



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

**Benas Paulius**

**BIOTECHNOLOGINIO PROCESO HIBRIDINIO MODELIO  
IDENTIFIKAVIMAS**

Baigiamasis magistro projektas

**Vadovas**

Prof. dr. Vytautas Galvanaukas

**KAUNAS, 2017**

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

**BIOTECHNOLOGINIO PROCESO HIBRIDINIO MODELIO**  
**IDENTIFIKAVIMAS**

Baigiamasis magistro projektas  
Valdymo technologijos (kodas 621H66001)

**Vadovas**

Prof. dr. Vytautas Galvanauskas

**Recenzentas**

Prof. dr. Donatas Levišauskas

**Projektą atliko**

Benas Paulius

**KAUNAS, 2017**



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Elektros ir elektronikos

(Fakultetas)

**Benas Paulius**

(Studento vardas, pavardė)

**Valdymo technologijos, 621H66001**

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Biotechnologinio proceso hibridinio modelio identifikavimas“

**AKADEMINIO SAŽININGUMO DEKLARACIJA**

20 17 m. gegužės 10 d.  
Kaunas

Patvirtinu, kad mano **Beno Pauliaus** baigiamasis projektas tema „Biotechnologinio proceso hibridinio modelio identifikavimas“ 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)

Paulius, Benas. Biotechnologinio proceso hibridinio modelio identifikavimas. *Valdymo sistemų magistro* baigiamasis projektas / vadovas prof. dr. Vytautas Galvanaukas; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, Automatikos katedra.

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

Reikšminiai žodžiai: *biotechnologija, rekombinuotų baltymų aktyvumas, identifikavimas, jautrumo lygčių metodas, dirbtiniai neuroniniai tinklai.*

Kaunas, 2017. 55 p.

## SANTRAUKA

Darbe apžvelgiami biotechnologiniai procesai, bioreaktoriai ir jų tipai, išskiriami matuojami kintamieji ir vykstančios reakcijos. Išnagrinėjamos procesų masės balanso lygtys ir reakcijų greičių modeliai. Pateikiamos tipinės hibridinių modelių struktūros ir jų identifikavimo metodai.

Tyrimams pasirinktas periodinis su pamaitinimu, rekombinuotų baltymų gamybos procesas. Tyrimui buvo pateiktas mechanistinis proceso modelis su atliktais eksperimentiniais duomenimis. Pasitelkiant proceso masės balanso, reakcijų greičių lygtimis ir kitais parametrais, mechanistinis modelis modifikuotas į hibridinį su dirbtiniu neuroniniu tinklu (DNT). Pagrindinis tyrimo tikslas – identifikuoti proceso hibridinį modelį. Identifikavimas atliktas *Levenberg – Marquardt* algoritmu, kurio rezultatai buvo patikslinti jautrumo lygčių metodu. Įsitikinti, jog tyrimams parinkti tinkamos struktūros dirbtiniai neuroniniai tinklai, atlikti mokymo ir testavimo bandymai. Pagrindinis dėmesys buvo skirtas rekombinuotų baltymų sintezės greičio DNT modeliui, kurio pagrindiniai įėjimo dydžiai yra santykinis biomasės augimo greitis ir rekombinuotų baltymų aktyvumas.

Tyrimo išvadose palygintos mechanistinio ir dinaminio modelių vidutinės absoliutinės ir vidutinės procentinės absoliutinės paklaidos. Aptartas identifikuoto modelio praktinis panaudojimas.

Paulius, Benas. IDENTIFICATION OF BIOTECHNOLOGICAL PROCES HYBRID MODEL: *Master`s thesis in Control System / supervisor prof. dr. Vytautas Galvanaukas. 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, recombinant protein production, identification, sensitivity equation approach, artificial neural network.*

Kaunas, 2017. 55 p.

## SUMMARY

In the Master's thesis an overview of biotechnological processes, bioreactors and their types is provided, as well as measured variables and the ongoing reactions are described. Further, the equations of mass balance of the processes and reaction rate models are analyzed. Typical structures of hybrid models and their identification methods are provided.

The fed-batch process of recombinant protein production was chosen as an object of the investigation. For the research, a mechanistic model of the process and the experimental data from the literature were used. Using the reaction rate equations, mass balance equations and model parameters, the mechanistic model was modified into a hybrid model with an artificial neural network (ANN). The main aim of the study is to identify hybrid model of the process. Identification was carried by using *Levenberg - Marquardt* algorithm, the results of which were adjusted by sensitivity equation approach. Within the study, it was ensured that the appropriate structure of artificial neural networks is chosen. Training and testing of the artificial neural networks were conducted. The major emphasis was given for ANN model of specific protein accumulation rate, which has two input variables: the specific biomass growth rate and specific protein activity.

The conclusion of the study gives a comparison the mean absolute and mean percentage absolute errors of mechanistic and hybrid models. The practical use of the identified model was considered.

# Turinys

ĮVADAS.....	8
1. LITERATŪROS APŽVALGA .....	9
1.1. Biotechnologija, procesai ir bioreaktoriai .....	9
1.1.1. Biotechnologija .....	9
1.1.2. Bioreaktoriai.....	10
1.1.3. Biotechnologinių parametrų tiesioginio matavimo priemonės ir netiesioginio įvertinimo metodai .....	11
1.2. Biotechnologinių procesų matematinis modeliavimas ir identifikavimas .....	14
1.2.1. Medžiagų masės balansai .....	14
1.2.2. Reakcijos greičių modeliavimas .....	16
1.2.3. Dirbtiniai neuroniniai tinklai (DNT).....	19
1.2.4. Neraiškių aibių logika .....	20
1.2.5. Tipinių hibridinių modelių struktūra.....	22
1.2.6. Modelių struktūros ir parametrų identifikavimas.....	23
2. HIBRIDINIO MODELIO IDENTIFIKAVIMAS .....	31
Tiriamosios dalies sutartinių ženklų ir matavimo vienetų sąrašas .....	31
2.1. Mechanistinis modelis.....	33
2.1.1. Mechanistinio modelio modeliavimo rezultatai.....	35
2.2. Rekombinuotų baltymų sintezės greičio modeliavimas $q_{px}$ dirbtiniu neuroniniu tinklu (DNT).....	37
2.2.1. DNT apmokymas Levenberg – Marquardt algoritmu.....	38
2.2.2. Hibridinio modelio parametrų identifikavimas jautrumo lygčių metodu ....	39
2.2.3. Mokymo ir testavimo rezultatai .....	43
2.2.4. Mokymo ir testavimo rezultatų palyginimas .....	47
2.3. Mechanistinio ir hibridinio modelio rezultatų palyginimas .....	49
IŠVADOS .....	53
INFORMACIJOS ŠALTINIŲ SĄRAŠAS .....	54
PRIEDAI.....	56
Priedas nr. 1, Matlab .m programinis kodas.....	56
Priedas nr. 2, Simulink blokų programiniai kodai.....	58

Priedas nr. 3, Simulink modelis.....	60
Priedas nr. 4, Levenberg – Marquart algoritmo programinis .m kodas.....	61
Priedas nr. 4, Simulink modelis.....	62
Priedas nr. 5, Simulink / Matlab function ir DEE blokų programiniai kodas .....	63
Priedas nr. 6, Matlab .m programinis kodas.....	64
Priedas nr. 7, Matlab .m programinis kodas.....	67

## IVADAS

Biotechnologija – mokslas, nagrinėjantis mikroorganizmų pritaikymą biotechnologiniuose procesuose. Ši mokslo šaka pritaikoma medicinos, farmacijos, maisto pramonės, žemės ūkio srityse. Biotechnologija tapo žinoma prieš mūsų erą, Mesopotamijoje, kai gaminant alui ar duonai buvo panaudotos mielės.

Biotechnologijos mokslas šiomis dienomis yra labai plačiai išsivystęs. Produktų, pagamintų naudojantis biotechnologija galime sutikti kiekvieną dieną, pradedant nuo maisto produktų ar mineralų iki degalų gamybos. Terminas „biotechnologija“ puikiai žinomas medicinoje, kur pritaikoma vaistų gamybai, mineralams, vitaminams išgauti, genetiniams tyrimams atlikti. Taip pat pritaikoma žemės ūkyje, kai genetiškai modifikuojami pasėliai ar augalai. Išrandamos naujos genetiškai modifikuotos maistinės kultūros rūšys, kurios gali būti atsparios tam tikriems kenkėjams, ligoms, stresinėms aplinkoms bei cheminiams priedams.

Iš apžvelgtų sričių matyti koks yra platus biotechnologinės mokslo srities pritaikymas ir galime suprasti kokie tai sudėtingi procesai, kuriuose vyksta begalės reakcijų. Norint užtikrinti biotechnologinių reakcijų vyksmą, svarbu parinkti tinkamus produktus gamybai. Kokybiškiems produktams reikia didelių išteklių, todėl svarbu sukurti efektyvius proceso modelius, kurie maksimizuotų ir išgautų kokybišką tikslinį produktą su mažiausiomis išlaidomis.

Yra sukurta daug įvairių hibridinių procesų modelių, tačiau šiame darbe taikomas modelis su dirbtiniu neuroniniu tinklu, kuris identifiukuotas *Levenberg – Marquardt* algoritmu ir jautrumo lygčių metodu.

### **Tyrimo tikslas**

Naudojantis proceso mechanistiniu modeliu sukurti hibridinį proceso modelį su dirbtiniu neuroniniu tinklu bei identifiukuoti modelio parametrus.

### **Tyrimo uždaviniai**

1. Atlikti literatūros analizę. Išanalizuoti biotechnologinius procesus, bioreaktorių tipus, vykstančias reakcijas. Pateikti tipinių hibridinių modelių struktūras, jų identifikavimo metodus.
2. Naudojantis pateiktu mechanistiniu modeliu, *Matlab / Simulink* terpėje sukurti hibridinį su dirbtiniu neuroniniu tinklu proceso modelį.
3. Identifiukuoti proceso modelį, atlikti mokymo ir testavimo bandymus.
4. Atlikti rezultatų analizę ir apibendrinimą, pateikti išvadas.



# 1. LITERATŪROS APŽVALGA

## 1.1. Biotechnologija, procesai ir bioreaktoriai

### 1.1.1. Biotechnologija

Biotechnologija – mokslas, nagrinėjantis mikroorganizmų pritaikymą biotechnologiniuose procesuose. Ši mokslo šaka pritaikoma medicinos, pramonės, žemės ūkio, aplinkosaugos srityse. Biotechnologija tapo žinoma prieš mūsų erą, Mesopotamijoje, kai gaminant alui ar duonai buvo panaudotos mielės.

Biotechnologijos sąvoka galima apibrėžti, kaip gyvų biologinių organizmų taikymą modifikuotų produktų tobulinimui. Biotechnologijos mokslas yra glaudžiai susijęs ir su kitais mokslais kaip biochemija, genetika, biologija, biomedicina, bioinformatika, molekulių, audinių ir procesų inžinerija. Kaip pavyzdžiui, biotechnologija yra nukreipta išgauti organiniams produktams, tai pienui ar alui. Dar vienas pavyzdys, bakterijos naudojamos pramoninėje kasyboje, tai druskų išvalymui. Taip pat naudojama perdirbti atliekoms ar net išgauti biologiniams ginklams [1].

Galime išskirti pagrindines biotechnologijos panaudojimo sritis: 1) Medicina, biotechnologijos padeda išrasti ir išgauti naujus vaistus, panaudojama genetiniuose tyrimuose. Panaudojama farmakogenetikoje, kur tirama individo atsaką į vaistus. Genetikoje tiriamos paveldimumo ligos, vaikų tėvystės. 2) Žemdirbystė. Naudojami genetiškai modifikuoti augalai, siekiama įvesti naują rūšį, kuri natūraliai neatsiranda. Tobulinamos augalų rūšys, kurios gebėtų būti atsparios tam tikriems kenkėjams, ligoms, stresinėms aplinkoms, atsparios trąšoms ar būtų išgaunamas didesnis derlius. Taip pat panaudojama biokurui išgauti. 3) Pramonė daugiausia skiriama dėmesio fermentaciniams procesams. 4) Dar galima išskirti aplinkosaugą, kur panaudojama atliekų valymui, diegiamos technologijos padedančios taupyti gamtos resursus. Kartais kyla ginčai ar genetiškai modifikuoti organizmai nepakenks natūraliai gamtos įvairovei. [2].

Biotechnologijoje naudojamų biologinių medžiagų yra labai plati įvairovė, tai ląstelės komponentai (DNR, RNR, baltymai, polisacharidai, lipidai ir kitos mažos molekulinės masės medžiagos), mikroorganizmai, augalų ir gyvūnų ląstelės, daugialąsčiai organizmai (augalai ir gyvūnai). Vieni iš daugiausia pasaulyje tyrinėjamų mikroorganizmų yra *E. coli*, jų tyrinėjimų metu išgauta labai daug informacijos apie genetiką. Šios bakterijos geba daugintis tiek esant aerobinėms sąlygoms, tiek anaerobinėms [3].

### **1.1.2. Bioreaktoriai**

Bioreaktorius – dažniausiai būna cilindro formos indas, kurio tūris svyruoja nuo litro iki tūkstančio litrų, gana dažnai būna pagamintas iš nerūdijančio plieno. Bioreaktoriuose vyksta biocheminiai procesai, kurių metu auginamos bakterijos, ląstelės ar audiniai. Bioreaktoriai klasifikuojami pagal jų veikimą, su pamaitinimu, be pamaitinimo, periodinio, nenutrūkstamo veikimo. Bioreaktoriuje augantys organizmai gali būti panardinti į skystąją terpę ar pritvirtinti prie kietos terpės paviršiaus. Panardinta kultūra gali būti suspenduota ar imobilizuota.

Bioreaktoriuje vykstantis procesas reikalauja optimalių sąlygų, norint užtikrinti tinkamą reakcijų eigą. Tam kad užtikrinti geras sąlygos reaktoriuose matuojami ir valdomi technologiniai parametrai, kaip pH, temperatūra, ištirpusio deguonies kiekis, slėgis, įvairių medžiagų koncentracijos ir kt. [4]

#### **Bioreaktorių klasifikacija**

##### *Anaerobiniai ir aerobiniai bioreaktoriai*

Anaerobiniuose reaktoriuose sistema uždara, iš aplinkos nėra tiekiamas deguonis ir jis negali patekti į reaktorių. Šiuose reaktoriuose yra įrengtos sistemos pašalinančios nereikalingas dujas. Anaerobiniai reaktoriai mažiau jautrūs dideliems apkrovimams, nereikalauja didelių energijos sąnaudų, bet metabolizmas lėtesnis, reakcijos užtrunka ilgiau.

Aerobiniuose reaktoriuose į sistema dirbtiniu būdu tiekiamas deguonis. Aerobiniuose sistemose procesai vyksta sparčiau. Tiekiamas deguonis padeda sumaišyti fermentacijos terpę ir greičiau pašalinamos nereikalingos medžiagos. Šie reaktoriai dažniau naudojami praktikoje, tačiau deguonies tiekimo sistemos reikalauja papildomų energijos sąnaudų [4].

##### *Nenutrūkstamo veikimo bioreaktoriai*

Vienas iš pavyzdžių yra chemostatinis bioreaktorius. Reaktorius nuolat papildomas šviežia terpe, o gautas produktas yra nuolat šalinamas, papildomas srautas lygus pašalintam. Atsižvelgiant į papildomus ir ištekančius srautus, tiekiamų medžiagų cheminė sudėtis parenkama tokia, kad reaktoriuje reakcijos vyktų pastovioje būsenoje, t.y. nekistų medžiagų koncentracijos ir augimo greičiai. Reaktoriai dar skirstomi į tiesinius ir idealaus sumaišymo, kai terpės nekintantis režimas užtikrinamas maišyklėmis [4].

##### *Periodinio ir periodinio su pamaitinimo bioreaktoriai*

Šiuose reaktoriuose nuosekliai vykdomi visi proceso etapai. Visų pirma, reaktorius yra sterilizuojamas, užpildomas terpe, vyksta cheminės reakcijos, gryninamas ir galiausiai pašalinamas produktas. Visi reaktoriaus veikimo etapai vyksta periodiškai, t.y. įvykus vienam

ciklui, vykdomas kitas. Reaktoriuose naudojamos maišyklės, prie kurių naudojami pritvirtinti papildomi ašmenys, suskaidantys srautą. Maišyklėmis su ašmenimis užtikrinamos efektyvesnės reakcijos, tačiau tai reikalauja aušinimo sistemų, kurios sukelia didelius energijos nuostolius.

Periodinio veikimo bioreaktoriuje, pamaitinimo srautas yra užpildomas prieš ciklą, o periodinio su pamaitinimu yra tiekiamas viso ciklo metu. Pastarojo veikimo reaktoriai užtikrina didesnės tikslinio produkto koncentracijas ir kiekius [5].

#### *Pusiau periodinio veikimo bioreaktoriai*

Pusiau periodinio reaktoriai veikia panašiai kaip periodinio veikimo. Procesas reaktoriuje vykdomas tol kol pasiekama tikslinio produkto koncentracija, tuomet pašalinama dalis terpės, o kita dalis lieka sekančiam ciklui. Sekančiam ciklui bioreaktorių papildomas nauja kultivavimo terpe [5].

#### *Bioreaktoriai su imobilizuota ar suspenduota mikroorganizmų masė*

Reaktoriuje imobilizuota terpė yra įtvirtinta ant paviršiaus ir viso proceso metu lieka neištraukiama. Suspenduota mikroorganizmų masė yra kultivavimo terpėje, proceso metu tik iš dalies gražinama į bioreaktorių [4].

#### *Konstruktiniai bioreaktorių ypatumai*

Bioreaktoriuose naudojami įvairių tipų konstrukciniai įrenginiai. Įrengiami maišymo įrenginiai, maišant terpę padidinamos deguonies tirpumas.

Bioreaktoriuose naudojamos įkrovos, ant įkrovos paviršiaus laikomi mikroorganizmai.

Prietaisai tiekiantys į bioreaktorių deguonį: difuzoriai – montuojami reaktoriaus apačioje ir tiekiantys deguonį; aeriliftiniai tiekimo prietaisai – tiekia deguonį tik tam tikrose vietose.

### **1.1.3. Biotechnologinių parametų tiesioginio matavimo priemonės ir netiesioginio įvertinimo metodai**

#### *Kintamųjų matavimo būdai*

Tam, kad vyktų efektyvios biotechnologinių procesų reakcijos, būtina užtikrinti tinkamas sąlygos procesui vykti. Svarbu tinkamai įvertinti ir valdyti matuojamus parametrus. Biotechnologiniai parametrai gali būti įvertinami tiesioginėmis ir netiesioginėmis matavimo priemonėmis, kai išmatavus vieną ar kelis parametrus, naudojant funkcinę priklausomybę randamas atitinkamas parametras.

Tiesioginio matavimo priemonės yra jutikliai, jie gali būti montuojami reaktoriuje (*angl. in-situ*) ar už jo (*angl. ex-situ*), t.y. medžiagos įvertinamos ištrauktos iš reaktoriaus. *In-situ*

įvertinami fizikiniai ir cheminiai parametrai, o *ex-situ* įvertinamos išėinančių dujų koncentracijos, srautai, masės spektrometrijos.

Parametrų įvertinimas dar gali būti skirstomas į realiu laiku (angl. *on-line*) ar diskrečia laiko forma (angl. *off-line*) [6].

### **Išskiriami matuojami technologiniai kintamieji:**

#### *Temperatūra*

Tai vienas iš svarbiausių ir dažniausiai matuojamų proceso parametrų, todėl temperatūra turi būti palaikoma  $\pm 0,5$  °C tikslumu. Temperatūra gali svyruoti diapazonu 0 – 130 °C. Matavimai atliekami naudojantis termoelementais arba termometrais. Vienas iš patikimiausių temperatūros įvertinimų yra varžiniai elementai, pvz.: platinos jutikliai. Tokie jutikliai vadinami, Pt – 100 arba Pt – 1000, varža atitinkamai 100  $\Omega$  arba 1000  $\Omega$  ties 0 °C riba [7].

#### *pH*

pH yra vienas iš dažnai valdomų kintamųjų bioreaktorių, kadangi fermentų veikla yra labai jautri pH pokyčiams. pH matuojamas naudojant potenciometrinę elektrodą pripildytą skysto arba želinio elektrolito. Stiklo elektrodai panardinti į tirpalą sukuria želinį sluoksnį su judančiais vandenilio jonais. pH pokyčiai sukelia jonų difuzijos procesą, kuris sukuria elektrodo potencialą. Potencialas yra dažniausiai matuojamas Ag / AgCl elektrodo atžvilgiu. pH gali būti palaikomas kelių šimtųjų tikslumu, jei yra pakankamai dažnai maišomas [6].

#### *Slėgis*

Mikroorganizmams slėgio pokytis neturi didelės reikšmės, jei tai neviršija kelių barų. Tačiau parcialinis slėgis turi įtakos dujų tirpumui. Viršslėgis procese gali sukelti infekciją. Sterilizacijos metu slėgis yra vienas iš svarbiausių parametrų, norint užtikrinti saugų reaktoriaus veikimą. Slėgį sterilizacijos procesų metu galima matuoti pjezoelektriniais, talpiniais, varžiniais jutikliais, bet nei vienas iš jų negali tiksliai įvertinti jei yra temperatūros pokyčių [6].

#### *Parcialinis deguonies slėgis ( $pO_2$ )*

Deguonies tirpumas vandens tirpaluose yra žemas ( 36 mg l<sup>-1</sup> bar<sup>-1</sup> esant 30 °C temperatūrai gryname ore). Norint užtikrinti pakankamą deguonies kiekį, būtina tiekti dirbtiniu būdu. Matuoti parcialinį deguonies slėgį dažniausiai naudojamas impulsinis redukuojantis potencialas, deguonis redukuojamas katodu [7].

### *Parcialinis anglies dioksido slėgis (pCO<sub>2</sub>)*

CO<sub>2</sub> daro įtaką mikroorganizmų anabolizmui, katabolizmui ir morfologijai. pCO<sub>2</sub> įvertinamas netiesioginiu būdu pagal pH bikarbonatinio buferio, pH vertė mažėja nuo didesnio anglies dioksido kiekio [7].

### *Biomosės koncentracija*

Daugelis biomasės matavimų principų yra pagrįsti optinių terpių filtravimo savybių, dielektrinėmis ląstelių savybėmis bei terpės tankio pokyčiais. Biomasė gali būti matuojama *ex-situ* arba *in-situ*.

Biomosės koncentraciją netiesioginiu matavimu galima įvertinti naudojant dalelių matmens spektro analizę, pagal deguonies sunaudojimo greitį (angl. *oxygen uptake rate, OUR*), iš buferinio šarmo tirpalo sunaudojimo greičio (angl. *base consumption rate, BCR*), įvertinama pagal iš reaktoriaus išeinančio CO<sub>2</sub> kiekio [6].

### *Kiti matavimai*

Masių spektrometrija taikoma kiekybiniam dujų įvertinimui, tokių kaip pO<sub>2</sub>, pCO<sub>2</sub>, pN<sub>2</sub>, pH<sub>2</sub>, pCH<sub>4</sub>, H<sub>2</sub>S ar lakioms medžiagoms (alkoholiui, acetonui).

Kultūrinės terpės fluorescencijos matavimai naudojami bioreaktoriaus techninės savybės apibūdinti ir pagrindinėms mikroorganizmų savybėms tirti.

Įpurškiamo srauto analizę galima panaudoti gliukozės ir biomasės koncentracijai nustatyti, vandens kokybei stebėti bei acetato koncentracijai nustatyti.

Substrato koncentracija įvertinama fermentų reakcijų naudojimu matuokliuose. Netiesiogiai įvertinama iš deguonies sunaudojimo greičio matavimų.

