



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Reaktoriaus valdymo sistemos su Beckhoff programuojamu loginiu valdikliu tyrimas

Baigiamasis magistro projektas

Rapolas Račys

Projekto autorius

Doc. dr. Leonas Balaševičius

Vadovas

Kaunas, 2021



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Reaktoriaus valdymo sistemos su Beckhoff programuojamu loginiu valdikliu tyrimas

Baigiamasis magistro projektas

Valdymo technologijos (6211EX014)

Rapolas Račys

Projekto autorius

Doc. dr. Leonas Balaševičius

Vadovas

Doc. dr. Gintaras Dervinis

Recenzentas

Kaunas, 2021



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Rapolas Račys

Reaktoriaus valdymo sistemos su Beckhoff programuojamu loginiu valdikliu 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.

Rapolas Račys

Patvirtinta elektroniniu būdu

Račys, Rapolas. Reaktoriaus valdymo sistemos su *Beckhoff* programuojamu loginiu valdikliu tyrimas. Magistro baigiamasis projektas vadovas doc. dr. Leonas Balaševičius; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas.

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

Reikšminiai žodžiai: programuojamas loginis valdiklis, *MAL*TAB/*SIMULINK*, reguliatorius.

Kaunas, 2021. 56 p.

Santrauka

Šis baigiamasis magistro projektas skirtas ištirti *MATLAB/SIMULINK* programinės įrangos pritaikymą realiai sistemai, kurioje nagrinėjamas reaktoriaus temperatūros ir skysčio valdymas imitaciniame stende. Projekto metu siekiama nustatyti kaip tiksliai galima perkelti teorinį proceso modelį į *Beckhoff* pramoninį kompiuterį. Pasinaudojus *TwinCAT 3* realaus laiko valdymo aplinka, siekiama atlikti optimizuotą proceso valdymą.

Račys, Rapolas. Reactor control system with Beckhoff programmable logic controller research. Master's Final Degree Project / supervisor Assoc. Prof. Dr., Leonas Balaševičius; Faculty of Electrical and Electronics Engineering, Kaunas University of Technology.

Study field and area (study field group): electronics engineering, engineering science.

Keywords: programmable logic controller, MATLAB/SIMULINK, regulator.

Kaunas, 2021. 56 pages.

Summary

The goal of this Master's thesis project is to examine *MATLAB/SIMULINK* software application to a real system, which examines reactor temperature and liquid control in an imitational stand. The goal for this project's duration was to establish applicational accuracy of using a theoretical process model on a *Beckhoff* industrial computer. The aim was to achieve optimized process control while using *TwinCAT 3* real time control environment.

Turinys

Lentelių sąrašas	7
Paveikslų sąrašas	8
Santrumpų ir terminų sąrašas	10
Įvadas.....	11
1. Valdymo sistemos analizė	13
1.1. <i>Beckhoff</i> kompanija	13
1.2. CX serijos pramoniniai kompiuteriai	14
1.3. <i>TwinCAT 3</i>	15
1.4. <i>MATLAB/SIMULINK</i>	16
1.5. <i>TwinCAT 3</i> sąsaja ir <i>Target for MATLAB/SIMULINK</i>	17
1.6. Įvairių PLV gamintojų programinės įrangos palyginimas	18
1.7. Reaktoriaus stendas	22
1.7.1. Reaktoriaus proceso aprašymas.....	22
1.8. Techninė įranga	23
1.8.1. PLV I/O maitinimo šaltinis	23
1.8.2. Bendras sistemos maitinimo šaltinis	24
1.8.3. Diskretinių išėjimų modulis	25
1.8.4. Analoginių įėjimų modulis	26
1.8.5. Analoginių išėjimų modulis	27
2. Tiriamoji dalis.....	29
2.1. Techninės įrangos prijungimas.....	29
2.2. Programinės įrangos konfigūravimas	29
2.2.1. <i>TwinCAT 3</i> konfigūravimas.....	29
2.2.2. <i>MATLAB/SIMULINK</i> konfigūravimas	32
2.3. <i>MATLAB/SIMULINK</i> modelio sudarymas temperatūros reguliavimui	33
2.3.1. Tolydinis reguliatorius.....	33
2.3.2. Diskretinis PI reguliatorius.....	36
2.4. <i>MATLAB/SIMULINK</i> modelio įkėlimas į <i>TwinCAT 3</i>	39
2.5. Lygio reguliavimas	42
3. Reaktoriaus valdymo sistemos tyrimas	44
3.1. PI diskretinis reguliatorius.....	44
3.2. <i>Fuzzy</i> reguliatorius.....	46
3.3. Modeliavimo rezultatai.....	52
Išvados	53
Literatūros sąrašas	54

Lentelių sąrašas

1 lentelė. CX2040 procesoriaus techniniai duomenys [14]	15
2 lentelė. PLV valdiklių programavimo kalbų palaikymas.....	18
3 lentelė. CX2100-0014 maitinimo šaltinio techniniai duomenys [29]	24
4 lentelė. Phoenix Contact Power UNO maitinimo šaltinio techniniai duomenys [31].....	25
5 lentelė. Beckhoff EL2809 diskretinių išėjimų modulio techniniai duomenys [33]	26
6 lentelė. Beckhoff EL3104 analoginių įėjimų modulio techniniai duomenys [35]	27
7 lentelė. Beckhoff EL4004 analoginių išėjimų modulio techniniai duomenys [37]	28

Paveikslų sąrašas

1.2.1 pav. <i>Beckhoff CX2040 Embedded PC</i>	15
1.3.1 pav. <i>TwinCAT 3</i> vartotojo sąsaja.....	16
1.5.1 pav. <i>TwinCat 3</i> ir <i>MATLAB/SIMULINK</i> integracija [21]	18
1.6.1 pav. <i>Simulink</i> (kairėje) ir <i>TwinCAT</i> (dešinėje) modelių pavyzdžiai [23].....	19
1.6.2 pav. Nuolatinės srovės variklio modelio modeliavimo rezultatų palyginimas [25].....	21
1.7.1 pav. <i>AEG Modicon ET722</i> stendas.	22
1.8.1 pav. <i>CX2100-0014</i> maitinimo šaltinio modulis [28].....	24
1.8.2 pav. <i>PhoenixContact Power UNO</i> maitinimo šaltinis [30]	25
1.8.3 pav. <i>Beckhoff EL2809</i> diskretinių išėjimų modulis [32].....	26
1.8.4 pav. <i>Beckhoff EL3104</i> analoginių įėjimų modulis [34].....	27
1.8.5 pav. <i>Beckhoff EL4004</i> analoginių išėjimų modulis [36]	28
2.1.1 pav. <i>Beckhoff</i> valdiklio ir reaktoriaus stendo sistema	29
2.2.1 pav. <i>WDK</i> Instaliavimo metu <i>Build Enviroments</i> pasirinkimas.....	30
2.2.2 pav. <i>WDK</i> diegimo metu saugojimo vietos pasirinkimas	30
2.2.3 pav. <i>System variables</i> nustatymas	31
2.2.4 pav. Sertifikato sukūrimas.....	31
2.2.5 pav. Sukuriamas sertifikato <i>System variable</i>	32
2.2.6 pav. Suaktyvuotas <i>Test Mode</i> režimas	32
2.2.7 pav. <i>TE1400</i> modulio instaliavimas.....	33
2.2.8 pav. <i>Visual Studio C++ 2015</i> yra <i>Simulink Coder</i> kompiliatorius	33
2.3.1 pav. Tolydinės reaktoriaus sistemos temperatūros reguliavimas.....	34
2.3.2 pav. Tolydinės reaktoriaus sistemos temperatūros reguliavimo rezultatai	34
2.3.3 pav. <i>MATLAB/SIMULINK</i> modelis.....	36
2.3.4 pav. Fiksuoto žingsnio pasirinkimas	37
2.3.5 pav. Užduoties failo pasirinkimas	38
2.3.6 pav. Platformos pasirinkimas	38
2.4.1 pav. Įkeliamas sugeneruotas <i>Simulink</i> CFC kodas.....	39
2.4.2 pav. Įkeltas <i>Simulink</i> modelis į <i>Visual Studio</i> programinę įrangą.....	39
2.4.3 pav. Globalus kintamųjų sąrašas	40
2.4.4 pav. PLV kintamųjų priskyrimas <i>Simulink</i> modelio įėjimams bei išėjimams	40
2.4.5 pav. Kintamojo priskyrimas <i>Simulink</i> modelio įėjimui	41
2.4.6 pav. Sukuriamas programos vykdymo laikas.....	41
2.4.7 pav. Priskiriamas programos vykdymo laikas <i>Simulink</i> programai	42
2.4.8 pav. <i>ADS</i> prisijungimo nustatymas	42
2.5.1 pav. Lygio reguliavimo programa.....	43
3.1.1 pav. Eksperimentiniai rezultatai keičiantis temperatūros nuostatos signalui.....	44
3.1.2 pav. Eksperimentiniai rezultatai keičiantis temperatūros nuostatos signalui, temperatūros nusistovėjimas	45
3.1.3 pav. Eksperimentiniai rezultatai keičiantis temperatūros nuostatos signalui, nusistovėjusi temperatūra	45
3.1.4 pav. Eksperimentiniai rezultatai keičiantis lygio nuostatos signalui	46
3.2.1 pav. <i>Fuzzy</i> įėjimų ir išėjimų kintamųjų funkcijos [40]	47
3.2.2 pav. <i>FAC</i> kubo išpjova [42]	48
3.2.3 pav. Temperatūros reguliavimas su <i>fuzzy</i> regulatoriumi	49

3.2.4 pav. <i>Fuzzy</i> regulatoriaus įėjimo parametrai.....	49
3.2.5 pav. <i>Fuzzy</i> regulatoriaus išėjimo parametrai	50
3.2.6 pav. <i>Fuzzy</i> regulatoriaus taisyklės.....	50
3.2.7 pav. Eksperimentiniai rezultatai su <i>fuzzy</i> regulatoriumi keičiantis temperatūros nuostatos signalui	51
3.2.8 pav. Eksperimentiniai rezultatai su <i>fuzzy</i> regulatoriumi keičiantis lygio nuostatos signalui ..	52

Santrumpų ir terminų sąrašas

Santrumpos:

PLV – programuojamas loginis valdiklis;

Terminai:

CFC (angl. *Continuous Function Chart*) - tai yra grafinė programavimo kalba, pratęsianti standartines *IEC 61131-3* kalbas [1].

NOVRAM (angl. *Non-Volatile Random Access Memory*) - yra specialus atminties komponentas, naudojamas patikimai išsaugoti svarbius duomenis. *NOVRAM* susideda iš dviejų sekcijų: pastovios atminties ir nepastoviosios atminties [2].

DVI-I (angl. *Digital Visual Interface*) - Skaitmeninė sąsaja naudojama prijungti vaizdo šaltiniui, prie rodymo įrenginio, pavyzdžiui kompiuterio monitoriaus. *DVI - I* kabelis, kuris gali perduoti skaitmeninį arba analoginį signalą [3].

Cfast - tai kortelės pagrįstos patikimesnė *SLC* (angl. *Single-Level Cell*) *NAND* atminties technologija, kuri siūlo > 2 milijonus ištvėmės ciklų, o tai prilygsta darbui dešimtmečius be sutrikimo beveik visose pramoninėse *OEM* (angl. *Original Equipment Manufacturer*) srityse [4].

UPS (angl. *Uninterruptible Power Supply*) - prietaisas, leidžiantis kompiuteriui veikti bent trumpam, kai nutraukiamas pagrindinis maitinimo šaltinis [5].

IEC 61131-3 - yra trečioji atviro tarptautinio standarto *IEC 61131*, skirto programuojamiems loginiams valdikliams dalis (iš 10), kurią pirmą kartą 1993 m. gruodžio mėn. paskelbė *IEC* (angl. *International Electrotechnical Commission*) [6].

Ethernet – yra laidinio kompiuterinio tinklo technologijų šeima, paprastai naudojama LAN (angl. *Local Area Network*), MAN (angl. *Metropolitan Area Networks*) ir WAN (angl. *Wide Area Networks*) [7].

Fieldbus - yra pramoninių kompiuterinių tinklas, naudojamas paskirstytam valdymui realiuoju laiku, šeimos pavadinimas [8].

Įvadas

Galingi, daugiafunkciai, kompaktiški ir nebrangūs pramoniniai kompiuteriai visada buvo paklausūs automatinio valdymo srityje. *Beckhoff Automation Company* teikia platų tokių pramoninių kompiuterių asortimentą ir specialią programinės įrangos sistemą, vadinamą *TwinCAT* (*Windows* valdymo ir automatikos technologija), kad galėtų su jais sąveikauti. Problema, su kuria galima susidurti dirbant su *TwinCAT*, yra žinių ir supratimo apie šią programinės įrangos aplinką stoka, nes ji nėra kitur naudojama išskyrus tik *Beckhoff* aparatūrinę įrangą.

Norint suprojektuoti sudėtingas valdymo sistemas ir algoritmus, reikia atlikti komplikuoatą ir kartais sunkiai suprantamą programuojamų loginių valdiklių programavimą. Automatikos sistemų reikalavimai per pastaruosius metus labai išaugo. Šiandieną gamyklų procesai tampa vis didesni ir sudėtingesni dėl to, kad bėgant laikui sumontuota daugiau kompleksinių valdymo komponentų. Nauji komponentai yra būtini norint įdiegti naujas valdymo metodų koncepcijas, kurios yra greitesnės, intelektualesnės bei efektyvesnės. Kompleksiniai ir didelės apimties sudarantys valdymo komponentai turi įtakos ir programinės įrangos kūrimui. Šiandieną sistemų projektavimas įgyvendinamas tekstinės struktūros programavimo kalbomis, tokiomis kaip *C/C++*, arba struktūrizuotu tekstu (ST) [9]. Dėl kylančio įrenginių sudėtingumo, atsiranda tokia problema kaip tekstinio mašininio kodo aiškumas, sudėtingesnis mašinos darbo optimizavimas bei gedimų šalinimas, dėl ko grafinės programavimo kalbos visus šiuos aspektus žymiai palengvina. Taip pat reikia nemažai laiko skiriant susipažinti su skirtingų gamintojų programine įranga. Reikia paminėti, kad *MATLAB/SIMULINK* programinės įrangos paketas yra pasaulyje gerai žinoma ir labai plačiai naudojama aplinka automatikos valdymo modeliams kurti. Šią programinę įrangą studijavo valdymo inžinieriai dar pačioje studijų pradžioje ir ji pramonėje turi didelę reikšmę. *MATLAB/SIMULINK* tapo universaliu įrankiu, kuris šiais laikais leidžia spręsti daugelį automatizavimo ir valdymo optimizavimo užduočių, o grafinės blokinės schemos programavimo principo buvimas jame leidžia kurti sudėtingas sistemas, patogiu ir intuityviu būdu. Galimybė konvertuoti blokinės diagramos vaizdavimą į *CFC* kodą yra galinga savybė, leidžianti vartotojo kodą įdiegti bet kurioje tikslinėje sistemoje ir palengvinti daugelio valdymo užduočių sprendimą. Kadangi nėra aiškių ir išsamių instrukcijų, kaip naudoti *MATLAB/SIMULINK* programinę įrangą, norint valdyti *Beckhoff* aparatūrinę įrangą, šis darbas bus skirtas šiai problemai išspręsti.

Pirmiausia, siekiama parodyti, kaip *Beckhoff Automation Company* pagamintas pramoninis kompiuteris gali veikti naudodamas programos kodą, sukurtą kaip funkcijų blokų schemą *MATLAB/SIMULINK* programinėje įrangoje. Pateikiamos instrukcijos su išsamiais paaiškinimais kaip gali būti naudojamas *MATLAB/SIMULINK* ir *TwinCAT 3* programinės įrangos integravimas. Tada siekiama sukurti reaktoriaus stendo temperatūros ir lygio valdymo algoritmą ir įgyvendinti šį algoritmą naudojant *Beckhoff Embedded PC* pramoninį kompiuterį, kuris veikia pagal *MATLAB/SIMULINK* blokų schemos kodą. Sujungus aparatūrinę įrangą ir įdiegus valdymo sistemą, atliekami eksperimentiniai bandymai, siekiant iširti valdymo sistemos veikimą ir reakcijas į trikdžius.

Tyrimo tikslas – *MATLAB/SIMULINK* aplinkoje sukurti reaktoriaus temperatūros ir skysčio lygio reguliatorių modelį ir perkelti į *Beckhoff* programuojamą loginį valdiklį.

Darbo uždaviniai:

- išanalizuoti pasirinkto programuojamo loginio valdiklio galimybes dirbti kartu su *MATLAB/SIMULINK* programine įranga;
- parinkti tinkamą reguliatorių imitaciniam reaktoriui valdyti;
- atlikti reaktoriaus temperatūros bei lygio valdymą pasinaudojus *MATLAB/SIMULINK* platforma ir perkelti į pramoninį valdiklį;
- išanalizuoti proceso veikimą perkėlus į programuojamą loginį valdiklį.

1. Valdymo sistemos analizė

1.1. Beckhoff kompanija

Beckhoff Automation GmbH yra Vokietijos automatikos įrangos sistemų bei atskirų komponentų gamintojas.

Bendrovė daugiausia dėmesio skiria su kompiuteriais suderinamai valdymo įrangai, pramoniniams kompiuteriams, integruotiesiems kompiuteriams, įvesties/išvesties moduliams, servo variklių technologijoms ir automatikos programinei įrangai. 2003 m. *Beckhoff* pristatė *EtherCAT* technologiją, skirtą valdyti sistemas realiuoju laiku naudojant *Ethernet* tinklą.

Beckhoff yra daugelio pagrindinių šiuolaikinės automatikos sričių kūrimo pradininkas. Pagrindinės produktų taikymo sritys yra energetika, naftos procesų valdymas, dujų ir chemijos pramonė, transporto sistemos, katilinės, vėjo energetika, gamyklų, pastatų automatizavimas.

Beckhoff gamina keturis pagrindinius produktų tipus: pramoninius kompiuterius, įvesties/išvesties modulius, servo variklius ir *TwinCAT* programinę įrangą [10].

Pramoniniai kompiuteriai

Pagal pramoninių kompiuterių gamybos apimtį bendrovė yra antra pagal dydį Vokietijoje.

Pramoninių kompiuterių asortimentą sudaro:

- integruotieji pramoniniai kompiuteriai ir serveriai;
- vartotojo sąsajos;
- kompaktiški moduliniai kompiuteriai su tiesiogine įvesties/išvesties jungtimi;
- pramoninių kompiuterių pagrindinės plokštės.

Integruotų kompiuterių serijos skiriasi viena nuo kitos procesoriaus našumu - pradedant paprastų procesorių, tokių kaip *ARM* (angl. *Advanced RISC Machines*), iki galingų *Intel Core* procesorių. Jie skirti įvairaus sudėtingumo automatikos užduotims spręsti, pradedant pastato inžinerinių sistemų valdymu ir namų automatizavimu, baigiant dideles technologines linijos valdymu, kurių įvesties/išvesties signalų skaičius matuojamas tūkstančiais. Skirtingo dizaino pramoniniuose kompiuteriuose su įvairiomis papildomomis funkcijomis yra sumontuoti aukšto našumo komponentai, pagrįsti atvirais standartais [11].

Įvesties/išvesties sistemos

Viena iš pagrindinių *Beckhoff* sričių yra įvesties/išvesties sistemų kūrimas bei gamyba.

Beckhoff gamina keturių tipų įvesties/išvesties sistemas:

- Klasikiniai įvesties/išvesties moduliai (galimi įvairūs signalų tipai);
- *EtherCAT I/O* (naujos kartos *Ethernet* pagrindu veikianti *fieldbus* sistema);
- *Fieldbus Box* (sistema sunkioms darbo sąlygoms - atitinka *IP67* apsaugos klasę);
- *Lightbus* (optinė sistema, atspari stipriems elektromagnetiniams ir pramoninės aplinkos trikdžiams). [12]

1.2. CX serijos pramoniniai kompiuteriai

Beckhoff Embedded PC yra pilnaverčiai kompiuteriai, turintys pramoninių PLV funkcionalumą. Šiame darbe nagrinėjama *Beckhoff Embedded PC CX* serija.

Embedded PC CX serija sujungia asmeninio kompiuterio technologiją ir modulinę įvesties/išvesties sistemą. Šie kompiuteriai paprastai montuojami ant *DIN* (vok. *Deutsche Industrie Norm*) bėgelio valdymo spintose. Sujungus pramoninių asmeninių kompiuterių ir aparatinės įrangos PLV savybes, *CX* prietaisų serija tinkama vykdyti įvairiausių poreikių užduotims. Modulinė įvesties/išvesties sąsajų sistema leidžia sukurti tinkamą valdymo užduotį PLV konfigūracijai, pakeičiant įvesties/išvesties blokų skaičių ir įdiegiant tik būtinus valdymo sistemos komponentus.

CX šeima aprėpia visą *Beckhoff* valdymo technologijos spektrą tiek kainos, tiek našumo požiūriu. Šis gaminių asortimentas skirtas užduotims, kurioms reikalingos pramoninių kompiuterių charakteristikos ir skaičiavimo pajėgumai, tačiau kurių biudžetas neviršija pilnaverčių pramoninių kompiuterių [13].

CX serijos asmeninių kompiuterių našumo klasėse yra keli pagrindiniai modeliai su skirtingais procesoriais, kurie pritaikomi atitinkamai valdymo užduočiai. Šiame tiriamajame darbe naudojamas *PC CX2040* su keturių branduolių *Intel Core i7* procesoriumi, kurio dažnis yra *2,1 GHz*.

Be procesorių tipo, *CX* serijos kompiuterių šeima taip pat skiriasi sistemos sąsajomis ir maitinimo blokais. Šie kompiuteriai palaiko ryšį per skirtingas įvesties/išvesties sistemas įvairiomis sąsajomis, tokiomis kaip magistralės terminalai ir *EtherCAT*. *CX* valdiklio konfigūracijos pasirinkimas paprastai grindžiamas automatikos algoritmo sudėtingumu. Pereiti iš vieno *CX* šeimos kompiuterio į kitą, pavyzdžiui, esant didesnio pajėgumo poreikiui, įmanoma net ir vėlesniame valdymo sistemos diegimo etape ir net nereikia keisti sistemos valdymo algoritmo.

