

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**PANEVĖŽIO TECHNOLOGIJŲ IR VERSLO FAKULTETAS**  
**TECHNOLOGIJŲ KATEDRA**

TVIRTINU

Katedros vedėjas doc. Arūnas Tautkus

(parašas)

(data)

**AUTOMATIZUOTŲ MATAVIMŲ PROGRAMINĖS ĮRANGOS**  
**SUKŪRIMAS IR TYRIMAS**

Baigiamasis magistro projektas  
Valdymo technologijos (kodas 621H66001)

**Vadovas prof. dr. Darius Viržonis**

(parašas)

(data)

**Recenzentas**

(parašas)

(data)

**Projektą atliko Donatas Pelenis**

(parašas)

(data)

**PANEVĖŽYS, 2015**



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Panevėžio technologijų ir verslo fakultetas

---

(Fakultetas)

Donatas Pelenis

---

(Studento vardas, pavardė)

Valdymo technologijos 621H66001

---

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Automatizuotų matavimų programinės įrangos sukūrimas ir tyrimas“

### AKADEMINIO SAŽININGUMO DEKLARACIJA

2015 m. birželio 1 d.

\_\_\_\_\_

Panevėžys

Patvirtinu, kad mano **Donato Pelenio** baigiamasis projektas tema „**Automatizuotų matavimų programinės įrangos sukūrimas ir tyrimas**“ 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)

## BAIGIAMOJO PROJEKTO UŽDUOTIS

**Išduota studentui:** Donatui Peleniui Grupė PME-3

**1. Darbo tema:**

Lietuvių kalba: Automatizuotų matavimų programinės įrangos sukūrimas ir tyrimas

Anglų kalba: The automated measurement software development and research

Patvirtinta 2015 m. vasario mėn. 13 d. dekanu potvarkiu Nr. *ST17-F-13-11*

**2. Darbo tikslas:**

*Sukurti ir iširti programinę įrangą kompiuterizuotam matavimų kompleksui su plačiomis matavimų būdų ir aparatūrinių bei programinių sąsajų parametrizavimo galimybėmis.*

**3. Reikalavimai ir sąlygos:**

*Automatizuotas matavimo kompleksas turi būti valdomas per MATLAB programą. Automatizuotų matavimų programinė įranga turi būti lanksti ir lengvai valdoma.*

**4. Projekto struktūra.** Turinys konkretizuojamas kartu su vadovu, atsižvelgiant į BP pobūdį.

1. Analitinė dalis

*Išanalizuoti dabartines automatizuotų matavimų sistemas ir pateikti automatizuotų matavimų trūkumus ir sprendimus.*

2. Metodologinė dalis

*Parinkti metodus, tinkančius automatizuotų matavimų trūkumų sprendimams. Sukurti automatizuotų matavimų programinės įrangos prototipą.*

3. Tiriamoji dalis

*Ištirti realaus laiko sąveiką ir duomenų perdavimo greitį tarp matavimo prietaisų ir programinės sąsajos, taikant skirtingus virtualius įrenginius;*

*Ištirti sukurtos programinės priemonės veikimo greičius, panaudojant skirtingą matavimo prietaisų kiekį. technines savybes;*

*Ištirti ir palyginti su laiku susijusias technines savybes, nuoseklų ir lygiagrečių užduočių planavimo metodus.*

**5. Ši užduotis yra neatskiriama baigiamojo projekto dalis.**

**6. Projekto pateikimo gynimui kvalifikacinėje komisijoje terminas**

2015-06-01

(data)

Užduotį gavau: Donatas Pelenis

2015-02-02

(studento vardas, pavardė, parašas)

(data)

Vadovas: prof. dr. Darius Viržonis

2015-02-02

(pareigos, vardas, pavardė, parašas)

(data)

Pelenis D., Automatizuotų matavimų programinės įrangos sukūrimas ir tyrimas / vadovas  
prof. dr. D. Viržonis; Kauno technologijos universitetas, Panevėžio technologijų ir verslo  
fakultetas, Technologijų katedra.

Panevėžys, 2015. 50 psl.

## SANTRAUKA

Magistro darbo tikslas - sukurti ir iširti programinę įrangą, skirtą automatizuotiems matavimams. Tyrimų objektas - sukurta automatizuotų matavimų programinė įranga, derinanti ir sinchronizuojanti į sistemą sujungtų matavimo prietaisų darbą. Programinės įrangos algoritmai organizuoja informacijos mainus tarp automatizuoto matavimo scenarijaus vykdančio programos, išpildyto MATLAB sintakse ir matavimo prietaisų, automatiškai parametrizuoja ir nustato prietaisus, lygiagrečiu ir/arba nuosekliu būdu planuoja matavimo prietaisų užduotis, dirbant realiuoju laiku. Atlikti eksperimentiniai programinės įrangos techninių savybių tyrimai:

- lygiagretaus ir nuoseklaus užduočių planavimo palyginimas;
- matavimo ciklo trukmės priklausomybės nuo į matavimo procesą sujungtų prietaisų kiekio suradimas lygiagretaus ir nuoseklaus užduočių planavimo atveju.

Programinė įranga išbandyta ir taikyta automatizuotame matavimo procese, tiriant talpinio mikromontuojamo ultragarsinio jutiklio darbo tašką. Apibendrinant atliktų tyrimų ir eksperimentinio taikymo rezultatus, galima teigti, kad sukurta programinė įranga yra parankus, greitai įsisavinamas kompiuterizuotos matavimo sistemos darbo organizavimo įrankis, užtikrinantis stabilų ir prognozuojamą sistemos darbą realiame laike.

**Raktiniai žodžiai:** Automatizuoti matavimai, programinė įranga, nuoseklusis ir lygiagretusis užduočių planavimas, sąsaja, realusis laikas.

Pelenis D., The automated measurement software development and research / guide prof. PhD. D. Viržonis; Kaunas University of Technology, Panevėžys Faculty of Technology and Business, Department of Technologies.

Panevėžys, 2015. 50 psl.

## SUMMARY

The main aim of the Master's work is to create and explore software for automated measurements system. The object of research - created automated measurement software, which optimizes and adjusts the operation of the measuring system. Algorithms of the software communicate with MATLAB scenario, which carries the essential measurement route schedules the measuring instruments, using sequential and parallel task scheduling methods. During the experiments sequential and parallel task scheduling was studied in terms of the system temporal performance efficiency. Software was also tested and applied in automated measurement process of determining the operation point of the capacitive micromachined ultrasound sensor. Results suggest that using developed software provides time saving opportunities when developing the measurement algorithm code. Also it reliably copes with the measurement tasks and provides robust operations of the system in the real time.

**Keywords:** Automated measurements, software, sequential and parallel task scheduling, interface, real time.

# TURINYS

<b>SANTRUPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS.....</b>	<b>8</b>
<b>ĮVADAS.....</b>	<b>9</b>
<b>1. ANALITINĖ DALIS .....</b>	<b>11</b>
1.1 Automatizuoto matavimo algoritmo analizė .....	12
1.2 Matavimo sistemų programinė įranga.....	13
1.2.1 Realaus laiko komunikavimas tarp LabVIEW ir MATLAB .....	16
1.3 Komunikacinės valdymo sąsajos.....	16
1.3.1 Komunikacinės architektūros.....	16
1.3.2 Virtualių įrenginių programinė įranga (VISA) .....	17
1.4 MATLAB ir LabVIEW platformų komunikavimas.....	19
1.4.1 MATLAB scenarijaus panaudojimas LabVIEW terpėje .....	19
1.4.2 TCP/IP API protokolinis valdymas.....	20
1.4.3 Dokumentinis duomenų keitimasis .....	21
1.5 Nuoseklusis ir lygiagretusis automatizuotų matavimų metodai .....	22
1.6 Kuriamos programinės įrangos priemonių taikymas ir parinkimas .....	23
1.7 Skyriaus išvados.....	24
<b>2. METODOLOGINĖ DALIS.....</b>	<b>25</b>
2.1 Pagrindiniai automatizuotų matavimų programinės įrangos algoritmai.....	25
2.2 Automatizuotų matavimų programinės įrangos valdymas .....	29
2.3 Matavimo prietaisų užduočių ir komutacijos laikų nustatymas .....	31
2.4 Programinės įrangos teorinis modelis .....	32
2.5 Ultragarso jutiklio matavimai naudojant automatizuotą matavimų programinę įrangą .....	36
2.6 Skyriaus išvados.....	38
<b>3. TIRIAMOJI DALIS .....</b>	<b>39</b>
3.1 Eksperimentinių tyrimų priemonės .....	39
3.2 Realaus laiko duomenų perdavimas tarp MATLAB ir LabVIEW tyrimas.....	39

3.3 Virtualiųjų prietaisų (VISA) tyrimas.....	41
3.4 Automatizuotų matavimų programinės įrangos greičio tyrimas .....	43
3.4.1 „MATLAB“ ir automatizuotų matavimų programinės įrangos greičio tyrimas .....	43
3.4.2 Automatizuotų matavimų programinės įrangos nuosekliojo ir lygiagrečiojo užduočių planavimo metodo tyrimas .....	44
3.5 Ultragarso jutiklio darbo taško paieška naudojant AMPį.....	45
3.6 Eksperimentų rezultatų analizė .....	48
3.7 Skyriaus išvados.....	51
<b>IŠVADOS .....</b>	<b>52</b>
<b>LITERATŪROS SĄRAŠAS.....</b>	<b>54</b>
<b>PRIEDAI .....</b>	<b>56</b>

## SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS

**VISA** – virtualūs prietaisai;

**AMPi** – Automatizuotų matavimų programinė įranga;

**LabVIEW** – „National Instruments“ grafinė programavimo platforma;

**MATLAB** – daugiaplatformė MathWorks programinė įranga, skirta įvairių mokslo šakų problemoms spręsti, ypač matematinėms;

**CMUT** – mikromontuojamas ultragarsinis keitiklis;

**DAQ** – Duomenų surinkimo plokštė;

**TCP/IP** - standartinis duomenų perdavimo protokolų rinkinys, kurio pagrindu veikia internetas bei daugelis privačių komercinių tinklų. Svarbiausi šio protokolo komponentai

– TCP ir IP protokolai;

**DLL** – Dinaminė biblioteka.



## IVADAS

Moksliniuose ir technologiniuose tyrimuose yra naudojamos kompiuterizuotos matavimų sistemos, sudarytos iš keleto matavimo prietaisų. Į sistemą sujungti matavimo prietaisai, daugeliu atvejų turi skirtingas aparatūrines ir programines sąsajas su kompiuteriu, dėl ko labai komplikuojasi programinės įrangos tokiai sistemai kūrimas. Pastebima, kad tyrimo eigoje gali prireikti operatyviai keisti matavimo duomenų, rezultatų gavimo ir pateikimo būdus bei scenarijus, daugelį kartų perstatyti matavimo sistemoje sujungtų prietaisų nustatymus. Pavyzdžiui, matuojant talpinio mikromontuojamo ultragarso keitiklio (CMUT) rezonansinio dažnio priklausomybę nuo įtampos dydžio, kiekvienam priklausomybės taškui surasti reikia atlikti mažiausiai dviejų skirtingų prietaisų - nuolatinės įtampos šaltinio ir grandinių analizatoriaus parametrų perstatymus. Priklausomybei surasti paprastai reikia kelių dešimčių taškų. Dažniausiai tokio tipo matavimams yra kuriamos automatizuotos programinės sąsajos. Toks būdas leidžia mokslininkui lengviau susidoroti su testuojamo prietaiso charakteristikų aptikimu. Matavimo prietaisai būna multifunkciniai, todėl tuos pačius įrenginius galima panaudoti ir kitų kuriamų įtaisų ar procesų fizikinių savybių tyrimams. Programinės sąsajos kodo rašymas įprastai užtrunka nuo kelių dienų iki kelių savaitių, neįskaičiuojant laiko, kurio reikia pačiam programavimo procesui įsisavinti, matavimo prietaisų programinio valdymo sintaksei bei kitiems techniniams ypatumams išsiaiškinti. Beveik visi šiuolaikiniai matavimo prietaisai turi valdymo – komunikacinius ryšio kanalus, todėl atsiranda galimybė naudojant komunikacines sąsajas įvairiai konfigūruoti matavimo sistemą konkrečiam matavimui atlikti. Naudojantis populiariomis šiuolaikinėmis daugiaplatformėmis modeliavimo bei prietaisų valdymo programomis – „MATLAB“ ir „LabVIEW“, visuomet yra galimybė sukurti patogią automatizuotų matavimų sąsają. Tačiau išlieka sąsajos kodo sukūrimo trukmės ir sukurto programinio produkto lankstumo problema. Dauguma tyrėjų geba greitai ir kokybiškai kurti modeliavimo ar eksperimento scenarijus, naudojant „MATLAB“ sintaksę. Tačiau realaus laiko kodo kūrimas, naudojant „MATLAB“ galimybes, reikalauja kitų žinių ir gebėjimų nei kad turi eilinis tyrėjas. Tuo pačiu metu „LabVIEW“ paketas pasižymi puikiai išvystyta, žymiai lankstesne nei „MATLAB“ virtualių prietaisų aplikacijų sistema (VISA). Todėl pagrindinis šio darbo uždavinys yra taip sukombinuoti abiejų populiarių platformų galimybes, kad eilinis matavimo sistemos vartotojas galėtų minimaliomis laiko sąnaudomis keisti tiek matavimo sistemos programinio aptarnavimo rutinas, tiek matavimo scenarijų ar matavimo rezultatų pateikimą.

**Tyrimo metodai:** mokslinės literatūros analizė, eksperimentiniai tyrimai, matematinis modeliavimas (MATLAB programa).

**Tyrimo objektas:** Automatizuotų matavimų programinė įranga.

**Tyrimo tikslas:** Sukurti ir ištirti programinę įrangą kompiuterizuotam matavimų kompleksui su plačiomis matavimų būdų ir aparatūrinių bei programinių sąsajų parametrizavimo galimybėmis.

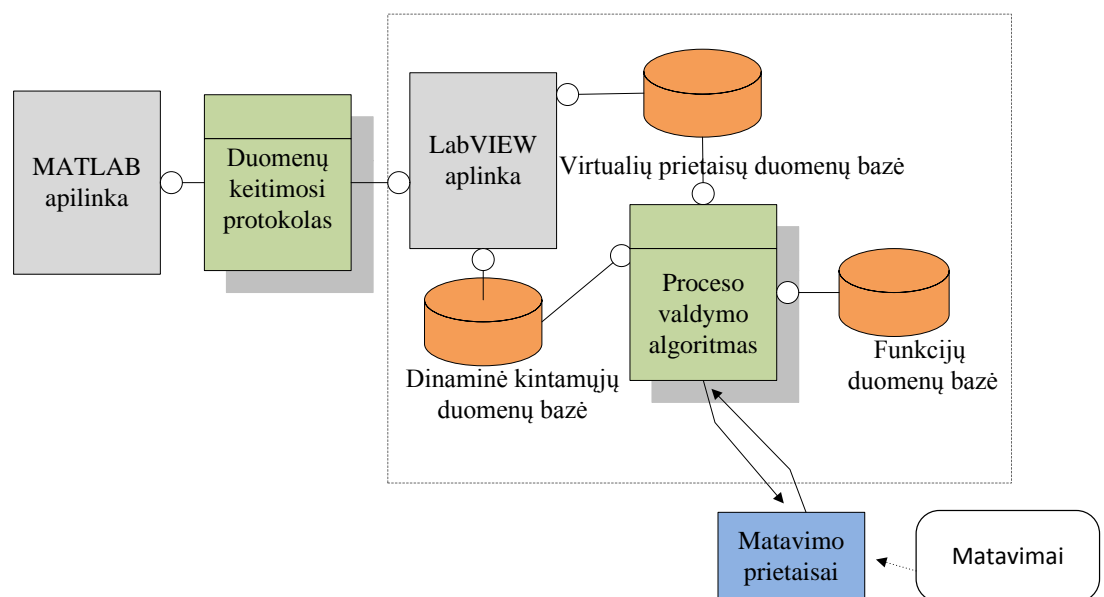
Darbo tikslui pasiekti keliami tokie **tyrimo uždaviniai:**

1. Išanalizuoti dabartines automatizuotų matavimo sistemas, pateikti automatizuotų matavimų trūkumus ir sprendimus;
2. Parinkti metodus, tinkančius automatizuotų matavimų trūkumų sprendimams;
3. Sukurti programinės įrangos prototipą;
4. Eksperimentų pagalba ištirti programinės įrangos parametrizavimo galimybių ir faktinių techninių savybių atitikimą reikalavimams;
5. Ištirti ir palyginti su laiku susijusias technines savybes, nuoseklųjį ir lygiagretųjį užduočių planavimo metodus.

# 1. ANALITINĖ DALIS

Viena populiariausių įvairiuose tyrimuose naudojamų programinių platformų yra „MATLAB“ - daugiafunkcinė, mokslo sričiai skirta programinė įranga, turinti savą virtualių įrenginių valdymo sistemą (VISA). Ši programinė sistema skirta sukommunikuoti matavimo prietaisus su „MATLAB“ scenarijais ir programomis. Realaus laiko aspektas matavimuose yra labai svarbus ir dažniausiai reiškia, kad matavimo duomenų persiuntimas turi užimti kuo mažesnę laiko intervalą. Nors „MATLAB“ programa turi realaus laiko sistemų programavimo įrankius, tačiau jų naudojimas nėra paprastas eiliniam tyrėjui. Taip pat žinoma, kad dinaminį matavimą atveju už „MATLAB“ programą pranašesnis yra „National Instruments“ sukurtas laboratorinių bei pramoninių matavimų instrumentas – „LabVIEW“ programa. Tačiau kuriant matavimo algoritmą „LabVIEW“ platformoje sunaudojamos didelės darbo ir laiko sąnaudos. Tuo tarpu, kuriant „MATLAB“ scenarijus, aukšta programuotojo kompetencija nėra būtina [1].

Kombinuojant „MATLAB“ ir „LabVIEW“ platformų privalumus atsiranda galimybė sukurti šio darbo uždavinius atitinkančią automatizuotą matavimų valdymo programinę įrangą (žr. 1 pav.). Tačiau skirtingų aplinkų sukombinavimui iškyla bent vieno bendro duomenų keitimosi protokolo poreikis. Protokolas turi veikti taip, kad duomenų pernešimas tarp programinių modulių nebūtų papildomas laiko trikdys matavimų metu. Todėl atliekant literatūros analizę buvo koncentruojamasi į duomenų keitimosi tarp „MATLAB“ ir „LabVIEW“ programų organizavimo metodus.



1 pav. Galimas automatizuotų matavimų sąsajos modelis

## 1.1 Automatizuoto matavimo algoritmo analizė

Tyrinėjant automatizuotų matavimų technologiją pastebėta, kad vieno ar kelių matavimo prietaisų neužtenka norint iširti tiriamojo objekto savybes. Nuodugniam objekto tyrimui naudojami keli matavimo prietaisai, kurių sąveikaujančios funkcijos leidžia atlikti eksperimentus su tiriamaisiais objektais. Objekto tyrimui reikalingą parašyti matavimo algoritmą, kuris valdo kelis matavimo prietaisus, bet matavimo algoritmo sukūrimui yra reikalingi nemaži laiko resursai. Įprastai algoritmo sukūrimas yra sudėtingas ir lėtas procesas dėl komunikacinių sąsajų ir matavimo prietaisų suderinimo. Geroji matavimo algoritmo kūrimo pusė yra ta, kad vieną kartą suderinus matavimo prietaisą (jei nesikeičia algoritmo kūrimo aplinka) galima naudoti prietaiso valdymo duomenis kitiems kuriamiems algoritmams. Galima pabrėžti ir tai, kad vienas iš automatizuotų matavimų programinės įrangos kūrimo pagrindimų yra matavimo prietaisų bibliotekos sudarymas. Sudarius kelių matavimo prietaisų valdymo funkcijų bibliotekas, palengvėja matavimo algoritmo kūrimas, tačiau atsiranda problema dėl jų suderinamumo. Derinant kelis matavimų algoritmus reikia suderinti gautųjų ir valdomųjų duomenų tipus (sintakse). Tokiam procesui įgyvendinti yra naudojamas duomenų suderinamumas, kurį privaloma standartizuoti. Standartizuotos funkcijų panaudojimas leidžia greičiau ir kokybiškiau sukurti matavimo algoritmą. Standartizuoti galima ne tik prietaiso valdymo funkcijas, bet ir matavimo procesą [2].

Automatizuoti matavimai skirstomi į: ciklinius matavimus, periodinius matavimus, matavimus pagal charakteristikų pokyčius. Tokiems matavimams reikalingi atskiri algoritmai, kurie valdytų visą matavimo procesą. Matavimo proceso valdymui yra naudojamos nuosekliosios ir lygiagrečiosios matavimo prietaisų valdymo architektūros. Nuoseklią matavimo prietaisų valdymo architektūrą vykdo matavimo scenarijus nuosekliai, todėl čia matavimai užtrunka ilgai. Norint tilpti į realaus laiko diapazonus, reikalingas lygiagretusis valdymas. Lygiagretusis matavimo prietaisų valdymas skiriasi nuo nuoseklaus tuo, kad valdomieji prietaisai yra valdomi tada, kai jie neatlieka jokio matavimo ar duomenų perdavimo. Tada matavimo prietaisai gali būti valdomi lygiagrečiai. Šį matavimo būdą galima standartizuoti tuo atveju, kai yra naudojamas atskiras programinis algoritmas, kuris atsakingas už lygiagretųjį matavimo valdymą [3].