Rekombinuotų baltymų koncentracija įvertinama iš SDS elektroforezės. Netiesioginiu matavimu įvertinama iš dalelės spektro analizės. Taip pat dar galima įvertinti iš deguonies sunaudojimo greičio matavimų, sudarius produkto masės balanso lygtį.

Optinio tankio jutikliai veikia remiantis perduota, atspindėta ir išsklaidyta šviesa. Bakterijoms matuoti naudojami spektro spinduliai, didesniems mikroorganizmams – infraraudonieji spektro spinduliai, didelių augalinės kilmės ląstelių koncentracija išmatuojama įvertinant terpės drumstumą. Šiuo metodu galima įvertinti ir biomasės koncentraciją.

Naudojant kalorimetrą galima išmatuoti aktyvios biomasės koncentraciją [7].

## 1.2. Biotechnologinių procesų matematinis modeliavimas ir identifikavimas

### 1.2.1. Medžiagų masės balansai

Masės balanso lygtys paremtos medžiagų tvermės dėsniu. Remiantis šiuo dėsniu, medžiagos kiekis gali kisti dėl vykstančių reakcijų, dėl įtekančio ar ištekančio srauto tam tikros medžiagos. Sudaroma diferencialinė medžiagos  $C_1$  masės balanso lygtis:

$$\frac{d(C_1V)}{dt} = q_1C_2V + C_{in1}F_{in} - C_1F_{out}, \quad (1.1)$$

Pirmosios medžiagos koncentracija -  $C_1$  [g/l], tūris -  $V$  [l]. Medžiagos kiekis -  $C_1V$  [g], kitimo greitis  $d(C_1V)/dt$  [g/h]. Kairioji lygties pusė parodo, kiek masės vienetų per laiko vienetą pakinta medžiagos  $C_1$  masė reaktoriuje. Pirmas narys dešinėje reiškia koku dydžiu pakinta medžiagos  $C_1$  masė.  $q_1$  [g/gh] – santykinis reakcijos kitimo greitis medžiagos  $C_2$  atžvilgiu. Antras narys dešinėje apibūdina kiek medžiagos  $C_1$  yra tiekama į reaktorių [g/l].  $F_{in}$  – tūrinis įtekančios medžiagos srautas [l/h]. Trečias narys dešinėje apibūdina kiek medžiagos  $C_1$  išteka iš reaktoriaus [g/h].  $F_{out}$  – tūrinis ištekančios medžiagos srautas [l/h].

Išdiferencijavus 1.1 lygtį ir pertvarkius, gaunama tūrio kitimo diferencialinė lygtis:

$$\frac{dV}{dt} = F_{in} - F_{out}, \quad (1.2)$$

Terpės tūrio pokytis vyksta dėl ištraukiamų ir tiekiamų srautų.

Toliau pateikiama medžiagos koncentracijos  $C_1$  kitimo laike formulė:

$$\frac{dC_1}{dt} = q_1C_2 + \frac{F_{in}}{V}(C_{in1} - C_1), \quad (1.3)$$

Medžiagų masės balansai yra skirtingi tipiniams biotechnologiniams procesams, jie skirstomi panašiai, kaip ir bioreaktoriai: 1) periodinis su pamaitinimu, 2) periodinis su pakartotiniu pamaitinimu, 3) periodinis procesas, 4) nenutrūkstamas procesas [8].

#### *Periodinis su pamaitinimu*

Šis procesas atitinka anksčiau minėtą periodinio su pamaitinimo bioreaktoriaus veikimą. Proceso metu yra tiekiamas substratas pamaitinimui ir dalis kultivavimo terpės yra ištraukiama. Pamaitinimo substratai yra skirtingi, todėl yra ir skirtingos medžiagų koncentracijos. Galima aprašyti masės balanso formulę tam tikrai medžiagai, analogiškai sudaromos ir kitoms medžiagoms [8]:

$$\frac{dC_i}{dt} = q_iC_i + \frac{F_{ini}}{V}C_{ini} - \frac{F_{in}}{V}C_i, \quad (1.4)$$

Suminis srautas visų įtekančių medžiagų substratų:

$$F_{in} = \sum_{i=1}^n F_{ini} \quad (1.5)$$

Terpės tūrio pokytis lygus:

$$\frac{dV}{dt} = F_{in} - F_{out}. \quad (1.6)$$

#### *Periodinis pakartotinis su pamaitinimu*

Proceso medžiagų koncentracijos, įtekančių srautų lygtys tokios pat, kaip ir periodinio su pamaitinimu proceso. Šio proceso skiriasi terpės pokyčio lygtis, kadangi yra ištraukiamas didesnis kultivavimo terpės kiekis. Prie mėginių ėmimo dažnio pridedamas tolydus srautas  $F_{wdr}$ , tuomet gaunamas ištekancios medžiagos srautas  $F_{out}$  [8]:

$$F_{out} = F_{smp} + F_{wdr}. \quad (1.7)$$

#### *Periodinis procesas*

Šio proceso metu nėra tiekiamas pamaitinimo substratas, todėl lygtyse nebelieka tiekiamų pamaitinimo substratų [8]:

$$\frac{dC_i}{dt} = q_i C_i - \frac{F_{in}}{V} C_i, \quad (1.8)$$

Terpės tūrio pokytis lygus:

$$\frac{dV}{dt} = F_{in} - F_{out} \quad (1.9)$$

#### *Nenutrūkstamas procesas*

Šiame procese ištraukiamos medžiagos kiekis yra lygus tiekiamos medžiagos kiekiui. Kultivavimo terpės tūris nepakinta ( $dV/dt=0$ ), tai  $F_{in}/V$  santykis žymimas  $D$  raide.  $D$  – terpės skiedimas. Šiam procesui kiekviena medžiagos balanso lygtis gali būti užrašoma [8]:

$$\frac{dC_i}{dt} = q_i C_i - D(C_{iin} - C_i) \quad (1.10)$$

Terpės tūrio pokytis lygus:

$$\frac{dV}{dt} = 0 \quad (1.11)$$

## 1.2.2. Reakcijos greičių modeliavimas

### *Kinetiniai (mechanistiniai) modeliai*

Vykstant reakcijoms susidaro naujos ląstelės, kurios dauginasi, dalijasi. Iškyla klausimas, kaip apibūdinti ląstelių skaičių. Yra priimta ląstelių skaičiui nustatyti - apskaičiuoti biomasę. Ląstelių skaičius įvertinamas biomasės atžvilgiu.

Vienas iš ląstelių įvertinančių matematinių modelių yra Monod, kuris apibūdina mikroorganizmų augimą. Šį modelį pasiūlė Jacques Monod susieti su mikroorganizmų augimo greičiu vandeninėje terpėje. Monod lygtis turi tą pačią formą, kaip Michaelio – Menten, bet skiriasi tuo, kad ji yra empirinė, o pastarojo yra teorinė.

$$\mu = \mu_{max} \frac{s}{K_s + s} \quad (1.12)$$

$\mu$  – mikroorganizmų santykinis augimo greitis;  $\mu_{max}$  - mikroorganizmų maksimalus augimo greitis.  $S$  – substrato ribojimo koncentracija ir  $K_s$  – greičio konstanta, parenkami pagal konkretų mikroorganizmą.  $\mu_{max}$  ir  $K_s$  yra empiriniai koeficientai, kurie yra skirtingi kiekvienam procesui [9].

### *Alternatyvieji kinetiniai modeliai*

Yra nemažai modelių įvertinti mikroorganizmų augimą, tačiau dažniausiai naudojami šie modeliai:

Blackman:

$$\mu = \mu_{max} \min\left(1, \frac{s}{2K_s}\right) \quad (1.13)$$

Teissier:

$$\mu = \mu_{max} \left(1 - \exp\left(-\frac{s}{K_T}\right)\right) \quad (1.14)$$

Moser:

$$\mu = \mu_{max} \frac{s^n}{K_s + s^n} \quad (1.15)$$

Contois:

$$\mu = \mu_{max} \frac{s}{xK_s + s} \quad (1.16)$$



Powel:

$$\mu = \mu_{max} \frac{s - K_1 t(s)}{K_s + s - K_1 t(s)} \quad (1.17)$$

Logistic law:

$$\mu = \mu_{max} \left(1 - \frac{x}{K_x}\right) \quad (1.18)$$

Herbert:

$$\mu = (\mu_{max} + m) \frac{S}{K_s + S} - m \quad (1.19)$$

Shehata ir Marr:

$$\mu = \mu_1 \frac{S}{K_1 + S} + (\mu_{max} - \mu_1) \frac{S}{K_2 + S} \quad (1.20)$$

Haldane:

$$\mu = \frac{\mu_{max} S}{K_s + S + K_i S^2} \quad (1.21)$$

### *Biomasės išėigos koeficientas ir gyvybinių funkcijų palaikymas*

Biomasė auga dėl tiekiamo substrato. Galima padaryti išvadą, jog biomasės augimo greitis yra proporcingas substrato sunaudojimo greičiui. Priklausomybė išreiškiama formule:

$$\sigma = \frac{\mu}{Y_{xs}} \quad (1.22)$$

Biomasei augant dalis energijos sunaudojama mikroorganizmų gyvybinių funkcijų palaikymui, todėl pilnas santykinis substrato sunaudojimo greitis:

$$\sigma = \frac{\mu}{Y_{xs}} + m \quad (1.23)$$

$m$  – gyvybinių funkcijų palaikymo koeficientas, gali svyruoti priklausomai nuo proceso [8].

### *Inhibavimas*

Gana dažnai procesus veikia slopinimo mechanizmai (inhibavimas), tuomet yra mažinamas substrato sunaudojimas ir biomasės augimo greitis. Inhibavimas atsiranda dėl šalutinio produkto.

Dydžio įvertinimui naudojami įvairūs daugikliai: Ierusaliminskio ( $\frac{K_i}{K_i+1}$ ), Hinshelwoodo ( $1 -$

$\frac{I}{I_{max}}$ ), Levenspilio  $\left(1 - \frac{I}{K_i}\right)$ , Aibos  $\left(\exp\left(-\frac{I}{K_i}\right)\right)$ , Pirto ir Bellgardto, Tessier, Andrews, Webb (Edwards), Yano ir Kago.

Ierusalimskio gaunama tokia išraiška [8]:

$$\sigma = \sigma_{max} \frac{s}{K_s + s} \frac{K_i}{K_i + I} \quad (1.24)$$

Hinshelwoodo:

$$\sigma = \sigma \frac{s}{K_s + s} \left(1 - \frac{I}{I_{max}}\right) \quad (1.25)$$

Pirto:

$$\sigma = \sigma_{max} \frac{s}{s + \left(1 + \frac{s}{K_{si}}\right)K_s} \quad (1.26)$$

Bellgardto:

$$\sigma = \sigma_{max} \frac{s}{K_s + s + \frac{s^2}{K_{si}}} \quad (1.27)$$

Tessier:

$$\sigma = \sigma_{max} \left( \exp\left(-\frac{S}{K_{L,S}}\right) - \exp\left(-\frac{S}{K_S}\right) \right) \quad (1.28)$$

Andrews:

$$\sigma = \frac{\sigma_{max}}{\left(1 + \frac{K_s}{S}\right)\left(1 + \frac{K_{L,S}}{S}\right)} \quad (1.29)$$

Webb (Edwards):

$$\sigma = \sigma_{max} \frac{C_s \left(1 + \frac{C_s}{K_i}\right)}{K_s + C_s + \frac{C_s^2}{K_i}} \quad (1.30)$$

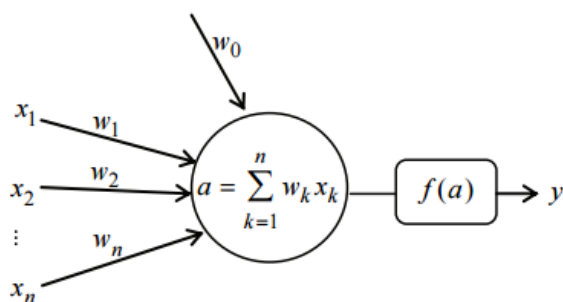
Yano ir Kago:

$$\sigma = \sigma_{max} \frac{C_s}{K_s + C_s + \frac{C_s^3}{K_i^2}} \quad (1.31)$$

### 1.2.3. Dirbtiniai neuroniniai tinklai (DNT)

Dirbtiniai neuroniniai tinklai (DNT) tai tarpusavyje sujungti neuronai, imituojantys nervinių ląstelių veiklą. DNT yra naudojami funkcijų aproksimavimui, prognozavimui, duomenų klasterizavimui, procesų optimizavimui ir modeliavimui. Modeliuojant hibridinį modelį dirbtiniais neuroniniais tinklais tikslas yra pagerinti proceso kokybę ir valdymą. DNT gali greitai ir tiksliai apdoroti nemažus kiekius informacijos, gali prisitaikyti prie naujų situacijų, toleruoti klaidas ir gali interpretuoti netikslią informaciją. Neuroninius tinklus apmokius pagal tam tikrus duomenis, tinklas geba prisitaikyti prie duomenų struktūros ir išmoksta atpažinti duomenis, kurių nebuvo apmokyme.

Dirbtiniai neuroniniai tinklai sukurti remiantis žmogaus smegenų veikimo principais. Kiekvienas neuronas gali išduoti informaciją. Biologiniai neuronai turi dendritus (įėjimai), somą (neuroso kūną), aksoną (neuroso išėjimas). Neuronai informaciją priima per dendritus ir gali bendrauti tarpusavyje su kitais neuronais per sinapses. Biologinis neuronas atiduoda informaciją per aksoną, kai gautas atitinkamas informacijos kiekis ir neuronas sužadinas [10].



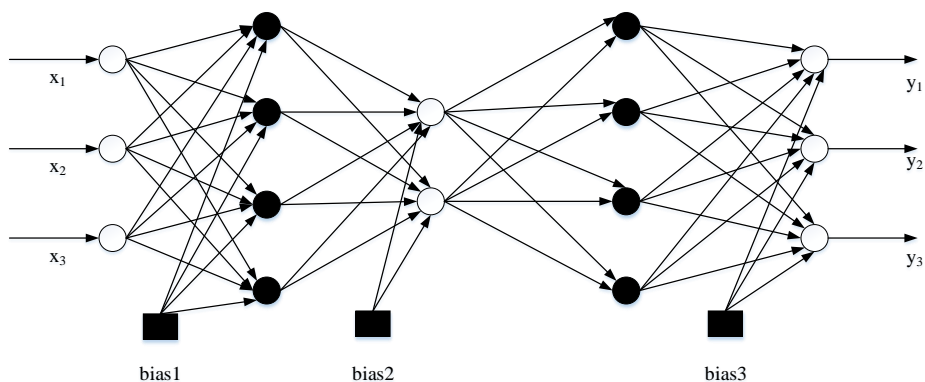
1.1 pav. Dirbtinis neuronas

Remiantis biologiniu neuronu sudaromas ir dirbtinis neuronas, kuris turi įėjimą, išėjimą ir aktyvavimo funkciją, kurioje vyksta skaičiavimai. Neuronas gali gauti keletą įėjimo reikšmių, todėl kiekvieną reikšmę atitinka svorinis koeficientas. Neuronas turi slenksčio ribą, kurią pasiekus neuronas sužadinas ir iš neuroso perduodama informacija. Neuroso išėjimo reikšmė apskaičiuojama iš perdavimo (aktyvacijos) funkcijos:

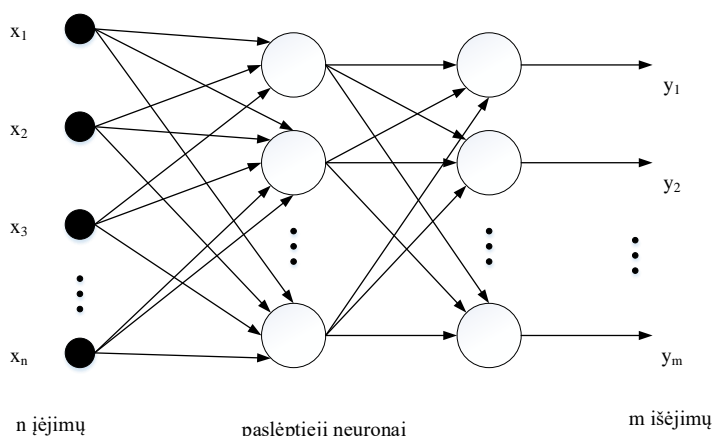
$$y = f(a) = f\left(\sum_{k=1}^n w_k x_k\right) \quad (1.32)$$

Pavieniai neuronai turi ribotas galimybes, todėl sudaromi tinklai. Neuroniniai tinklai dar turi ir paslėptąjį (tarpinį) sluoksnį. Paslėptasis sluoksnis jungia įėjimus su išėjimais ir atlieka vidinį vaidmenį tinkle. Įėjimuose perduodama informacija iš išorės į tinklus. Išėjimuose formuojamas valdymo signalas ar prognozė. Įėjimai ir išėjimai turi tiesinę aktyvavimo funkciją, o paslėpto

sluoksniu – netiesinę. Biotechnologiniuose procesuose dažniausiai naudojami: autoasociatyvus ar tiesioginio sklaidimo tinklai [8,11].



1.2 pav. Autoasociatyvus DNT



1.3 pav. Tiesioginio sklaidimo DNT

#### 1.2.4. Neraiškių aibių logika

Klasikinė logika apie teiginį leidžia daryti išvadas tik taip arba ne. Tačiau yra teiginiai, kurie iš dalies gali būti ir taip, ir ne. Pvz.: kai klausia žmonių grupės nustatyti tam tikrą spalvą, nuomonės gali nesutapti ir priskirti prie kelių spalvų. Atsirandant tokiems neraiškumams taikoma neraiškių aibių logika (angl. *fuzzy logic*), kuri apibūdinama neraiškių aibių taisyklėmis ir neraiškių aibių sistemomis. Būlio logikoje naudojamas 0 arba 1, neraiškioje logikoje reikšmės gali apimti net sakinius.

Neraiškių aibių logikoje naudojami lingvistiniai kintamieji, kurie padeda matematiškai apibrėžti teiginio teisingumą ar netikslumą. Fizikinio dydžio, pvz.: temperatūros, lingvistiniai kintamieji apibrėžiantys dydį yra: šalta, vėsu, šilta, karšta.

Neraiškioje aibėje elementas gali iš dalies priklausyti aibei, priklausomumas įvertinamas koeficientu. Elemento  $X$  priklausomumas aibei  $A$  aprašoma narystės funkcija [10]:

$$\mu_A: X \rightarrow [0,1], \quad (1.33)$$

Kiekviena neraiški aibė turi paviršių aprašančią funkciją, kuri parenkamai atitinkamai pagal norimą tikslumą. Dažniausiai tai būna trikampio formos paviršius, kuri aprašoma trimis taškais: pagrindo taškai ir viršūnė, tačiau galima aprašyti: trapecijos, varpo, S formos ir kitais paviršiais.

Neraiškių aibių sistemos pagrindas yra sprendimų, išvadų darymo mechanizmas, formuojamas neraiškėmis taisyklėmis. Neraiškios taisyklės formuojamos pagal loginę struktūrą: JEI..., TAI... Taisyklės apibrėžiamos išraiška: JEI Sąlyga  $\{x\}$ , TAI Išvada  $\{y\}$ .

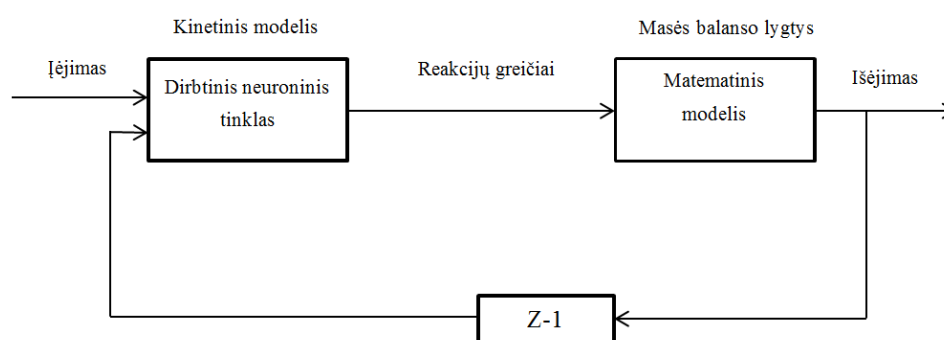
Neraiškioms aibėms, kaip ir Būlio logikoje naudojamos loginės operacijos: sankirta, neigimas ir sąjunga. Neraiškių aibių sistemą sudaro: įėjimo kintamųjų, neraiškumo įvedimo, išvadų darymo, neraiškumo eliminavimo bei išėjimo blokai. Neraiškumo eliminavimui dažniausiai taikoma svorio ir maksimumo vidurkio metodai. Neraiškių aibių taisyklėms aprašyti dažniausiai naudojamas Mamdani aibių modelis, dinaminiam modeliams taikomas Takagi – Sugeno aibių modelis [12].

Neraiškios aibės taikomos procesams modeliuoti, optimizuoti, gedimams nustatyti, duomenims atpažinti. Pritaikomos sritis: biotechnologija, medicina, bioinformatika, ekonomika, politika ir kt. Medicinoje pritaikoma diagnozuoti ligas, kai ligos turi kelių lygių neapibrėžtumus. Ligos kiekvienam pacientui gali pasireikšti skirtingai, priklausomai nuo paciento. Vienas simptomas gali sutapti su keliomis ligomis ir ligos simptomus galima aprašyti tiksliausiai žodžiais. Dažnai ligos statusas nėra visiškai aiškus ar pacientas perdėtai imituoja simptomus. Apie ligos diagnozę galima spręsti iš paciento istorijos ar kitų pacientų simptomų šios ligos. Neraiški aibių logika vaidina svarbų vaidmenį medicinoje padedant nustatyti ligas. Kaip neatsiejama sritis neraiškių aibių taikymo medicinoje yra bioinformatika. Bioinformatika panaudojama kaupti duomenis apie biologinius duomenis, pacientų statistiką, ligų požymius. Požymius apie DNR ląsteles kaupiami dideli duomenų kiekiai, todėl apdorojant didelius kiekius duomenų su netikslumais puikiai tinka neraiškių aibių logika [13]. Biologijoje pritaikoma apdoroti informacijai apie augalus, jų rūšys. Biocheminės medžiagų reakcijos yra gana sudėtingos, gaunama nemažai netikslios informacijos. Kartais, pvz.: biomasės charakteristikos nėra tiesiogiai išmatuojamos ir gaunama apytikrės informacijos, todėl pritaikoma *fuzzy* logika [14].

### 1.2.5. Tipinių hibridinių modelių struktūra

Hibridinių modelių tikslas aprašyti skirtingus procesus skirtingais būdais. Taikomi biotechnologinių procesų optimizavimui, stebėsenai ir valdymui. Hibridinis modelis parenkamas pagal turimus duomenis apie procesą, ne visada turima daug ir tikslios informacijos. Dažnai naudojamos sudėtingos reakcijos, kurioms aprašyti nepakanka mechanistinio modelio, tuomet ieškoma alternatyvių metodų aprašyti procesą.

