

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**ELEKTROS IR ELEKTRONIKOS FAKULTETAS**

**Andrius Morkevičius**

**MODELIU PAGRĪSTAS BIOTECHNOLOGINIO PROCESO  
OPTIMIZAVIMAS, TAIKANT RADIALINĖS BAZĖS FUNKCIJŲ  
TINKLĄ**

Baigiamasis magistro projektas

**Vadovas**  
Prof. dr. Vytautas Galvanauskas

**KAUNAS, 2017**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**ELEKTROS IR ELEKTRONIKOS FAKULTETAS**  
**AUTOMATIKOS KATEDRA**

**MODELIU PAGRĪSTAS BIOTECHNOLOGINIO PROCESO**  
**OPTIMIZAVIMAS, TAIKANT RADIALINĖS BAZĖS FUNKCIJŲ**  
**TINKLĄ**

Baigiamasis magistro projektas  
Valdymo technologijos (kodas 621H66001)

**Vadovas**

prof. dr. Vytautas Galvanauskas  
2017-06-05

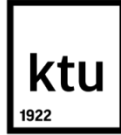
**Recenzentas**

doc. dr. Kęstutis Brazauskas  
2017-06-05

**Projektą atliko**

Andrius Morkevičius  
2017-06-05

**KAUNAS, 2017**



KAUNO TECHNOLOGIJOS UNIVERSITETAS

ELEKTROS IR ELEKTRONIKOS

(Fakultetas)

ANDRIUS MORKEVIČIUS

(Studento vardas, pavardė)

Valdymo technologijos, 621H66001

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto

„Modeliu pagrįstas biotechnologinio proceso optimizavimas, taikant radialinės bazės funkcijų tinklą“

#### AKADEMINIO SAŽININGUMO DEKLARACIJA

20 17 m. birželio 5 d.  
Kaunas

Patvirtinu, kad mano **Andriaus Morkevičiaus** baigiamasis projektas tema „Modeliu pagrįstas biotechnologinio proceso optimizavimas, taikant radialinės bazės funkcijų tinklą“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

\_\_\_\_\_  
(vardą ir pavardę įrašyti ranka)

\_\_\_\_\_  
(parašas)

Morkevičius, Andrius. Modeliu pagrįstas biotechnologinio proceso optimizavimas taikant radialinės bazės funkcijų tinklą. Valdymo sistemų magistro baigiamasis projektas / vadovas prof. dr. Vytautas Galvanauskas; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, Automatikos katedra.

Mokslo kryptis ir sritis: Elektros ir elektronikos inžinerija, Technologiniai mokslai

Reikšminiai žodžiai: biotechnologija, optimizavimas, radialinė, bazė, neuroninis, tinklas

Kaunas, 2017. 52 p.

## SANTRAUKA

Darbo tikslas – naudojant radialinės bazės funkcijų neuroninį tinklą optimizuoti periodinį su pamaitinimu biotechnologinį procesą.

Darbe analizuojamas biotechnologinių procesų modeliavimas taikant medžiagų masės balansą bei technologinių parametrų optimizavimas, dirbtinių neuroninių tinklų struktūra, jų veikimas ir pritaikymas. Analizuojamas straipsnyje aprašytas periodinis su pamaitinimu biotechnologinis procesas. Darbe buvo sudarytas periodinio su pamaitinimu biotechnologinio proceso imitacinis modelis ir gauti modeliavimo rezultatai, kurių metu naudojamas Pontriagino maksimumo principu apskaičiuotas optimalus pamaitinimo profilis. Optimalus pamaitinimo profilis pakeičiamas į radialinės bazės funkcijų neuroninį tinklą. Šešiais skirtingais atvejais atliekamas neuroninio tinklo optimizavimas taikant evoliucinį optimizavimo metodą, kurio tikslo funkcija – maksimalus galutinio produkto (baltymų) kiekis biotechnologinio proceso pabaigoje.

Visais nagrinėtais optimizavimo atvejais gauti būsenos kintamųjų  $x$ ,  $s$  ir  $p_x$  trajektorijų nuokrypiai galiniuose taškuose, palyginti su referenciniais duomenimis, yra mažesni už realiame procese matuojamų biomasės ir substrato koncentracijų bei tikslinio baltymo aktyvumo matavimo tikslumą (~10%). Tyrimo rezultatai demonstruoja, kad taikant parametrinio optimizavimo metodiką ir radialinės bazės funkcijų tinklą nagrinėtame biotechnologiame procese gaunami optimizavimo rezultatai, analogiški referenciniams duomenims. Taip dar kartą patvirtinamas abiejų metodikų rezultatų teisingumas ir atitikimas eksperimentiškai gautiems duomenims.

Morkevičius, Andrius. Model-Based Optimization of Biotechnological Process Using Radial Basis Function Network: Master's thesis in control systems degree / supervisor assoc. prof. dr. Vytautas Galvanauskas. Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, department of automation

Research area and field: Electrical and Electronics Engineering, Technological Sciences

Key words: biotechnology, optimization, radial, basis, neural, network

Kaunas, 2017. 52 p.

## SUMMARY

Project's goal is to optimize fed-batch biotechnological process using radial basis function network.

The work examines modeling of mass balance based biotechnological processes, optimization of technological parameters, structure, application and operation of artificial neural networks. Paper on fed-batch biotechnological process is analyzed. The work consists of fed-batch process imitation model where Pontriagin's principle of maximum is used to calculate optimal feeding profile. Later optimal feeding profile is substituted with approximation of radial basis function neural network. Neural network is optimized in six different cases, using evolutionary optimization method. Methods goal function is to maximize the amount of product (protein) at the end of cultivation process.

In all of the cases, state variables  $x$ ,  $s$  and  $p_x$  deviations of trajectories in process ending points, in comparison with reference data, are less than 10% which is the accuracy of measurements in real biotechnological process of biomass, substrate concentration and specific protein activity. Research results show that using parametric optimization method and radial basis function network, optimization results are analogical to reference data. This confirms that results obtained from both methodologies match experimental data and are correct.

# Turinys

Įvadas .....	7
1. Biotechnologija .....	8
2. Biotechnologinių procesų modeliavimas .....	9
2.1 Medžiagų masės balansas.....	9
3. Biotechnologinių procesų optimizavimas .....	11
3.1 Technologinių parametrų laiko profilių optimizavimas .....	12
3.2 Pontriagino maksimumo principo taikymas periodiniams fermentacijos procesams optimizuoti .....	13
3.3 Pontriagino maksimumo principo taikymas periodinių su pamaitinimu fermentacijos procesų optimizavimui.....	14
3.4 Parametrinės optimizacijos metodika.....	16
4. Dirbtiniai neuroniniai tinklai .....	19
4.1 Radialinės bazės funkcijos (RBF) neuroninis tinklas.....	20
5. Biotechnologinio proceso modeliavimas Matlab aplinkoje. ....	22
5.1 Modelio aprašymas.....	23
5.3 Galutinio produkto kiekio optimizavimas .....	28
5.4 Evoliucis optimizavimo metodas .....	30
5.5 Optimizavimo rezultatai .....	31
6.5.1 Pirmasis atvejis.....	32
5.5.2 Antrasis atvejis .....	33
5.5.3 Trečiasis atvejis. ....	35
5.5.4 Ketvirtasis atvejis. ....	36
5.5.5 Penktasis atvejis. ....	37
5.5.6 Šeštasis atvejis.....	39
5.3 Rezultatų palyginimas .....	40
Įšvados ir rezultatai .....	42
Literatūra .....	44
Priedai.....	46
Priedas 1. Pagrindinė programa „HW_0” .....	46
Priedas 2. Evoliucinio optimizavimo programa „rvo05” .....	46
Priedas 3. Biotechnologinio proceso modeliavimo programa „model” .....	50
Priedas 4. Baltymų kiekio modeliavimo programa „model_J” .....	51

## **Įvadas**

Biotechnologija - tai mokslas apie manipuliavimą biologiniais organizmais siekiant užtikrinti ir palengvinti žmogaus gyvenimą. Biotechnologija apima tokias įvairias sritis kaip maisto pramonė, atliekų perdirbimą, metalų gavybą, mediciną ir kitas. Nors biotechnologiniai procesai yra labai svarbūs, tačiau juos dažnai yra sunku pilnai suprasti fundamentaliai bei aprašyti juos matematinėmis lygtimis, todėl biotechnologinių procesų modeliavimui gali būti pasitelkiamos alternatyvos – juodos dėžės modeliai, neraiškių aibių logika arba dirbtiniai neuroniniai tinklai. Dirbtiniai neuroniniai tinklai šiuo metu yra sparčiai besivystanti skaitmeninio intelekto kūrimo technika. Šie tinklai apima informacijos apdorojimo metodus, kurie sukurti stengiantis imituoti gyvūnų smegenyse vykstančius procesus. Dirbtinis neuroninis tinklas yra matematinių modelių rinkinys, kuriuo bandoma imituoti biologinių sistemų savybes mokytis, prisitaikyti ir adaptuotis, todėl yra dažnai naudojamas mažai žinomiems procesams modeliuoti bei valdyti. Šių tinklų pagrindinis privalumas yra tai, kad jie gali būti apmokomi, taip sumažinant aproksimavimo paklaidas tarp realių eksperimentinių duomenų ir modelio rezultatų. Šiame darbe buvo apžvelgtas biotechnologinių procesų optimizavimas, optimizavimo metodų taikymas, biotechnologinių procesų modeliavimas, ir dirbtinių neuroninių tinklų sandara, veikimo principai, taikymo sritys. Sumodeliuotas biotechnologinis procesas ir optimizuotas taikant radialinės bazės funkcijų tinklą.

# 1. Biotechnologija

Biotechnologija – tai mokslas apie biologinių organizmų manipuliavimą, siekiant palengvinti žmogaus gyvenimą įvairiose jo gyvenimo srityse. Biotechnologija dar gali būti apibūdinama kaip mokslinių, inžinerinių principų taikymas biologinių organizmų gaminamoms medžiagoms gauti ir panaudoti pramonėje ir buityje. Biotechnologija apima įvairias sritis nuo maisto pramonės, medicinos iki atliekų perdirbimo ar kalnakasybos. Medicinoje biotechnologija yra labai sparčiai auganti mokslo sritis, kadangi biotechnologijos pagrindu pagaminti vaistai yra vieni iš veiksmingiausių gydant genetines, autoimunines, onkologines ligas. Tačiau, biotechnologinių vaistų gaminimas yra labai sudėtingas, kadangi net ir menkas nukrypimas nuo technologinio režimo gali pakeisti vaisto saugumą ir veiksmingumą, todėl jie yra gaminami griežtai kontroliuojamomis sąlygomis.

Manoma, kad biotechnologija jau buvo pradėta naudoti 6000 metų prieš mūsų erą, kai buvo panaudotos mielės alaus gaminimui. 4000 metų prieš mūsų erą Egipte mielės taip buvo naudojamos raugintai duonai gauti.

Pramoninis biotechnologijos procesas naudojantis mikroorganizmus komercinių produktų gamyboje vyksta pagal tokią schemą [17]: Žaliava → Pirminis apdorojimas → Fermentacija ir transformacija → Antrinis apdirbimas → Gautas produktas.

1. Pirminio apdorojimo metu žaliavos paruošiamos kaip pirminis maisto šaltinis mikroorganizmams
2. Fermentacijos ir transformacijos metu bioreaktoriuose išauginami reikalingas medžiagas gaminantys mikroorganizmai. Šie mikroorganizmai gali būti: antibiotikai, amino rūgštys ar įvairūs baltymai. Mikroorganizmų gaminamų reikalingų medžiagų kiekio didinimas yra vienas iš svarbiausių optimizavimo uždavinių.
3. Antrinio apdirbimo metu pagamintos tikslinės medžiagos yra išgryninamos iš bendros kultūros masės

Visi organizmai auga atitinkamomis temperatūrų sąlygomis. Aukštose temperatūrose prasideda ląstelės baltymų denatūracija ir kitų svarbių ląstelės sistemų bei komponentų irimo procesai vedantys į ląstelės žuvimą. Žemose temperatūrose gyvybiškai svarbūs ląstelės procesai dėl ląstelės baltymų struktūrinių pokyčių taip pat sustoja. Pagal optimalią augimo ir temperatūrą ląstelės yra klasifikuojamos į: termofilus (45 - 90°C), mezofilus (10 - 47°C) ir psichrofilus (-5 - -35°C). Mikroorganizmai augantys skirtingose temperatūrose naudojami specifinių biotechnologijos tikslų pasiekimui.



## 2. Biotechnologinių procesų modeliavimas

### 2.1 Medžiagų masės balansas

Atliekant biotechnologinių procesų modeliavimą daugeliu atvejų reikalinga apsisrašyti medžiagų masės balanso ir energijos srautų balanso lygtis.

Medžiagos masės lygtys paremtos medžiagų tvermės dėsnium, kuris šiuo atveju reiškia, kad tam tikros medžiagos kiekio kitimas reaktoriuje gali kisti dėl [1]:

- Reaktoriuje vykstančių cheminių ar biocheminių reakcijų, kuomet medžiaga skyla arba dalijasi;
- Dėl išorinio medžiagų tiekimo į bioreaktorių;
- Iš bioreaktoriaus pašalinamo kultivavimo terpės kiekio.

$$\frac{d(C_1V)}{dt} = q_1C_2V + C_{in1}F_{in} - C_1F_{out} \quad (1)$$

Kairioji lygybės pusė nusako C1 medžiagos kiekio kitimą laike [g/h].  $q_1$  – santykinis reakcijos greitis medžiagos C2 atžvilgiu [gC1/gC2h].  $C_{in1}$  – įtekančios medžiagos koncentracija (pvz. pamaitinant) [g/l].  $F_{in}$  – įtekančios medžiagos kiekio srautas [l/h].  $C_1$  – bioreaktoriuje esančios medžiagos koncentracija prieš pašalinant (pvz. imant mėginius) [g/l].  $F_{out}$  – pašalinamos medžiagos kiekio srautas [l/h]

Tirtame straipsnyje [6] buvo sudarytas periodinio su pamaitinimu biotechnologinio proceso matematinis modelis, paremtas masių balansu. Masės balansai buvo sudaryti: dalelės biomasei, pamaitinimo substratui bei kultivuojamos kultūros tūriui modeliuoti:

$$\frac{dx}{dt} = \mu x - \frac{F_s}{V} x \quad (2)$$

$$\frac{ds_i}{dt} = -q_{si} - \frac{F_s}{V} s_i + \frac{F_s}{V} s_{i,f}, i = 1,2,3 \quad (3)$$

$$\frac{dV}{dt} = F_s - F_{s\text{amp}} \quad (4)$$

čia,  $x$ ,  $s_1$ ,  $s_2$ ,  $s_3$  yra biomasės, saulėgrąžų aliejaus, kazeino hidrolizatas, mielių ekstraktas;  $s_{1,f}$ ,  $s_{2,f}$ ,  $s_{3,f}$ , yra substrato komponentų koncentracijos pamaitinimo tirpale,  $\mu$  – tikslinis biomasės augimo greitis,  $q_{si}$  – tikslinis substrato suvartojimo greitis,  $V$  – kultivavimo tūris,  $F_s$  – substrato pamaitinimo greitis (valdantysis poveikis),  $F_{s\text{amp}}$  – mėginių paėmimo greitis.

Straipsnyje buvo siekiama identifikuoti tinkamą funkcinės priklausomybės struktūrą tiksliniam biomasės augimo greičiui  $\mu$  (2), todėl kelios skirtingos kinetinės priklausomybės buvo iširtos [5, 7]:

$$\mu(x, s_1) = \mu_{max} \frac{s_1}{s_1 + xK_{xx}} \quad (5)$$

$$\mu(x, s_1) = \mu_{max} \frac{s_1}{s_1 + K_s} \frac{K_{xx}}{x + K_{xx}} \quad (6)$$

$$\mu(x, s_1) = \mu_{max} \frac{s_1}{s_1 + K_s} \frac{K_i}{s_1 + K_i} - K_{xx}x \quad (7)$$

kur  $\mu_{max}$ ,  $K_s$ ,  $K_i$  ir  $K_{xx}$  atitinkamai yra modelio parametrai apibūdinantys maksimalų tikslinį augimo greitį, Mono konstanta, substrato inhibicijos konstanta ir biomasės inhibicijos konstanta.

Taip pat yra daroma prielaida tarp biomasės augimo ir tikslinio substrato sunaudojimo greičio  $q_{si}$ :

$$q_{s1} = \frac{1}{Y_1} \mu + m \quad (8)$$

$$q_{s2} = \frac{1}{Y_2} \mu \quad (9)$$

$$q_{s3} = \frac{1}{Y_3} \mu \quad (10)$$

čia  $Y_1$ ,  $Y_2$ ,  $Y_3$  yra išeigos koeficientai.

Straipsnyje taip pat buvo atlikta eksperimentinių duomenų analizė, kuri parodė, kad uždelsta fazė, kuomet vyksta mikroorganizmų adaptacija prie aplinkos, yra labai svarbi, palyginus su viso proceso laiku vykstančiais procesais. Šis fenomenas yra apibūdinamas matematiname modelyje įterpiant išorinį faktorių  $\alpha(t)$  kartu su funkcinėmis priklausomybėmis  $\mu(x, s)$ .

$$\alpha(t) = \frac{\arctg\left(K_p \left(\frac{t - t_0}{\tau}\right)\right)}{\pi} + 0,5 \quad (11)$$

$t$  – proceso trukmė,  $K_p$ ,  $t_0$ , ir  $\tau$  yra modelio parametrai, kurie bus identifikuojami.

