностима временске константе и временског кашњења.....	59
7.3.2.4 Пример: Процес са интегратором	60
7.3.2.5 Анализа над целим скупом процеса.....	61
7.3.3 Предложена метода у режиму рада Балансирање.....	62
7.3.3.1 Пример: Процес са доминантном вредношћу временске константе	62
7.3.3.2 Пример: Процес са доминантном вредношћу временског кашњења ...	63
7.3.3.3 Пример: Процес са равномерним вредностима временске константе и временског кашњења.....	64
7.3.3.4 Пример: Процес са интегратором	66
7.3.3.5 Анализа над целим скупом процеса за овај режим рада.....	67
8 Експериментални резултати	69
8.1 Имплементација предложене методе на реалном контролеру	69

8.1.1 ПИД регулатор у FBD програмском језику.....	69
8.1.1.1 PID1 и PIDA блокови	70
8.1.1.2 PIDP блок	70
8.1.2 Опис имплементације	71
8.1.3 Надзорно-управљачки систем.....	74
8.2 Имплементација модела топлотне подстанице на реалном контролеру	74
8.2.1 Модел измењивача топлоте	75
8.2.2 Модел трокраког вентила.....	77
8.2.2.1 Модел вентила.....	77
8.2.2.2 Модел мешања	78
8.3 Експериментална поставка.....	78
8.4 Резултати и анализа	80
8.4.1 Аутоматско подешавање параметара са почетним подешавањем	80
8.4.1.1 Пример: Почетни параметри добијени применом О-Х методе.....	80
8.4.2 Аутоматско подешавање параметара без почетног подешавања.....	82
8.4.2.1 Пример: Почетни параметри са изразито апериодичним одзивом.....	83
8.4.2.2 Пример: Почетни параметри са изразито осцилаторним одзивом	84
9 Закључак	87
Литература.....	89
Прилози	95
I Прилог - Изворни код предложене методе у Simulink развојном окружењу	95
I.1 Функцијски блок <i>initTuningPID</i>	95
I.2 Функцијски блок <i>autoTuningPID</i>	98
I.3 Скрипте за промену режима рада.....	102
II Прилог - Изворни код предложене методе у MATLAB развојном окружењу	104
II.1 Функција <i>initTuningPID</i>	104
II.2 Функција <i>autoTuningPID</i>	104
II.3 Скрипта <i>tableResults</i>	107
III Прилог - Изворни код модела топлотне подстанице у LabView развојном окружењу	109
III.1 Модел измењивача топлоте	109
III.2 Модел вентила.....	110
III.3 Модел мешања	110
IV Прилог - Изворни код предложене методе у FBD програмском језику	111
IV.1 Основни функцијски блок <i>FuzzyAutoTuning</i>	111
План третмана података	126

СПИСАК СКРАЋЕНИЦА

ПИД	Пропорционално-интегрално-диференцијални закон управљања
ФЕС	Фази (расплинути) експертски систем
Ц-Н	Циглер-Николсова (Ziegler-Nichols) метода за подешавање параметара ПИД регулатора на основу параметара добијених идентификацијом процеса
О-Х	Острем-Хеглундова (Åström-Hägglund) метода за подешавање параметара ПИД регулатора на основу параметара добијених идентификацијом процеса
FOIPTD	First Order Integral Process With a Time Delay (модел заснован на процесу првог реда са временским кашњењем)
FOPD	First Order Process With Time Delay (модел заснован на интегралном процесу првог реда са временским кашњењем)
IAE	Integral of absolute error (интеграл апсолутне грешке)
FBD	Function block diagram (програмски језик Функцијски блок дијаграм)
НIL	Hardware-in-loop (експериментална поставка у оквиру које је процес којим се управља реализован као модел на посебном контролеру који омогућује извршавање истог у реалном времену)
TCP/IP	Transmission Control Protocol/Internet protocol (скуп комуникационих протокола који се користе у рачунарским мрежама и интернету)
FPGA	Field Programmable Gate Arrays (представља интегрисано коло пројектовано тако да је његова унутрашња структура прилагодљива)

ПРЕГЛЕД СЛИКА

Слика 3.1 Систем са повратном спрегом	6
Слика 3.2 Типови on-off регулатора	7
Слика 3.3 Илустрација рада ПИД закона управљања.....	8
Слика 3.4 Систем са два степена слободе	10
Слика 3.5 ПИД са anti-windup дејством	11
Слика 3.6 Први регулатор са сва три дејства (преузето из [65]).....	14
Слика 4.1 Графичка репрезентација скупова.....	16
Слика 4.2 Функције припадности класичног скупа (лево) и фази скупа (десно).....	16
Слика 4.3 Пример троугаоне функције припадности	17
Слика 4.4 Пример трапезоидне функције припадности	18
Слика 4.5 Пример Гаусове функције припадности	18
Слика 4.6 Пример звонасте функције припадности.....	19
Слика 4.7 Пример сигмоидалне функције припадности	19
Слика 4.8 Пример затворене сигмоидалне функције припадности.....	20
Слика 4.9 Графички приказ операција над фази скуповима	20
Слика 4.10 Структура фази система	21
Слика 4.11 Графички приказ Мамдани апроксимативног закључивања.....	23
Слика 4.12 Графички приказ метода дефазификације.....	24
Слика 4.13 Графички приказ Сугено апроксимативног закључивања.....	25
Слика 5.1 Перформансе система на одскочном одзиву на побуду.....	27
Слика 5.2 Перформансе система на одскочном одзиву на поремећај.....	28
Слика 5.3 Графичка репрезентација мера робусности	29
Слика 5.4 Метода тангенте	30
Слика 5.5 Метода тангенте за процесе са интегратором.....	31
Слика 5.6 Метода површина.....	31
Слика 5.7 Пример мапе подешавања	35
Слика 5.8 Принципијелна шема аутоматског подешавања параметара ПИД регулатора	36
Слика 6.1 Блок дијаграм предложене методе	37
Слика 6.2 Мапа подешавања за процес G_4 (6.10) са параметром $n=4$ (плава пуна $T_d=0,2$; црвена тачкаста $T_d=0,7$; црна испрекидана $T_d=1,1$).....	39
Слика 6.3 Блок дијаграм Мамдани фази система закључивања	40
Слика 6.4 Функције припадности за три улаза: количник времена успона, прескока и времена смирења.....	41

Слика 6.5 Функције припадности за три излаза: K_p , T_i и T_d множитељ.....	41
Слика 6.6 Блок дијаграм Сугено фази система закључивања.....	43
Слика 6.7 Функције припадности за три излаза: K_p , T_i и T_d множитељ.....	43
Слика 6.8 Позиције функција припадности за две различите вредности параметра P – пуне $P=50$, испрекидане $P=75$	44
Слика 6.9 Дијаграм тока предложене методе	45
Слика 7.1 Реализовани Simulink блок initTuningPID	46
Слика 7.2 Реализовани Simulink блок autoTuningPID.....	47
Слика 7.3 Алат за пројектовање фази система у оквиру симулационог софтвера	48
Слика 7.4 Симулациони модел за почетно подешавање. Пример за G_5 (6.11) са параметром $T=0,1$	49
Слика 7.5 Пример идентификације система у отвореној спреси за G_5 (6.11) са параметром $T=0,1$	49
Слика 7.6 Симулациони модел за аутоматско подешавање за процес G_5 (6.11) са параметром $T=0,1$	50
Слика 7.7 Simulink блок са реализацијом ПИД регулатора	50
Слика 7.8 Пример тока аутоматског подешавања параметара ПИД регулатора за процес G_5 (6.11) са параметром $T=0,1$ коришћењем предложене методе у режиму рада Смањење прескока.....	51
Слика 7.9 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_5 (6.11) са параметром $T=0,1$ применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно).....	52
Слика 7.10 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_2 (6.8) са параметром $T=0,02$ применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно).....	53
Слика 7.11 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_4 (6.10) са параметром $n=3$ применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно).....	54
Слика 7.12 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_6 (6.12) са параметром $T=0,9$ применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно).....	55
Слика 7.13 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_5 (6.11) са параметром $T=0,1$ постигнуте применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)	57
Слика 7.14 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_2 (6.8) са параметром $T=0,02$ применом	

предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно).....	58
Слика 7.15 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_4 (6.10) са параметром $n=3$ применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно).....	59
Слика 7.16 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_6 (6.12) са параметром $T=0,9$ применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно).....	60
Слика 7.17 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_5 (6.11) са параметром $T=0,1$ применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно).....	63
Слика 7.18 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_2 (6.8) са параметром $T=0,02$ применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно).....	64
Слика 7.19 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_4 (6.10) са параметром $n=3$ применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно).....	65
Слика 7.20 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_6 (6.12) са параметром $T=0,9$ применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно).....	66
Слика 8.1 ПИД блокови у FBD програмском језику	69
Слика 8.2 Блок дијаграм закона управљања PIDA блока	70
Слика 8.3 Блок дијаграм закона управљања PIDP блока.....	71
Слика 8.4 Хијерархијски блокови PIDAFuzzy и PIDPFuzzy	71
Слика 8.5 Реализација блока PIDPFuzzy	73
Слика 8.6 Реализовани надзорно-управљачки систем.....	74
Слика 8.7 Шематски приказ топлотне подстанице	75
Слика 8.8 Експериментална НПЛ поставка.....	78
Слика 8.9 SE SmartX AS-B-24 контролер	79
Слика 8.10 NI sbRIO-9636 контролер	79
Слика 8.11 Шематски приказ експерименталне НПЛ поставке	80
Слика 8.12 Поступак аутоматског подешавања применом предложене методе у режиму рада Балансирање након почетног подешавања О-Х методом за PIDAFuzzy блок	81

Слика 8.13 Поступак аутоматског подешавања применом предложене методе у режиму рада Балансирање након почетног подешавања О-Х методом за PIDPFuzzy блок.....	81
Слика 8.14 Одзив на скоковиту промену задате температуре мешања (горе) и на поремећај у виду скоковите промене температуре повратне воде (доле) добијен променом PIDAFuzzy блока (лево) и PIDPFuzzy блока (десно)	82
Слика 8.15 Поступак аутоматског подешавања кроз итерације применом предложене методе у режиму рада Убрзање за PIDAFuzzy блок.....	83
Слика 8.16 Поступак аутоматског подешавања кроз итерације применом предложене методе у режиму рада Убрзање за PIDPFuzzy блок	83
Слика 8.17 Одзив на скоковиту промену задате температуре мешања (горе) и на скоковиту промену температуре повратне воде у виду поремећаја (доле) добијен променом PIDAFuzzy блока (лево) и PIDPFuzzy блока (десно).....	84
Слика 8.18 Поступак аутоматског подешавања параметара кроз итерације применом предложене методе у режиму рада Смањење прескока за PIDAFuzzy блок	85
Слика 8.19 Поступак аутоматског подешавања параметара кроз итерације применом предложене методе у режиму рада Смањење прескока за PIDPFuzzy блок	85
Слика 8.20 Одзив на скоковиту промену задате температуре мешања (горе) и на скоковиту промену температуре повратне воде у виду поремећаја (доле) добијен променом PIDAFuzzy блока (лево) и PIDPFuzzy блока (десно).....	86

ПРЕГЛЕД ТАБЕЛА

Табела 6.1 Основни односи између повећања параметара ПИД регулатора и карактеристика одскочног одзива.....	40
Табела 6.2 Спецификација Мамдани фази система закључивања.....	40
Табела 6.3 База правила фази система закључивања	42
Табела 6.4 Спецификација Сугено фази система закључивања	43
Табела 7.1 Спецификација улаза и излаза Simulink блока initTuningPID	47
Табела 7.2 Спецификација улаза и излаза MATLAB функције initTuningPID.....	47
Табела 7.3 Преглед улаза и излаза Simulink блока autoTuningPID	48
Табела 7.4 Спецификација улаза и излаза MATLAB функције autoTuningPID	48
Табела 7.5 Перформансе и робусност ПИД регулатора за процес G_5 (6.11) са параметром $T=0,1$ постигнуте применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом и О-Х методом	52
Табела 7.6 Перформансе и робусност ПИД регулатора за процес G_2 (6.8) са параметром $T=0,02$ постигнуте применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом и О-Х методом	53
Табела 7.7 Перформансе и робусност ПИД регулатора за процес G_4 (6.10) са параметром $n=3$ постигнуте применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом и О-Х методом	54
Табела 7.8 Перформансе и робусност ПИД регулатора за процес G_6 (6.12) са параметром $T=0,9$ постигнуте применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом и О-Х методом	55
Табела 7.9 Просечно побољшање вредности перформанси и робусности за све процесе постигнуто предложеном методом у режиму рада Убрзање након почетног подешавања Ц-Н методом.....	56
Табела 7.10 Просечно побољшање вредности перформанси и робусности за све процесе постигнуто предложеном методом у режиму рада Убрзање након почетног подешавања О-Х методом.....	56
Табела 7.11 Перформансе и робусност ПИД регулатора за процес G_5 (6.11) са параметром $T=0,1$ предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом и О-Х методом.....	58
Табела 7.12 Перформансе и робусност ПИД регулатора за процес G_2 (6.8) са параметром $T=0,02$ постигнуте применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом и О-Х методом	59
Табела 7.13 Перформансе и робусност ПИД регулатора за процес G_4 (6.10) са параметром $n=3$ постигнуте применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом и О-Х методом	60
Табела 7.14 Перформансе и робусност ПИД регулатора за процес G_6 (6.12) са параметром $T=0,9$ постигнуте применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом и О-Х методом	61

Табела 7.15 Просечно побољшање вредности перформанси и робусности за све процесе постигнуто предложеном методом у режиму рада Смањење прескока након почетног подешавања Ц-Н методом.....	61
Табела 7.16 Просечно побољшање вредности перформанси и робусности за све процесе постигнуто предложеном методом у режиму рада Смањење прескока након почетног подешавања О-Х методом.....	62
Табела 7.17 Перформансе и робусност ПИД регулатора за процес G_5 (6.11) са параметром $T=0,1$ постигнуте применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом и О-Х методом.....	63
Табела 7.18 Перформансе и робусност ПИД регулатора за процес G_2 (6.8) са параметром $T=0,02$ постигнуте применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом и О-Х методом.....	64
Табела 7.19 Перформансе и робусност ПИД регулатора за процес G_4 (6.10) са параметром $n=3$ постигнуте применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом и О-Х методом.....	65
Табела 7.20 Перформансе и робусност ПИД регулатора за процес G_6 (6.12) са параметром $T=0,9$ постигнуте применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом и О-Х методом.....	66
Табела 7.21 Просечно побољшање вредности перформанси и робусности за све процесе постигнуто предложеном методом у режиму рада Балансирање након почетног подешавања Ц-Н методом.....	67
Табела 7.22 Просечно побољшање вредности перформанси и робусности за све процесе постигнуто предложеном методом у режиму рада Балансирање након почетног подешавања О-Х методом.....	68
Табела 8.1 Преглед улаза и излаза ПИД блокова.....	69
Табела 8.2 Преглед улаза и излаза блокова PIDAFuzzy и PIDPFuzzy.....	72
Табела 8.3 Перформансе ПИД регулатора постигнуте применом предложене методе у режиму рада Балансирање након почетног подешавања О-Х методом.....	82
Табела 8.4 Перформансе ПИД регулатора постигнуте применом предложене методе у режиму рада Убрзање користећи постојеће параметре као почетне.....	84
Табела 8.5 Перформансе ПИД регулатора постигнуте применом предложене методе у режиму рада Смањење прескока користећи постојеће параметре као почетне.....	86

1 Увод

Све аутоматске системе управљања карактерише специфичан закон управљања или закони управљања. За основне законе управљања ефекат може линеарно зависити од грешке (П закон), интеграла грешке (И закон) или првог извода грешке у времену (Д закон). Комбиновањем основних закона управљања добија се ПИД (пропорционално-интегрално-диференцијални) закон који је данас најчешће коришћен закон управљања. ПИД регулатор има три подесива параметра, по један за сваки закон управљања. Перформансе и робусност система у затвореној спреси у уској су вези са вредношћу параметара регулатора. Међутим, поред једноставне имплементације, постављање три подесива ПИД параметра није лак задатак и годинама представља велики изазов за многе научнике и инжењере.

Упркос многим софистицираним теоријама и техникама управљања које су осмишљене последњих деценија, ПИД регулатори су и даље најприхваћенији у практичним случајевима. Томе доприноси њихова једноставност са становишта теоријских основа као и са становишта имплементације на реалним контролерима. Штавише, често су доступни пошто су уграђени у програмабилне логичке контролере који се користе у управљању многим индустријским процесима као и у аутоматизацији кућа, стамбених и пословних зграда. Нажалост, у многим управљачким круговима ПИД регулатори често нису правилно подешени због варијација параметара процеса, промена радних услова итд.

У инжењерској пракси, након што се неком од познатих метода одреде параметри ПИД регулатора, често се врши њихова корекција у складу са специфичним захтевима који нису експлицитно математички дати у критеријуму за подешавања и/или у ограничењима. Ово прецизно подешавање се постиже применом малих корекција параметара на основу практичног искуства. Да би се смањило време и знање неопходно за подешавање параметара ПИД регулатора постоји значајна потреба за аутоматизацијом овог поступка. Поступак аутоматског подешавања у овом случају се обично састоји од три корака: генерисања поремећаја на улазу система, евалуације перформанси одзива и прорачуна параметара регулатора.

За имплементацију оваквих решења која у раду треба да опонашају искусног инжењера приликом ручног подешавања параметара потребно је имплементирати експертско знање у виду система базираног на рачунарској интелигенцији. Расплинута (фази) логика је препозната као одговарајући метод за имплементацију искуства оператера у пројектовању ПИД регулатора. Расплинута (фази) логика је уопштење класичне логике у оквиру које се могу дефинисати искази чија истинитосна вредност може припадати континуалном прелазу од тачног ка нетачном, за разлику од класичне логике где искази могу имати вредности тачно и нетачно. У литератури на српском језику се често употребљава термин расплинута логика. Међутим, без утицаја на разумевање а због једноставности код креирања сложеница, у наставку ће се користити термин фази логика.

Иако постоји много конвенционалних метода за аутоматско подешавање параметара ПИД регулатора, у оквиру дисертације акценат ће бити на пружању основних

идеја о томе како пројектовати експертски систем за аутоматско подешавање параметара ПИД регулатора применом фази логике.

Дисертацију чине девет поглавља, преглед литературе и прилози.

У првом поглављу је дат увод у дисертацију.

У другом поглављу су описане полазне основе истраживања као што су предмет и циљ истраживања и преглед тренутног стања у научној области од интереса.

У трећем поглављу су описани основни принципи регулационих кругова са ПИД регулаторима као и историјски осврт на развој истих.

У четвртом поглављу су представљене основе фази логике као и опис структуре најчешће коришћених фази система закључивања.

Пето поглавље је посвећено методама за аутоматско подешавања параметара ПИД регулатора.

У шестом поглављу је дат опис предложене методе за аутоматско подешавање параметара ПИД регулатора засноване на фази експертском систему.

Седмо поглавље је посвећено верификацији предложене методе применом рачунарских симулација. Описана је имплементација у симулационом софтверу и дат је преглед и анализа резултата добијених симулацијом над широким скупом различитих модела процеса.

Осмо поглавље је посвећено верификацији предложене методе у експерименталним условима. Приказан је опис имплементације предложене методе на реалном контролеру као и опис експерименталне поставке. Потом су приказани и анализирани добијени резултати експерименталним тестирањем.

У деветом поглављу су изложени основни закључци, завршне напомене и правци даљих истраживања.

У посебном поглављу је приказан списак консултоване литературе у току истраживања и писања дисертације.

У оквиру прилога приказан је изворни код предложене методе у симулационом софтверу и изворни код имплементације на реалном контролеру. Поред тога је дат и изворни код модела процеса који се извршава у реалном времену на контролеру у оквиру експерименталне поставке.

2 Полазне основе истраживања

У многим управљачким круговима ПИД регулатори често нису правилно подешени због варијација параметара процеса, промена радних услова или уопште због погрешног почетног подешавања приликом стављања система у рад. Подешавање параметара упркос многобројним доступним методама је често веома захтевно са становишта потребног предзнања, искуства и утрошеног времена. Због тога постоји значајна потреба за методама за аутоматско подешавање параметара које би се покретале на захтев корисника и применом одређеног алгорита извршиле подешавање параметара.

2.1 Предмет и циљ истраживања

Основна тема истраживања је имплементирање искуства оператера у процесу пројектовању система управљања заснованих на ПИД закону управљања. Фази логика као један од видова рачунарске интелигенције је препозната као веома погодна у примени искуства оператера у пројектовању таквих система.

Предмет истраживања је примена фази логике у поступку аутоматског подешавања параметара ПИД регулатора. Пројектовање, реализација и имплементација ових самоподешавајућих регулатора имају два доминантна изазова: синтезу експертског знања и развој наменског система закључивања базираног на фази логици. Такође, предмет истраживања је и анализа могућности примене предложене методе у реалним експерименталним условима у проблемима аутоматизације стамбено-пословних објеката.

Циљ истраживања је развијање нове методе за аутоматско подешавање параметара ПИД регулатора засноване на експертском знању интегрисаном у фази систем закључивања. Примарни циљ методе је побољшавање перформанси произвољног система у затвореној спреси уз одржавање робусности у прихватљивим границама минимизујући при томе улагање времена, труда и потребног знања. Предложена метода ће бити верификована коришћењем рачунарских симулација на широком скупу процеса како би могла да се потврди способност генерализације предложеног решења. Након верификације, предложена метода ће бити имплементирана на реалном контролеру који се користи у аутоматизацији стамбено-пословних објеката. Као крајњи резултат се очекује верификација предложене методе и у реалним експерименталним условима.

Мотивација за ово истраживање произлази из стварних проблема са којима су се кандидат и ментор сусрели током година у системима аутоматског управљања у стамбено-пословним објектима.

2.2 Стање у научној области

Први систематски начин подешавања параметара ПИД регулатора предложили су Циглер (Ziegler) и Николс (Nichols) кроз два рада [1], [2] почетком 40-тих година прошлог века. Инспирисани овим радом, многи аутори су предлагали побољшања

како би испунили различите захтеве перформанси или робусности [3], [4]. Ове методе су дате у виду једноставних израза који формализују претходно експериментално одређене карактеристике динамике процеса. Упркос својој несавршености, оригинални Циглеров и Николсов метод се и данас широко користи због своје једноставности и лакоће употребе. Почетком овога века, више од 60 година касније од оригиналног рада, Острем (Åström) и Хеглунд (Hägglund) су предложили ревитализацију оригиналне методе у [5], [6] уз задржавање једноставности, али уз побољшање перформанси за шири скуп различитих процеса. Недуго потом, Шекара и Матаушек су у [7] такође предложили ревитализацију оригиналне методе Циглера и Николса. Свеобухватан преглед развоја теорије и праксе ПИД регулатора је представљен у [8], док је резиме метода подешавања на систематски начин дат у [9].

Да би смањили време и знање неопходно за подешавање ПИД параметара, Острем и Хеглунд су предложили први метод аутоматског подешавања параметара [10]. Аутоматско подешавање параметара је остварено комбиновањем релејног експеримента као методе за одређивање динамике процеса са методама за прорачун параметара ПИД регулатора. Последњих година, аутори су предлагали различите приступе аутоматском подешавању параметара ПИД регулатора [11]–[16]. Најновије поређење метода аутоматског подешавања параметара ПИД регулатора је представљено у [17], [18].

Након што су параметри ПИД регулатора подешени, често је потребно ручно прецизније подесити параметре. Подешавање се постиже применом малих корекција параметара на основу практичног искуства. Фази логика је препозната као одговарајући метод за имплементацију искуства оператера у пројектовању ПИД регулатора. Заде (Zadeh) је 1965. године предложио теорију фази скупова као проширење класичног појма скупа [19]. Инспириран овим радом, 1974. године Мамдани (Mamdani) је предложио прву примену теорије фази скупова у управљачком алгоритму да би синтетизовао лингвистички управљачки механизам искусног оператера [20]. У наредним годинама, фази логика је успешно коришћена са веома запаженим резултатима у различитим областима попут аутоматизације у стамбено-пословним објектима [21]–[23], саобраћаја [24]–[26], индустрије [27], [28] као и осталим инжењерским применама [29]–[34]. У [35] дат је сажет преглед истакнутих резултата истраживања управљачких алгоритама заснованих на концепту фази логике који показује еволуцију ове области.

Цафестас (Tzafestas) и Папаникопоулос (Papnikoroulos) су били међу првима који су предложили приступ у коме се перформансе система у затвореној спреси побољшавају благом променом вредности ПИД параметара на основу фази матрице која садржи искуство оператера у облику кондензованих правила [36]. Аутори у [37]–[39] су предложили алгоритам за аутоматско подешавање заснован на механизму фази закључивања за континуирано подешавање ПИД параметара користећи сигнал грешке и извод сигнала грешке система у затвореној спреси као улазе. У [40] аутори су користили време смирења у одзиву на одскачну побуду као улаз у фази систем закључивања који на излазу генерише корективне факторе за сваки од параметара ПИД регулатора. Многи аутори су примењивали сличне принципе и увели мало другачије начине за имплементацију експертског знања у процесу подешавања параметара ПИД регулатора коришћењем концепта фази логике [41]–[49]. Поређење метода аутоматског подешавања параметара ПИД регулатора заснованих на фази логици је представљено у [50].

Потврду исправности развијених управљачких алгоритама је у потпуности могуће добити једино кроз њихову примену на реалном систему. Са обзиром на смањену

расположивост реалног система као и могућност тестирања нових алгоритама на њему, алтернативно решење је реализација математичког модела неког подсистема у реалном времену. У [51], [52] је дат детаљан преглед метода за моделовање подсистема који се најчешће могу наћи у стамбено-пословним објектима. Како би се реализован математички модел извршавао у реалном времену потребно га је имплементирати на посебном контролеру базираном на FPGA (Field Programmable Gate Arrays) технологији који омогућавају перформансе у реалном времену [53], [54].

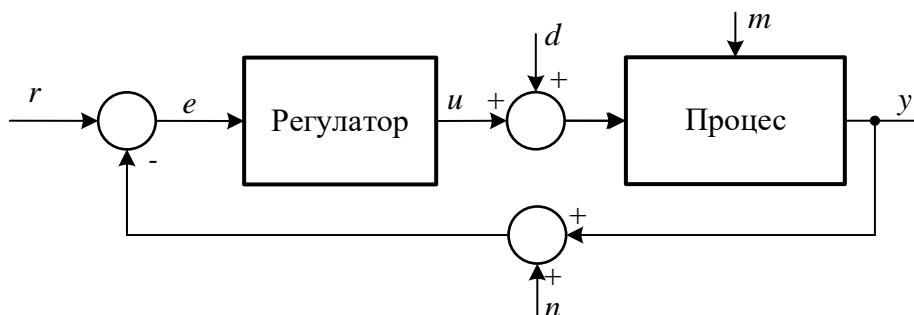
Приступ тестирању реализованог управљачког алгорита који садржи физички контролер повезан у затвореној спреси са виртуелним подсистемом, пружајући верне физичке реплике реалног система, у литератури се налази под називом Hardware-in-loop (HIL) [55], [56]. Предности овог приступа се огледају у спречавању оштећења на стварном систему, смањењу трошкова отклањања грешака и смањењу свеобухватног напора приликом тестирања [54].

3 Основе ПИД регулатора

У овом поглављу ће бити дате основе повратне спреге и најчешће коришћених закона управљања са акцентом на ПИД закон управљања. ПИД закон управљања ће бити разматран у континуалном и дискретном домену. Биће размотрене и различите модификације које побољшавају основну верзију. Такође, биће дат и историјски преглед развоја ПИД регулатора.

3.1 Повратна спрега

На слици 3.1 приказан је блок дијаграм из кога се може видети принцип рада система са негативном повратном спрегом. Упоредивањем тренутне вредности u и задате вредности r добија се одступање e , односно сигнал грешке. Регулатор на основу овог одступања генерише управљачки сигнал u који делује на процес у смеру смањења овог одступања. Основна предност повратне спреге је одржавање тренутне вредности излаза што приближније задатој вредности без обзира на спољне поремећаје d , n и m као и промену параметара процеса. Затварање повратне спреге поред свих добрих особина има и један недостатак. Мерењем излазног сигнала који је најчешће зашумљен мерним шумом може доћи до продора тог шума у систем што даље утиче на квалитет управљања.



Слика 3.1 Систем са повратном спрегом

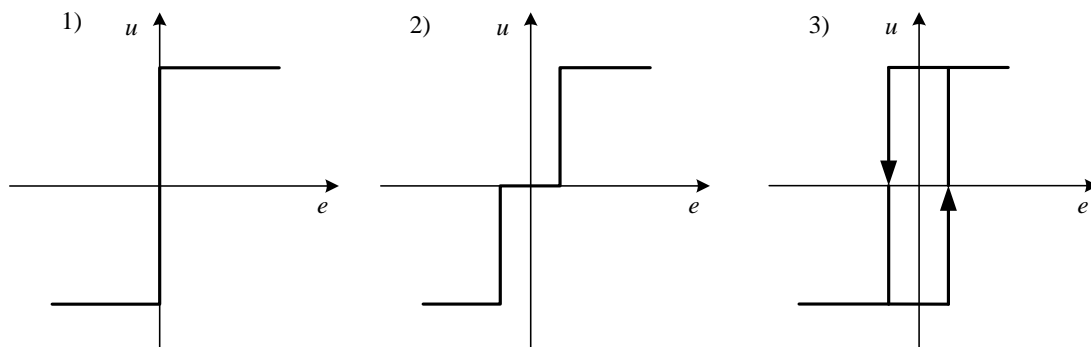
Идеја повратне спреге о генерисању корективних акција на основу разлике између задате и тренутне вредности излазне величине може се применити на много различитих начина. Најједноставнији је *on-off* закон управљања који се математички може приказати следећим изразом

$$u(t) = \begin{cases} u_{max} & e(t) > 0 \\ u_{min} & e(t) < 0 \end{cases} \quad (3.1)$$

где је $e(t)$ сигнал грешке који се добија као разлика задате $r(t)$ и тренутне вредности $u(t)$ излаза система. Овај закон управљања имплицира да се увек користи максимална корективна акција (Слика 3.2 под 1). Овај закон управљања често успева у одржавању излаза процеса близу задате вредности али ће обично резултирати одзивом који ће осцилирати око те вредности.

Модификацијом основног закона добијају се верзије које успевају да делом минимизују проблем али га не успавају елиминисати. Модификована верзија која уводи

Још један ниво управљачке променљиве може се видети на слици 3.2 под 2. Модификована верзија са хистерезисом код које управљачка променљива зависи од претходних вредности грешке приказана је на слици 3.2 под 3.



Слика 3.2 Типови on-off регулатора

3.2 ПИД закон управљања

Разлог зашто *on-off* регулатор често изазива осцилације је тај што систем претерано реагује јер ће мала промена у грешци учинити да се управљачка променљива промени у целом опсегу.

Овај ефекат се избегава код пропорционалног управљања где је карактеристика регулатора пропорционална сигналу грешке. Пропорционални регулатор уводи побољшање које се огледа у пропорционалном преношењу сигнала грешке $e(t)$ на излаз $u(t)$, према изразу

$$u(t) = K_p e(t) + u_b \quad (3.2)$$

где је K_p појачање а u_b је вредност управљања кад је сигнал грешке једнак нули. Вредност u_b се често поставља на унапред задату али се може и подесити тако да вредност грешке у устаљеном стању буде једнака нули за дату референцу. За дату вредност сигнала грешке велика вредност појачања даје велику и брзу промену у одзиву система али ако је појачање превелико, систем може постати нестабилан. Пропорционални закон управљања има недостатак да се грешка у устаљеном стању не може потпуно елиминисати.

Интегрални закон управљања даје сигнал управљања сразмеран акумулираним вредностима сигнала грешке и дат је следећим изразом

$$u(t) = \frac{K_p}{T_i} \int_0^t e(\tau) d\tau \quad (3.3)$$

где је T_i временска константа интеграљења. Јасно је да ће се излаз $u(t)$ мењати у смеру корекције одступања све док то одступања не ишчезне јер промена излаза престаје само када улаз односно одступање постане нула. Промена излаза система није више тренутна, као код пропорционалног закон управљања, већ поступна. Она је утолико бржа, уколико је временска константа T_i мања.

Због динамике процеса потребно је неко време пре него што промена сигнала управљања буде приметна на излазу система односно управљачки систем ће каснити у исправљању грешке. Акција регулатора са пропорционалним и диференцијалним дејством може се тумачити као да је управљање пропорционално предвиђеном

сигналу грешке. Предвиђање се врши екстраполацијом грешке тангентом на криву сигнала грешке. Диференцијални закон управљања је дат изразом

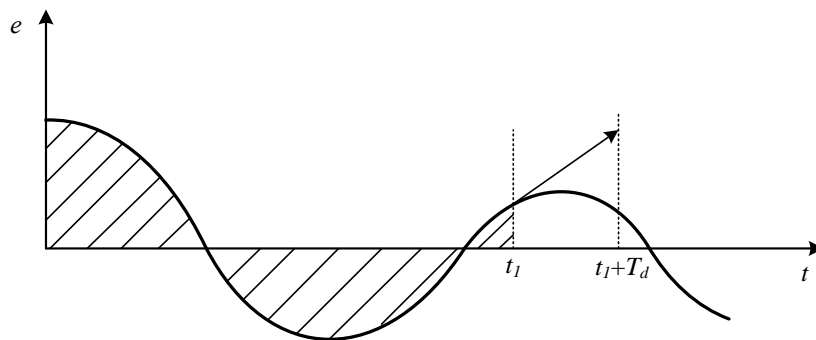
$$u(t) = K_p T_d \frac{de(t)}{dt} \quad (3.4)$$

где је T_d временска константа диференцирања и представља приближно време у којем се вредност грешке предвиђа унапред.

Комбинујући пропорционални, интегрални и диференцијални закон управљања, добија се закон управљања који гласи

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) dt + T_d \frac{de(t)}{dt} \right) \quad (3.5)$$

Сигнал управљања је стога збир три члана: П-члана (који је пропорционалан сигналу грешке), И-члана (који је пропорционалан интегралу грешке) и Д-члана (који је пропорционалан изводу сигнала грешке). Овај облик повратне спреге назива се пропорционално-интегрално-диференцијални (ПИД) закон управљања и његово деловање је илустровано на слици 3.3. У тренутку t_1 , пропорционални члан зависи од тренутне вредности грешке. Интегрални члан је заснован на интегралу грешке до времена t_1 (осенчени део). Диференцијални члан пружа процену вредности грешке у тренутку $t_1 + T_d$ [8].



Слика 3.3 Илустрација рада ПИД закона управљања

ПИД закон управљања дат изразом 3.5 у комплексном домену има облик

$$U(s) = K_p \left(1 + \frac{1}{sT_i} + sT_d \right) E(s) \quad (3.6)$$

који се назива стандардна форма.

Паралелна форма је дата изразом

$$U(s) = \left(k_p + \frac{k_i}{s} + s k_d \right) E(s) \quad (3.7)$$

где је $k_p=K_p$, $k_i=K_p/T_i$, а $k_d=K_p T_d$. Оваква репрезентација има предност јер је могуће добити чисто пропорционално, интегрално или диференцијално дејство коначним вредностима параметара. Међутим, параметри немају физичку репрезентацију као код стандардне форме.

3.3 Модификовани ПИД

ПИД регулатор који се заправо користи у пракси садржи неколико модификација које су условљене практичним проблемима. У наставку су дати најчешћи проблеми и модификације које их успешно решавају.

3.3.1 Филтрирање диференцијалног дејства

Недостатак диференцијалног дејства је то што идеалан извод има веома високо појачање за високофреквентне сигнале. То значи да ће шум мерења високе фреквенције генерисати велике варијације сигнала управљања. Ефекат шума мерења може се у извесној мери смањити применом нископропусног филтера у оквиру диференцијалног дејства датог изразом

$$U(s) = K_p \frac{sT_d}{1 + s\frac{T_d}{N}} E(s) \quad (3.8)$$

Идеални извод се филтрира системом првог реда са временском константом T_d/N . Ова апроксимација делује као извод за нискофреквентне компоненте сигнала, док се шум мерења високих фреквенција појачава највише за фактор KN . Типичне вредности N су од 2 до 20 [8].

Закон управљања у комплексном домену приказан у изразу 3.6 када се примени филтрирање постаје

$$U(s) = K_p \left(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + s\frac{T_d}{N}} \right) E(s) \quad (3.9)$$

3.3.2 ПИД са два степена слободе

Управљачки системи треба да задовоље различите захтеве односно да имају одговарајући одзив на промену задате вредности као и да имају елиминацију поремећаја и потискивање шума мерења. Регулатори који управљачки сигнал добијају само на основу сигнала грешке, покушавајући да на тај начин задовоље све поменуте захтеве, називају се регулаторима са једним степеном слободе (Слика 3.1). Како би се могло утицати посебно на одзив система на промену задате вредности а посебно на одзив на поремећај потребно је модификовати стандардну управљачку структуру.

Модификована верзија ПИД регулатора са два степена слободе је први пут уведена у [57]. За систем се каже да има два степена слободе јер је путања сигнала од задате вредности до управљачког сигнала различита од путање сигнала од мерене вредности до управљачког сигнала.

Постоји много могућих конфигурација система са два степена слободе а један од њих је дат изразом

$$u(t) = K_p \left(e_p(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de_d(t)}{dt} \right) \quad (3.10)$$

где је грешка у пропорционалном члану

$$e_p(t) = br(t) - y(t) \quad (3.11)$$

а грешка у диференцијалном члану

$$e_d(t) = cr(t) - y(t) \quad (3.12)$$

док је грешка у интегралном члану прави сигнал грешке

$$e(t) = r(t) - y(t) \quad (3.13)$$

како би се избегла грешка у устаљеном стању.

Параметри b и c се називају тежински фактори задате вредности. Они немају утицаја на одзив система на поремећај, али имају значајан утицај на одзив система на промене задате вредности. За вредност тежинског фактора c се обично усваја нула осим кад је промена задате вредности глатка односно постепена. Тежински фактор b се бира на основу процењених временских константи система којим се управља [6].

ПИД закона управљања са два степена слободе у комплексном домену ако се усвоји за c вредност нула је дат изразом

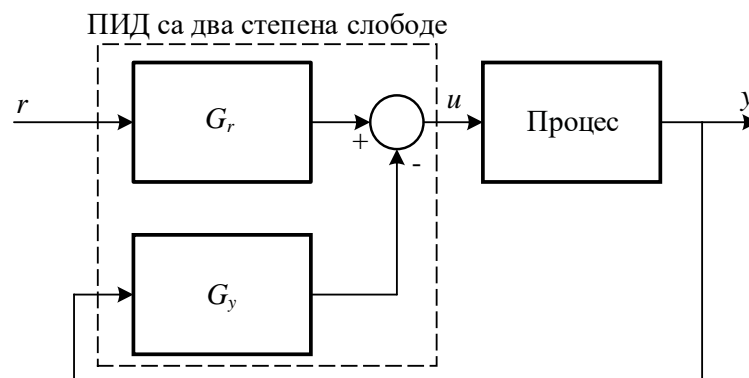
$$\begin{aligned} U(s) &= K_p \left(bR(s) - Y(s) + \frac{1}{sT_i} (R(s) - Y(s)) - sT_d Y(s) \right) = \\ &= K_p \left(b + \frac{1}{sT_i} \right) R(s) - K_p \left(1 + \frac{1}{sT_i} + sT_d \right) Y(s) \end{aligned} \quad (3.14)$$

Претходни израз се може представити структурним блок дијаграмом представљеним на слици 3.4 где важи

$$G_r(s) = K_p \left(b + \frac{1}{sT_i} \right) \quad (3.15)$$

односно

$$G_y = K_p \left(1 + \frac{1}{sT_i} + sT_d \right) \quad (3.16)$$



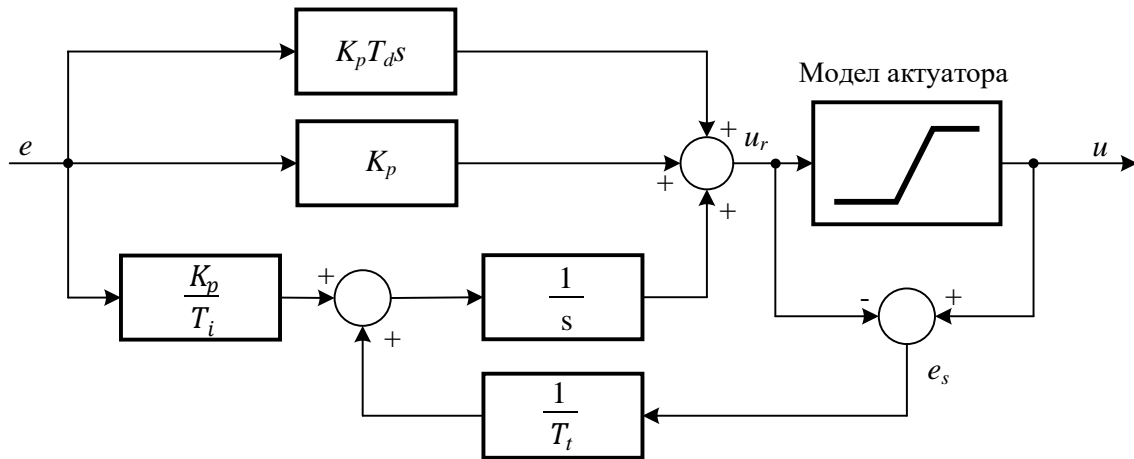
Слика 3.4 Систем са два степена слободе

3.3.3 ПИД са *anti-windup* дејством

Сви актуатори имају ограничења попут вентила који не може да буде више од потпуно затвореног или потпуно отвореног или мотора који не може да има већу брзину од номиналне. Код процеса којим се управља помоћу регулатора који има интегрално дејство може се десити да сигнал управљања стигне до границе актуатора. Сигнал грешке ће наставити да се интегрирали и са даљим порастом доћи ће до ефекта

integrator windup. Даља промена грешке на неко време неће правилно утицати на сигнал управљања све док грешка не промени знак и док се не поништи пребачај сигнал управљања преко опсега актуатора.

На слици 3.5 приказан је блок дијаграм ПИД регулатора са *anti-windup* дејством. Систем има додатну повратну спрегу у виду мерења стварног излаза актуатора или излаза математичког модела актуатора са ограничењима. На овај начин се формира сигнал грешке e_s као разлика између сигнала управљања u_r и излаза из модела актуатора u . Сигнал e_s се доводи на улаз интегратора преко појачања $1/T_i$. Сигнал e_s је нула када је сигнал управљања у границама модела актуатора и неће имати никаквог утицаја на нормалан рад регулатора. Када сигнал управљања прекорачи границе модела актуатора, сигнал e_s ће узети вредност тог прекорачења. Добијена вредност ће бити искоришћена како би се кориговао интегрални члан и спречио ефекат *integrator windup*. Односно, сигнал управљања ће бити ограничен на вредности ограничења модела актуатора. Временска константа T_i треба да буде већа од T_d а мања од T_i односно предлаже се вредност $\sqrt{T_i T_d}$ [8].



Слика 3.5 ПИД са *anti-windup* дејством

3.4 Дискретни ПИД

За имплементацију на дигиталном рачунару закона управљања дефинисаног у континуалном домену, као што је ПИД регулатор, потребно је апроксимирати извод и интеграл који се појављују у закону управљања.

Полазећи од тога да је T време одабирања постоји неколико могућих метода апроксимације [58]. Једна од метода је диференцирање уназад (*backward difference*) код које се извод апроксимира разликом између одбирка у тренутку kT и одбирка у претходном тренутку $(k-1)T$

$$\frac{dy(t)}{dt} \approx \frac{y(kT) - y(kT - T)}{T} \quad (3.17)$$

Са друге стране код методе диференцирања унапред (*forward difference*) извод се апроксимира разликом између одбирка у тренуцима $(k+1)T$ и kT

$$\frac{dy(t)}{dt} \approx \frac{y(kT + T) - y(kT)}{T} \quad (3.18)$$

Уколико је систем задат у комплексном домену, тада се може директно увести смена комплексне променљиве s [59] на неки од следећих начина

- Диференцирање уназад (*backward difference*)

$$s = \frac{z - 1}{zT} \quad (3.19)$$

- Диференцирање унапред (*forward difference*)

$$s = \frac{z - 1}{T} \quad (3.20)$$

- Централна разлика или Туситинова метода

$$s = \frac{2z - 1}{Tz + 1} \quad (3.21)$$

Генерално постоје мале разлике између апроксимација за мале вредности периоде одабирања T све док је добијени дискретни систем стабилан. Због практичних разлога како би се могло искључити диференцијално дејство код ПИД регулатора најбољи избор за апроксимацију је метода диференцирања уназад која даје стабилан систем за $T_d=0$ за разлику од преостале две методе [8].

Коришћењем методе апроксимације уназад, дискретизована верзија ПИД закона управљања у изразу 3.9, постаје

$$U(z) = K_p \left(1 + \frac{T}{(1 - z^{-1})T_i} + \frac{T_d(1 - z^{-1})}{T} \right) E(z) \quad (3.22)$$

односно

$$U(z)(1 - z^{-1}) = K_p(1 - z^{-1})E(z) + \frac{K_p T}{T_i} E(z) + \frac{K_p T_d}{T} (1 - z^{-1})^2 E(z) \quad (3.23)$$

Последњи израз се може записати у облику диференцне једначине која представља инкрементални ПИД закон управљања

$$u(k) = u(k - 1) + K_p(e(k) - e(k - 1)) + \frac{K_p T}{T_i} e(k) + \frac{K_p T_d}{T} (e(k) - 2e(k - 1) + e(k - 2)) \quad (3.24)$$

Дискретизована верзија закона управљања са два степена слободе приказаног у изразу 3.14 се добија на исти начин и гласи

$$u(k) = u(k - 1) + K_p(br(k) - y(k) - br(k - 1) + y(k - 1)) + \frac{K_p T}{T_i} (r(k) - y(k)) - \frac{K_p T_d}{T} (y(k) - 2y(k - 1) + y(k - 2)) \quad (3.25)$$

3.5 Историјски преглед развоја

Технологија која се користи за имплементацију регулатора се природно значајно променила током година. Први регулатори су били механички уређаји попут центрифугалних регулатора који су се користили првенствено за управљање млиновима погоњених ветром и парним машинама. Мерење угаоне брзине је комбиновано са

актуатором попут вентила. У годинама које су следиле начињена су многа унапређења овог система попут интегралне акције која је омогућавала нулту грешку у устаљеном стању [60].

Развој регулатора је био уско повезан са развојем сензора и актуатора све до почетка прошлог века када је дошло до значајних промена у вези са развојем управљања индустријским процесима. Тадашње велике компаније за производњу мерно-регулационе опреме су независно радиле на унапређењу својих регулатора на основу реалних проблема који су се јављали приликом имплементацији истих. Затим су развојене функције сензора, регулатора и актуатора. Интегрална акција названа је аутоматским ресетовањем јер је заменила ручно ресетовање које је коришћено у пропорционалним регулаторима да би се добила исправна вредност у устаљеном стању. Разматрано је увођење дејства које би било пропорционално изводу сигнала грешке које би омогућило агресивније управљање. Пренос мерних сигнала као и пренос и прорачун управљачких сигнала вршен је пнеуматски. Велики напредак се десио када је притисак у цевима за пренос стандардизован на 3-15 PSI (*pound per square inch*). Ово је омогућило комбиновање сензора, регулатора и актуатора различитих произвођача. Такође је омогућило да се контролери концентришу у одвојене контролне собе које су биле смештене далеко од сензора и актуатора [8].

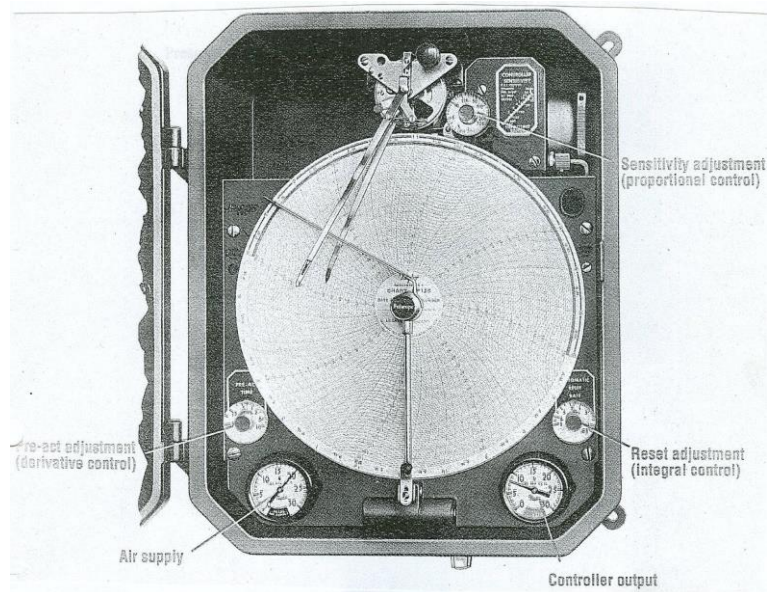
Многа практично корисна унапређења регулатора прво су се појавила као посебне хардверске функције. Оне нису биле изражене у математичком облику. Математичка анализа парне машине са регулатором је први пут објављена у [61] и поред осталог је показала разлику између пропорционалног и интегралног управљања. У [62] је представљена веома јасна аналитичка анализа управљачког закона за аутоматско управљање бродом састављеног од три дејстава које је касније постало познато као ПИД закон управљања.

Компанија Taylor Instrument Company је 1939. године представила потпуно редицајнирану верзију свог пнеуматског контролера који је поред до тада стандардног пропорционалног и интегралног дејства, обезбедио и диференцијално дејство. Ово је био први комерцијални ПИД регулатор са сва три дејства која су се могла независно подешавати у пуном опсегу вредности [63]. Пример једног таквог уређаја приказан је на слици 3.6. Могућност подешавања сва три параметра за разлику од до тада већ преподешених убрзо је постао проблем. Није постојао метод избора одговарајућих вредности за сваки од три дејства регулатора. У оквиру поменуте компаније спроведена су опсежна истраживања у покушају да се осмисли начин за избор оптималних параметара ПИД регулатора. Исход овог истраживања су два рада [1], [2] у којима су Циглер и Николс показали како се оптимални параметри регулатора могу подесити на основу експерименталних тестова у отвореној спрези а затим и у затвореној спрези.

Почевши од 50-их година прошлог века постале су доступне електронске верзије ПИД регулатора реализованих коришћењем једноставних аналогних рачунара заснованих на операционим појачавачима. Пренос сигнала је такође стандардизован и то у виду струјног сигнала у опсегу 4-20mA који погодује процесу накнадне дијагностике [8], [64].

У првим управљачким системима заснованим на рачунару, 60-их година прошлог века, дигитални рачунари су давали задате вредности за аналогне регулаторе. Појава микропроцесора средином 70-их година прошлог века омогућила је коришћење дигиталног управљања за контролере са једним управљачким кругом. То је довело до развоја дистрибуираних управљачких система за управљање процесима где је ПИД регулатор био кључни елемент. Како се рачунарска снага микропроцесора повећавала

било је могуће увести подешавање, прилагођавање и дијагностику у контролере са једним управљачким кругом [63], [64].



Слика 3.6 Први регулатор са сва три дејства (преузето из [65])

У последње време ПИД регулатори се могу наћи у многим облицима. Постоје наменски контролери који могу да регулишу један или неколико кругова док се ПИД регулатор као посебан блок може наћи у програмабилним логичким контролерима.

4 Основе фази система

Фази систем је било који статички или динамички систем који користи фази логику и одговарајућу математичку подршку (теорија фази скупова, фази правила и фази закључивање). Овако широка дефиниција обухвата фази системе који се користе у различитим подручјима и за које се користе називи: фази експертски систем (*fuzzy expert system*), фази регулатор (*fuzzy logic controller, FLC*), фази модел (*fuzzy model*) и систем фази правила (*fuzzy rule based system*). Фази систем прати човеков начин размишљања, тако да је стално присутна физичка интерпретација редоследа и начина доношења одлука. Сваки корак у алгоритму и сваки параметар фази система има јасну физичку интерпретацију. Важну предност фази система представља и чињеница да се без обзира на сложеност система може развити алгоритам без постојања било каквог модела физичког система [66]–[73].

У овом поглављу биће дат преглед основних појмова из теорије фази скупова. Структура основног фази система ће бити разложена на делове почевши од фазификације преко базе правила и механизма закључивања па све до дефазификације. Сваки део ће бити посебно разматран као увод у боље разумевање пројектовања фази експертског система за подешавање параметара ПИД регулатора.

4.1 Основе фази логике

Фази (расплинута) логика је проширење класичне логике у оквиру које се могу дефинисати искази чија истинитосна вредност може припадати континуалном прелазу од тачног ка нетачном. Концепт фази логике увео је Заде 1965. године [19].

Конкретније речено, оно што је централно за фази логику је да за разлику од класичних логичких система има за циљ моделирање непрецизних начина расуђивања који играју суштинску улогу у људској способности да доноси рационалне одлуке у окружењу неизвесности и непрецизности [74].