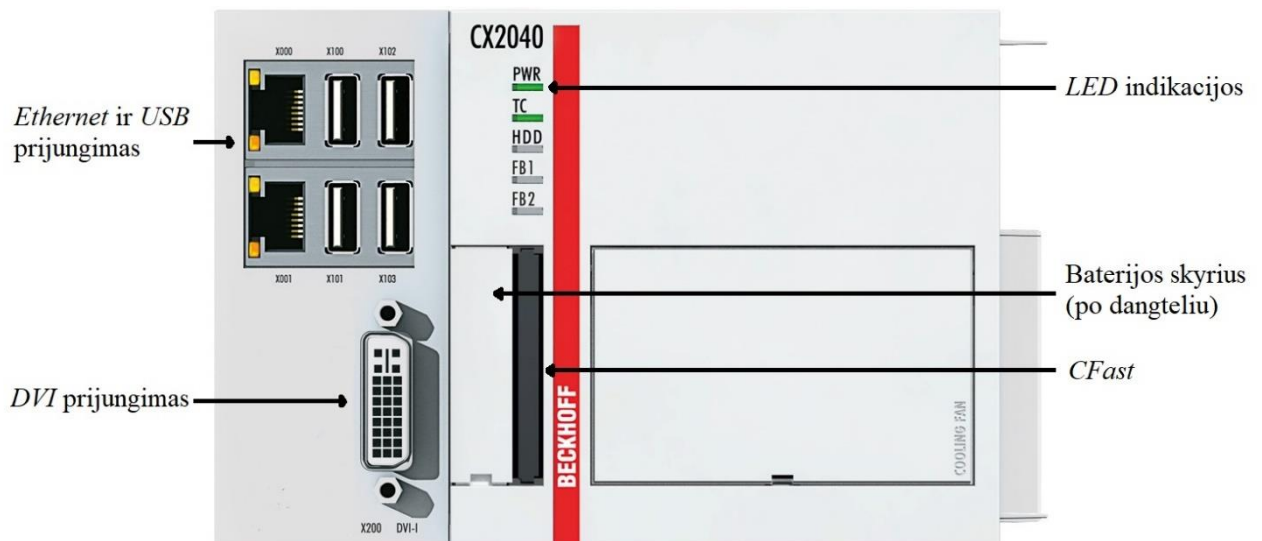
CX serijos pramoninių kompiuterių konfigūravimas ir valdymas atliekamas naudojant *TwinCAT* programinę įrangą, naudojančią tarptautinį standartą *IEC 61131-3*, kuris apibūdina pramoninių PLV programavimo kalbas. *CX* serijos valdikliai vienu metu yra *Windows* pagrindu veikiantys PLV ir asmeniniai kompiuteriai. Dėl iš anksto įdiegtos *OS Windows*, vartotojo užduotys, parašytos aukšto lygio programavimo kalbomis, *TwinCAT* aplinkoje apdorojamos realiuoju laiku.

CX2040 pramoninis kompiuteris

Šiame darbe *Beckhoff CX2040 Embedded PC* naudojamas kaip *RTDAQ* (angl. *Real-Time Data Acquisition Software*) kompiuterio ir valdymo algoritmo diegimo aparatūra.

CX2040 (1.2.1 pav.) Turi *2,1 GHz Intel Core i7* keturių branduolių procesorių.

Pagrindinėje *CX2040* konfigūracijoje yra *CFast* atminties kortelė, dvi nepriklausomos *RJ45* sąsajos, keturios *USB 2.0* sąsajos ir *DVI-I* sąsaja. Centrinis procesorius turi 128 kB nuolatinę *NOVRAM* duomenų atmintį situacijoms, kai nenaudojamas *UPS*. Operacinė sistema yra *Microsoft Windows Embedded 7*. *TwinCAT* automatikos programinė įranga paverčia *CX2040* sistemą į PLV, kurią galima valdyti su vizualizacija arba be jos.



1.2.1 pav. Beckhoff CX2040 Embedded PC

CX2040 procesoriaus techniniai duomenys aprašomi lentelėje 1.

1 lentelė. CX2040 procesoriaus techniniai duomenys [14]

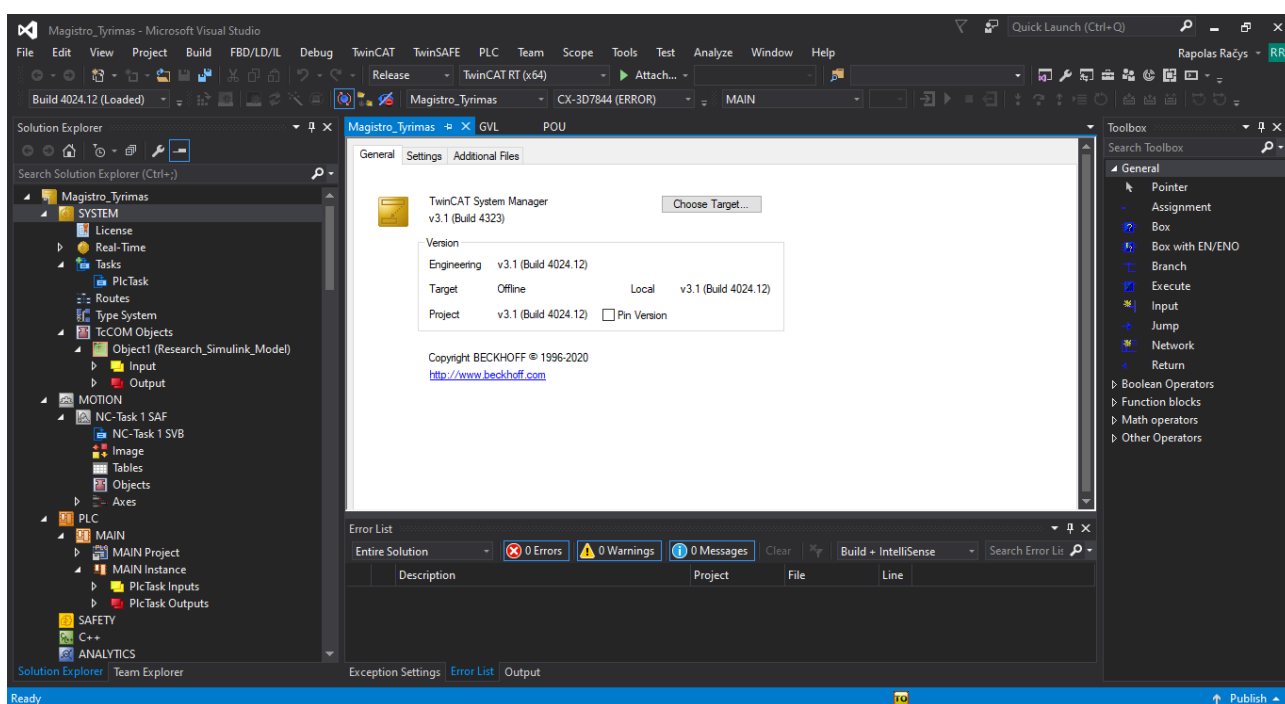
Parametras	Charakteristikos
Procesorius	Intel Core i7 2715QE 2.1 GHz, 4 branduoliai
Atmintuko ir vidinė atmintis	20 Gb arba 40 Gb CFast atminties kortelė (priklausomai nuo operacinės sistemos), gali būti išplėsta. 4 Gb DDR3 RAM (be galimybės praplėsti)
Pastovi atmintis	128 Kb NOVRAM integruota
Sąsajos	2 x RJ45 10/100/1000 Mbit/s, 1 x DVI-I, 4 x USB 2.0
Vėsinimas	Integruotas ventiliatorius su guoliais ir greičio stebėseną
LED diagnostikos indikacijos	1 x veikimo, 1 x TC būsenos, 1 x atmintuko priėjimas, 2 x fieldbus būseną
Laikrodis	Baterija palaikomas laikrodis laikui ir datai (Keičiama baterija)
Operacinė sistema	Windows Embedded Compact 7 (TwinCAT 3 palaiko tik vieną branduolį), Windows Embedded Standart 7 P, Windows 10 IoT Enterprise 2016 LTSB, TwinCAT/BSD
Valdymo programinė įranga	TwinCAT 2 runtime, TwinCAT 3 runtime (XAR)
I/O prijungimas	Per maitinimo šaltinio modulį (E-bus arba K-bus, automatinis atpažinimas)
Maitinimo šaltinis	24 V DC (-15%/+20%)
Maksimalus galios suvartojimas	32 W
Matmenys	144 mm x 99 mm x 91 mm
Svoris	Apie 1230 g
Darbinė/saugyklos temperatūra	-25...+60°C/-40...+85°C
Santykinė drėgmė	95 %, be kondensacijos
Vibracijos/smūgio atsparumas	Atitinka EN 60068-2-6/EN 60068-2-27
EMC atsparumas/spinduliavimas	Atitinka EN 61000-6-2/EN 61000-6-4
Atsparumo klasė	IP 20
TC 3 eksploatacinių savybių klasė	Aukšta (70)

1.3. TwinCAT 3

Beckhoff sukūrė TwinCAT programinę įrangą programuoti PLV, atlikti įvairaus sudėtingumo pozicionavimą, konfigūruoti I/O įrenginius bei kurti vizualizavimo sąsajas.

Remiantis *Beckhoff Automation TwinCAT* instrukcijomis, *TwinCAT 3* pagrindiniai privalumai yra šie:

- viena programavimo ir konfigūravimo programinė įranga;
- *Microsoft Visual Studio* integracija;
- laisvė pasirinkti programavimo kalbas;
- suderinamumas su plačiai paplitusiu *CODESYS* (PLV programų kūrimo aplinka pagal tarptautinį pramonės standartą *IEC 61131-3*);
- *C/C ++* naudojimas kaip programavimo kalba programoms realiuoju laiku;
- *MATLAB/SIMULINK* integracija;
- lanksti vykdymo trukmės aplinka;
- 64 bitų sistemų palaikymas;



1.3.1 pav. *TwinCAT 3* vartotojo sąsaja

TwinCAT 3 suteikia galimybę programuoti *C/C ++* kalbomis. *TwinCAT 3* integruoja save į esamą *Microsoft Visual Studio*, kad realiojo laiko užduotims naudoti programavimo kalbas *C* arba *C ++* su įvesties/išvesties konfigūracijomis bei kartu su *IEC 61131-3* standarto kalbomis. Sukurti objektai gali keistis duomenimis nepriklausomai nuo kalbos, kuria jie buvo parašyti. Tokiu būdu automatizavimo įrenginiams sukonfigūruoti, nustatyti, programuoti ir diagnozuoti reikia tik vienos programinės įrangos [15].

1.4. *MATLAB/SIMULINK*

Galimos priemonės atlikti uždaro ciklo valdymo procesui yra *Labview* [16] ir *Simulink* [17]. *Labview* sutelkia dėmesį į realaus laiko išorinės aparatūros valdymą. Tam, kad valdyti išorines sistemas, *Labview* programinei įrangai reikalinga speciali aparatinė įranga. Taigi *Labview* modeliavimo funkcija nėra visiškai išplėtotą ir nepalaikoma, kaip kituose modeliavimo įrankiuose. Be to, *Labview* neturi sąsajos su jokia PLV ir negali atlikti programos transformavimo į PLV aplinką grafiniu ar tekstiniu mašininu pavidalu. Taigi, pagrindinis *Labview* taikymo sritys yra duomenų analizė ir signalų matavimai [18].

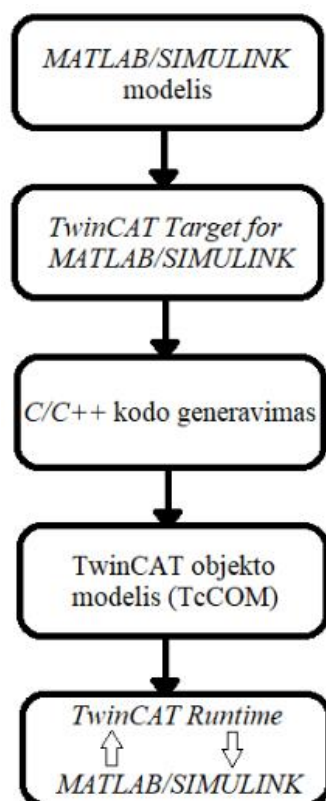
Tuo tarpu modeliavimo įrankis *MATLAB/SIMULINK R2018a* yra vienas iš plačiai paplitusių ir labiausiai palaikomas modeliavimo įrankis automatikos srityje [19]. *Simulink* taip pat dažnai mėgstamas procesų imitavimo įrankis, nes jis gali modeliuoti modelius diskretiškai. Diskretus modeliavimo režimas yra svarbus, nes PLV aplinka veikia cikliška. Taigi, rezultatų palyginimas tarp teorinio modeliavimo ir PLV aplinkos kodo galimas tik tuo atveju, jeigu imitavimo įrankis gali atlikti diskretinį teorinį modeliavimą. *Simulink* modelis gali būti imituojamas naudojant skirtingus diskretizavimo nustatymus ir ciklo laikus, uždaro ciklo valdymo modelio diskretizavimo laikas modeliavimo įrankyje yra svarbus kai reikia nustatyti diskretizavimo laiką PLV aplinkoje. Pasitelkus intuityvų ir palyginus lengvą būdą, *Simulink* gali palengvinti valdymo sistemų kūrimo ir diegimo laiką bei sumažinti išlaidas jau proceso diegimo stadijoje. Priežastis kodėl tai gali palengvinti diegimo procesą tai grafinis programavimas pavadintas blokų schema ir galimybė naudoti komercinį PLV kodavimo įrenginį, kuris automatiškai sugeneruoja grafinį mašininį kodą PLV aplinkoje. Jeigu *Simulink* ir PLV modeliai yra identiški ir rodo tas pačias proceso charakteristikas, modelių perprojektavimas gali būti atliekamas imitavimo įrankyje nenaudojant PLV aplinkos. Sukurtas *Simulink* modelis, kuris yra išsaugotas modelio faile, turi specialią tekstinę struktūrą. Šį formatą sukūrė *MathWorks*, šiame formate yra informaciją apie sukurto modelio blokus, jungtis, parametrus, įėjimus, išėjimus ir modeliavimo nustatymus. Vėliau visa svarbi informacija *MDL* (angl. *Model Definition Language*) faile filtruoja kodų generatorius ir transformuoja į suderinamą failą PLV aplinkoje.

Apibendrinant galima sakyti, kad *Labview* yra gera priemonė automobilių bei aeronautikos srityse, projektavimams ir matavimams, bet ne automatikos PLV aplinkoje. Naudojantis *Simulink* įrankiu yra daugiau galimybių kurti ar imituoti modelius skirtingoms PLV aplinkoms. *Simulink* taip pat teikia daugybę papildomų įrankių, skirtų modeliavimui ir projektavimui bei transformuoti į PLV aplinkas. Tai yra priežastis, kodėl *Simulink* šiame darbe yra naudojamas kaip modeliavimo įrankis.

1.5. *TwinCAT 3* sąsaja ir *Target for MATLAB/SIMULINK*

TwinCAT 3 ne tik leidžia programuoti klasikinėmis *IEC 61131-3* programuojamų loginių valdiklių kalbomis, bet ir leidžia tai daryti su aukštesnio lygio kalbomis tokiomis kaip *C/C++*, *CFC* bei *MATLAB/SIMULINK*.

TE 1400 MATLAB/SIMULINK integravimas leidžia vykdyti *TwinCAT 3* blokus, kurie buvo sukurti kaip modelis *MATLAB/SIMULINK* modeliavimo aplinkoje. Pasirinktas sąsajos tipas parodo parametrus ir kintamuosius *TwinCAT 3* grafinėje sąsajoje ir leidžia juos peržiūrėti bei keisti realiu valdymo laiku. *MATLAB/SIMULINK* aplinkos *Simulink Coder* (anksčiau vadinamas *Real-Time Workshop*) yra kodų generatorius, kuris gali sugeneruoti atitinkamą *C/C++* kodą iš *MATLAB/SIMULINK* modelio. *TwinCAT TE1400 Target for MATLAB/SIMULINK* veikia remiantis šiuo kodų generatoriumi. Jei kodų generatorius yra tinkamai sukonfigūruotas naudojant *TwinCAT Target*, sukuriama *TwinCAT* objektų modelis (*TcCOM*) su *MATLAB/SIMULINK* modelio įvesties ir išvesties elgsena [20]. Kadangi *Simulink coder* reikia papildinio *C++* kodui sugeneruoti, šiame darbe bus naudojama *Visual Studio* kompiliatorius vietoje *Matlab* siūlomo *MinGW-w64 C/C++* papildinio. Tai buvo pasirinkta dėl to, kad *TwinCAT 3* yra integruotas kaip papildinys *Visual Studio* platformoje, o ne instaliuota atskira programa. Norint, sumažinti *Simulink* modelio konversijas, pasirinkta ir naudojama šiame tyrime *Visual Studio 2015* programa. *TwinCAT 3* ir *MATLAB/SIMULINK* integravimo principas pavaizduotas 1.5.1 paveikslėlyje.



1.5.1 pav. TwinCat 3 ir MATLAB/SIMULINK integracija [21]

1.6. Įvairių PLV gamintojų programinės įrangos palyginimas

Šiame skyriuje pateikiama medžiaga iš [22] šaltinio kaip skiriasi transformuoto *Simulink* modelio į *CFC* kalbą struktūra nagrinėjamosiose PLV aplinkose. Šioje analizėje nagrinėjami naudoto kodų generatoriaus, kuris modeliavimo modelį paverčia *CFC* kalba kai kurios savybės, pavyzdžiui, modelio blokų vykdymo tvarka, modelių grafinė struktūra bei laiko parametrai.

Pirma, bus nagrinėjamos keturios skirtingos sistemos, tai trys programuojamų loginių valdiklių programinės įrangos, *TwinCAT 3*, *PCS7* ir *PC Worx*, kurie palaiko *CFC* kalbą (žr. 2 lentelė) bei modeliavimo įrankis *Simulink*.

2 lentelė. PLV valdiklių programavimo kalbų palaikymas

	C/C++ arba Java	IEC 61131-3	CFC
Beckhoff Automation (TwinCAT)	C/C++ (nuo V3)	X	X
Siemens (PCS7)		X	X
Phoenix Contact (PC Worx)		X	X
Bachmann (M-PLC)	Java	X	
B&R (Automation Studio)	C/C++	X	

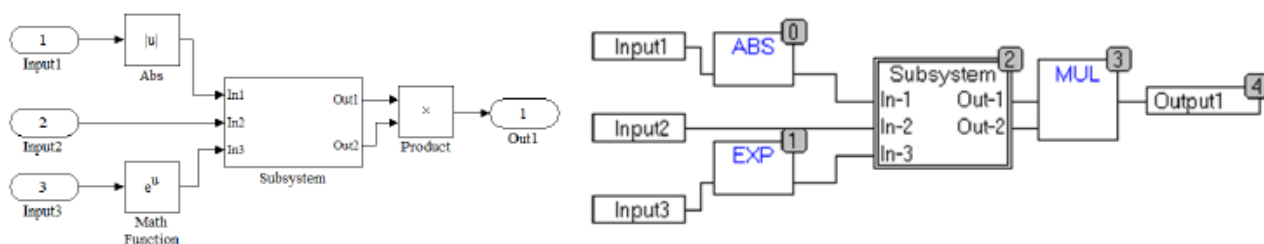
Taigi modelio blokų vykdymo tvarka yra labai svarbu norint palyginti programos rezultatus skirtingose sistemose. Vykdyto tvarka modelyje pateikia informaciją, kuris modelio blokas buvo įvykdytas pirmas. Ši tvarka yra labai svarbi modeliams su grįžtamoju ryšiu. Tik jei vykdymo tvarka sistemose yra lygiavertė, galima realizuoti sąžiningą palyginimą. Skirtingi modeliavimo vykdymo nurodymai sistemose iššaukia klaidas, kurios nusako skirtingą eiliškumo vykdymą nagrinėjamose programinėse įrangose.

Modeliavimo įrankyje *Simulink* eiliškumas vykdomas nepakeičiamai. Modeliavimo eiliškumas *Simulink* programinėje įrangoje nustatomas automatiškai prieš tai, kai pradamas modelio vykdymas. Paprastai blokai yra apdorojami modelio signalo srauto tvarka.

PLV aplinkos *TwinCAT* ir *PCS7 CFC* kalboje turi tą patį vykdymo nurodymą kaip ir *Simulink* įrankis. Abiejuose šiose PLV aplinkose *CFC* kodo vykdymas vyksta automatinio būdu, dažniausiai tai pagal signalo srautą modelio aplinkoje. Be to, abiejose sistemose galima pakeisti blokų vykdymo tvarką rankiniu būdu į norimą tvarką.

PC Worx CFC modelyje vykdymo nurodymas yra taip pat nustatomas automatiškai. Čia blokų vykdymas yra iš kairės į dešinę ir iš viršaus į apačią. *PC Worx* trūkumas yra tai, kad vykdymo nurodymo negalima pakeisti rankiniu būdu. Taigi visada yra ta pati *CFC* blokų vykdymo tvarka. Pasekmė yra ta, kad jei vykdymo tvarka skiriasi nuo modeliavimo, rezultatai daugeliu atvejų skiriasi, o galimybės korekcijoms nėra.

Norint įvertinti trijų *CFC* modelių PLV aplinkoje ir imitavimo įrankio *Simulink* grafines struktūras, sukuriama pavyzdiniai modeliai. Šiame palyginime nebuvo jokių blokų išdėstymo tvarkos ir padėties skirtumų, visuose keturiuose modeliuose. Vienintelis skirtumas tai, kad *PC Worx* ir *PCS7* negalima naudoti įėjimų ir išėjimų blokų iš *Simulink* įrankio konversijoje, taigi šiose dviejose PLV aplinkose įėjimų/išėjimų blokai yra vidiniai kintamieji. Trečioji sistema *TwinCAT* palaiko įėjimų/išėjimų blokų konversiją, dėl to *TwinCAT CFC* modelis turi lygiai tokia pačią struktūrą kaip ir modelis *Simulink* aplinkoje. *Simulink* ir *TwinCAT* modelių pavyzdžiai pateikiami 1.6.1 paveikslėlyje.



1.6.1 pav. *Simulink* (kairėje) ir *TwinCAT* (dešinėje) modelių pavyzdžiai [23]

Kita problema iškyla, jei transformuojamas imitacinis modelis turi posistemas, nes *PC Worx* nepalaiko posistemų. Kadangi kodo generatorius atlieka vienas prie vieno konversiją iš *Simulink* modelio į PLV aplinką. Taigi, modeliai su posistemėmis, galimi tik transformacijai į *TwinCAT* ir *PCS7* programinėje įrangoje. Tačiau šiame pavyzdyje, modelių struktūra be posistemų ir įėjimų bei išėjimų blokų išlieka tokia pati visuose trijuose *CFC* modeliuose PLV aplinkose, tik atsižvelgiama į PLV aplinkose palaikomą kodo į *CFC* konversiją. *TwinCAT* palaiko identišką modelio struktūrą kaip ir *Simulink* įrankyje, o *CFC* modeliai, esantys *PCS7* ir *PC Worx*, turi nedidelius skirtumus, palyginti su modeliu *Simulink* aplinkoje, atsižvelgiama į įėjimų/išėjimų kintamuosius bei posistemas.

