

Univerza v Ljubljani

Fakulteta za elektrotehniko

Gorazd Karer

**PREDIKTIVNO VODENJE
HIBRIDNIH SISTEMOV**

DOKTORSKA DISERTACIJA

Mentor: prof. dr. Borut Zupančič

Somentor: prof. dr. Igor Škrjanc

Ljubljana, 2009

Zahvala

Doktorska disertacija ne bi mogla nastati, če ne bi bilo nekaterih ljudi, ki so mi pri delu pomagali in mi stali ob strani.

Zahvaljujem se mentorju prof. dr. Borutu Zupančiču in somentorju prof. dr. Igorju Škrjancu za spodbudo, podporo in konstruktivne ideje, pomembno vlogo pa je imel tudi izr. prof. dr. Gašper Mušič, ki je pri nastajanju dela sodeloval že od vsega začetka.

Za sodelovanje sem hvaležen tudi čilskim kolegom – Alfredu Núñezu, Doris Sáez in Javierju Causi.

Zahvala gre vsem sodelavcem iz Laboratorija za modeliranje, simulacijo in vodenje in Laboratorija za avtonomne mobilne sisteme, ki so mi pomagali pri reševanju cele vrste strokovnih, tehničnih in ostalih težav, na katere sem naletel, predvsem pa so zaslužni za to, da moje delo v laboratoriju ni bilo samo navadna služba.

Zahvaljujem se tudi svojim staršem, bratu in prijateljem, nesporno prvo mesto za psihološko (in terminološko) podporo in razumevanje pa gre seveda Mojci. Nenazadnje pa si posebno priznanje v kategoriji osmišljanja življenja, proženja nujno potrebnih možganskih preklopov in zdrave mere pri nagajanju zasluži Živa.

Povzetek

V doktorski disertaciji obravnavamo problematiko s področja *prediktivnega vodenja hibridnih sistemov*. Disertacija je sestavljena iz štirih delov.

V prvem, uvodnem delu na kratko predstavimo problematiko prediktivnega vodenja hibridnih sistemov.

V drugem delu, ki je sestavljen iz več poglavij, obravnavamo modeliranje hibridnih sistemov za prediktivno vodenje.

V drugem poglavju podamo uvod v modeliranje hibridnih sistemov za prediktivno vodenje. Na začetku vpeljemo splošno matematično formulacijo dinamičnih sistemov, ki jo razširimo na formulacijo hibridnih dinamičnih sistemov in formulacijo vodenih hibridnih dinamičnih sistemov. Nato natančneje opišemo prepletanje zvezne in diskretne dinamike pri hibridnih sistemih. V tem okviru obravnavamo diskretne pojave v hibridnih sistemih, tj. samodejne preklope, samodejne skoke, kontrolirane preklope in kontrolirane skoke, ter za vsak pojav predstavimo preprost primer. V nadaljevanju navedemo pomembnejše prispevke k modeliranju hibridnih sistemov, na koncu pa predstavimo še razloge za uporabo modelov hibridnih sistemov za prediktivno vodenje, ki so določeni v diskretnem času.

V tretjem poglavju opišemo osnovni simulacijski model šaržnega reaktorja, na katerem smo preverjali pristope modeliranja in vodenja, ki jih obravnavamo

v disertaciji. Najprej predstavimo zgradbo šaržnega reaktorja, nato pa še njegov matematični model.

V četrtem poglavju predstavimo odsekoma afine in ekvivalentne modele. Opišemo odsekoma afine sisteme, mešane logično-dinamične sisteme, linearne sisteme s komplementarnima spremenljivkama, razširjene linearne sisteme s komplementarnima spremenljivkama ter sisteme diskretnih dogodkov, ki jih modeliramo z operacijami določanja maksimuma in minimuma, seštevanja in skalarnega množenja. Navedemo pogoje za možnost pretvorbe med posameznimi formulacijami. Na koncu opišemo še uporabo in omejitve odsekoma afinih in ekvivalentnih modelov pri prediktivnem vodenju hibridnih sistemov.

V petem poglavju razvijemo hibridni mehki model. Najprej uvedemo problematiko nelinearne dinamike in predstavimo mehki model tipa Takagi-Sugeno. Nadaljujemo z modeliranjem hibridnega mehkega modela. V tem okviru predstavimo hierarhično zgradbo hibridnega mehkega modela, posplošitev mehkega modela tipa Takagi-Sugeno na nelinearne hibridne sisteme, zapis hibridnega mehkega modela in posplošitev na multivariabilne sisteme in na sisteme višjega reda. Na koncu predstavimo še identifikacijo hibridnega mehkega modela, in sicer mehko rojenje, projekcije mehkih rojev v vhodni prostor hibridnega mehkega modela, globalni linearni model in ocenjevanje parametrov hibridnega mehkega modela z modificirano metodo najmanjših kvadratov.

V šestem poglavju obravnavamo postopek modeliranja in identifikacije odsekoma afinega modela na primeru šaržnega reaktorja. Identificiramo štiri različice odsekoma afinega modela, in sicer s hevristično metodo razdelitve prostora ter z razdelitvijo prostora z rojenjem. Parametre modela ocenimo s klasično metodo najmanjših kvadratov ter z modificirano metodo najmanjših kvadratov, ki je predstavljena v poglavju o hibridnem mehkem modelu. S prirejeno metodo za identifikacijo dobimo stabilen odsekoma afini model, ki zadovoljivo opisuje dinamiko šaržnega reaktorja, pri uporabi klasične identifikacijske metode za odsekoma afine sisteme pa ne dobimo stabilnega modela. Odsekoma afine modele na

koncu eksperimentalno testiramo, ovrednotimo natančnost predikcijskega modela in preverimo njeno odvisnost od števila rojev, ki jih določamo pri identifikaciji.

V sedmem poglavju podamo postopek modeliranja in identifikacije hibridnega mehkega modela na primeru šaržnega reaktorja. Identificiramo dve različici hibridnega mehkega modela, in sicer s hevristično metodo razdelitve prostora ter z razdelitvijo prostora z rojenjem. Hibridna mehka modela na koncu eksperimentalno testiramo, ovrednotimo natančnost predikcijskega modela in preverimo njeno odvisnost od števila rojev, ki jih določamo pri identifikaciji. Primerjamo tudi odsekoma afina modela in hibridna mehka modela in ugotovimo, da ob primerljivi kompleksnosti dosežemo večjo natančnost s hibridnim mehkim modelom.

V tretjem delu, ki je sestavljen iz več poglavij, obravnavamo metode prediktivnega vodenja hibridnih sistemov.

V osmem poglavju podamo uvod v metode prediktivnega vodenja hibridnih sistemov. Na začetku opišemo razloge za uveljavitev prediktivnega vodenja v praksi, nato pa predstavimo prediktivno vodenje kot poenostavitev reševanja problema optimalnega vodenja. Obravnavamo uporabo modelov v diskretnem času, delovanje algoritmov za prediktivno vodenje ter določanje kriterijske funkcije in računske zahtevnosti optimizacijskega problema. Na koncu na kratko uvedemo še prediktivno vodenje hibridnih sistemov.

V devetem poglavju obravnavamo metodo za prediktivno vodenje hibridnih sistemov na osnovi reševanja mešanih celoštevilskih optimizacijskih problemov, primerno za sisteme, ki jih lahko zapišemo v obliki odsekoma afinega ali ekvivalentnega modela. Najprej predstavimo dva osnovna pristopa, tj. sprotno in predhodno reševanje optimizacijskega problema. Nato opišemo delovanje algoritma za prediktivno vodenje, potem pa predstavimo še glavno oviro za uporabo metode v praksi, tj. prekletstvo razsežnosti. Pristop preizkusimo na primeru vodenja temperature v šaržnem reaktorju in ugotovimo, da uporaba zaradi kom-

pleksnosti ni smiselna.

V desetem poglavju opišemo metodo za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi, ki deluje na osnovi analize dosegljivosti. Najprej razložimo razvoj problemskega drevesa in predstavimo pristop z analizo dosegljivosti. Nato uvedemo pogoj za ustreznost kriterijske funkcije in predlagamo njeno splošno uporabno obliko. Obravnavamo tudi računsko zahtevnost in pristope za njeno zmanjšanje. Na koncu predstavimo še primer uporabe algoritma za vodenje temperature v šaržnem reaktorju in ovrednotimo obravnavani pristop. Ugotovimo, da algoritem omogoča izpolnitev zahtev vodenja, če je kriterijska funkcija ustrezno določena.

V enajstem poglavju opišemo metodo za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi, ki deluje na osnovi genetskega algoritma. Najprej opišemo uporabo genetskega algoritma pri optimizaciji problemov z diskretnimi spremenljivkami in mehanizem delovanja ter predstavimo genetska operatorja, tj. križanje in mutacijo. Nato obravnavamo suboptimalnost pristopa, kriterijsko funkcijo in računsko zahtevnost ter pristope za njeno zmanjšanje. Na koncu predstavimo še primer uporabe algoritma za vodenje temperature v šaržnem reaktorju in ovrednotimo obravnavani pristop. Glavna slabost genetskega algoritma je suboptimalnost rešitve, glavna prednost pa je neobčutljivost na spremembe reference ter predvidljivost in omejenost časa za izračun rešitve.

V dvanajstem poglavju obravnavamo metodo za adaptivno prediktivno vodenje sistemov s časovno spremenljivimi dinamičnimi lastnostmi, ki deluje na osnovi identifikacije lokalnega linearnega modela vodenega sistema. Najprej opišemo razloge za časovno spremenljive dinamične lastnosti, nato pa predstavimo delovanje algoritma. V tem okviru opišemo linearizirani model, ocenjevanje parametrov in algoritem za prediktivno funkcionalno vodenje. Na koncu predstavimo še primer uporabe algoritma za vodenje temperature v šaržnem reaktorju, v katerem potekata dve eksotermni kemični reakciji, in ovrednotimo obravnavani pristop. Čeprav sta reakciji, ki v model procesa vpeljujeta časovno spreminjajoče para-

metre, izrazito eksotermni, rezultati pokažejo, da uporaba algoritma omogoča izpolnitev zahtev vodenja, in to kljub temu, da kinetičnega modela reakcij ne poznamo.

V trinajstem poglavju razvijemo metodo za vodenje z inverznim modelom, primerno za sisteme z enim izhodom, ki jih lahko zapišemo v obliki hibridnega mehkega modela. Najprej opišemo strukturo algoritma za vodenje, ki ga sestavljata dva dela – krmilni del in regulacijski del. Nato izpeljemo inverzni hibridni mehki model. V nadaljevanju predstavimo delovanje krmilnega dela algoritma za vodenje, in sicer opišemo matriko dosegljivosti hibridnega mehkega modela, prilagoditev referenčnega signala, določitev diskretnega dela krmilnega signala in določitev zveznega dela krmilnega signala. Potem predstavimo še delovanje regulacijskega dela algoritma za vodenje, in sicer opišemo linearizacijo hibridnega mehkega modela, večkoračno predikcijo lineariziranega modela, inkrementalni model in določitev regulirnega signala. Na koncu predstavimo še primer uporabe algoritma za vodenje temperature v šaržnem reaktorju in ovrednotimo obravnavani pristop. Ugotovimo, da temperatura v jedru reaktorja dobro sledi referenčni temperaturi, poleg tega pa algoritem ustrezno kompenzira stopničasto motnjo na izhodu. Tudi temperatura v plašču ostaja znotraj predpisanega temperaturnega območja.

V četrtem, zaključnem delu povzamemo ugotovitve disertacije.

Pri optimizacijskih pristopih za prediktivno vodenje hibridnih sistemov se v realnih primerih pogosto ne moremo izogniti prekletstvu razsežnosti – problem je teoretično sicer rešljiv, v praksi pa ga zaradi omejene računske moči ni mogoče rešiti. Pri vodenju hibridnih sistemov je torej smiselno uporabiti tudi teorijo vodenja z drugih področij in jo s primernimi modifikacijami in ustreznimi poenostavitvami prilagoditi vodenju konkretnega sistema ter se na ta način izogniti algoritmom, ki so za praktično uporabo računsko prezahtevni.

Abstract

In the doctoral thesis we deal with *model predictive control of hybrid systems*. The thesis consists of four parts.

In the first, introductory part we first present the field of model predictive control of hybrid systems.

In the second part, which is made up of several chapters, we deal with modelling of hybrid systems for model predictive control.

In chapter 2 we introduce modelling of hybrid systems for model predictive control. First we present a general mathematical formulation of dynamical systems, which we gradually expand to hybrid dynamical systems and controlled hybrid dynamical systems. Next, we investigate the interlacement of the continuous and the discrete dynamics in hybrid systems. In this regard, we treat the discrete events in hybrid systems, i.e. autonomous switching, autonomous jumps, controlled switching, and controlled jumps. Each type of event is represented by an example. After that, we list some important contributions to modelling of hybrid systems. Finally, we present the motivation for using discrete-time models of hybrid systems for model predictive control.

In chapter 3 we describe the basic simulation model of the batch reactor that was used to verify modelling and control approaches, which are presented

in the thesis. First, we present a scheme of the reactor and then we develop its mathematical model.

Chapter 4 introduces the piecewise affine and equivalent models. We describe the piecewise affine systems, mixed logical dynamical systems, linear complementarity systems, extended linear complementarity systems, and min-max-plus-scaling systems. We state the conditions for transformations among the formulations. Finally, we discuss the use and limitations of the piecewise affine and equivalent models in model predictive control of hybrid systems.

In chapter 5 we develop the hybrid fuzzy model. First we introduce the nonlinear dynamics and present the Takagi-Sugeno fuzzy model. Next, we discuss modelling of the hybrid fuzzy model, namely the hierarchical structure of the hybrid fuzzy model, generalization of the Takagi-Sugeno fuzzy model to nonlinear hybrid systems, the formulation of the hybrid fuzzy model, and generalization to multi-variable systems and higher-than-first-order systems. Finally, we present the identification of the hybrid fuzzy model, namely fuzzy clustering, projection of the fuzzy clusters into the input space of the hybrid fuzzy model, the global linear model, and parameter estimation of a hybrid fuzzy model by means of a modified least-squares method.

In chapter 6 we deal with modelling and identification of a piecewise affine model based on the batch reactor. We develop four varieties of the piecewise affine model, namely by using a heuristic method for space division and by clustering. In addition, we estimate the parameters using the classic and the modified least-squares method. By using the latter method, which is presented in chapter 5 on hybrid fuzzy model, we obtain a stable piecewise affine model that suitably describes the dynamics of the batch reactor. By using the former method, on the other hand, we do not obtain a stable model. Finally, we test the piecewise affine models experimentally, verify the accuracy of the models, and examine its dependence on the number of clusters used in identification.

In chapter 7 we deal with modelling and identification of a hybrid fuzzy model based on the batch reactor. We develop two varieties of the hybrid fuzzy model, namely by using a heuristic method for space division and by using clustering. Finally, we test the hybrid fuzzy models experimentally, verify the accuracy of the models, and examine its dependence on the number of clusters used in identification. We compare the piecewise affine and hybrid fuzzy models. We can conclude that for similar complexity a better accuracy is obtained by the hybrid fuzzy model.

In the third part, which is made up of several chapters, we deal with methods for model predictive control of hybrid systems.

In chapter 8 we introduce the methods for model predictive control of hybrid systems. We start by describing the causes for the significant impact of model predictive control in engineering practice. Next, we present predictive control as a simplification of solving an optimal-control problem. We discuss discrete-time models, the principle of model predictive control, the cost function, and estimation of computational complexity of an optimal-control problem. Finally, we briefly introduce model predictive control of hybrid systems.

In chapter 9 we consider predictive control of hybrid systems based on mixed integer programming, which is suitable for systems that can be formulated as piecewise affine or equivalent models. First, we present two basic approaches, i.e. online and offline solving of the optimization problem. Next, we describe the principle of the control algorithm. After that, we present the main obstacle for using the method, i.e. the curse of dimensionality. The method is tested on the batch-reactor example, where it is ascertained that in this case the method is not usable due to excessive computational complexity.

In chapter 10 describe predictive control of hybrid systems with discrete inputs based on reachability analysis. First, we describe the tree of evolution and present

the reachability-analysis approach. We introduce the condition for cost-function suitability and propose a generally usable form. We also treat computational complexity and the approaches for reducing it. Finally, we verify the method on the batch-reactor example. We establish that the method meets the control demands, provided a suitable cost function is used.

In chapter 11 we deal with predictive control of hybrid systems with discrete inputs based on a genetic algorithm. First, we describe the use of genetic algorithms in optimization of discrete-input control problem, the principle of the genetic algorithm, and the genetic operators, i.e. crossover and mutation. Next, we discuss the suboptimality of the approach and the cost function. We also treat computational complexity and the approaches for reducing it. Finally, we verify the method on the batch-reactor example. The main drawback of the approach is the suboptimality of the solution, whereas the main advantage is its low sensitivity to changes of reference trajectory and boundedness of computation time.

In chapter 12 we deal with self-adaptive predictive control of systems with time-variant dynamics based on the identification of the linear internal model. First, we describe the causes for time-variant dynamics. Next, we present the principle of the algorithm, namely the linearized model, parameter estimation, and predictive-functional-control algorithm. Finally, we verify the method on the batch-reactor example with two exothermic chemical reaction that take place in the core of the reactor. The results suggest that such an implementation meets the control demands, despite the strongly exothermic nature of the chemical reactions with unknown kinetics.

In chapter 13 we develop a control method based on an inverse model. The method is suitable for single-output systems that can be formulated as a hybrid fuzzy model. First, we describe the structure of the algorithm that consists of two parts – the feedforward part and the feedback part. Next, we derive the inverse hybrid fuzzy model. Furthermore, we present the feedforward principle, namely

the reachability matrix, the adjustment of the reference signal, the determination of the discrete part of the feedforward control signal, and the determination of the continuous part of the feedforward control signal. Then, we present the feedback principle, namely the linearization of the hybrid fuzzy model, the prediction of the linearized model, the incremental model, and the determination of the feedback control signal. Finally, we verify the method on the batch-reactor example. We establish that the reference is well followed by the temperature in the core of the reactor. In addition, the algorithm copes well with the additive persistent disturbance in the core. The temperature in the jacket of the reactor also stays in within the prescribed boundaries.

In the fourth, closing part we summarize the conclusions of the thesis.

When using optimization approaches to model predictive control of hybrid systems in practice, we often can not avoid the curse of dimensionality – the problem is theoretically solvable, but can not be tackled in practice due to limited computing power. In hybrid-system control design it is therefore sensible to apply control approaches from other fields and adapting them to the particular system by employing suitable modifications, and appropriate simplifications and in this manner avoid the algorithms that are computationally too demanding for practical implementation.

Vsebina

Seznam slik	xxvii
Seznam tabel	xxxiii
I Uvod	1
1. Uvod	3
II Modeliranje hibridnih sistemov za prediktivno vodenje	9
2. Uvod v modeliranje hibridnih sistemov za prediktivno vodenje	11
2.1 Splošna matematična formulacija	12
2.1.1 Dinamični sistem	12
2.1.2 Hibridni dinamični sistem	13
2.1.3 Vodeni hibridni dinamični sistem	14
2.2 Prepletanje zvezne in diskretne dinamike	16
2.2.1 Samodejni preklopi	17
2.2.2 Samodejni skoki	18
2.2.3 Kontrolirani preklopi	19

2.2.4	Kontrolirani skoki	19
2.3	Pomembnejši prispevki k modeliranju hibridnih sistemov	20
2.3.1	Witsenhausnov model	21
2.4	Modeliranje hibridnih sistemov za prediktivno vodenje v diskretnem času	24
3.	Šaržni reaktor	27
3.1	Zgradba šaržnega reaktorja	27
3.2	Zapis matematičnega modela šaržnega reaktorja z diferencialnimi enačbami	30
3.2.1	Zakon o ohranitvi energije za jedro šaržnega reaktorja . . .	30
3.2.2	Zakon o ohranitvi energije za plašč šaržnega reaktorja . . .	31
4.	Odsekoma afini in ekvivalentni modeli	33
4.1	Odsekoma afnim sistemom ekvivalentne formulacije hibridnih sistemov	33
4.1.1	Odsekoma afini sistemi	34
4.1.2	Mešani logično-dinamični sistemi	37
4.1.3	Linearni sistemi s komplementarnima spremenljivkama . . .	39
4.1.4	Razširjeni linearni sistemi s komplementarnima spremenljivkama	39
4.1.5	Sistemi diskretnih dogodkov, ki jih modeliramo z operacijami določanja maksimuma in minimuma, seštevanja in skalarnega množenja	40
4.1.6	Pretvorba med posameznimi formulacijami modelov	42
4.2	Uporaba pri prediktivnem vodenju hibridnih sistemov	43

4.3	Omejitve odsekoma afinih in ekvivalentnih modelov	45
5.	Hibridni mehki model	47
5.1	Nelinearnost in mehki modeli	47
5.1.1	Mehki model tipa Takagi-Sugeno	48
5.2	Modeliranje hibridnega mehkega modela	50
5.2.1	Hierarhija hibridnega mehkega modela	50
5.2.2	Posplošitev modela tipa Takagi-Sugeno na nelinearne hibridne sisteme	51
5.2.3	Zapis hibridnega mehkega modela	54
5.2.4	Posplošitev hibridnega mehkega modela	56
5.3	Identifikacija hibridnega mehkega modela	58
5.3.1	Mehko rojenje	58
5.3.2	Projekcije mehkih rojev v vhodni prostor hibridnega mehkega modela	60
5.3.3	Globalni linearni model	65
5.3.4	Priprava podatkov za ocenjevanje parametrov hibridnega mehkega modela	65
5.3.5	Ocenjevanje parametrov hibridnega mehkega modela z modificirano metodo najmanjših kvadratov	67
6.	Primer postopka modeliranja in identifikacije odsekoma afinega modela	69
6.1	Podatki za identifikacijo	69
6.2	Razdelitev sistema	70
6.3	Temperatura v jedru šaržnega reaktorja	72

6.3.1	Struktura modela	72
6.3.2	Parametri modela	74
6.4	Temperatura v plašču šaržnega reaktorja	74
6.4.1	Odsekoma afini model – 1. pristop	75
6.4.2	Odsekoma afini model – 2. pristop	80
6.4.3	Odsekoma afini model – 3. pristop	82
6.4.4	Odsekoma afini model – 4. pristop	91
6.5	Vrednotenje	92
6.5.1	Podatki za vrednotenje	93
6.5.2	Eksperimenti	94
6.5.3	Komentar rezultatov	100
7.	Primer postopka modeliranja in identifikacije hibridnega mehkega modela	105
7.1	Signali za identifikacijo	105
7.2	Razdelitev sistema	105
7.3	Temperatura v jedru šaržnega reaktorja	106
7.4	Temperatura v plašču šaržnega reaktorja	107
7.4.1	Hibridni mehki model – 1. pristop	107
7.4.2	Hibridni mehki model – 2. pristop	111
7.5	Vrednotenje	116
7.5.1	Podatki za vrednotenje	116
7.5.2	Eksperimenti	116
7.5.3	Komentar rezultatov	121

III Metode prediktivnega vodenja hibridnih sistemov 125

8. Uvod v metode prediktivnega vodenja hibridnih sistemov	127
8.1 Optimalno vodenje	129
8.1.1 Hamilton-Jacobi-Bellmanova enačba	130
8.1.2 Optimalno vodenje linearnih sistemov	131
8.2 Prediktivno vodenje kot poenostavitev reševanja problema optimalnega vodenja	132
8.3 Uporaba modelov v diskretnem času	133
8.4 Delovanje algoritmov za prediktivno vodenje	133
8.5 Kriterijska funkcija	135
8.6 Računska zahtevnost optimizacijskega problema	136
8.7 Prediktivno vodenje hibridnih sistemov	137
9. Reševanje mešanih celoštevilskih optimizacijskih problemov	139
9.1 Odsekoma afini in ekvivalentni modeli	139
9.2 Sprotno in predhodno reševanje optimizacijskega problema	140
9.3 Delovanje algoritma za prediktivno vodenje	140
9.3.1 Politopi	140
9.3.2 Mešani celoštevilski optimizacijski problem	142
9.4 Prekletstvo razsežnosti	148
9.5 Primer uporabe algoritma za vodenje temperature v šaržnem reaktorju	150
9.5.1 Matematični model šaržnega reaktorja	150
9.5.2 Omejitev nabora mogočih kombinacij vhodnih vektorjev	151

9.5.3	Vodenje	151
9.5.4	Kriterijska funkcija	152
9.5.5	Komentar	152
10.	Prediktivno vodenje na osnovi analize dosegljivosti	155
10.1	Problemsko drevo	155
10.1.1	Razvoj problemskega drevesa	156
10.1.2	Pogoja, ki ju preverjamo med gradnjo problemskega drevesa v vsakem vozlišču	158
10.1.3	Nadaljevanje razvoja problemskega drevesa po izpolnitvi enega od pogojev	159
10.2	Analiza dosegljivosti	160
10.3	Kriterijska funkcija	161
10.3.1	Pogoj za ustreznost kriterijske funkcije	162
10.3.2	Predlog oblike kriterijske funkcije	163
10.4	Računska zahtevnost	165
10.4.1	Zmanjšanje računske zahtevnosti z uporabo analize dosegljivosti	166
10.4.2	Zmanjšanje računske zahtevnosti z omejitvijo nabora mogočih kombinacij vrednosti vektorja vhoda	166
10.4.3	Zmanjšanje računske zahtevnosti z zadrževanjem vrednosti vektorja vhodov skozi več časovnih korakov	167
10.5	Primer uporabe algoritma za vodenje temperature v šaržnem reaktorju	172
10.5.1	Matematični model šaržnega reaktorja	172
10.5.2	Omejitev nabora mogočih kombinacij vhodnih vektorjev	172

10.5.3	Vodenje	173
10.5.4	Kriterijska funkcija	174
10.5.5	Rezultati – 1. pristop	174
10.5.6	Rezultati – 2. pristop	177
10.5.7	Primerjava prediktivnega vodenja s hibridnim mehkim in hibridnim linearnim modelom	179
10.5.8	Komentar	183
11.	Prediktivno vodenje na osnovi genetskega algoritma	187
11.1	Uporaba genetskega algoritma pri optimizaciji problemov z diskre- tnimi spremenljivkami	187
11.2	Mehanizem optimizacije na osnovi genetskega algoritma	188
11.3	Genetski operatorji	189
11.3.1	Križanje	190
11.3.2	Mutacija	191
11.4	Suboptimalnost pristopa	192
11.5	Kriterijska funkcija	193
11.6	Računska zahtevnost	194
11.6.1	Zmanjšanje računske zahtevnosti z omejitvijo nabora mogočih kombinacij vrednosti vektorja vhoda	196
11.6.2	Zmanjšanje računske zahtevnosti z zadrževanjem vrednosti vektorja vhodov skozi več časovnih korakov	197
11.7	Primer uporabe algoritma za vodenje temperature v šaržnem re- aktorju	200
11.7.1	Matematični model šaržnega reaktorja	200

11.7.2	Omejitev nabora mogočih kombinacij vhodnih vektorjev	201
11.7.3	Vodenje	202
11.7.4	Kriterijska funkcija	204
11.7.5	Rezultati	204
11.7.6	Primerjava algoritmov za prediktivno vodenje na osnovi genetskega algoritma, na osnovi analize dosegljivosti in na osnovi ocenjevanja vseh mogočih rešitev	207
11.7.7	Komentar	215
12. Adaptivno prediktivno vodenje z identifikacijo lokalnega linearnega modela		217
12.1	Časovno spremenljive dinamične lastnosti	217
12.2	Delovanje algoritma za adaptivno prediktivno vodenje	218
12.2.1	Zapis lineariziranega modela	218
12.2.2	Ocenjevanje parametrov	219
12.2.3	Algoritem za prediktivno funkcionalno vodenje	223
12.3	Primer uporabe algoritma za vodenje temperature v šaržnem reaktorju	230
12.3.1	Modificirani matematični model šaržnega reaktorja	230
12.3.2	Kinetični model eksotermnih kemičnih reakcij	232
12.3.3	Ocenjevanje parametrov	234
12.3.4	Algoritem za prediktivno funkcionalno vodenje	237
12.3.5	Rezultati	238
12.3.6	Komentar	239
13. Vodenje z inverznim modelom		245

13.1 Dvodelnost algoritma za vodenje z inverznim modelom	245
13.2 Inverzni hibridni mehki model	247
13.3 Krmilni del algoritma za vodenje	248
13.3.1 Matrika dosegljivosti hibridnega mehkega modela	249
13.3.2 Prilagoditev referenčnega signala	250
13.3.3 Določitev diskretnega dela krmilnega signala	252
13.3.4 Določitev zveznega dela krmilnega signala	254
13.4 Regulacijski del algoritma za vodenje	256
13.4.1 Linearizacija hibridnega mehkega modela	257
13.4.2 Večkoračna predikcija lineariziranega modela	258
13.4.3 Inkrementalni model	258
13.4.4 Določitev regulirnega signala	261
13.5 Primer uporabe algoritma za vodenje temperature v šaržnem re- aktorju	263
13.5.1 Matematični model šaržnega reaktorja	263
13.5.2 Vodenje	264
13.5.3 Rezultati	265
13.5.4 Komentar	266
IV Zaključek	269
14. Zaključek	271
14.1 Glavni izvorni prispevki doktorske disertacije	276
Literatura	277

Dodatek A - Parametri modela in vodenja

295

Seznam slik

3.1	Shema šaržnega reaktorja.	28
4.1	Shema pretvorb med modeli: odsekoma afini sistemi (PWA); mešani logično-dinamični sistemi (MLD); linearni sistemi s komplementarnima spremenljivkama (LC); razširjeni linearni sistemi s komplementarnima spremenljivkama (ELC); sistemi diskretnih dogodkov, ki jih modeliramo z operacijami določanja maksimuma in minimuma, seštevanja in skalarnega množenja (MMPS).	44
6.1	Temperatura v jedru T (polna črta) in plašču T_w (prekinjena črta) šaržnega reaktorja.	71
6.2	Temperatura v jedru T (polna črta) in plašču T_w (prekinjena črta) šaržnega reaktorja – izsek.	72
6.3	Vhodni signali k_M , k_H in k_C	73
6.4	Vhodni signali k_M , k_H in k_C – izsek.	74
6.5	Delitev vhodnega prostora modela (6.4).	78
6.6	Temperatura v jedru T (polna črta) in plašču T_w (prekinjena črta) šaržnega reaktorja.	93
6.7	Temperatura v jedru T (polna črta) in plašču T_w (prekinjena črta) šaržnega reaktorja – izsek.	94
6.8	Vhodni signali k_M , k_H in k_C	95
6.9	Vhodni signali k_M , k_H in k_C – izsek.	96

6.10	Odziv modela na vhodne signale za vrednotenje (polna črta) in dejanski odziv (prekinjena črta) – 2. pristop.	97
6.11	Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila predikcijskih korakov h , ki jih računamo v posameznem eksperimentu – 2. pristop. Prekinjena črta označuje vrednost $\bar{J}_{T_w,2}$ pri $h = \infty$	98
6.12	Odziv modela na vhodne signale za vrednotenje (polna črta) in dejanski odziv (prekinjena črta) – 4. pristop.	99
6.13	Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila predikcijskih korakov h , ki jih računamo v posameznem eksperimentu – 4. pristop. Prekinjena črta označuje vrednost $\bar{J}_{T_w,4}$ pri $h = \infty$	100
6.14	Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila rojev, ki jih določamo pri mehkem rojenju pri 4. pristopu.	101
7.1	Pripadnostne funkcije.	109
7.2	Odziv modela na vhodne signale za vrednotenje (polna črta) in dejanski odziv (prekinjena črta) – 1. pristop.	117
7.3	Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila predikcijskih korakov h , ki jih računamo v posameznem eksperimentu – 1. pristop. Prekinjena črta označuje vrednost $\bar{J}_{T_w,1}$ pri $h = \infty$	118
7.4	Odziv modela na vhodne signale za vrednotenje (polna črta) in dejanski odziv (prekinjena črta) – 2. pristop.	119

7.5	Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila predikcijskih korakov h , ki jih računamo v posameznem eksperimentu – 2. pristop. Prekinjena črta označuje vrednost $\bar{J}_{T_w,2}$ pri $h = \infty$	120
7.6	Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila rojev, ki jih določamo pri mehkem rojenju pri 2. pristopu.	122
10.1	Primer razvitega problemskega drevesa za največji predikcijski horizont $H = 4$ in $M = 3$ mogoče različne vrednosti vhodnega vektorja. Končno optimalno vozlišče je V_{14} , zato je v tem koraku izbran in uporabljen vhodni vektor $u_{opt} = u_2$	156
10.2	Primer izseka problemskega drevesa v primeru zadrževanja vrednosti vektorja vhodov skozi $Z = 3$ časovne korake za $M = 3$ mogoče različne vrednosti vhodnega vektorja.	170
10.3	Temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta).	175
10.4	Temperatura v plašču reaktorja T_w in položaji ventilov k_M, k_H in k_C	176
10.5	Temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta).	177
10.6	Temperatura v plašču reaktorja T_w in položaji ventilov k_M, k_H in k_C	178
10.7	Temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta).	180
10.8	Temperatura v plašču reaktorja T_w in položaji ventilov k_M, k_H in k_C	181

10.9	Primerjava uporabe hibridnega mehkega modela (glej razdelek 10.5.6) in hibridnega linearnega modela (glej razdelek 10.5.7): temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta) – izsek.	182
10.10	Primerjava uporabe hibridnega mehkega modela (glej razdelek 10.5.6) in hibridnega linearnega modela (glej razdelek 10.5.7): temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta) – izsek.	183
10.11	Primerjava uporabe hibridnega mehkega modela (glej razdelek 10.5.6) in hibridnega linearnega modela (glej razdelek 10.5.7): temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta) – izsek.	184
11.1	Potek genetskega algoritma.	190
11.2	Temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta).	205
11.3	Temperatura v plašču reaktorja T_w in položaji ventilov k_M , k_H in k_C	206
11.4	Čas t_k , ki ga genetski algoritem porabi za izračun rešitve v enem koraku, in vrednost kriterijske funkcije J za $H_Z = H_{Z,1} = 4$	207
11.5	Čas t_k , ki ga genetski algoritem porabi za izračun rešitve v enem koraku, in vrednost kriterijske funkcije J za $H_Z = H_{Z,2} = 5$	208
11.6	Čas t_k , ki ga genetski algoritem porabi za izračun rešitve v enem koraku, in vrednost kriterijske funkcije J za $H_Z = H_{Z,3} = 6$	209
11.7	Normirani čas \check{t}_k , ki ga algoritem porabi za izračun rešitve v vsakem koraku za $H_Z = H_{Z,1} = 4$	210
11.8	Normirani čas \check{t}_k , ki ga algoritem porabi za izračun rešitve v vsakem koraku za $H_Z = H_{Z,2} = 5$	211

11.9	Normirani čas \check{t}_k , ki ga algoritem porabi za izračun rešitve v vsakem koraku za $H_Z = H_{Z,3} = 6$	213
12.1	Temperatura v jedru T , v plašču T_w in referenčni signal w	239
12.2	Vhodni signali k_M , k_H in k_C	240
12.3	Koncentracije reagentov in toplotna moč.	241
12.4	Ocenjene vrednosti parametrov θ_{11} , θ_{12} in θ_{13}	242
12.5	Ocenjene vrednosti parametrov θ_{21} in θ_{22}	243
13.1	Splošna shema vodenja.	246
13.2	Temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta).	265
13.3	Temperatura v plašču reaktorja T (polna črta) in omejitev temperaturnega območja (prekinjena črta).	266
13.4	Krmilni in regulacijski del signala za položaj mešalnega ventila $k_{M,K}$ in $k_{M,R}$ ter signala za položaj zapornih ventilov k_H in k_C	267

Seznam tabel

3.1	Vhodi šaržnega reaktorja.	28
3.2	Izhodi šaržnega reaktorja.	28
3.3	Ostale fizikalne količine in parametri, ki nastopajo v šaržnem reaktorju.	29
6.1	Rezultati simulacij – 2. pristop.	97
6.2	Rezultati simulacij – 4. pristop.	99
6.3	Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila rojev, ki jih določamo pri mehkem rojenju pri 4. pristopu.	102
7.1	Rezultati simulacij – 1. pristop.	117
7.2	Rezultati simulacij – 2. pristop.	119
7.3	Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila rojev, ki jih določamo pri mehkem rojenju pri 2. pristopu.	123
9.1	Čas t_k , ki ga algoritem porabi za izračun rešitve optimizacijskega problema v enem časovnem koraku.	153
11.1	Primerjava algoritmov za $H_Z = H_{Z,1} = 4$	212
11.2	Primerjava algoritmov za $H_Z = H_{Z,2} = 5$	212
11.3	Primerjava algoritmov za $H_Z = H_{Z,3} = 6$	212
12.1	Parametri kemičnih reakcij.	234

Del I

Uvod

1. Uvod

Veliko procesov, ki jih želimo voditi, ima zvezne in diskretne dinamične lastnosti. Zvezne količine, ki jih navadno opazujemo v sistemu, so temperatura, nivo vode, koncentracija snovi ipd. Diskretne lastnosti pa v sisteme uvajajo npr. zaporni ventili, stikala in omejitve.

Hibridni sistemi so dinamični sistemi, ki vsebujejo tako zvezna kot diskretna stanja ali vhode, zato moramo upoštevati, da lahko signali v takih sistemih vzamejo zvezne ali diskretne vrednosti. Poleg tega se v njih pogosto neločljivo prepletata zvezna in diskretna dinamika. Zvezno dinamiko lahko opišemo z diferencialnimi ali diferenčnimi enačbami, pri čemer je pomembna časovna odvisnost. Diskretno dinamiko pa lahko opišemo npr. s končnimi avtomati, pri čemer je pomembna dogodkovna odvisnost.

Področje hibridnih sistemov je izrazito interdisciplinarno. Že v okviru teorije vodenja obstaja več vej, ki jih lahko obravnavamo z vidika hibridnih sistemov (nelinearno vodenje, adaptivno vodenje, drsni režim vodenja, vodenje z razporejanjem ojačenj itd.). Pravzaprav lahko večino dinamičnih sistemov, ki jih srečujemo vsak dan (avtomobili, računalniki, letala, pralni stroji itd.), obravnavamo tudi s stališča hibridnih sistemov.

Kljub temu pa so v literaturi večinoma obravnavani popolnoma zvezni ali popolnoma diskretni modeli dinamičnih sistemov [137]. To je seveda razumljivo, saj je priporočljivo, da model obravnava le bistvene vidike realnega sistema, poleg tega pa mora biti kar najbolj enostaven – razvoj prekompleksnega modela je namreč nepraktičen in neekonomičen. V preteklosti so se zato pri obravnavi hibri-

dnih sistemov večinoma uporabljale poenostavljene formulacije. Sistematično je bil obravnavan samo zvezni ali samo diskretni vidik sistema. Z razvojem avtomatizacije pa se je pojavila potreba po enotnejši obravnavi hibridnih sistemov. Vedno večja vloga računalnikov pri vodenju procesov, pa tudi razvoj v računalništvu, teoriji vodenja, obravnavi dinamičnih sistemov, matematičnem programiranju in simulacijskih jezikih sta povzročila, da je v zadnjih letih na področju raziskav hibridnih sistemov prišlo do znatnega napredka [7, 89].

Pri mnogih realnih procesih lahko poleg hibridnih dinamičnih lastnosti opazimo tudi izrazito nelinearnost. Uveljavljen pristop k obravnavi nelinearnosti v sistemih je mehka logika, ki predstavlja uporabno izpeljavo teorije mehkih množic za obravnavo realnih kompleksnih problemov. Mehki modeli so učinkoviti univerzalni aproksimatorji nelinearne dinamike, saj lahko z njimi poljubno natančno aproksimiramo zvezne nelinearne funkcije.

Glavna prednost mehkih modelov je intuitivni opis vzročno-posledičnih dinamičnih povezav med vhodi in izhodi v sistemu s stavki *če-potem* (angl. *if-then*). Premisa (stavek *če*) opisuje razdelitev prostora stanj modela na mehke podprostore, ki jih določajo ustrezne pripadnostne funkcije. V posledičnem delu (stavek *potem*) je določen lokalni izhod modela za vsak podprostor posebej. Glede na vrednosti pripadnostnih funkcij lahko izračunamo dejanski izhod sistema oz. napovedani izhod sistema v naslednjem časovnem koraku.

Prediktivno vodenje predstavlja enega izmed naprednih pristopov k vodenju sistemov, ki se je tudi zelo uveljavil v praksi. Najprej se je prediktivno vodenje uporabljalo večinoma v petrokemični industriji, danes pa pridobiva vedno večjo veljavo kot splošno uporaben pristop za vodenje najrazličnejših sistemov.

Prediktivno vodenje je način vodenja dinamičnih sistemov, pri katerem uporabljamo model procesa za napovedovanje obnašanja sistema pri določenih vho-

dih. Optimalne vhode, ki jih uporabimo za dejansko vodenje sistemov, navadno določimo s pomočjo različnih optimizacijskih postopkov. Prediktivni regulatorji so uporabni za vodenje širokega spektra različnih tipov sistemov, zato se predvsem v primerih, ko so vodeni sistemi sorazmerno kompleksni (nelinearni sistemi, multivariabilni sistemi, sistemi z omejitvami, hibridni sistemi idr.), pokaže njihova prednost pred drugimi načini vodenja. Metode prediktivnega vodenja so se uveljavile zaradi enostavnosti, uporabnosti, relativno dobrega delovanja in razumljivosti algoritma.

Prediktivno vodenje danes ni več omejeno samo na počasne procese, kjer je časa za računanje med dvema korakoma oz. spremembama regulirnega signala veliko, ampak se uporaba vedno bolj širi tudi na področje hitrih procesov. Kljub temu pa je pri zahtevnih problemih vodenja, kjer nastopajo kompleksni dinamični procesi, računska kompleksnost še vedno glavna težava.

Načrtovanje prediktivnega vodenja hibridnih sistemov je zaradi prepletanja zvezne in diskretne dinamike v splošnem zelo zahtevna naloga in v literaturi lahko najdemo različne pristope k reševanju tega problema. Večina pristopov se omejuje na hibridne sisteme v diskretnem času, saj s tem zaobidejo več kompleksnih matematičnih pojavov.

Kot smo omenili, je prediktivno vodenje pristop, pri katerem z napovedovanjem obnašanja procesa z uporabo modela določamo ustrezne regulirne signale. Bistveno je modeliranje procesa, ki je zaradi kompleksne hibridne in nelinearne dinamike številnih procesov pogosto težavno. Za tovrstne sisteme klasične metode modeliranja in identifikacije, ki temeljijo na teoriji linearnih sistemov, niso primerne, zato potrebujemo posebne pristope. Kaže se torej potreba po ustreznih formulacijah modelov, ki naj bodo primerni za prediktivno vodenje, poleg tega pa morajo ustrezno upoštevati kompleksnost dinamike sistemov, ki jih želimo voditi. Modeli morajo imeti primerno strukturo, pomembna pa je tudi možnost eksperi-

mentalne identifikacije parametrov, ki jih ne moremo ali jih ni smiselno določiti s teoretičnim modeliranjem.

V literaturi najdemo več različnih formulacij modelov, ki so primerni za uporabo pri prediktivnem vodenju hibridnih sistemov. Najbolj sta se uveljavili formulaciji *odsekoma afinih sistemov* in *mešanih logično-dinamičnih sistemov*, za katere je dokazano, da jih je mogoče (pod določenimi v realnosti ne zelo omejujočimi predpostavkami) medsebojno pretvarjati, mogoča pa je tudi pretvorba v nekaj drugih v literaturi uveljavljenih formulacij hibridnih sistemov.

Glavna prednost sistemov, ki jih lahko zapišemo v obliki odsekoma afinega ali ekvivalentnega modela, je, da lahko prediktivno vodenje tovrstnih sistemov pretvorimo v matematični problem – tj. mešani celoštevilski optimizacijski problem – za katerega obstajajo učinkoviti (tudi komercialni) programi za reševanje optimizacijskih problemov.

Omenjene metode so v osnovi namenjene sistemom z zveznimi vhodi, čeprav lahko rešitve (*multiparametričnih mešanih celoštevilskih linearnih/kvadratičnih programov* (angl. *(multiparametric) mixed integer linear/quadratic programs – (mp)-MILP/MIQP*) uporabimo tudi za sisteme z diskretnimi vhodi. V tem primeru pride do povečanja števila diskretnih spremenljivk, pri čemer je treba upoštevati tudi ustrezne pomožne spremenljivke. Zato računsko zahtevnost eksponentno naraste in tovrstne metode lahko postanejo povsem neuporabne [144].

Večina dosedanjega razvoja na področju prediktivnega vodenja hibridnih sistemov je torej osredotočena na (odsekoma) afine in njim ekvivalentne modele. Tovrstni pristopi pa so lahko neučinkoviti pri vodenju hibridnih sistemov z izrazito izraženimi nelinearnostmi [11]. Odsekoma afini modeli tako za ustrezno natančno aproksimacijo nelinearnosti zahtevajo pretirano segmentacijo, kar v optimizacijski algoritem uvede nove diskretne pomožne spremenljivke, to pa lahko poveča računsko kompleksnost do te mere, da postane algoritem povsem neupo-

raben.

Vzrok za uporabo modelov, ki lahko opisujejo nelinearno dinamiko, za prediktivno vodenje je predvsem možnost izboljšave vodenja zaradi bolj natančne predikcije. Kljub temu pa kriteriji, kdaj bi bilo smiselno uporabiti nelinearni model za prediktivno vodenje, niso vedno jasni. V splošnem se pri regulaciji procesov, pri katerih se delovna točka ne spreminja in se v sistemu ne pojavljajo pretirane motnje, zdi, da uporaba nelinearnih modelov ni najbolj smiselna. Po drugi strani pa lahko pri vodenju procesov, pri katerih se vrednosti vhodov, izhodov in stanj sistema spreminjajo v širokem delovnem območju ali se v sistemu pojavljajo relativno velike motnje, z uporabo nelinearnega modela za prediktivno vodenje dosežemo znatno izboljšavo kakovosti vodenja.

Kaže se torej potreba po formulaciji modela nelinearnega hibridnega sistema, ki bi bil uporaben za prediktivno vodenje. Poleg tega bi bilo treba razviti tudi ustrezne metode za prediktivno vodenje sistemov, ki lahko uporabljajo model nelinearnega hibridnega sistema, pri čemer pa je pomembno zagotoviti čim manjšo računsko zahtevnost teh metod.

Doktorska disertacija je sestavljena iz štirih delov.

V drugem delu, ki je sestavljen iz več poglavij, obravnavamo modeliranje hibridnih sistemov za prediktivno vodenje. V drugem poglavju podamo uvod v modeliranje hibridnih sistemov za prediktivno vodenje. V tretjem poglavju opišemo osnovni simulacijski model šaržnega reaktorja, na katerem smo preverjali pristope modeliranja in vodenja, ki jih obravnavamo v disertaciji. V četrtem poglavju predstavimo odsekoma afine in ekvivalentne modele. V petem poglavju predstavimo hibridni mehki model. V šestem poglavju obravnavamo postopek modeliranja in identifikacije odsekoma afinega modela na primeru šaržnega reaktorja, v sedmem poglavju pa postopek modeliranja in identifikacije hibridnega mehkega modela na primeru šaržnega reaktorja.

V tretjem delu, ki je prav tako sestavljen iz več poglavij, obravnavamo metode prediktivnega vodenja hibridnih sistemov. V osmem poglavju podamo uvod v metode prediktivnega vodenja hibridnih sistemov. V devetem poglavju obravnavamo metodo za prediktivno vodenje hibridnih sistemov na osnovi reševanja mešanih celoštevilskih optimizacijskih problemov, primerno za sisteme, ki jih lahko zapišemo v obliki odsekoma afnega ali ekvivalentnega modela. V desetem poglavju opišemo metodo za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi, ki deluje na osnovi analize dosegljivosti. V enajstem poglavju opišemo metodo za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi, ki deluje na osnovi genetskega algoritma. V dvanaajstem poglavju obravnavamo metodo za adaptivno prediktivno vodenje sistemov s časovno spremenljivimi dinamičnimi lastnostmi, ki deluje na osnovi identifikacije lokalnega linearnega modela vodnega sistema. V trinajstem poglavju predstavimo metodo za vodenje z inverznim modelom, ki je primerna za sisteme z enim izhodom, ki jih lahko zapišemo v obliki hibridnega mehkega modela.

V četrtem, zaključnem delu povzamemo še glavne ugotovitve disertacije.

Del II

Modeliranje hibridnih sistemov za prediktivno vodenje

2. Uvod v modeliranje hibridnih sistemov za prediktivno vodenje

Zaradi kompleksne hibridne dinamike je modeliranje hibridnih sistemov za namen prediktivnega vodenja problem, ki zahteva ustrezno razširitev teorije sistemov. Uveljavljeni zapisi matematičnih modelov (linearnih) sistemov, kot so zapis z diferencialno enačbo, zapis s prenosno funkcijo, zapis v prostoru stanj ipd., v splošnem niso primerni za opisovanje dinamičnih lastnosti kompleksnejših sistemov, kakršni so tudi hibridni sistemi.

Kaže se torej potreba po ustreznih formulacijah modelov, ki bodo primerni za prediktivno vodenje, poleg tega pa morajo ustrezno upoštevati kompleksnost dinamike sistemov, ki jih želimo voditi. Modeli morajo torej imeti primerno strukturo, pomembna pa je tudi možnost eksperimentalne identifikacije parametrov, ki jih ne moremo ali jih ni smiselno določiti s teoretičnim modeliranjem.

V nadaljevanju poglavja bomo opisali splošno matematično formulacijo dinamičnih sistemov, hibridnih dinamičnih sistemov in vodenih hibridnih dinamičnih sistemov. Nato bomo opisali diskretne pojave, ki se odražajo v dinamiki hibridnih sistemov in jih moramo zato pri modeliranju upoštevati.

V naslednjih poglavjih drugega dela doktorske disertacije pa se bomo osredotočili na formulacije matematičnih modelov, ki so primerni za uporabo pri prediktivnem vodenju hibridnih sistemov.

2.1 Splošna matematična formulacija

V tem razdelku bomo predstavili splošno matematično formulacijo dinamičnih sistemov, ki jo bomo postopoma razširili na formulacijo hibridnih dinamičnih sistemov in formulacijo vodenih hibridnih dinamičnih sistemov [40].

2.1.1 Dinamični sistem

Dinamični sistem Σ lahko splošno zapišemo, kot kaže enačba (2.1) [124].

$$\Sigma = [X, \Gamma, \phi] \quad (2.1)$$

V enačbi 2.1 predstavlja X poljuben *prostor stanj* dinamičnega sistema Σ .

Γ predstavlja *polgrupo* z enoto.

Polgrupa $\Gamma = \{a, b, \dots\}$ je množica, nad katero je definirana asociativna operacija $\circ : \Gamma \times \Gamma \rightarrow \Gamma$, ki vsakemu urejenemu paru $(a, b) \in \Gamma$ priredi natanko en element $a \circ b \in \Gamma$. Operacija \circ mora zadoščati naslednjima pogojevma.

- Zaprtost: za vsak $a, b \in \Gamma$ velja $a \circ b \in \Gamma$.
- Asociativnost: za vsak $a, b, c \in \Gamma$ velja $(a \circ b) \circ c = a \circ (b \circ c)$.

V našem primeru je Γ monoid, kar pomeni, da, poleg tega, da je polgrupa, vsebuje tudi enoto, tj. element e , ki zadošča naslednjima pogojevma.

- $e \circ e = e$.
- Za vsak $a \in \Gamma$ velja $e \circ a = a \circ e = a$.

V enačbi 2.1 predstavlja ϕ *funkcijo* $\phi : X \times \Gamma \rightarrow X$, ki zadošča naslednjim pogojem.

- Enota e : za vsak $x \in X$ velja $\phi(x, e) = x$.

- Polgrupa: za vsak $x \in X$ in poljubna $a, b \in \Gamma$ velja

$$\phi(\phi(x, a), b) = \phi(x, a \circ b).$$
- Zveznost: za vsako okolico W točke $\phi(x, a)$ obstajata okolica U točke x in okolica V elementa a , tako da velja $\phi(U, V) \subset W$.

2.1.2 Hibridni dinamični sistem

Hibridni sistem lahko obravnavamo kot indeksirano zbirko dinamičnih sistemov z vključenim opisom za *preskakovanje* med posameznimi dinamičnimi sistemi, tj. dejanskim *preklapljanjem* med sistemi ali *ponastavljanjem* stanja [40]. Preskok se zgodi, ko stanje zadosti določenemu pogoju, tj. ko pride v določen podprostor prostora stanj.

Hibridni sistem lahko torej razumemo kot niz dinamičnih sistemov z začetnimi in končnimi stanji, kjer se vsakič, ko je doseženo določeno končno stanje, zgodi *preskok*. Preskok pa pomeni prehod v (v splošnem spremenjeno) začetno stanje v (v splošnem spremenjenem) dinamičnem sistemu.

Splošni hibridni dinamični sistem H lahko formalno podamo, kot je zapisano v enačbi (2.2).

$$H = [Q, \Sigma, \mathbf{A}, \mathbf{G}] \quad (2.2)$$

V enačbi (2.2) predstavlja Q množico indeksov, ki določajo zbirko dinamičnih sistemov. Te indekse imenujemo *diskretna stanja* hibridnega dinamičnega sistema.

$\Sigma = \{\Sigma_q\}_{q \in Q}$ označuje *zbirko dinamičnih sistemov* (glej razdelek 2.1.1), ki sestavljajo hibridni dinamični sistem. Vsak $\Sigma_q = [X_q, \Gamma_q, \phi_q]$ označuje en dinamični sistem, ki je označen z indeksom (diskretnim stanjem) q . V tem primeru predstavljajo X_q *prostore zveznih stanj*, ϕ_q *zvezno dinamiko* hibridnega dinamičnega sistema.

$\mathbf{A} = \{A_q\}_{q \in Q}$, pri čemer velja $A_q \subset X_q$ za vsak $q \in Q$, označuje zbirko *množic samodejnih preskokov*.

$\mathbf{G} = \{G_q\}_{q \in Q}$, pri čemer velja $G_q : A_q \rightarrow \bigcup_{q \in Q} X_q \times \{q\}$, predstavlja zbirko *funkcij samodejnih preskokov*, ki opisujejo *diskretno dinamiko* hibridnega dinamičnega sistema.

Glede na zgoraj zapisano lahko opišemo *prostor stanj hibridnega dinamičnega sistema* H , kot kaže enačba (2.3).

$$S = \bigcup_{q \in Q} X_q \times \{q\} \quad (2.3)$$

Primer delovanja hibridnega dinamičnega sistema H lahko opišemo na sledeč način.

1. Privzamemo, da je začetno stanje sistema $s_0 = (x_0, q_0) \in S \setminus A$.
2. Dinamiko sistema v tem dinamičnem podsistemu določa funkcija $\phi_{q_0}(x_0, *_{\Gamma_{q_0}})$, kjer je $*_{\Gamma_{q_0}} \in \Gamma_{q_0}$.
3. Če se zgodi, da stanje sistema zaide v podprostor A_{q_0} , tj. v točko $\bar{s}_1 = (\bar{x}_1, q_0)$, sistem preskoči – glede na funkcijo preskokov $G_{q_0}(\bar{x}_1) = (x_1, q_1)$ – v novo stanje $s_1 = (x_1, q_1)$.
4. Dinamiko sistema v tem dinamičnem podsistemu določa funkcija $\phi_{q_1}(x_1, *_{\Gamma_{q_1}})$, kjer je $*_{\Gamma_{q_1}} \in \Gamma_{q_1}$ itd.

2.1.3 Vodeni hibridni dinamični sistem

V razdelku 2.1.2 smo obravnavali *avtonomni* splošni hibridni dinamični sistem, ki nima možnosti vpliva na dinamiko preko zunanjih vhodov. V primeru tovrstnih sistemov seveda ne moremo govoriti o vodenju, zato bomo v tem razdelku definicijo razširili na vodeni splošni hibridni dinamični sistem.

Vodeni splošni hibridni dinamični sistem H_v lahko torej formalno podamo, kot je zapisano v enačbi (2.4).

$$H_v = [Q, \Sigma, \mathbf{A}, \mathbf{G}, \mathbf{V}, \mathbf{C}, \mathbf{F}] \quad (2.4)$$

Pomen elementov Q in \mathbf{A} , ki nastopata v zvezi s formulacijo vodenega splošnega hibridnega dinamičnega sistema, ki je podana v enačbi (2.4), je opisan v razdelku 2.1.2.

$\Sigma = \{\Sigma_q\}_{q \in Q}$ označuje zbirko vodenih dinamičnih sistemov (glej razdelek 2.1.1), ki sestavljajo vodeni hibridni dinamični sistem. Vsak $\Sigma_q = [X_q, \Gamma_q, \phi_q, U_q]$ označuje en vodeni dinamični sistem, ki je označen z indeksom (diskretnim stanjem) q .

V tem primeru predstavlja U_q množico za vodenje dinamičnega sistema Σ_q . Množica U_q določa parametrizacijo zvezne dinamike ϕ_q v posameznem dinamičnem podsistemu vodenega hibridnega dinamičnega sistema.

$\mathbf{G} = \{G_q\}_{q \in Q}$, pri čemer velja $G_q : A_q \times V_q \rightarrow \bigcup_{q \in Q} X_q \times \{q\}$, predstavlja zbirko funkcij samodejnih preskokov.

V tem primeru predstavlja $\mathbf{V} = \{V_q\}_{q \in Q}$ zbirko množic za vodenje samodejnih preskokov. Množica V_q določa parametrizacijo funkcij samodejnih preskokov G_q v posameznem dinamičnem podsistemu vodenega hibridnega dinamičnega sistema.

$\mathbf{C} = \{C_q\}_{q \in Q}$, pri čemer velja $C_q \subset X_q$, označuje zbirko množic vodenih preskokov.

$\mathbf{F} = \{F_q\}_{q \in Q}$, pri čemer velja $F_q : C_q \rightarrow 2^S$ pomeni zbirko funkcij vodenih preskokov.

Primer delovanja vodenega hibridnega dinamičnega sistema H_v lahko opišemo na sledeč način.

1. Privzamemo, da je začetno stanje sistema $s_0 = (x_0, q_0) \in S \setminus A$.

2. Dinamiko sistema v tem vodenem dinamičnem podsistemu določa funkcija $\phi_{q_0}(*x_0, *r_0, u)$, kjer je $*x_0 \in X_0$ in $*r_0 \in \Gamma_0$.
3. Če se zgodi, da stanje sistema zaide v enega od podprostorov A_{q_0} ali C_{q_0} , tj. v točko $\bar{s}_1 = (\bar{x}_1, q_0)$, imamo več možnosti.
 - Če je stanje sistema zašlo v podprostor A_{q_0} , potem sistem nujno preskoči – glede na funkcijo preskokov $G_{q_0}(\bar{x}_1, v)$ in izbrani parameter $v \in V_{q_0}$ – v novo stanje.
 - Če je stanje sistema zašlo v podprostor C_{q_0} , potem lahko izberemo, ali se bo preskok zgodil ali ne. Če izberemo, da se bo preskok zgodil, lahko izberemo cilj iz zaloge vrednosti $F_{q_0}(\bar{x}_1)$.

V vsakem primeru po preskoku ima vodeni hibridni dinamični sistem novo stanje, tj. točko $s_1 = (x_1, q_1)$.

4. Dinamiko sistema v tem vodenem dinamičnem podsistemu določa funkcija $\phi_{q_1}(*x_1, *r_1, u)$, kjer je $*x_1 \in X_1$ in $*r_1 \in \Gamma_1$ itd.

2.2 Prepletanje zvezne in diskretne dinamike

Kot smo že zapisali, se v hibridnih sistemih pojavljata zvezna in diskretna dinamika, ki sta pogosto prepleteni in vplivata druga na drugo, tako da ju ne moremo preprosto ločiti. V tem razdelku bomo pri opisu lastnosti hibridnega sistema izhajali s stališča, ki je značilno za področje avtomatike. Hibridni sistem bomo torej podali kot zvezni sistem, v katerem pa opišemo tudi določene diskretne pojave.

Dinamiko hibridnega sistema lahko z inženirskega vidika opišemo z diferencialno enačbo (2.5), ki obravnava dogajanje v hibridnem sistemu v odvisnosti od časa.

$$\begin{aligned} \dot{x}(t) &= \xi(t) \\ &\text{za } t \geq 0 \end{aligned} \tag{2.5}$$

V enačbi (2.5) je $x(t)$ zvezno stanje, $\xi(t)$ pa označuje vektorsko polje, ki je v splošnem odvisno od zveznega stanja $x(t)$ in diskretnih pojavov v hibridnem sistemu, v primeru vodenega hibridnega sistema pa tudi od vhoda $u(t)$.

Diskretne pojave, ki nastopajo v hibridnih sistemih, lahko razdelimo, kot je navedeno spodaj [40]:

1. samodejni preklopi;
2. samodejni skoki;
3. kontrolirani preklopi;
4. kontrolirani skoki.

Navedene diskretne pojave bomo podrobneje opisali v nadaljevanju.

2.2.1 Samodejni preklopi

Samodejni prekop je pojav, pri katerem se *vektorsko polje* $\xi(\cdot)$ *nezvezno spremeni*. To se zgodi takrat, ko zvezno stanje sistema $x(\cdot)$ doseže določen podprostor, kar sproži samodejni prekop.

Primer hibridnega sistema s samodejnim preklopom

Za primer vzemimo voziček z maso m , ki ga potiskamo s silo $F(t)$ po poti x . Trenje in ostale morebitne izgube zanemarimo. Pot je ravna do točke x_1 , nato pa se nadaljuje v hrib s konstantnim naklonskim kotom α . Gravitacijski pospešek označimo z g .

Za voziček lahko zapišemo diferencialno enačbo, kot kaže (2.6).

$$\ddot{x}(t) = \begin{cases} \frac{F(t)}{m} & ; \quad x \leq x_1 \\ \frac{F(t)}{m} - g \cdot \sin(\alpha) & ; \quad x > x_1 \end{cases} \quad (2.6)$$

2.2.2 Samodejni skoki

Samodejni skok je pojav, pri katerem se *zvezno stanje* $x(\cdot)$ *nezvezno spremeni*. To se zgodi takrat, ko zvezno stanje sistema $x(\cdot)$ doseže določen podprostor, kar sproži samodejni skok.

Primer hibridnega sistema s samodejnim skokom

Za primer vzemimo žogo z maso m , ki se odbija od vodoravnih tal pod vplivom gravitacije. Žogo spustimo z začetne višine nad tlemi x_0 , trenutno višino pa označimo z x . Gravitacijski pospešek označimo z g . Koeficient izgub pri odboju označimo z $\rho \in [0, 1]$. Zračni upor in ostale morebitne izgube zanemarimo.

Za žogo lahko zapišemo diferencialni enačbi (2.7) in (2.8) ter pogoj za samodejni skok, kot kaže enačba (2.9).

$$\dot{x}(t) = v(t) \quad (2.7)$$

$$\dot{v} = -g \quad (2.8)$$

$$x(\tau) = 0 \Rightarrow v(\tau^+) = -\rho \cdot v(\tau^-) \quad (2.9)$$

za vsak $t = \tau \geq 0$

2.2.3 Kontrolirani preklopi

Kontrolirani prekllop je pojav, pri katerem se *vektorsko polje* $\xi(\cdot)$ *nezvezno spremeni*. To se zgodi kot odziv na vodenje, kar pomeni, da kontrolirani prekllop sprožimo od zunaj z ustreznim vhodnim signalom.

Primer hibridnega sistema s kontroliranim preklpom

Za primer vzemimo model satelita z reakcijskimi motorji, ki skrbijo za regulacijo vrtenja satelita okrog lastne osi. Obravnavali bomo samo poenostavljen model v eni dimenziji. Z ω označimo kotno hitrost, J pomeni vztrajnostni moment, T pa absolutno vrednost navora, ki ga povzročajo reakcijski motorji, ki so lahko uga-snjeni ($m = 0$), delujejo s polno močjo v pozitivni ($m = 1$) ali v negativni smeri ($m = -1$). Diskretni vhod m lahko torej zavzame 3 vrednosti $m \in \{-1, 0, 1\}$.

Za satelit lahko zapišemo diferencialno enačbo, kot kaže (2.10).

$$\ddot{\omega}(t) = \frac{T}{J}m(t) \quad (2.10)$$

2.2.4 Kontrolirani skoki

Kontrolirani skok je pojav, pri katerem se *zvezno stanje* $x(\cdot)$ *nezvezno spremeni*. To se zgodi kot odziv na vodenje, kar pomeni, da kontrolirani skok sprožimo od zunaj z ustreznim vhodnim signalom.

Primer hibridnega sistema s kontroliranim skokom

Za primer vzemimo polnjenje posode s konstantnim presekom A . Nivo tekočine v posodi označimo s h . Posoda ima na dnu odtok, preko katerega se prazni s pretokom $\phi_{ven}(t)$, ki je odvisen od nivoja tekočine v posodi. V posodo priteka tekočina s konstantnim dotokom ϕ_{noter} . Poleg tega v posodo v trenutkih $\tau_1, \tau_2, \dots, \tau_n$ dolijemo polna vedra tekočine s konstantnim volumnom V .

Za posodo lahko zapišemo diferencialno enačbo, kot kaže (2.11).

$$\dot{h} = \frac{1}{A} \left(\phi_{noter} - \phi_{ven}(t) + \sum_{i=1}^n \delta(t - \tau_i) \cdot V \right) \quad (2.11)$$

V enačbi (2.11) označuje $\delta(t - \tau_i)$ Diracov impulz.

2.3 Pomembnejši prispevki k modeliranju hibridnih sistemov

Splošna in poenotena teorija, ki bi celovito obravnavala hibridne sisteme in njihovo modeliranje, (še) ne obstaja, zato pa se v literaturi pojavljajo mnogi bolj ali manj uspešni poskusi formalizacije modeliranja hibridnih sistemov.

Tako lahko naletimo na različne formulacije modelov v zveznem ali diskretnem času. Nekateri modeli samo delno zajemajo hibridno dinamiko (glej razdelka 2.1 in 2.2), nekateri pa so bolj celoviti. Nekateri so bolj primerni za simulacije, drugi za analizo, tretji pa so primerni za vodenje hibridnih sistemov. Vsekakor še vedno ne moremo govoriti o eni sami formulaciji modela hibridnega sistema, ki bi zadostil vsem potrebam.

Prvi poskus zajema hibridne dinamike v enotnem modelu je objavil Witsenhausen v [142] v šestdesetih letih prejšnjega stoletja. Kljub temu se je področje hibridnih sistemov začelo ponovno pospešeno razvijati šele sredi devetdesetih let prejšnjega stoletja. Glavni razlog za ponovno zanimanje so verjetno tehnične inovacije, posebno na področju vgrajenih sistemov. Njihova značilnost je, da je diskretno-logična naprava "vgrajena" v zvezno dinamično okolje, pri čemer sta naprava in okolje medsebojno povezana in vplivata drug na drugega. Poleg tega so se šele v zadnjem času pojavili ustrezni programski paketi za simulacijo ter numerično in simbolično računanje, ki so omogočili teoretični napredek [21].

Spodaj izpostavljamo nekatere pomembnejše modele hibridnih sistemov, na katere v literaturi pogosteje naletimo [3, 77, 21, 74, 58, 70, 4, 2, 5, 6, 40]:

- Witsenhausnov model [142];
- Taverninijev model [134];
- Back-Guckenheimer-Myersov model [13];
- Nerode-Kohnov model [105];
- Antsaklis-Stiver-Lemonov model [8];
- Brockettov model [41].

2.3.1 Witsenhausnov model

Kot smo omenili, je bil prvi poskus zajema hibridne dinamike v enotnem modelu objavljen v [142]. To delo predstavlja začetek razvoja področja hibridnih sistemov kot samostojne veje v okviru teorije sistemov, zato bomo Witsenhausnov model tudi podrobneje opisali v tem razdelku.

Witsenhausnov model opisuje razred sistemov, definiranih v zveznem času, ki imajo deloma zvezno in deloma diskretno stanje, tj. hibridne sisteme v zveznem času. Razred hibridnih sistemov, ki ga opisuje Witsenhausnov model, je omejen z naslednjimi pravili.

1. Pri preskoku, tj. v trenutku, v katerem se diskretno stanje sistema spremeni, mora ostati vektor zveznih stanj zvezen. To pomeni, da model ne obravnava nezveznih ponastavitev vektorja zveznih stanj, tj. skokov – glej razdelka 2.2.2 in 2.2.4 – ampak samo preklope.
2. Preskok se lahko zgodi če in samo če vektor zveznih stanj zadosti pogoju za določen preskok. To pomeni, da lahko z vhodnimi signali na preskoke vplivamo samo posredno, tj. preko diferencialnih enačb. Model torej ne obravnava kontroliranih preklopov – glej razdelek 2.2.3.

3. Model ima tudi nekatere tehnične zahteve v zvezi s podatki, s katerimi sistem opisujemo.

Witsenhausnov model izhaja iz opisa zveznega sistema, ki ga podaja enačba (2.12).

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t)) \\ x(t) &\in \mathbb{R}^n \text{ in } u(t) \in \mathbb{R}^m\end{aligned}\tag{2.12}$$

Vpliv diskretnega dela na dinamiko vpelje z vključitvijo *diskretnega stanja* $q(t)$, kot kaže enačba (2.13).

$$\dot{x}(t) = f(q(t), x(t), u(t))\tag{2.13}$$

Hibridno stanje s Witsenhausnovnega modela lahko opišemo s parom diskretnega stanja q in zveznega stanja x , kot kaže enačba (2.14).

$$\begin{aligned}s &= (q, x) \\ s \in S &\equiv Q \times \mathbb{R}^n\end{aligned}\tag{2.14}$$

Omenimo, da je množica diskretnih stanj $Q \subset \mathbb{Z}$ končna $m(Q) < \infty$.

Povezava dinamike v nasprotno smer pa je podana z opisom pogojev za zvezno stanje, pri katerih se zgodi sprememba diskretnega stanja. Preklop s stanja $q = i$ na $q = j$ ($i \neq j$) se sproži, ko zvezno stanje x zaide v podmnožico preklopa $M_{i,j} \subset \mathbb{R}^n$. Za vsak urejen par indeksov diskretnih stanj $i, j \in Q$ obstaja podmnožica preklopa $M_{i,j} \subset \mathbb{R}^n$, ki pa je lahko tudi prazna.

Enačba (2.15) določa *množico prihodov*.

$$M_i^+ \equiv \bigcup_{j \neq i} M_{j,i}\tag{2.15}$$

Množica prihodov M_i^+ je unija vseh vrednosti zveznega stanja x , ki predstavljajo pogoj za proženje preklopa diskretnega stanja v $q = i$, in sicer iz vseh ostalih diskretnih stanj $q = j \neq i$.

Podobno je v enačbi (2.16) določena *množica odhodov*.

$$M_i^- \equiv \bigcup_{j \neq i} M_{i,j} \quad (2.16)$$

Množica odhodov M_i^- je unija vseh vrednosti zveznega stanja x , ki predstavljajo pogoj za proženje preklopa iz diskretnega stanja $q = i$ v katero koli drugo diskretno stanje $q = j \neq i$.

Omenjene množice morajo izpolnjevati tri predpostavke, ki so navedene spodaj.

1. Za katere koli tri različne indekse $i, j, k \in Q$ velja $M_{i,j} \cap M_{i,k} = \emptyset$. Ta predpostavka pomeni, da se ne more nikoli zgoditi, da bi bil naenkrat izpolnjen pogoj za prekop na dve ali več različnih diskretnih stanj.
2. Za vsak $i \in Q$ je množica M_i^- zaprta v \mathbb{R}^n .
3. Za vsak $i \in Q$ velja $M_i^- \cap M_i^+ = \emptyset$. Ta predpostavka zagotavlja, da se po preklopu diskretnega stanja ne more takoj zgoditi nov prekop na novo diskretno stanje. Na ta način so preprečene *preklopovne zanke*, tj. pojav, ko se v istem trenutku zgodi neskončno prekopov.

Witsenhausnov model poleg tega predpostavlja, da je vsaka funkcija $f(q, x, u)$ zvezna glede na (x, u) pri konstantnem q in zvezno odvedljiva glede na x pri konstantnem (q, u) .

Množica za vodenje je definirana, kot sledi. Naj bosta t_f spremenljiv končni čas, tako da velja $t_f \in [t_0, \infty)$, in $\Omega \subset \mathbb{R}^m$ zaprta množica. Vodenje predstavlja par $[t_f, u(\cdot)]$, pri čemer je $u(\cdot)$ poljubna odsekoma zvezna funkcija na intervalu $[t_0, t_f]$, ki ima zalogo vrednosti v Ω .

Delovanje Witsenhausnovega hibridnega modela, od začetnega stanja $s_0 = (x_0, q_0)$ v času t_0 do končnega stanja $s_f = (x_f, q_f)$ v času t_f , lahko torej opišemo na sledeč način.

1. Privzamemo, da je začetno stanje sistema $s_0 = (x_0, q_0) \in S$.
2. Dinamiko sistema torej določa funkcija $\dot{x}(t) = f(q_0, x(t), u(t))$, kjer je $x(t_0) = x_0$. V primeru, da trajektorija x ne doseže podprostora M_0^- , ostane diskretno stanje $q(t) = q_0$ nespremenjeno, in sicer vse do končnega časa t_f .
3. Če se zgodi, da trajektorija x zaide v enega od podprostorov iz množice $M_{q_0}^-$, recimo v M_{q_0, q_1} . Denimo, da se to zgodi v točki x_1 v trenutku t_1 . Tedaj diskretno stanje preklopi s stare vrednosti $q(t_1^-) = q_0$ na novo $q(t_1^+) = q_1$. Definirajmo novo stanje sistema v trenutku t_1 : $s(t_1) = (x_1, q_1)$. Dodajmo, da mora veljati $t_1 > t_0$, sicer bi kot začetno stanje morali obravnavati (x_0, q_1) .
4. Velja torej $x_1 \in M_{q_1}^+$ in – glede na predpostavko v 3. točki – $x_1 \notin M_{q_1}^-$. Zaradi tega lahko ponovimo prejšnje korake – s to razliko, da tokrat začnemo v stanju (x_1, q_1) . Poleg tega lahko rečemo, da, če obstaja trenutek t_2 , v katerem se zgodi prekop diskretnega stanja, velja $t_2 > t_1$.

Če torej rekurzivno uporabimo zgornja pravila, lahko dobimo enoličen potek stanj Witsenhausnovega modela hibridnega sistema $[x(\cdot), q(\cdot)]$.

2.4 Modeliranje hibridnih sistemov za prediktivno vodenje v diskretnem času

Načrtovanje prediktivnega vodenja hibridnih sistemov je zaradi prepletanja zvezne in diskretne dinamike zelo zahtevno in v literaturi lahko najdemo različne pristope. Večina pristopov se omejuje na hibridne sisteme, ki jih modeliramo v

diskretnem času, saj se s tem izognemo več kompleksnim matematičnim problemom.

V doktorski disertaciji se bomo torej podrobneje posvetili modelom hibridnih sistemov, ki so definirani v diskretnem času. Zvezne dinamične lastnosti sistemov, ki jih želimo opisati s tovrstnimi modeli, lahko opišemo z diferenčnimi enačbami, diskretne pa podajajo končni avtomati, oboji pa so sinhronizirani na isto uro, tj. diskretni časovni korak.

Znano je, da lahko s primernim časom vzorčenja izgube pri opisovanju dinamike sistema, ki so posledica samega vzorčenja, pri večini aplikacij zanemarimo. Kljub temu pa lahko ugotovimo, da se nekateri zanimivi pojavi, ki so značilni za modele hibridnih sistemov, ki so definirani v zveznem času, v diskretnem času ne pojavijo. Takšen je na primer Zenonov fenomen [75].

Po drugi strani pa so takšni pojavi večinoma posledica preklpov v modelu, ki je definiran v zveznem času, in se v dejanskem realnem sistemu sicer sploh ne pojavljajo. Glavna prednost zapisov modelov v diskretnem času pa je njihova uporaba pri reševanju optimizacijskih problemov in ostalih matematičnih manipulacijah, ki bi bile v zveznem času precej bolj zahtevne, kar pa bi imelo za posledico precej večjo računsko zahtevnost algoritmov za vodenje in s tem tudi potrebo po bolj zmogljivi (dražji) strojni opremi. Zaradi tega bi lahko postali ti algoritmi v praksi povsem neuporabni.

V naslednjih poglavjih drugega dela doktorske disertacije se bomo torej osredotočili na formulacije matematičnih modelov, ki so primerni za uporabo v prediktivnem vodenju hibridnih sistemov.

3. Šaržni reaktor

Pristope modeliranja in vodenja, ki jih podrobneje obravnavamo v doktorski disertaciji, bomo preizkusili na simulacijskem primeru šaržnega reaktorja. V eksperimentih bo uporabljen model realnega šaržnega reaktorja, ki je postavljen v farmacevtskem podjetju v Sloveniji in se uporablja v proizvodnji zdravil.

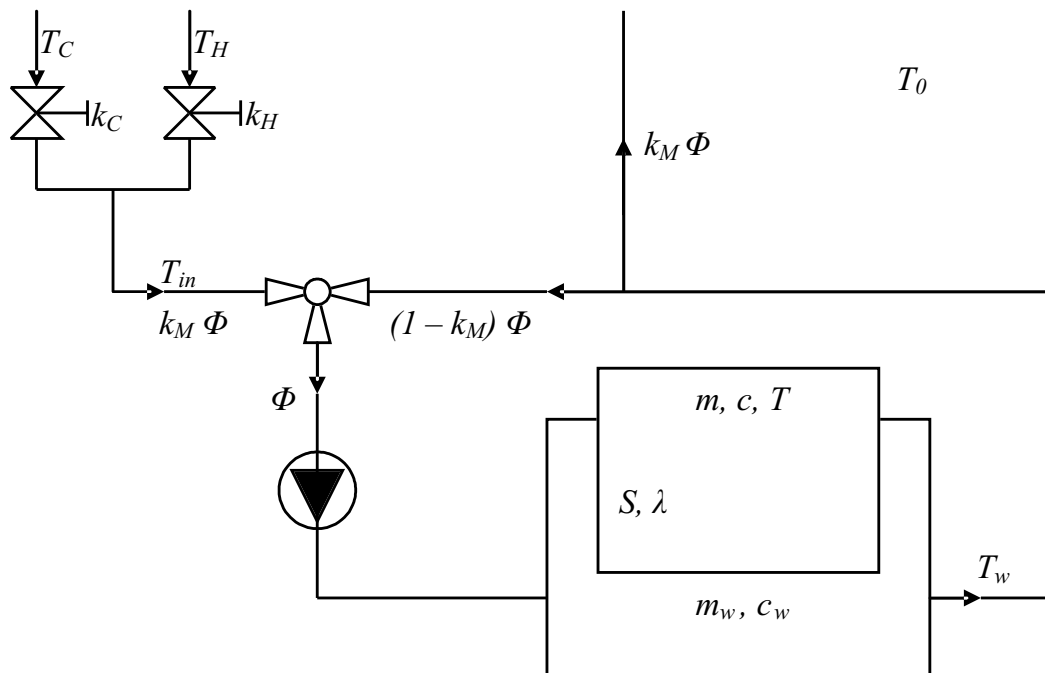
Cilj vodenja je regulacija temperature sestavin, ki se mešajo v jedru reaktorja, kjer nastaja končni produkt. Da bi dosegli ustrezno kakovost izhodnega produkta, moramo zagotoviti, da temperatura čim bolj sledi referenčni trajektoriji, ki je podana s predpisanim receptom.

3.1 Zgradba šaržnega reaktorja

Na sliki 3.1 je prikazana shema šaržnega reaktorja. Jedro reaktorja (temperatura T) se segreva oz. ohlaja skozi plašč reaktorja (temperatura T_w). Toplotni medij v plašču je mešanica sveže vhodne vode, ki vstopa v reaktor skozi zaporna ventila, in povratne vode. Voda v plašču ima stalen pretok ϕ .

Dinamika šaržnega reaktorja je odvisna od njegovih fizičnih lastnosti: mase m in specifične toplote c sestavin v jedru oz. plašču reaktorja (indeks w označuje plašč), koeficienta toplotne prevodnosti λ , površine toplotnega stika S in konstantne temperature okolice T_0 .

T_{in} označuje temperaturo sveže vhodne vode, T_C oz. T_H pa temperaturo hladne oz. vroče vhodne vode. k_C in k_H sta poziciji zapornih ventilov za hladno oz. vročo vhodno vodo, k_M pa za mešalni ventil.



Slika 3.1: Shema šaržnega reaktorja.

OZNAKA	POMEN
k_M	položaj mešalnega ventila
k_C	položaj zapornega ventila za hladno vodo
k_H	položaj zapornega ventila za hladno vodo

Tabela 3.1: Vhodi šaržnega reaktorja.

OZNAKA	POMEN
T	temperatura v jedru šaržnega reaktorja
T_w	temperatura v plašču šaržnega reaktorja
T_0	temperatura okolice

Tabela 3.2: Izhodi šaržnega reaktorja.

Temperatura sveže vhodne vode T_{in} je odvisna od dveh vhodov, tj. pozicije zapornih ventilov k_H in k_C . Mogoča sta dva načina delovanja: če je $k_C = 1$ in $k_H = 0$, je vhodna voda hladna ($T_{in} = T_C$), če pa je $k_C = 0$ in $k_H = 1$, je vhodna

OZNAKA	POMEN
T_0	temperatura okolice
T_{in}	temperatura sveže vhodne vode
T_C	temperatura hladne vode
T_H	temperatura vroče vode
ϕ	pretok vode v plašču šaržnega reaktorja
m	masa sestavin v jedru šaržnega reaktorja
c	specifična toplota sestavin v jedru šaržnega reaktorja
m_w	masa vode v plašču šaržnega reaktorja
c_w	specifična toplota vode v plašču šaržnega reaktorja
λ	prevodnost toplotnega stika med jedrom in plaščem šaržnega reaktorja
S	površina toplotnega stika med jedrom in plaščem šaržnega reaktorja
λ_0	prevodnost toplotnega stika med plaščem šaržnega reaktorja in okolico
S_0	površina toplotnega stika med plaščem šaržnega reaktorja in okolico

Tabela 3.3: Ostale fizikalne količine in parametri, ki nastopajo v šaržnem reaktorju.

voda vroča ($T_{in} = T_H$). Razmerje med vhodno in povratno vodo določa mešalni ventil k_M .