4.1.1 Фази скупови

У класичној теорији скупова коју је утемељио немачки математичар Кантор (Georg Cantor) скуп представља колекцију објеката који деле неку заједничку особину [71]. Припадност објекта класичном скупу егзактно је одређена, односно или је објекат унутар скупа или је изван њега. Ако се припадност објекта скупу жели изразити функцијски, тада функција припадности $\mu_A(x)$ објекта x скупу A има само две вредности и дефинисана је

$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \quad (4.1)$$

Подела објеката на скупове строгом дефиницијом припадности није увек прикладна јер понекад није могуће одредити задовољавајућу границу између објеката.

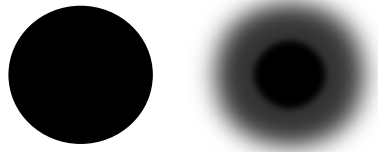
За разлику од класичног приступа, теорија фази скупова допушта делимичну или процентуалну припадност елемента скупу, а вредности функције припадности се налазе у опсегу од 0 до 1. Где 0 означава да је вредност потпуно изван скупа, а 1 да је

потпуно унутар скупа док било која вредност између означава делимичну припадност скупу.

Теорија фази скупова уводи континуалну функцију припадности $\mu_A(x)$. Ова функција показује колико $x \in X$ испуњава услов припадности скупу A у зависности од вредности x . Формално, фази скуп A се дефинише као скуп уређених парова

$$A = \{(x, \mu_A(x)) | x \in X\} \quad (4.2)$$

где је x члан скупа A , $\mu_A(x)$ представља функцију припадности елемента x фази скупу A . На овај начин се уводи неодређеност која потиче од особине људи да ствари не посматрају на дискретан начин. Графичка репрезентација примера класичног и фази скупа је приказан на слици 4.1 где је црном бојом означена припадност скупу.



Слика 4.1 Графичка репрезентација скупова

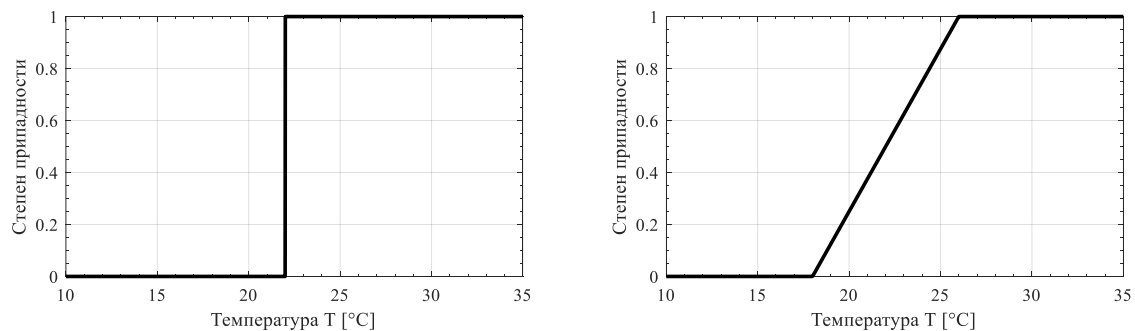
Као типичан пример често се наводи проблем математичког описа људске перцепције температуре околине где се по вредности врло блиске величине могу наћи у различитим скуповима. Тако на пример у скуп *toplo* у зависности од тога како ко перципира топлоту могу се сместити разне вредности али ако је 22°C доња граница овог скупа онда је вредност $21,9^\circ\text{C}$ већ у другом скупу нпр. *hladno*. Припадност скупу *toplo* се може записати као

$$\mu_{toplo}(x) = \begin{cases} 1, & \text{за } x \geq 22^\circ\text{C} \\ 0, & \text{за } x < 22^\circ\text{C} \end{cases} \quad (4.3)$$

Исти скуп би се могао дефинисати као фази скуп описан следећом функцијом припадности

$$\mu_{toplo}(x) = \begin{cases} 1 & x > 26^\circ\text{C} \\ (x - 18)/8 & 18^\circ\text{C} \leq x \leq 26^\circ\text{C} \\ 0 & x < 18^\circ\text{C} \end{cases} \quad (4.4)$$

који није јединствен већ је дат само као пример јер све зависи од интуиције онога ко га формира. Оваква дефиниција много је ближа начину на који човек доживљава и описује осећај топлоте. Упоредни приказ припадности скупа *toplo* у класичној и фази теорији је дата на слици 4.2.



Слика 4.2 Функције припадности класичног скупа (лево) и фази скупа (десно)

4.1.2 Функција припадности

Сви чланови фази скупа имају одређену (степенасту или процентуалну) припадност том скупу. Колика је припадност и ком скупу припада нека променљива одређује управо функција припадности.

Број функција припадности и њихов облик, за сваку улазну или излазну променљиву фази система, могу бити различити. Већи број функција припадности може побољшати квалитет управљања али истовремено захтева и повећање сложености. Излазне променљиве фази система обично захтевају већу резолуцију од осталих променљивих. Већа резолуција подразумева већи број функција припадности, као и бољу дискретизацију области дефинисаности. Приликом пројектовања фази система се највише примењује интуиција и искуство. Искуствено је познато да извесних преклапања треба да буде како систем не би ушао у недефинисано стање. Однос преклапања функција је искуствена процена у зависности од познавања система у домену проблема.

Функције припадности са линеарном део по део карактеристиком се чешће користе него нелинеарне. Један од разлога је једноставност ових функција која често омогућује унапред лакше предвиђање излаза система. Други разлог је тај што нелинеарност коју уводе фази скупови вишег реда а која захтева већу потрошњу ресурса, није директно пропорционална квалитету излаза фази модела. Међутим, проблем избора функције припадности још није теоријски решен. Различити истраживачи бирају бројне облике у различитим проблемима примене [68].

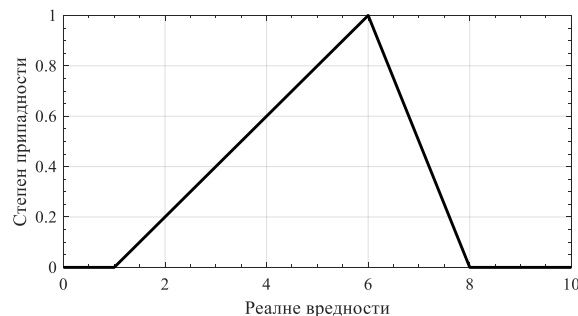
Троугаона функција припадности је део по део линеарна и континуална функција која је дефинисана са три параметра a , b и c уз услов $a \leq b \leq c$. Параметри a , b и c представљају темена троугла односно a и c параметри одређују подножје а параметар b врх троугла. Степен припадности фази скупу који је описан са троугаоном функцијом припадности [71], [75] се у општем случају добија по изразу

$$\mu(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (4.5)$$

или у експлицитном облику по изразу

$$\mu(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (4.6)$$

Пример троугаоне функције припадности са параметрима $a=1$, $b=6$ и $c=8$ приказан је на слици 4.3.



Слика 4.3 Пример троугаоне функције припадности

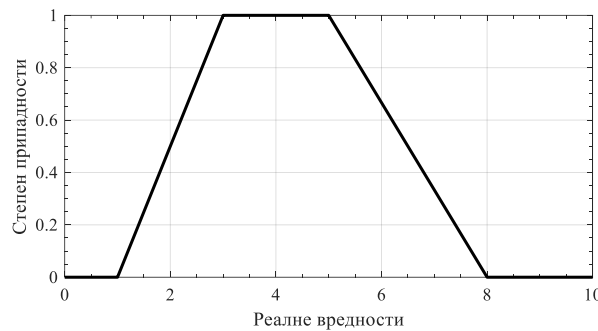
Трапезоидна функција припадности је део по део линеарна и континуална функција која је дефинисана са четири параметра a , b , c и d уз услов $a \leq b \leq c \leq d$. Параметри a и d одређују подножје а параметар b и c врх трапеза. У општем случају степен припадности фази скупу који је описан са трапезоидном функцијом припадности [71], [75] се рачуна по изразу

$$\mu(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (4.7)$$

или у експлицитном облику по изразу

$$\mu(x; a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \quad (4.8)$$

Пример трапезоидне функције припадности са параметрима $a=1$, $b=3$, $c=6$ и $d=8$ је приказан на слици 4.4.

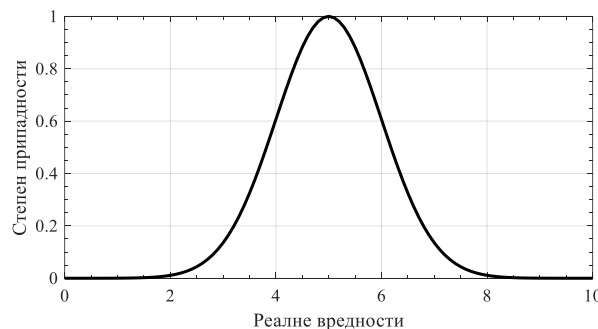


Слика 4.4 Пример трапезоидне функције припадности

Гаусова функција припадности је нелинеарна континуална функција која је дефинисана са два параметра σ и c . У општем случају степен припадности фази скупу који је описан са Гаусовом функцијом припадности [75] се рачуна по изразу

$$\mu(x; \sigma, c) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \quad (4.9)$$

Параметар σ представља ширину а параметар c положај врха. На слици 4.5 је приказан пример Гаусове функције припадности са параметрима $\sigma=1$ и $c=5$.

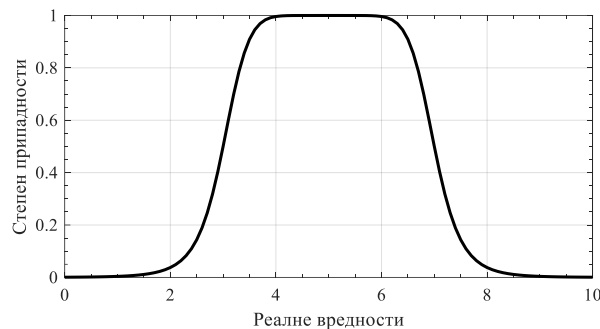


Слика 4.5 Пример Гаусове функције припадности

Звонаста функција припадности је нелинеарна континуална функција која је дефинисана са три параметра a , b и c уз услов да је $b > 0$. Функција је добијена директним уопштењем Кошијеве дистрибуције коришћене у теорији вероватноће па се може и назвати Кошијевом функцијом припадности. У општем случају степен припадности фази скупу који је описан са звонастом функцијом припадности [75] се рачуна по изразу

$$\mu(x; a, b, c) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}} \quad (4.10)$$

Пример звонасте функције припадности са параметрима $a=2$, $b=4$ и $c=5$ се може видети на слици 4.6.

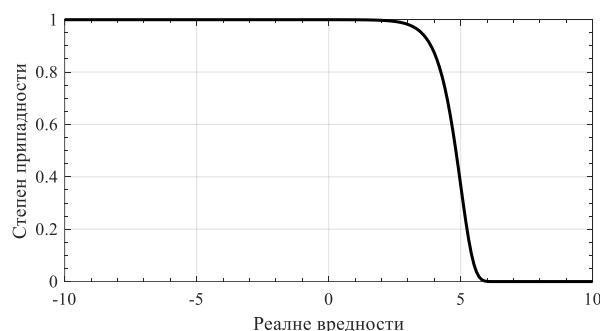


Слика 4.6 Пример звонасте функције припадности

Гаусова и звонаста функција припадности нису у могућности да специфицирају асиметричност која је важна у одређеним применама. Зато се дефинише сигмоидална функција припадности користећи два параметра a и c . У општем случају степен припадности фази скупу који је описан са сигмоидалном функцијом припадности [75] се рачуна по изразу

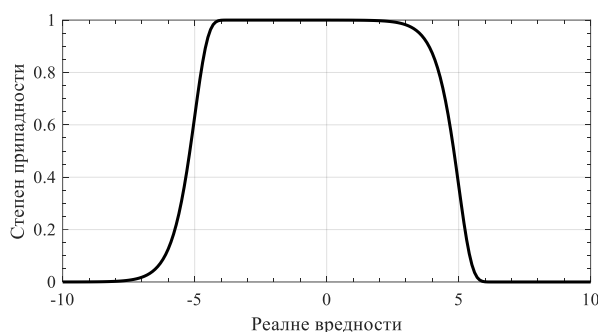
$$\mu(x; a, c) = \frac{1}{1 + e^{-a(x-c)}} \quad (4.11)$$

Пример сигмоидалне функције припадности са параметрима $a=2$ и $c=5$ се може видети на слици 4.7.



Слика 4.7 Пример сигмоидалне функције припадности

На овај начин се добија сигмоидална отворена функција а затворена сигмоидална функција припадности се добија апсолутном разликом или производом две сигмоидалне функције. Пример затворене сигмоидалне функције припадности добијене као разлика две сигмоидалне функције са параметрима $a=2$ и $c=5$ односно $a=2$ и $c=-5$ је приказан на слици 4.8.



Слика 4.8 Пример затворене сигмоидалне функције припадности

4.1.3 Фази операције

Фази скуп је уопштење класичног скупа, стога се операције могу извести као проширења одговарајућих операција са класичним скуповима. Најзначајније операције су једнакост, комплемент, пресек и унија и овде су дате као што су дефинисане у [19].

За фази скупове A и B каже се да су једнаки ако и само ако је $\mu_A(x) = \mu_B(x)$ за све елементе x из скупа X над којим су дефинисани фази скупови A и B .

Фази скуп A је подскуп фази скупа B ($A \subset B$) ако и само ако је $\mu_A(x) \leq \mu_B(x)$ за све елементе x из скупа X .

Фази скуп A' је комплемент фази скупа A ако важи $\mu_{A'}(x) = 1 - \mu_A(x)$.

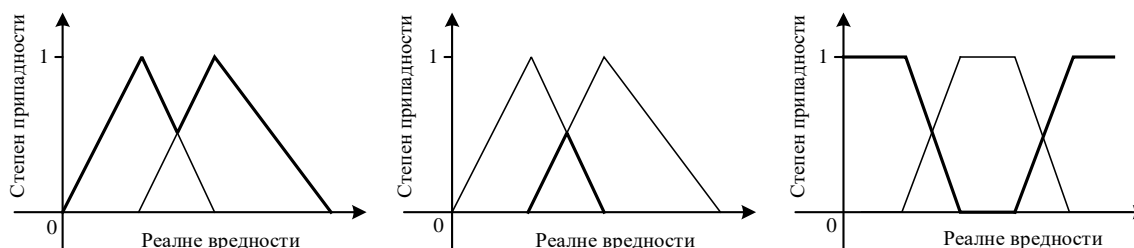
Унија фази скупова A и B ($A \cup B$) представља најмањи фази скуп који истовремено садржи све елементе из фази скупа A и све елементе из фази скупа B . Функција припадности уније $A \cup B$ је облика

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, \forall x \in X \quad (4.12)$$

Пресек фази скупова A и B ($A \cap B$) дефинише се као највећи фази скуп који је истовремено подскуп фази скупа A и фази скупа B . Функција припадности пресека $A \cap B$ је облика

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}, \forall x \in X \quad (4.13)$$

Графички приказ операција уније, пресека и комплемента је дат на слици 4.9.



Слика 4.9 Графички приказ операција над фази скуповима

4.1.4 Лингвистичке променљиве

Заде је предложио концепт лингвистичке променљиве кроз трилогију радова [76]–[78] и дефинисао је као променљиву чије су дозвољене вредности речи природног језика. Лингвистички изрази представљају везу између нумеричког представљања информација и човековог начина размишљања.

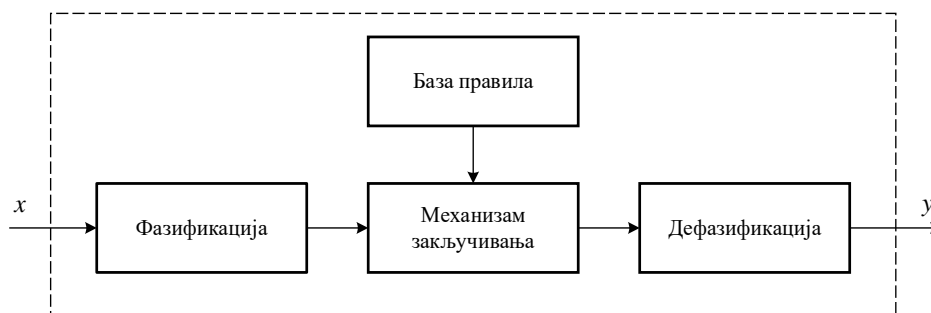
На пример, ако физичка променљива „време успона“ у одскочном одзиву система може да има вредности: „веома мали“, „мали“, „средњи“, „велики“ и „веома велики“, онда је „време успона“ лингвистичка променљива а „мали“, „средњи“ и „велики“ су основне лингвистичке вредности. Испред ових лингвистичких вредности често се додају модификатори као што је у овом примеру „веома“, да би се боље описала основна лингвистичка вредност. Лингвистички модификатори тако врше и модификацију функција припадности а математички се дефинишу у зависности од употребе и намене.

Лингвистичка променљива је дефинисана уређеном петорком (x, T_x, U, G, M_x) [66], која садржи

- x назив лингвистичке променљиве
- T_x скуп лингвистичких вредности (термина, израза) које може попримити лингвистичка променљива x
- U је стварна физичка величина у којој елементи из T_x попримају нумеричке вредности
- G је скуп синтаксних правила који генеришу скуп T_x из скупа основних термина
- M_x је семантичка функција која даје квантитативно значење лингвистичким изразима односно $\forall x \in T$ придружује фази подскуп од U .

4.2 Структура фази система

Фази системи који се разматрају у овој дисертацији имају n улаза $x_i \in X_i$, где је $i = 1, 2, \dots, n$ а X_i је домен за x_i и m излаза $y_j \in Y_j$, где је $j = 1, 2, \dots, m$ а Y_j је домен за y_j . Фази систем који је базиран на Мамдани механизму закључивања [20], [79], [80] има структуру приказану на слици 4.10.



Слика 4.10 Структура фази система

Улази x и излази y имају реалне вредности. Блок фазификација претвара реалне (нумеричке) вредности улаза у фази вредности. Механизам закључивања користи правила у бази правила за претварање ових фази вредности у излазне фази вредности. Блок дефазификација претвара резултат закључивања у реалне (нумеричке) вредности за сваки од излаза.

4.2.1 Фазификација

Фазификација (*fuzzification*) је процес претварања тренутне вредности улазне променљиве у фази скуп са одређеним степеном припадности помоћу функције припадности да би је начинио компатибилном са фази представом променљиве у бази правила.

Фазификација представља пресликавање нумеричких вредности улаза x у фази скуп

$$F: X \rightarrow X^{FUZ} \quad (4.14)$$

где су са X^{FUZ} представљени сви фази скупови који се могу дефинисати над доменом X . Придруживање фази скупа A_i^{FUZ} променљивој x_i може се представити релацијом

$$F(x_i) = A_i^{FUZ} \quad (4.15)$$

Посебан облик фазификације представља “синглтон” (*singleton*) фазификација [81]. Производ овог пресликавања је скуп A_i^{FUZ} чија функција припадности узима само једну дискретну вредност

$$\mu_{A_i}^{FUZ}(a) = \begin{cases} 1 & a = x_i \\ 0 & a \neq x_i \end{cases} \quad (4.16)$$

4.2.2 База правила

База знања садржи базу података и скуп фази АКО-ONDA правила. База података је декларативни део базе знања који описује дефиницију објеката (чињеница, појмова) и дефиницију функција припадности које се користе у фази правилима. Фази правила су процедурални део базе знања који садржи информације о томе како се ови објекти могу користити за формирање закључака [82].

При развоју базе правила се мора водити рачуна о томе да правила не буду контрадикторна. Велики број правила знатно усложњава структуру фази система. Због тога је потребно задржати само релевантна и значајна правила. Базу правила треба редуковати на најмању могућу меру уз минималан губитак корисних информација.

Правила се формирају да би се изразила последица коју производе тренутне вредности улазних променљивих. Ова правила имају облик АКО-ONDA и у општем формату су дата као АКО <премиса> ONDA <последица>.

У складу са тим може се формирати следећа правила као пример:

1. АКО A ЈЕ A_1 I АКО B ЈЕ B_1 ONDA C ЈЕ C_1
2. АКО A ЈЕ A_2 I АКО B ЈЕ B_2 ONDA C ЈЕ C_2

При чему су A_1 и A_2 вредности лингвистичке променљиве A , а B_1 и B_2 вредности лингвистичке променљиве B , док су C_1 и C_2 вредности лингвистичке променљиве C .

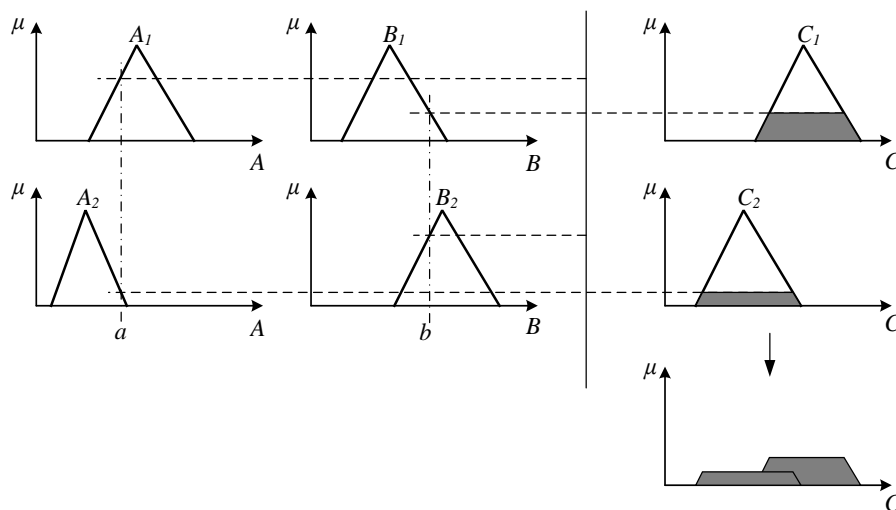
Очигледно је да ова правила не могу да буду сасвим прецизна. То је зато што променљиве узимају лингвистичке вредности и то из ограниченог скупа па самим тим је и скуп правила ограничен.

4.2.3 Апроксимативно закључивање

Апроксимативно закључивање је поступак закључивања који се користи за извођење закључака из скупа нејасних правила АКО-ONDA и једне или више премиса. То је генерализација класичног резоновања са типичним традиционалним правилима закључивања као што су модус поненс (*modus ponens*), модус толленс (*modus tollens*) и хипотетички силогизам. На основу познате чињенице A и правила $A \rightarrow B$ (ако A онда B), показује се да важи закључак B [82].

Апроксимативно закључивање је форма фази логике која садржи скуп правила закључивања чије су премисе фази пропозиције. То је облик закључивања који нуди пуно природнији оквир за људско резонување од традиционалне логике.

Механизам апроксимативног закључивања обрађује фази вредности у процесу који се састоји од три фазе: импликација (активација), агрегација и акумулација. На слици 4.11 је дат графички приказ процедуре Мамдани апроксимативног закључивања [82].



Слика 4.11 Графички приказ Мамдани апроксимативног закључивања

Активација је поступак којим се на основу степена припадности променљивих улазних фази скупова (вредности добијене фазификацијом) рачуна степен активације сваког појединачног правила. Активација на тај начин одлучује колико ће поједини излазни фази скупови учествовати у наредном кораку импликације. Она је уједно и мера колико свеукупно улазне променљиве припадају улазним фази скуповима који учествују у датом правилу. Код правила И (AND) типа се најчешће активација правила рачуна методом минимума односно као најмањи степен припадности од свих функција припадности које учествују у том правилу. Постоји и метода производа када се активација правила рачуна као производ свих степена припадности свих функција припадности које учествују у том правилу. Код правила ИЛИ (OR) типа се најчешће активација правила рачуна методом максимума односно као највећи степен припадности од свих функција припадности које учествују у том правилу. Постоји и метода алгебарске суме када се активација правила рачуна као разлика збира и производа свих степена припадности свих функција припадности које учествују у том правилу. На слици 4.11 је резултат активације за сваки фази скуп улазне променљиве представљен хоризонталном испрекиданом линијом за свако правило.

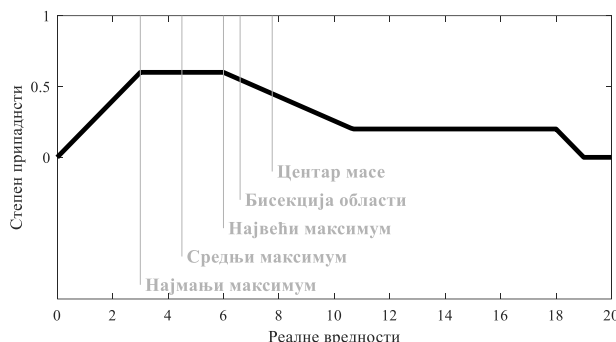
Импликација је поступак у оквиру којег се на основу вредности добијених активацијом формира излазни фази скуп за сваку излазну променљиву сваког правила. Најчешће коришћени метод импликације је метод минимума који одсеца врх излазног фази скупа на вредност добијену активацијом. Друга широко примењивана метода импликације је метода производа која фази скуп множи са вредношћу добијеном активацијом односно све вредности степена припадности се множе добијеном вредношћу (пропорционално смањење). На слици 4.11 резултат импликације (MIN метод) за свако правило је представљен као осенчени део фази скупа излазне променљиве.

Агрегација је поступак у оквиру којег се на основу свих фази променљивих добијених поступком импликације, којих има онолико колико има активних правила,

формира коначна излазна фази променљива. Сви активни закључци се у том смислу акумулирају на различите начине. Најчешће коришћене методе агрегације су метода максимума, метода суме и пробабилистичко ИЛИ. Метода агрегације се често комбинује са методом импликације па су тако познати типови механизма апроксимативног закључивања као што су MIN-MAX (метода минимума у поступку импликације и метода максимума у поступку агрегације) или PROD-SUM (метод производа у поступку импликације и метода суме у поступку агрегације). На слици 4.11 је резултат агрегације (MAX метод) приказан у доњем десном углу.

4.2.4 Дефазификација

Дефазификација представља процес супротан процесу фазификације који треба да претвори резултат агрегације у сигнал који је разумљив другом систему. Излаз фази система мора да има једну јединствену вредност, најчешће представљену реалним бројем. Методе које се најчешће користе за дефазификацију су центар масе (гравитације, површи), бисекције области, најмање и највеће вредности максимума и средине максимума [83]. На слици 4.12 се може видети графички приказ метода дефазификације над истим резултатом агрегације.



Слика 4.12 Графички приказ метода дефазификације

Критеријум центра површи (*Center Of Area/Gravity, COA/COG*) је најпознатији и најчешћи дефазификациони метод који у дискретном облику гласи

$$u = \frac{\sum_{i=1}^k \mu(x_i)x_i}{\sum_{i=1}^k \mu(x_i)} \quad (4.17)$$

где x_i представља тачку у дискретном скупу, а $\mu(u_i)$ представља вредност припадности фази скупу. За континуалан случај треба уместо суме користити интеграл. То је често коришћена метода иако је њена рачунска сложеност релативно висока. Центар масе је тачка на апсиси око које би овај облик био у равнотежи под условом да ту површину посматрамо као раван једнаке густине.

Критеријум бисекције области (*Bisection Of Area, BOA*) даје тачку која дели излазну област на два дела једнаких површина и у континуалном случају гласи

$$u = \left\{ x \mid \int_{\alpha}^{z_0} \mu_c(x) dx = \int_{z_0}^{\beta} \mu_c(x) dx \right\} \quad (4.18)$$

где x представља тачку из скупа, $\mu_c(x)$ његову вредност припадности, а $\alpha = \min\{x \mid x \in C\}$ и $\beta = \max\{x \mid x \in C\}$. *Min* је крајње лева вредност скупа, а *max* је крајње десна вредност скупа. Рачунање је доста сложено и може бити двосмислено. Ако фази скуп садржи

на пример два синглтона било која тачка између њих дели интервал на два дела. Тако се према томе може рећи да у дискретном случају није дефинисана.

Критеријум најмање вредности максимума (*First Of Maximum, LOM*) даје најмању (прву) тачку у којој резултат агрегације достиже максималну вредност. Критеријум највеће вредности максимума (*Last Of Maximum, LOM*) даје највећу (последњу) тачку у којој резултат агрегације достиже максималну вредност. Критеријум средине максимума (*Middle Of Maximum, MOM*) даје средњу тачку у којој могућност расподеле резултата агрегације достиже максималну вредност. Ако резултат агрегације има јединствен максимум онда ће ове три методе вратити исту вредност. Ове методе су неосетљиве на облик фази скупа и њихово рачунање је веома једноставно.

4.3 Сугено фази систем закључивања

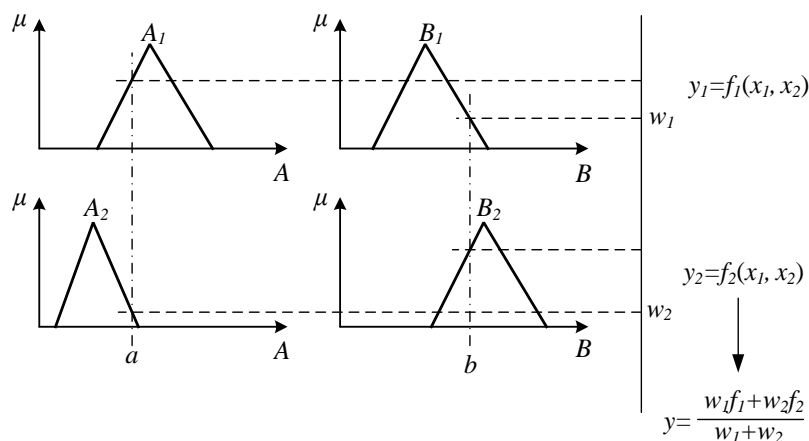
Алтернативни тип фази система, познат као Сугено (Такаги-Сугено-Канг) фази систем предложен је у [84], [85] у настојању да се развије систематски приступ апроксимацији нелинеарних функција. Уместо фази скупа у закључку код Сугено типа фази закључивања се налази линеарна функција између улаза и излаза.

Пример рада Сугено фази систем ће бити дат кроз базу правила:

1. АКО A ЈЕ A_1 И АКО B ЈЕ B_1 ОНДА $y=f_1(a, b)$
2. АКО A ЈЕ A_2 И АКО B ЈЕ B_2 ОНДА $y=f_2(a, b)$

При чему се A_1 и A_2 вредности лингвистичке променљиве A , а B_1 и B_2 вредности лингвистичке променљиве B док је f функција зависности излаза од улаза. Када су линеарни коефицијенти у функцији f једнаки нули то се може посматрати као посебан случај Мамдани фази система, где је последица сваког правила специфицирана фази синглтоном (singleton) [86].

Графички приказ Сугено апроксимативног закључивања је дат на слици 4.13. Излазна вредност се добија пондерисаним просеком нумеричких резултата фази правила. На овај начин се избегава дуготрајна процедура дефазификације код Мамдани закључивања [82].



Слика 4.13 Графички приказ Сугено апроксимативног закључивања

Једноставнији начин добијања излаза код Сугено фази система омогућује лакшу имплементацију на контролерима у практичним применама. Уз мању потрошњу ресурса приликом извршавања он омогућује и брже извршавање, што погодује захтевним применама.

5 Аутоматско подешавање параметара

Најбољи начин подешавања параметара ПИД регулатора за постизање оптималних перформанси је и даље отворено питање иако је ПИД закон управљања усвојен као стандард за управљање процесима у затвореној спреси.

Избор методе за подешавања параметара зависи од низа фактора, укључујући динамичко понашање управљаног процеса, перформанси које се очекују, као и доступног времена и предзнања за примену методе. Како би се смањило време и знање неопходно за подешавање параметара ПИД регулатора развијене су методе аутоматског подешавања.

Аутоматско подешавање параметара ПИД регулатора је процедура за аутоматски прорачун параметара, без икакве интеракције корисника, која се покреће на захтев. Након завршетка процедуре регулатор наставља да ради са новим параметрима до следећег евентуалног поновног покретања. За разлику од континуиране адаптације у оквиру које се ради континуиран прорачун нових параметара без интеракције са корисником. Међутим, ова два термина се често користе наизменично јер се и у оквиру процеса адаптације и аутоматског подешавања параметри регулатора добијају аутоматски [8], [87]–[89].

Технике за аутоматско подешавање су проистекле из истраживања адаптивног управљања. Адаптација је првобитно развијена за процесе са карактеристикама које су се мењале током времена или са условима рада. Истраживање општих приступа адаптивном управљању је започето у [90] док је у [91] предложен нови приступ овој области у виду самоподешавајућих регулатора заснованих на теорији оптималног управљања примењеној на општи линеарни модел система n -тог реда. Практично све адаптивне технике се могу користити за аутоматско подешавање.

Интензивнији развој метода за аутоматско подешавање започет је са истраживањем [10] у којем је представљена метода заснована на релејном експерименту који се веома лако могао имплементирати на комерцијалним контролерима. Након тога су предложена многа унапређења релејног експеримента и уопште метода за аутоматско подешавање што се може видети у упоредној анализи представљеној у [18].

Уопштено, приступи аутоматском подешавању параметара ПИД регулатора се могу поделити у приступе засноване на моделу и приступе засновани на интуитивним правилима. У приступима заснованим на моделима, модел процеса се изводи експлицитно а прорачун параметара се заснива на изведеном моделу користећи једну од доступних метода. У приступима заснованим на правилима, подешавање се ослања на експертски систем који аутоматизује процедуру са правилима која искусни инжењер користи за ручно подешавање параметара регулатора [8].

У овом поглављу биће дат преглед мера за евалуацију перформанси и робусности система. Затим ће бити дат преглед метода базираних на моделу од интереса за дисертацију у поступку аутоматског подешавања параметара ПИД регулатора. Поред метода заснованих на моделу биће дата и основна идеја метода заснованих на интуитивним правилима са посебним освртом на коришћење фази логике у аутоматизацији поступка подешавања параметара регулатора.

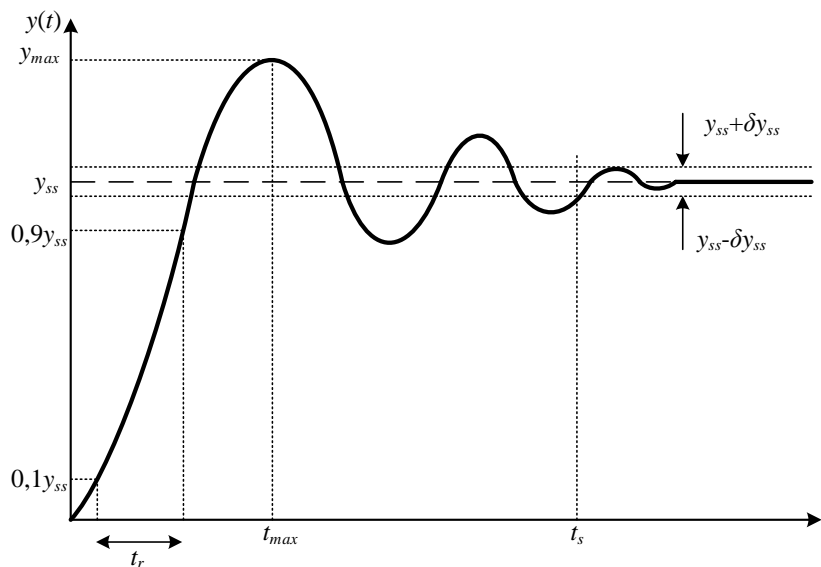
5.1 Перформансе и робусност

Постоје различити квантитативни показатељи перформанси и робусности система. Ти показатељи омогућавају оцену квалитета решења као и упоредну анализу различитих решења.

5.1.1 Мере перформанси

Перформансе система управљања се уобичајено вреднују на основу карактеристика прелазног режима након побуде. Побуда је најчешће у виду јединичног одскочног сигнала који се примењује као референца или као поремећај.

На одзиву система на побуду у виду одскочног сигнала, који је приказан на слици 5.1, означене су карактеристичне тачке на основу који се могу дефинисати мере перформанси, које ће бити дате у наставку.



Слика 5.1 Перформансе система на одскочном одзиву на побуду

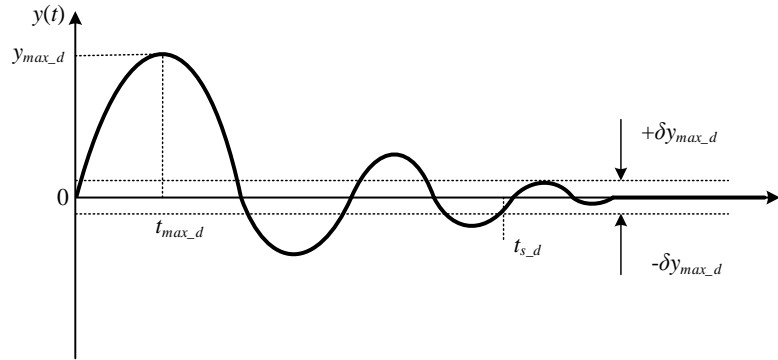
Време успона t_r се дефинише као време које је потребно да се вредност излаза $y(t)$ промени од 10 % до 90 % вредности у устаљеном стању y_{ss} . Време t_{max} се дефинише као тренутак када излаз $y(t)$ достигне своју максималну вредност y_{max} . Прескок $o\%$ се дефинише у процентима на следећи начин

$$o\% = \frac{y_{max} - y_{ss}}{y_{ss}} 100\% \quad (5.1)$$

Време смирења t_s је време потребно да амплитуда излаза $y(t)$ уђе у појас $y_{ss} \pm \delta y_{ss}$. За δ се најчешће усваја 2 % или 5 %.

На одзиву система на поремећај у виду одскочног сигнала који је приказан на слици 5.2 су означене карактеристичне тачке на основу који се могу дефинисати мере перформанси дате у наставку.

Време $t_{max,d}$ се дефинише као тренутак када излаз $y(t)$ достиже своју максималну вредност $y_{max,d}$. Време смирења $t_{s,d}$ је време потребно да амплитуда излаза $y(t)$ уђе у појас $0 \pm \delta y_{max,d}$ при чему се за δ најчешће усваја 2 % или 5 %.



Слика 5.2 Перформансе система на одскочном одзиву на поремећај

Поред набројаних мера перформанси који се дефинишу на основу излаза система постоје мере које се дефинишу на основу сигнала грешке попут интегралних мера перформанси. Један од најчешће коришћених је интеграл апсолутне грешке IAE (*Integral of absolute error*) који је дат изразом

$$IAE = \int_0^{\infty} |e(t)| dt \quad (5.2)$$

где је $e(t)=r(t)-y(t)$ сигнал грешке која представља разлику између задате вредности $r(t)$ и тренутне излазне вредности система $y(t)$. Интеграл апсолутне грешке се дефинише на исти начин и на одзиву на референцу као и на одзиву на поремећај. Код одзива на поремећај додаје се индекс d односно означава се са IAE_d .

5.1.2 Мере робусности

Систем у затвореној спрези треба да има висок ниво робусности односно да услед промене параметара процеса, грешке моделовања и малих нелинеарности одржи стабилност и обезбеди адекватно праћење референце и потискивање поремећаја.

Као квантитативна мера робусности система најчешће се користи максимална осетљивост M_s која је дефинисана изразом

$$M_s = \max_{\omega_s} \left| \frac{1}{1 + W(j\omega)} \right| \quad (5.3)$$

где је $W(j\omega)$ функција повратног преноса а ω_s учестаност на којој се постиже максимална вредност [92].

Вредност M_s се може тумачити као инверзна вредност најкраће удаљености између критичне тачке -1 и Никвистове криве функције повратног преноса. Графичка репрезентација максималне осетљивости представљена је кружницом са центром у критичној тачки са полупречником $1/M_s$ и може видети на слици 5.3.

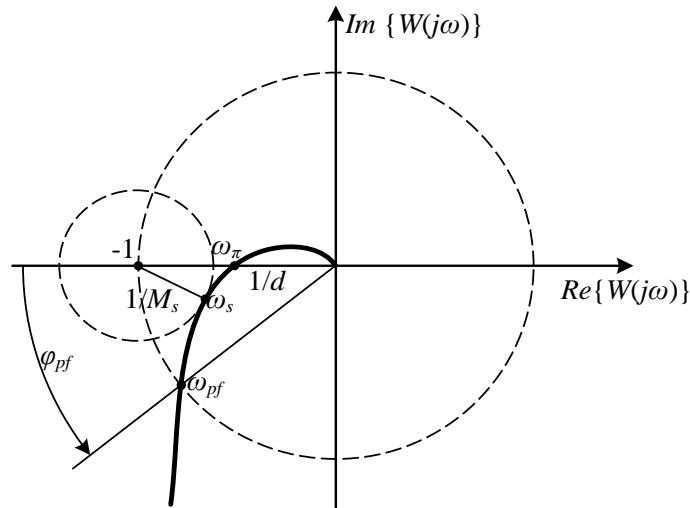
Алтернативно као мере робусности могу се користити претек појачања и претек фазе. Претек појачања се дефинише

$$d = \frac{1}{|W(j\omega_\pi)|} \quad (5.4)$$

где је ω_π учестаност у којој Никвистова крива сече негативни део реалне осе. Претек фазе се дефинише

$$\phi_{pf} = \pi + \text{Arg}\{Wj\omega_{pf}\} \quad (5.5)$$

где је ω_{pf} учестаност у којој модуо функције повратног преноса постиже вредност један односно где Никвистова крива сече јединичну кружницу са центром у координатном почетку. Графичка репрезентација претека појачања и претека фазе се може видети на слици 5.3.



Слика 5.3 Графичка репрезентација мера робустности

5.2 Методе базиране на моделу

Аутоматско подешавање ПИД регулатора може се постићи аутоматизацијом процедуре у оквиру које се користи метода за одређивање динамике процеса у комбинацији са методама за прорачун параметара ПИД регулатора.

Да би се утврдила динамика, процеса процес се мора побудити одскочном, импулсном или синусоидном побудом. На основу одзива одређује се модел процеса или једноставна карактеризација процеса. На основу идентификованих карактеристика процеса приступа се прорачуну параметара коришћењем неке од доступних метода.

5.2.1 Идентификација процеса

Идентификација процеса у поступку аутоматског подешавање параметара ПИД регулатора најчешће се изводи из улазно/излазних података добијених извођењем експеримента над процесом односно намерним стимулисањем процеса у виду одскочне импулсне или простопериодичне побуде.

Најчешће коришћен начин за идентификацију процеса је апроксимација динамике једноставним моделом, најчешће првог реда, чији се параметри могу лако експериментално одредити.

Други приступ је карактеризација процеса заснована на неколико карактеристика које су изабране тако да имају добру физичку интерпретацију и да се истовремено могу лако одредити на експерименталан начин.

У студији [93] је дат преглед развоја метода за идентификацију процеса на веома концизан начин.

5.2.1.1 Апроксимација моделом

За апроксимацију динамике процеса се могу користити различити линеарни модели нижег реда са временским кашњењем [94].

Најчешће коришћен модел за апроксимацију процеса је модел описан процесом првог реда са временским кашњењем (FOPTD - First-Order Process With Time Delay) који има следећу функцију преноса

$$G(s) = \frac{K}{1 + sT} e^{-sL} \quad (5.6)$$

где је K статичко појачање, T је временска константа а L је временско кашњење.

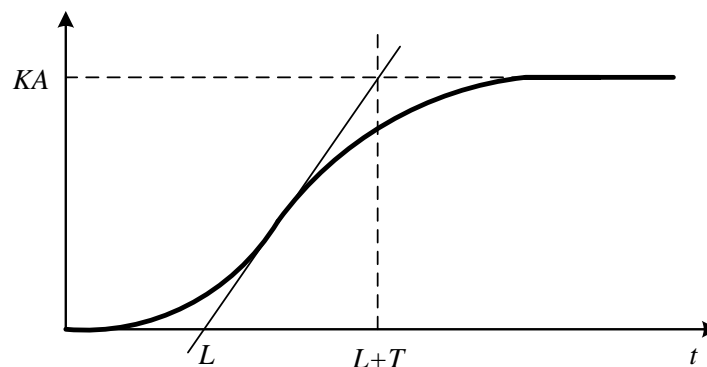
Постоје процеси са динамиком која садржи интегратор или веома велику временску константу а који неће достићи устаљено стање у условима отворене спреге. Комбинацијом интегратора и временског кашњења добија се модел описан интегралним процесом првог реда са временским кашњењем (FOIPTD - First-Order Integral Process With a Time Delay) са функцијом преноса

$$G(s) = \frac{K_v}{s} e^{-sL} \quad (5.7)$$

где је K_v брзинска константа, а L временско кашњење.

Једноставан модел процеса се може одредити из одскочног одзива у отвореној спрузи. Пре почетка експеримента процес мора бити у устаљеном стању. Амплитуда промене треба да буде довољно велика тако да је одзив лако видљив изнад шума мерења, односно да буде што је могуће мања да се процес не би пореметио више него што је потребно. Трајање експеримента је онолико колико је потребно да процес поново уђе у устаљено стање.

Једна од метода за одређивање параметара модела је метода тангенте [95]. Метода је базирана на конструкцији тангенте на одзиву система у тачки прегипа односно у тачки у којој ће извод имати највећу вредност, што је илустровано на слици 5.4.

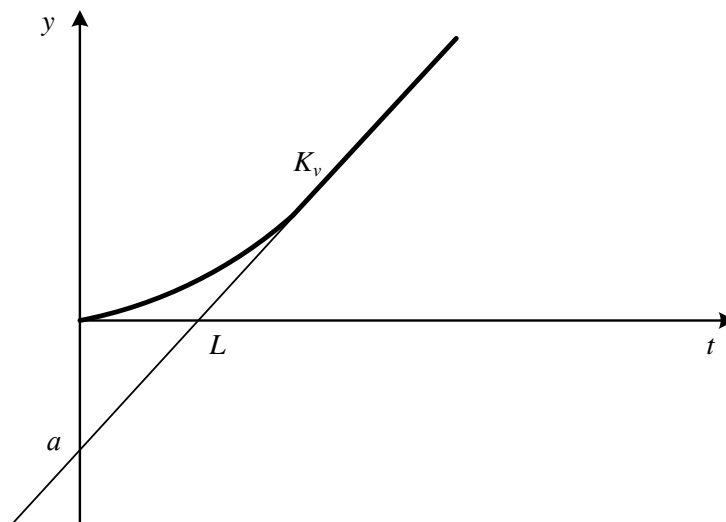


Слика 5.4 Метода тангенте

Статичко појачање процеса K се може одредити као однос промене вредности излаза у устаљеном стању KA и промене вредности улаза A . Тада се процењено време кашњења L одређује као временски интервал између промене улаза и пресека тангенте са временском осом. Вредност $T+L$ је одређена као временски интервал између промене улаза и пресека тангенте са правом која означава вредност у устаљеном стању

излаза система. Из ове вредности се затим временска константа T може веома лако одредити.

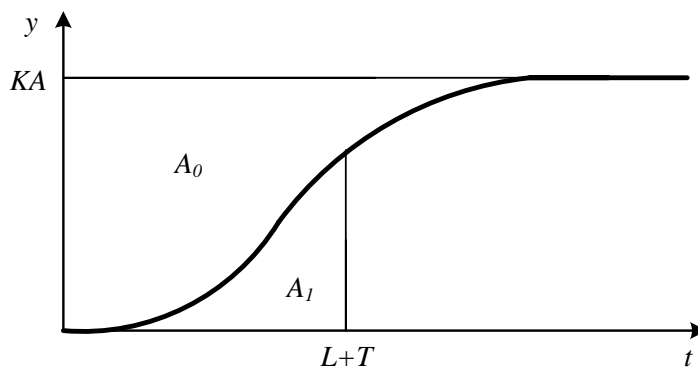
Метода се може применити и на одзиву система који има интегратор а који се у том случају апроксимира са FOIPTD моделом. На исти начин се конструише тангента на одзиву система у тачки прегива односно у тачки у којој ће извод имати највећу вредност што је приказано на слици 5.5. Овде је битно да се та највећа вредност нагиба усвоји као параметар K_v . Процењено време кашњења L одређује као временски интервал између промене улаза и пресека тангенте са временском осом. Док се параметар a одређује у пресеку тангенте са y осом.



Слика 5.5 Метода тангенте за процесе са интегратором

Главни недостатак ове методе је тај што се ослања на једну тачку у одзиву система (тачку прегива) и што је веома осетљива на шум мерења и може изазвати велике грешке у процени тачке прегива и првог извода у тој тачки.

Метода која је мање осетљива на шум мерења је метода површине [95], која се заснива на одређивању површина A_0 и A_1 приказаних на слици 5.6.



Слика 5.6 Метода површина

Узимајући у обзир да се статичко појачање K одређује на исти начин као у претходној методи а да се динамика процеса апроксимира помоћу FOIPTD модела чији је одзив на одскочну побуду дат изразом

$$y(t) = \left(K + Ke^{-\frac{t-L}{T}} \right) h(t-L) \quad (5.8)$$

површине се могу изразити као

$$A_0 = \int_0^{\infty} (K - y(t)) dt = K \int_0^L dt - \int_L^{\infty} Ke^{-\frac{t-L}{T}} dt = KL + KT = K(L + T) \quad (5.9)$$

$$A_1 = \int_0^{L+T} y(t) dt = \int_L^{L+T} \left(K + Ke^{-\frac{t-L}{T}} \right) dt = K(L + T) - KTe^{-1} \quad (5.10)$$

из чега на крају следе изрази

$$L + T = \frac{A_0}{K} \quad (5.11)$$

$$T = \frac{A_1}{K} e \quad (5.12)$$

Недостатак методе је што се површина A_1 не може израчунати док се не одреди површина A_0 и што се не може применити на процесу са интегратором чији одзив не конвергира ка некој сталној вредности.

У студији [96] је показано да се применом методе површина у процесу идентификације постижу генерално боље перформансе система користећи исте методе подешавања параметара ПИД регулатора.

5.2.1.2 Карактеризација у фреквенцијском домену

Други начин за идентификацију је карактеризација у фреквенцијском домену. На овај начин се не одређује експлицитан модел него се процес карактерише помоћу параметра који представљају једну тачку на Никвистовом дијаграму.

Ова метода је прво предложена у [1] и базирана је на експерименталном одређивању тачке граничне стабилности. Ова тачка се може одредити повећањем пропорционалног појачања регулатора све док процес не постане гранично стабилан. Појачање регулатора у овој тачки назива се критично појачање K_u а појачање периода осциловања се назива критична периода T_u . Недостатак методе је то што се процес доводи до границе стабилности, што може довести до опасних ситуација у пракси.

Овај приступ није погодан за аутоматску процедуру јер не постоји контрола над амплитудом осцилација. Као унапређење предложен је релејни експеримент [10] којим се генерише правоугли улазни сигнал на улазу процеса који осцилује са супротном фазом што значи да је периода осциловања критична периода T_u . Критично појачање се одређује из израза

$$K_u = \frac{4h}{\pi A} \quad (5.13)$$

где је са h означена амплитуда правоугаоног сигнала на улазу а са A означена амплитуда осцилација основног хармоника на излазу.

Релејни експеримент је након тога доживео многа унапређења у виду додавања хистерезиса и асиметричности облика што се може видети у [97] где је приказан ток развоја на веома јасан начин. Једно од унапређења је и могућност одређивања параметара FOPD модела из релејног експеримента чиме се постиже још већа

употребљивост [98]. У [99] је предложена процедура без прекидања повратне спреге са регулатором, коришћењем два филтера у каскади.

Поред вредности критичног појачања и критичне учесталости у [100] је предложено одређивање и угла тангенте и статичног појачања система као додатне карактеризације процеса. Вредности ових параметара се могу одредити из процењене Лапласове трансформације одскочног одзива система, што је приказано у оквиру студије објављене у [101].

5.2.2 Методе подешавања

На основу експериментално одређених карактеристика процеса, параметри ПИД регулатора се могу одредити применом једне од доступних метода подешавања. Током година развоја почевши од прве забележене методе [1] велики број правила за подешавање је осмишљен након тога. Преглед метода за подешавање параметара је на веома концизан начин представљен у [9] док ће у наставку бити приказане методе које су од интереса за дисертацију.

5.2.2.1 Циглер-Николсова метода

Циглер-Николсова метода [1], [2] је прва објављена метода за подешавање параметара ПИД регулатора. Међу најзаступљенијим је методама које се примењују у пракси па је усвојена као једна од метода коришћених у истраживању.

Кључни критеријум који је примењен и усвојен за пројектовање регулатора је пригушење амплитуде одзива за четвртину у односу на претходну вредност. Ради једноставности, Циглер-Николс метода ће у остатку ове дисертације бити означена као Ц-Н метода.

