



KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS

Irmantas Girdauskas

ARDUINO MIKROVALDIKLIO PLOKŠTĖS PANAUDOJIMO
PROCESŲ VALDYMUI GALIMYBIŲ TYRIMAS

Baigiamasis magistro projektas

Vadovas

Prof. dr. Rimvydas Simutis

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS
AUTOMATIKOS KATEDRA

ARDUINO MIKROVALDIKLIO PLOKŠTĖS PANAUDOJIMO
PROCESŲ VALDYMUI GALIMYBIŲ TYRIMAS

Baigiamasis magistro projektas
Valdymo technologijos (621H66001)

Vadovas

Prof. dr. Rimvydas Simutis

Recenzentas

Doc. dr. Kastytis Ratkevičius

Projektą atliko

Irmantas Girdauskas

KAUNAS, 2016



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Elektros ir elektronikos fakultetas

Irmantas Girdauskas

Valdymo technologijos (621H66001)

Baigiamojo projekto „Arduino mikrovaldiklio plokštės panaudojimo procesų valdymui galimybių tyrimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 ____ m. _____ d.
Kaunas

Patvirtinu, kad mano **Irmanto Girdausko** baigiamasis projektas tema „Arduino mikrovaldiklio plokštės panaudojimo procesų valdymui galimybių tyrimas“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

(vardą ir pavardę įrašyti ranka)

(parašas)

Girdauskas Irmantas, Arduino mikrovaldiklio plokštės panaudojimo procesų valdymui galimybių tyrimas. Valdymo sistemų magistro baigiamasis projektas / vadovas prof. dr. Rimvydas Simutis; Kauno technologijos universitetas, elektros ir elektronikos fakultetas, automatikos katedra.

Kaunas, 2016. 51 psl.

SANTRAUKA

Šiame magistro darbe yra atliekamas Arduino mikrovaldiklio plokštės pritaikomumo procesų valdyje tyrimas. Išaiškinama kokia yra mikrovaldiklių paskirtis, kaip jie klasifikuojami, kur taikomi. Darbe tiriamos Arduino mikrovaldiklio plokštės susiejimo su Matlab programavimo terpė galimybės. Grafikų vaizdavimui, programinio kodo kūrimui naudojama Matlab programinė terpė. Taip pat tiriamos PID reguliatoriaus realizavimo galimybės naudojant Arduino mikrovaldiklio plokštę. Pirmiausiai bandymai atliekami su eksperimentams skirtu stendu, po to su fiziniu objektu – šiltnamio modeliu. Po to darbe tiriamos automatinio derinimo galimybės ir efektyvumas.

Išanalizavus reguliatoriaus ir automatinio derinimo galimybes, sistema palyginama su kitomis valdymo alternatyvomis.

Reikšminiai žodžiai: Arduino, mikrovaldiklis, reguliatorius, šuolinė reakcija, fizinis modelis, automatinis derinimas.

Girdauskas Irmantas. Feasibility studies of Arduino Microcontroller for Applications in Process Control. Final project of control systems masters degree / supervisor prof. dr. Rimvydas Simutis; Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, department of Automation.

Kaunas, 2016. 51 p.

SUMMARY

In this final work of master thesis the analysis of Arduino microcontroller board application in process control is carried out. The purpose of microcontrollers, their classification and appliance is described. In this paper compliance capabilities of the Arduino microcontroller board and Matlab programming medium is investigated. Matlab programming medium is used for programming and Graphics display. Moreover, the possibility of PID controller realization using Arduino microcontroller board is investigated. In the beginning the tests are carried out with the experimental stand, then with the physical object - the greenhouse model. Also, the autotuning capabilities and efficiency is investigated.

After the regulator and autotuning possibilities are analysed, the system comparison with other control alternatives is submitted.

Keywords: Arduino, microcontroller, regulator, step response, physical model, autotune.

Turinys

1. Įvadas	8
2. Mikrovaldikliai ir jų pritaikymo sritys	9
2.1 Bazinė mikrovaldiklių sandara	9
2.2 Mikrovaldiklių skirstymas	9
2.3 Mikrovaldiklių taikymo sritys	11
3. Arduino mikrovaldiklio plokštė ir jos taikymas procesų valdymui	13
3.1 Arduino mikrovaldiklio plokštės sandara	13
3.2 Arduino mikrovaldiklio platformos I/O.....	14
3.3 Analoginio signalo nuskaitymas	15
3.4 PID regulatoriaus realizavimas su Arduino mikrovaldiklio plokšte.....	17
3.4.1 Reakcijos į nuostatos keitimą nustatymas	17
3.4.2 Reakcijos į trikdį nustatymas.....	19
3.5 Duomenų nuskaitymas iš Arduino mikrovaldiklio plokštės.....	21
3.6 Parametrų nuskaitymas naudojantis Matlab programine aplinka	21
4. Arduino mikrovaldiklio plokštės ir Matlab programinės aplinkos panaudojimas procesų valdyje	23
4.1 Duomenų mainai tarp Matlab programinės terpės ir Arduino mikrovaldiklio plokštės ...	23
4.2 Regulatoriaus realizavimas Matlab programinėje terpeje	23
4.3 Automatinis regulatoriaus parametrų derinimas.....	24
5. Šiltnamio fizinis modelis ir jo valdymas.....	29
5.1 Mikroklimato palaikymas	29
5.2 Tiriamasis objektas	29
5.3 Duomenų filtravimas	30
5.4 Temperatūros reguliavimas.....	31
5.4.1 Elektrinė šildymo sistemos schema.....	31
5.4.2 Temperatūros reguliavimas	32
5.4.3 Valdymas poziciniu reguliatoriumi	34
5.4.4 Reguliatorių valdymo rezultatai	36
5.4.5 Reguliavimo įtaisų palyginimas	39
5.5 Automatinis regulatoriaus derinimas	40
6. Valdymo sistemos, paremtos Arduino mikrovaldiklio plokšte, sąmata.....	44
7. Rezultatai ir išvados	45
8. Informacijos šaltinių sąrašas	46
9. Priedai.....	48

Santrumpų ir ženklų aiškinimo žodynas

RAM (angl. Random-access memory) - kompiuterio operatyvioji atmintis;

ROM (angl. Read-only memory) - specialus atmintinės tipas, iš kurio galima duomenis tik skaityti;

ADC – analoginis - skaitmeninis keitiklis;

DAC – skaitmeninis - analoginis keitiklis;

CISC (angl. Complex Instruction Set Computer) - grupė įvairių procesorių architektūrų;

RISC (angl. Reduced Instruction Set Computer) - centrinių procesorių architektūra.

1. Įvadas

Procesų valdymas tradiciškai yra atliekamas pramoniniais valdikliais, tačiau šios technologijos yra sudėtingos ir brangios. Šiuo metu rinkoje populiarėja nebrangios mikrovaldiklių platformos, kurias galima pritaikyti procesų valdymui. Arduino yra viena populiariausių tokių mikrovaldiklio platformų gamintojų.

Šio darbo tikslas yra ištirti Arduino mikrovaldiklio plokštės pritaikomumą procesų valdymui ir apžvelgti jo privalumus palyginus su kitomis procesų valdymo alternatyvomis.

Darbo uždaviniai yra trumpai apžvelgti Arduino ir kitų mikrovaldiklių platformų funkcijas, galimybes, tipus ir panaudojimo sritis. Ištirti Arduino mikrovaldiklio plokštės pritaikomumą procesų valdymui. Gilintis į Arduino mikrovaldiklio plokštės ir Matlab programinės terpės susiejimo galimybes, t.y. duomenų mainų, grafinio atvaizdavimo realiu laiku galimybes. Ištirti PID reguliatoriaus realizavimo su Arduino mikrovaldiklio plokšte galimybes, automatinio reguliatoriaus derinimo galimybes ir efektyvumą. Atlikti valdymo sistemos bandymus su sukonstruotu fiziniu objektu - šiltnamio modeliu. Pateikti atliktų eksperimentų grafinius duomenis, rezultatus, programų kodus, elektrinių jungimų schemas, bei palyginti Arduino mikrovaldiklio plokštę su kitomis alternatyvomis procesų valdymui.

2. Mikrovaldikliai ir jų pritaikymo sritys

2.1 Bazinė mikrovaldiklių sandara

Mikrovaldikliai dažniausiai turi tokias sudedamąsias dalis:

- Procesorius;
- Atmintis;
- I/O sąsajos;
- Laikmačiai;
- ADC ir DAC;

1. Procesorius (angl. central processing unit, CPU) – įtaisas apdorojantis duomenų srautą. Procesorius atlieka aritmetines, logines, valdymo, išėjimų/įėjimų kontroliavimo funkcijas. Procesoriaus sąvoka yra bendrinė, reiškianti abstrakčią informacinių sistemų rūšį, atliekančią manipuliacijas su duomenimis, tačiau dažniau naudojama, kalbant apie skaičiavimams skirtus procesorius, realizuotus, aparatinių įrenginių pavidalu (dažniausiai – mikroprocesoriai) ar programas, sugebančias vykdyti tam tikras komandų sekas. „Arduino“ mikrovaldiklio plokštėse šią funkciją atlieka Atmel AVR mikroprocesoriai.

2. Atmintyje talpinamos programos ir duomenys. Procesorius turi tiesioginę prieigą prie atminties, nuskaitęs duomenis iš jos, procesorius įvykdo skaičiavimus ir kitas komandas. Mikrovaldikliai turi RAM ir ROM tipų atmintį.

3. I/O sąsajos naudojamos procesoriui komunikuoti su kitais įrenginiais, pavyzdžiui, kompiuteriu ar jutikliais ir vartotoju.

4. Laikmačiai ir skaitikliai reikalingi impulsų, ateinančių iš išorės, skaičiavimui ir impulsų generavimui. Nuo impulsų generavimo dažnio priklauso procesoriaus dažnis, dažnį dažniausiai palaiko išorinė impulsų generavimo grandinė.

5. Analoginis-skaitmeninis keitiklis (ADC) ir skaitmeninis-analoginis keitiklis (DAC) reikalingas konvertuoti skirtingų tipų signalus iš analoginio į skaitmeninį ir atvirkščiai [1].

2.2 Mikrovaldiklių skirstymas

Mikrovaldikliai gali būti skirstomi pagal bitinę struktūrą, mikrovaldikliai gali būti:

- 8 bitų;
- 16 bitų;
- 32 bitų;
- 64 bitų;
- 128 bitų.

8 bitų struktūra pagrįsti mikrovaldikliai turi 8 bitų duomenų kanalą ir turi prieigą prie 8 bitų duomenų vieno ciklo metu. Tokie mikrovaldikliai gali atlikti logines ir aritmetines operacijas. 8 bitų struktūra naudojama tokiuose mikrovaldikliuose kaip Intel 8031/8051, ATmega 128/328.

16 bitų mikrovaldikliai gali prieiti prie 16 bitų duomenų vieno ciklo metu, palyginus su 8 bitų mikrovaldikliais jie pasižymi didesniu pajėgumu ir tikslumu. 16 bitų struktūra naudojama Intel 8096, IBM 1130 mikrovaldikliuose.

32 bitų mikrovaldikliai gali prieiti prie duomenų kurie turi iki 32 bitų dydžio. Šie mikrovaldikliai naudojami sudėtingesniuose įrenginiuose: biuro įrenginiuose, medicininiuose aparatuose.

Šiuo metu yra prienami ir mikrovaldikliai su 4, 64 ar 128 bitų architektūra. 64 ir 128 bitų architektūros labiau yra naudojamos mikroprocesoriuose, tačiau literatūroje aiškios ribos tarp mikroprocesorių ir mikrovaldiklių sąvokų nėra [1].

Pagal atminties tipą mikrovaldikliai skirstomi:

- Turintys vidinę atmintį;
- Turintys išorinę atmintį;

Vidinę atmintį turintys mikrovaldikliai turi visus reikalingus funkcinius blokus, kaip atmintį, I/O prievadus, laikmačius ir skaitiklius viename luste. Išorinę atmintį turintys mikrovaldikliai neturi visų blokų viename luste. Mikrovaldikliai Intel 8051 turi vidinę atmintį, o pavyzdžiui Intel 8031 neturi.

Pagal instrukcijų rinkinį mikrovaldikliai skirstomi:

- CISC
- RISC

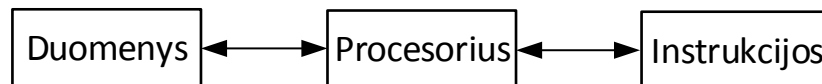
CISC procesoriai turi sudėtingą komandų rinkinį, yra orientuoti į efektyvesnį atminties panaudojimą, patogesnę programavimą. Tipiškuose CISC procesoriuose būna dešimtys ar net šimtai skirtingų komandų, daugelis komandų būna sudėtingos, atitinkančios ilgą seką aritmetinių veiksmų (pvz., šaknies traukimo ar kėlimo laipsniu komandos).

RISC procesoriai yra optimizuoti dideliame našumui, jų komandų rinkinys būna labai mažas (kai kada – mažiau, nei 10 komandų). Kai kurie RISC vadinami procesoriai turi gana didelį komandų skaičių, kelis adresacijos metodus, todėl laikytini tarpiniais tarp RISC ir CISC. Siekiant padidinti įprastų CISC procesorių našumą, jie dažnai turi RISC branduolį, kuris vykdo mikroprogramas, emuliuojančias CISC procesoriaus darbą [2].

