



VILNIAUS UNIVERSITETAS
ŠIAULIŲ AKADEMIJA

IŠMANIOSIOS GAMYBOS INŽINERIJOS MAGISTRO STUDIJŲ PROGRAMA

Donatas Kiznis

Magistro studijų baigiamasis darbas

Programų kūrimo "Offline" metodu robotinėms sistemoms tyrimas

Darbo vadovas (-ė): dr. Edvardas
Bielskis

Šiauliai, 2022

**Studijuojančiojo, teikiančio baigiamąjį
darbą, GARANTIJA**

WARRANTY of Final Thesis

Vardas, pavardė <i>Name, Surname</i>	Donatas Kiznis
Padalinys <i>Faculty</i>	Šiaulių akademija <i>Šiauliai Academy</i>
Studijų programa <i>Study Programme</i>	Išmaniosios gamybos inžinerija <i>Smart Manufacturing Engineering</i>
Darbo pavadinimas <i>Thesis topic</i>	Programų kūrimo "Offline" metodu robotinėms sistemoms tyrimas <i>Research of Offline Application Development for Robotic Systems</i>
Darbo tipas <i>Thesis type</i>	Baigiamasis darbas <i>Final Thesis</i>

Garantuoju, kad mano baigiamasis darbas yra *I guarantee that my thesis is prepared in good faith*
parengtas sąžiningai ir savarankiškai, kitų asmenų *and independently, there is no contribution to this*
indėlio į parengtą darbą nėra. Jokių neteisėtų *work from other individuals. I have not made any*
mokėjimų už šį darbą niekam nesu mokėjęs. *illegal payments related to this work.*

Šiame darbe tiesiogiai ar netiesiogiai panaudotos *Quotes from other sources directly or indirectly*
kitų šaltinių citatos yra pažymėtos literatūros *used in this thesis, are indicated in literature*
nuorodose. *references.*

Aš, Donatas Kiznis, pateikdamas (-a) šį darbą, patvirtinu (pažymėti)



SANTRAUKA

Šiame magistro darbe buvo ištirtas roboto „Touch sensor“ veikimas, ir kaip jis gali padėti ženkliai sumažinti paklaidas apdorojant metalo gaminius. Taip pat yra aprašomas ir įvertinamas „Offline“ programavimo būdas.

Darbe yra apžvelgiami „Touch sensor“ gaminio paieškos ir „Offline“ programavimo panaudojimai. Taip pat suprojektavimas ir aprašomas „Touch sensor“ veikimas ir jo nauda suvirinimo procesuose

Teorinėje dalyje yra apžvelgiami panašūs projektai. Aprašyta, kaip jie panaudojo „Touch sensor“ arba „Offline“ programavimą, ir kokius tikslus pasiekė.

Projektinėje dalyje yra aprašoma:

1. „Offline“ programavimo pranašumai darbo procesui pagreitinti.
2. Išmatuotos paklaidos atsiradusios dėl simuliacinės programos ir realaus roboto matematinio modulio nesutapimo.
3. Pritaikytas „Touch sensor“ veikimas, kad paklaidos būtų kiek galima labiau sumažintos.
4. Kaip papildoma priemonė tikslumui padidinti - atliktas ašių perkalibravimas, kad paklaidos būtų labiau sumažintos.

Šie metodai („Offline“ programavimas, „Touch sensor“) leidžia padidinti robotinės suvirinimo sistemos našumą, tikslumą – tai sumažina sistemos atsiperkamumo laikotarpį, pagerina suvirinamų gaminių kokybę.

Raktiniai žodžiai: „Touch sensor“, „Offline“ programavimas, robotas, suvirinimas.

Kiznis D. (2022). *Research of Offline Application Development for Robotic Systems*. Master degree final project. Vilnius University Šiauliai Academy, Institute of Regional Development, Šiauliai.

SUMMARY

This master thesis investigated the performance of a robotic Touch Sensor and how it can help to significantly reduce errors in metalworking. The offline programming method is also described and evaluated.

The applications of Touch sensor product retrieval and Offline programming are reviewed. It also designs and describes the operation of the Touch sensor and its usefulness in welding processes

In the theoretical part similar projects are reviewed. It describes how they have used Touch sensor or Offline Programming and what objectives they have achieved.

The project part describes:

- 1."The advantages of offline programming to speed up the work process.
2. Measured errors due to the mismatch between the simulation program and the mathematical module of the real robot.
- 3.The Touch Sensor operation is adapted to minimise the errors.
- 4.As an additional measure to increase accuracy, a recalibration of the axes was carried out to further reduce the errors.

These methods (Offline programming, Touch sensor) allow to increase the performance and accuracy of the robotic welding system, which reduces the payback period of the system and improves the quality of the welded products.

Keywords: touch sensor, offline programming, robot, welding

Turinys

1. Įvadas	11
2. Panašus įrenginiai.....	12
2.1 CAD pagrįsta suvirinimo darbų programavimo platforma	12
2.1.1 Sistemos apžvalga.....	12
2.1.2 Dizainas ir kūrimas	13
2.1.3 Rezultatai	16
2.2 Laivo korpuso suvirinimas ir siūlės paieška su lietimo sensorių	18
2.2.2 Lietimo jutiklio programos struktūra.....	20
2.3 Siūlių sekimas naudojant kelis sensorius	26
2.3.1 Jutiklis su kamera	26
2.3.2 Optimalaus degiklio aukščio nustatymas	28
2.4 Adaptpack „offline“ programavimas	29
2.4.1 Siulomas sprendimas	29
2.4.2 Sistemos architektura	30
2.4.3 Bandymai ir rezultatai	32
3. Teorinė dalys	35
3.1 Inostart projektas	35
3.1.1 Lietimo sensoriaus veikimas	35

3.1.2 Lietimo sensoriaus elektroninių grandinių projektavimas ir gamyba	37
3.1.3 Plazmos srovės lanko aukščio kontrolės sistema	40
3.1.4 Elektros plazmos oro tarpelio valdymo sistemos projektavimas ir gamyba	40
3.1.5 Inostart rezultatai	45
3.2 „Delfoi Arc“ programa	45
4. Tiriamoji dalys	46
4.1 Pasiruošimo darbai	46
4.2 Gaminio paruošimas	48
4.3 Ašių perkalibravimas	51
4.4 Programos kodas	53
5. Išvados	55

Paveikslėlių sąrašas

1 pav. Sistemos architektūra	12
2 pav. „Offline“ programavimo platforma.....	13
3 pav. Roboto koordinacijų sistema.....	13
4 pav. Kelio planavimas virinimo programai	15
5 pav. Simuliacinės programos diagrama	16
6 pav. Virtualus programos vaizdas.....	17
7 pav. Laivo korpuso dalys	18
8 pav. Pagrindinės programos diagrama.....	19
9 pav. Virinimo procedūros programos diagrama	20
10 pav. Vietos ištraukimo diagrama	21
11 pav. Ruošinio poslinkio diagrama.....	22
12 pav. Ruošinio krašto aptikimo sistema	23
13 pav. Koordinacijų sistemos transformacija.....	23
14 pav. Aptikimo moduliai	24
15 pav. Aptikimo sekos diagrama.....	24
16 pav. Suvirinimo siūlė	25
17 pav. Roboto sistema.....	26
18 pav. Santykis tarp degiklio ir ruošinio kai yra sekama ruošinio išorė	27
19 pav. Santykis tarp degiklio ir ruošinio kai yra sekama ruošinio vidus	27
20 pav. Kameros fiksuojamas vaizdas	27

21 pav. Sistemos architektūros diagrama.....	29
22 pav. Sistemos grafinė sąsaja	30
23 pav. Paletavimo kodas	32
24 pav. Grafinė paletizavimo simuliacija	33
25 pav. Sugeneruotas programos kodas.....	34
26 pav. Lietimo sensoriaus veikimas	36
27 pav. Lietimo sensoriaus geometrijos matavimo principo aiškinimas	36
28 pav. Projektuojamo lietim sensoriaus principinė elektrinė schema	37
29 pav. Sukurto lietim sensoriaus spausdintinė plokštė.....	39
30 pav. Pagamintų lietim sensorių bendras vaizdas.....	39
31 pav. Elektros plazminio metalo pjovimo veikimo principas	41
32 pav. Suprojektuoto oro tarpelio valdiklio metalo plazminiam pjovimui principinė elektrinė schema	42
33 pav. Sukurto oro tarpelio valdiklio spausdintinė plokštė.....	43
34 pav. Sukurto valdiklio plokštė	44
35 psv. Sukurto prietaiso bendras vaizdas (dešinėje); plazminio metalo pjovimo sistema (kairėje).....	44
36 pav. Kalibravimo įtaisas.....	46
37 pav. Patikrinta ar robotas gerai sukalibruotas	47
38 pav. Išmatuoti šeši taškai ant pirmo ruošinio.....	47
39 pav. Išmatuoti šeši taškai ant antro ruošinio	48
40 pav. Ruošinio atvaizdas simuliacijoje.....	48

41 pav. Kairėje ruošinio padėtis laboratorijoje, dešinėje ruošinių pozicijos atkurtos ant „Delfoi Arc“ programos	49
--	----

Lentelių sąrašas

1 lentelė. Paklaidos padarius pirminius matavimus	49
2 lentelė. Pirminiai matavimai naudojant „touch sensor“	50
3 lentelė. Matavimai atlikus „mastering“	51
4 lentelė. Matavimai atlikus „Masteringa“ ir naudojant „Touch sensor“	52

1. Įvadas

Programuojant „Offline“ metodu ir įkeliant sukurtą programą į realų robotą atsiranda nesutapimų. Tai nutinka dėl matematinio modelio nesutapimo tarp „Delfoi“ roboto ir realaus roboto. Problema iškyla kai „Delfoi“ robotas yra laikomas idealiu iš programinio požiūrio, o realaus roboto visos ašys turi būti kalibruojamos. Tai ir sukelia nesutapimus tarp virtualaus modelio aprašymo ir realios robotinės sistemos.

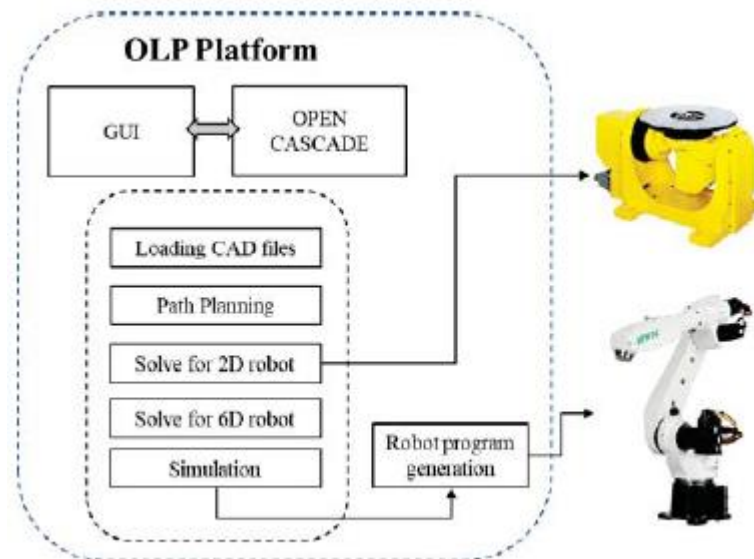
Vienas iš būdų eliminuoti šia problemą yra naudoti papildomą įrangą. Šiame darbe, buvo naudotas lietimasis sensorius. Vietos galiukas patikrina kur yra faktiškai ruošinys, kurį reikia suvirinti ir visa programa yra koreguojama pagal išmatuotus nesutapimus.

2. Panašus įrenginiai

2.1 CAD pagrįsta suvirinimo darbų programavimo platforma

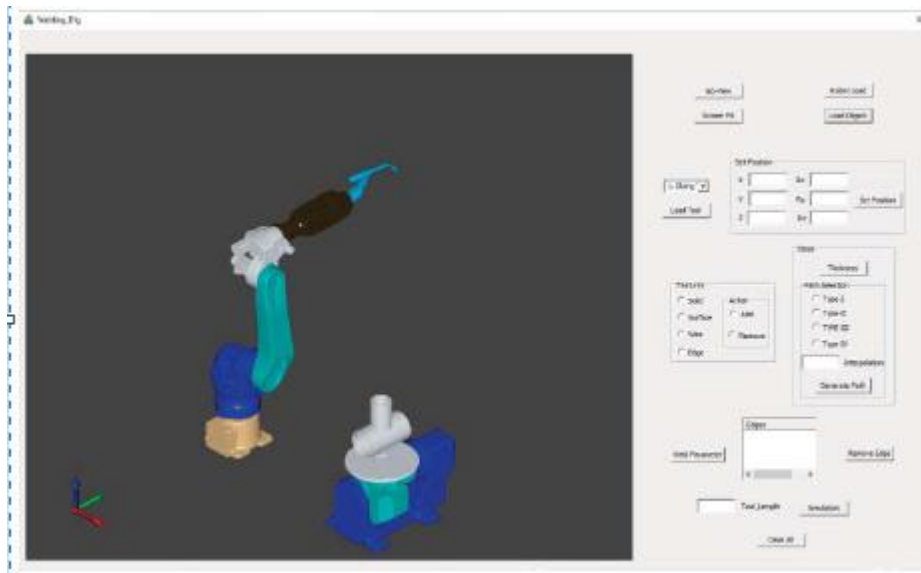
2.1.1 Sistemos apžvalga

Sukurta ir išplėtota CAD(angl. computer-aided design) pagrindu su „Offline“ modeliavimo platforma, skirta 6-DOF(angl. six degrees of freedom) ir 2-DOF(angl. two degrees of freedom) robotų suvirinimo užduočiai atlikti [1]. 6-DOF roboto programa buvo suprojektuota ir sukurta naudojant OCC biblioteką ir "Microsoft Visual Studio". Siūloma sistema gali įvertinti idealią padėtį ir laikyseną suvirinimo užduočiai atlikti, esant tolygiam mazgų pasiskirstymui ant kreivės. Be to, tam tikrai užduočiai atlikti galima sugeneruoti roboto judesių trajektoriją vengiančia susidūrimų su darbiniais paviršiais.