За стабилне процесе параметри се могу добити из следећих израза

$$K_p = \frac{1,2 T}{K L} \quad (5.14)$$

$$T_i = 2L \quad (5.15)$$

$$T_d = \frac{L}{2} \quad (5.16)$$

За процесе са интегратором претходне једначине 5.14-5.16 се могу изразити као

$$K_p = \frac{1,2}{LK_v} \quad (5.17)$$

$$T_i = 2L \quad (5.18)$$

$$T_d = \frac{L}{2} \quad (5.19)$$

5.2.2.2 Острем-Хеглундова метода

Острем-Хеглундова метода [6] је једна од последње објављених метода за подешавање параметара ПИД регулатора која постиже веома добре резултате над широким скупом процеса. Задржала је исту једноставност као друге методе сличне Циглер-Николсовој методи уз побољшање перформанси па је усвојена као једна од метода коришћених у истраживању.

Острем и Хеглунд су користили исте опште идеје усвајајући критеријум максимизирања интегралног појачања уз ограничење максималне осетљивости. Овај метод

је први пут коришћен у [5] за ПИ регулатор а исти приступ је са додатним ограничењима касније коришћен у [6] за ПИД регулатор. Ради једноставности, Острем-Хеглундова метода ће у остатку ове дисертације бити означена као О-Х метода. За стабилне процесе ПИД параметри се могу изразити применом следећих израза

$$K_p = \frac{1}{K} \left(0,2 + 0,45 \frac{T}{L} \right) \quad (5.20)$$

$$T_i = \frac{0,4L + 0,8T}{L + 0,1T} L \quad (5.21)$$

$$T_d = \frac{0,5LT}{0,3L + T} \quad (5.22)$$

За интегралне процесе изрази 5.20-5.22 се могу написати као

$$K_p = \frac{0,45}{K_v} \quad (5.23)$$

$$T_i = 8L \quad (5.24)$$

$$T_d = \frac{L}{2} \quad (5.25)$$

5.3 Методе базиране на правилима

За разлику од приступа заснованих на моделу, у приступима заснованим на правилима, експлицитан модел није неопходан, него се подешавање заснива на правилима сличним онима која искусни инжењер користи када ручно подешава регулатор. Аутоматизацијом овог поступка кроз увођење експертског система изводе се методе за аутоматско подешавање параметара ПИД регулатора које нису засноване на моделу процеса него на интуитивним правилима.

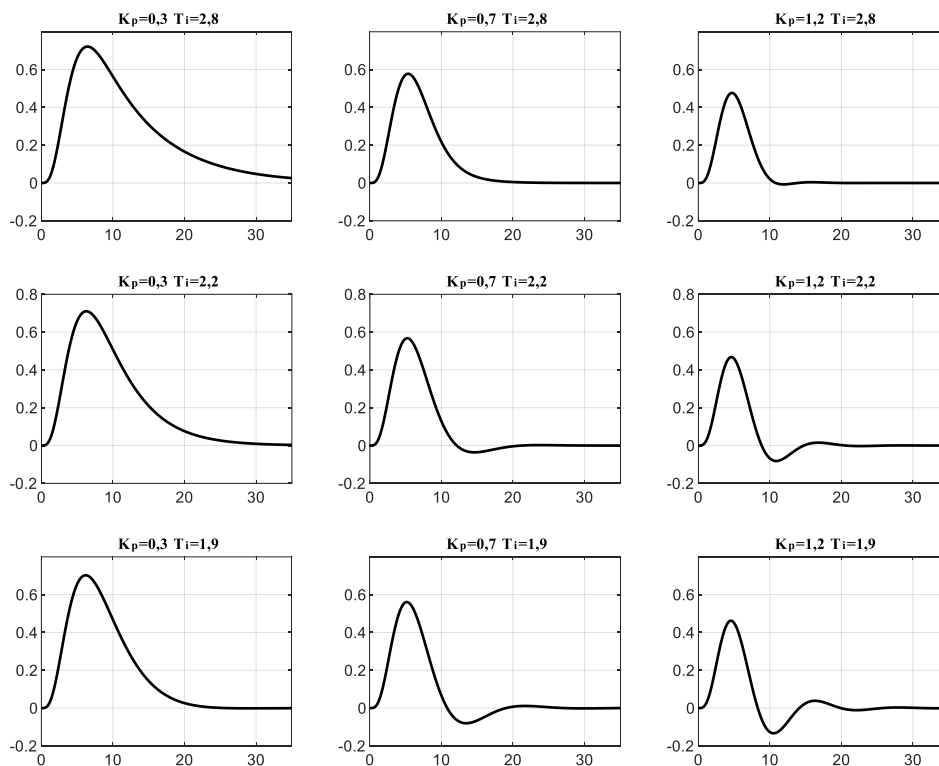
Ручно подешавање је коришћено пре него што су биле доступне било какве систематске методе подешавања. Касније је постало неопходна допуна класичним методама које нису увек гарантовале предвиђене перформансе па је често било потребно применити накнадно прецизно подешавање параметара да би се добиле боље перформансе. Ручно подешавање се обично ради кроз експерименте са системом у затвореној спреси. Посматра се одзив система па се на основу процене модификују параметри регулатора.

Модификације су засноване на смерницама у виду једноставних правила [8]. Правила се добијају опсежним експериментисањем и као пример су дата:

- Повећање пропорционалног појачања нарушава стабилност
- Грешка се брже смањује ако се временска константа интегралења смањи
- Смањење временске константе интегралења смањује стабилност
- Повећање временске константе диференцирања побољшава стабилност

Како би се лакше формализовала правила и стекла интуиција о понашању процеса формирају се мапе подешавања. Мапе подешавања су матрице одзива система у временском или фреквенцијском домену које одговарају систематским варијацијама у ПИД параметрима чија је суштинска функција да пружи увид у то како промене у параметрима регулатора утичу на понашање система у затвореној спреси [8].

Пример мапе подешавања са одзивима на јединични одскочни поремећај за процес са функцијом преноса $G(s) = \frac{1}{(s+1)^4}$ је дата на слици 5.7. На слици је илустровано како за подешену вредност $T_d=0,7$ промена појачања K_p и временске константе интегралне T_i утиче на одзив на поремећај. Може се закључити да са повећањем K_p долази до смањења максималне вредности али повећања осцилација. Такође са повећањем T_i долази до смањења осцилација уз задржавање максималне вредности и смањења времена смирења.



Слика 5.7 Пример мапе подешавања

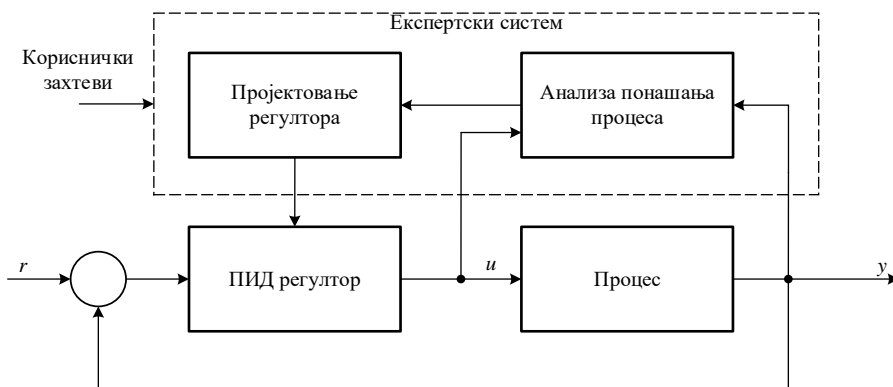
5.3.1 Методе засноване на фази логици

Већина регулатора у раду данас је пројектована коришћењем конвенционалних метода. Међутим, постоје многе ситуације у којима ови регулатори нису правилно подешени а доступно је хеуристичко знање о томе како их подесити док су у нормалном раду. Тада постоји могућност да се користе методе фази логики у експертском систему који може да води процес подешавања конвенционалних регулатора [67].

Конвенционални фази регулатори који генеришу управљачку акцију на основу синтетизованог лингвистичког управљачког механизма искусног оператера се први пут помињу у [20]. Међутим, фази експертски систем иако дели доста тога са фази контролером има доста специфичности. Једна од специфичности је излаз који није директни управљачки улаз у процес већ излаз који диктира промене параметара регулатора. Експертски систем као такав настоји да посматра тренутно понашање управљачког система и модификује параметре регулатора како би побољшао перформансе. Ова идеја је почела да се примењује почевши од [36] па затим кроз низ радова [39]–[48]

Принципијелна шема аутоматског подешавања параметара ПИД регулатора употребом фази експертског система која се јавља у доступној литератури је облика приказаног на слици 5.8. Блок за анализу понашања настоји да окарактерише тренутно

понашање процеса на начин који ће бити користан блоку за пројектовање регулатора. Целокупни експертски систем може бити имплементиран као фази систем са једноставним правилима где премисе правила чине део препознавања понашања, а последице формирају блок за пројектовање ПИД регулатора [67].



Слика 5.8 Принципијелна шема аутоматског подешавања параметара ПИД регулатора

Кандидати за правила за такав фази систем могу укључивати следеће:

- Ако је одговор осцилаторан онда треба повећати вредност диференцијалног дејства
- Ако је време успона мало онда треба повећати пропорционално појачање
- Ако је прескок велик онда треба смањите пропорционално појачање

Док изгледа да таква правила нуде веома једноставно решење за проблем аутоматског подешавања параметара ПИД регулатора, није увек лако измерити количине које су наведене у горњим претпоставкама. На пример, како прецизно квантификујемо „мало“ време успона? Како измерити степен осцилаторног понашања? Како израчунати прескок у реалном времену када не може да се гарантује да ће систем имати нулту грешку у устаљеном стању?

У поступку аутоматског подешавања систем генерише скуп референтних побуда (нпр. одскочна) и посматра и анализира одзив система на побуду да би одредио вредности премиса горњих правила. Затим се примењује механизам апроксимативног закључивања на основу базе правила и прорачунају се нови параметри. Управљачки систем у затвореној спреси се ставља у рад са овим новим параметрима.

Примарна предност овог приступа у аутоматском подешавању параметара ПИД регулатора је то што се веома брзо може пројектовати систем са основним скупом правила али са ограниченом применом у пракси. Међутим, пројектовање експертског система који ће моћи да се користи над широким скупом различитих система представља велики изазов који је и предмет ове дисертације.

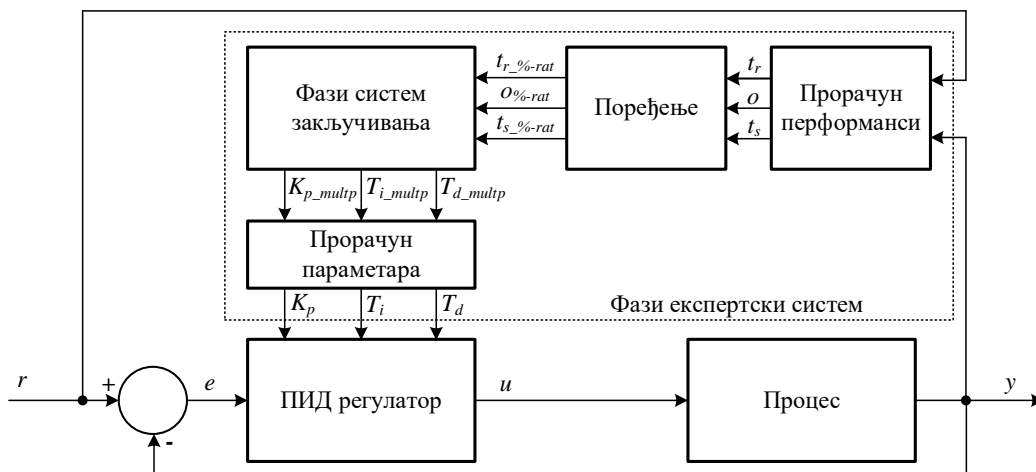
6 Предложена метода за аутоматско подешавање параметара

Предложена метода за аутоматско подешавање параметара ПИД регулатора је заснована на ставовима и идејама изнетим у поглављу 5.3.1. За разлику од других претходно поменутих метода [37]–[39] које користе сигнал грешке и први извод сигнала грешке, предложена метода користи време успона, прескок и време смирења као улаз у фази експертски систем. Поред тога, предложена метода функционише као класична метода аутоматског подешавања и неће бити намењена за континуирано прилагођавање за разлику од већине метода које користе концепт фази логике. На крају, предложена метода омогућава промену режима рада што је издваја од сличних метода за подешавање параметара [39]–[48]. Избор режима рада подиже употребљивост предложене методе на виши ниво омогућавајући постављање циљева у процедури подешавања параметара.

Предложена метода је у неизмењеном облику објављена у [102] и у овом поглављу ће бити разложена на делове од којих ће сваки бити детаљно описан. Потом ће бити дат опис процедуре рада.

6.1 Структура методе

Блок дијаграм предложеног фази експертског система (ФЕС) за аутоматско подешавање параметара ПИД регулатора приказан је на слици 6.1. Састоји се од дела за прорачун перформанси, дела за поређење, фази система закључивања и дела за прорачун параметара.



Слика 6.1 Блок дијаграм предложене методе

Главна сврха дела за прорачун перформанси је прорачун перформанси одскочног одзива система у затвореној спреси на основу тренутне вредности y и задате вредности r . Перформансе одзива система у затвореној спреси су евалуиране коришћењем времена успона t_r , прескока o и времена смирења t_s . У блоку за поређење се упоређују

тренутне вредности перформанси са референтним вредностима које су сачуване у првој итерацији. Као излаз се добија процентуални количник времена успона t_s , прескока $o\%-rat$ и времена смирења $t_{s\%-rat}$, који се могу изразити на следећи начин

$$t_{r\%-rat} = \frac{t_r}{t_{r_ref}} 100 \% \quad (6.1)$$

$$o\%-rat = \frac{o}{o_ref} 100 \% \quad (6.2)$$

$$t_{s\%-rat} = \frac{t_s}{t_{s_ref}} 100 \% \quad (6.3)$$

Фази систем закључивања на основу процентуалних количника даје множитеље ПИД параметара K_{p_multp} , T_{i_multp} и T_{d_multp} користећи експертско знање у оквиру базе правила и механизма фази закључивања. Део за прорачун параметара израчунава нове ПИД параметре K_p , T_i и T_d користећи множитеље добијене у претходном делу и тренутну вредност ПИД параметара, према следећим изразима

$$K_p = K_p + K_p K_{p_multp} \quad (6.4)$$

$$T_i = T_i + T_i T_{i_multp} \quad (6.5)$$

$$T_d = T_d + T_d T_{d_multp} \quad (6.6)$$

6.2 Скуп процеса задатих функцијама преноса

Скуп се састоји од 133 процеса чије су функције преноса дате у изразима 6.7–6.15. Скуп процеса је разматран у [6] и овде се користи практично непромењен. Скуп ће бити коришћен за добијање експертског знања као и за тестирање коначних решења. Скуп укључује процесе са великим временским константама, великим временским кашњењима као и процесе са интегратором.

$$G_1(s) = \frac{e^{-s}}{1 + sT}, T = 0,02; 0,05; 0,1; 0,2; 0,3; 0,5; 0,7; 1; 1,3; 1,5; 2; 4; 6; 8; 10; 20; 50; 100; 200; 500; 1000 \quad (6.7)$$

$$G_2(s) = \frac{e^{-s}}{(1 + sT)^2}, T = 0,01; 0,02; 0,05; 0,1; 0,2; 0,3; 0,5; 0,7; 1; 1,3; 1,5; 2; 6; 8; 10; 20; 50; 100; 200; 500 \quad (6.8)$$

$$G_3(s) = \frac{1}{(1 + s)(1 + sT)^2}, T = 0,005; 0,01; 0,02; 0,05; 0,1; 0,2; 0,5; 2; 5; 10 \quad (6.9)$$

$$G_4(s) = \frac{1}{(1 + s)^n}, n = 3; 4; 5; 6; 7; 8 \quad (6.10)$$

$$G_5(s) = \frac{1}{(1 + s)(1 + Ts)(1 + T^2s)(1 + T^3s)}, T = 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9 \quad (6.11)$$

$$G_6(s) = \frac{e^{-sL}}{s(1 + Ts)}, T = 0,01; 0,02; 0,05; 0,1; 0,3; 0,5; 0,7; 0,9; 1,0 T + L = 1 \quad (6.12)$$

$$G_7(s) = \frac{e^{-sL}}{(1 + sT)(1 + sT_1)}, L = 0,01; 0,02; 0,05; 0,1; 0,3; 0,5; 0,7; 0,9; 1,0 \quad (6.13)$$

$$T = 1; 2; 5; 10 \quad T_1 + L = 1$$

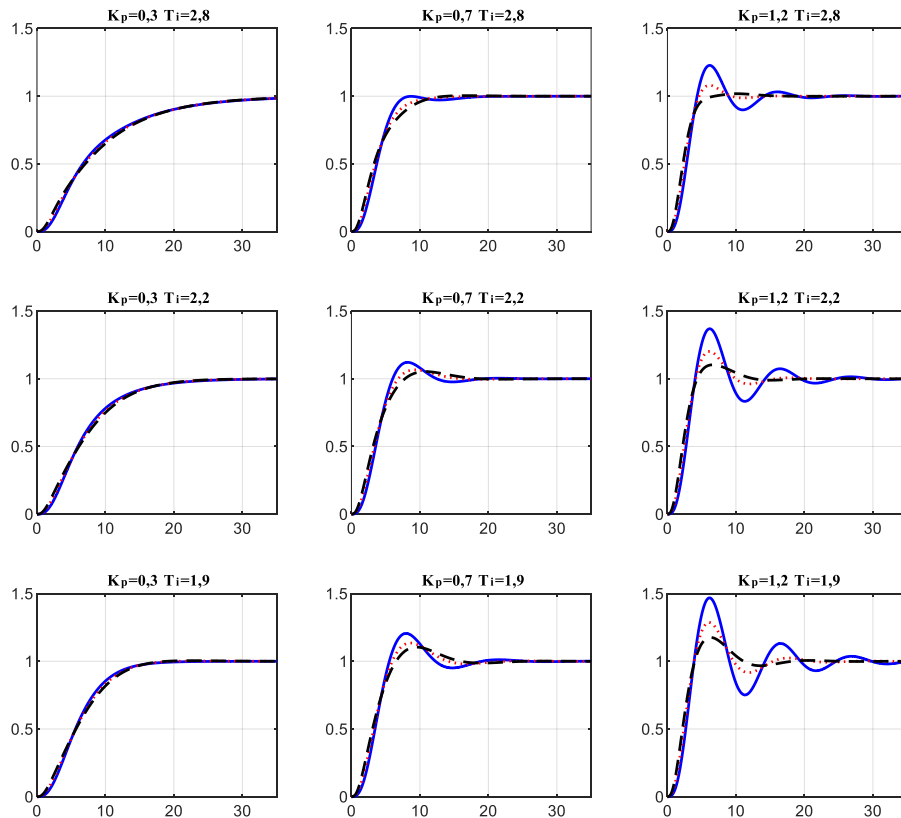
$$G_8(s) = \frac{1 + \alpha s}{(1 + s)^3}, \alpha = 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1,0; 1,1 \quad (6.14)$$

$$G_9(s) = \frac{1}{(1 + s)((sT)^2 + 1,4sT + 1)}, T = 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1,0 \quad (6.15)$$

6.3 Синтеза експертског знања

Посматрано је само неколико процеса из скупа процеса са функцијама преноса датих у изразима 6.7-6.15 како би се убрзала синтеза експертског знања без губљења општости. Сваки одабрани процес представља једну од четири уочене групе у скупу процеса. Процеси са интегратором су класификовани у прву групу коју представља процес G_6 описан изразом 6.12 са параметром $L=0,5$. Процес G_5 дат изразом 6.11 са параметром $\alpha=0,5$ представља другу групу процеса код којих је вредност временске константе доминантна у односу на вредност временског кашњења. Трећа група укључује процесе уравнотежених вредности временске константе и временског кашњења представљене процесом G_4 који је описан изразом 6.10 са параметром $n=4$. Процес G_2 описан изразом 6.8 са параметром $T=0,05$ представља последњу групу процеса код којих је вредност временског кашњења доминантна у односу на вредност временске константе.

Мапе подешавања, описане у поглављу 5.3, су креиране за сва четири процеса док је као пример, мапа подешавања за процес G_4 описан изразом 6.9 са параметром $n=4$ приказана на слици 6.2.



Слика 6.2 Мапа подешавања за процес G_4 (6.10) са параметром $n=4$ (плава пуна $T_d=0,2$; црвена тачкаста $T_d=0,7$; црна испрекидана $T_d=1,1$)

Слика 6.2 илуструје како на одскочни одзив утичу промене пропорционалног појачања K_p , време интегралења T_i и време диференцирања T_d . Параметар K_p расте дуж апсцисе а параметар T_i дуж ординате, док је одскочни одзив за сваку вредност параметра T_d представљен различитим бојама и типовима линија.

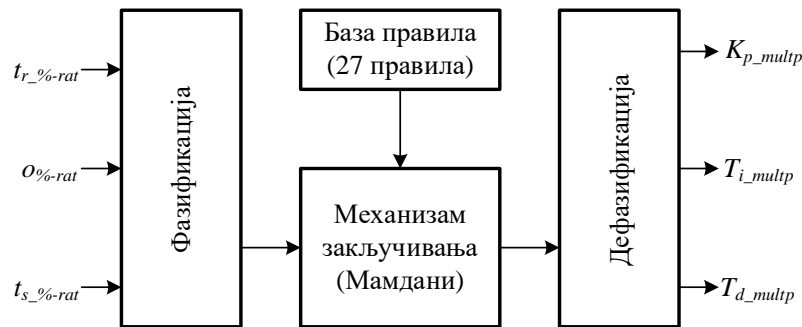
На основу креираних мапа подешавања могуће је извести релације (Табела 6.1) између растућих вредности параметара ПИД регулатора и вредности карактеристика одскочног одзива (време успона, прескок и време смирења).

Табела 6.1 Основни односи између повећања параметара ПИД регулатора и карактеристика одскочног одзива

Параметар	Време успона	Прескок	Време смирења
K_p	смањење	повећање	мало смањење
T_i	повећање	смањење	повећање/смањење
T_d	мало повећање	смањење	смањење

6.4 Фази систем закључивања (Мамдани)

За предложено методу је усвојен Мамдани фази систем закључивања са три улаза и три излаза. Блок дијаграм предложеног система је приказан на слици 6.3.



Слика 6.3 Блок дијаграм Мамдани фази система закључивања

Спецификација параметара усвојеног фази система је дата у табели 6.2.

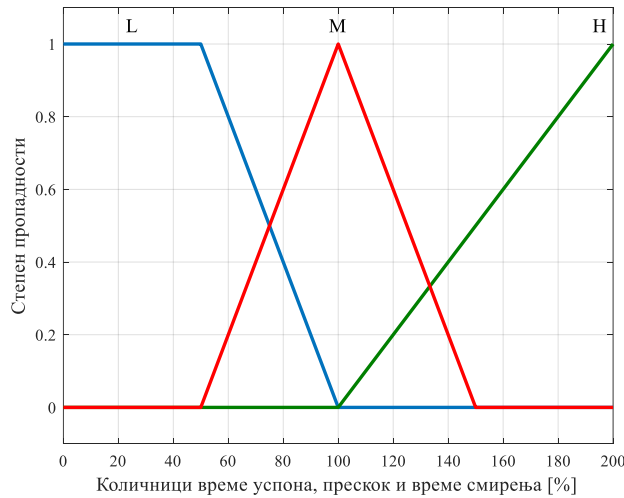
Табела 6.2 Спецификација Мамдани фази система закључивања

Параметар	Вредност
И метод	минимум
ИЛИ метод	максимум
Импликација	минимум
Агрегација	максимум
Дефазификација	центар масе
Број правила	27
Број улаза	3
Број излаза	3
Број функција преноса по улазу	3
Број функција преноса по излазу	5

6.4.1 Фазификација

Као улазни параметри су изабрани количници времена успона $t_r_%-rat$, прескока $O_%-rat$ и времена смирења $t_s_%-rat$ изражени у процентима. Вредност улазних параметара је подешена у распону од 0 % до 200 %. За процентуалне количнике веће од 200% вредности процењених перформанси су већ сигурно доста веће од референтних

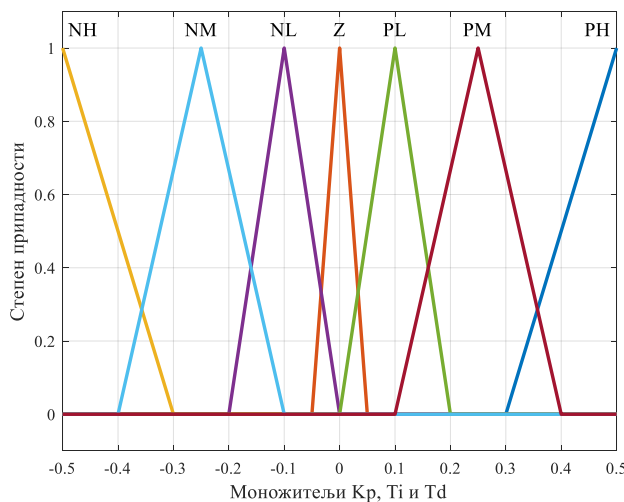
вредности па је горња граница подешена на ниво 200 %. Улазни параметри су декомпоновани на три различита фази лингвистичка нивоа: низак (L - low), средње (M - medium) и високо (H - high). Облик функција припадности за сва три улаза приказан је на слици 6.4.



Слика 6.4 Функције припадности за три улаза: количник времена успона, прескока и времена смирења

6.4.2 Дефазификација

Као излазни параметри изабрани су множитељи K_{p_multp} , T_{i_multp} и T_{d_multp} . Вредност излазних параметара је подешена да буде у опсегу од -0,5 до 0,5 тако да се ПИД параметри могу подесити до 50% њихове вредности на горе или на доле у свакој итерацији. Излазни параметри су декомпоновани на седам различитих фази језичких нивоа: негативан висок (NH - negative high), негативан средњи (NM - negative medium), негативан низак (NL - negative low), нула (Z - zero), позитиван низак (PL - positive low), позитиван средњи (PM - positive medium) и позитиван висок (PH - positive high). Облик функција припадности за сва три множитеља може се видети на слици 6.5.



Слика 6.5 Функције припадности за три излаза: K_p , T_i и T_d множитељ

6.4.3 База правила

За креирање базе правила коришћена је компаративна анализа релација добијених мапама подешавања четири репрезентативна процеса. Пошто је сваки улаз описан са три фази скупа, добро дефинисани фази систем закључивања треба да има $3 \times 3 \times 3 =$

27 фази правила. Свако правило дефинише како треба ПИД параметри да се мењају у зависности од вредности односа између тренутних и референтних мера перформанси. Преглед фази правила је приказан у табели 6.3.

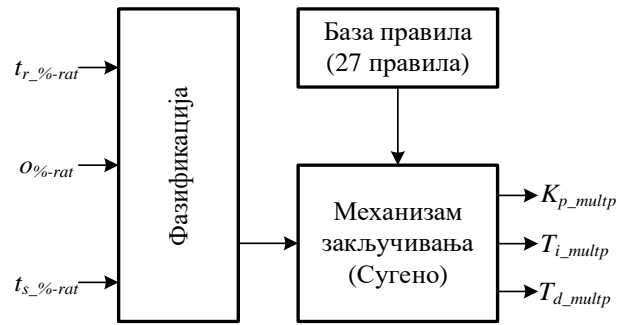
Табела 6.3 База правила фази система закључивања

Правило	$t_r\%$ -rat	$o\%$ -rat	$t_s\%$ -rat	K_p -multp	T_i -multp	T_d -multp
1	L	L	L	Z	Z	Z
2	L	L	M	Z	Z	Z
3	L	L	H	NL	NM	Z
4	L	M	L	NL	PM	PL
5	L	M	M	NL	PM	PL
6	L	M	H	NL	PM	NL
7	L	H	L	NL	PM	PL
8	L	H	M	NL	PM	PL
9	L	H	H	NL	NM	PL
10	M	L	L	PM	PL	NL
11	M	L	M	PM	PL	NL
12	M	L	H	PM	PL	NL
13	M	M	L	PM	PM	Z
14	M	M	M	PM	PM	Z
15	M	M	H	PM	PM	Z
16	M	H	L	NM	PH	PM
17	M	H	M	NM	PH	PM
18	M	H	H	NM	PH	PM
19	H	L	L	PH	NL	NL
20	H	L	M	PH	NL	NL
21	H	L	H	PH	NL	NL
22	H	M	L	PH	NM	NM
23	H	M	M	PH	NM	NM
24	H	M	H	PH	NM	NM
25	H	H	L	NH	NH	NH
26	H	H	M	NH	NH	NH
27	H	H	H	NH	PH	NH

6.5 Фази систем закључивања (Сугено)

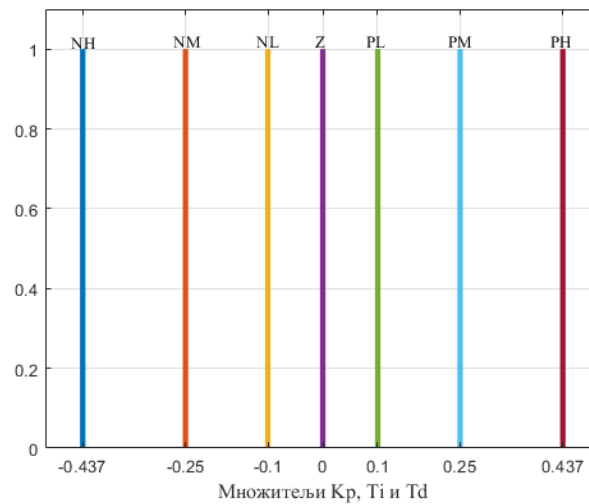
За имплементацију предложене методе на реалном контролеру је усвојен Сугено фази систем закључивања са три улаза и три излаза. Блок дијаграм предложеног система је приказан на слици 6.б.

Количници времена успона $t_r\%$ -rat, прескока $o\%$ -rat и времена смирења $t_s\%$ -rat изражени у процентима су изабрани као улазни параметри исто као и код претходно поменутог Мамдани фази система закључивања. Уопште, поступак фазификације је идентичан описаном у поглављу 6.4.1. База правила је такође идентична описаној у поглављу 6.4.3.



Слика 6.6 Блок дијаграм Сугено фази система закључивања

Као излазни параметри изабрани су множитељи K_p_multp , T_i_multp и T_d_multp . За формирање излаза се користе функција нултог реда (константа) за сваку од лингвистичких вредности фази излазних променљивих што се може видети на слици 6.7.



Слика 6.7 Функције припадности за три излаза: K_p , T_i и T_d множитељ

Спецификација параметара усвојеног фази система је дата у табели 6.4.

Табела 6.4 Спецификација Сугено фази система закључивања

Параметар	Вредност
И метод	минимум
ИЛИ метод	максимум
Импликација	минимум
Агрегација	максимум
Број правила	27
Број улаза	3
Број излаза	3
Број функција преноса по улазу	3
Број функција преноса по излазу	5

6.6 Режији рада

За промену својстава система у затвореној спреси пожељно је имати параметар којим је могуће потенцирати одређене индексе перформанси или робусности. Позиције функција припадности фази скупова L, M и H за сва три улаза имају значајан

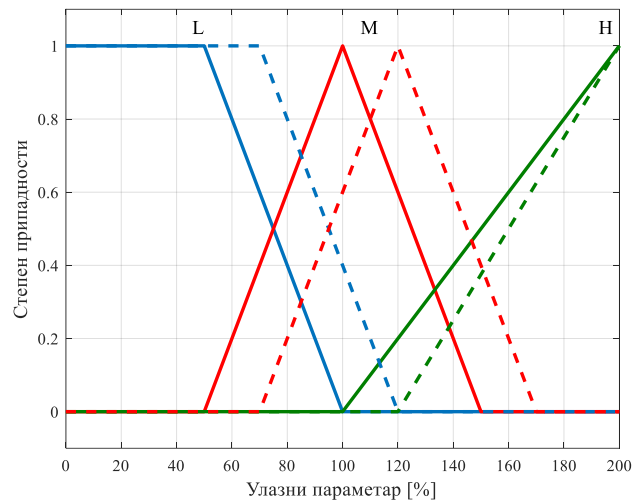
утицај на рад предложене методе, који се може користити за одабир критеријума за подешавање параметара ПИД регулатора. Троугласте и трапезоидне функције припадности могу се изразити скупом њихових темена. Положај функције припадности у зависности од параметра P може се описати следећим изразима

$$L = [0, 0, P, \max((50 + P), 100)] \quad (6.16)$$

$$M = [P, \max((50 + P), 100), \max((100 + P), 150)] \quad (6.17)$$

$$H = [\max((50 + P), 100), 200, 200] \quad (6.18)$$

На слици 6.8 приказан је положај функција припадности за две различите вредности параметра P . Из једначина 6.16-6.18 се може видети да када је $P = 0$, позиције функције припадности се сведе на основну верзију приказану на слици 6.4.



Слика 6.8 Позиције функција припадности за две различите вредности параметра P – пуне $P=50$, испрекидане $P=75$

Са правилним избором параметра P могу се добити различити режими рада предложеног алгорита. У оквиру ове студије предлажу се три начина рада:

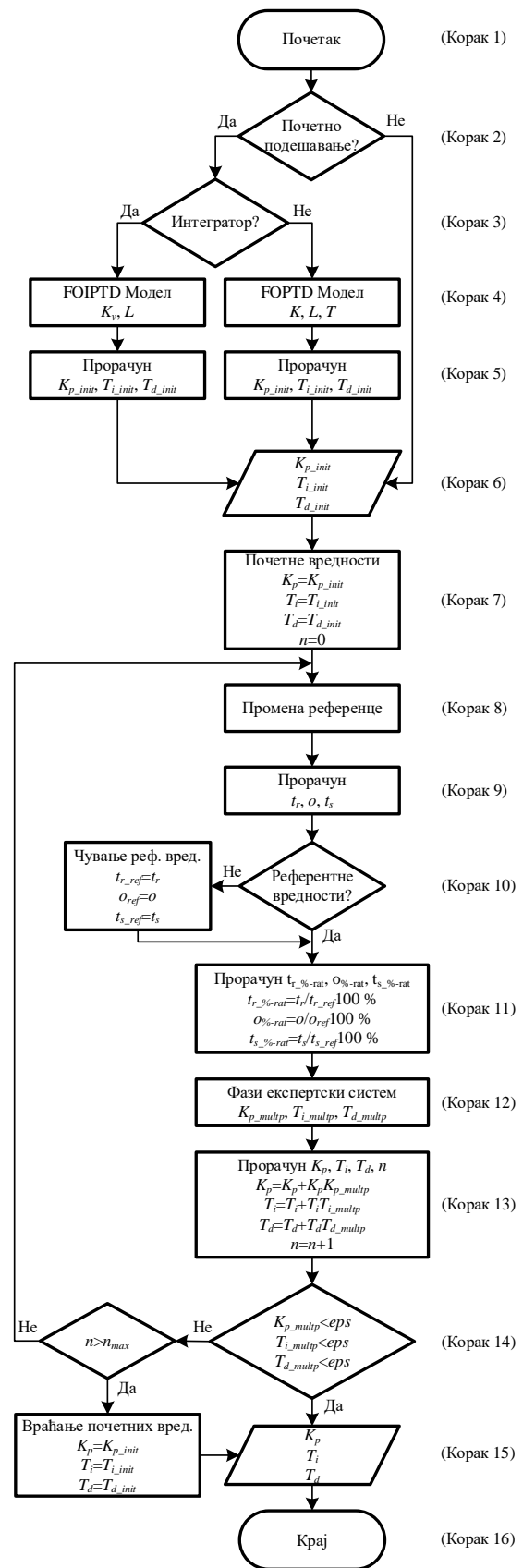
- Убрзање – режим рада чији је примарни циљ смањење времена успона система у затвореној спрези на првом месту а затим смањење времена смирења и прескока уз задржавање осетљивости у разумним границама. Усвојена вредност параметра P за први улаз $t_{r\%rat}$ је 40, за други улаз $o_{\%rat}$ је 100 а за трећи улаз $t_{s\%rat}$ је 100.
- Смањење прескока – режим рада чији је примарни циљ смањење прескока система у затвореној спрези на првом месту а затим смањење времена успона и времена смирења уз задржавање осетљивости у разумним границама. Усвојена вредност параметра P за први улаз $t_{r\%rat}$ је 100, за други улаз $o_{\%rat}$ је 40 а за трећи улаз $t_{s\%rat}$ је 100.
- Балансирање – режим рада чији је примарни циљ да уравнотежи смањење времена прескока и времена успона система у затвореној спрези на првом месту а затим да смањи време смирења уз задржавање осетљивости у разумним границама. Стога је вредност параметра P за сваки од три улаза ($t_{r\%rat}$, $o_{\%rat}$ и $t_{s\%rat}$) постављена на 75.

6.7 Опис поступка

Дијаграм тока предложене методе је приказан на слици 6.9. Ако нема почетних ПИД параметара или систем у затвореној спрези има лоше перформансе, нови почетни параметри се могу прорачунати у поступку почетног подешавања параметара. Алтернативно, почетно подешавање се може прескочити ако су перформансе у затвореној спрези задовољавајуће (корак 2).

У току поступка почетног подешавања, ПИД регулатор је искључен. На основу одскочног одзива у отвореној спрези, процес се апроксимира моделом описаним једначинама 5.6 или 5.7, у зависности од тога да ли систем поседује интегратор (кораци 3-4). Након успешног добијања почетних параметара (корак 6), ПИД регулатор се поново повезује и започиње се наредна фаза.

Друга фаза се изводи са почетним параметрима који се усвајају као тренутни (корак 7). Прва итерација укључује пертурбацију задате вредности (корак 8) и израчунавање перформанси одскочног одзива (корак 9). Израчунате перформансе се чувају као референтне вредности за ову и за наредне итерације (корак 10). Релативне вредности мера перформанси изражене као проценат сачуваних референтних вредности добијају се у кораку 11. Фази експертски систем генерише множитеље за ПИД параметре користећи израчунате релативне вредности перформанси као улаз у кораку 12. На основу израчунатих множитеља и тренутних вредности ПИД параметара, нови параметри се израчунавају у кораку 13. Ови кораци се понављају све док су множитељи већи од толеранције а број итерација не пређе максималан број (корак 14). Када множитељи постану мањи од толеранције, последњи ПИД параметри се усвајају као коначни (корак 15) и поступак се успешно завршава (корак 16). Ако је број итерација прекорачен, параметри се враћају на почетне вредности (корак 15) и процедура се неуспешно завршава (корак 16). Овај критеријум заустављања заснован на броју итерација је користан ако процедура подешавања не конвергира.



Слика 6.9 Дијаграм тока предложене методе

7 Симулациони резултати

У овом поглављу је дат опис имплементације предложене методе у софтверу за симулацију као и преглед и анализа резултата добијених симулацијом реализованих модела.

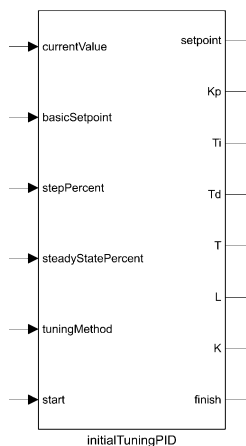
7.1 Имплементација предложене методе у симулационом софтверу

Предложена метода је имплементирана у симулационом софтверу у виду два одвојена дела почетног и аутоматског подешавања параметара. Софтвер за симулацију који је коришћен за имплементацију је MATLAB¹ и Simulink².

7.1.1 Имплементација почетног подешавања параметара

За идентификацију процеса, као предуслов за прорачун почетних параметара регулатора, усвојена је апроксимација моделом описана у поглављу 5.2.1.1. За прорачун параметара ПИД регулатора изабрана је Ц-Н метода описана у поглављу 5.2.2.1 и О-Х метода описана у поглављу 5.2.2.2.

Почетно подешавање је имплементирано као посебан Simulink блок са 6 улаза и 8 излаза приказан на слици 7.1. Преглед са описом улаза и излаза блока је дат у табели 7.1 док су детаљи реализације блока у виду изворног кода дати у прилогу I.1.



Слика 7.1 Реализовани Simulink блок *initTuningPID*

Поред Simulink блока почетно подешавање је реализовано и као посебна MATLAB функција чија је спецификација улаза и излаза дата у табели 7.2. Реализација је извршена коришћењем уграђених MATLAB функција (*step*, *stepinfo*) што убрзава поступак тестирања и анализу већег броја процеса. Детаљна реализација у виду изворног кода је дата у прилогу II.1.

¹ MATLAB је производ компаније MathWorks и сам назив је регистровани заштитни знак

² Simulink је производ компаније MathWorks и сам назив је регистровани заштитни знак

Табела 7.1 Спецификација улаза и излаза Simulink блока *initTuningPID*

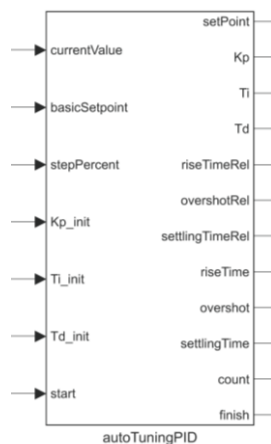
Улаз/излаз блока	Тип	Опис
currentValue	улаз	Тренутна вредност излаза процеса
basicSetpoint	улаз	Основна задата вредност
stepPercent	улаз	Амплитуда одсечно побуде у процентима задате вредности
steadyStatePercent	улаз	Граница за устаљено стање у процентима задате вредности
tuningMethod	улаз	Метода подешавања Ц-Н (1) или О-Х (2)
start	улаз	Сигнал за почетак
setpoint	излаз	Задата вредност за улаз процеса
K_p	излаз	Пропорционално појачање
T_i	излаз	Временска константа интегралне
T_d	излаз	Временска константа диференцирања
T	излаз	Процењена вредност временске константе
L	излаз	Процењена вредност временског кашњења
K	излаз	Процењена вредност статичког појачања
finish	излаз	Сигнал завршетка

Табела 7.2 Спецификација улаза и излаза MATLAB функције *initTuningPID*

Улаз/излаз блока	Тип	Опис
plant	улаз	Функција преноса процеса
tuningMethod	улаз	Метода подешавања Ц-Н (1) или О-Х (2)
K_p	излаз	Пропорционално појачање
T_i	излаз	Временска константа интегралне
T_d	излаз	Временска константа диференцирања
T	излаз	Процењена вредност временске константе
L	излаз	Процењена вредност временског кашњења
K	излаз	Процењена вредност статичког појачања

7.1.2 Имплементација аутоматског подешавања параметара

На основу описа предложене методе у поглављу 6.7 као и на основу описа режима рада датог у поглављу 6.5 реализован је посебан Simulink блок са 7 улаза и 12 излаза приказан на слици 7.2. Преглед улаза и излаза блока са описом је дат у табели 7.3 док је изворни код реализације дат у прилогу I.2.

Слика 7.2 Реализовани Simulink блок *autoTuningPID*

Поред Simulink блока аутоматско подешавање параметара је реализовано и као посебна MATLAB функција чија спецификација улаза и излаза је приказана у табели 7.4. Реализација је извршена коришћењем уграђених MATLAB функција (*step*, *stepinfo* и сл.) које се извршавају у итеративном поступку. Овакав начин реализације убрзава поступак тестирања што омогућава анализу већег броја процеса. Изворни код реализације се може видети у прилогу II.2.

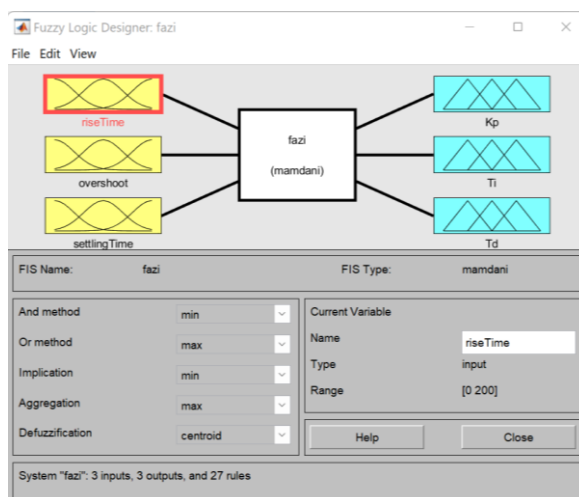
Табела 7.3 Преглед улаза и излаза Simulink блока autoTuningPID

Улаз/излаз блока	Тип	Опис
currentValue	улаз	Тренутна вредност излаза процеса
basicSetpoint	улаз	Основна задата вредност
stepPercent	улаз	Амплитуда одскочне побуде у процентима задате вредности
Kp_init	улаз	Почетна вредност пропорционалног појачања
Ti_init	улаз	Почетна вредност временске константе интегралења
Td_init	улаз	Почетна вредност временске константе диференцирања
start	улаз	Сигнал за почетак
setpoint	излаз	Задата вредност излаза процеса
Kp	излаз	Пропорционално појачање у тренутној итерацији
Ti	излаз	Временска константа интегралења у тренутној итерацији
Td	излаз	Временска константа диференцирања у тренутној итерацији
riseTimeRel	излаз	Процентуални количник времена успона у тренутној итерацији
overshootRel	излаз	Процентуални количник прескока у тренутној итерацији
settlingTimeRel	излаз	Процентуални количник времена смирења у тренутној итерацији
riseTime	излаз	Вредност времена успона у тренутној итерацији
overshoot	излаз	Вредност прескока у тренутној итерацији
settlingTime	излаз	Вредност времена смирења у тренутној итерацији
finish	излаз	Сигнал завршетка

Табела 7.4 Спецификација улаза и излаза MATLAB функције autoTuningPID

Улаз/излаз блока	Тип	Опис
plant	улаз	Функција преноса процеса
tuningMethod	улаз	Метода подешавања Ц-Н (1) или О-Х (2)
mode	улаз	Режим рада
Kp_init	улаз	Почетна вредност пропорционалног појачања
Ti_init	улаз	Почетна вредност временске константе интегралења
Td_init	улаз	Почетна вредност временске константе диференцирања
results	излаз	Коначни резултати
results_header	излаз	Назив колона
ctrl	излаз	Параметри регулатора по итерацијама
iterResults	излаз	Резултати по итерацијама

Фази систем закључивања описан у поглављу 6.4 који је језгро предложене методе је реализован коришћењем Fuzzy system designer алата који је саставни део MATLAB развојног окружења (Слика 7.3)



Слика 7.3 Алат за пројектовање фази система у оквиру симулационог софтвера

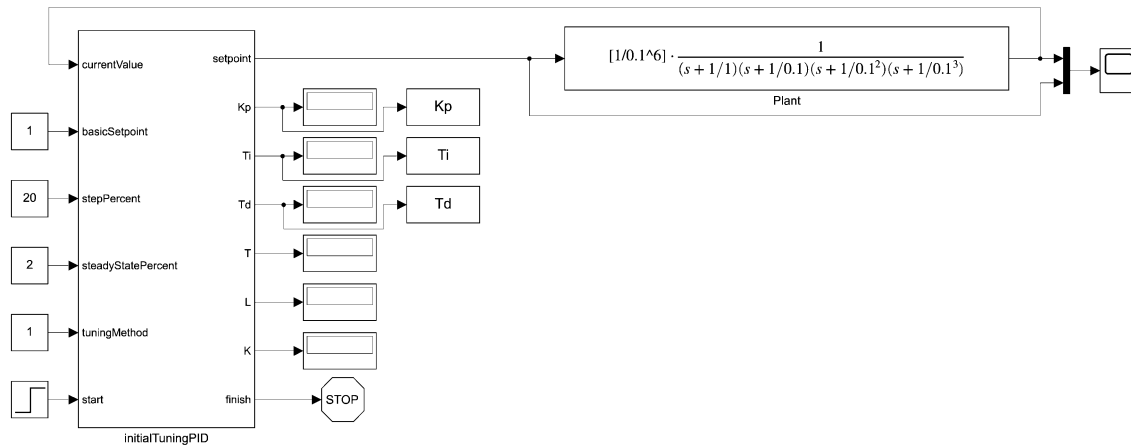
7.2 Симулациони модели

У наставку су дати описи симулационих модела насталих на основу реализованих блокова за почетно и аутоматско подешавање параметара. Поступак тестирања биће

описан кроз један пример почевши од почетног подешавања па до итеративног поступка аутоматског подешавања параметара.

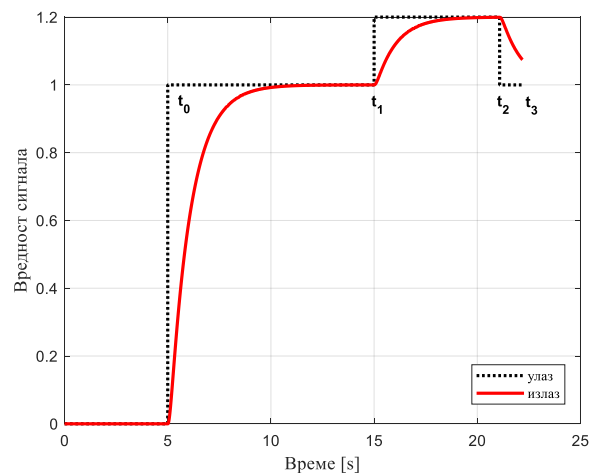
7.2.1 Симулациони модел почетног подешавања параметара

За потребе тестирања предложене методе реализован је симулациони модел који је приказан на слици 7.4. Модел се састоји од претходно описаног блока за почетно подешавање (*initialTuningPID*) и процеса за који се пројектује ПИД регулатор. У конкретном примеру приказаном на слици 7.4 је дата поставка за почетно подешавање параметара Ц-Н методом примењено на процес G_5 (6.11) са параметром $T=0,1$.



Слика 7.4 Симулациони модел за почетно подешавање. Пример за G_5 (6.11) са параметром $T=0,1$

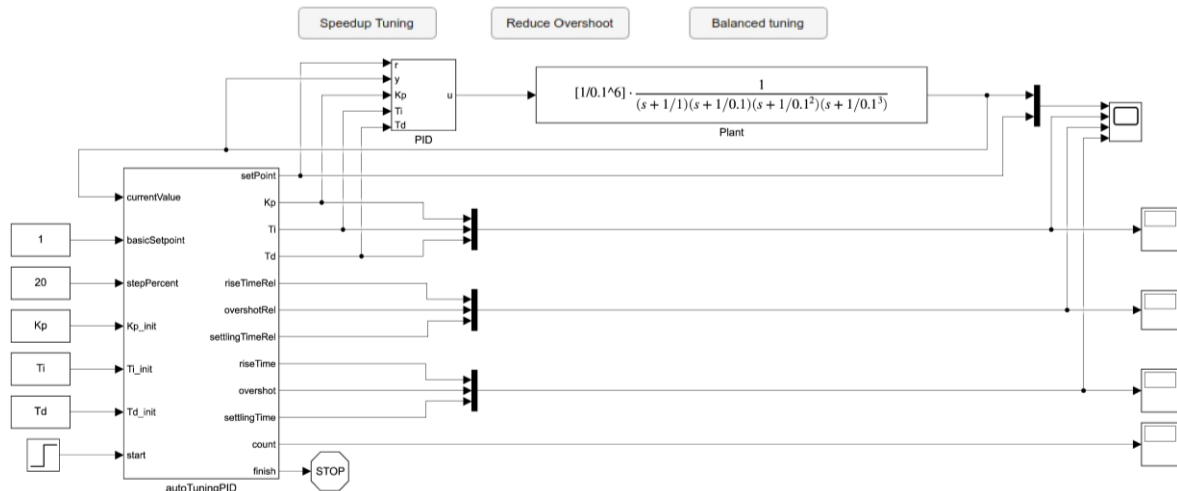
Резултати симулације за претходну поставку су приказани на слици 7.5. У тренутку t_0 на процес је доведен јединични одскочни сигнал ($basicSetpoint = 1$). Након што је процес ушао у устаљено стање ($steadyStatePercent = 2\%$) у тренутку t_1 започиње се идентификација процеса применом методе површина. На процес се у том тренутку доводи додатни јединични одскочни сигнал амплитуде 0,2 ($stepPercent = 20\%$) суперпониран са претходним. Након поновног уласка процеса у устаљено стање добија се вредност површине A_0 а на улаз процеса се опет доводи основна вредност сигнала. Поступак се завршава у тренутку t_3 када се и рачуна површина A_1 . У истом тренутку се на основу добијених површина врши прорачун процењених вредности временске константе T , временског кашњења L и статичког појачања K . На крају се на основу процењених вредности K , L и T применом Ц-Н методе ($tuningMethod = 1$) врши прорачун почетних параметара ПИД регулатора.



Слика 7.5 Пример идентификације система у отвореној спрези за G_5 (6.11) са параметром $T=0,1$

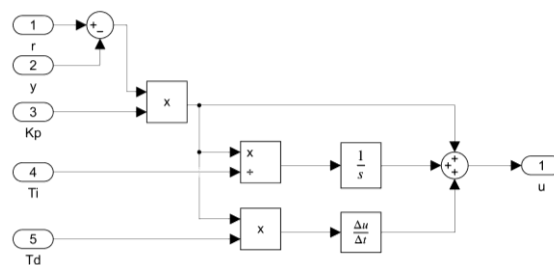
7.2.2 Симулациони модел аутоматског подешавања параметара

За потребе тестирања предложене методе реализован је симулациони модел који је приказан на слици 7.6. Модел се састоји од претходно описаног блока за аутоматско подешавање (autoTuningPID), ПИД регулатора, процеса за који се пројектује ПИД регулатор и дугмади за примену одговарајућег режима рада. Притиском на одговарајуће дугме извршава се програмски код (прилог I.3) који омогућује примену жељеног режима рада. У конкретном примеру је дата поставка за процес G_5 (6.11) са параметром $T=0,1$.



