



**Kauno technologijos universitetas**

Panevėžio technologijų ir verslo fakultetas

# **Eterneto pramoninių valdymo tinklų topologijų tyrimas**

Baigiamasis magistro studijų projektas

---

**Tomas Nuobaras**

Projekto autorius

**dr. Donatas Pelenis**

Vadovas

---

**Panevėžys, 2021**



**Kauno technologijos universitetas**  
Panevėžio technologijų ir verslo fakultetas

## **Eterneto pramoninių valdymo tinklų topologijų tyrimas**

Baigiamasis magistro studijų projektas  
Valdymo technologijos (6211EX014)

---

**Tomas Nuobaras**  
Projekto autorius

**dr. Donatas Pelenis**  
Vadovas

Recenzentas / Recenzentė

---

**Panevėžys, 2021**



**Kauno technologijos universitetas**

Panevėžio technologijų ir verslo fakultetas

Tomas Nuobaras

## **Eterneto pramoninių valdymo tinklų topologijų tyrimas**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Tomas Nuobaras

*Patvirtinta elektroniniu būdu*



**Kauno technologijos universitetas**

Panevėžio technologijų ir verslo fakultetas

**TVIRTINU**

TVKC vadovė

Doc. dr. Nida Kvedaraitė

**Baigiamojo magistro projekto užduotis**

Diplomantui **Tomui Nuobarui**

---

Baigiamojo projekto tema Eterneto pramoninių valdymo tinklų topologijų tyrimas  
(lietuvių kalba)

---

Baigiamojo projekto tema Research of Industrial Ethernet Control Network Topologies  
(anglų kalba)

---

Patvirtinta 2021 m. balandžio 2 d. dekanı potvarkiu Nr. V25-13-13

Parengto baigiamojo projekto įkėlimo į Moodle aplinką terminas iki 2021 m. gegužės 31 d.

Duomenys, reikalavimai ir sąlygos baigiamajam projektui

---

Baigiamojo projekto užduotys / uždaviniai, kurie turi būti atskleisti projekte

---

Atlikti mokslinės literatūros analizę susijusią su realaus laiko komunikaciniais tinklais. Išskirti realaus laiko *Ethernet* šakos pramonės komunikacinių technologijų privalumus ir trūkumus.

---

Išanalizuoti naudojamas technologines priemones ir taikymo metodus. Sudaryti metodus tinkančius tirti EtherCAT ir PROFINET tinklų laikinius parametrus, atsižvelgiant į tinklų topologijas ir aptarnaujamų tinklo prietaisų skaičių Taikant MATLAB programinę įrangą ir sudarytus tyrimo metodus, ištirti EtherCAT realaus laiko komunikacinio tinklo greitaveiką ir vėlinimo laikus.

---

Pateikti rekomendacijas, leidžiančias efektyvinti *Ethernet* šakos pramonės komunikacinį tinklą.

---

Vadovas dr. Donatas Pelenis

---

(vadovo pareigos, vardas, pavardė)

Užduotį gavau Tomas Nuobaras

---

(studento vardas, pavardė)

2021 m. balandžio 16 d.

Nuobaras Tomas. Eterneto pramoninių valdymo tinklų topologijų tyrimas. Magistro baigiamasis projektas / vadovas lekt. dr. Donatas Pelenis; Kauno technologijos universitetas, Panevėžio technologijų ir verslo fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): elektronikos inžinerija, technologijos mokslai, (inžinerijos mokslai).

Reikšminiai žodžiai: realaus laiko valdymo tinklai; *EtherCAT*; *PROFINET*.

Panevėžys, 2021. 65 p.

## Santrauka

Šiame darbe tiriami realaus laiko *Ethernet* šakos *EtherCAT* ir *PROFINET* pramoniniai valdymo tinklai. Pramoninėse automatinio valdymo sistemose projektuojant dideles ir sudėtingas pramonines valdymo sistemas, labai svarbu užtikrinti kuo mažesnę maksimalų reakcijos laiką. Sudėtingose sistemose su dideliu kiekiu pozicionavimo įrenginių pvz. manipuliatorių, įvairių vykdyklių, esant per dideliu reakcijos laikui išskyla pozicionavimo tikslumo problemos, todėl projektuojant tokias sistemas svarbu žinoti kaip kinta reakcijos laikas skirtingose komunikacijų magistralėse priklausomai nuo naudojamų valdomų prietaisų (angl. *Slave Device*) kiekio, magistralės jungimo tipo ir duomenų perdavimo spartos.

Šio darbo mokslinės literatūros analizės dalyje analizuojami moksliniai šaltiniai, susiję su realaus laiko *EtherCAT* ir *PROFINET* valdymo tinklais. Remiantis moksliniuose straipsniuose rastais duomenimis aprašomi šių dviejų valdymo tinklų jungimo būdai, pateikiamos formulės charakterizuojančios tinklo greitaveiką, aprašoma kokių protokolų pagrindu veikia šie du valdymo tinklai. Aprašomas duomenų siuntimo pavaldiesiems prietaisams ir duomenų gavimo iš pavaldžių prietaisų algoritmas. Aprašomi šių dviejų tinklų panašumai ir skirtumai.

Šio darbo tiriamojoje dalyje sudaromi metodai tinkantys tirti *EtherCAT* ir *PROFINET* tinklų laikinius parametrus, atsižvelgiant į tinklų topologijas ir aptarnaujamų tinklo prietaisų skaičių. Naudojantis sudarytais metodais tiriama, kokią įtaką tinklo greitaveikai turi pavaldžių prietaisų skaičius tinkle ir tinklo topologija. Taikant *MATLAB* ir *Beckhoff TwinCAT 3* programinę įrangą ir sudarytus tyrimo metodus, tiriama *EtherCAT* realaus laiko komunikacinio tinklo greitaveika ir vėlinimo laikai. Sudaromos rekomendacijos, leidžiančios efektyvinti *Ethernet* šakos pramonės komunikacinį tinklą.

Nuobaras Tomas. Research of Industrial Ethernet Control Network Topologies. Master's Final Degree Project / supervisor Dr. Donatas Pelenis; Panevėžys Faculty of Technology and business, Kaunas University of Technology.

Study field and area (study field group): Electronics Engineering, Technology Sciences (Engineering Sciences).

Keywords: real-time control networks; *EtherCAT*; *PROFINET*.

Panevėžys, 2021. 65 pages.

### Summary

This paper investigates real-time *Ethernet* branch *EtherCAT* and *PROFINET* industrial control networks. When designing large and complex industrial control systems, it is very important to ensure the minimum possible response time. In complex systems with many positioning devices e.g., manipulators, various actuators, in case of excessive response time, there are problems with positioning accuracy, so when designing such systems, it is important to know how the response time varies in different communication buses depending on the number of controlled devices used, bus connection type and data transfer rate.

The scientific literature analysis part of this work analyzes scientific sources related to real-time *EtherCAT* and *PROFINET* control networks. Based on the data found in the scientific articles, the methods of connecting these two control networks are described, formulas characterizing the network speed are given, and the protocols based on which these two control networks operate are described. The algorithm for sending data to slave devices and receiving data from slave devices is described. The similarities and differences between these two networks are described.

In the research part of this work, methods suitable for studying the time parameters of *EtherCAT* and *PROFINET* networks, depending on the network topologies and the number of network devices are developed. Using the developed methods, the influence of the number of slave devices in the network and the network topology on the network speed is investigated. Using *MATLAB* and *Beckhoff TwinCAT 3* software and developed test methods, the speed and delay times of the *EtherCAT* real-time communication network are investigated. Recommendations are being developed to streamline the industrial *Ethernet* communications network.

## Turinys

Lentelių sąrašas .....	9
Paveikslų sąrašas .....	10
Santrumpų ir terminų sąrašas .....	12
Įvadas.....	13
<b>1. Darbo aktualumas .....</b>	<b>15</b>
<b>2. Pramoninės komunikacijų magistralės .....</b>	<b>18</b>
2.1. <i>EtherCAT</i> realaus laiko valdymo tinklas .....	18
2.1.1. <i>EtherCAT</i> naudojamas protokolas .....	18
<i>EtherCAT</i> duomenų perdavimo terpė .....	20
2.1.2. Prietaisų sinchronizacija <i>EtherCAT</i> tinkle.....	21
2.1.3. Duomenų apsikeitimo ciklo laikas <i>EtherCAT</i> tinkle .....	21
2.1.4. <i>EtherCAT</i> tinklo jungimo būdai .....	23
2.2. <i>PROFINET</i> realaus laiko valdymo tinklas .....	26
2.2.1. <i>PROFINET</i> duomenų perdavimo terpė .....	27
2.2.2. <i>PROFINET IO</i> komunikacijos modelis.....	28
2.2.3. <i>PROFINETN RT</i> duomenų paketo struktūra .....	29
2.2.4. <i>PROFINET IRT</i> duomenų paketo struktūra .....	30
2.2.5. <i>PROFINET</i> tinklo jungimo būdai .....	30
<b>3. Tyrimo metodai .....</b>	<b>34</b>
3.1. Metodikos, skirtos minimaliam duomenų apsikeitimo laikui <i>EtherCAT</i> ir <i>PROFINET</i> tinkluose tirti, sudarymas .....	34
3.2. Metodai, skirti <i>EtherCAT</i> realaus laiko komunikacinio tinklo greitimeikiai ir vėlinimo laikams nustatyti <i>MATLAB</i> programine įranga.....	35
3.2.1. Tiriamojo <i>EtherCAT</i> tinklo sudarymas .....	36
3.2.2. Signalų vėlinimo modelio sudarymas <i>MATLAB Simulink</i> aplinkoje .....	38
3.2.3. <i>TwinCAT 3</i> programos prijungimas prie <i>MATLAB Simulink</i> paprogramės ir modeliavimo rezultatų gavimas.....	39
<b>4. Tiriamoji dalis.....</b>	<b>44</b>
4.1. <i>EtherCAT</i> ir <i>PROFINET</i> realaus laiko valdymo tinklų duomenų apsikeitimo ciklų laikų tyrimas 44	
4.1.1. Tinklo kabelių ilgio įtakos komunikacinio signalo vėlinimo laikams tyrimas.....	44
4.1.2. Minimalaus duomenų apsikeitimo laiko, taikant prietaisų adresavimą priklausomybė nuo tinklo topologijos tyrimas.....	45
4.1.3. Duomenų paketo išsiuntimo iš valdančiojo prietaiso trukmė, naudojant prietaisų arba loginį adresavimą tyrimas .....	46
4.1.4. Minimalaus duomenų apsikeitimo laiko <i>EtherCAT</i> tinkle palyginimas, naudojant prietaisų arba loginį adresavimus tyrimas .....	47
4.1.5. Minimalaus duomenų apsikeitimo laiko <i>PROFINET</i> tinkle priklausomybės nuo tinklo topologijos tyrimas .....	48
4.1.6. Minimalaus duomenų apsikeitimo laiko nuosekliai sujungtame <i>PROFINET</i> tinkle priklausomybės nuo vidutinio tinklo kabelio tarp prietaisų ilgio tyrimas .....	49
4.1.7. Minimalaus duomenų apsikeitimo laiko <i>EtherCAT</i> ir <i>PROFINET</i> tinkluose palyginimas... 50	
4.2. <i>EtherCAT</i> realaus laikos komunikacinio tinklo greitimeikos ir vėlinimo laikų tyrimas .....	52
4.2.1. Bandomojo <i>MATLAB Simulink</i> modelio sudarymas ir jo tyrimas.....	52

4.2.2. <i>MATLAB Simulink</i> modelio sudarymas ir jo tyrimas, 1 dalis.....	54
4.2.3. <i>MATLAB Simulink</i> modelio sudarymas ir jo tyrimas, 2 dalis.....	57
4.3. Rekomendacijos leidžiančios efektyvinti <i>Ethernet</i> šakos pramonės komunikacinį tinklą.....	62
<b>Išvados</b> .....	<b>63</b>
<b>Literatūros sąrašas</b> .....	<b>64</b>



## Lentelių sąrašas

<b>1 lentelė.</b> <i>EtherCAT</i> Kabelio gyslų reikšmė ir išdėstymas jungtyje.....	20
<b>2 lentelė.</b> <i>PROFINET</i> kabelio gyslų reikšmė ir išdėstymas jungtyje.....	27
<b>3 lentelė.</b> Gamintojo pateikti pavaldžių <i>EtherCAT</i> prietaisų įrangos vėlinimo laikai.....	34

## Paveikslų sąrašas

<b>1 pav.</b> Automatinė sienų panelių kalimo linija .....	15
<b>2 pav.</b> <i>EtherCAT</i> tinklo struktūra .....	16
<b>3 pav.</b> <i>EtherCAT</i> duomenų apsikeitimo ciklo laikai .....	17
<b>4 pav.</b> <i>EtherCAT</i> veikimas [4] .....	18
<b>5 pav.</b> <i>EtherCAT</i> paketo struktūra [6] .....	19
<b>6 pav.</b> <i>EtherCAT</i> protokolo paaiškinimas .....	20
<b>7 pav.</b> <i>EtherCAT</i> paketo struktūra .....	22
<b>8 pav.</b> Nuoseklaus jungimo tipo valdymo grandinė [12] .....	23
<b>9 pav.</b> Žiedo jungimo tipo valdymo tinklo grandinė [12] .....	24
<b>10 pav.</b> Gedimas žiedo jungimo tipo valdymo tinklo grandinėje [12] .....	24
<b>11 pav.</b> Žvaigždės jungimo tipo valdymo tinklo schema [12] .....	25
<b>12 pav.</b> Medžio jungimo tipo valdymo tinklo schema [12] .....	26
<b>13 pav.</b> <i>PROFINET</i> komunikacijos modelis [18] .....	28
<b>14 pav.</b> <i>PROFINET</i> RT protokolo struktūra [18] .....	29
<b>15 pav.</b> <i>PROFINET</i> IO duomenų kadrą [18] .....	29
<b>16 pav.</b> <i>PROFINET</i> IRT duomenų kadro struktūra [18] .....	30
<b>17 pav.</b> Nuoseklus <i>PROFINET</i> tinklo jungimo būdas [27] .....	31
<b>18 pav.</b> Žvaigždės tipo <i>PROFINET</i> tinklo jungimo būdas [24] .....	32
<b>19 pav.</b> Medžio tipo <i>PROFINET</i> tinklo jungimo būdas [24] .....	32
<b>20 pav.</b> Prie kompiuterio prijungtas pavaldus prietaisas .....	36
<b>21 pav.</b> Su <i>EtherCAT</i> prietaisais suderinamų tinklo adapterių sąrašas .....	37
<b>22 pav.</b> Programos aptiktų prietaisų sąrašas .....	38
<b>23 pav.</b> Tyrimui naudojamas bandomasis modelis .....	38
<b>24 pav.</b> Tyrimui naudojamas modelis įkeltas į simuliacijos programą .....	39
<b>25 pav.</b> Simuliacijos programos dialogo langas .....	41
<b>26 pav.</b> <i>MATLAB</i> programos <i>Simulink</i> paprogramės dialogo langas .....	41
<b>27 pav.</b> <i>Simulink</i> programos prijungimo prie simuliacijos programos dialogo langai .....	42
<b>28 pav.</b> Vykstančios simuliacijos indikacija <i>Simulink</i> paprogramėje .....	43
<b>29 pav.</b> Minimalaus duomenų apsikeitimo laiko nuosekliai sujungtame <i>EtherCAT</i> tinkle grafikas .....	45
<b>30 pav.</b> Minimalaus duomenų apsikeitimo laiko <i>EtherCAT</i> tinkle naudojant prietaisų adresavimą grafikas .....	46
<b>31 pav.</b> Duomenų paketo išsiuntimo laiko palyginimas, naudojant loginį arba prietaisų adresavimą .....	47
<b>32 pav.</b> Minimalaus duomenų apsikeitimo laiko <i>EtherCAT</i> tinkle naudojant prietaisų arba loginį adresavimą palyginamasis grafikas .....	48
<b>33 pav.</b> Minimalaus duomenų apsikeitimo laiko priklausomybė nuo prietaisų skaičiaus ir tinklo topologijos .....	49
<b>34 pav.</b> Minimalaus duomenų apsikeitimo laiko nuosekliai sujungtame <i>PROFINET</i> tinkle priklausomybė nuo tinklo kabelio ilgio tarp prietaisų ir prietaisų skaičiaus .....	50
<b>35 pav.</b> Minimalaus duomenų apsikeitimo laiko <i>EtherCAT</i> ir <i>PROFINET</i> tinkluose palyginimas .....	51
<b>36 pav.</b> Gautų rezultatų grafikas .....	52
<b>37 pav.</b> Gautų rezultatų grafikas su priartintais laiko ašies duomenimis, 1 dalis .....	53
<b>38 pav.</b> Gautų rezultatų grafikas su priartintais laiko ašies duomenimis, 2 dalis .....	54
<b>39 pav.</b> Tyrimui naudojamas modelis .....	55

<b>40 pav.</b> Diskretinių įėjimų ir išėjimų jungimas .....	55
<b>41 pav.</b> Tyrimui naudojamam modeliui rezultatus grafiniai .....	56
<b>42 pav.</b> Tyrimui naudojamam modeliui rezultatus grafiniai su priartintais laiko ašies duomenimis, 1 dalis .....	56
<b>43 pav.</b> Tyrimui naudojamam modeliui rezultatus grafiniai su priartintais laiko ašies duomenimis, 2 dalis .....	57
<b>44 pav.</b> Tyrimui naudojamas modelis .....	58
<b>45 pav.</b> Diskretinių įėjimų ir išėjimų jungimas .....	59
<b>46 pav.</b> Tyrimui naudojamam modeliui rezultatus grafiniai .....	59
<b>47 pav.</b> Tyrimui naudojamam modeliui rezultatus grafiniai su priartintais laiko ašies duomenimis, 1 dalis .....	60
<b>48 pav.</b> Tyrimui naudojamam modeliui rezultatus grafiniai su priartintais laiko ašies duomenimis, 2 dalis .....	60
<b>49 pav.</b> Tyrimui naudojamam modeliui rezultatus grafiniai su priartintais laiko ašies duomenimis ir sumažintu generuojamo signalo dažniu .....	61

## Santrumpų ir terminų sąrašas

### Terminai:

CPU – centrinis procesorius (angl. *Central Processing Unit*);

CSMA/CD – kolektyvioji kreiptis aptinkant nešlį ir kontroliuojant kolizijas (angl. *Carrier Sense Multiple Access / Collision Detect*);

IP – interneto protokolas (angl. *Internet Protocol*);

ISO – tarptautinė standartizavimo organizacija (angl. *International Standard Organization*);

MAC – kreipties į terpę valdymas (angl. *Media Access Control*);

PHY – fizinis lygis (angl. *Physical Layer*);

RT – realaus laiko (angl. *Real Time*);

RX – imtuvas arba priimti (angl. *Receiver* arba *Reveive*);

TCP – perdavimo valdymo protokolas (angl. *Transport Control Protocol*);

TX – siųsti arba siųstuvas (angl. *Transmit* arba *Transmitter*);

UDP – OSI trečiojo lygmens transporto protokolas (angl. *User Datagram Protocol*);

VLAN – virtualus LAN (angl. *Virtual LAN*).

## Ivadas

Vis sparčiau tobulėjant ir sudėtingėjant pramoniniams procesams, automatizuojant vis daugiau gamybinių procesų, įprastai naudojamos pramoninių komunikacijų magistralės tampa netinkamos dėl per mažo duomenų perdavimo greičio, mažo atsparumo trikdžiams ir didelių vėlinimo laikų. Todėl pasauliui žengiant į pramonės revoliuciją 4.0, vis dažniau automatizuotose gamybiniuose įrenginiuose galima sutikti realaus laiko valdymo tinklų, tokių kaip *EtherCAT* ar *PROFINET*.

Šių tinklų paskirtis – kaip galima labiau sumažinti vėlinimo laikus nuo valdymo signalo išsiuntimo iš valdiklio iki jo priėmimo ir apdorojimo vykdančiame prietaise, pvz. nutolę valdiklio įėjimų/ išėjimų moduliai, dažnio keitikliai, vykdikliai, žingsninių variklių valdymo moduliai.

Tačiau kaip ir su senesnėse pramoninėse komunikacijų magistralėse, taip ir dabartinėse realaus laiko komunikacijų magistralėse susiduriama su tam tikromis problemomis. Didėjant magistralėje naudojamų prietaisų kiekiui, ilgėja laikas per kurį valdomas signalas apkeliauja visus prietaisus. Projektuojant dideles ir sudėtingas pramonines valdymo sistemas, labai svarbu užtikrinti kuo mažesnę maksimalų reakcijos laiką. Sudėtingose sistemose su dideliu kiekiu pozicionavimo įrenginių pvz. manipuliatorių, įvairių vykdiklių, esant per dideliu reakcijos laikui iškyla pozicionavimo tikslumo problemos, todėl projektuojant tokias sistemas svarbu žinoti kaip kinta reakcijos laikas skirtingose komunikacijų magistralėse priklausomai nuo naudojamų valdomų prietaisų (angl. *Slave Device*) kiekio, magistralės jungimo tipo ( nuoseklus jungimas, žvaigždės tipo jungimas, žiedo tipo jungimas, medžio tipo jungimas), duomenų perdavimo spartos.

**Tyrimo objektas** – realaus laiko *Etherneto* šakos *EtherCAT* ir *PROFINET* pramoniniai valdymo tinklai.

**Darbo tikslas** – ištirti realaus laiko *Ethernet* šakos pramonės komunikacinį tinklą, charakterizuojant laikinius parametrus ir pateikiant rekomendacines priemones efektyvesnio pramonės tinklo topologijos sudarymui.

### Darbo uždaviniai:

1. Atlikti mokslinės literatūros analizę susijusią su realaus laiko komunikaciniais tinklais.
2. Išskirti realaus laiko *Ethernet* šakos pramonės komunikacinių technologijų privalumus ir trūkumus. Apžvelgti naudojamas technologines priemones ir taikymo metodus.
3. Sudaryti metodus tinkančius tirti *EtherCAT* ir *PROFINET* tinklų laikinius parametrus, atsižvelgiant į tinklų topologijas ir aptarnaujamų tinklo prietaisų skaičių.
4. Taikant *MATLAB* programinę įrangą ir sudarytus tyrimo metodus, ištirti *EtherCAT* realaus laiko komunikacinio tinklo greitaveiką ir vėlinimo laikus.
5. Sudaryti rekomendacijas, leidžiančias efektyvinti *Ethernet* šakos pramonės komunikacinį tinklą.

**Tyrimo metodai** – mokslinės literatūros analizė, modeliavimas.

Nuobaras, Tomas. *Ethernet* pramoninių valdymo tinklų topologijų tyrimas. Studentų mokslinė konferencija „Technologijų ir verslo aktualijos – 2021“. Panevėžys: Kauno technologijos universitetas, Panevėžio technologijų ir verslo fakultetas, 2021 m. balandžio 23 d.

Nuobaras, Tomas, Pelenis, Donatas. *Ethernet* pramoninių valdymo tinklų topologijų tyrimas // Technologijų ir verslo aktualijos – 2021: studentų mokslinių darbų konferencijos pranešimų

medžiaga, Lietuva, Panevėžys, 2021 m. balandžio 23 d. / Kauno technologijos universiteto Panevėžio technologijų ir verslo fakultetas. Kaunas: Kauno technologijos universitetas. ISSN 2538-8045. 2021.

## 1. Darbo aktualumas

Projektuojant automatinio valdymo sistemas dažnai susiduriama su problema, kai dėl nepakankamo duomenų apie naudojamus prietaisus kiekio, sunku nuspręsti ar norimas parinkti prietaisas tinkamas projektuojamai sistemai. Dėl nepakankamo duomenų kiekio sunku prognozuoti, kaip parinktas prietaisas veiks automatinėje valdymo sistemoje kartu su kitais prietaisais, ar keisis jo greitaveika.

Panaši situacija susiklostė modernizuojant 2007 metais švedų kompanijos *Randek BauTech* pagamintoje, pusiau automatinėje skydinių namų sienų panelių kalimo ir kraštų apipjovimo sistemoje. Specialistai užsakovo pageidavimu pusiau automatinę liniją perprojektavo į pilnai automatinę, pakeisdami esamą *PROFIBUS* valdymo tinklą į realaus laiko *EtherCAT* tinklą. Taip pat buvo pakeisti visi seno tinklo adapteriai nutolusioms valdiklio įėjimų / išėjimų saloms, dažnio keitiklių tinklo plokštės, pritaikytos veikti su *EtherCAT*, ir sistemos valdiklis. Duomenims įvesti ir tiltams valdyti buvo sumontuoti du pramoniniai kompiuteriai su lietimui jautriais ekranais. Visa ši sistema sudaryta iš konvejerių, skirtų sienų panelių karkasams pervežti iš vienos gamybos pozicijos į kitą ir dviejų tiltų, ant kurių sumontuoti plokščių prie karkaso prisukimo, kraštų ir langų plokštėje išpjovimo, skylių elektros lizdams gręžimo įrankiai. Vieno iš dviejų tiltų nuotrauka pateikta 1 paveiksle.



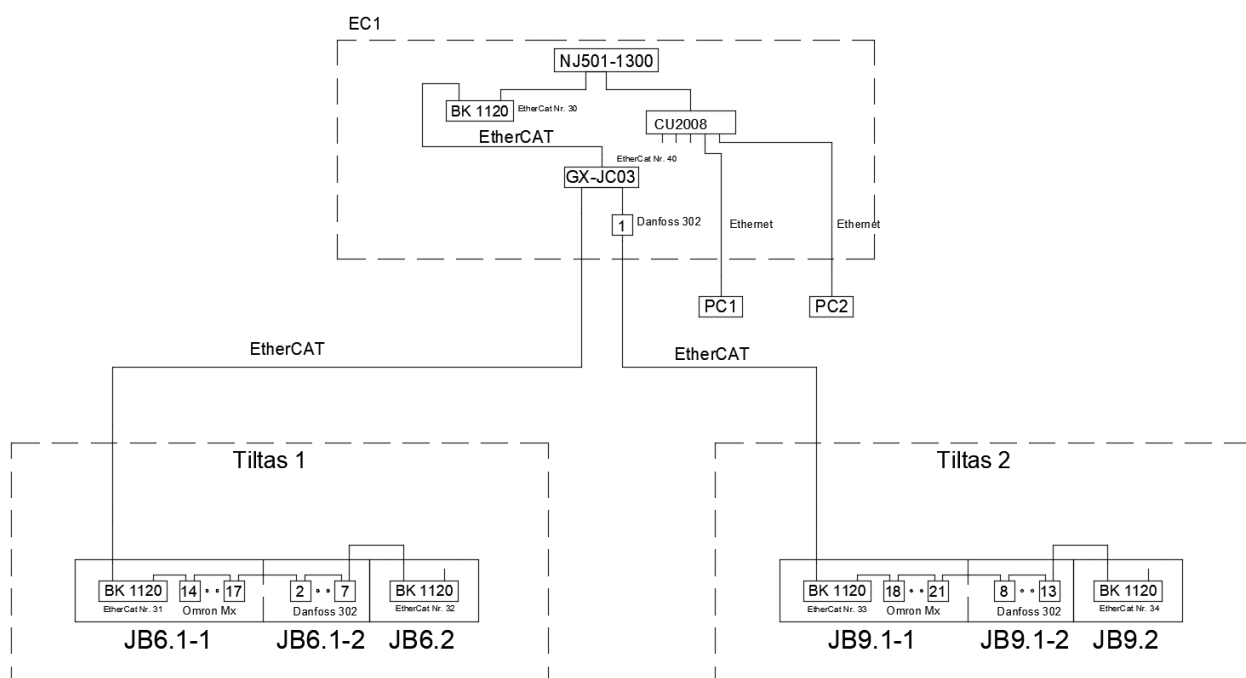
1 pav. Automatinė sienų panelių kalimo linija

Užsakovo pageidavimu papildomai buvo įdiegta karkasą sudarančių medinių tašelių geometrijos patikrinimo funkcija. Kad atlikti karkaso geometrijos tikrinimą, tilto, ant kurio sumontuoti kalimo ir pjovimo įrankiai, važinėjančio per karkasą, apačioje buvo sumontuoti du lazeriniai jutikliai (vienas kairėje tilto pusėje, kitas dešinėje). Šių lazerių spindulio diametras yra 0,5 milimetro, taip pat jie turi galimybę nustatyti suveikimo atstumą ir jautrumą. Šie jutikliai buvo sumontuoti taip, kad tiltui važiuojant per karkasą, lazerio spindulys šviečia statmenai karkaso paviršiui ir suveikia tik tada, kai

jo spindulys atsispindi nuo karkasą sudarančių medinių tašelių. Tiltas pirmyn ir atgal per karkasą važinėja tempiamas diržo varomo per krumpliaračių sistemą, kurią suka 2,2 kW trifazis asinchroninis variklis valdomas dažnio keitiklio. Variklio nominalus sūkių per vieną minutę skaičius – 1410. Ant variklio ašies sumontuotas enkoderis, turintis 8192 impulsų per vieną apsisukimą rezoliuciją. Pagal enkoderio duomenis nustatoma tilto padėtis  $x$  ašies atžvilgiu. Karkaso tikrinimo jutikliui priartėjus prie medinio tašelio krašto ir suveikus, sistemos valdiklis pagal gautus duomenis iš enkoderio, valdymo programoje aprašytu skaičiavimo metodu, apskaičiuoja tašelio pradžios krašto  $x$  koordinatę. Jutiklis tiltui pravažiavus per visą tašelį išsijungia ir valdiklis tokiu pat būdu apskaičiuoja antrąją, tašelio pabaigos  $x$  koordinatę. Atėmus pirmąją koordinatę iš antros, gaunamas tašelio plotis. O lyginant abiejų jutiklių gautas pradžios koordinates, nustatoma ar tašelis atitinka iš anksto nustatytas statmenumo paklaidas. Jei paklaidos atitinka nustatytas ribas, reiškia namo sienos karkasas sukaltas tinkamai ir operatoriams leidžiama dėti gipso kartono plokštes ant karkaso ir įjungti automatinį plokščių prikalinimą prie karkaso ir jų kraštų bei langų išpjovimą.