1 pav. Sistemos architektūra

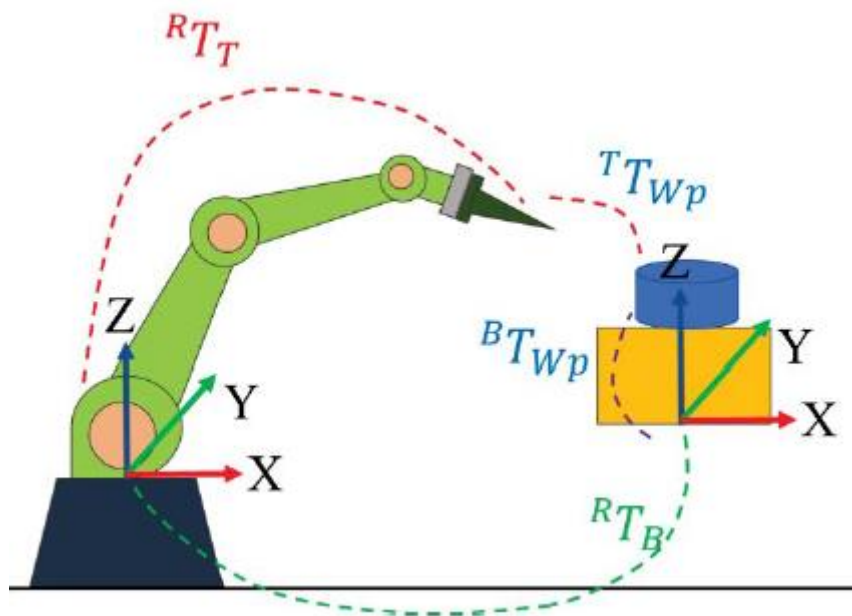
Siūlomos OLP(angl. off-line programming) platformos sistemos architektūra pavaizduota 1 pav. Ją sudaro įvairūs moduliai, pavyzdžiui: CAD failų įkėlimas, kelio planavimas, 2-DOF roboto algoritmas, 6-DOF roboto algoritmas ir modeliavimas, skirtas praktiškai patikrinti veikimą. 2 pav. pavaizduota sukurta 6-DOF ir 2-DOF robotų programavimo „Offline“ platforma suvirinimo užduotims atlikti.



2 pav. „Offline“ programavimo platforma

2.1.2 Dizainas ir kūrimas

„Offline“ programavimo platforma buvo sukurta su įvairiais moduliais, kad būtų galima sukurti pilną sistemą. Siūloma platforma savarankiškai generuoja suvirinimo pozą ir kelią naudodama CAD informaciją. Toliau aprašytas kiekvienas modulis.



3 pav. Roboto koordinačių sistema

A. Siūlomos sistemos koordinačių sistema

Siūlomos OLP sistemos koordinačių sistemų schema pateikta 3 pav. Koordinačių sistema yra tokia:

{R} 6-DOF roboto bazinė koordinačių sistema

{T} Suvirinimo degiklio koordinačių sistema

{B} 2-DOF roboto pagrindo koordinačių sistema

{Wp} Ruošinio koordinačių sistema

Vienarūšės transformacijos matricos tarp koordinačių sistemų nustatomos taip:

RTT: {T} santykis su {R}.

RTB: {B} santykis su {R}.

BTWp: BTB: {Wp}, santykis su {B}.

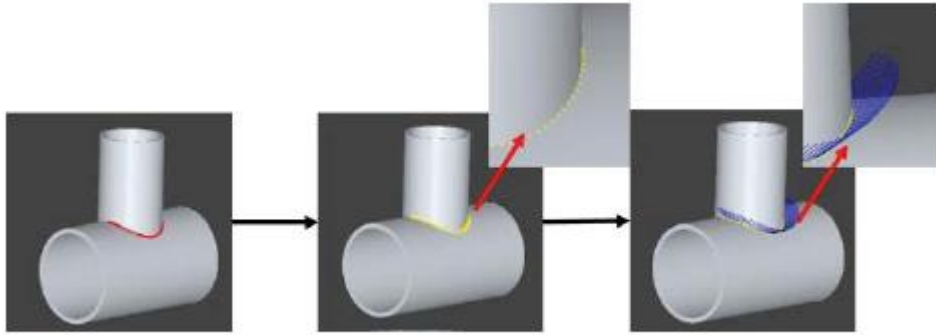
Pagal pirmiau nurodytus transformacijos santykius vienalytę transformacijos matricą RTT galima išreikšti taip:

$${}^R T_T = {}^R T_B {}^B T_{J1} R_Y R_Z {}^{J1} T_{Wp} {}^{Wp} T_T$$

Siekiant supaprastinti suvirinimo sistemą, 2DOF ir 6DOF roboto koordinačių sistema apibrėžiama tame pačiame rėme, kad būtų galima lengviau atlikti skaičiavimus.

B. Kelio planavimas

Siekiant įgyvendinti suvirinimo kelią, pirmiausia nustatoma padėtis ir pozicija išilgai suvirinimo kreivės, kurios tada yra transformuojamos į roboto koordinačių sistemą. Šiame modulyje suvirinimo kelio mazgai ant kreivės sukuriama naudojant OCC biblioteką, kaip parodyta 4 pav. Suvirinimo pozicija ir roboto poza buvo savarankiškai sugeneruotos naudojant CAD informaciją, kaip parodyta 4 pav.



4 pav. Kelio planavimas virinimo programai

Siūlomame metode naudotojas pasirenka ruošinio briauną, kad sukurtų suvirinimo kelią. Naudojant OCC biblioteką mazgai suvirinimo kreivėje generuojami tolygiai. Norint nustatyti suvirinimo užduoties pozą ir poziciją, iš CAD modelio automatiškai apskaičiuojami mėlynos spalvos normaliniai vektoriai, kaip parodyta 4 pav. Šie normaliniai vektoriai manipuliatoriui tarnauja kaip atskaitos taškas, kad jis laikytųsi tinkamos pozicijos (suvirinimo degiklis kiekviename taške visada yra statmena) ir atliktu norimą užduotį. Norint atlikti roboto modeliavimą reikia ir padėties, ir normalinių vektorių. Posvyris, polinkis ir nukrypimas vaizduojami kaip α , β ir γ , kuriuos galima apskaičiuoti naudojant lygtis (2), (3) ir (4)

$$\alpha = \text{atan2}(r_{21}, r_{22}) \quad (2)$$

$$\beta = \text{asin}(-r_{20}) \quad (3)$$

$$\gamma = \text{atan2}(r_{21}, r_{22}) \quad (4)$$

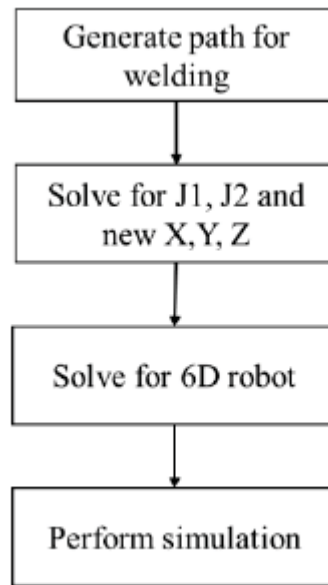
Šiame darbo etape taip pat atsižvelgiama į singularumo klausimus, kad būtų išvengta sunkumų, su kuriais susiduriama įgyvendinimo metu. Yra dvi skirtingos sąlygos, į kurias reikia atsižvelgti sprendžiant singularumo problemas: pirma, kai $\beta = +90^\circ$ ir $\beta = -90^\circ$, kaip parodyta (5) lygtyje.

$$\alpha \pm \beta = \text{atan2}(-r_{12}, r_{11}) \quad (5)$$

Apskaičiavus visą informaciją, imitavimo modulis patikrins siūlomo kelio planavimo algoritmo efektyvumą.

C. Simuliacija

Modeliavimas yra vienas iš pagrindinių sukurtos OLP platformos modulių, skirtų suvirinimo kelio efektyvumui patikrinti. OCC branduolys naudojamas informacijai iš CAD ruošinio (vamzdžio) išgauti. 5 pav. paašškintas siūlomos platformos modeliavimo procesas. Suvirinimo kelias ruošinyje sukuriama naudojant įvairias OCC funkcijas. Po to nustatomos koordinatės (x, y, z) nuo ruošinio ir orientacijos (α, β, γ) tam tikrame suvirinimo įrankio rėme. Šios padėtys ir orientacijos naudojamos kaip 2-DOF roboto įvestis sąnarių kampams ir naujoms 6-DOF roboto padėtimis (X, Y, Z) apskaičiuoti. Naudojant atvirkštinę kinematiką, gaunami atskiri kiekvieno manipulatoriaus ašių kampai, kad būtų galima atlikti modeliavimą. Šis modulis taip pat geba išvengti susidūrimų, kad sėkmingai atliktų duotą užduotį.

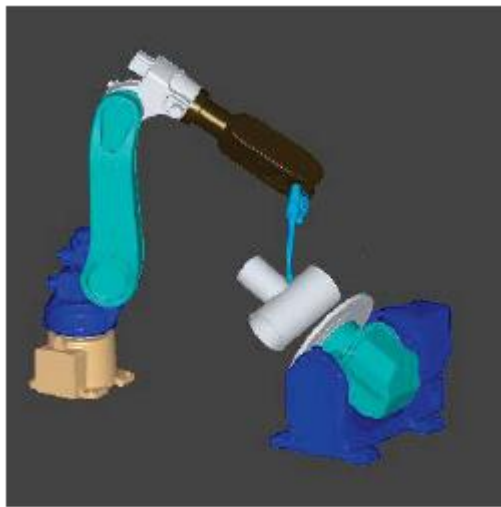


5 pav. Simuliacinės programos diagrama

2.1.3 Rezultatai

Naudojant OCC biblioteką ir "Visual Studio" buvo sukurta suvirinimo programavimo modeliavimo platforma, pagrįsta CAD programomis. Siūloma sistema gali importuoti 6-DOF ir 2-DOF robotų CAD modelius norimai užduočiai atlikti, kaip parodyta 6 pav. Kelio planavimo modulis paima informaciją iš CAD, kad sukurtų suvirinimo kelią su tolygiu mazgų pasiskirstymu ant pasirinktos kreivės. Be to, suvirinimo poza apskaičiuojama nustatant normalinį vektorius tarp dviejų paviršių. Sugeneruoto suvirinimo kelio rezultatai rodo, kad kreivės mazgai sukuriama tolygiai ir gaunama 6-DOF roboto pozicijos erdvėje skirtos suvirinimo užduočiai atlikti, kaip parodyta 6 pav. Kelio planavimo modulis gali sukurti kelią be susidūrimų.

Suvirinimo degiklio orientacija apibrėžiama pagal sukimosi matricą, gautą iš ruošinio santykinės padėties, kad būtų išlaikyta normalė aukštyrų išilgai z ašimi. Ši orientacija padeda 6-DOF robotui išlaikyti suvirinimo užduočiai atlikti reikalingą padėtį, kaip parodyta 6 pav.). Apskaičiavus naują padėtį su 2-DOF roboto sąnariais. Pagal atvirkštinę kinematiką apskaičiuojamas kiekvieno 6-DOF roboto sąnario kampas. Galiausiai sukurtoje OLP platformoje atlikta pilna pasirinkto suvirinimo kelio simuliacija, siekiant praktiškai patikrinti rezultatus, kaip parodyta 6 pav. Be to, į šią sukurta sistemą galima įkelti įvairius ruošinius, kad būtų galima atlikti bet kokią suvirinimo užduotį.



6 pav. Virtualus programos vaizdas

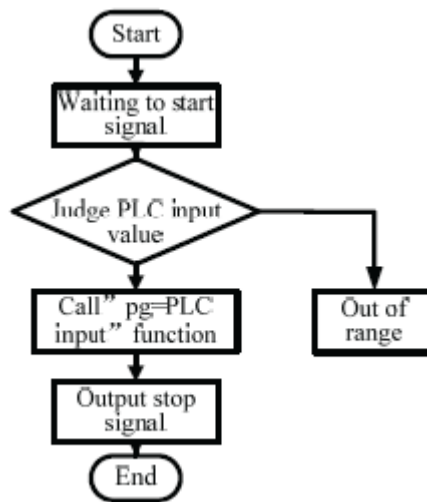
2.2 Laivo korpuso suvirinimas ir siūlės paieška su lietimųjų sensorių

2.2.1 Išmaniosios suvirinimo sistemos konstrukcija



7 pav. Laivo korpuso dalys

Laivo korpuso konstrukcinių dalių (pvz., 7 pav.) ypatybės yra: didelis detalių dydis, ilgas apdorojimo laikas ir surinkimo tikslumas, todėl sunku užtikrinti automatinę suvirinimo gamybą[2]. Transporto linija gabenamą detalę aptinka fotoelektrinis jutiklis, o pakartotinis padėties aptikimo tikslumas yra labai mažas. Todėl bendras gamybos linijos aptikimo tikslumas yra mažas, tai neabejotinai sukels daug problemų pramoniniam robotui, kurio darbas priklauso nuo lietimųjų sensoriaus tikslumo. Todėl robotas negali garantuoti konstrukcinių dalių suvirinimo kokybes, naudojant paprastą automatizavimo technologiją. Tikimasi, kad esama automatinė robotų gamybos linija papildžius išmaniuoju jutikliu, bus išspręstos pirmiau paminėtos problemos. Šiame darbe buvo suprojektuota korpuso konstrukcinių dalių lankinio suvirinimo robotų gamybos linija, kurioje taikomas lietimųjų jutiklis, jos valdymo sistemos pagrindinė darbo eigos programa parodyta 8 pav.