V tabeli 3.1 so zbrani vhodni signali, v tabeli 3.2 izhodni signali, v tabeli 3.3 pa ostale fizikalne količine in parametri, ki nastopajo v šaržnem reaktorju.

Obravnavamo torej multivariabilni sistem s tremi vhodi (k_M , k_H in k_C) in

dvema merljivima izhodoma (T in T_w). Zaradi konstrukcijskih lastnosti je časovna konstanta temperature vode v plašču reaktorja veliko krajša kot v jedru, zato govorimo o togem sistemu.

3.2 Zapis matematičnega modela šaržnega reaktorja z diferencialnimi enačbami

Matematični model šaržnega reaktorja lahko opišemo z diferencialnimi enačbami, ki jih izpeljemo iz termodinamičnega zakona o ohranitvi energije. Toplotne toke v šaržnem reaktorju lahko razdelimo tako:

- toplotna kondukcija med jedrom in plaščem šaržnega reaktorja;
- toplotna kondukcija med plaščem šaržnega reaktorja in okolico;
- toplotna konvekcija zaradi dovoda sveže vhodne vode v plašč šaržnega reaktorja;
- toplotna konvekcija zaradi odvoda vode iz plašča šaržnega reaktorja.

Zaradi jasnosti bomo posebej obravnavali matematična podmodela za jedro in za plašč šaržnega reaktorja. Na ta način bomo kompleksni multivariabilni sistem celotnega šaržnega reaktorja razdelili na dva preprostejša podsistema z več vhodi in po enim izhodom.

3.2.1 Zakon o ohranitvi energije za jedro šaržnega reaktorja

Izmed prej naštetih toplotnih tokov v šaržnem reaktorju vpliva na temperaturo v jedru šaržnega reaktorja T samo toplotna kondukcija med jedrom in plaščem šaržnega reaktorja. Iz zakona o ohranitvi energije lahko torej izpeljemo diferencialno enačbo za jedro šaržnega reaktorja (3.1).

$$mc \frac{dT}{dt} = \lambda S(T_w - T) \quad (3.1)$$

3.2.2 Zakon o ohranitvi energije za plašč šaržnega reaktorja

Na temperaturo v plašču šaržnega reaktorja T_w vplivajo vsi prej naštetih toplotni toki v šaržnem reaktorju. Iz zakona o ohranitvi energije lahko torej izpeljemo diferencialno enačbo za plašč šaržnega reaktorja (3.2).

$$\begin{aligned} m_w c_w \frac{dT_w}{dt} &= k_M \phi c_w T_{in} + (1 - k_M) \phi c_w T_w - \phi c_w T_w - \\ &\quad - \lambda S(T_w - T) - \lambda_0 S_0(T_w - T_0) = \\ &= k_M \phi c_w T_{in} - k_M \phi c_w T_w - \\ &\quad - \lambda S(T_w - T) - \lambda_0 S_0(T_w - T_0) \end{aligned} \quad (3.2)$$

4. Odsekoma afini in ekvivalentni modeli

V literaturi lahko najdemo več različnih formulacij modelov, ki so primerni za uporabo pri prediktivnem vodenju hibridnih sistemov. Večina dosedanjega razvoja na področju prediktivnega vodenja hibridnih sistemov je osnovana na *odsekoma afinih* (angl. *piecewise affine* – PWA) [129] in njim ekvivalentnih sistemih.

4.1 Odsekoma afinim sistemom ekvivalentne formulacije hibridnih sistemov

V literaturi, ki obravnava hibridne sisteme (in tudi njihovo prediktivno vodenje), se veliko pozornosti posveča različnim formulacijam modelov, ki pa v osnovi na različne načine opisujejo dinamiko hibridnih linearnih oz. afinih sistemov. Za nekatere od teh formulacij je dokazano [72, 18, 113], da jih je mogoče (pod določenimi v realnosti ne zelo omejujočimi predpostavkami) medsebojno pretvarjati v druge formulacije modelov. To pomeni, da so vse te formulacije (v okviru predpostavk) medsebojno ekvivalentne.

Sistemi, o katerih govorimo in jih bomo v nadaljevanju podrobneje opisali, so naštetih spodaj:

- *odsekoma afini sistemi* (angl. *piecewise affine* – PWA) [129];
- *linearni sistemi s komplementarnima spremenljivkama* (angl. *linear complementarity* – LC) [73, 138];

- *razširjeni linearni sistemi s komplementarnima spremenljivkama* (angl. *extended linear complementarity – ELC*) [56, 57, 55];
- *sistemi diskretnih dogodkov, ki jih modeliramo z operacijami določanja maksimuma in minimuma, seštevanja in skalarnega množenja* (angl. *max-min-plus-scaling – MMPS*) [57];
- *mešani logično-dinamični sistemi* (angl. *mixed logical dynamical – MLD*) [32].

Vsaka od teh formulacij ima svoje prednosti. Za odsekoma afine sisteme so bili predlagani stabilnostni kriteriji [76], za mešane logično-dinamične sisteme obstajajo pristopi za vodenje in verifikacijo [28, 32, 35], za linearne sisteme s komplementarnima spremenljivkama poznamo pogoje za obstoj in enoličnost trajektorij, ki predstavljajo rešitev sistema [73, 138], itd.

Ekvivalentnost formulacij nam omogoča, da lahko nekatera orodja in pristope, ki so bili razviti za modeliranje, analizo ali vodenje sistemov, katerih dinamiko opisuje posamezna od naštetih formulacij modela, prenesemo in uporabimo tudi v okviru ene od tej formulaciji ekvivalentne formulacije.

V nadaljevanju bomo vsako od omenjenih formulacij podrobneje opisali in navedli morebitne predpostavke, ki pogojujejo medsebojno pretvorbo.

4.1.1 Odsekoma afni sistemi

Odsekoma afni sistemi [129] so sistemi, pri katerih lahko razdelimo prostor stanj in vhodov na podprostore. V vsakem podprostoru dinamiko opišemo z afnim sistemom.

Odsekoma afni sistemi predstavljajo enega najpreprostejših zapisov, ki v zadostni meri posplošuje klasični zapis linearnega sistema v prostoru stanj – glej enačbo (4.1) – da lahko teoretično s poljubno natančnostjo opiše nelinearno di-

namiko in upošteva hibridne značilnosti procesa, zato se tudi pogosto pojavlja v literaturi.

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) \\y(k) &= Cx(k) + Du(k)\end{aligned}\tag{4.1}$$

V enačbi (4.1) označuje $x(k) \in \mathbb{R}^n$ stanje sistema, $u(k) \in \mathbb{R}^m$ vhod v sistem, $y(k) \in \mathbb{R}^l$ pa izhod sistema. S k je označen trenutni časovni korak sistema. Matrike $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{l \times n}$ in $D \in \mathbb{R}^{l \times m}$ določajo dinamiko linearnega sistema.

Z razširitvijo zapisa linearnega sistema v diskretnem času, kot ga kaže enačba (4.1), lahko izpeljemo zapis odsekoma afinega sistema. Podajata ga enačbi (4.2) in (4.3).

$$\begin{aligned}x(k+1) &= A_i x(k) + B_i u(k) + f_i \\y(k) &= C_i x(k) + D_i u(k) + g_i\end{aligned}\tag{4.2}$$

$$\begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \Omega_i\tag{4.3}$$

V enačbi (4.2) označuje $x(k) = [x_z(k)^T x_b(k)^T]^T$ stanje sistema. Tu je $x_z(k)$ zvezni del stanja, $x_b(k)$ pa binarni del stanja sistema. Podobno velja za vhod v sistem $u(k) = [u_z(k)^T u_b(k)^T]^T$. Tu je $u_z(k)$ zvezni del, $u_b(k)$ pa binarni del vhoda v sistem. Tudi izhod sistema $y(k) = [y_z(k)^T y_b(k)^T]^T$ je podobno definiran: $y_z(k)$ je zvezni del, $y_b(k)$ pa binarni del izhoda iz sistema.

Matrike sistema A_i , B_i , C_i in D_i ter vektorja f_i in g_i določajo afino dinamiko v posameznem podprostoru prostora stanj in vhodov, ki je označen z indeksom $i \in \mathbb{Z} < \infty$. Če sta vektorja f_i in g_i enaka 0, govorimo o odsekoma linearnem sistemu.

Ustrezne dimenzije matrik in vektorjev so določene v enačbah (4.4), (4.5) in (4.6).

$$\begin{aligned} x_z(k) &\in \mathbb{R}^{n_z}, \quad x_b(k) \in \{0, 1\}^{n_b} \\ u_z(k) &\in \mathbb{R}^{m_z}, \quad u_b(k) \in \{0, 1\}^{m_b} \\ y_z(k) &\in \mathbb{R}^{l_z}, \quad y_b(k) \in \{0, 1\}^{l_b} \end{aligned} \quad (4.4)$$

$$\begin{aligned} A_i &\in (\mathbb{R}^{n_z} \times \{0, 1\}^{n_b}) \times (\mathbb{R}^{n_z} \times \{0, 1\}^{n_b}) \\ B_i &\in (\mathbb{R}^{n_z} \times \{0, 1\}^{n_b}) \times (\mathbb{R}^{m_z} \times \{0, 1\}^{m_b}) \\ C_i &\in (\mathbb{R}^{l_z} \times \{0, 1\}^{l_b}) \times (\mathbb{R}^{n_z} \times \{0, 1\}^{n_b}) \\ D_i &\in (\mathbb{R}^{l_z} \times \{0, 1\}^{l_b}) \times (\mathbb{R}^{m_z} \times \{0, 1\}^{m_b}) \end{aligned} \quad (4.5)$$

$$\begin{aligned} f_i &\in \mathbb{R}^{n_z} \times \{0, 1\}^{n_b} \\ g_i &\in \mathbb{R}^{l_z} \times \{0, 1\}^{l_b} \end{aligned} \quad (4.6)$$

Posamezen podprostor $\Omega_i \subset (\mathbb{R}^{n_z} \times \{0, 1\}^{n_b}) \times (\mathbb{R}^{m_z} \times \{0, 1\}^{m_b})$ v enačbi (4.3) predstavlja konveksen polieder v prostoru stanj in vhodov. Vsak polieder je določen s sistemom neenačb v okviru prostora stanj in vhodov, kot kaže enačba (4.7).

$$\Omega_i \equiv \left\{ \begin{bmatrix} x(\cdot) \\ u(\cdot) \end{bmatrix} ; \begin{array}{l} H_i x(\cdot) + J_i u(\cdot) \leq K_i, \\ \tilde{H}_i x(\cdot) + \tilde{J}_i u(\cdot) < \tilde{K}_i \end{array} \right\} \quad (4.7)$$

V enačbi (4.7) označujejo H_i , J_i , K_i in \tilde{H}_i , \tilde{J}_i , \tilde{K}_i matrike, ki določajo podprostor Ω_i . Število vrstic v matrikah ustreza številu neenačb, tj. številu ploskev, ki omejujejo podprostor Ω_i .

Omenimo, da se v praksi zaradi numeričnih težav pri določanju poliedrov s sistemom neenačb znak $<$ nadomesti z \leq . Poliedri so v tem primeru določeni, kot

kaže enačba (4.8), kjer $\varepsilon \in \mathbb{R}^+$ določa majhno pozitivno število, ki ga izberemo glede na numerično natančnost algoritma.

$$\Omega_i \equiv \left\{ \begin{array}{l} \left[\begin{array}{c} x(\cdot) \\ u(\cdot) \end{array} \right]; \quad \begin{array}{l} H_i x(\cdot) + J_i u(\cdot) \leq K_i, \\ \tilde{H}_i x(\cdot) + \tilde{J}_i u(\cdot) \leq \tilde{K}_i + \varepsilon \end{array} \end{array} \right\} \quad (4.8)$$

4.1.2 Mešani logično-dinamični sistemi

Mešani logično-dinamični sistemi [32] predstavljajo razred sistemov, v katerem je v formulaciji vključena logika, dinamika in omejitve. V zapisu torej nastopajo diferencialne enačbe, logična pravila ter omejitve in medsebojne povezave, ki so podane z izjavnim računom.

Zapis mešanega logično-dinamičnega sistema v diskretnem času podajata enačbi (4.9), (4.10) in neenačba (4.11).

$$x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \quad (4.9)$$

$$y(k) = Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \quad (4.10)$$

$$E_1x(k) + E_2u(k) + E_3\delta(k) + E_4z(k) \leq g_5 \quad (4.11)$$

V enačbah (4.9), (4.10) in neenačbi (4.11) označuje $x(k)$ stanje sistema, $u(k)$ vhod v sistem, $y(k)$ pa izhod sistema. Vektorji so sestavljeni, kot je opisano v razdelku 4.1.1. Ustrezne dimenzije vektorjev so določene v enačbi (4.4).

Vektorja $z(k)$ in $\delta(k)$ sta pomožni spremenljivki. Ustrezne dimenzije vektorjev so določene v enačbi (4.12).

$$\begin{aligned} z(k) &\in \mathbb{R}^{r_z} \\ \delta(k) &\in \{0, 1\}^{r_\delta} \end{aligned} \quad (4.12)$$

S k je označen trenutni časovni korak sistema.

Matrike sistema $A, B_1, B_2, B_3, C_i, D_1, D_2, D_3, E_1, E_2, E_3$ in E_4 ter vektor g_5 določajo dinamiko mešanega logično-dinamičnega sistema.

Neenačbe, ki so podane v neenačbi (4.11), interpretiramo po posameznih komponentah.

Privzamemo, da za vsako stanje $x(k)$, vhod v sistem $u(k)$ in pomožni spremenljivki $z(k)$ in $\delta(k)$, kot so definirani zgoraj in ki zadostujejo neenačbi (4.11), velja, da sta stanje v naslednjem časovnem koraku $x(k+1)$ in izhod $y(k)$, ki ju določata enačbi (4.9) in (4.10), taka, da drži $x_b(k+1) \in \{0, 1\}^{n_b}$ in $y_b(k) \in \{0, 1\}^{l_b}$.

Zgornjo trditev lahko privzamemo brez izgube splošnosti, saj lahko binarna dela stanja in izhoda sistema $x_b(k)$ in $y_b(k)$ (če sploh obstajata) obravnavamo tudi kot pomožne spremenljivke – glej dokaz v [28]. Če npr. binarni del izhoda sistema $y_b(k) \in \{0, 1\}^{l_b}$ ni ustrezno vključen v neenačbe (4.11), potem lahko uvedemo pomožno spremenljivko $\delta_y(k) \in \{0, 1\}^{l_b}$ in neenačbe, kot jih podaja (4.13).

$$\begin{aligned} [Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k)]_b - \delta_y(k) &\leq 0, \\ [-Cx(k) - D_1u(k) - D_2\delta(k) - D_3z(k)]_b + \delta_y(k) &\leq 0 \end{aligned} \tag{4.13}$$

Označitev $[]_b$ pomeni, da obravnavamo samo vrstice iz neenačbe (4.11), ki se nanašajo na binarni del izhoda sistema $y_b(k)$. Vključitev neenačbe (4.13) uvede torej izenačitev $y_b(k) = \delta_y(k) \in \{0, 1\}^{l_b}$.

Na podoben način lahko uvedemo pomožne spremenljivke za binarni del vhoda sistema $u_b(k)$ in binarni del stanja sistema v naslednjem časovnem koraku $x_b(k+1)$.

4.1.3 Linearni sistemi s komplementarnima spremenljivkama

Linearni sistemi s komplementarnima spremenljivkama [73, 138] vključujejo v formulacijo komplementarni spremenljivki.

Zapis linearnega sistema s komplementarnima spremenljivkama v diskretnem času podajajo enačbe (4.14), (4.15), (4.16) in (4.17).

$$x(k+1) = Ax(k) + B_1u(k) + B_2w(k) \quad (4.14)$$

$$y(k) = Cx(k) + D_1u(k) + D_2w(k) \quad (4.15)$$

$$v(k) = E_1x(k) + E_2u(k) + E_3w(k) + g_4 \quad (4.16)$$

$$0 \leq v(k) \perp w(k) \geq 0 \quad (4.17)$$

Vektorji in matrike, ki nastopajo v enačbah (4.14), (4.15), (4.16) in (4.17), so definirani podobno kot v razdelkih 4.1.1 oz. 4.1.2. Vektorja $v(k) \in \mathbb{R}^s$ in $w(k) \in \mathbb{R}^s$ imenujemo komplementarni spremenljivki in sta medsebojno pravokotna: $v(k) \perp w(k)$, kar pomeni, da je $v(k)^T \cdot w(k) = 0$.

4.1.4 Razširjeni linearni sistemi s komplementarnima spremenljivkama

Razširjeni linearni sistemi s komplementarnima spremenljivkama [56, 57, 55] lahko v diskretnem času podamo z enačbami (4.18), (4.19), (4.20) in (4.21).

$$x(k+1) = Ax(k) + B_1u(k) + B_2d(k) \quad (4.18)$$

$$y(k) = Cx(k) + D_1u(k) + D_2d(k) \quad (4.19)$$

$$E_1x(k) + E_2u(k) + E_3d(k) \leq g_4 \quad (4.20)$$

$$\sum_{i=1}^p \prod_{j \in \phi_i} (g_4 - E_1x(k) - E_2u(k) - E_3d(k))_j = 0 \quad (4.21)$$

Vektorji in matrice, ki nastopajo v enačbah (4.18), (4.19), (4.20) in (4.21), so definirani podobno kot v razdelkih 4.1.1 oz. 4.1.2. Vektor $d(k) \in \mathbb{R}^r$ je pomožna spremenljivka.

Pogoj v enačbi (4.21) lahko zapišemo, kot je podano v enačbi (4.22), saj velja pogoj v neenačbi (4.20).

$$\prod_{j \in \phi_i} (g_4 - E_1x(k) - E_2u(k) - E_3d(k))_j = 0 \quad (4.22)$$

za vsak $i \in \{1, 2, \dots, p\}$

To pomeni, da lahko neenačbo (4.20) in enačbo (4.21) obravnavamo kot sistem linearnih neenačb, ki ga razdelimo na p skupin linearnih neenačb, tj. po eno skupino za vsako množico ϕ_i , tako da je v vsaki skupini vsaj ena neenačba degenerirana v enačbo.

4.1.5 Sistemi diskretnih dogodkov, ki jih modeliramo z operacijami določanja maksimuma in minimuma, seštevanja in skalarnega množenja

Sistemi diskretnih dogodkov, ki jih modeliramo z operacijami določanja maksimuma in minimuma, seštevanja in skalarnega množenja [57] predstavljajo še enega od mogočih zapisov hibridnih sistemov.

Pri tem zapisu predstavljajo osnovo *operacije za maksimizacijo, minimizacijo, seštevanje in skalarno množenje* (angl. *max-min-plus-scaling expressions*), ki jih

označimo s f in so definirane nad spremenljivkami x_1, x_2, \dots, x_n , kot kaže enačba (4.23).

$$f := x_i |\alpha| \max(f_k, f_l) |\min(f_k, f_l)| f_k + f_l |\beta| f_k$$

kjer je $i \in \{1, 2, \dots, n\}$ (4.23)

in $\alpha, \beta \in \mathbb{R}$

V enačbi (4.23) sta tudi f_k in f_l operaciji za maksimizacijo, minimizacijo, seštevanje in skalarno množenje.

Primer operacije za maksimizacijo, minimizacijo, seštevanje in skalarno množenje podaja enačba (4.24).

$$5x_1 - 3x_2 + 7 + \max(\min(2x_1, -8x_2), x_2 - 3x_3) \quad (4.24)$$

Zapis sistema diskretnih dogodkov, ki jih modeliramo z operacijami določanja maksimuma in minimuma, seštevanja in skalarnega množenja, v diskretnem času podajajo enačbe (4.25), (4.26) in (4.27).

$$x(k+1) = \mathcal{M}_x(x(k), u(k), d(k)) \quad (4.25)$$

$$y(k) = \mathcal{M}_y(x(k), u(k), d(k)) \quad (4.26)$$

$$\mathcal{M}_c(x(k), u(k), d(k)) \leq c \quad (4.27)$$

V enačbah (4.25), (4.26) in (4.27) predstavljajo \mathcal{M}_x , \mathcal{M}_y in \mathcal{M}_c operacije določanja maksimuma in minimuma, seštevanja in skalarnega množenja nad elementi vektorjev $x(k)$, $u(k)$ in $d(k)$. $d(k)$ je pomožna spremenljivka.

4.1.6 Pretvorba med posameznimi formulacijami modelov

Kot smo že omenili, je zgoraj opisane formulacije modelov mogoče (pod določenimi v realnosti ne zelo omejujočimi predpostavkami) medsebojno pretvarjati v druge formulacije modelov [72]. To pomeni, da so vse te formulacije (v okviru predpostavk) medsebojno ekvivalentne.

Pogoji v predpostavkah, ki omogočajo nekatere pretvorbe, se nanašajo na:

- *dobro postavljenost*¹ (angl. *well-posedness*),
- *omejitev vhodov*,
- *omejitev stanj*,
- *omejitev izhodov* in
- *omejitev pomožnih spremenljivk*.

V nadaljevanju bomo navedli možne pretvorbe med modeli in morebitne predpostavke, ki pogojujejo te pretvorbe.

1. Vsak *mešani logično-dinamični sistem* lahko zapišemo kot *linearni sistem s komplementarnima spremenljivkama*.
2. Vsak *linearni sistem s komplementarnima spremenljivkama* lahko zapišemo kot *mešani logično-dinamični sistem*, če so elementi vektorjev $w(K)$ in $v(k)$ omejeni.
3. Vsak *linearni sistem s komplementarnima spremenljivkama* lahko zapišemo kot *razširjeni linearni sistem s komplementarnima spremenljivkama*.
4. Vsak dobro postavljen *odsekoma afini sistem* lahko zapišemo kot *mešani logično-dinamični sistem*, če je množica dosegljivih stanj in vhodov omejena.

¹Dobra postavljenost sistema je pogoj za obstoj in enoličnost trajektorij rešitev.

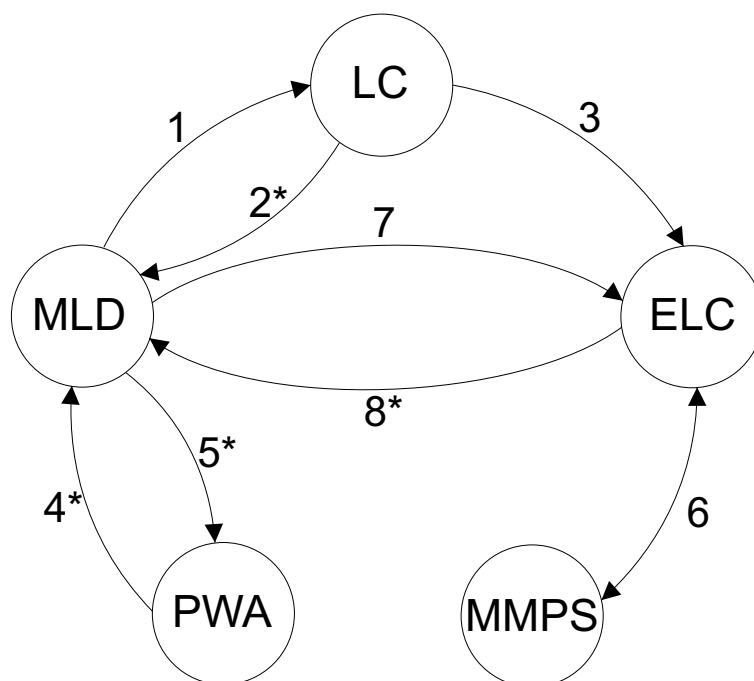
5. Vsak popolnoma dobro postavljen *mešani logično-dinamični sistem* lahko zapišemo kot *odsekoma afini sistem*.
6. *Sistem diskretnih dogodkov, ki jih modeliramo z operacijami določanja maksimuma in minimuma, seštevanja in skalarnega množenja* je popolnoma ekvivalenten *razširjenemu linearnemu sistemu s komplementarnima spremenljivkama*.
7. Vsak *mešani logično-dinamični sistem* lahko zapišemo kot *razširjeni linearni sistem s komplementarnima spremenljivkama*.
8. Vsak *razširjeni linearni sistem s komplementarnima spremenljivkama* lahko zapišemo kot *mešani logično-dinamični sistem*, če so elementi vektorja $(g_4 - E_1x(k) - E_2u(k) - E_3d(k))$ omejeni.

Za formalne dokaze posameznih zgoraj navedenih pretvorb glej [72].

Na sliki 4.1 je prikazana shema pretvorb med modeli, ki smo jih navedli zgoraj. Puščice ponazarjajo zgoraj navedene pretvorbe in so označene z zaporedno številko iz seznama pretvorb. Puščica iz enega v drugi model pomeni, da je izhodiščni model vključen v ciljnem modelu. Zvezdica (*) poleg številke, ki označuje puščico, pomeni, da je pretvorba mogoča, če so izpolnjeni nekateri pogoji, ki so navedeni v seznamu pretvorb pod ustrezno zaporedno številko.

4.2 Uporaba pri prediktivnem vodenju hibridnih sistemov

Za namene prediktivnega vodenja je najbolj razširjena uporaba *odsekoma afinih sistemov* in *mešanih logično-dinamičnih sistemov*. Formulaciji, ki omogočata popoln opis dinamike hibridnih linearnih oz. afinih sistemov, predstavljata osnovo za implementacijo teorije prediktivnega vodenja, ki temelji na reševanju mešanih celoštevilskih optimizacijskih problemov. Pristop je podprt z zaokroženo teorijo modeliranja in vodenja in se je že večkrat uporabil tudi v praktičnih primerih.



Slika 4.1: Shema pretvorb med modeli: odsekoma afini sistemi (PWA); mešani logično-dinamični sistemi (MLD); linearni sistemi s komplementarnima spremenljivkama (LC); razširjeni linearni sistemi s komplementarnima spremenljivkama (ELC); sistemi diskretnih dogodkov, ki jih modeliramo z operacijami določanja maksimuma in minimuma, seštevanja in skalarnega množenja (MMPS).

Poleg tega postopke modeliranja oz. zapisa sistema olajšujejo tudi nekateri programski paketi oz. jeziki, kot sta npr. *Hysdel* [136] in *Hybrid Identification Toolbox* [62].

Glavna prednost sistemov, ki so zapisani v obliki odsekoma afinih in ekvivalentnih modelov, je, da lahko prediktivno vodenje tovrstnih sistemov zapišemo kot matematični problem, za katerega obstajajo učinkoviti komercialni programi za reševanje optimizacijskih problemov. Za natančnejši opis glej 9. poglavje.

4.3 Omejitve odsekoma afinih in ekvivalentnih modelov

Modeliranje odsekoma afinih sistemov je razmeroma preprost postopek, če lahko teoretično določimo razdelitev vhodnega prostora odsekoma afinega modela in pripadajoče afine podmodele. V literaturi pa naletimo tudi na različne pristope za identifikacijo odsekoma afinih modelov, kjer obravnavamo sistem kot črno škatlo [108, 63, 120, 104, 141, 139]. Glavno težavo pri uporabi predlaganih pristopov predstavlja računski zahtevnost, posebno v primerih, ko imamo opravka s kompleksnimi sistemi, ki izkazujejo izrazito nelinearno in hibridno dinamiko.

Večina dosedanjega razvoja na področju prediktivnega vodenja hibridnih sistemov je osnovana na odsekoma afinih in njim ekvivalentnih modelih. Tovrstni pristopi pa so lahko neučinkoviti pri vodenju hibridnih sistemov z izrazitimi nelinearnostmi [116].

Odsekoma afini modeli teoretično omogočajo poljubno natančno aproksimacijo nelinearnosti. Kljub temu pa pri sistemih z izrazito nelinearno dinamiko zahtevajo pretirano segmentacijo, tj. delitev prostora stanj sistema na veliko število podprostorov, v katerih dinamiko opišemo z afinim modelom.

Postopek modeliranja, s katerim dobimo zadovoljiv model sistema z izrazito nelinearno dinamiko v odsekoma afini obliki, je zaradi delitve prostora stanj na veliko število podprostorov lahko precej zapleten in dolgotrajen. Poleg tega je treba ustrezno določiti meje med posameznimi podprostori, kar pa večinoma ni preprosta naloga.

Problem velike segmentacije je tudi, da v optimizacijski algoritem uvede nove diskretne zunanje spremenljivke. To lahko poveča računsko kompleksnost do te mere, da postane algoritem povsem neuporaben.

Zaradi tega se kaže potreba po formulaciji modela nelinearnega hibridnega sistema, ki bo uporaben za prediktivno vodenje. Poskus formulacije modela hibridnega sistema, ki bi ustrezal navedenim kriterijem, je predstavljen v 5. poglavju.

5. Hibridni mehki model

5.1 Nelinearnost in mehki modeli

Pri mnogih realnih industrijskih procesih lahko poleg ostalih kompleksnih dinamičnih lastnosti opazimo tudi izrazite nelinearnosti. Uveljavljen pristop k obravnavi nelinearnosti v sistemih je mehka logika [9, 12, 132, 131].

Mehka logika izhaja iz teorije mehkih množic [145], ki v klasično teorijo *ostrih* množic uvaja koncept *stopnje pripadnosti* [10]. Element torej množici ne samo *pripada* ali *ne pripada*, ampak ji pripada z določeno *stopnjo pripadnosti*, ki lahko zavzame vrednost med 0 (kar pomeni *ne pripada*) in 1 (kar pomeni *pripada*).

Na ta način lahko v teorijo množic vključimo tudi nekatere prislove, ki približno izražajo mero, kot so *rahlo*, *zelo*, *precej*, *zmerno* ipd. Poleg tega mehke množice uvajajo tudi lingvistične spremenljivke. Te lahko zavzamejo določeno lingvistično vrednost, interpretiramo pa jih lahko v obliki mehkih množic oz. pripadnostnih funkcij.

Prednost mehkih modelov je tudi intuitivni opis vzročno-posledičnih dinamičnih povezav med vhodi in izhodi v sistemu s stavki *če-potem* (angl. *if-then*) [67].

Bistvena lastnost mehkih modelov je, da so učinkoviti univerzalni aproksimatorji nelinearne dinamike [46, 68]. Dokazano je, da lahko z njimi poljubno natančno aproksimiramo vsako zvezno nelinearno funkcijo.

Mehka logika in z njo mehki modeli tako predstavljajo uporabno izpeljavo teorije mehkih množic za obravnavo realnih kompleksnih dinamičnih sistemov.

5.1.1 Mehki model tipa Takagi-Sugeno

Nelinearne dinamične sisteme pogosto opišemo z mehkim modelom tipa Takagi-Sugeno [133], ki se poleg mehkega modela tipa Mamdani [95] največkrat uporablja na področju vodenja nelinearnih dinamičnih sistemov.

Prednosti mehkega modela tipa Takagi-Sugeno so [97]:

- metode, ki uporabljajo mehki model tipa Takagi-Sugeno, so računsko učinkovite;
- dobro deluje v povezavi z linearnimi metodami (npr. vodenje z regulatorjem PID);
- dobro deluje v povezavi z optimizacijskimi in adaptivnimi algoritmi;
- zagotavlja zveznost izhodne karakteristike;
- primeren je za matematično analizo.

Zgradba modela tipa Takagi-Sugeno

Premisa (stavek *če*) opisuje razdelitev prostora stanj modela na mehke podprostore, ki jih določajo ustrezne pripadnostne funkcije. V posledičnem delu (stavek *potem*) je določen lokalni izhod modela za vsak podprostor posebej. Glede na vrednosti pripadnostnih funkcij lahko izračunamo dejanski izhod sistema oz. napovedani izhod sistema v naslednjem časovnem koraku [130].

Zapis modela tipa Takagi-Sugeno je podan v enačbi (5.1).

$$\begin{aligned}
 & \mathbf{R}^j : \\
 & \text{če } x_1 \text{ je } A_1^j \text{ in } \dots \text{ in } x_n \text{ je } A_n^j, \\
 & \text{potem } y = f_j(x_1, \dots, x_n) \\
 & \text{za } j = 1, \dots, K
 \end{aligned}
 \tag{5.1}$$

Pripadnostna funkcija je funkcija, ki v vsaki točki vhodnega prostora definira pripadnost $\mu_{A_i^j}$ v obliki realnega števila med 0 in 1, kot kaže enačba (5.2). Vrednost pripadnostne funkcije $\mu_{A_i^j}$ je mišljena v povezavi s pripadajočim pravilom \mathbf{R}^j v mehkem modelu oz. s pripadajočimi mehкими množicami $A_1^j \dots A_n^j$.

$$\begin{aligned} \mu_{A_i^j} &\in [0, 1] \\ &\text{za } j = 1, \dots, K \text{ in } i = 1, \dots, n \end{aligned} \quad (5.2)$$

V mehkem modelu tipa Takagi-Sugeno je za vsako pravilo \mathbf{R}^j ; $j = 1, \dots, K$ definirana funkcija f_j . Ta je v splošnem lahko nelinearna, brez posebej predpisanih omejitev. Kljub temu se v praksi velikokrat izkaže, da je smiselno predpostaviti, da je funkcija f_j afina ali linearna, kot kaže enačba (5.3).

$$\begin{aligned} y_j &= f_j(x_1, \dots, x_n) = a_{1j} x_1 + \dots + a_{nj} x_n + r_j \\ a_{1j}, \dots, a_{nj}, r_j &\in \mathbb{R} \end{aligned} \quad (5.3)$$

V enačbi (5.9) predstavljajo a_{1j}, \dots, a_{nj} in r_j posledične parametre modela, ki pripadajo posameznemu pravilu \mathbf{R}^j , za vsak indeks oz. pravilo $j = 1, \dots, K$ posebej.

Izhod modela tipa Takagi-Sugeno

Izhod modela tipa Takagi-Sugeno določimo z ostrenjem, in sicer glede na pravila \mathbf{R}^j in ustrezne vrednosti pripadnostnih funkcij $\mu_{A_i^j}$. Ostrenje lahko izvedemo z uporabo *T-norme* [130], kot podaja izraz (5.4).

$$y = \sum_{j=1}^K \frac{\prod_{i=1}^n \mu_{A_i^j}}{\sum_{k=1}^K \prod_{i=1}^n \mu_{A_i^k}} \cdot y_j \quad (5.4)$$

5.2 Modeliranje hibridnega mehkega modela

Modeli dinamičnih sistemov so navadno zgrajeni tako, da računajo nova stanja iz zakasnenih vhodnih in izhodnih signalov. Pri časovno diskretnih nelinearnih modelih je pogosta nelinearna samoregresijska struktura z zunanjimi vhodi (angl. *NARX – Nonlinear AutoRegressive with eXogenous inputs*) [125]. Takšna struktura podaja povezavo med vhodno-izhodnimi podatki do časovnega koraka k in izhodom modela \hat{y}_p v naslednjem časovnem koraku $k + 1$.

$$\begin{aligned} \hat{y}_p(k + 1) = F(y(k), \dots, y(k - n + 1), \\ u(k), \dots, u(k - m + 1)) \end{aligned} \quad (5.5)$$

V enačbi (5.5) so $y(k), \dots, y(k - n + 1)$ in $u(k), \dots, u(k - m + 1)$ zakasneni izhodni oz. vhodni signali. Model sistema torej predstavlja nelinearna funkcija F .

5.2.1 Hierarhija hibridnega mehkega modela

Posebna skupina hibridnih sistemov so *preklopni* sistemi, pri katerih ne pride do skokov zveznih stanj pri preklopu diskretnih stanj. V tem okviru se ukvarjamo s hibridnimi sistemi, ki so predstavljeni s hierarhičnim modelom, sestavljenim iz diskretnega in zveznega podmodela, pri čemer je diskretni del na vrhu hierarhije.

Zapis modela v diskretnem času je predstavljen v enačbah (5.6) in (5.7).

$$\mathbf{x}(k + 1) = \mathbf{f}_q(\mathbf{x}(k), \mathbf{u}(k)) \quad (5.6)$$

$$q(k) = g(\mathbf{x}(k), q(k - 1), \mathbf{u}(k)) \quad (5.7)$$

$\mathbf{x} \in \mathbb{R}^n$ je vektor zveznih stanj (število zveznih stanj je n), ki vsebuje tudi vse relevantne izhode sistema y , tj. merljiva zvezna stanja (zakasnjena in nezakasnjena), ki vplivajo na vektor stanj v naslednjem časovnem koraku – glej enačbo

(5.5). $\mathbf{u} \in \mathbb{R}^{m_z} \times \mathbb{Z}^{m_d}$ je vhodni vektor, ki je sestavljen iz m_z zveznih in m_d diskretnih vhodov. $q \in \mathbb{Q}$ (kjer je $\mathbb{Q} = \{1, \dots, s\}$) je diskretno stanje, ki določa podprostor v preklopnem prostoru modela. Diskretno stanje v bistvu označuje način delovanja modela. Model ima s načinov delovanja. Stanja modela so torej v vsakem časovnem koraku k podana z urejenim parom $(\mathbf{x}(k), q(k))$ v definicijskem območju $\mathbb{R}^n \times \mathbb{Q}$.

Lokalno obnašanje modela v enačbi (5.6) je odvisno od diskretnega stanja $q(k)$, ki določa trenutno funkcijo \mathbf{f}_q .

Enačba (5.7) uvaja modifikacijo Witsenhausenovega zapisa hibridnega modela [142] (glej tudi razdelek 2.3.1) v diskretnem časovnem prostoru, tako da upošteva – poleg vektorja zveznih stanj $\mathbf{x}(k)$ in diskretnega stanja v prejšnjem časovnem koraku $q(k-1)$ – tudi vpliv vhodnega vektorja $\mathbf{u}(k)$ na diskretno stanje v trenutnem časovnem koraku $q(k)$.

Zvezni del sistema, ki ga želimo formulirati v obliki hibridnega mehkega modela, je v splošnem nelinearen. Ta del lahko zapišemo z uporabo formulacije Takagi-Sugeno za mehke modele, kot je opisano v razdelku 5.2.2.

5.2.2 Posplošitev modela tipa Takagi-Sugeno na nelinearne hibridne sisteme

Model tipa Takagi-Sugeno [133] smo posplošili na nelinearne hibridne sisteme, tako da smo vključili diskretni del dinamike iz enačbe (5.7) v pravila mehkega modela tipa Takagi-Sugeno. Tako lahko zapišemo pravila hibridnega mehkega modela, kot kaže enačba (5.8).

$$\begin{aligned}
& \mathbf{R}^{jd} : \\
& \text{če } q(k) \text{ je } Q_d \text{ in } y(k) \text{ je } A_1^j \text{ in ...} \\
& \quad \dots \text{ in } y(k - n + 1) \text{ je } A_n^j, \\
& \text{potem } \hat{y}_p(k + 1) = f_{jd}(y(k), \dots, y(k - n + 1), \\
& \quad u(k), \dots, u(k - m + 1)) \\
& \text{za } j = 1, \dots, K \text{ in } d = 1, \dots, s
\end{aligned} \tag{5.8}$$

Premisa opisuje hibridno mehko razdelitev prostora vhodnih spremenljivk hibridnega mehkega modela. $q(k) \in \{1, \dots, s\}$ opisuje diskretno stanje nelinearnega hibridnega sistema, tj. način delovanja modela. Množice Q_d in A_i^j označujejo ostre oz. mehke podprostore, ki jih določajo ustrezne pripadnostne funkcije v prostoru stanj $\mathbb{R}^n \times \mathbb{Q}$.

Posledični del vsebuje funkcije vhodov hibridnega mehkega modela. Tu je $\hat{y}_p(k + 1)$ izhodna spremenljivka, ki predstavlja napovedani izhod sistema v naslednjem časovnem koraku – glej enačbo (5.5).¹

V hibridnem mehkem modelu je za vsako pravilo \mathbf{R}^{jd} , kjer je $j = 1, \dots, K$ in $d = 1, \dots, s$, definirana funkcija f_{jd} . Ta je v splošnem lahko nelinearna in brez posebej predpisanih omejitev. Kljub temu se v praksi velikokrat izkaže, da je smiselno predpostaviti, da je funkcija f_{jd} afina, kot kaže enačba (5.9).

$$\begin{aligned}
& f_{jd}(y(k), \dots, y(k - n + 1), u(k), \dots, u(k - m + 1)) = \\
& = a_{1jd} y(k) + \dots + a_{njd} y(k - n + 1) + \\
& \quad + b_{1jd} u(k) + \dots + b_{mj d} u(k - m + 1) + r_{jd} \\
& a_{1jd}, \dots, a_{njd}, b_{1jd}, \dots, b_{mj d}, r_{jd} \in \mathbb{R}, \text{ za vsak } j = 1, \dots, K \text{ in } d = 1, \dots, s
\end{aligned} \tag{5.9}$$

¹Ko v algoritmu za prediktivno vodenje uporabljamo model Takagi-Sugeno, lahko izhod mehkega modela $\hat{y}_p(k + 1)$ obravnavamo tudi kot napovedano stanje sistema $\hat{x}_p(k + 1)$ – glej enačbo (5.6).

V enačbi (5.9) predstavljajo $a_{1jd}, \dots, a_{njd}, b_{1jd}, \dots, b_{mjd}$ in r_{jd} posledične parametre modela, ki pripadajo posameznemu pravilu \mathbf{R}^{jd} , za vsak indeks $j = 1, \dots, K$ in $d = 1, \dots, s$ posebej.

Število pravil v hibridnem mehkem modelu

Število vseh pravil v hibridnem mehkem modelu znaša največ $K \cdot s$. V splošnem je K odvisen od števila mehkih pripadnostnih funkcij za vsako vhodno spremenljivko v premisi $y(k), \dots, y(k-n+1), u(k), \dots, u(k-m+1)$. Pripadnostne funkcije morajo prekri celotno območje delovanja sistema, pri čemer morajo pravila vsebovati oz. pokrivati (največ) vse mogoče kombinacije pripadnostnih funkcij v prostoru spremenljivk v premisi.

Število pripadnostnih funkcij v hibridnem mehkem modelu

K je torej manjši ali enak produktu števil pripadnostnih funkcij vsake spremenljivke v premisi $y(k), \dots, y(k-n+1), u(k), \dots, u(k-m+1)$. Poudariti je treba, da je odvisen samo od števila mehkih množic A_i^j , saj pripadnostne funkcije niso odvisne od diskretnega stanja hibridnega mehkega modela d . To pomeni da je mehka razdelitev prostora stanj (glede na pripadnostne funkcije) neodvisna od trenutnega diskretnega stanja hibridnega mehkega modela. Zaradi opisane nepovezanosti mehkih pravil in diskretnih stanj v hibridnem mehkem modelu lahko ugotovimo, da se normirane stopnje pripadnosti za posamezno pravilo izračunavajo samo na podlagi zveznih stanj $\mathbf{x}(k)$, trenutno diskretno stanje hibridnega mehkega modela $d(k)$ pa na izračun nima vpliva.

Število načinov delovanja v hibridnem mehkem modelu

Parameter s označuje število načinov delovanja hibridnega mehkega modela, tj. število pripadnostnih funkcij, ki določajo ostre množice Q_d . Število načinov delovanja je odvisno od razdelitve prostora stanj in števila diskretnih vhodov hibridnega mehkega modela.

Če imamo na primer sistem, ki ima dve diskretni vhodni spremenljivki, od katerih lahko prva zavzame 5, druga pa 4 diskretna stanja, je največje skupno število načinov delovanja (ki izhaja samo iz diskretnih vhodov) $5 \cdot 4 = 20$. Če pa upoštevamo še morebitne nedosegljive, nedovoljene, neželene ali nepotrebne kombinacije diskretnih vhodnih spremenljivk, potem se največje skupno število načinov delovanja (ki izhaja samo iz diskretnih vhodov) ustrezno zmanjša. Če na primer v omenjenem sistemu prepovemo 12 kombinacij vrednosti diskretnih vhodnih spremenljivk, se največje skupno število načinov delovanja (ki izhaja samo iz diskretnih vhodov) zmanjša na $5 \cdot 4 - 12 = 8$. Na ta način lahko v praksi marsikdaj precej zmanjšamo kompleksnost hibridnega mehkega modela.

5.2.3 Zapis hibridnega mehkega modela

Izhod hibridnega mehkega modela v kompaktni obliki

Izhod hibridnega mehkega modela v lahko zapišemo v kompaktni obliki. Zapis podaja naslednja enačba.

$$\hat{y}_p(k+1) = \boldsymbol{\beta}(k) \boldsymbol{\Theta}^T(q) \boldsymbol{\psi}(k) \quad (5.10)$$

Vektor normiranih stopenj pripadnosti

V enačbi (5.10) predstavlja $\boldsymbol{\beta}(k) \in \mathbb{R}^K$ vektor normiranih stopenj pripadnosti za vsa pravila ($j = 1, \dots, K$) v trenutnem časovnem koraku k . Elementi vektorja $\boldsymbol{\beta}(k) = [\beta_1(k) \ \beta_2(k) \ \dots \ \beta_K(k)]$ se nanašajo na posamezne pripadnostne funkcije, ki so povezane s pravili hibridnega mehkega sistema.

Predpostavljamo, da vektor normiranih stopenj pripadnosti $\boldsymbol{\beta}(k)$, ki je v splošnem časovno spremenljiv, ustreza enačbi (5.11) v vsakem časovnem koraku k .

$$\boldsymbol{\beta}^T(k) \cdot \mathbf{I} = \sum_{j=1}^K \beta_j(k) = 1 \quad (5.11)$$

za vsak $k = 0, 1, 2, \dots$

V enačbi (5.11) je \mathbf{I} vektor z dolžino K , ki ima vse elemente enake 1.

Vrednost elementa vektorja normiranih stopenj pripadnosti $\beta_j(k)$, ki pripada množici pravil \mathbf{R}^{jd} za vsak $d = 1, \dots, s$, izračunamo z uporabo T -norme [130]. V našem primeru to pomeni preprost normiran algebrski produkt, kot ga podaja enačba (5.12).

$$\beta_j(k) = \frac{\mu_{A_1^j}(y(k)) \cdot \dots \cdot \mu_{A_n^j}(y(k-n+1))}{\sum_{i=1}^K \mu_{A_1^i}(y(k)) \cdot \dots \cdot \mu_{A_n^i}(y(k-n+1))} \quad (5.12)$$

V enačbi (5.12) so $\mu_{A_1^j}(y(k)), \dots, \mu_{A_n^j}(y(k-n+1)) \in [0, 1]$ vrednosti pripadnostnih funkcij, ki so povezane z mehкими množicami A_1^j, \dots, A_n^j [10, 12, 130], in sicer za vsako posamezno pravilo hibridnega mehkega modela \mathbf{R}^{jd} .

Matrika posledičnih parametrov hibridnega mehkega modela

V enačbi (5.10) označuje $\boldsymbol{\Theta}(q) \in \mathbb{R}^{K \times (n+m+1)}$ matriko z $n+m+1$ vrsticami in K stolpci, ki vsebuje posledične parametre hibridnega mehkega modela v trenutnem časovnem koraku k . Kot kaže enačba (5.13), je $\boldsymbol{\Theta}(q)$ funkcija diskretnega stanja sistema $q(k)$.

$$\boldsymbol{\Theta}(q) = \boldsymbol{\Theta}(q(k)) = \begin{cases} \boldsymbol{\Theta}_1 & \text{če } q(k) = 1 \\ \vdots & \vdots \\ \boldsymbol{\Theta}_d & \text{če } q(k) = d \\ \vdots & \vdots \\ \boldsymbol{\Theta}_s & \text{če } q(k) = s \end{cases} \quad (5.13)$$

Matrike Θ_d v enačbi (5.13) vsebujejo posledične parametre hibridnega mehkega modela za vsak način delovanja $q = d \in \{1, \dots, s\}$ posebej. Matrike Θ_d so časovno nespremenljive.

Vsaka matrika Θ_d je sestavljena iz K stolpcev $\Theta_{jd} = [a_{1jd} \dots a_{njd} \quad b_{1jd} \dots b_{mjd} \quad r_{jd}]^T$, ki vsebujejo posledične parametre hibridnega mehkega modela za množico pravil $\{\mathbf{R}^{jd}\}$, pri čemer je d fiksni in $j = 1, \dots, K$.

Matrika $\Theta_d \in \mathbb{R}^{K \times (n+m+1)}$ je sestavljena, kot kaže enačba (5.14).

$$\Theta_d^T = \begin{bmatrix} a_{11d} & \cdots & a_{n1d} & b_{11d} & \cdots & b_{m1d} & r_{1d} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ a_{1Kd} & \cdots & a_{nKd} & b_{1Kd} & \cdots & b_{mKd} & r_{Kd} \end{bmatrix} \quad (5.14)$$

Regresor

V enačbi (5.10) je $\psi(k) = [y(k) \cdots y(k-n+1) \quad u(k) \cdots u(k-m+1) \quad 1]^T$ regresor v časovnem koraku k , ki vsebuje vse vhodne spremenljivke hibridnega mehkega modela, ki nastopajo v f_{jd} .

Regresor $\psi(k) \in \mathbb{R}^{n+m+1}$ je sestavljen, kot kaže enačba (5.15).

$$\psi^T(k) = \begin{bmatrix} y(k) & \cdots & y(k-n+1) & u(k) & \cdots & u(k-m+1) & 1 \end{bmatrix} \quad (5.15)$$

5.2.4 Posplošitev hibridnega mehkega modela

Multivariabilni sistemi

Sistemi, ki jih želimo modelirati s hibridnim mehkim modelom, imajo lahko več vhodov in izhodov. Takim sistemom pravimo multivariabilni sistemi.

Ko modeliramo sistem z več izhodi, lahko funkcije f_{jd} obravnavamo kot vektorske funkcije. Kljub temu se lahko pri modeliranju omejimo samo na hibridne

mehke modele z enim izhodom in s tem seveda funkcije f_{jd} obravnavamo kot skalarne funkcije. V primeru sistema z več izhodi lahko uporabimo več vzporednih hibridnih mehkih (pod)modelov z enim izhodom, ne da bi izgubili splošnost pristopa.

V primeru, da modeliramo sistem z več vhodi pa lahko regresor preprosto razširimo, tako da vključimo vse relevantne vhode modela.

Sistemi višjega reda

Podobno lahko ravnamo pri sistemih višjega reda ($n > 1$). Regresijski vektor v tem primeru vsebuje vse vrednosti izhoda sistema v prejšnjih časovnih korakih $y(k-1)$, ..., $y(k-n+1)$, ki so potrebni za določitev napovedanega izhoda sistema v naslednjem časovnem koraku $\hat{y}_p(k+1)$.

Če so ustrezna stanja sistema, ki nadomestijo zakasnjene izhode $y(k-1)$, ..., $y(k-n+1)$, potrebne za izračun $\hat{y}_p(k+1)$, merljiva, je primerneje uporabiti več vzporednih ustreznih, a preprostejših, hibridnih mehkih (pod)modelov prvega reda. Algoritem za prediktivno vodenje te podmodele poganja vzporedno, kar nadomesti uporabo enega samega kompleksnejšega hibridnega mehkega modela n -tega reda.

Če pristop s pod modeli prvega reda ni mogoč, je še vedno primerno uporabiti več podmodelov reda, ki je nižji od n . To pomeni, da je v splošnem v posameznem časovnem koraku smiselno uporabiti vse merljive signale sistema, ki ga modeliramo. Seveda pa zaradi nemerljivosti nekaterih stanj sistema tak pristop ni vedno mogoč.

5.3 Identifikacija hibridnega mehkega modela

5.3.1 Mehko rojenje

Pri identifikaciji sistemov, ki jih želimo zapisati s hibridnim mehkim modelom, večkrat naletimo na težavo, da dinamike sistema ne poznamo dovolj dobro, da bi lahko določili ustrezne mehke množice A_i^j , ki nastopajo v premisi hibridnega mehkega modela. To pomeni, da ne poznamo niti ustreznih pripadnostnih funkcij, kar pa je predpogoj za ocenjevanje parametrov hibridnega mehkega modela. V takih primerih si lahko pomagamo z algoritmi *mehkega rojenja*, npr. z *mehkim rojenjem c-središč* (angl. *fuzzy c-means clustering*).

Z mehkim rojenjem, ki ga izvajamo nad *vhodno-izhodnim* prostorom hibridnega mehkega modela \mathcal{D}_{VI} , želimo podatke za identifikacijo razdeliti na več *mehkih rojev*. Vsak podatek za identifikacijo tako pripada posameznemu mehchemu roju z določeno stopnjo pripadnosti, ki se izračuna glede na razdaljo podatka za identifikacijo do središč posameznih mehkih rojev, ki jih algoritem določa v vsakem koraku sproti.

Metoda *mehkega rojenja c-središč* temelji na minimizaciji kriterijske funkcije, ki jo podaja enačba (5.16).

$$J_{MR} = \sum_{i=1}^N \sum_{j=1}^C \mu_{ij}^m \|x_i - c_j\|^2 \quad (5.16)$$

V enačbi (5.16) je m izbrano realno število, pri čemer mora veljati $1 \leq m < \infty$. N je število podatkov, C je število rojev, μ_{ij} je vrednost pripadnostne funkcije j -tega roja za i -ti podatek x_i . c_j označuje središče roja j , $\|\cdot\|$ pa normo, s katero je določena podobnost med središčem roja c_j in podatkom x_i .

V praksi se najpogosteje uporablja evklidska norma, ki jo podaja enačba (5.17).

$$\|x\| = \sqrt{x^T \cdot x} \quad (5.17)$$

Mehko rojenje se izvaja iterativno, pri čemer se v vsakem koraku sproti izračunavajo vrednosti pripadnostnih funkcij μ_{ij} , kot kaže enačba (5.18), in središča rojev c_j , kot kaže enačba (5.19).

$$\mu_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}, \quad (5.18)$$

pri čemer za vsak $i \in \{1, \dots, N\}$ velja $\sum_{j=1}^C \mu_{ij} = 1$

$$c_j = \frac{\sum_{i=1}^N \mu_{ij}^m \cdot x_i}{\sum_{i=1}^N \mu_{ij}^m} \quad (5.19)$$

V enačbah (5.18) in (5.19) predstavlja i indeks posameznega podatka za identifikacijo, j in k pa označujeta indeks posameznega mehkega roja oz. njegovega središča. Vrednost parametra m določa *mehkost* oz. *ostrost* porazdelitve pripadnostnih funkcij v prostoru.

Za skrajni primer $m = 1$ velja, da so pripadnostne funkcije, ki so definirane nad prostorom, degenerirane v ostre stopnje pripadnosti. Pripadnostne funkcije lahko torej v vsaki točki prostora zavzamejo samo eno od vrednosti $\mu_{ij} \in \{0, 1\}$. Iz enačbe (5.18) vidimo, da je $\mu_{ij} = 1$, če je norma $\|x_i - c_j\|$ za i -ti podatek za identifikacijo in središče j -tega roja najmanjša glede na vsa središča rojev. Za ostale roje pa velja $\mu_{ik} = 0$, kjer je $k \in \{1, 2, \dots, C\} \setminus j$.

Za drugo skrajnost, $m = \infty$, pa velja, da so pripadnostne funkcije degenerirane zmeščane, kar pomeni, da so vrednosti pripadnostnih funkcij po vsem prostoru enake: $\mu_{ij} = \frac{1}{C}$, in sicer za vsak $j \in \{1, 2, \dots, C\}$.

V praksi sta najpogosteje uporabljeni vrednosti za parameter $m = 1, 25$ ali $m = 2$.

Postopek *mehkega rojenja c-središč* je opisan v nadaljevanju.

1. Izberemo število rojev C in parameter m ter določimo začetno matriko pripadnosti $\Upsilon(0) = [\mu_{ij}]$.
2. V k -ti iteraciji določimo središča rojev c_j za $j = 1, \dots, C$ glede na $\Upsilon(k)$.
3. Določimo novo matriko pripadnosti $\Upsilon(k+1)$.
4. Če je $\|\Upsilon(k+1) - \Upsilon(k)\| < \varepsilon$, končamo postopek, sicer se vrnemo na 2. korak.

5.3.2 Projekcije mehkih rojev v vhodni prostor hibridnega mehkega modela

Središča rojev (in ustrezne pripadnostne funkcije), ki so rezultat algoritma za mehko rojenje, so določena v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} . Tako določene pripadnostne funkcije lahko neposredno uporabimo pri identifikaciji hibridnega mehkega modela, problem pa se pojavi pri uporabi hibridnega mehkega modela za predikcijo obnašanja sistema, in sicer pri določanju normiranih stopenj pripadnosti za posamezen roj $\beta_j(k)$, kjer je $j \in \{1, 2, \dots, C\}$ – glej enačbo (5.12). Vrednosti pripadnostnih funkcij, kot jih določajo rezultati mehkega rojenja, so namreč odvisne od razdalj do središč posameznih rojev v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} – glej enačbo (5.18). Teh razdalj pa pri uporabi hibridnega mehkega modela za predikcijo obnašanja sistema ne poznamo, saj imamo opravka samo z vektorjem v *vhodnem* prostoru hibridnega mehkega modela \mathcal{D}_V .

Ker so pripadnostne funkcije določene z mehkim rojenjem v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} , moramo najti način, kako ustrezno preslikati informacijo o vrednosti pripadnostnih funkcij iz *vhodno-izhodnega* \mathcal{D}_{VI} v *vhodni* prostor hibridnega mehkega modela \mathcal{D}_V , da bi ga lahko uporabljali za predikcijo obnašanja sistema.

Kot kaže enačba (5.18), so vrednosti pripadnostnih funkcij μ_{ij} odvisne od raz-

dalj točke do središč posameznih rojev v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} . Če torej poznamo te razdalje, lahko določimo tudi vrednosti pripadnostnih funkcij μ_{ij} . Problem se torej poenostavi na projekcijo informacije o razdaljah do središč posameznih rojev iz *vhodno-izhodnega* \mathcal{D}_{VI} v *vhodni* prostor hibridnega mehkega modela \mathcal{D}_V .

Množice točk v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} , ki so enako oddaljene od izbrane točke – v našem primeru središča posameznega roja c_j – lahko, ob predpostavki, da velja enačba (5.17), geometrijsko predstavimo z obodom hiperkrogle s središčem v izbrani točki v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} , pri čemer radij krogle določa razdaljo do izbrane točke. Dimenzija hiperkrogle je določena z dimenzijo *vhodno-izhodnega* prostora \mathcal{D}_{VI} , tj. s številom vhodnih in izhodnih spremenljivk hibridnega mehkega modela.

Enačbe hiperkrogel v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} podaja izraz (5.20).

$$[x - c_j]^T \cdot [x - c_j] = r_j^2 = \|x - c_j\|^2, \quad (5.20)$$

kjer je $j \in \{1, 2, \dots, C\}$.

V enačbi (5.20) predstavlja vektor x točko v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} , ki pripada obodu hiperkrogle s središčem v točki, ki jo predstavlja vektor c_j . Radij hiperkrogle je enak evklidski normi $r_j = \|x - c_j\|$. Dimenzije vektorjev x oz. c_j so določene z dimenzijo *vhodno-izhodnega* prostora hibridnega mehkega modela \mathcal{D}_{VI} . Indeks j , kjer je $j \in \{1, 2, \dots, C\}$, določa indeks mehkega roja, ki ga trenutno obravnavamo.

Kot je opisano v razdelku 5.2.2, je v hibridnem mehkem modelu za vsako pravilo \mathbf{R}^{jd} definirana funkcija f_{jd} . Če predpostavimo, da je funkcija f_{jd} afina, kot kaže enačba (5.9), lahko rečemo, da nam predstavlja afini podmodel za primer, ko je $\beta_j = 1$ in $\beta_k = 0$, kjer velja $k \neq j$, ter $q = d$.

Množice točk v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} ,

ki jih določa afina funkcija f_{jd} , lahko geometrijsko predstavimo s hiperravnino v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} .

Enačbe hiperravnin v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} podaja izraz (5.21).

$$[x - s_j]^T \cdot n_j = 0, \quad (5.21)$$

kjer je $j \in \{1, 2, \dots, C\}$.

V enačbi (5.21) predstavlja vektor x točko v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} , ki pripada hiperravnini, ki jo določata poljubna točka s_j , ki leži na hiperravnini, in vektor normale n_j . Zaradi jasnosti lahko, ne da bi izgubili splošnost, predpostavimo, da je točka s_j tista točka na hiperravnini (5.21), ki je najbližja središču roja c_j . V primeru, da središče roja c_j leži na hiperravnini (5.21), točki s_j in c_j sovpadata.

Tudi za izraz (5.21) velja, da so dimenzije vektorjev x , s_j in n_j določene z dimenzijo *vhodno-izhodnega* prostora hibridnega mehkega modela \mathcal{D}_{VI} .

Vektor x , ki je določen v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} , lahko razdelimo na komponente x_V , ki so določene v *vhodnem* prostoru \mathcal{D}_V , in komponento x_I , ki je določena v *izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_I , kot kaže enačba (5.22).

$$x = \begin{bmatrix} x_V \\ x_I \end{bmatrix} \quad (5.22)$$

Pri uporabi hibridnega mehkega modela za predikcijo obnašanja sistema poznamo samo x_V , ne pa tudi x_I , kar pa je, kot bomo pokazali v nadaljevanju, rešljiva težava.

Če predpostavimo, da je x_I parameter, potem parametrizirani enačbi (5.20) in (5.21) določata v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI}

konturo, ki predstavlja presečišče hiperkrogle (5.20) in hiperravnine (5.21). Ko obravnavamo obstoj konture, imamo tri možnosti.

- Če je $0 < r_j < r_{j,min}$, potem ta kontura ne obstaja.
- Če je $r_j = r_{j,min}$, potem je ta kontura degenerirana v točko.
- Če je $r_{j,min} < r_j < \infty$, potem je ta kontura hiperkrožnica.

V zgornjih pogojih označuje $r_{j,min}$ minimalno vrednost parametra r_j , ki zagotavlja obstoj presečiščne hiperkrožnice.

Za nas je zanimiv tretji primer, ko je ta kontura hiperkrožnica. Tako določena hiperkrožnica predstavlja točke v *vhodno-izhodnem* prostoru hibridnega mehkega modela \mathcal{D}_{VI} , pri katerih je normirana razlika vektorjev enaka radiju hiperkrogle (5.20) $\|x - c_j\| = r_j$, in ki hkrati pripadajo hiperravnini (5.21). Veljati mora torej izraz (5.23).

$$0 \leq r_{j,min} \leq r_j < \infty \quad (5.23)$$

Če središče hiperkrogle (5.20) c_j leži na hiperravnini (5.21), potem velja $r_{j,min} = 0$.

Tako določeno hiperkrožnico lahko projiciramo v *vhodni* prostor hibridnega mehkega modela \mathcal{D}_V in na ta način izpeljemo hiperelipso, ki določa točke v *vhodnem* prostoru hibridnega mehkega modela \mathcal{D}_V , ki jim lahko pripišemo vrednost r_j .

Enačbe hiperelips v *vhodnem* prostoru hibridnega mehkega modela \mathcal{D}_V podaja izraz (5.24).

$$[x_V - s_{j,V}]^T A_{r_j} [x_V - s_{j,V}] = r_j^2, \quad (5.24)$$

kjer je $j \in \{1, 2, \dots, C\}$ in $0 \leq r_{j,min} \leq r_j < \infty$.

V izrazu (5.24) označuje x_V točko v *vhodnem* prostoru hibridnega mehkega modela \mathcal{D}_V , $s_{j,V}$ označuje središče hiperelipse v *vhodnem* prostoru hibridnega mehkega modela \mathcal{D}_V , ki se ga določi s projekcijo točke s_j iz *vhodno-izhodnega* prostora hibridnega mehkega modela \mathcal{D}_{VI} v *vhodni* prostor hibridnega mehkega modela \mathcal{D}_V , kvadratna matrika A_{r_j} pa v geometrijskem smislu določa orientacijo in dolžine polosi hiperelipse.

Različne vrednosti parametra r_j določajo različne hiperelipse v *vhodnem* prostoru hibridnega mehkega modela \mathcal{D}_V , pri čemer mora seveda veljati izraz (5.23).

Na opisani način lahko torej izpeljemo C funkcij, ki so definirane nad *vhodnim* prostorom hibridnega mehkega modela \mathcal{D}_V in ki vsaki točki v prostoru x_V priredijo vrednost $r_j(x_V)$, in sicer za $j \in \{1, 2, \dots, C\}$, kot podaja izraz (5.25).

$$\begin{aligned} r_j &: \mathcal{D}_V \rightarrow [r_{j,min}, \infty), \\ r_j &: x_V \mapsto r_j(x_V), \\ &\text{kjer je } j \in \{1, 2, \dots, C\}. \end{aligned} \tag{5.25}$$

Če poznamo funkcije $r_j(x_V)$ za vsak $j \in \{1, 2, \dots, C\}$, lahko izpeljemo pripadnostne funkcije, ki ustrezajo rezultatom mehkega rojenja in so definirane nad *vhodnim* prostorom hibridnega mehkega modela \mathcal{D}_V in ki vsaki točki v prostoru x_V priredijo vrednost $\mu_j(x_V)$, in sicer za $j \in \{1, 2, \dots, C\}$, kot podaja izraz (5.26).

$$\begin{aligned} \mu_j &= \frac{1}{\sum_{k=1}^C \left(\frac{r_j}{r_k}\right)^{\frac{2}{m-1}}}, \\ &\text{kjer je } j \in \{1, 2, \dots, C\}. \end{aligned} \tag{5.26}$$

Pripadnostne funkcije, ki so zapisane v izrazu (5.26), lahko neposredno implementiramo pri uporabi hibridnega mehkega modela za predikcijo obnašanja sistema – glej enačbo (5.10) – tj. pri določanju normiranih stopenj pripadnosti za posamezen roj $\beta_j(k)$, kjer je $j \in \{1, 2, \dots, C\}$ – glej enačbo (5.12).

5.3.3 Globalni linearni model

Hibridni mehki model s poenoteno strukturo posledičnega dela, ki je opisan v razdelku 5.2.3, lahko izrazimo kot globalni linearni model. Parametre, ki so odvisni od vhodnih signalov modela, lahko izpeljemo iz enačbe (5.14), kot kaže enačba (5.27).

$$\tilde{\Theta}(k) = \Theta(k) \beta(k)^T \quad (5.27)$$

Ob tej predpostavki lahko zapišemo izhod hibridnega mehkega modela iz enačbe (5.10), kot ga podaja enačba (5.28).

$$\hat{y}_p(k+1) = \tilde{\Theta}(k)^T \psi(k) \quad (5.28)$$

5.3.4 Priprava podatkov za ocenjevanje parametrov hibridnega mehkega modela

Identifikacija hibridnega mehkega modela pomeni določitev njegovih parametrov $a_{1jd}, \dots, a_{njd}, b_{1jd}, \dots, b_{mjd}$ in r_{jd} , in sicer za vsako pravilo \mathbf{R}^{jd} ; $j = 1, \dots, K$ in $d = 1, \dots, s$. To pomeni, da moramo določiti elemente vseh matrik Θ_d – glej enačbo (5.14).

Regresijsko matriko Ψ_{jd} v enačbi (5.29) za pravilo \mathbf{R}^{jd} določimo iz vhodnih podatkov sistema. Indeks k teče od k_1 do $k_{P_{jd}}$, pri čemer P_{jd} označuje število vhodno-izhodnih parov, ki ustrezajo pravilu \mathbf{R}^{jd} .

V regresijski matriki so uporabljeni samo podatki, ki ustrezajo pogoju (5.30) in (5.31). δ označuje majhno pozitivno število. Parametre modela izračunamo z inverzijo matrik, zato lahko pride do težav, če so matrike slabo pogojene. Pogoj v enačbi (5.31) zagotavlja dobro pogojenost matrik, kar pomeni, da se podatek upošteva samo v primeru, da je normirana stopnja pripadnosti $\beta_j(k)$ ustrezno velika.

$$\mathbf{\Psi}_{jd} = \begin{bmatrix} \beta_j(k_1) \psi^T(k_1) \\ \vdots \\ \beta_j(k_{P_{jd}}) \psi^T(k_{P_{jd}}) \end{bmatrix} \quad (5.29)$$

$$q(k) = d \quad (5.30)$$

$$\beta_j(k) \geq \delta \quad (5.31)$$

Izhodna spremenljivka sistema je vključena v vektor izhodnih podatkov \mathbf{Y}_{jd} , ki ustreza pravilu \mathbf{R}^{jd} , kot kaže enačba (5.32). Tudi v tem primeru velja, da so pri sestavljanju izhodnega vektorja podatkov \mathbf{Y}_{jd} uporabljeni samo podatki iz časovnih korakov $(k + 1)$, ki ustrezajo pogojem v enačbah (5.30) in (5.31).

$$\mathbf{Y}_{jd} = \begin{bmatrix} \beta_j(k_1) y(k_1 + 1) \\ \vdots \\ \beta_j(k_1) y(k_{P_{jd}} + 1) \end{bmatrix} \quad (5.32)$$

Del prispevka k izhodu hibridnega mehkega modela $\hat{y}_p^{jd}(k + 1)$, ki se nanaša na pravilo \mathbf{R}^{jd} , podaja enačba (5.33).

$$\beta^j(k_1) \hat{y}_p^{jd}(k + 1) = \mathbf{\Theta}_{jd}^T (\beta^j(k_1) \psi(k)) \quad (5.33)$$

V enačbi (5.33) predstavlja vektor $\mathbf{\Theta}_{jd}$ tisti stolpec v matriki $\mathbf{\Theta}_d$, ki vsebuje parametre hibridnega mehkega modela, ki se nanašajo na pravilo \mathbf{R}^{jd} , kot je zapisano v enačbi (5.34).

$$\mathbf{\Theta}_{jd}^T = [a_{1jd} \dots a_{njd} \quad b_{1jd} \dots b_{mjd} \quad r_{jd}] \quad (5.34)$$

5.3.5 Ocenjevanje parametrov hibridnega mehkega modela z modificirano metodo najmanjših kvadratov

Glede na enačbe (5.29), (5.32) in (5.33) lahko parametre hibridnega mehkega modela, ki pripadajo posameznemu pravilu \mathbf{R}^{jd} , ocenimo z modificirano metodo najmanjših kvadratov, kot opisuje enačba (5.35).

$$\Theta_{jd} = (\Psi_{jd}^T \Psi_{jd})^{-1} \Psi_{jd}^T \mathbf{Y}_{jd} \quad (5.35)$$

Parametre hibridnega mehkega modela določimo z modificirano metodo najmanjših kvadratov za vsako pravilo \mathbf{R}^{jd} ($j = 1, \dots, K$ in $d = 1, \dots, s$) posebej. Ko dobljene rezultate združimo, lahko končno zapišemo celoten hibridni mehki model.

Za identifikacijo parametrov hibridnega mehkega modela uporabimo izmerjene vhodno-izhodne signale in modificirano identifikacijsko metodo najmanjših kvadratov. Pristop temelji na dekompoziciji matrice vhodno-izhodnih podatkov Ψ v $K \cdot s$ podmatrik Ψ_{jd} . Parametri se za vsako pravilo \mathbf{R}^{jd} ($j = 1, \dots, K$ in $d = 1, \dots, s$) izračunajo posebej. Zaradi boljše pogojenosti podmatrik Ψ_{jd} v primerjavi s celotno matriko Ψ dobimo boljšo oceno parametrov hibridnega mehkega modela, tj. manjše variance ocen parametrov v primerjavi s klasičnim postopkom, znanim iz literature [10, 12, 130, 133].

Opisani pristop s trenutno linearizacijo sistema generira parametre globalnega linearnega modela (glej enačbo (5.27)). Parametri so odvisni od premise hibridnega mehkega modela, ki jo določajo spremenljivke $q(k)$, $y(k)$, ..., $y(k-n+1)$ in $u(k)$, ..., $u(k-m+1)$. Parametre globalnega linearnega modela bi lahko v algoritmu za prediktivno vodenje neposredno uporabili za napovedovanje obnašanja sistema, pri čemer bi se moral regulator seveda sproti prilagajati dinamičnim spremembam v procesu.

6. Primer postopka modeliranja in identifikacije odsekoma afinega modela

Predstavili bomo postopek modeliranja in identifikacije odsekoma afinega modela, in sicer na primeru šaržnega reaktorja, ki je opisan v 3. poglavju.

6.1 Podatki za identifikacijo

Za identifikacijo šaržnega reaktorja potrebujemo primerne vhodno-izhodne signale, iz katerih lahko ocenimo dinamične lastnosti sistema. Signale za identifikacijo tvorimo s psevdonaključnim vhodnim signalom, ki ga uporabimo na simulacijskem modelu šaržnega reaktorja,¹ in posnetimi odzivi sistema na ta vhodni signal.

Na vhode šaržnega reaktorja smo v vsakem časovnem koraku pripeljali vrednost, ki smo jo dobili iz psevdonaključnega generatorja, in sicer za mešalni ventil k_M in ločeno za zaporna ventila k_H in k_C . Vhodni signali so imeli obliko stopničaste funkcije. Signal za mešalni ventil k_M smo tvorili z uporabo mehanizma, ki preklaplja med psevdonaključno izbranimi vrednostmi v definicijskem območju $k_M \in \{0; 0,01; 0,02; 0,05; 0,1; 1\}$ in zadržuje isto izbrano vrednost psevdonaključno število časovnih korakov. S podobnim psevdonaključnim mehanizmom smo tvorili tudi signala za zaporna ventila k_H in k_C , kjer velja $(k_H, k_C) \in \{(0, 1); (1, 0)\}$. Čas vzorčenja je bil $T_s = 10$ s.

Vhodni signali morajo upoštevati celotno frekvenčno in amplitudno območje

¹Merjenim izhodom simulacijskega modela je bil dodan šum.

sistema, ki nas zanima. Na ta način zagotovimo, da bomo z uporabo signalov za identifikacijo lahko izpeljali model, ki bo opisoval celotno dinamiko sistema.

V našem primeru se v šaržnem reaktorju meša, greje in hladi granulati, iz katerega želimo dobiti homogeno zmes. V jedru ne potekajo ekso- ali endotermne reakcije. Zato lahko predpostavimo, da je sistem časovno nespremenljiv. Čas trajanja eksperimenta na šaržnem reaktorju, v katerem smo pridobili podatke za identifikacijo, je bil daljši od predvidenega časa za obdelavo ene šarže, vendar to zaradi prej navedenih razlogov ne vpliva na dinamične lastnosti šaržnega reaktorja.

V podatke za identifikacijo smo zajeli odziv šaržnega reaktorja na psevdona-ključni vhodni signal: posneli smo oba merljiva izhoda, tj. temperaturo v jedru T in temperaturo v plašču šaržnega reaktorja T_w .

Slika 6.1 prikazuje izhodna signala za identifikacijo T in T_w , na sliki 6.2 pa je prikazan še približan izsek poteka signalov.

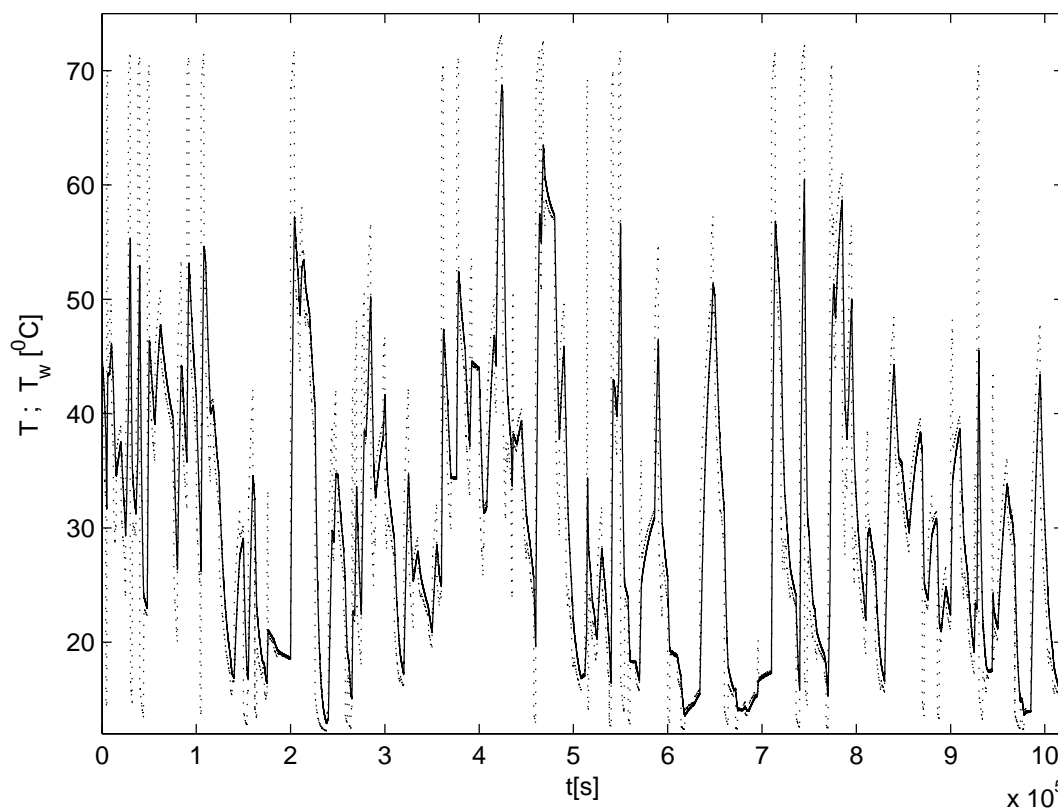
Slika 6.3 prikazuje vhodne signale za identifikacijo k_M , k_H in k_C , na sliki 6.4 pa je prikazan še približan izsek poteka signalov.

6.2 Razdelitev sistema

Model šaržnega reaktorja izpeljemo po korakih.

- Najprej razstavimo multivariabilni sistem na dva preprostejša podsistema z več vhodi in enim izhodom.
- Z upoštevanjem vpliva obeh izhodov na oba podsistema določimo strukturo podmodela za vsak podsystem posebej.

Vsekakor bi bilo model možno konstruirati tudi brez predhodne razdelitve sistema na posamezne podsisteme. Ker pa parametre modela identificiramo z inverzijo matrik, bi lahko pri tem naleteli na numerične težave zaradi slabše



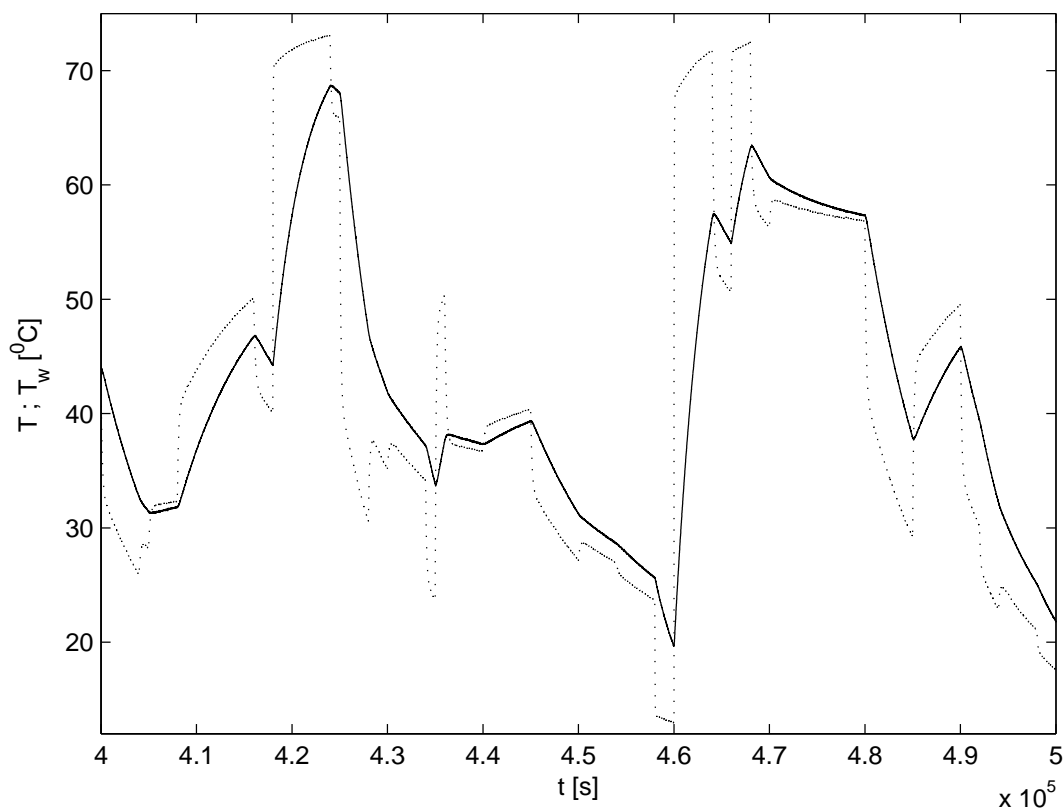
Slika 6.1: Temperatura v jedru T (polna črta) in plašču T_w (prekinjena črta) šaržnega reaktorja.

pogojenosti matrik, ki nastopajo pri identifikaciji. Poleg tega bi morali v tem primeru naenkrat ocenjevati večje število parametrov – tj. matriko parametrov Θ ustrezno večje dimenzije – to pa bi se nujno pokazalo v slabših rezultatih identifikacije.

Zaradi tega je seveda smiselno uporabiti predlagano razdelitev sistema in na ta način obravnavati poenostavljeno strukturo hibridnega mehkega modela. Tako se izognemo morebitnim numeričnim težavam pri ocenjevanju parametrov modela.

Glede na razdelitev toplotnih tokov, ki nastopajo v šaržnem reaktorju (glej razdelek 3.2), lahko sistem razdelimo na dva podsistema:

- temperatura v jedru šaržnega reaktorja;
- temperatura v plašču šaržnega reaktorja.



Slika 6.2: Temperatura v jedru T (polna črta) in plašču T_w (prekinjena črta) šaržnega reaktorja – izsek.

