



ROBOTTISOLUN KÄYTTÖÖN- OTTO

Miikka Heiskanen

OPINNÄYTETYÖ
Syyskuu 2023

Sähkö- ja automaatiotekniikka
Automaatiotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkö- ja automaatiotekniikka
Automaatiotekniikka

HEISKANEN MIIKKA:
Robottisolun käyttöönotto

Opinnäytetyö 49 sivua
Huhtikuu 2024

Opinnäytetyössä käsitellään kaivosteollisuuteen suunnatun robottisolun käyttöönottoa ja sen vaiheita. Työssä keskitytään käyttöönottoon työnantajan tiloissa ja asiakkaalla tehtävään käyttöönottoon liittyviä asioita ainoastaan sivutaan.

Käyttöönoton pohjana on robottisolu, jonka robottina toimi Fanucin M-710iC/50 6-akselinen robotti. Opinnäytetyössä käsitellään vaihe vaiheelta tämän käyttöönotto alkaen siitä, miten robotille luodaan kehykset sekä mitä ne tarkoittavat, aina siihen asti, kunnes solu on valmis testattavaksi.

Työssä olennaista on ohjelman sekä liikeratojen luominen sekä näihin liittyvät tekijät. Työssä käsitellään myös mitä konenäkö pitää sisällään sekä kuinka luodaan kameraohjelma, jonka avulla suoritetaan poiminnat. Työssä arvioidaan myös suunnitteluun sekä käyttöönottoon liittyviä turvallisuustekijöitä.

Lopuksi solun toiminta testataan ja luovutetaan jatkettavaksi työntekijälle, joka viimeistelee sen ja ottaa käyttöön asiakkaalla. Solun toiminnan seuraaminen jatkuu asiakkaalla toimivan operaattorin toimesta, jonka pyynnöstä toimintaa voidaan muokata paremmaksi.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Electrical and automation engineering
Automation engineering

HEISKANEN, MIIKKA:
The Deployment of a Robotic Cell.

Bachelor's thesis 49 pages
April 2024

The thesis deals with the implementation of a robotic cell aimed at the mining industry and its phases. The focus of the work is on the implementation in the premises of the commissioning company with only peripheral mention of aspects related to the implementation at the customer's site.

The basis for the implementation was a robotic cell with a Fanuc M-710iC/50 6-axis robot. The thesis cover the implementation process starting from the creation and features of frames for the robot, all the way to the final stage in which the cell was ready for testing.

Essential to the work is the creation of the program and motion paths, along with related factors. The thesis also addresses what machine vision involves and how to create a camera program for picking operations. Safety factors related to design and implementation are also evaluated in the work.

Finally, the operation of the cell was tested and handed over for further work by the company who finalises it and implements it at the customer's site. Monitoring of the cell's operation continues at the customer's site by an operator, who can request modifications to improve its performance.

Key words: robot cell, robotics, implementation, machine vision

SISÄLLYS

1	JOHDANTO	6
2	JTA CONNECTION OY	7
3	TEORIA	8
3.1	Ohjelma.....	8
3.1.1	Liikeradat ja poimintapisteet	8
3.2	Konenäkö	9
3.3	Teollisuusrobotti	9
3.4	Käyttäjäkehukset	10
3.5	Työkalukehukset	11
4	LAITTEISTOT JA OHJEMISTOT	12
4.1	Yleisesittely	12
4.2	Robotin työkalut	18
4.3	Työkalut.....	20
4.4	Ohjelmistot	21
5	OMA TYÖ	23
5.1	Työkalukehukset	23
5.2	Käyttäjäkehukset	26
5.3	Ohjelma.....	30
5.3.1	Rakenne	30
5.3.2	Liikeradat ja poimintapisteet	32
5.3.3	Turvatoiminnot.....	36
5.4	Konenäkö	38
5.4.1	Kappaleen ja holkin poiminnan opetus	39
5.5	Testaus	46
6	POHDINTA	47
	LÄHTEET	48

ERITYISSANASTO

PLC	Ohjelmoitava logiikka
IP-osoite	Yksilöivä verkko-osoite
HMI	Käyttöliittymä
Fanuc	Teollisuusrobottien valmistaja

1 JOHDANTO

Tässä opinnäytetyössä käsitellään robottisolun käyttöönottoa. Asiakas oli tilannut yrityksestä, jossa työskentelen, robottisolun, jonka tarkoitus oli syöttää työstökoneelle kappaleita. Kyseiset kappaleet on muokattu kuvista pois. Työssä käsitellään vain robotin puolen käyttöönottoa, ja ohjelmoitavan logiikan tuli hoitamaan toinen työntekijä.

Työn lähtökohtana oli luoda robotille kehykset, jotta käyttöönottajän työ helpotuisi. Työn edetessä kuitenkin työtä jatkettiin ja robotti ohjelmoitiin lähes täysin. Robotin ohjelmoinnin lisäksi kameroille luotiin ohjelma, jolla kappaleita saatiin poimittua. Työssä oli ulkopuolisina toimittajina työstökoneen toimittaja, joka toimitettiin vasta asiakkaalle sekä induktiolämmittimen toimittaja ja käyttöönottaja.

Yrityksellä ei ollut kohdallani tavoitetta työn määrälle, vaan tavoitteena oli lisätä oppimista ja saada mahdollisimman paljon valmiiksi ennen varsinaisen käyttöönottajän saapumista. Henkilökohtaiseksi tavoitteeksi asetettiin robotin ohjelmoinnin sekä kameraohjelmien valmiiksi saamisen, mistä käyttöönottaja voi jatkaa ja viimeistellä lopullisen työn valmiiksi sekä hoitaa yhteydet kuntoon ohjelmoitavan logiikan kanssa. Lisäksi tavoitteena oli oppia Fanucin robotin perustoiminnot ja ohjauksen sekä sen, kuinka robottia ohjelmoidaan.

Kun robottia lähdettiin ohjelmoimaan, ohjelmoitavalle logiikalle oli luotu peruspohja, jonka mukaan robotin ohjelmointi tulisi toteuttaa. Robotti oli kytketty ja käynnistetty, joten siihen pystyttiin lähtemään suoraan toteuttamaan ohjelmointia.

Työn aikana projektissa oli apuna useita yrityksen työntekijöitä, joita konsultoitiin aina tarvittaessa. Projektin toteutuskäytännöt ovat suoraan yrityksen muilta työntekijöiltä.

2 JTA CONNECTION OY

JTA Connection on Tampereella toimiva yritys, joka perustettiin vuonna 1999. Se tarjoaa räätälöityjä tuotanto- ja logistiikka-automaattioratkaisuja useille eri teollisuuden aloille, kuten robottisoluratkaisuja, kuljettimia ja kokonaisia tuotanto- ja kokoonpanolinjoja. Tuotteita ja palveluita on saatavilla maailmanlaajuisesti. Lisäksi JTA tarjoaa asiantuntevaa suunnittelupalvelua, joka on suunniteltu tehostamaan tuotantoprosesseja, sekä automaatiolaitteiden asennus- ja huoltotöitä. (JTA Connection Oy 2021b.)

JTA Connection Oy työllistää tällä hetkellä 125 henkilöä ja tarvitsee jatkuvasti uusia osaajia. Yritys on kasvanut vuosien varrella sekä yritysostojen kautta, että orgaanisesti. (JTA Connection Oy 2021b.)

Vuonna 2022 JTA Connection perusti tytäryhtiön Viroon, jonka tarkoituksena on tukea Suomen toimipisteen toimintaa. Käytännössä Viron toimipiste toimii puhtaasti suunnittelutoimistona, jossa työskentelee muutama mekaniikkasuunnittelija, PLC-suunnittelija sekä robottiohjelmoija. (JTA Connection 2021a.)

Vuonna 2021 Expert Service eriytyi erilliseksi tytäryhtiöksi, jonka omistaa 100-prosenttisesti emoyhtiö. Tytäryhtiön nimeksi tuli JTA Expert Service Oy. JTA Expert Service Oy on erikoistunut palveluliiketoimintaan. (JTA Connection 2021c.)

Itse aloitin työskentelyn yrityksessä vuoden 2023 keväällä, jonka aikana suoritin opinnäytetyön. Kesän jälkeen tein sopimuksen, joka mahdollistaa työskentelyn koulun ohella.

3 TEORIA

3.1 Ohjelma

Ohjelmalla tarkoitetaan jollekin laitteelle luotavia käskyjä, jotka on kirjoitettu sille ymmärrettävässä muodossa. Ohjelma on laitteiston vastakohta, joka puolestaan on tietokoneen fyysinen puoli (Linda Rosencrance n.d.)

Ohjelmointikielillä tarkoitetaan kieltä, jolla nämä kyseiset käskyt on luotu. Ohjelmointikieliä on satoja, mutta suurin osa niistä perustuu tekstipohjaiseen ohjelmointiin. Yleisimpiä ohjelmointikieliä ovat C, C++ sekä Python, mutta myös vanhempia kieliä, kuten Pascal on vielä käytössä. Osa robottien valmistajista, kuten Fanuc, on kehittänyt oman ohjelmointikielen. Nämä kielet ovat hyvin samantapaisia C-pohjaisien kielten kanssa. (Jacob Biba 2022.)

Ohjelman rakenteella tarkoitetaan yleisesti koodin rakennetta pääpiirteittäin. Ohjelmointitapoja on paljon ja tässä työssä keskitytään kyseisen yrityksen tapaan rakentaa ohjelma.

3.1.1 Liikeradat ja poimintapisteet

Robotin liikeradoilla tarkoitetaan sitä, miten robotti liikkuu tai suorittaa työn. Liikeratojen suunnittelulla voidaan pidentää robotin käyttöikä ja vähentää huoltoja. Voidaan tehdä niin sanottu Point to Point liike, jolloin robotti liikkuu pisteestä pisteeseen ilman, että sen väliliikkeitä on määritetty. Toinen tapa on määrittää robotille välipisteitä, jonka kautta se liikkuu pisteestä pisteeseen. Tämä tapa on turvallisempi, koska robotin liikeradat ovat tiedossa ja tällä voidaan välttää turvaväliin törmäykset sekä kuulumattomat liikkeet.

Liikepisteiden tallennuksessa on kahta eri tyyliä, joita ovat Joint ja Linear eli nivel ja lineaariliike. Yleensä nivelliikettä käytetään vapaampaan liikkumiseen ja lineaariliikettä poiminta and asetus toiminnoissa. Lineaariliikettä käytettäessä robotti liikkuu lineaarisesti koordinaatiston mukaan määrättyyn pisteeseen, kun taas nivelliikettä käytettäessä robotti liikkuu sen akselien mukaisesti helpointa reittiä. Pisteisiin voidaan myös määritellä, kuinka tarkasti robotin halutaan kulkevan

määrätyn pisteen kautta vai voiko oikaista ja lähteä jo seuraavaan pisteeseen ennen kuin edellinen on täysin saavutettu. Tällä toiminnolla saadaan nopeutettua robotin toimintaa ja tehtyä sen liikkeistä sulavampaa.

Samalla tavalla robotille määritetään poimintapisteet, jotka voidaan määrittää joko suoraan robotille tai ne voidaan saada konenäön kautta.

3.2 Konenäkö

Konenäöllä tarkoitetaan kuvankäsittelyyn ja kameraan perustuvaa havainnointijärjestelmää, jolla analysoidaan ja havainnoidaan digitaalista kuvainformaatiota. Konenäköjärjestelmä koostuu itse kamerasta, kuvattavasta kohteesta, valonlähteestä sekä tietokoneesta, joka sisältää kuvankäsittelyohjelman. (Teito 2023.)

Konenäöllä pystytään tunnistamaan virheitä kappaleissa tai kuten tässä tapauksessa tunnistetaan robotin poimintapiste, jonka avulla kappale tai holkki poimitaan. Muita käyttötarkoituksia konenäölle ovat muun muassa sarjanumeroiden lukeminen ja kappalelaskenta. Konenäöllä pystytään havaitsemaan poikkeamia, jota ihmissilmä ei kerkeä havaitsemaan. Tämän vuoksi konenäköä hyödynnetäänkin kohteissa, jossa tarkastuksen pitää olla tarkkaa, toistettavaa, nopeaa sekä ympärivuorokautista. (Teito 2023.)

Konenäköä hyödynnetään useassa eri teollisuuden alalla, esimerkiksi robottisoluissa. Teknologian kehitys mahdollistaa uusia käyttökohteita sekä parantaa aiemmin automatisoituja kohteita. (Algol 2018.)

3.3 Teollisuusrobotti

Robotilla tarkoitetaan tietokoneella ohjattavaa yleiskäyttöistä konetta, joka pystyy käsittelemään työkaluja tai kappaleita. Yleiskäyttöisellä tarkoitetaan sitä, että robotin ohjelma on helposti muokattavissa ja se soveltuu useaan eri käyttötarkoitukseen. Robotti saattaa toimia useammalla eri käyttövoimalla esimerkiksi pneumaattisesti, hydraulisesti tai sähköisesti. Jotta robotti voidaan määritellä te-

ollisuusrobotiksi, vaaditaan siltä vähintään kolme eri liikkuvaa akselia. Lisäksi ohjausteknisessä mielessä robotilta on löydyttävä servo-ohjaus. Robotin päätehtävänä on liikuttaa työkalua ohjelman määrittämiin pisteisiin. (Wikipedia 2024.)