8 pav. Pagrindinės programos diagrama

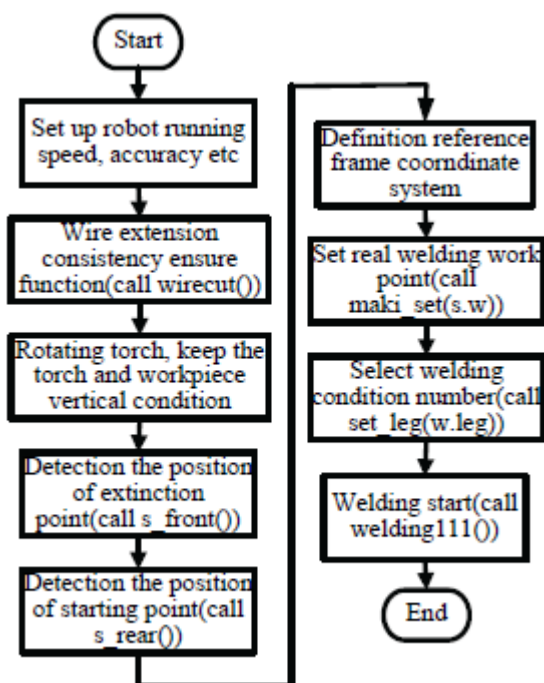
Pirma, transportavimo linija pristato detales į paskirtą vietą, pagal detalės modelį, PLC suteikia robotui proceso numerį, tada robotas atlieka suvirinimo darbus naudodamas lietimą jutiklį, kai suvirinimas būna baigtas, robotas siunčia sustabdymo signalą į PLC, transportavimo linija pristato baigtas suvirinimo detales į paruoštą ruošinių vietą ir laukia tolesnio naudojimo.

"Kawasaki" suvirinimo roboto lietimą vietos jutiklis yra realizuojamas naudojant techniką ir programinę įrangą; įranga turi užtikrinti pastovią įtampą (šiuo straipsnyje pasirinkta DC15 V įtampa), kad būtų aptiktas vielos galo ir ruošinio kontaktas. Programinė įranga susideda iš dviejų dalių: viena jų yra roboto valdymo programa, kuri naudojama roboto veiksmams valdyti ir robotui mokytį, kita - proceso valdymo programa, kuri naudojama papildomoms funkcijoms atlikti ir procesui valdyti. Proceso valdymo programa ir roboto valdymo programa naudoja bendrus kintamuosius ir vidinius signalus ryšiui palaikyti, visiškai kontroliuoja robotą ir periferinę įrangą. Roboto valdymo programoje saugomos iš anksto paruošto kelio mokymo, BASE ir TOOL koordinatų reikšmės, proceso valdymo programoje saugoma AS kalba parašyta lietimą jutiklio vietos nustatymo metodo programinė įranga.

Litimą jutiklio vietos nustatymo programa turi remtis iš anksto išmokytais roboto kelio koordinatų taškais. Surandama suvirinimo siūlė ir apskaičiuojamas nuokrypis tarp faktinės vertės ir iš anksto nustatytos vertės, planuojamas faktinis suvirinimo darbo kelias.

2.2.2 Lietimo jutiklio programos struktūra

Pagal ruošinio modelį, kurį atveža transportavimo linija, PLC perduoda atitinkamą programos numerį robotui, robotas pagal programos numerį pradeda suvirinimo programą, 9 pav. parodyta ruošinio suvirinimo procedūros programinės įrangos schema:



9 pav. Virinimo procedūros programos diagrama

Pirma, paruošiamieji darbai, susiję su jutiklio vietos nustatymo metodu: pagal ruošinio suvirinimo procesą nustatomas roboto važiavimo greitis ir tikslumas suvirinimo darbo procedūroje. Tada įjungiamas vielos pailginimo užtikrinanti funkciją `wirecut()`, jog suvirinimo sistemos vielos pailginimas atitiktų pastovias vertes prieš kiekvieną prisilietimo vietą. O tada naudojame įrankio koordinatinių sistemos Z ašį suvirinimo degiklio priekyje į vertikalią bazinės koordinatės OXY plokštumos kryptį per Eulerio koordinatinių transformaciją. Du taškai ant suvirinimo siūlės naudojami faktualios suvirinimo siūlės padėties erdvinei koordinatei nustatyti. Du taškai ant ruošinio krašto naudojami pradinio lanko taško ir galinio lanko taško erdvinėms koordinatėms apskaičiuoti; pabaigoje apskaičiuojama mokymo taškų ir faktinių darbo taškų nuokrypio reikšmė, roboto padėtis paslenkama, sukuriamas suvirinimo procesas, užbaigiama suvirinimo procedūra, apibrėžiama pagrindo rėmo ir atskaitos rėmo koordinatė, kuri naudojama atitinkamam numatytu taškų ir faktinių darbo taškų santykiui apskaičiuoti. Apskaičiuojama mokymo taškų ir faktinių darbo taškų nuokrypio reikšmė. Iškviečiama funkcija `maki_set(S.w)`,

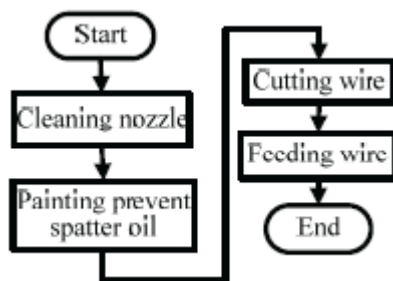
kuri naudojama numatytiems taškams pakeisti į tikruosius darbo taškus atmintyje, numatytų taškų erdvinės koordinatės nekeičiamos, jos bus pakartotinai naudojamos atliekant kitą darbo procedūrą.

2.2.3 Programos funkcijų kūrimas

Darbo vietoje naudojamos vielos skersmuo yra 1,4 mm, remiantis empirine formule, atliekant faktinę suvirinimo procedūrą vielos ilgis turėtų būti 17 mm. Atsižvelgiant į lankinio suvirinimo gamybos reikalavimus, kiekviena suvirinimo procedūra pirmiausia turi nupjauti vielą, dėl mechaninio pjovimo būdo vielos nuokrypis gali būti keičiamas. Šiame darbe buvo naudotas 5 mm iš antgalio vielos ilgis. Buvo suprojektuota funkcijų tikrinimo programinė įranga, kurią daugiausia sudaro vielos pratęsimo nuoseklumo užtikrinimo funkcijos modelis, krašto aptikimo funkcijos modelis ir ruošinio nuokrypio kompensavimo funkcijos modelis. Laido pratęsimo nuoseklumo užtikrinimo funkcijos modelis, kurį sudaro degiklio antgalio valymo, vielos pjovimo ir vielos pratęsimo nuoseklumo užtikrinimo funkciją; ruošinio suvirinimo siūlės ir krašto aptikimo funkcijos modelis, kurį sudaro suvirinimo siūlės ir ruošinio krašto erdvinės koordinatės nustatymo funkciją; ruošinio nuokrypio kompensavimo funkcijos modelis, kurį sudaro kompensavimo poslinkio, numatytų taškų erdvinės koordinatės nuokrypio ir faktinio darbo taško funkcijos ištaisymo funkciją.

A. Vielos iškišimo nuoseklumą užtikrinti funkciją:

Visoje sistemoje grandinės įtampa naudojama nustatyti, ar vielos galas liečiasi su ruošiniu, o tikrosios suvirinimo siūlės vieta apskaičiuojama pagal sąlyčio taškų erdvinę koordinatę. Vielos atmatavimas turėtų būti atliekamas prieš kiekvieną prisilietimo vietos aptikimą, nes kiekvienas sąlytis tarp vielos galo ir gaminio. 10 pav. pateikta programinės įrangos schema užtikrina vielos pailginimo nuoseklumą prieš aptikimą.

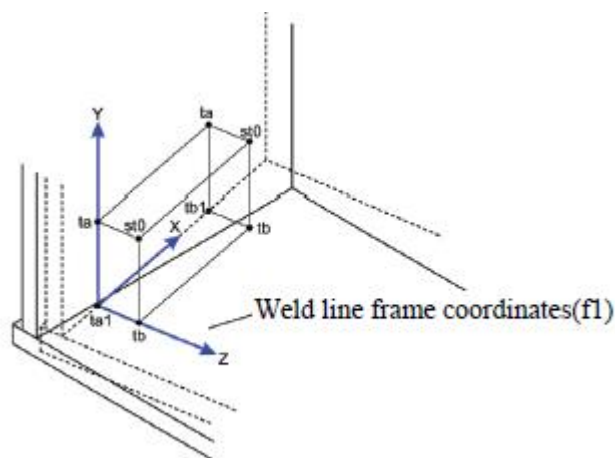


10 pav. Vielos ištraukimo diagrama

Suvirinimo metu susidarę purslai gali užteršti antgalį, taip pat gali sudaryti elektros grandinę kartu su suvirinimo viela. Roboto veiksmai, kuriuos valdo funkcija `wirecut()`, įgyvendina antgalio valymą. Pirmiausia robotas perkėlė degiklį į fiksuotą padėtį ir nusiuntė signalą, kad būtų išvalytas antgalis. Po valymo antgalis bus padengtas alyva, kad neleistų prilipti purslams. Antra, suvirinimo viela ištempinama 5 mm automatiškai būdu, kad būtų patogiau nupjauti mažą rutuliuką vielos gale. Viela buvo trumpai nukirpta, kad būtų išvengta lenkimo, o tai pagerino vielos pratęsimo nuoseklumą. Galiausiai servo valdymo sistema tolygiai paduoda vielą 0,5 mm/s greičiu 10 s, kad užtikrintų, jog viela būtų pailginta 17 mm.

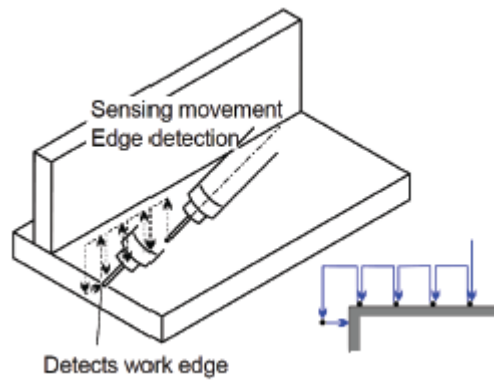
B. Ruošinio suvirinimo siūlės ir kraštų aptikimo funkcija:

Detalės suvirinimo siūlė ir briauna buvo aptiktos pagal `s_front()` ir `s_rear()` funkcijas. Funkcija `s_front()` buvo naudojama suvirinimo siūlės taškui ir vienai ruošinio briaunos pusei aptikti, o funkcija `s_rear()` - kitam suvirinimo siūlės taškui ir kitai ruošinio briaunos pusei aptikti. Šie du taškai nustatė tiesę, kuri sutapo su suvirinimo siūle, o funkcijomis nustatytos briaunos padėjo nustatyti lanko pradžios taško ir lanko pabaigos taško padėtį. Suvirinimo siūlės padėtis yra konkreči.

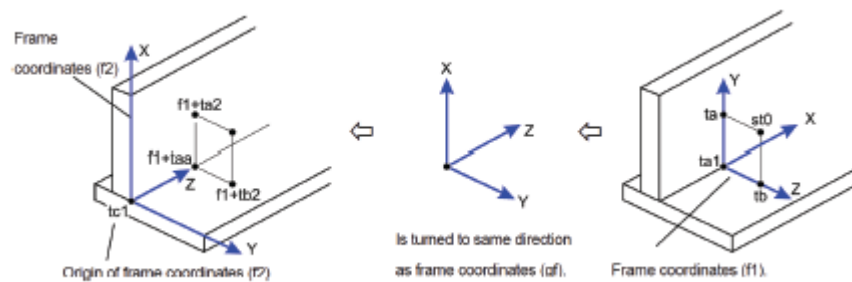


11 pav. Ruošinio poslinkio diagrama

11 pav. pateikta ruošinio poslinkio diagrama, kuri apima lygiagrečių poslinkio ir posūkio poslinkio diagramas. Dviejų taškų, kurie buvo aptikti taikant lietimvietos nustatymo metodą, koordinatės yra saugomos funkcijos `s_front()` kintamajame `tb` ir funkcijos `s_rear()` kintamajame `ta`. Tikrieji suvirinimo siūlės padėties taškai patvirtinami kaip `tb` ir `ta`. Tada buvo apibrėžta koordinačių sistema `fl`, kurios X ašis yra išilgai suvirinimo siūlės.



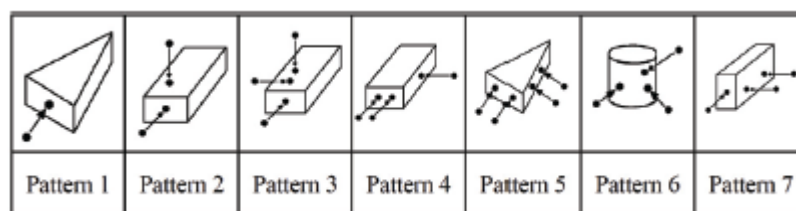
12 pav. Ruošinio krašto aptikimo sistema



13 pav. Koordinatinių sistemų transformacija

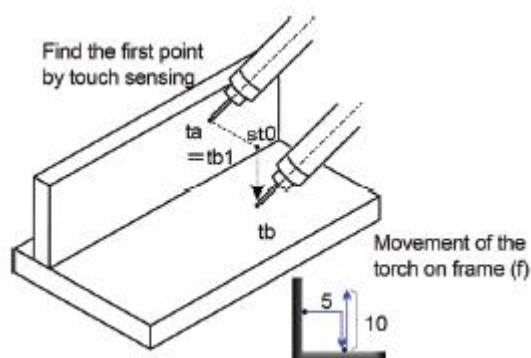
12 pav. parodytas kraštų aptikimo procesas. Abi aptiktos ruošinio briaunos projektuojamos kaip du kraštiniai taškai f_1 koordinačių sistemos X ašyje, o kraštiniai taškai perkeliama atitinkamai 5 mm į vidurį, kad būtų galima apibrėžti naujo pradžia lanko taško ir pabaigos lanko taško koordinatas. Koordinačių sistemos f_1 pradžia perkelta į ruošinio kraštą tc_1 , o f_1 kryptis pasukta į atskaitos rėmo koordinačių sistemos gf kryptį. Ši nauja koordinačių sistema buvo apibrėžta kaip f_2 . Tada buvo galima apskaičiuoti pradžia lanko taško ir galinio lanko taško padėties koordinatas. 13 pav. parodyta koordinačių transformacijos schema. Pirmiau aprašytu būdu buvo sukurtas santykis tarp faktinio pradžia lanko taško ir galinio lanko taško padėties bei atskaitos rėmo koordinačių sistemos, todėl bandomieji taškai ir numatyti taškai buvo konvertuoti į tą pačią koordinačių sistemą. Tad, galima užtikrinti tikslų suvirinimo procesą, net jei ruošinio padėtis nesutampa su numatyta padėtimi.

