

**Tallinna Tehnikaülikool**  
**Süsteemitehnika teaduskond**  
**Automaatikainstituut**

Andri Riid

**FERMENTATSIOONIPROTSESSI HÄGUS MODELLEERIMINE**  
**JA JUHTIMINE**

Juhendaja: E. Rüstern

Tallinn 1997

## Sisukord

Sisukord	v
Sissejuhatus	vii
<b>1. HÄGUS HULGATEOORIA</b>	<b>1-1</b>
1.1. Põhimõisted	1-1
1.2. Tehted hägusate hulkadega	1-3
1.3. Hägus loogika	1-4
<b>2. HÄGUSAD MUDELID</b>	<b>2-1</b>
2.1. Meeldetuletuseks	2-1
2.2. Hägus mudel	2-1
2.3. Lingvistilised märgendid	2-2
2.4. Hägusad reeglid	2-3
2.4.1. Reeglibaas üldkujul	2-5
2.5. Mudeli väljundi järeldusalgoritm	2-6
2.6. Häguärastamine	2-7
2.6.1. Näide	2-8
2.7. Liikmesfunktsioonid	2-9
2.8. Mudelite konstrueerimine	2-10
2.8.1. Hägus lahterdamine	2-11
2.8.2. Normeerimistegurid	2-12
<b>3. HÄGUS MODELLEERIMINE</b>	<b>3-1</b>
3.1 Sissejuhatus	3-1
3.2 Otsemeetod	3-2
3.3 Shabloonil baseeruv hägus modelleerimine	3-2
3.4 Modelleerimine närvivõrgulaadse mudeliga	3-5
<b>4. HÄGUS JUHTIMINE</b>	<b>4-1</b>
4.1 Tagasisidestatud juhtimise põhimõtted	4-1
4.2 Hägusloogikaregulaatori olemus	4-1
4.3 Analoogiad tavareguleerimisseadustega	4-3
4.3.1 Hägus PI-regulaator	4-3
4.3.2 Hägus PD-regulaator	4-3
4.3.3 Hägus PID-regulaator	4-4
4.4 Hägusloogikaregulaatori konstrueerimine	4-5
4.4.1 Hägusloogikaregulaatori normeerimistegurid	4-5
4.4.2 Reeglibaasi koostamine	4-6
4.5 Hägusloogikaregulaatorite häälestamine	4-7
4.6 Arendatud juhtimismeetodid	4-8
<b>5. FERMENTATSIOONIPROTSSESSIDE MODELLEERIMINE JA JUHTIMINE</b>	<b>5-1</b>
5.1 Fermentatsiooniprotsesside olemus	5-1
5.2 Fermentatsiooniprotsessi matemaatiline modelleerimine	5-4
5.2.1 Füsioloogilised mudelid	5-4
5.2.2 Mittestruktuursed mudelid	5-6
5.3 Fermentatsiooniprotsesside optimeerimine	5-9
5.3.1 Perioodiliste fermentatsiooniprotsesside optimeerimine	5-9

---

5.3.2 Poolperioodiliste fermentatsiooniprotsesside optimeerimine	5-12
5.3.3 Toitevoo optimaalsed profiilid	5-15
5.3.3.1 Monotoonselt kasvav $\mu$	5-16
5.3.3.2 Monotoonselt kasvav $q_p$	5-17
5.3.3.3 Mittemonotoonsed $\mu$ ja $q_p$	5-19
5.3.3.4 Kokkuvõtteks	5-20
<b>6. FERMENTATSIOONIPROTSESSI MUDEL</b>	<b>6-1</b>
<b>7. EKSPERIMENTIDE PLANEERIMINE JA JUHTIMISALGORITMI VÄLJATÖÖTLUS</b>	<b>7-1</b>
7.1 Eksperimentide planeerimine	7-2
7.1.1 Protsessi lühiiseloostus	7-2
7.1.2 Substraatide kontsentratsioonid fermentaatoris	7-3
7.2 Identifitseerimismeetodid	7-6
7.2.1 Otsemeetod	7-6
7.2.2 Identifitseerimine närvivõrguga	7-7
7.2.2.1 Aparatuuri kirjeldus	7-7
7.2.2.2 Strateegiad	7-9
7.2.3 Identifitseerimine lingvistilise meetodiga	7-10
7.2.3.1 Meetodi idee	7-10
7.2.3.2 Fermentatsiooni juhtimine	7-11
7.2.3.3 Meetodi kirjeldus	7-12
<b>8. JUHTIMISALGORITMI TULETAMINE LINGVISTILISEL MEETODIL</b>	<b>8-1</b>
8.1 Hägusate lahterduste defineerimine	8-1
8.2 Reeglite analüüs ja hägusa regulaatori süntees	8-3
8.2.1 Reeglite analüüsi alused	8-3
8.2.2 Seadesuuruste regulaatori süntees	8-6
8.2.2.1 Tootmisrežiimi defineerimine	8-7
8.2.2.2 Juhtimisstsenaariumide väljatöötamine	8-9
8.2.2.3 Edasine süntees	8-10
8.2.2.4 Sünteesi resultaat	8-13
8.3 Resultaatide analüüs	8-15
Kokkuvõte	
LISA	
Kasutatud kirjandus	

## Sissejuhatus

Biotehnoloogia on suhteliselt noor teadusharu, kuid tema tähtsust tänapäeva maailmas on raske alahinnata. Biotehnoloogiliste protsesside saadused on selleks liiga suure tähtsusega. Biotehnoloogilise protsessi edukus sõltub sellest, kas ja mil määral suudetakse rahuldada protsessi saaduse tekkeks vajalikke tingimusi, teisisõnu sellest, kuidas ja kui hästi teda juhitakse. Viimasest tuleneb, et protsessi juhtimisalgoritmi tuletamine on kriitilise tähtsusega ning et algoritmi täiustamine omab mõtet.

Protsessi juhtimisalgoritmi väljatöötamine eeldab enamikel juhtudel protsessi mudeli olemasolu. Protsessi mudel võib olla lihtne või keerukas, täpne või vähemtäpne. Sõltuvalt sellest osutub ka sünteesitud juhtimisalgoritm lihtsaks või keerukaks, edukaks või vähemedukaks. Automaatjuhtimise varasemal ajajärgul olid kasutusel ainult analüütilised matemaatilised mudelid, mis koosnevad protsessi muutujaid siduvaist algebralistest, diferentsiaal- või differentsvõrranditest. Fermentatsiooniprotsessid on aga enamasti keerulised protsessid ja sellise täpne matemaatiline modelleerimine on keeruline. Arvestama peab paljusid protsessi käiku mõjutavaid tegureid, ühes sellega kasvab võrrandite arv, kaob ülevaatlikkus ja suureneb modelleerimiseks vajaliku töö maht.

Alates käesoleva sajandi seitsmekümnendatest aastatest jõudsalt arenenud hägus hulgateooria annab inimese käsutusse vahendid, mis võimaldab toime tulla ka puudulikult defineeritud või mittelineaarsete süsteemidega, mille puhul tavamodelleerimismeetodid jõuetuks osutuvad. Toetudes hägusale hulgateooriale õnnestub konstrueerida hägusaid mudeleid, mis opereerivad protsessi sisendite ja väljundite väärtuste kvalitatiiivsete hinnangutega ning determineerivad protsessi seosed lingvistiliste reeglite kujul. Põhjalikum ülevaade hägusast hulgateooriast, sellele vastavast hägusast loogikast ning hägusatest mudelitest on esitatud käesoleva töö peatükkides 1 ja 2.

Automaatjuhtimise vaatevinklist võib hägusaid mudeleid võib laias laastus jagada kahte gruppi - juhitavate objektide hägusad mudelid ja juhtivate objektide (s.o. regulaatorite) hägusad mudelid. Lühiülevaate hägusatest regulaatoritest leiab käesoleva töö neljandast peatükist.

Hägusate mudelite üheks nõrgemaks kohaks on konkreetse modelleerimisteooria puudumine. Paljude probleemide lahendamiseks kas puudub üldse arvestatav meetodika või on viimane suuremal või väiksemal määral jäetud toetuma spetsialisti kogemustele ja intuitsioonile. Sellest hoolimata on sellised osaliselt determineeritud meetodid kasutatavad ja kolmandas peatükis ka neid vaadeldakse. Arendatumate meetodite puhul kasutatakse modelleerimisel üldiselt modelleeritava süsteemiga tehtud eksperimentidest saadud eksperimentiandmeid. Keskendutakse kahele sellisele põhimõtteliselt erinevale meetodile, millest üks põhineb hägusa närvivõrgu kasutamisel ning teine on puhtlingvistiline meetod.

Püstitatud eesmärk - juhtida fermentatsiooniprotsessi kasutades hägusloogika vahendeid - on mitterealiseeritav, kui me juhitavast protsessist midagi ei tea. 5. peatükk käsitleb

fermentatsiooniprotsesse, nende modelleerimise ja juhtimise põhitõdesid. Põhjalikult vaadeldakse fermentatsiooniprotsesside optimaalse juhtimise osas saavutatut, see on ka allikas, kust ammutame ideid protsessi juhtimiseks hägusloogika vahendeid kasutades.

Fermentatsiooniprotsess, mida käesolevas töös modelleeritakse ja juhitakse, kujutab endast omakorda fermentatsiooniprotsessi mudelit. Tegu on keeruka matemaatilise mudeliga, mis üsna täpselt imiteerib tegelikku protsessi. Lisaks on olemas ka senikasutatud juhtimisalgoritm nominaalsete juhtimisprofiilide näol. Täpsem mudeli kirjeldus on toodud peatükis kuus.

Seitsmendas peatükis käsitletakse hägusaid modelleerimismeetodeid konkreetse protsessi modelleerimise ja juhtimise vaatenurga alt. Käesoleva töö maht dikteerib, et praktiliselt realiseeritakse neist ainult üks. Modelleerimise tulemusena saadud lingvistiline mudel on lihtsustatud ja invertteeritud kujul kasutatav protsessi regulaatorina.

Viimases kaheksandas peatükis toimub juba konkreetne regulaatori süntees ja tulemuste analüüs

Töö kirjutamisel on peetud silmas ka tema kasutusvõimalust õppematerjalina.

# 1. Hägus hulgateooria

## 1.1. Põhimõisted

Hulgateooria põhimõisteks on hulk - mõiste, millel puudub definitsioon. Eeldatakse, et inimene mõistab intuiitiivselt, mida hulk endast kujutab. Teiseks algmõisteks hulga kõrval on element, mida samuti ei defineerita.

Hulk moodustub hulga elementidest. Kui element  $x$  on hulga  $A$  element, siis öeldakse, et ta kuulub hulka  $A$ . Võimalik on ka vastupidine juhtum - element  $x$  ei kuulu hulka  $A$ .

Universaalhulga  $U$  elemendi  $x$  kuuluvust hulka  $A$  võib väljendada ka liikmesfunktsiooniga  $\mu_A(x)$ , kusjuures:

$$\mu_A(x) = 1, \text{ kui } x \in A$$

$$\mu_A(x) = 0, \text{ kui } x \notin A$$

Liikmesfunktsiooni väärtuseks saab olla liikmesuste hulga  $M$  element, kusjuures antud juhul on valida vaid kahe väärtuse vahel:

$$M = \{0; 1\}$$

Tegelikkuses võib tekkida olukordi, kus selline lähenemine osutub jäigaks. Vaatleme järgmist näidet:

Olgu meil vaja kirjeldada hulka  $A$  - hulka, mille elementideks on noored inimesed. Inimese noorte inimeste hulka kuulumise kriteeriumiks on tema vanus. Kuna inimese vanus ei saa üldjuhul olla väiksem nullist, võib sellega ära määrata noorte inimeste hulka kuuluva inimese vanuse alampiiri.

Vanuse ülempiiri on aga juba hoopis raskem paika panna. Me võime võrdsustada selle näiteks 20 eluaastaga, s.t. vanemad kui 20-aastased noorte inimeste hulka enam ei kuulu

Otsekohe tekib küsimus - miks on keegi oma 20. sünnipäeval noor ja järgmisel päeval seda enam pole? Ülempiiri võib nihutada ülespoole ent probleem sellest ei kao.

Ilmselt ei tohiks vahetegemine noorte ja vanade vahel olla nii järsk, s.t. meil peaks olema küsimusele - kas isik kuulub noorte hulka - vastamisel olema võimalik kasutada vastusevariantide 'jah' ja 'ei' kõrval ka vahepealseid võimalusi - ta on peaaegu noor, ta on pooleldi noor jne.

Klassikalise hulgateooria kõrval eksisteeriv hägus hulgateooria, mis lubab liikmesfunktsiooni  $\mu_A(x)$  väärtustele 1 ja 0 kõrval ka teisi, kõiki 1 ja 0 vahele jäävaid väärtusi ( $M=[0, 1]$ ), annabki meile sellise võimaluse.

Sõnastada võib seda nii, et element  $x$  kuulub hägusasse hulka  $A$  mingil liikmesfunktsiooni  $\mu_A(x)$  väärtusega määratud määral. Hulk  $A$  hägustub seetõttu, et elementide kuuluvus temasse on osaline ehk hägus. Hägususe piirjuhtumiteks on olukorrad, kus  $\mu_A(x) = 1$ , mille korral element  $x$  kuulub hulka  $A$  täiel määral ning  $\mu_A(x) = 0$ , mille puhul ta sinna kindlasti ei kuulu.

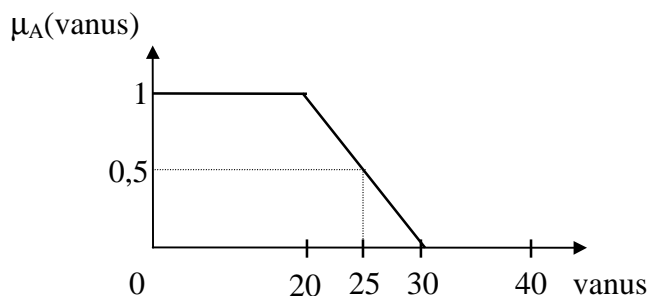
Pöördume nüüd tagasi oma näite juurde. Määrame kindlaks piirjuhtumid:

$$\mu_A(\text{vanus}) = 1, \text{ kui vanus} < 20 \text{ eluaastast}$$

$$\mu_A(\text{vanus}) = 0, \text{ kui vanus} > 30 \text{ eluaastast}$$

20 ja 30 eluaasta vahele jääb sellega eluea piirkond, mille korral kuulumine noorte inimeste hulka on osaline. Lihtsuse mõttes oletame, et 20- kuni 30-aastaste inimeste kuulumine noorte inimeste hulka on võrdelises seoses nende vanusega, s.t. graafiliselt saab seda väljendada sirgega (joonis 1-1):

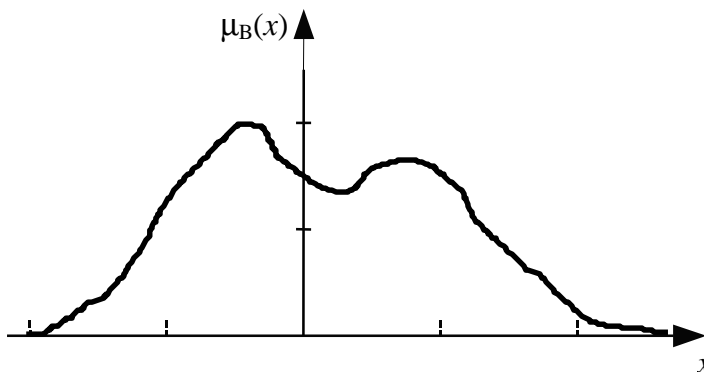
joonis 1-1



Seega on näiteks 25-aastane inimene pooleldi noor, kuna tema liikmesfunktsioon  $\mu_A(25) = 0,5$ .

Liikmesfunktsiooni väärtus on analoogiline tõenäosusteoorias tuntud tõenäosusega, millel võib samuti olla suvaline väärtus 1 ja nulli vahel ning liikmesfunktsioon ennast võib kõrvutada sealse jaotusfunktsiooniga, ehkki erinevalt viimasest pole liikmesfunktsiooni puhul nõutud, et ta oleks tingimata monotoonselt kasvav. Mingi hägusa hulga B liikmesfunktsioon  $\mu_B(x)$  on just täpselt selline nagu kõikide universaalhulga U elementide x hulka B kuulumise jaotus ette näeb:

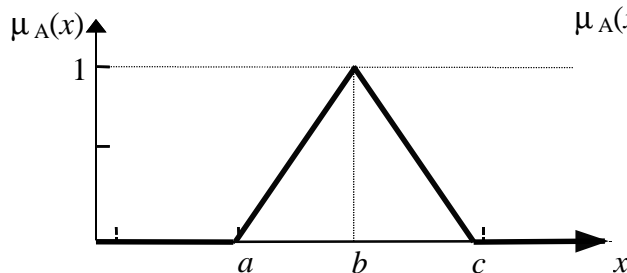
joonis 1-2



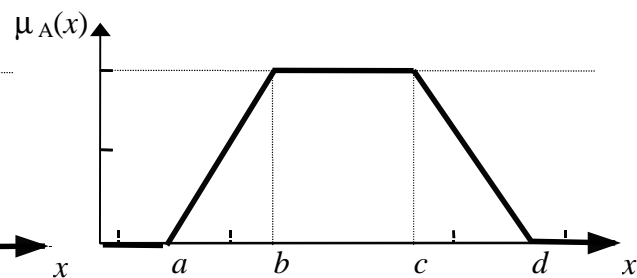
Hägusa hulgateooria rakendustes on enamasti siiski kasutusel vaid teatavat tüüpi lihtsustatud liikmesfunktsioonid. Näitena neist võib siinkohal tuua kolmnurkse (joonis 1-3a) ja trapetsikujulise liikmesfunktsiooni (joonis 1-3b) kui enamlevinumad.

joonis 1-3

a)



b)



## 1.2. Tehted hägusate hulkadega

Hägusate hulgateoorias on defineeritud kõik klassikalisesest hulgateooriast tuttavad hulka-  
seosed, omadused ja hulkadega sooritatavad tehted.

Kuna hägusa hulga elemendid on defineeritud nende liikmesfunktsioonide kaudu, on ka  
kõikvõimalikud operatsioonid hägusate hulkadega sooritatavad läbi  
liikmesfunktsioonide. Toome mõned näited kõige põhilisemate seoste kohta:

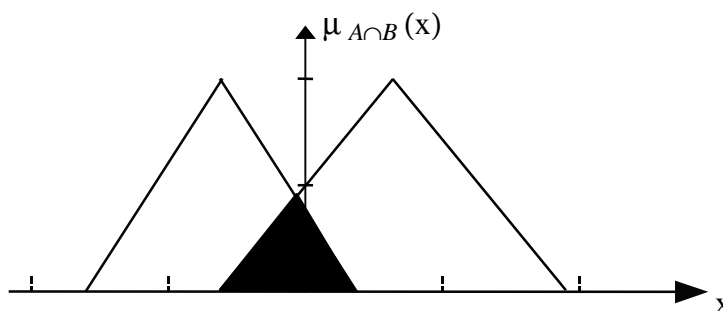
1. Universaalhulga E hägus alamhulk A on E hägusa alamhulga B osahulk kui

$$\exists x \in E : \mu_A(x) < \mu_B(x) \quad \text{avaldis 1-1}$$

2. Kahe hägusa hulga ühisosa on defineeritud järgnevalt:

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \quad \text{avaldis 1-2}$$

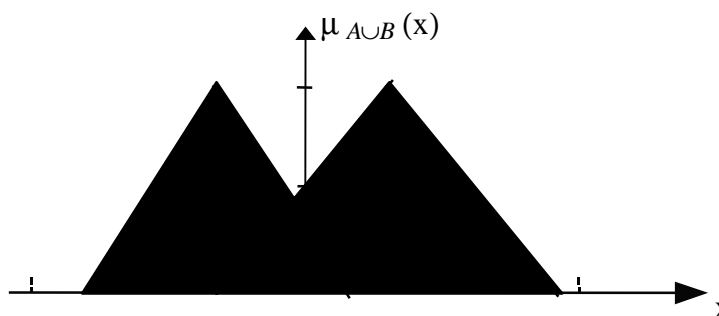
joonis 1-4



3. Kahe hägusa hulga ühend:

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \quad \text{avaldis 1-3}$$

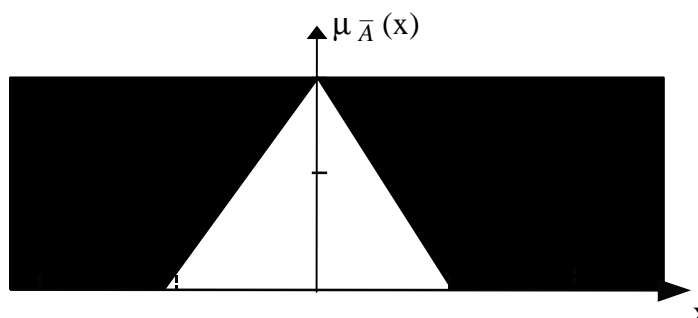
joonis 1-5



4. Ning hägusa hulga A täiend:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad \text{avaldis 1-4}$$

joonis 1-6



Näeme, et kui kasutame neis avaldistes liikmesfunktsioonide väärtustena klassikalises hulgateoorias lubatud suurusi (1 ja 0), saame klassikalises hulgateoorias kehtivad tulemused. Järelikult on alust väita, et hägusate hulkade teooria kujutab endast traditsionaalse hulgateooria üldistust. Viimane on ühtlasi antud peatüki kõige olulisemaks järelduseks.

### **1.3. Hägus loogika**

Sarnaselt hulgateooriale on võimalik “hägustada” ka loogikat. Laiemalt tuntud kaheväärtuselisel Boole'i loogikal, milles loogilise avaldise tõeväärtus saab olla kas tõene või väär, on ka väiksemaid vendi - mitmevalentseid loogikaid, kus olenevalt tõeväärtuste arvust on kasutusel mitmeid osalist tõe ja osalist vale kajastavaid mõisteid.

Hägusas loogikas võib tõeväärtuseks olla juba mistahes ratsionaalarv ühe, mis kajastab absoluutset tõesust ja nulli (absoluutne tõesuse puudumine) vahel.

Olemuselt on säärased hägusa loogika tõeväärtused sarnased hägusa hulga liikmesfunktsiooni väärtustega.

Hägus loogika on samas kujunenud universaalseks mõisteks, millega tähistatakse kõike nii hägusa hulgateooria kui hägusa loogikaga seonduvat.

## 2. Hägusad mudelid

### 2.1. Meeldetuletuseks

Juhtimisülesannete lahendamisel kasutatakse laialt juhitava süsteemi või protsessi matemaatilisi mudeleid. Süsteemi (protsessi) matemaatiline mudel kujutab endast süsteemi (protsessi) analüütilist kirjeldust algebraliste, diferentsiaal- või differentsvõrrandite abil. Taoline analüütilisuse nõue on klassikalisele juhtimisteooriale üldiselt omane. Laias laastus jagunevad objekti matemaatilised mudelid pidevateks ja diskreetseteks, sõltuvalt suhtumisest aega. Sealjuures tuntakse (diskreetsete mudelite näitel) sisend-väljundmudeleid, mida kirjeldab lineaarsete differentsvõrrandite süsteem:

$$y(t) + a_1y(t-1) + \dots + a_ny(t-n) = b_1u(t-k) + \dots + b_mu(t-k-m+1), \quad \text{avaldis 2-1}$$

kus  $y$  on väljundmuutujate vektor (ühe väljundi korral sisaldab 1 elementi),  $u$  - sisendmuutujate vektor ning  $a_1 \dots a_n$  ja  $b_1 \dots b_m$  koefitsentide vektorid.

Sisend-väljundmudelite kõrval on kasutusel ka olekumudelid, mis sisuliselt kirjeldavad sama tüüpi lineaarset differentsseost sisendite ja väljundite vahel, kuid on organiseeritud nii, et lubatud on vaid 1-taktiline hilistumine. Sealjuures on defineeritud olekumuutujad (muutuja  $x$  avaldises 2-2).

$$\begin{cases} x(t+1) = \mathbf{A}x(t) + \mathbf{B}u(t) \\ y(t) = \mathbf{C}x(t) + \mathbf{D}u(t) \end{cases} \quad \text{avaldis 2-2}$$

### 2.2. Hägus mudel

Süsteemi hägus mudel ei avaldu matemaatilisel kujul, vaid realiseerub hägusloogika vahenditega kirjeldatava kogemusliku algoritmina.

Hägusa mudelina mõistame süsteemi sellist mudelit, mis koosneb KUI ... SIIS - tüüpi süsteemi sisendeid ja väljundeid siduvatest loogilistest lausetest, kus süsteemi sisenditele vastav süsteemi reaktsioon arvutatakse hägusloogika järeldusmehhanismi abil (viimast nimetatakse ka ligikaudseks järeldamiseks).

Selliste lausete ehk reeglite komplekt tervikuna defineerib sisendite ja väljundite vahelise seose ehk relatsiooni.

Koostame nüüd selle illustreerimiseks ühe lihtsa süsteemi hägusa mudeli. Olgu tegu kahte sisendit ja üht väljundit omava süsteemiga. Näiteks tubast temperatuuri kirjeldava süsteemiga, kus süsteemi väljund - toa temperatuur - sõltub kahest tegurist: küttekeha temperatuur ja temperatuur õues, mis ongi antud süsteemi sisenditeks.

Süsteemi käitumist kirjeldavat reeglite kogu ehk mudeli reeglibaasi esitab tabel 2-1. Võime veenduda, et iga hägusa mudeli reegel on sisuliselt kahest osast - tingimuspoolest (algab sõnaga **KUI**) ja järelduspoolest (algab sõnaga **SIIS**) - koosnev liitlause.

tabel 2-1

Reegel 1	<b>KUI</b>	temperatuur õues on madal	<b>JA</b>	küttekeha temperatuur on kõrge	<b>SIIS</b>	toatemperatuur on paras
Reegel 2	<b>KUI</b>	temperatuur õues on madal	<b>JA</b>	küttekeha temperatuur on madal	<b>SIIS</b>	toatemperatuur on madal
Reegel 3	<b>KUI</b>	temperatuur õues on kõrge	<b>JA</b>	küttekeha temperatuur on kõrge	<b>SIIS</b>	toatemperatuur on paras
Reegel 4	<b>KUI</b>	temperatuur õues on kõrge	<b>JA</b>	küttekeha temperatuur on madal	<b>SIIS</b>	toatemperatuur on kõrge

### 2.3. Lingvistilised märgendid

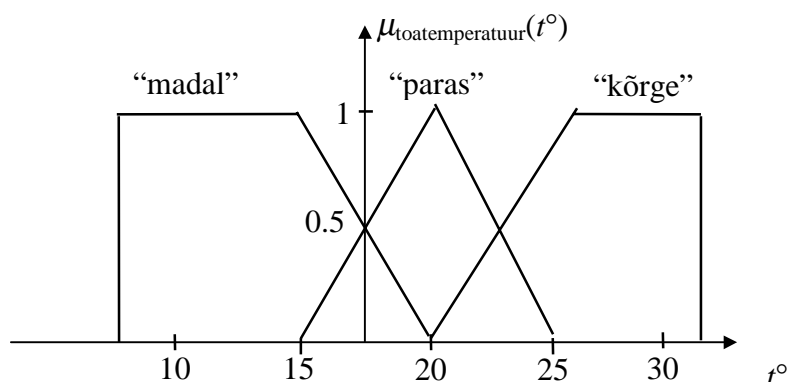
Informatsiooni põhjal mille annab tabel 2-1 selgub, et reeglid opereerivad mudeli sisendite ja väljundite lingvistiliste määratlustega (antud juhul on nendeks “külm”, “soe” ja “paras”). Viimased on esmapilgul laialivalguvad ja pigem kvalitatiivsed kui konkreetsed ja nii see ongi kuna reeglite funktsiooniks on kirjeldada mudeli käitumist mitte kvantitatiivselt vaid kvalitatiivselt. Lingvistilised määratlused (märgendid) defineeritakse hägusate hulkadena, kusjuures ühe muutuja lingvistilised märgendid moodustavad termihulga. Hulgad, mis on defineeritud ühe termihulga raames, peavad loomulikult omama sama dimensiooni ehk olema sama universaalhulga alamhulgad.

Antud juhul omavad kõik muutujad sama dimensiooni - kõik nad on temperatuurid, kõik nad on mõõdetavad kraadides.

Millal on näiteks toa temperatuur kõrge? See on defineeritud muutuja *toatemperatuur* vastavale lingvistilisele märgendile (“kõrge”) vastava hägusa hulgaga, mis määratleb, et kõik temperatuuri  $t^{\circ}$  väärtused, mille kuuluvus sellesse hulka (ehk liikmesfunktsiooni  $\mu_{\text{kõrge toatemperatuur}}(t^{\circ})$  väärtus) on suurem nullist, on kõrge temperatuurid.

Joonis 2-1 kujutab toatemperatuuri lingvistilisi märgendeid kujutavate hägusate hulkade liikmesfunktsioone:

joonis 2-1



Neist lingvistilistest märgenditest moodustub universaalhulga *temperatuur* ( $t^{\circ}$ ) termihulk *toatemperatuur*. Joonisel on näha kuidas on toatemperatuuri väärtused lahterdatud kolme lingvistilise märgenditega “madal”, “paras” ja “kõrge” määratud universaalhulga *temperatuur* alamhulkadesse. Kuna need alamhulgad on hägusad, on ka lahterdus hägus. Viimasest tuleneb, et ühe ja sama arvulise väärtusega temperatuur saab kuuluda mitmesse ühe

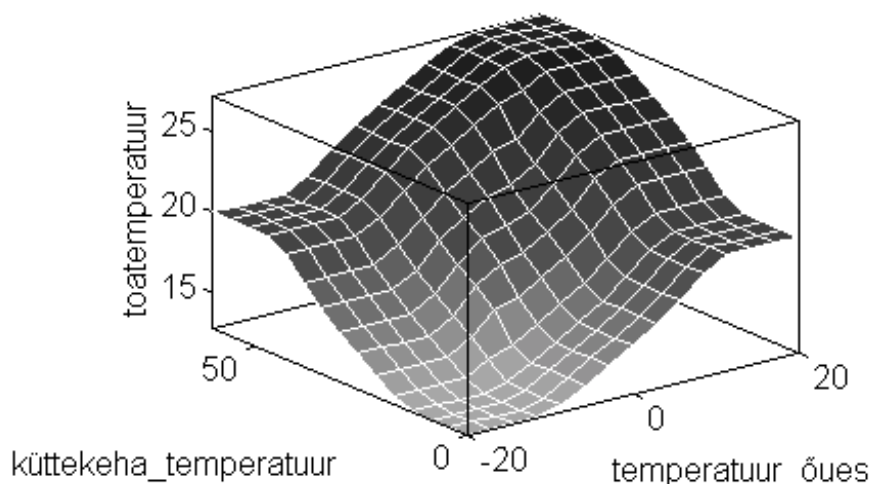
termihulga alamhulka. Konkreetset juhul on eeldatud, et *toatemperatuur* on määratav piirides [8, 32].

Analoogiliselt defineeritakse lingvistilised märgendid teistele süsteemi muutujatele *temperatuur õues* ja *küttekeha temperatuur*. Viimaste muutjate muutujate muutumiskiirkonnad võivad olla hoopis erinevad *toatemperatuuri* omast ja nii võib "madal küttekeha temperatuur" arvuliselt olla hoopis midagi muud kui "madal *toatemperatuur*".

## 2.4. Hägusad reeglid

Hägus mudel mida vaatlesime, peegeldab intuiitvset modelleerimismehhanismi, mida kasutab inimene, kel puudub võimalus süsteemi analüütiliselt kirjeldada. Antud mudel oli küll äärmiselt primitiivne, kuid ei maksa arvata et ta täiesti kasutu on.

joonis 2-2



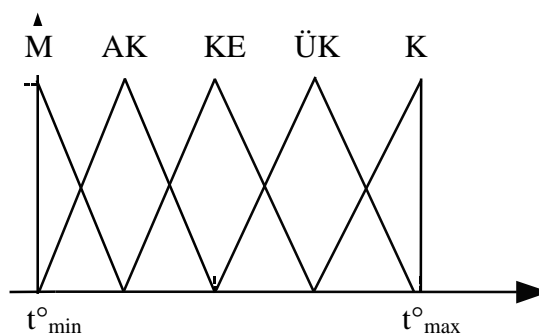
Joonis 2-2 esitab seost, mille antud mudel loob sisend- ja väljundmuutujate vahel, kusjuures sisendmuutujad *küttekeha temperatuur* ja *temperatuur õues* võivad muutuda vastavalt piirides [0, 60] ja [-20, 20]. Nagu näeme on seos oma olemuselt mittelineaarne ja see on hägusatele mudelitele üldiselt iseloomulik.

Nii keerukamate mudelite koostamiseks kui olemasolevate täiustamiseks vajame me juba rohkem teadmisi, ehk teisisõnu mahukamat reeglibaasi, millega mudelit täpsemalt kirjeldada. Esmaseks allikaks reeglite tuletamisel on inimekogemus, mis põhijoontes võib mingit süsteemi kujutada üsna täielikult.

Süsteemi kirjeldavate reeglite arv on piiratud muutujate hägusate lahterduste arvu korrutisega. Tabel 2-1 kirjeldas mudelit, mille puhul oli tegu kokku 7 lingvistilise märgendiga, kusjuures muutujad *temperatuur õues* ja *küttekeha temperatuur* olid lahterdatud kaheks, *toatemperatuur* kolmeks hägusaks alamhulgaks. See annab meile maksimaalselt  $2 \times 2 \times 3 = 12$  reeglit. Et kõikvõimalikud reeglid mõtet ei oma, siis tuleb reeglite lisamise võimaldamiseks suurendada muutujate lahterduste arvu.

Sisaldagu termihulgad *toatemperatuur*, *temperatuur õues* ja *küttekeha temperatuur* lingvistilisi märgendeid "madal" (M), "alla keskmise" (AK), "keskmise" (KE), "üle keskmise" (ÜK) ja "kõrge" (K). Vastavate hägusate hulkade liikmesfunktsioone kujutab joonis 2-3.

joonis 2-3



Laiendame oma reeglibaasi (tabel 2-2), kasutades ruumi kokkuhoiu mõttes kompaktsemat reeglite üleskirjutusviisi:

tabel 2-2

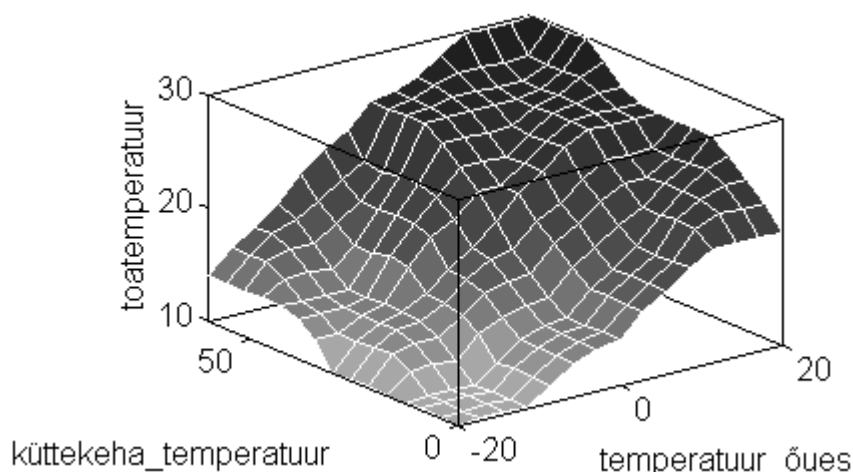
← küttekeha temperatuur →

	M	AK	KE	ÜK	K
M	M	M	M	AK	AK
↑ AK	M	M	AK	AK	KE
temp. KE	AK	AK	KE	ÜK	ÜK
õues ÜK	KE	KE	ÜK	ÜK	K
↓ K	KE	ÜK	ÜK	K	K

Iga tabeli väli on interpreeritav kui reegel:

**KUI** *temperatuur õues* on (rea päis) **JA** *küttekeha temperatuur* on (veeru päis) **SIIS** *toatemperatuur on* (tabeli väli)

joonis 2-4



Seost, mille antud reeglibaasiga hägus mudel loob sisend- ja väljundmuutujate vahel kujutab joonis 2-4. Näeme, et seost kujutav pind on tänu hägusate lahterduste arvu suurendamisele muutunud “krobelisemaks”, mis tähendab tegelikult seda, et seos on detailsemalt defineeritud.



## 2.5. Mudeli väljundi järelalusalgorithm

Järelalusalgorithm seisneb ligikaudse järeldamise meetodika rakendamises. Rajanegu meiepoolne järelalusalgorithmi kirjeldus 2 sisendi ja 1 väljundiga süsteemil, mille reeglid avalduvad järgmiselt:

**KUI**  $U_1$  on  $B_{i1}$  **JA**  $U_2$  on  $B_{i2}$  **JA** ... **JA**  $U_r$  on  $B_{ir}$  **SIIS**  $V$  on  $D_i$

Kui sellise m-reeglist koosneva hägusa mudeli sisendeiks on sisendmuutujate väärtused  $U_1 = x_1^*$ ,  $U_2 = x_2^*$ , ...,  $U_r = x_r^*$  siis tuleb neile vastava mudeli väljundi  $V$  määramiseks läbi teha järgmine protseduur:

1. Leida iga reegli kehtivusmäär.
2. Leida iga reegli väljund
3. Koondada kõikide reeglite väljundid üheks süsteemi väljundiks

Reegli kehtivusmäär on määratud reegli tingimuspoole komponentide rahuldatusena. Iga tingimus  **$U_1$  on  $B_{i1}$**  on interpreteeritav kui muutuja  $U_1$  liikmesfunktsiooni  $\mu_{B_{i1}}(x_1)$  väärtusena kohal  $x_1^*$  hägusa hulga  $B_{i1}$  suhtes, mis määrabki ära reegli kehtivuse määra. Samamoodi tuleb liikmesfunktsioonide väärtused arvutada tingimuse teistele komponentidele. Reegli tingimuspoole summaarse kehtivus- või tabamismäära väljaselgitamiseks tuleb arvutada liikmesfunktsioonide väärtuste ühisosa (tingimusi eraldab "JA"):

$$\tau_i = \mu_{B_{i1}}(x_1^*) \cap \mu_{B_{i2}}(x_2^*) \cap \dots \cap \mu_{B_{ir}}(x_r^*), \quad \text{avaldis 2-5}$$

kus  $\tau_i$  on i-nda reegli tingimuspoole tabamismäär sisendite  $U_1 = x_1^*$  ja  $U_2 = x_2^*$ , ...,  $U_r = x_r^*$  poolt, mis ühtlasi väljendab i-nda reegli rahuldatus määra.

