

Tuomas Leppälä

Etikettisolun robottisimulointi ja offline-ohjelmointi

Opinnäytetyö

Syksy 2015

SeAMK Tekniikka

Automaatiotekniikan koulutusohjelma



SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Tutkinto-ohjelma: Automaatiotekniikan koulutusohjelma

Suuntautumisvaihtoehto: Koneautomaatio

Tekijä: Tuomas Leppälä

Työn nimi: Etiketisolun robottisimulointi ja offline-ohjelmointi

Ohjaaja: Toni Luomanmäki

Vuosi: 2015

Sivumäärä: 65

Liitteiden lukumäärä: 0

Opinnäytetyön tavoitteena oli tutkia robotin offline-ohjelmointia Siemensin RobotExpert-ohjelmiston avulla. Offline-ohjelmointia sovellettiin etikettirobotisoluun, joka toimi osana paperirullien pakkaus- ja käsittelylinjastoa. Opinnäytetyön toimeksiantajana toimi Pesimal Oy.

Offline-ohjelmointi (OLP), toisin sanoen etäohjelmointi, on nykyaikaisin ja tehokkain tapa ohjelmoida robotti. Ohjelmointi tapahtuu simuloitussa solussa ulkoisella PC:llä, ilman fyysisen robotin irrottamista tuotannosta ohjelmoinnin ajaksi.

Tutkimuksessa perehdytään hieman simulointiin ja robottien ohjelmointiin, kerrotaan RobotExpert-ohjelmistosta yleisellä tasolla sekä tarkastellaan robotin simulointi- ja etäohjelmointiprosessin vaiheita

Avainsanat: offline-ohjelmointi, simulointi, robotiikka

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Automation Engineering

Specialisation: Machine Automation

Author/s: Tuomas Leppälä

Title of thesis: Robot simulation and offline programming of a label robot cell

Supervisor: Toni Luomanmäki

Year: 2015

Number of pages: 65

Number of appendices: 0

The aim of this study was to investigate robot offline programming with Siemens RobotExpert software. The offline programming was applied to a label robot cell, which operated as a part of a packaging and handling network of paper rolls. The thesis was made in cooperation with Pesimal Oy.

Offline programming (OLP), in other words, remote programming is the most modern and efficient way of robot programming. Programming is done in a virtually simulated cell with an external PC. The robot does not need to be separated from the production line during programming.

The study focuses on the simulation and programming of robots, and the Robot Expert software is introduced on a general level. The last paragraph goes through the process of robot simulation and offline programming.

Keywords: offline programming, robot simulation, robotics

SISÄLTÖ

| | |
|--|----|
| Opinnäytetyön tiivistelmä..... | 1 |
| Thesis abstract..... | 2 |
| SISÄLTÖ..... | 3 |
| Kuva-, kuvio- ja taulukkoluetelo..... | 5 |
| Käytetyt termit ja lyhenteet..... | 7 |
| 1 JOHDANTO..... | 8 |
| 1.1 Työn tausta..... | 8 |
| 1.2 Työn tavoite..... | 9 |
| 1.3 Työn rakenne..... | 9 |
| 2 PESMEL..... | 10 |
| 2.1 Material Flow How -konsepti..... | 10 |
| 2.1.1 Konseptin hyödyt..... | 11 |
| 2.1.2 Yrityksen historia..... | 12 |
| 3 SIMULOINTI..... | 13 |
| 3.1 Simulointi käsitteenä..... | 13 |
| 3.1.1 Simuloinnin hyödyt..... | 14 |
| 3.1.2 Simuloinnin heikkoudet..... | 15 |
| 3.2 Robottisimulointi..... | 16 |
| 4 TECNOMATIX ROBOTEXPERT -OHJELMISTO..... | 17 |
| 4.1 Ominaisuudet..... | 18 |
| 4.2 RRS- ja RCS-moduulit..... | 19 |
| 5 ROBOTIN OHJELMOINTI..... | 20 |
| 5.1 Teollisuusrobotin tyypilliset käskyt..... | 20 |
| 5.1.1 Käskyt ulkoisiin järjestelmiin..... | 21 |
| 5.1.2 Robotin valinta..... | 21 |
| 5.1.3 Liitynnät logiikkaan ja solun muihin toimilaitteisiin..... | 22 |
| 5.2 Ohjelmointitavat..... | 23 |
| 5.3 Online-ohjelmointi..... | 23 |
| 5.3.1 Opettamalla-ohjelmointi..... | 23 |
| 5.4 Offline-ohjelmointi..... | 24 |

| | |
|--|-----------|
| 5.4.1 Offline-ohjelmoinnin hyödyt..... | 24 |
| 5.5 Simuloinnin ja offline-ohjelmoinnin evoluutio | 25 |
| 6 OFFLINE-OHJELMOINTIPROSESSIN VAIHEET | 27 |
| 6.1 Lähtötietojen hankinta..... | 27 |
| 6.1.1 Simuloitavan ja etäohjelmoitavan robottisolun toimintakuvaus | 28 |
| 6.2 Solun 3D-mallinnus..... | 29 |
| 6.3 Solun toimilaitteiden paikoitus simulointimalliin layout-mittojen pohjalta ... | 32 |
| 6.4 Toimilaitteiden kinematiikan rakentaminen | 35 |
| 6.5 Robotin työkalujen määrittäminen ja kiinnittäminen | 38 |
| 6.5.1 Työkalun kiinnittäminen robottiin..... | 39 |
| 6.6 Robotin ja toimilaitteiden akselien liikuttaminen | 40 |
| 6.7 Liikekäskyjen rakentaminen..... | 42 |
| 6.7.1 Sequence Editor | 45 |
| 6.7.2 Path Editor | 47 |
| 6.7.3 Robotin konfiguraatio | 50 |
| 6.7.4 OLP-käskyjen lisääminen ohjelmaan | 51 |
| 6.8 Solun kalibrointi..... | 53 |
| 6.9 Robotin kontrollerin parametrien määrittäminen | 54 |
| 6.10 Ohjelman simulointi ja hienosäätö | 57 |
| 6.10.1 Törmäystarkastelu..... | 57 |
| 6.10.2 Ulottuvuuden tarkastelun työkalu | 58 |
| 6.10.3 Smart Place -työkalu | 59 |
| 6.10.4 Swept Volume -työkalu | 60 |
| 6.11 Ohjelman kääntäminen robotin käyttämään muotoon | 61 |
| 7 TULOKSET JA POHDINTA..... | 63 |
| LÄHTEET | 64 |

Kuva-, kuvio- ja taulukkoluetelo

| | |
|--|----|
| Kuvio 1. Paperinrullien pakkauslinjasto..... | 12 |
| Kuvio 2. Robottiohjelman kierrättäminen | 17 |
| Kuvio 3. Tecnomatix RobotExpert..... | 18 |
| Kuvio 4. Simuloinnin monimutkaistuminen..... | 26 |
| Kuvio 5. 3D-mallin konvertoiminen ja RobotExpertin tukemat formaatit..... | 29 |
| Kuvio 6. Valmiiden 3D-mallien lataaminen robottivalmistajan sivuilta | 30 |
| Kuvio 7. ABB IRB1600 -robotti..... | 31 |
| Kuvio 8. 3D-mallin tuominen simulaatioon | 32 |
| Kuvio 9. Etikettirobotin solu..... | 32 |
| Kuvio 10. Toimilaitteiden sijoittaminen layout-kuvan perusteella | 33 |
| Kuvio 11. Valmis robottisoluu..... | 34 |
| Kuvio 12. Kinematiikan rakentaminen..... | 35 |
| Kuvio 13. Kinematiikan viimeistely | 36 |
| Kuvio 14. Työkalun akselien määrittäminen..... | 37 |
| Kuvio 15. Työkalun määrittäminen | 38 |
| Kuvio 16. Työkalun kiinnittäminen robottiin..... | 39 |
| Kuvio 17. Robotin liikuttaminen Robot Jog -toiminnolla | 40 |
| Kuvio 18. Robotin liikuttaminen Joint Jog -toiminnolla | 41 |
| Kuvio 19. Operaatiopuu | 43 |
| Kuvio 20. Sekvenssieditori ennen linkitystä | 45 |

| | |
|---|----|
| Kuvio 21. Etikettisolun valmis sekvenssiesitys..... | 46 |
| Kuvio 22. Tool Data | 48 |
| Kuvio 23. Wobj Data | 48 |
| Kuvio 24. Robotin ohjelman avaaminen ja muokkaaminen Path Editorissa..... | 49 |
| Kuvio 25. Robotin konfiguraatio | 50 |
| Kuvio 26. OLP-komentojen lisääminen | 51 |
| Kuvio 27. OLP-komentojen lisääminen etikettirobotin ohjelmaan | 52 |
| Kuvio 28. Robotisoidun solun kalibrointi | 53 |
| Kuvio 29. Robot Properties -valikko | 55 |
| Kuvio 30. Controller settings -valikko | 55 |
| Kuvio 31. ABB Robot setup -valikko | 56 |
| Kuvio 32. Törmäystarkastelu..... | 57 |
| Kuvio 33. Etikettirobotille suoritettu ulottuvuustesti | 58 |
| Kuvio 34. Smart Place -toiminto..... | 59 |
| Kuvio 35. Swept Volume -toiminto | 60 |
| Kuvio 36. Program Inventory..... | 62 |
| Kuvio 37. Ote ladatusta robottiohjelmasta, käännettynä robotin kielelle | 62 |

Käytetyt termit ja lyhenteet

| | |
|---------------------|---|
| Frame | Frame tarkoittaa simuloinnissa kehystä tai koordinaatistoa. Simulaatiossa voidaan määrittää esimerkiksi työkalu-, työkohde- tai käyttäjäkoordinaatisto. |
| TCP | Tool Center Point eli robotin tai robotin työkalun keskipiste. |
| I/O-signaali | Input- ja output-signaalien eli tulo- ja lähtösignaalien määrittäminen. |
| OLP | Offline programming eli robotin etäohjelmointi ulkoisella PC:llä. |
| PLC | Programmable logic controller eli ohjelmoitava logiikka. PLC:llä voidaan ohjata automatisoituja laitteita. |

1 JOHDANTO

1.1 Työn tausta

Tämän opinnäytetyön kohdeyrityksenä toimii Pesmel Oy, Kauhajoelta. Yritys toimii monipuolisten automaatiojärjestelmien ja ratkaisujen toimittajana.

Robottien käyttö teollisuudessa lisääntyy koko ajan ja tuotantosolut monimutkaisuutensa vuoksi. Robotit ovat merkittävässä roolissa myös Pesmelin suunnittelemissa ja tuottamissa soluissa. Robotin ohjelmointi voidaan toteuttaa usealla eri tavalla. Yleisimmin ohjelmointi tapahtuu kuitenkin opettamalla. Opettamalla ohjelmoinnissa robotti ajetaan suunniteltuun paikkaan, jonka jälkeen robotin tarkat koordinaatit ja nivelten asennot tallennetaan muistiin. Tämän tapainen ohjelmointi on suhteellisen hidasta eikä aina sovellu markkinoiden suuriin kapasiteettivaatimuksiin.

Pesmelillä robottien ohjelmointi on yleisesti suoritettu opettamalla, eli Online-tilassa. Koko ajan yleistyvä etäohjelmointi eli offline-ohjelmointi(OLP) on tehokkain tapa hyödyntää robottien maksimikapasiteettia, vähentää tuotannon seisokkeja, ennaltaehkäistä laiterikkoja sekä suunnittelussa tapahtuvia virhelaskelmia. Tämän työn taustalla on etäohjelmoinnin hyötyjen ja haittapuolien löytäminen robottien ohjelmoinnissa, sekä samalla selvittää, miten Pesmel voisi offline-ohjelmoinnilla parantaa testauksen ja käyttöönoton tehokkuutta.

Etäohjelmointia testataan etiketti-robotisolussa aikatauluista riippuen. Etiketti-robotisolu on osa Pesmelin suunnitteleman paperirullien pakkauslinjaston kokonaisuutta. Simulointi ja etäohjelmointi toteutetaan Siemensin RobotExpert-ohjelmistolla.

1.2 Työn tavoite

Työn tavoitteena on tuottaa etikettirobotisolusta tarkka 3D-mallinnus, simuloida robotin ja muiden toimilaitteiden toiminnot, sekä tutustua offline-ohjelmointiin pääpiirteittäin.

Etäohjelmointia ei ole Pesmelillä varsinaisesti ennen testattu, tästä syystä tutkimuksen päätavoitteena on selvittää etäohjelmoinnin hyödyt verrattuna muihin ohjelmointitekniikoihin. RobotExpert on vain yksi tähän tarkoitukseen soveltuva ohjelma muiden ohjelmien joukossa. Tarkoituksena onkin verrata RobotExpert-ohjelmistoa samalla ABB:n omaan etäohjelmointialustaan RobotStudioon.

1.3 Työn rakenne

Työn ensimmäinen osio käsittelee tutkimuksen taustoja sekä tutkimuksen tavoitteita. Tämän jälkeen esitellään tutkimuksen toimeksiantajana toiminut yritys. Seuraavassa osiossa kerrotaan simuloinnista, robottisimuloinnista sekä RobotExpert-ohjelmistosta. Tutkimuksessa paneudutaan myös lyhyesti robotin ohjelmointitekniikoihin. Viimeisessä kappaleessa syvennyttään tarkemmin robotisoidun etikettisolun simulointi- ja etäohjelmointivaiheisiin.

2 PESMEL

Pesmel Oy, on maailman laajuisesti toimiva, laadukkaiden ja korkeasti automatisoitujen, tuotannon sisäisen logistiikan, varastoinnin ja pakkausjärjestelmien tuottaja. Tuotteet ja järjestelmät suuntautuvat pääasiassa paperi-, metalli- ja jatkojalostusteollisuuteen. Pesmelin palveluihin kuuluvat myös suunnittelu, asennukset, käyttöönotto, koulutus ja after sales -palvelut. (Pesmel, [Viitattu 19.8.2015].)

Pääkonttori sijaitsee Etelä-Pohjanmaalla Kauhajoen kaupungissa, ja yritys on yksi paikkakunnan suurimmista työllistäjistä. Kauhajoen yksikössä suoritetaan uusien automaatiojärjestelmien testaukset. Pesmelillä on myös sivukonttorit Seinäjoella, Tampereella ja Helsingissä. Suomen ulkopuolella on myös sivukonttoreita: Virossa, Yhdysvalloissa, Hong Kongissa ja Taiwanissa. Pesmel työllistää noin 160 ihmistä, joista 100 työskentelee Suomessa. (Pesmel, [Viitattu 19.8.2015].)

2.1 Material Flow How -konsepti

Material Flow How -konsepti perustuu ideaan, jossa koko tuotantoprosessin laitteisto on integroituna logistiseen järjestelmään. Konsepti tarjoaa yhtenäisen perustan kaikille materiaalinkäsittelyn toiminnoille sisältäen kuljettimet, lajittelun, varastoinnin, pakkaamisen ja kuljetuksen. Koko prosessi on kontrolloitu kehittyneellä WMS-tekniikalla (Warehouse Management System). (Pesmel, [Viitattu 19.8.2015].)

2.1.1 Konseptin hyödyt

Material Flow How -konseptin hyödyt tulevat ilmi seuraavassa luettelossa:

- Pakkauksella varmistetaan tuotteiden kestävyys sekä laatu. Pakkauksessa ei käytetä laisinkaan kemikaaleja, tällä varmistetaan pakkausmateriaalinen kierrätettävyys.
- Välivarastoinnilla tehtaan tuotantoprosessit integroidaan. Tuotteisiin sidottu pääoma pysyy kontrollissa, joka varmistaa tuotteiden saatavuuden prosessien välillä.
- Lähettämövarastolla varmistetaan asiakastyytyväisyys, lisätään tehokkuutta ja kilpailuetua kilpailijoihin nähden.

(Pesmel, [Viitattu 19.8.2015].)

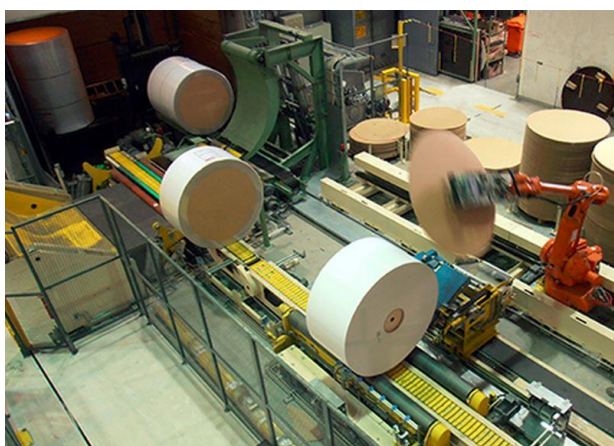
2.1.2 Yrityksen historia

Yrityksen perustivat veljekset Jari ja Hannu Mäki-Rahkola vuonna 1978. Ensimmäinen suurempi projekti Pesmelillä oli Ekokemi vuonna 1984. 1980-luvun aikana tuotevalikoima ja toimitukset laajenivat järjestelmiksi ja ensimmäinen pakkauslinja paperiteollisuuteen valmistui. (Pesmel, [Viitattu 19.8.2015].)

1990-luku oli suuren kasvun aikaa. Uusia yrityksiä syntyi ulkomaille ja uusia teollisuuden aloja tuli mukaan. Ensimmäinen pakkauslinja metalliteollisuuteen valmistui. Toinen tuotantoyksikkö, jossa nykyään testataan suurimmat projektit, perustettiin Eestiin vuonna 1996. (Pesmel, [Viitattu 19.8.2015].)

2000-luvulla käyttöön otettiin ISO 9001: 2000 -sertifikaatti. Kasvu metalliteollisuuden puolelle jatkui ja uusia tytäryhtiöitä syntyi maailmalle. Pesmel kehitti ensimmäisen silmästä käärintä -järjestelmän metalliteollisuuteen. Vuonna 2008 omistuspohjaa suurennettiin. Suurimpana osakkeenomistajana oli Suomalainen pääomasijoitusrahasto (60 %) ja loput, 40 % osakkeista kuului Mäki-Rahkolan veljesten omistukseen. Vuoden 2009 alussa AWA Oy, joka toimitti automaattivarastoja, fuusioitiin Pesmel Oy:n kanssa. (Pesmel, [Viitattu 19.8.2015].)