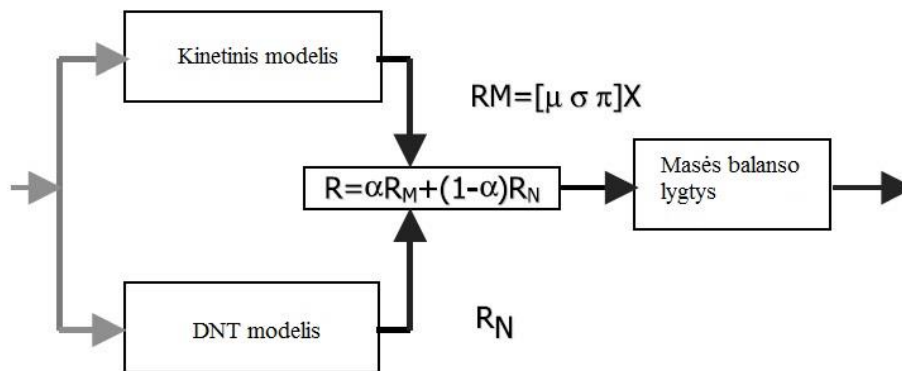
Pateikiama 1.4 paveiksle hibridinis modelis, kuriame naudojami DNT ir matematinis modelis. Matematiniam modelyje aprašomos proceso masės balanso lygtys, o DNT modeliuoja biologinių mikroorganizmų kinetika. Tačiau toks modelis tinkamas naudoti tik erdvėje, kurioje buvo apmokomas, t.y. turi blogas ekstrapoliavimo savybes [15].



1.4 pav. Hibridinio modelis.

Iš bandymų matoma, jog DNT puikiai tinka aprašyti netiesines funkcijas. Tačiau iškyla problemos su tinklo apmokymu, dažnai nepakanka duomenų tiksliai apmokyti. Tinklui aprašyti reikalingas nemažas kiekis svorinių koeficientų, todėl ieškoma alternatyvų, kaip supaprastinti apmokymą. Dažnai DNT apmokomi jautrumo lygčių metodu.

Dar vienas iš sprendimo būdų supaprastinti DNT apmokymą yra taikyti klasikinį Monod modelį pirminiam modelio aprašymui (1.5 pav.). Atitinkamai kinetiniai parametrai pritaikomi pagal proceso turimus duomenis. Tuomet DNT gali būti apmokomi naudojantis kinetiniais parametrais. Taip sutrumpinama paieška, nereikia pradėti paiešką nuo atsitiktinių svorinių koeficientų [15].



1.5 pav. Hibridinis modelis sudarytas iš kinetinio  $R_M$ , DNT  $R_N$  ir masės balanso lygčių modelių.

### 1.2.6. Modelių struktūros ir parametų identifikavimas

Sukūrus modelį svarbu įsitikinti ar jis tinkamos struktūros. Modelį reikia keisti, kai yra dideli nukrypimai tarp eksperimentinių duomenų ir modeliuotų rezultatų. Dažniausiai modelio tikslumo įvertinimui pasirenkamos atitinkamos paklaidos, kurios skaičiuoja modelio kokybės kriterijų. Modelis turi modeliuoti su kuo mažesnėmis paklaidomis, t.y. minimizuoti kokybės kriterijų.

Taip pat svarbu įvertinti kiekvieno parametro įtaką procesui, t.y. atlikti parametų jautrumo analizę. Parametrai siejami su galutine proceso optimizavimo funkcija, kurią siekiama minimizuoti (maksimizuoti). Svarbu parinkti parametų reikšmingumą taip, kad tikslo funkcija pasiektų ekstremumo vertę. Parametrai turintys didžiausią įtaką procesui, turi turėti didžiausią svorį (koeficientą), o parametrai turintys mažą reikšmingumą procesui atitinkamai turi turėti mažesnę svorį [16].

DNT modeliams įvertinti atliekami mokymo ir testavimo bandymai. Modeliui priskiriami duomenys apmokymui ir priskiriama kita duomenų grupė, kuri skirta testavimo rezultatams gauti. Palyginus modelių struktūrų mokymo ir testavimo bandymus, galima nuspręsti, kuris modelis labiausiai tinkamas procesui ir, kuris tiksliausiai prognozuoja eksperimentus [17].

#### *Identifikavimas naudojant Levenberg – Marquardt algoritmą*

*Levenberg – Marquardt* algoritmas sukurtas dviejų mokslininkų Kenneth Levenberg ir Donald Marquardt. Šis algoritmas naudojamas netiesinėms funkcijoms spręsti, taikomas dirbtinių neuroninių tinklų apmokymui. Algoritmas taip pat naudojamas optimizavimo uždaviniams spręsti, rasti minimumus, tačiau šiuo metodu galima rasti tik lokalųjį minimumą. Šis metodas yra derinys greičiausio nusileidimo metodo ir *Gauso – Niutono* algoritmo.

*Levenberg – Marquardt* algoritmas remiasi gana sudėtingais skaičiavimais. Dažniausiai apmokymo rezultatams įvertinti apskaičiuojama paklaidos kvadratų suma:

$$E(x, w) = \frac{1}{2} \sum_{p=1}^p \sum_{m=1}^M e_{p,m}^2, \quad (1.34)$$

kur,  $x$  – įėjimo vektorius,  $w$  – svorių vektorius,  $e_{p,m}$  – apmokymo paklaidos.

Kadangi *Levenberg – Marquardt* algoritme naudojami kelių algoritmų deriniai, tai žemiau aprašomas greičiausio nusileidimo metodas. Greičiausio nusileidimas metodas yra pirmos eilės algoritmas. Apskaičiuojama pirmos eilės išvestinė iš tikslo funkcijos, kuri yra šio metodo gradientas:

$$g = \frac{\partial E(x, w)}{\partial w} = \left[ \frac{\partial E}{\partial w_1} \quad \frac{\partial E}{\partial w_2} \quad \dots \quad \frac{\partial E}{\partial w_N} \right]^T, \quad (1.35)$$

Toliau Niutono metodu apskaičiuojami visi gradientai:

$$\begin{cases} g_1 = F_1(w_1, w_2, \dots, w_N) \\ g_2 = F_2(w_1, w_2, \dots, w_N) \\ \dots \\ g_N = F_N(w_1, w_2, \dots, w_N) \end{cases} \quad (1.36)$$

kur  $F_1, F_2, \dots, F_N$  yra netiesinės priklausomybės funkcijos svorinių koeficientų ir gradientų. Apskaičiavus pagal Teiloro eilutę ir atlikus kitus skaičiavimus gaunama Hesiano matrica. Apskaičiuojama antros eilės išvestinė iš kiekvienos funkcijos nario:

$$H = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_1 \partial w_N} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \frac{\partial^2 E}{\partial w_2^2} & \dots & \frac{\partial^2 E}{\partial w_2 \partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 E}{\partial w_N \partial w_1} & \frac{\partial^2 E}{\partial w_N \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_N^2} \end{bmatrix} \quad (1.37)$$

pagal 1.35 ir 1.37 formules gauname:

$$\Delta w = -H^{-1}g, \quad (1.38)$$

tuomet svoriniai koeficientai kiekviename iteracijos žingsnyje apskaičiuojami pagal formulę:

$$w_{k+1} = w_k - H_k^{-1}g_k, \quad (1.39)$$

Supaprastinti skaičiavimus naudojamas Gauso – Niutono algoritmas, kurio metu skaičiuojama Jakobiano matrica:



$$J = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_1} & \frac{\partial e_{1,1}}{\partial w_2} & \dots & \frac{\partial e_{1,1}}{\partial w_N} \\ \frac{\partial e_{1,2}}{\partial w_1} & \frac{\partial e_{1,2}}{\partial w_2} & \dots & \frac{\partial e_{1,2}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{p,m}}{\partial w_1} & \frac{\partial e_{p,m}}{\partial w_2} & \dots & \frac{\partial e_{p,m}}{\partial w_N} \end{bmatrix} \quad (1.40)$$

tuomet Hesiano matrica galima būti apskaičiuojama pagal Jakobiano matricą:

$$H = J^T J, \quad (1.41)$$

Tada svorinis koeficientas apskaičiuojamas kiekvieno iteracijoje:

$$w_{k+1} = w_k - (J^T J)^{-1} J_k e_k, \quad (1.42)$$

Akivaizdu, kad Gauso – Niutono metodas paprastesnis nei Niutono, nereikalauja skaičiuoti 2 eilės išvestinių.

Siekiant tiksliau apskaičiuoti naudojama Hesiano matricos išraiška:

$$H = J^T J + \mu I, \quad (1.43)$$

kur  $\mu$  – derinimo koeficientas,  $I$  – identifikavimo matrica.

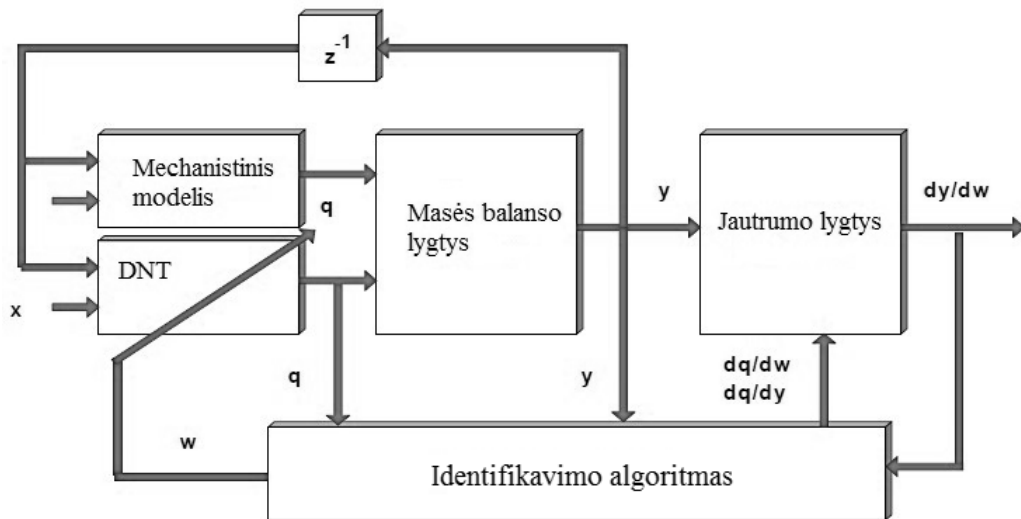
Remiantis 1.42 ir 1.43 formulėmis svorinis koeficientas apskaičiuojamas kiekvieno iteracijoje:

$$w_{k+1} = w_k - (J^T J + \mu I)^{-1} J_k e_k, \quad (1.44)$$

Naudojant *Levenberg – Marquardt* algoritmą, paieška vykdoma varijuojant greičiausio nusileidimo ir *Gauso – Niutono* algoritmus. Kai derinimo koeficientas  $\mu$  yra artimas nuliui, naudojamas *Gauso – Niutono* metodas, kai derinimo koeficientas yra didelis, naudojamas greičiausio nusileidimo metodas [18].

*Identifikavimas naudojant jautrumo lygčių metodą*

Matematinis modelis pateiktas 7 paveiksle, kuris sudarytas pagal netiesines diferencialines lygtis, tam tikras mechanistines išraiškas bei DNT.



1.6 pav. Identifikavimas DNT parametrų naudojant jautrumo lygtis.

Jautrumo lygčių metodo apmokymas vyksta tokia tvarka:

1. Nustatomos masės balanso lygtys;
2. Pasirenkami pirminiai DNT struktūros svoriai;
3. Suformuojamos jautrumo lygtys;
4. Integruojamos jautrumo lygtys ir nustatomos išvestinės  $\frac{\partial y}{\partial w_n}$ ;

5. Pasinaudojant  $\frac{\partial y}{\partial w_n}$  apmokomas DNT gradientiniuose algoritmuose. Apmokymai atliekami tol kol paklaida mažėja.

Diferencialinė masės balanso lygtis gali būti aprašoma tokia forma:

$$\frac{dy}{dt} = f(y(t), x(t), q(y(t), w)), \quad (1.45)$$

kur  $f$  – netiesinė sistemos funkcija, sistemos įėjimas  $x(t)$ , išėjimas  $y(t)$ .  $q(y(t), w)$  - biocheminės reakcijos įverčiai. Norint apmokyti neuroninį tinklą hibridiniame modelyje, būtina naudoti išmatuotus įėjimo/išėjimo dydžius. Mokymas DNT reikalauja, kad  $w$  svoris būtų nustatytas, taip kad suma, kurią sudaro nuokrypių kvadratų suma  $J$ , būtų minimizuota:

$$J = \frac{1}{2} \sum_i (y_i - y_{i,exp})^2 \rightarrow \min. \quad (1.46)$$

Minimizuoti  $J$  naudojamas gradiento metodas, pritaikant DNT svorius:

$$w_{n+1} = w_n - g \frac{\partial J}{\partial w_n}. \quad (1.47)$$

Tuomet formuojama išvestinė J atžvilgiu w:

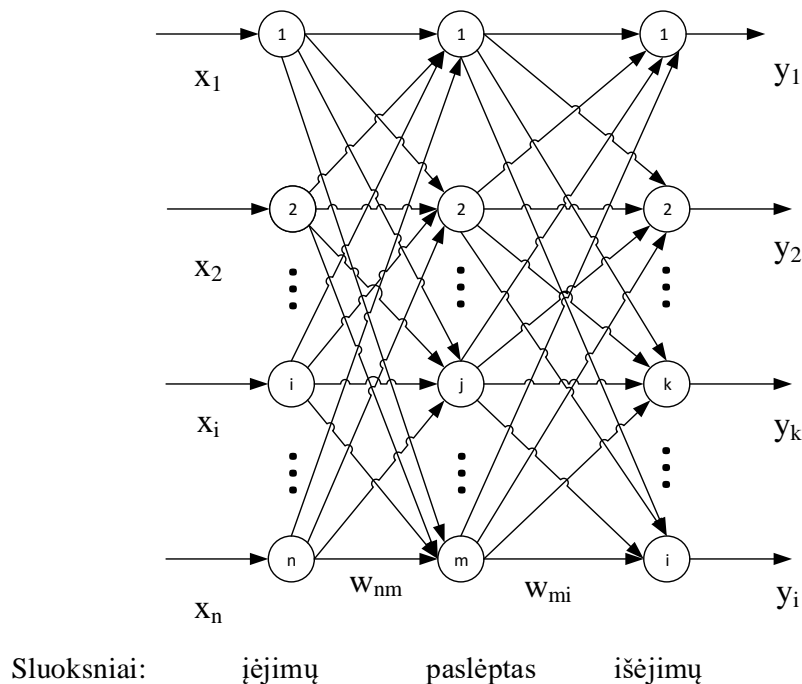
$$\frac{\partial J}{\partial w_n} = \sum_i (y_i - y_{i,exp}) \frac{\partial y_i}{\partial w_n}. \quad (1.48)$$

DNT išėjimas  $q(y(t), w)$  ir eksperimentų rezultatai  $y_{i,exp}$  yra skirtingų matavimų dimensijų, todėl negalimas standartinis apmokymas. Pritaikomos jautrumo lygtys. Rastos išvestinės  $\frac{\partial y_i}{\partial w_n}$  reikia diferencijuoti  $w_n$  atžvilgiu [19]:

$$\frac{d}{dt} \frac{\partial y}{\partial w_n} = \frac{\partial f}{\partial y} \frac{\partial y}{\partial w_n} + \frac{\partial f}{\partial w_n}. \quad (1.49)$$

### Identifikavimas klaidos atgalinio skleidimo metodu

Hibridinių modelių su dirbtiniais neuroniniais tinklais apmokymui gali būti naudojamas klaidos atgalinio skleidimo metodu (*error back - propagation*). Kai įėjimo signalas pasiekia tinklą, aktyvuoja paslėpto sluoksnio neuronus, tuomet paslėpto sluoksnio neuronai aktyvuoja išėjimo neuronus ir suformuoja išėjimo signalą. Klaidos atgalinio skleidimo metodu pirmiausia randami  $w_{mi}$  – svoriniai koeficientai, kurie aktyvuoja išėjimo sluoksnio neuronus. Toliau surandami  $w_{nm}$  – svoriniai koeficientai, kurie aktyvuoja paslėpto sluoksnio neuronus.



1.7 pav. DNT tinklas

Apmokymo procedūra pradama nuo kokybės kriterijaus parinkimo:

$$E_i = \frac{1}{2}(y_{d,i} - y_i)^2, \quad (1.50)$$

tuomet nustatoma  $w_{mi}$  svorių keitimo funkcija  $p$  - joje iteracijoje:

$$w_{mi}(p+1) = w_{mi}(p) - \alpha \frac{\partial E_i}{\partial w_{mi}(p)} = w_{mi}(p) - \alpha(y_{d,i} - y_i) \frac{\partial y_i}{\partial w_{mi}(p)}. \quad (1.51)$$

Tarkime, kad neurono funkcija yra sigmoidė, tada  $\frac{\partial y_i}{\partial w_{mi}(p)}$  išvestinė:

$$\frac{\partial y_i}{\partial w_{mi}(p)} = y_i(1 - y_i)yh_m. \quad (1.52)$$

Apjungus abi išraiškas, gaunama lygtis DNT svoriams  $w_{mi}$ :

$$w_{mi}(p+1) = w_{mi}(p) + \alpha(y_{d,i} - y_i)y_k(1 - y_i)yh_m. \quad (1.53)$$

Toliau identišku metodu gaunama lygtis  $w_{nm}$  svoriams:

$$w_{nm}(p+1) = w_{nm}(p) + \alpha x_n \delta_m. \quad (1.54)$$

$\delta$  – paslėpto sluoksnio paklaidos gradientas. Apskaičiuojamas:

$$\delta_m = yh_m(1 - y_m) \sum_{i=1}^l y_i(1 - y_i)(y_{d,i} - y_i)w_{mi}(p). \quad (1.55)$$

Šis metodas nereikalauja daug laiko skaičiavimams, palyginti su kitais metodais nėra itin sudėtingas [20].

#### *Atsitiktinės paieškos metodai. Chemotaxis algoritmas*

Tobulėjant kompiuteriniams sistemoms, vis dažniau naudojami atsitiktinės paieškos metodai. Taip pat atsitiktinės paieškos metodai efektyvesni lyginant su gradientiniais ar tiesioginės paieškos metodais, kurie reikalauja daug laiko analitinėms išraiškoms nustatyti ar papildomoms programoms.

Atsitiktinės paieškos metodas apibrėžiamas funkcija, kuri gali būti splainas, dirbtinis neuroninis tinklas, tam tikra lygtis, lygčių sistema ar sudarytas rinkinys (hibridinis modelis). Bendra forma funkcija apibrėžiama:

$$c = f(t, p), \quad (1.56)$$

kur,  $p$  – paieškos parametrų vektorius,  $c$  – būsenos kintamieji,  $f(t, p)$  – funkcinė priklausomybė ir naudojami proceso eksperimentiniai duomenys  $[t_e c_e]$ .

Paieškos procedūra pradedama sudarant pradinį duomenų vektorių  $p_0$ , toliau naudojant funkciją  $f(t, p)$  ir įsistačius  $t_e$ , apskaičiuojama funkcijos  $c$  vertė. Paieškos procedūros tikslas rasti optimalias  $p$  vertes, kurios minimizuotų tikslo funkciją. Naudojant tikslo funkciją įvertinamas

vektoriaus  $p$  tinkamumas. Tikslo funkcija apibrėžiama, kaip eksperimentinių ir modeliavimo rezultatų nuokrypis, tai gali būti vidutinis kvadratinis nuokrypis ar vidutinė absoliutinė paklaida, pvz.:

$$J = \frac{1}{n} * \sum_{i=1}^n |c(t, p) - c(t)|. \quad (1.57)$$

Paieška tęsiama iki išnaudojamas iteracijų skaičius arba gaunamas tikslo funkcijos pagerėjimas.

Vienas iš paprasčiausių atsitiktinės paieškos metodų Chemotaxis. Tai labai paprastas algoritmas, kurį sudaro vos keletas programos eilučių. Tačiau yra tikimybė, kad naudojant šį metodą galima patekti į lokalinį ekstremumą. Šis metodas panašus į mikroorganizmų elgseną ieškant pamaitinimo substrato. Mikroorganizmai naudoja substratą, juda nedidelėje aplinkoje, suradus substratą jį suvartoja ir juda ta pačia kryptimi, ieškodami kito substrato. Chemotaxis paiešką galima aprašyti tokiais žingsniais:

1. Pasirenkama tikslo funkcija  $f$ , įvertinanti eksperimentinių ir modeliavimo rezultatų nuokrypį.

2. Pasirenkamas pradinis parametrų vektorius  $x_0$ , jei vertės nėra žinomos, tai gali būti atsitiktiniai dydžiai.

3. Apskaičiuojama tikslo funkcija su pradiniu parametrų vektoriumi  $f(x_0)$ .

4. Pasirenkama parametrų mutacija. Prie pradinio parametrų vektoriaus  $x_0$  pridedami atsitiktiniai pokyčiai. Pokytis dažniausiai naudojamas Gauso kintamasis su vienetine dispersija ir nuline matematine viltimi.

$$x_n = x_{0n} + \Delta x_n * rand_n. \quad (1.58)$$

5. Apskaičiuojama tikslo funkcija su gautu vektoriumi  $f(x_n)$ .

6. Lyginama tikslo funkcija: jei  $f(x_n) < f(x_0)$ , paieška tęsiama tokiu pačiu būdu. Jei  $f(x_n) > f(x_0)$ , grįžtama prie buvusių verčių  $x_0$  ir paieška vykdoma vėl nuo 4 žingsnio.

7. Paieška vykdoma kol išnaudojamas iteracijų skaičius arba pasiekama atitinkama tikslo funkcijos vertė [21].

### *Identifikavimas naudojant evoliucinį algoritmą*

Biotechnologiniams uždaviniams spręsti gana dažnai naudojamas evoliucinis algoritmas, kuris tinka tiek modelio parametrų identifikavimui, tiek proceso optimizavimui. Evoliucinis algoritmas nereikalauja daug žinių apie procesą, kadangi atliekama atsitiktinė paieška. Metodas taikomas, kai yra didelė tikimybė, jog bus rastas lokalusis ekstremumas. Algoritmas gana efektyvus, tačiau paieška užtrunka ilgai.

Evoliucinis algoritmas yra paremtas biologinių modelių, natūralios gamtos atranka. Gamtoje evoliucija padeda prisitaikyti prie aplinkos. Gamtoje yra aibė veiksni: klimatas, apsirūpinimas maistu, plėšrūnų pavojus. Šie veiksniai daro įtaką organizmų išlikimui, vieną rūšį keičia kita rūšis. Atsitiktinai atsiranda mutacijos, kurios būna vieniems naudingos, tačiau kitiems gali būti ir kenksmingos. Šiais gamtoje vykstančiais veiksniais paremti ir evoliucinio algoritmo skaičiavimai.

Visų pirma, paieškai sukuriama pradinė populiacija, tai atsitiktinis arba tam tikras duomenų rinkinys. Įvertinamas kiekvienas narys. Geriausiai populiacijos nariai (tėvai) panaudojami naujų narių (vaikų) kūrimui. Vykdomas pakartotinis narių įvertinimas, kurio metu geriausi nariai panaudojami sekančiai kūrimo operacijai, o silpniausi nariai pašalinami. Kadangi tik stipriausi nariai turi išlikimo galimybę, populiacijos dydis ribotas, silpnesnius narius išstumia stipresni. Naujų narių informacija ir savybės sukuriamos kryžminimo ir mutacijos operacijomis.

Evoliucinį algoritmą taikant optimizavimo ar identifikavimo uždaviniams, sukuriama tikslo (kokybės) funkcija  $f(x,y) \rightarrow \min(\max)$ . Sprendžiant uždavinį reikia rasti funkcijos minimumą ar maksimumą. Dažnai tikslo funkcija yra absoliutinė paklaida ar vidutinis kvadratinis nuokrypis tarp eksperimentinių ir modeliavimo rezultatų. Paieškai sukuriami y populiacijos nariai, kurie evoliucionuoja pagal mutacijos ar kryžminimo dėsnius ir yra įvertinami pagal kokybės funkciją. Geresni nariai pakeičia blogesnius, o su geriausiais nariais vykdoma sekanti paieškos iteracija. Paieška tęsiama kol išnaudojamas iteracijų skaičius arba pasiekama atitinkama kokybės funkcijos vertė [22].

