



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

Mindaugas Gapšys

**ROBOTO SU UŽDARA KINEMATINE GRANDINE VALDYMO
GALIMYBIŲ TYRIMAS**

Baigiamasis magistro projektas

Vadovas

Prof. Dr. Vytenis Sinkevičius

PANEVĖŽYS, 2017

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

**ROBOTO SU UŽDARA KINEMATINE GRANDINE VALDYMO
GALIMYBIŲ TYRIMAS**

Baigiamasis magistro projektas

Valdymo technologijos (kodas 621H66001)

Vadovas

Prof. Dr. Vytenis Sinkevičius
2017 m. gegužės 31 d.

Recenzentas

Projektą atliko

Mindaugas Gapšys
2017 m. gegužės 31 d.

PANEVĖŽYS, 2017



KAUNO TECHNOLOGIJOS UNIVERSITETAS

PANEVĖŽIO VERSLO IR TECHNOLOGIJŲ FAKULTETAS

(Fakultetas)

Mindaugas Gapšys

(Studento vardas, pavardė)

Valdymo technologijos, 621H66001

(Studijų programos pavadinimas, kodas)

„Roboto su uždara kinematine grandine valdymo galimybių tyrimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 17 m. gegužės 31 d.
Panevėžys

Patvirtinu, kad mano, **Mindaugo Gapšio**, baigiamasis projektas tema „Roboto su uždara kinematine grandine valdymo 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)

TVIRTINU:

KTU Panevėžio technologijų ir verslo fakulteto
Technologijų katedros vedėjas

20..... ..

BAIGIAMOJO PROJEKTO UŽDUOTIS

Išduota studentui: Mindaugui Gapšiui Grupė PME-5

1. Darbo tema:

Lietuvių kalba: Roboto su uždara kinematine grandine valdymo galimybių tyrimas

Anglų kalba: Research of Control Possibilities of Closed Kinematic Chain Robot

Patvirtinta 2017 m. 03 mėn. 30 d. dekanų potvarkiu Nr. V25-13-8

2. Darbo tikslas:

Ištirti duotos konstrukcijos roboto su uždara kinematine grandine valdymo galimybes.

3. Reikalavimai ir sąlygos:

4. Projekto struktūra.

Turinys konkretizuojamas kartu su vadovu, atsižvelgiant į BP pobūdį.

LITERATŪROS APŽVALGA. Atlikti robotų su uždara grandine konstrukcijų ir jų valdymo būdų analizę.

PROJEKTINĖ DALIS. Išanalizuoti tiriamo roboto konstrukciją, sudaryti tiesioginio ir atvirkštinio kinematinio uždavinio sprendimo būdus, sukurti roboto valdymo algoritmą ir jį realizuojančią programą.

TIRIAMOJI DALIS. Ištirti roboto kalibravimo galimybes, roboto judesio valdymo duotai roboto konstrukcijai galimybes.

IŠVADOS. Pateikti išvadas apie galimybes valdyti duotos konstrukcijos robotą.

5. Ši užduotis yra neatskiriama baigiamojo projekto dalis.

6. Projekto pateikimo gynimui kvalifikacinėje komisijoje terminas

2017-05-31

(data)

Užduotį gavau: Mindaugas Gapšys

2017-02-01

(studento vardas, pavardė, parašas)

(data)

Vadovas: Prof. Dr. Vytenis Sinkevičius

2017-02-01

(pareigos, vardas, pavardė, parašas)

(data)

Gapšys, Mindaugas. Roboto su uždara kinematine grandine valdymo galimybių tyrimas. *Magistro* baigiamasis projektas / vadovas Prof. Vytenis Sinkevičius; Kauno technologijos universitetas, Panevėžio verslo ir technologijų fakultetas.

Mokslo kryptis ir sritis:

Reikšminiai žodžiai: *paraleliniai robotai*.

Panevėžys, 2017. 54 p.

SANTRAUKA

Magistro baigiamajame darbe aprašoma paralelinių robotų istorija, taikymo sritys, apibendrintos jų konstrukcijos. Išspręsti tiesioginio ir atvirkštinio kinematikos uždaviniai nagrinėjamam robotui, atliktas roboto kalibravimas. Atliktas judesio trajektorijos nuokrypių priklausomybės tyrimas. Pateikta kalibravimo ir programos algoritmai.

Gapšys, Mindaugas. *Research of Control Possibilities of Closed Kinematic Chain Robot*: Master's thesis in final project / supervisor assoc. prof. Vytėnis Sinkevičius. The Faculty of Panevėžio Verslo ir Technologijų, Kaunas University of Technology.

Research area and field:

Key words: parallel robots.

Panevėžys, 2017. 54 p.

SUMMARY

Master's thesis describes the history of parallel robots, applications, summarizes their design. Resolved direct and inverse kinematics tasks of the present robot, accomplished robot calibration. Performed motion trajectory deviations dependency research. Presented calibration and software algorithms.

TURINYS

IVADAS	8
1. LITERATŪROS APŽVALGA	9
1.1. Robotų su uždara grandine konstrukcijų analizė	9
1.2. Paralelinių robotų vystymosi istorija	12
1.3. Paraleliniai robotai ir jų taikymo pavyzdžiai	16
1.4. Paralelinių robotų valdymas	19
1.5. Piešiantys paraleliniai robotai	21
2. PROJEKTINĖ DALIS	23
2.1. Paralelinio roboto konstrukcija	23
2.2. Paralelinio roboto nustatoma nulinė pozicija.....	24
2.3. Tiesioginio kinematinio uždavinio sprendimas	28
2.4. Atvirkštinio kinematinio uždavinio sprendimas	30
2.5. Valdymo programa	31
2.6. Problemos	33
3. JUDESIO TRAJEKTORIJOS TYRIMAS	34
IŠVADOS	44
LITERATŪRA	45

ĮVADAS

Terminas „robotas“ apima pačias įvairiausias automatikos mašinas ir sistemas. Tačiau šias sistemas išskiria svarbiausia jų savybė: visos jos sukurtos pakeisti žmogų ten, kur reikalinga atlikti sunkų, monotonišką ar pavojingą sveikatai darbą. Taip pat pramoninių robotų poreikis auga norint išvengti žmogiškojo faktoriaus. Didžioji dalis robotų naudojama pramonėje, taip pat dažnai galima sutikti ir paralelinius (lygiagrečiuosius) robotus. Jų analizės ir tyrinėjimai atliekami jau daug metų ir nustatoma įvairių sričių jiems panaudoti. Paraleliniai robotai už nuosekliuosius robotus vertinami dėl tikslesnio ir greitesnio darbo.

Šiame darbe lyginami paraleliniai ir lygiagretūs robotai, nagrinėjamas paralelinių robotų panaudojimas, analizuojamos valdymo galimybės, tiriamos elektromechaninės dalies dinaminės savybės.

Tyrimo objektas – robotas su uždara kinematine grandine.

Darbo tikslas – ištirti duotos konstrukcijos roboto su uždara kinematine grandine valdymo galimybes.

Darbo uždaviniai

1. Robotų su uždara kinematine grandine panaudojimo literatūros analizė.
2. Sukurti roboto su uždara kinematine grandine kalibravimo būdą.
3. Sudaryti tiesioginio ir atvirkštinio kinematinio uždavinio sprendimo būdus.
4. Atlikti roboto su uždara kinematine grandine elektromechaninės dalies dinaminių savybių ir valdymo galimybių tyrimą.

Tyrimo metodai – mokslinės literatūros analizė, skaitiniai skaičiavimai.

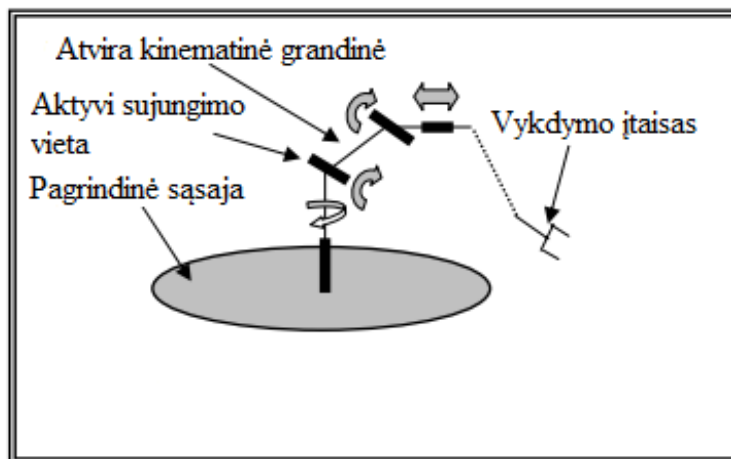
1. LITERATŪROS APŽVALGA

1.1. Robotų su uždara grandine konstrukcijų analizė

Pagal tarptautinės organizacijos standartizaciją (standart1 ISO 8373:2012), robotas yra programuojamas mechanizmas, judantis dviejose ar daugiau ašyse, atliekantis skirtas užduotis ribotoje aplinkoje. Nuoseklieji robotai (1 pav.) (*serial robot*) yra atviros kinematinės grandinės, kurių galiniame taške tvirtinamas vykdymo įtaisas. Judesiu valdyti naudojamos sukamojo ar slenkamojo tipo pavaros. Dauguma pramoninių robotų yra nuoseklūs. Paralelinių (lygiagrečių) robotų (3 pav.) (*parallel robot*) vykdymo įtaisas jungiasi prie pagrindo per kelias tarpusavyje sujungtas grandines, kitaip tariant, paralelinis robotas turi bent dvi „kojas“ („rankas“). Dauguma jo sujungimo vietų nėra tiesiogiai valdomos, ir daugelis tokių sąnarų turi po kelis laisvės laipsnius. Du populiariausi lygiagrečių robotų tipai yra „judesio platformos“ – naudojamos daugumoje simulatorių ir delta – robotai, naudojami greitam smulkių detalių dėliojimui. [17]



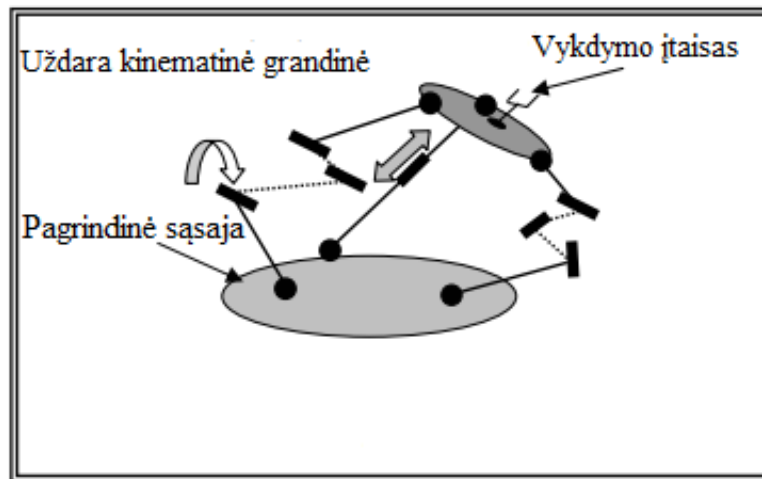
1 pav. „Mitsubishi“ nuoseklusis robotas



2 pav. Nuosekliojo roboto kinematinė schema



3 pav. „Mitsubishi“ paralelinis robotas



4 pav. Paralelinio roboto kinematinė schema

Paraleliniai robotai skirstomi į simetriškus (*symmetric*), plokštuminius (*planar*), sferinius (*spherical*), erdvinius (*spatial*). *Simetriški* manipuliatoriai turi galūnių skaičių, lygų laipsnio laisvės skaičiui, taip pat kilpų bendrą skaičių. *Plokštuminis* manipuliatorius yra suformuotas, kai dvi ar daugiau plokštuminės kinematinės grandinės veikia kartu bendroje platformoje. *Sferiniai* manipuliatoriai tik sugeba atkartoti galinio vykdymo įtaiso judėjimą pagal kontroliuojamus sferinius judesius. [7, 12]

1 Lentelė

Paralelinių robotų ir nuosekliųjų robotų palyginimas [7, 12]

Savybė	Paralelinis robotas	Nuoseklusis robotas
Kinematinės grandinės tipas	Uždaroji kinematinė grandinė	Atviroji kinematinė grandinė

1 lentelės tęsinys kitame puslapyje

Judėjimo aprašymas (natural description)	Dekarto erdvė (In Vartesian space)	Jungtinėje erdvėje (In joint space)
Pavarų vietos (location of actuators)	Šalia nejudančio pagrindo (Near the immobile base)	Ant sąsajų (on the links)
Inercijos jėgos ir standumas (inertia forces and stiffness)	Mažiau ir aukštai atitinkamai	Aukštai ir mažiau atitinkamai
Dizaino tvarkymas (design considerations)	Struktūra, darbo srities apsvartymas, ypatumai, ryšio trukdžiai (structure, workspace, considerations, singularities, link interference)	Stiprumo ir standumo apsvartymas, vibracijos savybės (Strength and stiffness considerations, vibration characteristics)
Pateiktos savybės (preferre property)	Kietumas (Stiffness)	Vikrumas (Dexterity)
Tiesioginė kinematika (use of direct kinematics)	Sunku ir sudėtinga (Difficult and complex)	Paprastas ir unikalus (straightforward and unique)
Atvirkštinė kinematika (use of inverse kinematics)	Paprastas ir unikalus (straightforward and unique)	Sudėtingas (Complicated)
Ypatingumas (singularity)	Statiškas (Static)	Kinematinė (Kinematic)
Tiesioginė jėgos transformacija (direct force transformation)	Gerai apibrėžtas ir unikalus (well defined and unique)	Nėra aiškiai apibrėžta, gali būti neegzistuojantis, unikalus ar begalinis (Not well defined, may be non-existent, unique or infinite)
Pageidautinas taikymas (preferred application)	Tikslus padėjimas (Precise positioning)	Platus judėjimas (gross motion)

Paralelinių robotų privalumai:

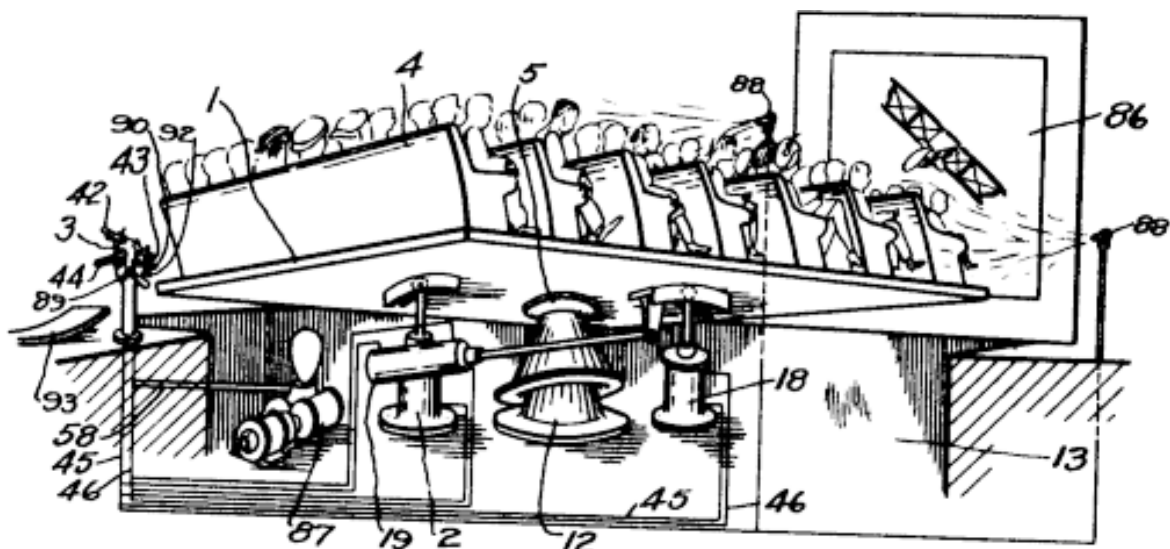
1. didelis greitis, palyginti su linijinio tipo robotais;
2. aukštas tikslumas;
3. standumas;
4. masė
5. judesių valdymui galima naudoti įvairių tipų pavaras, pvz. trosus.

Paralelinių robotų trūkumai:

1. kuo daugiau laisvės laipsnių, tuo sudėtingesnė konstrukcija;
2. reikalingi aukštos kvalifikacijos specialistai jiems eksploatuoti;
3. mažesnė darbinė zona.