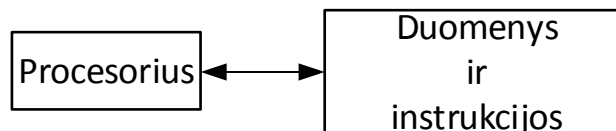
Pagal atminties architektūrą mikrovaldikliai skirstomi:

- Harvardo architektūra;
- Prinsono architektūra;

Kai adresai ir duomenys naudoja skirtingas atminties sistemas tada naudojama Harvardo atminties architektūra. Kada naudojama Prinsono architektūra (arba dar vadinama Von Neuman'o modeliu) tada ta pati atminties sistema saugo ir duomenis ir instrukcijas.



1 pav. Harvardo architektūra



2 pav. Prinsono architektūra

2.3 Mikrovaldiklių taikymo sritys

Mikrovaldikliai naudojami kaip siauros paskirties įterptinis elementas kituose įtaisuose, lyginant su mikroprocesoriais kurie naudojami namų kompiuteriuose. Mikrovaldikliai gali būti naudojami:

- namų elektronikoje;
- automobilių variklių valdymo sistemose ir automatikoje;
- industrinėse valdymo sistemose;
- matavimo įtaisuose;
- medicinoje;
- robotikoje;
- žaisluose.



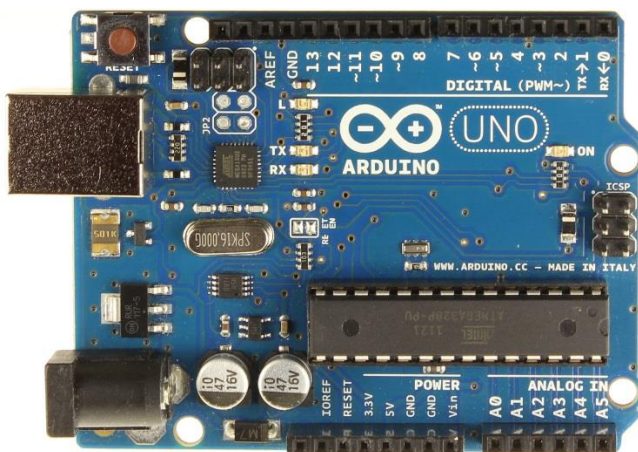
3 pav. Mikrovaldiklių taikymo sritys

Mikrovaldiklių naudojimas auga dėl jų nedidelės kainos ir pakeičiamumo. Arduino UNO mikrovaldiklio plokštė naudoja Atmega328 mikrovaldiklį.

3. Arduino mikrovaldiklio plokštė ir jos taikymas procesų valdymui

3.1 Arduino mikrovaldiklio plokštės sandara

Arduino – tai skaičiavimo platforma paremta mikrovaldiklio plokšte, turinti eilę analoginių bei diskretinių įėjimų ir išėjimų ir galinti atlikti įprasto valdiklio funkcijas. Arduino mikrovaldiklio plokštė programuojama C ir C++ pagrindais paremta programavimo kalba ir yra atviro kodo (open-source). Arduino kuria nemažai skirtingų mikrovaldiklio plokščių besiskiriančių savo pajėgumu, paskirtimi ir kaina, šiame darbe dėmesys sutelkiamas į Arduino Uno R3 mikrovaldiklio plokštę (4 pav.).



4 pav. Arduino Uno R3 mikrovaldiklio plokštė

Arduino Uno R3 mikrovaldiklio plokštės parametrai [3]:

- mikrovaldiklis: ATmega328;
- diskretinių įėjimų/išėjimų skaičius: 14, iš kurių 6 galima naudoti kaip analoginius išėjimus;
- analoginių įėjimų skaičius: 6;
- įtampa viename atvade: 5V;
- srovė viename atvade: 40 mA;
- srovė 3,3V atvade: 50mA
- maitinimo įtampa: 7-12V;
- atmintis: 32 KB;
- SRAM: 2 KB;
- EEPROM: 1 KB;

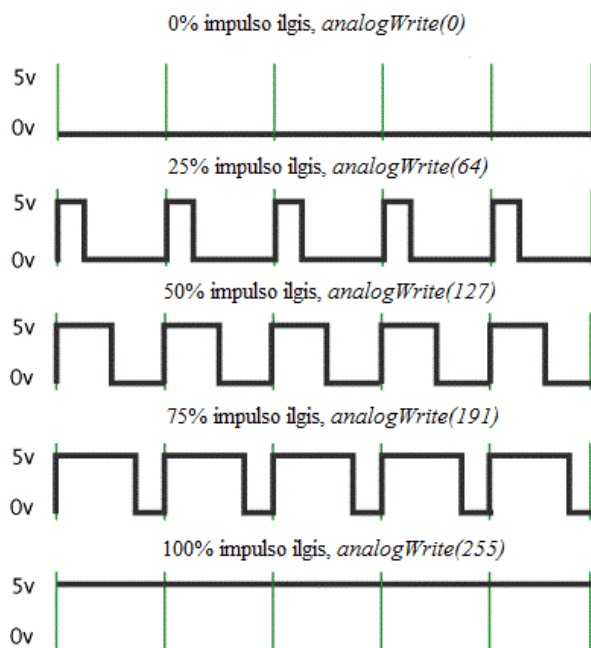
- dažnis: 16 MHz;
- 6 analoginiai įėjimai.

Yra ne viena platforma atliekanti panašias funkcijas kaip ir Arduino mikrovaldiklio plokštė. Ši mikrovaldiklio plokštė išsiskiria savo nedidele kaina, plačiu operacinių sistemų palaikymu, lanksčia programavimo aplinka [4]. Taip pat visi komponentai ir techninė įranga, įeinanti į Arduino mikrovaldiklio plokštę ir jo išplėtimo modulius yra atviro kodo, todėl šį valdiklį galima pasidaryti ir pačiam. Kitos panašios platformos:

- Parallax Basic Stamp;
- Netmedia's BX-24;
- Phidgets;
- Raspberry Pi.

3.2 Arduino mikrovaldiklio platformos I/O

Arduino mikrovaldiklio plokštė turi 14 diskretinių išėjimų iš kurių 6 galima naudoti kaip analoginius išėjimus. Kiekvienas diskretinis išėjimas turi 5 V įtampą ir 40 mA srovę. Analoginiai išėjimai realizuojami naudojant impulso pločio moduliaciją, taigi galima galime reguliuoti analoginį išėjimą režiuose 0 – 5 V keisdami diskretinio signalo impulso ilgį. Arduino mikrovaldiklio plokštės impulso pločio modulatorius turi apie 500 Hz dažnį, taigi ciklas trunka 2 ms. Analoginio signalo reikšmei sukurti viename iš išėjimų naudojame komandą *analogWrite()*. Impulso pločio reikšmė gali būti nuo 0 iki 255, taigi *analogWrite(250)* reiškia, kad išėjime 5 V impulso ilgis sudaro 100% ciklo, arba kuria 5 V analoginį signalą (5 pav.), *analogWrite(127)* reiškia, kad impulso ilgis sudaro 50% ciklo ir kuria 2,5 V analoginį signalą [5].



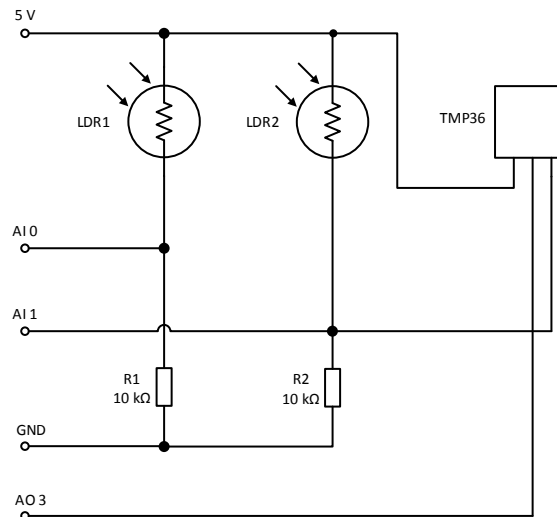
5 pav. Impulso pločio moduliacija

Atmega mikrovaldikliai, naudojami Arduino mikrovaldiklių plokštėse turi 6 analoginius įėjimus su 10 bitų analogas-kodas keitikliu. 10 bitų rezoliucija reiškia, kad įėjime paduodama 0 – 5 V įtampa gražina sveikąjį skaičių nuo 0 iki 1023, taigi analoginių įėjimų kvantavimo žingsnis yra 4,9 mV ($5 \text{ V} / 1024$). Analoginio signalo nuskaitymui naudojame komandą *analogRead(įėjimo Nr.)*.

Analoginio įėjimo nuskaitymas užtrunka apie 100 μs , taigi maksimaliai įėjimas nuskaitymas 10000 per sekundę [6].

3.3 Analoginio signalo nuskaitymas

Analoginio signalo nuskaitymui atliktas kambario temperatūros ir apšvietimo tyrimas. Apšvietimo matavimai atlikti pasitelkus fotorezistorių, temperatūrą matuota pasitelkus termovaržą. Fotorezistoriaus rodmenys tai santykinė reikšmė, nematuojama jokiais matavimo vienetais, tiesiog parodanti įėjimo reikšmę diapazone 0 – 1023 (žr. Arduino mikrovaldiklio plokštės I/O). Eksperimente naudojamas temperatūros jutiklis (TMP36) pasirinktas dėl veikimo esant tinkamai įtampai Arduino mikrovaldikliui, t. y. 2,7 - 5,5 V ir dėl to, kad nereikalinga kalibracija, jutiklis temperatūrą matuoja $^{\circ}\text{C}$ skalėje. Jungimo schema:



6 pav. Jungimo schema

1 lentelė. Naudojami komponentai

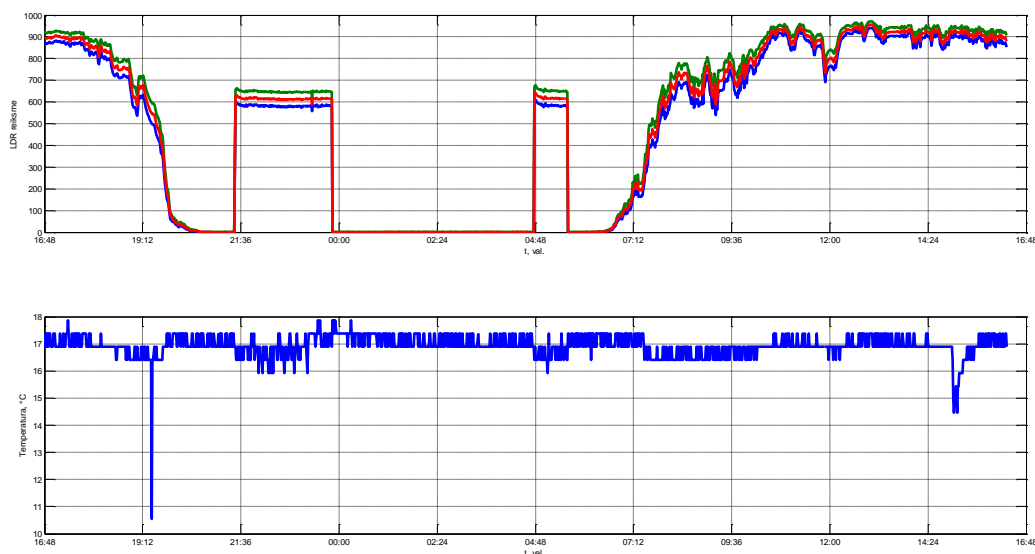
Komponentas	Žymėjimas	Detalės Nr.
Fotorezistorius	LDR1, LDR2	VT90N2
Rezistorius, 10 kΩ	R1, R2	281-10k/REEL-RC
Termorezistorius	TMP36	TMP36GZ

Matuojami dviejų fotorezistorių rodmenys, taip pat išvedamas vidurkis tarp jų rodmenų. Termorezistorius išėjime formuoja įtampą diapazone 0 – 5 V. Jutiklis fiksuoja iki -40 °C temperatūrą, tai sudaro 500 mV [7], skaičiuojant temperatūrą tai kompensuojama iš įtampos atėmus 0,5 . Temperatūra pagal termorezistoriaus parodymus apskaičiuojama taip [8]:

$$temperatūra = (įtampa - 0,5) \cdot 100;$$

$$įtampa = \left(\frac{jutiklio_parodymai}{1024} \right) \cdot 5.$$

Matavimai atliekami parą, jutiklių rodmenys nuskaitomi kas 1 minutę. Eksperimento kodas pateiktas 1 priede.



7 pav. Eksperimento rezultatai

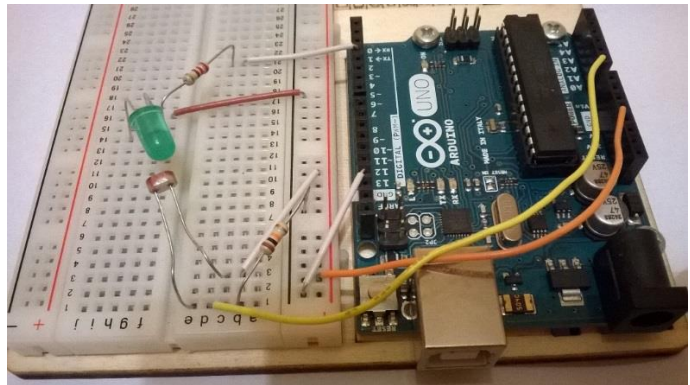
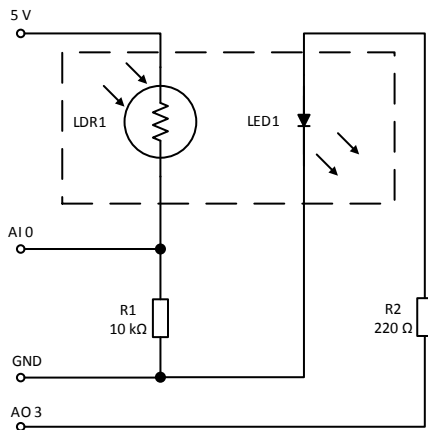
Atlikus eksperimentą (7 pav.) matyti, kad fotorezistorių rodmenys skiriasi dėl jutiklių paklaidų ir pozicijos skirtumų. Taip pat matyti, kad temperatūros jutiklio parodymai nėra tikslūs ir yra triukšmingi. Viena iš užduočių būtų pritaikyti žemo dažnio filtrą. Signalų filtravimui išbandyti RC grandinę ir skaitmeninį žemo dažnio filtrą.