Galiausiai atliekama diskretizacijos įtakos analizė bei tiriami modeliavimo metodai modeliams PLV aplinkoje. Nuo diskretizavimo laiko priklauso gautų verčių apskaičiavimas ir sujungimas kiekvieno ciklo metu. Sprendimas dėl teisingo metodo ir laiko parinkimo yra labai svarbus vėlesniems šio skyriaus vertinimams. *Simulink* palaiko penkis skirtingus diskretizavimo metodus diskretiniams modeliavimams. Šie penki metodai yra *ZOH* (angl. *zero-order hold*), *FOH* (angl. *first-order hold*), impulsų nekintamumo diskretizavimas, bilineari transformacija ir nulinių polių metodas [24].

TwinCAT, *PCS7* ir *PC Worx* palaiko iš šių penkių galimų metodų tik bilinearią transformaciją, t.y. visoms *Simulink* modeliavimo sistemoms turi būti naudojamas bilinearios transformacijos metodas, siekiant sėkmingai palyginti gautus rezultatus.

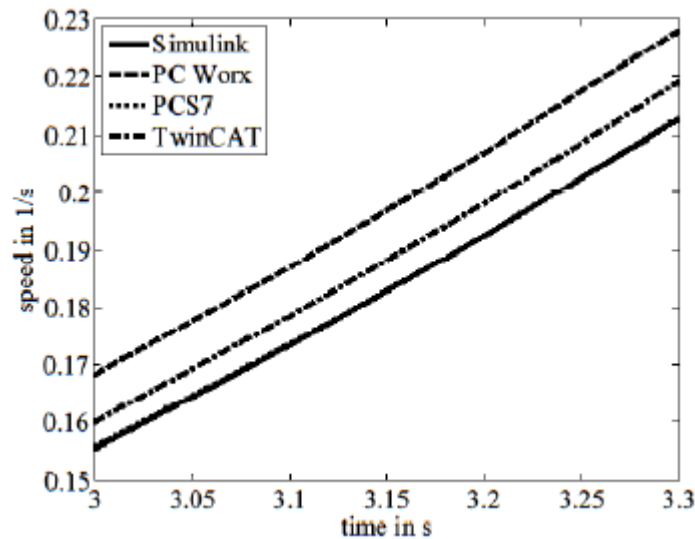
Visi suprojektuoti modeliai turi tuos pačius ciklo modeliavimo laikus. Vadinasi, PLV aplinkoje reikalinga tik viena užduotis (angl. *task*). Svarbu, kad *Simulink* modelyje būtų parinktas diskretizavimo laikas, kurį būtų galima suderinti su laiku PLV aplinkoje, nes skirtingi PLV procesoriai gali realizuoti atitinkamai skirtingus ciklo vykdymo laikus.

Pavyzdys

Vertinimui naudojamas paprastas *Simulink* variklio sukimosi greičio valdymo modelis. Modelis susideda iš proporcinio - integralinio (PI) regulatoriaus bei nuolatinės srovės variklio. Modelio struktūra tai PI reguliatorius, kuris sujungtas su nuolatinės srovės variklio perdavimo funkcija, valdymo sistemos išėjimą su įėjimu jungia neigiamas grįžtamasis ryšys, kuris siunčia paklaidą į reguliatorių. Vertinimo procesas vyko taip: pirma, aprašytas modelis buvo paleistas modeliuoti įrankyje *Simulink*, kur buvo surinkta informacija grafiškai apie variklio sukimosi greičio kitimą laike. Tada panaudojus kodo generatorių, *MDL* failas buvo automatiškai transformuotas į failus, kurie tinkami *TwinCAT*, *PCS7* ir *PC Worx* programinėms įrangoms. Šiuose failuose yra suprojektuoto *Simulink CFC* modelio tekstinis kodo modelis, įskaitant parametrinius funkcinius blokus bei papildomi duomenys apie diskretizavimo laikus. Po sėkmingo failų importavimo į atitinkamas PLV aplinkas, sugeneruotas *CFC* modelis buvo paleistas kiekvienoje sistemoje. Variklio sukimosi greičio duomenys surinkti grafiškai imituojant kiekvieną aplinką atskirai. Toliau atliktas visų keturių sistemų palyginimas bei analizė.

Rezultatas

Prieš imituojant suprojektuotą greičio valdymą aplinkose *Simulink*, *TwinCAT*, *PCS7* ir *PC Worx* reikalinga nustatyti kelis nustatymus, pavyzdžiui, PLV ciklo trukmė ir užduoties signalas. Tai yra svarbu, nes tik tada, kai nustatymai visose keturiose aplinkose yra lygūs, juos bus galima palyginti. Ciklo laikas visose keturiose aplinkose buvo nustatytas *100 ms* ir nustatytas visas modeliavimo laikas iki 5 sekundžių. Užduoties signalu buvo pasirinkta šuolinė funkcija. Pasirinktas šuolis yra vienetinis su vienos sekundės užlaikymu. Likusias keturias sekundes modeliavimo metu žingsnio vertė buvo vienetas. Po reikalingų parametrų nustatymo, buvo atliktas modeliavimas.



1.6.2 pav. Nuolatinės srovės variklio modelio modeliavimo rezultatų palyginimas [25]

1.6.2 paveiksle parodytas greičio kitimas 300 ms intervale visose keturiose aplinkose. Pagrindinis palyginimas tai yra su *Simulink* modeliavimo rezultatais. Tam, kad sukurti kokybės modeliavimo rezultatų reitingą PLV aplinkose turi būti atlikta analizė. Analizė buvo atlikta dviem etapais. Pirmiausia buvo apskaičiuotas nuokrypis tarp *Simulink* ir trijų PLV aplinkų. Tai buvo padaryta naudojant *RSME* (angl. *Root Mean Square Error*) metodą [26].

RSME apskaičiuojamas pagal (1) formulę:

$$RSME = \sqrt{\sum_{i=1}^N \frac{(y(i) - \hat{y}(i))^2}{N}}; \quad (1)$$

čia N – paimtų duomenų skaičius; $y(i)$ - i -toji vertė; $\hat{y}(i)$ - i -toji numatoma vertė.

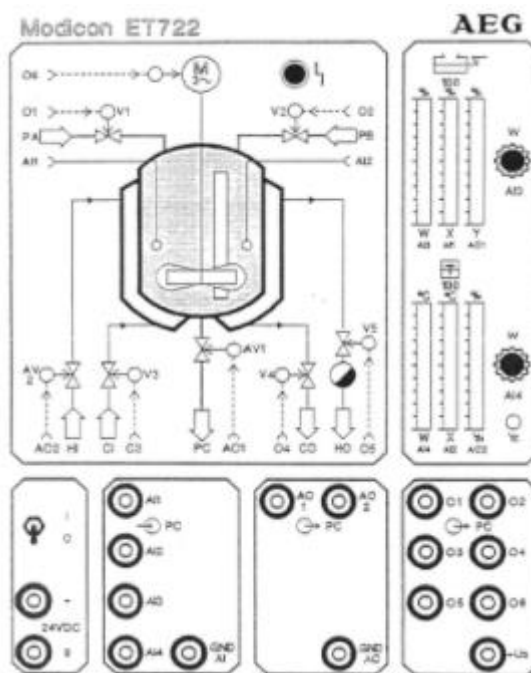
Šis metodas apskaičiuoja paklaidą procentais tarp *Simulink* ir trijų PLV aplinkos rezultatų. Tai yra pagrindinis kokybės reitingas, kuris rodo visų trijų rezultatų dispersiją PLV aplinkose, imituojant rezultatus, kurie buvo gauti *Simulink* aplinkoje. Apskaičiuotas sukurto *CFC* kodo *RSME* visų trijų PLV aplinkų buvo mažesnis nei 1,5 proc. *PC Worx* rezultatų variacijos yra susiję su skirtingais blokų vykdymo nurodymais *CFC* kode lyginant su *Simulink* modeliu, todėl atsiranda didesnė nuokrypio vertė. Finale atsižvelgiant į apskaičiuotus tris *RSME* rodiklius galima matyti, kad *PCS7* turi mažiausią modelio rezultatų nuokrypį lyginant su *Simulink* programine įranga.

Palyginimo apibendrinimas

Modeliavimo rezultatų įvertinimas ir transformuotų *CFC* modelių struktūros analizė trijose PLV aplinkose leidžia teigti, kuri PLV aplinka turi geriausias paprasto *Simulink* modelio vykdymo charakteristikas. *TwinCAT* *CFC* modelis yra lygus *Simulink* modelio struktūrai. Todėl buvo įmanomas visų blokų tiesioginis susiejimas variklio sukimosi greičio valdymo modelyje iš *Simulink* į *TwinCAT* aplinką. Lyginant *CFC* modelių rezultatus tarp *Simulink* ir *PCWorx* bei *PCS7*, nukrypimai nuo *Simulink* rezultatų yra nedideli. Modeliavimas rodo, kad *PCS7* rezultatai turi mažiausią nuokrypį nuo *Simulink* modeliavimo rezultatų. Taigi *PCS7* geriausiai atitinka modeliavimo rezultatus. Tačiau sugeneruotiems *CFC* kodams visi trys apskaičiuoti *RSME* yra mažesni nei 1,5 proc.

1.7. Reaktoriaus stendas

Tyrimo metu naudojamas *AEG Modicon ET722* reaktoriaus stendas, kuriame yra atliekamas cheminio reaktoriaus modeliavimas, tai yra tam tikro cheminio skysčio lygio ir temperatūros reguliavimas.



1.7.1 pav. AEG Modicon ET722 stendas.

1.7.1. Reaktoriaus proceso aprašymas

Reaktorius – dvigubas sienelės turintis indas, kuris gali būti šildomas arba šaldomas specialiu skysčiu. Indo sudėtį sudaro jutikliai ir maišyklė:

- A11 – faktinė lygio vertė. Stende atvaizduojamas lygis dviem indikatoriais: vienas reaktoriuje, kitas valdymo pulte.
- A12 – faktinė temperatūros vertė. Temperatūrai atvaizduoti naudojamas indikatorius valdymo pulte.
- maišyklė. Maišyklės veikimą rodo bėgančioji šviesa indo viduje. Maišyklės variklis M paleidžiamas valdiklio išėjimo signalu Q6.

Reaktoriaus indo pripildymas. Reaktorių galima pripildyti dviejų tipų produktais:

- įjungus ventilių V1 (O1), produktu A (PA);
- įjungus ventilių V2 (V2), produktu B (PB);

Virš ventilio esanti žalia lemputė rodo kada ventilis yra atidarytas, bėgančioji oranžinė šviesa vamzdyje rodo skysčio tekėjimą. Mygtukas esantis virš ventilio V2 yra trikdis. Paspaudus šį mygtuką, atidaromas ventilis V2 nepriklausomai nuo duodamo valdiklio išėjimo O2.

Skysčio iš reaktoriaus išleidimas. Produktas C iš reaktoriaus gali būti išleidžiamas atidarius analoginį ventilių AV1, kuris yra reguliuojamas valdiklio analoginiu išėjimu AO1. Ventilio AV1 atsidarymo dydis vaizduojamas valdymo pulte esančia stulpeline indikacija. Pradarius ventilių AV1, virš jo užsidega žalia indikacija ir bėgančioji oranžinė šviesa vaizduoja skysčio tekėjimą.

Šildymas. Indo marškiniuose pratekančio šildymo skysčio debitą valdo analoginis ventilis AV2. Ventilio AV2 atsidarymo dydis vaizduojamas valdymo pulte esančia stulpeline indikacija. Pradarius

ventilį AV1, virš jo užsidega žalia indikacija, bet šildymo skystis pradeda tekėti tik tuo atveju jeigu yra atsidaręs ventilis V5. Skysčiui tekant, vamzdžiuose tarp skysčio įtekėjimo ir ištekėjimo taško dega raudona bėgančioji šviesa. Raudona šviesa reaktoriaus korpuse vaizduoja įjungtą šildymą.

Aušinimas. Diskretinis ventilis V3 atidaro aušinimo skysčio tiekimo vamzdį. Valdymo pulte esanti žalia indikacija Y_c vaizduoja ventilio V3 būseną. Šaldymo skystis pradeda tekėti tik tuo atveju jeigu yra atsidaręs ventilis V4. Skysčiui tekant, vamzdžiuose tarp skysčio įtekėjimo ir ištekėjimo taško dega oranžinė bėgančioji šviesa. Oranžinė šviesa reaktoriaus korpuse vaizduoja įjungtą šildymą.

Priklausomai nuo esančio lygio, šildymo ir aušinimo bei maišymo galima pritaikyti tokius reaktoriaus šilumos mainų parametrus:

1. Šildymas ir aušinimas išjungtas, maišymas neturi įtakos:

lygis $h = 0 \%$	lygis $h = 100\%$
$T_1 = 32,7 \text{ s,}$	$T_1 = 256,9 \text{ s,}$
$D = 1,172,$	$D = 2,639,$
$\omega_0 = 0,0545s^{-1}$	$\omega_0 = 0,0191s^{-1}$

2. Šildymas ir aušinimas nemaišant:

lygis $h = 0 \%$	lygis $h = 100\%$
$T_1 = 18,1 \text{ s,}$	$T_1 = 146,8 \text{ s,}$
$D = 1,040,$	$D = 2,020,$
$\omega_0 = 0,0732s^{-1}$	$\omega_0 = 0,0257s^{-1}$

3. Šildymas ir aušinimas maišant:

lygis $h = 0 \%$	lygis $h = 100\%$
$T_1 = 9 \text{ s,}$	$T_1 = 73,0 \text{ s,}$
$D = 1,002,$	$D = 1,519,$
$\omega_0 = 0,1039s^{-1}$	$\omega_0 = 0,0356s^{-1}$

Reaktoriaus šilumos mainų perdavimo funkcijos:

$$W(s) = \frac{K_s}{(1+T_1s)(1+T_2s)} \quad (2)$$

arba

$$W(s) = \frac{K_s}{(1+\frac{2D}{\omega_0}s)(1+\frac{1}{\omega_0^2}s)} \quad (3)$$

Čia $T_2 = 10,3 \text{ s}$; $K_s = 8 \text{ }^\circ\text{C/V}$ šildant; $K_s = -15 \text{ }^\circ\text{C}$ aušinant [27].

1.8. Techninė įranga

1.8.1. PLV I/O maitinimo šaltinis

CX2040 sistemai ir jos I/O moduliams užmaitinti naudojamas CX2100-0014 maitinimo šaltinis pavaizduotas 1.8.1 paveikslėlyje, kuris yra valdomas TwinCAT 3 programinės įrangos.



1.8.1 pav. CX2100-0014 maitinimo šaltinio modulis [28]

CX2100-0014 maitinimo šaltinio techniniai duomenys pateikiami lentelėje 3.

3 lentelė. CX2100-0014 maitinimo šaltinio techniniai duomenys [29]

Parametras	Charakteristikos
Maitinimo šaltinis	24 V DC (-15%/+20%)
Maksimali išėjimo galia	130 W
I/O prijungimas	E-bus arba K-bus, automatinis atpažinimas
Ekranas	FSTN (angl. <i>Film compensated super-twisted nematic</i>) ekranas, 2 eilučių x 16 simbolių, apšviestas
LED diagnostikos indikacijos	1-veikimo, 1-I/O veikimas, 1-klaida
Maksimalus galios suvartojimas	3.5 W
Atsparumo klasė	IP 20

1.8.2. Bendras sistemos maitinimo šaltinis

CX2100-0014 maitinimo šaltinis skirtas tiekti maitinimo įtampą I/O moduliams ir procesoriui. Reaktoriaus standui ir PLV sistemai naudojamas *Phoenix Contact Power UNO 2902992 24 V DC/60W* maitinimo šaltinis pavaizduotas 1.8.2 paveikslėlyje.



1.8.2 pav. PhoenixContact Power UNO maitinimo šaltinis [30]

Phoenix Contact Power UNO maitinimo šaltinio techniniai duomenys pateikiami lentelėje 4.

4 lentelė. Phoenix Contact Power UNO maitinimo šaltinio techniniai duomenys [31]

Parametras	Charakteristikos
Maitinimo šaltinis	24 V DC (-15%/+20%)
Maksimali išėjimo galia	130 W
I/O prijungimas	E-bus arba K-bus, automatinis atpažinimas
Ekranas	FSTN (angl. <i>Film compensated super-twisted nematic</i>) ekranas, 2 eilučių x 16 simbolių, apšviestas
LED diagnostikos indikacijos	1-veikimo, 1-I/O veikimas, 1-klaida
Maksimalus galios suvartojimas	3.5 W
Atsparumo klasė	IP 20

1.8.3. Diskretinių išėjimų modulis

Stendo maišyklės bei ventilių valdymui naudojamas *Beckhoff EL2809* 16 diskretinių išėjimų modulis pavaizduotas 1.8.3 paveikslėlyje.



1.8.3 pav. Beckhoff EL2809 diskretinių išėjimų modulis [32]

Beckhoff EL2809 diskretinių išėjimų modulio techniniai duomenys pateikiami lentelėje 5.

5 lentelė. Beckhoff EL2809 diskretinių išėjimų modulio techniniai duomenys [33]

Parametras	Charakteristikos
Išėjimų skaičius	16
Apkrovos tipas	Omine, induktyvioji, lempinė
Maksimali išėjimo srovė	0.5 A (trumpojo jungimo apsauga) kiekviename kanale
Maksimali trumpo jungimo srovė	< 2 A
Persijungimo trukmė	T _{ON} : 60 μs, T _{OFF} : 300 μs,
Nominalus drėgnumas	95 % be kondensato
Darbinė temperatūra	-25...+60 °C
Atsparumo klasė	IP 20

1.8.4. Analoginių įėjimų modulis

Temperatūros bei lygio stebėjimui naudojamas Beckhoff EL3104 4 analoginių įėjimų modulis pavaizduotas 1.8.4 paveikslėlyje.



1.8.4 pav. Beckhoff EL3104 analoginių įėjimų modulis [34]

Beckhoff EL3104 analoginių įėjimų modulio techniniai duomenys pateikiami lentelėje 6.

6 lentelė. Beckhoff EL3104 analoginių įėjimų modulio techniniai duomenys [35]

Parametras	Charakteristikos
Įėjimų skaičius	4
Signalų įtampa	-10...+10 V
Įėjimo filtro dažnio riba	5kHz
Nuskaitymo rezoliucija	16 bit
Matavimo paklaida	< ±0.3 %
Darbinė temperatūra	-25...+60 °C
Atsparumo klasė	IP 20

1.8.5. Analoginių išėjimų modulis

Analoginių signalų ventilių valdymui naudojamas Beckhoff EL4004 4 analoginių išėjimų modulis pavaizduotas 1.8.5 paveikslėlyje.



1.8.5 pav. Beckhoff EL4004 analoginių išėjimų modulis [36]

Beckhoff EL4004 analoginių išėjimų modulio techniniai duomenys pateikiami lentelėje 7.

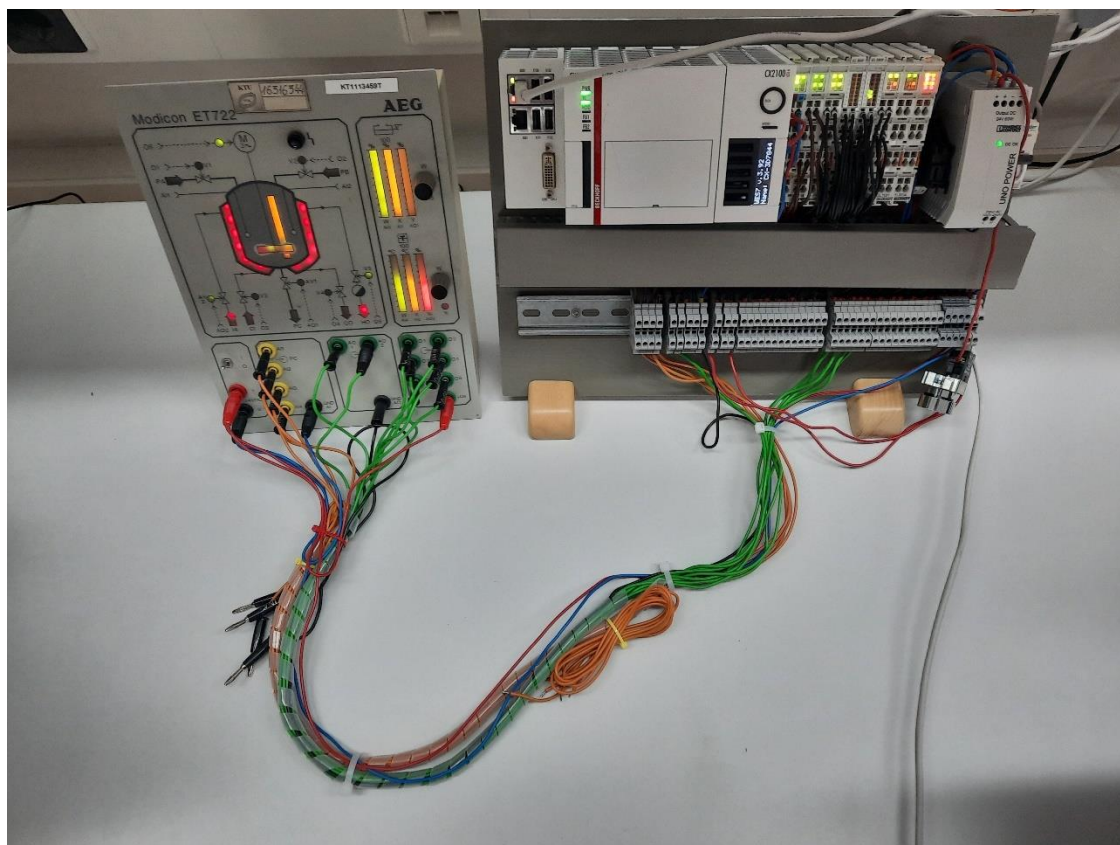
7 lentelė. Beckhoff EL4004 analoginių išėjimų modulio techniniai duomenys [37]

Parametras	Charakteristikos
Išėjimų skaičius	4
Signalų įtampa	0...10 V
Apkrova	> 5kΩ (trumpo jungimo apsauga)
Nuskaitymo rezoliucija	12 bit
Konversijos laikas	~ 250 μ
Darbinė temperatūra	-25...+60 °C
Atsparumo klasė	IP 20

2. Tiriemoji dalis

2.1. Techninės įrangos prijungimas

Kadangi šiame tiriamajame darbe nebus apsieinama be fizinės sistemos, tai yra *Beckhoff* programuojamo loginio valdiklio bei reaktoriaus imitacinio stendo. Kauno technologijos universiteto Automatikos katedros laboratorijoje sujungiama minėta įranga į bendrą sistemą (žr. 2.1.1 pav.).



