



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

Mantas Aukštakojis

PIEŠIANČIO ROBOTO VALDYMO SISTEMOS TYRIMAS

Baigiamasis magistro projektas

Vadovas

Prof. dr. Vytenis Sinkevičius

PANEVĖŽYS, 2017

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

PIEŠIANČIO ROBOTO VALDYMO SISTEMOS TYRIMAS

Baigiamasis magistro projektas
Valdymo technologijos (kodas 621H66001)

vadovas

(parašas) Prof. dr. Vytenis Sinevičius
(data)

Recenzentas

(parašas)
(data)

Projektą atliko

(parašas) Mantas Aukštakojis
(data)

PANEVĖŽYS, 2017



KAUNO TECHNOLOGIJOS UNIVERSITETAS
PANEVĖŽIO TECHNOLOGIJŲ IR VERSLO FAKULTETAS

(Fakultetas)

Mantas Aukštakojis

(Studento vardas, pavardė)

Valdymo technologijos, 621H66001

(Studijų programos pavadinimas, kodas)

„Baigiamojo projekto pavadinimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 17 m. Birželio 1 d.
Panevėžys

Patvirtinu, kad mano, **Manto Aukštakojo**, baigiamasis projektas tema „Piešiančio roboto valdymo sistemos 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)

20..... ..

BAIGIAMOJO PROJEKTO UŽDUOTIS

Išduota studentui: *Mantas Aukštakojis* Grupė *PME-5*

1. Darbo tema:

Lietuvių kalba: *Piešiančio roboto valdymo sistemos tyrimas*

Anglų kalba: *Research on Control System of Drawing Robot*

Patvirtinta 2017m. Kovo mėn. 30d. dekanų potvarkiu Nr. V25-13-8

2. Darbo tikslas:

Sukurti paveikslus piešiančio roboto valdymo sistemą ir ištirti jos valdymo galimybes

3. Reikalavimai ir sąlygos:

Tyrimo objektas- trijų laisvės laipsnių piešianti roboto ranka su analoginėmis RC servo pavaromis TowerPro MG945R, konstrukcija buvo duota..

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

1. *Literatūros apžvalga ir analizė*
 - *Robotų rankų konstrukcijų tipai*
 - *Paletavimo roboto rankos analizė*
 - *SCARA roboto rankos analizė*
2. *Projektinė dalis*
 - *Paveikslo vektorizavimas*
 - *Eksperimentinės roboto rankos konstrukcija*
 - *Eksperimentinės roboto rankos matematinis kampų skaičiavimas*
 - *Eksperimentinės roboto rankos valdymo sistema*
 - *Eksperimentinės roboto rankos valdymo algoritmų sudarymas*
 - *Eksperimentinės roboto rankos tikslumo tyrimas*
 - *Eksperimentinės roboto rankos judesio tyrimas*
 - *Eksperimentinės roboto rankos valdymo galimybių tyrimas*
3. *Išvados*
4. *Literatūros sąrašas*
5. *Priedai*
 - *Vektorizavimo algoritmas*
 - *Kampų skaičiavimo algoritmas*
 - *Duomenų perdavimo algoritmas*
 - *Valdymo bloko algoritmas*

5. Ši užduotis yra neatskiriama baigiamojo projekto dalis.

6. Projekto pateikimo gynimui kvalifikacinėje komisijoje terminas

Užduotį gavau:

(studento vardas, pavardė, parašas)

Vadovas:

(pareigos, vardas, pavardė, parašas)

(data)

2017-02-01

(data)

2017-02-01

(data)

Mantas Aukštakojis. Piešiančio roboto valdymo sistemos tyrimas. *Magistro* baigiamasis projektas / vadovas prof. Vytenis Sinkevičius; Kauno technologijos universitetas, Panevėžio technologijų ir verslo fakultetas.

Mokslo kryptis ir sritis: valdymo technologijos

Reikšminiai žodžiai: roboto ranka, valdymo sistema, pozicionavimas, koordinatės, atvirkštinė kinematika.

Panevėžys, 2017. 53 p.

SANTRAUKA

Roboto rankos dažniausiai naudojamos norint perteikti žmogaus rankų judesių galimybes. Piešiantys robotai puikiai atvaizduoja robotų rankų galimybes, be to tokiems robotams nekeliama ypatingi tikslumo reikalavimai. Jie skirti labiau meniniai išraiškai perteikti, todėl jų konstrukcijos būna paprastesnės nei, pavyzdžiui, pramoninių robotų rankų. Piešiantys robotai naudoja vektorizuotus paveikslus. Paveikslų vektorinės koordinatės, apskaičiuojant atvirkštinės kinematikos uždavinį, paverčiamos roboto rankos jungčių posūkio kampais.

Darbo tikslas: sudaryti paveikslus piešiančio roboto valdymo sistemą ir ištirti jos valdymo galimybes.

Pagrindinės šio darbo problemos: piešiančio roboto netikslumas, servo pavarų sinchronizavimas.

Darbo metodai: mokslines literatūros ir internetinių išteklių analizė, eksperimentai, darbas programomis *AUTOCAD*, *MATLAB* ir t.t.

Eksperimentai atlikti su trijų laisvės laipsnių piešiančiąja roboto ranka. Panaudotos trys tarpusavyje susietos analoginės RC tipo servo pavaros. Jos turi ribotą diskretiškumą, o jų greitis pasisukimo metu nėra valdomas.

Gautų rezultatų apibendrinimas: nustatyta, kad roboto rankos tikslumui didžiausią įtaką daro servo pavarų diskretiškumas ir dantračiuose atsirandantis laisvumas. Nustatyta, kad sklandžiam judesiui atlikti, didelę įtaką turi servo pavarų sinchronizavimas, kai jos pradeda ir sustoja suktis tuo pačiu metu.

Aukštakojis, Mantas. *Research on Control System of Drawing Robot: Master's thesis in / supervisor assoc. prof. Vytenis Sinkevičius. Panevėžys Faculty of technologies and business, Kaunas University of Technology.*

Research area and field: control technologies

Key words: robot arm, control system, positioning, coordinates, inverse kinematic

Panevėžys, 2017. 53 p.

SUMMARY

Robotic hands are usually used for an imitation of possibilities of human hand movement. Robots that draws or sketch contours splendidly demonstrate robotic hands capabilities while rigorous precision or accuracy requirements are not applied for such robots. They are designed for mostly artistic purpose, therefore the complexity of their construction is not as complicated as of industrial robotic hands. Drawing Robots use vector-images that vectoral coordinates are being converted to the rotation angles of robotic hand joints by solving an inverse kinematic query.

The goal of this Thesis is to create a control system of Drawing Robot and to analyze its performance and capabilities.

The main problems: inaccuracy of robotic hands; synchronization of servo gears.

Applied methods: analysis of scientific and online stock materials, experiments, use of drawing and mathematical modelling programs, such as AUTOCAD or MATLAB.

All the experiments are done with 3-axis robotic hand. Three interconnected RC type servo gears has been used. They have limited discretion and their rotation speed cannot be controlled.

Results summary: found, that the main factors of precision of robotic hand depends on discretion of servo gears and freedom between sun gears. Identified, that smooth robotic hand movement depends on synchronization of servo gears, when they start and stop rotating at the same time.

TURINYS

ĮVADAS	8
1. LITERATŪROS APŽVALGA IR ANALIZĖ	9
1.1. Robotų rankų konstrukcijų tipai	9
1.2. Paletavimo roboto rankos analizė	10
1.3. SCARA roboto rankos analizė.....	12
2. PROJEKTINĖ DALIS	14
2.1. Paveikslo vektorizavimas	14
2.2. Eksperimentinės roboto rankos konstrukcija.....	17
2.3. Eksperimentinės roboto rankos matematinis kampų skaičiavimas.....	19
2.4. Eksperimentinės roboto rankos valdymo sistema.....	22
2.4.1. Eksperimentinės roboto rankos valdymo sistema.	22
2.5. Eksperimentinės roboto rankos valdymo algoritmų sudarymas	24
2.6. Eksperimentinė dalis.....	28
2.6.1. Eksperimentinės roboto rankos tikslumo tyrimas	28
2.6.2. Eksperimentinės roboto rankos judesio tyrimas.....	32
2.6.3. Eksperimentinės roboto rankos valdymo galimybių tyrimas	35
IŠVADOS	38
LITERATŪROS SĄRAŠAS	39
PRIEDAI	40
1. VALDYMO SISTEMOS ALGORTIMAI	41
1.1. Vektorizavimo algoritmas.....	41
1.2. Kampų skaičiavimo algoritmas	43
1.3. Duomenų perdavimo algoritmas.....	50
1.4. Valdymo bloko algoritmas.....	51

ĮVADAS

Roboto rankos dažniausiai naudojamos norint perteikti žmogaus rankų judesių galimybes. Piešiantys robotai puikiai atvaizduoja robotų rankų galimybes. Tokiems robotams nekeliama ypatingi tikslumo reikalavimai. Jie skirti labiau meniniai išraiškai perteikti. Todėl jų konstrukcijos būna paprastesnės nei tarkim pramoninių robotų rankų. Jų įrangai nekeliama didelį reikalavimai. Piešiantys robotai naudoja vektorizuotus paveikslus. Paveikslų vektorinės koordinatės, apskaičiavus atvirkštinės kinematikos uždavinį, paverčiamos roboto rankos jungčių posūkio kampais. Eksperimentiniame robote naudojami analoginiai R/C tipo servo pavaros. Kurios turi ribotą diskretiškumą, jų greitis, judant iš taško A į tašką B, nėra valdomas.

Tyrimo objektas – eksperimentinės roboto rankos valdymo sistema.

Darbo tikslas – sukurti paveikslus piešančio roboto valdymo sistemą ir ištirti jos valdymo galimybes.

Darbo uždaviniai

- Išanalizuoti analogiškų robotų rankų konstrukcijų tipus.
- Vektorizuoti paveikslą.
- Sukurti valdymo sistemą.
- Sudaryti atvirkštinės kinematikos uždavinio sprendinius.
- Ištirti eksperimentinės roboto rankos tikslumą.
- Ištirti eksperimentinės roboto rankos valdymo galimybes.

Tyrimo metodai – mokslines literatūros ir internetinių išteklių analizė, darbas programomis Matlab, Autocad, Arduino IDE ir t.t.

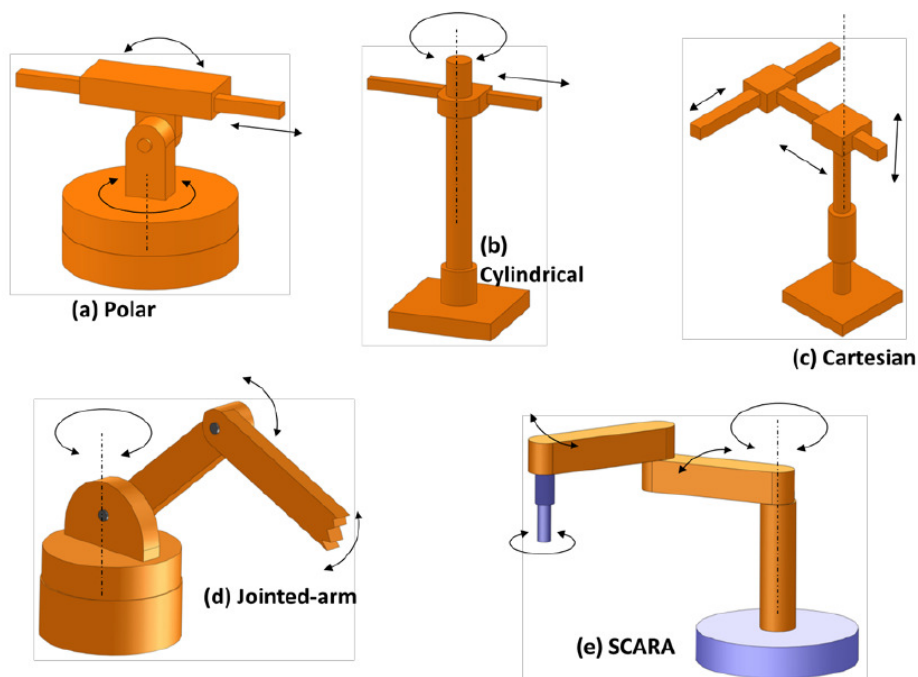
1. LITERATŪROS APŽVALGA IR ANALIZĖ

1.1. Robotų rankų konstrukcijų tipai

Šiame skyriuje bus analizuojami keli roboto rankų tipai, apžvelgiamos jų konstrukcijos, jų trūkumai ir privalumai.

Roboto rankos dažniausiai turi dvi dalis - korpusą ir ranką, prijungtą prie korpuso judamosiomis jungtimis. 1 paveiksle pavaizduoti dažniausiai naudojami robotų rankų konstrukcijų tipai:

- Polinė konstrukcija – ją sudaro slankiojantis L formos šarnyras, kuris gali sukotis aplink Z ašį 360° laipsnių, bei turi šarnyras horizontaliam judėjimui.
- Cilindrinė konstrukcija - ją sudaro vertikalus strypas, prie kurio pritvirtinta ranka gali judėti žemyn ir aukštyn, taip pat sukotis aplink savo ašį bei slankioti viena ar kita kryptimi.
- Dekarto koordinačių konstrukcija – ją sudaro 3 slankiojantys šarnyrai, judėjimą lemia tris koordinatės x-y-z.
- Šarnyrinė konstrukcija - sudėtis panaši į žmogaus rankos. Turi du šarnyrus peties, alkūnės rankos judėjimui bei gali sukotis aplink z ašį.
- SCARA (angl. *Selective Compliance Assembly Robot Arm*) konstrukcija - panaši į šarnyrinės konstrukcijos robotą, išskyrus peties ir alkūnės šarnyrus, kurie juda vertikaliai. Tai reiškia jog vertikaliai yra labai stabilus.[1]

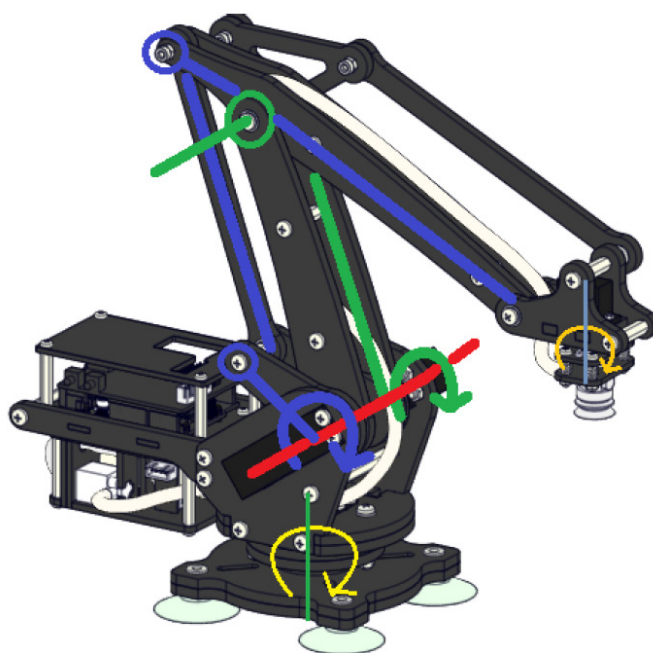


1 pav. Robotų rankų konstrukcijų tipai [1]: b) cilindrinė; c) dekartinė; d) šarnyrinė; e) SCARA

1.2. Paletavimo roboto rankos analizė

Paletavimo robotai daugiausia naudojami pramonėje. Todėl ir vadinami pakavimo (angl. *palletizing robot*) dėl paprastos konstrukcijos ir judesių amplitudės. Šis robotų tipas prigijo pramonėje kaip produktų dėliotai ant palečių. Nors tokių robotų pagrindinė panaudojimo sritis yra pramonė bet tokios konstrukcijos robotus galima panaudoti ir atliekant kitas funkcijas. Išanalizuosime, ar su tokiu robotu galima piešti.