1.2. Paralelinių robotų vystymosi istorija

Vienas iš pirmųjų judančią platformą (paralelinį robotą) suprojektavo James E. Gwinnett 1928 m. Paralelinis robotas turėjo trijų laisvės laipsnių sferiškai judančią platformą, kuri skirta naudoti kino teatruose, kur vaizdas ir garsas buvo naujiena (5 pav.). [15]



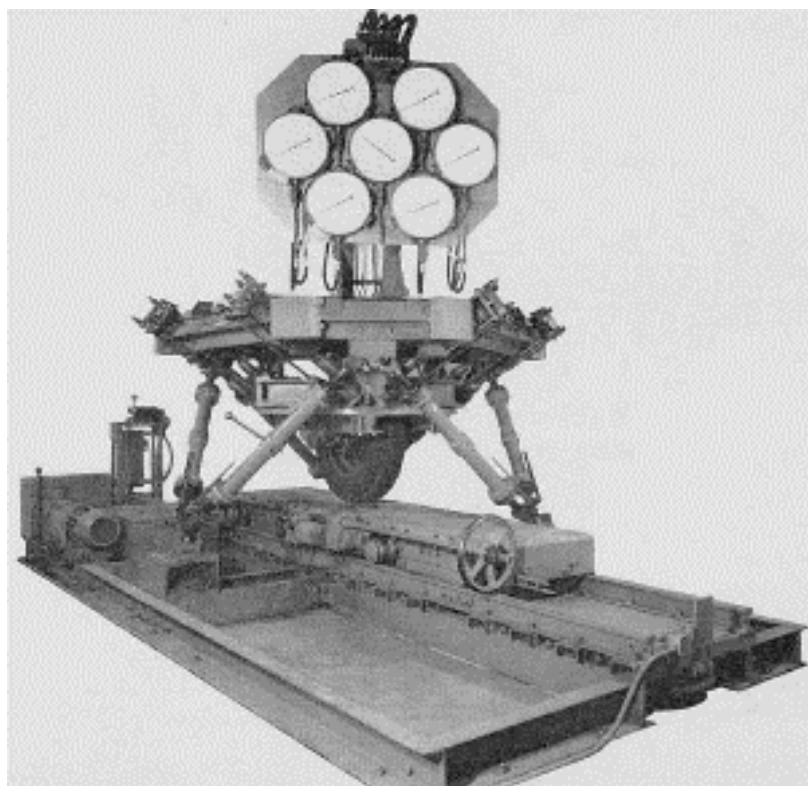
5 pav. Pirmasis paralelinis robotas

1934 m. Willard Pollard Jr. pateikė patentuoti tapybos purkštuvu mašiną. Trijų kojų robotas turėjo penkis laisvės laipsnius: trys kojos skirtos dažymo galvutės padėčiai orientuoti, o likusios dvi roboto orientacijai keisti.

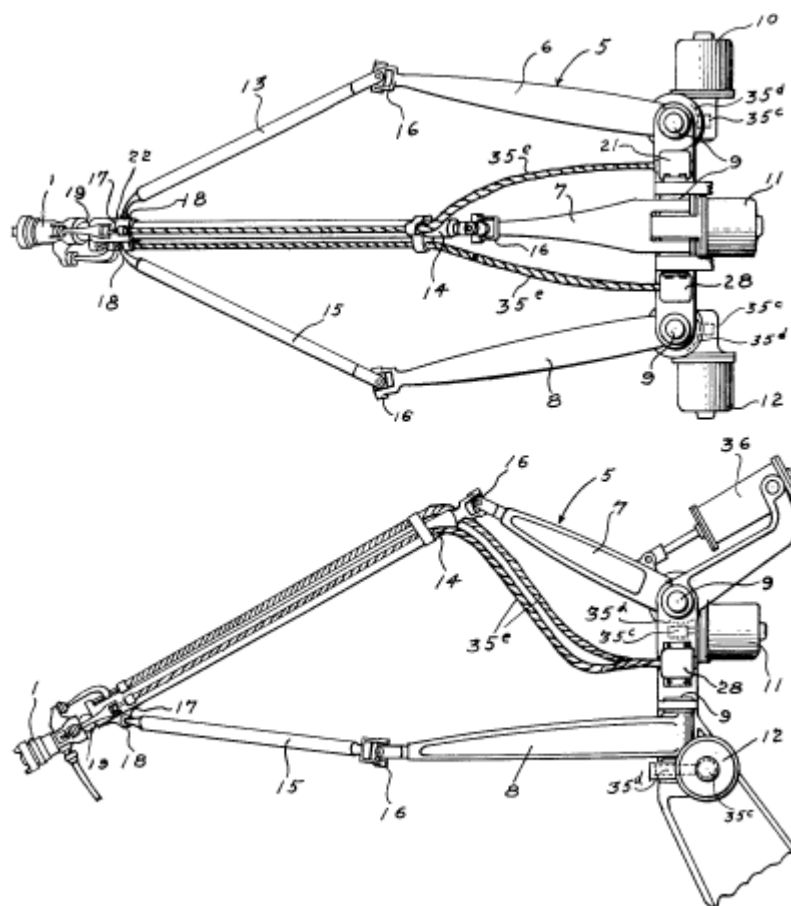
Pollard Jr. patentas išleistas 1942 m. birželio 16 d., o 1937 m. licenciją roboto gamybai suteikė DeVilbiss kompanijai. Siekdamas tapti pirmuoju robotų tiekėju 1941 m. DeVilbiss užbaigė pirmąjį prototipą pagal Harold Roselund. Roselund purkštuvu tapybos tipo robotas patentuotas 1944 m., ir tai nebuvo lygiagretusis robotas, o tik panaudota Pollard Jr. Pasiūlyta prototipo kontrolės sistema (7 pav.). [16]

1947 m. daktaro, automobilių inžinieriaus Eric Gough sukurtas naujas lygiagretusis robotas, kuris tuo metu buvo itin paplitęs, pakeitė pramonę. Robotas pavadintas „universalia padangų bandymo mašina“ (6 pav.). Robotas išrastas, atkreipiant dėmesį į lėktuvų nusileidimo problemą. Roboto pagrindinė paskirtis buvo testuoti padangų savybes, o roboto judesio platformos pagrindas šiam tikslui nebuvo kuriamas iš naujo. Tuo metu daktaras Gough atkreipė dėmesį į sistemas su trimis vertikaliomis ir trimis horizontaliais cilindrais, kurie buvo paplitę ir dažnai naudojami. Šios sistemos sudarytos iš šešių vienodų hidraulinių cilindų, jungiančių įtvirtintą pagrindą prie judančios platformos, prie kurios būdavo pritvirtinama padanga. Šie robotai tapo itin populiarūs, nes domkrato reguliavimas buvo paprastas ir jautrus mažiems svytavimams. Nauja funkcija, kurią

įdiegė Gough platformoje, buvo simetriškas išdėstymas iš šešių kojų. Roboto kojų ištiesimas reguliuojamas rankomis. Mašina pastatyta 1950 m. pradžioje, pradėta naudoti 1954 m. pradžioje.
[17]

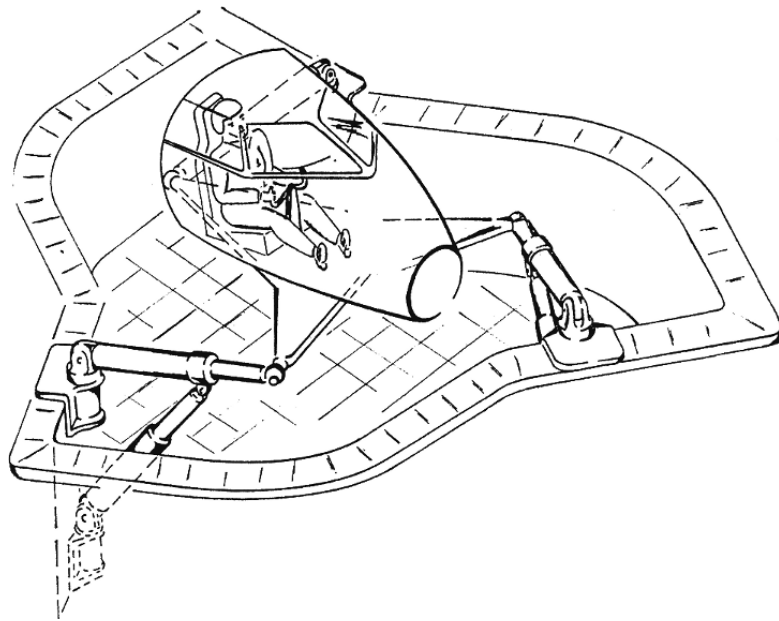


6 pav. Eric Gough „universali padangų bandymo mašina“



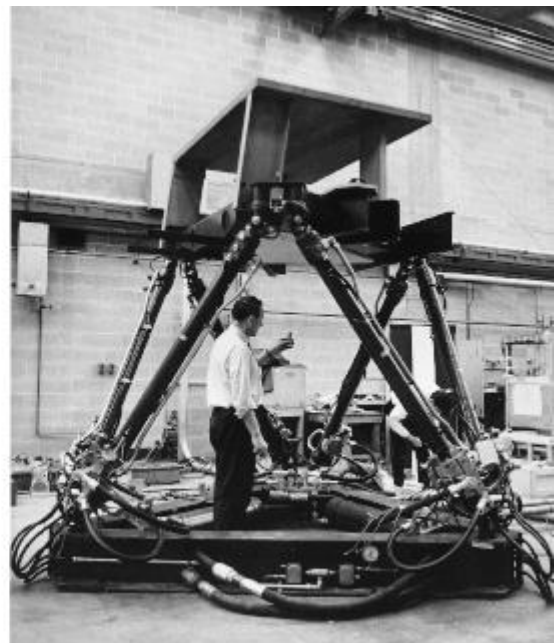
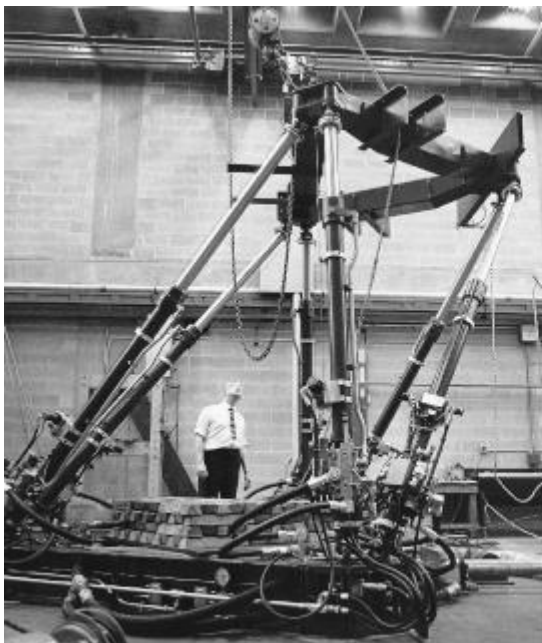
7 pav. Harold Roselund sukurtas purkštukas

1965 m. inžinierius mechanikas D. Stewart, nežinodamas apie Gough ir Cappel darbus, pasiūlė mechanizmą su šešiomis teleskopinėmis kojomis, skirtomis naudoti kaip skrydžio simulatorius (8 pav.). Stewart sukurta platforma buvo lygiagretusis mechanizmas. Jo idėjos apie simulatorių sukėlė nemažai diskusijų tarp mokslininkų, tarp kurių buvo ir Gough, kuris priminė apie savo padangų testavimo aparato veikimą. Nors Stewart nebuvo aštuonsienio šešiakojų išradėjas nei šiandieninių šešių kojų skrydžio simulatorių pirmtakas, bet neabejotina, kad jo brėžiniai turėjo didelę įtaką tolimesniam lygiagrečių robotų vystymuisi šioje sferoje. [17]



8 pav. Stewart platforma

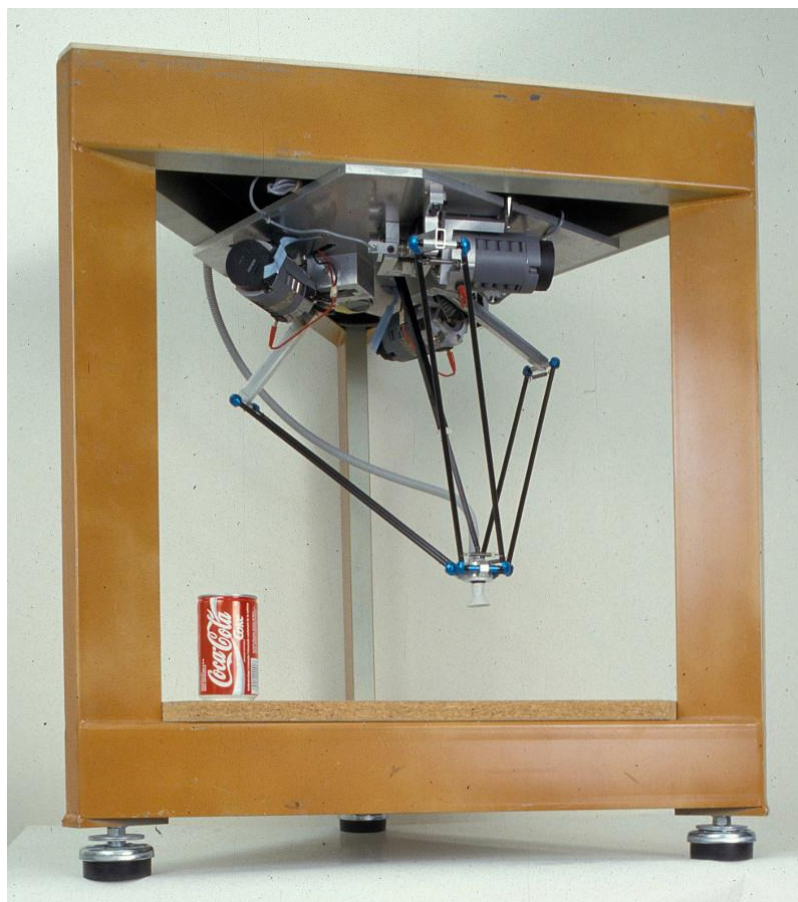
1971 m. Amerikos patento ir prekyženklis įstaiga suteikė patentą Klaus Cappel, jo išradimui ir naudojimui kaip judėjimo simulatoriaus. Patentas registruotas 1964 m. gruodžio 7 d., iš tikrųjų prašymas išduoti patentą ir idėja apie judėjimo simulatorių kilo atsižvelgiant į prie korporacinės įstaigos Sikorskylėktuvo skyriaus suvienytų technologijų projekto ir 6-DOF sraigtasparnio skrydžio simulatoriaus statybos prašymą. Todėl tokiu būdu padarytas pirmasis skrydžio simulatorius pagrįstas judesio platforma (9 pav.). [16, 17]



9 pav. Klaus Cappel judėjimo simulatorius

Beveik du dešimtmečius lygiagrečiaisiais robotais mažai kas domėjosi ir kreipė dėmesį, bet 1980 m. pradžioje jų populiarumas netikėtai pradėjo didėti. 1985 m. vasarą profesorius Reymond Clavel pasiūlė mintį panaudoti lygiagretinius, kad pastatytų 4-DOF lygiagretų manipuliatorių,

didelio greičio pasirinkimo ir vietos pasikeitimo operacijoms (10 pav.). Jis kreipėsi dėl serijos patentų, kad kelios kompanijos galėtų gaminti jo išradimą. Per pagrindinių patentų gyvenimo trukmę buvo pagaminta daugiau kaip 10 000 vienetų. Šiandien Delta roboto projektas nėra apsaugotas, ir daugybė kompanijų siūlo savo pasirinkimo ir vietos (pick-and-place) robotų versijas. Pvz., Fanuc, Motoman, Kawasaki. [17]



9 pav. Delta pasirinkimo ir vietos robotas

1.3. Paraleliniai robotai ir jų taikymo pavyzdžiai

Paraleliniai plokštuminiai robotai su uždara kinematine grandimi gali būti naudojami pozicionavimo sistemose, kai darbinio taško judesio trajektorija tarp dviejų galinių taškų nėra svarbi. Toks robotų valdymo būdas būdingas paletavimo robotams ar robotams, kurie atlieka objektų surinkimą plokštumoje (spausdinto montažo elektroninių komponentų talpinimas litavimo vietoje). Čia pagrindinis dėmesys į pozicionavimo tikslumą, o kiekvienos grandies judesio greitis dažniausiai nėra kaip nors sinchronizuojamas. Kita tokių robotų pritaikymo sritis yra įrankio gabenimas reikiama trajektorija. Pavyzdžiui, lazerinis plokščių objektų pjaustymas. Čia roboto valdymo užduotis yra sudėtingesnė, kadangi abiejų pavarų judesio greitis tarpusavyje turi būti sinchronizuotas tam, kad būtų gauta minimali judesio kontūro paklaida. Tokių robotų pavaros papildomai yra valdomos pagal judesio greitį, sinchronizuojant pavaras tarpusavyje, ir yra ženkliai

sudėtingesnės ir brangesnės. Šiais laikais paraleliniai robotai sutinkami ir naudojami daugelyje veiklos sričių, kur reikalingos aukštos tikslumo staklės: automobilių gamyboje, maisto pramonėje (11 pav.). [3]



11 pav. Demarex linijos įdiegimas rištainių pakavimui kepyklos pramonėje 1996 m.

Robotai sėkmingai naudojami medicinoje, atliekant vidaus organų operacijas (12 pav.). [3]



12 pav. „SurgiScope“ paralelinis robotas naudojamas chirurginėje robotų technikos laboratorijoje, Humboldt universitete Berlyne

Pasirinkimo ir vietos paraleliniai robotai naudojami dažniausiai automatuose, pvz., žaidimų (13 pav.). [3]



13 pav. Hitachi „Seiki“ robotai pasirinkimui ir vietai (pick-and-place)

Dviejų laisvės laipsnių paraleliniai plokštuminiai robotai dažniausiai naudojami moksliniams tyrimams, kaip mokomieji, įvairioms operacijoms simuliuoti bei robotams kalibruoti, kad padidintų galinio įtaiso padėties nustatymo tikslumą (14 pav.).