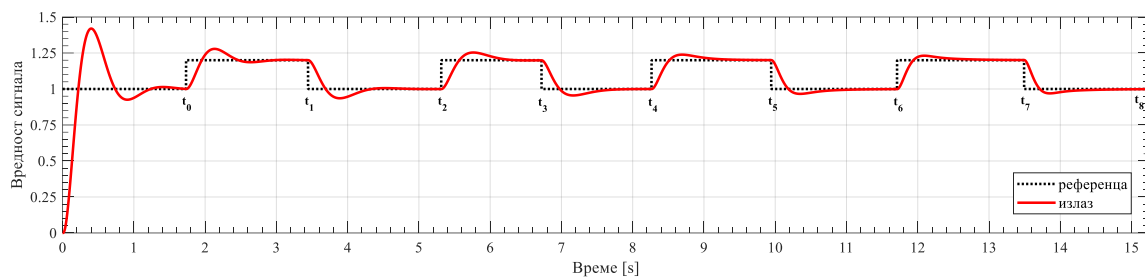
Слика 7.6 Симулациони модел за аутоматско подешавање за процес G_5 (6.11) са параметром $T=0,1$

У процесу тестирања коришћена је основна верзија ПИД регулатор дата изразом 3.5 чија је реализација приказана на слици 7.7.



Слика 7.7 Simulink блок са реализацијом ПИД регулатора

Резултати симулације за претходну поставку су приказани на слици 7.8. Са слике се може уочити да је на почетку на систем у затвореној спреси са регулатором са почетним параметрима доведен јединични одскочни сигнал референце ($basicSetpoint = 1$). Након уласка система у устаљено стање ($steadyStatePercent = 2\%$) у тренутку t_0 на улаз система се доводи додатни јединични одскочни сигнал амплитуде 0,2 ($stepPercent = 20\%$) суперпониран са претходним. Након поновног уласка система у устаљено стање у тренутку t_1 добијају се референтне вредности карактеристика које ће се користити у следећим итерацијама. Истовремено фази експертски систем прорачунава параметре ПИД и започиње се следећа итерација довођењем опет основне вредности сигнала референце на улаз система. Итерације (од t_1 до t_7) се понављају све док промене у прорачунатим вредностима параметара ПИД регулатора не буду мање од усвојене толеранције. У тренутку t_8 поступак се завршава јер су прорачунате вредности сва три параметара ПИД регулатора постале мање од толеранције. Кроз итерације се види опадање прескока што је и основни циљ изабраног режим рада Смањење прескока.



Слика 7.8 Пример тока аутоматског подешавања параметара ПИД регулатора за процес $G_5(6.11)$ са параметром $T=0,1$ коришћењем предложене методе у режиму рада Смањење прескока

7.3 Резултати и анализа

Коришћењем реализованих симулационих модела извршене су симулације над целим скупом процеса чије су функције преноса описане изразима 6.7-6.15, изузимајући четири процеса која су искоришћена за синтезу експертског знања. Примењена су оба начина почетног подешавања као и сва три режима рада предложене методе. Коришћен је опсежан скуп процеса за тестирање како би се потврдила моћ генерализације предложене методе.

У наставку је дат преглед добијених резултата груписаних по режиму рада предложене методе. Како би се могао остварити детаљнији увид у перформансе предложене методе у сваком од режима рада прво ће бити анализирани резултати кроз четири примера. У сваком од примера ће се детаљније анализирати процес који представља једну од следећих уочених група: процеси са доминантном вредношћу временске константе, процеси за доминантном вредношћу временског кашњења, процеси са равномерним вредностима временске константе и временског кашњења и процеси са интегратором.

Након детаљне анализе кроз четири примера биће дат преглед и анализа резултата добијених над целим скупом за сваки од режима рада предложене методе.

7.3.1 Предложена метода у режиму рада Убрзање

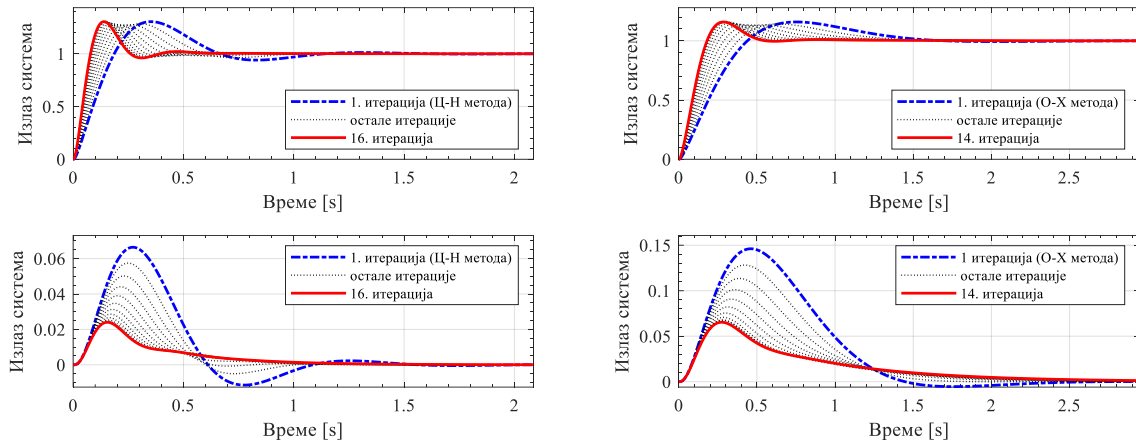
Резултати добијени применом предложене методе у режиму рада Убрзање, коришћењем обе методе за почетно подешавање параметара, су анализирани и испитана је испуњеност постављених циљева.

7.3.1.1 Пример: Процес са доминантном вредношћу временске константе

Процес $G_5(6.11)$ са параметром $T=0,1$ биће разматран у наставку као пример процеса са доминантном вредношћу временске константе.

У току почетног подешавања параметара извршена је идентификација процеса из одскочног одзива применом методе површина и добијени су параметри $T=1$ и $L=0,11$, односно $L/T=0,11$. Применом Ц-Н методе добијене су следеће вредности параметара ПИД регулатора: $K_p=11,11$, $T_i=0,22$ и $T_d=0,05$. Потом је примењено аутоматско подешавање у режиму рада Убрзање и након 16 итерација добијени су следећи параметри ПИД регулатора: $K_p=44,28$, $T_i=0,38$ и $T_d=0,03$. Одзиви система кроз итерације на јединичну одскочну побуду и поремећај приказани су на слици 7.9. Са одзива на одскочну побуду се може закључити да се време успона смањује кроз итерације. Прескок се веома мало мења кроз итерације док се време смирења благо побољшава из итерације у итерацију. Одзив на одскочни поремећај је приметно бољи у свакој итерацији.

Поступак почетног подешавања је поновљен коришћењем О-Х методе и добијене су следеће вредности параметара ПИД регулатора: $K_p=4,37$, $T_i=0,44$ и $T_d=0,05$. Након тога је поновљен и процес аутоматског подешавања у режиму рада Убрзање и после 14 итерација добијени су следећи параметри ПИД регулатора: $K_p=14,75$, $T_i=0,75$ и $T_d=0,03$. Одзиви система кроз итерације на јединичну одскочну побуду и поремећај су приказани на слици 7.9. Са слике се може закључити да се одзиви понашају на веома сличан начин као и код Ц-Н методе уз приметно повећање времена смирења код одскочног одзива на поремећај.



Слика 7.9 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес $G_5(6.11)$ са параметром $T=0,1$ применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)

Како би се омогућило боље поређење резултата израчунате су мере перформанси и робусности које су приказане у табели 7.5. Из табеле се може видети да предложена метода испуњава претходно постављене циљеве без обзира на вредност почетних параметара регулатора. Приметна је деградација осетљивости, која се очекује убрзањем система.

Табела 7.5 Перформансе и робусност ПИД регулатора за процес $G_5(6.11)$ са параметром $T=0,1$ постигнуте применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом и О-Х методом

	t_r [s]	σ [%]	t_s [s]	IAE	p	t_{sd} [s]	IAE _d	M_s
Ц-Н метода	0,14	30,44	1,04	0,20	0,07	1,40	0,03	1,31
ФЕС у режиму рада Убрзање	0,05	30,44	0,49	0,08	0,02	1,36	0,01	1,48
Унапређење [%]	-59,98	0,00	-53,04	-58,65	-63,83	-2,89	-67,53	12,66
О-Х метода	0,32	15,93	1,47	0,31	0,15	2,13	0,11	1,14
ФЕС у режиму рада Убрзање	0,13	15,93	0,49	0,13	0,07	2,90	0,05	1,30
Унапређење [%]	-59,96	0,00	-66,56	-57,85	-55,29	35,98	-52,61	14,08

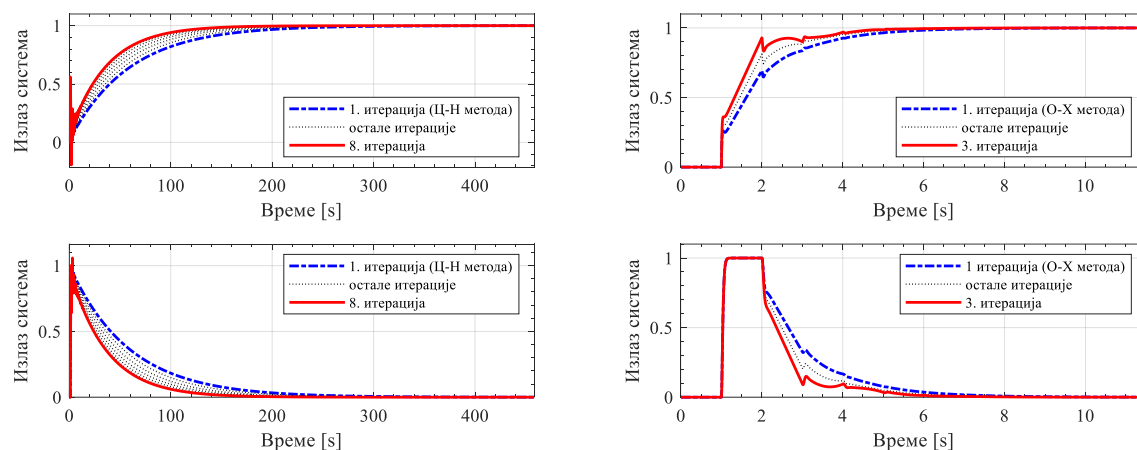
7.3.1.2 Пример: Процес са доминантном вредношћу временског кашњења

Као пример процеса са доминантном вредношћу временског кашњења у наставку ће бити разматран процес $G_2(6.8)$ са параметром $T=0,02$.

У току почетног подешавања параметара извршена је идентификација процеса из одскочног одзива применом методе површина и добијени су параметри $T=0,02$ и $L=1,01$, односно $L/T=50,5$. Применом Ц-Н методе добијене су следеће вредности параметара ПИД регулатора: $K_p=0,03$, $T_i=2,02$ и $T_d=0,51$. Затим је примењена предложена метода у режиму рада Убрзање и након 8 итерација су добијени следећи параметри ПИД регулатора: $K_p=0,08$, $T_i=2,78$ и $T_d=0,36$. На слици 7.10 се могу видети одзиви

система кроз итерације на јединичну одскочну побуду и поремећај. Посматрањем одзива на одскочну побуду се може закључити да се време успона и време смирења смањују кроз итерације док прескок остаје непромењен односно нула. Кроз итерације се види побољшање одзива на поремећај.

Поступак почетног подешавања је поновљен коришћењем О-Х методе и добијене су следеће вредности параметара ПИД регулатора: $K_p=0,21$, $T_i=0,43$ и $T_d=0,04$. Поновљено је и аутоматско подешавање у режиму рада Убрзање и након 3 итерације добијени су следећи параметри ПИД регулатора: $K_p=0,32$, $T_i=0,51$ и $T_d=0,04$. Одзиви система на јединичну одскочну побуду и поремећају приказани су на слици 7.10. Посматрајући одзиве на одскочну побуду и поремећај увиђа се слично понашање као и код Ц-Н методе с тим да се временска скала разликује за ред величине.



Слика 7.10 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_2 (6.8) са параметром $T=0,02$ применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)

Израчунате мере перформанси и робусности су приказане у табели 7.6 како би се омогућило боље поређење резултата. Из табеле се може видети да предложена метода испуњава претходно постављене циљеве и за једну и за другу вредност почетних параметара, али уз деградацију осетљивости.

Табела 7.6 Перформансе и робусност ПИД регулатора за процес G_2 (6.8) са параметром $T=0,02$ постигнуте применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом и О-Х методом

	t_r [s]	σ [%]	t_s [s]	IAE	p	t_{sd} [s]	IAE _d	M_s
Ц-Н метода	133,42	0,00	229,09	57,83	1,01	229,71	57,83	1,55
ФЕС у режиму рада Убрзање	80,01	0,00	138,93	34,21	1,06	137,93	34,21	2,08
Унапређење [%]	-40,04	0,00	-39,36	-40,84	5,08	-39,96	-40,84	33,62
О-Х метода	2,63	0,00	5,75	2,00	1,00	6,79	2,00	1,43
ФЕС у режиму рада Убрзање	0,95	0,00	4,50	1,58	1,00	5,54	1,58	1,66
Унапређење [%]	-63,77	0,00	-21,86	-21,15	0,00	-18,51	-21,15	16,30

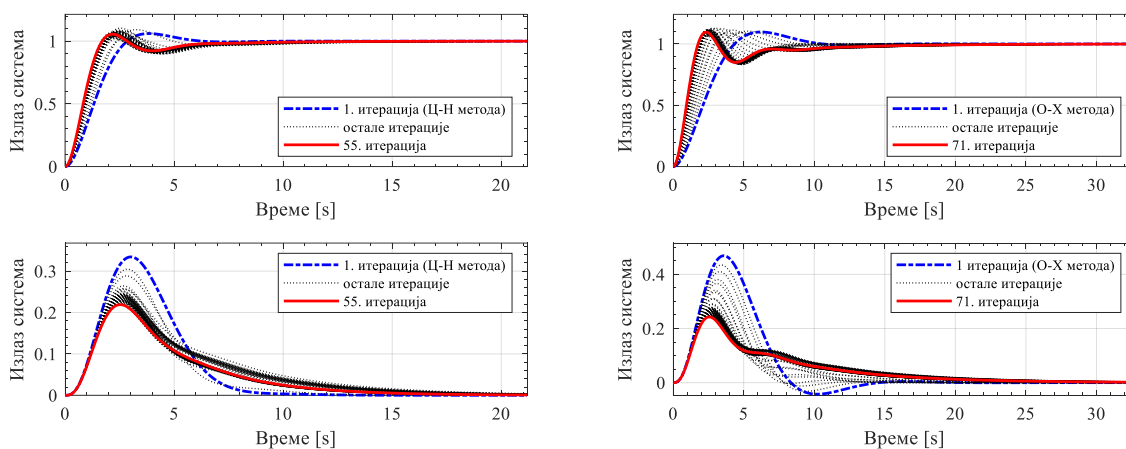
7.3.1.3 Пример: Процес са равномерним вредностима временске константе и временског кашњења

Процес G_4 (6.10) са параметром $n=3$ ће у наставку бити анализиран као пример процеса са равномерним вредностима временске константе и временског кашњења.

У току почетног подешавања параметара извршена је идентификација процеса из одскочног одзива применом методе површина и добијени су параметри $T=1,82$ и

$L=1,17$, односно $L/T=0,64$. Након примене Ц-Н методе добијене су следеће вредности параметара ПИД регулатора: $K_p=1,86$, $T_i=2,35$ и $T_d=0,59$. Аутоматско подешавање параметара у режиму рада Убрзање је након 55 итерација дало следеће параметре ПИД регулатора: $K_p=3,53$, $T_i=3,72$ и $T_d=0,71$. Одзиви система кроз итерације на јединичну одскочну побуду и поремећај се могу видети на слици 7.11. Примећује се да се време успона смањује а прескок веома мало мења кроз итерације док се време смирења благо повећава из итерације у итерацију.

Поступак почетног подешавања је поновљен коришћењем О-Х методе и добијене су следеће вредности параметара ПИД регулатора: $K_p=0,9$, $T_i=1,67$ и $T_d=0,49$. Аутоматско подешавање у режиму рада Убрзање је поновљено и након 71 итерације добијени су следећи параметри ПИД регулатора: $K_p=3,68$, $T_i=5,83$ и $T_d=0,46$. Одзиви система кроз итерације на јединичну одскочну побуду и поремећај приказани су на слици 7.11. Посматрањем одзива може се приметити да се одзиви понашају на веома сличан начин као код претходне методе почетног подешавања.



Слика 7.11 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_4 (6.10) са параметром $n=3$ применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)

Прорачунате мере перформанси и робусности приказане су у табели 7.7 како би се омогућило боље поређење резултата. Посматрањем добијених резултата може се уочити да предложена метода испуњава претходно постављене циљеве, али уз повећање времена смирења и код одзива на побуду и на поремећај. Такође се може приметити и деградација осетљивости која се очекује убрзањем система.

Табела 7.7 Перформансе и робусност ПИД регулатора за процес G_4 (6.10) са параметром $n=3$ постигнуте применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом и О-Х методом

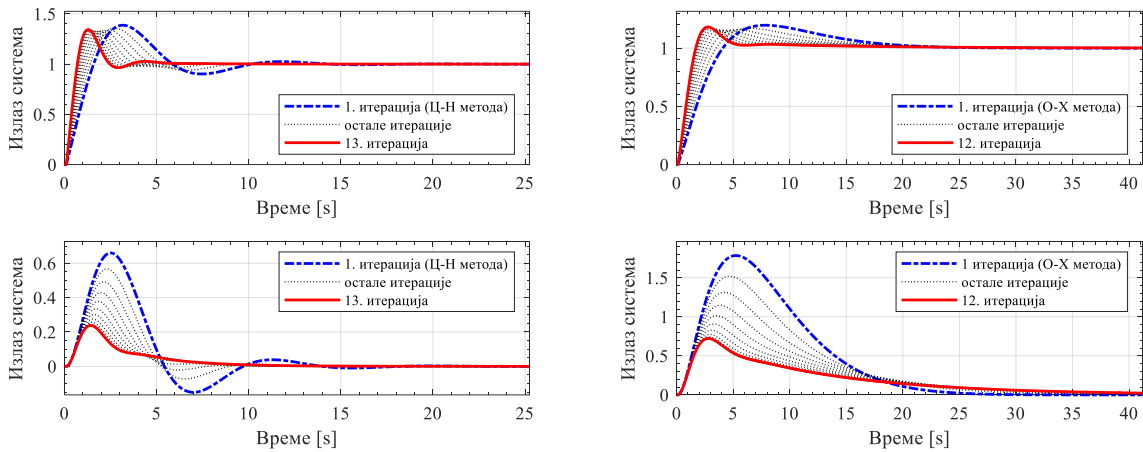
	t_r [s]	σ [%]	t_s [s]	IAE	p	t_{sd} [s]	IAE _d	M_s
Ц-Н метода	1,87	6,06	5,37	1,50	0,33	8,70	1,26	1,32
ФЕС у режиму рада Убрзање	1,12	5,57	6,44	1,14	0,22	15,64	1,05	1,41
Унапређење [%]	-40,40	-8,16	20,02	-24,09	-34,48	79,82	-16,45	7,18
О-Х метода	2,93	9,74	9,45	2,55	0,47	13,40	2,16	1,34
ФЕС у режиму рада Убрзање	1,17	9,62	13,91	1,75	0,24	26,06	1,59	1,58
Унапређење [%]	-59,98	-1,15	47,16	-31,31	-48,21	94,57	-26,62	17,85

7.3.1.4 Пример: Процес са интегратором

Као пример процеса са интегратором размотриће се процес G_6 (6.12) са параметром $T=0,9$.

У току почетног подешавања параметара извршена је идентификација процеса из одскочног одзива применом методе тангенте и добијени су параметри $L=1$ и $K_v=1$. Применом методе Ц-Н добијени су следећи параметри ПИД регулатора: $K_p=1,2$, $T_i=2$ и $T_d=0,5$. Потом је примењено аутоматско подешавање у режиму рада Убрзање и после 13 итерација добијени су следећи параметри ПИД регулатора: $K_p=4,58$, $T_i=3,32$ и $T_d=0,29$. На слици 7.12 су приказани одзиви система кроз итерације на јединичну одскочну побуду и поремећај. Уочава се смањење времена успона и времена смирења кроз итерације док прескок има благу тенденцију смањивања. Може се приметити побољшање одзива на одскочни поремећај кроз итерације.

Почетно подешавање је поновљено коришћењем О-Х методе и добијене су следеће вредности параметара ПИД регулатора: $K_p=0,45$, $T_i=8$ и $T_d=0,5$. Затим је примењено аутоматско подешавање у режиму рада Убрзање које је након 12 итерација дало следеће параметре ПИД регулатора: $K_p=1,45$, $T_i=12,46$, и $T_d=0,31$. Одзиви система на јединичну одскочну побуду и поремећај кроз итерације се могу видети на слици 7.12. Посматрајући одзиве долази се до закључка да се одзиви на одскочну побуду понашају на веома сличан начин као код претходне методе почетног подешавања док се код одзива на одскочни поремећај уочава повећање времена смирења.



Слика 7.12 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес $G_6(6.12)$ са параметром $T=0,9$ применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)

У табели 7.8 су приказане израчунате мере перформанси и робусности из којих се може закључити да предложена метода испуњава претходно постављене циљеве без обзира на вредност почетних параметара регулатора, али уз деградацију осетљивости, која се очекује убрзањем система. Приметно је повећање времена смирења код одзива на поремећај када су почетни параметри прорачунати применом О-Х методе.

Табела 7.8 Перформансе и робусност ПИД регулатора за процес $G_6(6.12)$ са параметром $T=0,9$ постигнуте применом предложене методе у режиму рада Убрзање након почетног подешавања Ц-Н методом и О-Х методом

	t_r [s]	σ [%]	t_s [s]	IAE	p	t_{sd} [s]	IAE _d	M_s
Ц-Н метода	1,16	38,70	12,62	2,15	0,66	13,17	2,51	1,45
ФЕС у режиму рада Убрзање	0,46	34,43	4,93	0,83	0,24	11,69	0,73	1,53
Унапређење [%]	-59,97	-11,05	-60,97	-61,56	-63,93	-11,21	-71,02	5,46
О-Х метода	2,98	19,67	20,59	3,70	1,79	23,58	17,79	1,12
ФЕС у режиму рада Убрзање	1,19	18,18	13,54	1,69	0,72	46,17	8,59	1,30
Унапређење [%]	-59,98	-7,53	-34,26	-54,20	-59,49	95,75	-51,72	16,42

7.3.1.5 Анализа над целим скупом процеса

Како би се убрзао процес симулација за свих 129 процеса из скупа, креирана је посебна програмска скрипта (прилог П.3) коришћењем реализованих функција за почетно и аутоматско подешавање параметара.

Поступак почетног подешавања је извршен за све процесе из скупа коришћењем Ц-Н методе. Потом је извршен и поступак аутоматског подешавања параметара применом предложене методе у режиму рада Убрзање. На основу добијених резултата израчуната су просечна побољшања перформанси и робусности, применом предложене методе, која су у табели 7.9 груписана по типу процеса.

Табела 7.9 Просечно побољшање вредности перформанси и робусности за све процесе постигнуто предложеном методом у режиму рада Убрзање након почетног подешавања Ц-Н методом

Процес	Број	t_r [%]	o [%]	t_s [%]	IAE [%]	p [%]	t_{sd} [%]	IAE _d [%]	M_s [%]	Итер.
G ₁ (6.7)	21	-44,56	4,46	198,29	225,12	1,95	250,61	494,18	156,77	36,29
G ₂ (6.8)	20	-64,78	-40,17	14,43	-23,99	-24,99	53,89	-19,22	59,10	22,05
G ₃ (6.9)	10	-56,10	19,59	80,15	12,78	-56,85	337,39	113,72	30,37	32,10
G ₄ (6.10)	5	-71,37	-70,73	75,85	26,66	-12,34	78,57	61,80	27,43	11,80
G ₅ (6.11)	8	-46,26	25,44	138,61	56,86	-36,89	191,68	145,91	26,75	41,75
G ₆ (6.12)	8	-52,35	9,22	130,20	255,57	-37,37	215,12	291,29	128,07	33,00
G ₇ (6.13)	36	-20,10	16,27	114,22	124,39	-33,11	177,39	327,79	76,85	32,53
G ₈ (6.14)	11	-69,57	-100,00	35,80	-2,90	-11,48	29,25	-2,14	25,99	16,55
G ₉ (6.15)	10	-43,52	68,84	224,00	97,06	-38,49	233,42	164,51	78,90	49,00
Укупно	129	-45,44	-3,23	112,63	96,32	-26,25	173,67	219,84	77,48	31,19

Предложена метода у режиму рада Убрзање постигла је значајно просечно смањење времена успона од преко 45 %, што је и примарни циљ овог режима рада. Истовремено, видљиво је мало просечно побољшање вредности прескока док је приметна деградација просечног времена смирења. Просечне вредности интеграла апсолутне грешке одскочног одзива на побуду и на поремећај су такође у просеку приметно повећане. Очекивано је да се са убрзањем долази до деградације осетљивости система и у просеку износи преко 77 %. Узимајући у обзир све процесе, просечно трајање поступка је око 31 итерације.

Поступак је поновљен за све процесе коришћењем О-Х методе за почетно подешавање и предложене методе у режиму рада Убрзање. Израчуната су просечна побољшања перформанси и робусности у односу на почетно подешавање, која се могу видети у табели 7.10 груписана по типу процеса.

Табела 7.10 Просечно побољшање вредности перформанси и робусности за све процесе постигнуто предложеном методом у режиму рада Убрзање након почетног подешавања О-Х методом

Процес	Број	t_r [%]	o [%]	t_s [%]	IAE [%]	p [%]	t_{sd} [%]	IAE _d [%]	M_s [%]	Итер.
G ₁ (6.7)	21	-63,13	-50,79	-21,68	-28,97	-14,29	0,73	-27,38	22,60	4,24
G ₂ (6.8)	20	-61,07	-38,00	-8,03	-32,88	-27,45	24,64	-31,39	22,56	11,50
G ₃ (6.9)	10	-59,63	-3,58	-9,94	-44,76	-49,93	102,48	-30,92	20,96	23,40
G ₄ (6.10)	5	-57,37	1,47	67,60	-10,47	-26,45	66,92	-10,48	46,00	49,00
G ₅ (6.11)	8	-58,40	-0,28	26,68	-29,98	-40,48	71,37	-22,46	27,35	37,38
G ₆ (6.12)	8	-60,15	-14,21	-21,99	-42,47	-45,56	73,21	-40,64	21,79	8,63
G ₇ (6.13)	36	-60,35	-14,13	-39,98	-44,57	-36,39	8,28	-42,12	18,95	10,78
G ₈ (6.14)	11	-60,82	-41,69	16,91	-23,49	-19,62	33,54	-20,18	37,30	19,36
G ₉ (6.15)	10	-52,51	-1,66	138,22	36,01	-39,38	164,73	133,53	32,14	46,40
Укупно	129	-60,04	-22,90	-1,63	-29,83	-31,69	41,38	-19,16	24,59	17,29

Предложена метода у режиму рада Убрзање постигла је значајно просечно смањење времена успона од преко 60 %, што је и примарни циљ овог режима рада. Истовремено, видљиво је просечно побољшање вредности осталих мера перформанси осим времена смирења код одзива на одскочни поремећај. Очекивано је да са убрзањем долази до деградације осетљивости система и у просеку износи преко 25 %, што је задовољавајуће. Узимајући у обзир све процесе, просечно трајање поступка је око 17 итерација.

7.3.2 Предложена метода у режиму рада Смањење прескока

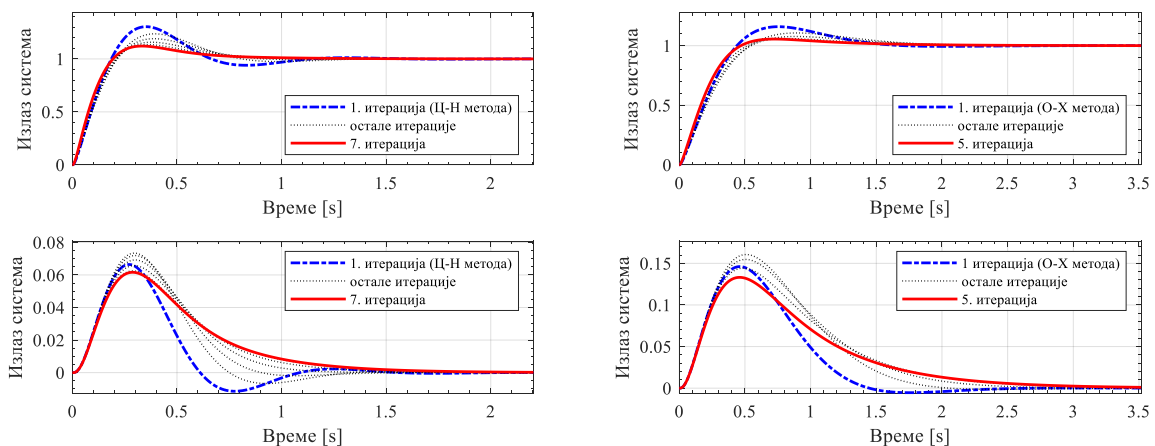
Анализирани су резултати добијени применом предложене методе у режиму рада Смањење прескока, коришћењем обе методе за почетно подешавање параметара и испитана је испуњеност постављених циљева.

7.3.2.1 Пример: Процес са доминантном вредношћу временске константе

У овом режиму рада као пример ће бити размотрен процес G_5 (6.11) са параметром $T=0,1$ исто као и у анализи претходног режима рада.

Након почетног подешавања параметара коришћењем Ц-Н методе добијене су следеће вредности параметара ПИД регулатора: $K_p=11,11$, $T_i=0,22$ и $T_d=0,05$. Применом аутоматског подешавања параметара у режиму рада Смањење прескока након 7 итерација добијени су следећи параметри ПИД регулатора: $K_p=13,32$, $T_i=0,46$ и $T_d=0,06$. Одзиви система на јединичну одскочну побуду и поремећај кроз итерације су приказани на слици 7.13. Код одскочног одзива се може приметити да се прескок и време смирења смањују кроз итерације док време успона остаје скоро непромењено. Код одзива на одскочни поремећај види се побољшање кроз итерације.

Почетно подешавање је поновљено коришћењем О-Х методе и добијене су следеће вредности параметара ПИД регулатора: $K_p=4,37$, $T_i=0,44$ и $T_d=0,05$. Затим је примењено аутоматско подешавање у режиму рада Смањење прескока и након 5 итерација добијени су следећи параметри ПИД регулатора: $K_p=5,46$, $T_i=0,74$ и $T_d=0,05$. На слици 7.13 се могу видети одзиви система на јединичну одскочну побуду и поремећај кроз итерације. Посматрајући одзиве може се приметити веома слично понашање као и код Ц-Н методе.



Слика 7.13 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_5 (6.11) са параметром $T=0,1$ постигнуте применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)

Упоредо су израчунате и мере перформанси и робусности због лакшег поређења резултата које су приказане у табели 7.11. Из табеле се може видети да предложена

метода испуњава претходно постављене циљеве уз побољшање осетљивости без обзира на вредност почетних параметара регулатора. Поред испуњених постављених услова приметне су слабије перформансе одзива на одскочни поремећај.

Табела 7.11 Перформансе и робустност ПИД регулатора за процес G_5 (6.11) са параметром $T=0,1$ предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом и О-Х методом

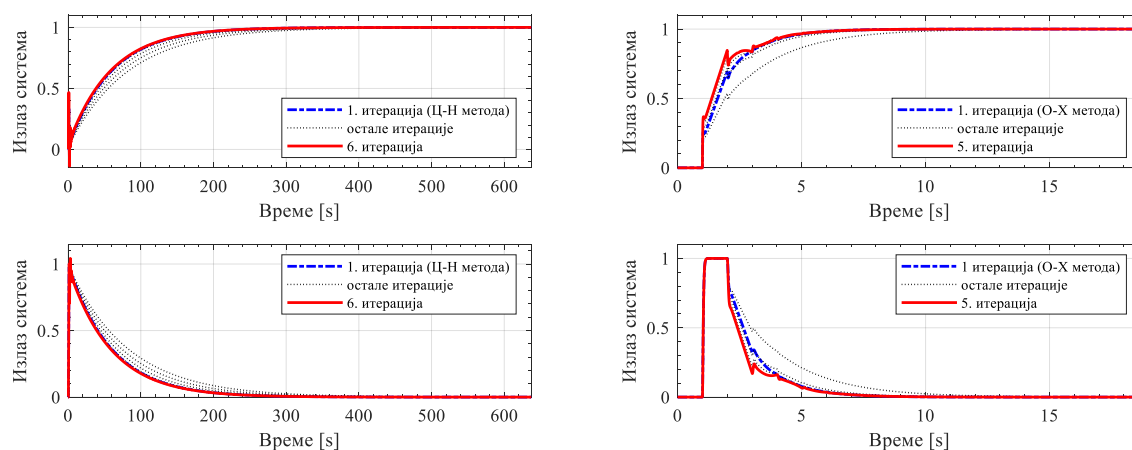
	t_r [s]	σ [%]	t_s [s]	IAE	p	t_{sd} [s]	IAE _d	M_s
Ц-Н метода	0,14	30,44	1,04	0,20	0,07	1,40	0,03	1,31
ФЕС у режиму рада Смањење прескока	0,14	12,13	0,82	0,13	0,06	1,63	0,03	1,11
Унапређење [%]	-0,49	-60,16	-21,76	-33,94	-7,22	15,82	31,77	-15,58
О-Х метода	0,32	15,93	1,47	0,31	0,15	2,13	0,11	1,14
ФЕС у режиму рада Смањење прескока	0,31	5,55	1,40	0,23	0,13	2,96	0,14	1,10
Унапређење [%]	-1,35	-65,18	-4,97	-25,05	-8,87	38,73	26,20	-4,01

7.3.2.2 Пример: Процес са доминантном вредношћу временског кашњења

У овом примеру ће бити размотрен процес G_2 (6.8) са параметром $T=0,02$ исто као и у анализи претходног режима рада.

Применом почетног подешавања параметара коришћењем Ц-Н методе добијене су следеће вредности параметара ПИД регулатора: $K_p=0,03$, $T_i=2,02$ и $T_d=0,51$. Након тога је примењено аутоматско подешавање параметара које је дало следеће параметре ПИД регулатора: $K_p=0,06$, $T_i=3,14$ и $T_d=0,44$. На слици 7.14 су представљени одзиви система на јединичну одскочну побуду и поремећају кроз итерације. Са слике се може закључити да се први и последњи одзив јако мало разликују. Ово је очекивано понашање у режиму Смањење прескока пошто почетни параметри дају одзив без прескока.

Поступак почетног подешавања је поновљен применом О-Х методе и добијене су следеће вредности параметара ПИД регулатора: $K_p=0,21$, $T_i=0,43$ и $T_d=0,04$. Предложена метода у режиму рада Смањење прескока је примењена и након 5 итерација добијени су следећи параметри ПИД регулатора: $K_p=0,32$, $T_i=0,57$ и $T_d=0,04$. Одзиви система кроз итерације на јединичну одскочну побуду и поремећај приказани су на слици 7.14. Примећује се понашање одзива на сличним основама као и код Ц-Н методе само се временске линије разликују за ред величина.



Слика 7.14 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_2 (6.8) са параметром $T=0,02$ применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)

Мере перформанси и робустности су израчунате и приказане у табели 7.12 како би се омогућило боље поређење добијених резултата. Посматрањем добијених резултата може се уочити да предложена метода испуњава претходно постављене циљеве без обзира на коришћену методу за почетно подешавање. У случају када почетно подешавање даје одзив без прескока, битно је напоменути да у овом режиму рада није дошло до погоршања перформанси, а нарочито прескока.

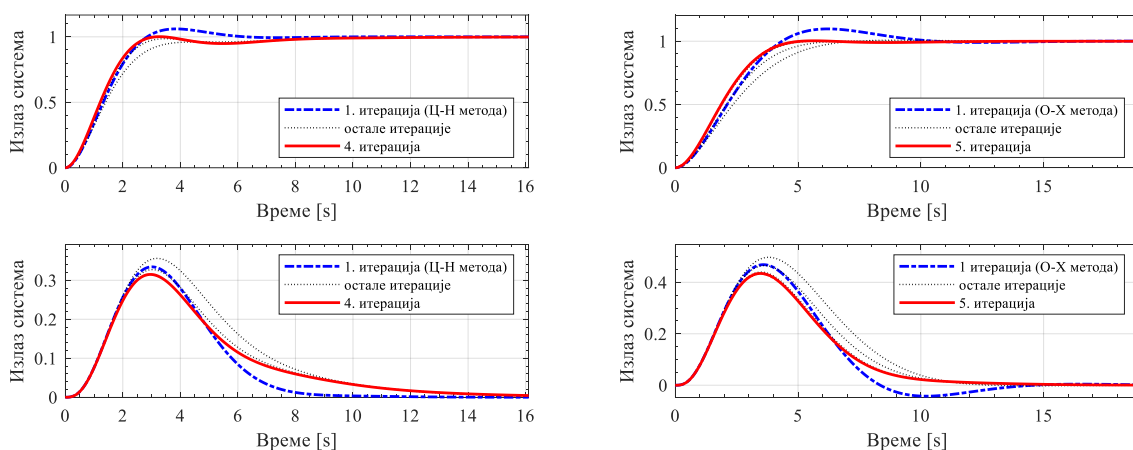
Табела 7.12 Перформансе и робустност ПИД регулатора за процес G_2 (6.8) са параметром $T=0,02$ постигнуте применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом и О-Х методом

	t_r [s]	σ [%]	t_s [s]	IAE	p	t_{sd} [s]	IAE _d	M_s
Ц-Н метода	133,42	0,00	229,09	57,83	1,01	229,71	57,83	1,55
ФЕС у режиму рада Смањење прескока	131,55	0,00	226,73	56,39	1,04	225,35	56,39	2,00
Унапређење [%]	-1,40	0,00	-1,03	-2,48	3,50	-1,90	-2,48	28,58
О-Х метода	2,63	0,00	5,75	2,00	1,00	6,79	2,00	1,43
ФЕС у режиму рада Смањење прескока	2,59	0,00	5,77	1,81	1,00	6,81	1,81	1,59
Унапређење [%]	-1,53	0,00	0,23	-9,78	0,01	0,18	-9,78	10,87

7.3.2.3 Пример: Процес са равномерним вредностима временске константе и временског кашњења

У овом примеру ће бити размотрен процес G_4 (6.10) са параметром $n=3$ исто као и у анализи претходног режима рада.

Након почетног подешавања параметара коришћењем Ц-Н методе добијене су следеће вредности параметара ПИД регулатора: $K_p=1,86$, $T_i=2,35$, и $T_d=0,59$. Након тога је примењено аутоматско подешавање параметара у режиму рада Смањење прескока и након 4 итерације су добијени следећи параметри ПИД регулатора: $K_p=2,16$, $T_i=3,23$ и $T_d=0,58$. Одзиви система кроз итерације на јединичну одскочну побуду и поремећају приказани су на слици 7.15. Код одскочног одзива се може приметити да се прескок смањују а да се време смирења повећава кроз итерације док време успона остаје скоро непромењено. Код одзива на одскочни поремећај види се повећање времена смирења.



Слика 7.15 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_4 (6.10) са параметром $n=3$ применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)

Поступак почетног подешавања је поновљен применом О-Х методе и добијене су следеће вредности параметара ПИД регулатора: $K_p=0,9$, $T_i=1,67$ и $T_d=0,49$. Примењена је предложена метода у режиму рада Смањење прескока и након 5 итерација су

добијени следећи параметри ПИД регулатора: $K_p=1,18$, $T_i=2,42$ и $T_d=0,46$. Одзиви система кроз итерације на јединичну одскочну побуду и поремећај приказани су на слици 7.15. Код одскочног одзива се може приметити да се прескок и време смирења смањују кроз итерације док време успона остаје скоро непромењено. Код одзива на одскочни поремећај је уочљиво побољшање кроз итерације.

Упоредо су израчунате и мере перформанси и робусности које су приказане у табели 7.13 што омогућава лакше поређење. Из добијених резултата се може видети да предложена метода испуњава претходно постављене циљеве. Међутим, уочљива је деградација осталих мера перформанси када је за почетно подешавање изабрана Ц-Н метода.

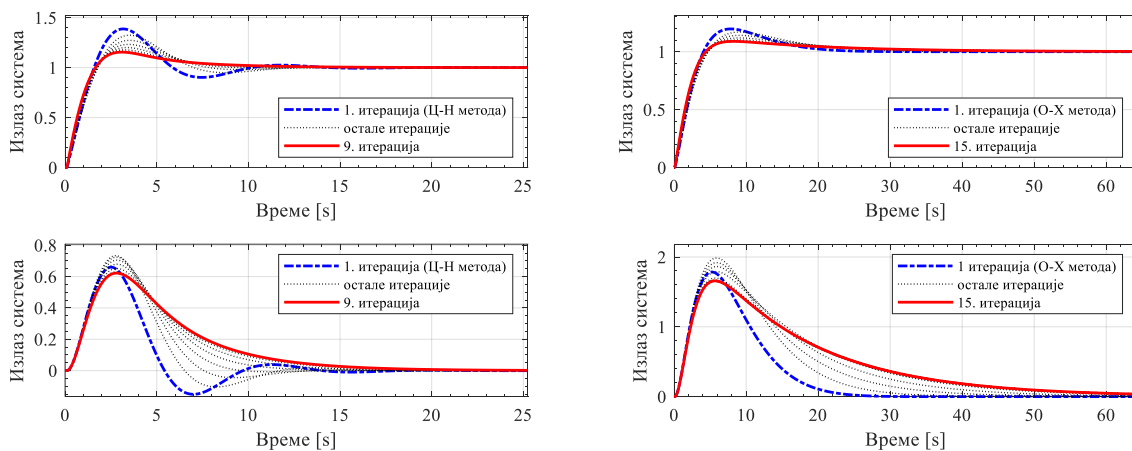
Табела 7.13 Перформансе и робусност ПИД регулатора за процес G_4 (6.10) са параметром $n=3$ постигнуте применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом и О-Х методом

	t_r [s]	σ [%]	t_s [s]	IAE	p	t_{sd} [s]	IAE _d	M_s
Ц-Н метода	1,87	6,06	5,37	1,50	0,33	8,70	1,26	1,32
ФЕС у режиму рада Смањење прескока	1,79	0,12	7,91	1,49	0,31	15,01	1,49	1,33
Унапређење [%]	-4,27	-98,07	47,33	-0,22	-5,80	72,57	18,32	0,67
О-Х метода	2,93	9,74	9,45	2,55	0,47	13,40	2,16	1,34
ФЕС у режиму рада Смањење прескока	2,88	0,34	4,46	2,05	0,43	12,64	2,04	1,28
Унапређење [%]	-1,85	-96,55	-52,80	-19,59	-7,29	-5,66	-5,36	-4,56

7.3.2.4 Пример: Процес са интегратором

Процес G_6 (6.12) са параметром $T=0,9$ ће бити размотрен у овом примеру исто као и у анализи претходног режима рада.

Након почетног подешавања параметара коришћењем Ц-Н методе добијене су следеће вредности параметара ПИД регулатора: $K_p=1,2$, $T_i=2$ и $T_d=0,5$. Аутоматско подешавање је затим примењено у режиму рада Смањење прескока и после 9 итерација добијени су следећи параметри ПИД регулатора: $K_p=1,42$, $T_i=5,28$ и $T_d=0,6$. Одзиви система на јединичну одскочну побуду и поремећај кроз итерације се могу видети на слици 7.16. Уочава се на одскочном одзиву да се прескок и време смирења смањују кроз итерације док време успона остаје непромењено. Са одскочног одзива на поремећај се уочава побољшање од итерације до итерације.



Слика 7.16 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_6 (6.12) са параметром $T=0,9$ применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)

Поступак почетног подешавања је поновљен применом О-Х методе и добијене су следеће вредности параметара ПИД регулатора: $K_p=0,45$, $T_i=8$ и $T_d=0,5$. Затим је поновљено и аутоматско подешавање у режиму рада Смањење прескока и након 15 итерација добијени су следећи параметри ПИД регулатора: $K_p=0,53$, $T_i=17,67$ и $T_d=0,6$. На слици 7.16 су приказани одзиви система кроз итерације на јединичну одскочну побуду и поремећај. У одскочном одзиву се уочава да се прескок и време смирења смањују кроз итерације док време успона остаје непромењено. На одскочном одзиву је видљиво повећање времена смирења за разлику од претходног случаја у овом примеру.

Мере перформанси и робусности су упоредо израчунате и приказане су у табели 7.14 како би се извршило лакше поређење. Из табеле се може видети да предложена метода испуњава претходно постављене циљеве за обе методе почетног подешавања чак уз побољшање осетљивости. Приметна је деградација одзива на одскочни поремећај код оба почетна подешавања.

Табела 7.14 Перформансе и робусност ПИД регулатора за процес G_6 (6.12) са параметром $T=0,9$ постигнуте применом предложене методе у режиму рада Смањење прескока након почетног подешавања Ц-Н методом и О-Х методом

	t_r [s]	o [%]	t_s [s]	IAE	p	t_{sd} [s]	IAE _d	M_s
Ц-Н метода	1,16	38,70	12,62	2,15	0,66	13,17	2,51	1,45
ФЕС у режиму рада Смањење прескока	1,16	15,49	10,16	1,44	0,62	17,83	3,73	1,11
Унапређење [%]	0,00	-59,97	-19,48	-32,93	-5,75	35,46	48,68	-23,50
О-Х метода	2,98	19,67	20,59	3,70	1,79	23,58	17,79	1,12
ФЕС у режиму рада Смањење прескока	3,01	8,92	31,84	3,39	1,66	65,51	33,15	1,07
Унапређење [%]	1,16	-54,65	54,61	-8,27	-7,11	177,77	86,36	-4,47

7.3.2.5 Анализа над целим скупом процеса

Како би се убрзао процес тестирања, за свих 129 процеса из скупа, за овај режим рада примењена је већ креирана програмска скрипта (прилог II.3).

Поступак почетног подешавања је извршен за све процесе из скупа коришћењем Ц-Н методе. Након тога је извршен и поступак аутоматског подешавања параметара применом предложене методе у режиму рада Смањење прескока. На основу добијених резултата израчуната су просечна побољшања перформанси и робусности у односу на почетно подешавање. Просечна побољшања груписана по типу процеса су приказана у табели 7.15.

Табела 7.15 Просечно побољшање вредности перформанси и робусности за све процесе постигнуто предложеном методом у режиму рада Смањење прескока након почетног подешавања Ц-Н методом

Процес	Број	t_r [%]	o [%]	t_s [%]	IAE [%]	p [%]	t_{sd} [%]	IAE _d [%]	M_s [%]	Итер.
G ₁ (6.7)	21	-13,28	-49,05	112,05	84,09	6,60	160,81	269,82	49,16	34,43
G ₂ (6.8)	20	-20,96	-83,92	-4,32	-8,26	-3,92	27,10	9,84	17,12	7,50
G ₃ (6.9)	10	-1,61	-64,69	-6,13	-22,82	-5,30	38,51	31,43	-5,72	6,30
G ₄ (6.10)	5	-64,31	-99,61	35,91	2,77	-7,87	41,15	6,48	30,53	6,20
G ₅ (6.11)	8	-2,18	-58,98	6,20	-6,69	-4,29	28,66	14,09	-0,28	10,63
G ₆ (6.12)	8	0,13	-53,87	17,09	-19,04	-3,36	80,28	106,72	-7,70	24,25
G ₇ (6.13)	36	-1,19	-56,45	19,85	6,42	-4,07	83,87	113,53	4,30	16,42
G ₈ (6.14)	11	-62,83	-86,11	28,57	1,81	-5,90	26,11	1,42	26,54	14,91
G ₉ (6.15)	10	-5,56	-85,49	29,76	-0,43	-7,02	74,35	31,68	-1,19	6,50
Укупно	129	-14,28	-66,59	30,21	11,06	-2,91	73,11	89,89	14,27	16,02

Предложена метода у режиму рада Смањење прескока постигла је значајно просечно смањење прескока од преко 66 %, што је примарни циљ овог режима рада. Истовремено, видљиво је просечно побољшање времена успона и мала деградација времена смирења и осетљивости. За остале перформансе је видљива деградација у мањој или већој мери након примене аутоматског подешавања. Просечно трајање поступка, узимајући у обзир све процесе, је приближно 16 итерација.

Поступак почетног подешавања је поновљен за све процесе коришћењем О-Х методе. Затим је поновљен и поступак аутоматског подешавања параметара применом предложене методе у режиму рада Смањење прескока. На основу добијених резултата израчуната су просечна побољшања перформанси и робусности у односу на почетно подешавање. Просечна побољшања груписана по типу процеса, приказана су у табели 7.16.

Табела 7.16 Просечно побољшање вредности перформанси и робусности за све процесе постигнуто предложеном методом у режиму рада Смањење прескока након почетног подешавања О-Х методом

Процес	Број	t_r [%]	σ [%]	t_s [%]	IAE [%]	p [%]	t_{sd} [%]	IAE _d [%]	M_s [%]	Итер.
G ₁ (6.7)	21	-15,50	-83,36	-0,83	-16,21	-5,64	56,10	20,59	15,33	5,10
G ₂ (6.8)	20	-3,21	-86,03	-17,91	-16,93	-5,19	-9,46	-7,69	2,02	4,70
G ₃ (6.9)	10	-2,11	-72,63	-11,68	-21,48	-8,52	50,34	21,59	-2,81	5,30
G ₄ (6.10)	5	-3,22	-99,18	-18,25	-9,82	-4,48	-5,57	-5,49	1,36	4,00
G ₅ (6.11)	8	-1,53	-90,90	-31,21	-17,50	-6,86	-5,27	-1,43	-2,35	4,13
G ₆ (6.12)	8	-6,10	-57,55	53,03	-11,62	-10,88	183,84	84,49	6,02	35,38
G ₇ (6.13)	36	-4,99	-75,99	-22,12	-23,43	-8,39	3,79	5,62	1,94	4,78
G ₈ (6.14)	11	-3,81	-100,00	-18,03	-12,48	-5,99	-3,80	-7,22	5,33	4,27
G ₉ (6.15)	10	-1,82	-94,35	-10,87	-14,44	-6,27	3,97	0,96	-1,40	4,40
Укупно	129	-5,64	-82,64	-11,73	-17,84	-7,00	23,47	9,80	3,76	6,61

Предложена метода у режиму рада Смањење прескока постигла је значајно просечно смањење прескока од преко 82 %, што је примарни циљ овог режима рада. Истовремено, видљиво је мало просечно побољшање времена успона и времена смирења. За остале перформансе је уочљиво мало просечно побољшање, осим код времена смирења код одскочног поремећаја где је уочљива деградација. Осетљивост је у просеку постигла благо погоршање које се може занемарити. Просечно трајање поступка узимајући у обзир све процесе је приближно 6 итерација.

7.3.3 Предложена метода у режиму рада Балансирање

Анализирани су резултати добијени применом предложене методе у режиму рада Балансирање, коришћењем обе методе за почетно подешавање параметара и испитана је испуњеност постављених циљева.

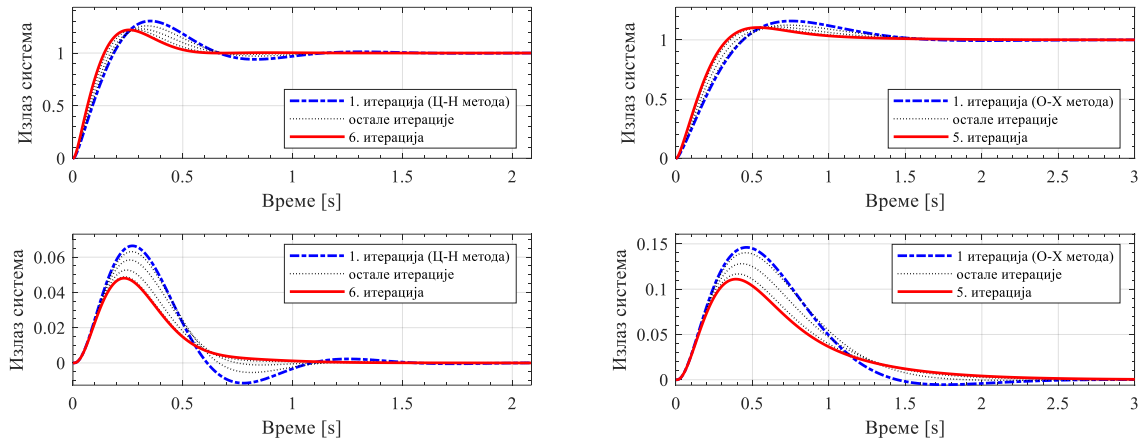
7.3.3.1 Пример: Процес са доминантном вредношћу временске константе

Као пример у овом режиму рада ће бити размотрен процес G_5 (6.11) са параметром $T=0,1$ исто као и у анализи претходног.