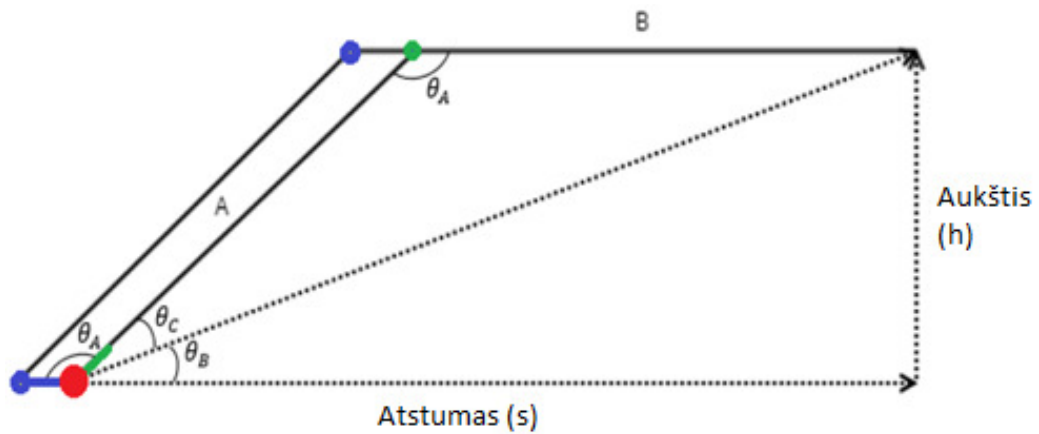
2 paveiksle pavaizduota viena iš tokio roboto konstrukcija. Jos judėjimą lemia keturios judėjimo ašys kurios leidžia šiai roboto rankai atlikti tam tikrus judesius. Vienas žingsninis variklis kontroliuoja pasisukimą aplink Z ašį 180° laipsniu. Dar du varikliai, kurie kontroliuoja rankos strėlės kryptį bei padėtį, sumontuoti iš šonų. Galima pamatyti 3 paveiksle. Dar vienas variklis, sumontuotas ant strėlės viršūnės, kontroliuoja objekto kryptį. Tokia roboto rankos konstrukcija leidžia su manipulatoriaus viršūnę pasiekti norimą tašką vieno variklio pagalba, o tai leidžia lengvesnį valdymo algoritmą. Skirtingai nei 6 ašių konstrukcija, kurios valdymą sunkina trajektorijų pasirinkimo variantai keliaujant link norimo taško. [2]



2 pav. Paletavimo roboto konstrukcija su pažymėtomis judesių ašimis [2]

3 paveiksle pavaizduota pakavimo roboto geometrinių ašių diagrama, iš kurios galima apskaičiuoti, kokių kampų reikia, kad būtų pasiekta norima roboto strėlės padėtis. Vienas žingsninis variklis sujungtas su rankos dalimi A, ir lemia jos judėjimą per sujungimo tašką (raudona spalva). Kitas žingsninis variklis lemia B roboto rankos B dalies judėjimą per jungimo taškus (mėlynos spalvos). Iš 3 paveikslo matoma, jog norint apskaičiuoti kampus θ_A , θ_B , θ_C reikės

žinoti aukštį h ir atstumą s iki taško, žinant atstumą galima apskaičiuoti kampus pasinaudojant kosinuso taisykle. [2]



3 pav. Pakavimo roboto geometrinė schema [2]

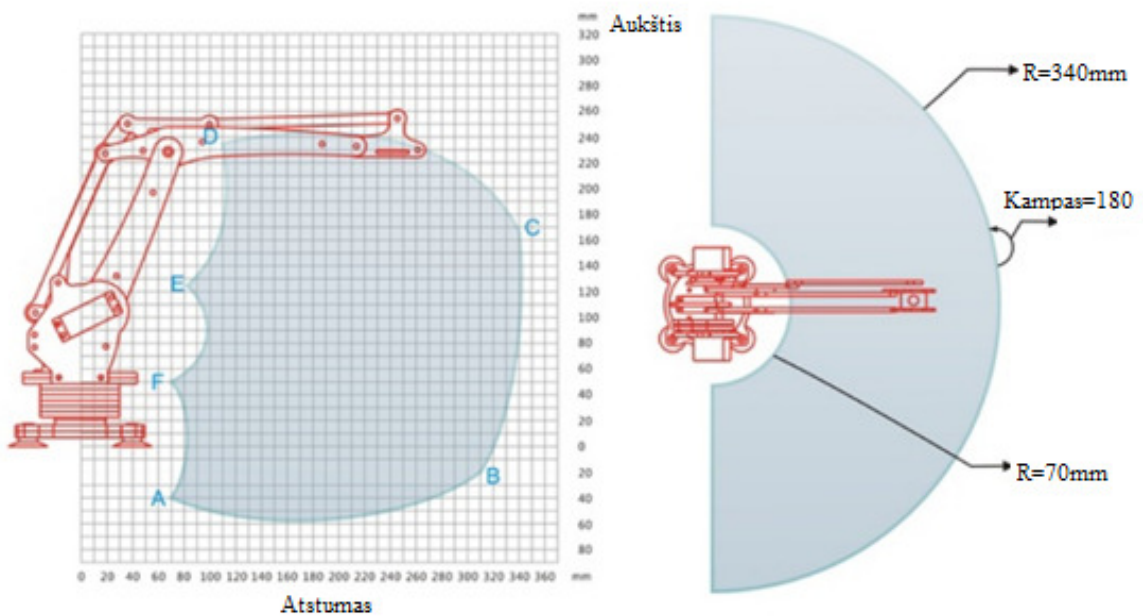
$$A^2 = B^2 + C^2 - 2BC \cos \theta_A, \quad (1)$$

Svirties A variklio žingsnio kampą galima rasti pagal formulę:

$$\theta_{\text{variklioA}} = \theta_B + \theta_C, \quad (2)$$

Svirties B variklio žingsnio kampą galima apskaičiuoti:

$$\theta_{\text{variklioB}} = 180 - \theta_A - \theta_B - \theta_C, \quad (3)$$



4 pav. Roboto rankos judesių amplitudžių diagramos [2]

Kaip matoma iš 4 paveikslo, rankos judesių amplitudė ar pasiekiamumas nėra labai platus, bet skirtingai nuo robotų su didesnėmis judesių amplitudėmis, ši roboto rankos konstrukcija išsiskiria žymiai lengvesniu valdymu. [2]

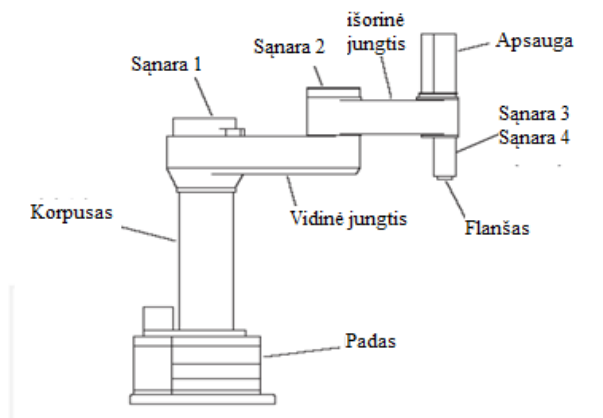
Kaip matoma iš formulių žinant atstumą iki taško ir priimant, jog aukštis bus visada vienodas, nesunku apskaičiuoti kampus. Šitokios rankos valdymo algoritmas nėra sudėtingas, nesunkiai ir neapkraunant valdymo bloko sudaromas algoritmas. Galima drąsiai teigti jog su tokia konstrukcija nesunkiai nubraižyti struktūras.

1.3. SCARA roboto rankos analizė

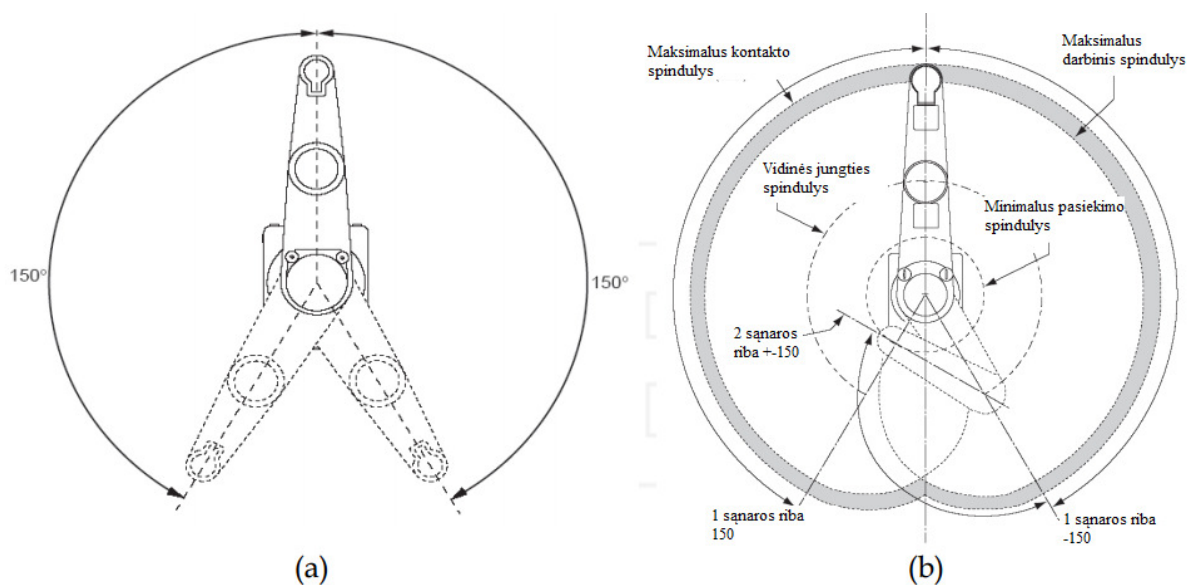
SCARA roboto konstrukcija sukurta pramonės šakoms, kurioms reikalingas surinkimo procesas ar žmogaus rankų darbo pakeitimas (5 pav.). Ši konstrukcija turi 4 sąnarų sistemą. Tokia roboto konstrukcija leidžia atlaikyti pakankamai dideles apkrovas, taip pat geros pasiekiamumo galimybes, bei judesių amplitudę.

Šio tipo roboto rankos turi 4 sąnarys. 3 sąnarys skirtas judesiui Z ašimi, 1, 2, 4 sąnarys sukamieji sąnarys. 6 paveikslo (b) dalyje parodyta sąnarys judesių amplitudė. Pirmojo sąnario, dar kitaip vadinamos petimi, veikimo principas panašus į žmogaus peties. Šio sąnario maksimalus pasisukimo laipsnis 300° , judesio amplitudė gali būti padalyti į dvi sritis - kairiąją ir dešiniąją po 150° laipsnių (6 pav., a). Antrasis sąnarys vadinamas alkūne, veikimo principas panašus į žmogaus alkūnę. Taip pat kaip ir pirmasis sąnarys, antrasis sąnarys gali sukis 300° kampu. 7 paveiksle pavaizduotos antrojo sąnario judesių galimybes. Norint išvengti dviprasmybės programuojant roboto ranką roboto ranka gali būti programuojama kaip kairės arba dešinės pusės.

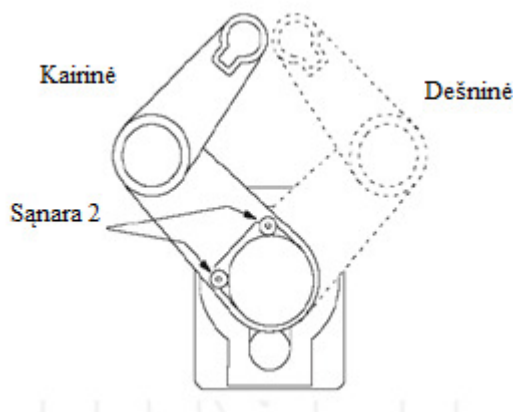
Šitokia roboto konstrukcija turi plačiausią darbinę zoną, bet turi ir trūkumų: norint išvengti žalos pačiam robotui, kiekvienas sąnarys valdomas su „lengvu - stabdymu“ ir „sunkiu - stabdymu“. Šie du valdymo veiksmai atsiranda tada, kai kuris nors sąnarys gali kirsti maksimalų leidžiamą kampą. [3]



5 pav. SCARA rankos konstrukcija [3]



6 pav. SCARA roboto rankos a) šarnyrų judesių amplitudės; b) darbo režimai [3]



7 pav. Antrojo šarnyro judesių galimybės. [3]

Apibendrinus analizės rezultatus galima teigti, kad roboto rankos gali braižyti paveikslus. Užtenka minimaliam braižymui dviejų laisvės laipsnių. Optimaliausias variantas tris laisvės laipsniai. Daugiau laisvės laipsnių apsunkina algoritmo kūrimą. Dėl to valdymo blokas turi spręsti sudėtingas užduotis.

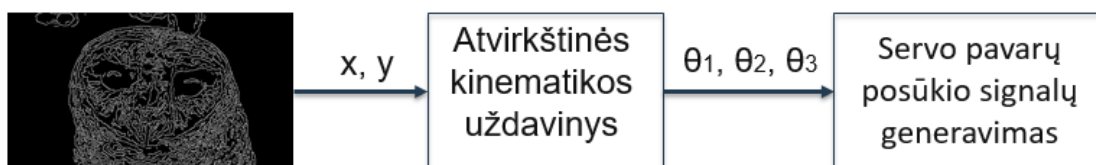
2. PROJEKVINĖ DALIS

Eksperimentą galima suskirstyti į tris etapus (8 pav.). Pirmas etapas paveikslo vektorizavimas. Antras etapas atvirkštinės kinematikos uždavinio sprendimas. Trečias servo signalų generavimas pagal gautas kampų vertes.

Pirmo etapo uždavinys yra gauti taško koordinatės x ir y iš paveikslo. Kadangi šio darbo tikslas yra braižyti paveikslo kontūrus. Bus reikalingas nuoseklus taškų generavimas ir perdavimas atvirkštinės kinematikos skaičiavimui.

Antro etapo uždavinys yra gautus duomenis iš pirmojo etapo paversti kampų vertėmis. Išspręsti atvirkštinės kinematikos uždavinį. Šio etapo kintamieji yra servo pavarų kampai. Šis etapas turi gauti trijų kampų vertes ir persiųsti trečiajam etapui.

Trečiojo etapo uždavinys yra iš gautų kampų verčių sugeneruoti pavaros posūkio signalą.



8 pav. Eksperimento struktūrinė schema

2.1. Paveikslo vektorizavimas

Prieš pradėdant vykdyti šį etapą reikia iškelti tikslą: koks bus reikalingas atvaizdavimas. Kadangi tiriama roboto ranka nėra tiksli ir negali braižyti taip kaip spausdintuvai. Todėl pasirenkama meninis atvaizdavimo būdas. Kai nėra kreipiamas dėmesys į tikslumą. Šiuo atveju didžiausias dėmesys bus kreipiamas į paveikslo atvaizdavimą, kuo panašesnis į originalą ar bent jau atpažįstamas.

Vektorizuojant paveikslą reikalingos tiesių koordinatės. Todėl buvo iškelti keletas užduočių: kaip atpažinti paveikslo struktūras, kaip atpažintas struktūras suskirsti į tieses, kaip tiesės koordinatės eilės tvarka perduoti kampų skaičiavimo algoritmui.

Vektorizavimo algoritmo sudarymui buvo pasirinktas konkretus paveikslas (9 paveikslas). Norint pritaikyti šį konkretų paveikslą tyrimui, reikalingas jo paveikslo sumažinimas iki eksperimentinės roboto rankos braižymo ribas. Eksperimentinės roboto rankos ribos yra: x -610mm, y -310mm. Pakeitus paveikslo matmenys, pradėdamas paveikslų kontūrų paieškos algoritmas. Šiam žingsniui atlikti buvo pasirinktos kelios Matlab funkcijos: *rgb2gray* ir *edge* (*,canny'*). Primoji funkcija paveikslą konvertuoja į pilkų atspalvių paveikslą (10 pav.). Antroji funkcija iš konvertuoto paveikslo suranda spalvų struktūras. Jas apveda linijomis (11 pav.).