Taigi tikslinis augimo greitis  $\mu(x, s)$  lygtyje (2) yra aprašomas tokia priklausomybe:

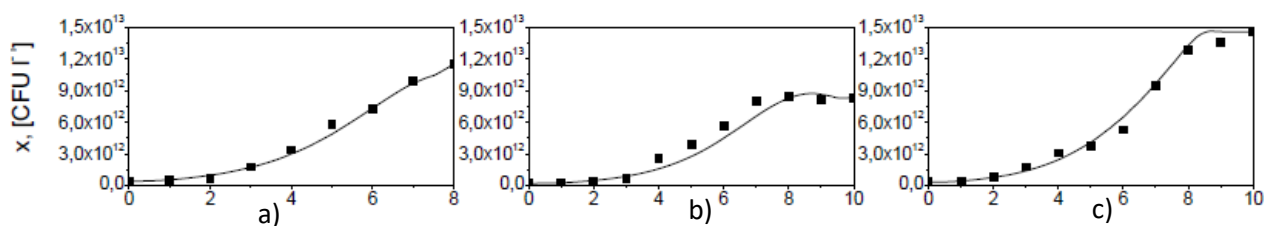
$$\mu(x, s) = \mu(x, s) \cdot \alpha(t) \quad (12)$$

Ši lygtis bus galiausiai išrinkta iš (5) – (7) lygties pagal modelio identifikavimo kokybę kiekvienai lygčiai.

Toliau straipsnyje yra atliekamas parametų identifikavimas pasitelkiant evoliucinio programavimo metodą, kuriuo siekiama gauti mažiausią kvadratinę paklaidą tarp modeliavimo rezultatų ir eksperimentinių duomenų.

Ištyrus tikslo funkcijos vertes apskaičiuotas atliekant modelio parametų identifikavimą naudojant skirtingas lygtis (5) – (7), kurios apibūdina tikslinį augimo greitį, buvo nustatyta, kad geriausi eksperimentinių duomenų aproksimavimo rezultatai gaunami kuomet naudojama (7) lygtis.

Modelio parametų identifikavimo rezultatai pateikti 2.1 pav.



2.1 pav. Eksperimentinių duomenų aproksimavimo rezultatai. a) procesas atliktas mėgintuvėlyje b) ir c) bioreaktoriuje. Taškais pavaizduoti eksperimentiniai duomenys, linijomis matematinio modeliavimo rezultatai.

Iš 2.1 pav. Galima matyti, kad pasitelkiant medžiagų masės balanso dėsnius biotechnologiniams procesams modeliuoti, galima gauti labai artimus rezultatus eksperimentiniams duomenims, gautiems atliekant tikrą biotechnologinį procesą.

### 3. Biotechnologinių procesų optimizavimas

Biotechnologiniuose procesuose dažniausia naudojami trys pagrindiniai bioreaktorių tipai, kuriuose kultivuojami mikroorganizmai: periodinio veikimo, periodinio veikimo su pamaitinimu ir nenutrūkstamo veikimo. Šiems procesams bendras optimizavimo uždavinys yra mitybinių terpių optimizavimas, tačiau kiekvienam iš šių procesų yra būdingi specifiniai optimizavimo uždaviniai.

Periodiniuose bei periodiniuose su pamaitinimu procesuose vyksta nenutrūkstamas mikroorganizmų kultūros būsenos kitimas, o procesų optimizavimo tikslas yra nustatyti optimalias manipuluojamų technologinių parametru laiko trajektorijas, kurios galėtų maksimizuoti ar minimizuoti pasirinktą proceso kokybės rodiklį.

Dinaminės ir statinės optimizacijos uždaviniams spręsti naudojami matematiniai optimizavimo metodai, kurie remiasi optimizuojamų procesų matematiniais modeliais.

Prieš optimizuojant procesą reikalinga parinkti optimizavimo kriterijų, kurie gali būti skirstomi į technologinius ir ekonominius kriterijus. Technologiniu kriterijumi gali būti parenkamas maksimalus tikslinio produkto kiekis, maksimalus produktyvumas, minimalus laikas ir panašūs parametrai. Prie ekonominių kriterijų priskiriamas produkto savikainos mažinimas ar pelno didinimas.

Remiantis optimizavimo kriterijumi, sudaroma tikslo funkcija, kuri nurodo optimizuojamo parametro priklausomybę nuo proceso parametru, būsenos kintamųjų ar valdančių poveikių. Bendru atveju tikslo funkcija aprašoma taip [1]:

$$J = f(x) \rightarrow \max(\min) \quad (13)$$

Optimizuojant dinامينius uždavinius ieškoma valdančiųjų poveikių  $u(t)$  laiko trajektorija, kurios suteikia tikslo funkcijai ekstreminę vertę proceso pabaigoje. Tokių uždavinių sprendimui taikomi matematiniai modeliai, kuriuos sudaro proceso būsenos lygtys:

$$\frac{dx}{dt} = f(x, u) \quad (14)$$

Būsenos lygtys sudaromos remiantis medžiaginio balanso sąlygomis. Tam, kad optimizuojant procesą būtų išlaikomos realios techninės ir fizikinės galimybės taikomi tam tikri techniniai ir technologiniai apribojimai pamaitinimo greičiui, kultūrinės terpės tūriui bei kitiems proceso kintamiesiems.

### 3.1 Technologinių parametrų laiko profilių optimizavimas

Norint gauti geresnius biotechnologinio proceso rezultatus, technologinius parametrus, kurie proceso metu palaikomi pastovūs, galima keisti. Todėl aktualus periodinių fermentacijos procesų optimizavimo uždavinys – rasti optimalias technologinių parametrų keitimo laiko trajektorijas vadinamas laiko profiliais [1]. Radus optimalias laiko trajektorijas galima pasiekti maksimalią produkto išėigą, proceso produktyvumą bei pagerinti kitus technologinius parametrus.

Laiko profilių optimizavimo uždaviniai dažniausiai sprendžiami pagal pateiktą metodiką:

1. Sudaromas proceso modelis, kuris nustato funkcinę ryši tarp optimizuojamų parametrų ir proceso būsenos kintamųjų. Kadangi vidiniai modeliuojamų priklausomybių mechanizmai praktiškai nėra žinomi, todėl modeliuose jie paprastai būna išreiškiami parenkant tipinę modelio struktūrą ir nustatant pasirinktos struktūros parametrų priklausomybę nuo optimizuojamų paramterų:

$$\frac{dx}{dt} = f(x, \mathbf{K}(u)) \quad (15)$$

Modeliui identifikuoti atliekami eksperimentai, kurių metu yra nustatoma optimizuojamų parametrų įtaka proceso išėigai. Eksperimento metu optimizuojami parametrai  $u$  stabilizuojami tam tikru lygiu. Atlikus eksperimentą gaunamos biomases augimo kreivės, atitinkančios eksperimentuose fiksuotas optimizuojamų parametrų vertes. Pagal gautus rezultatus identifikuojamos pasirinktos struktūros modelio parametrų  $\mathbf{K}$  vertės, o iš jų nustatomos funkcinės priklausomybės  $\mathbf{K}(u)$ .

2. Optimalios laiko trajektorijos ieškomos pasitelkiant matematinius optimizavimo metodus. Dažniausiai taikomi metodai yra Pontriagino maksimumo principas ar parametrinės optimizacijos metodai.

### 3.2 Pontriagino maksimumo principo taikymas periodiniams fermentacijos procesams optimizuoti

Maksimumo principas – tai visuma teoremų, nustatančių būtinas optimalumo sąlygas, kurias turi tenkinti optimalaus valdymo algoritmas [1]. Sudarius fermentacijos proceso modelį

$$\frac{dx}{dt} = f(x, K(u)) \quad (16)$$

ir parinkus tikslo funkciją

$$J = c^T x(t_g) \rightarrow \max \quad (17)$$

$c$  yra koeficientų vektorius. Technologinių parametrų laiko profilių optimizavimo uždavinys sprendžiamas tokiais etapais:

1. Sudaroma Hamiltono funkcija:

$$H = \lambda^T f(x, K(u)) \quad (18)$$

čia  $\lambda$  yra papildomų kintamųjų vektorius. Ryšys tarp pagrindinių ir papildomų kintamųjų  $x$  nustatomas per Hamiltono funkciją:

$$\frac{d\lambda}{dt} = -\frac{\partial H}{\partial x} \quad (19)$$

2. Nustatomos papildomos kraštinės sąlygos, naudojamos pagrindinių ir jungtinių kintamųjų lygčių sistemoms spręsti, ir sąlygos, įvertinančios Hamiltono funkcijos savybes. Šios sąlygos priklauso nuo tikslo funkcijos nustatymo būdo, pagrindinių kintamųjų lygčių sistemos kraštinių sąlygų bei uždavinio tipo: su fiksuotu ar nefiksuotu laiku.

Jei būsenos kintamieji proceso pabaigoje gali įgyti bet kokias vertes (yra nefiksuotos), tai jungtinių kintamųjų vertės proceso pabaigoje nustatomos iš sąlygos:

$$\lambda(t_k) = c \quad (20)$$

Jeigu proceso trukmė ( $t_g - t_0$ ) nefiksuota, tai  $H \equiv 0$ , jei fiksuota, tai  $H \equiv const$ .

3. Optimalūs technologinių parametrų laiko profiliai nustatomi remiantis sąlyga:

$$H(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{K}(\mathbf{u}_{opt})) = \max_{\substack{\mathbf{u} \in \mathbf{U} \\ t_0 \leq t \leq t_g}} H(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{K}(\mathbf{u})) \quad (21)$$

Tikslo funkciją maksimizuoja tokie valdymo poveikiai, kurie visame valdymo intervale suteikia Hamiltono funkcijai maksimalią vertę.

Kuomet optimizuojamo parametro kitimo diapazoną riboja technologiniai ar kiti apribojimai:

$$\mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max} \quad (22)$$

tuomet optimalų valdymo poveikį galima nustatyti remiantis sąlyga:

$$\frac{\partial H(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{K}(\mathbf{u}_{opt}))}{\partial \mathbf{u}} = 0, \text{ kai } \mathbf{u}_{min} < \mathbf{u}_{opt}(t) \leq \mathbf{u}_{max} \quad (23)$$

Kitais atvejais optimalios ribinės valdymo poveikių vertės  $\mathbf{u}_{min}$  arba  $\mathbf{u}_{max}$ .

Tokio tipo uždaviniai dažniausiai sprendžiami tik skaitmeniniais iteraciniais metodais. Maksimumo principo būtinomis optimalumo sąlygomis paremti skaitmeniniai optimalių trajektorijų paieškos metodai realizuojami dviem būdais:

1. Pagrindinių ir papildomų kintamųjų lygtys integruojamos tiesiogine ir atbuline kryptimis, naudojant nustatytas kraštines sąlygas ir integravimo metu nustatant optimizuojamų parametrų trajektorijas, maksimizuojančias Hamiltono funkciją. Paieškos procedūra baigiama, kai po tiesioginio integravimo apskaičiuotos papildomų kintamųjų vertės proceso pabaigoje sutampa su nustatytais iš būtinų optimalumo sąlygų.

2. Pagrindinių ir papildomų kintamųjų lygtys integruojamos viena kryptimi, integravimo metu nustatant optimizuojamų parametrų trajektorijas, maksimizuojančias Hamiltono funkciją, ir kiekvienos iteracijos metu koreguojant papildomų kintamųjų pradines sąlygas, kol apskaičiuotos papildomų kintamųjų vertės proceso pabaigoje sutampa su nustatytais iš būtinų optimalumo sąlygų.

### 3.3 Pontriagino maksimumo principo taikymas periodinių su pamaitinimu fermentacijos procesų optimizavimui

Bioreaktorius, kuriame vyksta periodinis su pamaitinimu biotechnologinis procesas suteikia svarbų pranašumą prieš periodinio veikimo bioreaktorius ir yra dažnai naudojamas fermentavimo technologijose [8]. Periodiniuose su pamaitinimu fermentacijos procesuose substrato koncentracija gali būti išoriškai valdovą manipuliuojant pamaitinimo greičiu. Taigi, nuo substrato koncentracijos

priklausomiems mikroorganizmams, bet kuriuo biotechnologinio proceso metu gali būti palaikoma biosintezei optimali substrato koncentracija pasitelkiant pamaitinimo greičio laiko trajektorijas. Šiuo metu pamaitinimo greičio optimizavimo problemos dažniausiai sprendžiamos matematiniais modeliu paremtais optimizavimo metodais.

Per kelis dešimtmečius buvo sukurti keli optimizavimo metodai ir algoritmai skirti optimalaus pamaitinimo profilio radimui. Matematiniais modeliams, kuriuos sudaro 3 – 4 netiesinės diferencialinės lygtis, gali būti taikomas Pontriagino maksimumo principu paremtas optimizavimo metodas [6, 9].

Formuluojant maksimumo principo būtinas optimalumo sąlygas, tikslo funkcija užrašoma integralinio funkcionalo forma [1]:

$$J = p(t_g), V(t_g) = \int_{t_0}^{t_g} q_p(s, x, p, \dots) x V dt \rightarrow \max. \quad (24)$$

Sudaroma Hamiltono funkcija:

$$H = q_p x V + \lambda_x \left( \mu x - u \frac{x}{V} \right) + \lambda_s \left( -q_s x + u \frac{s_p - s}{V} \right) + \lambda_p \left( q_p x - u \frac{p}{V} \right) + \lambda_V u \quad (25)$$

Iš lygties matyti, kad valdantis poveikis į modelio lygtis ir Hamiltono funkciją turi tiesinį poveikį. Sugrupavus funkcijos narius prie valdančio poveikio, gaunama:

$$H = \Psi(s, x, p, \lambda_x, \lambda_p) + u \Phi(s, x, p, V, \lambda_s, \lambda_x, \lambda_p, \lambda_V) \quad (26)$$

$$\Psi = q_p x V + \lambda_x \mu x - \lambda_s q_s x + \lambda_p q_p x \quad (27)$$

čia

$$\Phi = -\lambda_x \frac{x}{V} + \lambda_s \frac{s_p - s}{V} - \lambda_p \frac{p}{V} + \lambda_V \quad (28)$$

Ryšys tarp papildomų kintamųjų  $\lambda$  ir pagrindinių kintamųjų  $s, x, p, V$  nustatomas per Hamiltono funkciją:

$$\frac{d\lambda}{dt} = -\frac{\partial H}{\partial x} \quad (29)$$

Toliau nustatomos papildomos kraštinės sąlygos, naudojamos pagrindinių ir jungtinių kintamųjų lygčių sistemoms spręsti. Taip pat sudaromos ir sąlygos, įvertinančios Hamiltono funkcijos savybes.

Jeigu proceso trukmė ( $t_g - t_0$ ) nefiksuota, tai  $H \equiv 0$ , jei fiksuota, tai  $H \equiv \text{const}$ . Jei būsenos kintamieji proceso pabaigoje gali įgyti bet kurias vertes, tai  $\lambda(t_g) = 0$

Optimalūs technologinių parametrų laiko profiliai nustatomi iš sąlygos:

$$H(s, x, p, V, \lambda, u_{opt}) = \max_{\substack{0 \leq u \leq u_{max} \\ t_0 \leq t \leq t_g}} H(s, x, p, V, \lambda, u) \quad (30)$$

Pamaitinimo greičio varijavimo diapazoną riboja techniniai apribojimai:

$$0 \leq u(t) \leq u_{max} \quad (31)$$

Optimalus pamaitinimo greitis, remiantis būtinomis optimalumo sąlygomis, nustatomas taip:

$$u(t) = \begin{cases} u_{max}, & \text{kai } \Phi > 0, \\ u_{ypat.}, & \text{kai } \Phi = 0, \\ 0, & \text{kai } \Phi < 0 \end{cases} \quad (32)$$

čia  $u_{ypat.}$  - valdymo poveikis, optimalaus valdymo teorijoje vadinamas ypatingu valdymu. Šio tipo valdymas nustato optimalų pamaitinimo greitį.

Ypatingam valdymui apskaičiuoti naudojamas perjungimo funkcijos  $\Phi$  pakartotinis diferencijavimas pagal laiką:

$$\Phi = 0, \frac{d^k \Phi}{dt^k} = 0, k = 1, 2, 3, \dots \quad (33)$$

Iš šių funkcinių priklausomybių nustatoma pamaitinimo greičio priklausomybė nuo pagrindinių ir papildomų kintamųjų.

### 3.4 Parametrinės optimizacijos metodika

Pasitelkus parametrinės optimizacijos metodiką ir stochastinės optimizacijos algoritmus, dinaminės optimizacijos uždavinys formuluojamas kaip statinės optimizacijos uždavinys, kuriam spręsti gali būti taikomi gerai žinomi netiesinio programavimo ar stochastinės paieškos algoritmai. Skaičiavimo metu yra optimizuojami tam tikros struktūros laiko funkcijų parametrai. Optimizuotai trajektorijai aprašyti iš anksto parenkama universalios struktūros laiko funkcija [1]:

$$u = y(a_0, \dots, a_1, t) \quad (34)$$

Keičiant šios funkcijos parametrus  $a_0, \dots, a_1$ , galima aprašyti bet kokios formos laiko trajektoriją. Optimizuojant randami tokie laiko funkcijos parametrai, kuriais gaunama optimali aproksimuojamos trajektorijos tikslo funkcijos vertė:

$$J = \max_{a_0, \dots, a_1} f(x(t_g)) \quad (35)$$



Jeigu optimalių laiko trajektorijų pobūdis žinomas iš anksto, dažnai galima parinkti tinkamą paprastos struktūros laiko funkciją, kuri, naudojant minimalų optimizuojamų parametru skaičių, gali aproksimuoti optimalią manipuluojamo kintamojo trajektoriją.