C. Ruošinio nuokrypio kompensavimo funkcija:



14 pav. Aptikimo moduliai

"Kawasaki" roboto valdiklyje yra septynių rūšių kompensavimo programinės įrangos modeliai, skirti galimam ruošinio poslinkiams analizuoti. Buvo pasirinktas septintasis (šoninė plokštelė) ir ketvirtasis (apatinė plokštelė), kad būtų sudarytas kombinuotas mokymo modelis. Santykis tarp dalių turėtų būti apibrėžtas pagal apdirbimo reikalavimus (viena yra tėvo dalis, o kita - sūnaus dalis). Tėvo dalies, kuri buvo atskaitos rėmo koordinatinių sistemos poslinkis, faktinės koordinatės turėtų būti apskaičiuotos pagrindinėje koordinatinių sistemoje. Tada turėtų būti atliktas tėvo dalies ir sūnaus dalies poslinkis, o sūnaus dalies faktinės koordinatės būtų projektuojamos į atskaitos sistemos koordinatinių sistemą. Visos detalės apdirbimo padėties poslinkis buvo kompensuojamas apskaičiuojant faktinės koordinatės ir mokymo koordinatės skirtingą vertę atskaitos rėmo koordinatinių sistemoje.



15 pav. Aptikimo sekos diagrama

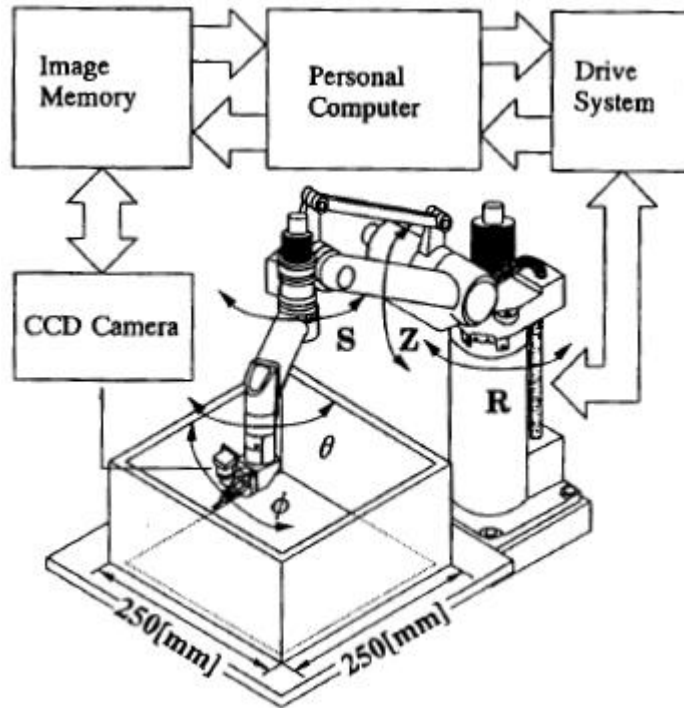


16 pav. Suvirinimo siūlė

15 pav. pateikta aptikimo sekos schema. Apatinė plokštė buvo apibrėžta kaip tėvo dalis, o šoninė plokštė - kaip sūnaus dalis. Prieš suvirinimą tėvo dalis ir sūnaus dalis turėtų būti paeiliui apskaičiuotos. Roboto valdiklis užbaigė lietimą vietos nustatymo metodo lankinį suvirinimą bendradarbiaujant robotų valdymo programai ir proceso valdymo programai. 16 pav. pavaizduotas suvirinimo siūlė.

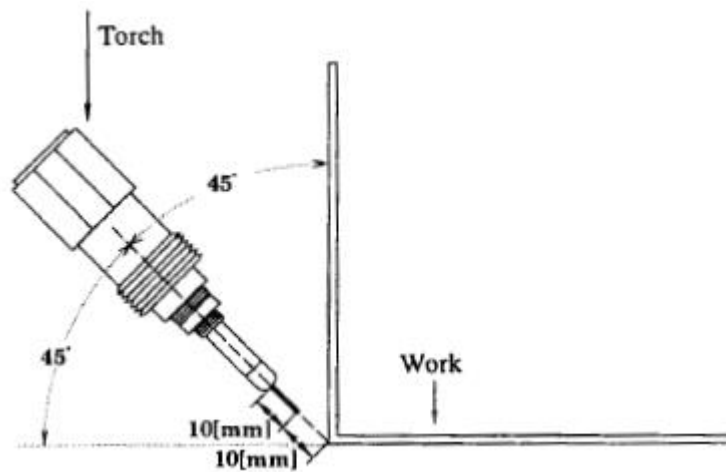
2.3 Siūlių sekimas naudojant kelis sensorius

2.3.1 Jutiklis su kamera

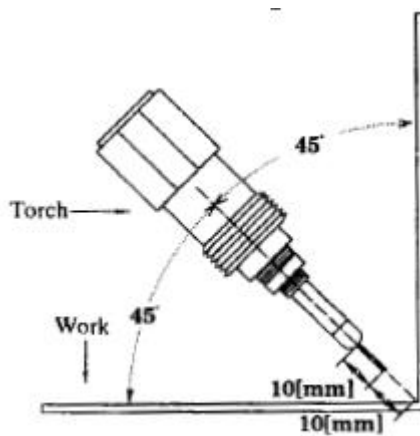


17 pav. Roboto sistema

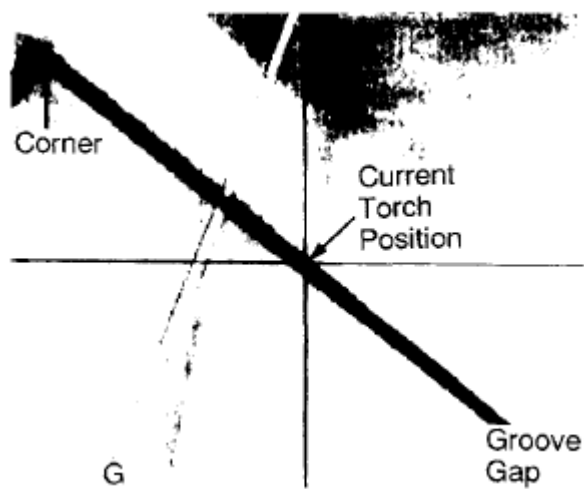
CCD kamera, pavaizduota 17 pav., yra pritvirtinta prie degiklio. Degiklio, 8 ašies ir darbo padėties prie komutatoriaus santykis parodytas 18 ir 19 pav. [3]. Kompiuteris jutikliu išmatuoja darbo formą, tam naudojamas elektrodas, kuris nustato padėtį, kurioje reikia pradėti sukuti nuline ašį. Degiklio padėtis yra pasvirusi iki 45" kampu. CCD kamera užfiksuoja griovelio tarpą, tipinis vaizdas parodytas 20 pav. G yra jutimo linija. Kompiuteris apdoroja vaizdą ir nustato degiklio judėjimo kryptį. Tikroji degiklio padėtis gaunama pagal judėjimo kryptį ir judėjimo atstumo laikotarpį. Kiekvieno sąnario kampas apskaičiuojamas pagal tikslią degiklio padėtį naudojant atvirkštinę kinematiką. Apskaičiuojamas impulsų skaičius. Kompiuteris judina degiklį siųsdamas impulsą robotui.



18 pav. Santykis tarp degiklio ir ruošinio kai yra sekama ruošinio išorė



19 pav. Santykis tarp degiklio ir ruošinio kai yra sekama ruošinio vidus



20 pav. Kameros fiksuojamas vaizdas

2.3.2 Optimalaus degiklio aukščio nustatymas

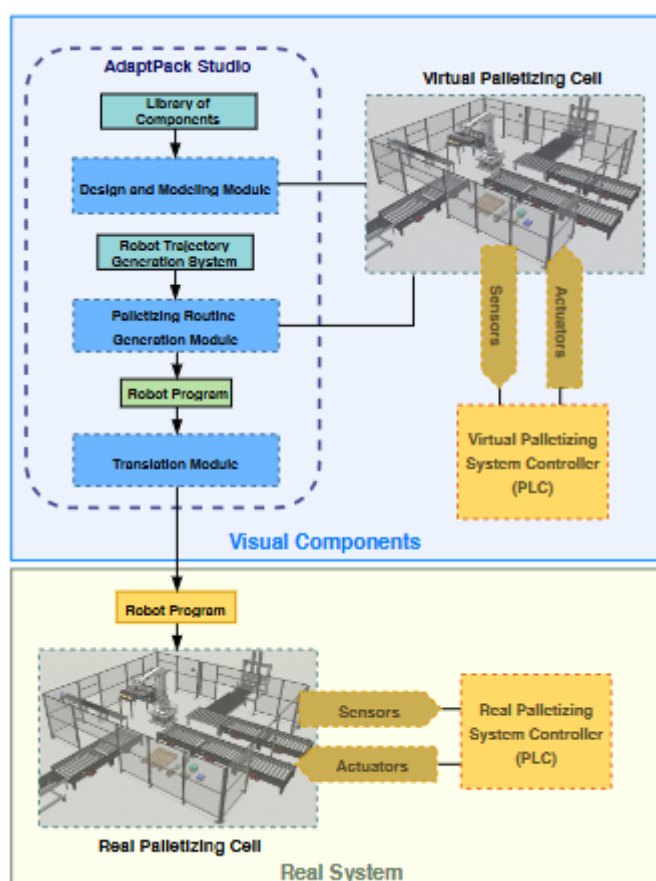
Degiklio nulinė ašis turi būti pasukta nuo 0" iki 90" ties kraštine, kad degiklio padėtis orbitos atžvilgiu išliktų vertikali. Pageidautina, kad virinimo metu būtų laikomasi šių sąlygų:

- a) degiklis išlaikomas vertikaliai orbitos atžvilgiu;
- b) kampinis greitis turi būti mažas;
- c) degiklis nepatektų į susidūrimą su ruošiniu;
- d) kampinis pagreitis būna mažas

2.4 Adaptpack „offline“ programavimas

2.4.1 Siulomas sprendimas

Vienas iš pirmųjų žingsnių kuriant naują paletavimo sistemą yra celės projektavimas ir modeliavimas „ant popieriaus“ arba virtualioje sistemoje [4]. Sumodeliuoti šią sistemą nuo nulio reikia įdėti daug darbo, reikia pasirinkti ir sumodeliuoti norimus komponentus. Siekiant paspartinti modeliavimo procesą, buvo sukurtas projektas, kuris turėjo bendrųjų komponentų modelius, kurie yra naudojami padėkluose paletavimo sistemose. Tie komponentai buvo įtraukti į VC biblioteką, pavadinta „AdaptPack Building Blocks“, kad sudarytų „Dizaino ir modeliavimo modulį“. Šis modulis leidžia vartotojams nuvilkti ir numesti komponentus modelio makete ir sujungti juos kaip blokus.



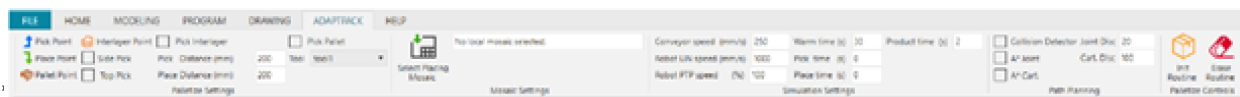
21 pav. Sistemos architektūros diagrama

Sumodeliavę celę, naudotojai gali naudotis grafiniu vartotojo sąsaja (GUI), kad būtų galima programuoti robotus „Offline“. Ši grafinė sąsaja buvo sukurta naudojant C# taikomąją programą programavimo sąsaja (API), kurią pateikė VC, suteikiant galimybę programuoti robotą

naudojant pultelio ir neparašant nė vienos kodo eilutės. Norint sukongūruoti paėmimo ir padėjimo taškų nustatymus, buvo sukurti grafinės sąsajos mygtukai ir teksto įvedimo laukai, skirti įrašyti konfigurācijas. Roboto kodas yra sukuriamas automatiškai nustačius visus kintamuosius. Šis procesas sudaro "Paletavimo modulį", kuris savo ruožtu keisis informacija su "Roboto trajektorijos generavimo sistema", kaip parodyta paveikslėlyje 1 pav. Trajektorijos generavimo sistemos tikslas – užprogramuoti roboto judėjimą tarp paėmimo ir padėjimo taškų, atsižvelgiant į operācijas modeliujamoje celėje, taip, kad būtų sukurtos trajektorijos be susidūrimų.

3D modeliavimas yra naudojamas sistemai patvirtinti. Modeliuojant roboto darbą ir paletavimo sistemą virtualioje aplinkoje, gali būti gerokai padidintas roboto efektyvumas. Taip pat, atsižvelgiant į tikslumą ir detalumo lygį, kurį siūlo virtuali aplinka, galima gauti modelius, kurių elgsena yra labai panaši į realią įrangą; tai leidžia tirti ir bandyti patikimą sprendimą prieš jį įgyvendinant. Parašytas kodas VC programine kalba (Robot simulation language - RSL), yra išverčiamas į skirtingų robotų valdiklių naudojamus kodus. (ABB, KUKA, FANUC ir Yaskawa).