Robotti koostuu tyypillisesti akseleista ja servomootoreista akseleita on tyypillisesti 6 kappaletta. Kolme ensimmäistä akselia ovat niin sanottuja pääakseleita, kun taas kolme jälkimmäistä akselia liikuttavat toimilaitteita, esimerkiksi työkaluja. (Laimonas Bačkys 2021, 14.)

Robottien koordinaatistot jaotellaan nivel-, käyttäjä-, maailma- sekä työkalukoordinaatistoon. Nivelkoordinaatistossa robottia ohjataan sen akseliston mukaisesti, käyttäjäkoordinaatisto on robottiin luotu käyttäjän oma koordinaatisto. Maailmakoordinaatistolla tarkoitetaan johonkin teollisuusrobotin ympäristöön, esimerkiksi oheislaitteisiin sidottua ulkopuolista koordinaatistoa. Työkalukoordinaatisto puolestaan on suorakulmainen koordinaatisto, mikä työkalumäärittämisellä sidotaan johonkin kohtaan robotin työkalua. (Laimonas Bačkys 2021, 14.)

Robottia voidaan hyödyntää usealla eri teollisuudenalalla. Se voi olla joko tuotantolinjan pääkone tai osa suurempaa järjestelmää. Teollisuusrobotin vaatimukset vaihtelevat prosessista ja kohteesta riippuen suuresti. Yleensä robotin tavoitteena on työskentelyolosuhteiden parantaminen tai tuottavuuden nostaminen. (Suomen Robotiikkayhdistys ry 2023a.)

3.4 Käyttäjäkehukset

Ensimmäisiä vaiheita robotin käyttöönotossa on käyttäjäkehysten rakentaminen. Käyttäjäkehysillä määritetään robotin koordinaatisto eli X, Y ja Z suunnat jollakin tietyllä robotin toiminta-alueella. Esimerkiksi tässä työssä määritettiin viisi eri käyttäjäkehystä ja aina kun robotti siirtyy kehyksestä toiselle, on ohjelmassa vaihdettava käytettävä kehys oikeaan. (Pickit 2023.)

3.5 Työkalukehykset

Samalla lailla kuin käyttöönottaessa rakennetaan käyttäjäkehykset, täytyy robotille rakentaa myös työkalukehykset. Myös työkalukehyksissä opetetaan robotin X, Y ja Z suunnat, mutta tässä tapauksessa ne opetetaan työkaluille eikä alueille. Jokaiselle työkalulle on opetettava kehys erikseen ja se on vaihdettava myös ohjelmassa, kun työkalu vaihdetaan. Molempien kehyksien suunnittelussa on tärkeää ottaa koordinaatiston suunnat. X ja Y suunnan saa itse päättää mutta Z+ suunnan on aina oltava ylöspäin. (Dan Sula 2023.)

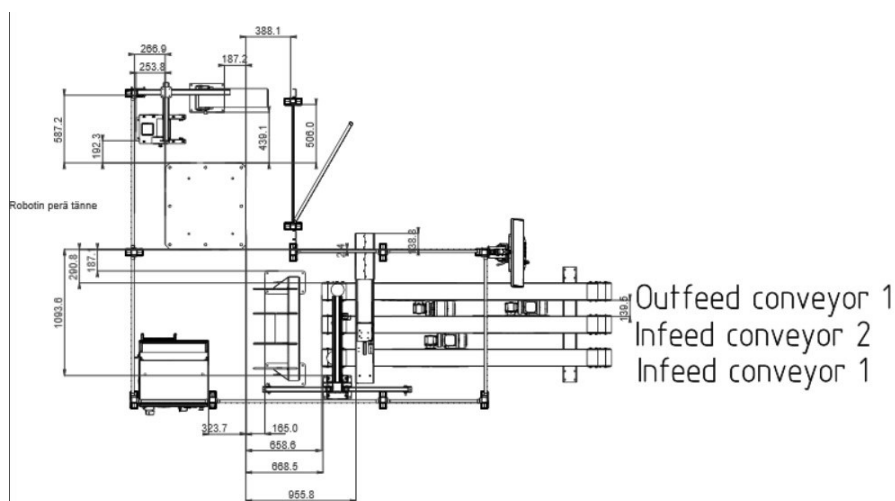
4 LAITTEISTOT JA OHJEMISTOT

4.1 Yleisesittely

Tässä kappaleessa käydään läpi työssä käytettävät laitteistot, ohjelmistot sekä perehdytään projektin yleiseen toimintaperiaatteeseen ja pohjakuvaan. Kuvissa esitettynä solun yleinen rakenne (KUVA 1, KUVA 2). Pohjakuvasta käy myös hyvin ilmi turva-aitojen sijainnit, jotka tekevät solusta erittäin ahtaan (KUVA 2).



KUVA 1. Robottisolun yleisrakenne.

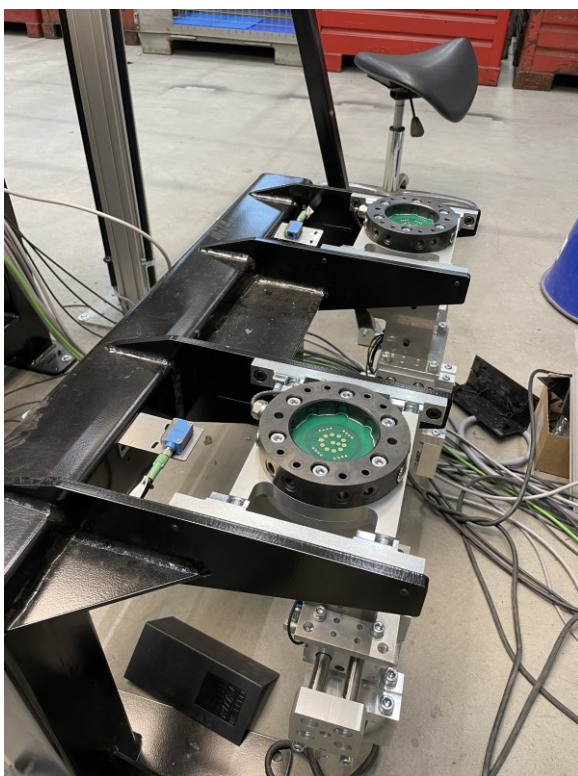


KUVA 2. Pohjakuva.

Ensimmäiseksi käydään läpi pohjakuva (KUVA 2). Solun vasemmassa alareunassa sijaitsee sähkökaappi sekä robotin ohjain. Oikeassa alareunassa on kolme kuljetinta (KUVA 3), joista kaksi on syöttävää ja yksi poistava kuljetin. Kuljettimien vasemmalla puolella ovat tarttujien 1 ja 2 telineet (KUVA 4).



KUVA 3. Lamellikuljettimet.



KUVA 4. Telineet työkaluille 1 ja 2.

Lopullisessa asennuksessa solun yläreunassa sijaitsee työstökone, joka toimitetaan vasta asiakkaalle. Työstökoneen tilalla oli tässä työssä puinen lava, jonka päälle kappaleet nostetaan (KUVA 5).



KUVA 5. Työstökoneena toimiva puinen teline.

Pohjakuvassa näkyy yläreunassa väliasema (KUVA 6), johon voidaan nostaa kappaleita odottamaan ennen niiden siirtämistä koneeseen, ja tätä kautta solun toimintaa saadaan nopeutettua. Väliaseman vasemmalla puolella on induktiolämmittimen teline, johon asetetaan induktiotyökalu.



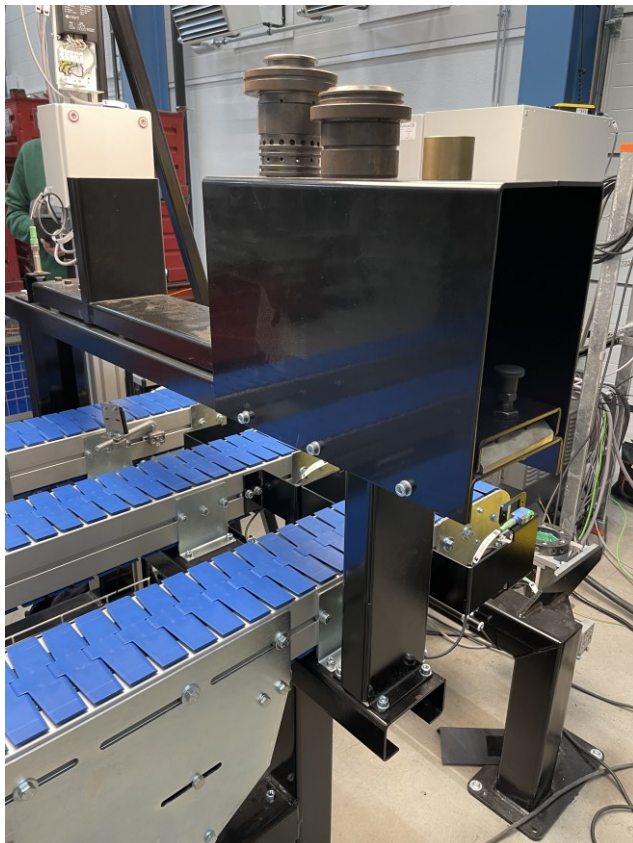
KUVA 6. Väliasema.

Kuvan keskellä on robotti, josta on esitettyä neliön muotoinen pohjalevy (KUVA 2). Robottina toimii Fanucin M-710iC/50 (KUVA 7). Robotti on Fanucin keskikokoisia 6 akselisia robotteja ja pystyy nostamaan jopa 50 kilon kuorman ja ulottuu 2050 millimetriin. (Fanuc 2024.)



KUVA 7. Fanuc M-710iC/50.

Tarkastusasema sijaitsee kuljettimen yläpuolella niiden vasemmassa reunassa (KUVA 8). Tarkastusasemalla pystytään poistamaan kappale turvapiirien sisäpuolelta niitä katkaisematta. Robottia voidaan pyytää käyttöliittymän kautta tuomaan kappale tarkastusasemalle, mikäli se halutaan tarkastaa erikseen.



KUVA 8. Tarkastusasema.

Poimintakuvaukseen tarkoitettu kamera ja sen valot sijaitsivat kuljettimien yläpuolella, pohjakuvasta katsottuna niiden vasemmalla reunalla (KUVA 2). Kamerana käytettiin Fanucin SC130EF2-mallia, johon oli kiinnitetty saman valmistajan 10–40 mm linssi (KUVA 9). Kamera oli kiinnitetty säädettävään telineeseen, jotta se voitiin kohdistaa optimaalisesti. Kuvausvaloilla pystytiin takamaan hyvä valaistus ja vähentämään kuvauksessa haittaa aiheuttavia varjoja.



KUVA 9. Fanuc SC130EF2.

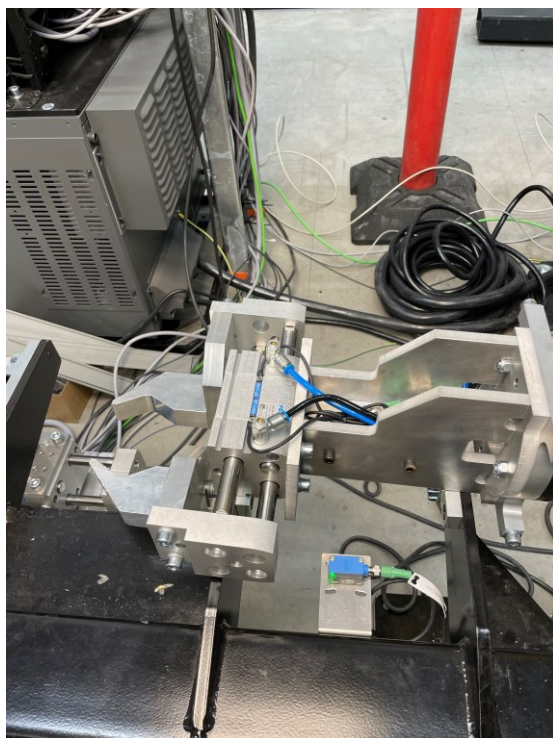
Solun yleinen toimintaperiaate on seuraava: Robottisolu on suunniteltu toimittamaan työstökoneeseen kappaleita ja holkkeja. Robotti noutaa kappaleen syöttökuljettimelta 1 ja toimittaa sen työstökoneeseen (KUVA 3). Samalla kun työstökone käsittelee syötettyä kappaletta, robotti vaihtaa työkaluksi induktiolämmittimen (KUVA 11). Kun työstökone on suorittanut kappaleen työstön, robotti lämmittää sen induktiolämmittimellä. Kappaleen lämmityksen jälkeen robotti vaihtaa työkaluksi holkkitarttujan ja noutaa holkin syöttävältä kuljettimelta 2 (KUVA 3). Tämän jälkeen robotti toimittaa holkin työstökoneeseen, missä työstökone asettaa holkin kuumennetun kappaleen sisään.