## 2.HIBRIDINIO MODELIO IDENTIFIKAVIMAS

### Tiriamosios dalies sutartinių ženklų ir matavimo vienetų sąrašas

Sutartinis ženklas	Matavimo vienetai	Apibūdinimas
$F_1$	kg/h	Masės srautas
$F_2$	kg/h	Mėginių ėmimo dažnis
$k_i$	g/kg	Slopavimo konstanta substrato suvartojimui
$k_{i\mu}$	U/g	Slopavimo konstanta baltymų modelio
$k_m$	U/g	Proporcijos koeficientas baltymų modelio
$k_s$	g/kg	Monod konstanta substrato suvartojimui
$k_\mu$	1/h	Monod konstanta baltymų modelio
$m$	g/g/h	Biomosės gyvybinių funkcijų palaikymo koeficientas
$p_{\max}$	U/g biomasės	Maksimalus rekombinuotų baltymų aktyvumas
$p_x$	U/g biomasės	Rekombinuotų baltymų aktyvumas
$q_{px}$	U/g/h	Rekombinuotų baltymų sintezės greitis
$q_s$	g/g/h	Substrato suvartojimo greitis
$s$	g/kg	Substrato koncentracija
$s_f$	g/kg	Substrato koncentracija pamaitinimo tirpale
$t$	h	Proceso esamas laikas
$t_f$	h	Proceso trukmė
$t_{\text{ind}}$	h	Indukcijos trukmė
$T$	°C	Vidutinė temperatūra
$T_{\text{ref}}$	°C	Etaloninė temperatūra
$u$	kg/h	Substrato pamaitinimo koeficientas
$w$	kg	Kultūros masės
$x$	g/kg	Biomosės koncentracija
$Y_{xs}$	g/g	Biomosės / substrato išeiga
$\alpha$	1/°C	Temperatūros konstanta
$\mu$	1/h	Biomosės augimo greitis
$n$	-	Eksperimentinių taškų skaičius
$x_{\text{eksp}}$	-	Eksperimentiniai duomenys
$x_{\text{model}}$	-	Modeliavimo rezultatai

Tyrimams pasirinktas periodinis su pamaitinimu kultivavimo procesas (rekombinuotų baltymų gamybos). Bioproceso tikslas - maksimizuoti tikslinį produktą, rekombinuotų baltymų koncentraciją. Baltymų susidarymo greičiui didžiausią įtaką daro biomasės augimo greitis. Mažesnis greitis lemia didesnę rekombinuotų baltymų koncentraciją.

Darbe pateikiami 4 proceso eksperimentai. Pateikiamos proceso masės balanso lygtys (biomasės ir substrato koncentracijos, rekombinuotų baltymų aktyvumo bei kultūros masės), biomasės augimo greitis, substrato suvartojimo greitis, baltymų augimo greitis, maksimalus baltymų aktyvumas, pamaitinimo profiliai bei kiti proceso parametrai.

Tyrimo tikslas - naudojantis pateiktu mechanistiniu modeliu, sukurti dinaminį proceso modelį. Dinaminiu proceso modeliu aproksimuojami mechanistinio modelio netikslumai. Dinaminiam modelyje naudojami dirbtiniai neuroniniai tinklai (DNT). DNT apmokomi naudojant *Levenberg – Marquardt* algoritimą, gauti rezultatai patikslinami jautrumo lygčių metodu. Pagrindinis dėmesys skiriamas rekombinuotų baltymų sintezės greičio DNT modeliui, kurio pagrindiniai įėjimai yra santykinės biomasės augimo greitis ir rekombinuotų baltymų aktyvumas.

Modelio taškų prognozavimo tikslumui įvertinti skaičiuojama vidutinė absoliutinė paklaida:

$$MAE = \frac{1}{n} * \sum_{i=1}^n |x_{eksp.} - x_{model.}|. \quad (2.1)$$

Įvertinama kiekvieno eksperimento taško nuokrypa tarp modeliuotų ir eksperimentinių duomenų.

Modelio prognozavimo tikslumui įvertinti skaičiuojama vidutinė procentinė absoliutinė paklaida:

$$MAPE = \frac{100\%}{n} * \sum_{i=1}^n \left| \frac{x_{eksp.} - x_{model.}}{x_{eksp.}} \right|. \quad (2.2)$$

Prognozavimas laikomas labai tikslus, kai  $MAPE < 10$ , tikslus, kai  $10 < MAPE < 20$ . Paklaidos skaičiuojamos tik rekombinuotų baltymų aktyvumo.



## 2.1. Mechanistinis modelis

Mechanistinis modelis modeliuojamas *Matlab / Simulink* programiniu paketu. Naudojantis *Simulink*, *DEE* (diferencinių lygčių) bloku, suvedamos masės balanso lygtys.

Biomasės koncentracijos:

$$\frac{dx}{dt} = \mu(s)x - (u + F_1) \frac{x}{w}, \quad (2.3)$$

Substrato koncentracijos:

$$\frac{ds}{dt} = -q_s(s)x + u \frac{s_f}{w} - (u + F_1) \frac{s}{w}, \quad (2.4)$$

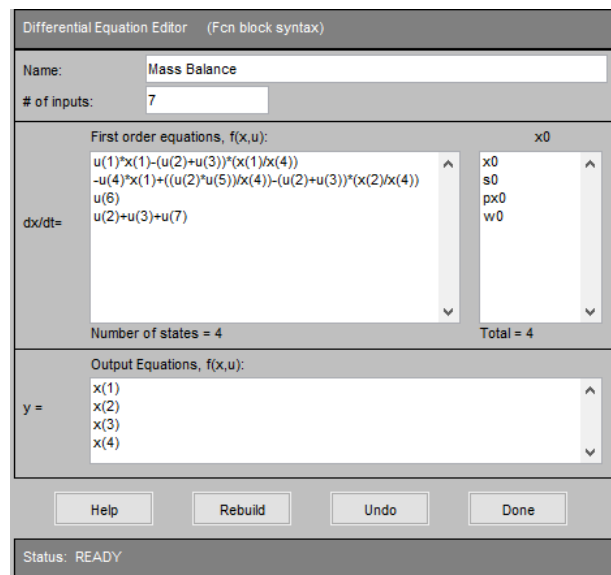
Rekombinuotų baltymų aktyvumo:

$$\frac{dp_x}{dt} = q_{px}(\mu, p_x), \quad (2.5)$$

Bakterijų kultūros masės:

$$\frac{dw}{dt} = u + F_1 + F_2, \quad (2.6)$$

Masės balanso lygtys *DEE* bloke pateikiamos 2.1 paveiksle.



2.1 pav. *DEE* blokas.

Modelyje naudojamas santykinis biomasės augimo greitis  $\mu$ :

$$\mu(s, T) = \mu_{max} \frac{s}{k_s + s} * \frac{k_i}{k_i + s} \exp(\alpha(T + T_{ref})), \quad (2.7)$$

Substrato suvartojimo greitis  $q_s$ :

$$q_s = \frac{1}{Y_{xs}} \mu(s) + m, \quad (2.8)$$

Rekombinuotų baltymų augimo greitis  $q_{px}$ :

$$q_{px}(\mu, p_x) = \frac{1}{T_{px}} (p_{max}(\mu) - p_x), \quad (2.9)$$

Maksimalus rekombinuotų baltymų aktyvumas  $p_{max}$ , [23]:

$$p_{max}(\mu) = \frac{\mu k_m}{k_\mu + \mu + \mu^2/k_{i\mu}}, \quad (2.10)$$

Masės srautas ( $F_1$ ):

$$F_1 = 0,001 * \mu * w * x + Fe, \\ \text{jei } t < t_{ind}, \text{ tai } Fe = -0,002, \quad \text{jei } t > t_{ind}, \text{ tai } Fe = -0,005 \quad (2.11)$$

Mėginių ėmimo dažnis ( $F_2$ ):

$$\text{jei eksper. 3, tai } F_2 = -0,06, \text{ jei eksper. 4, tai } F_2 = -0,045, \\ \text{jei eksper. 1 ar 2 ir } t < 4, \text{ tai } F_2 = 0, \\ \text{jei } t > 4, \text{ tai } F_2 = -0,015, \quad (2.12)$$

Sukuriamas likusioji *Simulink* modelio dalis. Sukuriamas *Matlab* .m programinis kodas, kuriame įvedamos biomasės ir substrato koncentracijos, rekombinuotų baltymų aktyvumo bei bakterijų kultūros masės pradinės reikšmės. Užkraunami eksperimentiniai ir pamaitinimo profilio duomenys. Sukuriamas grafinis palyginimas tarp modeliuotų ir eksperimentinių duomenų. Sukuriamas ciklas, eksperimento pasirinkimui (darbe naudojami 4 eksperimentai). Įvedamos formulės vidutinei absoliutinei ir vidutinei procentinei absoliutinei paklaidai apskaičiuoti.

Programinis kodas pateikiamas priede nr.1, *Simulink* blokų kodai pateikiami priede nr. 2, *Simulink* modelis pateikiamas priede nr. 3. Darbe naudojamos pradinės sąlygos bei kitos parametrų reikšmės pateikiamos žemiau:

2.1 lentelė. Pradinės reikšmės

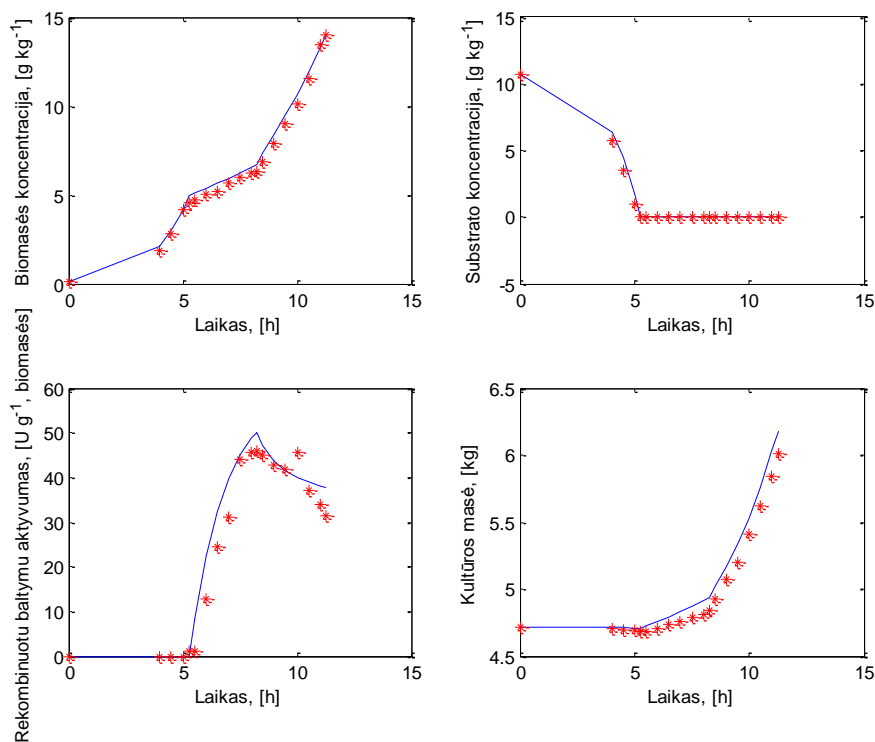
Dydis	Eksperimento numeris			
	1	2	3	4
x(0)	0,143	0,119	0,116	0,116
s(0)	10,7	10,1	9,6	9,5
w(0)	4,7	4,9	4,95	5,05
p <sub>x</sub> (0)	0	0	0	0
t <sub>ind</sub> (0)	5,27	5,47	6,75	8,0
Sf	100	100	156	151

2.2 lentelė. Modelio parametrų vertės

Dydis	Reikšmė
$k_i$	93,8
$k_{i\mu}$	0,0174
$k_m$	751
$k_\mu$	0,61
$k_s$	0,00333
$m$	0,0242
$T_{px}$	1,495
$Y_{xs}$	0,46
$\alpha$	0,0495
$\mu_{max}$	0,737

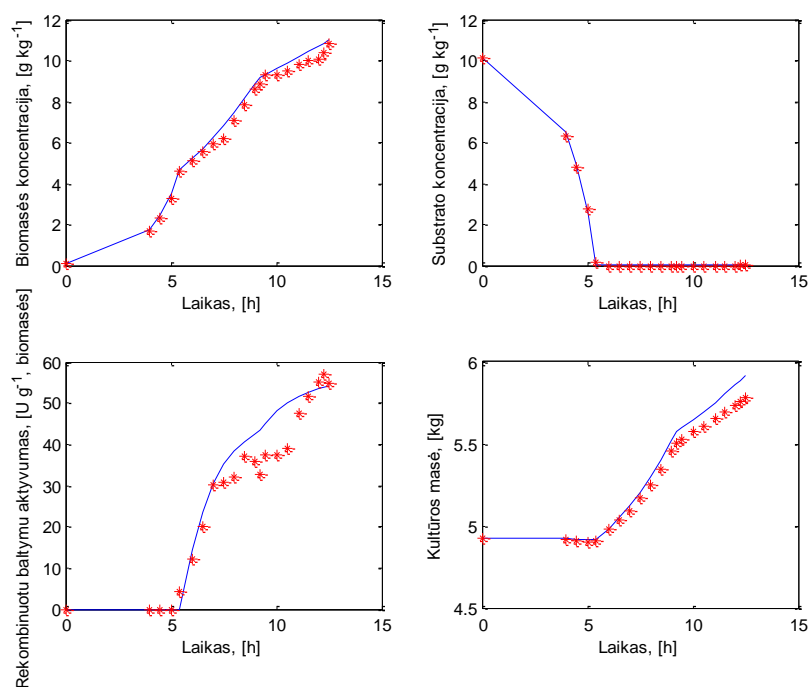
### 2.1.1. Mechanistinio modelio modeliavimo rezultatai

1 eksperimentas:



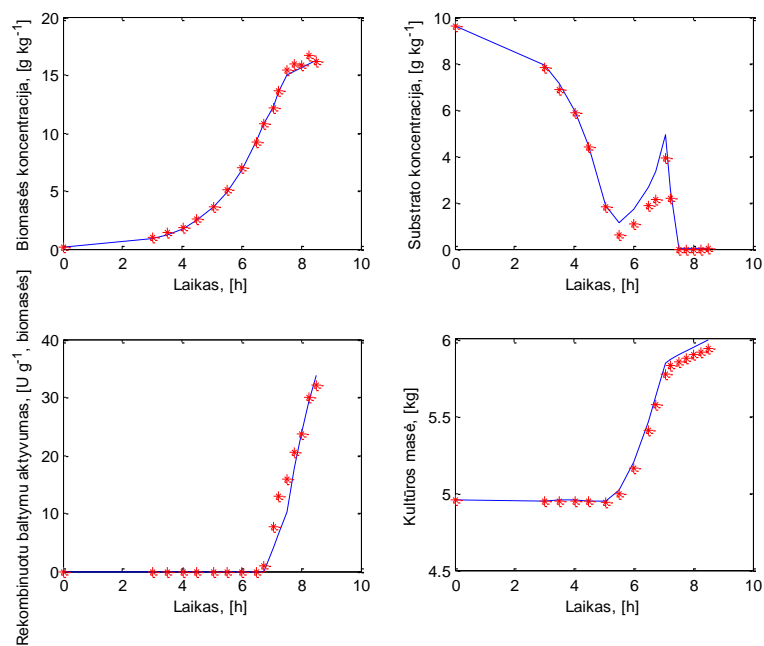
2.2 pav. Rekombinuotų baltymų aktyvumo eksperimentinių duomenų ir modeliavimo rezultatų palyginimas, kai modeliuojama mechanistiniu modeliu. Ištininė linija – mechanistinio modelio rezultatai, žvaigždės formos taškai – eksperimentiniai duomenys.

## 2 eksperimentas:



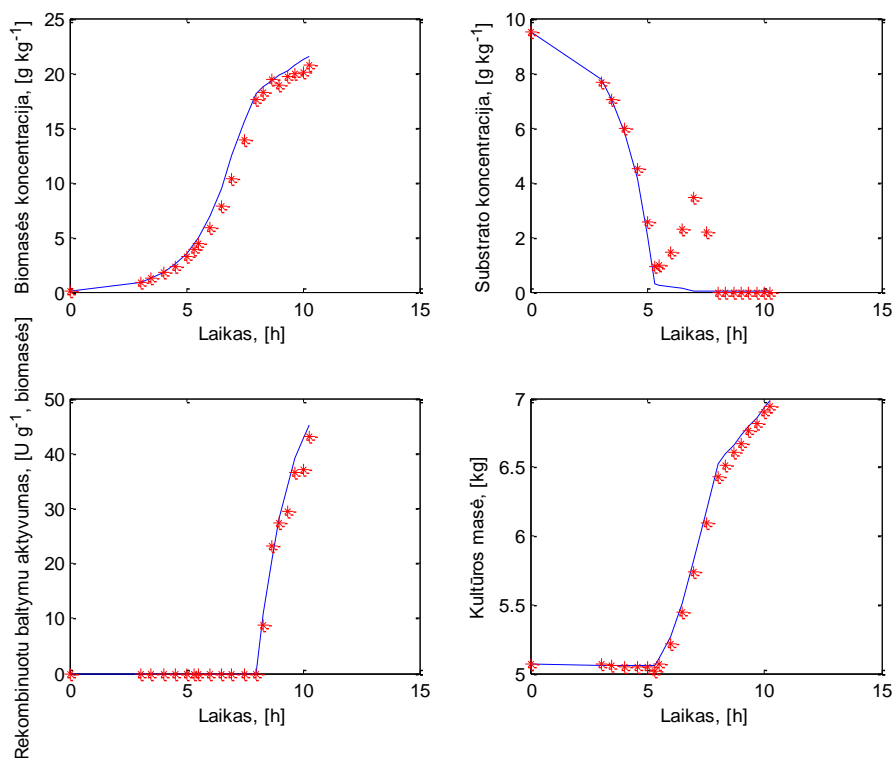
2.3 pav. Rekombinuotų baltymų aktyvumo eksperimentinių duomenų ir modeliavimo rezultatų palyginimas, kai modeliuojama mechanistiniu modeliu. Ištinė linija – mechanistinio modelio rezultatai, žvaigždės formos taškai – eksperimentiniai duomenys.

## 3 eksperimentas:



2.4 pav. Rekombinuotų baltymų aktyvumo eksperimentinių duomenų ir modeliavimo rezultatų palyginimas, kai modeliuojama mechanistiniu modeliu. Ištinė linija – mechanistinio modelio rezultatai, žvaigždės formos taškai – eksperimentiniai duomenys.

4 eksperimentas:



2.5 pav. Rekombinuotų baltymų aktyvumo eksperimentinių duomenų ir modeliavimo rezultatų palyginimas, kai modeliuojama mechanistiniu modeliu. Ištinė linija – mechanistinio modelio rezultatai, žvaigždės formos taškai – eksperimentiniai duomenys.

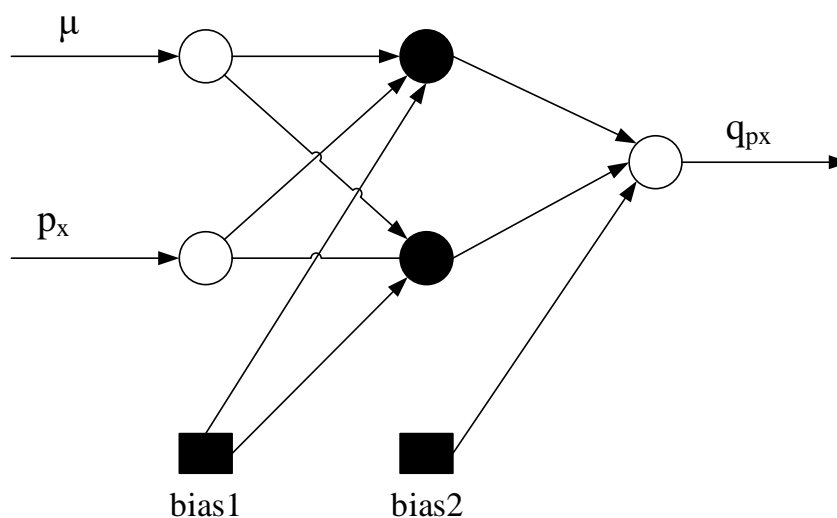
Mechanistinio modelio 1 – 4 eksperimento vidutinė absoliutinė paklaida gauta – 2,5 [U/g]. Vidutinė procentinė absoliutinė paklaida lygi 21,5 %.

## 2.2. Rekombinuotų baltymų sintezės greičio modeliavimas $q_{px}$ dirbtiniu neuroniniu tinklu (DNT)

Rekombinuotų baltymų sintezės greitis ( $q_{px}$ ) modeliuojamas dirbtiniu neuroniniu tinklu. Modeliuojamas dydis priklauso nuo biomasės augimo greičio ir rekombinuotų baltymų aktyvumo. DNT, visų pirma, apmokomi *Levenberg – Marquardt* algoritmu pagal mechanistinį modelį. Norint pagerinti apmokymo rezultatus, patikslinama jautrumo lygčių metodu.

Pasirenkama DNT struktūra, 2 įėjimai: biomasės augimo greitis ( $\mu$ ) ir rekombinuotų baltymų aktyvumas ( $p_x$ ), 2 paslėpto sluoksnio neuronai, 1 išėjimas: rekombinuotų baltymų sintezės greitis ( $q_{px}$ ). Prie dirbtinio neuroninio tinklo įėjimų ir paslėpto sluoksnio pridedama po papildomą neuroną (bias). Pagal šią DNT struktūrą, reikalingi 9 svoriniai koeficientai aprašyti tinklą. 9 svoriniai koeficientai, kadangi 2 įėjimai, 1 papildomas neuronas ir 2 paslėpto sluoksnio neuronai, tai  $(2+1)*2=6$  svoriniai koeficientai. Toliau 1 išėjimas, 2 paslėpto sluoksnio neuronai ir

1 papildomas, tai  $(2+1)*1=3$  svoriniai koeficientai. Bendra suma yra  $6+3=9$  svoriniai koeficientai.



Sluoksniai:      įėjimų              paslėptas              išėjimų

2.6 pav. DNT struktūra rekombinuotų baltymų sintezės greičio modeliavimui

### 2.2.1. DNT apmokymas *Levenberg – Marquardt* algoritmu

Dirbtinių neuroninių tinklų apmokymui naudojamas *Levenberg – Marquardt* algoritmas. Algoritmas realizuojamas *Matlab* programiniame pakete, naudojantis neuroninių tinklų sistemos identifikavimo rinkiniu (NNSYSID20).