2.4.2 Sistemos architektura



22 pav. Sistemos grafinė sąsaja

"AdaptPack" studija yra suskirstyta į tris modulius, iš kurių kiekvienas atsakingas už atitinkamos problemos tvarkymą. Daugiausia dėmesio yra skiriama paletavimo moduliui. VC sąsaja suskirstyta į skirtukus, kurių kiekvienas atsakingas už konkrečią užduotį, pavyzdžiui, maketo sudarymas, modeliavimas / redagavimas, programavimas ir kt. Naujas skirtukas, pavadintas "ADAPTPACK" buvo įterptas tarp esamų skirtukų, 22 pav., kuriame vaizduojamas paletavimo modulis. Šis naujas skirtukas suskirstytas į pogrupius, kurie leidžia intuityviai konfiguruoti modeliavimą. Šie pogrupiai, taip pat ir jų funkcijos, yra šie:

- Padėklų nustatymai:

Šiame pogrupyje yra rinkinys mygtukų ir teksto laukų, kuriuose apibrėžiami paletavimo nustatymai. Stalo taškai pasirenkami spustelėjus atitinkamą 3D komponentą, kuriame yra produktai, padėklai ir stalai. Vietos taškas pasirenkamas paspaudus ant komponento, apibrėžto kaip padėjimo vieta. Kadangi griebtuvas gali turėti kelis įrankių centro taškus (TCP), taip pat

pateikiamas meniu su visais roboto įrankiais su šia savybe. Be to, yra parinkti nustatanti metodika, pagal kurią apibrėžiama paėmimo ir padėjimo, bei priartėjimo taškų nustatymo metodika, suderintą su naudojamu griebtuvu rūšimi. Griebtuvai, kurie turi šoninį priartėjimą (kai naudojami dantyti griebtuvai), arba tie, kurie turi viršutinį (pavyzdžiui, siurbiamieji griebtuvai), yra apsvarstomi programos.

- Mozaikos nustatymai:

Mozaika yra kaip objektai bus sudėlioti ant padėklo. Šiame pogrupyje operatorius gali pasirinkti įvesties failą, kuriame yra padėklų sudėjimo nustatymai. Šis failas, parašytas JSON scenarijų kalba, yra generuojamas automatiškai išorinės programinės įrangos, atsižvelgiant į nustatyto padėklo ir gaminio matmenis, skaičių ir kt. Importuotą failą sudaro struktūros, nustatančios eiliškumą ir objektų padėtis padėklo erdvėje. Bendro failo struktūra yra suskirstyta pagal sluoksnius, o kiekviename sluoksnyje yra viena ar daugiau grupių, o kiekvienoje grupėje yra vienas ar daugiau gaminių. Todėl robotas vienu metu gali vienu metu tvarkyti vieną ar kelis gaminius.

- Modeliavimo nustatymai:

Šioje srityje galima atlikti modeliavimo aplinkos nustatymus, pavyzdžiui, roboto ir konvejerio greitį, sustojimo laiką, produkto gamybos greitį ir laiką.

- Kelio planavimas:

Šioje srityje yra mygtukai, kuriais galima įjungti "Roboto trajektorijos generavimo sistemą", pagrįstą paieškos algoritmu, sukurtu [16]. Be diskretizacijos sąnarių erdvėje [14], čia taip buvo naudojama „Cartesian space“. Šis naujas metodas leido sumažinti skaičiavimo laiką ir trajektorijos išlyginimą.

Atlikus visas konfigūracijas, sistema yra paruošta apskaičiuoti automatine paletizavimo procedūrą. Įjungus mygtuką "Init Routine" (Inicijuoti rutiną), sukuriama "Palletize.cs" konfigūracijos gaunamos iš grafinės sąsajos ir pasirinkto JSON failo, kaip parodyta 23 pav.

```

void Init():

    get simulation settings from GUI
    run simulation for x seconds
    WhenSimulationStops()

void WhenSimulationStops():

    ## Settings
    get info from selected components in GUI
    get picking and placing distance values
    get info about products and pallets
    get components dimensions
    get transformation matrices/pick points

    define tools to be used
    define robot base frames

    ## Robot program
    if(handlePallet enabled):
        define pallet picking and placing routine

    if(handleInterlayers enabled):
        define interlayers picking and placing routine

    define subroutines for controlling signals

    ## Path Planner routine
    if(A* enabled):
        call PathPlanning.cs class
        for(each point in returned trajectory):
            generate path routine

    ## Picking and placing routine
    for(each layer in json file):
        for(each group):
            get tool
            define pick and place points

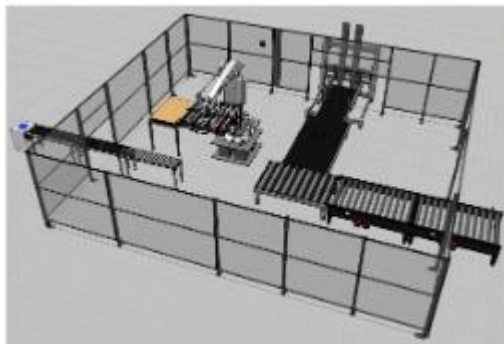
```

23 pav. Paletavimo kodas

Inicijavus programą, modulis imituoja komponentų (dėžių ir padėklų) judėjimą konvejeriais, kol jie pasiekia konkrečius paėmimo taškus. Po procedūros, išrenkami kintamieji apie 3D centrą ir apskaičiuojami tiksli paėmimo vieta. Taip pat išgaunama informacija apie aukštį, ilgį, plotį ir visų komponentų padėties ir posūkio matricas. Pagal mozaika generuojama padėtis, sistema interpretuoja JSON failo turinį ir apskaičiuoja taškus, į kuriuos reikia įdėti gaminius. Ši interpretacija atliekama pakartotinai, kol kiekviena grupė turi savo vietą modeliavime.

2.4.3 Bandymai ir rezultatai

Buvo sumodeliuota tipinė paletavimo celė, kad būtų galima patikrinti sistemos įgyvendinamumą 24 pav. Šioje celeje yra roboto ranka su specialiai pritaikytu griebtuvu, galinčiu tvarkyti pavienius arba kelis objektus. Papildoma įranga, pavyzdžiui, aptvarai, konvejeriai, ir tiekuvai buvo įtraukti, kad modeliavimas galėtų veikti tikroviškai ir parodyti visas programinės įrangos galimybes imituojant realias gamyklų operacijas.

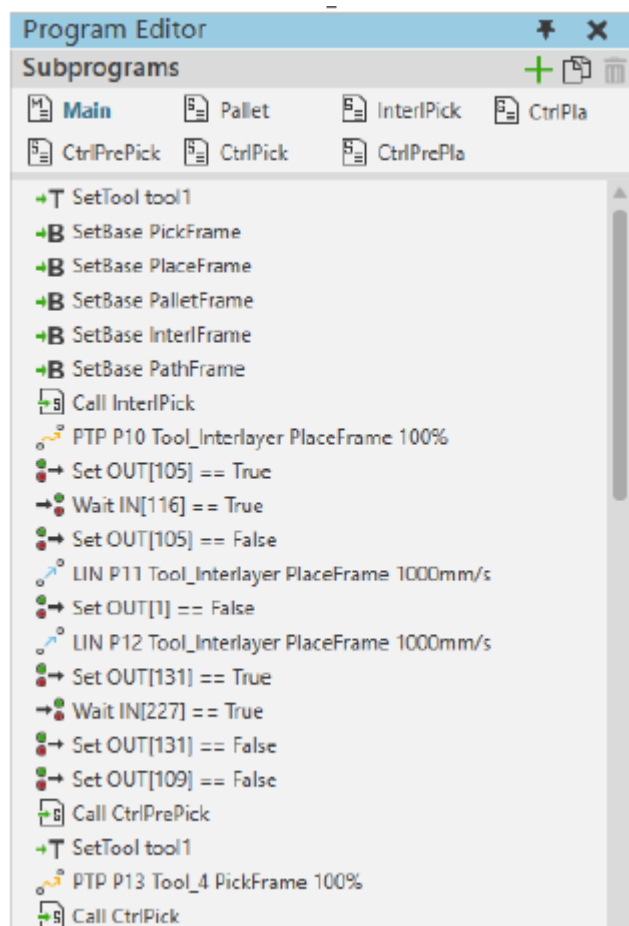


24 pav. Grafinė paletizavimo simuliacija

1) Paletizavimo procedūros generavimo modulis:

Naudotojas turi sukonfigūruoti modulį naudodamasis grafine vartotojo sąsaja ir tada paspausti mygtuką "Init Routine" (Inicijuoti rutiną). Pav 6 pavaizduotas konvejeris, iš kurio paimami produktai. (kairėje), stalas (viršuje kairėje) ir konvejeris, ant kurio dedami produktai (dešinėje). Reikia atkreipti dėmesį, kad keli paėmimo taškai yra vienas šalia kito, jie buvo sugeneruoti pagal grupinio tvarkymo sistemą. Kartais robotas turi paimti grupę, kurioje yra vienas produktas, o kartais - keletas. Tokiu būdu buvo apskaičiuoti skirtingi surinkimo taškai pagal esamos gaminių grupės centrą.

Įrankiai ir rėmas yra apibrėžiami, po to apibrėžiamos padėklų ir stalų procedūros. Paėmimo ir išdėstymo funkcijos buvo suskirstytos į paprogrames, kad būtų galima sekti roboto atliekamus veiksmus, kad jis visada judėtų žingsniais. Signalai buvo apibrėžti taip, kad veiktų palei imituojamąjį PLC. Šis komponentas veikia kaip valdytojas visai celei. Pavyzdžiui, prieš funkciją Pick, robotas bendrauja su PLC, kad gautų leidimą tęsti operaciją. PLC leidžia robotui paimti gaminius tik tuo atveju, jei komplektavimo konvejeriuje yra gaminių paruoštų paimti. 25 pav. yra parodyta sugeneruotos procedūros ištrauka. Šiame paveikslėlyje pavaizduotas programos redaktoriaus langas, kuriame deklaruojama pagrindinė programa ir paprogramės. Viduje pagrindinė programa, roboto įrankis ir rėmas yra apibrėžiami, po to seka stalo tvarkymo paprogramė ir paletavimo scenarijus.



25 pav. Sugeneruotas programos kodas

2) Roboto trajektorijos generavimo sistema:

Trajektorijos skaičiavimo sistema yra papildomas išteklius „AdaptPack“ studijoje. Be jo įjungimo robotas atlieka tiesiaieigį judesį nuo paėmimo iki padėjimo taško. Todėl „Roboto Trajektorijos generavimo sistema“ yra tinkamai naudojama, kai celėje yra griežtos ribos, yra susidūrimų galimybė arba būtina atlikti trajektorijos skaičiavimus ir analizę.

3. Teorine dalys

3.1 Inostart projektas

Inostart – tai programa leidžianti studentams bendradarbiauti su įmonėmis sprendžiant įvairius su inžinerija susijusius projektavimo klausimus. UAB „ATC Baltic“ įmonės pageidavimu buvo sukurtas lietimio sensorius (angl. Touch sensor) suvirinimo siūlėms rasti; metalo pjovimo su plazmos lanku aukščio kontrolės nustatymo prietaisas.

Įmonė UAB „ATC Baltic“ specializuojasi pramoninių robotų integravimu metalo apdirbimo gamybos procesuose. Pagrindinės sritys, robotinės suvirinimo, plazminio pjovimo sistemos, gaminių paletavimas.

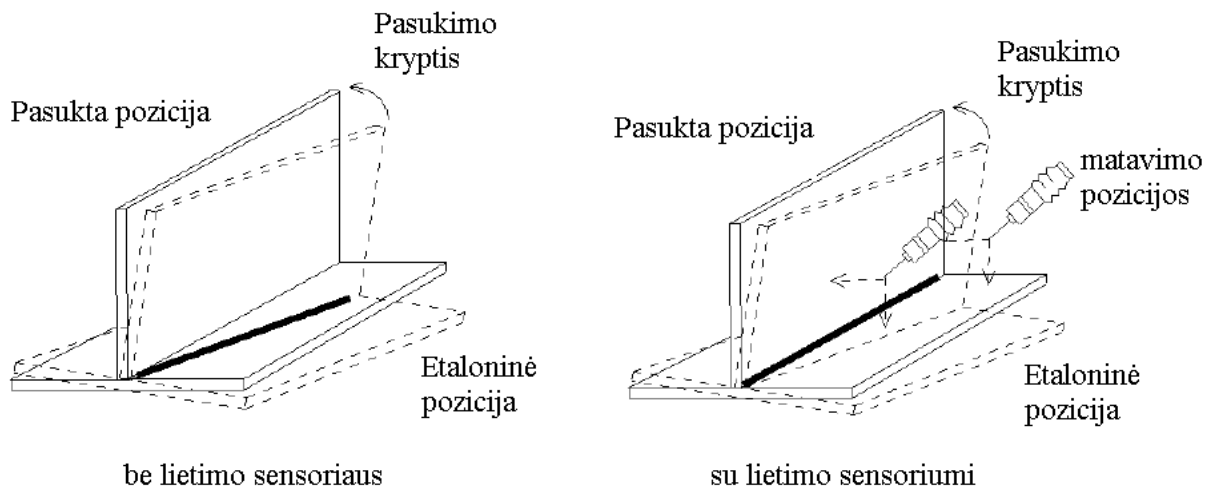
Suvirinimo robotinėse sistemose būtina palaikyti tikslią ruošinių geometriją, t.y. ruošinių matmenys turi būti kiek galima tikslesni ir mažai keistis pakartotiniuose gaminiuose. Tačiau realioje gamyboje ne visada tai pavyksta pasiekti dėl gamybinių staklių netobulumo arba dėl žmogiškojo faktoriaus, kurie kai kuriose gamybinėse operacijos yra neišvengiami. Todėl būtina prie gaminių matmenų išsibarstymo prisitaikyti naudojant papildomas priemones. Viena iš alternatyvų išmatuoti gaminių matmenų paklaidas yra naudoti lietimio sensorių (angl. *Touch sensor*). Šiuo siūlomu metodu robotas suvirinimo degiklio pagalba išsimatuoja gamino realią poziciją erdvėje ir įvertina paklaidas. Pagal išmatuotas paklaidas robotas pakoreguoja suvirinimo programą ir tokių būdu gaminį suvirina teisingai.