Kun kappale ja holkki on liitetty yhteen, kappale siirtyy uudelleen työstettäväksi. Tämä tapahtuu työstökoneen sisällä. Robotti hyödyntää työstöön kuluvaan aikaan ja siirtää toisen kappaleen odottamaan työstökoneen edessä olevaan väliasemaan (KUVA 6). Kun työstökone on käsitellyt kappaleen viimeisen kerran, robotti hakee valmiin kappaleen työstökoneesta väliasemaan ja asettaa uuden työstämättömän kappaleen työstökoneeseen. Viimeisenä vaiheena robotti noutaa valmiin kappaleen väliasemasta ja vie sen poistokuljettimelle, josta se kuljetetaan pois. Mikäli operaattori haluaa tarkastaa kappaleen hän voi ennen sen poisvientiä

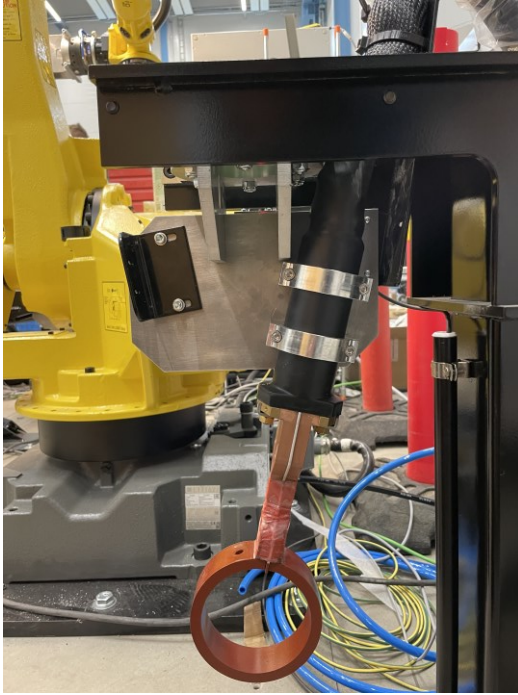
pyytää toimittamaan kappale tarkastusasemaan (KUVA 8). Tällöin kappaletta ei viedä poistavalle kuljettimelle vaan se toimitetaan kuljettimien yläpuolella sijaitsevaan tarkastusasemaan.

4.2 Robotin työkalut

Solussa käytössä olevalla robotilla on kolme työkalua: holkkitarttuja, kappaletarttuja sekä induktiotyökalu. Holkki- sekä kappaletarttuja ovat muuten samanlaisia, mutta kappale-tarttuja on isompi (KUVA 10). Lisäksi robotilla on induktiolämmitin, johon kuuluu lämmitinyksikkö (KUVA 12) sekä induktiopää (KUVA 11). Kappale- ja holkkitarttuja toimivat paineilmalla, ja sylinteriin on asennettu rajat kiinni- ja auki-tiloille. Induktiolämmittimen lämmitinpää sijaitsee työstökoneen edessä ja se oli liitetty varsinaiseen lämmitinkoneeseen paksulla kaapelilla.



KUVA 10. Kappaletarttuja.



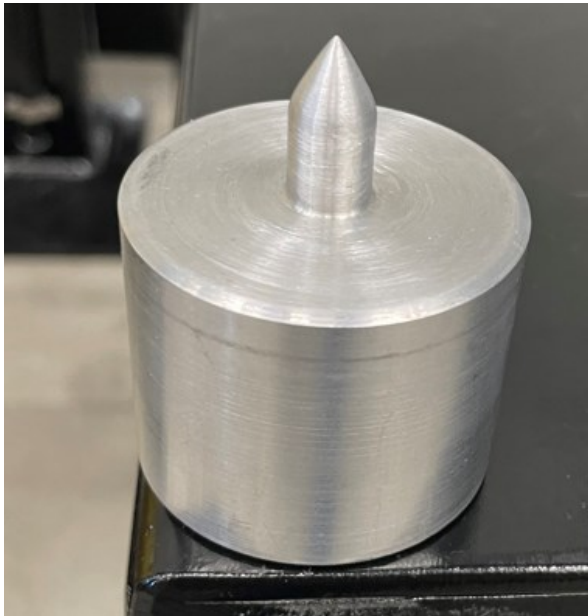
KUVA 11. Induktiolämmittimen lämmitinpää.



KUVA 12. Induktiolämmitin.

4.3 Työkalut

Robotin käyttöönotto vaatii useita eri erikoistyökaluja, joista monia toimittaa vain robotin valmistaja. Näitä olivat esimerkiksi robotin kehyksien opetukseen tarkoitettu vastapiikki (KUVA 13) sekä opetuspiikki. Kuvassa näkyvään opetuspiikkiin on kiinnitetty kiinnityksen avuksi jatkopala, joka avustaa kiinnitystä robottiin (KUVA 14).

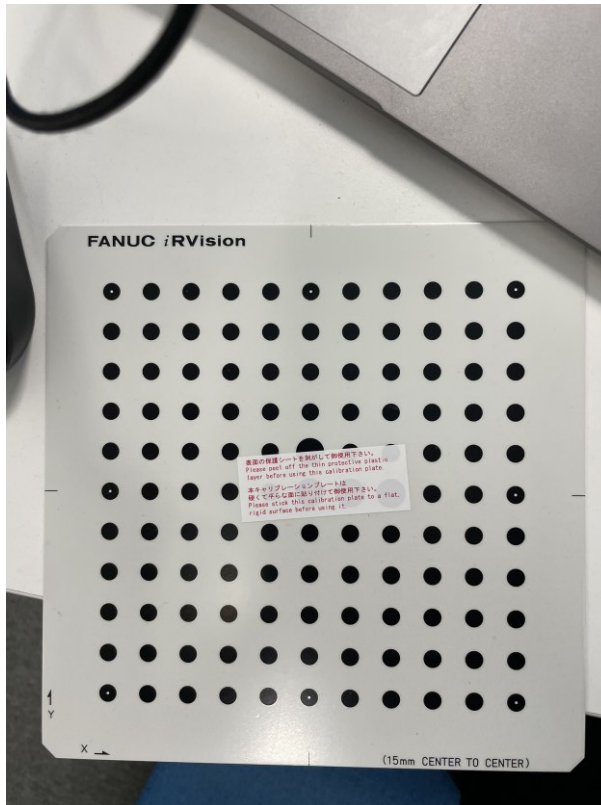


KUVA 13. Vastapiikki.



KUVA 14. Opetuspiikki.

Kameran opetuksen avuksi käytettiin Fanucin toimittamaa ruudukkoa, jolla saatiin kohdistettua kamera ja robotti keskenään samaan koordinaatistoon. Kyseisiä ruudukkoja on useita eri kokoja ja koko tulee valita aina työkohtaisesti sopivaksi. Tässä työssä käytettiin keskikokoista ruudukkoa (KUVA 15).



KUVA 15. Kameran opetusruudukko.

4.4 Ohjelmistot

Robotin ohjelmointi tapahtui lähes täysin Fanucin omalla käsipäätteellä (KUVA 16), joka oli kytkettynä suoraan robotin ohjaimeen. Projektissa oli mahdollisuus myös käyttää lisenssin vaativaa ohjelmointiohjelmaa nimeltä Roboguide.



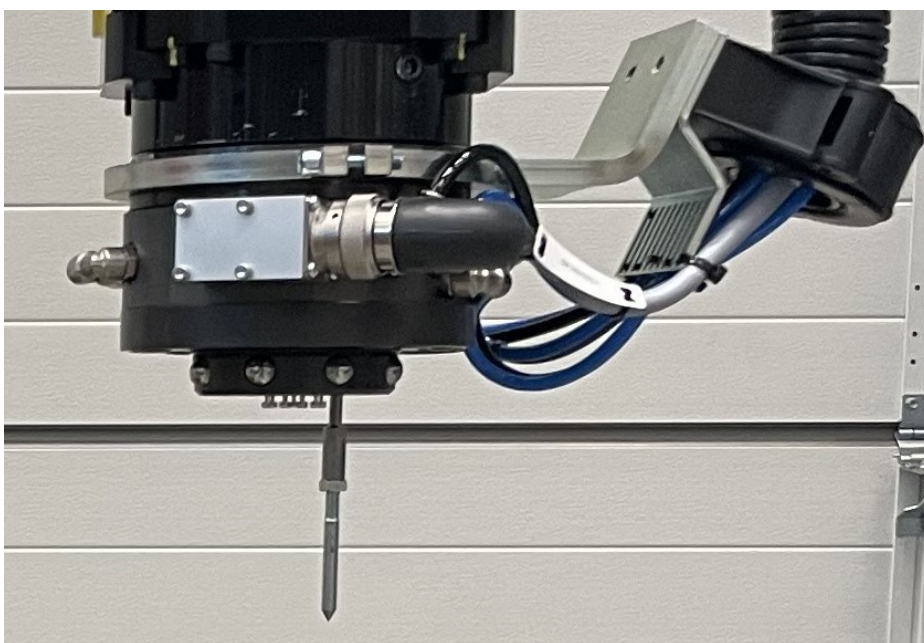
KUVA 16. Opetuskäsiohjain

Kameraohjelman luominen tapahtui Fanucin oman selainpohjaisen ohjelmiston avulla, johon yhdistettiin IP-osoitteen avulla samassa verkossa olevalla tietokoneella.

5 OMA TYÖ

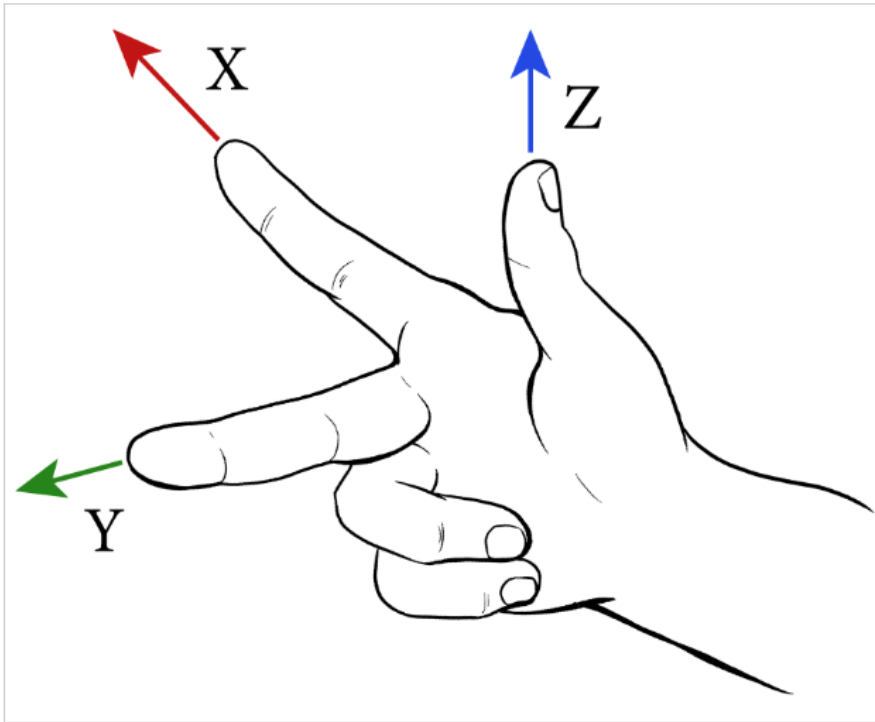
5.1 Työkalukehykset

Kehyksien opetus alkaa kiinnittämällä opetuspiikki robottiin. Piikin kiinnityskohdan saa päättää itse, mutta kiinnityksessä tulee muistaa, että se tulee olla toistettavissa eli kun robotti otetaan tehtaalla uudelleen käyttöön, tulee piikki pystyä kiinnittämään samaan paikkaan. Tässä tapauksessa piikki kiinnitettiin jatkopallalla, joka pienensi kierrettä ja näin piikki saatiin kiinnitettyä suoraan robotin tarttujaan (KUVA 17).

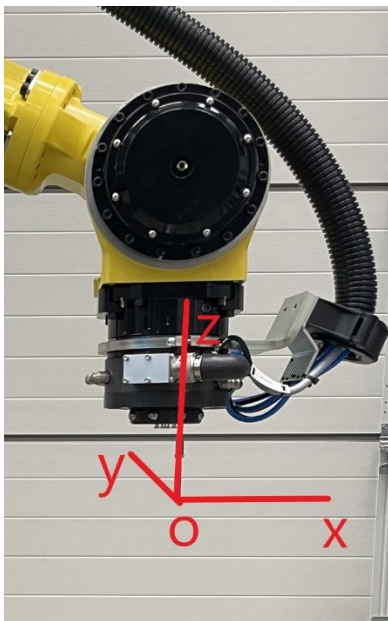


KUVA 17. Opetuspiikin kiinnitys.

Kun piikki on saatu tukevasti kiinnitettyä robotin tarttujaan, suunnitellaan seuraavaksi piikin koordinaatisto. Koordinaatiston opettaminen tapahtuu niin sanotun oikean käden säännön avulla. Tässä peukalo toimii Z akselina, etusormi X akselina ja keskisormi Y akselina (KUVA 18, KUVA 19). (Education Vex 2024.)



KUVA 18. Oikean käden sääntö. (Education Vex 2024.)



KUVA 19. Piikin koordinaatisto.