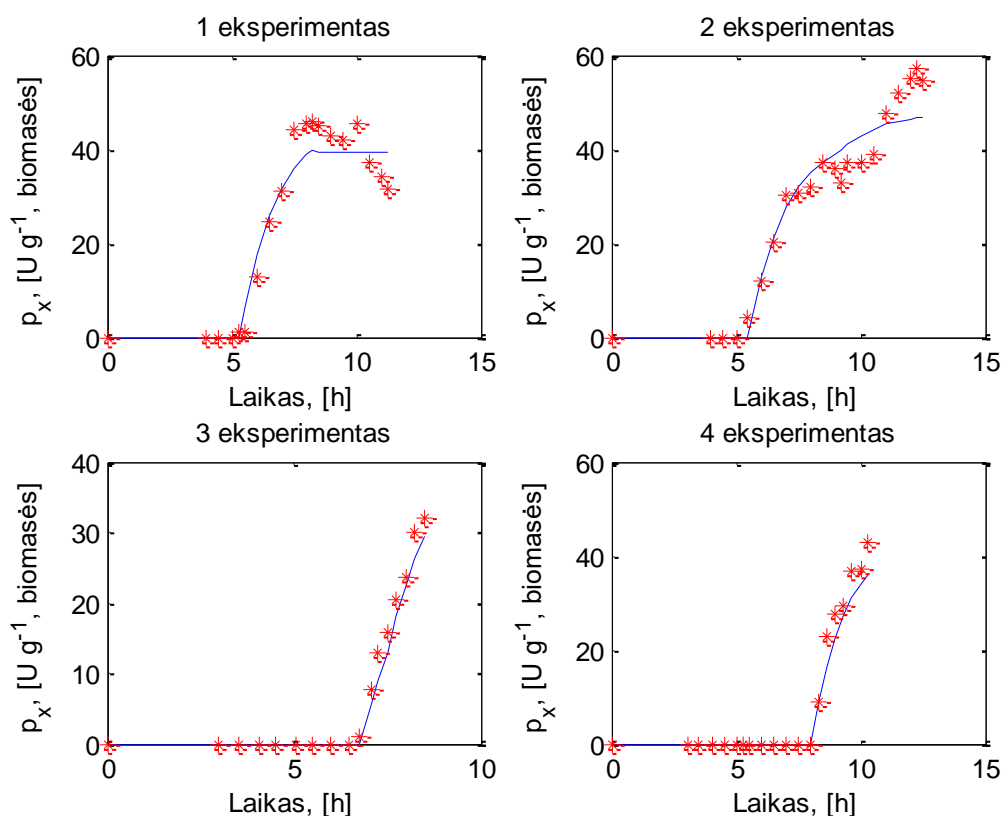
Visų pirma, sugeneruojami biomasės augimo greičio ( $\mu$ ) (0 – 0,4 intervalu 0,05) ir rekombinuotų baltymų aktyvumo ( $p_x$ ) (0 – 60 intervalu 1) duomenys. Naudojantis sugeneruotais duomenimis, pagal 2.9 ir 2.10 formules apskaičiuojamas rekombinuotų baltymų sintezės greitis ( $q_{px}$ ). Gauta duomenų matrica, kurią sudaro: biomasės augimo greitis ir rekombinuotų baltymų aktyvumas - įėjimai, o rekombinuotų baltymų sintezės greitis - išėjimas. DNT apmokymo metu siekiama rasti svorines vertes atitinkančias mechanistinį modelį. Pritaikyto *Levenberg – Marquardt* algoritmo programinis *Matlab* .m kodas pateikiamas priede nr. 4, *Levenberg – Marquardt* algoritmo apmokymo programos naudojamos *Matlab* paketo.

#### *Levenberg – Marquardt* algoritmo apmokymo rezultatai

Įvykdžius apmokymą, gauti svoriniai koeficientai: 0,1329; 0,0039; -0,4151; 30,8318; -104,8360; -9,7852; -183,7080; -1,9419; -41,6600.

Naudojant *Levenberg – Marquardt* algoritmo gautus svorinius koeficientus, modeliuojami eksperimentai. Modeliuojamas tik rekombinuotų baltymų aktyvumas ( $p_x$ ), kaip ir buvo minėta

anksčiau, jog pagrindinis dėmesys skiriamas šiam dydžiui. Pateikiamas modeliavimo ir eksperimentinių duomenų palyginimas:



2.7 pav. Rekombinuotų baltymų aktyvumo eksperimentinių duomenų ir modeliavimo rezultatų palyginimas, kai DNT apmokomas *Levenberg – Marquardt* algoritmu, naudojant 1 – 4 eksperimento duomenis su 2 neuronais paslėptame sluoksnyje. Ištinis linija – DNT modelio rezultatai, žvaigždės formos taškai – eksperimentiniai duomenys.

Naudojant *Levenberg – Marquardt* algoritmo svorinius koeficientus, 1 – 4 eksperimento vidutinė absoliutinė paklaida gauta – 2,5 [U/g], vidutinė procentinė absoliutinė paklaida lygi 17,7 %.

### 2.2.2. Hibridinio modelio parametrų identifikavimas jautrumo lygčių metodu

Remiantis *Levenberg – Marquardt* gautais modeliavimo rezultatais, iš grafikų ir paklaidų matoma, jog modeliavimo rezultatų nepavyko gauti geresniu nei mechanistinio modelio. Vidutinė absoliutinė paklaida mechanistinio modelio lygi 2,5 [U/g], o *Levenberg – Marquardt* gauto modelio – 2,5 [U/g]. *Levenberg – Marquardt* algoritmo rezultatų pagerinimui naudojamas jautrumo lygčių metodas. Įprastai jautrumo lygčių metode, paieška pradedama sugeneravus atsitiktinius dydžius. Šiuo atveju atlikus bandymus, naudojant atsitiktinius ir *Levenberg –*

*Marquardt* algoritmo svorinius koeficientus, su pastaraisiais gaunami tikslesni modeliavimo rezultatai, t.y. gaunamos mažesnės paklaidos.

Rekombinuotų baltymų augimo greitis ( $q_{px}$ ) modeliuojamas per biomasės augimo greitį ( $\mu$ ) ir rekombinuotų baltymų aktyvumą ( $p_x$ ), todėl naudojama 2.5 rekombinuotų baltymų aktyvumo masės balanso formulė:

$$\frac{dp_x}{dt} = q_{px}(\mu, p_x). \quad (2.13)$$

DNT apmokymui sudaroma jautrumo lygčių sistema. Naudojantis jautrumo lygčių formule (1.49), rekombinuotų baltymų augimo greitis gali būti aprašomas:

$$\frac{d}{dt} \frac{\partial p_x}{\partial w_n} = \frac{\partial f}{\partial p_x} \frac{\partial p_x}{\partial w_n} + \frac{\partial f}{\partial w_n}, \quad (2.14)$$

kur

$$\frac{\partial f}{\partial p_x} = \frac{q_{px}(\mu, p_x, w_n)}{p_x}, \quad (2.15)$$

ir

$$\frac{\partial f}{\partial w_n} = \frac{q_{px}(\mu, p_x, w_n)}{w_n}, \quad (2.16)$$

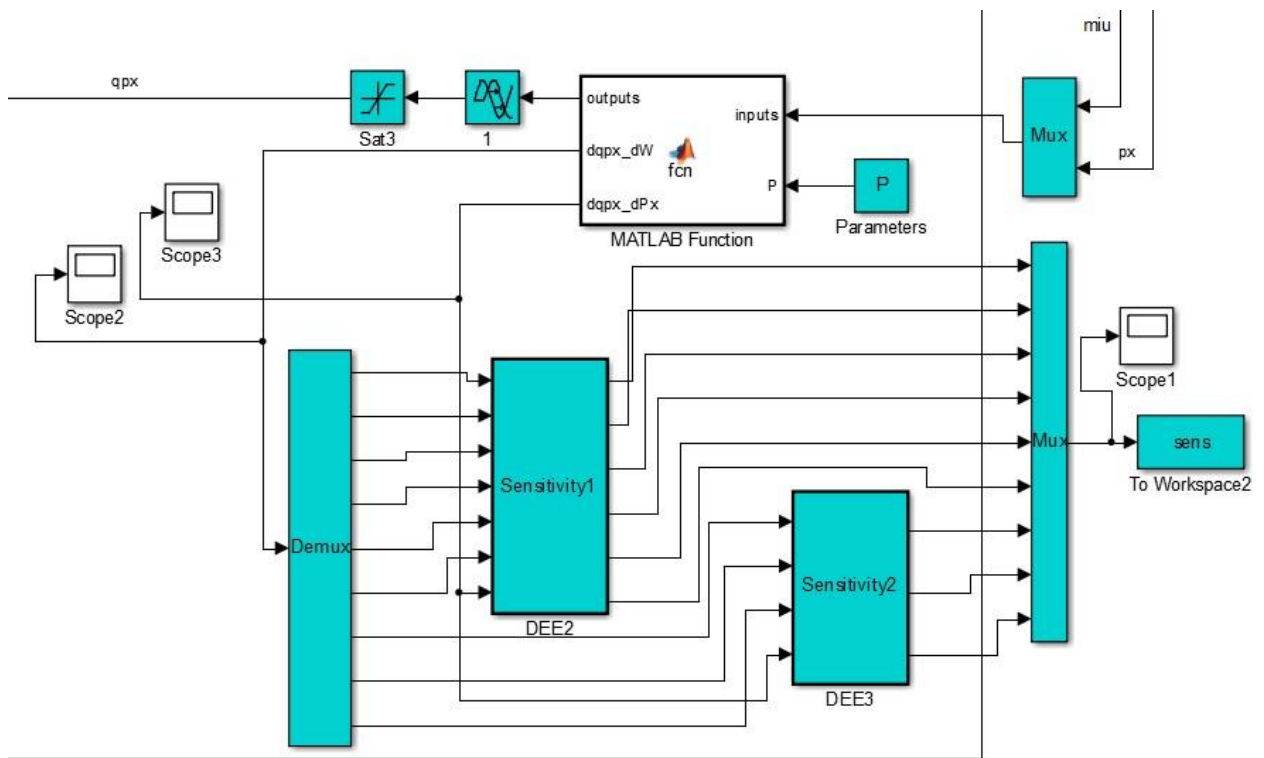
Gaunama jautrumo lygčių sistema:

$$\frac{d}{dt} \frac{\partial p_x}{\partial w_n} = \frac{q_{px}(\mu, p_x, w_n)}{p_x} \frac{\partial p_x}{\partial w_n} + \frac{q_{px}(\mu, p_x, w_n)}{w_n}. \quad (2.17)$$

Analogiškai sudaromos visos 9 jautrumo lygtys kiekvienam svoriniam koeficientui.

*Simulink* mechanistinis modelis (priedas nr. 3) koreguojamas, įterpiamas *Simulink* modelio fragmentas pavaizduotas 2.8 paveiksle. Rekombinuotų baltymų sintezės greitis ( $q_{px}$ ) modeliuojamas dirbtiniu neuroniniu tinklu.





2.8 pav. Simulink modelio fragmentas

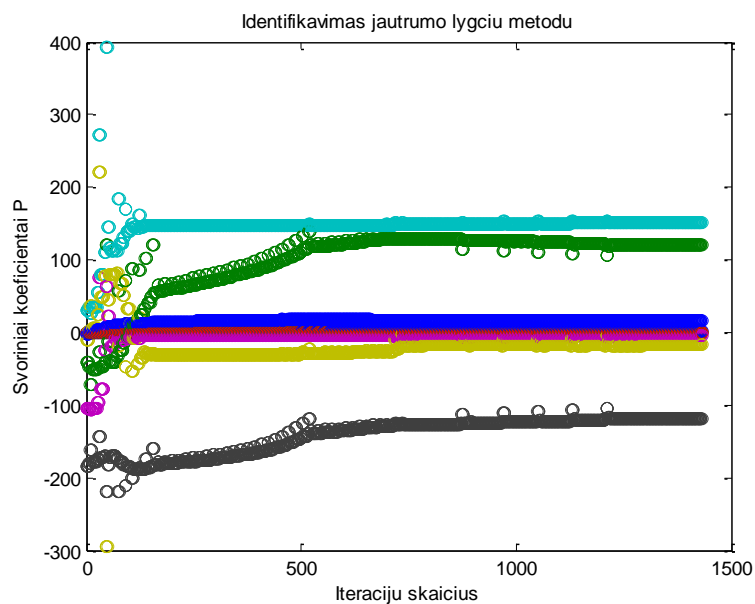
Modifikuotas proceso *Simulink* modelis pateikiamas priede Nr. 4. *Matlab function* bloke aprašomas DNT, tinklo struktūra bei atliekami skaičiavimai. Jautrumo lygtys suvedamos *DEE* blokuose. Priede nr. 5 pateikiamas *Matlab function* bloko programinis kodas ir *DEE* blokai.

Sukuriamas *Matlab* .m programinis kodas apmokyti dirbtinį neuroninį tinklą. Tęsimas anksčiau naudotas programinis kodas (priedas nr.1), kuris buvo naudotas mechanistiniam modeliui. *Levenberg – Marquardt* algoritmo gauti svoriniai koeficientai, naudojami kaip pradiniai paieškos taškai. Paieška įgyvendinama naudojant *Matlab lsqcurvefit* funkciją, kuri skirta netiesinių funkcijų skaičiavimui.

Paieškai sukuriama paprogramė, kurioje kiekvienoje iteracijoje generuojami P svoriniai koeficientai. Paieška vykdoma tol kol išnaudojamas iteracijų skaičius ar tenkinami paieškos kriterijai. Identifikuoti parametrai grąžinami į pagrindinę programą. Atliekamas eksperimentų duomenų palyginimas su modeliavimo rezultatais. Pagrindinė programa pateikiama priede nr.6, o paprogramė priede nr.7.

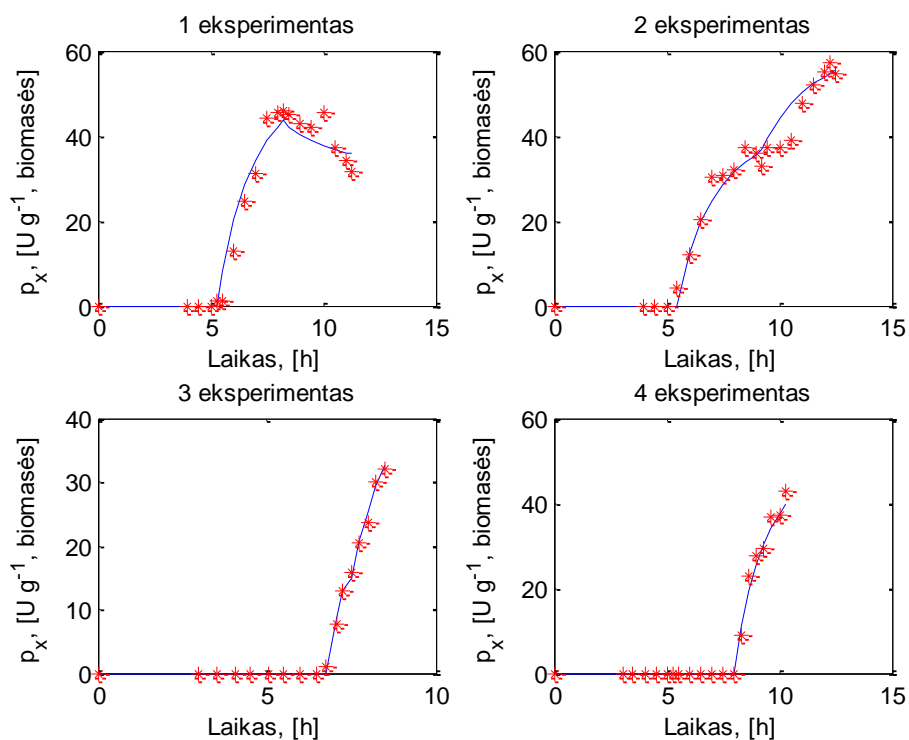
#### *Jautrumo lygčių metodo apmokymo rezultatai*

Apmokius DNT gautas P svorinių koeficientų vektorius: 1,7442; 0,0136; 0,4067; 151,1477; -5,2369; -17,5989; -115,7711; 15,4109; 117,3104. Apmokymo paieškos eiga (P vektoriaus dydžių kitimas didėjant iteracijų skaičiui):



2.9 pav. Identifikavimo procedūra jautrumo lygčių metodu, P dydžio priklausomybė nuo iteracijų.

Identifikavus modelio parametrus žemiau pateikiami rekombinuotų baltymų aktyvumo ( $p_x$ ) modeliavimo rezultatų palyginimas su eksperimentiniais duomenimis:



2.10 pav. Rekombinuotų baltymų aktyvumo eksperimentinių duomenų ir modeliavimo rezultatų palyginimas, kai DNT apmokomas jautrumo lygčių metodu, naudojant 1 – 4 eksperimento duomenis su 2 neuronais paslėptame sluoksnyje. Ištinė linija – DNT modelio rezultatai, žvaigždės formos taškai – eksperimentiniai duomenys.

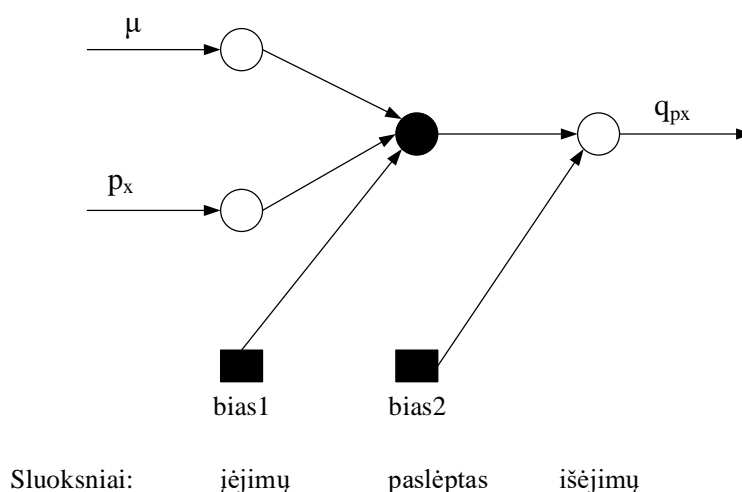
Identifikavus modelio parametrus jautrumo lygčių metodu, gaunama modelio vidutinė absoliutinė paklaida – 1,6 [U/g]. Visų eksperimentų vidutinė procentinė absoliutinė paklaida lygi 17,2 %.

### 2.2.3. Mokymo ir testavimo rezultatai

Įsitikinti, jog pasirinktas tinkamos struktūros dirbtinis neuroninis tinklas, atliekami mokymo ir testavimo bandymai. Naudojamas *cross - validation* metodas, kuris įvertina, kaip tiksliai modelis gali prognozuoti eksperimentus. Modeliui yra priskiriami duomenys, pagal kuriuos yra apmokomas ir duomenys, pagal kuriuos yra testuojamas. Šiuo atveju modelis apmokomas pagal 1 – 3 eksperimento duomenis, o testuojamas su 4 eksperimento duomenimis.

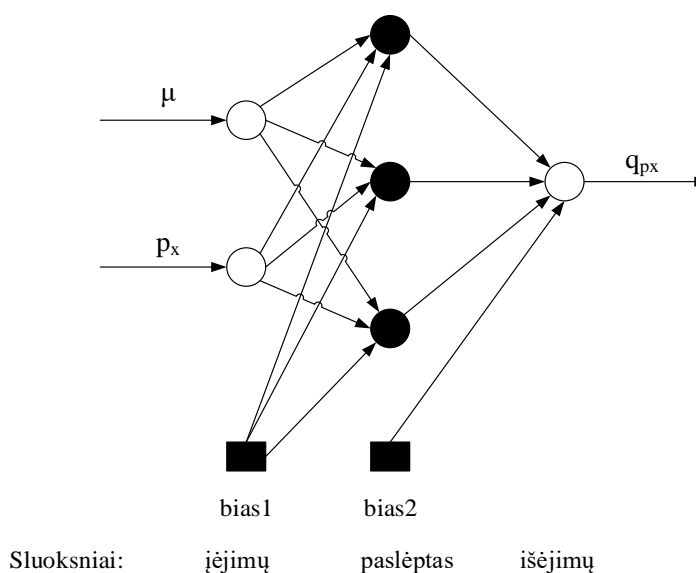
DNT apmokomas su 1 – 3 eksperimento duomenimis. Modeliuojamas 4 testavimo eksperimentas. Apmokoma naudojant 1, 2 ir 3 neuronus paslėptame sluoksnyje. Apmokius, apskaičiuojamos 4 testavimo eksperimento vidutinės absoliutinės ir vidutinės procentinės absoliutinės paklaidos. Apmokymas vykdomas analogiškai, kaip 2.2.1 ir 2.2.2 skyriuose. Pirmiausia apmokoma *Levenberg – Marquardt* algoritmu ir tikslinama jautrumo lygčių metodu.

Kai keičiamas DNT paslėptų neuronų skaičius, keičiasi ir svorinių koeficientų skaičius dirbtiniam neuroniniam tinklui. Pagal (2.11 pav.) DNT struktūrą, kurioje 1 neuronas paslėptame sluoksnyje, reikalingi 5 svoriniai koeficientai aprašyti tinklą. Kadangi 2 įėjimai, 1 papildomas neuronas (bias) ir 1 paslėpto sluoksnio neuronas, tai  $(2+1)*1=3$  svoriniai koeficientai. Toliau 1 išėjimas, 1 paslėpto sluoksnio neuronas ir 1 papildomas, tai  $(1+1)*1=2$  svoriniai koeficientai. Bendra suma yra  $3+2=5$  svoriniai koeficientai.



2.11 pav. DNT struktūra rekombinuotų baltymų sintezės greičio modeliavimui

Pagal (2.12 pav.) DNT struktūrą, kurioje 3 paslėpto sluoksnio neuronai, reikalinga 13 svorinių koeficientų aprašyti tinklą. Kadangi 2 įėjimai, 1 papildomas neuronas ir 3 paslėpto sluoksnio neuronai, tai  $(2+1)*3=9$  svoriniai koeficientai. Toliau 1 išėjimas, 3 paslėpto sluoksnio neuronai ir 1 papildomas, tai  $(3+1)*1=4$  svoriniai koeficientai. Bendra suma yra  $9+4=13$  svoriniai koeficientai.



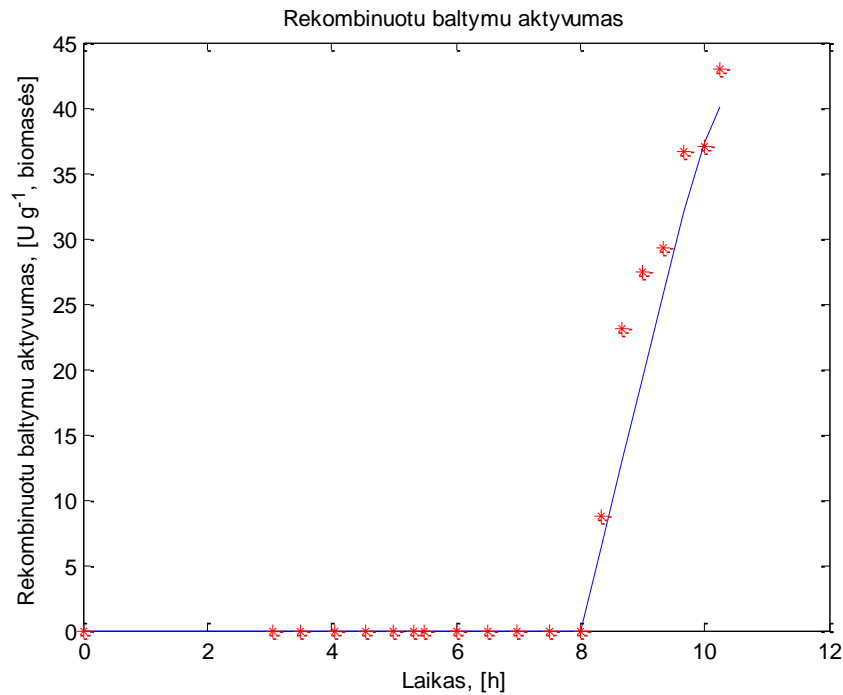
2.12 pav. DNT struktūra rekombinuotų baltymų sintezės greičio modeliavimui

### Apmokymas 1 – 3 eksperimento duomenimis, naudojant 1 neuroną paslėptame sluoksnyje

Atliekamas apmokymas su 1 – 3 eksperimento duomenimis naudojant 1 neuroną paslėptame sluoksnyje testavimo rezultatams gauti. Modeliuojamas tik 4 testavimo eksperimentas ir apskaičiuojamos paklaidos.

*Levenberg – Marquardt* algoritme sumažinamas paslėpto sluoksnio neuronų skaičius iki 1. Pirmiausiai DNT apmokomi su sugeneruota duomenų matrica *Levenberg – Marquardt* algoritmu. Apmokius gaunami svoriniai koeficientai: 4,9942; 0,0851; -3,6721; -14,4273; 10,4845. Gauti svoriniai koeficientai naudojami, kaip pradiniai paieškos taškai jautrumo lygčių metode. Prieš pradėdant identifikavimą, pakeičiama jautrumo lygčių metode DNT struktūra, t.y. *Matlab function* bloke nustatomas 1 neuronas paslėptame sluoksnyje. Identifikavus jautrumo lygčių metodu gaunami svoriniai koeficientai: 9,1455; 0,1593; -7,4351; -11,0624; 8,2607.