Taip pat pagal jutiklių duomenis suskaičiuojama, kiek tašelių yra karkaso viduje ir kokio jie pločio, šie duomenys automatiškai lyginami su duomenimis gautais iš brėžinio esančio automatinės skydinių namų sienų panelių kalimo ir kraštų apipjovimo sistemos valdymo kompiuteryje. Į šį kompiuterį brėžinį įkelia už projektą atsakingas operatorius, o brėžinio popierinį variantą pateikia karkasą kalantiems darbuotojams. Tokiu būdu visada užtikrinama, kad kalant karkasą būtų išvengta konstrukcinių klaidų, tokių kaip netinkamų, per plonų tašelių sukavimo, ar jų sukavimo netinkamose vietose, dėl kurių gali sumažėti skydo tvirtumas.

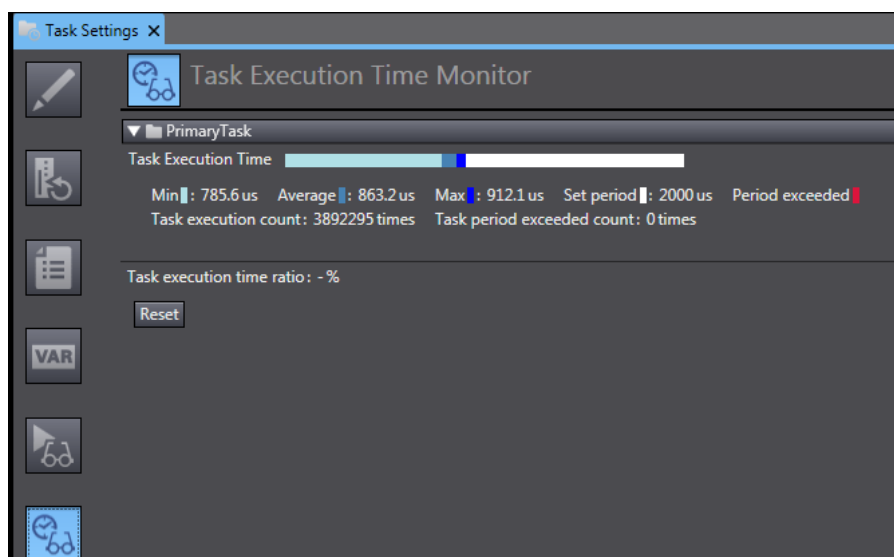
*EtherCAT* tinklas šioje sistemoje sujungtas žvaigždės tinklo jungimo būdu naudojant *EtherCAT* tinklo šakotuvą su trimis portais. 2 paveiksle pateikta sistemos *EtherCAT* tinklo jungimo schema. Tinklą sudaro 21 dažnio keitiklis, 5 nutolę valdiklio įėjimų / išėjimų salos (bendrai: 244 įėjimai, 112 išėjimų, 23 enkoderio moduliai), tinklo šakotuvas ir valdiklis. Bendras *EtherCAT* tinklo kabelio ilgis siekia 180 metrų.



2 pav. *EtherCAT* tinklo struktūra



Atliekant pirmuosius tinklo bandymus parinktas *EtherCAT* tinklo visų prietaisų apklausimo periodo laikas valdikliui. 3 paveiksle matyti, kad vieno iš matavimų metu minimalus ciklo laikas siekė 0,7856 milisekundės, maksimalus laikas siekė kiek daugiau nei 0,911 milisekundės, todėl buvo pasirinktas 1,5 milisekundės laikas. Bet paleidus automatinę sistemos režimą, kai vienu metu juda kelios ašys, valdiklis nespėdavo apklausti visų pavaldžių prietaisų per nustatytą laiką ir generuodavo klaidos signalą. Todėl siekiant užtikrinti tinklo stabilumą buvo pasirinktas 2 milisekundžių apklausimo ciklo laikas.



**3 pav.** EtherCAT duomenų apsikeitimo ciklo laikai

Atlikus pirmuosius šios sistemos bandymus nustatyta, kad dėl per didelio domenų apsikeitimo tarp visų *EtherCAT* tinko prietaisų ciklo laiko, tiltui judant maksimaliu greičiu, gaunami neteisingi duomenis apie karkasą sudarančių medinių tašelių padėtį ir plotį. Taip pat nustatyta, kad mažinant tilto važiavimo greitį, duomenys gaunami tikslesni, o tiltui judant 5 procentai nominalaus greičio, paklaidos tampa minimalios ir neviršija nustatytų matavimo tikslumo ribų. Bet mažinant greitį kyla kita problema, tai sumažėjęs sistemos našumas, kurio buvo siekiama sistemą modernizuojant.

Todėl šio tyrimo tikslas nustatyti, kaip apsikeitimo duomenimis laikas tarp valdiklio ir pavaldžių prietaisų pasikeistų, jei *EtherCAT* magistralė būtų sujungta kitokiu būdu (pvz. nuosekliai) ir ar būtų buvę geriau perprojektuojant sistemą pasirinkti kito gamintojo magistralę, pavyzdžiui *PROFINET*, kadangi visi esantys sistemos prietaisai buvo suderinami tiek su *EtherCAT* tiek su *PROFINET*. Atlikus mokslinės literatūros analizę, išsiaiškinus šių dviejų realaus laiko valdymo tinklų protokolus, naudojamą įrangą, tinklo jungimo būdus ir duomenų perdavimo užlaikymo laiko priklausomybes nuo tinklo jungimo būdo ir įrenginių kiekio, sudaryti *MATLAB Simulink* paprogramėje modelius, kurie padėtų nustatyti, kuri realaus laiko valdymo sistema ir jos jungimo tipas tinka konkrečioje sistemoje.

## 2. Pramoninės komunikacijų magistralės

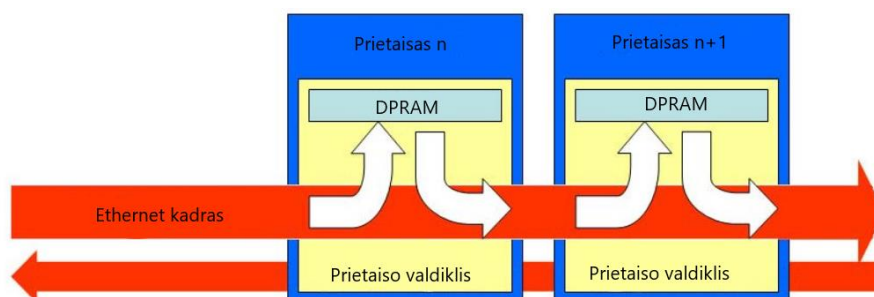
Pramoninėje didėjant gamybos apimtims ir pradedant naudoti sudėtingas sistemas, kyla poreikis valdyti gamybinius procesus realiu laiku. Šiam tikslui pasiekti naudojami įvairūs komunikacijų magistralių protokolai. Pagal reakcijos laiką realaus laiko valdymo tinklų sistemos skirstomos į šias tris protokolų klases [1]:

- 1 klasė: programinės: pasiekiamas maždaug 100 milisekundžių ciklo laikas;
- 2 klasė: aparatinės: ciklo trukmė nuo 1 iki 10 milisekundžių;
- 3 klasė: IRT (angl. *Isochronous Real-Time*): ciklo trukmė mažiau nei 1 milisekundė. Naudojama judesiams valdyti.

### 2.1. EtherCAT realaus laiko valdymo tinklas

*EtherCAT* – tai pramoninio realaus laiko valdymo tinklo standartas (ISO 17545-4) [2], priskiriamas pramoninio *Ethernet*’o grupei, naudojamas paskirstytiems realaus laiko valdymo tinklams. *EtherCAT* buvo sukurtas *Beckhoff* kompanijos pramoninio *Ethernet*’o protokolo pagrindu ir į rinką pristatytas 2004 metais [3].

*EtherCAT* valdomi prietaisai nepriima ir nesiunčia duomenų paketų arba kitaip kadrų (angl. *Frame*) juos apdorojė kaip tą daro *Ethernet*’o protokolu valdomi prietaisai. Vietoje to *EtherCAT* prietaisai skaito duomenis adresuotus jiems telegramos siuntimo per prietaisą metu (angl. *Processing Data „On the Fly“*) nuskaitoma (žr. 4 pav.). Tokiu pat būdu, telegramą persiunčiant pro prietaisą informacija apie prietaisą yra pridedama. Informacija pradedama apdoroti nespėjus priimti viso duomenų paketo, duomenų apdorojimo procesas pradedamas kaip galima anksčiau. Apdorojant informaciją tokiu būdu, pasiekiamas 100 mikrosekundžių reakcijos (ciklo) laikas [4].

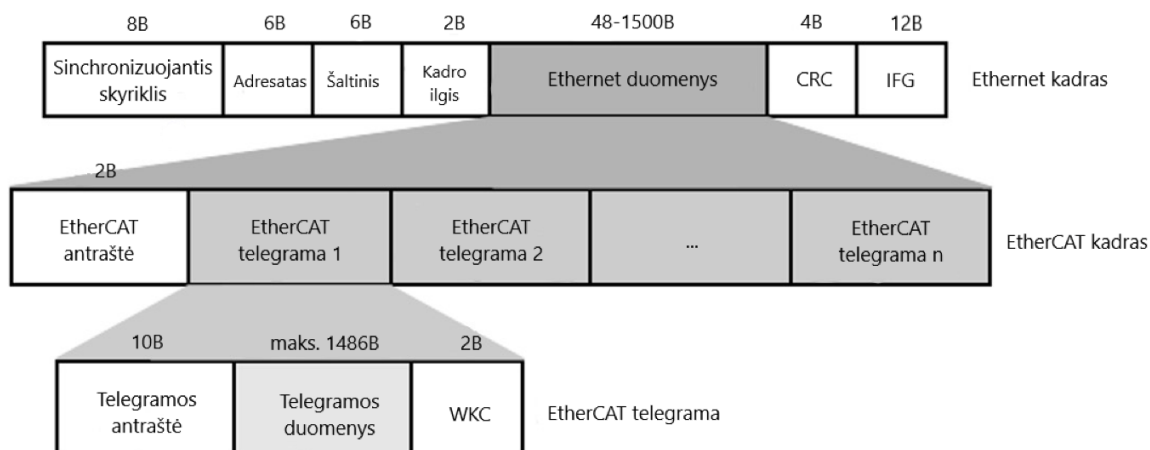


4 pav. EtherCAT veikimas [4]

Visi prietaisai esantys tinkle adresuojami viena telegrama, kuri nuosekliai apdorojama visuose tinklo prietaisuose. Telegramai pasiekus paskutinį prietaisą, ji yra apgręžiama atgal ir nuosekliai per visus prietaisus grąžinama į valdantįjį prietaisą.

#### 2.1.1. EtherCAT naudojamas protokolas

*EtherCAT* talpina savo duomenis į standartinį *Ethernet*’o kadą (žr. 5 pav.). Kadras yra identifikuojamas pagal identifikatorių *ox88A4* [5], kuris yra *Ethertype* lauke. Kadangi *EtherCAT* protokolas yra optimizuotas trumpiems cikliniams duomenų apsikeitimams, galima atsisakyti tokių protokolų rinkinių kaip TCP/IP ar UDP/IP naudojimo [6].



**5 pav.** EtherCAT paketo struktūra [6]

Sistemos paleidimo metu valdantysis įrenginys sukonfigūruoja ir nustato duomenų perdavimo kelius į pavaldžius renginius. Su kiekvienu pavaldžiu prietaisu gali vykti apsikeitimas skirtingu kiekiu informacijos. Informacijos kiekis gali svyruoti nuo vieno bito iki kelių baitų, kartais duomenų kiekis gali siekti kelis kilobaitus.

*EtherCAT* kadras (rėmas) susideda iš kelių telegramų (žr. 6 pav.). Telegramos antraštė nurodo kokią prieigą prie jos valdantysis prietaisas leidžia pavaldiems prietaisams. Antraštės būna trijų tipų:

1. Skaityti (angl. *Read*).
2. Rašyti (angl. *Write*).
3. Skaityti-rašyti (angl. *Read-Write*).

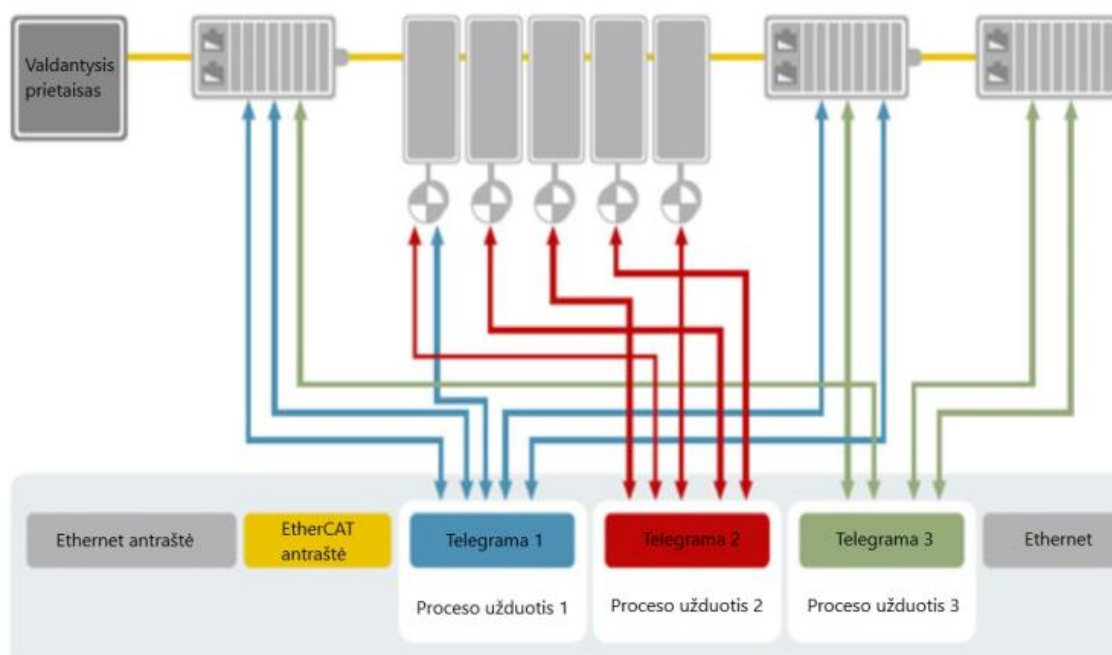
Taip pat antraštė nurodo kam bus skirta datagrama ar kuriam nors konkrečiam pavaldžiam prietaisui naudojant jo tiesioginį adresą ar keliems pavaldiems prietaisams naudojant loginį adresavimą, t. y. adresai išdėstyti matematiškai aprašoma tvarka, pvz. datagrama skirta pirmiems penkiems prietaisams arba kas antram prietaisui.

Loginis adresavimas naudojamas cikliniams duomenų apsikeitimams tarp valdančiojo ir pavaldžių prietaisų. Kiekviena telegrama skirta tam tikrai valdomo proceso daliai *EtherCAT* segmente. Paleidžiant tinklą kiekvienam pavaldžiam prietaisui priskiriamas vienas ar daugiau adresų iš šios adresų saugojimo vietos [7]. Jei keletui iš pavaldžių prietaisų priskiriamas toks pat adresas, tai jie visi duomenimis su valdančiuoju prietaisu apsikeis naudodami vieną telegramą.

Kadangi kiekviena telegrama turi visą reikiamą informaciją skirtą prieigai, valdantysis prietaisas gali spręsti kada ir kokius duomenis prieti. Pavyzdžiui valdantysis prietaisas gali naudoti telegramas su trumpais ciklų laikais valdyti dažnio keitikliams ar servo pavaroms, o telegramas su ilgesniais ciklų laikais naudoti apsikeisti duomenimis su įėjimų / išėjimų moduliais.

Tai taip pat sumažina valdančiojo prietaiso apkrovą lyginant su kitomis komunikacijų magistralėmis, kuriose duomenys iš kiekvieno pavaldaus elemento turi būti nuskaitomi individualiai, surūšiuojami proceso kontrolieriu ir įkopijuojami į atmintį.

Naudojant *EtherCAT*, valdantysis prietaisas turi tik užpildyti vieną *EtherCAT* kadra su siunčiamais duomenimis ir išsiusti jį per *DMA* į *MAC* valdiklį.



6 pav. *EtherCAT* protokolo paaiškinimas

Kai kadras su naujais įėjimų duomenimis gautas per *MAC* valdiklį, valdantysis prietaisas gali nukopijuoti kadra per *DMA* į kompiuterio atmintį, taip visiškai nenaudojant *CPU* resursų duomenims kopijuoti [8].

### ***EtherCAT* duomenų perdavimo terpė**

Duomenys tarp *EtherCAT* prietaisų perduodami ekranuotu ne žemesnės nei 5 kategorijos kabeliu (CAT 5) atitinkančiu EN 50173 arba ISO/IEC 11801 standartą [9]. *EtherCAT* naudoja keturis laidus duomenims tarp prietaisų apsikeisti. Du naudojami duomenims siųsti, o kiti du – duomenis gauti. Dažniausiai *EtherCAT* prietaisai naudoja standartinę RJ45 jungtį. Be RJ45 jungčių, taip pat naudojamos M8 ir M12 prisukamos jungtys. Laidų išdėstymas RJ45 jungtyje suderinamas su vidinio tinklo (*Ethernet*‘o) standartu ISO/IEC 8802-3. Standartinis *EtherCAT* laidų išdėstymas RJ45 jungtyje pateiktas 1 lentelėje.

1 lentelė. *EtherCAT* Kabelio gyslų reikšmė ir išdėstymas jungtyje

Laidelis	Laidelio spalva	Signalas	Reikšmė
1	geltona	TD +	TX duomenys +
2	oranžinė	TD -	TX duomenys -
3	balta	RD +	RX duomenys +
6	mėlyna	RD -	RX duomenys -

Taip pat svarbu paminėti, kad *EtherCAT* prietaisai naudoja automatinę kabelio detekciją ir tikrina ar kabelis „tiesus“ ar „verstas“. „Tiesaus“ kabelio laideliai abiejuose galuose į jungtį jungiami vienoda tvarka, o „versto“ viename iš galų 1 ir 2 laideliai sukeičiami vietomis su 3 ir 6 laideliais. Dėl

naudojamos automatinės kabelio detekcijos, prietaisams sujungti gali būti panaudotas tiek vieno tiek kito tipo kabelis.

Maksimalus kabelio ilgis tarp prietaisų negali viršyti 100 metrų. Tarp prietaisų galima naudoti ne vientisą kabelį, o iš kelių sujungtą į vieną, tam naudojant specialias jungtis. Norint išvengti trikdžių atsirandančių dėl kabelio ne vientisumo negalima naudoti daugiau nei šešių jungčių kabeliams sujungti tarp dviejų prietaisų. Esant didesniam jungčių kiekiui, gali atsirasti trikdžių, dėl kurių suprastėtų perduodamo signalo kokybė, sumažėtų duomenų perdavimo sparta. Viename *EtherCAT* tinklo segmente gali būti sujungta iki 65535 prietaisų.

### 2.1.2. Prietaisų sinchronizacija *EtherCAT* tinkle

*EtherCAT* naudoja paskirstytojo laiko mechanizmą, kuris užtikrina mažą gautų paketų vėlavimo trukmės kitimą, atitinkantį IEEE 1588 [10], [13] specifikacijas nenaudojant papildomos įrangos. Kiekvienas pavaldus prietaisas prideda laiko žynę į *EtherCAT* kadra du kartus [11]. Pirmąją laiko žymą pavaldus prietaisas prideda į *EtherCAT* kadra, kai jis gauna telegramą siunčiamą tinklu per prietaisus. Antrąją laiko žymą prietaisas į kadra prideda tada, kai telegrama yra gražinama iš paskutiniojo tinklo prietaiso per visus prietaisus atgal. Valdantysis prietaisas gauna kadra, kuriame yra po dvi laiko žymas kiekvienam tinkle esančiam pavaldžiam prietaisui. Pagal šiuos duomenis valdantysis prietaisas gali apskaičiuoti kiekvieno pavaldaus prietaiso vėlinimus. Valdantysis prietaisas kartuoja vėlinimo skaičiavimus kiekvieną kartą kai išsiųstas kadras yra gaunamas. Kadangi tinklui veikiant kadrai yra siunčiami nuolatos, atliekama daugybė skaičiavimų, kas užtikrina labai tikslius vėlinimo laiko duomenis, kurie su kiekvienu gaunamu kadru vis tikslinami [12].

### 2.1.3. Duomenų apsikeitimo ciklo laikas *EtherCAT* tinkle

Duomenų apsikeitimo ciklo laikas *EtherCAT* tinkle, tai laikas reikalingas duomenims iš valdančiojo prietaiso nusiųsti į pavaldžius prietaisus ir duomenims iš pavaldžių prietaisų gauti. Bendras duomenų apsikeitimo ciklo laikas gali būti padalintas į dvi dalis kaip pateikta 1 formulėje:

$$T_{ciklo} = T_{kadro} + T_{tinklo}; \quad (1)$$

Pirmoji dalis ( $T_{kadro}$ ) nurodo laiką reikalingą valdikliui ištransliuoti visus bitus esančius duomenų pakete. Šis laikas išreiškiamas visą duomenų paketo dydį bitais dalinant iš tinklo pralaidumo.  $T_{kadro}$  laikas aprašomas lygtimi:

$$T_{kadro} = \frac{L_{kadro} \times 8}{bw}; \quad (2)$$

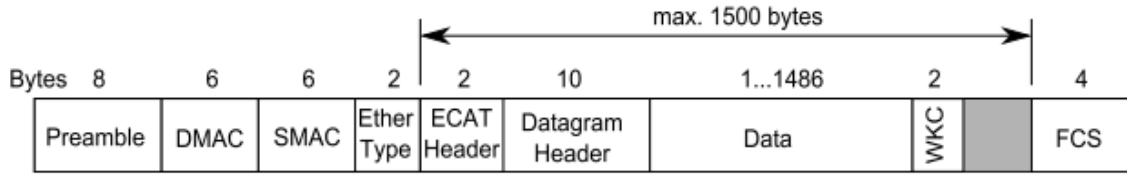
čia  $bw$  – tinklo pralaidumas bitais per sekundę;

$L_{kadro}$  – kadro dydis bitais.

Antroji dalis ( $T_{tinklo}$ ) nurodo tinklo vėlinimo laiką. Šis laikas susideda iš vėlinimų, kuriuos sukelia pavaldūs prietaisai, tinklo šakotuvai, valdančiojo prietaiso procesorius, taip pat šiam laikui įtaką daro vėlinimai susidarantys tinklo kabeliuose. Šį vėlinimo laiką įtakoja tiek vidutinis laido ilgis tarp prietaisų, tiek tinklo topologija. Vėlinimo laiko tinkle apskaičiavimui reikalingos formulės pateiktos tolimesniuose skyreliuose.

Kad apskaičiuoti laiką reikalingą valdikliui ištransliuoti visus duomenis esančius pakete reikia žinoti maksimalų paketo dydį bitais ir tinklo pralaidumą. Tinklo pralaidumas *EtherCAT* tinke remiantis

*Beckhoff automation* kompanijos pateiktais duomenimis yra 100 Mbit/s. Kad apskaičiuoti maksimalų paketo dydį, reikia žinoti *EtherCAT* paketo struktūrą, kuri pateikta 7 paveiksle ir vadovautis žemiau pateiktomis lygtimis.



7 pav. *EtherCAT* paketo struktūra

Kad duomenys būtų pristatyti į reikiamą prietaisą *EtherCAT* protokolas naudoja prietaisų adresavimą, pagal kurį kiekvienam prietaisui yra priskiriama atskira telegrama (datagrama). Kiekvienai telegramai dar priskiriama 10 bitų antraštė ir 2 bitų skaitiklis. Jei *EtherCAT* duomenų paketo dydis viršija 1500 bitų, duomenų paketas turi būti dalinamas į du mažesnius, kas didina duomenų apsikeitimo tinkle laiką. Tokiu atveju, kai priskiriant kiekvienam prietaisui po atskirą telegramą duomenų paketas viršija 1500 bitų apimtį, *EtherCAT* protokolas naudoja mechanizmą vadinamą loginiu adresavimu. Tokiu atveju visiems *EtherCAT* tinkle esantiems prietaisams proceso užduoties duomenys perduodami sutalpinti į vieną telegramą. Duomenis perduodant viena telegrama naudojama tik viena 10 bitų telegramos antraštė ir tik vienas 2 bitų skaitiklis.

Apskaičiuoti *EtherCAT* paketo dydį, kai naudojamas prietaisų adresavimas galima pagal formulę:

$$L_{kadro} = (n \times p + n \times (h_D + h_{wkc})) + \frac{n}{1498} \times (h_E + h_{ECAT}); \quad (3)$$

čia  $d = p + h_D + h_{wkc}$ ;

$n$  – prietaisų kiekis tinkle;

$p$  – duomenų kiekis bitais siunčiamas į kiekvieną iš prietaisų;

$h_E$  – *Ethernet* antraštė bitais;

$h_{ECAT}$  – *EtherCAT* antraštė bitais;

$h_{wkc}$  – skaitiklis (angl. *Working Counter*) bitais;

$h_D$  – telegramos antraštė bitais.

Kai vietoje prietaisų adresavimo naudojamas loginis adresavimas, maksimalus *EtherCAT* duomenų paketo ilgis apskaičiuojamas pagal formulę:

$$L_{kadro} = (n \times p + \frac{n \times p}{1486} \times (h_E + h_{ECAT} + h_D + h_{wkc})); \quad (4)$$

čia  $h_D$  – telegramos antraštė bitais;

$n$  – prietaisų kiekis tinkle;

$p$  – duomenų kiekis bitais siunčiamas į kiekvieną iš prietaisų;

$h_E$  – *Ethernet* antraštė bitais;

$h_{ECAT}$  – *EtherCAT* antraštė bitais;

$h_{wkc}$  – skaitiklis (angl. *Working Counter*) bitais.

### 2.1.4. EtherCAT tinklo jungimo būdai

EtherCAT prietaisai gali būti sujungti keturiais skirtingais jungimo būdais [14]:

- nuosekliu;
- žiedo tipo;
- žvaigždės tipo;
- medžio tipo.

Dažniausiai pramoninėse komunikacijų magistralėse yra naudojamas nuoseklus jungimas. Jungiant prietaisus tokiu būdu, kabelių komutavimas užima mažiau laiko, taip pat toks jungimas sumažina išlaidas sistemai įrengti, nes nereikia vesti magistralės kabelio iš paskutinio prietaiso išėjimo jungties iki valdančiojo prietaiso įėjimo jungties, taip sumažinant bendrą kabelių ilgį.

Pagrindinis šio tipo jungimo trūkumas yra nepakankamas patikimumas magistralės kabelio mechaniniams pažeidimams. Bet koks kabelio pažeidimas sutrikdo visos sistemos darbą, kadangi visi pavaldūs prietaisai esantys už kabelio pertraukimo vietos nustoja veikti.

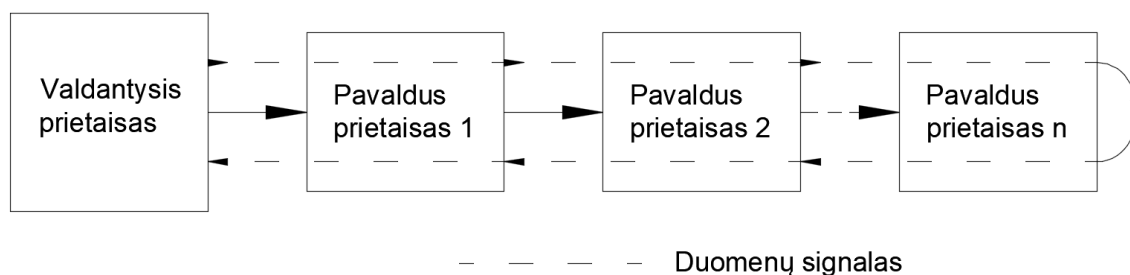
Nuosekli EtherCAT realaus laiko valdymo magistralė gaunama jungiant nuosekliai valdomus prietaisus turinčius po dvi jungtis (žr. 8 pav.). Viena jungtis skirta įeinančiam valdymo signalui, antroji išeinančiam. Paskutinis prietaisas nuosekloje valdymo grandinėje automatiškai aptinka atvirą jungtį ir ją „uždaro“, o valdymo signalą nukreipia atgal, tokiu būdu prietaisas veikia kaip vienos jungties [15].

Signalų apdorojimo laikas dviejų jungčių prietaise aprašomas lygtimi:

$$t_{\text{prietaiso}} = t_{MM_p} + t_{MM_f} + 2 \times (t_{Rx} + t_{Tx}); \quad (5)$$

čia  $t_{Rx}$  – gaunamo signalo vėlinimas;

$t_{Tx}$  – siunčiamo signalo vėlinimas.



8 pav. Nuoseklus jungimo tipo valdymo grandinė [12]

Paskutinis prietaisas nuosekloje valdymo grandinėje automatiškai aptinka atvirą jungtį ir ją „uždaro“, o valdymo signalą nukreipia atgal, tokiu būdu prietaisas veikia kaip turėdamas tik vieną jungtį.

Signalų apdorojimo laikas paskutiniame nuoseklos valdymo grandinės prietaise aprašomas lygtimi:

$$t_{\text{paskutinio prietaiso}} = t_{MM_p} + t_{MM_f} + t_{Tx}; \quad (6)$$

čia  $t_{Rx}$  – gaunamo signalo vėlinimas;

$t_{Tx}$  – siunčiamo signalo vėlinimas.