Након почетног подешавања параметара коришћењем Ц-Н методе добијене су следеће вредности параметара ПИД регулатора: $K_p=11,11$, $T_i=0,22$ и $T_d=0,05$. Примењена је предложена метода у режиму рада Балансирање и након 6 итерација су добијени следећи параметри ПИД регулатора: $K_p=17,95$, $T_i=0,31$ и $T_d=0,05$. Одзиви система на јединичну одскочну побуду и поремећај кроз итерације су приказани на слици 7.17. Са слике се може закључити да се време успона, прескок и време смирења код одзива

на одскочну побуду смањује кроз итерације. У одзиву на одскочни поремећај видљиво је побољшање од итерације до итерације.

Поступак почетног подешавања је поновљен применом О-Х методе и добијене су следеће вредности параметара ПИД регулатора: $K_p=4,37$, $T_i=0,44$ и $T_d=0,05$. Након примене аутоматског подешавања у режиму рада Балансирање кроз 5 итерација су добијени следећи параметри ПИД регулатора: $K_p=6,94$, $T_i=0,60$ и $T_d=0,05$. Одзиви система кроз итерације на јединичну одскочну побуду и поремећај кроз итерације приказани су на слици 7.17. Са слике се може закључити да се одзиви понашају на веома сличан начин као и код Ц-Н методе уз мало повећање времена смирења код одскочног одзива на поремећај.



Слика 7.17 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_5 (6.11) са параметром $T=0,1$ применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)

Мере перформанси и робусности су упоредо израчунате и приказане су у табели 7.17 ради лакше упоредне анализе. Посматрањем добијених резултата може се уочити да предложена метода испуњава претходно постављене циљеве без обзира на вредност почетних параметара уз побољшање скоро свих посматраних индекса.

Табела 7.17 Перформансе и робусност ПИД регулатора за процес G_5 (6.11) са параметром $T=0,1$ постигнуте применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом и О-Х методом

	t_r [s]	σ [%]	t_s [s]	IAE	p	t_{sd} [s]	IAE _d	M_s
Ц-Н метода	0,14	30,44	1,04	0,20	0,07	1,40	0,03	1,31
ФЕС у режиму рада Балансирање	0,10	21,98	0,53	0,12	0,05	1,05	0,02	1,21
Унапређење [%]	-25,34	-27,77	-49,40	-39,60	-27,62	-25,57	-33,59	-7,44
О-Х метода	0,32	15,93	1,47	0,31	0,15	2,13	0,11	1,14
ФЕС у режиму рада Балансирање	0,24	10,43	1,20	0,21	0,11	2,29	0,09	1,15
Унапређење [%]	-25,92	-34,50	-18,69	-31,63	-24,00	7,46	-19,10	0,82

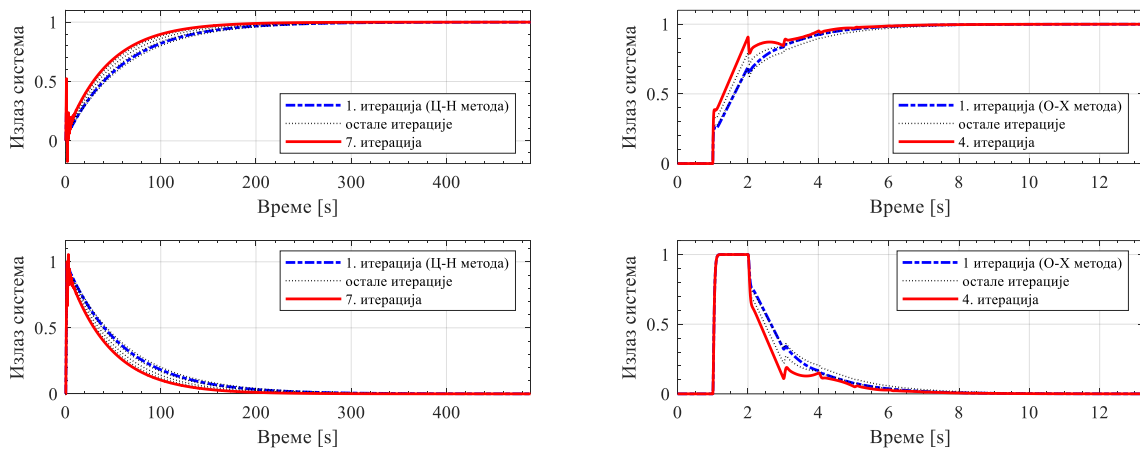
7.3.3.2 Пример: Процес са доминантном вредношћу временског кашњења

Као и у анализи претходног режима рада и у овом примеру ће бити размотрен процес G_2 (6.8) са параметром $T=0,02$.

Након почетног подешавања параметара коришћењем Ц-Н методе добијене су следеће вредности параметара ПИД регулатора: $K_p=0,03$, $T_i=2,02$ и $T_d=0,51$. Примењено је аутоматско подешавање у режиму рада Балансирање и после 7 итерација добијени су следећи параметри ПИД регулатора: $K_p=0,07$, $T_i=2,92$ и $T_d=0,40$. Одзиви

система на јединичну одскочну побуду и поремећају кроз итерације могу се видети на слици 7.18. Са слике се може закључити да се време успона смањује и време смирења смањују кроз итерације док је прескок остао непромењен. Код одзива на одскочни поремећај је видљиво побољшање од итерације до итерације.

Поступак почетног подешавања је поновљен применом О-Х методе и добијене су следеће вредности параметара ПИД регулатора: $K_p=0,21$, $T_i=0,43$ и $T_d=0,04$. Примењено је аутоматско подешавање у режиму рада Балансирање и након 4 итерација добијени су следећи параметри ПИД регулатора: $K_p=0,32$, $T_i=0,58$ и $T_d=0,04$. Одзиви система на јединичну одскочну побуду и поремећај приказани су на слици 7.18. Посматрајући одзиве на одскочну побуду и поремећај увиђа се слично понашање као и код Ц-Н методе с тим да се временска скала разликује за ред величине.



Слика 7.18 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_2 (6.8) са параметром $T=0,02$ применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)

Израчунате мере перформанси и робусности су приказане у табели 7.18 ради лакшег поређења. Посматрањем добијених резултата уочава се да предложена метода испуњава претходно постављене циљеве. Такође се примећује да није дошло до повећања прескока што је свакако очекивано у овом режиму рада. Због убрзања система дошло је до деградације осетљивости и то у већој мери кад је за почетно подешавање примењена Ц-Н метода у односу на О-Х методу.

Табела 7.18 Перформансе и робусност ПИД регулатора за процес G_2 (6.8) са параметром $T=0,02$ постигнуте применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом и О-Х методом

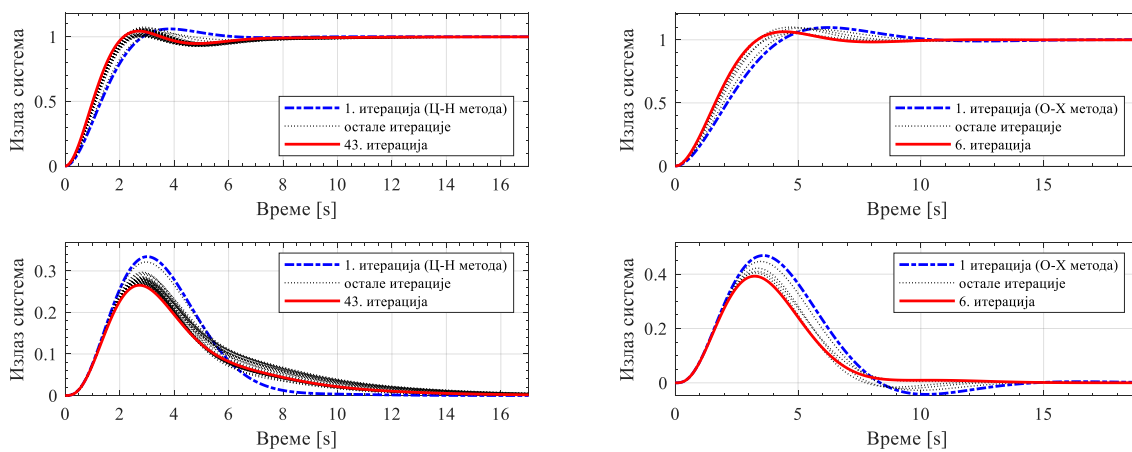
	t_r [s]	σ [%]	t_s [s]	IAE	p	t_{sd} [s]	IAE _d	M_s
Ц-Н метода	133,42	0,00	229,09	57,83	1,01	229,71	57,83	1,55
ФЕС у режиму рада Балансирање	99,80	0,00	172,65	42,73	1,06	171,29	42,73	3,14
Унапређење [%]	-25,20	0,00	-24,64	-26,11	4,78	-25,43	-26,11	101,95
О-Х метода	2,63	0,00	5,75	2,00	1,00	6,79	2,00	1,43
ФЕС у режиму рада Балансирање	0,98	0,00	5,35	1,70	1,00	6,40	1,70	1,67
Унапређење [%]	-62,56	0,00	-6,93	-15,20	0,01	-5,87	-15,20	16,69

7.3.3.3 Пример: Процес са равномерним вредностима временске константе и временског кашњења

Као пример, у овом режиму рада биће размотрен процес G_4 (6.10) са параметром $n=3$ исто као и у анализи претходног.

Након почетног подешавања параметара коришћењем Ц-Н методе добијене су следеће вредности параметара ПИД регулатора: $K_p=1,86$, $T_i=2,35$ и $T_d=0,59$. Затим је примењено аутоматско подешавање у режиму рада Балансирање и после 43 итерације добијени су следећи параметри ПИД регулатора: $K_p=2,77$, $T_i=3,23$ и $T_d=0,63$. Одзиви система кроз итерације на јединичну одскочну побуду и поремећај кроз итерације могу се видети на слици 7.19. Са одскочног одзива на побуду се може закључити да се време успона и прескок смањују кроз итерације док се време смирења повећава. У одзиву на одскочни поремећај видљиво је побољшање кроз итерације али уз мало повећање времена смирења.

Поступак почетног подешавања је поновљен применом О-Х методе и добијене су следеће вредности параметара ПИД регулатора: $K_p=0,9$, $T_i=1,67$ и $T_d=0,49$. Након примене аутоматског подешавања у режиму рада Балансирање кроз 6 итерација добијени су следећи параметри ПИД регулатора: $K_p=1,49$, $T_i=2,36$ и $T_d=0,42$. Одзиви система кроз итерације на јединичну одскочну побуду и поремећају приказани су на слици 7.19. Може се закључити да се одзиви понашају на веома сличан начин као код претходне методе почетног подешавања али за доста мање итерација.



Слика 7.19 Одзив систем на јединичну одскочну побуду (горе) и на јединични одскочни поремећај (доле) за процес G_4 (6.10) са параметром $n=3$ применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)

Упоредо су израчунате мере перформанси и робусности и приказане су у табели 7.19 ради лакшег поређења. Може се видети да предложена метода испуњава претходно постављене циљеве уз занемарљиву деградацију осетљивости за обе методе почетног подешавања. Видљиво је побољшање и осталих мера перформанси изузимајући времена смирења код одзива на одскочну побуду и поремећај за почетно подешене параметре применом Ц-Н методе.

Табела 7.19 Перформансе и робусност ПИД регулатора за процес G_4 (6.10) са параметром $n=3$ постигнуте применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом и О-Х методом

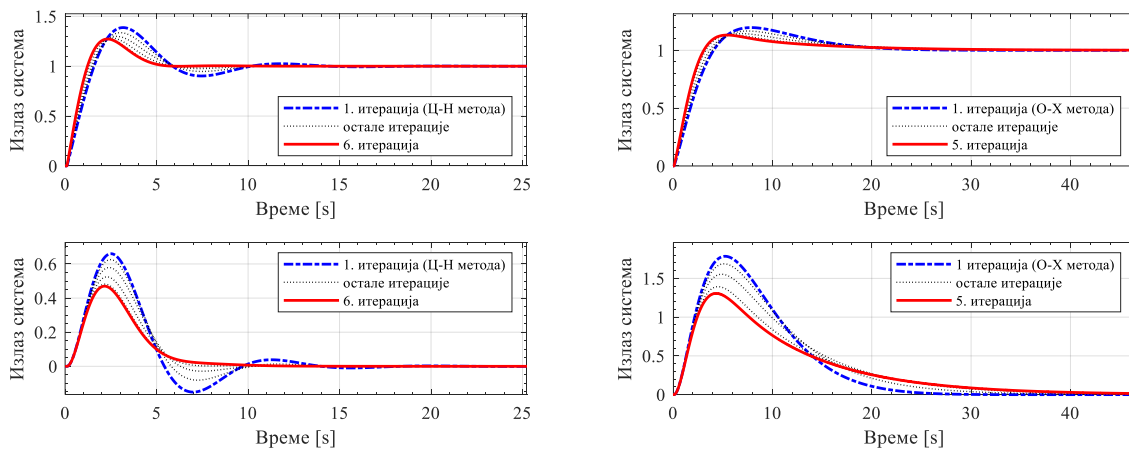
	t_r [s]	σ [%]	t_s [s]	IAE	p	t_{sd} [s]	IAE _d	M_s
Ц-Н метода	1,87	6,06	5,37	1,50	0,33	8,70	1,26	1,32
ФЕС у режиму рада Балансирање	1,39	4,39	6,66	1,24	0,27	13,92	1,17	1,37
Унапређење [%]	-25,49	-27,55	24,08	-16,92	-20,60	60,02	-7,63	4,22
О-Х метода	2,93	9,74	9,45	2,55	0,47	13,40	2,16	1,34
ФЕС у режиму рада Балансирање	2,19	6,41	5,94	1,84	0,39	11,85	1,59	1,37
Унапређење [%]	-25,13	-34,18	-37,11	-27,89	-16,23	-11,52	-26,54	2,54

7.3.3.4 Пример: Процес са интегратором

Као пример у овом режиму рада биће размотрен процес G_6 (6.12) са параметром $T=0,9$ исто као и у анализи претходног.

Након почетног подешавања параметара коришћењем Ц-Н методе добијене су следеће вредности параметара ПИД регулатора: $K_p=1,2$, $T_i=2$ и $T_d=0,5$. Аутоматско подешавање у режиму рада Балансирање је примењено и након 6 итерација добијени су следећи параметри ПИД регулатора: $K_p=1,94$, $T_i=2,97$ и $T_d=0,45$. Одзиви система на јединичну одскачну побуду и поремећају кроз итерације приказани су на слици 7.20. Код одскочног одзива се може видети да се време успона, прескок и време смирења смањују кроз итерације. Код одскочног одзива на поремећај видљиво је побољшање из итерације у итерацију.

Поступак почетног подешавања је поновљен применом О-Х методе и добијене су следеће вредности параметара ПИД регулатора: $K_p=0,45$, $T_i=8$, и $T_d=0,5$. Аутоматско подешавање у режиму рада Балансирање је примењено и након 5 итерација добијени су следећи параметри ПИД регулатора: $K_p=0,69$, $T_i=11,33$ и $T_d=0,45$. Одзиви система кроз итерације на јединичну одскачну побуду и поремећају приказани су на слици 7.20. Посматрајући одзиве долази се до закључка да се одзиви понашају на веома сличан начин као код претходне методе почетног подешавања уз изузетак да се код одзива на одскачни поремећај уочава повећање времена смирења.



Слика 7.20 Одзив систем на јединичну одскачну побуду (горе) и на јединични одскачни поремећај (доле) за процес G_6 (6.12) са параметром $T=0,9$ применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом (лево) и О-Х методом (десно)

Мере перформанси и робусности су упоредо израчунате и могу се видети у табели 7.20.

Табела 7.20 Перформансе и робусност ПИД регулатора за процес G_6 (6.12) са параметром $T=0,9$ постигнуте применом предложене методе у режиму рада Балансирање након почетног подешавања Ц-Н методом и О-Х методом

	t_r [s]	σ [%]	t_s [s]	IAE	p	t_{sd} [s]	IAE _d	M_s
Ц-Н метода	1,16	38,70	12,62	2,15	0,66	13,17	2,51	1,45
ФЕС у режиму рада Балансирање	0,86	27,24	4,98	1,21	0,47	9,38	1,54	1,24
Унапређење [%]	-25,48	-29,61	-60,56	-43,61	-28,83	-28,79	-38,73	-14,24
О-Х метода	2,98	19,67	20,59	3,70	1,79	23,58	17,79	1,12
ФЕС у режиму рада Балансирање	2,20	13,19	22,01	2,76	1,31	40,98	16,34	1,14
Унапређење [%]	-26,08	-32,92	6,87	-25,45	-26,83	73,77	-8,12	1,99

Из добијених резултата се може закључити да предложена метода испуњава претходно постављене циљеве. Видљиво је побољшање по свим индексима перформанси осим за време смирења код одскочног одзива на поремећај за почетно подешавање О-Х методом.

7.3.3.5 Анализа над целим скупом процеса за овај режим рада

Како би се убрзао процес тестирања, за свих 129 процеса из скупа, за овај режим рада примењена је већ креирана програмска скрипта (прилог П.3).

Поступак почетног подешавања је извршен са све процесе из скупа коришћењем Ц-Н методе. Након тога је извршен и поступак аутоматског подешавања параметара применом предложене методе у режиму рада Балансирање. На основу добијених резултата израчуната су просечна побољшања перформанси и робусности у односу на почетно подешавање. Просечна побољшања груписана по типу процеса су приказана у табели 7.21.

Предложена метода у режиму рада Балансирање је постигла значајно приближно подједнако просечно побољшање времена успона (скоро 23 %) и вредности прескока (око 30 %) што је и главни циљ овог режима рада. Због делимичног убрзавања система долази до деградације осетљивости у просеку за 40 %. За остале перформансе је видљива деградација у мањој или већој мери након примене аутоматског подешавања. Просечно трајање поступка узимајући у обзир све процесе је приближно 21 итерација.

Табела 7.21 Просечно побољшање вредности перформанси и робусности за све процесе постигнуто предложеном методом у режиму рада Балансирање након почетног подешавања Ц-Н методом

Процес	Број	t_r [%]	σ [%]	t_s [%]	IAE [%]	p [%]	t_{sd} [%]	IAE _d [%]	M_s [%]	Итер.
G ₁ (6.7)	21	-38,93	-24,50	198,76	255,29	3,06	207,83	481,93	127,31	34,24
G ₂ (6.8)	20	-39,18	-52,90	-14,94	-21,56	-11,29	8,96	-16,31	28,13	7,65
G ₃ (6.9)	10	-25,35	-27,66	-18,77	-30,44	-24,22	32,10	-12,20	4,06	9,20
G ₄ (6.10)	5	-67,43	-85,51	13,71	-9,84	-9,92	21,36	-7,98	24,13	11,20
G ₅ (6.11)	8	-26,51	-54,87	-2,05	-18,45	-18,07	30,39	-7,31	8,15	16,38
G ₆ (6.12)	8	-18,97	-15,63	61,05	95,60	-17,33	105,75	202,38	36,49	18,63
G ₇ (6.13)	36	7,49	-7,93	94,23	88,98	-12,91	132,13	253,41	31,32	22,44
G ₈ (6.14)	11	-41,70	-66,02	21,73	-0,15	-9,10	19,81	-0,17	27,55	25,18
G ₉ (6.15)	10	-20,19	-7,24	33,64	-4,81	-18,87	68,21	7,03	6,43	40,40
Укупно	129	-22,84	-30,42	63,53	64,71	-11,55	90,83	158,02	40,69	21,62

Поступак почетног подешавања је поновљен за све процесе коришћењем О-Х методе. Затим је поновљен и поступак аутоматског подешавања параметара применом предложене методе у режиму рада Балансирање. На основу добијених резултата израчуната су просечна побољшања перформанси и робусности у односу на почетно подешавање. Просечна побољшања груписана по типу процеса приказана су у табели 7.22.

Предложена метода у режиму рада Балансирање је постигла значајно приближно подједнако просечно побољшање времена успона (око 30 %) и вредности прескока (око 35 %) што је и главни циљ овог режима рада. Деградација осетљивости је скоро занемарљива и просечно износи око 4 %. За остале перформансе је видљиво просечно побољшање у мањој или већој мери након примене аутоматског подешавања. Просечно трајање поступка узимајући у обзир све процесе је приближно 4 итерације.

Табела 7.22 Просечно побољшање вредности перформанси и робусности за све процесе постигнуто предложеном методом у режиму рада Балансирање након почетног подешавања О-Х методом

Процес	Број	t_r [%]	σ [%]	t_s [%]	IAE [%]	p [%]	t_{sd} [%]	IAE _d [%]	M_s [%]	Итер.
G ₁ (6.7)	21	-44,45	-52,18	-14,31	-20,08	-8,64	-4,37	-17,32	13,57	3,33
G ₂ (6.8)	20	-31,38	-42,55	-22,95	-21,84	-10,41	-27,02	-25,01	6,53	4,25
G ₃ (6.9)	10	-25,97	-21,03	-20,09	-25,70	-19,18	-5,29	-25,25	3,80	4,80
G ₄ (6.10)	5	-26,65	-70,19	-34,12	-19,07	-7,82	-17,30	-17,90	8,63	4,40
G ₅ (6.11)	8	-25,88	-28,72	-30,41	-25,32	-14,28	-29,01	-25,52	5,30	5,50
G ₆ (6.12)	8	-29,24	-22,09	-0,21	-20,44	-21,30	38,64	-15,29	6,37	3,50
G ₇ (6.13)	36	-28,64	-23,49	-21,59	-24,69	-15,61	-29,54	-25,33	5,26	4,08
G ₈ (6.14)	11	-26,83	-59,30	-36,15	-21,41	-7,59	-23,71	-19,63	8,94	4,64
G ₉ (6.15)	10	-26,09	-16,39	-15,77	-24,47	-15,63	-5,17	-22,85	9,05	5,90
Укупно	129	-30,87	-35,48	-21,00	-22,84	-13,23	-16,05	-22,39	7,50	4,29

8 Експериментални резултати

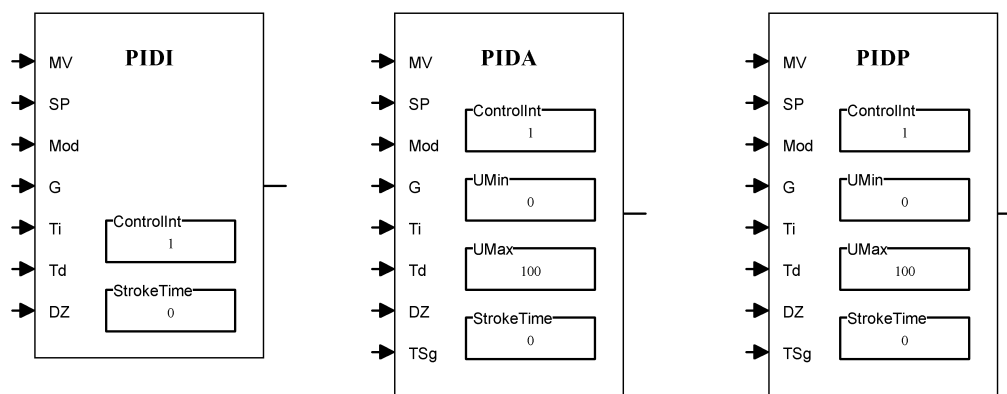
У овом поглављу биће дат опис имплементације предложене методе на реалном контролеру који се користи у аутоматизацији стамбено-пословних објеката. Биће описана експериментална поставка док ће на крају бити приказани добијени резултати са дискусијом.

8.1 Имплементација предложене методе на реалном контролеру

Предложена метода је имплементирана коришћењем FBD (Function block diagram) програмског језика на реалном контролеру који је део EcoStruxure Building Operation³ платформе за аутоматизацију стамбено-пословних објеката.

8.1.1 ПИД регулатор у FBD програмском језику

У FBD програмском језику које се користи за програмирање контролера ПИД закон управљања је реализован у три различита блока [103] која су приказана на слици 8.1. Преглед улаза и излаза као и параметара ових блокова је дат у табели 8.1.



Слика 8.1 ПИД блокови у FBD програмском језику

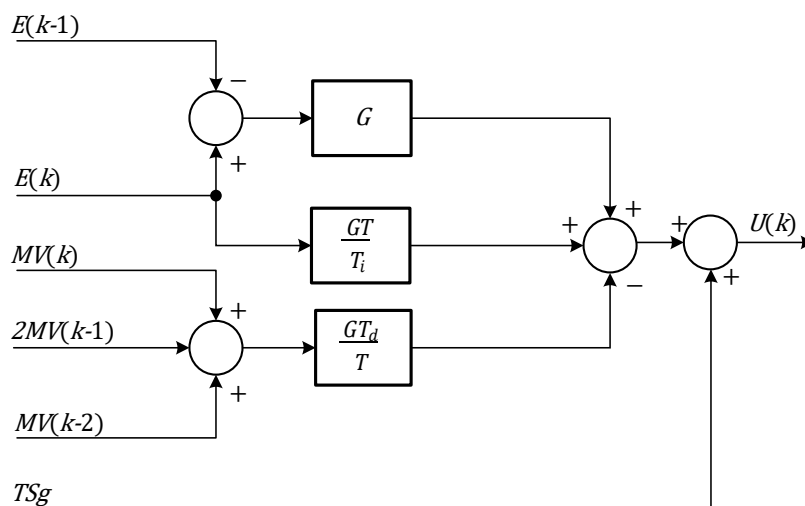
Табела 8.1 Преглед улаза и излаза ПИД блокова

Улаз/излаз блока	Тип	Опис
MV	улаз	Тренутна вредност
SP	улаз	Задата вредност
Mode	улаз	Режим рада
G	улаз	Пропорционално појачање
Ti	улаз	Временска константа интегралне
Td	улаз	Временска константа диференцирања
DZ	улаз	Зона неактивности
TSg	улаз	Вредност управљања у претходном кораку
Controllnt (T)	параметар	Време одабирања
U _{Min}	параметар	Минимална вредност управљања
U _{Max}	параметар	Максимална вредност управљања
StrokeTime	параметар	Временска константа актуатора
Output	излаз	Сигнал управљања

³ EcoStruxure Building Operation је производ компаније Schneider Electric и сам назив је регистровани заштитни знак

8.1.1.1 PIDI и PIDA блокови

У оквиру блокова PIDI и PIDA реализован је дискретни инкрементални ПИД закон управља са два степена слободe описан изразом 3.25 усвајајући да је тежински фактор $b=1$. У оквиру PIDI блока реализована је брзинска форма која на излазу даје само промену управљања. Насупрот томе, у оквиру PIDA блока је реализована позициона форма која даје комплетну вредност управљања на основу повратног сигнала управљања. Блок дијаграм реализације PIDA блока је да на слици 8.2 док би са PIDI исто изгледао само без улаза за претходну вредност управљања TSg .



Слика 8.2 Блок дијаграм закона управљања PIDA блока

Време одабирања односно време извршавања алгоритма *ControlInt* (T) мора да буде целобројан умножак програмског циклуса. Промена вредности управљања у току једне периоде извршавања се може ограничити временском константом актуатора *StrokeTime* (0 – ограничење искључено) код оба блока. Код PIDA управљачки сигнал се додатно ограничава у опсегу од U_{min} до U_{max} .

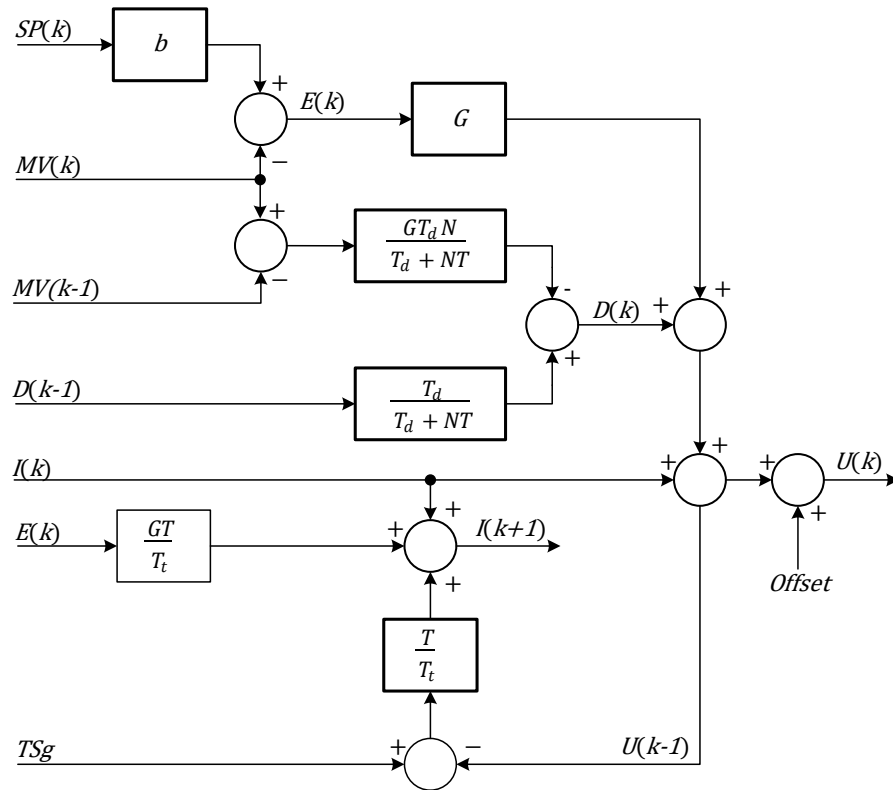
У оквиру реализације ова два блока нису разматрани проблеми филтрирања диференцијалног дејства. Такође код PIDA блока није разматран *integrator windup* проблем иако се излаз регулатора ограничава на опсег U_{min} до U_{max} . Са поменутиим недостацима ова два ПИД блока су у доста случајева неупотребљиви у пракси.

8.1.1.2 PIDP блок

У оквиру блока PIDP реализован је дискретни инкрементални ПИД закон управља са два степена слободe описан изразом 3.25 усвајајући да је тежински фактор $b=0$. Реализована је позициона форма која даје комплетну вредност управљања на основу повратног сигнала управљања.

PIDP блок је накнадно укључен у стандардну библиотеку блокова. Мотив за увођење и предлог реализације овог блока су описани у мастер тези [104]. Блок дијаграм реализације PIDP блока се може видети на слици 8.3.

Време одабирања односно време извршавања алгоритма *ControlInt* (T) мора да буде целобројан умножак програмског циклуса. Промена вредности управљања у току једне периоде извршавања се може ограничити временском константом актуатора *StrokeTime* (0 – ограничење искључено). Управљачки сигнал се додатно ограничава у опсегу од U_{min} до U_{max} .

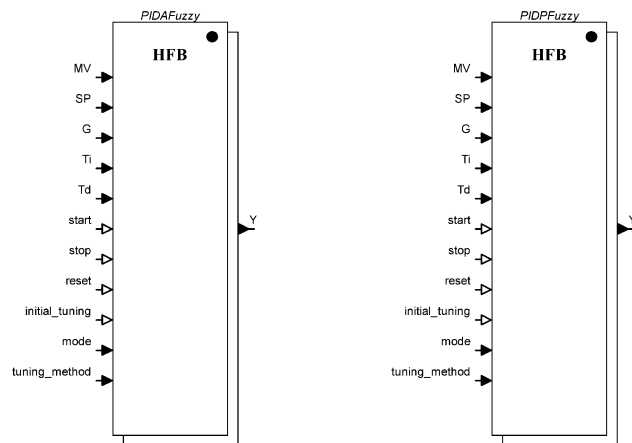


Слика 8.3 Блок дијаграм закона управљања PIDP блока

Ова реализација поседује све елементе модерног ПИД закона управљања. Филтрирање диференцијалног дејства је решено као што је описано у поглављу 3.3.1 уз усвајање временске константе $N=8$. Проблем *integrator windup* је решен као што је описано у поглављу 3.3.3 уз усвајање временске константе $T_i=T_i/2$. Решавањем ова два проблема блок је постао много употребљивији у различитим практичним применама.

8.1.2 Опис имплементације

Предложена метода је имплементирана кроз два посебна FBD хијерархијска блока коришћењем софтверског алата Function Block Editor који је саставни део EcoStruxure Building Operation платформе. Реализовани блокови се могу видети на слици 8.4 и по структури улаза и излаза опонашају већ постојеће FBD блокове са ПИД законом управљања уз додаток улаза за контролу рада методе. Спецификација улаза и излаза је дата у табели 8.2.



Слика 8.4 Хијерархијски блокови PIDAFuzzy и PIDPFuzzy

Табела 8.2 Преглед улаза и излаза блокова *PIDAFuzzy* и *PIDPFuzzy*

Улаз/излаз блока	Тип	Опис
MV	улаз	Тренутна вредност излаза процеса
SP	улаз	Задата вредност
G	улаз	Постојећа вредност пропорционалног појачања
Ti	улаз	Постојећа вредност временске константе интегралне
Td	улаз	Постојећа вредност временске константе диференцирања
start	улаз	Команда за почетак поступка
stop	улаз	Команда за завршетак поступка
reset	улаз	Команда за враћање на почетак
initial_tuning	улаз	Опција додатног почетног подешавања
tuning_method	улаз	Избор методе за почетно подешавање
mode	улаз	Избор режима рада
contorlValue	излаз	Вредност управљања

За примену је довољно само заменити ПИД блок у постојећем регулационом кругу одговарајућим реализованим блоком *PIDAFuzzy* или *PIDPFuzzy*. Овакав начин имплементације предложене методе је замишљен тако да не ремети постојеће управљачке кругове у нормалном раду. Све док се не покрене извршавање методе реализовани блок се понаша као обичан *PIDP* / *PIDA* блок са параметрима који му се прослеђују. Када се покрене процедура нови параметри се генеришу у итеративном поступку све до завршетка процедуре када се наставља рад са последњим прорачунатим параметрима.

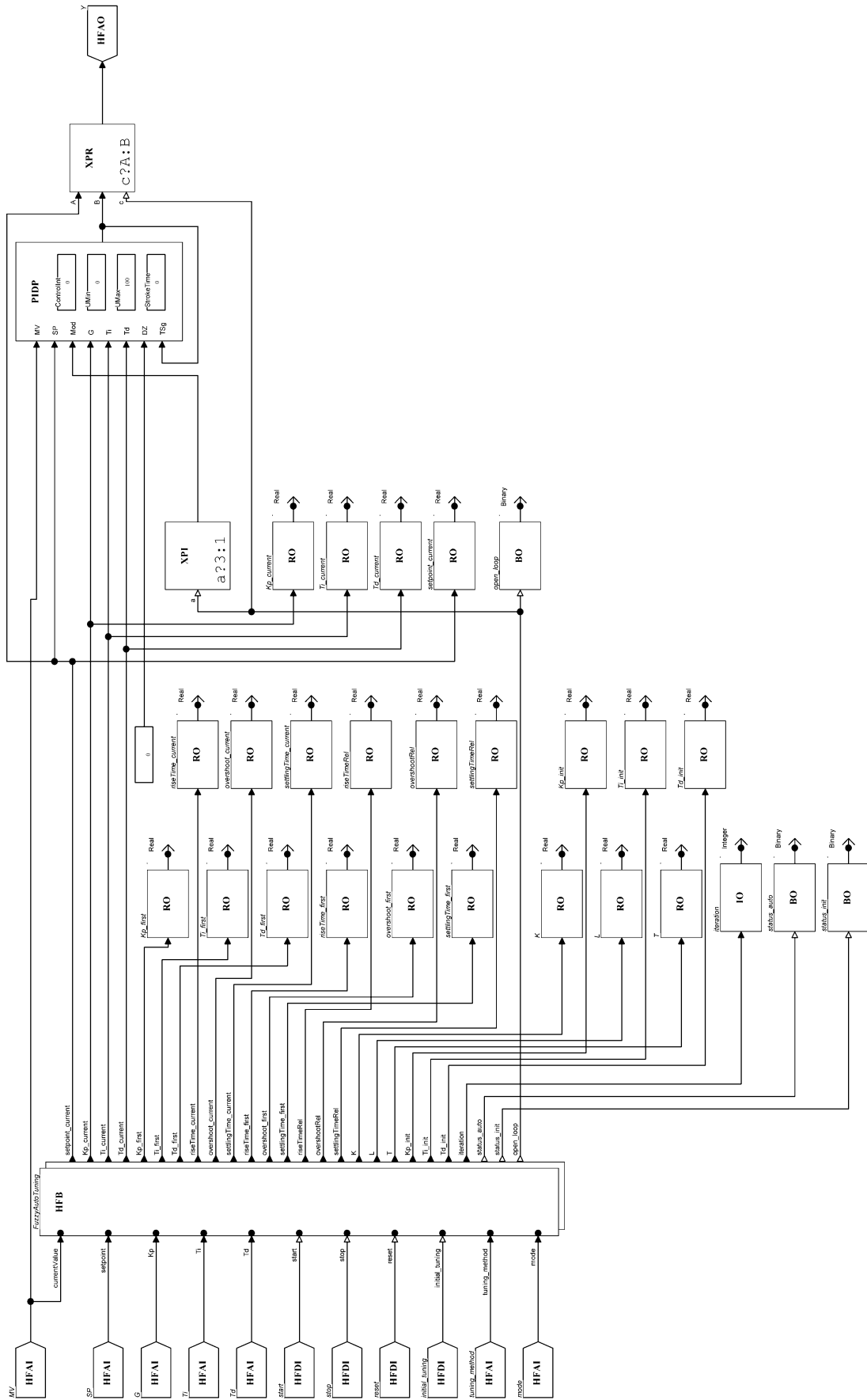
У оквиру реализације *PIDPFuzzy* блока, приказаној на слици 8.5, *PIDP* блок је повезан са посебним блоком *FuzzyAutoTuning* у којем је реализована предложена метода. Поред та два блока повезани су и многобројни излазни блокови који омогућавају надзор рада методе и евентуално повезивање са надзорно-управљачким системом. Реализација *PIDAFuzzy* је потпуно иста као и претходна једино што се као ПИД блок користи *PIDA* уместо *PIDP* блока.

Предложена метода је реализована у оквиру блока *FuzzyAutoTuning* на основу разматрања изнетих у поглављу 6. Фази систем закључивања је реализован као Сугено тип фази закључивања који је описан у поглављу 6.5. Детаљна реализација предложене методе у *FBD* програмском језику (блок *FuzzyAutoTuning*) је дата у прилогу IV.1.

У току имплементације предложене методе у *FBD* програмском језику појавили су се многи изазови који се огледају у ограничењу самог контролера и *FBD* програмског језика.

У оквиру *FBD* програмског језика не постоји могућност креирања променљивих које би могле да чувају вредност у току извршавања програмског кода. Програмски код се извршава пропагацијом сигнала од улазних блокова ка излазним блоковима. Овај проблем се решава применом посебних блокова (*DELR*, *DELI* и *DELB*) који могу да сачувају вредност до следећег програмског циклуса. Применом повратне спреге са таквим блоковима поменути недостатак се успешно решава али се реализација доста усложњава што узрокује успоравање развоја.

Прецизно мерење времена једна је од кључних ставки предложене методе. Међутим *FBD* програмски језик не поседује блок који омогућава мерење протеклог времена у прецизности мањој од једне секунде иако се програмски циклус може поставити и на вредност од 100 ms. Како би се решио овај проблем креиран је посебан блок (*RT* блок у прилогу IV.1) у оквиру којег се налази бројач програмских циклуса који се затим мултиплицирају са вредношћу тренутног програмског циклуса. Бројач је реализован на начин описан као у претходном проблему.

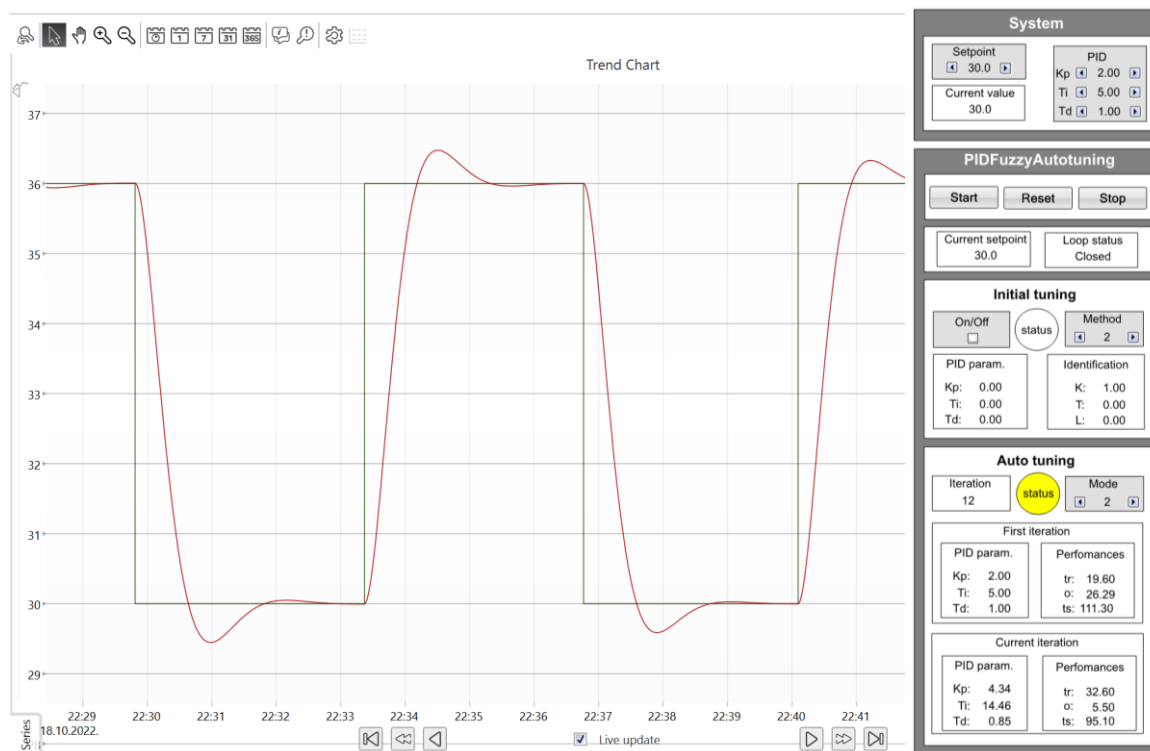


Слика 8.5 Реализација блока PIDPFuzzy

8.1.3 Надзорно-управљачки систем

Поред имплементације предложене методе на реалном контролеру реализован је и надзорно-управљачки систем који омогућава лакшу интеракцију са корисником у експлоатацији предложеног решења.

Графички интерфејс је реализован коришћењем Graphics Editor алата који је саставни део EcoStruxure Building Operation платформе и може се видети на слици 8.6. Састоји се од графичког приказа промене тренутне и задате вредности у зависности од протеклог времена, са леве стране и групе графичких елемената за интеракцију са корисником, са десне стране.



Слика 8.6 Реализовани надзорно-управљачки систем

У горњем делу групе графичких елемената за интеракцију са корисником се налазе контроле за преглед и измену параметара система попут тренутне и задате вредности као и постојећих параметара ПИД регулатора. У доњем делу се налазе контроле за надзор и управљање поступком подешавања параметара ПИД регулатора. Поступак подешавања се може покренути, зауставити или потпуно искључити. Могуће је изабрати и режим рада предложене методе. Након покретања поступка подешавања подразумевано се као почетни користе постојећи параметри ПИД регулатора. Нови почетни параметри се могу прорачунати укључивањем почетног подешавања. Поред индикација рада методе приказане су и вредности почетних и тренутних параметара и прорачунатих перформанси система.

8.2 Имплементација модела топлотне подстаннице на реалном контролеру

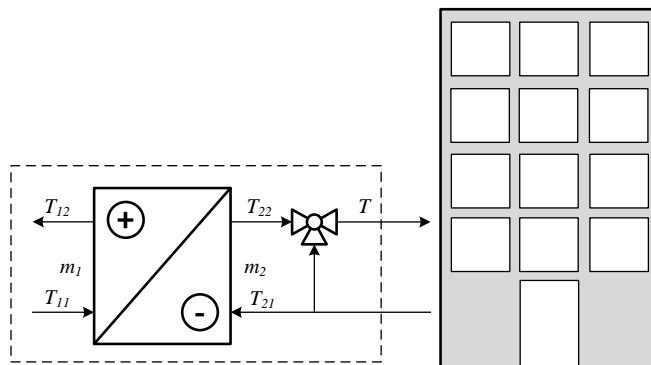
Више од 50 % светске популације, са предвиђањем даљег пораста, живи у урбаним срединама што проузрокује све већу распрострањеност система даљинског

грејања. Системи даљинског грејања су економични, ефикасни и еколошки. Такође, испуњавају строгу регулативу у вези са смањењем емисије CO_2 и повећањем удела обновљиве енергије [105].

У системима даљинског грејања топлотне подстаннице имају важну улогу јер се користе као сепаратори између примарног и секундарног снабдевања топлом водом за системе грејања стамбених и пословних објеката.

Развој управљачких и оптимизационих техника захтева верификацију истих на реалним системима. Међутим, верификација код система даљинског грејања често може бити отежана због недоступности појединих делова система за овакав вид тестирања. Алтернатива је коришћење HIL (Hardware-in-loop) поставке помоћу које ће се верификација извршити на моделу система који ће се извршавати у реалном времену. Предуслов за овај вид тестирања је постојање исправног математичког модела система. Један таква модел је описан у [106] где је извршена анализа и развој математичких модела измењивача топлоте, регулационог вентила, трокраког вентила и фреквентно регулисане центрифугалне пумпе који су максимално конфигурабилни и генерички.

За потребе верификације предложене методе развијен је модел топлотне подстаннице у реалном времену водећи се смерницама изнетим у [53]. Шематски приказ топлотне подстаннице је приказан на слици 8.7.



Слика 8.7 Шематски приказ топлотне подстаннице

У оквиру модела су размотрени један измењивач топлоте вода-вода и један трокраки вентил. На овај начин се само један тип крајњих корисника у објекту може снабдевати водом одговарајуће температуре (нпр. подно грејање, радијаторско грејање, вентилатор конвектор итд.). Модел пумпе је изостављен јер се претпоставља да је проток на секундарној страни константан.

У наставку ће бити дат опис сваког од елемената модела топлотне подстаннице који су имплементирани у LabView⁴ развојном окружењу а чији се изворни код може видети у прилогу III.

8.2.1 Модел измењивача топлоте

У измењивачу топлоте се одвија процес размене топлоте између два флуида различитих температура, односно у системима даљинског грејања два тока воде различитих температура. На основу закона термодинамике ако се претпостави константна специфична топлота флуида, брзина преноса топлоте Q је дата као

⁴ LabView је производ компаније National Instruments и сам назив је регистровани заштитни знак

$$Q = \dot{m}_1 c_{P1} (T_{11} - T_{12}) \quad (8.1)$$

$$Q = \dot{m}_2 c_{P2} (T_{22} - T_{21}) \quad (8.2)$$

где су \dot{m}_1 и \dot{m}_2 примарни и секундарни масени проток воде а c_{P1} и c_{P2} специфична топлота воде на примарној и секундарној страни док су T_{11} и T_{12} температуре разводне и повратне воде на примарној страни односно T_{21} и T_{22} температуре повратне и разводне воде на секундарној страни [107].

Пошто су претходни изрази независни од типа конструкције и распореда протока измењивача топлоте, веза се остварује применом ε - NTU методе која је детаљно описана у [107]. У том случају ефикасност измењивача топлоте је дефинисана максималном могућом брзином преноса топлоте

$$Q_{max} = C_{min} (T_{11} - T_{21}) \quad (8.3)$$

при чему је

$$C_{min} = \min\{\dot{m}_1 c_{P1}, \dot{m}_2 c_{P2}\} \quad (8.4)$$

На основу тога актуелни пренос топлоте је дефинисан

$$Q = \varepsilon Q_{max} = \varepsilon C_{min} (T_{11} - T_{21}) \quad (8.5)$$

где је са ε означена ефикасност измењивача топлоте. У овом случају је изабран тип измењивача топлоте са концентричним цевима у паралелном распореду протока па је ефикасност дата изразом

$$\varepsilon = \frac{1 - e^{-NTU(1+C_r)}}{1 + C_r} \quad (8.6)$$

где је са NTU означен број јединица преноса а са C_r означен количник топлотног капацитета

$$NTU = \frac{UA}{C_{min}} \quad (8.7)$$

$$C_r = \frac{C_{min}}{C_{max}} \quad (8.8)$$

при чему је са U означен укупни коефицијент преноса топлоте а са A је означена површина измењивача топлоте док је C_{max} дат изразом

$$C_{max} = \max\{\dot{m}_1 c_{P1}, \dot{m}_2 c_{P2}\} \quad (8.9)$$

На основу свега претходног могу да се изведу изрази за температуре обе стране измењивача

$$T_{12} = T_{11} - \frac{Q}{\dot{m}_1 c_{P1}} \quad (8.10)$$

$$T_{22} = T_{21} - \frac{Q}{\dot{m}_2 c_{P2}} \quad (8.11)$$

Пошто ова метода представља статички модел и даје максималну могућу брзину размене топлоте, стварне излазне температуре за обе стране измењивача се рачунају на следећи начин

$$T_{12}(t) = T_{12}(t-1) - \left(T_{12}(t-1) - \left(T_{11}(t) - \frac{Q}{\dot{m}_1 c_{p1}} \right) \right) \frac{Q}{\tau \dot{m}_1 c_{p1}} \quad (8.12)$$

$$T_{22}(t) = T_{22}(t-1) - \left(T_{22}(t-1) - \left(T_{21}(t) - \frac{Q}{\dot{m}_2 c_{p2}} \right) \right) \frac{Q}{\tau \dot{m}_2 c_{p2}} \quad (8.13)$$

где је са τ означена временска константа која дефинише време смирења одзива измењивача топлоте [53].

У конкретном моделу је за τ усвојена вредност од пола сата, односно 1800 секунди. Укупни коефицијент преноса топлоте U се креће између 850 до 1700 W/m²K [107] док је у конкретном примеру усвојена вредност од 1000 W/m²K. За површину измењивача топлоте је усвојена вредност 10 m². Специфична топлота воде је 4.1813 J/kgK па је за c_{p1} и c_{p2} усвојена иста вредност.

На основу претходних израза имплементиран је модел измењивача топлоте у развојном окружењу LabView чији изворни код се може видети у прилогу III.1.

8.2.2 Модел трокраког вентила

Трокраки вентил служи за мешање два тока воде различитих температура. Има два улаза који се обично означавају са A и B и један излаз који се обично означава као AB (претходно означен као секундарни ток \dot{m}_2). На улаз A долази вода високе температуре T_{22} притиска p_A , а на улаз B долази вода ниске температуре T_{21} притиска p_B док се на излазу AB добија мешана вода температуре T .

Трокраки вентили обично су, у овим применама, опремљени актуаторима који могу да затварају и отварају вентил помоћу електро мотора. Оваквим актуаторима, за пуну промену отворености од потпуно затвореног до потпуно отвореног вентила, је потребно у зависности од величине и до неколико десетина секунди. У конкретном моделу ово је реализовано помоћу процеса првог реда са временском константом од 30 секунди кроз који се пропушта управљачки сигнал са контролера.

8.2.2.1 Модел вентила

Као модел вентила биће искоришћена експериментално утврђена карактеристика једног вентила описана у [53] а која је апроксимирана полиномом петог реда

$$q = 7,1587 \cdot 10^{-8} x^5 - 1,7898 \cdot 10^{-5} x^4 + 0,0013 \cdot x^3 - 0,0137 \cdot x^2 + 0,3166x + 0,0103 \quad (8.14)$$

где је x отвореност вентила а q проток кроз вентила за ту отвореност.

Масени протоци на улазима A и B су дати следећим изразима

$$\dot{m}_A = \frac{1}{\sqrt{q \left(\frac{p_A}{p_B} + \left(1 - \frac{p_A}{p_B} \right) q^2 \right)}} \dot{m}_1 \quad (8.15)$$

$$\dot{m}_B = \dot{m}_2 - \dot{m}_A \quad (8.16)$$

На основу претходних израза имплементиран је модел вентила у развојном окружењу LabView чији је изворни код дат у прилогу III.2.