Identifikavus modelį, pateikiama 4 eksperimento modeliavimo rezultatų ir eksperimentinių duomenų palyginimas:



2.13 pav. Rekombinuotų baltymų aktyvumo eksperimentinių duomenų ir modeliavimo rezultatų (4 testavimo eksperimento) palyginimas, kai DNT apmokomas jautrumo lygčių metodu, naudojant 1 – 3 eksperimento duomenis su 1 neuronu paslėptame sluoksnyje. Ištinė linija – DNT modelio rezultatai, žvaigždės formos taškai – eksperimentiniai duomenys.

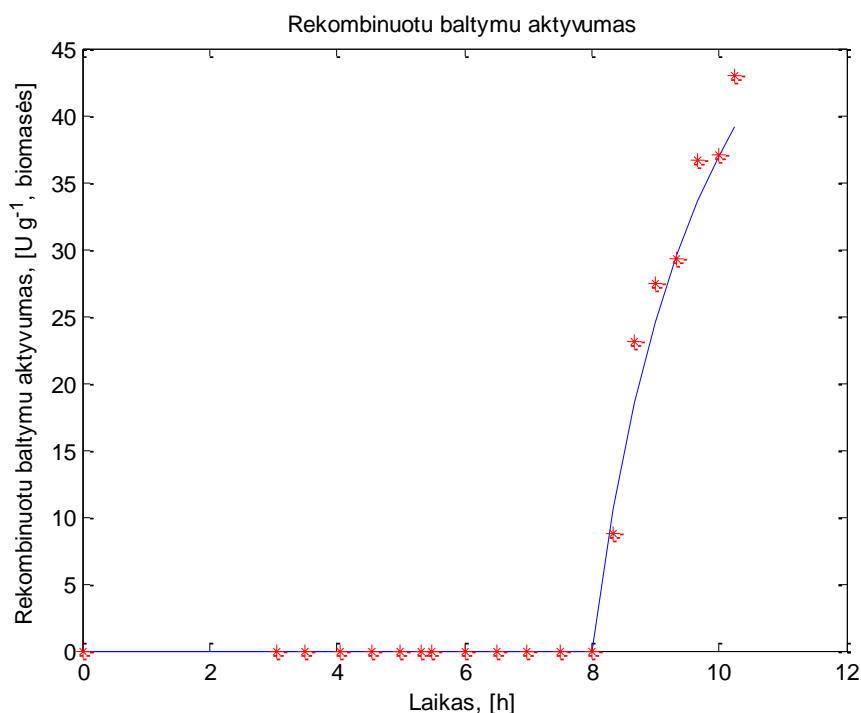
Apmokius pagal 1 – 3 eksperimento duomenis su 1 neuronu paslėptame sluoksnyje gaunama 4 eksperimento vidutinė absoliutinė paklaida – 1,6 [U/g], vidutinė procentinė absoliutinė paklaida – 6,7 %.

#### **Apmokymas 1 – 3 eksperimento duomenimis, naudojant 2 neuronus paslėptame sluoksnyje**

Atliekamas apmokymas su 1 – 3 eksperimento duomenimis naudojant 2 neuronus paslėptame sluoksnyje testavimo rezultatams gauti. Modeliuojamas tik 4 testavimo eksperimentas ir apskaičiuojamos paklaidos.

*Levenberg – Marquardt* algoritme pakeičiamas paslėpto sluoksnio neuronų skaičius į 2 neuronus. Jautrumo lygčių metode pradiniai paieškos taškai naudojami 2.2.1 skyriuje gauti *Levenberg – Marquardt* algoritmo svoriniai koeficientai: 0,1329; 0,0039; -0,4151; 30,8318; -104,8360; -9,7852; -183,7080; -1,9419; -41,6600. Prieš pradėdant identifikavimą, pakeičiama jautrumo lygčių metode DNT struktūra, t.y. *Matlab function* bloke nustatomas 2 neuronų skaičius. Identifikavus jautrumo lygčių metodu gaunami svoriniai koeficientai: 1,5866; 0,0125; 0,4663; 150,8282; -5,0808; -17,5057; -118,8574; 15,3300; 119,5956.

Identifikavus modelį, pateikiama 4 eksperimento modeliavimo rezultatų ir eksperimentinių duomenų palyginimas:



2.14 pav. Rekombinuotų baltymų aktyvumo eksperimentinių duomenų ir modeliavimo rezultatų (4 testavimo eksperimento) palyginimas, kai DNT apmokomas jautrumo lygčių metodu, naudojant 1 – 3 eksperimento duomenis su 2 neuronais paslėptame sluoksnyje. Ištinis linija – DNT modelio rezultatai, žvaigždės formos taškai – eksperimentiniai duomenys.

Apmokius pagal 1 – 3 eksperimento duomenis su 2 neuronais paslėptame sluoksnyje gaunama 4 eksperimento vidutinė absoliutinė paklaida – 0,8 [U/g], vidutinė procentinė absoliutinė paklaida – 3,4 %.

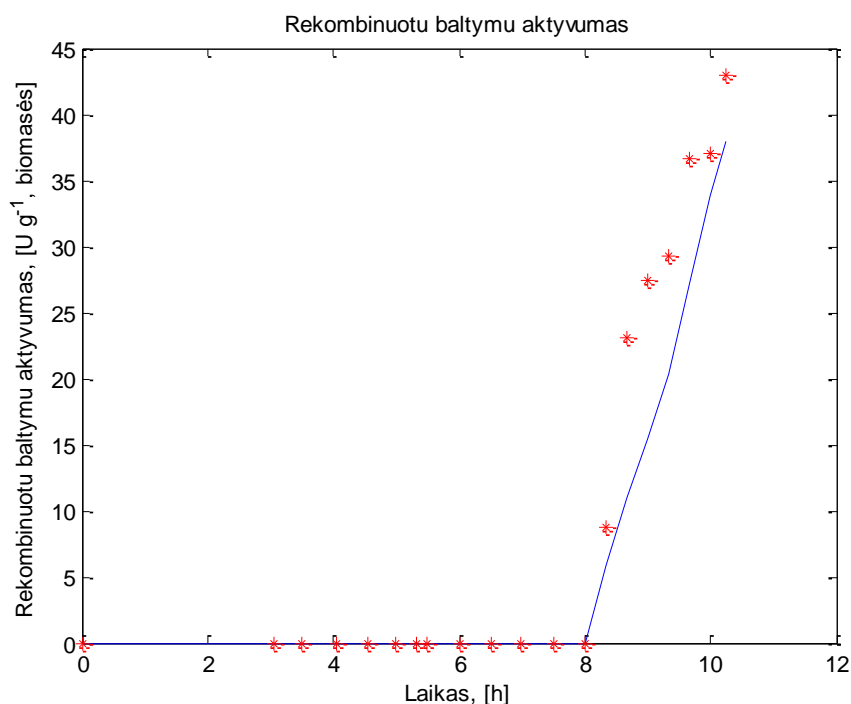
### **Apmokymas 1 – 3 eksperimento duomenimis, naudojant 3 neuronus paslėptame sluoksnyje**

Atliekamas apmokymas su 1 – 3 eksperimento duomenimis naudojant 3 neuronus paslėptame sluoksnyje testavimo rezultatams gauti. Modeliuojamas tik 4 testavimo eksperimentas ir apskaičiuojamos paklaidos.

*Levenberg – Marquardt* algoritme padidinamas paslėpto sluoksnio neuronų skaičius iki 3. Pirmiausiai DNT apmokomi *Levenberg – Marquardt* algoritmu. Apmokius gaunami svoriniai koeficientai: 3,6646; 0; 0,1307; 15,0747; 0; 0,1055; 0,0039; 0,0001; -0,0049; -40,8; 66,5; -4923,3; -25,6, kurie naudojami, kaip pradiniai paieškos taškai jautrumo lygčių metode. Prieš pradėdant identifikavimą, pakeičiama jautrumo lygčių metode DNT struktūra, nustatomi 3

neuronai paslėptame sluoksnyje. Identifikavus jautrumo lygčių metodu gaunami svoriniai koeficientai: 6,6074; 0,0177; -1,6089; -107,9560; -0,1996; 15,8525; -0,1577; 0,00011377; -0,0081; -45,3891; -10,8045; -866,9360; -24,0568

Naudojant šį modelį, pateikiama 4 eksperimento modeliavimo rezultatų ir eksperimentinių duomenų palyginimas:



2.15 pav. Rekombinuotų baltymų aktyvumo eksperimentinių duomenų ir modeliavimo rezultatų (4 testavimo eksperimento) palyginimas, kai DNT apmokomas jautrumo lygčių metodu, naudojant 1 – 3 eksperimento duomenis su 3 neuronais paslėptame sluoksnyje. Ištinė linija – DNT modelio rezultatai, žvaigždės formos taškai – eksperimentiniai duomenys.

Apmokius pagal 1 – 3 eksperimento duomenis su 3 neuronais paslėptame sluoksnyje gaunama 4 eksperimento vidutinė absoliutinė paklaida – 2,7 [U/g], vidutinė procentinė absoliutinė paklaida – 10,3 %.

#### 2.2.4. Mokymo ir testavimo rezultatų palyginimas

2.3 lentelėje pateikiamos paklaidos, kai DNT apmokomi tik pagal 1 – 3 eksperimento duomenis. Pateikiamos 1 – 3 eksperimento paklaidos, kai apmokyti naudoti 1, 2 ir 3 neuronai paslėptame sluoksnyje.

2.3 lentelė. 1 - 3 eksperimentų paklaidų palyginimas

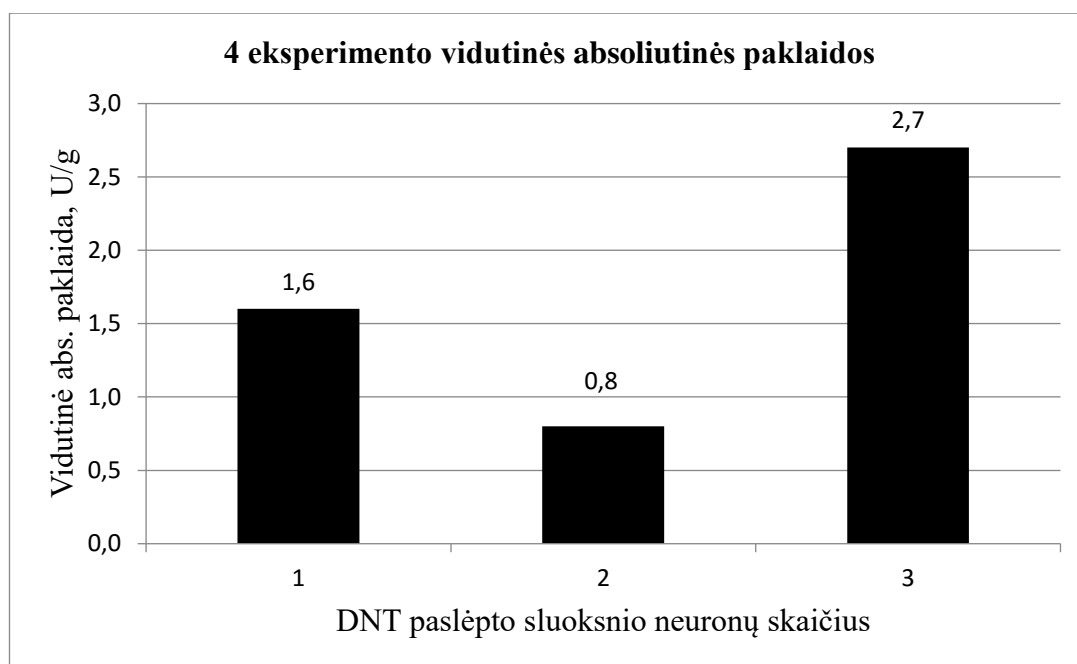
Neuronų sk.	1	2	3
Vidutinė abs. paklaida, U/g	2,3	1,9	1,1
Vidutinė procentinė absoliutinė paklaida, %	17,7	21,2	13,6

2.4 lentelėje pateikiamos paklaidos tik modeliuoto 4 testavimo eksperimento. Modeliavimui naudoti svoriniai koeficientai, kurie gauti DNT apmokius pagal 1 – 3 eksperimento duomenis. Naudoti 1, 2 ir 3 neuronai paslėptame sluoksnyje Pateikiama vidutinė absoliutinė paklaida bei vidutinė procentinė absoliutinė paklaida.

2.4 lentelė. 4 eksperimento paklaidų palyginimas

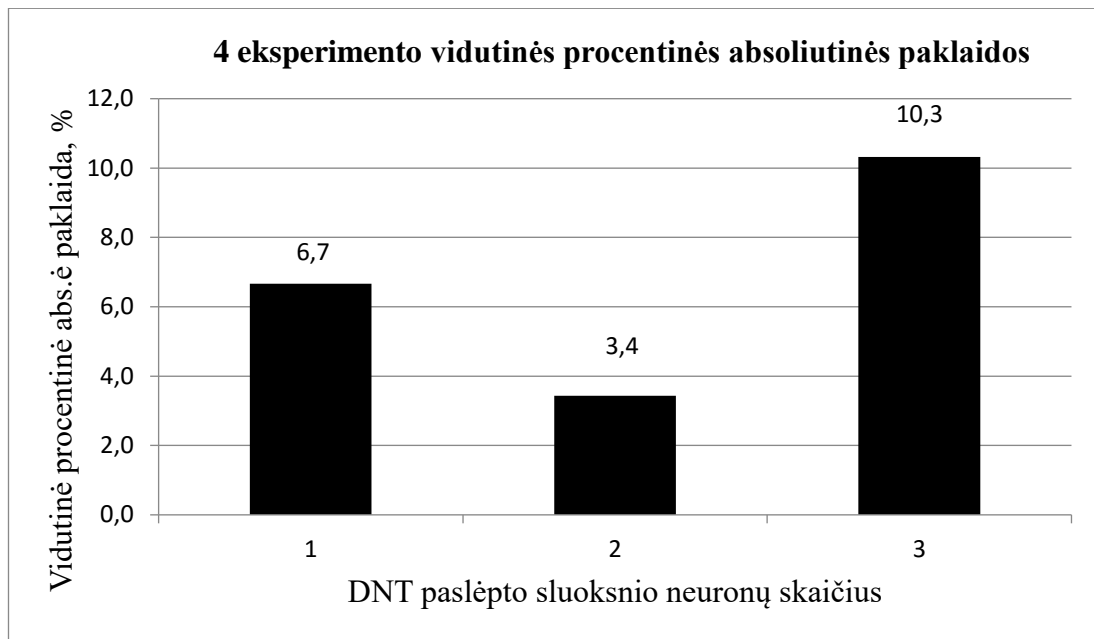
Neuronų sk.	1	2	3
Vidutinė abs. paklaida, U/g	1,6	0,8	2,7
Vidutinė procentinė absoliutinė paklaida, %	6,7	3,4	10,3

Remiantis 2.4 lentelės duomenimis, pateikiamos gautos paklaidų diagramos:



2.16 pav. 4 testavimo eksperimento vidutinių absoliutinių paklaidų diagrama, DNT apmokius pagal 1 – 3 eksperimento duomenis jautrumo lygčių metodu.





2.17 pav. 4 testavimo eksperimento vidutinių procentinių absoliutinių paklaidų diagrama, DNT apmokius pagal 1 – 3 eksperimento duomenis jautrumo lygčių metodu.

Naudojamas *cross - validation* metodas, kai modelio apmokymui priskiriami 1 – 3 eksperimentiniai duomenys, o testuojama su 4 eksperimento duomenimis. Iš 4 eksperimento rezultatų galime spręsti, kuris modelis tiksliausiai prognozuoja procesą

Iš 2.3 lentelės matome, jog DNT apmokius pagal 1 – 3 eksperimento duomenis, mažiausios paklaidos 1 – 3 eksperimento gaunamos su 3 neuronais paslėptame sluoksnyje. Tačiau iš 2.4 lentelės bei diagramų (2.16 pav. ir 2.17 pav.) matome, jog 4 eksperimento vidutinė absoliutinė ir vidutinė procentinė absoliutinė paklaida gauta mažiausia su 2 neuronais paslėptame sluoksnyje. Todėl remiantis *cross – validation* metodu galime teigti, kad dirbtinis neuroninis tinklas turintis 2 neuronus paslėptame sluoksnyje tiksliausiai prognozuoja šį procesą. Procesui pasirinktas teisingos struktūros dirbtinis neuroninis tinklas.

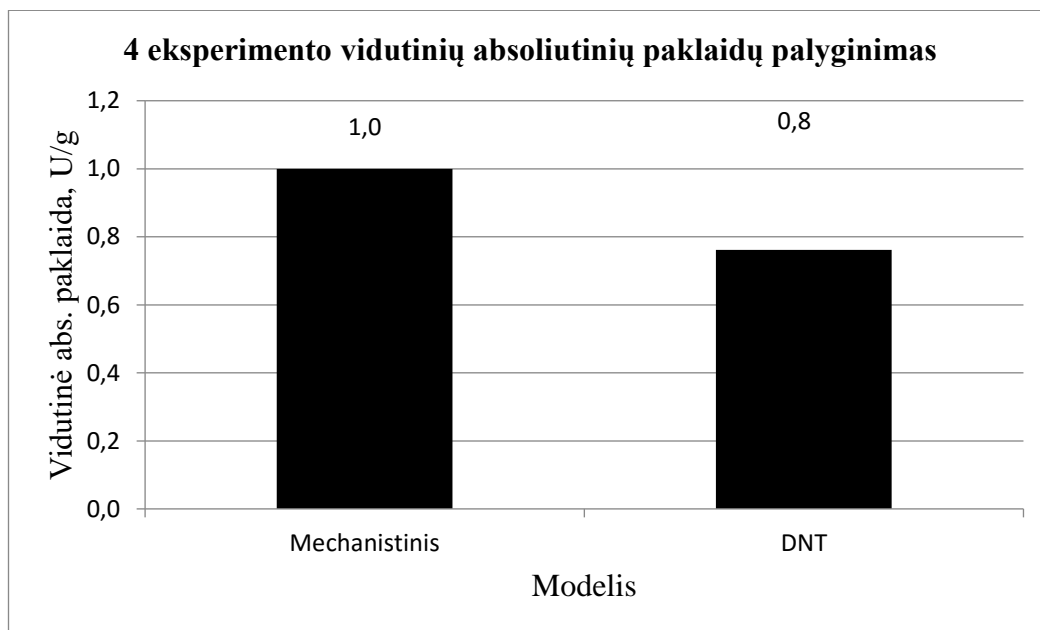
### 2.3. Mechanistinio ir hibridinio modelio rezultatų palyginimas

2.5 lentelėje pateikiamos mechanistinio ir dirbtiniu neuroniniu tinklu modeliuoto 4 eksperimento vidutinės absoliutinės ir vidutinės procentinės absoliutinės paklaidos. DNT apmokytas pagal 1 – 3 eksperimento duomenis su optimaliu neuronų skaičiumi paslėptame sluoksnyje, t.y. 2 neuronais.

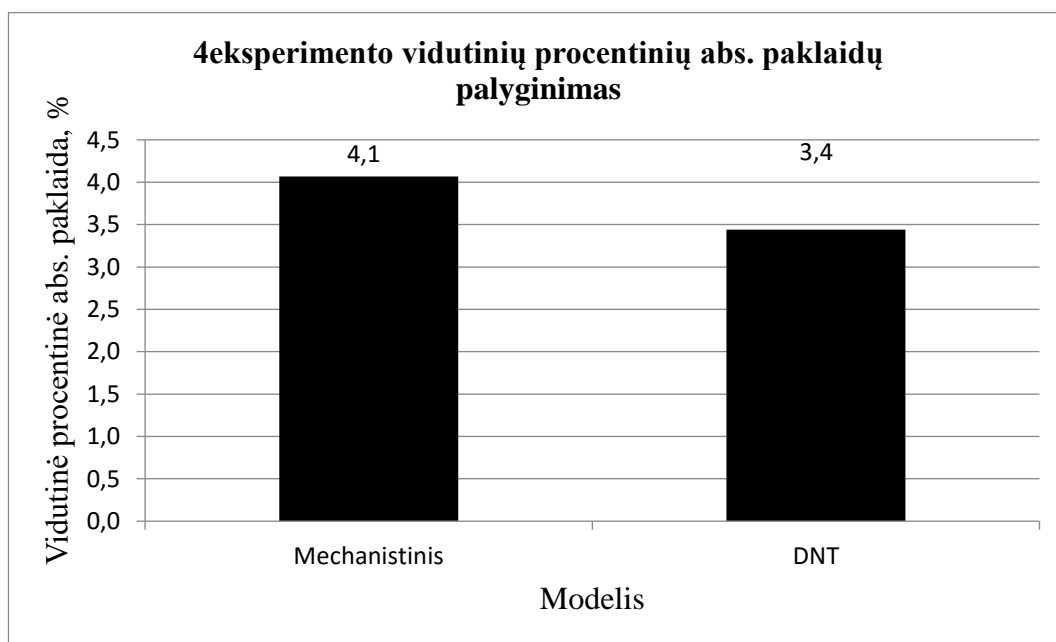
2.5 lentelė. 4 testavimo eksperimento paklaidų palyginimas

Modelis	Mechanistinis	DNT
Vidutinė abs. paklaida, U/g	1	0,8
Vidutinė procentinė absoliutinė paklaida, %	4,1	3,4

Remiantis 2.5 lentelės duomenimis, pateikiamos gautos paklaidų diagramos:



2.18 pav. Mechanistinio ir DNT modeliuto 4 testavimo eksperimento vidutinių absoliutinių paklaidų palyginimas.



2.19 pav. Mechanistinio ir DNT modeliuto 4 testavimo eksperimento vidutinių absoliutinių paklaidų palyginimas.

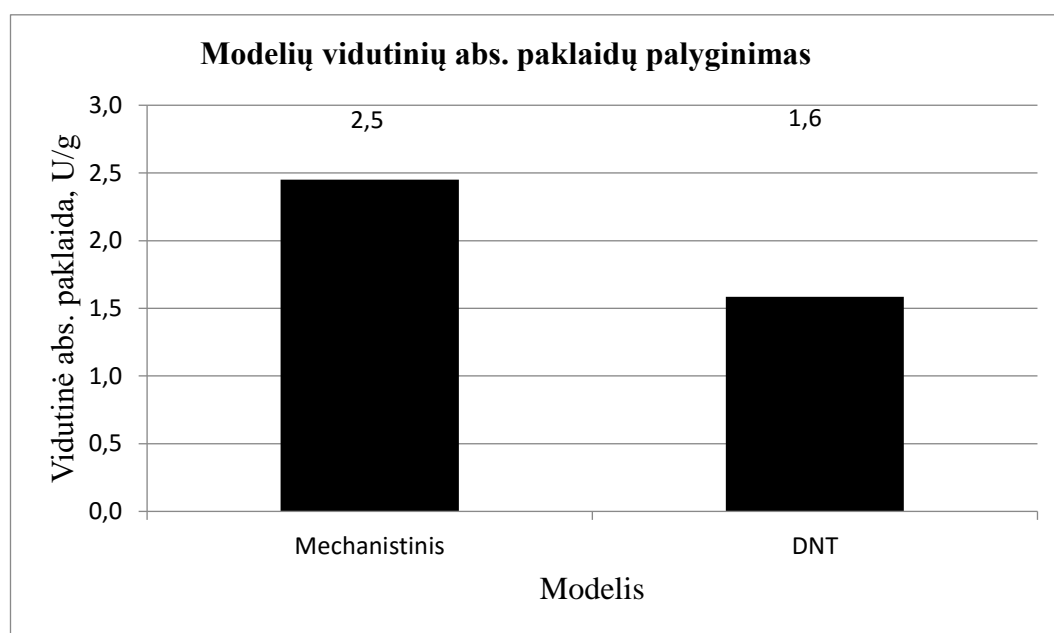
Iš 2.5 lentelės bei diagramų (2.18 pav. ir 2.19 pav.) matome, jog DNT modelio, 4 testavimo eksperimento vidutinė absoliutinė ir vidutinė procentinė absoliutinė paklaida gauta mažesnė lyginant su mechanistiniu modeliu.