3.4 PID reguliatoriaus realizavimas su Arduino mikrovaldiklio plokšte

PID reguliatoriaus realizavimui naudojama PID biblioteką [9] (kūrėjas Brett Beauregard). Arduino platformos PID biblioteka kaip ir standartinis PID reguliatorius turi palaikyti valdomo kintamojo reikšmę (nuostatą), keisdamas valdančio poveikio reikšmes. Paklaida tarp valdomo dydžio ir nuostatos reikšmės paduodama valdikliui, kuris pagal paklaidos dydį ir jos laikines charakteristikas formuoja valdantį poveikį procesui.

3.4.1 Reakcijos į nuostatos keitimą nustatymas

Reguliatoriaus išbandymui valdysime šviesos diodo LED1 ryškumą pagal fotorezistoriaus LDR1 apšviestumo rodmenis. Jungimo schema pavaizduota 8 pav.:



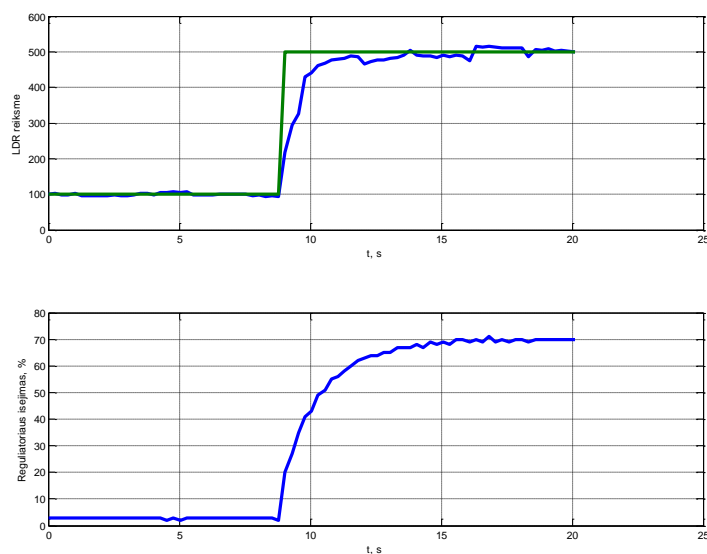
8 pav. Jungimo schema, realus eksperimentinis objektas

2 lentelė. Naudojami komponentai

Komponentas	Žymėjimas	Detalės Nr.
Fotorezistorius	LDR1	VT90N2
Rezistorius 220 Ω	R2	281-220/REEL-RC
Rezistorius 10 kΩ	R1	281-10k/REEL-RC
Fotodiodas	LED1	L7113GT

Tiriame sistemos reakciją į nuostatos keitimą. Fotorezistoriaus pradinė 100, po 8 sekundžių nuostata pasikeičia į 500. Reguliatoriaus parametrai: $K_p=0,1$; $K_i=0,6$; $K_d=0$. Reguliatoriaus ciklas 20 ms, stebėjimo duomenys atnaujinami kas 250 ms. Eksperimento kodas pateiktas 2 priede.

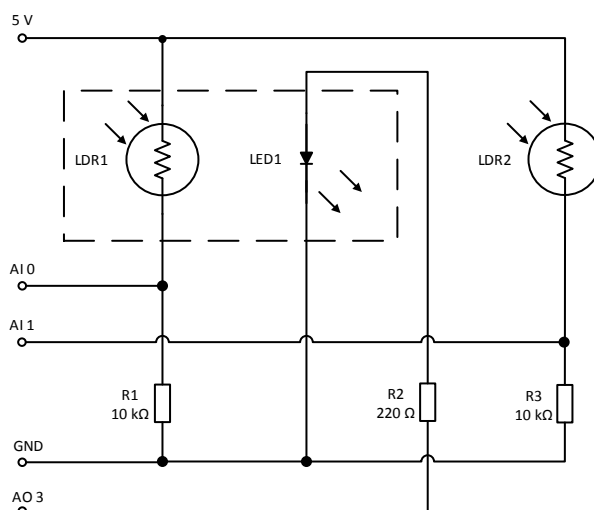
Stebėjimo duomenyse (9 pav.) atsispindi fotorezistoriaus parodymai keičiant nuostatą ir reguliatoriaus išėjimas procentais.



9 pav. Stebėjimo duomenys keičiant nuostatą

3.4.2 Reakcijos į trikdį nustatymas

Sistemos reakcijos į trikdį tyrimui prie esamos schemos pridedama papildomą fotorezistorių LDR2, kuris matuoja aplinkos apšvietumą. Taigi fotorezistoriai LDR1 ir LDR2 yra tose pačiose aplinkose, tačiau LDR1 yra veikiamas reguliuojamo šviesos diodo LED1, o LDR2 – neveikiamas. Jungimo schema pateikta 10 pav.

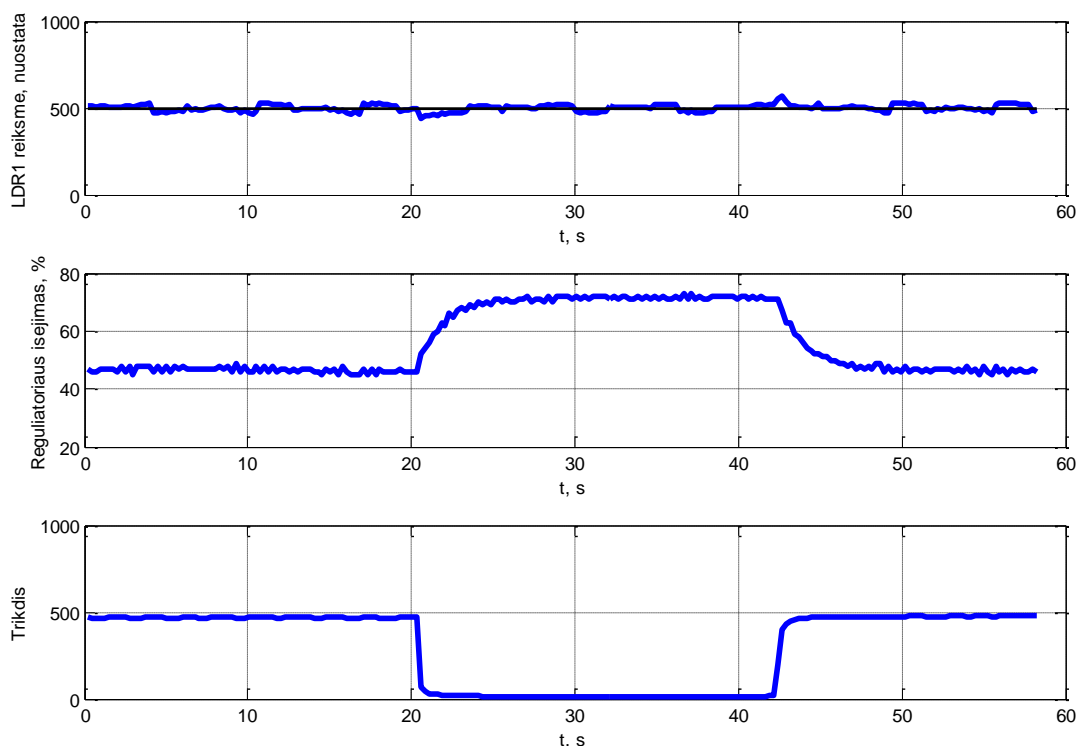


10 pav. Jungimo schema

3 lentelė. Naudojami komponentai

Komponentas	Žymėjimas	Detalės Nr.
Fotorezistorius	LDR1, LDR2	VT90N2
Rezistorius 220 Ω	R2	281-220/REEL-RC
Rezistorius 10 k Ω	R1, R3	281-10k/REEL-RC
Fotodiodas	LED1	L7113GT

Sistemos reakcija į trikdį pavaizduota 11 pav. Pirmame grafike pavaizduota fotorezistoriaus LDR1 parodymai ir nuostata, antrame grafike pavaizduota reguliatoriaus išėjimas procentais. Trečiame grafike pavaizduota nereguliuojamo fotorezistoriaus LDR2 rodmenys, šiame grafike atsispindi trikdys. Trikdys bandymui kuriamas išjungiant ($t = 20$ s) ir vėl įjungiant ($t = 43$ s) šviesą kambaryje. Iš pirmojo grafiko matyti, kad reguliatorius išlaiko reguliuojamą dydį pagal nuostatą ir atsiradus trikdžiui.



11 pav. Stebėjimo duomenys esant trikdžiui

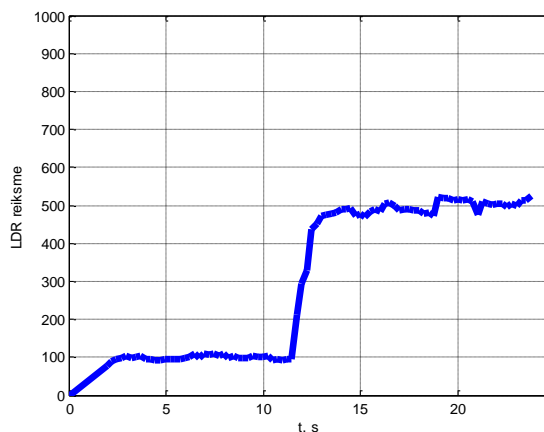
3.5 Duomenų nuskaitymas iš Arduino mikrovaldiklio plokštės

Iki šiol atliktuose eksperimentuose duomenų nuskaitymui iš Arduino platformos naudojama papildoma programine įranga. Pirmajame duomenų nuskaitymo bandyme naudojama programa „Gobetwino“ [10], bandymuose su PID reguliatoriumi naudojama programa „Bridge Control Panel“ [11]. Abi programos turi savo teigiamų ir neigiamų savybių, pavyzdžiui „Gobetwino“ gerai tinka ilgai trunkantiems matavimams, nes duomenis išsaugo tekstiniame faile su realaus laiko atžymomis, tačiau stebėjimo atžymų ciklas negali būti dažnesnis nei 1 sekundė. „Bridge Control Panel“ galima stebėti duomenis dažniau nei kas 1 sekundė, tačiau laikas skaičiuojamas nuo stebėjimo pradžios milisekundėmis arba atliktų stebėjimų skaičiumi. Taip pat naudojantis abejomis programomis reikalingas papildomas kodas (1, 2 priedas) Arduino mikrovaldiklio plokštei bei stebėjimo duomenys išsaugomi atskirame .txt faile kurį tik po to galima atvaizduoti grafike naudojantis Matlab programine terpe.

Tokiais būdais atliekant stebėjimus reikalinga papildomi veiksmai ir perteklinis kodas. Optimali išeitis būtų panaudoti Matlab programinę terpę duomenų iš Arduino mikrovaldiklio plokštės nuskaitymui ir atvaizdavimui. Duomenys iš Arduino mikrovaldiklio plokštės gaunami per COM prievadą, taigi tikslas yra naudojantis Matlab programine aplinka nuskaityti duomenis iš prievado ir grafiškai atvaizduoti.

3.6 Parametrų nuskaitymas naudojantis Matlab programine aplinka

Bandyme pavaizduota PID reguliatoriaus reakcija į nuostatos pakeitimą (3.2.1 skyrius). Matlab programinės terpės kodas pateiktas 3 priede.



12 pav. Reakcijos į trikdį atvaizdavimas, naudojantis Matlab programine aplinka

Grafike (12 pav.) pavaizduotas fotorezistoriaus parodymai keičiant nuostatą iš 100 į 500. Šiuo metu atvaizduojamas tik vienas kintamasis, vienas iš tikslų magistriniam darbui yra parašyti Matlab programinės aplinkos kodą, kuris atvaizduotų visus norimus duomenis, siunčiamus iš Arduino mikrovaldiklio plokštę.

4. Arduino mikrovaldiklio plokštės ir Matlab programinės aplinkos panaudojimas procesų valdyme

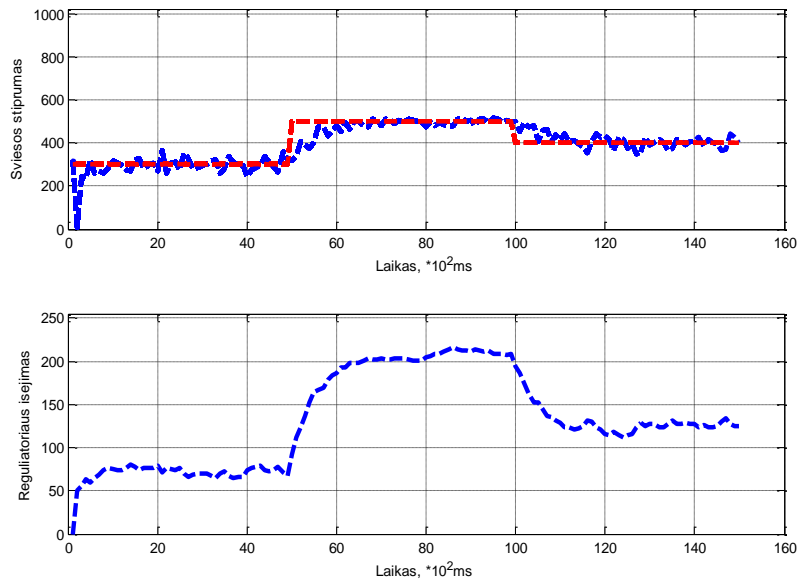
Dėl duomenų pavaizdavimo ir manipuliavimo privalumų, yra patogiu naudoti Matlab programinę terpę. Todėl visus skaičiavimus galima perkelti į kompiuterį ir panaudoti Matlab programinę aplinką, o Arduino mikrovaldiklio plokštę naudoti tik kaip platformą, skirtą surinkti duomenims iš jutiklių ir valdyti objektą su duomenimis, ateinančiais iš kompiuterio per nuoseklų prievadą. Nors reguliatoriaus sintezė ir visi derinimai patogumo dėlei yra atliekami Matlab programinėje aplinkoje realizuojant ryšį su plokšte, po to reguliatorius ir derinimo algoritmas, naudojant Matlab transliatorių, gali būti transformuojamas į stand-alone versiją ir pervedamas į mikrovaldiklį, tada MATLAB programinės terpės nereikia algoritmų realizavimui ir Arduino mikrovaldiklio plokštė gali veikti savarankiškai be kompiuterio ar Matlab programinės įrangos.