2.1.1 pav. *Beckhoff* valdiklio ir reaktoriaus stendo sistema

2.2. Programinės įrangos konfigūravimas

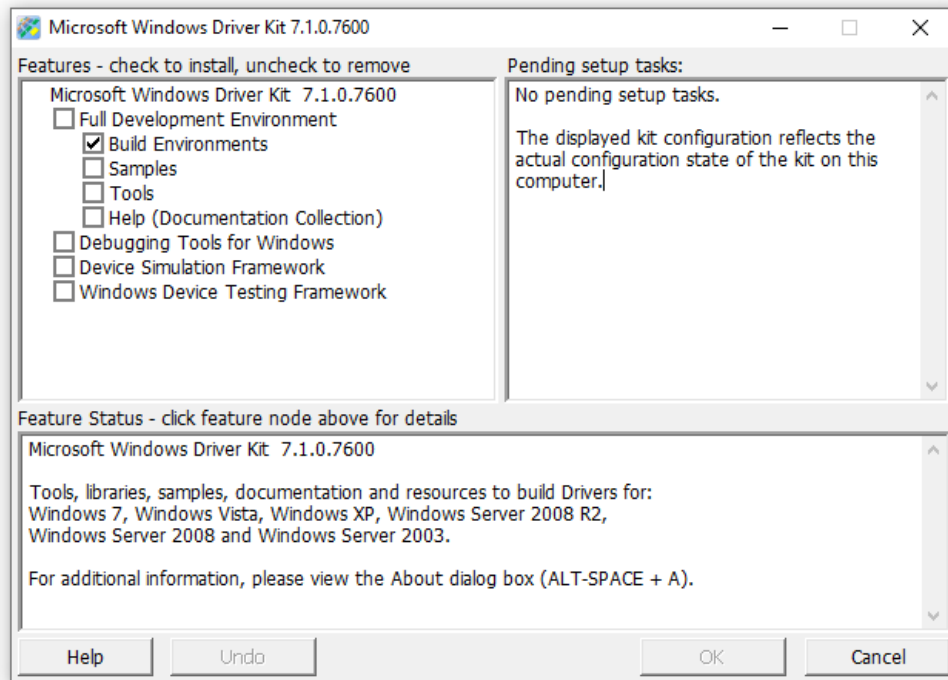
2.2.1. TwinCAT 3 konfigūravimas

Kompiliavimo programa

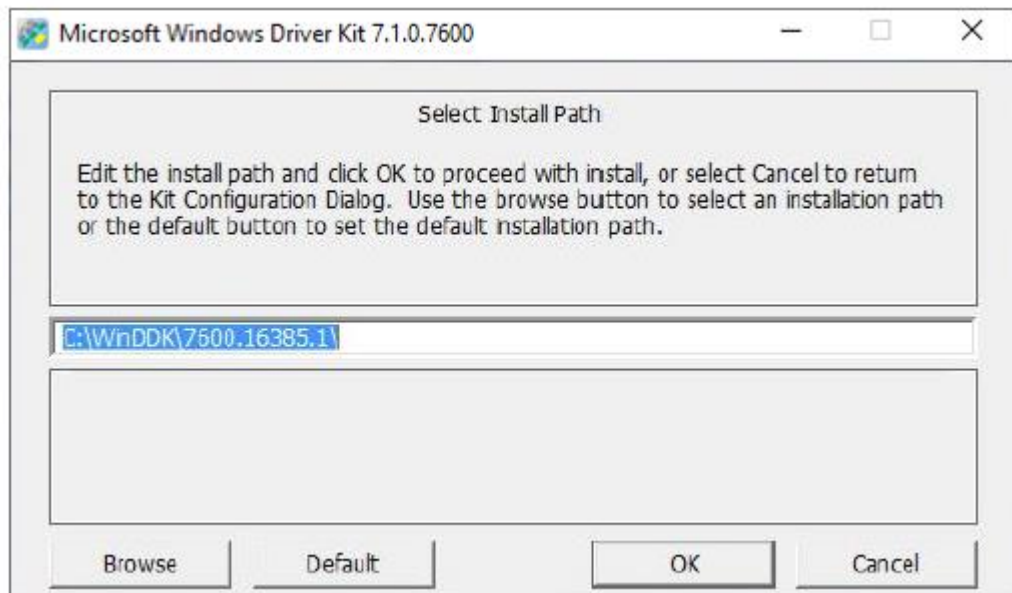
Tyrimo metu bus naudojama *Visual Studio 2015 Professional* programinė įranga kaip kompiliavimo programa konversijai iš *Simulink* modelio į C++ kodą.

TwinCAT 3 ir tvarkyklės

TwinCAT 3 programinė įranga paruošiama darbui įdiegus kaip *Visual Studio 2015 professional* papildinį. Taip pat bus reikalinga *WDK 7*(angl. *Windows Driver Kit*), ši tvarkyklė leis paleisti sugeneruotą C++ kodą *TwinCAT 3* platformoje. Diegiant šią tvarkyklę svarbu pažymėti *Build Environments* (žr. 2.2.1 pav.) ir pasirinkti saugojimo vietą, kuri bus naudojama vėliau [38] (žr. 2.2.2 pav.).



2.2.1 pav. WDK Instaliavimo metu *Build Enviroments* pasirinkimas

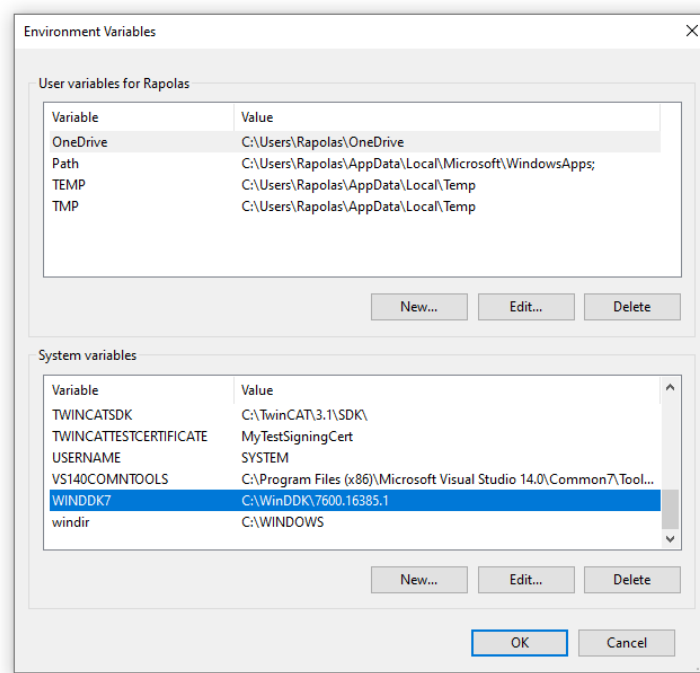


2.2.2 pav. WDK diegimo metu saugojimo vietos pasirinkimas

Toliau reikalinga nustatyti *System Variables* (žr. 2.2.3 pav.) į:

Variable name WINDDK7

Variable value C:nWinDDKn7600.16385.1

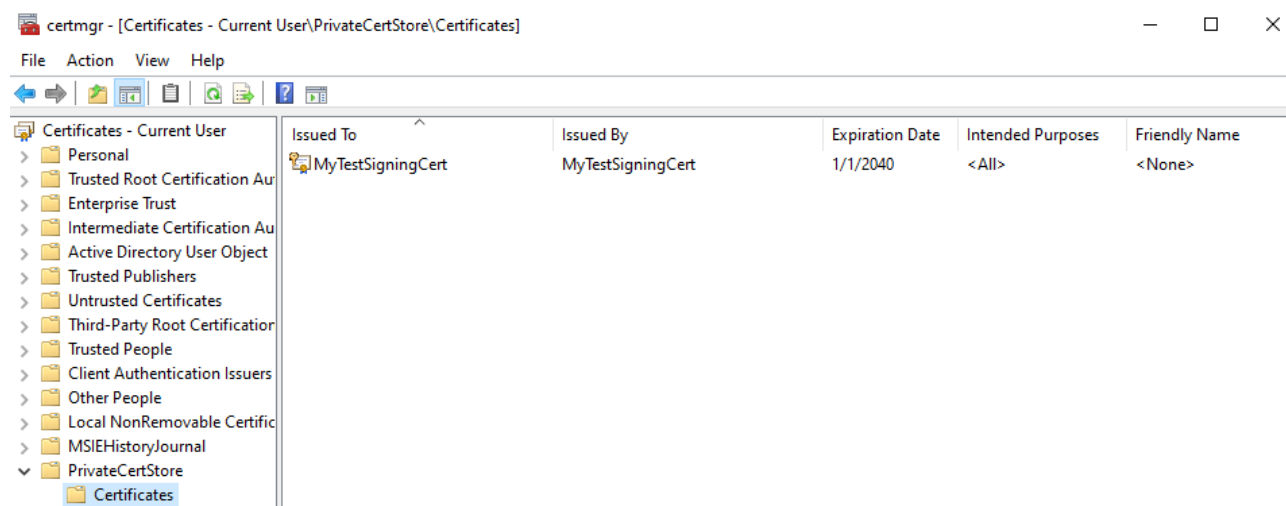


2.2.3 pav. *System variables* nustatymas

TwinCAT 3 C++ įgyvendinimui ant *x64* platformos, tvarkyklė turi būti sertifikuojama. Į *Visual Studio prompt* su administratoriaus teisėmis įrašoma ši eilutė:

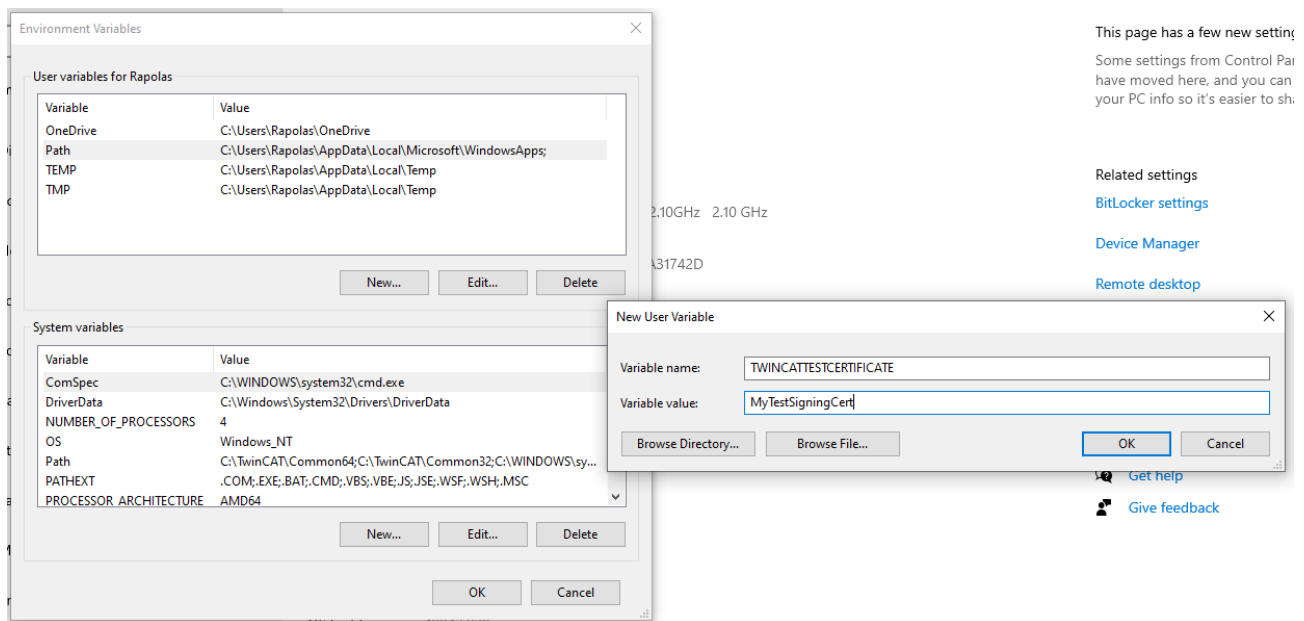
makecert -r -pe -ss PrivateCertStore -n CN=MyTestSigningCert MyTestSigningCert.cer

Taip sukuriant sertifikatą ir sukuriant *MyTestSigningCert.cer* failą, kuris randasi *Windows Certificate Store* (žr. 2.2.4 pav.).



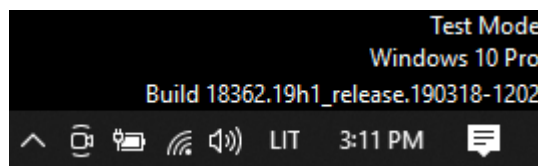
2.2.4 pav. Sertifikato sukūrimas

Toliau sukuriamas sertifikato *System variable* (žr. 2.2.5 pav.).



2.2.5 pav. Sukuriamas sertifikato *System variable*

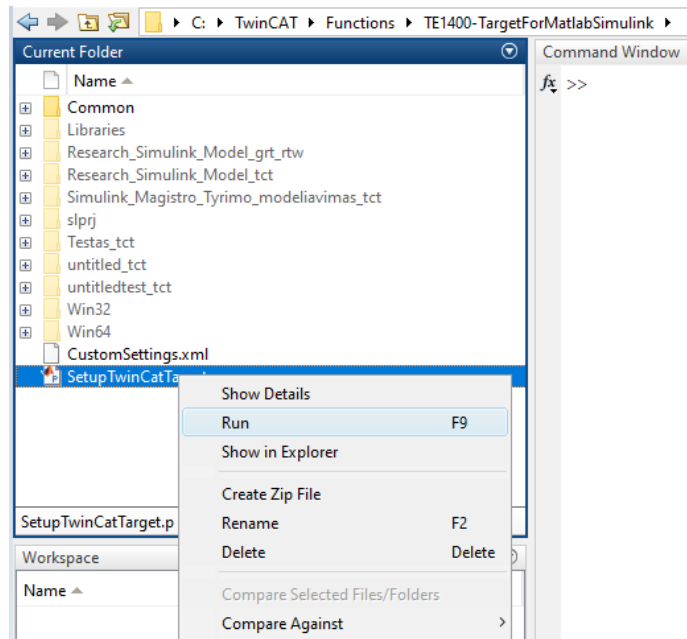
Kad įsitikinti ar kompiuteris yra paruoštas *MATLAB/SIMULINK* konversijai į *TwinCAT 3* programinę įrangą, perkrovus kompiuterį, dešiniame apatiniame kampe atsiranda užrašas *Test Mode* (žr. 2.2.6 pav.).



2.2.6 pav. Suaktyvuotas *Test Mode* režimas

2.2.2. *MATLAB/SIMULINK* konfigūravimas

Suinstaliavus *Matlab 2018a* programinę įrangą, taip pat suinstaliuojamas *TE1400-TargetForMatlabSimulink* papildinys ir paleidžiamas *Matlab* aplinkoje (žr. 2.2.7 pav.).



2.2.7 pav. TE1400 modulio instaliavimas

Taip pat įrašius komandų lange `mex -setup C++`, gaunamas patvirtinimas, kad *Visual Studio 2015 professional C++* yra *Simulink Coder* pagrindinis kompiliatorius (žr. 2.2.8 pav.).

```

Command Window
>> SetupTwinCatTarget
#####
##### TwinCAT TE1400 1.2.1240.0 Setup #####
TwinCAT TE1400 paths were added successfully.
TwinCAT TE1400 uses "Microsoft Visual C++" compilers to build TwinCAT modules from generated code.
It is recommended to use a "Microsoft Visual C++" compiler also for mex builds.
mex was configured to use the compiler "Microsoft Visual C++ 2015".
fx >>

```

2.2.8 pav. Visual Studio C++ 2015 yra Simulink Coder kompiliatorius

2.3. MATLAB/SIMULINK modelio sudarymas temperatūros reguliavimui

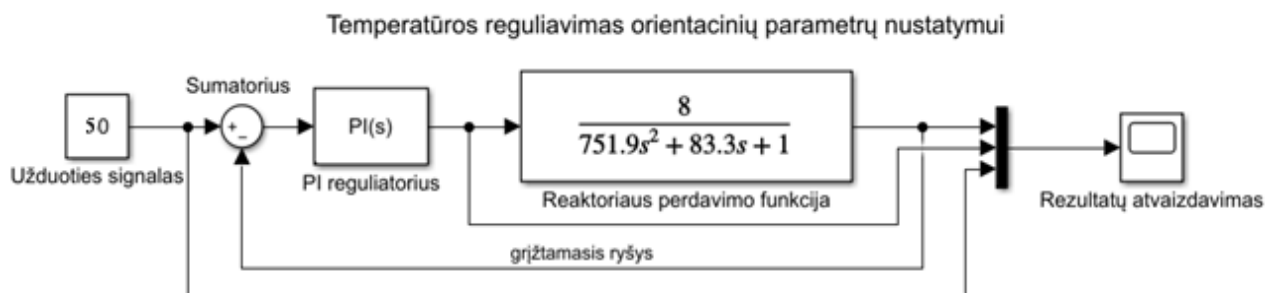
2.3.1. Tolydinis reguliatorius

Kaip ir buvo minėta 1.6 skyriuje, *TwinCAT 3* iš penkių galimų diskretizavimo metodų palaiko tik bilinearios transformacijos metodu perskaičiuotą reguliatorių. Taigi norint rasti diskretinio reguliatoriaus vertes, pirmiausia reikės *MATLAB/SIMULINK* aplinkoje sumodeliuoti teorinį reaktoriaus modelį, kad gauti tolydinio reguliatoriaus koeficientų vertes. Reaktoriaus šilumos mainų perdavimo funkcija gaunama pasinaudojus 1.7.1 skyriuje esančiais duomenimis. Perdavimo funkcija pasirinkta naudoti esant talpai užpildytai 100 %, įjungtas šildymas ir šaldymas bei maišymas.

Reaktoriaus perdavimo funkcija apskaičiuojama pagal (2) formulę:

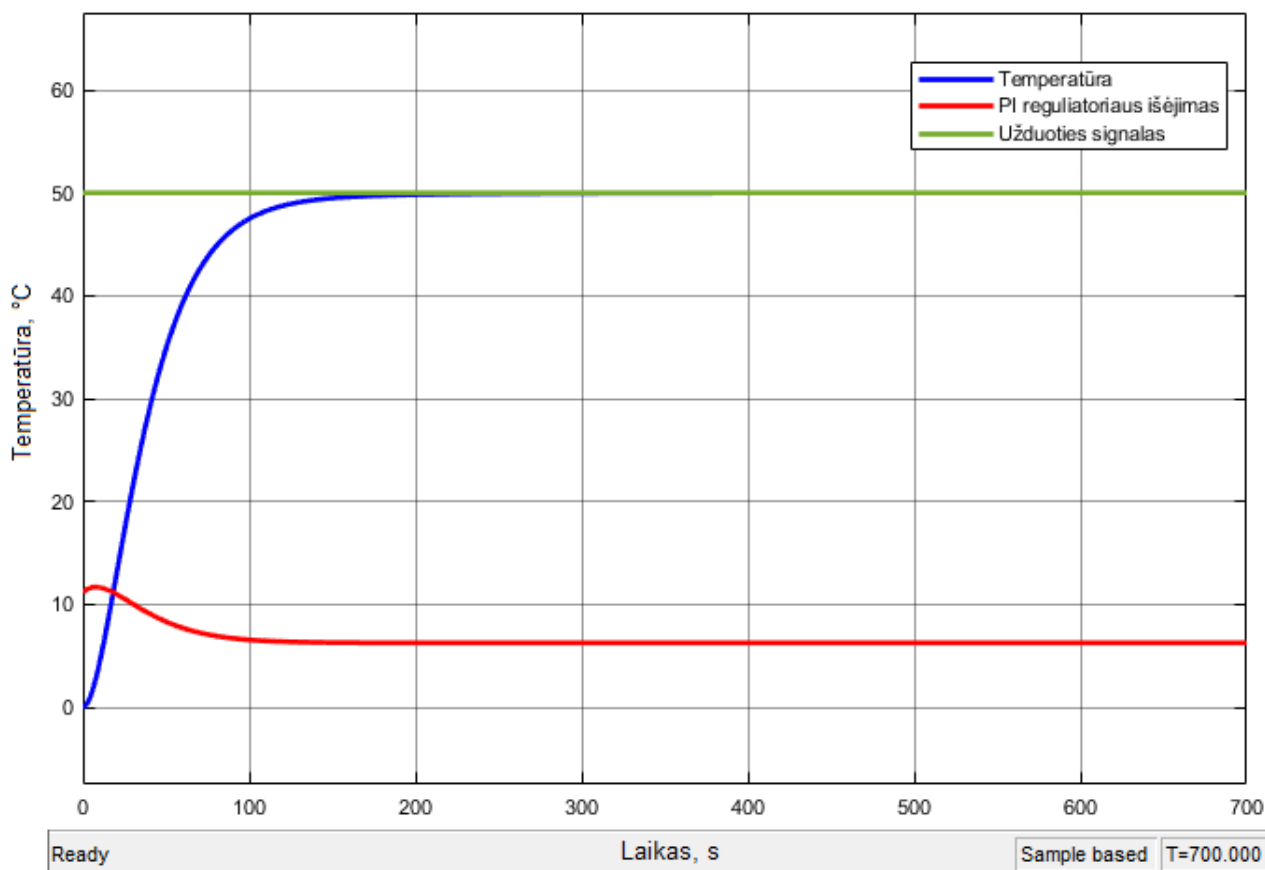
$$W(s) = \frac{8}{751.9s^2 + 83.3s + 1}; \quad (4)$$

MATLAB/SIMULINK aplinkoje atliekamas uždaros sistemos modeliavimas su PID tipo reguliatoriumi, orientaciniams parametrms nustatyti (žr. 2.3.1 pav.). PID tipo reguliatoriaus derinimas atliekamas *AutoTune* įrankiu, kuris yra integruotas pačiame reguliatoriuje.



2.3.1 pav. Tolydinės reaktoriaus sistemos temperatūros reguliavimas

Temperatūros valdymas su tolydiniu PI reguliatoriumi



2.3.2 pav. Tolydinės reaktoriaus sistemos temperatūros reguliavimo rezultatai

Atlikus tolydinės reaktoriaus temperatūros valdymo sistemos modeliavimą (žr. 2.3.2 pav.) paaiškėjo, kad sistemai suvaldyti užtenka tik PI reguliatoriaus. Temperatūra nusistovėjo per 243,667 s. be perreguliavimo ir statinės paklaidos. Toliau atliekama reguliatoriaus, kurio parametrai $P = 0,223347766410087$, $I = 0,00297703589193356$ transformacija į diskretinį, pasinaudojus bilinearos transformacijos metodu.