Programinės įrangos, automatizuotiems matavimams, reikalingumas pagrindžiamas standartizuojant matavimų metodus, duomenų tipų algoritmus, matavimo prietaisų valdymą. Programinei įrangai lankstumo duotų suderinamumas su kitomis programinėmis įrangomis: „MATLAB“, „Virtual programming laboratory“, „Agilent VEE“ ir kitomis populiariomis programomis.

## 1.2 Matavimo sistemų programinė įranga

Matavimas yra procedūra, kuria eksperimentiniu būdu yra randama fizikinio dydžio vertė, lyginant ją su matu. Matas atkuria matuojamojo dydžio vienetą. Matavimai yra skirstomi į rankinius ir automatinius. Rankiniuose matavimuose dalyvauja žmogus, kuris naudodamasis visomis techninėmis galimybėmis, valdo matavimų operacijų eigą. Tuo tarpu automatiniuose matavimuose žmogaus funkcijos apsiriboja matavimo scenarijaus sukūrimu, o visi matavimai atliekami automatizuotai.

Matavimus dažniausiai tenka atlikti skirtinguose matuojamų kintamųjų erdvės taškuose, realiame laike persijungiant nuo vieno taško prie kito. Tai reiškia, kad visi matavimo prietaisai turi veikti nepriklausomai vienas nuo kito, automatiškai valdomi kompiuterio. Naudojant nuoseklias ir lygiagrečias komunikacijos sąsajas yra galimybė sujungti asmeninį kompiuterį su visais matavimo prietaisais. Asmeninis kompiuteris tokioje tipologijoje atlieka vedlio vaidmenį, o visi kiti matavimo prietaisai yra valdomieji įrenginiai. Iš kompiuterio pateikus komandinę informaciją prietaisams, matavimo prietaisai įvykdo užduotą komandą su grįžtamuoju ryšiu arba be jo. Grįžtamajame ryšyje gali būti pirminiai matavimų duomenys arba komandos įvykdymo atsakymas. [4].

Matavimo sistemų programinė įranga skirstoma į tris pagrindines grupes:

1. Duomenų surinkimo programiniai paketai;
2. Surinktų duomenų analizavimo programos;
3. Duomenų atvaizdavimas (vizualizavimas).

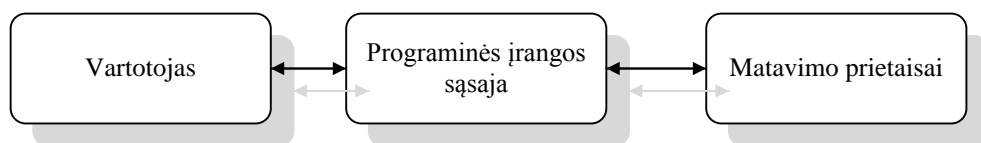
Toks suskirstymas yra pakankamai sąlygiškas, nes įvairių programinių paketų galimybės persidengia. Kai kurie iš jų leidžia vykdyti ir gretimų grupių funkcijas, pavyzdžiui, surinkti ir vizualizuoti matavimo duomenis. Nepaisant to, tokia klasifikacija leidžia lengviau palyginti programų paketų taikymo sritis ir pranašumus.

Kiekviena programinė įranga turi savų privalymų ir trūkumų. Vienos programos gali būti patogesnės vartotojui, kitos - turėti didelį aparatūrinį bei programinį lankstumą. Pagal šiuos aspektus programinė įranga suskirstoma į tokias klases:

1. Vartotojui adaptuota programinė įranga;
2. Kalbų sąsajos programinė įranga;
3. Praplėtimo programinė įranga;
4. Virtualių prietaisų programinė įranga.

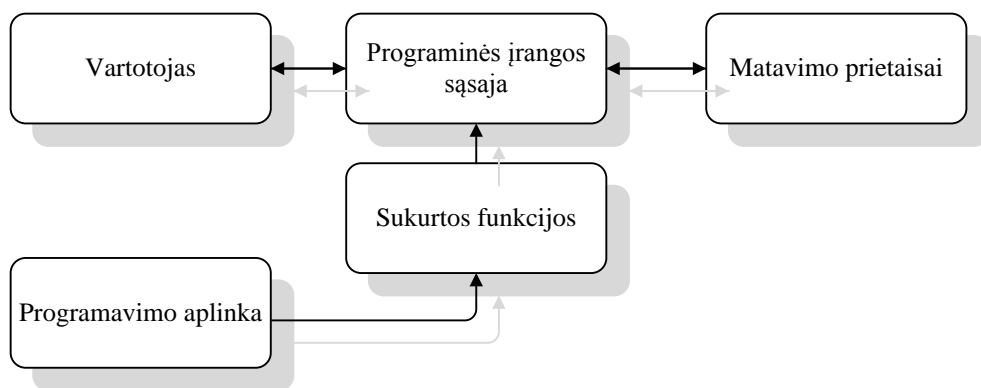
Vartotojui adaptuota programinė įranga yra skirta vienam ar keliems matavimo uždaviniais spręsti (žr. 2 pav.). Ši įranga būna pilnai parengta vartotojo poreikiams, todėl nieko papildomai programuoti nereikia. Toks matavimo sistemos darbo organizavimo būdas reikalauja

minimalių laiko sąnaudų, didina sistemos prieinamumą. Tačiau šis būdas nėra lankstus, nėra galimybės išplėsti matavimo galimybių nei pritaikyti programą kitiems matavimams [5].



2 pav. Adaptuotos programinės įrangos automatizuotų matavimų modelis

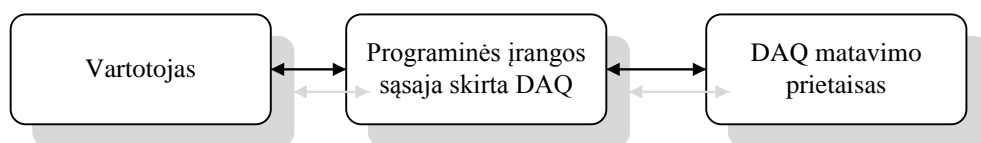
Kalbų sąsajos programinė įranga leidžia valdyti matavimus iškviečiant tam tikras programavimo kalbos funkcijas (žr. 3 pav.). Šiuo būdu vartotojas pats turi galimybę sukurti scenarijus reikiamam matavimui. Naudojant kalbos sąsajos programas, programavimas tampa kur kas paprastesnis, jei programavimo kalboje yra specialiųjų funkcijų biblioteka. Toks būdas leidžia vartotojui pačiam kurti programinį paketą, skirtą matavimams. Tačiau tokiam būdui yra reikalinga aukšta programuotojo kompetencija, kodo sukūrimo laikas yra ilgas.



3 pav. Kalbų sąsajos programinės įrangos automatizuotų matavimų modelis

Praplėtimo programinė įranga gali praplėsti kitų programinių paketų sąsajas. Praplėtimas gali būti pritaikomas matavimų duomenims surinkti. Tam dažniausiai naudojami programų paketai „MS Excel“ arba „Agilent VEE“. Gautus rezultatus galima atvaizduoti keliais būdais: lentelėse, diagramose ir t. t.

Didžiausias kuriamo kodo lankstumas pasiekiamas naudojant virtualių prietaisų sistemą (žr. 4 pav.). Virtualūs matavimo prietaisai papildomai valdo į kompiuterį įdiegtą universalią matavimų bei duomenų surinkimo plokštę (data acquisition hardware - DAQ). Šis būdas unikalus tuo, kad programiniu paketu galima imituoti matavimo prietaisą, panaudojant DAQ plokštę. Programiniu paketu taip pat galima tirti dažnines charakteristikas, įtampas, sroves ir kt.



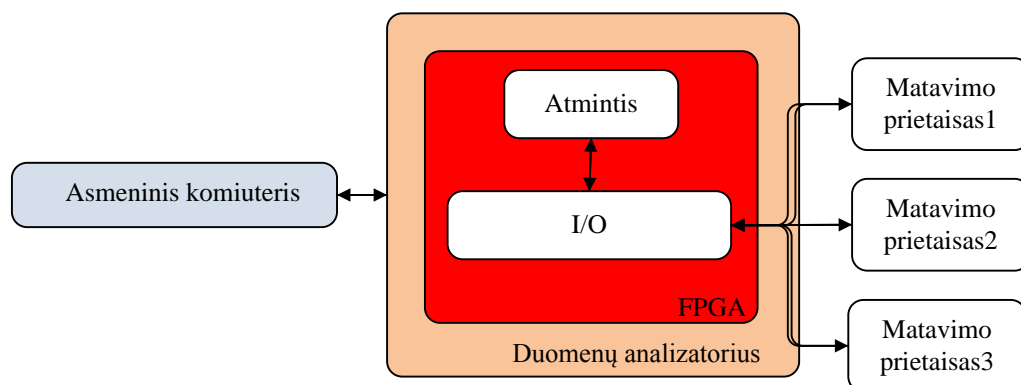
4 pav. Universalios matavimo plokštės (DAQ) taikymas automatizuotiems matavimams

Vienas asmeninis kompiuteris su keliomis DAQ plokštėmis gali valdyti, simuliuoti įvairiausių matavimų prietaisų funkcijas. Virtualių prietaisų būdu valdoma matavimų sistema gali tapti vienu iš pagrindinių matavimo įrenginių laboratorijose, galinčiu pakeisti voltmetrus, oscilografus. Programinis virtualių prietaisų aptarnavimas taip pat nėra komplikotas. Tačiau sudėtingesniems matavimams, pavyzdžiui norint pamatuoti kompleksinio impedanso dažnių spektrą, universalios DAQ tipo plokštės neužtenka, į sistemą reikia jungti didesnes galimybes turinčius matavimo prietaisus [6].

Kompanija National Instruments (NI) turi sukūrusi laboratorinių instrumentų valdymo programinę platformą LabVIEW (Laboratory Virtual Instrument Engineering Workbench). LabVIEW žinoma kaip programavimo kalba, su kuria galima kurti programines sąsajas. Ši platforma turi plačią komunikacijų funkcijų biblioteką, skirtą matavimo prietaisų sujungimui su asmeninio kompiuterio programomis.

Mokslinių, pramoninių ir medicininių matavimų kontrolės sistemos turi sudėtingus algoritmus, kurie atsakingi už savalaikius ir patikimus matavimo rezultatus. Visas automatizuoto matavimo sistemas galima priskirti realaus laiko sistemoms (RLS).

Dažniausiai norint dirbti realiame laike yra reikalingos aparatūrinės (hardware) ir programinės (software) dalies kombinacinės sistemos. Realaus laiko aparatūrai aptarnauti paprastai yra naudojama lygiagreti užduočių planavimo schema, kurią naudojant yra paprasčiau užtikrinti RLS keliamus reikalavimus, nei naudojant nuosekliąją kietąją užduočių planavimo schemą. Iš kitos pusės, nuosekioji užduočių planavimo schema pranašesnė už lygiagrečiąją dėl savo paprastumo ir pigumo. Kompromisas tarp nuoseklių sistemų lankstumo ir lygiagrečių sistemų greičio yra lengvai modifikuojama integrinė grandinė – FPGA (programuojama loginių elementų matrica). FPGA - tai aparatūrinė dalis, kurią dažniausiai naudoja matavimo prietaisai, skirti matuoti realiuoju laiku. Tačiau net ir matavimo prietaisais aparatūrinės dalies pakeisti negalima, nes matavimo prietaisai dažniausiai būna standartizuoti ir nenaudoja FPGA (žr. 5 pav.). Todėl duomenų keitimąsi galima padaryti greitesniu ir efektyvesniu tobulinant tik programinę dalį [7].



5 pav. FPGA struktūra

Aparatinėse RLS pavėluotas atsakas, net jeigu jis ir yra teisingas, yra nepriimtinas. Programinėse RLS, kur informacija ir įvykiai gali būti valdomi pakankamai greitai, apribojimai nėra apibrėžti. Programinės RLS turi sumažintus apribojimus vėlinimui ir veikia pakankamai greitai. Kai kurie realaus laiko procesai gali būti įvykdomi nuo kelių iki keliolikos mikrosekundžių. Taigi, realaus laiko sąvoka yra ne absoliutus, o santykinis atributas, priklausantis nuo konkrečios sistemos [8].

### 1.2.1 Realaus laiko komunikavimas tarp LabVIEW ir MATLAB

„LabVIEW“ programos sąsaja turi galimybę analizuoti, apdoroti ir dalintis duomenimis su kitomis programomis. Programoje naudojami virtualūs įrankiai, kurie suteikia galimybę susijungti su fizinėmis priemonėmis kaip osciloskopu, multimetru ir t. t. Kiekvienas programavimo elementas manipuliuoja įrankio įėjimus iš vartotojo sąsajos ar kitų duomenų šaltinių bei informacijos atvaizdavimus ar duomenų perkėlimus kitoms apdorojančioms funkcijoms [6].

„MATLAB“ yra aukšto lygio programavimo kalba, skirta techniniams skaičiavimams atlikti. Į programą integruotos skaičiavimo, vizualizavimo ir programavimo aplinkos, kurių problemos ir sprendimai išreiškiami matematiniais žymėjimais. „MATLAB“ programa turi galimybę dirbti su MathWorks Real-Time studija. Realaus laiko studija generuoja ir paleidžia atvirą „C“ kodo kūrimo ir testavimo algoritmo modelį. Gauti kodai gali būti naudojami sukommunikuoti „MATLAB“ su „LabVIEW“ programomis. „MATLAB“ terpei trūksta sąsajos su aparatūrine matavimo sistemos dalimi lankstumo. Savo ruožtu, „LabVIEW“ modeliavimo sąsajos įrankiai gali būti suderinami su realaus laiko programine įranga į „MATLAB“. [8, 9]

Modeliavimo sąsaja suteikia ryšį tarp „Simulink“ ir „LabVIEW“. Modeliavimo sąsaja taip pat leidžia kontroliuoti sumaketuotą „Simulink“ modelį naudojant „LabVIEW“ branduolį bei stebėti gaunamus rezultatus ir keisti modelio parametrus veikiant modeliui. Šis metodas suteikia galimybę realiame laike kontroliuoti duomenis.

## 1.3 Komunikacinės valdymo sąsajos

### 1.3.1 Komunikacinės architektūros

Norint valdyti matavimo prietaisą yra būtina, kad įrenginys turėtų komunikavimo sąsaja. Sąsaja duoda galimybę prijungti valdomąjį prietaisą arba asmeninį kompiuterį. Matavimų prietaisai su asmeniniu kompiuteriu, dažniausiai yra jungiami keliais būdais: lygiagrečiai ir nuosekliai. Nuoseklioms sąsajoms sujungti yra naudojamos RS-232, RS-422, USB ir kitos nuosekliosios sąsajos. Lygiagretiems jungimams dažniausiai naudojama IEEE488 (GPIB, HPIB)



šąsąja. Priklausomai nuo šąsąjos tipo yra taikomos matavimų, prietaisų ir sujungimų architektūros. Jos gali būti tokios: lygiagrečioji, žvaigždinė ir kilpinė [9].

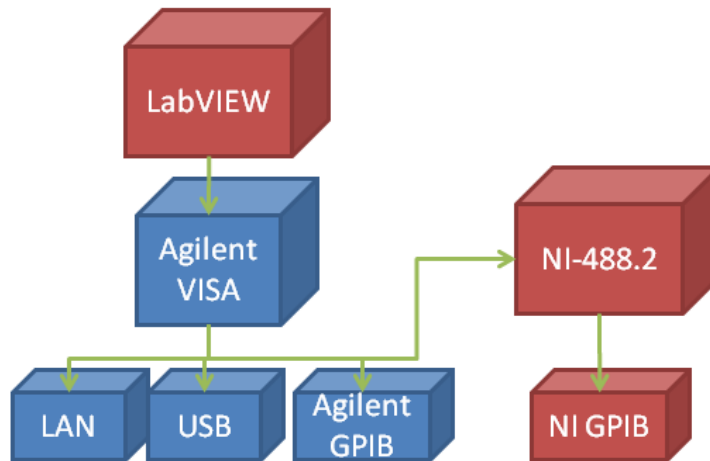
Viena iš populiariausių architektūrų yra žvaigždinė. Šiuo būdu matavimo prietaisai su asmeniniu kompiuteriu yra jungiami tiesiogiai vienas su kitu. Kiekvienam prietaisui sudaroma atskira ryšio linija. Šioje pozicijoje asmeninis kompiuteris atlieka vykdytojo vaidmenį. Kiekvienas prietaisas su kompiuteriu yra valdomas atskirai. Žvaigždinės architektūros privalumas yra toks, kad nutrūkus ryšio linijai tarp vieno prietaiso ir kompiuterio, sisteminiai matavimai vis tiek gali vykti, nepilnai funkcionuojant prietaisui, su kuriuo yra sutrikęs ryšys.

Lygiagrečios architektūros unikalios tuo, kad viena magistrale dirba visi įrenginiai. Pavyzdžiui, IEEE488 (GPIB, HPIB) šąsąja. Jos privalumas toks, kad nereikia daug laidų jungti prie asmeninio kompiuterio, bet įmatavimo prietaisams valdyti yra reikalingas papildomas adaptuojantis instrumentas iš IEEE488 standarto į asmeniniam kompiuteriui tinkantį standartą.

### 1.3.2 Virtualių įrenginių programinė įranga (VISA)

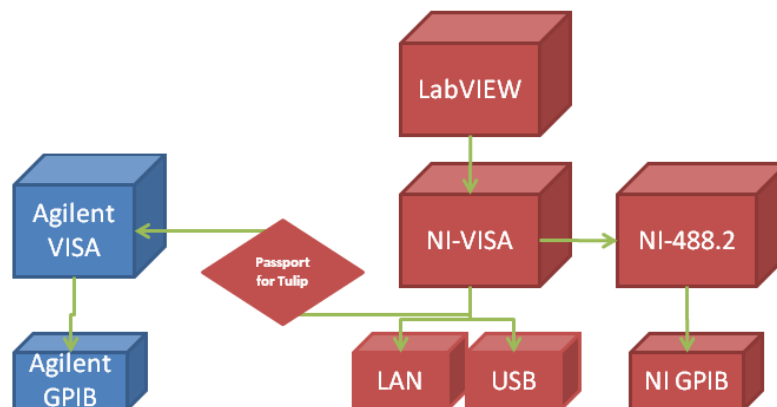
Viena iš populiariausių nuoseklaus tipo aparatūrinių šąsąjų yra USB (Universal Serial Bus). Tokio tipo jungtį turi kiekvienas asmeninis kompiuteris. USB priklauso nuosekliųjų šąsąjų valdymo grupei. Dar viena populiari jungtis, kurią naudoja kompiuteris, tai LPT jungtis. LPT priklauso lygiagrečių šąsąjų grupei. Kai kurie matavo įrenginiai turi tinklo adapterius. Šį prievadą turi ir kompiuteriai. Taigi, svarbu žinoti, kad programinė įranga turi turėti lygiagrečias ir nuoseklias tinklo šąsąjos valdymo galimybes. Pasaulyje populiariausios naudojamos instrumentinės valdymo programos yra „LabVIEW“, „MATLAB“, „Agilent VEE“. Kiekviena iš šių programų turi privalumų ir trūkumų.

„Agilent VEE“ - tai grafinė programinės įrangos kūrimo aplinka, kuri skirta automatizuotiems matavimams bei duomenų analizėms. „Agilent VEE“ programiniu paketu galima valdyti visus „Agilent“ matavimo prietaisus (žr. 6 pav.). VEE naudoja savo kurtą virtualių įrenginių programinės įrangos aplinką, į kurią įeina tinklo, nuoseklios ir lygiagrečios šąsąjų komunikacijos [10].



6 pav. Agilent virtualių įrenginių programinė įranga

„National Instruments LabVIEW“ taip pat yra grafinė programavimo kalba. Ji turi daugiau lankstumo nei kiti standartiniai laboratoriniai instrumentai, nes jos bazė yra programinė įranga. „LabVIEW“ gali charakterizuoti prietaisų funkcionalumą. Ši programinė įranga taip pat turi savą virtualių įrenginių programinės įrangos aplikaciją. Su šia aplikacija gali valdyti nuoseklias, lygiagrečias ir tinklo sąsajų komunikacijas. „LabVIEW“ gali naudoti ir „Agilent VEE“ virtualių įrenginių valdymo aplikacijas (žr. 7 pav.).