Tai pat įmonė UAB „ATC Baltic“ kartu su lietimio sensoriumi pageidavo suprojektuoti ir pagaminti plazminio pjovimo robotinei sistemai plazmos lankinio išlydžio aukščio kontrolės sistemą. Pjaunant metalą plazminio išlydžio metodu, pjūvio kokybė priklauso nuo oro tarpelio tarp degiklio ir pjaunamo ruošinio. Jeigu tarpas yra per mažas – pjūvio vietoje formuojasi daug šlako ir pjūvis tampa nekokybiškas. Jeigu tarpas yra per didelis – pjūvis tampa platus, todėl išpjautas ruošinys praranda apibrėžtus matmenis ir nėra tinkamas tolimesnėms gamybinėms operacijoms. Ši problema yra aktuali plazminio metalo pjovimo sistemose, todėl nuspręsta suprojektuoti ir sukurti plazmos lankinio išlydžio aukščio kontrolės sistemą.

3.1.1 Lietimo sensoriaus veikimas

Lietimo sensoriaus veikimas pagrįstas gaminio etaloniniu taškų paieška erdvėje (26 pav.) Pirmiausia robotas apmokomas susirasti etaloninės ruošinio pozicijos paieškų taškus. Šių

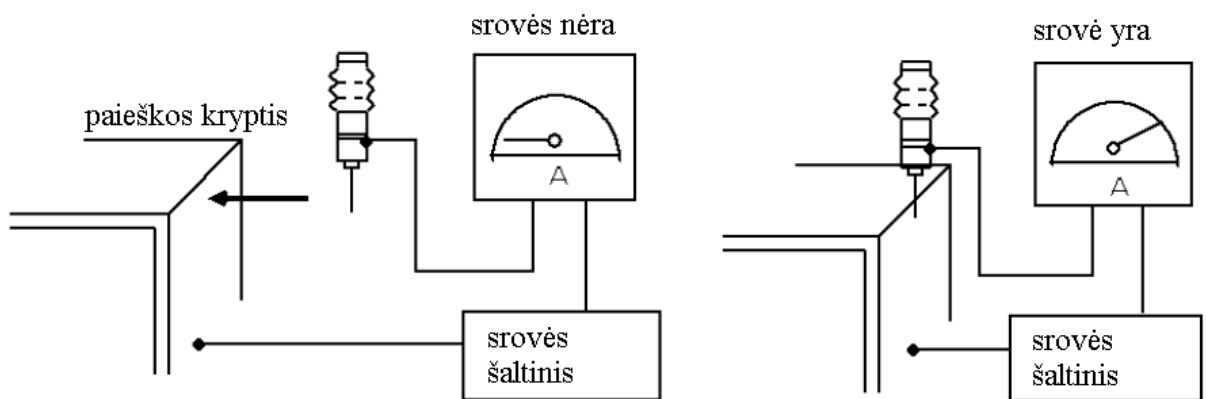
taškų pagrindu robotas matuojasi paklaidas – kiek nesutampa etaloninio ruošinio ir testuojamo ruošinio pozicijos. Gautomis paklaidomis robotas pakoreguoja suvirinimo programos trajektoriją pagal matuojamo ruošinio realią poziciją.



26 pav. Lietimo sensoriaus veikimas

Jeigu gamyboje gana ženkliai skiriasi ruošinių matmenys, lietimo sensorių naudoti būtina, nes kitaip robotinė sistema tinkamai nesuvirins gaminio.

Kaip privalumas šios sistemos yra paprastas ir pigus būdas atlikti suvirinimo programos adaptavimą netiksliems ruošiniams suvirinti. Trūkumas – ruošinio aptikimas reikalauja papildomo laiko, kurio metu gali būti virinami ruošiniai.

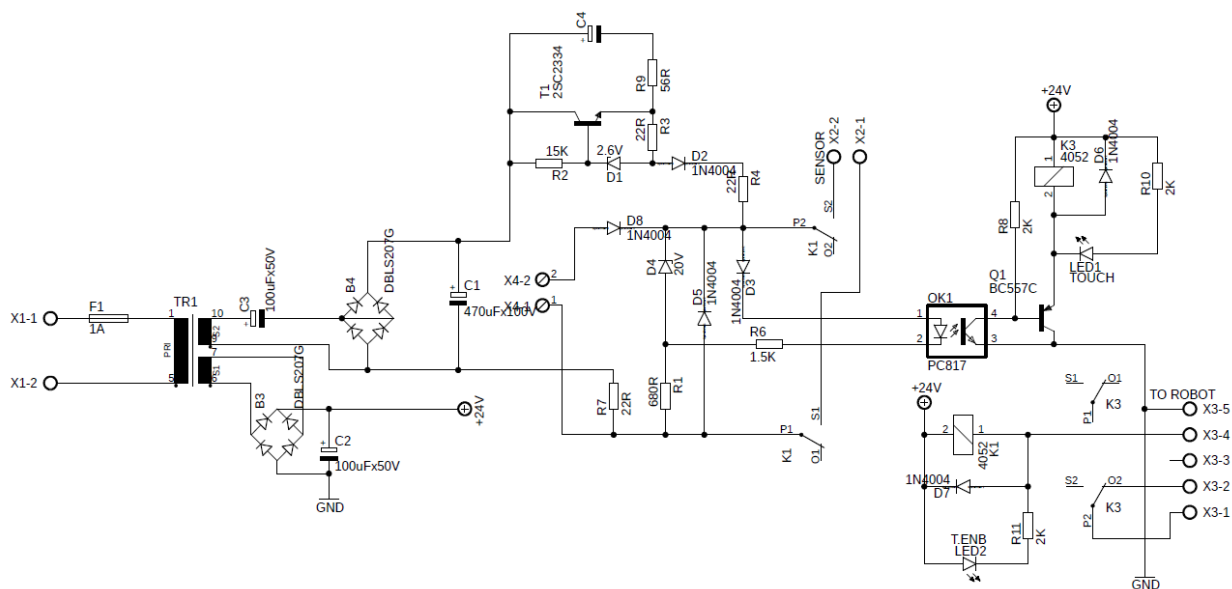


27 pav. Lietimo sensoriaus geometrijos matavimo principo aiškinimas

Lietimo sensoriaus geometrijos matavimo principas pagrįstas elektros srovės šaltinio veikimu tarp stalo ir suvirinimo aparato degiklio (27 pav.). Degiklis stumiamas paieškos kryptimi tol, kol paliečia ruošinį. Palietus ruošinį – sensoriumi ima tekėti silpna elektros srovė, kuri detektuojama ir siunčiama į roboto valdiklį. Srovės tekėjimo momentas yra vertinamas kaip ruošinio geometrijos aptikimas. Roboto valdymo sistema palygina išmatuotą koordinatę su etaloninę ir tokiu būdu nustato detalės geometriją erdvėje.

3.1.2 Lietimo sensoriaus elektroninių grandinių projektavimas ir gamyba

Lietimo sensorius inicijuoja srovės šaltinį, kurio pagrindu paviršiaus palietimo momentu yra išmatuojamos ruošinio paieškos taškų koordinatės (27 pav.). Srovės šaltiniui kurti reikia pritaikyti NPN bipoliarinį tranzistorių T1. Šio tranzistoriaus darbo režimas turėtų kurti pastovios amplitudės srovės šaltinį. Šio srovės šaltinio įtampa turėtų būti ne mažesnė kaip 36 V, kad pramuštu metalo paviršiuje esančią oksidų plėvelę. Kad užtikrinti roboto grandinių elektrinę išrišimą nuo lietimio sensoriaus elektros grandinių, numatyta naudoti elektromechanines rėles K1 ir K3. Bendra projektuojamo lietimio sensoriaus elektrinė schema pateikta 28 paveiksle.



28 pav. Projektuojamo lietimio sensoriaus principinė elektrinė schema

Kaip maitinimo šaltinis, panaudotas standartinis maitinimo transformatorius TR1 turintis dvi 24 V apvijas. Viena iš jų maitina roboto-lietimio sensoriaus interfeiso grandines, skirtas komunikacijai tarp roboto ir lietimio sensoriaus (28 pav. kairėje pusėje). Kita apvija maitina lietimio sensoriaus grandinę. Transformatoriaus išėjimo įtampa kondensatoriaus C3 ir diodų tiltelio B4 sujungimo būdu – paaukštinama iki dvigubos amplitudės (48 V) tinkamam sensoriaus veikimui

užtikrinti. Srovės šaltinis inicijuojamas tranzistoriumi T1, parenkant veikti pastovios srovės režimu, kuri gali būti apskaičiuojama pagal šia formulę:

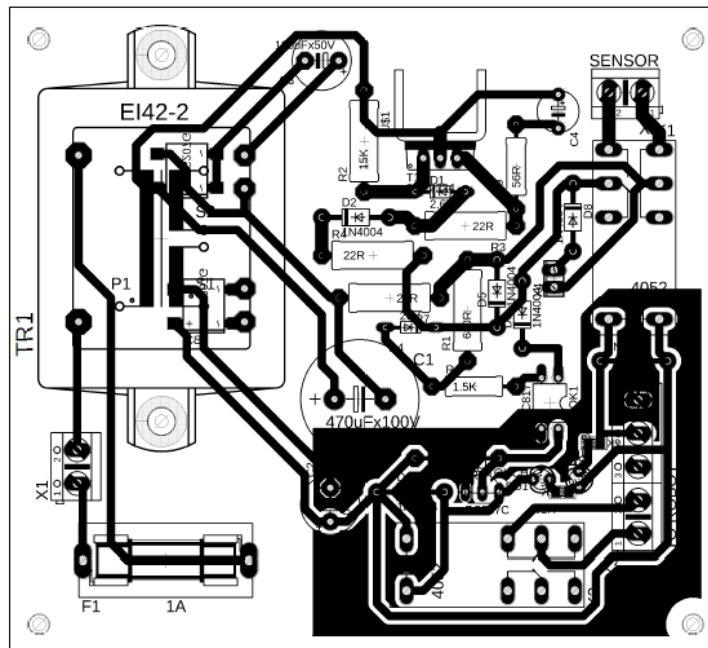
$$I_{SENS.} = \frac{U_{D1} + U_{BE}}{R_3} = \frac{2,5V + 0,7V}{22\Omega} = 145mA$$

čia: U_{D1} – stabilitrono $D1$ įtampa, U_{BE} – bipoliarinio tranzistoriaus T1 bazės-emiterio tiesioginė įtampa, skirta tranzistoriui atidaryti, R_3 – rezistoriaus R3 varža.

Veikimo principas:

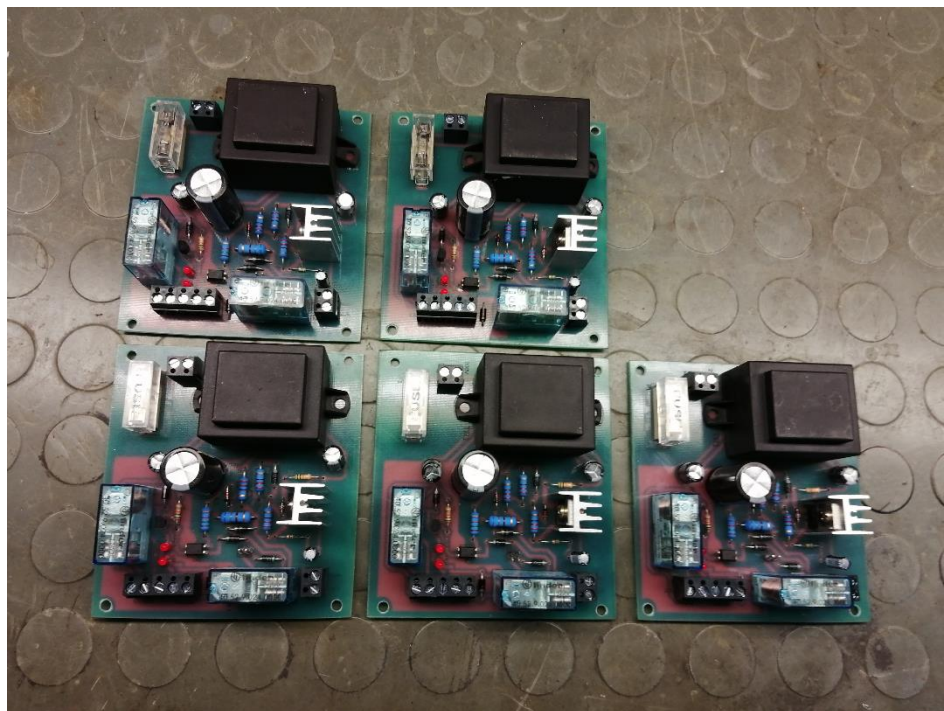
1. robotas per X3-4 kontaktą aktyvuoja sensorių – paveikiama K1 relė įjungia sensoriaus srovės grandinę į suvirinimo sistemos grandinę;
2. vyksta matavimas, kuriuo metu prilietus degiklio vielą prie suvirinimo ruošinio vyksta srovės šaltinio trumpasis jungimas;
3. Trumpojo jungimo metu, optronas OK1 nutraukia srovės tekėjimą relės K3 solenoidui – relės kontaktai užsidaro ir X3-1 ir X3-2 gnybtais perduodamas signalas roboto įėjimams, kad lietimo sensorius paveikė.
4. Baigus matavimą, robotas deaktyvuoja sensorių. Sensorius pereina į laukimo režimą.

