TAMPEREEN
AMMATTIKORKEAKOULU

JÄYSTEENPOISTOSOLU, ROBOTIN VOIMA-ANTURIN KÄYTTÖÖNOTTO JA SOLUN AUTOMATISOINTI

Mikael Pulakka

Opinnäytetyö
Huhtikuu 2019
Konetekniikka
Koneautomaatio

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Konetekniikka
Koneautomaatio

PULAKKA, MIKAEL:
Jäysteenpoistosolu, voima-anturin käyttöönotto ja solun automatisointi

Opinnäytetyö 71 sivua, joista liitteitä 15 sivua
Maaliskuu 2019

Opinnäytetyössä käydään läpi JTA Connectionille suunnitellun robottisolun automatisoinnin ja robotin voima-anturin käyttöönoton eri vaiheita. Opinnäytetyön toimeksiantajana toimi JTA Connection Oy. Opinnäytetyö koostuu robottisolun automatisoinnista, jossa tutustuttiin robotin ja PLC:n väliseen ohjelmointiin, sekä voima-anturin käyttöönotosta.

Projekti toteutettiin kahtena erillisenä osana. Projekti jaettiin toimeksiantajan toiveesta voima-anturin käyttöönottoon ja robottisolussa käytetyn konenäköjärjestelmän käyttöönottoon.

Opinnäytetyön tavoitteena oli suorittaa onnistuneesti voima-anturin käyttöönotto. Voima-anturia oli tavoitteena käyttää koneistettujen kappaleiden jäysteiden poistamisessa. Voima-anturilla piti saada robotti pitämään tasainen voima robotin työkalun ja kappaleen välillä. Robottisolun automatisoinnin tavoitteena oli luoda yhteys robotin ja järjestelmässä käytetyn ohjelmoitavan logiikan välille.

Automatisointi toteutettiin käyttämällä Beckhoff:n ohjelmoitavaa logiikkaa. Automatisoinnissa logiikalle luotiin ohjelma, jossa logiikan ja robotin väliset signaalit järjestettiin jatkoa varten. Robotin ohjelmointi ja voima-anturin käyttöönotto toteutettiin Fanuc:n M20iB/25 robotilla sekä FS-40iA voima-anturilla. Logiikan ja robotin ohjelmoinnissa tutkittiin laitteiden käyttäytymistä osana järjestelmää.

Työssä saavutettiin tavoitteet robottisolun automatisoinnissa sekä robotin voima-anturin käyttöönotossa. Automatisoinnissa saatiin robotin ja ohjelmoitavan logiikan välille luotua ohjelma, jonka avulla laitteet pystyivät siirtämään asetettuja arvoja. Voima-anturin käyttöönotossa anturia ja sen toimintoja pystyttiin käyttämään kappaleiden jäysteiden poistossa.

Opinnäytetyössä käytetyn voima-anturin toimintaa tulee kehittää, jotta kappaleiden työstön tarkkuutta saadaan parannettua. Vaihtoehtoisia menetelmiä opinnäytetyössäkäytetyille menetelmille tulee miettiä nykyisen järjestelmän rinnalle, jotta järjestelmän toiminta saataisiin varmemmaksi.

Asiasanat: JTA Connection, robottisolu, plc, voima-anturi

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Mechanical Engineering
Machine Automation

PULAKKA, MIKAEL
Deburring robot cell, commissioning force sensor and cells automation

Bachelor's thesis 71 pages, appendices 15 pages
March 2019

This thesis examines the different phases of robot cell automation designed for JTA Connection and the deployment of a robot force sensor. The thesis was commissioned by JTA Connection Oy. The thesis consists of robot cell automation, which introduced the programming between the robot and the PLC, as well as the introduction of the force sensor.

The project was implemented in two separate parts. The project was divided into the request of the principal for the commissioning of the force sensor and the commissioning of the machine vision system used in the robot cell.

The purpose of the thesis was to successfully commission the force sensor. The force sensor was designed to be used to remove burr of the machined parts. The force sensor had to get the robot to hold a steady force between the robot tool and the part. The purpose of the robot cell automation was to establish a connection between the robot and the programmable logic used in the system.

Automation was carried out by using Beckhoff's programmable logic. In automation, a program was created for the logic where the signals between the logic and the robot were arranged for the continuation. Robot programming and force sensors commissioning were carried out with Fanuc's M20iB / 25 robot and FS-40iA force sensor. The logic and the robot programming examined the behavior of the devices as part of the system.

Objectives for the robot cell automation and robot force sensor commissioning were achieved. In the automation, a program was created between the robot and the programmable logic to enable the devices to transfer the set values. In the commissioning of the force sensor, the sensor and its functions could be used to remove burr from the parts.

The work of the force sensor used in the thesis must be developed in order to improve the accuracy of the workpiece machining. Alternative methods for the methods used in the thesis should be considered alongside the current system to make the system more secure.

SISÄLLYS

1 JOHDANTO 7
2 AUTOMAATIO 8
2.1.1 PLC 8
2.1.2 B&R Panel PC 9
2.1.3 FANUC M-20iB/25 10
2.1.4 Fanuc FS-40iA voima-anturi 11
3 OHJELMOINTI 13
3.1 PLC käyttöönotto 13
3.2 Robotin ja PLC:n I/O-mapping 17
3.3 Muuttujat 21
4 ROBOTIN KÄYTTÖÖNOTTO 23
4.1 Työkalut 23
4.1.1 Kuorman määritys 23
4.1.2 Työkalupisteet 24
4.2 I/O-mapping 24
4.2.1 Digitaalisten tulojen ja -lähtöjen määritys 25
4.2.2 Ryhmien määritys 26
4.3 Ohjelmisto-optiot 27
5 VOIMA-ANTURIN KÄYTTÖÖNOTTO 28
5.1 Asennus 28
5.2 Käyttöönotto ja konfigurointi 29
5.2.1 Optiot 29
5.2.2 Voima-anturin ohjelmiston käyttöönotto 31
5.3 Funktiot 33
5.3.1 Constant push- / face match-funktio 33
5.3.2 Shaft insert- / groove insert- / square insert -funktio 36
5.3.3 Phase match insert -funktio 38
5.3.4 Search-funktio 40
5.3.5 Contouring -funktio 42
5.3.6 Threading-funktio 45
5.4 Funktioiden testit 46
5.4.1 Constant push -funktion kokeet 46
5.4.2 Contouring -funktion kokeet 49
5.4.3 Ratkaisut 52
6 POHDINTA 54
6.1 Tulokset 54

6.2 Jatkokehitys 54
LÄHTEET 56
LIITTEET 57
Liite 1. Beckhoff CX9020 esite 57
Liite 2. B&R PPC 2100 60
Liite 3. Fanuc M-20iB/25 Technical documents 61
Liite 4. Fanuc FS-40iA Force Sensor 62
Liite 5. Pushing_Dir_Auto_Change_2 – ohjelma 69

ERITYISSANASTO

PLC	Programmable logic controller eli ohjelmoitava logiikka
I/O mapping	Ohjelmoitavan logiikan tulojen ja lähtöjen paritus toisen laitteen tulojen ja lähtöjen kanssa
EtherCAT	Ohjelmoitavan logiikan tiedonsiirrossa käytetty väylätyyppi
Master	Ohjaava laite
UI	User Interface eli käyttöliittymä
Array	Tietorakenne, joka sisältää määrätyn määrän elementtejä.

1 JOHDANTO

Opinnäytetyön tarkoituksena oli luoda kohdeyritykselle robottisolu harjoitus- ja koulutuskäyttöön. Robottisolun piti olla sellainen, että sitä voitaisiin käyttää yksilöllisten kappaleiden jäysteenpoistossa. Työssä oli tavoitteena saada voima-anturin käyttöönotto suoritettua niin, että voima-anturia pystyttiin hyödyntämään yhdessä konenäön ja robotin kanssa, jotta yksilöllisestä koneistetusta kappaleesta saatiin poistettua jäysteet automaattisella työkierrolla. Automaation ja ohjelmoitavan logiikan osalta tavoite oli saada robotin ja logiikan välinen yhteys luotua myöhäisempiä jatkotoimenpiteitä varten.

Työ toteutettiin yritykselle JTA Connection Oy, joka toimi opinnäytetyön toimeksiantajana. JTA Connection on suomalainen yritys, joka sijaitsee Tampereella. Yritys tekee pääosin automaatiojärjestelmiä niiden suunnittelusta aina käyttöönottoon asti. JTA Connectionin edustaja oli valinnut ennalta määrätyt komponentit, joilla työ suoritettiin.

Opinnäytetyössä käydään läpi jäysteenpoistosolussa käytettyjen keskeisimpien laitteiden toimintaa, asennusta sekä ohjelmointia. Työssä käytettiin Beckhoff:n ohjelmoitavaa logiikkaa, B&R:n valmistamaa paneeli PC:tä, Fanuc M20iB/25 teollisuusrobottia sekä Fanuc FS40iA voima-anturia.

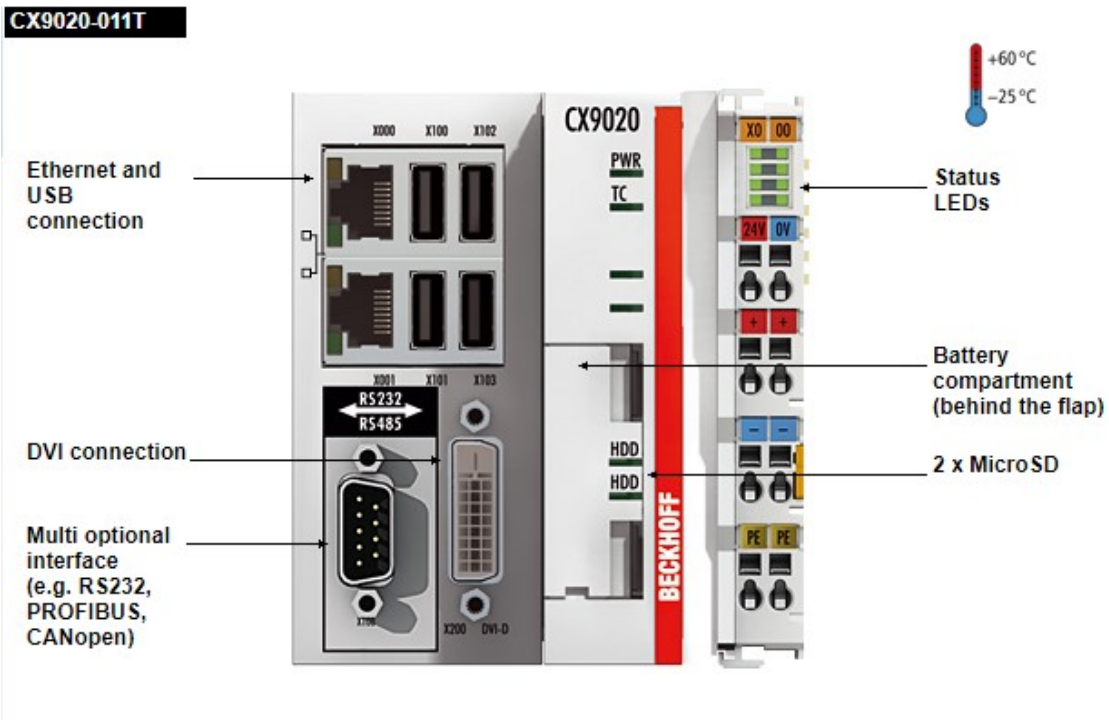
Työssä esitellään automaatiokomponenttien ominaisuuksia sekä niiden asennusta. Ohjelmoinnissa esitellään ohjelmoitavan logiikan käyttöönottovaiheet sekä robotin ja logiikan välisen kättelyn muodostaminen. Robotin ja voima-anturin käyttöönotot keskittyvät näiden laitteiden asennuksen sekä käyttöönoton aikana tehtyihin töihin sekä johtopäätöksiin. Voima-anturiin liittyviä ominaisuuksia ja toimintoja on eritelty voima-anturin käyttöönotossa tarkemmin.

2 AUTOMAATIO

Tässä kappaleessa kerrotaan robottisolussa käytetyistä komponenteista. Komponentit olivat ennalta valittuja työn toimeksiantajan toimesta. Robottisolun toiminnan kannalta keskeisiä mekaanisia komponentteja ei käydä tässä kappaleessa läpi, koska mekaniikkasuunnittelu toteutettiin erillisenä työnä, eikä se ollut osana opinnäytetyötä.

2.1.1 PLC

Solussa oli käytössä Beckhoffin valmistama ohjelmoitava logiikka. Logiikan prosessorimoduulina käytettiin CX9020 prosessori moduulia. Beckhoffin logiikassa on Windows-käyttöjärjestelmä. Tässä moduulissa oli kuvan 1 mukaisesti kaksi Ethernet-porttia, joista logiikan voi haaroittaa muihin laitteisiin saman verkon sisään. Moduulissa oli myös muita portteja kuten USB-, DVI- ja RS232-portti. Tarkemmat laitteen tiedot saatiin Beckhoff CX9020 esitteestä (Liite 1).



KUVA 1. Beckhoff CX9020 (https://www.beckhoff.com/english.asp?embedded_pc/cx9020.htm)

Prosessorimoduuliin syötetään myös digitaali- ja analogiakorttien käyttämä käyttöjännite, joka on kyseisessä järjestelmässä 24 volttia. Kyseisessä järjestelmässä logiikassa oli

kuusi turvakorttia, joista kolme korttia olivat EL1918 digitaalitulo-kortteja, kaksi EL2904 digitaalilähtö-korttia ja yksi EL6910 TwinSAFE-logiikkakortti. Järjestelmässä käytettiin myös viittä digitaalilähtö- ja viittä digitaalitulokorttia sekä yhtä analogiatulo-korttia.

Tulo- ja lähtökorttien määrän vuoksi järjestelmään piti lisätä yksi jännitteenjakokortti, jonka avulla varmistettiin, että kaikille järjestelmän korteille riitti virtaa. Jokaisella kortille on annettu virta, jonka kortti voi käyttää. Jos korttien yhteenlaskettu virta ylittää prosessorimoduulin antaman virran, pitää järjestelmän toimivuuden takaamiseksi lisätä erillinen jännitteenjakokortti varmistamaan, että kaikki kortit saavat tarvittavan määrän virtaa.

2.1.2 B&R Panel PC

Näyttöpaneelina solussa käytettiin B&R:n valmistamaa PPC2100 paneelitietokonetta. Tämän paneelitietokone toimi robottisolun käyttöliittymänä, josta oli tarkoitus antaa työstettävien kappaleiden parametrit järjestelmään sekä hallita robottia ja sen toimintoja. Paneelia käytettiin jo työn aikana varsinkin robotin konenäön määrityksessä.

Paneelitietokoneessa oli teknistentietojen mukaan Intel Atom E3827 1.17 GHz kaksiytiminen prosessori sekä 4 Gb SDRAM-muistia. Tietokoneen tekniset tiedot esitettiin tarjouksessa saadussa esitteessä (Liite 2), josta poistettiin yrityksen pyynnöstä laitteen tarjousnumero sekä laitteen sarjanumero. Paneelitietokoneen ollessa erittäin kompakti laite, näillä komponenteilla pystyttiin koneella suorittamaan kevyitä ohjelmia, joita pystyttiin tarvittaessa hyödyntämään työssä. Myös robottisolun jatkokäyttöä ajatellen paneelitietokoneella pystyttäisiin tekemään kokeellisia vedoksia erilaisista käyttöliittymistä ja kokeilemaan näiden toimivuutta.

Paneelitietokoneeseen oli tämän käyttöliittymän lisäksi varattu kolme painonappia, joiden toimintoina olivat käynnisty, pysäytys ja kuittaus. Näiden painikkeiden lisäksi paneelista löytyi avainkytkin, jota oli tarkoitus käyttää järjestelmän manuaali- ja automaattitilan valinnassa sekä hätä-seis-painike. Kaikki painikkeet kaapeloitiin yhdellä kaapelilla paneelitietokoneelta sähkökeskukselle, jossa niille oli varaus PLC:lle kytkemiseen.

2.1.3 FANUC M-20iB/25

Yrityksen valitsemana robottisolussa käytettiin kuvan 2 mukaista Fanuc M-20iB/25 -teollisuusrobottia. Robotin ulottuma on 1853 mm, joka soveltui solun koko huomioon ottaen hyvin työtehtävään. Robotin kaapelit olivat integroituja robotin käsivarren paneloinnin sisään ja jokaisen robotin nivel oli IP-luokitukseltaan IP67. Robotin käsittelykyky oli 25 kg, joka oli tarpeeksi työstämään ja käsittelemään solun käytössä olevia kappaleita. Voima-anturille on spesifioitu yhteen sopivia robotteja ja voima-anturille FS-40iA robotti M-20iB/25 on yksi yhteensopivista malleista.