Järgmiseks sammuks on iga reegli väljundi  $F_i$  määramine. Levinumaks meetodiks on nn. Mamdani meetod, kus reegli väljund  $F_i$  on defineeritud reegli tingimuspoole tabamismäära  $\tau_i$  ja järelaluspoole hägusa hulga  $D_i$  ühisosana.

$$F_i(y) = \tau_i \cap D_i \quad \text{avaldis 2-6}$$

Kolmandaks sammuks on üksikute reeglite väljundite kokkuvõtmine süsteemi väljundiks  $F$ , mis kujutab endast nende ühendit.

$$F(y) = \cup F_i(y) \quad \text{avaldis 2-7}$$

Viimane ongi ühtlasi hägusa mudeli reageering sisenditele  $x_1^*$ ,  $x_2^*$ , ...,  $x_r^*$ .

Hägusate hulkade ühend ja hägusate hulkade ühisosa on defineeritud meil vastavalt max ja min-operaatoritena (avaldised 1-3 ja 1-2). See ei ole ainus võimalus. Üldjuhul on hägusate hulkade ühend defineeritud t-normina ja ühisosa s-normina (t-kaasnormina).

T-norm on funktsioon, mis rahuldab tingimusi:

1.  $t(x,1) = x, t(x,0) = 0$
2.  $t(x,w) < t(y, z)$ , kui  $x < y, w < z$
3.  $t(x, y) = t(y, x)$
4.  $t(t(x, y), z) = t(x, t(y, z))$
5.  $x$ -i ja  $y$ -i drastiline korrutis  $\leq t(x, y) \leq \min(x, y)$

kus  $x, y, w, z \in [0,1]$

S-norm on funktsioon, mis rahuldab tingimusi:

1.  $s(x, 0) = x, s(x, 1) = 1$
2.  $s(x, w) < s(y, z)$ , kui  $x < y, w < z$
3.  $s(x, y) = s(y, x)$
4.  $s(s(x, y), z) = s(x, s(y, z))$
5.  $\max(x, y) \leq s(x, y) \leq x$ -i ja  $y$ -i drastiline korrutis

jällegi  $x, y, w, z \in [0,1]$

Seoses sellega on hägusa disjunktsioonina (ühend) max-opearaatori asemel võimalik kasutada ka teisi s-norme ning hägusa konjunktsioonina (ühisosa) min-opearaatori asemel teisi t-norme, mis on loetletud vastavates allikates [10, lk 25]. Seega on järelusmehhanismide osas olemas küllalt suur valik, kusjuures osa järelusalgoritme võivad osutada oluliselt lihtsustatuks võrreldes meie poolt toodud nn. Mamdani järelusalgoritmiga.

Sugeno tüüpi mudelite puhul, mille reeglite üldkuju oli antud avaldisega 2-5, saame iga reegli väljundi korrutades järeluspoole funktsiooni väärtust antud sisendite juures reegli kehtivusmääraga ning sellise mudeli summaarseks väljundiks on üksikute reeglite väljundite ühend.

## 2.6. Häguärastamine

Hägusa mudeli reaalses elus kasutuskõlblikuks tegemiseks on vajalik veel neljas samm, mis muudaks hägusa väljundi konkreetseks arvuks, kuna hägusa väljundhulgaga  $F$  pole suurt midagi peale hakata. Väljundhulgast  $F$  tuleb kasutada vaid ühte elementi  $y^*$ . Millist täpselt, selgitatakse välja tegevusega, mida nimetatakse häguärastamiseks ehk täppistamiseks.

Kaks levinumat täppistamismeetodit on raskuskeskme ning maksimumide keskmise meetod.

Raskuskeskme meetod defineerib hägusa hulga  $F$  täppistatud väärtuse järgmiselt:

$$y^* = \frac{\int yF(y)dy}{\int F(y)dy} \quad \text{avaldis 2-8}$$

Täppistatud väärtuse arvutamine lihtsustub märgatavalt, kui  $Y$  on lõplik ning liikmesfunktsioon  $F(y)$  diskreetne:

$$y^* = \frac{\sum_{j=1}^n F(y_j) y_i}{\sum_{j=1}^n F(y_j)}$$

avaldis 2-9

Maksimumide keskmise meetod defineerib hägusa mudeli väljundi selgistatud väärtuse kui kõigi, maksimaalset liikmesfunktsiooni väärtust omavate väljundi väärtuse keskmise.

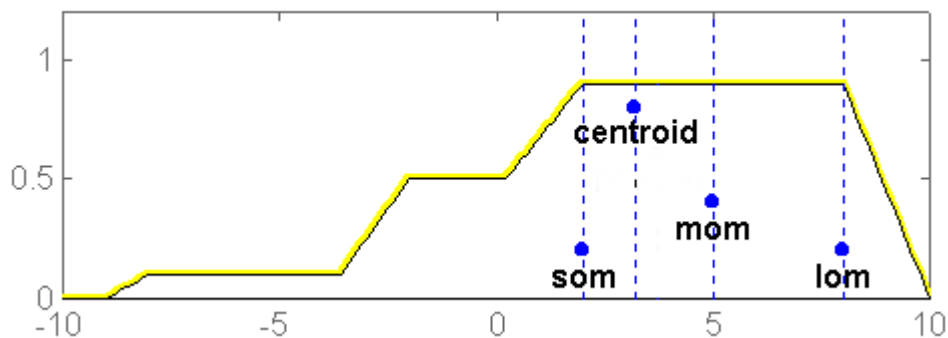
$$y^* = \frac{1}{q} \sum_{j \in J^*} y_j$$

avaldis 2-10

kus  $J^*$  on universaalhulga  $Y$  elementide hulk, mis omavad  $F(y)$  maksimumväärtust ja  $q$  nende arv.

Täppistamismeetodite illustreerimiseks on kujutatud (joonis 2-5) ühe hägusa hulga täppistatud väärtusi, mis on saadud erinevaid meetodeid rakendades. Lisaks siinkirjeldatud raskuskeskme meetodile (centroid) ja maksimumide keskmise meetodile (mom), on täppistatud väärtused arvatud ka kasutades minimaalse (som) ja maksimaalse maksimumi (lom) meetodeid.

joonis 2-5



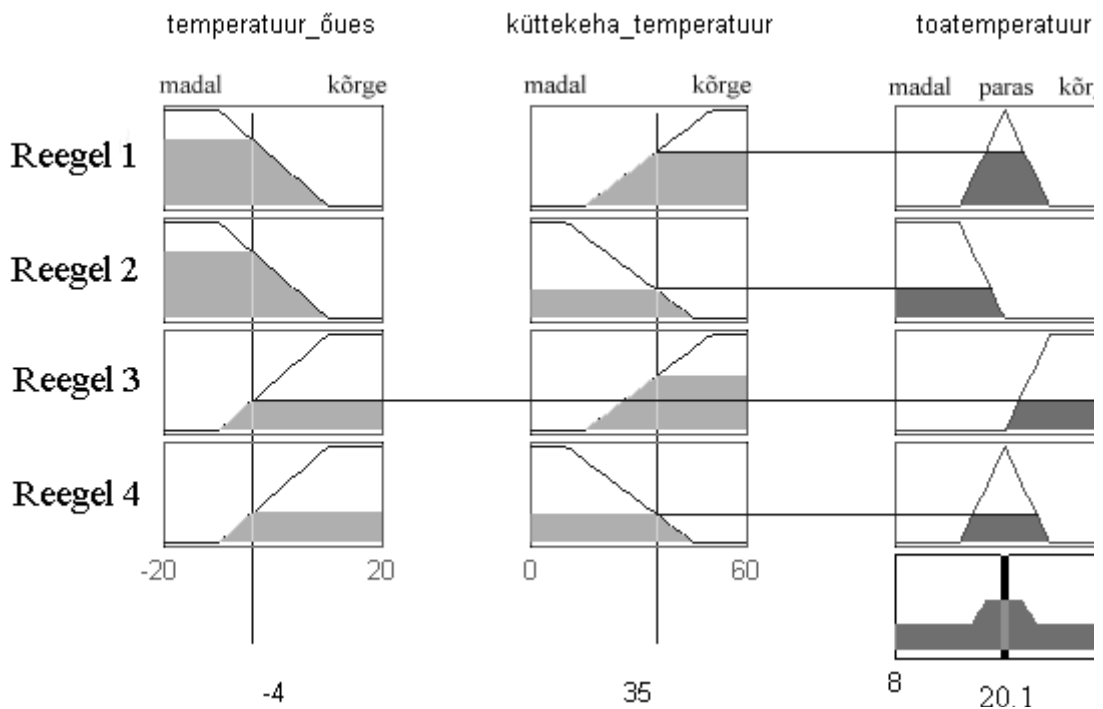
### 2.6.1.Näide

Järeldusalgoritmi illustreerimiseks on kasutatud esimese näite andmeid ning hägus järeldamine on kujutatud graafiliselt (joonis 2-6), kus kajastuvad kõik hägusa mudeli teatud sisenditele vastava väljundi saamiseks vajalikud sammud.

Lingvistilistele märgenditele *madal temperatuur õues*, *kõrge temperatuur õues*, *madal küttekeha temperatuur*, *kõrge küttekeha temperatuur*, *madal toatemperatuur*, *paras toatemperatuur* ja *kõrge toatemperatuur* vastavad hägusad on hulgal on vastavalt  $B_{11}$ ,  $B_{12}$ ,  $B_{21}$ ,  $B_{22}$ ,  $D_1$ ,  $D_2$ ,  $D_3$ .

Etteantud sisenditele  $x_1^* = -4^\circ\text{C}$ ,  $x_2^* = 35^\circ\text{C}$  vastava mudeli väljundi (mis on vastavalt joonisele  $20,1^\circ\text{C}$ ) tuletuskäik on toodud joonisel 2-6.

joonis 2-6



## 2.7. Liikmesfunktsioonid

Järeldusalgoritmi peamine efekt seisneb hägusas reeglibaasis sisalduva kvalitatiivse informatsiooni rakendamises viisil, mis annab iga antud sisendkombinatsiooni korral mudeli väljundi arvulise väärtuse  $V = f(U_1, U_2, \dots, U_r)$ . See seob mudeli väljundi  $V$  arvuväärtused üheselt sisenditega  $U_1, U_2, \dots, U_r$  arvuliste väärtustega. Lõpliku arvu sisendite võimalike väärtuste korral (mis eeldab diskreetset mudelit) on viimast seost võimalik kujutada mudeli väljundi väärtuste tabeli kujul.

Kui sisendite hulgad  $X_1$  ja  $X_2$  sisaldavad vastavalt elemente  $u_{11}, u_{12}, \dots, u_{1n}$  ja  $u_{21}, u_{22}, \dots, u_{2m}$  siis saab koostada väljundhulga  $Y$  elementidest koosneva tabeli:

tabel 2-3

	$u_{21}$	$u_{22}$	.....	$u_{2n}$
$u_{11}$	$v_{11}$	$v_{12}$	.....	$v_{1n}$
$u_{12}$	$v_{21}$	$v_{22}$	.....	$v_{2n}$
.....	.....	.....	.....	.....
$u_{1m}$	$v_{m1}$	$v_{m2}$	.....	$v_{mn}$

Sellise väljaarvutatud võimalike mudeli väljundite tabeli kasutamine lihtsustab hägusa mudeli sisemise struktuuri esitust. Eelnevalt kindlaksmääratud reeglibaasi ja liikmesfunktsioonide korral saab tabeli sisu arvutada mudeli konstrueerimise raames. Mudeli väljundi kindlaksmääramisel tuleb konkreetse sisendi väärtus samastada talle lähimale tabelis leiduva sisendi väärtusega (diskreetida).

Hägusate hulkade liikmesfunktsioonid võib sellest tulenevalt esitada vektoritena, mille elementideks on liikmesfunktsiooni diskreetsed väärtused. Vektori elementide arv ehk dimensioon on määratud vastava termihulga diskreetimisintervalliga, mille suuruse määramisel üritatakse leida mõistlikku kompromissi püüde lähendada diskreetset liikmesfunktsiooni võimalikult täpselt tegelikule liikmesfunktsioonile ja piirangute vahel, mis on seatud vektori mahule. Otsides kompromissi nende kahe, üksteisele vastukäiva nõude vahel, võib kõne alla tulla ka ebavõrdsete intervallidega diskreetimine. Järeldusalgoritmist tuleneb, et reeglite järelduspoolele vastavate hägusate hulkade diskreetimise viisist sõltub järeldusalgoritmi realiseerimise kiirus. Mida täpsemalt on hägus hulk esitatud, seda rohkem on temaga vaja läbi viia arvutusi. Seega omab õige diskreetimine tähtsust ka arvutuslikus aspektis.

Liikmesfunktsioone võib samas esitada ka funktsioonidena. Levinumate liikmesfunktsioonide analüütilised kujud on järgmised:

$$\text{kolmnurkne: } \mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right), \quad \text{avaldis 2-11}$$

kus  $a$ ,  $b$  ja  $c$  on funktsiooni parameetrid vastavalt joonisele 1-3a.

$$\text{trapets: } \mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right), \quad \text{avaldis 2-12}$$

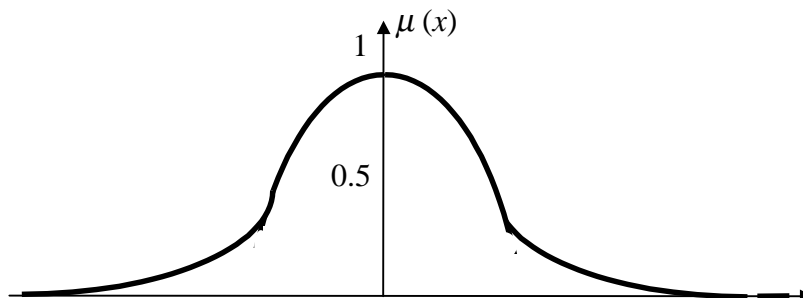
kus  $a$ ,  $b$ ,  $c$  ja  $d$  on funktsiooni parameetrid vastavalt joonisele 1-3b.

Gaussi funktsioon (joonis 2-7) avaldub seevastu täies ulatuses analüütiliselt:

$$\mu(x) = e^{-\frac{(x-c)^2}{2\sigma^2}}, \quad \text{avaldis 2-13}$$

kus  $c$  ja  $\sigma$  on funktsiooni kuju determineerivad parameetrid

joonis 2-7



Teiste kasutatavate liikmesfunktsioonide tüüpide () ja nende funktsionaalse kuju kohta võib andmeid leida kirjandusest [15].

## 2.8. Mudelite konstrueerimine

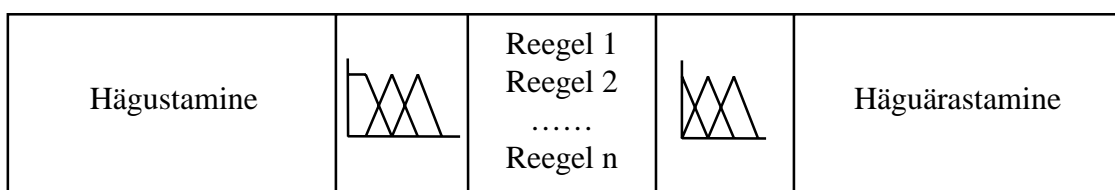
Seniesitatu põhjal võib tuua välja hägusa mudeli konstrueerimise põhilised etapid. Olemasolev olukord dikteerib meile sisend-väljundmuutujad ja nende muutumispiirkonnad mida kasutame. Mudeli täiustamise huvides on vahest kasulik võtta oma mudelisse ka

suuruseid, mille mõju mudeli väljundile pole ehk nii ilmne, ent siiski on muutujate hulk ja nende muutumispiirkonnad enamasti paika pandud protsessi või objektiga ja midagi üleloomulikku siin ette võtta ei saa.

Pisut teine on situatsioon liikmesfunktsioonide arvu, kuju ja hägusa järeldusalgoritmi valikul ning loomulikult reeglite komplekti kokkupanekul, kus konstrueerijal on vabamad käed. Järeldusalgoritmi ja liikmesfunktsioonide tüüpide ja arvu valikul võib rääkida nende eelnevast, a priori defineerimisest.

Mudeli lõplikul täpsustamisel on konstrueerija vahenditeks liikmesfunktsioonide parameetrid ja reeglikomplekt. Neid võiks klassikalise juhtimisteooria keeles kutsuda koguni hägusa mudeli häälestusparameetriteks. Selguse mõttes on joonisel 2-8 toodud hägusa mudeli struktuur

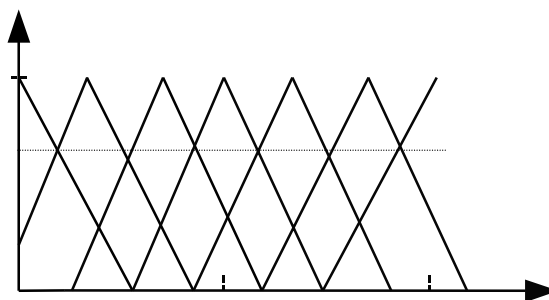
joonis 2-8



### 2.8.1. Hägus lahterdamine

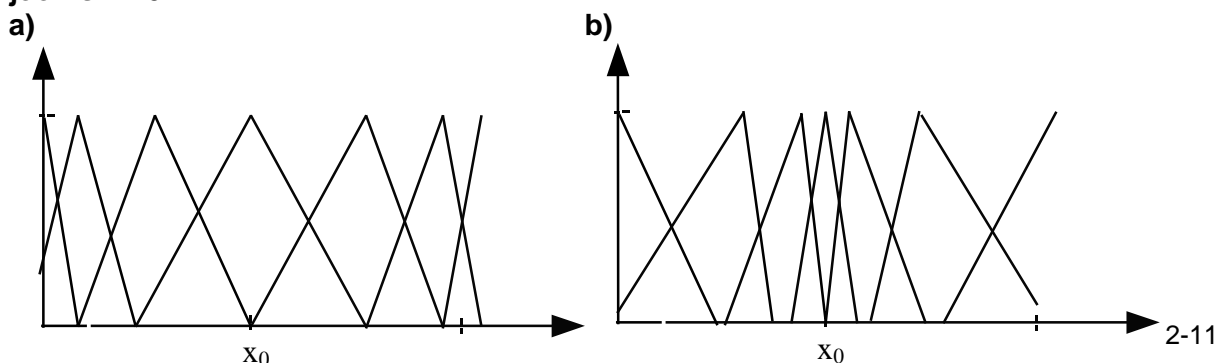
Mil moel teostada hägusat lahterdamist? Kui me oleme kindlaks määranud mitmeks liikmesfunktsiooniks me muutuja lahterdame, siis esimeseks võimaluseks on ühtlane lahterdus, mis tähendab, et mingi muutuja muutumispiirkond kaetakse ühtlaselt hägusate hulkadega, teatud ülekattumisega, mis joonisel 2-9 on kujutatud katkendjoonega. Üldiselt on see tase suurem liikmesfunktsiooni väärtusest 0,25 (joonisel 0,75).

joonis 2-9



Et tõsta või vähendada mudeli tundlikkust mingi (kriitilise) punkti(de) läheduses tuleb kõne alla ka ebahütlane jaotus. Nii kujutab joonis 2-10a olukorda, kus vastava hägusa lahterdusega on suurendatud mudeli stabiilsust punkti  $x_0$  ümbruses ning joonis 2-10b olukorda, kus on tõstetud tundlikkust samas punktis.

joonis 2-10



Üheks võimaluseks veelgi stabiilsust tõsta oleks näiteks kriitilisele punktile vastav hägus hulk defineerida trapetsikujulise liikmesfunktsioonina, mis tänu trapetsi tipu lamedusele tekitab kriitilise punkti ümber teatud tundetuse tsooni.

## 2.8.2. Normeerimistegurid

Käepäraseks vahendiks mudeli häälestamisel liikmesfunktsioonide kaudu on normeerimistegurid. Kui mingi muutuja  $x$  muutub piirides  $[-a_x, a_x]$ , siis saab defineerida muutuja  $x$ -i normeeritud muutumispiirkonna kus  $x$ -i normeeritud väärtus  $x^*$  võtab väärtusi normeeritud muutumispiirkonnast  $[-a_x^*, a_x^*]$ . Defineerime normeerimisteguri

$$k_x = \frac{a_x^*}{a_x} \quad \text{avaldis 2-14}$$

Iga normeeritud  $x$ -i väärtus  $x^* = k_x \cdot x$  avaldis 2-15

Põhimõtteliselt võivad normeeritud muutumispiirkonnad olla identsed muutujate tegelike muutumispiirkondadega, kuid enamikes rakendustes on nad piiratud intervalliga  $[-1, 1]$ .

Rakendades taolisi normeeritud sisendeid mudelile saame viimase reageeringuna täppistatud väärtuse  $y^*$  mis kuulub normeeritud universaalhulka  $[-a_y^*, a_y^*]$ . Tegeliku mudeli väljundi  $y$  saamiseks tuleb  $y^*$  denormeerida, kasutades järgmist seost:

$$y = k_y \cdot y^* \quad \text{avaldis 2-16}$$

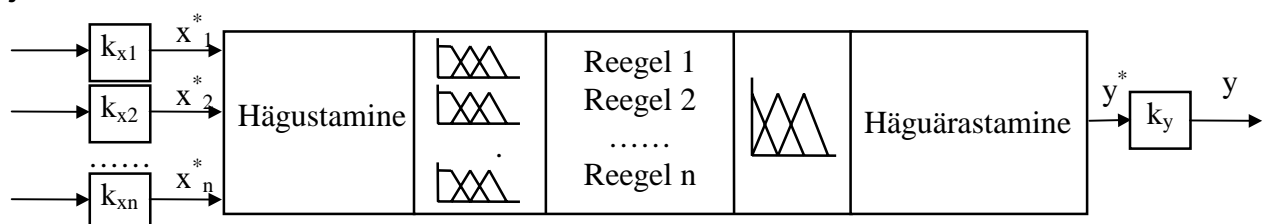
kus

$$k_y = \frac{a_y}{a_y^*} \quad \text{avaldis 2-17}$$

$y$  kuulub juba normeerimata universaalhulka  $[-a_y, a_y]$ .

Lisaks võimalusele defineerida kõik mudeli sisendid ja väljund normeeritud piirkondades  $[-1, 1]$  annab taoline lähenemine üsna mugava vahendi mudeli häälestamiseks. Vaadeldgem normeerimisteguritega varustatud hägusat mudelit, mille struktuuri kujutab joonis 2-11:

joonis 2-11



Iga normeerimistegur määrab ära muutuva muutumiskiirkonna ent samas (eeldades ühtlast lahterdust) muutub muutumiskiirkonna muutusega võrdeliselt ka ala, mille iga üksik liikmesfunktsioon hõlmab. Seega on võimalik normeerimistegurite kaudu muuta liikmesfunktsioonide parameetreid ilma hägusasse mudelisse tungimata. Muutujate liikmesfunktsioonide sõltumatuks häälestamiseks tuleks kasutada juba mittelineaarseid normeerimistegureid, mis on küll tülikas, ent teatud juhtudel võib anda suurt efekti. Hägusate mudelite konstrueerimine on põhjalikuma vaatluse all järgmises peatükis.

## **3. Hägus modelleerimine**

### **3.1 Sissejuhatus**

Eelmises kahes peatükis tutvustasime lähenemist, mis annab vahendid selliste süsteemide käitumise kirjeldamiseks, mis on liiga keerulised või puudulikult defineeritud, et selleks saaks kasutada täpset matemaatilist analüüsi. Lähenemist iseloomustab kolm printsiipi:

1. Numbriliste muutujate asendamine lingvistiliste muutujatega
2. Lihtsate muutujatevaheliste seoste kirjeldamine hägusate reeglitega
3. Keeruliste seoste kirjeldamine hägusate algoritmidega

Selliste mudelite konstrueerimine eeldab teatud metodoloogia olemasolu, kus hägusa modelleerimise all mõistame me olemasolevale süsteemile vastava hägusa mudeli loomist, vastavalt meie teadmiste süsteemist.

Hägusate mudelite konstrueerimise varasel ajajärgul püüti hägusaid mudeleid tuletada vahetult ekspertteadmiste baasil (nn. otsemeetod - direct approach). Sel viisil kirjeldatakse süsteem lingvistiliselt, kasutades inimkeele vahendeid, mis seejärel teisendatakse hägusmudeli formaalkujusse.

Lingvistiline kirjeldus on oma olemuselt subjektiivne, baseerudes süsteemist omataval teadmistel. Ekspertteadmised esitatakse loogiliste reeglitena. Seda meetodit võib vaadelda kui süsteemiteoorias kasutusel oleva modelleerimise kvalitatiivset versiooni.

Kuna selle meetodi edukus sõltub puhtalt eksperdi poolt antava süsteemi funktsioneerimise kirjelduse kvaliteedist, on ta muidugi teatavate piirangutega. Otsemeetodil pole süsteemi funktsioneerimise kvantitatiivsed konstanteeringud eriti kõlblikud mudeli struktuuri või parameetrite määramiseks. Kui eksperdi teadmised süsteemist on puudulikud, on resultaatiks kehv mudel.

Loogiliseks jätkuks on enam formaalsete meetodite tuletamine, mis suudaksid kasutada olemasolevat informatsiooni inimkogemuste täiendamiseks või isegi uute seoste tuletamiseks.

Teine hägusate mudelite arendamise suund, mis on inspireeritud klassikalisest süsteemiteooriast ja närvivõrkude arengust, baseerub sisend-väljundandmete kasutamisel. Süsteemiteooria keeles võiks seda meetodit nimetada süsteemi identifitseerimiseks.

Hägusa mudeli konstrueerimine sel meetodil koosneb kahest olulisest etapist. Esimene etapp on hägusa mudeli struktuuri identifitseerimine ja teine hägusa mudeli parameetrite hindamine.

Laias laastus tähendab struktuuri identifitseerimine sisend- ja väljundmuutujate, nendevaheliste seoste (reeglite struktuuri), reeglite arvu ja sisend- ning väljundmuutujate hägusa lahterduse kindlaksmääramist. Üldjuhul on struktuuri identifitseerimine keeruline ja äärmiselt puudulikult defineeritud protsess. Osaliselt lihtsustub see, kui teatud süsteemi ja olemasolevaid andmeid seostavad ekspertteadmised on saadaval.

Eksperimendiandmetest hägusa mudeli struktuuri tuletamisel on üheks võimaluseks nn. shabloonil baseeruvad meetodid. See meetod kasutab ära nii eksperimendiandmeid kui ekspertteadmisi. Süsteemiekspert annab sabloonsed lingvistilised määratlused, mida kasutatakse sisend- ja väljundpiirkondade lahterdamiseks. Need lahterdused on aluseks hägusa süsteemimudeli potentsiaalsete reeglite tuletamisel. Informatsiooni sisendi ja väljundi kohta kasutatakse seejärel reeglite kaalude või tõenäosuste tuletamiseks

Näiteks võib seostada üksiku reegli kaalu tema esinemise sagedusega sisend-väljundinformatsioonis.

Kui eksperteadmisi pole võimalik kasutada, tuleb süsteemi struktuur identifitseerida paljalt eksperimendiandmete najal. Selleks kasutatakse klasterdamise meetodeid.

Parameetrite idendifitseerimine hägusmodelleerimises on tihedalt seotud hägusate hulkade liikmesfunktsioonide või hägusa mudeli reeglibaasiga määratud sisend- ja väljundmuutujate vahelise hägusa seose idendifitseerimisega.

Tänu teatud tüüpi hägusate mudelite struktuurile on võimalik ära kasutada närivõrkude alalt tuttavaid põhimõtteid ning panna hägusat mudelit oma parameetreid õppimise abil tuletama.

## 3.2 Otsemeetod

Antud meetod sisaldab järgmisi etappe.

1. sisend- oleku- ja väljundmuutujate valimine
2. muutujate muutumispirkondade määramine
3. muutujate lingvistiliste märgendite ja nende vastavate hägusate hulkade determineerimine
4. süsteemi muutujate vahelisi seoseid kirjeldavate lingvistiliste reeglite konstrueerimine
5. hägusale mudelile sobiva järeldusalgoritmi valimine
6. mudeli adekvaatsuse hindamine

Hägusa mudeli konstrueerimine otsemeetodil on rohkem intuitsiooni kui täpse metodoloogia rakendamise küsimus. Ehkki täpne teooria puudub, kirjeldasime me mõningaid lähtekohti jaotises 2.8.

## 3.3 Shabloonil baseeruv hägus mudellerimine

Siin kasutatakse mudeli idendifitseerimisel kahte tüüpi informatsiooni. Esimene on kogum süsteemi sisendi- ja väljundi väärtuspaare  $(x_k, y_k)$ ,  $k = (1, K)$  (eeldame siin lihtsuse mõttes, et tegu on ühe sisend- ja ühe väljundmuutujaga hägusa süsteemiga), kus  $y_k$  on süsteemi sisendile  $x_k$  vastav süsteemi väljundi väärtus. Teiseks eeldab meetod eksperdi olemasolu, kes oma kogemuste põhjal defineerib lingvistilised väärtused, mis lahterdavad sisend- ja väljundmuutujate muutumispirkonnad vastavateks hägusateks hulkadeks.

Olgu sisendmuutuja  $X$  muutumispirkond lahterdatud hägusateks hulkadeks  $B_1, B_2, \dots, B_s$ .  $B_i$ -d on siin lingvistilised märgendid nagu *väga madal*, *madal*, *keskmine* jne, mis on defineeritud näiteks vastavate kolmnurksete hägusate  $X$ -i alamhulkadena. Sarnaselt on hägusateks alamhulkadeks  $D_1, D_2, \dots, D_t$  lahterdatud väljundpiirkond  $Y$ .

Oletame et iga  $B_i$  ja  $D_j$  paar annab potentsiaalse reegli, mis avaldub järgmisel kujul:

**KUI**  $U$  on  $B_i$  **SIIS**  $V$  on  $D_j$ ,  $i = (1, s)$ ,  $j = (1, t)$

**avaldis 3-1**

Tähistame sellised shabloonsetest  $B_i$ -st ja  $D_j$ -st tuletatud reeglid järgmiselt:  $R(B_i, D_j)$  ja nimetame sellist tüüpi reegleid elementaarreegliteks. On selge, et antud hägusa lahterduse juures saame me tuletada maksimaalselt  $m = s \cdot t$  elementaarreeglit.

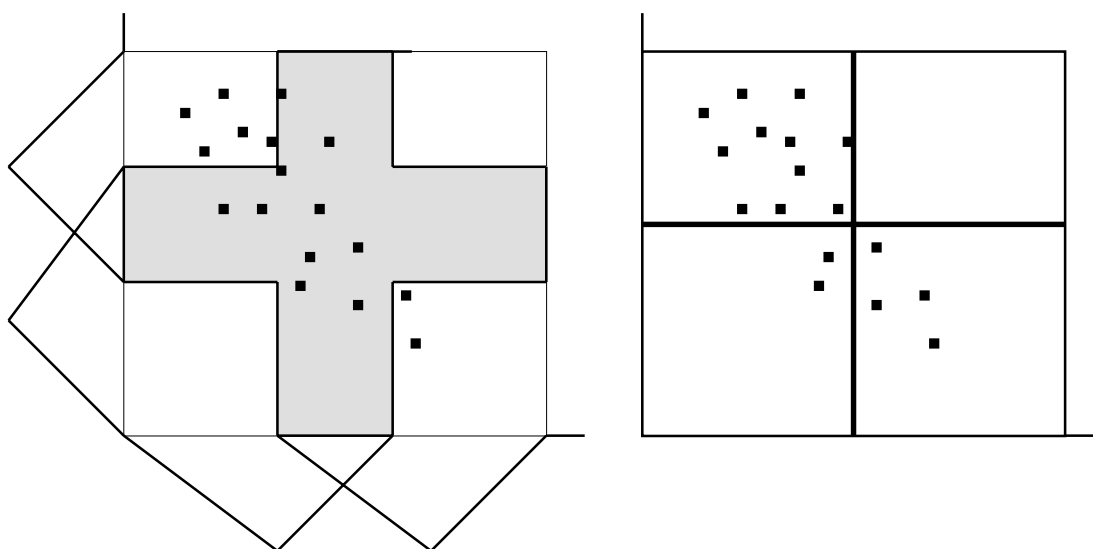
**tabel 3-1**

$D_t$	$R(B_1,$	$R(B_2,$	.....	$R(B_i,$	.....	$R(B_s,$
	.....	.....	.....	.....	.....	.....
$D_j$	$R(B_1,$	$R(B_2,$	.....	$R(B_i,$	.....	$R(B_s,$
	.....	.....	.....	.....	.....	.....
$D_2$	$R(B_1,$	$R(B_2,$	.....	$R(B_i,$	.....	$R(B_s,$
$D_1$	$R(B_1,$	$R(B_2,$	.....	$R(B_i,$	.....	$R(B_s,$
	$B_1$	$B_2$		$B_i$		$B_s$

Elementaarreeglite kogum  $R(B_i, D_j)$  ( $i = (1,s), j = (1,t)$ ) on nüüd shablooniks, mille peale me konstrueerime hägusat süsteemi kirjeldava reeglibaasi. Pole reaalne, et iga elementaarreegel  $R(B_i, D_j)$  kujutab endast modelleeritava süsteemi jaoks täielikult kehtivat reeglit. Me kirjeldame iga sellist reeglit kaaluga, mis väljendab antud elementaarreegli kehtivusastet antud süsteemis. Kaalude määramisel on aluseks sisend-väljundinformatsiooni kogum. Viimase operatsiooni teostamiseks on mitmeid teid.

Esimese variandi järgi omistatakse igale hägusale piirkonnale  $B_i \times D_j$  kaal  $w_q$ , mis on võrdne sisend-väljundpaaride  $(x_k, y_k)$  arvuga, mis satuvad sellesse piirkonda. Tänu piirkondade hägususele on võimalik, et osa paare kuulub üheaegselt mitmesse piirkonda. Probleemist võib üle saada näiteks nii nagu on näidatud joonisel 3-1,

joonis 3-1



kus sisendi ja väljundi hägusate alamhulkade ülekattumised asendatakse mittehägusate eraldajatega, teisisõnu jaotus  $B_i \times D_j$  asendub jaotusega  $C_q = b_i \times d_j$ .  $q$ -ndasse piirkonda kuuluvate sisend-väljundpaaride  $(x_k, y_k)$  arv  $n_q$  määrab sellele piirkonnale vastava reegli kaalu  $w_q$  järgmiselt:

$$w_q = n_q / n \quad q = (1, st), \quad \text{avaldis 3-2}$$

kus

$$n = \sum_{q=1}^{st} n_q \quad \text{avaldis 3-3}$$

on sisend-väljundpaaride koguarv. Osades tsoonides võib  $w_q$  olla null, mis ilmselt näitab, et antud reeglile konkreetses hägusas mudelis kohta ei ole.

Kirjeldatud meetodiga transformeeritakse hägus võrgustik tasandil  $X \times Y$  mittehägusaks. Eksisteerib ka alternatiivne meetod, mis sellist hägusust ei ignoreeri.

Antud sisend-väljundpaar  $(x_k, y_k)$  rahuldab elementaarreeglit, mis on toodud avaldisega 3-1 siis, kui selle reegli tingimuspoole tabamismäär  $\tau_i = B_i(x_k)$ ,  $i = (1, s)$  ning väljundi  $y_k$  liikmesus reegli järelduspoole hägusas alamfunktsioonis  $\gamma_j = D_j(y_k)$ ,  $j = (1, t)$  omavad mittenulliseid väärtusi.

Defineerime seda rahuldatuse astet väljendava sobivuse määra:

$$\delta_{ij}(k) = \tau_{ik} \gamma_{jk} \quad \text{avaldis 3-4}$$

$\delta_{ij}(k)$  kujutab endast vahemikku  $[0, 1]$  jäävat arvu, mis asetseb kolmemõõtmelises maatriksis, mille dimensioonid on  $s \times t \times K$ .  $s \times t$  on ühtlasi elementaarreeglite koguarv.

Tänu piirkondade  $B_i \times D_j$  hägususele võib üks sisend-väljundpaar  $(x_k, y_k)$  omada mittenullist sobivuse määra rohkem kui ühe reegli puhul. Normeerime ühe sisend-väljundpaari sobivusmäära. Normeeritud sobivusmäär  $v_{ij}(k)$  saadakse  $\delta_{ij}(k)$  jagamisel ühe sisend-väljundpaari kohta tuleva summaarse (kõikide reeglite) sobivusmääraga:

$$v_{ij}(k) = \frac{\delta_{ij}(k)}{\sum_{i=1}^s \sum_{j=1}^t \delta_{ij}(k)} \quad \text{avaldis 3-5}$$

Järgnevalt summeeritakse iga elementaarreegli kohta tulevad tabamismäärad:

$$v_{ij} = \sum_{k=1}^K v_{ij}(k) \quad \text{avaldis 3-6}$$

Tulemuseks on  $s \times t$  maatriks,

$$\begin{bmatrix} v_{11} & v_{12} & \dots & v_{1t} \\ v_{21} & v_{22} & \dots & v_{2t} \\ \dots & \dots & \dots & \dots \\ v_{s1} & v_{s2} & \dots & v_{st} \end{bmatrix},$$

mille elemendi reanumber  $i$  määrab sisendi liikmesfunktsiooni, veerunumber  $j$  väljundi liikmesfunktsiooni ning elemendi väärtus  $v_{ij}$  vastavaid liikmesfunktsioone siduva reegli tõenäosuse. Seega võib öelda, et kui sisend tabab liikmesfunktsiooni 2 mingi tabamismääraga, on väljund tõenäosusega  $v_{21}$  esimeses liikmesfunktsioonis, tõenäosusega  $v_{22}$  teises liikmesfunktsioonis jne.