2010-luvulla Pesmel alkoi panostaa isommalla luottamuksella asiakkaiden logistiikka- ja pakkaustoimintoihin. (Pesmel, [Viitattu 19.8.2015].)



Kuvio 1. Paperinrullien pakkauslinjasto
(Pesmel 2015)

3 SIMULOINTI

3.1 Simulointi käsitteenä

Simulaatiomalleihin voidaan törmätä jokapäiväisessä elämässä. Simuloimalla pystytään esimerkiksi havainnollistamaan sään kehittyminen viikoiksi eteenpäin. Nykyään simulointi tapahtuu yleensä tietokonesimulointina, mutta fysikaalisia ilmiöitä voidaan simuloida myös ilman tietokoneita, vaikkapa pienoismallien avulla. Kaikkein yksinkertaisimmassa merkityksessä simulointi voidaan määritellä järjestelmän imitaationa. (Robinson 2004, 2.)

Simulointi voidaan jakaa kahteen eri malliin: staattinen simulointi, jossa järjestelmää kuvataan tietyllä ajan hetkellä, ja dynaaminen simulointi, jossa imitoidaan järjestelmän prosessia kokonaisuutena. (Robinson 2004, 2.)

Järjestelmä yleisessä merkityksessä tarkoittaa johonkin tarkoitukseen koottujen osien tai asioiden kokonaisuutta. Järjestelmät voivat olla luonnon aiheuttamia tapahtumia tai ihmisten suunnittelemlia toimintoja, teknisiä laitteita, tai tässä tapauksessa automatisoituja tuotantolaitoksia. (Robinson 2004, 2.)

Yhteenvetona voidaan todeta, että simuloinnin päätarkoitus on saavuttaa parempi ymmärrys järjestelmän toiminnasta, ja sitä kautta löytää kehitysideoita mahdolliseen reaali maailman toimintoon tai laitteeseen.

3.1.1 Simuloinnin hyödyt

Simuloinnilla on useita hyötyjä verrattuna fyysiseen testaamiseen. Simulointi ei tietenkään ole ainoa tapa analysoida ja parantaa systeemejä, mutta useimmiten systeemit ovat niin monimutkaisia, että simuloinnin hyödyt tulevat selvästi esille.

- Kustannukset nousevat korkealle, jos systeemejä testataan reaali maailmassa. Kustannuksia tulee lisää, jos reaali maailman systeemi pitää ottaa pois tuotannosta uuden järjestelmän suunnittelun ajaksi. Tuotanto viivästyy ja menetetään asiakkaita. Simulointia hyödyntämällä reaali maailman systeemi voidaan pitää suunnittelun ajan täydessä vauhdissa. (Robinson 2004, 31.)
- Reaali maailmassa testaamiseen saattaa kulua useita viikkoja. Simuloinnilla kulutettu aika voidaan joissain tapauksissa pienentää muutamaan päivään tai jopa muutamaan minuuttiin. (Robinson 2004, 31.)
- Testausolosuhteiden kontrollointi. Testatessa eri systeemivaihtoehtoja on hyödyllistä pystyä kontrolloimaan testauksen olosuhteita, että saataisiin luotettavaa tutkimustietoa eri prosesseista. Tämä on vaikeaa reaali maailmassa. Esimerkiksi testatessa sairaalan uutta järjestelmää, olisi mahdotonta kontrolloida potilasvirtaa sairaalaan. Simuloinnilla testauksia voidaan suorittaa rajaton määrä. (Robinson 2004, 31.)
- Testattava systeemi ei vielä ole olemassa. Ilmeisin ongelma reaali maailman systeemin testaamisessa on, että systeemiä ei ole vielä edes olemassa. Vaihtoehtoisten sistemien rakentaminen oikeassa maailmassa olisi epäkäytännöllistä. Ainoa vaihtoehto on tehdä systeemistä mallinnus tai simulaatio. (Robinson 2004, 31.)

3.1.2 Simuloinnin heikkoudet

Simuloinnista voidaan myös löytää ongelmakohtia, joita ei pidä jättää huomiotta kun punnitaan systeemin analysointi- ja testausvaihtoehtoja.

- Arvokas ohjelmisto. Simulointiohjelmit ovat usein erittäin kalliita ja ohjelmit vaativat myös kalliita koulutuksia niiden käyttämiseen.
- Ajallinen miinus. Simulointi voi toisinaan viedä paljon aikaa ja hyödyt eivät ole aina heti nähtävissä.
- Tiedostojen suuri koko. Usein simulaatiomallit vaativat suuria määriä dataa.
- Tieto-taito. Simulaatio-ohjelmit ja mallintaminen vaativat tekijältään ymmärrystä tuotteiden ja rakennelmien mekaanisesta suunnittelusta.
- Itsevarmuus. Saattaa olla tilanne, jossa simulaation tekijä luottaa liikaa simuloidun mallin toimimiseen reaali maailmassa.

(Robinson 2004, 33–34.)

3.2 Robottisimulointi

Yrityksillä on jatkuva tarve kehittää teknologiaa, prosesseja ja käytäntöjään, pysyäkseen kilpailukykyisinä koko ajan kasvavilla markkinoilla. Yksi tärkeimmistä kehityskohteista on tuotannon nopeuttaminen ja joustavuuden lisääminen. Samalla myös tuotannosta aiheutuvat kustannukset pyritään pitämään alhaisina ja tuotteiden laatu korkealla. (Siemens PLM 2008, 4.)

Digitaalinen valmistus eli virtuaalisessa 3D-maailmassa tapahtuva simulointi on koko ajan lisääntyvää ja kehittyvää teknologiaa. Nykyisessä massatuotannossa robotisoidut tuotantosolut voivat olla todella monimutkaisia järjestelmiä, joilta vaaditaan tehokasta ja joustavaa toimintavarmuutta sekä kykyä kommunikoida solussa toimivien muiden toimilaitteiden kanssa. (Siemens PLM 2008, 4.)

Simuloidussa maailmassa tapahtuva ohjelmointi on merkittävässä roolissa robottien ohjauksessa. Simuloinnin avulla solussa oleville roboteille voidaan tehdä ohjelmointimuutoksia ja hienosäätöjä, jotka muuten mahdollisesti vaatisivat rahaa tai aikaa tuhlaavia toimenpiteitä. (Siemens PLM 2008, 4.)

Teknologian avulla voidaan reaali maailman robottisolu tuoda virtuaaliseen maailmaan. Solusta tuotetaan ensin tarkka 3D-mallinnus hyödyntäen valmiita tai itse mallinnettuja komponentteja. Toimilaitteiden kinematiikat, robottien työkalujen määritykset, asemoinnit ja koordinaatit määritellään ohjelmassa. 3D-mallinnuksen jälkeen solusta tuotetaan simulaatio, jonka avulla voidaan testata solun toimintaa virtuaalisesti ja riskittömästi ennen tuotannon aloittamista. (Siemens PLM 2008, 4.)

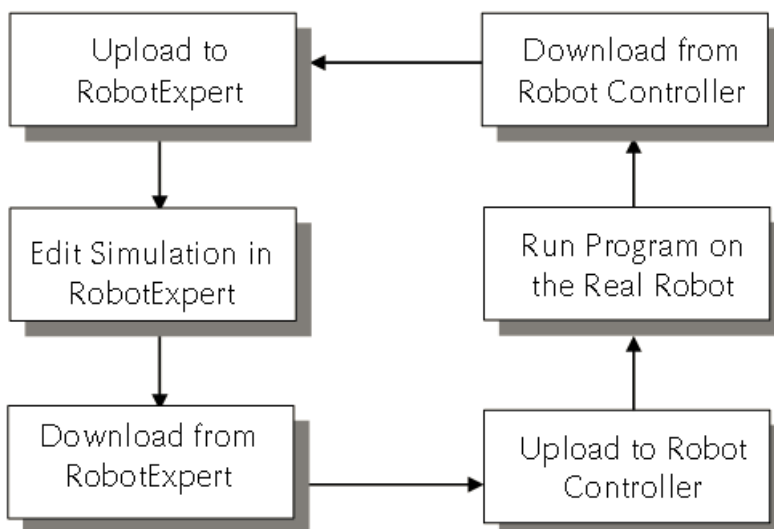
4 TECNOMATIX ROBOTEXPERT -OHJELMISTO

Tässä opinnäytetyössä on käytetty Siemensin RobotExpert-ohjelmistoa. RobotExpert on Tecnomatix®-tuoteperheeseen kuuluva ohjelmisto. Se on kevennetty versio Tecnomatix® Process Simulate -ohjelmistosta. (Siemens PLM 2012, 1)

RobotExpert-ohjelmisto on robottisimulointiin ja etäohjelmointiin suunniteltu työkalu, ja samalla yksi markkinoiden kehittyneimmistä tähän tarkoitukseen soveltuvista ohjelmista. (Siemens PLM 2012, 1)

RobotExpert-ohjelma sisältää useita teollisuus käyttöön tarkoitettuja sovelluksia kuten: kappaleen siirto, kaarihitsaus, hionta, maalaus jne. (Siemens PLM 2012, 3)

Robotin ohjelma voidaan rakentaa offline-ohjelmoinnin avulla RobotExpert-ohjelmassa, tai vaihtoehtoisesti suoraan fyysiseen robottikontrolleriin online-tilassa. Ohjelmoinnin jälkeen robotin koodia voidaan kierrättää ja muokata uudelleen. (Siemens PLM 2012, 16)



Kuvio 2. Robottiohjelman kierrättäminen
(Siemens PLM Robotics [Viitattu 30.7.2015])

4.1 Ominaisuudet

Tecnomatix RobotExpert -ohjelmisto sisältää useita hyödyllisiä ominaisuuksia, jotka on esitelty seuraavassa luettelossa:

- Kinematiikan, työkalujen ja robottien 3D-mallintaminen
- Tuki roboteille monilta valmistajilta
- Robottien törmäystarkastelu
- Robottien offline-ohjelmointi
- Työkiertojen tarkka optimointi käyttäen RRS (Realistic Robot Simulation)-simulaatiota.
- Windows-pohjainen käyttöjärjestelmä
- Mahdollisuus ladata robottien ohjelmat tuotannosta simulaatioon

(Siemens PLM, 1. [Viitattu 27.6.2015].)



Kuvio 3. Tecnomatix RobotExpert
(Siemens PLM Robotics [Viitattu 30.7.2015]).

4.2 RRS- ja RCS-moduulit

RRS (Realistic Robot Simulation), on autovalmistajien ja robottivalmistajien yhteistyössä luoma tarkka esitysmuoto robotin liikkeitä ja asemointia varten, algoritmien ja protokollien avulla. RRS on kansainvälinen standardiesitysmuoto, jota käytetään RobotExpert-ohjelmistossa ja muissa geneerisissä etäohjelmointiohjelmissa. (Siemens PLM, 24. [Viitattu 28.6.2015].)

RRS-moduulin avulla fyysinen robottikontrolleri saadaan kommunikoimaan virtuaalisen kontrollerin kanssa. Standardilla määritellään virtuaalinen kontrolleri niin sanotuksi ”mustaksi laatikoksi”, joka sisältää sille spesifioidut I/O-signaalit. (Siemens PLM, 24. [Viitattu 28.6.2015].)

RCS-moduuli (Robot Controller Simulation) on ohjelmisto- versio fyysisestä robottikontrollerista. RCS-moduuli muuttaa virtuaaliset simuloidut I/O-signaalit fyysiselle robotille soveltuviksi signaaleiksi. RCS-moduulit ovat aina laitevalmistajien, kuten ABB:n tai Fanucin itsensä kehittämiä. RCS-moduulin avulla etäohjelmoitu robotti saadaan liikkumaan suunnitellulla liiketarkkuudella. Ilman RCS-moduulia offline-ohjelmoidun robotin liikkeiden tarkkuus jää noin 80 prosentin tasolle. (Siemens PLM, 25. [Viitattu 28.6.2015].)

5 ROBOTIN OHJELMOINTI

Robotin ohjelmointi koostuu toimintajärjestyksen ja logiikan laatimisesta, niin että robottiin kiinnitettävän työkalun liikkeitä saadaan toteutettua optimaalisesti. Robotin sisäisen ohjausjärjestelmän lisäksi liikkeitä ohjaavat ja tahdistavat solussa toimivien muiden laitteiden lähettämät signaalit. Lopuksi määritellään vielä robotin käyttäytyminen häiriötilanteissa. (Aalto-yliopisto 2010, 33.)

5.1 Teollisuusrobotin tyypilliset käskyt

Robotille tyypillisiä käskyjä ovat esimerkiksi seuraavat:

- Liike voi olla lineaarista liikettä, ympyräliikettä, tai nivelinterpolaatioliikettä. Esimerkiksi ABB:lla komennot ovat MoveL, MoveC ja MoveJ.
- Liikerata koostuu kolmesta eri ulottuvuudesta xyz-koordinaatistossa.
- Tehtäväkoordinaatistoa voidaan siirtää kuudessa vapausasteessa.
- Kappaleen pintaa voidaan etsiä liikekäskyillä, esimerkiksi ulkoisen anturin nousevia ja laskevia reunoja hyödyntäen. ABB:n robotin ohjelmoinnissa pintaa etsitään esimerkiksi searchL-komennolla.
- Liikenopeutta voidaan ohjata ulkoisella anturilla
- Robottia voidaan liikuttaa suhteessa työkalukoordinaatistoon
- C-kielestäkin tuttuja ehto-, toisto- sekä silmukkarakenteita hyödynnetään runsaasti robotin ohjelmoinnissa.
- Robotin ohjelmoinnissa onnistuu myös binääri-, analogia- ja digitaalisignaalien lukeminen, sekä kirjoittaminen.

(Aalto-yliopisto 2010, 37.)

5.1.1 Käskyt ulkoisiin järjestelmiin

Robotti on myös yhteydessä ulkoisiin järjestelmiin:

- Kappaleiden erottelu ja poimiminen hyödyntämällä konenäköjärjestelmää.
- Robotin liikkeiden ohjaaminen pulssianturilla, esimerkiksi kuljettimen yhteydessä.
- Kaari- ja piste-hitsauslaitteiden ohjaaminen.

(Aalto-yliopisto 2010, 38.)

5.1.2 Robotin valinta

Robottia valitessa huomioidaan ensin robotin pääasiallinen käyttötarkoitus sekä käsiteltävän työkappaleen ominaisuudet, tuotantokapasiteetti, robotin hinta sekä soveltuvuus tuotantoon. Seuraavassa esitellään ominaisuuksia, jotka on myös huomioitava robottia valittaessa:

- robotin käsivarren ulottuvuus useassa eri asennoissa.
- työkalun ja siirrettävän massan maksimikuorma robotin ollessa ääriasennoissa.
- ohjelmointikieli.
- robotin kontrollerin liitynnät solussa toimiviin sekä ulkoisiin toimilaitteisiin.
- solussa tapahtuvat mahdolliset ohjelmointimuutokset on hyvä ottaa myös huomioon.

(Kuivanen 1999. 93,115.)

5.1.3 Liitännät logiikkaan ja solun muihin toimilaitteisiin

Robotin liikettä ohjaa niiden oma ohjausjärjestelmä eli robottikontrolleri. Toimies-
saan osana suurempaa tuotantojärjestelmää tarvitaan myös kommunikointia raja-
pintojen yli. Tärkeimpänä ovat rajapinnat ohjelmoidulle logiikalle (PLC). Robotin on
myös kyettävä keskustelemaan solussa toimivien muiden toimilaitteiden kanssa,
esimerkiksi etikettilaitteen, kuljettimen tai toisen robotin kanssa. (Aalto-yliopisto
2010, 6.)

Tiedonsiirto PLC:lle ja toimilaitteille mahdollistaa useiden tuotannolle elintärkeiden,
ja niiden toiminnan kannalta oleellisten tapahtumien seurannan ja hallinnan:

- tuotantoprosessin ohjaus, päättely ja toiminta.
- tuotantoprosessin valvominen käyttöliittymän avulla.
- tuotantohistorian (laatu, jäljitettävyyys, kapasiteetti) tallentaminen tietokan-
taan.
- häiriötilanteiden tunnistaminen ja tallentaminen.
- tuotantolaitteiden käyttöönotto on oltava vaivatonta.
- jotkut toimilaitteet vaativat asetuksien säätöä parametrien avulla.

(Aalto-yliopisto 2010, 8.)

Tiedonsiirrossa käytettäviä tekniikoita ovat:

- sarjayhteydet
- kenttäväylät
- ethernet
- wireless
- OPC

(Aalto-yliopisto 2010, 10.)

5.2 Ohjelmointitavat

Robotin ohjelmointi tapahtuu joko online-tilassa, jossa ohjelmointityylejä on erilaisia, tai offline-tilassa, jossa ohjelmointi tapahtuu etäohjelmointina, hyödyntäen 3D-malleihin perustuvaa simulointia (Tuunanen 2014, 34.)

5.3 Online-ohjelmointi

Online-ohjelmoinnissa robotti otetaan pois tuotannosta ohjelmoinnin ajaksi. Tuotanto pysähtyy määräämättömäksi ajaksi. Tästä luonnollisesti aiheutuu tuotannon heikentymistä. Yleisin online-tilassa tapahtuva ohjelmointi on opettamalla ohjelmointi. (Tuunanen 2014, 34.)

5.3.1 Opettamalla-ohjelmointi

Opettamalla ohjelmoinnissa robotti, tai siihen kiinnitetty työkalu, ajetaan käsiohjaimella johonkin paikoituspisteeseen. Pisteeseen koordinaatit (x, y, z), sekä robotin akselien astekulmat voidaan tallentaa tässä halutussa pisteessä ja nimetä piste halutulla tavalla. Liikekäsky voi olla esimerkiksi MoveL (lineaari-liikekäsky), MoveJ (Interpolaatio) tai MoveC (ympyrä-liikekäsky). Lisäksi pisteeseen määritetään liikenopeudet, millä nopeudella pisteeseen saavutaan. Tallennuksen jälkeen robotti ajetaan taas haluttuun asentoon ja tallennetaan paikka muistiin. (Kaarela 2007, 14.)