Visos nuoseklios valdymo magistralės valdymo reakcijos laikas gali apskaičiuojamas taip:

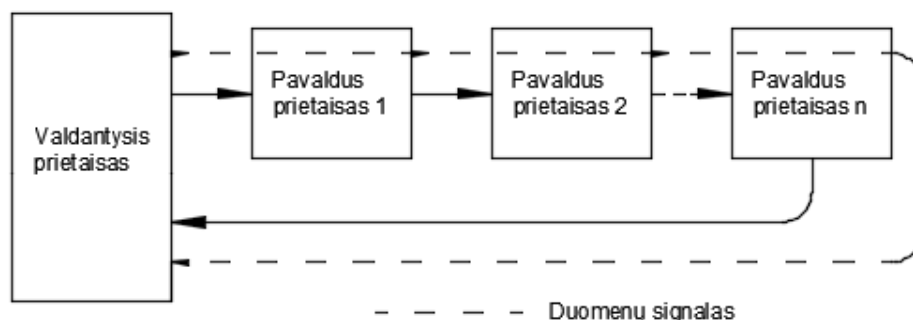
$$T_{Tinklo} = 2 \times n \times T_{Laido} + T_{valdančio\ prietaiso} + (n - 1) \times t_{prietaiso} + t_{paskutinio\ prietaiso}; \quad (7)$$

čia  $n$  – prietaisų kiekis;

$T_{Laido}$  – valdymo signalo sklidimo laikas laide tarp dviejų prietaisų;

$T_{valdančio\ prietaiso}$  – valdančio prietaiso signalo užlaikymas.

Pramoniniuose tinkluose žiedo jungimas taikomas didinant sistemos patikimumą. Žiedo tipo jungimas gaunamas į paskutinio valdomo prietaiso išėjimo jungtį įjungiant kabelį, kurio antrasis galas įjungiamas į valdančiojo įrenginio įėjimo jungtį (žr. 9 pav.).

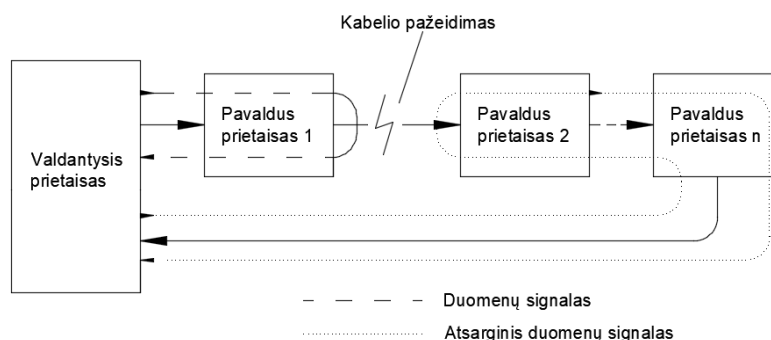


9 pav. Žiedo jungimo tipo valdymo tinklo grandinė [12]

Kabeliui nutrūkus gaunamos dvi nuoseklaus jungimo grandinės, viena valdoma per pirmąją (išėjimo) valdančiojo įrenginio jungtį, antra per antrąją (įėjimo) (žr. 10pav.).

Žiedo tipo magistralės reakcijos laikas apskaičiuojamas taip:

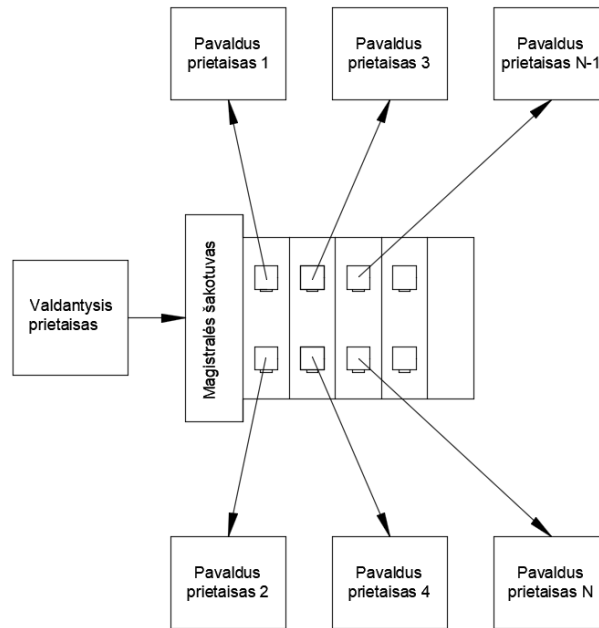
$$T_{Tinklo} = n \times T_{laido} + T_{valdančio\ prietaiso} + n \times t_{valdomo\ prietaiso}. \quad (8)$$



10 pav. Gedimas žiedo jungimo tipo valdymo tinklo grandinėje [12]

Žvaigždės jungimo tipas dažniausiai sutinkamas ofisuose naudojamuose kompiuterių tinkluose (žr. 11 pav.). Pramoninėje aplinkoje tokio tipo jungimas reikalauja žymiai didesnio kabelių kiekio ir papildomų prietaisų, tokių kaip tinklo šakotuvai. Šio tipo jungimas pasižymi dideliu patikimumu. Sugedus vienam ar keliems įrenginiams ar pažeidus ryšio kabelius einančius iki šių prietaisų, visos sistemos darbas nėra sutrikdomas.





**11 pav.** Žvaigždės jungimo tipo valdymo tinklo schema [12]

Norint sužinoti tokio jungimo tipo sistemos reakcijos laiką, pirmiausia reikia apskaičiuoti modulinio komutatoriaus (angl. *Switch*) reakcijos laiką. Šis komutatorius susideda iš magistralės šakotuvo adapterio ir reikiamo kiekio šakotuvų, šakotuvų kiekis parenkamas pagal maksimalų valdomų įrenginių kiekį.

Komutatoriaus reakcijos laikas apskaičiuojamas pagal formulę:

$$T_{Sw} = T_{Coupler} + \left[ \frac{n}{2} - 1 \right] \times T_{SwElement} + T_{SwEnd}; \quad (9)$$

$$\text{čia } T_{Coupler} = t_{ME_p} + t_{ME_f} + t_{Rx} + t_{Tx}; \quad (10)$$

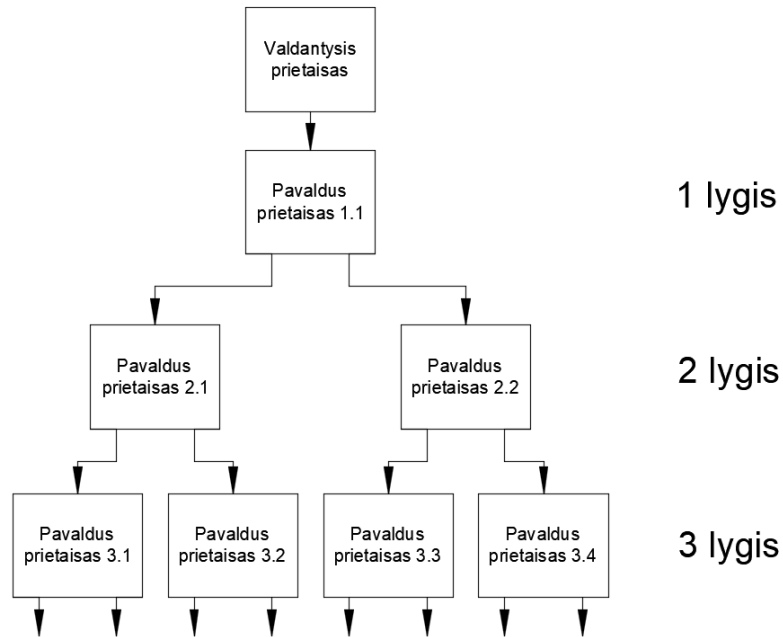
$$T_{SwElement} = t_{EM_p} + t_{MM_f} + t_{ME_f} + t_{EE_f} + 4 \times (t_{Rx} + t_{Tx});$$

$$T_{SwEnd} = t_{EM_p} + t_{MM_f} + t_{ME_f} + 4 \times (t_{Rx} + t_{Tx}).$$

Viso žvaigždės jungimo tipo tinklo reakcijos laikas apskaičiuojamas taip:

$$T_{tinklo} = 2 \times (n + 1) \times T_{Laido} + T_{Valdančio\ prietaiso} + T_{Sw} + n \times t_{paskutinio\ prietaiso}; \quad (11)$$

Medžio (angl. *Tree Topology*) tipo jungimas pramoniniuose tinkluose yra pats efektyviausias iš tinklų kuriuose naudojami tinklo šakotuvai. Taikant tokį jungimo būdą, reikalingas mažiausias įrangos vienetų kiekis, su mažiausiu jungčių skaičiumi. Dėl to užtikrinamas mažas reakcijos laikas [5]. Medžio tipo tinklo jungimas gaunamas naudojant valdomus prietaisus (angl. *Slave Devices*) su daugiau nei dvejomis jungtimis. 12 paveiksle pateiktas medžio tipo jungimo sistemos naudojant valdomus prietaisus su trimis jungtimis pavyzdys.



12 pav. Medžio jungimo tipo valdymo tinklo schema [12]

Sistemos reakcijos laikas šiai sistemai apskaičiuojamas taip:

$$T_{tinklo} = 2 \times n \times T_{laido} + T_{vald. \ prietaiso} + (n - m) \times t_{TreeNode} + m \times t_{paskutinio \ prietaiso}; \quad (12)$$

čia  $n$  – bendras valdomų prietaisų kiekis;

$m$  – galinių prietaisų kiekis;

$$t_{TreeNode} = t_{MM_p} + 2 \times t_{MM_f} + 3 \times (t_{Rx} + t_{Tx}). \quad (13)$$

Bendras prietaisų ir galinių prietaisų kiekis apskaičiuojamas pagal formules 10 ir 11.

$$n = 2^k - 1; \quad (14)$$

$$m = 2^{k-1}. \quad (15)$$

čia  $k$  – didžiausias tinklo lygis (žr. 11 pav.).

## 2.2. PROFINET realaus laiko valdymo tinklas

Realaus laiko valdymo tinklas sukurtas kompanijos *PROFIBUS & PROFINET International*, prastatytas į rinką 2003 metais. Naudojamas pramoninėms komunikacijoms, duomenis perduoda per pramoninį *Ethernet*‘ą.

*PROFINET* realaus laiko valdymo tinkle duomenimis valdantysis prietaisas su pavaldžiais prietaisais apsikeičia vieno ciklo metu siusdamas kiekvienam iš magistralėje esančių prietaisų po paketą duomenų, duomenis gavę ir apdoroję pavaldūs prietaisai atgal taip pat siunčia po paketą duomenų atgal į valdantįjį prietaisą. Tokiu būdu kiekvieno duomenų apsikeitimo ciklo metu valdantysis prietaisas magistrale persiunčia tiek duomenų paketų, kiek yra pavaldžių prietaisų. *EtherCAT* atveju per vieną duomenų apsikeitimo tarp visų prietaisų ciklą persiunčiamas tik vienas duomenų paketas. Kad kompensuoti dėl didelio kiekio paketų siuntimo ir gavimo magistralėje pailgėjusį reakcijos laiką, naudojama speciali paketų paskirstymo tvarka. Į toliausiai nuo valdančiojo prietaiso esančius

pavaldžius įrenginius paketai yra siunčiami pirmiausiai, o į arčiausiai esančius prietaisus vėliausiai, taip sumažinat ciklo laiką [16].

Sistemos apibrėžimas apima šias pagrindines galimybes:

- sistema užtikrina mažą reakcijos laiką; sistemos palaikančios daugiau kaip 1000 įėjimo ir išėjimo signalų tarp 32 magistralėje esančių įrenginių, apsikeitimo duomenimis laikas yra mažesnis nei 1 milisekundė;
- sistema gali dirbti tiek su vienu, tiek su keliais valdančiais prietaisais (valdikliais arba angl. *PLC*);
- sistema turi pažangius diagnostikos įrankius, leidžiančius atlikti įrenginių, modulių ir pačio tinklo diagnostiką;
- *PROFINET IO* tinklas gali būti jungiamas įvairiais būdais, taip užtikrinant lankstų sistemos taikymą pramonėje;
- sistema gali veikti kartu su kitais tame pačiame tinkle esančiais UDT/TCP pagrindu veikiančiais prietaisais be jokių apribojimų;
- *PROFINET IO* tinklas naudoja panašų, bet pagerintą *PROFINET DP* prietaisų pritaikymo modelį, todėl atnaujinant senas *PROFINET DP* sistemas į naujas *PROFINET IO* galima panaudoti senus tinko įrenginius.

*PROFINET IO* išskiria tris įrenginių tipus [14, 15]:

- **IO valdiklis**, dažniausiai tai *PLC* valdiklis arba skaitytuvas; IO valdiklis atsakingas už su juo susijusius pavaldžius įrenginius, IP adresų priskyrimą, konfigūraciją parametrų nuskaitymą, aliarmų valdymą ir įėjimų ir išėjimų valdymą;
- **IO įrenginys**, tai pavaldus įrenginys, dažniausiai tai nutolęs valdiklio įėjimų ar išėjimų modulis; paskirtis surinkti duomenis apie sistemos būseną iš jutiklių ir pagal duodamas valdiklio komandas aktyvuoti atitinkamus išėjimo kanalus;
- **IO supervizorius**, skirtas sistemos diagnostikai, nuotoliniam valdymui, priežiūrai, parametrų keisti ar nustatyti.

### 2.2.1. *PROFINET* duomenų perdavimo terpė

*PROFINET* naudoja kabelius atitinkančius 5 (angl. *Cat 5*) arba aukštesnės kategorijos standartus. Naudojami ekranuoti keturių gyslų kabeliai. Kabeliai sudaryti iš dviejų vytų porų po du laidus. Tokio tipo kabeliai gali būti naudojami duomenų perdavimui, kurio greitis iki 100 MBit/s. Kaip ir *EtherCAT* atveju, kabelio ilgis negali viršyti 100 metrų ilgio. Esant didesniems atstumams reikia naudoti ryšio kartotuvus. *PROFINET* kabelio gyslų išdėstymas jungtyse RJ45 ir M12 ir šių gyslų reikšmės pateiktos 2 lentelėje.

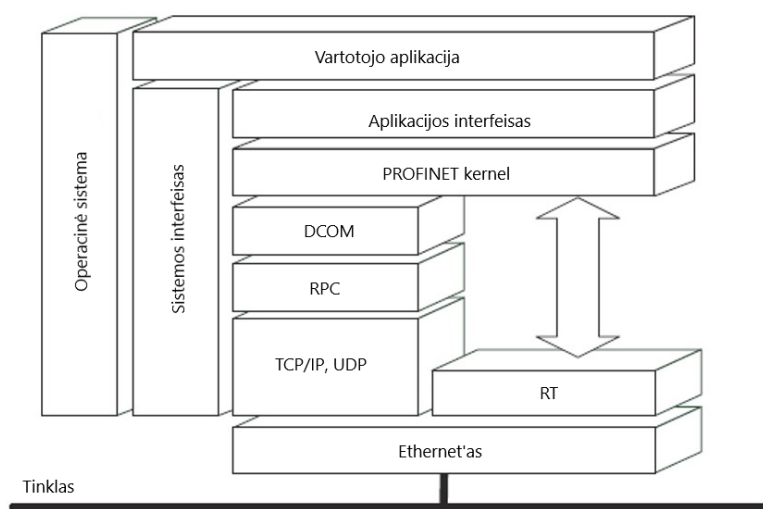
2 lentelė. *PROFINET* kabelio gyslų reikšmė ir išdėstymas jungtyje

Signalas	Laidelio spalva	Laidelio vieta RJ45 jungtyje	Laidelio vieta M12 jungtyje
TD+	Geltona	1	1
TD-	Oranžinė	2	3
RD+	Balta	3	2
RD-	Mėlyna	6	4
Kabelio ekranas	-	Prie jungties korpuso	Prie jungties korpuso

Svarbu, kad kabelio gyslos būtų sujungtos teisingai. Siunčiamų duomenų gyslos (geltona ir oranžinė) tarpusavyje turi būti susuktos į vyta porą, taip pat ir gaunamų duomenų gyslos (balta ir mėlyna turi būti tarpusavyje susuktos į vyta porą. Tinkamai susuktos siunčiamų ir gaunamų duomenų gyslų poros apsaugo nuo elektromagnetinės interferencijos, kas neleidžia atsirasti trikdžiams (angl. *Cross-Talk*). Taip pat svarbu, kad kabelio ekranas būtų tinkamai prijungtas abiejuose kabelio galuose. Visi prietaisai turi būti sujungti taip, kad vienas kabelio galas būtų įjungtas į vieno prietaiso duomenų išėjimo jungtį, o kitas kabelio galas jungtųsi į sekančio prietaiso duomenų įėjimo jungtį. Jungiant kai kuriuos *PROFINET* prietaisus, dažniausiai kompiuteris – *PROFINET* prietaisas, gali reikėti „versto“ kabelio (verstas kabelis kai viename kabelio gale laideliai 1 ir 2 sukeičiami vietomis su 3 ir 6). Tačiau toks kabelis reikalingas retai, nes beveik visi nauji *PROFINET* prietaisai automatiškai atlieka kabelio vertimą, jei to reikia [16, 17].

## 2.2.2. *PROFINET IO* komunikacijos modelis

*Ethernet*‘as veikia CSMA/CD protokolo pagrindu. Industriniame *Ethernet*‘e siekiant pagerinti jo realiojo laiko duomenų perdavimo efektyvumą naudojamas standartinis komunikacijų protokolas (TCP/IP, UDP/IP), bet jo perdavimo sparta visada priklauso nuo tinklo apkrovos, todėl sunku apskaičiuoti duomenų perdavimo vėlinimus. Atsiradus nenumatytam tinklo perkrovimui, padidės duomenų perdavimo trukmė ir duomenų apdorojimo laikas mikroprocesoriuje, dėl to suprastės realiojo laiko duomenų perdavimo charakteristikos.



13 pav. *PROFINET* komunikacijos modelis [18]

*PROFINET* technologija naudoja 13 paveiksle pateiktą *Ethernet*‘o modelį [21]. Kad sumažinti laiko sąnaudas ir valdiklio apkrovimą *PROFINET* protokolų rinkinį suprastina pašalindamas IP lygmenį, kad pagerintų realaus laiko duomenų perdavimo charakteristikas, naudodamas tarpinius lygmenis ant *Ethernet*‘o MAC sluoksnio siekiant užtikrinti didelę duomenų perdavimo spartą. Lygmens 7 funkcionalumas tiesiogiai priklauso nuo kitų tarinių lygmenų. *PROFINET IO* naudoja keletą savų sluoksnių ir standartų [15].

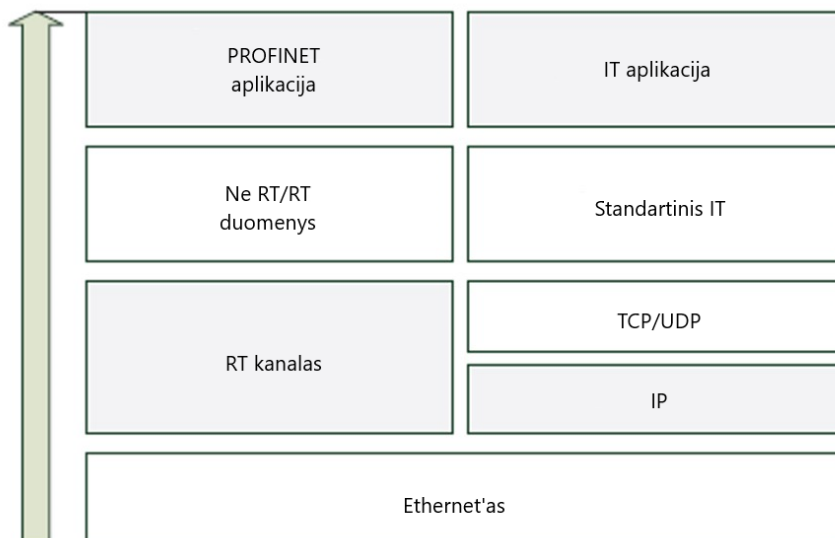
Komunikacija tarp *PROFINET* prietaisų paremta MAC adresais ir suderinama su TCP/IP protokolu.

Kad atskirti TCP/IP duomenis nuo RT duomenų, naudojamas *ethertipas*  $0 \times 8892$  identifikuojantis realaus laiko PDU (angl. *Protocol Data Units*).

13 paveiksle pateiktas *PROFINET* technologijoje naudojamas komunikacijos modelis.

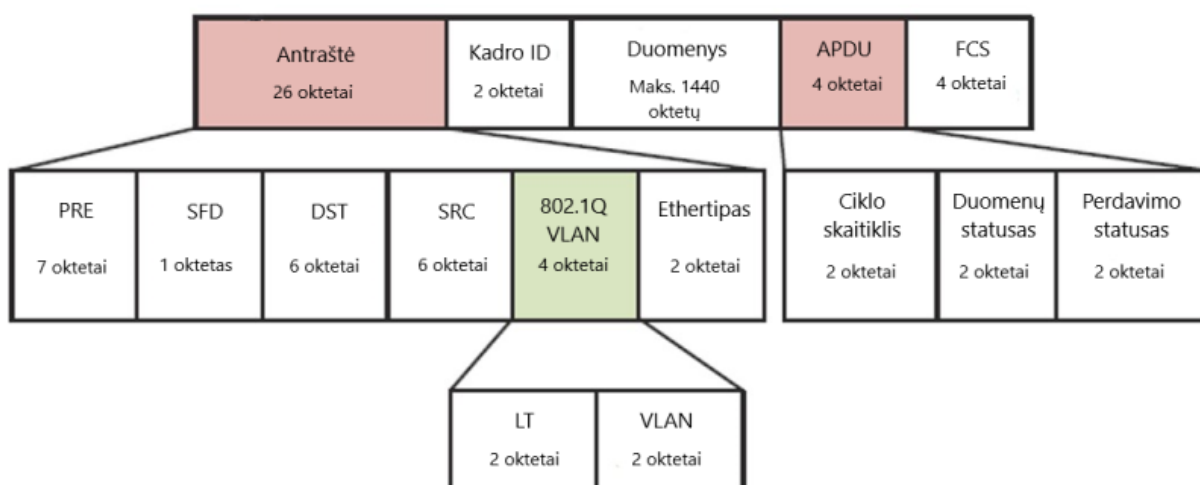
### 2.2.3. *PROFINET* RT duomenų paketo struktūra

*PROFINET* RT komunikacijos protokolo veikimas pagrįstas *Ethernet*’o standartu, aprašytu IEEE 802.3 [22]. Jo struktūra pavaizduota 14 paveiksle.



14 pav. *PROFINET* RT protokolo struktūra [18]

Duomenų kanalai gali būti išskirti į dvi kategorijas: RT kanalus ir standartinius kanalus. Įrenginių parametriniai, diagnostikos ir ne realaus laiko duomenys perduodami standartiniais kanalais. RT (realaus laiko) ir IRT periodiniai duomenys, kartu su aliarmų signalais yra perduodami per RT kanalus. Iš 14 paveikslo matyti, kad ne RT duomenų kanalas suderinamas su IT standartu, kas leidžia *PROFINET* vertikalų suderinamumą su kito lygio įrenginiais, t. y. gamybiniame lygyje naudojamus įrenginius leidžia suderinti su ofiso įrenginiais naudojant tą pačią duomenų perdavimo magistralę.



15 pav. *PROFINET* IO duomenų kadras [18]

*PROFINET* IO kadras naudoja MAC adresą, kad nustatyti iš kur ir į kur siusti duomenis. Realaus laiko duomenys yra perduodami *Ethernet*’o kadru turinčiu prioritetinę reikšmę. Prioriteto reikšmė

nurodoma VLAN žymoje (angl. *Tag*), ji žymima skaičiumi nuo 0 iki 7 (0 mažiausiais prioritetas, 7 didžiausiais). Remiantis 802.1Q, VLAN žyma prie standartinio *Ethernet*‘o kadro prideda 4 baitus ir yra žymima reikšme 0x8100. Žyma užima 16 bitu lauką, kuris sudarytas iš: 3 bitų lauko nurodančio prioritetą, 1 bito lauko nurodančio kurios kadrus būtų galima išmesti esant perkrovoms ir 12 bitų VLAN identifikatoriaus. IO duomenys yra perduodami naudojant prioriteto žymą 6 arba 7. *PROFINET* tinkui priskiriamas dviejų oktetų *Ethernet*‘o kadro laukas žymimas  $0 \times 8892$ , kuris naudojamas išskirti realaus laiko duomenis. Toliau einantis vartotojo duomenų laukas yra tarp 40 ir 1440 oktetų *PROFINET IO* apima tam tikrą kiekį APDU (angl. *Application Protocol Data Units*), susidedantį iš dviejų oktetų ciklų skaitiklio, 2 dviejų oktetų duomenų statuso ir dviejų oktetų duomenų perdavimo statuso.

Ciklo skaitiklis turi siuntimo laikrodžio vertę, kuri yra 31,25 mikrosekundės. Šis skaitiklis naudojamas įvertinti ar duomenys priimti reikiamu laiku, duomenis gaunančiame prietaise. Duomenų statusas ir duomenų siuntimo statusas naudojami nustatyti ar duomenys gavėją pasiekė tinkami, neiškraipyti [20, 21].

#### 2.2.4. *PROFINET IRT* duomenų paketo struktūra

*PROFINET IRT* (angl. *Isochronous Real Time*) priklauso realaus laiko valdymo tinklų klasei 3. Naudojamas aukščiausio prioriteto komunikacijoms, turi būti griežtai izochroninis (visada trunkantis tiek pat laiko). Šios klasės tinklo duomenų perdavimo kadro struktūra pateikta 16 paveiksle. Skirtingai nei RT protokolas, IRT protokolas suprastina duomenų perdavimą pašalindamas APDU ir VLAN žymos laukus.

Duomenims realiu laiku perduoti, būtina tinkamai paskirstyti perduodamus duomenis ir perdavimui skirtą laiką. Paprastai IRT sistemoje duomenys perduodami cikliška. Kad būtų užtikrintas tinkamas laikas kada prietaisas gauna, o kada siunčia duomenis, reikalingas precizinis laiko sinchronizavimas.

PRE	SFD	DST	SRC	Ethertipas	Kadro ID	Duomenys	FCS
7 oktetai	1 oktetas	6 oktetai	6 oktetai	2 oktetai	2 oktetai	Maks. 1490 oktetų	4 oktetai

16 pav. *PROFINET IRT* duomenų kadro struktūra [18]

*PROFINET IRT* naudoja PTCP (angl. *Precision Transparent Clock Protocol*), kad garantuoti griežtą savalaikiškumą. PTCP yra naudojamas, kad nustatyti laiką perduodant duomenis [22, 23].

#### 2.2.5. *PROFINET* tinklo jungimo būdai

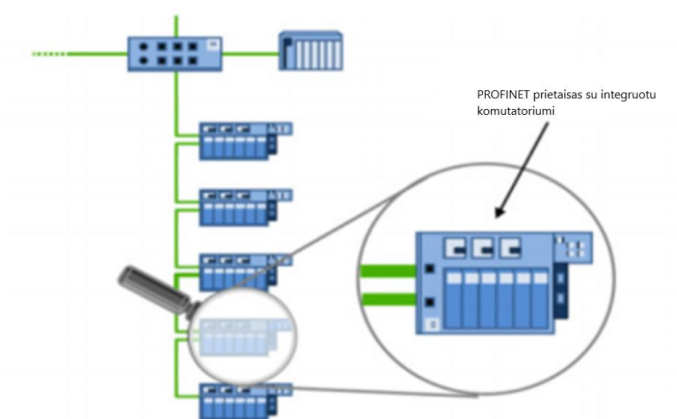
*PROFINET* prietaisai visada yra sujungti per tinklo komutatorių. Tinklo komutatorius automatiškai atlieka kabelio porų vertimą priklausomai ar tai reikalinga. Kaip jau minėta anksčiau, kabelio ilgis nuturi viršyti šimto metrų ilgio. Esant didesniems atstumams reikia naudoti tinklo kartotuvą, komutatorių arba esant labai dideliems atstumams naudoti specialius *PROFINET* prietaisus kurie tarpusavyje gali būti sujungti optiniu kabeliu. Jungiant prietaisus optiniu kabeliu, atstumas tarp prietaisų gali siekti kelis kilometrus.

*PROFINET* realaus laiko valdymo tinkle prietaisai gali būti sujungti trimis skirtingais jungimo būdais [23]:

- nuosekliu;
- žvaigždės tipo;
- medžio tipo.

Dauguma *PROFINET* prietaisų turi integruotą dviejų jungčių tinklo komutatorių, kas leidžia šiuos prietaisus jungti į nuoseklią grandinę. Kadangi kiekvienas prietaisas turi integruotą vidinį tinklo komutatorių, tai leidžiamas maksimalus šimto metrų atstumas tarp kiekvieno nuosekloje grandinėje esančio *PROFINET* prietaiso.

Tokio tinko jungimo tipo privalumas toks, kad esant didesniems nei šimto metrų atstumams tarp tinklo komutatoriaus ar valdančiojo prietaiso ir pavaldžių *PROFINET* prietaisų, sutaupoma daug kabelio ir nereikia daugybės tinklo komutatorių ar kartotuvų, norint užtikrinti, kad nei vieno iš kabelių ilgis nėra didesnis nei 100 metrų. Tačiau šis jungimo būdas pasižymi nedideliu patikimumu, nes pažeidus, kabelį tarp dviejų nuosekloje grandinėje esančių prietaisų, prarandamas ryšys su visais už kabelio pažeidimo vietos esančiais prietaisais. Paveiksle numeris 17 pateiktas nuoseklaus *PROFINET* tinklo jungimo būdas [23].



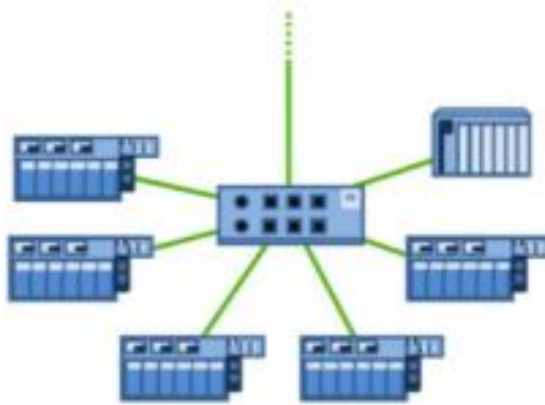
**17 pav.** Nuoseklus *PROFINET* tinklo jungimo būdas [27]

Nuoseklaus *PROFINET IO* tinklo jungimo tipo vėlinimo laikas apskaičiuojamas pagal formulę:

$$T_{\text{nuoseklaus}} = n \times T_c^f + T_c^p + T_d^f + T_p; \quad (16)$$

- čia  $T_c^f$  – paketo išsiuntimo iš valdančio prietaiso užlaikymo trukmė;  
 $T_c^p$  – užlaikymo trukmė kabelyje;  
 $T_d^f$  – paketo išsiuntimo iš pavaldaus prietaiso užlaikymo trukmė;  
 $T_p$  – maksimalus PHY ( $Rx + Tx$ ) užlaikymas;  
 $n$  – pavaldžių prietaisų kiekis.