Bilineari transformacija

Bilinearios transformacijos metodu tolydinės sistemos perdavimo funkcija tiesiogiai perskaičiuojama į z – perdavimo funkciją. Taip skaičiuota z – perdavimo funkcija geriau išsaugo tolydinės sistemos dažnines savybes. Tiesiogiai pritaikant z – transformaciją, žinomas Laplaso vaizdas $W(s)$ pirmiausia turi būti transformuotas atgal į laiko funkciją, po to laiko funkcija diskretizuota ir atlikta z – transformacija, kad gauti perdavimo funkciją $W(z)$. Šis metodas tiesiogiai iš tolydinės perdavimo funkcijos skaičiuoja diskretinę perdavimo funkciją [39]:

$$z = e^{sT} \text{ arba } s = \frac{1}{T} \ln z; \quad (5)$$

Natūrinis logaritmas išskleidžiamas begalinę eilute;

$$\ln z = 2 \cdot \left[\frac{z-1}{z+1} + \frac{(z-1)^3}{3(z+1)^3} + \frac{(z-1)^5}{5(z+1)^5} + \dots \right]; \quad (6)$$

Tada transformacijos formulė:

$$s = \frac{2}{T} \cdot \frac{z-1}{z+1}; \quad (7)$$

Literatūroje ji vadinama Tustino formule. Ši formulė s plokštumos menamąją ašį perveda į z plokštumos vienetinio spindulio apskritimą, kairiąją pusę atvaizduoja į apskritimo vidų. Tustino formulės taikymą vadina bilinearia transformacija, kurios bendra formulė yra:

$$s = \frac{az+b}{z+c}; \quad (8)$$

Tolydinio PID reguliatorius perdavimo funkcija perskaičiuojama į diskretinę. PID perdavimo funkcija:

$$W(s) = k \left(1 + \frac{1}{T_i s} + T_d s \right) = k \frac{1+T_i s+T_i T_d s^2}{T_i s}; \quad (9)$$

Perskaičiavimui naudojama (7) transformacijos formulė. Tada:

$$W(z) = k \frac{1+T_i \frac{2z-1}{Tz+1} + T_i T_d \frac{4(z-1)^2}{T^2(z+1)^2}}{T_i \frac{2z-1}{Tz+1}} = k \frac{(z+1)^2 + \frac{2T_i}{T}(z-1)(z+1) + T_i T_d \frac{4(z-1)^2}{T^2}}{\frac{(z+1)^2}{T(z+1)}} =$$

$$k \frac{1 - \frac{2T_i}{T} + \frac{4T_i T_d}{T^2} + \left(2 - \frac{8T_i T_d}{T^2} \right) z + \left(1 + \frac{2T_i}{T} + \frac{4T_i T_d}{T^2} \right) z^2}{\frac{2T_i}{T}(z^2-1)} = k \frac{\left(\frac{T}{2T_i} - 1 + \frac{2T_d}{T} \right) + \left(\frac{T}{T_i} - \frac{4T_d}{T} \right) z + \left(\frac{T}{2T_i} + 1 + \frac{2T_d}{T} \right) z^2}{z^2-1}; \quad (10)$$

Skaitiklis ir vardiklis dalinamas iš z^2 , tada gaunamas:

$$W(z^{-1}) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-2}}; \quad (11)$$

Čia:

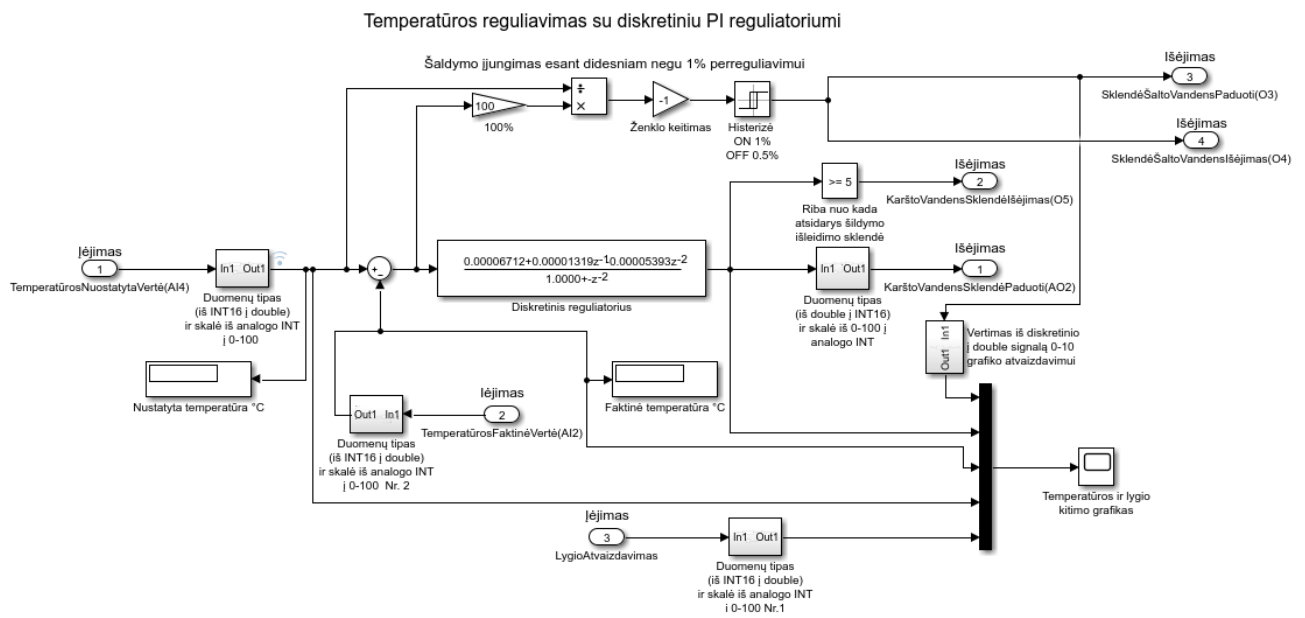
$$q_0 = k \left(1 + \frac{T}{2T_i} + 2 \frac{T_d}{T} \right); \quad (12)$$

$$q_1 = k \left(\frac{T}{T_i} - \frac{4T_d}{T} \right); \quad (13)$$

$$q_2 = k \left(\frac{T}{2T_i} - 1 + \frac{2T_d}{T} \right); \quad (14)$$

2.3.2. Diskretinis PI reguliatorius

Naudojantis išvesta bilinearos transformacijos formule (11) ir PI reguliatoriaus orientaciniais parametrais pateiktais 2.3.1 skyriuje, sudaromas diskretinis reguliatorius ir modelis reaktoriaus temperatūros valdymui (žr. 2.3.3 pav.). Modelio veikimo principas tai yra PI diskretinis reguliatorius į kurį ateina nuostatos signalo AI4 ir faktinės vertės AI2 paklaida. Kadangi minimalus AV1 šildymo ventilio valdantysis signalas yra 500 mV, šildymo išleidimo sklendė atidaroma esant įėjimo sklendei prasidarius 5 %. Šaldymo įėjimo bei išėjimo sklendės yra atidaromos esant faktinei temperatūros vertei daugiau negu 1 % ir uždaromos kai ši vertė nukrenta žemiau 0,5 %. Šaldymo įsijungimas vietoje 0 – 1 diskretiniu signalu vaizduojamas 0 – 10, kad aiškiau būtų matomas grafike eksperimento metu.



2.3.3 pav. MATLAB/SIMULINK modelis

Atliekant temperatūros valdymą realiu laiku, atliekamas reguliatoriaus parametrų derinimas geresniam rezultatui gauti pasinaudojus (12), (13), (14) formulėmis:

$$T = 0,001s;$$

$$P = 0,0000605272463207339;$$

$$I = 0,004588304687056643;$$

$$q_0 = k \left(1 + \frac{T}{2T_i} + 2 \frac{T_d}{T} \right) = 0,00006712;$$

$$q_1 = k \left(\frac{T}{T_i} - \frac{4T_d}{T} \right) = 0,00001319;$$

$$q_2 = k \left(\frac{T}{2T_i} - 1 + \frac{2T_d}{T} \right) = -0,00005393;$$

Toliau perkeliamas *Simulink* modelis į *TwinCAT 3* programinę įrangą. Tinkamai modelio konversijai reikalinga nustatyti nustatymus, kurie reikalingi, kad *Simulink* modelis veiktų *TwinCAT 3* programiniame pakete. Pirmiausia nustatomas modelio kintamųjų skenavimo fiksuotas žingsnis, kuris modelyje naudojamas $0,001$ s (žr. 2.3.5 pav.), toks koks yra *EtherCAT 1kHz* ciklo dažnis.

Solver

Simulation time

Start time: 0.0 Stop time: 10.0

Solver selection

Type: Fixed-step Solver: auto (Automatic solver selection)

▼ Solver details

Fixed-step size (fundamental sample time): 0.001

Tasking and sample time options

Periodic sample time constraint: Unconstrained

Treat each discrete rate as a separate task

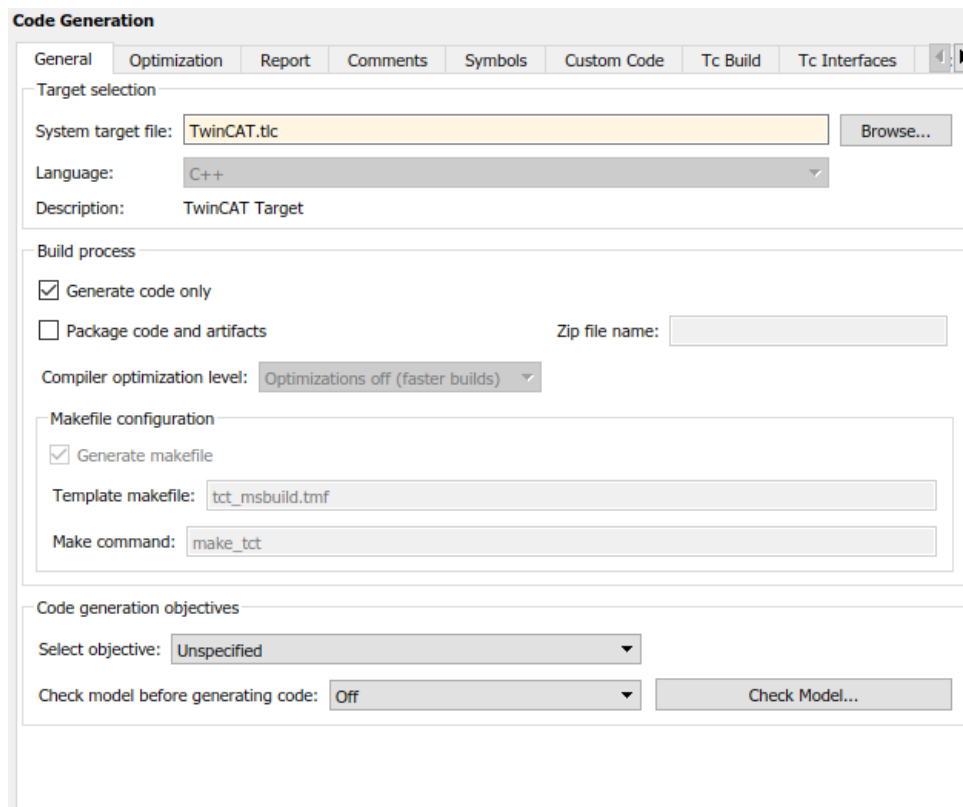
Allow tasks to execute concurrently on target

Automatically handle rate transition for data transfer

Higher priority value indicates higher task priority

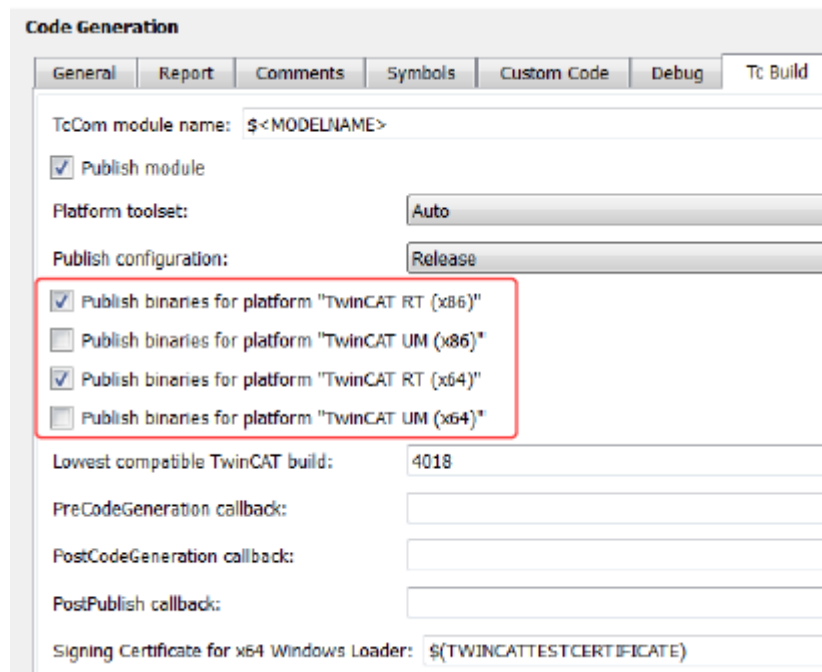
2.3.4 pav. Fiksuoto žingsnio pasirinkimas

Tuomet pasirenkamas sistemos kodo generavimo užduoties failas *TwinCAT.tlc*, kur bus sugeneruotas *Simulink C++* kodas konversijai į *TwinCAT 3* programinį paketą (žr. 2.3.6 pav.).



2.3.5 pav. Užduoties failo pasirinkimas

Tc Build skiltyje įsitikinama, kad pasirinkta teisinga platforma (žr. 2.3.7 pav.).

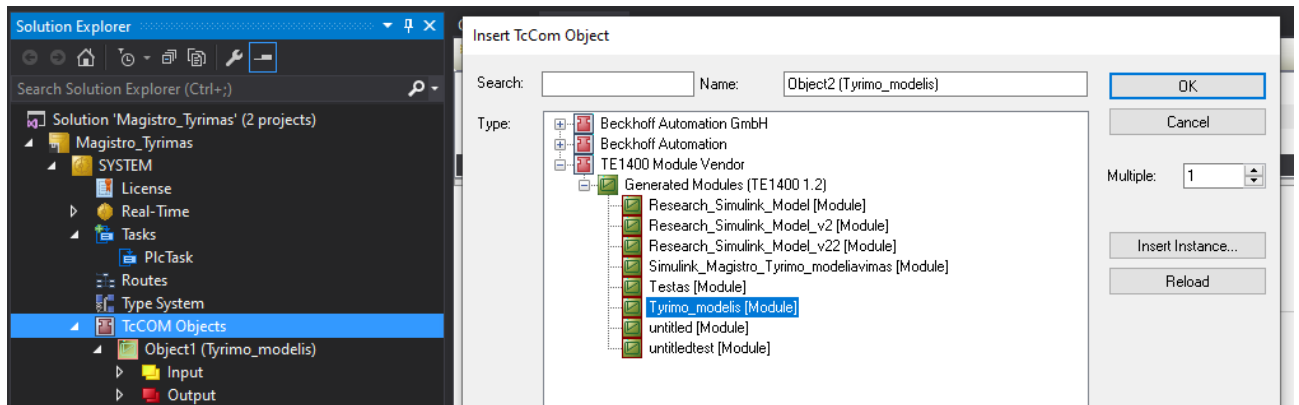


2.3.6 pav. Platformos pasirinkimas

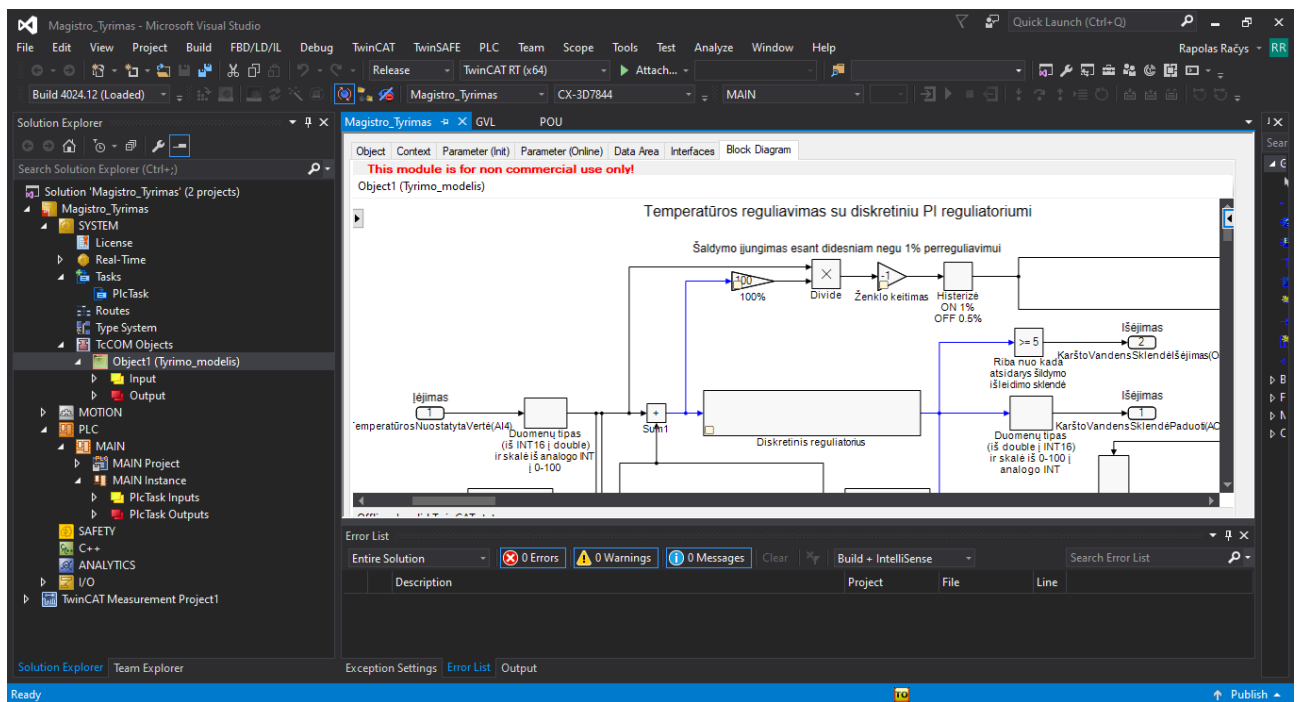
Simulink aplinkoje užduočių juostos dešinėje paspaudžiamas *Build Model* mygtukas, kuris sukuria *TMC* (angl. *Trace Message Control*) formato failą, kuris toliau bus naudojamas *TwinCAT 3* programinėje įrangoje.

2.4. MATLAB/SIMULINK modelio įkėlimas į TwinCAT 3

Atlikus tinkamus nustatymus atidaromas *Visual Studio 2015 Professional*. Pasirenkama *SYSTEM* skiltis ir *TcCOM Objects*, *Add New Item* ir *TE1400 Module Vendor* pasirenkamas sugeneruotas *Simulink* modelio kodas (žr. 2.3.8 pav.).

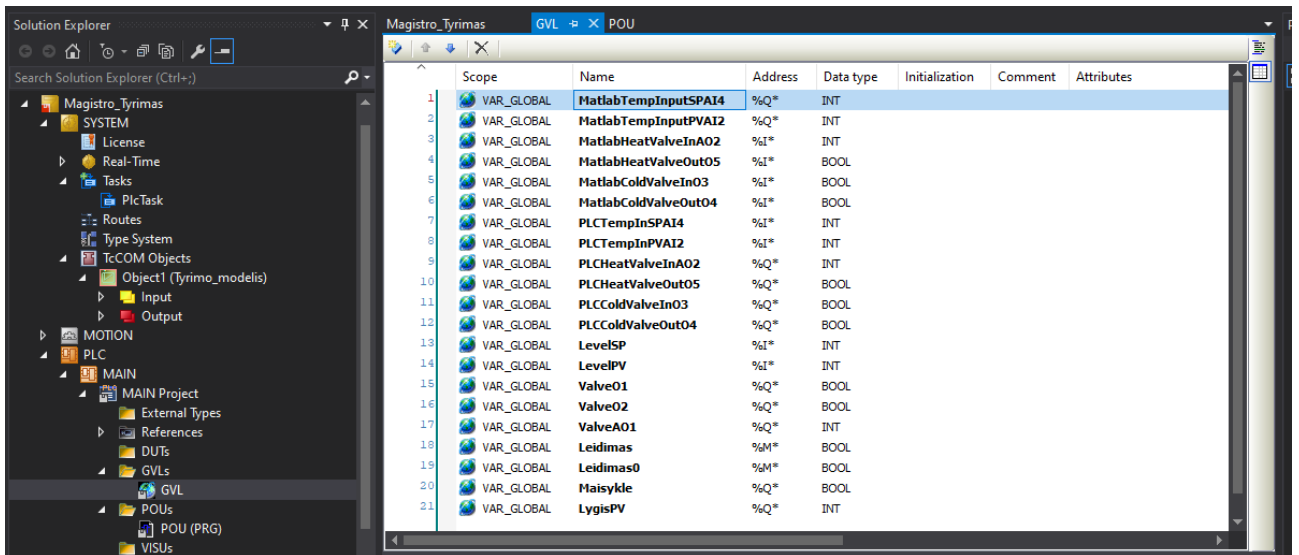


2.4.1 pav. Įkeliamas sugeneruotas *Simulink* CFC kodas



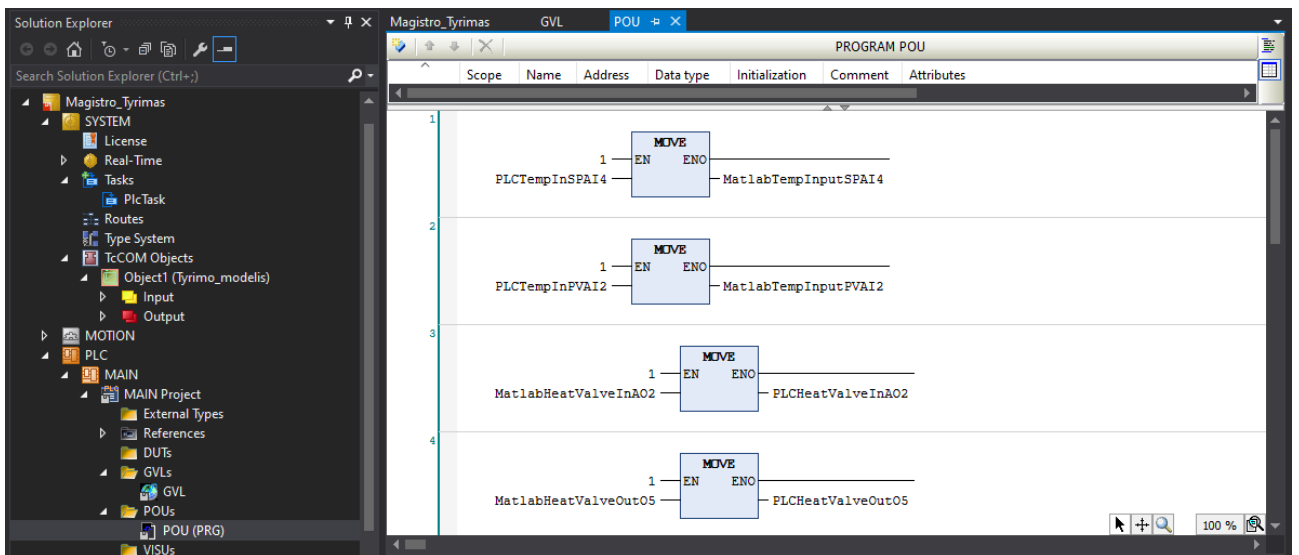
2.4.2 pav. Įkeltas *Simulink* modelis į *Visual Studio* programinę įrangą

Toliau reikalinga priskirti fizinius įėjimus bei išėjimus, kad tai atlikti sukuriamas globalus kintamųjų sąrašas (*GVL*), kur pagal sukurtą *Simulink* modelį sukuriami atitinkami kintamieji, kurie bus naudojami temperatūros bei lygio valdymui (žr. 2.4.3 pav.).