5.4 Offline-ohjelmointi

Etäohjelmoinnilla eli offline-ohjelmoinnilla (OLP) tarkoitetaan tuotannon ulkopuolella tapahtuvaa robottien ohjelmointia. Ohjelmointi suoritetaan erillisellä PC:llä olevalla tietokoneohjelmistolla. Tuotantosolu pystytään kokonaisuudessaan 3D-mallintamaan ja rakentamaan siitä reaali maailman mukainen tarkka simulointimalli. (Finnrobotics, [Viitattu 15.4.2015].)

5.4.1 Offline-ohjelmoinnin hyödyt

Merkittävin hyöty offline-ohjelmoinnissa on, että tuotantoa ja siinä toimivaa robottia ei tarvitse pysäyttää ohjelmoinnin ajaksi. Etäohjelmointi-ohjelmiston avulla uusi ja suunnitteilla oleva tuotantosolu voidaan simuloida, ja simuloinnin pohjalta voidaan robottien ohjelmointi tehdä etukäteen. (Finnrobotics, [Viitattu 15.4.2015].)

Simulaatio-ohjelmiston avulla robottien liikeradat ja ulottuvuudet voidaan havainnollistaa. Useimmilla ohjelmistoilla on myös mahdollisuus robotin ja solussa olevien laitteiden törmäystarkasteluun, eli säästyään myös mahdollisilta laiterikoilta. Yksi merkittävä ominaisuus on työkiertojen ja tahtiaikojen määrittäminen ja optimoiminen. (Finnrobotics, [Viitattu 15.4.2015].)

Etäohjelmointi tulee tarpeelliseksi tilanteissa, joissa tuotanto on tilausohjautuvaa, tuotantosarjat ovat pieniä ja tuotteiden elinkaaret ovat lyhyitä (Aalto-yliopisto 2010, 44).

Tuotantosoluissa, joissa robotilta vaaditaan suurta määrää paikoituspisteitä (hitsaus, leikkaus, jäysteytys jne.), tulee robotin etäohjelmointi varteen otettavaksi vaihtoehdoksi (Aalto-yliopisto 2010, 44).

Usein teollisuudessa tulee myös tilanteita, joissa robotin ohjelmointi opettamalla on mahdotonta tilan puutteen tai turvallisuusriskien takia. Tällaisia tuotantolaitoksia ovat esimerkiksi ydinvoimalaitokset, ampumatarvike- tai valimoteollisuus (Aalto-yliopisto 2010, 38.)

Merkittävä ominaisuus offline-ohjelmointijärjestelmässä on robotin työkalun asennon, nopeuden ja etäisyyden hallinta suhteessa työkappaleeseen. Tämä huomataan hyvin esimerkiksi kiillotus-, hitsaus- ja hiontaprosesseissa laadun paranemisenä. Suunnittelijoiden välillä on eroja ohjelmointitaidoissa. Etäohjelmoinnissa työkalun hallinta ja parametrien syöttö tapahtuu ohjelmistoteknisesti mikä poistaa ohjelmointitaidoista johtuvaa eroa tuotteiden laaduissa. (Delfoi, [Viitattu 21.4.2015].)

Etäohjelmointi tukee myös Lean-toimintamallia, vähentämällä hukkaan mennyttä aikaa. Uusien tuotantosolujen käyttöönotto saattaa lyhentyä viikoista yhteen päivään. Tällä mahdollistetaan piensarjatuotantojen robotisointi. (Delfoi, [Viitattu 21.4.2015].)

5.5 Simuloinnin ja offline-ohjelmoinnin evoluutio

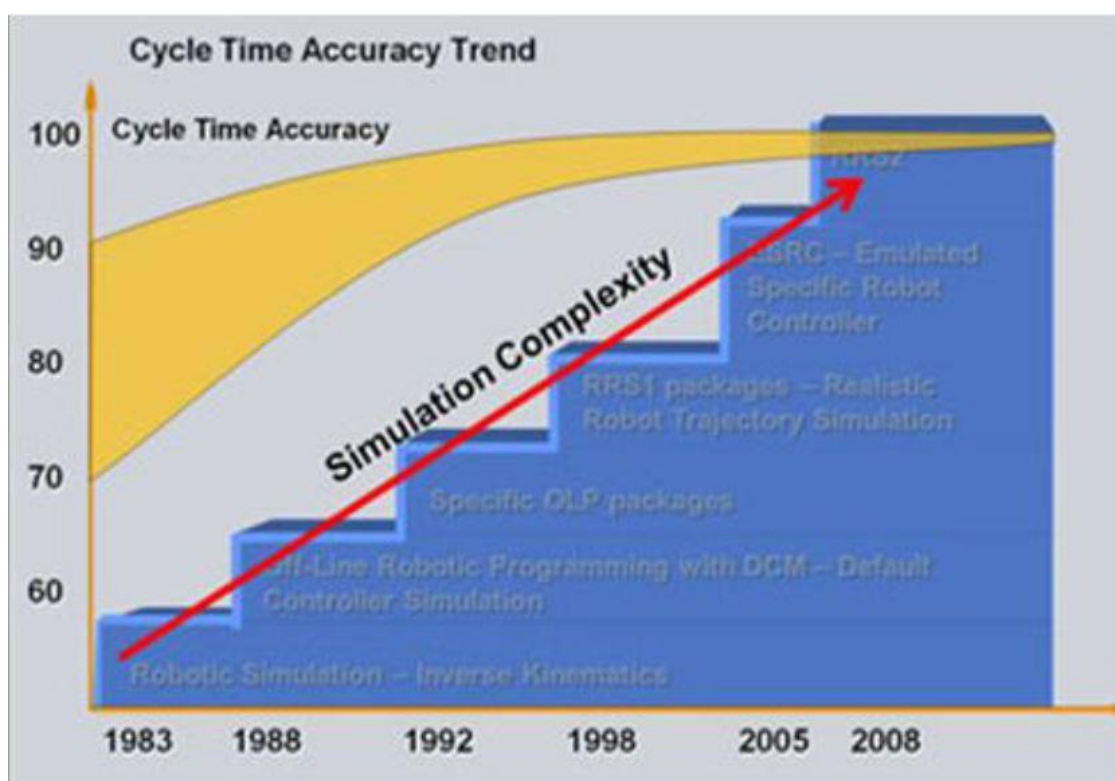
Tuotantokoneiden, robottien ja toimilaitteiden testaus ja suunnittelu hyödyntäen virtuaalista simulointia on lisääntynyt huomattavasti 1980-luvulta 2000-luvulle (kuvio 5). Samaan tahtiin on lisääntynyt myös 3D-mallinnus sekä simulointiohjelmien tarkkuus. Tuotantojärjestelmän muuttaminen virtuaalisesta fyysiseksi vaatii suurta tarkkuutta ja virheettömyyttä tuotantolaitteiden 3D-mallinnukselta ja simuloinnilta. (Siemens PLM 2008, 6-7.)

Jotta saadaan kaikki hyöty irti simulointi- ja OLP-ohjelmistoista, robottisolun tai minkä tahansa tuotantosolun suhteen, on mallinnetun ja simuloitun tuotoksen vastattava vähintään 95-prosenttisesti fyysistä solua. Ohjelmalta odotetaan erilaisia liikemalleja, logiikoiden ohjausta, virheetöntä robottien offline-ohjelmointia sekä tarkkoja työkiertojen ja häiriöiden käsittelyä. (Siemens PLM 2008, 6-7.)

Tuotantoympäristö ja valmistusprosessit muuttuvat koko ajan monimutkaisemmiksi. Tämä on seurausta automaatio-sovellusten lisääntymisestä. Merkittäviä vaikuttajia ovat myös joustavat, erikokoisten ja muotoisten kappaleiden kokoonpanolinjastot, sekä heterogeeniset tuotantolinjat, lukuisine robottivalmistajineen, logiikoineen ja toimilaitteineen. (Siemens PLM 2008, 7-8.)

Samaan tahtiin kuin tuotantolinjat monimutkaistuvat ja kasvavat, myös materiaali-virta kasvaa. Tämä on johtanut tilanteeseen, jossa simulointi ja offline-ohjelmointi on lähestulkoon välttämätöntä, jotta yritykset pystyisivät hallitsemaan monimuotoisia tuotantosoluja, sekä pitämään materiaalivirran tehokkaana, laadusta tinkimättä. (Siemens PLM 2008, 8.)

Offline-ohjelmoinnin ja simuloinnin on todettu, ei ainoastaan vähentävän automaatioprosessien suunnitteluvirheitä, vaan myös virtuaalisesti vahvistavan työsolun koko suunnitteluprosessia. Offline-ohjelmoinnilla voidaan saavuttaa merkittäviä säästöjä automaatiojärjestelmien fyysisessä testauksessa ja käyttöönotossa. (Siemens PLM 2008, 8.)



Kuvio 4. Simuloinnin monimutkaistuminen
(Siemens PLM Software 2008, 7)

6 OFFLINE-OHJELMOINTIPROSESSIN VAIHEET

6.1 Lähtötietojen hankinta

Lähdettäessä etäohjelmoimaan robotisoitua tuotantosolua, oli ensin selvítettävä kaikki solun toiminnan kannalta tärkeät lähtötiedot. Aluksi tutustuttiin solussa toimivan robotin ja toimilaitteiden tehtäviin. Simulointia ja ohjelmointia alettiin suorittaa, kun tiedossa oli karkeasti robotilta vaadittavat ominaisuudet, sekä robotin ja solun sisällä toimivien muiden toimilaitteiden välinen keskustelu.

6.1.1 Simuloitavan ja etäohjelmoitavan robottisolun toimintakuvaus

Etikettirobotti sai logiikalta tiedon kuljettimelle saapuvasta rullasta. Logiikalta tuleva tieto sisälsi fyysiset mitat rullasta, sekä tiedon, tuleeko rullaan yksi vai kaksi etikettiä. Solussa oli robotin lisäksi kaksi printteriä, joista toinen oli niin sanotusti reservissä, etikettien loppuessa ensimmäiseltä printteriltä. Logiikka lähetti tiedon robotille yhden tai kahden etiketin hakemiseen.

Robottiin oli kiinnitettynä imukupitarttuja työkaluksi etikettien hakuun ja liimaamiseen. Työkalu sisälsi kaksi tarttujaa, joiden sylintereitä ohjattiin paineilman avulla eteen sekä taakse. Imukupin toiminnot olivat imu, jolla etiketit saatiin kiinnitettyä työkaluun, sekä puhallus, jolla etiketti kiinnitettiin rullan pintaan. Työkalussa oli lisäksi induktiiviset anturit molemmissa tarttujissa, niiden avulla robotti sai tiedon, että etiketti oli kiinnittyneenä työkaluun.

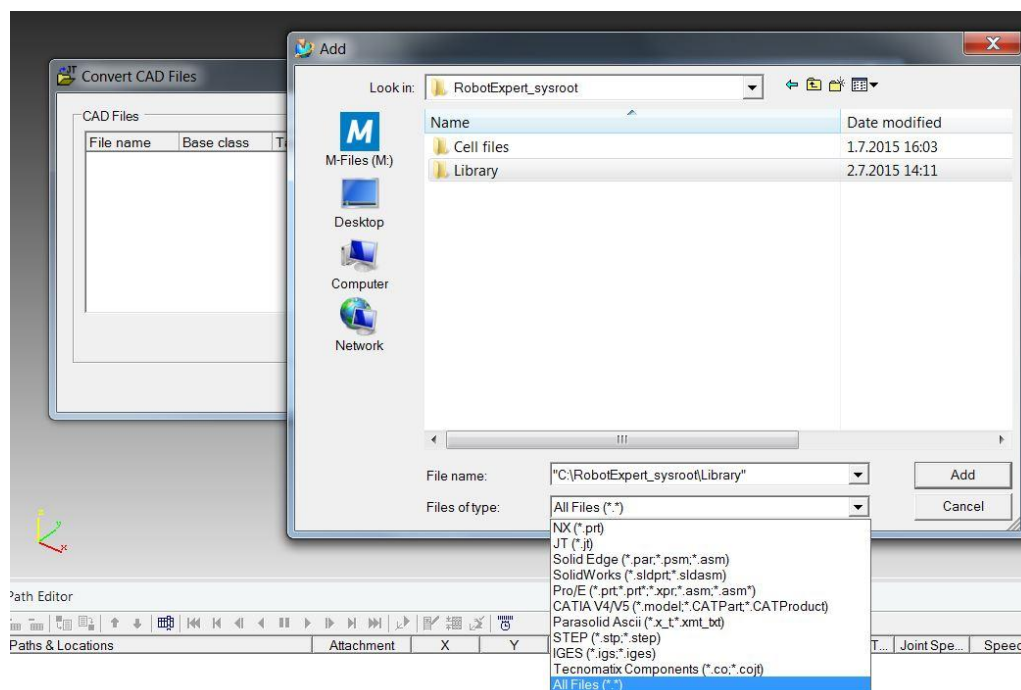
Logiikalta tulleen tiedon perusteella robotti kiinnitti työkaluunsa tarvitun määrän etikettejä ja siirtyi odottamaan rullaa kuljettimen läheisyyteen. Rullan saapuessa robotille robotti sai logiikalta luvan kiinnittää etiketit rullaan.

Ensimmäinen lappu kiinnitettiin rullan vaippaan. Rullan pinnan etsintä suoritettiin SearchL-liikekomennolla. Robotin saadessa käskyn kiinnittää myös toinen etiketti suoritettiin kiinnitys rullan päätyyn.

Suoritettuaan etikettien kiinnittämisen robotti siirtyi takaisin kotiasemaan odottamaan uutta työkiertoa.

6.2 Solun 3D-mallinnus

Solun toimilaitteiden 3D-mallintaminen on hyvä suorittaa mahdollisimman tarkasti. Mallintaminen voidaan suorittaa ulkoisella ohjelmistolla, kuten Autodesk Inventorilla tai Solid Edge -ohjelmistolla. RobotExpert-ohjelmisto hyväksyy useimmat 3D-malleissa käytetyt tiedosto muodot. Tiedostomuodot käyvät ilmi kuvioista 6.



Kuvio 5. 3D-mallin konvertoiminen ja RobotExpertin tukemat formaatit

Tarkkuudeltaan vaatimattomimpien kappaleiden mallinnus voidaan useimmiten suorittaa myös etäohjelmointiohjelmiston sisällä. Robottimalleja ei tarvitse simuloitiin erikseen mallintaa. Joidenkin robottivalmistajien robottimallien CAD-tiedostot löytyvät valmistajien omilta Internet-sivuilta ja ovat ilmaiseksi ladattavissa omalle PC:lle.

The screenshot shows the ABB website interface for 'ABB Suomessa'. The main navigation includes 'Etusivu', 'ABB lyhyesti', 'Tuotteet & palvelut', 'Media', 'ABB-uralle', 'Investor center', 'ABB Group', and 'Puheeni'. The breadcrumb trail is 'Tuotteet ja järjestelmät > Robotit > Robotit > IRB 1600'. The 'CAD Models' section is active, showing a table of download links for the M2004 robot.

| Ver | DXF/DWG | SAT | SW | STEP | PARASOLID | VDA | IGES | STL | VRML | ROBCAD | Robotstudio | IGRIP |
|-------|-----------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----|-----------------------|-----------------------------|-----------------------------|-----------------------------|---------------------|---------------------|
| X | | | | | | | | | | | | |
| X-120 | compl | compl joint | compl joint | compl joint | compl joint | | joint | compl joint | compl joint | compl joint | sim | sim |
| X-140 | | joint | | joint | joint | | joint | | | | | |
| X-145 | compl | compl joint | compl joint | compl joint | compl joint | | joint | compl joint | compl joint | compl joint | sim | sim |
| - | compl | | | | | | | | | | | |

Kuvio 6. Valmiiden 3D-mallien lataaminen robottivalmistajan sivuilta (ABB 2015).

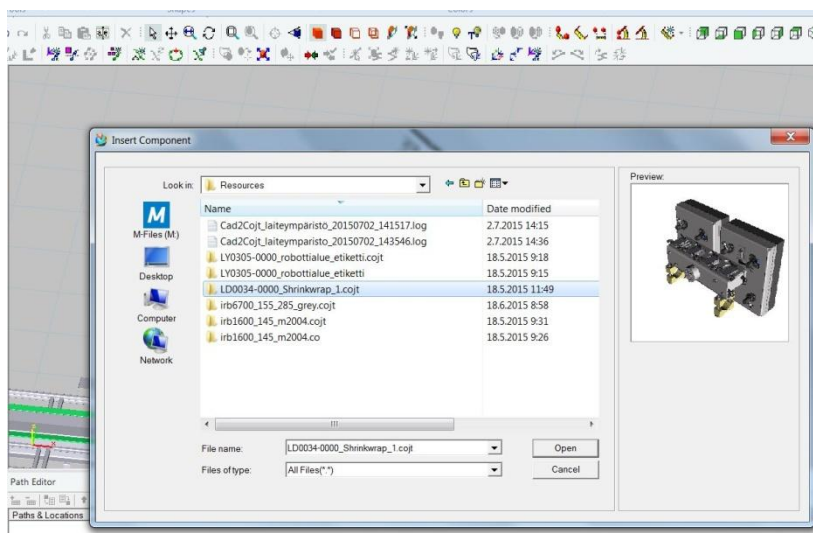
Tässä tutkimustyössä robottivalmistajana toimi ABB. Robotiksi valittiin IRB1600 -mallin robotti. Robotti oli käsivarsirobotti, jonka työkalupisteen ulottuvuus on 1,45 metriä ja maksimikuormitus työkalun kanssa 10 kilogrammaa. Robotin koko olisi saanut olla isompi, mutta johtuen työsolun ahtaudesta, ei isomman kokoluokan robottia olisi voitu järkevästi sijoittaa soluun.