Žvaigždės tipo *PROFINET* tinklo jungimas gaunamas naudojant daugiau nei dvi jungtis turinčius tinklo komutatorius. Šio tinklo privalumas, kad galima atjungti, bet kurį iš pavaldžių tinklo prietaisų, pavyzdžiui remontui ar keisti į kitą, nenutraukiant ryšio tarp valdančiojo prietaiso ir kitų pavaldžių prietaisų. 18 paveiksle pateiktas *PROFINET* tinklo jungimo žvaigžde pavyzdys [23].



**18 pav.** Žvaigždės tipo *PROFINET* tinklo jungimo būdas [24]

Žvaigždės jungimo tipo *PROFINET IO* tinklo vėlinimo laikas apskaičiuojamas pagal formulę:

$$T_{\text{žvaigždės}} = n \times T_c^f + 2 \times T_c^p + 2 \times T_d^f + 2 \times T_p; \quad (17)$$

čia  $T_c^f$  – paketo išsiuntimo iš valdančio prietaiso užlaikymo trukmė;

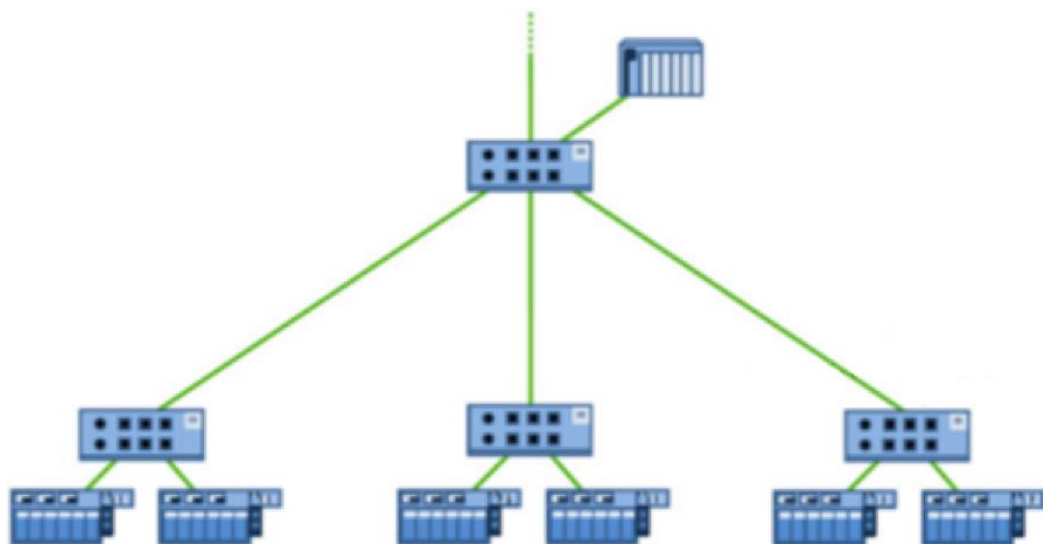
$T_c^p$  – užlaikymo trukmė kabelyje;

$T_d^f$  – paketo išsiuntimo iš pavaldaus prietaiso užlaikymo trukmė;

$T_p$  – maksimalus PHY ( $Rx + Tx$ ) užlaikymas;

$n$  – pavaldžių prietaisų kiekis.

Medžio tipo tinklo jungimo būdas *PROFINET* tinkle gaunamas sujungiant kelis tinklo komutatorius, prie kurių prietaisai jungiami žvaigždės tipo tinklo jungimu, į vieną bendrą tinklo komutatorių. Jungiant tokiu būdu tinklas išskirstomas į lygmenis. Toks tinklo jungimo būdas pavaizduotas 19 paveiksle [23].



**19 pav.** Medžio tipo *PROFINET* tinklo jungimo būdas [24]

Jungiant tokiu būdu tipo *PROFINET IO* tinklo vėlinimo laikas apskaičiuojamas pagal formulę:



$$T_{medžio\ tipo} = n \times T_c^f + 4 \times T_c^p + 2 \times T_d^f + 4 \times T_p; \quad (18)$$

čia  $T_c^f$  – paketo išsiuntimo iš valdančio prietaiso užlaikymo trukmė;

$T_c^p$  – užlaikymo trukmė kabelyje;

$T_d^f$  – paketo išsiuntimo iš pavaldaus prietaiso užlaikymo trukmė;

$T_p$  – maksimalus PHY ( $Rx + Tx$ ) užlaikymas;

$n$  – pavaldžių prietaisų kiekis.

### 3. Tyrimo metodai

Šio skyriaus pirmoje dalyje sudaromi metodai skirti tirti *EtherCAT* ir *PROFINET* tinklų laikinius parametrus, atsižvelgiant į tinklų topologijas ir aptarnaujamų tinklo prietaisų skaičių. Tyrimui naudojama *MATLAB* programa, mokslinės literatūros analizės dalyje pateiktos formulės, *EtherCAT* ir *PROFINET* prietaisų gamintojų pateikiami duomenys apie šiuos prietaisus, taip pat analitinėje dalyje išnagrinėtų mokslinių šaltinių duomenys, tokie kaip *EtherCAT* ir *PROFINET* tinklo jungimo būdai, kiekvieno iš šių jungimo būdų parametrai, formulės, skirtos apskaičiuoti kiekvieno iš galimų tinklo jungimo būdų minimalų duomenų apsikeitimo laiką.

Šio skyriaus antroje dalyje sudaromi metodai skirti tirti *EtherCAT* realaus laikos komunikacinio tinklo greitaveiką ir vėlinimo laikus. Šiam tikslui naudojamas nešiojamas kompiuteris su *Windows 7* operacine sistema, *MATLAB 2016b* programine įranga ir tinklo plokšte suderinama su *EtherCAT* prietaisais. Kompiuteryje taip pat suinstaliuota *Beckhoff TwinCAT 3* programa ir paprogramė *TE1400*, skirta komunikacijai tarp Matlab Simulink paprogramės ir *Beckhoff TwinCAT 3* programos.

#### 3.1. Metodikos, skirtos minimaliam duomenų apsikeitimo laikui *EtherCAT* ir *PROFINET* tinkluose tirti, sudarymas

Tyrimo skaičiavimai atliekami vadovaujantis skyreliuose pavadinimais „*EtherCAT* tinklo jungimo būdai“ ir „*PROFINET* tinklo jungimo būdai“ pateiktomis formulėmis ir imant, kad visi tinkle sujungti prietaisai yra to paties tipo ir to paties gamintojo. Taip pat skaičiavimuose naudojamas vidutinis laido ilgis tarp prietaisų imant, kad visi prietaisai sujungti vienodo ilgio ir tipo kabeliais. Tyrime naudojami trys laido ilgiai: 10, 50 ir 100 metrų. *EtherCAT* prietaisų minimalaus duomenų apsikeitimo laikui apskaičiuoti naudojami *Beckhoff* gamintojo pateikti duomenys apie šio gamintojo valdančiuose ir pavaldžiuose prietaisuose naudojamus procesorius [12]. Šie duomenys pateikti 3 lentelėje. *PROFINET* prietaisų minimalaus duomenų apsikeitimo laikui apskaičiuoti naudojami duomenys pateikti moksliniuose šaltiniuose [29, 30].

3 lentelė. Gamintojo pateikti pavaldžių *EtherCAT* prietaisų įrangos vėlinimo laikai

Žymėjimas	Paaiškinimas	Reikšmė, ns
$T_{valdiklio}$	Procesorius vėlinimas	460
$t_{MMp}$	Iš MII į MII apdorojimo vėlinimas	305
$t_{MMf}$	Iš MII į MII perdavimo vėlinimas	265
$t_{Mep}$	Iš MII į E-bus apdorojimo vėlinimas	280
$t_{MEf}$	Iš MII į E-bus perdavimo vėlinimas	260
$t_{EMp}$	Iš E-bus į MII apdorojimo vėlinimas	170
$t_{EMf}$	Iš E-bus į MII perdavimo vėlinimas	150
$t_{EEp}$	Iš E-bus į E-bus apdorojimo vėlinimas	150
$t_{EEf}$	Iš E-bus į E-bus perdavimo vėlinimas	130
$t_{RX}$	Duomenų gavimo vėlinimas	250
$t_{TX}$	Duomenų transliavimo vėlinimas	250

3 lentelėje pateikti duomenys bus naudojami teoriniams minimalaus vėlinimo laiko skaičiavimams, esant skirtingiems tinklo jungimo būdams.

### **3.2. Metodai, skirti *EtherCAT* realaus laiko komunikacinio tinklo greitaveikai ir vėlinimo laikams nustatyti *MATLAB* programine įranga**

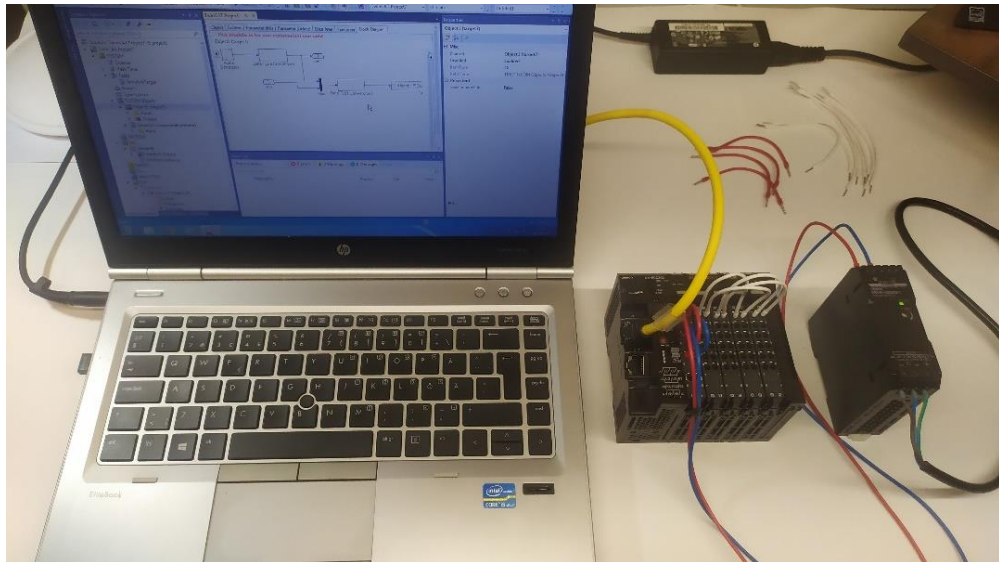
Šioje dalyje sudaromi tyrimo metodai skirti matuoti, vėlinimo laiką nuo komandos išsiuntimo iki jos įvykdymo. Šiam tyrimui naudojamas nešiojamas kompiuteris su *Windows 7* operacine sistema, *MATLAB 2016b* programine įranga ir tinklo plokšte suderinama su *EtherCAT* prietaisais.

Tyrimui atlikti *MATLAB* programos *Simulink* paprogramėje, naudojantis *Simulink real time* skiltyje esančiais blokais, sukuriamas *EtherCAT* valdančiojo prietaiso (angl. *EtherCAT Master*) modelis leisiantis nešiojamu kompiuteriu valdyti *EtherCAT* prietaisus nenaudojant specialaus pramoninio valdiklio palaikančio *EtherCAT* valdymo protokolą.

Tyrimo standą sudaro anksčiau aprašytas kompiuteris, *OMRON* gamintojo *EtherCAT* tinklo *coupler*'is *NX-ECC203*, diskretinių įėjimų modulis *NX-ID3417* turintis 4 įėjimo jungtis, diskretinių išėjimų modulis *NX-OD3256* turintis 4 išėjimo jungtis, *OMRON* gamintojo 24 voltų 30 vatų nuolatinės srovės maitinimo šaltinis *S8VK-G03024*, *OMRON* gamintojo 50 centimetrų ilgio ekranuotas *EtherCAT* duomenų perdavimo kabelis, atitinkantis *Cat. 6a* duomenų perdavimo kategoriją, su RJ 45 kabelio jungtimis užspaustomis abiejuose galuose.

Šiam tyrimui sujungiamas modeliavimo stendas. Prie kompiuterio prijungiamas *EtherCAT* tinklo *coupler*'is prijungiant vieną *EtherCAT* kabelio galą į kompiuterio *Ethernet* jungtį, o kitą kabelio galą įjungiant į *NX-ECC 203* prietaiso *EtherCAT* ryšio įėjimo jungtį. Prie *EtherCAT coupler*'io prijungiamas vienas diskretinis įėjimo ir vienas diskretinis išėjimo modulis. Į *NX-ECC 203* prietaiso *A1* jungtį įjungiamas nuo maitinimo šaltinio + 24 voltų gnybto einantis raudonos spalvos laidas, o į *A3* jungtį įjungiamas nuo šaltinio 0 voltų gnybto einantis mėlynos spalvos laidas. Taip pat jungtis pavadinimu *B1* sujungiama su *A5*, o jungtis *B3* su *B5*, tokiu būdu prijungiant prie maitinimo ne tik patį *EtherCAT coupler*'į, bet ir prie jo prijungtus modulius (jungtys *A1*, *B1* ir *A3*, *B3* skirtos *EtherCAT coupler*'io maitinimui, o jungtys *A5* ir *B5* prie *Coupler*'io prijungtų modulių maitinimui). Esant didesniam įėjimo ir išėjimo modulių kiekiui ar norint juos maitinti iš atskiro maitinimo šaltinio reikalingi papildomi maitinimo moduliai, įsisegantys tarp modulių, taip juos atskiriant. Prie kompiuterio prijungtų, tyrimui skirtų *EtherCAT* prietaisų nuotrauka pateikta 20 paveiksle.

*EtherCAT* prietaisų prijungti prie modelio tiesiogiai per kompiuterio *Ethernet* jungtį negalima. Modeliui su prietaisais sujungti reikalingas simuliacinio kompiuteris arba programa (angl. *Target*), kurioje vyksta modelio simuliacija realiu laiku, o *MATLAB Simulink* programa, esanti modeliavimo kompiuteryje (angl. *Modeling PC*), skirta atvaizduoti simuliacinio rezultatus (*Simulink* paprogramė tokiu atveju veikia kaip vizualizavimo sąsaja). Šiuo atveju naudojama *Beckhoff TwinCAT 3* programa, kurioje įdiegti *MATLAB Simulink* palaikymo moduliai *TE1400 TC3 Target for Simulink*, *TE1401 TC 3 Target for MATLAB*, *TE1410 TC 3 Interface For MATLAB Simulink*, *TC1320 TC 3 C++/MatSim*. *EtherCAT* prietaisai jungiami prie *TwinCAT* programos per kompiuterio *Ethernet* lizdą. Programa realiu laiku apsikeitinėja duomenimis su *EtherCAT* prietaisais ir šiuos duomenis perduoda į *MATLAB Simulink* programą, kurioje šiuos duomenis galima peržiūrėti ir išsaugoti, taip pat keisti sudaryto modelio blokų parametrus.



**20 pav.** Prie kompiuterio prijungtas pavaldis prietaisas

Nuo modelio sudarymo iki jo sėkmingo prijungimo prie *TwinCAT 3* programos ir simuliacinio rezultato gavimo, galima suskirstyti į tris dalis:

1. *EtherCAT* prietaisų prijungimas prie kompiuterio;
2. *MATLAB Simulink* modelio sudarymas ir modelio kompiliavimo parametrų parinkimas, kad modelį būtų galima tinkamai sukompiliuoti į *Beckhoff TwinCAT 3* programai reikiamą formatą.
3. Sukompiliuoto *MATLAB Simulink* modelio įkėlimas į *TwinCAT 3* programą. Fizinį išėjimą ir įėjimą priskyrimas modelyje esantiems blokams. *TwinCAT 3* programos prijungimas prie *MATLAB Simulink* programos ir modeliavimo rezultatų gavimas.

### 3.2.1. Tiriamao EtherCAT tinklo sudarymas

Kad tarp *MATLAB* programos *Simulink* paprogramės ir *Beckhoff TwinCAT 3* programos vyktų komunikacija, svarbu tam tikra tvarka įdiegti šias atitinkamų versijų programas: kompiuteryje, kuriame įdiegta *Windows 7* operacinė sistema pirmiausia įdiegiama *MATLAB 2015b* programos versija, kurioje yra *MATLAB Coder*. *MATLAB* programos paprogramėje *Simulink* sudarius modelį, jį reikės sukompiliuoti, kad *TwinCAT 3* programa galėtų tinkamai jį atpažinti, tam reikalingas C++ kompiliatorius. Oficialiame *MATLAB* programos internetiniame puslapyje galima rasti sąrašą visų konkrečiai versijai tinkančių C+ ir C++ kompiliatorių. Kadangi *Beckhoff* kompanijos rekomenduojamas kompiliatorius yra *Microsoft Visual Studio Professional*. Todėl atsižvelgiant į *Beckhoff* ir *MATLAB* kompanijų rekomendacijas pasirinkta 2015 metų *Microsoft Visual Studio Professional* programa.

Taip pat svarbu paminėti, kad pagal numatytuosius nustatymus ši programa įdiegiama be C+ ir C++ kompiliatoriaus, todėl sėkmingai įdiegus šią programą, reikia paleisti instaliavimo programą dar kartą ir pasirinkus *modify* režimą, pasirinkti punktus su C+ ir C++ kompiliatoriumi. Kitu atveju sukūrus *Simulink* modelį ir paspaudus mygtuką pavadinimu *build model*, bus indikuojama klaida ir pranešama apie nesėkmingą modelio generavimą.

Atlikus prieš tai paminėtus veiksmus iš oficialaus *Beckhoff* kompanijos internetinio puslapio parsisiunčiama *TwinCAT 3.1* programa ir paleidžiama instaliavimo programa. Diegimo metu viename

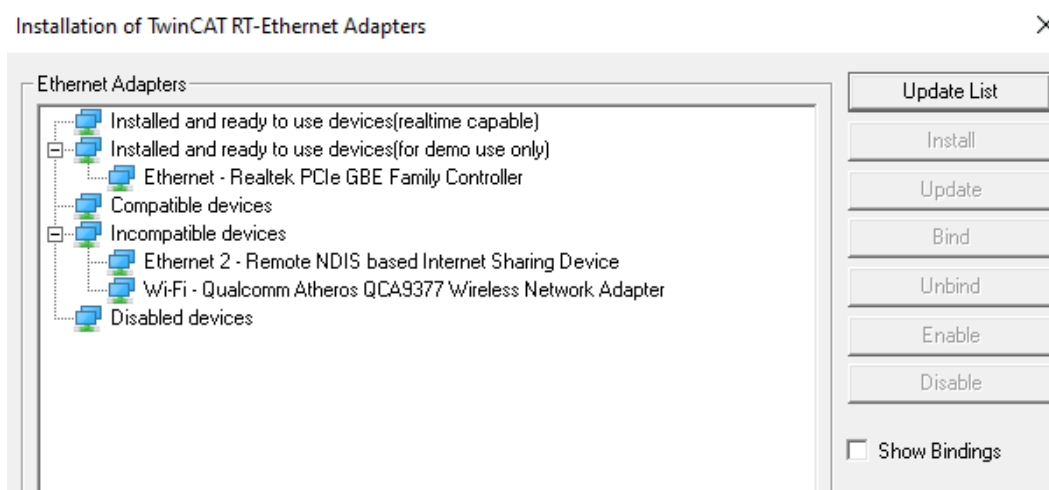
iš programos diegimo etapų, kai programa klausia ar integruoti *TwinCAT 3* programą į *Microsoft Visual Studio* programą, pasirenkamas laukelis reiškiantis sutikimą (angl. *Yes*). Kitu atveju, neintegravus *TwinCAT* programos į *Microsoft Visual Studio* programą, sugeneruotas *Simulink* modelis bus nekorektiškai atvaizduojamas *TwinCAT 3* programoje ir komunikacija tarp šios ir *MATLAB* programų bus negalima.

Kad *MATLAB* programos paprogramėje sugeneruotas modelis būtų atvaizduojamas *TwinCAT 3* programoje reikalingas šios programos papildomas modulis *TE1400*. Šis modulis taip pat atsiunčiamas iš oficialios gamintojo svetainės ir įdiegiamas. Taip pat šį modulį atskirai reikia įdiegti ir *MATLAB* programoje. Tai padaroma administratoriaus teisėmis paleidus *MATLAB* programą, kairėje programos pusėje esančiame lange iš pateiktų aplankų suradus aplanką, kuriame įdiegtas *TwinCAT* programos modulis *TE1400* ir jį atidarius, dešiniu pelės klavišu paspaudžiama ant *SetupTwinCatTarget.p* ir pasirenkama komanda vykdyti (angl. *Run*).

Norint prijungti *EtherCAT* prietaisus prie kompiuterio *Beckhoff TwinCAT 3* programa, pirmiausia reikia užtikrinti, kad programos *ESI* (angl. *EtherCAT Slave Information File*) failų bibliotekoje yra norimų prijungti prietaisų konfigūraciniai failai. Jei prie programos jungiamas *Beckhoff* gamintojo prietaisas, konfigūracinio failo nereikia, kadangi programoje jau yra visų šio gamintojo prietaisų konfigūraciniai failai, kitu atveju failas reikalingas.

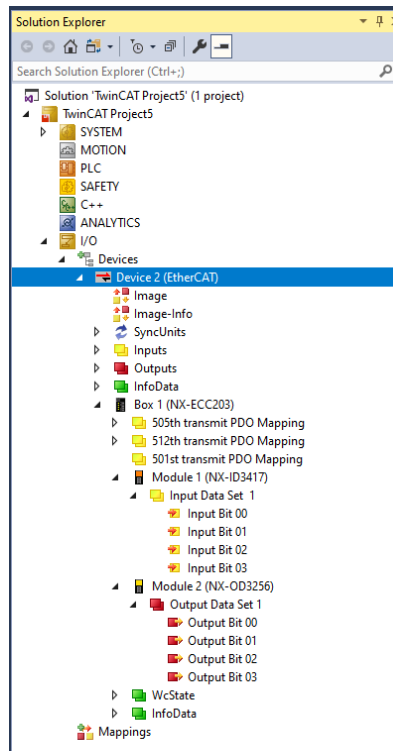
Paleidus *TwinCAT 3* programą ir pasirinkus sukurti naują projektą. Atidaromas pradinis programos langas, kurio kairėje pusėje pateikta pasirinkimų skiltis. Norint nuskaityti prie kompiuterio prijungtus *EtherCAT* prietaisus, praskleidžiamas pasirinkimų laukas pavadinimu *I/O*. Dešiniu pelės klavišu paspaudus ant atsiradusio skirtuko pavadinimu *Devices*, pateikiama keletas galimų pasirinkimų. Šiuo atveju pasirenkamas punktas *Scan* ir pradeda prijungtų *EtherCAT* prietaisų paieška.

Norint atlikti prijungtų prietaisų paiešką prieš tai reikia užtikrinti, kad kompiuterio *Ethernet* tinklo plokštė būtų suderinama su realaus laiko valdymo prietaisais. Tą padaryti galima *TwinCAT 3* programos nustatymų punkte skirtame *Ethernet* adapterių nustatymams. Nustatymo lango nuotrauka pateikta 21 paveiksle.



21 pav. Su *EtherCAT* prietaisais suderinamų tinklo adapterių sąrašas

*TwinCAT 3* programai baigus prie kompiuterio prijungtų prietaisų paiešką, jų sąrašas pateikiamas programos skiltyje pavadinimu *I/O* (žr. 22 pav.).

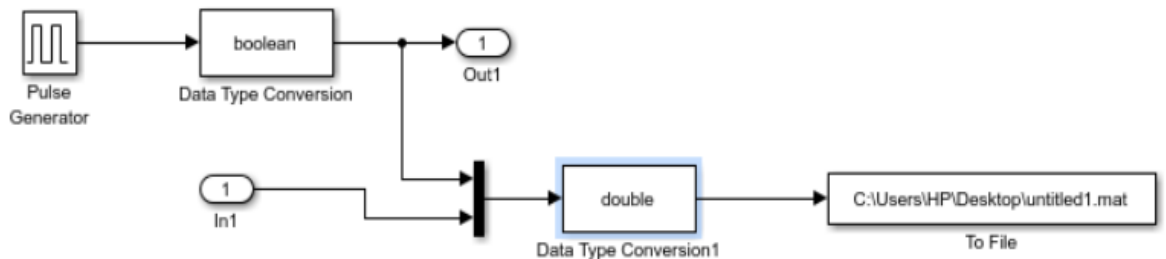


22 pav. Programos aptiktų prietaisų sąrašas

Programos aptiktų, tyrimui naudojamų prietaisų sąrašas pateiktas 22 paveiksle.

### 3.2.2. Signalų vėlinimo modelio sudarymas *MATLAB Simulink* aplinkoje

Tyrimui sudaromi trys modeliai. Pirmasis skirtas diskretinio signalo vėlinimui matuoti ir modelio bei programų išbandymui, nustatymų parinkimui. Antrasis ir trečiasis skirti tirti diskretinio signalo vėlinimus naudojant skirtingus diskretinių įėjimų ir išėjimų kiekius, taip pat naudojant skirtingus įėjimų/ išėjimų modulių kiekius. Diskretinio signalo perdavimo vėlinimui *EtherCAT* tinkle tyrimui sudaryto bandomojo *MATLAB Simulink* modelio nuotrauka pateikta 23 paveiksle.



23 pav. Tyrimui naudojamas bandomasis modelis

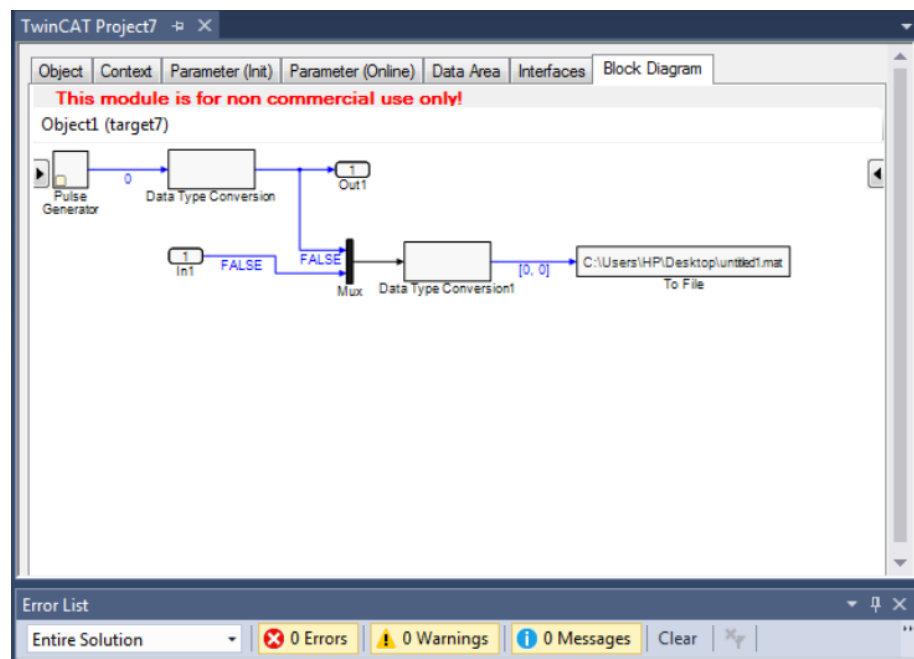
Modelyje naudojami tokie blokai:

- impulsų generavimo blokas pavadinimu *Pulse generator*, skirtas norimu dažniu ir užduota amplitudė generuoti signalą;

- įėjimo blokas pavadinimu *In1*; prie šio bloko *TwinCAT 3* programoje priskiriamas fizinis valdiklio įėjimo modulio įėjimas; aktyvavus šiam blokui priskirtą modulio įėjimą, šis blokas generuoja *boolean* duomenų tipo signalą.
- išėjimo blokas pavadinimu *Out1*; prie šio bloko *TwinCAT 3* programoje priskiriamas fizinis valdiklio išėjimo modulio išėjimas; kai *MATLAB* programos paprogramėje *Simulink* prie šio bloko prijungiama *boolean* duomenų tipo signalo reikšmė 1, aktyvuojamas šiam blokui priskirtas fizinis išėjimas;
- modelyje norint signalus sujungti į vieną naudojamas blokas pavadinimu *Mux*;
- simuliacijos duomenų įrašymui, kad vėliau būtų galima peržiūrėti rezultatus, naudojamas blokas pavadinimu *To File*; šiame bloke nurodoma pageidaujama failo išsaugojimo vieta ir failo pavadinimas;
- Modelyje naudojami du duomenų tipo keitimo moduliai; vienas naudojamas pulsų generatoriaus generuojamą signalą pakeisti į *boolean* signalo tipą, kitas iš *Mux* bloko gaunamą *boolean* duomenų tipo signalą keisti į *To file* blokui tinkamą signalo tipą *double*.

### 3.2.3. *TwinCAT 3* programos prijungimas prie *MATLAB Simulink* paprogramės ir modeliavimo rezultatų gavimas

*Matlab* programos paprogramėje *Simulink* sugeneruotas modelis į *TwinCAT 3* programą įkeliamas praskleidus kairėje programos pusėje esantį punktą *system*, dešiniu pelės klavišu paspaudus ant punkto *TcCOM objects* ir pasirinkus *add new item*. Atlikus šiuos veiksmus, atsidaro naujas programos langas pavadinimu *Insert TcCOM Object*. Praskleidžiamas punktas pavadinimu *TE1400 Module Vendor*, po to praskleidžiamas punktas pavadinimu *generated Modules (TE1400 1.2)* ir pasirenkamas failas, kurio pavadinimas sutampa su *MATLAB* programos paprogramėje sukurto *Simulink* failo pavadinimu, ir spaudžiama „gerai“. Norint patikrinti ar į programą įkeltas modelis atitinka *Simulink* paprogramėje sudarytą modelį, reikia du kartus kairiu pelės klavišu paspausti ant punkto *object1* (failo vardas), esančio *TcCOM Objects* skiltyje ir po to viduriniame programos lauke pasirinkti skiltį *Block diagram*. Į programą įkelto modelio ekrano nuotrauka pateikta 24 paveiksle.