14 pav. Moksliniams tyrimams naudojami DexTAR paraleliniai robotai

1.4. Paralelinių robotų valdymas

Dextar paralelinių robotų valdymo programą parašytą JavaScript kalba. [5]

Dauguma paralelinių robotų valdymas yra realizuojamas Arduino pagalba. Atsižvelgiant į a tašką x, y, z erdvėje, realiu laiku Arduino apskaičiuoja servo varikliams kampus, kurie leidžia pasiekti tą tašką erdvėje. [4]

UFaactory įsikūrusi Shenzhen Kinija ir Kickstarted uArm sukūrė stalinį 4 ašių paralelinį mechanizmą manipuliatorių valdomą Arduino (15 pav.). Kontroluojama Arduino ranka yra labai stabili dėl lygiagretaus mechanizmo, sukoncentruojančią didžiąją dalį masės ties pagrindu. Ši projekto detalė reiškia, kad viršutinė ranka gali veikti iš tikrųjų greitai (15 pav.). [8]



15 pav. UFaactory sukurtas stalinis 4 ašių paralelinis mechanizmas

Fuzzy reguliatorius šiuo atveju turės dvi įvestis, su klaida e tarp kampo atskaitos kiekvienai pavarai, tikrajai vertei sistemoje ir kintamo dažnio klaidos Δe . Manipuliatoriui bus įtampos diapazonas $-9 \dots 9$ V. Fuzzy reguliatoriaus projektavime yra trys žingsniai: fuzzification, išvada (*inference*) ir defuzzification. Fuzzification yra sąsaja, gaminanti neaišku poaibį iš matavimo. Defuzzification yra sąsaja, kuri gamina miglotą ryškių kontūrų išvestį iš sąsajos rezultatų. Rezultatai iš įvesties yra apdorojami naudojant taisyklių rinkinį, šis procesas yra vadinamas išvada. Fuzzy reguliatoriaus rezultatus palyginus su PID, pastebėta, kad Fuzzy reguliatoriaus atsakas (*responded*) greitesnis ir tikslesnis negu PID reguliatoriaus. [6]

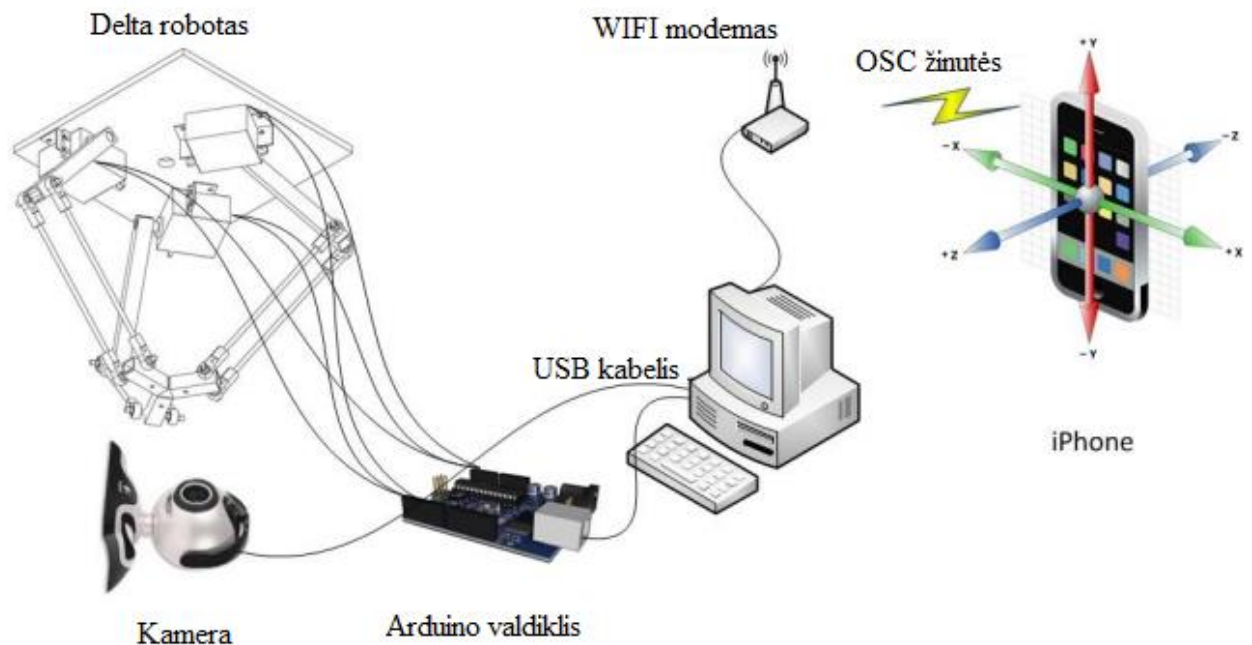
Fuzzy logika, skirta įgyvendinti panašią į žmogaus mąstymą kontrolės sistemą. Fuzzy reguliatorius gali būti suprojektuotas, kad pamėgdžiotų žmogaus dedukcinį gebėjimą kontrolės sistemoje. Fuzzy kontrolė įtraukia dviprasmišką žmogaus logiką į kompiuterio programas. Tai tinka kontrolės problemoms, kurioms negali lengvai atstovauti matematiniai modeliai dėl jų neatskiriamų tikslumų: parametro pakitimas, neprieinami ar neužbaigti duomenys, labai sudėtingi procesai. Fuzzy reguliatoriai tapo populiarūs pastaraisiais metais, nes jiems nebūtinai reikalingas teorinis modelis, kuris turi būti kontroliuojamas. [2]

Siekiant išvystyti Fuzzy reguliatorių, kiekvienas turi iš pradžių turėti šalia ekspertą, kurio patirtimi vadovaujantis, reikia nustatyti žemėlapių sudarymą nuo įrangos su fuzzy priemonėmis, su kuriomis eksperto žinios yra nustatytos kiekybiškai. PID reguliatorius dirba gerai linijiniame procese, bet to nepakankama, kad prieiti prie netiesinio proceso. [2]

Padėties kontrolės ir trajektorijos sekimas yra bendriausios robotų manipuliatorių užduotys. Atsižvelgiant į pageidaujama trajektorija, jungtinė jėga yra pasirinkta (the joint force is chosen), kad manipuliatorius ją sektu. PID valdiklis yra vienos įvesties/ vienos išvesties (SISO) valdiklis, gaminantis (produces) kontrolės signalą, kuris yra trijų terminų suma (sum of three terms). Pirmą trukmę (first term) yra proporcinga (P) padėjimo klaidai, antra trukmė yra proporcinga integralo (I) klaidai ir trečias yra proporcingas išvestinio žodžio (D) klaidai. PID tipo reguliatoriai naudojami pramoniniuose robotuose, nes tai lengva įgyvendinti ir reikalingas mažas laiko tarpas per operatyvią operaciją. Klaida tarp tikros ir pageidaujamos jungtinės padėties yra panaudota, kaip grįžtamasis ryšys, siekiant kontroliuoti pavarą, jungiančią su kiekviena sujungimo vieta. [1]

Yra du tradiciniai dinamiški reguliatoriai lygiagrečiai manipuliatoriams: APD (augmented PD) reguliatorius ir CT (computed-torque) reguliatorius. Abiejų reguliatorių bendras bruožas šalinti sekimo klaidą prie linijinės PD kontrolės. Tačiau, linijinė PD kontrolė nėra tvirta prieš abejojančius faktorius tokius, kaip modeliavimo klaidos ir išorinius sutrikdymus. Kad įveiktų šias problemas, NPD (nonlinear PD) kontrolė gali būti sukombinuota su tradicinėmis kontrolės strategijomis pagerinančiomis kontrolės tikslumo ir sutrikdymo atmetimo galimybes. Akivaizdus tradicinių CT reguliatorių trūkumas yra sekimo klaidos pašalinimas prie linijinio PD algoritmo. Tačiau, linijinis PD algoritmas nėra tvirtas prieš abejojančius faktorius tokius, kaip modeliavimo klaidos ir netiesinės trinties. Kad įveiktų šią problemą, NPD algoritmas gali būti sukombinuotas su tradicinėmis kontrolės strategijomis siekiant pagerinti kontrolės tikslumą. NCT reguliatorius vystėsi šiame linijiniame PD CT reguliatoriaus su NPD algoritmu. [11]

Panaudojant Arduino valdiklį, kuris valdomas per USB. Tokių būdų galima užprogramuoti Arduino, tam jog skaitytų OSC nuoseklias žinutes, bei užprogramuoti iPhone interfeisą skaityti akcelerometro duomenis ir siųsti juos į serverio paraišką per OSC žinutes. Kai serveris gauna juos, jis apskaičiuoja kinematika ir siunčia kampų vertes į Arduino. Kampai panaudojami variklių valdymui. Signalas, gautas iš iPhone akcelerometro, turi būti apdirbtas, kadangi nėra jokio tiesioginio pakeitimo tarp pagreičio ir padėties. Norint signalą pritaikyti naudojamas dvigubas integralas (16 pav.). [9]



16 pav. Sistemos architektūra

Genetinių algoritmų optimizavimo metodo populiarumas auga pastaraisiais metais, nes vis daugiau mokslininkų atrado savo pritaikomumo naudą. Genetinis algoritmas buvo išrastas John Holland 1960 metais. Genetinis algoritmas yra, bet koks gyventojais pagrįstas modelis, kuris naudoja pasirinkimą ir kombinacijas iš naujo operatoriaus, kad sukurtų naujus tipinius punktus paieškos erdvėje. Daug genetinių algoritmo modelių buvo įvesti tyrėjų, didžia dalimi dirbusių nuo bandomosios perspektyvos, kurie domėjosi genetiniais algoritmais, kaip optimizavimo įrankiais.

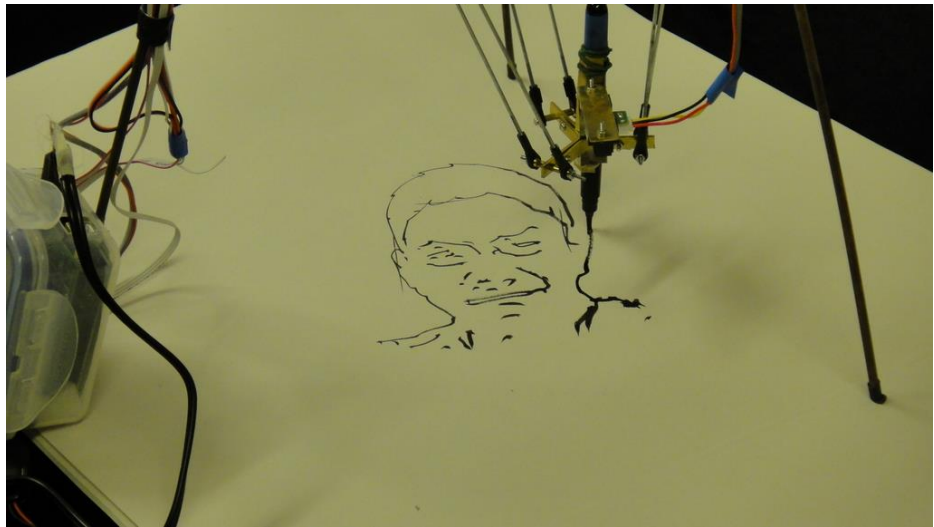
Kai kurie iš genetinių algoritmų pranašumu yra jų gebėjimai:

- optimizuoti su nuolatinais ar atskirais kintamaisiais;
- nereikalingos išvestinės informacijos;
- spręsti su dideliu kiekiu kintamųjų;
- tinka lygiagrečiai sujungtiems kompiuteriams;
- gali koduoti kintamuosius, kad optimizavimas būtų padarytas su užkoduotais kintamaisiais.

[10]

1.5. Piešiantys paraleliniai robotai

Portretus piešiantis robotas Sketchy valdomas Arduino programos (17 pav.). [13]



17 pav. Sketchy piešiantis robotas

Garsus dizaineris Robert'as Howsare'as sukūrė prietaisą, kuris iš dviejų grotuvų plokštelių gebą piešti. Šiuo prietaisu, galima piešti sudėtingas geometrines figūras, pritvirtinus prie plokštelių dvi tarpusavyje sujungtas medines lenteles ir įstačius pieštuką, deja tuo pačiu momentu muzikos klausytis nepavyks (18 pav.). [14]



18 pav. Dizainerio Robert Howsare sukurtas robotas

2. PROJEKTINĖ DALIS

2.1. Paralelinio roboto konstrukcija

Paralelinių plokštuminių robotų mechanikos paprastumas ir pigumas, atveriantis jiems plačias galimybes taikyme, reikalauja tokių pačių savybių ir valdymo sistemoms. Vienas iš valdymo sistemų variantų yra gana pigių įterptinių kompiuterių panaudojimas. Tokio tipo kompiuterių skaičiavimo greitis, operatyviosios atminties apimtis, aparatūrinių pertraukimų galimybės yra ribotos, todėl valdymo algoritmai turi būti supaprastinti tiek, kad užtektų resursų spręsti valdymo uždavinius. Plokštuminių robotų pavarose, kaip taisyklė, naudojami optiniai impulsiniai enkoderiai, kurių skiriamosios gebos pakanka. Tačiau įterptiniai kompiuteriai, dirbantys su enkoderiais, turi naudoti aparatūrinius pertraukimus. Tai susiaurina pasirinkimo galimybes. Antra optinių impulsinių enkoderių panaudojimo problema ta, kad enkoderiai tiesiogiai jungiami su variklio velenu. Variklio sukimo momentas į kinematinę jungtį perduodamas per reduktorių, todėl pilnas kinematinės grandies apsisukimas atitinka kelis šimtus variklio veleno, ir tuo pačiu enkoderio, apsisukimų. Išjungus robotui maitinimą yra „užmirštama“ kuriame apsisukime buvo enkoderis. Pigiose plokščiųjų robotų valdymo sistemose diegti autonominių enkoderių ir jų impulsų skaitiklius, dirbančių su pertraukimais, maitinimo sistema yra per brangu ir sudėtinga. Todėl, kiekvieną kartą prieš pradėdant darbą plokščiasis, robotas turi atlikti kalibravimą – pasukti pavarų velenus į iš anksto žinomas pozicijas ir priimti, kad tai yra nulinis enkoderio apsisukimas.

Paralelinio roboto konstrukcijoje naudojamas nuolatinės srovės variklis su reduktoriumi:



19 pav. Nuolatinės srovės variklis

2 lentelė

Nuolatinės srovės variklių su reduktoriumi micromotors E192.12.125 duomenys [14]

Variklis	Reduktoriaus perdavimo skaičius	Maksimalus momentas, Nm	Aps/min be apkrovos	Aps/min prie maksimalios apkrovos	Srovė be apkrovos, I	Srovė prie maksimalios apkrovos, I
micromotors E192.12.125	1:125	3	32	24	<0,4	1,32

Žingsniams skaičiuoti konstrukcijoje naudojami variklių posūkio kampo optiniai davikliai (20 pav.) RI58-D/1024ED.32IF, reikalinga įtampa 10...30 V, jungimo tipas (schema) Push-pull.

3 lentelė

RI58-D/1024ED.32IF Push-pull jungimo schema

Spalva (TPE)	Push-pull
Ruda	Kanalas A
Pilka	Kanalas B
Raudona	Kanalas N
Rudas/žalias	5/10...30 V DC
Balta/žalias	Žemė (GND)



20 pav. Variklių posūkio kampo optiniai davikliai

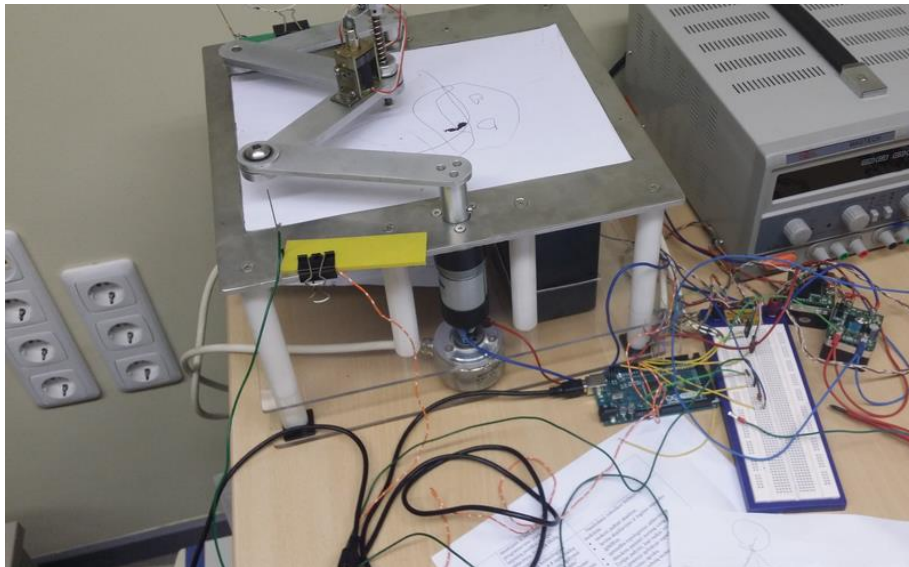
4 lentelė

Variklių posūkio kampo optinio daviklio RI58-D/1024ED.32IF duomenys

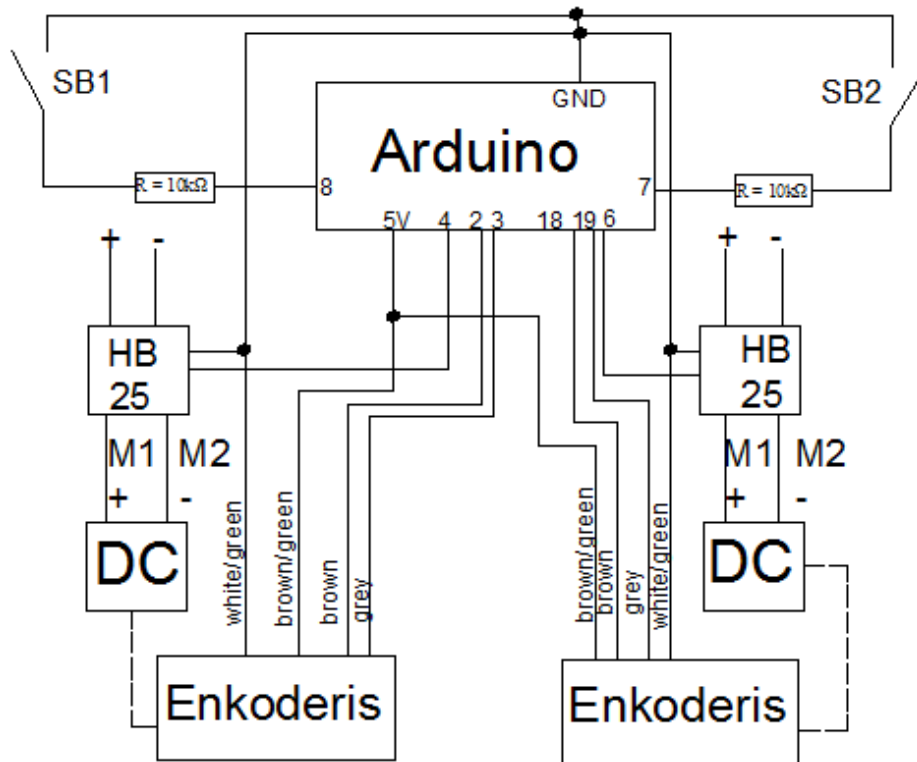
Impulsų skaičius:	1/2/3/4/5/10...3600/4000/4096/5000
Matmenys:	Ø58,8mm, ašis - Ø10/12/14mm
Maksimalus greitis:	6000 min ⁻¹
Apsaugos klasė:	IP64 korpusui, IP64 guoliui
Aplinkos temperatūra:	nuo -10° iki 70°C
Korpusas:	aliuminis
Maitinimo įtampa:	5V dc ±10% arba 10...30V dc
Maksimali srovė:	40mA (5V dc), 60mA (10V dc), 30mA (24V dc)