Dažnai optimalių trajektorijų pobūdis nėra žinomas, todėl yra parenkama universalios struktūros laiko funkcija su plačiomis aproksimavimo galimybėmis. Tam yra naudojamos polinominės funkcijos ir dirbtiniai neuroniniai tinklai.

Geromis aproksimavimo galimybėmis pasižymi netiesinė funkcija:

$$y = \frac{A}{1 + \exp(\sum_{i=0}^n a_i t^i)} \quad (36)$$

Taip pat gali būti naudojama paprastesnė eksponentinė funkcija:

$$y = A \exp\left(\sum_{i=0}^n a_i t^i\right) \quad (37)$$

Ar paprastas laipsninis polinomas:

$$y = \sum_{i=0}^n a_i t^i \quad (38)$$

čia  $A$ ,  $a_i$  - optimizavimo procedūros metu nustatyti parametrai.

Sudėtingiems modeliams, kuriuose naudojamas didelis kiekis būsenos kintamųjų lygčių ir hibridiniams modeliams, kurie apjungia matematinius modelius su dirbtiniais neuroniniais tinklais, parametrinio optimizavimo metodai yra dažniausiai naudojami dėl galimybių pritaikyti gerai išstobulintus netiesinio programavimo ir stochastinius paieškos algoritmus [10, 11]. Stochastinių paieškos algoritmų pritaikymas biotechnologinio proceso pamaitinimo profilių optimizavimui yra patrauklus ir dėl to, kad jo skaičiavimo algoritmo sukūrimas reikalauja mažiau pastangų ir laiko, nei alternatyvūs metodai.

Analizuotame straipsnyje [12] aprašoma substrato pamaitinimo greičio optimizavimo problema, kuri yra susijusi su sudėtingu riebalų atliekų utilizavimu. Riebalų atliekų utilizavimui periodinio su pamaitinimu biotechnologinio proceso eigoje yra kultivuojami mikroorganizmų kultūra. Optimizavimo tikslas – maksimizuoti biomasės kiekį periodinio su pamaitinimu proceso pabaigoje. Šios optimizavimo problemos sprendimui, reikalinga sudaryti proceso matematinį modelį, tačiau substrato organinių komponentų koncentracijos negali būti išmatuotos tiesioginiu būdu, dėl to matematinio modelio parametru identifikavimas yra apsunkinamas. Todėl, optimizavimo uždavinys

yra sprendžiamas panaudojant atsitiktinį paieškos algoritmą, kuris pritaikomas pamaitinimo laiko profiliui optimizuoti, taikant radialinės bazės funkcijų tinklą.

Modeliu paremto optimizavimo uždavinys sprendžiamas dvejais žingsniais. Pirmuoju žingsniu pamaitinimo profilis yra apskaičiuojamas remiantis preliminaromis prielaidomis apie technologines optimalaus proceso būsenas. Antruoju žingsniu, pirmame etape apskaičiuotas pamaitinimo profilis yra optimizuojamas naudojant parametrinio optimizavimo metodu.

Pirmiausia, norint išspręsti dinaminės optimizacijos uždavinį, straipsnyje yra sukuriamas proceso modelis paremtas masių balansu. Norint identifikuoti matematinio modelio parametrus, su kuriais modelis galėtų nuspėti proceso elgseną esant įvairioms darbo sąlygoms, buvo naudojami eksperimentiniai duomenys gauti iš kelių atliktų periodinio su pamaitinimų procesų mėgintuvėliuose.

Modelio parametrų identifikavimo procedūra yra atliekama naudojant skaitmeninį paieškos algoritmą, paremtą evoliucinio programavimo metodu [13], kuris apskaičiuoja netiesinio dinaminio modelio parametrų vertes. Identifikavimo procedūros metu ieškomos tokios modelio parametrų vektoriaus vertės, su kuriomis minimizuojama vidutinė kvadratinė paklaida tarp eksperimentinių verčių ir modeliavimo rezultatų [14]. Parametrų identifikavimo procedūros tikslo funkcija:

$$J(P) = \sqrt{\frac{\sum_{j=1}^k \sum_{i=1}^n w_{ij} (y_{ij}(P) - y_{e,ij})^2}{kn}} \rightarrow \min \quad (39)$$

$w_{ij}$  – svoriniai koeficientai,  $i$  – eksperimentų indeksas,  $j$  – išmatuoto kintamojo indeksas,  $n$  – eksperimentų skaičius,  $k$  – ismatuotų kintamųjų skaičius.

Svorinių koeficientų  $w_{ij}$  vertės yra pasirenkamos empyriškai atsižvelgiant į kiekvieno iš kintamųjų kitimo ribas, patikimumą ir įtaką identifikavimo uždaviniui. Skirtingų eksperimentų svarba identifikavimo procedūrai taip pat yra įvertinama svoriniais koeficientais.

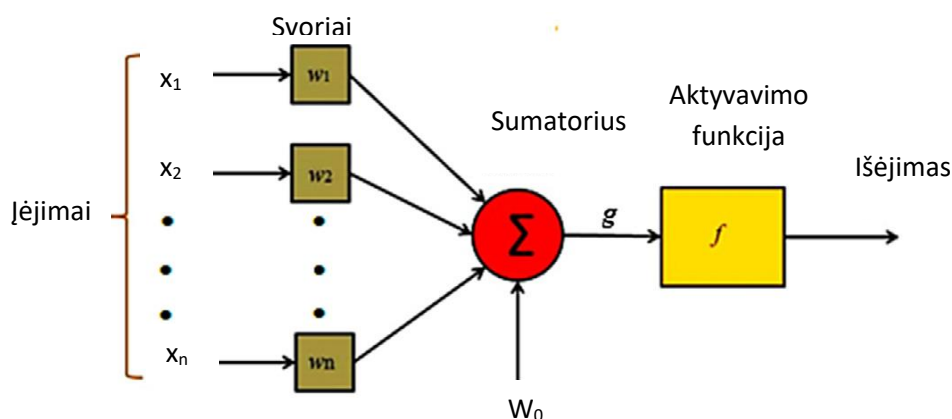
Identifikavus parametrus, straipsnyje toliau prieinama prie galutinio produkto kiekio optimizavimo. Pamaitinimo profilio aproksimacijai pasitelkiamas radialinės bazės funkcijų tinklas. Optimizavimo uždavinys transformuojamas į aproksimuojamosios funkcijos parametrų verčių suradimą, kurios nustato maksimalų kokybės indeksą atitinkantį pamaitinimo profilį. Straipsnyje naudojamas neuroninis tinklas, kurį sudaro 10 funkcijų.

Modeliavimo rezultatai parodė, kad optimizuoto periodinio su pamaitinimu kultivavimo proceso metu, buvo gautas 25% biomasės kiekio padidėjimas, palyginus su tais rezultatai, kurie buvo naudojami matematinio modelio parametrų identifikavimui.

## 4. Dirbtiniai neuroniniai tinklai

Dirbtinis neuroninis tinklas yra matematinių modelių rinkinys, kuriuo bandoma imituoti biologinių sistemų savybes mokytis, prisitaikyti ir adaptuoti. Dirbtiniai neuroniniai tinklai sudaryti iš daugelio tarpusavyje sujungtų skaičiuojamųjų elementų. Šie elementai yra panašūs į biologinius neuronus ir jungiami vieni su kitais įvairaus stiprumo jungtimis, kurios yra analogiškos biologinių neuronų jungtims. Neuroniniai tinklai gali būti naudojami biotechnologiniams procesams modeliuoti, kurie yra sunkiai suprantami ir, kurių nepavyksta sumodeliuoti mechanistiniu būdu. Kaip pavyzdys tai gali būti sudėtingų kinetinių priklausomybių modeliavimas.

Dirbtiniai neuroniniai tinklai turi kelis privalumus, kuomet yra taikomi modeliuoti ir valdyti tikrus biotechnologinius procesus. Neuroniniai tinklai gali būti taikomi netiesinėms ir daug kintamųjų turinčioms sistemų priklausomybėms tirti. Neuroninio tinklo schema pateikta 4.1 pav.



4.1 pav. Dirbtinio neuroninio tinklo principinė schema [3]

Kiekvienas neuronas gali turėti keletą įėjimo signalų iš kitų neuronų ar iš įėjimo signalų. Kiekviena įėjimo jungtis turi savo perdavimo koeficientą (svorį). Kiekvienas neuronas gali turėti savo slenksčio reikšmę  $w_0$ . Neurono sužadavimo reikšmė formuojama skaičiuojant svorinę įėjimo signalų sumą ir atimant slenksčio reikšmę. Pagal sužadavimo signalo reikšmę, naudojant neurono perdavimo funkciją skaičiuojama neurono išėjimo reikšmė. Neurono išėjimo signalas apskaičiuojamas pagal formulę:

$$y = f\left(\sum_{i=1}^n w_i x_i + w_0\right) \quad (40)$$

Neuroninių tinklų galimybės priklauso nuo sluoksnių kiekio ir tuose sluoksniuose esančių neuronų skaičiaus, todėl yra naudojamos kelios neuroninių tinklų struktūros. Fermentacijos procesuose tiesioginio sklaidimo struktūros tinklai naudojami labai dažnai. Nors daugiasluoksniuose neuroniniuose tinkluose labiausiai paplitusi sigmoidinė aktyvavimo funkcija, kitos struktūros tinkluose naudojamos

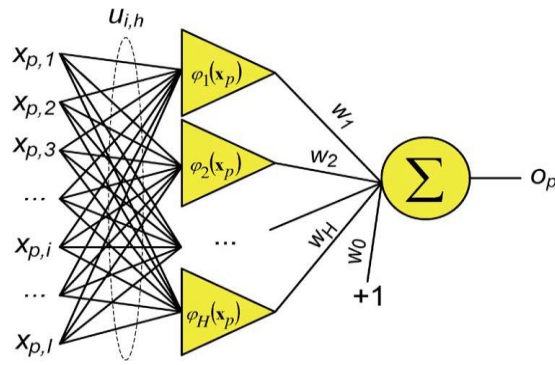
kitos funkcijos kaip pvz. radialinės bazės funkcija paremtuose tinkluose naudojamos Gauso formos aktyvavimo funkcijos.

Pasirinkus struktūrą, tinklai yra apmokomi t.y. pagal tam tikrą algoritmą keičiami tinklo svoriai bei aktyvavimo slenksčiai, taip parenkant geriausiai norimą funkciją aproksimuojančias reikšmes. Dažniausiai naudojami du apmokymo tipai [4]: prižiūrimas ir neprižiūrimas apmokymas. Prižiūrimo apmokymo algoritmai naudojami kuomet turimi proceso įėjimų – išėjimų duomenys, pagal tai būna keičiami tinklo svoriai.

Boškovic ir Narendra [5] atliko neuroniniais tinklais paremtos fermentacijos sistemos valdymo tyrimą, kurio metu buvo tiriami tiesinio, netiesinio ir dirbtinių neuroninių tinklų reguliatoriai periodinio veikimo su pamaitinimu sistemose. Tyrime buvo naudojama fermentacijos proceso kompiuterinė simuliacija, kiekvieno tyrimo metu naudojant identišką sistemos būseną. Gavus rezultatus, buvo nustatyta, kad neuroniniais tinklais paremta sistema parodė geriausius rezultatus, kuomet proceso dinamiką apibūdinančių duomenų yra nedaug. Kuomet buvo naudojami netiesiniai arba adaptyvūs netiesiniai reguliatoriai buvo gauti geriausi įmanomi rezultatai, tačiau tokios valdymo sistemos buvo pripažintos kaip nerealistiškos, dėl reikiamo didelio kiekio išankstinės informacijos apie procesą. Žymiai mažiau išankstinės informacijos reikalavo tiesinio adaptyvaus reguliatoriaus modeliavimas, tačiau jis tyrimo metu pasirodė prasčiausiai. Papildomai buvo tiriamos dvi skirtingos neuroninio tinklo bazinės funkcijos. Abiems tinklų mokymams buvo naudojami duomenų rinkiniai, kurie buvo išmatuoti prie pastovios temperatūros, vėliau jie buvo išbandomi prie kintančių temperatūros verčių. Grįžtamojo ryšio neuroninis tinklas pasirodė geriau nei tiesioginio sklidimo tinklas, kurių suminė kvadratinė paklaida gauta atitinkamai 0,06 ir 0,14.

#### **4.1 Radialinės bazės funkcijos (RBF) neuroninis tinklas**

Radialinės bazės funkcijų tinklai, kaip ir daugiasluoksniai tinklai, taip pat buvo pripažinti universaliais aproksimatoriais. Dėl šių tinklų paprastos ir fiksuotos trijų sluoksnių architektūros (5.2 pav.) RBF tinklai yra paprasčiau sudaromi ir apmokomi, nei daugiasluoksniai tinklai. RBF tinklai taip pat atsparūs įėjimų triukšmams, o tai pagerina sistemų stabilumą, todėl tokios struktūros tinklai yra geras pasirinkimas kaip alternatyvus metodas netiesinių sistemų modeliavimui ir valdymui.



5.2 pav. Dirbtinio neuroninio tinklo principinė schema [2]

Valdymo sistemos tampa sudėtingesnės jei sistemos netiesiškumai keičiasi laike, taip padarant procesą nenuspėjamą ir sunkiai valdomą *offline* režimu. Siekiant to išvengti, buvo pradėti naudoti aproksimatoriai su *online* mokymosi galimybėmis, kurie gali dinamiškai sureaguoti į įeinančių duomenų pokytį [2].

4.2 pav. pavaizduotas trijų sluoksnių architektūros RBF tinklas susidedantis iš  $I$  įėjimų,  $H$  RBF vienetų ir vieno išėjimo. Į tinklą padavus duomenų rinkinį  $x_p = (x_{p1}, x_{p2}, x_{p3}, \dots, x_{pi})$ , tinklo skaičiavimo etapai yra tokie [2]:

1) Įėjimo sluoksnio skaičiavimai: įėjimo sluoksnyje kiekvienas įėjimas  $x_{pi}$  yra padauginamas iš įėjimo svorio  $u_{ih}$ . Tai apibūdina svorinę jungtį tarp  $i$ -tojo įėjimo ir RBF vieneto  $h$

$$y_{p,h,i} = x_{p,i}u_{i,h} \quad (41)$$

Paprasčiausiu atveju visi įėjimo svoriai  $u$  yra nustatomi 1.

2) Paslėpto sluoksnio skaičiavimai: RBF išėjimas yra skaičiuojamas taip:

$$\varphi_h(x_p) = \exp\left(-\frac{\|y_{p,h} - c_h\|^2}{\sigma_h}\right) \quad (42)$$

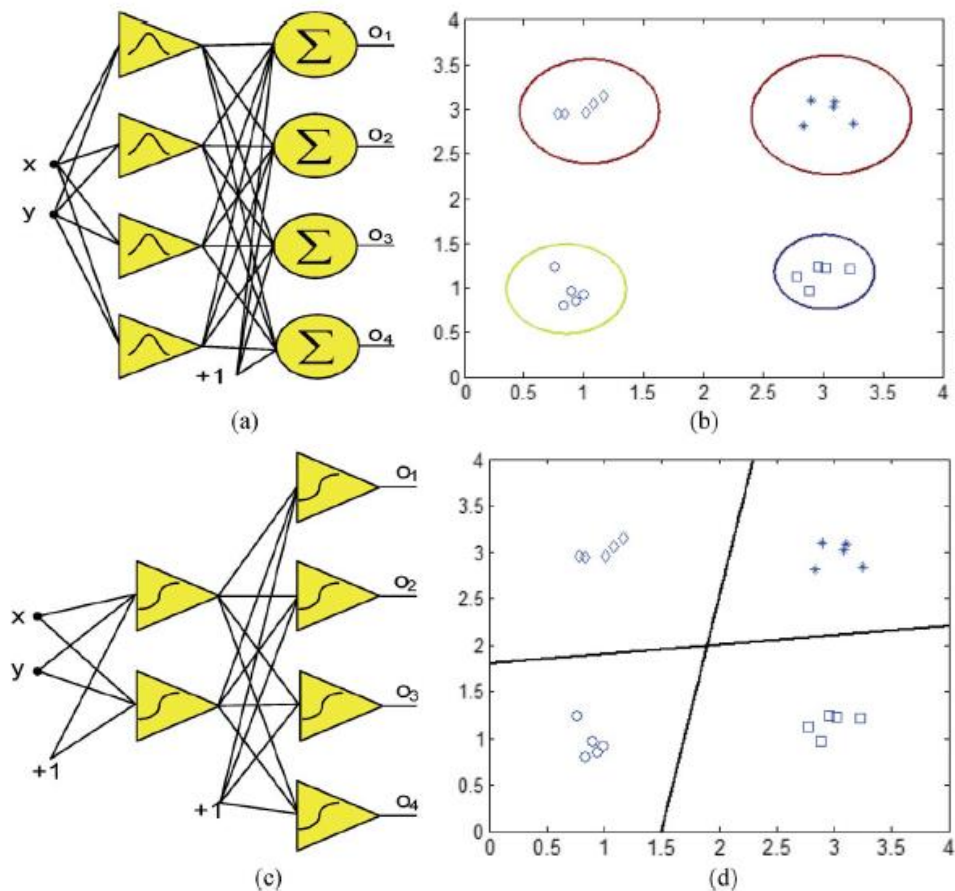
$\varphi_h$  - aktyvavimo funkcija.  $c_h$  ir  $\sigma_h$  yra funkcijos centras ir plotis atitinkamai, kurie yra pagrindiniai parametrai apibūdinantys radialinės bazės funkciją.