Kun piikki on saatu tukevasti kiinnitettyä robottiin ja sen koordinaatisto on valittu, täytyy se opettaa robotille. Tässä piikin kärki opetetaan robotille ja opetuksen jälkeen robotti tietää missä kärki sijaitsee ja pystyy jatkossa liikkumaan sen mukaisesti. Opetus tapahtuu opetuskäsiohjaimella eli kapulalla. Opetustapoja on useita, mutta tässä tapauksessa käytetään kolmen pisteen opetusta. Robotti aje-

taan vasten vastapiikkiä kolmelta eri puolelta ja jokainen piste tallennetaan robotille (KUVA 20). Mitä kauempana nämä pisteet sijaitsevat toisistaan sitä tarkempi opetus on. Tässä vaiheessa täytyy olla tarkkana, ettei piikin kärjet osu toisiinsa ja piikki pääse liikahtamaan. Mikäli piikki liikahtaa, täytyy opetus aloittaa alusta.

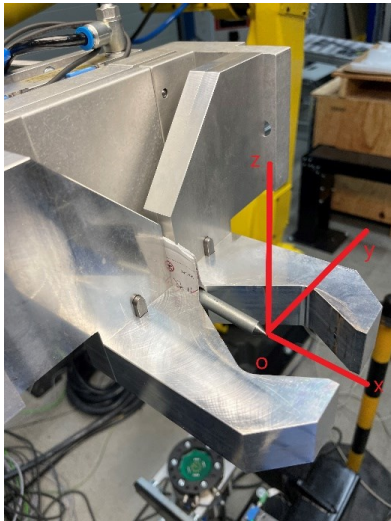


KUVA 20. Pisteiden opetus.

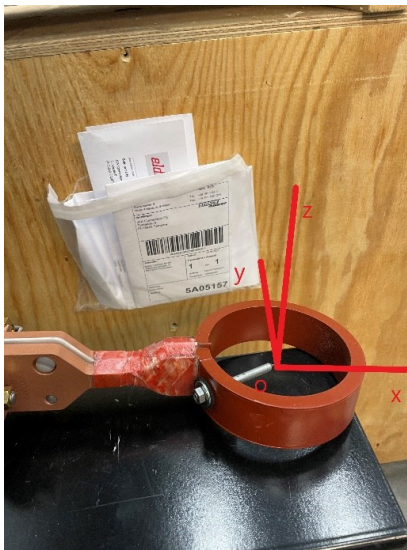
Kun robotille on opetettu pisteet, täytyy tarkistaa, liikkuuko robotti valitulla työkalulla koordinaatiston mukaisesti. Mikäli suunnat eivät ole koordinaatiston mukaisia, pystytään ne korjaamaan vaihtamalla opetustapa kolmen pisteen opetuksesta suoraan opetukseen ja muokkaamalla W, P tai R arvoa esimerkiksi 90 tai 180 astetta. Opetuksen tarkkuus saadaan testattua valitsemalla robotista työkaluksi opetettu työkalu, ajamalla se vasten vastapiikkiä ja katsoa liikkuuko se vastapiikin kärjessä kiinni, kun robottia pyöritetään esimerkiksi X akselin mukaisesti. Jos robotin piikin kärki heiluu liikaa, kun sitä vertaa vastapiikkiin on opetus epäonnistunut.

Muiden työkalujen opetus tapahtuu täsmälleen samanlaisesti kuin piikin, ainoana erona on piikin kiinnitys työkaluun ja koordinaatiston valinta. Piikin kiinnityksessä ja koordinaatiston valinnassa on hyvä muistaa, että ne tulee olla toistettavissa ja koordinaatiston Z+ suunta tulee olla ylöspäin. Koordinaatiston suunnat opetetaan

robotille yleensä sen mukaisesti, missä asennossa ne ovat työskentelypisteessä eli esimerkiksi missä asennossa tarttuja on, kun se poimii kappaletta. Kuvissa esitettynä robotin työkalujen koordinaatistot (KUVA 21, KUVA 22).



KUVA 21. Työkalujen 1 ja 2 koordinaatisto.



KUVA 22. Induktiopään koordinaatisto.

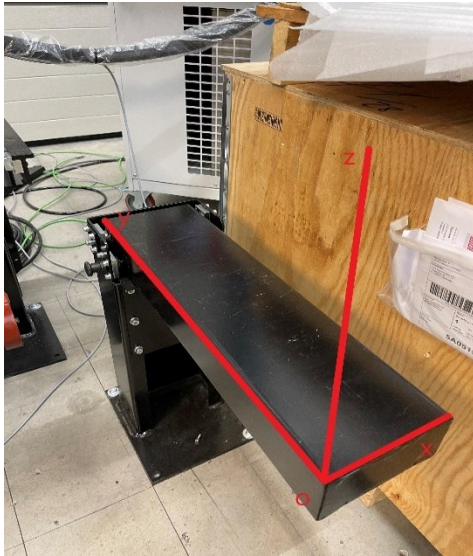
Kun jokainen työssä käytettävä työkalu on opetettu, dokumentoidaan työ hyvin, jotta niiden opetus pystytään toistamaan asiakkaalla.

5.2 Käyttäjäkehukset

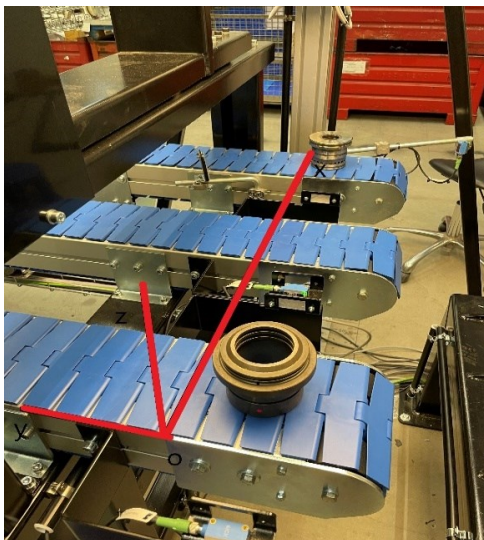
Kun opetuspiikki on opetettu tyhjään tarttujaan, voidaan tämän avulla opettaa käyttäjäkehukset. Kun piikki on kiinni ja opetettu, suunnitellaan, montako kehystä

työhön tarvitaan sekä mihin suuntaan näiden koordinaatisto valitaan. Tähän työhön kehyksiä valittiin neljä. Lisäksi työssä oli viides kehys, joka kuului työstökoneelle. Tässä työssä sitä ei kuitenkaan käsitellä, koska työstökone asennettiin vasta tehtaalla.

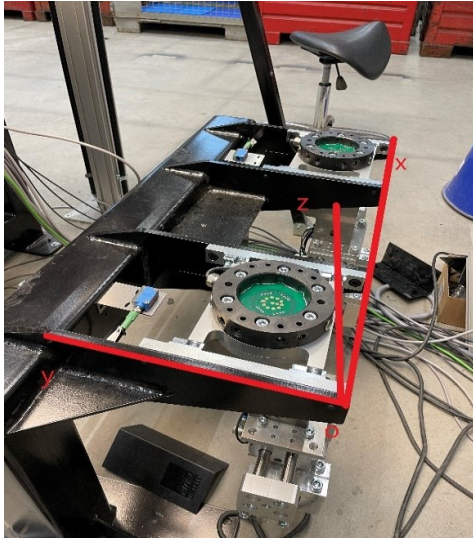
Kuvissa esitettynä tässä työssä käytettävät kehykset sekä niiden koordinaatistot (KUVA 23, KUVA 24, KUVA 25, KUVA 26).



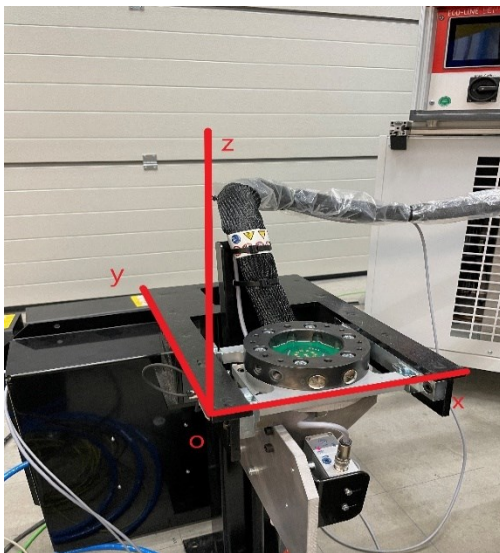
KUVA 23. Väliaseman kehys.



KUVA 24. Kuljettimien kehys.



KUVA 25. Työkalutelineet 1 ja 2.



KUVA 26. Induktiotyökalun teline.

Kun koordinaatistot on valittu, alkaa varsinainen opetus. Käyttäjäkehysten opetus tapahtuu myös kolmen pisteen avulla, mutta tässä tapauksessa pisteet ovat origo sekä X ja Y piste. Opetuksen ensimmäisessä vaiheessa piikki ajetaan halutun kehysten origoon (KUVA 27).



KUVA 27. Opetuspiikki origossa.

Kun piikki on riittävän tarkasti halutun kehyksen origossa, tallennetaan se robotille. Sama toistetaan sekä X, että Y pisteelle. Opetuksessa täytyy kiinnittää huomiota, että valitut pisteet ovat tarkkoja. Useassa tapauksessa valituissa pisteissä ei ole tarkkaa kiintopistettä, johon piikin voisi suoraan kohdistaa. Näissä tapauksissa voi apuna käyttää esimerkiksi mattopuukon terää (KUVA 28).



KUVA 28. Opetusprosessi.

Opetuksen viimeisessä vaiheessa täytyy vielä testata, onko opetus onnistunut ja liikkuuko robotti suunnitellun koordinaatiston mukaisesti. Tämä tapahtuu valitsemalla opetettu kehys ja ajamalla robottia jonkin akselin mukaisesti ja katsomalla mihin robotti liikkuu. Mikäli robotti liikkuu kehyksen reunojen suuntaisesti, mutta ei koordinaatiston mukaisesti voidaan suunnat korjata koordinaatiston mukaiseksi samalla lailla kuin työkalujen kehyksien koordinaatistojen korjauksessa eli muokkaamalla W, P tai R arvoa esimerkiksi 90 tai 180 astetta. Samat vaiheet toistetaan jokaiselle kehykselle ja kuten työkalujen opetuksessa, tulee ne dokumentoida hyvin.

Yhtenä muista poikkeavana kehyksenä oli kameran kehyksen luominen. Tämä tapahtui asettamalla kalibroituruudukko kameran alle ja luomalla tälle oma käyttäjäkehys, jonka avulla pystyttiin liittämään kamera sekä solu samaan koordinaatistoon.

5.3 Ohjelma

5.3.1 Rakenne

Kun aloitetaan ohjelmoimaan robottia, on robotin ohjelmalle hyvä luoda ensimmäisenä rakenne. Robotti saa tehtävänumeron ohjelmoitavalta logiikalta ja ohjelman rakenne luodaan tämän perusteella. Kun robotti on suorittanut logiikalta saadun tehtävän, palaa se takaisin pääohjemaan odottamaan uutta tehtävää.

Kuvassa esitettynä osa pääohjelman koodista, josta näkee, kuinka ohjelman rakenne on luotu (KUVA 30).

```
13: !Valitaan tehtava ;
14: SELECT R[1:Task]=100,CALL KPL_PICKPLACE ;
15:     =200,CALL HOLKKI_PICKPLACE ;
16:     =450,CALL KPL_PICKTOREG ;
17:     =300,CALL READY_MC_PICK ;
18:     =600,CALL INDUCTION_HEAT ;
19:     =300,CALL READY_MC_PICK ;
20:     =700,CALL READY_REG_PLACE ;
21:     =800,CALL RDY_PLACE_CONV ;
22:     =900,CALL RDY_PLACE_MEAS ;
23:     =310,CALL RDY_RGRS_PICKPLACE_MEAS ;
24:     =320,CALL RDY_RGRS_PICKPLACE_CONV ;
25:     =910,CALL RDY_DIRECTPLACE_MEAS ;
26:     =810,CALL RDY_DIRECTPLACE_CONV ;
27:     ELSE,JMP LBL[99] ;
28: LBL[99] ;
```

KUVA 30. Ohjelman runko.

Käydään esimerkkinä läpi tehtävännumero 200. Kun pääohjelma on siirtynyt odottamaan tehtävää ohjelmoitavalta logiikalta ja saa tehtävän 200, siirtyy se tämän tehtävänumeron mukaiseen aliohjelmaan (KUVA 31).

Aliohjelman alussa tarkistetaan, että robotilla on oikea tarttuja kiinni, robotti on oikealla puolella solua sekä että kuljetin on syöttänyt kappaleen. Tämän jälkeen tapahtuu poiminta kuljettimelta, puolen vaihto työstökoneelle, työstökoneelle syöttö ja lopuksi kotiasemaan ajo. Tämän ohjelman sisältä löytyy vielä useampia aliohjelmia, jotka on luotu, koska tämä selkeyttää huomattavasti ohjelman rakennetta ja helpottaa vianhakua (KUVA 31).

```

1: !Tarkistetaan tarttuja ;
2: R[21:req tool]=2 ;
3: IF R[20:act tool]<>R[21:req tool],CALL TOOLCHANGE ;
4: ;
5: !Tarkistetaan puoli ;
6: R[23:req side]=1 ;
7: IF R[22:act side]<>R[23:req side],CALL SIDECHANGE ;
8: ;
9: !Odotetaan Kuljetin ;
10: //WAIT DI[70:Infeed 2 stopped]=ON ;
11: ;
12: !Poimitaan Kuljettimelta ;
13: CALL HOLKKI_CNV_PICK ;
14: GO[2:Robot task number]=500 ;
15: ;
16: !Puolen vaihto koneelle ;
17: CALL SIDECHANGETOEMAG ;
18: ;
19: !Lasku MClle ;
20: CALL HOLKKI_MC_PLACE ;
21: ;
22: !Kotiin ajo ;
23: J PR[1:Home] 30% FINE ;
24: ;

```

KUVA 31. Holkin poimintaohjelma.