4.1 Duomenų mainai tarp Matlab programinės terpės ir Arduino mikrovaldiklio plokštės

Arduino platformos sujungimui su kompiuteriu yra naudojamas nuoseklus prievadas, per jį yra siunčiama programa į valdiklį, taip pat gaunami duomenys iš jo. Duomenų mainams panaudojamas ArduinoIO [12] palaikymas, leidžiantis kreiptis į Arduino mikrovaldiklio plokštės išėjimus ir įėjimus naudojantis Matlab programine aplinka. Taip galima rašyti kodą Matlab programinėje terpėje, o objekto duomenis gauti iš mikrovaldiklio prievadų.

4.2 Reguliatoriaus realizavimas Matlab programinėje terpėje

Išsiaiškinus kaip gauti jutiklių duomenis iš Arduino mikrovaldiklio plokštės realiu laiku, suformavau PID reguliatorių Matlab terpėje, valdantį eksperimentams naudojamą objektą (3.2.1 skyrius). Reguliatoriaus ciklo kodas pateiktas 4 priede. Reguliatoriaus reakciją į nuostatos keitimą parodyta 13 pav.



13 pav. Regulatoriaus reakcija į nuostatos keitimą

4.3 Automatinis regulatoriaus parametrų derinimas

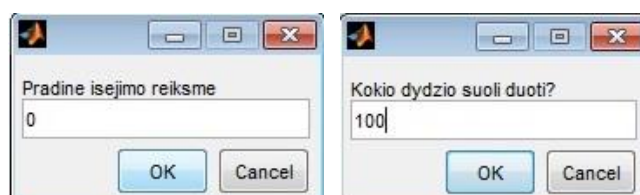
Perkėlus regulatorių ir visus su objektu susijusius skaičiavimus į Matlab programinę aplinką galima realizuoti automatinį regulatoriaus suderinimą. Regulatoriaus parametrus apskaičiuoju panaudojus inžinerinius regulatorių derinimo parametrų apskaičiavimo metodus [13].

Norint pritaikyti regulatorių parametrų derinimo metodus reikia gauti objekto perdavimo funkciją. Tiriamas objektas yra statinis, todėl jį išreiškiame pirmos eilės aperiodine grandimi su vėlavimu:

$$W_0(s) = \frac{k_0}{T_0s + 1} \cdot e^{-\tau_0s}$$

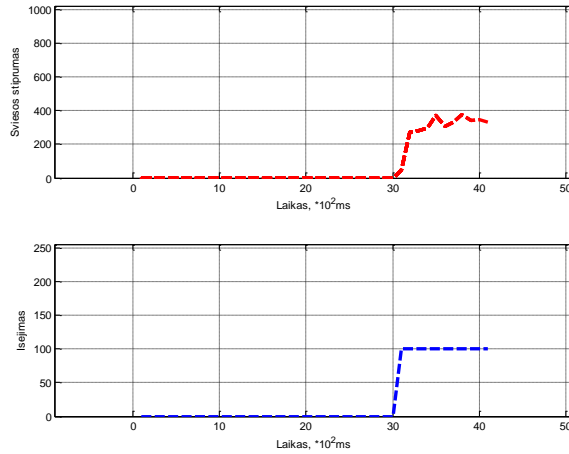
čia k_0 – perdavimo koeficientas, T_0 – laiko pastovioji, τ_0 – vėlavimas.

Šiuos parametrus galime nustatyti turėdami eksperimentinę valdymo objekto reakciją į šuolinį įėjimo signalą. Šuolinės reakcijos generavimui suformavau kodą Matlab programinėje terpėje, kuri pirma paprašo pateikti pradinę išėjimo reikšmę bei kokio dydžio šuolį sukurti:



14 pav. Pradinių duomenų reikalaujantys dialogo langai

Su gautais pradiniais duomenimis kuriamas šuolinis signalas ir gaunama objekto reakcija:



15 pav. Šuolinis signalas ir objekto reakcija

Iš gautos reakcijos reikia apskaičiuoti pirmosios eilės objekto su vėlavimo grandimi parametrus k_0 , T_0 ir τ_0 . Teorinė šuolinės reakcijos kreivė laiko momentams, didesniems negu vėlavimas, gali būti užrašyta tokia forma:

$$y(t) = y_{\infty} \left(1 - e^{-\frac{t-\tau_0}{T_0}} \right).$$

Čia patogų įvesti naują kintamąjį y_{∞} , kuris žymi nusistovėjusį reakcijos kreivės įvertį. Kadangi proceso išėjimas stebimas laiko momentais t_k , pastarąją lygtį patogų užrašyti taip:

$$y(k) = y_{\infty} \left(1 - e^{-\frac{t_k-\tau_0}{T_0}} \right).$$

Šios lygties nežinomus parametrus apskaičiuojame šitaip:

Perdavimo koeficientas k_0

Galutinis šuolinės reakcijos kreivės įvertis y_{∞} ir šuolinės funkcijos, veikiančios įėjime, amplitudės įvertis A yra susiję šitokia priklausomybe:

$$y_{\infty} = Ak_0.$$

Iš čia proceso perdavimo koeficiento įvertis gali būti apskaičiuotas šitaip:

$$k_0 = \frac{y_{\infty}}{A}.$$

Laiko pastovioji T_0 ir vėlavimas τ_0

Teorinės šuolinės reakcijos kreivės išraiškoje parametrai T_0 ir τ_0 įeina netiesiškai. Tačiau šią lygtį galima perrašyti šitokiu būdu:

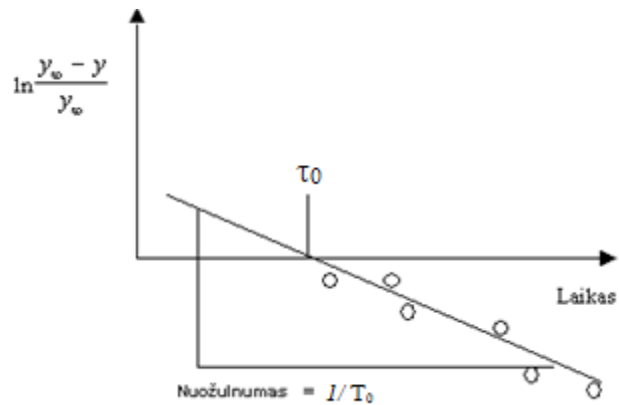
$$\frac{y_{\infty} - y}{y_{\infty}} = e^{-\frac{t - \tau_0}{T_0}}.$$

Iš šios lygties gauname:

$$\ln\left(\frac{y_{\infty} - y}{y_{\infty}}\right) = \frac{\tau_0}{T_0} - \frac{t}{T_0}$$

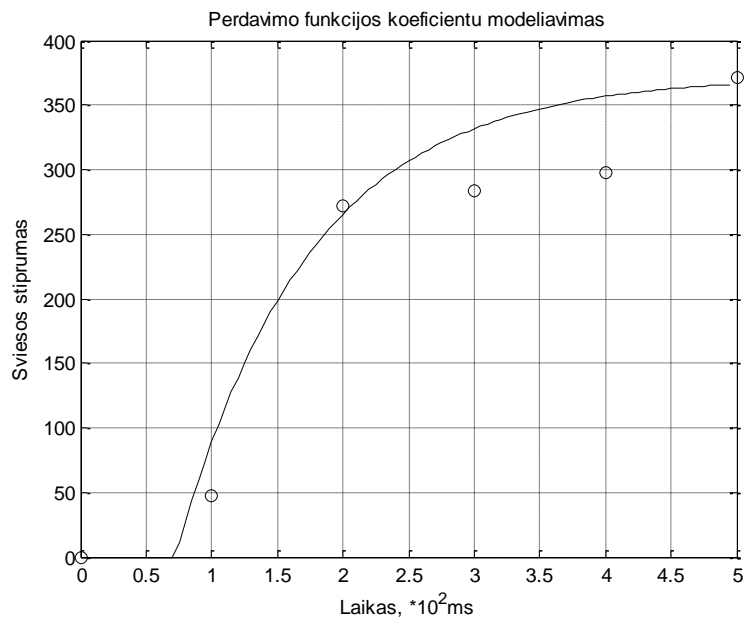
Braižant transformuotos funkcijos grafiką, kurio ordinatės atitinka lygties kairės pusės įverčius, o abscisės - laiką t , gaunama tiesė, kurios pakrypimo kampas yra $1/T_0$, o susikirtimo su ordinate taškas - τ_0/T_0 . Nesunku pastebėti, kad ši tiesė su t ašimi susikerta taške $t = \tau_0$. Šis grafikas pavaizduotas 16 pav.

Tokiu būdu, minėtus parametrus galima rasti grafiškai, tačiau tikslesni įverčiai gaunami iš tiesinės regresijos lygties.



16 pav. Laiko pastoviosios ir vėlavimo laiko įvertinimas, taikant šuolinės reakcijos kreivės eksperimentinius duomenis

Modelio perdavimo funkcijų nustatymo Matlab kodas pateiktas 5 priede. Apskaičiuojame objekto parametrus ir palyginame modeliavimo rezultatus su atlikto eksperimento duomenimis:



17 pav. Modeliavimo rezultatų ir eksperimento duomenų palyginimas

Eksperimento metu gauti objekto perdavimo koeficientai:

$$k_0 = 3,72;$$

$$T_0 = 1,02;$$

$$\tau_0 = 0,72.$$

Iš gautų objekto perdavimo koeficientų apskaičiuojame reguliatoriaus parametrus, naudojame analizinį metodą, derinimo formulės, kai valdomas statinis objektas, reguliavimo dėsnis – PI, tipinis pereinamasis procesas – aperiodinis [13]:

$$k_r = \frac{0,6}{\frac{k_0 \tau_0}{T_0}} ;$$

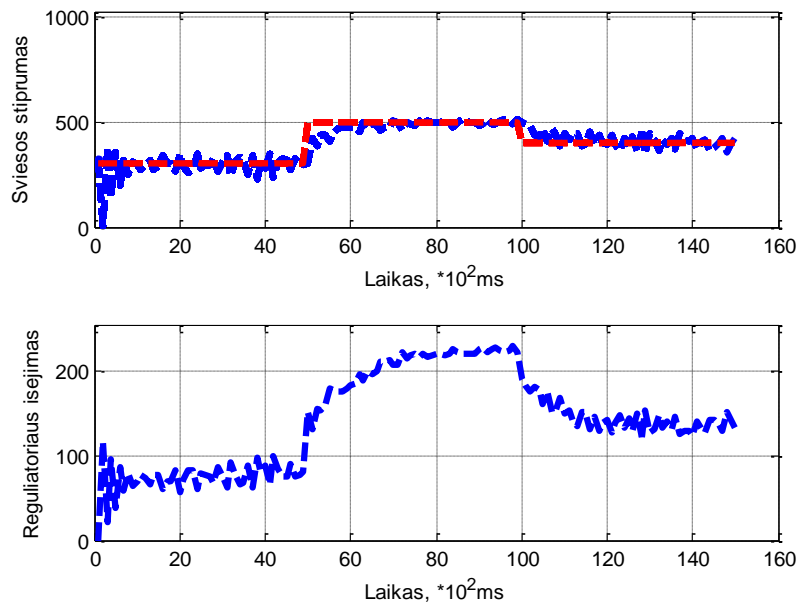
$$T_i = 0,6T_0.$$

Gauti reguliatoriaus parametrai:

$$k_r = 0,22;$$

$$T_i = 0,61.$$

Programa iš karto įjungia reguliatorių su gautais parametrais, veikimą stebime tikrindami objekto reakciją į nuostatos keitimą:



18 pav. Objekto reakcija į nuostatos keitimą

Iš gautų duomenų galima teigti, kad automatinis reguliatoriaus parametrų nustatymas veikia. Tiesa iš perdavimo funkcijos koeficientų modelio matyti, kad gaunami koeficientai nėra itin tikslūs.