Kontūrai yra sužymimi vektorių matricioje. Balti taškai matricioje pažymimi „1“. Juodi taškai „0“. Matricos dydis yra toks kaip paveikslo.



9 pav. Pirminis vektorizuojamo paveikslo vaizdas



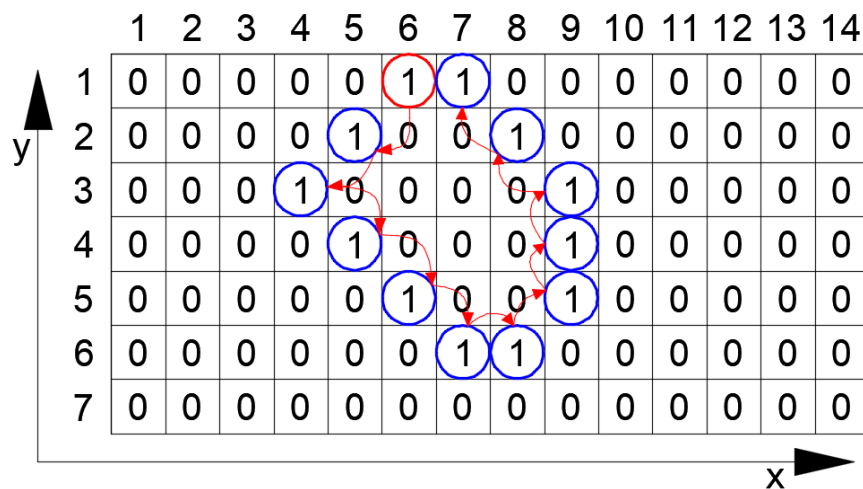
10 pav. Pakeistų matmenų ir konvertuoto į pilką vaizdas



11 pav. Vekturizuoto paveikslo vaizdas

Turint vektorių matricą kyla klausimas, kaip sužymėtus taškus galima sujungti į tieses ar kontūrus. Taip pat reikalingas kontūrų ir tiesių suradimas, tam, kad eksperimentinė roboto ranka galėtų atliktų poslinkį. Tai pat reikia žinoti tiesės pradžia ir pabaiga. Kad būtų žinoma kada reikia pakelti ir nuleisti braižymo priemonę. Tiesių suradimui buvo pasirinkta naudoti funkcija *bwtraceboundary*, šita funkcija buvo pasirinkta todėl, kad ji seka tiesės taškus pagal laikrodžio rodyklę arba prieš laikrodžio rodyklę, ir surašo duomenys, x ir y koordinatas į naują vienos tiesės matricą (12 pav.). Dviejų stulpelių matricą. Kur pirmas stulpelis y reikšmės, antras stulpelis x

reikšmės. 12 paveiksle pavaizduotas funkcijos *bwtraceboundary* veikimo principas. Raudonas apskritimas žymi pradžios tašką, nuo jo prasideda sekimas prieš laikrodžio rodyklę. Pradinis taškas randamas tikrinant kiekvieną vektorių matricos vertę: pradedant nuo pirmos eilutės, pirmo stulpelio. Kai patikrina visą pirmą eilutę, pradeda tikrinti antrąją eilutę, taip tesia tol kol patriną visą matricą. Suradus pirmąjį „1“ algoritmas pradeda tikrinti ar aplink rastą „1“ yra daugiau „1“. Jai randamas vienetas yra traktuojama, kad rasta tiesės pradžia. Algoritmas nutraukia vektorių ieškojimą ir pradeda vykdyti *bwtraceboundary* funkciją. Rastos tiesės taškų koordinatės x ir y surašomos į naują matricą – tik šios tiesės matricą. Taigi turim vienos tiesės duomenys, bet to neužtenka, reikia rasti visas paveikslo tieses, kontūrus. Todėl reikia rastą tiesę ištrinti iš pirminės vektorių matricos, nes kitu atveju antra kartą bus rasta ta pati tiesė.



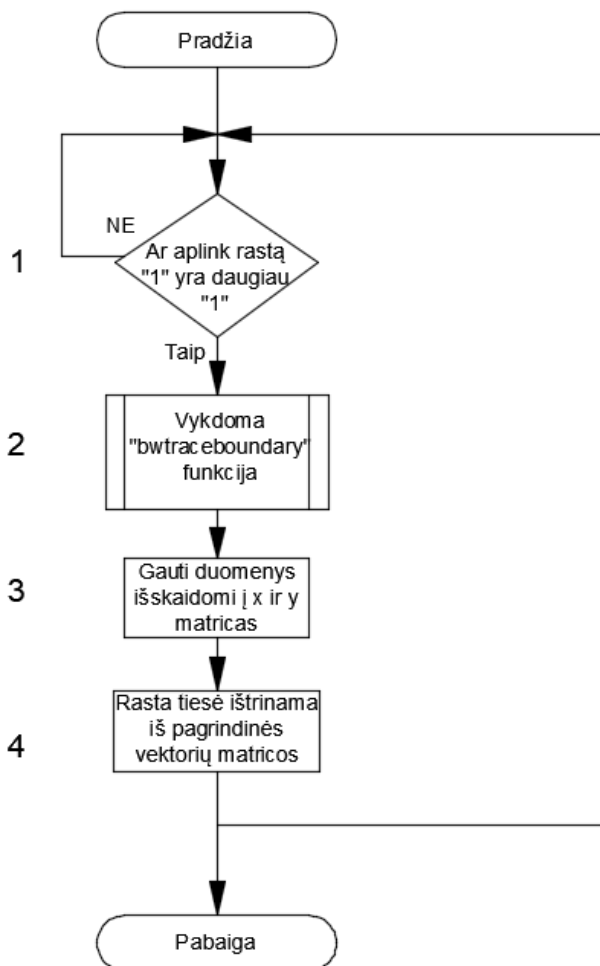
12 pav. *Bwtraceboundary* funkcijos veikimas

Kad būtų patogiau matyti ar algoritmas veikia, visos rastos tiesės paryškinamos žalia spalva (13 pav.). Rastų tiesių koordinatės surašomos atskirai į x ir y matricas, kur kiekvienas stulpelis reiškia naują tiesę. Vektorizavimo algoritmas atskirus taškus, kurie nepatenka į tieses ar kontūrus yra panaikinami.



13 pav. Paveikslo vektorizavimo gautas rezultatas

Matomas algoritmo vekturizuojačio paveikslus struktūrinė schema (14 pav.). Kaip matoma iš struktūros, algoritmas susideda iš keturių pagrindinių žingsnių. Kuria sudaro du dideli ciklai. Pirmas ciklas ieško „1“ vektorių matricoje. Antras ciklas leidžia: po rastos kreivės ištrynimo, pradėti ieškoti naujos kreivės. Algoritmo rezultatas yra dvi matricos: x ir y. Kiekvienoje matricoje stulpelis žymi nauja tiesę. O rezultatai surašomi į eilutes nuosekliai sekant tiesę nuo pradžių iki galo.



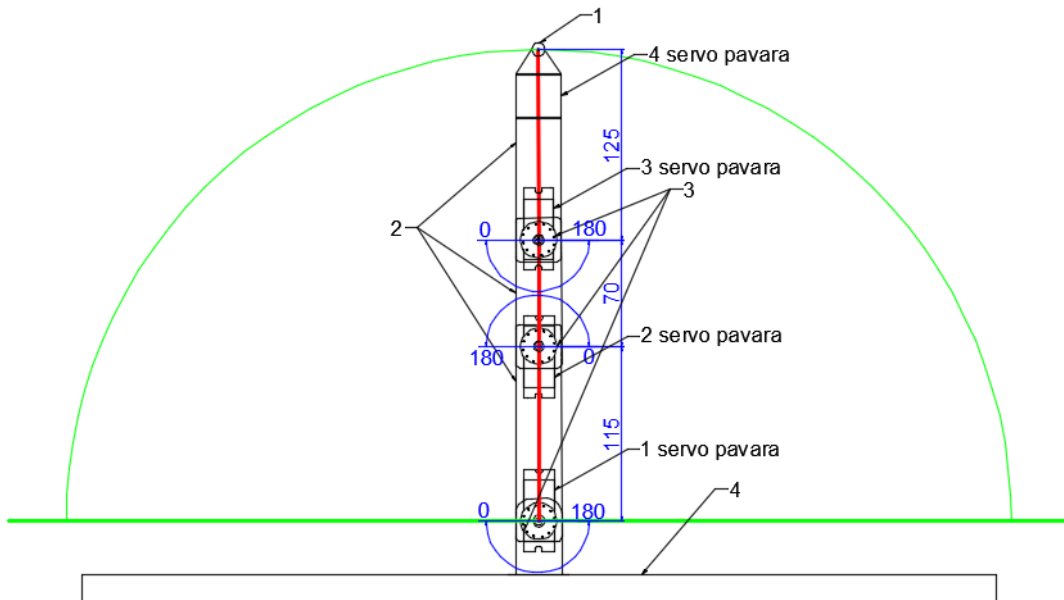
14 pav. Paveikslų vektorizavimo algoritmo struktūrinė schema

2.2. Eksperimentinės roboto rankos konstrukcija

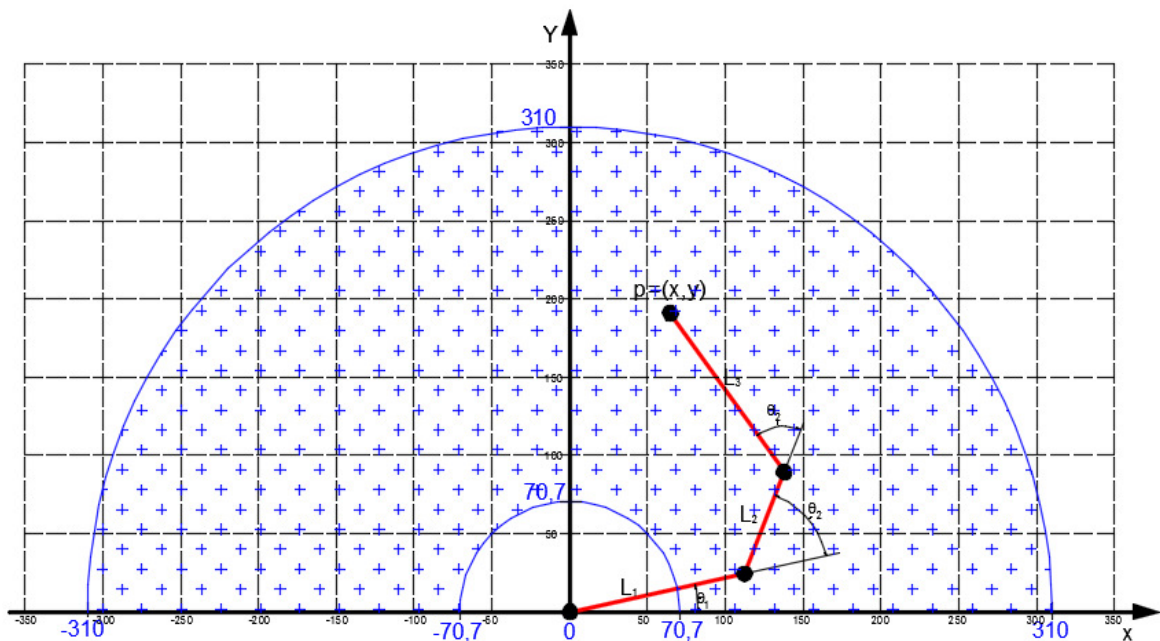
Gavus paveikslų vektorizavimo rezultatus galima jau pradėti eksperimento atlikimą. Tam reikia išanalizuoti eksperimentinės roboto rankos sudėtį ir konstrukcines galimybes. Šioje darbo dalyje bus analizuojama eksperimentinės roboto rankos konstrukcija: mechaninė dalis.

Eksperimentinės roboto rankos konstrukcija matoma (15 pav.), konstrukcija padaryta tokia, kad sugebėtų laisvai judėti horizontalia kryptimi į dešinę ar kairę puses. Roboto horizontalius judesius lemia 3 servo pavaros nuosekliai susietos. Judesių amplitudę apriboja rankos ilgis, bei 2 ir 3 servo pavaros maksimalus pasukimo kampai, kadangi jų pasukimo kampas θ_2 yra vienodas. Roboto rankos geometrinė diagrama pavaizduota (16 pav.). Matoma mėlyna

spalva užbrėžtus du lankus, tai yra roboto rankos pasiekiamumo ribos. Pirmą lanką apriboja roboto rankos ilgis, x ašyje nuo -310 iki 310 ir y ašyje nuo 0 iki 310. Mažesnę lanką apriboja 2 ir 3 pavarų pasukimo ribos. x ašyje nuo -70,7 iki 70,7 ir y ašyje nuo 0 iki 70,7. Šita riba atsiranda dėlto, kad 2 ir 3 pavaros sukasi vienodu kampu. Tokiu atveju roboto ranka gali susisukti 90° viena kitos atžvilgiu.



15 pav. Eksperimentinės roboto rankos konstrukcija, kur 1- galinis taškas, 2 – jungtys, 3- mini servo pavaros, 4- roboto rankos laikančioji konstrukcija



16 pav. Tiriamos roboto rankos geometrinė diagrama

2.3. Eksperimentinės roboto rankos matematinis kampų skaičiavimas

Šioje stadijoje sprendžiamas kampų skaičiavimas pagal turimas koordinates. Kampų skaičiavimai daromi pagal matematinės atvirkštinės kinematikos formules. Žemiau pateikta atvirkštinės kinematikos kampų skaičiavimo formulės.

Kadangi turimo roboto rankos veikia vienoje plokštumoje [x ir y], z ašis šiame darbe nebus vertinama.

Atvirkštinio uždavinio tikslas surasti funkcijas, kurios aprašytų θ posūkio kampų priklausomybes nuo galo taško koordinatų x ir y:

$$\left. \begin{aligned} \theta_1 &= f_1(x, y), \\ \theta_2 &= f_2(x, y), \\ \theta_3 &= f_3(x, y). \end{aligned} \right\} \quad (4)$$

Kad supaprastinti sprendinio analizę priimama, kad

$$\theta_{23} = \theta_2 = \theta_3. \quad (5)$$

Tai reiškia, kad antrosios ir trečiosios sąnaros pavaros visada pasisuka tuo pačiu kampu. Kadangi servo pavarų posūkio kampai tik $\pm 90^\circ$. Roboto konstrukcija padaryta tokia kad leistų pasukti 180° kampu atžvilgiu pirmos sąnaros. Dabar galima užrašyti lygčių sistema tokia:

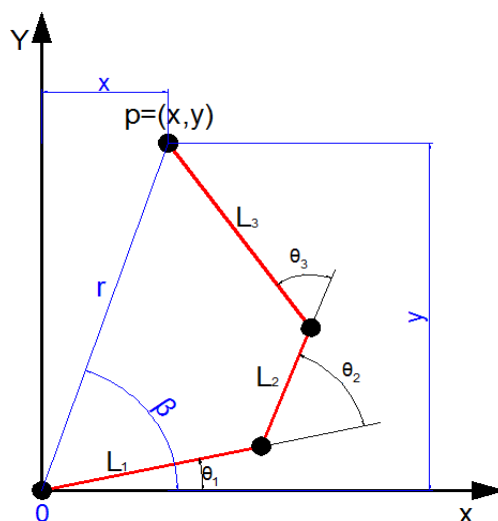
$$\left. \begin{aligned} x &= L_1 \cdot \cos \theta_1 + L_2 \cdot \cos(\theta_1 + \theta_{23}) + L_3 \cdot \cos(\theta_1 + 2 \cdot \theta_{23}) \\ y &= L_1 \cdot \sin \theta_1 + L_2 \cdot \sin(\theta_1 + \theta_{23}) + L_3 \cdot \sin(\theta_1 + 2 \cdot \theta_{23}) \end{aligned} \right\} \quad (6)$$

Lygčių sistema (1) supaprastėjo ir dabar reikia rasti tik dvi funkcijas:

$$\left. \begin{aligned} \theta_1 &= f_1(x, y), \\ \theta_{23} &= f_2(x, y). \end{aligned} \right\} \quad (7)$$