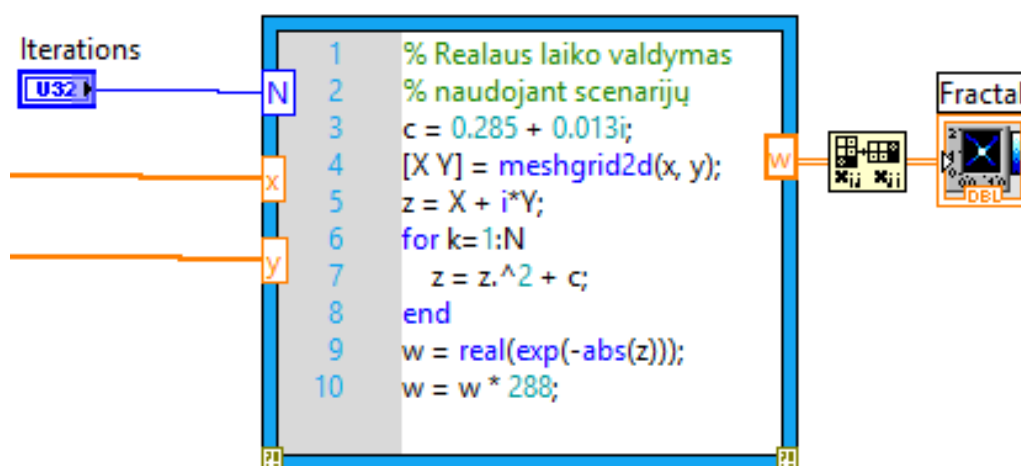
7 pav. LabVIEW virtualių įrenginių programinė įranga

„MATLAB“ yra daugiaplatformė programinė įranga, skirta įvairių mokslo šakų problemoms spręsti. Ši platforma taip pat turi virtualių įrenginių valdymo aplikacijas. „MATLAB“ programinė įranga yra labai populiari mokslininkų tarpe. Įrangos bibliotekos yra plačios, daug funkcijų ir galimybių atliekančios. „MATLAB“ programa dėl jos sudėties praranda greitį, todėl visos virtualių įrenginių valdymo aplikacijos dirba lėčiau nei „Agilent VEE“ ar „LabVIEW“.

## 1.4 MATLAB ir LabVIEW platformų komunikavimas

### 1.4.1 MATLAB scenarijaus panaudojimas LabVIEW terpėje

Vienas iš programinių metodų matavimo sektoriuje - tai tiesioginis įrenginių programavimas „LabVIEW“ aplinkoje su „MATLAB“ scenarijumi. Šis metodas unikalus tuo, kad pasižymi greitu duomenų komunikavimu tarp „LabVIEW“ aplinkos ir „MATLAB“ scenarijaus (žr. 8 pav.). Scenarijaus branduolys, įterptas programoje, naudoja „LabVIEW“ kintamuosius, o visus skaičiavimus atlieka „MATLAB“ branduolyje.



8 pav. MATLAB scenarijus

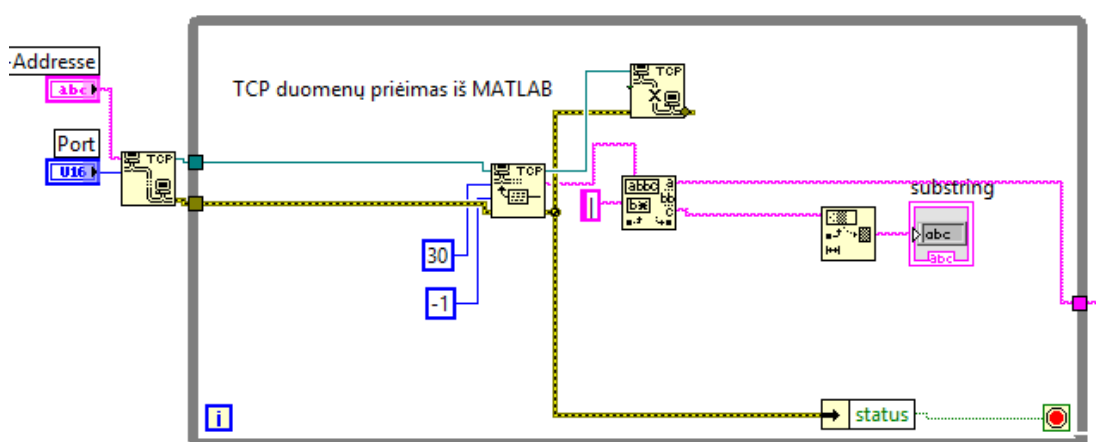
Apžvelgiant metodo struktūrą detaliau galima pasakyti, kad šis metodas naudoja „ActiveX“ technologija. „ActiveX“ valdiklis yra .NET technologijos objektas, taigi tai yra tiesiogiai operacinėje sistemoje vykdomas dvejetainis modulis. Valdikliai yra maži programų kūrimo blokai, kuriuos galima naudoti platinamoms programoms, veikiančioms internete (žiniatinklio naršyklėms kurti). „ActiveX“ gali būti taikomos ir duomenų rinkimui. „LabVIEW“ aplinkoje „ActiveX“ naudojama kaip sąsaja sukommunikuoti „LabVIEW“ ir „MATLAB“ scenarijų.

Šis komunikavimo būdas leidžia programoms dirbti tik iš anksto aprašius scenarijų. Scenarijaus panaudojimas nepatogus tuo, kad sukūrus „LabVIEW“ aplikaciją kintamųjų keisti nebegalima, reikia naudotis tik standartiniais kintamaisiais su vienodais tipais. Kitas būdas padidinti kintamųjų kiekį, tai kurti kintamųjų masyvus. Juos galima apibrėžti dideliu kintamųjų skaičiui, bet visi kintamieji gali būti vienodi, jei jų neapibrėš kitais kuriamais kintamaisiais.

„MATLAB“ scenarijaus pritaikymas „LabVIEW“ terpėje yra patogus tol, kol programa dar yra derinimo stadijoje. Sukūrus aplikaciją šis metodas praranda dinaminį lankstumą kintamųjų kūrimo ir pritaikyme.

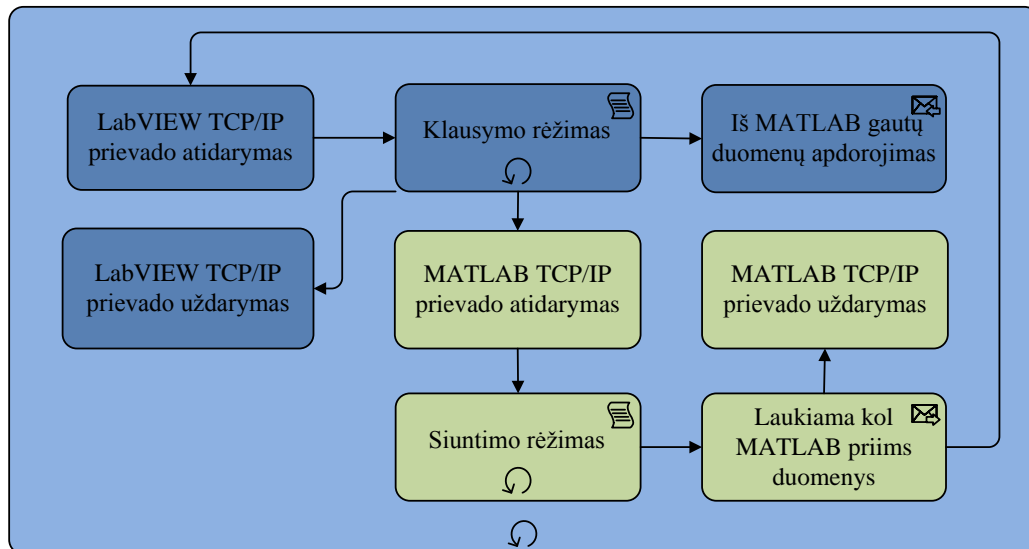
## 1.4.2 TCP/IP API protokolinis valdymas

API (angl. *Application Programming Interface*) tai Microsoft Windows operacinių sistemų šeimai sukurtas funkcijų bei bibliotekų rinkinys, kuris leidžia aplikacijai (programai) tiesiogiai bendrauti su operacine sistema. Šiuo atveju API funkcijos yra taikomos susinchronizuoti duomenų siuntimą ir gavimą tarp „LabVIEW“ ir „MATLAB“ naudojant TCP protokolą (žr. 9 pav.). TCP/IP (Standartinis duomenų perdavimo protokolų rinkinys, kurio pagrindu veikia internetas bei daugelis privačių komercinių tinklų. Svarbiausi šio protokolo komponentai – TCP ir IP protokolai.) Taip pat, tai viena iš galimybių, kuri leidžia keistis duomenimis realiu laiku. Tinkamai suderinus prievadus ir duomenų keitimosi laiko intervalus, galima išgauti didelį duomenų perdavimo greitį [11].



9 pav. LabVIEW TCP/IP prievado valdymas

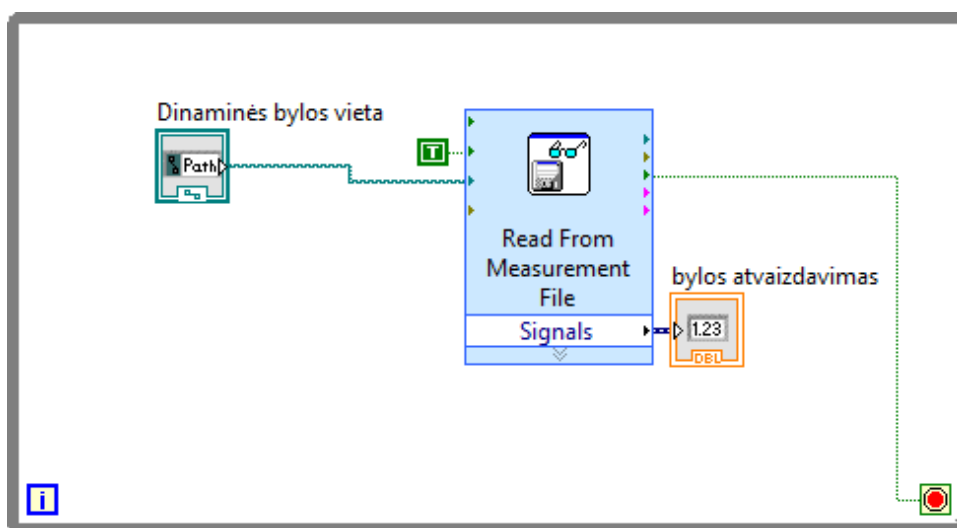
Duomenys gali keistis tarp „MATLAB“ ir „LabVIEW“ naudodami kodo šifrą, taip išvengtų trikdžių tame pačiame kanale, bet šifravimas ir dekodavimas užimtų didelį laiko tarpą. Šiuo būdu matavimai realiu laiku tampa lėtesni. Kitas sprendimo būdas - tai tikrinti ar prievadas nėra užimtas kitų programų. Toks būdas leistų užtikrinti svarų duomenų keitimąsi tarp MATLAB ir LabVIEW (žr. 10 pav.).



10 pav. TCP/IP komunikavimo algoritmas tarp LabVIEW ir MATLAB

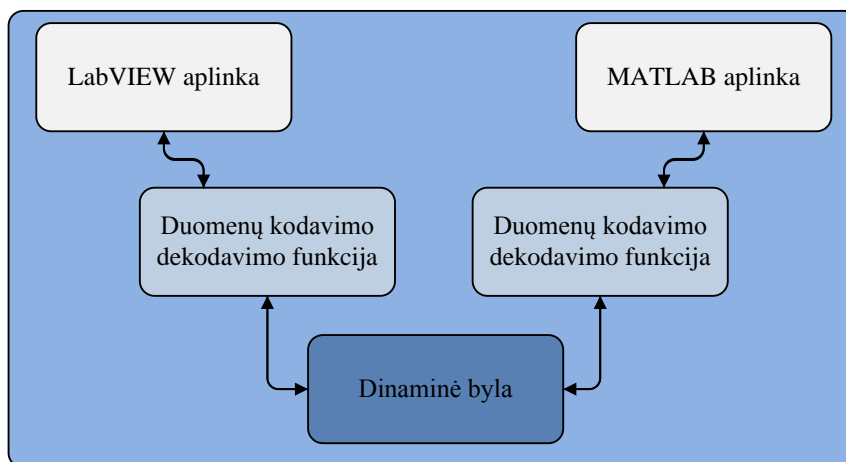
### 1.4.3 Dokumentinis duomenų keitimasis

Tikriausiai vienas iš seniausių duomenų keitimosi metodų, kuris naudoja dokumentą kaip duomenų keitimosi terpę (žr. 11 pav.). Šis būdas pagrįstas laikino dokumento sukūrimu, duomenų įrašymu ir nuskaitymu. Siuntėjas sukuria dokumentą pažymėtoje direktorijoje. Į dokumento sudėtį duomenis įrašo naudodamasis standartizuotą ASCII simbolių koduotę bei atskirdamas duomenis tabuliatoriumi, tipas – dvigubas (>>SAVE filename X -ascii -double - tabs). Kita programa ar sąsajos algoritmas tikrina ar nurodytoje vietoje nėra sukurtų dinaminių dokumentų. Atradus dokumentą vartotojas nuskaityto duomenų sudėtį ir pašalina laikinojo dokumento sudėtį [12].



11 pav. Dokumentinis informacijos dalinimosi metodas

Pastebėta, kad šis būdas turi trūkumą – su dokumento gali dirbti tik viena programa (žr. 12 pav.). Greitiems duomenų mainams galima naudoti kelis dinامينius dokumentus. Taikant tokį būdą atsirastų greitesnis duomenų pernešimas.



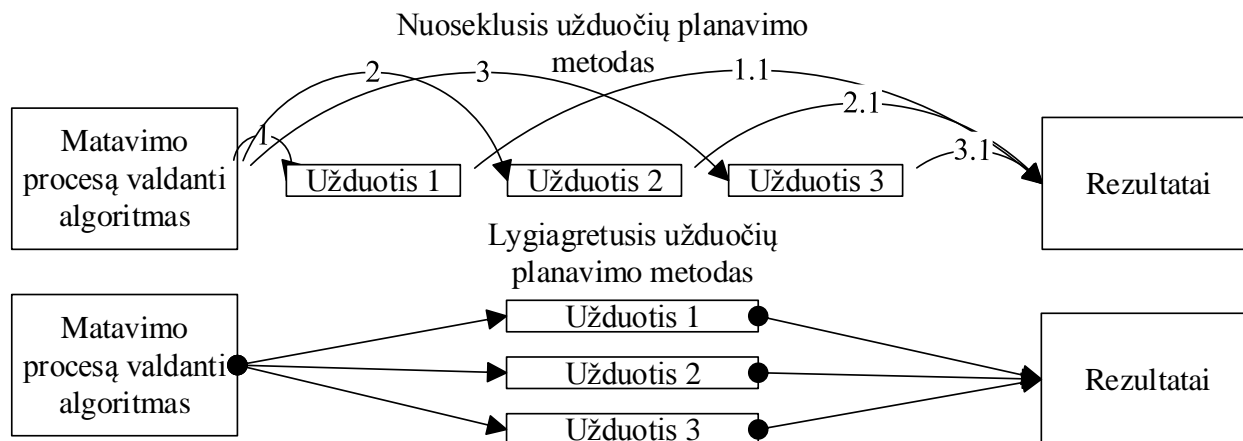
12 pav. Dinaminis dokumentinis komunikavimas tarp LabVIEW ir MATLAB

### 1.5 Nuoseklusis ir lygiagretusis automatizuotų matavimų metodai

Automatizuojant matavimo procesą svarbu užtikrinti greitą matavimo prietaisų aptarnavimą, todėl automatizuotiems matavimams yra priskiriama realaus laiko sąvoka. Norint aptarnauti matavimo prietaisus realiu laiku, reikalinga brangi ir tiksli techninė aparatūra, su greitais duomenų perdavimo kanalais. Dažniausiai ši aparatūra yra specializuota. Kitas būdas kaip būtų galima užtikrinti matavimo proceso aptarnavimą realiu laiku tai yra programiškai. Programuojant algoritmą, kuris vykdo matavimo procesą, turi būti paprastas, mažai programinio kodo užimantis algoritmas. Tokia metodika naudojama dėl to, kad papildoma kodo dalis gali stabdyti ir lėtinti matavimo procesą. Jeigu automatizuoto matavimo procese yra vykdomi sudėtingi skaičiavimai, su kuriais sudėtingėja matavimo algoritmas, tada yra taikomas atvirkštinis sinchronizavimo metodas. Sinchronizuojant matavimo prietaisus, naudojami tam tikri vėlinimo algoritmai, kurie padaro sinchronizuotą matavimo procesą. Toks programinis metodas turi ir minusų. Realiu laiku vartotojas negali sekti duomenų. Automatizuojant procesą atsiranda tam tikri proceso valdymo nuokrypiai dėl informacijos perskaičiavimo vėlinimo. Jeigu vykdomas procesas yra lėtas, tada laiko vėlinimas turi nedidelę įtaką proceso valdymui. Aptarnaujant matavimo aparatūrą programiškai yra naudojami du matavimo metodai – nuoseklusis ir lygiagretusis užduočių planavimo metodai. Taikant šiuos metodus atsiranda programinis lankstumas valdant matavimo prietaisus.

Nuoseklusis užduočių planavimo metodas taikomas tokiam matavimo procesui, kurio rezultatams didžiulę įtaką daro kitų prietaisų parametrai. Naudojant nuoseklųjį metodą visos uždutys, iš programinės įrangos, prietaisams yra siunčiamos ir gaunamos nuosekliai. Naudojant

lygiagretųjų užduočių planavimo metodą, užduotys siunčiamos matavimo prietaisams taip, kad skirtingi matavimo prietaisai galėtų matuoti – valdyti lygiagrečiai. Naudojant šią užduočių planavimo metodiką yra funkcionaliau išnaudojamas matavimo algoritmą vykdomasis kompiuteris (žr. 13 pav.).



13 pav. Nuoseklusis ir lygiagretusis užduočių planavimo metodai

## 1.6 Kuriamos programinės įrangos priemonių taikymas ir parinkimas

Analitinėje dalyje analizuojami metodai, kurie leistų įgyvendinti programinės įrangos kūrimo ir projektavimo darbus. Gilintasi į alternatyvių produktų programines sandaras. Realizuojant iškeltus uždavinius, svarbu surasti tinkamiausias sprendimų priemones. Priemonių suradimui yra formuojama lentelė, kuri skirta programinės įrangos, automatizuotiems matavimams, uždavinių ir galimų sprendimų formavimui.

1 lentelė

### Programinės įrangos automatizuotiems matavimams įgyvendinimo užduotys

Nr.	Uždaviniai	Galimi sprendimai
1.	Matavimo prietaisų valdymas per programuojamą platformą	Virtualiųjų prietaisų architektūra;
2.	Realaus laiko duomenų dalinimosi galimybės tarp scenarijaus vykdymo programos ir automatizuotų matavimų programinės įrangos	TCP/IP; Informacijos dalinimasis naudojant dinaminis dokumentus; Dinaminės bibliotekos taikymas;
3.	Vartotojo valdymo sąsaja (scenarijaus vykdymo aplinka)	„MATLAB“, „Virtual programming laboratory“, „Agilent VEE“
4.	Matavimo prietaisų užduočių planavimo galimybės.	Nuoseklusis užduočių planavimo metodas; Lygiagretusis užduočių planavimo metodas;

Skyriuje, komunikacinės valdymo sąsajos, nagrinėjama ir ieškoma metodų, kurios leistų sukombinuoti vykdomo matavimo scenarijaus programą su valdomais matavimo prietaisais. Rezultate nagrinėjama virtualių valdymo prietaisų platforma (VISA). Ši platforma naudoja tiek

lygiagrečias, tiek nuosekliai komunikacines sąsajas. Prietaisų valdymo platforma būtų tinkama automatizuotų matavimų programinei įrangai, jei turėtų lygiagrečią prietaisų valdymo programinę architektūrą. Platforma geba duomenis perduoti ir gražinti lygiagrečiu režimu. „VISA“ turi dinaminę biblioteką (DLL), per kurią galima valdyti kiekvieną komunikacinę sąsają [14].

Programinės įrangos, skirtos automatizuotiems matavimams, funkcija yra perduoti ir gražinti duomenis tarp valdymo prietaiso algoritmo ir matavimo įrenginių realiu laiku, kartu maksimaliai išnaudojant prietaisų dinaminis savikontrolės laiko momentus. Patobulinus „VISA“ įrenginių valdymo algoritmą, atsiranda galimybė užtikrinti realų laiką matavime, išnaudojant visus laiko dydžius valdymo funkcijose.

Duomenų keitimasis tarp automatizuotos matavimo programinės įrangos ir vartotojo valdymo sąsajos yra viešas ir prieinamas metodas iš bet kokios išorinės programinės sistemos. Toks būdas suteikia programinei įrangai lankstumo jos pritaikyme. Tokio sprendimo esminis minusas yra tai, kad perduodant duomenis iš vienos programinės sistemos į kitą programinę sistemą atsiranda laiko uždelsimas matavimuose. Spendžiant laiko uždelsimo problemą yra ieškomas duomenų perdavimo metodas, kuris turėtų mažiausią įtaką laikui, duomenų perdavimo metu.

## **1.7 Skyriaus išvados**

1. Analitinėje dalyje nagrinėjami metodai, kurie gali būti taikomi, kuriant automatizuotų matavimų programinę įrangą. Išanalizuoti taikomi automatizuotų matavimų metodai. Lanksčiausias ir lengviausias vartotojui dirbti su:
  - Virtualių prietaisų (VISA) įranga;
  - Praplėtimo programine įranga.
2. Tai pat nagrinėjami programinės įrangos algoritmo įgyvendinimo priemonės duomenų perdavimo galimybės, vykdant automatizuotą procesą:
  - Dokumentinis duomenų perdavimo metodas;
  - TCP/IP duomenų perdavimo metodas.Šie metodai tinkami ir taikyti programos algoritmo įgyvendinime.
3. Programinės įrangos matavimo prietaisų valdymui naudojami du užduočių planavimo metodai:
  - Nuoseklusis užduočių planavimo metodas;
  - Lygiagretusis užduočių planavimo metodas.