5. Šiltnamio fizinis modelis ir jo valdymas

5.1 Mikroklimato palaikymas

Šiltnamis, terariumas ar kiti mikroklimatą palaikantys objektai aprūpina reikalingas klimato sąlygas augalams ar gyvūnams, esantiems jo viduje bei atskiria nuo galimai netinkamų klimato sąlygų išorėje. Šiltnamio viduje palaikoma reikiama oro temperatūra ir drėgmė, apšviestumas bei dirvožemio savybės. Augalų kokybė ir vaisingumas stipriai priklauso nuo mikroklimato valdymo kokybės šiltnamio viduje, o valdymas kokybė priklauso nuo surinktų stebėjimo duomenų šiltnamio aplinkoje. Taigi reikalinga stebėti tokius kintamuosius kaip temperatūra, oro ir dirvožemio drėgmė, kad augintojas geriau suprastų kaip kiekvienas iš šių kintamųjų veikia augimą ir tai galėtų išnaudoti maksimaliam pasėlių produktyvumui ir atsiperkamumui.

Anksčiau atliktuose eksperimentuose aiškinomės Arduino platformos galimybes ir pritaikomumą procesų valdymui ir stebėjimui. Darbo tikslas yra sukurti ir optimizuoti aplinką, palaikančią reikalingas mikroklimato sąlygas stebint oro temperatūrą bei drėgmę, apšviestumą, dirvožemio temperatūrą bei drėgmę, kontroliuojant dirvožemio drėkinimą naudojantis valdymo sistema, paremta Arduino platforma.

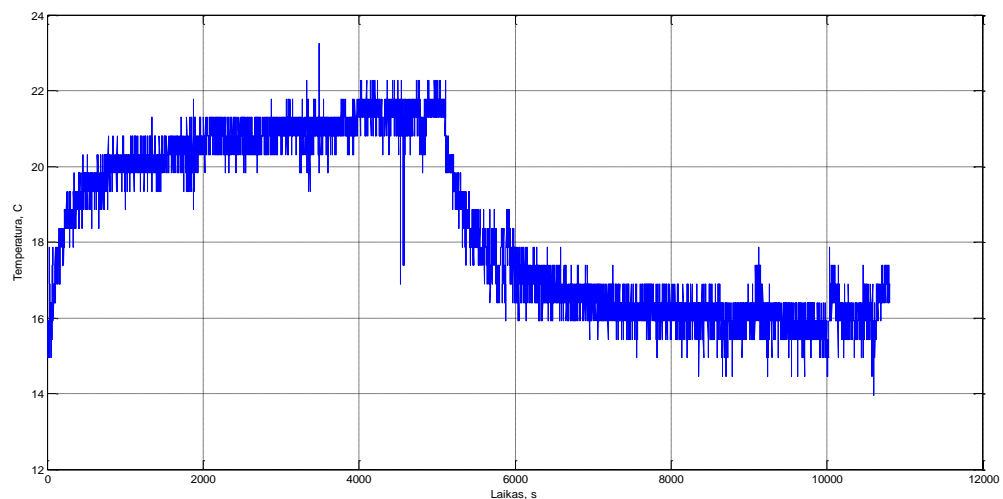
5.2 Tiriamasis objektas

Šiame darbe bus tiriamas šiltnamio mikroklimatas. Objekto tyrimams reikalingas fizinis šiltnamio modelis. Modelio kūrimui panaudota paprasta metalo konstrukcija, turinti kubo formą su 0,55 m kraštine (19 pav.). Taip pat viena iš sienų gali būti atidaroma.

Šiltnamio modelyje bus tiriamas temperatūros valdymas. Temperatūrai reguliuoti naudojamos 100 W kaitinamosios lempos. Pirmasis bandymas atliekamas įjungiant ir po 5000 sekundžių išjungiant lempas, matuojama šiltnamio temperatūra. Gauti rezultatai parodo, kad dviejų 100 W kaitinamųjų lempų pakanka šiltnamyje temperatūrai pakelti 6 – 7 °C. (20 pav.).



19 pav. Šiltnamio fizinis modelis



20 pav. Temperatūra šiltnamyje

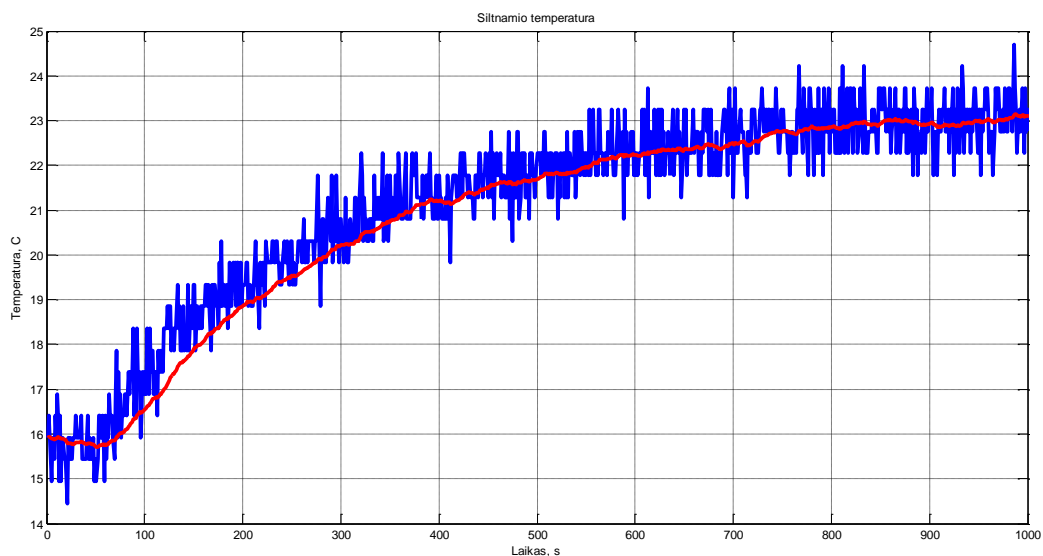
5.3 Duomenų filtravimas

Temperatūros matavimui naudojamas termorezistorius TMP36, šio termorezistoriaus tikslumas nėra didelis, parodymai gaunami su triukšmais. Matavimų netikslumai kenkia reguliavimo kokybei, panaudojus diskretųjį eksponentinį filtrą galima duomenis, gautus iš termorezistoriaus, filtruoti. Naudojama lygtis [14]:

$$(x_f)_n = \alpha \cdot (x_f)_{n-1} + (1 - \alpha) \cdot x_n$$

čia α – filtro koeficientas (gali būti 0 – 1), $(x_f)_n$ – nufiltruotas išėjimas, $(x_f)_{n-1}$ – buvusi reikšmė.

Filtro koeficientas α nurodo kiek filtruojame dydį, koeficientą galime parinkti nuo 0 iki 1. Parinkus 0 reikštų, kad eksperimentinė funkcija nefiltruojama, parinkus 1 gaunama tiesė ordinačių ašyje lygi tokiam dydžiui, koks buvo pirmasis eksperimentinės kreivės taškas. Atlikus bandymus darbui parenkamas filtro koeficientas $\alpha = 0,97$. Gaunama tokia filtruota reakcijos kreivė (21 pav.):

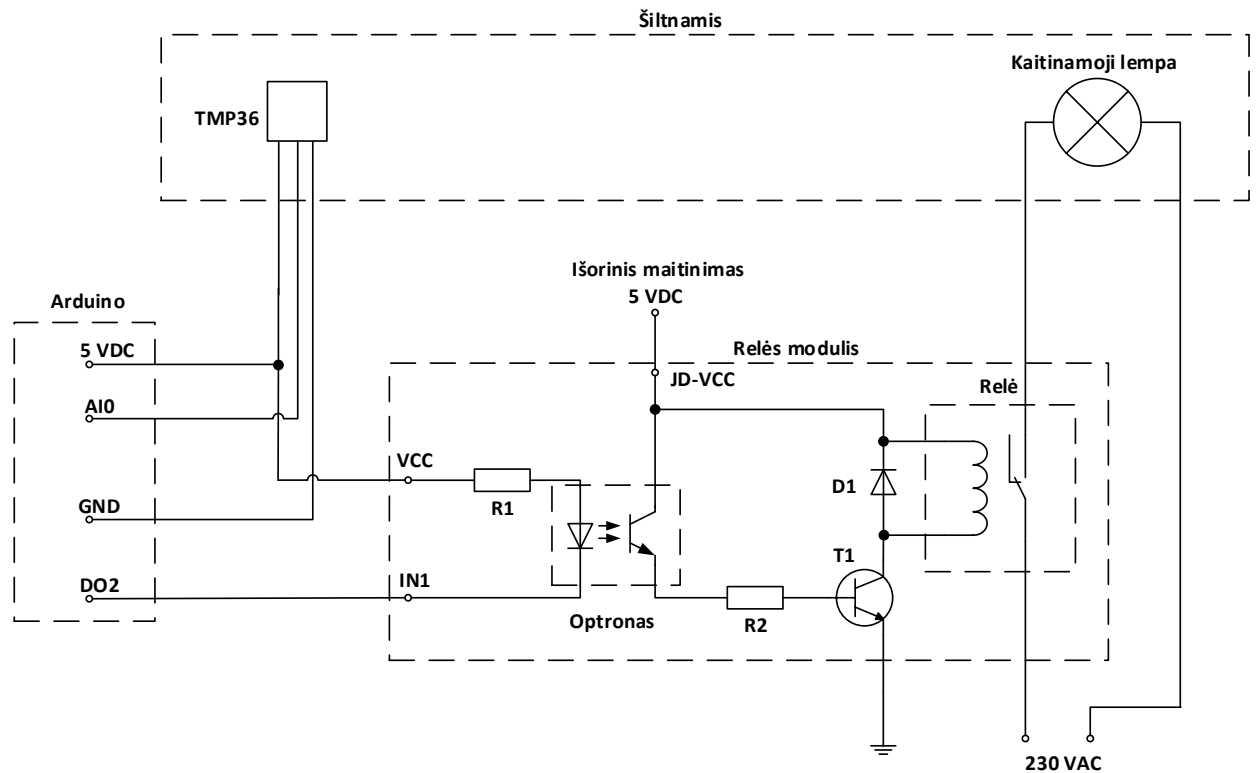


21 pav. Filtruota reakcijos kreivė, $\alpha = 0,97$

5.4 Temperatūros reguliavimas

5.4.1 Elektrinė šildymo sistemos schema

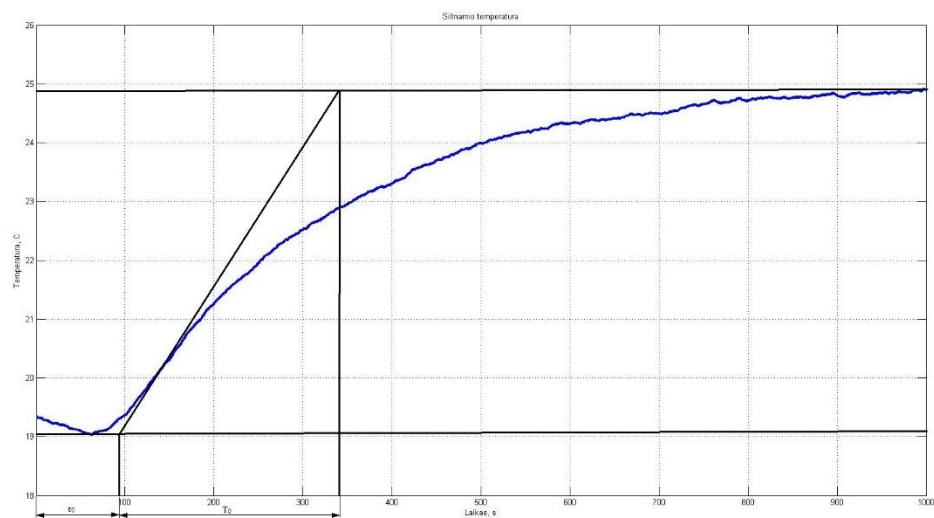
Temperatūros palaikymui šiltnamio medelyje panaudojama 2 kaitinamosios 100 W lempos, jų pakanka šiltnamio temperatūrai pakelti apie 6 – 7 °C. Kadangi tolydžiai reguliuoti kaitinamąsias lempas reikėtų papildomos įrangos ir kaštų, pasirenkamas impulsinis reguliavimo būdas. Taigi temperatūra šiltnamyje bus reguliuojama įjungiant arba išjungiant lempas pagal PI reguliatoriaus dėsni. Reguliavimui panaudojama 5 V relė, šildymo sistemos elektrinė schema (22 pav.):



22 pav. Šildymo sistemos elektrinė schema

5.4.2 Temperatūros reguliavimas

Kaip minėta temperatūros reguliavimui šiltnamyje bus naudojama impulsinis reguliatorius norint išsaugoti sistemą kuo paprastesnę ir pigesnę. Jau turėtas tolydžiojo reguliatoriaus kodas papildomas impulsinio reguliatoriaus dėsniu. Objekto reakcijos kreivė aproksimuojama inžineriniu metodu iš šuolinės charakteristikos (23 pav.):