Et selliste tõenäosuste summa ei oleks suurem kui 1, summeeritakse  $i$ -ndat sisendi liikmesfunktsiooni siduvate reeglite tabamismäärad

$$v_i = \sum_{j=1}^t v_{ij}, \quad i = (1, s) \quad \text{avaldis 3-7}$$

ning normeeritakse  $v_{ij}$  -d nende järgi

$$P_{ij} = \frac{v_{ij}}{v_i} \quad \text{avaldis 3-8}$$

kus iga  $P_{ij}$  kujutab endast elementaarreegli  $R(B_i, D_j)$  esinemise tõenäosust ehk kaalu.

Seega saame lõpptulemusena  $s \cdot t$  reeglit kaaludega  $P_{ij}$ , mis moodustavadki modelleerimise tulemusena saadud mudeli reeglibaasi.

### 3.4 Modelleerimine närvivõrgulaadse mudeliga

Lihtsustatud järelalusalgoritme kasutavate hägusate mudelite põhiline eelis seisneb selles, et nad annavad süsteemi sisendite ja väljundite vahelise seose analüütilisel kujul. See omadus lubab hägusaid mudeleid esitada kui närvivõrke ja nende parameetrite identifitseerimiseks kasutada õppimisalgoritme.

Hägusa mudeli ja närvivõrgu vahelise sarnasuse esiletoomiseks vaatleme mitme sisendi ja ühe väljundiga lingvistilist mudelit:

**KUI**  $U_1$  on  $B_{i1}$  **JA**  $U_2$  on  $B_{i2}$  **JA** ... **JA**  $U_r$  on  $B_{ir}$  **SIIS**  $V$  on  $D_j$ ,  $i = (1, m)$  **avaldis 3-9**

Sellise mudeli täppistatud väljund, vastavalt lihtsustatud hägusale järelalusalgoritmile on:

$$y = \frac{\sum_{i=1}^m \tau_i y_i^*}{\sum_{i=1}^m \tau_i}, \quad \text{avaldis 3-10}$$

kus  $y^*$ -ga on tähistatud reeglite järelaluspoole hägusate hulkade  $D_i$  täppistatud väärtused (raskuskeskmed) ja  $\tau_i$ -d,  $i = (1, m)$  on reeglite tabamismäärad:

$$\tau_i = B_{i1}(u_1) \cap B_{i2}(u_2) \cap \dots \cap B_{ir}(u_r) \quad \text{avaldis 3-11}$$

antud mittehägusate sisendväärtuste  $u_1, u_2, \dots, u_r$  korral. Selline lihtsustatud järelalusmeetod annab küll hägusa mudeli täppistatud väljundi analüütilise funktsioonina erinevate reeglite tabamismääradest, kuid mitte funktsioonina sisenditest. Selleks tuleks asendada min-operaator produkt-operaatoriga ja lisaks eeldada, et hägusa mudeli tingimuspoole hägusad alamhulgad on Gaussi funktsioonid:

$$B_{ij}(x_j) = \exp\left(-\frac{1}{2}\left(\frac{x_j - x_{ij}^*}{\sigma_{ij}}\right)^2\right) \quad \text{avaldis 3-12}$$

parameetritega  $x_{ij}^*$  ja  $\sigma_{ij}$ . Sellest tulenevalt saame, et:

$$\begin{aligned} \tau_i &= B_{i1}(u_1) B_{i2}(u_2) \dots B_{ir}(u_r) = & \text{avaldis 3-13} \\ &= \prod_{j=1}^r \left[ \exp\left(-\frac{1}{2}\left(\frac{x_j - x_{ij}^*}{\sigma_{ij}}\right)^2\right) \right] = \exp\left(-\frac{1}{2} \sum_{j=1}^r \left(\frac{x_j - x_{ij}^*}{\sigma_{ij}}\right)^2\right) \end{aligned}$$

Edasiste teisenduste teel saame, et:

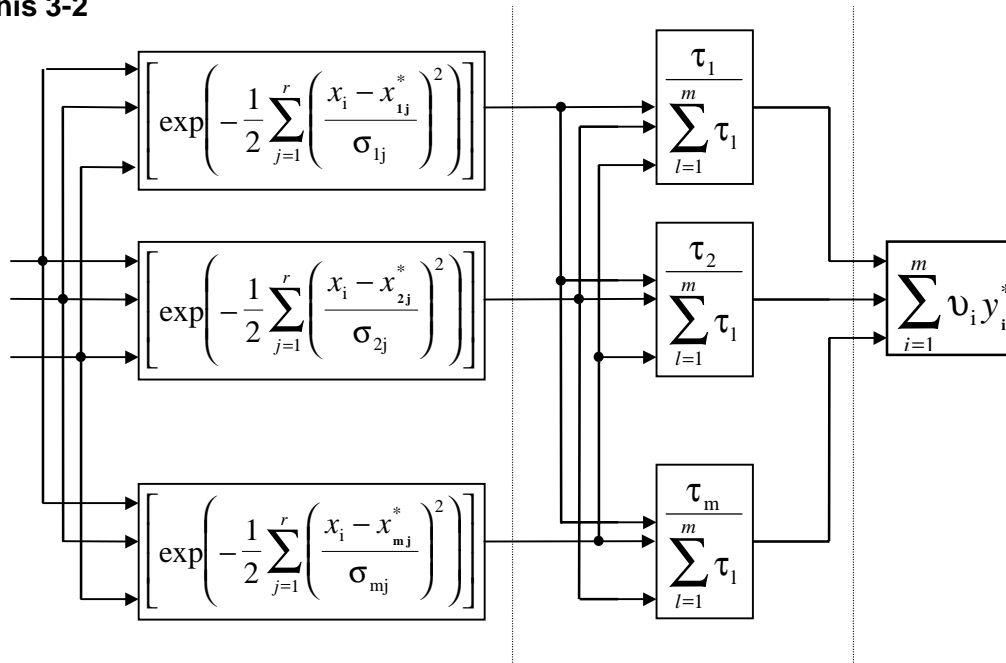
avaldis 3-14

$$y = \frac{\sum_{i=1}^m \tau_i y_i^*}{\sum_{i=1}^m \tau_i} = \frac{\sum_{i=1}^m y_i^* \left[ \exp \left( -\frac{1}{2} \sum_{j=1}^r \left( \frac{x_i - x_{ij}^*}{\sigma_{ij}} \right)^2 \right) \right]}{\sum_{i=1}^m \left[ \exp \left( -\frac{1}{2} \sum_{j=1}^r \left( \frac{x_i - x_{ij}^*}{\sigma_{ij}} \right)^2 \right) \right]} = \sum_{i=1}^m v_i y_i^*$$

kus  $v_i$ ,  $i=(1, m)$  on üksikute reeglite normeeritud tabamismäärad.

Seega võime lingvistilist mudelit esitada kolmekihilise võrguna nagu on näidatud joonisel 3-2:

joonis 3-2



Võime visuaalselt veenduda, et tegemist on närvivõrgu neuroniga.

Järelikult seisneb defineeritud reeglite arvu  $m$  korral lingvistilise mudeli konstrueerimine nende  $m$ -i “neuroni” parameetrite määramises. Antud olemasoleva sisendväljundinformatsiooni kogumi  $(u_{1k}, \dots, u_{rk}, y_k)$ ,  $k = (1, K)$  korral võime me selle ülesande formuleerida kui hägusa mudeli väljundi  $y$  ja vastava  $y_k$  erinevuse ruudu minimiseerimisena, sõltuvalt reeglite tingimus- ja järelduspoolte hägusate alamhulkade  $(B_{ij}, D_i)$  liikmesfunktsioonide parameetritest  $y_i^*$ ,  $x_{ij}^*$  ja  $\sigma_{ij}$ ,  $i = (1, m)$ ,  $j = (1, r)$ :

$$E_k = \frac{1}{2} (y - y_k)^2 = \frac{1}{2} e^2$$

avaldis 3-15

Alguses eeldame me, et tingimuse ja järelduspoolte hägusad alamhulgad on tundmatud ja et ainult osaline ja reeglite arvu  $m$  puudutav informatsioon on olemas. Määrame esialgu parameetrid  $y_i^*$ ,  $x_{ij}^*$  ja  $\sigma_{ij}$ ,  $i = (1, m)$ ,  $j = (1, r)$ , kasutades seda puudulikku informatsiooni, tähistades neid algseid hinnanguid vastavalt  $y_i^*(0)$ ,  $x_{ij}^*(0)$  ja  $\sigma_{ij}(0)$   $i = (1, m)$ ,  $j = (1, r)$ .

Rakendades ahelreeglit tuleme me reeglid tundmatute parameetrite  $y_i^*$ ,  $x_{ij}^*$  ja  $\sigma_{ij}$  täpsustamiseks.

$$y_i^*(k+1) = y_i^*(k) - \alpha \frac{\partial E_k}{\partial y_i^*} = y_i^*(k) - \alpha v_i e \quad \text{avaldis 3-16}$$

$$x_{ij}^*(k+1) = x_{ij}^*(k) - \alpha \frac{\partial E_k}{\partial x_{ij}^*} = x_{ij}^*(k) - \alpha v_i (y_i^* - y) e \frac{u_j - x_{ij}^*(k)}{\sigma_{ij}^2(k)} \quad \text{avaldis 3-17}$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \alpha \frac{\partial E_k}{\partial \sigma_{ij}(k)} = \sigma_{ij}(k) - \alpha v_i (y_i^* - y) e \frac{(u_j - x_{ij}^*(k))^2}{\sigma_{ij}^3(k)} \quad \text{avaldis 3-18}$$

kus  $\alpha$ -ga tähistatakse õppimismäära.

Õppimisalgoritm töötab kahes järgus: esmalt arvutatakse reeglite tabamismäärad  $\tau_i$ , nende normeeritud väärtused  $v_i$  ning hägusa mudeli väljund  $y$ , kasutades selleks määratavate parameetrite  $y_i^*$ ,  $x_{ij}^*$  ja  $\sigma_{ij}$  jooksvaid väärtusi  $y_i^*(k)$ ,  $x_{ij}^*(k)$  ja  $\sigma_{ij}(k)$ . Teises järgus täpsustatakse neidsamu parameetreid  $y_i^*(k)$ ,  $x_{ij}^*(k)$  ja  $\sigma_{ij}(k)$  vastavalt õppimisreeglitele:

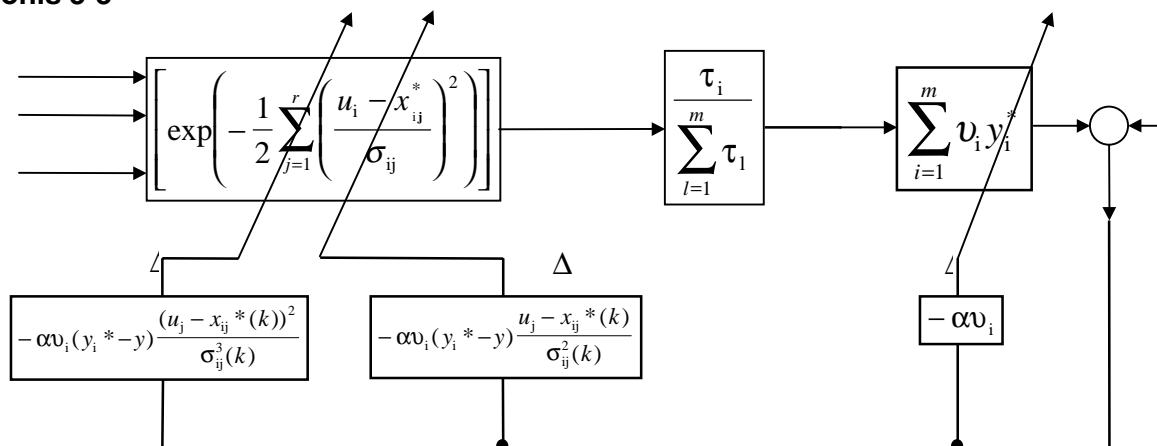
$$\Delta y_i^* = -\alpha v_i e \quad \text{avaldis 3-19}$$

$$\Delta x_{ij}^* = -\alpha v_i (y_i^* - y) e \frac{u_j - x_{ij}^*(k)}{\sigma_{ij}^2(k)} \quad \text{avaldis 3-20}$$

$$\Delta \sigma_{ij} = -\alpha v_i (y_i^* - y) e \frac{(u_j - x_{ij}^*(k))^2}{\sigma_{ij}^3(k)} \quad \text{avaldis 3-21}$$

Antud arvutuskeem on toodud joonisel 3-3:

joonis 3-3



kus  $\alpha$  kontrollib õppimisprotsessi stabiilsust. Parameetrite modifitseerimine kestab, kuni nende muutused on piisavalt väikesed.

Samamoodi tuletame algoritmi Sugeno tüüpi mudelite parameetrite leidmiseks:

avaldis 3-22

**KUI**  $U_1$  on  $B_{i1}$  **JA**  $U_2$  on  $B_{i2}$  **JA** ... **JA**  $U_r$  on  $B_{ir}$  **SIIS**  $y = b_{i0} + b_{i1}u_1 + b_{i2}u_2 + \dots + b_{ir}u_r$   
 $i = (1, m)$

Kasutades produkt-opeeraatorit ja Gaussi tüüpi liikmesfunktsioone sisendite defineerimisel, saame et mudeli väljund avaldub järgmiselt:

avaldis 3-23

$$y = \frac{\sum_{i=1}^m \tau_i (b_{i0} + b_{i1}u_1 + \dots + b_{ir}u_r)}{\sum_{i=1}^m \tau_i} = \frac{\sum_{i=1}^m \left[ \exp \left( -\frac{1}{2} \sum_{j=1}^r \left( \frac{x_i - x_{ij}^*}{\sigma_{ij}} \right)^2 \right) \right] (b_{i0} + b_{i1}u_1 + \dots + b_{ir}u_r)}{\sum_{i=1}^m \left[ \exp \left( -\frac{1}{2} \sum_{j=1}^r \left( \frac{x_i - x_{ij}^*}{\sigma_{ij}} \right)^2 \right) \right]} =$$

$$= \sum_{i=1}^m v_i y_i^*$$

Seades eesmärgiks vea ruudu minimiseerimise ja rakendades ahelreeglit, saame järgmised õppimisreeglid:

$$b_{i0}(k+1) = b_{i0}(k) - \alpha \frac{\partial E_k}{\partial b_{i0}} = b_{i0}(k) - \alpha v_i e \quad \text{avaldis 3-24}$$

$$b_{ij}(k+1) = b_{ij}(k) - \alpha \frac{\partial E_k}{\partial b_{ij}} = b_{ij}(k) - \alpha v_i e u_j, \quad j = (1, r) \quad \text{avaldis 3-25}$$

avaldis 3-26

$$x_{ij}^*(k+1) = x_{ij}^*(k) - \alpha \frac{\partial E_k}{\partial x_{ij}^*} = x_{ij}^*(k) - \alpha v_i (b_{i0} + b_{i1}u_1 + \dots + b_{ir}u_r - y) e \frac{u_j - x_{ij}^*(k)}{\sigma_{ij}^2(k)}$$

avaldis 3-27

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \alpha \frac{\partial E_k}{\partial \sigma_{ij}} = \sigma_{ij}(k) - \alpha v_i (b_{i0} + b_{i1}u_1 + \dots + b_{ir}u_r - y) e \frac{(u_j - x_{ij}^*(k))^2}{\sigma_{ij}^3(k)}$$

kus  $\alpha > 0$  tähistab õppimise kiirust (learning rate)

Sellise mudeli reeglite tingimuspooled defineerivad sisendpiirkonnad, mille puhul reegel rakendub, järelduspool defineerib tegevuse mis sooritatakse.

Sellega olemasolevate hägusate modelleerimismeetodite hulk ei piirdu, käesolevas töös aga me rohkem meetodeid ei käsitle ning edasises käsitletakse siintoodud meetodite kasutusvõimalusi.

## 4. Hägus juhtimine

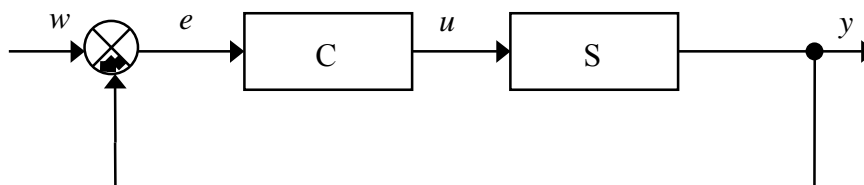
Protsessi või objekti mudelit kasutatakse enamasti protsessi või objekti analüüsimiseks ja/või juhtimisalgoritmi sünteesimiseks. Viimase lõppsaaduseks on reaalses olukorras juhtimisalgoritmi realiseeriv regulaator. Ka regulaatori konstrueerimise esimese etapina luuakse mudel, sel juhul siis regulaatori mudel. Eelmistes peatükkides näitasime hägusate mudelite eksistentsi võimalikkust, käesolevas tutvume hägusate regulaatoritega.

### 4.1 Tagasisidestatud juhtimise põhimõtted

Üldjuhul seostatakse juhtimisteooria teket matemaatilise juhtimisteooria kiire arenguga alates teisest maailmasõjast, ehkki kogemuste, intuitsiooni ja praktiliste oskuste tasemel olid tagasisidestatud juhtimise põhimõtted kasutusel juba sajandeid.

Joonisel 4-1 on toodud tagasisidestatud juhtimissüsteemi põhimõtteskeem

joonis 4-1



Juhitavat objekti (seade, mootor vm.) nimetatakse süsteemiks (S). Regulaatori C ülesandeks on garanteerida soovitud väljund  $y$ . Protsessi, mis hoiab süsteemi väljundit  $y$  lähedasena seadesuurusele  $w$ , vaatamata häiringutele või süsteemi parameetrite muutumisele aja jooksul, nimetatakse reguleerimiseks. Seaduspärasust, millele vastavalt regulaatori väljundsuurus ehk juhttoime  $u$  muutub, nimetatakse juhtimisalgoritmiks. Juhtimisseaduse, mis kirjeldab seost regulaatori sisendite ja väljundite vahel, üldkuju on järgmine:

$$u(k) = f(e(k), e(k-1), \dots, e(k-v), u(k-1), u(k-2), \dots, u(k-v)), \quad \text{avaldis 4-1}$$

kus  $e = w - y$  kujutab endast viga etteantud seadesuuruse ja objekti väljundi vahel,  $v$  on regulaatori järk ning  $f$  on üldjuhul mittelineaarne funktsioon. Sellest juhtimisseadusest on tuletatud erinevad juhtimisalgoritmid: proportsionaalne (P), diferentseeriv (D), integreeriv (I), proportsionaalne + integreeriv (PI), proportsionaalne + integreeriv + diferentseeriv (PID) jne.

Käsitletud juhtimisseadust võib nimetada ka tavajuhtimisseaduseks.

Tavajuhtimise puhul eeldatakse loomulikult, et juhtimisalgoritmi kirjeldatakse analüütiliselt - algebraliste, diferentsiaal- või differentsvõrranditega ning sellise juhtimisalgoritmi süntees eeldab juba omakorda juhitava süsteemi matemaatilist mudelit. Järgnevas näitame, et kasutades hägusloogika vahendeid taoline nõue ei kehti.

### 4.2 Hägusloogikaregulaatori olemus

Hägusloogikaregulaatori põhiosaks on selliste reeglibaas, mis moodustub hulgast regulaatori sisendeid  $e$  ja väljundeid  $u$  seostavaist loogilistest lausetest. Põhimõtteliselt võib hägusloogikaregulaatorit esitada tavajuhtimisseadusele sarnases vormis (avaldis 4-1), kuid see ei tähenda, et hägusloogikaregulaator kujutaks endast mingit ülekandefunktsiooni või differentsvõrrandit. Hägusloogikaregulaatori kogemusliku olemuse tõttu on vea  $e$  ja juhttoime

$u$  minevikuväärtuste kasutamine piiratud. See-eest on hägusloogikaregulaatorites kasutusel aga  $e$  ja  $u$  mõtet omavad erinevad kombinatsioonid: vea muutus

$$\Delta e(k) = e(k) - e(k-1), \quad \text{avaldis 4-2}$$

vigade summa, juhttoime muutus

$$\Delta u(k) = u(k) - u(k-1) \quad \text{avaldis 4-3}$$

jne.

Tüüpiline hägusloogikaregulaator kirjeldab ühelt poolt juhttoime muutust  $\Delta u(k) = u(k) - u(k-1)$  ja teiselt poolt viga  $e(k)$  ja vea muutust  $\Delta e(k) = e(k) - e(k-1)$ . Sellise juhtimisseaduse üldkuju on järgmine:

$$\Delta u(k) = F(e(k), \Delta e(k)) \quad \text{avaldis 4-4}$$

Regulaatori tegelik väljund  $u(k)$  arvutatakse avaldisest 4-3:

$$u(k) = u(k-1) + \Delta u(k) \quad \text{avaldis 4-5}$$

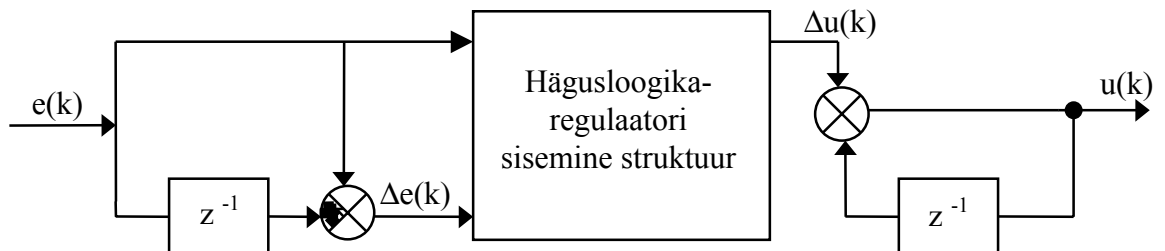
Toome ära juhtimisseadust realiseeriva lihtsa hägusloogikaregulaatori lausete komplekti ehk reeglibaasi prototüübi, kus piirides  $[-1, 1]$  muutuvad muutujad *viga e*, *vea muutus  $\Delta e$*  ja *juhttoime muutus  $\Delta u$*  on lahterdatud kolmeks hägusaks hulgaks “negatiivne”, “positiivne” ja “ligikaudu null”.

tabel 4-1

Reegel 1:	<b>KUI</b>	viga $e(k)$ on positiivne	<b>JA</b>	vea muutus $\Delta e(k)$ on ligikaudu null	<b>SIIS</b>	juhttoime muutus $\Delta u(k)$ peab olema positiivne
Reegel 2:	<b>KUI</b>	viga $e(k)$ on negatiivne	<b>JA</b>	vea muutus $\Delta e(k)$ on ligikaudu null	<b>SIIS</b>	juhttoime muutus $\Delta u(k)$ peab olema negatiivne
Reegel 3:	<b>KUI</b>	viga $e(k)$ on ligikaudu null	<b>JA</b>	vea muutus $\Delta e(k)$ on ligikaudu null	<b>SIIS</b>	juhttoime muutus $\Delta u(k)$ peab olema ligikaudu null
Reegel 4:	<b>KUI</b>	viga $e(k)$ on ligikaudu null	<b>JA</b>	vea muutus $\Delta e(k)$ on positiivne	<b>SIIS</b>	juhttoime muutus $\Delta u(k)$ peab olema positiivne
Reegel 5:	<b>KUI</b>	viga $e(k)$ on ligikaudu null	<b>JA</b>	vea muutus $\Delta e(k)$ on negatiivne	<b>SIIS</b>	juhttoime muutus $\Delta u(k)$ peab olema negatiivne

Me võime vaadelda regulaatorit süsteemina, mille sisendeiks on tingimusteosas kirjeldatud muutujad ja mille väljundiks on järelusmuutuja. Need muutujad on seotud regulaatori sisemise struktuuriga, ega pole üldjuhul samastatavad regulaatori reaalse sisendi  $e$  ja väljundiga  $u$ .

joonis 4-2



Täpsemalt väljendudes on hägusloogikaregulaatori sisendid ja väljundid juhitava süsteemi olekud; järelikult on see regulaator teatavate reeglite ja hägusa järelusmehhanismiga mõjutatav olekumuutujaregulaator.

Tabel 4-1 tõi ära üldlevinud juhtimisstrateegia, mida saab kasutada laia klassi objektide juhtimisel, mille dünaamika on lähendatud esimest järku süsteemile hilistumisajaga  $T_d$  ja ajakonstandiga  $T$ :

$$W(s) = \frac{e^{-T_d s}}{1 + Ts} \quad \text{avaldis 4-6}$$

Hägus juhtimisalgoritm peegeldab juhtimismehhanismi, mida kasutab inimene, kel puuduvad igasugused teadmised juhitavast objektist selles osas, mis puudutab objekti matemaatilist mudelit ja tema analüütilist kirjeldust. Hoolimata sellest saab näidata, et ülaltoodud algoritm suudab lahendada ka keerukamaid probleeme. Loogiline on, et keerukamate ülesannete lahendamiseks vajame me rohkem teadmisi, ehk mahukamat reeglibaasi, millega täpsemalt kirjeldada juhtimisstrateegiat.

## 4.3 Analoogiad tavareguleerimisseadustega

### 4.3.1 Hägus PI-regulaator

Käsitletud hägusloogikaregulaatori sisemine mehhanism tegi juhttoime muutumise  $\Delta u(k) = u(k) - u(k-1)$  sõltuvaks veast  $e(k)$  ja vea muutusest  $\Delta e(k) = e(k) - e(k-1)$ :

$$\Delta u(k) = f(e(k), \Delta e(k)) \quad \text{avaldis 4-7}$$

Viimane avaldis sarnaneb märgatavalt hästituntud diskreetse PI-regulaatori algoritmile:

$$\Delta u(k) = K_i e(k) + K_p \Delta e(k), \quad \text{avaldis 4-8}$$

kus  $K_i$  ja  $K_p$  on regulaatori parameetrid. Mõlemad juhtimisseadused seovad samu sisend- ja väljundmuutujaid, seega võib taolist, Mamdani-tüüpi hägusloogikaregulaatorit pidada hägusaks PI-regulaatoriks. Erinevus konventsionaalse PI-regulaatoriga ilmneb üksnes seose tüübis. Tavapärase PI-regulaatori puhul on seos lineaarne, hägusloogikaregulaatoris üldjuhul mittelineaarne.

### 4.3.2 Hägus PD-regulaator

Kui hägusloogikaregulaatoriväljundiks on juhttoime muutuse asemel juhttoime enda väärtus  $u(k)$ , on regulaatori funktsioon, mille ta realiseerib vastavalt

$$u(k) = f(e(k), \Delta e(k)) \quad \text{avaldis 4-9}$$

Sellise regulaatori reeglibaas moodustub järgmist tüüpi lausetest:

**KUI** viga  $e(k)$  on ... **JA** vea muutus  $\Delta e(k)$  on ... **SIIS** juhttoime  $u(k)$  peab olema ...

Regulaatori poolt realiseeritav funktsioon on kõrvutatav PD-regulaatori algoritmiga:

$$u(k) = K_p e(k) + K_D \Delta e(k), \quad \text{avaldis 4-10}$$

kus  $K_p$  ja  $K_D$  on PD-regulaatori parameetrid. Seetõttu kutsutakse eeltoodud juhtimisalgoritmiga hägusloogikaregulaatorit hägusaks PD-regulaatoriks.

Hägusate PI- ja PD-regulaatorite mehhanismid on sarnased, erinevus seisneb ainult väljund-muutujas. Seega saab lihtsa modifitseerimisega teha ühest teise.

### 4.3.3 Hägus PID regulaator

PD-tüüpi hägusloogikaregulaatorit saab laiendada kui lisisendmuutujana kasutada vigade summat:

$$\sum e(k) = \sum_{l=0}^k e(l-1) \quad \text{avaldis 4-11}$$

Regulaatori poolt realiseeritav funktsioon näeb siis välja järgmine:

$$u(k) = F(e(k), \Sigma e(k), \Delta e(k)) \quad \text{avaldis 4-12}$$

Taolise hägusloogikaregulaatori reeglibaas koosneb järgmist tüüpi reeglitest:

**KUI** viga  $e(k)$  on ... **JA** viga muutus  $\Delta e(k)$  on ... **JA** vigade summa  $\Sigma e(k)$  on ... **SIIS** juhttoime  $u(k)$  peab olema ...

Järeldusmehhanism kasutab täpselt samu muutujaid kui laialt levinud PID-regulaator:

$$u(k) = K_P e(k) + K_I \Sigma e(k) + K_D \Delta e(k), \quad \text{avaldis 4-13}$$

kus  $K_P$ ,  $K_I$  ja  $K_D$  on PID-regulaatori parameetrid. Ühesõnaga, tegu on PID-regulaatori analoogiga hägusloogikaregulaatorite maailmas.

Alternatiivne hägus PID-regulaator põhineb inkrementaalsel PID-juhtimisalgoritmil:

$$\Delta u(k) = K_0 e(k) + K_1 e(k-1) + K_2 e(k-2), \quad \text{avaldis 4-14}$$

realiseerib järgmist funktsiooni:

$$\Delta u(k) = f((e(k), e(k-1), e(k-2))) \quad \text{avaldis 4-15}$$

ja tema reeglibaas koosneb järgmist tüüpi reeglitest:

**KUI** viga  $e(k)$  on ... **JA** viga  $e(k-1)$  on ... **JA** viga  $e(k-2)$  on ... **SIIS** juhttoime muutus  $\Delta u(k)$  peab olema ...

Teoreetiliselt on võimalik ka hägusloogikaregulaatori sisendmuutujate hulga edasine laiendamine, mis muudab aga regulaatori sisemise struktuuri keerukaks. Veelgi enam, sellisel juhul on reegleid koostaval eksperdil raske formuleerida selget ja mõistlikku juhtimisstrateegiat. Toimivate reeglite koostamine hõlmates sinna vega kaugemaid minevikuväärtusi osutub pea ülejõukäivaks ja see on ka põhjus, miks rõhuv enamus hägusloogikaregulaatoreid on PI või PD tüüpi.

## 4.4 Hägusloogikaregulaatori konstrueerimine

### 4.4.1 Hägusloogikaregulaatori normeerimistegurid

Hägusloogikaregulaatorite konstrueerimise juures kerkib sarnaselt mudelitele esile kaks probleemi: esimeseks võib pidada lingvistilistele märgistele vastavate liikmesfunktsioonide määramist, teine probleem on seotud hägusloogikaregulaatori tegevust determineeriva reeglistiku väljatöötamisega. Hägusloogikaregulaatori konstrueerimise käigus tuleb need mõlemad probleemid lahendada.

PI-regulaatori näitel:

Vea  $e(k) = w(k) - y(k)$  määramispiirkond on määratud muutujate  $w$  ja  $y$  maksimaalsete ja minimaalsete väärtustega, see on intervall  $[e_{\min}, e_{\max}]$ , kus

$$e_{\max} = w_{\max} - y_{\min} \quad e_{\min} = w_{\min} - y_{\max} \quad \text{avaldis 4-16}$$

Analoogiliselt on vea muutusel  $\Delta e(k) = e(k) - e(k-1)$  ja juhtimiseaduse muutusel  $\Delta u(k) = u(k) - u(k-1)$  oma määramispiirkonnad  $[\Delta e_{\min}, \Delta e_{\max}]$  ja  $[\Delta u_{\min}, \Delta u_{\max}]$ , kusjuures:

$$\begin{aligned} \Delta e_{\max} &= e_{\max} - e_{\min} & \Delta e_{\min} &= e_{\min} - e_{\max} = -\Delta e_{\max} & \text{avaldis 4-17} \\ \Delta u_{\max} &= u_{\max} - u_{\min} & \Delta u_{\min} &= u_{\min} - u_{\max} = -\Delta u_{\max} & \text{avaldis 4-18} \end{aligned}$$

Ka siin on mõistlik sisendi ja väljundi muutumispiirkonnad normeerida.

Olgu  $e(k)$ ,  $\Delta e(k)$ , ja  $\Delta u(k)$  määramispiirkonnad vastavalt  $[-a_e, a_e]$ ,  $[-a_d, a_d]$  ja  $[-a_{du}, a_{du}]$ . Olgu vastavad normeeritud universaalhulgad tähistatud järgmiselt:

$$X_1 = [-a_e^*, a_e^*], X_2 = [-a_d^*, a_d^*], Y = [-a_{du}^*, a_{du}^*]$$

Muutujate  $e(k)$ ,  $\Delta e(k)$  määramispiirkondade transformeerimine normeeritud universaalhulkadesse  $X_1$  ja  $X_2$  on samane nende korrutamisega normeerimisteguritega:

$$k_e = \frac{a_e^*}{a_e}; k_d = \frac{a_d^*}{a_d} \quad \text{avaldis 4-19}$$

Normeerimistegurid  $k_e$  ja  $k_d$  teisendavad tegelikud  $e(k)$  ja  $\Delta e(k)$  väärtused normeeritud väärtusteks  $e^*(k) \in X_1$  ja  $\Delta e^*(k) \in X_2$ . Eeldades lineaarset seost tegelike ja normeeritud väärtuste vahel saame:

$$e^*(k) = k_e e(k); \quad \Delta e^*(k) = k_d \Delta e(k) \quad \text{avaldis 4-20}$$

Rakendades taolisi normeeritud sisendeid regulaatorile saame viimase reageeringuna täppistatud väärtuse  $\Delta u^*(k)$ , mis kuulub normeeritud universaalhulka  $Y = [-a_{du}^*, a_{du}^*]$ . Tegelik juhttoime  $\Delta u(k)$  saamiseks tuleb  $\Delta u^*(k)$  denormeerida, kasutades järgmist seost:

$$\Delta u(k) = k_{du} \Delta u^*(k), \quad \text{avaldis 4-21}$$

kus

$$k_{du} = \frac{a_{du}}{a_{du}^*} \quad \text{avaldis 4-22}$$

$\Delta u(k)$  kuulub juba normeerimata universaalhulka  $[-a_{du}, a_{du}]$ .

Analoogiliselt PI-regulaatoriga on sissetoodavad ja arvutatavad normeerimistegurid  $k_e$ ,  $k_d$  ja  $k_{du}$  ka PD-regulaatori jaoks:

$$e^*(k) = k_e e(k); \quad \Delta e^*(k) = k_d \Delta e(k); \quad u(k) = k_{du} u^*(k). \quad \text{avaldis 4-23}$$

## 4.4.2 Reeglibaasi koostamine

Hägasloogikaregulaatori reeglibaasi konstrueerimisel on mõistlik aluseks võtta teatud standartne näidislausete komplekt, mida võib käsitleda kui vahendit, mis ühendab terve insenerimõistuse ja hägusjuhtimise alased kogemused. Näiteks võib tuua MacVicar-Whelani reeglibaasi, milles termihulgad viga  $e$ , vea muutus  $\Delta e$  ja juhttoimemuutus  $\Delta u$  on kõik lahterdatud kaheksaks hägusaks hulgaks, mida tähistavad lingvistilised märgendid “+L” - “väga positiivne”, “+M” - “keskmiselt positiivne”, “+S” - “vähe positiivne”, “+Z” - “positiivne null”, “-Z” - “negatiivne null”, “-S” - “vähe negatiivne”, “-M” - “keskmiselt negatiivne”, “-L” - “väga negatiivne”.

Viga  $e$  on defineeritud kui seadesuuruse  $u$  ja süsteemi väljundi  $y$  vahe:

$$e(k) = w(k) - y(k)$$

**avaldis 4-24**

Reeglibaasi aluseks on kolm metareeglit:

**MR1:** Kui viga  $e(k)$  ja vea muutus  $\Delta e(k)$  on nullid, siis tuleb säilitada senine regulaatori väljund.

**MR2:** Kui viga  $e(k)$  läheneb nullile rahuldava kiirusega, siis tuleb säilitada senine regulaatori väljund.

**MR3:** Kui viga  $e(k)$  ei lähe ise nulliks, siis juhttoimemuutus peab olema mittevõrdne nulliga ja sõltub  $e(k)$  ja  $\Delta e(k)$  märkidest ja suuruselt.

Reeglibaasi esitab tabel 4-2:

**tabel 4-2**

		tsoon 4			tsoon 5				
		←			Δe(k)			→	
		-L	-M	-S	-Z	+Z	+S	+M	+L
↑ e(k) ↓	-L	-L	-L	-L	-L	-L	-M	-S	-Z
	-M	-L	-L	-M	-M	-M	-S	-Z	+S
	-S	-L	-M	-S	-S	-S	-Z	+S	+M
	-Z	-M	-M	-S	-Z	+Z	+S	+M	+M
	+Z	-M	-M	-S	-Z	+Z	+S	+M	+M
	+S	-M	-S	+Z	+S	+S	+S	+M	+L
	+M	-S	+Z	+S	+M	+M	+M	+L	+L
	+L	+Z	+S	+M	+L	+L	+L	+L	+L

tsoon 1  
tsoon 3  
tsoon 2

Tabelis võib eristada 5 homogeenset tsooni - 1. ja 2. tsooni reeglid vastavad olukordadele, kus  $e$  ja  $\Delta e$  on vastasmärgilised, reeglid tsoonides 3 ja 4 situatsioonidele, kus  $e$  ja  $\Delta e$  on sama märgiga, 5. tsooni reeglitele on iseloomulik e ja  $\Delta e$  väike suurus.

