



**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS**

**Lukas Biliūnas**

**INDUSTRINIO ROBOTO IMITACINIO APMOKYMO  
GALIMYBIŲ TYRIMAS**

Baigiamasis magistro projektas

**Vadovas**

Prof. dr. Robertas Damaševičius

**KAUNAS, 2018**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**  
**PROGRAMŲ INŽINERIJOS KATEDRA**

**INDUSTRINIO ROBOTO IMITACINIO APMOKYMO**  
**GALIMYBIŲ TYRIMAS**

Baigiamasis magistro projektas  
Programų sistemų inžinerija (621E16001)

**Vadovas**

Prof. dr. Robertas Damaševičius

**Recenzentas**

Doc. Lina Narbutaitė

**Projektą atliko**

Lukas Biliūnas

**KAUNAS, 2018**



## KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Lukas Biliūnas

(Studento vardas, pavardė)

Programų sistemų inžinerija (621E16001)

(Studijų programos pavadinimas, kodas)

„Industrinio roboto imitacinio apmokymo galimybių tyrimas“

### AKADEMINIO SAŽINGUMO DEKLARACIJA

20 18 m. Gegužės 29 d.  
Kaunas

Patvirtinu, kad mano, **Luko Biliūno**, baigiamasis projektas tema „Industrinio roboto imitacinio apmokymo galimybių tyrimas“ yra parašytas visiškai savarankiškai ir 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)

## SANTRAUKA

Magistrinio tyrimo tikslas – ištirti ar Kinect sensorius gali būti pritaikomas industrinių ABB robotų imitaciniam apmokymui. Pirmojoje šio darbo dalyje yra analizuojama teorinė dalis, aprašomi naudojami įrankiai ir jų techninės galimybės.

Antroji – projektinė dalis. Joje pagrindžiamas projekto kūrimas, suformuota projekto paskirtis ir panaudos atvejai. Skyriuje aprašoma visa sistemos architektūra, reikalavimai sistemai, komponentai, sistemos veiklos diagrama.

Trečioji – tyrimo ir eksperimentinė dalis. Aprašomi ir detalizuojami atlikti tyrimai ir eksperimentai. Šių tyrimų metu siekiama ištirti kaip Kinect sensorius gali būti pritaikomas industrinių robotų apmokymui, ieškoma potencialių problemų, jų sprendimų ir tiriamas Kinect sensoriaus tikslumas skirtingomis aplinkos sąlygomis.

Biliūnas Lukas. Industrinio roboto imitacinio apmokymo galimybių tyrimas. Magistro baigiamasis projektas / vadovas doc. dr. Robertas Damaševičius; Kauno technologijos universitetas, Informatikos fakultetas, Programų inžinerijos katedra.

Reikšminiai žodžiai: Abb, industrinis robotas, Kinect, Hausdorff, Savitzky-Golay  
Kaunas, 2018. 52 puslapiai.

## SUMMARY

Aim of this master thesis – to make a research, does Kinect sensor could be used for learning industrial ABB robots by imitation or not. In the first part of this work theoretical part is analyzed, tools and their technical capabilities described.

Second part is all about implementation. It justifies the development of the project, describes the purpose and use cases of the project. The chapter also describes the entire system architecture, system requirements, components, system activity diagram.

The third is a research and experimental part. Studies and experiments described and detailed. Trying to investigate how the Kinect sensor can be applied to industrial robot learning by imitation, looking for potential problems and solutions to them, testing Kinect sensors accuracy on different conditions of environment.

Biliūnas Lukas. Robot Learning By Imitation Using Kinect Device: Master's thesis in Software Engineering / supervisor assoc. prof. Robertas Damaševičius. The Faculty of Informatics, Kaunas University of Technology.

Key words: Abb, industrial robot, Kinect, Hausdorff, Savitzky-Golay  
Kaunas, 2018. 52 p.

# Turinys

<b>1. ANALITINĖ DALIS.....</b>	<b>11</b>
1.1 Projekto tikslas ir adresatas.....	11
1.2 Įranga .....	11
1.2.1 Aparatinė įranga .....	11
1.2.2 Programinė įranga .....	12
1.3 Programinės įrangos pasirinkimas .....	14
1.4 Egzistuojantys sprendimai .....	14
1.4.1 Kuka .....	15
1.4.2 Yaskawa .....	15
1.5 Situacijos Lietuvoje įvertinimas .....	15
<b>2. PROJEKTINĖ DALIS .....</b>	<b>15</b>
2.1 Projekto paskirtis .....	15
2.2 Sistemos paskirtis ir panaudos atvejai .....	15
2.3 Bendri apribojimai .....	16
2.4 Reikalavimai sistemai .....	17
2.5 Sistemos architektūra .....	18
2.5.1 Sistemos statinis vaizdas .....	18
2.5.2 „UserInterface“ komponentas .....	19
2.5.3 „Hardware.Robot“ paketas .....	20
2.5.4 „Hardware.Kinect“ paketas .....	21
2.6 Veiklos diagrama .....	22
<b>3. TYRIMO IR EKSPERIMENTINĖ DALYS .....</b>	<b>23</b>
3.1 Esamas funkcionalumas.....	23
3.2 Esamos sistemos paklaidų įvertinimas.....	29
3.3 Koreliacijos skaičiavimas .....	46
3.4 Hausdorff'o atstumas .....	47
3.5 Problemos .....	48
<b>4. IŠVADOS.....</b>	<b>49</b>
<b>5. LITERATŪROS SĄRAŠAS .....</b>	<b>50</b>

## Paveikslų sąrašas

Pav. 1.1 Kinect jutiklis.....	12
Pav. 1.2 Kinect sensoriaus komunikacija .....	13
Pav. 1.3 Kinect sensoriaus fiksuojami „sąnariai“ .....	13
Pav. 2.1 Sistemos panaudos atvejai .....	16
Pav. 2.2 Sistemos komponentų diagrama .....	18
Pav. 2.3 „UserInterface“ klasių diagrama.....	19
Pav. 2.4 „RobotController“ klasė .....	20
Pav. 2.5 „RobotControllerExplorer“ klasė .....	20
Pav. 2.6 „KinectExplorer“ klasė.....	21
Pav. 2.7 „KinectHelper“ klasė .....	21
Pav. 2.8 Veiklos diagrama .....	22
Pav. 3.1 Supaprastintas programos tėkmės modelis.....	24
Pav. 3.2 Perduotų koordinačių trajektorija .....	25
Pav. 3.3 Roboto įvykdyta trajektorija .....	25
Pav. 3.4 Perduota koordinačių trajektorija pritaikius Savitzky – Golay filtrą.....	27
Pav. 3.5 Roboto trajektorijos koordinatės struktūra .....	27
Pav. 3.6 6-ių ašių industrinis robotas.....	28
Pav. 3.7 Ašių valdymo zonos .....	29
Pav. 3.8 Sumodeliuota trajektorija.....	30
Pav. 3.9 Sumodeliuota trajektorija 2.3m atstumu.....	31
Pav. 3.10 Sumodeliuota trajektorija 3m atstumu.....	33
Pav. 3.11 Vidutinė X koordinatės paklaida 10-ties bandymų metu .....	34
Pav. 3.12 Vidutinė Y koordinatės paklaida 10-ties bandymų metu .....	35
Pav. 3.13 Vidutinė Z koordinatės paklaida 10-ties bandymų metu.....	35
Pav. 3.14 X ašies koordinačių paklaidos vidurkis, dispersija ir standartinis nuokrypis. 36	
Pav. 3.15 Y ašies koordinačių paklaidos vidurkis, dispersija ir standartinis nuokrypis. 37	
Pav. 3.16 Z ašies koordinačių paklaidos vidurkis, dispersija ir standartinis nuokrypis . 37	
Pav. 3.17 Kinect gylio jutiklio matomas vaizdas esant tiesioginiams saulės spinduliams .....	39
.....	
Pav. 3.18 Vidutinė X koordinatės paklaida 10-ties bandymų metu .....	43

Pav. 3.19 Vidutinė Y koordinatės paklaida 10-ties bandymų metu .....	43
Pav. 3.20 Vidutinė Z koordinatės paklaida 10-ties bandymų metu .....	44
Pav. 3.21 Vidutinė Y koordinatės paklaida 10-ties bandymų metu .....	45
Pav. 3.22 Vidutinė Y koordinatės paklaida 10-ties bandymų metu .....	45
Pav. 3.23 Vidutinė Y koordinatės paklaida 10-ties bandymų metu .....	46

### **Lentelių sąrašas**

Lentelė 2.1 Reikalavimai sistemai .....	17
Lentelė 2.2 Reikalavimai sistemai .....	17
Lentelė 2.3 Reikalavimai sistemai .....	18
Lentelė 3.1 Paklaidos 1.3m atstumu .....	30
Lentelė 3.2 Minimalios ir maksimalios paklaidos 1.3m atstumu .....	31
Lentelė 3.3 Paklaidos 2.3m atstumu .....	32
Lentelė 3.4 Minimalios ir maksimalios paklaidos 2.3m atstumu .....	32
Lentelė 3.5 Paklaidos 3m atstumu .....	33
Lentelė 3.6 Minimalios ir maksimalios paklaidos 3m atstumu .....	34
Lentelė 3.7 Paklaidos vidurkis, dispersija ir standartinis nuokrypis .....	36
Lentelė 3.8 Paklaidos 1.3m atstumu .....	40
Lentelė 3.9 Minimalios ir maksimalios paklaidos 1.3m atstumu .....	40
Lentelė 3.10 Paklaidos 2.3m atstumu .....	41
Lentelė 3.11 Minimalios ir maksimalios paklaidos 2.3m atstumu .....	41
Lentelė 3.12 Paklaidos 3m atstumu .....	42
Lentelė 3.13 Minimalios ir maksimalios paklaidos 3m atstumu .....	42
Lentelė 3.14 Paklaidos vidurkis, dispersija ir standartinis nuokrypis .....	44
Lentelė 3.15 Koreliacijos koeficientai .....	46
Lentelė 3.16 Modifikuotas Hausdorff'o atstumas .....	47



## SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS

*Industrinis robotas* - pramonėje naudojami robotai (robotinės rankos) skirtos atlikti pasikartojančius mechaninius darbus.

*Kinect kamera* - kamera turinti gylio jutiklį.

*Roboto valdymo blokas* – tai industrinio roboto kompiuteris, kuris tarpininkauja tarp roboto ir vartotojo. Jame saugojama roboto konfigūracija ir vykdymo programa, atliekami ašių, judėjimo ir kiti skaičiavimai.

*LAN* – (ang. Local Area Network) kompiuterių tinklas, jungiantis kompiuterius ribotame plote tokiame kaip universitetas, biuras, namai.

## ĮVADAS

Daugybė pramonės įmonių jau dabar gamyboje naudoja pramoninius robotus, o dar daugiau įmonių planuoja tai padaryti netolimoje ateityje. Priežastis paprasta – pasikartojančias operacijas robotai atlieka greičiau, pigiau ir su mažesne klaidų tikimybe. Tačiau šiuolaikinė pramonė nėra statiška ir vis dažniau pareikalauja robotų ir žmonių bendradarbiavimo šalia vienas kito tam, kad žmogaus rankomis būtų atliktos užduotys, kurios yra sunkiai automatizuojamos, o robotai atliktų kitus, monotoniškus darbus, su maža klaidų tikimybe.

Pramonėje vis dažniau girdint „Industry 4.0“ sąvoką, kyla poreikis turėti lankstesnes ir interaktyvesnes industrinių robotų apmokymo galimybes, su mažesnėmis laiko sąnaudomis. Kadangi industriniai robotai yra pakankamai kompleksiški įrengimai – jų programavimas trunka ilgai, reikalauja specialių žinių ir didelių išlaidų jeigu dirbantis personalas neturi reikalingos kompetencijos.

Įvertinus šiuos faktus bus bandoma iširti ar Kinect jutiklis gali būti panaudotas industriniams robotams apmokyti imitaciniu būdu.

# 1. ANALITINĖ DALIS

## 1.1 Projekto tikslas ir adresatas

Projekto tikslas yra ištirti ar Kinect kamera gali būti panaudojama industrinio ABB roboto imitaciniam apmokymui. Produkto paskirtis yra įgalinti „Industry 4.0“ standartą [1] ir panaudoti lankstesnes priemones industrinių robotų apmokymui. Industrinio roboto apmokymas tokiu būdu padėtų inžinieriams sutaupyti laiko programuojant robotus tais atvejais, kai roboto judėjimo koordinatės neprivalėtų būtų precizinio tikslumo.

Uždaviniai:

- Įvertinti ar siūlomos ABB priemonės robotų programavimui yra pakankamai efektyvios ir funkcionalios šio projekto įgyvendinimui;
- Ištirti ir išanalizuoti sukurtos industrinio roboto imitacinio apmokymo sistemos problemas;
- Ištirti Kinect jutiklio tikslumą skirtingomis aplinkos sąlygomis;

## 1.2 Įranga

### 1.2.1 Aparatinė įranga

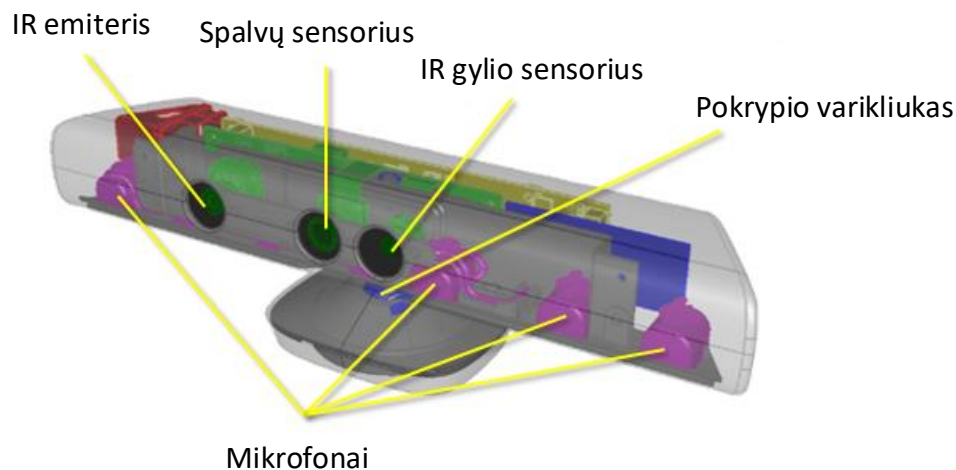
Kinect kamera iš viso turi 3 skirtingus prietaisus, kurie dirbdami kartu sugeba užfiksuoti viso kūno erdvinius judesius, įgalina veido ir balso atpažinimą [2]. Tie trys prietaisai yra:

- VGA kamera – tai RGB (angl. red, green, blue) spalvų kamera, kuri turi 640x480 pikselių skiriamąją gebą (angl. resolution) ir filmuoja 30 kadru per sekundę (angl. FPS) dažniu. Kinect įrenginys taip pat gali perduoti vaizdą tiesiogiai iš IR kameros - gylio srautą. Kinect sensorius gali sėkmingai veikti 0.7m – 6m atstumo diapazone. Sensoriaus matymo laukas yra 57° horizontaliai ir 43° vertikalčiai. Motorizuota ašis leidžia pakelti ar nuleisti Kinect sensorių dar 27°.
- IR (angl. infrared) gylio jutiklis – jis sudarytas iš monochrominio CMOS jutiklio ir infraraudonųjų spindulių projektorius, kurie padeda sukurti 3D vaizdą. Šio sensoriaus veikimo atstumas yra derinamas ir Kinect programinė įranga geba automatiškai sukalibruoti sensorių, priklausomai nuo fizinės aplinkos vartotojo aplinkos, prisitaikyti prie kambario esančių baldų ar kitų kliūčių. Kiekvienas žmogaus skeleto taško atstumas iki kameros yra išmatuojamas perduodant infraraudonųjų spindulių šviesą ir skaičiuojant

jos kelionės laiką pirmyn ir atgal kai atsispindi nuo žmogaus kūno. Vienu metu įmanomas iki šešių žmonių sekimas, tačiau tik dviejų žmonių judesiai bus fiksuojami pilnai, tai yra su 20-ies “sąnarių” (angl. joints) išskyrimu.