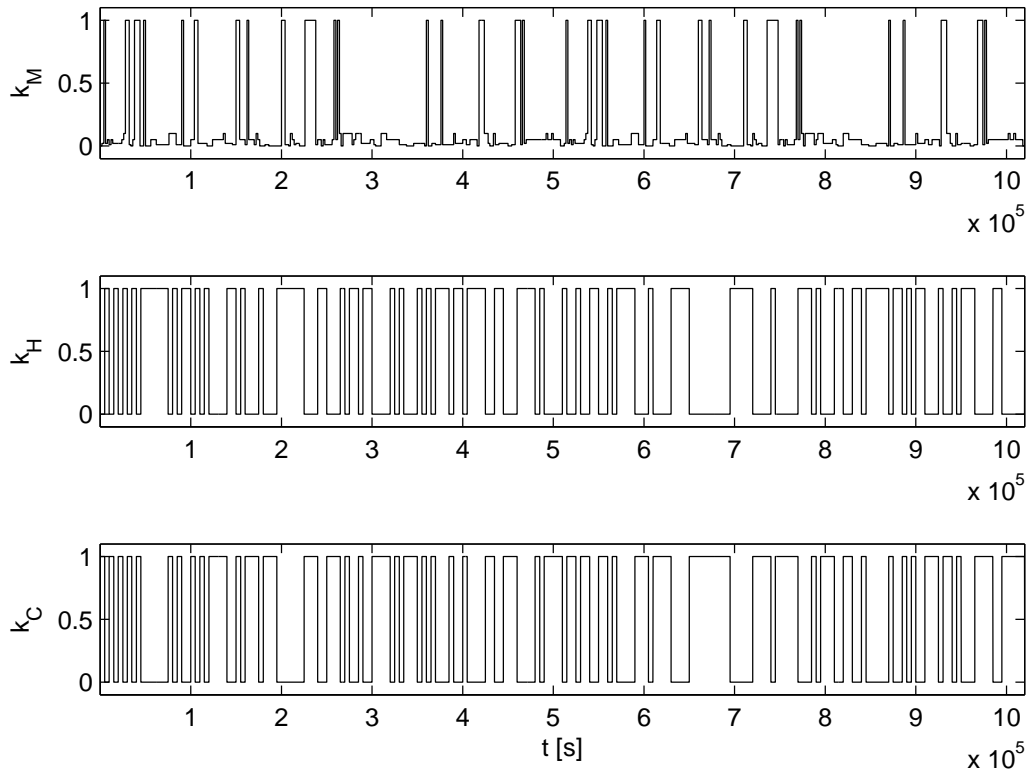
Na ta način izkoristimo predhodno poznavanje strukture sistema: v procesu modeliranja torej identificiramo *sivo škatlo*, kar predstavlja neke vrste srednjo pot med identifikacijo *črne škatle* in popolnoma teoretičnim modeliranjem.

6.3 Temperatura v jedru šaržnega reaktorja

6.3.1 Struktura modela

Temperatura v jedru reaktorja T je odvisna samo od pretoka toplote med jedrom in plaščem reaktorja, ki je odvisen od temperature v plašču T_w in jedru T šaržnega reaktorja.

Predpostavimo lahko model z več vhodi in enim izhodom, kot kaže enačba

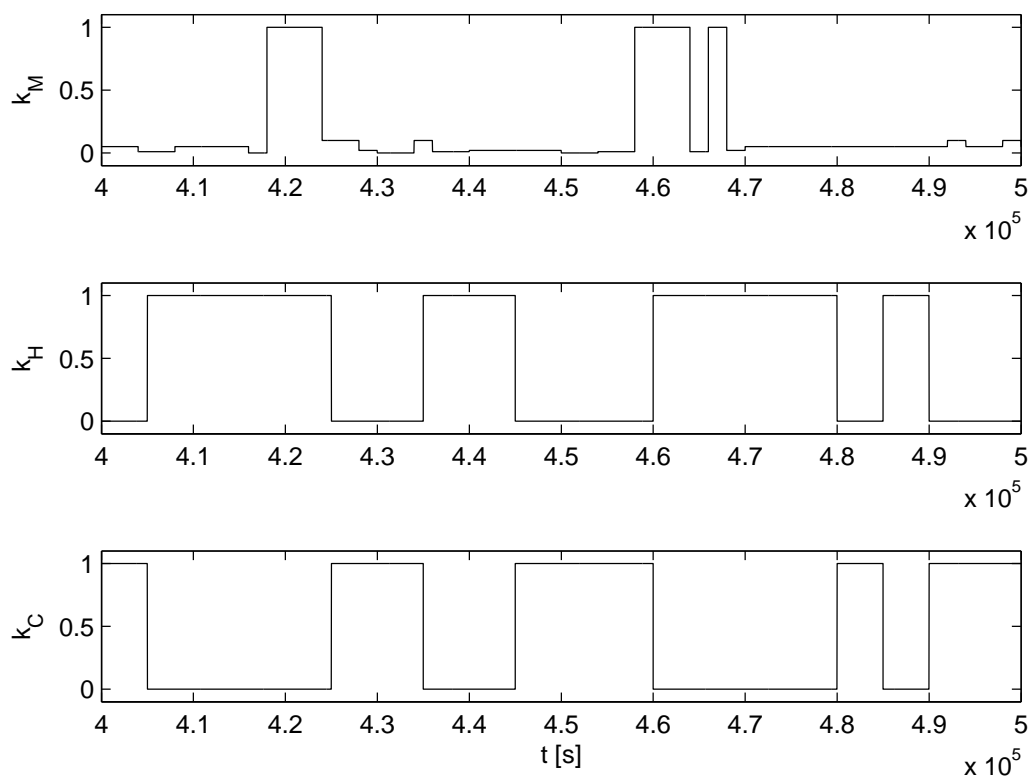
Slika 6.3: Vhodni signali k_M , k_H in k_C .

(6.1). Regresor je sestavljen iz temperature v plašču $T_w(k)$ in jedru $T(k)$ šaržnega reaktorja v trenutnem časovnem koraku k .

$$\hat{T}(k+1) = f(T_w(k), T(k)) \quad (6.1)$$

Predpostavimo lahko, da je toplotni tok proporcionalen temperaturni razliki med plaščem T_w in jedrom T šaržnega reaktorja, zato lahko zapišemo linearni model prvega reda, kot kaže enačba (6.2).

$$\hat{T}(k+1) = \boldsymbol{\theta}^T [T_w(k) \quad T(k)]^T \quad (6.2)$$

Slika 6.4: Vhodni signali k_M , k_H in k_C – izsek.

6.3.2 Parametri modela

Po identifikaciji parametrov z metodo najmanjših kvadratov dobimo rezultat θ – glej enačbo (6.3).

$$\theta = [0.0033 \ 0.9967]^T \quad (6.3)$$

6.4 Temperatura v plašču šaržnega reaktorja

Temperatura v plašču reaktorja T_w je odvisna od pretoka toplote med plaščem reaktorja in jedrom ter med plaščem reaktorja in okolico, upoštevati pa moramo tudi pretok toplote zaradi dotoka sveže vhodne vode in iztoka vode iz plašča.

Predpostavimo lahko model z več vhodi in enim izhodom, kot kaže enačba (6.4). Regresor je sestavljen iz temperature v plašču $T_w(k)$ in jedru $T(k)$ šaržnega reaktorja ter vhodnih signalov, in sicer mešalnega ventila $k_M(k)$ ter zapornih ventilov za vročo vodo $k_H(k)$ in za hladno vodo $k_C(k)$, v trenutnem časovnem koraku k .

$$\hat{T}_w(k+1) = F(T_w(k), T(k), k_M(k), k_C(k), k_H(k)) \quad (6.4)$$

Dinamika podsistema temperature v plašču šaržnega reaktorja T_w izkazuje tako hibridne kot nelinearne lastnosti, ki jih bomo predstavili z odsekoma afinim modelom.

Pri modeliranju odsekoma afinega sistema je prvi korak razdelitev vhodnega prostora modela na ustrezne podprostore (konveksne poliedre), v katerih je dinamika sistema aproksimirana z lokalnim afinim modelom – glej razdelek 4.1.1. V okviru posameznega podprostora nato izvedemo identifikacijo parametrov, z dobljenimi rezultati pa lahko sestavimo celoten odsekoma afini model.

6.4.1 Odsekoma afini model – 1. pristop

Razdelitev vhodnega prostora modela

Vhodni prostor modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]}^T$ v sistemu, ki opisuje temperaturo v plašču šaržnega reaktorja T_w , določa definicijsko območje spremenljivk T_w , T , k_M , k_C in k_H in je podan v enačbi (6.5).

$$\mathcal{D}_{[T_w, T, k_M, k_C, k_H]}^T \equiv \mathbb{R}^2 \times [0, 1] \times \{(0, 1), (1, 0)\} \quad (6.5)$$

Prva in očitna delitev vhodnega prostora modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]}^T$ na podprostore je gotovo glede na način delovanja, tj. glede na položaj zapornih ventilov.

- Prvi način delovanja je, ko je sveža vhodna voda vroča, tj. $T_{in} = T_H$ oz. $k_C(k) = 0$ in $k_H(k) = 1$.

- Drugi način delovanja je, ko je sveža vhodna voda hladna, tj. $T_{in} = T_C$ oz. $k_C(k) = 1$ in $k_H(k) = 0$.

V naslednjem koraku moramo vhodni prostor modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]}^T$ dodatno razdeliti tako, da bo končni odsekoma afini model z uporabo več afinih modelov v posameznih podpodročjih zaobjel nelinearno dinamiko v sistemu. Prostor torej dodatno segmentiramo glede na temperaturo v plašču reaktorja $T_w(k)$. Privzamemo, da so središča ustreznih podprostorov pri naslednjih vrednostih temperature v plašču reaktorja $T_w(k)$: $12\text{ }^{\circ}\text{C}$, $20\text{ }^{\circ}\text{C}$, $40\text{ }^{\circ}\text{C}$, $60\text{ }^{\circ}\text{C}$ in $70\text{ }^{\circ}\text{C}$. Na ta način smo zajeli celotno območje delovanja.

Meje med posameznimi podprostori določimo na simetrali ustreznih središč: $16\text{ }^{\circ}\text{C}$, $30\text{ }^{\circ}\text{C}$, $50\text{ }^{\circ}\text{C}$ in $65\text{ }^{\circ}\text{C}$.

Delitev vhodnega prostora modela (6.4) podaja enačba (6.6).

$$[T_w(k), T(k), k_M(k), k_C(k), k_H(k)]^T \in$$

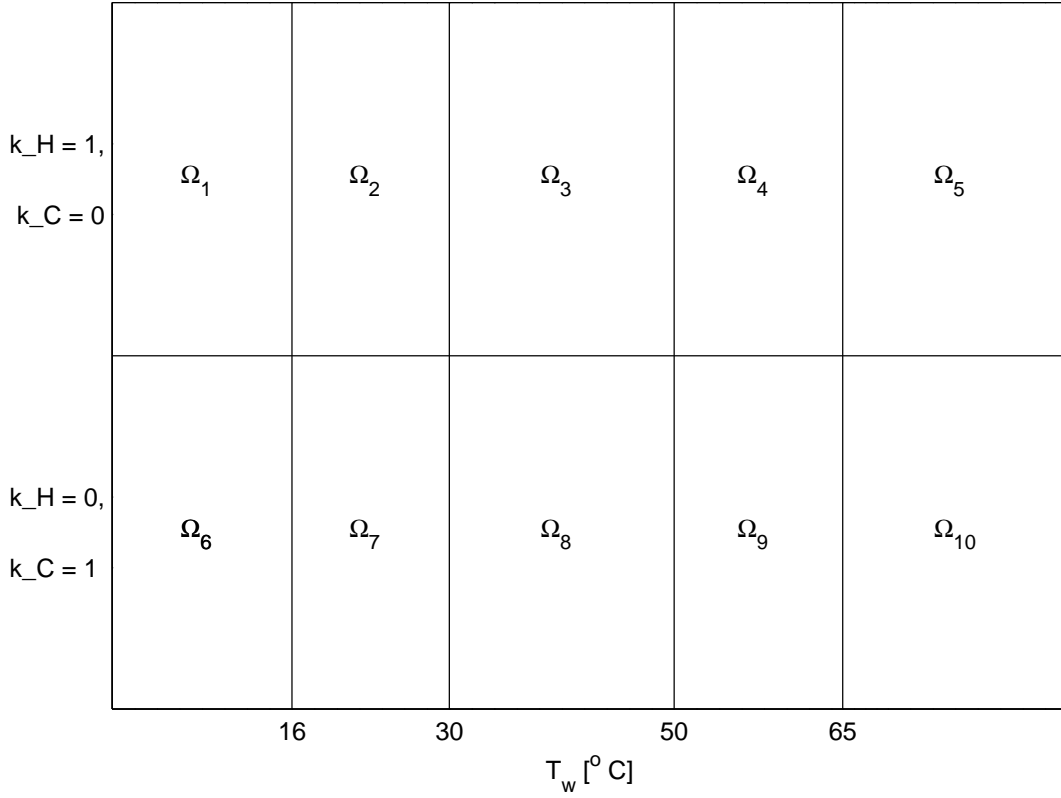
$$\in \left\{ \begin{array}{l} \Omega_1; \quad \text{če velja} \quad T_w(k) \leq 16 \text{ }^\circ\text{C} \wedge k_C(k) = 0 \wedge k_H(k) = 1 \\ \Omega_2; \quad \text{če velja} \quad 16 \text{ }^\circ\text{C} < T_w(k) \leq 30 \text{ }^\circ\text{C} \wedge k_C(k) = 0 \wedge k_H(k) = 1 \\ \Omega_3; \quad \text{če velja} \quad 30 \text{ }^\circ\text{C} < T_w(k) \leq 50 \text{ }^\circ\text{C} \wedge k_C(k) = 0 \wedge k_H(k) = 1 \\ \Omega_4; \quad \text{če velja} \quad 50 \text{ }^\circ\text{C} < T_w(k) \leq 65 \text{ }^\circ\text{C} \wedge k_C(k) = 0 \wedge k_H(k) = 1 \\ \Omega_5; \quad \text{če velja} \quad 65 \text{ }^\circ\text{C} < T_w \text{ }^\circ\text{C} \wedge k_C = 0 \wedge k_H = 1 \\ \Omega_6; \quad \text{če velja} \quad T_w(k) \leq 16 \text{ }^\circ\text{C} \wedge k_C(k) = 1 \wedge k_H(k) = 0 \\ \Omega_7; \quad \text{če velja} \quad 16 \text{ }^\circ\text{C} < T_w(k) \leq 30 \text{ }^\circ\text{C} \wedge k_C(k) = 1 \wedge k_H(k) = 0 \\ \Omega_8; \quad \text{če velja} \quad 30 \text{ }^\circ\text{C} < T_w(k) \leq 50 \text{ }^\circ\text{C} \wedge k_C(k) = 1 \wedge k_H(k) = 0 \\ \Omega_9; \quad \text{če velja} \quad 50 \text{ }^\circ\text{C} < T_w(k) \leq 65 \text{ }^\circ\text{C} \wedge k_C(k) = 1 \wedge k_H(k) = 0 \\ \Omega_{10}; \quad \text{če velja} \quad 65 \text{ }^\circ\text{C} < T_w(k) \text{ }^\circ\text{C} \wedge k_C(k) = 1 \wedge k_H(k) = 0 \end{array} \right.$$

(6.6)

Omenimo, da oblika zapisa v spodnji enačbi ni povsem identična splošnemu zapisu segmentacije prostora na konveksne poliedre, kot ga podaja enačba (4.7), ampak je nekoliko bolj kompaktna. Kljub temu pa je mogoče zapis v enačbi (6.6) brez težav pretvoriti v zapis v enačbi (4.7). Poleg tega lahko opazimo, da so podprostor Ω_1 , Ω_5 , Ω_6 in Ω_{10} v eno smer neomejeni. V praksi to pomeni, da jih omejimo pri ustrezno veliki oz. majhni vrednosti temperature v plašču reaktorja $T_w(k)$.

Slika 6.5 grafično prikazuje razdelitev vhodnega prostora modela (6.4).

Ker zaporna ventila k_C in k_H naenkrat ne moreta zavzeti iste vrednosti, veljata enačbi (6.7) in (6.8). Zato lahko rečemo, da je vhodni prostor modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$ ustrezno razdeljen na konveksne poliedre, ki so osnova za zapis modela v obliki odsekoma afinega sistema.



Slika 6.5: Delitev vhodnega prostora modela (6.4).

$$\Omega_i \cap \Omega_j \equiv \emptyset; \quad (6.7)$$

za vsak $i, j \in \{1, \dots, 10\}$ in $i \neq j$

$$\bigcup_{i=1}^{10} \Omega_i \equiv \mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T} \quad (6.8)$$

Lokalni afini modeli

Za vsak podprostor Ω_i vhodnega prostora modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$, ki smo ga določili v prejšnjem razdelku, lahko zapišemo lokalni afini model, kot ga podaja enačba (6.9).

$$T_w(k+1) = A_i T_w(k) + B_i \begin{bmatrix} T(k) \\ k_M(k) \end{bmatrix} + f_i \quad (6.9)$$

$$\text{za } [T_w, T, k_M, k_C, k_H]^T \in \Omega_i$$

Omenimo nekatere posebnosti oz. razlike glede na enačbi (4.2) in (4.3), ki se pojavijo v enačbi (6.9) in so vezane na konkretni primer.

- V enačbi (6.9) ni izhodnega dela, ki ga formulacija odsekoma afinega modela vsebuje – glej enačbo (4.2). Ker stanje sistema že samo po sebi predstavlja tudi izhod sistema ($y(k) = x(k) = T_w(k)$), zapis izhodnega dela ni smiseln. Kljub temu lahko podamo: $C_i = 1$, $D_i = 0$ in $g_i = 0$ za vsak Ω_i , kjer je $i \in \{1, \dots, 10\}$.
- Vektor vhodov v sistem $[T(k), k_M(k), k_C(k), k_H(k)]^T$ je v enačbi (6.9) skrčen na $[T(k), k_M(k)]^T$, saj položaj zapornih ventilov $k_C(k)$ in $k_H(k)$ vpliva samo na določitev trenutnega podprostora Ω_i , v katerem se sistem nahaja v časovnem koraku k .
- Razdelitev vhodnega prostora modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$ na podprostore Ω_i določa enačba (6.6).

Parametri modela

Podatke za identifikacijo, ki smo jih dobili, kot je opisano v razdelku 6.1, pred ocenjevanjem parametrov razdelimo glede na razdelitev vhodnega prostora modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$ na podprostore Ω_i , ki jo določa enačba (6.6).

Parametre ocenimo s klasično metodo najmanjše vsote kvadratov odstopanj za vsak podprostor Ω_i posebej. Parametri odsekoma afinega modela, ki nastopajo v enačbi (6.9), so podani v enačbi (6.10).

$$\begin{aligned}
\Omega_1 : A_1 &= 0,8525; B_1 = [0,3597 \quad -10,7342]; f_1 = -3,1043 \\
\Omega_2 : A_2 &= 1,0626; B_2 = [-0,0624 \quad -17,2953]; f_2 = 0,0564 \\
\Omega_3 : A_3 &= 1,0168; B_3 = [-0,0159 \quad -3,7872]; f_3 = 0,0283 \\
\Omega_4 : A_4 &= 0,9505; B_4 = [0,0144 \quad 1,5736]; f_4 = 1,8701 \\
\Omega_5 : A_5 &= 0,7438; B_5 = [0,0266 \quad 1,7284]; f_5 = 15,1342 \\
\Omega_6 : A_6 &= 0,9750; B_6 = [0,0025 \quad -0,0666]; f_6 = 0,3388 \\
\Omega_7 : A_7 &= 0,9991; B_7 = [0,0024 \quad 0,0827]; f_7 = 0,0773 \\
\Omega_8 : A_8 &= 0,9879; B_8 = [0,0163 \quad 4,4435]; f_8 = 0,9739 \\
\Omega_9 : A_9 &= 0,7579; B_9 = [0,1571 \quad 9,9894]; f_9 = 1,5204 \\
\Omega_{10} : A_{10} &= -171,3926; B_{10} = [18,6703, \quad -2030,58]; f_{10} = 9805,72
\end{aligned} \tag{6.10}$$

6.4.2 Odsekoma afini model – 2. pristop

Postopek modeliranja

Pri modeliranju odsekoma afinega modela z 2. pristopom postopamo popolnoma enako, kot pri 1. pristopu, ki je opisan v razdelku 6.4.1. Vhodni prostor modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$ je torej razdeljen na podprostore Ω_i , ki jih določa enačba (6.6), odsekoma afini model pa podaja enačba (6.9). Edina razlika med pristopoma je v ocenjevanju parametrov modela.

Parametri modela

Podatke za identifikacijo smo dobili, kot je opisano v razdelku 6.1. Pri prejšnjem pristopu smo podatke pred ocenjevanjem parametrov razdelili glede na razdelitev vhodnega prostora modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$ na podprostore Ω_i , ki jih določa enačba (6.6), in jih ocenili s klasično metodo najmanjše vsote kvadratov odstopanj za vsak podprostor Ω_i posebej.

Pri opisanem pristopu se pojavi problem porazdelitve podatkov za identifikacijo v posameznem podprostoru: podatki z roba podprostora vplivajo na oceno enako kot podatki s sredine podprostora. Ker podatki v splošnem niso enakomerno porazdeljeni po posameznem podprostoru, lahko za take podprostore dobimo neustrezne ocene parametrov. Poleg tega lahko problem predstavlja tudi število identifikacijskih podatkov, ki pripadajo posameznemu podprostoru: če posameznemu podprostoru pripada premalo identifikacijskih podatkov, bo ocena podmodela za ta podprostor slaba ali pa je celo sploh ne bomo mogli določiti.

Pri 2. pristopu bomo zato za identifikacijo parametrov uporabili metodo, ki je opisana v razdelkih 5.3.4 in 5.3.5 ter se naslanja na teorijo mehkih sistemov. Ista metoda je uporabna tudi za identifikacijo hibridnega mehkega modela. Za konkreten primer identifikacije modela temperature v plašču šaržnega reaktorja T_w je metoda podrobno opisana v razdelku 7.4.

Parametre odsekoma afinega modela, ki nastopajo v enačbi (6.9), smo torej za odsekoma afini model prilagodili iz zapisa hibridnega mehkega modela – glej razdelek 7.4 ter enačbi (7.8) in (7.9). Podani so v enačbi (6.11).

$$\begin{aligned}
\Omega_1 : A_1 &= 0,9453; B_1 = [0,0376 \ 19,6748]; f_1 = 0,3021 \\
\Omega_2 : A_2 &= 0,9431; B_2 = [0,0458 \ 16,7605]; f_2 = 0,2160 \\
\Omega_3 : A_3 &= 0,9429; B_3 = [0,0395 \ 10,5969]; f_3 = 0,5273 \\
\Omega_4 : A_4 &= 0,9429; B_4 = [0,0339 \ 3,9536]; f_4 = 1,2701 \\
\Omega_5 : A_5 &= 0,7910; B_5 = [0,0225 \ 1,6856]; f_5 = 12,0404 \\
\Omega_6 : A_6 &= 0,9803; B_6 = [0,0025 \ -0,0704]; f_6 = 0,2707 \\
\Omega_7 : A_7 &= 0,9740; B_7 = [0,0153 \ -0,6956]; f_7 = 0,2033 \\
\Omega_8 : A_8 &= 0,9322; B_8 = [0,0466 \ -7,8013]; f_8 = 0,5650 \\
\Omega_9 : A_9 &= 0,9076; B_9 = [0,0446 \ -12,2555]; f_9 = 1,9179 \\
\Omega_{10} : A_{10} &= 0,8945; B_{10} = [0,0111 \ -18,7457]; f_{10} = 5,6129
\end{aligned} \tag{6.11}$$

6.4.3 Odsekoma afini model – 3. pristop

Razdelitev vhodnega prostora modela

Vhodni prostor modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$ v sistemu, ki opisuje temperaturo v plašču šaržnega reaktorja T_w , določa definicijsko območje spremenljivk T_w , T , k_M , k_C in k_H in je podan v enačbi (6.5).

Kot pri pristopih v razdelkih 6.4.1 in 6.4.2 je tudi pri tem pristopu prva in očitna delitev vhodnega prostora modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$ na podprostore glede na način delovanja, tj. glede na položaj zapornih ventilov.

- Prvi način delovanja je, ko je sveža vhodna voda vroča, tj. $T_{in} = T_H$ oz. $k_C(k) = 0$ in $k_H(k) = 1$.
- Drugi način delovanja je, ko je sveža vhodna voda hladna, tj. $T_{in} = T_C$ oz. $k_C(k) = 1$ in $k_H(k) = 0$.

V naslednjem koraku moramo vhodni prostor modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$ dodatno razdeliti tako, da bo končni odsekoma afini model z uporabo več afinih modelov v posameznih podpodročjih zaobjel nelinearno dinamiko v sistemu. V tem primeru za vsak način delovanja preostanek vhodnega prostora modela (6.4) $\mathcal{D}_{[T_w, T, k_M]^T}$ razdelimo na za opis nelinearne dinamike čim bolj ustrezne konveksne poliedre, pri čemer moramo zajeti celotno območje delovanja, tj. celoten preostanek vhodnega prostora modela (6.4) $\mathcal{D}_{[T_w, T, k_M]^T}$. Vsak konveksni polieder bo omejil podprostor, v katerem bomo dinamiko opisali z odsekoma afnim modelom.

Mehko rojenje

K segmentaciji preostanka vhodnega prostora modela (6.4) $\mathcal{D}_{[T_w, T, k_M]^T}$ pristopimo z metodo *mehkega rojenja c-središč* (angl. *fuzzy c-means clustering*) – za podrobnejši opis glej tudi razdelek 5.3.

Z rojenjem, ki ga izvajamo nad prostorom vhodnih in izhodnih spremenljivk modela (6.4), želimo podatke za identifikacijo razdeliti na več *mehkih rojev*, tako da vsak podatek za identifikacijo pripada posameznemu mehkeemu roju z določeno stopnjo pripadnosti, ki je odvisna od razdalj do središč posameznih mehkih rojev. Vsak roj želimo v končni fazi predstaviti z afinim podmodelom, območje delovanja posameznega podmodela pa v prostoru vhodnih spremenljivk modela (6.4) omejiti s konveksnimi poliedri.

Pred rojenjem podatke za identifikacijo normiramo, tj. skaliramo njihove vrednosti tako, da so njihove preslikave v intervalu $[0, 1]$. Prostor, nad katerim izvajamo rojenje, lahko torej zapišemo v enačbi (6.12).

$$\mathcal{D}_{[\check{T}_w(\tau), \check{T}(\tau), \check{k}_M(\tau)]^T} \times \mathcal{D}_{\check{T}_w(\tau+1)} \equiv [0, 1]^4 \quad (6.12)$$

Označba $\check{}$ nad spremenljivko pomeni, da se zapis nanaša na normirani prostor.

V našem primeru smo pri mehkeem rojenju uporabili vrednost parametra $m = 1, 25$.

Z rojenjem razdelimo podatke za identifikacijo glede na vhodne spremenljivke podmodela $T_w(\tau)$, $T(\tau)$ in $k_M(\tau)$ ter izhodno spremenljivko $T_w(\tau + 1)$ na mehke roje. Indeks τ tu označuje časovni korak posameznega podatka za identifikacijo.

Z algoritmom za rojenje določimo tudi središča mehkih rojev v normiranem prostoru (6.12) \check{c}_j , kjer je $j \in \{1, 2, \dots, C\}$, in vsaki točki iz množice podatkov za identifikacijo pripišemo ustrezne vrednosti pripadnostnih funkcij. Za vsak način delovanja sistema določimo po C rojev.

Projekcije središč mehkih rojev v vhodni prostor modela in pretvorba v originalne koordinate

Središča rojev (in ustrezne pripadnostne funkcije), ki so rezultat algoritma za mehko rojenje, so določena v normiranem vhodno-izhodnem prostoru modela, ki

ga podaja izraz (6.12). Tako določene pripadnostne funkcije lahko neposredno uporabimo pri identifikaciji odsekoma afinega modela, problem pa se pojavi pri uporabi odsekoma afinega modela za predikcijo obnašanja sistema.

V naslednjem koraku moramo torej središča rojev projicirati v vhodni prostor modela (6.4) $\mathcal{D}_{[\check{T}_w, \check{T}, \check{k}_M]^T}$. Središča rojev, določena v normiranem vhodnem prostoru modela, so za primer $C = 5$ podana v enačbi (6.13). Enačba podaja rezultate obeh rojenj:

- središča rojev od \check{c}_1 do \check{c}_5 za prvi način delovanja ($k_C = 0$ in $k_H = 1$);
- središča rojev od \check{c}_6 do \check{c}_{10} za drugi način delovanja ($k_C = 1$ in $k_H = 0$).

$$\begin{aligned}
 \check{c}_1 &= [0, 9495 \ 0, 5507 \ 1, 0000] \\
 \check{c}_2 &= [0, 4880 \ 0, 4765 \ 0, 0305] \\
 \check{c}_3 &= [0, 6715 \ 0, 6935 \ 0, 0500] \\
 \check{c}_4 &= [0, 1148 \ 0, 0980 \ 0, 0053] \\
 \check{c}_5 &= [0, 3180 \ 0, 2655 \ 0, 0218] \\
 \check{c}_6 &= [0, 3718 \ 0, 5257 \ 0, 0328] \\
 \check{c}_7 &= [0, 2180 \ 0, 2939 \ 0, 0279] \\
 \check{c}_8 &= [0, 0818 \ 0, 1010 \ 0, 0412] \\
 \check{c}_9 &= [0, 0147 \ 0, 1273 \ 1, 0000] \\
 \check{c}_{10} &= [0, 0562 \ 0, 4671 \ 1, 0000]
 \end{aligned} \tag{6.13}$$

Razmejitvene hiperravnine

Glede na rezultate rojenja lahko določimo delitev vhodnega prostora odsekoma afinega modela (6.4) $\mathcal{D}_{[T_w, T, k_M]^T}$. Prostor razdelimo na konveksne poliedre, ki jih

določajo razmejitvene hiperravnine.² Za vsak par podprostorov Ω_i in Ω_j lahko določimo razmejitveno hiperravnino, kot jo podaja enačba (6.14).

$$(\check{x} - \check{s}_{ij}) \cdot \check{n}_{ij} = 0 \quad (6.14)$$

V enačbi (6.14) pomeni \check{x} krajevni vektor splošne točke v normiranem prostoru stanj in vhodov, ki pripada razmejitveni hiperravnini, \check{s}_{ij} določa poljubno točko na razmejitveni hiperravnini, \check{n}_{ij} pa je normala razmejitvene hiperravnine. Zaradi jasnosti lahko, ne da bi izgubili splošnost, predpostavimo, da je točka \check{s}_{ij} tista točka na razmejitveni hiperravnini (6.14), ki leži na premici skozi središči i -tega in j -tega roja \check{c}_i in \check{c}_j .

Točko \check{s}_{ij} in vektor \check{n}_{ij} lahko določimo iz središč rojev, kot kažeta enačbi (6.15) in (6.16).

$$\check{s}_{ij} = \frac{\check{c}_i + \check{c}_j}{2} \quad (6.15)$$

$$\check{n}_{ij} = \check{c}_j - \check{c}_i \quad (6.16)$$

Če v enačbi (6.14) znak $=$ zamenjamo z znakom $<$ oz. $>$, dobimo neenačbo, ki določa hiperpolprostor, ki mu pripada središče roja \check{c}_i oz. \check{c}_j . Oba hiperpolprostora ločuje razmejitvena hiperravnina. Na ta način razdelimo hiperprostor – za vsak par (i, j) , kjer sta $i, j \in \{1, 2, \dots, C\}$ – na pare hiperpolprostorov. Pri tem velja, da enemu hiperpolprostoru pripadajo vse tiste točke hiperprostora, ki so bližje enemu od središč.

Omenimo, da so razmejitvene hiperravnine v enačbah (6.14), (6.15) in (6.16) določene v normiranem vhodnem prostoru $\mathcal{D}_{[T_w, T, k_M]^T}$.

²Glede na to, da imamo v našem primeru opravka z razdelitvijo trirazsežnega vhodnega prostora $\mathcal{D}_{[T_w, T, k_M]^T}$, so tudi razmejitvene hiperravnine definirane v treh dimenzijah.

Normiranje je preslikava, ki točko iz originalnega koordinatnega sistema $\mathcal{D}_{[T_w, T, k_M]^T}$ preslika v točko v normiranem prostoru $\mathcal{D}_{[\tilde{T}_w, \tilde{T}, \tilde{k}_M]^T}$. V našem primeru gre za afino preslikavo, tj. kompozitum linearne transformacije in translacije, zato lahko krajevne vektorje posameznih točk v originalnih koordinatah preprosto določimo z inverznim normiranjem, ki ga ni težko izpeljati.

Pozoren pa je treba biti pri pretvorbi enačb (6.14) in (6.16). Ortogonalnost – tj. skalarni produkt, s katerim so določene razmejitvene hiperravnine v enačbi (6.14) – se namreč pri afinih preslikavah v splošnem ne ohranja. Ob upoštevanju tega dejstva pa lahko končni rezultat razdelitve vhodnega prostora modela (6.4) vseeno zapišemo v koordinatah originalnega prostora $\mathcal{D}_{[T_w, T, k_M]^T}$.

Razdelitev prostora $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$ na konveksne poliedre Ω_i torej podaja enačba (6.17).

$$\begin{aligned}
 & [T_w(k), T(k), k_M(k), k_C(k), k_H(k)]^T \in \Omega_i; \\
 & \text{če velja} \quad \begin{aligned}
 & H_i T_w(k) + J_i [T(k), k_M(k), k_C(k), k_H(k)]^T \leq K_i \\
 & \tilde{H}_i T_w(k) + \tilde{J}_i [T(k), k_M(k), k_C(k), k_H(k)]^T < \tilde{K}_i
 \end{aligned} \quad (6.17) \\
 & \text{za } i = 1, \dots, 10
 \end{aligned}$$

Parametri razdelitve prostora $H_i, J_i, K_i, \tilde{H}_i, \tilde{J}_i$ in \tilde{K}_i so za vsak $i = 1, \dots, 10$ podani (v originalnih koordinatah) v enačbah od (6.18) do (6.27).

$$\begin{aligned}
 H_1 &= \begin{bmatrix} 0 \end{bmatrix}; & J_1 &= \begin{bmatrix} 0 & 0 & 1 & -1 \end{bmatrix}; & K_1 &= \begin{bmatrix} 0 \end{bmatrix} \\
 \tilde{H}_1 &= \begin{bmatrix} -0,0076 \\ -0,0046 \\ -0,0138 \\ -0,0105 \end{bmatrix}; & \tilde{J}_1 &= \begin{bmatrix} -0,0013 & -0,9695 & 0 & 0 \\ 0,0026 & -0,9500 & 0 & 0 \\ -0,0082 & -0,9947 & 0 & 0 \\ -0,0051 & -0,9781 & 0 & 0 \end{bmatrix}; & \tilde{K}_1 &= \begin{bmatrix} -0,9845 \\ -0,6595 \\ -1,3758 \\ -1,2181 \end{bmatrix} \\
 & & & & & (6.18)
 \end{aligned}$$

$$\begin{aligned}
H_2 &= \begin{bmatrix} 0,0076 \\ 0 \end{bmatrix}; & J_2 &= \begin{bmatrix} 0,0013 & 0,9695 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}; & K_2 &= \begin{bmatrix} 0,9845 \\ 0 \end{bmatrix} \\
\tilde{H}_2 &= \begin{bmatrix} 0,0030 \\ -0,0062 \\ -0,0028 \end{bmatrix}; & \tilde{J}_2 &= \begin{bmatrix} 0,0039 & 0,0195 & 0 & 0 \\ -0,0068 & -0,0252 & 0 & 0 \\ -0,0038 & -0,0087 & 0 & 0 \end{bmatrix}; & \tilde{K}_2 &= \begin{bmatrix} 0,3249 \\ -0,3913 \\ -0,2336 \end{bmatrix} \\
& & & & & (6.19)
\end{aligned}$$

$$\begin{aligned}
H_3 &= \begin{bmatrix} 0,0046 \\ -0,0030 \\ 0 \end{bmatrix}; & J_3 &= \begin{bmatrix} -0,0026 & 0,9500 & 0 & 0 \\ -0,0039 & -0,0195 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}; & K_3 &= \begin{bmatrix} 0,6595 \\ -0,3249 \\ 0 \end{bmatrix} \\
\tilde{H}_3 &= \begin{bmatrix} -0,0092 \\ -0,0059 \end{bmatrix}; & \tilde{J}_3 &= \begin{bmatrix} -0,0107 & -0,0448 & 0 & 0 \\ -0,0077 & -0,0282 & 0 & 0 \end{bmatrix}; & \tilde{K}_3 &= \begin{bmatrix} -0,7163 \\ -0,5586 \end{bmatrix} \\
& & & & & (6.20)
\end{aligned}$$

$$H_4 = \begin{bmatrix} 0,0138 \\ 0,0062 \\ 0,0092 \\ 0 \end{bmatrix}; \quad J_4 = \begin{bmatrix} 0,0082 & 0,9947 & 0 & 0 \\ 0,0068 & 0,0252 & 0 & 0 \\ 0,0107 & 0,0448 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}; \quad K_4 = \begin{bmatrix} 1,3758 \\ 0,3913 \\ 0,7163 \\ 0 \end{bmatrix}$$

$$\tilde{H}_4 = \begin{bmatrix} 0,0034 \end{bmatrix}; \quad \tilde{J}_4 = \begin{bmatrix} 0,0030 & 0,0166 & 0 & 0 \end{bmatrix}; \quad \tilde{K}_4 = \begin{bmatrix} 0,1577 \end{bmatrix} \quad (6.21)$$

$$\begin{aligned}
H_5 &= \begin{bmatrix} 0,0105 \\ 0,0028 \\ 0,0059 \\ -0,0034 \\ 0 \end{bmatrix}; \quad J_5 = \begin{bmatrix} 0,0051 & 0,9781 & 0 & 0 \\ 0,0038 & 0,0087 & 0 & 0 \\ 0,0077 & 0,0282 & 0 & 0 \\ -0,0030 & -0,0166 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}; \quad K_5 = \begin{bmatrix} 1,2181 \\ 0,2336 \\ 0,5586 \\ -0,1577 \\ 0 \end{bmatrix} \\
\tilde{H}_5 &= []; \quad \tilde{J}_5 = []; \quad \tilde{K}_5 = []
\end{aligned} \tag{6.22}$$

$$\begin{aligned}
H_6 &= []; \quad J_6 = []; \quad K_6 = [] \\
\tilde{H}_6 &= \begin{bmatrix} 0 \\ -0,0026 \\ -0,0049 \\ -0,0060 \\ -0,0053 \end{bmatrix}; \quad \tilde{J}_6 = \begin{bmatrix} 0 & 0 & -1 & 1 \\ -0,0042 & -0,0048 & 0 & 0 \\ -0,0077 & 0,0084 & 0 & 0 \\ -0,0072 & 0,9672 & 0 & 0 \\ -0,0011 & 0,9672 & 0 & 0 \end{bmatrix}; \quad \tilde{K}_6 = \begin{bmatrix} 0 \\ -0,2260 \\ -0,3570 \\ 0,1343 \\ 0,3246 \end{bmatrix}
\end{aligned} \tag{6.23}$$

$$H_7 = [0,0026]; \quad J_7 = [0,0042 \quad 0,0048 \quad 0 \quad 0]; \quad K_7 = [0,2260]$$

$$\begin{aligned}
\tilde{H}_7 &= \begin{bmatrix} 0 \\ -0,0023 \\ -0,0034 \\ -0,0027 \end{bmatrix}; \quad \tilde{J}_7 = \begin{bmatrix} 0 & 0 & -1 & 1 \\ -0,0035 & 0,0132 & 0 & 0 \\ -0,0030 & 0,9721 & 0 & 0 \\ 0,0031 & 0,9721 & 0 & 0 \end{bmatrix}; \quad \tilde{K}_7 = \begin{bmatrix} 0 \\ -0,1310 \\ 0,3603 \\ 0,5507 \end{bmatrix}
\end{aligned} \tag{6.24}$$

$$\begin{aligned}
H_8 &= \begin{bmatrix} 0,0049 \\ 0,0023 \end{bmatrix}; & J_8 &= \begin{bmatrix} 0,0077 & -0,0084 & 0 & 0 \\ 0,0035 & -0,0132 & 0 & 0 \end{bmatrix}; & K_8 &= \begin{bmatrix} 0,3570 \\ 0,1310 \end{bmatrix} \\
\tilde{H}_8 &= \begin{bmatrix} 0 \\ -0,0011 \\ -0,0004 \end{bmatrix}; & \tilde{J}_8 &= \begin{bmatrix} 0 & 0 & -1 & 1 \\ 0,0005 & 0,9588 & 0 & 0 \\ 0,0066 & 0,9588 & 0 & 0 \end{bmatrix}; & \tilde{K}_8 &= \begin{bmatrix} 0 \\ 0,4913 \\ 0,6816 \end{bmatrix}
\end{aligned} \tag{6.25}$$

$$\begin{aligned}
H_9 &= \begin{bmatrix} 0,0060 \\ 0,0034 \\ 0,0011 \end{bmatrix}; & J_9 &= \begin{bmatrix} 0,0072 & -0,9672 & 0 & 0 \\ 0,0030 & -0,9721 & 0 & 0 \\ -0,0005 & -0,9588 & 0 & 0 \end{bmatrix}; & K_9 &= \begin{bmatrix} -0,1343 \\ -0,3603 \\ -0,4913 \end{bmatrix} \\
\tilde{H}_9 &= \begin{bmatrix} 0 \\ 0,0007 \end{bmatrix}; & \tilde{J}_9 &= \begin{bmatrix} 0 & 0 & -1 & 1 \\ 0,0062 & -0,0000 & 0 & 0 \end{bmatrix}; & \tilde{K}_9 &= \begin{bmatrix} 0 \\ 0,1903 \end{bmatrix}
\end{aligned} \tag{6.26}$$

$$\begin{aligned}
H_{10} &= \begin{bmatrix} 0,0053 \\ 0,0027 \\ 0,0004 \\ -0,0007 \end{bmatrix}; & J_{10} &= \begin{bmatrix} 0,0011 & -0,9672 & 0 & 0 \\ -0,0031 & -0,9721 & 0 & 0 \\ -0,0066 & -0,9588 & 0 & 0 \\ -0,0062 & 0,0000 & 0 & 0 \end{bmatrix}; & K_{10} &= \begin{bmatrix} -0,3246 \\ -0,5507 \\ -0,6816 \\ -0,1903 \end{bmatrix} \\
\tilde{H}_{10} &= \begin{bmatrix} 0 \end{bmatrix}; & \tilde{J}_{10} &= \begin{bmatrix} 0 & 0 & -1 & 1 \end{bmatrix}; & \tilde{K}_{10} &= \begin{bmatrix} 0 \end{bmatrix}
\end{aligned} \tag{6.27}$$

Ker zaporna ventila k_C in k_H naenkrat ne moreta zavzeti iste vrednosti, veljata enačbi (6.7) in (6.8). Zato lahko rečemo, da je vhodni prostor modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$ ustrezno razdeljen na konveksne poliedre, ki so osnova za zapis modela v obliki odsekoma afinega sistema.

Lokalni afini modeli

Za vsak podprostor Ω_i vhodnega prostora modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$, ki smo ga določili v prejšnjem razdelku, lahko zapišemo lokalni afini model, kot ga podaja enačba (6.28).

$$T_w(k+1) = A_i T_w(k) + B_i \begin{bmatrix} T(k) \\ k_M(k) \end{bmatrix} + f_i \quad (6.28)$$

za $[T_w, T, k_M, k_C, k_H]^T \in \Omega_i$

Tudi v enačbi (6.28) ni izhodnega dela, ki ga formulacija odsekoma afinega modela vsebuje – glej enačbo (4.2). Ker stanje sistema že samo po sebi predstavlja tudi izhod sistema ($y(k) = x(k) = T_w(k)$), zapis izhodnega dela ni smiseln. Kljub temu lahko podamo: $C_i = 1$, $D_i = 0$ in $g_i = 0$ za vsak Ω_i , kjer je $i \in \{1, \dots, 10\}$.

Parametri modela

Podatke za identifikacijo, ki smo jih dobili, kot je opisano v razdelku 6.1, pred ocenjevanjem parametrov razdelimo glede na razdelitev vhodnega prostora modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$ na podprostore Ω_i , ki jih določajo enačbe (6.17)–(6.27).

Parametre ocenimo s klasično metodo najmanjše vsote kvadratov odstopanj za vsak podprostor Ω_i posebej. Parametri odsekoma afinega modela, ki nastopajo v enačbi (6.28), so podani v enačbi (6.29).

$$\begin{aligned}
\Omega_1 : A_1 &= 0,6314; B_1 = [0,0400 \ 238193]; f_1 = -290512 \\
\Omega_2 : A_2 &= 0,9448; B_2 = [0,0376 \ 9,2129]; f_2 = 0,5571 \\
\Omega_3 : A_3 &= 0,9392; B_3 = [0,0379 \ 6,8623]; f_3 = 0,8985 \\
\Omega_4 : A_4 &= 0,9353; B_4 = [0,0541 \ 19,8684]; f_4 = 0,1955 \\
\Omega_5 : A_5 &= 0,9368; B_5 = [0,0434 \ 13,9952]; f_5 = 0,5135 \\
\Omega_6 : A_6 &= 0,9224; B_6 = [0,0485 \ -8,3390]; f_6 = 0,8485 \\
\Omega_7 : A_7 &= 0,9383; B_7 = [0,0436 \ -4,0763]; f_7 = 0,3900 \\
\Omega_8 : A_8 &= 0,9501; B_8 = [0,0337 \ -1,4182]; f_8 = 0,2888 \\
\Omega_9 : A_9 &= 0,6267; B_9 = [0,0406 \ -13128]; f_9 = 7153 \\
\Omega_{10} : A_{10} &= 0,6312; B_{10} = [0,0400 \ -194,05]; f_{10} = 663,36
\end{aligned} \tag{6.29}$$

6.4.4 Odsekoma afini model – 4. pristop

Postopek modeliranja

Pri modeliranju odsekoma afinega modela s 4. pristopom postopamo popolnoma enako kot pri 3. pristopu, ki je opisan v razdelku 6.4.3. Vhodni prostor modela (6.4) $\mathcal{D}_{[T_w, T, k_M, k_C, k_H]^T}$ je torej razdeljen na podprostore Ω_i , ki jih določajo enačbe (6.17)–(6.27), odsekoma afini model pa podaja enačba (6.28). Edina razlika med pristopoma je v ocenjevanju parametrov modela.

Parametri modela

Podatke za identifikacijo smo dobili, kot je opisano v razdelku 6.1. Kot smo omenili že v razdelku 6.4.2, se pri pristopu s klasično metodo najmanjše vsote kvadratov odstopanj pojavi problem porazdelitve podatkov za identifikacijo v posameznem podprostoru: podatki z roba podprostora vplivajo na oceno enako kot podatki s sredine podprostora. Ker podatki v splošnem niso enakomerno porazdeljeni po posameznem podprostoru, lahko za take podprostore dobimo neustrezne

ocene parametrov. Poleg tega lahko problem predstavlja tudi število identifikacijskih podatkov, ki pripadajo posameznemu podprostoru: če posameznemu podprostoru pripada premalo identifikacijskih podatkov, bo ocena podmodela za ta podprostor slaba ali pa je celo sploh ne bomo mogli določiti.

Pri tem pristopu bomo zato za identifikacijo parametrov uporabili metodo, ki je opisana v razdelku 5.3 in se naslanja na teorijo mehkih sistemov. Ustrezne vrednosti pripadnostnih funkcij za vsak podatek za identifikacijo določimo iz rezultatov rojenja: odvisne so od oddaljenosti točk, ki predstavljajo podatek za identifikacijo, od središča posameznega roja. Parametri odsekoma afinega modela, ki nastopajo v enačbi (6.28), so podani v enačbi (6.30).

$$\begin{aligned}
\Omega_1 : A_1 &= 0,6312; B_1 = [0,0400 \ 1,7115]; f_1 = 22,4874 \\
\Omega_2 : A_2 &= 0,9257; B_2 = [0,0513 \ 12,5240]; f_2 = 0,7210 \\
\Omega_3 : A_3 &= 0,9361; B_3 = [0,0384 \ 6,9758]; f_3 = 1,0316 \\
\Omega_4 : A_4 &= 0,9404; B_4 = [0,0490 \ 19,3915]; f_4 = 0,1955 \\
\Omega_5 : A_5 &= 0,9277; B_5 = [0,0515 \ 16,8357]; f_5 = 0,5233 \\
\Omega_6 : A_6 &= 0,9059; B_6 = [0,0630 \ -10,8609]; f_6 = 0,8869 \\
\Omega_7 : A_7 &= 0,9337; B_7 = [0,0468 \ -4,5854]; f_7 = 0,4228 \\
\Omega_8 : A_8 &= 0,9462; B_8 = [0,0359 \ -1,5090]; f_8 = 0,3163 \\
\Omega_9 : A_9 &= 0,6279; B_9 = [0,0404 \ -15,3616]; f_9 = 19,3905 \\
\Omega_{10} : A_{10} &= 0,6312; B_{10} = [0,0400 \ -14,9129]; f_{10} = 18,9084
\end{aligned} \tag{6.30}$$

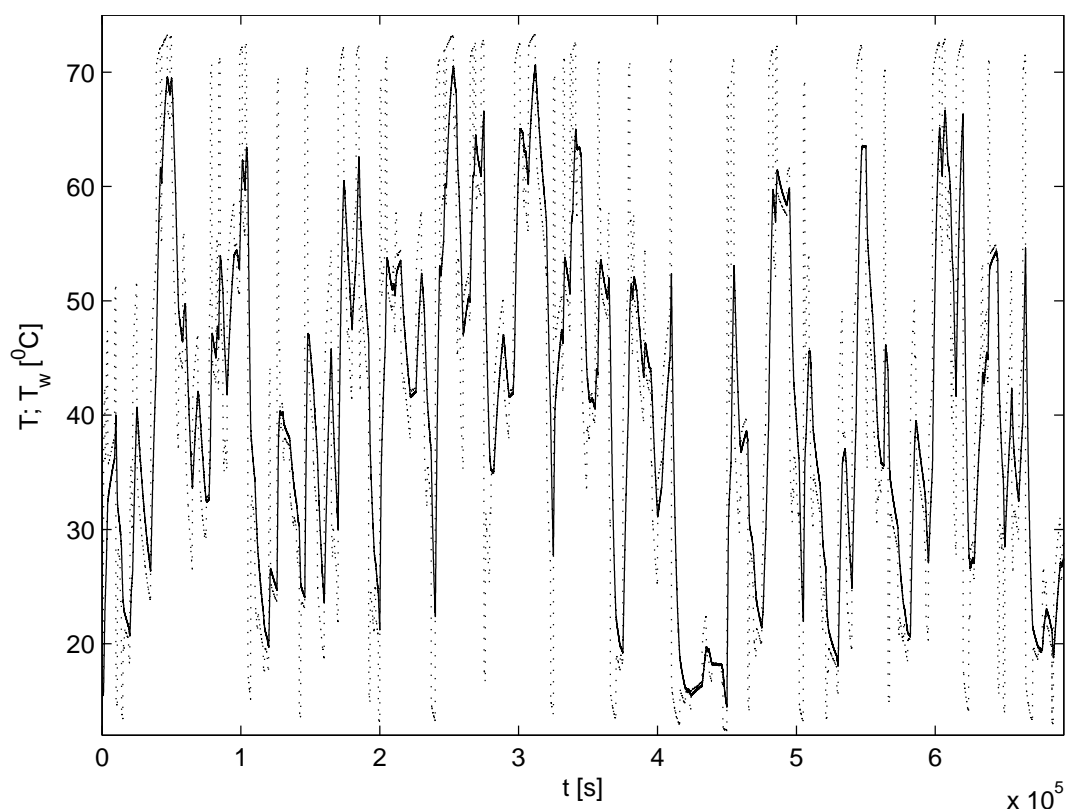
6.5 Vrednotenje

Odsekoma afine modele, ki so rezultat pristopov, opisanih v razdelkih 6.4.1, 6.4.2, 6.4.3 in 6.4.4, smo ovrednotili tako, da smo jih primerjali medsebojno in s simulacijskim modelom šaržnega reaktorja.

6.5.1 Podatki za vrednotenje

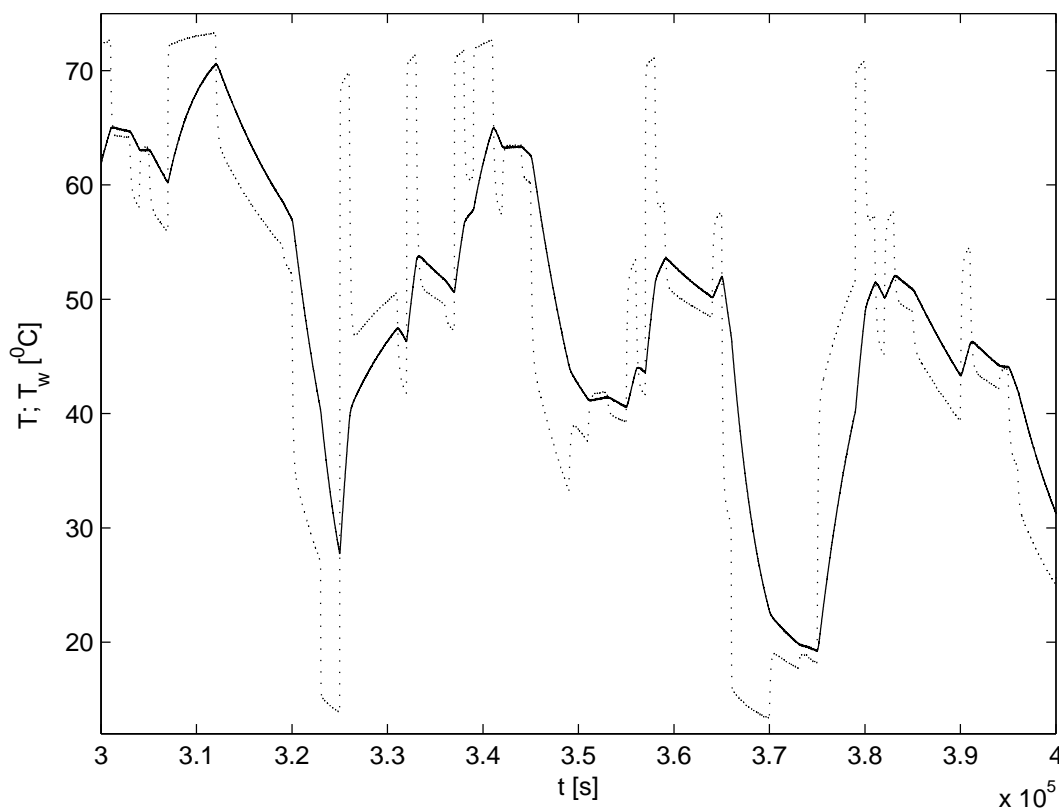
Podatke za vrednotenje smo tvorili podobno kot podatke za identifikacijo v razdelku 6.1: v podatke za vrednotenje smo zajeli odziv šaržnega reaktorja na psevdonaključni vhodni signal. Posneli smo oba merljiva izhoda, tj. temperaturo v jedru T in temperaturo v plašču šaržnega reaktorja T_w .

Slika 6.6 prikazuje izhodna signala za vrednotenje T in T_w , na sliki 6.7 pa je prikazan še približan izsek poteka signalov.



Slika 6.6: Temperatura v jedru T (polna črta) in plašču T_w (prekinjena črta) šaržnega reaktorja.

Slika 6.8 prikazuje vhodne signale za vrednotenje k_M , k_H in k_C , na sliki 6.9 pa je prikazan še približan izsek poteka signalov.



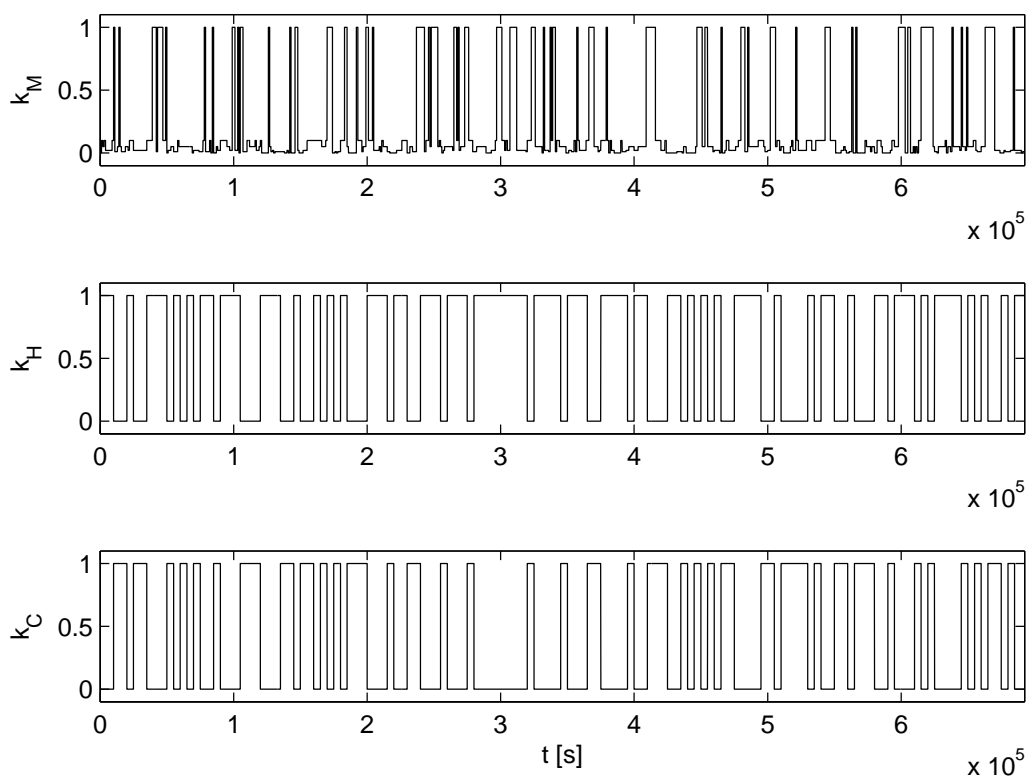
Slika 6.7: Temperatura v jedru T (polna črta) in plašču T_w (prekinjena črta) šaržnega reaktorja – izsek.

6.5.2 Eksperimenti

Primerjali smo odzive odsekoma afinih modelov, ki so rezultat pristopov, opisanih v razdelkih 6.4.1, 6.4.2, 6.4.3 in 6.4.4, na vhodne signale za vrednotenje, ki so opisani v razdelku 6.5.1. V eksperimentih smo uporabili omenjene modele za napovedovanje poteka trajektorij temperature v plašču šaržnega reaktorja T_w , in sicer za h korakov naprej, pa tudi za celoten čas poteka signalov za vrednotenje.

Omenimo, da smo pri eksperimentih uporabljali celoten model šaržnega reaktorja, tj. podmodel za temperaturo v jedru T in podmodel za temperaturo v plašču T_w , vendar je pri vseh pristopih podmodel za temperaturo v jedru T , ki je opisan v razdelku 6.3, isti.

Sliki 6.10 oz. 6.12 prikazujeta izsek trajektorije temperature v plašču šaržnega

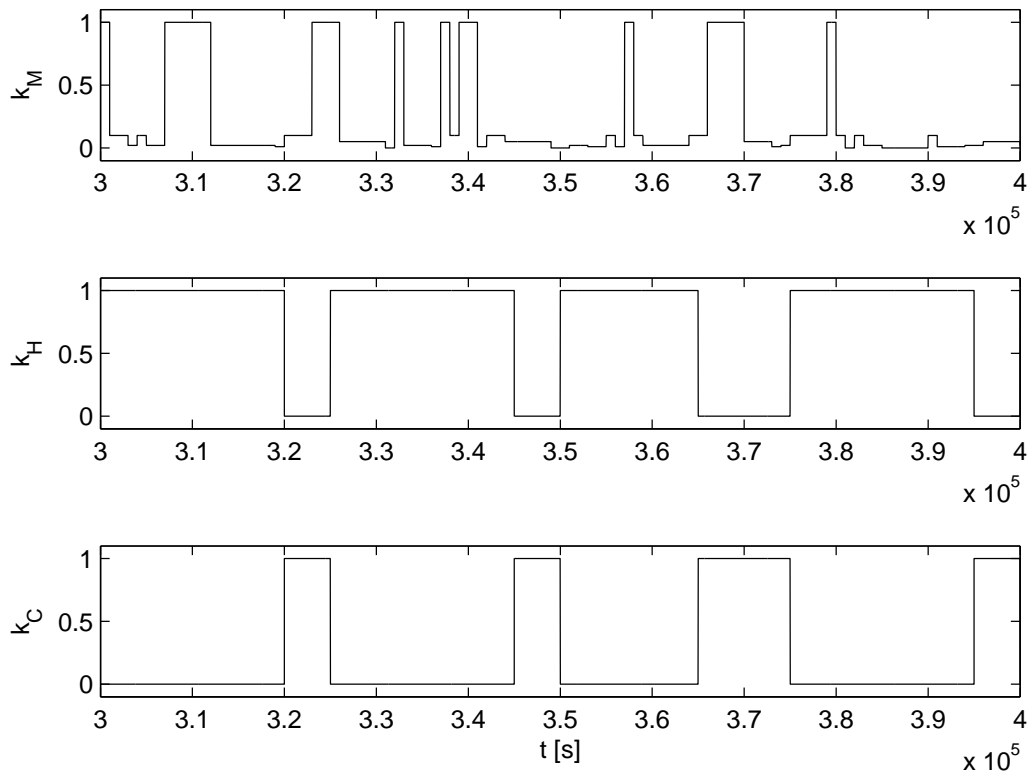
Slika 6.8: Vhodni signali k_M , k_H in k_C .

reaktorja T_w in odziv modela, ki smo ga dobili z 2. oz. s 4. pristopom, ki so opisani v razdelkih 6.4.2 oz. 6.4.4, na vhodne signale za vrednotenje. Za primerjavo je na slikah s prekinjeno črto označen dejanski odziv na vhodne signale za vrednotenje.

Slik, ki bi prikazovale odziv modelov, ki smo ju dobili s 1. oz. s 3. pristopom, ki sta opisana v razdelku 6.4.1 oz. 6.4.3, nismo vključili, ker sta modela nestabilna.

Tabela 6.1 prikazuje rezultate simulacij z modelom, ki smo ga dobili z 2. pristopom, opisanim v razdelku 6.4.2, tabela 6.2 pa rezultate simulacij z modelom, ki smo ga dobili s 4. pristopom, opisanim v razdelku 6.4.4.

Tabel, ki bi prikazovale odziv modelov, ki smo ju dobili s 1. oz. s 3. pristopom, ki sta opisana v razdelku 6.4.1 oz. 6.4.3, nismo vključili, ker sta modela

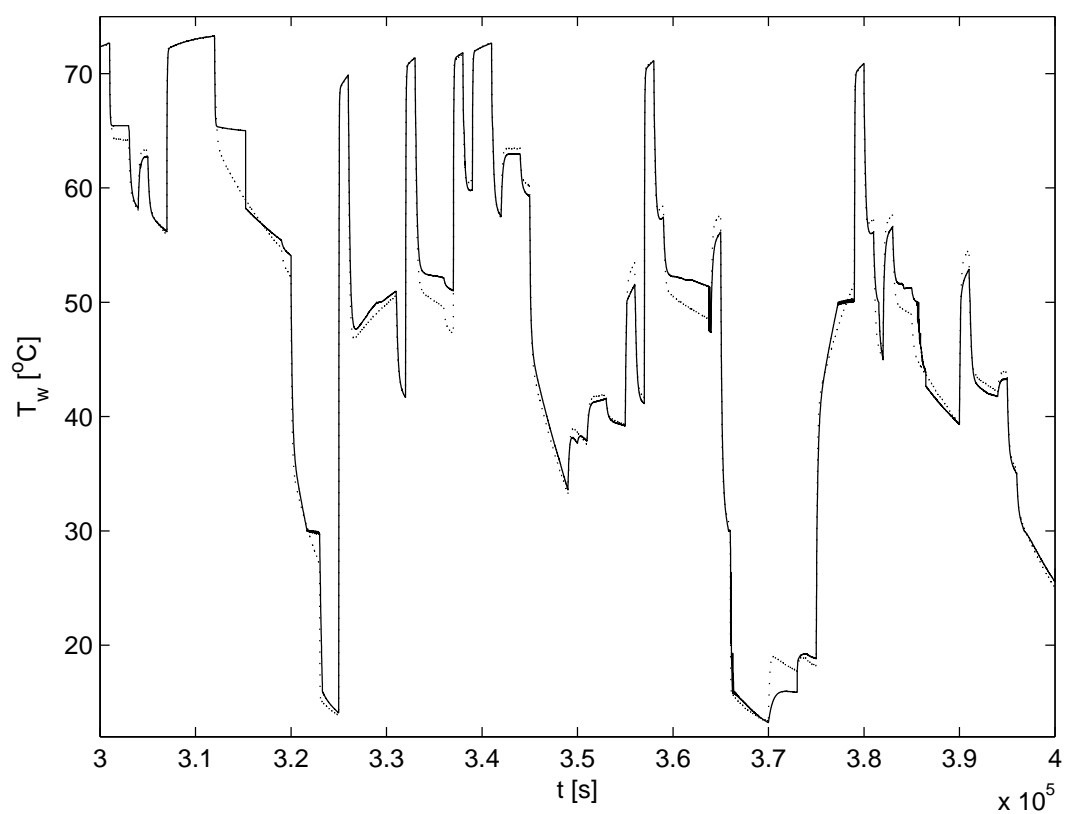
Slika 6.9: Vhodni signali k_M , k_H in k_C – izsek.

nestabilna.

V tabelah 6.1 in 6.2 so v vsaki vrstici podani rezultati simulacij, in sicer glede na število predikcijskih korakov h , ki smo jih računali v posameznem eksperimentu. Označba $h = \infty$ pomeni, da gre v tem primeru za celotno simulacijo, tj. odziv modela na vhodne signale za vrednotenje, brez sprotnih popravkov stanj, kakršne v vsakem koraku naredimo pri h -koračni predikciji za $h < \infty$.

Stolpci v tabelah 6.1 in 6.2 predstavljajo parametre za vrednotenje modelov, kot so navedeni spodaj:

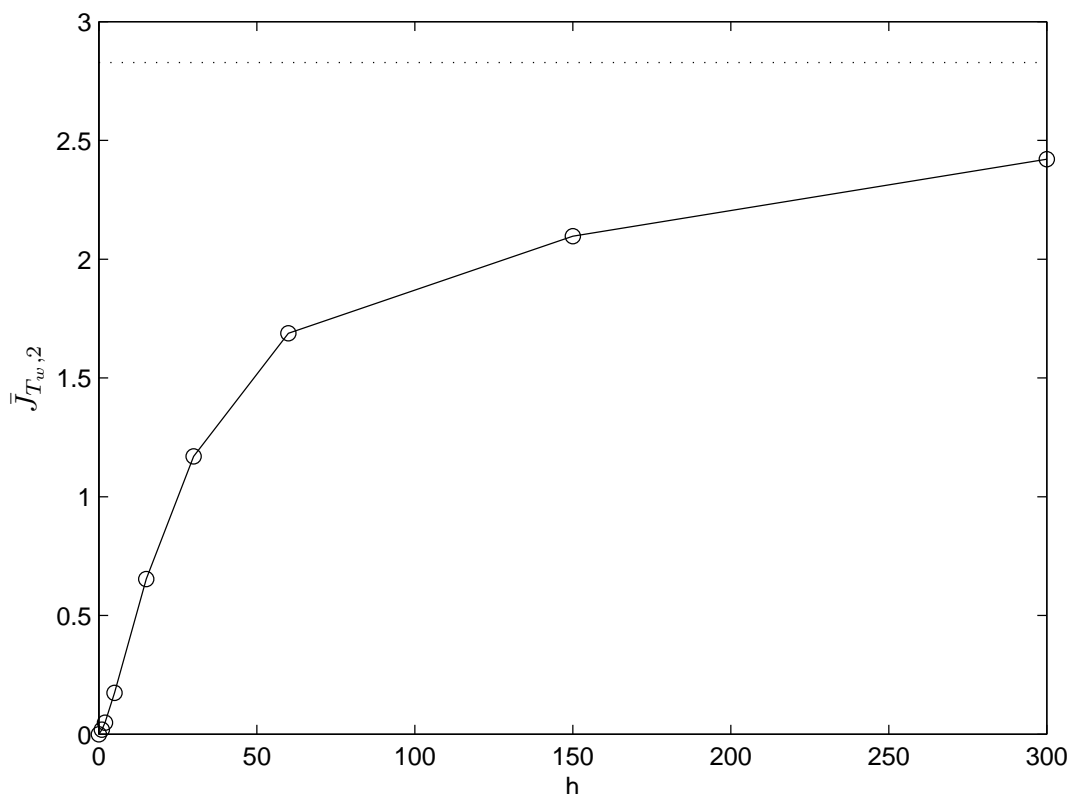
- $\bar{J}_{T_w,i}$ predstavlja povprečje kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w za i -ti pristop;
- $\bar{J}_{T,i}$ predstavlja povprečje kvadratov odstopanj vrednosti temperature v je-



Slika 6.10: Odziv modela na vhodne signale za vrednotenje (polna črta) in dejanski odziv (prekinjena črta) – 2. pristop.

h	$\bar{J}_{T_w,2}$	$\bar{J}_{T,2}$	$t_{sim,2}$ [s]
1	0,0190	0,0020	4,04
2	0,0488	0,0020	7,71
5	0,1743	0,0020	18,81
15	0,6532	0,0025	55,81
30	1,1692	0,0064	110,75
60	1,6882	0,0278	225,78
150	2,0967	0,1570	570,38
300	2,4212	0,4090	1164
∞	2,8274	1,1951	139,43

Tabela 6.1: Rezultati simulacij – 2. pristop.



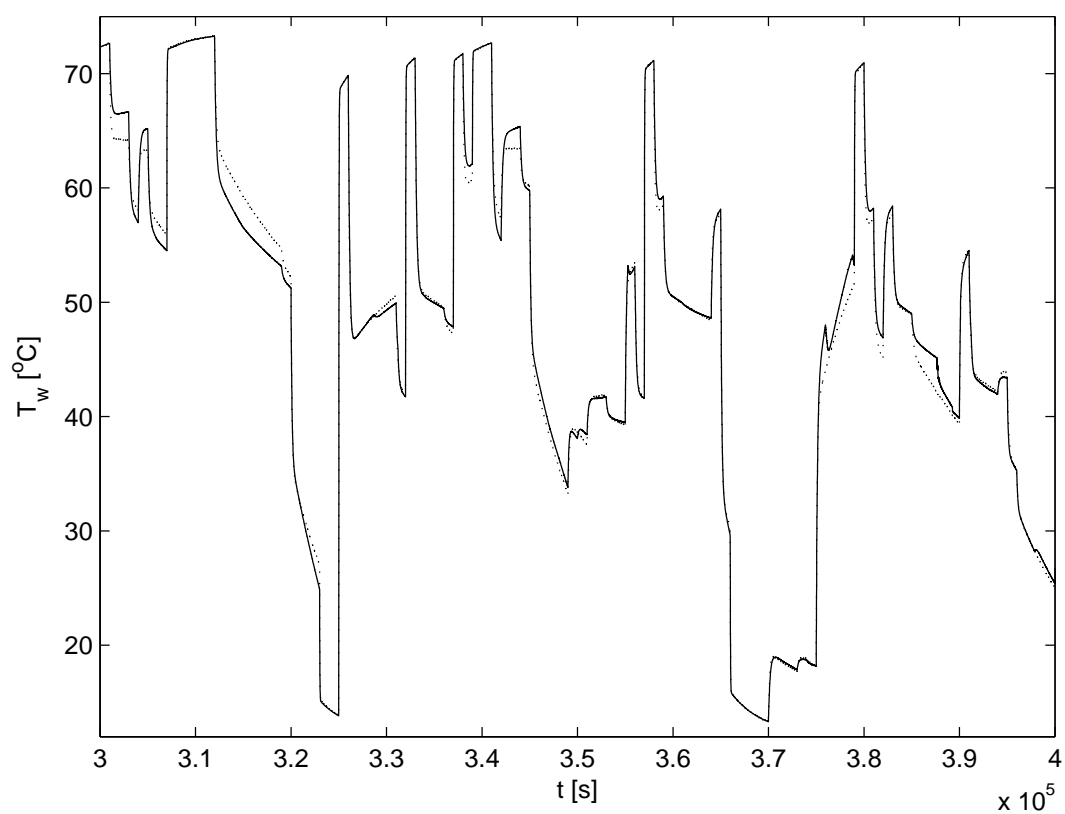
Slika 6.11: Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila predikcijskih korakov h , ki jih računamo v posameznem eksperimentu – 2. pristop. Prekinjena črta označuje vrednost $\bar{J}_{T_w,2}$ pri $h = \infty$.

dru šaržnega reaktorja T za i -ti pristop;

- $t_{sim,i}$ predstavlja čas,³ ki je bil potreben za simulacijo h -koračne predikcije za i -ti pristop. Pri ∞ -koračni predikciji je bil uporabljen preprostejši simulacijski program, zato rezultati $t_{sim,i}$ pri $h = \infty$ in $h \leq \infty$ niso primerljivi.

Sliki 6.11 in 6.13 prikazujeta odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila predikcijskih korakov h , ki jih računamo v posameznem eksperimentu.

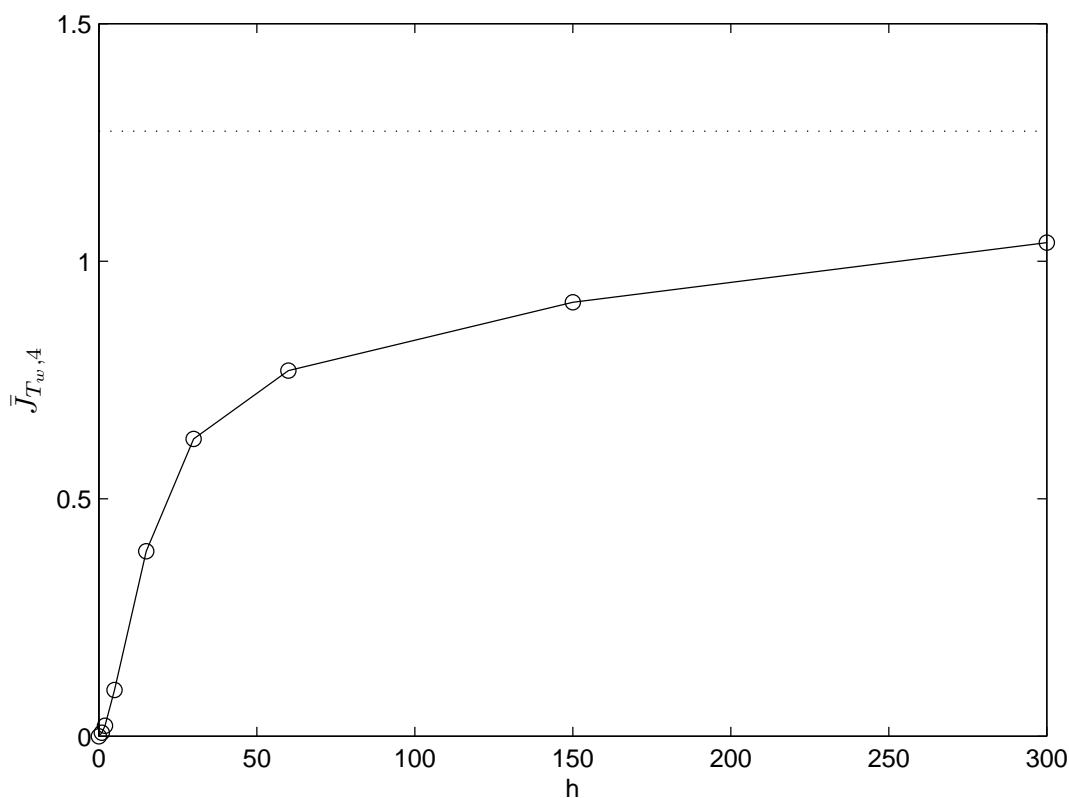
³Eksperimenti so bili izvedeni v programskem paketu *Matlab*, verzija 7.0.1.24704, na računalniku s procesorjem *Intel Pentium M*, 1,6 GHz, in z 1,5 GB delovnega spomina.



Slika 6.12: Odziv modela na vhodne signale za vrednotenje (polna črta) in dejanski odziv (prekinjena črta) – 4. pristop.

h	$\bar{J}_{T_w,4}$	$\bar{J}_{T,4}$	$t_{sim,4}$ [s]
1	0,0075	0,0020	31,40
2	0,0220	0,0020	61,62
5	0,0972	0,0020	152,66
15	0,3892	0,0023	463,52
30	0,6256	0,0047	902,94
60	0,7698	0,0171	1806
150	0,9133	0,0884	4524
300	1,0388	0,2313	9053
∞	1,2735	0,6780	115,48

Tabela 6.2: Rezultati simulacij – 4. pristop.



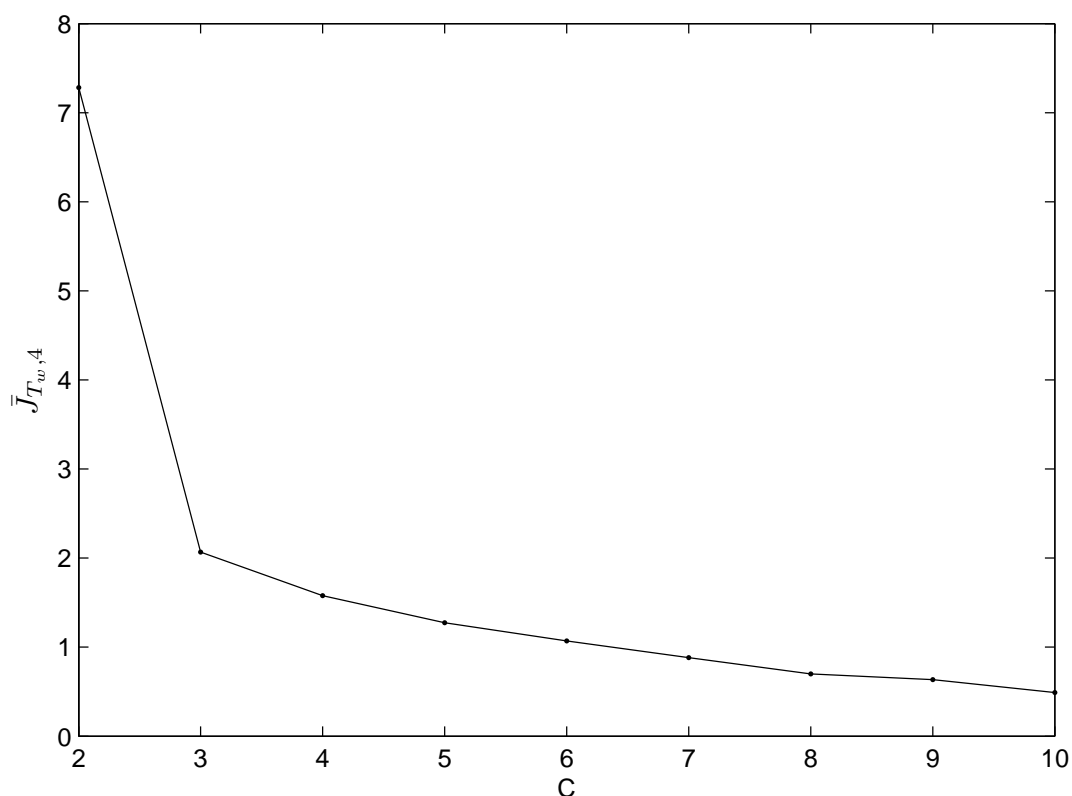
Slika 6.13: Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila predikcijskih korakov h , ki jih računamo v posameznem eksperimentu – 4. pristop. Prekinjena črta označuje vrednost $\bar{J}_{T_w,4}$ pri $h = \infty$.

Slika 6.14 in tabela 6.3 prikazujeta odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w $\bar{J}_{T_w,4}$ od števila rojev C , ki jih določamo pri mehkem rojenju pri 4. pristopu.

6.5.3 Komentar rezultatov

Iz rezultatov, ki smo jih predstavili v prejšnjem razdelku, lahko zaključimo, da je šaržni reaktor mogoče ustrezno predstaviti z odsekoma afnim modelom.

Pristopi, ki smo jih opisali v razdelkih 6.4.1, 6.4.2, 6.4.3 in 6.4.4, se razlikujejo po identifikacijski metodi. Pri vseh pristopih je vhodni prostor odsekoma afinega



Slika 6.14: Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila rojev, ki jih določamo pri mehkem rojenju pri 4. pristopu.

modela v prvem koraku razdeljen glede na način delovanja, tj. glede na položaj zapornih ventilov, v naslednjem koraku pa poskušamo nelinearno dinamiko opisati z odsekoma afinimi modeli, ki veljajo vsak v svojem podprostoru.

Podprostore pri 1. in 2. pristopu ločimo glede na temperaturo v plašču reaktorja T_w , pri čemer meje podprostorov določamo hevristično, pri 3. in 4. pristopu pa pri določanju mej podprostorov uporabimo metodo mehkega rojenja.

Parametre lokalnih afinih modelov, ki pripadajo posameznemu podprostoru, ocenimo pri 1. in 3. pristopu s klasično metodo najmanjše vsote kvadratov odstopanj, pri 2. in 4. pristopu pa uporabimo metodo, ki se naslanja na teorijo mehkih sistemov in je primerna tudi za ocenjevanje parametrov hibridnih mehkih modelov.

C	$\bar{J}_{T_w,4}$
2	7,2825
3	2,0667
4	1,5788
5	1,2735
6	1,0699
7	0,8828
8	0,6985
9	0,6364
10	0,4900

Tabela 6.3: Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila rojev, ki jih določamo pri mehkem rojenju pri 4. pristopu.

Ugotovimo lahko, da se v praktičnih primerih, kot je npr. šaržni reaktor, s podatki za identifikacijo, ki jih imamo na voljo, pojavi problem porazdelitve podatkov za identifikacijo v posameznem podprostoru: podatki z roba podprostora vplivajo na oceno enako kot podatki s sredine podprostora. Ker podatki v splošnem niso enakomerno porazdeljeni po posameznem podprostoru, lahko za take podprostore dobimo neustrezne ocene parametrov. Poleg tega lahko problem predstavlja tudi število identifikacijskih podatkov, ki pripadajo posameznemu podprostoru: če posameznemu podprostoru pripada premalo identifikacijskih podatkov, bo ocena podmodela za ta podprostor slaba ali pa je celo sploh ne bomo mogli določiti.

Zaradi opisane težave sta bila rezultata pri 1. in 3. pristopu nestabilna odsekoma afina modela, ki sta seveda povsem neuporabna za prediktivno vodenje.