Täpsustame tingimusi - oletame et nii  $e$  kui  $\Delta e$  on defineeritud ühe ja sama normeeritud  $([-1, 1])$  universaalhulga alamhulkadena. Oletame samuti, et ajavahemik  $e$  ja  $\Delta e$  kahe järjestikuse mõõtmise vahel on piisavalt väike nii et seadesuurus  $w$  selle aja jooksul muutuda ei jõua - s.t.  $w(k) = w(k-1)$  ja järelikult:

$$\Delta e(k) = e(k) - e(k-1) = w(k) - y(k) - (w(k-1) - y(k-1)) = y(k-1) - y(k)$$

**avaldis 4-25**

•Tsoonides 1 ja 2 paiknevate reeglite kohta kehtib seega üks järgmistest võrratustest:

$$e(k) < 0 \text{ ja } \Delta e(k) > 0 \Rightarrow y(k) < y(k-1)$$

$$e(k) > 0 \text{ ja } \Delta e(k) < 0 \Rightarrow y(k) > y(k-1)$$

Järelikut on viga isekorrigeeruv ja  $\Delta u$  praktiliselt null - regulaatori väljund ei muutu. Reeglid selles tsoonis on kooskõlas metareegluga MR2

•Tsoonides 3 ja 4:

$$e(k) \geq 0 \text{ ja } \Delta e(k) \geq 0 \Rightarrow w(k) \geq y(k) \text{ ja } y(k) \leq y(k-1)$$

$$e(k) \leq 0 \text{ ja } \Delta e(k) \leq 0 \Rightarrow w(k) \leq y(k) \text{ ja } y(k) \geq y(k-1)$$

Viga pole isekorrigeeruv ning  $\Delta u(k)$  on kas positiivne või negatiivne.  $\Delta u(k)$  suurus on seotud  $e(k)$  ja  $\Delta e(k)$  suurusetega. Reeglid selles tsoonis vastavad metareeglile MR3.

•5. tsoonis on nii  $e(k)$  kui  $\Delta e(k)$  lähedased nullile. Süsteem on stabiilses olekus ning  $\Delta u(k)$  on peaaegu null - kehtivaks jääb senine regulaatori väljund. Selle tsoonis reeglid on kooskõlas metareegluga MR1.

Peaaegu kõigi tuntud hägusloogikaregulaatorite reeglibaasid on realiseeritavad McVicar-Whelani reeglibaasi alusel, modifitseerides regulaatori sisend-väljundmuutujate  $e$ ,  $\Delta e$ ,  $\Delta u$  lingvistilisi märgendeid

Paljukasutatud McVicar-Whelani reeglibaasi modifikatsioon on selline, kus lingvistilised märgendid "positiivne null" ja "negatiivne null" on ühendatud üheks lingvistiliseks märgendiks "null" (Z) ja muutujate märgendid "keskmine" on asendatud märgendiga "palju" juhtudel, kus viga  $e(k)$  on null ja vea muutus on suur.

Samuti ei osutu paljudes rakendustes kõik tabel 4-2 toodud reeglid vajalikuks. Kui juhitavaks objektiks on 1. järku hilistumisega lüli, piisab ka tabel 4-3 toodud reeglitest.

**tabel 4-3**

				←	$\Delta e(k)$	→				
				-L	-M	-S	Z	+S	+M	+L
							-L			
							-M			
							-S			
	↑						Z	+S	+M	+L
	$e(k)$						Z	+S	+M	+L
	↓						+S			
							+M			
							+L			

McVicar-Whelani reeglibaasi saab kasutada allikana keerukamate probleemorienteeritud reeglibaaside koostamisel modifitseerides või välja jättes olemasolevaid ja lisades uusi reegleid.

## 4.5 Hägusloogikaregulaatorite häälestamine

Teeme nüüd kokkuvõtte hägusloogikaregulaatori konstrueerimise sammudest.

1. **Hägusloogikaregulaatori sisend- ja väljundmuutujate defineerimine.** Sisend- ja väljundmuutujate vastava valikuga määratakse kindlaks regulaatori tüüp (PI, PD või PID-regulaator). Millist juhtimisseadust soovitakse konkreetsel juhul kasutada, sõltub juba sellest, mida reguleerimisega saavutada tahetakse.

2. **Hägusloogikaregulaatori parameetrite määramine.** Sisend-väljundmuutujate lingvistiliste märgendite ja nende vastavate liikmesfunktsioonide määramine ning normeeritud universaalhulkade normeerimistegurite arvutamine.

3. **Hägasloogikaregulaatori reeglibaasi koostamine.** Kui puuduvad aprioorsed reeglid, on mõttekas võtta reeglibaasi koostamisel aluseks McVicar-Whelani reeglibaas.

Kui liikmesfunktsioonid ja reeglibaas on tuletatud, on viimaseks sammuks hägasloogikaregulaatori konstrueerimisel viimase häälestamine. Kuna hägasloogikaregulaator on äärmiselt paindlik süsteem, mille käitumine sõltub paljudest parameetritest, on tema häälestamine palju keerukam kui tavaliste regulaatorite häälestamine. Enamasti on häälestusparameetritena kasutusel normeerimistegurid, mille muutmisel muutuvad sisend- $v$ äljundmuutujate liikmesfunktsioonid ja seetõttu ka regulaatori reaktsioon mingitele sisend- $v$ äljundtasetele.

Välja on selgitatud järgmised seaduspärasused:

*Suured  $k_e$  väärtused tõstavad süsteemi reageerivust (väike tõusuaeg ja staatiline viga) kuid vähendavad süsteemi stabiilsust (suur ülereguleerimine).*

*Väike reguleerimisaeg on piiratud suurte  $k_d$ ,  $k_e$  ja suhteliselt väikeste  $k_{du}$  väärtustega.*

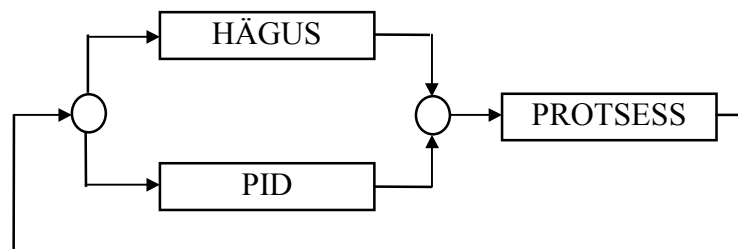
*$k_{du}$  väikesed väärtused suurendavad tõusuaega ja ruutkeskmist viga.*

Taolise häälestamise efektiivsus on piiratud, kuna ühe või teise suuruse parendamiseks tehtud normeerimisteguri muutus halvendab mingit teist suurust. Raskusi tekib enamasti üheaegselt hea reageerivuse ja stabiilsuse nõudlemisel. Kuid eks samad probleemid ole ka tavaliste regulaatorite kasutamisel.

## 4.6 Arendatud juhtimismeetodid

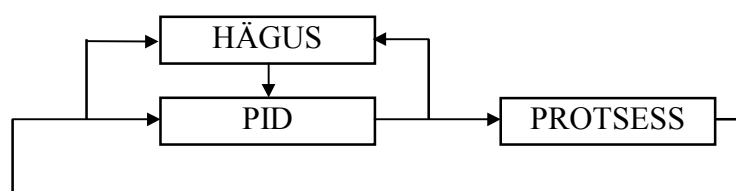
Hägasloogikaregulaatorite edukad rakendused on õigustuseks nende olemasolule. Lisaks tavalisele lülitusskeemile, mida kujutab joonis 4-1 on võimalikud ka teised variandid. Eriti tõhusaks on osutunud hägusa ja tavajuhtimise ühendid. Näiteks on võimalik variant, kus hägas regulaator täiendab PID-regulaatori reguleerimist, astudes tegevusse juhtudel, mil tavajuhtimise vahenditega pole protsessi stabiliseerimine võimalik või on see liiga aeglane (joonis 4-3)

joonis 4-3



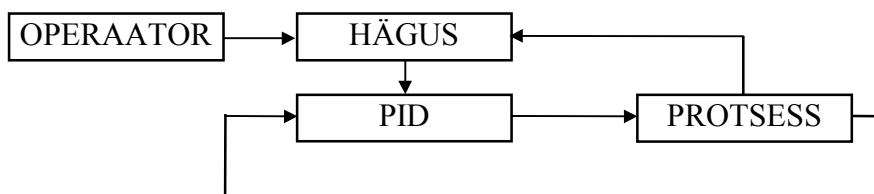
Või on tegu variandiga, kus hägas moodul tegeleb PID-regulaatori ümberhäälestamisega vastavalt olukorrale. Seda võib nimetada juba adaptiivseks reguleerimiseks (joonis 4-4).

joonis 4-4



PID-regulaatori häälestamine võib toimuda ka nii, et seda häälestavat hägusat regulaatorit häälestab omakorda inimõistus. Viimasel juhul on juhtimishierahia juba mitmetasandiline (joonis 4-5).

joonis 4-5



Eraldi võib juttu teha Sugeno-tüüpi PI/PD/PID regulaatorist, mis kuivõrd tema väljundiks on sisendite (antud juhul vea ja vea muutuste) lineaarsed kombinatsioonid ei kujuta endast midagi muud kui regulaatorite ümberlülitit. Iga reegli väljund sellise regulaatori reeglibaasis on tavapärane erinevate parameetritega regulaator ning hägusa regulaatori väljundiks on tingimuste rahuldatusega määratud kaaludega kaalutud selliste regulaatorite väljundite keskmine. Reaalselt näeb see välja niiviisi, et sõltuvalt tingimustest omandab suurema kaalu üks või teine regulaator. Tänu hägususele on selline ümberlülitamine sujuv, mis teeb reguleerimise täiuslikumaks.

## **5. Fermentatsiooniprotsesside modelleerimine ja juhtimine**

### **5.1 Fermentatsiooniprotsesside olemus**

Fermentatsioon kujutab endast inimesele vajalike produktide saamist bioloogiliste agentide - mikroorganismide, viiruste, taimsete või loomsete rakkude - abiga.

Fermentatsiooniprotsessi komponentideks on: bioloogiline agent, substraat, lõpp-produkt, aparatuur, režiim, protsessi juhtimismeetodid.

Fermentatsiooniprotsessi eesmärgiks on biokeemiliste ja füsioloogiliste protsesside mõistmise alusel ja konkreetse agendi geneetiliselt kindlaks määratud omaduste piirides saavutada maksimaalne lõpp-produkti ja biomassi (kui see on seatud eesmärgiks) kogus.

Erinevusena keemilistest protsessidest võib tuua bioloogiliste agentide tundlikkuse, mis teeb fermentatsiooniprotsessist raskustega läbiviidava ja juhitava protsessi.

Kultiveerimisvariandid

- süvitsi ja pinnakihil (vedel või tahke keskkond)
- perioodiline, poolperioodiline ja katkematu
- steriilne ja tingimisi steriilne
- aeroobne ja anaeroobne

Fermentatsioon viiakse läbi kolmes osas:

1. fermentatsioonieelsed protsessid - toitekeskkonna valmistamine, doseerimine, sterilisatsioon, jahutus
2. fermentatsioon (biokatalüüs) - substraadist produkti saamine
3. eraldamine, puhastus, kontsentreerimine ja lõppproduktile kaubandusliku välimuse andmine

Tehnoloogilised erinevused algavad kolmandast osast

#### **1. fermentatsioonieelsed protsessid**

- toitekeskkonna komponentide transport ja doseerimine
- vedelate keskkondade valmistamine
- toitekeskkonna steriliseerimine
- vedelate steriilsete toitekeskkondade ja teiste vedelate komponentide doseerimine
- õhu steriliseerimine

#### **2. fermentatsiooniprotsess**

Fermentatsioonis osalevate komponentide viimine fermentaatorisse ja eemaldamine fermentaatorist, vajalike suuruste mõõtmine, optimaalse substraadi/produkti kontsentratsiooni kindlustamine.

#### **3. produkti saamine**

- fermentatsiooni keskkonna fraktsioneerimine
- biomassi desintegratsioon
- mikroobsete kontsentraatide saamine

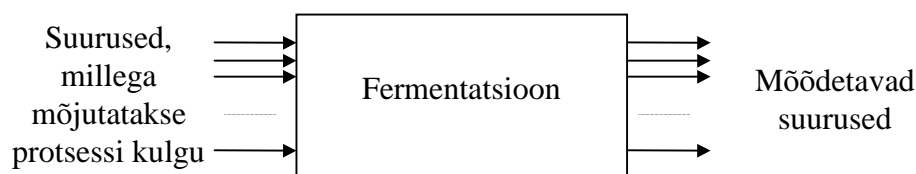
- biomassi saamine
- erineva puhastusastmega lõppproduktide saamine

Fermentatsiooniprotsessi läbiviimine taandub järgmisele:

- tehnoloogilisele režiimile vastav algkomponentide doseerimine
- toitainete andmine fermentaatorisse
- fermentaatorist keskkonna, soojuse, kasutatud õhu eemaldamine ja bioloogilise agendi jaoks vajalike keskkonna füüsiliskeemiliste parameetrite mõõtmine.

Fermentatsiooniprotsessi kõige üldisemal kujul esitab joonis 5-1:

joonis 5-1



Biotehnoloogias mõõdetavad füüsikalised suurused võib jagada kolme gruppi:

1. Mõõtmised, mis on analoogilised keemiatehnoloogias, toiduainetetööstuses ja mujal majandusharudes:

- fermentatsiooni temperatuur
- pH
- rõhk
- keskkondade, substraatide ja produktide nivoo ja mass
- vedelike ja gaaside kulu
- segisti pöörlemiskiirus
- elektrimootori(te) poolt tarbitav võimsus
- hapniku ja süsinikhappegaasi ja teiste gaaside kontsentratsioon
- kuivainete kontsentratsioon
- kaioonide ja anioonide kontsentratsioon vedelikes.

2. Ainult biotehnoloogiale omased suurused.

- Peamiste substraatide  $s$  ja produktide  $p$  kontsentratsioon fermentatsioonikeskkonnas
- tähtsamate rakusiseste ainete kontsentratsioon
- biomassi kontsentratsioon  $x$
- kõrvalise mikrofloora olemasolu
- fermentatsioonikeskkonnas lahustunud hapniku ja süsinikdioksiidi kontsentratsioon
- vahu olek ja nivoo
- lõpp-produkti kontsentratsioon tehnoloogilise protsessi erinevais staadiumeis
- lõpp-produkti kvaliteedinäitajad
- ebaõnnestunud fermentatsiooni identifitseerimiseks olulised näitajad

3. Ainult biotehnoloogiale omased eelmise kahe grupi näitajate põhjal arvutatavad suurused.

- süsteemi tootlikus  $Q_p$
- kasvu piirkiirus

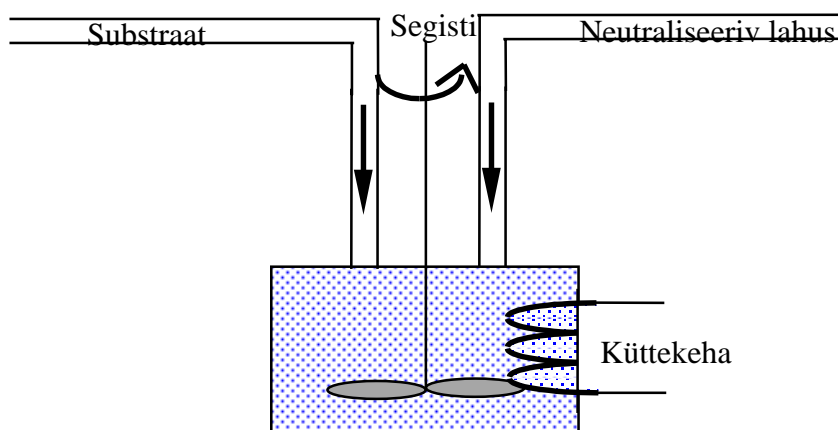
- substraadi tarbimise piirkiirus
- produkti moodustumise piirkiirus
- majanduslik koefitsent  $Y$

Fermentatsiooniprotsesside juhtimise seisukohalt on tähtsamateks näitajateks

- temperatuur
- segisti pöörlemiskiirus
- fermentatsiooni keskkonna mass ja maht
- pH
- lahustunud hapniku kontsentratsioon
- lahustunud süsinikdioksiidi kogus
- substraatide ja produktide kontsentratsioon

Käesolevas peatükis ei süüvita fermentatsiooniprotsesside tehnoloogilistesse üksikasjadesse, selle asemel keskendutakse protsessi juhtimis- ja modelleerimisvõimaluste uurimisele.

### joonis 5-2



Fermentatsiooniprotsessi juhtimine võib kulgeda näiteks nii nagu kujutab joonis 5-2, kus fermentatsiooni temperatuuri reguleeritakse küttekehale antava võimsusega, keskkonna pH-d stabiliseeritakse neutraliseeriva lahuse lisamisega ning biomassi ja produkti juurdekasv on mõjutatav eelkõige substraadi lisamisega. Segisti funktsiooniks on keskkonna hapnikuga paremini varustamine (mõeldav on ka teatav õhupump). Need reguleerimiseadised toimivad vastavalt juhtimiseadustele, mis üldjuhul on sõltuvad protsessi mõõdetavatest suurustest.

Et suur osa protsessi suurustest pole pidevalt või on seotud nende mõõtmine suurte raskustega (kehtib eriti fermentatsioonikeskkonna komponentide kontsentratsioonide kohta), teeb see protsessi eduka läbiviimise keeruliseks.

## **5.2 Fermentatsiooniprotsessi matemaatiline mudelleerimine**

Fermentatsiooniprotsesside optimeerimisel ja juhtimisel kasutatavad mudelid jagunevad nelja rühma:

- füsioloogilised
- struktuursed
- mittestruktuursed
- formaalsed

Füsioloogilised mudelid kujutavad protsessi bioloogilise kasvu füsioloogiat loogiliste tingimuste jadana, mis ei pea olema tingimata esitatud matemaatilisel kujul. Sellist tüüpi mudelid on leidnud laia kasutamise fermentatsiooniprotsesside automaatjuhtimissüsteemide konstrueerimisel.

Struktuursed mudelid kujutavad endast tavalist diferentsiaalvõrrandite või osatuletistega diferentsiaalvõrrandite süsteemi. Mudeleis võetakse arvesse rakkude jaotus vanuse, mahu, massi, füsioloogilise seisukorra jmt. järgi. Olulisimaks osutuvad sellist tüüpi mudelid, mis kirjeldavad rakkude jaotust nende vanuse järgi, kuna kasv, lahustuvus, teisejärguliste produktide biosüntees ja mikroorganismide hapnikutarbimine sõltub nende vanusest. Suure tömahukuse ja keerukuse tõttu kasutatakse sellist tüüpi mudeleid aga inseneriülesannete lahendamisel harva.

Mittestruktuursed matemaatilised mudelid rajanevad oletusel, et biosünteesi protsessis osaleb üksainus homogeenelt kasvav organism. Peale selle eeldatakse sageli, et substraate on samuti üksainus. Mittestruktuursed mudelid kujutavad üldjuhul endast tavaliste diferentsiaal- või algebraliste võrrandite süsteemi ja neid kasutatakse enim optimeerimis- ja insenerstatistiliste ülesannete lahendamisel.

Formaalsed mudelid ehitatakse musta kasti põhimõttel. Kuna mudelid ei sisalda füüsiliste hüpoteeside struktuuri, aproksimeerivad sellist tüüpi mudelid formaalselt eksperimendiandmeid. Selliseid mudeleid kasutatakse tavaliselt statistilise optimeerimise ülesannete lahendamiseks (näiteks toitekeskkonna koguselise koostise või statsionaarsete režiimide tehnoloogiliste parameetrite optimeerisel). Formaalsete matemaatiliste mudelite ehitamine on seotud eksperimenditeooria ja regressioonanalüüsiga.

Järgnevalt vaatame lähemalt füsioloogilisi ja mittestruktuurseid mudeleid.

### **5.2.1 Füsioloogilised mudelid**

Fermentatsiooniprotsessi täpse matemaatilise mudeli konstrueerimine on tömahukas ettevõtmine. Seetõttu on loogiline püüd konstrueerida juhtimissüsteeme, mis ei vaja matemaatilisi mudeleid. Sellised füsioloogilised mudelid rajanevad inimese arusaamal Pasteuri, Crebry jm. efektidest

Näiteks pagaripärmi aeroobse kultiveerimise juhtimiseks võib kasutada järgmisi fermentatsiooniprotsessis ilmnevaid seaduspärasusi:

- Rakkude kasvukiirus on maksimaalne ja kõrvalprodukte (etanooli) ei teki, kui substraadi kontsentratsioon on kriitilisel tasemel.

- Kasvukiirus on optimaalne, kui kasvu piirkiirus muutub vastavalt määratud trajektoorile.
- Kasvukiirus on optimaalne, kui etanooli kontsentratsioon ei muutu.
- Kasvukiirus on maksimaalne etanooli tekkimiseta vabal kasvul ning toiteaine maksimaalse kiirusega etteandmisel.

Samuti on võimalik mudelit koostada kasutades nende reeglite kombinatsioone või muid analoogilisi reegleid. Analoogia hägusates mudelites kasutatavate reeglitega on silmaga nähtav

Füsioloogilistes mudelites ja vastavais juhtimissüsteemides kasutatakse enamasti järgmisi mõõdetavaid füsioloogiliseid muutujaid:

### 1. Hapnikutarbimine

Aeroobse fermentatsiooni puhul eksisteerib selge korrelatsioon mikroorganismide kasvu kiiruse ja hapnikutarbimise vahel

pärmikultuuridel:

$$\frac{m(O_2)}{m(\text{rakud})} = \frac{1,5}{Y} - 1,33$$

bakteriaalsetel kultuuridel:

$$\frac{m(O_2)}{m(\text{rakud})} = \frac{1,5}{Y} - 1,5$$

kus Y on majanduslik tegur

### 2. Lämmastikutarbimine

Lämmastik on süsiniku ja hapniku järel tähtsuset kolmas biomassi rakkude koostisosa. (moodustab 8-12% rakkude kuivmassist). Lämmastikutarbimine lubab seega usaldusväärset hinnata biomassi kasvu. Põhiliseks lämmastikku sisaldavaks komponendiks on ammoniaak, mille tarbimine on tavaliselt seotud keskkonna pH muutumisega. Neutraliseeriva lahuse kulu seega peegeldab lämmastikutarbimist.

Juhul kui lämmastiku kasutatakse ka produkti sünteesimiseks, ei pea aga eelnev väide nii hästi paika.

### 3. Süsinikhappe eraldus

Süsinikhappe eraldumine iseloomustab kasvu aktiivsust. Biomassi ühiku kohta eralduva gaasilise süsinikhappe kogus sõltub sellistest faktoritest nagu biosünteesi kataboolne kulg, anaboolse energia tarbimisega kataboolse energia genereerimise efektiivsus.

Paljude fermentatsiooni liikide puhul (nii aeroobsete kui anaeroobsete) puhul on need faktorid konstantsed, mistõttu süsinikhappe eraldumise kiirus on hea kasvu indikaator.

Et süsinikhape on katabolismi lõpp-produkt peegeldab tema eraldumine ATP moodustumist.

Teiste fermentatsiooniprotsessi iseloomustavate ja juhtimiseesmärkidel kasutatavate katabolismi produktideks on katabolismi anaeroobse kulgemise lõpp-produktid nagu etanool, laktaat, atsetaat, metaan jms. Nende produktide moodustumine on tihti seoses suhtelise kasvukiiruse  $\mu$  ja rakkude biomassiga vastavalt järgmisele sõltuvusele.

$$\frac{dp}{dt} = \alpha \mu x + \beta x$$

avaldis 5-1

$\alpha$  - kasvuga soetud koefitsent  
 $\beta$  - kasvuga mittesoetud koefitsent.

Selliste produktide suhteline kasvukiirus on lineaarselt seotud vastavalt biomassi suhtelise kasvukiirusega.

$$q_p = \frac{1}{x} \frac{dp}{dt} = \alpha \mu + \beta$$

avaldis 5-2

#### 4. Soojuseeraldus

Biokeemilise reaktsiooni tulemusena eraldub fermentatsiooniprotsessis soojust. Fermentatsiooniprotsessi temperatuur on stehhiomeetriliselt seotud biomassi kasvu ja produkti moodustumisega. Soojuseeralduse kiirus on proportsionaalselt seotud biomassi kasvu kiiruse ja eralduva soojuse üldkogus on proportsionaalne biomassi kontsentratsiooniga fermentaatoris. Biomassi ühiku kohta eralduv soojushulk sõltub substraadi kasutamise efektiivsusest. Seega võib soojuseeraldust kasutada mikroorganismide kasvu kaudseks määramiseks. Rea bakterite, pärmide ja hallituste soojuseeraldus fermentatsiooni käigus korreleerub samuti hästi hapnikutarbimise kiirusega. Samuti on olemas seos süsinikhappe gaasi eraldumise kiiruse ja soojuseeralduse vahel.

Soojuseralduse informatsiooni praktiliseks kasutamiseks on vajalik reguleerida soojus-  
tasakaalu fermentaatoris, mis osutub keeruliseks tehniliseks ülesandeks.

## 5.2.2 Mittestruktuursed mudelid

Mittestruktuursed mudelid kujutavad endast biotehnoloogias kõige enam rakendust leidnud matemaatiliste mudelite tüüpi. Mittestruktuursete mudelite puhul eeldatakse, et biosünteesi protsessis osaleb üksainus homogeenselt kasvav mikroorganism, et fermentaatoris toimub ideaalne segamine ja et fermentatsiooniprotsessi võib vaadelda kui determineeritud protsessi. Mudelite koostamisel on aluseks materiaalse tasakaalu tingimused fermentaatoris.

#### Fermentatsioonikeskkonna massi bilansi võrrand:

*fermentatsioonikeskkonna massi akumulatsiooni kiirus = toitesubstraadi lahuse laekumise kiirus - fermentatsioonikeskkonna väljutamise kiirus (kg/h)*

Matemaatilisel kujul näeb see välja nii:

$$\frac{dV\rho_1}{dt} = F_1\rho_t - F_2\rho_1$$

avaldis 5-3

kus  $V$  on fermentatsioonikeskkonna ruumala

$\rho_1$  fermentatsioonikeskkonna tihedus

$\rho_t$  toitelahuse tihedus

$F_1$  toitesubstraatlause etteandmise mahuline kiirus

$F_2$  fermentatsioonikeskkonna väljaviimise mahuline kiirus

Praktikas  $\rho_t \approx \rho_1$ , seega avaldis lihtsustub:

$$\frac{dV}{dt} = F_1 - F_2 \quad \text{avaldis 5-4}$$

Biosünteesi produktide tasakaalu võrrand:

*biosünteesi produktide akumulatsiooni kiirus fermentaatoris = produkti biosünteesi kiirus fermentaatoris - biosünteesi produkti fermentaatorist väljaviimise kiirus (kg/h)*

matemaatilisel kujul:

$$\frac{dVp_i}{dt} = r_{pi}V - F_2p_i \quad \text{avaldis 5-5}$$

kus  $p_i$  - i-nda produkti (sealhulgas ka biomassi) kontsentratsioon fermentatsioonikeskkonnas.  
 $r_{pi}$  - toitekeskkonna komponentide biokeemiliste muundumiste resultaadina i-nda biosünteesi produkti moodustumise kiirus fermentatsioonikeskkonna mahuühiku kohta.

Toitesubstraadi komponentide bilansi võrrand:

*toitesubstraadi komponendi akumulatsioonikiirus fermentaatoris = toitesubstraadi komponendi fermentaatorisse laekumise kiirus - toitesubstraadi komponendi tarbimise kiirus fermentaatoris - toitesubstraadi komponendi fermentaatorist väljaviimise kiirus (kg/h)*

matemaatilisel kujul:

$$\frac{dVs_i}{dt} = F_1s_{ti} - r_{si}V - F_2s_i \quad \text{avaldis 5-6}$$

kus:  $s_i$  - toitesubstraadi i-nda komponendi kontsentratsioon fermentaatoris  
 $s_{ti}$  - toitesubstraadi i-nda komponendi kontsentratsioon toitelahuses  
 $r_{si}$  - toitesubstraadi i-nda komponendi tarbimiskiirus biosünteesi protsessi tulemusena fermentatsioonikeskkonna mahuühiku kohta.

Mikroorganismide kultuuri füsioloogiline olukord peegeldub võrrandites biokeemilise muundumise kiiruse läbi ( $r_{si}$ ,  $r_{pi}$ ). Objektiivsemate näitajatena kasutatakse aga mitte absoluutseid, vaid suhtelisi kiirusi, mis on taandatud biomassi ühikule.

$$R_{pi} = q_{pi}x \quad \text{avaldis 5-7}$$

$$r_{si} = q_{si}x, \quad \text{avaldis 5-8}$$

kus  $x$  on biomassi kontsentratsioon

$q_{pi}$  - biosünteesi i-nda produkti moodustumise suhteline kiirus (biomassi puhul kasutatakse tähistust  $\mu$ )

$q_{si}$  - toitesubstraadi i-nda komponendi tarbimise suhteline kiirus.

Mainitud suhtelised kiirused kujutavad endast fermentatsiooniprotsessi sisemiste ja väliste tingimuste (toitesubstraadi komponentide kontsentratsioon, biosünteesi produktid, keskkonna pH ja temperatuur, segamise ja aeratsioonitingimused, protsessi ajalugu jne.) keerukaid funktsioone. Matemaatilise mudeli töötlemine konkreetse optimeerimis- või juhtimisülesande lahendamiseks seisneb nende sõltuvuste leidmises (identifitseerimises).

**Kui  $F_1 = F_2 = 0$ , on tegu perioodilise protsessiga,  $F_1 \neq 0$ ,  $F_2 = 0$  puhul on tegu poolperioodilise protsessiga,  $F_1 = F_2 \neq 0$  puhul leiab fermentaatoris aset katkematu protsess.**

Praktiliste ülesannete lahendamisel piirduakse 1-4 võrrandiga. Võrrandid koostatakse ainult oluliste muutujate jaoks: mikroorganismide biomass, biosünteesi lõppprodukt, toitesubstraadi komponendi limiteeriv kasv, fermentatsioonikeskkonna ruumala. Teatud juhtudel kasutatakse ka teisi muutujaid nagu biosünteesi vaheprodukt, mikroorganismide poolt tarbitav hapnik jm. Optimeerimis- ja juhtimisülesannete lahendamiseks mõeldud matemaatilise mudeli dimensioon peab olema ratsionaalne, kuna mudeli keerukus raskendab optimeerimise matemaatiliste meetodite efektiivset rakendamist.

Kirjanduses on ära toodud suur arv funktsionaalsete sõltuvuste variante, mis seovad biomassi kasvu, substraadi tarbimise ja lõppprodukti moodustumise suhtelisi kiirusi fermentatsiooniprotsessi väliste tingimustega. **Põhilisteks biosünteesi kulgu mõjutavateks faktoriteks on substraadi, biomassi ja biosünteesi produkti kontsentratsioonid, temperatuur, fermentatsioonikeskkonna pH jt.**

Levinumaks biomassi moodustumise suhtelise kiiruse  $\mu$  sõltuvust limiteeriva substraadi kontsentratsioonist esitavaks vormiks on Monod' võrrand:

$$\mu(s) = \frac{\mu_m s}{s + K_s} \quad \text{avaldis 5-9}$$

Laialt on kasutusel ka Monod' võrrandite erinevad modifikatsioonid, aga ka teistsugused sõltuvuste tüübid

$$\mu(s) = \frac{\mu_m s}{\frac{s^2}{K_i} + s + K_s} \quad \text{avaldis 5-10}$$

$$\mu(s) = k_1 s - k_2 \quad \text{avaldis 5-11}$$

$$\mu(s, x) = \frac{\mu_m s}{s + K_s} - \frac{K_x}{K_x + x} \quad \text{avaldis 5-12}$$

$$\mu(s, x) = k_1 s - k_2 x \quad \text{avaldis 5-13}$$

$$\mu(s, p) = \frac{\mu_m s}{s + K_s} - K_p p \quad \text{avaldis 5-14}$$

jne.

Samuti on välja töötatud teatud tüüpi sõltuvused substraadi tarbimise suhtelise kiiruse

$$q_s(\mu) = \frac{\mu}{Y_{x/s}} \quad \text{avaldis 5-15}$$

$$q_s(\mu) = \frac{\mu}{Y_{x/s}} + m \quad \text{avaldis 5-16}$$

$$q_s(\mu, q_p) = \frac{\mu}{Y_{x/s}} + \frac{q_p}{Y_{p/s}} + m \quad \text{avaldis 5-17}$$

... ja produkti moodustumise suhtelise kiiruse ( $q_p$ ) jaoks:

$$q_p(\mu) = \alpha\mu + \beta \quad \text{avaldis 5-18}$$

$$q_p(\mu) = \alpha\mu^2 + \beta\mu + \gamma \quad \text{avaldis 5-19}$$

$$q_p(x) = q_m \frac{1}{1 + x/k} \quad \text{avaldis 5-20}$$

$$q_p(p) = q_m - kp \quad \text{avaldis 5-21}$$

Temperatuuri, pH ja teiste selliste muutujate mõju suhtelistele kasvukiirustele ( $\mu$ ,  $q_{pi}$ ,  $q_s$ ) arvestamiseks, käsitletakse neid matemaatilise mudeli parameetritena, nii et kontsentratsioonimuutujaid determineerivad funktsionaalsed sõltuvused osutuvad omakorda funktsionaalselt sõltuvaks keskkonnaparameetritest.

Näiteks  $\mu(s, \text{pH})$  avaldub kujul:

$$\mu(s, \text{pH}) = \mu(s, K(\text{pH})), \quad \text{avaldis 5-22}$$

kus K-ga tähistatud on mudeli parameetrid.

## 5.3 Fermentatsiooniprotsesside optimeerimine

### 5.3.1 Perioodiliste fermentatsiooniprotsesside optimeerimine

Teatud tehnoloogiliste parameetrite (temperatuur, pH jne.) optimeerimine perioodilise fermentatsiooniprotsessi kestel annab paremaid resultate võrreldes protsessidega, milles tehnoloogilisi parameetreid hoitakse ühel kindlal tasemel.

Perioodilise fermentatsiooniprotsessi optimeerimisülesanne seisneb tehnoloogiliste parameetrite profiilide niisuguses määramises, mis tagab optimeerimiskriteeriumi maksimeerimise. Ülesande klassikaline lahendustee seisneb matemaatilise mudeli väljatöötamises ning optimaaljuhtimise teooria rakendamises mudelile.

Ülesande praktiliseks lahendamiseks kasutatakse lihtsustatud meetodeid, mis arvutavad välja ligikaudsed optimaalsed profiilid, mida võib edaspidi korrigeerida n.-ö. töö käigus.

Matemaatilisel mudelil, millel optimeerimismeetod töötab, on järgmine struktuur:

$$\begin{cases} \frac{dx}{dt} = x\mu(s, u) \\ \frac{ds}{dt} = -xq_s(s, u) \\ \frac{dp}{dt} = xq_p(s, u) \end{cases} \quad \text{avaldis 5-23}$$

kus  $u$  on optimeeritavate parameetrite vektor.

$$u = \begin{bmatrix} \text{temperatuur} \\ pH \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

Optimeerimisülesandeks on lõpp-produkti koguse maksimeerimine

$$J = p(T) \rightarrow \max, \quad \text{avaldis 5-24}$$

kus  $T$  on fermentatsiooniprotsessi kestus, mis võib olla nii fikseeritud kui vaba.

Peale mudeli teisendusi saadakse produkti koguse arvutamiseks järgmine integraalvõrrand:

$$p(T) = \int_{s(T)}^{s(0)} \frac{q_p(s, u)}{q_s(s, u)} ds + p(0) \quad \text{avaldis 5-25}$$

kus  $s(0)$  ja  $s(T)$  on vastavalt substraadi alg- ja lõppkontsentratsioonid fermentaatoris.

$p(0)$  - produkti algkontsentratsioon fermentaatoris (integreerimiskonstant).

Võttes aluseks Bellmani dünaamilise planeerimise optimaalsusprintsipi, saame järgmised optimaaljuhtimise strateegiad:

a) fikseerimata kestusega fermentatsiooniprotsess

$$p_{\max}(T) = \int_{s(T)}^{s(0)} \left( \max \frac{q_p}{q_s} \right) ds + p(0) \quad \text{avaldis 5-26}$$

Igal protsessi momendil määratakse parameetrite vektor  $\mathbf{u}$  sellise substraadi kontsentratsiooni funktsioonina, mis maksimiseerib suhet  $q_p/q_s$ . Sisuliselt tähendab see seda, et igal ajahetkel maksimeeritakse protsessi lõpp-produkti ökonoomilist tegurit  $Y$ :

$$Y = q_p / q_s = \frac{dp}{ds} \quad \text{avaldis 5-27}$$

b) fikseeritud kestusega fermentatsiooniprotsess

$$p_{\max}(T) = \max_{\substack{u \in U \\ s \in [s(0), s(T)]}} \left[ \int_{s(T)}^{s(0)} \frac{q_p}{q_s} ds + \max_{\substack{u \in U \\ s \in [s(t), s(T)]}} \int_{s(T)}^{s(t)} \frac{q_p}{q_s} ds \right] \quad \text{avaldis 5-28}$$

kus  $0 < t < T$

Parameetrite vektor  $\mathbf{u}$  määratakse suvalisel ajamomendil sellise substraadi kontsentratsiooni ja protsessi lõpuni jääva ajavahemiku  $\Delta t = T - t$  funktsioonina, mis maksimiseerib avaldise teist liiget. Sisuliselt tähendab see, et maksimeeritakse ökonoomilise koeffitsendi keskmist väärtust fikseeritud protsessi aja vältel.

Optimiseerimisülesanne lahendatakse kolmes etapis

1. Ülesande lahendamiseks vajalike algandmete kogumine

Perioodilise kultiveerimise eksperimentide planeerimine ja realiseerimine. Eksperimentide eesmärgiks on optimeeritavate parameetrite poolt fermentatsiooniprotsessile avaldatava mõju uurimine erinevatel nivoodel. Igas eksperimendis hoitakse optimeeritavaid parameetreid eelnevalt kindlaksmääratud nivool. Plaani koostamiseks võib kasutada levinud eksperimentide planeerimise meetodeid.