- Multidimensinis mikrofonas – jį sudaro keturi mikrofonai, kurie gali išskirti žmogaus balsą iš aplinkos triukšmo ir panaudoti jį kaip papildomą valdymo funkciją.

Kartu šie komponentai gali aptikti stebėti iki 48 skirtingų žmogaus taškų, 30 kartų per sekundę dažniu.



Pav. 1.1 Kinect jutiklis

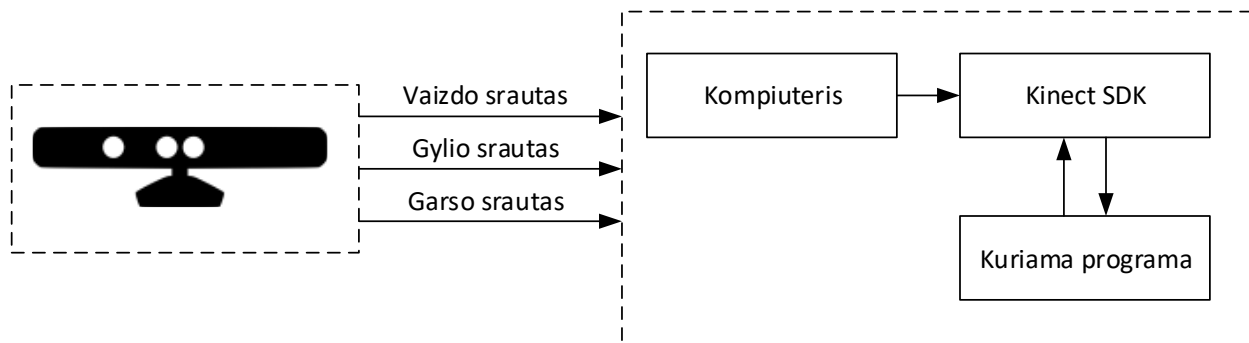
### 1.2.2 Programinė įranga

Tyrimui atlikti buvo naudojama pirmosios versijos Microsoft Kinect kamera, kuri buvo sukurta kaip priemonė žaisti žaidimus naudojant Xbox 360 konsolę [3]. Tačiau nepraėjus nei metams nuo kameros išleidimo, buvo išleistas Kinect SDK programinės įrangos kūrimo rinkinys (angl. software development kit) [4]. Jis suteikė galimybę programuotojams kurti programas Kinect jutikliui naudojant C++, C# ar Visual Basic kalbas. Kartu su šiuo SDK vartotojas gauna ir NUI – natūralios vartotojo sąsajos (angl. Natural User Interfaces) [5] biblioteką. Kinect kameros komunikacija su kuriama programa yra iliustruota 1.1 Pav.

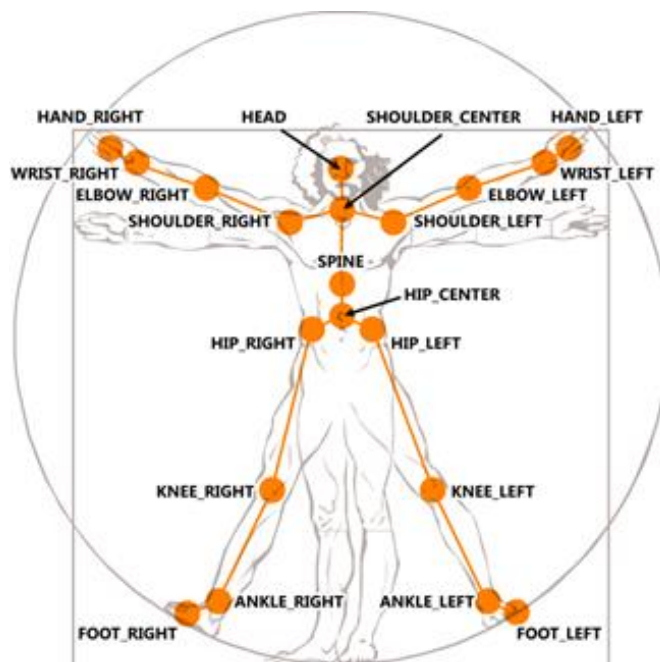
Kinect kamera kompiuteriui perduoda tokius duomenis:

- Vaizdo ir gylio srautą;
- Garso srautą;

- Reguliuojamą kameros pokrypį;
- 20 skirtingų „sąnarių“ (1.3 Pav.).



Pav. 1.2 Kinect sensoriaus komunikacija



Pav. 1.3 Kinect sensoriaus fiksuojami „sąnariai“

Industrinio roboto simuliacijai buvo naudojama oficiali ABB programinė įranga – Robotstudio [6]. Ši įranga skirta ne tik programuoti ABB industrinius robotus, tačiau suteikia galimybę pilnai simuluoti skirtingų modelių robotus kaip virtualias mašinas. Visa tai įvertinus, ši programinė įranga tampa puikiu įrankiu, skirtu testuoti tyrimo metu kuriamą sistemą.

### 1.3 Programinės įrangos pasirinkimas

Programinės įrangos kūrimui su Kinect kamera egzistuoja trys pagrindinės bibliotekos:

- Windows SDK [4]. Galima naudoti tik su Windows operacine sistema. Programavimo kalbų pasirinkimas apribotas iki C#, Visual Basic ir C++ (Visual Basic nepalaikoma nuo SDK 2.0 versijos). Nemokama tik nekomerciniam naudojimui. Turi įgyvendintą skeleto sekimo algoritmą.
- OpenNI. Galima naudoti ne Windows operacinėse sistemose. Yra galimi perkėlimai į kitas kalbas (angl. Wrappers), kas suteikia daugiau lankstumo. Atvirojo kodo [7].
- OpenKinect. Taip pat galima naudoti ne Windows operacinėse sistemose ir taipogi su kitomis programavimo kalbomis nei C# ir C++. Atvirojo kodo [8].

Šiam projektui buvo pasirinkta Windows SDK biblioteka, dėl turimo žmogaus skeleto (angl. skeleton) sekimo algoritmo, taip pat ir dėl puikios ir detalios dokumentacijos ir pakankamai aktyvios vartotojų bendruomenės.

Kinect programos kūrimui buvo naudojama „Visual Studio“ programavimo aplinka.

ABB yra viena didžiausių industrinių robotų gamintojų. Šiam projektui buvo pasirinkta dirbti su ABB robotu, kadangi ABB yra išleidusi „Robotstudio“ programinę įrangą, kurioje yra ne tik robotų programavimo aplinka, bet taip pat ir visi ABB robotai, kuriuos galima naudoti kaip virtualias mašinas ir simuliuoti jų darbą realiu metu.

Duomenų perdavimui iš Kinect į Robotstudio programinėje įrangoje veikiančią virtualų robotą buvo naudojama ABB PC SDK biblioteka [9]. Artimiausia alternatyva ABB PC SDK bibliotekai yra kita ABB biblioteka „Robotstudio SDK“, kuri naudojama kaip programinės įrangos Robotstudio add-in paketų kūrimo plėtinys. Pasirinktosios bibliotekos dokumentacija silpnoka, o bendruomenė nėra aktyvi.

### 1.4 Egzistuojantys sprendimai

Visiškai analogiškų sistemų nėra sukurta, tačiau yra kitais būdais įgyvendintų robotų ir žmogaus bendradarbiavimo sistemų.

### **1.4.1 Kuka**

KUKA – ši industrinių robotų gamintoja įgyvendino roboto ir žmogaus bendradarbiavimo užduotį naudojant didelio tikslumo ir greitaveikos sensorius bei pažangias programavimo technologijas. Taikant šią technologiją, robotai jaučia kai jų kelyje atsiranda net ir menka kliūtis (pavyzdžiui, žmogaus ranka) ir tuo metu sustoja [10].

### **1.4.2 Yaskawa**

YASKAWA – naudojant lazerinius skanerus zonų identifikavimui yra nustatomos ribos, į kurias įžengus ar įkišus galūnę, robotas sulėtėja iki saugaus 250mm/s greičio. Papildomai išskyrus kritinę zoną, robotą galima visiškai sustabdyti[11].

## **1.5 Situacijos Lietuvoje įvertinimas**

Nors Lietuva yra pakankamai inovatyvi valstybė, tačiau pramonėje industriniai robotai vis dar nėra dažnai sutinkami. Būtent dėl šios priežasties nėra naudojama jokių išmanių sistemų, įgalinančių robotų ir žmonių bendradarbiavimą saugiomis aplinkybėmis.

## **2. PROJEKTINĖ DALIS**

### **2.1 Projekto paskirtis**

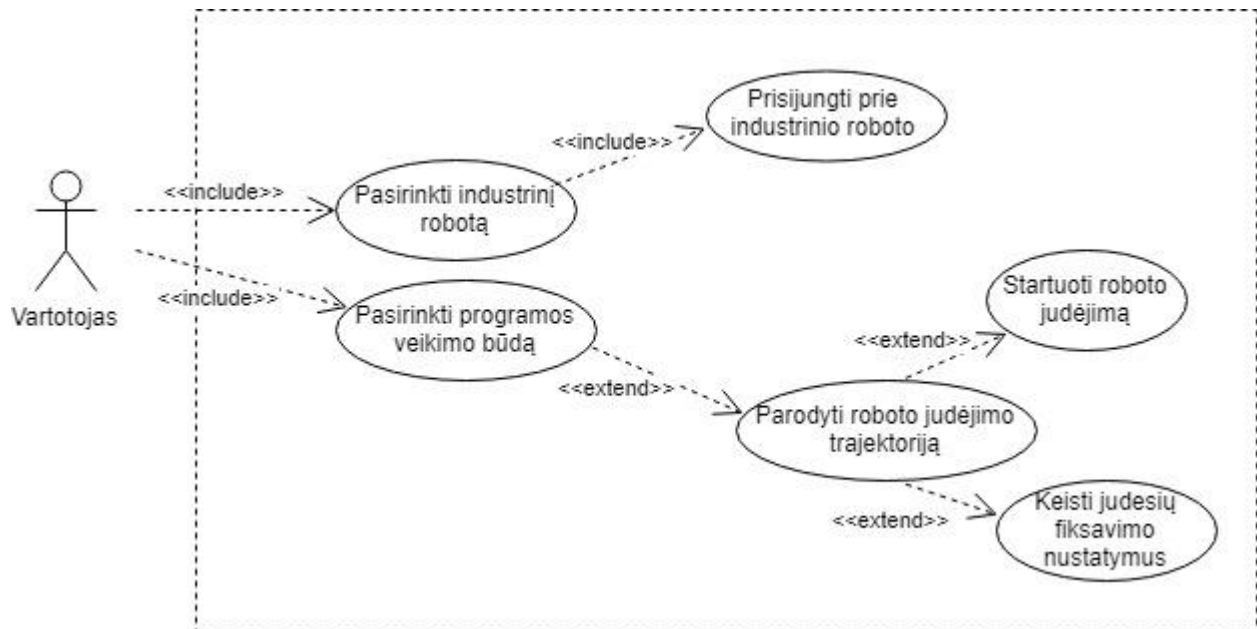
Projekto tikslas yra ištirti ar Kinect kamera gali būti panaudojama industrinio ABB roboto imitaciniam apmokymui. Produkto paskirtis yra įgalinti „Industry 4.0“ standartą [1] ir panaudoti lankstesnes priemones apmokyti industrinį robotą. Industrinio roboto apmokymas tokiu būdu leistų inžinieriams sutaupyti laiko programuojant robotus tais atvejais, kai roboto judėjimo koordinatės neprivalėtų būtų precizinio tikslumo. Taip pat, procesas reikalautų mažiau išlaidų, dėl nebūtinės didelės darbuotojų kompetencijos robotų programavime.

### **2.2 Sistemos paskirtis ir panaudos atvejai**

Kuriama programinė įranga bus instaliuojama kaip atskira programa Windows operacinėje sistemoje. Programinė įranga leis imitaciniu pobūdžiu apmokyti industrinį ABB robotą judėjimo koordinatėmis naudojant žmogaus judesius ir Kinect sensorių. Sistemos panaudos atvejai pavaizduoti 2.1 paveiksle.

Sistema leis:

- Prisijungti / atsijungti nuo norimo industrinio roboto, esančio vietiniame tinkle (angl. LAN);
- Užfiksuoti žmogaus rodomas trajektorijos koordinatas;
- Apmokyti ABB industrinį robotą judėjimo koordinatėmis.



Pav. 2.1 Sistemos panaudos atvejai

### 2.3 Bendri apribojimai

- Neturi būti naudojamos specialios pirštinės ar jutikliai, tvirtinami prie rūbų;
- Trajektorijos įvedimo pradžia/pabaiga neturi būti inicializuojama mygtuko paspaudimu;
- Programinė įranga veikia tik Windows operacinėse sistemose;
- Įdiegta Kinect SDK;
- Įdiegta Robotstudio programinė įranga;
- Stabilus veikimas kompiuteryje, kurio techniniai parametrai ne mažesni nei:
  - Operatyvioji atmintis: 2GB;
  - Procesoriaus taktinis dažnis 1.6 Ghz;
  - Dedikuota vaizdo plokštė.



## 2.4 Reikalavimai sistemai

Projektuojant sistemą buvo apibrėžti 6 funkciniai ir 5 nefunkciniai reikalavimai. Nefunkciniai reikalavimai nurodo ir apibrėžia programinės įrangos išvaizdą ir veikimą, todėl buvo atrinkti keli svarbiausi ir didžiausią naudą turintys reikalavimai:

**Lentelė 2.1** Reikalavimai sistemai

Reikalavimas#: 1	Reikalavimo tipas: 10	Įvykis/panaudojimo atvejis: -	
Aprašymas	Sistema turi leisti vartotojui pasirinkti tinkle esantį industrinį robotą		
Pagrindimas	Leis prisijungti / atsijungti prie industrinio roboto, kuris bus apmokomas imitaciniu būdu.		
Šaltinis	Užsakovas		
Tinkamumo kriterijus	Robotas vykdys judesius, kai startuosime roboto judėjimą.		
Užsakovo	5	Užsakovo	5
patenkinimas	nepatenkinimas		
Priklausomybės	Konfliktai		Nėra
Papildoma medžiaga	Robotstudio PC SDK dokumentacija		
Istorija	2017-2-07		

**Lentelė 2.2** Reikalavimai sistemai

Reikalavimas#: 2	Reikalavimo tipas: 10	Įvykis/panaudojimo atvejis: -	
Aprašymas	Sistema turi fiksuoti naudotojo rodomą roboto judėjimo trajektoriją.		
Pagrindimas	Leis naudotojui ranka parodyti norimą industrinio roboto judėjimo trajektoriją.		
Šaltinis	Užsakovas		
Tinkamumo kriterijus	Robotas atkartos vartotojo ranka parodytą trajektoriją.		
Užsakovo	5	Užsakovo	5
patenkinimas	nepatenkinimas		
Priklausomybės	Konfliktai		Nėra
Papildoma medžiaga	Microsoft Kinect SDK dokumentacija		
Istorija	2017-2-07		

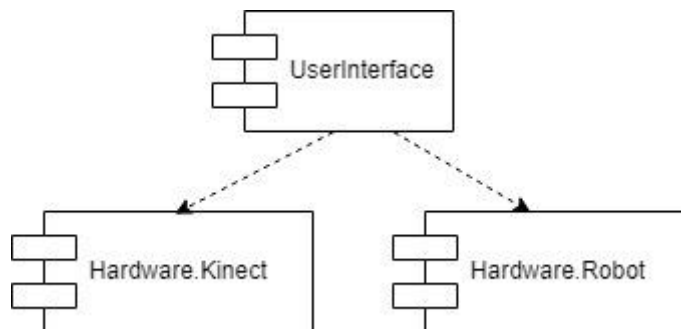
**Lentelė 2.3** Reikalavimai sistemai

Reikalavimas#: 3	Reikalavimo tipas: 10	Įvykis/panaudojimo atvejis: -	
Aprašymas	Perduoti koordinates robotui ir pradėti judėjimą ta trajektorija.		
Pagrindimas	Ranka įvesta trajektorija bus perduota industriniam robotui suprantamu formatu ir robotas bus startuojamas.		
Šaltinis	Užsakovas		
Tinkamumo kriterijus	Robotas judės perduotomis koordinatėmis.		
Užsakovo patenkinimas	5	Užsakovo nepatenkinimas	5
Priklausomybės		Konfliktai	Nėra
Papildoma medžiaga	Robotstudio PC SDK dokumentacija		
Istorija	2017-2-07		

## 2.5 Sistemos architektūra

Šio poskyrio tikslas yra atskleisti pagrindinius architektūrinius sistemos aspektus. Sistema yra suskirstyta į tris grupes: vartotojo sąsają, industrinio roboto sąsają ir Kinect sensoriaus sąsają.