Po drugi strani pa smo z metodo mehkega ocenjevanja parametrov pri 2. in 4. pristopu, ki smo jo priredili za identifikacijo odsekoma afinih modelov, dobili modela, ki bolj ali manj ustrezno aproksimirata dinamiko šaržnega reaktorja in

bi ju lahko uporabili pri prediktivnem vodenju.

Sliki 6.11 in 6.13 prikazujeta, kako se s povečevanjem števila predikcijskih korakov h , ki jih računamo v posameznem eksperimentu, večja povprečje kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w . To pomeni, da potrebujemo pri prediktivnem vodenju z velikimi predikcijskimi horizonti, tj. v primerih, ko s predikcijo računamo obnašanje sistema za relativno veliko korakov naprej, tudi bolj natančne modele, ki jih za ta namen uporabljamo. Pri majhnih predikcijskih horizontih natančnost modela ni tako bistvena.

Iz slik je razvidno, da – če se omejimo na odsekoma afine modele z $2 \cdot C = 10$ podprostoru – dobimo s 4. pristopom bolj natančen model kot z 2. pristopom, kar smo tudi pričakovali, saj naj bi z mehkim rojenjem v splošnem dobili bolj ustrezne rezultate kot s hevrističnim določanjem mej med konveksnimi poliedri v vhodnem prostoru odsekoma afinega modela. Po drugi strani pa je določitev in uporaba modela, ki je rezultat 2. pristopa, preprostejša, saj se izognemo rojenju.

Če število rojev C , ki jih določamo pri rojenju pri 4. pristopu za vsak način delovanja, povečujemo, se izboljšuje tudi natančnost modela, kar prikazujeta tudi slika 6.14 in tabela 6.3. Omenimo, da pretirano povečevanje števila rojev ni smiselno, saj bi na ta način dobili preveč kompleksen odsekoma afini model, poleg tega pa bi neenakomerna porazdelitev podatkov za identifikacijo in numerične težave pri ocenjevanju parametrov povzročili, da bi se natančnost modela začela poslabševati, na koncu pa bi lahko dobili celo nestabilen model.

Odločitev, katerega od pristopov je bolj ustrezno uporabiti, je torej v končni fazi načrtovalska odločitev. Odvisna je od sistema, ki ga želimo modelirati, in od zahtev algoritma za prediktivno vodenje.

7. Primer postopka modeliranja in identifikacije hibridnega mehkega modela

Predstavili bomo postopek modeliranja in identifikacije hibridnega mehkega modela, in sicer na primeru šaržnega reaktorja, ki je opisan v 3. poglavju.

7.1 Signali za identifikacijo

Za identifikacijo šaržnega reaktorja potrebujemo primerne vhodno-izhodne signale, iz katerih lahko ocenimo dinamične lastnosti sistema. Signale za identifikacijo tvorimo s psevdonaključnim vhodnim signalom in posnetimi odzivi sistema na ta vhodni signal.

Za podrobnejši opis glej razdelek 6.1.

7.2 Razdelitev sistema

Model šaržnega reaktorja izpeljemo po korakih.

- Najprej razstavimo multivariabilni sistem na dva preprostejša podsistema z več vhodi in enim izhodom.
- Z upoštevanjem vpliva obeh izhodov na oba podsistema določimo strukturo podmodela za vsak podsistem posebej – glej razdelek 5.2.
- Vsak podsistem identificiramo z metodo, opisano v razdelku 5.3.

Vsekakor bi bilo model možno konstruirati tudi brez predhodne razdelitve sistema na posamezne podsisteme. Ker pa parametre modela identificiramo z inverzijo matrik (glej razdelek 5.3), bi lahko pri tem naleteli na numerične težave zaradi slabše pogojenosti matrik, ki nastopajo pri identifikaciji. Poleg tega bi morali v tem primeru hkrati ocenjevati večje število parametrov – tj. matriko parametrov Θ ustrezno večje dimenzije – to pa bi se nujno pokazalo v slabših rezultatih identifikacije.

Zaradi tega je seveda smiselno uporabiti predlagano razdelitev sistema in na ta način obravnavati poenostavljeno strukturo hibridnega mehkega modela. Tako se izognemo nekaterim morebitnim numeričnim težavam pri ocenjevanju parametrov modela.

Za podrobnejšo razlago pristopov k posplošitvi hibridnega mehkega modela in smiselnosti poenostavitev v zvezi z multivariabilnimi sistemi in sistemi višjega reda glej razdelek 5.2.4.

Glede na razdelitev toplotnih tokov, ki nastopajo v šaržnem reaktorju (glej razdelek 3.2), lahko sistem razdelimo na dva podsistema:

- temperatura v jedru šaržnega reaktorja;
- temperatura v plašču šaržnega reaktorja.

Na ta način izkoristimo predhodno poznavanje strukture sistema: v procesu modeliranja torej identificiramo *sivo škatlo*, kar predstavlja neke vrste srednjo pot med identifikacijo *črne škatle* in popolnoma teoretičnim modeliranjem.

7.3 Temperatura v jedru šaržnega reaktorja

Postopek modeliranja in identifikacije je opisan v razdelku 6.3. Model za temperaturo v jedru šaržnega reaktorja podaja enačba (7.1).

$$\hat{T}(k+1) = \boldsymbol{\theta}^T [T_w(k) \ T(k)]^T \quad (7.1)$$

Parametre modela $\boldsymbol{\theta}$ prikazuje enačba (7.2).

$$\boldsymbol{\theta} = [0.0033 \ 0.9967]^T \quad (7.2)$$

7.4 Temperatura v plašču šaržnega reaktorja

Temperatura v plašču reaktorja T_w je odvisna od pretoka toplote med plaščem reaktorja in jedrom ter med plaščem reaktorja in okolico, upoštevati pa moramo tudi pretok toplote zaradi dotoka sveže vhodne vode in iztoka vode iz plašča.

Predpostavimo lahko model z več vhodi in enim izhodom, kot kaže enačba (7.3). Regresor je sestavljen iz temperature v plašču $T_w(k)$ in jedru $T(k)$ šaržnega reaktorja ter vhodnih signalov, in sicer mešalnega ventila $k_M(k)$ ter zapornih ventilov za vročo $k_H(k)$ in hladno vodo $k_C(k)$, v trenutnem časovnem koraku k .

$$\hat{T}_w(k+1) = F(T_w(k), T(k), k_M(k), k_C(k), k_H(k)) \quad (7.3)$$

Podmodel temperature v plašču šaržnega reaktorja T_w bomo zapisali v obliki hibridnega mehkega modela saj dinamika izkazuje tako hibridne kot nelinearne lastnosti. Splošni postopek modeliranja in identifikacije hibridnih mehkih modelov je opisan v razdelkih 5.2 in 5.3.

7.4.1 Hibridni mehki model – 1. pristop

Diskretna dinamika

Očitno je, da lahko v podmodelu, ki opisuje temperaturo v plašču šaržnega reaktorja T_w , ločimo dva načina delovanja, tj. vpeljemo dve diskretni stanji ($s = 2$).

- Prvi način delovanja ($q = 1$) je, ko je sveža vhodna voda vroča, tj. $T_{in} = T_H$ oz. $k_C(k) = 0$ in $k_H(k) = 1$.
- Drugi način delovanja ($q = 2$) je, ko je sveža vhodna voda hladna, tj. $T_{in} = T_C$ oz. $k_C(k) = 1$ in $k_H(k) = 0$.

Opisana razdelitev modela z več vhodi in enim izhodom, ki je zapisan v enačbi (7.3), določa diskretni del hibridnega mehkega modela, ki ga predstavlja enačba (7.4).

$$q(k) = q(k_H(k), k_C(k)) = \begin{cases} 1 & ; \quad k_C(k) = 0 \wedge k_H(k) = 1 \\ 2 & ; \quad k_C(k) = 1 \wedge k_H(k) = 0 \end{cases} \quad (7.4)$$

Nelinearna dinamika

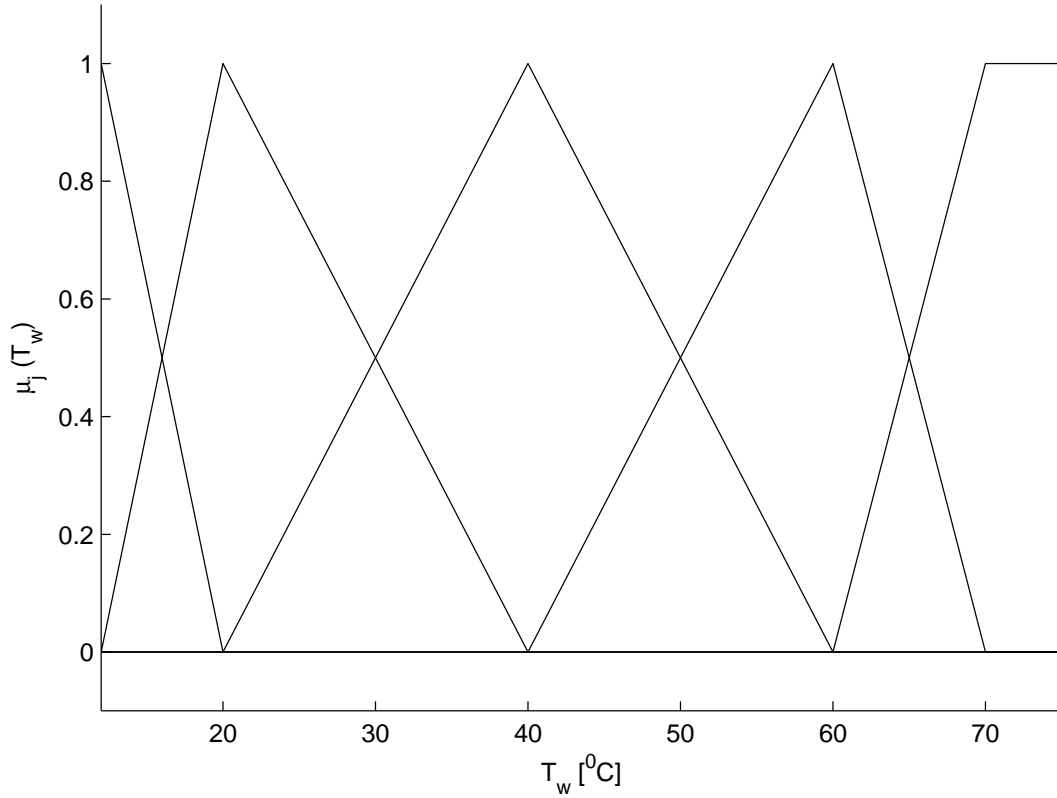
V naslednjem koraku moramo določiti primerne pripadnostne funkcije, da bomo lahko v hibridnem mehkem modelu ustrezno zajeli nelinearno dinamiko sistema. Mehčanje izvedemo glede na temperaturo v plašču reaktorja $T_w(k)$. Izbrali smo $K = 5$ normiranih trikotnih pripadnostnih funkcij z maksimumi pri 12 °C, 20 °C, 40 °C, 60 °C in 70 °C, tako da smo zajeli celotno območje delovanja.

Pripadnostne funkcije so prikazane na sliki 7.1.

Oblika pripadnostnih funkcij zagotavlja, da so normirane stopnje pripadnosti $\beta_j(T_w)$ enake pripadnostim $\mu_j(T_w)$ v celotnem območju delovanja za vsako pravilo \mathbf{R}^{jd} posebej.

Pravila

Pravila hibridnega mehkega sistema podaja enačba (7.5). Predpostavimo, da lahko lokalni model, ki pripada posameznemu pravilu \mathbf{R}^{jd} , zapišemo v afini obliki – glej tudi enačbo (5.9) in razdelek 5.2.2.



Slika 7.1: Pripadnostne funkcije.

\mathbf{R}^{jd} :

če $q(k)$ je Q_d in $T_w(k)$ je A_1^j

$$\text{potem } T_w(k+1) = a_{1jd}T_w(k) + a_{2jd}T(k) + b_{1jd}k_M(k) + r_{jd} \quad (7.5)$$

za $j = 1, \dots, 5$ in $d = 1, 2$

Izhod hibridnega mehkega modela, ki pripada pravilu \mathbf{R}^{jd} , lahko zapišemo, kot kaže enačba (7.6).

$$\mathbf{R}^{jd} : \hat{T}_w^{jd}(k+1) = \Theta_w^{jdT}(q) [T_w(k) \ T(k) \ k_M(k) \ 1]^T \quad (7.6)$$

Diskretni del hibridnega mehkega modela določa enačba (7.7).

$$\Theta_w(q) = \Theta_w(q(k)) = \begin{cases} \Theta_{w1} & \text{če } q(k) = 1 \\ \Theta_{w2} & \text{če } q(k) = 2 \end{cases} \quad (7.7)$$

Parametri modela

Postopek za ocenjevanje hibridnih mehkih modelov je natančneje opisan v razdelku 5.3. Po določitvi parametrov modela za vsako pravilo \mathbf{R}^{jd} posebej lahko zapišemo matriki s parametri modela, ki jih podajata enačbi (7.8) in (7.9).

$$\Theta_{w1} = \begin{bmatrix} 0,9453 & 0,9431 & 0,9429 & 0,9396 & 0,7910 \\ 0,0376 & 0,0458 & 0,0395 & 0,0339 & 0,0225 \\ 19,6748 & 16,7605 & 10,5969 & 3,9536 & 1,6856 \\ 0,3021 & 0,2160 & 0,5273 & 1,2701 & 12,0404 \end{bmatrix} \quad (7.8)$$

$$\Theta_{w2} = \begin{bmatrix} 0,9803 & 0,9740 & 0,9322 & 0,9076 & 0,8945 \\ 0,0025 & 0,0153 & 0,0466 & 0,0466 & 0,0111 \\ -0,0704 & -0,6956 & -7,8013 & -12,2555 & -18,7457 \\ 0,2707 & 0,2033 & 0,5650 & 1,9179 & 5,6129 \end{bmatrix} \quad (7.9)$$

Izhod modela v kompaktni obliki

Če povzamemo, lahko izhod hibridnega mehkega modela, ki obravnava temperaturo v plašču šaržnega reaktorja T_w , zapišemo v kompaktni obliki, kot kaže enačba (7.10).

$$\hat{T}_w(k+1) = \beta(k) \Theta_w^T(q) [T_w(k) \ T(k) \ k_M(k) \ 1]^T \quad (7.10)$$

7.4.2 Hibridni mehki model – 2. pristop

Diskretna dinamika

Podobno kot pri 1. pristopu (glej razdelek 7.4.1) lahko tudi tu v podmodelu, ki opisuje temperaturo v plašču šaržnega reaktorja T_w , ločimo dva načina delovanja, tj. vpeljemo dve diskretni stanji ($s = 2$).

- Prvi način delovanja ($q = 1$) je, ko je sveža vhodna voda vroča, tj. $T_{in} = T_H$ oz. $k_C(k) = 0$ in $k_H(k) = 1$.
- Drugi način delovanja ($q = 2$) je, ko je sveža vhodna voda hladna, tj. $T_{in} = T_C$ oz. $k_C(k) = 1$ in $k_H(k) = 0$.

Opisana razdelitev modela z več vhodi in enim izhodom, ki je zapisan v enačbi (7.3), določa diskretni del hibridnega mehkega modela, ki ga predstavlja enačba (7.4) – glej razdelek 7.4.1.

Nelinearna dinamika

V naslednjem koraku moramo določiti primerne pripadnostne funkcije, da bomo lahko v hibridnem mehkem modelu ustrezno zajeli nelinearno dinamiko sistema.

V tem primeru si bomo pri mehčanju vhodnega prostora hibridnega mehkega modela pomagali z metodo *mehkega rojenja c-središč* (angl. *fuzzy c-means clustering*) – za podrobnejši opis glej tudi razdelek 5.3.

Z rojenjem, ki ga izvajamo nad prostorom vhodnih in izhodnih spremenljivk modela (7.3), želimo podatke za identifikacijo razdeliti na več *mehkih rojev*, tako da vsak podatek za identifikacijo pripada posameznemu mehkeemu roju z določeno stopnjo pripadnosti, ki je odvisna od razdalj do središč posameznih mehkih rojev. Vsak roj želimo v končni fazi predstaviti z afinim podmodelom.

Pred rojenjem podatke za identifikacijo normiramo, tj. skaliramo njihove

vrednosti tako, da so njihove preslikave v intervalu $[0, 1]$. Prostor, nad katerim izvajamo rojenje lahko torej zapišemo v enačbi (7.11).

$$\mathcal{D}_{[\check{T}_w(\tau), \check{T}(\tau), \check{k}_M(\tau)]^T} \times \mathcal{D}_{\check{T}_w(\tau+1)} \equiv [0, 1]^4 \quad (7.11)$$

Označba $\check{}$ nad spremenljivko pomeni, da se zapis nanaša na normirani prostor.

V našem primeru smo pri mehkem rojenju uporabili vrednost parametra $m = 1, 25$.

Z rojenjem razdelimo podatke za identifikacijo glede na vhodne spremenljivke podmodela $T_w(\tau)$, $T(\tau)$ in $k_M(\tau)$ ter izhodno spremenljivko $T_w(\tau + 1)$ na mehke roje. Indeks τ tu označuje časovni korak posameznega podatka za identifikacijo.

Z algoritmom za rojenje določimo tudi središča mehkih rojev v normiranem prostoru (7.11) \check{c}_j , kjer je $j \in \{1, 2, \dots, C\}$, in vsaki točki iz množice podatkov za identifikacijo pripišemo ustrezne vrednosti pripadnostnih funkcij. Za vsak način delovanja sistema določimo po C rojev.

Središča rojev, določena v normiranem vhodnem prostoru modela, so za primer $C = 5$ podana v enačbah (7.12) in (7.13). Enačba (7.12) podaja rezultate za prvi način delovanja ($k_C = 0$ in $k_H = 1$), enačba (7.13) pa rezultate za drugi način delovanja ($k_C = 1$ in $k_H = 0$).

$$\begin{aligned} \check{c}_{1,1} &= [0, 9495 \ 0, 5507 \ 1, 0000 \ 0, 9511] \\ \check{c}_{2,1} &= [0, 4880 \ 0, 4765 \ 0, 0305 \ 0, 4838] \\ \check{c}_{3,1} &= [0, 6715 \ 0, 6935 \ 0, 0500 \ 0, 6682] \\ \check{c}_{4,1} &= [0, 1148 \ 0, 0980 \ 0, 0053 \ 0, 1080] \\ \check{c}_{5,1} &= [0, 3180 \ 0, 2655 \ 0, 0218 \ 0, 3126] \end{aligned} \quad (7.12)$$

$$\begin{aligned}
\check{c}_{1,2} &= [0, 3718 \ 0, 5257 \ 0, 0328 \ 0, 4323] \\
\check{c}_{2,2} &= [0, 2180 \ 0, 2939 \ 0, 0279 \ 0, 2540] \\
\check{c}_{3,2} &= [0, 0818 \ 0, 1010 \ 0, 0412 \ 0, 0954] \\
\check{c}_{4,2} &= [0, 0147 \ 0, 1273 \ 1, 0000 \ 0, 0168] \\
\check{c}_{5,2} &= [0, 0562 \ 0, 4671 \ 1, 0000 \ 0, 0620]
\end{aligned} \tag{7.13}$$

Projekcije mehkih rojev v vhodni prostor modela

Središča rojev (in ustrezne pripadnostne funkcije), ki so rezultat algoritma za mehko rojenje, so določena v normiranem vhodno-izhodnem prostoru modela, ki ga podaja izraz (7.11). Tako določene pripadnostne funkcije lahko neposredno uporabimo pri identifikaciji odsekoma afinega modela, problem pa se pojavi pri uporabi odsekoma afinega modela za predikcijo obnašanja sistema.

V naslednjem koraku moramo torej mehke roje projicirati v vhodni prostor modela (7.3) $\mathcal{D}_{[T_w, T, k_M]^T}$. V tem primeru postopamo, kot je opisano v razdelku 5.3.

Na ta način lahko torej za vsako rojenje izpeljemo po C funkcij, ki so definirane nad *vhodnim* prostorom hibridnega mehkega modela $\mathcal{D}_{[T_w, T, k_M]^T}$ in ki vsaki točki $[T_w, T, k_M]^T$ v tem prostoru priredijo vrednosti $r_{j,1}([T_w, T, k_M]^T)$ za $q = 1$ oz. $r_{j,2}([T_w, T, k_M]^T)$ za $q = 2$, in sicer za $j \in \{1, 2, \dots, C\}$, kot podajata izraza (7.14) in (7.15).

$$\begin{aligned}
r_{j,1} &: \mathcal{D}_{[T_w, T, k_M]^T} \rightarrow [r_{j,1, \min}, \infty), \\
r_{j,1} &: [T_w, T, k_M]^T \mapsto r_{j,1}([T_w, T, k_M]^T),
\end{aligned} \tag{7.14}$$

kjer je $j \in \{1, 2, \dots, C\}$.

$$\begin{aligned}
r_{j,2} &: \mathcal{D}_{[T_w, T, k_M]^T} \rightarrow [r_{j,2, \min}, \infty), \\
r_{j,2} &: [T_w, T, k_M]^T \mapsto r_{j,2}([T_w, T, k_M]^T), \tag{7.15} \\
&\text{kjer je } j \in \{1, 2, \dots, C\}.
\end{aligned}$$

Če poznamo funkcije $r_{j,1}([T_w, T, k_M]^T)$ oz. $r_{j,2}([T_w, T, k_M]^T)$ za vsak $j \in \{1, 2, \dots, C\}$, lahko izpeljemo pripadnostne funkcije, ki ustrezajo rezultatom mehkega rojenja in so definirane nad *vhodnim* prostorom hibridnega mehkega modela $\mathcal{D}_{[T_w, T, k_M]^T}$ in ki vsaki točki v prostoru $[T_w, T, k_M]^T$ priredijo vrednosti $\mu_{j,1}([T_w, T, k_M]^T)$ za $q = 1$ oz. $\mu_{j,2}([T_w, T, k_M]^T)$ za $q = 2$, in sicer za $j \in \{1, 2, \dots, C\}$, kot podajata izraza (7.16) in (7.17).

$$\mu_{j,1} = \frac{1}{\sum_{k=1}^C \left(\frac{r_{j,1}}{r_{k,1}} \right)^{\frac{2}{m-1}}}, \tag{7.16}$$

kjer je $j \in \{1, 2, \dots, C\}$.

$$\mu_{j,2} = \frac{1}{\sum_{k=1}^C \left(\frac{r_{j,2}}{r_{k,2}} \right)^{\frac{2}{m-1}}}, \tag{7.17}$$

kjer je $j \in \{1, 2, \dots, C\}$.

Pripadnostne funkcije, ki so zapisane v izrazih (7.16) in (7.17), lahko neposredno implementiramo pri uporabi hibridnega mehkega modela za predikcijo obnašanja sistema – glej enačbo (5.10) – tj. pri določanju normiranih stopenj pripadnosti za posamezen roj $\beta_{j,1}(k)$ oz. $\beta_{j,2}(k)$, kjer je $j \in \{1, 2, \dots, C\}$ – glej enačbo (5.12).

Definicija pripadnostnih funkcij, ki je osnovana na razdalji od središč posameznih rojev, zagotavlja, da so normirane stopnje pripadnosti $\beta_{j,1}([T_w(k), T(k), k_M(k)]^T)$ oz. $\beta_{j,2}([T_w(k), T(k), k_M(k)]^T)$ enake vrednostim pripadnostnih funkcij $\mu_{j,1}([T_w(k), T(k), k_M(k)]^T)$ oz. $\mu_{j,2}([T_w(k), T(k), k_M(k)]^T)$ v celotnem območju delovanja za vsako pravilo \mathbf{R}^{jd} posebej.

Pravila

Pravila hibridnega mehkega sistema podaja enačba (7.18). Predpostavimo, da lahko lokalni model, ki pripada posameznemu pravilu \mathbf{R}^{jd} , zapišemo v afini obliki – glej tudi enačbo (5.9) in razdelek 5.2.2.

\mathbf{R}^{jd} :

če $q(k)$ je Q_d in $[T_w(k), T(k), k_M(k)]^T$ je $A_{1,d}^j$

$$\text{potem } T_w(k+1) = a_{1jd}T_w(k) + a_{2jd}T(k) + b_{1jd}k_M(k) + r_{jd} \quad (7.18)$$

za $j = 1, \dots, 5$ in $d = 1, 2$

Izhod hibridnega mehkega modela, ki pripada pravilu \mathbf{R}^{jd} , lahko zapišemo enako kot pri 1. pristopu – kot kaže enačba (7.6). Tudi diskretni del hibridnega mehkega modela je določen enako kot pri 1. pristopu, in sicer ga podaja enačba (7.7) – glej razdelek 7.4.1.

Parametri modela

Postopek za ocenjevanje hibridnih mehkih modelov je natančneje opisan v razdelku 5.3. Po določitvi parametrov modela za vsako pravilo \mathbf{R}^{jd} posebej lahko zapišemo matriki s parametri modela, ki jih podajata enačbi (7.19) in (7.20).

$$\Theta_{w1} = \begin{bmatrix} 0,6312 & 0,9257 & 0,9361 & 0,9404 & 0,9277 \\ 0,0400 & 0,0513 & 0,0384 & 0,0490 & 0,0515 \\ 1,7115 & 12,5240 & 6,9758 & 19,3915 & 16,8357 \\ 22,4874 & 0,7210 & 1,0316 & 0,1955 & 0,5233 \end{bmatrix} \quad (7.19)$$

$$\Theta_{w2} = \begin{bmatrix} 0,9059 & 0,9337 & 0,9462 & 0,6279 & 0,6312 \\ 0,0630 & 0,0468 & 0,0359 & 0,0404 & 0,0400 \\ -10,8609 & -4,5854 & -1,5090 & -15,3616 & -14,9129 \\ 0,8869 & 0,4228 & 0,3163 & 19,3905 & 18,9084 \end{bmatrix} \quad (7.20)$$

Izhod modela v kompaktni obliki

Če povzamemo, lahko izhod hibridnega mehkega modela, ki obravnava temperaturo v plašču šaržnega reaktorja T_w , zapišemo v kompaktni obliki, kot kaže enačba (7.21).

$$\hat{T}_w(k+1) = \beta(k, q) \Theta_w^T(q) [T_w(k) \ T(k) \ k_M(k) \ 1]^T \quad (7.21)$$

7.5 Vrednotenje

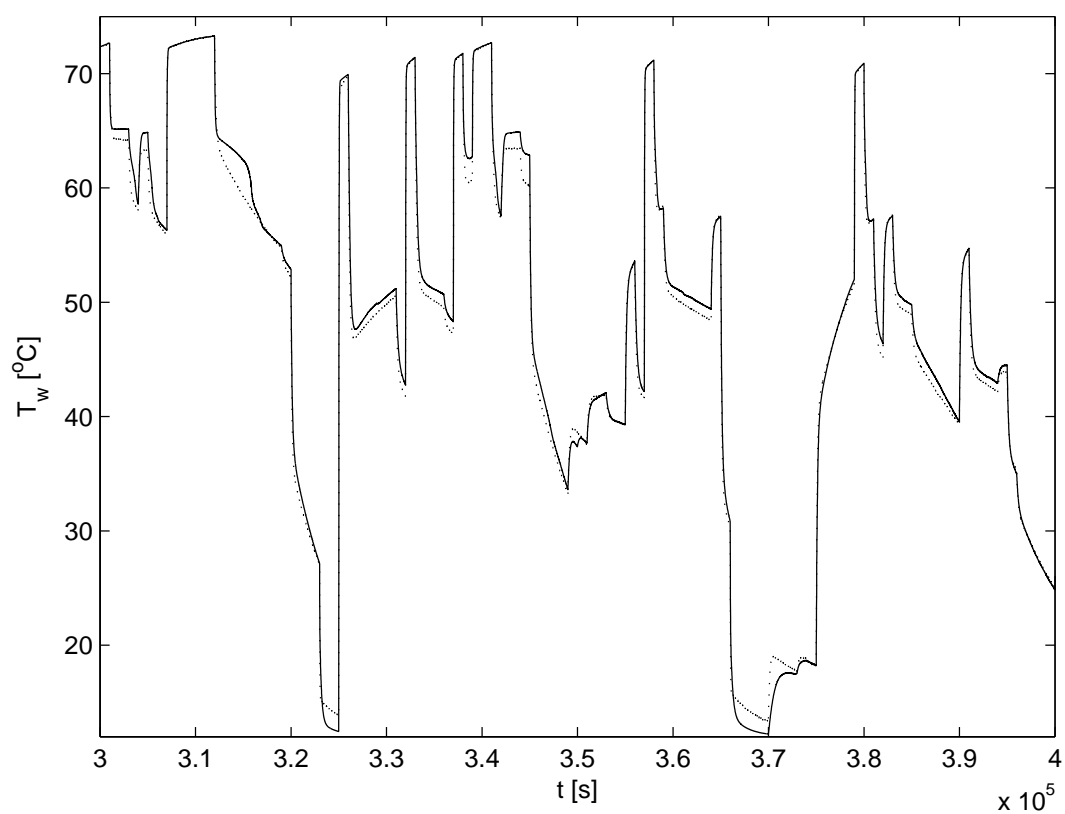
Hibridna mehka modela, ki sta rezultat pristopov, opisanih v razdelkih 7.4.1 in 7.4.2, smo ovrednotili tako, da smo jih primerjali medsebojno in s simulacijskim modelom šaržnega reaktorja.

7.5.1 Podatki za vrednotenje

Podatke za vrednotenje smo tvorili, kot je opisano v razdelku 6.5.1. V podatke za vrednotenje smo zajeli odziv šaržnega reaktorja na psevdonaključni vhodni signal. Posneli smo oba merljiva izhoda, tj. temperaturo v jedru T in temperaturo v plašču šaržnega reaktorja T_w .

7.5.2 Eksperimenti

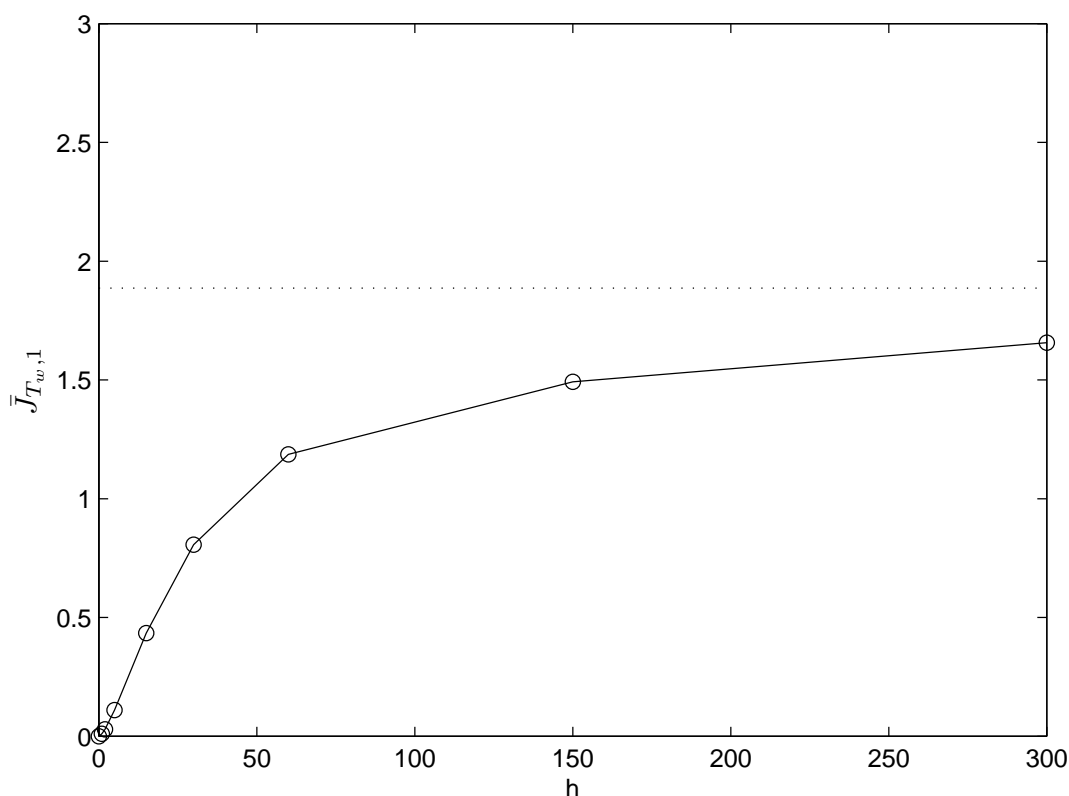
Primerjali smo odzive hibridnih mehkih modelov modelov, ki sta rezultat pristopov, opisanih v razdelkih 7.4.1 in 7.4.2, na vhodne signale za vrednotenje, ki



Slika 7.2: Odziv modela na vhodne signale za vrednotenje (polna črta) in dejanski odziv (prekinjena črta) – 1. pristop.

h	$\bar{J}_{T_w,1}$	$\bar{J}_{T,1}$	$t_{sim,1}$ [s]
1	0,0105	0,0020	10,59
2	0,0293	0,0020	20,94
5	0,1101	0,0020	51,77
15	0,4343	0,0024	154,56
30	0,8067	0,0052	314,04
60	1,1867	0,0214	624,20
150	1,4917	0,1232	1566
300	1,6568	0,3188	3230
∞	1,8861	0,8288	82,18

Tabela 7.1: Rezultati simulacij – 1. pristop.

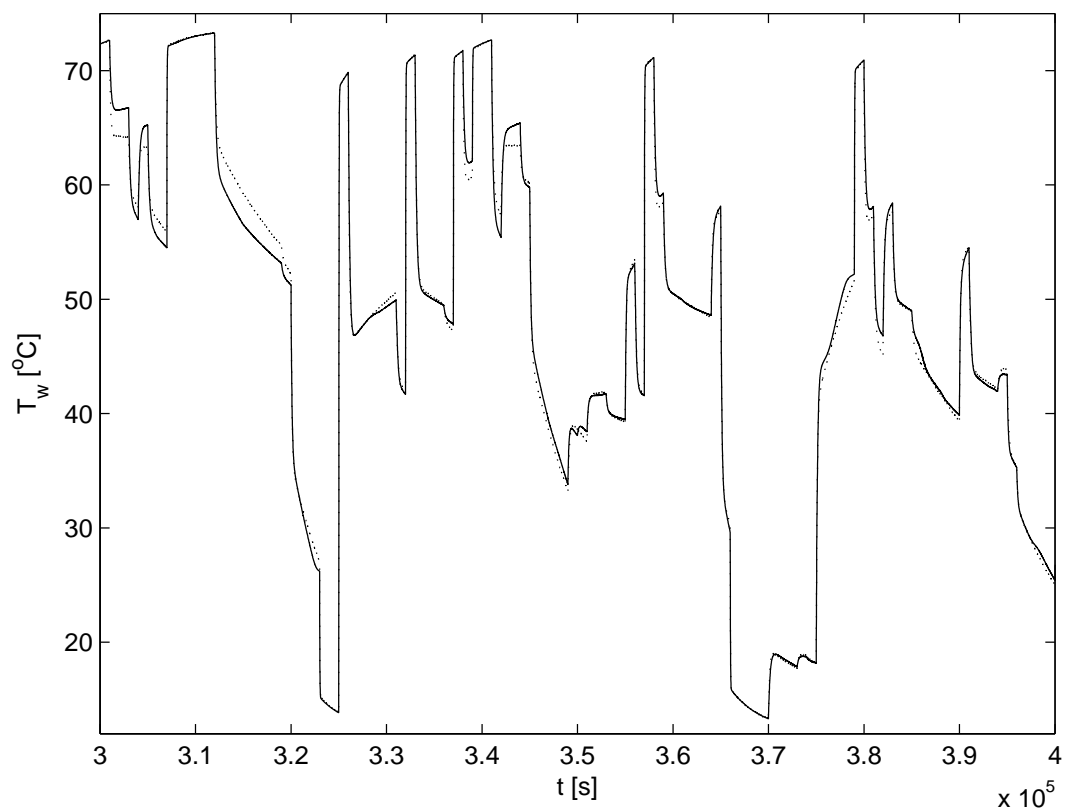


Slika 7.3: Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila predikcijskih korakov h , ki jih računamo v posameznem eksperimentu – 1. pristop. Prekinjena črta označuje vrednost $\bar{J}_{T_w,1}$ pri $h = \infty$.

so opisani v razdelku 6.5.1. V eksperimentih smo uporabili omenjene modele za napovedovanje poteka trajektorij temperature v plašču šaržnega reaktorja T_w , in sicer za h korakov naprej, pa tudi za celoten čas poteka signalov za vrednotenje.

Omenimo, da smo pri eksperimentih uporabljali celoten model šaržnega reaktorja, tj. podmodel za temperaturo v jedru T in podmodel za temperaturo v plašču T_w , vendar je pri vseh pristopih podmodel za temperaturo v jedru T , ki je opisan v razdelku 6.3, isti.

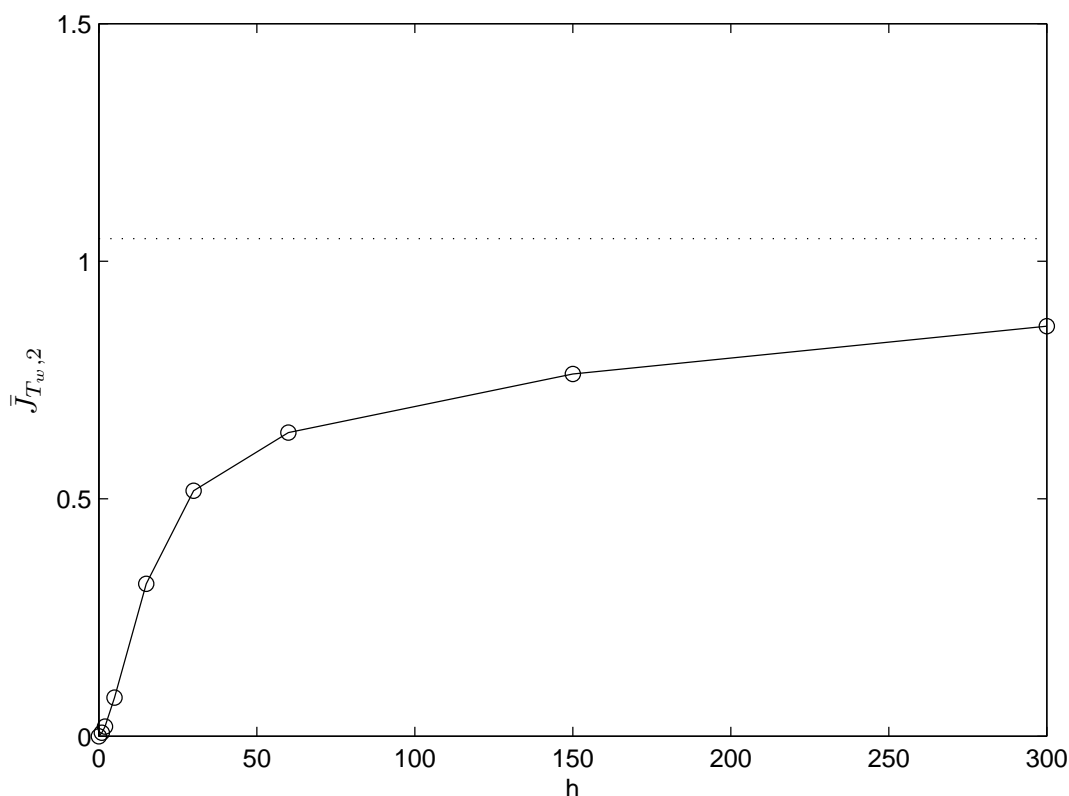
Sliki 7.2 oz. 7.4 prikazujeta izsek trajektorije temperature v plašču šaržnega reaktorja T_w in odziv modela dobljenega s 1. oz. z 2. pristopom, ki sta opisana v razdelkih 7.4.1 in 7.4.2, na vhodne signale za vrednotenje. Za primerjavo je na



Slika 7.4: Odziv modela na vhodne signale za vrednotenje (polna črta) in dejanski odziv (prekinjena črta) – 2. pristop.

h	$\bar{J}_{T_w,2}$	$\bar{J}_{T,2}$	$t_{sim,2}$ [s]
1	0,0075	0,0020	8,33
2	0,0201	0,0020	15,42
5	0,0814	0,0020	38,30
15	0,3208	0,0022	108,76
30	0,5167	0,0042	214,56
60	0,6393	0,0146	430,79
150	0,7623	0,0757	1085
300	0,8629	0,1968	2196
∞	1,0474	0,5574	90,79

Tabela 7.2: Rezultati simulacij – 2. pristop.



Slika 7.5: Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila predikcijskih korakov h , ki jih računamo v posameznem eksperimentu – 2. pristop. Prekinjena črta označuje vrednost $\bar{J}_{T_w,2}$ pri $h = \infty$.

slikah s prekinjeno črto označen dejanski odziv na vhodne signale za vrednotenje.

Tabela 7.1 prikazuje rezultate simulacij z modelom, ki smo ga dobili s 1. pristopom, opisanim v razdelku 7.4.1; tabela 7.2 pa rezultate simulacij z modelom, ki smo ga dobili z 2. pristopom, opisanim v razdelku 7.4.2.

V tabelah 7.1 in 7.2 so v vsaki vrstici podani rezultati simulacij, in sicer glede na število predikcijskih korakov h , ki smo jih računali v posameznem eksperimentu. Označba $h = \infty$ pomeni, da gre v tem primeru za celotno simulacijo, tj. odziv modela na vhodne signale za vrednotenje, brez sprotnih popravkov stanj, kakršne v vsakem koraku naredimo pri h -koračni predikciji za $h < \infty$.

Stolpci v tabelah 7.1 in 7.2 predstavljajo parametre za vrednotenje modelov, kot so navedeni spodaj:

- $\bar{J}_{T_w,i}$ predstavlja povprečje kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w za i -ti pristop;
- $\bar{J}_{T,i}$ predstavlja povprečje kvadratov odstopanj vrednosti temperature v jedru šaržnega reaktorja T za i -ti pristop;
- $t_{sim,i}$ predstavlja čas,¹ ki je bil potreben za simulacijo h -koračne predikcije za i -ti pristop. Pri ∞ -koračni predikciji je bil uporabljen preprostejši simulacijski program, zato rezultati $t_{sim,i}$ pri $h = \infty$ in $h \leq \infty$ niso primerljivi.

Sliki 7.3 in 7.5 prikazujeta odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila predikcijskih korakov h , ki jih računamo v posameznem eksperimentu.

Slika 7.6 in tabela 7.3 prikazujeta odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w $\bar{J}_{T_w,2}$ od števila rojev C , ki jih določamo pri mehkem rojenju pri 2. pristopu.

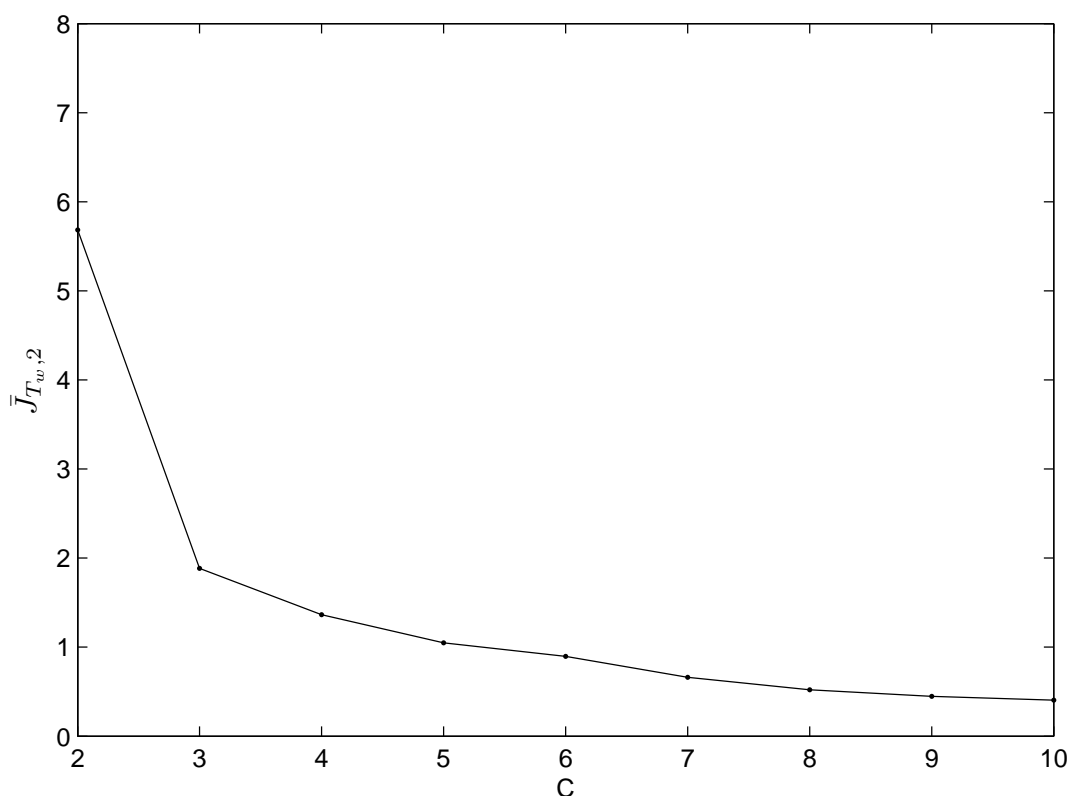
7.5.3 Komentar rezultatov

Iz rezultatov, ki smo jih predstavili v prejšnjem razdelku 7.5, lahko zaključimo, da je šaržni reaktor mogoče ustrezno predstaviti s hibridnim mehkim modelom.

Pristopi, ki smo jih opisali v razdelkih 7.4.1 in 7.4.2 se razlikujejo po identifikacijski metodi.

Podprostore pri 1. pristopu ločimo glede na temperaturo v plašču reaktorja T_w , pri čemer pripadnostne funkcije določimo hevristično, pri 2. pristopu pa pri določanju pripadnostnih funkcij uporabimo metodo mehkega rojenja.

¹Eksperimenti so bili izvedeni v programskem paketu *Matlab*, verzija 7.0.1.24704, na računalniku s procesorjem *Intel Pentium M*, 1,6 GHz, in z 1,5 GB delovnega spomina.



Slika 7.6: Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila rojev, ki jih določamo pri mehkem rojenju pri 2. pristopu.

Parametre lokalnih afinih modelov, ki pripadajo posameznemu podprostoru, ocenimo z metodo, ki se naslanja na teorijo mehkih sistemov. Na ta način smo dobili modela, ki bolj ali manj ustrezno aproksimirata dinamiko šaržnega reaktorja in bi ju lahko uporabili pri prediktivnem vodenju.

Sliki 7.3 in 7.5 prikazujeta, kako se s povečevanjem števila predikcijskih korakov h , ki jih računamo v posameznem eksperimentu, večja povprečje kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w . To pomeni, da potrebujemo pri prediktivnem vodenju z velikimi predikcijskimi horizonti, tj. v primerih, ko s predikcijo računamo obnašanje sistema za veliko korakov naprej, tudi bolj natančne modele, ki jih za ta namen uporabljamo. Pri majhnih predikcijskih horizontih natančnost modela ni tako bistvena.

C	$\bar{J}_{T_w,2}$
2	5,6844
3	1,8832
4	1,3633
5	1,0474
6	0,8970
7	0,6615
8	0,5214
9	0,4487
10	0,4048

Tabela 7.3: Odvisnost povprečja kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w od števila rojev, ki jih določamo pri mehkem rojenju pri 2. pristopu.

Iz slik je razvidno, da – če se pri številu pripadnostnih funkcij omejimo na $C = 5$ (za vsak način delovanja) – dobimo z 2. pristopom bolj natančen model kot s 1. pristopom, kar smo tudi pričakovali, saj naj bi z mehkim rojenjem v splošnem dobili bolj ustrezne rezultate kot s hevrističnim določanjem pripadnostnih funkcij. Po drugi strani pa je določitev in uporaba modela, ki je rezultat 1. pristopa, preprostejša, saj se izognemo rojenju.

Če število rojev C , ki jih določamo pri rojenju v 2. pristopu za vsak način delovanja, povečujemo, se izboljšuje tudi natančnost modela, kar prikazujeta slika 7.6 in tabela 7.3. Omenimo naj, da povečevanje števila rojev v nedogled ni smiselno, saj bi na ta način dobili preveč kompleksen hibridni mehki model, poleg tega pa bi neenakomerna porazdelitev podatkov za identifikacijo in numerične težave pri ocenjevanju parametrov povzročili, da bi se natančnost modela začela poslabševati, na koncu pa bi lahko dobili celo nestabilen model.

Primerjava med hibridnim mehkim modelom in ustreznim odsekoma afinim modelom (glej 6. poglavje) pokaže, da – ob primerljivi zahtevnosti pristopa za

modeliranje in identifikacijo ter kompleksnosti na primerljiv način dobljenega končnega modela – dosežemo večjo natančnost s hibridnim mehkim modelom.

Če primerjamo hibridni mehki model, ki smo ga dobili s 1. pristopom, in odsekoma afini model, ki smo ga dobili z 2. pristopom, lahko ugotovimo, da znaša povprečje kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w pri simulaciji za hibridni mehki model $\bar{J}_{T_w,1} |_{h=\infty} = 1,8861$, za odsekoma afini model pa $\bar{J}_{T_w,1} |_{h=\infty} = 2,8274$.

Tudi pri primerjavi hibridnega mehkega modela, ki smo ga dobili z 2. pristopom, in odsekoma afinega modela, ki smo ga dobili s 4. pristopom, so rezultati v prid hibridnega mehkega modela. Povprečje kvadratov odstopanj vrednosti temperature v plašču šaržnega reaktorja T_w znaša pri simulaciji za hibridni mehki model $\bar{J}_{T_w,2} |_{h=\infty} = 1,0474$, za odsekoma afini model pa $\bar{J}_{T_w,4} |_{h=\infty} = 1,2735$.

Primerjava slik 6.14 in 7.6 oz. tabel 6.3 in 7.3 pokaže, da odsekoma afini model preseže natančnost hibridnega mehkega modela, ki je rezultat mehkega rojenja s po $C = 5$ rojev, šele pri $C = 7$ rojih. Za primerljivo natančnost moramo torej uporabiti kompleksnejši odsekoma afini model glede na hibridni mehki model, kar pa lahko bistveno poveča računsko zahtevnost algoritma za prediktivno vodenje.

Odločitev, katerega od pristopov je ustrezneje uporabiti, je torej v končni fazi načrtovalska odločitev. Odvisna je od sistema, ki ga želimo modelirati, in od zahtev algoritma za prediktivno vodenje.

Del III

Metode prediktivnega vodenja hibridnih sistemov

8. Uvod v metode prediktivnega vodenja hibridnih sistemov

Ena od naprednih metodologij vodenja sistemov, ki se je zelo uveljavila v industrijski praksi, je prediktivno vodenje [45, 117]. Večinoma se prediktivno vodenje uporablja v petrokemični industriji, vedno večjo veljavo pa pridobiva tudi v ostalih industrijskih sektorjih. Glavni razlogi za uspeh prediktivnega vodenja v praksi so [93]:

- preprosta obravnava multivariabilnih sistemov;
- možnost upoštevanja omejitev aktuatorjev;
- dovoljuje delovanje bližje omejitvam (v primerjavi s klasičnimi pristopi), kar vodi k bolj ekonomičnemu delovanju;
- v mnogih primerih so dovoljeni relativno dolgi časi med spremembami regulirnih signalov, kar omogoči dovolj časa za njihovo sprotno (angl. *online*) izračunavanje.

Prediktivno vodenje je način vodenja dinamičnih sistemov, pri katerem uporabljamo model procesa za napovedovanje obnašanja sistema pri določenih vhodih. Optimalne vhode, ki jih uporabimo za dejansko vodenje sistemov, navadno določimo s pomočjo različnih optimizacijskih postopkov.

Prediktivni regulatorji so uporabni za vodenje širokega spektra različnih tipov sistemov, zato se predvsem v primerih, ko so vodeni sistemi sorazmerno

kompleksni [64, 59, 121, 84, 106, 60, 71, 94, 61] (nelinearni sistemi, multivariabilni sistemi, sistemi z omejitvami, hibridni sistemi idr.), pokaže njihova prednost pred drugimi regulatorji.

Pomembna prednost prediktivnega vodenja je, da lahko razmeroma preprosto upoštevamo omejitve v procesu, kot so omejitve aktuatorjev, nevarna ali prepovedana območja delovanja ipd. V procesni industriji je pogosto pogoj za optimalno, tj. najbolj donosno, vodenje, da proces deluje blizu omejitev. Te so večinoma povezane s ceno, porabo energije ipd. Prav tako je lahko predpisan minimalni kakovostni standard končnega produkta, ki predstavlja omejitev v procesu. V takem primeru poskušamo s prediktivnim vodenjem proces voditi tako, da minimiziramo proizvodne stroške, ob tem pa ne prekršimo omejitve, ki jo predstavlja minimalni kakovostni standard.

Metode prediktivnega vodenja so se v industriji uveljavile [118], ker se odlikujejo po svoji enostavnosti, uporabnosti, relativno dobrem delovanju in razumljivem algoritmu. Z razvojnega vidika bi lahko pristope k prediktivnemu vodenju razdelili na dve glavni veji: poleg prediktivnega vodenja, pri katerem je bistvena optimizacija z upoštevanjem omejitev, se pojavlja tudi prediktivno vodenje, pri katerem sta bistvena preprosto umerjanje parametrov vodenja in jasnost pristopa. V drugem primeru sama optimizacija in omejitve v algoritmu niso bistvenega pomena, ampak je poudarjena predvsem preprostost in majhna računska zahtevnost algoritmov, ki so primerni predvsem za sisteme z enim vhodom in enim izhodom (angl. *single-input, single-output* – *SISO*).

Moderni računalniki omogočajo vedno večje hitrosti računanja, zato meja med zgoraj omenjenima vejama prediktivnega vodenja vse bolj izginja [20]. Prediktivno vodenje danes ni več omejeno samo na počasne procese, kjer je časa za računanje med dvema korakoma oz. spremembama regulirnega signala veliko, ampak se uporaba vedno bolj širi tudi na področje hitrih procesov. Kljub temu pa je pri zahtevnih problemih vodenja, kjer nastopajo kompleksni dinamični procesi, računska zahtevnost še vedno glavna težava. V literaturi se zato pojavljajo pri-

stopi z optimizacijskimi algoritmi [143, 123, 122, 82, 51, 140, 128], ki so računsko sicer manj zahtevni, a zagotavljajo suboptimalne rešitve.

V nadaljevanju poglavja bomo opisali osnovno idejo in formulacijo problema optimalnega vodenja ter glavno omejitev, na katero naletimo pri iskanju splošne rešitve. Nato bomo predstavili prediktivno vodenje kot uveljavljeno poenostavitev splošnega problema optimalnega vodenja, opisali osnovni mehanizem delovanja in poudarili nekatere pomembne vidike, ki jih moramo upoštevati pri načrtovanju.

V naslednjih poglavjih tretjega dela doktorske disertacije pa se bomo osredotočili na nekaj pristopov k prediktivnemu vodenju hibridnih sistemov.

8.1 Optimalno vodenje

Ideja o reševanju problema vodenja kot optimizacijskega problema sega že med leti 1955 in 1970, ko se je razvijala teorija optimalnega vodenja. Glavno gonilo razvoja so bili optimizacijski problemi, ki so se pojavljali v letalski in vesoljski industriji, predvsem pri vodenju letal in raket ter pri vzletanju, letanju in pristajanju vesoljskih plovil [93, 103].

V nekaterih pogledih je teorija optimalnega vodenja rešila izredno širok nabor problemov. Predpostavimo, da v sistemu, ki ga želimo voditi, predstavlja u vektor vhodov in x vektor stanj. Dinamiko sistema, ki je v splošnem nelinearna, naj podaja enačba (8.1).

$$\frac{dx}{dt} = f(x, u) \quad (8.1)$$

Denimo, da je cilj vodenja minimizacija kriterijske funkcije, ki jo podaja enačba (8.2).

$$J(x, u, t) = \int_0^{t_{kon}} l(x(\tau), u(\tau), \tau) d\tau + F_{kon}(x(t_{kon})) \quad (8.2)$$

Naj bo vhod omejen na množico $u(t) \in \mathcal{U}$. V enačbi (8.2) je $l(x(\tau), u(\tau), \tau)$ nenegativna funkcija, F_{kon} pa je funkcija, ki ovrednoti stanje sistema v trenutku t_{kon} , ko se vodenje konča.

Opisana formulacija predstavlja zelo splošen zapis problema optimalnega vodenja. S primerno določitvijo funkcij f , F in l lahko zaobjamemo praktično vsak smiseln problem vodenja.

8.1.1 Hamilton-Jacobi-Bellmanova enačba

V osnovi je potek reševanja tovrstnih splošnih optimizacijskih problemov znan: problem se prevede na reševanje *Hamilton-Jacobi-Bellmanove enačbe* [42], ki jo podaja enačba (8.3).

$$\frac{\partial}{\partial t} J^0(x, t) = \min_{u \in \mathcal{U}} H \left(x, u, \frac{\partial}{\partial x} J^0(x, t) \right) \quad (8.3)$$

Funkcijo $H(x, u, \lambda)$ iz enačbe (8.3) podaja enačba (8.4).

$$H(x, u, \lambda) = l(x, u) + \lambda f(x, u) \quad (8.4)$$

Robni pogoj podaja enačba (8.5).

$$J^0(x, t_{kon}) = F_{rob}(x) \quad (8.5)$$

Ko rešimo enačbo (8.3), izberemo vhodni signal tako, da minimiziramo funkcijo H , kot kaže enačba (8.6).

$$u^0(x, t) = \arg \min_{u \in \mathcal{U}} H \left(x, u, \frac{\partial}{\partial x} J^0(x, t) \right) \quad (8.6)$$

Poudariti je treba, da zgornje enačbe predstavljajo zaprtozančni regulirni zakon, saj je v trenutku t odvisen od stanja sistema $x(t)$, ki pa je posledica dogajanja v sistemu od začetka do trenutka t .

Žal je Hamilton-Jacobi-Bellmanovo enačbo v splošnem praktično nemogoče rešiti, zato se v praksi omejimo na ožje razrede optimizacijskih problemov, ki strukturno poenostavijo osnovni problem optimalnega vodenja.

8.1.2 Optimalno vodenje linearnih sistemov

En način poenostavitve je obravnava samo linearnih sistemov. V tem primeru lahko enačbo (8.1) zapišemo, kot kaže enačba (8.7)

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (8.7)$$

Predpostavimo, da sta l in F kvadratni funkciji. Cilj vodenja je torej minimizacija kriterijske funkcije, ki jo podaja enačba (8.8).

$$\begin{aligned} J(x, u, t) = & \\ & \int_0^{t_{kon}} (x(\tau)^T Q(\tau) x(\tau) + u(\tau)^T R(\tau) u(\tau)) d\tau + \\ & + x(t_{kon})^T S x(t_{kon}) \end{aligned} \quad (8.8)$$

V enačbi (8.8) so $Q(\tau)$, $R(\tau)$ in S kvadratne, simetrične in pozitivno definitne matrike. V opisanem primeru se Hamilton-Jacobi-Bellmanova enačba v (8.3) pretvori v navadno diferencialno *Riccatijevo enačbo*. Rešitev Riccatijeve enačbe je linearen časovno spremenljiv regulirni zakon, ki ga podaja enačba (8.9).

$$u(t) = F(t)x(t) \quad (8.9)$$

V enačbi (8.9) določa matrika $F(t)$ regulator, ki je odvisen od rešitev Riccatijeve enačbe. Večina razvoja optimalnega vodenja je temeljila na opisani poenostavitvi Hamilton-Jacobi-Bellmanove enačbe, kar je prineslo razvoj zelo močne in izpopolnjene teorije, ki je bila pogosto uporabljena tudi v praksi, posebno v dualni obliki v zvezi s *Kalmanovim filtrom* [93].

Slabost pristopa je, da je omejen na linearne sisteme v obliki, kot jo podaja enačba (8.7), in kriterijske funkcije v obliki, kot jo podaja enačba (8.8), v formulaciji in rešitvi pa ne upošteva omejitev in ostalih morebitnih nelinearnosti v sistemu.

8.2 Prediktivno vodenje kot poenostavitev reševanja problema optimalnega vodenja

Glavna težava optimalnega vodenja v osnovni obliki je, da rešuje problem optimizacije nad funkcijo, tj. vhodnim signalom u . Da bi našli optimalno vhodno funkcijo med vsemi mogočimi moramo torej uporabiti variacijski račun.

Uveljavljen pristop k poenostavitvi problema reševanja Hamilton-Jacobi-Bellmanove enačbe v okviru optimalnega vodenja je prediktivno vodenje. Osnovna ideja prediktivnega vodenja je, da splošni variacijski račun poenostavimo tako, da omejimo množico mogočih vhodov, in sicer do te mere, da se optimizacija izvaja samo nad končno množico parametrov, namesto nad množico funkcij.

Navadno se poenostavitev izvede tako, da diskretiziramo zvezni čas, tj. čas razdelimo na diskretne intervale, in optimizacijo izvajamo nad končnim predikcijskim horizontom H oz. nad končnim številom časovnih intervalov. Pri tem predpostavimo, da je vhodni signal u v času trajanja enega diskretnega časovnega intervala, tj. časovnega koraka, konstanten.

Na ta način pretvorimo – v splošnem nerešljiv – problem optimizacije vhodne funkcije na bolj obvladljiv problem optimizacije končnega števila vrednosti vhodnega signala, ki bi jih uporabili v posameznem prihodnjem časovnem intervalu do predikcijskega horizonta H .

8.3 Uporaba modelov v diskretnem času

Kot smo omenili v razdelku 2.4, se bomo v disertaciji pri obravnavi prediktivnega vodenja omejili na pristope z modeli, ki so določeni v diskretnem času. V tem primeru postane pomembna ustrezna izbira časa vzorčenja t_S , od katerega je odvisno modeliranje diskretnega modela ali pretvorba zveznega modela sistema v diskretni model, ki ga potem uporabimo pri prediktivnem vodenju.

Po eni strani je z izbiro časa vzorčenja povezana natančnost modela oziroma predikcije obnašanja sistema: izbira predolgega časa vzorčenja namreč onemogoča modeliranje hitrejše dinamike v sistemu. Izbira časa vzorčenja je s tega vidika predvsem modelerski problem, seveda pa ima posledice tudi pri vodenju sistema, saj algoritem za prediktivno vodenje s slabim modelom ne more doseči ustrezne kakovosti vodenja.

Po drugi strani pa izbira časa vzorčenja vpliva na računsko zahtevnost (glej razdelek 8.6): število računskih operacij pri računanju predikcije obnašanja sistema v določenem časovnem intervalu je namreč obratno sorazmerno s časom vzorčenja. Z drugimi besedami – za dosego nekega končnega časa predikcije t_H je potrebnih več predikcijskih korakov, tj. daljši predikcijski horizont H . Podaljšanje predikcijskega horizonta H pa v splošnem povzroči (eksponentno) povečanje računske zahtevnosti algoritma za prediktivno vodenje, kar lahko v skrajnem primeru tudi onemogoči uporabo algoritma v realnem času.

8.4 Delovanje algoritmov za prediktivno vodenje

Prediktivno vodenje je pristop, pri katerem se z uporabo matematičnega modela sistema, ki ga želimo voditi, napoveduje bodoče obnašanje sistema [45]. Pri prediktivnem vodenju v diskretnem času gre torej v osnovi za to, da ob vsakem časovnem koraku k izberemo tisto vrednost vektorja vhodov v sistem, ki je najprimernejša glede na trenutno stanje sistema in želeni potek stanj oziroma izhodov

sistema.

Določanje *primernosti* vektorja vhodov predstavlja optimizacijski problem, ki ga rešujemo v okviru končnega predikcijskega horizonta H , tj. za določeno število korakov naprej. Optimizacijo izvajamo glede na napoved bodočega obnašanja sistema pri določenih vhodnih signalih, ki jo določimo z modelom, ki ga uporabljamo za predikcijo, in ustrezne kriterijske funkcije,¹ pri čemer pa moramo upoštevati tudi vse omejitve za vhode, izhode in stanja sistema.

Rešitev problema vodenja v času $t = k \cdot t_S$ (tj. v k -tem časovnem koraku) predstavlja zaporedje vrednosti vektorja vhodov, ki jo podaja enačba (8.10).

$$U_k^{k+H-1} = (u(k | k), u(k+1 | k), \dots, u(k+H-1 | k)) \quad (8.10)$$

V enačbi (8.10) označuje $u(k+i-1 | k)$ vektor vhodov, ki ga uporabimo za predikcijo obnašanja sistema v $(k+i)$ -tem časovnem koraku,² tj. v i -tem koraku predikcije, ki se izračunava v časovnem koraku k ; $H \in \mathbb{N}$ pa predstavlja predikcijski horizont, tj. število korakov naprej od koraka k , ki jih algoritem za prediktivno vodenje izračuna.

Pri prediktivnem vodenju želimo model sistema, ki ga vodimo, (v vsakem časovnem koraku k) iz začetnega stanja $x(k)$ – preko vseh vmesnih korakov $(k+1 | k, \dots, k+H-1 | k)$ in pripadajočih stanj v okviru največjega predikcijskega horizonta H , ki jih podaja enačba (8.11) – pripeljati v končno stanje $x(k+H)$, in sicer s tako izbiro zaporedja vrednosti vhodnih vektorjev (8.10), da minimiziramo kriterijsko funkcijo.

$$X_{k+1}^{k+H} = (x(k+1 | k), x(k+2 | k), \dots, x(k+H | k)) \quad (8.11)$$

Od celotnega optimiziranega zaporedja vrednosti vektorja vhodov (8.10) se za vodenje sistema uporabi samo prvi vhodni vektor iz zaporedja $u(k | k)$, in sicer

¹Glej razdelek 8.5.

² $i \in \{1, 2, \dots, H\}$.

samo v času trajanja časovnega koraka k .

V naslednjih časovnih korakih $k+j$, kjer je $j \in \{1, 2, \dots\}$, moramo vsakič rešiti nov optimizacijski problem, ki je določen z novim trenutnim stanjem sistema $x(k+j)$ in želenim potekom stanj v okviru premaknjenega predikcijskega horizonta X_{k+j+1}^{k+j+H} .

Omenimo, da prediktivno vodenje v osnovni obliki ne rešuje zaprtozančnega problema, ampak niz odprtozančnih problemov. Zaprtozančno delovanje dosežemo tako, da začetnim pogojem v nizu odprtozančnih problemov priredimo v vsakem časovnem koraku k zadnje izmerjene signale, ki predstavljajo stanje sistema.

8.5 Kriterijska funkcija

Kot smo povedali v prejšnjih razdelkih, predstavlja temelj delovanja algoritmov za prediktivno vodenje minimizacija kriterijske funkcije, zato je od njene določitve neposredno odvisna kakovost vodenja pa tudi računski zahtevnost.

Splošen zapis kriterijske funkcije za prediktivno vodenje lahko izpeljemo z modifikacijo splošnega zapisa kriterijske funkcije za optimalno vodenje v enačbi (8.2). Zapis podaja enačba (8.12).

$$J = \sum_{h=k}^{k+H} l(X_{k+1}^{k+H}, U_k^{k+H-1}, k, h) \quad (8.12)$$

za $h = 1, 2, \dots, H$

Funkcija $l(X_{k+1}^{k+H}, U_k^{k+H-1}, k, h)$ v enačbi (8.12) ni nujno enaka funkciji $l(x(\tau), u(\tau), \tau)$ v enačbi (8.2), opravlja pa podobno vlogo pri vrednotenju kakovosti vodenja. Vlogo funkcije $F_{kon}(x(t_{kon}))$ iz enačbe (8.2) lahko prevzame funkcija $l(x(\tau), u(\tau), \tau)$ v enačbi (8.2), če je to za H korakov naprej smiselno.

Različne metode prediktivnega vodenja zahtevajo različne formulacije krite-

rijske funkcije – primerjaj npr. (8.2) in (8.8). Pri nekaterih so poleg tega pogoj za delovanje tudi posebne lastnosti kriterijske funkcije, kot so monotonost ipd. Posebne oblike kriterijskih funkcij pa lahko izhajajo tudi iz samega sistema oz. modela, ki ga uporablja algoritem za prediktivno vodenje. Vse te omejitve in zahteve seveda posredno in neposredno vplivajo na kakovost vodenja in na računsko zahtevnost reševanja problema prediktivnega vodenja.

8.6 Računska zahtevnost optimizacijskega problema

Če hočemo sistem voditi v realnem času, se mora algoritem za prediktivno vodenje v vsakem koraku izvesti do konca. To pa lahko pri uporabi predstavlja problem, zato moramo poznati računsko zahtevnost problema vodenja in dejavnike, ki nanjo vplivajo.

Računska zahtevnost je odvisna od:

- sistema, ki ga želimo voditi, oz. modela sistema, ki ga uporablja algoritem za prediktivno vodenje – torej od števila stanj, vhodov, izhodov, omejitev, (ne)zveznosti, (ne)linearnosti itd.;
- predikcijskega horizonta H ;
- algoritma za prediktivno vodenje, tj. načina reševanja optimizacijskega problema;
- kriterijske funkcije.

Poudariti je treba, da računsko zahtevnost predstavlja enega bistvenih vidikov pri uporabi prediktivnega vodenja v praksi: pri nekaterih kombinacijah zgoraj navedenih dejavnikov se namreč lahko tako poveča, da prediktivnega vodenja sploh ne moremo implementirati.

8.7 Prediktivno vodenje hibridnih sistemov

Načrtovanje vodenja hibridnih sistemov je zaradi prepletanja zvezne in diskretne dinamike v splošnem zelo zahtevna naloga. V literaturi lahko zato najdemo različne pristope k reševanju tega problema [126, 127, 107, 30, 31, 26, 17, 135, 29]. Večina pristopov se omejuje na hibridne sisteme v diskretnem času, saj s tem zaobidejo več kompleksnih matematičnih problemov – glej razdelek 2.4. V praksi so se uveljavile nekatere metode za optimalno oziroma prediktivno vodenje hibridnih sistemov.

Kot smo omenili, je prediktivno vodenje pristop, pri katerem z napovedovanjem obnašanja procesa z uporabo modela določamo ustrezne regulirne signale. Bistveno je torej modeliranje procesa, kar je zaradi kompleksne hibridne in nelinearne dinamike številnih industrijskih procesov pogosto težavna naloga. Za tovrstne sisteme klasične metode modeliranja in identifikacije, ki temeljijo na teoriji linearnih sistemov, niso primerne, zato potrebujemo posebne pristope [125].

Kaže se torej potreba po ustreznih formulacijah modelov, ki naj bodo primerni za prediktivno vodenje, poleg tega pa morajo ustrezno upoštevati kompleksnost dinamike sistemov, ki jih želimo voditi. Modeli morajo torej imeti primerno strukturo, pomembna pa je tudi možnost eksperimentalne identifikacije parametrov, ki jih ne moremo ali jih ni smiselno določiti s teoretičnim modeliranjem. Nekatere od formulacij modelov hibridnih sistemov, ki so primerni za uporabo pri prediktivnem vodenju, smo opisali v prejšnjih poglavjih.

Od formulacije modela hibridnega sistema, ki ga uporabljamo za prediktivno vodenje, je odvisna tudi možnost uporabe različnih pristopov k vodenju. Nekateri algoritmi za vodenje tako zahtevajo točno določeno formulacijo modela hibridnega sistema, da jih sploh lahko uporabimo.

V naslednjih poglavjih tretjega dela doktorske disertacije se bomo torej osredotočili na nekatere pristope k prediktivnemu vodenju hibridnih sistemov.

9. Reševanje mešanih celoštevilskih optimizacijskih problemov

V tem poglavju bomo opisali metodo za prediktivno vodenje hibridnih sistemov na osnovi reševanja mešanih celoštevilskih optimizacijskih problemov, primerno za sisteme, ki jih lahko zapišemo v obliki odsekoma afinega ali ekvivalentnega modela – glej 4. poglavje.

9.1 Odsekoma afini in ekvivalentni modeli

Kot smo omenili že v 4. poglavju, je za namene prediktivnega vodenja hibridnih sistemov najbolj razširjena uporaba zapisov sistema, ki ga želimo voditi, v obliki *odsekoma afinih* in njim ekvivalentnih modelov (predvsem *mešanih logično-dinamičnih modelov*) [109]. Zapis je podprt z zaokroženo teorijo modeliranja in je bil za namene prediktivnega vodenja že večkrat uporabljen v praksi [22, 102, 66, 39, 65, 43, 27, 44, 119]. Poleg tega postopke modeliranja oz. zapisa sistema olajšujejo tudi nekateri programski paketi oz. jeziki, kot sta npr. *Hysdel* [136] in *Hybrid Identification Toolbox* [62]. Formulacije odsekoma afinih in njim ekvivalentnih modelov omogočajo opis dinamike hibridnih linearnih oz. afinih sistemov.

Problem prediktivnega vodenja sistemov, ki so formulirani kot odsekoma afini in njim ekvivalentni modeli, lahko pretvorimo v *mešani celoštevilski optimizacijski problem* (angl. *mixed integer optimization problem*), ki ga rešujemo s pomočjo programov za reševanje optimizacijskih problemov (angl. *solver*).

9.2 Sprotno in predhodno reševanje optimizacijskega problema

Pri reševanju optimizacijskega problema ločujemo dva osnovna pristopa.

1. Problem lahko rešujemo v vsakem časovnem koraku sproti (angl. *online*) [90, 98].
2. Lahko zastavimo in predhodno (angl. *offline*) rešimo *multiparametrični mešani celoštevilski optimizacijski problem* (angl. *multiparametric mixed integer optimization problem*) [25, 38] ter na ta način izkoristimo dejstvo, da je rešitev strukturno enolična. Izkaže se, da jo lahko izrazimo v obliki eksplicitno določenega odsekoma afinega regulirnega zakona [34].

9.3 Delovanje algoritma za prediktivno vodenje

Za uporabo metode za prediktivno vodenje hibridnih sistemov na osnovi reševanja mešanih celoštevilskih optimizacijskih problemov obstajajo različna programska orodja [33], kot sta npr. *Multi-Parametric Toolbox* [87] in *Hybrid Toolbox* [19].

V tem razdelku bomo opisali pristop z multiparametričnim programiranjem, kakršnega uporablja programsko orodje *Multi-Parametric Toolbox* [87].

9.3.1 Politopi

Politopi, ali splošneje poliedri, imajo osrednjo vlogo pri multiparametričnem programiranju.

Polieder $Q \subseteq \mathbb{R}^n$ je konveksna množica, ki je presek končnega števila zaprtih polprostorov, kot podaja enačba (9.1).

$$Q = \{x \in \mathbb{R}^n; Q^x x \leq Q^c\} \quad (9.1)$$

Politop $\mathcal{P} \subset \mathbb{R}^n$ je omejeni polieder, kot kaže enačba (9.2).

$$\mathcal{P} = \{x \in \mathbb{R}^n; P^x x \leq P^c\} \quad (9.2)$$

Vsak politop predstavlja konveksno, omejeno in zaprto množico. Politop je *polnodimenzionalen*, če obstaja tak $x \in \mathbb{R}^n$, da velja $P^x x < P^c$. Če je $\|(P^x)_i\| = 1$, pri čemer $(P^x)_i$ označuje i -to vrstico matrike P^x , pravimo, da je politop \mathcal{P} *normiran*. Politop lahko opišemo tudi z oglišči, kot kaže enačba (9.3).

$$\mathcal{P} = \{x \in \mathbb{R}^n; x = \sum_{i=1}^{v_P} \alpha_i V_P^{(i)}, 0 \leq \alpha_i \leq 1, \sum_{i=1}^{v_P} \alpha_i = 1\} \quad (9.3)$$

V enačbi (9.3) označuje $V_P^{(i)}$ i -to oglišče, v_P pa skupno število oglišč politopa \mathcal{P} .

Opis s polprostori v enačbi (9.2) označimo kot \mathcal{H} -zapis, opis z oglišči v enačbi (9.3) pa kot \mathcal{V} -zapis.

Linearna neenačba $a'x \leq b$ je *veljavna* za polieder \mathcal{P} , če velja za vsak $x \in \mathcal{P}$. Podmnožici poliedra rečemo *ploskev*, če jo za veljavno neenačbo $a'x \leq b$ lahko predstavimo kot presek, ki ga podaja izraz (9.4).

$$\mathcal{F} = \mathcal{P} \cap \{x \in \mathbb{R}^n; a'x = b\} \quad (9.4)$$

Pravimo, da je politop $\mathcal{P} \in \mathbb{R}^n$, $\mathcal{P} = \{x \in \mathbb{R}^n; P^x x \leq P^c\}$ v *minimalni obliki*, če bi ga odstranitev katere koli vrstice v $P^x x \leq P^c$ spremenila. Normirani in polnodimenzionalni politop ima enolično določeno minimalno obliko.

Osnovne operacije nad politopi

Razlika množic, ki določata politopa \mathcal{P} in \mathcal{Q} , je unija politopov $\mathcal{R} = \bigcup_i \mathcal{R}_i$, kot kaže enačba (9.5).

$$\mathcal{R} = \mathcal{P} \setminus \mathcal{Q} := \{x \in \mathbb{R}^n; x \in \mathcal{P}, x \notin \mathcal{Q}\} \quad (9.5)$$

Pontryaginova razlika politopov \mathcal{P} in \mathcal{W} je politop, ki ga določa enačba (9.6).

$$\mathcal{P} \ominus \mathcal{W} := \{x \in \mathbb{R}^n; x + w \in \mathcal{P}, \forall w \in \mathcal{W}\} \quad (9.6)$$

Vsota Minkovskega politopov \mathcal{P} in \mathcal{W} je politop, ki ga določa enačba (9.7).

$$\mathcal{P} \oplus \mathcal{W} := \{x + w \in \mathbb{R}^n; x \in \mathcal{P}, w \in \mathcal{W}\} \quad (9.7)$$

Konveksna ovojnica unije politopov $\mathcal{P}_i \subset \mathbb{R}^n$, $i = 1, \dots, p$ je politop, ki ga določa enačba (9.8).

$$\square\left(\bigcup_{i=1}^p \mathcal{P}_i\right) := \left\{x \in \mathbb{R}^n; x = \sum_{i=1}^p \alpha_i x_i, x_i \in \mathcal{P}_i, 0 \leq \alpha_i \leq 1, \sum_{i=1}^p \alpha_i = 1\right\} \quad (9.8)$$

Ovojnica dveh politopov v \mathcal{H} -zapisu $\mathcal{P} = \{x \in \mathbb{R}^n; P^x x \leq P^c\}$ in $\mathcal{Q} = \{x \in \mathbb{R}^n; Q^x x \leq Q^c\}$ je politop, ki ga določa enačba (9.9).

$$\square(\mathcal{P}, \mathcal{Q}) := \{x \in \mathbb{R}^n; \bar{P}^x x \leq \bar{P}^c, \bar{Q}^x x \leq \bar{Q}^c\} \quad (9.9)$$

V enačbi (9.9) označuje $\bar{P}^x x \leq \bar{P}^c$ podsistem sistema neenačb $P^x x \leq P^c$, ki ga dobimo z odstranitvijo tistih neenačb, ki niso veljavne za politop \mathcal{Q} . Podobno je določen podsistem $\bar{Q}^x x \leq \bar{Q}^c$ sistema neenačb $Q^x x \leq Q^c$, ki ga dobimo z odstranitvijo tistih neenačb, ki niso veljavne za politop \mathcal{P} .

9.3.2 Mešani celoštevilski optimizacijski problem

Linearni sistemi

Najprej bomo obravnavali osnovno metodo za prediktivno vodenje linearnih sistemov na osnovi reševanja (multiparametričnih) optimizacijskih problemov.

Linearni časovno nespremenljivi sistem v diskretnem času podaja enačba (9.10).

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) \\y(k) &= Cx(k) + Du(k)\end{aligned}\tag{9.10}$$

V enačbi (9.10) označuje $x(k)$ stanje sistema, $u(k)$ vhod v sistem, $y(k)$ pa izhod sistema. S k je označen trenutni časovni korak sistema. Velja $A \in \mathbb{R}^{n \times n}$ in $B \in \mathbb{R}^{n \times m}$.

Predikcijo stanja sistema v časovnem koraku k čez h predikcijskih korakov označimo z $x(k+h | k)$. Krajše označimo $x_h = x(0+h | 0)$. Izračunani vhod v času h (glede na $x(0)$) kratko označimo z u_h .

Predpostavimo, da so stanja in vhodi sistema v izrazu (9.10) omejeni, kot je zapisano v izrazu (9.11).

$$\begin{aligned}x &\in \mathbb{X} \subset \mathbb{R}^n \\u &\in \mathbb{U} \subset \mathbb{R}^m\end{aligned}\tag{9.11}$$

V izrazu (9.11) sta z \mathbb{X} in \mathbb{U} označena omejena in zaprta poliedra, ki vsebujeta koordinatno izhodišče.

Problem omejenega optimalnega vodenja v končnem času

Problem omejenega optimalnega vodenja v končnem času podaja izraz (9.12), pri čemer veljajo izrazi (9.13), (9.14), (9.15), (9.16) in (9.17).

$$J_N^*(x(0)) = \min_{u_0, \dots, u_{H-1}} \|Q_f x_H\|_l + \sum_{h=0}^{H-1} \|Ru_h\|_l + \|Qx_h\|_l \tag{9.12}$$

$$x_h \in \mathbb{X}, \forall h \in \{1, \dots, H\} \tag{9.13}$$

$$x_H \in \mathcal{X}_{kon} \quad (9.14)$$

$$u_h \in \mathbb{X}, \forall h \in \{0, \dots, H-1\} \quad (9.15)$$

$$x_0 = x(0), x_{k+1} = Ax_k + Bu_k, \forall h \in \{0, \dots, H-1\} \quad (9.16)$$

$$\begin{cases} \text{rang}(Q) = n, \text{rang}(R) = m; & \text{če velja } l \in \{1, \infty\} \\ Q = Q^T \preceq 0, Q_f = Q_f^T \preceq 0, R = R^T \prec 0; & \text{če velja } l = 2 \end{cases} \quad (9.17)$$

V enačbi (9.14) označuje \mathcal{X}_{kon} omejitev končnega stanja, ki jo določi uporabnik tako, da zagotovi stabilnost zaprtozančnega sistema [99]. Kriterijska funkcija v izrazu (9.12) je lahko linearna ($l \in \{1, \infty\}$) [23] ali kvadratična ($l = 2$) [34]. Matrike Q , R in Q_f pa vsebujejo uteži posameznih signalov v kriterijski funkciji, ki jih določi uporabnik.