Iga eksperimendi käigus mõõdetakse substraadi ja lõpp-produkti kontsentratsioonid erinevatel ajamomentidel. Resultaadina saadakse eksperimendiandmete tabel kujul

$$s_i(t_j) \text{ ja } p_i(t_j)$$

kus  $i$  - eksperimendi järjekorranumber

## 2. Eksperimendiandmete aproksimeerimine

Identifitseeritakse funktsioonid  $s_i(t)$  ja  $p_i(t)$ , mis aproksimeerivad katseandmeid  $s_i(t_j)$  ja  $p_i(t_j)$ . Selleks võib kasutada suvalist diskreetsete mõõtmiste matemaatilist aproksimeerimise meetodit.

## 3. Funktsiooni $Y_i = Y(s)$ tuletamine

Funktsioon arvutatakse järgmiselt:

$$Y_i(t) = \left( \frac{q_p}{q_s} \right)_i(t) = \frac{\left( \frac{dp}{dt} \right)_i(t)}{\left( \frac{ds}{dt} \right)_i(t)},$$

avaldis 5-29

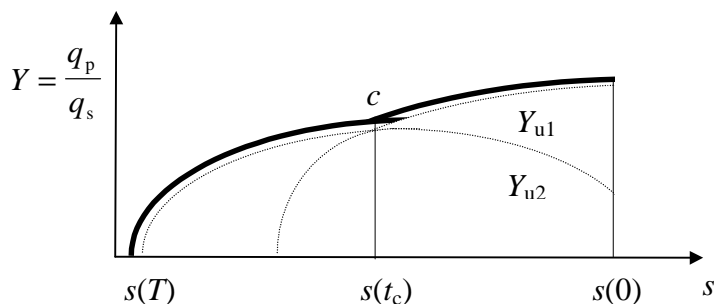
kus sõltuvused  $\left( \frac{dp}{dt} \right)_i$  ja  $\left( \frac{ds}{dt} \right)_i$  saadakse identifitseeritud funktsioonide  $s_i(t)$  ja  $p_i(t)$  diferentseerimise teel.

Siit tuleneb, et optimeerimisülesande lahendamise kvaliteet sõltub eksperimendiandmete aproksimeerimise täpsusest.

## 4. Juhttoimete optimaalsete parameetrite arvutus (mittefikseeritud kestusega protsessi näitel)

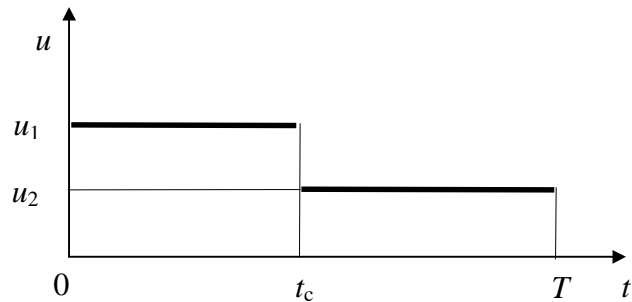
Antud juhul tuleneb fermentatsiooni kestus kaudselt substraadi üldkogusest fermentaatoris. Juhttoimete optimaalsete profiilide määramiseks tuleb eelmises etapis töödeldud eksperimendiandmed esitada graafiliselt tasandil  $Y(s) \times s$ . Joonis 5-3 kujutab optimaaljuhtimise strateegiat.

joonis 5-3



Oletame, et eksperimendiandmete töötlemise tulemusena saame 2 faasitrajektoori  $Y_{u1}$  ja  $Y_{u2}$  (suurem faasitrajektoore hulk ülesande lahendamist oluliselt keerulisemaks ei tee), mis on joonisel kujutatud kriipsjoontega. Vastavalt juhtimisstrateegiale (avaldis 5-26) peab optimaalne faasitrajektoori iga  $s$ -i väärtuse korral ühtima parajasti maksimaalse faasitrajektooriga, mis garanteerib suhte  $q_p/q_s$  võimaliku maksimumväärtuse protsessi kestel. Pinna suurus, mille selline trajektoori joonisel katab on võrdeline fermentatsiooni kestel toodetud produkti kogusega. Strateegia realiseerimiseks tuleb kasutada vastavaid juhttoime nivooisid ( $u_1$  ja  $u_2$ ) ning ümberlülitumise momendi määrab substraadi kontsentratsioon  $s(t_c)$  (joonis 5-4).

joonis 5-4



### 5.3.2 Poolperioodiliste fermentatsiooniprotsesside optimeerimine

Poolperioodilise kultiveerimise puhul viiakse toitained fermentaatorisse pidevalt ning optimeeritavaks suuruseks on nende fermentaatorisse viimise kiirus protsessi vältel. Ülesande lahendamisel eeldatakse tavaliselt, et biomassi kasvu ja produkti moodustumisel on oluline üksainus toitesegu komponent, teised komponendid protsessi kulgu ei mõjuta. Samuti eeldatakse välistingimuste s.o tehnoloogiliste parameetrite, mida eelmises jaotises optimeeriti, püsivust ühel kindlal tasemel protsessi vältel.

Matemaatilised mudelid, mida antud ülesande lahendamisel kasutatakse rajanevad massi-tasakaalu võrranditel, ja avalduvad järgmisel kujul:

$$\left\{ \begin{array}{l} \frac{d(xV)}{dt} = \mu(x, s, p)xV, x(0) = x_0 \\ \frac{d(sV)}{dt} = -q_s(x, s, p)xV + s_n F, s(0) = s_0 \\ \frac{d(pV)}{dt} = q_p(x, s, p)xV, p(0) = p_0 \\ \frac{dV}{dt} = F \end{array} \right.$$

avaldis 5-30

kus  $x$ ,  $s$  ja  $p$  on vastavalt biomassi, substraadi ja produkti kontsentratsioonid fermentatsioonikeskkonnas.

$s_n$  - substraadi kontsentratsioon toitelahuses

$V$  - fermentatsioonikeskkonna maht

$F$  - toitelahuse laekumise kiirus

Piirangud:

fermentatsioonikeskkonna lõplik maht:

$$V(T) = V_k \quad \text{avaldis 5-31}$$

toitelahuse laekumise kiirus:

$$0 = F_{min} \leq F \leq F_{max} \quad \text{avaldis 5-32}$$

Ülesandeks on optimaalse juhtimisprofili  $F(t)$  määramine, mis maksimeeriks sihi-funktsiooni, mis on sõltuvuses fermentatsiooniprotsessi lõppresultaatidest

$$J(F) = f(x_k, s_k, p_k, V_k, T) \quad \text{avaldis 5-33}$$

Protsessi kestus  $T$  võib olla kindlaksmääratud või vaba.

Võrrandisüsteemi 5-30 võib ümber kirjutada tähistades:

$$x_1 = X = xV$$

$$x_2 = S = sV$$

$$x_3 = P = pV$$

$$x_4 = V$$

$$x_5 = T$$

saame seega, et

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} \mu x_1 \\ -q_s x_1 \\ q_p x_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ s_n \\ 0 \\ 1 \\ 0 \end{bmatrix} F \quad \text{avaldis 5-34}$$

Maatrikskujul, kui:

$$\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T$$

$$\mathbf{g} = [\mu x_1, -q_s x_1, q_p x_1, 0, 1]^T$$

$$\mathbf{h} = [0 \ s_n, 0 \ 1 \ 0]^T$$

võime kirjutada ülesande kompaktses kujus:

$$\frac{dx}{dt} = \mathbf{g}(\mathbf{x}) + \mathbf{h}F, \mathbf{x}(0) = \mathbf{x}_0 \quad \text{avaldis 5-35}$$

$$J(F) = f(\mathbf{x}(T)) \quad \text{avaldis 5-36}$$

### Lahendamise üldskeem

Vastavalt maksimumi printsiibile, leitakse optimaalne lahend Hamiltoni funktsiooni maksimeerimise teel.

$$H = \lambda^T (\mathbf{g}(\mathbf{x}) + \mathbf{h}F) \quad \text{avaldis 5-37}$$

kus  $\lambda^T = [\lambda_1 \ \lambda_2 \ \lambda_3 \ \lambda_4 \ \lambda_5]$  on seosemuutujate vektor, mis on seotud  $\mathbf{x}$ -ga võrrandite

$$\frac{d\lambda}{dt} = \frac{\delta H}{\delta \mathbf{x}} = - \left( \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \right)^T \lambda \quad \text{avaldis 5-38}$$

kaudu, kus  $\lambda$  lõppväärtused saame tingimusest:

$$\lambda_i(T) = \frac{\delta t}{\delta x_i}(T), \quad i = 1,2,3,4,5 \quad \text{avaldis 5-39}$$

Vastavalt ülesande optimaalsuse tingimustele osutub Hamiltoni funktsioon konstantseks optimaalsuse trajektoiril. Tähistame hamiltoniaani konstantse väärtuse  $H^*$ -ga

$$H = H^* \quad \text{avaldis 5-40}$$

Piiramata kestusega protsessi puhul  $H^* = 0$ , etteantud kestusega protsessi puhul:

$$H^* = H(T) = \lambda^T(T)(\mathbf{g}(x(T)) + \mathbf{h}F(T)) \quad \text{avaldis 5-41}$$

tähistades  $\Psi(x, \lambda) = \lambda^T \mathbf{g}$  ja  $\Phi(x, \lambda) = \lambda^T \mathbf{h}$ , avaldub hamiltoniaan 5-41 kujul:

$$H = \Psi(x, \lambda) + \Phi(x, \lambda)F \quad \text{avaldis 5-42}$$

Avaldisest 5-42 tuleneb, et hamiltoniaani maksimaalväärtuse tagab  $F(t)$  sõltuvalt  $\Phi(x, \lambda)$  märgist. Kui  $\Phi$  märk on positiivne, peab  $F$  olema maksimaalne, kui negatiivne siis minimaalne. Kui  $\Phi$  on ajaintervalli jooksul null, toob see sisse  $F$ -i määramatuse ning viimane tuleb määrata erimeetodil.

Niisiis seisneb protsessi optimaalne juhtimine tava ja erijuhtimise rakendamises sõltuvalt  $\Phi$  märgist.

$$\begin{cases} F_{\max}, & \text{kui } \Phi > 0 \\ F_{\text{eri}}, & \text{kui } \Phi = 0, t_j \leq t \leq t_{j+1} \\ F_{\min}, & \text{kui } \Phi < 0 \end{cases} \quad \text{avaldis 5-43}$$

Üheks erijuhtimise arvutamise mooduseks on avaldise  $\Phi(x, \lambda)$  korduv diferentseerimine aja järgi, mis peab võrduma nulliga erijuhtimise ajal.

$$\Phi = 0, \quad \frac{d^k \Phi}{dt^k} = 0, \quad k = 1, 2, \dots \quad \text{avaldis 5-44}$$

$\Phi$  diferentseerimist aja järgi jätkatakse kuni avaldises on võimalik ilmutada juhttoime  $F$ .

$$\frac{d\Phi}{dt} = \left( \frac{d\lambda}{dt} \right)^T \mathbf{h} = -\lambda^T \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \mathbf{h} = 0, \quad \text{avaldis 5-45}$$

avaldis 5-46

$$\frac{d^2 \Phi}{dt^2} = - \left( \frac{d\lambda}{dt} \right)^T \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \mathbf{h} - \lambda^T \frac{\delta \left( \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \mathbf{h} \right)}{\delta \mathbf{x}} \frac{d\mathbf{x}}{dt} = \lambda^T \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \mathbf{h} - \lambda^T \frac{\delta \left( \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \mathbf{h} \right)}{\delta \mathbf{x}} (\mathbf{g} + \mathbf{h}F) = 0$$

viimasest võrrandist avaldub  $F_{\text{eri}}$  :

$$F_{\text{eri}} = \frac{\lambda^T \left( \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \mathbf{h} - \frac{\delta \left( \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \mathbf{g} \right)}{\delta \mathbf{x}} \mathbf{g} \right)}{\lambda^T \frac{\delta \left( \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \mathbf{h} \right)}{\delta \mathbf{x}} \mathbf{h}} \quad \text{avaldis 5-47}$$

Avaldis 5-47 määrab erioptimaaljuhtimise funktsionaalse sõltuvusena põhi- ja vahemuutujatest. Selle kasutamiseks tuleb samaaegselt lahendada nii teadaolevate algtingimustega põhivõrrandid (5-35) kui vahemuutujatega võrrandid (5-38) arvestades seatud lõppväärtusi (5-39).

Arvutusprotseduuri lihtsustamiseks võib kasutada optimaalsuse täiendavaid tingimusi (5-40 ja 5-44),  $k=1$ :

$$\lambda^T \mathbf{g} = H^* \quad \text{avaldis 5-48}$$

$$\lambda^T \mathbf{g} = 0 \quad \text{avaldis 5-49}$$

$$\lambda^T \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \mathbf{h} = 0 \quad \text{avaldis 5-50}$$

Need kolm mittetriviaalset võrrandit seovad 5 vahemuutujat ja võimaldavad avaldada suvalised kolm vahemuutujat ülejäänud kahe ning konstantse hamiltoniaani väärtuse  $H^*$  kaudu.

Sellisel juhul pole optimiseerimisülesande lahendamiseks vaja teada rohkem kui kahte vahemuutujat ja hamiltoniaani väärtust.

Paljudes praktilistes ülesannetes puudub sihifunktsiooni (5-36) struktuuris fermentatsiooniprotsessi kestus  $T$ . See lihtsustab märgatavalt ülesande lahendamist kuna sel juhul  $\lambda_5 = 0$  ning ülesande lahendamiseks peame teadma kõigest ühte vahemuutujat ning  $H^*$  väärtust.

### 5.3.3 Toitevoe optimaalsed profiilid

Eelmisest jaotisest selgus, et toitesegu optimaalse juhtimise algoritm lülitub üldjuhul kolme erineva juhtimisrezhiimi vahel (avaldis 5-43). Kui ümberlülitumiste järjekord ja optimaalsed juhtimisprofiilid määratakse apriorselt, taandab see juhtimisülesande lihtsalt lülitusmomentide õigeaegsele määramisele.

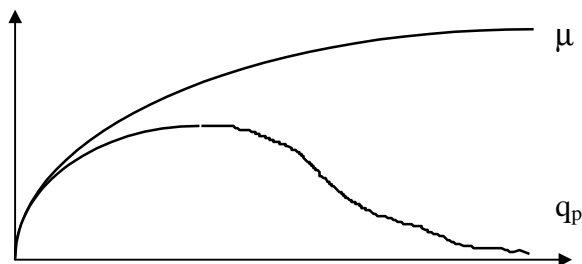
Kui biomassi ja produkti moodustumise suhteliste kiiruste  $\mu$  ja  $q_p$  sõltuvus substraadi kontsentratsioonist  $s$  on teada, võimaldab see ilma keerulisemate arvutusteta määrata kindlaks ligilähedasel optimaalse juhttoime profiili.

Üldise strateegia järgi püütakse protsessi algosas tagada biomassi optimaalne juurdekasv ning teises osas sundida biomassi vastava juhtimisega optimaalselt tootma lõpp-produkti.

Vajadus erijuhtimise järele tekib, kui  $\mu$  ja/või  $q_p$  on ilmutatud maksimumiga mitte-monotoonsed funktsioonid. Suhteliste juurdekasvude  $\mu$  ja  $q_p$  muutumise iseloomu järgi võib protsessid jagada kolme liiki:

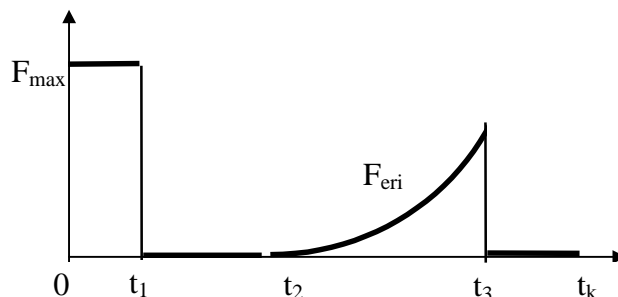
### 5.3.3.1 Monotoonselt kasvav $\mu$

joonis 5-5



Et biomassi kasvu suhteline kiirus on substraadi kontsentratsioonist sõltuv monotoonselt kasvav funktsioon, paljunevad rakud optimaalselt kõrgemate substraadi kontsentratsioonide juures. Järelikult peab toitmise kiirus olema protsessi algosas maksimaalne. Kui aga biomassi kontsentratsiooni tõstmisest tulenev kasu osutub väiksemaks produkti moodustumisest tulenevat kasust, mida päsrib kõrge substraadi kontsentratsioon, lülitatakse toitmine välja (joonis 5-6, ajahetk  $t_1$ ). Selline olukord kestab seni, kuni substraadi, biomassi ja produkti kontsentratsioonid rahuldavad erijuhtimise tingimust, misjärel viimane ka sisse lülitatakse (ajahetk  $t_2$ ) ja mis töötab seni kuni fermentaator täitub (ajahetk  $t_3$ ). Seejärel toite andmine fermentaatorisse lõpetatakse ning fermentatsioon jätkub kuni teiste tingimustega määratud fermentatsiooni lõppmomentini  $t_k$ .

joonis 5-6

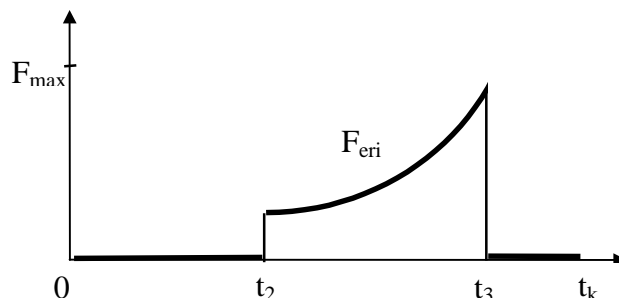


Fermentatsiooni lõppmoment võib olla näiteks ajaliselt fikseeritud või võib lõpetamise tingimuseks olla etteantud substraadi kontsentratsioonini jõudmine, produkti tootmise katkemine või muu taoline tingimus.

Olenevalt fermentatsiooniprotsessi algtingimustest ( $x(0)$ ,  $s(0)$ ,  $p(0)$ ,  $V(0)$ ) võib toitesegu fermentaatorisse andmise optimaalne profiil olla lihtsam kirjeldatud üldvariandist.

Kui substraadi kontsentratsioon algmomentil on väga suur (joonis 5-7), võib fermentatsiooniprotsess alata perioodilise kultiveerimise intervallist kuni substraadi kontsentratsioon langeb ning biomassi ja produkti kontsentratsioonid tõusevad väärtusteni, mis lubavad rakendada erijuhtimist (ajahetk  $t_2$ ).

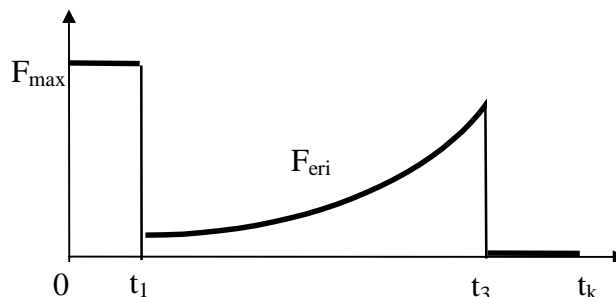
joonis 5-7



Erijuhtimine kestab kuni fermentaatori täitumiseni (ajahetk  $t_3$ ).

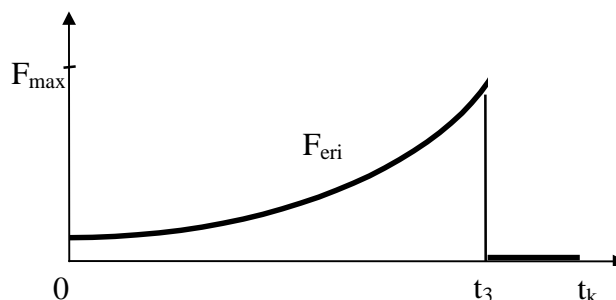
Kui substraadi algkontsentratsioon on madal ja biomassi oma kõrge, siis juhitakse fermentatsiooni algul fermentaatorisse maksimaalne substraadi voog, mis kestab seni kuni on rahuldatud erijuhtimisele ümberlülitamise tingimus. Erijuhtimist sooritatakse kuni fermentaatori täitumiseni (joonis 5-8).

joonis 5-8



Kui on võimalus määrata ise ülesande algtingimusi, siis loomulik oleks valik, mille puhul  $x(0)$ ,  $s(0)$ ,  $p(0)$ ,  $V(0)$  oleksid sellised, mis lubaks koheselt käivitada erijuhtimise. Fermentaatori täitumisel erijuhtimine lõpetatakse (joonis 5-9), ning jäädakse ootama tingimuste täitumist, mis lubavad fermentatsiooni lõpetada.

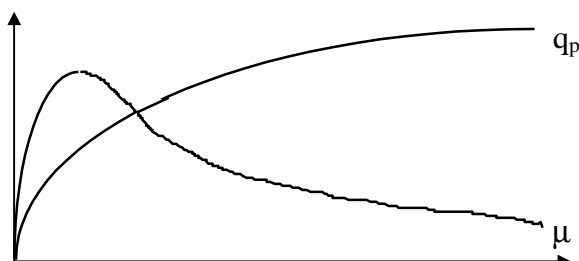
joonis 5-9



Viimane variant peaks ühtlasi tagama parimad resultaadid.

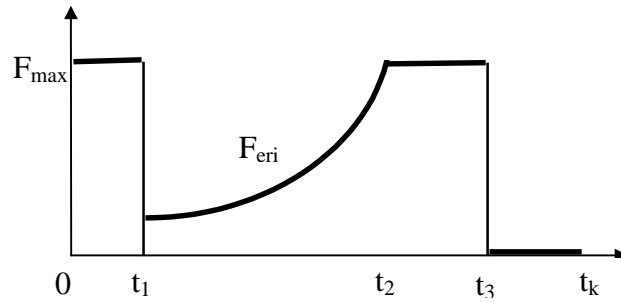
### 5.3.3.2 Monotoonselt kasvav $q_p$

joonis 5-10



Kui substraadi algkontsentratsioon on madal, võib juhtimine alata maksimaalse substraadi koguse fermentaatorisse saatmisega, kuni ajamomendil  $t_1$  saab rahuldatud erijuhtimise tingimus. Erijuhtimist jätkatakse kuni momendini ( $t_2$ ), mil biomassi rakkude arvu suurendamisest tulenev kasu võib kompenseerida produkti tootmisest saadava.

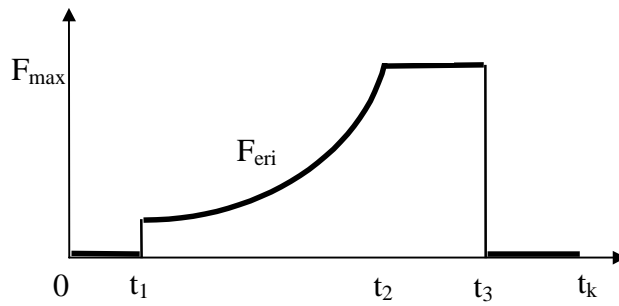
joonis 5-11



Kuni momendini  $t_3$ , mil fermentaator saab täis, juhitakse fermentaatorisse maksimaalne substraadi voog.

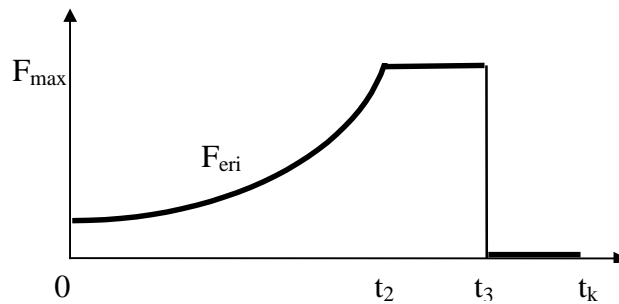
Kui substraadi algkontsentratsioon on kõrge, pole esialgu substraati fermentaatorisse lisada vaja. Momendil  $t_2$ , mil saab rahuldatud erijuhtimise tingimus minnakse üle erijuhtimisele. Juhtimise edasine kulg ei erine eelmisest (joonis 5-12).

joonis 5-12



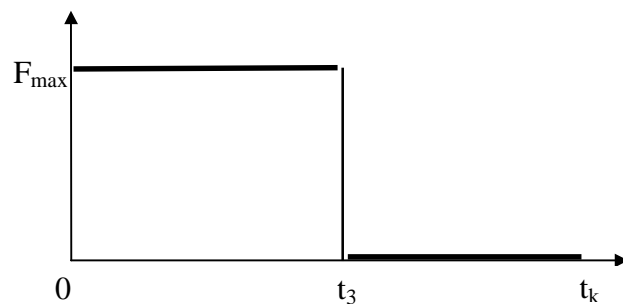
Ka võivad algtingimused lubada koheselt rakendada erijuhtimist (joonis 5-13).

joonis 5-13



Kui biomassi algkontsentratsioon on suur, pole vajadust ressurse biomassi kasvatamisele kulutada, ning substraadi profiil omandab astmelise kuju (joonis 5-14)

joonis 5-14



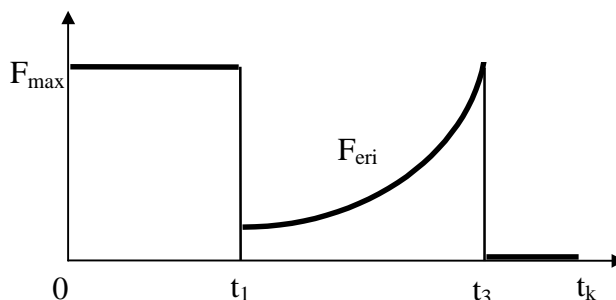
### 5.3.3.3 Mittemonotoonsed $\mu$ ja $q_p$

joonis 5-15



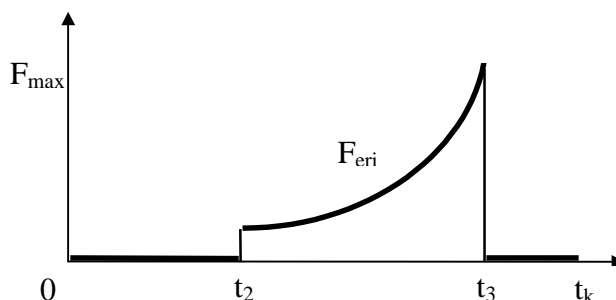
Kuna  $\mu$  on mittemonotoonne funktsioon, kasvab biomass optimaalselt erijuhtimise korral. Eeldades seejuures et substraadi algkontsentratsioon on madal, tuleks alustada maksimaalse substraadi koguse fermentaatorisse paiskamisega, kuni kujuneb välja olukord, mis vastab erijuhtimise tingimusele. Et ka  $q_p$  on mittemonotoonne, peab erijuhtimine olema selline, et selle esimeses osas tagatakskiire biomassi juurdekasv ning lõpposas produkti suur toodang. Joonis 5-16 esitab vastavat juhtimisprofiili

joonis 5-16



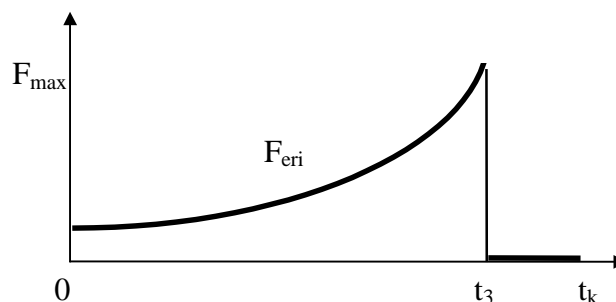
Kui substraadi algkontsentratsioon on suur, ei tehta seni midagi, kuni on tarbitud substraadi liiasus ja sel teel jõutud situatsiooni kus võib rakendada erijuhtimist (joonis 5-17)

joonis 5-17



Ning kui algingimused lubavad alustada erijuhtimisega, näeb vastav juhtimisprofiil välja selline nagu seda kujutab joonis 5-18.

joonis 5-18



#### 5.3.3.4 Kokkuvõtteks

Joonised 5-5, 5-10 ja 5-15 viitavad üsna ilmselt sellele, et eksisteerivad sellised substraadi kontsentratsioonid  $s$ , mille puhul biomassi ja produkti suhtelised juurdekasvud on maksimaalsed. Seega võib oletada, et poolperioodilise fermentatsiooniprotsessi optimaalne juhtimine seisneb fermentaatorisse sellise koguse substraadi  $S$  viimises, millega tagatakse viimase kontsentratsiooni püsimine sellisel (konstantsel) tasemel, mille puhul  $\mu$  või  $q_p$  väärtus on maksimaalne. Kui too hüpotees vastab tõele, taandub juhtimisülesanne üksnes selliste kontsentratsiooni väärtuste väljaselgitamisele ning ajamomentide  $t_1$ ,  $t_2$ ,  $t_3$  ja  $t_k$  õigele määramisele.

Maksimaalset kasvukiirust tagava substraadi kontsentratsiooni määramine eeldab funktsioonide  $\mu(s)$  ja/või  $q_p(s)$  identifitseerimist.

Ülesande võimalikud lahendusvariandid kasutades hägusaid modelleerimis- ja juhtimisvõttes on lähemalt vaatluse all 7-ndas peatükis. Järgmine peatükk on aga pühendatud käesolevas töös kasutatud fermentatsiooniprotsessi mudeli tutvustusele.

## 6. Fermentatsiooniprotsessi mudel

Käesolevas töös hädusa modelleerimise ja juhtimise võimaluste hindamiseks kasutatav objekt kujutab endast poolperioodilise fermentatsiooniprotsessi mudelit, mis on saadud Interneti vahendusel Westminsteri Ülikooli Tööstusliku Juhtimise Keskusest.

Programmselt realiseeritud mudel on vabalt hangitav Interneti aadressil <http://www.wmin.ac.uk/ICC/compete/modcomp.html> ja kujutab endast musta kasti, mida modelleerima ja juhtima on kutsutud huvilised üle kogu maailma. Mõni sõna mudelist:

Poolperioodilise fermentatsiooniprotsessi mudel kehastab fermentatsiooniprotsessi, mille saaduseks on teisejärguline mikroorganismi ainevahetuse saadus. Tootmisprotsessis osalev mikroorganism vajab kahte substraati ( $S1$  ja  $S2$ ) kasvamiseks ja toodangu tekitamiseks. Protsess on juhitud nende kahe substraadi lisamisega fermentaatorisse. Sealjuures on ekspertidele teada järgmised asjaolud:

- Etteantud juhtimisprofiilid (millega mudel on varustatud) annavad mõistliku resultati. Liiga suured või väikesed substraadi nivood fermentaatoris annavad kehvema tulemuse.
- Substraat  $S2$  näib täiendavat substraadi  $S1$  mõju. Üksnes  $S2$  lisamine ei anna mingit toodangut, samal ajal kui üksnes  $S1$  lisamine resulteerub väikeses toodangu hulgas.

Samas ei peeta etteantud juhtimisprofiili parimaks võimalikuks, ning välja tuleks töötada optimaalsed juhtimisprofiilid protsessi tootlikuse tõstmiseks. **Eeldatud on seejuures, et fermentatsioonikeskkonna parameetrid nagu temperatuur ja pH, on konstantsed ning optimaalsed.**

Modelleerimiseks ja juhtimiseks väljapakutud mudel on esitatud ilmutamata kujul, et teha ülesande tingimused lähedaseks olukorraga, mis valitseb reaalses elus. Antud on vaid järgmine:

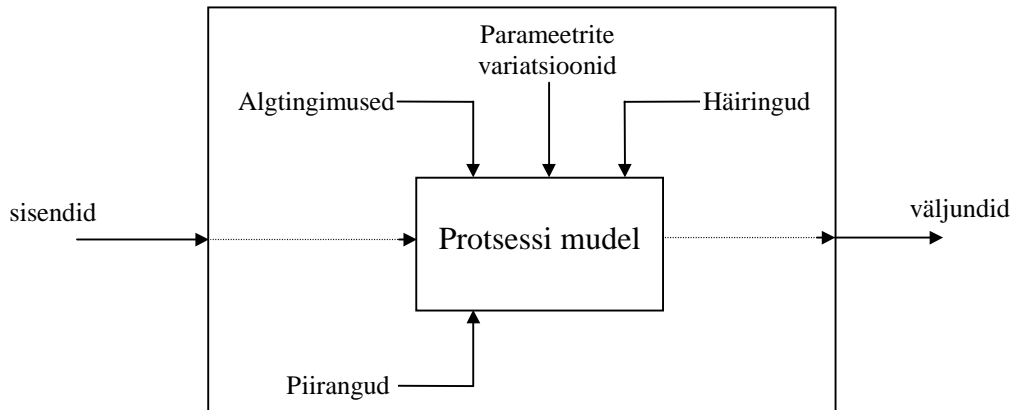
- mudel musta kastina
- tüüpilised sisend-väljundandmed (nominaalsed juhtimisprofiilid ja objekti reageering nendele)
  - mõningad kehtivad piirangud
  - juhtimisülesanne

Must kast sisaldab lisaks peamistele differentsvõrranditele teatud mehhanisme, mis garanteerivad järgmist:

- fermentatsioonikeskkonna komponentide ning nende summaarse mahu väärtused alghetkel varieeruvad teatud ulatuses, s.t. et iga mudeli abil simuleeritud fermentatsioon algab erinevast lähteolekust
- mudeli parameetrid varieeruvad teatud piirides, s.t. et iga mudeli abil simuleeritud fermentatsioon on unikaalne
- protsessi läbiviimisel ilmnevad mittemõõdetavad häiringud
- eksisteerivad piirangud mõnedele protsessi suurustele

Sellist musta kasti kujutab joonis 6-1.

joonis 6-1



Objektil on kaks sisendit - substraatide 1 ja 2 fermentaatorisse pumpamise kiirused  $f_1$  ja  $f_2$ . Mõõdetakse viit suurust, mis on biomassi kontsentratsioon  $x$ , substraadi 1 kontsentratsioon  $s_1$ , substraadi 2 kontsentratsioon  $s_2$ , toodangu kontsentratsioon  $p$  ning fermentatsioonikeskkonna maht  $V$ .

Kõik siintoodud mõõdetavad suurused omavad arvulisi väärtusi, mille ühikuid pole täpsustatud, seega saame edaspidi opereerida vaid anonüümsete kontsentratsiooni-, mahu-, ja ajaühikutega.

Piiratud on fermentatsioonikeskkonna maht ning substraatide fermentaatorisse pumpamise kiirused. Fermentatsioonikeskkonna maksimaalne maht  $V_{\max}$  on 4000 mahuühikut ning kiirused  $f_1$  ja  $f_2$  võivad muutuda piirides  $[0, 50]$ .

Kõik, mis jääb musta kasti sisse, sealhulgas ka sisendite ja väljundite vahelisi seoseid kirjeldavad diferentsvõrrandid on mudeli kasutajale kättesaamatud.

Antud mudelit saab kasutada nii modelleerimiseks kui juhtimiseks, kuna ta simuleerib küllalt hästi reaalse fermentatsioonisüsteemi tööd, kusjuures väljapakutud juhtimiskriteerium  $J$ , mida tasuks maksimeerida, avaldub järgmiselt:

$$J = \frac{\text{toodangu kontsentratsioon} \cdot \text{maht}}{\text{fermentatsiooni kestus}} = \frac{pV}{t_f} \quad \text{avaldis 6-1}$$

Lahtisõnastatult on eesmärgiks saada fermentatsioonist võimalikult lühikese aja jooksul maksimaalne kogus toodangut. Edaspidi nimetame seda juhtimiskriteeriumit kvaliteedinäitajaks.

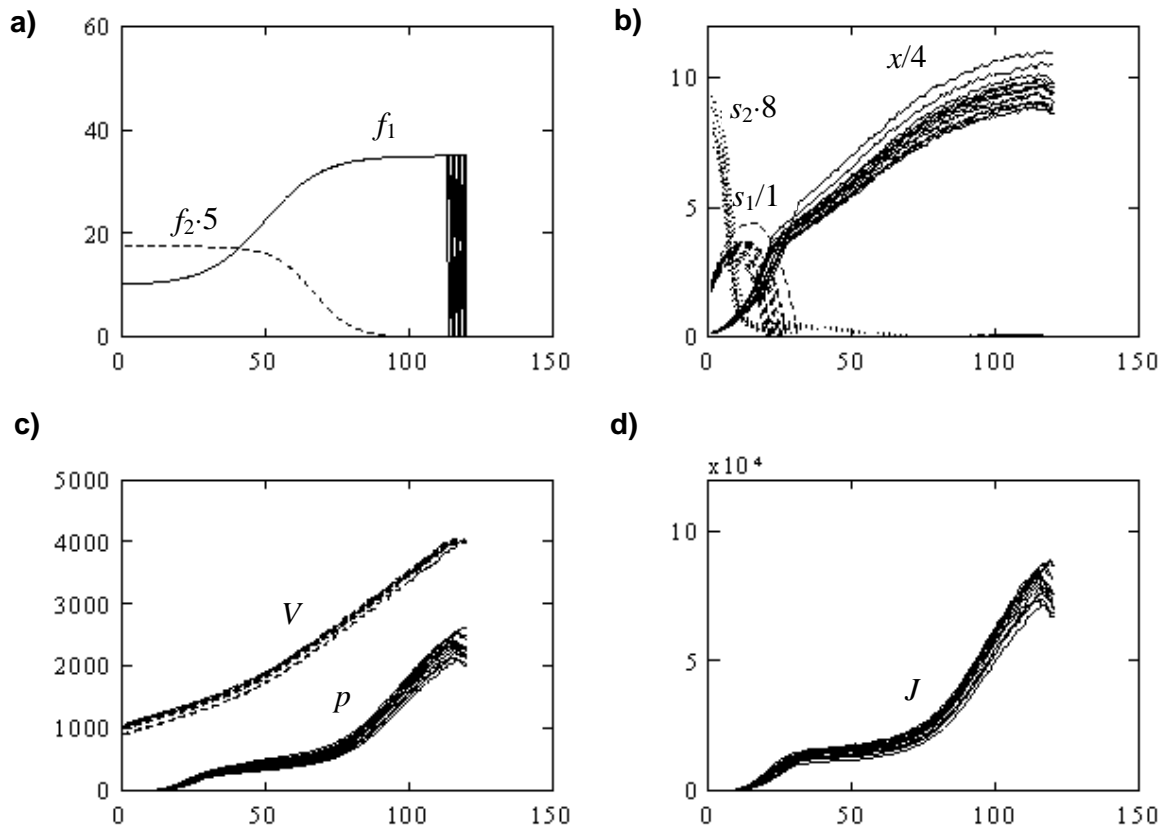
Protsessi kestuse määramisel võtame aluseks nominaalsetes juhtimisprofiilides kasutatud 120 ajaühikut, seega  $t_f = 120$ . Kuna praktiliselt tuleb meil alustada tühjal kohalt - mudeliga kaasasolev informatsioon on tühine - siis aitab taoline ajaline piirang meil oma jõupingutusi kontsentreerida pisut konkreetsemale ülesandele - saavutada etteantud aja jooksul maksimaalne võimalik kogus toodangut.