Sukurto sensoriaus spausdintinė plokštė „Eagle“ programiniu paketu pavaizduota žemiau esančiame 29 paveiksle.



29 pav. Sukurto lietimio sensoriaus spausdintinė plokštė

Pagal „Eagle“ programinio paketo sukurtą spausdintinės plokštės elementų išdėstymą pagamintos spausdintinio montažo plokštės. Sulitavus elektronikos komponentus pagamintas lietimio sensorius. Įmonė UAB „ATC Baltic“ užsisakė pagaminti penkis tokius lietimio sensorius. Pagamintų prietaisų bendras vaizdas parodytas 30 paveiksle.



30 pav. Pagamintų lietimio sensorių bendras vaizdas

Įmonė UAB „ATC Baltic“ pagal poreikį sukurtą lietimą sensorių diegia savo robotinėse sistemose. Pagamintas prietaisas turi analogų rinkoje, tačiau kainuoja ženkliai mažiau, nei rinkoje sutinkami lietimą sensoriai. Ateityje įmonė pati gaminsis šiuos lietimą sensorius ir pritaikys savo projektuojamų sistemų poreikiams.

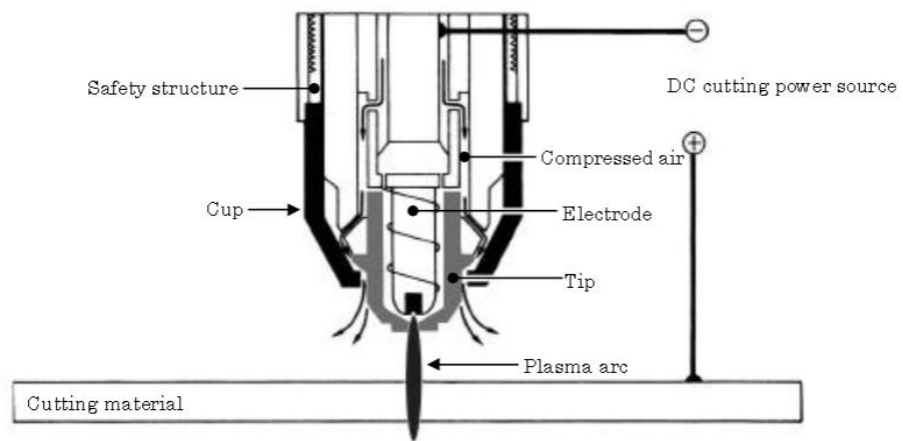
3.1.3 Plazmos srovės lanko aukščio kontrolės sistema

Plazminio metalo pjovimo metodas buvo išrastas 1950 metais ir iki šiol tai yra vienas iš populiariausių metalo pjovimo būdų. Populiarumą lemia sistemos paprastumas, pigios atsarginės dalys ir paprastas aptarnavimas. Įrangai nereikia naudoti jokių papildomų apsauginių dujų – užtenka naudoti įprastą aplinkos orą.

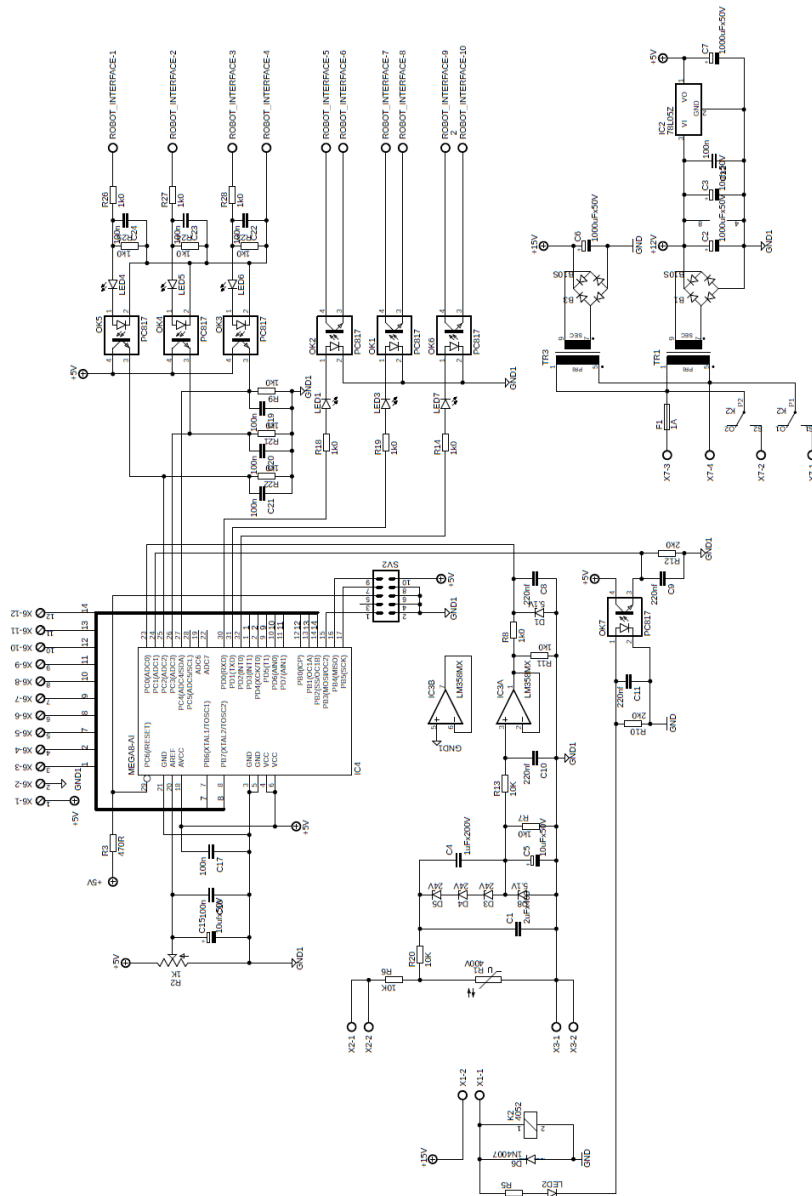
Įmonė UAB „ATC Baltic“ vykde projektą, kuriame robotas turi pjaustyti plieno ruošinius. Įmonės inžinieriai pasiūlė išspręsti aktualią problemą pjaunant metalą, t.y. sukurti oro tarpelio, kuriame dega elektrės plazmos lankas valdymo sistemą. Šis poreikis atsirado dėl to, kad nuo šio tarpelio, kuriame dega srovės lankas priklauso pjūvio kokybė. Jeigu tarpas yra per mažas – pjūvio vietoje formuojasi daug šlako ir pjūvis tampa nekokybiškas. Jeigu tarpas yra per didelis – pjūvis tampa platus, todėl išpjautas ruošinys praranda apibrėžtus matmenis ir nėra tinkamas tolimesnėms gamybinėms operacijoms. Ši problema yra aktuali plazminio metalo pjovimo sistemose, todėl nuspręsta suprojektuoti ir sukurti plazmos lankinio išlydžio aukščio kontrolės sistemą.

3.1.4 Elektros plazmos oro tarpelio valdymo sistemos projektavimas ir gamyba

Plazminio metalo pjovimo veikimas pagrįstas metalą veikiant jonizuotomis oro dalelėmis. Esant plazmos aukštai temperatūrai, ore esantis deguonis stipriai oksiduoja metalą ir jis sudega. Iš elektrodo (31 pav.) išpūstas elektros lankas (*Plasma arc*) dega tarp elektrodo ir pjaunamo ruošinio (*Cutting material*). Plazminio lanko temperatūra siekia net 3500 laipsnių pagal Celsijų. Tokioje temperatūroje metalo ruošinys pagauna įkaitintą deguoni ore ir sudega – oksiduojasi. Toje vietoje kur degė plazminis lankas susiformuoja tuštuma – pjūvis. Kadangi nuo oro tarpelio priklauso pjūvio kokybė, kyla poreikis išlaikyti tarpelį optimalų per visą ruošinio pjūvį.



31 pav. Elektros plazminio metalo pjovimo veikimo principas



32 pav. Suprojektuoto oro tarpelio valdiklio metalo plazminiam pjovimui principinė elektrinė schema

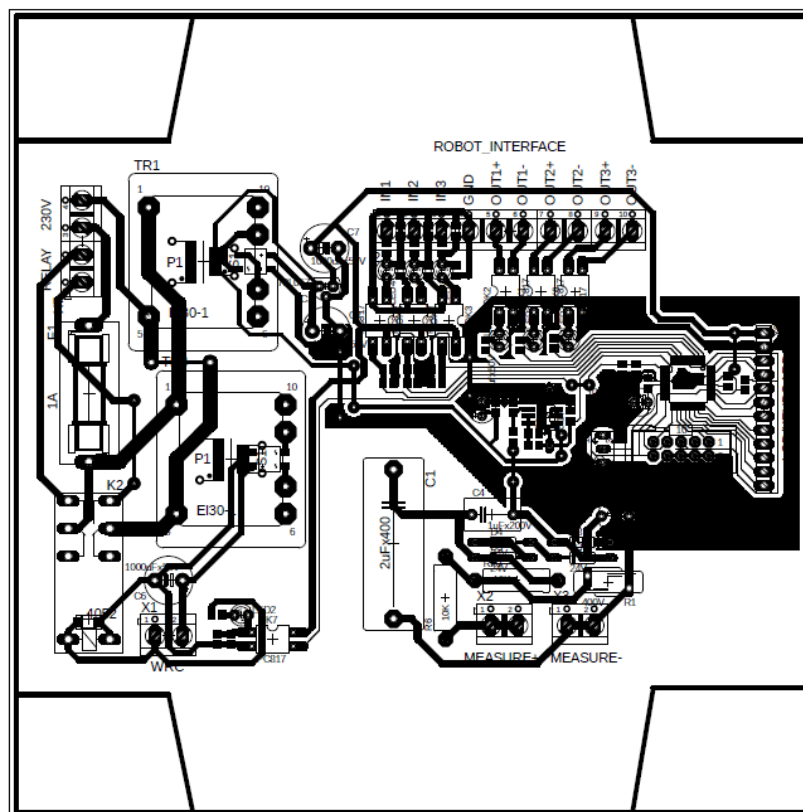
Plazminio pjovimo aparatas veikia lyg srovės šaltinis, t.y. palaiko pastovios amplitudės srovę. Srovės lanko įtampa proporcingai priklauso nuo oro tarpelio kuriame dega elektros lankas. Taigi norint išlaikyti pastovų oro tarpelį pjūvio metu, reikia išlaikyti pastovią elektros lanko įtampą.

Suprojektuojamas valdiklis oro tarpeliui palaikyti yra skaitmeninis, turintis Atmega8 mikrovaldiklio platformą (32pav.). Komunikacijai su robotu valdiklis turi tris diskretinius įėjimus *ROBOT_INTERFACE-1-4* ir tris diskretinius išėjimus *ROBOT_INTERFACE-5-10*. Elektros

plazmos lanko įtampa matuojama *X2-1* ir *X3-1* gnybtais, kur per įtampos daliklius ir operacinį stiprintuvą LM358 signalas apdorojamas ir perduodamas į mikrovaldiklio analoginį skaitmeninį keitiklį. Valdiklio valymui klaviatūra ir LCD ekranas prijungiamas prie *X6-1-12* jungties.

Atsiradus plazminiam lankui, iš plazminio aparato ateina WCR signalas. WCR signalas yra reikalingas plazminiam lankui nustatyti, kai jis užsidega. Užsidegus lankui mikrovaldiklis matuoja realią lanko įtampa ir atvaizduoja ją ekrane. Pagal nustatytą vertę valdiklis palaiko oro tarpelio aukštį. Jeigu tarpelis yra per didelis, *ROBOT_INTERFACE* išėjimais siunčiami diskretiniai signalai robotui sumažinti oro tarpelį, esant tarpeliui per mažam – oro tarpelis per *ROBOT_INTERFACE* išėjimus yra padidinamas.

Sukurto oro tarpelio valdiklio spausdintinė plokštė „Eagle“ programiniu paketu pavaizduota žemiau esančiame 33 paveiksle.



33 pav. Sukurto oro tarpelio valdiklio spausdintinė plokštė

Pagal „Eagle“ programinio paketo sukurtą spausdintinės plokštės elementų išdėstymą pagaminta spausdintinio montažo plokštė. Sulitavus elektronikos komponentus

pagamintas oro tarpelio valdiklis elektros plazmos pjovimo sistemai. Pagamintos plokštės bendras vaizdas parodytas 34 paveiksle; pagaminto prietaiso bendras vaizdas parodytas 35 paveiksle.



34 pav. Sukurto valdiklio plokštė



35 psv. Sukurto prietaiso bendras vaizdas (dešinėje); plazminio metalo pjovimo sistema (kairėje)

Suprojektuotas prietaisas iki šiol sėkmingai yra pritaikytas viename iš UAB „ATC Baltic“ projektų ir naudojamas elektros lanko oro tarpeliui stabilizuoti.

3.1.5 Inostart rezultatai

- 1) Pagaminti lietimų sensorių kainavo gerokai pigiau nei perkant panašų analogą rinkoje;
- 2) Ateityje įmonė lietimų sensorius gaminsis pagal poreikį savo projektams įgyvendinti;
- 3) Taip pat, iš tu pačių projektui skirtų lėšų pavyko pagaminti ir plazmos lanko oro tarpelio kontrolės sistemą. Todėl galima teikti, kad projekto iškelti tikslai yra netik įgyvendinti bet ir pranokti;
- 4) Sukurtos sistemos yra naudojamos šiuolaikinėse metalo apdirbimo robotinėse sistemose.