Taikant šiuos metodus tikimasi, jog automatizuotų matavimų programinė įranga funkcionaliai valdys matavimo procesą ir prietaisus.



## 2. METODOLOGINĖ DALIS

Programinės įrangos automatizuotiems matavimams sukūrimui naudojami analitinėje dalyje paminėti metodai. Vienas iš keliamų uždavinių, jog kuriama programinė įranga turi realiaje laike aptarnauti matavimo prietaisus. Matavimo prietaisai valdomi per virtualiuosius prietaisus (VISA). Analitinėje dalyje buvo nagrinėjami kelių kūrėjų virtualūs prietaisai: „National Instruments“ ir „Mathworks“. „National Instruments“ VISA greičiau ir kokybiškiau (su mažesniu susijungimo klaidų kiekiu) aptarnauja matavimo prietaisus, todėl programinės įrangos automatizuotiems matavimams kūrimui pasirinkta „National Instruments LabVIEW“ grafinio programavimo kalba.

Programos įgyvendinimui kuriami trys pagrindiniai algoritmai, kurie pagal gautas užduotis iš scenarijaus vykdymo programos, valdys matavimo prietaisus. Matavimo algoritmą aprašančio scenarijaus interpretatoriaus funkcija yra adaptuoti matavimo scenarijaus užduotis matavimo prietaisams suprantamomis komandomis. Šiam algoritmui realizuoti naudojami duomenų keitimosi metodai: TCP/IP protokolas ir dokumentinis keitimosi metodas. Gautos užduotys iš scenarijaus vykdymo programos yra išskaidomos matavimo prietaisams suprantamomis komandomis. Už matavimo proceso valdymą yra atsakingas užduotis valdantis algoritmas. Šis algoritmas kaupia užduotis dinaminėje duomenų bazėje ir vykdo jas nuoseklia būsena. Algoritmo realizavimui panaudotas FI-FO metodas. Matavimo prietaisų valdymo algoritmui realizuoti naudojamas virtualių prietaisų (VISA) valdymo metodas.

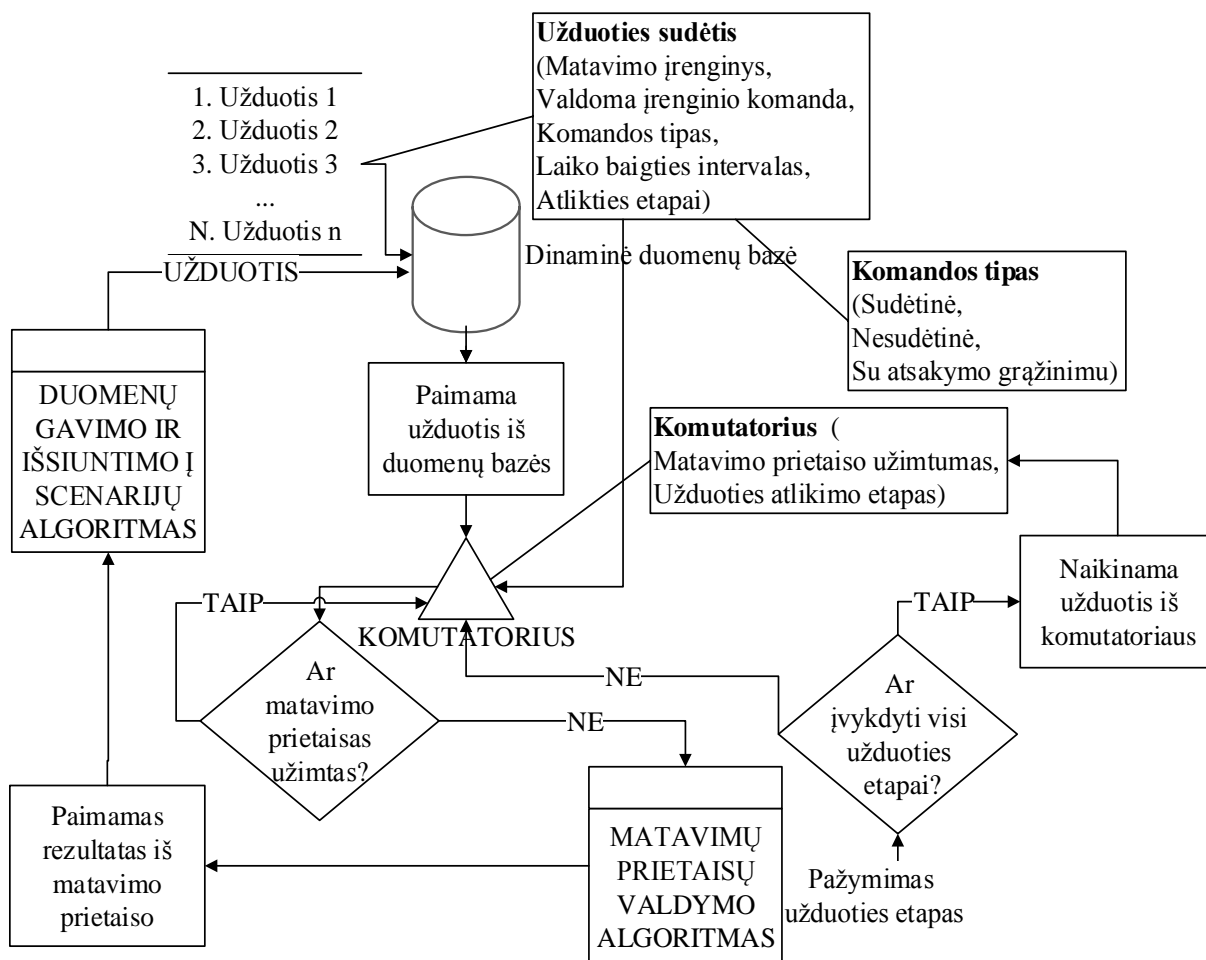
### 2.1 Pagrindiniai automatizuotų matavimų programinės įrangos algoritmai

Išanalizavus programinės įrangos poreikį bei parinkus įrangos kūrimo metodus, pereinama prie programinės įrangos kūrimo etapo. Programos branduolys kuriamas taikant virtualius prietaisus.

Programinę įrangą sudaro trys sąlyginai nepriklausomai veikiančios procesai: užduočių matavimo prietaisams valdymo algoritmas, matavimo algoritmą aprašančio scenarijaus interpretatorius, matavimo prietaisų valdymo ir duomenų perdavimo algoritmas.

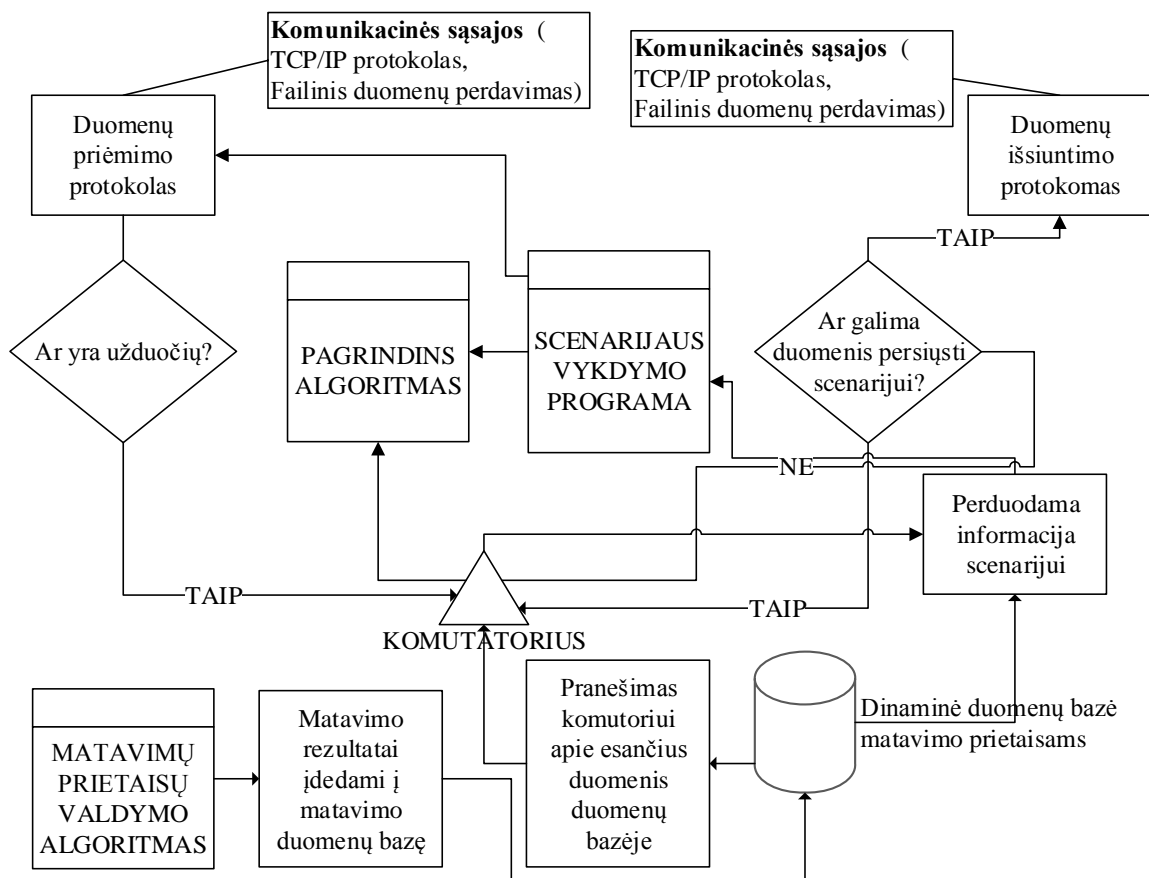
Pirmame programos variklio kūrimo etape yra kuriamas algoritmas, kuris gebėtų savarankiškai valdyti prietaisus. Branduolys susideda iš trijų pagrindinių algoritmų. Pirmasis algoritmas kaupia užduotis sukurtoje slenkančioje duomenų bazėje. Toks sprendimas leidžia valdyti skirtingus prietaisus naudojant tą pačią sąsają. Užduotys yra įvertinamos atlikties etapais. Gauta užduotis įdedama į paskutinę eilę duomenų bazėje. Kiekviena užduotis yra priversta laukti savo eilės matavimo vykdyme. Duomenų bazės algoritmas prižiūri užduoties būseną. Užduotis susideda iš tokių matuotų parametrų kaip: matavimo įrenginys, valdoma įrenginio komanda,

komandos tipas, laiko baigties intervalas. Paleidus užduotį vykdymui komanda pati pasirenka matavimo prietaisą, kurį naudos komandos įgyvendinimui. Valdomajai įrenginio komandai yra siunčiamas pranešimas, kurio reikšmę supranta tik pats matavimo prietaisas. Komandos tipas nurodo, kaip matavimo įrenginys elgsis. Komandos naudojamos virtualių prietaisų (VISA) valdymui. Tipas naudojamas trejais skirtingais variantais: ar siunčiama komanda yra sudėtinė, ar nesudėtinė, ar prietaisas turės grąžinti atsakymą. Priklausomai nuo tipo yra apskaičiuojamas laiko baigties intervalas. Šio algoritmo privalumas yra toks, kad jis geba vykdyti kelias užduotis vienu metu. Viena komandos užduotis yra vykdoma etapais. Etapinis užduoties valdymas leidžia valdyti kelis prietaisus „lygiagrečiai“ vienu metu. Jei nuo užklauso prietaisas turės grąžinti rezultatą, tada atsiranda laikinis užlaikymas duomenų perdavimui iš prietaiso į programinę įrangą. Per tą laiką, kuomet užklausa yra siunčiama programinei įrangai, sekanti užduotis gali būti vykdoma, jei ji nėra skirta tam pačiam įrenginiui. Kiekviename tikrinimo cikle yra numeruojami užklauso etapai. Įvykdžius visus užklauso etapus, užklausa yra pašalinama iš duomenų bazės ir atkeliamas sekanti užklausa, kuri buvo išsiųsta po jos (žr. 14 pav.).



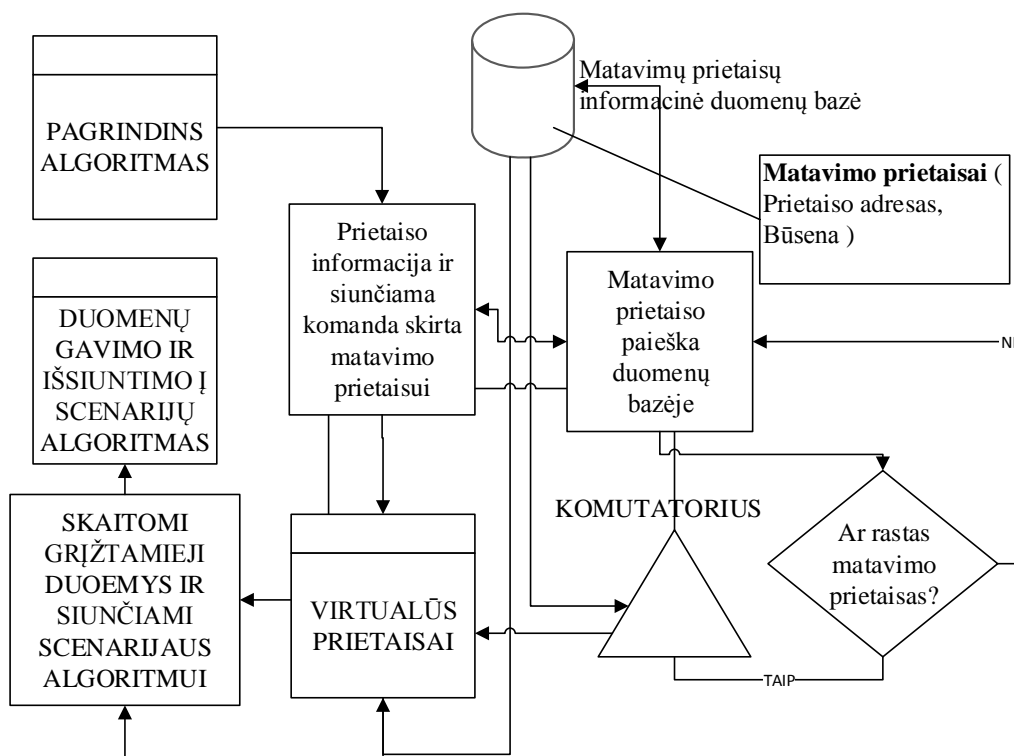
14 pav. Užduotis valdantis algoritmas

Antrojo pagrindinio algoritmo pagrindinė funkcija yra priimti užklausas ir išsiųsti atliktus užklausų atsakymus programai, iš kurios yra valdomas matavimo procesas. Šio algoritmo veikimas nėra priklausomas nuo pirmojo algoritmo, kuris kaupia ir apdoroja užklausas. Tokio tipo algoritmas komunikuoja su matavimo scenarijumi, kuris valdo matavimą. Komunikavimo būdų gali būti įvairių. Siekiant užtikrinti realaus laiko duomenų perdavimą, yra naudojami TCP/IP protokolas ir dokumentinis duomenų komunikavimo būdas. Automatizuotų matavimų programinės įrangos duomenų komunikavimo algoritmas turi kelis režimus. Pirmame režime algoritmas budi ir laukia ateinančių užklausų iš matavimo scenarijaus. Gavęs užklausą algoritmas patikrina, ar jos formatas yra tinkamas vykdymui. Tikrinimas atliekamas stacionarioje duomenų bazėje, kur yra saugomi visi konfiguracioniai duomenys apie įrenginį ir jo valdymą. Jei užklausoje yra randama atitikčių, tada užklausa yra įdedama į pirmojo algoritmo dinaminę duomenų bazę. Kitu atveju užklausa yra sunaikinama. Antrasis algoritmo režimas yra perjungiamas, kai bent vienas įrenginys yra išdavęs užklausos rezultatą. Tuo momentu yra perjungiamas budėjimo režimas ant išsiuntimo režimo. Algoritmas signalizuoja scenarijui apie gautąjį rezultatą. Išsiuntus rezultatą matavimo algoritmui, algoritmas grąžinamas į budėjimo režimą (žr. 16 pav.).



15 pav. Matavimo algoritmą aprašančio scenarijaus interpretatorius

Trečiasis algoritmas yra atsakingas už matavimo prietaisų valdymą ir rezultatų prieinamumą. Po pagrindinio algoritmo išsiųstos užduoties apdorojimui, užklausa patenka į prietaisų valdymo algoritmą, tada algoritmas tikrina, kokiam prietaisui yra skirta užklausa. Veikimas labai panašus į komutatorių, kuris perduoda ir priima duomenis. Aptikus naudojamą matavimo prietaisą, komanda yra išsiunčiama matavimo prietaisui. Išsiuntus komandą, prietaisas yra įtraukiamas į registrą. Registras reikalingas tam, kad tikrintų ar įrenginys grąžino rezultatą. Grąžinus rezultatą, įrenginys yra išimamas iš tikrinimo registro. Grąžintas rezultatas yra perduodamas užduočių priėmimo ir išsiuntimo algoritmui, kuris išsiunčia rezultatą matavimo scenarijui (žr. 16 pav.).



16 pav. Matavimo prietaisų valdymo ir duomenų perdavimo algoritmas

Automatizuotų matavimų programinės įrangos kodą galima rasti 10 priede. Programinės įrangos kodas rašytas „LabVIEW“ grafine programavimo kalba. Programavimo kalba pasirinkta dėl virtualiųjų prietaisų adaptacijos „LabVIEW“ platformai.

## 2.2 Automatizuotų matavimų programinės įrangos valdymas

Programinė įranga sudaryta iš trijų skirtingų valdymo algoritmų. Algoritmai aprašyti skyriuje - pagrindiniai automatizuotų matavimų programinės įrangos algoritmai. Programinės įrangos algoritmai savarankiškai analizuoja matavimo prietaisų būsenas, valdo juos. Pagrindinis programinės įrangos funkcionalumas yra tas, kad gali valdyti ir kontroliuoti automatizuotą matavimo procesą.

Programinėje įrangoje sukurta vartotojo sąsaja leidžia vartotojui pasirinkti matavimo prietaisus ir komunikacijos metodus, per kuriuos jie bus valdomi. Programa unikali tuo, kad ji gali būti adaptuota kitoms programinėms įrangoms. Programoje įdiegti du komunikavimo metodai: TCP/IP protokolas (2 priedas) ir informacijos keitimas dokumentais (1 priedas). Taikant šiuos metodus, keistis duomenimis galima ir su kitomis programomis: C, C++, JAVA, Python pagrindu padarytomis programomis.

Įgalinti matavimo prietaiso valdymo funkciją darbo režimui, reikalingą ją įjungti per programinę įrangą. Tokia procedūra atliekama pridėjus užduotį, paspaudus ant programos pelės klavišą „Pridėti užduotį“. Atsiradus naujam prietaiso skirtukui, parenkamas matavimo prietaisas, kuris bus naudojamas matavimo metu „Device“ lange. Parenkama komanda „Command“, kuria nusakoma, koks uždavinys bus atliekamas matavimo prietaise. Parinkus matavimo prietaisą ir komandą lauke „Variable in MATLAB“ pavadiname užduotį vardu. Šiuo kintamojo vardu bus galima iššaukti matavimo prietaiso funkciją automatizuoto matavimo vykdymo metu. Programoje padaryta galimybė importuoti arba išeksportuoti prietaisų nustatymus į tam tikrą informacinę laikmeną (žr. 17 pav.).

Duomenų perdavimo būdas

M-Failas

Serverio IP adresas  
localhost

Agilent 4395a  
COM3

N5752  
COM5

33500B  
COM2

Fluke 196C  
COM1

STM32  
COM9

Device	Command	Variable in MATLAB	Value
Agilent33500B	1k įtampa	itampa	
Agilent4395A	Dažnis START	daznio_nust	
AgilentN5752	N5752A Srovė	srove	
Fluke196C	Fluke196C A	nusk_duom	

Pridėti užduotį  
Importuoti nustatymus  
Eksportuoti nustatymus

17 pav. Vartotojo sąsaja

Duomenų perdavimo bloke pasirenkamas būdas, kaip keisis informacija tarp sukurtos programinės įrangos ir scenarijaus vykdymo programos (MATLAB). Suprogramuoti du metodai: duomenų keitimasis per TCP/IP protokolą arba keitimasis per dinaminį dokumentą (M-Failas). Tiek vienu, tiek kitu metodu keistis informacija galima nuotoliniu būdu. Naudodamiesi TCP/IP protokolu galima keistis duomenimis tiesiogiai be tarpinio serverio. Taikant dinaminį dokumentą duomenų keitimuisi nuotoliniu būdu yra reikalingas tarpinis serveris. Populiariausi naudojami serveriai gali būti pastatyti ant internetinių puslapių palaikančių platformų, dokumentų perdavimo protokolo (FTP) ar kitų alternatyvų, kuriomis būtų galima dalintis duomenimis nuotoliniu būdu.