Tähelepanu tuleb pöörata ka asjaolule, et mudeli tinglik ajaühik kujutab endast tegelikult üsna pikka ajavahemikku. Sellega on aeg diskreeditud üsna suure diskreetimisintervalliga ja mõõdetavate suuruste väärtusi on võimalik saada üksnes diskreeditud ajahetkedel (seega mitte

jooksvalt). Reaalses elus on fermentatsioonikeskkonna komponentide kontsentratsioonide määramine keerukas ja aeganõudev toiming. Sellise diskreetimisintervalliga simuleeritakse ajahulka, mis kulub kontsentratsioonide mõõtmiseks. Samas on see veel üks asjaolu, millega püütakse protsessi mudelit lähendada reaalsele protsessile. Protsessi juhtimise muudab hõredalt diskreeditud aeg muidugi keerukamaks, eriti kui kasutame juhtimisel tagasisidet.

Nominaalsed juhtimisprofiilid ja 15 nendega läbiviidud fermentatsiooni andmed (joonis 6-2) on esmaseks allikaks informatsiooni hankimisel protsessi kohta.

joonis 6-2



Kõige esimene ja tähtsam järeldus, mille võime teha, on järgmine: jooniselt selgub, et nominaalseid juhtimisprofiile rakendades täitub fermentaator vahetult enne protsessi lõppu ( $V$  väärtus joonisel 6-2c). Mida varem see täitumine aset leiab, seda halvem on resultaat, mida iseloomustab protsessi kvaliteedinäitaja joonisel 6-2d. Teisest küljest pole soovitatav ka pooltühi fermentaator, sest produkti lõplik kogus ( $p \cdot V$ ) sõltub lisaks produkti keskmisest kogusest mahuühikus (kontsentratsioon) ka fermentatsioonikeskkonna mahust. Kõik see viib meid järeldusele, et ideaalsel juhul täitub fermentaator täpselt protsessi lõpphetkeks ja see oleks üks eesmärkidest, mida protsessi juhtimisel saavutada.

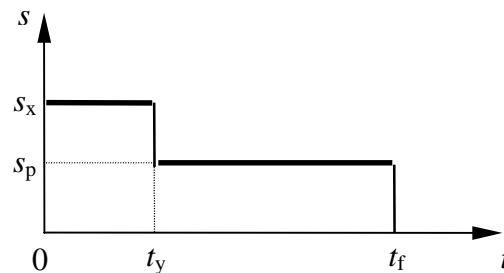
Nominaalsete juhtimisprofiilide üksikasjaliku analüüsiga on võimalik jõuda ka kahtlemata muudele olulistele järeldustele, ent siinkohal me üheainsa ehkki väga olulise järeldusega piirdume. Mittebiotehnoloog võib protsessi sügavuti tundmata jõuda ka ekslikele järeldustele.

## 7. Eksperimentide planeerimine ja juhtimisalgoritmi väljatöötlus

Jaotises 5.3.3 jõudsime järeldusele, et poolperioodilise fermentatsiooniprotsessi optimaalseks juhtimiseks tuleb identifitseerida sõltuvused  $x = f(s)$  ja  $q_p = g(s)$  ning nende põhjal töötada välja optimaalsed juhtimisprofiilid. Toodud olid ka erinevatele  $x = f(s)$  ja  $q_p = g(s)$  iseloomudele vastavad soovitatavad üldised juhtimisprofiilid.

Sellele tuginedes püstitasime hüpoteesi, et toitesegu fermentaatorisse juhtimise optimaalne profiil on selline, mille korral tagatakse protsessi esimeses järgus  $[0, t_y]$  maksimaalne biomassi juurdekasv ja teises järgus  $[t_y, t_f]$  maksimaalne toodangu juurdekasv.

joonis 7-1



Substraadi  $S$  kontsentratsioonid  $s_x$  ja  $s_p$ , mille puhul vastavalt biomassi juurdekasv ja toodangu juurdekasv on maksimaalsed, on leitavad identifitseeritud sõltuvustest  $x = f(s)$  ja  $q_p = g(s)$ .

Seega nihkub juhtimisülesanne optimaalse juhtimisprofiili väljaarvutamisel protsessi järgule vastava optimaalse substraadi kontsentratsiooni määramisele ja nende tagamisele fermentaatoris protsessi erinevates järkudes. Viimasel eesmärgil võib kasutada näiteks PI-regulaatoreid.

Lahtiseks jääb siin see, mil moel teha kindlaks ajamoment  $t_y$ , millal toimub ümberlülitumine biomassi tootmiselt toodangu tootmisele - teisisõnu kuidas identifitseerida protsessi järgud.

Ometi leiame, et selle ajamomendi määramine on juba oluliselt lihtsam ülesanne ja leitav katseliselt mõne vähemõnnestunud fermentatsiooni hinnaga.

Sõltuvuste  $x = f(s)$  ja  $q_p = g(s)$  identifitseerimise eelduseks on üldjuhul protsessi matemaatilise mudeli olemasolu. Kuid kui selline puudub?

Käesolevas jaotises vaatleme mil moel on teostatav sõltuvuste  $x = f(s)$  ja  $q_p = f(s)$  identifitseerimine ilma matemaatilise mudelita (hägusa loogika või muude vahenditega) ning kuivõrd on antud meetodika üldse rakendatav meie konkreetse protsessi puhul.

## 7.1 Eksperimentide planeerimine

### 7.1.1 Protsessi lühiiseloostus

Võrreldes fermentatsiooni optimaalse juhtimise teoorias kirjeldatud tüüpjuhtumiga teeb peatükis 6 tutvustatud protsessi juhtimise tublisti keerulisemaks asjaolu, et protsessi kulgu mõjutatakse mitte ühe vaid kahe substraadiga ( $s_1$  ja  $s_2$ ).

Teoorias oli üheks meetodi eelduseks, et kasutusel on üks substraat. Proovime sellest hoolimata lahendada juhtimisülesannet eeltoodud järeldustele tuginedes.

Seega tuleks esmalt identifitseerida juba keerukamad sõltuvused  $x = f(s_1, s_2)$  ja  $q_p = g_1(s_1, s_2)$  ning siis nende põhjal leida substraatide  $s_1$  ja  $s_2$  optimaalsed kontsentratsioonid  $s_{x1}$ ,  $s_{x2}$ ,  $s_{p1}$  ja  $s_{p2}$ .

Matemaatilise mudeli puudumisel tuleb selline või temaga samaväärne hägus mudel mingi meetodi abil tuletada, lähtudes objektiga tehtavatest eksperimentidest laekuvatest eksperimendiandmetest. Seega tuleks  $\mu = f(s_1, s_2)$  ja  $q_p = g(s_1, s_2)$  identifitseerimiseks viia eeskätt läbi teatav arv eksperimente, millest vastavad seosed oleksid piisava täpsusega määratavad. Missugune eksperimentide kogum tagab nende seoste piisava täpsusega määratuse? See on küsimus, millele pole vastust niisama lihtne leida. Lisaks puudub ka absoluutne kriteerium, millega hinnata juba määratud  $\mu = f(s_1, s_2)$  ja  $q_p = g_1(s_1, s_2)$  õigsust. Seega võib kohe alguses viidata kahele tõsisele lahendust vajavale probleemile.

1. seoste  $\mu = f(s_1, s_2)$  ja  $q_p = g(s_1, s_2)$  identifitseerimiseks kasutatavate eksperimendiandmete valik
2. identifitseeritud seoste  $\mu = f(s_1, s_2)$  ja  $q_p = g(s_1, s_2)$  kontroll

Vaatleme fermentatsiooniekspriimentide läbiviimise korda. Meil on võimalik otsitavate sõltuvuste määramiseks läbi viia  $\eta$  fermentatsiooni, varieerides sisendeid  $f_1$  ja  $f_2$  ning registreerides väljundite  $x$ ,  $p$ ,  $s_1$ ,  $s_2$ ,  $V$  väärtusi.

Iga üksik fermentatsiooniprotsess kestab  $n$  ajaühikut ning selle aja jooksul registreeritakse  $n$  eksperimendiandmete lugemit (tabel 7-1) :

**tabel 7-1**

$$f_1(k) \quad f_2(k) \quad x(k) \quad s_1(k) \quad s_2(k) \quad p(k) \quad V(k),$$

kus  $k = 1, 2, \dots, n$ .

Seega koosneb identifitseerimiseks kasutatavate eksperimendiandmete kogum summaarselt  $n \cdot \eta$  lugemist.

Suurused  $\mu(k)$  ja  $q_p(k)$  on katseandmete põhjal arvutatavad järgmiselt:

$$\mu(k) = \frac{x(k) - x(k-1)}{x(k)} = f(x(k), x(k-1))$$

**avaldis 7-1**

$$q_p(k) = \frac{p(k) - p(k-1)}{x(k)} = g(p(k), p(k-1), x(k))$$

**avaldis 7-2**

kus  $k = 1, 2, \dots, 120$  ja algkontsentratsioonid  $p(0)$ ,  $x(0)$  on meil teada.

Diskussiooni objektiks võib seejuures olla murru nimetaja - biomassi kontsentratsioon, millele biomassi ja toodangu juurdekasvud ühe ajaühiku jooksul taandatakse. Kas õige oleks selleks valida kontsentratsioon ajaühiku algul  $x(k-1)$ , ajaühiku lõpul  $x(k)$  või kogunisti nende kahe suuruse keskmine  $\frac{x(k) + x(k-1)}{2}$ ?

Arvame, et pole oluline missugune konkreetset neist variantidest on valitud - oluline on vaid, et järgitaks ühte kindlat meetodikat kõikide mõõtmiste jaoks, et tulemused oleksid võrreldavad - antud juhul oleme murru nimetajas fikseerinud  $x(k)$ .

Igal ajahetkel  $k$  on niisiis võimalik mõõdetavate suuruste põhjal arvutada üheselt vastavad  $\mu(k)$  ja  $q_p(k)$ . Meie eesmärgiks on leida nende sõltuvus suurustest  $s_1$  ja  $s_2$ .

Kui meil õnnestuks soovikohaselt tekitada fermentaatoris substraatide  $s_1$  ja  $s_2$  kontsentratsioonid  $s_{1i}$  ja  $s_{2j}$ , siis ei pruugiks teha muud, kui mõõta neile vastavad  $\mu$  ja  $q_p$  väärtused  $\mu_{ij}$  ja  $q_{pij}$  (tabel 7-2).

**tabel 7-2**

	$s_{11}$	...	$s_{1i}$	...	$s_{1s}$
$s_{21}$	$\mu_{11},$ $q_{p11}$	...	$\mu_{i1}, q_{pi1}$	...	$\mu_{s1}, q_{ps1}$
...	...	...	...	...	...
$s_{2j}$	$\mu_{1j}, q_{pj1}$	...	$\mu_{ij}, q_{pij}$	...	$\mu_{sj}, q_{psj}$
...	...	...	...	...	...
$s_{2t}$	$\mu_{1t}, q_{pt1}$	...	$\mu_{it}, q_{pit}$	...	$\mu_{st}, q_{pst}$

Selle tabeliga oleks määratud sõltuvused  $\mu = f(s_1, s_2)$  ja  $q_p = g(s_1, s_2)$  substraatide kontsentratsioonide muutumispiirkondades  $[s_{11}, s_{1s}]$  ja  $[s_{11}, s_{1t}]$ . Mida väiksemad on  $s_1$  ja  $s_2$  kvantimisintervallid, seda täpsemalt on seosed identifitseeritud. Tulemuste usaldatavust tõestab see, kui  $\mu_{ij}$  ja  $q_{pij}$  pole mitte üksikmõõtmised vaid kujutavad endast mitmete mõõtmiste keskmisi.

Et substraatide kontsentratsioonid kuuluvad katsete käigus mõõdetavate, mitte varieeritavate suuruste hulka (tabel 7-1), siis meil puudub otsene võimalus neid soovikohaste väärtusteni viia, mis oleks kõige otsem tee meid huvitavate sõltuvuste identifitseerimisele.

Otseselt me seega  $s_1$  ja  $s_2$  väärtustele ligi ei pääse. See-eest on nad aga kaudselt mõjutatavad sisenditega  $f_1$  ja  $f_2$ . Mida rohkem me fermentaatorisse substraate viime, seda suurem on eeldatavasti nende kontsentratsioon seal. Kuid eksisteerivad seosed  $s_1 = \xi(f_1)$  ja  $s_2 = \psi(f_2)$  on kõike muud kui lineaarsed ning pole antud võrranditega täielikult defineeritud, kuivõrd suuruste  $s_1$  ja  $s_2$  väärtused sõltuvad lisaks ka teistest suurustest - eelkõige biomassi kontsentratsioonist  $x(k)$ . Sellest hoolimata pole meil muud reaalselt võimalust mõjutada protsessi kulgu kuidagi teistmoodi kui sisenditega  $f_1$  ja  $f_2$ .

## 7.1.2 Substraatide kontsentratsioonid fermentaatoris

Oleme jõudnud püstitada eesmärgi - viia läbi teatav arv eksperimente varieerides sisendsuurusi  $f_1$  ja  $f_2$  nii, et resultaadina saadav eksperimendiandmete kogum sisaldaks substraatide kontsentratsioonide  $s_{1i}$  ja  $s_{2j}$  ühtlaselt ja väikese intervalliga jaotunud väärtusi kogu nende muutumispiirkonna ulatuses. Seda vajaksime sõltuvuste  $\mu = f(s_1, s_2)$  ja  $q_p = g(s_1, s_2)$  täielikuks ja täpseks identifitseerimiseks.

Kas selline täielik informatsioon on aga üldse meile vajalik? Mida täpsemalt üritame sõltuvusi määrata, seda suurem peab olema eksperimente kogum, seda rohkem vajame aega ja kulutame resursse. Võime jõuda olukorda, kus suudame küll soovitud seosed absoluutse täpsusega identifitseerida ent selle hind kujuneb liiga kõrgeks.

Meie eesmärgiks on luua protsessi mudel, kuid see mudel peab andma meile üksnes informatsiooni protsessi juhtimisalgoritmi väljatöötamiseks. Mudeli kvaliteedi ainsaks kriteeriumiks sellest vaatevinklist on üksnes juhtimisalgoritmi toimimise kvaliteet. Viimast saame paraku hinnata üksnes tagantjärele. Küll aga suudame protsessi eelneva avalüüsi ning isegi eksperimentide läbiviimise käigus selgitada välja olulisemad substraatide kontsentratsioonid, millele oma tähelepanu koondada.

Mis puutub suuruste  $f_1$  ja  $f_2$  varieerimisse, ehk teisisõnu eksperimentide juhtimisprofiilide määramisse, siis vähemalt teoreetiliselt võivad need juhtimisprofiilid olla mistahes funktsioonid. Praktikas on soovitud piirduda üksnes konstantsete juhtimisprofiilidega, vastasel juhul seame endale liiga keerulise ülesande, mille lahendamine on seotud väga suurte raskustega. Otsustamisega konstantsete juhtimisprofiilide kasuks taganeme me esialgselt seatud eesmärkidest, sest on selge, et üksnes selliste profiilidega opereerides ei suuda me tekitada fermentaatoris kõikvõimalikke substraatide kontsentratsioone, kuid võib loota, et see ei olegi esmatähtis.

Esimeses lähenduses on meil teada suuruste  $f_1$  ja  $f_2$  muutumiskiirused  $[0, f_{1\max}]$ ,  $[0, f_{2\max}]$ . Konkreetse protsessi puhul  $f_{1\max} = f_{2\max} = 50$ .

Konstantsete juhtimisprofiilide korral valime kvantarvud  $m$  ja  $n$ , millega vastavalt  $f_1$  ja  $f_2$  muutumiskiiruseid kvantida. Nende kvantarvude valikuga on ühtlasi määratud nii sooritata eksperimentide arv  $(m + 1)(n + 1)$  kui vastavad juhtimisprofiilid (tabel 7-3).

**tabel 7-3**

<i>eksp. nr.</i>	$f_1$	$f_2$	<i>eksp. nr.</i>	$f_1$	$f_2$
1	0	0	$2m + 1$	$\frac{f_{1\max}}{m}$	$(n - 1) \cdot \frac{f_{2\max}}{n}$
2	0	$\frac{f_{2\max}}{n}$	$2m + 2$	$\frac{f_{1\max}}{m}$	$f_{2\max}$
3	0	$2 \cdot \frac{f_{2\max}}{n}$	.....	.....	.....
.....	.....	.....	$n(m + 1) + 1$	$f_{1\max}$	0
$m$	0	$(n - 1) \cdot \frac{f_{2\max}}{n}$	$n(m + 1) + 2$	$f_{1\max}$	$\frac{f_{2\max}}{n}$
$m + 1$	0	$f_{2\max}$	$n(m + 1) + 3$	$f_{1\max}$	$2 \cdot \frac{f_{2\max}}{n}$
$m + 2$	$\frac{f_{1\max}}{m}$	0	.....	.....	.....
$m + 3$	$\frac{f_{1\max}}{m}$	$\frac{f_{2\max}}{n}$	$n(m + 1) + m$	$f_{1\max}$	$(n - 1) \cdot \frac{f_{2\max}}{n}$
$m + 4$	$\frac{f_{1\max}}{m}$	$2 \cdot \frac{f_{2\max}}{n}$	$(m + 1)(n + 1)$	$f_{1\max}$	$f_{2\max}$
.....	.....	.....			

Kui oleme meie konkreetse protsessi puhul valinud jargmised kvantarvude vaartused  $m = n = 50$ , mis annab juhtimisprofiilide kvantimisintervalliks parajasti 1, saame uhes sellega kohustuse sooritada 2601 eksperimenti. Arv tundub hirmuaratavana ja seda ta ka on. Esmase tahtsusega on leida tee, kuidas seda arvu vahendada. Koige lihtsam on vahendada kvantarve, kuid see vahendab ka eksperimendiandmete kasutuskoiblikkust. Ent on ka teisi teid. Nende leidmiseks analuisigem pisut konkreetset fermentatsiooniprotsessi.

Teadaolevalt katkestab fermentaatori taitumine automaatselt igasuguse juhtimise - protsessi kulgu pole enam voimalik mojutada. Fermentaator taitub seda kiiremini, mida rohkem me substraate fermentaatorisse pumpame. Et samal ajal on ebaotstarbekas ka eksperimendi labiviimine pooltuhja fermentaatoriga, eksisteerib seega teatud optimaalne substraatide kogus, mis tuleb protsessi kestel fermentaatorisse viia kindlustamaks maksimaalne resultaat. See substraatide kogus peab olema selline, mis taidaks protsessi lopp hetkeks kogu olemasoleva vaba ruumi fermentaatoris.

Ruumalauhikutes avaldub optimaalne toitainete kogus  $V_{opt}$  jargmiselt:

$$V_{opt} = V_{max} - V_{alg}, \quad \text{avaldis 7-3}$$

kus  $V_{max}$  on fermentaatori maht ning  $V_{alg}$  fermentatsioonikeskkonna maht protsessi alghetkel kusjuures:

$$V_{opt} = \sum_{i=1}^{120} (f_{1i} + f_{2i}), \quad \text{avaldis 7-4}$$

kus  $f_{1i}$  ja  $f_{2i}$  kujutavad endast vastavalt 1. ja 2. substraadi fermentaatorisse viimise kiiruseid  $i$ -ndal ajahetkel.

Avaldis 7-4 on uhes tahtsamaks piiranguks fermentatsiooni juhtimisel, mida saame kasutada ka soltuvuste identifitseerimisel kasutatavate eksperimentide hulga piiramiseks, kuna ennekoike huvitavad meid siiski vaid sellised fermentatsiooniandmed, mis oleksid vastavuses fermentatsiooni edukal labiviimisel figureerivatega.

Kui toitainete fermentaatorisse viimise kiirused on konstantsed protsessi valtel, ( $f_1 = \text{const}$ ,  $f_2 = \text{const}$ ), siis tulevad kone alla uksnes sellised kiirused, mille puhul on taidatud tingimus:

$$t_f (f_1 + f_2) = V_{opt}, \quad \text{avaldis 7-5}$$

kus  $t_f$  on protsessi kestus (120).

Meie konkreetse fermentatsiooni puhul saame umbkaudselt valja arvutada suuruse  $V_{opt}$ . Alghetkel jaab fermentatsioonikeskkonna maht  $V_{alg}$  ligikaudselt vahemikku [800, 1200] ning  $V_{max} = 4000$ , seega  $V_{opt} = [2800, 3200]$ . Vastavate juhtimete summa  $f_{1opt} + f_{2opt}$  jaab vahemikku [23, 27].

Avaldist 7-4 piiranguna kasutades ja eelnevaid kvantimisarve aluseks vottes saame, et isegi juhul kui  $V_{opt} = 3200$  (maksimaalne voimalik), vaheneb sooritatavate eksperimentide arv 28-ni. Minimaalse voimaliku  $V_{opt}$  korral on eksperimentide arv 23. Tegelik eksperimentide arv jaab siis kusagile nende kahe arvu vahele (kujuneb valja eksperimente sooritades). See on oluline voit ja mitte loplik. Eksperimentide arvu vahendamise edasine vahendamine toimub jallegi protsessi analuisides.

Paneme veelkord kirja faktid, mida me teame fermentatsiooniprotsessidest üldse ja sellest konkreetsest protsessist, mida juhtida üritame:

- a) *protsessi algul tuleb biomassi kontsentratsioon võimalikult kiiresti viia tasemele, mis võimaldab toodangu suurt juurdekasvu.*
- b) *kui biomassi kontsentratsioon on sellisele tasemele viidud pole tema edasine suurendamine optimaalsuse printsiibist lähtudes oluline.*
- c) *Toodangu suure juurdekasvu teiseks eelduseks on selleks parima substraatide kontsentratsioonide vahekorra tagamine fermentaatoris*
- d) *protsess on kõige edukam siis, kui fermentaator täitub alles fermentatsiooni lõpphetkeks*
- e) *substraatide liiga suurel või väikesel hulgal fermentaatorisse viimine ei anna häid tulemusi. (see on seotud faktiga d, fermentaator kas täitub liiga kiiresti või jääb pooltühjaks!)*
- f) *fermentatsioonikeskkonda ainult substraadi 2 viimine ei anna mingit toodangut, üksnes substraadi 1 kasutamine annab vähest toodangut*

Ekspereimendite planeerimisel saame siit kasutada fakti f, mis võimaldab meil jätta ära sellised eksperimendid, kus kas  $f_1$  või  $f_2$  või mõlemad on võrdsed nulliga. Sõelale jäänud eksperimendi hulgast võime seega veel eemaldada 2.

Samuti on otstarbekas on kehtestada fermentatsioonikeskkonna komponentidele teatud piirväärtused. Nendest suuremad väärtused näitavad, et tegu on ekstreemsetele resultaatile viiva eksperimendiga, mille väärtus juhtimisalgoritmi väljatöötamise vaatevinklist on küsitav. Sama kehtib eksperimendite kohta, kus  $f_1$  ja  $f_2$  väärtused üksteisest palju erinevad. Neid piiranguid saame kehtestada eksperimente läbi viies.

Ekspereimendite läbiviimise lõplikuks tulemuseks on 120-lõplik\_eksperimentide\_arv lugemit kujul:

$$f_1(k) \quad f_2(k) \quad x(k) \quad s_1(k) \quad s_2(k) \quad p(k) \quad V(k),$$

Iga üksiku eksperimendi 120 lugemit salvestatakse eksperimendi lõppedes eraldi faili. Kõigi eksperimendite tulemused moodutavad failiseeria (failinimi1.dat, failinimi2.dat ... failinimiη.dat). Saadud failiseeriat kasutame peale täiendavat kontrollimist (ilmselt ebaõnnestunud fermentatsioonide andmete elimineerimist) edasises identifitseerimisprotsessis.

Väljakujunenud tootmisrezhiimi olemasolu (nominaalsed juhtimisprofiilid antud juhul), viib meid aga ideele kasutada identifitseerimiseks ainult nendega esilekutsutud andmekogumit. Ekspereimendite sooritamise mõte on koguda andmeid sõltuvuste  $\mu = f(s_1, s_2)$  ja  $q_p = g(s_1, s_2)$  identifitseerimiseks, kuid peaaesjalikult huvitavad meid ainult  $\mu$  ja  $q_p$  maksimaalsed väärtused ja et nominaalsete juhtimisprofiilide näol on olemas seni kõige soodsamad tootmisrezhiimid, on nende kasutamine identifitseerimisel küllaltki loogiline.

Järgnevalt vaatleme, kuidas peatükis 3 tutvustatud modelleerimis-identifitseerimis-meetodeid kasutada fermentatsiooniprotsessi juhtimisel.

## 7.2 Identifitseerimismeetodid

### 7.2.1 Otsemeetod

Eksperimentide sooritamisel kogutud andmete näol on meil olemas nii substraatide kontsentratsioonide  $s_1(k)$  ja  $s_2(k)$  lugemid, kui võimalus arvutada välja vastavate  $\mu(k)$  ja  $q_p(k)$  väärtused. Kuivõrd suurused  $\mu$  ja  $q_p$  on taandatud biomassi kontsentratsioonile - peaks vähemalt teoreetiliselt olenemata protsessi staadiumist (teiste muutujate kontsentratsioonist) ühtede ja samade  $s_1$  ja  $s_2$  väärtuste korral olema  $\mu$  ja  $q_p$  väärtused alati ühed ja samad. Põhiline probleem seisneb nüüd selles, et kuivõrd  $s_1$  ja  $s_2$  väärtuste jaotumist on väga raske reguleerida (tegelikult me ei üritagi seda), jaotuvad eksperimenditulemused väga ebahühtlaselt, seega on mõnes punktis seose määramisel aluseks palju lugemeid teises kohas väga vähe. On üsna ilmne, et paljudes punktides jääbki seos määramata. Võib muidugi eeldada, et kui seos pole eksperimendiandmetest määratav, on vähetõenäoline, et sellises punktis süsteem kunagi viibibki või et sellises punktis süsteemi viibimine oleks soovitatav. Omaette küsimuseks on siinjuures sellise määramise täpsus.

Arvutame eksperimendiandmete kogumi

$$f_1(k) \quad f_2(k) \quad x(k) \quad s_1(k) \quad s_2(k) \quad p(k) \quad V(k)$$

põhjal välja  $\mu(k)$  ja  $q_p(k)$  väärtused ja esitame tulemused kujul:

$$s_1(k) \quad s_2(k) \quad \mu(k) \quad q_p(k)$$

Kuidas analüüsida saadut ja tuletada sõltuvusi  $\mu = f(s_1, s_2)$  ja  $q_p = g(s_1, s_2)$ ? Et tegu on kolmemõõtmeliste süsteemidega, osutuks kahemõõtmelise süsteemi puhul oma kohta täitev graafiline aproksimeerimine üsnagi keerukaks.

Kui sõltuvused õnnestuks kuidagi identifitseerida, saaksime selgitada välja biomassi ja toodangu juurdekasvuks optimaalsed kontsentratsioonid ja rakendada neid protsessi juhtimisel. Edasine probleem seisneb sobiva ümberlüümisemomendi määramises. Siin võib lihtsamal juhul olla kriteeriumiks mingi biomassi kontsentratsiooni väärtus.

Mitmel põhjusel aga arvame et sellise identifitseerimismeetodi praktiline väärtus on väike.

## 7.2.2 Identifitseerimine närvivõrguga

Kui meil õnnestuks identifitseerida protsessi mudel (kasvõi osaliselt), siis oleks sealt ehk võimalik tuletada meid huvitavad sõltuvused, kusjuures mudeli adekvaatsust on võimalik hinnata treeninguandmete järgimise kvaliteedi järgi.

Jättes kõrvale süsteemide identifitseerimise klassikalised (analüütilised) meetodid, kuna nende kasutamiseks on tegu keeruka süsteemiga, jõuame tehisintellekti meetodite juurde.

Esmalt vaatleme närvivõrgu võimalusi antud olukorras.

### 7.2.2.1 Aparatuuri kirjeldus

Peatükis 3 vaatlesime mitmeid hägusaid modellerimismeetodeid, millest lähemalt käsitleme nüüd sellist, kus modelleerimise saaduseks on Sugeno hägus mudel, mille iga reeglit võib käsitleda kui närvivõrgu üht neuronit. See annab meile võimaluse kasutada sellise mudeli õppimisvõimet protsessi mudeli tuletamiseks.

Vastavalt treenimisandmetega etteantud väljundi ja mudeli poolt genereeritava väljundi erinevusele modifitseeritakse nii sisendite liikmesfunktsioonide (Gaussi funktsioonid) parameetreid kui väljundi defineerivate sisendite lineaarkombinatsioonide parameetreid. Siin on meil võimalus kasutada Matlabi loojate poolt tehtud adaptiivse närvihägusat mudelit realiseerivat funktsiooni - `anfis.m`.

Funktsiooni `anfis.m` kasutatakse kujul:

```
[hägus_mudel, viga] = anfis(eksperimendiandmed, eelmudel, epohhide_arv, lubatud_viga)
```

sisendmuutuja `eksperimendiandmed` kujutab endast maatriksit, mille iga rida avaldub kujul:

```
[ sisendi_1_lugem; sisendi_2_lugem; ..... väljundi_lugem; ]
```

`elmudel` kujutab endast eelnevat defineeritud hägusat mudeli struktuuri (viimane on määratud sisendite arvu ja muutumispiirkondade, nende liikmesfunktsioonide arvu ja jaotusega, mida `anfis` kasutab shabloonina lõpliku mudeli `hägus_mudel` genereerimisel.

Sellise shablooni koostamine pole kohustuslik, `anfis` on võimeline genereerima sellise ka ise kasutades treeninguandmeid, millest leitakse muutujate muutumispiirkonnad ja kasutaja poolt etteantud või vaikeväärtuslikku (default) sisendite liikmesfunktsioonide arvu. Genereeritava mudeli reeglite arv tuleneb liikmesfunktsioonide arvust, reeglite järelduspoole parameetrite arv sisendite arvust.

Muutuja `epohhide_arv` määrab treenimisepohhide arvu, `lubatud_viga` on ruutkeskmise vea lävi millest madalamad väärtused on kasutaja arvates piisavad treenimise lõpetamiseks.

Pärast vajalike andmete edastamist `anfis`ile ja viimase käivitamist kestab treenimine seni kuni kas saab läbi tehtud etteantud kogus epohhe või osutub mudeli ruutkeskmise viga väiksemast mudelile lubatud veast. Väljundmuutuja `viga` on ühtlasi esmaseks näitajaks mis võimaldab hinnata treenimise kvaliteeti.

Tänu funktsiooni `anfis.m` olemasolule on treenimine automatiseeritud ja kiire, samas on kasutaja võimalused üsnagi piiratud.

Funktsiooni `anfis.m` kasutamiseks tuleb eksperimendiandmete failiseeria teisendada sobilikule kujule. Esmalt tuleb eksperimendiandmete failiseeria teisendada funktsiooniga `anfis.m` genereeritava hägusa mudeli sisendeid ja väljundeid sisaldavaks failiseeriaks, teiseks liita saadud failiseeria kokku üheks failiks.

Vaatleme nüüd, milliste vahenditega õnnestuks meil modelleerimise kvaliteeti mõjutada.

Primaarseks vahendiks mudeli täpsuse tõstmisel on sisendi liikmesfunktsioonide arvu õige valik. Mingi konkreetse eksperimendiandmete kogumi korral eksisteerib teatud optimaalne sisendite hägus lahterdus, mille korral saavutame parimad tulemused. Paraku on mitmemuutuja süsteemide puhul pea võimatu sellist lahterdust apriorset tuletada. Seega on mõistlik piirduda üksnes liikmesfunktsioonide arvuga manipuleerimisega. Liikmesfunktsioonide arvu suurendamine viib aga treenimisaja olulisele pikenemisele.

Kahtlemata on mudel seda täpsem mida kauem treenimine on kestnud, kuid viga väheneb aeglaselt ning tihtipeale ei anna alates teatud punktist treenimine mitte mingisugust efekti. Seniomandatud kogemuste põhjal võib väita, et treenimisepohhide arvu tõstmine pole vahend mudeli kvaliteedi kardinaalseks tõstmiseks.

Ning loomulikult on treenimise kvaliteet kõige otsemas seoses eksperimendiandmete hulga ja olemusega. Kui andmeid on vähe, on põhimõtteliselt lihtne jõuda meid rahuldavale tulemusele. Seda ka juhul kui andmed ei räägi üksteisele vastu. Seetõttu on kasulik enne treenimisele asumist kontrollida treeninguandmeid sellest seisukohast lähtudes. Närvivõrku võib võrrelda inimesega. Kui andmete vahel eksisteerib seos, siis on ta võimeline selle leidma, kui ei, siis pole kasu ka kõige pikemast õpiajast. Seega osutub treenimise kvaliteedi seisukohast kõige olulisemaks treeninguandmete olemus.

### 7.2.2.2 Strateegiad

Strateegia all on mõistetud siin eelkõige seda, millise struktuuriga mudel sõltuvuste  $q_p = f(s_1, s_2)$ ,  $\mu = f(s_1, s_2)$  identifitseerimiseks genereeritakse, s.t. missugused sisendid ja väljundid on mudelil mida genereerime.

#### 1) Otsene strateegia

Identifitseerime sõltuvused mudelitena  $q_p = f(s_1, s_2)$ ,  $\mu = f(s_1, s_2)$ . Treenimisandmed saadakse eksperimendiandmete, mis on kujul:

$$f_1(k) \quad f_2(k) \quad x(k) \quad s_1(k) \quad s_2(k) \quad p(k) \quad V(k)$$

töötlemisel kasutades avaldisi 7-1 ja 7-2, funktsioonile anfis.m söödavad eksperimendiandmete maatriksid on kujul:

$$s_1(k) \quad s_2(k) \quad \mu(k) \quad \text{ja} \quad s_1(k) \quad s_2(k) \quad q_p(k)$$

Mida annab meile selliste mudelite identifitseerimine kui  $\mu$  ja  $q_p$  väärtused arvutame ju me nii või teisiti välja? Sellised mudelid teevad meie eest ära aproksimeerimise ja loodetavasti täpsemalt kui ükskõik milline graafiline meetod.

Sellise struktuuriga mudelite puhul on positiivseks omaduseks muutujate väike arv, mis lubab defineerida neile suurema hulga liikmesfunktsioone ja siiski saada suhteliselt lühikesi treenimisaegu. Negatiivseks küljeks on see, et sõltuvused  $q_p = f(s_1, s_2)$  ja  $\mu = f(s_1, s_2)$  kujutavad endast küllaltki raskesti identifitseeritavaid funktsioone. Suured identifitseerimisvead pole haruldased. Seega tuleb resultaatidesse suhtuda teatud reservatsiooniga. Tulemuste hindamisel tuleb ettevaatlikumalt suhtuda seostesse mis kehtivad piirkondade kohta, millel ei olnud katet treeninguandmete näol, kuna need seosed on närvivõrk loonud ise ja seetõttu ei pruugi need tegelikkusele vastata.

#### 2) Kaudne strateegia

Loome eksperimendiandmete põhjal mudelid:

$$p(k) = f(p(k-1), x(k-1), s_1(k-1), s_2(k-1))$$
$$x(k) = f(x(k-1), s_1(k-1), s_2(k-1)).$$

Siin on  $q_p$  ja  $\mu$  arvutatavad mudelite sisend- ja väljundmuutujatest.

Sellist olekumuutuja-mudelit on palju lihtsam identifitseerida. Küsimusele - miks? - saame vastuse kui meenutame et sugeno-mudelite reeglite väljundpool avaldub kujul

$$p(k) = b_0 + b_1p(k-1) + b_2x(k-1) + b_3s_1(k-1) + b_4s_2(k-1)$$

**avaldis 7-6**

ja et  $p(k)$  ja  $p(k-1)$  erinevad üksteisest vähe. Seega on sellise struktuuriga mudelit suhteliselt lihtne treenida. Treenimisvead on väikesed. Ehkki sisendite arv on suur, annab rahuldavat efekti ka väikese ulatusega hägus lahterdamine ja treenimisajad väga suured ei tule.

Mis mõte on sellise mudeli identifitseerimisel? See mudel võimaldab meil kiiresti ja mugavalt arvutada soovitud näitajate väärtused kasutades avaldisi 7-1 ja 7-2 ning nimetajas suurust  $x(k-1)$ .

Nüüsiis varustades mudelid toodangu, biomassi ja substraatide kontsentratsioonidega ajahetkel  $k-1$ , saame me tulemuseks toodangu ja biomassi kontsentratsioonid järgmisel ajahetkel, mis võimaldab arvutada biomassi kogusele taandatud biomassi või toodangu kontsentratsiooni juurdekasvu.

Käesoleva töö raames on närvivõrgule viidatud vaid kui potentsiaalsele identifitseerimisvõimalusele, praktilist rakendust ta siin ei leia, kuna töö maht seda lihtsalt ei võimalda.