KUVA 2. Robottisolu – Fanuc M-20iB/20

Robotin koon mukaan myös esimerkiksi työtaso, jolla robotti työskentelee, suunniteltiin oikean kokoiseksi, jotta robotti ylettyy taso eri kohtiin ilman, että robotille piti suunnitella erillistä jalustaa. Robotti asennettiin solualueen lattiaan käyttämällä robotin ja asennus tason välissä asennuslevyä. Robotin nivelten sisällä kulki pneumatiikkaletkut ja robottiin oli sisäänrakennettu kolme 5/3-venttiiliä, joissa jokaisessa oli suljettu keskiasento. Myös voima-anturin kaapelointi sekä muut työkalussa tarvittavat kaapelit olivat integroitu robotin käsivarteen. Näiden kaapeleiden toisen pään liitin oli robotin takana sen alaosassa

ja toinen pää oli sijoitettu robotin 4-akselin kylkeen paneelin alle. Robotin ulottuma, käsittelykyky ja muut tekniset tiedot saatiin valmistajan verkkosivuilta käytetyn robotin esitteestä (liite 2).

Robotti sopisi ominaisuuksiltaan työskentelemään myös puhtaissa tiloissa, elintarviketeollisuudessa sovelluksissa, joissa robotti pesisi kappaleita korkean tiiviysluokkansa ansiosta. Robotti on ominaisuuksiltaan täysin pölytiivis ja kestää upotuksen hetkellisesti veteen 1m syvyyteen.

2.1.4 Fanuc FS-40iA voima-anturi

Robottisolussa käytettiin jäysteenpoiston voiman tunnistuksessa Fanuc FS-40iA voima-anturia (Kuva 3). Voima-anturin tarkoituksena on yhdessä sen ohjelmien kanssa hallita työstössä käytettyä voimaa. Voima-anturin on tarkoitus pitää voima vakiona koko työstön ajan. Voima-anturilla pystytään myös vaihtamaan työntövoiman suuntaa suhteessa siihen, mihin suuntaan robotti kussakin tilanteessa liikuttaa työkaluaan. Voima, jota työstössä käytetään, voidaan määritellä kappalekohtaisesti voima-anturin parametreistä, joka mahdollistaa tasaisen laadun poistettaessa materiaalia kappaleista.

KUVA 3. Fanuc force sensors (http://ryeok.com/robot-force-sensing)

Opinnäytetyön robottisolussa käytettiin edellä mainittua FS-40iA voima-anturia, jonka käsittelykyky yltää 40 kg asti. Voima-anturin suurin sallittu voima oli siis noin 392,4 Newtonia. Tämä kokoluokka oli sopiva kyseiseen työstöön, sillä vaikka työstössä käytetyt voimat pyrittiin pitämään mahdollisimman matalina niin 40 kg käsittelykyky mahdollisti robotin täyden 25 kg käsittelykyvyn hyödyntämisen tarvittaessa. Robotin käsittelykykyä voitiin suunnitella käytettäväksi saman robotin tulevaisuuden työtehtävissä, vaikka robotissa oli voima-anturi kytkettynä.

Voima-anturin ohjelmallisiin ominaisuuksiin kuului muun muassa data viewer, jonka tarkoituksena oli, että voima-anturin työstön aikana tallennettua tietoa pystyttiin tutkimaan työstön päätyttyä. Tämä ja muut ominaisuudet esiteltiin Fanuc:n esitteessä (liite 4). Datan tallennus tuli varsinkin vikatilanteissa erittäin hyödylliseksi, koska tämän avulla päästiin käsiksi vikojen juurisyihin ja pystyttiin tarkasti havaitsemaan, onko voima pysynyt asetetuissa arvoissa.

3 OHJELMOINTI

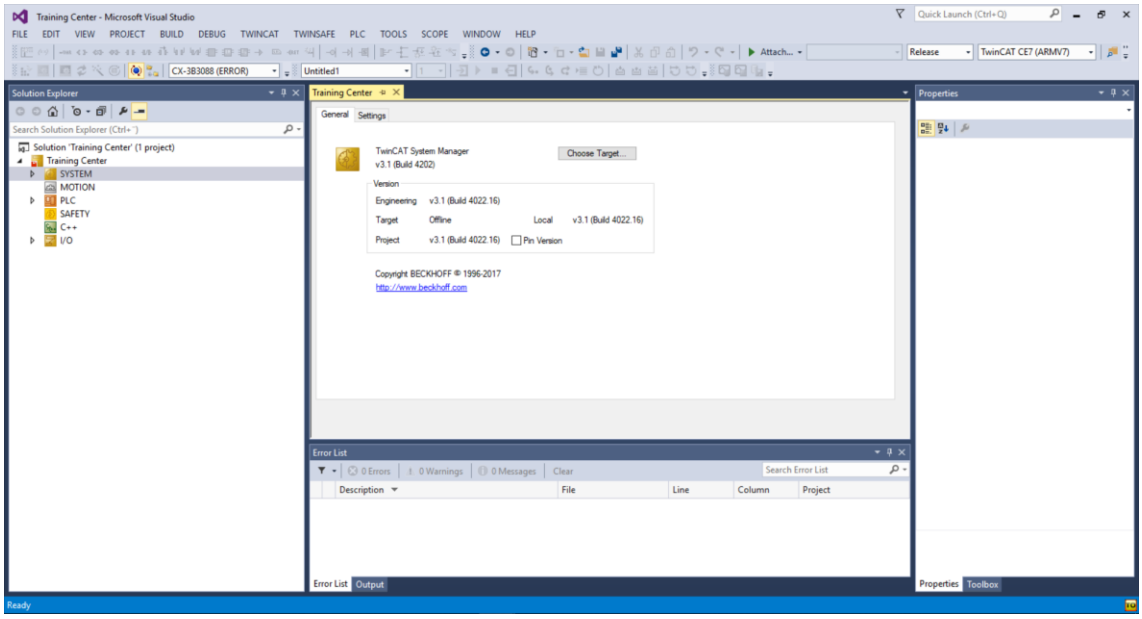
Tässä kappaleessa käsitellään asioita ja toimenpiteitä, joita tehtiin opinnäytetyön aikana PLC:n käyttöönotossa sekä robotin ja PLC:n kättelyn muodostamisessa. Alkuperäisestä suunnitelmasta poiketen opinnäytetyö ei käsittänyt automatisoinnin osalta valmiin ohjelman luomista PLC:lle eikä kaikkien laitteiden kytkentää ja käyttöönottamista. Ohjelmoinnissa kuitenkin otettiin huomioon työn jälkeiset toimenpiteet jättämällä varauksia laitteille ohjelmaan sekä suunnittelemalla ohjelman rakennetta tulevia töitä varten. Kaikkia ohjelman osia ei ole esitetty opinnäytetyössä, vaan ohjelmasta on esitetty osia havainnollistamaan tehtyä työtä.

Ohjelmoinnissa käytettiin TwinCAT3 ohjelmistoa, joka on Microsoft Visual Studioon pohjautuva ohjelmointiympäristö. Ohjelma tehtiin solussa käytetylle Beckhoff:n PLC:lle, joka on esitelty aikaisemmin kappaleessa 2.1.1 PLC.

3.1 PLC käyttöönotto

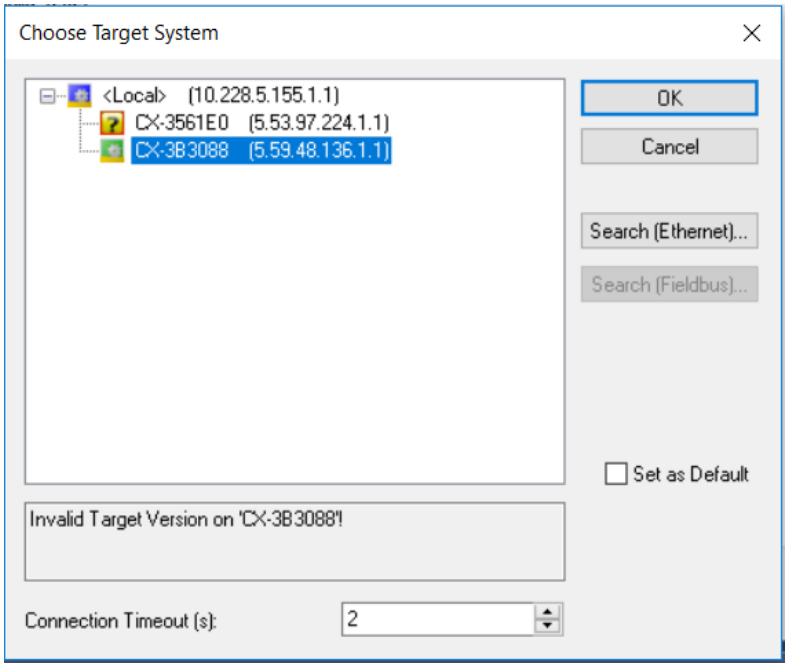
Logiikan käyttöönotto aloitettiin kytkemällä näyttö sekä tietokoneen hiiri logiikan portteihin. Logiikan IP-osoite saatiin haettua näiden laitteiden avulla. Logiikka pyörii Windows 7 käyttöjärjestelmällä, joten näytön avulla päästiin käsiksi logiikan asetuksiin. Asetuksista voitiin logiikan IP-osoite muuttaa halutuksi. Tässä tapauksessa logiikan IP-osoitteeksi muutettiin 192.168.0.101. Tässä vaiheessa kaikki muutkin laitteet olivat samassa IP-avaruudessa, joten logiikan IP-osoite oli loogista asettaa samaan avaruuteen.

Seuraavassa vaiheessa käyttöönotto aloitettiin käyttäen TwinCAT3 ohjelmistoa. Aluksi luotiin uusi projekti kyseiselle työlle. Projekti luotiin muotoon TwinCAT Projekti. Kun ohjelmalla luotiin täysin uusi projekti, se oli kuvan 4 mukaisesti täysin tyhjä. Tässä vaiheessa logiikka tai logiikkaan EtherCat-väylällä kytketyt laitteet eivät vielä näkyneet projektilla. Jotta laitteet saatiin käyttöön ohjelmassa, piti ensimmäisenä hakea käytössä oleva PLC ohjelman käyttöön.



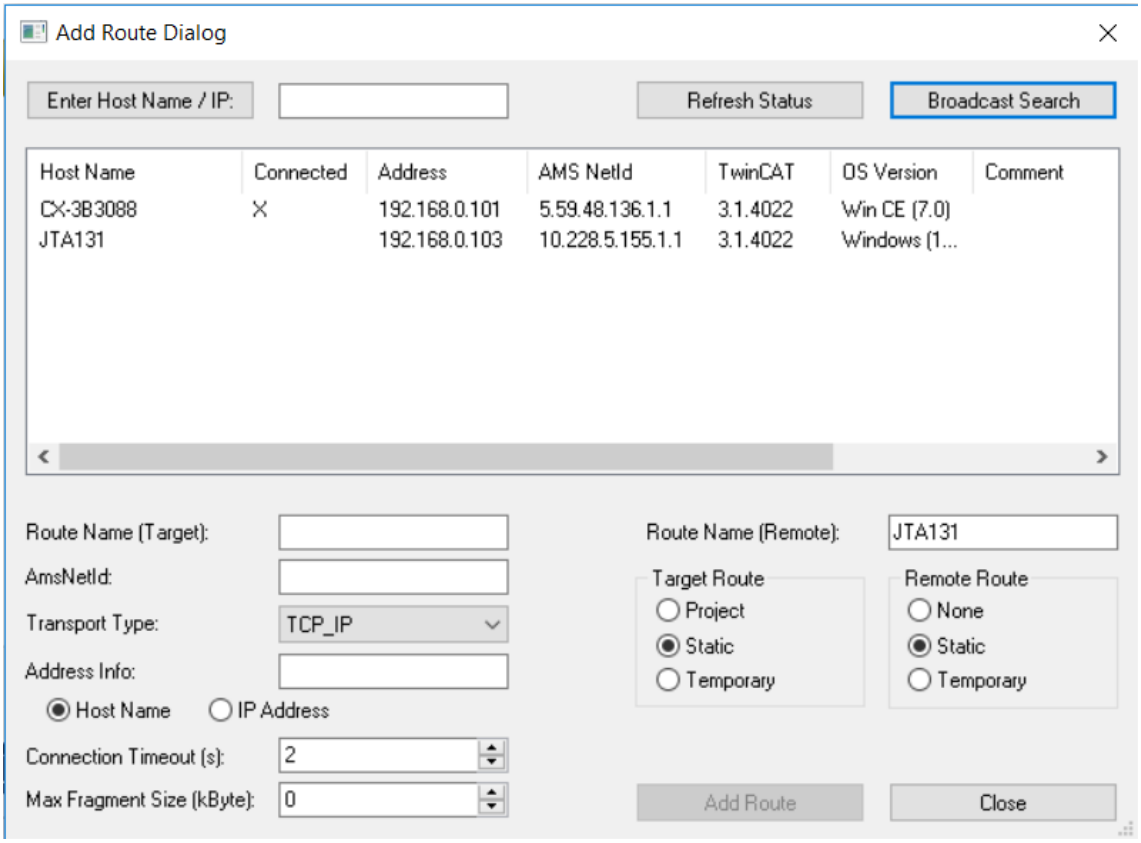
KUVA 4. TwinCAT3 tyhjä projekti

Kohdelaitteen haku tapahtui ohjelman välilehdeltä SYSTEM. Tältä välilehdeltä voitiin valita toiminto ”Choose target”. Kun tämä toiminto oli valittu avautui kuvan 5 mukainen ikkuna, jossa oli listattuna kaikki mahdolliset käytössä olevat laitteet. Tässä vaiheessa ei kuitenkaan ollut kuvan mukaisesti vielä mahdollista ottaa yhteyttä laitteeseen ilman, että laitetta etsittiin verkosta. Laitteen hakua varten valittiin ikkunasta Search (Ethernet).



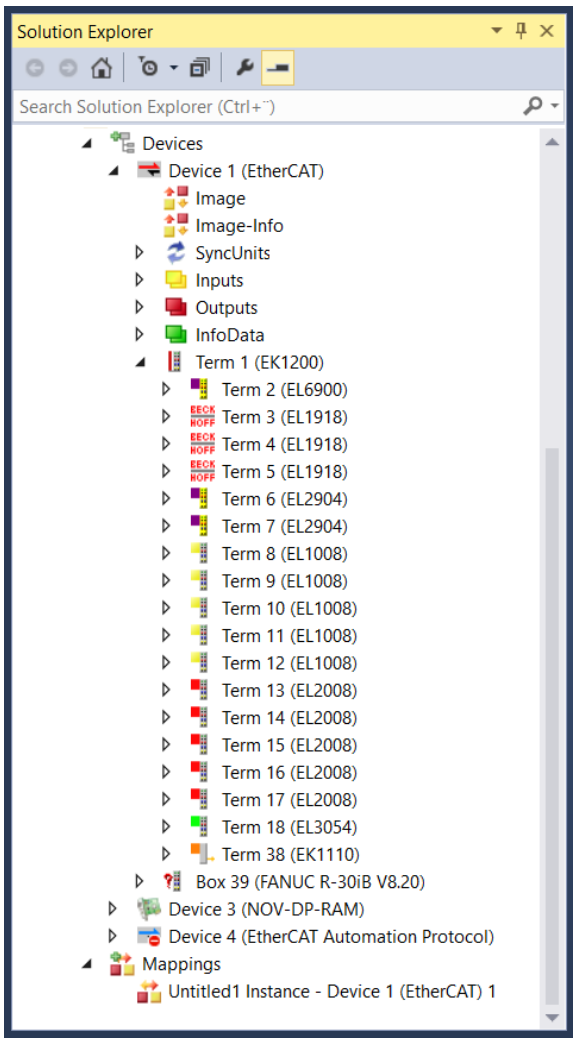
KUVA 5. Choose target system

Kun ohjelmassa oltiin valittu laitteen haku verkosta, päästiin kuvan 6 mukaiselle ikkunalle, josta voitiin suorittaa logiikan haku. Tästä ikkunasta pystyttiin skannaaman koko tietokoneen verkko tai hakemaan tietyn IP-osoitteen omaavaa laitetta. Koska IP-osoite kohteelle oli asetettu jo työn aikaisemmassa vaiheessa, voitiin laitteen IP-osoitetta käyttää laitteen haussa. Kohdelaitteen IP-osoitteeksi oli asetettu 192.168.0.101. Tällä osoitteella laite löydettiin verkosta. Kokeilun kannalta laiteen haku suoritettiin myös kuvan 6 mukaisesti broadcast search-toiminnolla, joka haki jokaisen laitteen verkosta, johon tietokone oli yhdistettynä. Toiminnolla löydettiin myös PLC, jota ohjelman oli tarkoitus etsiä. Kun oikea laite oli löydetty, valittiin se ikkunasta aktiiviseksi ja valittiin lisää reitti.