Programinė įranga, kuri skirta automatizuotiems matavimams, yra matavimo sistemos komplekso dalis. Pilnai automatizuoti matavimo procesą vien su šia programa nėra galimybės. Kad ši programa būtų funkcionali, yra reikalinga naudoti papildomą programinę įrangą, kuri turi dinamines scenarijaus vykdymo galimybes. Naudodamiesi duomenų keitimosi TCP/IP protokolu arba dokumentinio keitimosi metodu programinę įrangą automatizuotiems matavimams galima sukombinuoti su kita programa, kuri turi dinamines scenarijaus vykdymo galimybes. Eksperimentams ir tyrimams panaudojama „MATLAB“ programa (žr. 18 pav.). „MATLAB“ programoje parašyta paprogramė, kuri realiame laike keičiasi duomenimis tarp automatizuotų matavimų programinės įrangos ir „MATLAB“ programos.

```

14 - while true
15
16
17     %% duomenų nuskaitymas iš Fluke196C
18
19     Q = fluke(labwM('fluke',0)); % užduotis automatizuotų matavimų programinei įragai
20     laikas = clock;
21     Q_ilgis = length(Q);
22     Nauji_duom = zeros(2,Q_ilgis*1000)';
23     Nauji_duom(1:Q_ilgis,1:2) = Q;
24     if Q_ilgis == 1200
25
26     U = length(Nauji_duom);
27     %%%% furje transformacija
28     if U == 5970000
29         Fs = abs((Nauji_duom(2,2) - Nauji_duom(1,2)))*0.4975;
30     else
31         Fs = abs((Nauji_duom(2,2) - Nauji_duom(1,2)));
32     end

```

18 pav. MATLAB automatizuoto matavimo scenarijus

Statinė duomenų bazė sukurta naudojant „Kableliais atskirtos reikšmės“ formatą (CSV pagal angl. comma - separated values) – dokumentų formatas, skirtas saugoti duomenis lentelėms. Šis formatas sukurta dar iki asmeninių kompiuterių paplitimo, todėl yra plačiai naudojamas daugelyje kompiuterių platformų. Naudojantis šiuo formatu sukurta duomenų bazė matavimo prietaisų komandoms (žr. 2 lent.).

**Matavimo prietaisų duomenų bazė**

Komandos pavadinimas	Prietaiso valdymo komanda	Komandos valdymo metodas
Kanalas 1	CHAN1	FALSE
Kanalas 2	CHAN2	FALSE
Dažnis START	STAR	TRUE
Dažnis STOP	STOP	TRUE
Trendo nuskaitymas	OUTPMKR?	TRUEOUT
Grafiko nuskaitymas X	OUTPLML?	TRUEOUT
Grafiko nuskaitymas Y	OUTPDTRC?	TRUEOUT
Impedance: Resist(R)	MEAS  IRE	FALSE
4395a DATA MEMORY	DATMEM	FALSE
4395a DATA-MEMORY	MATH  DMNM	FALSE
4395a DATA MATH : DATA	MATH  DATA	FALSE
4395a NETWORK: A/R	MEAS  AR	FALSE

Duomenų bazė suskaidyta į tris atskiras grupes: komandos pavadinimas, prietaiso valdymo komanda ir komandos valdymo metodas. Komandos pavadinimuose aprašomos matavimo prietaisų valdymo komandų funkcinės veiklos. Pagal šiuos pavadinimus vartotojas pildo programinės įrangos užduočių skiltis. Prietaisų valdymo komandų skiltyje pildomi prietaisų valdymo raktažodžiai. Pagal šiuos raktažodžius, naudojant komunikacinę sąsają, yra valdomi matavimo prietaisai. Komandos valdymo metode nurodomas būdas, kaip bus valdomas matavimo prietaisas. Naudojami trys matavimo prietaiso valdymo būdai: „True“, „False“ ir „Trueout“. Pirmasis prietaiso valdymo būdas „True“ nurodo, kad matavimo prietaisui bus siunčiama komanda su papildomu nustatymu. Papildomas nustatymas gali būti tiek raidinė, tiek skaitinė vertė. Antras būdas „False“ nurodo, kad siunčiama komanda bus be papildomo nustatymo. Jei komanda aprašyta su „Trueout“ būdu, tai reiškia, kad išsiuntus valdymo komandą bus gražinamas rezultatas. Taikant tokį prietaisų valdymo metodą yra lengviau aprašyti matavimo prietaisus. Pilną matavimų prietaisų duomenų bazę galima rasti 9 priede.

### 2.3 Matavimo prietaisų užduočių ir komutacijos laikų nustatymas

Matavimo prietaisai yra automatizuotų matavimų komplekso dalis. Programinėje įrangoje matavimo prietaisų funkcijos apibrėžiamos kaip vykdikliai su grįžtamoju ryšiu. Ištirti programinės įrangos laikinį poveikį matavimo procesui reikalingą žinoti matavimo prietaisų laike apibrėžtas charakteristikas. Naudojantis laiko charakteristikomis galima teoriškai apskaičiuoti automatizuoto matavimo proceso vykdymo trukmę ir nustatyti įtaką atliekamam matavimo procesui.

Matavimo prietaisų laikinių charakteristikų nustatymui taikomi tokie metodai: prietaiso komutacijos trukmės nustatymo metodas ir prietaiso užduoties atlikimo laiko trukmės nustatymo metodas. Taikant prietaiso komutacijos laiko trukmės nustatymo metodą yra išmatuojama komunikacijos laiko trukmė iki prietaiso ir atgal. Komutacijos laikai gali skirtis dėl skirtingų matavimo prietaisų sąsajų ir matavimo prietaiso charakteristikų. Prie asmeninio kompiuterio prijungiamas matavimo prietaisas ir siunčiama neįgyvendinama užduotis į matavimo prietaisą. Matavimo prietaisas, gavęs neįgyvendinamą užduotį, neatlikdamas jokio matavimo proceso, siunčia atsakymą atgal į asmeninį kompiuterį. Yra registruojami išsiuntimo ir gavimo laikai. Išsiuntimo gavimo laikų skirtumas yra laikomas matavimo prietaiso komutaciniu laiku. Išreikšta formulė:

$$t_{kom} = t_{gavimas} - t_{išsiuntimas} ; \quad (1)$$

čia:  $t_{kom}$  - matavimo prietaiso komutacinė laiko trukmė, s;  $t_{išsiuntimas}$  - išsiuntimo metu užfiksuotas laikas, s;  $t_{gavimas}$  - gavimo metu užfiksuotas laikas, s.

Prietaiso užduoties atlikimo laiko trukmės metodu yra išmatuojama matavimo prietaiso vienos užduoties atlikimo laiko trukmė. Taip pat matavimo prietaisas yra jungiamas prie asmeninio kompiuterio ir siunčiama matavimo prietaisui jam suprantama užduotis. Matavimo prietaisas, gavęs užduotį, įvykdo ją ir tik tada grąžina rezultatą į asmeninį kompiuterį. Laikas matuojamas užduoties išsiuntimo ir gavimo metu. Iš šio laiko intervalo atėmus matavimo prietaiso komutacinę laiko trukmę yra gaunamas matavimo prietaiso užduoties įvykdymo laikas. Išreikšta formulė:

$$t_{užd} = t_{gavimas} - t_{išsiuntimas} - t_{kom} ; \quad (2)$$

čia:  $t_{užd}$  – matavimo prietaiso užduoties įvykdymo laikas, s;  $t_{išsiuntimas}$  - išsiuntimo metu užfiksuotas laikas, s;  $t_{gavimas}$  - gavimo metu užfiksuotas laikas, s;  $t_{kom}$  - matavimo prietaiso komutacinė laiko trukmė, s.

## 2.4 Programinės įrangos teorinis modelis

Kuriant programinės įrangos automatizuotiems matavimams teorinį modelį, siekiama pateisinti programinės įrangos veikimo efektyvumą, valdant matavimo procesą. Matavimo proceso efektyvumo gerinimui naudojami nuoseklūs ir lygiagretūs programinės įrangos užduočių planavimo metodai. Taikant užduočių planavimo metodus tikimasi efektyviai išnaudoti

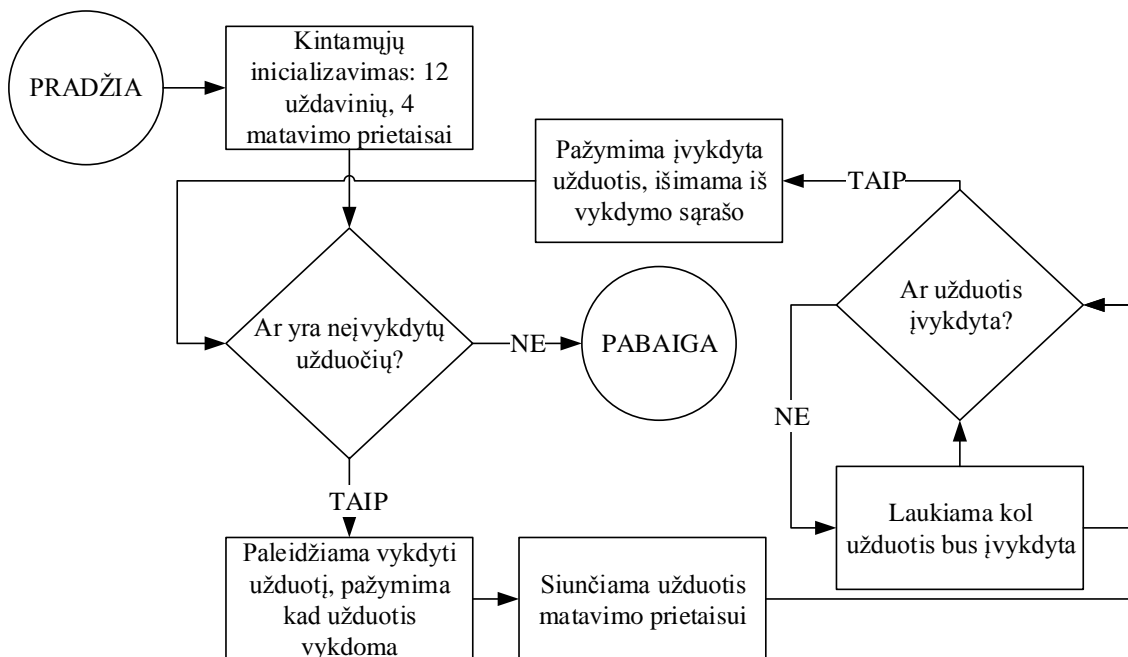


matavimo prietaisus taip pagreitinant matavimo procesą. Matavimo algoritmo užduočių planavimui yra parengiamas programinės įrangos automatizuotiems matavimams teorinis modelis. Modelyje nagrinėjami programinės įrangos nuoseklūs ir lygiagretūs užduočių planavimo metodai. Teorinio modelio realizacijai pasitelkta „MATLAB“ modeliavimo programa, pagal kurią programuojama automatizuotiems matavimams skirta programinė įranga.

Taikant užduočių planavimo metodus, siekiama optimizuoti matavimo procesą. Nuoseklūs užduočių matavimo metodas dažniausiai taikomas cikliniams matavimams. Taikant šį metodą matavimo prietaisai valdomi pagal matavimo scenarijų. Šiuo atveju matavimo scenarijaus funkcijos yra realiu laiku surišamos su matavimo prietaisais. Matavimo scenarijaus funkcijos yra sinchronizuotai vykdomos su matavimo prietaisais. Šis užduočių planavimo metodas yra naudingas tuo, kad lengva logine forma (scenarijus – prietaisas, prietaisas - scenarijus) galima realizuoti ir valdyti matavimo procesą. Lygiagrečiojo užduočių planavimo modelio kodą galima pamatyti 3 priede.

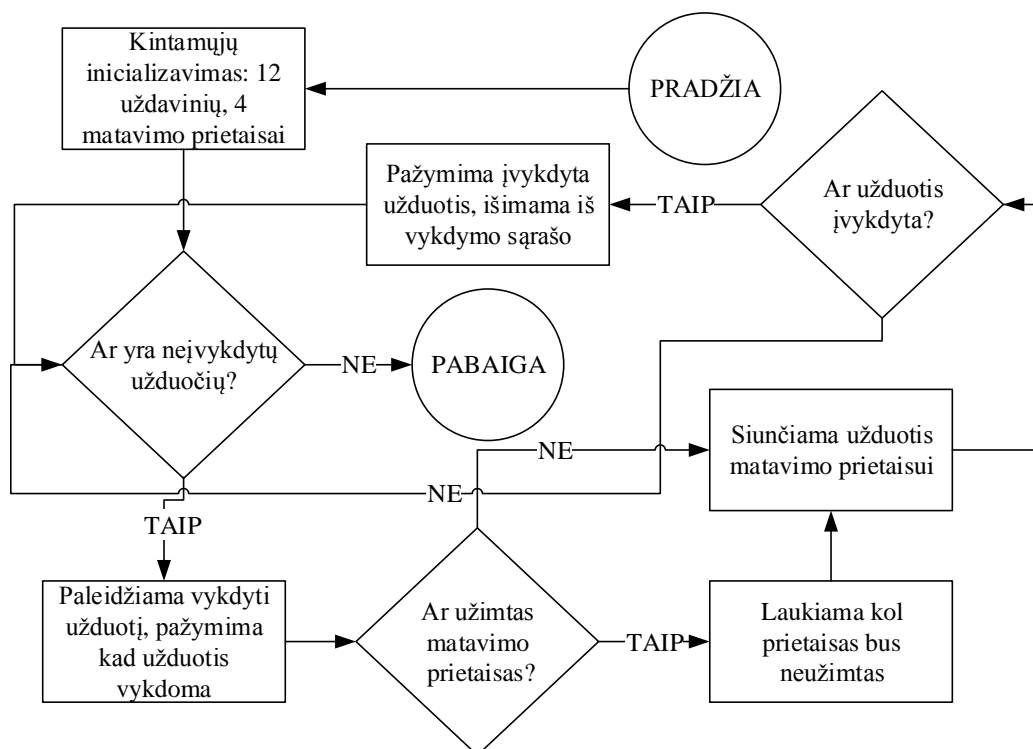
Lygiagretūs užduočių planavimo metodas tinkamesnis taikyti tada, kada vykdomi atskiri matavimo procesai arba matavimo rezultatai nėra susiję tarpusavyje. Matavimo scenarijaus vykdymo metu matavimo prietaisai valdomi lygiagrečiai. Taikant šį metodą matavimo scenarijuje rezultatai kaupiami ir apdorojami atskiruose duomenų masyvuose. Lygiagrečiojo užduočių planavimo modelio kodą galima pamatyti 4 priede.

Taikant nuoseklų ir lygiagretų užduočių planavimo metodus MATLAB programoje sumodeliuojamas automatizuotų matavimų programinės įrangos modelis. Modelis atspindi programinę įrangą palaikančio procesoriaus darbą, užduočių vykdymo metu. Modelyje aprašyti matavimų prietaisų komutacinės ir užduočių vykdymo laikų intervalai. Remiantis matavimo prietaisų ir matavimo procesą vykdančio kompiuterio charakteristikomis yra atvaizduojamas realaus laiko programos teorinis modelis (žr. 19, 20 pav.).



19 pav. Nuoseklus užduočių planavimo modelio algoritmas

Nuosekliame užduočių planavimo modelio algoritme užduotys vykdomos nuoseklia tvarka. Kita užduotis nebus vykdoma tol, kol nebus įvykdyta pirminė užduotis. Lygiagretusis modelis turi galimybę užduoties vykdymo metu įterpti kitą matavimo prietaiso užduotį. Jeigu visi matavimo prietaisai užimti, tai sekanti užduotis nebus vykdoma.



20 pav. Lygiagretusis užduočių planavimo modelio algoritmas

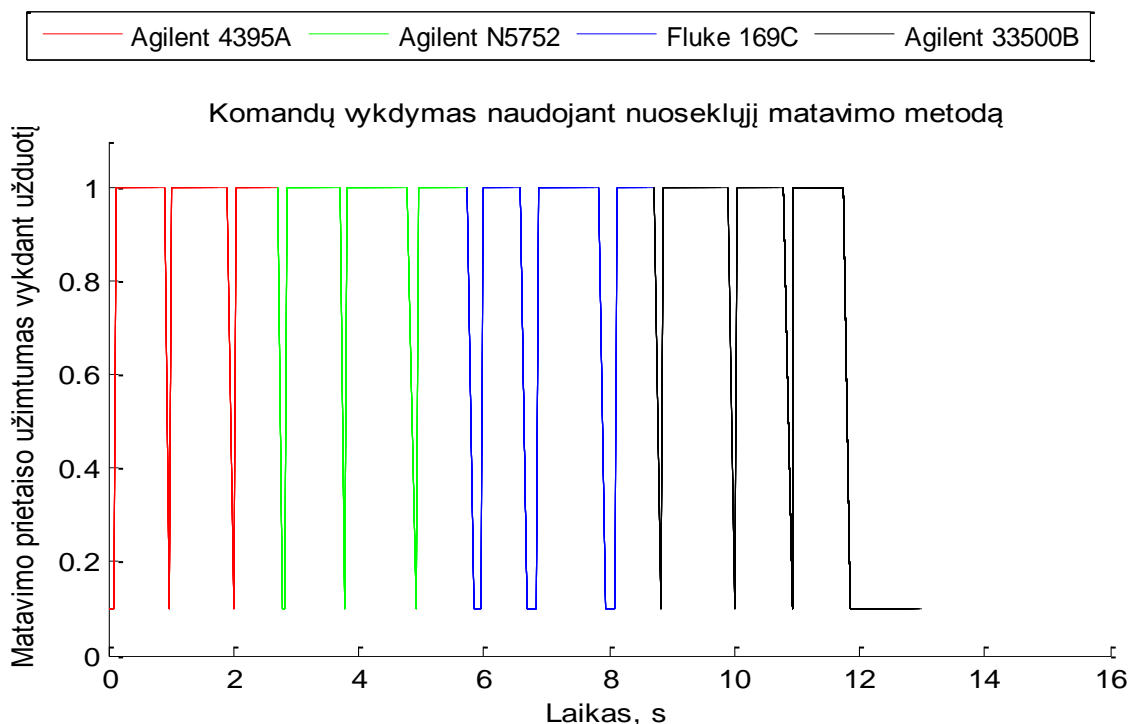
Ištestuoti modeliui, reikalingi matavimų prietaisų užduočių įvykdymo ir susijungimo su prietaisais komutacijos laiko intervalai. Laikams išmatuoti naudojami matavimo prietaisų užduočių ir komutacijos laikų nustatymo metodai. Atliekami eksperimentai su matavimo prietaisais. Rezultatai pavaizduoti 3 lentelėje.

3 lentelė

**Matavimo prietaisų užduočių ir komutacijos laikai**

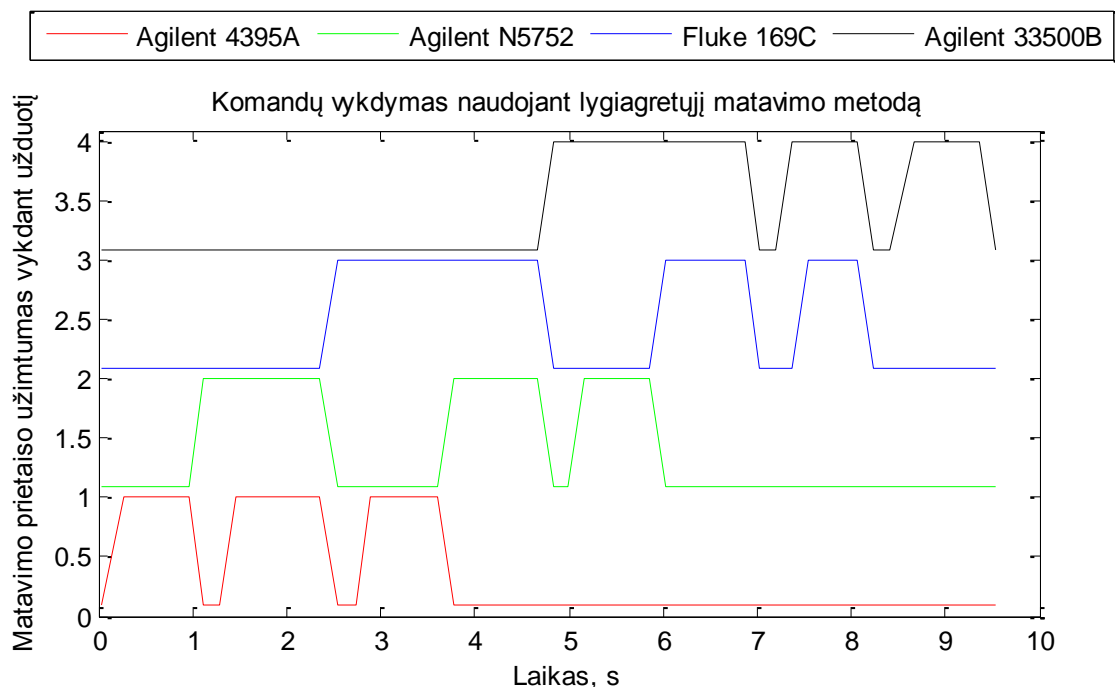
Matavimo prietaisas	1 užduoties įvykdymo laikas, s	2 užduoties įvykdymo laikas, s	3 užduoties įvykdymo laikas, s	Matavimo prietaiso komutacijos laikas, s
„Agilent 4395A“	0,711	0,912	0,753	0,15
„Agilent N5752A“	0,801	1,002	0,723	0,1
„Fluke 196C“	0,570	0,950	0,586	0,3
„Agilent 33500B“	1,125	0,763	0,830	0,1

3 lentelės užduočių įvykdymo ir komutacijos laikai panaudojami programinės įrangos teoriniame modelyje. Lentelės duomenys suvedami į „MATLAB“ automatizuotą matavimų programinės įrangos teorinį modelį. Teorinį modelį sudaro du algoritmai. Pirmas teorinio modelio algoritmas yra skirtas atvaizduoti nuoseklų užduočių planavimo metodą, antrasis - skirtas lygiagrečiam užduočių planavimo metodui. Modelio nuosekliojo užduočių planavimo rezultatai pavaizduoti diagramoje (žr. 20 pav.).