24 pav. Tyrimui naudojamas modelis įkeltas į simuliacijos programą

Modelyje iš karto galima matyti reikšmes ateinančias ir išeinančias iš kiekvieno bloko. Šiuo atveju naudojamas diskretinis signalas, todėl galimos reikšmės yra dvi: 1 ir 0. Kadangi simuliacija dar nepaleista, visos reikšmės išlieka pastovios.

Prieš paleidžiant simuliaciją *Beckhoff TwinCAT 3* programoje dar atliekami šie veiksmai: sukuriama tuščia *PLC* programa, sukuriama užduotis (angl. *Task*), įterpiamas dar vienas *TcCOM* objektas skirtas duomenų perdavimui iš *TwinCAT* programos į *MATLAB* programą.

Kairėje *TwinCAT 3* programos pusėje esančioje pasirinkimų skiltyje dešiniu pelės klavišu paspaudus ant punkto pavadinimu užduotys (angl. *Tasks*) pasirenkamas punktas pavadinimu „ pridėti naują ...“ (angl. *Add New Item*). Atsiradusiame naujame lange pavadinimu „įterpti užduotį“ (angl. *Insert Task*) pasirenkamas punktas *TwinCAT Task*, sukuriamas pavadinimas, šiuo atveju, *SimulinkTarget* ir spaudžiama „gerai“. Vidurinėje programos skiltyje atsiradusiame lange, pakeičiamas ciklo laikas iš numatytojo 10 milisekundžių į 1 milisekundės. Įterpus naują užduotį reikia pakeisti į programą įterpto *Simulink* modelio parametrą pavadinimu „užduotis“. Tai atliekama du kartus kairiu pelės klavišu paspaudus ant punkto *object1* (failo vardas), esančio *TcCOM Objects* skiltyje ir po to viduriniame programos lauke pasirinkti skiltį *Context*. Naujai pasirodžiusiame lauke išskleidžiamas apačioje esantis punktas pavadinimu *task* ir iš pateikto sąrašo pasirenkamas pavadinimas sutampantis su sukurtos užduoties pavadinimu.

Sekantis punktas - *PLC* programos įterpimas. Tai atliekama kairėje *TwinCAT 3* programos pusėje, esančioje pasirinkimų skiltyje dešiniu pelės klavišu paspaudus ant punkto pavadinimu *PLC* ir pasirinkus punktą pavadinimu „ pridėti naują“. Iš galimų dviejų pasirinkimų, kurie yra arba „standartinis *PLC* projektas“ arba „tuščias *PLC* projektas“, pasirinktas antrasis, „tuščias projektas“. Projekto vardas nėra svarbus tolesniems veiksams ir galutiniams modeliavimo rezultatams, todėl projekto pavadinimas paliktas numatytasis.

Norint išsaugoti simuliacijos rezultatus reikia įterpti *TcCOM* objektą pavadinimu *TcExtendedFileWriter*, be jo simuliacijos rezultatai neperduodami į išorines programas, šiuo atveju tai programos *MATLAB Simulink* paprogramė. Šis objektas pridėtas kairėje *TwinCAT 3* programos pusėje, esančioje pasirinkimų skiltyje dešiniu pelės klavišu paspaudus ant punkto pavadinimu *TcCOM Objects* ir pasirinkus punktą pavadinimu *add new item*. Naujai pasirodžiusiame programos dialogo lange išskleidžiamas punktas pavadinimu *Beckhoff Automation*, naujai pasirodžiusiame punkte pavadinimu *Extended file Access*, pasirenkamas punktas pavadinimu *TcExtendedFileWriter* ir spaudžiama „gerai“.

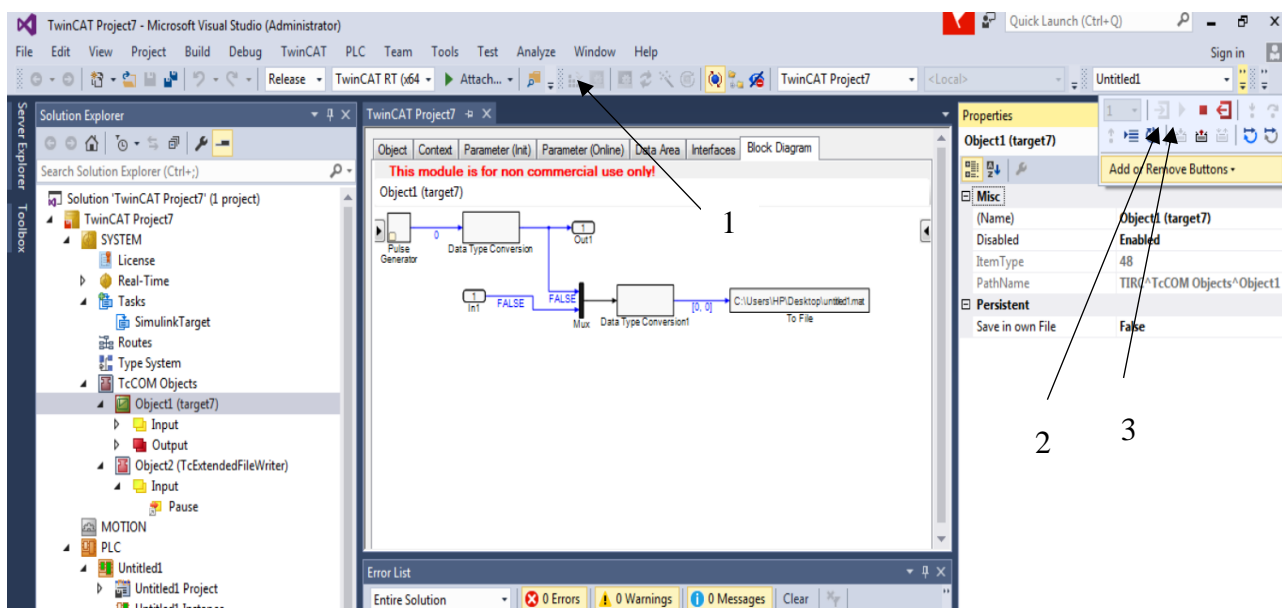
Atlikus šiuos veiksmus pagrindiniame *TwinCAT 3* programos lange spaudžiamas mygtukas pavadinimu „patvirtinti konfigūraciją“. Šis mygtukas 25 paveiksle pažymėtas pirmu numeriu. Patvirtinus konfigūraciją spaudžiamas mygtukas „prisijungti“, pažymėtas antru numeriu, o po to spaudžiamas mygtukas pavadinimu „pradėti“, kuris 25 paveiksle pažymėtas numeriu 3.

Sėkmingą *TwinCAT* programos komunikaciją su *EtherCAT* prietaisais galima įvertinti stebint indikacines lemputes ant prietaiso. Šiuo atveju naudojamas *OMRON NX-ECC203 EtherCAT coupler* is, kurio *EtherCAT* įėjimo jungties būseną galima įvertinti stebint indikatorius pavadinimais *RUN*, *TS* ir *L/A IN*.

Jei naudojamo prietaiso konfigūracija atitinka programoje naudojamą konfigūraciją, tai indikatorius *TS* šviečia žalia spalva. Jei programoje paspaudus mygtuką pavadinimu „pradėti“, indikatorius *L/A*

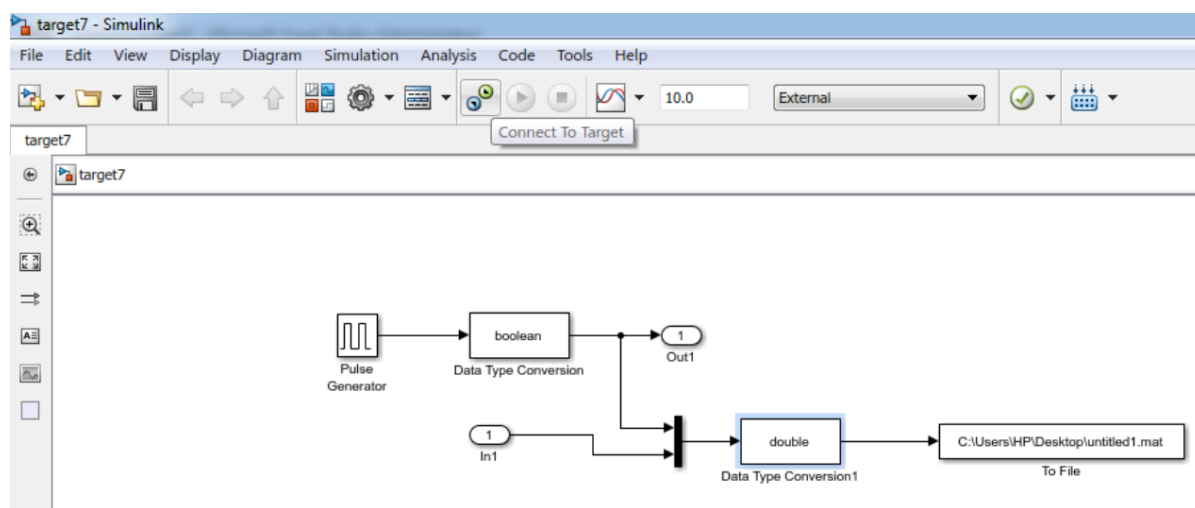


IN pradeda greitai mirksėti, o indikatorius *RUN* šviečia žalia spalva, tai reiškia, kad klaidų nėra ir tarp *EtherCAT* prietaiso ir *TwinCAT 3* programos vyksta komunikacija.



25 pav. Simuliavimo programos dialogo langas

Sekantis žingsnis – duomenų apsikeitimo tarp *MATLAB* programos paprogramės *Simulink* ir *Beckhoff TwinCAT 3* programos įjungimas. Tai atliekama pagrindiniame programos *MATLAB* paprogramės *Simulink* dialogo lange pasirinkus išorinį simuliacijos tipą (angl. *External*) ir paspaudus „prisijungti prie simuliacijos tikslo“ (angl. *Connect to Target*). Toks paprogramės *Simulink* dialogo langas pateiktas 26 paveiksle.

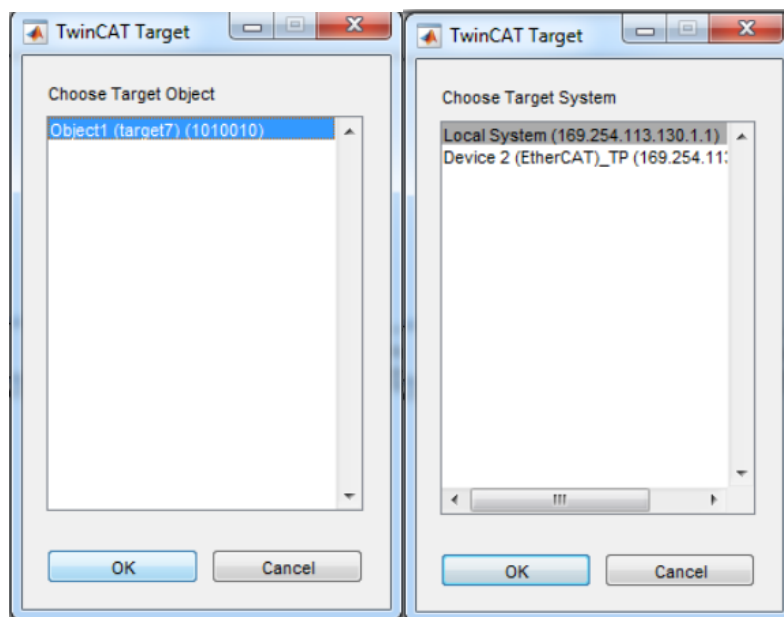


26 pav. *MATLAB* programos *Simulink* paprogramės dialogo langas

Paspaudus mygtuką pavadinimu „prisijungti prie tikslo“, pateikiamas *Simulink* paprogramės dialogo langas su tikslo objekto pasirinkimu. Iš pateikto sąrašo pasirenkamas objektas, kurio pavadinimas sutampa su *TwinCAT 3* programos skiltyje *TcCOM* objektai sukurtu naujo *TcCOM* objekto pavadinimu. Šio programos dialogo lango ekrano nuotrauka pateikta 27 paveikslo kairėje pusėje.

Pasirinkus objektą ir paspaudus mygtuką „gerai“, pateikimas dar vienas paprogramės dialogo langas su tikslo sistemos pasirinkimu (angl. *Target System*). Šio paprogramės dialogo lango ekrano

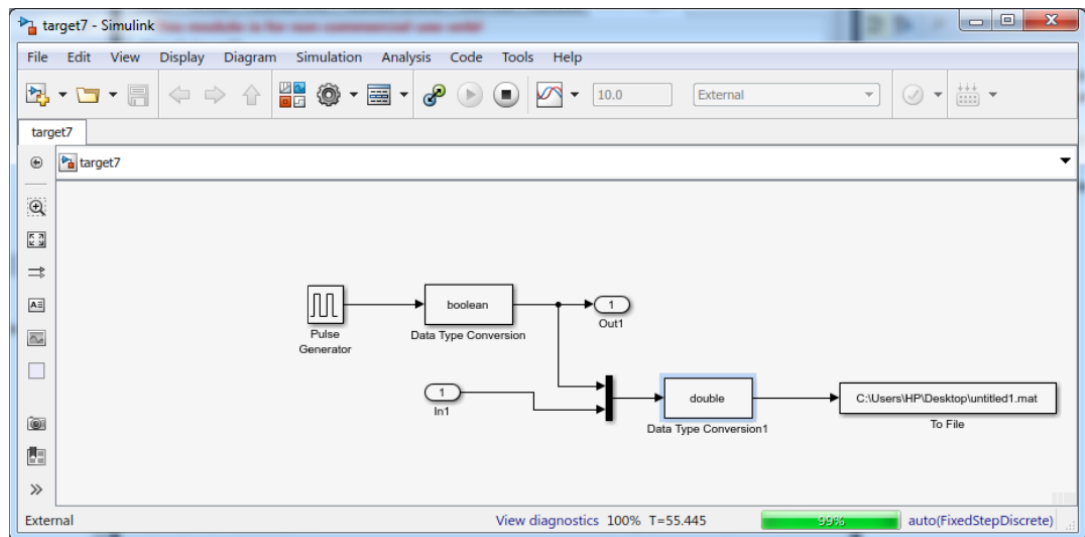
nuotrauka pateikta 27 paveikslo dešinėje pusėje. Iš pateikto sąrašo pasirenkama vietinė sistema (angl. *Local System*), kadangi šiuo atveju *TwinCAT 3* programa simuliaciją vykdo tame pačiame kompiuteryje jo procesoriaus resursus padalinant į dvi dalis – viena paliekama *Windows* operacinei sistemai ir visoms kitoms kompiuteryje naudojamoms programoms, įskaitant ir *MATLAB Simulink*, kita procesoriaus resursų dalis skiriama *TwinCAT 3* programai. Toks metodas tinkamas nesudėtingoms realaus laiko simuliacijoms. Vykdamas sudėtingas simuliacijas, naudojamas galingas simuliacijoms skirtas kompiuteris (angl. *Target Computer*) su dvejomis vietinio tinklo jungtimis. Viena naudojama komunikacijai su *EtherCAT* prietaisais, kita komunikacijai su kompiuteriu, kuriame stebimi simuliacijos duomenys ir keičiami parametrai (angl. *Modeling Computer*).



27 pav. *Simulink* programos prijungimo prie simuliacijos programos dialogo langai

Pasirinkus vietinę simuliacijos sistemą ir paspaudus mygtuką „gerai“, simuliacija pradeda ir pagrindinio *Simulink* paprogramės dialogo lango apačioje pradamas simuliacijos laiko skaičiavimas. Tuo pat metu *EtherCAT* prietaiso diskretinių išėjimo modulių kanalų indikacija pradeda mirksėti *MATLAB* programos *Simulink* paprogramėje naudojamo signalo generavimo bloko generuojamo signalo dažniu, tai reiškia, kad tarp *Simulink* paprogramės, *TwinCAT 3* programos ir *EtherCAT* prietaisų vyksta komunikacija. Vykstančios komunikacijos indikacijos *Simulink* paprogramėje ekrano nuotrauka pateikta 28 paveiksle.

Norint peržiūrėti simuliacijos rezultatus, reikia sustabdyti simuliaciją *Simulink* paprogramėje ir *TwinCAT 3* programoje. Tai atliekama *Simulink* paprogramėje paspaudus mygtuką „stop“ ir atsijungiant nuo simuliacijos tikslo paspaudus mygtuką pavadinimu „atsijungti nuo tikslo“. Atsijungus nuo simuliacijos tikslo *Simulink* paprogramėje, *TwinCAT 3* programoje paspaudžiamas mygtukas „stop“, o po to paspaudžiamas mygtukas „atsijungti“. Kai kuriais atvejais neuždarius abiejų programų ir norint peržiūrėti išsaugotus duomenis, *MATLAB* programa bandant atidaryti failą indikuodavo klaidą, pranešančią apie sugadintą failą.



**28 pav.** Vykstančios simuliacijos indikacija *Simulink* paprogramėje

Norint išvengti tokio atvejo, tiek *MATLAB*, tiek *Simulink* paprogramės po simuliacijos yra uždaromos.

## 4. Tiriamoji dalis

Šio darbo tiriamoji dalis sudaryta iš dviejų dalių. Pirmoje dalyje tiriami realaus laiko valdymo tinklų *EtherCAT* ir *PROFINET* minimalūs duomenų apsikeitimo ciklų laikai. Šiam tyrimui atlikti naudojama *MATLAB* programa, *EtherCAT* ir *PROFINET* prietaisų gamintojų pateikiami duomenys apie šiuos prietaisus, taip pat analitinėje dalyje išnagrinėtų mokslinių šaltinių duomenys, tokie kaip *EtherCAT* ir *PROFINET* tinklo jungimo būdai, kiekvieno iš šių jungimo būdų parametrai, formulės, skirtos apskaičiuoti kiekvieno iš galimų tinklo jungimo būdų minimalų duomenų apsikeitimo laiką. Šioje dalyje tirama, kaip kinta minimalus duomenų apsikeitimo laikas tarp valdančiojo ir pavaldžių prietaisų, atsižvelgiant į prietaisų kiekį tinkle, tinklo jungimo būdą ir prietaisams sujungti naudojamo kabelio vidutinį ilgį tarp prietaisų.

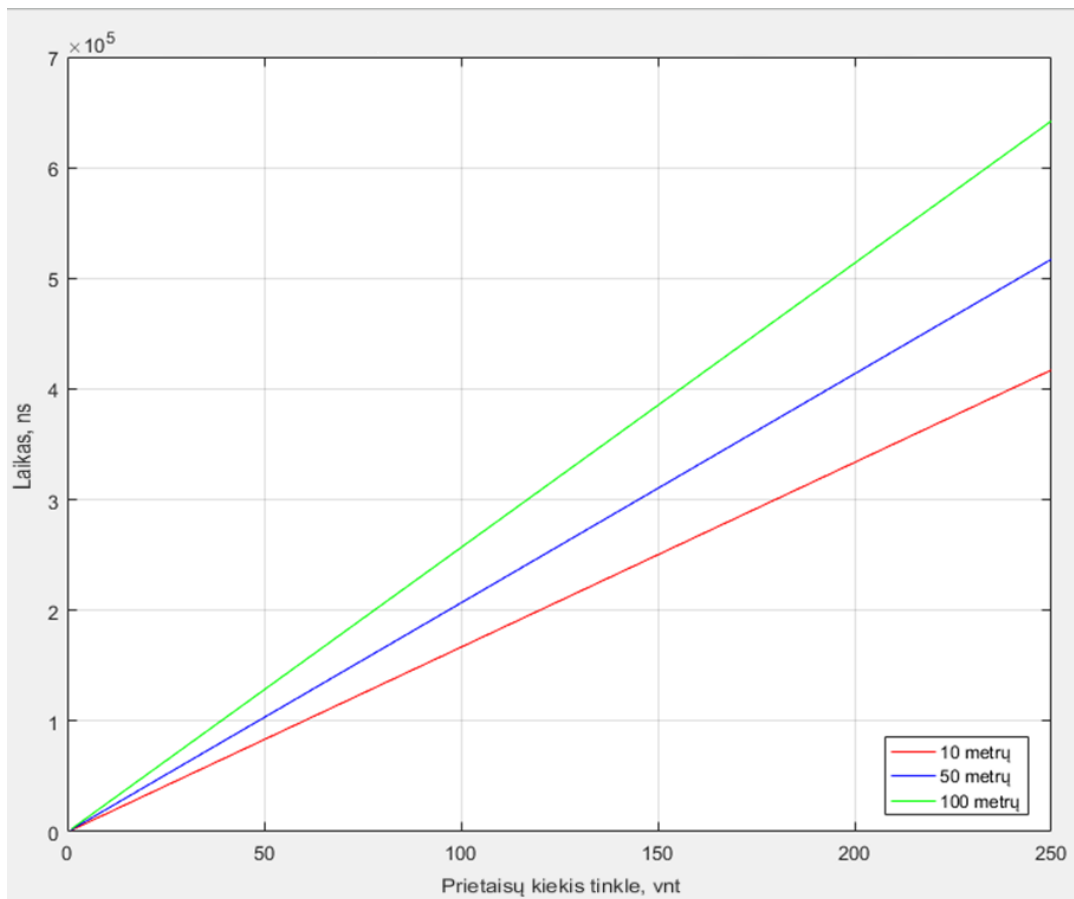
Antroje dalyje sudaroma metodika skirta tirti, kiek laiko trunka *EtherCAT* komandos įvykdymas nuo komandos išsiuntimo iki jos įvykdymo. *MATLAB Simulink* paprogramėje sudaromas modelis, kuriuo galima valdyti *EtherCAT* prietaisus Windows kompiuteriu, ir tirama, kaip keičiasi vėlinimo laikas atsižvelgiant į užduodamo išėjimo signalo dažnį naudojant diskretinius įėjimų ir išėjimų modulius. Kokią įtaką tinklo greitaveikai turi skirtingas modulių kiekis prijungtas prie *EtherCAT* prietaiso. Šiam tyrimui atlikti naudojamas nešiojamasis kompiuteris su *Windows 7* operacine sistema ir įdiegta *MATLAB 2016b* bei *TwinCAT 3* programine įranga. Šiam tyrimui naudojamas OMRON gamintojo *EtherCAT* prietaisas *NX-ECC 203 EtherCAT Coupler*. Tai dviejų *EtherCAT* jungčių prietaisas, skirtas nutolusiems įėjimo ir išėjimo signalų moduliams prijungti, tokie nutolę moduliai dar kitaip vadinami nutolusiomis valdiklio salomis. Prie šio prietaiso galima prijungti iki 63 įėjimų / išėjimų signalų modulių įskaitant papildomus ir atskiriamuosius maitinimo šaltinius. Minimalus duomenų apsikeitimo tarp valdančiojo ir pavaldžių prietaisų laikas, kurį palaiko šis prietaisas yra 125 mikrosekundės.

### 4.1. *EtherCAT* ir *PROFINET* realaus laiko valdymo tinklų duomenų apsikeitimo ciklų laikų tyrimas

Šioje dalyje tirama vidutinio tinklo kabelio ilgio tarp prietaisų įtaka realaus laiko valdymo tinklų minimaliems vėlinimo laikams. Taip pat šioje dalyje atliekami tokie tyrimai: *EtherCAT* minimalaus duomenų apsikeitimo laiko, taikant prietaisų adresavimą priklausomybė nuo tinklo topologijos; *EtherCAT* duomenų paketo išsiuntimo iš valdančiojo prietaiso trukmė, naudojant prietaisų arba loginį adresavimą; minimalaus duomenų apsikeitimo laiko *EtherCAT* tinkle palyginimas, naudojant prietaisų arba loginį adresavimus tyrimas; minimalaus duomenų apsikeitimo laiko *PROFINET* tinkle priklausomybės nuo tinklo topologijos tyrimas; minimalaus duomenų apsikeitimo laiko nuosekliai sujungtame *PROFINET* tinkle priklausomybės nuo vidutinio tinklo kabelio tarp prietaisų ilgio tyrimas; minimalaus duomenų apsikeitimo laiko *EtherCAT* ir *PROFINET* tinkluose palyginimas.

#### 4.1.1. Tinklo kabelių ilgio įtakos komunikacinio signalo vėlinimo laikams tyrimas

Naudojantis literatūros analizės dalyje pateiktomis formulėmis, *EtherCAT* prietaisus gaminančios kompanijos *Beckhoff automation* pateiktais *EtherCAT* prietaisų laikais, kurie pateikti 3 lentelėje ir moksliniuose straipsniuose rasta informacija, kuri pateikta poskyryje „Tinklo jungimo būdai“, sudaryta *MATLAB* programa, skirta apskaičiuoti minimalų duomenų apsikeitimo laiką, priklausomai nuo *EtherCAT* prietaisų tinkle kiekio bei laido tarp šių prietaisų ilgio. Minimalaus duomenų apsikeitimo laiko nuosekliai sujungtame *EtherCAT* tinkle grafikas pateiktas 29 paveiksle.



29 pav. Minimalaus duomenų apsikeitimo laiko nuosekliai sujungtame *EtherCAT* tinkle grafikas

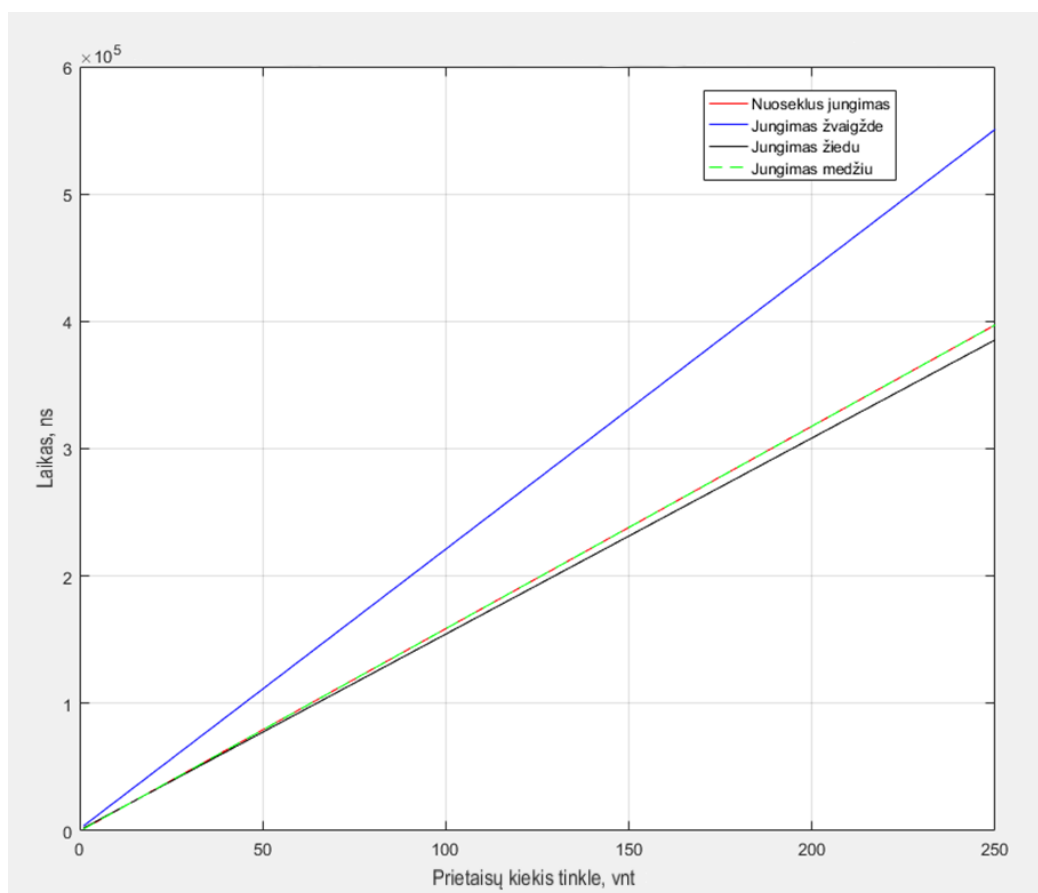
Atlikus skaičiavimus, nustatyta, kad tinkle, kuriame sujungta 250 pavaldžių *EtherCAT* prietaisų, minimalus vėlinimo laikas, kai vidutinis laido ilgis tarp prietaisų, yra apytiksliai lygus 0,41 milisekundės. Kai vidutinis kabelio ilgis tarp prietaisų yra 50 metrų, minimalus vėlinimo laikas siekia 0,515 milisekundės, o kai laiko ilgis siekia maksimalų leidžiamą laido ilgį tarp *EtherCAT* prietaisų, minimalus vėlinimo laikas yra 0,642 milisekundės.

#### 4.1.2. Minimalaus duomenų apsikeitimo laiko, taikant prietaisų adresavimą priklausomybė nuo tinklo topologijos tyrimas

Šiame skyrelyje aptariamas tyrimas, kaip keičiasi minimalus duomenų apsikeitimo laikas *EtherCAT* tinkle, šį tinklą jungiant skirtingomis tinklo topologijomis. Skaičiavimuose imama, kad *EtherCAT* protokolą naudoja prietaisų adresavimą. Taip pat imama, kad vidutinis tinklo kabelio ilgis tarp prietaisų lygus 10 metrų. Toks kabelio ilgis sukuria apytiksliai 50 nanosekundžių vėlinimą. Duomenų paketo dydžio skaičiavimams imama, kad vidutinis duomenų kiekis skirtas kiekvienam prietaisui lygus 50 baitų, atsižvelgiant į *Ethernet* paketo sandarą, skaičiavimams imama, kad *Ethernet* antraštė (angl. *Ethernet Header*) yra 14 baitų, *EtherCAT* antraštė sudaro 2 baitus, o telegramos antraštė 10 baitų. Taip pat skaičiavimams imama ir skaitiklio (angl. *Working Counter*) vertė, kuri yra 2 baitai. Tyrimo rezultatai pavaizduoti grafiškai 30 paveiksle.