23 pav. Šuolinė charakteristika

Gaunami perdavimo funkcijos parametrai:

$$k_0 = 0,059;$$

$$T_0 = 250;$$

$$\tau_0 = 90.$$

Naudojamas PI reguliatoriaus dėsnis, reguliatoriaus lygtis:

$$U(t) = k_r e(t) + k_i \int_0^t e(t) dt,$$

$$\text{čia } k_i = \frac{k_r}{T_i}.$$

Reguliatoriaus parametrai skaičiuojami analiziniu metodu, kai tipinis pereinamasis procesas yra aperiodinis, derinimo formulės [13]:

$$k_r = \frac{0,6}{\frac{k_0 \tau_0}{T_0}};$$

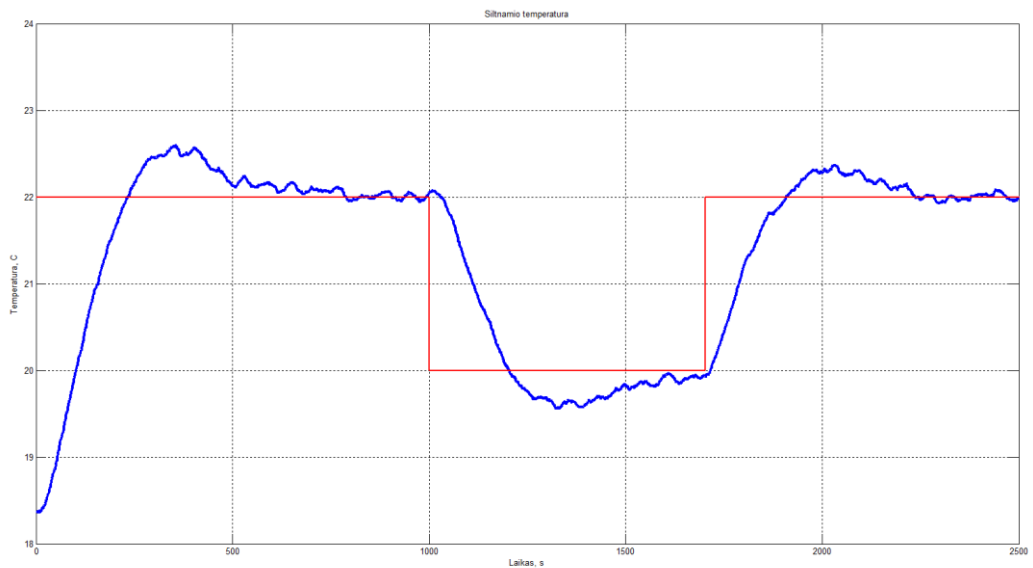
$$T_i = 0,6 T_0.$$

Gauti reguliatoriaus parametrai:

$$k_r = 28;$$

$$k_i = 0,19$$

Išbandomas reguliatoriaus veikimas, darbo metu keičiama nuostata (24 pav.):



24 pav. Šiltnamio temperatūra

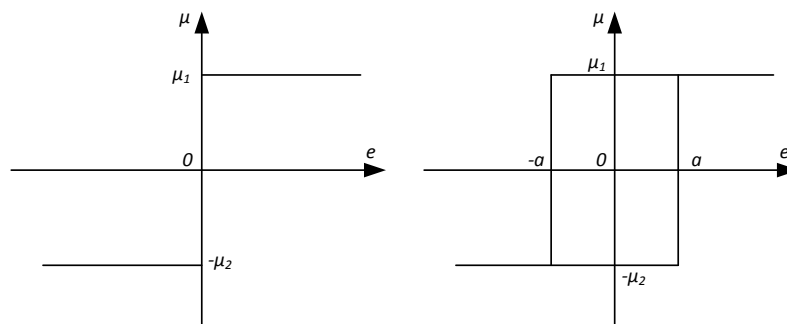
5.4.3 Valdymas poziciniu reguliatoriumi

Šiltnamio temperatūros palaikymui praktikoje dažniausiai renkamasi paprastesnę valdymo sistema su poziciniu reguliatoriumi. Pozicinių reguliatorių veikimas paremtas principu įjungta-išjungta. Jie gali būti dvipoziciniai, tripoziciniai ir daugiapoziciniai. Pozicinių reguliatorių galima atvaizduoti 25 pav. pavaizduota schema, paprastai naudojamas termostatas.



25 pav. Pozicinio reguliatoriaus struktūrinė schema

Dvipozicio reguliatoriaus statinės charakteristikos be grįžimo zonos ir su grįžimo zona pateiktos 26 pav. [17].



26 pav. Pozicinio reguliatoriaus statinės charakteristikos

Grįžimo zonos ribos yra $-a$ ir $+a$. Grįžimo zonos plotis $\Delta_{gr}=2a$. Analitinė dvipozicio reguliatoriaus be grįžimo zonos statinės charakteristikos išraiška yra tokia:

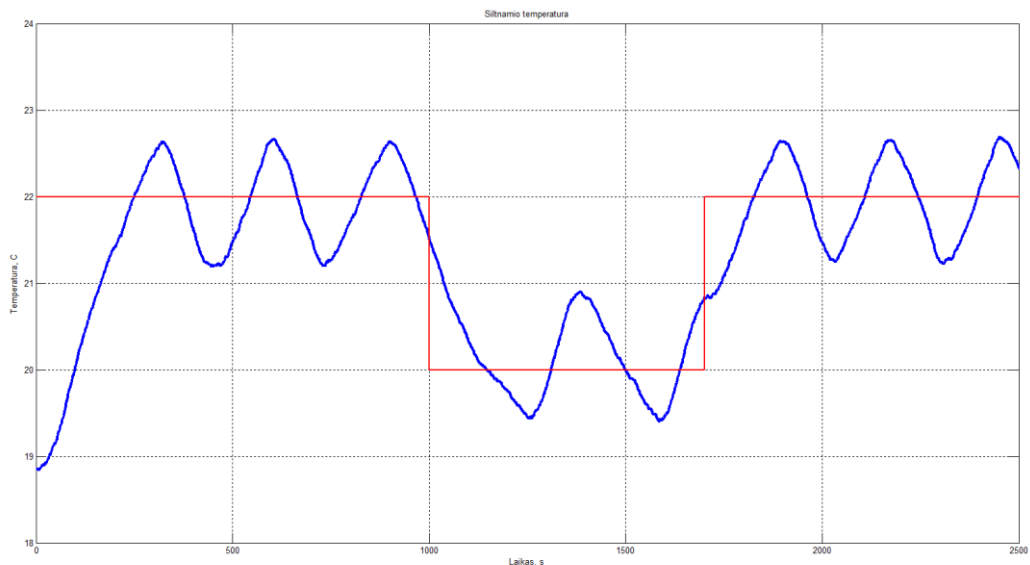
$$\begin{aligned}\mu &= \mu_1, \text{ kai } e > 0, \\ \mu &= -\mu_2, \text{ kai } e < 0,\end{aligned}$$

su grįžimo zona:

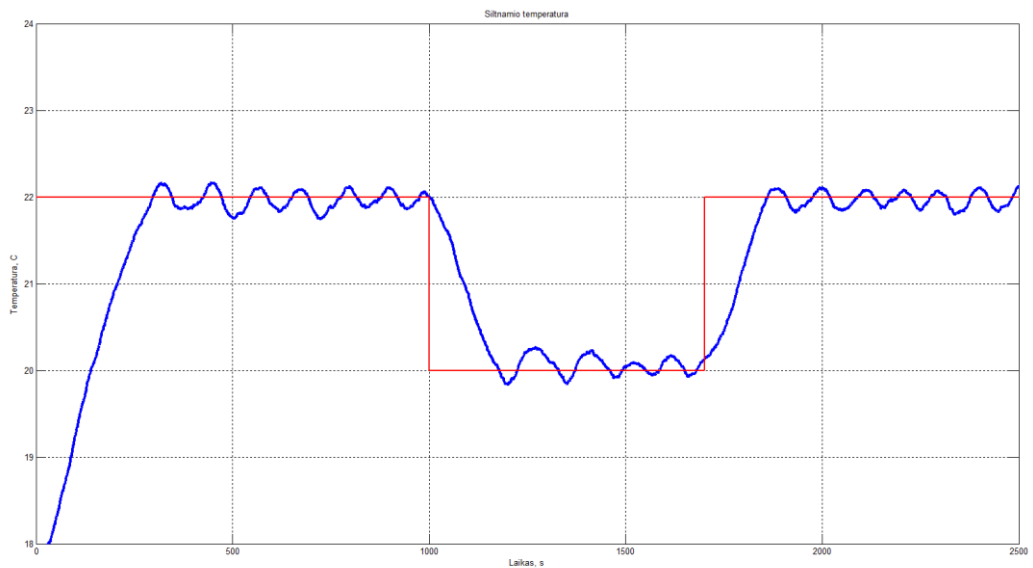
$$\begin{aligned}\mu &= \mu_1, \text{ kai } e \geq 0, \\ \mu &= -\mu_2, \text{ kai } e \leq 0, \\ \mu &= \mu_1, \text{ kai } -a < e < a \text{ ir } \frac{de}{dt} < 0, \\ \mu &= -\mu_2, \text{ kai } -a < e < a \text{ ir } \frac{de}{dt} > 0.\end{aligned}$$

Iš šių lygčių ir charakteristikų matyti, jog dvipozicis reguliatorius nuolat veikia valdymo objektą ir tas poveikis yra didesnis negu reikėtų, kad sistema būtų pusiausvira, t. y. $e = 0$. Todėl sakoma, kad sistema su dvipoziciumi reguliatoriumi veikia auto svyravimų režimu pusiausvyros būsenos aplinkoje[16].

Naudojant tą pačią šildymo sistemos schemą ir Arduino mikrovaldiklį nesunku imituoti termostato veikimą. Išbandomas reguliatoriaus veikimas, darbo metu keičiama nuostata taip pat kaip ir prieš tai darytame bandyme su impulsiniu reguliatoriumi. Šiltnamio temperatūra, kai veikia dvipozicis reguliatorius su grįžimo zona $a = 0,5^\circ\text{C}$ pavaizduota 27 pav., be grįžimo zonos 28 pav.



27 pav. Temperatūra veikiant dvipoziciumi reguliatoriui su grįžimo zona

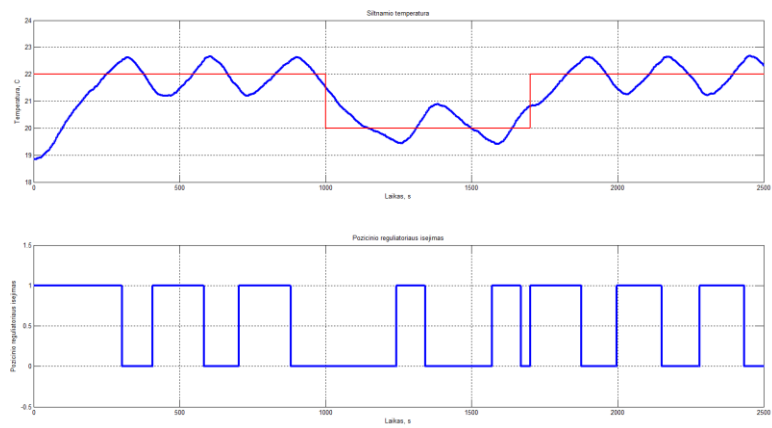


28 pav. Temperatūra veikiant dvipoziciniui reguliatoriui be grįžimo zonos

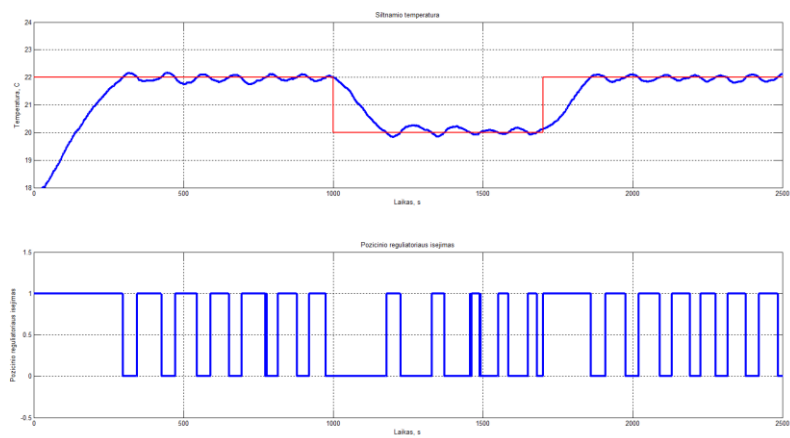
5.4.4 Reguliatorių valdymo rezultatai

Iš gautų duomenų matosi, kad temperatūra naudojant pozicinį reguliatorių turi pastovių svyravimų. Svyravimai dideli, kai naudojamas reguliatorius su grįžimo zona, svyravimai mažesni, kai reguliatorius be grįžimo zonos. Reguliatorius, veikiantis PI dėsnio nuostatą seka tolygiau, neturi pastovių svyravimų.

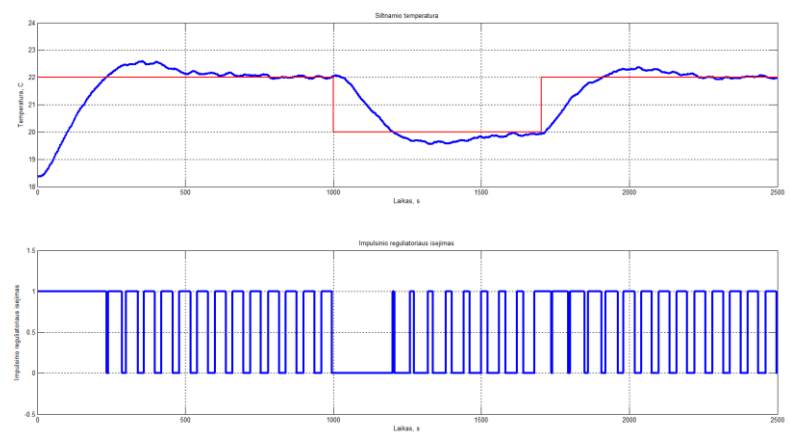
Kita vertus PI reguliatorius turi didesnę dinaminę nuokrypį, tai matoma 24 pav. kai keičiama nuostata. Pozicinis reguliatorius be grįžimo zonos dinaminis nuokrypis yra mažesnis. Taip pat, jei atkreiptume dėmesį į reguliatorių impulsų trukmę ir skaičių, pozicinių reguliatorių impulsai, šiuo atveju, dažniausiai yra ilgesni ir jų skaičius mažesnis. Pozicinių reguliatorių išėjimai kartu su temperatūra pavaizduoti 29, 30 pav., PI reguliatoriaus rezultatai pavaizduoti 31 pav. Tai gali turėti įtakos jei svarbu vykdyklio įsijungimo ciklą dažnumas ir skaičius.