Kuvio 7. ABB IRB1600 -robotti (ABB 2015).

6.3 Solun toimilaitteiden paikoitus simulointimalliin layout-mittojen pohjalta

Mallinnuksen jälkeen solun toimilaitteet konvertoidaan etäohjelmointiohjelman tukemaan muotoon, jonka jälkeen 3D-tiedosto voidaan tuoda simulaatioon (kuvio 9.)



Kuvio 8. 3D-mallin tuominen simulaatioon

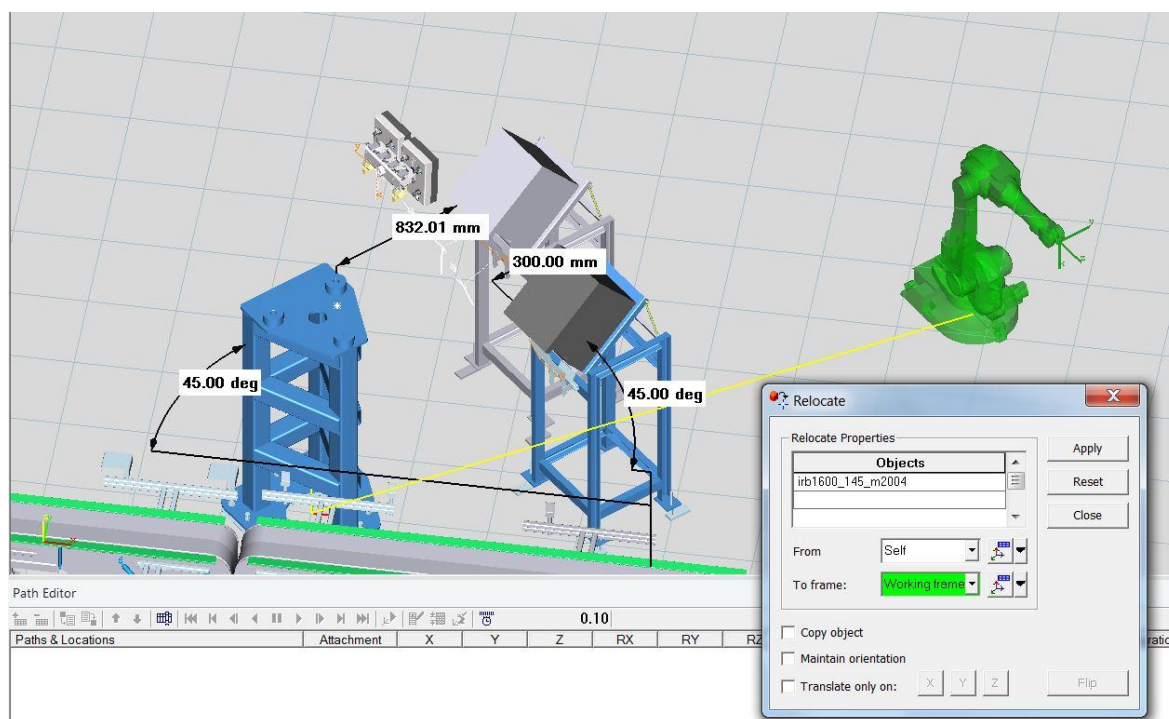
Solun toimilaitteet ja työtasot paikoitetaan mahdollisimman tarkasti suunnittelijan tekemän layout-kuvan perusteella (Kuvio 10.).



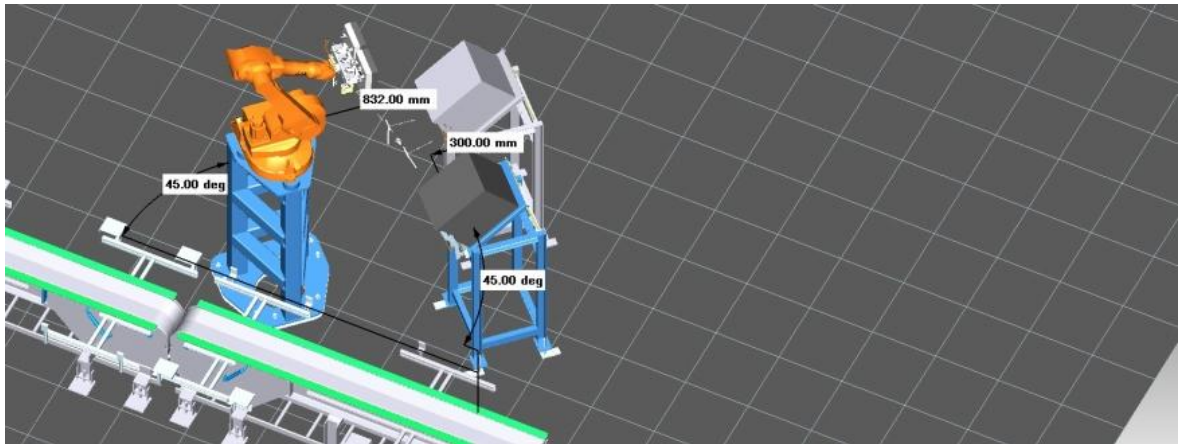
Kuvio 9. Etikettirobotin solu

Robotti sijoitetaan koordinaatistossa origon keskelle eli sille suunniteltuun työasemaan. Ohjelmassa on käytetty relocate-työkalua, jolla robotti voidaan siirtää tarkasti ohjelmoijan haluaan paikoituspisteeseen (Kuvio 11.). From frame -ruutuun määritellään, mitä kohtaa siirrettävästä objektista käytetään. To frame -ruutuun määritellään kohdepiste, johon objektista valittu piste kohdistetaan. Lisäksi työkalussa on valintaruudut:

- Copy object -valinta, jolla voidaan kopioida robotti tai muu objekti toiseen paikoituspisteeseen.
- Maintain orientation -valinnalla robotin orientaatio voidaan säilyttää samana lähtöpisteeseen suhteutettuna.
- Translate only -valinnalla siirretään objektia x-, y- ja z-koordinaattien mukaan.



Kuvio 10. Toimilaitteiden sijoittaminen layout-kuvan perusteella



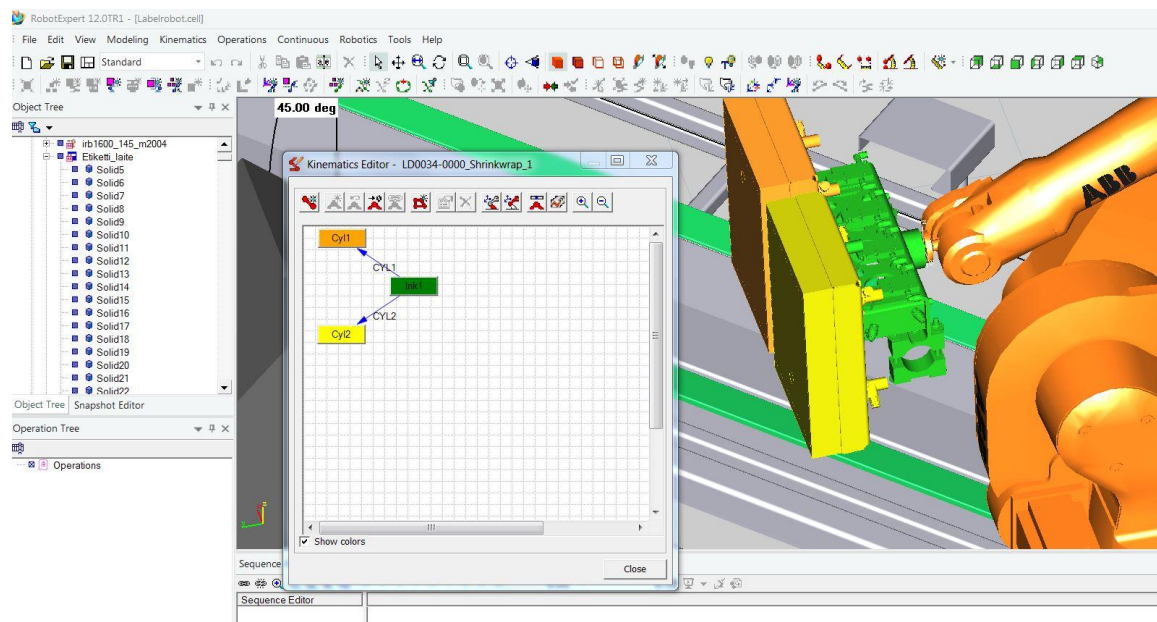
Kuvio 11. Valmis robottisolu

6.4 Toimilaitteiden kinematiikan rakentaminen

3D-malleille voidaan rakentaa kinematiikka Kinematics Editor -työkalun avulla. Kinematiikan rakentaminen tulee kyseeseen, kun simulaatiosta halutaan näyttävämpi. Etäohjelmoinnin kannalta kinematiikan rakentaminen ei niinkään ole aina välttämätöntä. Kinematiikalla tarkoitetaan simuloinnissa jonkin yksittäisen kappaleen tai työkalun mekaanisten toimintojen simulointia.

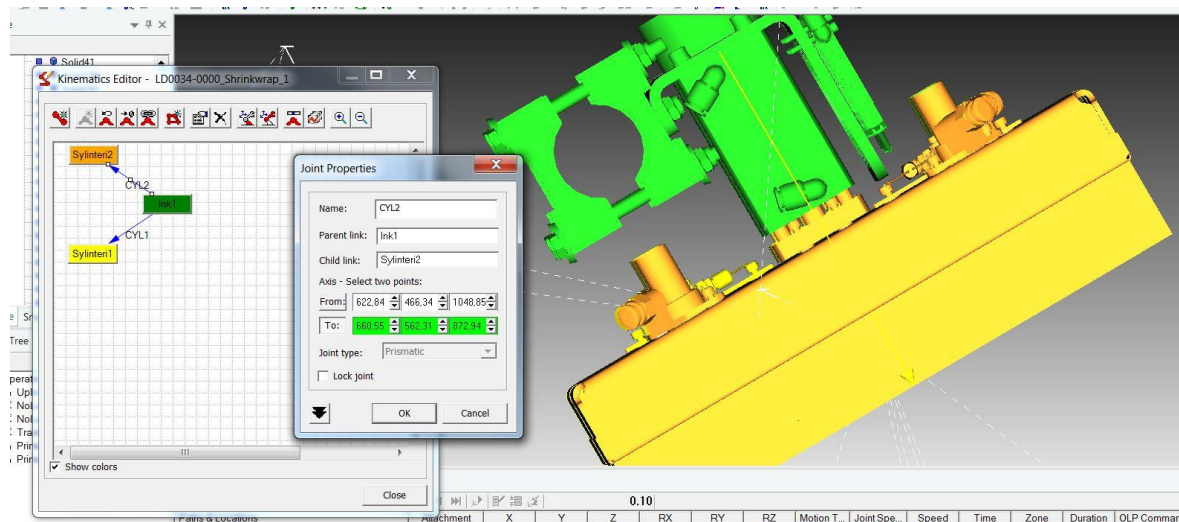
Kinemaattinen malli koostuu kinemaattisista ketjusta ja nivelistä. Kinemaattinen nivel on joko suora akseli tai pyörähdysakseli.

Seuraavissa kuvioissa (kuviot 13. ja 14.) esimerkkinä on rakennettu imukupitartujalle kinematiikka. Aluksi työkalun runko ja liikkuvat osat, tässä tapauksessa sylinterit, määritellään linkeiksi, jotka erottuvat toisistaan väreillä. Kahden linkin välille vedetään nuoli, joka synnyttää näiden linkkien väliin akselin. Kuviossa 14 nähdään, että vihreästä linkistä on vedetty nuoli sekä oranssille että keltaiselle linkille. Oranssi ja keltainen linkki ovat sylintereitä, jotka liikkuvat eteen- ja taaksepäin suhteessa vihreään linkkiin.



Kuvio 12. Kinematiikan rakentaminen

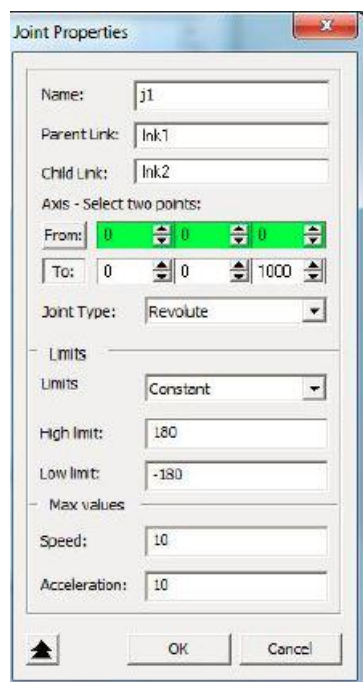
Seuraavana määriteltiin sylinterien kinematiikka eli liikerata. Tuloksena syntyi suora akseli, jota pitkin sylinterit liikkuvat suhteessa tarttujan runkoon (kuvio 14.). Kuviossa haaleasti näkyvä keltainen nuoli osoittaa sylinterin liikeradan.



Kuvio 13. Kinematiikan viimeistely

Työkalussa akselille voidaan määrittää myös seuraavia parametreja, joilla kinematiikasta saadaan vielä realistisempi.

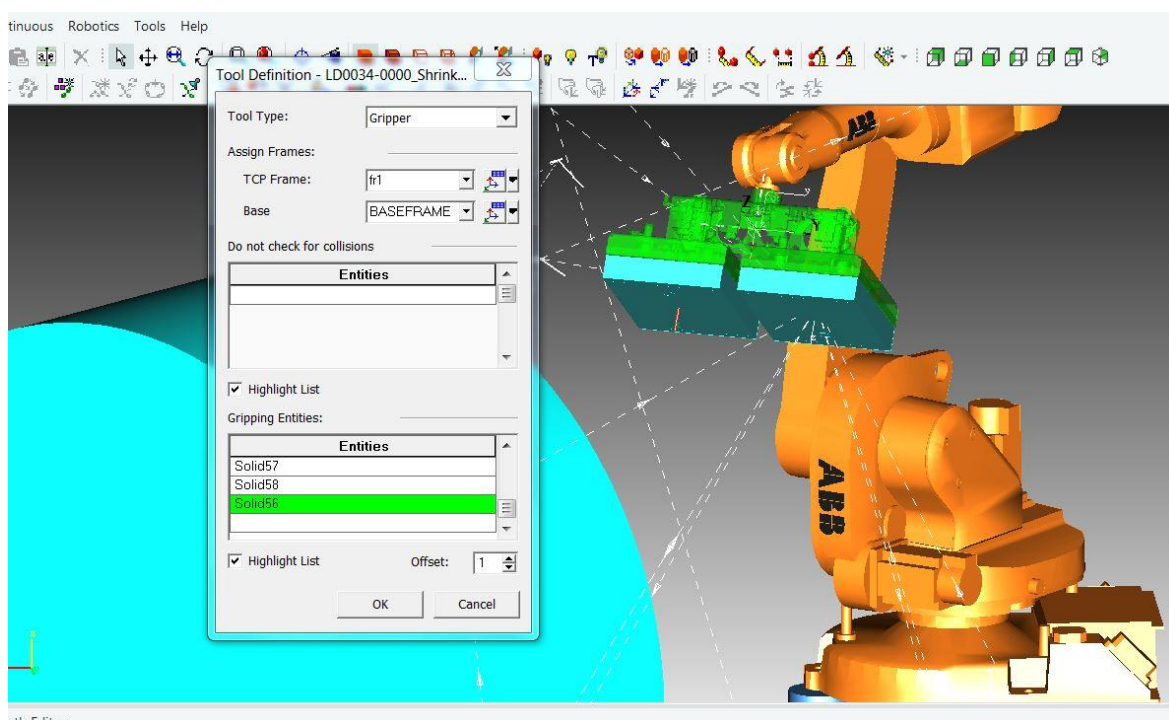
- Akselille voidaan määrittää, miten suuren liikkeen sylinteri voi tehdä miinus- ja plus-suunnassa.
- Akselille voidaan määrittää myös nopeus ja kiihtyvyys.



Kuvio 14. Työkalun akselien määrittäminen (Siemens PLM Robotics 2014)

6.5 Robotin työkalujen määrittäminen ja kiinnittäminen

Robottivalmistajien kuten ABB:n omissa etäohjelmointiohjelmissa on yleensä valmiita työkaluja, jotka haetaan ohjelman omasta sisäisestä kirjastosta. RobotExpert-ohjelmisto on geneerinen etäohjelmointialusta usealle eri robottivalmistajalle. Tästä syystä työkalut joudutaan yleensä lataamaan ohjelman sisään sille tarkoitetussa formaatissa, ja määrittämään ne erikseen työkaluiksi. Työkaluksi määrittämisen jälkeen pystytään työkalut kiinnittämään robotin ranteeseen.



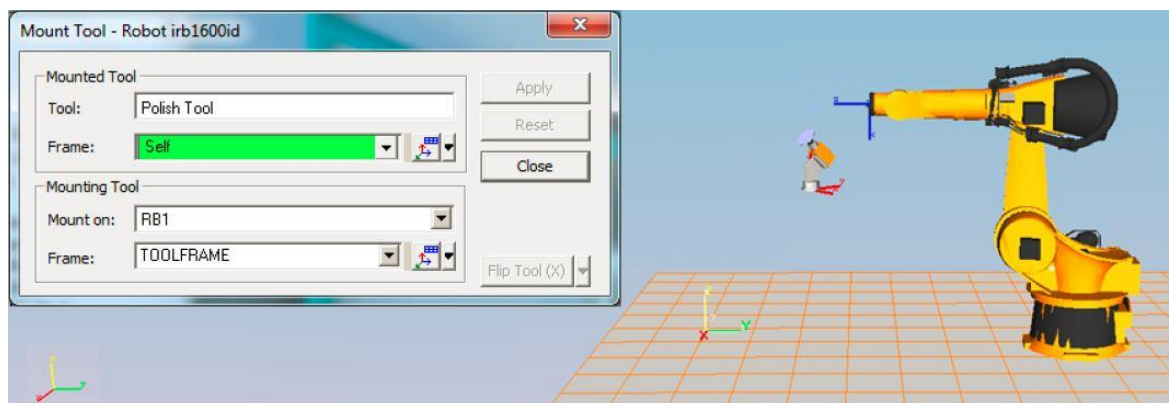
Kuvio 15. Työkalun määrittäminen

Työkalun tyyppi valitaan esimerkiksi tarttujaksi, hitsauslaitteeksi, maalauslaitteeksi jne. Tämän jälkeen työkalulle määritellään TCP-frame eli piste, josta työkalu kiinnitetään robotin käsivarteeseen, sekä työkalun baseframe eli geometrinen keskipiste.