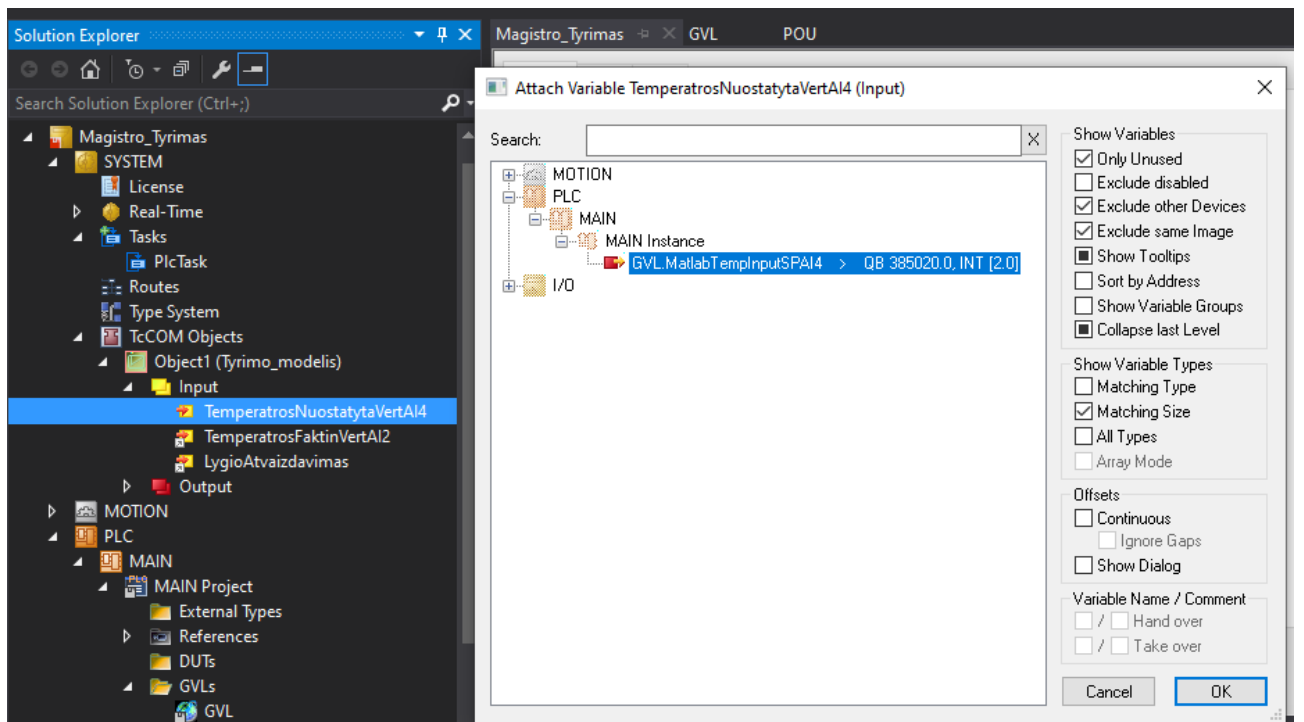
2.4.3 pav. Globalus kintamųjų sąrašas

Sukurti kintamieji priskiriami *Simulink* modeliui, tai yra *POU* (angl. *Program Organization Unit*) skiltyje fiziniai PLV įėjimai ir išėjimai perkeltami į *Simulink* vidinius išėjimus ir įėjimus pasinaudojus *MOVE* blokais (žr. 2.4.4 pav.).



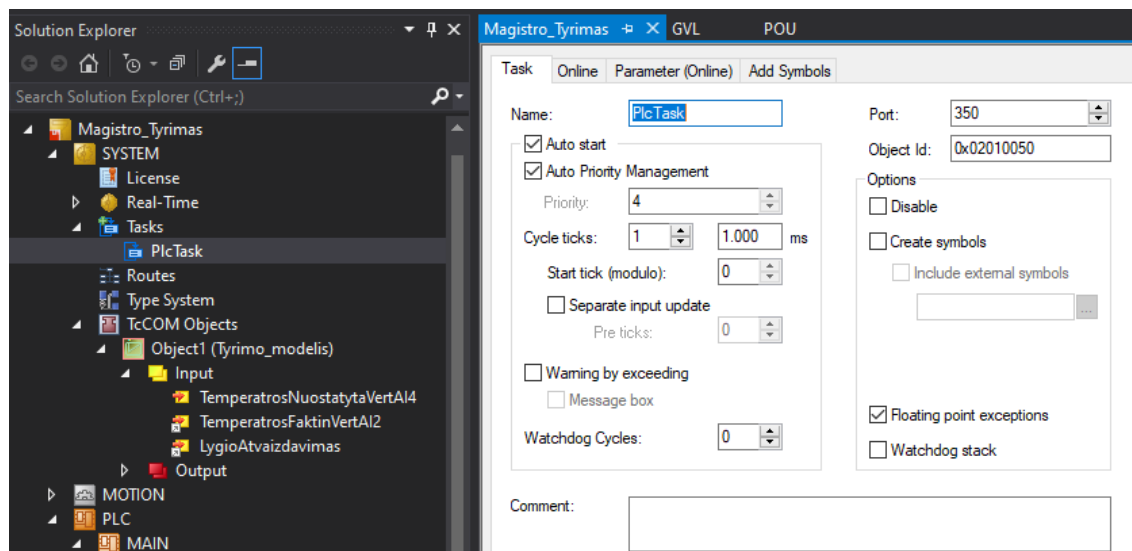
2.4.4 pav. PLV kintamųjų priskyrimas *Simulink* modelio įėjimams bei išėjimams

Sutvarkius kintamuosius galima juos pridėti prie *Simulink* modelio. Pasirinkus *TcCOM Objects*, *Object1*, *Input*, pasirenkamas reikalingas įėjimas ar išėjimas, paspaudus dešinį pelės klavišą pasirenkamas *Change Link* ir atsidariusiame lange išsirenkamas sukurtas kintamasis (žr. 2.4.5 pav.).

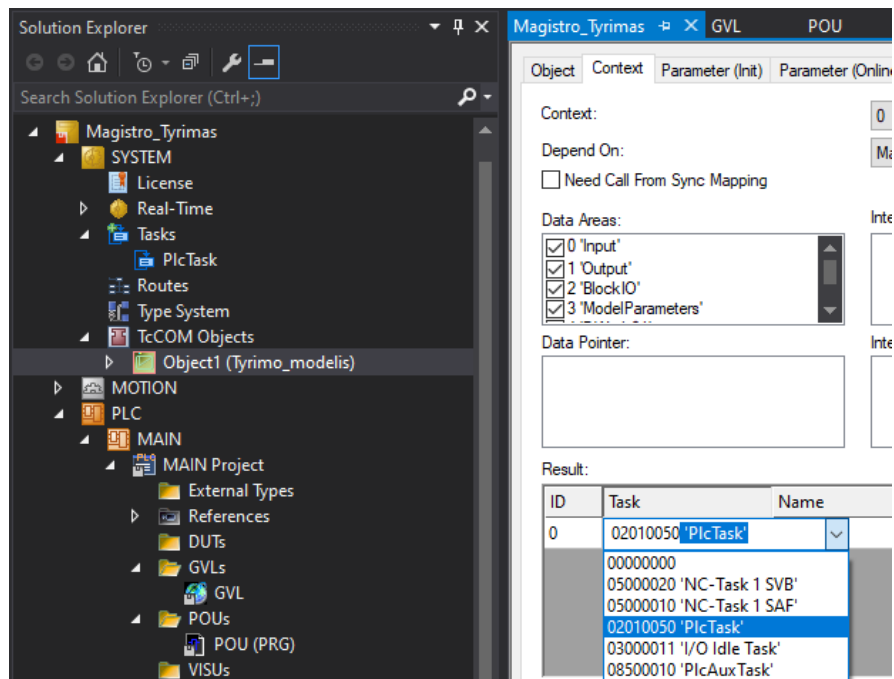


2.4.5 pav. Kintamojo priskyrimas *Simulink* modelio įėjimui

Atlikus veiksmus su kintamaisiais reikalinga nustatyti ciklą skaitymo dažnį, šis laikas turi sutapti su prieš tai *MATLAB/SIMULINK* nustatytu laiku, šiuo atveju $0,001$ s. Taigi Tasks skiltyje sukuriamas Plc Task užduotis, kurioje nustatomas reikalingas programos skaitymo dažnis (žr. 2.4.6 pav.) ir toliau TcCOM Objects, Object1, Context skiltyje, *Rezult* lange pasirenkama sukurta užduoties vykdymo programa PlcTask (žr. 2.4.7 pav.).

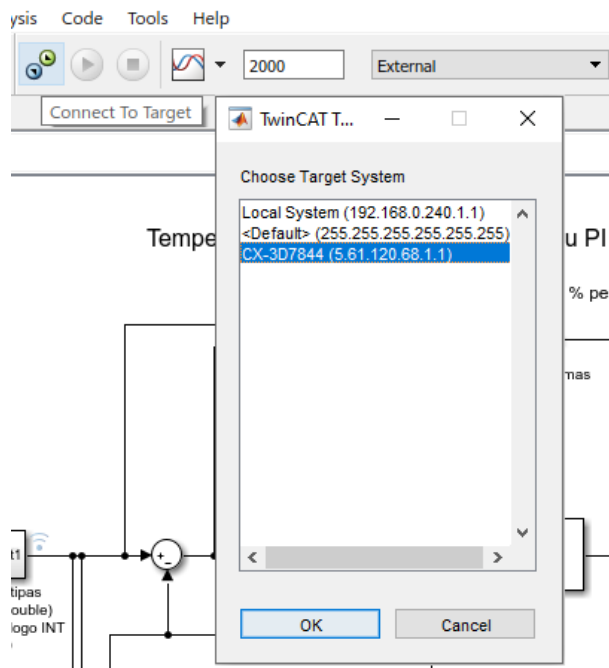


2.4.6 pav. Sukuriamas programos vykdymo laikas



2.4.7 pav. Priskiriamas programos vykdymo laikas *Simulink* programai

Toliau galima pereiti prie programos paleidimo, kad būtų galima pradėti atlikti temperatūros reguliavimą, tam reikalinga *Simulink* modelyje užduočių juostoje vietoje *Normal* pasirinkti *External* programos vykdymą, paspaudus *Connect to Target* mygtuką, atsidariusiame lange pasirinkti PLV adresą prie kurio bus jungiamasi ir žaliu mygtuku užduočių juostoje paleisti programos vykdymą (žr. 2.4.8 pav.).

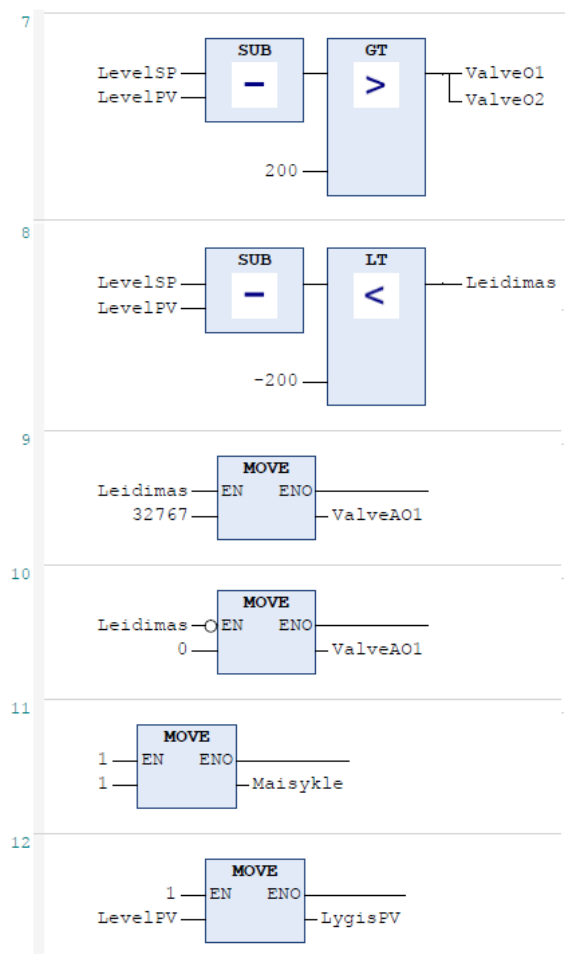


2.4.8 pav. ADS prisijungimo nustatymas

2.5. Lygio reguliavimas

Kadangi *MATLAB/SIMULINK* programinėje įrangoje *TE1400 Target for MATLAB/SIMULINK demo* versija leidžia naudotis tik iki 5 įėjimų, 5 išėjimų ir iki 100 blokų, lygio reguliavimas sukuriamas

PLV viduje naudojantis *TwinCAT 3* programinę įrangą. Lygio palaikymas atliekamas relinės logikos reguliatoriumi, esant signalo skirtumui daugiau arba mažiau 200, sistema skysti išleidžia arba papildo. Realizuota programa pavaizduota 2.5.1 paveikslėlyje.

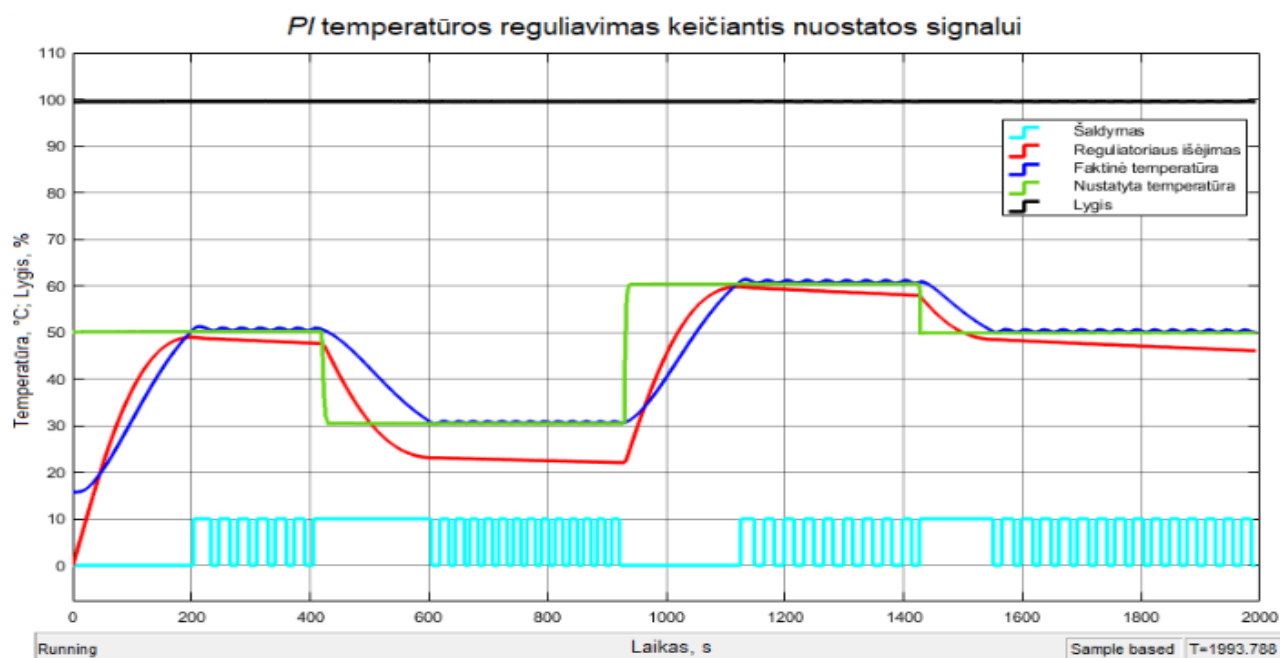


2.5.1 pav. Lygio reguliavimo programa

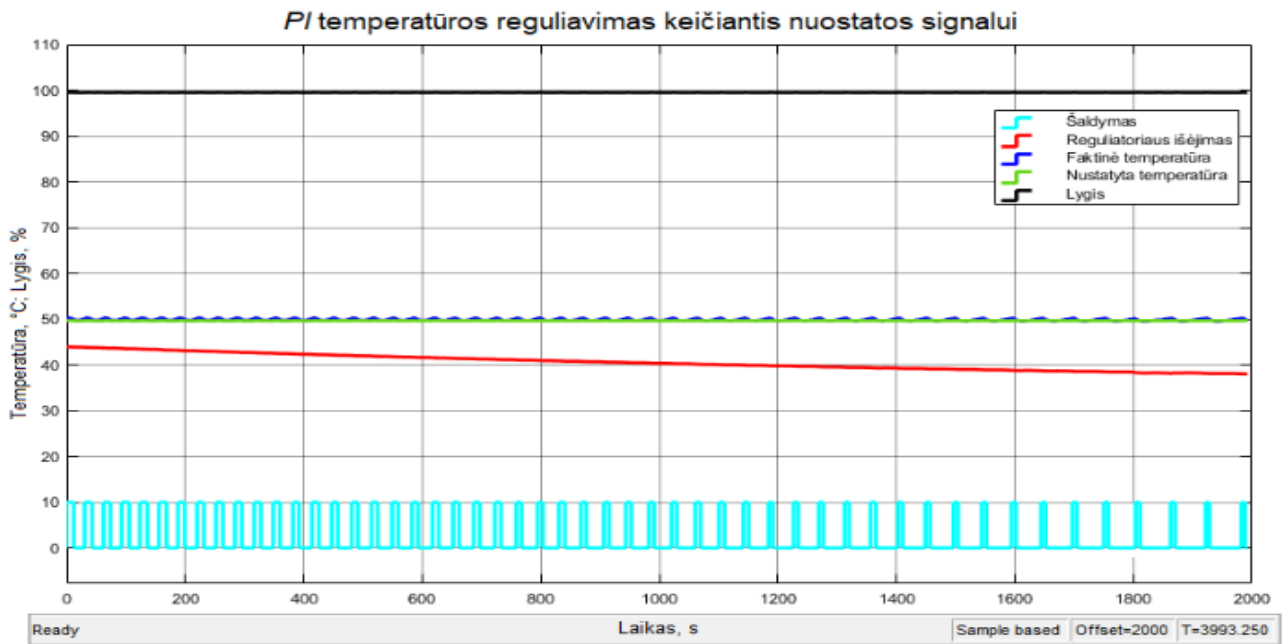
3. Reaktoriaus valdymo sistemos tyrimas

3.1. PI diskretinis reguliatorius

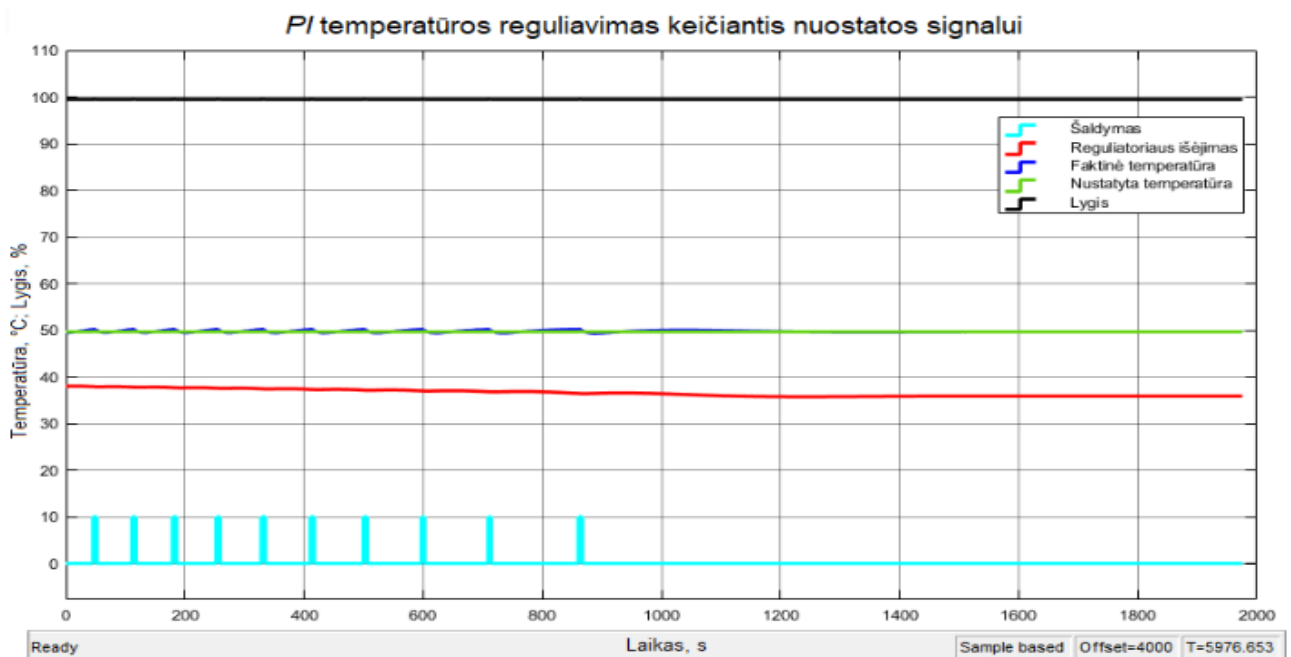
Paleidus ir suderinus realizuotą reaktoriaus temperatūros bei lygio palaikymo programą, atlikti eksperimentai siekiant išsiaiškinti kaip sistema reaguoja į temperatūros ir lygio nuostatos signalo pokyčius. Pirmu bandymu keičiamas temperatūros nuostatos signalas keturis kartus (žr. 3.1.1 pav.). Keičiant temperatūrą nuo nusistovėjusios $16,26\text{ }^{\circ}\text{C}$ iki $50,22\text{ }^{\circ}\text{C}$, temperatūra pakyla per 200 s su $1,99\%$ perreguliavimu ir nusistovi su nežymiais $1,12\%$ svyravimais, reguliatoriaus išėjimas pradeda kristi. Toliau po $423,2\text{ s}$ temperatūra pakeista iki $30,45\text{ }^{\circ}\text{C}$, sistema ją pasiekia per $182,68\text{ s}$ be perreguliavimo tačiau taip pat su $1,08\%$ svyravimais. Trečiu temperatūros pakeitimu nuostatos signalas pakeliamas iki $60,31\text{ }^{\circ}\text{C}$, temperatūra nusistovi per $196,55\text{ s}$ ir ketvirtu bandymu temperatūra vėl nuleidžiama iki $49,88\text{ }^{\circ}\text{C}$, temperatūra nukrenta per $134,25\text{ s}$ su nežymiais svyravimais ir šiuo paskutiniu bandymu laukiama kol temperatūra galutinai nusistovės be svyravimų ir pastoviu reguliatoriaus išėjimu be šaldymo įsijungimo. Sistemos reakcija galutinai nusistovi po 3765 s (žr. 3.1.2 pav. ir 3.1.3 pav.) be papildomo šaldymo tik su $37,22\%$ atsidariusią šildymo sklende.