Lygčių sistemos sprendiniams gauti bus naudojama 17 paveikslo schema, kur vektoriaus r ir posvyrio kampas β randami iš žinomų koordinatų x ir y:

$$\left. \begin{aligned} r &= \sqrt{x^2 + y^2}, \\ \beta &= \arctg\left(\frac{y}{x}\right) \end{aligned} \right\} \quad (8)$$



17 pav. Atvirkštinės kinematikos uždavinio sprendimo schema

Pertvarkius (3) lygčių sistema galima užrašyti taip:

$$x^2 + y^2 = L_1^2 + L_2^2 + L_3^2 + 2 \cdot L_1 \cdot L_2 \cdot c(\theta_{23}) + 2 \cdot L_1 \cdot L_3 \cdot c(2 \cdot \theta_{23}) + 2 \cdot L_2 \cdot L_3 \cdot c(\theta_{23}). \quad (9)$$

Kadangi:

$$\cos(2 \cdot A) = 2 \cdot \cos^2(A) - 1, \quad (10)$$

Įrašius (7) lygtį į (6) lygtį ir pažymėjus, kad

$$B = \cos(\theta_{12}). \quad (11)$$

Gaunama tokia lygtis:

$$x^2 + y^2 = L_1^2 + L_2^2 + L_3^2 + 2 \cdot L_1 \cdot L_2 \cdot B + 2 \cdot L_1 \cdot L_3 \cdot 2 \cdot B^2 - 2 \cdot L_1 \cdot L_2 + 2 \cdot L_2 \cdot L_3 \cdot B. \quad (9)$$

Lygtį (9) sutvarkius ir įrašius į ją pažymėjimus gaunama:

$$\left. \begin{aligned} b_{01} &= -L_2 \cdot (L_1 + L_3), \\ b_{02} &= L_1^2 \cdot (L_1 + L_3)^2 = b_0^2, \\ b_{03} &= 4 \cdot L_1 \cdot L_3, \\ b_{04} &= (L_1 - L_3)^2 + L_2^2, \\ B_{1,2} &= \frac{b_{01} \pm \sqrt{b_{02} - b_{03} \cdot (b_{04} - x^2 - y^2)}}{b_{03}} \end{aligned} \right\}. \quad (12)$$

Iš apskaičiuotų šaknų B surandamas θ_{23} kampas:

$$\theta_{23} = \pm \arccos(B_1), \pm \arccos(B_2). \quad (13)$$

Liko paskaičiuoti kampą θ_1 . Tam pasitelkta bus schema (18 pav.). Kai žinomas kampas θ_{23} ir taško koordinatės x_1 ir y_1 , kampas apskaičiuojamas:

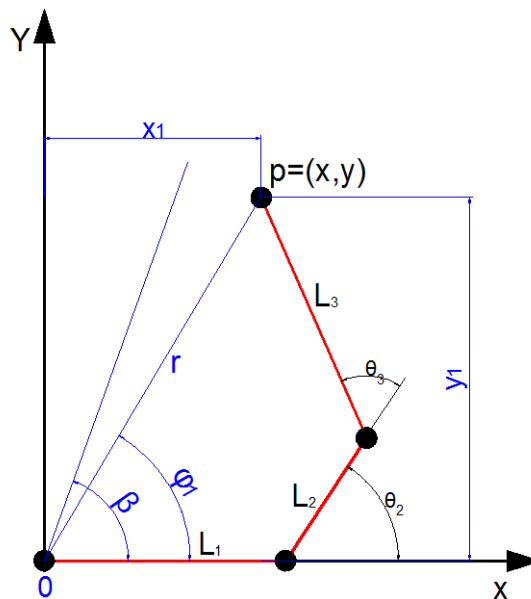
$$\left. \begin{aligned} x_1 &= L_1 + L_2 \cdot \cos(\theta_{23}) + L_3 \cdot \cos(2 \cdot \theta_{23}), \\ y_1 &= L_2 \cdot \sin(\theta_{23}) + L_3 \cdot \sin(2 \cdot \theta_{23}) \end{aligned} \right\} \quad (14)$$

Vektoriaus r kampas yra lygus:

$$\varphi_1 = \arctg\left(\frac{y_1}{x_1}\right). \quad (15)$$

Toliau kampas gaunamas taip:

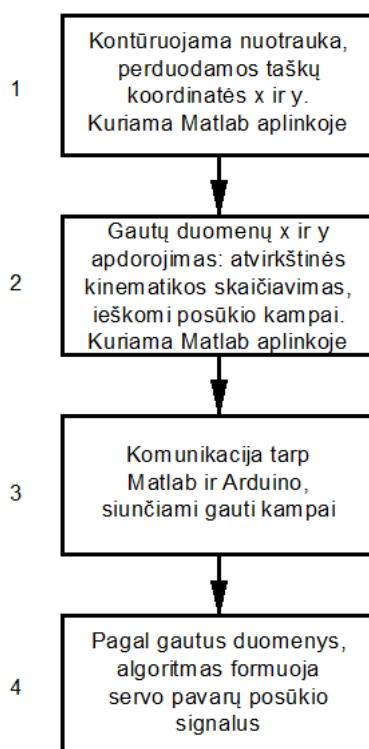
$$\left. \begin{aligned} \theta_1 &= \beta - \varphi_1, \Rightarrow \theta_1 < 0, \\ \theta_1 &= \pi + \beta - \varphi_1, \Rightarrow \theta_1 \geq 0. \end{aligned} \right\} \quad (16)$$



18 pav. Pakoreguotas atvirkštinės kinematikos uždavinio sprendimo schema

2.4. Eksperimentinės roboto rankos valdymo sistema.

Radus taškų koordinates bei išanalizavus eksperimentinės roboto rankos konstrukcija, galima pradėti kurti valdymo algoritmą. 19 paveiksle pavaizduota roboto rankos valdymo struktūrinė schema. Šita struktūrinė schema bus vadovaujama kuriant algoritmą. Kuri susideda iš 4 kūrimo etapų. Algoritmas yra vykdomas nuoseklia tvarka. Kiekvienas etapas yra vykdomas tol, kol pasiekiamas etapo rezultatas. Jai kažkuriame etape įvyksta nenumatyti įvykiai, visas algoritmas nustoja veikti arba gaunami neteisingi duomenys.



19 pav. Valdymo algoritmo struktūrinė schema

2.4.1. Eksperimentinės roboto rankos valdymo sistema.

20 paveiksle pavaizduota roboto rankos elektrinė jungimo schema kurią sudaro valdymo blokas, 4 servo pavaros, maitinimo šaltinis.

Valdymo blokas buvo parinktas *Arduino mega 2560*, kurio pagrindą sudaro ATmega2560 mikroprocesorius.

Mikroprocesoriaus parametrai:

- 54 skaitmeniniai įėjimai/išėjimai, iš jų 15 išėjimų PWM signalui gauti;
- 16 analoginių įėjimų;
- 4 UART kanalai;
- gali dirbti 16MHz taktiniu dažniu;

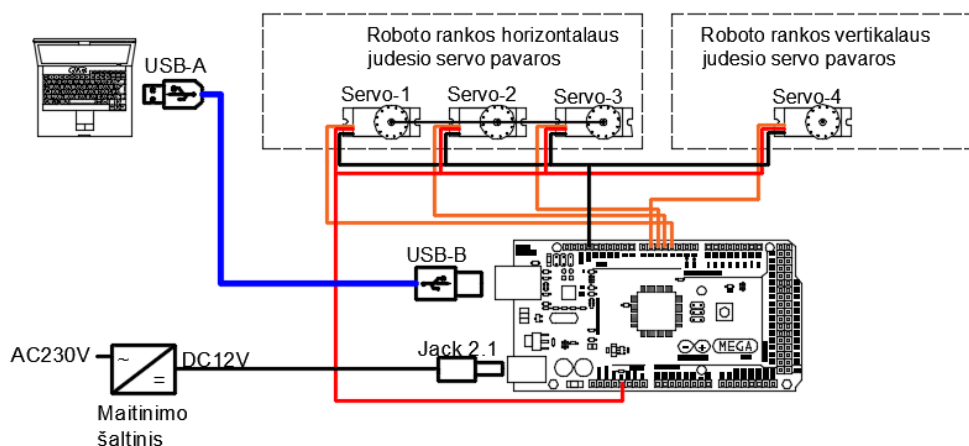
- turi 256 KB įrašymo atminties talpą;
- laikinosios atminties talpa SRAM 8 KB;
- maitinimas įtampa: nuolatinė nuo 7 iki 12V.

4 servo pavaros sudaro galimybę atlikti judesį pagal apskaičiuotus rezultatus. Kaip apskaičiuojamas judesys bus pateikta kitame skyriuje. Dėl pailgėjusių laidų atsiranda triukšmas, kuris gali turėti įtakos klaidingai veikiančiai įrangai. Sprendimas šalinant energijos triukšmą buvo prijungiami kondensatoriai lygiagrečiai maitinimo šaltiniui, kurie sumažino įtampos triukšmų lygį. Buvo panaudoti du tipai kondensatorių. Vienas elektrolitinis ir vienas keraminis kondensatoriai, vienai servo pavarai, tvirtinant kuo arčiau servo pavaros.

Servo pavaros parametrai:

- matmenys 40,7x19,7x42,9mm;
- svoris 55g;
- greitis 0,23s/60 laipsnių prie 4,8V ir 0,2s/60 laipsnių prie 6V;
- veikimo įtampa 4,8-6,6V;
- metalinis reduktorius;
- maitinimo šaltinis 12VDC, 800mA.

Komunikacijai su personaliniu kompiuteriu naudojamas USB-A/USB-B laidas. Komunikacija reikalinga tam, kad būtų galima perduoti reikalingus parametrus į valdymo bloką. Duomenims apdoroti naudojama Matlab programinė įranga. Arduino mega 2560 mikroprocesoriui programuoti naudojama Arduino IDE programinė įranga.



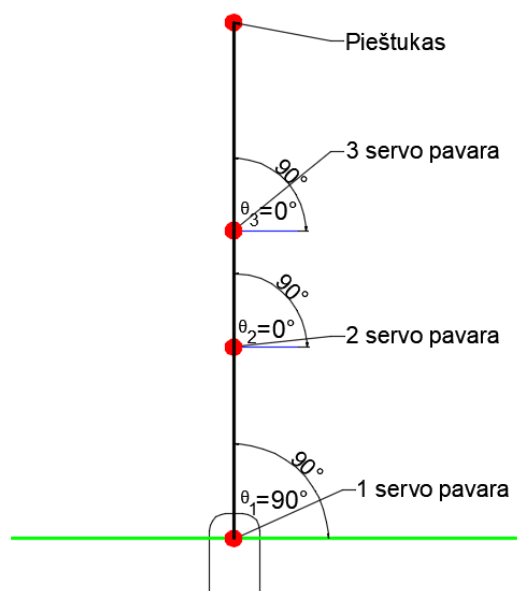
20 pav. Eksperimentinės roboto rankos elektrinė jungimo schema

2.5. Eksperimentinės roboto rankos valdymo algoritmų sudarymas

Kadangi gautų kampų vertės yra apskaičiuotos radianais, reikia šias vertes paversti į laipsnines vertes, kokias galima būtų siusti į Arduino servo pavarų valdymui. Kampų reikšmes paverčiamos taip:

$$\Theta = 180 \cdot \theta_{1,2,3} / \pi . \quad (17)$$

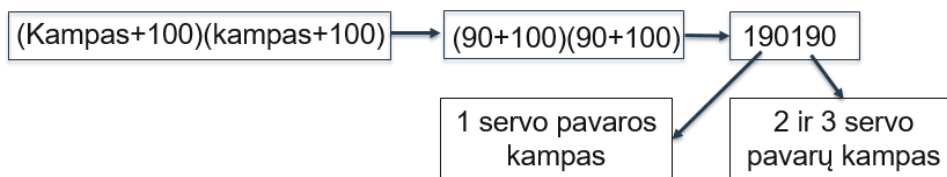
Kad sulyginti realiai veikiančią roboto ranką ir atvirkštinės kinematikos apskaičiuotus kampus. Reikia jų reikšmes paversti į servo pavarų naudojamus kampų ribas. Iš 21 paveikslo matoma, jog apskaičiuoti kampai 2 ir 3 servo pavaros skiriasi. Dėl to kad kampai skaičiuojasi nuo prieš tai esančios sąnaros. Išskyrus 1 sąnarą, nes sutampa servo pavaros ir skaičiuojamo kampo ašys. Norint suvienodinti servo pavarų kampus ir apskaičiuojamus kampus, reikia prie apskaičiuotų kampų 2 ir 3 servo pavaroms pridėti po 90° kampą. [14]



21 pav. Roboto rankos kampų skirtumas

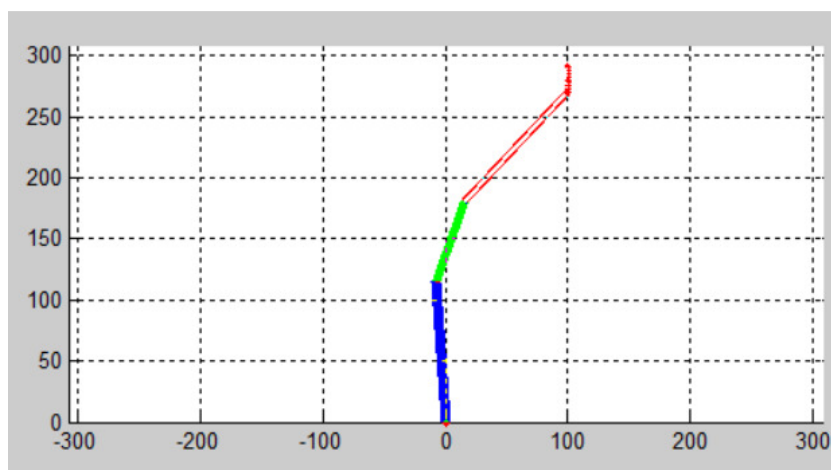
Apskaičiavus ir pavertus apskaičiuotas reikšmes tinkamomis perduoti į servo pavaras. Reikalinga komunikacija tarp personalinio kompiuterio ir valdymo bloko. Kadangi matematiniai veiksmai atliekami Matlab aplinkoje, o servo pavarų valdymas Arduino pagalba. Buvo priimtas sprendimas, kad valdymo blokas atliks tik servo pavarų signalų generavimo funkcija, o pagrindines skaičiavimo, vektorizavimo funkcijas šiame tyrime atliks Matlab programa. Todėl priimtas sprendimas, kad abu apskaičiuoti kampai bus siunčiami vienu metu. Atsirado poreikis tuos kampus sujungti į vieną duomenų paketą (22 pav.). Buvo nuspręsta, jog reikia visada siųsti 6 skaičių paketą. Todėl prie kiekvieno kampo, prieš sujungiant kampų vertes į vieną skaičių, ir juos išsiunčiant, yra prideda po 100. Taip gaunama jog siunčiamas duomenų paketas susideda iš 6

skaičių. Toks sprendimas palengvino Arduino valdymo algoritmo kūrimą. Nes valdymo blokui belieka atlikti kelis paprastus matematinius veiksmus.



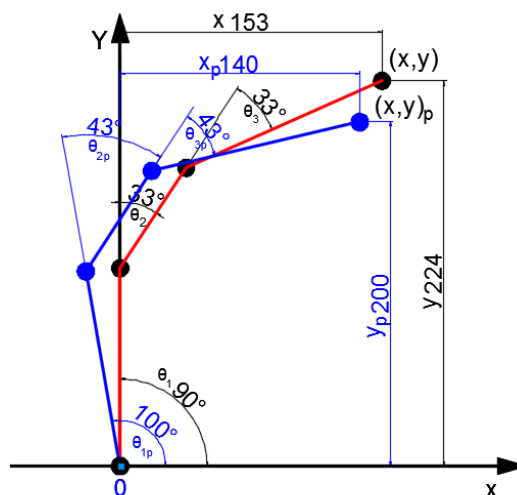
22 pav. Perduodamų duomenų paketo struktūra

Kad lengviau suprasti, kaip turi judėti ranka judesio metu, buvo parašytas algoritmas, kuris braižytų roboto rankos judesį Matlab aplinkoje (22 pav.). Tam buvo pasitelkta tiesioginės kinematikos uždavinio sprendimas. Kadangi jau yra apskaičiuoti sąnarų kampai. Apskaičiuoti tiesioginės kinematikos uždavinį yra daug lengviau nei atvirkštinės kinematikos.