6.5.1 Työkalun kiinnittäminen robottiin

Kinematiikan ja työkaluksi määrittämisen jälkeen työkalu kiinnitettiin robotin ranteeseen. Työkalun koordinaatistosta (TCP) saatiin yhtenevä robotin työkalukoordinaatiston kanssa. Tästä syystä robotin liikkuessa työkalu liikkuu robotin mukana seuraten robotin työkalukoordinaatistoa.

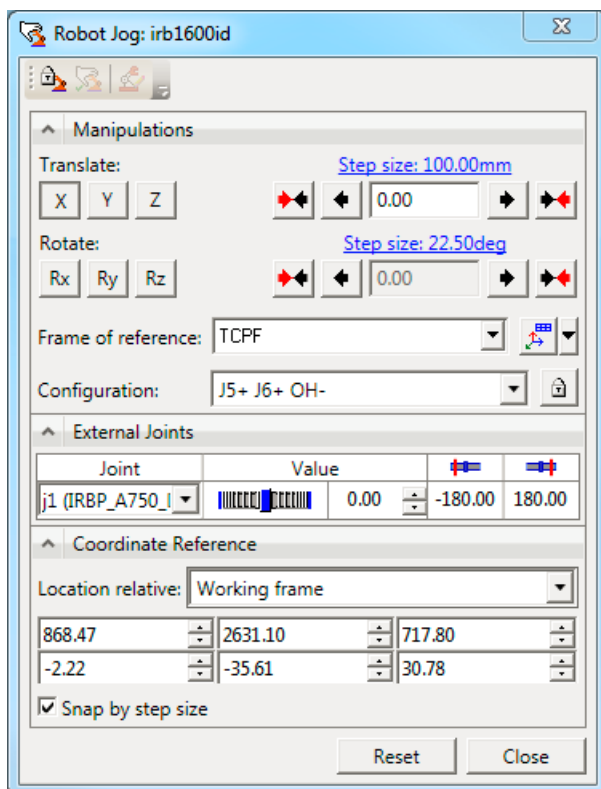
Työkalu kiinnitettiin robotin ranteeseen Mount Tool -työkalulla. Ensin valittiin työkalu, joka haluttiin kiinnittää. Toiseksi valittiin työkalun koordinaatisto eli kohta josta kiinnitys tapahtui, ja miten päin työkalu kiinnittyi robotin ranteessa. Viimeisenä valittiin robotti sekä koordinaatisto, mihin työkalu kiinnitettiin (Kuvio17).



Kuvio 16. Työkalun kiinnittäminen robottiin (Siemens PLM Software 2014).

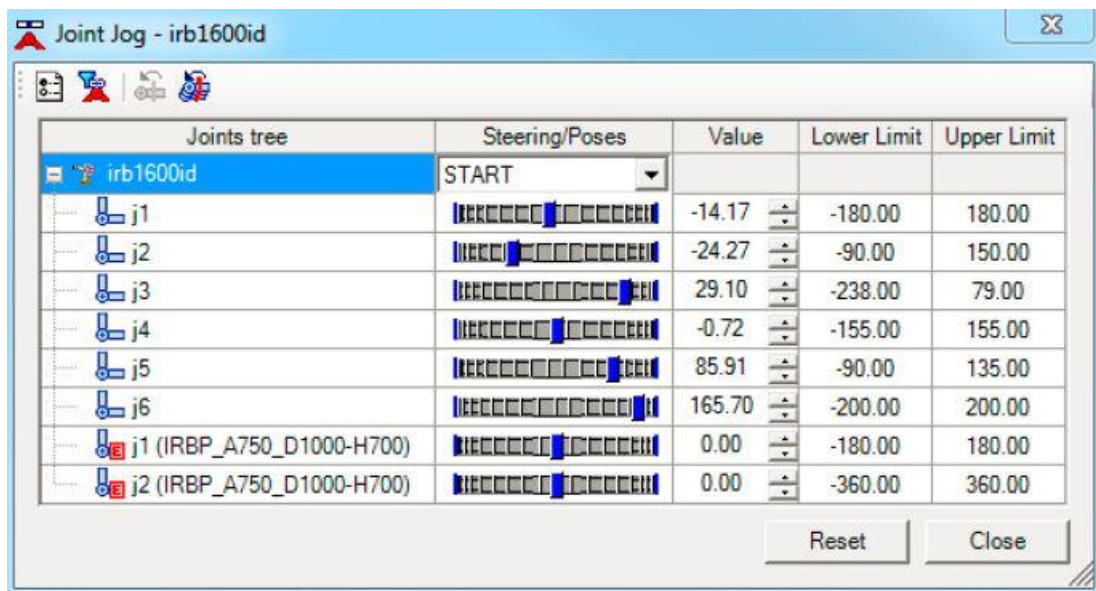
6.6 Robotin ja toimilaitteiden akselien liikuttaminen

Robotin akselien liikuttaminen tapahtui Robot Jog -toiminnon avulla. Työkalulla robottia voitiin liikuttaa kaikkiin sille kinemaattisesti mahdollisiin asentoihin lineaarisesti tai vapaita akseleita hyödyntäen.



Kuvio 17. Robotin liikuttaminen Robot Jog -toiminnolla

Joint Jog -toiminnolla robottia tai muuta kinematiikan omaavaa laitetta pystytään ohjaamaan yksi akseli kerrallaan. Työkalulla voitiin liikuttaa sekä robotin niveliä että työkalun niveliä.

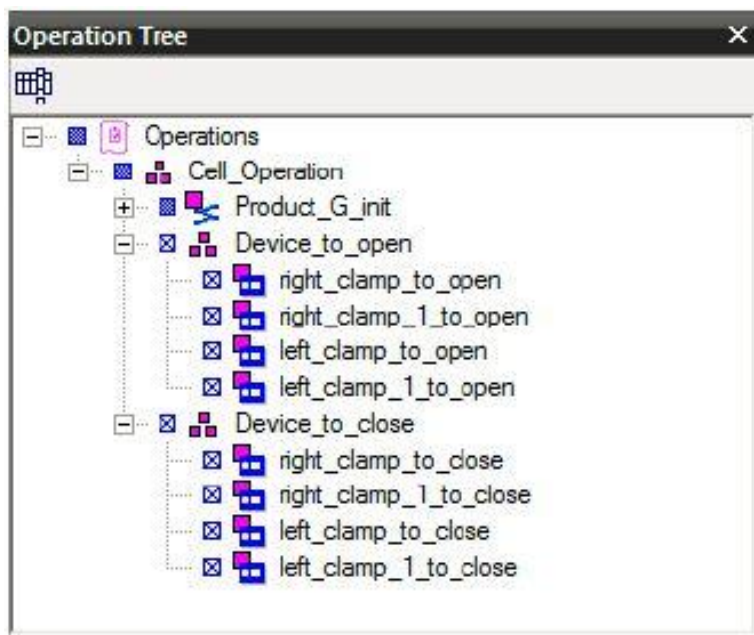


Kuvio 18. Robotin liikuttaminen Joint Jog -toiminnolla

6.7 Liikekäskyjen rakentaminen

Robotin ohjelma rakennettiin runkoon, jota kutsutaan ohjelmassa operaatiopuuksi (Operation Tree). Operaatiopuu alkaa ohjelmamoduulilla, jonka voi nimetä halutulla tavalla. Moduuli koostuu useista yhdistetyistä operaatioista eli rutiineista, jotka on ketjutettu toisiinsa (Kuvio 20). Operaatiot koostuvat simuloituista robotti- ja työkaluoperaatioista:

- Compound-operaatio on hierarkkinen operaatio, joka koostuu muista operaatioista ja voi myös sisältää muita compound-operaatioita (Siemens PLM Robotics, 19. [Viitattu 30.7.2015]).
- Object flow -operaatio on simuloitava operaatio, jolla pystytään luomaan työstettävän kappaleen liikkuminen kokoonpanolinjastolla (Siemens PLM Robotics, 19. [Viitattu 30.7.2015]).
- Device-operaatio on laiteoperaatio, jossa mitä tahansa kinematiikan omaavaa laitetta, kuten robottia tai työkalua voidaan liikuttaa asennosta toiseen (Siemens PLM Robotics, 19. [Viitattu 30.7.2015]).
- Robotic-operaatio on simuloitava operaatio, joka koostuu robotin liikesevensseistä suhteessa robotin työkalukeskipisteeseen. Tätä operaatiota käytetään materiaalin käsittelyyn ja hitsaukseen. (Siemens PLM Robotics, 19. [Viitattu 30.7.2015]).
- Pick and place -operaatio on määritelty operaatio robotille, jolla on tarttuja työkalu. Operaatio on kappaleen siirtämistä varten paikasta toiseen. (Siemens PLM Robotics, 19. [Viitattu 30.7.2015]).



Kuvio 19. Operaatiopuu
(Siemens PLM Software 2014).

Robotti-operaatiot voivat sisältävät seuraavaa informaatiota:

- liiketyyppi
- nopeus
- koordinaatistoalue
- työkalu
- prosessiin kuuluvia parametreja.
- OLP-komentoja eli signaaleja, joilla viestitään muille toimilaitteille ja ohjataan työkalun toimintoja.
- robotin nivelten konfiguraatioita.

Osa peruskäskyistä ja informaatiosta voidaan ottaa suoraan geneeriseltä oletuskontrollerilta eli ohjelman sisäiseltä kontrollerilta, joka kääntää ne robotin käyttämälle kielelle. Kuitenkin suuri osa näistä komennoista on robottivalmistajien omille roboteille spesifioituja komentoja ja ne toimivat eri tavoin kuin geneerisen kielen komennot.

Useimmiten on paras kirjoittaa komennot suoraan robotin käyttämälle kielelle kuin käyttää geneeristä kieltä ja kääntää koodi myöhemmin robotin käyttämälle koodikielelle. Käytettäessä RobotExpertin geneeristä kontrolleria ja liikekomentoja jää liiketarkkuus noin 80 prosenttiin. (Siemens PLM Robotics, 19–20. [Viitattu 5.8.2015].)

6.7.1 Sequence Editor

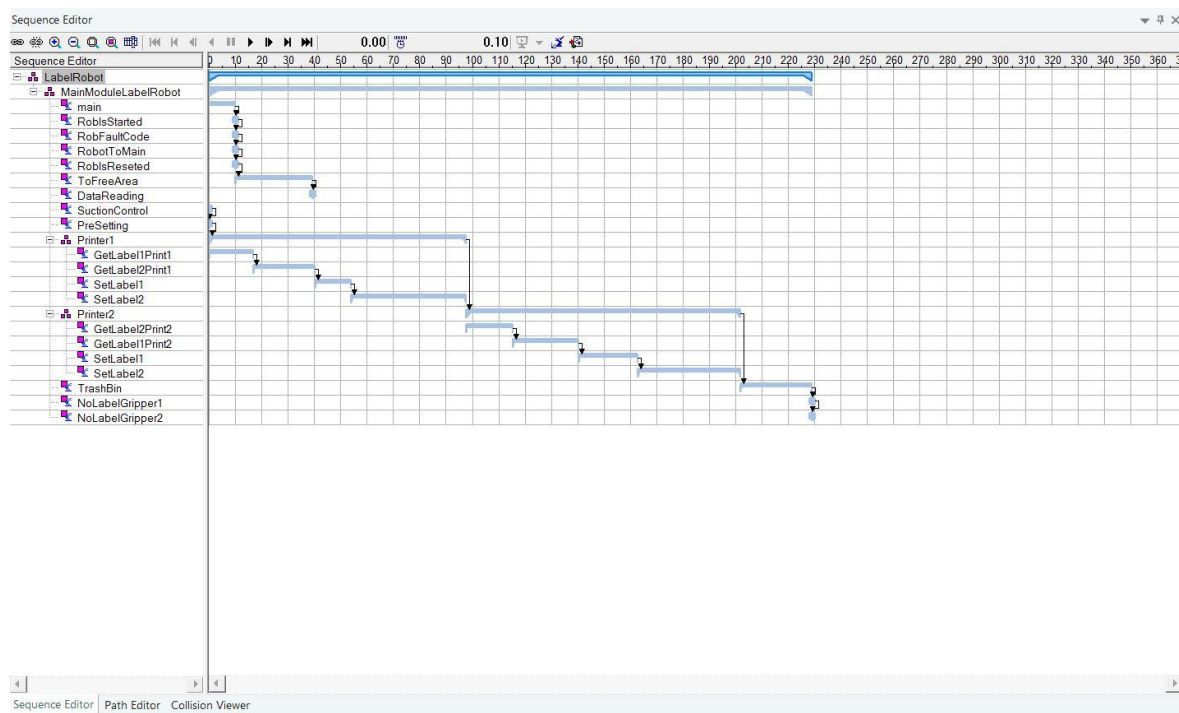
Operaatiopuusta valitaan haluttu operaatio ja siirretään sekvenssieditoriin. Tärkein tehtävä sekvenssieditorilla on eri operaatioiden järjestäminen solun toiminnan mukaiseen järjestykseen. Sekvenssieditorissa operaatiot ketjutetaan yhteen. Robotti suorittaa ketjutetut operaatiot järjestyksessä. Editorissa säädetään operaatioille haluttu alkamisajankohta sekä operaation kestoaika.

Sekvenssieditori on Gantt-kaavio esitysmuodossa. Kaaviossa on näkyvissä kaikki operaatiot ja niiden sisältämät tehtävät, jotka kuuluvat robotille tai jollekin muulle automatisoidulle laitteelle.



Kuvio 20. Sekvenssieditori ennen linkitystä

Gantt-kaavion avulla robotisoidun työsolun visualisointi ja työkierron optimointi on helppoa suorittaa selkeän graafisen käyttöliittymän avulla (Siemens PLM Robotics, 90. [Viitattu 10.8.2015]). Työkierron pituus voitiin säätää viemällä hiiren kursori Gantt-kaaviossa näkyvän sinisen palkin päälle, jonka jälkeen palkkia vedettiin vasemmalle tai oikealle. Työkiertoon kuluva aika näkyi kaavion ylälaudassa. Musta nuoli eri operaatioiden välillä määrittä, minkä toiminnon robotti suoritti missäkin vaiheessa.



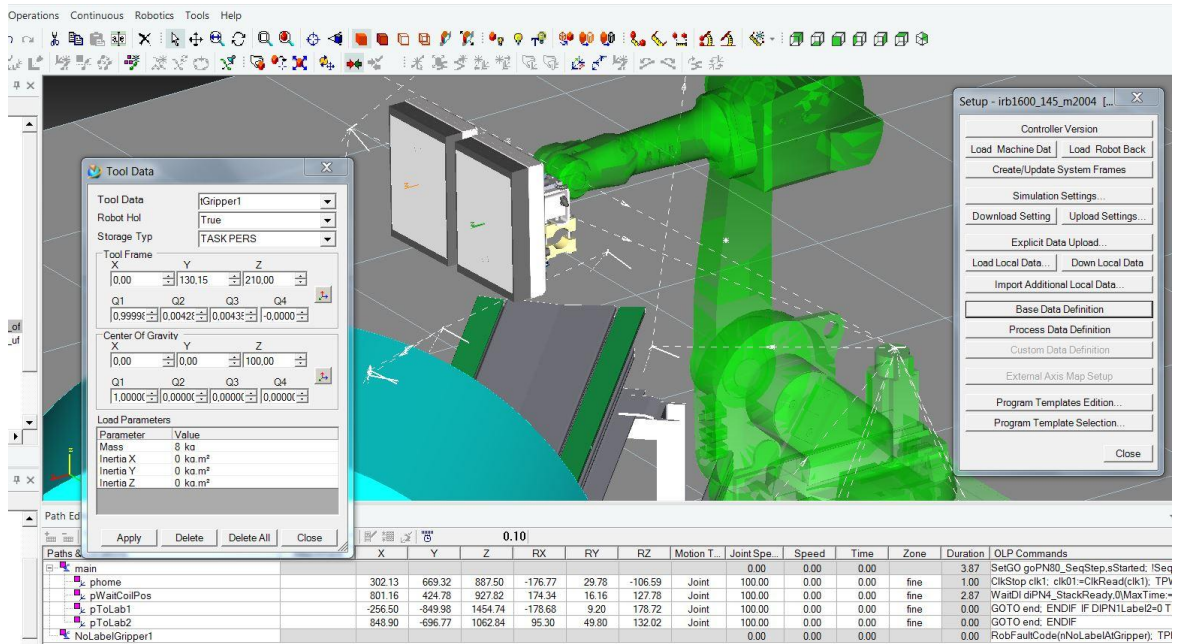
Kuvio 21. Etikettisolun valmis sekvenssisesitys

6.7.2 Path Editor

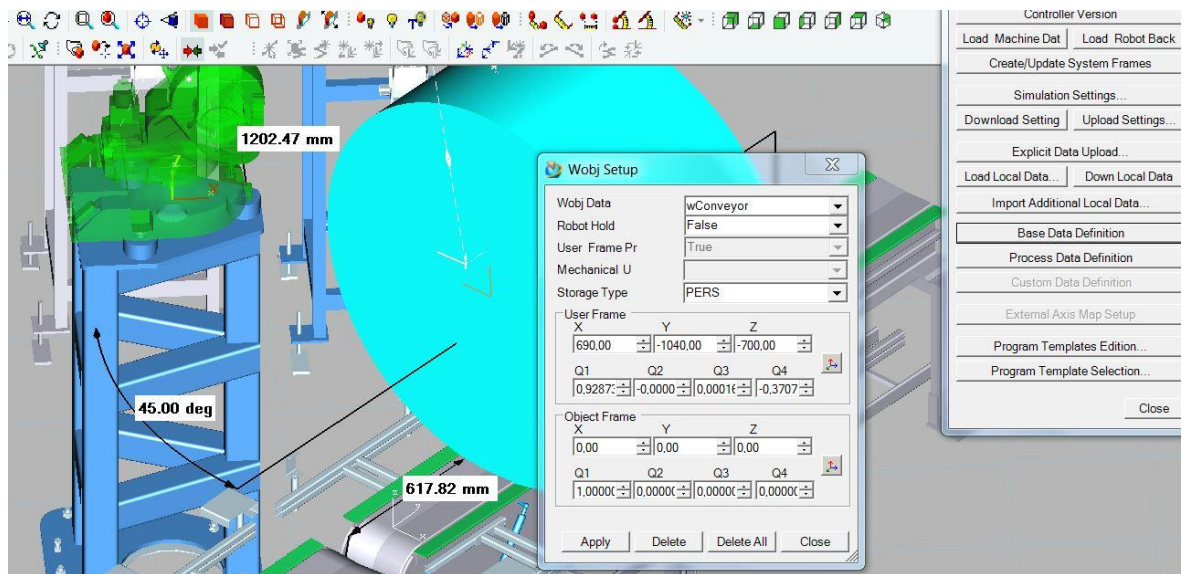
Path editor on tarkoitettu robotin ohjelmien avaamiseen ja yksityiskohtaiseen muokkaamiseen. Path editorissa robotin liikeparametrit säädetään kohdilleen. Seuraavassa listassa on esitelty tärkeimmät robotin liikkeille välttämättömät parametrit.