20 pav. Nuoseklusis užduočių planavimo teorinis modelis

Nagrinėjant diagramą matyti, kad prietaisų užduotys vykdomos nuosekliai. Diagramos spalvos nurodo matavimo prietaisą, o aukštas signalo lygis parodo teorinį prietaiso užimtumą. Taikant lygiagretųjį užduočių planavimo metodą teoriniame modelyje užduotys vykdomos lygiagrečiai (žr. 21 pav.)



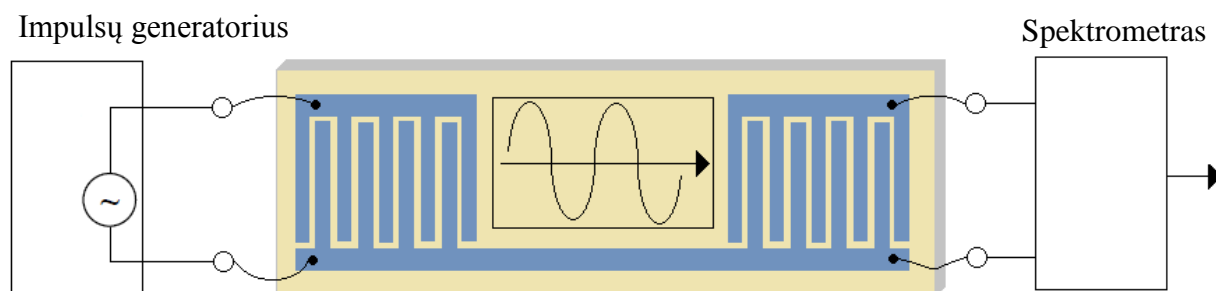
21 pav. Lygiagretusis užduočių planavimo teorinis modelis

Analizuojant teorinio modelio diagramą pastebima, kad užduotys planuojamos lygiagrečiai dėl skirtingų matavimo prietaisų užimtumo. Nors matavimų prietaisų užduočių vykdymo ir komutaciniai laikai yra vienodi, kaip nuosekliajame užduočių planavimo modelyje, tačiau lygiagrečiajame modelyje kai kurios užduotys vykdomos ilgiau. Šis reiškinys atsiranda dėl matavimo prietaisų prioritetinio valdymo. Matavimo prietaisui, turinčiam aukštesnį prioritetą, leidžiama pirmiau grąžinti matavimo rezultatus matavimo procesą vykdančiam scenarijui, nei žemesnį prioritetą turinčiam prietaisui.

## 2.5 Ultragarso jutiklio matavimai naudojant automatizuotą matavimų programinę įrangą

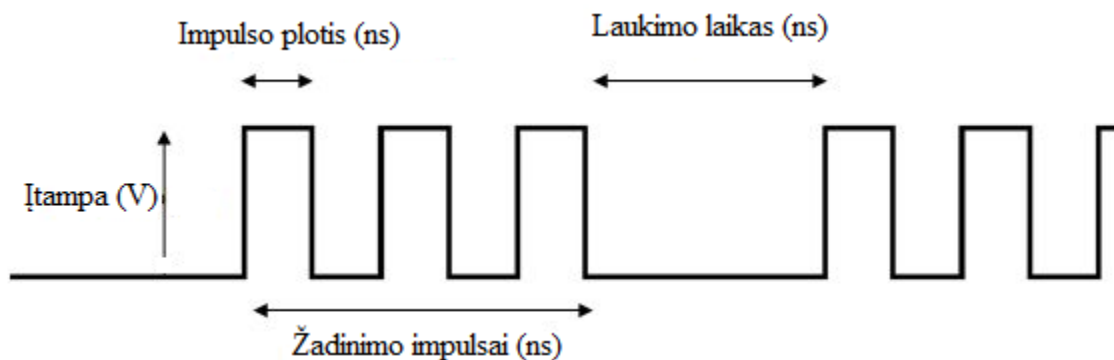
Talpiniai mikromontuojami ultragarso keitikliai CMUT (capacitive micro machined ultrasound transducers – angl.) buvo sukurti Stanfordo universitete 1994-1996 metais. Jie sudaryti iš kelių šimtų ar tūkstančių mikronų eilės talpinių elementų, sujungtų lygiagrečiai. Šie elementai dirba elektrostatiniame lauke kaip vienas keitiklis (žr. 22 pav.) [14].

CMUT tipo keitikliai gali būti naudojami ultragarsiniuose skaneriuose jutiklių kūrimui. Mažai tirta sritis yra paviršinių akustinių bangų, sklindančių kieto kūno ir dujinės aplinkos sandūroje bei Scholte bangų, sužadinamų kieto kūno ir skysčio sandūroje, įtaisai [15].



22 Pav. CMUT paviršinių bangų keitiklis

Tokios struktūros konstruojamos kaip užlaikymo linijos su atskirais žadinančių ir priimančių ultragarsinę bangą keitiklių masyvais. Keitiklių žadinimui dažniausiai naudojami harmoniniai virpesiai. CMUT keitikliams reikalingas tik vienas paviršinis elektrodas. Tai leidžia plokštumoje išdėstyti du „šukų“ formos keitiklius taip, kad tarp jų būtų  $90^\circ$  laipsnių fazės poslinkis ( $1/4$  bangos ilgio), pasirinktam bangos sklidimo greičiui. Tai leidžia žadinti keitiklius dviem signalais tarp kurių yra  $90^\circ$  fazės poslinkis. Ši sąlyga tiek priimančiame, tiek perduodančiame keitiklyje leidžia gauti iki dviejų kartų didesnę signalo amplitudę ir geresnį signalo kryptingumą. Keitiklio žadinimui naudojamas impulsų generatoriaus „Agilent 33500B“, generuojami žadinimo impulsai (žr. 23 pav.) [15, 16].



23 pav. CMUT keitiklio žadinimas

Impulsams sužadinus keitiklio membranas (siųstuvui), keitiklio paviršiumi bangos sklinda į visas puses. Kitoje keitiklio pusėje esančiose membranose (imtuve) susidaro Scholte bangos poveikio virpesiai, kurie mechaniškai išjudina membranas. Imtuvo išvadose susiformuoja įtampos pavidalo impulsai, kurie priimami su spektrometru „Fluke 196C“.

## 2.6 Skyriaus išvados

1. Suprogramuota automatizuotų matavimų programinė įranga. Programinę įrangą sudaro šie algoritmai:
  - Matavimo algoritmą aprašančio scenarijaus interpretatorius;
  - Užduotis valdantis algoritmas;
  - Matavimo prietaisų valdymo algoritmas.
2. Matavimo algoritmą aprašančio scenarijaus interpretatorius priima užduotį iš scenarijaus vykdymo programos ir interpretuoja automatizuotų matavimų programinei įrangai suprantančiomis komandomis.
3. Užduotis valdantis algoritmas valdo automatizuotą matavimo procesą nuosekliai arba lygiagrečiai užduočių planavimo principu.
4. Matavimo prietaisų valdymo algoritmas planuoja matavimo prietaisų darbą ir valdo naudodamasis virtualiais prietaisais.
5. Išmatuoti matavimo prietaisų užduočių atlikimo ir komutacijos su matavimo prietaisais laikai.
6. Sukurtas ir susimuliuotas automatizuotų matavimų programinės įrangos nuoseklusis ir lygiagretusis užduočių planavimo teorinis modelis. Modelis išbandytas su išmatuotų prietaisų užduočių atlikimo ir komutacijos laikais.
7. Išanalizuotas CMUT paviršinių bangų keitiklio veikimo principas. Nustatyta, kad keitiklio įėjimas žadinamas įtampos impulsais, o keitiklio išėjime matuojama „Scholte“ banga, kuri atvaizduojama laikinėje formoje.

### 3. TIRIAMOJI DALIS

#### 3.1 Eksperimentinių tyrimų priemonės

Eksperimentų tikslas – ištirti sukurtą automatizuotų matavimų programinę įrangą, palyginti eksperimentinius duomenis su analoginių priemonių duomenimis, taikant tuos pačius matavimo metodus.

Tyrimui atlikti naudojamos priemonės: asmeninis kompiuteris, sukurta programinė įranga, skirta automatizuotiems matavimams, „MATLAB“ programa ir matavimo prietaisai:

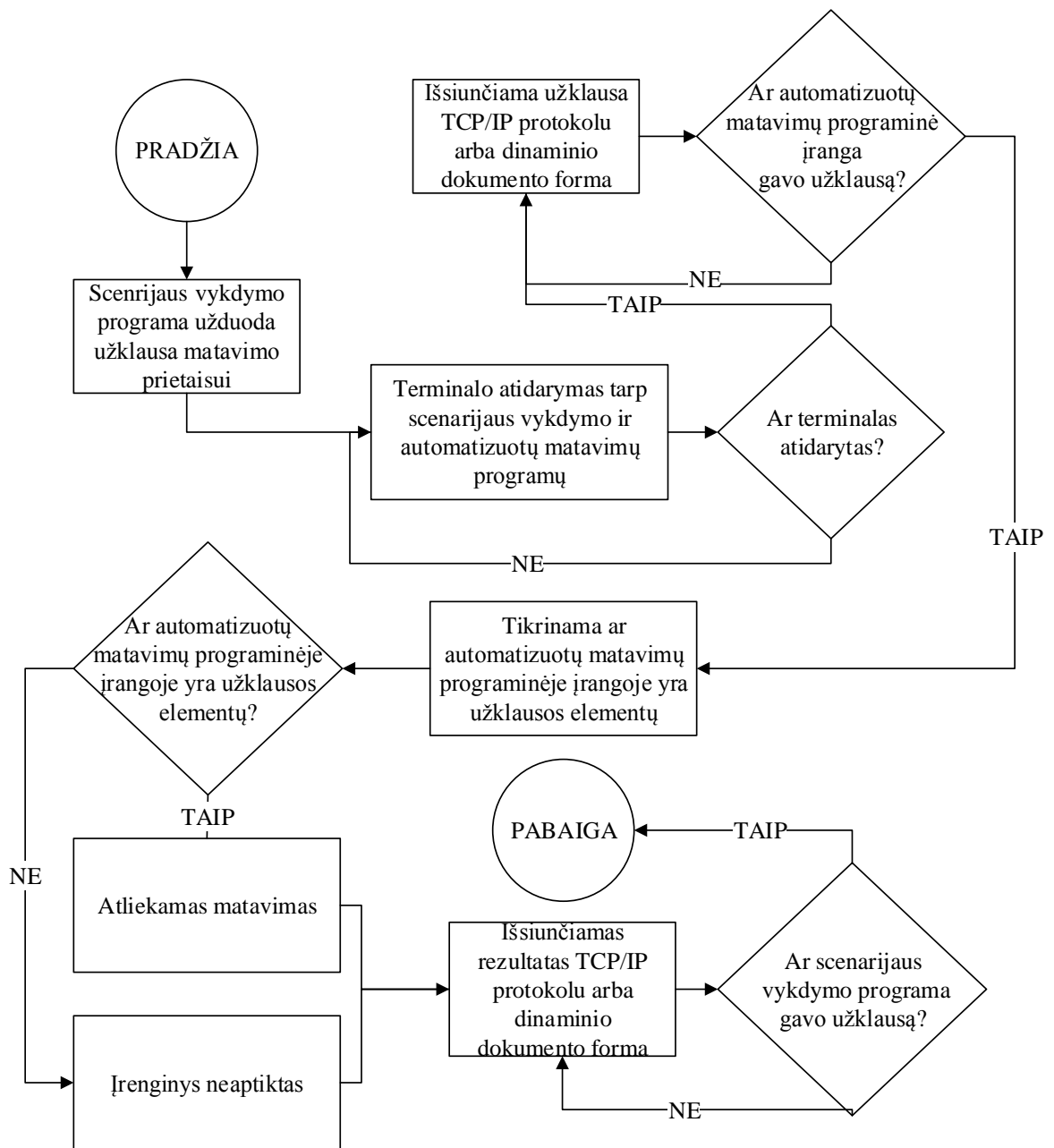
1. „Agilent a4395“ grandinių analizatorius. Matavimo prietaisas valdomas GPIB - USB magistrale. Prietaisas valdomas GPIB komandomis;
2. „Agilent N5752A“ įtampos šaltinis. Prietaisas valdomas per TCP/IP protokolą. Prietaisas valdomas SCPI komandomis;
3. „Fluke 196C“ spektrometras. Komunikavimo standartas RS-232. Prietaisas valdomas ASCII komandiniu rinkiniu;
4. „Agilent 33500B“ signalų generatorius. Prietaisas valdomas per TCP/IP protokolą. Prietaisas valdomas SCPI komandomis.

Tyrimai susiejami su eksperimento metu vykdomais matavimas. Eksperimentams taikomas ciklinio matavimo metodas. Šis metodas taikomas dėl to, kad dažniausiai yra sutinkamas automatizuotuose matavimuose. Eksperimente šis metodas taikomas norint išmatuoti prietaiso išnaudojimo intervalus laiko atžvilgiu. Kuo matavimo prietaisai funkcionaliau išnaudojami matavime, tuo trumpėja vieno ciklo matavimo laiko tarpas.

#### 3.2 Realus laiko duomenų perdavimas tarp MATLAB ir LabVIEW tyrimas

Analitinėje dalyje, komunikacinės duomenų valdymo skyriuje, nagrinėjami komunikaciniai metodai, kuriuos galima naudoti duomenų keitimuisi tarp „MATLAB“ ir „LabVIEW“. Parenkant tinkamiausią metodą yra reikalinga atlikti eksperimentinį tyrimą. Tyrimui atlikti pasirinktas TCP/IP duomenų perdavimo protokolų rinkinys ir duomenų perdavimas taikant dokumentus.

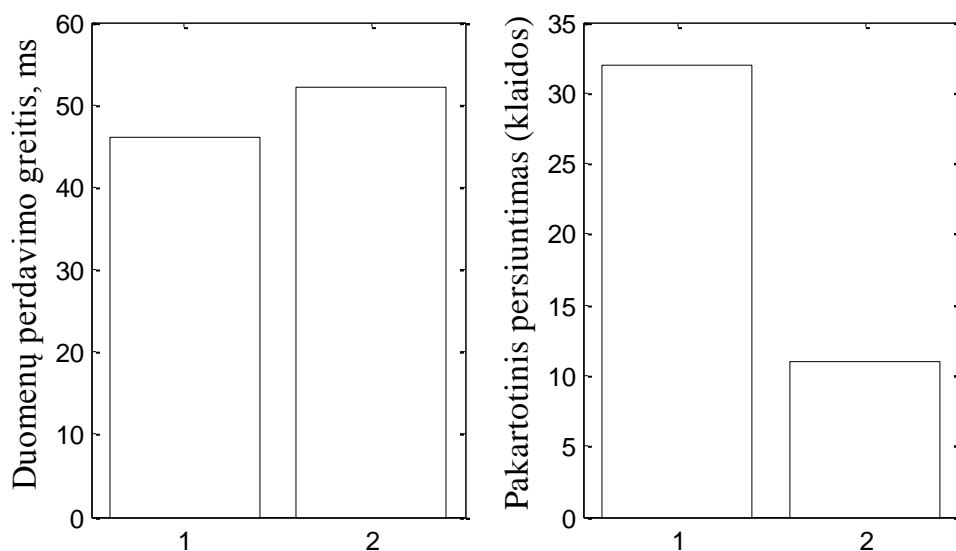
Eksperimentui atlikti parašomas kodas, kuriame aprašyti šie duomenų perdavimo metodai. Kodą galima rasti 5 priede. Tam, kad lengviau būtų interpretuoti kodo algoritmą, kodas išreiškimas blokine diagrama (žr. 24 pav.).



24 pav. Duomenų perdavimas tarp AMP ir scenarijus vykdymo programos

Eksperimente tiriama duomenų perdavimo greičiai ir informacinis patikimumas, naudojant TCP/IP protokolą ir dokumentinį duomenų perdavimą. Tyrimui realizuoti naudojamas 24 diagramos algoritmas. Eksperimentuojama su 16 KB duomenų masyvu, kuris buvo išsiųstas ir gautas 500 kartų. Matavimo metu registruojamos dvi diagramos. Pirmoje diagramoje atvaizduojami vidutiniai išsiųstų ir gautų duomenų greičiai. Antroje diagramoje atvaizduotos duomenų perdavimo metu įvykusios klaidos, kurioms reikėjo pakartotinio persiuntimo (žr. 25 pav.).





25 pav. Duomenų perdavimo laikai ir nesėkmingai perduoti duomenys. 1 – TCP/IP protokolas, 2 – dokumentinis duomenų perdavimas

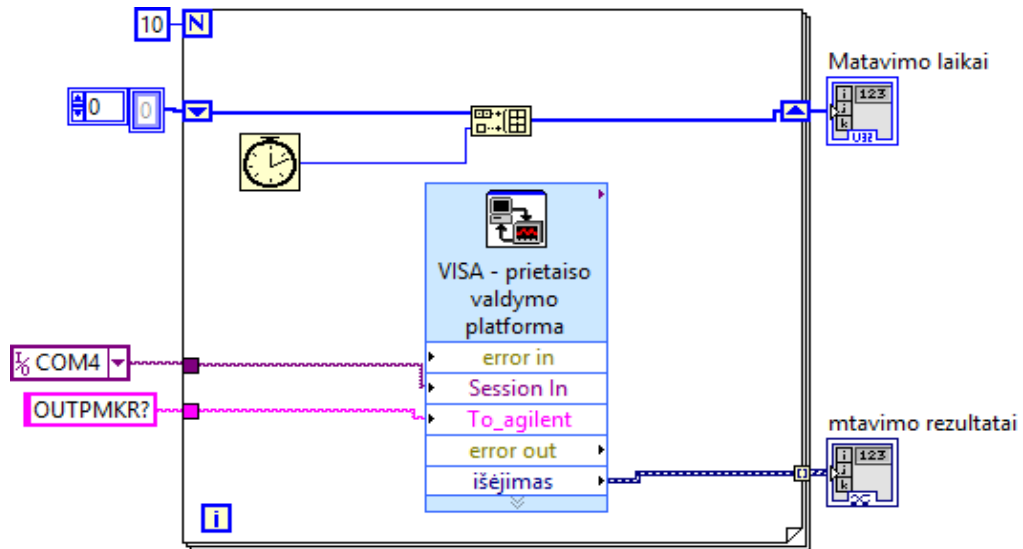
Analizuojant rezultatus galima daryti prielaidą, kad naudojant TCP/IP protokolą, duomenys išsiunčiami ir gaunami greičiau nei naudojant dokumentinį duomenų perdavimą. matuojant klaidų kiekį, kurios įvyko duomenų išsiuntimo ir gavimo metu, Žymiai mažiau klaidų sugeneruota taikant dokumentinį duomenų perdavimo metodą.

### 3.3 Virtualiųjų prietaisų (VISA) tyrimas

Eksperimento tikslas yra išsiaiškinti, koku greičiu veikia virtualiųjų prietaisų įrenginiai. Šio eksperimento rezultatai bus naudojami tolimesniame tyrime, taikant automatizuotų matavimų programines įrangas. Virtualūs prietaisai - tai programinė architektūra, kuri valdo komunikacines sąsajas. Šiame eksperimente lyginama „MATLAB“ naudojama virtualiųjų prietaisų valdymo architektūra (VISA) ir „National Instrument“ (VISA). Eksperimento rezultate turi atsispindėti duomenų perdavimo greičiai. Automatizuotuose matavimuose duomenys perduodami nuosekliai, todėl matavimuose atsiranda komutacinis laiko vėlinimas. Vėlinimas daro įtaką matavimo rezultatų fiksavimo laikui.

Eksperimentui naudojamas „Agilent 4395a“ grandinių analizatorius. Analizatorius priima ir perduoda duomenis per universalią nuosekliają magistralę (USB). Magistralės greitis įtakos atliekamam eksperimentui neturi. Eksperimento metu bus keičiami ir matuojami prietaiso parametrai. Eksperimento matavimo programinis kodas. Matavimo scenarijus bus įgyvendinimas su „MATLAB“ programine įranga ir „LabVIEW“ suprogramuota priemone, kurios naudoja atskiras VISA prietaisų valdymo architektūras.

- Naudojant „NI LabVIEW“ programinę priemonę, suprogramuojamas matavimo scenarijus (žr. 26 pav.).



26 pav. NI-VISA testavimo aplinka

Matavimo scenarijus tai pat suprogramuojamas ir „MATLAB“ sistemoje (žr. 27 pav.).