2.2. Paralelinio roboto nustatoma nulinė pozicija

Išanalizavus informaciją apie paralelinius robotus, sujungta elektrinė roboto valdymo schema ir susipažinta su roboto valdymu. Eksperimentas buvo vykdomas išmokstant judinti, valdyti vieną pavarą ir užkomentavus kodą pirmosios pavaros buvo tie patys veiksmai atlikti su antrąja pavarą. Sujungus ir suderinus kodo gabaliukus buvo pradėtos judinti abi pavaros kartu, prijungti galiniai išjungikliai, kad būtų galima atlikti kalibravimą (nustatyti nulinę padėtį).



21 pav. Eksperimentinis stalas su galiniais išjungikliais

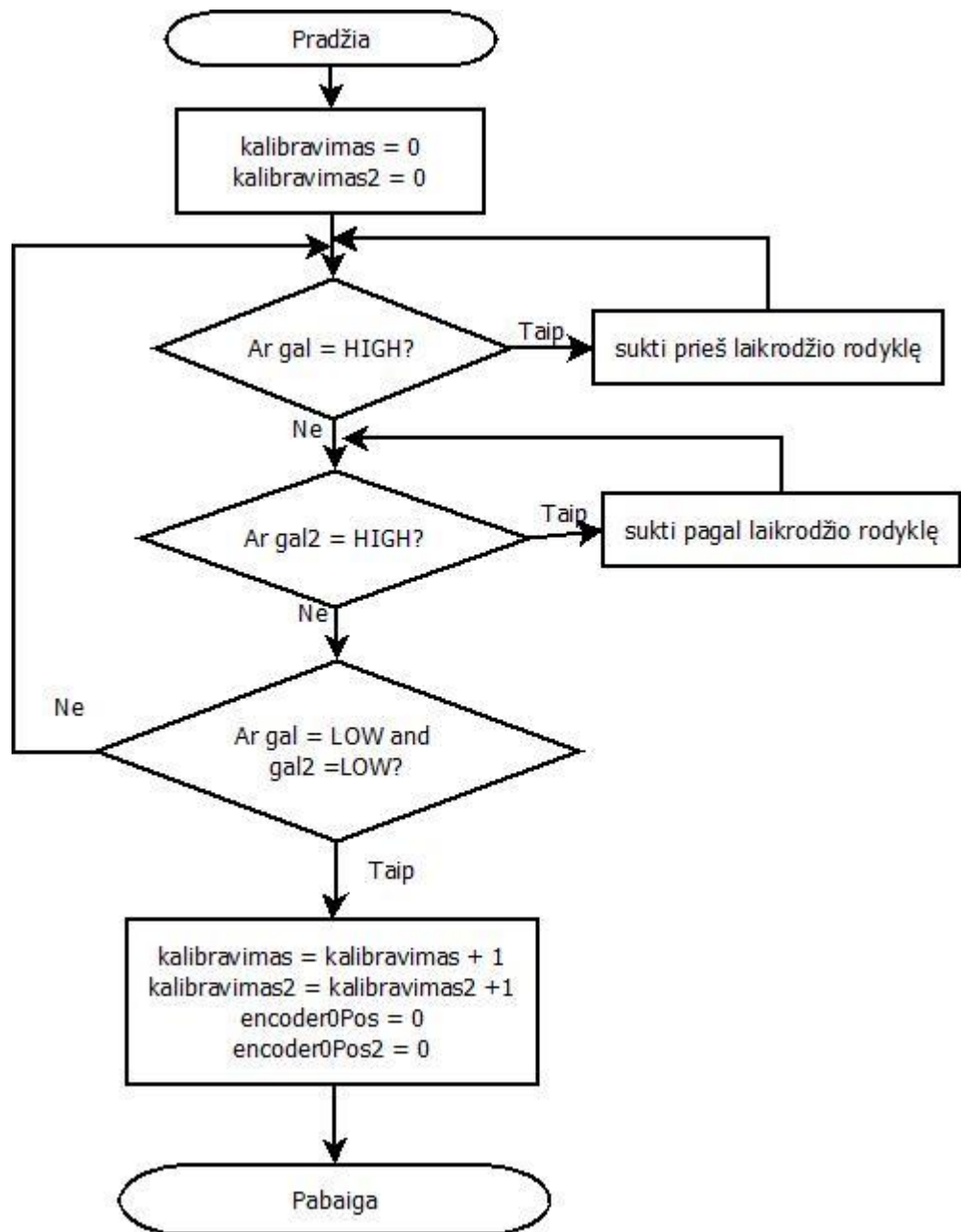


22 pav. Elektrinė roboto valdymo schema

Roboto valdymo schema sudaro (22 pav.) įterptinis Arduino genties kompiuteris, du platuminės impulsinės moduliacijos galios keitikliai HB25, komutuojantys srovę iki 25A, du nuolatinės srovės varikliai su reduktoriais (redukcijos koeficientas 125), du posūkio kampo enkoderiai, generuojantys 1000 impulsų viename apsisukime. Yra galimybė dirbti du dvigubu arba keturgubu enkoderio tikslumu. Schemoje yra SB1 ir SB2 jutikliai, skirti užfiksuoti roboto jungčių padėtis kalibravimo pozicijose. Numatyta, kad robotas bus judinamas pasinaudojant koordinatėmis.

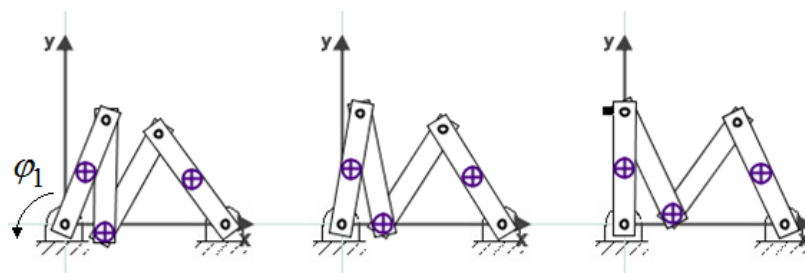
Pirmas veiksmas atliekamas prieš roboto judinimą nustatomas pradžios taškas t.y. abi roboto „rankos“ sukalibruojamos, nustatomos variklio posūkio kampo daviklių pozicijos nulinės ir nuo pradžios taško matuojami kampai. Kalibravimui nustatyti sumontuoti galiniai išjungikliai (21 pav.), kurių pagalba bus nustatoma pradinė padėtis. Kalibravimo rezultatai galios iki tol, kol bus neišjungtas maitinimas roboto valdymo sistemai. Kalibravimo algoritmas priklauso nuo roboto kinematinės schemos ir pasirinkto tiesioginio ir atvirkštinio kinematikos uždavinio matematinio sprendimo būdo. Naudojami įterptiniai kompiuteriai turi savo maksimalią greitaveiką. Valdymo sistema turi gebėti dirbti realiame laike, todėl labai svarbu tiesioginio ir atvirkštinio kinematikos uždavinio sprendime naudoti kuo mažiau matematinių ir loginių veiksmų. Daugeliu atveju nėra tinkami bendrieji sprendimai, naudojantys apibendrintas transformacijos matricas – programos kodas tampa per ilgas ir dėl perteklinio kiekio komandų neįmanoma realizuoti valdymą realiame laike.

Algoritmas pagal kurį vykdomas kalibravimas:



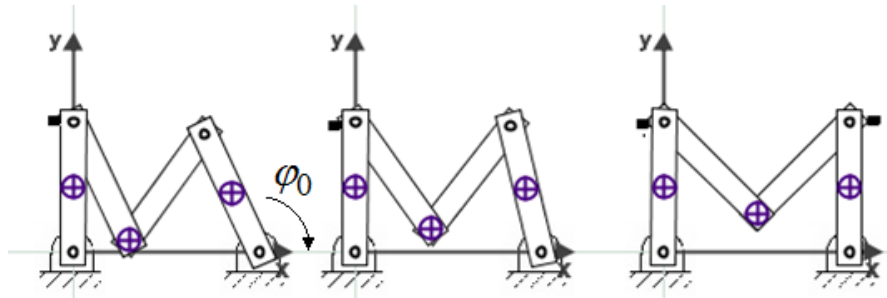
23 pav. Kalibravimo algoritmas

Algoritmas nurodo, kokia tvarka bus atliekamas kalibravimas. Kalibravimas atliekamas dviem etapais. Pirmiausiai keičiamas kampas φ_1 taip, kad jungties L_1 ašis (28 pav.) sutaptų su Y ašies kryptimi (24 pav.). Sutapimo pozicijoje yra nuspaudžiamas SB1 jutiklio kontaktas.



24 pav. Pirmosios pavaros kalibravimo etapai

Antrajame etape keičiamas kampas φ_0 taip, kad jungties L_4 ašis (28 pav.) sutaptų su Y ašies kryptimi (25 pav.). Sutapimo pozicijoje yra nuspaudžiamas SB2 jutiklio kontaktas.

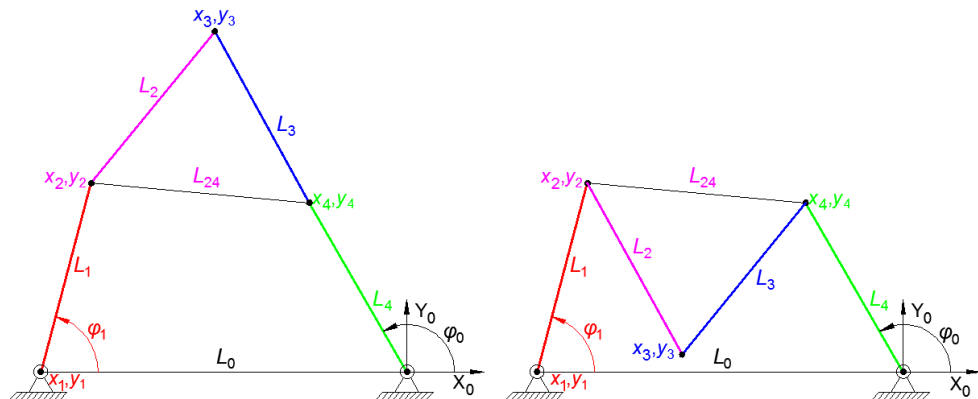


25 pav. Nulinės pavaros kalibravimo etapai

Atlikus kalibravimą roboto valdymo sistema nuosekliai skaito iš masyvo x_3, y_3 taškų koordinates, apskaičiuojami tas koordinates atitinkančius pavarų velenų φ_0 ir φ_1 posūkių kampus, kiekvienos pavaros sukimosi greičiai ir vykdomas pasisukimas į reikiamas pozicijas. Judesio trajektorijos nuokrypių priklausomybių tyrimas yra sekantis tyrimų etapas.

2.3. Tiesioginio kinematinio uždavinio sprendimas

Tiesioginis kinematikos uždavinys skirtas pagal turimus pavarų velenų posūkių kampus φ_0 ir φ_1 , apskaičiuoti atitinkamas darbinio taško koordinates x_3, y_3 . Tiesioginio uždavinio sprendimo kinematinė schema pateikta 26 pav.



26 pav. Kinematinė schema

Pirmosios pavaros veleno ašies koordinatės, kai nulinės pavaros veleno ašis sutapdinta su x_0, y_0 tašku:

$$\left. \begin{aligned} x_1 &= x_0 - L_0, \\ y_1 &= y_0. \end{aligned} \right\} \quad (1)$$

čia x_0, y_0 ir x_1, y_1 – pirmosios ir antrosios sąnaros, prie kurių prijungtos pavaros, koordinatės, L_0 – atstumas tarp variklių velenų ašių.

Šarnyro mazgo 4 koordinatės:

$$\left. \begin{aligned} x_4 &= x_0 + L_4 \cdot \cos(\varphi_0), \\ y_4 &= y_0 + L_4 \cdot \sin(\varphi_0). \end{aligned} \right\} \quad (2)$$

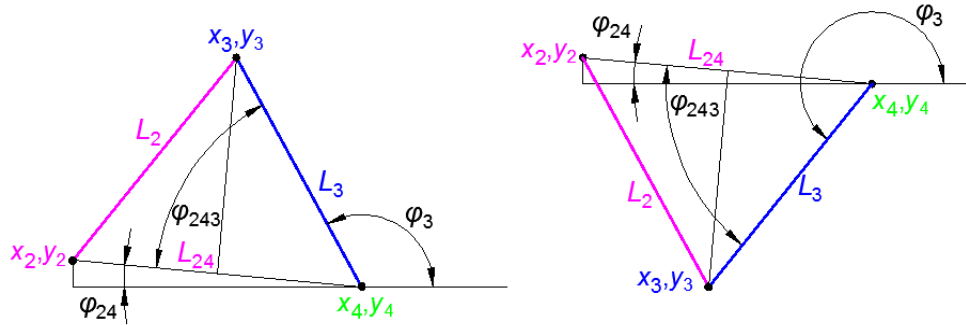
čia x_4, y_4 – pasyvaus šarnyro, jungiančio L_4 ir L_3 jungtis, koordinatės, L_4 – grandies ilgis, φ_0 – L_4 jungties ir tuo pačiu pavaros veleno posūkio kampas.

Šarnyro mazgo 2 koordinatės:

$$\left. \begin{aligned} x_2 &= x_1 + L_1 \cdot \cos(\varphi_1), \\ y_2 &= y_1 + L_1 \cdot \sin(\varphi_1). \end{aligned} \right\} \quad (3)$$

čia x_2, y_2 – pasyvaus šarnyro, jungiančio L_1 ir L_2 jungtis, koordinatės, L_1 – jungties ilgis, φ_1 – L_1 jungties ir tuo pačiu pavaros veleno posūkio kampas.

Jungtys L_2 ir L_3 , kartu su menama kraštine L_{24} sudaro lygiašonį trikampį (26 pav.).



27 pav. Kinematinės grandinės galo koordinatės

Menamos kraštinės ilgis:

$$L_{24} = \sqrt{(x_4 - x_2)^2 + (y_4 - y_2)^2}, \quad (4)$$

čia x_4, y_4 ir x_2, y_2 – ketvirtosios ir antrosios sąnaros, prie kurių prijungtos pavaros, koordinatės.

Kampas tarp L_3 jungties ir menamos kraštinės L_{24} :

$$\varphi_{243} = \arccos\left(\frac{L_{24}}{2 \cdot L_3}\right), \quad (5)$$

čia L_3 – grandinės ilgis, L_{24} – menamos kraštinės ilgis.

Kampas tarp menamos kraštinės L_{24} ir horizontalios X ašies

$$\varphi_{24} = \arctg\left(\frac{y_4 - y_2}{x_4 - x_2}\right), \quad (6)$$

čia x_4, y_4 ir x_2, y_2 – ketvirtosios ir antrosios sąnaros, prie kurių prijungtos pavaros, koordinatės.

Jungties L_3 posvyrio kampas X_0 ašies atžvilgiu

$$\varphi_3 = \pi - (\varphi_{243} + \varphi_{24}), \quad (7)$$

čia φ_{243} – kampas tarp L_3 jungties ir menamos kraštinės L_{24} , φ_{243} – kampas tarp menamos kraštinės L_{24} ir horizontalios X ašies.

Darbinio taško, arba šarnyro 3 koordinatės:

$$\left. \begin{aligned} x_3 &= x_4 + L_3 \cdot \cos(\varphi_3), \\ y_3 &= y_4 + L_3 \cdot \sin(\varphi_3). \end{aligned} \right\} \quad (8)$$

čia x_4, y_4 ketvirtosios sąnaros, prie kurių prijungtos pavaros, koordinatės, L_3 – grandinės ilgis.