3.2 „Delfoi Arc“ programa

„Delfoi Arc“ yra robotinė "offline" programavimo sistema su simuliacijomis tinkanti daugeliui robotų gamintojų[5]. Programa naudoja 3D CAD geometriją kuriant robotų suvirinimo trajektorijas. Užtikrina tikslų ir labai greitą suvirinimo proceso kūrimą su patogia grafine vartotojo sąsaja.

„Delfoi Arc“ siūlo įvairių funkcijų:

- Tiesios suvirinimo siūlės
- Išlenktos ir dvigubo išlenkimo siūlės
- Suvirinimas aplink aštrų kampą
- Daugiasluoksnis suvirinimas
- Tiesių ir išlenktų siūlių sukabinimas. ISO AWS standartai
- Suvirinimo siūlės sekimas 1D, 2D ir 3D
- Automatinis masės centro suradimas (bi-sector)
- Susidūrimo aptikimas realiu laiku
- Robotų pasiekiamumas nustatant ašių skaičių
- Klonavimas, perkėlimas, veidrodinio atspindžio f-ja, papildomai aptinkama išorinė

ašis

- Programos patvirtinimas (syntax check)
- Suvirinimo tvarkos valdymas
- Lengva naudojime grafinė vartotojo sąsaja

4. Tiriamoji dalys

4.1 Pasiruošimo darbai

Įsitikiname, kad roboto degiklio centrinis darbinis taškas yra tinkamai sukalibruotas, pasinaudodami kalibravimo įtaisų.



36 pav. Kalibravimo įtaisas

Kalibravimo įtaisą prisukę prie roboto, matome, kad jutiklio galas beveik tobulai liečiasi su kalibravimo įtaiso adata. Tai reiškia, kad robotas yra sukalibruotas teisingai, ir nurodyti taškai erdvėje tarp „Delfoi Arc“ programos sutaps su taškais realybėje.



37 pav. Patikrinta ar robotas gerai sukalibruotas

Darant bandymus buvo pasirinkti du vienodi T formos šablonai. Jie buvo padėti ant pozicionieriaus laboratorijoje, ir pasukti skirtingais kampais. Roboto programoje pažymėjau po šešis taškus ant kiekvienos detalės.



38 pav. Išmatuoti šeši taškai ant pirmo ruošinio

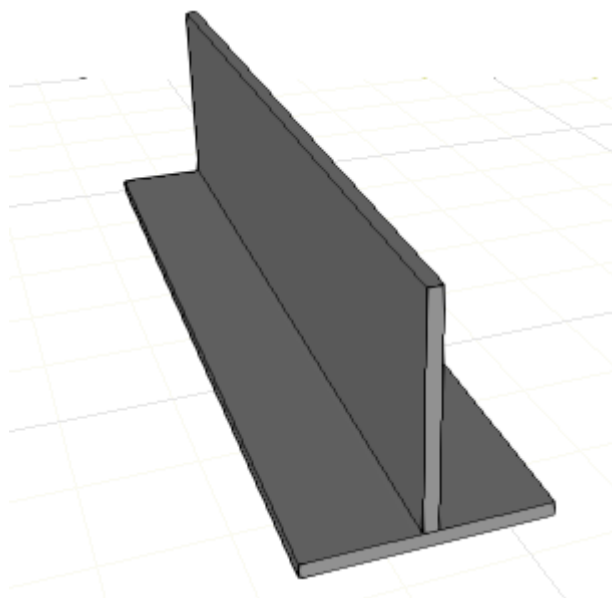


39 pav. Išmatuoti šeši taškai ant antro ruošinio

Programą įkėliau į nešiojamą duomenų laikmeną ir perkėliau į „Offline“ programavimo aplinką Delfoi Arc.

4.2 Gaminio paruošimas

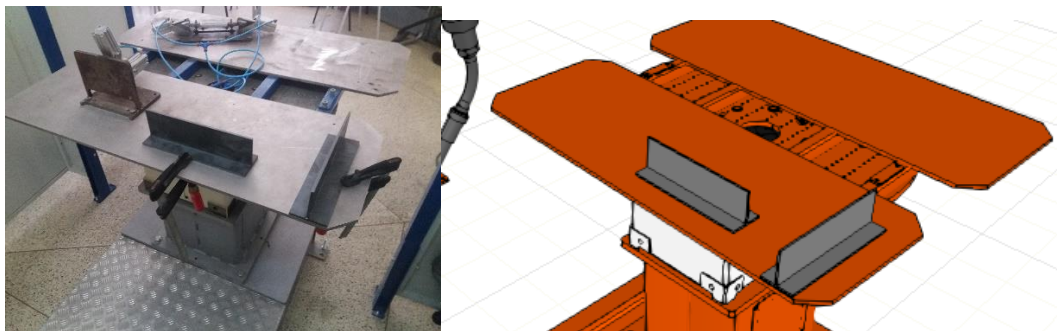
Išsimatuoju šablono matmenis, kurie yra 300mm ilgio, 80mm pločio, 85mm aukščio, 5mm storio, ir sukūriau Šablono kopija „Delfoi Arc“ programoje.



40 pav. Ruošinio atvaizdas simuliacijoje

Paspaudžiau „attach“ funkciją, kad detalę padėčiau ant pozicionieriaus programoje, tada pasinaudodamas funkcija „6/9 pt“ nurodau ant kurios plokštumos buvo padėti trys taškai, ant kurios du taškai, ir ant kurios vienas taškas.

Tą patį padariau ir antrai detalei ir jos buvo pastatytos į tokią pat poziciją kaip ir laboratorijoje.



41 pav. Kairėje ruošinio padėtis laboratorijoje, dešinėje ruošinių pozicijos atkurtos ant „Delfoi Arc“ programos

„Delfoi Arc“ programoje nurodau suvirinimo gijas stalui stovint pradinėje pozicijoje ir pasukus jį 45 laipsniais. Šią programą įsirašiau į nešiojamą duomenų laikmeną ir perkėliau į robotą laboratorijoje.

Paleidžiu programą laboratorijoje ir išmatuoju paklaidas tarp, kur turėtų būti viela ir kur ji yra. Matavau po du taškus, siūles pradžioje ir pabaigoje. Matavimai atlikti su tarpumačiu.

1 lentelė. Paklaidos padarius pirminius matavimus

Pozicionierius pasuktas 0°						
	Δx_1 , mm	Δx_2 , mm	Δy_1 , mm	Δy_2 , mm	Δz_1 , mm	Δz_2 , mm
W1	2	3.5	2	2	2	2

W2	1	1	1	2	2	3
W3	2	2	2	1.5	3	2
W4	3	1	2	3	2	2
Pozicionierius pasuktas 45°						
W5	2	1	1	1.5	2	1
W6	2	1.5	1	1	1	1.5
W7	2	2	1.5	1.5	2	1
W8	1	2	3	2	2	1

2 lentelė. Pirminiai matavimai naudojant „touch sensor“

Pozicionierius pasuktas 0°						
	Δx_1 , mm	Δx_2 , mm	Δy_1 , mm	Δy_2 , mm	Δz_1 , mm	Δz_2 , mm
W1	1	0.5	1	1	0	0.5
W2	0.5	0	0	0	0	1
W3	0	1	0	0.5	0.5	0.5
W4	0	1	1	1.5	0	0.5

Pozicionierius pasuktas 45°						
W5	0.5	1	1	0.5	1	0.5
W6	0	0.5	0.5	0.5	0	1
W7	1	0.5	0	0	0.5	0.5
W8	1	1	0	0	0.5	0

4.3 Ašių perkalibravimas

Ašių perkalibravimas atliekamas panaudojant „Mastering“ metodą. Šis metodas yra taikomas norint pagerinti roboto matematinio modelio kinematikos sutapimą su jo manipulatorius ašių kinematika. Tam tikslui sukuriama speciali programa, kuri atlieka matavimus panaudojant sukalibruotą įranki kaip šio atveju TOOL1 – degiklio centrinė darbo koordinatė. Pasikliaujant šio įrankio tikslumu, apmatuojama smeigė, kurios koordinatė erdvėje pasirenkama laisvai. Į kurią kalibravimo programą įeina 12 taškų. Kiekvienas iš šių taškų turi pajudinti vienos ašies posūkį kiek galima labiau į vieną pusę ir taip du kartus – pirmyn ir atgal visoms šešioms ašims. Gauta programa paleidžiama per roboto specializuotą funkciją, kuri ir atlieka šią „Mastering“ procedūrą. Paprastai „Mastering“ metodas yra iteracinis ir kartojamas tol kol gaunamas pakankamai geras tikslumas. Šiuo atveju „Mastering“ procedūrą teko daryti du kartus, kol buvo gautas geras rezultatas.

3 lentelė. Matavimai atlikus „mastering“

Pozicionierius pasuktas 0°						
	Δx_1 , mm	Δx_2 , mm	Δy_1 , mm	Δy_2 , mm	Δz_1 , mm	Δz_2 , mm
W1	1	3	1	2	1	1

W2	1	1	0	1	1	2
W3	2	1	1	1.5	2	1
W4	1	1	2	2.5	1.5	2
Pozicionierius pasuktas 45°						
W5	0.5	1	1	1	0.5	1
W6	1	1	0	1	0.5	1
W7	0.5	0.5	1	1	1	1
W8	0.5	1	1	2	0	1

4 lentelė. Matavimai atlikus „Masteringa“ ir naudojant „Touch sensor“

Pozicionierius pasuktas 0°						
	Δx_1 , mm	Δx_2 , mm	Δy_1 , mm	Δy_2 , mm	Δz_1 , mm	Δz_2 , mm
W1	0	0.5	0.5	0.5	0	0.5
W2	0	0	0	0	0	0.5
W3	0	0.5	0	0	0	0
W4	0	0	0.5	0.5	0	0

Pozicionierius pasuktas 45°						
W5	0.5	0.5	0	0.5	0.5	0.5
W6	0	0	0	0	0	1
W7	0.5	0.5	0	0	0	0
W8	0.5	0.5	0	0.5	0	0

4.4 Programos kodas

```

PTP P9 Tool0 Base1 100%
# CALLFAR 499,0
- S9 OK for W1 [ Tool0, Base1, Defaults, (1) ]
  <no references, 15 positions>
- S10 OK for W1 [ Tool0, Base1, Defaults, (2) ]
  <no references, 15 positions>
- S11 OK for W3 [ Tool0, Base1, Defaults, (3) ]
  <no references, 15 positions>
- S12 OK for W3 [ Tool0, Base1, Defaults, (4) ]
  <no references, 15 positions>
- S13 for W5 [ Tool0, Base1, Defaults, (5) ]
  <no references, 15 positions>
- S14 for W5 [ Tool0, Base1, Defaults, (6) ]
  <no references, 15 positions>
- S15 for W7 [ Tool0, Base1, Defaults, (7) ]
  <no references, 15 positions>
- S16 for W7 [ Tool0, Base1, Defaults, (8) ]
  <no references, 15 positions>

```



Pirmoje programos eilutėje robotas yra pastatomas į namų poziciją. „Callfar“ iškviečia funkciją, kuri yra skirta nukirpti degiklio vielą, tai užtikrina tikslus matavimus. Toliau vyksta „search“ funkcija, S9-S16, skirta išmatuoti kiek šablono pozicija skiriasi nuo jo numatytos vietos. Aštuonios „search“ funkcijos išmatuoja paklaidas visoms siūlėms, naujos šablonų pozicijos erdvėje yra įrašomos į roboto atmintį.

W1-W8 yra siūlės sekimo funkcijos. Funkcijos W1-W4 seka keturias siūles roboto pozicionieriui esant pasuktam 0° kampu, funkcijos W5-W8 seka tas pačias siūles, bet pozicionierius yra pasuktas 45° kampu. Paskutinė programos funkcija sugrąžina robotą atgal į namų poziciją.

5. Išvados

„Offline“ programavimas pagerina robotinės sistemos našumą ir atsiperkamumą, nes sumažina robotinės sistemos paruošimo darbui laiką, kuris yra nenaudingas gamyboje.

Programuojant sistemas „Offline“ metodu ir programą perkėlus į realią robotinę suvirinimo sistemą - neapibrėžtys atsiranda dėl „Offline“ naudojamos programos ir manipulatoriaus kinematikos nesutapimo. Tai paaiškina bandymo metu gautų duomenų rezultatai.

Gautas neapibrėžtys leidžia sumažinti panaudotą papildomą įrangą, kuri buvo suprojektuota „Inostart“ projekto metu. Pagamintu „Touch“ jutikliu pamatuoju realaus gaminio trajektoriją erdvėje įvedamos programoje pataisos sumažinančios gautas neapibrėžtis. Tai įrodo bandymų metu gauti rezultatai.

Atlikus „Mastering“ procedūrą kinematinis roboto modelis priartėjo prie „Offline“ programos virtualaus matematinio modelio kinematikos – todėl pagerėjo ir kuriamų programų tikslumas.

Šaltiniai

- [1] Wen Guojun, Xu Linhong and He Fulun, „Offline Kinematics Simulation of 6-DOF Welding Robot“, *2009 International Conference on Measuring Technology and Mechatronics Automation*, 2009
- [2] Gao F, Chen Q and Guo L „Study on arc welding robot weld seam touch sensing location method for structural parts of hull“ *2015 International Conference on Control, Automation and Information Sciences* 42-46
- [3] S. Yamane; Y. Kaneko; A. Hirai; K. Ohshima „Fuzzy control in seam tracking of the welding robots using sensor fusion“ *Proceedings of 1994 IEEE Industry Applications Society Annual Meeting*
- [4] A. Castro, J. P. Souza, L. Rocha and M. F. Silva, "Adaptpack studio: Automatic offline robot programming framework for factory environments", *19th IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC 2019)*, vol. 1, 2019.
- [5] <https://atcbaltic.com/wp-content/uploads/2018/01/Delfoi-Robotics-OLP-3.pdf>