Atlikus minimalaus duomenų apsikeitimo laiko *EtherCAT* tinkle, šį tinklą jungiant skirtingomis topologijomis, tyrimą, nustatyta, kad tinklą jungiant nuosekliai ir medžio tipo jungimu, gaunamas vienodas vėlinimo laikas. Vėlinimo laikas jungiant, bet kuriuo iš šių tinklo jungimo būdų, tinkle esant

100 pavaldžių prietaisų yra 0,158 milisekundės, o kai prietaisų tinkle yra 250, vėlinimo laikas padidėja iki 0,397 milisekundės.



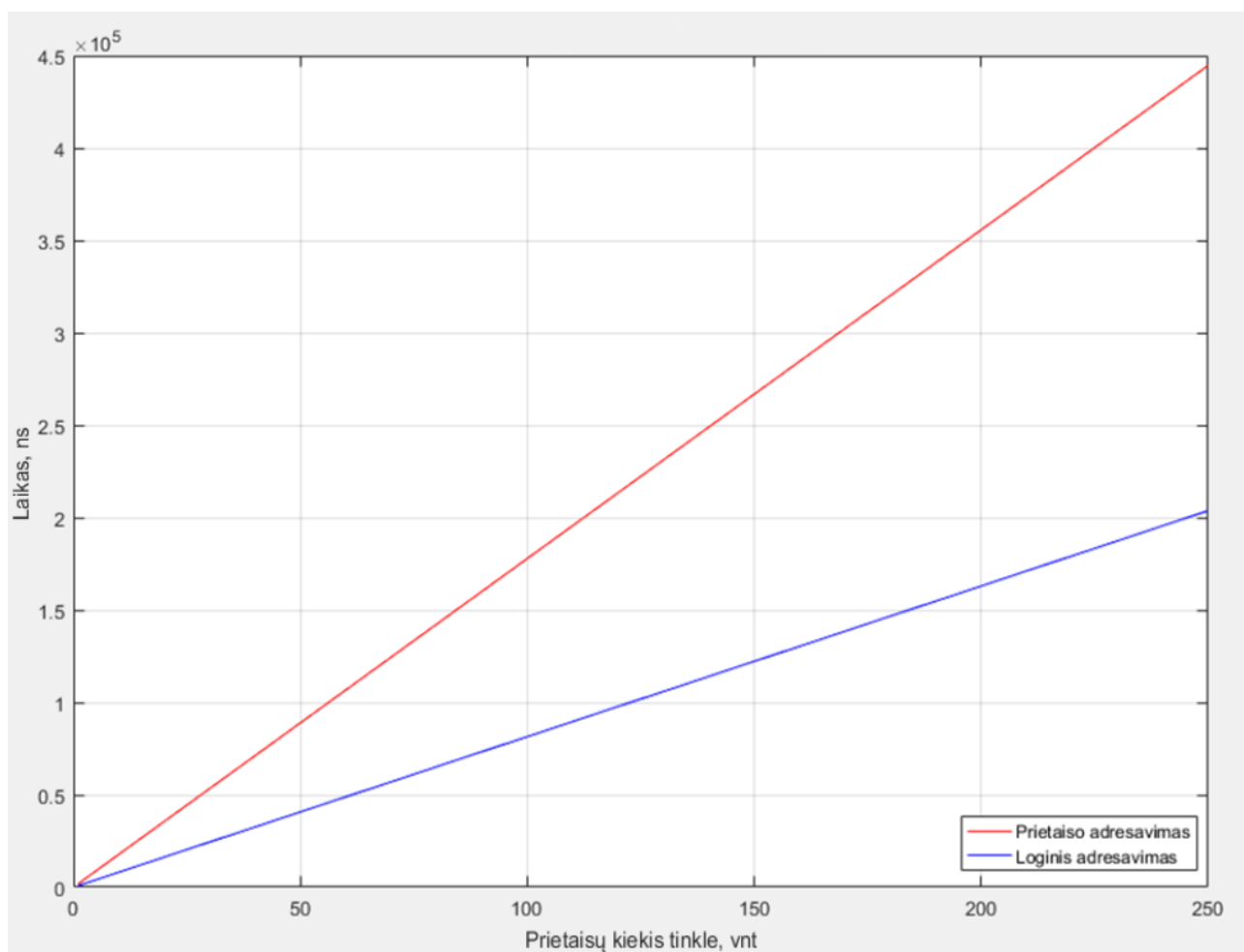
**30 pav.** Minimalaus duomenų apsikeitimo laiko *EtherCAT* tinkle naudojant prietaisų adresavimą grafikas

Didžiausiu vėlinimo laiku pasižymi žvaigždės tinklo topologija. Tinklą jungiant žvaigžde ir tinkle esant 100 pavaldžių prietaisų vėlinimo laikas yra 0,221 milisekundės. Padidinus prietaisų skaičių iki 250, gaunamas 0,5508 milisekundės vėlinimas. Pats mažiausias vėlinimo laikas, iš visų galimų tinklo jungimo būdų, gaunamas tinklą jungiant žiedo topologija. Tinkle šiuo būdu sujungus 100 pavaldžių *EtherCAT* prietaisų, gaunamas 0,154 milisekundės vėlinimo laikas, prietaisų skaičių padidinus iki 250, vėlinimo laikas padidėja iki 0,385 milisekundės.

#### **4.1.3. Duomenų paketo išsiuntimo iš valdančiojo prietaiso trukmė, naudojant prietaisų arba loginį adresavimą tyrimas**

Atsižvelgiant į *EtherCAT* duomenų paketo dydį, kuris priklauso nuo valdančiojo prietaiso aptarnaujamų pavaldžių prietaisų kiekio, naudojami skirtingi adresavimo tipai. Kol bendras *EtherCAT* duomenų paketo dydis neviršija 1500 baitų, tol naudojamas prietaisų adresavimas. Šis adresavimas reiškia, kad kiekvienam prietaisui priskiriama atskira telegrama, kuri turi jai priskirtą telegramos antraštę, kuri yra 10 baitų dydžio. Esant dideliame telegramų kiekiui, telegramų antraštės sudaro nemažą bendro *EtherCAT* duomenų paketo dalį. Kai prietaisų daug, o bendras kiekvienoje telegramoje esančių duomenų ir telegramų antraštės sudarančių duomenų kiekis viršija 1500 baitų, pradedamas taikyti loginis adresavimas. Prietaisus adresuojant šiuo būdu, siunčiama tik viena telegrama, kurioje yra visiems prietaisams skirti duomenys. Taikant tokį adresavimą atsisakoma duomenų kiekio, kuris lygus telegramos antraštės dydžiui baitais ir telegramų skaičiaus sandaugai.

Atlikto duomenų paketo išsiuntimo laiko, naudojant prietaisų ir loginį adresavimus, tyrimo rezultatai grafiškai pateikti 31 paveiksle.

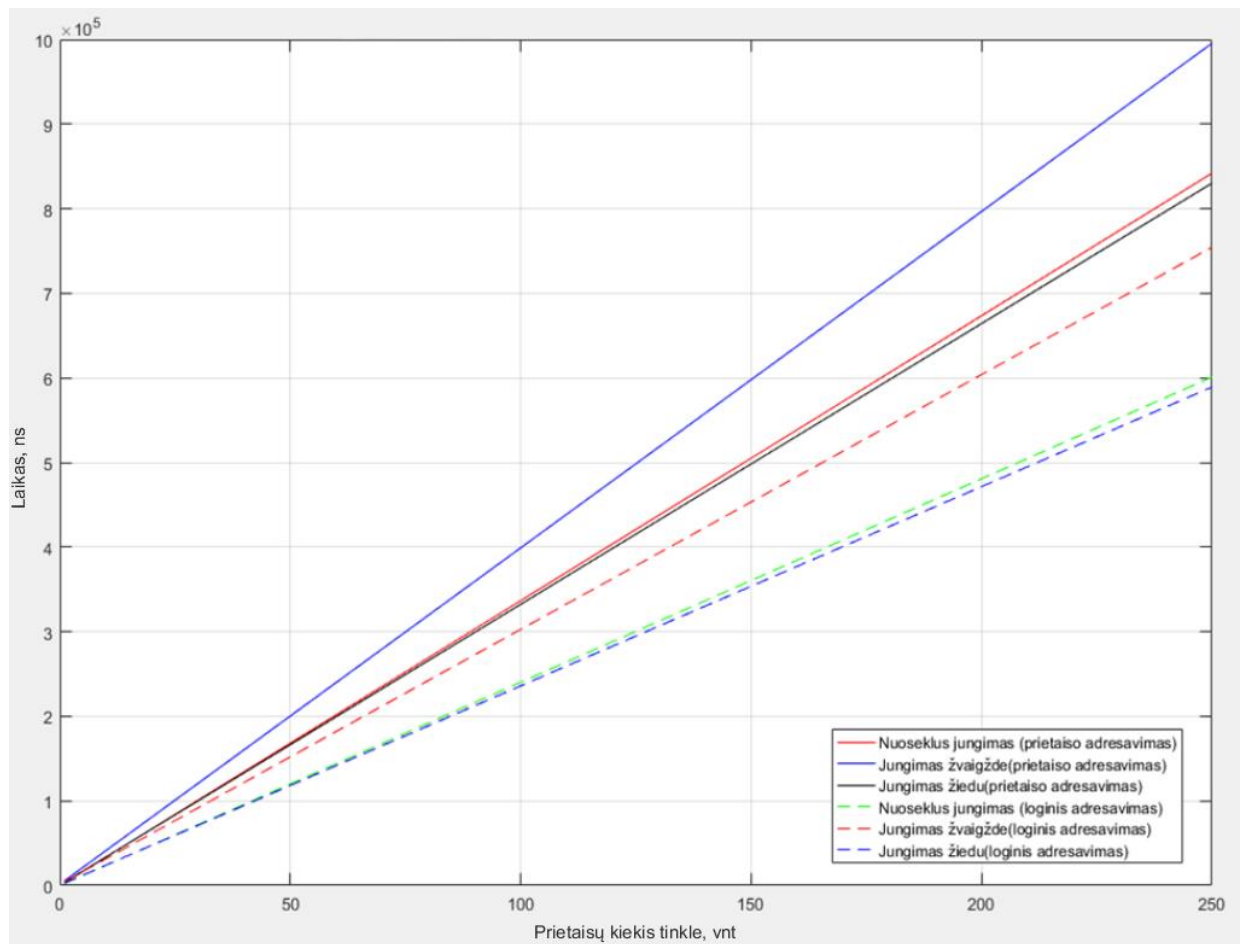


**31 pav.** Duomenų paketo išsiuntimo laiko palyginimas, naudojant loginį arba prietaisų adresavimą

Atlikus tyrimą nustatyta, kad duomenų paketo išsiuntimo laikas, naudojant prietaisų adresavimą yra didesnis už vėlinimo laiką, kai naudojamas loginis adresavimas. Vėlinimo laikų skirtumai aiškiai matomi tada, kai duomenų kiekis vienam prietaisui mažas. Šiuo atveju duomenų kiekis vienam prietaisui siekia 50 baitų. Duomenų kiekį didinant, loginio adresavimo vėlinimo vertė artėja prie vėlinimo vertės, kai naudojamas prietaisų adresavimas. Duomenų paketo išsiuntimo vėlinimo vertei tinklo jungimo topologija įtakos neturi. Tinkle esant 100 prietaisų ir naudojant loginį adresavimą, paketo išsiuntimo vėlinimo vertė yra lygi 0,0814 milisekundės, kai naudojamas prietaisų adresavimas ši vertė lygi 0,177 milisekundės. Prietaisų kiekį tinkle padidinus iki maksimalaus grafike pateikto 250, vėlinimo vertė naudojant loginį adresavimą yra 0,203 milisekundės, adresavimą pakeitus į prietaisų, esant tam pačiam prietaisų skaičiui, vėlinimo vertė yra lygi 0,444 milisekundės.

#### **4.1.4. Minimalaus duomenų apsikeitimo laiko *EtherCAT* tinkle palyginimas, naudojant prietaisų arba loginį adresavimus tyrimas**

Atlikus duomenų paketo išsiuntimo laiko skaičiavimus prietaisų ir loginio adresavimo būdais, šie laikai pridėti prie tinklo vėlinimo laiko, priklausomai nuo tinklo topologijos ir gauti rezultatai pateikti grafiškai 32 paveiksle. Grafikas parodo duomenų apsikeitimo ciklo laiko skirtumus kiekvienam tinklo jungimo būdai ir adresavimo tipui.



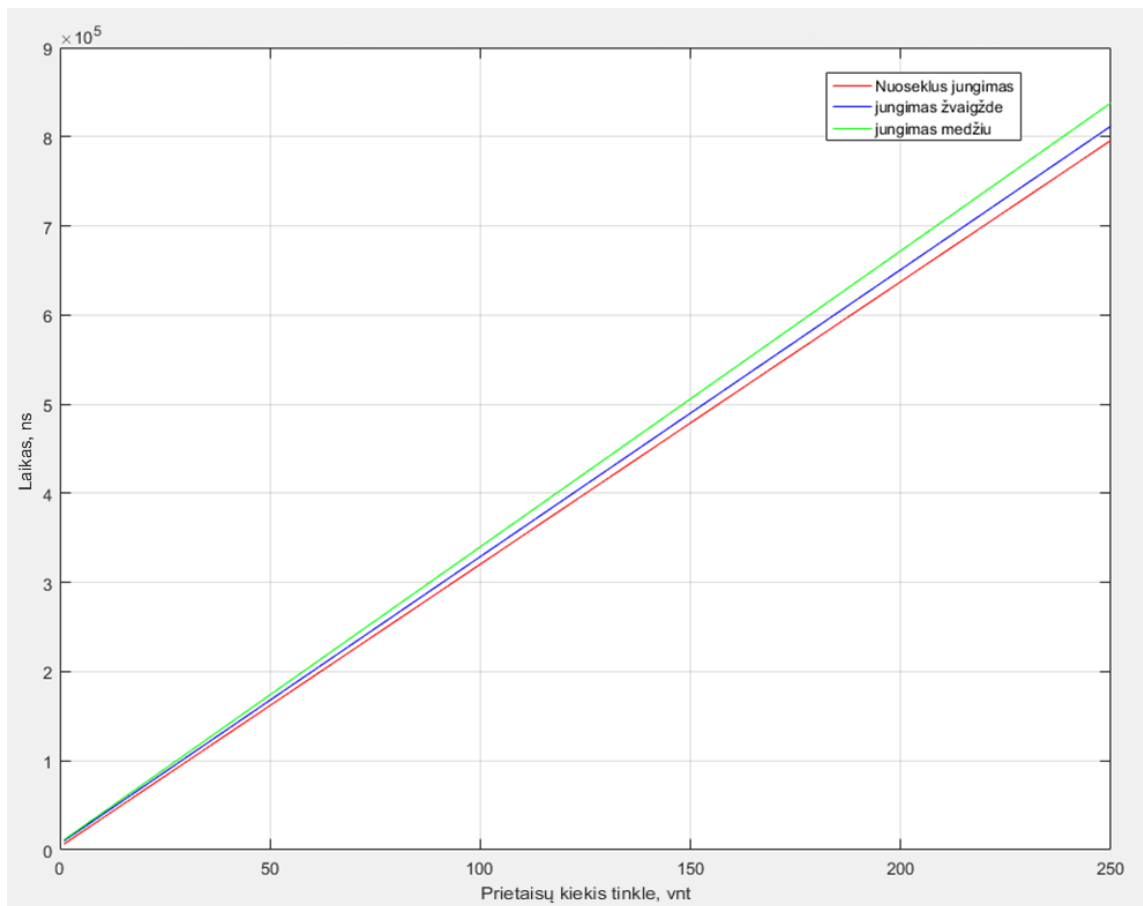
**32 pav.** Minimalaus duomenų apsikeitimo laiko *EtherCAT* tinkle naudojant prietaisų arba loginį adresavimą palyginamasis grafikas

Ištyrus kaip skiriasi minimalūs duomenų apsikeitimo *EtherCAT* tinkle laikai, priklausomai nuo to, koks pavaldžių prietaisų adresavimo tipas taikomas, nustatyta, kad naudojant loginį adresavimą vėlinimo laikai gaunami mažesni, nei naudojant prietaisų adresavimą. Vėlinimo skirtumas, naudojant tokią pat jungimo topologiją ir tinkle sujungus 100 pavaldžių *EtherCAT* prietaisų, sudaro 0,096 milisekundės. Lyginant tokias pat topologijas, bet naudojant skirtingą adresavimo tipą, kai tinkle yra 250 pavaldžių *EtherCAT* prietaisų, vėlinimo laiko skirtumas sudaro 0,2409 milisekundės.

#### 4.1.5. Minimalaus duomenų apsikeitimo laiko *PROFINET* tinkle priklausomybės nuo tinklo topologijos tyrimas

Naudojantis literatūros analizės dalyje pateiktomis formulėmis, moksliniuose šaltiniuose pateiktomis skaičiavimams reikalingomis kintamųjų vertėmis ir oficialaus gamintojo pateiktomis duomenimis, atliktas tyrimas, kaip keičiasi minimalus vėlinimo laikas *PROFINET* tinkle, tinklą jungiant skirtingomis topologijomis. Skaičiavimuose imta, kad vidutinis tinklo kabelio ilgis tarp pavaldžių prietaisų siekia 10 metrų. Tokio ilgio kabelyje susidarantis vėlinimas pagal moksliniuose straipsniuose pateiktus duomenis sudaro 50 nanosekundžių [28]. Duomenų paketo iš pavaldaus prietaiso išsiuntimo užlaikymo vertė imama 3000 nanosekundžių. Maksimali *PHY* užlaikymo vertė imama 500 nanosekundžių (250 nanosekundžių išsiuntimo vėlinimas, 250 nanosekundžių priėmimo vėlinimas) [29]. Paketo išsiuntimo vertė, pagal gamintojo deklaruojamą vidutinę vertę imama 3120 nanosekundžių [30]. Tyrimo naudojamų skaičiavimų rezultatai grafiškai pavaizduoti 33 paveiksle.





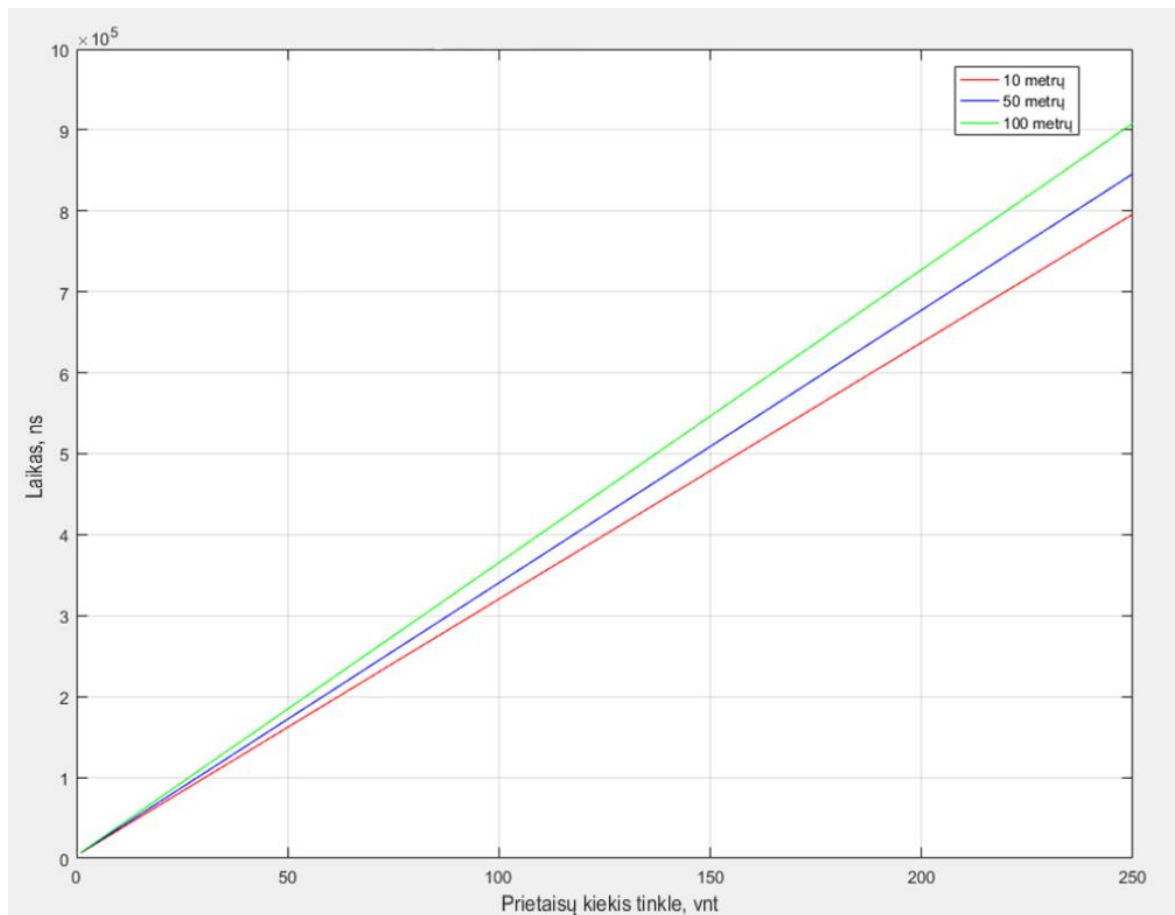
**33 pav.** Minimalaus duomenų apsikeitimo laiko priklausomybė nuo prietaisų skaičiaus ir tinklo topologijos

*PROFINET* tinklą jungiant skirtingomis topologijomis, didžiausiu vėlinimo laiku pasižymi jungimas medžio topologija. Tinklą jungiant tokiu būdu, kai tinkle yra 100 pavaldžių prietaisų, vėlinimo laikas siekia 0,34 milisekundės. Pavaldžių prietaisų skaičiui padidėjus iki 250, vėlinimo laikas pakyla iki 0,838 milisekundės. Mažiausiu vėlinimo laiku esant bet kokiam prietaisų kiekiui pasižymi nuoseklus *PROFINET* tinklo jungimo būdas. Nuosekliai sujungus 100 prietaisų, vėlinimo laikas yra 0,32 milisekundės. Nuosekliai sujungtų prietaisų skaičių padidinus iki 250, gaunamas 0,796 milisekundės vėlinimo laikas. *PROFINET* tinklą jungiant žvaigžde, kai pavaldžių prietaisų skaičius lygus 100, vėlinimo laikas siekia 0,329 milisekundės. Tinkle sujungus 250 prietaisų, vėlinimo laikas padidėja iki 0,812 milisekundės.

#### **4.1.6. Minimalaus duomenų apsikeitimo laiko nuosekliai sujungtame *PROFINET* tinkle priklausomybės nuo vidutinio tinklo kabelio tarp prietaisų ilgio tyrimas**

Atlikus minimalaus duomenų apsikeitimo laiko *PROFINET* tinkle priklausomybės nuo topologijos tyrimą, šiame skyrelyje aptariamas tyrimas, kokią įtaką vėlinimo laikui turi vidutinis kabelio ilgis tarp prietaisų. Tyrimui pasirinktas nuoseklus tinklo jungimas. Prietaisų skaičius keičiamas nuo 1 iki 250. Laido ilgis paimtas trimis intervalais: 10, 50 ir 100 metrų. Rezultatai pateikti grafiškai 34 paveiksle.

Atlikus gautų duomenų analizę, nustatyta, kad minimalus duomenų apsikeitimo laikas nuosekliai sujungtame *PROFINET* tinkle, kai tinkle yra 100 pavaldžių prietaisų, o vidutinis laiko ilgis tarp prietaisų yra 10 metrų, 0,32 milisekundės.



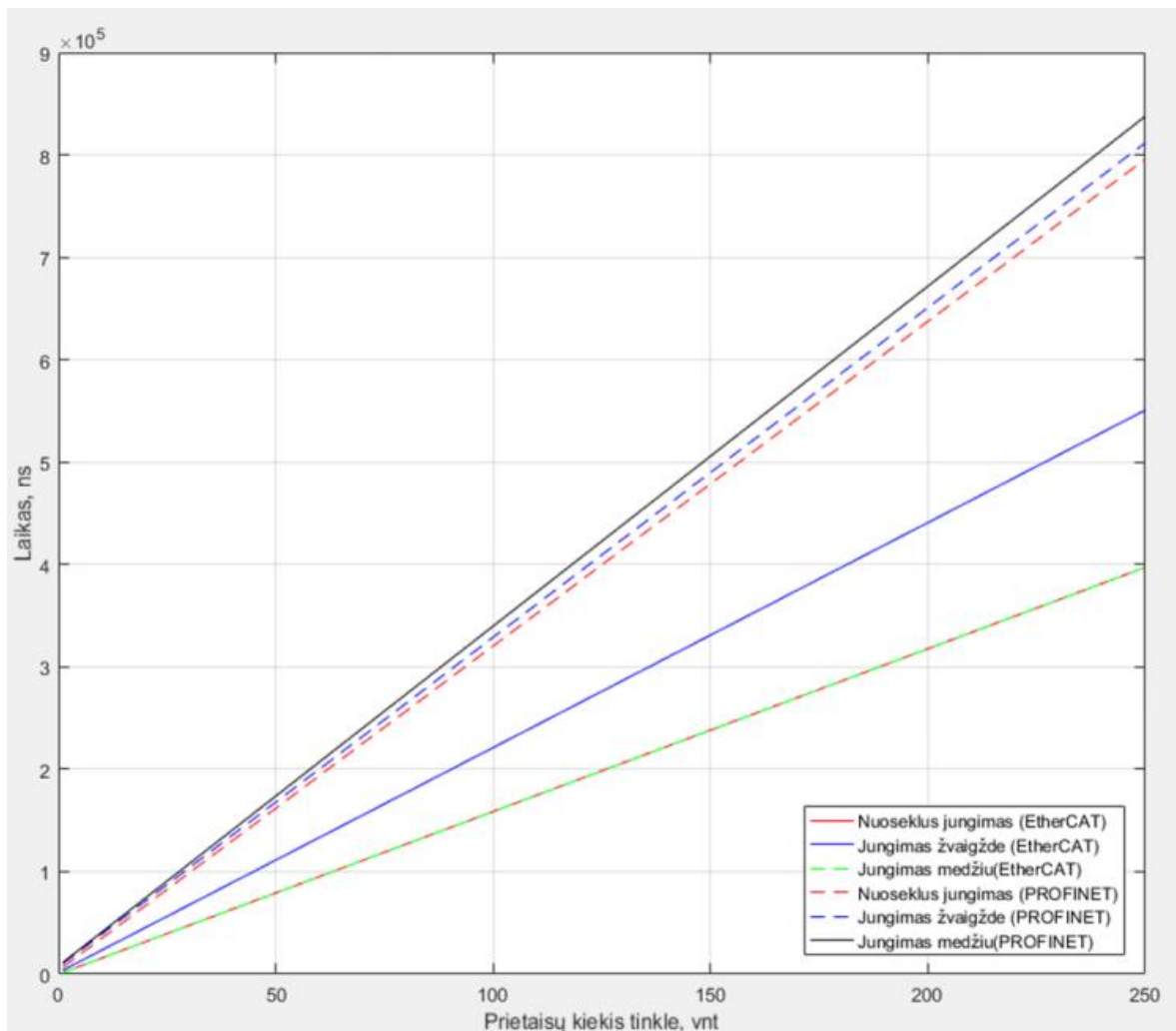
**34 pav.** Minimalaus duomenų apsikeitimo laiko nuosekliai sujungtame *PROFINET* tinkle priklausomybė nuo tinklo kabelio ilgio tarp prietaisų ir prietaisų skaičiaus

Vidutinį kabelio ilgį padidinus iki 50 metrų, vėlinimo laikas tampa 0,34 milisekundės, kabelio ilgį pailginus iki vidutinio 100 metrų ilgio, vėlinimo laikas tampa lygus 365500 milisekundės. Imant tuos pačius ilgius, kai tinkle pavaldžių prietaisų skaičius yra 250, gaunamas vėlinimo laikas atitinkamai: 0,796 milisekundės, 0,846 milisekundės, ir 0,908 milisekundės.

#### 4.1.7. Minimalaus duomenų apsikeitimo laiko *EtherCAT* ir *PROFINET* tinkluose palyginimas

Atlikus *EtherCAT* ir *PROFINET* realaus laiko valdymo tinklų duomenų perdavimo vėlinimų tyrimą, gauti abiejų šių tinklų vėlinimo laikai pateikti grafiškai 35 paveiksle. Grafike lyginamos nuoseklaus jungimo, jungimo žvaigžde ir jungimo medžiu topologijos. Visose topologijose skaičiavimams buvo naudojamas 10 metrų vidutinis tinklo kabelio ilgis tarp prietaisų.

Iš grafike pateiktų rezultatų matyti, kad *EtherCAT* valdymo tinklo vėlinimo laikai, esant bet kuriai iš šiam tinklui leidžiamų topologijų, yra mažesni nei *PROFINET* valdymo tinklo, tinklą jungiant analogiškais tinklo topologijomis. Lyginant tris grafike pateiktas *EtherCAT* tinklo jungimo topologijas, didžiausiu vėlinimo laiku pasižymi tinklo jungimas žvaigžde. Šis laikas tinkle esant 100 pavaldžių prietaisų sudaro 0,221 milisekundės, o kai prietaisų skaičius padidinamas iki 250, laikas siekia 0,55 milisekundės. Mažiausiu vėlinimo laiku iš visų *EtherCAT* topologijų pasižymi jungimas žiedu, bet kadangi *PROFINET* nepalaiko žiedo topologijos, o *EtherCAT* prietaisuose toks jungimas sutinkamas labai retai dėl to, kad valdantieji prietaisai dažniausiai turi tik vieną *EtherCAT* jungtį, todėl šiame grafike tokio jungimo būdo vėlinimo laikas nepateikiamas.



**35 pav.** Minimalaus duomenų apsikeitimo laiko *EtherCAT* ir *PROFINET* tinkluose palyginimas

Iš 35 paveiksle pateikto grafiko matyti, kad mažiausiu vėlinimo laiku pasižymi jungimas nuosekliu bei jungimas medžio būdu. Šių abiejų jungimo tipų vėlinimo laikai yra visiškai vienodi. Tinkle, kuriame yra 100 pavaldžių prietaisų, tinklą jungiant bet kuriuo iš šių dviejų būdų, vėlinimo laikas siekia 0,158 milisekundės, o kai prietaisų kiekis padidinamas iki 250 vienetų, vėlinimas tampa lygus 0,397 milisekundės.