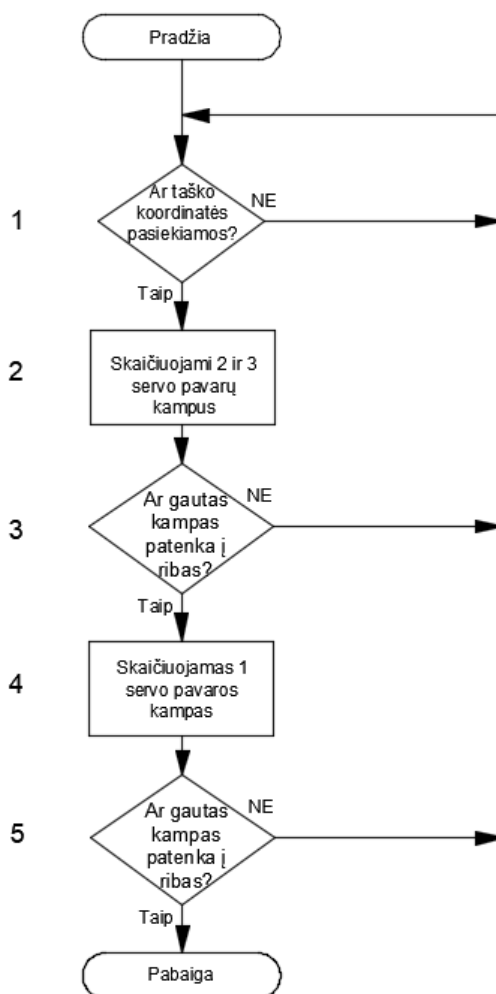
23 pav. Roboto rankos judesio atvaizdavimas Matlab aplinkoje

23 paveiksle pavaizduota, kaip apskaičiuojamas roboto rankos poslinkis. Poslinkis skaičiuojamas koordinatų pokyčiu. Sąnarų kampus skaičiuojant kiekviename taške. Kaip matome 23 paveiksle parodyta dviejų taškų roboto rankos sąnarų kampų verčių skirtumai. Tokiu būdu siunčiant valdymo blokui kampų vertes viena po kitos, galima gauti norimus judesius. Kuo daugiau taškų piešiamos kreivės turima, tuo daugiau apskaičiuojama sąnarų posūkio kampų. Tokiu būdu gali atlikti tolygesnį judesį, ne tokį kampuota. Todėl buvo išspręstas klausimas, ko reikia kad eksperimentinės roboto ranka atliktų judesį iš žinomo taško A į žinomą tašką B. Judesį suskaidant žingsniais. Savaime suprantama, kad kuo daugiau taškų bus žinoma kelyje tarp A ir B, tuo galima bus nubraižyti tikslesnį kelią.



24 pav. Roboto rankos poslinkio skaičiavimo pavyzdys

24 paveiksle matomas algoritmo skaičiuojančio servo pavarų posūkio kampus struktūrinė schema. Algoritme naudojamos prieš tai išvestos formulės. Algoritmo antroje dalyje naudojama 13 formulė. Ketvirtoje dalyje naudojama 16 formulė. Pilnas Matlab kodas pateiktas 3 priede Šio algoritmo gaunamas rezultatas yra dviejų servo pavarų kampai.



25 pav. Eksperimentinės roboto rankos algoritmo struktūrinė schema kampams skaičiuoti

Turint apskaičiuotas kampų vertes ir buvo išspręstas duomenų perdavimo klausimas. Buvo sukurtas valdymo bloko algoritmas. Kuris generuoja posūkio kampų signalus (26 pav.). Pradžioje buvo galvojama, kad servo kampų vertes bus saugomos valdymo bloke, kampų masyvuose. Bet eksperimento metu pasirodė, kad toks algoritmas nėra tinkamas naudoti. Todėl, kad:

- per didelis kiekis duomenų, trukdo sukaupti kampus į masyvus. Dėl valdymo bloko nepakankamos atminties talpos;
- algoritmo lėtumas. Daug laiko užima, tol kol perduodamas visi duomenys. Atsiranda didelė tikimybė klaidoms;
- komplikotas servo pavarų valdymas.

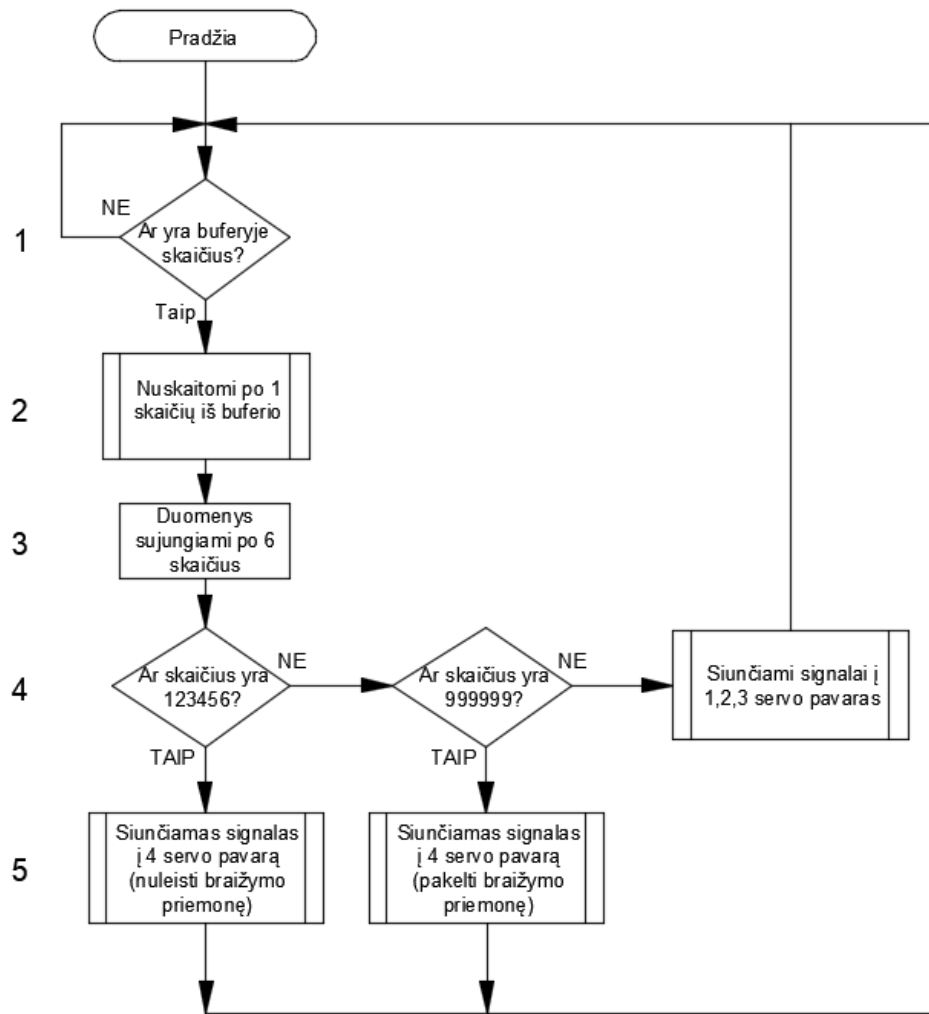
Atmetus tokio algoritmo struktūrą. Buvo priimtas sprendimas kampų masyvus sukaupti ne valdymo bloke o Matlab aplinkoje. Ir kai baigiamas kampų skaičiavimas, visi kampai sukaupti masyvuose, žinomos kreivės pradžia ir pabaiga, tik po to pradedamas kampų siuntimas į valdymo bloką, nuosekliai nuo kreivės pradžios iki pabaigos, prieš pradedant braižyti kreivę, pirmas veiksmas atliekamas yra siunčiamas kodinis skaičius, kad eksperimentinė roboto ranka atsistotų į pradinį tašką ir nuleistų pieštuką į braižymo padėtį. Baigus braižyti kreivę, siunčiamas kodinis skaičius kad baigėsi kreivė reikia pakelti pieštuką. Išskirti kodinius skaičius iš gaunamų duomenų naudojama funkcija *Switch case*. Ciklas atliekamas su kiekvienų duomenų paketu. Nėra laukiama kol bus susiunčiami visi duomenys. Toks algoritmas gerai veikia, kai kreivės yra ilgos ir netiesios. Kitu atveju atsiranda vėlavimas, kai reikia sinchronizuoti eksperimentinės roboto rankos judesį. Nes tolygiam judesiui atilikti reikia:

- signalų siuntimo laiko intervalų parinkimas;
- sinchronizuoti servo pavarų pasisukimus;
- kreivės taškų skaičius.

Kad servo pavaros gautų signalą ne vėliau, nei pasisuka užduotą kampą, bet taip pat ir ne greičiau, nes tokiu atveju signalai persidengia. Tokiu atveju yra praleidžiami tam tikras skaičius kampų verčių. O to pasėkoje, norima nubrėžti tiesę, brėžiama ne taip kaip užduodama. Kad šito neatsitiktų yra dvi galimybės valdyti užduodamu signalų siuntimo intervalus:

1. matlab algoritme trumpinti arba pailginti laiko intervalus, prieš pat siunčiant duomenų paketą;
2. valdymo bloko algoritme trumpinti arba pailginti laiko intervalus, prieš siunčiant signalus į servo pavaras.

Sekantis žingsnis tolygiam judesiui atilikti yra sinchronizavimas. Tai yra servo pavaros turi judėti vienu metu arba su labai mažu uždelsimu, ir servo pavaros idealiu atveju turi pasisukti vienodu poslinkiu. Bet toks atvejis sunkiai įgyvendinamas. Nes braižant kreives nėra galimybes skaidyti taškus pagal servo pavaros poslinkį.

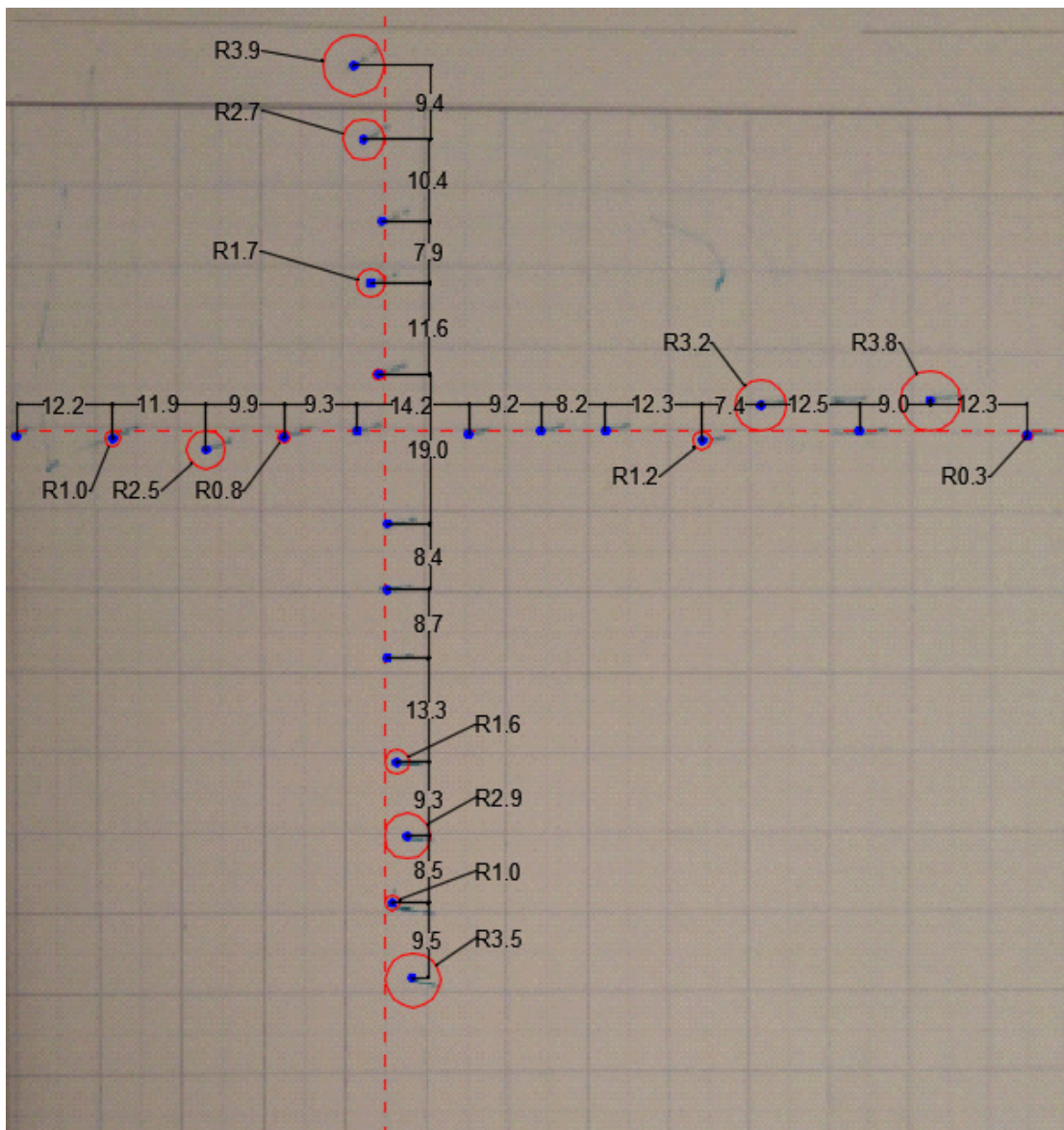


26 pav. Valdymo bloko algoritmo struktūrinė schema

2.6. Eksperimentinė dalis

2.6.1. Eksperimentinės roboto rankos tikslumo tyrimas

Prieš pradėdant eksperimentuoti su roboto ranka, ir išdiriant sukurtą algoritmą. Reikia iširti eksperimentinės roboto rankos tikslumą. Tada bus galima daryti prielaidą, kokios yra braižymo galimybės šios eksperimentinės roboto rankos. Norint iširti eksperimentinės roboto rankos tikslumą reikia sukurti metodiką kaip galima būtų nustatyti tikslumą. Kadangi braižymo lentoje sunku būtų tiksliai pažymėti tašką kuri norima nubrėžti ir tik po to liepti eksperimentiniai roboto rankai ją nubrėžti, ir po to paskaičiuoti paklaidą. Buvo priimtas sprendimas pasirinkti tikrinti tikslumą pagal eksperimentinės roboto rankos sužymėtus taškus vienas kito atžvilgiu. Buvo pasirinktos dvi tiesės braižymo ribose. Vieną kitą kertančios statmenai. Toje tiesėje atidėti kas 10 milimetrų nubraižyti taškus (27 pav.). Tokiu būdu galima pamatuoti kiek sudėlioti taškai nukrypo nuo tikslo. Kaip matoma iš 27 paveikslo, kad eksperimentinė roboto ranka nėra pakankamai tiksli. Nuokrypis gali siekti net iki 3.9mm.



27 pav. Eksperimentinės roboto rankos tyrimo rezultatas

Turint tikslumo rezultatus, galima išanalizuoti, kodėl eksperimentinės roboto rankos netikslumas yra didelis. Analizuojant rezultatus buvo padarytos 3 prielaidos turinčios didžiausią įtaką:

1. kampų vertės buvo apvalinamos iki sveikųjų skaičių;
2. eksperimentinės roboto rankos konstrukcijos įtaka;
3. ketvirtosios servo pavaros piešimo režimu nuleidžiama per daug, kad pasiektų pieštukas popieriaus lapą.