3) Išėjimo sluoksnio skaičiavimai: tinklo išėjimas yra skaičiuojamas kaip suma visų pasvertų išėjimų iš paslėpto sluoksnio.

$$o_p = \sum_{h=1}^H \varphi_h(x_p)w_h + w_0 \quad (43)$$

$w_h$  - svorinė reikšmė tarp RBF  $h$  vieneto ir tinklo įėjimo.  $w_0$ - slenksčio reikšmė.

Nepaisant panašumų tarp tinklų topologijos, RBF tinklai ir daugiasluoksniai tinklai turi skirtingas savybes. Pirma, RBF tinklai yra paprastesni nei daugiasluoksniai, kurie dažnai turi sudėtingesnę struktūrą. Dažniausiai RBF tinklai yra paprasčiau treniruojami dėka fiksuotos trijų sluoksnių architektūros. Jie taip pat veikia kaip lokaliniai aproksimuojantys tinklai ir tinklo išėjimas yra nustatomas pagal tam tikrą neuroną paslėptame sluoksnyje, o daugiasluoksniai tinklai veikia globaliai ir jo išėjimai yra apsprendžiami visų neuronų. Taip pat yra būtina nustatyti teisingas pradines sąlygas RBF tinklui, kuomet duagiasluoksniam tinklui tinka atsitiktinai sugeneruotos reikšmės. Skiriasi ir šių tinklų klasifikavimo mechanizmas: RBF klasteriai yra atskiriami sferomis, kuomet kiti tinklai klasterius atskiria griežtomis linijomis. Klasterizavimo skirtumas pateiktas 4.3 pav.



4.3 pav. Skirtingi klasterizavimo metodai [2]

## 5. Biotechnologinio proceso modeliavimas Matlab aplinkoje.

Darbo metu buvo modeliuojamas ir evoliucinio programavimo metodu optimizuojamas periodinis su pamaitinimu biotechnologinis procesas, kurio valdantysis poveikis yra pakeičiamas radialinės bazės funkcijos dirbtinio neuroninio tinklo aproksimacija. Darbas yra paremtas „*Model-based optimization*“

of viral capsid protein production in fed-batch culture of recombinant *Escherichia coli*“[6] straipsniu, kuriame aprašomas Pontriagino maksimumo principu optimizuojamas *E. coli* bakterijų kultivavimo procesas. Optimizavimo tikslas – maksimizuoti bendrą baltymų kiekį gale kultivavimo proceso. Tikslas funkcija aprašoma tokia lygtimi:

$$J = p_x(t_f)x(t_f)w(t_f) \rightarrow \max \quad (44)$$

$t_f$  – laikas proceso pabaigoje

## 5.1 Modelio aprašymas

Biotechnologinio proceso modeliavimas paremtas medžiagų masių balanso lygtimis biomasei  $x$ , pamaitinimo substratui  $s$ , baltymų aktyvumui  $px$  ir kultūros masei  $w$ . Šie išvardinti kintamieji apibūdina būsenos vektorių  $c$ .

Matematinis modelis sudarytas pagal šias diferencialines lygtis:

$$\frac{dx}{dt} = \mu(s)x - (u + F_1) \frac{s}{w} \quad (45)$$

$$\frac{ds}{dt} = -q_s(s)x + u \frac{s_f}{w} - (u + F_1) \frac{s}{w} \quad (46)$$

$$\frac{dp_x}{dt} = q_{px}(\mu, px) \quad (47)$$

$$\frac{dw}{dt} = u + F_1 + F_2 \quad (48)$$

Kultūros masės  $w$  kitimo greitis yra priklausomas nuo trijų masių srautų:  $u$ , substrato  $s$  padavimo į reaktorių greitis, yra pagrindinis valdantysis kintamasis.  $F1$  apibūdina  $CO_2$  ištekėjimą pro ventiliavimo liniją ir išgaravimą.  $F2$  – apibūdina svorio pasikeitimus, kurie atsiranda imant mėginius analizei.

Šie bei kiti kintamieji yra apskaičiuojami atskiromis algebrinėmis lygtimis.

Santykinis biomasės augimo greitis  $\mu$ , priklausantis nuo substrato koncentracijos  $s$  ir temperatūros  $T$  apskaičiuojamas pagal šią formulę:

$$\mu(s, T) = \mu_{max} \frac{s}{k_s + s} \cdot \frac{k_i}{k_i + s} \exp(\alpha(T - T_{ref})) \quad (49)$$

Biomasės augimo metu, iki indukcijos temperatūra bioreaktoriuje buvo palaikoma  $T=T_{ref}=37^\circ C$ . Kadangi ši temperatūra yra per aukšta baltymų gamybos fazei ir sumažina maksimalų įmanomą biomasės augimo greitį, todėl temperatūra to po indukcijos buvo palaikoma  $T=32^\circ C$ .

Priimama, kad santykinis substrato sunaudojimo greitis  $q_s$  daugiausiai priklauso nuo santykinio biomasės augimo greičio  $\mu$ :

$$q_s(s) = \frac{1}{Y_{xs}} \mu(s) + m \quad (50)$$

Santykinis baltymų koncentracijos kitimo greitis  $q_{px}$  aprašomas sekancia išraiška:

$$q_{px}(\mu, p_x) = \frac{1}{T_{px}} (p_{\max}(\mu) - p_x) \quad (51)$$

Reikia paminėti, kad baltymai yra negaminami esant biomasės augimo fazei, todėl augimo fazės metu ši išraiška yra pakeičiama į  $q_{px}=0$ .

Maksimali santykinės produkto koncentracijos vertė  $p_{\max}$  apskaičiuojama taip:

$$p_{\max}(\mu) = \frac{\mu k_m}{k_\mu + \mu + \mu^2/k_{i\mu}} \quad (52)$$

Taip pat apskaičiuojamas deguonies suvartojimo greitis  $OUR$ :

$$OUR = Y_{ox} \mu x + m_{ox} x \quad (53)$$

Visų pirma Matlab aplinkoje, naudojant aukščiau aprašytas medžiagų balanso ir pagalbinės lygtis, buvo bandoma kuo tiksliau atkartoti straipsnyje aprašytus 4 eksperimento rezultatus. Modeliavimui naudojamos parametrų vertės pateiktos 5.1 lentelėje.

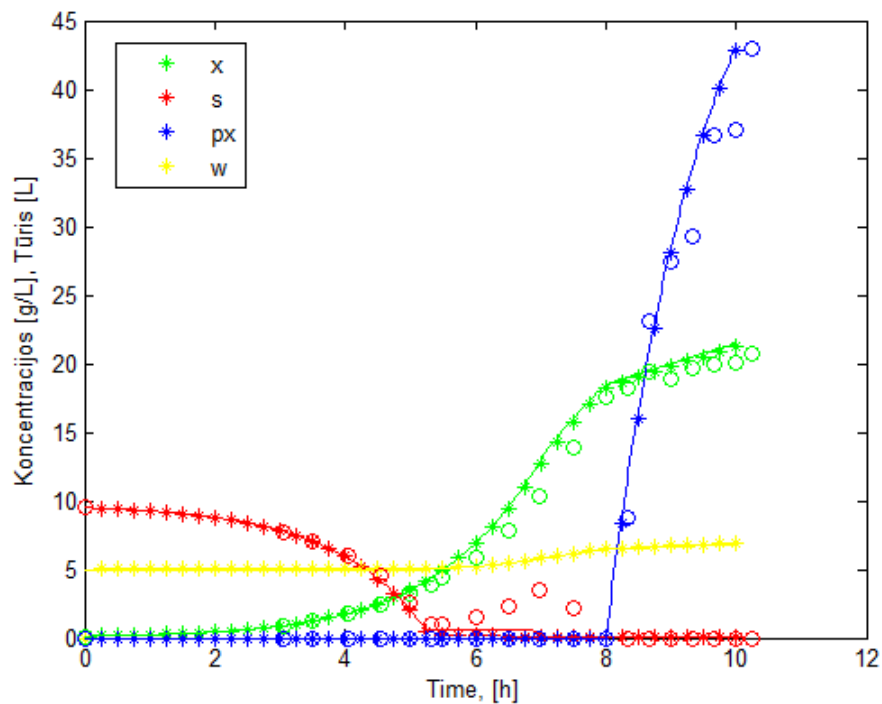
5.1 lentelė

Parametras	Vertė	Matavimo vienetai
$x(0)$	0,116	g/kg
$s(0)$	9,5	g/kg
$w(0)$	5,05	g/kg
$t_{ind}$	8,0	h
$s_f$	151	g/kg
$k_i$	93,8	g/kg
$k_{i\mu}$	0,0174	l/h
$k_m$	751	U/g
$k_\mu$	0,61	l/h
$k_s$	0,00333	g/kg
$m$	0,0242	g/g/h
$T_{px}$	1,495	h
$Y_{xs}$	0,46	g/g



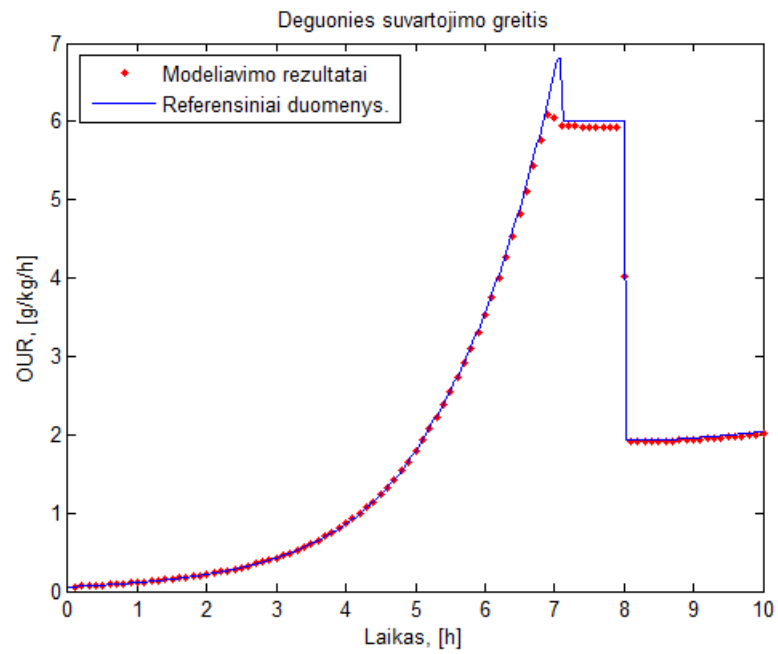
$\alpha$	0,0495	1/°C
$\mu_{\max}$	0,737	1/h

Atlikus imitacinį modeliavimą buvo gauti biomasės koncentracijos  $x$ , substrato koncentracijos  $s$ , santykinės baltymų koncentracijos  $p_x$  ir kultūros masės  $w$  kitimo grafikai. Šie rezultatai yra palyginami su referensiniais duomenimis, kurie buvo gauti iš straipsnio 5.1 pav.

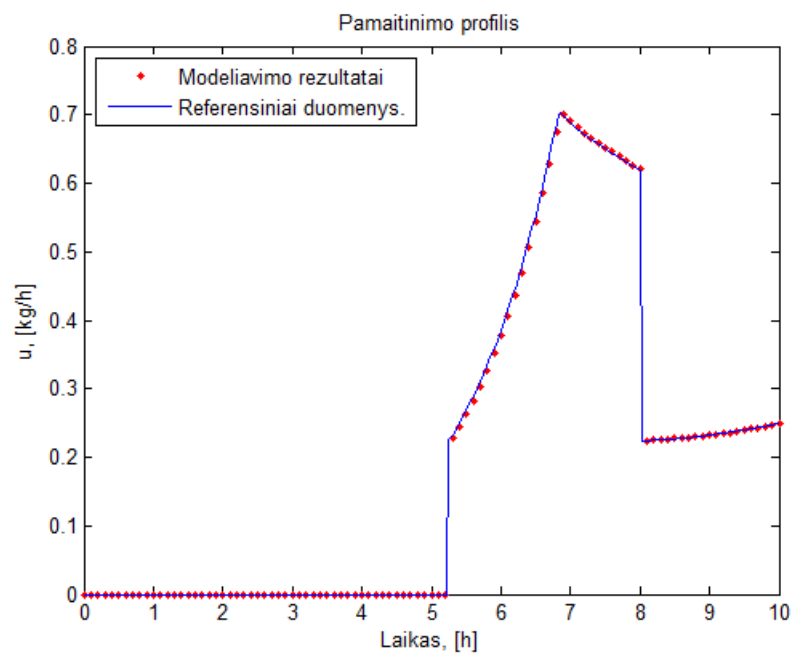


5.1 pav. Imitacinio modeliavimo ir referensinių taškų palyginimo rezultatai. Žvaigždėmis pavaizduotos modeliavimo vertės, linijomis - referensiniai duomenys, apskritimais - straipsnyje gauti eksperimentiniai rezultatai

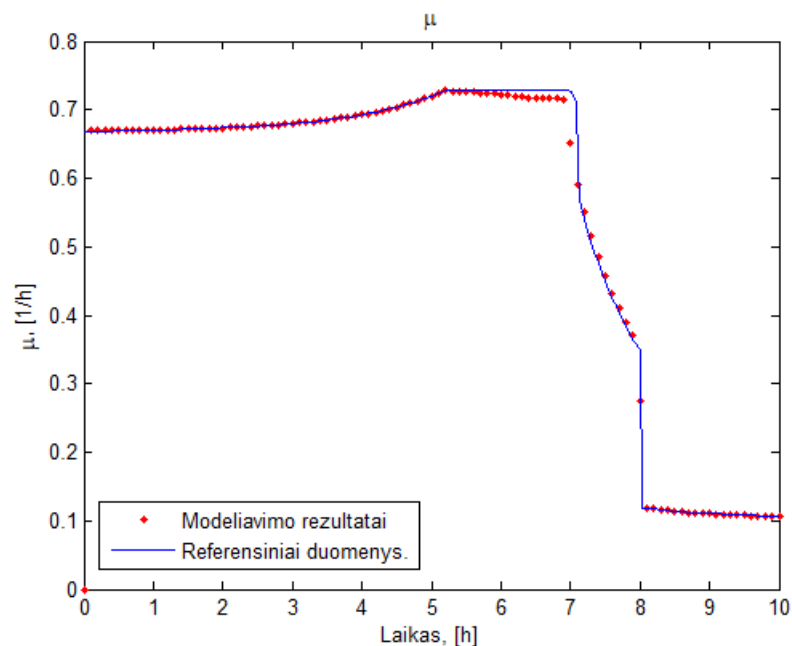
Taip pat atliktas ir pamaitinimo profilio  $u$ , santykinio biomasės augimo greičio  $\mu$ , ir deguonies suvartojimo greičio  $OUR$  rezultatų palyginimas. Rezultatai pavaizduoti 5.2, 5.3 ir 5.4 pav.



5.2 pav. Deguonies suvartojimo greičio imitacinio modeliavimo ir referensinių taškų palyginimo rezultatai. Taškais pavaizduotos modeliavimo vertės, linijomis - referensiniai duomenys



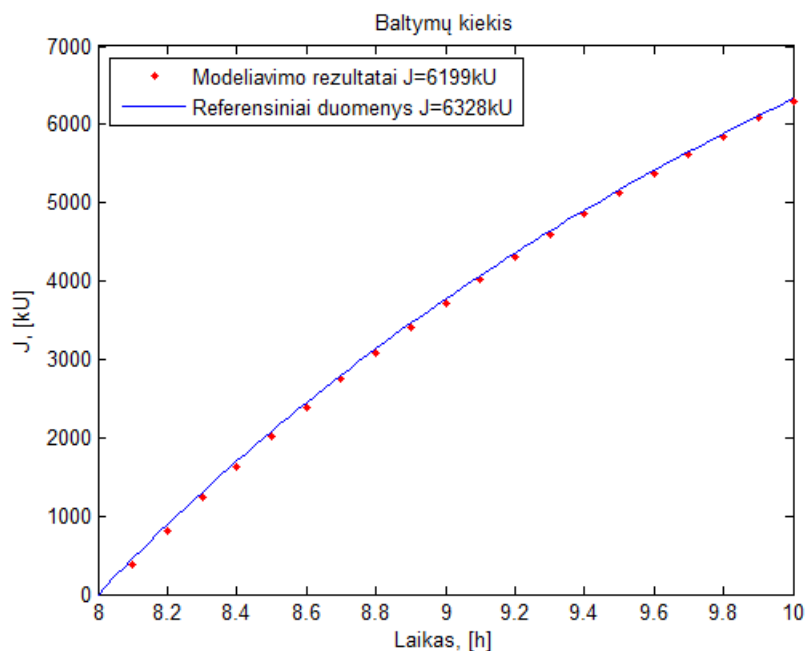
5.3 pav. pamaitinimo profilio  $u$  modeliavimo ir referensinių taškų palyginimo rezultatai. Taškais pavaizduotos modeliavimo vertės, linijomis referensiniai duomenys



**5.4 pav. santykinio biomasės augimo greičio modeliavimo ir referensinių taškų palyginimo rezultatai. Taškais pavaizduotos modeliavimo vertės, linijomis referensiniai duomenys**

Būsenos kintamųjų  $x$ ,  $px$  ir  $w$  modeliavimo rezultatai labai gerai atitinka eksperimentinius taškus, tačiau šiek tiek skiriasi substrato sunaudojimo greičio  $s$  kreivės nuo 5 iki 7 valandos. Modeliavimo metu substrato nuo 5 valandos jau nebėra, tai pasireiškia tam tikrais netikslumais santykinio masės augimo greičio grafike, kuriame matomas verčių mažėjimas nuo 5 iki 7 valandos, palyginus su eksperimentiniais taškais. Kadangi deguonies suvartojimo greitis OUR tiesiogiai priklauso nuo santykinio biomasės augimo greičio, pastarajam kintamajam nepasiekus savo maksimalių verčių taip pat nepasiekiamos ir OUR maksimalios vertės. Tai matoma 5.2 pav. apie 7 val.