KUVA 6. Add route dialog

Kun laitteiden välille oltiin luotu reitti, voitiin aktiivinen valinta nähdä edellisessä ikkunassa, jossa kuvan 5 mukaisesti reitti on esitetty olevan aktiivinen. Laitteen määrityksen jälkeen voitiin projektin pääsivulla havaita, että sivun puukaavioon oli ilmestynyt laitteet, jotka järjestelmään oli kytketty. Laitteet olivat esitettynä kuvan 7 osoittamalla tavalla, jossa laite 1 oli PLC, jonka alta löytyivät kaikki järjestelmään kytketyt logiikkakortit sekä robotti, jonka tulojen ja lähtöjen määritys on käsitelty kappaleessa 4.2 I/O-mapping. Laitteet 3 ja 4 olivat logiikan omia sisäisiä laitteita, joita ei tarkemmin käsitelty tässä työssä.



KUVA 7. Lisätyt laitteet

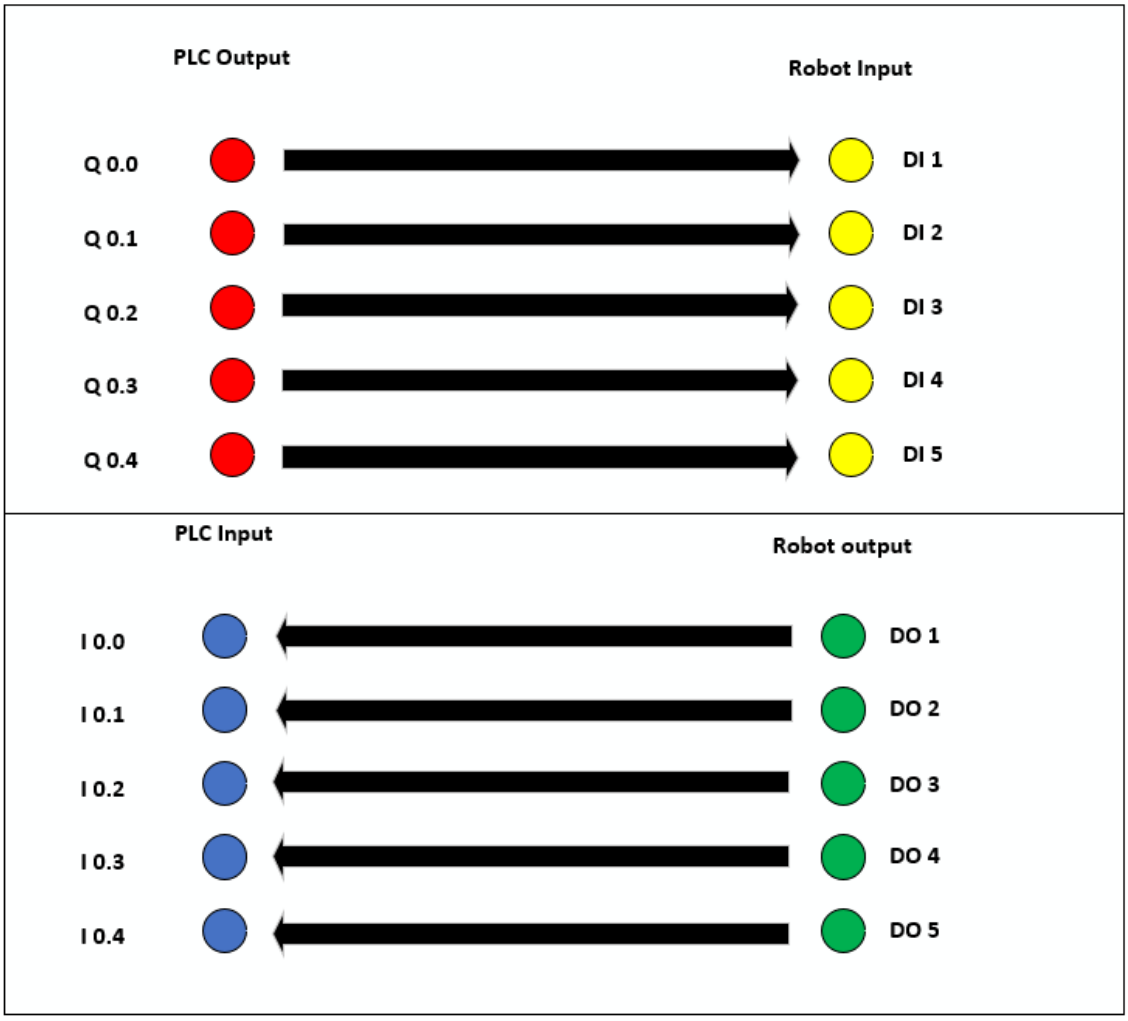
Laitteen 1 kortit saatiin näkyville skannaamalla laite. Logiikan kortit eivät olleet oletuksena näkyvillä ohjelmassa ennen skannausta. Laitteen skannaus tapahtui laitteen 1 päällä painettaessa hiiren oikeaa näppäintä, jolloin avautui valikko, josta voitiin valita skannaus suoritettavaksi. Skannauksen jälkeen kaikki PLC:n kortit sekä siihen yhdistetty robotti saatiin näkymään ohjelmassa. Tämän vaiheen jälkeen ohjelma oli valmis robotin ja PLC:n I/O-mapping:a varten.

Laitteiden yhdistämisen aikana huomattiin, että lisättävät laitteet tulee aina kytkeä erilliseen EtherCat-väylään, eikä tietokoneen ja PLC:n väliseen käyttöön tarkoitettuun porttiin. Esimerkiksi tässä projektissa ongelmaksi muodostui se, että PLC ei tunnistanut Fanucin robottia, joka oli kytketty PLC:n kytkimen kautta järjestelmään, vaan logiikkaan piti lisätä erillinen EK1110 -kortti, joka toimi EtherCat-väylän jakajana. Kun oltiin päästy vaiheeseen, jossa kaikki laitteet oli skannattu ja laitteet olivat lisätty järjestelmään onnis-

tuneesti, laitteet näyttivät niihin kytketyt laitteet sekä kortit järjestelmässä. Vaikka kuvassa 7 robotti näkyi olevan kytkettynä laitteeseen 1, se oli kuitenkin erillinen laite, joka oli kytketty EK1110-kortin välityksellä järjestelmään käyttäen EtherCat-väylää.

3.2 Robotin ja PLC:n I/O-mapping

Koska robotin ja PLC:n välillä ei ole muuta fyysistä yhteyttä kuin EtherCat-väylä, piti laitteille suorittaa tulojen ja lähtöjen kartoitus eli I/O-mapping. Tässä toimenpiteessä laitteiden digitaalisille tuloille ja lähdöille määriteltiin vastaavat parit. Eli käytännössä määriteltiin, miten vastakkainen puoli reagoi, jos esimerkiksi PLC:n asettaa lähtönsä Q0.1 aktiiviseksi. Tämä määrittely on vapaasti tehtävissä ja robotin ja PLC:n tulojen ja lähtöjen alueet voidaan valita vapaasti muiden rajoitteiden, kuten PLC:n jo olemassa olevien osoitteiden, puitteissa.



KUVA 8. Robotin ja PLC:n tulojen ja lähtöjen havainnollistaminen

Kuvassa 8 on esitetty havainnollistava kuva, jossa on esitetty, miten PLC:n ja robotin välinen kommunikointi EtherCat-väylää käyttäen voidaan toteuttaa. Tässä kuvassa PLC:n lähdöt Q0.0–0.4 on yhdistetty eli kartoitettu robotin digitaalisiin tuloihin DI 1-5 ja vastaavalla tavalla robotin lähdöt on yhdistetty PLC:n tuloihin. Tämä on kahden laitteen välisen I/O-mapping:n perusperiaate. Robotin ja PLC:n tulojen ja lähtöjen määrityksessä tulee aina muistaa, että robotin tulot ovat aina PLC:n lähtöjä ja Robotin lähdöt ovat aina PLC:n tuloja.

Robotin ja PLC:n I/O-mapping aloitettiin päättämällä, kuinka monta bittiä järjestelmässä tarvittiin. Koska järjestelmässä käskytys logiikalta robotille oli tarkoitus toteuttaa ohjelmanumerolla ja robotille tarvittiin useita muita parametrejä, jotka olivat 16 ja 32 bittisiä, valittiin, että järjestelmään otetaan käyttöön 512 bittiä eli 64 tavua. Alustava ohjelma tehtiin kuitenkin niin, että järjestelmään otettiin aluksi 16 bittiä käyttöön, jotta voitiin kokeellisesti testata järjestelmän toimivuus.

Tulot ja lähdöt tuli ohjelmassa määritellä ja nimetä, jotta niitä pystyttäisiin ohjelmassa jatkossa kutsumaan ja käyttämään. Ohjelman osat määriteltiin eri struktuureihin niiden jatko käyttöä varten. Ohjelman osuudet oli helpompi jakaa eri osiin, jotta niiden muokkaaminen ja laajentaminen helpottui tarvittaessa.

Ensin luotiin kuvan 9 mukainen struktuuri, jossa määriteltiin kättelyssä käytetty tavu olemaan tavu välillä 1-2. Tämä määritys toteuttaa tavun juoksevan numeroinnin, kun määritettyä muuttujaa *byte_num* käytetään. Array määritetään sisältävän siis arvot yhdestä kahteen. Nämä arvot määritetään kuvan lausekkeessa olemaan tavuja, joten määritetyllä array:lla saada arvot 1.0 – 2.8.

```
STRUCT
    byte_num : ARRAY [1..2] OF BYTE;
END_STRUCT
```

KUVA 9. Robot link byte struct

Robotille lähtevien signaaleiden tyyppi sekä nimi pitää myös määritellä. Robotille lähtevä signaali voi olla muodossa boolean, byte, word tai double word. Robotille lähtevän ensimmäisen tavun bitit määriteltiin kuvan 10 mukaisesti luoden niistä yksi struktuuri.

Kaikki robotille lähtevät signaalit pidettiin tässä vaiheessa bitteinä eli niiden tyypiksi määriteltiin boolean, jota ohjelmassa käytetään tyypillä BOOL. Jotkin signaaleista myös nimettiin, jotta niitä olisi helppo kutsua luodussa ohjelmassa.

```
(*Byte 1*)
start_req                                   : BOOL;     (**)
stop_req                                    : BOOL;     (**)
reset_req                                   : BOOL;     (**)
abort_req                                   : BOOL;     (**)
home_req                                    : BOOL;     (**)
RDI6                                        : BOOL;     (**)
RDI7                                        : BOOL;     (**)
RDI8                                        : BOOL;     (**)
```

KUVA 10. PLC to Robot out struct

Struktuurilla, jossa määriteltiin muuttuja *robotti*, oli ainoana tarkoituksena helpottaa lopullisen ohjelman ymmärrettävyyttä ja luettavuutta. Kuvassa 11 on esitetty struktuuri, jossa muuttuja määriteltiin.

```
STRUCT

    (* Robotit *)
    robot                : PLC_to_Robot_out_struct;

END_STRUCT
```

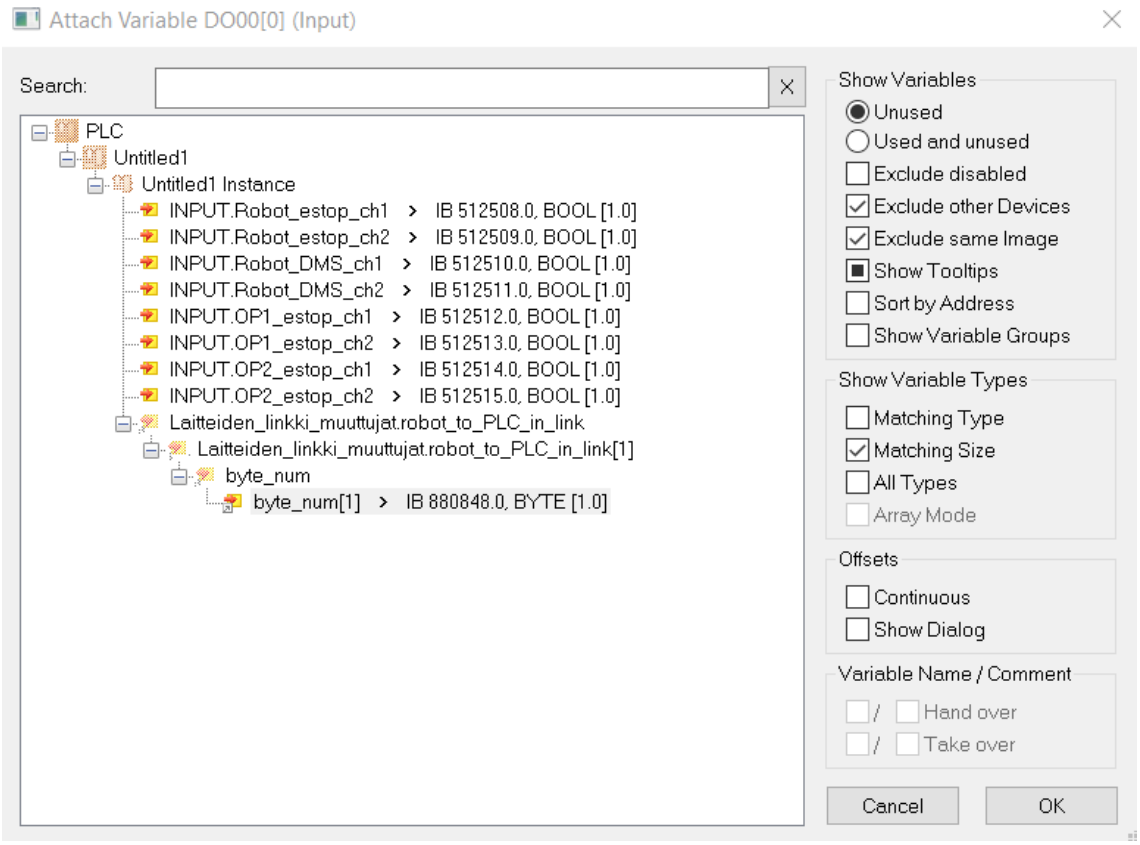
KUVA 11. output structure

Varsinaisessa ohjelmassa määriteltiin PLC:lle vastaava signaali robotilta. Ohjelma toteutettiin kahdelle tavulle. Tässä ohjelmassa voitiin kuvan 12 mukaisesti nähdä, että aiemmat määrittelyt lyhensivät ohjelman koodia huomattavasti ja että täten ohjelma pysyi ymmärrettävässä muodossa. Ohjelman oikean puolen muuttujat määriteltiin vastaamaan ohjelman koodia tarkoittavaa tuloa tai lähtöä ja ohjelmassa oli myös varaus sille, että järjestelmässä olisi useampi kuin yksi robotti. Tätä osuutta ohjelmasta ei esitetä opinnäytetyössä, koska se oli täysin yrityksen toisen edustajan kirjoittama.

```
(*Byte 1*)
PLC_to_robot_out_link[1].byte_num[1].0          := dout.robot.start_req          ;(**)
PLC_to_robot_out_link[1].byte_num[1].1          := dout.robot.stop_req           ;(**)
PLC_to_robot_out_link[1].byte_num[1].2          := dout.robot.reset_req          ;(**)
PLC_to_robot_out_link[1].byte_num[1].3          := dout.robot.abort_req          ;(**)
PLC_to_robot_out_link[1].byte_num[1].4          := dout.robot.home_req           ;(**)
PLC_to_robot_out_link[1].byte_num[1].5          := dout.robot.RDI6               ;(**)
PLC_to_robot_out_link[1].byte_num[1].6          := dout.robot.RDI7               ;(**)
PLC_to_robot_out_link[1].byte_num[1].7          := dout.robot.RDI8               ;(**)
```

KUVA 12. Link outputs to robot

Ohjelman kirjoituksen jälkeen se voitiin suorittaa onnistuneesti. Ohjelman suorituksen jälkeen robotin ja PLC:n I/O-mapping oli suoritettu. Avattaessa robotin digitaaliset lähdöt ja niiden linkitys ohjelmasta huomattiin, että robotin lähtö oli linkitetty PLC:n tuloon kuvan 13 mukaisesti.



KUVA 13. I/O-mapping

Robotin ja PLC:n signaaleiden määrityksessä tehtiin jatkomahdollisuudet helpoksi toteuttaa, koska 16 bittiä nostettaisiin jossakin työn vaiheessa 512 bittiin. Tässä työssä ei kuitenkaan enää tämän määrityksen jälkeen muutettu laitteiden kättelyä ja bittien määrä jätettiin määritettyyn 16 bittiin.