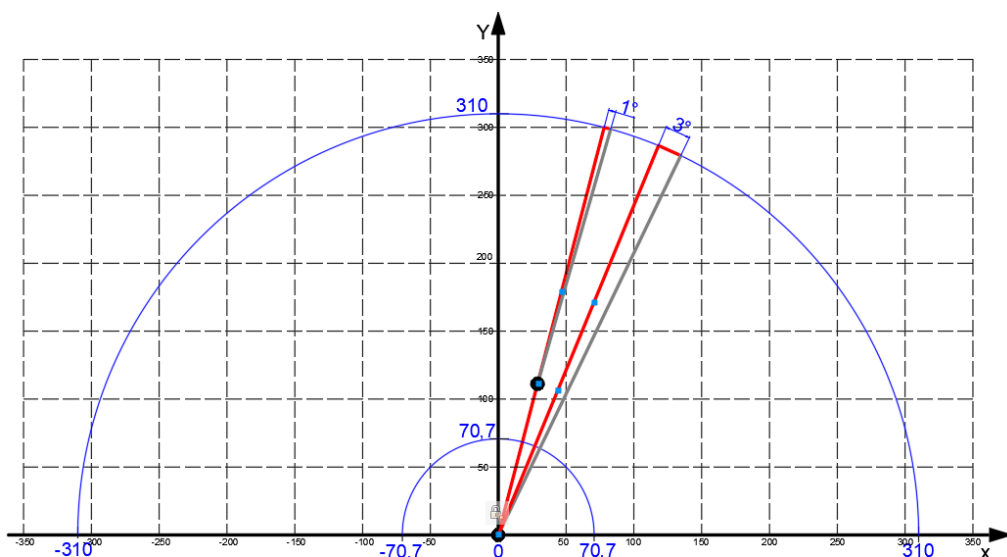
Plačiau apie pirmąją prielaidą. Kadangi servo kampai apvalinami iki sveikųjų skaičių galima daryti prielaidą, kad tai turi įtakos eksperimentinės roboto rankos galutiniam rezultatui.

Buvo iširta ar galima būtų į algoritmą įterpti kurios perduotų signalų vertes vieno skaičiaus po kablelio tikslumo. Po tyrimo buvo priimta išvada, kad vienas skaičius po kablelio neturi įtakos

servo pavarai. Buvo išanalizuota kodėl neturi įtakos. Kadangi mažiausia signalo reikšmė kuri gali būti siunčiama į servo pavara yra 1μs. taigi reikia sužinoti kiek laipsnių sudaro ta viena μs.

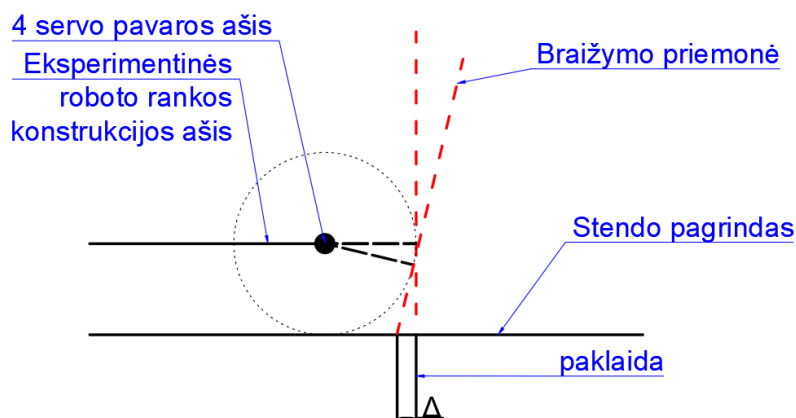
Buvo ištirta servo pavarų mažiausia ir didžiausia signalų reikšmės mikrosekundėmis. Didžiausia reikšmė, kuri leidžia servo pavarai pasisukti iki 180° padėties atitinka 2480 μs. O mažiausia reikšmė, kuri leidžia servo pavarai pasisukti iki 0° padėties atitinka 680μs. Gaunama, kad 1° laipsnis atitinka 10μs vertę. Taip gaunama, jog 0.1° atitinka 1μs. Gali pasirodyti, kad servo pavara turėtų labai tiksliai pasisukti. Bet įtakoti neleidžia pačios servo pavaros jautrumas. Pagal aprašymą šių servo pavarų jautrumas yra 5μs, taigi ką tai reiškia. O reiškia tą, kad servo pavara nereaguoja į signalus kurių reikšmė mažesnės nei 5μs, kitaip tariant šita reikšmė yra jos paklaida. Kad šitą teiginį būtų pagrįsta buvo atliktas tyrimas. Buvo išardyta servo pavara, ir buvo stebima ar tikrai nereaguoja į tokias mažas signalų reikšmes. Buvo siunčiami signalai nuo 1μs iki 20μs reikšmės, didinant po 1μs kas kelias sekundes. Tokiu būdu buvo pastebėta, kad servo pavara iš tiesų signalams iki 5μs nereaguoja. Tyrimo metu buvo pastebėtas dar vienas dalykas, tai kad signalai iki 10 μs neturi įtakos pagrindinio dantračio pasisukimui. Tokiu būdu galima daryti pagrindinę išvadą, kad mažiausia reikšmė kuria galima siųsti į servo pavarą ir turėtų įtakos pasisukimui yra daugiau nei 10μs. O tai yra lygu vienam laipsniui.

Prieš tai aprašyta prielaida daro didžiausia įtaka eksperimentinės roboto rankos tikslumui. Bet ji nevienintelė prielaida daranti įtaka tikslumui. Antroji prielaida turinti įtakos tikslumui yra pati eksperimentinės roboto rankos konstrukcijos ypatumai. Kadangi naudojamos analoginės servo pavaros, kurios po kurio laiko nuo naudojimo pradžios, dėl atsiradusių apkrovų, pradeda klibėti, tai yra atsiranda laisvumas. Kuris gali siekti net iki 3° pirmoje servo pavaroje. Ir iki 1° laipsnio antroje servo pavaroje. Trečios servo pavaros, dėl jai tenkančios mažiausios apkrovos, laisvumas nyksta mažas, todėl jos galima neverti (28 pav.). Kada paklaida atsiranda negali numatyti, nes judesio metu roboto ranka iš inercijos stojant nubėga iki laisvumo ribos, ir gali grįžti ne į užduota pozicija, bet gali ir grįžti. Atvejais kada atsitinka vienas ar kitas atvejas sunku nustatyti netgi norint pasukti tą patį kampą. Taigi sudėjus abiejų servo pavarų laisvumus, gaunamas galimas nukrypimas net iki 4° laipsnių. Kuris gali dar didėti. Netgi galima daryti prielaidą, kad dabar nevertintos 3 servo pavaros paklaidos, po ilgesnio naudojimo ir šitoje servo pavaroje gali atsirasti laisvumo kampai.



28 pav. Eksperimentinės roboto rankos servo pavarų laisvumo tyrimas

Trečioji prielaida yra braižymo priemonės nuleidimas (29 pav.). Kadangi 4 servo valdo braižymo priemonės poziciją. O priemonę nuleidžia radiuso trajektorija. Tai braižymo priemonė neatsiduria statmena stendui. Norint, kad braižymo priemonė pasiektų popierių, reikia 4 servo pavarą pasukti daugiau nei 90° laipsnių stendo atžvilgiu. Dėl to braižymo priemonės galas atsiranda netiksloje vietoje. Tokiu būdu atsiranda paklaida kuri nėra didelė, bet turi įtakos galutiniam rezultatui.



29 pav. Eksperimentinės roboto rankos ketvirtos servo pavaros paklaidos tyrimas

Apibendrinat eksperimentinės roboto rankos tikslumo tyrimus, buvo nustatytos kelios priežastys, kurios daro didžiausia įtaka netikslumui. Didžiausia įtaka daro servo pavarų kampų apvalinimas iki sveikųjų skaičių prieš siunčiant jį į servo pavarą. Taip pat didelę įtaka tikslumui daro, gal net didesnę nei kampų apvalinimas, analoginių servo pavarų laisvumas, kuris atsiranda dėl servo pavarai patiriamos apkrovos, atsiranda oro tarpeliai tarp dantračių. Dėl kurių servo pavara gali neatsistoti į norimą poziciją. Ir mažiausia įtaka daro brėžimo priemonės pozicija, ją nuleidus į brėžimo režimą. Įvertinus visas paklaidas: tai yra diskretiškumo paklaida 1° laipsnis

atitinkantys 6mm, laisvumo paklaida gali siekti iki 4° laipsnių atitinkantys 25mm, pieštuko paklaida gali siekti iki 2mm priklausomai kurioje pozicijoje yra roboto ranka. Sudėjus visas paklaidas gaunama, kad netikslumas gali siekti net iki 5,2° arba 33mm.

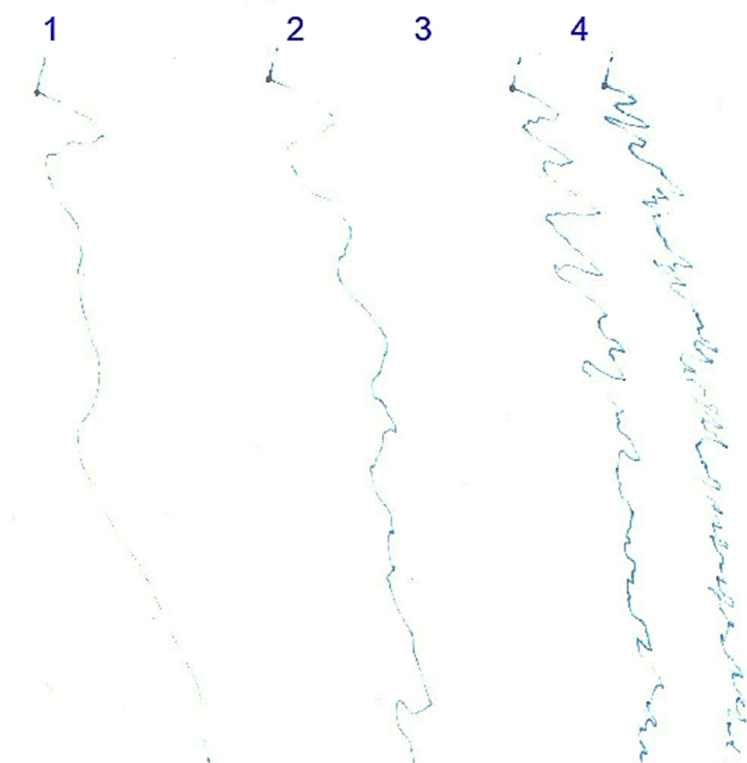
2.6.2. Eksperimentinės roboto rankos judesio tyrimas

Ištyrus eksperimentinės roboto rankos tikslumą. Ir padarius išvadas, kad eksperimentinė roboto ranka nėra tiksli, dėl tam tikrų tikslumui įtakojančių faktorių. Sekantis žingsnis buvo ištirti kaip eksperimentinė roboto ranka braižo vertikales ir horizontalias tieses. Tyrimas buvo atliekamas atsižvelgiant į taškų kiekį tiesėje.

Pirmuose bandymuose nepavyko nubrėžti norimas tieses, nes judesiui didelę įtaką darė servo pavarų laisvumas. Kai dėl servo pavarų laisvumo eksperimentinė roboto ranka darant judesį švytuoja (30 pav.). Tiesės buvo brėžiamos skirtingais žingsniais:

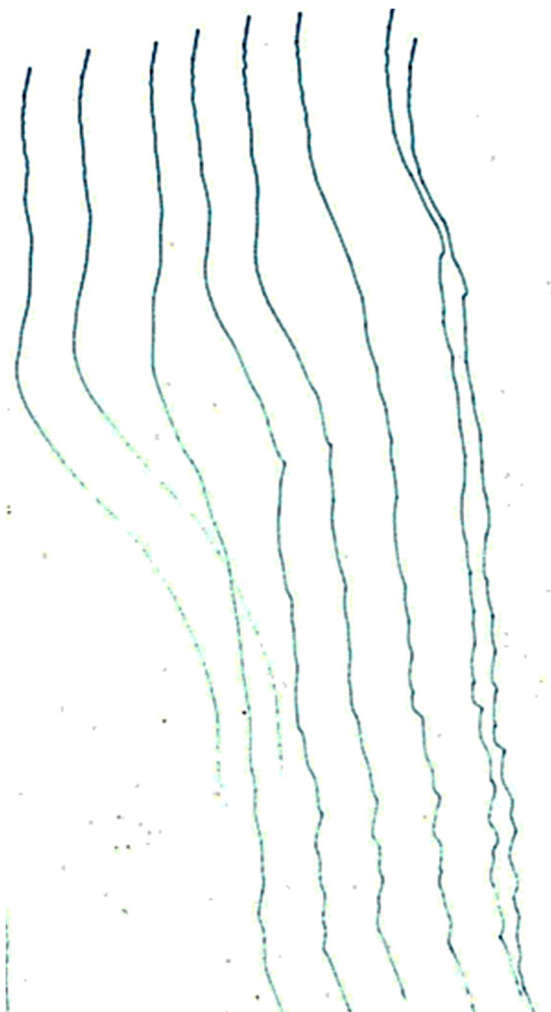
- 1 tiesėje taškai imami kas 10mm,
- 2 tiesėje taškai imami kas 5mm;
- 3 tiesėje taškai imami kas 2mm;
- 4 tiesėje taškai imami kas 1mm.

Tokiu principu bus vaduojamasi viso tyrimo metu. Dėl per didelės laisvumo įtakos, neimama padaryti išvadas, dėl žingsnio įtakos tolygiam judesiui.



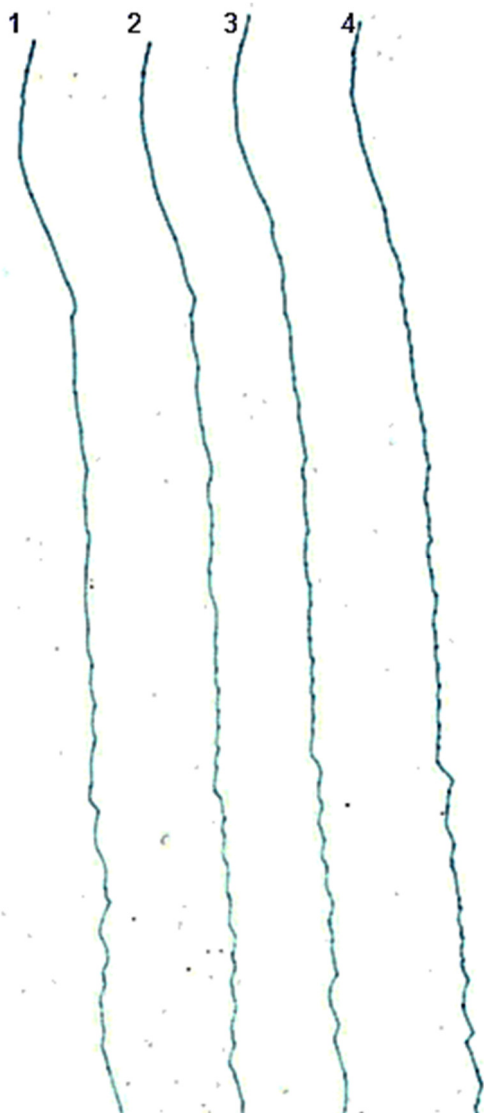
30 pav. Eksperimentinės roboto rankos servo pavarų laisvumo įtaka, kur: 1 - tiesė, žingsnis 10; 2 - tiesė, žingsnis 5; 3 - tiesė, žingsnis 2; 4 - tiesė, žingsnis 1

Padarius kelis pakeitimus eksperimentinės roboto rankos konstrukcijoje. Kurios leido sumažinti švytavimus. Tereikėjo, kad roboto ranka buvo truputi gautų įtempimą, kuri roboto ranką įtemptų į viršų. Sumažinus švytavimus atsirado kita problema. Synchronizavimas (31 pav.), kai servo pavaros skirtingus pasisukimus įveikia skirtingais laikais. 31 paveiksle pirmo dvi tiesės puikiai parodo ką reiškia, kai servo pavaros nėra sinchronizuotos. Iš kitų tiesių matoma, kaip vyksta servo pavarų sinchronizavimas. Visiškai sinchronizuoti servo pavarų neįmanoma, nes naudojamos analoginės servo pavaros, kurių greitį negalima valdyti norint pasiekti norimą tašką. Synchronizavimas vyksta duomenų perdavimo laiko intervalų pagalba. Kai duomenų perdavimo laiko intervalai yra ilginami ar trumpinami. Kad servo pavaros veiktų ne visu greičiu. Didžiausiu greičiu besisukanti servo pavarą didina eksperimentinės roboto rankos apkrovas, taip didėja servo pavarų laisvumas. Dėl didelių smūgiu sustojant ir pradedant judėti. Kad greiti sumažinti naudojama laiko įterptis, kai į signalas kampo signalą yra įterpiamos pauzės. Tokiu būdu norimo kampo signalas ištešiamas iki tam tikro laiko intervalo, tokiu būdu servo pavara sukasi lėčiau iki norimo kampo.