2.6 lentelėje pateikiamos mechanistinio ir dirbtiniu neuroniniu tinklu modeliuotų 1 – 4 eksperimento vidutinės absoliutinės ir vidutinės procentinės absoliutinės paklaidos. DNT apmokytas pagal 1 – 3 eksperimento duomenis su optimaliu neuronų skaičiumi paslėptame sluoksnyje, t.y. 2 neuronais. Pateikiamos 1 – 4 eksperimento paklaidos.

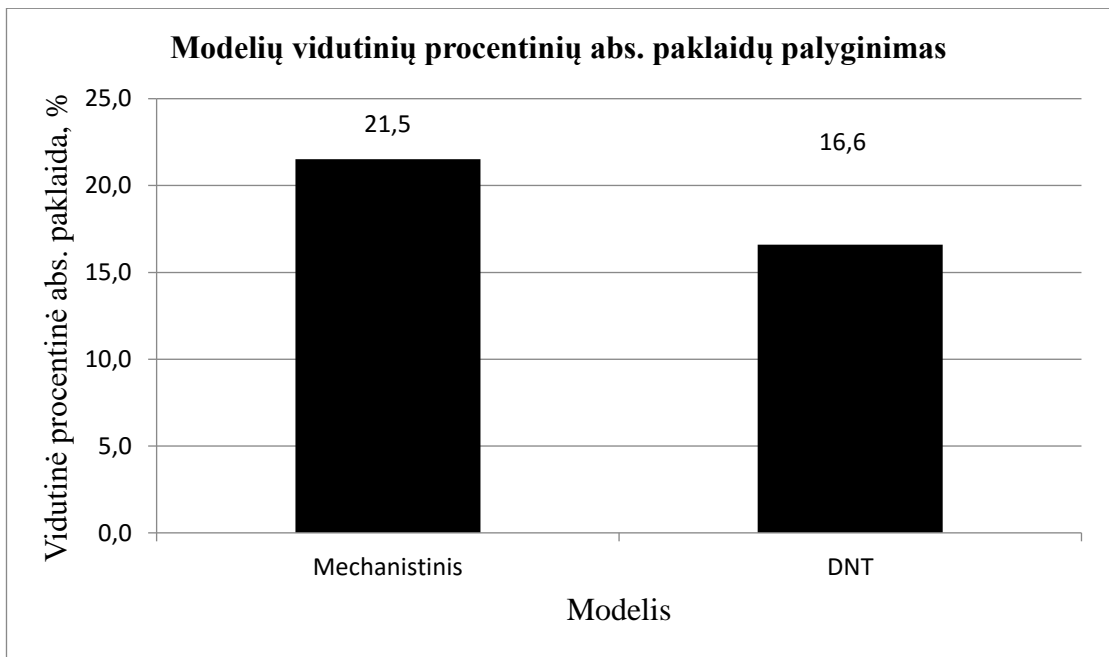
2.6 lentelė. Modelių paklaidų palyginimas

Modelis	Mechanistinis	DNT
Vidutinė abs. paklaida, U/g	2,5	1,6
Vidutinė procentinė absoliutinė paklaida, %	21,5	16,6

Remiantis 2.6 lentelės duomenimis, pateikiamos gautos paklaidų diagramos:



2.20 pav. Mechanistinio ir DNT modelių vidutinių absoliutinių paklaidų palyginimas.



2.21 pav. Mechanistinio ir DNT modelių vidutinių absoliutinių paklaidų palyginimas.

Iš 2.6 lentelės bei diagramų (2.20 pav. ir 2.21 pav.) matome, jog modeliuojant dirbtiniu neuroniniu tinklu, rezultatai pagerėjo, gaunamas modeliavimas tiksliau atitinkantis eksperimentinius duomenis. Sumažėjo vidutinė absoliutinė ir vidutinė procentinė absoliutinė paklaida.

Rekombinuotų baltymų sintezės greičio ( $q_{px}$ ) modeliavimui pakeitus proceso mechanistinį modelį į hibridinį su dirbtiniu neuroniniu tinklu pastebime, jog modeliavimo rezultatai pagerėja. Proceso vidutinė absoliutinė ir vidutinė procentinė paklaida sumažėja. Modelis tiksliau prognozuoja eksperimentus. Identifikuotas modelis gali būti taikomas tolimesniam proceso optimizavimui, kurio tikslas apskaičiuoti optimalią valdymo strategiją tikslinio produkto kiekiui maksimizuoti.

## IŠVADOS

1. Atlikta literatūros analizė. Apžvelgti biotechnologiniai procesai, bioreaktoriai, jų tipai, sudedamosios dalys, veikimas, vykstančios reakcijos, išskirti matuojami kintamieji. Išanalizuotos medžiagų masės balansų lygtys bei reakcijų greičių modeliai. Apžvelgti hibridiniai modeliai, tipinės modelių struktūros, modelių identifikavimo metodai.

2. Pagal pateiktas proceso balanso lygtis, greičių reakcijas ir kitus parametrus sukurtas mechanistinis proceso *Matlab / Simulink* modelis.

3. Rekombinuotų baltymų sintezės greičio ( $q_{px}$ ) modeliavimui mechanistinis modelis pakeistas dinaminis su dirbtiniu neuroniniu tinklu. DNT apmokyti *Levenberg – Marquardt* algoritmu. Modelio identifikavimo kokybė gerinama jautrumo lygčių metodu.

4. Atliekant modelio testavimą, 1 – 3 eksperimento duomenys priskirti DNT apmokymui, 4 eksperimentas testavimui. Atlikus bandymus su 1, 2 ir 3 neuronais paslėptame sluoksnyje, 4 eksperimento testavimo paklaidos gautos mažiausios su 2 neuronais paslėptame sluoksnyje. Vidutinė absoliutinė paklaida – 0,8 [U/g], vidutinė procentinė absoliutinė paklaida – 3,4 %. DNT su 2 neuronais paslėptame sluoksnyje tiksliausiai prognozuoja šį procesą.

5. Mechanistinio modelio 1 – 4 eksperimento vidutinė absoliutinė paklaida lygi - 2,5 [U/g], vidutinė procentinė absoliutinė paklaida - 21,5 % , 4 testavimo eksperimento vidutinė absoliutinė paklaida - 1 [U/g], vidutinė procentinė absoliutinė paklaida - 4,1 %. DNT apmokyto pagal 1 – 3 eksperimento duomenis su 2 neuronais paslėptame sluoksnyje, 4 testavimo eksperimento vidutinė absoliutinė paklaida - 0,8 [U/g], vidutinė procentinė absoliutinė paklaida - 3,4 %. Modelio 1 – 4 eksperimento vidutinė absoliutinė paklaida gauta - 1,6 [U/g], vidutinė procentinė absoliutinė paklaida – 16,6 %, Palyginus paklaidas pastebėta, jog DNT modelis tiksliau prognozuoja procesą, gaunamos mažesnės paklaidos.

6. Identifikuotas modelis su DNT gali būti taikomas tolimesniam proceso optimizavimui, kurio tikslas apskaičiuoti optimalią valdymo strategiją tikslinio produkto kiekiui maksimizuoti.

## INFORMACIJOS ŠALTINIŲ SĄRAŠAS

1. Biotechnology. [Žiūrėta: 2016-03-15]. Prieiga per internetą: <https://en.wikipedia.org/wiki/Biotechnology>
2. Rolandas Meškys. Biotechnologija. [Žiūrėta: 2016-03-15]. Prieiga per internetą: [http://www.esparama.lt/es\\_parama\\_pletra/failai/ESFproduktai/2011\\_Biotechnologija.pdf](http://www.esparama.lt/es_parama_pletra/failai/ESFproduktai/2011_Biotechnologija.pdf)
3. A. Kanopka. Biotechnologijos. [Žiūrėta: 2016-03-15]. Prieiga per internetą: <http://molbio.vdu.lt/medziaga/Kanopka/Biotechnologija.pdf>
4. Bioreactor. [Žiūrėta: 2016-03-16]. Prieiga per internetą: <https://en.wikipedia.org/wiki/Bioreactor>
5. Chemical reactor. [Žiūrėta: 2016-03-18]. Prieiga per internetą: [https://en.wikipedia.org/wiki/Chemical\\_reactor](https://en.wikipedia.org/wiki/Chemical_reactor)
6. M. Pohlscheidt, S. Charaniya, C. Bork. BIOPROCESS AND FERMENTATION MONITORING. [Žiūrėta: 2016-03-15]. Prieiga per internetą: [https://lifescience.roche.com/wcsstore/CBCatalogAssetStore/Articles/2013\\_Encyclopedia.pdf](https://lifescience.roche.com/wcsstore/CBCatalogAssetStore/Articles/2013_Encyclopedia.pdf)
7. B. Sonnleitner. Advances in biochemical engineering biotechnology, vol.66. 1999m. Berlin.
8. Vytautas Galvanauskas, Donatas Levišauskas. Biotechnologinių procesų modeliavimas, optimizavimas ir valdymas. 2008 m. Vilnius. Mokomoji knyga.
9. Ghasem D. Najafpour. Biochemical Engineering and Biotechnology. Professor of Ghasem D. Najafpour - Chemical Engineering Noshirvani Institute of Technology University of Mazandaran Babol, Iran. Knyga.
10. Dzemyda G., Kurasova O., Žilinskas J. Daugiamačių duomenų vizualizavimo metodai. Vilnius: Mokslo aidai. 2008. [Žiūrėta: 2016-03-31]. Prieiga per internetą: <http://web.vu.lt/mii/j.zilinskas/DzemydaKurasovaZilinskasDDVM.pdf>
11. Egidijus Dragašius, Arūnas Lipnickas. Intelektualios mechatroninės sistemos. Mokomoji knyga. 2008m. Vilnius. [Žiūrėta: 2016-03-31]. Prieiga per internetą: <https://www.ebooks.ktu.lt/eb/437/intelektualios-mechatronines-sistemas/>
12. Sandra Strigūnaitė, Dalia Krikščiūnienė. Intelektualus duomenų analizės metodas virtualaus komandinio darbo vertinimui. 2012m. [Žiūrėta: 2016-03-31]. Prieiga per internetą: [https://www.mruni.eu/upload/iblock/1ad/001\\_strigunaite\\_kriksciuniene.pdf](https://www.mruni.eu/upload/iblock/1ad/001_strigunaite_kriksciuniene.pdf)
13. A. Torres, J. J. Nieto. Fuzzy logic in Medicine and Bioinformatics. 2006m. [Žiūrėta: 2017-04-15]. Prieiga per internetą: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1559939/>

14. B. Center, B. P. Verma Fuzzy logic for Biological and Agricultural Systems. 1998m. [Žiūrėta: 2017-04-15]. Prieiga per internetą: <https://link.springer.com/article/10.1023/A:1006577431288>
15. V. Galvanauskas, R. Simutis, A. Luebbert. Hybrid process models for process optimisation, monitoring and control. 2004m. [Žiūrėta: 2016-04-15]. Prieiga per internetą: <http://link.springer.com/article/10.1007%2Fs00449-004-0385-x>
16. C. Pertev, M. Turker, R. Berber. Dynamic modeling, sensitivity analysis and parameter estimation of industrial yeast fermenters. [Žiūrėta: 2017-04-15]. Prieiga per internetą: [https://www.researchgate.net/profile/Mustafa\\_Tuerker/publication/223376004\\_Dynamic\\_modeling\\_sensitivity\\_analysis\\_and\\_parameter\\_estimation\\_of\\_industrial\\_yeast\\_fermenters/links/565da71b08aefe619b26648b/Dynamic-modeling-sensitivity-analysis-and-parameter-estimation-of-industrial-yeast-fermenters.pdf](https://www.researchgate.net/profile/Mustafa_Tuerker/publication/223376004_Dynamic_modeling_sensitivity_analysis_and_parameter_estimation_of_industrial_yeast_fermenters/links/565da71b08aefe619b26648b/Dynamic-modeling-sensitivity-analysis-and-parameter-estimation-of-industrial-yeast-fermenters.pdf)
17. Cross-validation (statistics). [Žiūrėta: 2017-04-16]. Prieiga per internetą: [https://en.wikipedia.org/wiki/Cross-validation\\_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))
18. H. Yu, B. M Wilamowski. Levenberg-Marquardt Training. [Žiūrėta: 2016-04-20]. Prieiga per internetą: [http://www.eng.auburn.edu/~wilambm/pap/2011/K10149\\_C012.pdf](http://www.eng.auburn.edu/~wilambm/pap/2011/K10149_C012.pdf)
19. D. Levisauskas, V. Galvanauskas, S. Henrich, K. Wilhelm, N. Volk, A. Luebbert. Model-based optimization of viral capsid protein production in fed-batch culture of recombinant Escherichia coli. 2002m. [Žiūrėta: 2016-04-15]. Prieiga per internetą: <http://link.springer.com/article/10.1007%2Fs00449-002-0305-x>
20. R. Simutis. Sistemų modeliavimas ir identifikavimas. 2008m. Vilnius. Mokomoji knyga.
21. Vytautas Galvanauskas, Donatas Levišauskas. Biotechnologinių procesų modeliavimas, optimizavimas ir valdymas. 2008 m. Vilnius. Praktikumai ir uždaviniai.
22. F. Streichert. University of Tuebingen. Introduction of Evolutionary Algorithms. [Žiūrėta: 2016-04-20]. Prieiga per internetą: [http://www.ra.cs.uni-tuebingen.de/mitarb/streiche/publications/Introduction\\_to\\_Evolutionary\\_Algorithms.pdf](http://www.ra.cs.uni-tuebingen.de/mitarb/streiche/publications/Introduction_to_Evolutionary_Algorithms.pdf)
23. D. Levišauskas, V. Galvanauskas, S. Henrich, K. Vilhelm, N. Volk, A. Luebbert. Model-based optimization of viral capsid protein production in fed-batch culture of recombinant Escherichia coli. 2003m. [Žiūrėta: 2016-04-20]. Prieiga per internetą: <https://link.springer.com/article/10.1007%2Fs00449-002-0305-x>

# PRIEDAI

## Priedas nr. 1, Matlab .m programinis kodas

```
close all, clear all, clc, warning off
global t1 px1 sens u_x u_y x1 s1 w1 x0 s0 px0 w0 t_ind sf iexp P
global it px_start_ind x00 s00 px00 w00 t_ind0 sf00 px_start_indukcija
global pirm_band_laikas antr_band_laikas trec_band_laikas ketv_band_laikas
global pirm_band_x antr_band_x trec_band_x ketv_band_x
global pirm_band_s antr_band_s trec_band_s ketv_band_s
global pirm_band_w antr_band_w trec_band_w ketv_band_w
global pirm_band_px antr_band_px trec_band_px ketv_band_px
global u_x_1 u_y_1 u_x_2 u_y_2 u_x_3 u_y_3 u_x_4 u_y_4
data=load('new.DAT'); %Analiziniai duomenys, 4eksperm.
data1=load('vp1online11.DAT');%Pamaitinimo profilio duomenys.
pirm_band_laikas=data(1:19,1); % 1 eksp laikas.
antr_band_laikas=data(20:40,1);% 2 eksp laikas.
trec_band_laikas=data(41:57,1);% 3 eksp laikas.
ketv_band_laikas=data(58:77,1);% 4 eksp laikas.
pirm_band_x=data(1:19,2);% 1 eksp x.
antr_band_x=data(20:40,2);% 2 eksp x.
trec_band_x=data(41:57,2);% 3 eksp x.
ketv_band_x=data(58:77,2);% 4 eksp x
pirm_band_s=data(1:19,3);% 1 eksp s.
antr_band_s=data(20:40,3);% 2 eksp s.
trec_band_s=data(41:57,3);% 3 eksp s.
ketv_band_s=data(58:77,3);% 4 eksp s
pirm_band_w=data(1:19,4);% 1 eksp laikas.
antr_band_w=data(20:40,4);% 2 eksp laikas.
trec_band_w=data(41:57,4);% 3 eksp laikas.
ketv_band_w=data(58:77,4);% 4 eksp laikas
pirm_band_px=data(1:19,5);%1eksper px
antr_band_px=data(20:40,5);%2eksper px
trec_band_px=data(41:57,5);%3eksper px
ketv_band_px=data(58:77,5);%4eksper px
%%%pradines reikšmes pagal eksp.
x00=[0.143 0.119 0.116 0.116];
s00=[10.7 10.1 9.6 9.5];
px00=[0 0 0 0];
w00=[4.715 4.93 4.96 5.063];
t_ind0=[5.27 5.47 6.75 8.0];
sf00=[100 100 156 151];
px_start_indukcija=[5 5 10 14];
u_x_1=[data1(1:452,1)];%Pamaitinimo profilis 1
u_y_1=[data1(1:452,2)]/1000;%Pamaitinimo profilis 1
u_x_2=[data1(453:954,1)];%Pamaitinimo profilis 2
u_y_2=[data1(453:954,2)]/1000;%Pamaitinimo profilis 2
u_x_3=[data1(955:1297,1)];%Pamaitinimo profilis 3
u_y_3=[data1(955:1297,2)]/1000;%Pamaitinimo profilis 3
u_x_4=[data1(1298:1710,1)];%Pamaitinimo profilis 4
u_y_4=[data1(1298:1710,2)]/1000;%Pamaitinimo profilis 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Grafikai%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MAE=0;%pradine verte paklaidos
MAPE=0;%pradine verte paklaidos
n=0;
for i=[1:4]
    iexp=i %eksperimento nr.
    x0=x00(1,iexp);
    s0=s00(1,iexp);
    px0=px00(1,iexp);
    w0=w00(1,iexp);
    t_ind=t_ind0(1,iexp);
```



```

sf=sf00(1,iexp);
if iexp==1,
    u_x=u_x_1;
    u_y=u_y_1;
    t1=pirm_band_laikas;
    x1=pirm_band_x;
    s1=pirm_band_s;
    w1=pirm_band_w;
    px1=pirm_band_px;
    px_start_ind=5;
end
if iexp==2,
    u_x=u_x_2;
    u_y=u_y_2;
    t1=antr_band_laikas;
    x1=antr_band_x;
    s1=antr_band_s;
    w1=antr_band_w;
    px1=antr_band_px;
    px_start_ind=5;
end
if iexp==3,
    u_x=u_x_3;
    u_y=u_y_3;
    t1=trec_band_laikas;
    x1=trec_band_x;
    s1=trec_band_s;
    w1=trec_band_w;
    px1=trec_band_px;
    px_start_ind=10;
end
if iexp==4,
    u_x=u_x_4;
    u_y=u_y_4;
    t1=ketv_band_laikas;
    x1=ketv_band_x;
    s1=ketv_band_s;
    w1=ketv_band_w;
    px1=ketv_band_px;
    px_start_ind=14;
end

% Simulation using SIMULINK MODEL
time=t1;
OPTIONS=simset('DstWorkspace', 'current', ...
               'SrcWorkspace', 'current', ...
               'OutputPoints', 'specified', ...
               'SaveFormat', 'Array');
sim('mechanistinis.mdl',time,OPTIONS);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(iexp)
subplot(2,2,1)
plot(time,x_s_px_w(:,1),t1,x1,'r*')
xlabel('Laikas, [h]')
ylabel('Biomasės koncentracija, [g kg-1]')
subplot(2,2,2)
plot(time,x_s_px_w(:,2),t1,s1,'r*')
xlabel('Laikas, [h]')
ylabel('Substrato koncentracija, [g kg-1]')
subplot(2,2,3)
plot(time,x_s_px_w(:,3),t1,px1,'r*')
xlabel('Laikas, [h]')
ylabel('Rekombinuotu baltymu aktyvumas, [U g-1, biomasės]')
subplot(2,2,4)
plot(time,x_s_px_w(:,4),t1,w1,'r*')

```

```

xlabel ('Laikas, [h]')
ylabel ('Kultūros masė, [kg]')
n=n+length(px1);
err0=sum(abs(px1-(x_s_px_w(:,3))));%mae paklaida
MAE=MAE+err0;
err2=nansum(abs((px1-x_s_px_w(:,3))./px1));%mape paklaida
MAPE=MAPE+err2;
end
MAPE=MAPE*(100/n) %isvedama vidutine procentine absoliutine paklaida
MAE=MAE/n

```

## Priedas nr. 2, *Simulink* blokų programiniai kodai

*Simulink* biomasės augimo greičio, substrato suvartojimo greičio, maksimalaus rekombinuotų baltymų aktyvumo, rekombinuotų baltymų augimo greičio bloko programinis kodas:

```

%miu=Biomasės augimo greitis.%qs=Substrato suvartojimo greitis.
%pmax=Maksimalus rekombinuotų baltymų aktyvumas.
%qpx=Rekombinuotų baltymų augimo greitis.
function [miu,qs,qpx,qpx2] = fcn (s,t,t_ind,px,qpxnn)
T_px=1.495;
alfa=0.0495;
ki=93.8;
ks=0.00333;
miu_max=0.737;
m=0.0242;
Y_xs=0.46;
if t<t_ind,
    miu=miu_max*(s/(ks+s))*(ki/(ki+s))*exp(alfa*(37-37));
else
    miu=miu_max*(s/(ks+s))*(ki/(ki+s))*exp(alfa*(32-37));
end
if t<t_ind,
    qpx=0;    qpx2=0;
else
    p_max=(miu*751)/(0.61+miu+(miu*miu)/0.0174);
    qpx=(1./T_px)*(p_max-px);
    qpx2=qpxnn;
end
qs=(1/Y_xs)*miu+m;

```

*Simulink* masės srauto F1 bloko programinis kodas:

```

function Fe = fcn(t,t_ind)
if (t<t_ind),
    Fe=-0.002;
else
    Fe=-0.005;
end

```

*Simulink* mėginių ėmimo dažnio F2 bloko programinis kodas:

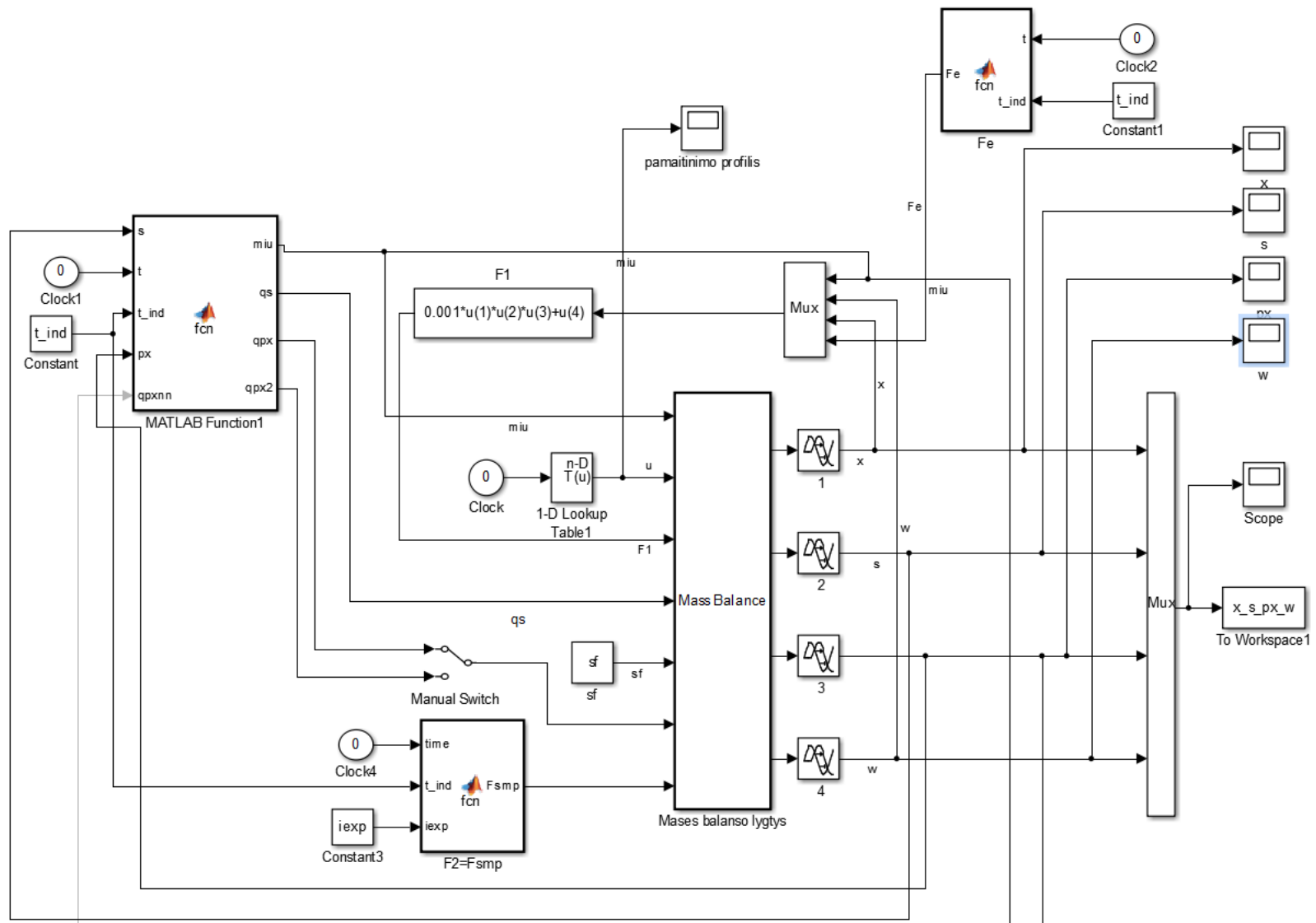
```

function Fsmp = fcn(time, iexp)
if (iexp==3),
    Fsmp=-0.06;
elseif (iexp==4),
    Fsmp=-0.045;
else
    if (time <4),

```

```
Fsmp=0;  
else  
    Fsmp=-0.015;  
end  
end
```

### Priedas nr. 3, Simulink modelis



#### Priedas nr. 4, *Levenberg – Marquart* algoritmo programinis .m kodas

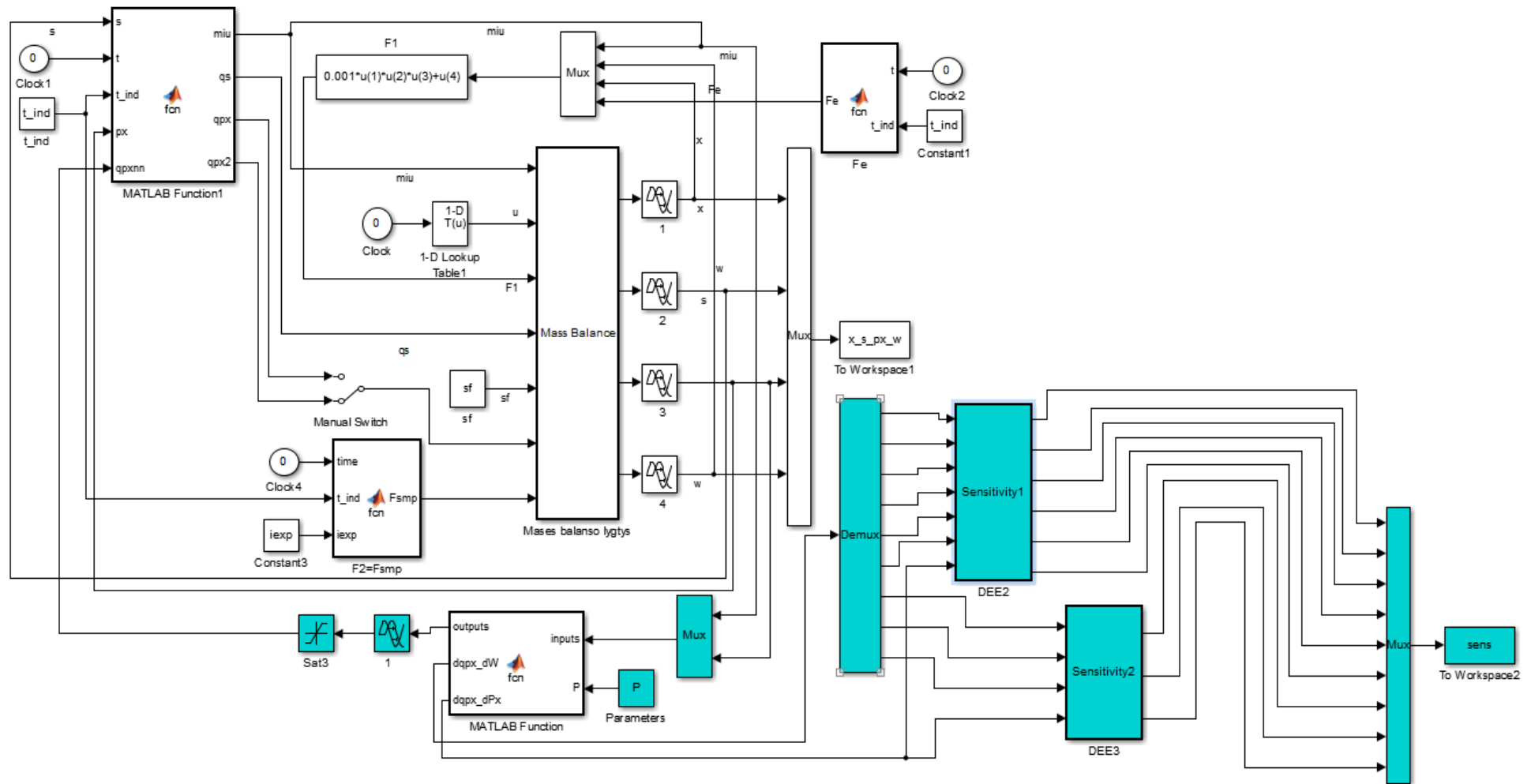
```

clear all, close all, clc
mu=0.005:0.005:0.4;%sugeneruojami biomases augimo greicio duomenys
px=0:1:60;%sugeneruojami rekombinuotu baltymu aktyvumo duomenys
km=751;
kmu=0.61;
kimu=0.0174;
Tpx=1.495;
A=[];
for i=1:length(mu)
    mul=mu(i);
    for j=1:length(px)
        px1=px(j);
        pmax=(mul*km) ./ (kmu+mul+(mul.^2/kimu));%8formule
        qpx(i,j)=(1/Tpx) * (pmax-px1);%apskaiciuojamas baltymu sintezes greitis
        A=[A; mul px1 qpx(i,j)];%matrica apmokymui
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
train_data=A;
%Data normalization
mean_train_data=mean(train_data)
std_train_data=std(train_data)
for i=1:length(std_train_data)
    norm_train_data(:,i)=(train_data(:,i)-...
        mean_train_data(i))./(std_train_data(i));
end
input_var_ind=[1,2];
output_var_ind=[3];
xt=train_data(:, [1 2])'; % norm_train_data(:,input_var_ind)'; %Weight+
Lactose
yt=train_data(:, [3],: )'; %norm_train_data(:,output_var_ind)';%kLA
% The variable 'NetDef' defines the structure of ANN in our case we use
% one input,
% four hidden nodes 'HHHH' with tanh-activation functions
% and aone linear output 'L---', the estimated biomass
% for details of this decription confer the header of the
% Levenber-Marquardt rourtine using "help marg"

number_inputs=length(input_var_ind);
nodes_hidden_layer=2;
nodes_output_layer=length(output_var_ind);
NetDef=['HH----';'L-----'];
%now we initialize the starts weights of ANN, We take random weights W1,W2,
%produced with Matlabs 'rand' function-Dimensions of W1:
%W1=[(nodes_hidden_layer* (inputs+1)); in our case: No_hidden_nodes=3,
%No_input_layer=1 -Dimension of W2: W2[(outpu nodes)*(hidden layer
%nodes+1)]
W1=rand(nodes_hidden_layer,number_inputs+1)-0.5;
W2=rand(nodes_output_layer, nodes_hidden_layer+1)-0.5;
TRPARMS=settrain(settrain,'maxiter',1000);
[W1A,W2A,err,it,lambda]=marq(NetDef,W1,W2,xt,yt,TRPARMS)
%Evaluation
kLanorm=nneval(NetDef,W1A,W2A,xt,yt);
kLA=kLanorm;
save('W1A_W2A.mat','W1A','W2A','NetDef',...
    'mean_train_data','std_train_data');

```

### Priedas nr. 4, Simulink modelis

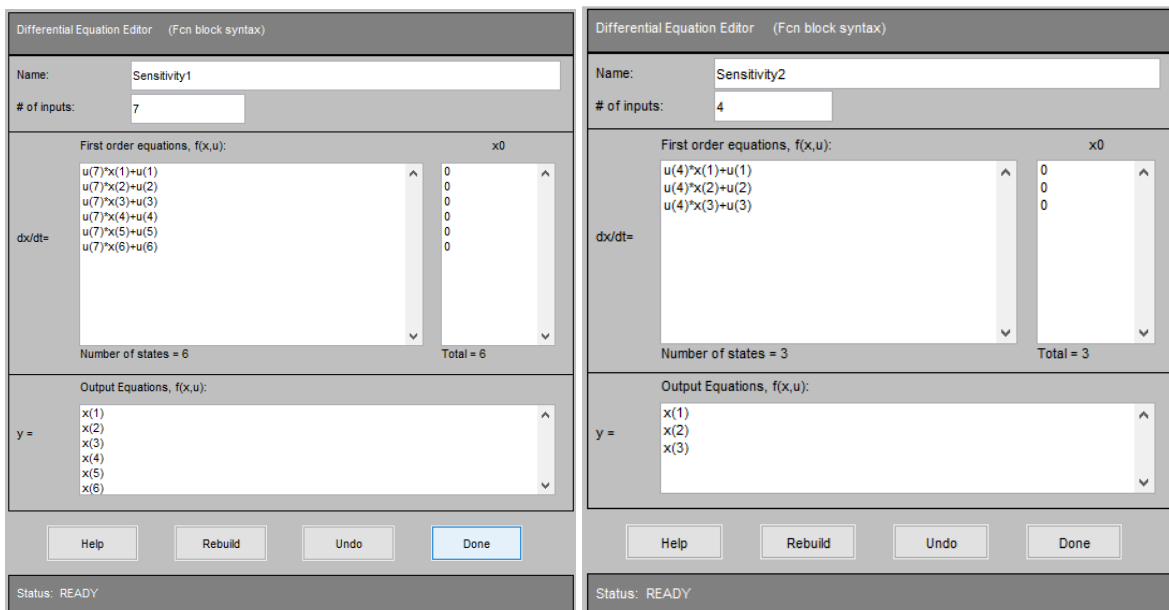


## Priedas nr. 5, Simulink / Matlab function ir DEE bloką programiniai kodas

Function bloko programinis kodas:

```
function [outputs,dqpx_dW,dqpx_dPx] = fcn (inputs, P)
% This block supports an embeddable subset of the MATLAB language.
% See the help menu for details.
inp=[inputs; 1.e0];
w1=[P(1:3)'; P(4:6)'];
w2=[P(7:9)'];
ni=length(inp);
nh=2;
no=1;
h1=w1(1:nh,1:ni)*inp(1:ni);
y1=[tanh(h1); 1.e0];
y2=w2(1:nh+1)*y1;
outputs=y2;
w22=w2(:,1:nh);
y12=y1(1:nh);
%-----
% Estimation of derivatives doutput/dweight and doutput/dinput
% Estimate derivatives of w1
dw1=(w22'.*(1.e0-y12.*y12))*inp';
% and derivatives of w2
%   for i=1:no,
%       for j=1:length(y1),
%           dw2(i,j)=y1(1,j);
%       end
%   end
dw2=y1';
% Reshape derivatives
dww1=reshape(dw1',nh*ni,1);
dww2=reshape(dw2',(nh+1)*no,1);
dqpx_dW=[dww1(1:6); dww2(1:3)];
% Then estimate derivatives doutput/dinput
dout_din=w1'*(w22'.*(1.e0-y12.*y12));
% For sensitivity equation we use only the d(qp)/dp (second row)
dqpx_dPx=dout_din(2); %naudojam (2), nes px yra antras iejimas
```

DEE (diferencialinių jautrumo lygčių) blokai:



## Priedas nr. 6, Matlab .m programinis kodas

```
close all, clear all, clc, warning off
global t1 px1 sens u_x u_y x1 s1 w1 x0 s0 px0 w0 t_ind sf iexp P
global it px_start_ind x00 s00 px00 w00 t_ind0 sf00 px_start_indukcija
global pirm_band_laikas antr_band_laikas trec_band_laikas ketv_band_laikas
global pirm_band_x antr_band_x trec_band_x ketv_band_x
global pirm_band_s antr_band_s trec_band_s ketv_band_s
global pirm_band_w antr_band_w trec_band_w ketv_band_w
global pirm_band_px antr_band_px trec_band_px ketv_band_px
global u_x_1 u_y_1 u_x_2 u_y_2 u_x_3 u_y_3 u_x_4 u_y_4 i
data=load('new.DAT'); %Analiziniai duomenys, 4eksperm.
data1=load('vp1online11.DAT');%Pamaitinimo profilio duomenys.
pirm_band_laikas=data(1:19,1); % 1 eksperm laikas.
antr_band_laikas=data(20:40,1);% 2 eksperm laikas.
trec_band_laikas=data(41:57,1);% 3 eksperm laikas.
ketv_band_laikas=data(58:77,1);% 4 eksperm laikas.
pirm_band_x=data(1:19,2);
antr_band_x=data(20:40,2);
trec_band_x=data(41:57,2);
ketv_band_x=data(58:77,2);
pirm_band_s=data(1:19,3);
antr_band_s=data(20:40,3);
trec_band_s=data(41:57,3);
ketv_band_s=data(58:77,3);
pirm_band_w=data(1:19,4);
antr_band_w=data(20:40,4);
trec_band_w=data(41:57,4);
ketv_band_w=data(58:77,4);
pirm_band_px=data(1:19,5);%1eksper px
antr_band_px=data(20:40,5);%2eksper px
trec_band_px=data(41:57,5);%3eksper px
ketv_band_px=data(58:77,5);%4eksper px
x00=[0.143 0.119 0.116 0.116];
s00=[10.7 10.1 9.6 9.5];
px00=[0 0 0 0];
w00=[4.715 4.93 4.96 5.063];
t_ind0=[5.27 5.47 6.75 8.0];
sf00=[100 100 156 151];
px_start_indukcija=[5 5 10 14];
u_x_1=[data1(1:452,1)];%Pamaitinimo profilis 1
u_y_1=[data1(1:452,2)]/1000;%Pamaitinimo profilis 1
u_x_2=[data1(453:954,1)];%Pamaitinimo profilis 2
u_y_2=[data1(453:954,2)]/1000;%Pamaitinimo profilis 2
u_x_3=[data1(955:1297,1)];%Pamaitinimo profilis 3
u_y_3=[data1(955:1297,2)]/1000;%Pamaitinimo profilis 3
u_x_4=[data1(1298:1710,1)];%Pamaitinimo profilis 4
u_y_4=[data1(1298:1710,2)]/1000;%Pamaitinimo profilis 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Identifikavimas%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% P=randn(9,1); %atsitiktiniai w svoriai
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%gauti pagal levenberg%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
P=[0.1329 0.0039 -0.4151 30.8318 -104.8360 -9.7852 -183.7080 -1.9419
-41.6600]' %25err
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Jautrumo metodu gauti%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% P =[1.7374 0.0136 0.4245 150.6794 -5.2212 -17.5364 -119.0090
15.4638 120.5723]'%122.09err
PAR0=[P]; %pradinis P
YDATA=[pirm_band_px(5:end);antr_band_px(5:end);trec_band_px(10:end);ketv_band
_px(14:end)];
i=[1:4];%eksper nr.
Ntrials=1000;
DesirPerform=1.e-9;
it=1;
```



```

options=optimset(... % Nelder-Mead Parameters
    'MaxFunEvals',Ntrials,... % Maximal # function evaluations
    'TolX',1.e-6,... % Tolerance in parameter values
    'TolFun',abs(DesirPerform),... % Tolerance in function values
    'LargeScale','on',...
    'MaxIter',Ntrials,...
    'Display','iter',...
    'Jacobian','on',...
    'Algorithm','levenberg-marquardt');
[PAR1,dummy,RESID]=lsqcurvefit('antr',PAR0,[],YDATA,[],[],options);
P=PAR1 %identifikuotas P
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Grafikai%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MAE=0;%pradine verte paklaidos
MAPE=0;%pradine verte paklaidos
n=0;
for iexp=i %eksperimento nr.
x0=x00(1,iexp);
s0=s00(1,iexp);
px0=px00(1,iexp);
w0=w00(1,iexp);
t_ind=t_ind0(1,iexp);
sf=sf00(1,iexp);
if iexp==1,
    u_x=u_x_1;
    u_y=u_y_1;
    t1=pirm_band_laikas;
    x1=pirm_band_x;
    s1=pirm_band_s;
    w1=pirm_band_w;
    px1=pirm_band_px;
    px_start_ind=5;
end
if iexp==2,
    u_x=u_x_2;
    u_y=u_y_2;
    t1=antr_band_laikas;
    x1=antr_band_x;
    s1=antr_band_s;
    w1=antr_band_w;
    px1=antr_band_px;
    px_start_ind=5;
end
if iexp==3,
    u_x=u_x_3;
    u_y=u_y_3;
    t1=trec_band_laikas;
    x1=trec_band_x;
    s1=trec_band_s;
    w1=trec_band_w;
    px1=trec_band_px;
    px_start_ind=10;
end
if iexp==4,
    u_x=u_x_4;
    u_y=u_y_4;
    t1=ketv_band_laikas;
    x1=ketv_band_x;
    s1=ketv_band_s;
    w1=ketv_band_w;
    px1=ketv_band_px;
    px_start_ind=14;
end
% Simulation using SIMULINK MODEL
time=t1;

```

```

OPTIONS=simset('DstWorkspace', 'current', ...
              'SrcWorkspace', 'current', ...
              'OutputPoints', 'specified', ...
              'SaveFormat', 'Array');
sim('jautr_ident.mdl',time,OPTIONS);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Grafikai%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(iexp)
subplot(2,2,1)
plot(time,x_s_px_w(:,1),t1,x1,'r*')
xlabel('Laikas, [h]')
ylabel('Biomasės konc., [g kg-1]')
subplot(2,2,2)
plot(time,x_s_px_w(:,2),t1,s1,'r*')
xlabel('Laikas, [h]')
ylabel('Substrato koncentracija, [g kg-1]')
subplot(2,2,3)
plot(time,x_s_px_w(:,3),t1,px1,'r*')
xlabel('Laikas, [h]')
ylabel('Spec. baltymo aktyv., [U g-1, biomasės]')
subplot(2,2,4)
plot(time,x_s_px_w(:,4),t1,w1,'r*')
xlabel('Laikas, [h]')
ylabel('Kultūros masė, [kg]')
figure(5)
if iexp==1
subplot(2,2,1)
plot(time,x_s_px_w(:,3),t1,px1,'r*')
title('1 eksperimentas')
xlabel('Laikas, [h]')
ylabel('p_x, [U g-1, biomasės]')
end
if iexp==2
subplot(2,2,2)
plot(time,x_s_px_w(:,3),t1,px1,'r*')
title('2 eksperimentas')
xlabel('Laikas, [h]')
ylabel('p_x, [U g-1, biomasės]')
end
if iexp==3
subplot(2,2,3)
plot(time,x_s_px_w(:,3),t1,px1,'r*')
title('3 eksperimentas')
xlabel('Laikas, [h]')
ylabel('p_x, [U g-1, biomasės]')
end
if iexp==4
subplot(2,2,4)
plot(time,x_s_px_w(:,3),t1,px1,'r*')
title('4 eksperimentas')
xlabel('Laikas, [h]')
ylabel('p_x, [U g-1, biomasės]')
end
n=n+length(px1); %eksp tasku skaicius
err0=sum(abs(px1-(x_s_px_w(:,3))));%mae paklaida
MAE=MAE+err0;
err2=nansum(abs((px1-x_s_px_w(:,3))./px1));%mape paklaida
MAPE=MAPE+err2;
end
MAPE=MAPE*(100/n) %isvedama vidutine procentine absoliutine paklaida
MAE=MAE/n %isvedama vidutine absoliutine paklaida

```

## Priedas nr. 7, Matlab .m programinis kodas

```
function [Jval,Jsens]=J(pfit,dummy)
global t1 px1 sens u_x u_y x1 s1 w1 x0 s0 px0 w0 t_ind sf iexp P
global it px_start_ind x00 s00 px00 w00 t_ind0 sf00 px_start_indukcija
global pirm_band_laikas antr_band_laikas trec_band_laikas ketv_band_laikas
global pirm_band_x antr_band_x trec_band_x ketv_band_x
global pirm_band_s antr_band_s trec_band_s ketv_band_s
global pirm_band_w antr_band_w trec_band_w ketv_band_w
global pirm_band_px antr_band_px trec_band_px ketv_band_px
global u_x_1 u_y_1 u_x_2 u_y_2 u_x_3 u_y_3 u_x_4 u_y_4 i
    OPTIONS=simset('DstWorkspace', 'current', ...
                  'SrcWorkspace', 'current', ...
                  'OutputPoints','specified', ...
                  'SaveFormat', 'Array');
Jval=[]; Jsens=[];
residuuml=[]; residuum2=[];
for iexp=i; %eksperimento nr.
x0=x00(1,iexp);
s0=s00(1,iexp);
px0=px00(1,iexp);
w0=w00(1,iexp);
t_ind=t_ind0(1,iexp);
sf=sf00(1,iexp);
px_start_ind=px_start_indukcija(1,iexp);
if iexp==1,
    u_x=u_x_1;
    u_y=u_y_1;
    t1=pirm_band_laikas;
    x1=pirm_band_x;
    s1=pirm_band_s;
    w1=pirm_band_w;
    px1=pirm_band_px;
end
if iexp==2,
    u_x=u_x_2;
    u_y=u_y_2;
    t1=antr_band_laikas;
    x1=antr_band_x;
    s1=antr_band_s;
    w1=antr_band_w;
    px1=antr_band_px;
end
if iexp==3,
    u_x=u_x_3;
    u_y=u_y_3;
    t1=trec_band_laikas;
    x1=trec_band_x;
    s1=trec_band_s;
    w1=trec_band_w;
    px1=trec_band_px;
end
if iexp==4,
    u_x=u_x_4;
    u_y=u_y_4;
    t1=ketv_band_laikas;
    x1=ketv_band_x;
    s1=ketv_band_s;
    w1=ketv_band_w;
    px1=ketv_band_px;
end
end
figure (5)
plot(it,P,'o'), hold on
```

```
it=it+1;
P=pfrit;
sim ('jautr_ident.mdl',t1,OPTIONS);
    residuum1=x_s_px_w(px_start_ind:end,3);
    Jval=[Jval; residuum1];
    residuum2=sens(px_start_ind:end,:);
    Jsens=[Jsens; residuum2];
end
```