8.2.2.2 Модел мешања

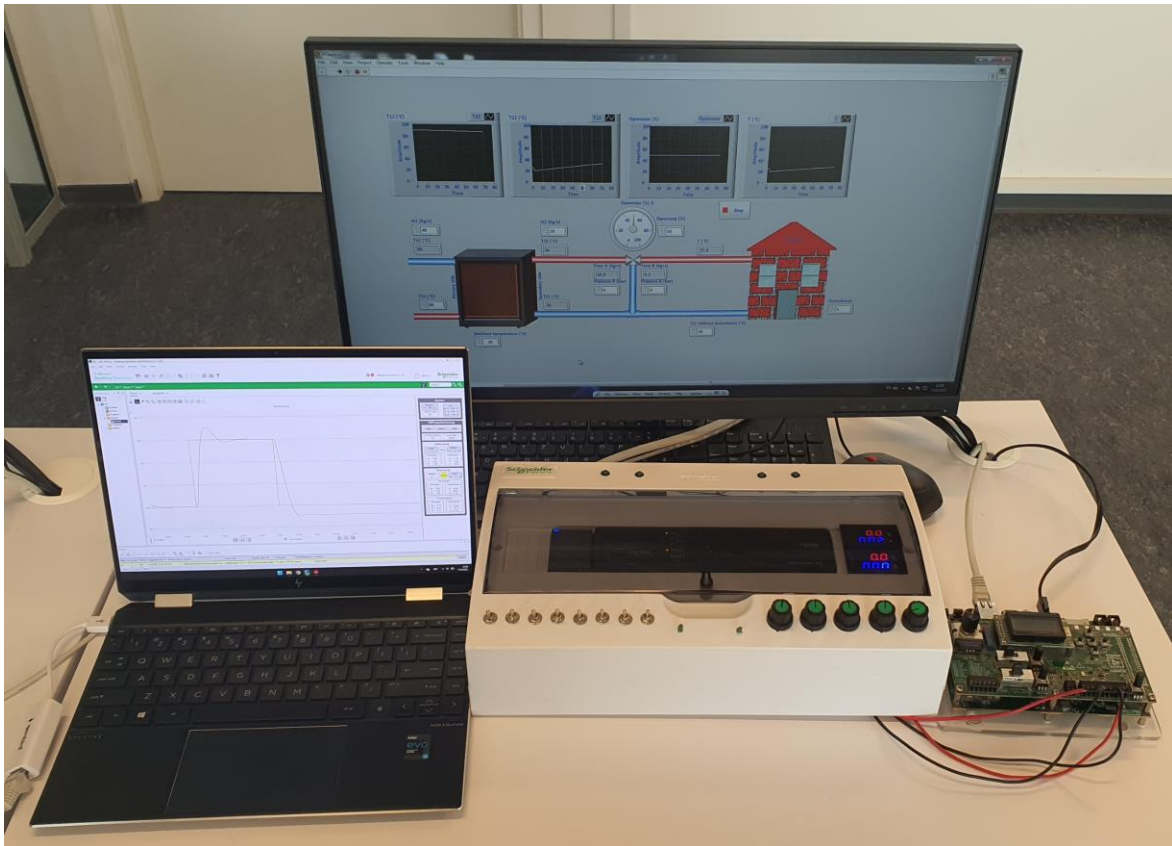
Са претходно развијеним моделом вентила температура мешане воде се може добити из статичног модела [53] описаног изразом

$$T = \frac{\dot{m}_A T_{22} + \dot{m}_B T_{21}}{\dot{m}_2} \quad (8.17)$$

На основу претходних израза имплементиран је модел мешања у развојном окружењу LabView чији изворни код се може видети у прилогу III.3.

8.3 Експериментална поставка

Како би се тестирала имплементација предложене метода на реалном контролеру реализована је експериментална поставка у HIL (Hardware-in-loop) аранжману приказана на слици 8.8. Поставка се састоји од контролера SE AS-B-24 на којем се извршава предложена метода и контролера NI sbRIO-9636 на којем се извршава модел топлотне подстаннице у реалном времену, као пример процеса из домена аутоматизације стамбено-пословних објеката.



Слика 8.8 Експериментална HIL поставка

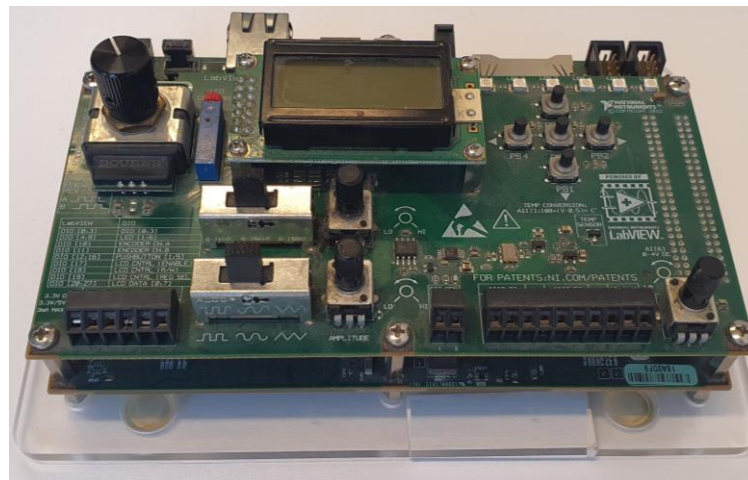
SE SmartX AS-B-24, приказан на слици 8.9, је компактни контролер компаније Schneider Electric за примену у аутоматизацији стамбених или пословних објеката мале до средње величине. Садржи флексибилну уграђену универзалну ИО конфигурацију као и уграђено напајање. Укупно поседује 24 порта од који је 16 универзалних улаза-излаза, 4 дигитална излаза и 4 дигитална улаза. Подржава многе комуникационе протоколе попут TCP/IP, BACNet, Modbus и MQTT што омогућава широку примену.

Конфигурација и програмирање се изводи из развојног окружења EcoStruxure Building Operation.



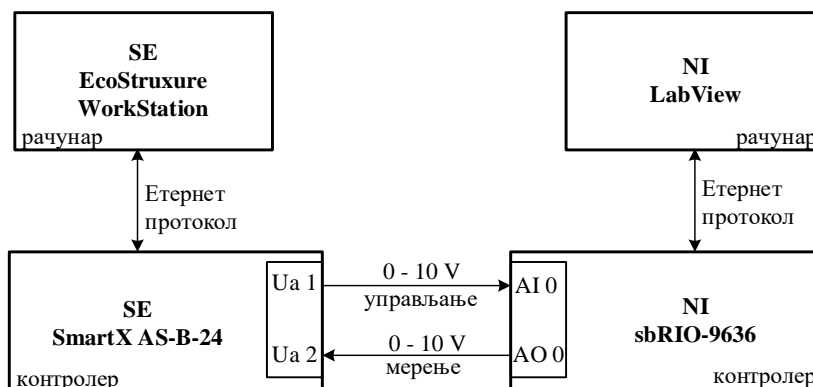
Слика 8.9 SE SmartX AS-B-24 контролер

NI sbRIO-9636, приказан на слици 8.10, је контролер компаније National Instruments који може да ради у реалном времену. Базиран је на FPGA технологији и намењен је за развој сложених алгоритама који захтевају извршавање у реалном времену. Поред предефинисаних тастера, потенциометара и LED индикација поседује 6 аналогних улаза и 2 аналогна излаза као и 4 дигитална улаза-излаза. Конфигурација и програмирање се изводи из развојног окружења LabView.



Слика 8.10 NI sbRIO-9636 контролер

Шематски приказ експерименталне поставке са назначеним комуникационим протоколима је приказан на слици 8.11. Контролери су међусобно повезани преко својих аналогних улазних-излазних терминала користећи стандардне напонске сигнале. NI контролер на свом аналогном излазу генерише напонски сигнал у опсегу од 0 V до 10 V сразмеран тренутној температури воде после мешног вентила, која је у опсегу од 20 °C до 90 °C. SE контролер на свом аналогном излазу генерише напонски сигнал у опсегу од 0 V до 10 V сразмеран управљачком сигналу са излаза ПИД регулатора који је у опсегу од 0 % до 100 %. Сваки од контролера је повезан путем Етернет (Ethernet) протокола са рачунаром на којем је инсталиран софтверски пакет за програмирање и конфигурацију као и за покретање графичког корисничког интерфејса за надзор и управљање.



Слика 8.11 Шематски приказ експерименталне HIL поставке

8.4 Резултати и анализа

На експерименталној поставци извршена је серија експеримената применом реализованих PIDAFuzzy и PIDPFuzzy блокова. Предложена метода је тестирана кроз сва три режима рада. Као почетни параметри су коришћени параметри добијени применом O-X методе као и параметри који дају изразито апериодичан и осцилаторан одзив.

Како би се могли поредити резултати добијени применом реализованих PIDAFuzzy и PIDPFuzzy блокова експерименти су изабрани тако да управљачки сигнал не улази у засићење имајући у виду недостатак *anti-windup* дејства код уграђеног PIDA блока који је саставни део PIDAFuzzy блока.

У конкретним експериментима, а и због саме HIL поставке, не очекују се сметње које би могле да утичу на рада PIDAFuzzy имајући у виду недостатак филтрирања диференцијалног дејства у уграђеном PIDA који је његов саставни део.

У току експерименталног тестирања температура воде развода примарног дела измењивача топлоте T_{11} је подешена на $90\text{ }^{\circ}\text{C}$. Температура повратне воде на секундарној страни T_{21} одговара температури околине просторије односно $20\text{ }^{\circ}\text{C}$. Повратна вода на секундарној страни измењивача топлоте може бити додатно поремећена помоћу потенциометра са NI контролера како би се симулирала варијација финалне потрошње топле воде од стране крајњих корисника у згради. Масени протоци воде на примарној и на секундарној страни измењивача топлоте \dot{m}_1 и \dot{m}_2 су постављени на 40 kg/s и 20 kg/s , респективно. Притисци воде у примарном и секундарном делу измењивача p_A и p_B су постављени на $1,5\text{ bar}$.

Добијени експериментални резултати су уз анализу приказани у наставку.

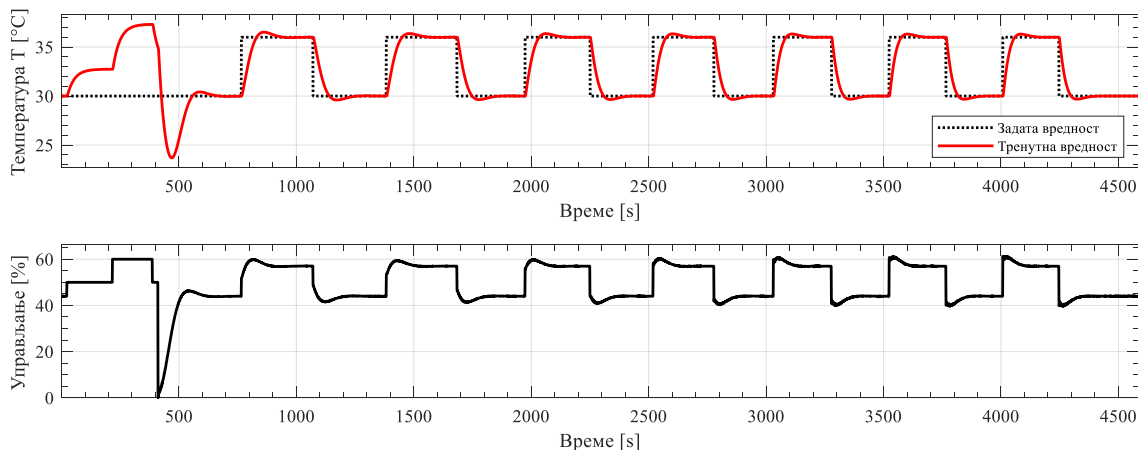
8.4.1 Аутоматско подешавање параметара са почетним подешавањем

Применом почетних подешавања добијају се резултати који су по питању брзине одзива и вредности прескока релативно добри. Стога ће у овом случају режим рада предложене методе бити постављен на режим Балансирање како би се покушало истовремено смањење времена успона и прескока. Демонстрирање рада је извршено кроз следећи пример.

8.4.1.1 Пример: Почетни параметри добијени применом O-X методе

Први експеримент је извршен коришћењем реализованог PIDAFuzzy блока у режиму рада Балансирање са укљученим почетним подешавањем и са изабраном O-X методом. Кретање задате и тренутне вредности температуре на излазу из трокраког

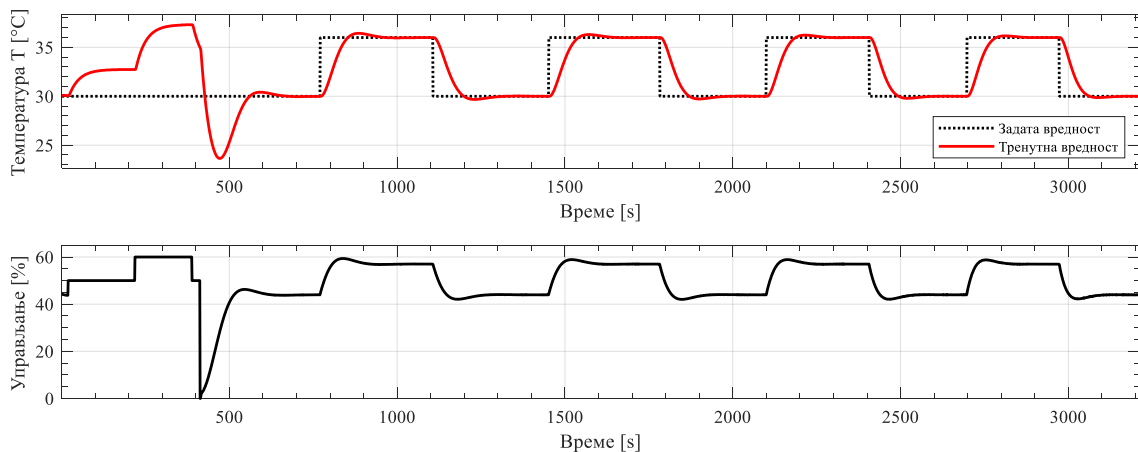
вентила T као и кретање вредности управљачког сигнала кроз итерације је приказано на слици 8.12.



Слика 8.12 Поступак аутоматског подешавања применом предложене методе у режиму рада Балансирање након почетног подешавања $O-X$ методом за PIDAFuzzy блок

Након покретања поступка ПИД регулатор се искључује и започиње се идентификација процеса у отвореној спрези. Управљање се поставља на 50% и након постизања устаљеног стања температура на излазу из трокраког вентила T достиже $32,72\text{ }^{\circ}\text{C}$ што резултира да статичко појачање има вредност 0,62. Управљање се затим подиже на 60% и након постизања устаљеног стања опет се враћа на 50%. Истовремено се на основу забележене температуре T прорачунавају површине A_0 и A_1 . На основу добијених површина врши се прорачун процењене временске константе која износи 13,27 s и процењеног временског кашњења које износи 10,72 s. У следећем кораку се на основу добијених параметара FOPDT модела прорачунавају почетни параметри ПИД регулатора који износе: $K_p=1,22$, $T_i=13,25$ и $T_d=4,3$. Са добијеним почетним параметрима започиње се процес аутоматског подешавања параметара. Након 13 итерација добијене су следеће вредности параметара: $K_p=2,76$, $T_i=21,11$ и $T_d=3,2$.

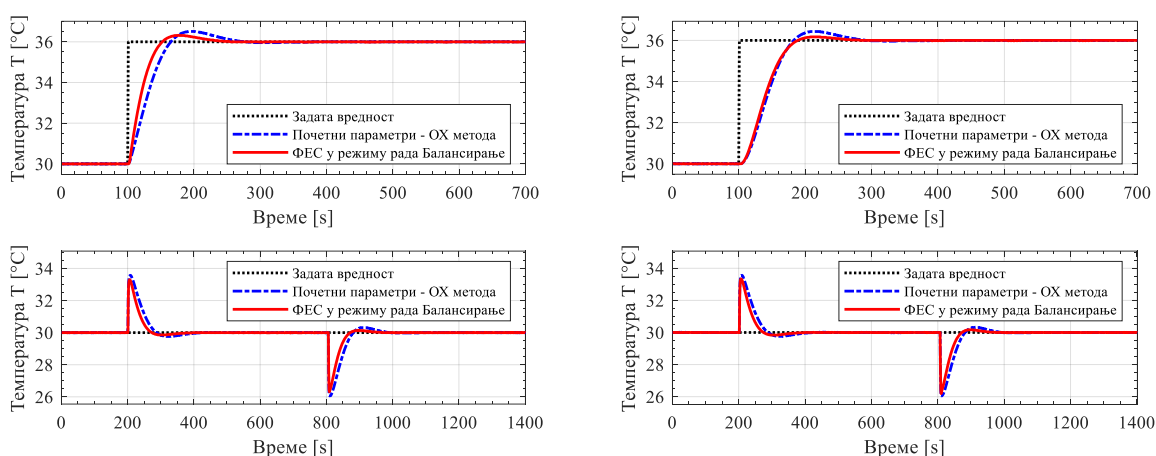
Други експеримент је извршен коришћењем реализованог PIDPFuzzy блока у режиму рада Балансирање са укљученим почетним подешавањем и са изабраном $O-X$ методом. Кретање задате и тренутне вредности температуре на излазу из трокраког вентила T као и кретање вредности управљачког сигнала кроз итерације је приказано на слици 8.13.



Слика 8.13 Поступак аутоматског подешавања применом предложене методе у режиму рада Балансирање након почетног подешавања $O-X$ методом за PIDPFuzzy блок

Параметри модела су остали непромењени па су у поступку почетног подешавања добијени исти почетни параметри ПИД регулатора као и у претходном случају. Са добијеним почетним параметрима поступак аутоматског подешавања параметара након 7 итерација даје следеће вредности параметара: $K_p=2,53$, $T_i=19,95$ и $T_d=3,27$.

На основу добијених параметара у првој и последњој итерацији извршени су додатни експерименти поређења одзива за оба ПИД блока на скоковиту промену задате вредности температуре које се може видети на слици 8.14. Такође извршени су и додатни експерименти поређења одзива на поремећај у виду скоковите промене температуре повратне воде T_{21} (тренутак $t=200$ s повећање за $5\text{ }^\circ\text{C}$; тренутак $t=800$ s смањење за $5\text{ }^\circ\text{C}$). Добијени одзиви се могу видети на слици 8.14. Са слике се може закључити да су за оба ПИД блока постигнута побољшања при одскочном одзиву на промену задате вредности као и на поремећај. С тим да треба нагласити да је у одзиву на одскочну побуду код PIDPFuzzy блока приметно мање смањење времена успона, што је и очекивано због саме структуре регулатора.



Слика 8.14 Одзив на скоковиту промену задате температуре мешања (горе) и на поремећај у виду скоковите промене температуре повратне воде (доле) добијен променом PIDAFuzzy блока (лево) и PIDPFuzzy блока (десно)

Како би се омогућило боље поређење резултата на основу сачуваних вредности у току трајања експеримента накнадно су израчунате мере перформанси које су приказане у табели 8.3. Из табеле се може видети да предложена метода испуњава претходно постављене циљеве за режим рада Балансирање без обзира на избор типа ПИД регулатора.

Табела 8.3 Перформансе ПИД регулатора постигнуте применом предложене методе у режиму рада Балансирање након почетног подешавања О-Х методом

		t_r [s]	σ [%]	t_s [s]	p_1	t_{sd_1} [s]	p_2	t_{sd_2} [s]
PIDAFuzzy	О-Х метода	45,87	8,50	152,50	33,58	186,84	26,05	166,10
	ФЕС у режиму рада Балансирање	34,54	5,33	119,50	33,30	146,40	26,31	135,64
	Унапређење [%]	-24,71	-37,25	-21,64	-0,83	-21,64	-1,00	-18,34
PIDPFuzzy	О-Х метода	53,83	7,17	167,00	33,57	187,86	26,04	168,08
	ФЕС у режиму рада Балансирање	53,33	2,83	138,50	33,34	162,17	26,19	142,40
	Унапређење [%]	-0,93	-60,47	-17,07	-0,69	-13,68	-0,58	-15,28

8.4.2 Аутоматско подешавање параметара без почетног подешавања

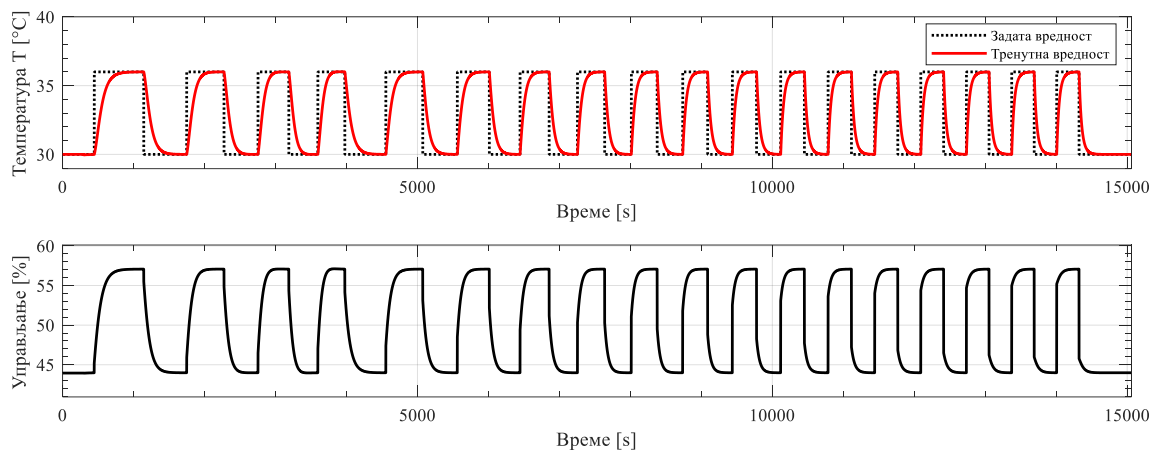
Како би се проверио рад предложене методе и без почетног подешавања параметара у наставку је размотрен пример када постојећи параметри дају изразито

апериодичан одскачни одзив са спором динамиком и пример када постојећи параметри дају осцилаторан одзив са брзом динамиком.

8.4.2.1 Пример: Почетни параметри са изразито апериодичним одзивом

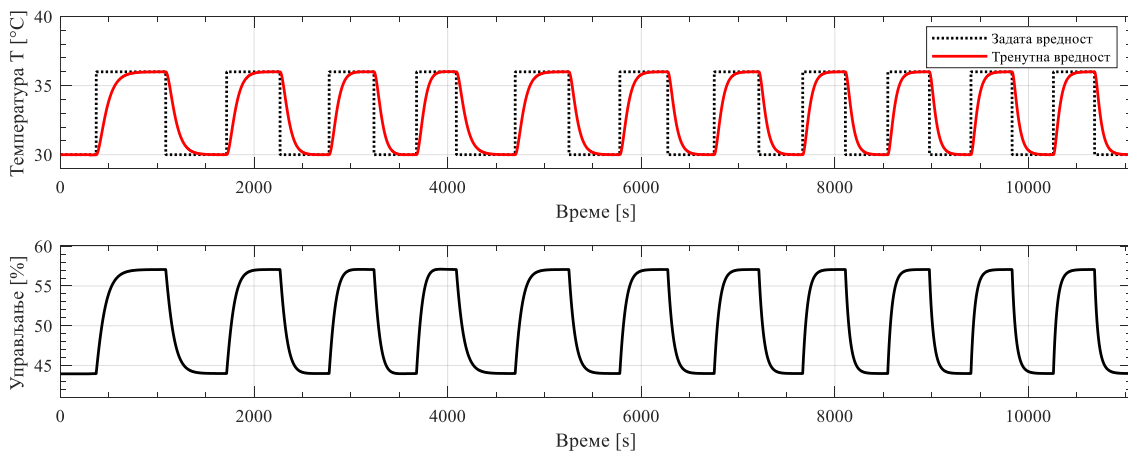
У овом примеру постојећи параметри ПИД регулатора $K_p=0,2$, $T_i=10$ и $T_d=0,1$ дају апериодичан одзив са веома спором динамиком тако да ће предложена метода бити примењена у режиму рада Убрзање.

Први експеримент је извршен коришћењем реализованог PIDAFuzzy блока у режиму рада Убрзање са искљученим почетним подешавањем. Кретање задате и тренутне вредности температуре на излазу из трокраког вентила T као и кретање вредности управљачког сигнала кроз итерације је приказано на слици 8.15. Након завршетка поступка аутоматског подешавања који је трајао 35 итерација добијени су следећи параметри: $K_p=1,86$, $T_i=32,27$ и $T_d=0,04$.



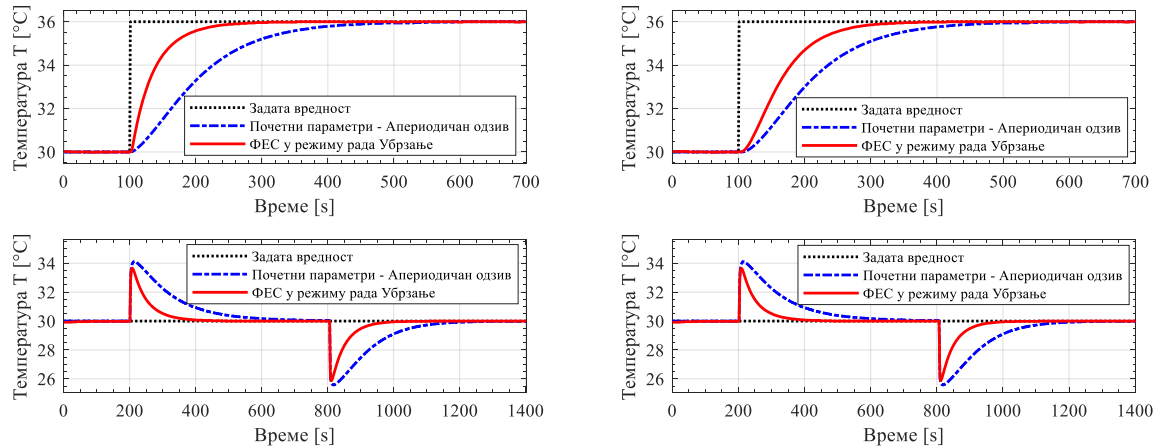
Слика 8.15 Поступак аутоматског подешавања кроз итерације применом предложене методе у режиму рада Убрзање за PIDAFuzzy блок

Други експеримент је извршен коришћењем реализованог PIDPFuzzy блока у режиму рада Убрзање са искљученим почетним подешавањем. Кретање задате и тренутне вредности температуре на излазу из трокраког вентила T као и кретање вредности управљачког сигнала кроз итерације је приказано на слици 8.16. Након завршетка поступка аутоматског подешавања који је трајао 21 итерацију, добијени су следећи параметри: $K_p=1,95$, $T_i=32,98$ и $T_d=0,04$.



Слика 8.16 Поступак аутоматског подешавања кроз итерације применом предложене методе у режиму рада Убрзање за PIDPFuzzy блок

На основу добијених параметара у првој и последњој итерацији извршени су додатни експерименти поређења одзива за оба ПИД блока на скоковиту промену задате вредности температуре које се може видети на слици 8.17. Такође извршени су и додатни експерименти поређења одзива на поремећај у виду скоковите промене температуре повратне воде T_{21} (тренутак $t=200$ s повећање за $5\text{ }^{\circ}\text{C}$; тренутак $t=800$ s смањење за $5\text{ }^{\circ}\text{C}$). Добијени одзиви се могу видети на слици 8.17. Са слике се може закључити да су оба ПИД блока постигла велико побољшања при одскочном одзиву на промену задате вредности као и на поремећај.



Слика 8.17 Одзив на скоковиту промену задате температуре мешања (горе) и на скоковиту промену температуре повратне воде у виду поремећаја (доле) добијен променом PIDAFuzzy блока (лево) и PIDPFuzzy блока (десно)

Како би се омогућило боље поређење резултата на основу сачуваних вредности у току трајања експеримента накнадно су израчунате мере перформанси које су приказане у табели 8.4. Из табеле се може видети да предложена метода испуњава претходно постављене циљеве за режим рада Убрзање без обзира на избор типа ПИД регулатора.

Табела 8.4 Перформансе ПИД регулатора постигнуте применом предложене методе у режиму рада Убрзање користећи постојеће параметре као почетне

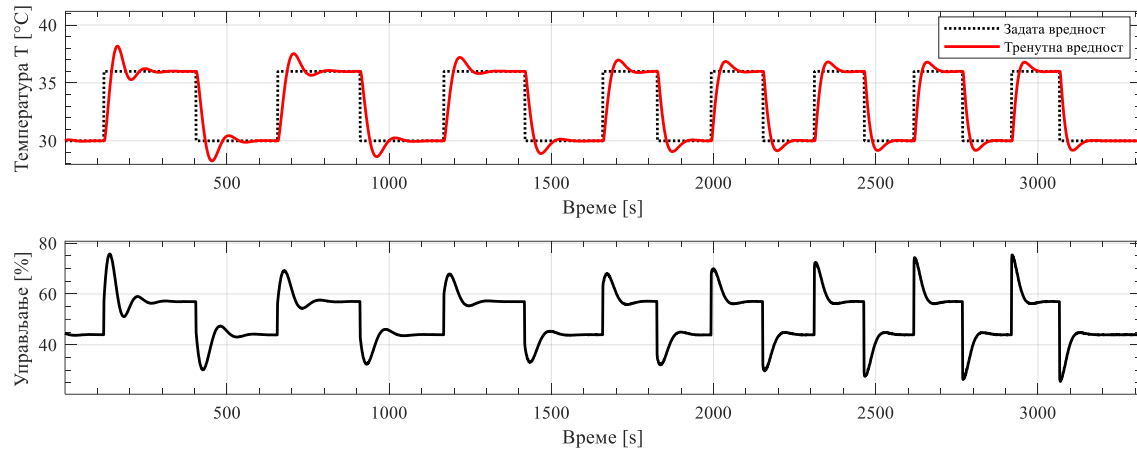
		t_r [s]	σ [%]	t_s [s]	p_1	t_{sd_1} [s]	p_2	t_{sd_2} [s]
PIDAFuzzy	Почетни параметри	196,33	0,00	339,00	34,12	357,10	25,58	337,66
	ФЕС у режиму рада Убрзање	79,50	0,00	149,00	33,66	198,68	25,87	177,76
	Унапређење [%]	-59,51	0,00	-56,05	-1,35	-44,36	-1,13	-47,36
PIDPFuzzy	Почетни параметри	197,00	0,00	348,00	34,12	357,10	25,58	338,32
	ФЕС у режиму рада Убрзање	116,67	0,00	206,00	33,65	196,72	25,88	178,76
	Унапређење [%]	-40,78	0,00	-40,80	-1,38	-44,91	-1,17	-47,16

8.4.2.2 Пример: Почетни параметри са изразито осцилаторним одзивом

У овом примеру постојећи параметри ПИД регулатора $K_p=2$, $T_i=5$ и $T_d=1$ дају осцилаторан одзив са веома брзом динамиком тако да ће предложена метода бити примењена у режиму рада Смањење прескока.

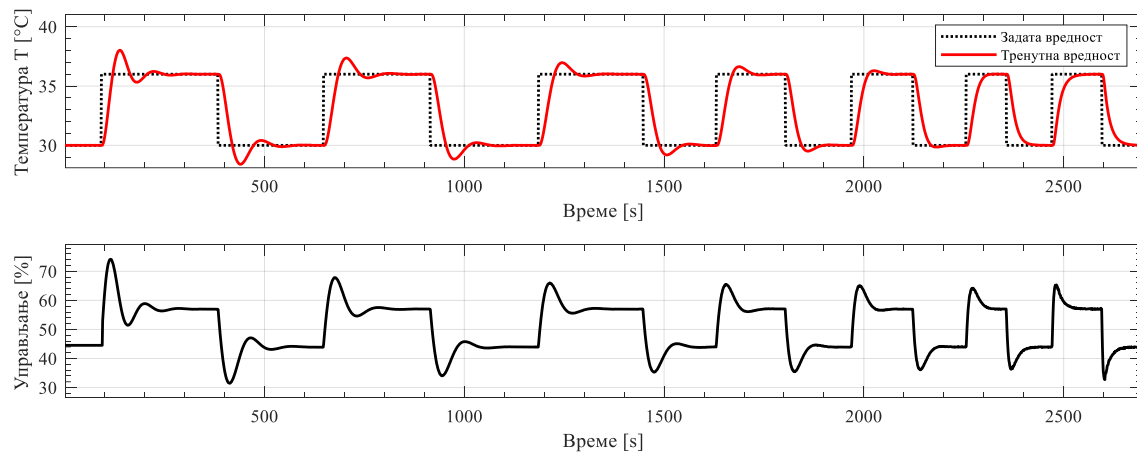
Први експеримент је извршен коришћењем реализованог PIDAFuzzy блока у режиму рада Смањење прескока са искљученим почетним подешавањем. Кретање задате и тренутне вредности температуре на излазу из трокраког вентила T као и кретање вредности управљачког сигнала кроз итерације је приказано на слици 8.18.

Након завршетка поступка аутоматског подешавања који је трајао 15 итерација добијени су следећи параметри: $K_p=4,84$, $T_i=15,01$ и $T_d=0,87$.



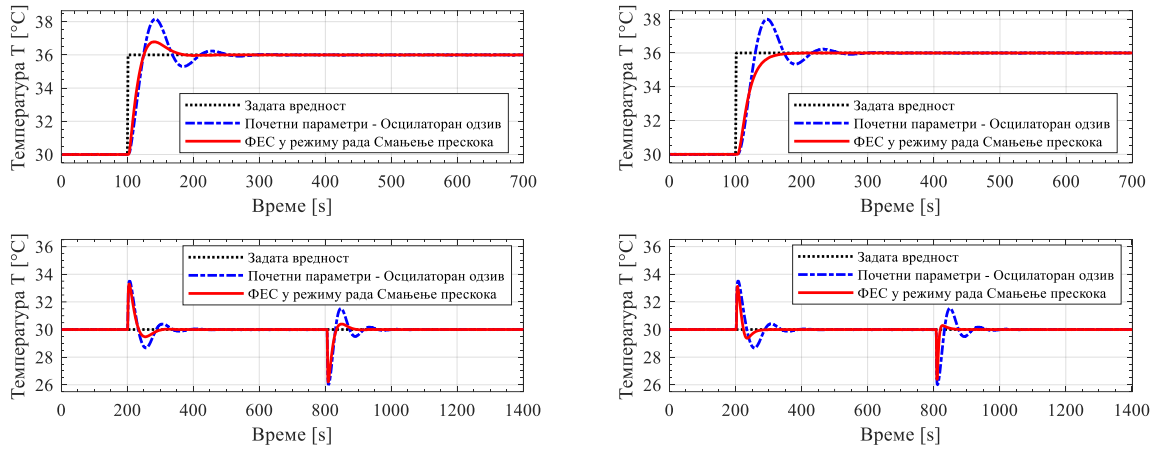
Слика 8.18 Поступак аутоматског подешавања параметара кроз итерације применом предложене методе у режиму рада Смањење прескока за PIDAFuzzy блок

Други експеримент је извршен коришћењем реализованог PIDPFuzzy блока у режиму рада Смањење прескока са искљученим почетним подешавањем. Кретање задате и тренутне вредности температуре на излазу из трокраког вентила T као и кретање вредности управљачког сигнала кроз итерације је приказано на слици 8.19. Након завршетка поступка аутоматског подешавања који је трајао 13 итерација добијени су следећи параметри: $K_p=14,48$, $T_i=17,43$ и $T_d=0,51$.



Слика 8.19 Поступак аутоматског подешавања параметара кроз итерације применом предложене методе у режиму рада Смањење прескока за PIDPFuzzy блок

На основу добијених параметара у првој и последњој итерацији извршени су додатни експерименти поређења одзива за оба ПИД блока на скоковиту промену задате вредности температуре које се може видети на слици 8.20. Такође извршени су и додатни експерименти поређења одзива на поремећај у виду скоковите промене температуре повратне воде T_{21} (тренутак $t=200$ s повећање за $5\text{ }^{\circ}\text{C}$; тренутак $t=800$ s смањење за $5\text{ }^{\circ}\text{C}$). Добијени одзиви су приказани на слици 8.20 са које се може закључити да је предложена метода за оба ПИД блока постигла велико побољшања при одскочном одзиву на промену задате вредности као и на поремећај.



Слика 8.20 Одзив на скоковиту промену задате температуре мешања (горе) и на скоковиту промену температуре повратне воде у виду поремећаја (доле) добијен променом PIDAFuzzy блока (лево) и PIDPFuzzy блока (десно)

Како би се омогућило боље поређење резултата на основу сачуваних вредности у току трајања експеримента накнадно су израчунате мере перформанси које су приказане у табели 8.5. Из табеле се може видети да предложена метода испуњава претходно постављене циљеве за режим рада Смањење прескока без обзира на избор типа ПИД регулатора. С тим да треба нагласити да је у одзиву на одскочну побуду код PIDPFuzzy блока приметно повећање вредности времена успона што је и очекивано због саме структуре регулатора.

Табела 8.5 Перформансе ПИД регулатора постигнуте применом предложене методе у режиму рада Смањење прескока користећи постојеће параметре као почетне

		t_r [s]	σ [%]	t_s [s]	p_1	t_{sd_1} [s]	p_2	t_{sd_2} [s]
PIDAFuzzy	Почетни параметри	16,41	36,00	142,00	33,52	170,46	26,00	149,50
	ФЕС у режиму рада Смањење прескока	16,41	13,17	74,50	33,29	94,40	26,20	81,40
	Унапређење [%]	0,00	-63,43	-47,54	-0,69	-44,62	-0,77	-45,55
PIDPFuzzy	Почетни параметри	18,24	33,33	147,00	33,52	176,48	26,00	151,50
	ФЕС у режиму рада Смањење прескока	32,59	0,00	63,00	33,13	71,74	26,39	51,78
	Унапређење [%]	78,69	-100,00	-57,14	-1,16	-59,35	-1,50	-65,82

9 Закључак

Основна тема докторске дисертације је унапређење поступка аутоматског подешавања параметара ПИД регулатора са акцентом на примени у аутоматизацији стамбено-пословних објеката. Представљени су резултати оригиналног поступка заснованог на рачунарској интелигенцији у виду фази експертског система који су верификовани у међународном часопису [102].

Предложена метода користи снагу концепта фази логике за побољшање процеса подешавања параметара ПИД регулатора. Овај приступ имплементира искуство инжењера у фази експертски систем који води процедуру подешавања и пружа побољшање перформанси у односу на ПИД регулатор подешен коришћењем класичне методе подешавања. Предложена шема ажурира параметре ПИД регулатора кроз итерације на основу карактеристика одскочног одзива. Процес подешавања је вођен експертским системом који анализира процесне податке много брже од човека, што резултира значајним убрзањем процеса подешавања параметара.

Метода је имплементирана у симулационом софтверу и затим тестирана кроз рачунарске симулације на широком скупу различитих процеса. Опсежан скуп процеса коришћен је да се истакне моћ генерализације предложеног решења. Као референтна тачка за поређење добијених резултата коришћене су Ц-Н и О-Х метода како би се предложене методе тестирале на различитим почетним вредностима параметара. О-Х за разлику од Ц-Н методе већ даје добре резултате за све посматране процесе, што оставља мање простора за побољшање. Међутим, резултати су показали да предложена метода може да унапреди перформансе и да у исто време задржи робусност у разумним границама без обзира на начин добијања почетних параметара. Режији рада подижу употребљивост предложене шеме на виши ниво омогућавајући избор приоритета за циљеве процедуре подешавања. Резултати су показали да су сва три режима рада постигла побољшање перформанси пратећи почетно постављене приоритете (убрзавање, смањење прескока или балансирање) у релативно кратком времену са малом деградацијом робусности.

Метода је имплементирана на реалном контролеру који се користи у аутоматизацији стамбено-пословних објеката коришћењем FBD програмског језика. Експериментална верификација је извршена на HIL поставци у оквиру које је реализован модел топлотне подстаннице на посебном контролеру који омогућава извршавање у реалном времену. Предложена метода је примењена са укљученим почетним подешавањем као и са почетним параметрима који дају изразито апериодичан и осцилаторан одзив и то за оба реализована блока. Добијени резултати су показали да предложена метода може да унапреди перформансе без обзира на начин добијања почетних параметара и избора ПИД блока а у складу са изабраним режимом рада. Предложена метода примењена на ПИД регулатор реализован у PIDP блоку остварила је слабије перформансе приликом убрзавања динамике система односно смањења времена успона у односу на PIDA блок што је и очекивано због саме реализације закона управљања са два степена слободе.

Могуће је идентификовати неколико праваца у даљем истраживању. Први би био унапређење начина рада експертског система тако да се корекције параметара ПИД

регулатора у свакој итерацији добијају на основу перформанси које би се добиле из одскочног одзива на поремећај. То би направило простор за побољшање перформанси предложене методе у регулационим круговима где се употребљава модификована верзија ПИД регулатора са два степена слободe. Други правац би се огледао у имплементацији постојеће методе на другим контролерима који нису стриктно везани за аутоматизацију стамбено-пословних објеката. Трећи правац би био анализа могућности рада методе тако да се поступак аутоматског подешавања извршава уз нормално функционисање система. Последњи правац би био анализа могућности коришћења прикупљених података у току итеративног поступка аутоматског подешавања параметара. Систематско коришћење аутоматског подешавања укључује много експеримената на процесу омогућујући прикупљање информација које представљају стање радних услова процеса. Поред подешавања регулатора, добија се обиље информација о процесу, чиме се надограђује знање о процесу којим се управља. Складиштењем ових података у вишој инстанци надзорно-управљачког система, могуће је извршити накнадну анализу.

Литература

- [1] J. G. Ziegler and N. B. Nichols, "Optimum Settings For Automatic Controllers," *Transactions of the American Society of Mechanical Engineers*, vol. 64, pp. 759–768, 1942.
- [2] J. G. Ziegler and N. B. Nichols, "Process lags in automatic control circuits," *Transactions of the American Society of Mechanical Engineers*, vol. 65, pp. 433–444, 1943.
- [3] K. L. Chien, J. A. Hrones, and J. B. Reswick, "On the automatic control of generalised passive systems," *Transactions of the American Society of Mechanical Engineers*, vol. 74, pp. 175–185, 1952.
- [4] G. H. Cohen and G. A. Coon, "Theoretical considerations of retarded control," *Transactions of the American Society of Mechanical Engineers*, vol. 75, no. 1, pp. 827–834, 1953.
- [5] T. Hägglund and K. J. Åström, "Revisiting The Ziegler-Nichols Tuning Rules For PI Control," *Asian J Control*, vol. 4, no. 4, pp. 364–380, 2002, doi: 10.1111/j.1934-6093.2002.tb00076.x.
- [6] K. J. Åström and T. Hägglund, "Revisiting the Ziegler–Nichols step response method for PID control," *J Process Control*, vol. 14, no. 6, pp. 635–650, 2004, doi: <http://dx.doi.org/10.1016/j.jprocont.2004.01.002>.
- [7] T. B. Šekara and M. R. Mataušek, "Revisiting the Ziegler–Nichols process dynamics characterization," *J Process Control*, vol. 20, no. 3, pp. 360–363, Mar. 2010, doi: 10.1016/j.jprocont.2009.08.004.
- [8] K. J. Åström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC, USA: ISA - Instrumentation, Systems, and Automation Society, 2006.
- [9] A. O'Dwyer, *Handbook of PI and PID controller tuning rules*, 3rd ed. London, UK: Imperial College Press, 2009.
- [10] K. J. Åström and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, vol. 20, no. 5, pp. 645–651, 1984, doi: 10.1016/0005-1098(84)90014-1.
- [11] H.-P. Huang and C.-H. Lin, "A Stable on-Line Self-Tuning Optimal PID Controller for a Class of Unknown Systems," *Asian J Control*, vol. 9, no. 2, pp. 151–162, 2008, doi: 10.1111/j.1934-6093.2007.tb00318.x.
- [12] C. Dey and R. K. Mudi, "An improved auto-tuning scheme for PID controllers," *ISA Trans*, vol. 48, no. 4, pp. 396–409, 2009, doi: 10.1016/j.isatra.2009.07.002.
- [13] J. A. Romero, R. Sanchis, and P. Balaguer, "PI and PID auto-tuning procedure based on simplified single parameter optimization," *J Process Control*, vol. 21, no. 6, pp. 840–851, Jul. 2011, doi: 10.1016/j.jprocont.2011.04.003.
- [14] D. Pavković, S. Polak, and D. Zorc, "PID controller auto-tuning based on process step response and damping optimum criterion," *ISA Trans*, vol. 53, no. 1, pp. 85–96, 2014, doi: 10.1016/j.isatra.2013.08.011.
- [15] D. Matić, I. Kamenko, P. Nikolić, and V. Bugarski, "Optimal control of fruit and vegetables drying process," *Journal on Processing and Energy in Agriculture*, vol. 18, no. 3, 2014.
- [16] M. Cetin and S. Iplikci, "A novel auto-tuning PID control mechanism for nonlinear systems," *ISA Trans*, vol. 58, pp. 292–308, 2015, doi: 10.1016/j.isatra.2015.05.017.

- [17] R. Kumar, S. K. Singla, and V. Chopra, "Comparison among some well known control schemes with different tuning methods," *Journal of Applied Research and Technology*, vol. 13, no. 3, pp. 409–415, 2015, doi: 10.1016/j.jart.2015.07.007.
- [18] J. Berner, K. Soltesz, T. Hägglund, and K. J. Åström, "An experimental comparison of PID autotuners," *Control Eng Pract*, vol. 73, pp. 124–133, 2018, doi: 10.1016/j.conengprac.2018.01.006.
- [19] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965, doi: [http://dx.doi.org/10.1016/S0019-9958\(65\)90241-X](http://dx.doi.org/10.1016/S0019-9958(65)90241-X).
- [20] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proceedings of the Institution of Electrical Engineers*, vol. 121, no. 12, pp. 1585–1588, 1974, doi: 10.1049/piee.1974.0328.
- [21] V. Čongradac, M. Prica, M. Paspalj, D. Bojanić, and D. Čapko, "Algorithm for blinds control based on the optimization of blind tilt angle using a genetic algorithm and fuzzy logic," *Solar Energy*, vol. 86, no. 9, pp. 2762–2770, 2012, doi: 10.1016/j.solener.2012.06.016.
- [22] R. Z. Homod, K. S. M. Sahari, H. A. F. Almurib, and F. H. Nagi, "Gradient auto-tuned Takagi–Sugeno Fuzzy Forward control of a HVAC system using predicted mean vote index," *Energy Build*, vol. 49, pp. 254–267, Jun. 2012, doi: 10.1016/j.enbuild.2012.02.013.
- [23] S. Soyguder, M. Karakose, and H. Alli, "Design and simulation of self-tuning PID-type fuzzy adaptive control for an expert HVAC system," *Expert Syst Appl*, vol. 36, no. 3, pp. 4566–4573, 2009, doi: 10.1016/j.eswa.2008.05.031.
- [24] V. Bugarski, D. Matić, P. Nikolić, and I. Kamenko, "Controlling ship lock with fuzzy expert system," *Journal on Processing and Energy in Agriculture*, vol. 16, no. 4, pp. 162–164, 2012.
- [25] V. Bugarski, T. Bačkalić, and U. Kuzmanov, "Fuzzy decision support system for ship lock control," *Expert Syst Appl*, vol. 40, no. 10, pp. 3953–3960, Aug. 2013, doi: 10.1016/J.ESWA.2012.12.101.
- [26] T. Bačkalić, V. Bugarski, F. Kulić, and Ž. Kanović, "Adaptable Fuzzy Expert System for Ship Lock Control Support," *Journal of Navigation*, vol. 69, no. 6, pp. 1341–1356, 2016, doi: 10.1017/S0373463316000242.
- [27] F. Kulić, D. Matić, B. Dumnić, and V. Vasić, "Optimal fuzzy controller tuned by TV-PSO for induction motor speed control," *Advances in Electrical and Computer Engineering*, vol. 11, no. 1, pp. 49–54, 2011, doi: 10.4316/aece.2011.01008.
- [28] D. Kukulj, S. Kuzmanovic, E. Levi, and F. Kulic, "Design of a near-optimal, wide-range fuzzy logic controller," *Fuzzy Sets Syst*, vol. 120, no. 1, pp. 17–34, 2001, doi: 10.1016/S0165-0114(99)00007-X.
- [29] A. Zdešar, O. Cerman, D. Dovžan, P. Hušek, and I. Škrjanc, "Fuzzy control of a helio-crane: Comparison of two control approaches," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 72, no. 3–4, pp. 497–515, 2013, doi: 10.1007/s10846-012-9796-0.
- [30] Y. Gao, F. Xiao, J. Liu, and R. Wang, "Distributed Soft Fault Detection for Interval Type-2 Fuzzy-Model-Based Stochastic Systems With Wireless Sensor Networks," *IEEE Trans Industr Inform*, vol. 15, no. 1, pp. 334–347, Jan. 2019, doi: 10.1109/TII.2018.2812771.
- [31] S. Devi Vidhya and M. Balaji, "Hybrid fuzzy PI controlled multi-input DC/DC converter for electric vehicle application," *Automatika*, vol. 61, no. 1, pp. 79–91, 2020, doi: 10.1080/00051144.2019.1684038.

- [32] P. Du, Y. Pan, H. Li, and H. Lam, "Nonsingular Finite-Time Event-Triggered Fuzzy Control for Large-Scale Nonlinear Systems," *IEEE Transactions on Fuzzy Systems*, p. 1, 2020, doi: 10.1109/TFUZZ.2020.2992632.
- [33] H. Farsizadeh, M. Gheisarnejad, M. Mosayebi, M. Rafiei, and M. H. Khooban, "An Intelligent and Fast Controller for DC/DC Converter Feeding CPL in a DC Microgrid," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 6, pp. 1104–1108, 2020, doi: 10.1109/TCSII.2019.2928814.
- [34] M. Si Fodil, P. Siarry, F. Guély, and J. L. Tyran, "A fuzzy rule base for the improved control of a pressurized water nuclear reactor," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 1, pp. 1–10, 2000, doi: 10.1109/91.824761.
- [35] A. T. Nguyen, T. Taniguchi, L. Eciolaza, V. Campos, R. Palhares, and M. Sugeno, "Fuzzy control systems: Past, present and future," *IEEE Comput Intell Mag*, vol. 14, no. 1, pp. 56–68, 2019, doi: 10.1109/MCI.2018.2881644.
- [36] S. Tzafestas and N. P. Papanikolopoulos, "Incremental Fuzzy Expert PID Control," *IEEE Transactions on Industrial Electronics*, vol. 37, no. 5, pp. 365–371, 1990, doi: 10.1109/41.103431.
- [37] H. Jigang, W. Jie, and F. Hui, "An anti-windup self-tuning fuzzy pid controller for speed control of brushless dc motor," *Automatika*, vol. 58, no. 3, pp. 321–335, 2017, doi: 10.1080/00051144.2018.1423724.
- [38] M. Esfandyari, M. A. Fanaei, and H. Zohreie, "Adaptive fuzzy tuning of PID controllers," *Neural Comput Appl*, vol. 23, no. SUPPL1, pp. 19–28, 2013, doi: 10.1007/s00521-012-1215-8.
- [39] S. Z. He, S. Tan, F. L. Xu, and P. Z. Wang, "Fuzzy self-tuning of PID controllers," *Fuzzy Sets Syst*, vol. 56, no. 1, pp. 37–46, 1993, doi: 10.1016/0165-0114(93)90183-I.
- [40] H.-P. Hong *et al.*, "A design of auto-tuning PID controller using fuzzy logic," in *International Conference on Industrial Electronics, Control, Instrumentation, and Automation*, 1992, no. 13 Novemba, pp. 971–976. doi: 10.1109/IE-CON.1992.254452.
- [41] Z. Y. Zhao, M. Tomizuka, and S. Isaka, "Fuzzy Gain Scheduling of PID Controllers," *IEEE Trans Syst Man Cybern*, vol. 23, no. 5, pp. 1392–1398, 1993, doi: 10.1109/21.260670.
- [42] M. Petrov, T. Proychev, and A. Topalov, "Expert PID Controller with Fuzzy Self-Tuning," *IFAC Proceedings Volumes*, vol. 28, no. 5, pp. 367–372, 1995, doi: 10.1016/S1474-6670(17)47254-5.
- [43] O. da M. Almeida and A. A. R. Coelho, "A fuzzy logic method for autotuning a PID controller: SISO and MIMO systems," in *15th Triennial World Congress*, 2002, pp. 1–6.
- [44] Ł. Niewiara and K. Zawirski, "Auto-tuning of PID controller based on fuzzy logic," *Computer Applications in Electrical Engineering*, vol. 11, pp. 230–240, 2013.
- [45] Y. Wang, Q. Jin, and R. Zhang, "Improved fuzzy PID controller design using predictive functional control structure," *ISA Trans*, vol. 71, pp. 354–363, 2017, doi: 10.1016/j.isatra.2017.09.005.
- [46] R. Sivalingam, S. Chinnamuthu, and S. S. Dash, "A modified whale optimization algorithm-based adaptive fuzzy logic PID controller for load frequency control of autonomous power generation systems," *Automatika*, vol. 58, no. 4, pp. 410–421, 2017, doi: 10.1080/00051144.2018.1465688.
- [47] M. H. Khooban, "Hardware-in-the-loop simulation for the analyzing of smart speed control in highly nonlinear hybrid electric vehicle," *Transactions of the Institute of*

- Measurement and Control*, vol. 41, no. 2, pp. 458–467, 2019, doi: 10.1177/0142331218764784.
- [48] M. Gheisarnejad and M. H. Khooban, “Design an optimal fuzzy fractional proportional integral derivative controller with derivative filter for load frequency control in power systems,” *Transactions of the Institute of Measurement and Control*, vol. 41, no. 9, pp. 2563–2581, 2019, doi: 10.1177/0142331218804309.
- [49] F. Al-Sunni and T. Al-Nemer, “A Fuzzy Logic Based PID Self-Tuning Scheme For Process Control,” *IFAC Proceedings Volumes*, vol. 30, no. 6, pp. 737–743, 1997, doi: 10.1016/S1474-6670(17)43453-7.
- [50] A. Visioli, “Comparison of fuzzy logic based tuning methods for PID controllers,” in *European Control Conference, ECC 1999 - Conference Proceedings*, 1999, pp. 497–502. doi: 10.23919/ecc.1999.7099353.
- [51] C. Underwood and F. Yik, *Modelling Methods for Energy in Buildings*. Blackwell Science Ltd, 2008.
- [52] C. P. Underwood, *HVAC Control Systems: Modelling, Analysis and Design*. Taylor & Francis, 1999.
- [53] S. Lazarević, V. Čongradac, A. S. Anđelković, D. Čapko, and Ž. Kanović, “A novel approach to real-time modelling of the district heating substation system using LabVIEW,” *J Clean Prod*, vol. 217, pp. 360–370, Apr. 2019, doi: 10.1016/J.JCLEPRO.2019.01.279.
- [54] F. Mihalič, M. Truntič, and A. Hren, “Hardware-in-the-Loop Simulations: A Historical Overview of Engineering Challenges,” *Electronics (Switzerland)*, vol. 11, no. 15. MDPI, Aug. 01, 2022. doi: 10.3390/electronics11152462.
- [55] N. Brayanov and A. Stoyanova, “Review of hardware-in-the-loop-a hundred years progress in the pseudo-real testing,” *Electrotechnica & Electronica (E+E)*, vol. 54, no. 3–4, pp. 77–84, 2019.
- [56] M. Bacic, “On hardware-in-the-loop simulation,” in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*, 2005, vol. 2005, pp. 3194–3198. doi: 10.1109/CDC.2005.1582653.
- [57] I. M. Horowitz, *Synthesis of feedback systems*. New York, USA: Academic Press, 1963.
- [58] K. J. Åström and B. Witténmark, *Computer-Controlled Systems - Theory and Design*, 3rd ed. Prentice Hall, 1996.
- [59] Milić Stojić, *Digitalni upravljački sistemi*. Beograd: JP Nauka, 1994.
- [60] O. Mayr, *Feedback mechanisms*. Washington, DC, USA: Smithsonian Institution Press, 1971. doi: 10.1016/b978-0-12-200750-7.50011-2.
- [61] J. C. Maxwell, “On governors,” *Proceedings of the Royal Society of London*, vol. 16, pp. 270–283, Dec. 1868, doi: 10.1098/rspl.1867.0055.
- [62] N. Minorsky, “Directional stability of automatically steered bodies,” *Journal of the American Society of Naval Engineering*, vol. 34, no. 2, pp. 280–309, 1922, doi: doi.org/10.1111/j.1559-3584.1922.tb04958.x.
- [63] S. Bennett, “Development of the PID Controller,” *IEEE Control Syst*, vol. 13, no. 6, pp. 58–62, 1993, doi: 10.1109/37.248006.
- [64] S. Bennett, “The Past of PID Controllers,” *IFAC Proceedings Volumes*, vol. 33, no. 4, pp. 1–11, Apr. 2000, doi: 10.1016/S1474-6670(17)38214-9.
- [65] A. O’Dwyer, “PID control: the early years,” *Control in the IT Sector Seminar – May 3, 2005*, pp. 1–12, 2005.
- [66] H.-J. Zimmermann, *Fuzzy set theory and its applications*, 4th ed. New York, NY, USA: Springer Science+Business Media, 2001.