5.3.2 Liikeradat ja poimintapisteet

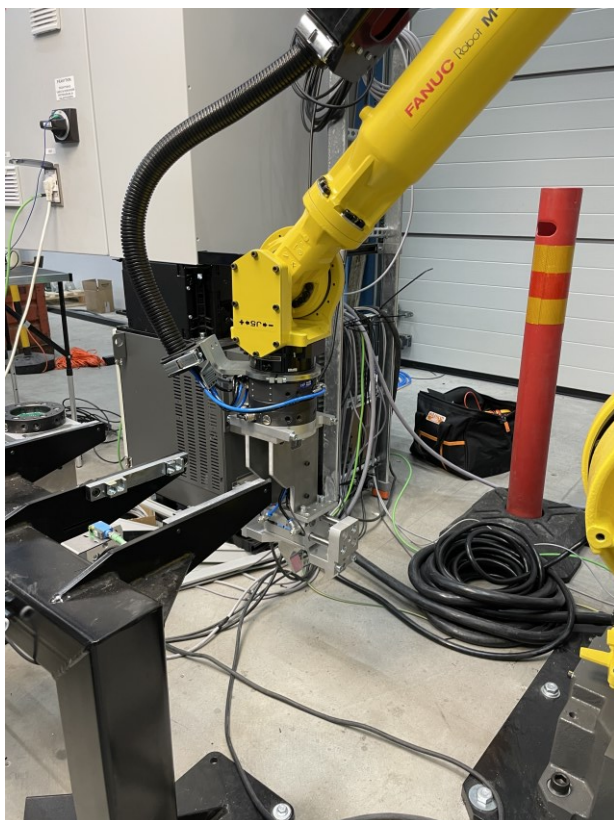
Käydään läpi yksi edellisessä kohdassa mainitusta aliohjelmasta (KUVA 32) sekä miten liikeradat tähän robottisoluun suunniteltiin.

Ohjelman alussa kutsutaan oikeat kehykset (työkalu, käyttäjä), jonka jälkeen toteutetaan puolen vaihto (KUVA 32). Liikeradan opettelu tapahtuu ajamalla robotti käsin pisteisiin, jonka kautta robotti halutaan kulkevan toiselle puolelle ja tämän jälkeen tallentamalla ne robotin muistiin. Nyt kun robottia pyydetään vaihtamaan puolta, vaihtaa se puolen näiden pisteiden kautta. Tässä kohtaa tulee olla tarkkana, että robotti kulkee kaikissa tilanteissa haluttua reittiä, jotta törmäyksiä ei synny.


```
1: !Asetetaan framet ;
2: CALL FRAMES(6,5) ;
3: ;
4: !Puolen vaihto koneelle ;
5:J PR[15:conveyer] 100% FINE ;
6:J P[2:sidepos] 100% CNT50 ;
7:J P[4:up] 100% CNT50 ;
8:J P[1:turn] 100% CNT50 ;
9:J PR[1:Home] 100% CNT100 ;
10: ;
11: !Vaihdetaan työkalu takaisin ;
12: UTOOL_NUM=R[20:act tool] ;
13: ;
14: !Asetetaan oikea puoli ;
15: //R[22:act side]=2 ;
16: ;
```

KUVA 32. Puolen vaihto työstökoneelle.

Samalla tavalla opetettiin liikeradat myös työkalujen poimintaan. Kuvasta näemme robotin poistumisreitit työkalun poiminnasta (KUVA 33). Juuri tällaisissa tilanteissa on hyvin tärkeää valita liikepisteet tarkkaan. Opetus- ja testitilanteissa on myös hyvä muistaa valita hiljaiset nopeudet robotille, jotta törmäyksiltä vältytään.



KUVA 33. Robotti poistumisreitti työkalupoiminnasta.

Yksi ongelmallinen kohta liikeratojen suunnittelussa oli siirtyminen väliasemaan. Kuten aikaisemmin mainittu robottisolu oli hyvin ahdas ja tässä vaiheessa täytyi olla tarkkana, jotta robotti ei menisi turva-alueen ulkopuolelle ja aiheuta törmäyksiä (KUVA 34).



KUVA 34. Robotin siirto väliasemaan.

Poiminta- sekä jättöpisteiden opetus robotille tapahtuu hyvin saman tapaisesti kuin liikeratojen opettelu. Käydään seuraavaksi läpi kappaleen poiminta työstökoneelta (KUVA 35). Työstökoneelta poiminta on toteutettu ajamalla robotti pisteeseen, missä poiminta halutaan suoritettavan, jonka jälkeen robotin tarttuja suljetaan ja siirrytään kappale tarttujassa suunniteltua liikerataa pitkin ulos työstökoneesta.

```

1: !Asetetaan framet ;
2: UFRAME_NUM=8 ;
3: UTOOL_NUM=1 ;
4: ;
5: !Siirrytaan poimintaan ;
6:J PR[1:Home] 100% FINE ;
7:J P[3:turn] 100% CNT100 ;
8:J P[2:above] 100% CNT100 ;
9:L P[1:pickpos] 100mm/sec FINE Offset,PR[55:kpl z offset] ;
10: ;
11: !Poiminta ;
12: RO[3:Irroitus]=OFF ;
13: RO[2:Tartunta]=ON ;
14: WAIT 1.00(sec) ;
15: DO[65:Part picked]=PULSE,2.0sec ;
16: ;
17: !Tarkastetaan tartunta ;
18: CALL CHECKGRIP ;
19: ;
20: !Takaisin kotiasemaan ;
21:L P[1:pickpos] 100mm/sec FINE Offset,PR[55:kpl z offset] ;
22:J P[2:above] 100% FINE ;
23:J P[3:turn] 100% CNT100 ;
24:J PR[1:Home] 100% CNT100 ;

```

KUVA 35. Poiminta työstökoneelta.

Muista poiminta- ja jättöpisteistä poiketen kappaleen sekä holkin poiminta toteutettiin tässä ohjelmassa konenäön avulla. Kun poimitaan konenäön avulla, täytyy poimittavalle kappaleelle luoda ensin poimintaohjelma, jonka jälkeen kamera pystyy löytämään tämän kappaleen kuljettimelta. Kun ohjelmat on luotu, voidaan kameraa käskä ohjelman puolella suorittamaan kuvaus, jonka jälkeen kamera antaa robotille kappaleen sijainnin kuljettimella, jonka perusteella poiminta suoritetaan. Kuvassa esitettynä esimerkki holkin poimintaohjelmasta (KUVA 36).

```

1: !Asetetaan Framet ;
2: UFRAME_NUM=2 ;
3: UTOOL_NUM=2 ;
4: ;
5: !kaanto ;
6:J PR[15:conveyer] 100% FINE ;
7: DO[60:Kuvausvalo paalle]=ON ;
8:J PR[16:horizontal] 100% CNT50 ;
9: ;
10: !Kuvaus ;
11: VISION RUN_FIND 'HOLKKI_KUVAUS' ;
12: VISION GET_OFFSET 'HOLKKI_KUVAUS' VR[2] JMP LBL[99] ;
13: PR[31]=VR[2].FOUND_POS[1] ;
14: //DO[60:Kuvausvalo paalle]=OFF ;
15: ;
16: !Korjausarvot ;
17: PR[31,6:holkkicam]=0 ;
18: PR[31,3:holkkicam]=0+R[56:holkkiz] ;
19: ;
20: !Tarkastetaan kuvausalue ;
21: CALL CHECK_HOLKKI_AREA ;
22: ;
23: !Poimintaan ajo ;
24:J P[2:front] 100% FINE Offset,PR[31:holkkicam] ;
25:L P[1:pick] 100mm/sec FINE Offset,PR[31:holkkicam] ;
26: ;
27: !Poimitaan holkki ;
28: RO[3:Irroitus]=OFF ;
29: RO[2:Tartunta]=ON ;
30: WAIT 1.00(sec) ;
31: DO[65:Part picked]=PULSE,2.0sec ;
32: ;
33: !Takaisin keskiasemaan ;
34:L P[1:pick] 100mm/sec FINE Offset,PR[31:holkkicam] Tool_Offset,PR[40:Nousuz] ;
35:J PR[16:horizontal] 100% CNT100 ;
36:J PR[15:conveyer] 100% FINE ;

```

KUVA 36. Holkin poimintaohjelma.

5.3.3 Turvatoiminnot

Yleisten turvatoimintojen lisäksi kuten turvapiirit ja hätäseis, robotissa on useita turvallisuustekijöitä, joita tulee ottaa huomioon. Käydään niitä seuraavaksi läpi.

Koska robotti ei mahtunut liikkumaan kuljettimilta työstökoneelle suoraan, tähän työhön täytyi suunnitella puolen vaihto robotille. Puolen vaihdon liikeratojen suunnittelu käytiin läpi jo aikaisemmin, mutta ohjelmaan tarvitsi myös luoda tarkastelu, joka kertoo, kummalla puolella solua robotti tällä hetkellä liikkuu. Tämä tapahtui tallentamalla robotin puoli muistipaikkaan, jonka jälkeen, kun puolen vaihto-ohjelma suoritettiin, kirjoitettiin puolen numero muistipaikkaan. Lisäyksenä tähän täytyy vielä tehdä tarkastelu, joka kertoo robotin koordinaatiston mukaan, kummalla puolella robotti oikeasti on, jotta vältetään törmäykset. Kuvassa esitettynä kuinka ylös robotin täytyi nousta, jotta se ei törmäisi turva-aitoihin puolen vaihdon yhteydessä (KUVA 37).



KUVA 37. Puolen vaihto työstökoneelle, yläasento.

Yksi turvatoiminto täytyi luoda tarttujen poimintaan. Likaisissa olosuhteissa tai jonkin vioittuneen osan takia robotin tarttuja ei välttämättä aina ole tartunnassa kunnolla kiinni, joten tähän täytyi luoda varmistus takaisinkytkennän avulla. Robotin tarttujissa on kiinni sekä auki rajat, joiden avulla pystyttiin luomaan takaisinkytkentä robotille. Mikäli toinen näistä tiedoista tuli perille, robotille voitiin todeta, että tartunta on onnistunut. Kuvassa esitettynä tartunnan tarkastelu, joka tehtiin aina työkalun poiminnan jälkeen (KUVA 38).

```

1: !Tarkastetaan tartunta ;
2: IF (RI[2:Tartunta kiinni]=ON OR RI[3]=ON) THEN ;
3: UALM[5] ;
4: PAUSE ;
5: ENDIF ;

```

KUVA 38. Tartunnan tarkastus.

Vaikka kameraohjelman avulla saadaan asetettua rajat, jonka ulkopuolelle robotin tarttuja ei saa poiminnassa liikkua. On hyvä asettaa myös fyysiset rajat robotille. Tämä tapahtui ajamalla robotti alueen äärirajoihin ja tallentamalla ne robotin muistiin. Mikäli robotti saa kuvausohjelmalta käskyn mennä alueen ulkopuolelle ilmoittaa robotti virheestä. Kuvassa esitettynä yhden kuvausalueen rajat (KUVA 39).

```

1: !Tarkastetaan kuvausalue y ;
2: IF (PR[30,2:kplcam]<(-80) OR PR[30,2:kplcam]>(133)) THEN ;
3: JMP LBL[99] ;
4: ENDIF ;
5: ;
6: !Tarkastetaan kuvausalue x ;
7: IF (PR[30,1:kplcam]<(-218) OR PR[30,1:kplcam]>(-140)) THEN ;
8: JMP LBL[99] ;
9: ENDIF ;
10: ;
11: !Tarkastetaan kuvausalue z ;
12: IF (PR[30,3:kplcam]<(14.5) OR PR[30,3:kplcam]>(150)) THEN ;
13: JMP LBL[99] ;
14: ENDIF ;

```

KUVA 39. Kuvausalueen rajat.

Yhtenä tärkeänä turvallisuustekijänä on miettiä mitä tapahtuu turvapiirien lauetua. Tätä ominaisuutta tulee pohtia jokaisessa solussa erikseen. Esimerkiksi tässä solussa turvapiirien lauettua robotille jätetään paineilma päälle, koska paineen katoaminen aiheuttaisi työkalun putoamisen ja tämä työkalun putoaminen on suurempi vaara kuin päälle jätetyn paineen aiheuttama puristusvaara. Lisäksi työkalun putoaminen turvapiirin lauetessa tekisi solusta hyvin epäkäytännöllisen.