### 7.2.3 Identifitseerimine lingvistilise meetodiga

Sugeno tüüpi hägusa mudeli kuju põhjal on inimesel keeruline hinnata tema õigsust ja toimet. Samuti pole närvivõrguna treenitud hägus mudel lõpuni usaldatav, kuna ta loob seosed sisend- ja väljundmuutujate vahel kogu nende muutumispiirkondade kohta treeninguandmete baasil, mis ei pruugi katta kogu muutumispiirkondi. Piltlikult öeldes jätkab treeninguandmetega katmata piirkondades närvivõrk seoste laiendamist, mis kehtivad treeninguandmetega kaetud piirkondades. Teatud juhtudel võivad sel moel tuletatud seosed olla tõesed, kuid meil on alust karta, et enamikel juhtudel ja keerukamate süsteemide korral on sellised seosed "õhust võetud". Seega, mida vähem on treeninguandmeid ja mida vähem hajutatud nad on, seda ebausaldatavam on mudel. Kõik see teeb närvivõrkude kasutamise juhtimisalgoritmi tuletamisel problemaatiliseks.

Jaotises 3.2 väljapakutud shabloonil baseeruvad hägusa modelleerimise meetodid (mida antud peatükis ja edaspidi kontrastiks närvivõrgule nimetatakse lingvistilisteks meetoditeks) võivad olla küll vähem täpsemad ja mitte nii efektsed, kuid see-eest kajastab nende abil tuletatud mudel täpselt treeninguandmeid ning on seetõttu usaldatavam. Jällegi piltlikult väljendades registreerib taoline meetod lihtsalt eksperimendiandmed sellisel moel, mis teeb võimalikuks nii tuletatud struktuuri käsitleda süsteemi mudelina. Lisaks avaldub ta lingvistilisel kujul, mis autori tagasihoidlikul arvamusel on inimesele arusaadavam.

Seetõttu on järgnevalt seda meetodit ka kasutatud fermentatsiooniprotsessi juhtimisalgoritmi tuletamiseks:

#### 7.2.3.1 Meetodi idee

Me identifitseerime protsessile järgmised kaks mudelit:

kui  $x$  on ... ja  $s_1$  on ... ja  $s_2$  on ... siis  $dp$  on ...

**avaldis 7-7**

kus  $x$  - biomassi kontsentratsioon fermentaatoris

$s_1$  - substraadi 1 kontsentratsioon fermentaatoris

$s_2$  - substraadi 2 kontsentratsioon fermentaatoris  
 $dp$  - toodangu kontsentratsiooni juurdekasv fermentaatoris

kui  $x$  on ... ja  $s_1$  on ... ja  $s_2$  on ... siis  $dx$  on ...

**avaldis 7-8**

$x$  - biomassi kontsentratsioon fermentaatoris  
 $s_1$  - substraadi 1 kontsentratsioon fermentaatoris  
 $s_2$  - substraadi 2 kontsentratsioon fermentaatoris  
 $dx$  - biomassi kontsentratsiooni juurdekasv fermentaatoris

Avaldis 7-7 kujutab endast toodangu, avaldis 7-8 biomassi juurdekasvu mudelit.

Muutujate  $x$ ,  $s_1$  ja  $s_2$  hägusad lahterdused peavad mõlemas mudelis olema ühed ja samad.

Sellised reeglid annavad meile tegelikult toodangu ja biomassi kontsentratsioonide muutumise seaduspärasused lingvistilisel kujul. Ühtlasi saab mudeleid invertteerides leida ka tingimused, mille täidetuse puhul võib oodata üht või teistsugust juurdekasvu.

Näiteks kui toodangu juurdekasvu mudel deklareerib oma reeglitega, et toodangu juurdekasv on positiivne ja suur siis kui  $x$  on keskmine,  $s_1$  väike ja  $s_2$  väike, siis tuleb suure toodangu juurdekasvu saamiseks tagada üheaegselt keskmine biomassi ja väikesed  $s_1$  ning  $s_2$  kontsentratsioonid.

Inverteeritud mudelit kasutamegi juhtimisalgoritmi lõplikul väljatöötamisel

### 7.2.3.2 Fermentatsiooni juhtimine

Kuidas me niiviisi saadud reeglite täitmist protsessi juhtimisel tagame?  $s_1$  ning  $s_2$  on juhitavad substraatide fermentaatorisse juhtimise tempode ( $f_1$  ja  $f_2$ ) kaudu, biomassi soovitud kontsentratsiooni saamiseks kasutame samamoodi biomassi juurdekasvu mudeli reegleid.

Seejärel võime resultaatina panna kokku hägusa regulaatori, mis arvutab välja biomassi kontsentratsiooniga määratud protsessi staadiumi jaoks sobivad  $s_1$  ja  $s_2$  kontsentratsioonid.

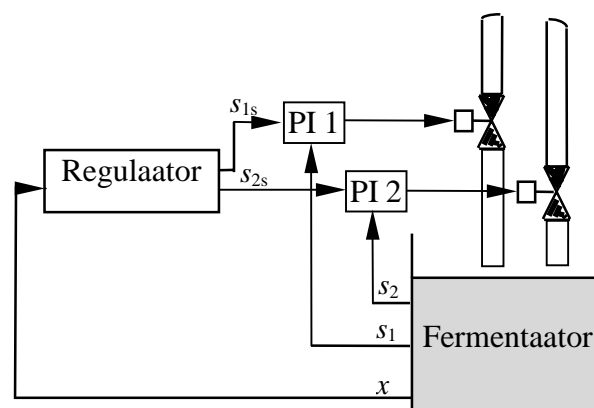
Regulaator koosneb järgmist tüüpi reeglitest:

KUI  $x$  on ... SIIS  $s_1$  peab olema ... ja  $s_2$  peab olema ...

Reeglite arv on määratud muutuja  $x$  hägusa lahterdusega. Kõigi regulaatoriga seotud muutujate hägusad lahterdused võtame biomassi juurdekasvu mudelist.

Nende regulaatorite väljundite väärtusi saame kasutada seadesuurustena PI-tüüpi regulaatoritele, mis tagavad substraatide soovitud kontsentratsioonid fermentaatoris.

**joonis 7-2**



Juhime tähelepanu ka sellele, et nii tuletatud juhtimisalgoritmi puhul kaob ära vajadus eraldi ajamomendi  $t_y$  identifitseerimiseks, kuna see on määratud ära hetkega, mil biomassi on fermentaatoris selline kogus, mille puhul toodangu juurdekasv hoogustub märgatavalt.

Kuna selline biomassi koguse määratlus on hägus, ei lülitu me biomassi tootmiselt toodangu produtseerimisele mitte hüppeliselt vaid sujuvalt.

Tegemist on niisiis hierarhilise juhtimisega - alumine tase (PI regulaatorid) teostab otsest juhtimist ning ülemine tase (hägus regulaator) määrab juhtimise eesmärgid.

Taolise meetodi realisatsiooni edukus ripub ära identifitseeritud mudelite täpsusest ja tegelikkusele vastavusest, viimane omakorda kahest asjaolust

- a) liikmesfunktsioonide determineerimine
- b) eksperimendiandmete olemus

Liikmesfunktsioonid peavad olema häälestatud, s.t. piirkondades, kus väike kontsentratsiooni muutus on suure mõjuga protsessi kulgemisele, peab olema defineeritud suurem hulk liikmesfunktsioone. Samuti on suurema prioriteediga substraatide kontsentratsioonid, s.t nende lahterdus peab olema tihedam. Ülekattumise määr võiks olla 0,5 või selle ümbruses, sest see annab kõige sujuvamad üleminekud. Konkreetse lahterduse kirjutavad ette eksperimendiandmed.

### 7.2.3.3 Meetodi kirjeldus

Identifitseerimiseks kasutame jaotise 3.2 teises pooles tutvustatud hägusmodelleerimise meetodit, mis sobib antud identifitseerimisülesande lahendamiseks. Et näitena oli toodud SISO mudeli identifitseerimisalgoritm, tuleb antud süsteemi juures kasutamiseks kõigepealt seda laiendada MISO mudelitele. Teeme seda.

Antud sisend-väljundhulk  $(x_{1k}, x_{2k}, x_{3k}, \dots, y_k)$ , kus  $x_{1k}, x_{2k}, x_{3k}, \dots, x_{rk}$  on vastavalt ajahetkel  $k$  eksisteerivad sisendite  $U_1, U_2, U_3, \dots, U_r$  väärtused ja  $y_k$  väljundi  $V$  väärtus samal hetkel, rahuldab elementaarreeglit

**KUI**  $U_1$  on  $B_{i1}$  **JA**  $U_2$  on  $B_{i2}$  **JA** ... **JA**  $U_r$  on  $B_{ir}$  **SIIS**  $V$  on  $D_j$

siis, kui kõik tabamismäärad  $\tau_{i1} = B_{i1}(x_{1k})$ ,  $\tau_{i2} = B_{i2}(x_{2k})$ , ...  $\tau_{ir} = B_{ir}(x_{rk})$  ( $i_1 = (1, s_1)$ ,  $i_2 = (1, s_2)$ , ...,  $i_r = (1, s_r)$ ,  $k = (1, K)$ ) ja väljundi  $y_k$  liikmesus reegli järelduspoole hägusas alamfunktsioonis  $\gamma_j = D_j(y_k)$ ,  $j = (1, t)$  omavad mittenulliseid väärtusi.

Defineerime seda rahuldatus astet väljendava sobivuse määra:

$$\delta_{i_1 i_2 \dots i_r j}(k) = \tau_{i_1} \tau_{i_2} \dots \tau_{i_r} \gamma_j,$$

**avaldis 7-9**

kus  $i_1 = (1, s_1), i_2 = (1, s_2), \dots, i_r = (1, s_r), j = (1, t), k = (1, K)$

Elementidest  $\delta_{i_1 \dots i_r j}(k)$  moodustub  $r + 2$  dimensionaalne maatriks  $\Delta$ , mille  $r + 1$  dimensiooni mõõdud on määratud sellele vastava muutuja liikmesfunktsioonide arvudega  $(s_1, s_2, \dots, s_r, t)$  ning üle jääva dimensiooni mõõt on määratud eksperimendiandmete maatriksi ridade arvuga  $K$ . Näiteks kui meil on defineeritud kolm sisendit, mis on lahterdatud vastavalt 2,4 ja 3-ks hägusaks hulgaks ja väljund, mis on jaotatud 3-ks hägusaks hulgaks (nende korrutis annab

tulevase mudeli potentsiaalsete reeglite arvu) ning oleme teinud 6 katset, saame tulemusena 5-mõõtmelise 432-elementilise deltade maatriksi.

Normeeritud sobivusmäär  $v_{i_1 i_2 \dots i_r j}(k)$ , mis saadakse  $\delta_{i_1 i_2 \dots i_r j}(k)$  jagamisel ühe sisendväljundkombinatsiooni kohta tuleva summaarse (kõikide reeglite) sobivusmääraga: tuleb mitmemõõtmelise süsteemi puhul järgmine:

$$v_{i_1 i_2 \dots i_r j}(k) = \frac{\delta_{i_1 i_2 \dots i_r j}(k)}{\sum_{i_1=1}^{s_1} \dots \sum_{i_r=1}^{s_r} \sum_{j=1}^t \delta_{i_1 i_2 \dots i_r j}(k)} \quad \text{avaldis 7-10}$$

Nendest elementidest moodustuva maatriksi  $\mathbf{Y}$  dimensioon on täpselt sama, mis maatriksil  $\Delta$ .

Teeme läbi ka ülejäänud elementaarreegli kaalu tuletamiseks vajalikud sammud:

$$v_{i_1 i_2 \dots i_r j} = \sum_{k=1}^K v_{i_1 i_2 \dots i_r j}(k). \quad \text{avaldis 7-11}$$

Tulemuseks on maatriks  $\mathbf{Y}_\Sigma$ , mille dimensioon on  $s_1 \times s_2 \times \dots \times s_r \times t$  ja mis seetõttu on oluliselt väiksem kui  $\mathbf{Y}$  või  $\Delta$ . Jätkame algoritmi arendamist:

$$v_{i_1 i_2 \dots i_r} = \sum_{j=1}^t v_{i_1 i_2 \dots i_r j}; \quad i_1 = (1, s_1), i_2 = (1, s_2), \dots, i_r = (1, s_r) \quad \text{avaldis 7-12}$$

Ja elementaarreegli kaal  $p_{i_1 i_2 \dots i_r j}$  maatriksis  $\mathbf{P}$  avaldub järgmiselt:

$$p_{i_1 i_2 \dots i_r j} = \frac{v_{i_1 i_2 \dots i_r j}}{v_{i_1 i_2 \dots i_r}}; \quad i_1 = (1, s_1), i_2 = (1, s_2), \dots, i_r = (1, s_r), j = (1, t) \quad \text{avaldis 7-13}$$

$p$ -de maatriks sisaldab elemendi asukohaga defineeritud reeglite kaalud.

Võib tekkida vajadus teisendada kõik need mitmemõõtmelised maatriksid kahemõõtmelisteks, kui programeerimiskeskond sedalaadi piirangu massiividele ette seab. Maatrikseid võib teisendada näiteks nii, et  $s_1 \times s_2 \times \dots \times s_r \times t$ -dimensionaalne maatriks omaks teisendatuna  $s_1 \cdot s_2 \dots s_r \times t$  dimensiooni. Suurema dimensiooniga maatriksid  $\mathbf{Y}$  ja  $\Delta$  on otstarbekas tükeldada  $s_1 \cdot s_2 \dots s_r \times t$  dimensiooniga maatriksiteks  $\mathbf{Y}_k$  ja  $\Delta_k$  ( $k = (1, K)$ ), kus iga selline maatriks kujutaks endast ühele eksperimendiandmete lugemile ( $x_{1k}, x_{2k}, x_{3k}, \dots, y_k$ ) vastavat  $\mathbf{Y}$  või  $\Delta$ . See ei ole eriti keeruline.

See algoritm on realiseeritud MatLab'i keskkonnas kirjutatud programmiga, mis on kirjutatud käesoleva töö autori poolt.

Kuna igale eksperimendiandmete lugemile ( $x_{1k}, x_{2k}, x_{3k}, \dots, y_k$ ) vastab üks maatriks  $\Delta_k$  (ja  $\mathbf{Y}_k$ ) ning maatriks  $\mathbf{Y}_\Sigma$  saadakse  $\mathbf{Y}_k$ -de summeerimisel, siis tänu sellele on meil võimalik vähendada maatriksite maksimaalset mahtu arvutamisel.

Iga tsükli jooksul luuakse vaid  $s_1 \times s_2 \times \dots \times s_r \times t$ -dimensionaalsed maatriksid  $\Delta_k$  ja  $\mathbf{Y}_k$  (jooksev  $K = 1$ ) ning moodustatakse sama dimensiooniga  $\mathbf{Y}_\Sigma$ . Tsükleid korratakse kuni kõikide olemasolevate eksperimendiandmete lugemid ( $x_{1k}, x_{2k}, x_{3k}, \dots, y_k$ ) on akumulieeritud maatriksisse  $\mathbf{Y}_\Sigma$ . Maatriks  $\mathbf{Y}_\Sigma$  on ekvivalentne maatriksiga, mille oleksime saanud arvutades

välja ühekorraga kõikidele olemasolevate eksperimendiandmete lugemitele vastavad maatriksid  $\Delta$  ja  $Y$  ja kasutades avaldist.

See on ka arvutusvõimsuse otstarbeka kasutamise seisukohalt peamine eelis, mida programm omab funktsiooni **anfis.m** ees. Viimane nõuab, et kõik andmed sisestataks ühekorraga.

Meetodi realisatsioon on aeganõudvam kui on defineeritud palju liikmesfunktsioone, ülekattumise määr on suurem ja treenimisandmete hulk on suur - kõik see toob kaasa maatriksite suured mahud.

Meetod ei hiilga totaalse täpsusega, kuid meid ei huvita mitte niivõrd kvantitatiivsed kui kvalitatiivsed hinnangud ja seetõttu võime selle koha pealt silma kinni pigistada.

Üheks võimalikuks meetodi puuduseks võib nimetada normeerimist, mille tõttu on võrdse kaaluga nii reeglid, mida kinnitab 1000 eksperimenti kui reeglid, mida kinnitab vaid üks. Seetõttu on parem kui mudelit on treenitud üksteisest vähe erinevate andmetega ning samuti on kasulik suhtuda ettevaatlikult tulemustesse, mis on ekstreemsed.

Algoritmi programse realisatsiooni töötamise eeldus on, et MatLabi Fuzzy Logic ToolBoxi abil oleks defineeritud nii biomassi kui toodangu juurdekasvu mudelite hägusad lahterdused (need on sisendmuutujate osas identsed) ning et eksperimendiandmete failiseeria oleks teisendatud sobivale kujule. Programm on paindlik, tema abil saab genereerida eksperimendiandmete alusel kuni 9 sisendiga mudeleid.

Biomassi juurdekasvu mudeli saamiseks teisendame failiseeria

$$f_1(k) \quad f_2(k) \quad x(k) \quad s_1(k) \quad s_2(k) \quad p(k) \quad V(k)$$

kujule:

$$x(k) \quad s_1(k) \quad s_2(k) \quad dx(k),$$

kus  $dx(k) = x(k) - x(k - 1)$

Toodangu juurdekasvu mudeli jaoks teisendame failiseeria kujule:

$$x(k) \quad s_1(k) \quad s_2(k) \quad dp(k),$$

kus  $dp(k) = p(k) - p(k - 1)$ .

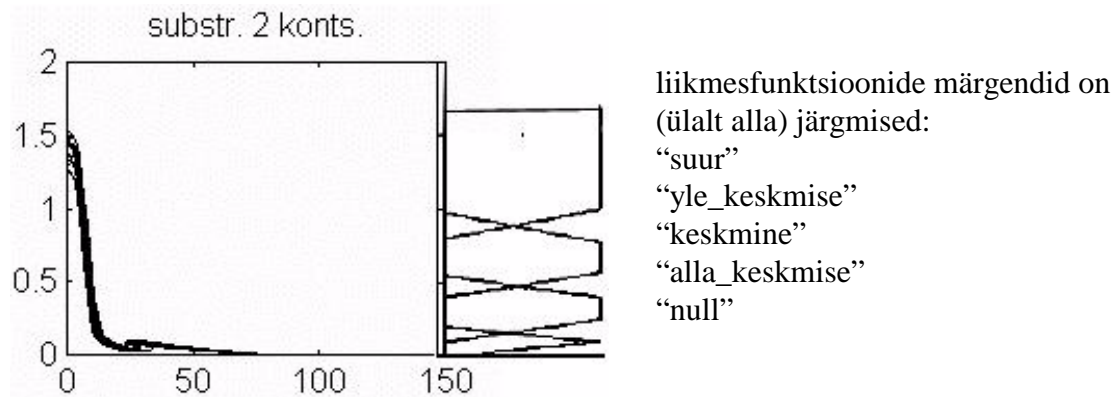
Programmi töö resultaatina genereeritakse algele, vaid hägusaid lahterdusi sisaldanud mudelile reeglikomplekt, teisisõnu saame nii sõltuvalt eksperimendiandmetest biomassi või toodangu juurdekasvu lingvistilised mudelid.

Praktilisest vaatevinklist on käesolev meetod üksikasjalikult vaatluse all peatükis 8.



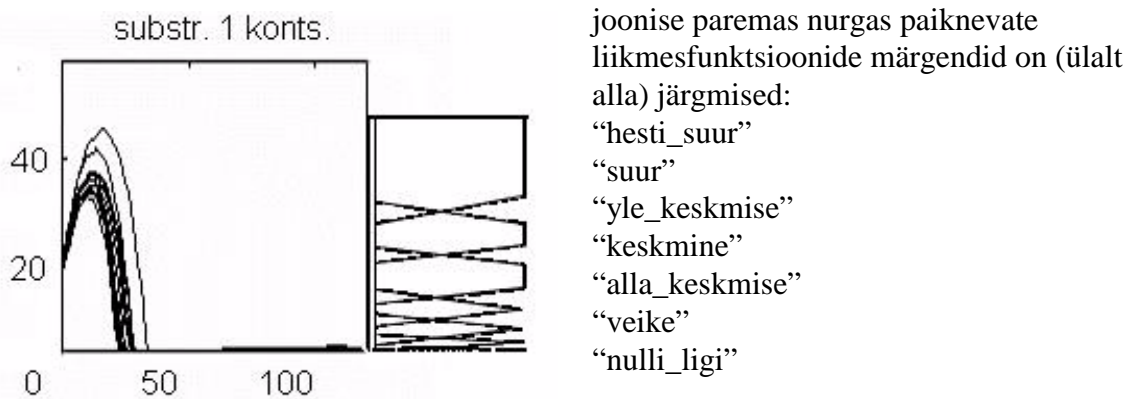
printsipi, et väiksemate kontsentratsioonide väärtuste poolt oleks substraadi kontsentratsioon tihedamalt lahterdatud kui suuremate poolt - ebahütlane lahterdus.

### joonis 8-2



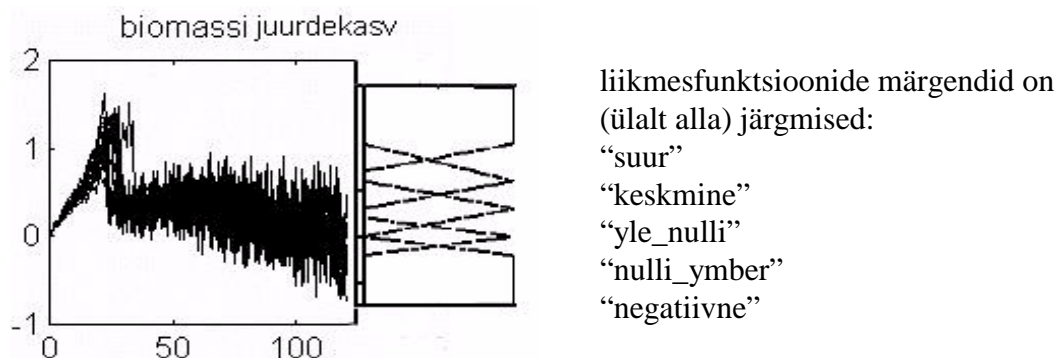
3. Substraadi 1 kontsentratsiooni otsustasime lahterdada rohkemaks hulgaks liikmesfunktsioonideks, kuna tema muutumispiirkond on suur võrreldes substraadi 2 kontsentratsiooniga. Muus osas kehtib kõik substraadi 2 kontsentratsiooni kohta öeldu

### joonis 8-3



4. Ülejäänud 2 defineerimist vajavad muutujat - toodangu juurdekasv ja biomassi juurdekasv on lahterdatud arvestades seda, et lingvistilised märgendid peegeldaksid nende reaalseid väärtuseid ja oleksid ilmekad. See lihtsustab edasist regulaatori projekteerimist. Näiteks olgu siin toodud biomassi juurdekasvu lahterdus:

### joonis 8-4



Biomassi ja toodangu juurdekasvu mudeleid genereerivat programmi kirjeldasime juba jaotises 7. Lisada võib veel, et genereerimise tulemusena tekivad `fis`-laiendiga failid, mis on käsitsetavad nii MatLabi käsuaknas kui Fuzzy Logic ToolBoxis.

## 8.2 Reeglite analüüs ja hägusa regulaatori süntees

### 8.2.1 Reeglite analüüsi alused

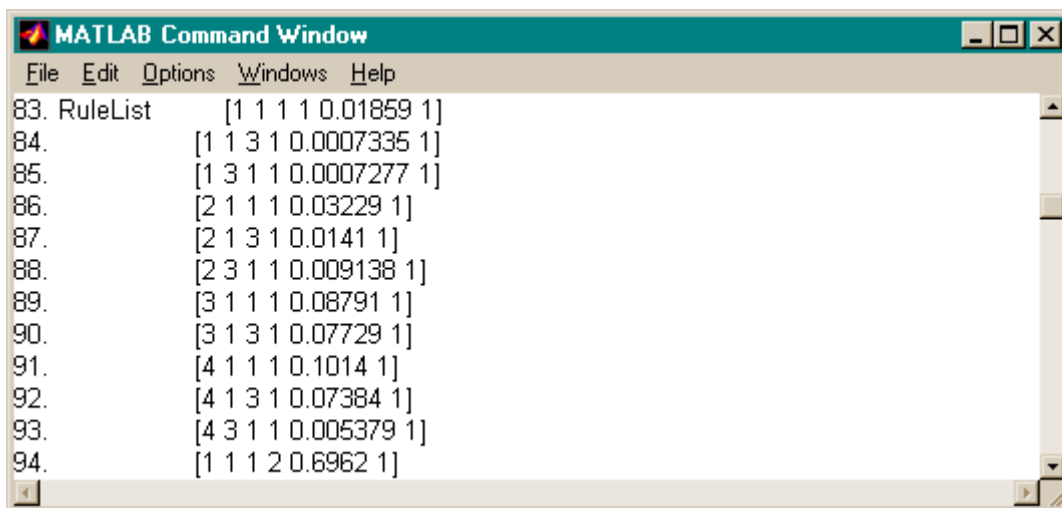
Genereeritud hägusa mudeli parameetrid (sisend- ja väljundmuutujate nimed, nende liikmesfunktsioonide hulgad, tüübid ja vastavad parameetrid, järeldusmeetodite tüübid, reeglikomplekt jne.) paiknevad kõik ühes ja samas `fis`-tüüpi failis.

Kui mällu lugeda sellest failist tuletatud `fis`-maatriks (vastav käsk: `fis_maatriks = readfis('failinimi.fis')`), siis seda maatriksit saab vaadelda käsu `showfis(fis_maatriks)` abil sellisel kujul, kus igal real paikneb esmalt võtmesõna ja teisalt võtmesõna parameetrid.

Näiteks:

1. Name juhtiv
2. Type mamdani
3. Inputs/ Outputs [4 1]
4. ....

Võtmesõna puudumine mingil real näitab, et kehtib eelmisel real defineeritud võtmesõna. Võtmesõna `RuleList` defineerib hägusa mudeli reeglid, kusjuures iga reegel esitatakse eraldi real.



```
MATLAB Command Window
File Edit Options Windows Help
83. RuleList      [1 1 1 1 0.01859 1]
84.              [1 1 3 1 0.0007335 1]
85.              [1 3 1 1 0.0007277 1]
86.              [2 1 1 1 0.03229 1]
87.              [2 1 3 1 0.0141 1]
88.              [2 3 1 1 0.009138 1]
89.              [3 1 1 1 0.08791 1]
90.              [3 1 3 1 0.07729 1]
91.              [4 1 1 1 0.1014 1]
92.              [4 1 3 1 0.07384 1]
93.              [4 3 1 1 0.005379 1]
94.              [1 1 1 2 0.6962 1]
```

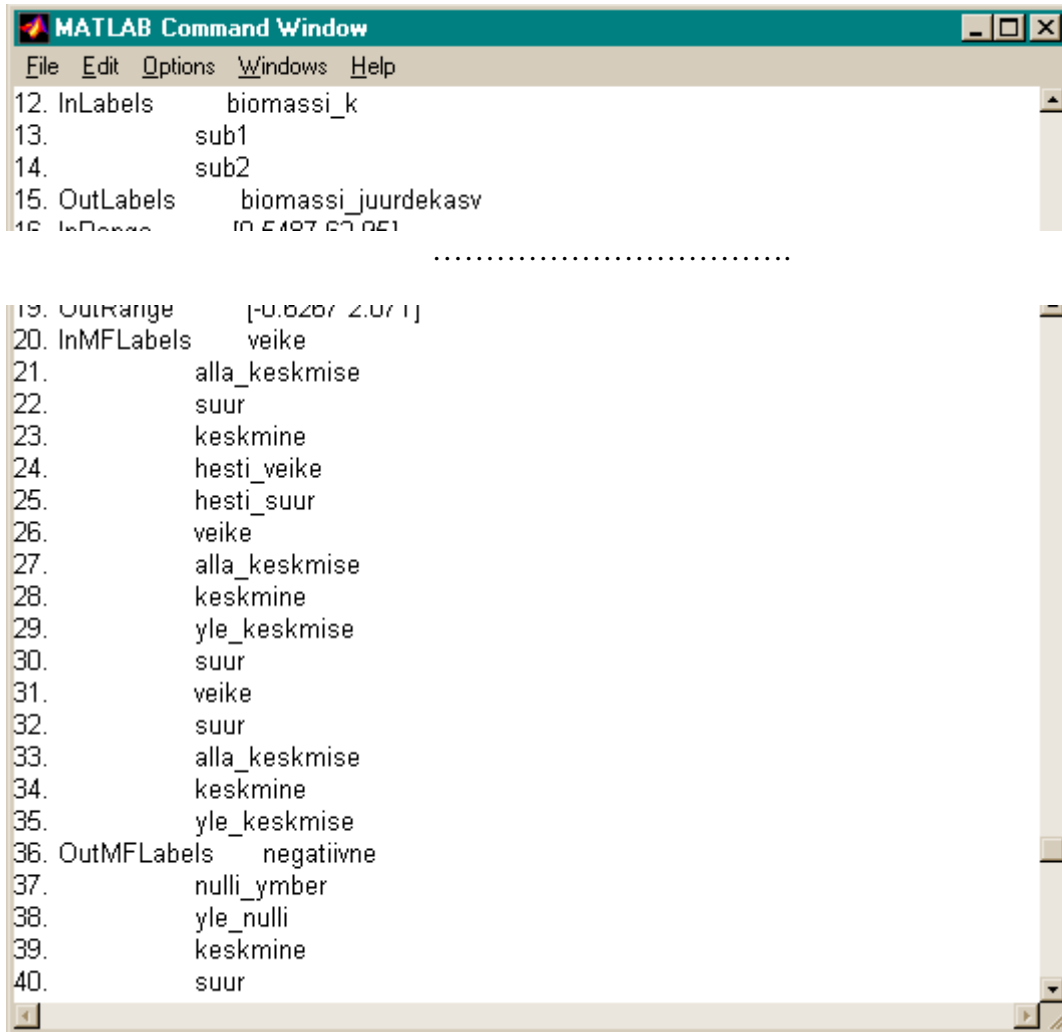
Kandiliste sulgude sisse jääv osa ehk reeglimateatriks sisaldab kogu infot, mida reegli täielikuks defineerimiseks vajatakse. Kui süsteemil on  $m$  sisendit ja  $n$  väljundit, siis esimesed  $m$  (antud juhul 3) elementi defineerivad reeglita hõlmatud sisendmuutujate liikmesfunktsioonid, järgmised  $n$  (antud juhul 1) reeglita hõlmatud väljundmuutujate liikmesfunktsioonid, eelviimane number on reegli kaal, viimane määrab mis tüüpi loogilist tehet reegli tingimus- ja järelduspoole komponentide sidumiseks kasutatakse (AND = 1, OR = 2).

Seega defineerib näiteks real nr. 87 paiknev maatriks `[2 1 3 1 0.0141 1]` järgmise reegli:

KUI (sisendmuutuja nr. 1 on liikmesfunktsioon nr. 2) JA (sisendmuutuja nr. 2 on liikmesfunktsioon nr. 1) JA (sisendmuutuja nr. 3 on liikmesfunktsioon nr. 3) SIIS (väljundmuutuja nr. 1 on liikmesfunktsioon nr. 1)

Reegli kaal on 0.0141.

Et sama fis-maatriksiga on defineeritud ka sisend- väljundmuutujad (võtmesõnad InLabels ja OutLabels) ning nende liikmesfunktsioonide lingvistilised märgendid (võtmesõnad InMFLLabels ja OutMFLLabels),



```
MATLAB Command Window
File Edit Options Windows Help
12. InLabels      biomassi_k
13.              sub1
14.              sub2
15. OutLabels     biomassi_juurdekasv
16. InMFLLabels  in_0.0207 0.0251
.....
19. OutRange     [-0.0207 2.071]
20. InMFLLabels  veike
21.              alla_keskmise
22.              suur
23.              keskmise
24.              hesti_veike
25.              hesti_suur
26.              veike
27.              alla_keskmise
28.              keskmise
29.              yle_keskmise
30.              suur
31.              veike
32.              suur
33.              alla_keskmise
34.              keskmise
35.              yle_keskmise
36. OutMFLLabels negatiivne
37.              nulli_ymber
38.              yle_nulli
39.              keskmise
40.              suur
```

tähendab märgistus [ 2 1 3 1 0.0141 1 ] ühtlasi järgmist reeglit (reegli kaal on reegli lõpus sulgudes olev arv):

KUI (biomassi\_k on “alla\_keskmise”) JA (sub1 on “keskmise”) JA (sub2 on “veike”) SIIS (biomassi\_juurdekasv on “negatiivne”) (0.1401)

MatLabis on reegleid võimalik esitada ka lingvistilisel kujul, mis on inimesele kindlasti mugavam reeglikuju, ent meie ei kasuta reeglite maatrikskujult lingvistilisele kujule teisendamiseks mitte MatLabi vahendeid, vaid spetsiaalselt Visual Basicus selle jaoks kirjutatud programijupikest. Miks? Vastus sellele on lihtne. Visual Basic on tihedalt seotud tabelarvutusprogrammiga MS Excel, mille andmetöötlusvõimalusi tahame kasutada reeglite analüüsimisel.

Reegliinfo viime MatLabist Excelisse lihtsa copy-paste protseduuri abil:

	A	B	C	D	E	F	G	H
1		[1 1 1 1 0.01859 1]						
2	84.	[1 1 3 1 0.0007335 1]						
3	85.	[1 3 1 1 0.0007277 1]						
4	86.	[2 1 1 1 0.03229 1]						
5	87.	[2 1 3 1 0.0141 1]						
6	88.	[2 3 1 1 0.009138 1]						
7	89.	[3 1 1 1 0.08791 1]						
8	90.	[3 1 3 1 0.07729 1]						
9	91.	[4 1 1 1 0.1014 1]						
10	92.	[4 1 3 1 0.07384 1]						
11	93.	[4 3 1 1 0.005379 1]						
12	94.	[1 1 1 2 0.6962 1]						
13	95.	[1 1 3 2 0.402 1]						
14	96.	[1 1 4 2 0.2742 1]						

Visual Basicus kirjutatu käivitamine resulteerub kümnekonna sekundi möödudes järgmises ekraanipildis:

	A	B	C	D	F
1	<b>biomassi_konts.</b>	<b>substr_1_konts.</b>	<b>substr_2_konts.</b>	<b>biomassi_juurdek.</b>	<b>reegli_kaal</b>
2	veike	hesti_veike	veike	negatiivne	0,01859
3	veike	hesti_veike	keskmine	negatiivne	0,0007335
4	veike	alla_keskmise	veike	negatiivne	0,0007277
5	alla_keskmise	hesti_veike	veike	negatiivne	0,03229
6	alla_keskmise	hesti_veike	keskmine	negatiivne	0,0141
7	alla_keskmise	alla_keskmise	veike	negatiivne	0,009138
8	keskmine	hesti_veike	veike	negatiivne	0,08791
9	keskmine	hesti_veike	keskmine	negatiivne	0,07729
10	suur	hesti_veike	veike	negatiivne	0,1014
11	suur	hesti_veike	keskmine	negatiivne	0,07384
12	suur	alla_keskmise	veike	negatiivne	0,005379
13	veike	hesti_veike	veike	nulli_ymber	0,6962
14	veike	hesti_veike	keskmine	nulli_ymber	0,402

Kuna iga muutuja liikmesfunktsiooni märgendid paiknevad eraldi veergudes, on Excelisse sisseehitatud vahendeid kasutades hõlbustada manada vaid teatud tingimusi täitvaid reegleid, mis hõlbustab nende analüüsimist. Näiteks tahame näha vaid selliseid reegleid, kus biomassi kontsentratsioon oleks “veike”, substraadi 1 kontsentratsioon “keskmine”, substraadi 2 kontsentratsioon “keskmine” ning reegli kaal oleks suurem kui 0,1 (liiga väikeste kaaludega reeglid on ebaolulised). Vastav ekraanipilt oleks järgmine:

	A	B	C	D	F
1	biomassi kont	substr_1 kont	substr_2 kon	biomassi juurde	reegli kaal
93	veike	keskmine	keskmine	yle_nulli	0,4637
178	veike	keskmine	keskmine	yle_keskmise	0,429
312					
313					0,8927
314					

2 of 310 records found      Sum=0,8927      NUM

Näeme, et kui biomassi kontsentratsioon on “veike”, substraadi 1 kontsentratsioon “keskmine” ja substraadi 2 kontsentratsioon “keskmine”, siis oodatav biomassi juurdekasv on tõenäosusega 0,4637 “yle\_nulli” ja tõenäosusega 0,429 “yle\_keskmise”. Summaarne tõenäosus on 0,8927, mis on väiksem kui 1 kuna oleme elimineerinud hulga väikeste kaaludega reegleid.

Olukorda, kus oleme sorteerinud välja reeglid, kus igale sisendile vastab vaid üks liikmesfunktsioon, nagu vaadeldud olukorras, nimetame edaspidi situatsiooniks. Situatsioonile vastavat reeglite järelaluspoole liikmesfunktsioonide kogumit nimetame sündmuseks. Lihtne arvutus näitab et situatsioonide koguarv on võrdne sisendite liikmesfunktsioonide arvu korrutisega, mis antud juhul on  $4 \cdot 7 \cdot 5 = 140$ .

Situatsioonile vastavate reeglite summaarne kaal saab olla ainult 1 (reaalses situatsioonis toimub sündmus kindlasti). Ainus alternatiiv on situatsioonile vastavate reeglite täielik puudumine. Ka siis kui oleme osa reegleid elimineerinud - näiteks väikeste kaaludega reeglid - saab summaarne kaal olla ühest väiksem, olles seda erinevam ühest, mida kõrgema oleme valinud mudeli reeglite kehtivusläve (antud näite puhul oli see mäletatavasti 0,1).

Sama protseduuri teeme läbi teise genereeritud hädusa mudeli - toodangu juurdekasvu mudeli - *fis*-failiga.