3.3 Muuttujat

Ohjelmoinnissa käytetyt muuttujat suunniteltiin järjestelmän tarpeiden mukaisesti. Järjestelmän muuttujat jaoteltiin eri kategorioihin, joita käytettäisiin ohjelman eri osioissa. Koska ohjelmaa järjestelmälle ei opinnäytetyössä tehty, oli muuttujien lista alustava. Näiden muuttujien avulla voitaisiin kuitenkin tehdä ohjelmalle perusrakenne, jota voitaisiin hyödyntää varsinaisessa ohjelmassa. Ohjelmaan kirjoitettiin muuttujat ja esimerkiksi robotille lähtevät muuttujat kirjoitettiin valmiiksi ohjelmaan I/O-mapping:ssa. Taulukossa 3 esitettiin jokaiselle ohjelmassa käytetylle muuttujalle sen nimi, muuttujan tyyppi sekä kommentti muuttujan toiminnosta ohjelmassa.

TAULUKKO 1. Muutujalista

	Muuttuja	**Tyyppi**	**Kommentti**
Safety	Robot_estop_ch1	Boolean	Robotin kontrollerin hätä-seis-painike kanava 1
	Robot_estop_ch2	Boolean	Robotin kontrollerin hätä-seis-painike kanava 2
	Robot_DMS_ch1	Boolean	Robotin ohjaimen hätä-seis-painike kanava 1
	Robot_DMS_ch2	Boolean	Robotin ohjaimen hätä-seis-painike kanava 2
	OP1_estop_ch1	Boolean	Näyttöpaneelin hätä-seis-painike kanava 1
	OP1_estop_ch2	Boolean	Näyttöpaneelin hätä-seis-painike kanava 2
	OP2_estop_ch1	Boolean	Painikekotelon hätä-seis-painike kanava 1
	OP2_estop_ch2	Boolean	Painikekotelon hätä-seis-painike kanava 2
Inputs	OP1_Reset	Boolean	Näyttöpaneelin reset painike
	OP1_Start	Boolean	Näyttöpaneelin start painike
	OP1_Stop	Boolean	Näyttöpaneelin stop painike
	OP1_Manual	Boolean	Näyttöpaneelin avainkytkin manual
	OP1_Auto	Boolean	Näyttöpaneelin avainkytkin automaatti
	OP2_Reset	Boolean	Painikekotelon reset painike
	OP2_Start	Boolean	Painikekotelon start painike
	OP2_Stop	Boolean	Painikekotelon stop painike
	OP2_Manual	Boolean	Painikekotelon avainkytkin manual
	OP2_Auto	Boolean	Painikekotelon avainkytkin automaatti
Errors	Error_num	Integer	PLC:n virherumero
	Error_num_robot	Integer	Robotin virhenumero PLC:lle
Käyttöliittymä	UI_New_Model_Req	Boolean	Uuden kappaleen tietojen kirjoitus
	UI_Model_ID	Integer	Kappaleen tunnus
	UI_Path_Cnt	Byte	Työstöratojen määrä
	UI_Height	Byte	Työstöradan korkeus
	UI_Kuvaus	Boolean	Kuvaus käsky
	UI_PRG_Num	Integer	Tehtävänumero

Robot	To_Robot_Start_Req	Boolean	Käynnistyspyyntö
	To_Robot_Stop_Req	Boolean	Pysäytyspyyntö
	To_Robot_Abort	Boolean	Ohjelman keskeytys

4 ROBOTIN KÄYTTÖÖNOTTO

Tässä kappaleessa käsitellään robotin käyttöönottoa. Robotin käyttöönottoon kuului työkalujen määritys, työkalun kuorman määritys, robotin tulojen ja lähtöjen määritys sekä ohjelmisto-optioiden käyttöönotto.

4.1 Työkalut

Robotilla käytettiin jäysteenpoistoon sopivia koneviiloja. Koneviilat olivat paineilmatoimisia. Koneviiloja oli kahta eri kokoa, mutta niissä kaikissa oli 8 mm kara. Tähän karaan tehtiin koeajoja varten PLA:sta 3D tulostettuja tappeja, joilla voima-anturin ja kameran toimintaa voitiin kokeilla. Yhden koneviilan kara oli niin sanotusti kelluva eli sen jäykkyyttä pystyttiin säätämään paineilman avulla. Voima-anturi tarvitsi kuitenkin jäykän työkalun toimiakseen, joten tästä työkalusta jouduttiin luopumaan heti työn alussa. Työkalujen vaihdossa käytettiin Schunkin valmistamaa työkalunvaihtajaa.

4.1.1 Kuorman määritys

Robotille määriteltiin kuorma, kun sen työkalulaippaan ei oltu asennettu vielä työkaluvaihtajaa tai mitään käytetyistä työkaluista. Kuorma siis määritettiin robotin työkalulaipalle, jonka piti olla nolla. Tällä saatiin robotin akseleiden momentit niin sanotusti kalibroitua, koska määritys tehtiin ilman massaa.

Kuorman määritys suoritettiin järjestelmävalikosta, josta valittiin liike. Tästä polusta päästiin ikkunaan, jossa kaikki käytetyn robotin kuormat oli määritelty. Oletuksena robotille voitiin määritellä maksimissaan kymmenen eri kuormaa. Kuorman määritys tulee suorittaa kaikille työkaluille sekä tilanteissa, joissa robotti käsittelee kappaleita myös kappaleiden tuottama kuorma pitää määritellä.

Kuorman määritystä ensimmäistä kertaa tehdessä määrityksessä pitää ottaa käyttöön kalibrointitila. Tämä toiminto on oletuksena kytketty pois päältä. Jos kalibrointia ei ole tehty käytetyllä robotilla, kalibroinnin tila on tekemätön. Jos kalibrointi on suoritettu, kalibroinnin tila on tehty. Ennen kalibroinnin suorittamista voitiin kuorma määritellä, jos

kuorma oli tiedossa. Koska kuormaa robotilla ei ollut, voitiin valita kuorman olevan tiedossa ja asettaa sen olleen 0 kilogrammaa. Kun kalibrointi kytkettiin päälle ja valittiin kalibroinnin suoritus, robotti lähti suorittamaan tehtävää. Tässä tehtävässä robotti liikkuu kahden pisteen välillä. Muiden robotin akseleiden ollessa 0 asteen kulmassa, akselit 5 ja 6 akselit lähtivät ensin kulmasta -90 astetta. Tämän jälkeen robotti käänsi nämä kaksi akselia hitaasti kulmaan 90 astetta. Robotti palasi takaisin lähtöasemaan, jonka jälkeen robotti suoritti samat liikkeet nostaen nopeutensa huippunopeuteen. Näillä liikkeillä robotti pystyi laskemaan akselivirroista, paljonko robotin kuorma oli.

4.1.2 Työkalupisteet

Robotin työkalupisteet määriteltiin kunkin käytetyn työkalun mukaan. Robotille piti asettaa työkalupisteet jokaiselle koneviilalle niin, että koordinaatistojen Z-akseli oli vastakkaissuuntainen työkalun kärjen kanssa. Viilojen kanssa käytetylle testityökalulle eli muoviselle tapille luotiin oma työkalupiste kokeellisten ajojen ajaksi. Työkaluvaihtajakin tarvitsi oman työkalupisteen, koska sitä käytettiin vaihtaessa työkalua. Työkaluvaihtajan työkalukoordinaatisto sijaitsi vaihtajan keskiosassa koordinaatiston Z-akselin osoittaessa robotin työkalulaipasta vaihtajan lävitse.

Voima-anturin konfiguraatiota varten voima-anturi tarvitsi työkalukoordinaatiston voima-anturin pintaan, johon työkalukin lopulta kiinnitettiin. Voima-anturin koordinaatisto asetettiin parametrisesti, koska 3D mallinnettu sovite robotin työkalulaipan ja voima-anturin välissä ei ollut valmistajan standardikappale. Standardisovitteella voima-anturin oletuskoordinaatisto olisi ollut 63 mm päässä robotin työkalulaipasta, mutta itse suunnitellulla sovitteella työkalupiste asettui 67 mm päähän. Tämä koordinaatisto oli ehdottoman tärkeä voima-anturin toiminnan kannalta ja se tuli asettaa niin, että koordinaatiston Z-akseli osoitti robotin työkalulaipan suuntaan. Oletustyökalukoordinaatisto, jonka voima-anturi itse loi järjestelmään, on 180 astetta väärässä kulmassa käännettäessä sitä Y-akselin ympäri.

4.2 I/O-mapping

Termillä I/O-mapping tarkoitetaan sitä, kun kahden eri järjestelmän digitaaliset ja analogiset tulot ja lähdöt kartoitetaan vastaamaan toistensa signaaleja. Esimerkiksi, jos robotin

digitaalinen lähtö Q0.1 asetetaan robotilla tilaan yksi, voidaan määrittää, että tässä tapauksessa PLC:n tulo I0.1 tulee aktiiviseksi. Tämä onnistuu käyttämällä laitteiden välillä jotakin väylää. Opinnäytetyön järjestelmässä käytettiin EtherCAT-väylätekniikkaa.

4.2.1 Digitaalisten tulojen ja -lähtöjen määritys

Robotilla digitaalisten tulojen ja lähtöjen määrittäminen tehtiin päävalikon osiosta I/O, josta valittiin ryhmä Digital. Tässä ikkunassa on mahdollista monitoroida robotin tuloja ja lähtöjä sekä asettaa lähtöportteja päälle tai pois.

TAULUKKO 2. Tulojen ja lähtöjen määritys

#	RANGE		RACK	SLOT	START	STAT.
1	DO	[1 - 492]	106	1	1	ACTIV
2	DO	[493 - 512]	0	0	0	UNASG

#	RANGE		RACK	SLOT	START	STAT.
1	DI	[1 - 512]	106	1	1	ACTIV

Robotin tulojen ja lähtöjen määrä määriteltiin siten, että PLC:n kanssa kommunikointiin jätettiin runsaasti tuloja ja lähtöjä varalle. Jos järjestelmien kommunikoinnissa tarvitaan esimerkiksi analogiasignaaleja ne vievät yleensä aina 16-32 bittiä järjestelmästä. Taulukossa 1 on esitetty kopio Fanuc robotin järjestelmän taulukosta, jossa tulojen ja lähtöjen tiedot on esitetty. Kaikki tulojen ja lähtöjen parametrit olivat muokattavissa. Tulojen ja lähtöjen alue on ilmoitettu taulukon toisessa sarakkeessa. Alueita voidaan asettaa monia, jos järjestelmässä on käytössä useita laitteita, joiden välistä kommunikaatiota halutaan helpottaa asettamalla tulot ja lähdöt omiin osioihinsa. Työssä valittiin käytettäväksi 512 digitaalista tuloa ja lähtöä. Tämä määrä oli riittävä, jotta kaikki tarvittava tieto saatiin siirrettyä laitteiden välillä.

Rack-sarake ilmaisee kyseisen rivin käyttämän tiedonsiirtotekniikan. Rack 106 on parametri, joka ilmaisee järjestelmän käyttävän EtherCAT-väylää laitteiden välillä. Tämän parametrin avulla robotti ymmärtää, kuinka se siirtää ja käsittelee lähetetyt ja vastaanotetut signaalit. Start-sarake määrittelee mistä tulosta tai lähdöstä järjestelmä alkaa määrittelemään asetettua aluetta. Taulukon statussarake ilmaisee, onko alue aktiivinen. Kun järjestelmän I/O määrittely tehtiin, robotin kontrolleri piti käynnistää uudestaan, jotta asetetut parametrit astuivat voimaan.

4.2.2 Ryhmien määritys

Robotin ryhmät koostuvat digitaalisista tuloista ja lähdöistä sen perusteella, onko kyseinen ryhmä tuleva vai lähtevä. Ryhmä tarkoittaa lähtöjen tai tulojen muodostamaa kokonaisuutta, jonka avulla voidaan siirtää dataa tavuina, sanoina tai kaksoissanoina. Fanucin robottien ja PLC:n välillä ei voi suoraan siirtää analogista dataa digitaalisten tulojen ja lähtöjen avulla vaan näistä analogisen signaalin arvoista pitää muodostaa ensin ryhmä, jota signaali käyttää. Ryhmän koko riippuu siitä, mitä sen avulla halutaan siirtää. Jos logiikalta halutaan siirtää robotille integer muotoinen arvo, se pitää ensin muuttaa sanaksi. Sana koostuu 16:sta bitistä eli 16:sta digitaalisesta tulosta tai lähdöstä. Tällainen ryhmän määrittely voidaan tehdä esimerkiksi taulukon 2 ensimmäisen rivin mukaisella määrittelyllä, jossa digitaalisista tuloista 1-16 on luotu yksi EtherCAT-väylää käyttävä ryhmä. Työssä luotiin kolmelle PLC:ltä tulevalle sanalle ryhmät, joita käytettiin laitteiden välisessä tiedonsiirrossa. Ryhmät luotiin robotin digitaalisille tuloille 1-48, joista saatiin koostettua kolme 16 bittiä sisältävää ryhmää.

TAULUKKO 3. Tuloryhmä

GI #	RACK	SLOT	START PT	NUM PTS
1	106	1	1	16
2	106	1	17	16
3	106	1	33	16
4	0	0	0	0
5	0	0	0	0

Koska työssä PLC:n oli tarkoitus antaa robotille tehtävä ja olla näin robotin masteri, robotti sai ensimmäistä ryhmää käyttäen PLC:ltä tehtävänumeron, jonka mukaan se suoritti asetetun tehtävän. Nämä arvot pystyttiin luomaan siten, että ne sisälsivät robotin tehtävän sekä tietoa kappaleesta, jota työsuoritus koski. Kappaleen parametrit ja muut työkiertoon liittyvät tiedot tuotiin robotille käyttäen näitä ryhmiä. Esimerkiksi kappaleen työstöradan korkeus tuotiin robotille käyttäen luotuja ryhmiä.

4.3 Ohjelmisto-optiot

Fanuc-robottien ohjelmalliset ominaisuudet vaativat ohjelmisto-option toimiakseen. Optiolla tarkoitetaan ohjelmallista ominaisuutta, joka ei kuulu perusvarustellun robotin oletusohjelmistoihin.

Työn alussa tutkittiin, mitä työmenetelmiä voima-anturin osalta työssä voitaisiin käyttää. Voima-anturin perusohjelmisto on optio, joka voima-anturin yhteydessä oltiin yrityksen puolesta tilattu. Jäysteen poistoon tarvittiin kuitenkin contouring-funktiota, jota käyttämällä työ voitiin suorittaa. Monimutkaisissa kappaleissa contouring-funktion työntösuunnan automaattinen kääntö oli ominaisuus, jota tarvittiin. Tämä ominaisuus sisältyi deburring-optioon, johon myös contouring-funktio sisältyi. Samassa optiossa oli myös ominaisuus deburring path auto generation ja tämän toiminnon käytettävyyttä sovelluksessa on tarkoitus tutkia opinnäytetyön jälkeen.

Konenäkökameran ja robotin muut optiot tilattiin niiden tarpeen mukaan. Näiden optioiden tarpeen määritteli konenäköosuuden tekijä sekä yrityksen edustaja.

5 VOIMA-ANTURIN KÄYTTÖÖNOTTO

Tässä kappaleessa käsitellään voima-anturin käyttöönottoa ja siihen liittyviä seikkoja kuten esimerkiksi asennusta ja konfigurointia. Asiat käsitellään siinä järjestyksessä, jossa työ on suoritettu. Kappale käsittelee myös käyttöönoton jälkeisten funktioiden käsittelyt ja testit, joissa funktioiden toimintaa on havainnollistettu sekä mahdolliset vikatilanteet ja niiden selvitykset.

5.1 Asennus

Voima-anturi asennettiin robotin työkalulaippaan, johon tarvittiin sovitekappale. Sovitekappale robotin työkalulaipan ja voima-anturin välille oli suunniteltu ja koneistettu ottaen huomioon robotin tyyppi sekä voima-anturin malli. Fanucin robottiin M-20iB/25 ja voima-anturin malliin FS-40 olisi ollut mahdollista tilata laitevalmistajan valmistama sovite, mutta toimitusaikojen sekä kustannusten puitteissa oli järkevämpää koneistuttaa sovitekappale.

Voima-anturin työkalun puoleiseen laippaan koneistettiin myös sovite, jonka avulla voima-anturiin voitiin kiinnittää Schunkin valmistama työkalunvaihtaja. Tämä sovite asennettiin kiinni vasta voima-anturin käyttöönoton ja konfiguroinnin jälkeen, jotta anturin antaman voiman virhemarginaali olisi mahdollisimman pieni käyttöönoton aikana tehdyssä kalibroinnissa.