### 2.5.1 Sistemos statinis vaizdas



**Pav. 2.2** Sistemos komponentų diagrama

## 2.5.2 „UserInterface“ komponentas

Komponente realizuota vartotojo sąsaja. Visa vartotojo sąsaja buvo sutalpinta viename lange (dviejuose skirtukuose). Šis paketas įgyvendina duomenų gavimą ir perdavimą į/iš kitų paketų, šiuo atveju „Hardware.Robot“ ir „Hardware.Kinect“.

*Atsakomybės:*

Komponentas realizuoja visą vartotojo sąsają. Taip pat teisingą duomenų, gautų iš Kinect jutiklio suformavimą ir perdavimą industriniam ABB robotui.

*Struktūra:*

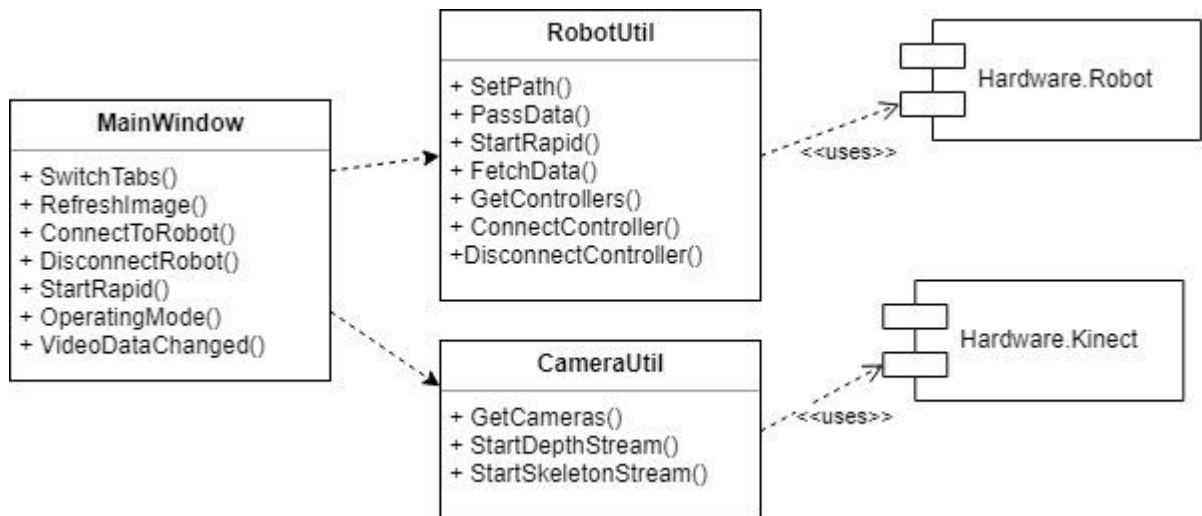
Komponentą sudarančios klasės pateiktos pav 2.3.

*Sąveika:*

Komponentas naudoja „Hardware.Robot“ ir „Hardware.Kinect“ paketus.

*Resursai:*

Naudojamas Microsoft.NET, ABB PC SDK ir Kinect SDK bibliotekos.



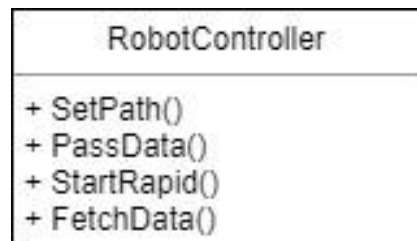
Pav. 2.3 „UserInterface“ klasių diagrama

- MainWindow - ši klasė atsakinga už vartotojo sąsają, kiekvieno mygtuko paspaudimą, parametro įvedimą ar skirtuko pasirinkimą. Kuo daugiau logikos, susijusios su Kinect jutiklio ir industrinio roboto duomenimis buvo stengiamasi iškelti atitinkamai į „CameraUtil“ ir „RobotUtil“ klases, todėl šioje klasėje veiksmai, susiję su duomenų apdorojimu, neatliekami.

- RobotUtil – šioje klasėje atliekami veiksmai su duomenimis, gautais iš industrinio roboto. Ši klasė yra tarsi tarpinis sluoksnis tarp klasės „MainWindow“ ir paketo „Hardware.Robot“.
- CameraUtil - šioje klasėje atliekami veiksmai su duomenimis, gautais iš Kinect jutiklio. Ši klasė yra tarsi tarpinis sluoksnis tarp klasės „MainWindow“ ir paketo „Hardware.Kinect“.

### 2.5.3 „Hardware.Robot“ paketas

Šiame poskyryje aprašytos pagrindinės paketo klasės ir jų metodai.



**Pav. 2.4** „RobotController“ klasė

RobotController – klasė yra atsakinga už duomenų gavimą iš roboto ir perdavimą jam. Kalbant apie duomenų perdavimą, turima galvoje perduodamas trajektorijos koordinatės, kurias vartotojas „įveda“ Kinect jutiklio pagalba. Metodas „SetPath“ atlieka pakeitimus su perduotomis koordinatėmis ir paverčia jas į industriniam robotui suprantamą formatą. „PassData“ metodas perduoda paruoštų koordinačių masyvą industriniam robotui. „StartRapid“ metodas startuoja industrinio roboto judėjimą perduotų koordinačių trajektorija. Metode „FetchData“ iš industrinio roboto yra paimama informacija, reikalinga komunikacijai ir teisingam duomenų perdavimui.



**Pav. 2.5** „RobotControllerExplorer“ klasė

RobotControllerExplorer – klasės metodas „GetControllers“ atsakingas už roboto valdymo bloką paiešką vietiniame interneto tinkle. Rasti valdymo blokai atvaizduojami lentelėje su tam tikrais, vartotojui aktualiais duomenimis. Metodas „ConnectController“ realizuoja

prisijungimą prie roboto valdymo bloko, o metodas „DisconnectController“ – atsijungimą, jeigu prie valdymo bloko jau yra prisijungta.

#### 2.5.4 „Hardware.Kinect“ paketas



Pav. 2.6 „KinectExplorer“ klasė

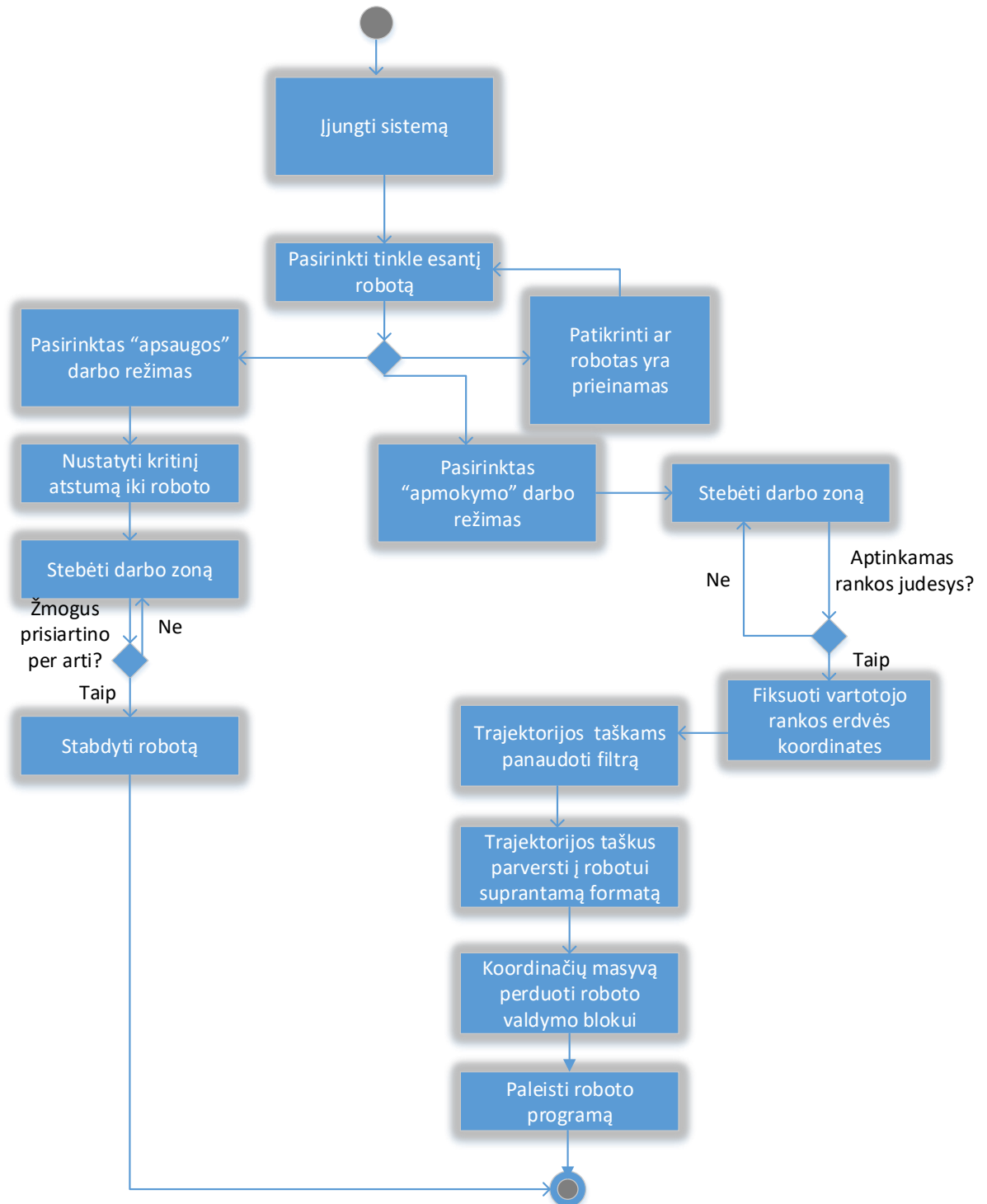
KinectExplorer – šios klasės metodas „GetCameras“ atsakingas už prie kompiuterio prijungto Kinect jutiklio radimą. Metodai „ConnectKinect“ ir „DisconnectKinect“ atitinkamai realizuoja prisijungimą prie jutiklio ir atsijungimą nuo jo.



Pav. 2.7 „KinectHelper“ klasė

KinectHelper – klasė atsakinga už vaizdo stebėjimą ir apdorojimą. Metodas „InitializeSensor“ inicializuoja kameros pradinius nustatymus, susijusius su vaizdo stebėjimu. Metodas „StartDepthStream“ realizuoja užfiksuoto vaizdo „gylio“ duomenų perdavimą. „StartSkeletonStream“ metodas realizuoja užfiksuoto vartotojo apdorotą vaizdą, kuriame išskiriamos jo kūno dalys ir prieš tai šiame darbe minėti „sąnariai“.

## 2.6 Veiklos diagrama



Pav. 2.8 Veiklos diagrama

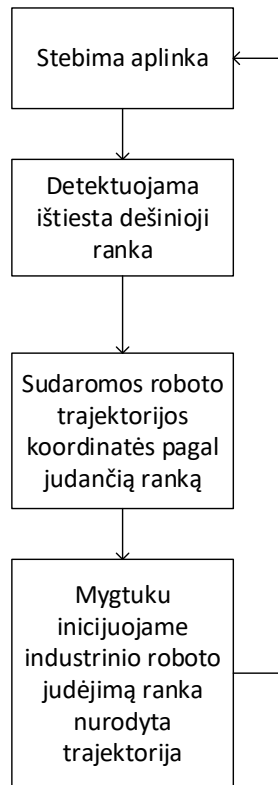
Sistemos veikla (Pav. 2.8) prasideda industrinio valdymo bloko ir programos veikimo režimo pasirinkimo skirtuke. Iš lentelės pasirenkamas tinkle esantis robotas ir programos veikimo režimas – apmokymo arba apsaugos. Prisijungus prie industrinio roboto ir pasirinkus veikimo režimą, pereinama į pagrindinį sistemos langą. Pasirinkus „apsaugos“ režimą privaloma nustatyti kritinę atstumo ribą, kurią peržengus industrinis robotas bus stabdomas. Pagrindinis sistemos veikimo režimas yra „apmokymo“. Šio režimo metu vartotojas gali apmokyti industrinį robotą judėjimo trajektorijomis. Tam reikia atsistoti prieš kamerą ir ištiesti dešiniąją ranką – programoje matomas visas fiksuojamas žmogaus kūnas ir indikacija (raudonas apskritimas ant vartotojo delno), jog koordinatės yra fiksuojamos. Koordinačių fiksavimas baigiamas kai vartotojas ištiestą ranką pritraukia prie savęs. Paspaudus mygtuką – „SendCoordinates“ koordinatės yra apdorojamos. Jos yra filtruojamos, konvertuojamos į industriniam robotui suprantamą formatą ir perduodamos kaip masyvas.

### **3. TYRIMO IR EKSPERIMENTINĖ DALYS**

#### **3.1 Esamas funkcionalumas**

Šio projekto metu sukurta programinė įranga skirta industrinio roboto apmokymui imitaciniu būdu naudojant Kinect jutiklį. Projekto kūrimą galima išskirti į kelis etapus, kurių metu jis buvo tobulinamas, priklausomai nuo problemų, su kuriomis buvo susidurta. Pirmojo etapo metu sukurta visa programinė įranga, kuri veikė taip (Pav. 3.1):

1. Pasirinkus Kinect jutiklį ir pradėjus aplinkos stabėjimą yra tikrinama ar fiksuojamo žmogaus dešinioji ranka yra ištiesta į priekį nuo krūtinės;
2. Jei dešinioji ranka yra ištiesta į priekį nuo krūtinės - traktuojama, kad vartotojas nori įvesti roboto judėjimo trajektoriją. Žmogaus dešinėsios rankos delno koordinatės yra fiksuojamos. Esant neištiestai dešiniajai rankai – koordinatės nebefiksuojamos;
3. Iš užfiksuotų koordinačių yra sudaromas tinkamo formato trajektorijos koordinačių masyvas industriniam robotui. Šio etapo metu iš rankos judesių yra nustatomos tik X, Y ir Z erdvės koordinatės, kuriose turi atsidurti industrinio roboto ašys;
4. Paspaudus starto mygtuką, sugeneruotas koordinačių masyvas yra perduodamas industriniam robotui ir yra startuojamas judesys perduotų koordinačių trajektorija.