Tačiau svarbiausias modeliuojamas parametras – galutinis produkto (baltymų) kiekis biotechnologinio proceso pabaigoje  $t_f = 10$  val. Dėl anksčiau aprašytų modeliavimo netikslumų, esančių nuo 5 iki 7 val., galutinis gauto produkto kiekis yra 6299 kU, tuo tarpu straipsnyje pateikto eksperimento metu buvo gauta 6328 kU, tai sudaro 0,45% skirtumo, kurį galima laikyti nežymiu ir atsiradusį dėl skaičiavimo paklaidų. Gauti galutinio produkto modeliavimo rezultatai pavaizduoti 5.5 pav.



5.5 pav. Galutinio produkto kiekio modeliavimo ir referensinių taškų palyginimo rezultatai. Taškais pavaizduotos modeliavimo vertės, linijomis referensiniai duomenys

### 5.3 Galutinio produkto kiekio optimizavimas

Šio darbo tikslas – naudojant radialinės bazės funkcijų (RBF) neuroninį tinklą, kurio veikimas išsamiau aprašytas 4.1 skyriuje, ir naudojant evoliucinio optimizavimo procedūrą, kuri apskaičiuotų tokias dirbtinio neuroninio tinklo svorinių koeficientų vertes  $w_i$ , gauti maksimalų galutinio produkto biotechnologiniame procese kiekį. Taigi šiuo atveju optimizavimo procedūra optimizuoja ne modelio parametrų reikšmes, o valdymo profilio  $u$  aproksimavimo funkcijos parametrų reikšmes.

Visų pirma, substrato pamaitinimo profilis yra pakeičiamas RBF tinklo funkcija:

$$y = \sum_{i=1}^n w_i \Phi_i \quad (54)$$

čia  $\Phi_i$  yra Gauso funkcijos, aprašančios varpo formos priklausomybes

$$\Phi_i = \exp(-r_i^2 / (\rho_i^2 + \xi)) \quad (55)$$

$r_i = t - c_i$ ,  $\xi$  yra maža konstanta, šiuo atveju  $\xi = 0,001$ , naudojama skaitmeniniam stabilumui užtikrinti,  $c_i$ ,  $\rho_i$  ir  $w_i$  yra optimizuojami parametrai:  $c_i$  nustato varpo centro padėtį,  $\rho_i$  nustato varpo plotį,  $w_i$  nustato  $i$ -tosios Gauso funkcijos svorį funkcijoje.

Pradžioje darbo yra sudaroma 3 skirtingi RBF tinklai, kurie turi skirtingus savus parametrus  $w_i$  ir  $c_i$ , o varpo plotis  $p_i$  visiems nustatomas vienodas  $p_i=0,5$ . Pradiniai neuroninių tinklų parametrai yra parenkami rankiniu būdu, stengiantis kuo geriau atkartoti optimalaus pamaitinimo profilio kreivę. Pirmąjį neuroninį tinklą sudaro po 12 centro padėčių  $c_i$  ir svorinių koeficientų  $w_i$  turintis tinklas.

Antrasis sudarytas iš 8 centro padėties ir svorinių koeficientų parametru, trečiasis turi 6 parametrus. Skirtingas dirbtinių neuroninių tinklų parametru skaičius parenkamas norint atrasti ne tik optimalius optimizuojamus parametrus, tačiau ir optimalų kiekį šių parametru.

Kiekviename realiame procese esantys technologiniai ar fizikiniai apribojimai yra svarbi problema. Tik tiksliai įvertinus procese esančius apribojimus, galima apskaičiuoti realistiškus ir įgyvendinamus valdymo profilius ir būsenos kintamųjų sąlygas. Kitu atveju, vienu ar kitu būdu atlikus optimizavimo procedūrą, iš pažiūros galima gauti labai geras optimizuojamos tikslo funkcijos vertes, kurių realiuose eksperimentuose atkartoti nepavyks. Egzistuoja keli būdai, kaip šiuos apribojimus galima įvertinti [7].

Paprasčiausias būdas yra apskaičiuvus valdymo profilius, patikrinti, ar jie nerviršija nustatytų technologinių apribojimų. Patikrinus, galima tiesiog apriboti už nustatyto diapazono išeinančias valdymo kintamųjų reikšmes. Sudėtingiau yra tuomet, kai apribojimai taikomi ne valdymo profiliams, o būsenos kintamiesiems. Jei apskaičiuvus valdymo profilį pasirodo, kad vienas ar keli būsenos kintamieji išeina už leistinų ribų, negalima tiesiog koreguoti jų laiko trajektorijų – tam būtina sumažinti arba padidinti valdymo kintamųjų reikšmes. Šiuo atveju yra naudojamas taip vadinamas baudos funkcijų metodas.

Kuomet taikomas šis metodas ir būsenos kintamieji  $X$  viršija apribojimus, optimizuojamas proceso kokybės kriterijus  $J$  yra pabloginamas tam tikra apribojimų pažeidimo dydžiui proporcinga verte. Svorio koeficientas  $\alpha$  parenkamas taip, kad baudos funkcijos  $F(X, p)$  daromas poveikis būtų pakankamai svarus, tačiau pernelyg neiškreiptų galutinės tikslo funkcijos priklausomybės nuo optimizuojamų valdymo parametru  $p$ :

$$J = J_1(X, p) + \alpha F(X, p) \quad (56)$$

Teisingai parinktus  $\alpha$  koeficientą ir atlikus optimizavimo procedūrą, antrasis sumos narys turėtų tapti labai mažas ir nedaryti įtakos optimizuoto valdymo profilio pritaikymui realiuose procesuose.

Taikant šią baudos funkcijos optimizavimo procedūrą, optimizuojamų parametru reikšmės gali būti parenkamos gana plačiu diapazonu, nes apie valdymo profilio optimalumą bus sprendžiama

analizuojant galutines tikslo funkcijos reikšmes, tokiu būdu nereikalingos optimizuojamų parametru vertės tiesiog bus atmetamos.

Optimizavimo procedūrai buvo įvestas švelnus deguonies suvartojimo greičio OUR apribojimas iki  $OUR_{max}=6$  g/kg/h. Procedūros metu, OUR vertei viršijus nustatytą ribą, apskaičiuotai tikslo funkcijos vertei  $J$  buvo priskaičiuojami baudos balai, tokiu būdu yra užtikrinama, jog optimizuojami kintamieji, verčiantys viršyti apribojimus nebūtų priimami kaip optimalūs.

## 5.4 Evoliucis optimizavimo metodas

Evoliucinis programavimas yra vienas iš plačiausiai paplitusių atsitiktinės paieškos optimizavimo metodų. Pagrindiniai šio metodo privalumai yra universalumas, kadangi tinka įvairiam parametru skaičiui optimizuoti, nesudėtingas realizavimas ir galimybė rasti globalinį ekstremumo tašką. Pagrindinis metodo trūkumas yra tas, kad norint nustatyti ekstremumo tašką, reikalingas didelis iteracijų skaičius, dėl ko optimizavimo procedūra užtrunka ilgiau.

Evoliucinio programavimo algoritmo realizacija pradedama nuo kintamųjų vektorių pradinės sekos parinkimo:  $x_1, x_2, \dots, x_p$  ši seka dar gali būti vadinama *tėvais*.

Sekantis žingsnis yra mutacija, po kurio gaunamos naujos kintamųjų kombinacijos – vadinamos *palikuonimis*. Vektoriaus  $x_i$  mutacija atliekama taip:

$$x_i^* = x_i + \Delta x \cdot RAND \quad (57)$$

*RAND* yra atsitiktinai sugeneruotas skaičius.  $\Delta x$  - parametrai nustatantys mutacijų laipsnį. Šie kintamieji yra parametrai, kurie įtakoja algortimų konvergavimo greitį.

Nauja kintamųjų seka yra sudaroma iš ankstesnės sekos, atrenkant vektorius, kurie atitinka didžiausias tikslo funkcijos vertes. Algoritmas atlieka šią atranką, palygindamas kiekvieną kintamųjų vektorių porą, ir atrinkdamas tą vektorius, kuris duoda didesnę tikslo funkcijos vertę. Atrinktas vektorius tampa *tėvais* sekančių mutacijų generavimui.

Optimizavimo procedūra yra užbaigiama, kada pasiekiamas tikslo funkcijos ekstremumas ir nebegaunama tikslo funkcijos pagerėjimo arba pasiekiamas užduotas iteracijų skaičius.

Taigi evoliucinio programavimo algoritmas yra realizuojamas pagal šiuos žingsnius:

1. Parenkama pradinė vektorių-tėvų populiacija  $x_1, x_2, \dots, x_p$ , kur kiekvienas vektorius turi  $n$  verčių.
2. Apskaičiuojamos tikslo funkcijos vertės  $f(x_i), i = 1, \dots, p$ .

3. Iš kiekvieno *tėvų* vektoriaus sudaromas atitinkamas palikuonių vektorius, prie kiekvieno vektoriaus elementų pridedant atsitiktines Gauso kintamųjų reikšmes, padaugintas iš parametru  $\Delta x_i$ , kurios nustato mutacijų laipsnį.
4. Apskaičiuojamos vektorių - palikuonių tikslo funkcijos vertės.
5. Toliau vektoriai yra palyginami pagal tikslo funkcijos vertę. Laimi tas vektorius, kuris tikslo funkcijai suteikia didesnę vertę. Toliau vektoriai yra surikiuojami pagal jų laimėjimų skaičių.
6. Atrenkami vektoriai su didžiausiu laimėjimų skaičiumi, kurie toliau bus naudojami kaip vektoriai-*tėvai* naujų kintamųjų populiacijai sudaryti.
7. Algoritmas baigiamas tada, kai mutacijos nebepagerina tikslo funkcijos verčių, arba pasiekiamas maksimalus užduotas iteracijų skaičius. Kitu atveju skaičiavimas tęsiamas nuo trečio žingsnio.

Kadangi optimizavimo algoritmas yra stochastinis ir tikslo funkcijos maksimali vertė kiekvieno optimizavimo metu gaunama šiek tiek skirtinga, todėl kiekvienam neuroniniui tinklui procedūra buvo kartojama kelis kartus.

## 5.5 Optimizavimo rezultatai

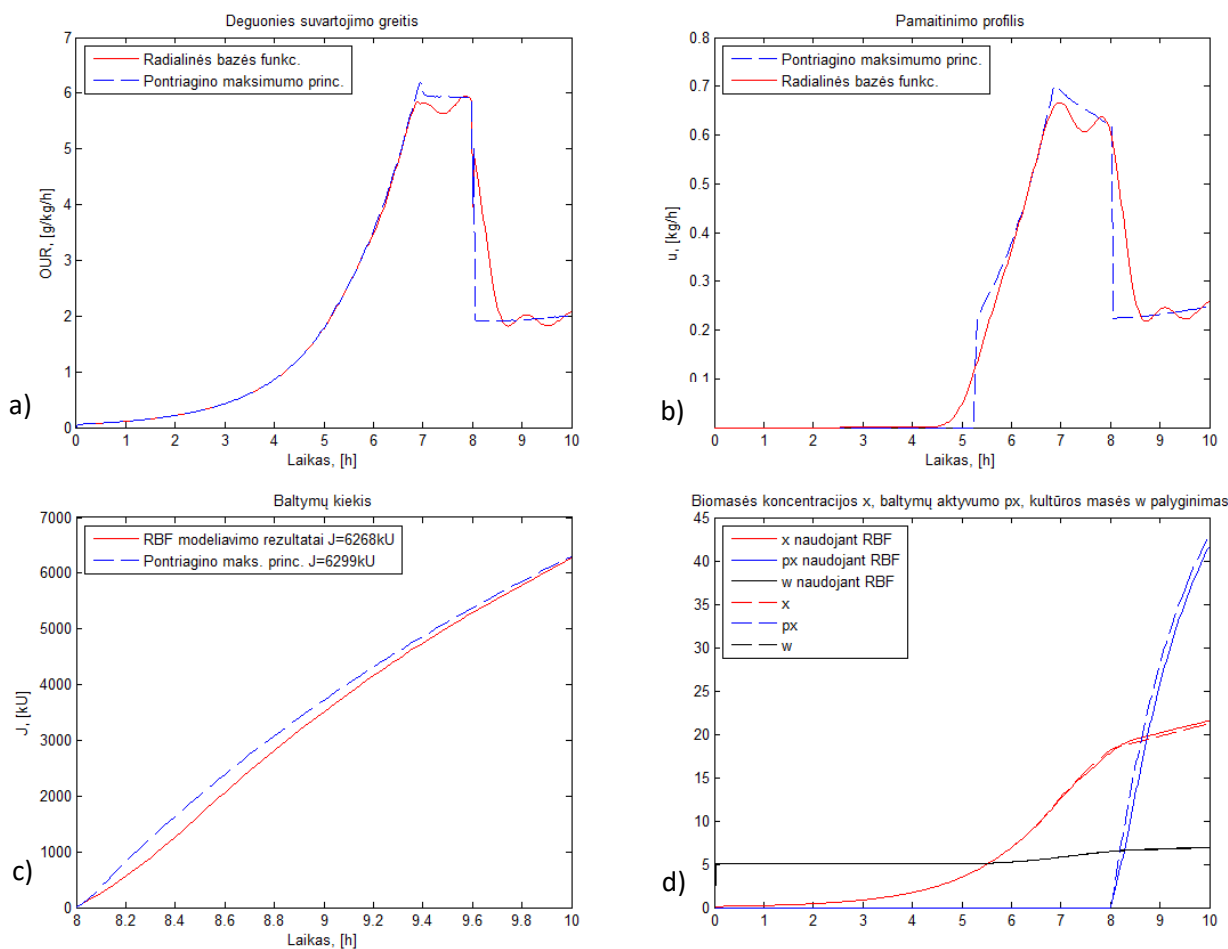
Pradžioje, norima išsiaiškinti kokius geriausius rezultatus galima gauti optimizuojant tik vieną iš trijų neuroninio tinklo kintamųjų – svorinį koeficientą. Tam buvo pasirinkti 12, 8 ir 6 radialinių funkcijų neuroniniai tinklai. Kiekvienam iš jų pradžioje buvo numatomos pradedamosios svorinių koeficientų vertės, kurios buvo parenkamos rankiniu būdu, modeliuojant procesą ir stebint pamaitinimo profilio kreivės panašumą į optimalų, gautą taikant Pontriagino maksimumo principą. Kuomet profiliai tapdavo panašūs savo forma, svorinių koeficientų vertės būdavo fiksuojamos ir naudojamos kaip pradinės vertės optimizavimo procedūrai. Tai buvo atliekama siekiant sumažinti optimizavimo procedūros populiacijų ir iteracijų skaičių, taip sumažinant ir laiką, reikalingą rasti optimaliam sprendiniui. Taip pat svarbūs optimizavimo procedūrai kintamieji – tai žemutinės ir viršutinės optimizuojamų parametru ribos. Šios ribos, taip pat kaip ir pradinės parametru vertės, buvo parenkamos logiškai, atsižvelgiant į realiai galimas įgyti parametru vertes.

Atlikus optimizavimo procedūra pirmajam dirbtiniui neuroniniui tinklui gauti rezultatai, kurie pavaizduoti 5.6 pav. Optimizacijos metu buvo keičiamia dvylika radialinės bazės funkcijos svorinių koeficientų. Produkto kiekis yra lyginamas su modeliavimo rezultatais, kurių metu buvo naudojamas optimalus pamaitinimo profilis  $u$ , gautas naudojant Pontriagino maksimumo principą, su kuriuo buvo pasiektas maksimalus baltymų kiekis 6299 kU.

Prie gautų rezultatų pateikiamos lentelės, kuriose nurodytos optimizavimo procedūrai pateiktos pradinės parametrų vertės  $par_0$ , apatiniai  $par_L$  bei viršutiniai  $par_H$  parametrų vektoriai, kurių ribose bus ieškoma optimalių verčių. Taip pat nurodytos centrinės Gauso funkcijų padėtys  $c_i$ .  $Par_{opt}$  – tai optimizacijos metu gauti parametrai, kurie naudojami RBF neuroninio tinklo išėjimui skaičiuoti

### 6.5.1 Pirmasis atvejis

Imitacinio modeliavimo rezultatai, optimizuojant dvylikos radialinės bazės funkcijų neuroninio tinklo svorinius koeficientus:



5.6 pav. a) deguonies suvartojimo, b) pamaitinimo profilių, c) baltymų kiekio, d) x, px, w palyginimamieji rezultatai.