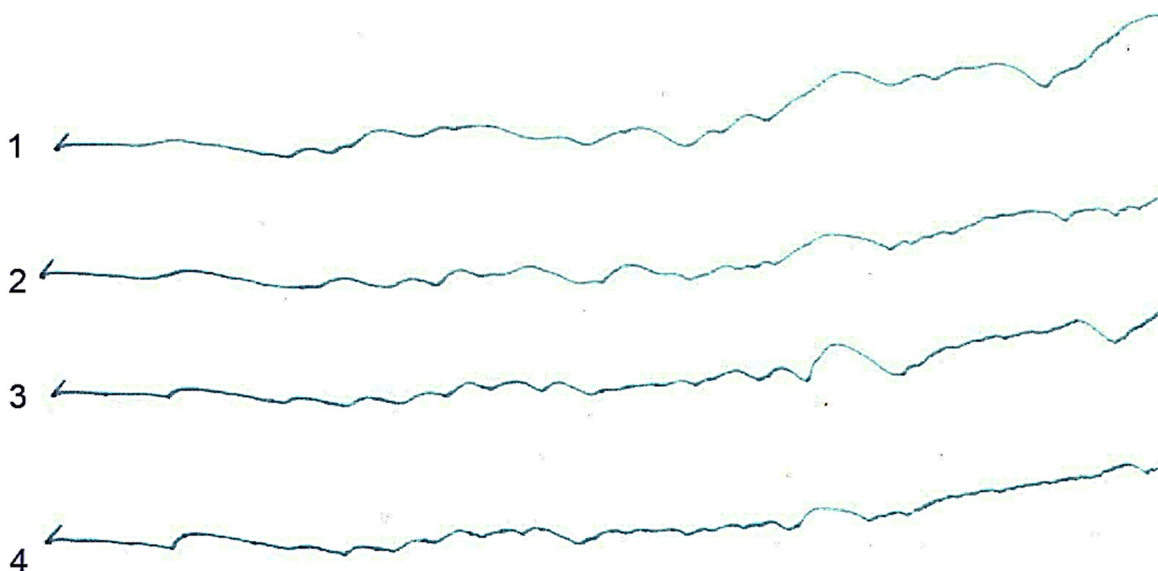
31 pav. Eksperimentinės roboto rankos sinchronizavimo tyrimo rezultatai

32 paveiksle pavaizduotos vertikalios tiesių priklausomybė nuo taškų žingsnio braižymo tyrimas. Kuo daugiau tiesėje turint taškų tuo tikslesnę tiesę galima nubraižyti. Iš 32 paveikslo galima konstatuoti, kad tiesė nesigauna idealios, tam įtaka daro tikslumo trukumas, kuris jau buvo aptartas.



32 pav. Eksperimentinės roboto rankos vertikalios tiesės brėžimo tyrimas, kur: 1 - tiesė, žingsnis 10; 2 - tiesė, žingsnis 5; 3 - tiesė, žingsnis 2; 4 - tiesė, žingsnis 1

33 paveiksle matoma horizontalių tiesių braižymo tyrimas 4 žingsnio intervalais. Galima padaryti išvadą iš rezultatų, kad vertikalus ir horizontalus braižymas skiriasi. Vertikalios tiesės artimesnės idealiai tiesiai. Horizontalioms tiesėms didesnė įtaka daro nei vertikalios servo pavarų sinchronizavimas. Nes brėžiant, pirma servo pavarą turi pasisukti didesnę kampą nei 2 ir 3 servo pavaros.



33 pav. Eksperimentinės roboto rankos horizontalios tiesės brėžimo tyrimas, kur: 1 - tiesė, žingsnis 10; 2 - tiesė, žingsnis 5; 3 - teisė, žingsnis 2; 4 - tiesė, žingsnis 1

Apibendrinat judesio tyrimo rezultatus gali daryti prielaida, kad judesiui turi įtakos servo pavarų sinchronizavimas. Turimų kreivės taškus skaičius. Didelę įtaka daro eksperimentinės roboto rankos tikslumo trukumas. Tolimesniems eksperimentams buvo pasirinkta braižyti mažiausiu žingsniu. Todėl, kad paveiksluose didžioji dalis kreivių būna trumpos, todėl bijant prarasti taškų ar net kreivės atkarpų, buvo priimtas sprendimas kreives braižyti su taškų intervalais 1mm.

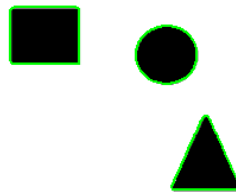
Kadangi didelio poveikio braižymo metu neįmanoma suskirstyti kreives į vertikalias ir horizontalias, buvo priimtas sprendimas eksperimentinę ranka sureguliuoti, taip kad parametrai kiek įmanoma tiksliau braižytu abejas kreives. Braižymo metu servo pavarų parametrų keisti labai sunku arba kitais atvejais neįmanoma. Todėl buvo priimtas sprendimas, kad braižant didelius poveikslus parametrų keisti negalima.

2.6.3. Eksperimentinės roboto rankos valdymo galimybių tyrimas

Ištirus eksperimentinės roboto rankos tikslumą bei jos galimybes atlikti tolygius judesius, galima konstatuoti, kad galutiniai rezultatai skirsis nuo siekiamo rezultato. Norint iširti kaip šios savybės paveikia struktūrų braižymą buvo pasitektos tris elementarios figūros (34 pav.): kvadratas, apskritimas, trikampis. Šios tris figūros leidžia pamatyti roboto rankos judesių atlikimą skirtingose situacijose. Bus patikrinta kaip roboto ranka atlieka paprastus judesius: įstrižainės, lanko, vertikalaus, horizontalaus. Pasirinktos figūros buvo nupieštos užtušuos. Pasirinktas poveikslas pirmiausiai buvo vektorizuotas (35 pav.). Žalia linija pažymėta roboto rankos judėjimo trajektorija.

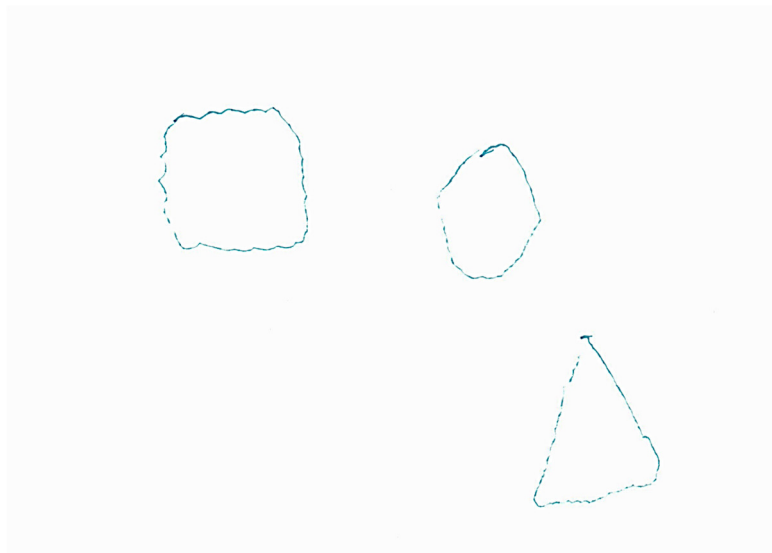


34 pav. Eksperimentui parinktas paveikslas



35 pav. Rezultatas po vektorizavimo

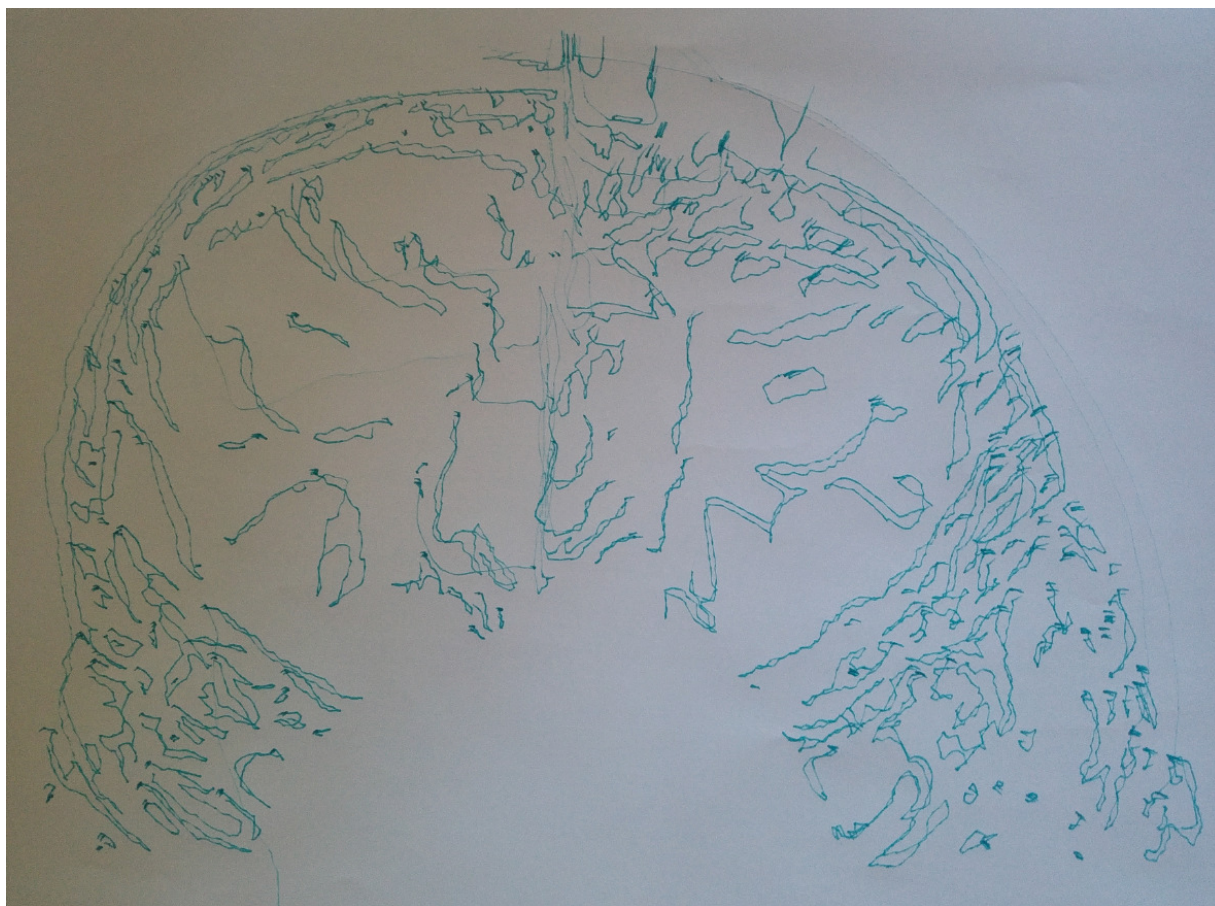
Atlikus vektorizavimą ir apskaičiavus koordinates galima skaičiuoti kampus ir siųsti juos į servo pavaras. Gautas rezultatas matomas (36 pav.). Figūros gavosi panašios, bet ne tokios, kokios turėtų gautis, pagal 35 paveikslą.



36 pav. Valdymo sistemos gautas galutinis rezultatas

Atlikus visus tyrimus, atsirado poreikis ištirti kaip eksperimentinė roboto ranka braižo didelius brėžinius su nestandartinėmis kreivėmis. Tam buvo pasitelktas jau ištirtas 13 paveikslas vektorizavimo skyriuje. Gautas rezultatas matomas (37 paveiksle). Paveikslo apačioje per vidurį aiškiai matoma mažoji pasiekimo riba. Kuri įsiterpia į paveikslą ir užima nemažą dalį brėžinio. Dar galima pamatyti neryškių įbrėžimų, jos atsiranda tada kai valdymo bloke įvyksta klaida ir

roboto ranka nepakeldama piešimo priemonės nuvažiuoja iki pradinio taško. Tokie įvykiai yra neprognozuojami, ir gali atsitikti bet kuriuo momentu, kaip ir jų dažnumas neprognozuojamas.



37 pav. Eksperimentinės roboto rankos valdymo sistemos tyrimo vaizdas

Apibendrinus šio skyriaus ištirtus rezultatus galima teigti, kad tikslumas ir judesio atlikimas yra tiesiogiai susiję. Tikslumą įtakoja trys veiksniai: pirmas veiksnys roboto rankos jungčių kampų apskaičiavimas vyksta realiais skaičiais, antras veiksnys servo pavarų laisvumas atsirandantis dėl oro tarpelių tarp dantračių, trečias veiksnys piešimo priemonės nuleidžiama ne statmenai stendo atžvilgiu.

Tolygiems judesiams atlikti įtaka daro servo pavarų sinchronizavimas. Kai servo pavaros pradeda ir baigia posūkius tuo pačiu metu. Kadangi naudojamos analoginės RC servo pavaros, kurių greitis nėra valdomas, idealiai sinchronizuoti servo pavaras neįmanoma.

Kreivių braižymo rezultatai leidžia teigti, kad taškų skaičius leidžia nubrėžti tikslesnes kreives. Buvo priimtas sprendimas tolimesniems eksperimentams naudoti 1 mm taškų intervalą.

IŠVADOS

1. Išanalizuotos analogiškų robotų rankų tipai. Nustatyta, kad braižymui užtenka dviejų laisvės laipsnių roboto rankos. Tinkamiausia konstrukcija su kuria galima braižyti paveikslus yra trijų laisvės laipsnių.
2. Nustatyta, kad vienas iš būdu vektorizuoti paveikslą yra „canny“ operatoriaus naudojimas Matlab aplinkoje, o vieno vektoriaus koordinačių suradimui naudojamas „bwtraceboundary“ operatorius.
3. Sudaryta eksperimentinės roboto valdymo sistema. Ją sudaro du pagrindiniai įrenginiai: valdymo blokas ir asmeninis kompiuteris. Valdymo sistemos algoritmą sudaro keturios nuosekliai veikiančios dalys.
4. Sudarytas ir išbandytas teoriškai bei eksperimentiškai atvirkštinės kinematikos uždavinio sprendimo būdas, įvertinantys specifinę kinematinę roboto rankos schemą.
5. Ištirtas eksperimentinės roboto rankos tikslumas. Eksperimentinės roboto rankos nuokrypiai gali siekti iki 33 mm. Nustatyta, kad nuokrypiams didelę įtaką turi: kampų apvalinimas iki sveikųjų skaičių, servo pavarose atsirandantis laisvumas tarp krumpliaračių.
6. Ištirtos eksperimentinės roboto rankos valdymo galimybės. Nustatyta, kad didelę įtaka tolygiems judesiam atlikti turi: servo pavarų sinchronizavimas, taškų kiekis, tikslumas.
7. Nustatyta, kad eksperimentinė roboto ranka nėra tinkama naudoti piešimui, dėl naudojamų analoginių RC servo pavarų.