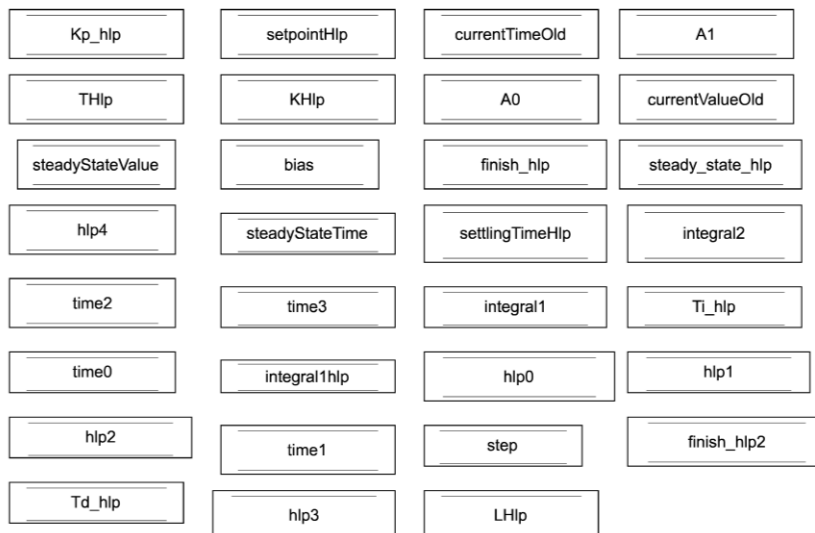
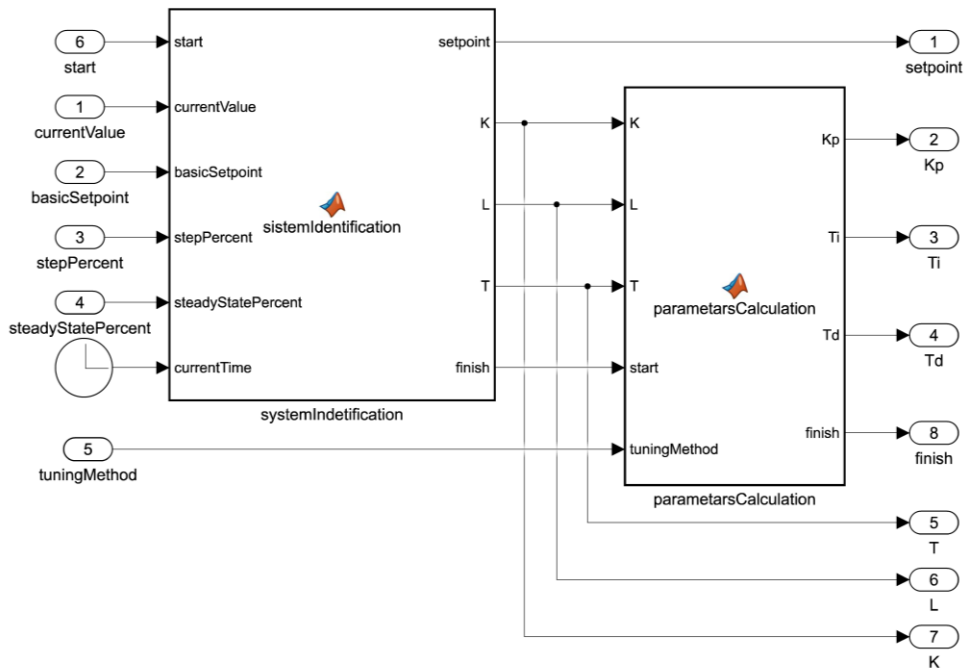
- [67] K. M. Passino and S. Yurkovich, *Fuzzy control*. Menlo Park, CA, USA: Addison Wesley Longman, Inc. All, 2009.
- [68] L. Reznik, *Fuzzy Controllers*. Newton, MA, USA: Butterworth-Heinemann, 1997.
- [69] D. S. Hooda and V. Raich, *Fuzzy Logic Models and Fuzzy Control*, vol. 53, no. 9. 2017. doi: 10.1017/CBO9781107415324.004.
- [70] Y. Song, S. Wu, and Y. Yan, “Development of Self-Tuning Intelligent PID Controller Based,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 11, 2013.
- [71] J. Jantzen, *Foundations of Fuzzy Control*. Chichester, UK: John Wiley & Sons, 2007. doi: 10.1002/9780470061176.
- [72] George J. Klir and B. Yuan, *Fuzzy sets and fuzzy logic, theory and applications*. Upper Saddle River, NJ, USA: Prentice Hall, 1995.
- [73] T. Terano, K. Asai, and M. Sugeno, *Applied fuzzy systems*. AP Professional, 1994.
- [74] L. A. Zadeh, “Fuzzy Logic,” *Computer (Long Beach Calif)*, vol. 21, no. 4, pp. 83–93, 1988, doi: 10.1109/2.53.
- [75] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. London, UK: Prentice Hall, 1997.
- [76] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning—I,” *Inf Sci (N Y)*, vol. 8, no. 3, pp. 199–249, 1975, doi: 10.1016/0020-0255(75)90036-5.
- [77] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning—II,” *Inf Sci (N Y)*, vol. 8, no. 4, pp. 301–357, 1975, doi: 10.1016/0020-0255(75)90046-8.
- [78] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning—III,” *Inf Sci (N Y)*, vol. 9, no. 1, pp. 43–80, 1975, doi: 10.1016/0020-0255(75)90017-1.
- [79] E. H. Mamdani, “Application of fuzzy logic to approximate reasoning using linguistic synthesis,” *IEEE Transactions on Computers*, vol. C-26, no. 12, pp. 1182–1191, 1977, doi: 10.1109/TC.1977.1674779.
- [80] E. H. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *Int J Man Mach Stud*, vol. 7, no. 1, pp. 1–13, Jan. 1975, doi: 10.1016/S0020-7373(75)80002-2.
- [81] J. H. Lilly, *Fuzzy Control and Identification*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2010.
- [82] G. Feng, *Analysis and Synthesis of Fuzzy Control Systems: A Model-Based Approach*. CRC Press, 2010. doi: 10.1201/EBK1420092646.
- [83] J. Jantzen, *Design Of Fuzzy Controllers*. 1998.
- [84] T. Takagi and M. Sugeno, “Fuzzy Identification of Systems and Its Applications to Modeling and Control,” *IEEE Trans Syst Man Cybern*, vol. SMC-15, no. 1, pp. 116–132, 1985, doi: 10.1109/TSMC.1985.6313399.
- [85] M. Sugeno and G. T. Kang, “Structure identification of fuzzy model,” *Fuzzy Sets Syst*, vol. 28, no. 1, pp. 15–33, Oct. 1988, doi: 10.1016/0165-0114(88)90113-3.
- [86] S. N. Sivanandam, S. Sumathi, and S. N. Deepa, *Introduction to fuzzy logic using MATLAB*. 2007. doi: 10.1007/978-3-540-35781-0.
- [87] A. Leva, C. Cox, and A. E. Ruano, *Hands-on PID autotuning: a guide to better utilisation*. IFAC, 2002.
- [88] K. J. Åström and B. Wittenmark, *Adaptive Control*, 2nd ed. Mineola, NY, USA: Dover Publications, Inc, 2008.
- [89] V. Bobál, J. Böhm, J. Fessl, and J. Macháček, *Digital Self-tuning Controllers*. Springer, 2005.

- [90] R. Kalman, "Design of a Self-Optimizing Control System," *Transactions of the American Society of Mechanical Engineers*, vol. 80, no. 2, pp. 468–477, Jan. 1958, doi: 10.1115/1.4012407.
- [91] K. J. Åström and B. Wittenmark, "On self tuning regulators," *Automatica*, vol. 9, no. 2, pp. 185–199, Mar. 1973, doi: 10.1016/0005-1098(73)90073-3.
- [92] K. J. Åström, H. Panagopoulos, and T. Häggglund, "Design of PI Controllers based on Non-Convex Optimization," *Automatica*, vol. 34, no. 5, pp. 585–601, 1998, doi: 10.1016/S0005-1098(98)00011-9.
- [93] T. Liu, Q. G. Wang, and H. P. Huang, "A tutorial review on process identification from step or relay feedback test," *J Process Control*, vol. 23, no. 10, pp. 1597–1623, Nov. 2013, doi: 10.1016/J.PROCONT.2013.08.003.
- [94] K. J. Åström and T. Häggglund, *PID Controllers: Theory, Design, and Tuning*, 2nd ed. ISA: The Instrumentation, Systems, and Automation Society, 1995.
- [95] A. Visioli, *Practical PID Control*, vol. 2006. London, UK: Springer, 2006.
- [96] I. Kamenko, F. Kulić, V. Bugarski, and P. Nikolić, "Analiza uticaja metoda za modelovanje realnih industrijskih procesa u postupku podešavanja parametara PID regulatora," in *XXXIV Nacionalna konferencija sa međunarodnim učešćem procesna tehnika i energetika PTEP 2022*, 2022, p. 29.
- [97] C.-C. Yu, *Autotuning of PID Controllers - A Relay Feedback Approach*, 2nd ed. Springer, 2006.
- [98] W. Li, E. Eskinat, and W. L. Luyben, "An Improved Autotune Identification Method," *Ind Eng Chem Res*, vol. 30, no. 7, pp. 1530–1541, Jul. 1991, doi: 10.1021/IE00055A019.
- [99] T. B. Šekara and M. R. Mataušek, "Relay-based critical point estimation of a process with the PID controller in the loop," *Automatica*, vol. 47, no. 5, pp. 1084–1088, May 2011, doi: 10.1016/j.automatica.2011.02.010.
- [100] T. B. Šekara and M. R. Mataušek, "Classification of dynamic processes and PID controller tuning in a parameter plane," *J Process Control*, vol. 21, no. 4, pp. 620–626, Apr. 2011, doi: 10.1016/j.jprocont.2010.12.004.
- [101] M. R. Mataušek and T. B. Šekara, "A fast closed-loop process dynamics characterization," *ISA Trans*, vol. 53, no. 2, pp. 489–496, 2014, doi: 10.1016/j.isatra.2013.12.006.
- [102] I. Kamenko, V. Čongradac, and F. Kulić, "A novel fuzzy logic scheme for PID controller auto-tuning," *Automatika*, vol. 63, no. 2, pp. 365–377, 2022, doi: 10.1080/00051144.2022.2043988.
- [103] "EcoStruxure Building Operation," *Online Help*. <https://ecostruxure-building-help.se.com/bms/home/index.castle> (accessed Sep. 28, 2022).
- [104] F. Holmberg, "Implementation of a PID Controller for Building Automation," Master thesis, Lund Institute of Technology, 2001.
- [105] R. Wiltshire, *Advanced District Heating and Cooling (DHC) Systems*. Woodhead Publishing, 2015.
- [106] S. L. Lazarević, V. D. Čongradac, A. S. Andjelković, M. v. Kljajić, and Ž. S. Kanović, "District heating substation elements modeling for the development of the real-time model," *Thermal Science*, vol. 23, no. 3B, pp. 2061–2070, 2019, doi: 10.2298/TSCI181226031L.
- [107] D. P. D. Theodore L. Bergman, Adrienne S. Lavine, Frank P. Incropera, *Fundamentals of Heat and Mass Transfer*, 6th ed. Hoboken, NJ, USA: John Wiley & Sons, 2006.

Прилози

I Прилог - Изворни код предложене методе у Simulink развојном окружењу

I.1 Функцијски блок *initTuningPID*



Функција *\systemIdentification*

```
function [setpoint, K, L, T, finish] = sistemIdentification(start, currentValue, basicSetpoint,
stepPercent,steadyStatePercent, currentTime)
global steadyStateValue;
global steadyStateTime;
global integral1hlp;
```

```
global settlingTimeHlp;
global time0;
global time1;
global time2;
global time3;
global hlp0;
global hlp1;
global hlp2;
global hlp3;
global hlp4;
global integral1;
global integral2;
global A0;
global A1;
global currentTimeOld;
global KHlp;
global setpointHlp;
global step;
global LHlp;
global THlp;
global finish_hlp;
global currentValueOld;
global bias;
global steady_state_hlp;

if start
    if ~hlp0
        time0=currentTime;
        hlp0=true;
        hlp1=true;
    end
    % provera da li je sistem u stacionarnom stanju
    if hlp1
        setpointHlp=basicSetpoint;
        if abs(currentValue-steadyStateValue)<=steadyStatePercent*steadyStateValue/100
            steady_state_hlp=true;
        else
            steadyStateTime=currentTime;
            steadyStateValue=currentValue;
            steady_state_hlp=false;
        end
        if steadyStateTime-time0<0.66*(currentTime-time0) && steady_state_hlp &&
            currentTime-time0>10
            KHlp=currentValue/setpointHlp;
            hlp1=false;
            hlp2=true;
            time1=currentTime;
            currentTimeOld=currentTime;
            currentValueOld=currentValue;
        end
    end
    end
    %računanje provršine A0
    if hlp2
        step=stepPercent*basicSetpoint/100;
        setpointHlp=basicSetpoint+step;
        if abs(KHlp*setpointHlp-currentValue)<steadyStatePercent*step*KHlp/100
            if ~integral1hlp
                integral1hlp=true;
                settlingTimeHlp=currentTime;
            end
        else
            integral1hlp=false;
        end
        integral1=integral1+ ((KHlp*setpointHlp-currentValue)+(KHlp*setpointHlp-
            currentValueOld))*(currentTime-currentTimeOld)/2;
        currentTimeOld=currentTime;
        currentValueOld=currentValue;
        if settlingTimeHlp-time1<0.66*(currentTime-time1) && integral1hlp
            A0=integral1;
            integral1hlp=false;
            integral1=0;
            hlp2=false;
            hlp3=true;
            time2=currentTime;
        end
    end
end
```

```

        currentTimeOld=currentTime;
        currentValueOld=currentValue;
        bias=KHlp*setpointHlp-currentValue; %bias
    end
end
%računanje provršine A1
if hlp3
    step=stepPercent*basicSetpoint/100;
    setpointHlp=basicSetpoint;
    integral2=integral2+((KHlp*(setpointHlp+step)-bias-
        currentValue)+(KHlp*(setpointHlp+step)-bias-currentValueOld))*(currentTime-
        currentTimeOld)/2;
    currentTimeOld=currentTime;
    currentValueOld=currentValue;
    if currentTime-time2>A0/(step*KHlp)
        A1=integral2;
        integral2=0;
        hlp3=false;
        hlp4=true;
        time3=currentTime;
        currentTimeOld=currentTime;
        currentValueOld=currentValue;
    end
end
%računanje vremenske konstante i transportnog kašnjenja
if hlp4
    setpointHlp=basicSetpoint;
    THlp=A1*exp(1)/((stepPercent*basicSetpoint/100)*KHlp);
    LHlp=A0/((stepPercent*basicSetpoint/100)*KHlp)-THlp;
    finish_hlp=true;
end
else
    setpointHlp=basicSetpoint;
    hlp0=false;
    hlp1=false;
    hlp2=false;
    hlp3=false;
    hlp4=false;
    finish_hlp=false;
    A0=0;
    A1=0;
    LHlp=0;
    THlp=0;
    integral1=0;
    integral2=0;
    steadyStateTime=0;
    settlingTimeHlp=0;
end
K=KHlp;
setpoint=setpointHlp;
L=LHlp;
T=THlp;
finish=finish_hlp;

```

Функција *parametarsCalculation*

```

function [Kp,Ti,Td,finish] = parametarsCalculation (K, L, T, start, tuningMethod)
global Kp_hlp;
global Ti_hlp;
global Td_hlp;
global finish_hlp2;

if start && L~=0 && T~=0
    if tuningMethod==1
        %ZN
        Kp_hlp=1.2*T/(K*L);
        Ti_hlp=2*L;
        Td_hlp=0.5*L;
    elseif tuningMethod==2
        %Astrom AMIGO
        Kp_hlp=(0.2+0.45*T/L)/K;
        Ti_hlp=(0.4*L+0.8*T)*L/(L+0.1*T);
        Td_hlp= 0.5*L*T/(0.3*L+T);
    end
end

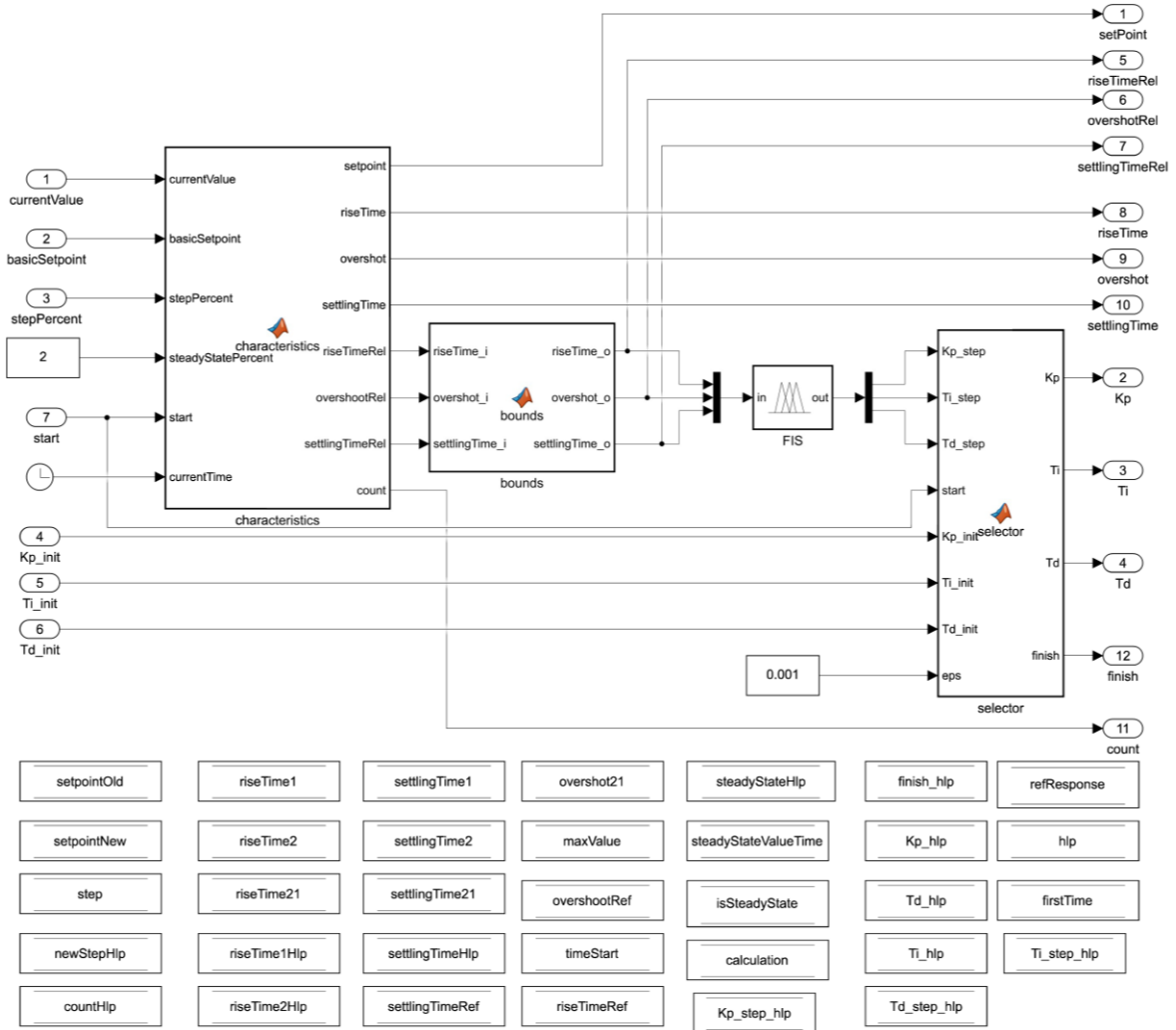
```

```

        finish_hlp2=true;
    else
        finish_hlp2=false;
    end
    Kp=Kp_hlp;
    Ti=Ti_hlp;
    Td=Td_hlp;
    finish=finish_hlp2;

```

I.2 Функцијски блок *autoTuningPID*



Функција *characteristics*

```

function [setpoint, riseTime, overshoot, settlingTime, riseTimeRel, overshootRel, settlingTimeRel, count]
= characteristics(currentValue, basicSetpoint, stepPercent, steadyStatePercent, start, currentTime)
    global setpointOld;
    global setpointNew;
    global step;
    global newStepHlp;
    global riseTime1;
    global riseTime2;
    global riseTime21;
    global riseTime1Hlp;
    global riseTime2Hlp;
    global riseTimeRef;
    global settlingTime1;
    global settlingTime2;
    global settlingTime21;
    global settlingTimeHlp;
    global settlingTimeRef;
    global overshoot21;

```

```
global maxValue;
global overshootRef;
global steadyStateValueTime;
global isSteadyState;
global calculation;
global refResponse;
global hlp;
global countHlp;
global firstTime;
global steadyStateHlp;
global timeStart;

if start
    if ~firstTime
        firstTime=true;
        riseTime21=0;
        overshoot21=0;
        settlingTime21=0;
        countHlp=0;
        timeStart=currentTime;
    end
    if ~isSteadyState
        setpointNew=basicSetpoint;
        setpointOld=basicSetpoint;

        if abs(currentValue-setpointNew)<=steadyStatePercent*setpointNew/100
            if ~steadyStateHlp
                steadyStateValueTime=currentTime;
                steadyStateHlp=true;
            end
        else
            steadyStateHlp=false;
        end
        if steadyStateValueTime-timeStart<0.5*(currentTime-timeStart) && steadyStateHlp &&
            abs(currentValue-setpointNew)<0.0001
            setpointNew=basicSetpoint+basicSetpoint*stepPercent/100;
            setpointOld=basicSetpoint;
            isSteadyState=true;
            calculation=true;
            settlingTime1=currentTime;
            maxValue=setpointOld;
            settlingTimeRef=100;
            newStepHlp=false;
            countHlp=0;
        end
    end
    if newStepHlp && ~refResponse
        countHlp=countHlp+1;
        refResponse=true;
        newStepHlp=false;

        if riseTime1Hlp && riseTime2Hlp
            riseTimeRef=riseTime2-riseTime1;
        else
            riseTimeRef=currentTime-settlingTime1;
        end
        riseTime1Hlp=false;
        riseTime2Hlp=false;
        riseTime2=0;
        riseTime1=0;

        if settlingTimeHlp
            settlingTimeRef=settleTime2-settleTime1;
        else
            settlingTimeRef=currentTime-settleTime1;
        end
        settlingTimeHlp=false;
        settlingTime1=currentTime;
        settlingTime2=currentTime;
        overshootRef=abs(maxValue-setpointNew)/abs(setpointNew-setpointOld)*100;

        hlp=setpointOld;
        setpointOld=setpointNew;
        setpointNew=hlp;
    end
end
```



```
maxValue=setpointOld;

riseTime21=riseTimeRef;
overshot21=overshootRef;
settlingTime21=settlingTimeRef;

if overshootRef<0.001
    overshootRef=0.001;
end
end
if newStepHlp && refResponse
    countHlp=countHlp+1;
    newStepHlp=false;

    if riseTime1Hlp && riseTime2Hlp
        riseTime21=riseTime2-riseTime1;
    else
        riseTime21=currentTime-settlingTime1;
    end
    riseTime1Hlp=false;
    riseTime2Hlp=false;
    riseTime2=0;
    riseTime1=0;
    if settlingTimeHlp
        settlingTime21=settlingTime2-settlingTime1;
    else
        settlingTime21=currentTime-settlingTime1;
    end
    settlingTimeHlp=false;
    settlingTime1=currentTime;
    settlingTime2=currentTime;

    if setpointNew-setpointOld~=0
        overshoot21=abs(maxValue-setpointNew)/abs(setpointNew-setpointOld)*100;
    end

    hlp=setpointOld;
    setpointOld=setpointNew;
    setpointNew=hlp;
    maxValue=setpointOld;
end
if calculation
    step=setpointNew-setpointOld;
    if step>0
        if currentValue>=setpointOld+0.1*step && ~riseTime1Hlp
            riseTime1=currentTime;
            riseTime1Hlp=true;
        end
        if currentValue>=setpointOld+0.9*step && ~riseTime2Hlp
            riseTime2=currentTime;
            riseTime2Hlp=true;
        end
        if abs(currentValue-setpointNew)<steadyStatePercent*step/100
            if ~settlingTimeHlp
                settlingTime2=currentTime;
                settlingTimeHlp=true;
            end
            if currentTime-settlingTime2>0.5*(settlingTime2-settlingTime1)
                newStepHlp=true;
            end
        else
            settlingTimeHlp=false;
        end
        if currentTime-settlingTime1>settlingTimeRef*5
            newStepHlp=true;
        end
        if currentValue>maxValue
            maxValue=currentValue;
        end
    else
        if currentValue<=setpointOld+0.1*step && ~riseTime1Hlp
            riseTime1=currentTime;
            riseTime1Hlp=true;
        end
    end
end
```

```

        if currentValue<=setpointOld+0.9*step && ~riseTime2Hlp
            riseTime2=currentTime;
            riseTime2Hlp=true;
        end
        if abs(currentValue-setpointNew)<steadyStatePercent*abs(step)/100
            if ~settlingTimeHlp
                settlingTime2=currentTime;
                settlingTimeHlp=true;
            end
            if currentTime-settlingTime2>0.5*(settlingTime2-settlingTime1)
                newStepHlp=true;
            end
        else
            settlingTimeHlp=false;
        end
        if currentTime-settlingTime1>settlingTimeRef*5
            newStepHlp=true;
        end
        if currentValue<maxValue
            maxValue=currentValue;
        end
    end
end
else
    isSteadyState=false;
    calculation=false;
    refResponse=false;
    setpointOld=0;
    setpointNew=basicSetpoint;
    settlingTime1=currentTime;
    maxValue=setpointNew;
    newStepHlp=false;
end
if riseTimeRef~=0
    riseTimeRel=(riseTime21/riseTimeRef)*100;
else
    riseTimeRel=0;
end
if overshootRef~=0
    overshootRel=overshot21/overshootRef*100;
else
    overshootRel=0;
end

if settlingTimeRef~=0
    settlingTimeRel=(settlingTime21/settlingTimeRef)*100;
else
    settlingTimeRel=0;
end
count=countHlp;
riseTime=riseTime21;
overshot=overshot21;
settlingTime=settlingTime21;
setpoint=setpointNew;
end
end

```

Функција *\bounds*

```

function [riseTime_o,overshot_o,settlingTime_o] = bounds(riseTime_i,overshot_i,settlingTime_i)
    riseTime_o=min(riseTime_i,200);
    overshoot_o=min(overshot_i,200);
    settlingTime_o=min(settlingTime_i,200);
end

```

Функција *\selector*

```

function [Kp,Ti,Td,finish] = selector(Kp_step, Ti_step, Td_step, start,Kp_init,Ti_init,Td_init,eps)
    global Kp_hlp;
    global Ti_hlp;
    global Td_hlp;
    global Kp_step_hlp;
    global Ti_step_hlp;
    global Td_step_hlp;

```

```

global finish_hlp;

if ~start
    Kp_hlp=Kp_init;
    Ti_hlp=Ti_init;
    Td_hlp=Td_init;
    Kp_step_hlp=Kp_step;
    Ti_step_hlp=Ti_step;
    Td_step_hlp=Td_step;
    finish_hlp=false;
else
    if Kp_step_hlp~=Kp_step
        Kp_hlp=Kp_hlp+Kp_hlp*Kp_step;
        Kp_step_hlp=Kp_step;
        if abs(Kp_step)<eps && abs(Ti_step)<eps && abs(Td_step)<eps
            finish_hlp=true;
        end
    end
    if Ti_step_hlp~=Ti_step
        Ti_hlp=Ti_hlp+Ti_hlp*Ti_step;
        Ti_step_hlp=Ti_step;
        if abs(Kp_step)<eps && abs(Ti_step)<eps && abs(Td_step)<eps
            finish_hlp=true;
        end
    end
    if Td_step_hlp~=Td_step
        Td_hlp=Td_hlp+Td_hlp*Td_step;
        Td_step_hlp=Td_step;
        if abs(Kp_step)<eps && abs(Ti_step)<eps && abs(Td_step)<eps
            finish_hlp=true;
        end
    end
end
finish=finish_hlp;
Kp=Kp_hlp;
Ti=Ti_hlp;
Td=Td_hlp;
End

```

I.3 Скрипте за промену режима рада

Скрипта за примену режима рада Убрзање

```

riseTimeInput=40;
overshootInput=100;
settlingTimeInput=100;

FIS=readfis('fuzzyPID.fis');

FIS.input(1).mf(1).params=[0,0,riseTimeInput,max((50+riseTimeInput),100)];
FIS.input(2).mf(1).params=[0,0,overshootInput,max((50+overshootInput),100)];
FIS.input(3).mf(1).params=[0,0,settlingTimeInput,max((50+settlingTimeInput),100)];

FIS.input(1).mf(2).params=[riseTimeInput, max((50+riseTimeInput),100), max((100+riseTime-
Input),150)];
FIS.input(2).mf(2).params=[overshootInput, max((50+overshootInput),100), max((100+oversho-
otInput),150)];
FIS.input(3).mf(2).params=[settlingTimeInput, max((50+settlingTimeInput),100), max((100+settlingTi-
meInput),150)];

FIS.input(1).mf(3).params=[max((50+riseTimeInput),100), 200, 200, 200];
FIS.input(2).mf(3).params=[max((50+overshootInput),100), 200, 200, 200];
FIS.input(3).mf(3).params=[max((50+settlingTimeInput),100), 200, 200, 200];

```

Скрипта за примену режима рада Смањење прескока

```

riseTimeInput=100;
overshootInput=40;
settlingTimeInput=100;

```

```
FIS=readfis('fuzzyPID.fis');

FES.input(1).mf(1).params=[0,0,riseTimeInput,max((50+riseTimeInput),100)];
FES.input(2).mf(1).params=[0,0,overshootInput,max((50+overshootInput),100)];
FES.input(3).mf(1).params=[0,0,settlingTimeInput,max((50+settlingTimeInput),100)];

FIS.input(1).mf(2).params=[riseTimeInput, max((50+riseTimeInput),100), max((100+riseTime-
Input),150)];
FIS.input(2).mf(2).params=[overshootInput, max((50+overshootInput),100), max((100+oversho-
otInput),150)];
FIS.input(3).mf(2).params=[settlingTimeInput, max((50+settlingTimeInput),100), max((100+settlingTi-
meInput),150)];

FIS.input(1).mf(3).params=[max((50+riseTimeInput),100), 200, 200, 200];
FIS.input(2).mf(3).params=[max((50+overshootInput),100), 200, 200, 200];
FIS.input(3).mf(3).params=[max((50+settlingTimeInput),100), 200, 200, 200];
```

Скрипта за примену режим рада Балансирање

```
riseTimeInput=75;
overshootInput=75;
settlingTimeInput=75;

FIS=readfis('fuzzyPID.fis');

FES.input(1).mf(1).params=[0,0,riseTimeInput,max((50+riseTimeInput),100)];
FES.input(2).mf(1).params=[0,0,overshootInput,max((50+overshootInput),100)];
FES.input(3).mf(1).params=[0,0,settlingTimeInput,max((50+settlingTimeInput),100)];

FIS.input(1).mf(2).params=[riseTimeInput, max((50+riseTimeInput),100), max((100+riseTime-
Input),150)];
FIS.input(2).mf(2).params=[overshootInput, max((50+overshootInput),100), max((100+oversho-
otInput),150)];
FIS.input(3).mf(2).params=[settlingTimeInput, max((50+settlingTimeInput),100), max((100+settlingTi-
meInput),150)];

FIS.input(1).mf(3).params=[max((50+riseTimeInput),100), 200, 200, 200];
FIS.input(2).mf(3).params=[max((50+overshootInput),100), 200, 200, 200];
FIS.input(3).mf(3).params=[max((50+settlingTimeInput),100), 200, 200, 200];
```

II Прилог - Изворни код предложене методе у MATLAB развојном окружењу

II.1 Функција *initTuningPID*

```
function [kp,ti,td,T,L,Kv]= initTuningPID(plant,tuningMethod)
%-----aproksimacija KLT modelom-----
h=1;
[y,t]=step(plant);
t_end=t(end);
y_end=y(end);

K=round(y_end,2);
simStep=0.001;
dy=diff(y)./diff(t);
[dy_max,dy_ind]=max(dy);

if y_end/K>1.1 || round(y_end,2)==...
    round((dy_max*t_end+y(dy_ind)-dy_max*t(dy_ind)),2)
    %metod tegente za integralne procese
    dy=diff(y)./diff(t);
    [dy_max,dy_ind]=max(dy);
    processType=1;
    L=-(y(dy_ind)-dy_max*t(dy_ind))/(dy_max);
    Kv=dy_max;
    T=0;
else
    info=stepinfo(plant);
    t_end=info.SettlingTime*1.5;
    [y,t]=step(plant,0:simStep:t_end);
    %metod površina
    A0=trapz(t,abs(K*ones(length(t),1)-y));
    A1=trapz(t(t<=A0/K),y(t<=A0/K));
    T=A1*exp(1)/(h*K);
    L=A0/(h*K)-T;
    Kv=0;
    processType=0;
end
if processType==0
if tuningMethod==1
    %ZN
    kp=1.2*T/(K*L);
    ti=2*L;
    td= 0.5*L;
elseif tuningMethod==2
    %Astrom AMIGO
    kp=(0.2+0.45*T/L)/K;
    ti=(0.4*L+0.8*T)*L/(L+0.1*T);
    td= 0.5*L*T/(0.3*L+T);
end
else
if tuningMethod==1
    %ZN
    kp=1.2/(Kv*L);
    ti=2*L;
    td= 0.5*L;
elseif tuningMethod==2
    %Astrom AMIGO
    kp=0.45/Kv;
    ti=8*L;
    td= 0.5*L;
end
end
end
```

II.2 Функција *autoTuningPID*

```
function [results, results_header, ctrl,iterResults]= autoTuningPID(tuningMethod,plant,mode,
```

```

Kp_init_input, Ti_init_input, Td_init_input)

simStep=0.001;
eps=0.001;
maxNum=100;
fazi='FIS\fazi';

%---kreiranje zaglavlja rezultata---
if nargin==0
    results=[];
    results_header={'T','L','Kv','L/T','tau','N',...
        'Tr_init','Tr_final','Tr_%',...
        'O_init','O_final','O_%',...
        'Ts_init','Ts_final','Ts_%',...
        'IAE_init','IAE_final','IAE_%'...
        'P_init','P_final','P_%'...
        'U_init','U_final','U_%',...
        'Tsd_init','Tsd_final','Tsd_%',...
        'IAE_D_init','IAE_D_final','IAE_D_%'...
        'M_init','M_final','M_%',...
        'Kp_init','Kp_final',...
        'Ti_init','Ti_final',...
        'Td_init','Td_final','Stab_init','Stab_final'};
    return;
else
    results_header=[];
end

%---inicijalno podešavanje parametara PID regulatora---
if nargin<4
    [Kp,Ti,Td,T,L,Kv]= initTuningPID(plant, tuningMethod);
else
    Kp=Kp_init_input;Ti=Ti_init_input;Td=Td_init_input;
    L=0;T=0;Kv=0;
end

if Kp==0 && Ti==0 && Td==0
    results=cell(1,41);
    return
end

if mode==1
    riseTimeInput=40;
    overshootInput=100;
    settlingTimeInput=100;
elseif mode==2
    riseTimeInput=100;
    overshootInput=40;
    settlingTimeInput=100;
elseif mode==3
    riseTimeInput=75;
    overshootInput=75;
    settlingTimeInput=75;
else
    riseTimeInput=100;
    overshootInput=100;
    settlingTimeInput=100;
end

%---Učitavanje i podešavanje FES-a-----
FES=readfis(fazi);
FES.input(1).mf(1).params=[0,0,riseTimeInput,max((50+riseTimeInput),100)];
FES.input(2).mf(1).params=[0,0,overshootInput,max((50+overshootInput),100)];
FES.input(3).mf(1).params=[0,0,settlingTimeInput,max((50+settlingTimeInput),100)];

FES.input(1).mf(2).params=[riseTimeInput, max((50+riseTimeInput),100),
    max((100+riseTimeInput),150)];
FES.input(2).mf(2).params=[overshootInput, max((50+overshootInput),100),
    max((100+overshootInput),150)];
FES.input(3).mf(2).params=[settlingTimeInput, max((50+settlingTimeInput),100),
    max((100+settlingTimeInput),150)];

FES.input(1).mf(3).params=[max((50+riseTimeInput),100), 200, 200, 200];
FES.input(2).mf(3).params=[max((50+overshootInput),100), 200, 200, 200];

```

```

FES.input(3).mf(3).params=[max((50+settlingTimeInput),100), 200, 200, 200];

%---iterativno poboljšavanje odziva sistema pomoću FES-a-----
for i=1:maxNum
    ctrl{i}=pidstd(Kp(i),Ti(i),Td(i));
    system=feedback(ctrl{i}*plant,1);
    if i==1 %računanje k-ka referentnog odziva
        stp_info=stepinfo(system,1);
        settlingTime(i)=stp_info.SettlingTime;
    end
    %step setpoint
    [y,t]=step(system,0:simStep:5*settlingTime(1));
    stp_info=stepinfo(y,t,1);
    riseTime(i)=stp_info.RiseTime;
    overshoot(i)=stp_info.Overshoot;
    if overshoot(1)<0.001
        overshoot(1)=0.001;
    end

    settlingTime(i)=stp_info.SettlingTime;
    if isnan(stp_info.SettlingTime)
        settlingTime(i)=5*settlingTime(1);
    end
    IAE_step(i)=trapz(t,abs(1-y));

    riseTimeRel(i)=min(riseTime(i)*100/riseTime(1),200);
    overshootRel(i)=min(overshoot(i)*100/overshoot(1),200);
    settlingTimeRel(i)=min(settlingTime(i)*100/settlingTime(1),200);

    %step disturbance
    system=connect(append(ctrl{i},plant),[1 -2;2 1],[2],[2]);
    if i==1
        [y,t]=step(system);
        dist_info=stepinfo(y,t,0);
        settlingTimeD(i)=dist_info.TransientTime;
    end
    [y,t]=step(system,0:simStep:5*settlingTimeD(1));
    dist_info=stepinfo(y,t,0);
    peak(i)=dist_info.Peak;

    undershoot(i)=abs(min(y));
    if undershoot(1)<0.0001
        undershoot(1)=0.0001;
    end
    settlingTimeD(i)=dist_info.TransientTime;
    if isnan(dist_info.TransientTime)
        settlingTimeD(i)=5*settlingTimeD(1);
    end
    IAE_dist(i)=trapz(t,abs(0-y));

    [re1,im1]=nyquist(plant*ctrl{i});
    M(i)=max(1./(sqrt((-1-reshape(re1,1,length(re1))).^2+(-reshape(im1,1,length(im1))).^2)));

    %računanje parametara PIDa
    FESStep=evalfis(FES,[riseTimeRel(i), overshootRel(i),settlingTimeRel(i)]);
    if abs(FESStep(1))<eps && abs(FESStep(2))<eps && abs(FESStep(3))<eps % FES ne daje promenu
        break;
    end
    if i<maxNum
        Kp(i+1)=Kp(i)+Kp(i)*FESStep(1);
        Ti(i+1)=Ti(i)+Ti(i)*FESStep(2);
        Td(i+1)=Td(i)+Td(i)*FESStep(3);
    end
end
tmpOvershoot=(overshoot(i)-overshoot(1))*100/overshoot(1);
if abs(tmpOvershoot)<0.001
    tmpOvershoot=0;
end

tmpUndershoot=(undershoot(i)-undershoot(1))*100/undershoot(1);
if abs(tmpUndershoot)<0.0001
    tmpUndershoot=0;
end
end

```

```

%---rezultati-----
results={T,L,Kv,L/T,L/(L+T),i,...
riseTime(1), riseTime(i),(riseTime(i)-riseTime(1))*100/riseTime(1),...
overshoot(1), overshoot(i),tmpOvershoot...
settlingTime(1), settlingTime(i),(settlingTime(i)-settlingTime(1))*100/settlingTime(1),...
IAE_step(1), IAE_step(i),(IAE_step(i)-IAE_step(1))*100/IAE_step(1),...
peak(1), peak(i),(peak(i)-peak(1))*100/peak(1),...
undershoot(1), undershoot(i), tmpUndershoot,...
settlingTimeD(1), settlingTimeD(i),(settlingTimeD(i)-
settlingTimeD(1))*100/settlingTimeD(1),...
IAE_dist(1), IAE_dist(i),(IAE_dist(i)-IAE_dist(1))*100/IAE_dist(1),...
M(1),M(i),(M(i)-M(1))*100/M(1),...
Kp(1),Kp(i),...
Ti(1),Ti(i),...
Td(1),Td(i),...
stepinfo(feedback(ctrl{1}*plant,1)).Peak<10,stepinfo(feedback(ctrl{i}*plant,1)).Peak<10};
iterResults=[Kp;Ti;Td;riseTime;overshoot;settlingTime;M; peak ;undershoot;settlingTimeD];
%-----
end

```

П.3 Скрипта *tableResults*

```

function [data]=tableResults(mode)
disp('PIDFuzzyAutoTuning is running')
disp(['mode=',num2str(mode)]);

s=tf('s');
m_mean=[];
for j=1:2
    for m=1:9
        if m==7
            k_end=4;
        else
            k_end=1;
        end
        for k=1:k_end
            if m==1
                param = [0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 1, 1.3, 1.5, 2, 4, 6, 8, 10, 20,
                    50, 100, 200, 500, 1000];
            elseif m==2
                param = [0.01, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 1, 1.3, 1.5, 2, 4, 6, 8, 10, 20,
                    50, 100, 200, 500]; %izbacujem 0.02
            elseif m==3
                param = [0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 2, 5, 10];
            elseif m==4
                param = [3, 5, 6, 7, 8]; %izbacujem 4
            elseif m==5
                param = [0.1, 0.2, 0.3, 0.4,0.6 0.7, 0.8, 0.9]; % izbacujem 0.5
            elseif m==6
                param = [0.01, 0.02, 0.05, 0.3, 0.5, 0.7, 0.9, 1.0]; %izbacujem 0.1
            elseif m==7
                param = [0.01, 0.02, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0];
                param_ = [1, 2, 5, 10];
            elseif m==8
                param = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1];
            elseif m==9
                param = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0];
            end

            results=[];
            disp(['Method', num2str(j),' / Plant', num2str(m)]);
            for i=1:length(param)
                if m==1
                    plant=exp(-s)/(param(i)*s+1);
                elseif m==2
                    plant=exp(-s)/(param(i)*s+1)^2;
                elseif m==3
                    plant=1/(s+1)/((1+s*param(i))^2);
                elseif m==4
                    plant=1/(s+1)^param(i);
                elseif m==5
                    plant=1/(1+s)/(1+param(i)*s)/(1+param(i)^2*s)/(1+param(i)^3*s);
                elseif m==6
                    plant=exp(-s*param(i))/s/(1+s*(1-param(i)));
            end

```



```

elseif m==7
    plant=(param_(k))*exp(-s*param(i))/(1+param_(k)*s)/(1+s*(1-param(i)));
elseif m==8
    plant=(1-param(i)*s)/(s+1)^3;
elseif m==9
    plant=1/(s+1)/((s*param(i))^2+1.4*s*param(i)+1);
end
results_=AutoTuningPID(j,plant,mode);
results=[results;[m,j,param(i),results_]];
results=results(cell2mat(results(:,44))==true,:);
columns=[12,15,18,21,24,27,30,33,36,9];
m_mean=[m_mean;results(end,[1,2,3,columns])];
end
end
end
end
m_mean_mat=cell2mat(m_mean);

zn=[];
for i=1:9
    zn=[zn; size(m_mean_mat(m_mean_mat(:,1))==i & m_mean_mat(:,2)==1,:),1)
    mean(m_mean_mat(m_mean_mat(:,1))==i & m_mean_mat(:,2)==1,:));
end
zn_mean=mean(m_mean_mat(m_mean_mat(:,2)==1,:));
zn_mean=[size(m_mean_mat(m_mean_mat(:,2)==1,:),1) zn_mean];

ah=[];
for i=1:9
    ah=[ah; size(m_mean_mat(m_mean_mat(:,1))==i & m_mean_mat(:,2)==2,:),1)
    mean(m_mean_mat(m_mean_mat(:,1))==i & m_mean_mat(:,2)==2,:));
end
ah_mean=mean(m_mean_mat(m_mean_mat(:,2)==2,:));
ah_mean=[size(m_mean_mat(m_mean_mat(:,2)==2,:),1) ah_mean];

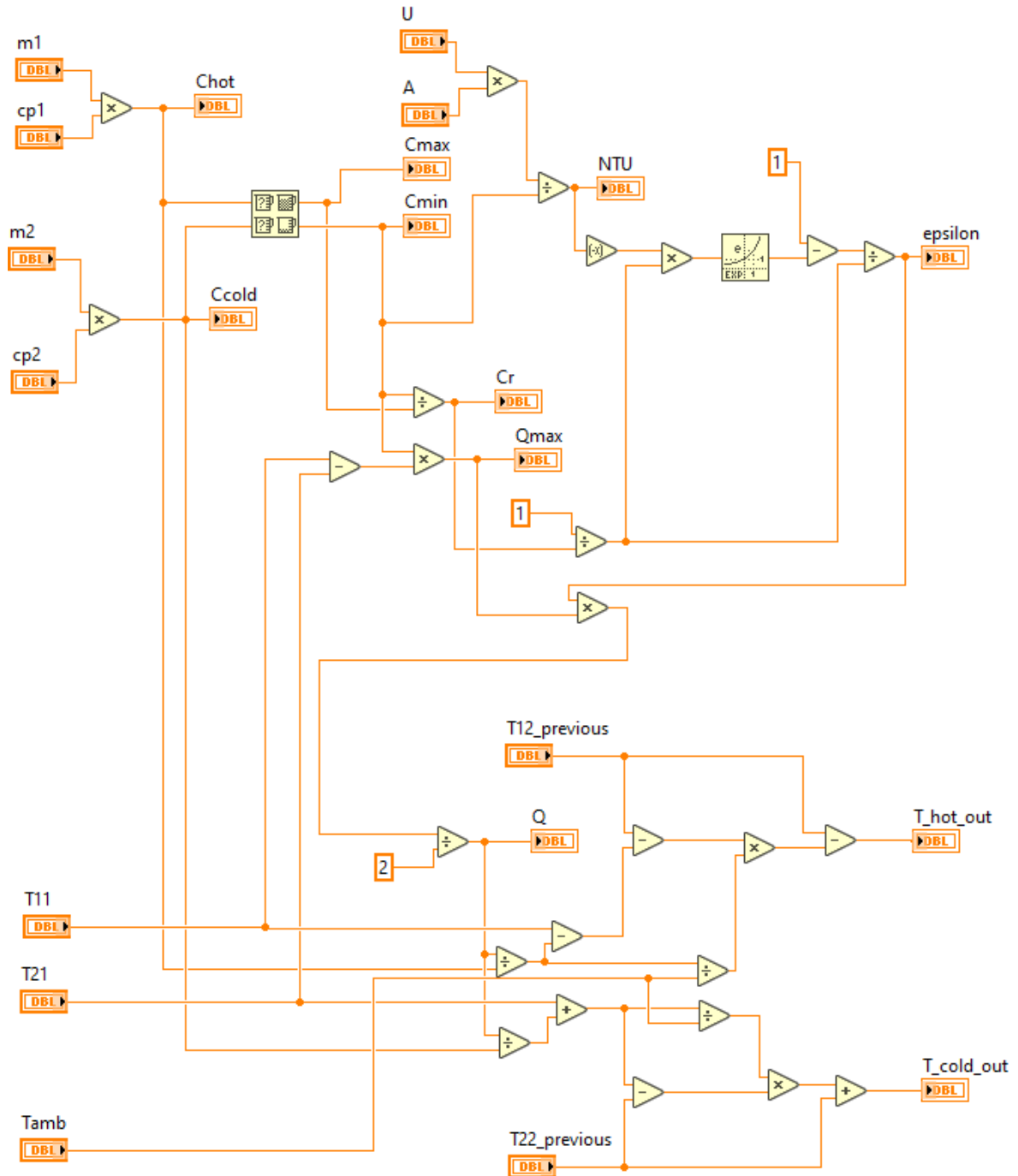
all=[];
for i=1:9
    all=[all; size(m_mean_mat(m_mean_mat(:,1))==i,:),1) mean(m_mean_mat(m_mean_mat(:,1))==i,:));
end
all_mean=mean(m_mean_mat);
all_mean=[size(m_mean_mat,1) all_mean];

data=[zn;zn_mean;ah;ah_mean;all;all_mean];
data=data(:,[1, 5:end]);
end

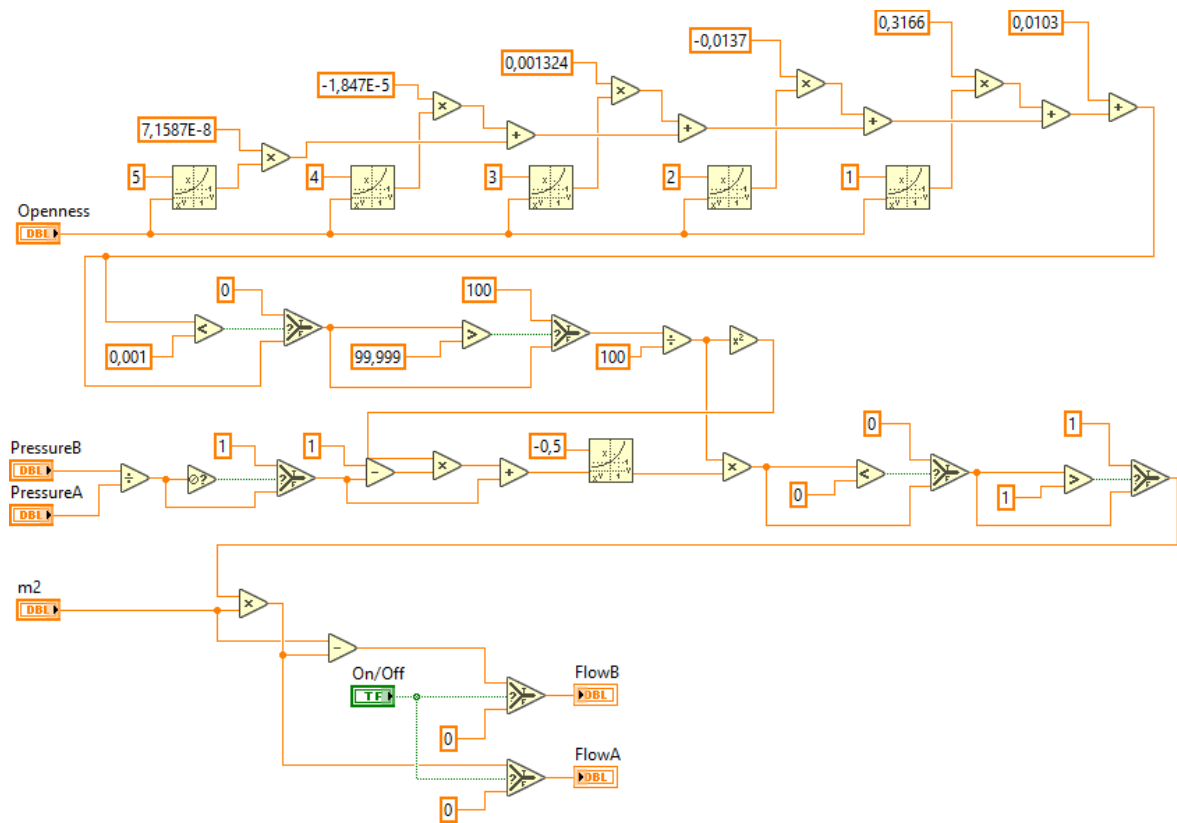
```