Šio taško koordinatės x_3, y_3 yra tiesioginio uždavinio sprendinys

$$\left. \begin{aligned} x_3 &= f_x(\varphi_1, \varphi_0), \\ y_3 &= f_y(\varphi_1, \varphi_0). \end{aligned} \right\} \quad (9)$$

kai žinomi abiejų pavarų velenų posūkio kampai φ_1, φ_0 .

2.4. Atvirkštinio kinematinio uždavinio sprendimas

Atvirkštinis kinematikos uždavinys skirtas pagal turimas darbinio taško koordinates x_3, y_3 apskaičiuoti toms koordinatėms atitinkančius φ_0 ir φ_1 posūkio kampus. Atvirkštinio uždavinio sprendimo kinematinė schema pateikta 28 pav. Pirmiausiai nubrėžiamas pagalbinis vektorius L_{34} ilgio, kuris jungia pavaros veleno sukimosi centrą x_0, y_0 su darbinio tašku x_3, y_3 . Tokiu pat būdu sukuriama vektorius L_{12} ilgio, jungiantis kitos pavaros veleno sukimosi centrą x_1, y_1 su tuo pačiu x_3, y_3 darbinio tašku. Sprendžiant atvirkštinį uždavinį koordinatės x_3, y_3 visuomet yra žinomos, todėl pagalbinių vektorių L_{34} ir L_{12} parametrai randami taip:

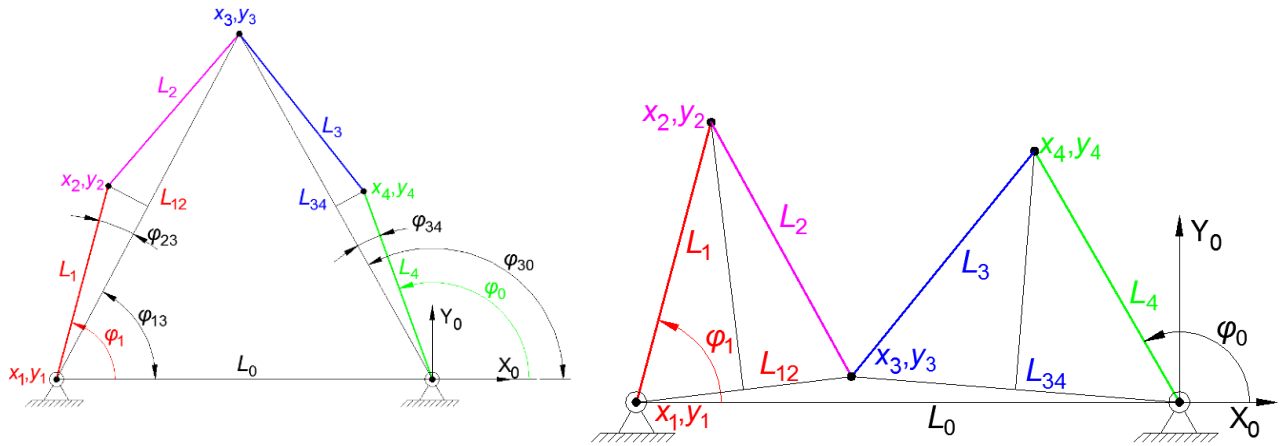
$$\left. \begin{aligned} L_{12} &= \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}, \\ L_{34} &= \sqrt{(x_3 - x_0)^2 + (y_3 - y_0)^2}, \\ \varphi_{13} &= \arctg\left(\frac{y_3 - y_1}{x_3 - x_1}\right), \\ \varphi_{30} &= \pi + \arctg\left(\frac{y_3 - y_0}{x_3 - x_0}\right). \end{aligned} \right\} \quad (10)$$

čia x_1, y_1 ir x_3, y_3 – pirmosios ir trečiosios sąnaros, prie kurių prijungtos pavaros, koordinatės, L_{12} ir L_{34} – menamų kraštinių ilgiai.

Kampas φ_{23} tarp vektoriaus L_{12} ir jungties L_1 randamas taip:

$$\varphi_{23} = \arccos\left(\frac{L_{12}}{2 \cdot L_1}\right), \quad (11)$$

čia L_1 – grandinės ilgis, L_{12} – menamos kraštinės ilgis.



28 pav. Atvirkštinio uždavinio sprendimo kinematinė schema

Tokiu pat būdu randamas kampas φ_{34} tarp vektoriaus L_{34} ir jungties L_4 randamas taip:

$$\varphi_{34} = \arccos\left(\frac{L_{34}}{2 \cdot L_4}\right), \quad (12)$$

čia L_4 – grandinės ilgis, L_{34} – menamos kraštinės ilgis.

Tada pavarų velenų posūkio kampai φ_0 ir φ_1 , kurie užtikrina kad darbinis taškas bus reikiamose x_3, y_3 koordinatėse, apskaičiuojami taip:

$$\left. \begin{aligned} \varphi_0 &= \varphi_{30} - \varphi_{34}, \\ \varphi_1 &= \varphi_{13} + \varphi_{23}. \end{aligned} \right\} \quad (13)$$

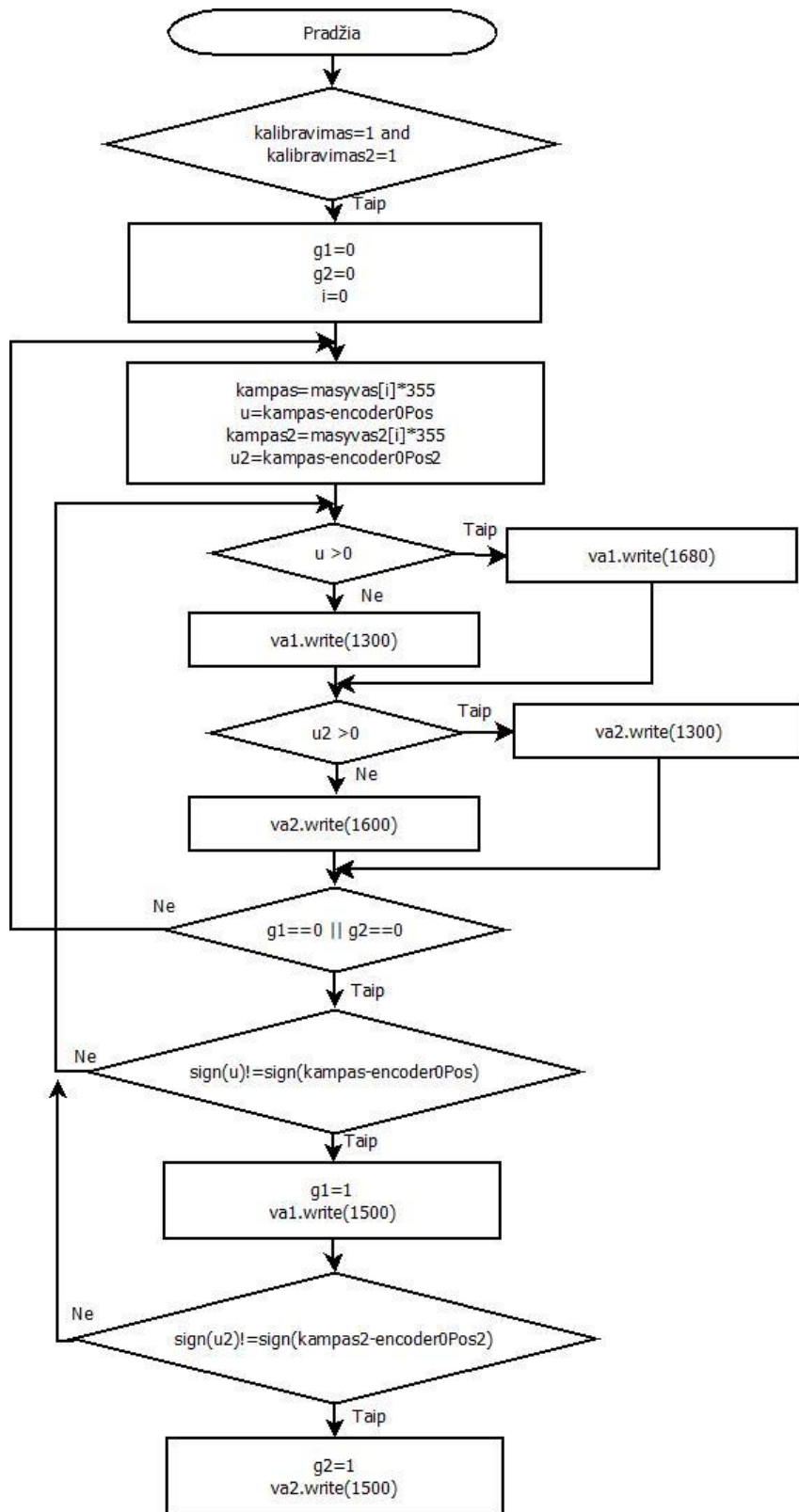
čia φ_{30} – kampas tarp menamos kraštinės L_{34} ir X ašies, φ_{34} – kampas tarp menamo kraštinės L_{34} ir grandinės L_4 , φ_{13} – kampas tarp menamos kraštinės L_{12} ir X ašies, φ_{23} – kampas tarp menamo kraštinės L_{34} ir grandinės L_1 .

Tyrimė naudotam robotui pateikiamas judesio trajektorijos darbinių taškų koordinatėms x_3, y_3 masyvas. Kiekvienam x_3, y_3 taškui panaudojant lygtis (10, 11, 12 ir 13) apskaičiuojami tas koordinatės atitinkantys pavarų velenų φ_0 ir φ_1 posūkio kampai.

2.5. Valdymo programa

Programa pradeda patikrinimu ar atliktas kalibravimas. Kalibravimą atlikus apskaičiuojami reikalingi kampai kuriais turi pasisukti „rankos“ ir enkoderio pozicija, kurioje sustojusios „rankos“. Pagal esama „rankų“ pozicija parenkama į kurią pusę turi sukstis „rankos“ ar į kairę ar į dešinę. Norint pasiekti užduotą tašką yra pasinaudojama „sign“ funkcija, kad palyginti esamą ir užduotą koordinatę. Rankoms pasiekus užduotą koordinatę, yra paaimamas naujas elementas iš masyvo ir atliekami veiksmai, kol iš masyvų yra paaimamos visos koordinatės.

Reikšmės yra apskaičiuojamos „Microsoft excel“ programa pagal kinematinės formules ir surašomos į masyvą, siekiant išvengti valdiklio apkrovimo papildomais skaičiavimais dėl kurių atsiranda uždelsimai. Imant reikšmes iš masyvo, nekreipiamas dėmesys į judėjimo trajektoriją tarp taškų, svarbiausia pasiekti užduotą tašką.



29 pav. Programos valdymo algoritmas

2.6. Problemos

Atliekant eksperimentą susidurta su problemomis:

1. Pradėjus valdyti vieną pavara, kita pavara pradeda judėti.

Problemos sprendimas: atsirandanti vibracija paleidus pirmąją pavara persiduoda antrosios pavaros enkoderiui, dėl ko pradeda judėti antroji pavara kuriai kodo judėti išskyrus pertraukimų paprogramės nėra parašyta.

2. Piešiant tą pačią liniją kelis kartus atsiranda „plaukimas“ į vieną pusę.

Priežastys: laisvumas mechanikoje, variklių nesuspėjimas sustoti.

Problemos sprendimas: sumažintas laisvumas mechanikoje.

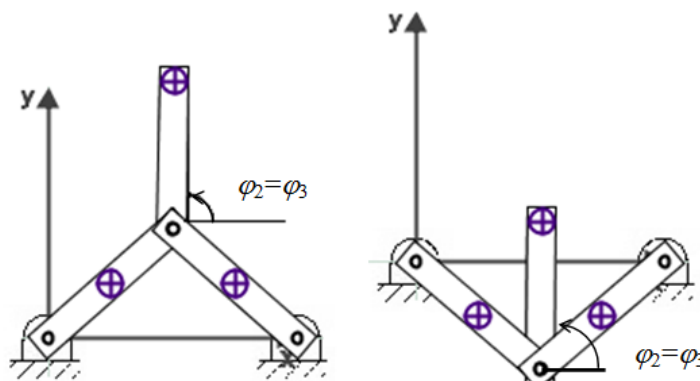
3. Enkoderio žingsnio skaičiavimo paklaidos.

Priežastis: variklius sustabdžius dėl inercijos variklis iškart nesustoja ir kartu su enkoderiu pasisuka keliais žingsniais.

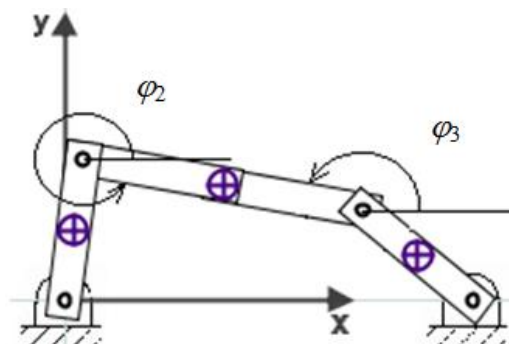
4. Atliekant užduotą operaciją paralelinio roboto tapimas neveiksniu.

Priežastis: Vykdamt operaciją „rankos“ patenka į neapibrėžties (singularity) zoną (30, 31 pav.) Sprendžiant Jakobio matricą ir gaunant determinantą lygu nuliui, tai tas taškas yra neapibrėžtis (singularity).

Problemos sprendimas: vienas iš problemos sprendimas galimas šiam robotui darbo zoną nustatyti vienoje x ašies pusėje.



30 pav. Neapibrėžties vietos, kai $\varphi_2 = \varphi_3$



31 pav. Neapibrėžties vietos, kai $\varphi_2 - \varphi_3 = \pi$

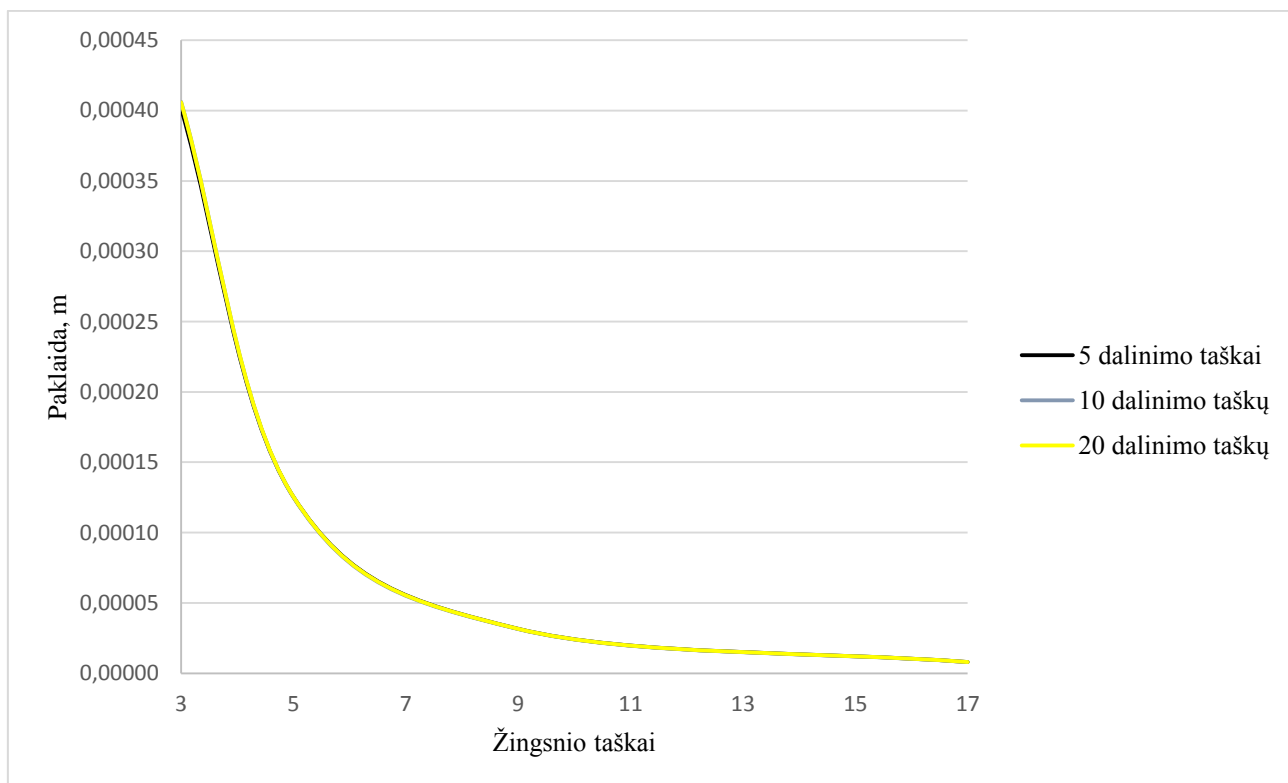
3. JUDESIO TRAJEKTORIJS TYRIMAS

Judesio trajektorijos nuokrypių priklausomybių tyrimas atliekamas, kai roboto pavarų greitis reguliuojamas ir nereguliuojamas. Tyrimo objektui dėl techninių galimybių negalima reguliuoti greičio, todėl bus atliekamas tyrimas modeliuojant.