Množica H -koračne rešljivosti $\mathcal{X}_f^H \subseteq \mathbb{R}^n$ je množica začetnih stanj $x(0)$, za katere je problem omejenega optimalnega vodenja v končnem času, ki ga podaja izraz (9.12), pri čemer veljajo izrazi (9.13), (9.14), (9.15), (9.16) in (9.17), rešljiv. Množico podaja izraz (9.18).

$$\begin{aligned} \mathcal{X}_f^H &= \{x(0) \in \mathbb{R}^n; \exists [u_0, \dots, u_{H-1}] \in \mathbb{R}^{Hm}, \\ &x_h \in \mathbb{X}, u_{h-1} \in \mathbb{U}, \forall h \in \{1, \dots, H\}\} \end{aligned} \quad (9.18)$$

Za določeno začetno stanje $x(0)$ lahko problem omejenega optimalnega vodenja¹ v končnem času rešimo kot *linearni program* (angl. *linear program* – *LP*) za $l \in \{1, \infty\}$ oz. kot *kvadratični program* (angl. *quadratic program* – *QP*) za $l = 2$.

¹*Mešani celoštevilski linearni program* oz. *mešani celoštevilski kvadratični program* predstavljata posplošitev *linearnega programa* oz. *kvadratičnega programa*, s katero lahko obravnavamo probleme optimalnega vodenja odsekoma afinih in njim ekvivalentnih sistemov.

V tem primeru rešujemo program v vsakem časovnem koraku k sproti. Velika računaska zahtevnost pa lahko, predvsem pri hitrejših sistemih, onemogoči vodenje.

Eksplicitni regulirni zakon

x_h lahko določimo, kot podaja enačba (9.19).

$$x_h = A^h x(0) + \sum_{j=0}^{h-1} A^j B u_{h-1-j} \quad (9.19)$$

Z upoštevanjem enačbe (9.19) lahko ponovno zapišemo problem omejenega optimalnega vodenja v končnem času kot ga podaja izraz (9.20), pri čemer velja izraz (9.21).

$$J_H^*(x(0)) = x(0)^T Y x(0) + \min_{U_H} \{U_H^T H U_H + x(0)^T F U_H\} \quad (9.20)$$

$$G U_H \leq W + E x(0) \quad (9.21)$$

V enačbah (9.20) in (9.21) označuje $U_H = [u_0^T, \dots, u_{H-1}^T]^T \in \mathbb{R}^{Mn}$ stolpni vektor, ki ga določamo z optimizacijo. H , F , Y , G , W in E izpeljemo iz matrik Q , R in Q_f ter izrazov (9.10) in (9.11) [34].

Optimizacijski problem v izrazih (9.20) in (9.21) je odvisen od $x(0)$, zato ga lahko rešimo tudi kot multiparametrični program: če obravnavamo $x(0)$ kot parameter, lahko optimizacijski problem v izrazih (9.20) in (9.21) rešujemo za vse parametre $x(0)$. Na ta način izpeljemo eksplicitni regulirni zakon.

Obravnavamo problem omejenega optimalnega vodenja v končnem času, ki ga podaja izraz (9.12), pri čemer veljajo izrazi (9.13), (9.14), (9.15), (9.16) in (9.17). V tem primeru [34, 36] je množica H -koračne rešljivosti \mathcal{X}_f^H konveksna, vektor, ki ga iščemo z optimizacijo, pa določa zvezna in odsekoma afna funkcija

$U_H^* : \mathcal{X}_f^H \rightarrow \mathbb{R}^{Mn}$, kot je zapisano v izrazu (9.22).

$$U_H^*(x(0)) = F_r x(0) + G_r, \quad (9.22)$$

če velja $x(0) \in \mathcal{P}_r = \{x \in \mathbb{R}^n; H_r x \leq K_r\}$, $r = 1, \dots, R$.

Pri tem velja, da je optimalna kriterijska funkcija $J_H^* : \mathcal{X}_f^H \rightarrow \mathbb{R}$ zvezna, konveksna in odsekoma kvadratična ($l = 2$) ali odsekoma linearna ($l \in \{1, \infty\}$).

Množica H -koračne rešljivosti \mathcal{X}_f^H je torej razdeljena na R politopov: $\mathcal{X}_f^H = \{\mathcal{P}_r\}_{r=1}^R$. V vsakem od politopov \mathcal{P}_r je eksplicitno določen regulirni zakon, kot kaže izraz (9.22).

Odsekoma afini sistemi

Multiparametrično programiranje je možno posplošiti tudi na odsekoma afine sisteme [24, 37, 16], ki jih podajata enačbi (9.23) in (9.24).

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) + f_i \\ y(k) &= C_i x(k) + D_i u(k) + g_i \end{aligned} \quad (9.23)$$

$$\begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \Omega_i \quad (9.24)$$

V enačbi (9.23) označuje $x(k)$ stanje, $u(k)$ vhod in $y(k)$ izhod odsekoma afinega sistema. S k je označen trenutni časovni korak sistema. Velja $A_i \in \mathbb{R}^{n \times n}$ in $B_i \in \mathbb{R}^{n \times m}$.

Predpostavimo, da so stanja in vhodi sistema v izrazu (9.23) v okviru podprostora Ω_i omejeni, kot je zapisano v izrazu (9.25).

$$L_i x(k) + E_i u(k) \leq W_i \quad (9.25)$$

Množica $I \subset \mathbb{H}$, $I = \{1, \dots, d\}$ predstavlja vse mogoče dinamike v odsekoma afinem sistemu, pri čemer d označuje število različnih dinamik.

Enačbi (9.23) in (9.24) lahko krajše zapišemo v izrazu (9.26).

$$x(k+1) = f_{PWA}(x(k), u(k)) \quad (9.26)$$

Funkcija v izrazu (9.26) ni nujno zvezna.

Problem omejenega optimalnega vodenja odsekoma afnega sistema v končnem času lahko zapišemo, kot podaja izraz (9.27), pri čemer veljajo izrazi (9.28), (9.29), (9.30) in (9.31).

$$J_H^*(x(0)) = \min_{u_0, \dots, u_{H-1}} \|Q_f x_H\|_l + \sum_{h=0}^{H-1} \|Ru_h\|_l + \|Qx_h\|_l \quad (9.27)$$

$$L_i x(k) + E_i u(k) \leq W_i, \text{ če je } \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \Omega_i, \forall h \in \{1, \dots, H\} \quad (9.28)$$

$$x_H \in \mathcal{X}_{kon} \quad (9.29)$$

$$x_0 = x(0), x_{k+1} = f_{PWA}(x_k, u_k), \forall h \in \{0, \dots, H-1\} \quad (9.30)$$

$$\begin{cases} \text{rang}(Q) = n, \text{ rang}(R) = m; & \text{če velja } l \in \{1, \infty\} \\ Q = Q^T \preceq 0, Q_f = Q_f^T \preceq 0, R = R^T \prec 0; & \text{če velja } l = 2 \end{cases} \quad (9.31)$$

V enačbi (9.29) označuje \mathcal{X}_{kon} omejitev končnega stanja, ki jo določi uporabnik tako, da zagotovi stabilnost zaprtozančnega sistema [98, 69, 36]. Po drugi strani lahko uporabimo pristop z neskončnim predikcijskim horizontom, ki prav tako zagotavlja stabilnost [15].

Za izvedbo robustnejših regulatorjev, ki lahko kompenzirajo aditivne motnje v sistemu, je predlagan pristop z vodenjem *min-max* [36, 83]. Pristop deluje na principu razširitve kriterijske funkcije, tako da regulirni zakon išče njen minimum v najslabšem primeru.

9.4 Prekletstvo razsežnosti

Kot je zapisano v razdelku 9.2, ločujemo pri reševanju optimizacijskega problema dva osnovna pristopa.

1. Prvi pristop (sprotno reševanje optimizacijskega problema) ni namenjen vodenju hitrih sistemov, ampak samo počasnejših, tj. takih z relativno velikim časom vzorčenja. Pri vodenju počasnejših sistemov je namreč dovolj časa za izvedbo optimizacijskih postopkov v vsakem časovnem koraku sproti. Poleg tega ima sprotno reševanje optimizacijskega problema še naslednje slabosti:
 - za uporabo regulatorja je potrebna relativno draga strojna oprema, na kateri poteka optimizacija;
 - programska koda je zapletena in zato nepripravna za analizo (slabost pri aplikacijah, kjer je ključno zagotavljanje varnega delovanja);
 - računsko zahtevnost optimizacije je težko določiti, zato je ocena časa, ki je potreben za rešitev optimizacijskega problema v enem časovnem koraku, zahtevna naloga.

S sprotnim reševanjem optimizacijskega problema torej regulirni zakon določimo implicitno, tj. v okviru programa za reševanje optimizacijskih problemov.

2. Po drugi strani lahko z drugim pristopom (predhodno reševanje optimizacijskega problema za vse mogoče začetne vrednosti vektorja stanj) določimo

regulirni zakon v eksplicitni obliki, kar nam lahko omogoči hitro delovanje regulatorja, ki ne zahteva drage strojne opreme [83].

Kljub bistveni prednosti eksplicitno izraženega regulirnega zakona pa ima predhodno reševanje optimizacijskega problema dve glavni slabosti, ki v praksi pogosto onemogočita uporabo prediktivnega vodenja hibridnih sistemov na osnovi reševanja multiparametričnih mešanih celoštevilskih optimizacijskih problemov.

- Predhodno reševanje optimizacijskega problema je bistveno bolj računsko zahtevno od sprotnega reševanja. Problem moramo namreč rešiti za vse mogoče začetne vrednosti vektorja stanj $x(0)$, pri sprotnem reševanju pa rešujemo problem samo za konkretno začetno vrednost vektorja stanj.

Število podprostorov R , na katere je razdeljena množica H -koračne rešljivosti $\mathcal{X}_f^H = \{\mathcal{P}_r\}_{r=1}^R$, je v primeru multiparametričnega programiranja eksponentno odvisno od največjega predikcijskega horizonta H . Izkaže se lahko, da je optimizacijski problem preveč računsko zahteven že za relativno majhne predikcijske horizonte H , zato določitev eksplicitnega regulirnega zakona v praksi pogosto ni mogoča.

- Predpostavimo, da lahko – kljub veliki računski zahtevnosti – multiparametrični optimizacijski problem vseeno rešimo.

Kot smo omenili, je število podprostorov R , na katere je razdeljena množica H -koračne rešljivosti $\mathcal{X}_f^H = \{\mathcal{P}_r\}_{r=1}^R$, v primeru multiparametričnega programiranja eksponentno odvisna od največjega predikcijskega horizonta H . Rešitev multiparametričnega mešanega celoštevilskega optimizacijskega problema, tj. eksplicitni regulirni zakon, je lahko zaradi prevelikega števila podprostorov R prekompleksna. To pomeni, da je ne moremo implementirati na razpoložljivi strojni opremi ali da delovanje regulatorja ne bi bilo dovolj hitro, kar v praksi onemogoča uporabo obravnavane metode za vodenje.

Zaključimo lahko, da se v realnih primerih pri uporabi (multiparametričnih) (mešanih celoštevilskih) optimizacijskih algoritmov in njihovih različic pogosto ne moremo izogniti *prekletstvu razsežnosti* – problem je teoretično rešljiv, v praksi pa ga zaradi omejene računske moči ni mogoče rešiti.

9.5 Primer uporabe algoritma za vodenje temperature v šaržnem reaktorju

Algoritem za prediktivno vodenje hibridnih sistemov na osnovi reševanja mešanih celoštevilskih optimizacijskih problemov, primeren za sisteme, ki jih lahko zapišemo v obliki odsekoma afinega ali ekvivalentnega modela, smo preizkusili na primeru vodenja šaržnega reaktorja, ki je opisan v 3. poglavju.

9.5.1 Matematični model šaržnega reaktorja

Matematični model šaržnega reaktorja je podan v enačbah v razdelku 3.2. Dinamiko v jedru šaržnega reaktorja opisuje enačba (3.1), dinamiko v plašču pa enačba (3.2).

Masa komponent v jedru šaržnega reaktorja je $m = 600$ kg, masa vode v plašču je $m_w = 40$ kg. Specifična toplota komponent v jedru $c_w = 4200$ J kg⁻¹ K⁻¹, specifična toplota vode v plašču je $c_w = 4200$ J kg⁻¹ K⁻¹. Pretok vode v plašču znaša $\phi = 1,6$ kg s⁻¹, prevodnost toplotnega stika med jedrom in plaščem znaša $\lambda = 420$ W m⁻² K⁻¹, prevodnost toplotnega stika med plaščem in okolico znaša $\lambda_0 = 84$ W m⁻² K⁻¹, površina toplotnega stika med jedrom in plaščem znaša $S = 2$ m², površina toplotnega stika med plaščem in okolico znaša $S_0 = 2$ m². Temperatura sveže vhodne vode je lahko $T_C = 12$ °C ali $T_H = 75$ °C.

9.5.2 Omejitev nabora mogočih kombinacij vhodnih vektorjev

Že na začetku načrtovanja vodenja je smiselno omejiti množico vrednosti vektorja vhoda. Omejitev izhaja iz fizičnih in tehnoloških zahtev in pogojev v sistemu, ki ga želimo voditi, saj na ta način izločimo vse nemogoče ali nesmiselne kombinacije vhodnih vektorjev.

Na ta način lahko določimo matriko vhodnih vektorjev P , ki jo podaja izraz (9.32).

$$P = \begin{bmatrix} 0 & 0,01 & 0,02 & 0,05 & 0,1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (9.32)$$

V izrazu (9.32) označuje vsak stolpec enega izmed mogočih vhodnih vektorjev. Vrstice v matriki vhodnih vektorjev P imajo naslednji pomen:

- prva vrstica označuje položaj mešalnega ventila
 $k_M \in \{0, 0.01, 0.02, 0.05, 0.1, 1\}$;
- druga vrstica označuje položaj zapornega ventila za hladno vodo
 $k_C \in \{0, 1\}$;
- tretja vrstica označuje položaj zapornega ventila za vročo vodo
 $k_H \in \{1, 0\}$.

9.5.3 Vodenje

Algoritem za prediktivno vodenje, ki ga preizkušamo v tem eksperimentu, za delovanje potrebuje primeren matematični model šaržnega reaktorja. V našem primeru smo izbrali odsekoma afini model, katerega razvoj je natančno opisan v razdelku 6.4.4.

Pri vodenju temperature v jedru šaržnega reaktorja smo izbrali največji predikcijski horizont $H = 4$. Na ta način algoritem upošteva predikcijo obnašanja sistema za $t_{pred} = H \cdot t_S = 4 \cdot 10 \text{ s} = 40 \text{ s}$ naprej.

9.5.4 Kriterijska funkcija

Določitev kriterijske funkcije ima velik vpliv na kakovost vodenja, saj neposredno določa obnašanje zaprtozančnega sistema, poleg tega pa vpliva tudi na računsko zahtevnost algoritma za prediktivno vodenje.

Preizkusili smo vodenje z uporabo preproste kriterijske funkcije, ki jo podaja izraz (9.33).

$$J_H = \sum_{h=1}^H (T_h - T_{h,ref})^2 \quad (9.33)$$

Opisana kriterijska funkcija upošteva kvadrat odstopanja temperature v jedru T , ki jo določa predikcijski model, od referenčne temperature T_{ref} .

Parametri modela in vodenja, ki so potrebni za zagon algoritma za prediktivno vodenje hibridnih sistemov na osnovi reševanja mešanih celoštevilskih optimizacijskih problemov, so zbrani v standardni strukturi, ki se uporablja v programskem orodju Multi-Parametric Toolbox [87] in je prikazana v dodatku A.

9.5.5 Komentar

Že osnovne različice² mešanega celoštevilskega kvadratičnega programa, ki ga uporabljamo pri vodenju pri pogojih, ki so opisani v prejšnjih razdelkih, ne moremo uporabiti za vodenje v realnem času.

²Osnovna različica pomeni vodenje, pri katerem uporabimo preprosto kriterijsko funkcijo v enačbi 9.33 in relativno kratek predikcijski čas $t_{pred} = H \cdot t_S = 4 \cdot 10 \text{ s} = 40 \text{ s}$, kot je opisano v razdelku 9.5. Primerjaj s kompleksnejšo kriterijsko funkcijo v enačbah (10.20), (10.23) in (10.22) in daljšim predikcijskim časom $t_{pred} = H \cdot t_S = 60 \cdot 10 \text{ s} = 600 \text{ s}$, kot je opisano v razdelku 10.5.

Tabela 9.1: Čas t_k , ki ga algoritem porabi za izračun rešitve optimizacijskega problema v enem časovnem koraku.

H	t_{pred} [s]	t_k [s]
1	10	0,12
2	20	0,32
3	30	30
4	40	7323

Računska zahtevnost algoritma za vodenje je odvisna od največjega predikcijskega horizonta. Tabela 9.1 podaja čas t_k , ki ga algoritem porabi za izračun rešitve optimizacijskega problema v enem časovnem koraku, glede na nastavitev največjega predikcijskega horizonta H oz. časa predikcije t_{pred} .

Ugotovimo lahko, da uporaba algoritma za prediktivno vodenje hibridnih sistemov na osnovi reševanja mešanih celoštevilskih optimizacijskih problemov za vodenje šaržnega reaktorja, kot je opisano v razdelku 9.5, ni smiselna.

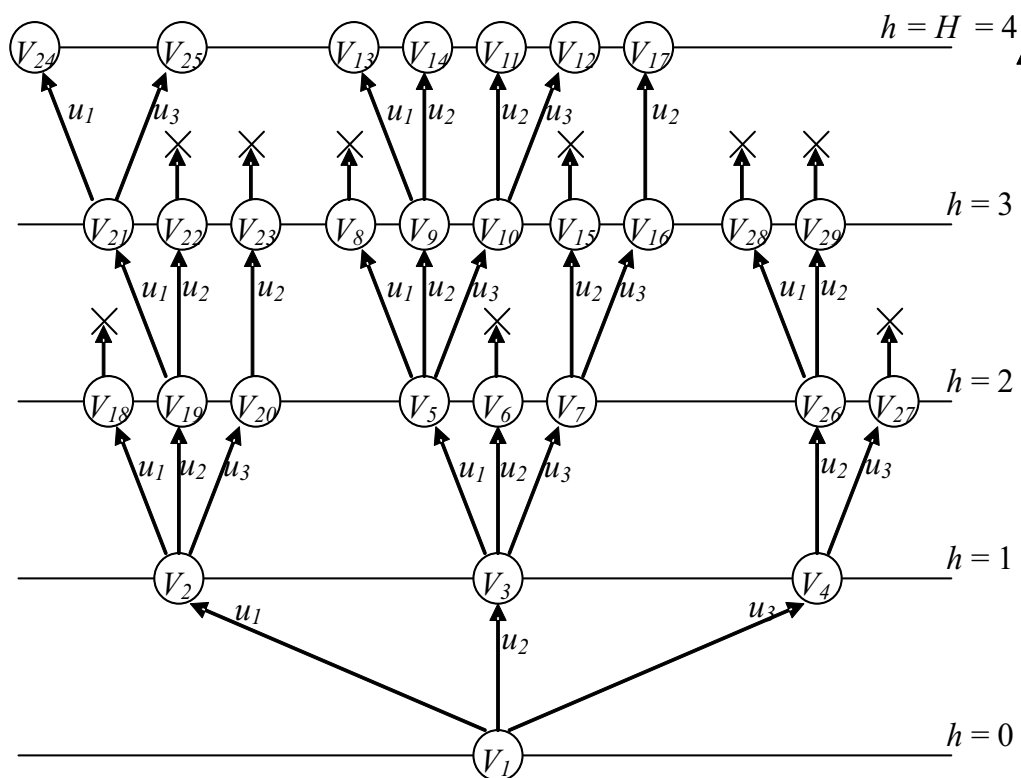
10. Prediktivno vodenje na osnovi analize dosegljivosti

V tem poglavju bomo opisali metodo za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi, ki deluje na osnovi analize dosegljivosti [110, 114, 78, 111, 79, 112].

10.1 Problemsko drevo

Pri tej metodi prediktivnega vodenja smo se omejili na sisteme s samimi diskretnimi vhodi, zato lahko predikcijo oziroma razvoj stanj sistema preko predikcijskih korakov (do največjega predikcijskega horizonta, tj. za največ H korakov naprej) predstavimo s pomočjo problemskega drevesa, ki je prikazano na sliki 10.1. Slika prikazuje primer problemskega drevesa, kjer je največji predikcijski horizont $H = 4$, število mogočih različnih vrednosti vhodnega vektorja pa $M = 3$.

Na sliki 10.1 so s krogi označena vozlišča drevesa, ki pripadajo dosegljivim stanjem, s puščicami pa veje drevesa, ki – glede na vrednost vektorja vhodov oz. zaporedno označbo $j = 1, 2, \dots, M$, ki določa vektor vhodov iz določenega nabora kombinacij vrednosti vektorja vhoda – povezujejo po dve vozlišči, če je naslednje vozlišče dosegljivo. Vsako vejo torej določa ustrezen vektor vhodov, katerega posledica je povezava dveh vozlišč, ki je s to vejo predstavljena.



Slika 10.1: Primer razvitega problemskega drevesa za največji predikcijski horizont $H = 4$ in $M = 3$ mogoče različne vrednosti vhodnega vektorja. Končno optimalno vozlišče je V_{14} , zato je v tem koraku izbran in uporabljen vhodni vektor $u_{opt} = u_2$.

10.1.1 Razvoj problemskega drevesa

Korenino drevesa predstavlja začetno vozlišče V_1 , ki zaznamuje izhodiščno stanje sistema $x(k | k)$, pri čemer je $h = 0$. Vozlišče se označi kot raziskano, nato pa vsa mogoča naslednja stanja sistema (vozlišča) algoritem določi s predikcijo, pri čemer upošteva določeni nabor kombinacij vrednosti vektorja vhoda u_j , kjer je $j = 1, 2, \dots, M$, in dosegljivost naslednjih vozlišč. V naslednjem predikcijskem koraku $h = 1$ lahko torej drevo razvejimo na največ M novih vozlišč V_i , do katerih vodijo pripadajoče veje, ki jih označimo z ustreznim vektorjem vhodov.

Indeks $i \in \{1, 2, \dots\}$ predstavlja vrstni red vozlišč (in vej) v problemskem drevesu (glej sliko 10.1). Vsakemu novemu vozlišču V_i priredi algoritem vrednost

kriterijske funkcije J_i . Iz vsakega od novih vozlišč moramo razviti drevo naprej v višino (glede na predikcijske korake h), nato pa v širino (glede na določeni nabor kombinacij vrednosti vektorja vhoda u_j , kjer je $j = 1, 2, \dots, M$ in dosegljivost naslednjih vozlišč).

Algoritem za prediktivno vodenje mora čimprej zaznati in izločiti *neperspektivne* smeri razvoja stanj sistema. Prej ko jih algoritem izloči, bolj se zmanjša kompleksnost, saj se tako izognemo računanju z neperspektivnimi stanji oziroma z neperspektivnimi smermi razvoja stanj sistema. Z neperspektivno označujemo tisto smer razvoja stanj vodenega sistema, za katero lahko z gotovostjo trdimo, da, tudi če bi nadaljevali s predikcijo do največjega predikcijskega horizonta H , ne more biti optimalna rešitev algoritma za prediktivno vodenje glede na izbrano kriterijsko funkcijo. (Ne)perspektivnost stanj sistema določimo z uporabo ustrezne kriterijske funkcije, ki jo predpišemo v algoritmu za prediktivno vodenje in podaja oceno optimalnosti vodenja (modela) sistema.

Neperspektivno smer razvoja stanj sistema je torej dobro čimprej zaznati in zavreči. Zato se mora problemsko drevo najprej razviti v višino, šele nato v širino. Razvoj drevesa nadaljujemo iz tistega vozlišča, ki je v določenem predikcijskem koraku h najperspektivnejše, kar določimo glede vrednosti kriterijske funkcije J_i , ki pripadajo posameznim vozliščem na horizontu h .

Ko izberemo najboljše vozlišče na horizontu h , ga označimo kot raziskano, algoritem pa s predikcijo ponovno določi vsa mogoča naslednja stanja sistema (vozlišča v naslednjem predikcijskem koraku), pri čemer seveda upošteva določeni nabor kombinacij vrednosti vektorja vhoda u_j , kjer je $j = 1, 2, \dots, M$ in dosegljivost naslednjih vozlišč. V naslednjem predikcijskem koraku $h + 1$ lahko torej drevo spet razvejimo na največ M novih vozlišč, do katerih vodijo pripadajoče veje, ki jih označimo z ustreznim vektorjem vhodov. Vsem tem novim vozliščem moramo ponovno določiti pripadajoče vrednosti kriterijske funkcije.

Gradnja problemskega drevesa se torej nadaljuje najprej v višino, šele nato pa

v širino. Tak način razvoja omogoča, da že po prvi določitvi zaporedja vrednosti vektorja vhoda – glej enačbo (8.10) – z veliko verjetnostjo dobimo rešitev, ki je blizu (ali celo enaka) končnemu *optimalnemu* zaporedju vrednosti vektorja vhoda. To lahko v splošnem precej pohitri izvajanje algoritma, saj lahko – glede na začetku določeno suboptimalno rešitev – relativno zgodaj izločimo neperspektivne smeri razvoja in na ta način ne izgublamo časa z računanjem predikcije v teh smereh.

10.1.2 Pogoja, ki ju preverjamo med gradnjo problemskega drevesa v vsakem vozlišču

Gradnja problemskega drevesa se nadaljuje na opisan način, dokler ni izpolnjen eden od dveh pogojev:

1. vrednost kriterijske funkcije v trenutnem vozlišču V_i presega trenutno optimalno vrednost kriterijske funkcije $J_i \geq J_{opt}|_{V_{i-1}}$;
2. dosežena je zgornja meja predikcijskega horizonta: $h|_{V_i} = H$.

Omenimo, da algoritem za prediktivno vodenje v vsakem koraku pred začetkom računanja predikcije nastavi trenutno optimalno vrednost kriterijske funkcije na neskončno $J_{opt}|_{V_1} = \infty$.

Prvi pogoj poskrbi, da čim prej zavržemo neperspektivne smeri razvoja problemskega drevesa, drugi pa pove, da smo drevo razvili že dovolj v višino, tj. do največjega predikcijskega horizonta H .

Če je izpolnjen kateri od pogojev, nadaljujemo, kot je navedeno spodaj.

1. Če vozlišče V_i izpolnjuje prvi pogoj ($J_i \geq J_{opt}|_{V_{i-1}}$), ga kot neperspektivnega zavržemo in razvoj drevesa v tej smeri ustavimo.
2. Če vozlišče V_i ne izpolnjuje prvega, torej velja $J_i < J_{opt}|_{V_{i-1}}$, izpolnjuje pa drugi pogoj ($h|_{V_i} = H$), to pomeni, da smo dobili novo, bolj optimalno

rešitev zaporedja vrednosti vektorja vhoda $U_k^{k+H-1}|_{V_i}$, kot je bila prejšnja.

V tem primeru moramo narediti tri stvari:

- shranimo to vozlišče kot novo optimalno vozlišče: $V_{opt}|_{V_i} = V_i$;
- shranimo zaporedje $U_k^{k+H-1}|_{V_i}$, ki je privedlo do vozlišča V_i , kot novo optimalno zaporedje: $U_{k, opt}^{k+H-1}|_{V_i} = U_k^{k+H-1}|_{V_i}$;
- shranimo vrednost kriterijske funkcije, ki pripada vozlišču V_i , kot novo optimalno vrednost kriterijske funkcije: $J_{opt}|_{V_i} = J_i$.

Poudariti je treba, da v primeru, ko algoritem za prediktivno vodenje šele prvič doseže največji predikcijski horizont $h = H$, zaporedje vrednosti vektorja vhoda ni novo, ampak šele prvo potencialno optimalno zaporedje $U_{k, opt}^{k+H-1}|_{V_i}$. Tedaj prvi pogoj ($J_i \geq J_{opt}|_{V_{i-1}}$) nikakor ne more biti izpolnjen, saj je trenutna optimalna vrednost kriterijske funkcije nastavljena na neskončno ($J_{opt}|_{V_{i-1}} = \infty$).

10.1.3 Nadaljevanje razvoja problemskega drevesa po izpolnitvi enega od pogojev

V primeru, da je prišlo do izpolnitve katerega od pogojev, ki smo ju opisali v razdelku 10.1.2, nadaljujemo razvoj problemskega drevesa z iskanjem optimalnega zaporedja vrednosti vektorja vhoda $U_{k, opt}^{k+H-1}|_{V_{i+1}}$ tako, da se vrnemo nazaj po drevesu na najvišji predikcijski korak h , na katerem še najdemo neraziskana vozlišča. Nato nadaljujemo iskanje v smeri še neraziskanih vozlišč, pri čemer najprej izberemo najbolj perspektivno neraziskano vozlišče, tj. tisto, ki ima najmanjšo vrednost kriterijske funkcije na svojem predikcijskem koraku.

V algoritmu za prediktivno vodenje se opisani postopek ponavlja, dokler niso raziskana vsa vozlišča v problemskem drevesu in lahko rečemo, da je problemsko drevo (v okviru največjega predikcijskega horizonta H) do konca razvito. Tedaj lahko privzamemo, da je zadnje optimalno zaporedje vrednosti vektorja vhoda v časovnem koraku k tudi končno optimalno zaporedje vrednosti vektorja vhoda.

Prvi element iz tega zaporedje uporabimo kot vhod v sistem v časovnem koraku k , kot kaže enačba (10.1).

$$u_{opt}(k) = U_{k, opt}^{k+H-1}|_{V_{i_{max}}}(1); \quad (10.1)$$

kjer je $i_{max} = \max\{i : \exists U_{k, opt}^{k+H-1}|_{V_i}\}$

V naslednjem časovnem koraku $k + 1$ celotni algoritem iskanja optimalnega zaporedja vektorja vhoda ponovimo.

10.2 Analiza dosegljivosti

V splošnem lahko ugotovimo, da na sistemu ni vedno mogoče uporabiti vseh mogočih kombinacij vhodov, tj. vseh različnih vrednosti vektorja vhodov – glej npr. vhodni vektor u_1 v vozlišču V_4 na sliki 10.1. To je po eni strani posledica omejitev kombinacij vrednosti vhodnega vektorja, ki jih lahko določimo že na začetku načrtovanja vodenja in izhajajo iz zahtev za delovanje sistema, po drugi strani pa možnosti, da določeno zaporedje vrednosti vektorja vhodov pripelje stanja sistema v nedovoljeno področje. Do omejitev v sistemu lahko pride iz različnih razlogov:

- fizikalne ipd. omejitve – te morajo biti vključene že v model sistema, ki ga vodimo;
- omejitve v zvezi s kakovostjo vodenja – te so navadno implicitno vključene v kriterijski funkciji.

Ker na sistemu ne moremo (vedno) uporabiti vseh mogočih kombinacij vrednosti vektorja vhodov oziroma vseh mogočih zaporedij vrednosti vhodnih vektorjev U_k^{k+H-1} v okviru največjega predikcijskega horizonta H , je smiselno v algoritmu za prediktivno vodenje to upoštevati in zavreči nemogoče vrednosti vhodnega

vektorja. Na ta način lahko neposredno zmanjšamo računsko zahtevnost problema vodenja sistema. Izbiro vrednosti vhodnega vektorja, ki jih bomo zavrgli, lahko določimo z uporabo analize dosegljivosti.

Algoritem za prediktivno vodenje torej pri gradnji problemskega drevesa, kot smo ga opisali v razdelku 10.1, uporablja strategijo *razmejitev in omejitev* (angl. *branch and bound*). To pomeni, da algoritem razvija problemsko drevo, ki ga po eni strani s predikcijo *razveji*, po drugi strani pa ga, z uporabo analize dosegljivosti, določenega (omejenega) nabora kombinacij vrednosti vektorja vhoda ter čimprejšnjim zaznavanjem in izločanjem neperspektivnih smeri razvoja stanj sistema, *omeji*.

Algoritem za prediktivno vodenje raziskuje samo tista stanja sistema, ki so dosegljiva, oziroma tiste vhode v sistem, ki so dovoljeni. Denimo, da je $x(k+h | k)$ stanje sistema v predikcijskem koraku h . Z analizo dosegljivosti izračunamo vsa mogoča stanja sistema $x_j(k+h+1 | k)$ v predikcijskem koraku $h+1$. S tem določimo tudi vse mogoče kombinacije vrednosti vektorja vhoda $u_j(k+h | k)$, ki sistema ne pripeljejo v nedovoljeno področje stanj oziroma so sploh dovoljene, kar izhaja iz delovanja samega sistema. Pri tem indeks j označuje dosegljiva stanja $x_j(k+h+1 | k)$ in pripadajoče vhode $u_j(k+h | k)$, ki so sistem pripeljali v ta stanja.

10.3 Kriterijska funkcija

Ustrezna določitev kriterijske funkcije ima velik vpliv na kakovost vodenja, saj neposredno določa obnašanje zaprtozančnega sistema.

Poleg tega kriterijska funkcija določa (ne)perspektivnost posameznih vozlišč v problemskem drevesu in s tem posledično izbiro vozlišča, iz katerega bomo nadaljevali razvoj problemskega drevesa – glej razdelek 10.1 – zato je povezana tudi z velikostjo do konca razvitega problemskega drevesa. S tem pa posredno vpliva tudi na računsko zahtevnost algoritma za prediktivno vodenje.

Kriterijsko funkcijo (za vozlišče V_i), v kateri je lahko vsebovan tudi referenčni potek signalov, v obliki, ki je primerna za uporabo hibridnega mehkega modela (glej 5. poglavje) zapišemo, kot kaže enačba (10.2).

$$J_i = l(X_{k+1}^{k+h} |_{V_i}, Q_{k+1}^{k+h} |_{V_i}, U_k^{k+h-1} |_{V_i}, k, h |_{V_i}) \quad (10.2)$$

V enačbi (10.2) označuje $X_{k+1}^{k+h} |_{V_i}$ zaporedje vektorjev zveznih stanj x , $Q_{k+1}^{k+h} |_{V_i}$ je zaporedje diskretnih stanj q , $U_k^{k+h-1} |_{V_i}$ pa zaporedje vhodov u , ki so pripeljali v vozlišče V_i . $h |_{V_i}$ označuje predikcijski korak v vozlišču V_i .

10.3.1 Pogoji za ustreznost kriterijske funkcije

Kot smo opisali v razdelku 10.1, je najboljši način za doseganje časovne učinkovitosti algoritma za prediktivno vodenje ta, da razvoj drevesa nadaljujemo iz tistega vozlišča, ki je v določenem predikcijskem koraku h najbolj perspektivno. Perspektivnost posameznih vozlišč v predikcijskem koraku h določimo glede vrednosti kriterijske funkcije, ki jim pripadajo.

Če hočemo, da bi predlagani mehanizem algoritma za prediktivno vodenje deloval, moramo zagotoviti, da vrednost kriterijske funkcije v vsakem primeru z višanjem predikcijskega koraka h proti končnemu predikcijskemu horizontu H monotono narašča, kot je zapisano v izrazu (10.3).

$$\begin{aligned} & l(X_{k+1}^{k+h} |_{V_i}, Q_{k+1}^{k+h} |_{V_i}, U_k^{k+h-1} |_{V_i}, k, h |_{V_i}) - \\ & - l(X_{k+1}^{k+h} |_{V_i}, Q_{k+1}^{k+h} |_{V_i}, U_k^{k+h-1} |_{V_i}, k, h |_{V_i} - 1) \geq 0 \quad (10.3) \\ & \text{za } h = 2, 3, \dots, H \end{aligned}$$

Na ta način smo lahko prepričani, da v neperspektivnih smereh razvoja problemskega drevesa na največjem predikcijskem horizontu H ne bomo dobili boljše rešitve, kot je trenutna optimalna rešitev $U_{k, opt}^{k+H-1} |_{V_i}$. Če torej kriterijska funkcija

(10.2) izpolnjuje pogoj (10.3), lahko neperspektivne smeri razvoja brez izgube zavržemo. Na ta način lahko bistveno zmanjšamo razvejanost do konca raziskalnega problemskega drevesa in s tem računsko zahtevnost algoritma za prediktivno vodenje.

10.3.2 Predlog oblike kriterijske funkcije

V nadaljevanju bomo opisali predlog uporabne¹ oblike kriterijske funkcije, ki zadošča pogoj monotonega naraščanja (10.3).

Kriterijska funkcija naj ima obliko, ki jo podaja izraz (10.4). Pri tem naj veljata pogoja v izrazih (10.5) in (10.6).

$$\begin{aligned}
 J_i &= l(X_{k+1}^{k+h} |_{V_i}, Q_{k+1}^{k+h} |_{V_i}, U_k^{k+h-1} |_{V_i}, k, h |_{V_i}) = \\
 &= l(X_{k+1}^{k+h-1} |_{V_i}, Q_{k+1}^{k+h-1} |_{V_i}, U_k^{k+h-2} |_{V_i}, k, h |_{V_i} - 1) + \\
 &+ \Delta l(x(k+h | k) |_{V_i}, q(k+h | k) |_{V_i}, u(k+h-1 | k) |_{V_i}, k, h |_{V_i})
 \end{aligned} \tag{10.4}$$

za vsak V_i in $h = 1, 2, \dots, H$

$$\Delta l(x(k+h | k) |_{V_i}, q(k+h | k) |_{V_i}, u(k+h-1 | k) |_{V_i}, k, h |_{V_i}) \geq 0 \tag{10.5}$$

za vsak V_i in $h = 1, 2, \dots, H$

$$\begin{aligned}
 l((), (), (), k, 0) &= 0 \\
 \text{za } h &= 0
 \end{aligned} \tag{10.6}$$

Kriterijska funkcija v vozlišču V_i je torej v glede na predikcijski korak h določena kot vsota dveh členov.

- Prvi člen določa vrednost kriterijske funkcije

¹Z uporabno obliko kriterijske funkcije mislimo obliko, ki jo lahko uporabimo pri večini sistemov, ki jih želimo voditi z algoritmom za prediktivno vodenje.

$l(X_{k+1}^{k+h-1} |_{V_i}, Q_{k+1}^{k+h-1} |_{V_i}, U_k^{k+h-2} |_{V_i}, k, h |_{V_i} - 1)$, izračunano v vozlišču v prejšnjem predikcijskem koraku $h - 1$, iz katerega je sistem prišel v trenutno vozlišče V_i . Omenimo, da je vrednost kriterijske funkcije v korenini problemskega drevesa, tj. v izhodiščnem vozlišču V_1 enaka 0, kar določa enačba (10.6).

- Drugi člen je poljubna nenegativna (glej pogoj (10.5)) funkcija

$\Delta l(x(k + h | k) |_{V_i}, q(k + h | k) |_{V_i}, u(k + h - 1 | k) |_{V_i}, k, h |_{V_i})$, ki oceni stopnjo optimalnosti vodenja v trenutnem vozlišču.

V veliki večini primerov lahko kriterij kakovosti vodenja Δl izrazimo z vsoto ustrezno (pozitivno) obteženih normiranih² razlik med dejanskim in želenim potekom vrednosti komponent vektorja stanj sistema oziroma vektorja vhodov v sistem. V tem primeru je funkcija Δl nenegativna in s tem je izpolnjen pogoj (10.5).

Po potrebi lahko v funkciji Δl upoštevamo tudi odvisnost od koraka v diskretnem času k – če je npr. referenčni potek signalov že vsebovan v kriterijski funkciji – pa tudi predikcijskega koraka h – če so uteži v kriterijski funkciji odvisne od predikcijskega koraka h , v katerem se nahaja vozlišče V_i . Pozorni moramo biti le, da je pogoj (10.5) vedno izpolnjen.

Vrednost $\Delta l(x(k + h | k) |_{V_i}, q(k + h | k) |_{V_i}, u(k + h - 1 | k) |_{V_i}, k, h |_{V_i})$ se prišteva vsoti vrednosti kriterija kakovosti vodenja, ki so bile izračunane vzdolž poti od korenine problemskega drevesa V_1 do vozlišča V_i . Kriterijsko funkcijo lahko torej zapišemo v nerekurzivni obliki, kot jo podaja enačba (10.7).

$$J_i = \sum_{j=0}^h \Delta l(x(k + j | k) |_{V_i}, q(k + j | k) |_{V_i}, u(k + j - 1 | k) |_{V_i}, k, j |_{V_i}) \quad (10.7)$$

²Uporabne so različne vektorske norme $\|\cdot\|$, ki določajo skalarno vrednost razdalje med vektorjema.

Glede na to, da kriterijska funkcija Δl ni omejena z ničemer drugim kot s pogojem (10.5), ki pa ga ni težko izpolniti, lahko rečemo, da je predlagana oblika kriterijske funkcije uporabna v praksi.

Če je pogoj (10.5) izpolnjen, lahko dokažemo, da je izpolnjen tudi pogoj za delovanje algoritma za prediktivno vodenje (10.3): če v izraz (10.3) vstavimo razčlenjeno kriterijsko funkcijo v izrazu (10.4), dobimo po krajšavi pogoj (10.5).

10.4 Računska zahtevnost

Računska zahtevnost obravnavanega algoritma za prediktivno vodenje je prvenstveno odvisna od največjega predikcijskega horizonta H , ki določa višino problemskega drevesa, in števila diskretnih vhodov z ustreznim številom mogočih vrednosti, ki jih lahko zavzame posamezen diskretni vhod, kar vpliva na razvejavnost drevesa.

Denimo, da lahko i -ti diskretni vhod, kjer je $i \in \{1, \dots, l\}$, zavzame m_i različnih diskretnih vrednosti. V tem primeru lahko število vseh mogočih kombinacij vrednosti vhodnega vektorja izračunamo, kot kaže enačba (10.8).

$$M_0 = m_1 \cdot \dots \cdot m_l \quad (10.8)$$

Množico mogočih zaporedij vhodnih vektorjev podaja izraz (10.9).

$$U_k^{k+H-1} \in \{u_1, \dots, u_{M_0}\}^H \quad (10.9)$$

V izrazu (10.9) označuje u_j , kjer je $j \in \{1, \dots, M_0\}$, enega od mogočih vhodnih vektorjev.

Ocena računske zahtevnosti obravnavanega algoritma za prediktivno vodenje je sorazmerna velikosti nabora vseh mogočih zaporedij vrednosti vektorja vhoda M_0 v okviru največjega predikcijskega horizonta H , ki jih ocenjuje algoritem v

enem časovnem koraku k . Omeniti je treba, da ocena velja za večja števila M_0 in H , saj imajo poraba časa zaradi inicializacije in ostali režijski stroški pri manjših številih večji relativni vpliv. Oceno računske zahtevnosti podaja izraz (10.10).

$$C_0 \propto M_0^H \quad (10.10)$$

10.4.1 Zmanjšanje računske zahtevnosti z uporabo analize dosegljivosti

Prvi pristop k zmanjšanju računske zahtevnosti v obravnavanem algoritmu za prediktivno vodenje predstavlja uporaba analize dosegljivosti – glej razdelek 10.2. Učinkovitost pristopa oz. stopnja zmanjšanja računske zahtevnosti je tesno povezana s sistemom, ki ga želimo voditi, oz. njegovim modelom, izbiro kriterijske funkcije, referenčno trajektorijo in trenutnim stanjem sistema, zato je v splošnem ni mogoče enostavno računsko določiti.

10.4.2 Zmanjšanje računske zahtevnosti z omejitvijo nabora mogočih kombinacij vrednosti vektorja vhoda

Zaradi fizičnih in tehnoloških zahtev in pogojev v sistemu, ki ga želimo voditi, je v splošnem smiselno že na začetku načrtovanja vodenja omejiti nabor mogočih kombinacij vrednosti vektorja vhoda. To povzroči zmanjšanje računske zahtevnosti problema vodenja.

Predpostavimo, da smo nabor mogočih kombinacij vrednosti vektorja vhoda, ki izhajajo iz zahtev za delovanje sistema, omejili na $M < M_0$. V tem primeru množico mogočih zaporedij vhodnih vektorjev podaja izraz (10.11).

$$U_k^{k+H-1} \in \{u_1, \dots, u_M\}^H \quad (10.11)$$

V izrazu (10.11) označuje u_j , kjer je $j \in \{1, \dots, M\}$, enega od mogočih vho-

dnih vektorjev.

Kot smo že omenili, je ocena računske zahtevnosti obravnavanega algoritma za prediktivno vodenje sorazmerna velikosti nabora vseh mogočih zaporedij vrednosti vektorja vhoda v okviru največjega predikcijskega horizonta H , ki jih ocenjuje algoritem v enem časovnem koraku k . V tem primeru jo podaja enačba (10.12).

$$C_M \propto M^H \quad (10.12)$$

Kot je zapisano v enačbi (10.13), lahko z zmanjšanjem nabora kombinacij vrednosti vektorja vhoda zmanjšamo tudi kompleksnost problema vodenja za faktor, ki je eksponentno odvisen od končnega predikcijskega horizonta H , osnova pa je odvisna od stopnje zmanjšanja nabora kombinacij vrednosti vektorja vhoda.

$$\frac{C_M}{C_0} \approx \left(\frac{M}{M_0} \right)^H \quad (10.13)$$

Zmanjšanje števila vseh mogočih kombinacij vrednosti vektorja vhoda M_0 na ustrezen nabor mogočih kombinacij vrednosti vektorja vhoda M ima velik vpliv na kompleksnost problema vodenja, kar še posebej velja v primerih, ko je izbrana večja vrednost največjega predikcijskega horizonta H .

10.4.3 Zmanjšanje računske zahtevnosti z zadrževanjem vrednosti vektorja vhodov skozi več časovnih korakov

V modelih, ki jih uporabljamo za prediktivno vodenje, je čas vzorčenja t_S vezan na več dejavnikov: dinamiko³ sistema, ki ga vodimo, želena natančnost⁴ predikcije in morebitni čas vzorčenja senzorjev,⁵ ki jih v sistemu za vodenje uporabljamo.

³Hitrim sistemom ustreza krajši, počasnejšim pa daljši čas vzorčenja t_S .

⁴Za doseganje ustrezne natančnosti predikcije moramo uporabiti dovolj kratek čas vzorčenja t_S .

⁵V primeru, da senzorji merjeno količino vzorčijo, je potrebna ustrezna sinhronizacija. Če senzorji podajajo merjeno količino zvezno, nimajo vpliva na čas vzorčenja t_S .

V splošnem želimo načrtati prediktivno vodenje, ki v vsakem koraku izvaja predikcijo za določen največji predikcijski čas t_{pred} naprej. Z največjim predikcijskim časom pa sta povezana tudi največji predikcijski horizont H in čas vzorčenja t_S , kot kaže enačba (10.14).

$$H = \frac{t_{pred}}{t_S} \quad (10.14)$$

Čas vzorčenja t_S je določen glede na sistem, ki ga želimo voditi. Dejavniki, ki vplivajo na čas vzorčenja, so dinamika sistema, potrebna natančnost predikcije, morebitni čas vzorčenja senzorjev ipd. – glej tudi razdelka 2.4 in 8.3. Kot vidimo iz enačbe (10.14), lahko ob izbranem času vzorčenja t_S pri večanju največjega predikcijskega časa t_{pred} največji predikcijski horizont H precej naraste. Z naraščanjem največjega predikcijskega horizonta H v splošnem eksponentno narašča tudi računski zahtevnost problema prediktivnega vodenja, kar pa lahko kmalu privede do situacije, ko je čas, ki ga porabi algoritem za prediktivno vodenje za izračun rešitve, daljši od časa vzorčenja t_S , tako da vodenje sistema v realnem času ni več mogoče.

Opisani situaciji se lahko v mnogo primerih izognemo z zadrževanjem vrednosti vektorja vhodov skozi več časovnih korakov. V mnogo sistemih lahko namreč dopustimo (ali celo želimo), da se vrednosti vektorja vhodov ne spreminjajo z vsakim časom vzorčenja t_S . Če je npr. čas vzorčenja t_S relativno kratek, akuatorji pa bi bili s spreminjanjem vhodnega vektorja v vsakem koraku preveč obremenjeni, lahko predpišemo, naj se vektor vhodov spreminja samo vsakih Z časovnih korakov, tj. da zadrži svojo vrednost za čas $Z \cdot t_S$.

Algoritem za prediktivno vodenje je potrebno v primeru zadrževanja vrednosti vektorja vhodov skozi več časovnih korakov prilagoditi. Upoštevati moramo vrednost spremenljivke parametra Z : razvoj problemskega drevesa – glej razdelek 10.1 – moramo nekoliko modificirati.

Prehod na naslednji predikcijski horizont izvedemo s predikcijo preko *vmesnih*

vozlišč $V_{i,m,z}$. Pri tem indeksi i , m , kjer velja $m \in \{1, 2, \dots, M\}$, in z , kjer velja $z \in \{2, 3, \dots, Z\}$, pomenijo sledeče:

- i označuje, iz katerega vozlišča izhajamo;
- m označuje, v kateri smeri peljemo sistem, tj. katera iz nabora kombinacij vrednosti vektorja vhoda pripada veji v vmesno vozlišče $V_{i,m,z}$;
- z označuje, na kateri stopnji na poti proti naslednjemu predikcijskemu koraku h se nahaja vmesno vozlišče $V_{i,m,z}$.

Ugotovimo torej, da mora algoritem, da bi dosegel naslednji predikcijski korak, predikcijo izvesti Z -krat. Slika 10.2 prikazuje spremenjen izsek problemskega drevesa v primeru zadrževanja vrednosti vektorja vhodov skozi $Z = 3$ časovne korake za $M = 3$ mogoče različne vrednosti vhodnega vektorja.

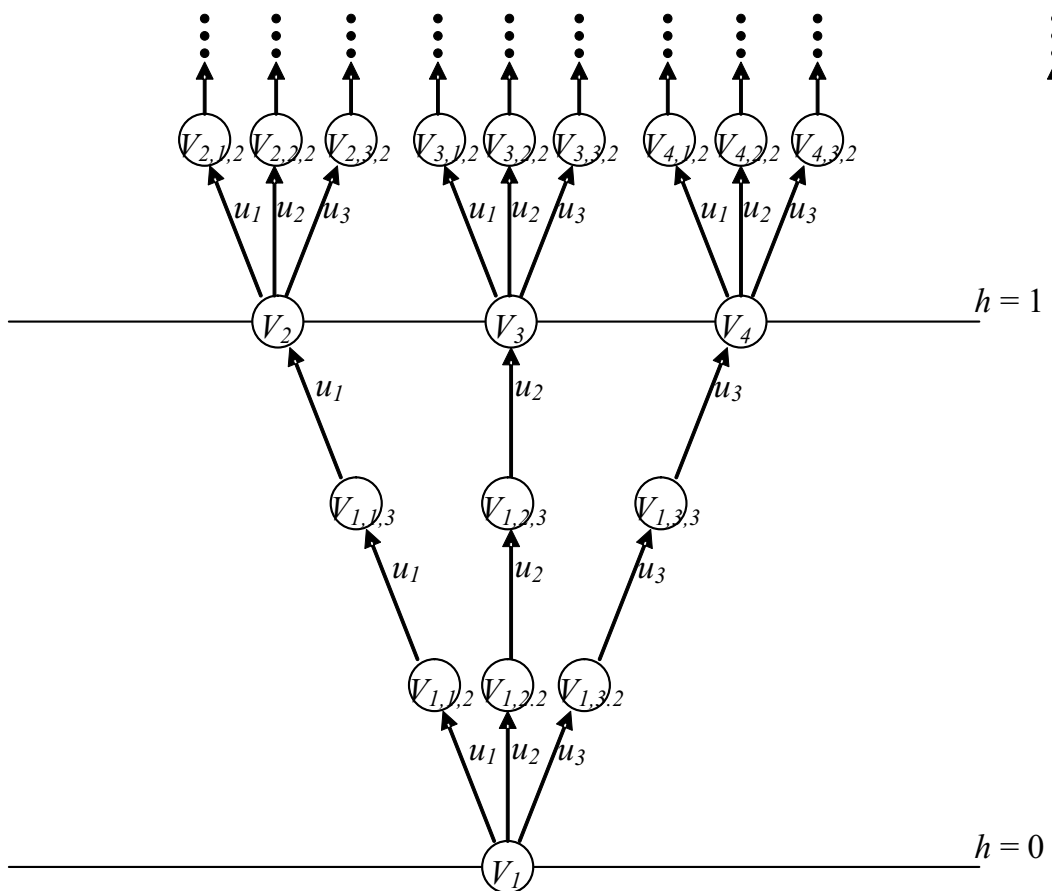
Ker v splošnem želimo načrtati prediktivno vodenje, ki v vsakem koraku izvaja predikcijo za izbrani največji predikcijski čas t_{pred} naprej, hkrati pa predpišemo, da naj vektor vhodov zadrži svojo vrednost za čas $Z \cdot t_S$, se z večanjem števila Z manjša potreben največji predikcijski horizont H_Z , kot je zapisano v enačbi (10.15).

$$H_Z = \frac{t_{pred}}{t_S} \cdot \frac{1}{Z} \quad (10.15)$$

Če predpostavimo, da sta izbrani največji predikcijski čas t_{pred} in čas vzorčenja t_S fiksna – torej velja enačba (10.16) – lahko ocenimo računsko zahtevnost problema prediktivnega vodenja z zadrževanjem vrednosti vektorja vhodov skozi Z časovnih korakov, kot jo podaja izraz (10.17).

$$H_Z = \frac{1}{Z} \cdot H \quad (10.16)$$

$$C_Z \propto Z \cdot M^{H_Z} = Z \cdot M^{\frac{H}{Z}} \quad (10.17)$$



Slika 10.2: Primer izseka problemskega drevesa v primeru zadrževanja vrednosti vektorja vhodov skozi $Z = 3$ časovne korake za $M = 3$ mogoče različne vrednosti vhodnega vektorja.

Z zadrževanjem vrednosti vektorja vhodov skozi Z časovnih korakov lahko torej zmanjšamo računsko zahtevnost problema vodenja glede na (10.12). Oceno stopnje zmanjšanja računske kompleksnosti podaja enačba (10.18).

$$\frac{C_Z}{C_M} \approx \frac{Z \cdot M^{H_Z}}{M^H} = \frac{Z}{M^{\frac{H \cdot (Z-1)}{Z}}} = \frac{Z}{M^{((Z-1) \cdot H_Z)}} \quad (10.18)$$

Z manjšanjem potrebnega največjega predikcijskega horizonta H se zmanjša tudi računsko zahtevnost problema prediktivnega vodenja. Dodatna obremenitev je sicer računanje $Z - 1$ vmesnih stanj sistema pri predikciji, ki ima za posledico linearno naraščanje računske zahtevnosti, kar pa je v splošnem obvladljiv problem.

Poleg tega smo zaradi zmanjšanja največjega predikcijskega horizonta H na H_Z računsko zahtevnost eksponentno zmanjšali, to pa pomeni bistven prihranek pri računski zahtevnosti.

Z zadrževanjem vrednosti vektorja vhodov skozi Z časovnih korakov lahko dosežemo več učinkov:

- zmanjšamo potrebni največji predikcijski horizont H in s tem računsko zahtevnost algoritma za prediktivno vodenje;
- omogočimo pri določenem času vzorčenja t_S večji največji predikcijski čas t_{pred} ;
- omogočimo pri izbranem največjem predikcijskem času t_{pred} manjši čas vzorčenja t_S ;
- razbremenimo aktuatorje, ki bi bili s spreminjanjem vhodnega signala v vsakem časovnem koraku preobremenjeni.

Zadrževanje vrednosti vektorja vhodov skozi Z časovnih korakov torej po eni strani lahko omogoči vodenje tudi v primerih, ko imamo težave s preveliko računsko zahtevnostjo problema vodenja, po drugi strani pa lahko razbremeni sistem oz. aktuatorje morebitne prevelike frekvence spreminjanja vrednosti vektorja vhoda.

Zaključimo lahko, da je pristop z zadrževanjem vrednosti vektorja vhodov skozi Z časovnih korakov primeren predvsem za sisteme, pri katerih ne potrebujemo ali ne želimo sprememb vhodnega signala v vsakem časovnem koraku k , npr. za toge sisteme. Nastavitev parametra Z je odvisna od sistema in zahtev pri vodenju: večji Z poveča prej naštete učinke in lahko izboljša kakovost vodenja, če le ni frekvenca sprememb vhodnega signala preveč omejena. Ustrezno vrednost parametra Z določimo s poskušanjem, pri čemer si lahko pomagamo s simulacijami.

10.5 Primer uporabe algoritma za vodenje temperature v šaržnem reaktorju

Algoritem za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi na osnovi analize dosegljivosti smo preizkusili na primeru vodenja šaržnega reaktorja, ki je opisan v 3. poglavju.

10.5.1 Matematični model šaržnega reaktorja

Matematični model šaržnega reaktorja je podan v enačbah v razdelku 3.2. Dinamiko v jedru šaržnega reaktorja opisuje enačba (3.1), dinamiko v plašču pa enačba (3.2).

Masa komponent v jedru šaržnega reaktorja je $m = 600$ kg, masa vode v plašču je $m_w = 40$ kg. Specifična toplota komponent v jedru $c_w = 4200$ J kg⁻¹ K⁻¹, specifična toplota vode v plašču je $c_w = 4200$ J kg⁻¹ K⁻¹. Pretok vode v plašču znaša $\phi = 1,6$ kg s⁻¹, prevodnost toplotnega stika med jedrom in plaščem znaša $\lambda = 420$ W m⁻² K⁻¹, prevodnost toplotnega stika med plaščem in okolico znaša $\lambda_0 = 84$ W m⁻² K⁻¹, površina toplotnega stika med jedrom in plaščem znaša $S = 2$ m², površina toplotnega stika med plaščem in okolico znaša $S_0 = 2$ m². Temperatura sveže vhodne vode je lahko $T_C = 12$ °C ali $T_H = 75$ °C.

10.5.2 Omejitev nabora mogočih kombinacij vhodnih vektorjev

Kot smo opisali v razdelku 10.4.2, je v splošnem smiselno že na začetku načrtovanja vodenja omejiti nabor mogočih kombinacij vrednosti vektorja vhoda, kar povzroči zmanjšanje računske zahtevnosti problema vodenja. Omejitev izhaja iz fizičnih in tehnoloških zahtev in pogojev v sistemu, ki ga želimo voditi, saj na ta način izločimo vse nemogoče ali nesmiselne kombinacije vhodnih vektorjev.

V tem smislu lahko določimo matriko vhodnih vektorjev P , ki jo podaja izraz (10.19).

$$P = \begin{bmatrix} 0 & 0,01 & 0,02 & 0,05 & 0,1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (10.19)$$

V izrazu (10.19) označuje vsak stolpec enega izmed mogočih vhodnih vektorjev. Vrstice v matriki vhodnih vektorjev P imajo naslednji pomen:

- prva vrstica označuje položaj mešalnega ventila
 $k_M \in \{0, 0.01, 0.02, 0.05, 0.1, 1\}$;
- druga vrstica označuje položaj zapornega ventila za hladno vodo
 $k_C \in \{0, 1\}$;
- tretja vrstica označuje položaj zapornega ventila za vročo vodo
 $k_H \in \{1, 0\}$.

10.5.3 Vodenje

Algoritem za prediktivno vodenje, ki ga preizkušamo v tem eksperimentu, za delovanje potrebuje primeren matematični model šaržnega reaktorja. V našem primeru smo izbrali hibridni mehki model, katerega razvoj je natančno opisan v razdelku 7.4.1.

Pri vodenju temperature v jedru šaržnega reaktorja smo izbrali največji predikcijski horizont $H_Z = 4$. Število časovnih korakov, skozi katere algoritem zadržuje vrednosti vektorja vhodov, smo nastavili na $Z = 15$ – glej razdelek 10.4.3. Na ta način algoritem upošteva predikcijo obnašanja sistema za $H = H_Z \cdot Z = 4 \cdot 15 = 60$ korakov oz. $t_{pred} = H \cdot t_S = 60 \cdot 10 \text{ s} = 600 \text{ s}$ naprej.

10.5.4 Kriterijska funkcija

Kot smo opisali v razdelku 10.3, ima ustrezna določitev kriterijske funkcije velik vpliv na kakovost vodenja, saj neposredno določa obnašanje zaprtozančnega sistema, poleg tega pa je povezana tudi z velikostjo do konca razvitega problemskega drevesa, s tem pa posredno vpliva tudi na računsko zahtevnost algoritma za prediktivno vodenje.

V našem primeru mora kriterijska funkcija, ki jo uporabljamo v algoritmu, zadoščati pogoju, da njena vrednost v vsakem primeru z višanjem predikcijskega koraka h proti končnemu predikcijskemu horizontu H monotono narašča, kot je zapisano v izrazu (10.3) v razdelku 10.3.1.

Pri vodenju temperature v jedru šaržnega reaktorja smo si pomagali s predlogom oblike ustrezne kriterijske funkcije, ki je opisana v izrazih (10.4), (10.5) in (10.6) v razdelku 10.3.2.

10.5.5 Rezultati – 1. pristop

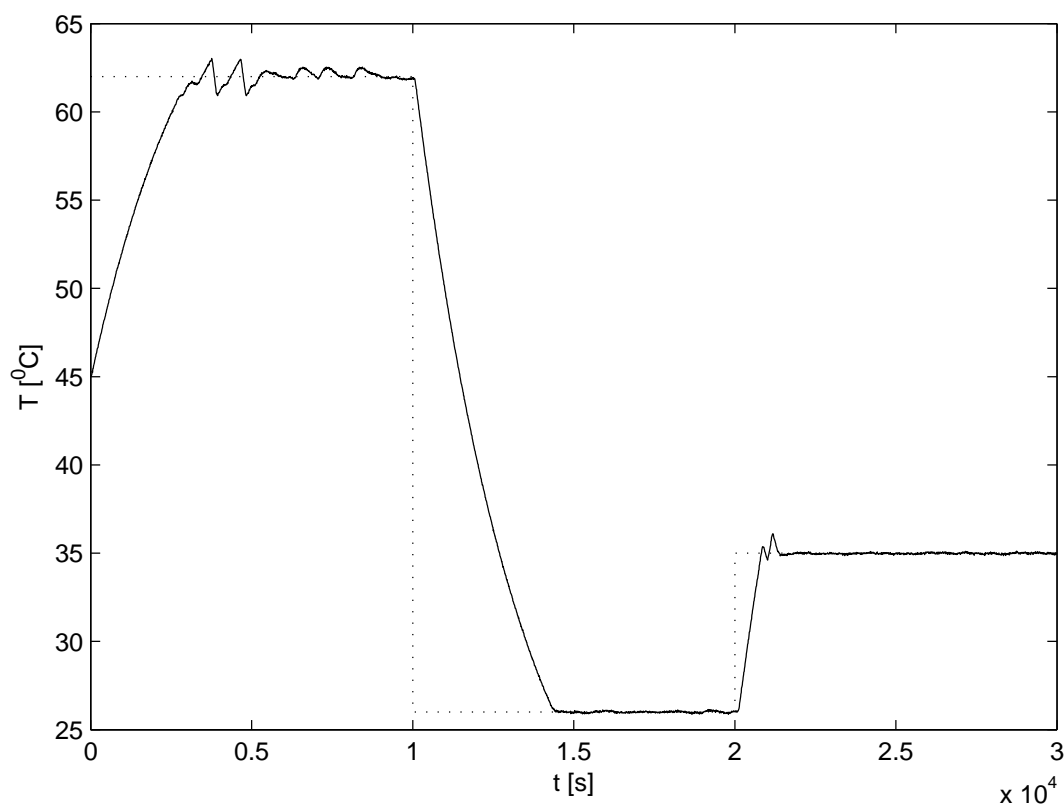
Preizkusili smo vodenje z uporabo preproste kriterijske funkcije, ki jo podajajo enačbe (10.20), (10.21) in (10.22).

$$\begin{aligned}
 J_i &= l(X_{k+1}^{k+h} |_{V_i}, Q_{k+1}^{k+h} |_{V_i}, U_k^{k+h-1} |_{V_i}, k, h |_{V_i}) = \\
 &= l(X_{k+1}^{k+h-1} |_{V_i}, Q_{k+1}^{k+h-1} |_{V_i}, U_k^{k+h-2} |_{V_i}, k, h |_{V_i} - 1) + \\
 &+ \Delta l(x(k+h | k) |_{V_i}, q(k+h | k) |_{V_i}, u(k+h-1 | k) |_{V_i}, k, h |_{V_i})
 \end{aligned} \tag{10.20}$$

za vsak V_i in $h = 1, 2, \dots, H$

$$\begin{aligned}
 \Delta l(x(k+h | k) |_{V_i}, q(k+h | k) |_{V_i}, u(k+h-1 | k) |_{V_i}, k, h |_{V_i}) &= \\
 = (T(k+h | k) - T_{ref}(k+h | k))^2
 \end{aligned} \tag{10.21}$$

$$\begin{aligned}
 l((), (), (), k, 0) &= 0 \\
 \text{za } h &= 0
 \end{aligned} \tag{10.22}$$

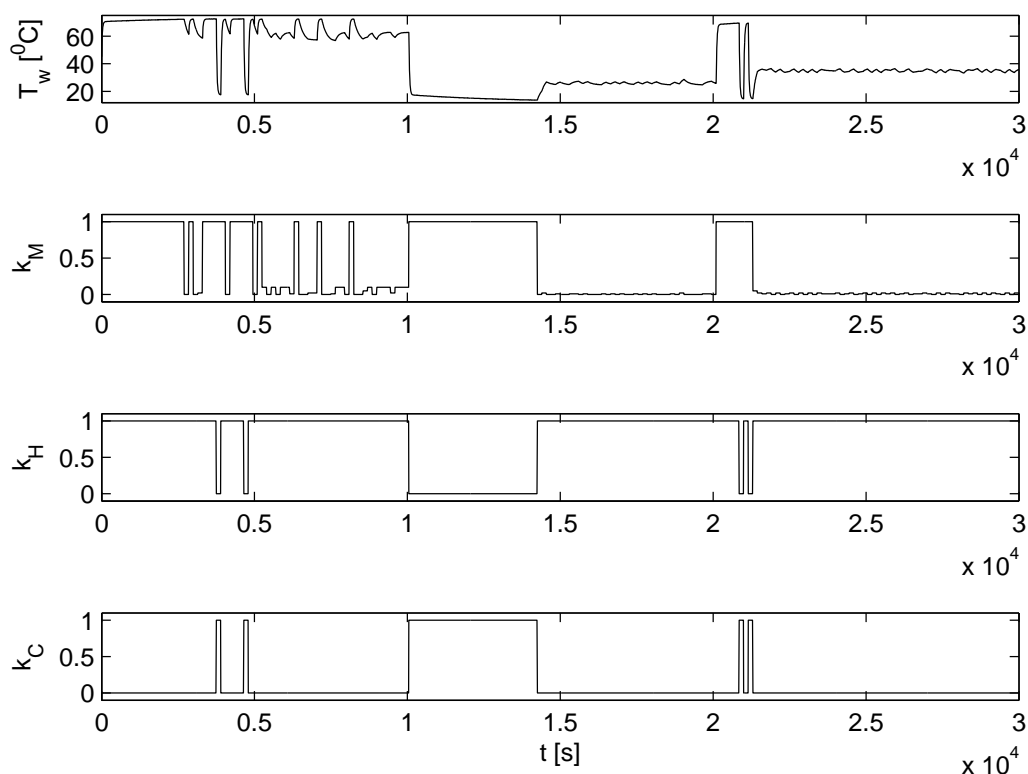


Slika 10.3: Temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta).

Očitno je, da kriterijska funkcija, ki jo podajajo enačbe (10.20), (10.21) in (10.22), zadostuje pogoju (10.5). Opisana kriterijska funkcija upošteva kvadrat odstopanja temperature v jedru T , ki jo določa predikcijski model, od referenčne temperature T_{ref} .

Rezultati eksperimenta so prikazani na naslednjih slikah. Slika 10.3 prikazuje potek temperature v jedru šaržnega reaktorja T in referenčne temperature T_{ref} . Slika 10.4 prikazuje potek temperature v plašču šaržnega reaktorja T_w in položaje ventilov k_M , k_H in k_C .

Ugotovimo lahko, da temperatura v jedru reaktorja T sicer sledi referenčni temperaturi T_{ref} , vendar se kljub temu kažejo trije očitni vidiki vodenja, kjer je potrebna izboljšava.



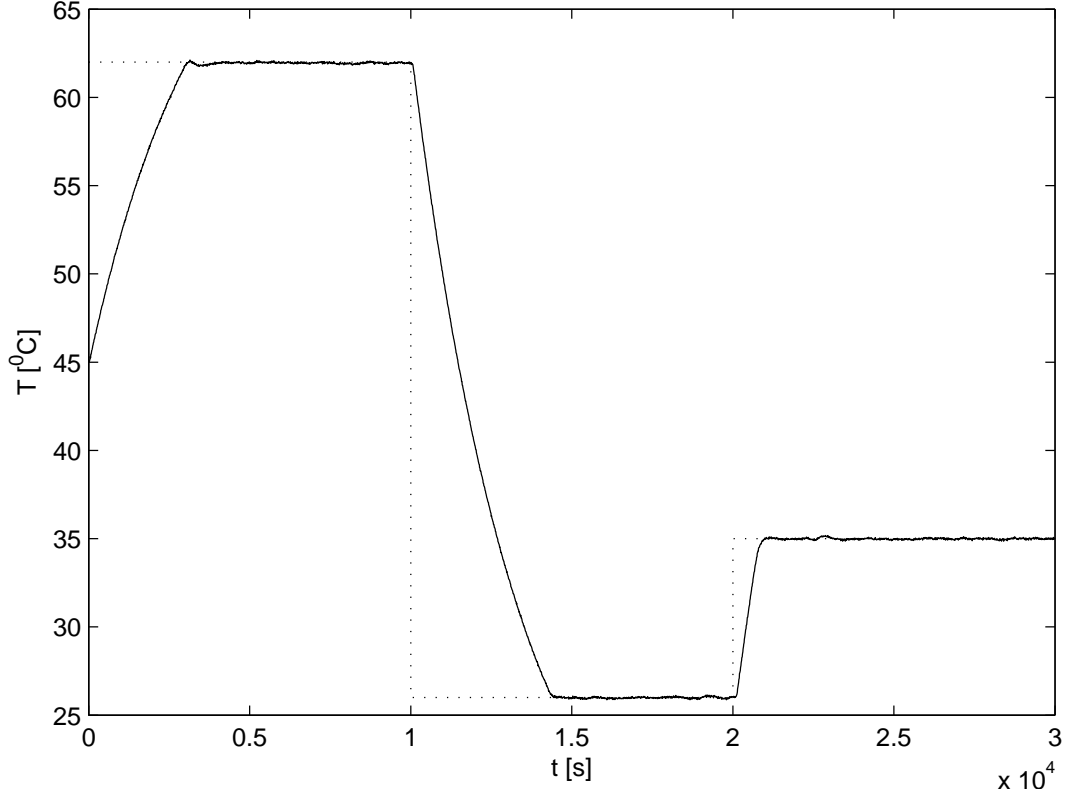
Slika 10.4: Temperatura v plašču reaktorja T_w in položaji ventilov k_M , k_H in k_C .

- Položaji ventilov se pretirano spreminjajo.
- V nekaterih časovnih intervalih lahko opazimo, da se vodenje pretežno izvaja s spreminjanjem položaja zapornih ventilov k_H in k_C , pri čemer ostaja mešalni ventil k_M relativno odprt, kar pri obratovanju v splošnem ni zaželeno. Takšno obnašanje zaprtozadržnega sistema smo sicer pričakovali, saj se v primeru uporabe kriterijske funkcije, ki jo podajajo enačbe (10.20), (10.21) in (10.22), zaporna ventila k_H in k_C obnašata kot še en mešalni ventil, ki pa ima večji⁶ vpliv na temperaturo v jedru od dejanskega mešalnega ventila k_M .
- Poraba sveže vhodne vode je prevelika, kar ima za posledico preveliko porabo energije pri obdelavi šarže. Razlog za to je pretirana odprtost

⁶Razlog za večji vpliv je hitrejša dinamika zaradi večjega temperaturnega razpona.

mešalnega ventila k_M .

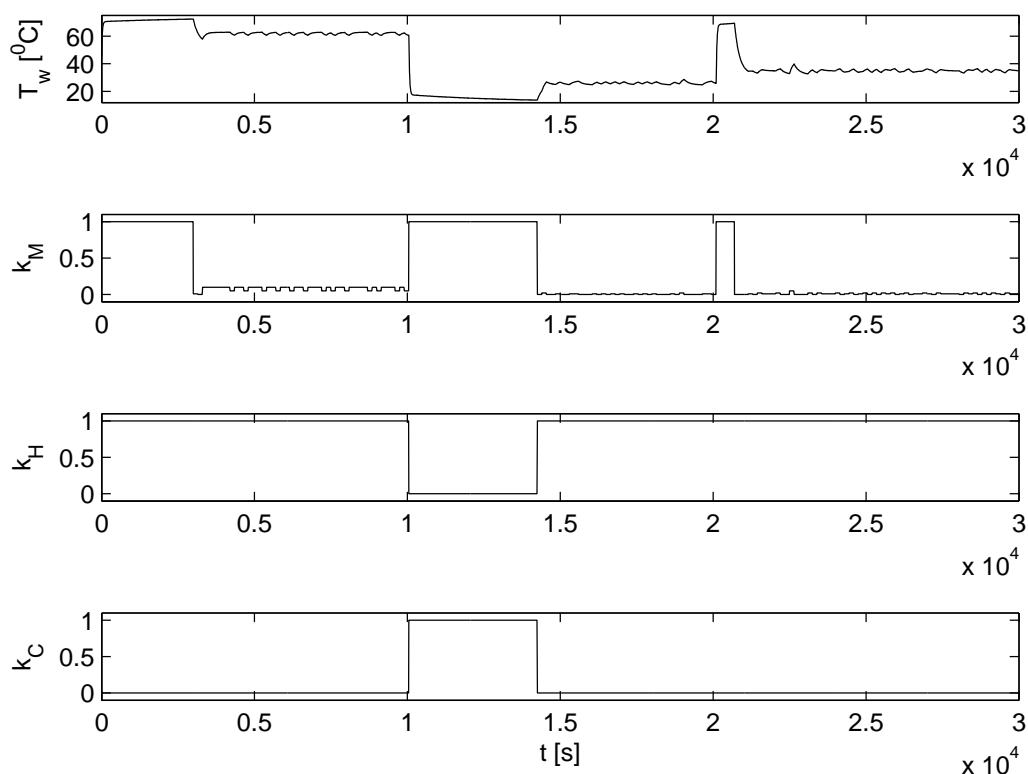
10.5.6 Rezultati – 2. pristop



Slika 10.5: Temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta).

Problematične vidike vodenja, ki so se pojavili pri pristopu v razdelku 10.5.5, smo poskusili izboljšati z vključitvijo členov v kriterijsko funkcijo, ki kaznujejo spreminjanje položaja ventilov k_M , k_H in k_C . Spremenjeno kriterijsko funkcijo podajajo enačbe (10.20), (10.23) in (10.22).

$$\begin{aligned}
 & \Delta l(x(k+h|k) |_{V_i}, q(k+h|k) |_{V_i}, u(k+h-1|k) |_{V_i}, k, h |_{V_i}) = \\
 & = w_1 (T(k+h|k) - T_{ref}(k+h|k))^2 + \\
 & + w_2 (k_C(k+h|k) \cdot k_H(k+h-1|k)) + \\
 & + w_3 |k_M(k+h|k) \cdot k_M(k+h-1|k)| k_H(k+h-1|k)
 \end{aligned} \tag{10.23}$$



Slika 10.6: Temperatura v plašču reaktorja T_w in položaji ventilov k_M , k_H in k_C .

Členi kriterijske funkcije v enačbi (10.23) so uteženi, kot je opisano v nadaljevanju.

- Kvadrat odstopanja temperature v jedru T , ki jo določa predikcijski model, od referenčne temperature T_{ref} , je utežen glede na izbiro parametra Z . V našem primeru je torej $w_1 = \frac{1}{Z}$.
- Utež w_2 utežuje preklop zapornih ventilov k_H in k_C na hladno svežo vhodno vodo. To preprečuje preklop položaja zapornih ventilov k_H in k_C , kadar se referenčni signal T_{ref} ne spremeni stopničasto. Utež w_2 moramo nastaviti na čim večjo vrednost, še vedno pa dovolj majhno, da algoritem dopusti preklop položaja zapornih ventilov k_H in k_C , ko je to res potrebno. V našem primeru je $w_2 = 15$.
- Utež w_3 preprečuje prekomerno premikanje mešalnega ventila. Vseeno pa

moramo utež w_3 nastaviti na dovolj majhno vrednost, da onemogočimo vodenje samo z zapornimi ventili, kot pri pristopu iz razdelka 10.5.5. V našem primeru je $w_3 = 0,03$.

Rezultati eksperimenta so prikazani na naslednjih slikah. Slika 10.5 prikazuje potek temperature v jedru šaržnega reaktorja T in referenčne temperature T_{ref} . Slika 10.6 prikazuje potek temperature v plašču šaržnega reaktorja T_w in položaje ventilov k_M , k_H in k_C .

Ugotovimo lahko, da temperatura v jedru reaktorja T dobro sledi referenčni temperaturi T_{ref} , poleg tega pa smo tudi ustrezno izboljšali problematične vidike vodenja, ki so se pojavili pri pristopu v razdelku 10.5.5: položaji ventilov se manj spreminjajo; regulacijo izvaja pretežno mešalni ventil, medtem ko zaporna ventila preklapljata samo pri večjih spremembah reference; poraba sveže vhodne vode se je bistveno zmanjšala (za natančnejše podatke glej [78]).

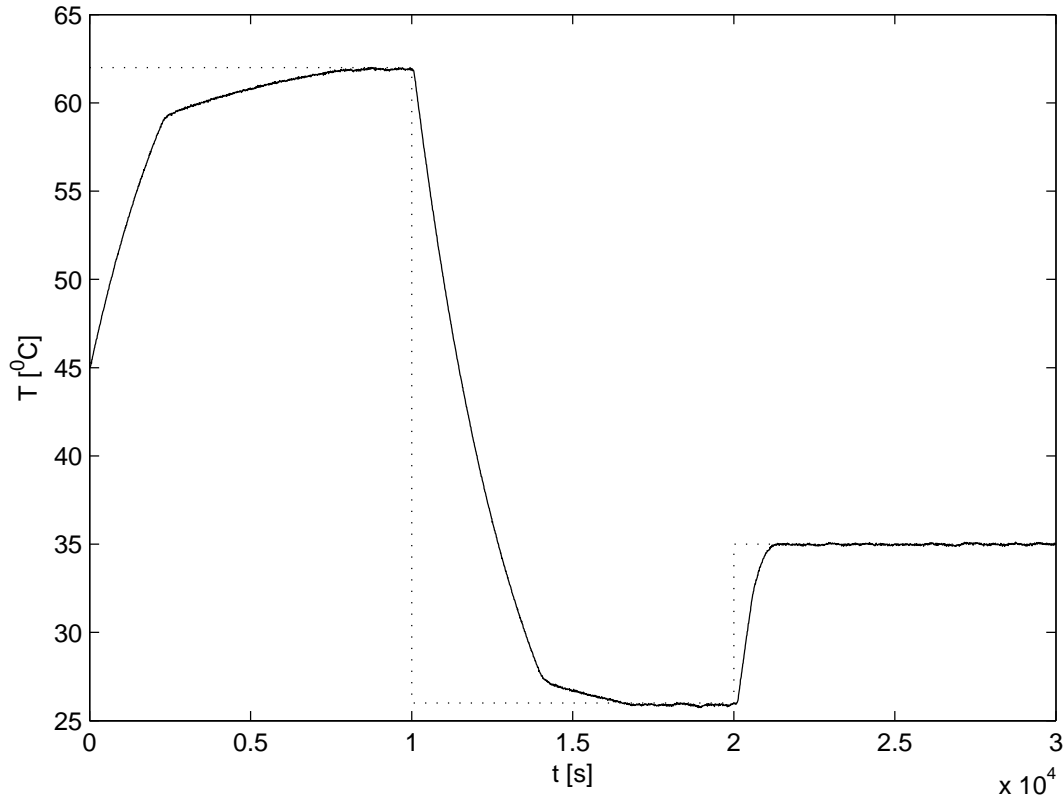
10.5.7 Primerjava prediktivnega vodenja s hibridnim mehkim in hibridnim linearnim modelom

Pristop iz razdelka 10.5.6 smo primerjali s prediktivnim vodenjem s hibridnim linearnim modelom. Hibridni linearni model za temperaturo v plašču šaržnega reaktorja T_w smo izpeljali iz hibridnega mehkega modela – glej enačbo (7.10) v razdelku 7.4.1.

Hibridni mehki model lineariziramo v bližini središča območja delovanja, in sicer tako, da uporabimo fiksen vektor normiranih stopenj pripadnosti, kot kaže enačba (10.24).

$$\beta = [0 \ 0 \ 1 \ 0 \ 0] \quad (10.24)$$

Izhod modela je zapisan v enačbi (10.25), parametri pa v enačbi (10.26).