29 pav. Šiltnamio temperatūra kartu su pozicinio regulatoriaus išėjimu



30 pav. Šiltnamio temperatūra kartu su pozicinio regulatoriaus išėjimu



31 pav. Šiltnamio temperatūra kartu su PI regulatoriaus išėjimu

Reguliavimo kokybės palyginimui paskaičiuojama vidutinė absoliutinė bei vidutinė kvadratinė paklaida. Matavimo rezultato vidutinė absoliutinė paklaida gali būti įvertinta taip:

$$\Delta x_a = \frac{\sum_{i=1}^n |\Delta x_i|}{n}$$

vidutinė kvadratinė paklaida:

$$\Delta x_k = \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2$$

čia n – matavimų skaičius, x_i - i -tojo matavimo metu gauta matuojamojo dydžio vertė, y_i - i -tojo matavimo metu gauta nuostatos dydžio vertė. Visais skaičiuojamais atvejais regulatoriai veikia identiškomis sąlygomis, matavimų skaičius $n=1000$.

Paklaidos, gautos naudojant PI reguliatorių:

- vidutinė absoliutinė paklaida $\Delta x_a=0,5599$;
- vidutinė kvadratinė paklaida $\Delta x_k=1,1278$.

Paklaidos, gautos naudojant pozicinį reguliatorių, kai nejautrumo zona $a = 0$ °C:

- vidutinė absoliutinė paklaida $\Delta x_a=0,6666$;
- vidutinė kvadratinė paklaida $\Delta x_k=1,7512$.

Paklaidos, gautos naudojant pozicinį reguliatorių, kai nejautrumo zona $a = 0,5$ °C




- vidutinė absoliutinė paklaida $\Delta x_a=0,7214$;
- vidutinė kvadratinė paklaida $\Delta x_k=1,8843$.

Taigi įvertinus procentais PI reguliatoriaus vidutinė absoliutinė paklaida yra 19 % mažesnė už pozicinio reguliatoriaus be nejautos zonos ir 28,8% mažesnė už reguliatoriaus su nejautos zona lygia 0,5 °C. Vidutinė kvadratinė paklaida su PI reguliatoriumi yra 55,2% mažesnė už pozicinio reguliatoriaus be nejautos zonos ir 67% mažesnė už reguliatoriaus su nejautos zona lygia 0,5 °C.

5.4.5 Reguliavimo įtaisų palyginimas

Termostatai šiltnamių šildymo sistemose naudojami dėl savo aptarnavimo paprastumo ir nedidelės kainos palyginus su pramoniniais valdikliais. Tačiau Arduino mikrovaldiklio plokštė turi nedidelę kainą ir plačias valdymo galimybes. Palyginimui galima pasirinkti Siemens LOGO! iš pramoninių valdiklių kategorijos, skirtų paprastoms automatinėms sistemoms, bendros paskirties termostatą Velleman VM148 su ekranėliu, galima histereze ir temperatūros jutikliu ir darbe naudojamą Arduino UNO R3 mikrovaldiklio plokštę.

4 lentelė. Reguliavimo įtaisai

Pavadinimas	Siemens LOGO!	Velleman VM148	Arduino UNO R3
			
Funkcijos	<ul style="list-style-type: none"> • PI reguliatorius • plačios programavimo galimybės 	<ul style="list-style-type: none"> • pozicinis reguliatorius su histereze 	<ul style="list-style-type: none"> • PID reguliatorius • plačios programavimo galimybės
Kaina*, Eur	115-170	33	20-25

* - kainos galiojančios darbo rašymo metu, oficialiose produkto svetainėse arba Lietuvos rinkoje

Matyti, kad pramoninis valdiklis yra keletą kartų brangesnis nei termostatas ir Arduino mikrovaldiklio plokštė. Tačiau Siemens LOGO! valdikliu galima realizuoti PI reguliatorių ir kitas logines funkcijas. Termostatas pigus, tačiau juo galimas tik pozicinis reguliatorius. Arduino mikrovaldiklio plokštė turi nedidelę kainą, juo galimi abu reguliatoriaus tipai, taip pat kitos funkcijos priklausančios tik nuo parašytos programos sudėtingumo ir programuotojo išmonės. Darbe pateikti pavyzdžiai: PI reguliatorius, pozicinis reguliatorius, filtras, automatinis derinimas, duomenų atvaizdavimas grafikuose realiu laiku, sąsaja su Matlab programine terpe.

5.5 Automatinis reguliatoriaus derinimas

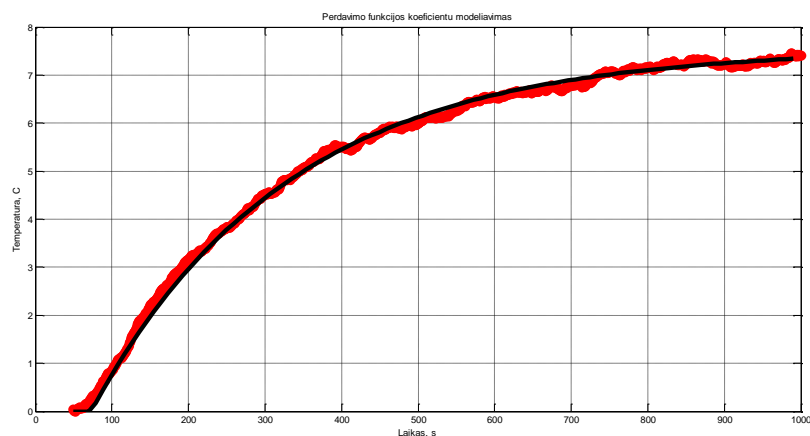
Automatinis reguliatoriaus derinimas tai metodas, kai reguliatorius operatoriaus nurodymu susiderina automatiškai. Tai naudinga kai operatorius neturi gilių žinių ir patirties apie reguliatorių derinimą. Taip pat šis metodas naudingas, kai dėl vienokių ar kitokių priežasčių pasikeičia objekto dinamika ir tuo metu naudojami reguliatoriaus parametrai tampa neoptimalūs.

Yra įvairių automatinio derinimo būdų, tačiau dažniausiai automatiniam derinimui naudojamas atviro kontūro šuolinės reakcijos eksperimentas. Eksperimento metu gaunamas proceso modelis, iš kurio apskaičiuojami reguliatoriaus parametrai. Eksperimento metu proceso neturi veikti pašaliniai trikdžiai, nes gautas proceso modelis gali neatitikti realaus proceso. Reguliatoriaus parametrai apskaičiuojami iš žinomų derinimo taisyklių.

Automatinio derinimo metu yra momentaliai išjungiamas reguliatorius, sistema paliekama dirbti atviro kontūro režimu. Į sistemą duodamas poveikis, kuris turi būti pakankamai mažas, kad nesutrikdytų proceso, bet ir pakankamai didelis, kad būtų galima kuo tiksliau nustatyti proceso modelį ir reguliatoriaus parametrus [18].

Skyriuje 4.3 atliktuose bandymuose su šviesos diodu ir fotorezistoriumi, įsitikinta automatinio derinimo pritaikomumu Arduino mikrovaldiklio plokštei. Toks pats principas panaudojamas ir su šiltnamio modeliu. Mikrovaldiklis suprogramuojamas taip, kad kiekvieną kartą paleidžiant reguliatorių jis atlieka atviro kontūro šuolinės reakcijos eksperimentą, iš eksperimento nustatoma pirmos eilės perdavimo funkcija ir PI reguliatoriaus parametrai. Bandymas užtrunka 1000 sekundžių, jam pasibaigus iš karto įsijungia reguliatorius su gautais parametrais ir atlieka objekto valdymą. Kadangi objekto modelyje naudojamas impulsinis reguliatorius todėl skaitoma, kad šuoliškai pakeistas šildančiojo elemento poveikis $\Delta\mu = 100\%$.

Atliekamas eksperimentas su automatinio derinimu, bandymo eksperimentiniai taškai ir gautos perdavimo funkcijos koeficientų modeliavimas pavaizduotas 32 pav.



32 pav. Eksperimentiniai taškai ir perdavimo funkcijos koeficientų modeliavimas

Gauti modelio parametrai:

$$k_0 = 0,075;$$

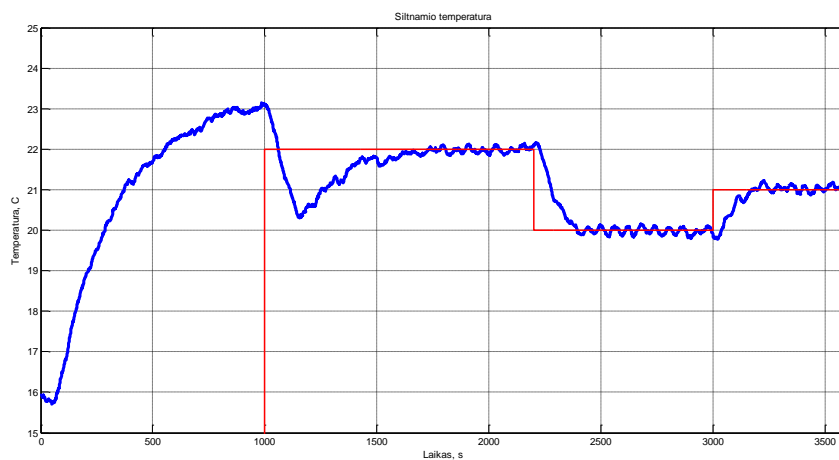
$$T_0 = 255,6;$$

$$\tau_0 = 72,6.$$

Regulatoriaus parametrai skaičiuojami naudojantis analizinį metodą PI reguliavimo dėsniai, kai tipiniam pereinamajam procesui būdingas 20 % santykinis dinaminis nuokrypis. Gauti regulatoriaus parametrai:

$$k_r = 32,8;$$

$$k_i = 0,18.$$

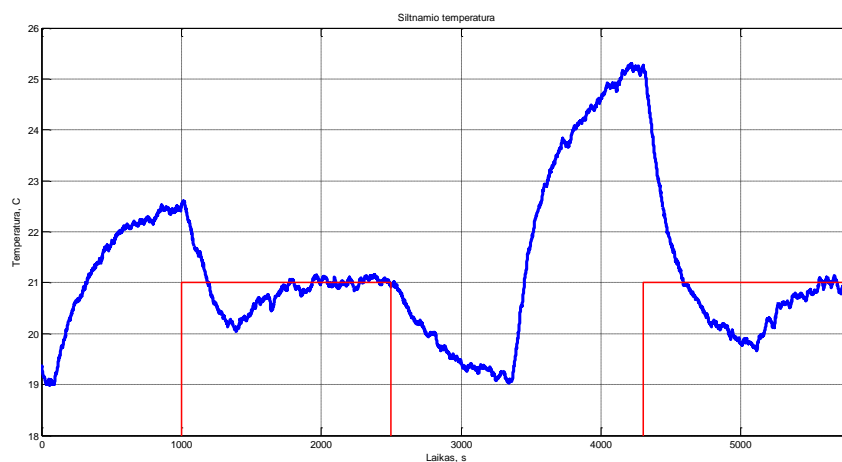


33 pav. Šiltnamio temperatūros reguliavimas

Grafike atspindi pirmosiomis 1000 sekundžių atliekamas šuolinis eksperimentas, po šuolinės reakcijos eksperimento įsijungia reguliatorius ir parodoma, kaip objekte sekama nuostata (30 pav.).

Tokio tipo automatinis derinimas naudingas kai nežinomos objekto dinamika, neturima praktinių reguliatorių derinimo įgūdžių. Kitas automatinio derinimo pritaikymas galimas kai laikui bėgant keičiasi objekto dinamika ir reguliatoriaus parametrai tampa nebe optimalūs ar net nepakankami objekto stabilumo išlaikymui. Tokiu atveju galima inicijuoti pakartotinį šuolinį eksperimentą ir gauti naujus reguliatoriaus parametrus.

Atliekamas eksperimentas, kai darbo metu automatinis derinimas pakartojamas (34 pav.):



34 pav. Šiltnamio temperatūros reguliavimas

Pirmoje bandymo dalyje, nuo 0 iki 2500 sekundžių objekte imituojamas objekto dinaminė charakteristikų pasikeitimas, lempos uždengiamos, todėl atiduoda mažiau šilumos. Objekto dinamika pasikeičia, tai matyti iš bandymo temperatūros kreivės. Temperatūra per šuolinį eksperimentą pakyla tik iki 22,5 °C. Su imituojamu dinaminė charakteristikų pasikeitimu reguliatorius susiderina sėkmingai, temperatūra atitinka nuostatą. Įgytos perdavimo funkcijos parametrų reikšmės ir reguliatoriaus parametrai:

$$\begin{aligned}
 k_0 &= 0,035; & k_r &= 73,6; \\
 T_0 &= 289,7; & k_i &= 0,36. \\
 \tau_0 &= 78,2;
 \end{aligned}$$

Antroje eksperimento dalyje dinaminė charakteristikų pasikeitimas nebe imituojamas. Ties 2500-tąja sekunde reguliatorius išjungiamas ir inicijuojamas pakartotinis automatinis derinimas. Per šuolinį eksperimentą temperatūra pakyla iki 25 °C, tada įsijungia pagal naują dinamiką suderintas

regulatorius ir ima sekti nuostatą. Įgytos perdavimo funkcijos parametrų reikšmės ir regulatoriaus parametrai:

$$\begin{aligned}k_0 &= 0,063; & k_r &= 40,6; \\T_0 &= 268,2; & k_i &= 0,21. \\ \tau_0 &= 73,3.\end{aligned}$$

Bandymas sėkmingas, abejais automatinio derinimo atvejais regulatorius susiderino sėkmingai, pasikeitusi objekto dinamika valdymui įtakos nepadarė.