Tyrimė braižoma tiesė 12 cm ilgio. Pagal turimas darbinių taškų koordinates x_3 , y_3 toms koordinatėms apskaičiuota atitinkantys φ_0 ir φ_1 posūkio kampai pasinaudojant atvirkštinio kinematikos uždavinio formulėmis. Kiekvienų bandymu tiesė yra padalinama į:

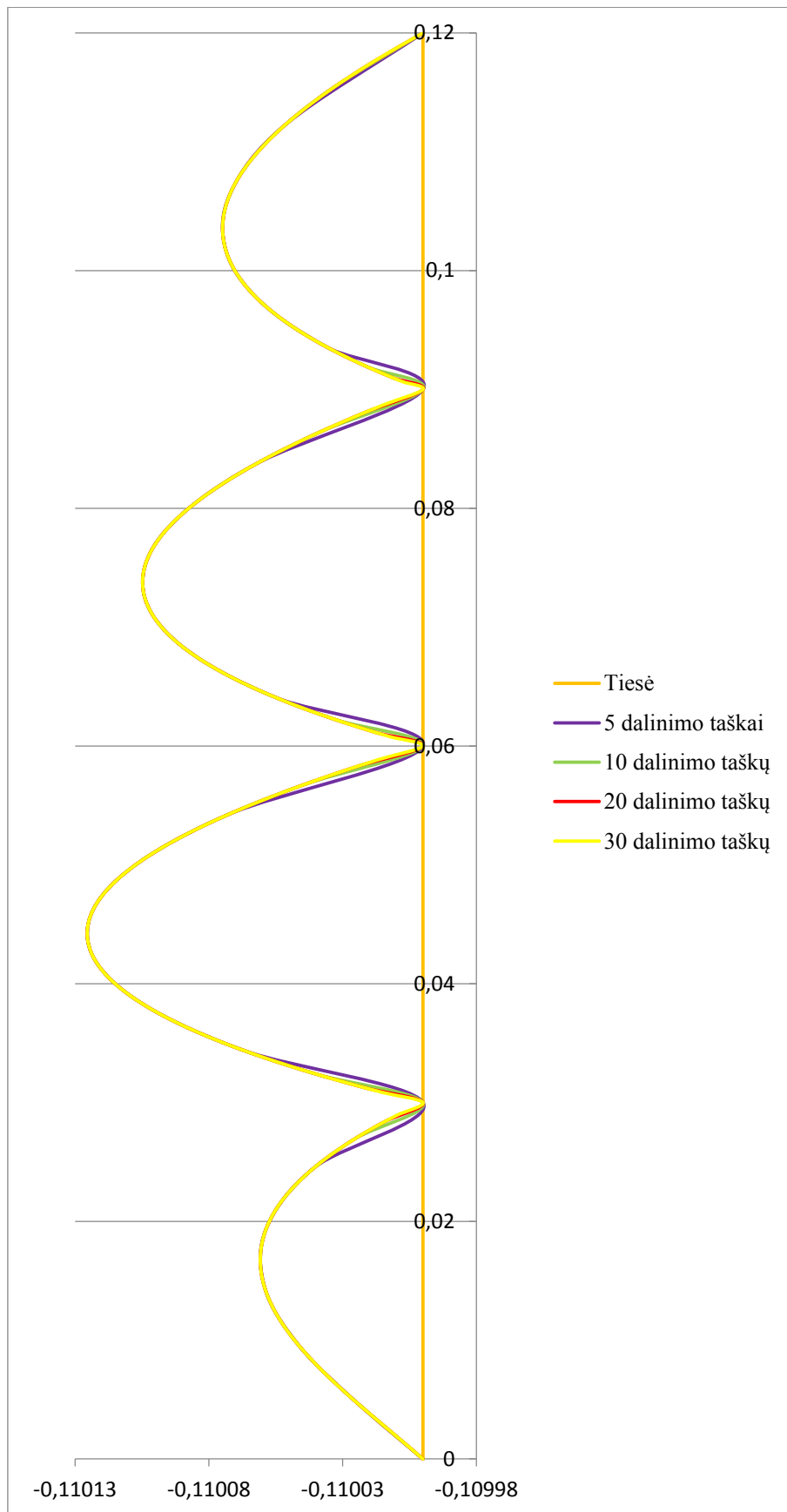
- 3 žingsnio taškus, atstumas tarp taškų 6 cm.
- 5 žingsnio taškus, atstumas tarp taškų 3 cm.
- 9 žingsnio taškus, atstumas tarp taškų 1,5 cm.
- 17 žingsnio taškų, atstumas tarp taškų 0,075 cm.

Visų pirma atliekant tyrimą siekiama išsiaiškinti į kiek dalinimo taškų padalinti atstumus tarp žingsnio taškų, kad būtų gauta kuo mažesnė paklaida. Pirmojo tyrimo rezultatai atskleidė, jog dalinimo taškų padalinimas atstumu tarp žingsnio taškų nedaro didelės įtakos norimiems rezultatams gauti (32 pav.). Pasirinkta, kad bus 10 dalinimo taškų.



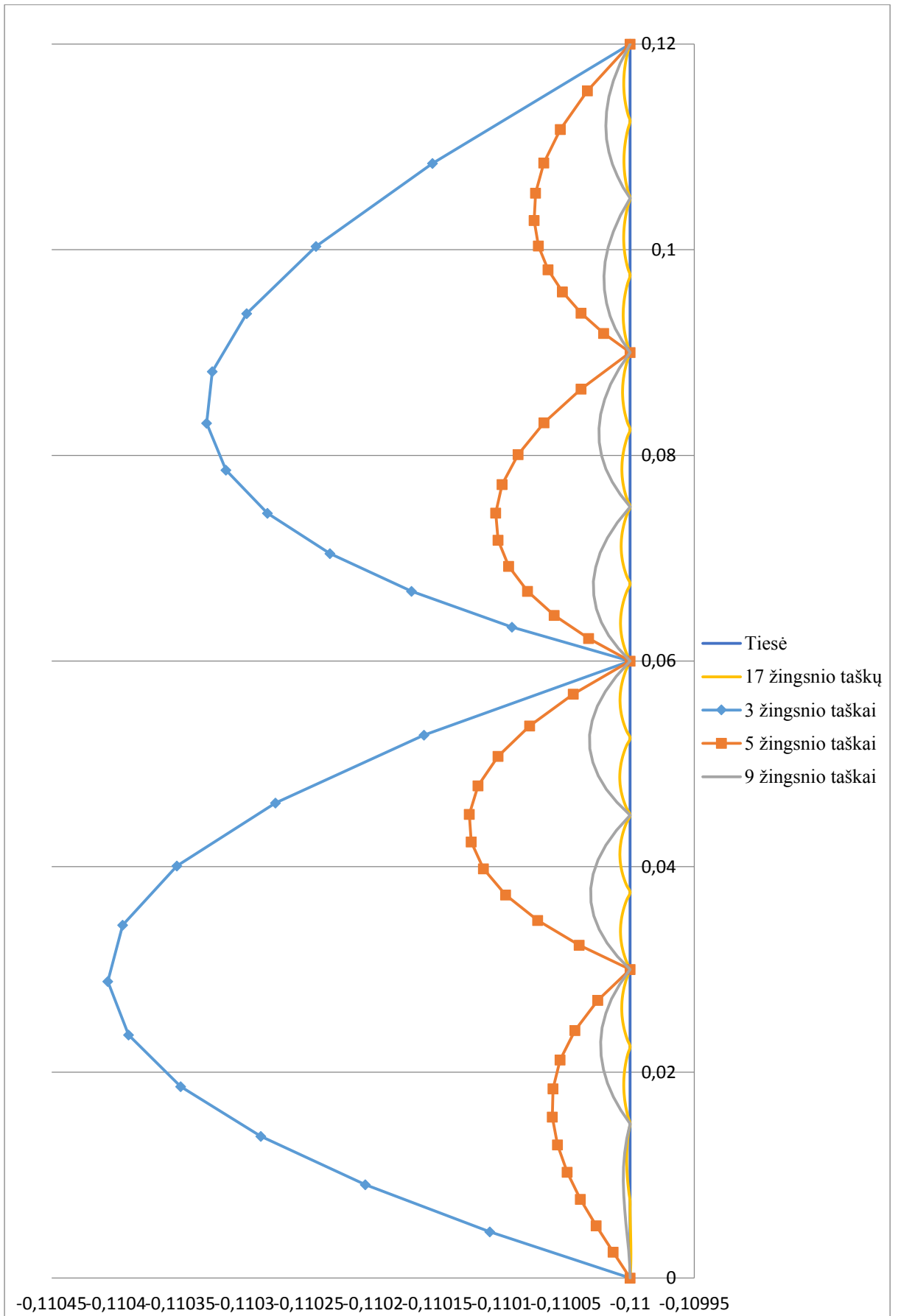
32 pav. Paklaidos priklausomybė nuo dalinimo taškų

Braižomos tiesės su 5 žingsnio taškais su skirtingu dalinimo taškų kiekiu. Skaičiavimo duomenys pateikti 33 pav.



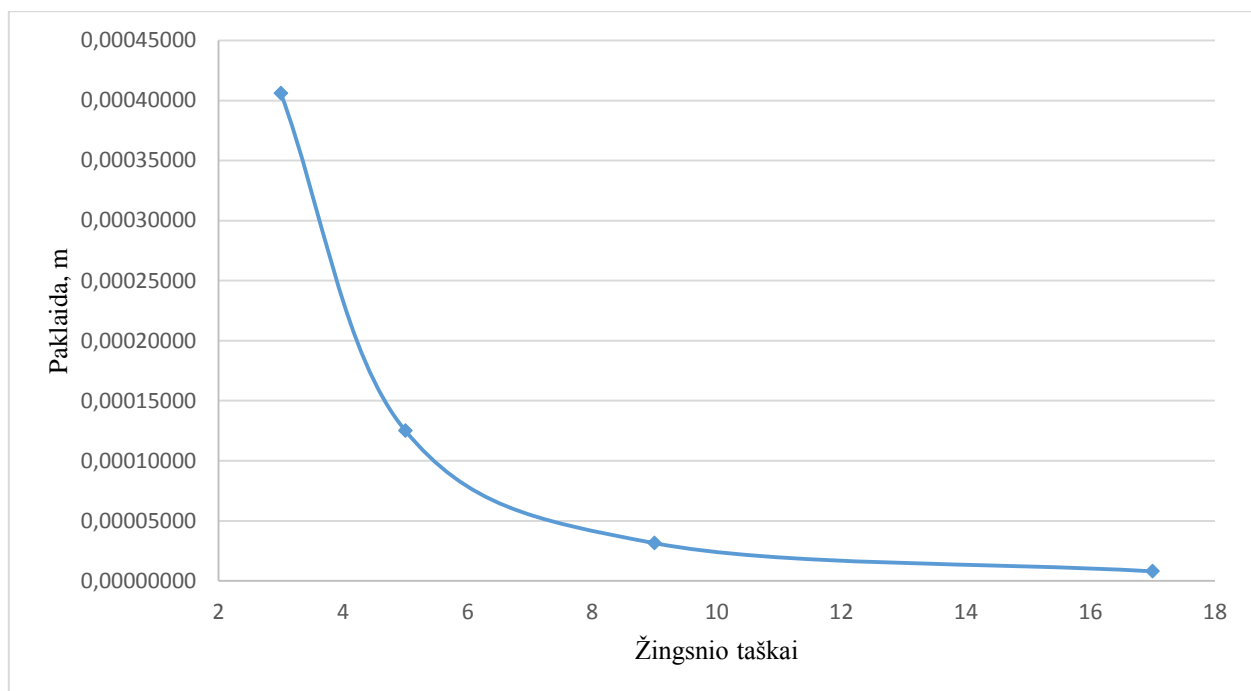
33 pav. Tiesės brėžimas, kai skirtingas dalinimo taškų kiekis

Pagal apskaičiuotus posūkio kampus braižomos keturios tiesės suskirstant į žingsnio taškus, kai greitis reguliuojamas. Skaičiavimo duomenys pateikti 34 pav.



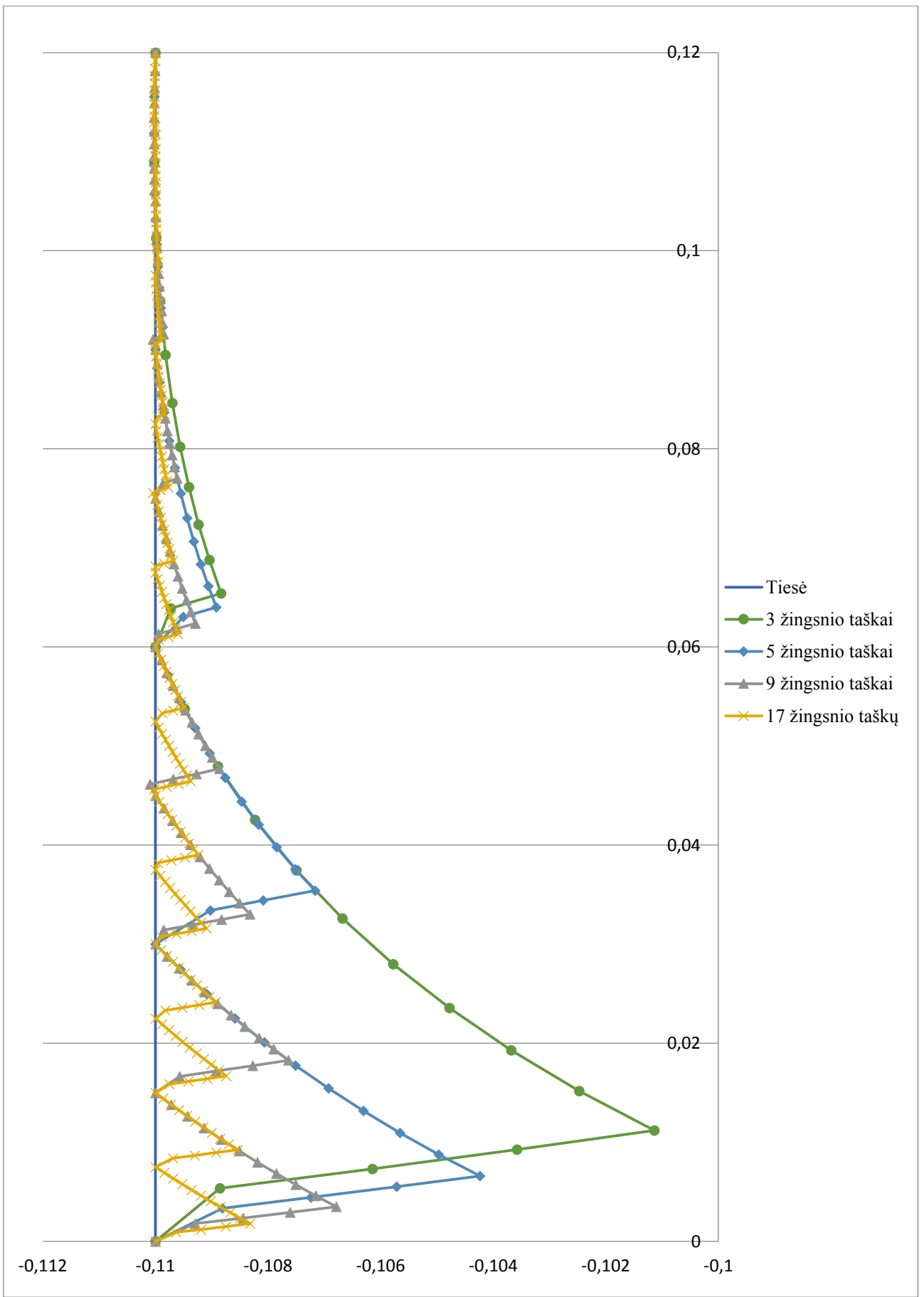
34 pav. Tiesės brėžimas, kai greitis reguliuojamas

Pagal nubraižytas tieses, kai greitis reguliuojamas gauname paklaidų grafiką, kuriame matoma, kad didinant žingsnio taškus gaunama mažesnė paklaida ir galima nubrėžti tiesesnę tiesę, kai 3 žingsnio taškai paklaida $4,06 \times 10^{-4}$, kai 17 žingsnio taškų paklaida $7,9 \times 10^{-6}$.