## 8.2.2 Seadesuuruste regulaatori süntees

Seadesuuruste regulaatori süntees toimub järk-järgult. Mäletatavasti oli eesmärgiks luua hädus regulaator, mille ainsaks sisendmuutujaks on biomassi kontsentratsioon ning väljundmuutujateks substraatide 1 ja 2 kontsentratsioonid (joonis 8-5).

## joonis 8-5



Iga etteantud biomassi kontsentratsiooni  $x$  kohta peab selline regulaator genereerima vastavad substraatide kontsentratsioonid  $s_1$  ja  $s_2$ . Sellise regulaatori reeglid avalduvad kujul:

KUI  $x$  on ... SIIS  $s_1$  on ... ja  $s_2$  on ...

Selline regulaator on täielik kui on defineeritud biomassi kontsentratsiooni liikmesfunktsioonide koguarvuga võrdne arv reegleid.

Nende reeglite tuletamiseks kasutame samaaegselt nii biomassi kui toodangu juurdekasvu mudeleid.

Meil oli muutujale biomassi kontsentratsioon defineeritud hägus lahterdus “veike”, “alla\_keskmise”, “keskmise”, “suur”, seega sünteesitava regulaatori reeglite arv on samuti 4.

Reeglite moodustamisel selgitame kõigepealt välja toodangu juurdekasvu seisukohalt kõige soodsamad substraatide kontsentratsioonid kasutades selleks toodangu juurdekasvu mudelit. See tähendab, et kõigepealt valime välja protsessi tootmisrežiimi.

Võtame aluseks, et biomassi kontsentratsioon tootmisrežiimis võib olla kas “suur” või “keskmise”, äärmisel juhul “alla\_keskmise”. Soovitav toodangu juurdekasv aga kas “suur” või “keskmise”.

Selge on see, et sellistest kombinatsioonidest parim on see, mille puhul biomassi kontsentratsioon on “alla\_keskmise”, toodangu juurdekasv aga “suur”. Iseasi on, kas selline kombinatsioon ka realselt teostatav on. Seda peavadki näitama toodangu juurdekasvu mudeli reeglid.

### 8.2.2.1 Tootmisrežiimi defineerimine

Alustame reeglite analüüsi sellistest reeglitest, kus biomassi kontsentratsioon  $x$  on “alla\_keskmise”. Toodangu juurdekasvu mudelis vastab antud tingimusele 15 reeglit. Toome välja perspektiivsemad situatsioonid.

#### Situatsioon 1.1.1

biomass	substraat 1	substraat2	toodangu juurdekasv	reegli kaal
alla_keskmise	keskmise	veike	yle_nulli	0,1904
alla_keskmise	keskmise	veike	keskmise	0,7507
				<b>0,9411</b>

#### Situatsioon 1.1.2

biomass	substraat 1	substraat2	toodangu juurdekasv	reegli kaal
alla_keskmise	alla_keskmise	veike	keskmise	0,8379
				<b>0,8379</b>

Situatsioon 1.1.2 ületab situatsiooni 1.1.1, kuid mitte oluliselt, seega, teoreetiliselt võiks olla tootmisrežiimiks nii situatsioon 1.1.1 kui situatsioon 1.1.2

x on "keskmine"

Sellist tingimust rahuldavad toodangu juurdekasvu mudelis 22 reeglit. Perspektiivsemad neist oleks:

Situatsioon 1.2.1

biomass	substraat 1	substraat2	toodangu juurdekasv	reegli kaal
keskmine	null	null	yle_nulli	0,2063
keskmine	null	null	keskmine	0,499
keskmine	null	null	suur	0,2853
				<b>0,9906</b>

Situatsioon 1.2.2

biomass	substraat 1	substraat2	toodangu juurdekasv	reegli kaal
keskmine	veike	null	yle_nulli	0,1982
keskmine	veike	null	keskmine	0,4922
keskmine	veike	null	suur	0,2992
				<b>0,9896</b>

x on "suur"

Situatsioon 1.3.1

biomass	substraat 1	substraat2	toodangu juurdekasv	reegli kaal
suur	veike	null	keskmine	0,1282
suur	veike	null	suur	0,8241
				<b>0,9523</b>

Situatsioon 1.3.2

biomass	substraat 1	substraat2	toodangu juurdekasv	reegli kaal
suur	alla_keskmise	null	yle_nulli	0,1418
suur	alla_keskmise	null	suur	0,7817
				<b>0,9235</b>

Situatsioon 1.3.3

biomass	substraat 1	substraat2	toodangu juurdekasv	reegli kaal
suur	null	null	keskmine	0,1495
suur	null	null	suur	0,6856
				<b>0,8351</b>

Situatsioon 1.3.4

biomass	substraat 1	substraat2	toodangu juurdekasv	reegli kaal
suur	alla_keskmise	veike	yle_nulli	0,1153
suur	alla_keskmise	veike	suur	0,8224

Biomassi suur kontsentratsioon annab konkurentsitult kõige suurema toodangu juurdekasvu, mis on ka loogiline.

Lõppresultaadine valime siit välja 5 tootmisrezhiimi, mida realiseerida - situatsioonid 1.3.1, 1.3.2, 1.3.4 kui perspektiivsemad ning võrdluseks situatsioonid 1.1.2 ja 1.2.2. Seega:

	biomassi kontsentratsioon	substr. 1 kontsentratsioon	substr. 2 kontsentratsioon
1.	alla_keskmise	alla_keskmise	veike
2.	keskmine	veike	null
3.	suur	veike	null
4.	suur	alla_keskmise	null
5.	suur	alla_keskmise	veike

Sõelalejäänud variandid tingivad meie edasise tegutsemise. Iga väljavalitud variant nõuab unikaalset stsenaariumi. Kui oleme välja valinud viis enda arust heade näitajatega tootmisrezhiimi, siis edasise tegutsemise tulemusena valmib meie käe all viis erinevat hägusat regulaatorit, mille kvaliteedi lõplikuks näitajaks on toimetulek fermentatsiooniprotsessiga.

### 8.2.2.2 Juhtimisstsenaariumide väljatöötamine

Tootmisrezhiimidest lähtudes otsime nüüd biomassi juurdekasvu mudelist situatsioone, mis tagaksid biomassi kiire juurdekasvu kuni biomassi väärtuseni

- “alla\_keskmise” (situatsioon 1.1.2 - stsenaarium A)
- ”keskmine” (situatsioon 1.2.2 - stsenaarium B)
- “suur” (situatsioon 1.3.1 - stsenaarium C1, situatsioon 1.3.2 - stsenaarium C2, situatsioon 1.3.4 - stsenaarium C3)

Kõik need unikaalsed stsenaariumid algavad ühtemoodi. Stsenaariumi alguse dikteerib fermentatsiooniprotsessi algne seis, mis on määratud fermentatsioonikeskonna komponentide algkontsentratsioonidega.

Eksperimendiandmete vastav analüüs annab meile, et alghetkel on fermentatsioonikeskonna komponentide kontsentratsioonid järgmised:

biomass  $0,5 \pm 0,1$   
substraat1  $20,0 \pm 4,0$   
substraat2  $1,5 \pm 0,3$   
(toodang on 0)

Need arvulised kontsentratsioonid vastavad komponentide liikmesfunktsioonidele ligilähedaselt järgmiselt:

biomass - “veike”  
substraat1 - “yle\_keskmise”  
substraat2 - ”suur”

Analüüsigem neid tingimusi täitvaid situatsioone, kasutades biomassi ja toodangu juurdekasvu mudeleid. Saame, et:

Situatsioon 2.1.1

**biomass      substraat 1      substraat2      biomassi juurdekasv      reegli kaal**

veike yle\_keskmise suur nulli\_ymber 1

Situatsioon 2.1.1 viitab sellele, et fermentatsioonikeskkonna komponentide alkonsentratsioonid pole kuigi sobilikud biomassi kiireks kasvuks. Samad suurused tagavad küll sobiliku toodangu juurdekasvu, toetudes toodangu juurdekasvu mudelile (situatsioon 1.4.1), kuid meil tuleb siiski valida efektiivsem biomassi juurdekasvu režiim.

Situatsioon 1.4.1

biomass	substraat 1	substraat2	toodangu juurdekasv	reegli kaal
veike	yle_keskmise	suur	nulli_ymber	1

Selleks tuleb analüüsida reegleid biomassi juurdekasvu mudelis kus x on "veike"  
Võimalikud variandid:

Situatsioon 2.1.2/1.4.2

biomass	substraat 1	substraat2	biomassi juurdekasv	reegli kaal
veike	alla_keskmise	veike	suur	1
			<b>toodangu juurdekasv</b>	
veike	alla_keskmise	veike	yle_nulli	0,5827
veike	alla_keskmise	veike	keskmise	0,4173

Situatsioon 2.1.3/1.4.3

biomass	substraat 1	substraat2	biomassi juurdekasv	reegli kaal
veike	keskmise	veike	suur	0,977
			<b>toodangu juurdekasv</b>	
veike	keskmise	veike	yle_nulli	0,5189
veike	keskmise	veike	keskmise	0,4811

Situatsioon 2.1.4/1.4.4

biomass	substraat 1	substraat2	biomassi juurdekasv	reegli kaal
veike	yle_keskmise	veike	keskmise	0,1285
veike	yle_keskmise	veike	suur	0,8715
			<b>toodangu juurdekasv</b>	
veike	yle_keskmise	veike	yle_nulli	0,6487
veike	yle_keskmise	veike	keskmise	0,3508

Kõige kiirema biomassi kasvu tagab situatsioon 2.1.2, kõige väiksema samaaegse toodangu juurdekasvu situatsioon 2.1.4. Antud regulaatori sünteesi staadiumis on kõik kolm võrdse potentsiaaliga võimalused. Siit tuleneb, et meie võimalike regulaatorite arv kasvab kolmekordseks

### 8.2.2.3 Edasine süntees

Edasine süntees toimub biomassi kontsentratsiooniga määratud protsessi staadiumite kaupa. Kusjures stsenaariumiti on nõutav järgmine (kasvurežiimi all on mõistetud biomassi kiireks juurdekasvuks soodsat režiimi, režiimivälise all olukorda kus pole õnnestunud stsenaariumijärgset tootmisrežiimi säilitada):

<b>Biomassi konts.</b>	<b>Stsen.</b>	<b>Režiimi iseloom</b>	<b>ülesanne</b>
alla_keskmise	A	tootmisrežiim	toodangu kiire kasv, biomassi kasvu pidurdus
-- // --	B	kasvurežiim	biomassi kiire, toodangu aeglane kasv
-- // --	C1	-- // --	-- // --

-- // --	C2	-- // --	-- // --
-- // --	C3	-- // --	-- // --
keskmine	A	rezhiimiväline	stsenaarium muundub stsenaariumiks B
-- // --	B	tootmirezhiim	toodangu kiire kasv, biomassi kasvu pidurdus
-- // --	C1	kasvurezhiim	biomassi kiire, toodangu aeglane kasv
-- // --	C2	-- // --	-- // --
-- // --	C3	-- // --	-- // --
suur	A	rezhiimiväline	stsenaarium muundub stsenaariumiks C
-- // --	B	-- // --	-- // --
-- // --	C1	tootmirezhiim	toodangu kiire kasv, biomassi kasvu pidurdus
-- // --	C2	-- // --	-- // --
-- // --	C3	-- // --	-- // --

Protsessi esimeses, juba vaadeldud staadiumis on kõik stsenaariumid loomulikult kasvurezhiimis. Kui stsenaariumi tootmisrezhiimis toimub üleminek järgmisesse protsessi staadiumisse seetõttu, et biomassi kasvu pole vajalikul määral õnnestunud pidurdada, tuleb üle võtta selles staadiumis tootmisrezhiimis oleva stsenaariumi juhtimisreeglid. See tähendab et stsenaarium A muundub protsessi kolmandas staadiumis (biomass on “keskmine”) stsenaariumiks B. Protsessi viimases staadiumiks saab stsenaarium muunduda ainult stsenaariumiks C. Vaatleme nüüd lähemalt kõiki neid stsenaariumeid erinevas protsessi staadiumite kaupa lähemalt.

Protsessi teine staadium - biomass on “alla\_keskmise”

### stsenaarium A

Selles süsteemi olekus dikteerib stsenaarium A ette tootmisrezhiimi, seega tuleb meil üksnes kontrollida, mil määral situatsiooniga 1.1.1 määratud tingimustel pidurdub biomassi juurdekasv.

Situatsiooni 1.1.2 paariliseks biomassi juurdekasvu mudelis on situatsioon 2.2.1:

Situatsioon 2.2.1

biomass	substraat 1	substraat2	biomassi juurdekasv	reegli kaal
alla_keskmise	alla_keskmise	veike	suur	1

Stsenaariumisse A on sisse programmeeritud vältimatu muundumine stsenaariumiks B ja tema realiseerimine pole sisuliselt võimalik

### Stsenaariumid B ja C\*

Protsessi staadiumis, kus biomassi kontsentratsioon on “alla\_keskmise”, tuleb meil stsenaariumide B, C1, C2 ja C3 järgi tagada edasine biomassi kiire juurdekasv. Esimene võimalus on kasutada situatsiooni 2.2.1. Samas on konkurentsivõimelised ka situatsioonid 2.2.2 ja 2.2.3, millele vastavad toodangu juurdekasvu mudeli situatsioonid tagavad väiksema toodangu juurdekasvu

Situatsioon 2.2.2/1.5.1

biomass	substraat 1	substraat2	biomassi juurdekasv	reegli kaal
---------	-------------	------------	---------------------	-------------

alla_keskmise	yle_keskmise	veike	suur	0,9546
			<b>toodangu juurdekasv</b>	
alla_keskmise	yle_keskmise	veike	yle_nulli	0,3798
alla_keskmise	yle_keskmise	veike	keskmine	0,6192

Situatsioon 2.2.3/1.5.2

<b>biomass</b>	<b>substraat 1</b>	<b>substraat2</b>	<b>biomassi juurdekasv</b>	<b>reegli kaal</b>
alla_keskmise	keskmine	veike	suur	0,9922
			<b>toodangu juurdekasv</b>	
alla_keskmise	keskmine	veike	yle_nulli	0,1904
alla_keskmise	keskmine	veike	keskmine	0,7507

Protsessi kolmas staadium - x = "keskmine"

Stsenaariumide B jaoks on tegu tootmisrezhiimiga, milleks sobiv situatsioon on ülalpool juba välja valitud, seetõttu tuleb kontrollida vaid et biomassi juurdekasv pidurduks.

### stsenaarium B

Situatsiooni 1.2.2 paariliseks on situatsioon 2.6.1

<b>biomass</b>	<b>substraat 1</b>	<b>substraat2</b>	<b>biomassi juurdekasv</b>	<b>reegli kaal</b>
keskmine	veike	null	negatiivne	0,1272
keskmine	veike	null	null_ymber	0,1552
keskmine	veike	null	yle_nulli	0,5227
keskmine	veike	null	keskmine	0,1942
				<b>0,9993</b>

Biomassi juurdekasv pidurdub ehkki mitte täielikult ja seega on ka temasse sisse programmeeritud muundumine stsenaariumiks C

### Stsenaarium C

Stsenaariumi C kohaselt tuleb tagada biomassi kiire juurdekasv.

Seda tagavad kõik praktiliselt kõik reeglid, milles substraadi 2 kontsentratsioon on "veike" (tabel 8-2)

#### tabel 8-1

<b>biomass</b>	<b>substraat 1</b>	<b>substraat2</b>	<b>biomassi juurdekasv</b>	<b>reegli kaal</b>
keskmine	alla_keskmise	veike	suur	1
keskmine	keskmine	veike	suur	0,9972
keskmine	yle_keskmise	veike	suur	0,9895
keskmine	suur	veike	suur	0,9327
keskmine	hesti_suur	veike	suur	0,9336

Sobiva variandi väljavalikul on aluseks kontsentratsioonid eelmises protsessi staadiumis ja samaaegne toodangu juurdekasv. Mida suurem on substraadi 1 kontsentratsioon, seda väiksem on toodangu juurdekasv

Igal juhul toimub samaaegselt biomassi kiire juurdekasvuga ka juba toodangu küllaltki kiire juurdekasv. Seda meil vältida ei õnnestu.

Protsessi viimane staadium - biomassi kontsentratsioon on "suur"

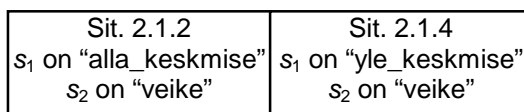
Stsenaariumid C on tootmisrežiimis vastavalt jaotisele 8.2.2.1

#### 8.2.2.4 Sünteesi resultaat

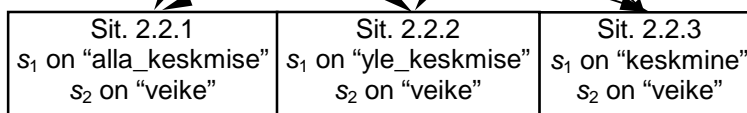
Võtame regulaatorite sünteesi resultaadid kokku ja esitame nad graafiliselt (tootmisrežiimid joonisel 8-6 on ümbritsetud jämeda joonega):

**joonis 8-6**

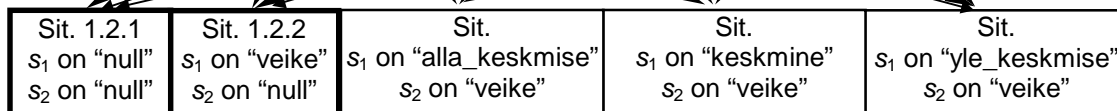
biomass on "veike"



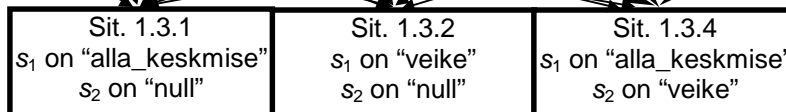
biomass on "alla\_keskmise"



biomass on "keskmise"



biomass on "suur"



Väljavalitud situatsioonid viitavad, et substraadi 2 kontsentratsiooni ajaline muutumine on triviaalne - kui tegu on kasvurežiimiga, peab kontsentratsioon olema "veike" ja kui tootmisrežiimiga, siis "null". Substraadi 1 kontsentratsiooni puhul sellist lihtsat seost ei eksisteeri.

Väljavalitud situatsioonide alusel on meil võimalik konstrueerida 90 (unikaalsete teede arv situatsioonist 2.1.\* situatsiooni 1.3.\*) hägusat regulaatorit. Siit näeme, et meie poolt koostatud fermentatsiooniprotsessi mudel on peaaegu ammendamatu allikas hägusa regulaatori sünteesiks.

Väljavalitud situatsioonide näol oleme selekteerinud välja protsessi optimaalseks kulgemiseks sobilikud substraatide kontsentratsioonid ja järjestanud need vastavalt. Vastavad regulaatorid ei ole aga päriselt vastavuses reaalse protsessiga, kuivõrd me regulaatori sünteesil

pole arvesse võtnud ei protsessi etappide tegelikku kestust ega fermentatsioonikeskkonna mahtu.

Meetod sellisel kujul nagu me ta välja arendasime seda ka kuidagi ei võimalda, seega tuleb tema reaalsel rakendamisel lahendada see probleem eraldi. Selleks on põhimõtteliselt kolm võimalust

A) Kõige lihtsam ja loomulikum võimalus - viia protsessi läbi täpselt niikaua kuni fermentaator on täidetud. Sellisel juhul on ilmselt eelistatumad stsenaariumid, mille puhul fermentaator täitub kiiremini, sest nii saame lisaks suuremale produkti kogusele ka märkimisväärse ajalise võidu võrreldes nominaalsete juhtimisprofiilidega. Lähenedamine viib tagajärjeni, et iga üksik fermentatsioon on ette mitte ennustatava kestusega. Et me peatükis 6 seadsime juhtimise eesmärgiks saavutada etteantud aja kestel maksimaalne kogus toodangut, siis me sellise lahendusega ei piirdu, ehkki see oleks põhimõtteliselt võimalik.

B) Kasutada protsessi juhtimiseks ainult taolisi regulaatori versioone, mis täidavad fermentaatori protsessi etteantud lõpphetkeks või selle vahetus läheduses. Stsenaariumid, mille täitmisel viiakse fermentaatorisse suuremad kogused substraate ja mis sellega täidavad fermentaatori kiiremini, on oletatavasti sellised, mis seavad ette suuremad substraatide kontsentratsioonid. Kuna aga iga protsess on n.ö. unikaalne ja mitteetteennustatavate tagajärjedega, võib ühe ja sama juhtimisalgoritmiga juhitud fermentatsiooni tagajärjeks olla üleajav, täis või pooltühi fermentaator. Protsess ei ole lihtsalt sedavõrd juhitav. Seega ei anna ka selline lähenedamine soovitud resultaati.

C) Kõige kindlamaks võimaluseks tagada fermentaatori täitumine protsessi lõpphetkeks on väline sekkumine protsessi juhtimisse. Protsessi algul laseme asjadel kulgeda oma rada hägusa regulaatori juhtimisel. Protsessi lõpuosas tuleb aga viia sisse täiendavad kontrollprotseduurid fermentaatori täitmise tarbeks. Võib näiteks kehtestada teatud tingimused, mille järgi tunneme ära, kas tegu on juhtumiga, kus fermentaator täitub kas liiga kiiresti või liiga aeglaselt ja juhtida fermentaatorisse sellised substraatide kogused, et järelejäänud vaba ruum fermentaatoris täituks järelejäänud aja jooksul ühtlaselt.

Selliseid kontrollprotseduure on protsessi juhtimisel ka kasutatud. Sisuliselt toimub siis protsessi juhtimine kolmes etapis:

1. etapp - biomassi kontsentratsiooni tõstmine toodangu andmiseks optimaalsele tasemele
2. etapp - produkti kontsentratsiooni kiire juurdekasv
3. etapp - fermentaatori täitumise tagamine, kusjuures biomassi ja produkti kontsentratsioonid jäävad enam-vähem saavutatud tasemetele.

Kõigi võimalike stsenaariumide realiseerimine pole mõeldav ega oma ka mõtet. Sisuliselt piisab ju ainult ühest realisatsioonist, kui see tagab piisavalt head resultaadid. Kuid milline valida ja millest valikut alustada?

Käesoleva töö raames on läbi mängitud 5 stsenaariumi:

1) Stsenaarium C3.1 kui maksimaalseid substraatide kontsentratsioone nõudev stsenaarium.

KUI  $x$  on "veike" SIIS  $s_1$  on "yle\_keskmise" JA  $s_2$  on "veike"

KUI  $x$  on "alla\_keskmise" SIIS  $s_1$  on "yle\_keskmise" JA  $s_2$  on "veike"

KUI  $x$  on "keskmise" SIIS  $s_1$  on "yle\_keskmise" JA  $s_2$  on "veike"

KUI  $x$  on "suur" SIIS  $s_1$  on "alla\_keskmise" JA  $s_2$  on "veike"

2) Stsenaarium C2.1 kui kõige väiksemaid substraatide kontsentratsioone nõudev stsenaarium

- KUI  $x$  on "veike" SIIS  $s_1$  on "alla\_keskmise" JA  $s_2$  on "veike"
- KUI  $x$  on "alla\_keskmise" SIIS  $s_1$  on "alla\_keskmise" JA  $s_2$  on "veike"
- KUI  $x$  on "keskmise" SIIS  $s_1$  on "nulli\_ligi" JA  $s_2$  on "null"
- KUI  $x$  on "suur" SIIS  $s_1$  on "veike" JA  $s_2$  on "null"

3) Stsenaarium C1.1

- KUI  $x$  on "veike" SIIS  $s_1$  on "yle\_keskmise" JA  $s_2$  on "veike"
- KUI  $x$  on "alla\_keskmise" SIIS  $s_1$  on "yle\_keskmise" JA  $s_2$  on "veike"
- KUI  $x$  on "keskmise" SIIS  $s_1$  on "yle\_keskmise" JA  $s_2$  on "veike"
- KUI  $x$  on "suur" SIIS  $s_1$  on "alla\_keskmise" JA  $s_2$  on "null"

4) Stsenaarium C2.2

- KUI  $x$  on "veike" SIIS  $s_1$  on "yle\_keskmise" JA  $s_2$  on "veike"
- KUI  $x$  on "alla\_keskmise" SIIS  $s_1$  on "yle\_keskmise" JA  $s_2$  on "veike"
- KUI  $x$  on "keskmise" SIIS  $s_1$  on "veike" JA  $s_2$  on "null"
- KUI  $x$  on "suur" SIIS  $s_1$  on "veike" JA  $s_2$  on "null"

5) Stsenaarium C1.2

- KUI  $x$  on "veike" SIIS  $s_1$  on "alla\_keskmise" JA  $s_2$  on "veike"
- KUI  $x$  on "alla\_keskmise" SIIS  $s_1$  on "alla\_keskmise" JA  $s_2$  on "veike"
- KUI  $x$  on "keskmise" SIIS  $s_1$  on "alla\_keskmise" JA  $s_2$  on "veike"
- KUI  $x$  on "suur" SIIS  $s_1$  on "alla\_keskmise" JA  $s_2$  on "null"

Katsetulemuste graafikud on toodud lisas lehekülgedel 2-6

Mõni sõna ka kasutatavatest PI-regulaatoritest. Substraatide kontsentratsioonide tagamiseks on siin kasutatud täiesti tavalisi diskreetseid PI regulaatoreid, mis üldjuhul suudavad tagada substraatide kontsentratsioonide seadesuuruste järgimise rahuldavalt. Nende häälestamine pole ka eriti keeruline. Samas on täiesti mõeldav hägusate PI regulaatorite kasutamine, viimasel juhul oleks juhtimissüsteem läbinisti hägus.

## 8.3 Resultaatide analüüs

Osutub, et praktiliselt kõik valitud stsenaariumid toovad endaga kaasa resultaadid, mis märgatavalt ületavad nominaalsete juhtimisprofiilidega saavutatu. Kui protsessi kvaliteedinäitaja oli nominaalsete juhtimisprofiilide korral maksimaalselt  $6 \cdot 10^4$ , siis hägusa juhtimisalgoritmi puhul ei lange see naljalt allapoole väärtusest  $10 \cdot 10^4$ , see tähendab keskmiselt 30% suuremat produkti kogust ühest fermentatsioonist. Kuna juhtimisalgoritmi sünteesi kuludena tulevad antud juhul sisuliselt arvesse ainult selle projekteerimis- ja konstrueerimiskulud (täiendavaid ressursse raiskavaid eksperimente me ei sooritanud), tasub sellise juhtimissüsteemi reaalne rakendamine igal juhul ära, kui ainult ka realses elus simulaatoril saavutatud resultaadid kehtivad.

Erandiks edukate stsenaariumide hulgas on stsenaarium C3.1, mida järgides saavutatud resultaate on raske edukaks nimetada. Sellel aga on küllaltki lihtne põhjus. Nimelt tingib suurte substraatide kontsentratsioonide nõue fermentaatori varase täitumise ning protsessi lõpuosas on sisuliselt võimatu protsessi kulgu soovitud suunas mõjutada. Sealhulgas ei

õnnestu meil tagada ka nõutud substraatide kontsentratsioonid. Viga on niisiis tingitud pigem stsenaariumiga püstitatud tingimuste mitterahuldatusel kui vigasest stsenaariumist.

Üldjoontes jagunevad edukamad stsenaariumid kahte gruppi - esimese grupi stsenaariumid (C2.1, C2.2) toovad enesega kaasa fermentaatori enneaegse täitumise ning rakendada tuleb ületäitumise erirezhiimi, teised (C1.1, C1.2) alatäitumise ning tööle rakendatakse alatäitumise erirezhiim. Sõltub see peaaesjalikult tootmisrezhiimiga determineeritud substraadi 1 kontsentratsioonist. Kuna tootmisrezhiimis viibib süsteem kõige pikemat aega, mõjutavad just tootmisrezhiimi substraatide kontsentratsioonid kõige rohkem fermentaatori täitumise kiirust.

Täheldatav on veel järgmine - ületäitumise erirezhiim, milles substraadi 1 kontsentratsioon langeb järsult, on toodangu kontsentratsiooni juurdekasvu seisukohast soodsam kui alatäitumise erirezhiim. Hoolimata sellest, et toitainete nappuses hakkab biomass vaikselt välja surema.

Kui vaadelda nüüd seda, kuidas suudab juhtimisüsteem tagada substraatide etteantud kontsentratsioonid protsessi käigus, siis üldjuhul kui nõutud substraatide kontsentratsioonid ei ole väga suured ning süsteem ei viibi üle- või alatäitumise erirezhiimis, suudab ta seda Erandiks on protsessi järgud, kus toimub kas biomassi või toodangu väga intensiivne juurdekasv ja siin segab juhtimiseesmärgi saavutamist peamiselt suur diskreetimisintervall.

Niisiis võib tõdeda, et fermentatsiooniprotsessi lingvistilise mudeli kasutamine invertteeritud ja lihtsustatud kujul protsessi regulaatorina on edukas.

## 9 Kokkuvõte

Käesoleva töö ülesandeks oli uurida hägusate modelleerimis- ja juhtimisvõtete kasutusvõimalusi fermentatsiooniprotsessi juhtimisel. Meetodite testimisel kasutati reaalse fermentatsiooniprotsessi simulaatorit, mis kujutab endast programmeerimiskeeles C++ kirjutatud vastavat programmi.

Seostades fermentatsiooniprotsesside optimaaljuhtimise põhimõtteid ja hägusloogika võimalusi töötati välja konkreetse fermentatsiooniprotsessi võimalikud juhtimisstrateegiad. Seejuures peeti võimalikuks kasutada juhtimisalgoritmi sünteesiks nii närvivõrgulaadse kui klassikalise hägusa struktuuriga protsessi mudeleid. Lõpliku juhtimissüsteemi projekteerimise aluseks oli protsessile eksperimendiandmete põhjal genereeritud hägus lingvistiline mudel.

Fermentatsiooniprotsessi modelleerimiseks kasutati nominaalsete juhtimisprofiilidega läbi viidud protsessi eksperimendiandmeid. Seega võib tööga saavutatut nimetada ka senise juhtimisstrateegia täiustamiseks. Samas kirjutati lahti ka lähtekohad juhuks, kui tuleb alustada n.-ö. puhtalt lehelt. Fermentatsiooniprotsessi lingvistilise mudeli inverteeritud ja lihtsustatud versioon ongi kasutusel protsessi regulaatorina. Kuna fermentatsiooniprotsessi muutujate koguselised määratlused on lingvistilised, teeb see regulaatori projekteerimise inimolevuse jaoks loomulikuks ja hõlpsaks. Juhtimisalgoritmi süntees eeldas lingvistilise mudeli meetodi laiendamist mitmemõõtmeliste süsteemidele, mis on läbi viidud käesoleva töö autori poolt

Protsessi regulaator kujutab endast mehhanismi, mis määrab konkreetse biomassi kontsentratsiooni korral soovitud substraatide kontsentratsioonid fermentaatoris. Viimaste tagamiseks kasutatakse PI-regulaatoreid. Tegu on niisiis hierarhilise juhtimissüsteemiga, kus ülemine tase (protsessi regulaator) determineerib juhtimisülesande ning alumine tase (PI-regulaatorid) realiseerib selle. Väärub märkimist, et juhtimissüsteem on tegelikult küllaltki primitiivne.

Nii modelleerimisalgoritm kui juhtimissüsteem realiseeriti MatLabi keskkonnas ning ilmnis, et hägusa juhtimise abil tõusis fermentatsioonisaaduse kontsentratsioon ca 30%.

Teatud mõttes on käesolev töö üksnes autori esimene arglik samm hägusas modelleerimis- ja juhtimisvaldkonnas. Edasist uurimist vääriskid nii käesolevas töös lõpuni käimata teharud (näiteks fermentatsiooniprotsessi modelleerimine ja juhtimine närvivõrgulaadse hägusa mudeliga) kui ka juhtimisalgoritmi süntees teistsuguste mõõdetavate suurustega fermentatsiooniprotsessidele. Mõeldav on ka sellise protsessi regulaatori projekteerimine, mis juhtimise käigus jooksvalt salvestab eksperimendiinformatsiooni ning kasutab seda pidevalt juhtimisalgoritmi täiustamiseks. Samuti väärisk hägusad modelleerimis- ja juhtimismeetodid tõsisest analüüsi ka laiemas plaanis - missuguste protsesside puhul saab neid edukalt rakendada, missuguste puhul mitte.

Käesoleva töö head resultaadid üksnes julgustavad seda tegema.

## Kasutatud kirjandus

- [1] Petri Vuorimaa. Fuzzy Self-Organizing Map and Its Applications. Tampere University of Technology, 1995
- [2] Yoichi Kisuta and Michimasa Kishimoto. Fuzzy supervisory control of glutamic acid production. *Biotechnology and Bioengineering*, Vol.44, lk 87-94 1994
- [3] Pekka Isomursu. A Software Engineering Approach to the Development of Fuzzy Control Systems. Technical Research Centre of Finland, Espoo. 1995
- [4] Ennu Rūstern. Intelligentised regulaatorid. Loengukonspekt. Tallinna Tehnikaūlikool. 1995
- [5] Simon Haykin. Neural Networks. A Comprehensive Foundation. Maxwell MacMillan Canada Inc. 1994
- [6] A.J. Chipperfield and P.J. Fleming. MatLab® Toolboxes and Applications for Control. Peter Peregrinus Ltd., 1993
- [7] Rene Jager. Fuzzy Logic in Control. Thesis Technische Universiteit Delft, 1995
- [8] Ronald R. Yager, Dimitar P. Filev. Essentials of Fuzzy Modeling and Control. John Wiley and Sons, 1994
- [9] H. Nicholson. Modelling of Dynamical Systems Vol 1. Santype International Ltd, 1980
- [10] Ching Ten Lin, C.S. George Lee. Neural Fuzzy Systems. Prentice Hall, 1995
- [11] Yong Zai Lu. Industrial Intelligent Control. John Wiley and Sons, 1996
- [12] David M. Skapura. Building Neural Networks. Addison Wesley, 1995
- [13] Martin Brown, Chris Harris. Neurofuzzy Adaptive Modelling and Control. Prentice Hall, 1994
- [14] L.A. Zadeh, J. Kasparzyk. Fuzzy Logic for the Management of Uncertainty. John Wiley and Sons, 1992
- [15] J.-S. Roger Jang, Ned Gulley. Fuzzy Logic Toolbox User's Guide. The MathWorks Inc. 1995
- [16] K.J. Astrōm. The future of control. *Modeling, Identification and Control*, vol 15 no. 3 lk 127-134, 1994
- [17] K. Hirota. Industrial Applications of Fuzzy Technology. Springer, Tokyo, 1993
- [18] Ronald R. Yager, Dimitar P. Filev. Template based fuzzy systems modeling. *Journal of Intelligent Systems* 2, lk 39-54, 1994
- [19] B. Kosko. Fuzzy systems as universal approximators, *Proceedings IEEE International Conference on Fuzzy Systems*, San Diego, lk. 1153-1162, 1992
- [20] E.H. Mamdani, B.R. Gaines. Fuzzy Reasoning and Its Applications. Academic Press, 1981

- [21] D. Dubois, H. Prade. Possibility Theory. Pergamon Press, 1982
- [22] Modelling and Control Competition.  
<http://www.wmin.ac.uk/ICC/compete/modcomp.html>
- [23] H. Shiba et al. The searching of the optimal enzyme proportion on the solid state fermentation process by fuzzy inference learned by genetic algorithm. *Proceedings Asian Control Conference*. Tokyo, 1994
- [24] L.X. Wang. Stable adaptive fuzzy control of nonlinear systems. *IEEE Trans. Fuzzy Systems*, 1993
- [25] Toomas Paalme et al. The computer-controlled continuous culture of Escherichia Coli with smooth change of dilution rate. *Journal of Microbiological Methods* no. 24, lk 145-153, 1995
- [26] K. Tomson et al. Automatic laboratory-scale fed-batch procedure for production of recombinant proteins using inducible expression systems of Escherichia Coli. *Biotechnology Techniques* Vol. 9 No. 11, 1995
- [27] C.C. Lee. Fuzzy sets in approximate reasoning Part I: fuzzy logic controller. *IEEE Transactions on Systems, Man and Cybernetics* 20, lk. 419-435, 1990
- [28] Dimitar Filev. Fuzzy modelling of complex systems. *International Journal of Approximate Reasoning* 4, lk. 281-290, 1991
- [29] L.X. Wang and J.M. Mendel. Back-propagation fuzzy system as nonlinear dynamic system identifier. *Proceedings 1<sup>st</sup> International IEEE Conference on Fuzzy Systems*. San Diego lk. 967-972, 1992
- [30] M. Sugeno, T. Yasukawa. A fuzzy logic based approach to qualitative modelling. *IEEE Transactions on Fuzzy Systems* 1, lk. 7-31, 1993
- [31] H-J. Zimmermann. Fuzzy Set Theory and Its Applications. Kluwer, 1991
- [32] J.-S. R. Jang. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics* 23, lk. 665-685, 1993

## LISA

### Sünteesitud hägusate regulaatorite juhtimisel läbiviidud fermentatsioonide tulemused.

#### Tähised joonistel

$p$  - produkti kontsentratsioon fermentaatoris

$V$  - fermentatsioonikeskkonna maht

$x$  - biomassi kontsentratsioon fermentaatoris

$J$  - protsessi kvaliteedinäitaja

$r_1$  - substraadi 1 kontsentratsiooni seadesuurus

$r_2$  - substraadi 2 kontsentratsiooni seadesuurus

$s_1$  - substraadi 1 kontsentratsioon fermentaatoris

$s_2$  - substraadi 2 kontsentratsioon fermentaatoris

$f_1$  - substraadi 1 fermentaatorisse viimise kiirus

$f_2$  - substraadi 2 fermentaatorisse viimise kiirus

Stsenaarium C1.1 - lk. 9-5

Stsenaarium C1.2 - lk. 9-6

Stsenaarium C2.1 - lk. 9-7

Stsenaarium C2.2 - lk. 9-8

Stsenaarium C3.1- lk. 9-9