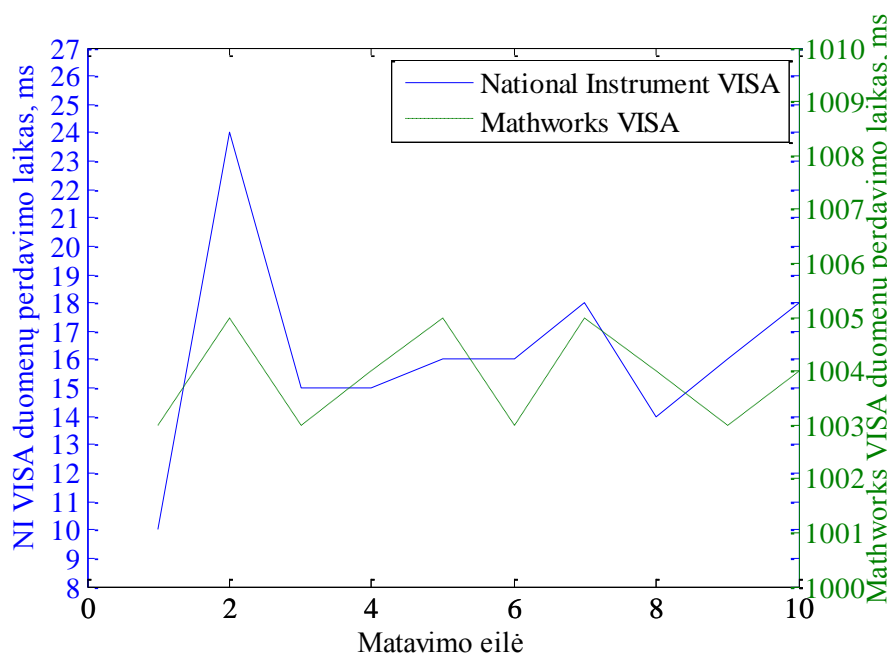
```

sport = serial('COM4');
sport.Terminator = 'CR';
%sport.Terminator = 'LF/CR';
sport.InputBufferSize=DBufSize;
sport.OutputBufferSize=DBufSize;
sport.BaudRate = 38400;
fopen(sport);
for i =1:10
fprintf(sport, 'OUTPMKR?');
yValues=fscanf(sport, '%c', DBufSize-dDBuf);
time = clock;
laikas(i) = time(6);
end
fclose(sport);

```

27 pav. „MATLAB VISA“ testavimo aplinka

Eksperimento metu buvo palygintos dvi programavimo sistemos, kurios turi galimybę valdyti prietaisus naudojant „VISA“ architektūrą. Eksperimento metu buvo matuojami virtualių prietaisų valdymo platformų programiniai komunikaciniai reiškiniai bei matuojami susijungimo ir atsijungimo laikai duomenų perdavimo metu. Matavimo prietaisai valdomi skirtingomis programinėmis platformomis: „MATLAB“ programa, kuri naudoja „MATLAB“ virtualiuosius prietaisus (VISA) ir „National Instruments LabVIEW“ programa, naudojanti NI – VISA (žr. 28 pav.).



28 pav. Virtualiųjų prietaisų tyrimas

Gauti rezultatai rodo, kad „National Instrument“ virtualiųjų prietaisų valdymo (VISA) architektūra greičiau perduoda duomenis nei „MATLAB“ naudojama „VISA“. Matavimo prielaida tokia, kad „MATLAB VISA“ architektūra veikia kitokiu principu nei „National instruments VISA“. Taip yra todėl, kad „MATLAB“ programą stabdo vidiniai programos procesai.

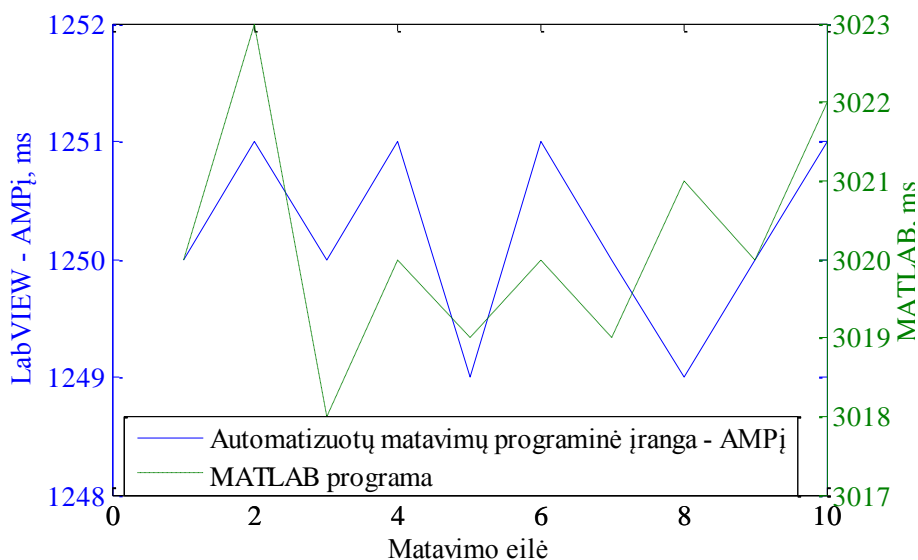
### 3.4 Automatizuotų matavimų programinės įrangos greičio tyrimas

Eksperimentas suskaidomas į atskirus du eksperimentus. Pirmasis eksperimentas bus atliekamas naudojant vieną matavimo prietaisą. Antroji eksperimento dalis bus vykdoma, naudojant kelis matavimo prietaisus, tik su automatizuotų matavimų programine įranga (AMPi).

#### 3.4.1 „MATLAB“ ir automatizuotų matavimų programinės įrangos greičio tyrimas

Pirmo eksperimento realizacijai naudojama „MATLAB“ programa. Programoje aprašomas scenarijus, kuris tiesiogiai valdo matavimo prietaisą per „MATLAB“ virtualiuosius prietaisus (VISA). Visa tai yra vadinama standartinė automatizuotų matavimų programine priemone. Eksperimentas yra apribojamas matavimo cikliniu kiekiu. Scenarijaus kodai pateikti 1, 5 priede. Eksperimentas yra atliekamas naudojant vieną matavimo prietaisą. Pasirinktas prietaisas yra „Agilent a4395“ grandinių analizatorius. Eksperimento metu prietaiso valdymas buvo vykdomas iš sukurtos programinės įrangos. Matavime naudojamos dvi funkcijos: be grįžtamųjų duomenų ir viena funkcija su grįžtamaisiais duomenimis. Analogiškai eksperimentas

atliekamas su standartine automatizuotų matavimų programine įranga. Rezultatai pavaizduoti 29 pav.



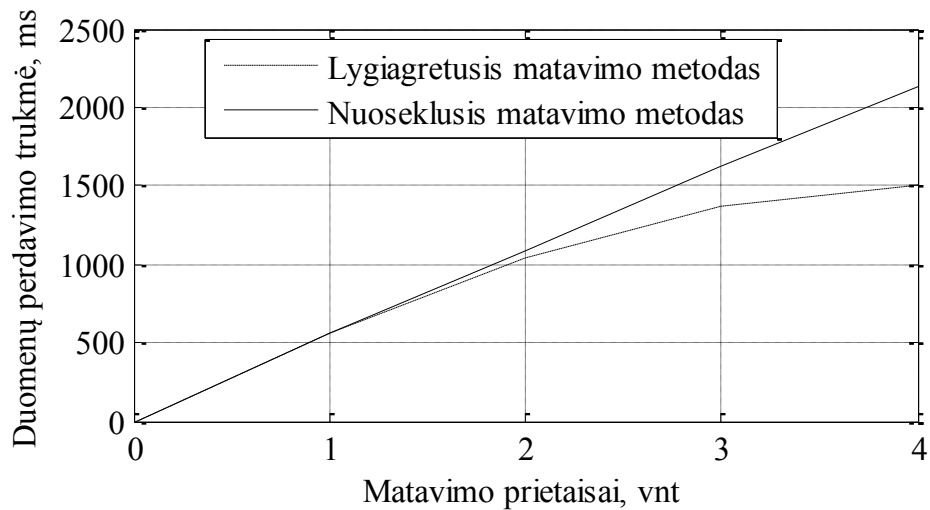
29 pav. Ciklinis matavimas naudojant vieną matavimo prietaisą

Interpretavus rezultatus galima daryti prielaidą, kad sukurtoje programinėje įrangoje matavimai atliekami greičiau. Standartinėje automatizuotų matavimų programinėje įrangoje matavimus vėlina „MATLAB“ programoje vykstantys komutaciniai reiškiniai.

### 3.4.2 Automatizuotų matavimų programinės įrangos nuosekliojo ir lygiagrečiojo užduočių planavimo metodo tyrimas

Eksperimento pirmojoje dalyje naudojami keli matavimo prietaisai: „Agilent 4395a“ grandinių analizatorių ir maitinimo šaltinį „Agilent N5752A“. Scenarijuje naudojamos 3 grandinių analizatoriaus funkcijos. Kitos 3 scenarijaus funkcijos yra skirtos „N5752A“ maitinimo šaltiniui. Analogiškai vykdomi antroji ir trečioji eksperimento dalys. Trečiam eksperimentui naudojami: „Agilent 4395a“, „Agilent N5752A“ ir „Fluke 196C“ matavimo prietaisai. Ketvirtame eksperimente naudojami: „Agilent 4395a“, „Agilent N5752A“, „Fluke 196C“ ir „Agilent 33500B“ signalų generatorius.

Eksperimentui galioja tas pats 10 ciklų scenarijus. Eksperimento metu tyrinėjami programinės įrangos AMPi nuoseklusis ir lygiagretusis užduočių planavimo metodai. Iš gautų rezultatų apskaičiuojamas matavimų vidurkis. Rezultatai pateikti laiko priklausomybės nuo prietaisų kiekio diagramoje (žr. 30 pav.).

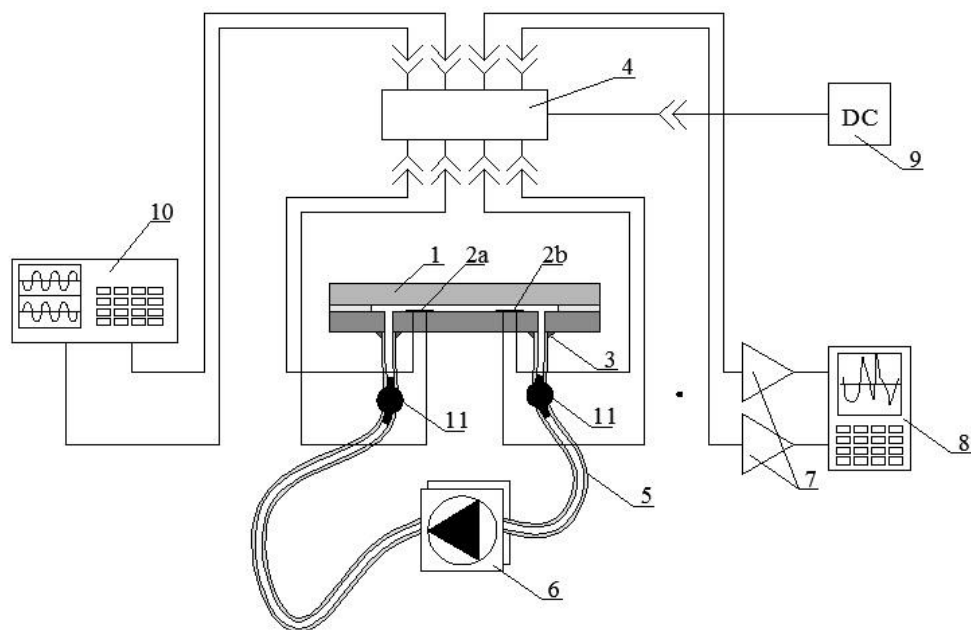


30 pav. Matavimo greitaveikos priklausomybė nuo prietaisų kiekio

Sukurtos programinės įrangos nuoseklusis užduočių planavimo metodas skiriasi nuo lygiagrečiojo. Nuosekliajame metode užduotys siunčiamos į prietaisus nuosekliai pagal matavimo scenarijų. Scenarijaus kodą galima rasti 7 priede. Scenarijaus sekantis veiksmas nėra vykdomas tol, kol matavimo scenarijus negaus atsakymo iš matavimo prietaiso. Taikant lygiagretųjį užduočių planavimo metodą (6 priedas), scenarijus taip pat yra siunčiamas nuosekliajame būdu, tačiau rezultatai matuojami tada, kada matavimo prietaisas juos išduoda. Lygiagrečiajame metode užduotis išsiunčiama ir nelaukiama, kol grįš atsakymas iš prietaiso. Algoritmas vykdo sekančią scenarijaus dalį, tai reiškia, kad yra siunčiama kita užklausa. Rezultatai palyginami su teorinio modelio rezultatais skyriuje - Eksperimentų rezultatų analizė.

### 3.5 Ultragarso jutiklio darbo taško paieška naudojant AMPį

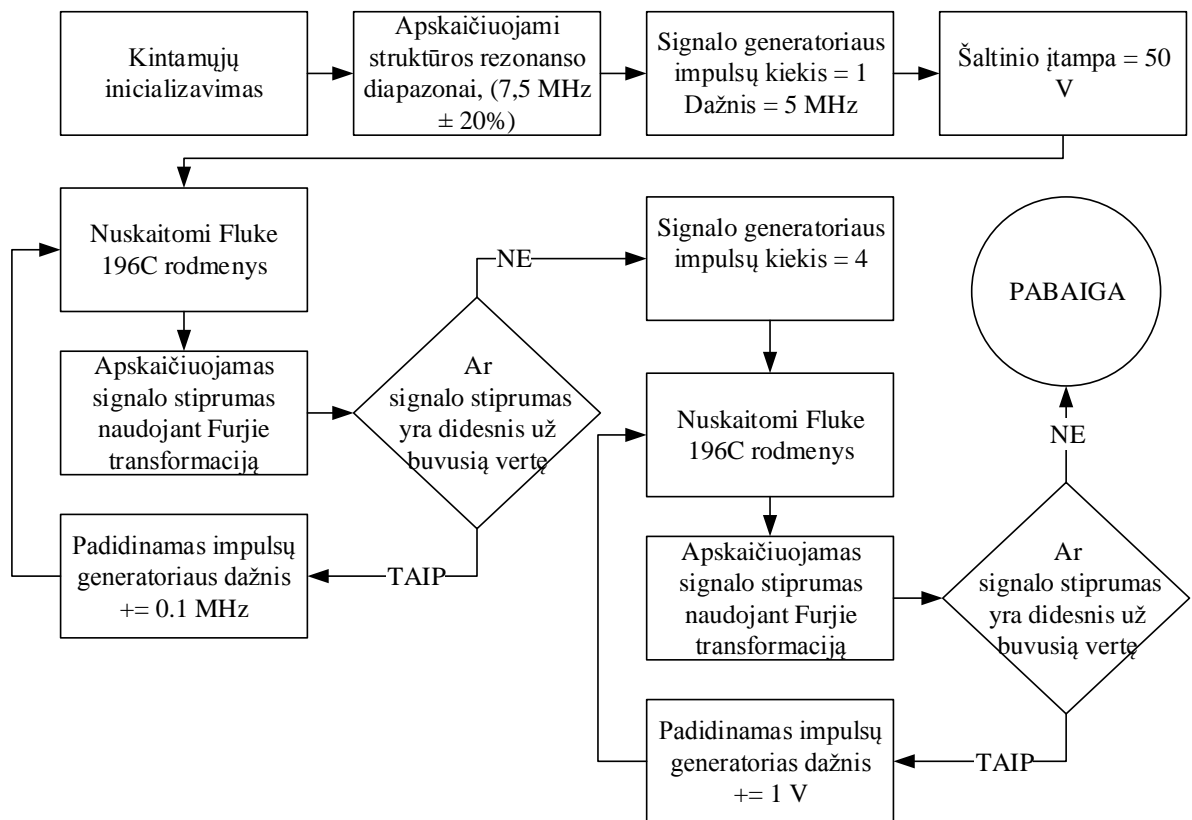
Ieškant ultragarso jutiklio darbo tašką eksperimente naudojami prietaisai: grandinių analizatorius „Agilent 4395A“, „Fluke 196C“ spektrometras, įtampos šaltinis „Agilent N5752A“ ir impulsų generatorius „Agilent 33500B“. Eksperimentui realizuoti naudojama 31 pav. pavaizduota schema.



31 pav. Matavimo struktūrinė schema. 1 – mikrokanalas, 2a – „šukų“ tipo siūstuvai, 2b – „šukų“ tipo imtuvai, 3 – įvadas į mikrokanalą, 4 – signalų atskyrimo įrenginys, 5 – Skysčio perdavimo terpės, 6 – Siurblys, 7 – signalo stiprintuvai, 8 – spektrometras „Fluke 196C“, 9 - įtampos šaltinis „N5752A“, 10 - impulsų generatorius „Agilent 33500B“

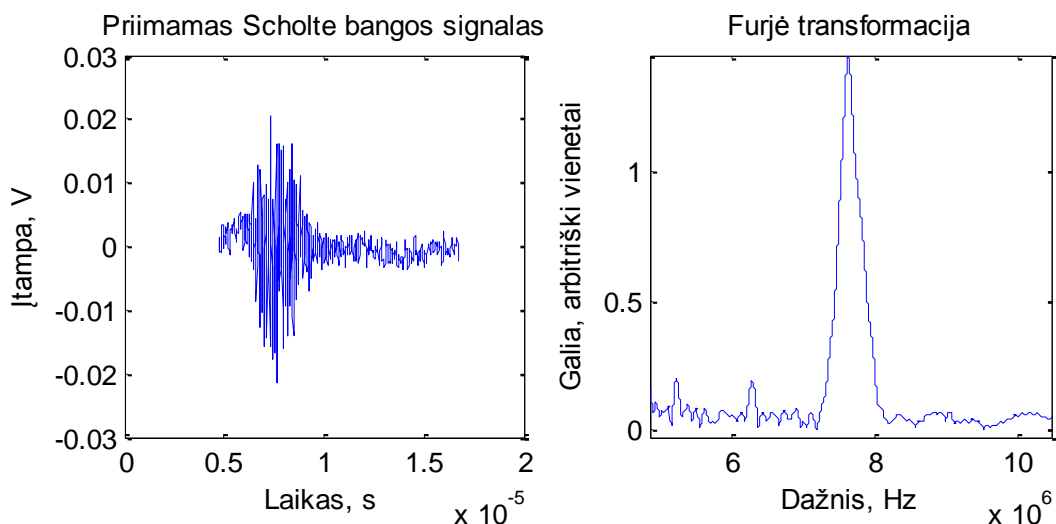
Signalas registruojamas spektrometre „Fluke 196C“, matuojant „šukų“ tipo imtuvo signalą. Analizuojant signalo rodmenis yra nagrinėjamos tyrinėjamos medžiagos savybės. Ultragarinio jutiklio darbo taško paieškai naudojamas spiritas. Priklausomai nuo maitinimo šaltinio įtampos, signalo generatoriaus impulsų žadinimo periodo, yra reguliuojamas signalo stiprumas. Keičiant maitinimo šaltinio įtampą ir impulsų generatoriaus periodą, signalas sustiprėja arba susilpnėja. Tikslas – taikant automatizuotą matavimų programinį kompleksą rasti ultragarinio jutiklio stipriausią signalą, kuris vadinamas darbo tašku.

Ieškant ultragarinio jutiklio darbo taško yra valdomi trys prietaisų parametrai: Maitinimo šaltinio „Agilent N5752A“ maitinimo įtampa, signalų generatoriaus impulsų kiekis ir žadinimo signalo periodas. Kad nebūtų sugadintas ultragarinis jutiklis, pagal jutiklio mechaninius duomenis yra parenkamos ribinės vertės. Jutiklis suprojektuotas 120 voltų nuolatinės srovės įtampai. Teoriškai jutiklio struktūros rezonansas yra apie 7,5 Mhz. Naudojantis įvardintais parametrais yra suprojektuojamas jutiklio darbo taško paieškos algoritmas, kuris pavaizduotas 32 pav. diagramoje.



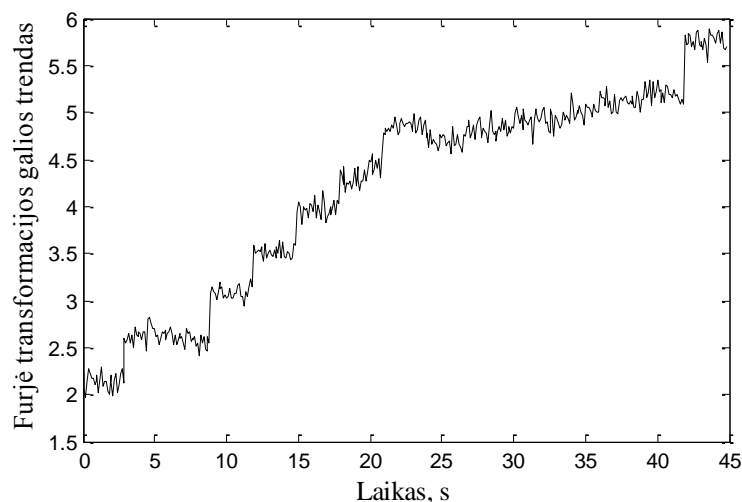
32 pav. Ultragarso jutiklio darbo taško paieškos algoritmas

Naudodamiesi 32 pav. esančiu algoritmu parašomas matavimo scenarijus. Nuskaitytus duomenis iš „Fluke 196C“ spektrometro, signalas yra perskaitytuojamas į dažninę charakteristiką naudojant „Furjė“ transformaciją. Signalas ir dažninė charakteristika pavaizduota 33 paveiksle.