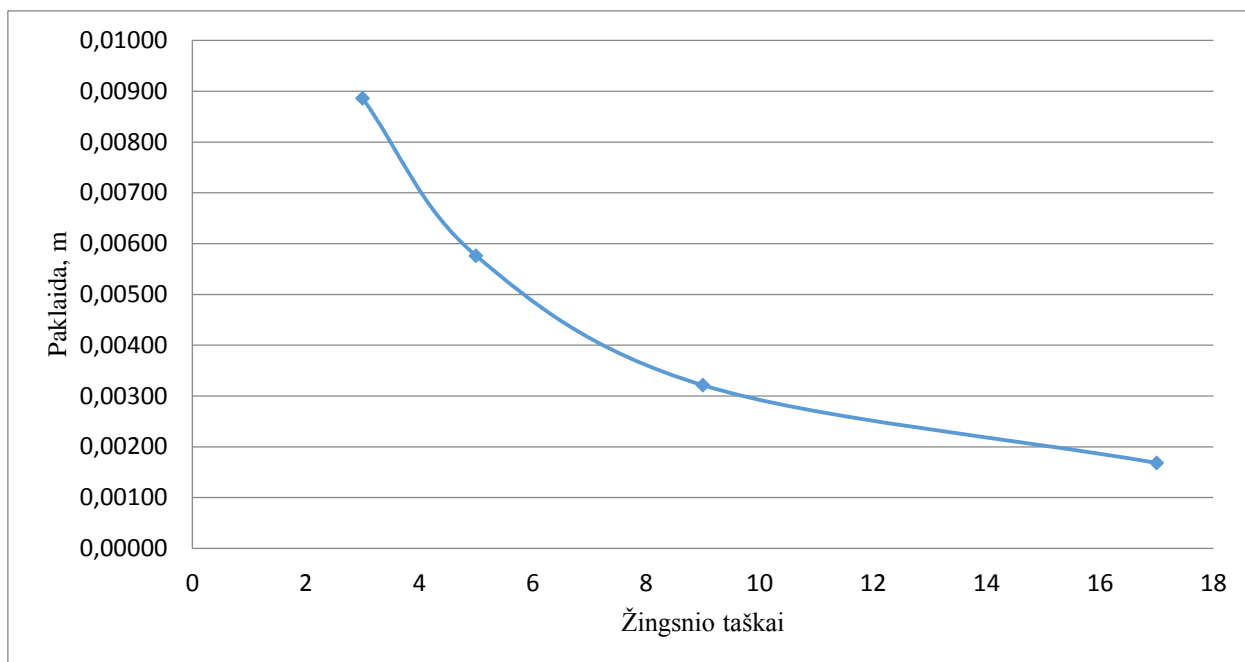
35 pav. Paklaidos priklausomybė nuo žingsnio taškų

Pagal apskaičiuotus posūkio kampus braižomos keturios tiesės suskirstant į žingsnio taškus, kai greitis nereguliuojamas. Skaičiavimo duomenys pateikti 36 pav.



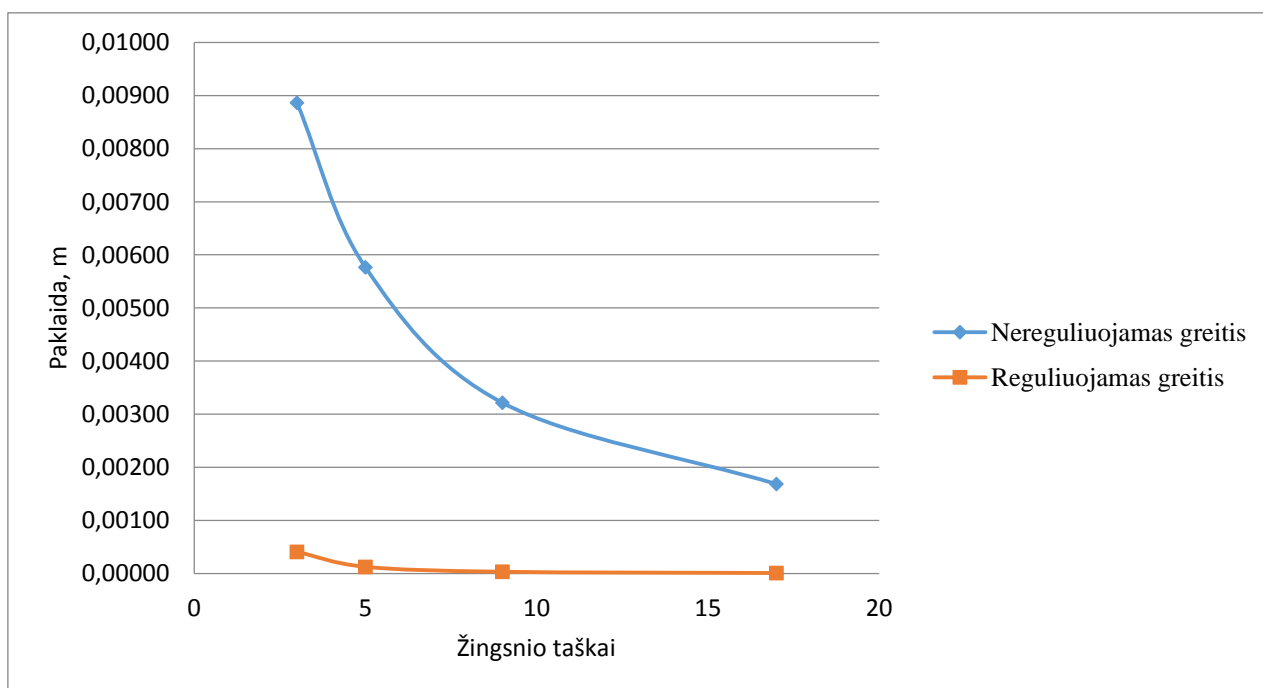
36 pav. Tiesės brėžimas, kai greitis nereguliuojamas

Pagal braižytas tieses, kai greitis nereguliuojamas gauname paklaidų grafiką, kuriame matoma, kad didinant žingsnio taškų kiekį gaunama mažesnė paklaida, kai 3 žingsnio taškai paklaida $8,86 \times 10^{-3}$, kai 17 žingsnio taškų paklaida $1,684 \times 10^{-3}$.



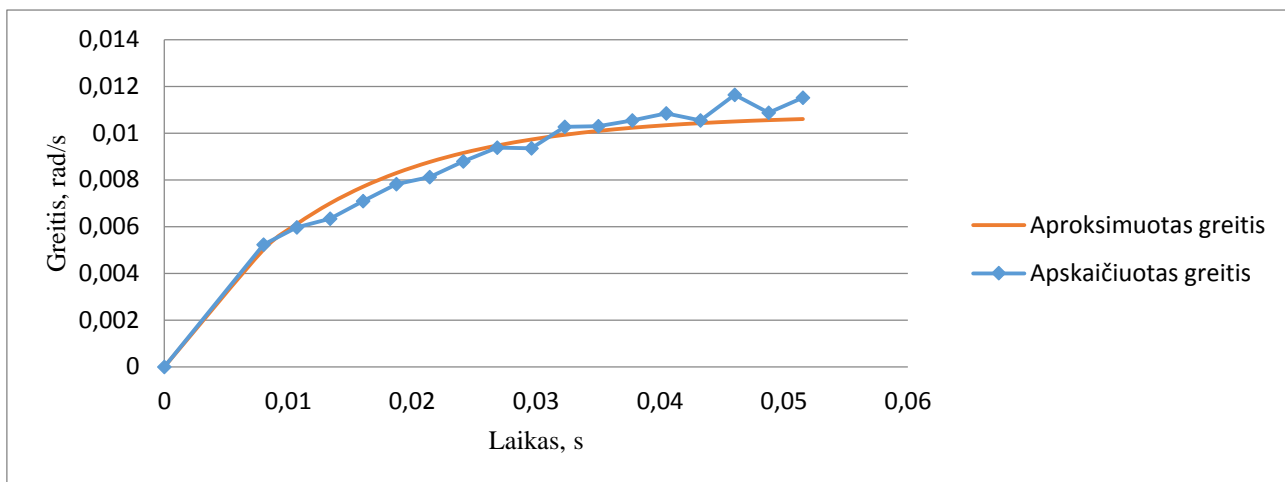
37 pav. Paklaidos priklausomybė nuo žingsnio taškų

Turint paklaidas reguliuojamo ir nereguliuojamo greičio, galima daryti išvada, kad norint pasiekti kuo didesnę tikslumą greitis turi būti reguliuojamas. Kai 3 žingsnio taškai skirtumas tarp reguliuojamo ir nereguliuojamo greičio paklaidų 8,46 mm arba 21 kartas, kai 17 žingsnio taškų skirtumas tarp reguliuojamo ir nereguliuojamo greičio paklaidų 1,676 mm arba 212 kartų.



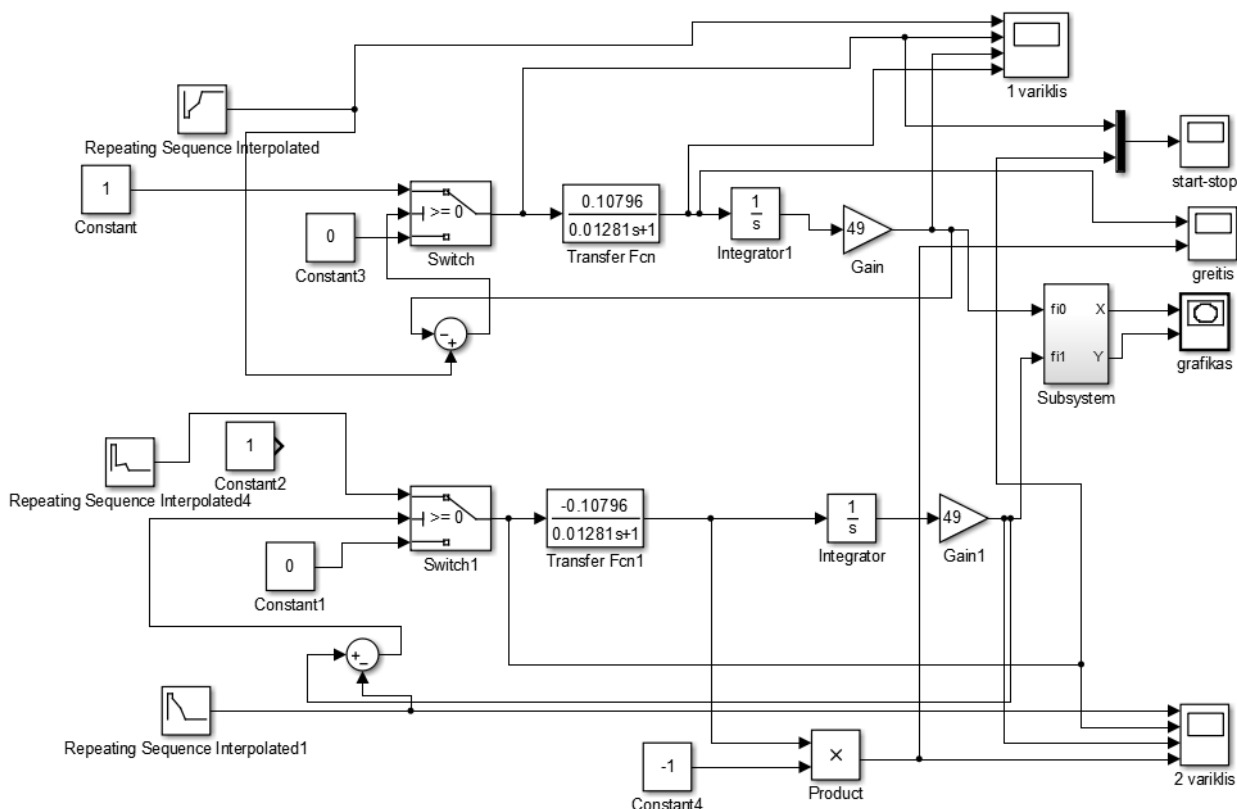
38 pav. Paklaidos priklausomybė nuo žingsnio taškų

Išanalizavus greičio priklausomybės grafiką apskaičiuotas kampinis greitis apie 0,0108 rad/s. Paklaidos pagal apskaičiuotą aproksimuotą greitį atsiranda dėl triukšmų.

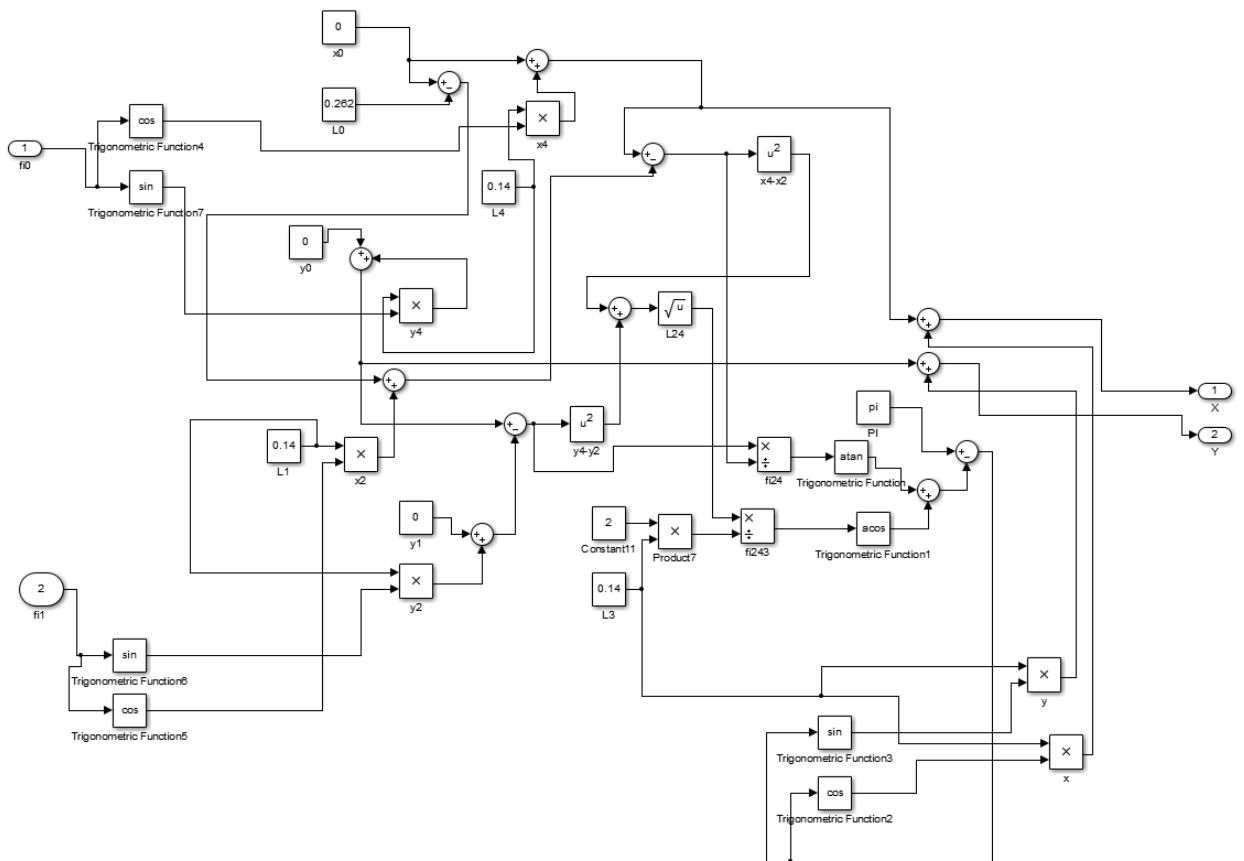


39 pav. Greičio priklausomybė nuo laiko

Dinaminėms charakteristikoms nagrinėti pasinaudota Matlab paketu, kuriame sukurtas realus tiriamojo roboto modelis (40, 41 pav).

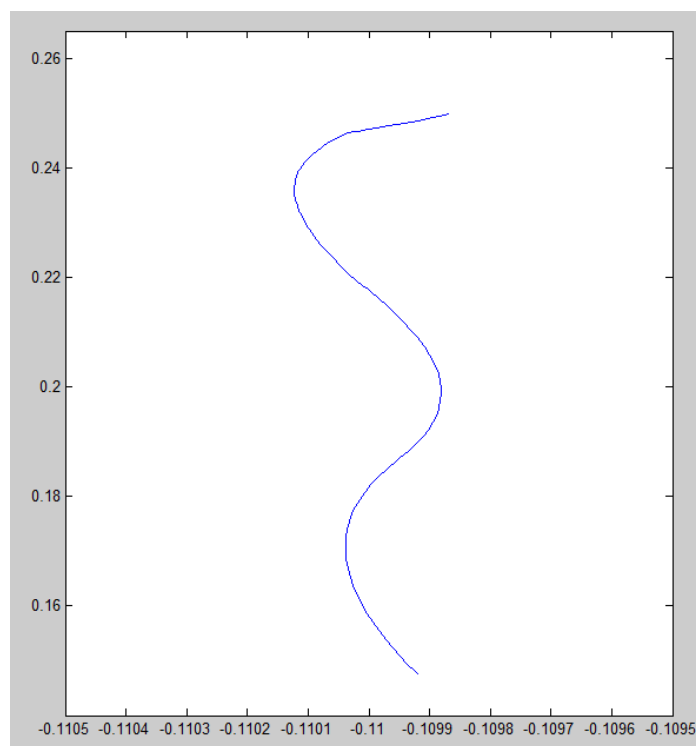


40 pav. Tiriamojo roboto modelis sukurtas Matlab paketu



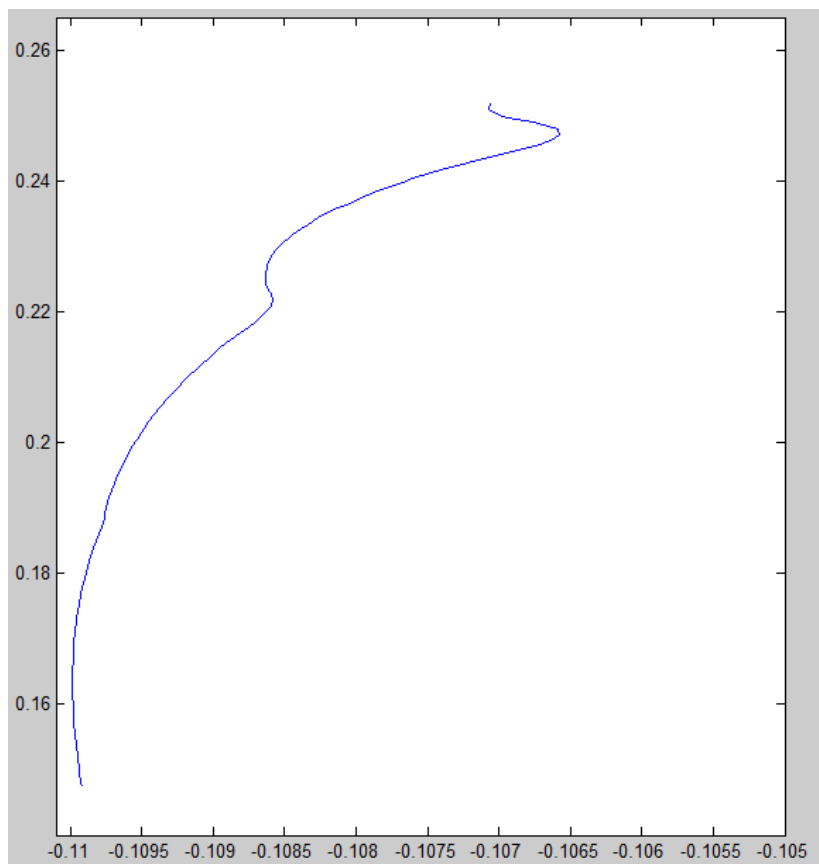
41 pav. Tiriamojo roboto modelio tiesioginio kinematikos uždavinio sprendimo posistemė (subsystem)

Braižoma tiesė, kai greitis reguliuojamas. Greitis reguliuojamas mažinant ar didinant varikliams įtampą. Skaičiavimo duomenys pateikiami 42 pav.