3.1.1 pav. Eksperimentiniai rezultatai keičiantis temperatūros nuostatos signalui

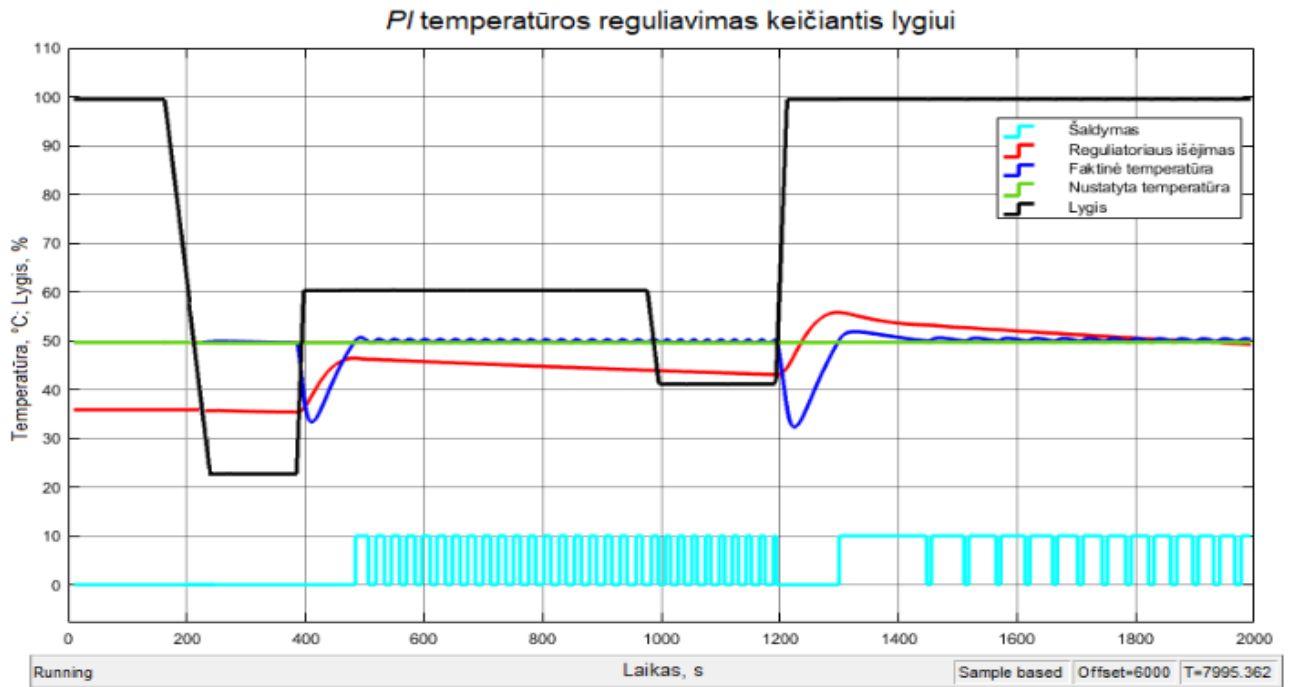


3.1.2 pav. Eksperimentiniai rezultatai keičiantis temperatūros nuostatos signalui, temperatūros nusistovėjimas



3.1.3 pav. Eksperimentiniai rezultatai keičiantis temperatūros nuostatos signalui, nusistovėjusi temperatūra

Toliau atliekamas sistemos reakcijos stebėjimas keičiantis lygio nuostatui (žr. 3.1.4 pav.). Sistemos procesas tęsiamas toliau su nusistovėjusia $49,88^{\circ}\text{C}$ temperatūra. Pirmu bandymu lygis sumažinamas iki $22,24\%$, kuris nukrenta per $58,21\text{ s}$, temperatūros kreivėje matyti nežymus pakilimas lygiui pasiekus nuostato vertę. Antru bandymu lygis pakeliamas iki $60,05\%$, kadangi pripilama 5°C temperatūros skystis, temperatūra talpoje nukrenta iki $33,69^{\circ}\text{C}$ ir vėl pasiekia nustatytą vertę per $101,24\text{ s}$, su $1,11\%$ svyravimais. Trečiu bandymu, lygis sumažinamas iki $41,78\%$, temperatūros kreivėje pokyčių nesimato po $197,88\text{ s}$ lygis pakeliamas iki pradinio 100% vertės, temperatūra sureaguoja analogiškai antruoju bandymu, tačiau nukrenta iki $32,53^{\circ}\text{C}$ ir atsistato su $3,2\%$ perreguliuavimu.



3.1.4 pav. Eksperimentiniai rezultatai keičiantis lygio nuostatos signalui

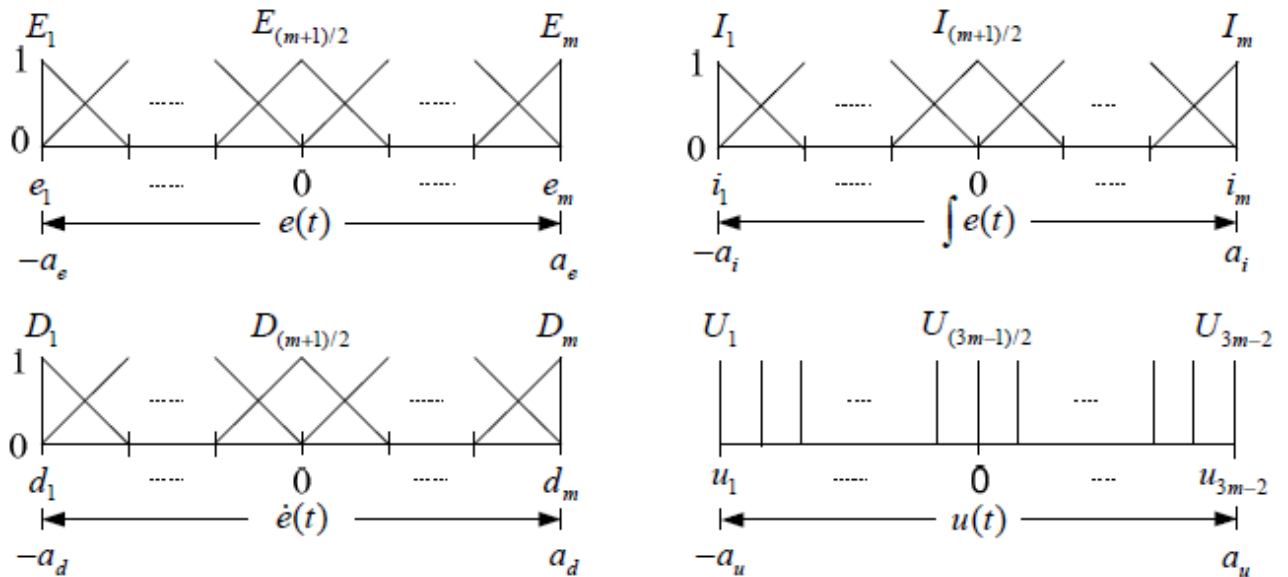
3.2. Fuzzy reguliatorius

Fuzzy reguliatorius sudaromas ekvivalentiškai tradiciniam PID reguliatoriui, kurio formulė yra:

$$u(t) = K_p e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt}; \quad (15)$$

Išėjimas $u(t)$ ir trys įėjimai $e(t)$, $\int e(t)$ ir $e'(t)$ gali būti kaip fuzzy kintamieji FLC (angl. *fuzzy logic controller*) modelyje. Priskiriama, kad $u(t)$, $e(t)$, $\int e(t)$ ir $e'(t)$ intervalai yra atitinkamai $OR_u = [-a_u, a_u]$, $OR_e = [-a_e, a_e]$, $OR_i = [-a_i, a_i]$ ir $OR_d = [-a_d, a_d]$.

Fuzzy įėjimų kintamieji $e(t)$, $\int e(t)$ ir $e'(t)$, kiekvienas atvaizduojamas lygių tarpų trikampės formos m fuzzy rinkiniais (žr. pav.). Tuo tarpu išėjimas $u(t)$ yra 3m-2 pavienėmis grupės funkcijomis (angl. *Membership functions*).



3.2.1 pav. Fuzzy įėjimų ir išėjimų kintamųjų funkcijos [40]

Pagal taip kaip vaizduojamos taisyklės *FAM* (angl. *fuzzy associative memory*) kubo išpjovoje, taisyklės yra apibrėžiamos šitaip:

$$\text{IF } e(t) \text{ is } E_i \text{ and } \int e(t) \text{ is } I_j \text{ and } \dot{e}(t) \text{ is } D_k \quad \text{THEN } u(t) \text{ is } U_l, l = i + j + k - 2; \quad (16)$$

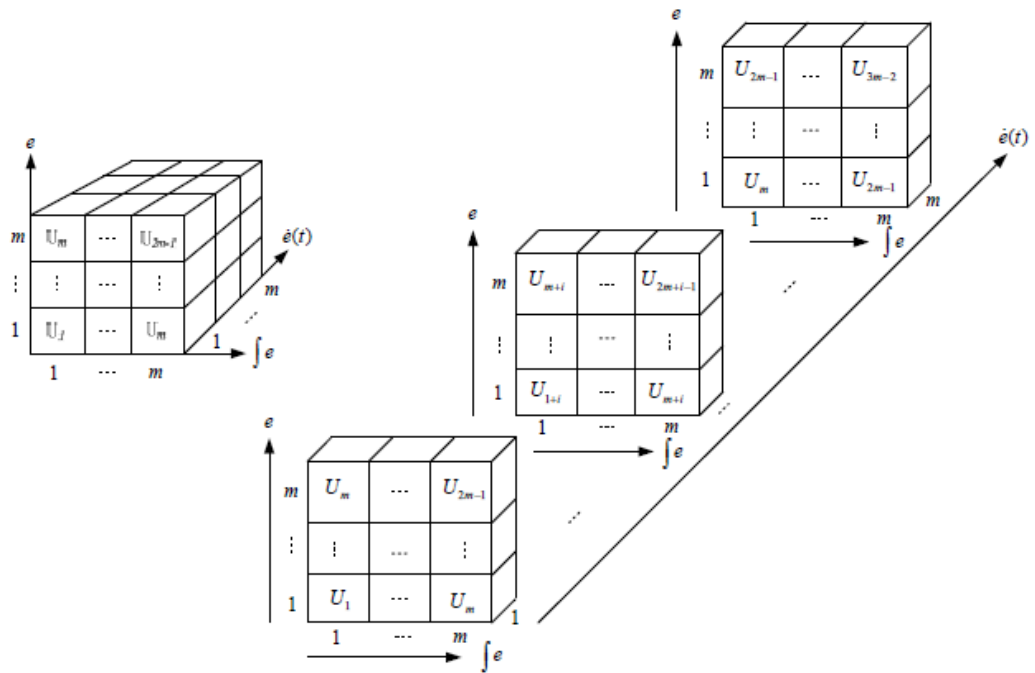
Tuomet iš čia išplaukia, kad *fuzzy* reguliatoriaus išėjimas gali būti aprašomas taip:

$$u(t) = \frac{a_u}{3a_e} e(t) + \frac{a_u}{3a_i} \int e(t) dt + \frac{a_u}{3a_d} \frac{de(t)}{dt}; \quad (17)$$

Tai reiškia, kad PID reguliatoriaus parametrai sutinka taip:

$$K_P = \frac{a_u}{3a_e}, K_I = \frac{a_u}{3a_i}, K_D = \frac{a_u}{3a_d}; \quad (18)$$

Taigi tradiciniam PID reguliatoriui ekvivalentiško *fuzzy* reguliatoriaus derinimo parametrai yra a_e, a_i, a_d ir a_u [41].



3.2.2 pav. FAC kubo išpjova [42]

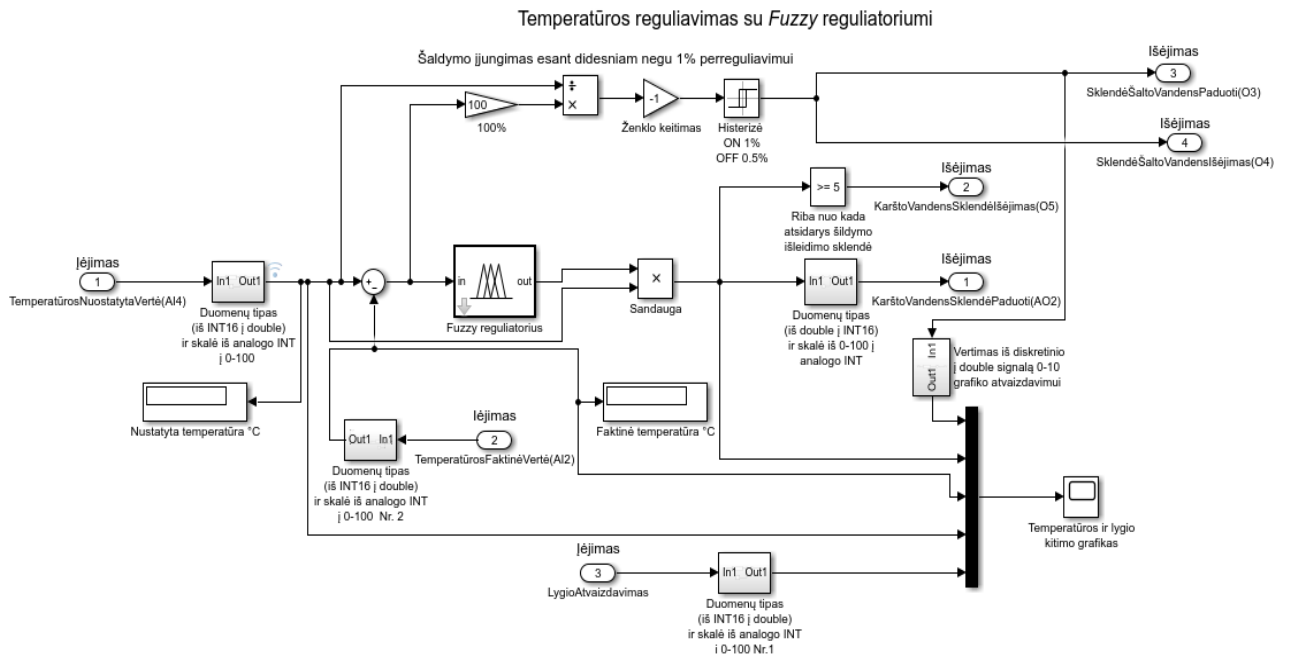
Reaktoriaus temperatūros palaikymo modelis su fuzzy reguliatoriumi

Šiam modeliui sudaryti naudojami tie patys 2.3.1 skyrelyje aprašyti duomenys tai yra $P = 0,223347766410087$, I dedamoji šiame eksperimente nebus naudojama, nes gauti rezultatai parodė, kad reguliatorius susitvarko tik su P dedamąja be didelės statinės paklaidos. Toliau pasinaudojus formule (18) apskaičiuojami diskretinio fuzzy parametrai, kurie bus naudojami modelyje (žr. 3.2.3 pav.):

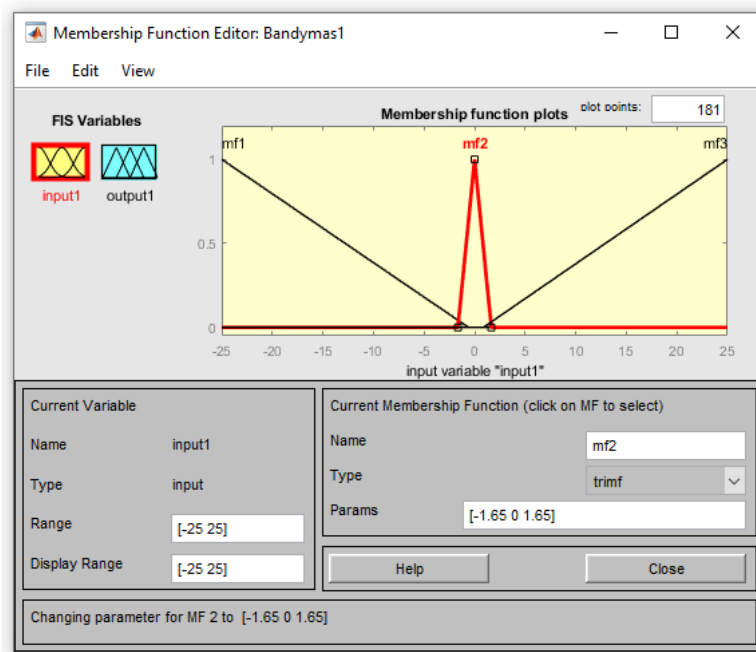
$$OR_e[-a_e, a_e] = [-1,65; 1,65], \text{ kuris nusako įėjimo } e(t) \text{ intervalą (žr. 3.2.4 pav.)}$$

$$OR_u[-a_u, a_u] = [-1,11; 1,11], \text{ kuris tenkina } K_p = 0,223347766410087 \text{ (žr. 3.2.5 pav.)}$$

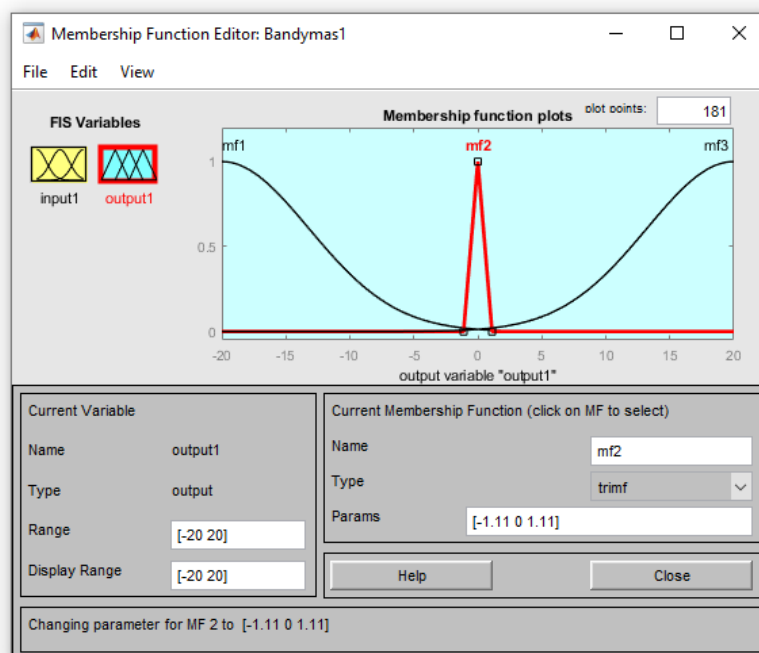
Modelio veikimo principas yra ekvivalentiškas modeliui su PI diskretiniu reguliatoriumi, tik šiame modelyje vietoje PI diskretinis reguliatorius pakeistas diskretiniu fuzzy reguliatoriumi.



3.2.3 pav. Temperatūros reguliavimas su fuzzy regulatoriumi

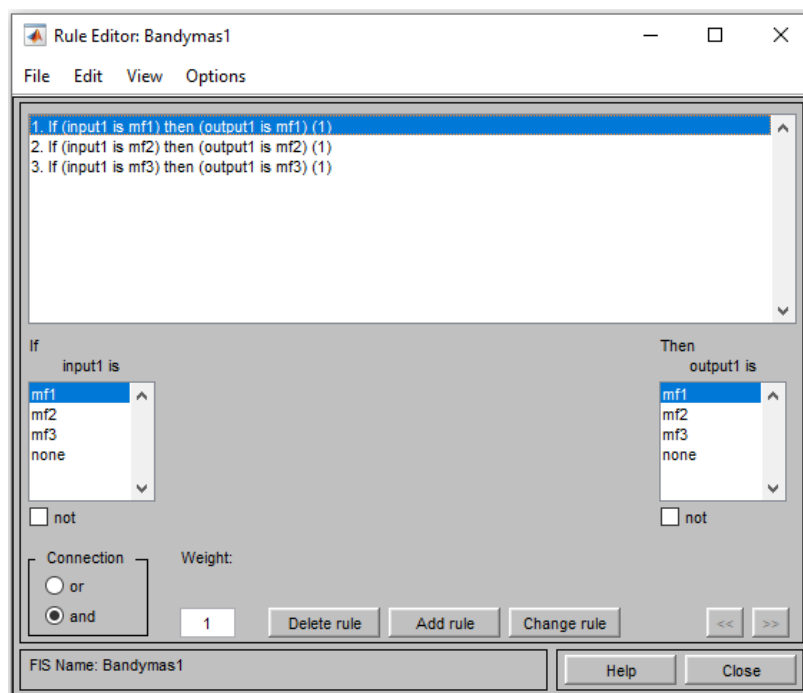


3.2.4 pav. Fuzzy regulatoriaus įėjimo parametrai



3.2.5 pav. Fuzzy regulatoriaus išėjimo parametrai

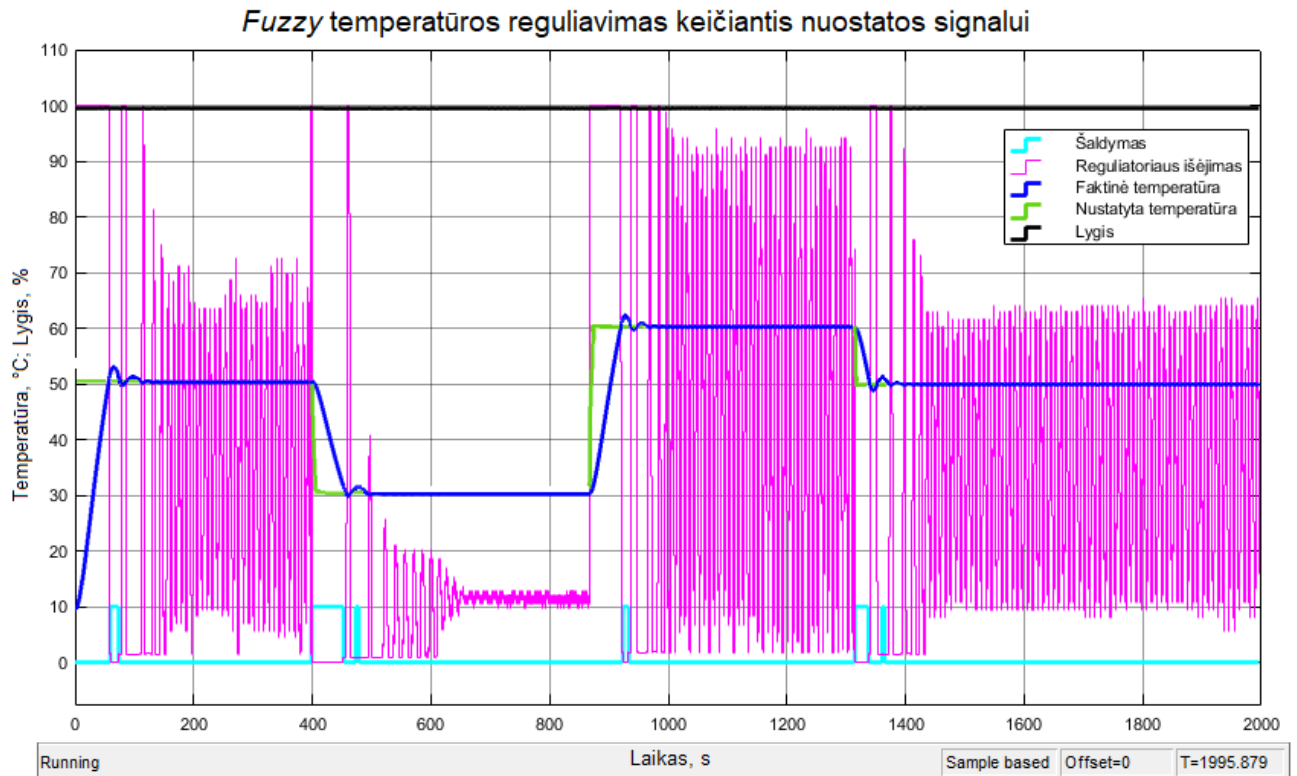
Sudaromos taisyklės priskiriant įėjimą išėjimui (žr. 3.2.6 pav.).