*PROFINET* atveju didžiausiu vėlinimo laiku pasižymi jungimas medžio topologija. Tinklą jungiant tokiu būdu, kai tinkle yra 100 pavaldžių prietaisų, vėlinimo laikas siekia 0,339 milisekundės. Pavaldžių prietaisų skaičiui padidėjus iki 250, vėlinimo laikas pakyla iki 0,837 milisekundės. Mažiausiu vėlinimo laiku esant bet kokiam prietaisų kiekiui pasižymi nuoseklus *PROFINET* tinklo jungimo būdas. Nuosekliai sujungus 100 prietaisų, vėlinimo laikas yra 0,32 milisekundės, o prietaisų skaičių padidinus iki 250, gaunamas 0,795 milisekundės vėlinimo laikas. *PROFINET* tinklą jungiant žvaigžde, kai pavaldžių prietaisų skaičius lygus 100, vėlinimo laikas siekia 0,328 milisekundės. Tinkle sujungus 250 prietaisų, vėlinimo laikas padidėja iki 0,811 milisekundės.

Iš grafiko matyti, kad vėlinimo laikų *PROFINET* tinkle skirtumai tarp skirtingų tinklo jungimo topologijų yra mažesni, lyginant su *EtherCAT* tinklu, tai reiškia, kad tinklo jungimo būdas *PROFINET* tinkle daro mažesnę įtaką vėlinimo laikams lyginant šį tinklą su *EtherCAT*.

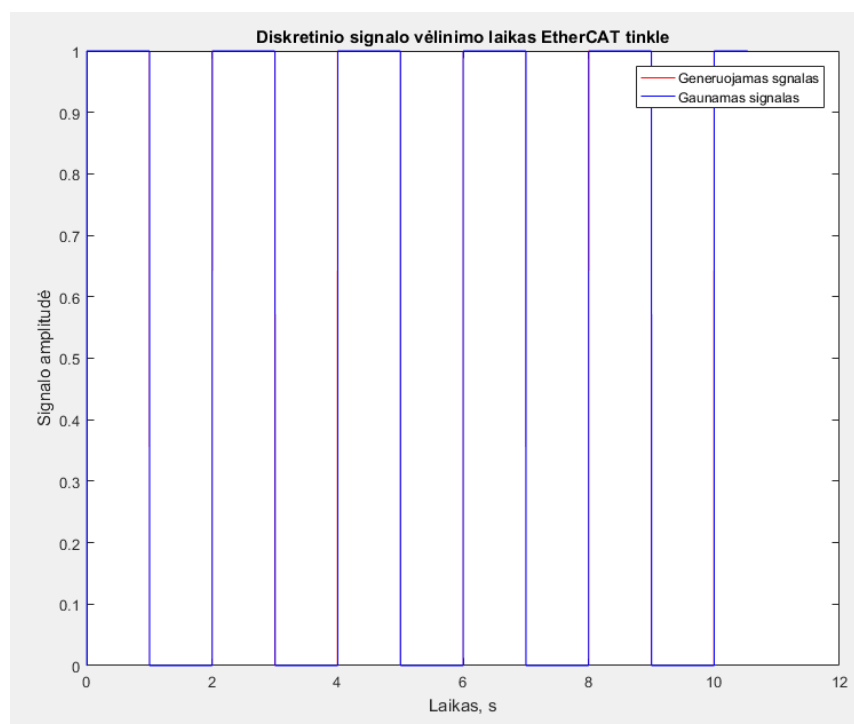
## 4.2. EtherCAT realaus laikos komunikacinio tinklo greitaveikos ir vėlinimo laikų tyrimas

Šioje darbo dalyje tiriama, kaip keičiasi vėlinimo laikas atsižvelgiant į užduodamo išėjimo signalo dažnį naudojant diskretinius įėjimų ir išėjimų modulius. Kokią įtaką tinklo greitaveikai turi skirtingas modulių kiekis prijungtas prie *EtherCAT* prietaiso. Šiam tyrimui atlikti naudojamas nešiojamasis kompiuteris su *Windows 7* operacine sistema ir įdiegta *MATLAB 2016b* bei *TwinCAT 3* programine įranga. Šiam tyrimui naudojamas *OMRON* gamintojo *EtherCAT* prietaisas *NX-ECC 203 EtherCAT Coupler*. Tai dviejų *EtherCAT* jungčių prietaisas, skirtas nutolusiems įėjimo ir išėjimo signalų moduliams prijungti, tokie nutolę moduliai dar kitaip vadinami nutolusiomis valdiklio salomis. Prie šio prietaiso galima prijungti iki 63 įėjimų / išėjimų signalų modulių įskaitant papildomus ir atskiriamuosius maitinimo šaltinius. Minimalus duomenų apsikeitimo tarp valdančiojo ir pavaldžių prietaisų laikas, kurį palaiko šis prietaisas yra 125 mikrosekundės.

### 4.2.1. Bandomojo *MATLAB Simulink* modelio sudarymas ir jo tyrimas

Išbadyti sudarytai metodikai, skirtai tirti *EtherCAT* realaus laikos komunikacinio tinklo greitaveikai ir vėlinimo laikams, taikant *MATLAB* programinę įrangą, sudaromas bandomasis *MATLAB Simulink* modelis, kurio ekrano nuotrauka pateikta 23 paveiksle. Vadovaujantis metodais, aprašytais poskyryje pavadinimu „Metodai, skirti *EtherCAT* realaus laiko komunikacinio tinklo greitaveikai ir vėlinimo laikams nustatyti *MATLAB* programine įranga“, prie kompiuterio prijungiamas pavaldus *EtherCAT* prietaisas ir programose *Bechhoff TwinCAT 3* ir *MATLAB Simulink* paleidžiama simuliacija. Praėjus 10 sekundžių nuo simuliacijos pradžios, simuliacija abiejose programose stabdoma, o programos išjungiamos, kad simuliacijos duomenys išsisaugotų tinkamai.

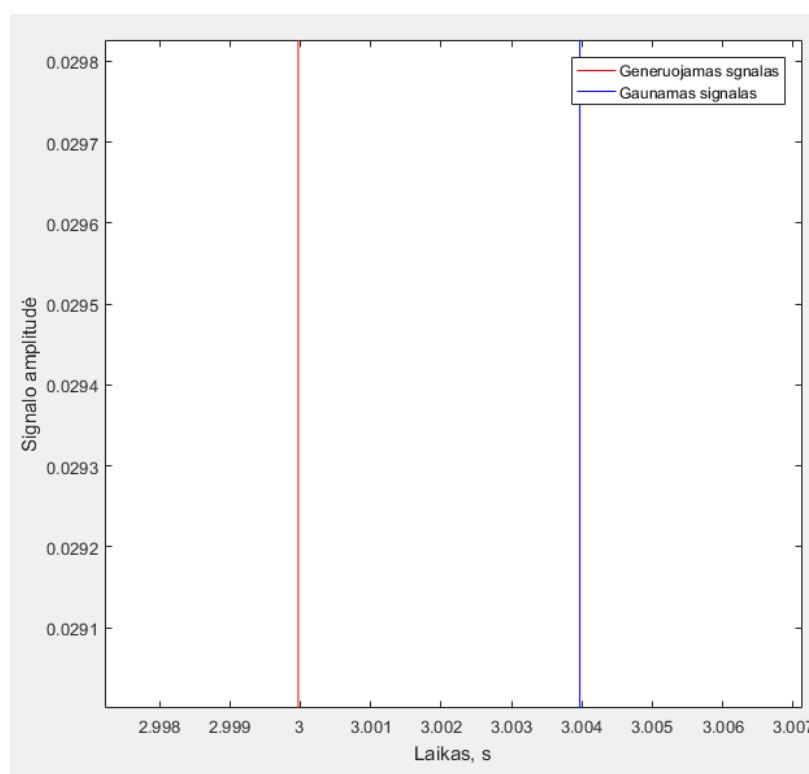
Duomenų peržiūrėjimui paleidžiama *MATLAB* programa ir sukūrus naują *MATLAB* programos langą, naudojantis *load* ir *plot* funkcijomis gauti simuliacijos duomenys atvaizduojami grafiškai. 36 paveiksle pateiktas simuliacijai naudojamo modelio rezultatų grafikas.



36 pav. Gautų rezultatų grafikas

Modelyje naudojamas impulsų generavimo blokas pavadinimu *Pulse generator*. Jo nustatymuose pasirinktas vienos sekundės dažnis. Minimali reikšmė yra 0, o maksimali lygi 1. Šio bloko išėjimo signalas konvertuojamas į *boolean* duomenų tipą naudojantis duomenų tipo keitimo bloku, kurio pavadinimas *data type conversion*. Iš duomenų keitimo bloko duomenys siunčiami į išėjimo bloką *Out1*. Šiam blokui *Beckhoff TwinCAT 3* programoje priskirtas fizinis išėjimas. Kai į šį bloką ateina aukštas signalo lygis, fizinis išėjimas taip pat generuoja aukštą signalo lygį. Naudojamas fizinis išėjimas trumpu laidu sujungtas su fiziniu įėjimu. Šiam įėjimui *TwinCAT 3* programoje priskirtas įėjimo blokas *In1*. Kai į fizinį įėjimą prijungiamas aukštas signalo lygis, blokas *In1* taip pat generuoja aukštą signalo lygį.

36 paveiksle grafike raudona spalva pažymėta kreivė žymi signalą, kuris siunčiamas į išėjimo bloką *Out1*. Mėlyna spalva pažymėtas kreivė vaizduoja signalą gaunamą į įėjimo bloką *In1*. Kadangi reikšmės yra labai artimos, dėl to 37 ir 38 paveiksluose pateikti grafikai su priartintomis ir aiškiau matomomis laiko vertėmis.

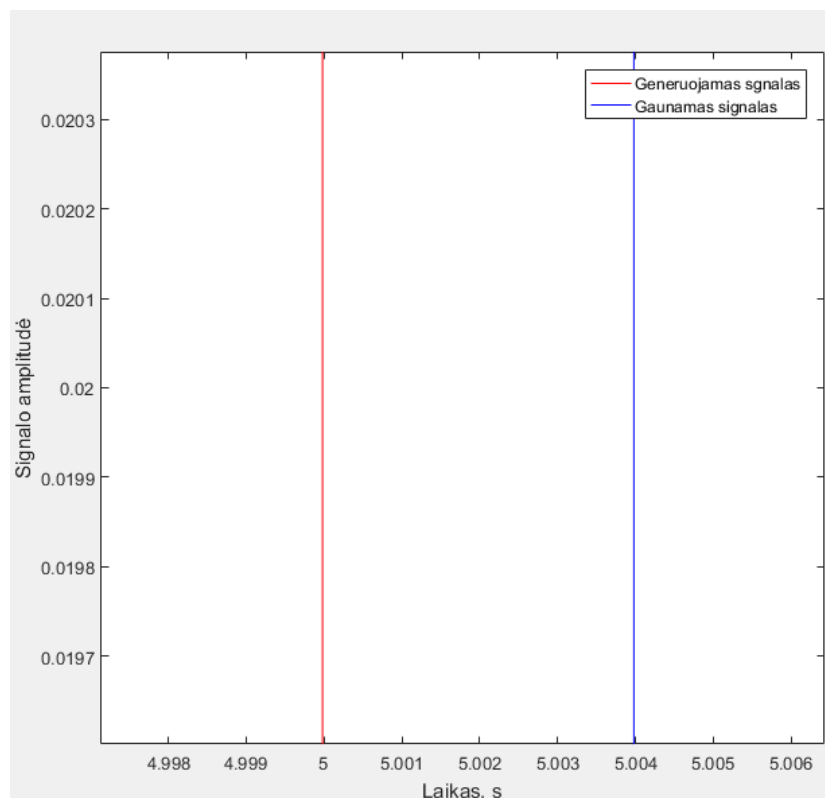


**37 pav.** Gautų rezultatų grafikas su priartintais laiko ašies duomenimis, 1 dalis

Įvertinus 37 paveiksle pateikto grafiko duomenis matyti, kad skirtumas tarp generuojamo signalo, pažymėto raudona linija ir gaunamo signalo, pažymėto mėlyna linija yra maždaug 4 milisekundės. Pateiktame grafike matomas tik vienas vėlinimo signalas, nes vėlinimo signalų laiko vertės yra vienodos.

Įvertinus 38 paveiksle pateikto grafiko duomenis matyti, kad skirtumas tarp generuojamo signalo, pažymėto raudona linija ir gaunamo signalo, pažymėto mėlyna linija yra maždaug 4 milisekundės.

Peržiūrėjus daugiau 31 paveiksle pateiktame grafike esančių verčių, nustatyta, kad vėlinimas tarp generuojamo signalo ir gaunamo signalo yra apytiksliai lygus 4 milisekundėms. Tačiau reikia įvertinti, kad *OMRON NX-ID3417* diskretinių įėjimų modulio įėjimo užlaikymo filtras pagal nutylėjimą yra 1 milisekundė.



**38 pav.** Gautų rezultatų grafikas su priartintais laiko ašies duomenimis, 2 dalis

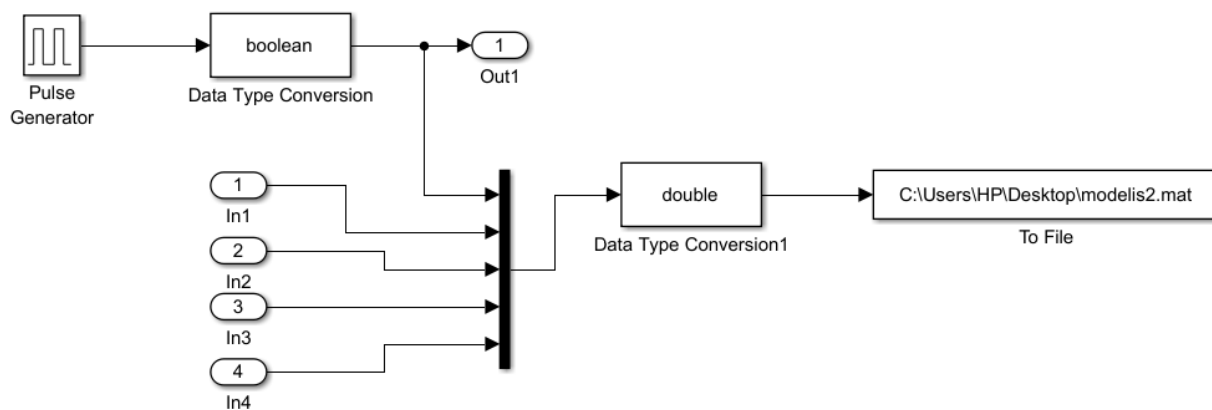
Šią vertę galima pakeisti nuo 0 iki 256 milisekundžių. Šio modelio maksimalus įjungimo vėlinimas siekia 20 mikrosekundžių, o maksimalus išjungimo užlaikymas 400 mikrosekundžių. Tyrime naudojamo diskretinių išėjimų modulio *OMRON NX-OD3256* maksimalus įjungimo vėlinimas siekia 0,5 milisekundės, o maksimalus išjungimo vėlinimas lygus 1 milisekundei.

Tyrime imami tik amplitudės kitimo nuo 1 iki 0 taškai, kurie reiškia išjungimą. Įvertinus diskretinio įėjimo modulio išjungimo užlaikymą ir diskretinio išėjimų modulio išėjimo išjungimo vėlinimo vertes, gaunamas signalo vėlinimo laikas lygus 2,6 milisekundės.

#### 4.2.2. *MATLAB Simulink* modelio sudarymas ir jo tyrimas, 1 dalis

Baigus tyrinėti bandomojo *MATLAB Simulink* modelio duomenis, sudaromas *MATLAB Simulink* modelis, kuriame naudojamas 1 išėjimų blokas ir 4 įėjimų blokai. Modelis skirtas tirti ar generuojant du ar daugiau signalų į skirtingus įėjimų modulio įėjimus gaunamas skirtumas tarp užfiksuotų įėjimų suveikimo laiko reikšmių. Šio modelio ekrano nuotrauka pateikta 39 paveiksle.

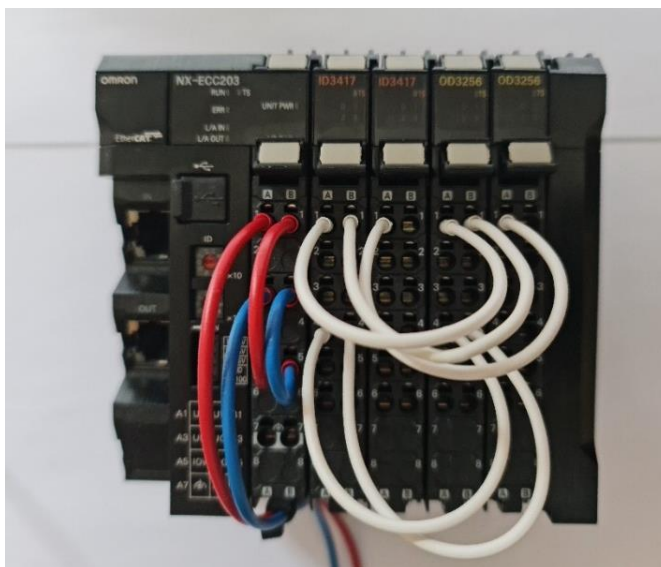
Šiame modelyje naudojamas impulsų generavimo blokas pavadinimu *Pulse generator*, kurio nustatymuose pasirinktas 1 Hz dažnis. Minimali reikšmė yra 0, o maksimali lygi 1. Šio bloko išėjimo signalas konvertuojamas į *boolean* duomenų tipą, naudojantis duomenų tipo keitimo bloku, kurio pavadinimas *data type conversion*. Iš duomenų keitimo bloko duomenys siunčiami į išėjimo bloką pavadinimu Out1. Šiam blokui *Beckhoff TwinCAT 3* programoje priskiriamas fizinis išėjimas. Kai į šį bloką ateina aukštas signalo lygis, fizinis išėjimas taip pat generuoja aukštą signalo lygį. Naudojamas fizinis išėjimas trumpu laidu sujungtas su greito sujungimo, 5 jungčių kaladėle, *Wago 221*. Iš šios jungties vienodo ilgio laidais prijungiami 4 fiziniai įėjimai. Šiems įėjimams *TwinCAT 3* programoje priskirti įėjimo blokai pavadinimais *In1, In2, In3, In4*. Kai į fizinį įėjimą prijungiamas aukštas signalo lygis, blokai *In1–In4* taip pat generuoja aukštą signalo lygį.



39 pav. Tyrimui naudojamas modelis

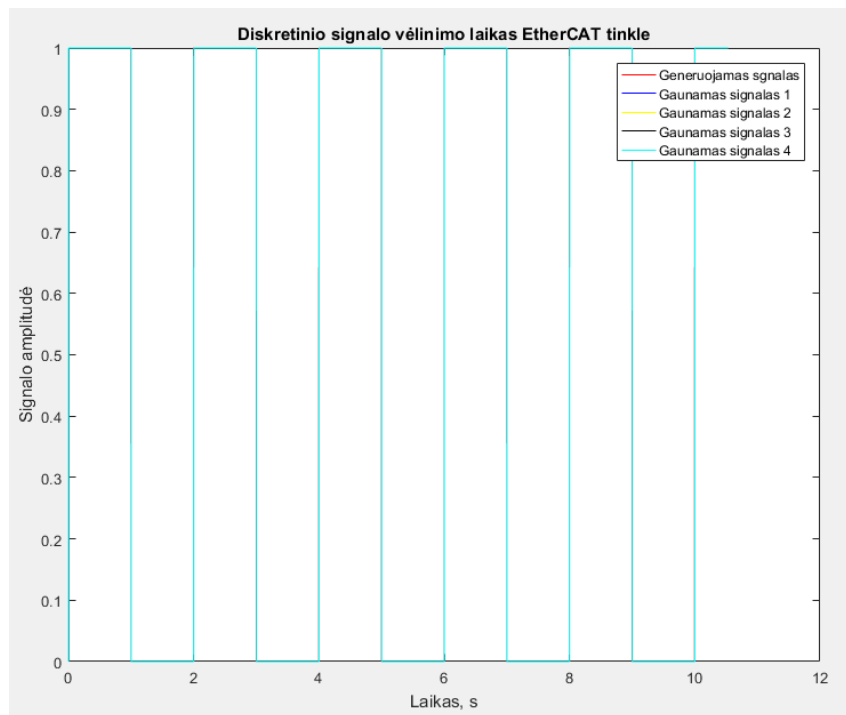
40 paveiksle pateiktas paveikslas parodo kaip pavaldaus *EtherCAT* prietaiso įėjimai ir išėjimai yra sujungti tarpusavyje. Pirmojo išėjimo modulio kontaktas pavadinimu *A1* laidu sujungtas su 5 jungčių *Wago 221* greitojo sujungimo kaladėle. Iš šios kaladėlės išvesti 4 vienodo ilgio laidai, kurie įjungti į pirmojo įėjimų modulio įėjimų jungtis: *A1*, *B1*, *A4*, *B4*.

Prijungus laidus prie reikiamų jungčių, *Beckhoff TwinCAT 3* programoje šiems panaudotiems fiziniams įėjimams ir išėjimams priskiriami sudaryto *MATLAB Simulink* modelio išėjimų ir įėjimų blokai. Naudojamam fiziniam išėjimui, kurio kontaktas *A1* priskiriamas *MATLAB Simulink* modelio blokas pavadinimu *Out1*. Naudojamiems fiziniams įėjimams priskiriami blokai: *A1* priskiriamas blokas *In1*, *B1* kontaktui priskiriamas blokas *In2*, *A4* kontaktui priskiriamas blokas *In3* ir *B4* kontaktui priskiriamas blokas pavadinimu *In4*.



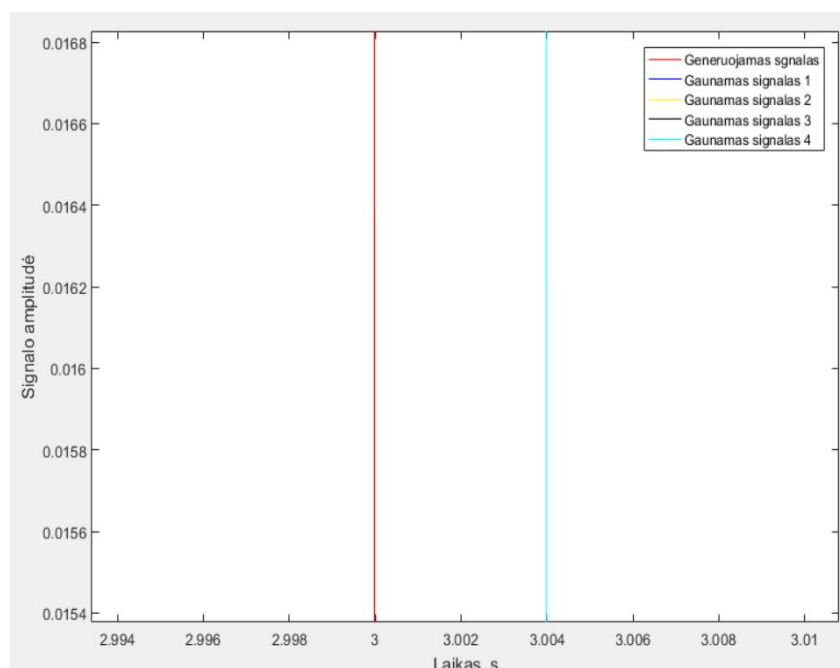
40 pav. Diskretinių įėjimų ir išėjimų jungimas

Visus įėjimus ir išėjimus sujungus kaip parodyta 40 paveiksle, *MATLAB Simulink* modelį įkėlus į *TwinCAT 3* programą ir gavus simuliacijos duomenis, gautų duomenų peržiūrėjimui paleidžiama *MATLAB* programa ir sukuriamas naujas *MATLAB* programos langas, naudojantis *load* ir *plot* funkcijomis gauti simuliacijos duomenys atvaizduojami grafiškai. 41 paveiksle pateiktas tyrimui naudojamo modelio rezultatų grafikas.



**41 pav.** Tyrimui naudojamam modeliui rezultatų grafikas

41 paveiksle grafike raudona spalva pažymėtas linija žymi signalą, kuris yra siunčiamas į išėjimo blokus *Out1–Out4*. Mėlyna spalva pažymėta linija nurodo signalą 1, gaunamą iš įėjimo bloko *In1*, geltona spalva žymi gaunamą signalą 2 iš įėjimo bloko pavadinimu *In2*, juoda spalva žymi gaunamą signalą 3 iš įėjimo bloko pavadinimu *In3*, cyan spalva žymi gaunamą signalą 4 iš įėjimo bloko pavadinimu *In4*.

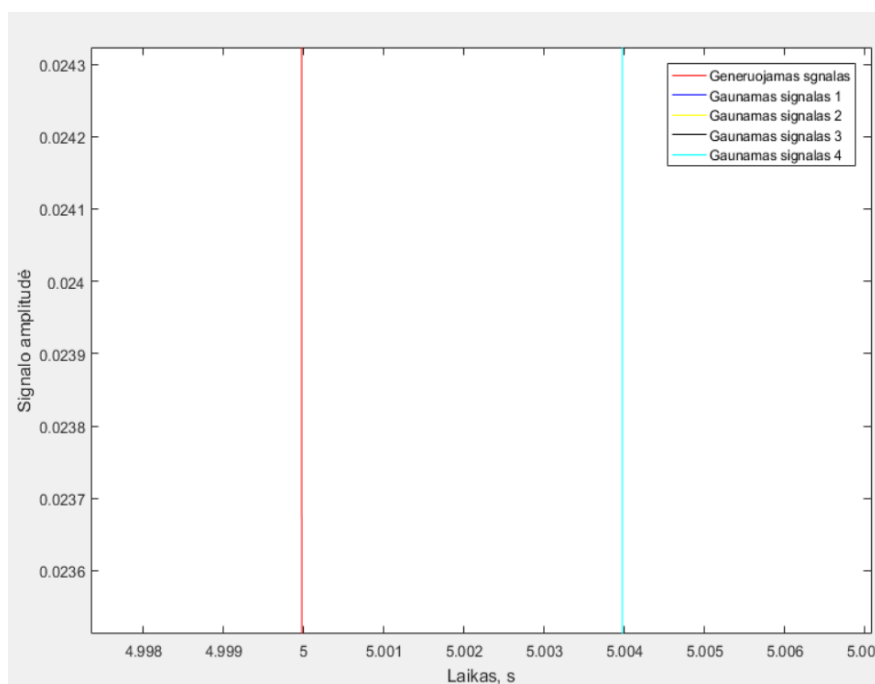


**42 pav.** Tyrimui naudojamam modeliui rezultatų grafikas su priartintais laiko ašies duomenimis, 1 dalis

Įvertinus 42 paveiksle pateikto grafiko duomenis matyti, kad skirtumas tarp generuojamo signalo, pažymėto raudona linija ir gaunamo signalo, pažymėto cyan spalvos linija yra maždaug 4



milisekundės. Pateiktame grafike matomas tik vienas vėlinimo signalas, nes vėlinimo signalų laiko vertės yra vienodos.



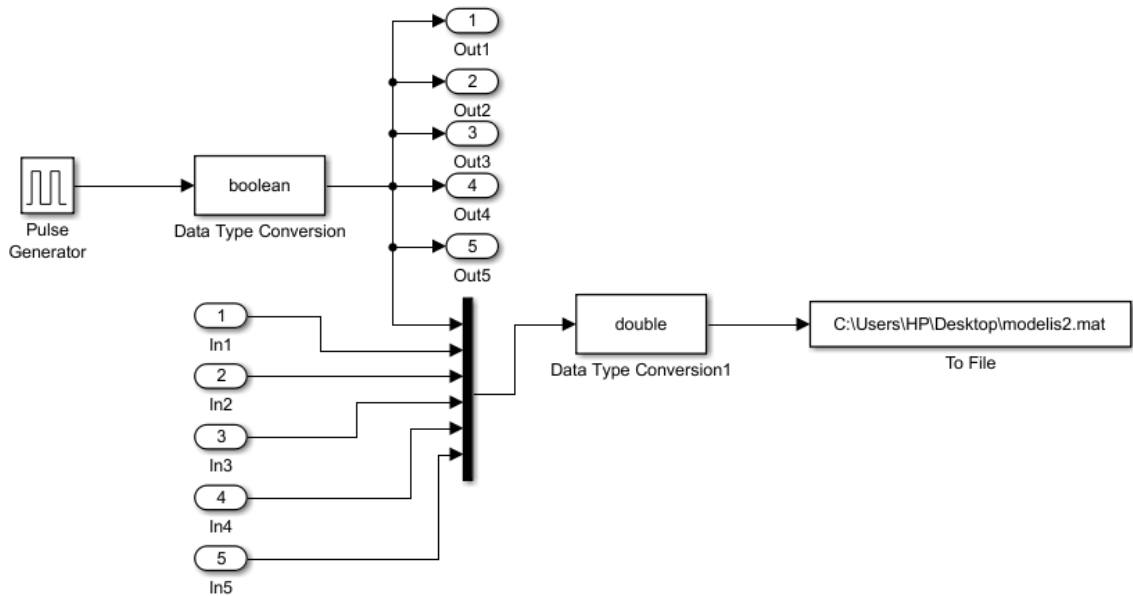
**43 pav.** Tyrimui naudojamo modelio rezultatų grafikas su priartintais laiko ašies duomenimis, 2 dalis

Įvertinus 43 paveiksle pateikto grafiko duomenis matyti, kad skirtumas tarp generuojamo signalo, pažymėto raudona linija ir gaunamo signalo, pažymėto cyan spalvos linija yra maždaug 4 milisekundės.