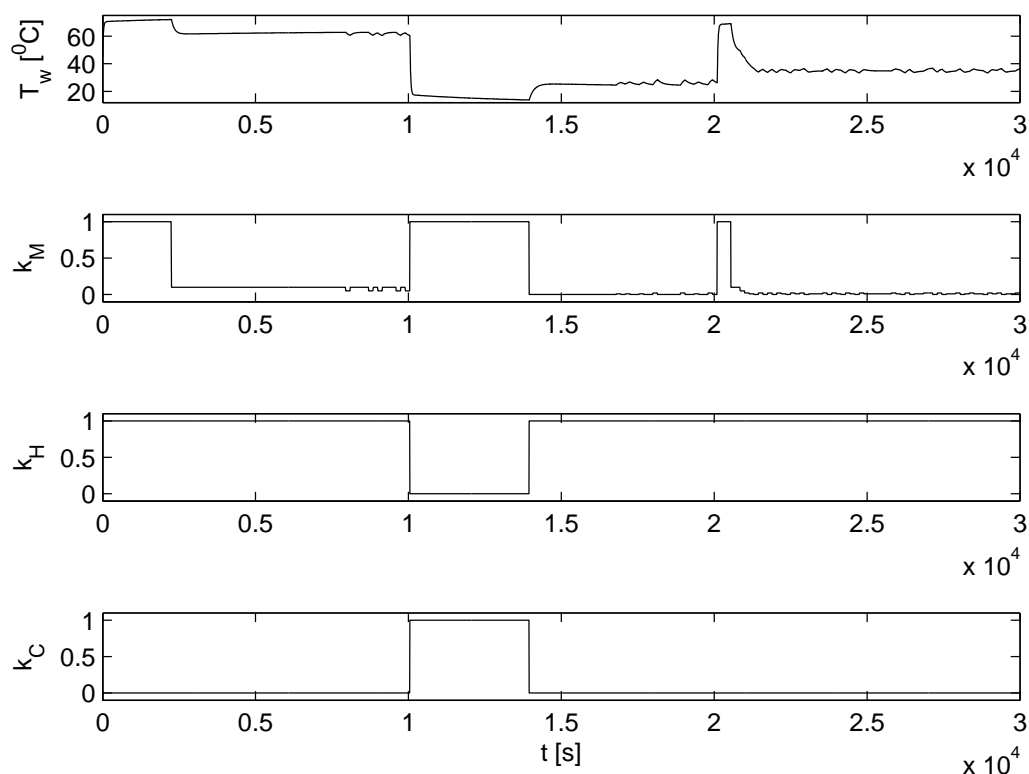
Slika 10.7: Temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta).

$$\hat{T}_w(k+1) = \begin{cases} \Theta_{1,lin}^T [T_w(k) \ T(k) \ k_M(k) \ 1]^T; & k_C = 0 \wedge k_H = 1 \\ \Theta_{2,lin}^T [T_w(k) \ T(k) \ k_M(k) \ 1]^T; & k_C = 1 \wedge k_H = 0 \end{cases} \quad (10.25)$$

$$\begin{aligned} \Theta_{1,lin} &= [0,9429 \ 0,0395 \ 10,5969 \ 0,5273]^T \\ \Theta_{2,lin} &= [0,9322 \ 0,0466 \ -7,8013 \ 0,5650]^T \end{aligned} \quad (10.26)$$

Rezultati eksperimenta so prikazani na naslednjih slikah. Slika 10.7 prikazuje potek temperature v jedru šaržnega reaktorja T in referenčne temperature T_{ref} . Slika 10.8 prikazuje potek temperature v plašču šaržnega reaktorja T_w in položaje ventilov k_M , k_H in k_C .

Razlika med pristopoma v razdelku 10.5.6 in razdelku 10.5.7 je samo v modelu,

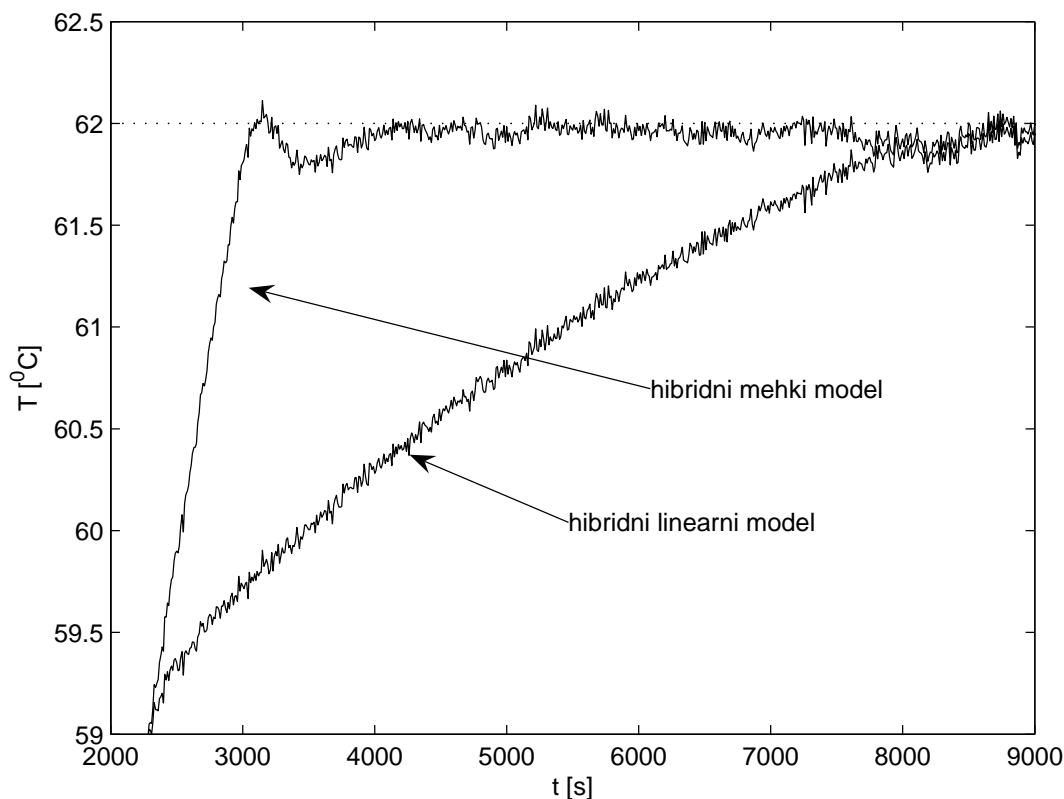


Slika 10.8: Temperatura v plašču reaktorja T_w in položaji ventilov k_M , k_H in k_C .

ki ga uporablja algoritem za prediktivno vodenje. Kriterijska funkcija in vsi ostali parametri vodenja so pri obeh pristopih isti.

Na slikah 10.9, 10.10 in 10.11 so prikazani približani izseki primerjave poteka temperature v jedru šaržnega reaktorja T za pristop, pri katerem je uporabljen hibridni mehki model, in pristop, pri katerem je uporabljen hibridni linearni model.

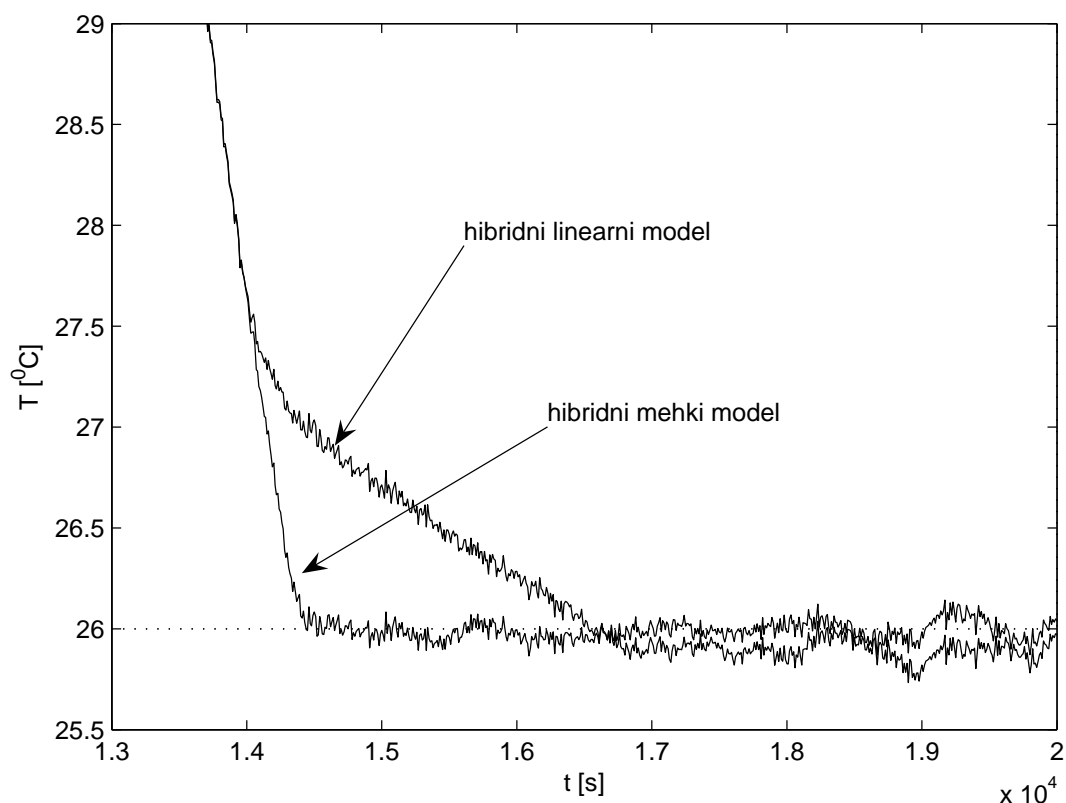
Slika 10.9 prikazuje prvi del poskusa, kjer je referenčna temperatura $T_{ref} = 62$ °C. Ugotovimo lahko, da pri vodenju s hibridnim mehkim modelom prej dosežemo ustaljeno stanje kot pri vodenju s hibridnim linearnim modelom. Kljub hitremu dosegu referenčne vrednosti pa algoritem (v obeh primerih) zagotavlja vodenje brez prevzpona, kar je tudi ena od prednosti pristopov s prediktivnim vodenjem.



Slika 10.9: Primerjava uporabe hibridnega mehkega modela (glej razdelek 10.5.6) in hibridnega linearnega modela (glej razdelek 10.5.7): temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta) – izsek.

Podobno lahko ugotovimo tudi v drugem delu poskusa – na sliki 10.10 – kjer je referenčna temperatura $T_{ref} = 26$ °C.

V tretjem delu poskusa, tj. v primeru stopničaste spremembe referenčne vrednosti na $T_{ref} = 35$ °C – glej sliko 10.11 – predstavlja hibridni linearni model boljši približek dinamike dejanskega sistema, kot pri $T_{ref} = 62$ °C ali $T_{ref} = 26$ °C, zato sta tudi odziva zaprtozančnega sistema pri uporabi vodenja s hibridnim mehkim modelom in vodenja s hibridnim linearnim modelom precej bolj podobna. Kljub temu pa dobimo pri vodenju s hibridnim mehkim modelom tudi v tem delu poskusa boljše rezultate kot pri vodenju s hibridnim linearnim modelom.

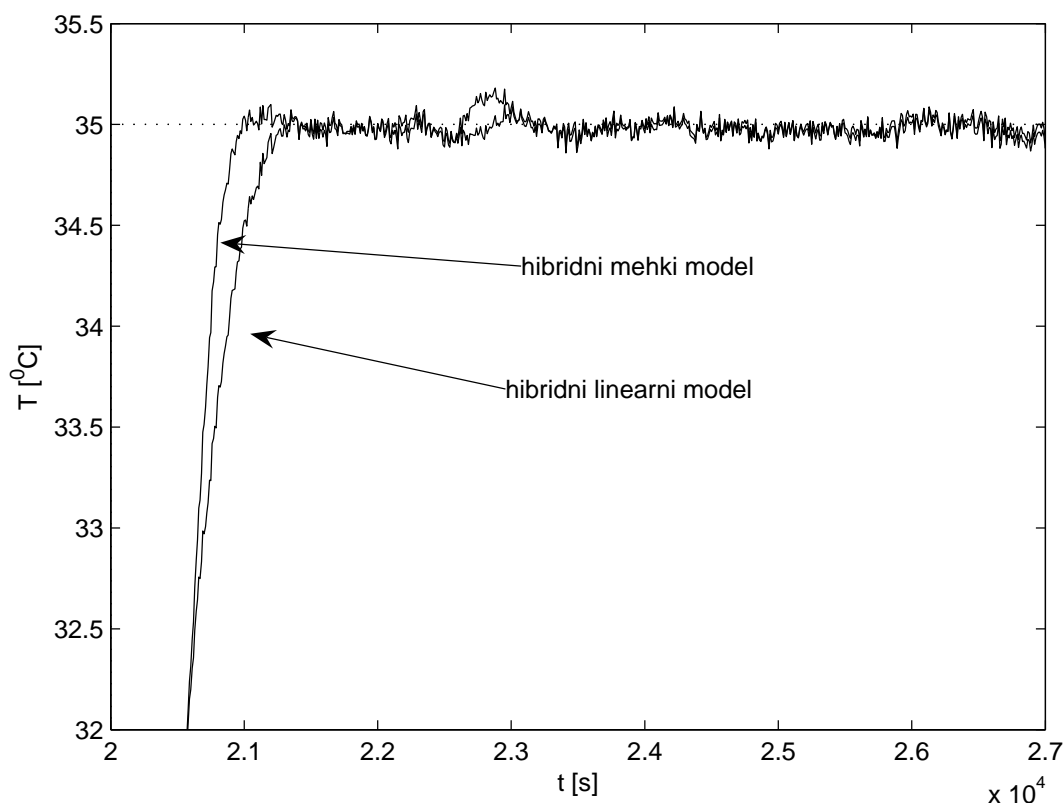


Slika 10.10: Primerjava uporabe hibridnega mehkega modela (glej razdelek 10.5.6) in hibridnega linearnega modela (glej razdelek 10.5.7): temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta) – izsek.

10.5.8 Komentar

Uporabnost algoritma za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi na osnovi analize dosegljivosti smo preverili na primeru vodenja šaržnega reaktorja. Delovanje smo preizkusili s tremi različicami algoritma za prediktivno vodenje.

- V prvi različici smo preizkusili vodenje z uporabo preproste kriterijske funkcije, ki jo podajajo enačbe (10.20), (10.21) in (10.22). Algoritem je za predikcijo uporabljal hibridni mehki model, ki je natančno opisan v razdelku 7.4.1.



Slika 10.11: Primerjava uporabe hibridnega mehkega modela (glej razdelek 10.5.6) in hibridnega linearnega modela (glej razdelek 10.5.7): temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta) – izsek.

Ugotovili smo, da temperatura v jedru reaktorja T sicer sledi referenčni temperaturi T_{ref} , vendar se kljub temu kažejo nekateri vidiki vodenja, kjer so potrebne izboljšave, in sicer predvsem v zvezi s spreminjanjem položaja ventilov k_M , k_H in k_C .

- V drugi različici smo kriterijsko funkcijo razširili s členi, ki kaznujejo spreminjanje položaja ventilov k_M , k_H in k_C , kot kaže enačba (10.23).

Ugotovili smo, da temperatura v jedru reaktorja T dobro sledi referenčni temperaturi T_{ref} , poleg tega pa smo tudi ustrezno izboljšali problematične vidike vodenja, ki so se pojavili pri prvi različici.

- V zadnji različici smo v algoritmu iz drugega pristopa hibridni mehki model zamenjali s hibridnim linearnim modelom.

Ugotovili smo, da pri vodenju s hibridnim mehkim modelom prej dosežemo ustaljeno stanje kot pri vodenju s hibridnim linearnim modelom, predvsem v primerih, ko je delovna točka bolj oddaljena od območja, kjer lahko s hibridnim linearnim modelom še zadovoljivo aproksimiramo dinamiko sistema. Hibridni mehki model nam torej omogoča bolj kakovostno vodenje, kar se močneje odraža predvsem pri sistemih, ki imajo izrazito nelinearno dinamiko.

Rezultati so pokazali, da uporaba algoritma omogoča izpolnitev zahtev vodenja. Temperatura v jedru zadovoljivo sledi referenčnemu signalu, kar zagotavlja kratek čas obdelave posamezne šarže. Z ustrezno določitvijo kriterijske funkcije pa lahko tudi število preklpov zapornih ventilov omejimo na minimum. Kljub hitremu sledenju referenčni vrednosti algoritem zagotavlja vodenje brez prevzpona, kar tudi je ena od prednosti pristopov s prediktivnim vodenjem.

11. Prediktivno vodenje na osnovi genetskega algoritma

V tem poglavju bomo opisali metodo za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi, ki deluje na osnovi genetskega algoritma [96, 47, 101, 86].

11.1 Uporaba genetskega algoritma pri optimizaciji problemov z diskretnimi spremenljivkami

Genetski algoritem predstavlja pristop, ki je primeren za optimizacijo problemov z diskretnimi spremenljivkami. Optimizacijo izvaja nad zaporedjem *genov*, ki sestavljajo *kromosom* posameznega *osebka*.

Geni neposredno opisujejo diskretne optimizacijske spremenljivke:

- vsak *gen* predstavlja enega izmed mogočih vhodnih vektorjev v sistem u_j , kjer je $j \in \{1, 2, \dots, M\}$;
- vsak *osebek* oz. njegov *kromosom* predstavlja eno izmed mogočih rešitev optimizacijskega problema, tj. zaporedje vrednosti vektorja vhodov v sistem U_k^{k+H-1} , kot ga podaja enačba (8.10).

Vsak *kromosom* posameznega *osebka* torej predstavlja eno izmed mogočih rešitev in ga lahko zapišemo z zaporedjem H *genov*. Dolžina kromosoma je odvisna od največjega predikcijskega horizonta H .

Binarno kodiranje spremenljivk pri genetskem algoritmu ni potrebno, saj vsak izmed mogočih vhodnih vektorjev v sistem u_j , kjer je $j \in \{1, 2, \dots, M\}$, določa po en gen.

Genetski algoritem je torej že v zasnovi primeren za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi in predstavlja učinkovit način reševanja nelinearnih mešanih celoštevilskih optimizacijskih problemov z omejitvami [96]. Poleg tega se pristopi k prediktivnemu vodenju na osnovi genetskega algoritma izognejo izračunavanju gradientov kriterijske funkcije, kakršne izvajajo standardni nelinearni optimizacijski algoritmi, kar pomeni bistveno manjšo računsko zahtevnost pristopa.

Genetske algoritme lahko uporabimo pri reševanju nelinearnih mešanih celoštevilskih optimizacijskih problemov z omejitvami. Zaradi kompleksnih lastnosti genetskih operatorjev pa je algoritem problematičen za analizo in v splošnem tudi ne zagotavlja rešljivosti problema. V literaturi lahko najdemo veliko primerov optimizacije z omejitvami, pri katerih se uporabljajo genetski algoritmi [14, 52, 100, 85]. Kljub temu pa še ne obstaja splošna in sistematična rešitev tega problema.

11.2 Mehanizem optimizacije na osnovi genetskega algoritma

Optimizacijo na osnovi genetskega algoritma lahko opišemo z naslednjimi koraki.

1. Inicializacija: določitev naključne *populacije osebkov*, ki ustrezajo možnim rešitvam.
2. Izračun vrednosti *kriterijske funkcije* za vsak osebek v populaciji.
3. Izbor naključnih *staršev* v trenutni populaciji.

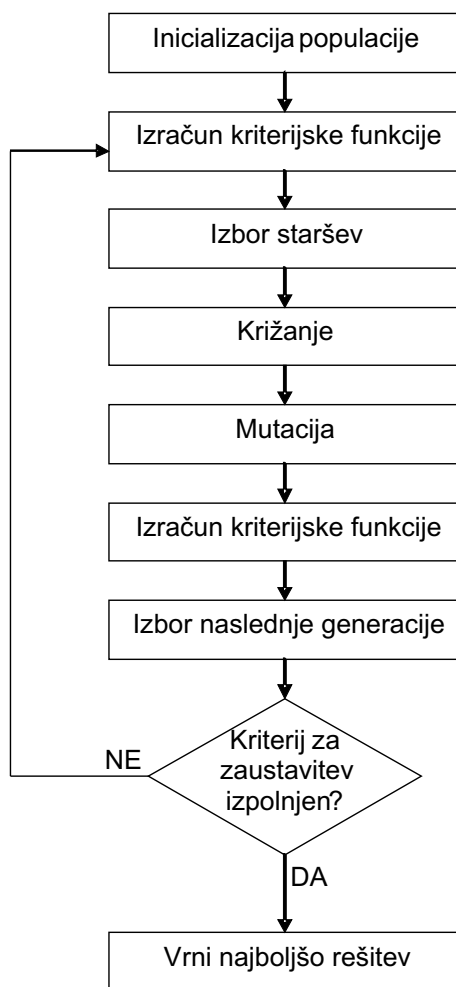
4. Uporaba *genetskih operatorjev*, tj. *križanja* in/ali *mutacije*, na izbranih starših in izpeljava osebkov nove generacije.
5. Izračun vrednosti *kriterijske funkcije* za vsak osebek v novi generaciji.
6. Izbor *najboljših osebkov* (glede na vrednost kriterijske funkcije).
7. Zamenjava *najslabših osebkov* prejšnje generacije z najboljšimi nove generacije, ki so bili izpeljani v 6. koraku.
8. Če kriterijska funkcija doseže določeno *tolerančno mejo* ali če je doseženo *maksimalno število generacij*, se algoritem ustavi, sicer se nadaljuje z 2. korakom.

Potek algoritma je grafično prikazan na sliki 11.1.

Opisani genetski algoritem torej poskuša posnemati naravno evolucijo. Algoritem v vsaki generaciji izbere najboljše in zavrže najslabše osebe glede na vrednost kriterijske funkcije. Na ta način dobimo vedno boljše generacije novih osebkov, dokler ni dosežen kriterij za zaustavitev, tj. dokler kriterijska funkcija ne doseže določene tolerančne meje ali dokler ni doseženo maksimalno število generacij.

11.3 Genetski operatorji

Genetska operatorja, ki ju uporabljamo v algoritmu, sta *križanje* in *mutacija*. Operatorja predstavljata analogijo genetskim procesom, ki se dogajajo v naravi in zagotavljajo raznolikost in razvoj vrst. V algoritmu za prediktivno vodenje torej izvajamo proces, ki je analogen naravni evoluciji, pri čemer za *naravni izbor* skrbi algoritem z izbiranjem najboljših *osebkov* glede na vrednost kriterijske funkcije, ki jo izračuna za vsak obravnavani *osebek*.



Slika 11.1: Potek genetskega algoritma.

11.3.1 Križanje

Križanje je genetski operator, ki deluje nad kromosomoma dveh staršev. Pri križanju se za določeno verjetnostjo izmenjata dela kromosomov križanih osebkov (staršev) in na ta način dobimo kromosoma dveh novih osebkov.

Vzemimo za primer križanje dveh osebkov – O_1 in O_2 – katerih kromosoma podajata izraza (11.1) in (11.2). Denimo, da kromosom v tem primeru sestavlja zaporedje 5 genov.

$$O_1 = [g_1(1) \ g_1(2) \ g_1(3) \ g_1(4) \ g_1(5)] \quad (11.1)$$

$$O_2 = [g_2(1) g_2(2) g_2(3) g_2(4) g_2(5)] \quad (11.2)$$

V izrazih (11.1) in (11.2) označuje $g_1(i)$ gen, ki je v kromosomu osebk O_1 na i -tem mestu, $g_2(j)$ pa gen, ki je v kromosomu osebk O_2 na j -tem mestu.

Če osebk O_1 in O_2 križamo, pri čemer za ta primer predpostavimo, da se križanje zgodi po 3. mestu, dobimo nova osebk – O_3 in O_4 – katerih kromosoma podajata izraza (11.3) in (11.4).

$$O_3 = [g_1(1) g_1(2) g_1(3) g_2(4) g_2(5)] \quad (11.3)$$

$$O_4 = [g_2(1) g_2(2) g_2(3) g_1(4) g_1(5)] \quad (11.4)$$

Dobljena kromosoma osebkov O_3 in O_4 sta sestavljena iz *genskega materiala* svojih staršev O_1 in O_2 , kar je tudi značilnost križanja v naravi.

11.3.2 Mutacija

Mutacija je genetski operator, ki deluje nad kromosomom enega osebk. Pri mutaciji se gen v kromosomu z določeno verjetnostjo zamenja z drugim genom in na ta način dobimo kromosom novega osebk.

Vzemimo za primer mutacijo osebk O_1 , katerega kromosom podaja izraz (11.1).

Če osebk O_1 mutira, pri čemer za ta primer predpostavimo, da se mutacija zgodi na 3. mestu, dobimo mutirani osebk $O_{1,m}$, katerega kromosom podaja izraz (11.5).

$$O_{1,m} = [g_1(1) g_1(2) g_m(3) g_1(4) g_1(5)] \quad (11.5)$$

V izrazu (11.5) označuje $g_m(3)$ mutirani gen, ki je zamenjal gen $g_1(3)$ na 3. mestu v kromosomu osebk O_1 .

Del dobljenega kromosoma mutiranega osebk $O_{1,m}$ je sestavljen iz *genskega materiala* svojega nemutiranega predhodnika O_1 , kar je tudi značilnost mutacije v naravi.

11.4 Suboptimalnost pristopa

Genetski algoritem zagotovi rešitev problema prediktivnega vodenja, ki je blizu optimalni, tj. *suboptimalno* rešitev. Suboptimalni pristopi, kakršno je tudi prediktivno vodenje na osnovi genetskega algoritma, se v praksi izkažejo kot učinkoviti, vendar je njihova analiza težavna, saj je neposredna primerjava z optimalnimi pristopi vprašljiva.

Delovanje algoritma in konkretna rešitev problema sta odvisna od več načrtovalskih parametrov, ki jih je pred zagonom algoritma treba nastaviti. Poleg tega pa algoritem strogo gledano ni ponovljiv, saj vsebuje korake, ki so povezani s (psevdo)naključnimi izbirami oz. dogodki. V našem primeru so taki deli v algoritmu 1., 3., 4. in 6. korak.

Parametri, ki jih je pred uporabo algoritma treba nastaviti, so:

- *število osebkov*;
- *število generacij*;
- *verjetnost križanja*;
- *verjetnost mutacije*;
- *tolerančna meja* kriterijske funkcije za prekinitev algoritma.

Nastavitve parametrov vplivajo tako na kakovost vodenja kot tudi na računsko

zahtevnost algoritma, zato jih je treba karseda skrbno določiti, pri čemer si lahko pomagamo s simulacijami.

11.5 Kriterijska funkcija

Pri izbiri kriterijske funkcije, ki jo uporabimo v genetskem algoritmu, imamo relativno proste roke, saj sam algoritem za delovanje ne zahteva nobene posebne oblike funkcije (8.12).

V praksi se za uporabno izkaže oblika kriterijske funkcije, ki jo podajajo enačbe (11.6), (11.7) in (11.8).

$$J = J_1 + \lambda \cdot J_2 \quad (11.6)$$

$$J_1 = \sum_{h=1}^H (y(k+h | k) - r(k+h))^2 \quad (11.7)$$

$$J_2 = \sum_{h=1}^H (u(k+h | k) - u(k+h-1 | k))^2 \quad (11.8)$$

V enačbi (11.6) označuje λ faktor uteženosti spreminjanja vhodnega signala v kriterijski funkciji.

Enačba (11.7) podaja del kriterijske funkcije, ki kaznuje odstopanje od referenčne trajektorije. Tu $y(k+h | k)$ označuje predikcijo izhoda modela za h korakov naprej, $r(k)$ pa referenčno trajektorijo.

Enačba (11.8) pa podaja del kriterijske funkcije, ki kaznuje spreminjanja vhodnega signala pri predikciji. Tu $u(k+h | k)$ označuje vhodni signal, ki se uporablja za izračun predikcije izhoda modela za h korakov naprej.

11.6 Računska zahtevnost

Računska zahtevnost obravnavanega algoritma za prediktivno vodenje je prvenstveno odvisna od dveh dejavnikov.

- Število generacij, ki vpliva na število izvedb oz. ponovitev genetskega algoritma. Večje število generacij pomeni večjo računsko zahtevnost genetskega algoritma.
- Število osebkov, ki nastopajo v genetskem algoritmu. Več osebkov ima za posledico več operacij v posameznem koraku v genetskem algoritmu in s tem večjo računsko zahtevnost.

Ocena računske zahtevnosti obravnavanega algoritma za prediktivno vodenje je sorazmerna produktu števila generacij N_{G_0} in števila osebkov N_{O_0} . Omeniti je treba, da ocena velja za večja števila N_{G_0} in N_{O_0} , saj imajo poraba časa zaradi inicializacije in ostali *režijski stroški* pri manjših številih večji relativni vpliv. Oceno računske zahtevnosti podaja enačba (11.9).

$$C_0 \propto N_{G_0} \cdot N_{O_0} \quad (11.9)$$

Poleg števila generacij N_{G_0} in števila osebkov N_{O_0} , ki nastopajo v genetskem algoritmu, vplivajo na računsko zahtevnost tudi drugi dejavniki, ki jih bomo našteali v nadaljevanju.

Največji predikcijski horizont H vpliva na dolžino zaporedja genov v kromosomih, nad katerimi delujejo genetski operatorji. Daljši kromosomi pomenijo večjo računsko zahtevnost genetskega algoritma.

Število diskretnih vhodov z ustreznim številom mogočih vrednosti, ki jih lahko zavzame posamezen diskretni vhod, pa vpliva na število mogočih različnih genov, ki jih moramo upoštevati v genetskem algoritmu.

Denimo, da lahko i -ti diskretni vhod, kjer je $i \in \{1, \dots, l\}$, zavzame m_i različnih diskretnih vrednosti. V tem primeru lahko število vseh mogočih kombinacij vrednosti vhodnega vektorja izračunamo, kot kaže enačba (11.10).

$$M_0 = m_1 \cdot \dots \cdot m_l \quad (11.10)$$

Množico mogočih zaporedij vhodnih vektorjev podaja izraz (11.11).

$$U_k^{k+H-1} \in \{u_1, \dots, u_{M_0}\}^H \quad (11.11)$$

V izrazu (11.11) označuje u_j , kjer je $j \in \{1, \dots, M_0\}$, enega od mogočih vhodnih vektorjev, H pa največji predikcijski horizont.

Število mogočih vhodnih vektorjev M_0 torej določa število mogočih različnih genov, največji predikcijski horizont H pa dolžino kromosomov. Število mogočih različnih genov sicer na računsko zahtevnost genetskega algoritma ne vpliva neposredno. Kljub temu pa moramo pri večjem številu mogočih različnih genov in večji dolžini kromosomov v splošnem povečati tudi število osebkov in število generacij, da bi dosegli primerljivo kakovost vodenja.

Opozoriti je treba, da odvisnosti ustreznih stopenj povečanja števila osebkov in števila generacij pri večjem številu mogočih različnih genov in večji dolžini kromosomov, pri čemer želimo ohraniti primerljivo kakovost vodenja, ni mogoče enostavno računsko določiti. Število osebkov in število generacij sta načrtovalska parametra, ki ju določimo s simulacijami.

Povečanje števila osebkov in števila generacij ima, kot smo že zapisali, za posledico večjo računsko zahtevnost genetskega algoritma. Oceno stopnje povečanja računske zahtevnosti podaja izraz (11.12).

$$\frac{C_1}{C_0} \approx \frac{N_{G_1} \cdot N_{O_1}}{N_{G_0} \cdot N_{O_0}} \quad (11.12)$$

V izrazu (11.12) označuje C_1 spremenjeno računsko zahtevnost, N_{G_1} spremenjeno število generacij in N_{O_1} spremenjeno število osebkov v genetskem algoritmu.

Preostali načrtovalski parametri – verjetnost križanja, verjetnost mutacije in tolerančna meja kriterijske funkcije za prekinitev algoritma – prav tako posredno vplivajo na računsko zahtevnost genetskega algoritma.

- Verjetnost križanja in verjetnost mutacije sta povezani s hitrostjo konvergence genetskega algoritma k rešitvi. Vpliv nastavitve verjetnosti križanja in verjetnosti mutacije na učinkovitost genetskega algoritma je tesno povezan s sistemom, ki ga želimo voditi, oz. njegovim modelom, izbiro kriterijske funkcije, referenčno trajektorijo in trenutnim stanjem sistema, zato ga v splošnem ni mogoče enostavno računsko določiti.
- Tolerančna meja kriterijske funkcije za prekinitev algoritma lahko – če je postavljena dovolj visoko – izvajanje genetskega algoritma prekine, še preden je doseženo maksimalno predpisano število generacij. Po drugi strani pa lahko previsoko postavljena tolerančna meja negativno vpliva na kakovost vodenja.

11.6.1 Zmanjšanje računske zahtevnosti z omejitvijo nabora mogočih kombinacij vrednosti vektorja vhoda

Zaradi fizičnih in tehnoloških zahtev in pogojev v sistemu, ki ga želimo voditi, je v splošnem smiselno že na začetku načrtovanja vodenja omejiti nabor mogočih kombinacij vrednosti vektorja vhoda. To povzroči zmanjšanje računske zahtevnosti problema vodenja.

Predpostavimo, da smo nabor mogočih kombinacij vrednosti vektorja vhoda, ki izhajajo iz zahtev za delovanje sistema, zmanjšali z M_0 na M , pri čemer velja $M < M_0$. V tem primeru množico mogočih zaporedij vhodnih vektorjev podaja izraz (11.13).

$$U_k^{k+H-1} \in \{u_1, \dots, u_M\}^H \quad (11.13)$$

V izrazu (11.13) označuje u_j , kjer je $j \in \{1, \dots, M\}$, enega od mogočih vhodnih vektorjev.

Omejitev nabora mogočih kombinacij vrednosti vektorja vhoda ima za posledico omejitev števila mogočih različnih genov, s tem pa tudi nabora vseh možnih osebkov oz. njihovih kromosomov – glej enačbo (11.13).

Število vseh možnih kromosomov se torej z M_0^H zmanjša na M^H . To omogoča zmanjšanje števila osebkov in števila generacij v genetskem algoritmu, in sicer ob ohranitvi primerljive kakovosti vodenja. Zmanjšanje števila osebkov in števila generacij pa ima, kot smo že zapisali, za posledico manjšo računsko zahtevnost genetskega algoritma – glej izraz (11.12).

11.6.2 Zmanjšanje računske zahtevnosti z zadrževanjem vrednosti vektorja vhodov skozi več časovnih korakov

V modelih, ki jih uporabljamo za prediktivno vodenje, je čas vzorčenja t_S vezan na več dejavnikov: dinamiko¹ sistema, ki ga vodimo, zeleno natančnost² predikcije in morebitni čas vzorčenja senzorjev,³ ki jih v sistemu za vodenje uporabljamo.

V splošnem želimo načrtati prediktivno vodenje, ki v vsakem koraku izvaja predikcijo za določen največji predikcijski čas t_{pred} naprej. Z največjim predikcijskim časom pa sta povezana tudi največji predikcijski horizont H in čas vzorčenja

¹Hitrim sistemom ustreza krajši, počasnejšim pa daljši čas vzorčenja t_S .

²Za doseganje ustrezne natančnosti predikcije moramo uporabiti dovolj kratek čas vzorčenja t_S .

³V primeru, da senzorji merjeno količino vzorčijo, je potrebna ustrezna sinhronizacija. Če senzorji podajajo merjeno količino zvezno, nimajo vpliva na čas vzorčenja t_S .

t_S , kot kaže enačba (11.14).

$$H = \frac{t_{pred}}{t_S} \quad (11.14)$$

Čas vzorčenja t_S je določen glede na sistem, ki ga želimo voditi. Dejavniki, ki vplivajo na čas vzorčenja, so dinamika sistema, potrebna natančnost predikcije, morebitni čas vzorčenja senzorjev ipd. – glej tudi razdelka 2.4 in 8.3. Kot vidimo iz enačbe (11.14), lahko ob izbranem času vzorčenja t_S pri večanju največjega predikcijskega časa t_{pred} največji predikcijski horizont H precej naraste. Z naraščanjem največjega predikcijskega horizonta H narašča tudi dolžina zaporedja genov v kromosomih, nad katerimi delujejo genetski operatorji. Daljši kromosomi pa pomenijo večjo računsko zahtevnost genetskega algoritma.

Dolžine kromosomov lahko v mnogo primerih zmanjšamo z zadrževanjem vrednosti vektorja vhodov skozi več časovnih korakov. V veliko sistemih lahko namreč dopustimo (ali celo želimo), da se vrednosti vektorja vhodov ne spreminjajo z vsakim časom vzorčenja t_S . Če je npr. čas vzorčenja t_S relativno kratek, akuatorji pa bi bili s spreminjanjem vhodnega vektorja v vsakem koraku preveč obremenjeni, lahko predpišemo, naj se vektor vhodov spreminja samo vsakih Z časovnih korakov, tj. da zadrži svojo vrednost za čas $Z \cdot t_S$.

Ker v splošnem želimo načrtati prediktivno vodenje, ki v vsakem koraku izvaja predikcijo za izbrani največji predikcijski čas t_{pred} naprej, hkrati pa predpišemo, da naj vektor vhodov zadrži svojo vrednost za čas $Z \cdot t_S$, se z večanjem števila Z manjša potreben največji predikcijski horizont H_Z , kot je zapisano v enačbi (11.15).

$$H_Z = \frac{t_{pred}}{t_S} \cdot \frac{1}{Z} \quad (11.15)$$

Če predpostavimo, da sta izbrani največji predikcijski čas t_{pred} in čas vzorčenja t_S fiksna – torej velja enačba (11.16) – lahko množico mogočih zaporedij vhodnih vektorjev zapišemo v izrazu (11.17).

$$H_Z = \frac{1}{Z} \cdot H \quad (11.16)$$

$$U_k^{k+H-1} \in \{u_1, \dots, u_M\}^{H_Z} = \{u_1, \dots, u_M\}^{\frac{H}{Z}} \quad (11.17)$$

V izrazu (11.17) označuje u_j , kjer je $j \in \{1, \dots, M\}$, enega od mogočih vhodnih vektorjev.

Omejitev nabora mogočih kombinacij vrednosti vektorja vhoda ima za posledico omejitev števila mogočih različnih kromosomov – glej enačbo (11.17). Število vseh možnih kromosomov se torej z M^H zmanjša na $M^{H_Z} = M^{\frac{H}{Z}}$. V splošnem to omogoča zmanjšanje števila osebkov in števila generacij v genetskem algoritmu, in sicer ob ohranitvi primerljive kakovosti vodenja. Zmanjšanje števila osebkov in števila generacij pa ima, kot smo že zapisali, za posledico manjšo računsko zahtevnost genetskega algoritma – glej izraz (11.12).

Z zadrževanjem vrednosti vektorja vhodov skozi Z časovnih korakov lahko dosežemo več učinkov:

- zmanjšamo potrebni največji predikcijski horizont H in s tem računsko zahtevnost algoritma za prediktivno vodenje;
- omogočimo pri določenem času vzorčenja t_S večji največji predikcijski čas t_{pred} ;
- omogočimo pri izbranem največjem predikcijskem času t_{pred} manjši čas vzorčenja t_S ;
- razbremenimo aktuatorje, ki bi bili s spreminjanjem vhodnega signala v vsakem časovnem koraku preobremenjeni.

Zadrževanje vrednosti vektorja vhodov skozi Z časovnih korakov torej po eni strani lahko omogoči vodenje tudi v primerih, ko imamo težave s preveliko

računsko zahtevnostjo problema vodenja, po drugi strani pa lahko razbremeni sistem oz. aktuatorje morebitne prevelike frekvence spreminjanja vrednosti vektorja vhoda.

Podobno kot v 10. poglavju lahko zaključimo, da je pristop z zadrževanjem vrednosti vektorja vhodov skozi Z časovnih korakov primeren predvsem za sisteme, pri katerih ne potrebujemo ali ne želimo sprememb vhodnega signala v vsakem časovnem koraku k , npr. za toge sisteme. Nastavitev parametra Z je odvisna od sistema in zahtev pri vodenju: večji Z poveča prej naštete učinke in lahko izboljša kakovost vodenja, če le ni frekvenca sprememb vhodnega signala preveč omejena. Ustrezno vrednosti parametra Z določimo s poskušanjem, pri čemer si lahko pomagamo s simulacijami. Omenimo naj, da za razliko od pristopa v 10. poglavju, pri genetskem algoritmu računaska zahtevnost ni eksponentno odvisna od največjega predikcijskega horizonta H , ampak ta nanjo vpliva posredno, in sicer preko spremenjene dolžine kromosoma, nad katerim delujejo genetski operatorji, in preko spremenjenega števila generacij in spremenjenega števila osebkov v genetskem algoritmu – glej izraz (11.12).

11.7 Primer uporabe algoritma za vodenje temperature v šaržnem reaktorju

Algoritem za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi na osnovi genetskega algoritma smo preizkusili na primeru vodenja šaržnega reaktorja, ki je opisan v 3. poglavju.

11.7.1 Matematični model šaržnega reaktorja

Matematični model šaržnega reaktorja je podan v enačbah v razdelku 3.2. Dinamiko v jedru šaržnega reaktorja opisuje enačba (3.1), dinamiko v plašču pa enačba (3.2).

Masa komponent v jedru šaržnega reaktorja je $m = 600$ kg, masa vode v plašču je $m_w = 40$ kg. Specifična toplota komponent v jedru $c_w = 4200$ J kg⁻¹ K⁻¹, specifična toplota vode v plašču je $c_w = 4200$ J kg⁻¹ K⁻¹. Pretok vode v plašču znaša $\phi = 1,6$ kg s⁻¹, prevodnost toplotnega stika med jedrom in plaščem znaša $\lambda = 420$ W m⁻² K⁻¹, prevodnost toplotnega stika med plaščem in okolico znaša $\lambda_0 = 84$ W m⁻² K⁻¹, površina toplotnega stika med jedrom in plaščem znaša $S = 2$ m², površina toplotnega stika med plaščem in okolico znaša $S_0 = 2$ m². Temperatura sveže vhodne vode je lahko $T_C = 12$ °C ali $T_H = 75$ °C.

Pri vodenju tovrstnih šaržnih reaktorjev lahko predpostavimo dve osnovni vrsti motenj na izhodu:

- motnja, ki se prišteje k temperaturi v jedru šaržnega reaktorja T in
- motnja, ki se prišteje k temperaturi v plašču šaržnega reaktorja T_w .

Zaradi hitrejše dinamike je pri vodenju precej lažja naloga odpraviti motnjo, ki se prišteje k temperaturi v plašču T_w . V eksperimentu zato predpostavimo samo motnjo, ki se prišteje k temperaturi v jedru T : v trenutku $t = 23000$ s torej temperaturi v jedru dodamo stopničasto motnjo, pri čemer velikost stopnice znaša $\Delta T = 3$ °C.

11.7.2 Omejitev nabora mogočih kombinacij vhodnih vektorjev

Kot smo opisali v razdelku 11.6.1, je smiselno že na začetku načrtovanja vodenja omejiti nabor mogočih kombinacij vrednosti vektorja vhoda, kar povzroči zmanjšanje računske zahtevnosti problema vodenja. Omejitev izhaja iz fizičnih in tehnoloških zahtev in pogojev v sistemu, ki ga želimo voditi, saj na ta način izločimo vse nemogoče ali nesmiselne kombinacije vhodnih vektorjev.

V tem smislu lahko določimo matriko vhodnih vektorjev P , ki jo podaja izraz (11.18).

$$P = \begin{bmatrix} 0 & 0,01 & 0,02 & 0,05 & 0,1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (11.18)$$

V izrazu (11.18) označuje vsak stolpec enega izmed mogočih vhodnih vektorjev. Vrstice v matriki vhodnih vektorjev P imajo naslednji pomen:

- prva vrstica označuje položaj mešalnega ventila
 $k_M \in \{0, 0.01, 0.02, 0.05, 0.1, 1\}$;
- druga vrstica označuje položaj zapornega ventila za hladno vodo
 $k_C \in \{0, 1\}$;
- tretja vrstica označuje položaj zapornega ventila za vročo vodo
 $k_H \in \{1, 0\}$.

11.7.3 Vodenje

Algoritem za prediktivno vodenje, ki ga preizkušamo v tem eksperimentu, za delovanje potrebuje primeren matematični model šaržnega reaktorja. V našem primeru smo izbrali hibridni mehki model, katerega razvoj je natančno opisan v razdelku 7.4.1.

Pri vodenju temperature v jedru šaržnega reaktorja smo izvajali eksperimente z različnimi vrednostmi največjega predikcijskega horizonta, in sicer s $H_{Z,1} = 4$, $H_{Z,2} = 5$ in $H_{Z,3} = 6$. Število časovnih korakov, skozi katere algoritem zadržuje vrednosti vektorja vhodov, smo nastavili na $Z = 15$ – glej razdelek 11.6.2. Na ta način algoritem upošteva predikcijo obnašanja sistema za H_1 , H_2 in H_3 korakov oz. $t_{pred,1}$, $t_{pred,2}$ in $t_{pred,3}$ sekund naprej, kot podajata izraza (11.19) oz. (11.20).

$$\begin{aligned}H_1 &= H_{Z,1} \cdot Z = 4 \cdot 15 = 60 \\H_2 &= H_{Z,2} \cdot Z = 5 \cdot 15 = 75 \\H_3 &= H_{Z,3} \cdot Z = 6 \cdot 15 = 90\end{aligned}\tag{11.19}$$

$$\begin{aligned}t_{pred,1} &= H_1 \cdot t_S = 60 \cdot 10 \text{ s} = 600 \text{ s} \\t_{pred,2} &= H_2 \cdot t_S = 75 \cdot 10 \text{ s} = 750 \text{ s} \\t_{pred,3} &= H_3 \cdot t_S = 90 \cdot 10 \text{ s} = 900 \text{ s}\end{aligned}\tag{11.20}$$

Matrika vhodnih vektorjev P v izrazu (11.18) določa tudi nabor genov. Vsak gen predstavlja enega od mogočih vhodnih vektorjev. Izraz (11.21) torej podaja 7 različnih genov, ki jih algoritem za prediktivno vodenje upošteva pri delovanju.

$$\begin{aligned}g_1 &= [0 \ 0 \ 1]^T \\g_2 &= [0,01 \ 0 \ 1]^T \\g_3 &= [0,02 \ 0 \ 1]^T \\g_4 &= [0,05 \ 0 \ 1]^T \\g_5 &= [0,1 \ 0 \ 1]^T \\g_6 &= [1 \ 0 \ 1]^T \\g_7 &= [1 \ 1 \ 0]^T\end{aligned}\tag{11.21}$$

Kot smo že povedali v razdelku 11.1, je vsak osebek v genetskem algoritmu enolično določen s svojim kromosomom, ki ga sestavlja niz genov iz nabora (11.21).

Vsak osebek predstavlja eno izmed mogočih rešitev optimizacijskega problema. Dolžina kromosoma je odvisna od največjega predikcijskega horizonta, ki v našem primeru znaša $H_{Z,1} = 4$, $H_{Z,2} = 5$ oz. $H_{Z,3} = 6$.

Nabor vseh mogočih rešitev problema vodenja šaržnega reaktorja podaja izraz (11.22).

$$\begin{aligned}
U_k^{k+H_i-1} &\equiv U_k^{k+Z \cdot H_{Z,i}-1} \in \\
&\in \{ \{g_1\}^Z, \{g_2\}^Z, \{g_3\}^Z, \{g_4\}^Z, \{g_5\}^Z, \{g_6\}^Z, \{g_7\}^Z \}^{H_{Z,i}}, \quad (11.22) \\
&\text{kjer je } i \in \{1, 2, 3\}.
\end{aligned}$$

Za primer, ko je $H_Z = H_{Z,1} = 4$, imamo v naboru vseh mogočih rešitev problema vodenja šaržnega reaktorja $7^4 = 2401$ osebek.

11.7.4 Kriterijska funkcija

Zaradi primerljivosti algoritmov smo izbrali enako kriterijsko funkcijo kot v pristopu, ki je opisan v razdelku 10.5.6. Kriterijsko funkcijo torej podajajo enačbe (10.20), (10.23) in (10.22).

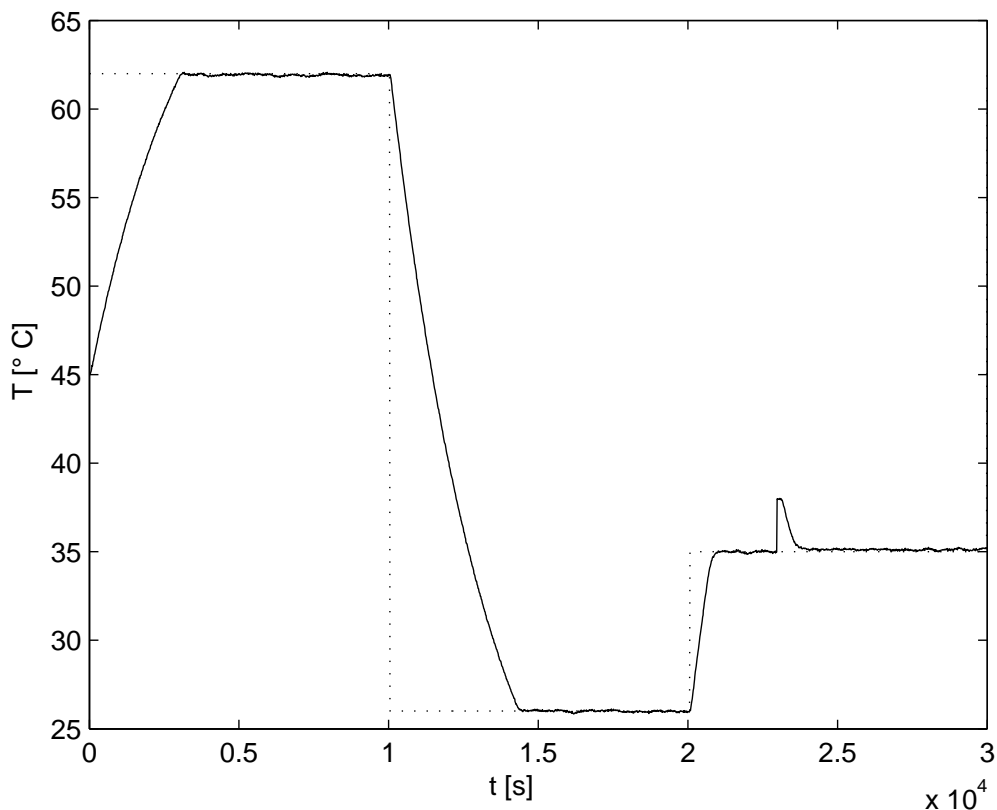
11.7.5 Rezultati

Rezultati eksperimenta za primer vodenja z genetskim algoritmom so prikazani na slikah 11.2 in 11.3. Slika 11.2 prikazuje potek temperature v jedru šaržnega reaktorja T in referenčne temperature T_{ref} . Slika 11.3 prikazuje potek temperature v plašču šaržnega reaktorja T_w in položaje ventilov k_M , k_H in k_C .

V prikazanem primeru je največji predikcijski horizont $H_Z = H_{Z,1} = 4$, število osebkov $N_O = 30$, število generacij $N_G = 10$, verjetnost križanja $p_{kri} = 0,8$, verjetnost mutacije $p_{mut} = 0,001$, tolerančna meja kriterijske funkcije za prekinitev algoritma pa je nastavljena na 0, kar pomeni, da se algoritem izračunava vseh $N_G = 10$ generacij osebkov. Parametri algoritma so izbrani hevristično.

Prediktivno vodenje na osnovi genetskega algoritma v vsakem koraku zagotovi suboptimalno rešitev, ki je lahko bolj ali manj blizu optimalni in je odvisna od parametrov, ki jih moramo pred uporabo algoritma nastaviti – glej razdelek 11.4.

Ugotovimo lahko, da temperatura v jedru reaktorja T dobro sledi referenčni temperaturi T_{ref} , poleg tega pa algoritem tudi ustrezno kompenzira stopničasto

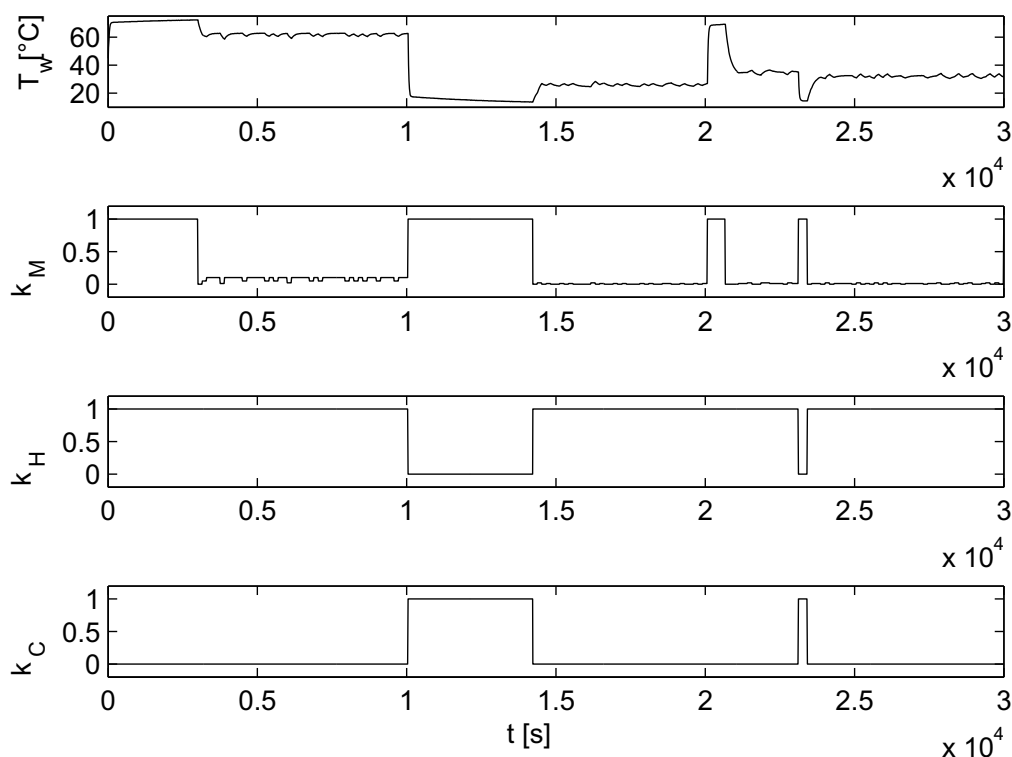


Slika 11.2: Temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta).

motnjo na izhodu, ki se zgodi v trenutku $t = 23000$ s.

Slike 11.4, 11.5 in 11.6 prikazujejo, kako izbira števila osebkov N_O in izbira števila generacij N_G vpliva na čas t_k , ki ga genetski algoritem porabi za izračun rešitve v enem koraku, in kako vpliva na vrednost kriterijske funkcije J , ki določa kakovost vodenja. Na slikah so prikazane srednje vrednosti in standardni odkloni. Slika 11.4 se nanaša na primer, ko je največji predikcijski horizont $H_Z = H_{Z,1} = 4$, slika 11.5 na primer, ko je $H_Z = H_{Z,2} = 5$, slika 11.6 pa na primer, ko je $H_Z = H_{Z,3} = 6$.

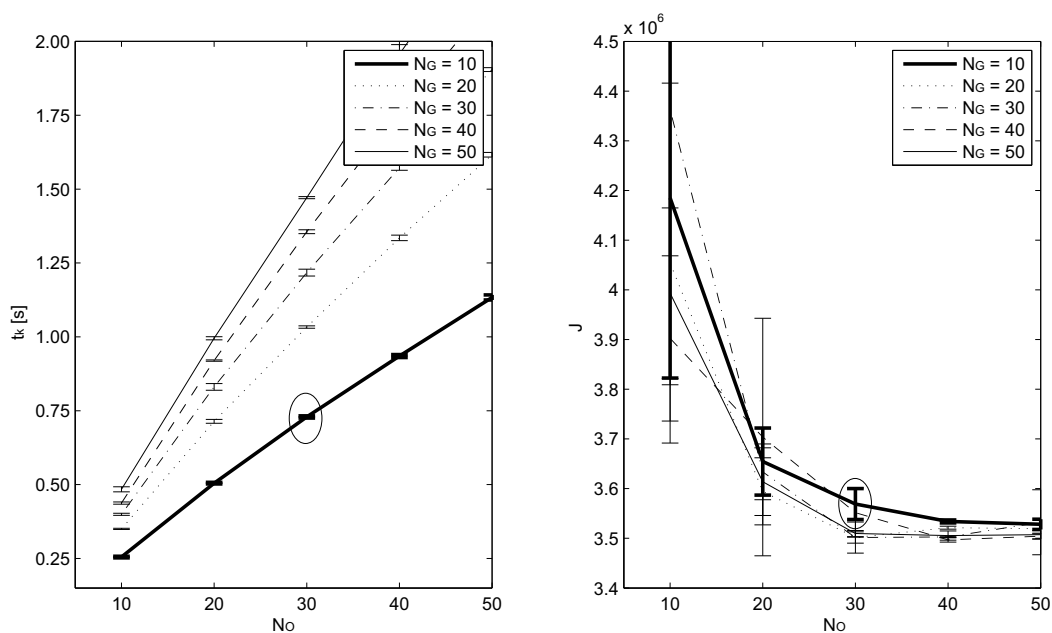
Z rezultati na slikah 11.4, 11.5 in 11.6 si lahko pomagamo pri izbiri parametrov algoritma, tj. števila osebkov N_O in števila generacij N_G . Pri tem iščemo pri-



Slika 11.3: Temperatura v plašču reaktorja T_w in položaji ventilov k_M , k_H in k_C .

merno razmerje med časom t_k , ki ga genetski algoritem porabi za izračun rešitve v enem koraku, in vrednostjo kriterijske funkcije.

- V primeru, ko je $H_Z = H_{Z,1} = 4$, smo izbrali število osebkov $N_O = 30$ in število generacij $N_G = 10$. V tem primeru je znašal povprečni čas, ki ga genetski algoritem porabi za izračun rešitve v enem časovnem koraku, $\bar{t}_k = 0,72$ s.
- V primeru, ko je $H_Z = H_{Z,2} = 5$, smo izbrali število osebkov $N_O = 30$ in število generacij $N_G = 30$. V tem primeru je znašal povprečni čas, ki ga genetski algoritem porabi za izračun rešitve v enem časovnem koraku, $\bar{t}_k = 1,93$ s.
- V primeru, ko je $H_Z = H_{Z,3} = 6$, smo izbrali število osebkov $N_O = 30$



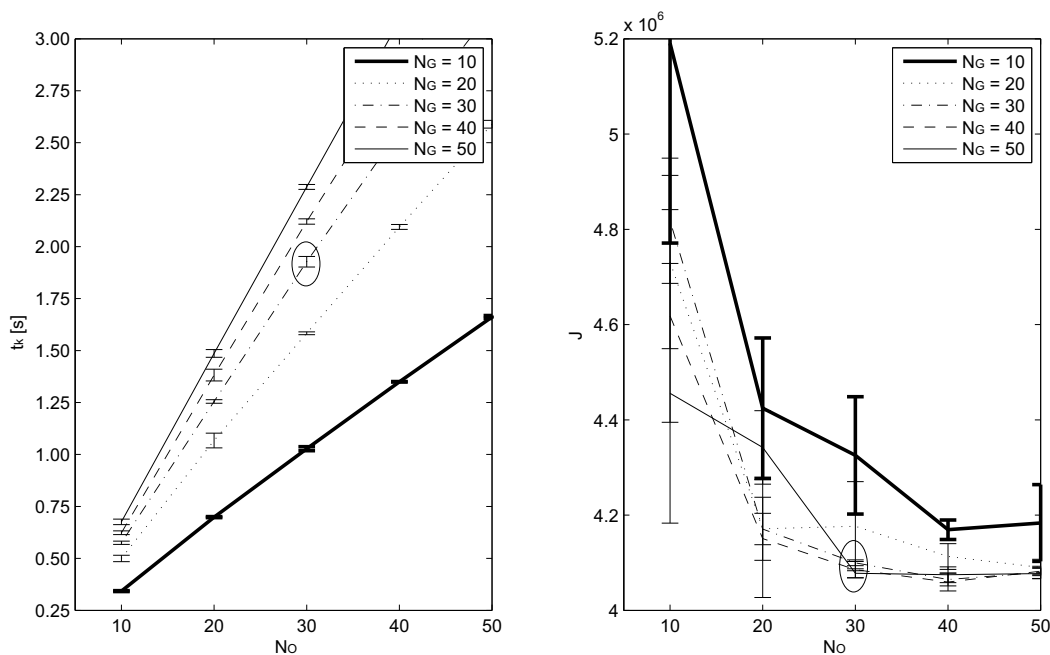
Slika 11.4: Čas t_k , ki ga genetski algoritem porabi za izračun rešitve v enem koraku, in vrednost kriterijske funkcije J za $H_Z = H_{Z,1} = 4$.

in število generacij $N_G = 30$. V tem primeru je znašal povprečni čas, ki ga genetski algoritem porabi za izračun rešitve v enem časovnem koraku, $\bar{t}_k = 6,1$ s.

V vseh treh primerih je čas t_k , ki ga genetski algoritem porabi za izračun rešitve v enem koraku, krajši od časa vzorčenja $t_S = 10$ s, kar pomeni, da lahko algoritem uporabimo za vodenje v realnem času.

11.7.6 Primerjava algoritmov za prediktivno vodenje na osnovi genetskega algoritma, na osnovi analize dosegljivosti in na osnovi ocenjevanja vseh mogočih rešitev

V tem razdelku bomo primerjali rezultate vodenja šaržnega reaktorja z algoritmi za prediktivno vodenje na osnovi genetskega algoritma, na osnovi analize dosegljivosti (glej 10. poglavje) in na osnovi ocenjevanja vseh mogočih rešitev.

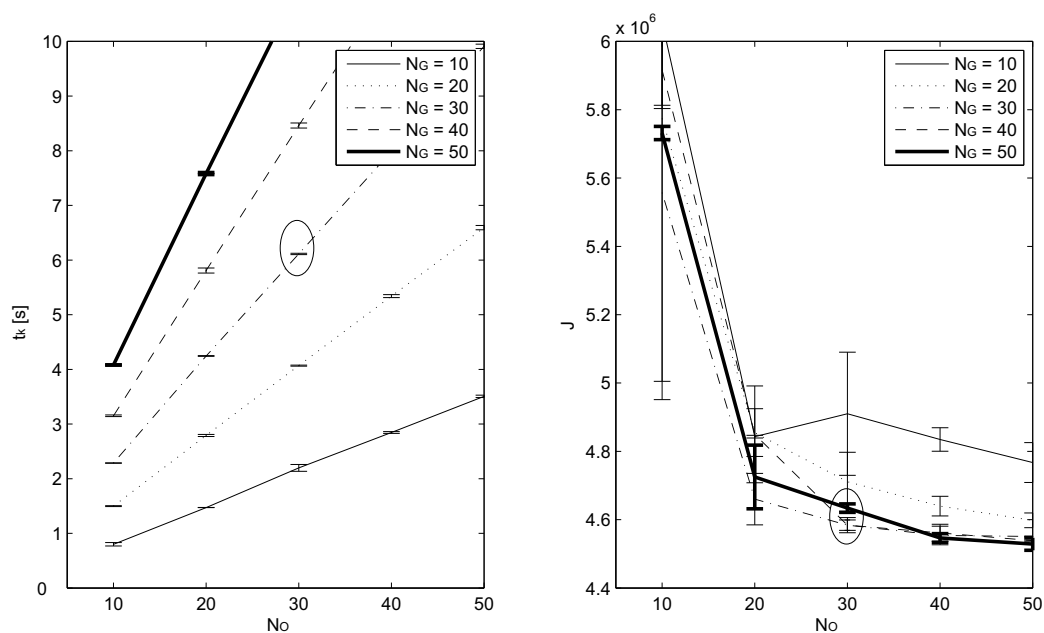


Slika 11.5: Čas t_k , ki ga genetski algoritem porabi za izračun rešitve v enem koraku, in vrednost kriterijske funkcije J za $H_Z = H_{Z,2} = 5$.

Algoritem za prediktivno vodenje na osnovi ocenjevanja vseh mogočih rešitev deluje tako, da izračuna vrednost kriterijske funkcije za vse mogoče rešitve optimizacijskega problema, in sicer v okviru nabora mogočih kombinacij vhodnih vektorjev in največjega predikcijskega horizonta. Na koncu algoritem vrne najustreznejšo rešitev, tj. tisto, ki ima najnižjo vrednost kriterijske funkcije. Mehanižem delovanja lahko, podobno kot pri algoritmu na osnovi analize dosegljivosti, opišemo z razvojem problemskega drevesa, pri čemer pa v vsakem časovnem koraku raziščemo vsa mogoča vozlišča, tj. brez upoštevanja dosegljivosti in določanja neperspektivnih smeri razvoja.

Na slikah 11.7, 11.8 in 11.9 je prikazan normirani čas \check{t}_k , ki ga algoritem porabi za izračun rešitve v vsakem koraku, in sicer za $H_Z = H_{Z,1} = 4$, $H_Z = H_{Z,2} = 5$ in $H_Z = H_{Z,3} = 6$.

Normirani čas \check{t}_k izračunamo tako, da čas t_k , ki ga algoritem porabi za izračun

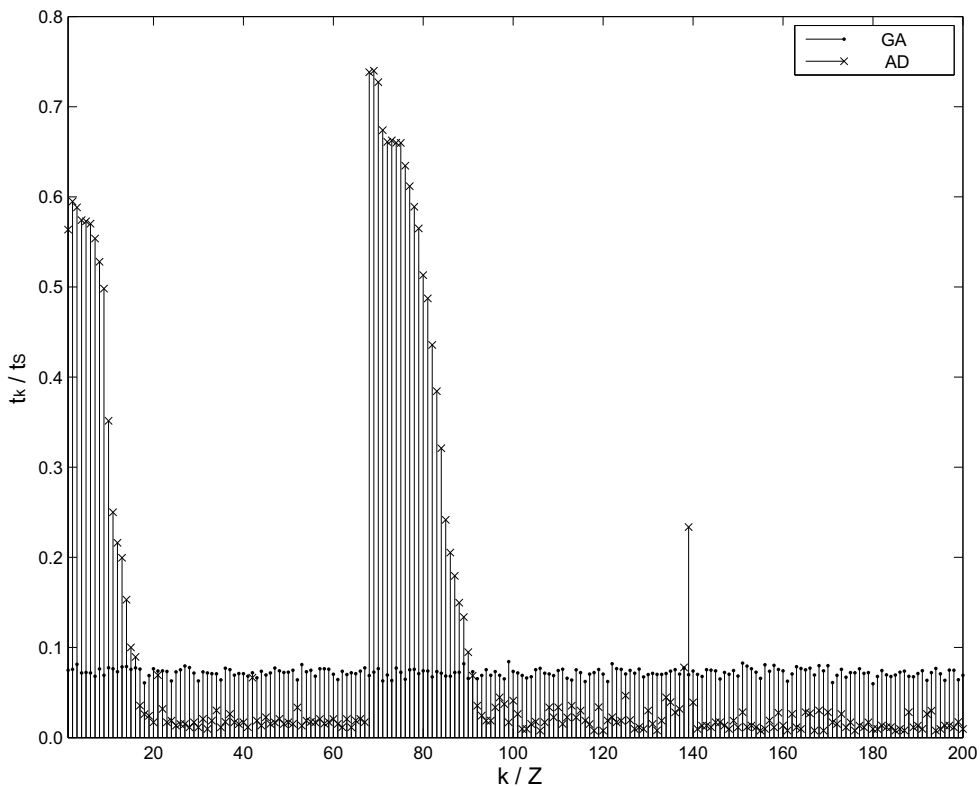


Slika 11.6: Čas t_k , ki ga genetski algoritem porabi za izračun rešitve v enem koraku, in vrednost kriterijske funkcije J za $H_Z = H_{Z,3} = 6$.

rešitve v vsakem koraku, delimo s časom vzorčenja $t_S = 10$ s.

Na slikah 11.7, 11.8 in 11.9 je s pikami označen normirani čas, ki ga porabi algoritem za prediktivno vodenje na osnovi genetskega algoritma (oznaka GA), s križci pa normirani čas, ki ga porabi algoritem za prediktivno vodenje na osnovi analize dosegljivosti (oznaka AD). Normiranega časa, ki ga porabi algoritem za prediktivno vodenje na osnovi ocenjevanja vseh mogočih rešitev, zaradi preglednosti nismo vključili na slike.

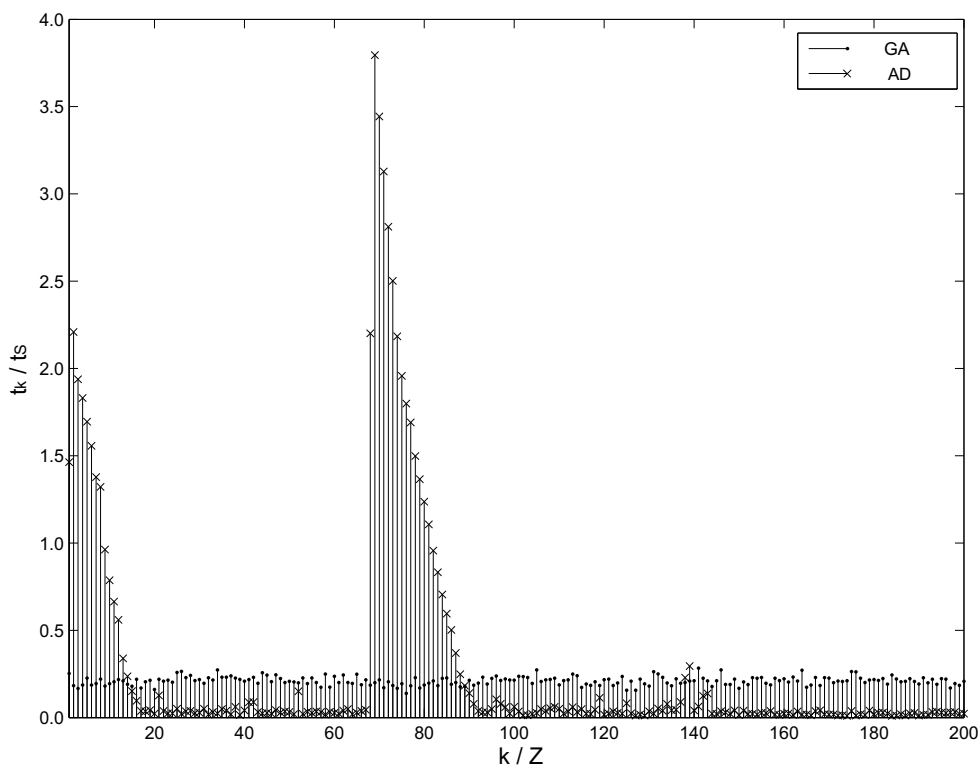
Ugotovimo lahko, da ostaja normirani čas, ki ga algoritem porabi za izračun rešitve v vsakem koraku, v primeru uporabe algoritma za prediktivno vodenje na osnovi genetskega algoritma približno konstanten skozi celoten potek poskusa. Po drugi strani pa se normirani čas, ki ga algoritem porabi za izračun rešitve v vsakem koraku, v primeru uporabe algoritma za prediktivno vodenje na osnovi analize dosegljivosti pri spremembah referenčnega signala bistveno poveča.



Slika 11.7: Normirani čas \check{t}_k , ki ga algoritem porabi za izračun rešitve v vsakem koraku za $H_Z = H_{Z,1} = 4$.

Na slikah 11.7, 11.8 in 11.9 vidimo tudi, da v primerih, ko je $H_Z = H_{Z,2} = 5$ in $H_Z = H_{Z,3} = 6$, normirani čas \check{t}_k , ki ga algoritem na osnovi analize dosegljivosti porabi za izračun rešitve v vsakem koraku, v nekaterih časovnih oknih eksperimenta naraste nad vrednost 1. To pomeni, da je možnost vodenja v realnem času vprašljiva, saj moramo za ustrezno vodenje v splošnem zagotoviti, da je čas izračunavanja rešitve krajši od časa vzorčenja.⁴ Glede na to, da je normirani čas \check{t}_k , ki ga genetski algoritem porabi za izračun rešitve v vsakem koraku, približno konstanten, lahko vidimo, da zanesljivo ostaja pod vrednostjo 1 tudi pri spre-

⁴Tudi v primeru, ko je normirani čas \check{t}_k , ki ga algoritem porabi za izračun rešitve v vsakem koraku, manjši od 1, lahko pride do težav pri vodenju zaradi zakasnitev regulirnega signala. Kljub temu se v analizo teh težav nismo spuščali.



Slika 11.8: Normirani čas \check{t}_k , ki ga algoritem porabi za izračun rešitve v vsakem koraku za $H_Z = H_{Z,2} = 5$.

membah reference, in sicer v vseh obravnavanih primerih – ko je $H_Z = H_{Z,1} = 4$, $H_Z = H_{Z,2} = 5$ in $H_Z = H_{Z,3} = 6$.

Neobčutljivost na spremembe reference ter predvidljivost in omejenost časa za izračun rešitve v vsakem koraku predstavljajo glavno prednost pristopa na osnovi genetskega algoritma. Predvidljivost časa za izračun rešitve olajša težave ali celo sploh omogoči zanesljivo delovanje algoritma za prediktivno vodenje v realnem času.

Tabele 11.1, 11.2 in 11.3 podajajo rezultate vodenja šaržnega reaktorja z algoritmi za prediktivno vodenje na osnovi genetskega algoritma (oznaka GA), na osnovi analize dosegljivosti (oznaka AD) in na osnovi ocenjevanja vseh mogočih rešitev (oznaka OVR). Stolpci v tabelah imajo naslednji pomen:

Tabela 11.1: Primerjava algoritmov za $H_Z = H_{Z,1} = 4$.

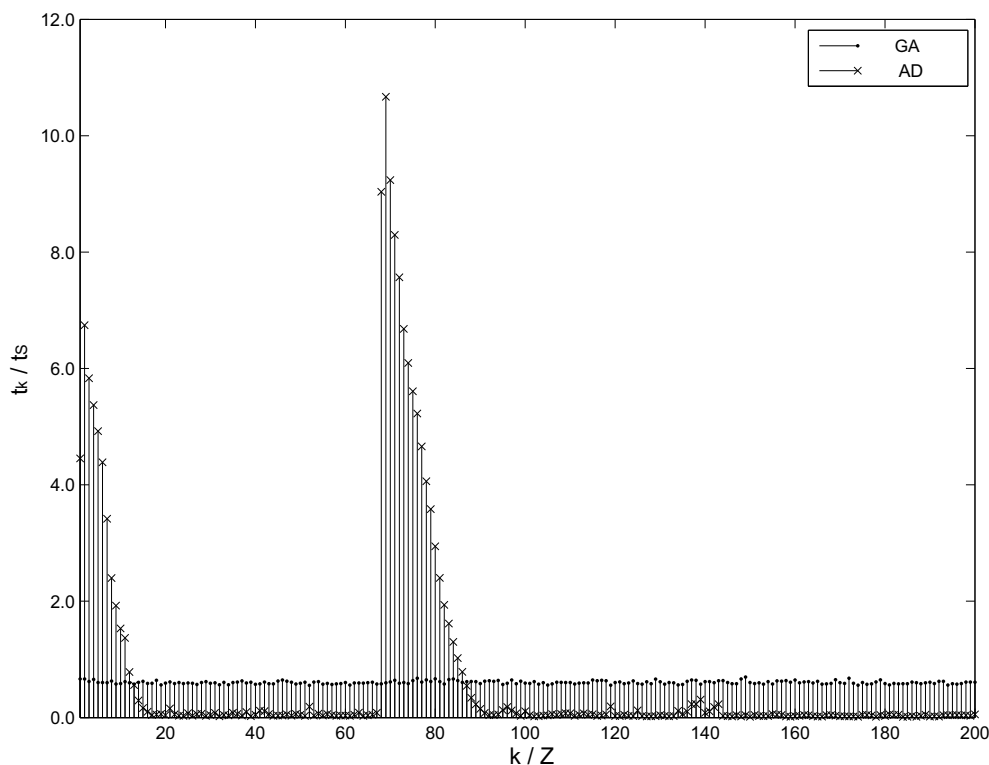
ALGORITEM	\bar{t}_k [s]	$\sigma(t_k)$ [s]	\bar{J}	$\sigma(J)$
GA ($N_O = 30, N_G = 10$)	0,72	0,04	3494511	25342
AD	1,11	0,19	3401524	0
OVR	17,47	0,38	3401524	0

Tabela 11.2: Primerjava algoritmov za $H_Z = H_{Z,2} = 5$.

ALGORITEM	\bar{t}_k [s]	$\sigma(t_k)$ [s]	\bar{J}	$\sigma(J)$
GA ($N_O = 30, N_G = 30$)	1,93	0,09	4065722	6054
AD	2,97	0,52	4035237	0
OVR	158,1	1,24	4035237	0

Tabela 11.3: Primerjava algoritmov za $H_Z = H_{Z,3} = 6$.

ALGORITEM	\bar{t}_k [s]	$\sigma(t_k)$ [s]	\bar{J}	$\sigma(J)$
GA ($N_O = 30, N_G = 30$)	6,10	0,11	4432356	11463
AD	7,35	0,93	4406738	0
OVR	–	–	–	–



Slika 11.9: Normirani čas \check{t}_k , ki ga algoritem porabi za izračun rešitve v vsakem koraku za $H_Z = H_{Z,3} = 6$.

- \bar{t}_k označuje srednjo vrednost časa, ki ga algoritem porabi za izračun rešitve v vsakem koraku;
- $\sigma(t_k)$ označuje standardni odklon časa, ki ga algoritem porabi za izračun rešitve v vsakem koraku;
- \bar{J} označuje srednjo vrednost kriterijske funkcije, ki določa kakovost vodenja;
- $\sigma(J)$ označuje standardni odklon kriterijske funkcije, ki določa kakovost vodenja.

Srednja vrednost časa, ki ga algoritem porabi za izračun rešitve v vsakem koraku, je v vseh primerih najmanjša pri genetskem algoritmu. Pri uporabi algo-

ritma na osnovi analize dosegljivosti se celoten čas za izračun rešitve za primer, ko je $H_Z = H_{Z,1} = 4$, poveča za 54 %, za primer, ko je $H_Z = H_{Z,2} = 5$, se poveča za 54 %, za primer, ko je $H_Z = H_{Z,3,1} = 6$, pa za 20 %. Pri uporabi algoritma na osnovi ocenjevanja vseh mogočih rešitev pa se celoten čas za izračun rešitve za primer, ko je $H_Z = H_{Z,1} = 4$, poveča 16-krat, za primer, ko je $H_Z = H_{Z,2} = 5$, pa kar 82-krat. Primera, ko je $H_Z = H_{Z,3} = 6$ nismo vključili, saj bi celoten eksperiment trajal približno 3 dni, zato rezultati seveda ne bi bili smiselni.

Standardni odklon časa, ki ga algoritem porabi za izračun rešitve v vsakem koraku, vsebuje informacijo o njegovi porazdelitvi. Ugotovimo lahko, da je standardni odklon pri genetskem algoritmu v primerjavi z algoritmom na osnovi analize dosegljivosti precej manjši, kar se vidi tudi iz slik 11.7, 11.8 in 11.9. Tudi pri uporabi algoritma na osnovi ocenjevanja vseh mogočih rešitev je standardni odklon časa relativno majhen, saj je računaska zahtevnost ocenjevanja vseh mogočih rešitev v vsakem koraku približno enaka.

Algoritem na osnovi analize dosegljivosti in algoritem na osnovi ocenjevanja vseh mogočih rešitev v vsakem koraku vrmeta optimalno rešitev optimizacijskega problema, zato je srednja vrednost kriterijske funkcije, ki določa kakovost vodenja, v vseh primerih enaka, njen standardni odklon pa je 0.

Po drugi strani pa je rezultat genetskega algoritma suboptimalna rešitev – glej razdelek 11.4 – zato je tudi srednja vrednost kriterijske funkcije, ki določa kakovost vodenja, višja od vrednosti kriterijske funkcije pri optimalni rešitvi. Poleg tega algoritem strogo gledano ni ponovljiv, saj vsebuje korake, ki so povezani s (psevdo)naključnimi izbirami oz. dogodki, zato lahko izračunamo tudi standardni odklon kriterijske funkcije, ki določa kakovost vodenja. Kljub suboptimalnosti genetskega algoritma pa je rezultat vodenja, tj. zaprtozančni odziv, praktično identičen odzivu pri vodenju z eno izmed optimalnih metod – glej slike 11.2 in 11.3.

11.7.7 Komentar

Uporabnost algoritma za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi na osnovi genetskega algoritma smo preverili na primeru vodenja šaržnega reaktorja. Delovanje genetskega algoritma smo primerjali tudi z algoritmoma na osnovi analize dosegljivosti in na osnovi ocenjevanja vseh mogočih rešitev.

Ugotovili smo, da temperatura v jedru reaktorja T dobro sledi referenčni temperaturi T_{ref} , poleg tega pa genetski algoritem tudi ustrezno kompenzira stopničasto motnjo na izhodu.

Raziskali smo vpliv izbire števila osebkov N_O in izbire števila generacij N_G na čas t_k , ki ga genetski algoritem porabi za izračun rešitve v enem koraku, in na vrednost kriterijske funkcije J , ki določa kakovost vodenja. S povečevanjem števila osebkov N_O in števila generacij N_G lahko izboljšamo kakovost vodenja, tj. zmanjšamo vrednost kriterijske funkcije J , kar pa gre na račun povečanja računske zahtevnosti, tj. časa t_k , ki ga genetski algoritem porabi za izračun rešitve v enem koraku.

V vseh obravnavanih primerih – ko je največji predikcijski horizont $H_Z = H_{Z,1} = 4$, $H_Z = H_{Z,2} = 5$ in $H_Z = H_{Z,3} = 6$ – je čas t_k , ki ga genetski algoritem porabi za izračun rešitve v enem koraku, krajši od časa vzorčenja t_S , kar pomeni, da lahko genetski algoritem uporabimo za vodenje v realnem času.

Glavna slabost genetskega algoritma je suboptimalnost rešitve optimizacijskega problema in s tem negotovi rezultati pri neustrezni nastavitvi parametrov algoritma (števila osebkov, števila generacij, verjetnosti križanja in verjetnosti mutacije). Primerjava genetskega algoritma z algoritmoma na osnovi analize dosegljivosti in na osnovi ocenjevanja vseh mogočih rešitev pa je pokazala, da je rezultat vodenja, tj. zaprtozančni odziv – kljub suboptimalnosti genetskega algoritma – lahko praktično identičen odzivu pri vodenju z eno izmed optimalnih metod.

Srednja vrednost časa, ki ga algoritem porabi za izračun rešitve v vsakem

koraku, je v vseh obravnavanih primerih – ko je največji predikcijski horizont $H_Z = H_{Z,1} = 4$, $H_Z = H_{Z,2} = 5$ in $H_Z = H_{Z,3} = 6$ – najmanjša pri genetskem algoritmu. Poleg tega ostaja čas, ki ga genetski algoritem porabi za izračun rešitve v vsakem koraku, približno konstanten skozi celoten potek poskusa.

Neobčutljivost na spremembe reference ter predvidljivost in omejenost časa za izračun rešitve v vsakem koraku predstavljajo glavno prednost pristopa na osnovi genetskega algoritma. Predvidljivost časa za izračun rešitve pa olajša težave pri uporabi ali celo sploh omogoči zanesljivo delovanje algoritma za prediktivno vodenje v realnem času.