3.2.6 pav. Fuzzy regulatoriaus taisyklės

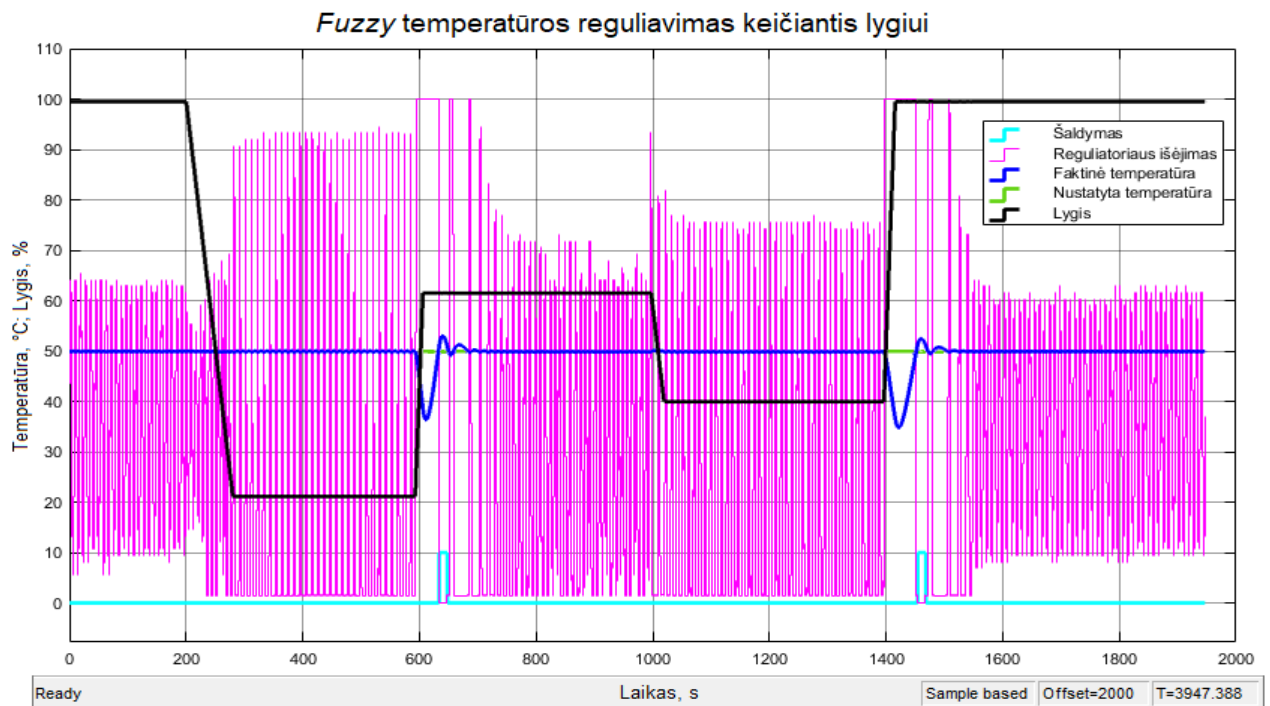
Eksperimentai su *fuzzy* reguliatoriumi (žr. 3.2.7 pav.) atliekami ekvivalentiškai kaip ir su PI diskretiniu reguliatoriumi, temperatūra keičiama keturis kartus ir po kiekvieno pakeitimo stebima sistemos reakcija į pokytį. Pirmuoju bandymu temperatūra sukeliama iki $50,40\text{ }^{\circ}\text{C}$, nuostatos signalą temperatūra pasiekia per $62,98\text{ s}$, su $4,12\%$ perreguliavimu ir nusistovi be švytavimų ir statinės paklaidos. Antruoju bandymu temperatūra nuleidžiama iki $30,62\text{ }^{\circ}\text{C}$, temperatūrą sistema pasiekia per $54,23\text{ s}$, su $2,29\%$ perreguliavimu. Trečiu atveju temperatūra pakeliama iki $60,22\text{ }^{\circ}\text{C}$, kurią

pasiekia per 48,66 s su 3,9 % perreguliuavimu ir ketvirtuoju bandymu temperatūra numetama iki 50,68 °C, kurią sistema pasiekia per 21,49 s su 1,7 % perreguliuavimu. Visais atvejais temperatūra palaiko tik darinėjant šildymo sklendę, šaldymas įjungiamas kaip siekiama sumažinti temperatūrą.



3.2.7 pav. Eksperimentiniai rezultatai su *fuzzy* regulatoriumi keičiantis temperatūros nuostatos signalui

Toliau atliekamas stebėjimas kaip sistema reaguoja į lygio nuostatos pakeitimą (žr. 3.2.8 pav.). Pirmu atveju lygis sumažinamas iki 21,98 %, temperatūros kreivėje pokyčiai nepastebimi, po 342,11 s lygis pakeliamas iki 63,66 %, temperatūra nukrenta 37,64 °C ir atsistato per 26,47 s su 3,27 % perreguliuavimu. Trečiuoju atveju lygis sumažinamas iki 39,99 %, kaip ir pirmu atveju pokyčių temperatūroje nematyti ir paskutiniuoju atveju lygis atstatomas į pradinę 100 % vertę, temperatūra nukrenta iki 35,29 °C ir vėl atsistato iki nuostatos signalo per 51,63 s su 2,7 % perreguliuavimu.



3.2.8 pav. Eksperimentiniai rezultatai su *fuzzy* reguliatoriumi keičiantis lygio nuostatos signalui

3.3. Modeliavimo rezultatai

Atlikus eksperimentus su PI ir *fuzzy* reguliatoriais, rezultatai parodė, kad sistema valdoma su PI reguliatoriumi yra 3 kartus lėtesnė, negu su *fuzzy* reguliatoriumi. Tačiau sistemoje su *fuzzy* reguliatoriumi keičiantis nuostato signalui atsiranda didesni perreguliavimai, bet abiejose sistemose jie neviršija 5 % ribos. Taip pat pastebėta, kad valdant su PI reguliatoriumi sistemoje ilgiau išlieka nežymus temperatūros svyravimai, ko sistemoje su *fuzzy* reguliatoriumi pavyksta išvengti. Tačiau šie svyravimai laikui bėgant dingsta, nes nusistovi PI reguliatoriaus išėjimo signalas. Sistemoje su PI reguliatoriumi dažniau veikia šaldymas, dėl ilgiau trukusiančio mažėjančio reguliatoriaus išėjimo signalo, tuo tarpu *fuzzy* reguliatorius susitvarko su temperatūros palaikymu be šaldymo.

Išvados

1. Remiantis literatūros šaltiniais, atlikta *Beckhoff CX2040* programuojamo loginio valdiklio, bei *TwinCAT 3* programinės įrangos galimybės dirbti kartu su *MATLAB/SIMULINK* platforma analizė parodė, kad:
 - *TwinCAT 3* programinė įranga turi vieną iš geriausiai išplėtotų įrankių vykdyti *Simulink* platformoje sukurtus modelius lyginant su kitais PLV gamintojais.
 - *Simulink* modelis ir jame sukonfigūruoti parametrai yra identiški perkelti į *TwinCAT 3* be jokių struktūros praradimų modelio konversijos metu. *TwinCAT 3* leidžia keisti modelio parametrus programos vykdymo metu, jeigu eksperimento metu yra matoma, kad pakeitimai yra būtini.
 - *TwinCAT 3* programinėje įrangoje modelio struktūros pakeitimai yra negalimi, tai galima padaryti *Simulink* platformoje ko pasėkoje reikia iš naujo atlikti modelio konversiją į *TwinCAT 3* programinę įrangą.
2. Atliktas programinės įrangos konfigūravimas, t.y. sistemos paruošimas konversijai iš *MATLAB/SIMULINK* platformos į *TwinCAT 3* programinę įrangą, leidžia teigti, kad:
 - norint *Simulink* modelį perversi į kodą suprantamą *TwinCAT 3* programinei įrangai, reikia suinstaliuoti papildomą programinę įrangą - *Visual Studio 2015*, kurią palaiko *MATLAB/SIMULINK 2018a*, bei *Windows Driver Kit*.
 - svarbu pridėti reikalingus sertifikatus bei licencijas. Visa tai turi būti atlikta eilės tvarka, nes kitu atveju konversijos nėra įmanoma atlikti.
3. Reaktoriaus šilumos mainų valdymui suprojektuoti dviejų tipų reguliatoriai: PI ir *fuzzy*. Lygio reaktoriuje palaikymo programa buvo sukurta *TwinCAT 3* programinėje įrangoje, dėl to, kad buvo naudojama *TwinCAT TE1400 Target for MATLAB/SIMULINK demo* versija.
4. Atliekant eksperimentinius bandymus su sukurtu PI diskretiniu reguliatoriumi gauti rezultatai parodė, kad *Simulink* platformoje sukurtas modelis sėkmingai susitvarko su realios sistemos temperatūros reguliavimu be žymaus perreguliavimo iki 1,99 %, tačiau temperatūros palaikymo metu atsiranda svyravimai kurie neviršija 1,12 % ribos, šie svyravimai dingsta nusistovėjus reguliatoriaus išėjimui per 3765 s ir sistema veikia tik įsijungus šildymui be šaldymo. Valdymo sistema taip pat sėkmingai reaguoja su lygio kitimu talpoje, temperatūrai neviršijant 3,2 % perreguliavimo.
5. Lyginant teorinio ir praktinio modelių rezultatus pastebėta, kad temperatūros pereinamasis procesas 200 s sutampa abiejuose eksperimentuose, tačiau reguliatoriaus išėjimas ženkliai skiriasi, teoriniame modeliavime išėjimas nusistovi ties 7,58 %, o praktiniame eksperimente išėjimo signalas yra didesnis – 37,42 %.
6. Atlikus bandymus su PI reguliatoriumi buvo atliekamas eksperimentas su *fuzzy* reguliatoriumi tomis pačiomis sąlygomis. Gauti rezultatai parodė, kad *fuzzy* reguliatorius pagreitina sistemos reakciją 3 kartus į temperatūros nuostato signalo bei lygio pokyčius, tačiau tai iššaukia didesnę perreguliavimą iki 4,12%, bet panaikina švytavimus proceso eigoje.

Literatūros sąrašas

1. *Continuous Function Chart (CFC)* [Interaktyvus] [žiūrėta 2021-02-05]. Prieiga per internetą: https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/2526723211.html&id=
2. *NOVRAM* [Interaktyvus] [žiūrėta 2021-02-05]. Prieiga per internetą: https://infosys.beckhoff.com/english.php?content=../content/1033/cx9020_hw/34562841713613782283.html&id=
3. *What is DVI?* [Interaktyvus] [žiūrėta 2021-02-05]. Prieiga per internetą: https://www.datapro.net/techinfo/dvi_info.html
4. *Steve Larrivee. What is a CFast SSD?* (2015) [Interaktyvus] [žiūrėta 2021-02-05]. Prieiga per internetą: <https://www.cactus-tech.com/resources/blog/details/what-is-a-cfast-ssd/>
5. *Robert McFarlane, Shen Milsom, Wilke LLC. Uninterruptible power supply (UPS)* (2019) [Interaktyvus] [žiūrėta 2021-02-05]. Prieiga per internetą: <https://searchdatacenter.techtarget.com/definition/uninterruptible-power-supply>
6. *IEC 61131-3* (2013) [Interaktyvus] [žiūrėta 2021-02-05]. Prieiga per internetą: <https://webstore.iec.ch/publication/4552>
7. *What Is Ethernet?* [Interaktyvus] [žiūrėta 2021-02-05]. Prieiga per internetą: <https://www.linksys.com/us/r/resource-center/basics/whats-ethernet/>
8. *Mondi Anderson (2019). WHAT IS FIELDBUS?* [Interaktyvus] [žiūrėta 2021-02-05]. Prieiga per internetą: <https://realpars.com/fieldbus/>
9. *Edouard Tisserant, Laurent Bessard, Mario de Sousa (2007). An Open Source IEC 61131-3 Integrated Development Environment* [Interaktyvus] [žiūrėta 2021-02-15]. Prieiga per internetą: <https://ieeexplore.ieee.org/document/4384753>
10. *Beckhoff Automation* [Interaktyvus] [žiūrėta 2021-02-19]. Prieiga per internetą: <https://www.beckhoff.com/en-us/company/>
11. *Scalable Industrial PC solutions* [Interaktyvus] [žiūrėta 2021-02-19]. Prieiga per internetą: <https://www.beckhoff.com/en-us/products/ipc/>
12. *Fieldbus components for all common I/Os and fieldbus systems* [Interaktyvus] [žiūrėta 2021-02-19]. Prieiga per internetą: https://www.beckhoff.com/en-us/products/i-o/#text_bild_1
13. *Embedded PC Modular DIN rail IPCs and Industrial Motherboards* [Interaktyvus] [žiūrėta 2021-02-19]. Prieiga per internetą: https://download.beckhoff.com/download/document/Catalog/Main_Catalog/english/Beckhoff_Embedded-PC-CX.pdf
14. *CX2040 | Basic CPU module* [Interaktyvus] [žiūrėta 2021-03-01]. Prieiga per internetą: <https://www.beckhoff.com/en-us/products/ipc/embedded-pcs/cx20x0-intel-celeron-core-i7/cx2040.html>
15. *New Features TwinCAT 3.1* [Interaktyvus] [žiūrėta 2021-03-02]. Prieiga per internetą: <https://www.beckhoff.com/en-us/products/automation/twincat/twincat-3-build-4024/>
16. *What Is LabVIEW?* [Interaktyvus] [žiūrėta 2021-03-15]. Prieiga per internetą: <https://www.ni.com/en-us/shop/labview.html>
17. *Simulation and Model-Based Design* [Interaktyvus] [žiūrėta 2021-03-15]. Prieiga per internetą: https://uk.mathworks.com/products/simulink.html?s_tid=hp_ff_p_simulink

18. G. W. Johnson (1997), *LabVIEW graphical programming: practical applications in instrumentation and control*, McGraw-Hill.
19. Lina Liu, Jian Hu (2012). *The practice of MATLAB simulation in modern control theory course teaching* [Interaktyvus] [žiūrėta 2021-03-15]. Prieiga per internetą: <https://ieeexplore.ieee.org/document/6463300>
20. *TC3 Target for Matlab®/Simulink®* (2019) [Interaktyvus] [žiūrėta 2021-03-16]. Prieiga per internetą: https://download.beckhoff.com/download/document/automation/twincat3/TE1400_TC3_Target_Matlab_EN.pdf
21. *Paveikslas TwinCAT 3 eXtended Automation Engineering Environment (XAE) allows the integration of MATLAB and Simulink* [Interaktyvus] [žiūrėta 2021-03-25]. Prieiga per internetą: <https://www.researchgate.net/profile/Bram-Vanderborght/publication/323718831/figure/fig4/AS:606177935237120@1521535490636/TwinCAT-3-eXtended-Automation-Engineering-Environment-XAE-allows-the-integration-of.png>
22. *Gülden Bayrak, Patrik J. Murr, Sebastian Ulewicz, Birgit Vogel-Heuser (2012). Comparison of a transformed Matlab/Simulink model into the programming language CFC on different IEC 61131-3 PLC environments.* Krokuva, Lenkija [Interaktyvus] [žiūrėta 2021-03-25]. Prieiga per internetą: <https://ieeexplore.ieee.org/document/6489667>
23. *Paveikslas Comparison of the designed simulation model in Simulink (on the left side) with the generated CFC model in TwinCAT (on the right side).* *Gülden Bayrak, Patrik J. Murr, Sebastian Ulewicz, Birgit Vogel-Heuser.* Miūncheno technikos universitetas. *Comparison of a transformed Matlab/Simulink model into the programming language CFC on different IEC 61131-3 PLC environments.* 6 p.
24. L. Champine, P. Gahinet (2004). *Software for modeling and analysis of linear systems with delays.* *Proceedings of the American Control Conference.* 5600-5605
25. *Paveikslas Results of the simulation for the speed control model in Simulink, TwinCAT, PCS7 and PC Worx. The development of the speed is shown over an interval of 200 milliseconds for all four environments.* *Gülden Bayrak, Patrik J. Murr, Sebastian Ulewicz, Birgit Vogel-Heuser.* Miūncheno technikos universitetas. *Comparison of a transformed Matlab/Simulink model into the programming language CFC on different IEC 61131-3 PLC environments.* 7 p.
26. B. Levinson (1947). *The wiener rms (root mean square) error criterion in filter design and prediction.* Norman Levinson
27. Leonas Balaševičius, Napalys Ezerskis (2008). *Programuojamieji valdikliai. Laboratoriniai darbai.* Vilniaus pedagoginio universiteto leidykla. 103-118.
28. *Paveikslas CX2100-0014* [Interaktyvus] [žiūrėta 2021-03-25]. Prieiga per internetą: https://multimedia.beckhoff.com/media/cx2100-0014__web.jpg
29. *CX2100-0014 | Power supply unit for CX20xx, 130 W* [žiūrėta 2021-03-27]. Prieiga per internetą: <https://www.beckhoff.com/en-us/products/ipc/embedded-pcs/cx20x0-intel-celeron-core-i7/cx2100-0014.html>
30. *Paveikslas PhoenixContact Power UNO* [Interaktyvus] [žiūrėta 2021-03-25]. Prieiga per internetą: https://www.distrelec.lt/Web/WebShopImages/portrait_medium/3-/01/phoenix-contact-uno-ps-1ac-24dc-60w.jpg
31. *Power supply - UNO-PS/1AC/24DC/ 60W* [žiūrėta 2021-03-27]. Prieiga per internetą: <https://www.phoenixcontact.com/en-in/products/power-supply-uno-ps1ac24dc-60w-2902992>

32. Paveikslas *EL2809* [Interaktyvus] [žiūrėta 2021-03-25]. Prieiga per internetą: https://multimedia.beckhoff.com/media/el2809__web_preview.png
33. *EL2809 | HD EtherCAT Terminal, 16-channel digital output 24 V DC, 0.5 A* [žiūrėta 2021-03-27]. Prieiga per internetą: <https://www.beckhoff.com/en-us/products/i-o/ethercat-terminals/el2xxx-digital-output/el2809.html>
34. Paveikslas *EL3104* [Interaktyvus] [žiūrėta 2021-03-25]. Prieiga per internetą: https://multimedia.beckhoff.com/media/el3104__web_preview.png
35. *EL3104 | 4-channel analog input terminal -10...+10 V* [žiūrėta 2021-03-27]. Prieiga per internetą: https://download.beckhoff.com/download/Document/Catalog/Main_Catalog/english/separate-pages/EtherCAT/EL3104.pdf
36. Paveikslas *EL4004* [Interaktyvus] [žiūrėta 2021-03-25]. Prieiga per internetą: https://multimedia.beckhoff.com/media/el4004_es4004__web_preview.png
37. *EL4004 | 4-channel analog output terminal 0...10 V, 12 bit* [žiūrėta 2021-03-27]. Prieiga per internetą: https://download.beckhoff.com/download/Document/Catalog/Main_Catalog/english/separate-pages/EtherCAT/EL4004.pdf
38. Kevin Langlois, Tom van der Hoevena, David Rodriguez Ciancaa, Tom Verstratena, Tomislav Baceka, Bryan Convensa, Carlos Rodriguez, Guerreroa, Victor Grosua, Dirk Lefeberra and Bram Vanderborghta (2018). *EtherCAT Tutorial, an introduction for real-time hardware communication on Windows* [žiūrėta 2021-02-01]. Prieiga per internetą: <https://ieeexplore.ieee.org/document/8314597>
39. Norman S. Nise. *Control Systems Engineering*. Kalifornijos valstijos politechnikos universitetas. Pomona. 766-769.
40. Paveikslas *Graphical definition of MFs for fuzzy variables, $e(t)$, $\int e(t) e'(t)$ ir $u(t)$* . Chun-Tang Chao, Nana Sutarna, Juing-Shian Chiou, Chi-Jo Wang (2019). *An Optimal Fuzzy PID Controller Design Based on Conventional PID Control and Nonlinear Factors*. Elektros inžinerijos fakultetas. Pietų Taivano mokslo ir technologijų universitetas. 3 p.
41. Chun-Tang Chao, Nana Sutarna, Juing-Shian Chiou, Chi-Jo Wang (2019). *An Optimal Fuzzy PID Controller Design Based on Conventional PID Control and Nonlinear Factors*. Elektros inžinerijos fakultetas. Pietų Taivano mokslo ir technologijų universitetas [žiūrėta 2021-04-20]. Prieiga per internetą: <https://www.mdpi.com/2076-3417/9/6/1224/htm>
42. Paveikslas *Sliced cube fuzzy associative memory (FAM) representation of the knowledge base*. Chun-Tang Chao, Nana Sutarna, Juing-Shian Chiou, Chi-Jo Wang (2019). *An Optimal Fuzzy PID Controller Design Based on Conventional PID Control and Nonlinear Factors*. Elektros inžinerijos fakultetas. Pietų Taivano mokslo ir technologijų universitetas. 4 p.