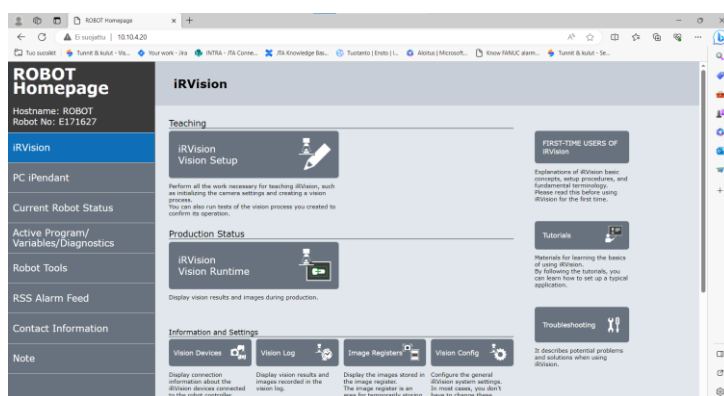
5.4 Konenäkö

Tässä kappaleessa käydään läpi yksi tapa luoda kameraohjelma. Ohjelmien luomisessa on useita eri tapoja. Myös valikoita ja asetuksia on useita, joista käydään seuraavaksi läpi vain tärkeimmät. Mikäli on tarve tarkemmalle tarkastelulle, voidaan halutun asetuksen vierestä valita puhekuplassa oleva i kirjain. Tätä painiketta painamalla aukeaa info-valikko, josta pystytään tarkastella asetuksia syvemmin.

5.4.1 Kappaleen ja holkin poiminnan opetus

Ensimmäisenä kameraohjelman luomisessa täytyy käytössä oleva tietokone liittää samaan verkkoon kameran ja robotin kanssa. Kun kone on liitetty samaan verkkoon, täytyy sille vielä asettaa tähän verkkoon sopiva IP-osoite. Kun yhteys on saatu kuntoon, yhdistetään kameraan kirjoittamalla sen IP-osoite verkkoselaimen osoitekenttään.

Tämän jälkeen selain yhdistää robottiin. Yhdistämisen jälkeen avataan vasemmasta sivupalkista iRVision. Nyt on päästy kameran asetusten päävalikkoon (KUVA 40). Mikäli halutaan nähdä kamera live-tilassa, voidaan avata iRVision Vision Runtime. Tässä tapauksessa, kun kameraohjelmaa ei ole vielä luotu avataan kuitenkin iRVision Vision Setup.



KUVA 40. Kameran päävalikko.

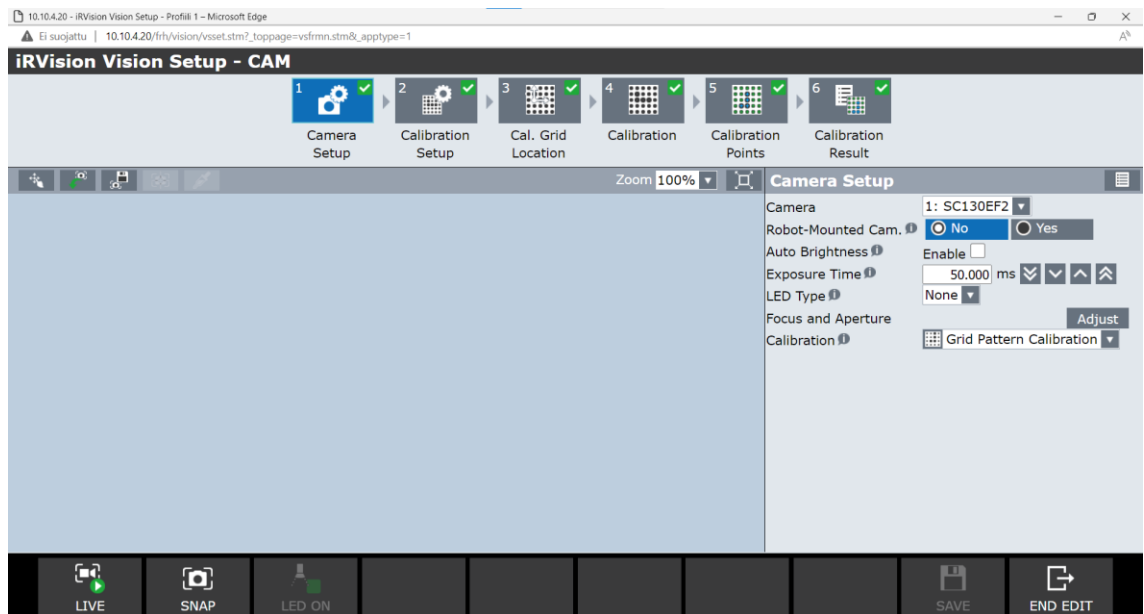
Seuraavassa valikossa avautuu ylös kamerrat ja näiden alapuolelle kameraohjelmat (KUVA 41). Kameroihin luodaan jokaiselle kameralle oma ohjelma, joita tässä tapauksessa oli vain yksi. Kameraohjelmiin puolestaan luodaan jokaiselle kuvattavalle kappaleelle oma ohjelma, mikäli ei ole mahdollista kuvata useita kappaleita samalla ohjelmalla. Jokainen kameraohjelma liitetään kameraan, joten ensimmäiseksi täytyy siirtyä luomaan kameralle ohjelma.

The screenshot shows the 'iRVision Vision Setup' application interface. At the top, there is a header with the application name and a navigation bar. Below this, there are two main sections: 'Camera Data (2)' and 'Vision Process Tools (13)'. The 'Camera Data' section contains a table with columns for Name, Comment, Type, Created, Modified, and Size. The 'Vision Process Tools' section contains a list of tools with columns for Name, Comment, Type, Created, Modified, and Size. At the bottom of the application, there is a toolbar with icons for CREATE, EDIT, FILTER, COPY, DETAIL, and DELETE.

Name	Comment	Type	Created	Modified	Size
Camera Data (2)					
CAM		2D Camera	08-AUG-2023 09:22:34	16-AUG-2023 09:46:38	4444
CAM_CAL		2D Camera	05-SEP-2023 10:20:00	05-SEP-2023 12:38:44	4444
Vision Process Tools (13)					
HOLKKITUNNISTUS		Depalletizing Vision Process	17-AUG-2023 08:46:00	06-SEP-2023 13:27:50	29594
HOLKKITUNNISTUSVANHA		2-D Single-View Vision Process	14-AUG-2023 15:10:34	17-AUG-2023 10:32:18	34009
HOLKKI_1		Depalletizing Vision Process	05-SEP-2023 13:39:00	05-SEP-2023 13:44:36	14810
HOLKKI_2		Depalletizing Vision Process	05-SEP-2023 13:39:12	05-SEP-2023 13:56:10	16818
HOLKKI_3		Depalletizing Vision Process	05-SEP-2023 13:39:26	05-SEP-2023 13:59:38	28081
HOLKKI_KUVAUS		Depalletizing Vision Process	06-SEP-2023 13:31:18	06-SEP-2023 13:34:52	29344
KAPPALETUNNISTUS		Depalletizing Vision Process	17-AUG-2023 12:47:02	05-SEP-2023 08:25:46	57449
KAPPALETUNNISTUSVANHA		2-D Single-View Vision Process	14-AUG-2023 15:10:34	17-AUG-2023 08:45:48	38743
KPL_1		Depalletizing Vision Process	05-SEP-2023 12:47:08	05-SEP-2023 13:20:40	20869
KPL_2		Depalletizing Vision Process	05-SEP-2023 13:03:44	05-SEP-2023 13:20:56	16736
KPL_3		Depalletizing Vision Process	05-SEP-2023 13:25:28	05-SEP-2023 13:30:12	33348
KPL_4		Depalletizing Vision Process	05-SEP-2023 13:30:24	05-SEP-2023 13:35:50	25174
KPL_KUVAUS		Depalletizing Vision Process	06-SEP-2023 12:54:00	07-SEP-2023 12:17:26	24335

KUVA 41. Kameran ja kameraohjelmat.

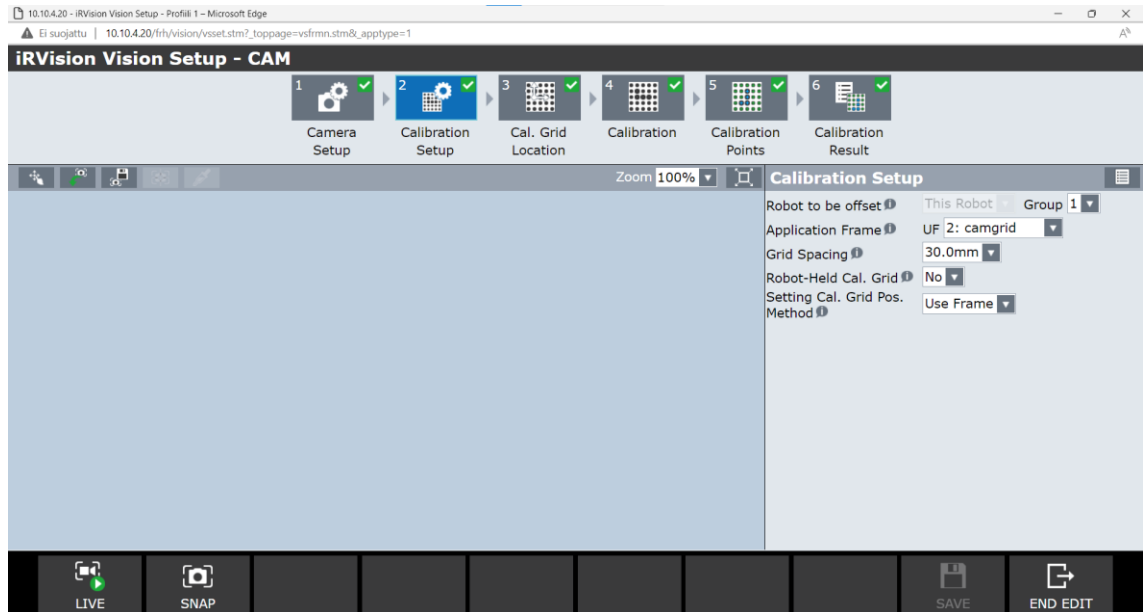
Kameran luomiseen ja kalibrointiin tarkoitettu valikko on kuusi valikkoa. Ensimmäisessä valikossa valitaan käytössä oleva kamera sekä kalibrointitapa. Valikosta löytyy myös muita asetuksia esimerkiksi kameran valotusaika (KUVA 42).



KUVA 42. Kameran lisääminen valikko 1.

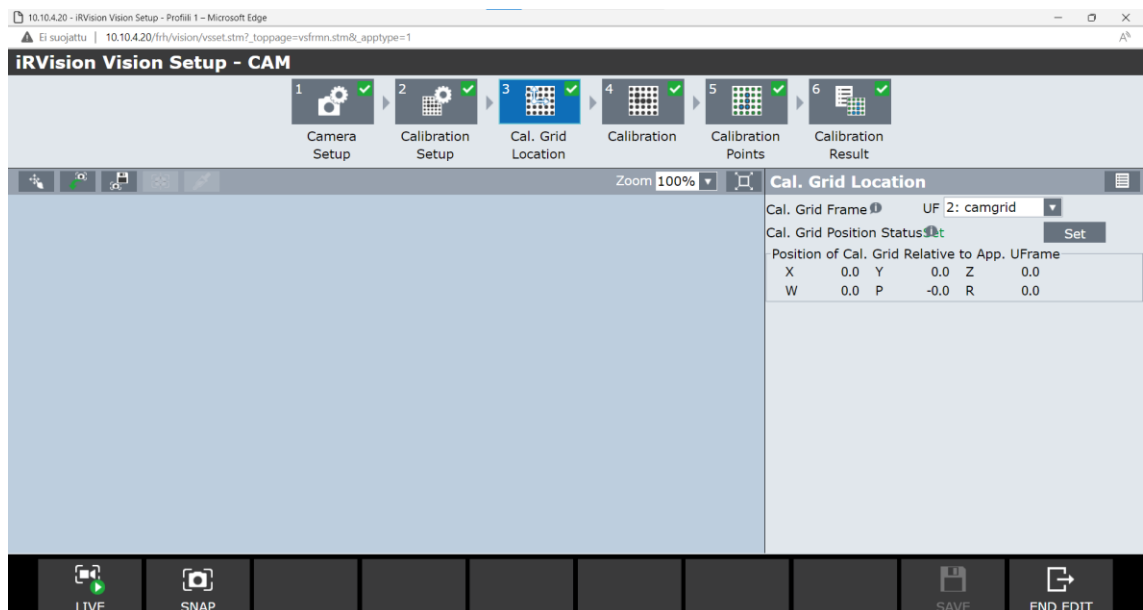
Valikossa kaksi on kalibrointiasetuksia joista tärkein on käytettävän kehyksen asettaminen. Kameraan luotiin aikaisemmin käyttäjäkehys, johon kamera

liitetään tässä vaiheessa. Valikossa asetetaan myös käytetyn ruudukon koko (KUVA 43).