12. Adaptivno prediktivno vodenje z identifikacijo lokalnega linearnega modela

V tem poglavju bomo opisali metodo za adaptivno prediktivno vodenje sistemov s časovno spremenljivimi dinamičnimi lastnostmi, ki deluje na osnovi sprotne identifikacije modela vodenega sistema [81, 1, 118, 132].

12.1 Časovno spremenljive dinamične lastnosti

Pri procesih, ki se pojavljajo v kemičnih, farmacevtskih, biotehnoških in podobnih industrijskih obratih, pogosto vnaprej ne poznamo dinamičnih lastnosti. Te se lahko izkažejo za izrazito kompleksne (hibridna dinamika, nelinearnost), velikokrat pa so dinamične lastnosti tudi časovno spremenljive. Vzrok za časovno spreminjanje dinamike so lahko pojavi, kot so endo- ali eksotermne reakcije ipd. Tovrstni pojavi lahko, če jih pri načrtovanju vodenja ne upoštevamo, pripeljejo do velikih izgub – toplotni pobeg zaradi eksotermne reakcije lahko npr. povzroči popolno uničenje šarže v šaržnem reaktorju – ali celo nevarnost za ljudi in opremo.

Za večino kemičnih reakcij v industriji ne poznamo natančnih kinetičnih modelov[53]. Razvoj natančnega kinetičnega modela reakcije, ki bi ga lahko uporabili pri prediktivnem vodenju, bi namreč zahteval preveč časa in denarja. Poleg tega se reakcije in tehnološki postopki v praksi pogosto spreminjajo, prilagajajo ali izboljšujejo, zato njihovo natančno modeliranje v večini primerov ni smiselno.

Poleg tega se v nekaterih procesih z isto strojno opremo in algoritmom za

vodenje obdelujejo različni reagenti, količine itd., tako da so dinamične lastnosti procesov lahko povsem različne. Klasični pristopi k vodenju, pri katerih predpostavimo časovno nespremenljiv model procesa, so v tovrstnih primerih neustrezni. V praksi so se zato uveljavili različni pristopi z adaptivnimi [49], optimalnimi [48, 92, 54] in prediktivnimi [88, 91] pristopi k vodenju.

12.2 Delovanje algoritma za adaptivno prediktivno vodenje

12.2.1 Zapis lineariziranega modela

Algoritem za adaptivno prediktivno vodenje z identifikacijo lokalnega linearnega modela je primeren za vodenje sistemov, ki jih lahko zapišemo v obliki *časovno spremenljive linearne samoregresijske strukture z zunanjimi vhodi*, kot jo podaja enačba (12.1).

$$y(k) = \theta^T(k)\psi(k) \quad (12.1)$$

Model, zapisan v enačbi (12.1), je sicer v osnovi linearen, je pa tudi časovno spremenljiv, zato lahko načeloma opisuje tudi nelinearno dinamiko sistema, ki ga vodimo, saj model v vsakem časovnem koraku k lineariziramo glede na delovno točko, v kateri se nahaja. Takšen pristop lahko uporabimo pri procesih, kjer se referenčni signal oz. delovna točka ne spreminja, ali pa se spreminja dovolj počasi ali redko, da ne vpliva preveč na oceno dinamičnih lastnosti in s tem na predikcijo izhoda sistema. Omenimo, da je takšen pristop bolj problematičen pri večjih predikcijskih horizontih H oz. časih predikcije t_{pred} , saj v tem primeru nenatančnost predikcije zaradi nelinearnosti pride bolj do izraza.

Denimo, da sistem, ki ga želimo voditi, podaja zapis v obliki časovno spremenljive nelinearne samoregresijske strukture z zunanjimi vhodi, kot ga podaja enačba (12.2).

$$y(k) = F(k, \psi_{nelin}(k)) \quad (12.2)$$

V enačbi (12.2) označuje $\psi_{nelin}(k)$ regresor, ki nastopa v zapisu modela. Argument funkcije F je tudi časovni korak k , kar vpeljuje časovno spremenljivost sistema.

Če je mogoče, je smiselno model v enačbi (12.2) linearizirati z vpeljavo pomožnega vhoda u_{pom} , ki je odvisen od elementov regresorja v originalnem sistemu, in sicer tako, da lahko model zapišemo v obliki, kot jo podaja enačba (12.1).

Preslikavo, ki določa pomožni vhod u_{pom} , podaja enačba (12.3).

$$u_{pom} = f_{pom}(\psi_{nelin}(k)) \quad (12.3)$$

Pri določanju pomožnega vhoda u_{pom} izberemo tako preslikavo f_{pom} , da parametri $\theta(k)$, ki določajo linearizirani model, niso odvisni od delovne točke, ampak samo od dejanskih časovnih sprememb dinamičnih lastnosti sistema. Na ta način lahko obravnavamo pomožni vhod u_{pom} v algoritmu za adaptivno prediktivno vodenje z identifikacijo lokalnega linearnega modela kot regulirno spremenljivko sistema.

12.2.2 Ocenjevanje parametrov

Sprotna identifikacija parametrov procesa je pri adaptivnem vodenju bistvenega pomena. Glede na predhodno poznavanje procesa lahko pri modeliranju določimo strukturo modela in časovno nespremenljive parametre, ki jih poznamo. Parametre procesa, ki so časovno spremenljivi in jih ne poznamo, pa algoritem za adaptivno prediktivno vodenje ocenjuje med delovanjem, tj. v vsakem časovnem koraku k sproti.

Lokalni linearni model

Obravnavani algoritem za ocenjevanje parametrov je osnovan na lokalni linearnizaciji modela procesa, ki se izvaja med samim delovanjem. Model procesa lahko zapišemo kot *časovno spremenljivo linearno samoregresijsko strukturo z zunanjimi vhodi*, kot jo podaja enačba (12.4).

$$y_f(k) = \theta^T(k)\psi_f(k) \quad (12.4)$$

Omenimo naj, da je model, zapisan v enačbi (12.4), sicer v osnovi linearen, je pa tudi časovno spremenljiv, kar omogoča zajem časovno spremenljivih dinamičnih lastnosti procesa.

V enačbi (12.4) označuje $y_f(k)$ filtrirani izhod modela, $\psi_f(k)$ filtrirani regresor, ki je odvisen od predhodno določene strukture modela, $\theta(k)$ pa parametre modela. Parametri modela so časovno spremenljivi in jih določa algoritem med delovanjem.

Filtriranje signalov

V algoritmu za adaptivno prediktivno vodenje uporabljamo model v inkrementalni obliki. Na ta način se izognemo enosmerni komponenti signalov, poleg tega pa lahko namesto afnega modela uporabljamo linearni model.

Filtriranje in diferenciacijo izmerjenih signalov izvedemo s prenosno funkcijo, ki jo podaja v enačba (12.5).

$$H_f(z) = \frac{\Delta(z)}{F(z)} \quad (12.5)$$

V enačbi (12.5) označuje diferencialni operator, kot je zapisan v enačbi (12.6). Imenovalec filtra, ki ga podaja enačba (12.7), določimo eksperimentalno.

$$\Delta(z) = 1 - z^{-1} \quad (12.6)$$

$$F(z) = (1 - k_f z^{-1})^p \quad (12.7)$$

Filtrirani signal torej določimo, kot kaže enačba (12.8).

$$x_f(z) = H_f(z) x(z) \quad (12.8)$$

V enačbi (12.8) označuje $x_f(z)$ Z-transformacijo filtriranega, $x(z)$ pa Z-transformacijo nefiltriranega izmerjenega signala.

Rekurzivno ocenjevanje parametrov z eksponentnim pozabljanjem

Model v enačbi (12.4) je linearen v parametrih, kar nam omogoča analitično izračunavanje parametrov z metodo najmanjše vsote kvadratov odstopanj. Ena glavnih prednosti adaptivnega regulatorja je sposobnost sledenja spremembam v dinamiki procesa. Zaradi tega mora algoritem za ocenjevanje parametrov sproti *pozabljati* stare meritve in izračunavati parametre procesa s poudarkom na novejših meritvah.

V algoritmu uporabimo rekurzivno metodo najmanjše vsote kvadratov odstopanj z eksponentnim pozabljanjem [1]. Parametre procesa algoritem ocenjuje na podlagi enačb (12.9), (12.10) in (12.11).

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k) \left(y_f(k) - \psi_f^T(k) \hat{\theta}(k-1) \right) \quad (12.9)$$

$$K(k) = P(k-1) \psi_f(k) \left(\gamma + \psi_f^T(k) P(k-1) \psi_f(k) \right)^{-1} \quad (12.10)$$

$$P(k) = (I - K(k)\psi_f^T(k)) P(k-1)/\gamma \quad (12.11)$$

V enačbah (12.9), (12.10) in (12.11) označuje $P(k)$ je kovariančno matriko, pri čemer velja $P(k) \in \mathbb{R}^{n \times n}$, kjer je n dolžina regresorja. $\hat{\theta}(k)$ je vektor identificiranih parametrov procesa. γ označuje faktor pozabljanja. I pa označuje enotino matriko, pri čemer velja $I \in \mathbb{R}^{n \times n}$. Rekurzivna identifikacijska algoritma se torej izvaja sproti – v vsakem časovnem koraku k – in vrača parametre procesa $\hat{\theta}(k)$.

Začetne kovariančne matrike morajo biti pozitivno definitne in ustrezno velike. Če metodo za rekurzivno ocenjevanje parametrov interpretiramo v smislu Kalmanovega filtra, lahko ugotovimo, da na ta način pri zagonu algoritma predpostavimo, da imajo parametri začetno porazdelitev s kovarianco $P(0)$ in začetne srednje vrednosti $\hat{\theta}(0)$ [1].

Faktor pozabljanja γ je definiran v enačbi (12.12), kjer označuje t_γ časovno konstanto eksponentnega pozabljanja, t_S pa čas vzorčenja.

$$\gamma = e^{-\frac{t_S}{t_\gamma}} \quad (12.12)$$

Metoda za rekurzivno ocenjevanje parametrov upošteva časovno spremenljivost dinamike procesa, zato uporabljamo pristop z eksponentnim pozabljanjem. Faktor pozabljanja mora biti nastavljen tako, da velja $0 < \gamma \leq 1$. Na ta način se utežitev podatkov za ocenjevanje parametrov spreminja s časom, pri čemer se zadnji podatki utežijo z utežjo 1, podatki, ki so stari k_s časovnih korakov, pa se utežijo z utežjo γ^k .

Eksponentno pozabljanje deluje samo v primeru, ko je proces primerno vzbujan. V praksi pa najdemo veliko procesov, za katere ni primerno, da jih vzbujamo, ko dosežejo ustaljeno stanje. V takem primeru lahko pride do nestabilnosti ocene $\hat{\theta}(k)$ in kovariančne matrike $P(k)$, kar pomeni, da bi se ocene skokoma spremenile, takoj ko regresor ψ_f ne bi bil več nič. Pojav je poznan kot *pobeg algoritma za ocenjevanje parametrov* (angl. *estimator windup*) in poskrbeti moramo, da do

pobega med delovanjem algoritma ne pride.

Da bi se torej pobegu algoritma za ocenjevanje parametrov izognili, v algoritme dodamo pogoj, ki dopusti izračunavanje novih ocen parametrov in kovariančne matrike samo takrat, ko je proces primerno vzbujan. Pogoj podaja izraz (12.13). k_{DZ} označuje pogoj mrtve cone, ki ga določimo hevristično.

$$\psi_f^T(k)P(k-1)\psi_f(k) > k_{DZ}(1-\gamma) \quad (12.13)$$

Uvedba mrtve cone torej izklopi algoritem za ocenjevanje parametrov, če vhodni signal ni primeren. Ocene parametrov modela se torej posodablja le, če je napaka dovolj velika.

12.2.3 Algoritem za prediktivno funkcionalno vodenje

Eden od v praksi najbolj razširjenih pristopov k vodenju je *prediktivno funkcionalno vodenje* [118, 132]. Pristop je tudi del algoritma za adaptivno prediktivno vodenje z identifikacijo lokalnega linearne modela.

Glavna prednost prediktivnega funkcionalnega vodenja je ekspliciten zapis regulirnega zakona. Uporaba prediktivnega funkcionalnega vodenja torej ne zahteva velike računske moči, zato lahko to metodo uporabimo za vodenje širokega spektra sistemov v realnem času, pri čemer lahko algoritem implementiramo na strojni opremi, ki je relativno poceni.

Obnašanje zaprtozančnega sistema

Obnašanje zaprtozančnega sistema določa referenčna trajektorija, ki je podana v obliki referenčnega modela. Cilj vodenja je določiti vhodne signale tako, da bo predvideni izhod procesa sledil referenčni trajektoriji. Točko, kjer se križata referenčna in predvidena trajektorija, imenujemo predikcijski horizont H . Izhod procesa računamo pri predpostavki, da so vhodni signali do horizonta konstantni, kot podaja enačba (12.14).

$$u(k | k) = u(k + 1 | k) = \dots = u(k + H - 1 | k) \quad (12.14)$$

Predvideni izhod sistema v predikcijskem koraku $(k + H | k)$ določa enačba (12.15). V našem primeru obravnavamo model sistema, ki je zapisan v prostoru stanj. Omenimo, da lahko model, ki je zapisan kot *linearna samoregresijska struktura z zunanjimi vhodi* in ga obravnavamo v razdelku 12.2.1, vedno pretvorimo v zapis v prostoru stanj.

$$y_m(k + H | k) = C_m(A_m^H x_m(k | k) + (A_m^H - I)(A_m - I)^{-1} B_m u(k | k)) \quad (12.15)$$

V enačbi (12.15) indeks m pomeni, da se količine nanašajo na model sistema, ki ga uporablja algoritem za prediktivno vodenje. $y_m(k + H | k)$ označuje izhod sistema v predikcijskem koraku $k + H$. A_m , B_m in C_m so sistemske matrike, ki določajo model, $I \in \mathbb{R}^{n \times n}$ pa je enotina matrika, ki je enakih dimenzij kot matrika A_m . $u(k | k)$ označuje vhodni vektor, ki vpliva na predikcijo izhoda sistema $y_m(k + H | k)$ in zanj velja izraz (12.14).

Referenčni model

Referenčni model podaja diferenčna enačba (12.16).

$$\begin{aligned} x_r(k + 1) &= a_r x_r(k) + b_r w(k) \\ y_r(k) &= c_r x_r(k) \end{aligned} \quad (12.16)$$

V enačbi (12.16) je referenčni signal označen z w . V našem primeru referenčni model deluje kot filter prvega reda, ki zgladi referenčni signal w .

Parametre referenčnega modela moramo izbrati tako, da zadostijo enačbi (12.17).

$$c_r (1 - a_r)^{-1} b_r = 1 \quad (12.17)$$

Na ta način zagotovimo, da je ojačenje referenčnega modela 1, kar omogoča sledenje referenčnemu signalu brez pogreška v ustaljenem stanju, tj. asimptotično sledenje referenci.

Parametre torej lahko določimo, kot podajata enačbi (12.18) in (12.19).

$$c_r = 1 \quad (12.18)$$

$$b_r = 1 - a_r \quad (12.19)$$

Parameter a_r določa časovno konstanto referenčnega modela. Parameter vpliva na hitrost odziva zaprtozančnega sistema in ga je treba določiti glede na časovne konstante procesa.

Če upoštevamo enačbe (12.16), (12.18) in (12.19), lahko izpeljemo predvideni izhod referenčnega modela v predikcijskem koraku ($k + H | k$), kot ga določa enačba (12.20).

$$y_r(k + H | k) = a_r^H y_r(k | k) + (1 - a_r^H) w(k | k) \quad (12.20)$$

Enačba (12.20) velja, če predpostavimo konstanten in omejen referenčni signal, kot podaja izraz (12.21).

$$w(k) = w(k + 1) = \dots = w(k + H - 1) \quad (12.21)$$

Regulirni zakon

Cilj obravnavanega algoritma za prediktivno funkcionalno vodenje je določitev takega regulirnega zakona, ki bi zagotovil čim boljše sledenje izhoda sistema $y(k)$

referenčnemu signalu $w(k)$.

Osnovna ideja obravnavane metode je, da v vsakem časovnem koraku k določimo tak vhodni signal $u(k)$, ki bo zagotovil sledenje predvidenega izhoda modela $y(k)$ izhodu referenčnega modela $y_r(k)$ v okviru predikcijskega horizonta H , kot podaja enačba (12.22).

$$y(k+i | k) = y_r(k+i | k), \quad (12.22)$$

kjer je $i = 0, 1, \dots, H$.

Najprej zapišemo enačbo (12.20) v obliki, ki jo podaja enačba (12.23).

$$w(k+H | k) - y_r(k+H | k) = a_r^H (w(k | k) - y_r(k | k)) \quad (12.23)$$

Iz enačb (12.22) in (12.23) lahko zapišemo enačbo za sledenje referenci (12.24).

$$y(k+H | k) = w(k+H | k) - a_r^H (w(k | k) - y(k | k)) \quad (12.24)$$

Prediktivno funkcionalno vodenje deluje na predpostavki, da sta želeni inkrement izhoda procesa Δ_y in inkrement izhoda modela Δ_{y_m} , ki ga uporablja algoritem, enaka, kot podaja enačba (12.25).

$$\Delta_y = \Delta_{y_m} \quad (12.25)$$

Želeni inkrement izhodnega vektorja Δ_y je določen v enačbi (12.26), inkrement vektorja modela Δ_{y_m} , ki ga uporablja algoritem, pa v enačbi (12.27).

$$\Delta_y = y(k+H | k) - y(k | k) \quad (12.26)$$

$$\Delta_{y_m} = y_m(k+H | k) - y_m(k | k) \quad (12.27)$$

Če v enačbo (12.26) vstavimo enačbo (12.24), dobimo želeni inkrement izhodnega vektorja Δ_y , kot je določen v enačbi (12.28).

$$\Delta_y = w(k + H | k) - a_r^H (w(k | k) - y(k | k)) - y(k | k) \quad (12.28)$$

Če v enačbo (12.25) vstavimo enačbi (12.28) in (12.27), lahko – ob upoštevanju enačb (12.24) in (12.15) – zapišemo regulirni zakon, ki ga podajata enačbi (12.29) in (12.30).

$$u(k) = \eta^{-1}((1 - a_r^H)(w(k) - y(k)) + y_m(k) - C_m A_m^H x_m(k)), \quad (12.29)$$

$$\eta = C_m (A_m^H - I) (A_m - I)^{-1} B_m \quad (12.30)$$

Pogoj, da v enačbi (12.30) velja $\eta \neq 0$, je izpolnjen, če je proces stabilen, vodljiv in spoznaven.

Regulirni zakon, ki ga podajata enačbi (12.29) in (12.30), ima integrirni značaj. Pogoji za stabilnost so obravnavani v [132], kjer je tudi dokazano, da lahko za stabilne sisteme vedno določimo stabilen regulirni zakon, če je horizont H večji ali enak redu sistema.

Preslikava regulirnega signala v prostor originalnih vhodov sistema

Pri določanju regulirnega zakona v razdelku 12.2.3 smo se hibridni in nelinearni dinamiki izognili z vpeljavo pomožnega vhoda u_{pom} – glej enačbo 12.3. Na ta način smo model sistema, ki ga želimo voditi, lahko zapisali v obliki *časovno spremenljive linearne samoregresijske strukture z zunanjiimi vhodi*, kot jo podaja enačba (12.1). Regulirni zakon v tem primeru vrne izračunano vrednost pomožnega vhoda u_{pom} , ki pa jo moramo še preslikati v prostor originalnih vhodov sistema, kar naredimo v zadnjem koraku algoritma.

Pri določanju preslikave pomožnega vhoda u_{pom} v prostor originalnih vhodov sistema imamo več možnosti, ki so odvisne od lastnosti preslikave f_{pom} v enačbi (12.3). Lastnosti preslikave in ustrezne postopke obravnavamo v nadaljevanju.

- Preslikava f_{pom} v enačbi (12.3) je *bijektivna*.

V tem primeru obstaja inverzna preslikava f_{pom}^{-1} . Ustrezni regresor $\psi_{nelin}(k)$ oz. ustrezne originalne vhode določimo torej z inverzno preslikavo, kot jo podaja enačba (12.31).

$$\psi_{nelin}(k) = f_{pom}^{-1}(u_{pom}) \quad (12.31)$$

Na ta način so originalni vhodi sistema enolično določeni.

- Preslikava f_{pom} v enačbi (12.3) *ni bijektivna*.

V tem primeru imamo pri nadaljnji obravnavi dve možnosti.

- Preslikava f_{pom} v enačbi (12.3) *ni surjektivna*.

To pomeni, da za nekatere vrednosti pomožnega vhoda u_{pom} ne obstaja inverzna preslikava f_{pom}^{-1} .

Do tega lahko pride zaradi lastnosti originalnih vhodov, ki so navzgor in navzdol omejeni ali ne morejo zavzeti nekaterih vrednosti, po drugi strani pa so lahko razlog neustrezne zahteve za vodenje, kot so prehitra dinamika referenčnega modela, prehitre spremembe referenčnega signala ipd.

V tem primeru z uporabo ustrezne nadzorne logike določimo vrednosti originalnih vhodov tako, da je preslikava ustreznega regresorja $f_{pom}(\psi_{nelin}(k))$ čim bolj podobna¹ izračunanemu pomožnemu vhodu u_{pom} .

¹Podobnost algoritem določa glede na predločen kriterij, npr. evklidsko razdaljo.

- Preslikava f_{pom} v enačbi (12.3) *ni injektivna*.

To pomeni, da za nekatere vrednosti pomožnega vhoda u_{pom} obstaja več različnih regresorjev $\psi_{nelin}(k)$, ki se preslikajo v te vrednosti u_{pom} . Predpostavimo, da s $\psi'_{nelin}(k)$ označimo tisti del regresorja $\psi_{nelin}(k)$, ki vsebuje samo originalne vhode v sistem (ne pa tudi stanj oz. izhodov sistema). Če omejimo definicijsko območje preslikave f_{pom} samo na tiste regresorje $\psi_{nelin}(k)$, ki v preostalem delu (ki ni zajet v $\psi'_{nelin}(k)$) vsebujejo samo tista stanja oz. izhode, ki dejansko določajo sistem v časovnem koraku k , lahko definiramo novo – preslikavi f_{pom} pripadajočo – preslikavo f'_{pom} , kot jo podaja enačba (12.32).

$$u_{pom} = f'_{pom}(\psi'_{nelin}(k)) \quad (12.32)$$

Preslikava f'_{pom} v enačbi (12.32) preslika v u_{pom} samo tisti del regresorja $\psi_{nelin}(k)$, ki ga označimo s $\psi'_{nelin}(k)$. Glede na preslikavo f_{pom} je preostali del regresorja $\psi_{nelin}(k)$ torej fiksni in ga določajo dejanska stanja oz. izhodi sistema v časovnem koraku k .

Pri nadaljnji obravnavi imamo več možnosti, ki so odvisne od lastnosti preslikave f'_{pom} v enačbi (12.32).

- * Preslikava f'_{pom} , ki jo podaja enačba (12.32), *je bijektivna*.

V tem primeru obstaja inverzna preslikava f'^{-1}_{pom} . Ustrezni regresor $\psi'_{nelin}(k)$, tj. ustrezne originalne vhode, določimo torej z inverzno preslikavo, kot podaja enačba (12.33).

$$\psi'_{nelin}(k) = f'^{-1}_{pom}(u_{pom}) \quad (12.33)$$

Na ta način so originalni vhodi sistema enolično določeni.

- * Preslikavi f_{pom} pripadajoča preslikava f'_{pom} v enačbi (12.32) *ni bijektivna*.

V tem primeru imamo pri nadaljnji obravnavi dve možnosti.

- Preslikavi f_{pom} pripadajoča preslikava f'_{pom} v enačbi (12.32) *ni surjektivna*.

To pomeni, da za nekatere vrednosti pomožnega vhoda u_{pom} ne obstaja inverzna preslikava f'^{-1}_{pom} . Takrat postopamo podobno kot v zgoraj navedenem primeru, ko preslikava f_{pom} v enačbi (12.3) *ni surjektivna*.

- Preslikavi f_{pom} pripadajoča preslikava f'_{pom} v enačbi (12.32) *ni injektivna*.

V tem primeru moramo uporabiti nadzorno logiko, s katero določimo najprimernejše vrednosti originalnih vhodov glede na preddefiniran (hevristični) kriterij, po katerem se nadzorna logika ravna.

12.3 Primer uporabe algoritma za vodenje temperature v šaržnem reaktorju

Algoritem za adaptivno prediktivno vodenje z identifikacijo lokalnega linearnega modela smo preizkusili na primeru vodenja šaržnega reaktorja, ki je opisan v 3. poglavju. V eksperiment smo vključili tudi dve eksotermni kemični reakciji, ki potekata v jedru šaržnega reaktorja in povzročata časovno spremenljivost dinamike celotnega sistema.

12.3.1 Modificirani matematični model šaržnega reaktorja

Matematični model šaržnega reaktorja je podan v enačbah v razdelku 3.2. Dinamiko v jedru šaržnega reaktorja opisuje enačba (3.1), dinamiko v plašču pa enačba (3.2).

Ker v jedru reaktorja potekata eksotermni kemični reakciji, moramo enačbo

za dinamiko jedra šaržnega reaktorja razširiti, kot jo podaja izraz (12.34).

$$mc \frac{dT}{dt} = \lambda S(T_w - T) + Q_r \quad (12.34)$$

V enačbi (12.34) označuje Q_r toplotno moč, ki jo proizvajata eksotermni kemični reakciji v jedru šaržnega reaktorja.

Model, ki ga uporablja algoritem za adaptivno prediktivno vodenje z identifikacijo lokalnega linearnega modela, mora biti zapisan v obliki *časovno spremenljive linearne samoregresijske strukture z zunanji vhodi*, kot jo podaja enačba (12.1) v razdelku 12.2.1. V enačbo za dinamiko plašča šaržnega reaktorja (3.2) torej vpeljemo pomožni vhod u_{pom} , kot ga podaja preslikava nelinearnega regresorja iz enačbe (3.2). Preslikava je zapisana v izrazu (12.35).

$$u_{pom} = k_M T_{in} + (1 - k_M) T_w \quad (12.35)$$

Omenimo naj, da pomožnega vhoda u_{pom} ne moremo direktno meriti. Kljub temu pa ga lahko ocenimo iz podatkov o temperaturi sveže vhodne vode T_{in} , temperature v plašču T_w in položaja mešalnega ventila k_M , kot kaže enačba (12.35). Pomožni vhod u_{pom} je tudi omejen, kot kaže izraz (12.36).

$$T_C = 12 \text{ }^\circ\text{C} \leq u_{pom} \leq T_H = 150 \text{ }^\circ\text{C} \quad (12.36)$$

Ob upoštevanju enačbe (12.35) lahko enačbo za dinamiko v plašču šaržnega reaktorja (3.2) zapišemo v obliki, kot jo podaja enačba (12.37).

$$\begin{aligned} m_w c_w \frac{dT_w}{dt} = & \phi c_w u_{pom} - \phi c_w T_w - \\ & - \lambda S(T_w - T) - \lambda_0 S_0(T_w - T_0) \end{aligned} \quad (12.37)$$

Maso komponent v jedru šaržnega reaktorja m določa enačba (12.48), masa vode v plašču pa je $m_w = 200$ kg. Specifično toploto komponent v jedru c določa

enačba (12.50), specifična toplota vode v plašču pa je $c_w = 4200 \text{ J kg}^{-1} \text{ K}^{-1}$. Pretok vode v plašču znaša $\phi = 1,6 \text{ kg s}^{-1}$, prevodnost toplotnega stika med jedrom in plaščem znaša $\lambda = 420 \text{ W m}^{-2} \text{ K}^{-1}$, prevodnost toplotnega stika med plaščem in okolico znaša $\lambda_0 = 84 \text{ W m}^{-2} \text{ K}^{-1}$, površina toplotnega stika med jedrom in plaščem znaša $S = 2 \text{ m}^2$, površina toplotnega stika med plaščem in okolico pa znaša $S_0 = 4 \text{ m}^2$.

12.3.2 Kinetični model eksotermnih kemičnih reakcij

Kinetična modela eksotermnih reakcij, ki ju obravnavamo v tem eksperimentu, sta bila razvita za *Warren Spring Laboratory* [115, 53, 50]. Obravnavamo torej dve kemični reakciji, ki ju podajata izraza (12.38) in (12.39).



Koncentracija (število kilomolov) komponent A, B, C in D se spreminja v skladu s hitrostjo reakcije, v kateri se proizvaja komponenta C, in hitrostjo reakcije, v kateri se proizvaja komponenta D. Če hitrost prve reakcije označimo z R_1 , druge pa z R_2 , lahko zapišemo dinamiko spreminjanja koncentracij posameznega reagenta, kot jih podajajo enačbe (12.40), (12.41), (12.42) in (12.43).

$$\frac{dM_A}{dt} = -R_1 - R_2 \quad (12.40)$$

$$\frac{dM_B}{dt} = -R_1 \quad (12.41)$$

$$\frac{dM_C}{dt} = +R_1 - R_2 \quad (12.42)$$

$$\frac{dM_D}{dt} = +R_2 \quad (12.43)$$

Če predpostavimo, da so reagenti v tekočem agregatnem stanju in dobro premešani, potem sta hitrosti reakcij R_1 in R_2 odvisni od koncentracij reagentov in hitrostnih konstant k_1 in k_2 , kot je zapisano v enačbah (12.44) in (12.45).

$$R_1 = k_1 M_A M_B \text{ kmol}^{-1} \text{ s}^{-1} \quad (12.44)$$

$$R_2 = k_2 M_A M_C \text{ kmol}^{-1} \text{ s}^{-1} \quad (12.45)$$

Hitrostni spremenljivki k_1 in k_2 reakcij (12.38) in (12.39) sta odvisni od temperature reagentov, kar opisuje *Arrheniusova enačba*. Spremenljivki sta določeni v enačbah (12.46) in (12.47).

$$k_1 = e^{(k_1^1 - \frac{k_1^2}{T+273,15 \text{ } ^\circ\text{C}})} \quad (12.46)$$

$$k_2 = e^{(k_2^1 - \frac{k_2^2}{T+273,15 \text{ } ^\circ\text{C}})} \quad (12.47)$$

Ostale fizikalne količine, ki nastopajo v reakciji, so podane v enačbah (12.48), (12.49), (12.50) in (12.51).

$$m = w_A M_A + w_B M_B + w_C M_C + w_D M_D \quad (12.48)$$

$$M = M_A + M_B + M_C + M_D \quad (12.49)$$

$$c = \frac{c_A M_A + c_B M_B + c_C M_C + c_D M_D}{M} \quad (12.50)$$

Tabela 12.1: Parametri kemičnih reakcij.

OZNAKA	VREDNOST	POMEN
w_A	30 kg kmol ⁻¹	specifična masa komponente A
w_B	100 kg kmol ⁻¹	specifična masa komponente B
w_C	130 kg kmol ⁻¹	specifična masa komponente C
w_D	160 kg kmol ⁻¹	specifična masa komponente D
c_A	75,31 kJ kmol ⁻¹ °C ⁻¹	specifična toplota komponente A
c_B	167,36 kJ kmol ⁻¹ °C ⁻¹	specifična toplota komponente B
c_C	217,57 kJ kmol ⁻¹ °C ⁻¹	specifična toplota komponente C
c_D	334,73 kJ kmol ⁻¹ °C ⁻¹	specifična toplota komponente D
k_1^1	20,9057	hitrostna konstanta reakcije
k_1^2	10000 °C	hitrostna konstanta reakcije
k_2^1	38,9057	hitrostna konstanta reakcije
k_2^2	17000 °C	hitrostna konstanta reakcije
ΔH_1	-41840 kJ kmol ⁻¹	entalpija reakcije
ΔH_2	-25105 kJ kmol ⁻¹	entalpija reakcije

$$Q_r = -\Delta H_1 R_1 - \Delta H_2 R_2 \quad (12.51)$$

V enačbah (12.48), (12.49), (12.50) in (12.51) označuje m skupno maso komponent; M označuje skupno koncentracijo, M_A , M_B , M_C in M_D pa koncentracijo posamezne komponente v jedru; c pa je specifična toplota zmesi v jedru šaržnega reaktorja.

Parametri reakcije so podani v tabeli 12.1.

12.3.3 Ocenjevanje parametrov

V primeru šaržnega reaktorja lahko iz predhodnega znanja o procesu oz. poznavanja enačb (12.34), (12.35) in (12.37) predpostavimo strukturo inkrementalnega

modela v časovnodiskretni obliki.

Parametre procesa, vključno s časovno spremenljivimi lastnostmi kemične reakcije, ki se odvija v reaktorju, moramo identificirati sproti, tj. med delovanjem reaktorja.

Kot kažeta enačbi (12.52) in (12.53), uporabljamo dva neodvisna modela v obliki časovno spremenljive linearne samoregresijske strukture z zunanjimi vhodi, ki jima tudi neodvisno ocenjujemo parametre. Indeks f označuje filtrirani signal.

$$y_{f1}(k) = \psi_{f1}^T(k)\theta_1(k) \quad (12.52)$$

$$y_{f2}(k) = \psi_{f2}^T(k)\theta_2(k) \quad (12.53)$$

$y_{f1}(k) = T_{w,f}(k)$ označuje filtrirani izhod prvega modela, ki predstavlja temperaturo v plašču, $y_{f2}(k) = T_f(k)$ pa filtrirani izhod drugega modela, ki predstavlja temperaturo v jedru šaržnega reaktorja. Regresorji vsebujejo temperaturo plašča in jedra reaktorja ter pomožni vhod, kot kažeta enačbi (12.54) in (12.55).

$$\psi_{f1}^T(k) = [T_{w,f}(k-1) \ T_f(k-1) \ u_{pom,f}(k-1)] \quad (12.54)$$

$$\psi_{f2}^T(k) = [T_{w,f}(k-1) \ T_f(k-1)] \quad (12.55)$$

Vektorja parametrov, ki ju podajata enačbi (12.56) in (12.57), sta sestavljena iz treh oz. dveh parametrov, ki jih moramo sproti identificirati.

$$\theta_1^T = \left[\theta_{11}(k) \ \theta_{12}(k) \ \theta_{13}(k) \right] \quad (12.56)$$

$$\theta_2^T = \left[\theta_{21}(k) \ \theta_{22}(k) \right] \quad (12.57)$$

Inkrementalni model dobimo tako, da s filtriranjem in diferenciacijo merjenih signalov odštejemo enosmerne komponente signalov, kot kaže enačba (12.5) v razdelku 12.2.2. V našem primeru sta parametra filtra, kot ju določa enačba (12.7), $k_f = 0,95$ in $p = 3$.

Oba modela v enačbah (12.52) in (12.53) sta linearna v parametrih, kar nam omogoča analitično izračunavanje parametrov z metodo najmanjše vsote kvadratov odstopanj. Parametre procesa algoritem ocenjuje na podlagi enačb (12.9), (12.10) in (12.11) v razdelku 12.2.2.

Če enačbi (12.52) in (12.53) pretvorimo v z -prostor, lahko opišemo dinamiko spremenljivk $T_{w,f}(k)$ in $T_f(k)$, ki je odvisna od pomožnega vhoda $u_{pom,f}$, s prenosnima funkcijama $G_{mw}(z)$ in $G_m(z)$, ki ju podajata enačbi (12.58) in (12.59).

$$G_{mw}(z) = \frac{T_{w,f}(z)}{u_{pom,f}(z)} = \frac{b_{1j}z - b_{0j}}{z^2 - a_1z - a_0} \quad (12.58)$$

$$G_m(z) = \frac{T_f(z)}{u_{pom,f}(z)} = \frac{b_0}{z^2 - a_1z - a_0} \quad (12.59)$$

Parametri v enačbah (12.58) in (12.59) so podani v izrazih (12.60).

$$\begin{aligned} b_{0j} &= \theta_{22}\theta_{13} \\ b_{1j} &= \theta_{13} \\ b_0 &= \theta_{21}\theta_{13} \\ a_1 &= \theta_{22} + \theta_{11} \\ a_0 &= \theta_{12}\theta_{21} - \theta_{11}\theta_{22} \end{aligned} \quad (12.60)$$

Če predpostavimo, da je proces spoznaven, lahko prenosni funkciji združimo v prostoru stanj in zapišemo v spoznavnostni kanonični obliki.

Parametri algoritma za ocenjevanje parametrov modela, ki smo jih nastavili pred izvedbo eksperimenta, so podani spodaj.

- Čas vzorčenja signalov je $t_S = 20$ s. Čas vzorčenja je bil izbran glede na časovne konstante procesa.
- Začetne vrednosti kovariančnih matrik so bile nastavljene na $P_1(0) = 100I_3$ in $P_2(0) = 100I_2$. Tu označuje I_j enotino matriko dimenzije $j \times j$.
- Začetne vrednosti parametrov procesa so bile nastavljene na $\hat{\theta}_{11}(0) = \hat{\theta}_{22}(0) = 1$ in $\hat{\theta}_{12}(0) = \hat{\theta}_{13}(0) = \hat{\theta}_{21}(0) = 0$.
- Faktorja pozabljanja za oba modela sta bila nastavljena na $\gamma_1 = \gamma_2 = 0,995$.
- Faktorja mrtve cone za oba modela sta bila nastavljena na $k_{DZ1} = k_{DZ2} = 0,01$.

12.3.4 Algoritem za prediktivno funkcionalno vodenje

V primeru šaržnega reaktorja želimo, da vodenje zagotovi hitro sledenje referenčni trajektoriji in da čim bolj zmanjšamo število preklpov zapornih ventilov.

V eksperimentu smo predikcijski horizont algoritma za prediktivno funkcionalno vodenje nastavili na $H = 10$. Parameter referenčnega modela smo nastavili na $a_r = 0,925$.

Regulirni zakon za šaržni reaktor izpeljemo, kot je opisano v razdelku 12.2.3. Podajata ga enačbi (12.29) in (12.30).

Regulirni zakon v vsakem časovnem koraku k vrne izračunano vrednost pomožnega vhoda² $u_{pom}(k)$, ki pa jo moramo še preslikati v prostor originalnih vhodov sistema – $k_M(k)$, $k_H(k)$ in $k_C(k)$ – kar naredimo v zadnjem koraku algoritma.

Položaj zapornih ventilov določimo z nadzorno logiko, ki jo podaja enačba (12.61).

²Preslikava je določena v enačbi (12.35).

$$\begin{aligned} \text{\textbf{če je}} \quad w(k) - T(k) < \delta_e \quad \text{\textbf{potem}} \quad k_H(k) = 0 \wedge k_C(k) = 1 \\ \text{\textbf{sicer}} \quad k_H(k) = 1 \wedge k_C(k) = 0 \end{aligned} \quad (12.61)$$

V enačbi (12.61) δ_e določa prag za preklon ($\delta_e = -1^{\circ}\text{C}$).

Položaj mešalnega ventila $k_M(k)$, ki ima vlogo direktne regulirne veličine, je določen z enačbo (12.62).

$$k_M(k) = \frac{u_{pom}(k) - T_w(k)}{T_{in}(k) - T_w(k)} \quad (12.62)$$

Temperaturo sveže vhodne vode $T_{in}(k)$ v enačbi (12.62) določa položaj zapornih ventilov $k_H(k)$ in $k_C(k)$.

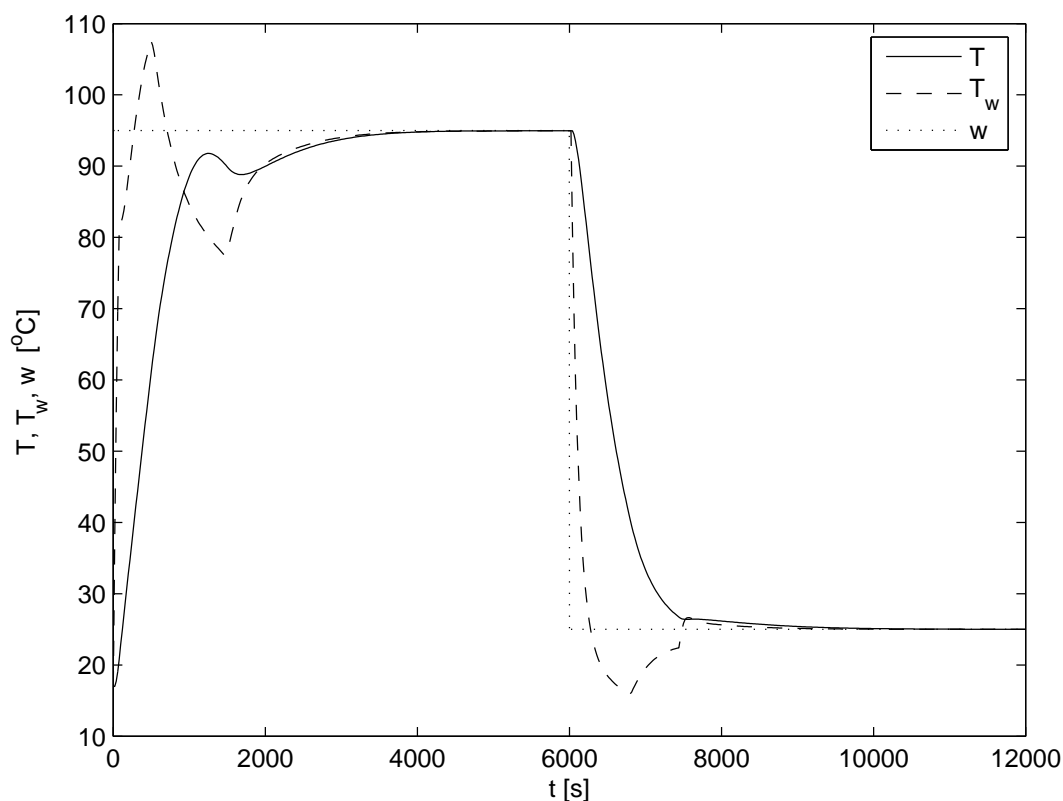
12.3.5 Rezultati

V [115] so za optimalne rezultate predlagali ekvimolarno začetno polnjenje reaktorja z reagentoma A in B, optimalna reakcijska temperatura pa naj bi se vzdrževala pri 95°C . Na ta način smo določili potek referenčne temperature kot odsekoma konstantno funkcijo z začetkom pri 95°C , kar zagotavlja optimalen potek reakcije, ko pa se reakcija umiri, se temperatura zniža na 25°C , da se vsebina reaktorja pred menjavo šarže ohladi.

Začetna količina reagentov A in B je bila $M_A(0) = M_B(0) = 2$ kmol, začetna temperatura pa $T(0) = T_w(0) = T_0 = 17^{\circ}\text{C}$.

Slika 12.1 prikazuje potek referenčne temperature w , temperature v jedru T in v plašču reaktorja T_w , slika 12.2 pa potek vhodnih signalov k_M , k_H in k_C . Vidimo lahko, da temperatura v jedru zadovoljivo sledi referenčni temperaturi.

Na sliki 12.3 je prikazan potek reakcije, tj. koncentracij posameznih reagentov M_A , M_B , M_C , M_D . Opazimo lahko zmanjševanje koncentracij komponent A in B in povečevanje koncentracij komponent C in D med potekom reakcij. Hitrost poteka posamezne reakcije lahko določimo iz naklona poteka koncentracije



Slika 12.1: Temperatura v jedru T , v plašču T_w in referenčni signal w .

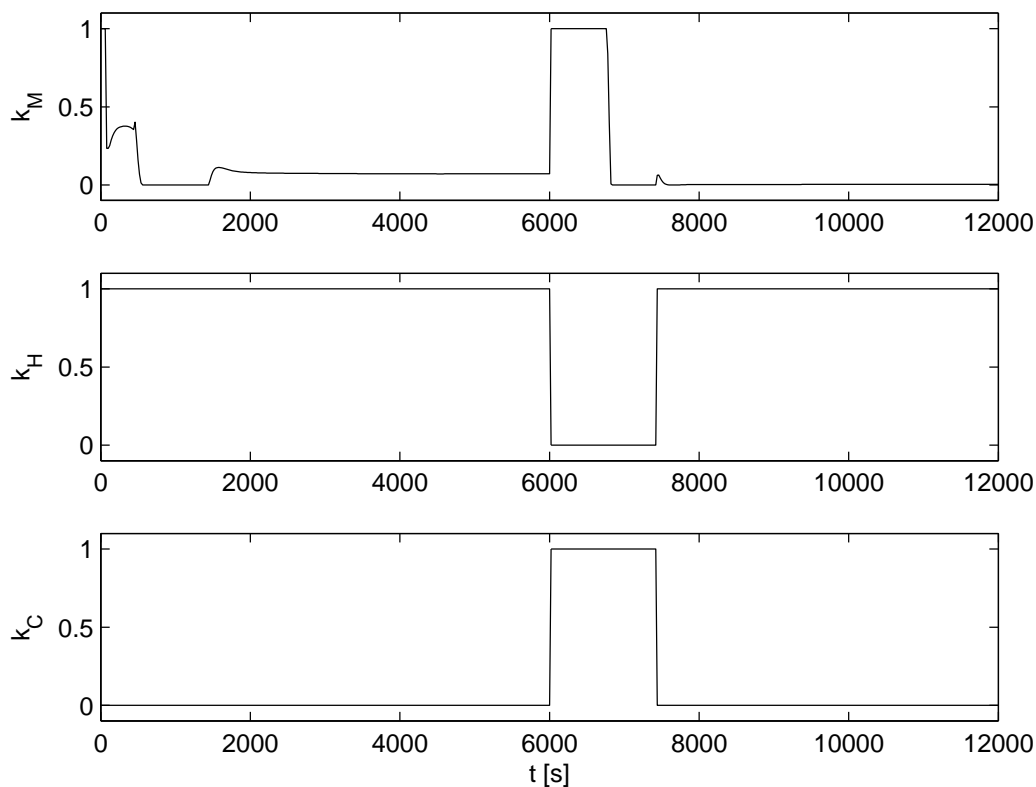
ustreznega reagenta. Slika 12.3 prikazuje tudi toplotno moč, ki se sprošča med potekom eksotermnih kemičnih reakcij.

Sliki 12.4 in 12.5 prikazujeta ocenjene vrednosti parametrov procesa θ_{11} , θ_{12} in θ_{13} oz. θ_{21} in θ_{22} . Slike so zaradi jasnosti ustrezno približane, vendar zato ne prikazujejo začetnih vrednosti parametrov.

12.3.6 Komentar

Uporabnost algoritma za adaptivno prediktivno vodenje z identifikacijo lokalnega linearnega modela smo preverili na primeru vodenja šaržnega reaktorja, ki je vključevalo tudi eksotermni reakciji.

Rezultati so pokazali, da uporaba algoritma omogoča izpolnitev zahtev vodenja – kljub izrazito eksotermnim reakcijam, ki v model procesa vpeljujeta časovno

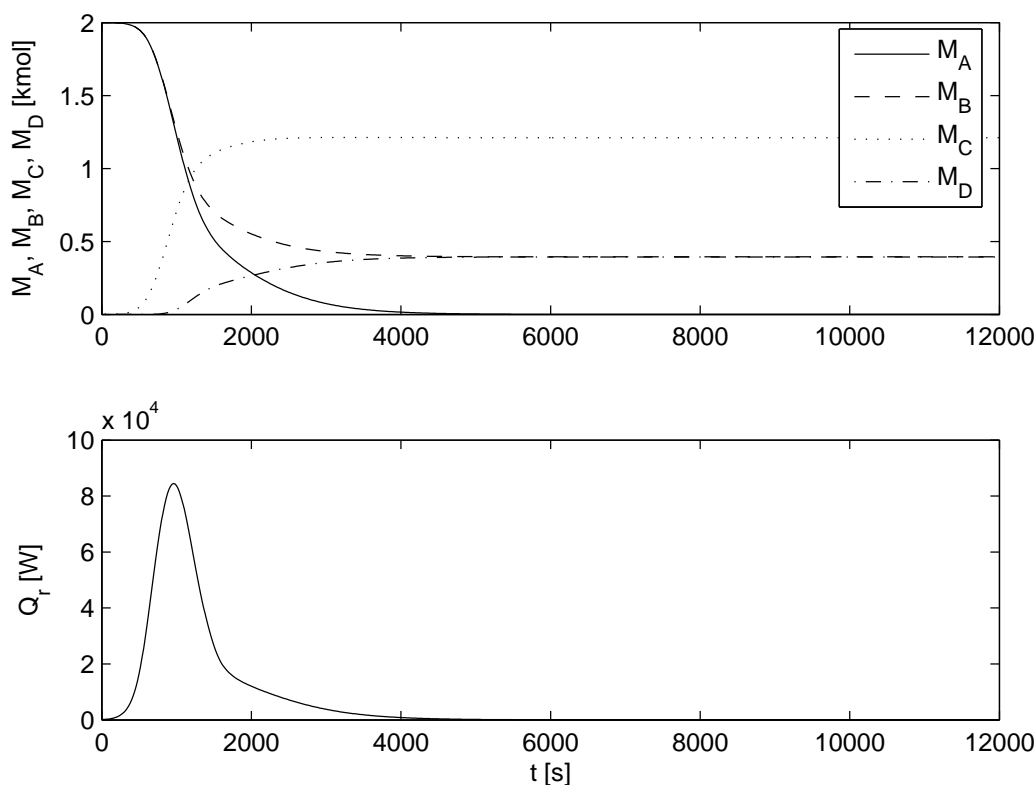
Slika 12.2: Vhodni signali k_M , k_H in k_C .

spreminjajoče parametre – in to kljub temu, da kinetičnega modela reakcij (v okviru algoritma) ne poznamo.

Temperatura v jedru zadovoljivo sledi referenčnemu signalu, kar zagotavlja kratek čas obdelave posamezne šarže, poleg tega pa vodenje brez prevzpona zagotavlja kakovost produkta. Tudi število preklopov zapornih ventilov je minimalno.

Adaptivna narava algoritma ima za posledico dobro kompenziranje izrazito eksotermnih kemičnih reakcij. S prilagajanjem časovno spremenljivim dinamičnim lastnostim procesa algoritem prepreči toplotni pobeg in omogoča odziv brez prevzpona, kar ne bi bilo mogoče, če bi uporabljali časovno nespremenljivi model.

Šaržni reaktorji imajo velikokrat izrazito nelinearno in hibridno dinamiko. Prav tako jih ne moremo vedno obravnavati kot časovno nespremenljive sisteme, posebno če se v njih odvijajo endo- ali eksotermne reakcije s pogosto neznanimi



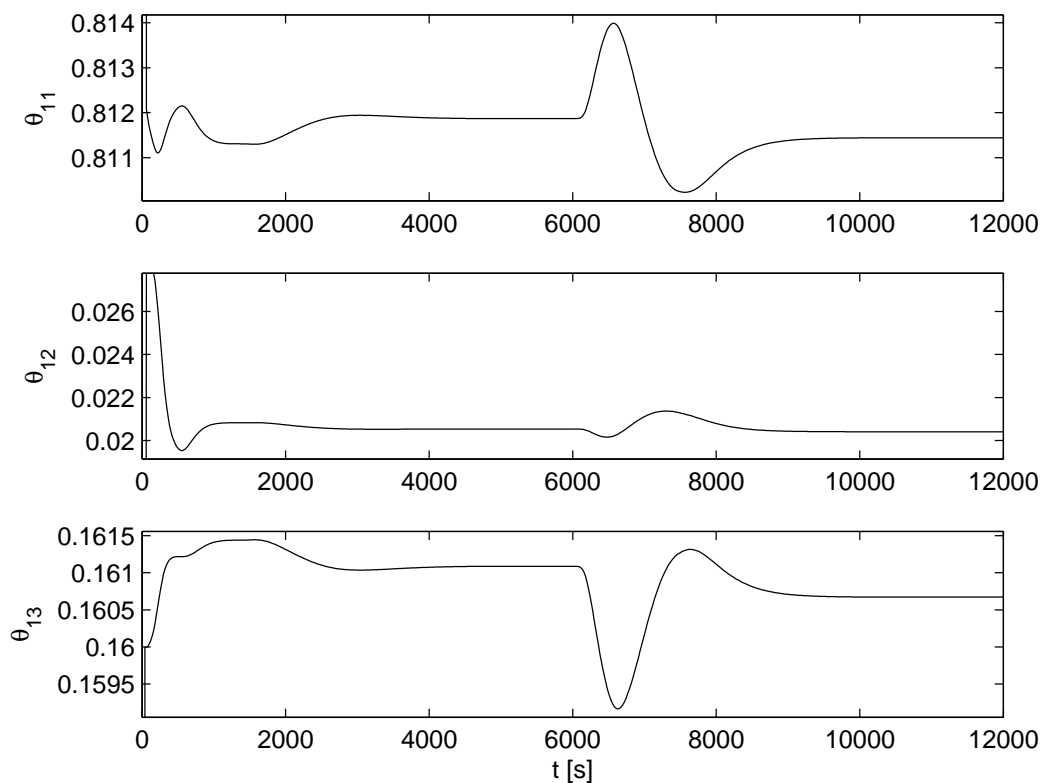
Slika 12.3: Koncentracije reagentov in toplotna moč.

kinetičnimi lastnostmi. V tem poglavju smo predstavili pristop, s katerim se lahko nekaterim omenjenim kompleksnim lastnostim procesa pri načrtovanju vodenja izognemo in na ta način omogočimo zadovoljivo vodenje tovrstnih sistemov.

Z vpeljavo pomožnega vhoda lahko sistem, ki ga želimo voditi, lineariziramo. Pomožni vhod v algoritmu obravnavamo kot sistemski vhod.

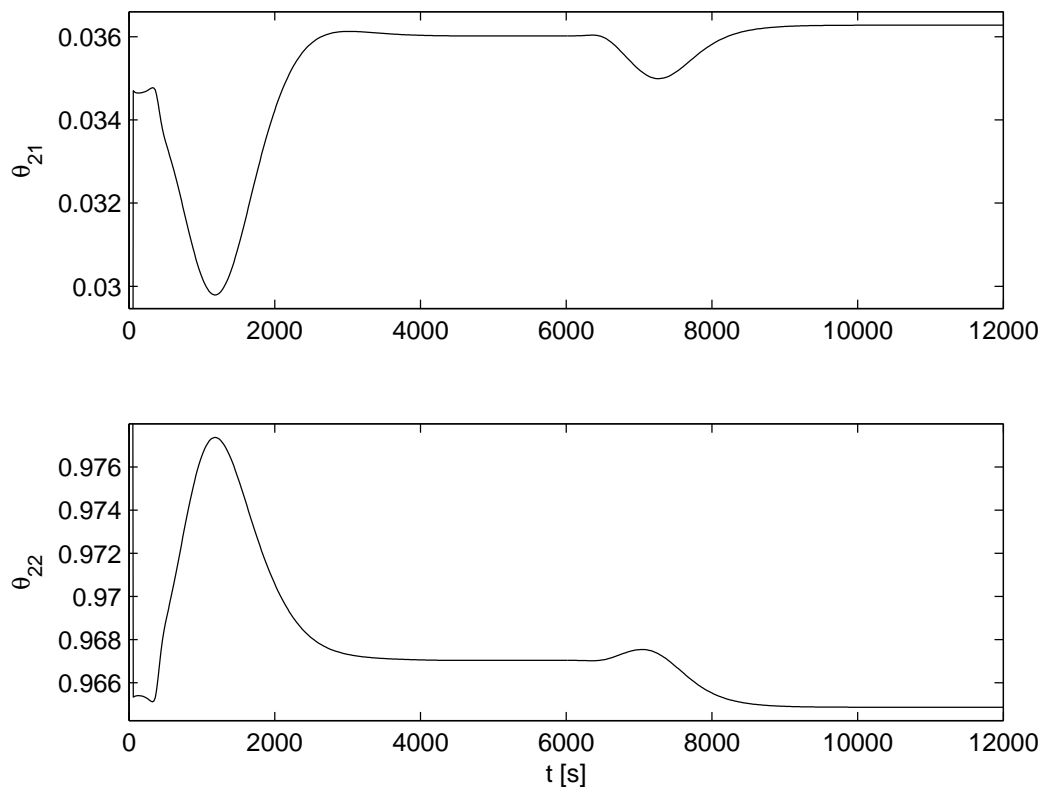
Linearizacija sistema olajša implementacijo metode za adaptivno prediktivno vodenje. Na ta način je mogoče ustrezno upoštevati časovno spremenljivost dinamike procesa, kar je nujno, saj imajo eksotermne reakcije, ki potekajo v jedru šaržnega reaktorja in katerih kinetičnega modela v osnovi ne poznamo, lahko velik vpliv na dinamične lastnosti celotnega sistema.

Hibridno dinamiko smo v primeru šaržnega reaktorja obravnavali v okviru nadzorne logike v zadnjem koraku algoritma za vodenje, tj. pri preslikavi

Slika 12.4: Ocenjene vrednosti parametrov θ_{11} , θ_{12} in θ_{13} .

pomožnega vhoda, ki ga regulirni zakon vrne v vsakem časovnem koraku, v prostor originalnih vhodov sistema. Na ta način smo lahko število preklpov zapornih ventilov omejili na minimum in s tem razbremenili aktuatorje, kar ima za posledico manjšo obrabo in manjšo porabo energije, ki je potrebna za vodenje šaržnega reaktorja.

Algoritem določa regulirni zakon v eksplicitni obliki, zato je računski zahtevnost izvedbe majhna, posebno v primerjavi z optimizacijskimi pristopi. Algoritem za implementacijo torej ne zahteva drage strojne opreme.



Slika 12.5: Ocenjene vrednosti parametrov θ_{21} in θ_{22} .

13. Vodenje z inverznim modelom

V tem poglavju bomo opisali metodo za vodenje z inverznim modelom [80], primerno za sisteme z enim izhodom, ki jih lahko zapišemo v obliki hibridnega mehkega modela – glej 5. poglavje.

13.1 Dvodelnost algoritma za vodenje z inverznim modelom

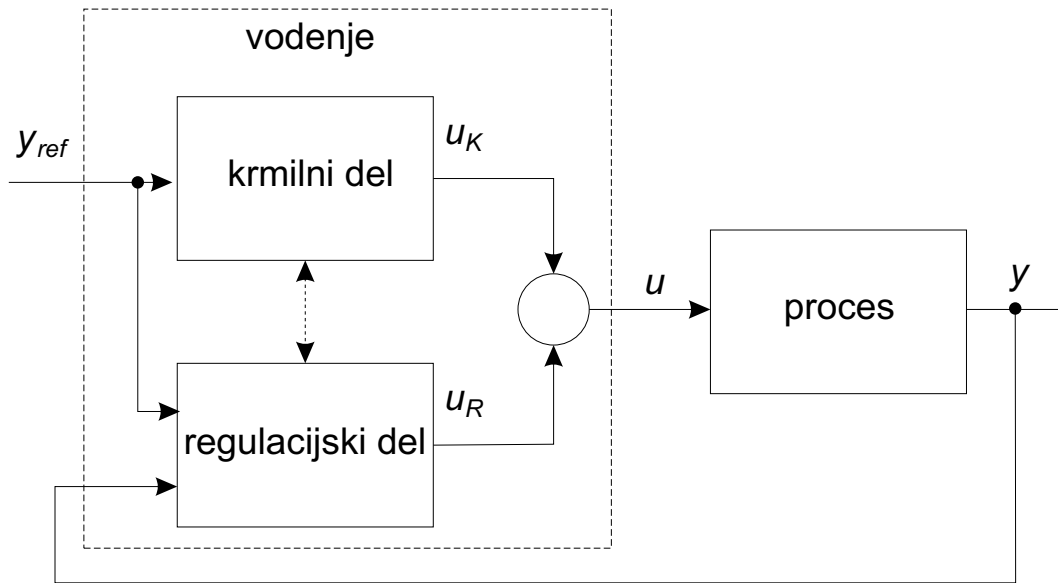
Osnovna ideja obravnavanega pristopa je dvodelnost algoritma za vodenje. Algoritem je sestavljen iz:

- *krmilnega dela* (angl. *feedforward*) in
- *regulacijskega dela* (angl. *feedback*).

Splošna shema vodenja je prikazana na sliki 13.1.

Krmilni del algoritma za vodenje uporablja hibridni mehki model (glej 5. poglavje) sistema, ki ga želimo voditi, oz. njegov inverzni model, s katerim določi primerne krmilne signale u_K za sledenje referenčni trajektoriji y_{ref} . Krmilni del algoritma torej iz referenčnega signala y_{ref} v vsakem časovnem koraku k določa krmilne signale u_K .

Regulacijski del algoritma izračuna ustrezne popravke krmilnih signalov u_K , tj. regulirne signale u_R , ki jih določi glede na izmerjeni izhodni signal sistema y .



Slika 13.1: Splošna shema vodenja.

Osnovnim krmilnim signalom u_K algoritem prišteje izračunani popravek, tj. regulirne signale u_R . Vsota obeh signalov je končni vhodni signal u , ki se dejansko uporabi za vodenje procesa v vsakem časovnem koraku k , kot kaže enačba (13.1).

$$u(k) = u_K(k) + u_R(k) \quad (13.1)$$

Glavna prednost predlaganega pristopa je, da lahko s krmilnim delom algoritma za vodenje izhod sistema y pripelje v bližino referenčne trajektorije y_{ref} .

Omenimo, da je dejanska oddaljenost izhoda sistema y od referenčne trajektorije y_{ref} , ki jo lahko dosežemo samo s krmilnim delom algoritma za vodenje, odvisna od natančnosti hibridnega mehkega modela in dosegljivosti referenčne trajektorije glede na morebitne omejitve vhodov, izhodov in stanj sistema. Omejitve lahko izhajajo iz fizičnih lastnosti sistema ali iz tehnoloških zahtev, ki so nujne za kakovostno vodenje.

Z uporabo hibridnega mehkega modela krmilni del pri določanju krmilnih signalov u_K upošteva nelinearno in hibridno dinamiko sistema, pa tudi morebitne omejitve vhodov, izhodov in stanj sistema.

Ko je izhod sistema y v bližini referenčne trajektorije y_{ref} , lahko uporabimo relativno preprost regulacijski del, s katerim zagotovimo kakovostno vodenje. Glede na to, da regulacijski del algoritma za izračun regulirnega signala u_R upošteva izmerjeni izhodni signal sistema y , predstavlja dopolnitev krmilnega dela algoritma za vodenje in na ta način kompenzira odstopanja izhoda sistema y od referenčne trajektorije y_{ref} . Do odstopanj lahko pride zaradi različnih vzrokov, kot so natančnost hibridnega mehkega modela, šum, morebitne motnje v realnem sistemu ipd.

13.2 Inverzni hibridni mehki model

Kot smo omenili v razdelku 13.1, uporablja krmilni del algoritma za vodenje za izračun primernih krmilnih signalov u_K za sledenje referenčni trajektoriji y_{ref} *inverzni hibridni mehki model* sistema, ki ga želimo voditi. V tem razdelku bomo torej izpeljali *inverzni hibridni mehki model*.

Hibridni mehki model je natančneje opisan v 5. poglavju. Predpostavimo, da hibridni mehki model sistema, ki ga želimo voditi, predhodno poznamo.

Izhod hibridnega mehkega modela v kompaktni obliki, ki ga podaja enačba (5.10) v razdelku 5.2.3, lahko ponovno zapišemo, kot kaže enačba (13.2).

$$\hat{y}_p(k+1) = \beta(y(k)) \Theta^T(q(k)) \psi(k) \quad (13.2)$$

Kot je zapisano v enačbi (13.2), predpostavimo, da ima sistem en izhod in da so mehke množice oz. pripadnostne funkcije v modelu določene glede na ta izhod ($\beta(k) = \beta(y(k))$). Poleg tega predpostavimo tudi, da ima sistem en zvezni vhod u_{zv} , ki je v zapisu (13.2) vključen v regresorju $\psi(k)$.

Predpostavimo, da je način delovanja, tj. diskretno stanje $q(k)$, določen z diskretnimi vhodi. V primeru, da je način delovanja odvisen tudi od izhoda modela (in ne samo od diskretnih vhodov), lahko to upoštevamo že v zapisu

modela: poleg mehkih množic A_i^j uvedemo v pravila hibridnega mehkega modela R^{jd} tudi ostre množice, ki določajo odsekoma konstantne pripadnostne funkcije v zvezi z načinom delovanja – glej izraz (5.8) v razdelku 5.2.2.

Če upoštevamo enačbe (5.12), (5.13), (5.14) in (5.15) v razdelku 5.2.3 in pomen parametrov v teh enačbah, lahko zapišemo izraze (13.3), (13.4), (13.5) in (13.6).

$$S_{A,inv} = \sum_{i=1}^n \sum_{j=1}^K \beta_j(y(k)) a_{ijd}(q(k)) y(k - i + 1) \quad (13.3)$$

$$S_{B,inv} = \sum_{i=2}^m \sum_{j=1}^K \beta_j(y(k)) b_{ijd}(q(k)) u_{zv}(k - i + 1) \quad (13.4)$$

$$S_{R,inv} = \sum_{j=1}^K \beta_j(y(k)) r_{jd}(q(k)) \quad (13.5)$$

$$S_{U,inv} = \sum_{j=1}^K \beta_j(y(k)) b_{1jd}(q(k)) \quad (13.6)$$

Z nekaj matematičnimi operacijami lahko z uporabo enačb (5.12), (5.13), (5.14) in (5.15) v razdelku 5.2.3 ter prej zapisanih izrazov (13.3), (13.4), (13.5) in (13.6) zapišemo izhod inverznega modela, ki ga podaja enačba (13.7).

$$u_{zv}(k) = \frac{\hat{y}_p(k + 1) - S_{A,inv} - S_{B,inv} - S_{R,inv}}{S_{U,inv}} \quad (13.7)$$

Inverzni model v obliki, ki je zapisan v izrazu (13.7), predstavlja osnovo za načrtovanje krmilnega dela algoritma za vodenje.

13.3 Krmilni del algoritma za vodenje

Inverzni hibridni mehki model, ki smo ga v razdelku 13.2 izpeljali iz zapisa hibridnega mehkega modela, uporabimo v krmilnem delu algoritma za vodenje.

Osnovni mehanizem krmiljenja z inverznim modelom je, da na vhod krmilnega dela pripeljemo referenčni signal y_{ref} , izhod pa v vsakem časovnem koraku k vrača izračunane krmilne signale u_K , ki naj bi dejanski izhod sistema y čim bolj približali referenčnemu signalu y_{ref} .

Z uporabo hibridnega mehkega modela krmilni del pri določanju krmilnih signalov u_K upošteva nelinearno in hibridno dinamiko sistema. Seveda je pri izračunavanju krmilnih signalov u_K treba upoštevati tudi morebitne omejitve vhodov, izhodov in stanj sistema, ki lahko izhajajo iz fizičnih lastnosti sistema ali iz tehnoloških zahtev, ki so nujne za kakovostno vodenje.

Krmilni del algoritma za vodenje mora v vsakem časovnem koraku k izvesti naslednje korake.

- Izvorni referenčni signal y_{ref} mora prilagoditi tako, da ga premakne v območje zaloge vrednosti predikcije izhoda sistema, ki ga želimo voditi, v naslednjem časovnem koraku $\hat{y}_p(k+1)$.
- Izbrati mora najustreznejši način delovanja, tj. diskretni del krmilnega signala u_K .
- Izbrati mora najustreznejši zvezni vhod v sistem, tj. zvezni del krmilnega signala u_K .

Na ta način bo lahko krmilni del algoritma za vodenje izhod sistema y približal prilagojenemu referenčnemu signalu \hat{y}_{ref} .

13.3.1 Matrika dosegljivosti hibridnega mehkega modela

V prvem koraku krmilni del algoritma za vodenje uporabi hibridni mehki model sistema, ki ga podaja enačba (13.2), za izračun zaloge vrednosti predikcije izhoda sistema, ki ga želimo voditi, v naslednjem časovnem koraku $\hat{y}_p(k+1)$. Pri tem upošteva morebitne omejitve vhodov oz. izhoda sistema.

Poudariti je treba, da je hibridni mehki model monoton v smislu, da ima skrajna¹ vrednost zveznega dela vhodnega signala u_{zv} za posledico skrajno vrednost izhoda hibridnega mehkega modela $\hat{y}_p(k+1)$ v okviru vsakega načina delovanja ($q = 1, \dots, s$). Zaradi tega lahko samo s po dvema izračunoma za vsak način delovanja ($q = 1, \dots, s$) določimo matriko dosegljivosti hibridnega mehkega modela $\hat{R}_p(k+1)$, ki jo podaja izraz (13.8).

$$\hat{R}_p(k+1) = \begin{bmatrix} \hat{y}_{p,min}(k+1)|_{q(k)=1} & \hat{y}_{p,max}(k+1)|_{q(k)=1} \\ \vdots & \vdots \\ \hat{y}_{p,min}(k+1)|_{q(k)=s} & \hat{y}_{p,max}(k+1)|_{q(k)=s} \end{bmatrix} \quad (13.8)$$

Matrika dosegljivosti hibridnega mehkega modela $\hat{R}_p(k+1)$, ki jo podaja izraz (13.8), je sestavljena iz s vrstic in 2 stolpcev in vsebuje intervale zaloge vrednosti predikcije izhoda sistema, ki ga želimo voditi, v naslednjem časovnem koraku $\hat{y}_p(k+1)$, in sicer glede na način delovanja $q(k)$. Vsaka vrstica označuje enega od mogočih načinov delovanja ($q(k) = 1, \dots, s$).

13.3.2 Prilagoditev referenčnega signala

Pri večini sistemov, ki jih želimo voditi, imamo opravka z omejitvami vhodov, izhodov in stanj sistema, ki lahko izhajajo iz fizičnih lastnosti sistema ali iz tehnoloških zahtev, ki so nujne za kakovostno vodenje. Te omejitve moramo seveda upoštevati, kar pa pri osnovnem vodenju z inverznim modelom ni mogoče.

Inverzni model namreč iz referenčnega signala y_{ref} , ki ga pripeljemo na njegov vhod, na svojem izhodu vrne izračunani vhodni signal sistema, pri tem pa se lahko zgodi, da je vhodni signal sistema nedosegljiv (zaradi omejitev v realnem sistemu) ali celo nerealen (neskončen signal, npr. diracov impulz). Do izračuna neustreznega vhodnega signala pride zaradi nedosegljivosti referenčne trajektorije, ki ne

¹Skrajna v smislu, da je vrednost zveznega dela vhodnega signala u_{zv} na skrajnem robu omejenega definicijskega območja u_{zv} .

upošteva omejitve v sistemu. Če torej želimo z inverznim modelom določiti ustrezen vhodni signal, mora algoritem predhodno referenčni signal prilagoditi, tako da postane dosegljiv v okviru sistema in njegovih omejitev. Prilagojeni referenčni signal označimo z \hat{y}_{ref} in ga določimo, kot je opisano v nadaljevanju.

Pri določanju prilagojenega referenčnega signala \hat{y}_{ref} obravnavamo dve možnosti.

1. Referenčni signal pripada (vsaj) enemu izmed intervalov zaloge vrednosti predikcije izhoda sistema v naslednjem časovnem koraku, ki sestavljajo matriko dosegljivosti hibridnega mehkega modela $\hat{R}_p(k+1)$ v enačbi (13.8). To pomeni, da velja izraz (13.9).

$$\begin{aligned} \exists d \in \{1, 2, \dots, s\}, \text{ tako da velja} \\ y_{ref}(k) \in [\hat{y}_{p,min}(k+1)|_{q(k)=d}, \hat{y}_{p,max}(k+1)|_{q(k)=d}]. \end{aligned} \quad (13.9)$$

Če velja izraz (13.9), lahko ugotovimo, da je referenčna trajektorija v tem časovnem koraku dosegljiva.

V tem primeru prilagojenemu referenčnemu signalu priredimo vrednost izvornega referenčnega signala, kot podaja izraz (13.10).

$$\hat{y}_{ref}(k) = y_{ref}(k) \quad (13.10)$$

2. Referenčni signal ne pripada nobenemu izmed intervalov zaloge vrednosti predikcije izhoda sistema v naslednjem časovnem koraku, ki sestavljajo matriko dosegljivosti hibridnega mehkega modela $\hat{R}_p(k+1)$ v enačbi (13.8). To pomeni, da velja izraz (13.11).

$$\begin{aligned} \nexists d \in \{1, 2, \dots, s\}, \text{ tako da bi veljalo} \\ y_{ref}(k) \in [\hat{y}_{p,min}(k+1)|_{q(k)=d}, \hat{y}_{p,max}(k+1)|_{q(k)=d}]. \end{aligned} \quad (13.11)$$

Če velja izraz (13.11), lahko ugotovimo, da referenčna trajektorija v tem časovnem koraku ni dosegljiva.

Izvorni referenčni signal moramo v tem primeru prilagoditi tako, da ga premaknemo v območje zaloge vrednosti predikcije izhoda sistema v naslednjem časovnem koraku. Algoritem torej izbere tisti element matrike dosegljivosti hibridnega mehkega modela $\hat{R}_p(k+1)$, ki je najbližji izvorni referenčni vrednosti $y_{ref}(k)$.

V tem primeru prilagojenemu referenčnemu signalu priredimo vrednost izbranega elementa matrike dosegljivosti hibridnega mehkega modela $\hat{R}_p(k+1)$, kot podaja izraz (13.10).

$$\begin{aligned} \hat{y}_{ref}(k) &= \arg_x \min |x - y_{ref}(k)|, \\ \text{kjer velja } x &\in \{[\hat{R}_p(k+1)]_{i,j}\} \\ &\text{za } i = 1, 2, \dots, s \text{ in } j = 1, 2 \end{aligned} \quad (13.12)$$

V izrazu (13.10) označuje $[\hat{R}_p(k+1)]_{i,j}$ element v i -ti vrstici in j -tem stolpcu matrike dosegljivosti hibridnega mehkega modela $\hat{R}_p(k+1)$.

13.3.3 Določitev diskretnega dela krmilnega signala

Diskretni del krmilnega signala, tj. način delovanja oz. diskretno stanje hibridnega mehkega modela, algoritem določi glede na prilagojeni referenčni signal $\hat{y}_{ref}(k)$ in matriko dosegljivosti hibridnega mehkega modela $\hat{R}_p(k+1)$.

Pri izbiri načina delovanja sistema $q(k)$ moramo upoštevati obstoj oz. število (r) načinov delovanja $q(k)$, ki zadoščajo enačbi (13.13).

$$\begin{aligned} y_{ref}(k) &\in [\hat{y}_{p,min}(k+1)|_{q(k)=d_i}, \hat{y}_{p,max}(k+1)|_{q(k)=d_i}] \\ &\text{pri čemer } d_i \in \{1, 2, \dots, s\} \text{ za vsak } i = 1, \dots, r. \end{aligned} \quad (13.13)$$

Glede na število r obravnavamo tri različice, ki so opisane v nadaljevanju.

- Ne obstaja noben način delovanja sistema $q(k) = d_i$, ki bi zadoščal enačbi (13.13), tj. $r = 0$, kot je podano v množici rešitev Q_r v izrazu (13.14).

$$q(k) \in Q_r = \{\} \quad (13.14)$$

Glede na to, da smo referenčni signal premaknili v območje zaloge vrednosti predikcije izhoda sistema v naslednjem časovnem koraku, kot je opisano v razdelku 13.3.2, se opisana situacija sploh ne more zgoditi, zato lahko njeno obravnavo v algoritmu za vodenje izpustimo.

- Število načinov delovanja sistema $q(k) = d_i$, ki zadoščajo enačbi (13.13), je enako $r = 1$, kot je podano v množici rešitev Q_r v izrazu (13.15).

$$q(k) \in Q_r = \{d_1\} \quad (13.15)$$

V tem primeru je diskretni del krmilnega signala enolično določen v množici rešitev Q_r v izrazu (13.15), saj množica vsebuje samo en element.

- Število načinov delovanja sistema $q(k) = d_i$, ki zadoščajo enačbi (13.13), je enako $r > 1$, kot je podano v množici rešitev Q_r v izrazu (13.15).

$$q(k) \in Q_r = \{d_1, d_2, \dots, d_r\} \quad (13.16)$$

V tem primeru so v množici rešitev Q_r v izrazu (13.16) podane vse tiste rešitve za določitev diskretnega dela krmilnega signala, z uporabo katerih lahko – glede na hibridni mehki model – predikcija izhoda sistema v naslednjem časovnem koraku $\hat{y}_p(k+1)$ zavzame vrednost prilagojenega referenčnega signala $\hat{y}_{ref}(k)$.

Ugotovimo torej, da imamo več rešitev $q(k)$, za katere je prilagojeni referenčni signal $\hat{y}_{ref}(k)$ dosegljiv.

Ustrezno rešitev mora algoritem izbrati z uporabo *nadzorne logike*. To v splošnem pomeni, da, za vsako izmed mogočih rešitev iz množice Q_r za določitev diskretnega dela krmilnega signala, najprej določi še ustrezni zvezni del krmilnega signala, kot je opisano v razdelku 13.3.4. Na ta način dobimo r rešitev krmilnega signala u_K , ki jih algoritem za vodenje v okviru nadzorne logike ovrednoti (glede na predoločeni kriterij) in izbere najustreznejšo izmed njih.

13.3.4 Določitev zveznega dela krmilnega signala

Osnovni zakon, ki ga uporablja krmilni del algoritma za vodenje, lahko izpeljemo z modifikacijo enačb (13.7), (13.3), (13.4), (13.5) in (13.6) v razdelku 13.2, in sicer z naslednjimi zamenjavami.

- Zvezni del vhodnega signala sistema zamenjamo z zveznim delom krmilnega signala, ki ga računamo v trenutnem časovnem koraku oz. smo ga določili v prejšnjih časovnih korakih, kot kaže izraz (13.17).

$$u_{zv}(i) \rightarrow u_{K,zv}(i) \quad (13.17)$$

za $i = k - m + 1, k - m + 2, \dots, k$

- Predikcijo izhoda sistema v naslednjem časovnem koraku zamenjamo z želenim izhodom sistema, tj. s prilagojeno referenčno vrednostjo, kot kaže izraz (13.18).

$$\hat{y}_p(k+1) \rightarrow \hat{y}_{ref}(k) \quad (13.18)$$

- Izhode sistema v prejšnjih časovnih korakih zamenjamo z za en korak zamaknjenimi želenimi izhodi sistema, tj. s prilagojenim referenčnim signalom, kot kaže izraz (13.19).

$$y(i) \rightarrow \hat{y}_{ref}(i-1) \quad (13.19)$$

$$\text{za } i = k - n + 1, k - n + 2, \dots, k$$

- Pri določanju normiranih stopenj pripadnosti izhode sistema zamenjamo s prilagojeno referenčno vrednostjo v prejšnjem časovnem koraku, kot kaže izraz (13.20).

$$\beta_j(y(k)) \rightarrow \beta_j(\hat{y}_{ref}(k-1)) \quad (13.20)$$

Z upoštevanjem omenjenih zamenjav lahko torej izpeljemo osnovni zakon, ki ga uporablja krmilni del algoritma za vodenje, in ga zapišemo v enačbah (13.21), (13.22), (13.23) in (13.24).

$$u_{K,zv}(k) = \frac{\hat{y}_{ref}(k) - S_A - S_B - S_R}{S_U} \quad (13.21)$$

$$S_A = \sum_{i=1}^n \sum_{j=1}^K \beta_j(\hat{y}_{ref}(k-1)) a_{ijd}(q(k)) \hat{y}_{ref}(k-i) \quad (13.22)$$

$$S_B = \sum_{i=2}^m \sum_{j=1}^K \beta_j(\hat{y}_{ref}(k-1)) b_{ijd}(q(k)) u_{K,zv}(k-i+1) \quad (13.23)$$

$$S_R = \sum_{j=1}^K \beta_j(\hat{y}_{ref}(k-1)) r_{jd}(q(k)) \quad (13.24)$$

$$S_U = \sum_{j=1}^K \beta_j(\hat{y}_{ref}(k-1)) b_{1jd}(q(k)) \quad (13.25)$$

Enačbe (13.21), (13.22), (13.23) in (13.24) predstavljajo jedro krmilnega dela algoritma za vodenje. S temi enačbami algoritem v vsakem časovnem koraku k uporabi inverzni hibridni mehki model in na ta način določi ustrezen zvezni del

$u_{K,zv}$ krmilnega signala u_K , ki naj bi dejanski izhod sistema y čim bolj približal referenčnemu signalu \hat{y}_{ref} . Omenimo, da bi bil v idealnem primeru – tj. ko hibridni mehki model popolnoma natančno opisuje dinamiko sistema, ki ga želimo voditi – zaradi kavzalnosti dejanski izhod sistema enak za en časovni korak zakasnjenu referenčnemu signalu ($y_{id}(k) = \hat{y}_{ref}(k - 1)$).

13.4 Regulacijski del algoritma za vodenje

Krmilni del algoritma za vodenje, ki smo ga obravnavali v razdelku 13.3, uporablja hibridni mehki model (glej 5. poglavje) sistema, ki ga želimo voditi, oz. njegov inverzni model, ki je opisan v razdelku 13.2. V vsakem časovnem koraku k določa primerne krmilne signale u_K , ki naj bi dejanski izhod sistema y čim bolj približali referenčnemu signalu y_{ref} oz. prilagojenemu referenčnemu signalu \hat{y}_{ref} (glej razdelek 13.3.2). Pri tem upošteva nelinearno in hibridno dinamiko sistema ter morebitne omejitve vhodov, izhodov in stanj sistema, ki lahko izhajajo iz fizičnih lastnosti sistema ali iz tehnoloških zahtev, ki so nujne za kakovostno vodenje.

Krmilni del algoritma pri delovanju upošteva samo model sistema, ki ga želimo voditi, ne pa tudi dejanskih izhodov (oz. stanj), ki jih lahko merimo. Zaradi tega lahko pri vodenju pride do odstopanj izhoda sistema y od prilagojene referenčne trajektorije \hat{y}_{ref} . Do odstopanj lahko pride zaradi različnih vzrokov, kot so nenačnost hibridnega mehkega modela, šum, morebitne motnje v realnem sistemu, zakasnitve ipd.

Regulacijski del algoritma za vodenje izračuna ustrezne popravke krmilnih signalov u_K , tj. regulirne signale u_R , ki jih določi glede na odstopanje izmerjenega izhodnega signala sistema y od prilagojene referenčne trajektorije \hat{y}_{ref} . Na ta način regulacijski del algoritma dopolnjuje krmilni del v smislu upoštevanja povratnozančnih signalov, kar je v splošnem nujno za kakovostno vodenje.