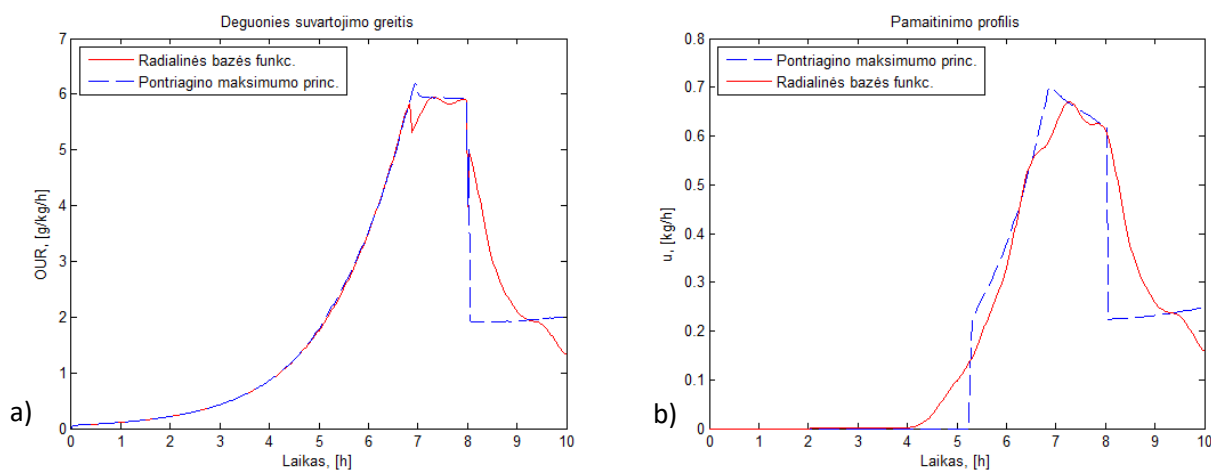
5.2 lentelė

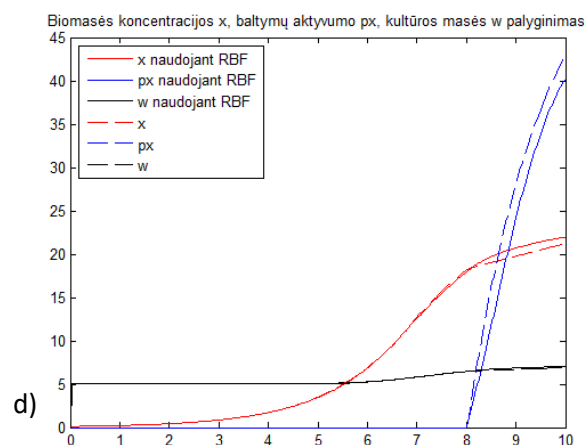
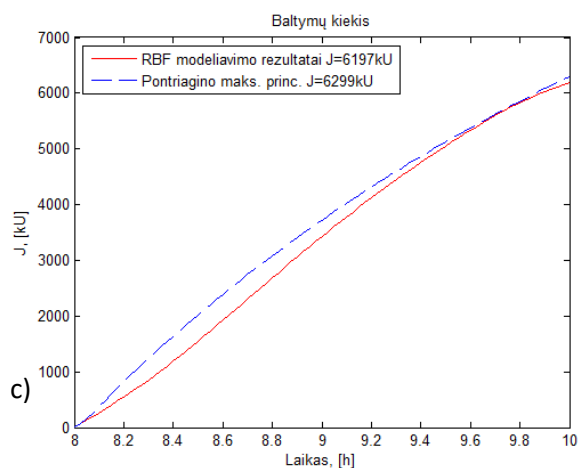
	$w_i$												$J_{\text{Pontr,}}$ kU	$J_{\text{opt,}}$ kU
parL	0,0	0,0	0,0	0,1	0,3	0,1	0,3	0,0	0,0	0,0	0,0	0,0	6299	6268
par0	0,0	0,12	0,2	0,3	0,45	0,25	0,5	0,0	0,2	0,08	0,18	0,12	Skirtumas, %	
parH	0,3	0,23	0,4	0,4	0,55	0,45	0,6	0,2	0,4	0,2	0,3	0,2	- 0,49	
ci	5,0	5,5	6,0	6,5	7,0	7,5	8,0	8,5	9,0	9,5	10,0	10,5		
par <sub>opt</sub>	0,0	0,12	0,2	0,3	0,45	0,25	0,5	0,0	0,2	0,08	0,18	0,12		

Naudojant 12 radialinės bazės funkcijų, optimizuojant tik funkcijų svorinį koeficientą, buvo gauti labai artimi rezultatai tiems, kurių metu buvo naudojamas Pontriagino maksimumo principu optimizuotas pamaitinimo profilis. Deguonies suvartojimo greičio ir pamaitinimo profilio kreivės daugiausiai skiriasi apie 5 ir 8 valandomis, kuomet optimali vertė staigiai pasikeičia. Tai lemia RBF varpo forma ir sąlyginai mažas funkcijų kiekis, kuriam esant staigūs verčių pokyčiai yra sunkiai realizuojami. Modeliavimo pabaigoje, nuo 8 valandos, baltymų aktyvumas gaunamas mažesnis, todėl baltymų kiekio kreivės atitolsta, tačiau biomasės koncentracijai išaugus, skirtumas pradeda mažėti. Proceso pabaigoje gaunama 0,49% mažiau baltymų, nei naudojant optimalų pamaitinimo profilį.

### 5.5.2 Antrasis atvejis

Imitacinio modeliavimo rezultatai optimizuojant aštuonių radialinės bazės funkcijų neuroninio tinklo svorinius koeficientus:





5.7 pav. a) deguonies suvartojimo, b) pamaitinimo profilių, c) baltymų kiekio, d) x, px, w palyginimamieji rezultatai.

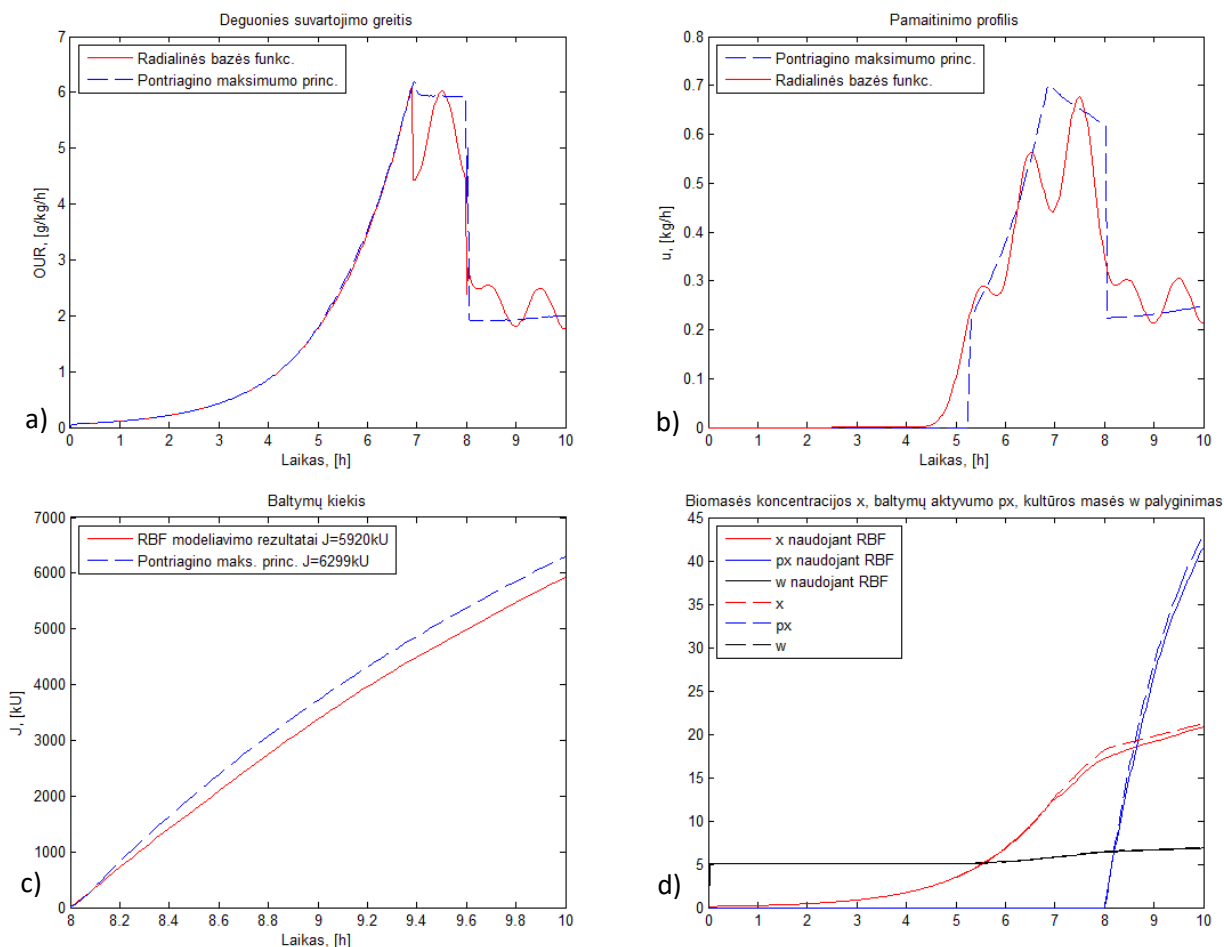
5.3 lentelė

	$w_i$								$J_{Pontr}, kU$	$J_{opt}, kU$
parL	0,0	0,0	0,4	0,45	0,45	0,05	0,1	0,1	6299	6197
par0	0,0	0,2	0,5	0,55	0,55	0,15	0,2	0,2	Skirtumas, %  - 1,65	
parH	0,1	0,25	0,55	0,6	0,6	0,25	0,25	0,25		
ci	5,0	5,75	6,5	7,25	8,0	8,75	9,5	10,25		
par <sub>opt</sub>	0,07	0,19	0,46	0,56	0,52	0,23	0,19	0,1		

Sekančios optimizacijos metu buvo sumažintas RBF kiekis nuo dvylikos iki astuonių, kas lėmė jų pasikeitusią centrinę padėtį laiko ašies atžvilgiu – 0,5 val. tarpas buvo padidintas iki 0,75 siekiant tolygiai išdėstyti funkcijas laiko ašyje. Sumažinus funkcijų kiekį, taip pat sumažėjo ir neuroninio tinklo aproksimavimo galimybės. Lyginant gautus rezultatus su praėjusiais, deguonies suvartojimo greičio ir pamaitinimo profilio kreivės dar labiau skyrėsi nuo optimalių esant staigiems verčių pasikeitimams. Nors gautas pamaitinimo profilis leido padidinti biomasės koncentraciją, tačiau baltymų aktyvumo sumažėjimas lėmė mažesnę galutinę produkto kiekį. Jo gauta 1,65% mažiau.

### 5.5.3 Trečiasis atvejis.

Imitacinio modeliavimo rezultatai, optimizuojant šešių radialinės bazės funkcijų neuroninio tinklo svorinius koeficientus:



5.8 pav. a) deguonies suvartojimo, b) pamaitinimo profilių, c) baltymų kiekio, d) x, px, w palyginimamieji rezultatai.

5.4 lentelė

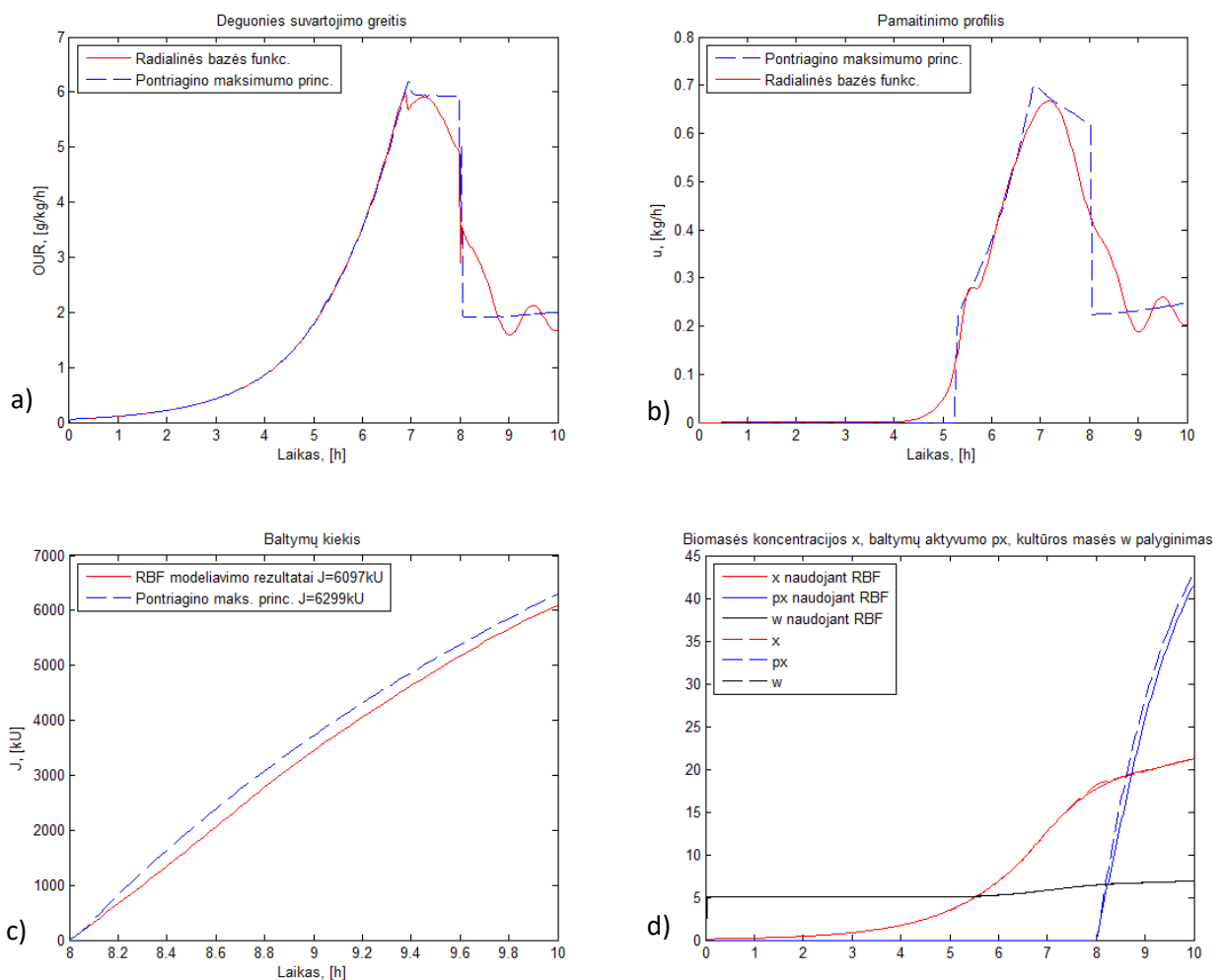
	$w_i$						$J_{\text{Pontr, kU}}$	$J_{\text{opt, kU}}$
parL	0,2	0,5	0,55	0,1	0,15	0,2	6299	5920
par0	0,3	0,67	0,67	0,22	0,25	0,3	Skirtumas, %  - 6,4	
parH	0,4	0,75	0,75	0,35	0,35	0,4		
ci	5,5	6,5	7,5	8,5	9,5	10,5		
par <sub>opt</sub>	0,27	0,54	0,66	0,28	0,29	0,28		

Toliau buvo optimizuojamas šešių RBF neuroninis tinklas, kuomet kiekvienos funkcijos centrinė padėtis laiko atžvilgiu skyrėsi 1 valanda. Iš optimizavimo rezultatu matyti, kad sumažinus funkcijų skaičių, optimizuotas profilis nebetiksliai aproksimuoja Pontriagino maksimumo principu gautą

pamaitinimo profilį. Modeliavimo metu biomasės koncentracija ir baltymų aktyvumas gaunamas mažesnis, todėl ir kultivavimo pabaigoje galutinio produkto kiekis gaunamas 6,4% mažesnis. Iš gautos pamaitinimo profilio kreivės galima matyti, kad mažinant funkcijų skaičių, bet optimizavimo metu nekeičiant jų kitų parametrų: varpo pločio arba centrinės padėties, RBF neuroninis tinklas nebegali sklandžiai aproksimuoti norimos kreivės, atsiranda bangavimų. Tolimesniame etape bus bandoma pagerinti radialinės bazės funkcijų aproksimavimo galimybes tuo pačiu metu optimizuojant svorinius koeficientus bei varpų pločius.

#### 5.5.4 Ketvirtasis atvejis.

Imitacinio modeliavimo rezultatai, optimizuojant šešių radialinės bazės funkcijų neuroninio tinklo svorinius koeficientus ir funkcijų plotį:



5.9pav. a) deguonies suvartojimo, b) pamaitinimo profilių, c) baltymų kiekio, d) x, px, w palyginimamieji rezultatai.