LITERATŪROS SĄRAŠAS

1. NPTEL – Mechanical – Mechatronics and Manufacturing Automation. Module 7: CNC Programming and Industrial Robotics.
2. Jason Price, Gareth Walley, Peter Hill, Lyudmil Vladimirov, Tsvetan Zhivkov. Robot Object Search and Retrieval Team, 2015.
3. Adelhard Beni Rehiara. Kinematics of AdeptThree Robot Arm, 2011.
4. V.Kumar. Kinematics of a Planar 3R Manipulator.
5. Arduino Matlab Robotic Arm Coding. Prieiga per internetą:
<<https://www.youtube.com/watch?v=NBSizR01yX0>> [žiūrėta 2015-10-15]
6. Modeling Inverse Kinematics in a Robotic Arm. Prieiga per internetą:
<<http://se.mathworks.com/help/fuzzy/modeling-inverse-kinematics-in-a-robotic-arm.html>>
[žiūrėta 2015-10-15]
7. Lab 1 - Forward and Inverse Kinematics and Robotic Toolbox. Prieiga per internetą:
<http://www.moodle2.tfe.umu.se/pluginfile.php/56443/mod_resource/content/1/Lab%201.pdf>
[žiūrėta 2015-10-15]
8. Mohammed Reyad AbuQassem. Simulation and Interfacing of 5 DOF Educational Robot Arm.
9. Motion Actuation in SimMechanics R2013b. Prieiga per internetą:
<<http://blogs.mathworks.com/simulink/2013/11/13/motion-actuation-in-simmechanics-r2013b/>>
[žiūrėta 2015-10-15]
10. Inverse Kinematics / Math for a Robotic Arm. Prieiga per internetą:
<<http://www.cerebralmeltdown.com/Robot%20Arm/Robot%20Arm%20Math/index.html>>
[žiūrėta 2015-10-15]
11. Internet Controlled Robotic Arm. Prieiga per internetą:
<<http://www.rustynailworkshop.com/archives/8>> [žiūrėta 2015-10-15]
12. Viliam Fedák, František Ďurovský and Róbert Ůveges. Analysis of Robotic System Motion in SimMechanics and MATLAB GUI Environment.
13. Introduction to Computer Vision. Prieiga per internetą:
<http://cs.brown.edu/courses/cs143/proj5/>> [žiūrėta 2015-10-15]
14. Vytenis Sinkevičius. 10-ties judrumo laipsnių valdomo griebtuvo prototipas ir jo panaudojimo galimybių tyrimas, 2015.
15. Brian Evans. Beginning Arduino Programming, 2011.
16. Drawing robot. Prieiga per internetą: <<http://www.instructables.com/id/Drawing-Robot/step4/Coding/>> [žiūrėta 2015-10-15]

PRIEDAI

1. VALDYMO SISTEMOS ALGORTIMAI

1.1. Vektorizavimo algoritmas

```

clear a;
clc;clf
BW4=imread('7.png');
V=imresize(BW4,[310 610]);
I=rgb2gray(V);
F=edge(I,'canny',[.01,.25]);
imshow(V);
impixelinfo;
set(0,'RecursionLimit',2000);
X=[];
Y=[];
d=[];
xxxxx=[];
yyyyy=[];
e=size(F);
k=1;
ss=e;
tt=1;
while (ss(1,1)>0)
for m=2:1:e(1,1)
    for n=2:1:e(1,2)
        if F(m,n)==1
            if F(m-1,n-1)==1 || F(m,n-1)==1 || F(m-1,n)==1 || F(m+1,n-1)==1 || F(m-1,n+1)==1 ||
F(m,n+1)==1 || F(m+1,n)==1 || F(m+1,n+1)==1
                X(1,k)=n;
                Y(1,k)=m ;
                k=k+1;
            else
                F(m,n)=0;
            end
        end
    end
end

```

```

    end
end
end
d=bwtraceboundary(F,[Y(1), X(1)],'E',8,Inf,'clockwise');
hold on;
plot(d(:,2),d(:,1),'g','LineWidth',2);
s=size(d);
for r=1:1:s(1,1)
xxxxx(r,tt)=d(r,2);
yyyyy(r,tt)=d(r,1);
end
for p=1:1:s(1,1)
    xxx=d(p,2);
    yyy=d(p,1);
    if ((xxx-3)>0&&(yyy-3)>0)&&(((yyy+2)<e(1,1))&&(xxx+2)<e(1,2))
        for i=yyy-1:1:yyy+1
            for j=xxx-1:1:xxx+1
                if F(i,j)==1
                    F(i,j)=0;
                end
            end
        end
    end
end
else
    F(yyy,xxx)=0;
end
end
[xxxx yyy]=find(F) ;
ss=size(xxxx);
tt=tt+1;
k=1;
end

```

1.2. Kampų skaičiavimo algoritmas

```

% rankytes modelis, standartine DH transformacija
clc;clf % isvalyti komandos langa ir grafikus
%_____
delete (instrfind({'port'},{'COM8'}));
clear a;
global s;
s=serial('COM8','BAUD', 9600);
fopen(s);
%%%_____
result1=[];
result11=[];
result22=[];
DH = [0, 0, 0, pi/2; % pirma jungtis
      0, 115, 0, 0; % antra jungtis
      0, 70, 0, 0; % trecia jungtis
      0, 123, 0, 0; % ketvirta jungtis
      0, 1, 0, pi/2]; % laisva
a0=DH(1,2);a1=DH(2,2);a2=DH(3,2);a3=DH(4,2);a4=DH(5,2);
w = ['r';'r';'r';'r'];
symbolic =0 ; xxa=0; yy=0;
plane = [0; 0; 0; 1]; % nulines koordinates irankiui arba bazei
maxxyz=abs(DH(1,2))+DH(2,2)+DH(3,2)+DH(4,2); % maksimalus ilgis rankos
h4 = axes('position', [ .1 0.1 .8 .8]);
axis([-maxxyz maxxyz 0 maxxyz -maxxyz DH(4,2)]) % X,Y,Z asiu ribos
axis equal % ('square')
view([0,0,1]); % view([1,1,1])
grid on
%-----define the objects called bar , hinge , and path .
data = [0 0; 0 1;0 1] ; % coordinates of endpoints of the bar
data0= [0 0 ; 0 1; 0 1] ; % xy centro

```

```

bar0
line('xdata',data(1,:),'ydata',data(2,:),'zdata',data(3:),'linewidth',6,'color','black','erase','xor');
bar1 = line('xdata',data(1,:),'ydata',data(2,:),'zdata',data(3:),'linewidth',5,'color','b','erase','xor');
bar2 = line('xdata',data(1,:),'ydata',data(2,:),'zdata',data(3:),'linewidth',4,'color','g','erase','xor');
bar3 = line('xdata',data(1,:),'ydata',data(2,:),'zdata',data(3:),'linewidth',3,'color','r','erase','xor');
bar4 = line('xdata',data(1,:),'ydata',data(2,:),'zdata',data(3:),'linewidth',2,'color','r','erase','xor');
htrail = line(xxa,yy,'marker','.', 'color','r', 'erase','none') ;
% konstantu skaiciavimas
aa0=a1;
aa1=a2;
aa2=a3;
aa3=a4;

b01=-aa1*(aa0+aa2);
b02=b01^2;
b03=4*aa0*aa2;
b04=aa0*aa0+aa1*aa1+aa2*aa2-2*aa0*aa2;
ra = aa0+aa1+aa2;% maksimalus rankos ilgis
% nustatomos ribos linijiniam judėjimui
% dx=1; % linijinio judesio zingsnis
% xmax=150;
% xmin=150;
% ymax=155;
% ymin=150;
% dy=1;

f=size(xxxxx);
i=1;
ti=1;
ri=1;
pause(4);
while ri<f(1,2)+1
    for ti=1:1:f(1,1)
        xxa=xxxxx(ti,ri)-305;
        yy=310-yyyyy(ti,ri);

```

```

if xxa>0
axy = xxa*xxa+yy*yy;
r=sqrt(axy);

posa= b02-b03*(b04-axy);% posaknis ir jo teigiamumas
if posa >= 0 && r <= ra % ar pasiekia ranka taska
B12p = (b01+sqrt(posa))/b03;
tetabp = sign(-xxa)*acos(B12p); %
if tetabp > -pi/2 && tetabp < pi/2 %
tetabm = acos(B12p);
fi = atan(yy/xxa);
xg = aa0+aa1*cos(tetabp)+aa2*cos(2*tetabp);
yg = aa1*sin(tetabp)+aa2*sin(2*tetabp);
fig = atan(yg/xg);
% teto kampas is pasukto paveikslo
teta00 = fi-fi;
if tetabp > 0
teta00 = pi+fi-fi;
end
if teta00 > 0 && teta00 < pi % pirmo kampo sukimosi ribos
% surasti kampai priskiriami DH parametrms

DH(1,4) = teta00;
DH(2,4) = tetabp;
DH(3,4) = tetabp;
DH(4,4) = pi/4;

pos1=180*teta00/pi;
pos11=round(pos1)+101-9;
%pos111=round(pos11);
pos2=(180*tetabp/pi)+90;
pos22=round(pos2)+100;
pos3=(180*tetabp/pi)+90;
pos33=round(pos3)+100;
result = strcat(num2str(pos11),num2str(pos22));

```

```

result11(ti,ri)= str2num(result);

% cia skaiciuojamas tiesioginis uzdaviny
to=0;
for from=1:5
f = FKmod(DH,w,symbolic,from,to);
tool(:,from) = f*plane;
end
data0(1,2)=tool(1,1);
data0(2,2)=tool(2,1);
data0(3,2)=tool(3,1);
set(bar0, 'xdata', data0(1,:), 'ydata', data0(2,:), 'zdata', data0(3,:));
set(bar1, 'xdata', tool(1,1:2), 'ydata', tool(2,1:2),'zdata', tool (3,1:2));
set(bar2, 'xdata', tool(1,2:3), 'ydata', tool(2,2:3),'zdata', tool (3,2:3));
set(bar3, 'xdata', tool(1,3:4), 'ydata', tool(2,3:4),'zdata', tool (3,3:4));
set(bar4, 'xdata', tool(1,3:5), 'ydata', tool(2,3:5),'zdata', tool (3,3:5));
set(htrail,'xdata', xxa, 'ydata', yy);
drawnow
pause(0.01)
    end
    end % cia pabaiga, kai teta12 telpa tarp -pi/2 ir pi/2
    end % cia pabaiga, kai teto yra tarp 0 ir pi
else
axy = xxa*xxa+yy*yy;
r=sqrt(axy);

    posa= b02-b03*(b04-axy);% posaknis ir jo teigiamumas
if posa >= 0 && r <= ra % ar pasiekia ranka taska
    B12p = (b01+sqrt(posa))/b03;
    tetabp = sign(-xxa)*acos(B12p); %
if tetabp > -pi/2 && tetabp < pi/2 %
    tetabm = acos(B12p);
    fi = atan(yy/xxa);
    xg = aa0+aa1*cos(tetabp)+aa2*cos(2*tetabp);
    yg = aa1*sin(tetabp)+aa2*sin(2*tetabp);

```

```

    fig = atan(yg/xg);
% teto kampas is pasukto paveikslo
    teta00 = fi-fi;
    if tetabp > 0
        teta00 = pi+fi-fi;
    end
    if teta00 > 0 && teta00 < pi % pirmo kampo sukimosi ribos
% surasti kampai priskiriami DH parametrans

    DH(1,4) = teta00;
    DH(2,4) = tetabp;
    DH(3,4) = tetabp;
    DH(4,4) = pi/4;

    pos1=180*teta00/pi;
    pos11=round(pos1)+101-9;
    %pos111=round(pos11);
    pos2=(180*tetabp/pi)+90;
    pos22=round(pos2)+100;
    pos3=(180*tetabp/pi)+90;
    pos33=round(pos3)+100;
    result = strcat(num2str(pos11),num2str(pos22));
    result22(ti,ri)= str2num(result);

% cia skaiciuojamas tiesioginis uždavinys
%     to=0;
%     for from=1:5
%         f = FKmod(DH,w,symbolic,from,to);
%         tool(:,from) = f*plane;
%     end
%     data0(1,2)=tool(1,1);
%     data0(2,2)=tool(2,1);
%     data0(3,2)=tool(3,1);
%     set(bar0, 'xdata', data0(1,:), 'ydata', data0(2,:), 'zdata', data0(3,:));
%     set(bar1, 'xdata', tool(1,1:2), 'ydata', tool(2,1:2),'zdata', tool (3,1:2));

```

```

%      set(bar2, 'xdata', tool(1,2:3), 'ydata', tool(2,2:3),'zdata', tool (3,2:3));
%      set(bar3, 'xdata', tool(1,3:4), 'ydata', tool(2,3:4),'zdata', tool (3,3:4));
%      set(bar4, 'xdata', tool(1,3:5), 'ydata', tool(2,3:5),'zdata', tool (3,3:5));
%      set(htrail,'xdata', xxa,      'ydata', yy);
%      drawnow
%      pause(0.01)
      end
      end % cia pabaiga, kai teta12 telpa tarp -pi/2 ir pi/2
      end % cia pabaiga, kai teto yra tarp 0 ir pi
      end
end % cia pabaiga, kai posaknis teigiamas
%%end
ri=ri+1;
end

rez=size(result11);
rez1=size(result22);
for oo=1:1:rez(1,2)
    fprintf(s,'%6d',result11(1,oo));
    pause(4);
    fprintf(s,'%6d',123456);
    pause(2);
    for o=1:1:rez(1,1)
        if result11(o,oo)>0
            fprintf(s,'%6d',result11(o,oo))
            pause(0.05);
        end
    end
end
fprintf(s,'%6d',999999);
pause(1);

end

for oo=1:1:rez1(1,2)
    fprintf(s,'%6d',result22(1,oo));

```



```
pause(4);
fprintf(s,'%6d',654321);
pause(2);
for o=1:1:rez1(1,1)
    if result22(o,oo)>0
        fprintf(s,'%6d',result22(o,oo))
        pause(0.05);
    end
end
fprintf(s,'%6d',999999);
pause(1);

end

fclose(s)
```

1.3. Duomenų perdavimo algoritmas

```
delete (instrfind({'port'},{'COM3'}));% nustatymai
clear a;
global s;
s=serial('COM3','BAUD', 9600); % Com portas turi sutapti su Arduino komunikavimo portu.
fopen(s); %atidaroma komunikacija
fprintf(s,'%6d',servalue); %duomenų siuntimo funkcija
pause(1);%uždelsimas
fclose(s); % uždaromas komunikacija.
```

1.4. Valdymo bloko algoritmas

```

#include <VarSpeedServo.h>
int servo1,servo2, servo3;
//LiquidCrystal lcd(12,11,10,9,8,7);
VarSpeedServo myservo1;
VarSpeedServo myservo2;
VarSpeedServo myservo3;
VarSpeedServo myservo4;
int i=0;
int p;
void setup()
{
    Serial.begin(9600);
    myservo1.attach(6);//pieštuko servukas
    myservo2.attach(5);//3 servukas
    myservo3.attach(4);//2 servukas
    myservo4.attach(3);//1 servukas
    myservo2.write(94,127,true);
    myservo3.write(88,127,true);
    myservo4.write(99,127,true);
    delay(200);
}
void loop()
{
    long angle = 0;
    if (Serial.available())
    {
        delay(5);
        byte incomingByte = Serial.read();
        for (int n=0;n<6;n++){
            if (incomingByte >= '0' && incomingByte <= '9')
                angle = angle * 10 + (incomingByte - '0');
            incomingByte = Serial.read();
            delay(1);
        }
    }
}

```

```

    }
switch(angle)
{
  case 123456:
    myservo1.write(76,127,true);
    break;
  case 654321:
    myservo1.write(88,127,true);
    break;
  case 999999:
    myservo1.write(110,127,true);
    break;
  case 333333:
    servo1=angle/1000;
    servo2=abs((servo1*1000)-angle);
    servo1=servo1-101;
    servo2=servo2-100;
    myservo4.write(180-servo1,30,false);
    myservo3.write(servo2,30,false);
    myservo2.write(180-servo2,30,false);
    break;
  default:
if (angle>100100 && angle <= 289280)
{
  servo1=angle/1000;
  servo2=abs((servo1*1000)-angle);
  servo1=servo1-101;
  servo2=servo2-100;
  myservo4.write(180-servo1,30,false);
  myservo3.write(servo2,30,false);
  myservo2.write(180-servo2,30,false);
  i++;
}
break;
}

```