Asennuksessa piti liitoksista tehdä välyksetön. Anturin toiminnan kannalta oli tärkeää, että työkalu ei päässyt liikkumaan työkierron aikana, koska tämä vaikutti radikaalisti testeissä tutkittuihin tuloksiin ja anturin käyttäytymiseen työn aikana. Koska robotin työkalulaipan ja robotin työkalun välillä oli kaikkien komponenttien jälkeen kuusi liitoskohtaa, antoi työkalu myötää suurilla voimilla. Tämä ei kuitenkaan oikean kiinnitystavan ansiosta aiheuttanut suurta virhettä.

Sähköisessä kytkennässä anturilla käytettiin valmistajan toimittamaa kaapelia. Kaapeli kytkettiin suoraan anturin ja robotin käsivarressa sijaitsevan liittimen välille. Kaapeli saatiin vedettyä robotin 5-akselin takaa, josta kaapelin pituuden vuoksi se saatiin optimaalisesti vietyä 4-akselin sivussa sijaitsevalle liittimelle.

Valmistajan asennusohjeiden mukaisesti anturinkaapeli pitäisi anturin juuresta suojata vaahtomuovilla tai muulla vastaavalla materiaalilla, jotta kaapelin värinä robotin liikkuessa ei aiheuttaisi virhettä anturissa. Tämä toimenpide jätettiin kuitenkin tekemättä, koska tämän tyyppiselle kiinnitykselle ei olisi ollut sopivaa kiinnityspaikkaa ja robotin liikkeet olisivat voineet lisätä virhettä. Testeissä huomattiin, että kaapelin suojaamisella ei ollut vaikutusta voima-anturin toimintaan.

5.2 Käyttöönotto ja konfigurointi

Käyttöönotossa ja konfiguroinnissa keskitytään voima-anturin ohjelmistojen asennukseen sekä avustetun asennuksen eri vaiheisiin. Funktioita sekä niiden käyttäytymistä tutkittiin niiden toiminnallisuuksien tarpeen mukaan ja kaikki funktiot on kuvattu kappaleessa ainakin pääpiirteittäin.

5.2.1 Optiot

Voima-anturin käyttöönottoa varten robotti tarvitsi anturille tarvittavat ohjelmisto-optiot. Toiminnan kannalta pakollisena optiona oli robotin omien optioiden lisäksi Force Control Basic, joka sallii voima-anturin käytön robotilla. Tämä optio avaa vain muutaman anturin funktion, joita on mahdollista käyttää robotin ohjelmassa. Tämän option lisäksi käyttötarkoituksen takia tarvittiin ohjelmisto-optiot Force control contouring, Force control fitting sekä Force control deburring package. Force control contouring ohjelmisto-optio avaa contouring funktion käyttöön. Tätä funktiota käytettiin robotin työssä. Muut edellä mainitut optiot avasivat kaikki muut voima-anturin funktiot käyttöön sekä kaksi ohjelmaa, joiden avulla olisi ollut mahdollista luoda työstöratoja voima-anturia käyttäen tai luoda työstöradat mallinnetusta kappaleesta. Tässä työssä käytettiin kuitenkin konenäköä tämän toiminnon suorittamiseen, joten näitä ohjelmistoja ei käsitellä.

Optioiden käyttöönottamiseksi Fanuc toimitti imagetiedoston robotista. Image on kuin robotin varmuuskopio, joka sisältää kaikki tiedot robotin kontrollerista. Näihin tiedostoihin valmistaja pystyy tilattaessa lisäämään tilattuja ohjelmisto-optioita. Image ei kuitenkaan sisällä robotille tehtyjä ohjelmia, työkalupisteitä tai muita koordinaatistoja, joten ennen saadun tiedoston lataamista piti robotista ottaa varmuuskopio. Näin saatiin säilytettyä jo aiemmin tehdyt työt.

Robotin varmuuskopio saatiin tallennettua muistitikulle liittämällä muistitikku robotin kontrollerissa sijaitsevaan USB-porttiin tai vaihtoehtoisesti käsiohjaimessa sijaitsevaan USB-porttiin. Tämän jälkeen tallennuslaite kytkettiin aktiiviseksi robotin file-valikosta. Kun tallennussijainti oltiin valittu, voitiin valita varmuuskopiovalikosta ”All of Above”. Tämä toiminto tekee varmuuskopion kaikista robotin muistiin tallennetuista tiedostoista kuten esimerkiksi rekisteriin tallennetuista arvoista, luoduista ohjelmista sekä koordinaatistoista. Varmuuskopio sisältää myös robotin sisäisiä asetuksia ja järjestelmämuuttujien arvoja. Varmuuskopioita luodessa ja niitä ladatessa robottiin on tärkeää muistaa ajaa kaikki robotin akselit nollaan asteeseen. Tämä on varotoimi, joka varmistaa, että robotin akselien kulmat pysyvät aina samana riippumatta siitä hukkaako robotti akselikulmiensa arvot varmuuskopiota ladattaessa. Varsinkin image -varmuuskopiota ladattaessa robotti saattaa hukata akselikulmansa. Jos näin käy, pitää robotti masteroida ja oikeat akselikulmat asettaa uudestaan.

Kun robotille ladataan image -varmuuskopio pitää robotilla avata boot monitor -valikko. Tämä valikko ei aukea normaaleiden valikoiden tavalla vaan robotin kontrolleri pitää käynnistää uudelleen. Kun robotin kontrolleri käynnistyy pitää ohjaimen F1 ja F5 painikkeita pitää pohjassa, kunnes boot monitor -valikko aukeaa. Tätä toimintoa voi verrata Windows-käyttöjärjestelmää käyttävien tietokoneiden BIOS-valikon avaamiseen, joka on toteutettu samalla tavalla kuin Fanucin boot monitor -valikko. Tästä valikosta voidaan kuitenkin valita kontrollerin palautus, jonka jälkeen valitaan, miten kontrollerin palautus halutaan tehdä. Koska optiot sijaitsevat valmistajalta saadussa imagessa, pitää suoritettavaksi toiminnoksi valita palauta kontrolleri imaget. Tämän jälkeen valitaan laite, josta lataus suoritetaan. Muistitikun, jossa image on tallennettuna, tulee kytkeä kontrollerin tai ohjaimen USB-porttiin. On myös suositeltavaa tallentaa tämän kaltaiset tiedostot puhtaalle muistitikulle eli muistitikulle, jossa ei asennushetkellä ole muita tiedostoja kuin tiedostot, jotka asennukseen kuuluvat.

Onnistuneen asennuksen jälkeen boot monitor-valikosta pitää suorittaa valvottu käynnistys. Valvotulla käynnistyksellä päästään käsiksi järjestelmän tiedostoihin ja muihin suojattuihin toimenpiteisiin tarvittaessa. Valvotun käynnistyksen jälkeen suoritetaan järjestelmän kylmäkäynnistys. Kylmäkäynnistys tarkoittaa järjestelmän uudelleenkäynnistystä niin, että sammuttaessa järjestelmän, mikään järjestelmän toiminto ei pysy käynnissä sammutushetkellä vaan järjestelmä käynnistää kaikki alijärjestelmänsä uudelleen. Tämä

toiminto vaaditaan järjestelmän palautuksen jälkeen, jotta kaikki asennetut tiedostot käynnistyvät, eikä taustalle jää vanhojen tiedostojen luomia prosesseja.

Näiden edellä mainittujen toimenpiteiden jälkeen ohjelmisto-optioiden avaamat funktiot ja muut toiminnot ovat valmiita käytettäviksi. Kaikki ohjelmat saadaan palautettua aiemmin tehdystä varmuuskopiosta, joka voidaan asentaa tiedostovalikosta valitsemalla palauta. Tavallisen varmuuskopion palautus ei vaadi järjestelmän kylmäkäynnistystä, mutta sen suorittaminen on suositeltavaa. Tämä vähentää järjestelmässä tapahtuvien virheiden mahdollisuutta.

5.2.2 Voima-anturin ohjelmiston käyttöönotto

Ennen voima-anturin ja sen funktioiden käyttöä piti tehdä voima-anturin asennusohjeen mukaisesti avustettu asennus, jossa määriteltiin anturin asennustapa, käyttäjäkoordinaatisto sekä muita anturille keskeisiä parametreja. Ennen avustettua asennusta piti asennettuna olla kaikki tarvittavat optiot, koska optioiden lisääminen pyyhkisi asennuksen aikana järjestelmään syötetyt parametrit ja asennus tulisi suorittaa uudelleen. Ennen asennusta myös kaikki voima-anturiin liittyvät komponentit tuli olla valmisteltuna asennukseen.

Ensimmäisenä anturin asennuksessa piti valita käytetyn sovitteen tyyppi. Sovitteella tarkoitetaan tässä tapauksessa robotin työkalulaipan ja voima-anturin välisessä kiinnityksessä käytettyä sovitelaippaa. Kiinnitystavalle asennuksessa oli kaksi vaihtoehtoa. Ensimmäisenä vaihtoehtona oli Fanucin valmistama standardisovite ja toisena vaihtoehtona standardisovite ja tilaustyönä tehty sovite. Vaikka järjestelmässä käytettiin itse koneistettua sovitetta, voitiin järjestelmään valita käytettäväksi Fanucin valmistamaa standardisovitetta. Mitoiltaan molemmat kappaleet olivat saman kokoisia, eivätkä niiden fyysiset ominaisuudet eronneet toisistaan merkittävästi. Ainoa ero näiden kahden vaihtoehdon välillä oli, että standardisovite piti työkalulaipan ja voima-anturin keskiön samassa positiossa xy-koordinaatistossa. Tilaustyönä tehty laippa olisi mahdollistanut voima-anturin suuntauksen samassa xy-koordinaatistossa. Tässä työssä käytetty sovitelaippa piti anturin ja työkalulaipan samassa asemassa kuin standardikin, joten ohjelmallisesti voitiin valita se käytettäväksi.

Asennusta varten robotti piti ajaa asentoon, jossa robotin työkalulaippa osoitti suoraan ylöspäin. Tämä helpotti anturin asennusta. Avustavassa asennuksessa tämä oli esitetty

akseleiden referenssikulmilla sekä akseleiden todellisilla kulmilla. Jokainen akseli piti ajaa esitettyyn kulmaan. Robotin ollessa asetetussa asennossa voitiin voima-anturin asennus suorittaa. Voima-anturin asennettiin kappaleen 5.1 Asennus mukaisesti.
Anturin toiminnan kannalta voima-anturi piti kalibroida ennen käyttöä. Valmistaja toimitti voima-anturin kanssa anturille tehdyn CCSCB2.CM tiedoston, joka sisälsi anturille luodun kalibrointidatan. Tiedosto toimitettiin CD-levyllä, joka piti siirtää USB-laitteelle siirtoa varten. Asennushetkellä voima-anturi ei saanut olla kytkettynä, joten anturin kaapeli piti irrottaa. Tiedoston lataamista varten tallennuslaite piti asettaa kontrollerin tai ohjaimen USB-porttiin. Ohjaimella mentäessä tiedostovalikkoon asetettiin järjestelmälle laite, josta lataus suoritettiin. Laite, johon kalibrointidata ladattiin ei saanut sisältää muita tiedostoja, kuin tarvittavan tiedoston. Ylimääräiset tiedostot olisivat voineet vaikuttaa tiedoston asennukseen. Latauksen jälkeen avustettu asennus asensi tiedoston ja pyysi järjestelmän uudelleenkäynnistystä.

Avustavan asennuksen seuraavassa vaiheessa hyväksyttiin järjestelmän lämpötila. Lämpötila on keskeinen suure voima-anturin toiminnassa. Lämpötilan muutos vaikuttaa voima-anturin antamiin arvoihin, koska lämpötila vaikuttaa anturin metalliseen runkoon ja sen sisäisiin komponentteihin laajentaen tai kutistaen niitä. Tämä ilmiö luo vetoa anturissa, jonka anturin signaali ilmaisee voimana. Valmistajan antamien ohjeiden mukaisesti anturin annetaan olla paikallaan 30 – 60 minuuttia tai niin kauan, että lämpötila on stabiloitunut anturissa. Tässä prosessissa voidaan tarkastella voima-anturin reaaliaikaista lämpötilaa sekä voimia. Lämpötilan stabiloitumisen aikana voitiin huomata, kuinka anturin ilmoittama voima nousi noin 20 Newtonia, kun lämpötila nousi noin 6 celsiusastetta. Lämpötilan tasaannuttua voitiin anturi kiristää lopulliseen momenttiinsa ja hyväksyä lämpötila järjestelmälle.

Anturin kiristämisen jälkeen varmistettiin anturilta saatavan signaalin muutos kiristysmomentin aiheuttaman voiman suhteen. Tämä toiminto kuului avustavan asennuksen toimintoihin, jossa mittausdata esitettiin 8 eri anturin pisteestä. Jokaisesta pisteestä voitiin lukea anturin antama signaali. Tämän arvon tuli olla alle 15000. Jos pisteiden luettu arvo oli alle sallitun maksimin, antoi ohjelma merkin hyväksynnästä. Tässä työssä kaikki arvot olivat alle sallitun maksimin ja suurin arvo oli noin 6300. Jos järjestelmä havaitsisi arvon, joka ylittäisi sallitun maksimiarvon, se ilmoittaisi kyseisen pisteen kohdalla ylityksen. Ylityksen sattuessa kaikki voima-anturin pultit tulee löystyttää ja kiristää takaisin annet-

tuun momenttiin. Jos uudelleen kiristäminen ei auta, ongelmaan tulee asentajan olla yhteydessä valmistajaan, koska kyseessä voi olla laitevirhe, joka vaikuttaisi mittaustuloksiin.

Viimeisenä vaiheena voima-anturin asennuksesta syntyvä virhe piti poistaa. Anturin antamaa voimaa pystyi järjestelmässä koko ajan lukemaan reaaliaikaisesti ja sen pystyi nollaamaan tarvittaessa. Virheen poistossa voima-anturi oli asennuksen ajan pysynyt samassa asennossa eikä ulkoisia voimia ollut kohdistunut anturiin. Tässä tapauksessa, kun voima-anturi antoi signaalina voimaa, jota ei mittaushetkellä ollut, vaan se oli aiheutunut asennuksesta, voitiin jokaisen akselin suuntaiseksi voimaksi asettaa 0. Tällöin anturin antama arvo vastasi todellisuutta.

Käyttöönoton jälkeen voitiin voima-anturille asettaa oletusarvona käyttäjäkoordinaatisto ja hyväksyä käyttöönotto suoritetuksi. Tämän jälkeen järjestelmä piti käynnistää uudelleen, jotta konfiguraation arvot asettuivat voimaan. Prosessin jälkeen pystyttiin voima-anturin ja sen funktioiden käyttö aloittamaan normaalisti.

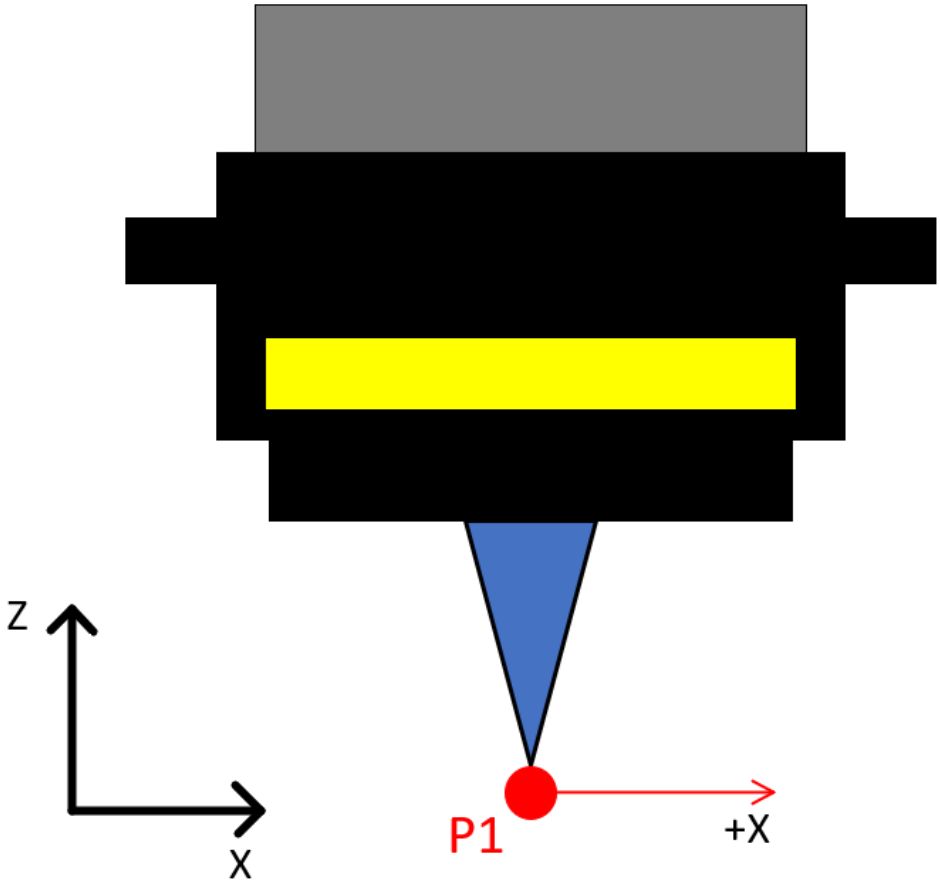
5.3 Funktiot

Käytettäessä voima-anturin mahdollistamia toimintoja luodaan robotin ohjelmalistaan funktio, jota halutaan käyttää ohjelmassa. Voima-anturilla voidaan määritellä erilaisia toimintoja funktiosta riippuen. Sitä voidaan käyttää esimerkiksi tunnistamaan kosketus kappaleen ja robotin työkalun välillä tai sovittamaan kappale sille tehtyyn uraan tai reikään.