6. Valdymo sistemos, paremtos Arduino mikrovaldiklio plokšte, sąmata

Į sąmatą įtraukti tik valdymo sistemai reikalingi komponentai, neįtraukiami į didesniam objektui reikalingus didesnės galios šildymo, drėkinimo ir kitus įrenginius. Priimama, kad realus šiltnamis nuo tirtu fizinio modelio skiriasi tik savo masteliu.

5 lentelė. Valdymo sistemos sąmata

Nr.	Prietaisas	Paskirtis	Kaina, Eur
1.	Arduino UNO R3	Objekto valdymas	22,90
2.	4 kanalų 5V rėlių modulis [15]	Vykdyto įtaisų, maitinamų 230 VDC, junginėjimas	6,20
3.	Termorezistorius TMP36	Temperatūros matavimas šiltnamyje	1,20
4.	Impulsinis maitinimo šaltinis	Išorinis maitinimas relėms	7,00
5.	Laidai, trumpikliai, bandomoji lenta	Elektrinio jungimo elementai	6,00
Iš viso:			43,30

Lentelėje pateikti šildymo reguliavimui reikalingi komponentai. Paprastai šią funkciją tokiuose objektuose kaip šiltnamis atlieka termostatas, bendros paskirties termostato kaina yra apie 30 eurų, todėl tai yra pigesnė alternatyva. Tačiau atsižvelgiant į reguliavimo kokybę (5.4.4 skyrius) pranašesnis yra Arduino platforma su PI reguliavimo dėsniumi. Taip pat reikia atsižvelgti, kad naudojant mikrovaldiklį išauga sistemos lankstumas, norint prijungti papildomų įrenginių nereikėtų įsigyti papildomų prietaisų. Pavyzdžiui laiko relė, reikalinga drėkinimo sistemai, kainuoja nuo 10 eurų. Naudojant valdiklį papildomų komponentų prijungimas galimas ir be papildomos įrangos įsigijimo, reikalinga tik papildyti valdymo algoritmą.

7. Rezultatai ir išvados

Pagrindinius baigiamojo magistro darbo rezultatus ir išvadas galima apibūdinti taip:

1. Darbe atlikta literatūros analizė, apžvelgianti Arduino ir kitas mikrovaldiklio plokštes.
2. Ištirtos Matlab programinės aplinkos ir Arduino mikrovaldiklio plokštės susiejimo, duomenų mainų, grafikų pavaizdavimo realiu laiku galimybės.
3. Realizuotas PI reguliatorius naudojant Matlab programine aplinka ir Arduino mikrovaldiklio plokšte ir eksperimentiškai įsitikinta jo korektišku veikimu.
4. Padarytas automatiškas reguliatoriaus parametrų nustatymas iš objekto perdavimo funkcijos koeficientų ir įsitikinta jo efektyvumu.
5. Sukurtas fizinis modelis – šiltnamis ir jam pritaikyta temperatūros valdymo sistema, paremta Arduino mikrovaldiklio plokšte.
6. Palygintos Arduino mikrovaldiklio plokštės valdymo galimybės ir privalumai su kitomis alternatyvomis.
7. Darbe aptarti sprendimai gali būti panaudoti patikimam procesų valdymui su minimaliomis sąnaudomis.
8. Panašios valdymo schemas gali būti pritaikomos kitų nestacionarių procesų valdymui.

8. Informacijos šaltinių sąrašas

1. KHADSE, R., GAWAI, N., FARUK, B. M., *Overview and Comparative Study of Different Microcontrollers*, India, 2014 (mikrovaldiklių klasifikacijos).
2. KAELI, D., SACHS, K., *Computer Performance Evaluation and Benchmarking*, USA, 2009, p. 57 (CISC ir RISC architektūros).
3. Arduino Uno R3 mikrovaldiklio plokštės specifikacijos, prieiga per internetą: <http://www.arduino.cc/en/Main/ArduinoBoardUno> [žiūrėta 2016 m. gegužės 18 d.].
4. Arduino oficiali internetinė svetainė, prieiga per internetą: <http://www.arduino.cc/en/Guide/Introduction/> [žiūrėta 2016 m. gegužės 18 d.].
5. Impulso pločio moduliacija realizuojama Arduino mikrovaldiklio plokštėje, prieiga per internetą: <http://www.arduino.cc/en/Tutorial/PWM> [žiūrėta 2016 m. gegužės 18 d.].
6. Arduino mikrovaldiklio plokštės analoginių įėjimų aprašymas, prieiga per internetą: <http://www.arduino.cc/en/Reference/AnalogRead> [žiūrėta 2016 m. gegužės 18 d.].
7. Termorezistoriaus specifikacijų lapas, prieiga per internetą: http://www.arduino.cc/documents/datasheets/TEMP-TMP35_36_37.pdf [žiūrėta 2016 m. gegužės 18 d.].
8. Temperatūros skaičiavimas pagal termorezistoriaus parodymus, prieiga per internetą: <https://learn.adafruit.com/tmp36-temperature-sensor/using-a-temp-sensor> [žiūrėta 2016 m. gegužės 18 d.].
9. PID bibliotekos aprašymas, prieiga per internetą: <http://playground.arduino.cc/Code/PIDLibrary> [žiūrėta 2016 m. gegužės 18 d.].
10. Duomenų rinkimo programos „Gobetwino“ aprašymas, prieiga per internetą:
11. <http://mikmo.dk/gobetwino.html> [žiūrėta 2016 m. gegužės 18 d.].
12. Duomenų rinkimo programos „Bridge Control Panel“ aprašymas, prieiga per internetą: <http://www.cypress.com/?rID=39244> [žiūrėta 2016 m. gegužės 18 d.].
13. Arduino mikrovaldiklio plokštės palaikymas Matlab programinėje aplinkoje, prieiga per internetą: <http://www.mathworks.com/matlabcentral/fileexchange/32374-legacy-matlab-and-simulink-support-for-arduino> [žiūrėta 2016 m. gegužės 18 d.].
14. ALEKSA, V., GALVANAUSKAS V., *Technologinių procesų automatizavimas ir valdymas*, Kaunas, 2008, p. 107 (inžinieriniai reguliatorių derinimo parametrų apskaičiavimo metodai).

15. LEVIŠAUSKAS D., *Automatinio reguliavimo sistemų derinimas*, Vilnius, 2008, p. 17 (eksponentinio filtro algoritmai).
16. 4 kanalų 5V rėlių modulis, specifikacijos, prieiga per internetą:
17. <http://www.duino.lt/valdymo-ir-stiprinimo-moduliai/13553-arduino-4-kanalu-5v-reliu-modulis.html> [žiūrėta 2016 m. gegužės 18 d.].
18. ALEKSA, V, GALVANAUSKAS V., *Technologinių procesų automatizavimas ir valdymas*, Kaunas, 2008, p. 86 (poziciniai reguliatoriai).
19. ASTROM, K. J., HAGGLUND, T., *Advanced PID Control*, 2006, p. 3 (paprasti valdymo su grįžtamoju ryšiu būdai).
20. LEVIŠAUSKAS D., *Automatinio reguliavimo sistemų derinimas*, Vilnius, 2008, p. 69 (automatinis derinimas).

9. Priedai

1 priedas. Analoginio signalo iš jutiklių nuskaitymo Arduino plokštės kodas

```
int fotorezistoriaus1_reiksme;
int fotorezistoriaus2_reiksme;
int termorezistoriaus_reiksme;
float voltai;
float temperatura;
int vidutine_fotorez_reiksme;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  fotorezistoriaus1_reiksme = analogRead(A0);
  fotorezistoriaus2_reiksme = analogRead(A1);
  termorezistoriaus_reiksme = analogRead(A3);

  voltai = (termorezistoriaus_reiksme/1024.0) * 5.0;
  temperatura = (voltai - 0.5) * 100;

  vidutine_fotorez_reiksme = (fotorezistoriaus1_reiksme + fotorezistoriaus2_reiksme)/2;

  Serial.print("#S|LOGTEST|[]#");
  Serial.print(fotorezistoriaus1_reiksme);
  Serial.print(";");
  Serial.print(fotorezistoriaus2_reiksme);
  Serial.print(";");
  Serial.print(vidutine_fotorez_reiksme);
  Serial.print(";");
  Serial.print(temperatura);
  Serial.print(";");
  Serial.println("]#");
  delay(250);
}
```

2 priedas. Regulatoriaus kodas Arduino platformoje

```
#include <PID_v1.h>

#define PIN_INPUT 0
#define PIN_OUTPUT 3

//Define Variables we'll be connecting to
double Setpoint, Input, Output;

double Kp=0.1, Ki=0.6, Kd=0;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

long stebejimo_Millis = 0;           //stebejimo ciklo pradziai
long stebejimo_intervalas = 250;    //stebejimo ciklo intervalo ilgis

long setpointo_Millis = 0;
long setpointo_intervalas = 10000;  //setpointo keitimo liakas

void setup()
{
  Input = analogRead(PIN_INPUT);    //initialize the variables we're linked to
  Setpoint = 100;
}
```



```

myPID.SetMode(AUTOMATIC);           //turn the PID on
myPID.SetSampleTime(20);

Serial.begin(9600);
}

void loop()
{
  Input = analogRead(PIN_INPUT);
  myPID.Compute();
  analogWrite(PIN_OUTPUT, Output);

  int outputas=Output;
  int out100=map(outputas,0,255,0,100);
  int nuostata=Setpoint;

  //=====setpointo keitimas-----

  unsigned long currentMillis_setpointo = millis();
  if(currentMillis_setpointo - setpointo_Millis > setpointo_intervalas)
  {
    setpointo_Millis = currentMillis_setpointo;
    Setpoint=500;
  }
  //=====setpointo keitimas-----

  //=====duomenų atvaizdavimas, pradžia=====

  unsigned long currentMillis_stebejimo = millis();
  if(currentMillis_stebejimo - stebejimo_Millis > stebejimo_intervalas)
  {
    stebejimo_Millis = currentMillis_stebejimo;
    int sviesa_reg=analogRead(A0);
    int sviesa_nereg=analogRead(A1);

    Serial.print("C");
    Serial.write(sviesa_reg>>8);
    Serial.write(sviesa_reg&0xff);
    Serial.write(out100>>8);
    Serial.write(out100&0xff);
    Serial.write(sviesa_nereg>>8);
    Serial.write(sviesa_nereg&0xff);
    Serial.write(nuostata>>8);
    Serial.write(nuostata&0xff);

  }
  //=====duomenų atvaizdavimas, pabaiga=====
}

```

3 priedas. Matlab programinės aplinkos kodas, duomenų iš Arduino mikrovaldiklio plokštės nuskaitymui

```

delete(instrfindall); %nutraukia rysi su COM3
clear all
close all
clc

arduino=serial('COM3','BaudRate',9600);
fopen(arduino);

sviesa_reg(1)=0;
time(1)=0;

```

```

i=1;
tic;

while (toc<=250)
    sviesa_reg(2)=fscanf(arduino,'%f');
    time(2)=toc;
    figure(1);
    grid on;
    axis([0, toc+1, 0, 1000])
    ylabel('LDR reikšme');
    xlabel('t, s');
    h(i)=plot(time,sviesa_reg,'b','LineWidth',5);
    hold on
    sviesa_reg(1)=sviesa_reg(2);
    time(1)=time(2);
    i=i+1;
end

```

4 priedas. Matlab programos fragmentas, regulatoriaus realizavimas

```

while i<150      %iteracijų skaičius
    i=i+1;
    tic;         %laikmacio statravimas

    dT = t;     %laiko skirtumas nuo tik iki toc

    Input = a.analogRead(0);      %nuskaito iejima is arduino

    error = Setpoint - Input;
    P = error * Kp;
    I = I + (error * Ki * dT);
    D = ((lastErr - error) * Kd) / dT;

    Output = P + I + D;
    a.analogWrite(3, Output);     %siuncia isejimo verte

    pause(0.1);                  %diskretizavimo zingsnis cia parenkamas
    lastErr = error;
    t = toc;                      %laikmacio reikšme nuo tic irasoma i t kintamaji
end

```

5 priedas. Matlab programos fragmentas, modelio perdavimo funkcijų nustatymas iš šuolinės charakteristikos

```

dmiu = 100;      %vienetinio suolio POKYTIS
xee=S-min(S);

t=[1:(size(S, 2))]' ;
x=[xee]' ;
te=t(1:suolio_ciklu_kiekis);
xe=x(1:suolio_ciklu_kiekis);

xe_inf = (max(S)-min(S)+0.1); %reguliuojamo dydzio nusistovejusi reikšme

log_x_rel=log((xe_inf - xe(:))./xe_inf);
T=[ones(length(te),1) -te] ;

PAR = T \ log_x_rel;

k0 = (xe_inf-xe(1))/dmiu
T0 = 1/PAR(2)
tau0 = PAR(1)*T0

%reakcijos kreives modeliavimas

```

```

for ii=1:100
    t_mod(ii) = (ii-1)*(te(end)-te(1))/100+te(1);
    if (t_mod(ii)>=tau0),
        x_mod(ii) = xe_inf*(1.0-exp(-(t_mod(ii)-tau0)/T0));
    else
        x_mod(ii)=0;
    end
end

%REGULIATORIAUS PARAMETRAI-----

Kp=0.6/(k0*tau0/T0)
Ki=Kp/(0.6*T0)

```