13.4.1 Linearizacija hibridnega mehkega modela

Hibridni mehki model sistema, ki ga želimo voditi, podaja enačba (13.2). V vsakem časovnem koraku k ga lahko lineariziramo v delovni točki ($y(k)$) in zapišemo v prostoru stanj, kot podaja izraz (13.26).

$$\begin{aligned}x_{lin}(k+1) &= A_{lin}(k) x_{lin}(k) + B_{lin}(k) u_{lin}(k) \\y_{lin}(k) &= C_{lin}(k) x_{lin}(k) + D_{lin}(k) u_{lin}(k)\end{aligned}\tag{13.26}$$

V izrazu (13.26) označuje $x(k) \in \mathbb{R}^n$ stanje sistema, $u(k) \in \mathbb{R}^m$ vhod v sistem, $y(k) \in \mathbb{R}$ pa izhod sistema. S k je označen trenutni časovni korak sistema. Matrike $A_{lin}(k) \in \mathbb{R}^{n \times n}$, $B_{lin}(k) \in \mathbb{R}^{n \times m}$, $C_{lin}(k) \in \mathbb{R}^{1 \times n}$ in $D_{lin}(k) \in \mathbb{R}^{1 \times m}$ določajo dinamiko lineariziranega sistema.

Omenimo, da lahko v obliki izraza (13.26) zapišemo tudi lineariziran model dinamike sistema, ki ga želimo voditi, ki je v osnovni obliki sestavljen iz vzporedno ali zaporedno povezanih hibridnih mehkih modelov.

Glede na predpostavke v razdelku 13.2, lahko vhod lineariziranega modela zapišemo v enačbi (13.27).

$$u_{lin}(k) = \begin{bmatrix} u_{zv,lin}(k) \\ 1 \end{bmatrix}\tag{13.27}$$

Zadnji element vektorja vhoda lineariziranega modela $u_{lin}(k)$ omogoča upoštevanje delovne točke, okrog katere je določen model.

Iz zapisa hibridnega mehkega modela v enačbi (13.2) lahko brez izgube splošnosti ugotovimo, da je izhod lineariziranega sistema y_{lin} v določenem časovnem koraku odvisen samo od stanja x_{lin} , ne pa tudi od vhoda lineariziranega sistema u_{lin} , kar predpišemo v izrazu (13.28).

$$D_{lin}(k) = 0\tag{13.28}$$

13.4.2 Večkoračna predikcija lineariziranega modela

Glede na izpeljavo v razdelku 13.4.1 lahko izraz za enokoračno predikcijo lineariziranega sistema zapišemo, kot podaja enačba (13.29).

$$x_{lin}(k+1 | k) = A_{lin}(k) x_{lin}(k | k) + B_{lin}(k) \begin{bmatrix} u_{zv,lin}(k | k) \\ 1 \end{bmatrix} \quad (13.29)$$

$$y_{lin}(k | k) = C_{lin}(k) x_{lin}(k | k)$$

Izhod procesa računamo pri predpostavki, da so vhodni signali do horizonta konstantni, kot podaja enačba (13.30).

$$u_{zv,lin}(k | k) = u_{zv,lin}(k+1 | k) = \dots = u_{zv,lin}(k+H-1 | k) \quad (13.30)$$

V tem primeru lahko iz izraza za enokoračno predikcijo (13.29) izpeljemo izraz za H -koračno predikcijo lineariziranega sistema, kot ga podaja enačba (13.31).

$$x_{lin}(k+H | k) = A_{lin}^H(k) x_{lin}(k | k) + (A_{lin}^H(k) - I) (A_{lin}(k) - I)^{-1} B_{lin}(k) \begin{bmatrix} u_{zv,lin}(k | k) \\ 1 \end{bmatrix}$$

$$y_{lin}(k+H | k) = C_{lin}(k) x_{lin}(k+H | k) \quad (13.31)$$

V enačbi (13.31) predstavlja $I \in \mathbb{R}^{n \times n}$ enotino matriko ustrezne dimenzije.

13.4.3 Inkrementalni model

Stanje, vhod in izhod sistema lahko zapišemo v obliki vsote, kot podajajo izrazi (13.32), (13.33) in (13.34).

$$x = x_0 + \Delta x \quad (13.32)$$

$$u_{zv} = u_{zv,0} + \Delta u_{zv} \quad (13.33)$$

$$y = y_0 + \Delta y \quad (13.34)$$

V tem primeru lahko enokoračno predikcijo lineariziranega modela zapišemo v enačbi (13.35).

$$\begin{aligned} x_0(k+1 | k) + \Delta x(k+1 | k) = & A_{lin}(k) (x_0(k | k) + \Delta x(k | k)) + \\ & + B_{lin}(k) \begin{bmatrix} u_{zv,0}(k | k) + \Delta u_{zv}(k | k) \\ 1 \end{bmatrix} \end{aligned} \quad (13.35)$$

$$\begin{aligned} y_0(k+1 | k) + \Delta y(k+1 | k) = \\ C_{lin}(k) (x_0(k+1 | k) + \Delta x(k+1 | k)) \end{aligned}$$

Predpišemo, da signali x_0 , $u_{zv,0}$ in y_0 ustrezajo referenčnim oz. krmilnim signalom, ki jih algoritem za vodenje določa v krmilnem delu. Krmilni del algoritma torej zagotavlja, da velja enačba (13.36).

$$x_0(k+1 | k) = A_{lin}(k) x_0(k | k) + B_{lin}(k) \begin{bmatrix} u_{zv,0}(k | k) \\ 1 \end{bmatrix} \quad (13.36)$$

$$y_0(k+1 | k) = C_{lin}(k) x_0(k+1 | k)$$

Če od izraza (13.35) odštejemo izraz (13.36), dobimo inkrementalni model, ki ga podaja izraz (13.37).

$$\Delta x(k+1 | k) = A_{lin}(k) \Delta x(k | k) + B_{lin}(k) \begin{bmatrix} \Delta u_{zv}(k | k) \\ 0 \end{bmatrix} \quad (13.37)$$

$$\Delta y(k+1 | k) = C_{lin}(k) \Delta x(k+1 | k)$$

H -koračno predikcijo inkrementalnega modela podaja enačba (13.38).

$$\begin{aligned} \Delta x(k+H | k) &= A_{lin}^H(k) \Delta x(k | k) + \\ &+ (A_{lin}^H(k) - I) (A_{lin}(k) - I)^{-1} B_{lin}(k) \begin{bmatrix} \Delta u_{zv}(k | k) \\ 0 \end{bmatrix} \end{aligned} \quad (13.38)$$

$$\Delta y(k+H | k) = C_{lin}(k) \Delta x(k+H | k)$$

V izrazu (13.38) označuje $\Delta x(k | k)$ odstopanje stanja sistema od prilagojenega referenčnega stanja v časovnem koraku k , kot podaja enačba (13.39).

$$\Delta x(k | k) = x(k) - \hat{x}_{ref}(k-1) \quad (13.39)$$

Izhod inkrementalnega modela $\Delta y(k+H | k)$, ki ga določimo iz predikcije stanj inkrementalnega modela $\Delta x(k+H | k)$, lahko zapišemo kot vsoto, ki jo podaja enačba (13.40).

$$\Delta y(k+H | k) = \Delta y_u(k+H | k) + \Delta y_x(k+H | k) + \Delta y_m(k+H | k) \quad (13.40)$$

Na podoben način lahko vhod inkrementalnega modela v izrazu (13.38) zapišemo kot vsoto, ki jo podaja enačba (13.41).

$$\Delta u_{zv}(k | k) = u_{R,zv}(k | k) + \Delta u_{zv,m}(k | k) \quad (13.41)$$

V izrazih (13.40) in (13.41) označuje:

- $\Delta y_u(k + H | k)$ predikcijo spremembe izhoda zaradi prispevka regulirnega vhoda $u_{R,zv}(k | k)$;
- $\Delta y_x(k + H | k)$ predikcijo spremembe izhoda zaradi odstopanja stanj v inkrementalnem modelu $\Delta x(k | k)$;
- $\Delta y_m(k + H | k)$ predikcijo spremembe izhoda, ki je posledica nemodelirane dinamike, tj. nenatančnosti modela, motenj v sistemu, šuma ipd.;
- $u_{R,zv}(k | k)$ regulirni signal v časovnem koraku k in
- $\Delta u_{zv,m}(k | k)$ motnjo na vhodu v proces, za katero predpostavimo, da je vzrok za spremembo izhoda zaradi prispevka nemodelirane dinamike $\Delta y_m(k + H | k)$ (tj. nenatančnosti modela, motenj v sistemu, šuma ipd.), kot je označeno v izrazu (13.40).

13.4.4 Določitev regulirnega signala

Osnovna ideja regulacijskega dela algoritma za vodenje je, da se predikcija izhoda inkrementalnega modela s povečevanjem predikcijskih korakov od $h = 1$ do $h = H$ približuje željeni spremembi izhoda procesa čez H korakov, ki je enaka odstopanju izhoda procesa od (zakasnjene) prilagojene reference v trenutnem časovnem koraku. Povezavo podaja enačba (13.42).

$$\Delta y(k + H | k) = (1 - a_r^H) (\hat{y}_{ref}(k - 1) - y(k)) \quad (13.42)$$

V enačbi (13.42) označuje a_r parameter, ki določa hitrost približevanja ($a_r \in [0, 1)$). Dinamiko približevanja opišemo s prenosno funkcijo 1. reda: $\frac{1-a_r}{z-a_r}$.

Regulirni signal lahko v nadaljevanju zapišemo kot vsoto, ki jo podaja enačba (13.43).

$$\Delta u_{R,zv}(k | k) = u_{R,zv,u}(k | k) + u_{R,zv,x}(k | k) + u_{R,zv,m}(k | k) \quad (13.43)$$

Členi v izrazu (13.43) imajo naslednji pomen.

- $u_{R,zv,u}(k | k)$ označuje prispevek regulirnega signala, ki bi v idealnem primeru – tj. ko je $\Delta y_x(k + H | k) = 0$ in $\Delta y_m(k + H | k) = 0$ – povzročil želeno spremembo izhoda $\Delta y(k + H | k)$.

Formulo za izračun $u_{R,zv,u}$ izpeljemo iz enačb (13.38), (13.40), (13.41) in (13.42). Zapisana je v enačbi (13.44).

$$u_{R,zv,u}(k | k) = \frac{(1 - a_r^H) (\hat{y}_{ref}(k - 1) - y(k))}{V_1} \quad (13.44)$$

V_1 , ki nastopa v imenovalcu izraza (13.44), je določen v enačbi (13.45).

$$V_1 = [C_{lin}(k) (A_{lin}^H(k) - I) (A_{lin}(k) - I)^{-1} B_{lin}(k)]_1 \quad (13.45)$$

Zapis $[x]_i$ označuje i -ti element vektorja x .

- $u_{R,zv,x}(k | k)$ označuje prispevek regulirnega signala za kompenzacijo spremembe izhoda $\Delta y_x(k + H | k)$ zaradi odstopanja stanj v inkrementalnem modelu $\Delta x(k | k)$.

Formulo za izračun $u_{R,zv,x}$ izpeljemo iz enačb (13.38), (13.39), (13.40) in (13.41). Zapisana je v enačbi (13.46).

$$u_{R,zv,x}(k | k) = \frac{-C_{lin}(k) A_{lin}^H(k) \Delta x(k | k)}{V_1} \quad (13.46)$$

- $u_{R,zv,m}(k | k)$ označuje prispevek regulirnega signala za kompenzacijo spremembe izhoda $\Delta y_m(k + H | k)$, ki je posledica nemodelirane dinamike, tj. nenatančnosti modela, motenj v sistemu, šuma ipd.

Ocenimo, da je $\Delta u_{zv,m}(k | k) \approx u_{R,zv,u}(k - 1 | k - 1)$ in zapišemo enačbo (13.47).

$$u_{R,zv,m}(k | k) = u_{R,zv,u}(k - 1 | k - 1) \quad (13.47)$$

Na opisani način v vsakem časovnem koraku k določamo regulirni signal u_R . Vhodni signal procesa u izračunamo z enačbo (13.1), seveda pa moramo pri tem upoštevati morebitne omejitve v sistemu.

13.5 Primer uporabe algoritma za vodenje temperature v šaržnem reaktorju

Algoritem za prediktivno vodenje z inverznim modelom smo preizkusili na primeru vodenja šaržnega reaktorja, ki je opisan v 3. poglavju.

13.5.1 Matematični model šaržnega reaktorja

Matematični model šaržnega reaktorja je podan v enačbah v razdelku 3.2. Dinamiko v jedru šaržnega reaktorja opisuje enačba (3.1), dinamiko v plašču pa enačba (3.2).

Masa komponent v jedru šaržnega reaktorja je $m = 600$ kg, masa vode v plašču je $m_w = 40$ kg. Specifična toplota komponent v jedru je $c_w = 4200$ J kg⁻¹ K⁻¹, specifična toplota vode v plašču je $c_w = 4200$ J kg⁻¹ K⁻¹. Pretok vode v plašču znaša $\phi = 1,6$ kg s⁻¹, prevodnost toplotnega stika med jedrom in plaščem znaša $\lambda = 420$ W m⁻² K⁻¹, prevodnost toplotnega stika med plaščem in okolico znaša $\lambda_0 = 84$ W m⁻² K⁻¹, površina toplotnega stika med jedrom in plaščem znaša

$S = 2 \text{ m}^2$, površina toplotnega stika med plaščem in okolico znaša $S_0 = 2 \text{ m}^2$. Temperatura sveže vhodne vode je lahko $T_C = 12 \text{ }^\circ\text{C}$ ali $T_H = 75 \text{ }^\circ\text{C}$.

Pri vodenju tovrstnih šaržnih reaktorjev lahko predpostavimo dve osnovni vrsti motenj na izhodu:

- motnja, ki se prišteje k temperaturi v jedru šaržnega reaktorja T ;
- motnja, ki se prišteje k temperaturi v plašču šaržnega reaktorja T_w .

Zaradi hitrejšje dinamike je pri vodenju precej lažje odpraviti motnjo, ki se prišteje k temperaturi v plašču T_w . V eksperimentu zato predpostavimo samo motnjo, ki se prišteje k temperaturi v jedru T : v trenutku $t = 23000 \text{ s}$ torej temperaturi v jedru dodamo stopničasto motnjo, pri čemer velikost stopnice znaša $\Delta T = -3 \text{ }^\circ\text{C}$.

Algoritem za prediktivno vodenje, ki ga preizkušamo v tem eksperimentu, za delovanje potrebuje primeren matematični model šaržnega reaktorja. V našem primeru smo izbrali hibridni mehki model, katerega razvoj je natančno opisan v razdelku 7.4.1.

13.5.2 Vodenje

Kot smo opisali v razdelku 13.3.2, krmilni del algoritma prilagaja referenčne signale glede na omejitve vhodov, izhodov in stanj sistema, ki lahko izhajajo iz fizičnih lastnosti sistema ali iz tehnoloških zahtev, ki so nujne za kakovostno vodenje. V našem primeru je omejen položaj mešalnega ventila ($0 \leq k_M \leq 1$) in temperaturno območje vode v plašču šaržnega reaktorja ($20 \text{ }^\circ\text{C} \leq T_w \leq 70 \text{ }^\circ\text{C}$).

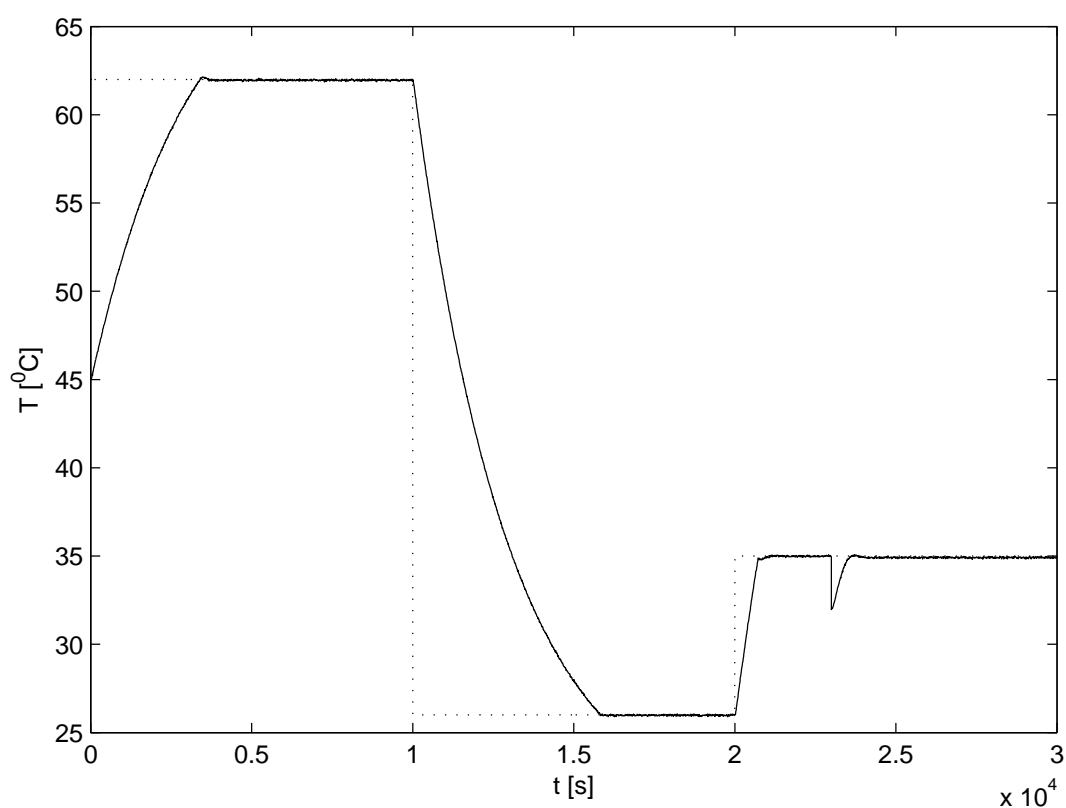
Krmilni del algoritma določa v vsakem časovnem koraku k krmilni signal u_K na način, ki je opisan v razdelku 13.3, regulacijski del pa regulacijski signal u_R na način, ki je opisan v razdelku 13.4.

Glede na predikcijo kršenja temperaturnega območja vode v plašču šaržnega

reaktorja regulacijski del algoritma preklaplja med sledenjem prilagojeni referenčni temperaturi v jedru (\hat{T}_{ref}) oz. v plašču šaržnega reaktorja ($\hat{T}_{w,ref}$).

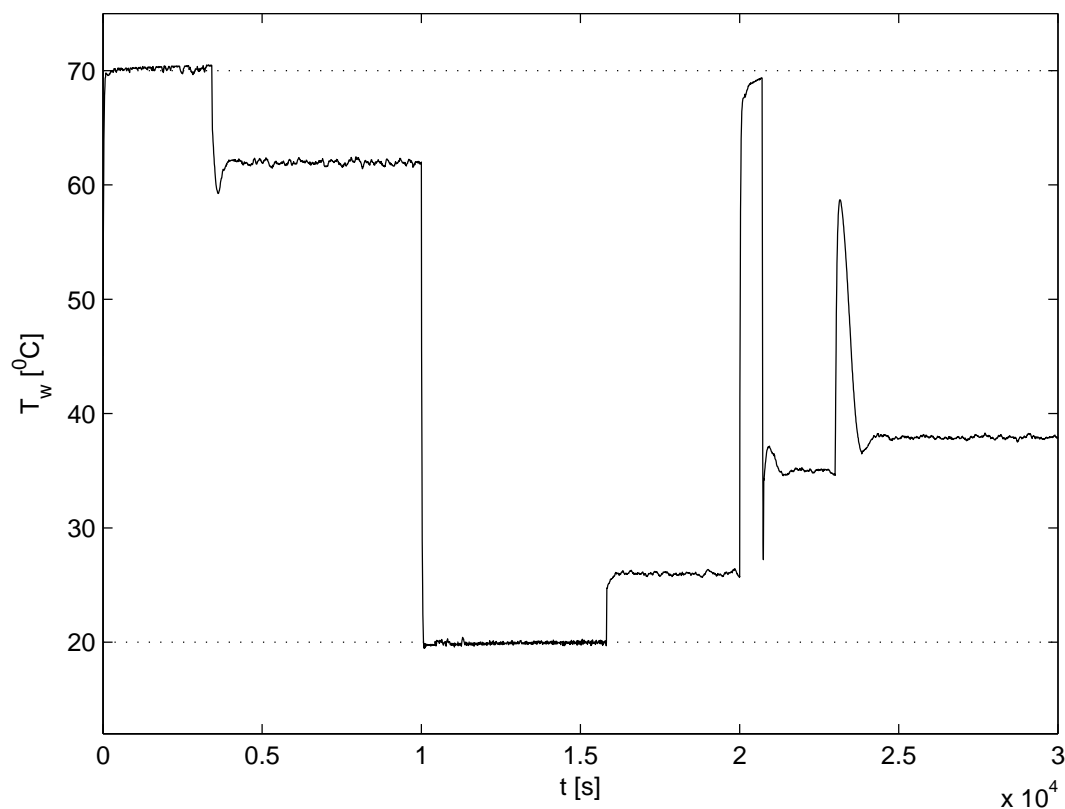
V primeru sledenja prilagojeni referenčni temperaturi v jedru (\hat{T}_{ref}) so parametri regulacijskega dela algoritma $H = 50$ in $a_r = 0$, v primeru sledenja prilagojeni referenčni temperaturi v plašču ($\hat{T}_{w,ref}$) pa $H = 5$ in $a_r = 0$.

13.5.3 Rezultati



Slika 13.2: Temperatura v jedru reaktorja T (polna črta) in referenčna temperatura T_{ref} (prekinjena črta).

Rezultati eksperimenta so prikazani na slikah 13.2, 13.3 in 13.4. Slika 13.2 prikazuje potek temperature v jedru šaržnega reaktorja T in referenčne temperature T_{ref} . Slika 13.3 prikazuje potek temperature v plašču šaržnega reaktorja T_w . Slika 13.4 prikazuje krmilni in regulacijski del signala za položaj mešalnega ventila $k_{M,K}$ in $k_{M,R}$ ter signala za položaj zapornih ventilov k_H in k_C .



Slika 13.3: Temperatura v plašču reaktorja T (polna črta) in omejitev temperaturnega območja (prekinjena črta).

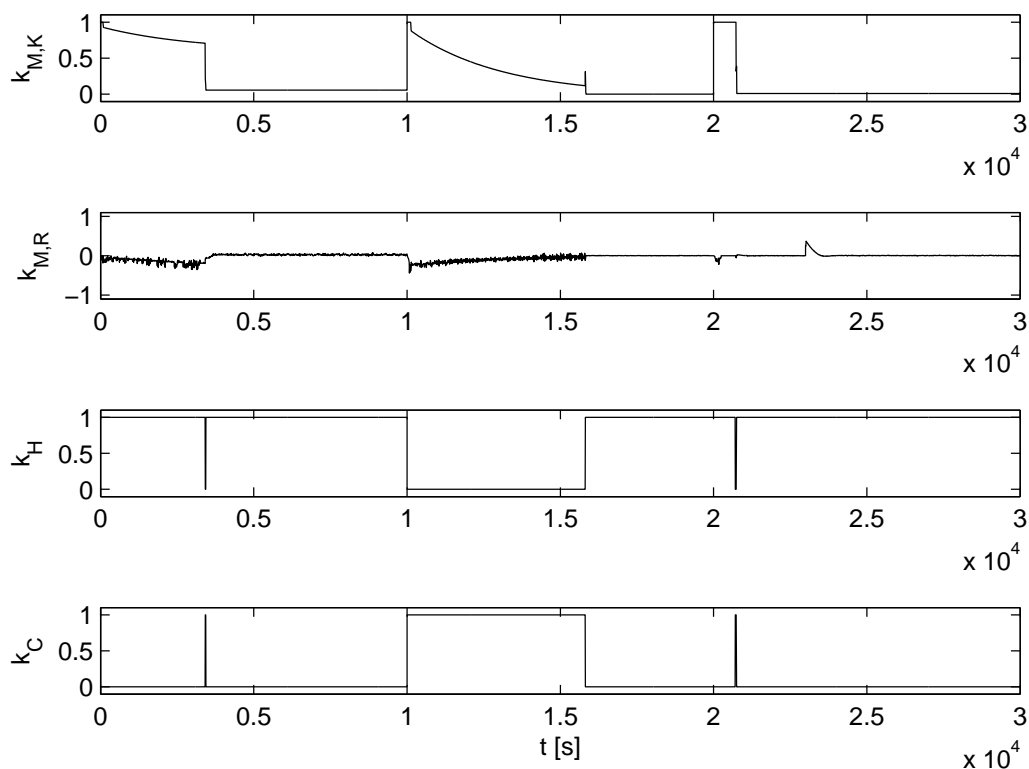
13.5.4 Komentar

Uporabnost algoritma za vodenje z inverznim modelom smo preverili na primeru vodenja šaržnega reaktorja.

Ugotovili smo, da temperatura v jedru reaktorja T dobro sledi referenčni temperaturi T_{ref} , poleg tega pa algoritem tudi ustrezno kompenzira stopničasto motnjo na izhodu, ki se zgodi v trenutku $t = 23000$ s. Tudi temperatura v plašču T_w ostaja znotraj predpisanega temperaturnega območja.

Osnovna ideja obravnavanega pristopa je dvodelnost algoritma za vodenje. Algoritem je sestavljen iz krmilnega in regulacijskega dela.

Z uporabo hibridnega mehkega modela krmilni del pri določanju krmilnih signalov u_K upošteva nelinearno in hibridno dinamiko sistema, pa tudi morebitne



Slika 13.4: Krmilni in regulacijski del signala za položaj mešalnega ventila $k_{M,K}$ in $k_{M,R}$ ter signala za položaj zapornih ventilov k_H in k_C .

omejitve vhodov, izhodov in stanj sistema. Glavna prednost predlaganega pristopa je, da lahko krmilni del algoritma za vodenje izhod sistema y pripelje v bližino referenčne trajektorije y_{ref} , kar omogoča uporabo relativno preprostega regulacijskega dela, s katerim zagotovimo kakovostno vodenje. Glede na to, da regulacijski del algoritma za izračun regulirnega signala u_R upošteva izmerjeni izhodni signal sistema y , predstavlja dopolnitev krmilnega dela algoritma za vodenje, saj na ta način kompenzira odstopanja izhoda sistema y od referenčne trajektorije y_{ref} . Do odstopanj lahko pride zaradi različnih vzrokov, kot so nenačnost hibridnega mehkega modela, šum, morebitne motnje v realnem sistemu ipd.

Rezultati so torej pokazali, da uporaba algoritma omogoča izpolnitev zahtev vodenja. Temperatura v jedru zadovoljivo sledi referenčnemu signalu, kar zagota-

vlja kratek čas obdelave posamezne šarže. Poleg tega algoritem določa krmilni in regulirni zakon v eksplicitni obliki, zato je računska zahtevnost izvedbe majhna, posebno v primerjavi z optimizacijskimi pristopi. Obravnavani algoritem za implementacijo torej ne zahteva drage strojne opreme.

Del IV

Zaključek

14. Zaključek

V doktorski disertaciji smo obravnavali problematiko prediktivnega vodenja hibridnih sistemov. Opisali smo nekatere težave in pomanjkljivosti, ki se pojavljajo na tem področju, in poskušali poiskati ustrezne rešitve, jih ovrednotiti in njihovo uporabnost preveriti na simulacijskem primeru šaržnega reaktorja.

V nadaljevanju povzemamo prispevke disertacije.

- Razvili smo formulacijo hibridnega mehkega modela, ki je primerna za opisovanje sistemov, ki izkazujejo hibridno in nelinearno dinamiko, in jo lahko uporabimo pri prediktivnem vodenju. Obravnavali smo strukturo modela in njegov zapis ter opisali možnost posplošitve na multivariabilne sisteme in sisteme višjega reda.
- Predlagali smo metodo za identifikacijo hibridnih mehkih modelov. Razvili smo pristop z mehkim rojenjem podatkov za identifikacijo v vhodno-izhodnem prostoru hibridnega mehkega modela in predlagali metodo za projiciranje mehkih rojev v vhodni prostor hibridnega mehkega modela. Pristop smo preizkusili na primeru šaržnega reaktorja. Ovrednotili smo natančnost predikcijskega modela in preverili njeno odvisnost od števila rojev, ki jih določamo pri identifikaciji.
- Metodo za identifikacijo hibridnih mehkih modelov smo priredili za odsekoma afine modele in jo preizkusili na primeru šaržnega reaktorja. Ovrednotili smo natančnost predikcijskega modela in preverili njeno odvisnost od števila rojev, ki jih določamo pri identifikaciji.

S prirejeno metodo za identifikacijo smo dobili stabilen odsekoma afini model, ki zadovoljivo opisuje dinamiko šaržnega reaktorja, pri uporabi klasične identifikacijske metode za odsekoma afine sisteme pa stabilnega modela nismo mogli dobiti. Vzrok za to je nezadostno število podatkov za identifikacijo, še bolj pomembna pa je enakomernost porazdelitve podatkov v vhodno-izhodnem prostoru modela, ki je v realnih primerih večinoma ne moremo zagotoviti, zato se je metoda izkazala za uporabno tudi za identifikacijo odsekoma afinih modelov.

- Primerjali smo natančnost odsekoma afinega modela in hibridnega mehkega modela in ugotovili, da – ob primerljivi zahtevnosti pristopa za modeliranje in identifikacijo ter kompleksnosti na primerljiv način dobljenega končnega modela – dosežemo večjo natančnost s hibridnim mehkim modelom. Za primerljivo natančnost moramo torej uporabiti odsekoma afini model, ki je kompleksnejši od hibridnega mehkega modela, kar pa lahko bistveno poveča računsko zahtevnost algoritma za prediktivno vodenje.
- Predstavili smo metodo za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi, ki deluje na osnovi analize dosegljivosti. Vpeljali smo splošno uporabno obliko kriterijske funkcije. Raziskali smo računsko zahtevnost algoritma in predlagali možnosti za njeno zmanjšanje z uporabo analize dosegljivosti, omejitvijo nabora mogočih kombinacij vrednosti vektorja vhoda in zadrževanjem vrednosti vektorja vhodov skozi več časovnih korakov.

Pristop smo preizkusili na primeru vodenja šaržnega reaktorja. Pokazali smo, da je bistvenega pomena določitev ustrezne kriterijske funkcije.

Primerjali smo vodenje s hibridnim mehkim modelom in hibridnim linearnim modelom. Hibridni mehki model nam omogoča bolj kakovostno vodenje, kar se močneje odraža predvsem pri sistemih z izrazito nelinearno dinamiko.

- Opisali smo metodo za prediktivno vodenje hibridnih sistemov na osnovi

reševanja mešanih celoštevilskih optimizacijskih problemov, primerno za sisteme, ki jih lahko zapišemo v obliki odsekoma afinega ali ekvivalentnega modela. Pristop smo preizkusili na primeru vodenja temperature v šaržnem reaktorju in ugotovili, da uporaba zaradi kompleksnosti ni smiselna.

- Predstavili smo metodo za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi, ki deluje na osnovi genetskega algoritma. Obravnavali smo računsko zahtevnost algoritma in predlagali možnosti za njeno zmanjšanje z omejitvijo nabora mogočih kombinacij vrednosti vektorja vhoda in zadrževanjem vrednosti vektorja vhodov skozi več časovnih korakov.

Pristop smo preizkusili na primeru vodenja šaržnega reaktorja, in sicer za različne največje predikcijske horizonte, število osebkov in število generacij in primerjali kakovost vodenja in računsko zahtevnost.

Glavna slabost genetskega algoritma je suboptimalnost rešitve optimizacijskega problema in s tem negotovi rezultati pri neustrezni nastavitvi parametrov algoritma. Primerjava genetskega algoritma z algoritmoma na osnovi analize dosegljivosti in na osnovi ocenjevanja vseh mogočih rešitev je pokazala, da je rezultat vodenja, tj. zaprtozančni odziv – kljub suboptimalnosti genetskega algoritma – lahko praktično identičen odzivu pri vodenju z eno izmed optimalnih metod, pri tem pa je računsko zahtevnost bistveno manjša.

Neobčutljivost na spremembe reference ter predvidljivost in omejenost časa za izračun rešitve v vsakem koraku predstavljajo glavno prednost pristopa na osnovi genetskega algoritma. Predvidljivost časa za izračun rešitve pa olajša težave pri uporabi ali celo sploh omogoči zanesljivo delovanje algoritma za prediktivno vodenje v realnem času.

- Predstavili smo metodo za adaptivno prediktivno vodenje sistemov s časovno spremenljivimi dinamičnimi lastnostmi, ki deluje na osnovi identifi-

kacije lokalnega linearnega modela vodenega sistema. Opisali smo vpeljavo pomožnega vhoda za linearizacijo.

Metodo smo preizkusili na primeru vodenja šaržnega reaktorja. V eksperiment smo vključili tudi dve eksotermni kemični reakciji, ki potekata v jedru šaržnega reaktorja in povzročata časovno spremenljivost dinamike celotnega sistema. Čeprav sta reakciji, ki v model procesa vpeljujeta časovno spreminjajoče parametre, izrazito eksotermni, so rezultati pokazali, da uporaba algoritma omogoča izpolnitev zahtev vodenja, in to kljub temu, da kinetičnega modela reakcij ne poznamo.

Algoritem določa regulirni zakon v eksplicitni obliki, zato je računsko zahtevnost izvedbe majhna, posebno v primerjavi z optimizacijskimi pristopi. Algoritem za implementacijo torej ne zahteva drage strojne opreme.

- Razvili smo metodo za vodenje z inverznim modelom, ki je primerna za sisteme z enim izhodom, ki jih lahko zapišemo v obliki hibridnega mehkega modela in izpeljali inverzni hibridni mehki model.

Algoritem za vodenje je sestavljen iz dveh delov, in sicer iz krmilnega in regulacijskega dela. Z uporabo hibridnega mehkega modela oz. inverznega hibridnega mehkega modela krmilni del pri določanju krmilnih signalov upošteva nelinearno in hibridno dinamiko sistema, pa tudi morebitne omejitve vhodov, izhodov in stanj sistema. Omejitve lahko izhajajo iz fizičnih lastnosti sistema ali iz tehnoloških zahtev, ki so nujne za kakovostno vodenje. Glavna prednost predlaganega pristopa je, da lahko krmilni del algoritma za vodenje izhod sistema pripelje v bližino prilagojene referenčne trajektorije, kar omogoča uporabo relativno preprostega regulacijskega dela, s katerim zagotovimo kakovostno vodenje. Regulacijski del algoritma za izračun regulirnega signala upošteva povratnozančni izmerjeni izhodni signal sistema, zato predstavlja dopolnitev krmilnega dela algoritma za vodenje. Na ta način kompenzira odstopanja izhoda sistema od prilagojene referenčne

trajektorije, do katerih lahko pride zaradi različnih vzrokov, kot so nena-
tančnost hibridnega mehkega modela, šum, morebitne motnje v realnem
sistemu ipd.

Pristop smo preizkusili na primeru vodenja šaržnega reaktorja. Ugotovili
smo, da temperatura v jedru reaktorja dobro sledi referenčni temperaturi,
poleg tega pa algoritem tudi ustrezno kompenzira stopničasto motnjo na
izhodu. Tudi temperatura v plašču ostaja znotraj predpisanega tempera-
turnega območja.

Algoritem določa krmilni in regulirni zakon v eksplicitni obliki, zato je
računska zahtevnost izvedbe majhna, posebno v primerjavi z optimizacij-
skimi pristopi. Obravnavani algoritem za implementacijo torej ne zahteva
drage strojne opreme.

- V slovenščino smo vpeljali nekaj novih terminov s področja prediktivnega
vodenja hibridnih sistemov, ki jih do sedaj še ni bilo ali pa so obstajali samo
v angleškem jeziku.

Zaključimo lahko, da na področju prediktivnega vodenja hibridnih sistemov
še nimamo teorije, ki bi bila uporabna za vodenje splošnega hibridnega sistema.

Hibridni sistemi namreč predstavljajo zelo široko področje najrazličnejših sis-
temov, ki so dinamično bistveno zahtevnejši od npr. linearnih sistemov, za katere
obstaja splošna teorija vodenja, ki je uporabna tudi v praksi. Zato se zdi, da je do
splošne teorije za vodenje hibridnih sistemov še daleč, morda pa zaradi komple-
ksnosti sploh nima smisla obravnavati vodenja vseh tipov in podtipov hibridnih
sistemov v okviru enotne teorije.

Najboljši približek enotne teorije za prediktivno vodenje hibridnih sistemov je
pristop z reševanjem mešanih celoštevilskih optimizacijskih problemov. Omejen
je na vodenje sistemov, ki so formulirani kot odsekoma afini in njim ekvivalen-
tni modeli. Problem vodenja je mogoče pretvoriti v (*multiparametrični*) *mešani*

celoštevilski optimizacijski problem, ki ga lahko rešujemo s (komercialnimi) programi za reševanje optimizacijskih problemov. Pristop z reševanjem mešanih celoštevilskih optimizacijskih problemov je bil večkrat uporabljen tudi v praksi. Kljub temu pa v mnogih primerih s kompleksnejšo dinamiko in večjimi predikcijskimi horizonti žal ostaja samo na ravni teorije, saj njegovo uporabo preprečuje prevelika računska zahtevnost.

Pri optimizacijskih pristopih za prediktivno vodenje hibridnih sistemov se v realnih primerih pogosto ne moremo izogniti *prekletstvu razsežnosti* – problem je teoretično sicer rešljiv, v praksi pa ga zaradi omejene računske moči ni mogoče rešiti. Pri vodenju hibridnih sistemov je torej smiselno uporabiti tudi teorijo vodenja z drugih področij in jo s primernimi modifikacijami in ustreznimi poenostavitvami prilagoditi vodenju konkretnega sistema ter se na ta način izogniti algoritmom, ki so za praktično uporabo računsko prezahtevni.

14.1 Glavni izvirni prispevki doktorske disertacije

- Formulacija hibridnega mehkega modela, ki je primerna za opisovanje sistemov s hibridno in nelinearno dinamiko in jo lahko uporabimo pri prediktivnem vodenju.
- Metoda identifikacije hibridnih mehkih modelov z mehkim rojenjem podatkov za identifikacijo v vhodno-izhodnem prostoru modela.
- Razvoj metode za prediktivno vodenje hibridnih sistemov z diskretnimi vhodi, ki deluje na osnovi genetskega algoritma.
- Razvoj metode za adaptivno prediktivno vodenje sistemov s časovno spremenljivimi dinamičnimi lastnostmi, ki deluje na osnovi linearizacije hibridnega modela vodenega sistema.
- Formulacija inverznega hibridnega mehkega modela in razvoj metode za vodenje z inverznim modelom.

Literatura

- [1] K. J. Åström, B. Wittenmark. *Adaptive control*. Electrical Engineering: Control Engineering. Addison-Wesley Publishing Company, second edition, 1995.
- [2] R. Alur, T. A. Henzinger, E. D. Sontag, uredniki. *Hybrid Systems III: Verification and Control*, Volume 1066 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Heidelberg, 1996.
- [3] P. J. Antsaklis. A brief introduction to the theory and applications of hybrid systems. *Proceedings of the IEEE*, Volume 88, 2000.
- [4] P. J. Antsaklis, W. Kohn, A. Nerode, S. Sastry, uredniki. *Hybrid Systems II*, Volume 999 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Heidelberg, 1995.
- [5] P. J. Antsaklis, W. Kohn, A. Nerode, S. Sastry, uredniki. *Hybrid Systems IV*, Volume 1273 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Heidelberg, 1997.
- [6] P. J. Antsaklis, W. Kohn, A. Nerode, S. Sastry, uredniki. *Hybrid Systems V*, Volume 1567 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Heidelberg, 1999.
- [7] P. J. Antsaklis, X. D. Koutsoukos. *Software-Enabled Control*, poglavje Hybrid Systems: Review and Recent Progress, str. 273–298. IEEE Press, 2005. DOI: 10.1002/047172288X.

-
- [8] P. J. Antsaklis, J. A. Stiver, M. D. Lemmon. *Hybrid systems*, Volume 736 of *Lecture Notes in Computer Science*, poglavje Hybrid sistem modeling and event identification, str. 366–392. Springer Verlag, Berlin, Heidelberg, 1993.
- [9] R. Babuška. *Fuzzy modeling and identification*. Technische Universiteit Delft, 1996.
- [10] R. Babuška. *Fuzzy Modelling for Control*. KAP, 1998.
- [11] R. Babuška, J. Sousa, H. Verbruggen. Predictive control of nonlinear systems based on fuzzy and neural models. str. 667. European Control Conference, 1999.
- [12] R. Babuška, H. B. Verbruggen. An overview of fuzzy modelling for control. *Control Engineering Practice*, Vol. 4, No. 11, str. 1593–1606, 1996.
- [13] A. Back, J.hn Guckenheimer, M. Myers. *Hybrid systems*, Volume 736 of *Lecture Notes in Computer Science*, poglavje A dynamical simulation facility for hybrid systems, str. 255–267. Springer Verlag, Berlin, Heidelberg, 1993.
- [14] T. Back, D. Fogel, Z. Michalewics, uredniki. *Advanced Algorithms and Operators*. Institute of Physics Publishing, Bristol, UK, 2000.
- [15] M. Baotić, F. Christophersen, M. Morari. Infinite time optimal control of hybrid systems with a linear performance index. *Proc. of the Conf. on Decision and Control*, Maui, 2003.
- [16] M. Baotić, F. Christophersen, M. Morari. A new algorithm for constrained finite time optimal control of hybrid systems with a linear performance index. University of Cambridge, UK, 2003. European Control Conference.

-
- [17] A. G. Beccuti, T. Geyer, M. Morari. Temporal lagrangian decomposition of model predictive control for hybrid systems. str. 2509–2514. European Control Conference, IEEE, 2003.
- [18] A. Bemporad. Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form. *IEEE Trans. Automatic Control*, Vol. 49, No. 5, str. 832–838, 2004.
- [19] A. Bemporad. Hybrid Toolbox - User's Guide, 2004.
- [20] A. Bemporad. Model-based predictive control design: New trends and tools. *Proc. of 45th IEEE Conference on Decision and Control*, San Diego, 2006. IEEE.
- [21] A. Bemporad. Modeling and control of hybrid dynamical systems: The hybrid toolbox for matlab. Inge Troch, Felix Breitenecker, uredniki, *MATHTMOD Vienna 09: Proceedings; Full Papers CD Volume*, Number 35 in ARGESIM Report, str. 82–100, 2009.
- [22] A. Bemporad, P. Borodani, M. Mannelli. *Hybrid Systems: Computation and Control*, poglavje Hybrid control of an automotive robotized gearbox for reduction of consumptions and emissions, str. 81–96. Number 2623 in Lecture Notes in Computer Science. Springer-Verlag, 2003.
- [23] A. Bemporad, F. Borrelli, M. Morari. Explicit solution of lp-based model predictive control. *Proc. 39th IEEE Conference on Decision and Control*, Sydney, 2000.
- [24] A. Bemporad, F. Borrelli, M. Morari. Optimal controllers for hybrid systems. *Proc. 39th IEEE Conference on Decision and Control*, Sydney, 2000.
- [25] A. Bemporad, F. Borrelli, M. Morari. Piecewise linear optimal controllers for hybrid systems. *Proceedings of American Control Conference*, str. 1190–1194, 2000.

-
- [26] A. Bemporad, F. Borrelli, M. Morari. On the optimal control law for linear discrete time hybrid systems. *Hybrid Systems: Computation and Control: Lecture Notes in Computer Science*, Vol. 2289, str. 105–119, 2002.
- [27] A. Bemporad, S. Di Cairano, I. V. Kolmanovsky. Hybrid modeling and control of a multibody magnetic actuator for automotive applications. *Proc. 46th IEEE Conf. on Decision and Control*, str. 5270—5275, New Orleans, 2007.
- [28] A. Bemporad, G. Ferrari-Trecate, M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control*, Vol. 45, No. 10, str. 1846–1876, 2000.
- [29] A. Bemporad, N. Giorgetti. Logic-based methods for optimal control of hybrid systems. *IEEE Trans. Autom. Control*, Vol. 51, No. 6, str. 963–976, 2006.
- [30] A. Bemporad, L. Giovanardi, F. D. Torrisi. Performance driven reachability analysis for optimal scheduling and control of hybrid systems. *Proc. 39th IEEE Conf. Decision and Control*, str. 969–974, Sydney, 2000.
- [31] A. Bemporad, W. Heemels, B. De Schutter. On hybrid systems and closed-loop MPC system. *IEEE Transactions on Automatic Control*, Vol. 47, No. 5, str. 863–869, 2002.
- [32] A. Bemporad, M. Morari. Control of systems integrating logic, dynamics and constraints. *Automatica*, Vol. 35, No. 3, str. 407–427, 1999.
- [33] A. Bemporad, M. Morari. Optimization-based hybrid control tools. *Proc. ACC 2001 American Control Conf.*, str. 1689–1703, Arlington, 2001.
- [34] A. Bemporad, M. Morari, V. Dua, P. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, Vol. 38, No. 1, str. 3–20, 2002.

-
- [35] A. Bemporad, F. Torrisi, M. Morari. *Hybrid systems: Computation and control*, Volume 1790 of *Lecture Notes in Computer Science*, poglavje Optimization-based verification and stability characterization of piecewise affine and hybrid systems, str. 45–58. Springer-Verlag, Berlin, Heidelberg, 2000.
- [36] F. Borrelli. Constrained optimal control of linear and hybrid systems. *Lectures Notes in Control and Information Sciences*, , , 2003.
- [37] F. Borrelli, M. Baotić, A. Bemporad, M. Morari. An efficient algorithm for computing the state feedback solution to optimal control of discrete time hybrid systems. str. 4717–4722, Denver, Colorado, USA, 2003. American Control Conference.
- [38] F. Borrelli, M. Baotić, A. Bemporad, M. Morari. Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica*, Vol. 41, No. 10, , 2005.
- [39] F. Borrelli, A. Bemporad, M. Fodor, D. Hrovat. An mpc/hybrid system approach to traction control. *IEEE Trans. Contr. Systems Technology*, Vol. 14, No. 3, str. 541—552, 2006.
- [40] M. S. Branicky. *Studies in hybrid systems: modeling, analysis, and control*. Doktorska disertacija, Masechusetts institute of technology, 1995.
- [41] R. W. Brockett. *Hybrid models for motion control systems*, str. 29–53. Essays in Control: Perspectives in the Theory and its Applications. Birkhäuser, Boston, 1993.
- [42] A. E. Bryson, Y.-C. Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. Taylor and Francis, 1975.
- [43] S. Di Cairano, A. Bemporad, I. V. Kolmanovsky, D. Hrovat. Model predictive control of magnetically actuated mass spring dampers for automotive

- applications. *Int. J. Control*, Vol. 80, No. 11, str. 1701—1716, 2007.
- [44] S. Di Cairano, D. Yanakiev, A. Bemporad, I. V. Kolmanovsky, D. Hrovat. An mpc design flow for automotive control and applications to idle speed regulation. *Proc. 47th IEEE Conf. on Decision and Control*, str. 5686—5691, Cancun, 2008.
- [45] E. F. Camacho, C. Bordons. *Model predictive control*. Advanced Textbooks in Control and Signal Processing. Springer-Verlag, London, 1998.
- [46] J. Castro. Fuzzy logic controllers are universal approximators. *IEEE Trans. System Man Cybernet*, Vol. 25, str. 629–635, 1995.
- [47] J. Causa, G. Karer, A. Núñez, D. Sáez, I. Škrjanc. Hybrid fuzzy predictive control based on genetic algorithms for the temperature control of a batch reactor. *Computers and chemical engineering*, Vol. 32, No. 12, str. 3254–3263, 2008.
- [48] J. S. Chang, W. Y. Hseih. Optimization and control of semibatch reactors. *Industrial and Engineering Chemistry Research*, Vol. 34, str. 545–556, 1995.
- [49] L. Chen, G. Bastin, V. Van Breusegam. A case study of adaptive nonlinear regulation of fed-batch reactors. *Automatica*, Vol. 31, str. 55–65, 1995.
- [50] W. Cho, T. F. Edgar, J. Lee. Iterative learning dual-mode control of exothermic batch reactors. *2005 5th Asian Control Conference*, str. 1270–1275, 2005.
- [51] J. P. Coelho, P. B. de Moura Oliveira, J. Boaventura Cunha. Greenhouse air temperature predictive control using the particle swarm optimisation algorithm. *Computers and Electronics in Agriculture*, Vol. 49, str. 330–344, 2005.
- [52] C. Coello. Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer*

-
- Methods in Applied Mechanics and Engineering*, Vol. 191(11-12), str. 1245–1287, 2002.
- [53] B. J. Cott, S. Macchietto. Temperature control of exothermic batch reactors using generic model control. *Industrial and Engineering Chemistry Research*, Vol. 28, str. 1177–1184, 1989.
- [54] J. E. Cuthrell, L. T. Biegler. Simultaneous optimization and solution methods for batch reactor control. *Computers and Chemical Engineering*, Vol. 13, str. 49–69, 1998.
- [55] B. De Schutter. Optimal control of a class of linear hybrid systems with saturation. *SIAM Journal of applied Control and Optimization*, Vol. 39, No. 3, str. 835–851, 2000.
- [56] B. De Schutter, B. De Moor. *Hybrid systems V*, Volume 1567 of *Lecture Notes in Computer Science*, poglavje The extended linear complementary problem and the modeling and analysis of hybrid systems, str. 70–85. Springer-Verlag, Berlin, Heidelberg, 1999.
- [57] B. De Schutter, T. Vand den Boom. On model predictive control for max-min-plus-scaling discrete event systems. Tehnično poročilo bds:00-04, Control Lab, Fac. ITS, Delft Univ. Techn., Delft, 2000.
- [58] S. Engell, G. Frehse, E. Scnieder, uredniki. *Modelling, Analysis and Design of Hybrid Systems*, Volume 279 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin, Heidelberg, 2002.
- [59] J. Espinosa, J. Vandewalle. Predictive control using fuzzy models applied to a steam generating unit. str. 151–160. 3rd International FLINS Workshop on Fuzzy Logic and Intelligent Technologies for Nuclear Science Industry, 1998.

-
- [60] J. Espinosa, J. Vandewalle. Predictive control using fuzzy models - comparative study. str. 273. European Control Conference, EUCA, 1999.
- [61] J. Espinosa, J. Vandewalle, V. Wertz. *Fuzzy Logic, Identification and Predictive Control*. Spriger-Verlag, 2005.
- [62] G. Ferrari-Trecate. Hybrid Identification Toolbox (HIT), 2005.
- [63] G. Ferrari-Trecate, M. Muselli, D. Liberati, M. Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, Vol. 39, No. 2, str. 205–217, 2003.
- [64] B. A. Foss, T. A. Johansen, A. V. Sorensen. Nonlinear predictive control using local models-applied to a batch fermentation process. *Control Engineering Practice*, Vol. 3, str. 389–396, 1995.
- [65] N. Giorgetti, A. Bemporad, H. E. Tseng, D. Hrovat. Hybrid model predictive control application towards optimal semi-active suspension. *Int. J. Control*, Vol. 79, No. 5, str. 521—533, 2006.
- [66] N. Giorgetti, G. Ripaccioli, A. Bemporad, I. V. Kolmanovsky, D. Hrovat. Hybrid model predictive control of direct injection stratified charge engines. *IEEE/ASME Transactions on Mechatronics*, Vol. 11, No. 5, str. 499–506, 2006.
- [67] D. Girimonte, R. Babuška. Structure for nonlinear models with mixed discrete and continuous inputs: a comparative study. *Proc. of IEEE International Conf. on system, Man and Cybernetics*, str. 2392–2397, 2004.
- [68] F. Girosi, T. Poggio. Networks and the best approximation property. *Biological Cybernetics*, Vol. 63, str. 169–176, 1990.
- [69] P. Grieder, M. Kvasnica, M. Baotić, M. Morari. Low complexity control of piecewise affine systems with stability guarantee. *American Control Conference*, Boston, 2004.

-
- [70] R. L. Grossman, A. Nerode, A. P. Ravn, H. Rischel, uredniki. *Hybrid Systems*, Volume 736 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Heidelberg, 1993.
- [71] M. Hadjili, V. Wertz. Generalized predictive control using Takagi-Sugeno fuzzy models. str. 405–410. IEEE International Symposium on Intelligent Control, Intelligent Systems & Semiotics, IEEE, 1999.
- [72] W. P. M. H. Heemels, B. De Schutter, A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, Vol. 37, No. 7, str. 1085–1091, 2001.
- [73] W. P. M. H. Heemels, J. M. Schumacher, S. Weiland. Linear complementarity systems. *SIAM Journal of applied Mathematics*, Vol. 60, No. 4, str. 1234–1269, 2000.
- [74] T. A. Henzinger, S. Sastry, uredniki. *Hybrid Systems: Computation and Control*, Volume 1386 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Heidelberg, 1998.
- [75] K. H. Johansson, M. Egerstedt, J. Lygeros, S. Sastry. On the regularization of zeno hybrid automata. *System and Control Letters*, Vol. 38, str. 141–150, 1999.
- [76] M. Johansson, A. Rantzer. Computation of piecewise quadratic lypunov functions for hybrid systems. *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, str. 555–559, 1998.
- [77] R. Johansson, urednik. *Nonlinear and Hybrid Systems in Automotive Control*. Springer-Verlag, London, 2003.
- [78] G. Karer. Prediktivno vodenje sistemov z diskretnimi vhodi na osnovi analize dosegljivosti: diplomsko delo, 2004.

-
- [79] G. Karer, G. Mušič, I. Škrjanc, B. Zupančič. Hybrid fuzzy model-based predictive control of temperature in a batch reactor. *Computers and Chemical Engineering*, Vol. 31, str. 1552–1564, 2007.
- [80] G. Karer, G. Mušič, I. Škrjanc, B. Zupančič. Feedforward control of a class of hybrid systems using an inverse model. I. Troch, F. Breitenecker, uredniki, *MATHMOD Vienna 09: Proceedings; Full Papers CD Volume*, Number 35 in ARGESIM Report, str. 1687–1697, 2009.
- [81] G. Karer, I. Škrjanc, B. Zupančič. Self-adaptive predictive functional control of the temperature in an exothermic batch reactor. *Chemical Engineering and Processing*, Vol. 47, str. 2379–2385, 2008.
- [82] J. Kennedy, R. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, 2001.
- [83] E. C. Kerrigan, D. Q. Mayne. Optimal control of constrained, piecewise affine systems with bounded disturbances. *Proc. 41st IEEE Conference on Decision and Control*, Las Vegas, 2002.
- [84] J. Kim, U. Huh. Fuzzy model based predictive control. str. 405–409. IEEE International Conference on Fuzzy Systems, IEEE, 1998.
- [85] S. Kimbrough, G. Koehler, M. Lu, D. Wood. On a feasible-infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch. *European Journal of Operational Research*, Vol. 190(2-16), str. 310–327, 2008.
- [86] M. Klopčič. *Uporaba genetskih algoritmov na področju vodenja*. Magistrsko delo, Fakulteta za elektrotehniko, Univerza v Ljubljani, 1997.
- [87] M. Kvasnica, P. Grieder, M. Baotić. *Multi-Parametric Toolbox (MPT)*, 2004.

-
- [88] N. Lakshmanan, Y. Arkun. Estimation and control of batch processes using multiple models. *International Journal of Control*, Vol. 72, No. 7/8, str. 659–675, 1999.
- [89] M. Lazar, M. Heemels, S. Weiland, A. Bemporad. Stability of hybrid model predictive control. *IEEE Trans. Autom. Control*, Vol. 51, No. 11, str. 1813–1818, 2006.
- [90] M. Lazar, M. Heemels, S. Weiland, A. Bemporad. Stabilizing model predictive control of hybrid systems. *IEEE Transactions on Automatic Control*, Vol. 51, No. 11, str. 1813–1818, 2006.
- [91] C. Loeblin, J. D. Perkins, B. Srinivasan, D. Bonvin. Economic performance analysis in the design of on line batch optimization system. *Journal of Process Control*, Vol. 9, str. 61–78, 1999.
- [92] R. Luus, O. N. Okongwu. Towards practical optimal control of batch reactors. *Chemical Engineering Journal*, Vol. 75, str. 1–9, 1997.
- [93] J. M. Maciejowski. *Predictive control: with constraints*. Prentice Hall, Harlow, 2002.
- [94] M. Mahfouf, S. Kandiah, D. A. Linkens. Fuzzy model-based predictive control using an arx structure with feedforward. *Fuzzy Sets and Systems*, Vol. 125, str. 39–59, 2002.
- [95] E. H. Mamdani, S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, Vol. 7, No. 1, str. 1–13, 1975.
- [96] K. Man, K. Tang, S. Kwong. *Genetic Algorithms, Concepts and Designs*. Springer-Verlag, 1998.
- [97] MathWorks. *Fuzzy Logic Toolbox User's Guide*. MathWorks.

-
- [98] D. Q. Mayne, S. Raković. Model predictive control of constrained piecewise affine discrete-time systems. *International Journal of Robust and Nonlinear Control*, Vol. 13, No. 3, str. 261–279, 2003.
- [99] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, Vol. 36, No. 6, str. 789–814, 2000.
- [100] Z. Michalewicz. Do not kill unfeasible individuals. *Dabrowski, Michalewicz, Ras, (Eds.), Proceedings of the Fourth Intelligent Information Systems Workshop, (IIS'95)*, str. 110–123, 1995.
- [101] Z. Michalewicz, G. Nazhiyath. Genocop iii: a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. *IEEE International Conference on Evolutionary Computation*, Volume 2, str. 647–651, 1995.
- [102] R. Möbus, M. Baotić, M. Morari. *Multi-objective adaptive cruise control*, poglavje Hybrid control of an automotive robotized gearbox for reduction of consumptions and emissions, str. 359–374. Number 2623 in Lecture Notes in Computer Science. Springer-Verlag, 2003.
- [103] D. S. Naidu. *Optimal Control Systems*. Electrical Engineering Textbook. CRC Press, Boca Raton, 2003.
- [104] H. Nakada, K. Takaba, T. Katayama. Identification of piecewise affine systems based on statistical clustering technique. *Automatica*, Vol. 41, No. 5, str. 905–913, 2005.
- [105] A. Nerode, W. Kohn. *Hybrid systems*, Volume 736 of *Lecture Notes in Computer Science*, poglavje Models for hybrid systems: Automata, topologies and stability, str. 317–356. Springer Verlag, Berlin, Heidelberg, 1993.

-
- [106] H. Nounou, K. Passino. Fuzzy model predictive control: techniques, stability issues, and examples. str. 423–428. IEEE International Symposium on Intelligent Control, Intelligent Systems & Semiotics, IEEE, 1999.
- [107] R. Palm, D. Driankov. Fuzzy switched hybrid systems – modelling and identification. *Proc. of the 1998 IEEE/ISCI/CIRA/SAS Joint Conf., Gaithersburg MD*, str. 130–135, 1998.
- [108] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, R. Vidal. Identification of hybrid systems: a tutorial. *European Journal of Control*, Vol. 13, No. 2–3, str. 242–260, 2007.
- [109] B. Potočnik. *Modeliranje, simulacija in vodenje mešanih logično dinamičnih sistemov*. Magistrsko delo, Fakulteta za elektrotehniko, Univerza v Ljubljani, 2001.
- [110] B. Potočnik. *Modeliranje, simulacija in vodenje hibridnih dinamičnih sistemov*. Doktorska disertacija, Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, 2004.
- [111] B. Potočnik, A. Bemporad, F. D. Torrisi, G. Mušič, B. Zupančič. Hybrid modelling and optimal control of a multiproduct batch plant. *Control eng. pract.*, Vol. 12, No. 9, str. 1127–1137, 2004.
- [112] B. Potočnik, G. Mušič, I. Škrjanc, B. Zupančič. Model-based predictive control of hybrid systems: a probabilistic neural-network approach to real-time control. *J. intell. robot. syst.*, Vol. 51, No. 1, str. 45–63, 2008.
- [113] B. Potočnik, G. Mušič, B. Zupančič. A new technique for translating discrete hybrid automata into piecewise affine systems. *Math. comput. model. dyn. syst.*, Vol. 10, No. 1, str. 41–57, 2004.
- [114] B. Potočnik, G. Mušič, B. Zupančič. Model predictive control of discrete time hybrid systems with discrete inputs. *ISA Transactions*, Vol. 44, No.

- 2, str. 199–211, 2005.
- [115] R. A. Pulley. Batch process modelling. Club Report CR 2828(CON), Warren Springs Laboratory, Herts, UK, 1986.
- [116] Y. Qin, L.-M. Jia. Fuzzy hybrid control and its application in complex combustion processes. *2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS'02)*, str. 78. IEEE, 2002.
- [117] J. B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, , str. 38–52, 2000.
- [118] J. Richalet. Industrial application of model based predictive control. *Automatica*, Vol. 29, No. 5, str. 1251–1274, 1993.
- [119] G. Ripaccioli, A. Bemporad, F. Assadian, C. Dextreit, S. Di Cairano, I. V. Kolmanovsky. *Hybrid Systems: Computation and Control*, poglavje Hybrid modeling, identification, and predictive control: an application to hybrid electric vehicle energy management. Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, 2009. V tisku.
- [120] J. Roll, A. Bemporad, L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, Vol. 40, No. 1, str. 37–50, 2004.
- [121] J. Roubos, R. Babuška, P. Bruijn, H. Verbruggen. Predictive control by local linearization of a Takagi-Sugeno fuzzy model. *Proceedings of the IEEE International Conference on Fuzzy Systems*, str. 37–42, 1998.
- [122] H. Sarimveis, G. Bafas. Fuzzy model predictive control of non-linear processes using genetic algorithms. *Fuzzy Sets and Systems*, Vol. 139, str. 59–80, 2003.
- [123] S. C. Shin, S. B. Park. Ga - based predictive control for nonlinear processes. *IEEE Electronics Letters*, Vol. 34, No. 20, str. 1980–1981, 1998.

-
- [124] K. S. Sibirsky. *Introduction to topological dynamics*. Noordhoff international publishing, Leyden, 1975.
- [125] J. Sjöberg, A. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarson, A. Juditsky. Nonlinear black-box modelling in system identification: A unified overview. *Automatica*, Vol. 31, No. 12, str. 1691–1724, 1995.
- [126] O. Slupphaug, B. Foss. Model predictive control for a class of hybrid systems. European Control Conference, EUCA, 1997.
- [127] O. Slupphaug, J. Vada, B. Foss. MPC in systems with continuous and discrete control inputs. American Control Conference, 1997.
- [128] J. Solis, D. Sáez, P. Estévez. Particle swarm optimization-based fuzzy predictive control strategy. str. 8525–8530. IEEE World Congress on Computational Intelligence, IEEE, 2006.
- [129] E. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic control*, Vol. 26, No. 2, str. 346–358, 1981.
- [130] M. Sugeno, K. Tanaka. Successive identification of a fuzzy model and its application to prediction of a complex system. *Fuzzy Sets and Systems*, Vol. 42, str. 315–334, 1991.
- [131] I. Škrjanc, S. Blažič, O. Agamenonni. Identification of dynamical systems with a robust interval fuzzy model. *Automatica*, Vol. 41, str. 327–332, 2005.
- [132] I. Škrjanc, D. Matko. Fuzzy predictive functional control in the state space domain. *Journal of Intelligent and Robotic Systems*, Vol. 31, str. 283–297, 2001.
- [133] T. Takagi, M. Sugeno. Fuzzy identification of systems and its application to modelling and control. *IEEE Trans. System Man Cybernet.*, Vol. 15, str. 116–132, 1985.

-
- [134] L. Tavernini. Differential automata and their discrete simulators. *Nonlinear Analysis: Theory, Methods and Applications*, Vol. 11, No. 6, str. 665–683, 1987.
- [135] J. Thomas, D. Dumur, J. Buisson. Predictive control of hybrid systems under a multi-mld formalism with state space polyhedral partition. Boston, Massachusetts, USA, 2004. American Control Conference.
- [136] F. D. Torrisi, A. Bemporad. Hysdel - a tool for generating computational hybrid models. *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 2, str. 235–249, 2004.
- [137] A. Van der Schaft, H. Schumacher. An introduction to hybrid dynamical systems. *Lecture Notes in Control and Information Sciences*, Vol. 251, str. v–vii, 1999.
- [138] A. J. Van der Schaft, J. M. Schumacher. Complementarity modelling of hybrid systems. *IEEE Transactions on Automatic Control*, Vol. 26, No. 2, str. 346–357, 1998.
- [139] R. Vidal. Recursive identification of switched arx systems. *Automatica*, Vol. 44, No. 9, str. 2274–2287, 2008.
- [140] X. Wang, J. Xiao. Pso-based model predictive control for nonlinear processes. *Lecture Notes in Computer Science*, Vol. 3611, str. 196–203, 2005.
- [141] C. Wen, S. Wang, X. Jin, X. Ma. Identification of dynamic systems using piecewise-affine basis function models. *Automatica*, Vol. 43, No. 10, str. 1824–1831, 2007.
- [142] H. S. Witsenhausen. A class of hybrid-state continuous time dynamic systems. *IEEE Trans. on Automatic Control*, Vol. 11, No. 2, str. 161–167, 1966.

-
- [143] I. Woolley, C. Kambhampati, D. Sandoz, K. Warwick. Intelligent control toolkit for an advanced control system. str. 445–450. UKACC International Conference on Control, IEE, 1998.
- [144] X. Xu, P. J. Antsaklis. Optimal control of switched systems based on parameterization of the switching instants. *IEEE Transactions on Automatic Control*, Vol. 49, No. 1, str. 105–119, 2004.
- [145] L. Zadeh. Fuzzy sets. *Information and control*, Vol. 8, No. 3, str. 338–353, 1965.

Dodatek A

Parametri modela in vodenja

Spodaj je podana standardna struktura, ki se uporablja v programskem orodju Multi-Parametric Toolbox. V njej so zbrani parametri modela in vodenja, ki so potrebni za zagon algoritma za prediktivno vodenje hibridnih sistemov na osnovi reševanja mešanih celoštevilskih optimizacijskih problemov – glej razdelek 9.5.

```
Using a PWA model...
+++++
| ID| Constraint| Type| Tag|
+++++
| #1| Numeric value| Element-wise 6x1| umin < u(k-1) < umax|
| #2| Numeric value| Element-wise 2x1| ymin < reference < ymax|
| #3| Numeric value| Element-wise 6x1| umin < u_3 < umax|
| #4| Numeric value| Equality constraint (derived) 1x1| u_3(1) in [0 0.01 0.02 0.05 0.1 1]|
| #5| Numeric value| Binary constraint| u_3(2) in [0 1]|
| #6| Numeric value| Binary constraint| u_3(3) in [0 1]|
| #7| Numeric value| Element-wise 4x1| xmin < x_3 < xmax|
| #8| Numeric value| Element-wise 4x1| xmin < x_4 < xmax|
| #9| Numeric value| Element-wise 2x1| ymin < y_3 < ymax|
| #10| Numeric value| Equality constraint (derived) 1x1| d_3(1) => x_4 == A{1}*x_3 + B{1}*u_3 + f{1}|
| #11| Numeric value| Equality constraint (derived) 1x1| d_3(1) => guardX{1}*x_3 + guardU{1}*u_3 <= guardC{1}|
| #12| Numeric value| Equality constraint (derived) 1x1| d_3(2) => x_4 == A{2}*x_3 + B{2}*u_3 + f{2}|
| #13| Numeric value| Equality constraint (derived) 1x1| d_3(2) => guardX{2}*x_3 + guardU{2}*u_3 <= guardC{2}|
| #14| Numeric value| Equality constraint (derived) 1x1| d_3(3) => x_4 == A{3}*x_3 + B{3}*u_3 + f{3}|
| #15| Numeric value| Equality constraint (derived) 1x1| d_3(3) => guardX{3}*x_3 + guardU{3}*u_3 <= guardC{3}|
| #16| Numeric value| Equality constraint (derived) 1x1| d_3(4) => x_4 == A{4}*x_3 + B{4}*u_3 + f{4}|
| #17| Numeric value| Equality constraint (derived) 1x1| d_3(4) => guardX{4}*x_3 + guardU{4}*u_3 <= guardC{4}|
| #18| Numeric value| Equality constraint (derived) 1x1| d_3(5) => x_4 == A{5}*x_3 + B{5}*u_3 + f{5}|
| #19| Numeric value| Equality constraint (derived) 1x1| d_3(5) => guardX{5}*x_3 + guardU{5}*u_3 <= guardC{5}|
| #20| Numeric value| Equality constraint (derived) 1x1| d_3(6) => x_4 == A{6}*x_3 + B{6}*u_3 + f{6}|
| #21| Numeric value| Equality constraint (derived) 1x1| d_3(6) => guardX{6}*x_3 + guardU{6}*u_3 <= guardC{6}|
| #22| Numeric value| Equality constraint (derived) 1x1| d_3(7) => x_4 == A{7}*x_3 + B{7}*u_3 + f{7}|
| #23| Numeric value| Equality constraint (derived) 1x1| d_3(7) => guardX{7}*x_3 + guardU{7}*u_3 <= guardC{7}|
| #24| Numeric value| Equality constraint (derived) 1x1| d_3(8) => x_4 == A{8}*x_3 + B{8}*u_3 + f{8}|
| #25| Numeric value| Equality constraint (derived) 1x1| d_3(8) => guardX{8}*x_3 + guardU{8}*u_3 <= guardC{8}|
| #26| Numeric value| Equality constraint (derived) 1x1| d_3(9) => x_4 == A{9}*x_3 + B{9}*u_3 + f{9}|
| #27| Numeric value| Equality constraint (derived) 1x1| d_3(9) => guardX{9}*x_3 + guardU{9}*u_3 <= guardC{9}|
| #28| Numeric value| Equality constraint (derived) 1x1| d_3(10) => x_4 == A{10}*x_3 + B{10}*u_3 + f{10}|
| #29| Numeric value| Equality constraint (derived) 1x1| d_3(10) => guardX{10}*x_3 + guardU{10}*u_3 <= guardC{10}|
```

```

| #30| Numeric value| Equality constraint 1x1| sum d_3 = 1|
| #31| Numeric value| Equality constraint 1x1| y_3 == C*x_3 + D*u_3 + g|
| #32| Numeric value| Element-wise 6x1| umin < u_2 < umax|
| #33| Numeric value| Element-wise 6x1| dumin < u_3 - u_2 < dumax|
| #34| Numeric value| Equality constraint (derived) 1x1| u_2(1) in [0 0.01 0.02 0.05 0.1 1]|
| #35| Numeric value| Binary constraint| u_2(2) in [0 1]|
| #36| Numeric value| Binary constraint| u_2(3) in [0 1]|
| #37| Numeric value| Element-wise 4x1| xmin < x_2 < xmax|
| #38| Numeric value| Element-wise 2x1| ymin < y_2 < ymax|
| #39| Numeric value| Equality constraint (derived) 1x1| d_2(1) => x_3 == A{1}*x_2 + B{1}*u_2 + f{1}|
| #40| Numeric value| Equality constraint (derived) 1x1| d_2(1) => guardX{1}*x_2 + guardU{1}*u_2 <= guardC{1}|
| #41| Numeric value| Equality constraint (derived) 1x1| d_2(2) => x_3 == A{2}*x_2 + B{2}*u_2 + f{2}|
| #42| Numeric value| Equality constraint (derived) 1x1| d_2(2) => guardX{2}*x_2 + guardU{2}*u_2 <= guardC{2}|
| #43| Numeric value| Equality constraint (derived) 1x1| d_2(3) => x_3 == A{3}*x_2 + B{3}*u_2 + f{3}|
| #44| Numeric value| Equality constraint (derived) 1x1| d_2(3) => guardX{3}*x_2 + guardU{3}*u_2 <= guardC{3}|
| #45| Numeric value| Equality constraint (derived) 1x1| d_2(4) => x_3 == A{4}*x_2 + B{4}*u_2 + f{4}|
| #46| Numeric value| Equality constraint (derived) 1x1| d_2(4) => guardX{4}*x_2 + guardU{4}*u_2 <= guardC{4}|
| #47| Numeric value| Equality constraint (derived) 1x1| d_2(5) => x_3 == A{5}*x_2 + B{5}*u_2 + f{5}|
| #48| Numeric value| Equality constraint (derived) 1x1| d_2(5) => guardX{5}*x_2 + guardU{5}*u_2 <= guardC{5}|
| #49| Numeric value| Equality constraint (derived) 1x1| d_2(6) => x_3 == A{6}*x_2 + B{6}*u_2 + f{6}|
| #50| Numeric value| Equality constraint (derived) 1x1| d_2(6) => guardX{6}*x_2 + guardU{6}*u_2 <= guardC{6}|
| #51| Numeric value| Equality constraint (derived) 1x1| d_2(7) => x_3 == A{7}*x_2 + B{7}*u_2 + f{7}|
| #52| Numeric value| Equality constraint (derived) 1x1| d_2(7) => guardX{7}*x_2 + guardU{7}*u_2 <= guardC{7}|
| #53| Numeric value| Equality constraint (derived) 1x1| d_2(8) => x_3 == A{8}*x_2 + B{8}*u_2 + f{8}|
| #54| Numeric value| Equality constraint (derived) 1x1| d_2(8) => guardX{8}*x_2 + guardU{8}*u_2 <= guardC{8}|
| #55| Numeric value| Equality constraint (derived) 1x1| d_2(9) => x_3 == A{9}*x_2 + B{9}*u_2 + f{9}|
| #56| Numeric value| Equality constraint (derived) 1x1| d_2(9) => guardX{9}*x_2 + guardU{9}*u_2 <= guardC{9}|
| #57| Numeric value| Equality constraint (derived) 1x1| d_2(10) => x_3 == A{10}*x_2 + B{10}*u_2 + f{10}|
| #58| Numeric value| Equality constraint (derived) 1x1| d_2(10) => guardX{10}*x_2 + guardU{10}*u_2 <= guardC{10}|
| #59| Numeric value| Equality constraint 1x1| sum d_2 = 1|
| #60| Numeric value| Equality constraint 1x1| y_2 == C*x_2 + D*u_2 + g|
| #61| Numeric value| Element-wise 6x1| umin < u_1 < umax|
| #62| Numeric value| Element-wise 6x1| dumin < u_2 - u_1 < dumax|
| #63| Numeric value| Equality constraint (derived) 1x1| u_1(1) in [0 0.01 0.02 0.05 0.1 1]|
| #64| Numeric value| Binary constraint| u_1(2) in [0 1]|
| #65| Numeric value| Binary constraint| u_1(3) in [0 1]|
| #66| Numeric value| Element-wise 4x1| xmin < x_1 < xmax|
| #67| Numeric value| Element-wise 2x1| ymin < y_1 < ymax|
| #68| Numeric value| Equality constraint (derived) 1x1| d_1(1) => x_2 == A{1}*x_1 + B{1}*u_1 + f{1}|
| #69| Numeric value| Equality constraint (derived) 1x1| d_1(1) => guardX{1}*x_1 + guardU{1}*u_1 <= guardC{1}|
| #70| Numeric value| Equality constraint (derived) 1x1| d_1(2) => x_2 == A{2}*x_1 + B{2}*u_1 + f{2}|
| #71| Numeric value| Equality constraint (derived) 1x1| d_1(2) => guardX{2}*x_1 + guardU{2}*u_1 <= guardC{2}|
| #72| Numeric value| Equality constraint (derived) 1x1| d_1(3) => x_2 == A{3}*x_1 + B{3}*u_1 + f{3}|
| #73| Numeric value| Equality constraint (derived) 1x1| d_1(3) => guardX{3}*x_1 + guardU{3}*u_1 <= guardC{3}|
| #74| Numeric value| Equality constraint (derived) 1x1| d_1(4) => x_2 == A{4}*x_1 + B{4}*u_1 + f{4}|
| #75| Numeric value| Equality constraint (derived) 1x1| d_1(4) => guardX{4}*x_1 + guardU{4}*u_1 <= guardC{4}|
| #76| Numeric value| Equality constraint (derived) 1x1| d_1(5) => x_2 == A{5}*x_1 + B{5}*u_1 + f{5}|
| #77| Numeric value| Equality constraint (derived) 1x1| d_1(5) => guardX{5}*x_1 + guardU{5}*u_1 <= guardC{5}|
| #78| Numeric value| Equality constraint (derived) 1x1| d_1(6) => x_2 == A{6}*x_1 + B{6}*u_1 + f{6}|
| #79| Numeric value| Equality constraint (derived) 1x1| d_1(6) => guardX{6}*x_1 + guardU{6}*u_1 <= guardC{6}|
| #80| Numeric value| Equality constraint (derived) 1x1| d_1(7) => x_2 == A{7}*x_1 + B{7}*u_1 + f{7}|
| #81| Numeric value| Equality constraint (derived) 1x1| d_1(7) => guardX{7}*x_1 + guardU{7}*u_1 <= guardC{7}|
| #82| Numeric value| Equality constraint (derived) 1x1| d_1(8) => x_2 == A{8}*x_1 + B{8}*u_1 + f{8}|
| #83| Numeric value| Equality constraint (derived) 1x1| d_1(8) => guardX{8}*x_1 + guardU{8}*u_1 <= guardC{8}|
| #84| Numeric value| Equality constraint (derived) 1x1| d_1(9) => x_2 == A{9}*x_1 + B{9}*u_1 + f{9}|
| #85| Numeric value| Equality constraint (derived) 1x1| d_1(9) => guardX{9}*x_1 + guardU{9}*u_1 <= guardC{9}|
| #86| Numeric value| Equality constraint (derived) 1x1| d_1(10) => x_2 == A{10}*x_1 + B{10}*u_1 + f{10}|
| #87| Numeric value| Equality constraint (derived) 1x1| d_1(10) => guardX{10}*x_1 + guardU{10}*u_1 <= guardC{10}|
| #88| Numeric value| Equality constraint 1x1| sum d_1 = 1|
| #89| Numeric value| Equality constraint 1x1| y_1 == C*x_1 + D*u_1 + g|

```

```

| #90| Numeric value| Element-wise 6x1| u_min < u_0 < u_max|
| #91| Numeric value| Element-wise 6x1| d_min < u_0 - u_prev < d_max|
| #92| Numeric value| Equality constraint (derived) 1x1| u_0(1) in [0 0.01 0.02 0.05 0.1 1]|
| #93| Numeric value| Binary constraint| u_0(2) in [0 1]|
| #94| Numeric value| Binary constraint| u_0(3) in [0 1]|
| #95| Numeric value| Element-wise 4x1| x_min < x_0 < x_max|
| #96| Numeric value| Element-wise 2x1| y_min < y_0 < y_max|
| #97| Numeric value| Equality constraint (derived) 1x1| d_0(1) => x_1 == A{1}*x_0 + B{1}*u_0 + f{1}|
| #98| Numeric value| Equality constraint (derived) 1x1| d_0(1) => guardX{1}*x_0 + guardU{1}*u_0 <= guardC{1}|
| #99| Numeric value| Equality constraint (derived) 1x1| d_0(2) => x_1 == A{2}*x_0 + B{2}*u_0 + f{2}|
| #100| Numeric value| Equality constraint (derived) 1x1| d_0(2) => guardX{2}*x_0 + guardU{2}*u_0 <= guardC{2}|
| #101| Numeric value| Equality constraint (derived) 1x1| d_0(3) => x_1 == A{3}*x_0 + B{3}*u_0 + f{3}|
| #102| Numeric value| Equality constraint (derived) 1x1| d_0(3) => guardX{3}*x_0 + guardU{3}*u_0 <= guardC{3}|
| #103| Numeric value| Equality constraint (derived) 1x1| d_0(4) => x_1 == A{4}*x_0 + B{4}*u_0 + f{4}|
| #104| Numeric value| Equality constraint (derived) 1x1| d_0(4) => guardX{4}*x_0 + guardU{4}*u_0 <= guardC{4}|
| #105| Numeric value| Equality constraint (derived) 1x1| d_0(5) => x_1 == A{5}*x_0 + B{5}*u_0 + f{5}|
| #106| Numeric value| Equality constraint (derived) 1x1| d_0(5) => guardX{5}*x_0 + guardU{5}*u_0 <= guardC{5}|
| #107| Numeric value| Equality constraint (derived) 1x1| d_0(6) => x_1 == A{6}*x_0 + B{6}*u_0 + f{6}|
| #108| Numeric value| Equality constraint (derived) 1x1| d_0(6) => guardX{6}*x_0 + guardU{6}*u_0 <= guardC{6}|
| #109| Numeric value| Equality constraint (derived) 1x1| d_0(7) => x_1 == A{7}*x_0 + B{7}*u_0 + f{7}|
| #110| Numeric value| Equality constraint (derived) 1x1| d_0(7) => guardX{7}*x_0 + guardU{7}*u_0 <= guardC{7}|
| #111| Numeric value| Equality constraint (derived) 1x1| d_0(8) => x_1 == A{8}*x_0 + B{8}*u_0 + f{8}|
| #112| Numeric value| Equality constraint (derived) 1x1| d_0(8) => guardX{8}*x_0 + guardU{8}*u_0 <= guardC{8}|
| #113| Numeric value| Equality constraint (derived) 1x1| d_0(9) => x_1 == A{9}*x_0 + B{9}*u_0 + f{9}|
| #114| Numeric value| Equality constraint (derived) 1x1| d_0(9) => guardX{9}*x_0 + guardU{9}*u_0 <= guardC{9}|
| #115| Numeric value| Equality constraint (derived) 1x1| d_0(10) => x_1 == A{10}*x_0 + B{10}*u_0 + f{10}|
| #116| Numeric value| Equality constraint (derived) 1x1| d_0(10) => guardX{10}*x_0 + guardU{10}*u_0 <= guardC{10}|
| #117| Numeric value| Equality constraint 1x1| sum d_0 = 1|
| #118| Numeric value| Equality constraint 1x1| y_0 == C*x_0 + D*u_0 + g|
+++++
Quadratic scalar (real, homogeneous, 5 variables)

VARS =

x: {[2x1 sdpvar] [2x1 sdpvar] [2x1 sdpvar] [2x1 sdpvar] [2x1 sdpvar]}
u: {[3x1 sdpvar] [3x1 sdpvar] [3x1 sdpvar] [3x1 sdpvar]}
y: {[1x1 sdpvar] [1x1 sdpvar] [1x1 sdpvar] [1x1 sdpvar]}
d: {[10x1 sdpvar] [10x1 sdpvar] [10x1 sdpvar] [10x1 sdpvar]}
ref: [1x1 sdpvar]
uprev: [3x1 sdpvar]
type: 'online'

```


Izjava

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