5.3.1 Constant push- / face match-funktio

Funktio constant push on suunniteltu käytettäväksi robotilla niin, että robotti työntää kevyesti yhtä työstettävän kappaleen pistettä kohden. Tätä funktiota voidaan hyödyntää, jos työstössä jotakin työstettävän kappaleen osaa tarvitsee painaa tietyllä voimalla tai robotin pitää saada jokin kappale liikkumaan tietyllä voimalla. Constant push-funktiolle voidaan määrittää työntösuunta, joka tulee olla jokin asetetun työkalu- tai käyttäjäkoordinaatiston positiivisista tai negatiivisista akselin suunnista. Tässä suunnassa robotti lähtee tekemään työntöliikettä komentoa edeltävästä pisteestä. Funktiolle pitää myös määrittää voima, josta robotti ymmärtää, että se on saanut kosketuksen kappaleeseen sekä

lähestymisnopeus. Joissakin tapauksissa työkalun tai työstettävän kappaleen ominaisuudet voivat aiheuttaa tilanteen, jossa kosketusvoiman kynnykseksi pitää asettaa suuri voima kappaleen tunnistamiseksi. Optimaalisessa tilanteessa kuitenkin työkalu sekä kappale ovat jäykkiä, joten kosketuksen tunnistukseen ei tarvita suurta voimaa. Lähestymisnopeuden säätö on tärkeä osa kosketuksen tunnistusta ja tarkan tunnistuksen aikaan saamiseksi hidas nopeus on suositeltavaa. Tällä päästään parempaan tarkkuuteen, koska robotilla on enemmän aikaa havaita asetettu voima, jonka kosketus kappaleeseen synnyttää. Kuvan 14 mukaisesti nähdään, että asetettaessa työntösuunnaksi tässä tapauksessa +X, robotti lähtee edellisestä pisteestään kyseiseen suuntaan, kun jokin force control -funktio antaa sille käskyn.



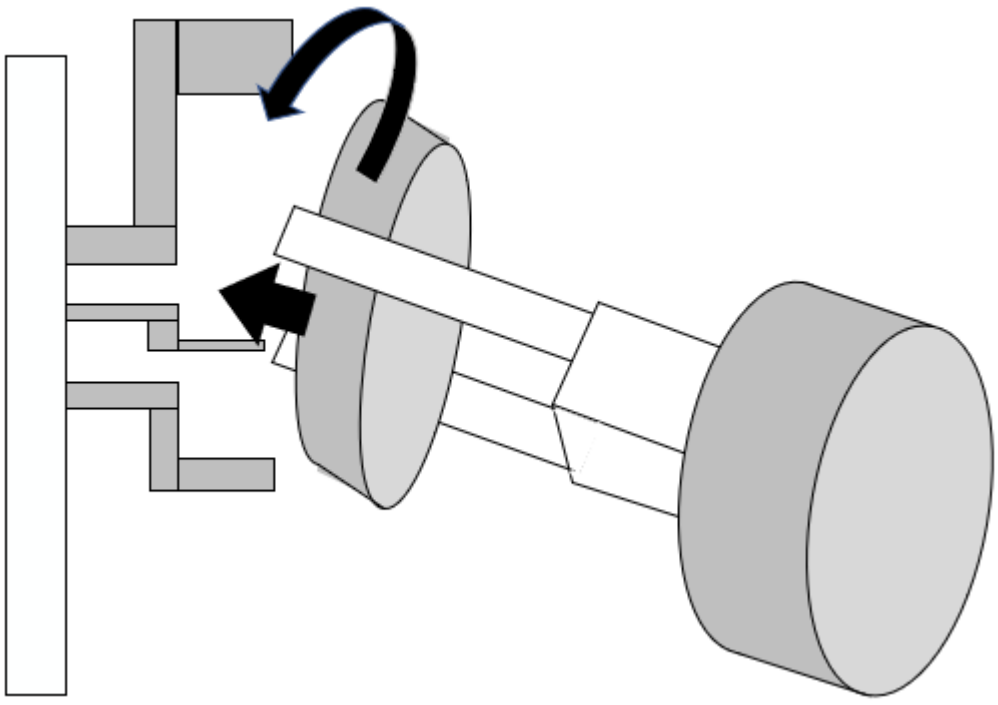
KUVA 14. Työntösuunta

Funktion parametrejä muokattaessa tärkein parametri on työntövoima. Työntövoima ei saa koskaan olla pienempi kuin kosketuksen havaitsemiseen asetettu voima. Jos voima on pienempi kuin kosketusvoiman kynnys, robotti luulee laskennassaan menettäneensä kosketuksen kappaleeseen, johon se vastaa virheilmoituksella pysäyttäen ohjelman.

Voimaa ei saa myöskään koskaan asettaa ylittämään voima-anturin kestävyyden rajoja, joka tässä tapauksessa oli noin 392 Newtonia.

Asetettu työntöaika määrittelee kuinka kauan robotti suorittaa työntöä. Kun työntöaika saavutetaan, robotti lopettaa funktion ja jatkaa ohjelmaa. Robotille ei tällä funktiolla ole mahdollista asettaa matkaa, joka robotin pitäisi kulkea. Tämä johtuu siitä, että joissakin tilanteissa tulisi virheitä ja robotti saattaisi haluamattaan joutua nostamaan voimarajoja matkan saavuttamiseksi.

Face match -funktio on tarkoitettu käytettäväksi silloin, kun kappale täytyy asettaa kohteeseen niin sanotusti ujuttaen. Hyvä esimerkki funktion käyttötarkoituksesta on, kun robotilla asetetaan kappaletta sorvin kolmileukapakkaan. Tässä tapauksessa ei ole aina absoluuttista tarkkuutta robotin tartunnasta, jolloin kappaletta ei aina voida tuoda pakalle samassa asennossa. Kappaleen asettaminen tapahtuu kuvan 15 mukaisella tavalla, jossa robotti kääntää kappaletta sorvin pakkaa kohti, jolloin kappaleen pinta sekä sorvin pakka ovat saman suuntaiset, kun kappale on asetettu sorviin. Tämä funktio osaa liikuttaa kappaletta pois kohteen ruonoista, jos törmäys tapahtuu asetuksen aikana.



KUVA 15. Asettaminen sorvin pakkaan face match -funktiolla

Niin kuin funktiolla constant push, tälläkin funktiolla on samat asetettavat parametrit. Ainut ero näiden funktioiden välillä on, että constant push on tarkoitettu tekemään vain yksi suoraviivainen liike, kun taas face match on tarkoitettu tekemään kaareva liike, jolloin kappaletta ei tarvitse tuoda kohtisuoraan kohteeseen nähden. Kaarevan liikkeen etu on siinä, että tällä tavalla pystytään kappaleita asettamaan kohteisiinsa ahtaissakin paikoissa, joihin ei muuten olisi mahdollisuutta lähestyä. Asetetulla voimalla varmistetaan, että robotti on työntänyt kappaleen haluttuun asemaan halutulla voimalla.

5.3.2 Shaft insert- / groove insert- / square insert -funktio

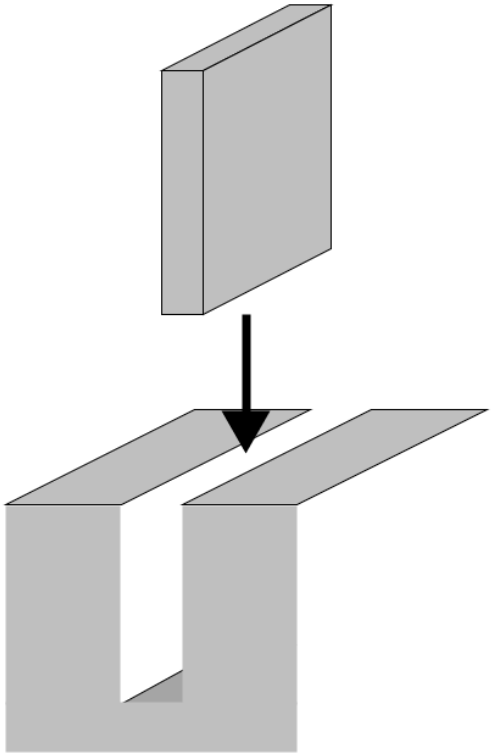
Kappaleessa käsitellään funktioita shaft insert, groove insert ja square insert, jotka tarkoittavat akseli-, ura- ja neliöasetusta. Nämä ovat myös samalla tapoja, joita voidaan käyttää robotin asettaessa kappaleita erilaisiin uriin tai väleihin. Funktioita ei käytetty kyseisessä opinnäytetyössä tai sen aikana suoritetuissa kokeissa, joten funktiot on käsitelty pintapuolisesti sen tiedon perusteella, jota on saatu valmistajien dokumenteista sekä epävirallisista kokeista.

Funktiota shaft insert on tarkoitus käyttää robotin työtehtävissä, joissa asetettavassa kappaleessa tai vastaavasti vastakappaleessa on jokin sylinterin muotoinen uloke, joka pitää asettaa vastakappaleen uloketta vastaavaan reikään. Funktio yrittää kääntää kappaletta asettaessaan sitä paikalleen niin, että kappaleiden pinnat kohtaisivat ja robotti pystyisi suorittamaan kappaleen asettamisen haluttuun pisteeseen.

Käytetylle funktiolle pitää asettaa työntösuunta, johon robotti pyrkii työntämään kappaletta paikoilleen. Tämä suunta voi olla minkä tahansa luodun työkalu- tai käyttäjäkoordinaatiston akselin suuntainen. Funktio tarvitsee myös samat perusparametrit, kuin muutkin funktiot. Kosketuksen tunnistusvoima ja lähestymisnopeus määrittelevät, kuinka robotti tunnistaa, että kappaleet ovat kosketuksissa toisiinsa, jolloin se voi aloittaa kappaleen sovittamisen paikalleen. Lähestymisnopedella pystytään säätelemään tunnistuksen tarkkuutta, koska hitaasti liikuttaessa robotilla on enemmän aikaa reagoida anturin ilmaisemaan kosketukseen. Insert-funktioille yksilöllisinä parametreina voidaan määritellä asetusnopeus, asetusvoima, asetussyvyys, asetuksen toleranssi sekä asetuksen kallistuksen maksimi. Asetusvoima sekä asetusnopeus vaikuttavat merkittävästi kappaleen asettamiseen. Voima ja nopeus tulisi aina määritellä tapauskohtaisesti, jos ennakkotietoja kappaleiden asetuksesta ei entuudestaan ole. Asetussyvyys määrittelee matkan, jonka robotti

työntää kappaleita kiinni toisiinsa sen jälkeen, kun se on saavuttanut asetetun kosketusvoiman kynnyksen. Asetuksen toleranssit määrittelevät, kuinka paljon kappaleen lopullinen paikka saa erota asetetusta asemasta. Toleranssit määritellään millimetreinä ja ne vaikuttavat ennalta asetettuun pisteeseen nähden työntösuunnan akselin mukaisessa suunnassa.

Funktio voidaan asettaa myös tilaan, jossa se valvoo robotin työkalun orientaatiota. Orientaation valvonnassa robotin työkalu ei saa ylittää asetettuja kulma-arvoja. Joissakin tapauksissa robotin työkalu ei saa päästä kääntymään, koska se voisi aiheuttaa vahinkoa kappaleille, robotin työkalulle tai muulle ympäristölle, joten orientaation valvonta on tehty mahdolliseksi. Tilan ollessa aktiivinen robotti ei pysty muuttamaan orientaatiotaan funktion ollessa käynnissä enempää kuin on asetettu. Jos robotilla ei tilanteessa ole muuta mahdollisuutta kuin muuttaa orientaatiotaan yli asetettujen raja-arvojen, se pysäyttää ohjelman ja ilmoittaa virheestä käyttäjälle.



KUVA 16. Kappaleen asetus uraan funktiolla groove insert

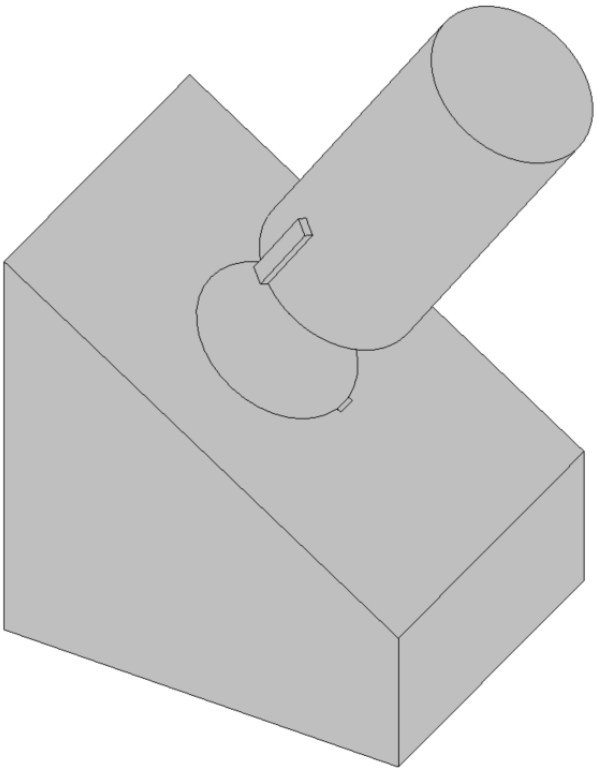
Groove insert -funktio on tarkoitettu tilanteeseen, jossa robotilla asetetaan kappaletta uraan. Kuvan 16 mukaisessa tilanteessa robotilla asetetaan kappale uraan. Voima-anturin tunnistaessa kappaleen olevan kontaktissa vastakkaiseen kappaleeseen voidaan se asettaa

hakemaan kappaleelle sopivaa uraa muuttamalla työkalun orientaatiota. Tällä tavoin löydetään asema, josta voidaan lähteä työntämään kappaletta asetettuun suuntaan. Funktion parametrit ovat samat kuin edellä mainitussa funktiossa. Funktiot toimivat saman kaltaisesti ja niiden toiminnassa on vain pieniä eroja, jotka erottavat ne toisistaan. Erot syntyvät lähinnä käsiteltävien kappaleiden muodoista ja siitä, miten niiden asetuksen aikana on optimaalisinta käyttäytyä.

Funktiota square insert on tarkoitettu käytettäväksi silloin, kun robotilla on tarkoitus asettaa pinnaltaan neliön muotoinen kappale sitä vastaavaan reikään. Voima-anturin tunnistaessa kappaleen kontakti vastakkaiseen kappaleeseen, alkaa robotti työntämään kappaletta asetettuun suuntaan samalla muuttaen orientaatiotaan niin, että neliön muotoinen kappale tai kappaleen osa saadaan asetettua sen vastakappaleeseen. Funktio toimii saman kaltaisesti kuin edellä mainitut insert-funktiot. Tämän funktion parametrit ovat yhtenevät edellä mainittujen kanssa.

5.3.3 Phase match insert -funktio

Phase match insert -funktio on tarkoitettu työstöön, jossa robotin pitää asettaa sen työkalussa oleva kappale vastakappaleeseen, kun näissä kappaleissa on kieli tai muu uloke, jolloin työkalua pitää kiertää yhden akselin ympäri kappaleen paikalleen saamiseksi. Tässä funktiossa työntösuunnan valinnalla on kaksi tarkoitusta. Työntösuunta määrittää suunnan, johon robotti yrittää työntää kappaletta, jotta se asettuisi haluttuun asemaan. Tätä arvoa käytetään myös kertomaan akseli, jonka ympäri robotti pyörittää kappaletta sovittaen kieltä tai muuta uloketta oikeaan kulmaan kappaleen asettamisen onnistumiseksi. Kuvan 17 mukaisesti voi olla tilanne, jossa kappaleet eivät ole noudettuna aina samassa asennossa, joten kappaleen pyöritys kuvan 17 mukaisessa tilanteessa akselin pituussuunnan ympäri on välttämätöntä työn suorittamiseksi.