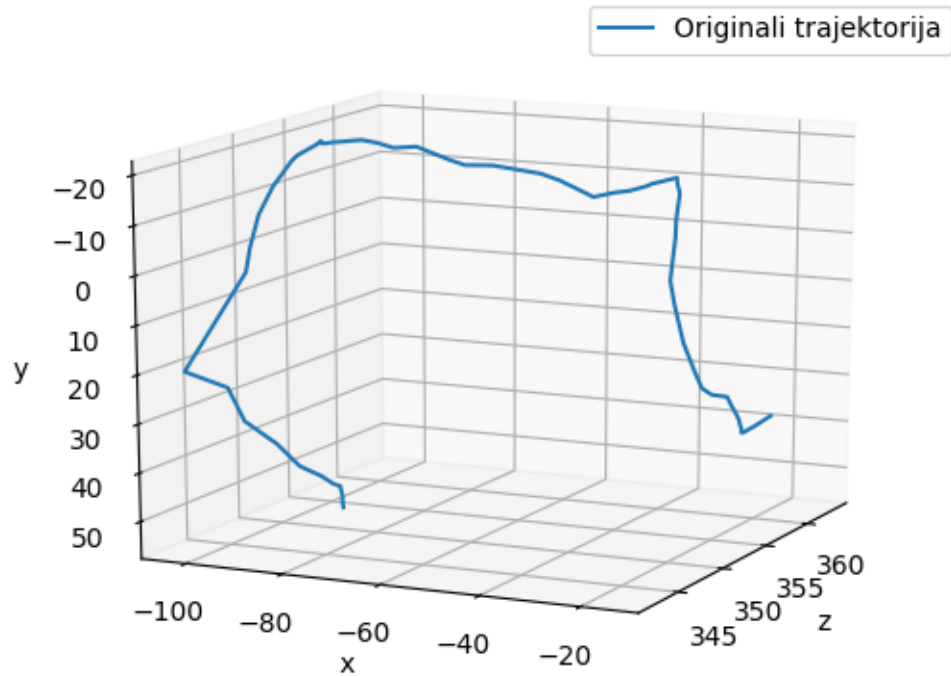


**Pav. 3.1** Supaprastintas programos tėkmės modelis

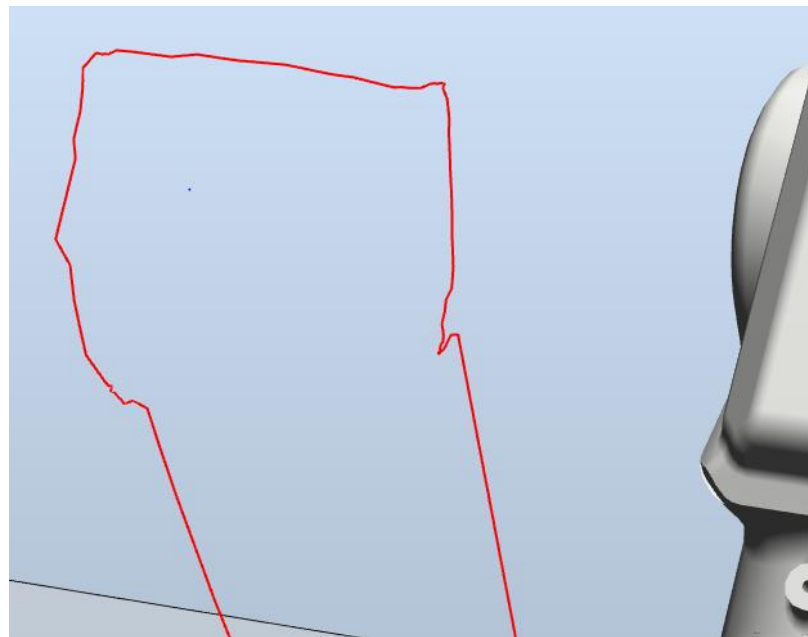
Grafiniam tyrimų duomenų atvaizdavimui buvo pasirinkta naudoti Python 2.7 programavimo kalbą su matplotlib, numpy ir scipy bibliotekomis [13], skirtomis duomenų apdorojimui ir atvaizdavimui. Toks sprendimas pasirinktas dėl to, kad programinė įranga ir jos bibliotekos yra nemokamos, turi didelę bendruomenę ir suteikia didelį funkcionalumą ir lankstumą apdorojant ir atvaizduojant duomenis. Taip pat, kai kuriems skaičiavimams buvo naudota MATLAB programinė įranga, dėl turimų specifinių funkcijų.

Pirmojo etapo metu sukurta programinė įranga veikė ir atliko norimus veiksmus, tačiau su tam tikrais niuansais ir nesklandumais. Lyginant perduodamų koordinačių trajektoriją erdvėje (Pav.3.2) ir roboto įvykdytą kelią perduotomis erdvės koordinatėmis (Pav.3.3) matome, kad trajektorijos vizualiai sutampa, kas reiškia, kad duomenys perduodami teisingai. Kitas dalykas ką galima pastebėti - tai yra „aštrūs“ trajektorijos keitimai, kurie atsiranda dėl rankos judėjimo erdvėje į visas puses ir Kinect užfiksuotų koordinačių triukšmų. Norint sumažinti šių reiškinių atsispindėjimą galutiniame rezultate reikia apdoroti trajektorijos koordinates.





**Pav. 3.2** Perduotų koordinačių trajektorija



**Pav. 3.3** Roboto įvykdyta trajektorija

Apdorojant koordinates nenorima prarasti duomenų teisingumo, tačiau norima trajektorijos pokyčius padaryti sklandesnius, panaikinti triukšmus, tačiau tuo pačiu nenorima didelio vėlavimo į trajektorijos pokyčius [15].

Sistemos užfiksuoti trajektorijos taškai, prieš perduodant industriniam robotui, yra išlyginami – tam panaudojamas Savitzky-Golay išlyginimo algoritmas [17], todėl sumodeliuota trajektorija taip pat buvo išlyginta. Sumodeliuota trajektorija yra matoma 3.4 paveiksle. Šio filtro veikimas paremtas daugianario mažiausių kvadratų lyginimu nustatyto dydžio judančiame lange. Aukšto lygio daugianaris ( $n = 4$ ) lemia didelį lyginimą be duomenų informacijos praradimo.

Duotuoju atveju naudoti tokie parametrai:

- Slenkančio lango dydis: 21;
- Daugianario lygis: 5;

Lyginant su slenkančio vidurkio lyginimo metodu, Savitzky-Golay filtras yra pranašesnis, kadangi jis sugeba išlaikyti tam tikras duomenų savybes (piko dydis, plotis), kurie dažniausiai yra prarandami naudojant slenkančio vidurkio filtrą.

Savitzky – Golay filtro aprašymas:

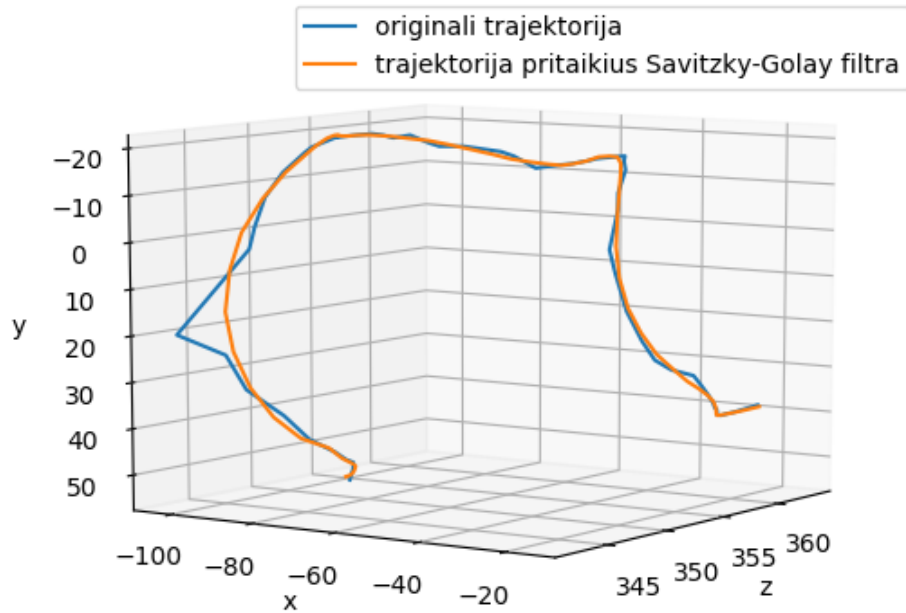
Duomenys susideda iš  $n$   $\{x_j, y_j\}$  taškų, kur  $x$  yra nepriklausomas kintamasis, o  $y$  – užfiksuota reikšmė.

$$Y_j = \sum_{i = -\frac{m-1}{2}}^{\frac{m-1}{2}} C_j y_{j+1}, \quad (1)$$

$$\frac{m-1}{2} \leq j \leq n - \frac{m-1}{2} \quad (2)$$

Kur  $m$  – sąsūkos koeficientas (angl. convolution),  $C_j$  – analizuojamieji duomenys.

Pritaikius filtrą trajektorija atrodo taip:



**Pav. 3.4** Perduota koordinačių trajektorija pritaikius Savitzky – Golay filtra

Galima pastebėti, kad duomenų teisingumas pakito minimaliai ir yra išlaikytas trajektorijos kelias, tačiau tuo pačiu yra išvengiama grubių trajektorijos pasikeitimų juos padarant sklandesniais, atsikratoma pavienių triukšmų.

Po pirmųjų dviejų etapų turimas sprendimas turi problemą, kad vartotojui yra neįmanoma nustatyti roboto ašių posūkio ketvirčių. Šešių ašių ABB roboto trajektorijos koordinatė (Pav. 3.5) yra sudaryta iš keturių pagrindinių dalių [12]:

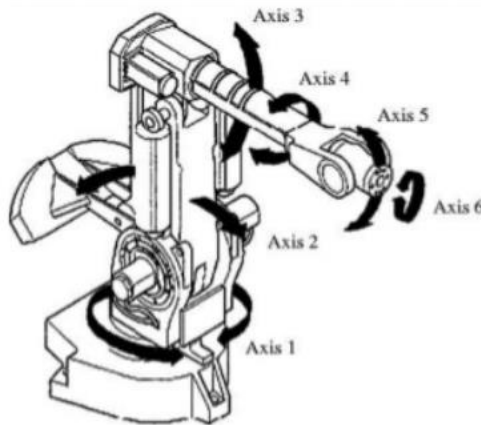
1. Roboto ašių pozicijos  $x, y, z$  koordinačių sistemoje. Pavyzdžio atveju [600, 500, 255.3];
2. Įrankio orientacijos. Pavyzdžio atveju [1, 0, 0, 0];
3. Roboto ašių konfigūracijos, kuri yra nurodyta ketvirčiais: 1 ir 4 a, 6 ašies.  $0-90^\circ$  atitinka 0,  $90^\circ-180^\circ$  atitinka 1 ir t.t. Pavyzdžio atveju [1, 1, 0, 0];
4. Papildomų loginių ašių pozicijos išreikštos laipsniais arba milimetrais. Parametras priklauso nuo ašių ir roboto tipo, kai kurios ašys būna neapibrėžtos. Pavyzdžio atveju [11, 12.3, 9E9, 9E9, 9E9, 9E9]

```
CONST robtarget p15 := [ [600, 500, 225.3], [1, 0, 0, 0], [1, 1, 0, 0], [ 11, 12.3, 9E9, 9E9, 9E9, 9E9] ];
```

**Pav. 3.5** Roboto trajektorijos koordinatės struktūra

Šiai problemai išspręsti buvo sugalvota panaudoti kairiąją žmogaus ranką, kuria būtų nustatomi šių trijų ašių ketvirčiai.

Naudojant kairę ir dešinę žmogaus rankas skirtingoms industrinio roboto ašims valdyti. Skirtingos rankos įgyvendina du skirtingus ašių valdymo režimus – linijinį (angl. linear) ir rotacinį (angl. radial). Kairė ranka atlieka rotacinius roboto ašių judesius (1, 4, 6 ašys) Pav. 3.6. Šios ašys pasirinktos todėl, nes šių ašių pozicijos laipsniais yra nurodomos kiekviename roboto trajektorijos taške.



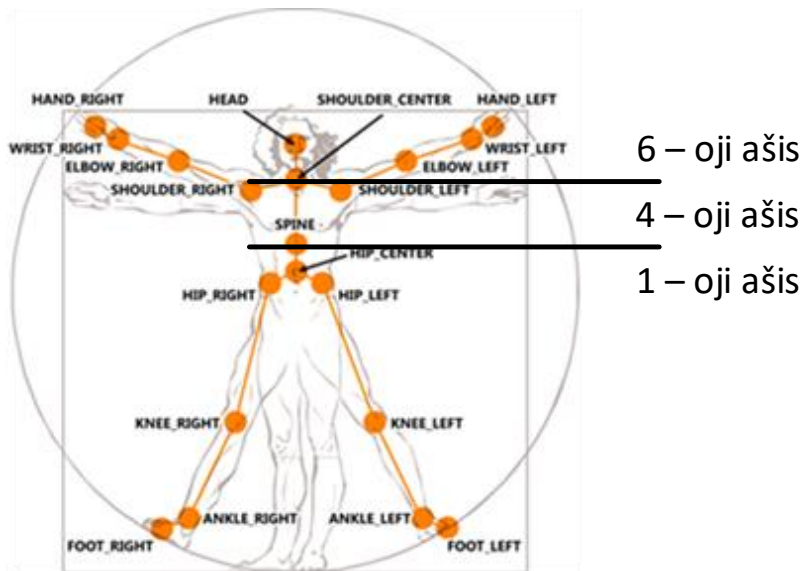
**Pav. 3.6** 6-ių ašių industrinis robotas

Šių trijų ašių valdymas yra išskiriamas naudojant kairės rankos plaštakos ir peties atstumo vienas nuo kito santykį. Keičiant kairės rankos plaštakos aukštį erdvėje (Kinect “Y” koordinačių ašis) yra pasirenkamos skirtingos roboto ašys, o keičiant rankos plaštakos poziciją kairėn ir dešinėn (Kinect “X” koordinačių ašis) – atitinkamos ašies padėtis erdvėje laipsniais (Pav 3.7).

Tokiu atveju, sudarinėjant roboto trajektorijos koordinates, reikia atsižvelgti ir įvertinti abiejų rankų pozicijas erdvėje. Dešinėsios rankos plaštakos užfiksuotos koordinatės atsispindi pirmojoje sudedamojoje dalyje, o kairiosios rankos plaštaka nurodo trečiosios dedamosios rezultata laipsniais. Roboto įrankio pozicijos keisti nepavyksta, nes tam reikia dar vienos kintančios dedamosios.

Įgyvendinus šį patobulinimą buvo susidurta su dar viena problema – industrinis robotas negali pasiekti visų nurodytų erdvės taškų. Tai nutinka kai koordinatės būna nurodytos neteisingai, su iškreiptais skaičiais. Tai atsitiko todėl, kad vartotojas gali detalčiau keisti koordinačių duomenis. Siekiant išspręsti šią problemą, buvo papildytas industrinio roboto programos kodas, kuris

patikrina kiekvieną iš mūsų programinės įrangos gautą koordinatę ir nustato ar ji yra pasiekiamą, ar ne. Atitinkamai, į ją yra keliaujama arba ji yra tiesiog praleidžiama.

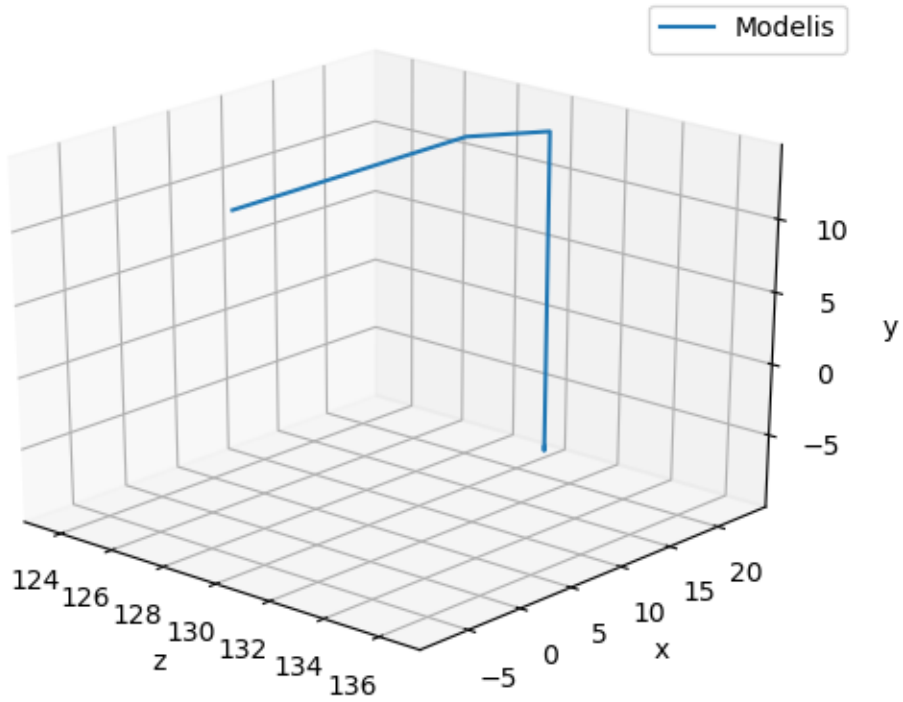


Pav. 3.7 Ašių valdymo zonos

Įgyvendinus trečiąjį etapą sistema tapo sąlyginai kompleksiška, o jos funkcionalumas yra tik iš dalies patenkinamas.