- Motion data, jossa määritellään robotin liikekäsken tyyppi.
- Speed data, jolla robotin sekä ulkoisten akselien liikenopeudet säädetään halutulle tasolle.
- Zone data määrittelee, miten lähellä ohjelmoitua asentoa robotin akselit ovat ennen siirtymistä seuraavaan pisteeseen.
- Tool data -kohdassa määritellään robotin käyttämä työkalu. Tiettyä operaatiota suoritettaessa täytyy olla valittuna jokin työkalu.
- Wobjdatan avulla määritellään, minkä työkohteen mukaan robotti suorittaa tehtävää.

Näitä datamäärittämiä kutsutaan robotin perustietojen määrittämiseksi. Uusia data-tietoja ja parametreja pystytään luomaan ja vanhoja muokkaamaan Robot Setup valikossa(Kuvio 32.), kohdasta Basedata definition.



Kuvio 22. Tool Data



Kuvio 23. Wobj Data

Jokainen paikoituspiste ja kiertymät akselien ympäri (Rx, Ry, Rz) on nähtävissä taulukossa. Motion-sarakkeessa voidaan robotille määrätä, millä liikekäskyllä siirtyminen paikoituspisteeseen suoritetaan. Duration-sarake kertoo työkierron kokonaispituuden, sekä eri liikkeiden ja toimintojen pituudet ajassa mitattuna. OLP Commands -sarakeeseen on määritelty I/O-signaalit, ehtolauseet ja screenmarker-komennot.

| Attachment | X | Y | Z | RX | RY | RZ | Motion T... | Joint Spe... | Speed | Time | Zone | Duration | OLP Commands |
|-----------------|--------|---------|---------|---------|-------|---------|-------------|--------------|-------|------|------|----------|----------------------------|
| | | | | | | | | | | | | 229.07 | |
| | | | | | | | | | | | | 229.07 | |
| main | | | | | | | | 0.00 | 0.00 | 0.00 | | 10.11 | !SetGO goPN80_SeqSte |
| pHome | 302.13 | 669.32 | 887.50 | -176.77 | 29.78 | -106.59 | Joint | 100.00 | 0.00 | 0.00 | fine | 1.00 | !SetGO goPN80_SeqSte |
| pWaitCoiPos | 801.16 | 424.78 | 927.82 | 174.34 | 16.16 | 127.78 | Joint | 100.00 | 0.00 | 0.00 | fine | 2.87 | !WaitDI diPN4_StackRea |
| pToLab1 | 256.50 | -845.00 | 1054.74 | -178.68 | 9.20 | 178.72 | Joint | 100.00 | 0.00 | 0.00 | fine | 4.01 | !GOTO end: ENDIF IF DI |
| pToLab2 | 848.90 | -696.77 | 1062.84 | 95.30 | 49.80 | 132.02 | Joint | 100.00 | 0.00 | 0.00 | fine | 2.23 | !GOTO end: ENDIF |
| RobIsStarted | | | | | | | | 0.00 | 0.00 | 0.00 | | 0.00 | !Robot settings after rese |
| RobFaultCode | | | | | | | | 0.00 | 0.00 | 0.00 | | 0.00 | !Set Fault output and ser |
| RobotToMain | | | | | | | | 0.00 | 0.00 | 0.00 | | 0.00 | !PP to main after reset E |
| RobIsReseted | | | | | | | | 0.00 | 0.00 | 0.00 | | 0.00 | !Robot settings after rese |
| ToFreeArea | | | | | | | | 0.00 | 0.00 | 0.00 | | 29.17 | !SetGO goPN80_SeqSte |
| DesJointAng | -63.69 | 6.33 | 1423.04 | 0.00 | 0.00 | 0.00 | Joint | 100.00 | 0.00 | 0.00 | fine | 5.57 | |
| pUpPosition | 764.87 | -826.69 | 1166.95 | 95.77 | 15.34 | 131.68 | Joint | 100.00 | 0.00 | 0.00 | fine | 2.64 | |
| pHome | 302.13 | 669.32 | 887.50 | -176.77 | 29.78 | -106.59 | Joint | 100.00 | 0.00 | 0.00 | fine | 2.24 | ELSEIF DOutput(DO10_1 |
| pToPrinter2 | 909.45 | 889.51 | -115.37 | 173.68 | 84.59 | -94.84 | Joint | 100.00 | 0.00 | 0.00 | fine | 4.38 | |
| pToPrinter21 | 857.56 | 806.53 | 8.86 | 179.77 | 58.01 | -89.75 | Linear | 100.00 | 0.00 | 0.00 | fine | 0.80 | |
| pHome | 302.13 | 669.32 | 887.50 | -176.77 | 29.78 | -106.59 | Joint | 100.00 | 0.00 | 0.00 | fine | 2.88 | ELSEIF DOutput(DO10_1 |
| pToPrinter1Apr | 30.03 | 1060.00 | -108.47 | 177.62 | 75.33 | -90.86 | Joint | 100.00 | 0.00 | 0.00 | fine | 3.53 | |
| pToPrinter1 | 35.89 | 974.62 | -21.62 | 177.79 | 62.61 | -90.73 | Joint | 100.00 | 0.00 | 0.00 | fine | 0.82 | |
| pHome | 302.13 | 669.32 | 887.50 | -176.77 | 29.78 | -106.59 | Joint | 100.00 | 0.00 | 0.00 | fine | 3.32 | elseif DOutput(DO13_Hk |
| pHome | 302.13 | 669.32 | 887.50 | -176.77 | 29.78 | -106.59 | Joint | 100.00 | 0.00 | 0.00 | fine | 3.00 | ELSE TPWrt |
| pHome | 302.13 | 669.32 | 887.50 | -176.77 | 29.78 | -106.59 | Joint | 100.00 | 0.00 | 0.00 | fine | 0.00 | endif |
| DataReading | | | | | | | | 0.00 | 0.00 | 0.00 | | 0.00 | !Get/check data from PLC |
| SuctionControl | | | | | | | | 0.00 | 0.00 | 0.00 | | 0.00 | !Control of suction ! Cont |
| PreSetting | | | | | | | | 0.00 | 0.00 | 0.00 | | 0.00 | ! Things which have to re |
| Printer1 | | | | | | | | | | | | 97.61 | |
| GetLabel1Print1 | | | | | | | | 0.00 | 0.00 | 0.00 | | 16.86 | MoveJ pToPrinter1.V100 |
| pToPrinter12 | -2.62 | 947.21 | 251.94 | 176.32 | 73.23 | -92.22 | Joint | 100.00 | 0.00 | 0.00 | fine | 7.98 | |
| pToPrinter1Apr | 30.03 | 1060.00 | -108.47 | 177.62 | 75.33 | -90.86 | Joint | 100.00 | 0.00 | 0.00 | fine | 4.29 | SetDO1_Cyl1_Front Wa |
| pPrinter | 30.04 | 1167.07 | -108.32 | 177.66 | 75.34 | -90.82 | Linear | 100.00 | 0.00 | 0.00 | fine | 1.67 | ResetDO1_Cyl1_Front. |
| pToPrinter1Apr | 30.03 | 1060.00 | -108.47 | 177.62 | 75.33 | -90.86 | Linear | 100.00 | 0.00 | 0.00 | fine | 1.17 | |
| pToPrinter1 | 35.89 | 974.62 | -21.62 | 177.79 | 62.61 | -90.73 | Joint | 100.00 | 0.00 | 0.00 | fine | 1.75 | |
| GetLabel2Print1 | | | | | | | | 0.00 | 0.00 | 0.00 | | 23.41 | |
| pToPrinter1 | 35.89 | 974.62 | -21.62 | 177.79 | 62.61 | -90.73 | Joint | 100.00 | 0.00 | 0.00 | fine | 2.33 | |
| pToPrinter1Apr | 30.03 | 1060.00 | -108.47 | 177.62 | 75.33 | -90.86 | Joint | 100.00 | 0.00 | 0.00 | fine | 2.28 | SetDO2_Cyl2_Front: W: |
| pPrinter | 30.04 | 1167.07 | -108.32 | 177.66 | 75.34 | -90.82 | Linear | 100.00 | 0.00 | 0.00 | fine | 1.67 | SuctionControl 3.1: Rese |
| pToPrinter1Apr | 30.03 | 1060.00 | -108.47 | 177.62 | 75.33 | -90.86 | Linear | 100.00 | 0.00 | 0.00 | fine | 1.17 | |
| pToPrinter1 | 35.89 | 974.62 | -21.62 | 177.79 | 62.61 | -90.73 | Joint | 100.00 | 0.00 | 0.00 | fine | 1.78 | |
| pHome | 302.13 | 669.32 | 887.50 | -176.77 | 29.78 | -106.59 | Joint | 100.00 | 0.00 | 0.00 | fine | 14.18 | |
| SetLabel1 | | | | | | | | 0.00 | 0.00 | 0.00 | | 13.59 | |

Kuvio 24. Robotin ohjelman avaaminen ja muokkaaminen Path Editorissa

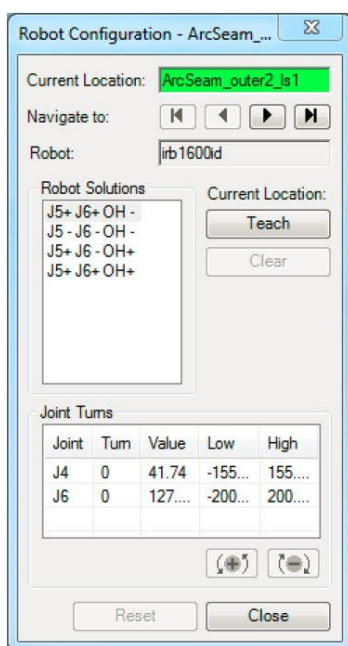
6.7.3 Robotin konfiguraatio

Robotin tarkkojen liikkeiden aikaansaamiseksi tärkeä ominaisuus Path Editorissa on konfiguraatioiden opettaminen robotille. Konfigurointi suoritetaan kaikissa robotin paikoituspisteissä.

Käsivarsirobotin liikkeet edellyttävät käänteistä algoritmia laskeakseen akseliensa asennot. Tämä algoritmi tuottaa yleensä vähintään kaksi ratkaisua tai akselien yhdistelmää, jotka mahdollistavat robotin saavuttamaan TCPF:n kohdepaikassa. (Siemens Industry Software 2011, 37.)

Robotilla voi olla useita asentoja, joista se yltää suorittamaan esimerkiksi kappaleen noutamisen. Automaattisella konfiguraation opettamisella kuitenkin saavutetaan optimaalisimmat asennot ja liikesarjat operaatioissa, joissain tapauksissa myös välttään laiterikoilta. Robotin konfiguraation voidaan opettaa myös manuaalisesti. Toisin sanoen liikkeet opetetaan robotille. Tämä on myös välttämätöntä, että robotin ohjelma voitaisiin ladata fyysiseen robottiin. (Siemens Industry Software 2011, 4.)

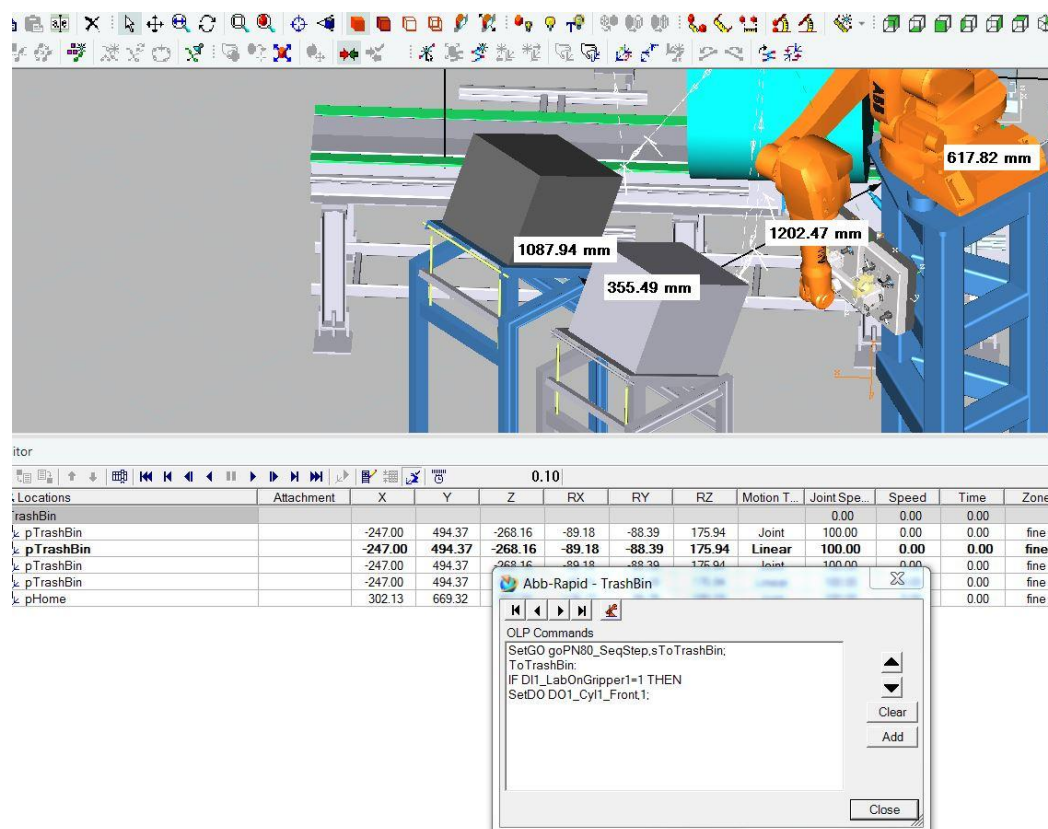
Robotin liikkeiden konfigurointi ei onnistunut ilman Motion parameters -tiedostoa, joka kopioitiin simulaatioon



Kuvio 25. Robotin konfiguraatio (Siemens PLM Software 2014).

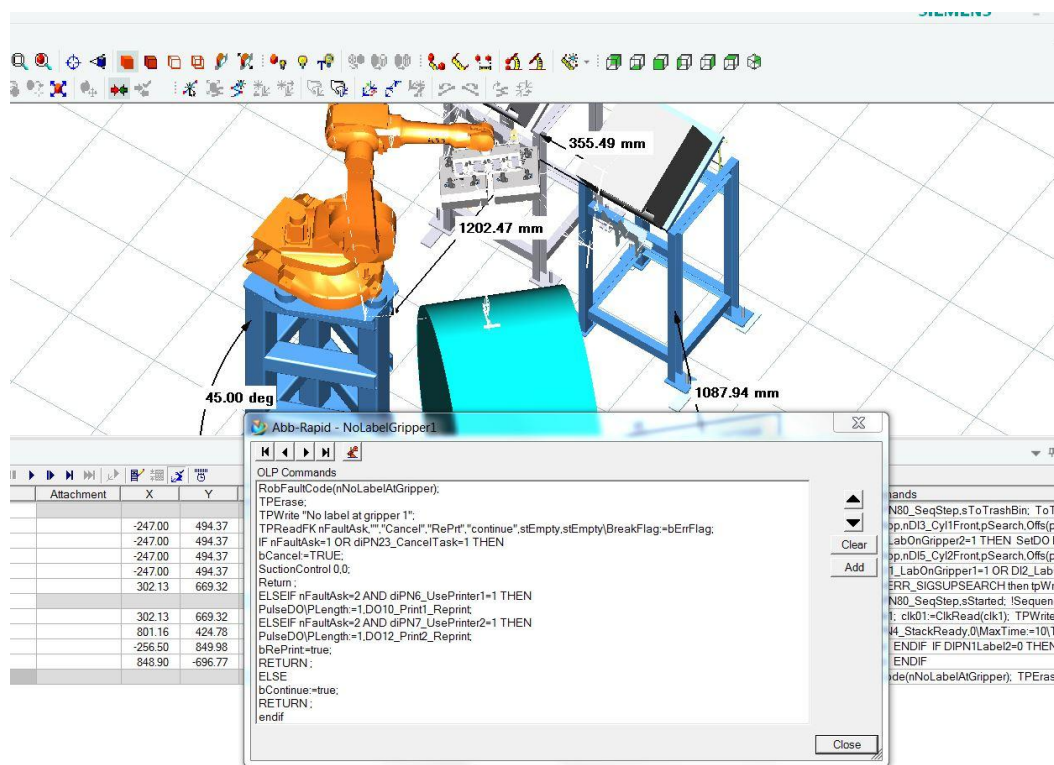
6.7.4 OLP-käskyjen lisääminen ohjelmaan

Riippumatta minkä valmistajan robotteja simuloidaan ja etäohjelmoidaan, RobotExpert-ohjelmisto vaatii asennettavaksi myös asennuspaketin kontrollerin parametrien määrittämiseen. Tämän tiedosto sisältää kontrollerin omat OLP-komennot. Ilman kyseistä tiedostoa ei RobotExpert pysty generoimaan robotin ohjelmaa robotin käyttämälle kielelle. Jokaisen valmistajan robottikontrollereilla on niiden omat OLP-käskyt. RobotExpert sisältää valmiita OLP-komentoja, jotka se generoi robotivalmistajan esimerkiksi ABB:n käyttämään muotoon. Tällaisia peruskäskyjä ovat työkalun käsittely, tavarankäsittely ja I/O-signaalien käsittely. (Siemens Industry Software, 134–136. [Viitattu 30.7.2015].)