KUVA 17. Funktio phase match insert

Funktiolle täytyy antaa perusparametrit sen toimivuuden varmistamiseksi. Funktiolle täytyy antaa parametrinä kosketusvoiman kynnys, joka tarkoittaa voimaa, joka saavutetaan kosketuksesta kappaleesaan. Lähestymisnopeus, joka on nopeus, jolla robotti lähestyy kappaletta edellisestä pisteestään. Funktio tarvitsee myös koordinaatistot, joita se käyttää sen suorituksessa. Nämä koordinaatistot ovat työkalu- sekä käyttäjäkoordinaatisto, joiden avulla valitaan myös asetussuunta. Asetussuunta pitää olla yhdensuuntainen jonkun käyttäjä- tai työkalukoordinaatiston akselin kanssa. Jos ulkoinen osa pysyy aina robottiin nähden samassa asemassa, kannattaa käytettäväksi tehdä käyttäjäkoordinaatisto kappaleelle niin, että koordinaatiston Z-akseli osoittaa haluttuun asetussuuntaan.

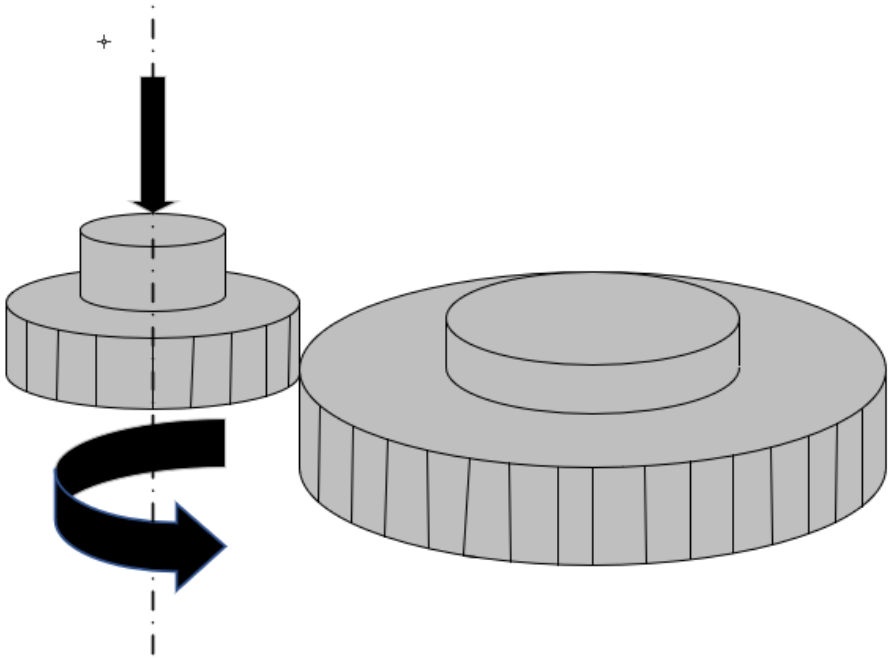
Koska tämä funktio koostuu kahdesta osasta, jotka ovat asteittainen sovitus ja asetus, pitää näille vaiheille antaa halutut parametrit. Asteittainen sovitus tarkoittaa, että kosketuksen saatuaan robotti alkaa pyörittää kappaletta asetetun akselin ympäri asetetulla nopeudella. Samalla robotti työntää kappaletta saman akselin mukaisesti asetetulla voimalla, jotta oikealle kohdalle saavuttua kappale pääsee asettumaan vastakappaleeseensa. Kun kappale on asettunut oikeaan kulmaan, robotille kerrotaan voima, jolla se työntää kappaletta sen vastakappaleeseen. Työntösyvyys on myös asetettava, jotta robotti tietää onko se saavuttanut halutun aseman kappaleelle. Jos robotti ei jostakin syystä pääse haluttuun

asemaan se lopettaa funktion suorittamisen ja jatkaa käyttäjän määrittämillä virhetoimenpiteillä. Niin kuin muillakin insert -funktioilla, tälläkin funktiolla voidaan asettaa toleranssit kappaleen asetukselle sekä tarkastus orientaation muuttumiselle. Toleranssi vaikuttaa siihen, kuinka pitkälle kappale on saatu työnnettyä eli onko haluttuun asemaan päästy. Orientaation tarkastus pakottaa robotin pitämään orientaationsa asetettujen kulma-arvojen sisällä, kun robotti suorittaa työntöä. Tämä toiminto ei vaikuta, kun robotti hakee kappaleelle oikeaa kulmaa.

5.3.4 Search-funktio

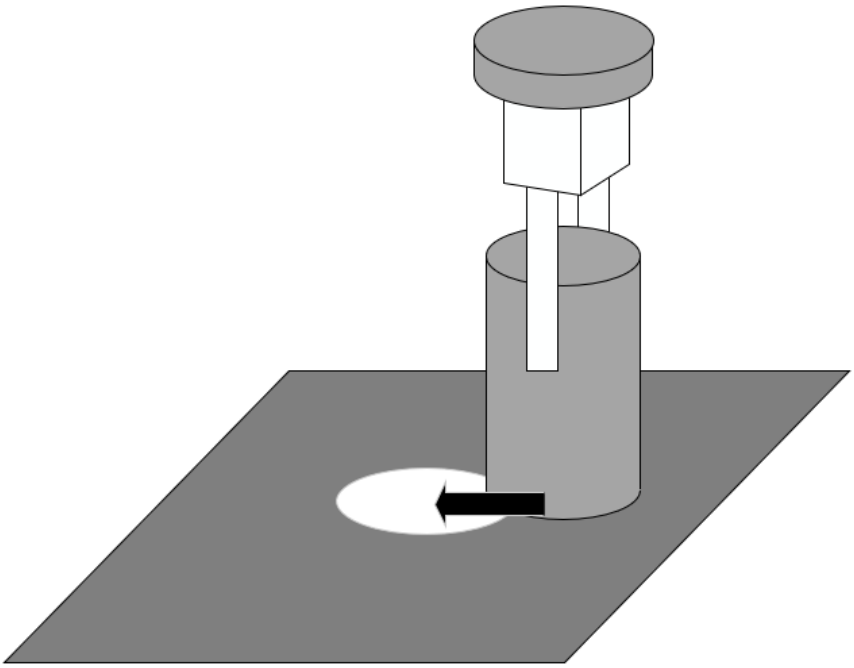
Search -funktio voidaan jakaa neljäksi erityyppiseksi funktioksi, koska niiden käyttökohteet eroavat toisistaan. Tavalliseksi search -funktioksi, jolla robotti voidaan asettaa etsimään voima-anturin avulla jopa viidestä eri suunnasta haluttuja pintoja. Tämä on funktion perusasetus, jolla on tarkoituksena esimerkiksi etsiä kappaleen reunoja ja tallentaa näiden kohtien paikat muistiin. Näistä pisteistä voidaan muodostaa esimerkiksi käyttäjän määrittelemä käyttäjäkoordinaatisto. Perusfunktiolla voidaan myös esimerkiksi paikoittaa jokin kappale myöhempää käyttötarkoitusta varten.

Phase search -funktio on tarkoitettu kappaleen sovittamiseen niin, että robotti pyörittää kappaletta asetetun akselin ympärillä. Tällä tavalla voidaan esimerkiksi asettaa hammaspyörä paikalleen tietämättä, miten hammaspyörästä on otettu kiinni siihen tartuttaessa. Kuvan 18 mukaisessa tilanteessa hammaspyörää pyöritetään etsien voima-anturin avulla oikeaa kulmaa asettamista varten. Robotti pyörittää kappaletta samalla työntäen sitä kohti vastakappaletta asetetulla voimalla. Kun robotti kiertää kappaleen oikeaan asentoon, voima laskee ja robotti tulkitsee, että se on saavuttanut halutun kiertymän. Tämän jälkeen robotti voi työntää kappaleen paikoilleen.



KUVA 18. Phase search – Hammasrattaan paikotus

Hole search -funktiossa robotti liikuttaa työkalulla tartuttua kappaletta sen vastakappaleen pinnalla etsien kappaleelle reikää, johon robotti voi sen asettaa. Kuvan 19 mukaisessa tilanteessa robotti on työntänyt kappaletta ensin tason pintaa vasten, jonka jälkeen se alkaa tekemään sahalaitaista liikettä kohti reikää. Sahalaitainen liike auttaa robottia löytämään oikean kohdan, johon käsitelty kappale kuuluu. Saman aikaisesti robotti työntää kappaletta pintaa kohti asetetulla voimalla. Kun robotti on saavuttanut kohdan, josta se voisi työntää kappaleen paikoilleen voima-anturin antama voima laskee, koska työnnettävällä kappaleella ei ole tukivoimaa. Tämän jälkeen robotti pysähtyy ja alkaa työntämään kappaletta paikalleen.

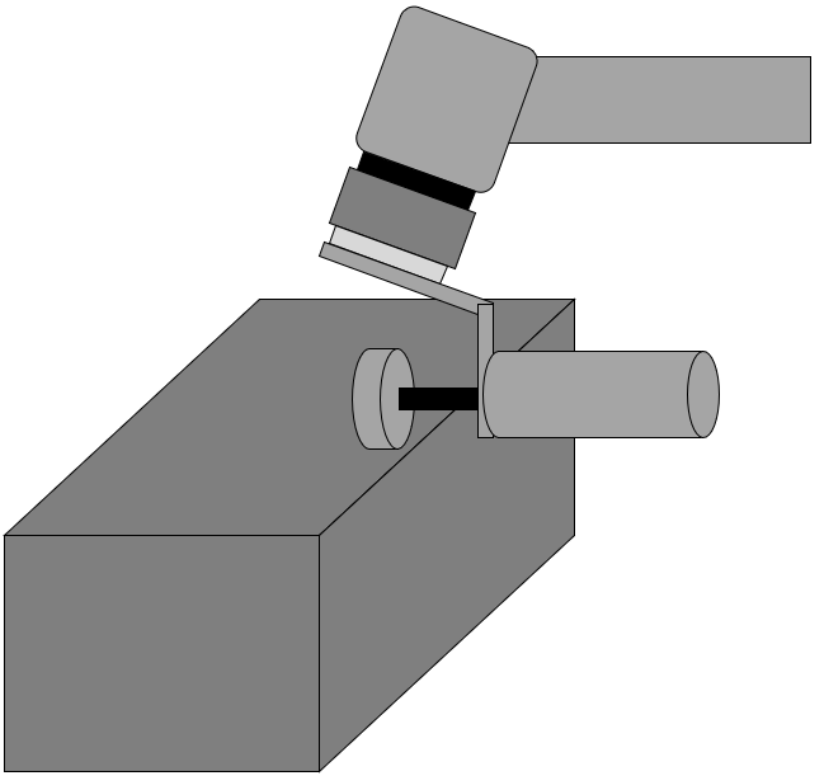


KUVA 19. Hole search -funktio

Clutch search -funktio on tarkoitettu monitasoisten sovitteiden etsimiseen. Funktio toimii kuin edellä mainittu phase search -funktio, mutta tässä funktiossa, kun robotti on löytänyt oikean kohdan, johon sen pitämä kappale sopii, se asettaa kappaleen paikalleen ja suorittaa etsinnän uudelleen. Esimerkiksi asettaessa kytkimen napaa kytkinlevyille robotin täytyy saada napa sovitettua useammalle levylle, jotka on sijoitettu päällekkäin. Jokaisen kerroksen välissä robotti kiertää kytkimen napaa löytääkseen oikean kulman, josta se voi asettaa navan paikalleen levyn tullessa sen ympärille.

5.3.5 Contouring -funktio

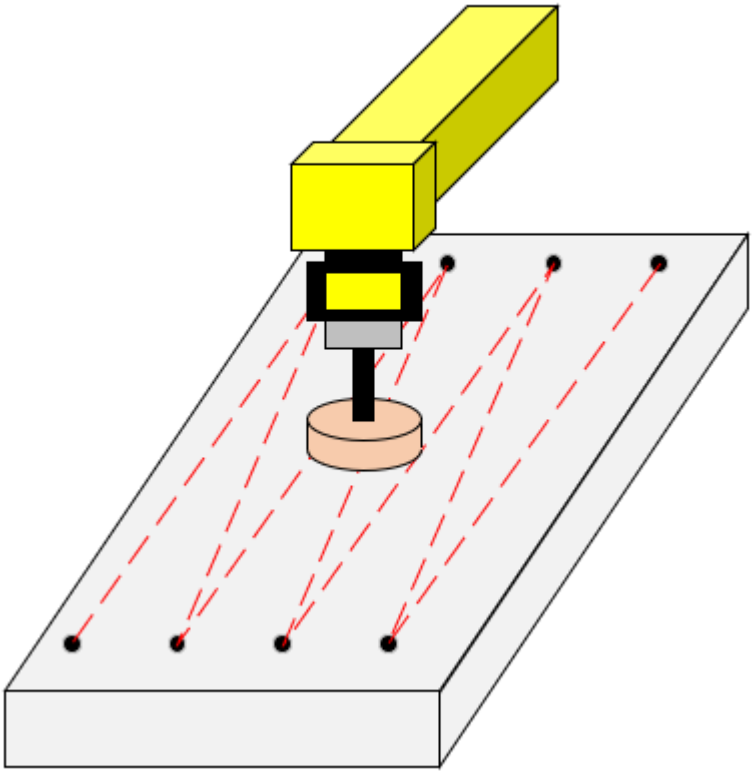
Contouring-funktio on opinnäytetyössä käytetty funktio. Contouring eli muotoilu -termillä voidaan tarkoittaa tässä tapauksessa mitä tahansa työtä, jonka tarkoituksena on poistaa materiaalia työstettävästä kappaleesta, jonkin käytetyn työkalun avulla. Kuvassa 20 on esitetty tilanne, jossa koneviilalla tai muulla vastaavan kaltaisella työkalulla robotti työstää kappaletta käyttäen contouring-funktiota. funktion tarkoituksena on pitää työkalun ja työstettävän kappaleen välinen voima vakiona. Tällä saavutetaan ainakin teoriassa tasainen työstöjälki. Funktiolla on monia ominaisuuksia, joilla voidaan helpottaa monimutkaisten kappaleiden työstöä. Esimerkiksi työntösuunnan automaattinen kääntö on funktio, jota opinnäytetyössäkin käytettiin. Tämän ominaisuuden tarkoituksena on kääntää työntösuuntaa liikevektorin suunnan mukaan.



KUVA 20. Contouring-funktio

Tämä funktio pyrkii pitämään työkalun työstettävän kappaleen pinnassa aina tasaisella voimalla. Joissakin tapauksissa voidaan kuitenkin määritellä funktiolle tasoja, joissa robotti työskentelee. Tätä ominaisuutta voidaan käyttää esimerkiksi kappaleiden pinnan epätasaisuuksien poistamisessa, joissa poistettavan materiaalin esiintyminen on epäsäännöllistä. Tässä tapauksessa robotti lähestyisi kappaleen pintaa käyttäjän määrittelemän askeleen verran kerrallaan ja poistaisi samalla ylimääräisen materiaalit kappaleen pinnalta.

Ilman työntösuunnan automaattista kääntöä robotin työntösuunta on aina sidottu funktion parametreihin asetettuun arvoon. Työntösuunnan arvo pitää aina olla yhdensuuntainen työkalu- tai käyttäjäkoordinaatiston jonkin akselin kanssa. Jos työntösuunta sidotaan robotin työkalukoordinaatistoon, se kääntyy robotin muuttaessa työkalun orientaatiota. Tätä voidaan tietyissä tapauksissa hyödyntää, mutta varsinkin työstöradan tehdessä yli 360 asteisen kierroksen voi robotin olla mahdotonta pystyä kääntämään työkalua vaadittua määrää. Tällaisessa tilanteessa tulee käyttää ominaisuutta työntösuunnan automaattinen kääntö.



KUVA 21. Contouring-funktio käytössä kiillotuksessa.