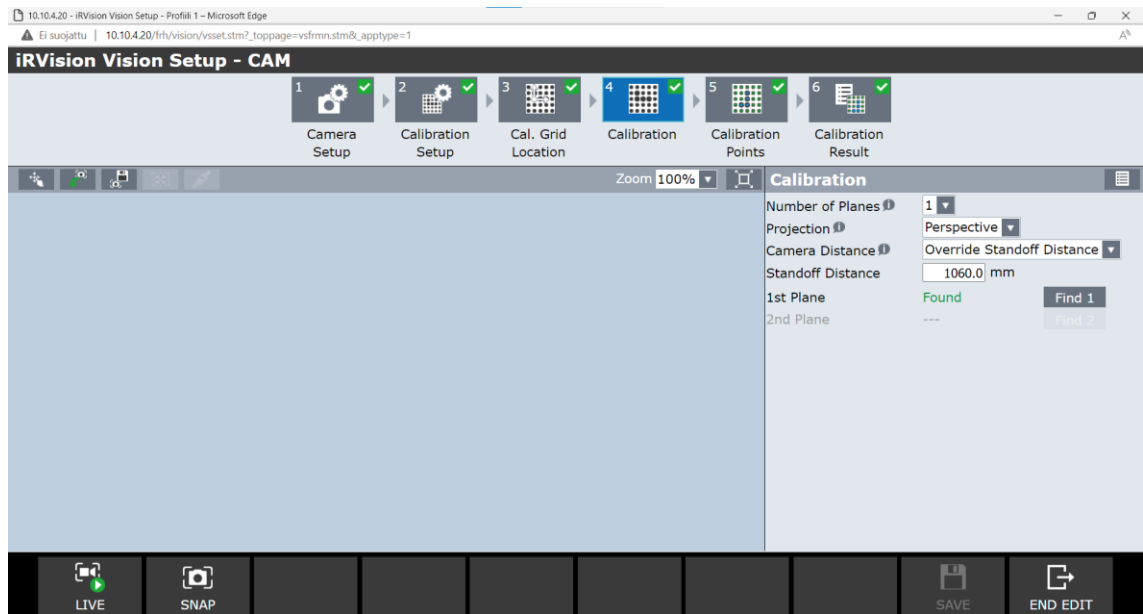
KUVA 43. Kameran lisääminen valikko 2.

Valikossa kolme määritellään ruudukon sijainti (KUVA 44). Tässä tilanteessa ruudukko on kameran alla täsmälleen samassa kohtaa kuin se oli kehyksen luomisen aikana, jotta virheitä ja ruudukon siirtymiseltä vältyttäisiin, luodaan kameralle ohjelma yleensä välittömästi kehyksen luomisen jälkeen.



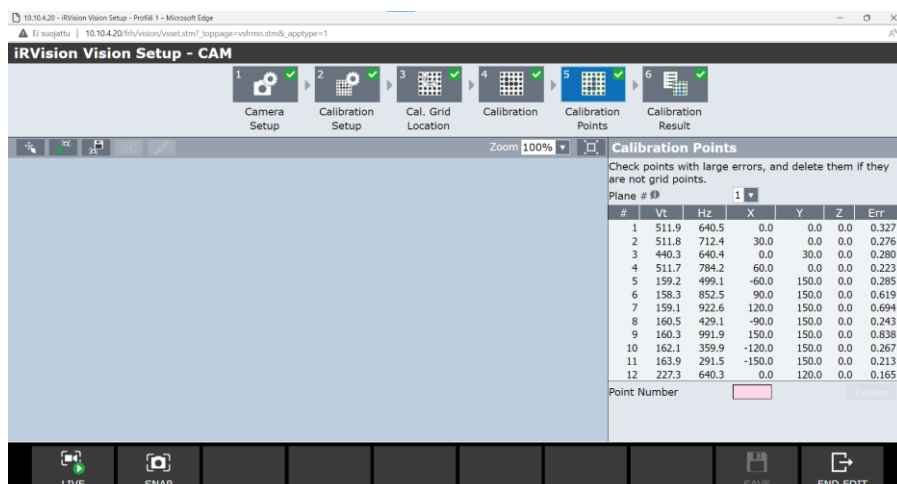
KUVA 44. Kameran lisääminen valikko 3.

Valikossa neljä määritellään kameran etäisyys kalibroitiruudukkoon (KUVA 45). Tämä tapahtuu mittaamalla etäisyys rullamitalla kameran linsistä ruudun pintaan ja asettamalla se asetuksiin.



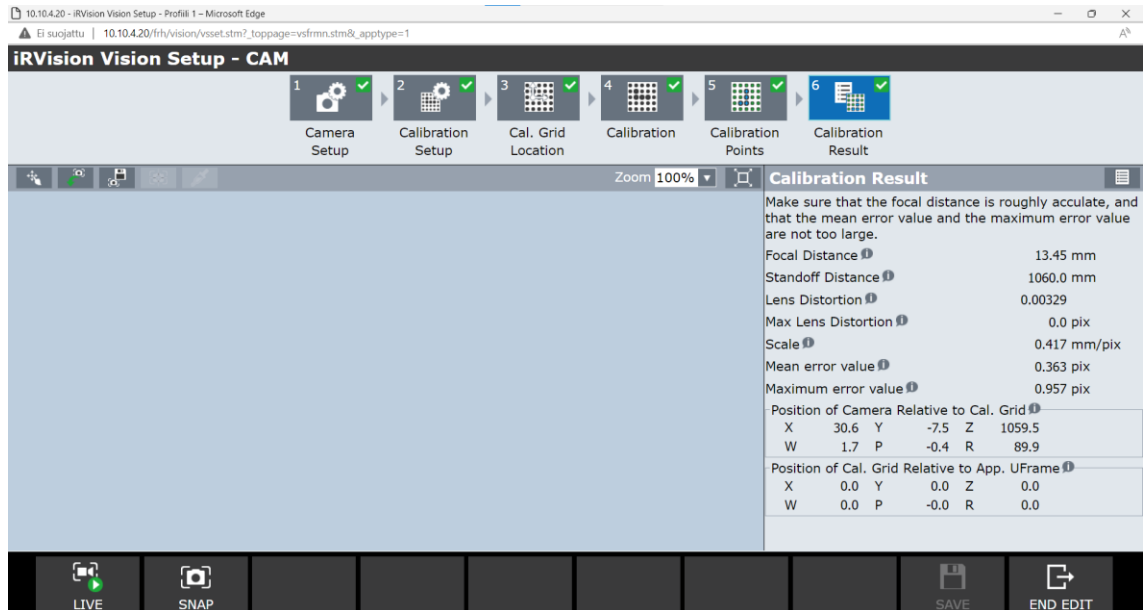
KUVA 45. Kameran lisääminen valikko 4.

Valikossa viisi nähdään, onko kameran kalibrointi onnistunut (KUVA 46). Kamera näyttää pisteet ja kuinka suuri virhe mittauksessa on tullut. Mikäli on paljon suuria virheitä täytyy kalibrointi suorittaa uudestaan, mutta mikäli jokin yksittäinen piste aiheuttaa jostain syystä suurta virhettä, voidaan se poistaa käytöstä. Näytölle ilmestyy myös kuva ruudukosta sekä mitatuista pisteistä, josta näkee onko kamera valinnut vahingossa pisteeksi jonkin pisteen ruudun ulkopuolelta, joka aiheuttaa virhettä kalibroinnissa. Myös nämä pisteet tulee poistaa.



KUVA 46. Kameran lisääminen valikko 5.

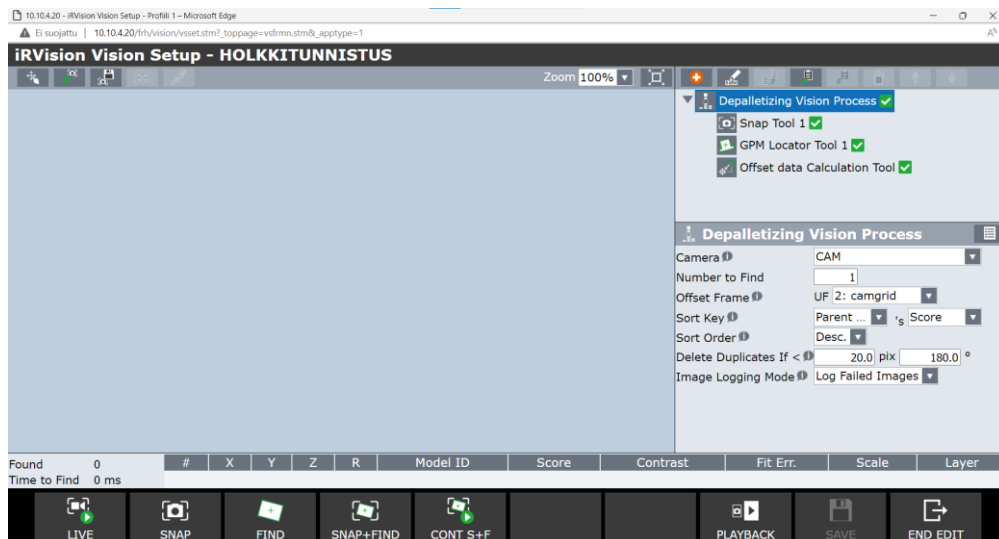
Viimeinen valikko toimii yhteenvetovalikkona, joka näyttää asetukset sekä virheen kalibroinnissa (KUVA 47).



KUVA 47. Kameran lisääminen valikko 6.

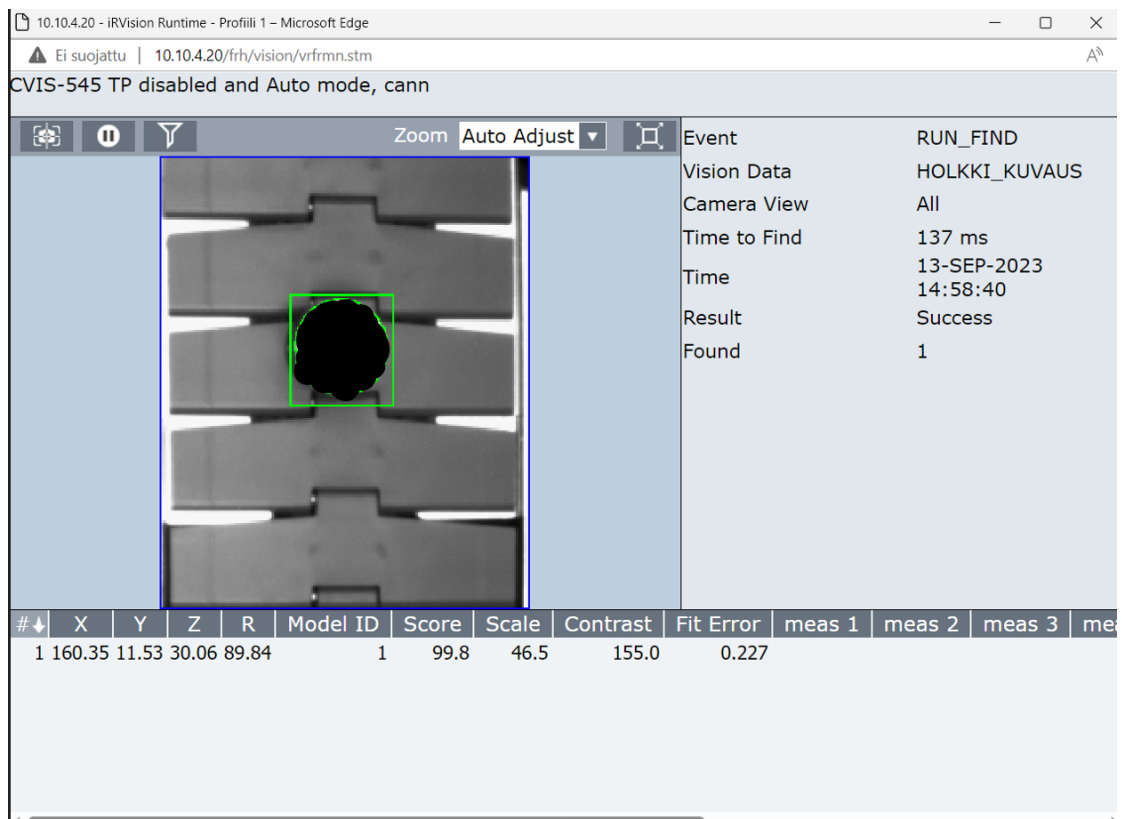
Kun kamera on saatu kalibroitu oikein sekä lisätty järjestelmään, luodaan poiminnalle kameraohjelma. Jokaiselle kappaleelle täytyi luoda oma ohjelma, joista meidän toimesta opetettiin muutama. Tämän jälkeen asiakkaalle opetettiin, kuinka kameraohjelma luodaan, jotta he voivat jatkossa luoda niitä lisää.

Kameraohjelman lisääminen alkaa päävalikosta (KUVA 41). Valitaan uuden ohjelman luominen, jonka jälkeen ohjelmalle valitaan nimi ja tyyppi. Ohjelmatyyppejä on useita erilaisia, mutta tässä tapauksessa keskitytään ohjelmaan nimeltä Depalletizin Vision Process. Kun ohjelmapohja on luotu, aukeaa valikko, josta valitaan juuri luotu kamera sekä käyttäjäkehys (KUVA 48).