Kuvio 26. OLP-komentojen lisääminen

OLP-komennot kirjoitetaan Path Editorissa OLP-kenttään. Kuviossa 21 on nähtävissä komentoja ABB:n Rapid-kielessä. Komentoja voidaan kirjoittaa kenttään Free Write -tyylillä tai vaihtoehtoisesti lisätä Add-painikkeen takaa generisiä työkalu- ja robottikomentoja (Kuvio 28), jotka ohjelma kääntää lopuksi ABB:n omalle kielelle.



Kuvio 27. OLP-komentojen lisääminen etikettirobotin ohjelmaan

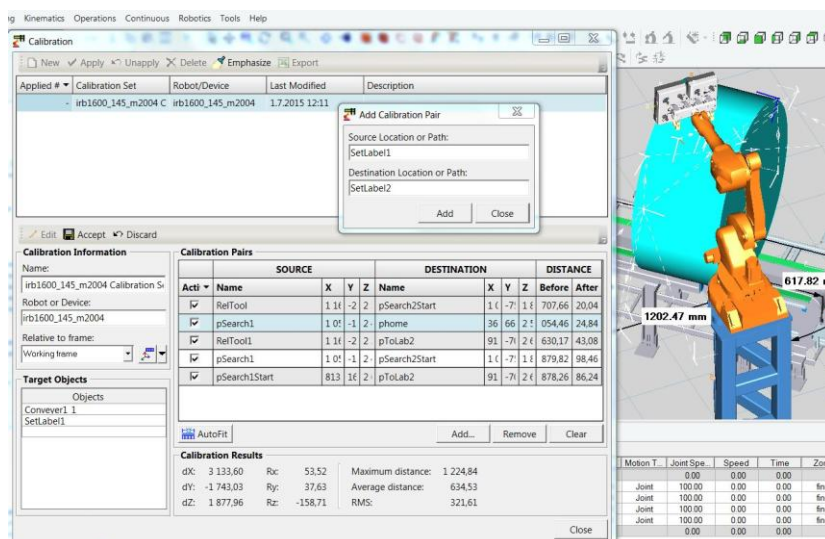
6.8 Solun kalibrointi

Kalibrointi on prosessi, jossa mallinnettu solu tehdään yhdenmukaiseksi fyysisen solun kanssa. Kalibrointi tapahtuu luomalla niin sanottu suodatin, jolla kuvataan, miten fyysinen solu eroaa mallinnetusta solusta.

Kalibrointi tulee tarpeeseen, kun prosessilta vaaditaan korkeaa tarkkuutta robotin liikkeille tai prosessissa, jossa robotilla on useita eri sijainteja. Kalibrointi vaatii joissain tapauksissa fyysisen solun mittaamista eri pisteistä ja asettamalla nämä mitat simulointiohjelman kalibrointityökaluun.

RobotExpert-ohjelman kalibrointityökalu käyttää algoritmeja säätääkseen lähde-kohteen asemoinnin ja orientaation vastaamaan fyysistä solua mahdollisimman tarkasti. Ohjelma suosittelee työkaluun syötettäväksi maksimissaan 10 asemointipistettä samanaikaisesti. Tällöin kalibrointi toimii halutulla tarkkuudella. (Siemens Industry Software, 160–161. [Viitattu 30.7.2015].)

Etiketirobotin solua simuloitaessa kalibrointi ei ollut tarpeen, koska robottia ohjelmoitiin samaan aikaan myös online-tilassa, ja robotti oli koko ajan läsnä ohjelmointia suoritettaessa. Lisäksi alkuperäinen toimilaitteiden sijainti koordinaatistossa muuttui muutamaa otteeseen.



Kuvio 28. Robotisoidun solun kalibrointi

6.9 Robotin kontrollerin parametrien määrittäminen

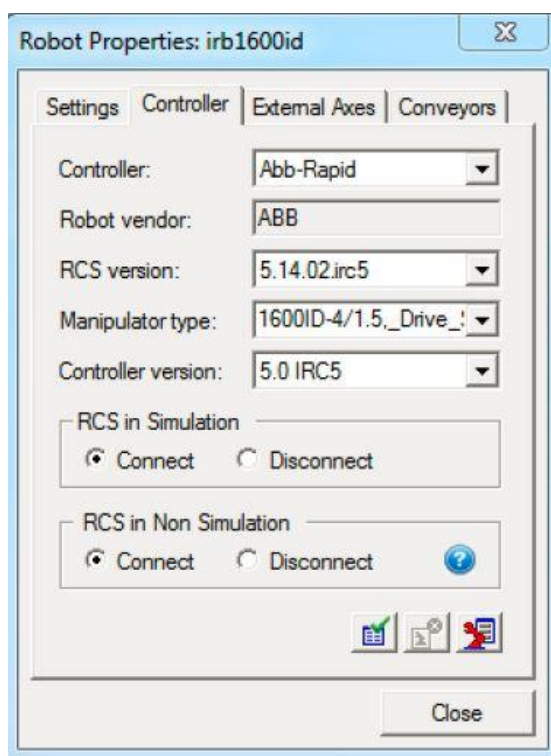
RobotExpert-ohjelmaan tuotava robotti sisältää oletuskontrollerin. Oletuskontrollerin tarkkuus on 80 prosentin luokkaa verrattuna oikeaan robottikontrolleriin. (Miroslav 2013, 33.) Robotin paikoituspisteiden sijainnit ovat samat, mutta liikerata jolla robotti siirtyy pisteeseen, voi olla erilainen. Vaihtoehtoisesti ohjelmassa voidaan käyttää robotille räätälöityä kontrolleri-pakettia, joka sisältää:

- robottikontrollerin simulaatiomodulin (RCS)
- etäohjelmointipaketin (OLP)

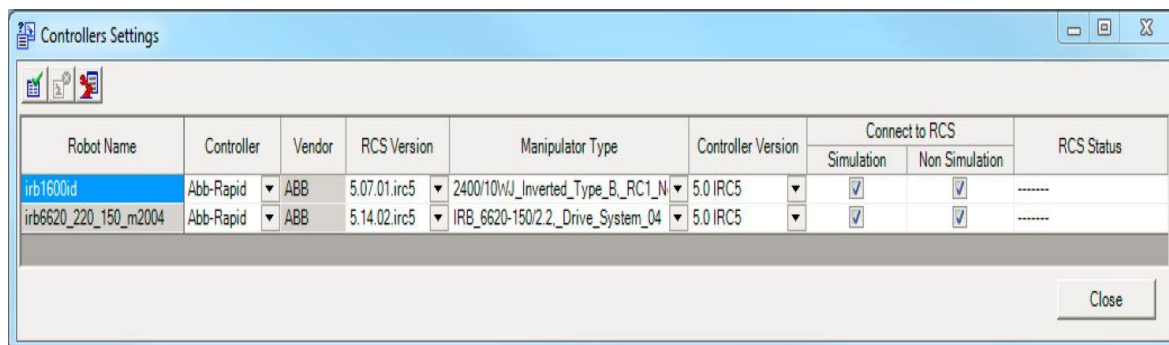
Robottivalmistajalle räätälöity RCS-moduuli mahdollistaa robotille tarkat liikeradat ja sijainnit koordinaatistossa. RCS-moduuli antaa tarkat määritelmät RobotExpertille, jotta ohjelmassa pystyttäisiin luomaan realistista robottisimulaatiota (RRS). (Miroslav 2013, 33.)

OLP-komentojen ja simuloinnin hienosäätöjen jälkeen solussa käytetyn robotin parametrit syötettiin kontrollerin asetuksiin. Asennetun ABB:n OLP-etäohjelmointipaketin avulla ohjelma tunnisti soluun asennetun robotin ABB:n valmistamaksi IRB1600-malliksi. Kieleksi valittiin vielä ABB:n Rapid-ohjelmointikieli ja kontrollerin versioksi 5.15 IRC5, joka oli uusin käytössä oleva päivitys. RCS-versio ja manipulaattorin tyyppi jätettiin tässä testissä laittamatta.

Kontrolleri tiedot voitiin määrittää robotin asetuksiin (kuvio 31) tai vaihtoehtoisesti suoraan kontrollerin asetusvalikon sisään (Kuvio 31).

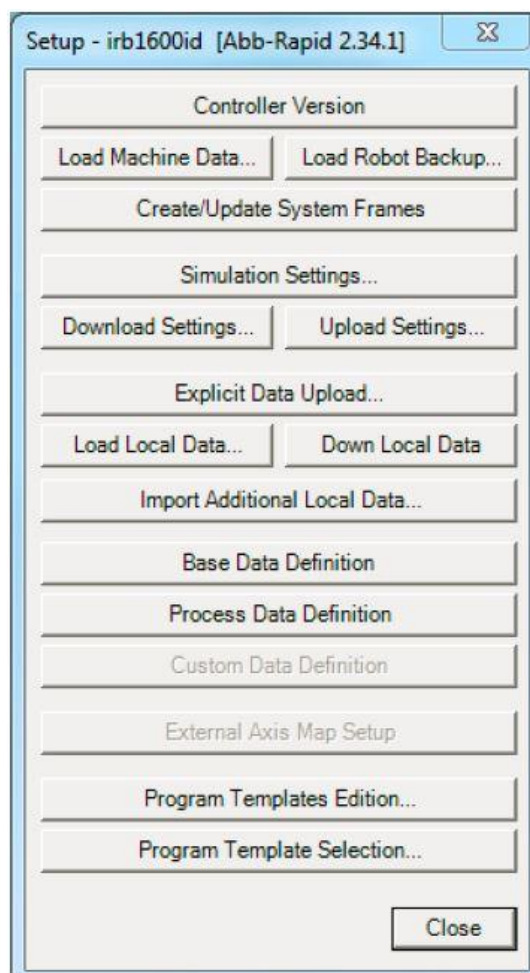


Kuvio 29. Robot Properties -valikko



Kuvio 30. Controller settings -valikko

Robot setup -valikko sisältää kaikki robotin liikkeille ja toiminnoille pakolliset parametrit. Jokaiselle kontrollerille (ABB, KUKA, Fanuc, Motoman) on omat parametrit. Tästä syystä myös valikko on jokaiselle kontrollerille erilainen. (Siemens PLM Robotics 2014, 135.)



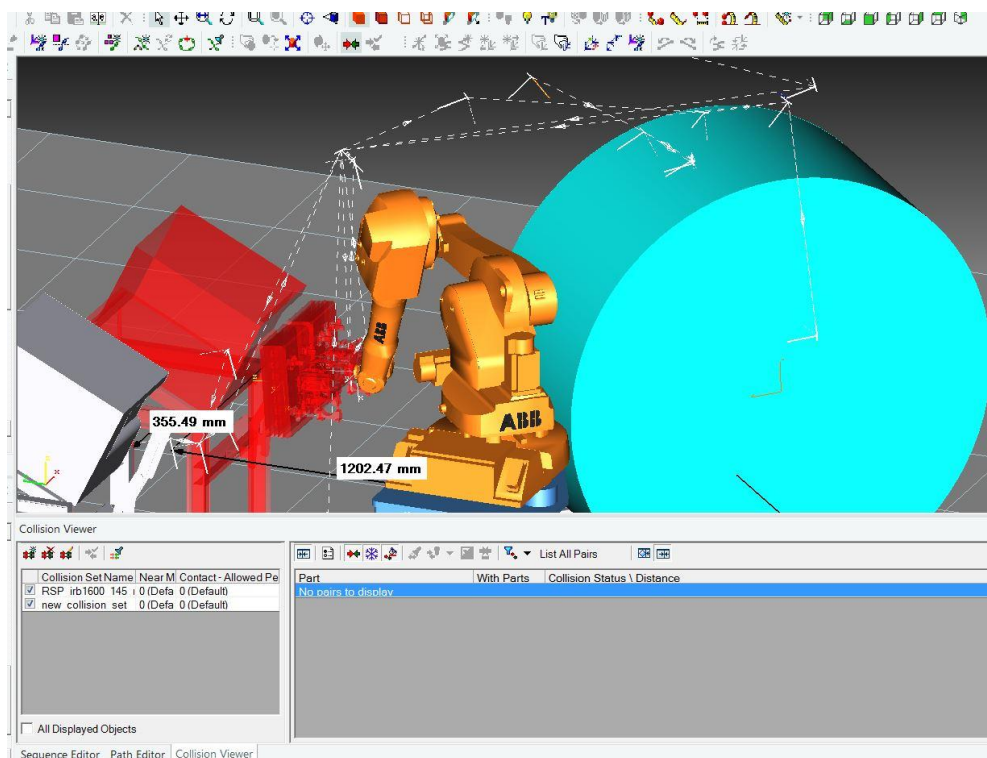
Kuvio 31. ABB Robot setup -valikko

6.10 Ohjelman simulointi ja hienosäätö

RobotExpert-ohjelmassa robotisoitua solua päästiin hienosäätämään ja analysoimaan monella tapaa. Hyödyllisiä työkaluja olivat esimerkiksi törmäystarkastelu, ulottuvuustesti, smart place -työkalu sekä swept volume -työkalu.

6.10.1 Törmäystarkastelu

RobotExpert auttaa suunnittelijaa dynaamisesti estämään robotin ja toimilaitteiden väliset törmäykset robottisimulaation ja liikkeen aikana. Törmäystarkastelulla pystytään ennalta ehkäisemään robottisolun läheisyydessä työskentelevien henkilövahingot sekä estämään laiterikot. Törmäystarkastelussa pystytään analysoimaan, tapahtuuko robottisolussa työkalun uppoaminen objektiin, törmäys tai ainoastaan läheltä piti -tilanne. Kun robotin työkalu tai käsivarsi osuu liikeradallaan johonkin objektiin, näkyy törmäyksen kohteena oleva objekti punaisena. Simulaatiossa kävi ilmi, että robotin työkalu osui etikettilaitteen etuosaan. Paikoituspistettä ja työkalun asemointia säätämällä pystyttiin tilanne korjaamaan helposti (kuvio 33).

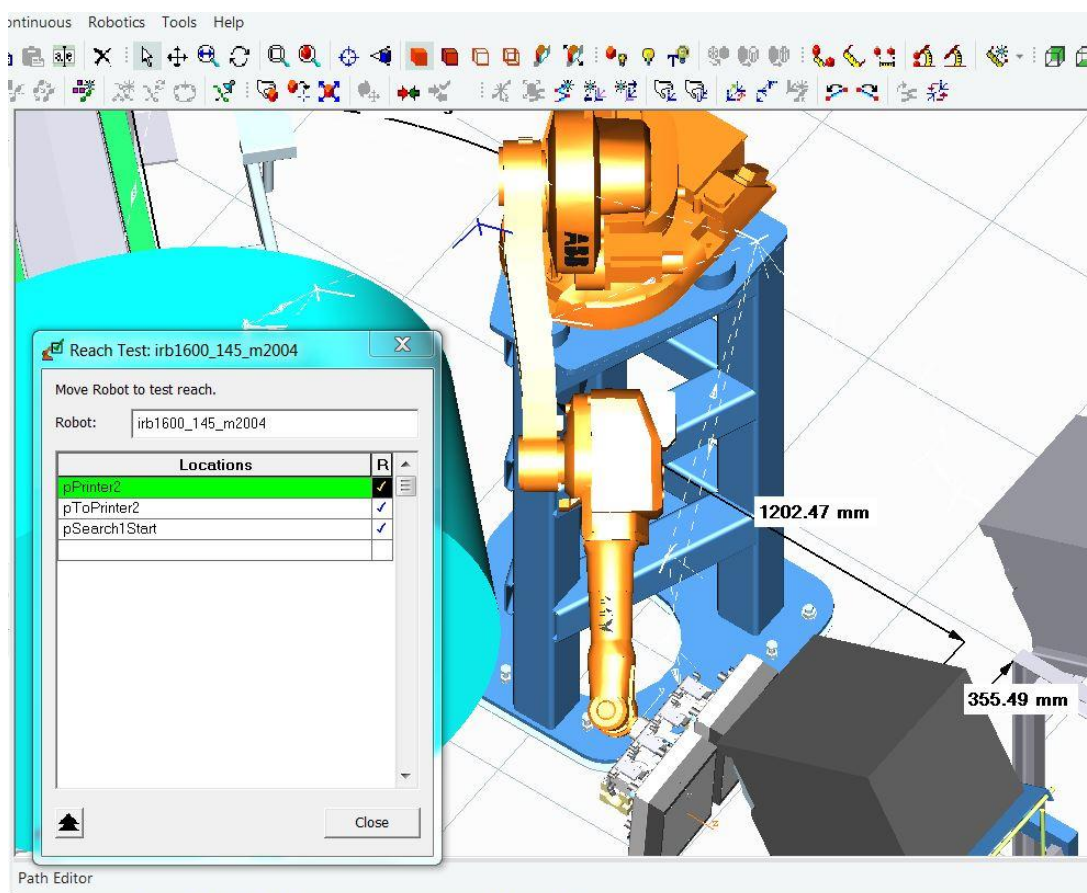


Kuvio 32. Törmäystarkastelu

6.10.2 Ulottuvuuden tarkastelun työkalu

Reach test -työkalua käytettiin hankalissa asennoissa varmistamaan, että robotti ulottui kaikkiin paikoituspisteisiin. Kuvio 34 näyttää tilanteen, jossa robotin ulottuvuutta testattiin otettaessa tarttujaan etikettiä tulostimelta numero kaksi.