### 3.2 Esamos sistemos paklaidų įvertinimas

Sistemos paklaidų tyrimui įvertinti buvo sumodeliuota sistema, duomenų iš Kinect jutiklio analizei. Tyrimo metu buvo fiksuojamas identišką judesys po 10 kartų skirtingu atstumu nuo jutiklio: 1.3m, 2.3m ir 3m. Siekiant kiekvieną kartą užfiksuoti kuo mažiau besiskiriančią trajektoriją, buvo pasirinktas stacionarus objektas (dėžė, kurios aukštis yra 25cm, o plotis 33cm). Fiksuojant trajektoriją, ranka buvo vedama dėžės kraštinėmis nuo apatinio kampo aukštyn ir į šoną per visą dėžės plotį. Duomenų fiksavimas buvo atliktas dviem atvejais patalpoje: esant dirbtiniam apšvietimui ir esant tiesioginiams saulės spinduliams. Grafikuose matomos koordinatinių ašys yra išreikštos centimetrais. Paveiksle 3.8 matoma sumodeliuota „ideali“ trajektorija.



**Pav. 3.8** Sumodeliuota trajektorija

Paklaidos įvertinimui apskaičiuojama užfiksuotos trajektorijos vidutinė paklaida kiekvienai koordinatinių ašiai. Gauti duomenys surašyti 3.1, 3.3 ir 3.5 lentelėse.

**Lentelė 3.1** Paklaidos 1.3m atstumu

Testo Nr.	X ašies paklaida, cm	Y ašies paklaida, cm	Z ašies paklaida, cm
1.	0.92	0.68	0.74
2.	0.86	0.74	0.99
3.	1.12	0.62	0.74
4.	1.21	0.62	0.62
5.	1.01	0.59	0.86
6.	0.98	0.68	0.99
7.	0.99	0.76	0.58
8.	0.83	0.7	1.18
9.	1.14	0.75	1.1
10.	1.51	0.84	0.96

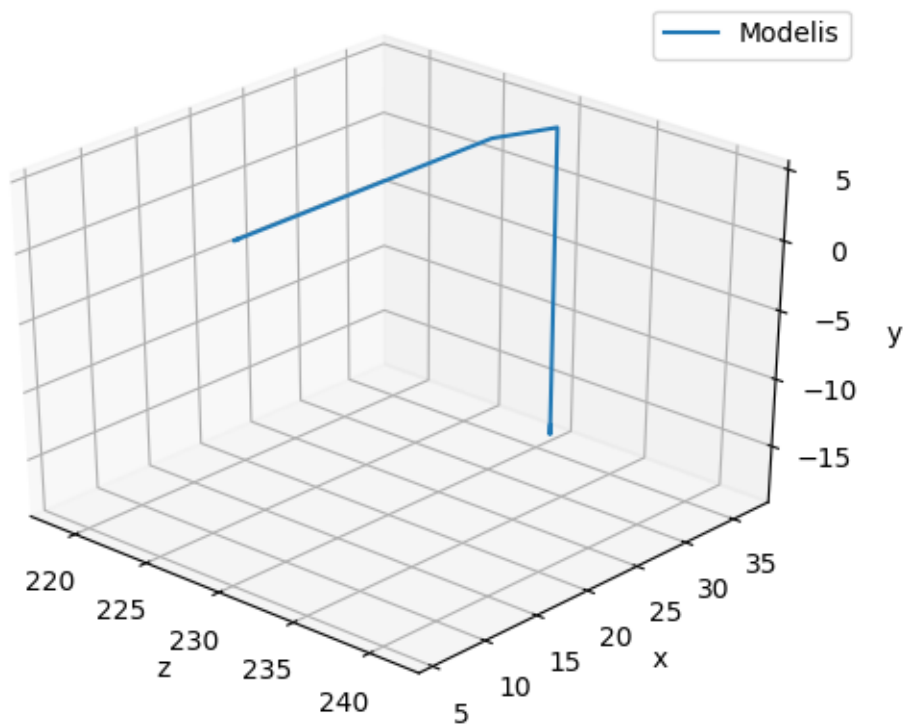
Prisiminus anksčiau minėtus dėžės, kurios kraštinėmis buvo atliekamas trajektorijos įvedimas, dydžio reikšmės ir remiantis 3.1 lentelėje surašytais paklaidų reikšmėmis galima paskaičiuoti procentais išreikštas minimalią ir maksimalią nuokrypio nuo sumodeliuotos trajektorijos reikšmes.

3.2 lentelėje pateikiamos minimalios ir maksimalios nuokrypių reikšmės, trajektoriją fiksuojant 1.3m atstumu.

**Lentelė 3.2** Minimalios ir maksimalios paklaidos 1.3m atstumu

	Min, cm	Max, cm	%Min	%Max
X ašis	0.83	1.51	2.52	4.56
Y ašis	0.59	0.84	2.36	3.36
Z ašis	0.58	1.18	0.45	0.91

Iš gautų rezultatų matome, kad gautos paklaidos yra nedidelės ir nesiekia net gi 5%. Tačiau matavimai buvo atlikti iš nedidelio atstumo – 1.3m. Antruoju atveju matavimai atlikti iš 2.3m atstumo.



**Pav. 3.9** Sumodeliuota trajektorija 2.3m atstumu

**Lentelė 3.3** Paklaidos 2.3m atstumu

Testo Nr.	X ašies paklaida, cm	Y ašies paklaida, cm	Z ašies paklaida, cm
1.	0.89	1.05	2.59
2.	0.68	0.92	1.23
3.	0.75	0.8	2.78
4.	1.46	0.82	0.63
5.	1.18	0.74	1.39
6.	1.06	0.84	0.71
7.	0.95	0.65	1.33
8.	0.87	0.73	1.6
9.	0.59	0.84	1.2
10.	0.58	0.78	1.79

Vėl gi, naudojant anksčiau minėtus dėžės, kurios kraštinėmis buvo atliekamas trajektorijos įvedimas, dydžio reikšmės ir remiantis 3.3 lentelėje surašytais paklaidų reikšmėmis paskaičiuojamos procentais išreikštos minimalaus ir maksimalaus nuokrypio nuo sumodeliuotos trajektorijos reikšmės.

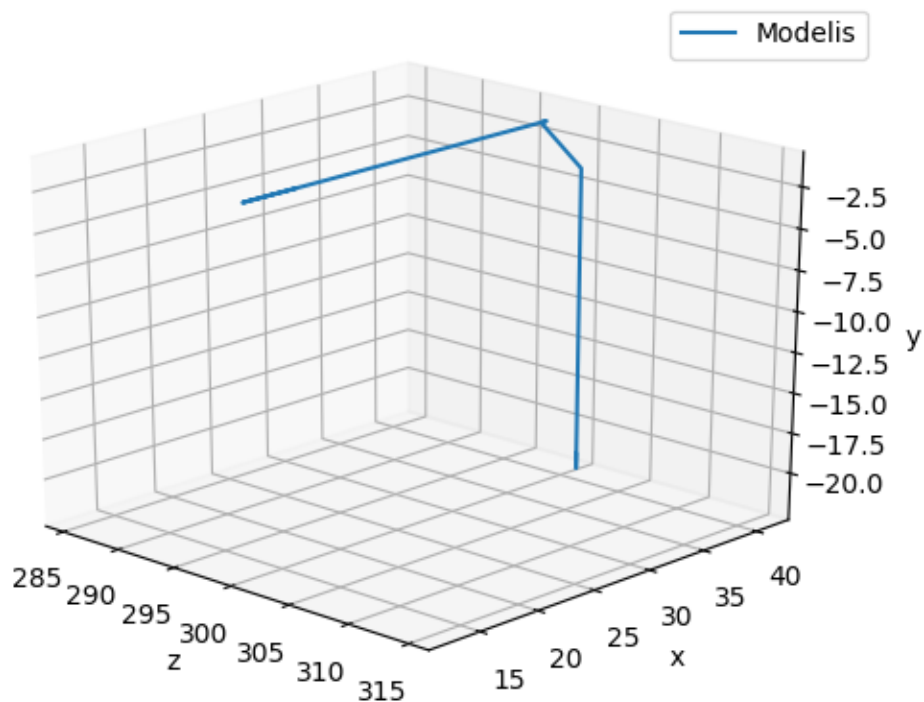
3.4 lentelėje pateikiamos minimalios ir maksimalios nuokrypių reikšmės, trajektoriją fiksuojant 2.3m atstumu.

**Lentelė 3.4** Minimalios ir maksimalios paklaidos 2.3m atstumu

	Min, cm	Max, cm	% <sub>Min</sub>	% <sub>Max</sub>
X ašis	0.59	1.46	1.79	4.42
Y ašis	0.65	1.05	2.6	4.2
Z ašis	0.63	2.78	0.27	1.2

Iš gautų rezultatų matome, kad gautos paklaidos vėl yra nedidelės ir nesiekia net gi 5%. Šį kartą skaičiavimai atlikti su duomenimis, kurie gauti matuojant iš 2.3m atstumo. Galima pastebėti, kad truputį padidėjo maksimalus nuokrypis Y ir Z ašimis, lyginant su rezultatais, gautais iš 3.2 lentelės. Trečiuoju atveju, bandymai atlikti iš 3m atstumo.





**Pav 3.10** Sumodeliuota trajektorija 3m atstumu

**Lentelė 3.5** Paklaidos 3m atstumu

Testo Nr.	X ašies paklaida, cm	Y ašies paklaida, cm	Z ašies paklaida, cm
1.	0.9	1.18	2.76
2.	0.83	1.35	2.69
3.	1.73	1.43	3.17
4.	0.82	1.47	2.2
5.	1.12	1.52	2.55
6.	0.94	1.8	2.86
7.	1.03	1.94	2.53
8.	1.04	1.45	2.17
9.	1.27	1.21	2.33
10.	0.96	1.32	1.9

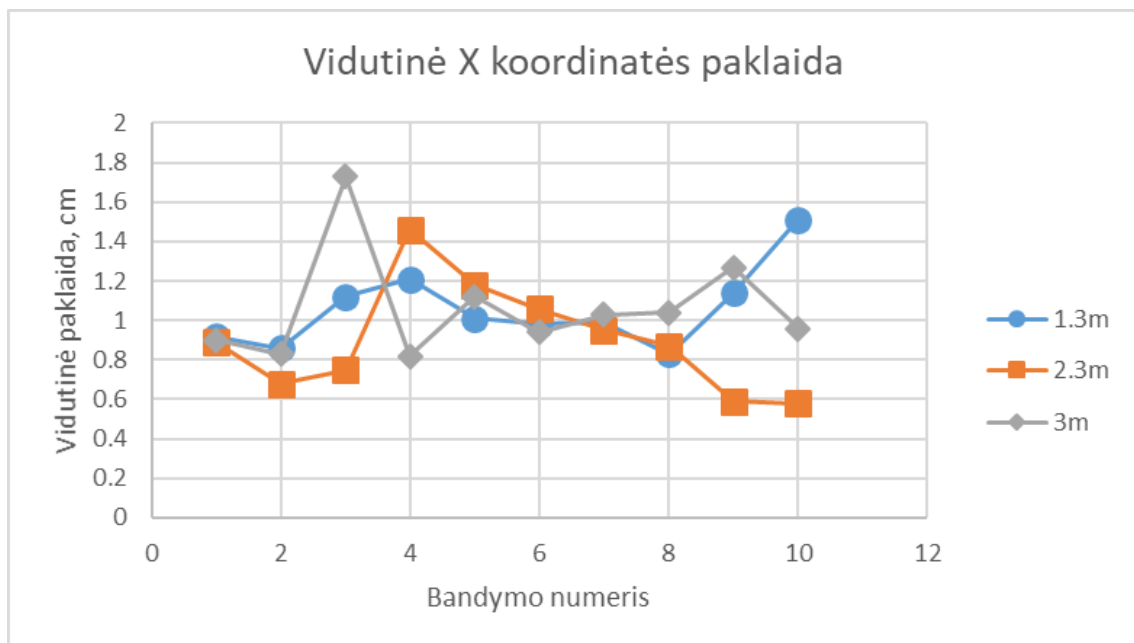
Remiantis 3.5 lentelėje surašytomis paklaidų reikšmėmis paskaičiuojamos procentais išreikštos minimalaus ir maksimalaus nuokrypio nuo sumodeliuotos trajektorijos reikšmės.

Lentelė 3.6 pateikiamos minimalios ir maksimalios nuokrypių reikšmės, trajektoriją fiksuojant 3m atstumu.

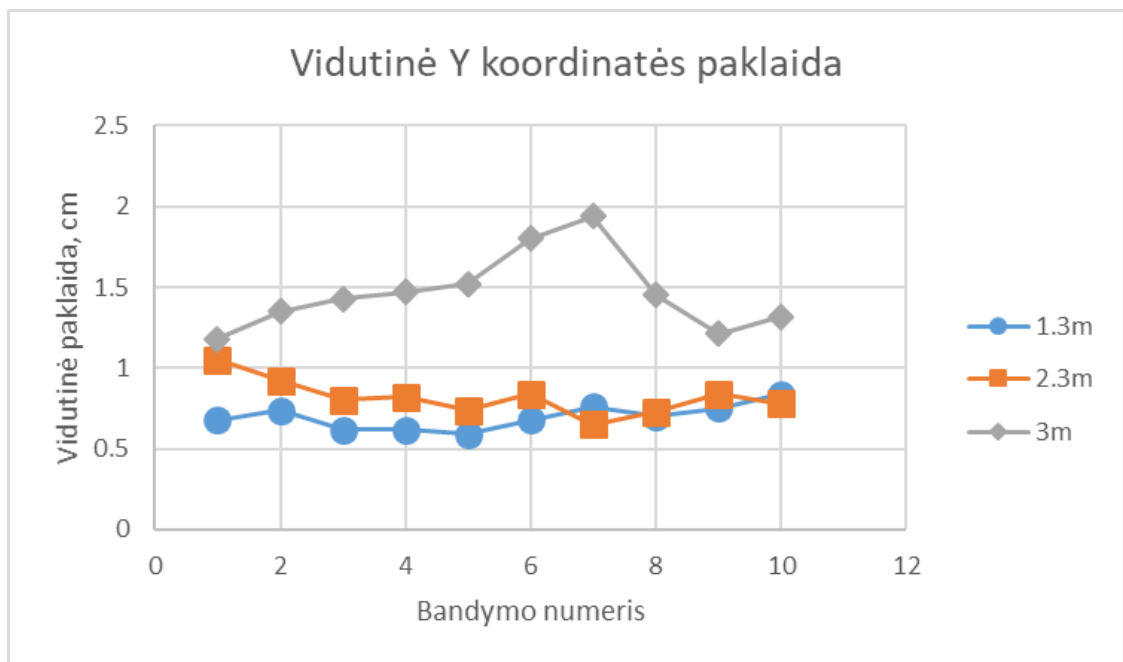
**Lentelė 3.6** Minimalios ir maksimalios paklaidos 3m atstumu

	Min, cm	Max, cm	%Min	%Max
X ašis	0.82	1.73	2.48	5.24
Y ašis	1.18	1.94	4.72	7.76
Z ašis	1.9	3.17	0.63	1.01

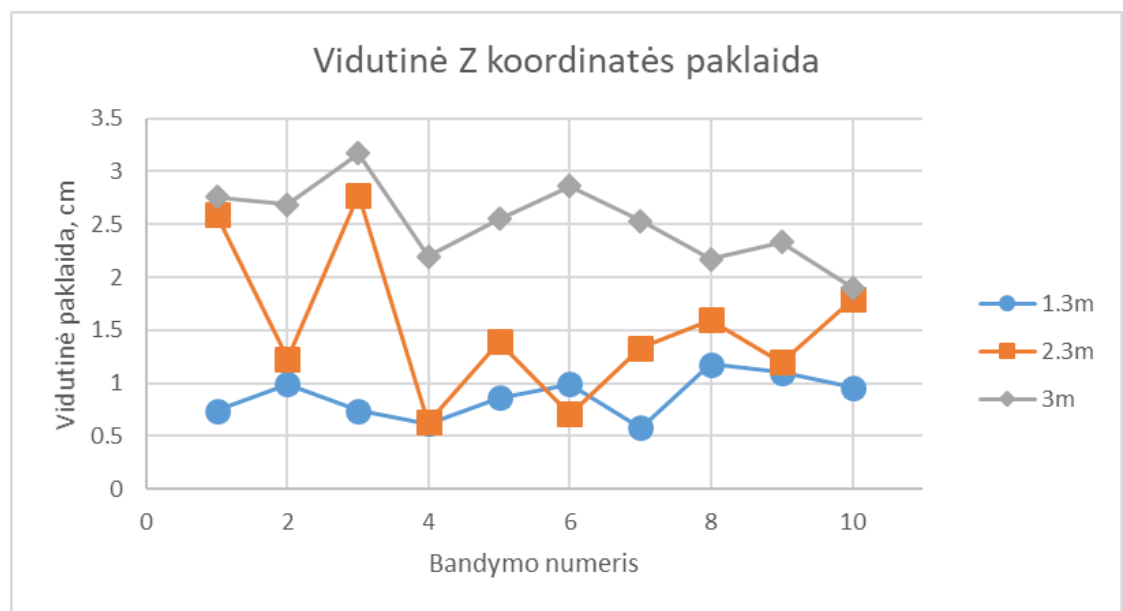
Šį kartą, atlikus skaičiavimus su duomenimis, gautais matuojant iš 3m atstumo, gautos paklaidos viršija 5%. Lyginant su rezultatais, gautais iš 3.2 ir 3.4 lentelių, galima pastebėti, kad padidėjo minimali ir maksimali paklaida X ir Y koordinatinių ašimis. Minimali paklaida taip pat padidėjo Z ašimi, tačiau nežymiai.