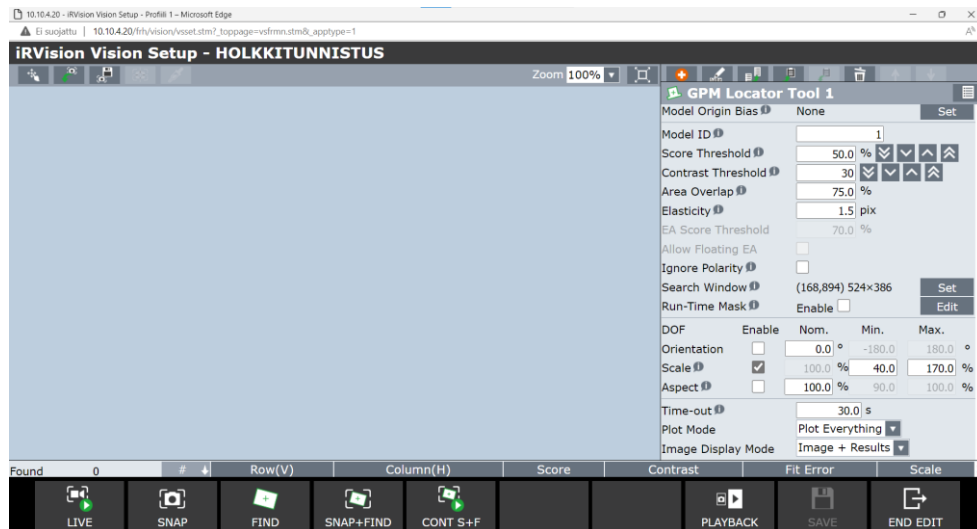
KUVA 48. Kameraohjelman luominen valikko 1.

Seuraavassa valikossa määritetään tunnistettava kappale. Asetetaan kameran alle haluttu kappale, jonka jälkeen se kuvataan. Kuvauksen jälkeen ohjelma piirtää kappaleeseen vihreällä tunnistetut alueet. Tähän alueeseen saattaa tulla mukaan ylimääräisiä alueita, tai sieltä saattaa puuttua jotain. Tästä syystä alueesta rajataan käsin pois kaikki ylimääräinen ja jätetään haluttu tunnistettava alue (KUVA 49).



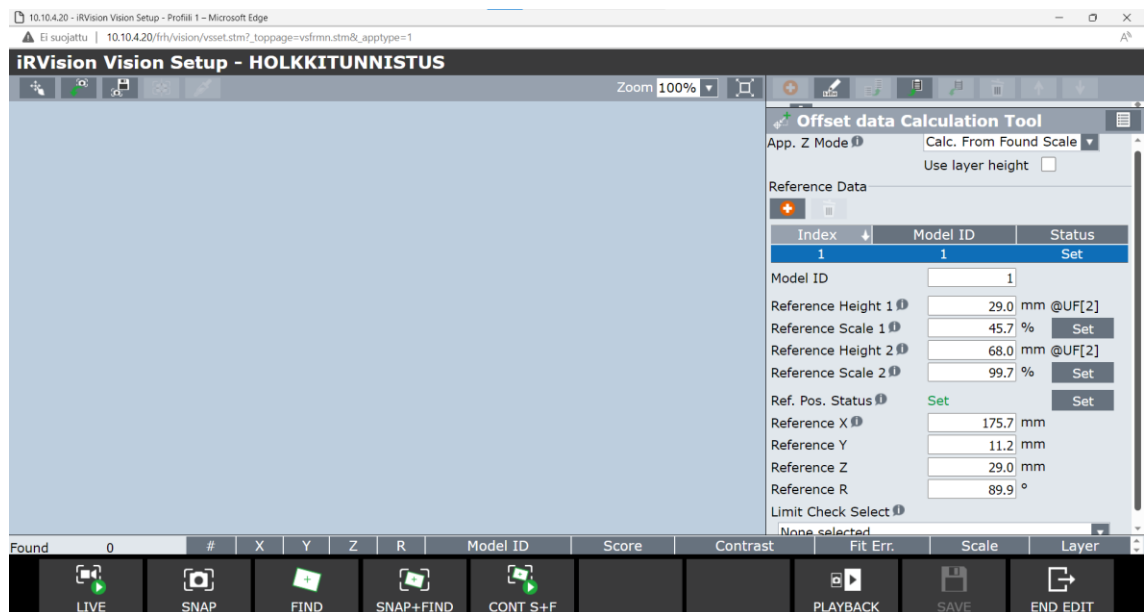
KUVA 49. Kameraohjelman luominen valikko 2.

Seuraavassa valikossa asetetaan asetukset kuvaukselle (KUVA 50). Valikosta saadaan säädettyä muun muassa miten tarkasti kappaleen täytyy vastata juuri kuvattua kappaletta, kontrastia, sekä kokoskaalausta. Kokoskaalauksella saadaan asetettua, kuinka paljon kokoeroa saa kuvatussa kappaleessa olla verrattuna kappaleeseen, jolla määrittys on suoritettu.



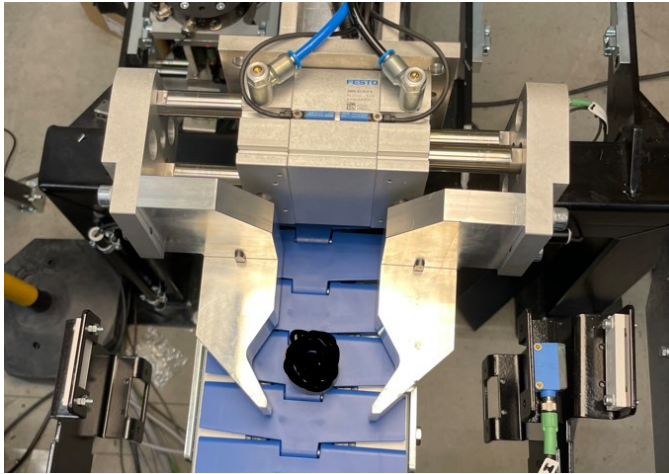
KUVA 50. Kameraohjelman luominen valikko 3.

Viimeisessä valikossa saadaan asetettua käsin kappaleen mitat ja kulma, jossa kappale on alustalla (KUVA 51). Kulman asetus on suurimmassa osassa tapauksissa 90 astetta, mutta poikkeuksiakin löytyy.



KUVA 51. Kameraohjelman luominen valikko 4.

Kun kameraohjelma on luotu, tarkastetaan, onko kuvaus onnistunut. Robotti ajetaan poimintaohjelmalla kuvattuun pisteeseen ja tarkkaillaan mihin robotti liikkuu. Robotti saattaa tässä tilanteessa törmätä, joten testausvaiheessa tulee olla tarkkana. Kuvasta voimme päätellä, että kuvaus ei täysin ole onnistunut, koska kappale ei ole täysin keskellä poimintaa (KUVA 52). Tämä johtui siitä, että kuljettimelle oli asetettu eri korkuinen kappale kuin ohjelmaan oli asetettu ja robotti luuli, että kappale on sivummassa. Kuvaus oli hyväksytty, kun oikea kappale asetettiin kuljettimelle.



KUVA 52. Robotti poiminnassa

5.5 Testaus

Kun robottisolun käyttöönoton jokainen vaihe oli suoritettu voitiin solun toiminta testata. Testauksessa robottisolun nopeudet asetettiin alhaiseksi, jonka jälkeen testattiin vaihe vaiheelta, jokaisen ohjelman toiminta. Mikäli tässä vaiheessa löytyi korjattavaa jossakin toiminnassa, palattiin tähän vaiheeseen sekä suoritettiin korjaukset. Kun toimintaan oltiin tyytyväisiä, lisättiin soluun nopeutta, jonka jälkeen testaukset suoritettiin uudestaan. Tämä toistettiin, kunnes solun toiminta täytti odotukset ja se voitiin todeta valmiiksi. Suurimpana korjauksena tässä työssä oli liikeratojen muokkaus sulavammaksi, jotta solun toimintaa saataisiin paremmaksi sekä nopeammaksi.

6 POHDINTA

Alkuperäisenä tavoitteena työlle oli luoda kehykset, jotta käyttöönottaja pystyisi jatkamaan työtä helpommin. Loppujen lopuksi robotille luotiin ohjelma lähes kokonaan ja tämän lisäksi tehtiin myös konenäköön testiohjelmat. Tästä voidaan todeta, että tehdyn työmäärän osalta työ onnistui hyvin.

Henkilökohtaisena tavoitteena oli oppia robotin toimintaa ja ohjelmointia, jossa myös onnistuttiin mielestäni todella hyvin. Työssä pääsi erinomaisesti tutustumaan robotin eri toimintoihin ja työskentelemässäni yrityksessä löytyi työntekijät, jotka opastivat aina tarvittaessa. Työtä sai tehdä myös kaikessa rauhassa mikä edisti omaa oppimista. Fanucin robotti ja ohjelmat vaikuttivat käyttäjäystävällisiltä ja oppiminen oli niiden osalta nopeampaa kuin odotin.

Työstä jäi varsinaiselle käyttöönottajalle käyttöliittymä, ohjelmoitavan logiikan ohjelmointi sekä muiden ohjelmien viimeistely. Lisäksi käyttöönottaja otti laitteen vielä käyttöön asiakkaalla. Tällä hetkellä laite on jo tuotantokäytössä ja toimii kuten pitääkin.

LÄHTEET

Pickit. 2023. Fanuc example user frame program. Verkkosivu. Viitattu 7.4.2024. https://docs.pickit3d.com/en/latest/robots/robot-brands/fanuc/fanuc_example_user_frame_program.html

JTA Connection Oy 2021a. JTA Connection Oy on perustanut tytäryhtiön JTA Connection OÜ .Verkkosivu. Viitattu 7.4.2024. <https://www.jtaconnection.fi/jta-connection-oy-on-perustanut-tytaryhtion-jta-connection-ou%ef%bf%bc/>

JTA Connection Oy 2021b. Teollisuuden kokonaisvaltainen automaatiokumppani. Verkkosivu. Viitattu 7.4.2024. <https://www.jtaconnection.fi/jta-connection-yrityksena/>

JTA Connection Oy 2021c. Expert Service eriytyy tytäryhtiöksi, uudeksi nimeksi JTA Expert Service Oy. Viitattu 7.4.2024. <https://www.jtaconnection.fi/expert-service-eriytyy-tytaryhtioksi-uudeksi-nimeksi-jta-expert-service-oy/>

Dan Sula 2023. Robot Frames as Street Addresses. Verkkosivu. Viitattu 7.4.2024. <https://www.pattiengineering.com/blog/robot-frames-street-addresses-2/>

Linda Rosencrance n.d. Software. Verkkosivu. Viitattu 7.4.2024. <https://www.techtarget.com/searcharchitecture/definition/software>
<https://education.vex.com/stemlabs/workcellv1/stemlab/manual-movements/robot-movement-and-robot-configurations> oikean käden sääntö

Wikipedia 2024. Teollisuusrobotti. Verkkosivu. Viitattu 7.4.2024. <https://fi.wikipedia.org/wiki/Teollisuusrobotti>

Suomen Robottiikkayhdistys ry 2023a. 2. Teollisuusrobotiikan sovelluksia. Viitattu 7.4.2024. <https://teollisuudenrobotiikka.fi/>

Laimonas Bačkys, Povilas Čepulkovskis ,Gintautas Dervinis, Laurent Daguet, Olivier Fortin, Olivier Fortier, Federica Gallicchio, Mika Heikkilä, Bastien Hervé du Penhoat, Sirkka-Helena, Ilveskoski Gené, Jakubauskienė, Ritva Klaavu, Marc Manguin, Bilel Miled, Ari Mäkinen, Dmitrij Novikov, Mindaugas Petravičius, Raimundas Petravičius, Pirjo Pietikäinen, Marjan Ranogajec, Ari Rannisto, Christian Raoelison, Jolanta Sakalauskienė, Živilė Šatienė, Edita Šidlauskaitė, Jarmo Tikka, Kęstutis Viselga, Gražina Žardalevičienė 2021. Opettajan kirja Osa 10. Digitaalisesti ohjattavassa tuotannossa toimiminen. Eurooppa: UP-SKILL

Teito 2023. Konenäkö. Verkkosivu. Viitattu 8.4.2024. https://teito.com/brochures-for-new-products/konenako/?gclid=Cj0KCQjwiMmwBhDmARIsA-BeQ7xSVREZw1JSTUJ3Tk1BrKB4EjjNd355z369cIMOgme-hUNW0BEdeHvl8aAqJIEALw_wcB

Algol 2018. Konenäkö tarkistaa ja ohjaa. Viitattu 7.4.2024. <https://www.algol-technics.fi/artikkelit-ja-asiakastarinat/konen%C3%A4k%C3%B6-tarkistaa-ja-ohjaa>

Jacob Biba 2022, Top 8 Programming Languages. Verkkosivu. Viitattu 8.4.2024. <https://builtin.com/robotics/robotic-programming-language>

Fanuc 2024. M-710iC/50. Verkkosivu. Viitattu 8.4.2024. <https://www.fanuc.eu/it/en/robots/robot-filter-page/m-710-series/m-710ic-50>

Education Vex 2024. Manual movements. Verkkosivu. Viitattu 8.4.2024. <https://education.vex.com/stemlabs/workcellv1/stemlab/manual-movements/robot-movement-and-robot-configurations>