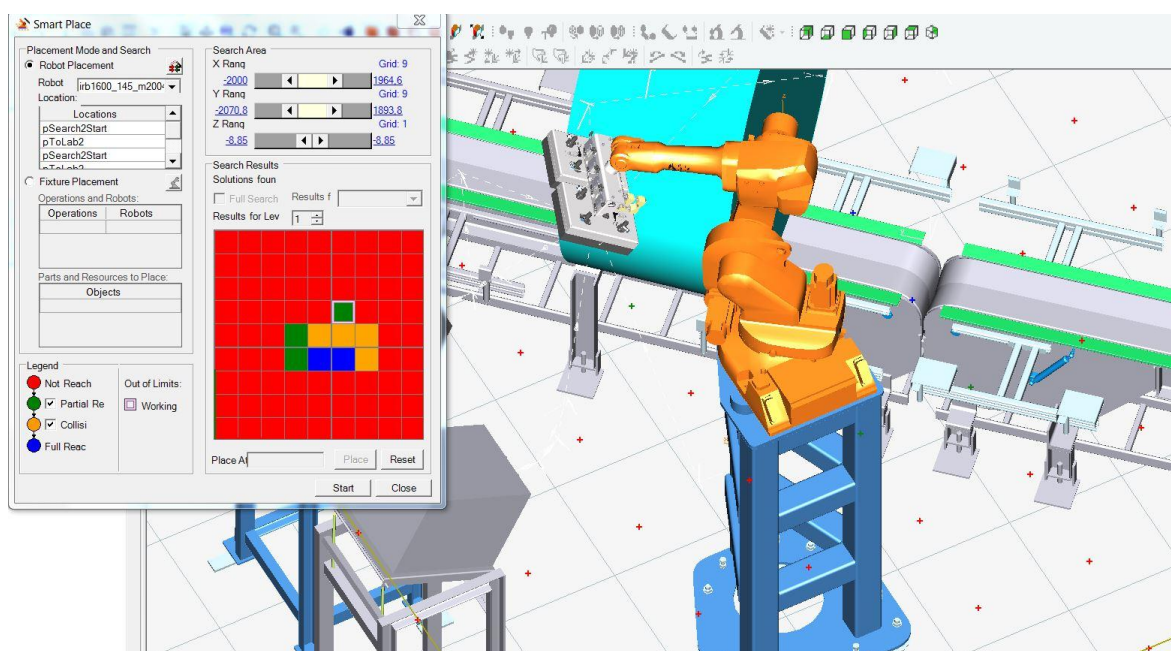
Ohjelmassa valittiin ensin haluttu operaatio. Tämän jälkeen ohjelma näytti sijainnin oikealla puolella $\sqrt{\text{}}$ -symbolin, jos robotti ulottui kyseiseen paikoituspisteeseen, ja x-symbolin jos ulottuvuus ei riittänyt.



Kuvio 33. Etikettirobotille suoritettu ulottuvuustesti

6.10.3 Smart Place -työkalu

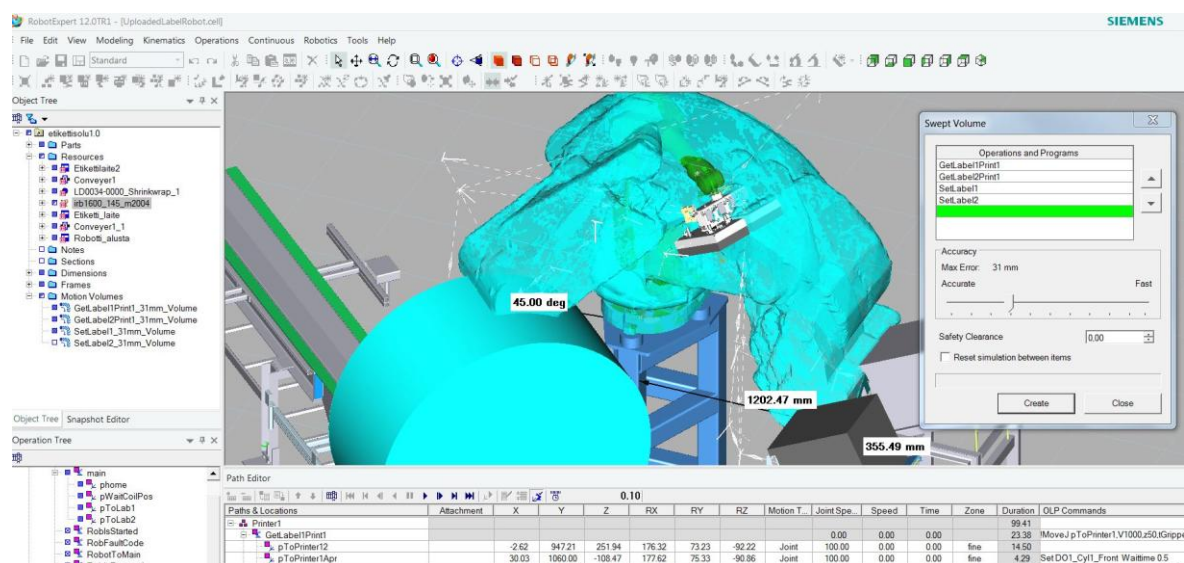
Smart place -työkalu auttaa robottia etsimään optimaalisimmat sijainnit robotille. Työkalu laskee robotin työkoordinaatistosta kaikki mahdolliset asennot, joihin roboti yltää halutussa operaatiossa. Siniset neliöt näyttävät robotin ulottuvuusalueet ja punaiset näyttävät robotille ulottumattomissa olevat sijainnit. Oranssi neliö näyttää sijainnit, joissa roboti mahdollisesti törmää johonkin objektiin. (Miroslav 2013, 29.)



Kuvio 34. Smart Place -toiminto

6.10.4 Swept Volume -työkalu

Swept volume -työkalun avulla testattiin, miten suurta aluetta robotin akselit ja työkalu käyttivät suorittaessaan ohjelmoitua työkiertoa. Swept volume -komento luo verhomaisen objektin, joka näyttää robotin käyttämän tilan pisteavaruudessa valitun operaation aikana. Työkalu helpottaa robotin asentojen optimoimista, etenkin prosesseissa, joissa työstedään homogeenisiä kappaleita yhdessä sijainnissa. Robotin käyttämä työalue prosessin aikana pystytään määrittämään 2 millimetrin tarkkuudella. (Siemens PLM 2014, 423.)



Kuvio 35. Swept Volume -toiminto

6.11 Ohjelman kääntäminen robotin käyttämään muotoon

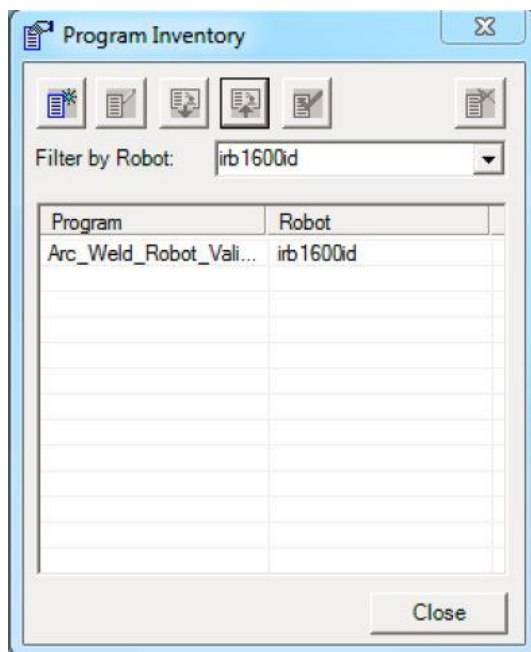
OLP-komentojen lisäämisen ja hienosäätöjen jälkeen simuloitu ja valmis ohjelma käännettiin robotin käyttämälle kielelle.

Operation tree -välilehdellä valittiin ohjelma eli compound operation, joka oli tarkoitus generoida. Tämän jälkeen Robotics-valikosta valittiin Create robotic program for compound operation -toiminto.

RobotExpert loi uuden robottiohjelman ja nimesi sen valitun operaation mukaan. Kaikki liikekäskyt ja I/O-signaalikomennot tallentuivat ohjelmaan samassa järjestyksessä, kuin ne oli linkitetty aiemmin sekvenssieditorissa.

Luotu ohjelma tallentui Program inventory -valikkoon (kuvio 37). Tässä vaiheessa ohjelmaa voitiin vielä tarvittaessa korjailia, ennen kääntämistä ABB:n rapid-kielelle. Program inventory -välilehdeltä löytyvät seuraavat ohjelman käsittelyyn tarvittavat toiminnot:

- Create new program -toiminto luo uuden ohjelman valitulle robotille.
- Open in Program Editor -toiminnolla saadaan luotu ohjelma avattua tekstimuodossa. Editori on ainoa tapa lisätä ohjelmaan uusia liikekäskyjä tai korjailia vanhoja.
- Download program -toiminnolla valittu ja luotu ohjelma voidaan ladata oikeaan robottikontrolleriin.
- Upload program -toiminnolla saadaan fyysiseltä robottikontrollerilta ladattua ohjelma RobotExpertin muokattavaksi.



Kuvio 36. Program Inventory
(Siemens PLM Software 2014).

```

File Edit Format View Help
%%
VERSION:1
LANGUAGE:ENGLISH
%%

MODULE MainModuleLabelRobot_Program
!# Process Simulate RobotExpert I2 TRI to ABB Rapid download
!# -----
!# CREATED : 31/07/2015 19:47:35 by TUOMAS LEPPALA
!# CELL : etikettisolu1.0
!# ROBOT : irb1600_145_m2004

!# -----
!# ----- ERROR CODES
!# -----
CONST num nNoFault:=0;
CONST num nWaitingLabel:=1;
CONST num nDataFault:=2;
CONST num nSearchError:=3;
CONST num nLabelAtGripperAfterLb1:=4;
CONST num nMovingNotAllowed:=5;
CONST num nNoLabelAtGripper:=6;

!# -----
!# ----- SEQUENCE STEPS
!# -----
CONST num sStarted:=0;
CONST num sToFreeArea:=1;
CONST num sToTrashBin:=2;
CONST num sWaitingTask:=3;
CONST num sDataReading:=4;
CONST num sGetLabel:=5;
CONST num sCheckLabel:=6;
CONST num sSetBody:=7;
CONST num sSetBack:=8;
CONST num sWaitingStack:=9;
CONST num sTaskCancelled:=10;

```

Kuvio 37. Ote ladatusta robotiohjelmasta, käännettynä robotin kielelle

7 TULOKSET JA POHDINTA

Opinnäytetyön pääasiallisena tavoitteena oli etäohjelmoida robotisoitu etikettisolu Siemens Tecnomatix RobotExpert -ohjelmistolla. Samalla oli tavoitteena myös yleisesti tutustua offline-ohjelmoinnin hyötyihin ja haittoihin.

Robotisoidun solun testaukset alkoivat samoihin aikoihin kun lisenssi RobotExpert-ohjelmistoon oli saatu toimittajalta. Toisin sanoen sekä online- että offline-ohjelmointia suoritettiin rinnakkain. Hyvänä lähtökohtana ja vertailuna, toimi myös ABB:n robottikoulutus Helsingissä, jossa koulutuksen yhtenä aiheena oli offline-ohjelmointi ABB:n Robot Studiolla.

RobotExpertin osalta etäohjelmointi myöhästyi hieman robottisolun testeistä, jotka suoritettiin Pesmelin Kauhajoen yksikössä toukokuun lopulla. Offline-ohjelmoinnin osalta ei siis täysin päästy tavoitteeseen. Tästä huolimatta tutustuminen offline-ohjelmointiin oli opettavaista ja erittäin mielenkiintoista.

Toisena tavoitteena oli offline-ohjelmoinnin hyötyjen etsiminen Pesmelin näkökulmasta. Tässä tavoitteessa mielestäni onnistuttiin hyvin. Tulevaisuudessa etäohjelmointi tulee mitä todennäköisimmin olemaan aina osana Pesmelin automatisoitujen järjestelmien suunnittelua. RobotExpert-ohjelmistoon ei päädytty tällä erää vielä hankinta mielessä, vaan ABB:n oma etäohjelmointi-ohjelmisto nähtiin Pesmelille hyödyllisemmäksi ja kustannustehokkaammaksi vaihtoehdoksi. Pesmel Oy käyttää projekteissaan useimmiten ABB:n valmistamia robotteja ja toisinaan myös KUKA:n robotteja. Geneerisellä etäohjelmointiohjelmistolla ei tästä syystä ole varsinaista hyötyä yritykselle.

LÄHTEET

- Aalto-yliopisto. 2010. Teollisuusrobotit. [PDF- julkaisu]. Aalto-yliopisto. [Viitattu 21.4.2015]. Saatavissa: https://noppa.aalto.fi/noppa/kurssi/as-116.1100/luennot/AS-116_1100_luentokalvot_6.pdf
- Aalto-yliopisto. 2010. Tiedonsiirto. [PDF-julkaisu]. Aalto-yliopisto. [Viitattu 20.5.2015]. Kenttäväylät ja OPC. Saatavana: https://www.noppa.aalto.fi/noppa/kurssi/as-116.1100/luennot/AS-116_1100_luentokalvot.1100_s10_kenttavaylat_ja_opc_web.pdf
- Delfoi. Ei päiväystä. Robotiikkaa piensarjoille. [verkkolähde]. Delfoi. [Viitattu 21.4.2015]. Saatavana: http://www.delfoi.com/web/solutions/robotiikka/fi_FI/piensarjoille/
- Finnrobotics. Ei päiväystä. Offline-ohjelmointi. [verkkolähde]. Finnrobotics Oy. [Viitattu 15.4.2015]. Saatavana: <http://www.finnrobotics.fi/?sivu=offlineohjelmointi>
- Järvinen. J. 2012. Delfoi hitsauskäyttöliittymän käyttöohje. [verkkolähde]. Opin- näytetyö. Tampereen ammattikorkeakoulu. Kone- ja tuotantotekniikka. Modernit tuotantojärjestelmät. [Viitattu 21.8.2015]. Saatavissa: <https://www.theseus.fi/handle/10024/42128>
- Kaarela, J. 2007. Robotin etäohjelmoinnin käyttöönoton vaiheet robotisoidussa ohutlevyn särmäyksessä. [verkkolähde]. Opinnäytetyö. Keski-Pohjanmaan ammattikorkeakoulu. Tekniikan ylempi ammattikorkeakoulu. Teknologiaosaa- misen johtaminen. [Viitattu 13.8.2015]. Saatavissa: <http://www.cou.fi/opinnaytetyot/julkaistut/Kaarela.pdf>
- Kuivanen, R. 1999. Robotiikka. Vantaa. Suomen Robotiikkayhdistys Ry. Talentum. [Viitattu 21.4.2015].
- Miroslav, K., 2013, Simulation of production processes. [verkkolähde]. Diploma Thesis. Czech technical University in Prague. Faculty of electrical Engineering. Cybernetics and Robotics. Systems and Control. [Viitattu 12.8.2015]. Saatavana: https://support.dce.felk.cvut.cz/mediawiki/images/f/fa/Dp_2013_konopa_miroslav.pdf
- Pesmel Oy. Ei päiväystä. Internal logistics, storing and packing solutions. [Verkkosivu]. Pesmel Oy. [Viitattu 13.6.2011]. Saatavana: <http://www.pesmel.com/english/company/>
- Pesmel Oy. Ei päiväystä [www-lähde]. Yrityksen intranet. [viitattu 19.8.2015]. Vaatii käyttöoikeuden.

- Robinson, S. 2004. Simulation. The Practice of Model Development and Use. [PDF-julkaisu]. S. Robinson. [Viitattu 21.4.2015]. Saatavana: http://197.14.51.10:81/pmb/GENIE_DES_PROCEDES/Simulation%20The%20Practice%20of%20Model%20Development%20and%20Use.pdf
- Siemens PLM 2011. Siemens Industry software. Motion planning definition. [PDF-julkaisu]. [Viitattu 30.7.2015.]. julkaisematon.
- Siemens PLM Robotics 2014. Tecnomatix RobotExpert. Quick Start Guide. [PDF-julkaisu]. Siemens PLM. [Viitattu 29.7.2015.6.2015]. Julkaisematon.
- Siemens PLM Robotics. Ei päiväystä. RobotExpert. Advanced Robotics (OLP) [PDF-julkaisu]. [Viitattu 30.7.2015]. Julkaisematon.
- Siemens PLM Software. 2008. Robotics. Simulation Validating & Commissioning the Virtual Work cell. [PDF-julkaisu]. Siemens PLM Software. [Viitattu 21.6.2015]. Saatavana: https://www.plm.automation.siemens.com/en_us/Images/Siemens%20PLM%20Robotics%20WP-%20Final_tcm1023-58463.pdf
- Siemens PLM software. 2012. RobotExpert. 'Plug-n-Play' software solution for robotics simulation and programming. [PDF-julkaisu]. Siemens PLM Software. [Viitattu 10.8.2015]. Saatavana: http://www.plm.automation.siemens.com/en_us/products/tecnomatix/manufacturing-simulation/robotics/robotexpert.shtml
- Siemens PLM. 2014. Student Guide. [PDF-julkaisu]. Process Simulate Standalone Basic Robotic simulation. [Viitattu 12.8.2015]. Julkaisematon.
- Tuunanen, T. 2007. Teollisuusrobotin käyttöönotto ja ohjelmointi. [verkkolähde]. Opinnäytetyö. Mikkelin ammattikorkeakoulu. Sähkötekniikka. [Viitattu 20.9.2015]. Saatavissa: https://www.theseus.fi/bitstream/handle/10024/75698/Tuunanen_Tommi.pdf?sequence=1