**Pav. 3.11** Vidutinė X koordinatės paklaida 10-ties bandymų metu



**Pav. 3.12** Vidutinė Y koordinatės paklaida 10-ties bandymų metu



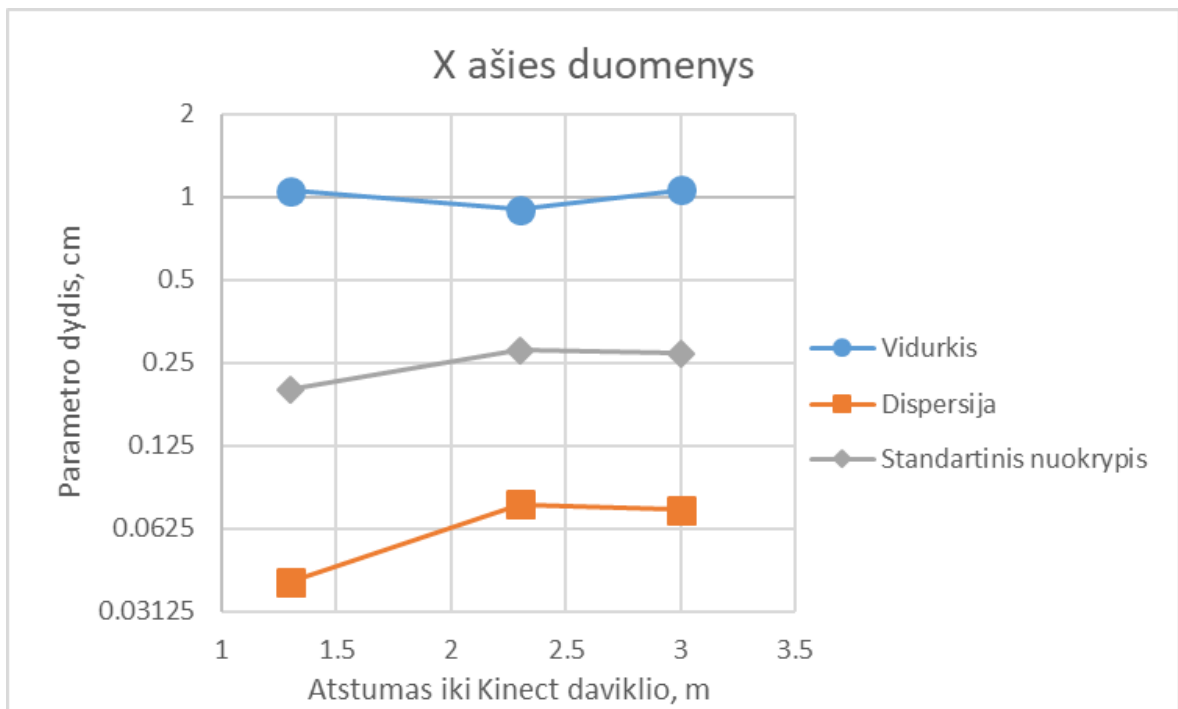
**Pav. 3.13** Vidutinė Z koordinatės paklaida 10-ties bandymų metu

Apibendrinus rezultatus, iš procentinių paklaidų išraiškų galima pastebėti, kad X ir Y ašių paklaidos yra panašios, tačiau Z ašis stipriai išsiskiria su maža procentinės paklaidos reikšme. Tačiau žiūrint į 3.13 paveikslą galima matyti, kad šios koordinatės ašies paklaida savo vidutiniu dydžiu nėra mažesnė už paklaidas X ir Y ašimis. Procentinė paklaida Z ašimi yra maža, kadangi ji skaičiuojama naudojant atstumą iki jutiklio, kuris yra sąlyginai didelis.

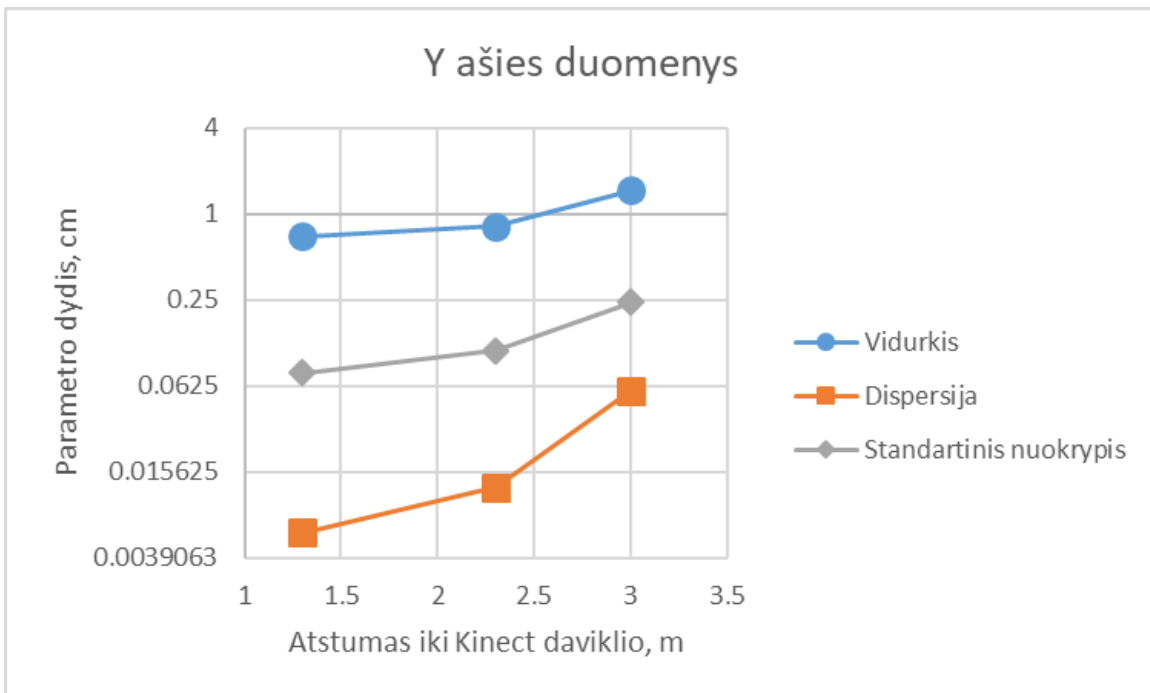
Iš turimų duomenų galima paskaičiuoti kitas vertingas statistines charakteristikas, kurios padės geriau suvokti, kokio dydžio paklaidos galima tikėtis atliekant kitus matavimus. Skaičiuojamos šios charakteristikos: paklaidos vidurkis, dispersija ir standartinis nuokrypis. Gauti duomenys pavaizduoti 3.7 lentelėje.

**Lentelė 3.7** Paklaidos vidurkis, dispersija ir standartinis nuokrypis

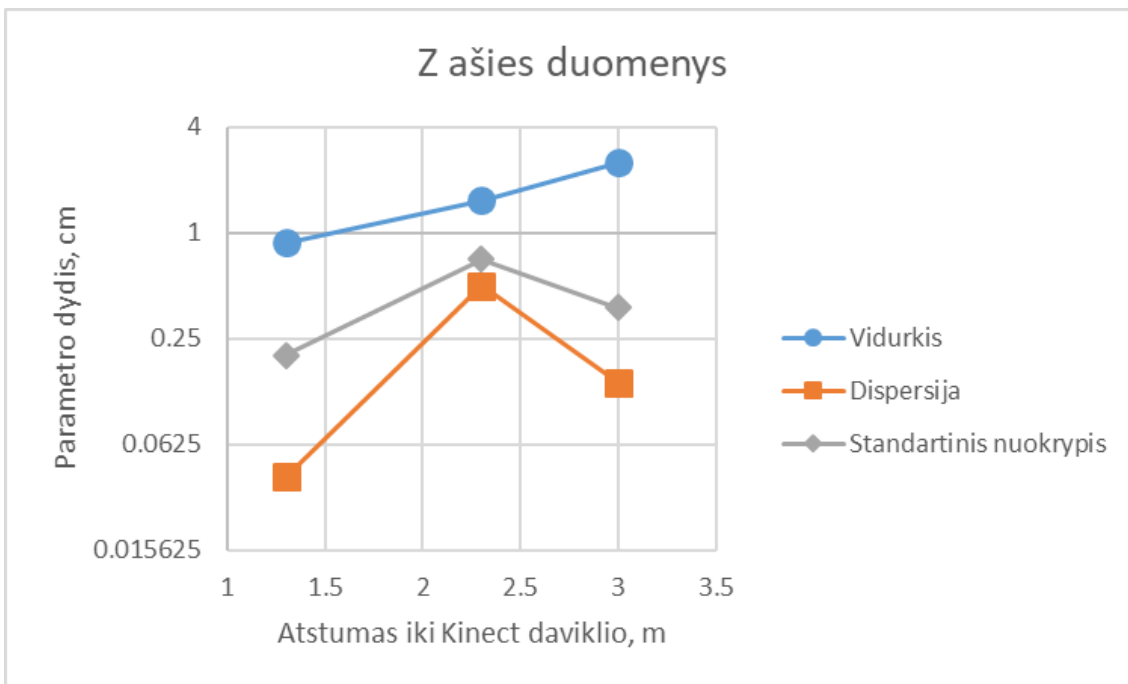
	1.3m atstumas			2.3m atstumas			3m atstumas		
	X	Y	Z	X	Y	Z	X	Y	Z
Paklaidos vidurkis, cm	0.876	1.057	0.698	1.525	0.901	0.817	2.516	1.064	1.467
Dispersija, cm	0.041	0.040	0.006	0.500	0.077	0.012	0.141	0.073	0.058
Standartinis nuokrypis, cm	0.202	0.200	0.077	0.707	0.277	0.110	0.376	0.270	0.241



**Pav. 3.14** X ašies koordinatų paklaidos vidurkis, dispersija ir standartinis nuokrypis



**Pav. 3.15** Y ašies koordinacių paklaidos vidurkis, dispersija ir standartinis nuokrypis



**Pav. 3.16** Z ašies koordinacių paklaidos vidurkis, dispersija ir standartinis nuokrypis

Remiantis 3.14, 3.15 ir 3.16 lentelėse atvaizduotais duomenimis, galima pastebėti, kad esant mažiausiam atstumui, kuriuo buvo atlikti bandymai (1.3m) paklaidos vidurkis, dispersija bei standartinis nuokrypis visais atvejais yra mažiausias. Microsoft nėra pateikusi oficialių duomenų apie pirmosios versijos Kinect jutiklio tikslumą ir paklaidas. Tačiau remiantis bendruomenės ir kitų tyrimų duomenimis [11], yra teigiama, kad šio Kinect sensoriaus tikslumas bene eksponentiškai mažėja didėjant atstumui iki jutiklio. Iš 3.15 ir 3.16 paveikslėlių galima sutikti, kad tokia teorija nėra klaidinga, tačiau šiuo atveju tikslumas mažėjo kiek mažiau nei eksponentiškai.

Kitu etapu buvo bandoma imituoti tokią pačią roboto judėjimo trajektoriją, tik kitokiomis sąlygomis – esant tiesioginiams saulės spinduliams.

Kinect gylio jutiklis veikia infraraudonųjų spindulių pagalba, o tai reiškia, kad tiesioginiai saulės spinduliai gali padaryti didelę įtaką užfiksuotiems rezultatams. Kinect gylio kamera projektuoja tankų infraraudonųjų taškų masę ir fiksuoja juos infraraudonųjų spindulių kamera, taip sudarydama „gylis“ vaizdą [9]. Atstumas iki kiekvieno pikselio (gylis) yra nustatomas atsižvelgiant į taškų žemėlapių iškraipymus. Kinect gylio kamera gražina 320x240 rezoliucijos 16 bitų paveikslėlį (programiškai šis paveikslėlis išdidinamas iki 640x480 rezoliucijos), kurio kiekvieno pikselio reikšmė atspindi realaus pasaulio atstumą iki Kinect jutiklio. Gautą gylio žemėlapią galima iš karto panaudoti arba konvertuoti jį į taškų debesį (angl. point cloud). Kadangi gylio informacijai gauti naudojami infraraudonųjų spindulių taškai yra ganėtinai neryškūs, Kinect IR kamera jų neatpažįsta, jeigu yra per daug infraraudonos šviesos aplinkoje (šiuo atveju tiesioginių saulės spindulių).

Visa tai pasitvirtino bandant išskirti žmogaus siluetą patalpoje, apšviestoje tiesioginiais saulės spinduliais.



**Pav. 3.17** Kinect gylio jutiklio matomas vaizdas esant tiesioginiams saulės spinduliams

Iš to galima daryti išvadą, kad pirmosios versijos Kinect jutiklis negali būti naudojamas ten, kur yra daug tiesioginių saulės spindulių, kadangi jie pernelyg stipriai sumažina vaizdo kokybę.

Įvertinus tai, buvo atlikti bandymai pavakario metu, kai saulės šviesa nebuvo tokia intensyvi, tačiau jos poveikis žmogaus atpažinimo kokybei vis dar buvo pastebimas. Gauti rezultatai pateikiami 3.8, 3.10 ir 3.12 lentelėse.

Gauti tokie duomenys:

**Lentelė 3.8** Paklaidos 1.3m atstumu

Testo Nr.	X ašies paklaida, cm	Y ašies paklaida, cm	Z ašies paklaida, cm
1.	3.54	3.91	4.55
2.	3.82	4.46	4.99
3.	8.72	5.38	6.91
4.	6.27	4.25	5.8
5.	5.41	4.25	4.88
6.	3.88	2.72	5.12
7.	5.98	3.1	5.03
8.	8.73	3.53	5.58
9.	3.8	5.11	5.04
10.	8.71	2.84	4.95

Vėl gi, procentinėms minimalaus ir maksimalaus nuokrypio nuo sumodeliuotos trajektorijos paklaidų reikšmėms apskaičiuoti, naudojami trajektorijoms fiksuoti naudotos dėžės matmenys (aukštis 25cm, plotis 33cm) ir 3.9 lentelėje pateikiamos minimalios ir maksimalios nuokrypių reikšmės, trajektoriją fiksuojant 1.3m atstumu ir esant tiesioginiams saulės spinduliams.

**Lentelė 3.9** Minimalios ir maksimalios paklaidos 1.3m atstumu

	Min, cm	Max, cm	%Min	%Max
X ašis	3.54	8.73	10.73	26.45
Y ašis	2.72	5.38	8.24	21.52
Z ašis	4.55	6.91	3.5	5.32

Iš gautų rezultatų galima pastebėti, kad apskaičiuotos paklaidos yra kur kas didesnės, lyginant su duomenimis iš 3.2, 3.4 ir 3.6 lentelių.

3.10 lentelėje pateikiamos ašių paklaidų reikšmės, trajektoriją fiksuojant 2.3m atstumu ir esant tiesioginiams saulės spinduliams.