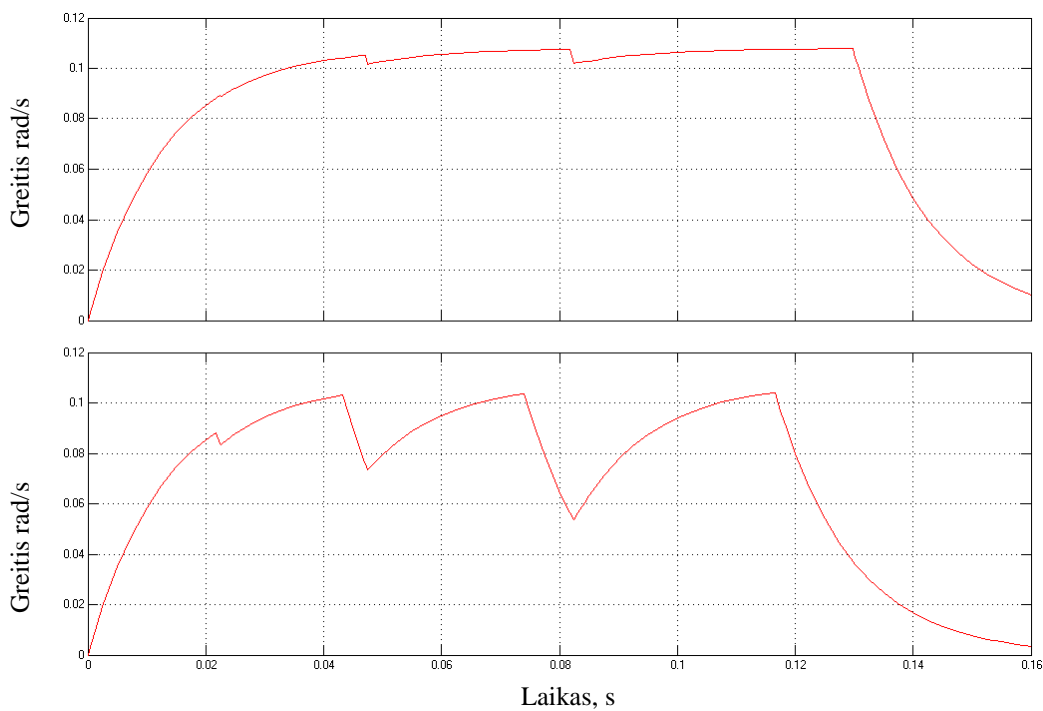
42 pav. Nubrėžta tiesė, kai pavarų greitis reguliuojamas

Braižoma tiesė, kai greitis nereguliuojamas. Skaičiavimo duomenys pateikiami 43 pav.

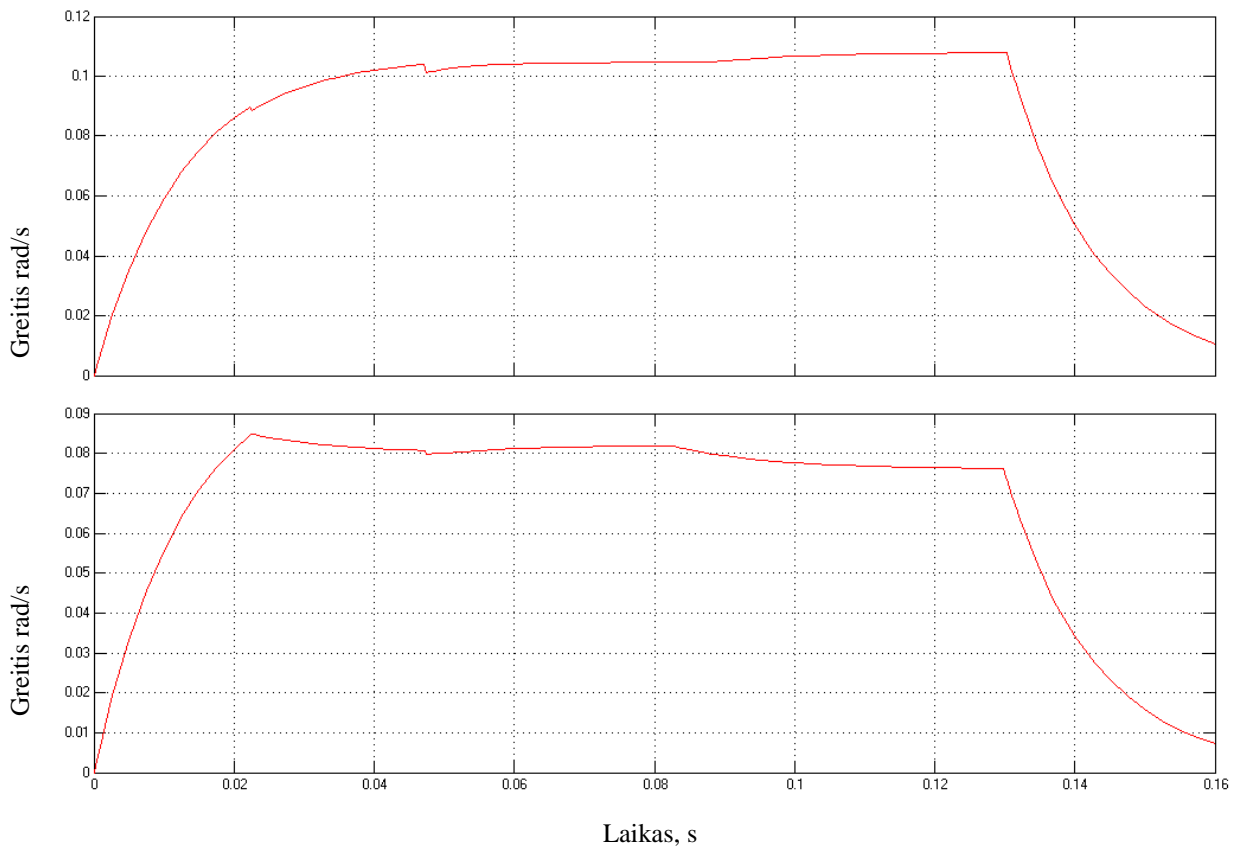


43 pav. Nubrėžta tiesė, kai pavarų greitis nereguliuojamas

Greičių kitimo grafikai, kai reguliuojamas ir nereguliuojamas greitis. Skaičiavimai pateikti 44 pav. ir 45 pav.



44 pav. Pavarų judėjimas, kai greitis nereguliuojamas



45 pav. Pavarų judėjimas, kai greitis reguliuojamas

Pagal gautus skaičiavimus šiuo atveju galime daryti išvada, kad kaip ir teoriniame modelyje paklaidų skirtumas tarp nereguliuojamo ir reguliuojamo yra keliolika kartų. Paklaidos nereguliuojamose pavarose atsiranda, nes pavarų sustojimo metu pavaros dar juda iš inercijos. Išanalizavus visus tiek teorinio tiek dinaminio modelio grafikus galima daryti išvada, kad norint gauti kuo tiesesnę tiesę turi būti pavarų greitis reguliuojamas.

IŠVADOS

1. Atlikta robotų su uždara kinematine grandine literatūros analizė, susipažinta su valdymo galimybėmis, panaudojimo sritimis;
2. Sumontavus galinius išjungėjus, sukurtas kalibravimo algoritmas ir atliktas roboto su uždara kinematine grandine kalibravimas, nustatant pradžios tašką;
3. Sudarytos lygčių sistemos, išspręsti tiesioginį ir atvirkštinį kinematinį uždavinį. Pagal lygčių sistemas apskaičiuoti roboto su uždara kinematine grandine posūkio kampai;
4. Norint nubrėžti kuo tiesesnę tiesę pavaros turi būti reguliuojamo greičio, nes pagal teorinį modelį tarp reguliuojamo ir nereguliuojamo greičio prie 3 žingsnio taškų paklaidos skirtumas 8,46 mm, o prie 17 žingsnio taškų 1,676 mm. Kuo daugiau žingsnio taškų tuo paklaida mažesnė, nes prie 17 žingsnio taškų paklaida tik 8×10^{-6} mm. Pagal dinaminį modelį įvertinant pašalinius trukdžius paklaidos apie 1 mm. Darbinėje zonoje braižant neišvengiamai bus susidurta su neapibrėžtėmis (singularity).

LITERATŪRA

1. AYSSAM, E., SARWAT, H., TAREK, S. Cartesian Parallel Manipulator Modeling, Control and Simulation. *Parallel Manipulators, towards New Applications*, 2008, p. 269-294.
2. BALAN, R., MATIES, V., RAD, C., et al. Kinematics and fuzzy control of ISOGLIDE3 medical parallel robot. *Mechanika*, 1, 2009, p. 62-66.
3. Delta parallel robot – the story of success. Prieiga per internetą <http://robotics.caltech.edu/~jwb/courses/ME115/handouts/DeltaRobotHistory.pdf>, 2017.
4. Delta robot with Arduino. Prieiga per internetą <http://robottini.altervista.org/delta-robot-with-arduino>, 2017.
5. Dual – arm SCARA educational robot Prieiga per internetą <http://www.mecademic.com/DexTAR.html>, 2017.
6. LAPUSAN, C., MATIES, V., BALAN R., et el. Fuzzy logic controller design for planar parallel robot. *DAAAM International*, 2009, p. 43-44.
7. PATEL, Y. D., GEORGE, P. M. Parallel Manipulators applications – a survey. *Modern mechanical engineering*, 2012, p. 57-64.
8. PEARSON, K. *Arduino-controlled industrial robotic arm for your desktop*, 2014, Prieiga per internetą <http://makerflux.com/arduino-controlled-industrial-robotic-arm-desktop/>, 2017.
9. SANFILIPPO, F. *Study of a parallel delta robot controlled via iphone accelerometer*, 2010. Prieiga per internetą <http://filipposanfilippo.inspitivity.com/reports/sanfilippo-parrallel-delta-robot-iphone-accelerometer.pdf>, 2017.
10. SERGIU-DAN, S., MILOS, M., VISTRIAN, M., et al. Genetic algorithms for workspace optimization of planar medical parallel robot. Prieiga per internetą http://www.people.vcu.edu/~mmanic/papers/2009/ICETIC09_StanManicBalanMaties_Gene ticMedParallRbt.pdf, 2017.
11. SHANG, W., CONG, S. Nonlinear dynamic control and friction compensation of parallel manipulators, *Serial anr Parallel robot manipulators – kinematics, dynamics, control and optimization*, 2012, p. 231-252.

12. SIMAAN, N. *Analysis and Synthesis of Parallel Robots for Medical Applications*. Haifa: Israel institute of technology, 1999.
13. Sketchy. Prieiga per internetą <http://jarkman.co.uk/catalog/robots/sketchy.htm>, 2017.
14. Technical data. Peržiūrėta Prieiga per internetą http://www.micromotors.eu/planetary_gear_motor_e192.html, 2017.
15. The drawing apparatus by Robert Howsare. Prieiga per internetą <http://www.juxtapoz.com/news/the-drawing-apparatus-by-robert-howsare/>, 2017.
16. The true origins of Parallel robots. Prieiga per internetą <http://www.parallelic.org/Reviews/Review007.html>, 2017.
17. What is a parallel robot?. Prieiga per internetą <http://www.mecademic.com/What-is-a-parallel-robot.html>, 2017.

PRIEDAI

1. Kalibravimo kodas:

```

if (kalibravimas == 0 )
{
int gal = 0;
gal = digitalRead(galini);
if (gal == HIGH)
{
va1.write(1300);
}
else
{
va1.write(1500);
}
}
if (kalibravimas2 == 0)
{
int gal2 = 0;
gal2 = digitalRead(galini2);
if (gal2 == HIGH)
{
va2.write(1650);
}
else
{
va2.write(1500);
}
}
int gal = 0;
gal = digitalRead(galini);
int gal2 = 0;
gal2 = digitalRead(galini2);

if (gal2 == LOW and gal == LOW)
{
kalibravimas = kalibravimas +1;
kalibravimas2 = kalibravimas2 +1;
encoder0Pos = 0;
encoder0Pos2 = 0;
}

```

sukalibruojama „pirmoji“ ranka

sukalibruojama „antroji“ ranka

tikrinama ar abi rankos susikalibravo t.y. sekančius veiksmus atlieka kai abi rankos susikalibruoja

Abiejų rankų enkoderių reikšmės prilyginamos nuliui.

1. Programos kodas:

```

#include <Servo.h>
Servo va1;
Servo va2;           //2 variklis
#define encoder0PinA 2
#define encoder0PinB 3
#define encoder0PinAA 19    //2 variklis
#define encoder0PinBB 18    //2 variklis
long int encoder0Pos = 0;
long int encoder0Pos2 = 0; //2 variklis
int kalibravimas = 0;
const int galinis = 8;
int kalibravimas2 = 0;
const int galinis2 = 7;
int i;
float masyvas[12] = {6230, 8492, 10878, 14041, 17614};
float masyvas2[12] = {-4375, -1303, 2351, 7567, 13268};
void setup() {
  pinMode(encoder0PinA, INPUT); pinMode(encoder0PinB, INPUT);
  attachInterrupt(0, doEncoderA, CHANGE);
  va1.attach(6);
  pinMode(encoder0PinAA, INPUT); pinMode(encoder0PinBB, INPUT); //2 variklis
  attachInterrupt(4, encoderis2, CHANGE); //2 variklis
  va2.attach(4); //2 variklis
  Serial.begin (250000);
  for (i=0;i<3;i++){Serial.print(masyvas[i]);}
  long sign(long int u){
    int result;
    if (u>0){result = 1; return result;}
    else result = -1; return result;
    Serial.print("result\n");
    Serial.print(result);
    Serial.print(" \n");
  }

  void loop() {
    if (kalibravimas == 0 ){
      int gal = 0;
      gal = digitalRead(galinis);
      if (gal == HIGH){va1.write(1300);}
      else { va1.write(1500);}
    }
    if (kalibravimas2 == 0){
      int gal2 = 0;
      gal2 = digitalRead(galinis2);
      if (gal2 == HIGH){ va2.write(1650);}
      else { va2.write(1500); }
    }
    int gal = 0;

```

```

gal = digitalRead(galinis);
int gal2 = 0;
gal2 = digitalRead(galinis2);
if (gal2 == LOW and gal == LOW){
kalibravimas = kalibravimas +1;
kalibravimas2 = kalibravimas2 +1;
encoder0Pos = 0;
encoder0Pos2 = 0;
}
if (kalibravimas2 == 1 and kalibravimas == 1){
long int kampas;
long int u;
long int kampas2;
long int u2;

for (i=0;i<5;i++){
kampas=masyvas[i];
u = kampas-encoder0Pos;
kampas2=masyvas2[i];
u2 = kampas2-encoder0Pos2;
int g1 = 0;
int g2 = 0;
if(u>0)
{va1.write(1650);}
if(u<0)
{va1.write(1350);}
if(u2>0)
{va2.write(1350);}
if(u2<0)
{va2.write(1650);}
while (g1==0 || g2==0){
if(sign(u) != sign(kampas-encoder0Pos))
{
g1 = 1;
va1.write(1500);
}
if(sign(u2) != sign(kampas2-encoder0Pos2))
{
g2 = 1;
va2.write(1500);
}
}
va1.write(1500);
va2.write(1500);
Serial.print(" uzduotas ");
Serial.print(kampas);
Serial.print(" encoderpos ");
Serial.println(encoder0Pos);
Serial.print(" uzduotas2 ");
Serial.print(kampas2);
Serial.print(" encoderpos2 ");
Serial.println(encoder0Pos2);

```

```

delay (5000);
}
}
}
void doEncoderA() {
if (digitalRead(encoder0PinA) == HIGH) {
if (digitalRead(encoder0PinB) == LOW) {
encoder0Pos = encoder0Pos + 1;
}
else {
encoder0Pos = encoder0Pos - 1;
}
}
}
}
void encoderis2() {
if (digitalRead(encoder0PinAA) == HIGH) { //2 variklis
if (digitalRead(encoder0PinBB) == LOW) {
encoder0Pos2 = encoder0Pos2 + 1;
}
else {
encoder0Pos2 = encoder0Pos2 - 1;
}
}
}
}
}

```