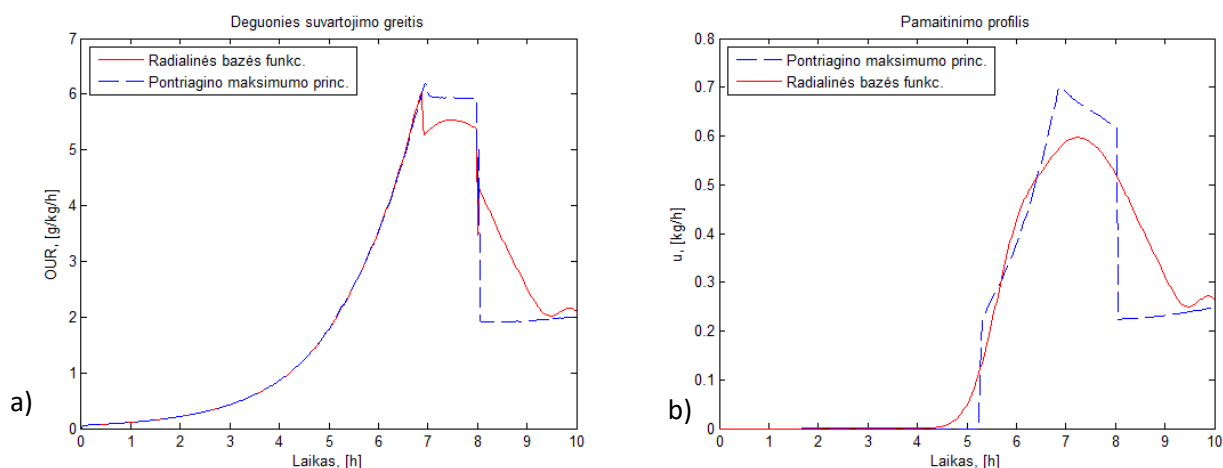
5.5 lentelė

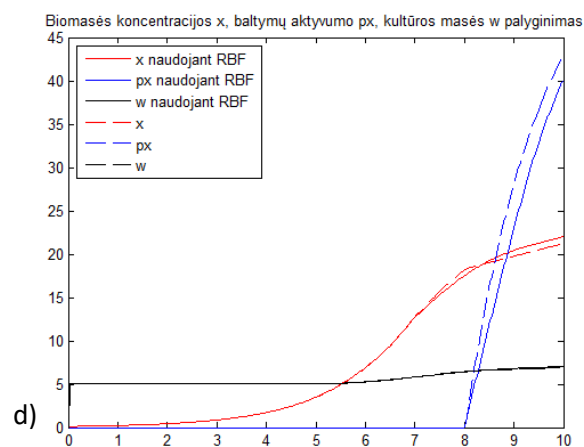
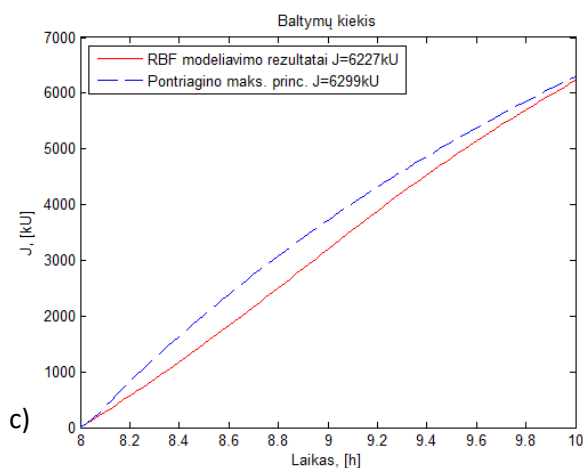
	$w_i$						$\rho_i$						$J_{\text{Pontr, kU}}$	$J_{\text{opt, kU}}$
parL	0,05	0,25	0,25	0,11	0,12	0,15	0,1	0,05	0,4	0,25	0,25	0,25	6299	6097
par0	0,1	0,45	0,45	0,22	0,25	0,3	0,2	0,1	0,8	0,5	0,5	0,5	Skirtumas, %	
parH	0,2	0,9	0,9	0,44	0,5	0,6	0,4	0,2	1,6	1	1	1	- 3,31	
ci	5,5	6,5	7,5	8,5	9,5	10,5								
par <sub>opt</sub>	0,1	0,45	0,45	0,22	0,25	0,3	0,2	0,1	0,8	0,5	0,5	0,5		

Atlikus šešių radialinės bazės funkcijų optimizaciją su dvylika optimizuojamų parametru: šešiais svoriniais koeficientais  $w_i$  ir tiek pat varpo pločio koeficientų  $\rho_i$  baltymų gauta mažiau, nei naudojant optimalų pamaitinimo profilį, tačiau lyginant su ankstesniu rezultatu, gautas žymus pagerėjimas. Biomės koncentracijos kreivės beveik identiškos, tačiau baltymų aktyvumas išlieka mažesnis viso kultivavimo metu, kas nulemia 3,3 % mažesnę produkto kiekį. Palyginus 5.8 ir 5.9 grafiko b) paveikslus, galima pastebėti, kad į optimizavimo procedūrą pridėjus varpo pločio parametrus, galima gauti glotnesę ir geriau optimalią pamaitinimo trajektoriją atitinkančią kreivę.

### 5.5.5 Penktasis atvejis.

Imitacinio modeliavimo rezultatai, optimizuojant penkių radialinės bazės funkcijų neuroninio tinklo svorinius koeficientus ir funkcijos plotį:





5.10 pav. a) deguonies suvartojimo, b) pamaitinimo profilių, c) baltymų kiekio, d) x, px, w palyginimamieji rezultatai.

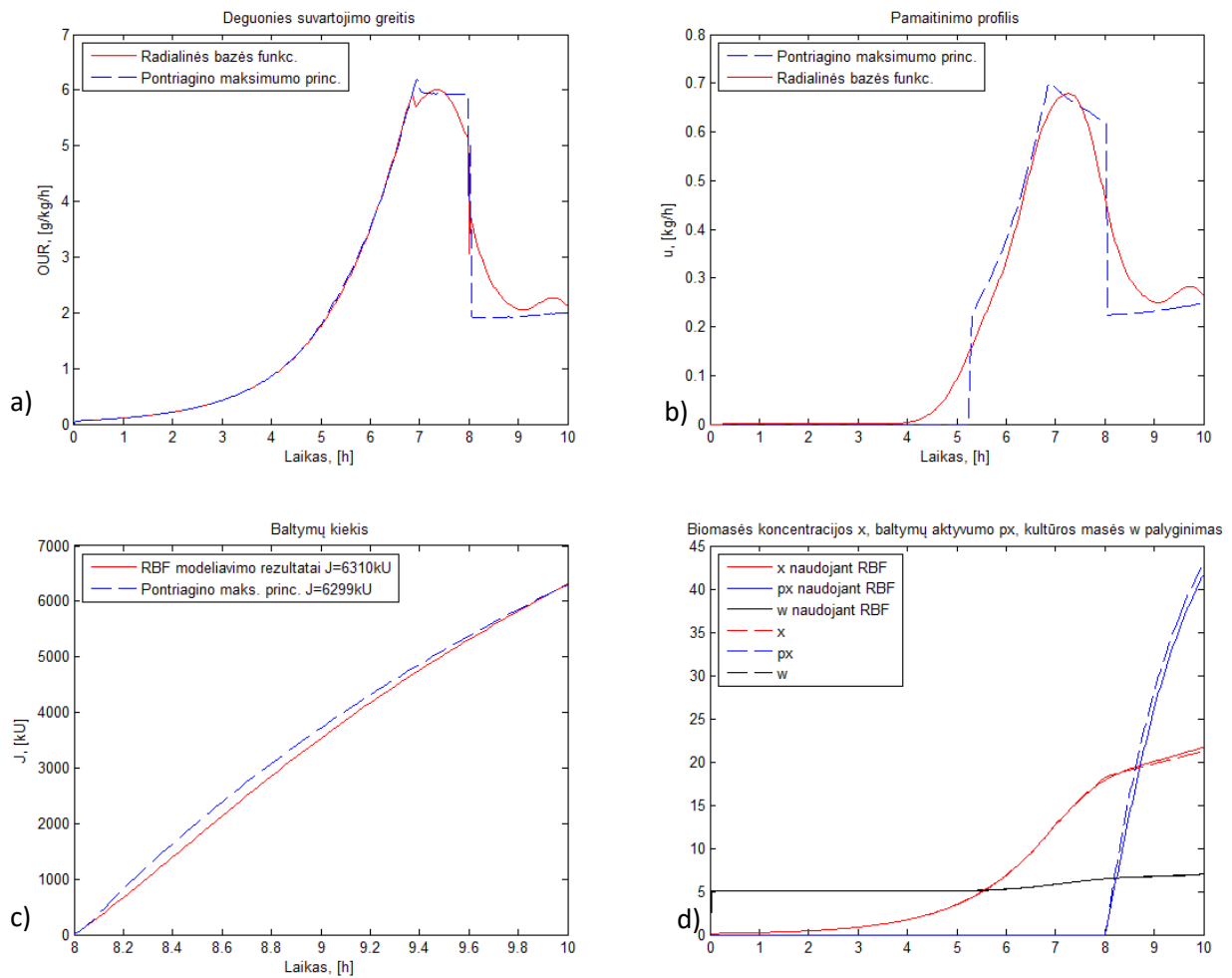
5.6 lentelė

	$w_i$					$\rho_i$					$J_{\text{Pontr}}$ , kU	$J_{\text{opt}}$ , kU
parL	0,0	0,2	0,2	0,22	0,25	0,0	0,0	0,0	0,0	0,0	6299	6227
par0	0,3	0,5	0,5	0,1	0,2	0,5	0,5	0,5	0,5	0,5	Skirtumas, %	
parH	0,36	0,55	0,55	0,2	0,2	1,0	1,0	1,0	1,0	1,0	- 1,15	
ci	6	7	8	9	10							
par <sub>opt</sub>	0,32	0,45	0,38	0,22	0,23	0,72	0,82	0,81	0,73	0,55		

Atlikus dešimties parametrų optimizaciją: penkių svorinių ir penkių varpo pločio koeficientų, gauti rezultatai pavaizduoti 5.10 pav. Iš gautų rezultatų galima matyti, kad optimizacija pavyko geriau, nei ketvirtuoju atveju, kuomet buvo optimizuojama daugiau parametrų. Taip galėjo nutikti dėl skirtingų radialinių funkcijų centrinių pozicijų laiko atžvilgiu, kurios šiuo atveju buvo palankesnės optimalaus rezultato paieškoje. Nors tarp 7 – 8 valandos pamaitinimo profilis nepasiekė optimalių verčių, tačiau didesnių verčių išlaikymas po 8 valandos lėmė biomasės koncentracijos didėjimą, kuris su palyginti mažu baltymų aktyvumu kultivavimo pabaigoje leido pasiekti gerus rezultatus. Baltymų buvo gauta 1,15 procentų mažiau, nei naudojant optimalų pamaitinimo profilį.

### 5.5.6 Šeštasis atvejis

Imitacinio modeliavimo rezultatai, optimizuojant šešių radialinės bazės funkcijų neuroninio tinklo svorinius koeficientus, funkcijos plotį ir centrinę varpo padėtį



5.11 pav. a) deguonies suvartojimo, b) pamaitinimo profilių, c) baltymų kiekio, d) x, px, w palyginimamieji rezultatai.

5.7 lentelė

	$W_i$						$\rho_i$						$C_i$							
parL	0,0	0,3	0,3	0,1	0,1	0,1	0,2	0,2	0,2	0,2	0,2	0,2	5,0	5,5	6,5	7,5	8,5	9,5		
par0	0,1	0,45	0,45	0,22	0,25	0,3	0,5	0,5	0,5	0,5	0,5	0,5	5,5	6,5	7,5	8,5	9,5	10,5		
parH	0,3	0,6	0,6	0,4	0,4	0,4	1,0	1,0	1,0	1,0	1,0	1,0	6,5	7,5	8,5	9,5	10,5	11,5		
par <sub>opt</sub>	0,25	0,56	0,3	0,19	0,18	0,12	0,95	0,73	0,61	0,68	0,73	0,87	5,9	7,5	6,7	8,6	9,6	10,2		
$J_{Pontr}$ , kJ	6299						$J_{opt}$ , kJ	6310						Skirtumas, %	0,18					

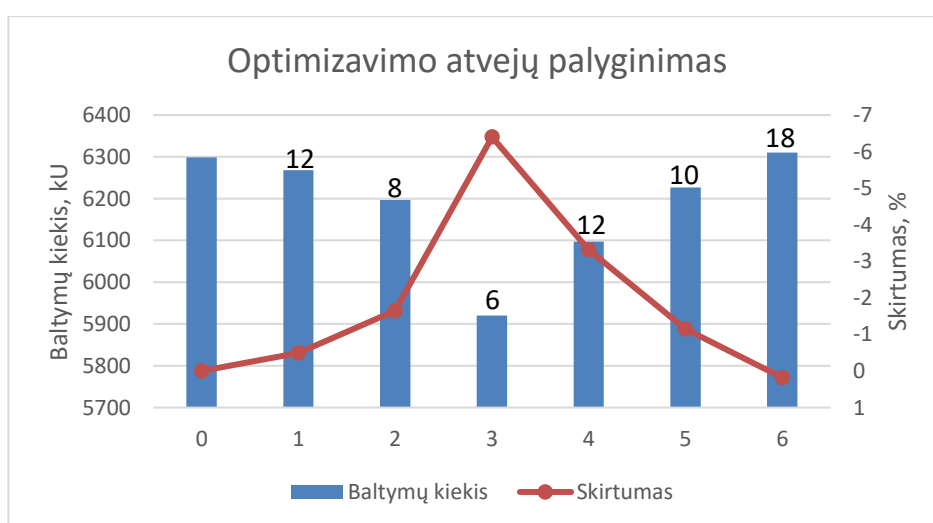
Šeštuoju atveju buvo atliktas aštuoniolikos radialinės funkcijos parametru optimizavimas. Kaip pagrindas buvo pasirinktas šešių funkcijų tinklas, kadangi juo buvo gauti prasčiausi rezultatai iš visų

atvejų. Kaip galima priežastis dėl prastų rezultatų galėjo būti nepalankios radialinių funkcijų centrinės padėties laiko ašyje. Šiuo atveju optimizavimo procedūra taip pat ieškojo ir optimalios varpo padėties. Iš gautų rezultatų matyti, jog varpo padėtis daro didelę įtaką galutiniam modeliavimo rezultatui. Pamaitinimo profilio kreivė gauta artima Pontriagino maksimumo principu apskaičiuotam pamaitinimo profiliui, biomasės koncentracijos, baltymų aktyvumo ir kultūros masės kreivės taip pat mažai skyrėsi nuo optimalių kreivių. Nors baltymų kiekio kreivės po indukcijos kurį laiką tolo viena nuo kitos, modeliavimo pabaigoje gauti beveik identiški rezultatai. Gautas baltymų kiekis modeliavimo metu 0,18% viršijo baltymų kiekį gautą naudojant Pontriagino maksimumo principu apskaičiuotą pamaitinimo profilį.

### 5.3 Rezultatų palyginimas

5.8 lentelė

	Optimizuojamų parametų skaičius				Baltymų kiekis	Skirtumas, %
	$w_i$	$\rho_i$	$c_i$	Iš viso		
1 atvejis	12	-	-	12	6268	- 0,49
2 atvejis	8	-	-	8	6197	- 1,65
3 atvejis	6	-	-	6	5920	- 6,4
4 atvejis	6	6	-	12	6097	- 3,31
5 atvejis	5	5	-	10	6227	- 1,15
6 atvejis	6	6	6	18	6310	0,18



5.11 pav. Optimizavimo atvejų palyginamasis grafikas



Imitacinio modeliavimo atvejų bendri rezultatai pateikti 5.8 lentelėje ir 5.11 paveiksle. Nuliniame stulpelyje pavaizduoti rezultatai gauti naudojant Pontriagino maksimumo principu apskaičiuotą optimalų pamaitinimo profilį. Toliau iš eilės yra nuo 1 iki 6 atvejų rezultatai. Iš jų galima spręsti apie radialinių funkcijų kiekio, funkcijos pločio ir centrinės padėties įtaką neuroninio tinklo aproksimavimo galimybėms. Gerus aproksimavimo rezultatus galima gauti net ir tuomet, kai optimizuojamas tik vienas kintamasis – funkcijos svorinis koeficientas, o kiti parametrai laikomi konstantomis. Tačiau geri rezultatai gaunami tik esant pakankamam funkcijų skaičiui. Iš 5.11 paveikslo galima matyti, kad antruoju ir trečiuoju atveju sumažinus radialinės bazės funkcijų skaičių atitinkamai nuo 12 iki 8 ir 6, neuroninio tinklo lankstumas mažėja, gauti optimizuoti pamaitinimo profiliai, modeliuojamam procesui nebeleidžia priartėti prie maksimalaus galimo gauti baltymų kiekio proceso pabaigoje.

Radialinės bazės funkcijų tinklo aproksimavimo galimybes gali padidinti optimizuojami papildomi kintamieji: funkcijos varpo plotis arba centrinė padėtis. Net ir naudojant mažiau radialinių funkcijų, bet optimizuojant daugiau parametrų galima gauti panašius optimizavimo rezultatus į tuos, kuomet naudojamas didesnis kiekis radialinių funkcijų. Tai galima matyti iš 4, 5 imitacinio modeliavimo atvejo, kuomet buvo optimizuojami šešių ir penkių radialinių bazės funkcijų tinklai su dvejais optimizuojamais parametrais: svoriniu koeficientu  $w_i$  ir varpo pločiu  $\rho_i$ . Matomas aiškus rezultatų pagerėjimas nuo 3-jo atvejo. Nors ketvirtuoju atveju buvo optimizuojama daugiau parametrų nei penktuoju atveju, rezultatai gauti prastesni. Tai galėjo lemti skirtingas funkcijų išdėstymas laiko ašyje.

Geriausieji rezultatai buvo gauti naudojant didžiausią kiekį optimizuojamų parametrų šeštuoju atveju, tačiau optimizavimo procedūra truko ilgiausiai. Įvertinus laiko sąnaudas ir tai, kad pirmuoju atveju buvo gauti panašūs rezultatai, galima teigti, jog šis parametrų kiekis neduoda apčiuomiamo pranašumo, nei pirmuoju atveju.