**Lentelė 3.10** Paklaidos 2.3m atstumu

Testo Nr.	X ašies paklaida, cm	Y ašies paklaida, cm	Z ašies paklaida, cm
1.	5.98	5.49	3.11
2.	5.46	1.83	1.1
3.	8.85	3.17	1.29
4.	8.72	5.41	5.96
5.	9.34	6.08	3.43
6.	9.22	3.43	4.45
7.	7.39	3.16	2.3
8.	6.73	3.14	2.72
9.	6.88	2.72	2.44
10.	7.51	3.06	2.44

3.11 lentelėje pateikiamos minimalios ir maksimalios nuokrypių reikšmės, trajektoriją fiksuojant 2.3m atstumu ir esant tiesioginiams saulės spinduliams.

**Lentelė 3.11** Minimalios ir maksimalios paklaidos 2.3m atstumu

	Min, cm	Max, cm	% <sub>Min</sub>	% <sub>Max</sub>
X ašis	5.46	9.34	16.54	28.3
Y ašis	1.83	6.08	7.32	24.32
Z ašis	1.1	5.96	0.47	2.59

Iš 3.11 lentelės duomenų matoma, kad paklaidos vėl gi yra didelės ir siekia net gi ketvirdalį trajektorijos poslinkio X ir Y ašimis.

3.12 lentelėje pateikiamos ašių paklaidų reikšmės, trajektoriją fiksuojant 3m atstumu ir esant tiesioginiams saulės spinduliams.

**Lentelė 3.12** Paklaidos 3m atstumu

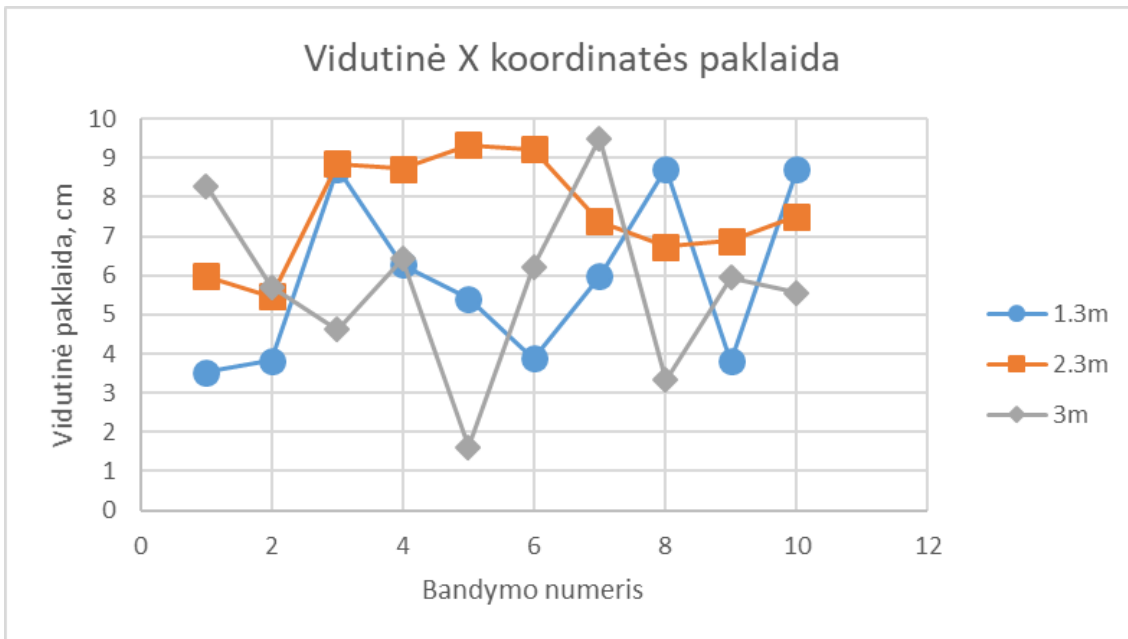
Testo Nr.	X ašies paklaida, cm	Y ašies paklaida, cm	Z ašies paklaida, cm
1.	8.28	6.09	5.23
2.	5.71	4.27	4.3
3.	4.62	4.77	5.74
4.	6.43	5.03	4.47
5.	1.61	3.73	4.92
6.	6.22	5.09	8.8
7.	9.51	4.38	6.05
8.	3.33	4.33	6.07
9.	5.94	4.24	4.15
10.	5.55	4.41	5.17

3.13 lentelėje pateikiamos minimalios ir maksimalios nuokrypių reikšmės, trajektoriją fiksuojant 3m atstumu ir esant tiesioginiams saulės spinduliams.

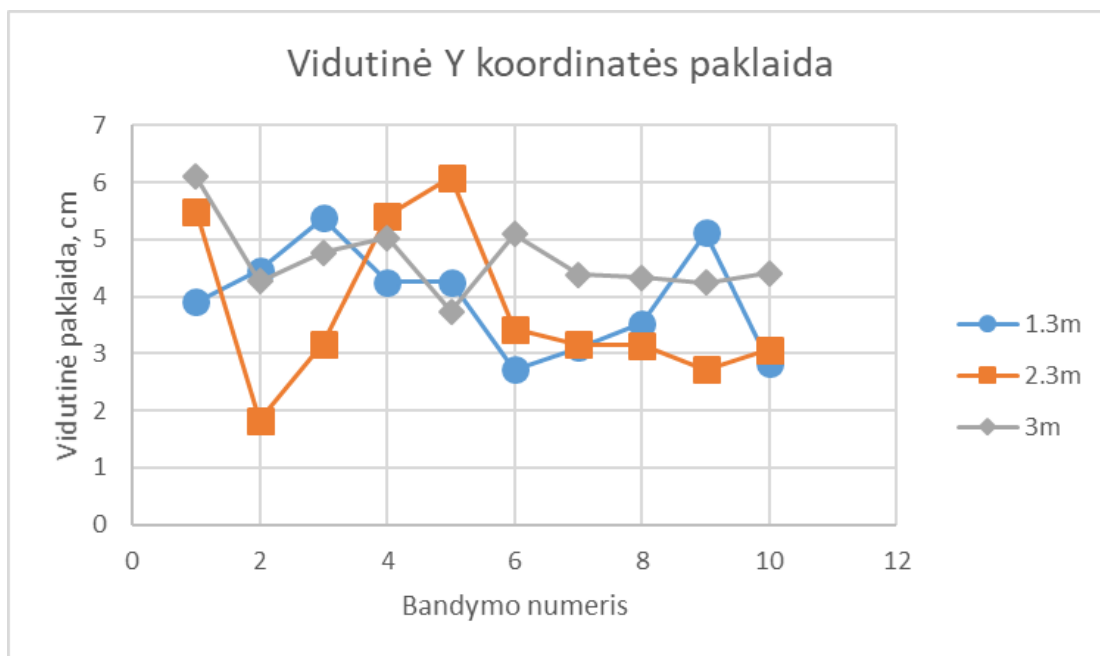
**Lentelė 3.13** Minimalios ir maksimalios paklaidos 3m atstumu

	Min, cm	Max, cm	%Min	%Max
X ašis	1.61	8.28	4.88	25.1
Y ašis	3.73	6.09	14.92	24.36
Z ašis	4.3	8.8	1.43	2.93

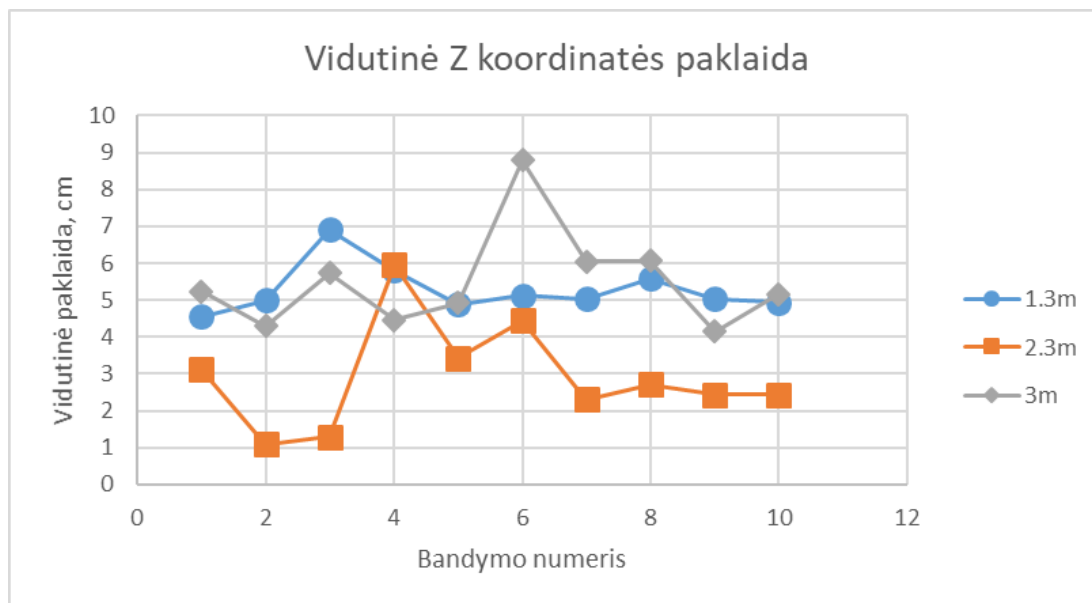
Lyginant 3.13 lentelės rezultatus su 3.9 ir 3.11 lentelių duomenimis, matosi, kad didelio skirtumo nėra – duomenys panašiu dydžiu iškraipyti tiesioginių saulės spindulių, nesvarbu kokiu atstumu nuo Kinect sensoriaus atliekami matavimai.



**Pav. 3.18** Vidutinė X koordinatės paklaida 10-ties bandymų metu



**Pav. 3.19** Vidutinė Y koordinatės paklaida 10-ties bandymų metu



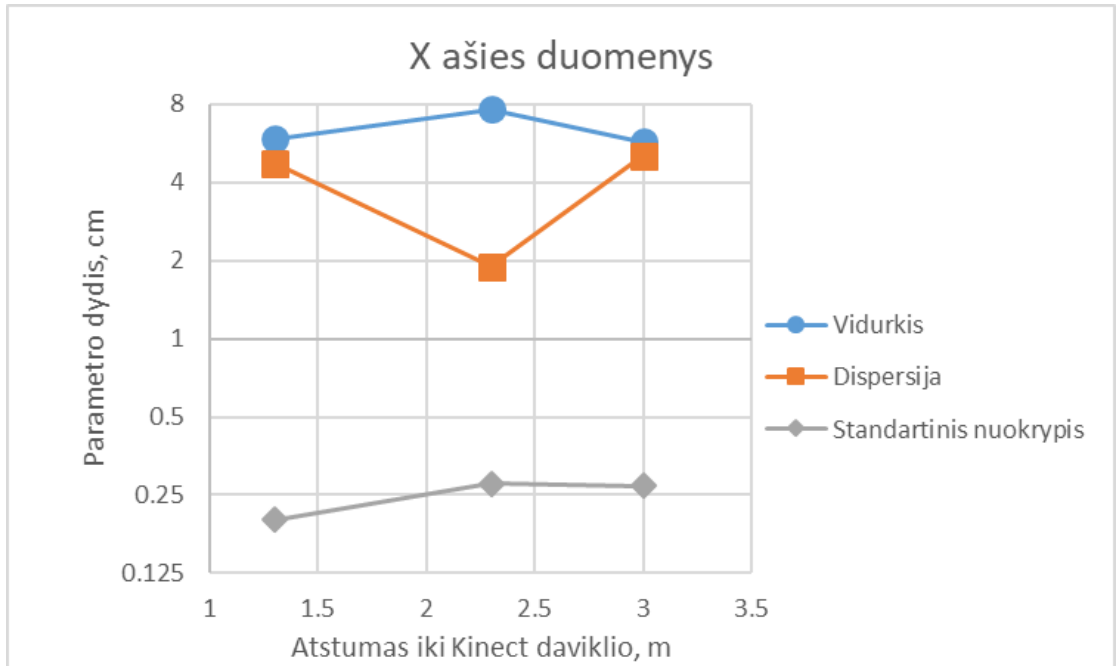
**Pav. 3.20** Vidutinė Z koordinatės paklaida 10-ties bandymų metu

Z koordinatėms ašimi procentiniai paklaidos skaičiavimai vėl atrodo nedideli, tačiau taip yra dėl anksčiau minėtos priežasties (sąlyginai didelio atstumo iki Kinect jutiklio). 3.18, 3.19 ir 3.20 paveikslai atvaizduoja vidutinę paklaidą. Iš grafinių duomenų matosi, kad paklaidos dydžiui mažai įtakos turi atstumas iki Kinect jutiklio – paklaidos dydis visais atvejais yra sąlyginai didelis. Visa tai tik patvirtina, kad Kinect jutiklis nėra pritaikytas naudoti ten, kur yra tiesioginių saulės spindulių tikimybė.

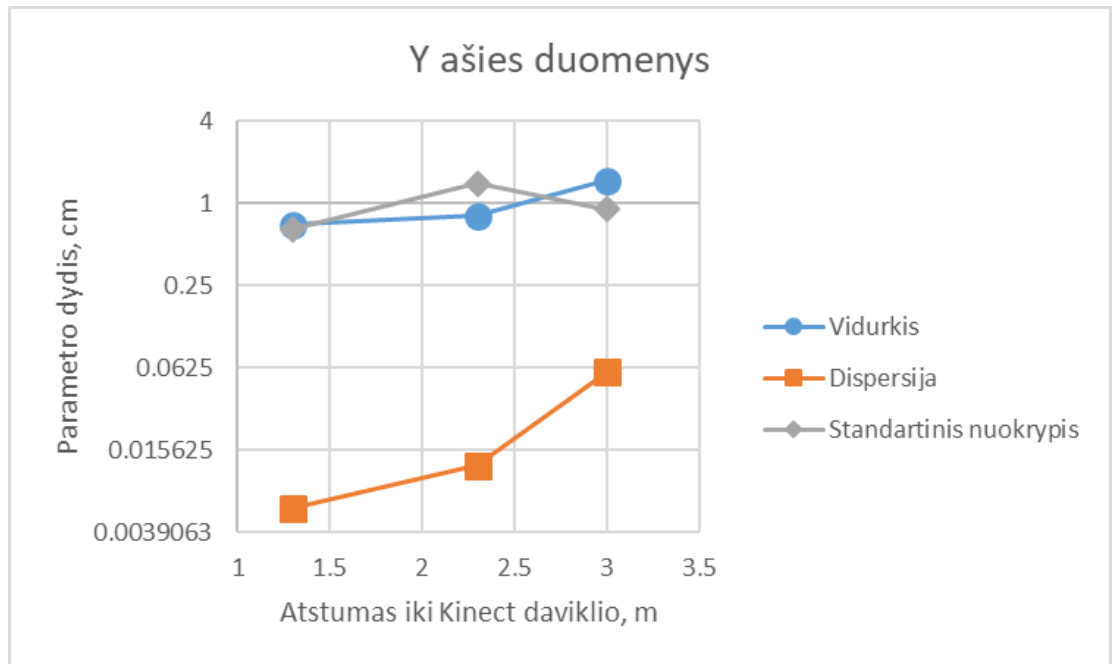
Iš turimų duomenų vėl galima paskaičiuoti paklaidos vidurkį, dispersiją ir standartinį nuokrypį. Gauti duomenys pavaizduoti 3.14 lentelėje.