33 pav. Matuojamo ultragarso jutiklio signalas ir dažnis

Matuojant dažninės charakteristikos maksimumą, yra ieškoma geriausios signalo kokybės. Signalų kokybę apibrėžiama galia. Kuo didesnė dažninės charakteristikos galia, tuo signalo kokybė yra geresnė (žr. 34 pav.).



34 pav. Ultragarinio jutiklio darbo taško paieška

Pateiktuose rezultatuose (žr. 34 pav.) matyti, kad automatizuotų matavimų programinės įrangos pagalba yra įvykdytas darbo taško paieškos algoritmas ir surastas ultragarinio jutiklio stipriausias signalas. Diagramos laiptuota charakteristika nuo 0 s iki 24 s yra dažnio didinimo pokytis. Nuo 24 s iki 41 s yra įtampos didinimas iki tol kol signalas nebedidėja. Nuo 41 sekundės ultragarinio jutiklio impulsų sustiprinimas iki 4 kartų. Scenarijaus kodas pateiktas 8 priede.

### 3.6 Eksperimentų rezultatų analizė

Tyrimo atlikimui buvo naudotasi skirtingais matavimo prietaisais su skirtingais komunikacinių sąsajų tipais. Eksperimente buvo matuojamas įrenginių aptarnavimo laikas nuo užklauso išsiuntimo iki duomenų gavimo. Užduočių skirstymo variantai iliustruoti 30 iliustracijoje. Standartiniame automatizuotame matavime duomenų siuntimas į matavimo prietaisus ir gavimas iš jų yra nuoseklus (žr. 35 pav.).

Laikas, ms	648	1820	352	812	1493	707	332	2202	556	1762	852	491
	Užd 1	Užd 2	Užd 3	Užd 4	Užd 5	Užd 6	Užd 7	Užd 8	Užd 9	Užd 10	Užd 11	Užd 12
Agilent 4395a	Užd 1	Užd 2	Užd 3									
Agilent N5752				Užd 4	Užd 5	Užd 6						
Fluke 196C							Užd 7	Užd 8	Užd 9			
Agilent 33500B										Užd 10	Užd 11	Užd 12

35 pav. Nuoseklus užduočių planavimas

Sukurtos automatizuotų matavimų programinės įrangos algoritmas valdo virtualių prietaisų valdymo duomenų išsiuntimo ir gavimo užduotis. Nuosekliajame matavime užduočių priėmimas ir išsiuntimas nėra valdomas. Naudojant „MATLAB VISA“ keturių prietaisų architektūrą, vieno ciklo matavimas siekia apie 11 sekundžių. Naudojant „NI-VISA“ ir sukurtą programinę įrangą 4 prietaisų matavimo ciklas trunka apie 1,5 sekundės (žr. 36 pav.).



Agilent 4395a	Užd 1 (648 ms)	Užd 2 (1820 ms)		Užd 3 (312ms)		
Agilent N5752		Užd4 (812 ms)	Užd 5 (1493ms)		Užd 6 (707 ms)	
Fluke 196C			Užd 7 (332 ms)	Užd 8 (2202 ms)		Užd 9 (556 ms)
Agilent 33500B				Užd 10 (1762 ms)	Užd 11 (852 ms)	Užd 12 (491 ms)

36 pav. Lygiagretus užduočių skirstymas

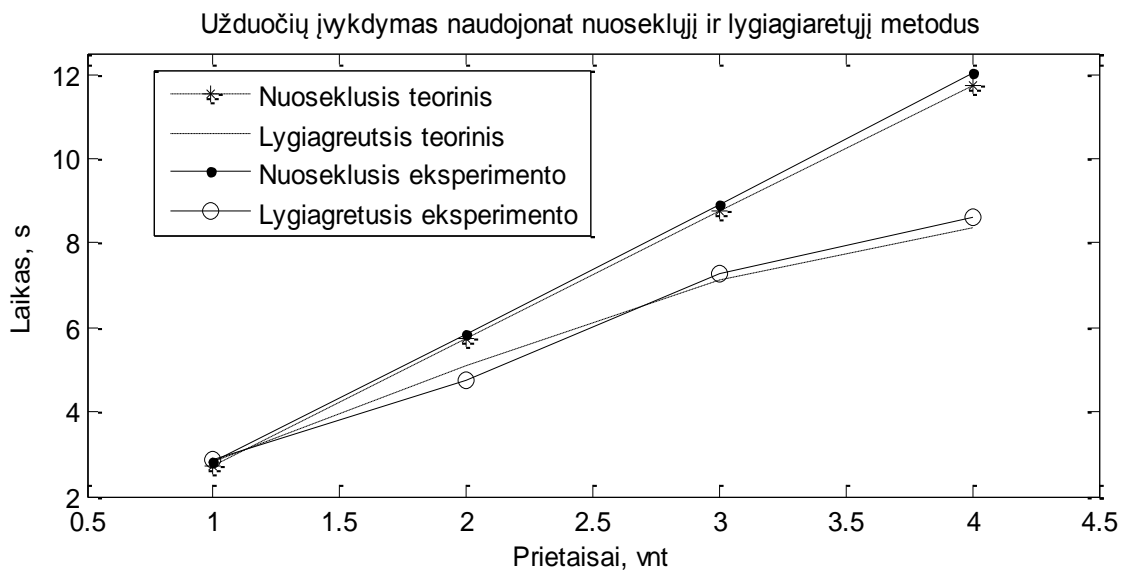
Prietaisai yra valdomi komandomis, kurios siunčiamos matavime dalyvaujančiam prietaisui. Prietaisų valdymo komandos saugomos stacionariame duomenų formate. Kiekviena siunčiama užduotis turi savo siunčiamos komandos vertę. Užduočių reikšmės aprašytos 4 lentelėje.

4 lentelė

#### Prietaisų užduočių paaiškinimas

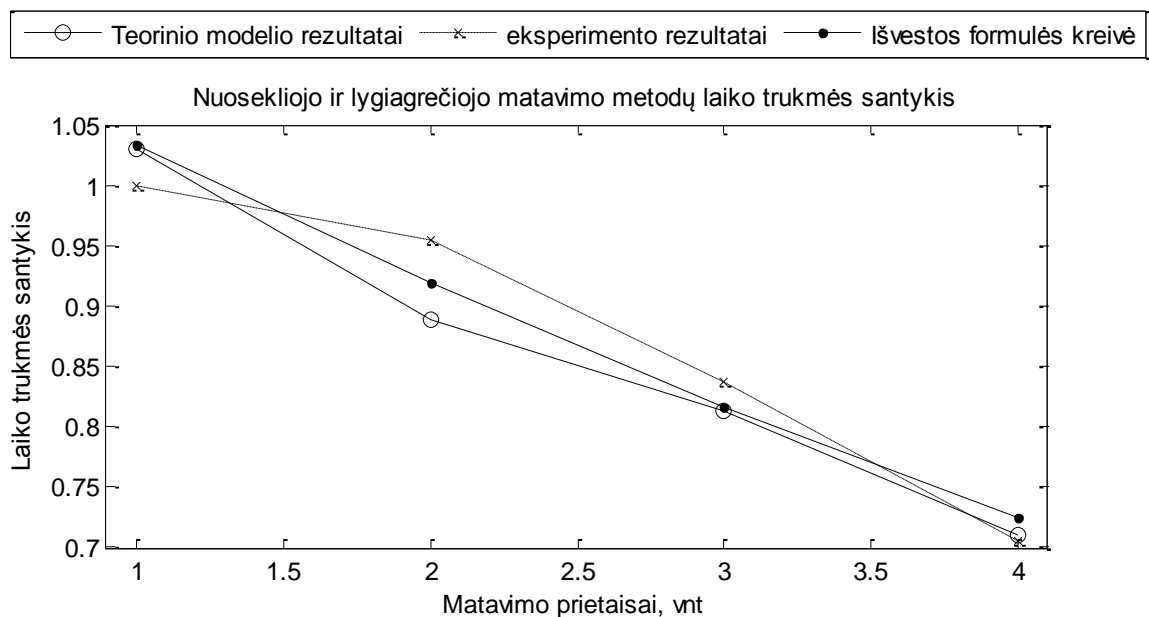
Užduotis	Prietaisas	Komandos tipas	Reikšmė prietaise
Užduotis 1	<b>Agilent 4395a</b>	Be grįžtamųjų duomenų	Prietaiso dažnio nustatymas
Užduotis 2		Su grįžtamaisiais duomenimis	Duomenų nuskaitymas
Užduotis 3		Be grįžtamųjų duomenų	Prietaiso pradinių parametru atstatymas
Užduotis 4	<b>Agilent N5752</b>	Be grįžtamųjų duomenų	Įtampos nustatymas
Užduotis 5		Be grįžtamųjų duomenų	Ribinės srovės nustatymas
Užduotis 6		Su grįžtamaisiais duomenimis	Prietaiso parametru nuskaitymas
Užduotis 7	<b>Fluke 196C</b>	Be grįžtamųjų duomenų	Magistralės greičio nustatymas
Užduotis 8		Su grįžtamaisiais duomenimis	Prietaiso matavimo nuskaitymas
Užduotis 9		Be grįžtamųjų duomenų	Prietaiso kanalo nustatymas
Užduotis 10	<b>Agilent 33500B</b>	Su grįžtamaisiais duomenimis	Prietaiso parametru paketo nusiuntimas
Užduotis 11		Be grįžtamųjų duomenų	1 Kanalo įjungimas
Užduotis 12		Be grįžtamųjų duomenų	1 Kanalo dažnio parametro nustatymas

Buvo atlikti teoriniai ir praktiniai eksperimentai, kuriuose tyrinėjami nuosekliojo ir lygiagrečiojo užduočių planavimo metodai. Teorinio modelio rezultatai paimti iš temos „Programinės įrangos teorinis modelis“. Rezultatai pateikti diagramoje (žr. 37 pav.). Tiek teorinio, tiek praktinio tyrimo rezultate matomas vizualinis atitikimas. Nuosekliojo užduočių planavimo teorinio modelio ir eksperimentinių rezultatų nuokrypis laiko atžvilgiu siekia 0,094 %, lygiagrečiajame – 2,16 % laiko atžvilgiu.



37 pav. Nuosekliojo ir lygiagrečiojo užduočių planavimo metodų eksperimento ir teorinio modelio rezultatai

Iš rastųjų teorinio modelio ir eksperimento užduočių planavimo laiko trukmės tarp matavimo prietaiso rezultatų, apskaičiuojamas santykis tarp nuosekliojo ir lygiagrečiojo metodų. Diagrama pateikta 38 pav.



38 pav. Nuosekliojo ir lygiagrečiojo matavimo metodų laiko trukmės santykis

Iš gautų nuosekliojo ir lygiagrečiaus matavimo metodų rezultatų, apskaičiuojamas laiko trukmės santykis tarp matavimo prietaisų kiekio. Pagal eksponentinį pasiskirstymo dėsnį išvedama formulė:

$$s_{pr} = 1,15 \cdot \exp(-0,12 \cdot pr), \quad (3)$$

čia:  $pr$  – matavimo prietaisų kiekis, vnt;

Taikant nuoseklųjį matavimo metodą, matavimo prietaisų panaudojimas scenarijaus vykdyme proporcingai priklausomas nuo užduočių vykdymo trukmės. Scenarijaus vykdymo metu sukurta programinė įranga, kuri naudoja lygiagretųjį matavimo metodą, valdo kelis prietaisus lygiagrečiai. Eksperimentų rezultatuose matyti, kad matavimo prietaisų užduočių vykdymo trukmė yra vienoda tol, kol yra naudojamas vienas ar du matavimo prietaisai.

### 3.7 Skyriaus išvados

1. Ištirtas duomenų perdavimo greitis ir duomenų perdavimo patikimumas tarp scenarijaus vykdymo programos „MATLAB“ ir automatizuotų matavimų programinės įrangos. Taikant TCP/IP protokolo metodą, 14 ms duomenys perduodami greičiau, nei naudojant dokumentinį duomenų perdavimo metodą. Naudojant dokumentinį duomenų perdavimo metodą, duomenų persiuntimo metu, užfiksuota 3 kartus mažiau klaidų nei naudojant TCP/IP metodą.
2. Ištirti „Mathworks“ ir „National instruments“ virtualūs prietaisai (VISA). Tyrimo rezultatuose matyti, kad 987 ms greičiau su užduotimis susidorojo „National Instruments“ virtualūs prietaisai.
3. Ištirta automatizuotų matavimų programinė įranga, ultragarsinio paviršinių bangų keitiklio darbo taško paieškos eksperimente. Ultragarsinio keitiklio darbo tašką aptiko per 45 sekundes. Paieška vyko lėtai dėl signalo spektro nuskaitymo ir konvertavimo į „Furjė“ transformaciją.
4. Ištirti automatizuotų matavimų programinės įrangos užduočių planavimo metodai ir palyginti su modeliuotu teoriniu modeliu. Programinės įrangos eksperimento rezultatai atitinka teorinio modelio rezultatus.
5. Išvesta formulė, pagal kurią galima apskaičiuoti nuosekliojo ir lygiagrečiojo užduočių planavimo metodų santykį panaudojant matavimo skirtingą matavimo prietaisų kiekį.

## IŠVADOS

1. Analitinėje dalyje išnagrinėtos automatizuotų matavimų programinės platformos. Populiariausios naudojamos programinės platformos automatizuotiems matavimams: „Vartotojui adaptuota programinė įranga“, „Kalbų sąsajos programinė įranga“, „Praplėtimo programinė įranga“ ir „Virtualių prietaisų programinė įranga“. Lanksčiausios – „Virtualių prietaisų (VISA)“ ir „Praplėtimo programinės įrangos“. Analizuojant automatizuotų matavimo platformas pastebėta, jei programinė platforma yra lanksti, tai jos adaptavimas automatizuotiems matavimams yra sudėtingas.
2. Automatizuotų matavimų trūkumų sprendimams pasiūlytas automatizuotų matavimų programinės įrangos prototipas. Trūkumams išspręsti parinkti metodai. Duomenų perdavimui tarp scenarijaus vykdymo programos ir automatizuotų matavimų programinės įrangos metodas. Matavimo proceso valdymas nuoseklia ir lygiagrečiuoju užduočių planavimo metodais, išsprendžiama automatizuotų matavimų lankstumą ir automatizavimo sudėtingumą. Virtualiaisiais prietaisais išsprendžiamas matavimų prietaisų valdymas realia laike.
3. Suprogramuota automatizuotų matavimų programinė įranga (AMP). Programa suprojektuota taip, kad vartotojas lengva forma galėtų suprojektuoti scenarijaus algoritmą automatizuotam matavimo procesui. AMP pagreitina programinį pasiruošimą automatizuotiems matavimams. Programinėje įrangoje naudoti TCP/IP ir dokumentinis duomenų perdavimo metodai, komunikacijai tarp AMP ir scenarijaus vykdymo programos. „National Instruments“ virtualiųjų prietaisų valdymui pritaikytas užduočių planavimo algoritmas, kuris nepriklausomai nuo operatoriaus valdo matavimo prietaisus.
4. Ištirtas duomenų perdavimo greitis ir patikimumas tarp scenarijaus vykdymo programos „MATLAB“ ir automatizuotų matavimų programinės įrangos. Taikant TCP/IP protokolo metodą, 14 ms duomenys perduodami greičiau nei naudojant dokumentinį duomenų perdavimo metodą. Taikant dokumentinį duomenų keitimosi metodą, duomenų persiuntimo metu, užfiksuota 3 kartus mažiau klaidų, nei naudojant TCP/IP metodą. Ištirti „MathWorks“ ir „National instruments“ kūrėjų virtualūs prietaisai (VISA). Rezultatai parodė, kad „National Instruments VISA“ 987 ms greičiau aptarnauja matavimo prietaisus. Ištirta automatizuotų matavimų programinė įranga, ultragarsinio paviršinių bangų keitiklio darbo taško paieškos eksperimente. Eksperimente buvo naudojami trys matavimo prietaisai. Ultragarsinio keitiklio darbo tašką aptiko per 45 sekundes. Paieška vyko lėtai dėl signalo spektro nuskaitymo ir konvertavimo į „Furjė“ transformaciją.

5. Ištirtas automatizuotų matavimų programinės įrangos užduočių planavimo metodas ir palygintas su modelio rezultatais. Nuosekliojo užduočių planavimo eksperimentinių rezultatų nuokrypis nuo teorinio modelio skiriasi 0,094 %, lygiagrečiojo – 2,16 %.

## LITERATŪROS SĄRAŠAS

1. **CHENG Qiong.** *Implementation of Virtual Laboratory of Power Electronics Based on Matlab and LABVIEW.* Computer Science & Education (ICCSE). 2012 p. 1539 – 1544, [žiūrėta 2014 m. balandžio 15 d.]. Prieiga per internetą:  
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6295356>>
2. **HOMAN Deborah.** *Accelerate Insstrument Control And Automation.* AUTOTESTCON, 2012 IEEE. 2012, p. 130-134 [žiūrėta 2014 m. balandžio 15 d.]. Prieiga per internetą:  
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6334573>>
3. **CACCIOTA M.; GIARNETTI S.; LECCESE F.; TRINCA D.** *A multi-platform Data Acquisition Device for Power Quality metrological certification.* Roma Tre. 2010, p. 404-407 [žiūrėta 2014 m. balandžio 2 d.]. Prieiga per internetą:  
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5489922>>
4. **BAZDRESCH Miguel.** *A real-time, matlab-based undergraduate Digital communications course.* Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE). 2011, p. 408-413 [žiūrėta 2014 m. balandžio 15 d.]. Prieiga per internetą:  
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5739249>>
5. **KAŽYS R; ŠILERIS R.** *Matavimų automatizavimas ir matavimo sistemos.* – Kaunas: Technologija, 2006. – 246 p. ISBN 978-609-02-0429-0
6. **RAJU Solomin; DEVI Lakshmi.** *Real-time Digital Simulation of Control System with LabVIEW Simulation Interface Toolkit.* Impact Journals. 2007. [žiūrėta 2014 m. balandžio 15 d.]. Prieiga per internetą:  
<<http://www.impactjournals.us/download.php?fname=-1385548220-3.%20Eng-Real-Time%20Digital%20Simulation-U%20Solomon%20Raju.pdf>>
7. **PUNK Oliver; HEUERT Uwe.** *Multiple-Interface Design for FPGA-Based Measurement Systems.* Electronics Technology (ISSE). 2011, p. 592-596 [žiūrėta 2014 m. balandžio 15 d.]. Prieiga per internetą:  
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6053956>>
8. **MOMOH-JIMOH E, TIJANI I.B.** *Development of Real-Time Software Interface for Multicomponent Transient Signal Analysis Using Labview and Matlab.* Mechatronics (ICOM). 2012, p. 1-5 [žiūrėta 2014 m. balandžio 15 d.]. Prieiga per internetą:  
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6053956>>
9. **YONG Xu; Jiam Li; JING Li.** *Cmmunication desigh for Vehicle Dynamic Data Exchange.* Information Engineering (ICIE). 2012, p. 356-359 . [žiūrėta 2014 m. balandžio 15 d.] prieiga per internetą: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5571584>>
10. **VISHNU R.** *An Automated Spurious Immunity Test Setup for Wideband Radio Receivers Based on LabVIEW.* Control Communication and Computing (ICCC). 2013, p. 420-425 [žiūrėta 2014 m. balandžio 15 d.]. Prieiga per internetą:  
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6731691>>

11. **KALANTZOPOULOS Athanasios**; ZIGOURIS Evangelos. *Real-Time DSP application remotely controlled through customized GUIs*. Education and Research Conference (EDERC). 2012, p. 71-75. [žiūrėta 2014 m. balandžio 15 d.] prieiga per internetą: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6532228>>
12. **FANG Zeping**. *Obstacle Detection for Intelligent Vehicle Based on LabVIEW and Laser Measurement System*. Information and Automation (ICIA). 2013, p. 202-206. [žiūrėta 2014 m. balandžio 15 d.] prieiga per internetą: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6731691>>
13. **ZHENG Kai**; YAO Yu; JIANG Yi, WANG Ruqi. *A New Real-time Multitasking Simulation Scheme Based on MATLAB RTW and RTX*. Intelligent Control and Information Processing (ICICIP), 2012 Third International Conference on. 2012, p. 185-190 [žiūrėta 2014 m. balandžio 15 d.]. Prieiga per internetą: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6391433>>
14. **SAPELIAUSKAS E**; **BARAUSKAS D**; **VANAGAS G**; **VIRZONIS D**. *Surface micromachined CMUTs for liquid phase sensing*. Ultrasonics Symposium (IUS), 2014 IEEE International, 2014. [žiūrėta 2015 m. gegužės 26 d.] prieiga per internetą: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6931951>>
15. **KAROSAS D**; **PELENIS D.**; **MIKUTA T**; **VIRŽONIS D**. *Komutatorius ultragarso keitiklių matavimų automatizavimui*. TELEKOMUNIKACIJOS IR ELEKTRONIKA – 2014., P. 31 – 33 [žiūrėta 2015 m. balandžio 15 d.] prieiga per internetą: <[http://ktu.edu/uploads/files/fakultetai /Elektros%20ir%20elektronikos%20fakultetas/files/TiE-2014.pdf](http://ktu.edu/uploads/files/fakultetai/Elektros%20ir%20elektronikos%20fakultetas/files/TiE-2014.pdf)>

# **PRIEDAI**