## Išvados ir rezultatai

Darbo metu buvo sumodeliuotas ir tiksliai atkartotas periodinis su pamaitinimu biotechnologinis procesas. Iš palyginamųjų kreivių galima matyti, kad būsenos kintamieji: biomasės koncentracija, santykinis baltymų aktyvumas bei kultivuojamos kultūros masė labai gerai atitinka referensinius proceso duomenis, tačiau substrato koncentracija nuo 5 iki 7 valandos įgija tam tikrus netikslumus, kurie padaro įtaką ir santykinio biomasės greičio augimo bei deguonies suvartojimo greičio modeliavimo rezultatams. Dėl šių skirtumų modeliuojamo proceso pabaigoje buvo gautas mažesnis galutinio produkto kiekis, tačiau tai sudarė tik -0,45%, todėl galima teigti, kad gautas modelis yra tikslus.

Iš šešių skirtingų imitacinio modeliavimo atvejų, kurių metu buvo naudojami radialinės bazės funkcijų neuroniniai tinklai valdančiajam poveikiui optimizuoti, geriausi rezultatai buvo gauti šeštoju atveju, kuomet buvo optimizuojamas šešių radialinės bazės funkcijų tinklas su aštuoniolika optimizuojamų parametru. Tuomet skirtumas tarp sumodeliuoto baltymų kiekio buvo tik 0,18%. Taip pat geri rezultatai gauti pirmuoju atveju, naudojant dvylikos radialinės bazės funkcijų neuroninį tinklą, kuomet optimizuojama dvylika parametru, visi jie – funkcijos svoriai  $w_i$ . Šiuo atveju baltymų kiekio skirtumas sudarė -0,49 %. Toliau mažinant radialinės bazės funkcijų ir optimizuojamų parametru skaičių modeliavimo rezultatai buvo gaunami vis prastesni.

Galima pastebėti tendenciją, kad radialinės bazės funkcijų tinklui didžiausią įtaką daro bendras optimizuojamų parametru skaičius, o ne kažkuris vienas iš atskirų parametru: svorinis koeficientas, varpo plotis ar centro padėtis. Augant optimizuojamų parametru skaičiui didėja tinklo aproksimavimo galimybės, tačiau tuo pačiu išauga ir optimizavimo procedūros sudėtingumas, bei laikas reikalingas optimaliems parametrų rasti, todėl parametru skaičius kiekvienu skirtingu atveju turėtų būti parenkamas atskirai.

Be to, svarbu paminėti, kad visais nagrinėtais optimizavimo atvejais gauti būsenos kintamųjų  $x$ ,  $s$  ir  $p_x$  trajektorijų nuokrypiai galiniuose taškuose, palyginti su referenciniais duomenimis, yra mažesni už realiam procese matuojamų biomasės ir substrato koncentracijų bei tiksline baltymo aktyvumo matavimo tikslumą (~10%). Todėl galima teigti, kad netgi optimizuojant procesą su minimaliu nagrinėtu optimizuojamų valdančiojo poveikio aproksimacinių parametru skaičiumi (6) gaunami rezultatai, kurie praktinio taikymo požiūriu yra lygiaverčiai ir užtikrina artimą maksimaliai tikslo funkcijos vertę. Tyrimo metu gautos tikslo funkcijos reikšmės niekada paklaidų ribose neviršijo rezultatų, gautų taikant Pontriagino maksimumo principą. Tyrimo rezultatai demonstruoja, kad taikant parametrinio optimizavimo metodiką ir radialinės bazės funkcijų tinklą nagrinėtame biotechnologijame

processe gaunami optimizavimo rezultatai, analogiški referenciniams duomenims. Taip dar kartą patvirtinamas abiejų metodikų rezultatų teisingumas ir atitikimas eksperimentiškai gautiems duomenims.

## Literatūra

1. V. Galvanauskas, D. Levišauskas. Biotechnologinių procesų modeliavimas, optimizavimas ir valdymas. Vilnius, 2008.
2. Hao Yu, Tiantian Xie, Stanisław Paszczyński, Bogdan M. Wilamowski. Advantages of radial basis function networks for dynamic system design. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, VOL. 58, NO. 12, 2011
3. Richard Fiifi Turkson, Fuwu Yan, Mohamed Kamal Ahmed Ali, Jie Hua, „Artificial neural network applications in the calibration of spark-ignition engines” (2016)
4. Jeongseok Lee, Sang Yup Lee, Sunwon Park, Anton P.J. Middelberg. Control of fed-batch fermentations. *Biotechnology Advances* 17 (1999)
5. Boškovic JD, Narendra KS. Comparison of linear, nonlinear and neural-network-based adaptive controllers for a class of fed-batch fermentation processes. *Automatica* 1995;31(6)817–40.
6. D. Levišauskas, V. Galvanauskas, S. Henrich, K. Wilhelm, N. Volk, A. Lubbert, „*Model-based optimization of viral capsid protein production in fed-batch culture of recombinant Escherichia coli*“ (2003)
7. D. Levišauskas, V. Galvanauskas, „Biotechnologinių procesų modeliavimas, optimizavimas ir valdymas. Praktikumai ir uždaviniai“. Vilnius, 2008
8. B.J. Minihane, D.E. Brown, Fed-batch culture technology. *Biotechnology advances, Pergamon Press*, 1986, Vol.4, No.2, 207-218.
9. H.S. Shin, H.C. Lim. Cell-mass maximization in fedbatch cultures. *Bioprocess and Biosystems Engineering*, 2006, Vol.29, 335-347.
10. D. Levišauskas, V. Galvanauskas, G. Žunda, S. Grigiškis. Model-based optimization of biosurfactant production in fed-batch culture *Azotobacter vinelandii*. *Biotechnology Letters*, 2004, Vol.26, No.14, 1141-1146.
11. D. Levišauskas, T. Tekorius. Model-based optimization of fed-batch fermentation processes using predetermined type feed-rate time profiles: a comparative study. *Information Technology and Control*, 2005, Vol.34, 231-236.
12. D. Levišauskas, V. Galvanauskas. Optimization of feed-rate in fed-batch culture *enterobacter aerogenes* 17 e13 for maximization of biomass productivity. *Information technology and control*, 2009, vol.38, no.2, 101-107.
13. D. Fogel. Evolutionary Computation: toward a new philosophy of machine intelligence. *IEEE Press, New York*, 1995.

14. L. Ljung, System identification. *Theory for the user. Englewood Cliffs: Prentice Hall, 1999.*
15. V.V. Biriukov, V.M. Kantere. Optimization of batch processes of microbial synthesis. *Nauka, Moscow, 1985.*
16. J.Nielsen, J. Villadsen. Bioreaction engineering principles. *Plenum Press, New York, 1994.*
17. A. Kanopka. Biotechnologija. [žiūrėta 2017 m. balandžio 20 d.] Prieiga per internetą:  
<http://molbio.vdu.lt/medziaga/Kanopka/Biotechnologija.pdf>

## Priedai

### Priedas 1. Pagrindinė programa „HW\_0”

```
function HW_0
clc
clear all
global tg q1 J par Npop DesirPerform Ntrials eta errmin Ro

Ro=0.5;
tg=[5.0 5.5 6 6.5 7 7.5 8 8.5 9 9.5 10 10.5];
parL=[0.0 0.0 0.0 0.0 0.4 0.15 0.4 0 0.1 0.0 0.10
0.00];
par0=[0.0 0.12 0.15 0.25 0.45 0.2 0.45 0 0.15 0.08 0.12
0.12];

errmin=0;
%err0=hipsc_ident_err2(par0)

% options2=optimset('MaxIter',500,'MaxFunEvals',5000);
% [par,err]=fminsearch(@hipsc_ident_err2,par0,options2);

Npop=300; DesirPerform=-1e5; Ntrials=300; eta=2.0;
%parmat=[parL; par0; parH]';
parmat=[par0*0.8;par0;par0/0.8]';

[par,J]=rvo05(@model_J,parmat); % Initial parameter vector
save q1.dat q1 -ascii
save J.dat J -ascii
save Par11.dat par -ascii
```

### Priedas 2. Evoliucinio optimizavimo programa „rvo05”

```
%
%=====
====
%
% MATLAB 7 BASED SOFTWARE TOOL
% FOR HYBRID MODEL-BASED IDENTIFICATION AND OPTIMISATION
% OF BIOCHEMICAL PROCESSES
%
% v3.0
%
% CREATED BY:
% DR. VYTAUTAS GALVANAUSKAS
% KAUNAS UNIVERSITY OF TECHNOLOGY
% FACULTY OF ELECTRICAL AND CONTROL ENGINEERING
% DEPARTMENT OF PROCESS CONTROL
% mailto:vytautas.galvanauskas@ktu.lt
% http://www.personalas.ktu.lt/~vygal
%
% PROJECT SUPPORTED BY:
% NATO SECURITY THROUGH SCIENCE PROGRAMME
% NATO REINTEGRATION GRANT EAP.RIG.981479
```

```

%
% 2004-2008
%
%=====
====
% Real value optimization algorithm
%
% [Par,Pbest]=rvo3(ObjFunct,parma) is the calling convention
%
% Input:          ObjFunct: function defining the objectiv
%                  parma    : Parameter matrix:
%                          first column lower parameter limit
%                          second column first guess for the
parameters
%                  third column upper parameter limit
% Output:         Par      : Parameter vector determined (Column vector)
%                  Pbest   : Corresponding performance measure
%
%=====
====
function [Par,Pbest]=rvo05(model_J,parma)%rvo05(ObjFunct,parma)
%=====
====
ObjFunct=model_J;
global Penalt Npop DesirPerform RelErr Ntrials NW eta
global errmin numfueval err_mode err_mode2 ident_method

% Npop=10...50
% DesirPerform=-1.e6
% RelErr=1e-6
% Ntrials=100-1000

% Step 1: Create the initial population
% =====
if Ntrials==0

    Pbest=feval(ObjFunct,parma(:,2));      % Performance of the
members
    Par=parma(:,2);
    return;
else

Population=InitPopul(parma,Npop);          % Initial population
PopBcp=Population;
for i=1:Npop
    Par=Population(:,i);
    Performance(i)=feval(ObjFunct,Par);%ObjFunct      % Performance
of the members
end                                          % of the initial
population
numfueval=0;

```

```

[Pworst,Nworst]=max(Performance);           % Worst element of the
population
[Pbest,Nbest]=min(Performance);           % Best element of the
population
disp('Initial population established')

% Some parameters and flags:
expo=1/(1+eta);
flag=1;
itrials=0;
improved=0;
Pcrit=0;
flag_maxit=0;

while flag==1                               % Production loop

% Step 2: Select at random two parents from the population
% =====

i1=ceil(Npop*rand(1));                       % Select first parent
i2=ceil(Npop*rand(1));                       % Select second parent
while i1==i2 i2=ceil(Npop*rand(1)); end      % Repeat until i1.ne.i2
u=rand(1);                                   % Take a random number
[0 1]

% Step 3: Cross-over between the two parents: Generates two children
% =====

if u<=0.5
    bq=(1/(2*(1-u)))^expo;
    % Children
    x(:,1)=0.5*((1+bq)*Population(:,i1)+(1-bq)*Population(:,i2));
    x(:,2)=0.5*((1-bq)*Population(:,i1)+(1+bq)*Population(:,i2));
else
    bq=(2*u)^expo;
    x(:,1)=0.5*((1+bq)*Population(:,i1)+(1-bq)*Population(:,i2));
    x(:,2)=0.5*((1-bq)*Population(:,i1)+(1+bq)*Population(:,i2));

[nparam,dummy]=size(parma);
for j=1:nparam
    if x(j,1) < parma(j,1) x(j,1)=parma(j,1); end;
    if x(j,2) < parma(j,1) x(j,2)=parma(j,1); end;
end

end

Perf(1)=feval(ObjFunct,x(:,1));             % Fitness of child 1
Perf(2)=feval(ObjFunct,x(:,2));           % Fitness of child 2

% Step 4: Selections process
% =====

```



```

for jc=1:2
    if Perf(jc)<Pworst
        Population(:,Nworst)=x(:,jc);           % Replace the worst
element
        Performance(Nworst)=Perf(jc);           % Update its performance
value
        if Perf(jc)<DesirPerform
            disp('Desired performance obtained');
            flag=0;
        end
        if Perf(jc)<Pbest
            Pbold=Pbest;
            Pbest=Perf(jc);
            Pcrit=Pbold-Pbest;
            improved=improved+1;
            if improved>5
                if Pcrit<RelErr
                    disp('No significant further improvement found')
                    flag=0;
                end
            end
            Par=x(:,jc);
            flag_maxit=1;
        end
        [Pworst,Nworst]=max(Performance);       % New worst element
    end
end

% Step 5: Test for the termination criteria
% =====

% Number of trials criterion
itrials=itrials+1;
if itrials>Ntrials
    if flag_maxit==0 Par=PopBcp(:,Nbest); end
    disp('Max # of iterations reached')
    flag=0;
end %if itrials

end % while flag==1
end
%-----> End of RealValOpt01 <-----
-----

function Population=InitPopul (parma,Npop)
%
% InitPopul determines the initial population for an evolutionary
% search algorithm
% Population is the result
% It contains the first guess as its first column,
% All other members are homogeneously distributed within

```

```

%           the feasibility region provided by parma
%
% parma     Parameter Matrix containing 3 columns
%           1:   Lower bound of the parameters
%           2:   Initial guess of the parameter values
%           3:   Upper bound of the parameters
% Npop      Size of the population = number of individuals
%
Population(:,1)=parma(:,2);
dpar=parma(:,3)-parma(:,1);

for i=2:Npop
    Population(:,i)=parma(:,3)-dpar.*rand(size(dpar));
end

```

### **Priedas 3. Biotechnologinio proceso modeliavimo programa „model“**

```

function dc_dt=model(t,c)
global tg q OUR mumax time_ind sf ki T Tref ind
global alfa ks Yxs m Tpx km kimu kmu mox Yox
global wi Ro
%-----Modelio parametrai
mumax=0.737;
x0=0.116;
s0=9.5;
w0=5.05;
time_ind=8;
sf=151;
ki=93.8;
Tref=37;
alfa=0.0495;
ks=0.00333;
Yxs=0.46;
m=0.0242;
Tpx=1.495;
km=751;
kmu=0.61;
kimu=0.0174;
%-----Modelio parametrai baigiasi
if t < time_ind
    Fe = -0.002;
    if t<4.0
        Fsmpl = -0.0;
    else
        Fsmpl = -0.015;
    end
else
    Fe = -0.005;
    Fsmpl = -0.045;
end

if t < time_ind

```

```

        mox=0.15;
        Yox=0.5;
    else
        mox=0.015;
        Yox=0.75;
    end
    if t<time_ind, T=Tref; else T=32; end
%=====
===
% Redefinition of the state variables for convinience
x=c(1); s=c(2); px=c(3); w=c(4);
mu=mumax*(s/(ks+s))*(ki/(ki+s))*exp(alfa*(T-Tref));
qs=(mu/Yxs)+m;
pmax=(mu*km)/(kmu+mu+(mu*mu/kimu));
qpx=(pmax-px)/Tpx;
F2=Fsm;
F1=0.001*mu*x*w+Fe;
ut=0;
OUR=(Yox*mu*x)+(mox*x);
if t<time_ind
    qpx=0;
end
%Gauso f-jos parametrai ir skaiciavimas
psi=0.001;
F=0.0;
umax=0.7;
for ind=1:length(tg)
    fi=exp(-(t-tg(ind)).^2./(Ro^2+psi));
    F=F+(wi(ind)*fi);
    if F>=umax
        F=umax;
    end
end
end
%---Dif lygtys su RBF reikšmemis-----
dc_dt(1,1) = mu*x-(F+F1)*x/w; %dx/dt
dc_dt(2,1) = -qs*x+F*(sf/w)-(F+F1)*s/w; %ds/dt
dc_dt(3,1) = qpx; %dpx/dt
dc_dt(4,1) = F+F1+F2; %dw/dt
q=[q;mu OUR ut x s px w qs F];

```

#### Priedas 4. Baltymų kiekio modeliavimo programa „model\_J”

```

function J=model_J(Par)
global q1 q OUR wi tg errmin

%----- Baudos funkcija
PenWt=100;
OURmax=6.0;

wi=Par(1:6); %svoriniai koeficientai
%Ro=Par(6:end); %funkcijos plotis
tg=Par(7:end); %centrinė funkcijos padėtis

```

```

eps=1e-6;
options = odeset('RelTol', eps);

dt=0.025;
tspan=[0.0:dt:10.0];           % [h]      Process time span

%      [x      s  px      w ];
c0 =      [0.116 9.5 0.0 5.05]; % Initial concentrations / volume

% Call of ODE numerical integration function
c00=c0; c=c0;
t=tspan(1);
q1=zeros(1,9); % Vector of additional variables

Penalt=0.0;

for k=1:length(tspan)-1
    q=[];
    tspan0=tspan(k:k+1);
    [t01,c01]=ode15s(@model,tspan0,c00,options);

    c00=[c01(end,:)']';
    t=[t;t01(end)];
    c=[c;c01(end,:)];
    q1=[q1;q(end,:)];

    OUR=q(end,2);

    if OUR>=OURmax,
        Penalt=Penalt+PenWt*(OUR-OURmax)^2;
    end

end

J=(-c01(end,1)*c01(end,3)*c01(end,4))+Penalt; %px*x*w
if J<errmin,
    errmin=J
    Penalt
    save Par.dat Par -ascii
    save J_model.dat J -ascii
    save Penalt.dat Penalt -ascii
end

```