**Lentelė 3.14** Paklaidos vidurkis, dispersija ir standartinis nuokrypis

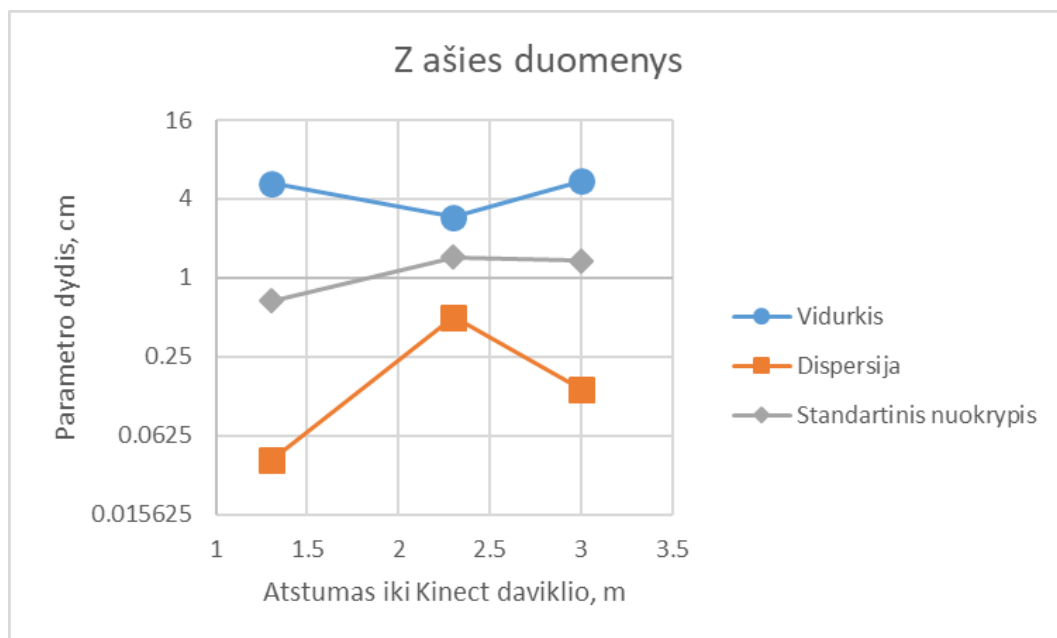
	1.3m atstumas			2.3m atstumas			3m atstumas		
	X	Y	Z	X	Y	Z	X	Y	Z
Paklaidos vidurkis, cm	5.285	5.886	3.955	2.924	7.608	3.749	5.49	5.72	4.634
Dispersija, cm	0.450	4.736	0.831	2.081	1.889	1.955	1.824	5.063	0.424
Standartinis nuokrypis, cm	0.671	2.176	0.911	1.443	1.374	1.398	1.351	2.250	0.651



**Pav. 3.21** Vidutinė Y koordinatės paklaida 10-ties bandymų metu



**Pav. 3.22** Vidutinė Y koordinatės paklaida 10-ties bandymų metu



**Pav. 3.23** Vidutinė Y koordinatės paklaida 10-ties bandymų metu

Kadangi šiuo atveju duomenys buvo iškraipyti tiesioginių saulės spindulių, nėra matomos atvirkščios priklausomybės tarp atstumo iki jutiklio ir jutiklio tikslumo, kuris buvo pastebėtas iš 3.13, 3.14 ir 3.15 lentelių.

### 3.3 Koreliacijos skaičiavimas

Atlikus duomenų filtravimą naudojant Savitzky-Golay filtrą, norėta įsitikinti ar duomenys nėra iškraipyti ir ar statistškai yra priklausomi nuo sumodeliuotos idealios trajektorijos. Tam buvo paskaičiuota koreliacija tarp idealios trajektorijos ir trajektorijos, kurios koordinatės gautos po Savitzky-Golay filtro. Lentelėje 3.15 surašyti rezultatai gauti paskaičiavus koreliaciją kiekvienam trajektorijos fiksavimui trimis skirtingais atstumais.

**Lentelė 3.15** Koreliacijos koeficientai

Testo Nr.	1	2	3	4	5	6	7	8	9	10
Koreliacija 1.3m	0.9988	0.9981	0.9997	0.9995	0.9989	0.9996	0.9995	0.9998	0.9978	0.9974
Koreliacija 2.3m	0.9999	0.9995	0.999	0.9994	0.9995	0.9996	0.9987	0.9986	0.9997	0.9982
Koreliacija 3m	0.9996	0.9996	0.9998	0.9997	0.9995	0.9996	0.9996	0.9996	0.9996	0.9995

Galima pastebėti, kad visais atvejais koreliacija yra artima vienetui, o tai reiškia, kad trajektorija po Savitzky-Golay filtro yra neiškraipyta ir stipriai statistiškai priklauso nuo sumodeliuotos idealios trajektorijos.

### 3.4 Hausdorff'o atstumas

Siekiant palyginti kaip stipriai panašios yra sumodeliuota ideali ir po Savitzky-Golay filtro gauta trajektorijos buvo paskaičiuotas Hausdorff'o atstumas, dar kitaip vadinamas Hausdorff'o metrika arba Pompeiu - Hausdorff atstumu. Tiesioginis Hausdorff'o atstumas parodo kaip stipriai vienas nuo kito yra nutolę du duomenų rinkiniai ir yra skaičiuojamas taip:

$$h(A, B) = \max_{a \in A} \left\{ \min_{b \in B} \{d(a, b)\} \right\} \quad 3$$

Tiesioginis Hausdorff'o atstumas yra jautrus stipriau nuo visos aibės nutolusiems taškams. Vienas ar keli stipriau nutolę taškai gali stipriai pakeisti Hausdorff'o atstumo reikšmę, skaičiuojant šia formule. Siekiant panaikinti šią problemą, skaičiuojamas modifikuotas Hausdorff'o atstumas (MHD), kuris yra apskaičiuojamas taip (Shao, Cai and GU, 2010, p.167) [21]:

$$h(A, B) = \frac{1}{m_a} \sum_{a \in A} \left( \min_{b \in B} d(a - b) \right) \quad 4$$

Gauti rezultatai pateikiami 3.16 lentelėje.

**Lentelė 3.16** Modifikuotas Hausdorff'o atstumas

Testo Nr.	1	2	3	4	5	6	7	8	9	10
1.3m MHD, cm	1.47	1.60	3.05	2.52	1.75	3.09	2.80	3.28	1.79	4.86
2.3m MHD, cm	4.06	2.92	4.60	4.48	2.38	5.20	5.46	5.36	3.74	5.53
3m MHD, cm	4.46	4.79	3.42	4.06	4.54	5.21	4.33	4.72	4.48	4.75

Iš gautų rezultatų galima pastebėti, kad trajektorijos kreivės labiausiai sutampa ir mažiausias Hausdorff'o atstumas gaunamas matuojant 1.3m atstumu. Tačiau apibendrinant kitus

du atvejus, trajektoriją fiksuojant 2.3m ir 3m atstumais, galima teigti, kad ideali ir užfiksuotos trajektorijos kreivės taip pat yra panašios, kadangi gauti atstumai tarp jų yra sąlyginai nedideli.

### 3.5 Problemos

Atliekant šį projektą išaiškėjo daugybė trikdžių ir kliūčių, trukdančių įgyvendinti pilnai ir tinkamai funkcionuojančią sistemą. Sudarytas sąrašas problemų, kurios kyla dėl pakankamai sudėtingos industrinio roboto valdymo struktūros ir ABB PC SDK bibliotekos, skirtos industrinių robotų programavimui:

1. Naudojant ABB PC SDK biblioteką ir robotstudio programinę įrangą, koordinačių masyvai turi būti iš anksto aprašyti „Robotstudio“ programoje ir negali būti tiesiogiai sukurti iš mūsų sukurtos aplikacijos. Tai yra didelis minusas sistemos dinamiškumui;
2. Industriniai robotai (robotinė ranka) dažniausiai turi 6 ašis. Dėl šios priežasties jo valdymas Kinect kamera tampa labai kompleksiškas arba neišpildantis viso funkcionalumo;

Pirmosios problemos sprendimui vėl gi reikėtų atlikti tyrimą su alternatyviais įrankiais. Artimiausia alternatyva ABB PC SDK bibliotekai yra kita ABB biblioteka „Robotstudio SDK“, kuri naudojama kaip programinės įrangos Robotstudio add-in paketų kūrimo plėtinys. Taip pat būtų galima iširti ROS programinę įrangą [14] Linux platformoje, kuri suteikia lanksčias industrinių robotų programavimo galimybes. ROS suteikia standartinę robotų OS (operacinę sistemą), kuri pasižymi aparatūros abstrakcija, žemo lygio įrenginio valdymu, bendrų OS funkcijų įgyvendinimu, pranešimų perdavimu tarp procesų ir paketų valdymu (Abdul Muis and Wisnu Indrajit, 2012, p. 601) [19].



## 4. IŠVADOS

1. Kompleksiška 6 ašių industrinio roboto struktūra neleidžia paprastu ir nuovokiu būdu valdyti roboto pasinaudojant vaizdo atpažinimo priemones. Dėl šios priežasties, naudojant vaizdo atpažinimo priemones, galima pasiekti tik dalį industrinio roboto valdymo funkcionalumo;
2. Lankstesnes industrinio roboto valdymo galimybes leistų įgyvendinti ROS programinė įranga, kadangi šiuo atveju naudojama oficiali ABB C# biblioteka PC SDK negali veikti pavieniui be Robotstudio RAPID programavimo kalbos;
3. Kinect sensoriaus tikslumas esant tiesioginiams saulės spinduliams yra nepriimtinas. Duomenys yra iškraipomi arba visiškai neužfiksuojami, priklausomai nuo saulės šviesos intensyvumo. Tokio paties rezultato reikėtų tikėtis, jeigu Kinect jutiklio regos zonoje būtų įjungta infraraudonųjų spindulių lempa;
4. Nesant tiesioginiams saulės spinduliams Kinect sensoriaus tikslumo rezultatai tenkina. Neapdorojus ir nepritaikius jokio glotninimo filtro duomenų naudoti nepatartina, kadangi jie yra „triukšmingi“;
5. Ideali trajektorijos koordinatės labai stipriai koreliuoja su koordinatėmis, apdirbtomis naudojant Savitzky-Golay filtrą. Tai parodo, kad jos nėra iškreiptos erdvėje;
6. Sumodeliuota ideali trajektorija ir trajektorijos, užfiksuotos Kinect jutikliu ir apdirbtos Savitzky-Golay lyginimo filtru skiriasi nežymiai. Tačiau tas skirtumas yra didesnis kai matavimai atliekami esant didesniam atstumui nuo Kinect jutiklio. Tai įrodo atliktas Hausdorff'o atstumo skaičiavimas.

## 5. LITERATŪROS SARAŠAS

1. Dominic Gorecky , Mathias Schmitt, Matthias Loskyll , Detlef Zühlke, „Human-Machine-Interaction in the Industry 4.0 Era“, 2014, adresas: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6945523>, [Internetinis, žiūrėta 2017-11-26]
2. Kinect for Windows Sensor Components and Specifications, adresas: <https://msdn.microsoft.com/en-us/library/jj131033.aspx> [Internetinis, žiūrėta 2017-11-07]
3. Kinect Sensor, adresas: <https://msdn.microsoft.com/en-us/library/hh438998.aspx> [Internetinis, žiūrėta 2017-11-07]
4. Kinect for Windows SDK 1.8, adresas: <https://msdn.microsoft.com/en-us/library/hh855354.aspx>, [Internetinis, žiūrėta 2017-11-07]
5. Natural User Interface for Kinect for Windows, adresas: <https://msdn.microsoft.com/en-us/library/hh855352.aspx>, [Internetinis, žiūrėta 2017-11-13]
6. ABB Robotstudio, adresas: <http://new.abb.com/products/robotics/robotstudio>, [Internetinis, žiūrėta 2017-11-13]
7. OpenNI, the standard framework for 3d sensing, adresas: <http://www.openni.ru/about/index.html>, [Internetinis, žiūrėta 2017-11-15]
8. OpenKinect, adresas: [https://openkinect.org/wiki/Main\\_Page](https://openkinect.org/wiki/Main_Page), [Internetinis, žiūrėta 2017-11-15]
9. Robotstudio Software Development Kit, adresas: <http://developercenter.robotstudio.com/pcsdk>, [Internetinis, žiūrėta 2017-11-15]
10. Kuka.SafeOperation, adresas: [https://www.kuka.com/en-my/products/robotics-systems/software/hub-technologies/kuka\\_safeoperation](https://www.kuka.com/en-my/products/robotics-systems/software/hub-technologies/kuka_safeoperation) , [Internetinis, žiūrėta 2017-12-02]
11. Yaskawa innovations, adresas: <https://www.yaskawa.eu.com/uk/solutions/innovations>, [Internetinis, žiūrėta 2017-12-02]

12. Technical reference manual RAPID Instructions, Functions and Data types, adresas:  
[https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technica1%20reference%20manual\\_RAPID\\_3HAC16581-1\\_revJ\\_en.pdf](https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technica1%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf), [Internetinis, žiūrėta 2017-10-10]
13. NumPy library, adresas:  
<https://docs.scipy.org/doc/numpy-1.14.2/reference/index.html>, [Internetinis, žiūrėta 2018-04-10]
14. G.S. Sandhu, C. McGinn and K. Kelly, „Investigation into a low cost robotic vision system“, adresas:  
[http://www.tara.tcd.ie/bitstream/handle/2262/74606/IMC32\\_-\\_Investigation\\_into\\_a\\_low\\_cost\\_robotic\\_vision\\_system.pdf;sequence=1](http://www.tara.tcd.ie/bitstream/handle/2262/74606/IMC32_-_Investigation_into_a_low_cost_robotic_vision_system.pdf;sequence=1), [Internetinis, žiūrėta 2018-04-12]
15. ANDRE GUI~ZIEC\* AND NICHOLAS AYACHE „Smoothing and Matching of 3-D Space Curves“, 1993, adresas:  
<https://link.springer.com/article/10.1007%2FBF01420985>, [Internetinis, žiūrėta 2018-04-12]
16. Jesus Suarez, Robin R. Murphy „Using the Kinect for Search and Rescue Robotics“ 2012, adresas:  
[https://www.researchgate.net/publication/236160253\\_Using\\_the\\_Kinect\\_for\\_Search\\_and\\_Rescue\\_Robotics](https://www.researchgate.net/publication/236160253_Using_the_Kinect_for_Search_and_Rescue_Robotics), [Internetinis, žiūrėta 2018-04-28]
17. José Luis Guiñón, Emma Ortega, José García-Antón, Valentín Pérez-Herranz „Moving Average and Savitzki-Golay Smoothing Filters Using Mathcad“, 2007, adresas:  
[https://www.researchgate.net/publication/228407245\\_Moving\\_average\\_and\\_Savitzki-Golay\\_smoothing\\_filters\\_using\\_Mathcad](https://www.researchgate.net/publication/228407245_Moving_average_and_Savitzki-Golay_smoothing_filters_using_Mathcad), [Internetinis, žiūrėta 2018-05-05]
18. Oliver Wasenmüller and Didier Stricker „Comparison of Kinect v1 and v2 Depth Images in Terms of Accuracy and Precision“, 2016, adresas:  
[https://www.researchgate.net/publication/308615218\\_Comparison\\_of\\_Kinect\\_V1\\_and\\_V2\\_Depth\\_Images\\_in\\_Terms\\_of\\_Accuracy\\_and\\_Precision](https://www.researchgate.net/publication/308615218_Comparison_of_Kinect_V1_and_V2_Depth_Images_in_Terms_of_Accuracy_and_Precision), [Internetinis, žiūrėta 2018-05-07]

19. Abdul Muis and Wisnu Indrajit, “Realistic Human Motion Preservation-Imitation Development on Robot with Kinect”, 2012, adresas:  
<http://journal.uad.ac.id/index.php/TELKOMNIKA/article/view/847>,  
[Internetinis, žiūrėta 2018-05-17]
20. Rositsa Botsova, Anna Lekova, Ivan Chavdarov “Imitation learning of robots by integrating Microsoft Kinect and PID Controller with a sensor for angular displacement in a robot joint”, 2015, adresas:  
[http://delivery.acm.org/10.1145/2820000/2812455/p268-botsova.pdf?ip=193.219.170.65&id=2812455&acc=ACTIVE%20SERVICE&key=1FA3353941FE8055%2E4A4E353D8E57EBBB%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&\\_\\_acm\\_\\_=1526921610\\_973625d6718a11f36d330e952e1a3c0c](http://delivery.acm.org/10.1145/2820000/2812455/p268-botsova.pdf?ip=193.219.170.65&id=2812455&acc=ACTIVE%20SERVICE&key=1FA3353941FE8055%2E4A4E353D8E57EBBB%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1526921610_973625d6718a11f36d330e952e1a3c0c), [Internetinis, žiūrėta 2018-05-17]
21. Fei Shao, Songmei Cai, Junzhong GU “A Modified Hausdorff Distance based Algorithm for 2-Dimensional Spatial Trajectory Matching”, 2010, adresas:  
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5593666>  
[Internetinis, žiūrėta 2018-05-20]