Peržiūrėjus daugiau 41 paveiksle pateiktame grafike esančių verčių, nustatyta, kad vėlinimas tarp generuojamo signalo ir gaunamo signalo yra apytiksliai lygus 4 milisekundėms. Tačiau reikia įvertinti, kad *OMRON NX-ID3417* diskretinių įėjimų modulio įėjimo užlaikymo filtras pagal nutylėjimą yra 1 milisekundė. Šią vertę galima pakeisti nuo 0 iki 256 milisekundžių. Šio modelio maksimalus įjungimo vėlinimas siekia 20 mikrosekundžių, o maksimalus išjungimo užlaikymas 400 mikrosekundžių. Tyrime naudojamo diskretinių išėjimų modulio *OMRON NX-OD3256* maksimalus įjungimo vėlinimas siekia 0,5 milisekundės, o maksimalus išjungimo vėlinimas lygus 1 milisekundei. Tyrime imami tik amplitudės kitimo nuo 1 iki 0 taškai, kurie reiškia įjungimą, kadangi tiek diskretinių įėjimų, tiek diskretinių išėjimų vėlinimo laikai šiais atvejais yra mažesni, taip mažinant matavimo paklaidas. Įvertinus diskretinio įėjimo modulio įėjimo ir diskretinio išėjimų modulio išėjimo vėlinimo vertes, gaunamas signalo vėlinimo laikas lygus 2,6 milisekundės.

#### **4.2.3. MATLAB Simulink modelio sudarymas ir jo tyrimas, 2 dalis**

Baigus tyrimą, naudojant modelį su vienu išėjimo signalu ir 4 įėjimo signalais, sudaromas *MATLAB Simulink* modelis skirtas tirti kaip keičiasi įėjimo ir išėjimo signalų vėlinimo laikai *EtherCAT* prietaise, kai vienu metu yra junginėjami keli išėjimai ar įėjimai, taip pat keičiamas dažnis. Taip pat tiriama ar *EtherCAT* prietaise naudojamų diskretinių įėjimų ir išėjimų modulių vieta prietaiso konfigūracijoje turi įtakos vėlinimo laikams. Šiame modelyje naudojamas maksimalus bazinės *Beckhoff TwinCAT 3* programos paprogramės *TE 1400* leidžiamas įėjimų ir išėjimų skaičius, kuris yra 5 įėjimo signalai ir 5 išėjimo signalai. Šio modelio ekrano nuotrauka pateikta 44 paveiksle.



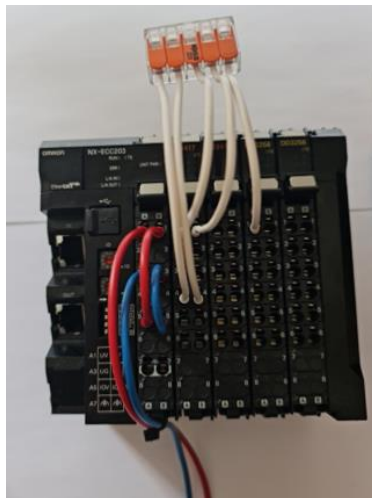
44 pav. Tyrimui naudojamas modelis

Šiame modelyje naudojamas impulsų generavimo blokas pavadinimu *Pulse generator*, kurio nustatymuose pasirinktas vienos sekundės dažnis. Minimali reikšmė yra 0, o maksimali lygi 1. Šio bloko išėjimo signalas konvertuojamas į *boolean* duomenų tipą, naudojantis duomenų tipo keitimo bloku, kurio pavadinimas *data type conversion*. Iš duomenų keitimo bloko duomenys siunčiami į išėjimo blokus pavadinimais *Out1*, *Out2*, *Out3*, *Out4*, *Out5*. Šiems blokams *Beckhoff TwinCAT 3* programoje priskiriami fiziniai išėjimai. Kai į šiuos blokus ateina aukštas signalo lygis, fiziniai išėjimai taip pat generuoja aukštą signalo lygį. Šiems įėjimams *TwinCAT 3* programoje priskirti įėjimo blokai pavadinimais *In1*, *In2*, *In3*, *In4*, *In5*. Kai į fizinį įėjimą prijungiamas aukštas signalo lygis, blokai *In1–In5* taip pat generuoja aukštą signalo lygį. Šis aukštas – žemas signalo lygis vėliau atvaizduojamas grafiškai modeliavimo rezultatų grafike.

45 paveiksle pateiktas paveikslas parodo kaip pavaldaus *EtherCAT* prietaiso įėjimai ir išėjimai yra sujungti tarpusavyje. Pirmojo išėjimo modulio kontaktas pavadinimu *A1* laidu sujungtas su pirmojo įėjimo modulio kontaktu *A1*, pirmojo išėjimo modulio kontaktas pavadinimu *A4* laidu sujungtas su pirmojo įėjimo modulio kontaktu *A4*. Analogiškai sujungti kontaktai pavadinimais *B1* ir *B4*. Penktasis išėjimo signalas, kurį generuoja antrasis išėjimo modulis, tokiu pat trumpu laidu perduodamas į antrąjį įėjimo modulį. Šiam signalui perduoti antrojo išėjimų modulio kontaktas pavadinimu *A1* sujungiamas su įėjimų modulio kontaktu *A1*.

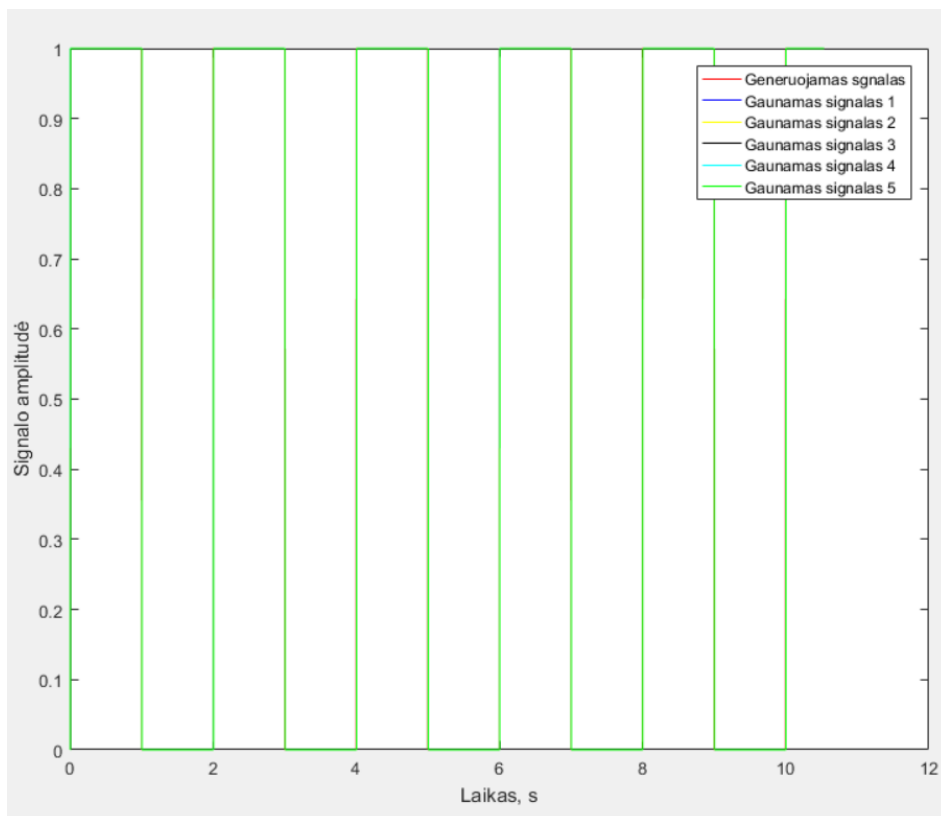
Prijungus laidus prie reikiamų jungčių, *Beckhoff TwinCAT 3* programoje šioms panaudotiems fiziniams įėjimams ir išėjimams priskiriami sudaryto *MATLAB Simulink* modelio išėjimų ir įėjimų blokai. Naudojamiems pirmojo išėjimų modulio fiziniams išėjimams priskiriami *MATLAB Simulink* modelio išėjimų blokai tokia tvarka: *A1 – Out1*, *B1 – Out2*, *A4 – Out3*, *B4 – Out4*. Naudojamam antrojo išėjimų modulio fiziniam išėjimui priskiriamas *MATLAB Simulink* modelio išėjimo blokas *Out5*.

Naudojamiems pirmojo įėjimų modulio fiziniams įėjimams priskiriami *MATLAB Simulink* modelio įėjimų blokai tokia tvarka: *A1 – In1*, *B1 – In2*, *A4 – In3*, *B4 – In4*. Naudojamam antrojo įėjimų modulio fiziniam įėjimui priskiriamas *MATLAB Simulink* modelio įėjimo blokas *In5*.



**45 pav.** Diskretinių įėjimų ir išėjimų jungimas

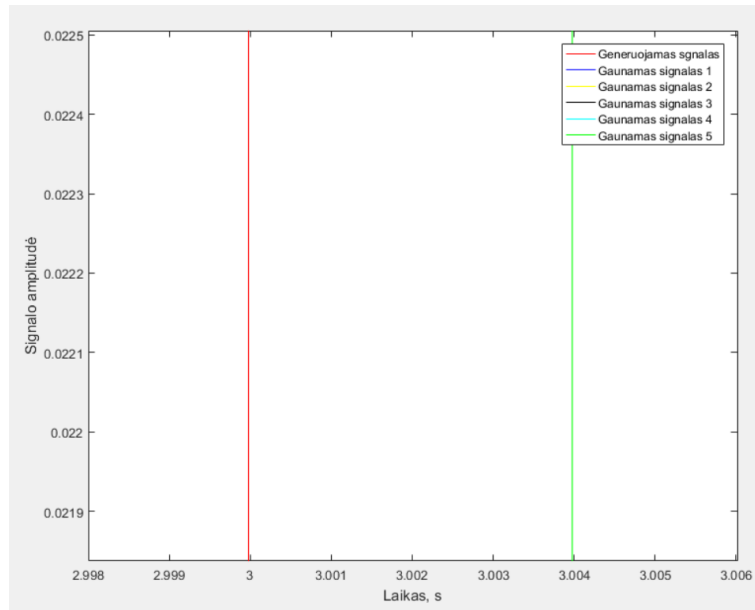
Gautų duomenų peržiūrėjimui paleidžiama *MATLAB* programa ir sukuriamas naujas *MATLAB* programos langas, naudojantis *load* ir *plot* funkcijomis, gauti simuliacijos duomenys atvaizduojami grafiškai. 46 paveiksle pateiktas tyrimui naudojamo modelio rezultatų grafikas.



**46 pav.** Tyrimui naudojamo modelio rezultatų grafikas

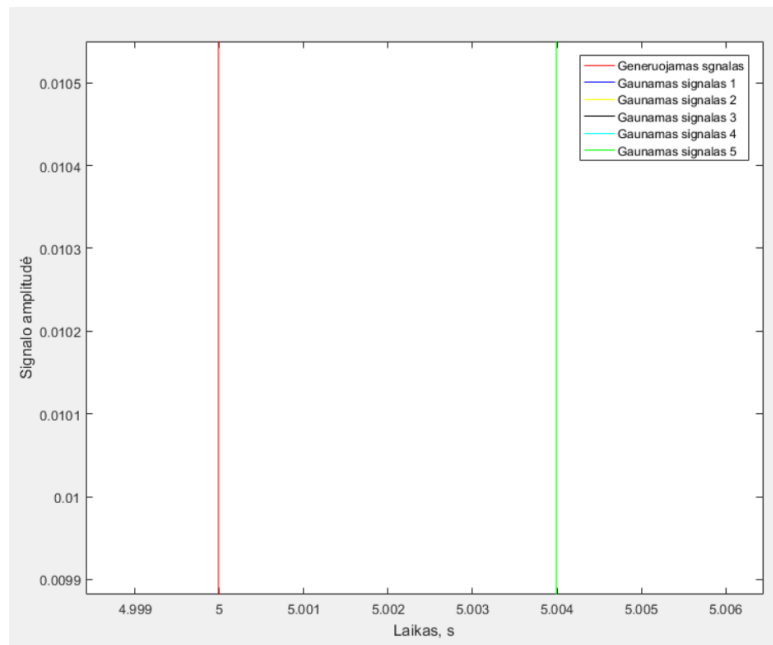
46 paveiksle grafike raudona spalva pažymėta linija žymi signalą, kuris yra siunčiamas į išėjimo blokus *Out1–Out5*. Mėlyna spalva pažymėta linija nurodo signalą gaunamą 1 iš įėjimo bloko *In1*, geltona spalva žymi gaunamą signalą 2 iš įėjimo bloko pavadinimu *In2*, juoda spalva žymi gaunamą

signalą 3 iš įėjimo bloko pavadinimu *In3*, cyan spalva žymi gaunamą signalą 4 iš įėjimo bloko pavadinimu *In4*, žalia spalva žymi gaunamą signalą 5 iš įėjimo bloko pavadinimu *In5*.



**47 pav.** Tyrimui naudojamo modelio rezultatų grafikas su priartintais laiko ašies duomenimis, 1 dalis

Įvertinus 47 paveiksle pateikto grafiko duomenis matyti, kad skirtumas tarp generuojamo signalo, pažymėto raudona linija ir gaunamo signalo, pažymėto žalia linija yra maždaug 4 milisekundės. Pateiktame grafike matomas tik vienas vėlinimo signalas, nes vėlinimo signalų laiko vertės yra vienodos.



**48 pav.** Tyrimui naudojamo modelio rezultatų grafikas su priartintais laiko ašies duomenimis, 2 dalis

Įvertinus 48 paveiksle pateikto grafiko duomenis matyti, kad skirtumas tarp generuojamo signalo, pažymėto raudona linija ir gaunamo signalo, pažymėto žalia linija yra maždaug 4 milisekundės.

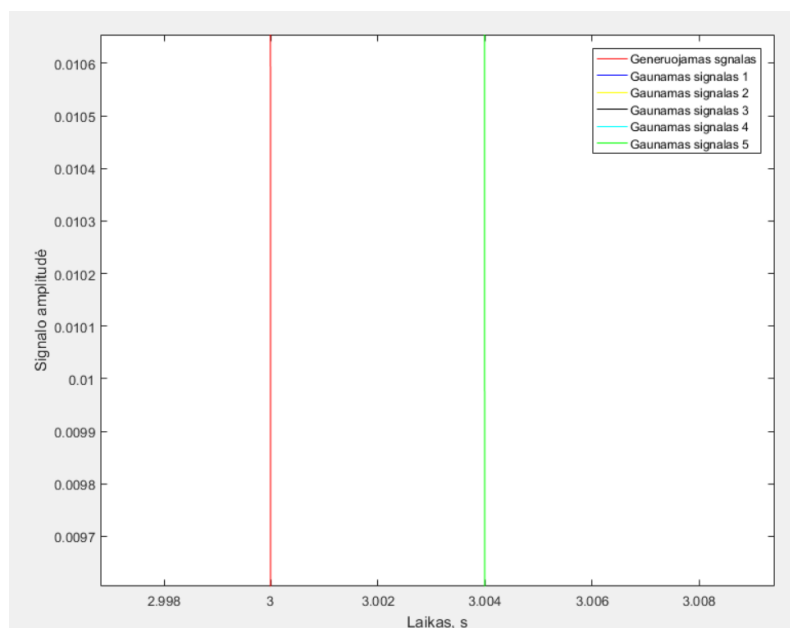
Pateiktame grafike matomas tik vienas vėlinimo signalas, nes vėlinimo signalų laiko vertės yra vienodos.

Peržiūrėjus daugiau 46 paveiksle pateiktame grafike esančių verčių, nustatyta, kad vėlinimas tarp generuojamo signalo ir gaunamo signalo yra apytiksliai lygus 4 milisekundėms. Tačiau reikia įvertinti, kad *OMRON NX-ID3417* diskretinių įėjimų modulio įėjimo užlaikymo filtras pagal nutylėjimą yra 1 milisekundė. Šią vertę galima pakeisti nuo 0 iki 256 milisekundžių. Šio modelio maksimalus įjungimo vėlinimas siekia 20 mikrosekundžių, o maksimalus išjungimo užlaikymas 400 mikrosekundžių. Tyrime naudojamo diskretinių išėjimų modulio *OMRON NX-OD3256* maksimalus įjungimo vėlinimas siekia 0,5 milisekundės, o maksimalus išjungimo vėlinimas lygus 1 milisekundei. Įvertinus diskretinio įėjimo modulio įėjimo ir diskretinio išėjimų modulio išėjimo išjungimo vėlinimo vertes, gaunamas signalo vėlinimo laikas lygus 2,6 milisekundės.

Lyginant gautas vėlinimo vertes su ankstesniame tyrime gautomis reikšmėmis, kai modelyje naudojamas tik vienas įėjimo ir išėjimo blokas, reikšmės gaunamos lygios, tai reiškia, kad vienu metu naudojant tiek vieną, tiek kelis, šiuo atveju 5, įėjimo ir išėjimo signalus, vėlinimo laikas gaunamas vienodas. Taigi galima daryti išvadą, kad *EtherCAT* prietaise naudojant skirtingus įėjimų ir išėjimų kiekius vienu metu, duomenų apsikeitimo vėlinimo laikas nesikeičia.

Taip pat iš grafiko matyti, kad pirmųjų 4 įėjimo signalų, kurie gaunami iš pirmojo diskretinių įėjimų modulio ir penktojo įėjimo signalo, kuris gaunamas iš antrojo įėjimų modulio laikai yra vienodi. Taigi galima daryti išvadą, kad nepriklausomai į kurį iš kelių naudojamų įėjimų modulių signalas bus generuojamas, gaunamo signalo vėlinimo laikas bus vienodas. Tai reiškia, kad įėjimų modulio vieta valdiklio išdėstyme signalo vėlinimams įtakos neturi.

Žemiau pateiktame 49 paveiksle pateiktas grafikas vaizduoja vėlinimo laiką tarp generuojamo ir gaunamų signalų, kai naudojamas 44 paveiksle pateiktas modelis, bet generuojamo signalo dažnis sumažintas. Signalo periodas vietoje ankstesnėje tyrimo dalyje naudoto 1 sekundės pakeistas į 0,5 sekundės.



**49 pav.** Tyrimui naudojamo modelio rezultatų grafikas su priartintais laiko ašies duomenimis ir sumažintu generuojamo signalo dažniu

Kaip galima matyti iš 49 paveiksle grafiškai pateiktų simuliacijos rezultatų, vėlinimo laikas tarp generuojamo ir gaunamo signalo prieš įvertinant naudojamos įrangos vėlinimus sudaro 4 milisekundes. Kaip ir ankstesnėse tyrimo dalyse įvertinus naudojamos įrangos vėlinimus, gaunamas 2,6 milisekundžių vėlinimo laikas. Palyginus gautą vėlinimo laiką, naudojant 0,5 sekundės generuojamo signalo dažnį su ankstesnėje tyrimo dalyje gautais vėlinimo laikais, kai buvo naudojamas 1 sekundės generuojamo signalo dažnis, vėlinimo laikas gaunamas vienodas. Todėl galima daryti išvadą, kad nepriklausomai nuo kokių dažnių yra įjungiamas / išjungiamas įėjimų modulio kanalas, laikas per kurį duomenys bus perduoti į valdiklį nesikeičia.

#### **4.3. Rekomendacijos leidžiančios efektyvinti *Ethernet* šakos pramonės komunikacinį tinklą**

Įvertinus atliktų tyrimų rezultatus, rekomenduojama projektuojant automatines valdymo sistemas, rinktis kuo trumpesnę atstumą tinklo kabeliams tarp tinklo prietaisų, esant galimybei rinktis *EtherCAT* vietoje *PROFINET* realaus valdymo tinklo technologijos, kadangi tyrimo metu nustatyta, kad *PROFINET* pasižymi 50,6 proc. didesniu vėlinimo laiku tinklą jungiant nuosekliu tinklo jungimo būdu. Tinklus jungiant žvaigžde, *PROFINET* pasižymėjo 32.7 % ilgesniu vėlinimo laiku.

Naudojant *EtherCAT* realaus laiko valdymo tinklą, rekomenduojama rinktis nuoseklią arba medžio topologiją. Tinklą jungiant vienu iš šių būdų, kai tinkle yra 100 pavaldžių prietaisų gaunamas vėlinimo laikas siekia 0,158 milisekundės, kai pasirinkus tinkle jungimą žvaigžde vėlinimo laikas siektų 0,221 milisekundės.

Naudojant *PROFINET* realaus laiko valdymo tinklą, rekomenduojama rinktis nuoseklią topologiją. Vėlinimo laikas tinklą jungiant nuosekliai, kai tinkle 100 pavaldžių prietaisų, siekia 0,32 milisekundės. Sekantis rekomenduojamas tinklo jungimo būdas yra tinklo jungimas žvaigžde. Tinklą jungiant žvaigžde, kai tinkle 100 pavaldžių prietaisų, vėlinimas siekia 0,328 milisekundės. Didžiausiu vėlinimo laiku iš visų *PROFINET* valdymo tinklo topologijų pasižymi jungimas medžiu. Tinklą jungiant medžio topologija, vėlinimas siekia 0,339 milisekundės.

## Išvados

1. Išanalizuota mokslinė literatūra, susijusi su realaus laiko valdymo tinklais. Aprašytos realaus laiko valdymo tinklų *EtherCAT* ir *PROFINET* technologijos, naudojama įranga. Išnagrinėti šių tinklų protokolai ir aprašytas veikimas. Nustatyta, kad *EtherCAT* technologija duomenis siunčia sinchroniškai, griežtai nustatytu pavaldžių prietaisų apklausimo laiku. Vienu metu siunčiamas tik vienas duomenų paketas, todėl nėra galimybės įvykti duomenų paketų kolizijai. *PROFINET*, priešingai nei *EtherCAT*, duomenis siunčia asinchroniškai. Valdantysis prietaisas kiekvienam pavaldžiam prietaisui siunčia po paketą duomenų. Pavaldus prietaisas gavęs ir apdorojęs duomenis, siunčia duomenų paketą atgal į valdantįjį prietaisą. Kadangi kiekvienas prietaisas gauna ir siunčia po atskirą duomenų paketą, atsiranda tikimybė įvykti duomenų paketų kolizijai, todėl *PROFINET* technologijoje naudojamas *CSMA/CD* protokolas.
2. Išskirti realaus laiko *Ethernet* šakos pramonės komunikacinių technologijų privalumai ir trūkumai. Apžvelgtos naudojamos technologinės priemonės ir taikymo metodai. Nustatyti *EtherCAT* technologijos privalumai: griežtai nustatytas duomenų apsikeitimo ciklo laikas, mažesni vėlinimo laikai nei *PROFINET* technologijos. Nustatyti trūkumai: tinkle gali būti tik vienas valdantysis prietaisas, mažas suderinamumas su periferiniais įrenginiais. Nustatyti *PROFINET* technologijos privalumai: tinkle gali būti keli valdantieji prietaisai, tinklas lengvai suderinamas su periferiniais įrenginiais. Nustatyti trūkumai: nuolatos kintantis duomenų apsikeitimo laikas tarp valdančiojo ir pavaldžių prietaisų, apsunkinantis tikslaus ašių pozicionavimo reikalaujantis užduotis.
3. Sudarius metodus tinkančius tirti *EtherCAT* ir *PROFINET* tinklų laikinius parametrus, atsižvelgiant į tinklų topologijas ir aptarnaujamų tinklo prietaisų skaičių ir atlikus tyrimą nustatyta, kad *EtherCAT* valdymo tinklo vėlinimo laikai esant bet kuriai iš šiam tinklui leidžiamų topologijų, yra mažesni nei *PROFINET* valdymo tinklo, tinklą jungiant analogiškais tinklo topologijomis. Lyginant tris *EtherCAT* tinklo jungimo topologijas, didžiausiu vėlinimo laiku pasižymi tinklo jungimas žvaigžde. Šis laikas tinkle esant 100 pavaldžių prietaisų sudaro 0,221 milisekundės. Mažiausiu vėlinimo laiku pasižymi jungimas nuosekliu būdu ir jungimas medžio būdu. Šių abiejų jungimo tipų vėlinimo laikai yra vienodi. Tinkle, kuriame yra 100 pavaldžių prietaisų, tinklą jungiant bet kuriuo iš šių dviejų būdų, vėlinimo laikas siekia 0,158 milisekundės. *PROFINET* atveju didžiausiu vėlinimo laiku pasižymi jungimas medžio topologija. Tinklą jungiant tokiu būdu, kai tinkle yra 100 pavaldžių prietaisų, vėlinimo laikas siekia 0,339 milisekundės. Mažiausiu vėlinimo laiku esant bet kokiam prietaisų kiekiui pasižymi nuoseklus *PROFINET* tinklo jungimo būdas. Nuosekliai sujungus 100 prietaisų, vėlinimo laikas yra 0,32 milisekundės. *PROFINET* tinklą jungiant žvaigžde, kai pavaldžių prietaisų skaičius lygus 100, vėlinimo laikas siekia 0,328 milisekundės.
4. Atlikus tyrimą su sudarytais *MATLAB Simulink* modeliais, nustatyta, kad *EtherCAT* prietaise naudojant skirtingus įėjimų ir išėjimų kiekius vienu metu, duomenų apsikeitimo vėlinimo laikas nesikeičia. Taip pat nustatyta, kad nepriklausomai nuo to, koku dažniu yra įjungiamas/išjungiamas įėjimų modulis, laikas per kurį duomenys bus perduoti į valdiklį nesikeičia. Visais atvejais pamatuotas vėlinimo laikas įvertinus paklaidas siekė 2,6 milisekundės.
5. Sudarytos rekomendacijos leidžiančios efektyvinti *Ethernet* šakos pramonės komunikacinį tinklą. Mažesniu vėlinimo laiku pasižymi *EtherCAT* realaus laiko valdymo tinklas. Mažiausiu vėlinimo laiku pasižyminti tinklo topologija tiek *EtherCAT*, tiek *PROFINET* tinkluose yra tinklo jungimas nuosekliu būdu.

## Literatūros sąrašas

1. NEUMANN, P. PÖSCHMANN, A. Ethernet-based real-time communications with PROFINET IO. *WSEAS Transactions on Communications*. 2005, 235–245.
2. WU, X., XIE, L. Performance evaluation of industrial Ethernet protocols for networked control application. *Control Engineering Practice* [interaktyvus]. 2019, 208–217. Prieiga per: <https://doi.org/10.1016/j.conengprac.2018.11.022>.
3. ROSTAN, M., FRANCISCO, S. *High Speed Industrial Ethernet for Semiconductor Equipment Presented at the SEMI Technology Symposium: Innovations in Semiconductor Manufacturing (STS: ISM)*. 2006.
4. ROSTAN, M. et al. EtherCAT enabled advanced control architecture. *ASMC (Advanced Semiconductor Manufacturing Conference) Proceedings*. 2010, 39–44.
5. KANG, C. et al. Design of EtherCAT slave module. *2011 IEEE International Conference on Mechatronics and Automation, ICMA 2011*. 2011, 1600–1604.
6. KNEZIC, M. et al. Increasing EtherCAT performance using frame size optimization algorithm. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. 2011, 1–4.
7. LIU, J. et al. A high precision clock synchronization algorithm for the EtherCAT. *Proceedings of the 2017 12th IEEE Conference on Industrial Electronics and Applications*. 2018, 1369–1374.
8. LEI, W., JUNYAN, Q. The real-time networked data gathering systems based on EtherCAT. *Proceedings - 2009 International Conference on Environmental Science and Information Application Technology, ESIAT 2009*. 2009, 513–515.
9. ORFANUS, D. et al. EtherCAT-based platform for distributed control in high-performance industrial applications. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. 2013.
10. PRYTZ, G., SKAALVIK, J. Redundant and synchronized EtherCAT network. *2010 International Symposium on Industrial Embedded Systems, SIES 2010 - Conference Proceedings*. 2010, 201–204.
11. CENA, G. et al. On the accuracy of the distributed clock mechanism in EtherCAT. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*. 2010, 43–52.
12. CENA, G. et al. Evaluation of EtherCAT distributed clock performance. *IEEE Transactions on Industrial Informatics*. 2012, 20–29.
13. BENZAID, M. et al. Analysis and simulation of Fast-OLSR. *IEEE Vehicular Technology Conference*. 2003, 1788–1792.
14. SENO, L., ZUNINO, C. A simulation approach to a Real-Time Ethernet protocol: EtherCAT. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. 2008, 440–443.
15. KNEZIC, M. Topology aspects in EtherCAT networks. *Proceedings of EPE-PEMC 2010 - 14th International Power Electronics and Motion Control Conference*. 2010, T1-1-T1-6.
16. FERRARI, P. et al. Large PROFINET IO RT networks for factory automation: A case study. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. 2011, 1–4.



17. JASPERNEITE, J., FELD, J. PROFINET: An integration platform for heterogeneous industrial communication systems. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. 2005, 815–822.
18. KLEINES, H. et al. Performance aspects of PROFINET IO. *2007 15th IEEE-NPSS Real-Time Conference, RT*. 2007, 290–294.
19. FERRARI, P. et al. Performance analysis of PROFINET networks. *Computer Standards and Interfaces*. 2006, 369–385.
20. KJELLSSON, J. et al. Integration of a wireless I/O interface for PROFIBUS and PROFINET for factory automation. *IEEE Transactions on Industrial Electronics*. 2009, 4279–4287.
21. YANG, M., LI, G. Analysis of PROFINET IO communication protocol. *Proceedings - 2014 4th International Conference on Instrumentation and Measurement, Computer, Communication and Control, IMCCC 2014*. 2014, 945–949.
22. ANTOLOVIĆ, M. et al. PLC communication using PROFINET: Experimental results and analysis. In *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. 2006, 1–4.
23. WU, X. et al. Network delay analysis of EtherCAT and PROFINET IRT protocols. *IECON Proceedings (Industrial Electronics Conference)*. 2014, 2597–2603.
24. FELD, J. Profinet - Scalable factory communication for all applications. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*. 2004, 33–38.
25. SCHUMACHER, M. et al. A new approach for increasing the performance of the industrial ethernet system PROFINET. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*. 2008, 159–167.
26. PRYTZ, G. A performance analysis of EtherCAT and PROFINET IRT. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. 2008, 408–415.
27. TRAINING, V. *PROFINET Installation Guidelines*. 2016.
28. SCHLESINGER, R. et al. New approach for improvements and comparison of high performance real-time ethernet networks. *19th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2014*. 2014, 1–4.
29. FERRARI, P. et al. Experimental analysis to estimate jitter in PROFINET IO Class 1 networks. In *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. 2006, 429–432.
30. HANZÁLEK, Z. et al. Profinet IO IRT message scheduling. *Proceedings - Euromicro Conference on Real-Time Systems*. 2009, 57–65.