III Прилог - Изворни код модела топлотне подстанице у LabView развојном окружењу

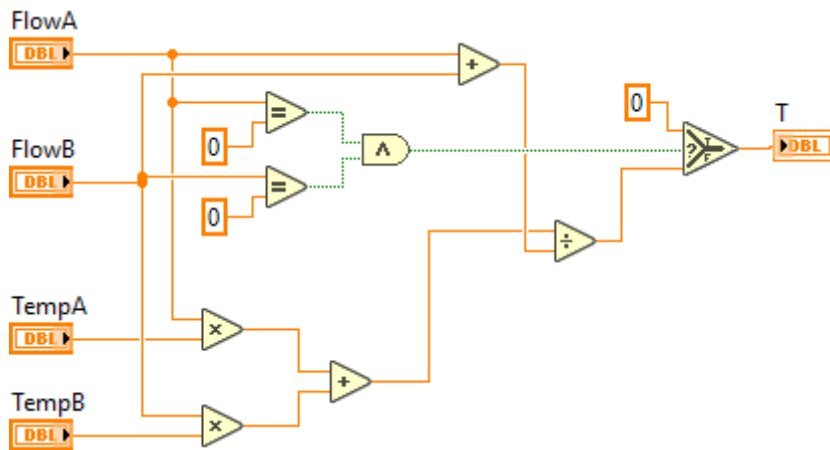
III.1 Модел измењивача топлоте



III.2 Модел вентила

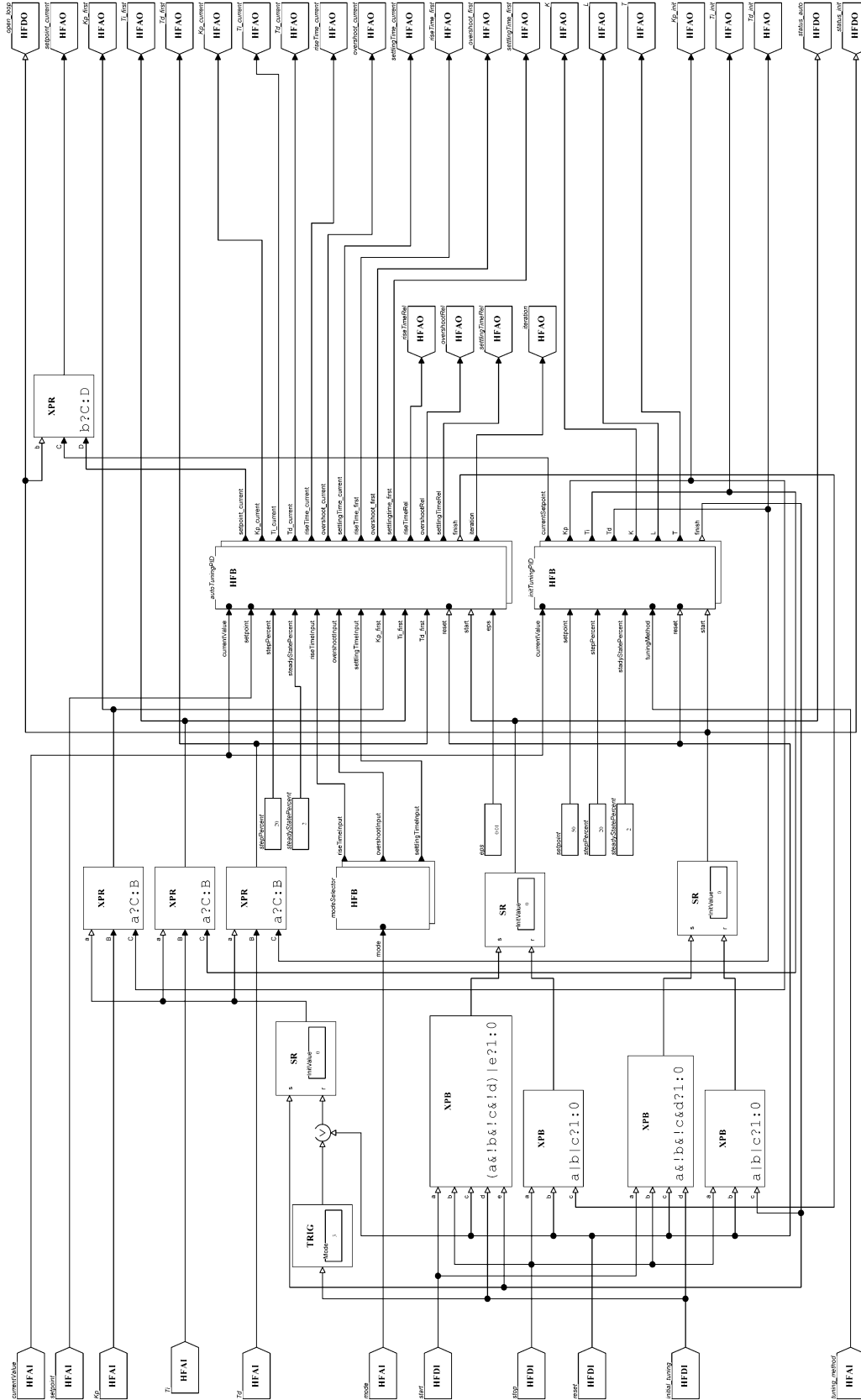


III.3 Модел мешања

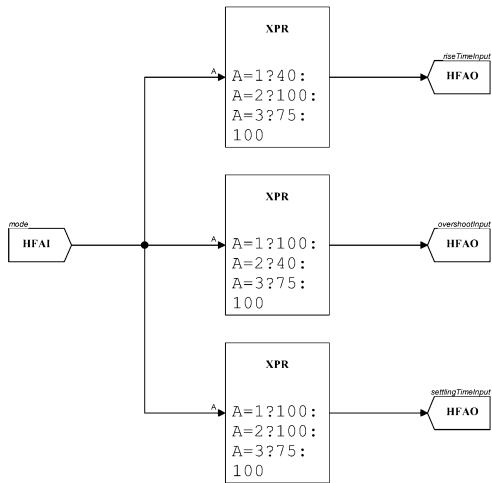


IV Прилог - Изворни код предложене методе у FBD програмском језику

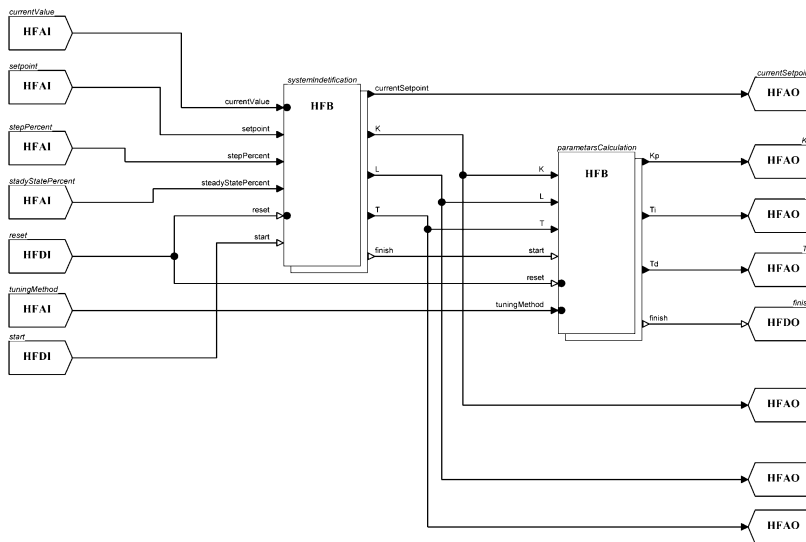
IV.1 Основни функцијски блок *FuzzyAutoTuning*



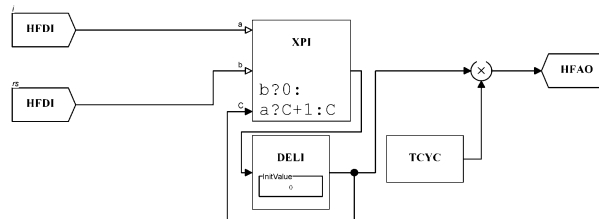
Функцијски блок *\modeSelector*



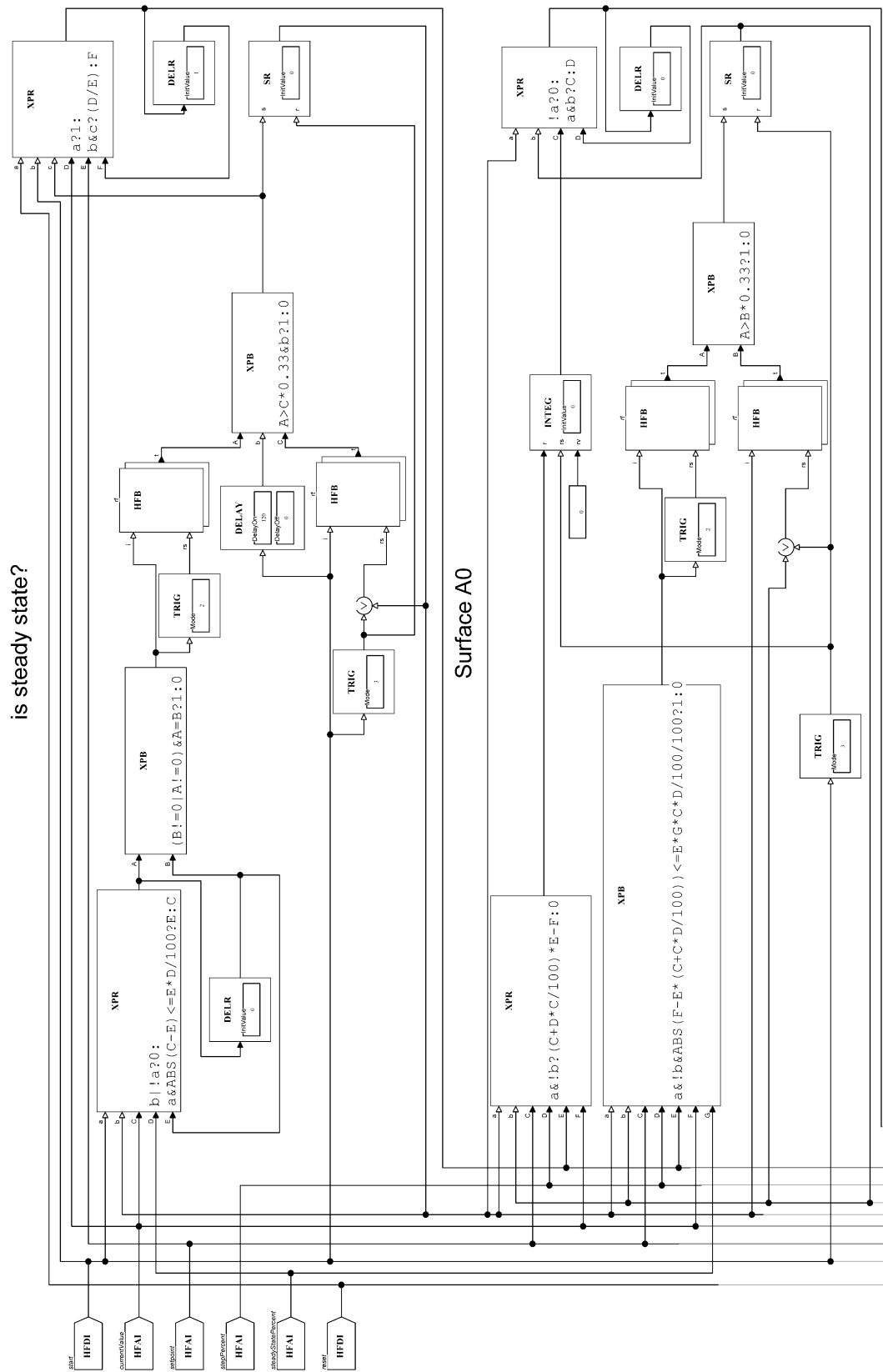
Функцијски блок *\initTuningPID*



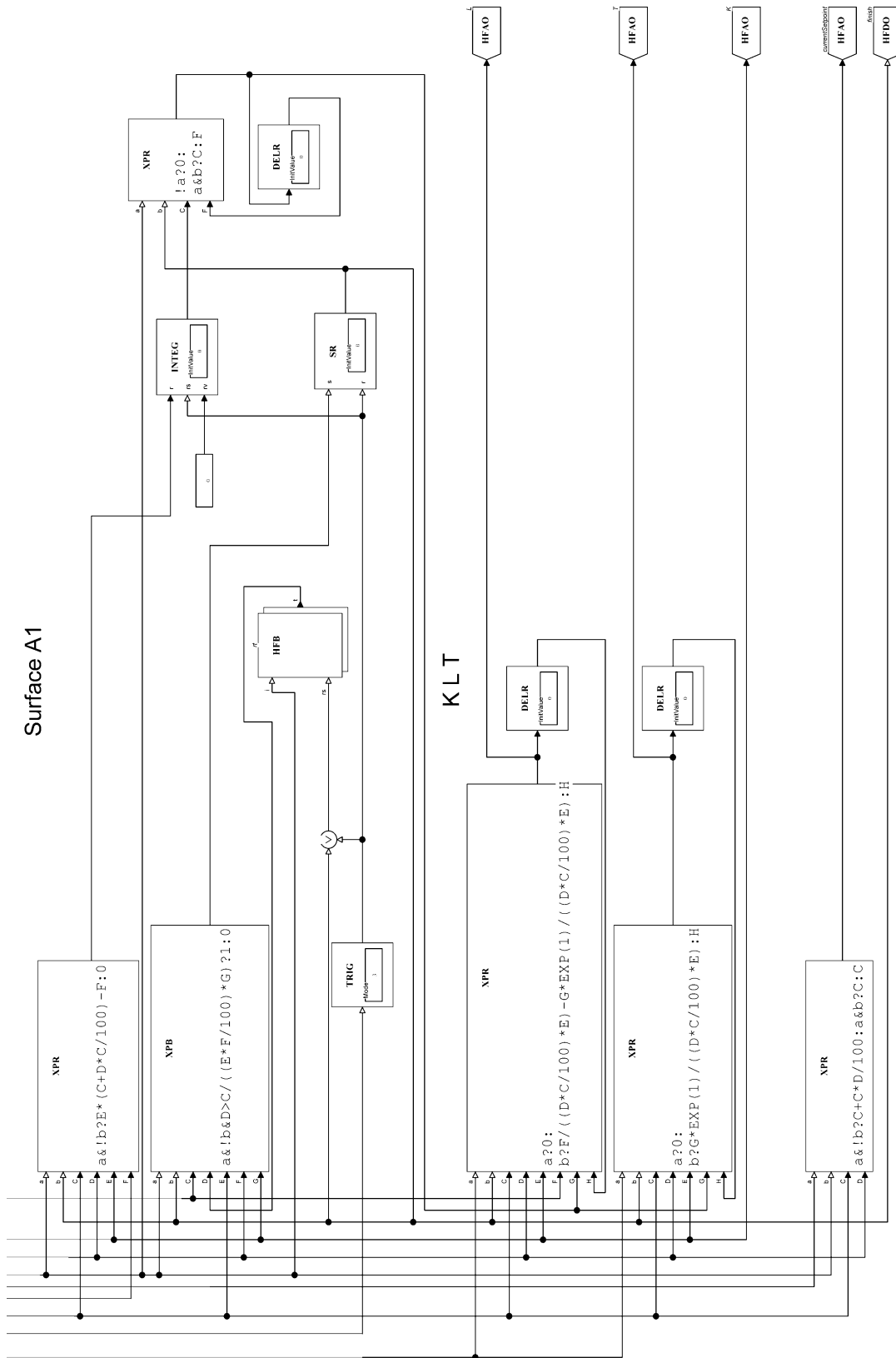
Функцијски блок *\initTuningPID\systemIdentification\rt*



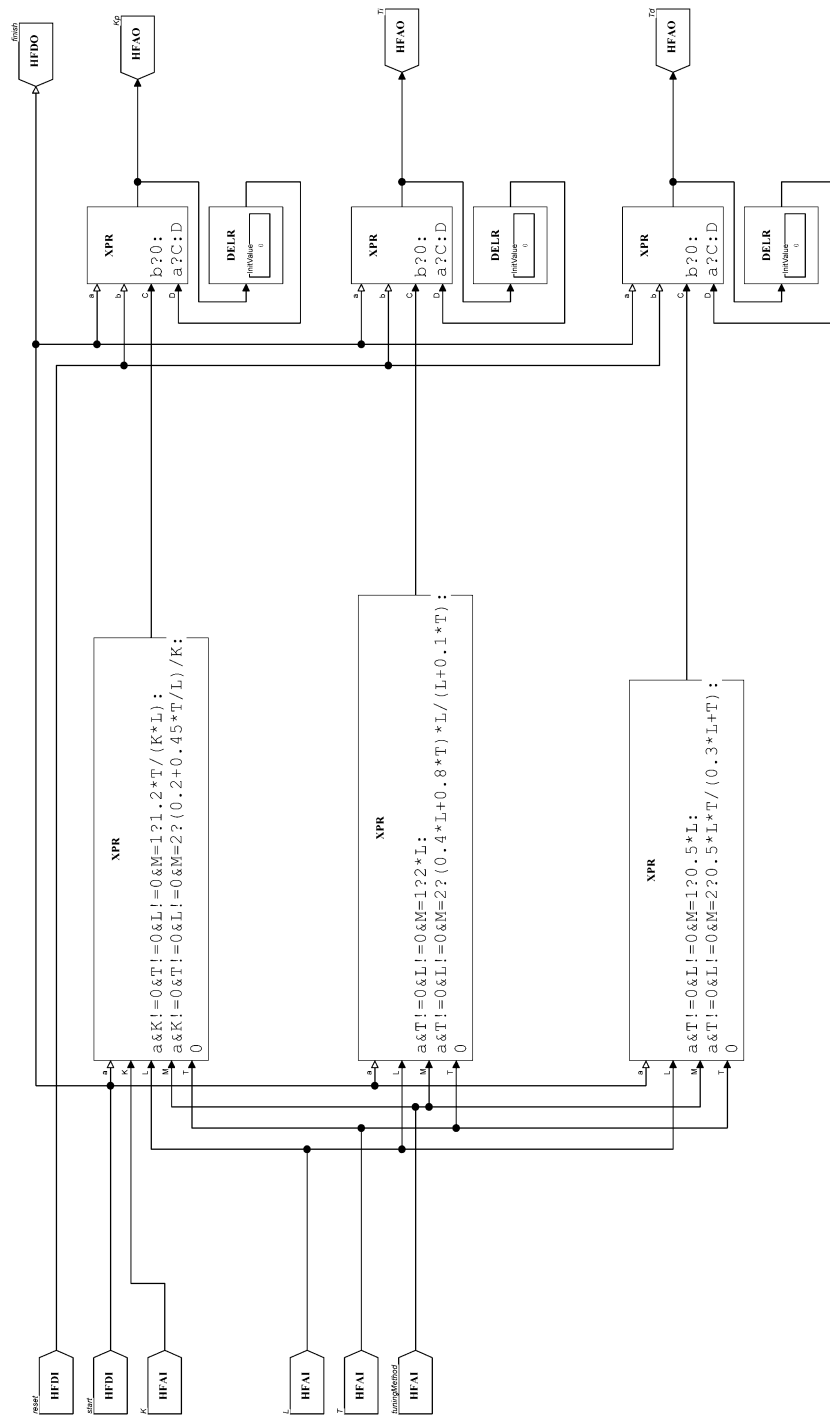
Функцијски блок `\initTuningPID\systemIdentification (1/2)`



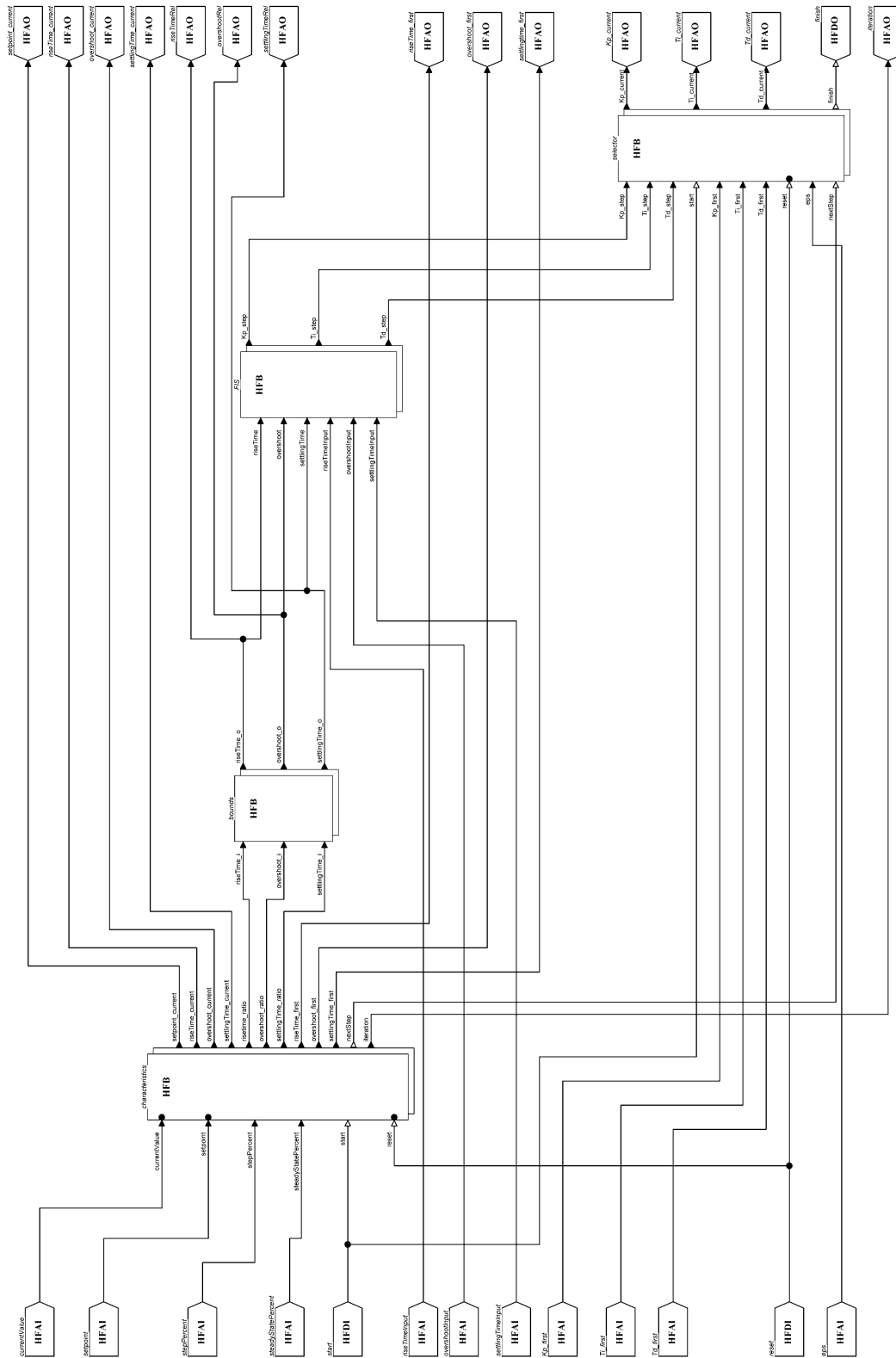
Функцијски блок *initTuningPID\systemIdentification (2/2)*



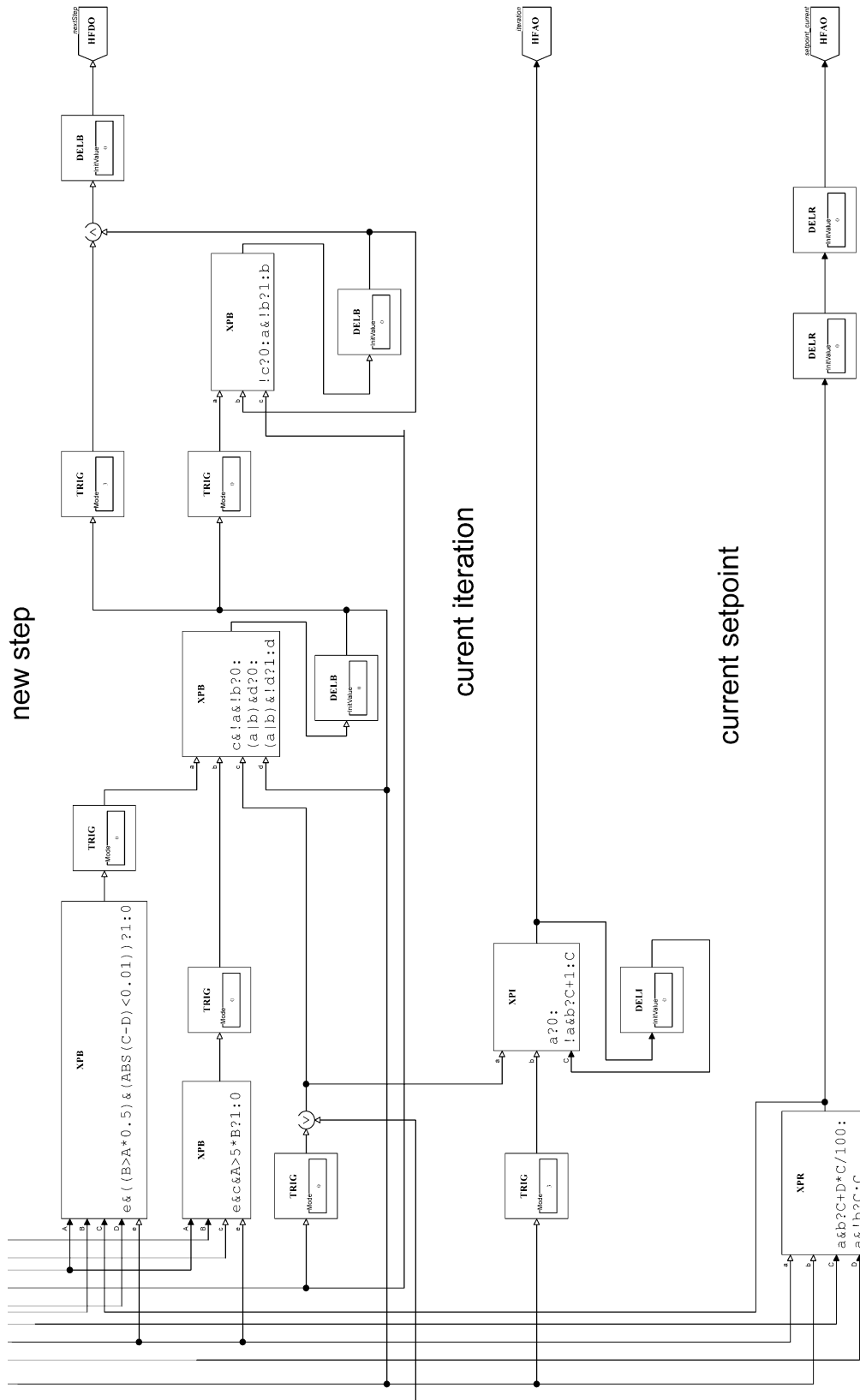
Функцијски блок *initTuningPIDparametersCalculation*



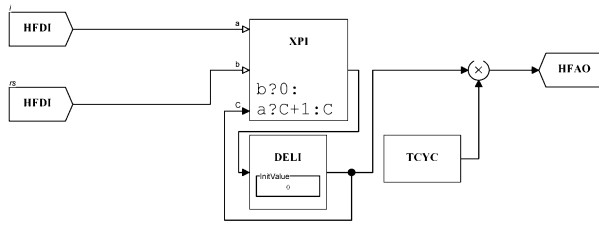
Функцијски блок *autoTuningPID*



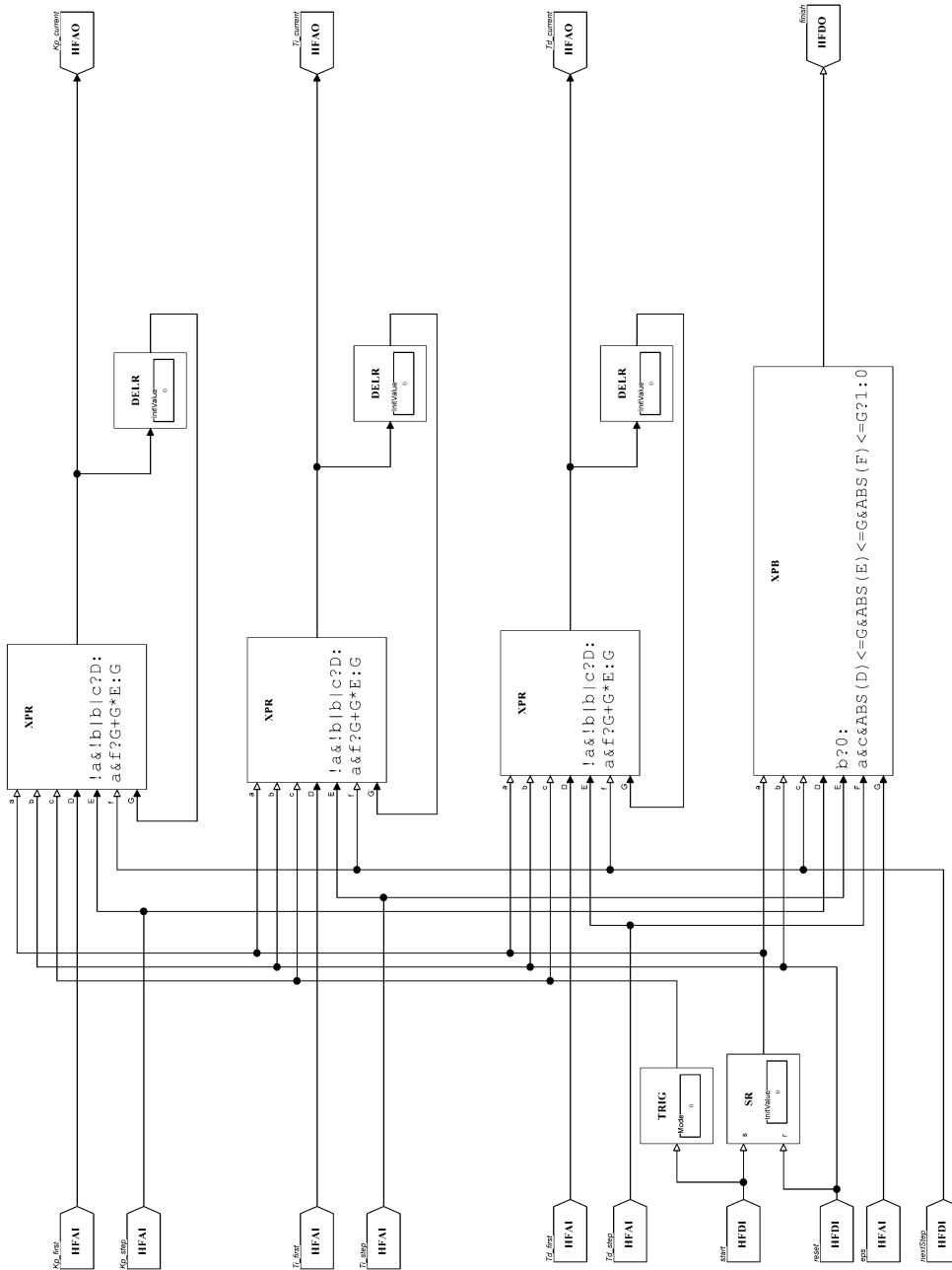
Функцијски блок *autoTuningPID* characteristics (3/3)



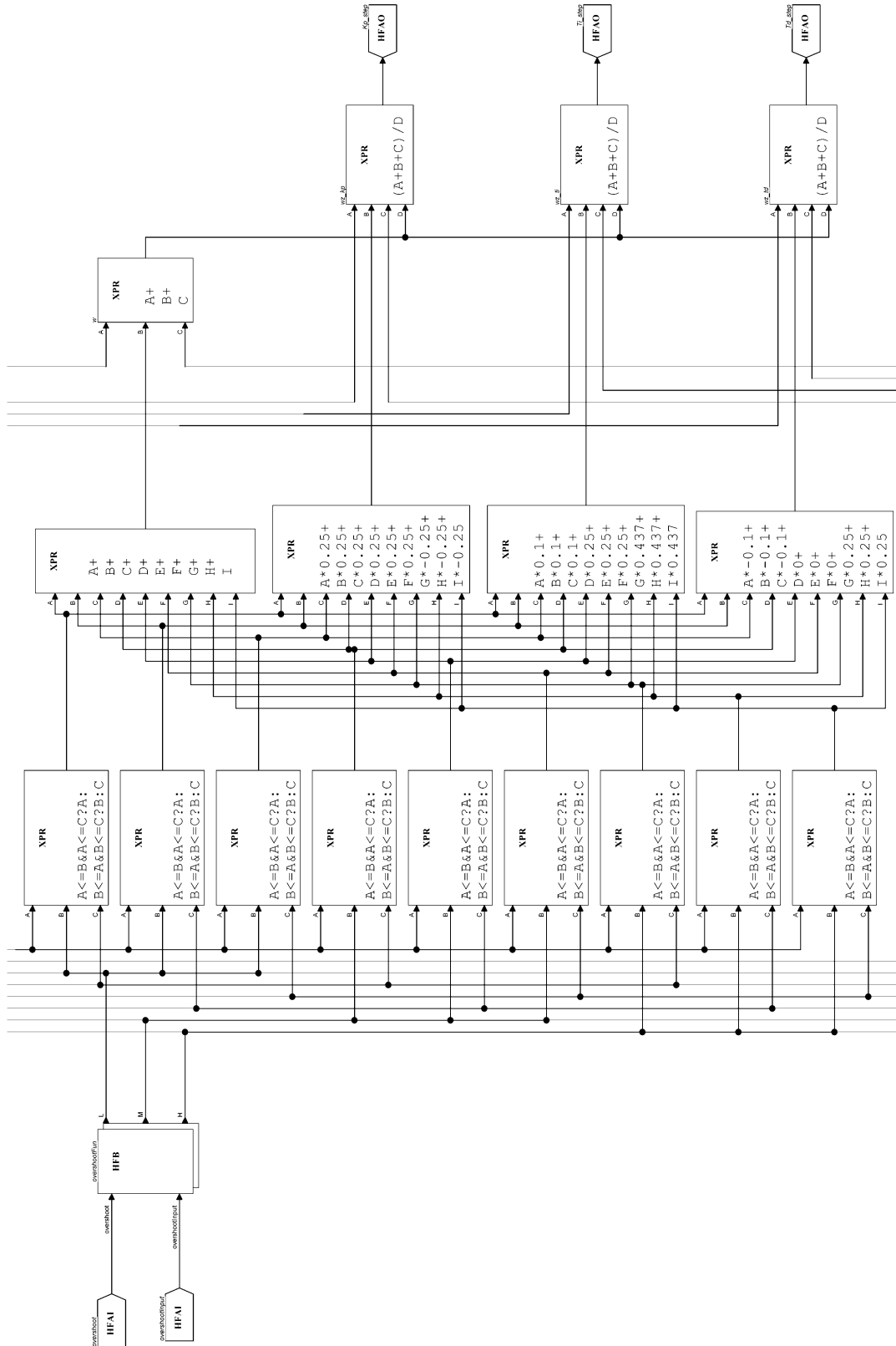
Функцијски блок *autoTuningPID\characteristics\rt*



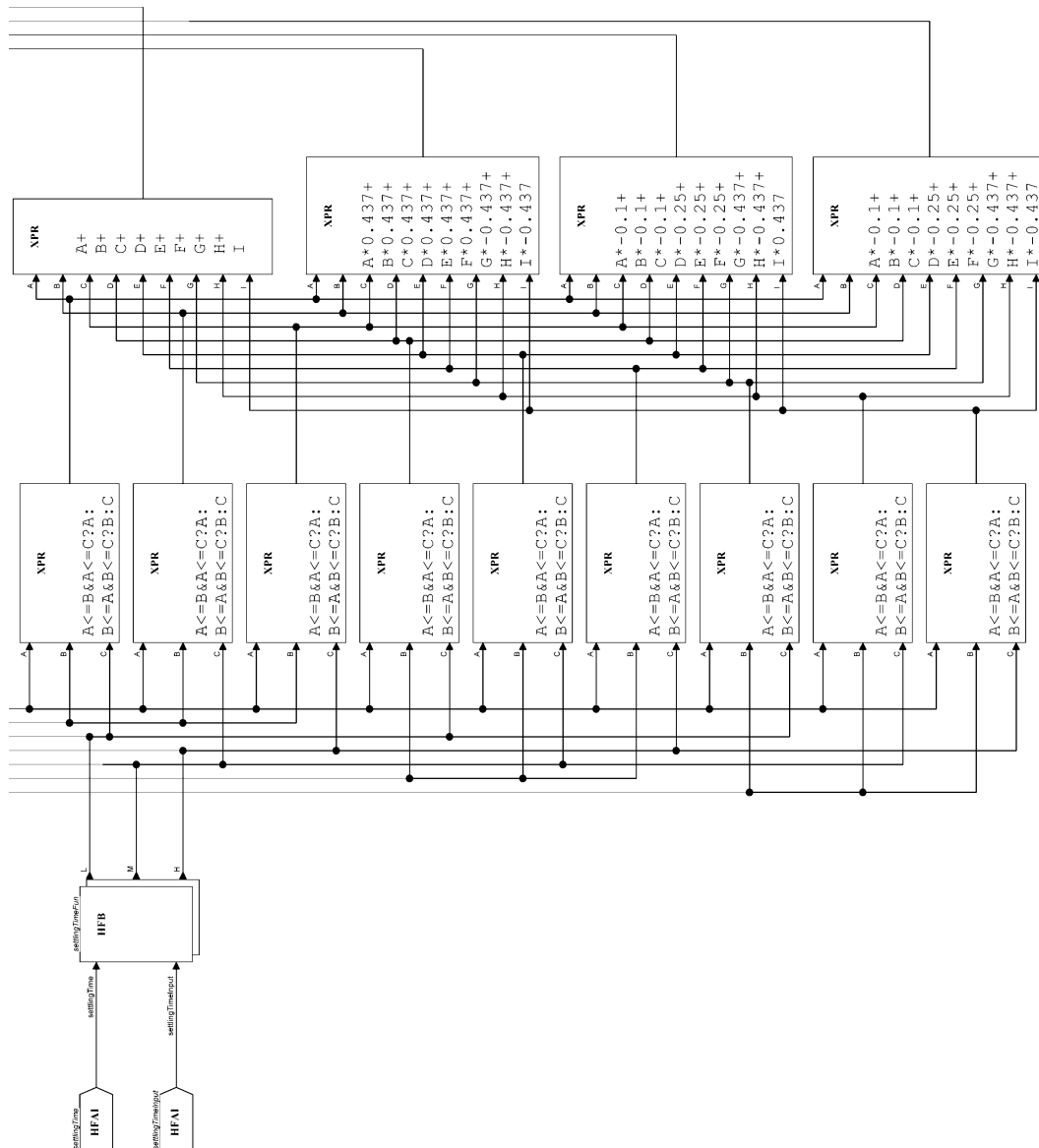
Функцијски блок *autoTuningPID\selector*



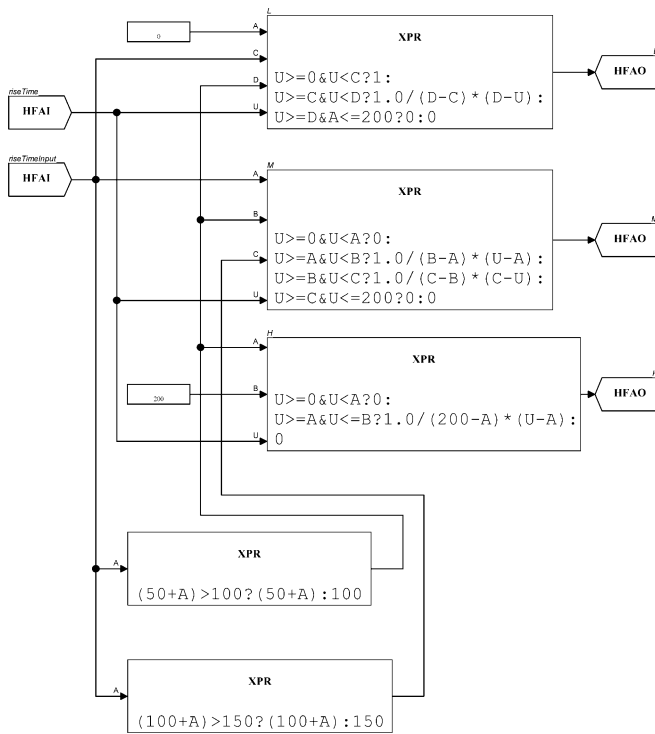
Функцијски блок *autoTuningPID\FIS* (2/3)



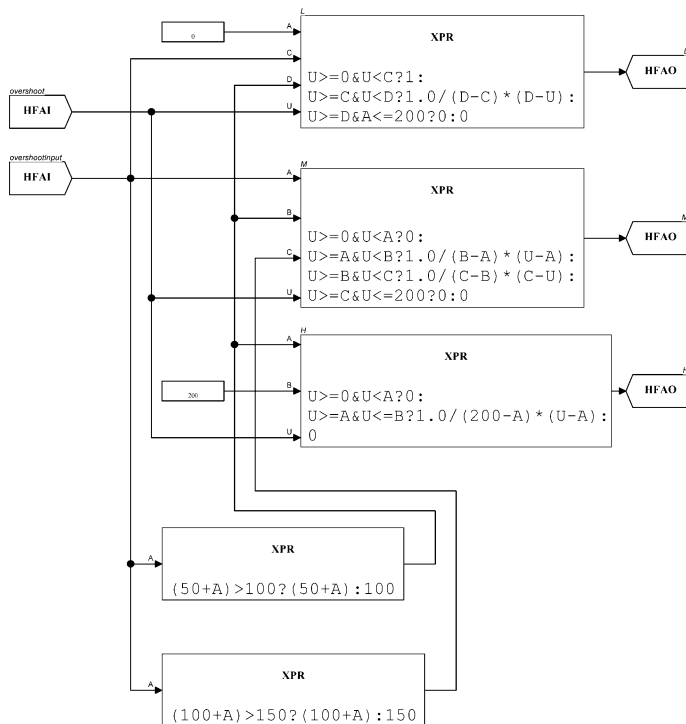
Функцијски блок *autoTuningPID\FIS* (3/3)



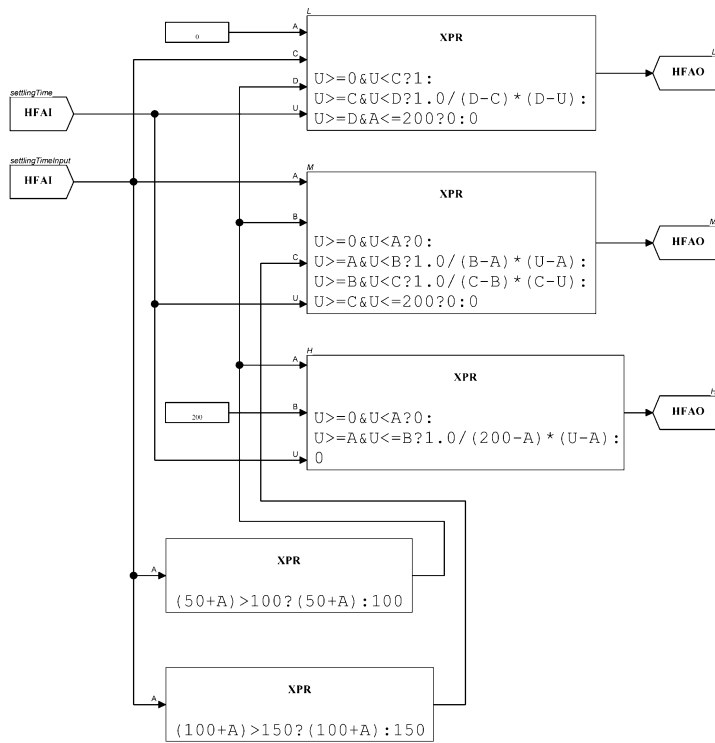
Функцијски блок $\backslash autoTuningPID \backslash FIS \backslash riseTimeFun$



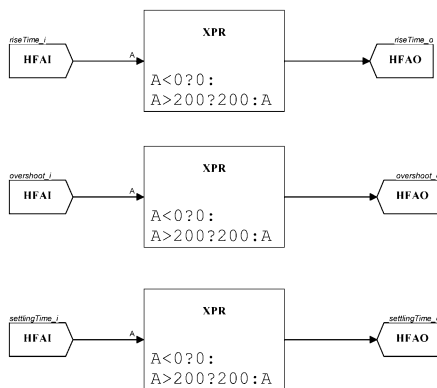
Функцијски блок $\backslash autoTuningPID \backslash FIS \backslash overshootFun$



Функцијски блок `\autoTuningPID\FIS\settlingTimeFun`



Функцијски блок `\autoTuningPID\bounds`



Овај Образац чини саставни део докторске дисертације, односно докторског уметничког пројекта који се брани на Универзитету у Новом Саду. Попуњен Образац укоричити иза текста докторске дисертације, односно докторског уметничког пројекта.

План третмана података

Назив пројекта/истраживања
Аутоматско подешавање параметара ПИД регулатора засновано на расплинутој логици са применом у аутоматизацији стамбено-пословних објеката
Назив институције/институција у оквиру којих се спроводи истраживање
Депарتمان за рачунарство и аутоматику, Факултет техничких наука, Универзитет у Новом Саду
Назив програма у оквиру ког се реализује истраживање
Истраживање је спроведено у сврху израде докторске дисертације у оквиру Докторских академских студија на студијском програму Рачунарство и аутоматика на Факултету техничких наука Универзитета у Новом Саду
1. Опис података
<p>1.1 Врста студије</p> <p><i>Укратко описати тип студије у оквиру које се подаци прикупљају</i></p> <p>У оквиру студије нису прикупљани подаци</p> <hr/> <hr/>
<p>1.2 Врсте података</p> <p>а) квантитативни</p> <p>б) квалитативни</p>
<p>1.3. Начин прикупљања података</p> <p>а) анкете, упитници, тестови</p> <p>б) клиничке процене, медицински записи, електронски здравствени записи</p> <p>в) генотипови: навести врсту _____</p> <p>г) административни подаци: навести врсту _____</p> <p>д) узорци ткива: навести врсту _____</p> <p>ђ) снимци, фотографије: навести врсту _____</p> <p>е) текст, навести врсту _____</p> <p>ж) мапа, навести врсту _____</p> <p>з) остало: описати _____</p>

1.3 Формат података, употребљене скале, количина података

1.3.1 Употребљени софтвер и формат датотеке:

- а) Ехсел фајл, датотека _____
- б) SPSS фајл, датотека _____
- в) PDF фајл, датотека _____
- г) Текст фајл, датотека _____
- д) JPG фајл, датотека _____
- ђ) Остало, датотека _____

1.3.2. Број записа (код квантитативних података)

- а) број варијабли _____
- б) број мерења (испитаника, процена, снимака и сл.) _____

1.3.3. Поновљена мерења

- а) да
- б) не

Уколико је одговор да, одговорити на следећа питања:

- а) временски размак између поновљених мера је _____
- б) варијабле које се више пута мере односе се на _____
- в) нове верзије фајлова који садрже поновљена мерења су именоване као _____

Напомене:

Да ли формати и софтвер омогућавају дељење и дугорочну валидност података?

а) Да

б) Не

Ако је одговор не, образложити

2. Прикупљање података

2.1 Методологија за прикупљање/генерисање података

2.1.1. У оквиру ког истраживачког нацрта су подаци прикупљени?

- а) експеримент, навести тип _____
- б) корелационо истраживање, навести тип _____
- ц) анализа текста, навести тип _____
- д) остало, навести шта _____

2.1.2 Навести врсте мерних инструмената или стандарде података специфичних за одређену научну дисциплину (ако постоје).

2.2 Квалитет података и стандарди

2.2.1. Третман недостајућих података

а) Да ли матрица садржи недостајуће податке? Да Не

Ако је одговор да, одговорити на следећа питања:

а) Колики је број недостајућих података? _____

б) Да ли се кориснику матрице препоручује замена недостајућих података? Да Не

в) Ако је одговор да, навести сугестије за третман замене недостајућих података

2.2.2. На који начин је контролисан квалитет података? Описати

2.2.3. На који начин је извршена контрола уноса података у матрицу?

3. Третман података и пратећа документација

3.1. Третман и чување података

3.1.1. Подаци ће бити депоновани у _____ репозиторијум.

3.1.2. URL адреса _____

3.1.3. DOI _____

3.1.4. Да ли ће подаци бити у отвореном приступу?

а) Да

б) Да, али после ембарга који ће трајати до _____

в) Не

Ако је одговор не, навести разлог _____

3.1.5. Подаци неће бити депоновани у репозиторијум, али ће бити чувани.

Образложење

3.2 Метаподаци и документација података

3.2.1. Који стандард за метаподатке ће бити примењен?

3.2.1. Навести метаподатке на основу којих су подаци депоновани у репозиторијум.

Ако је потребно, навести методе које се користе за преузимање података, аналитичке и процедуралне информације, њихово кодирање, детаљне описе варијабли, записа итд.

3.3 Стратегија и стандарди за чување података

3.3.1. До ког периода ће подаци бити чувани у репозиторијуму?

3.3.2. Да ли ће подаци бити депоновани под шифром? Да Не

3.3.3. Да ли ће шифра бити доступна одређеном кругу истраживача? Да Не

3.3.4. Да ли се подаци морају уклонити из отвореног приступа после извесног времена? Да Не

Образложити

4. Безбедност података и заштита поверљивих информација

Овај одељак МОРА бити попуњен ако ваши подаци укључују личне податке који се односе на учеснике у истраживању. За друга истраживања треба такође размислити заштиту и сигурност података.

4.1 Формални стандарди за сигурност информација/података

Истраживачи који спроводе испитивања с људима морају да се придржавају Закона о заштити података о личности

(https://www.paragraf.rs/propisi/zakon_o_zastiti_podataka_o_licnosti.html) и одговарајућег институционалног кодекса о академском интегритету.

4.1.2. Да ли је истраживање одобрено од стране етичке комисије? Да Не

Ако је одговор Да, навести датум и назив етичке комисије која је одобрила истраживање

4.1.2. Да ли подаци укључују личне податке учесника у истраживању? Да Не
Ако је одговор да, наведите на који начин сте осигурали поверљивост и сигурност информација везаних за испитанике:

- а) Подаци нису у отвореном приступу
- б) Подаци су анонимизирани
- в) Остало, навести шта

5. Доступност података

5.1. Подаци ће бити

- а) јавно доступни
- б) доступни само уском кругу истраживача у одређеној научној области
- ц) затворени

Ако су подаци доступни само уском кругу истраживача, навести под којим условима могу да их користе:

Ако су подаци доступни само уском кругу истраживача, навести на који начин могу приступити подацима:

5.4. Навести лиценцу под којом ће прикупљени подаци бити архивирани.

6. Улоге и одговорност

6.1. Навести име и презиме и мејл адресу власника (аутора) података

6.2. Навести име и презиме и мејл адресу особе која одржава матрицу с подацима

6.3. Навести име и презиме и мејл адресу особе која омогућује приступ подацима другим истраживачима