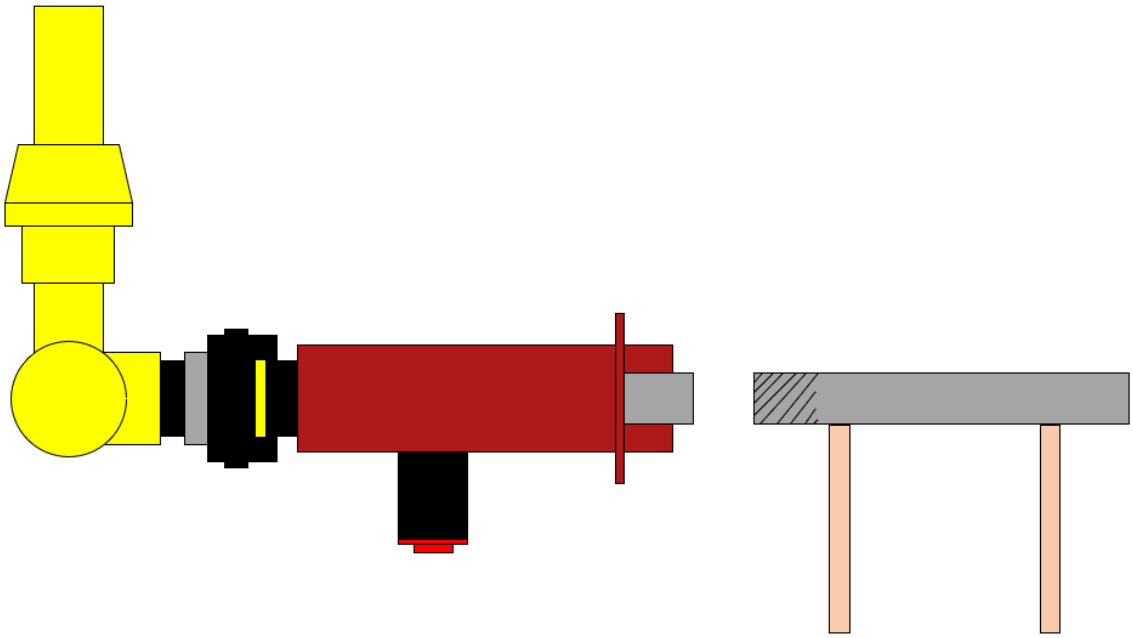
Funktiota voidaan käyttää myös esimerkiksi kappaleiden kiillotuksessa. Kuvan 21 mukaisesti funktion avulla voidaan pitää tasainen voima työkalun ja kiillotettavan kappaleen välillä koko kiillotuksen ajan saaden tasainen työstöjälki.

Funktiolle pitää antaa työntösuunta. Työntösuunta voi olla joko työkalukoordinaatiston tai käyttäjäkoordinaatiston jonkun akselin suuntainen. Jotkin funktion ominaisuudet vaativat, että funktio on sidottu käyttäjäkoordinaatistoon, jolloin työkalukoordinaatiston käyttöä ei sallita. Kuten muissakin funktioissa, myös tässä tarvitaan kosketusvoiman kynnysarvo sekä lähestymisnopeus. Parametreistä voi valita lähteekö robotti kulkemaan työstörataa ennen vai jälkeen kosketuksen varmistamisen. Tällä voidaan joissakin tapauksissa saada työlle parempi laatu sekä helpottaa työstöä tietyissä tapauksissa. Työstövoima sekä työntömatkan raja-arvo tulee asettaa tapauskohtaisesti. Työstövoima ilmaisee kuinka suurella voimalla robotti työntää työkaluaan työstettävää kappaletta kohti. Työstövoima ja työstö nopeus kulkevat käsi kädessä. Jos voimaa kasvatetaan, työstössä työstettävästä kappaleesta poistetaan enemmän materiaalia, jos robotin nopeutta ei nosteta. Voiman ja nopeuden kompensaatioon on tässä funktiossa olemassa erilaisia ominaisuuksia, joiden avulla voidaan nopeuttaa työstöä tai säätää voimaa laadun kärsimättä muutoksesta. Nämä

muutokset täytyy kuitenkin ottaa käyttöön tapauskohtaisesti. Työntömatkan raja-arvo ilmaisee kuinka paljon robotti saa poiketa asetetusta pisteestä, jotta se saa vielä jatkaa työstöä. Koska contouring-funktiota käytettäessä robotin paikkapisteitä ei tarvitse asettaa kiinni kappaleen pintoihin se joutuu aina painamaan vähän matkaa ennen, kuin se osuu kappaleeseen. Jos kuitenkin robotti esimerkiksi joidenkin ominaisuuksien takia hidastaa nopeuttaan, mutta pitää voiman vakiona, se ei saa ylittää jotakin matkallista arvoa.

5.3.6 Threading-funktio

Threading -funktiota on tarkoitus käyttää, kun robotilla suoritetaan työstöä, jossa tarvitaan momentin hallintaa. Threading tarkoittaa kierteitystä ja sitä käytetään nimensä mukaisesti kierteittämiseen, jossa momentin tarkkailu on tärkeää. Kuvassa 22 on esitetty tilanne, jossa robotti kierteittää kappaleen työkalullaan. Robotin työkalussa on käytössä servomoottori, joka toimii robotin ulkoisena akselina. Kierteytys voidaan toteuttaa tämän kaltaisessa tilanteessa robotin ulkoisella akselilla ja joissakin tapauksissa robotilla. Robotin ominaisuuksista riippuen ei ole aina järkevää käyttää robottia kierteittämään kappaletta vaan on parempi käyttää ulkoisen akselin hallitsemaa työkalua, koska robotin työkalulaipassa sijaitseva kuudes akseli ei kestä jokaisessa mallissa suurta momenttia. Tätä funktiota voidaan käyttää myös esimeriksi poraukseen ja ruuvien ruuvaamiseen työkohteesta riippuen. Työstöt, joissa tarvitaan hallittu momentti ovat sopivat tehtäviksi tämän funktion kanssa.



KUVA 22. Kierteitys työkalulla, jossa ulkoinen akseli.

Funktion parametreistä voidaan asettaa, käytetäänkö kierteittävässä työstössä kääntömekanismina robottia vai esimerkiksi työkalun ulkoista akselia. Työstösuunta valitaan käyttäjäkoordinaatiston tai työkalukoordinaatiston jonkin akselin suuntaiseksi. Koska kyseessä on kierteitys, pitää kierteelle valita noususuunta myötäpäivään tai vastapäivään. Tämä parametri määrittelee robotin työkalun pyörimissuunnan työstössä. Koska funktiota voidaan käyttää myös ruuvaamiseen ja poraamiseen pitää parametreissa kertoa kiristetäänkö vai löystytetäänkö esimerkiksi ruuvattavaa ruuvia. Kun robotti saavuttaa kosketusvoima kynnyksen se lähtee työntämään työkalua kohti kappaletta asetetulla voimalla samalla, kun se pyörittää työkalua asetetulla momentilla ja nopeudella. Työstölle pitää myös antaa minimi ja maksimisyvyydet, johon robotti voi ruuvata tai kierteittää kappaletta. Näillä perusparametreillä voidaan luoda toimiva kierteitysfunktio.

5.4 Funktioiden testit

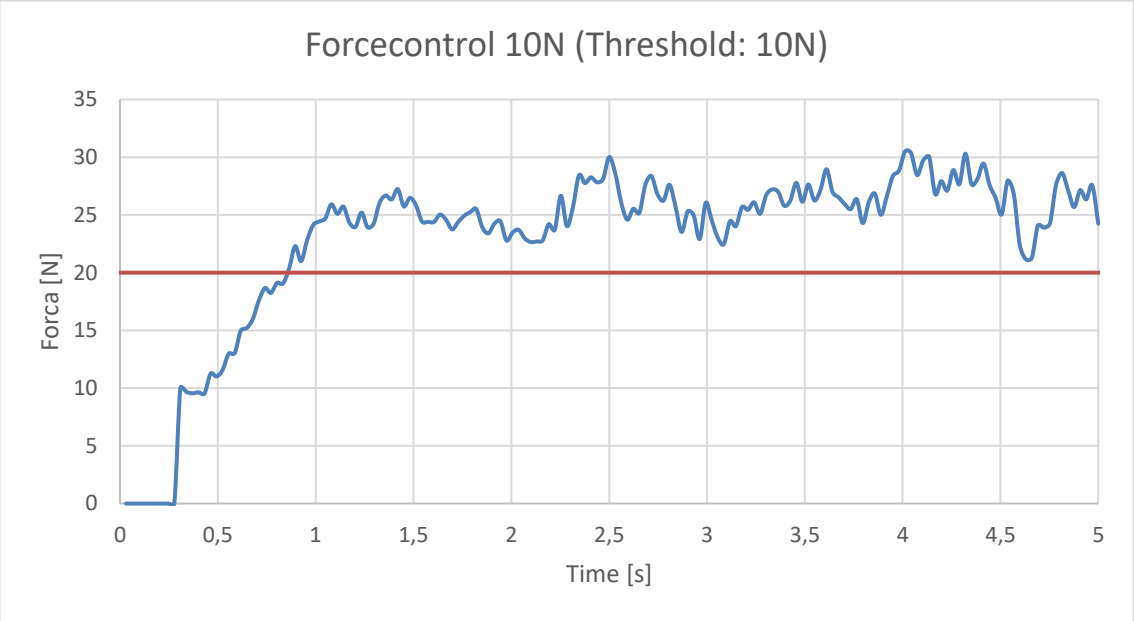
Tässä kappaleessa käsitellään testejä ja kokeellisia suorituksia työssä käytetyillä funktioilla. Testien aikana ilmenneisiin ongelmiin esitetään ratkaisut kappaleen lopussa. Nämä toimenpiteet olivat lopullinen ratkaisu opinnäytetyön suorituksessa.

5.4.1 Constant push -funktion kokeet

Voima-anturin käyttö aloitettiin kokeilemalla constant push -funktiota. Tämä funktio oli käytössä ennen muiden funktioiden optioiden tilaamista, koska tämä toiminto kuului voima-anturin kanssa toimitettuun perusoptioihin. Constant push -funktiota varten robotin työtasolle kiinnitettiin suorakulmion mallinen kappale, joka asetettiin niin, että pöydän kulmaan luotu käyttäjäkoordinaatiston positiivinen X-akseli osoitti kappaleen pintaa kohti.

Funktion parametreihin asetettiin robotin kosketusvoiman kynnykseksi 10 Newtonia, jolla robotin kosketuksen pystyi selkeästi havaitsemaan seuraten reaaliaikaista voiman monitorointia. Nopeudet lähestyttäessä pidettiin vakiona. Suurimman sallitun työntöetäisyyden asetettiin olevan 20 mm ja työntövoimaksi asetettiin alussa 10 Newtonia. Näillä parametreilla lähdettiin suorittamaan ensimmäisiä testejä robotilla, jotta voima-anturin

toimintaa pystyttiin paremmin tulkitsemaan ja ymmärtämään. Testeissä huomattiin jo ennen voimien nauhoitusta, että voima ei pysynyt tasaisena. Tämä huomattiin, kun robotti pysäytti ohjelmansa ja antoi virheeksi suurimman voiman rajan ylityksen. Tässä vaiheessa otettiin voiman nauhoitus käyttöön, jotta voitiin selvittää, mistä virhe johtui.

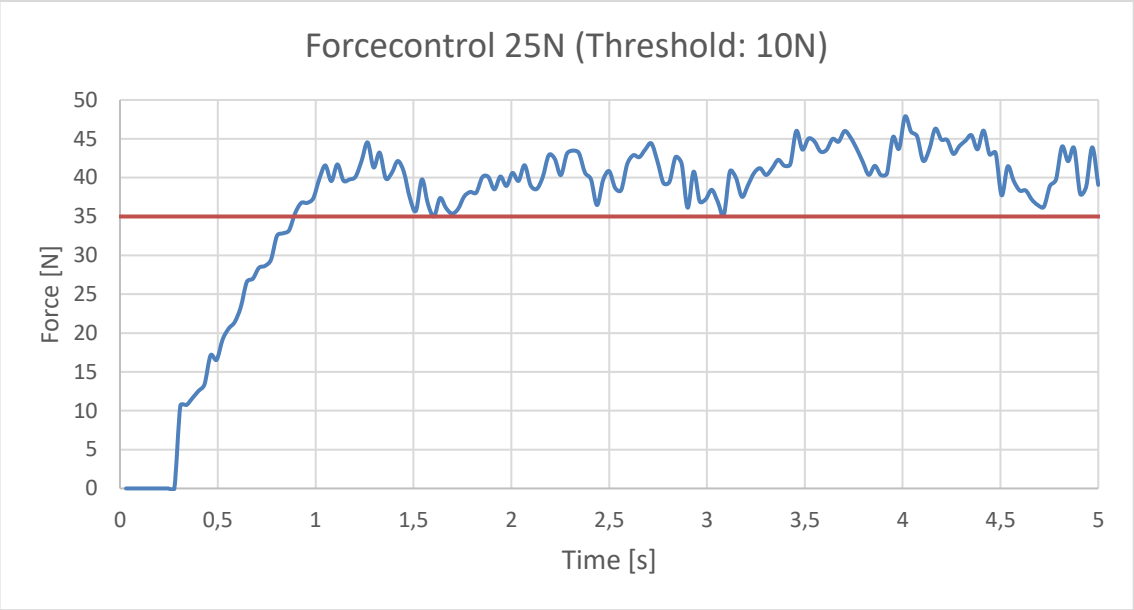


KUVIO 1. Voiman kuvaaja voimalla 10 N.

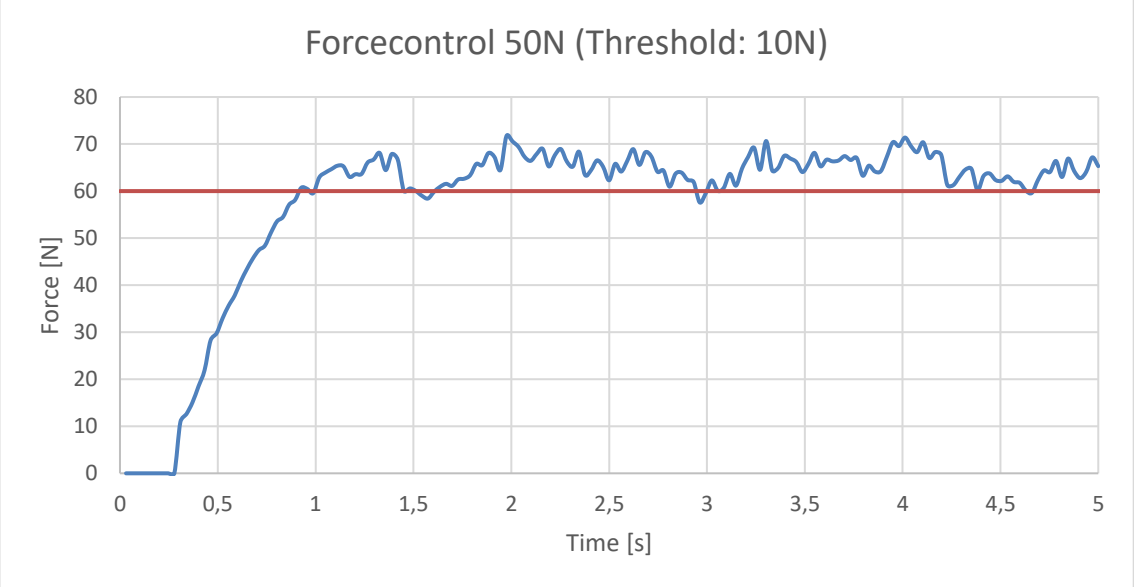
Ohjelma suoritettiin pitämällä kosketusvoiman kynnys vakiona 10 Newtonissa. Työntö voimaa muutettiin 10, 25, 50 ja 75 Newtoniin eri suorituskertojen välillä. Tallennetut voimat ja momentit ilmoitettiin jokaisessa koordinaatiston suunnassa ajan suhteen. Järjestelmä tallensi myös robotin aktiivisen työkalun sijainnin aktiivisessa käyttäjäkoordinaatistossa. Näistä tiedoista luodut kuviot asetettiin ilmaisemaan robotin voimaa ajan suhteen, koska näillä arvoilla voiman käyttäytymistä oli helpoin ymmärtää. Kuviosta 1 huomattiin, että voima ei pysynyt aseteussa 10 Newtonissa, joka lisättiin kosketusvoiman kynnyksen arvoon. Täten kuvaajassa esitetty tavoitearvo sijoitettiin 20 Newtoniin. Kuviosta nähtiin kuitenkin tilanne, jossa robotti saa ensimmäisen kerran kosketuksen kappaleeseen. Tässä tilanteessa robotti oli liikuttanut itseään hieman kappaleesta poispäin saavutettuaan asetetun kosketusvoiman.

Tuloksista voitiin päätellä jonkin olevan pielessä, koska voima saattoi joissakin tapauksissa ylittää asetetun arvon yli 50% tavoitearvosta. Voima oli kuitenkin lähes aina suu-

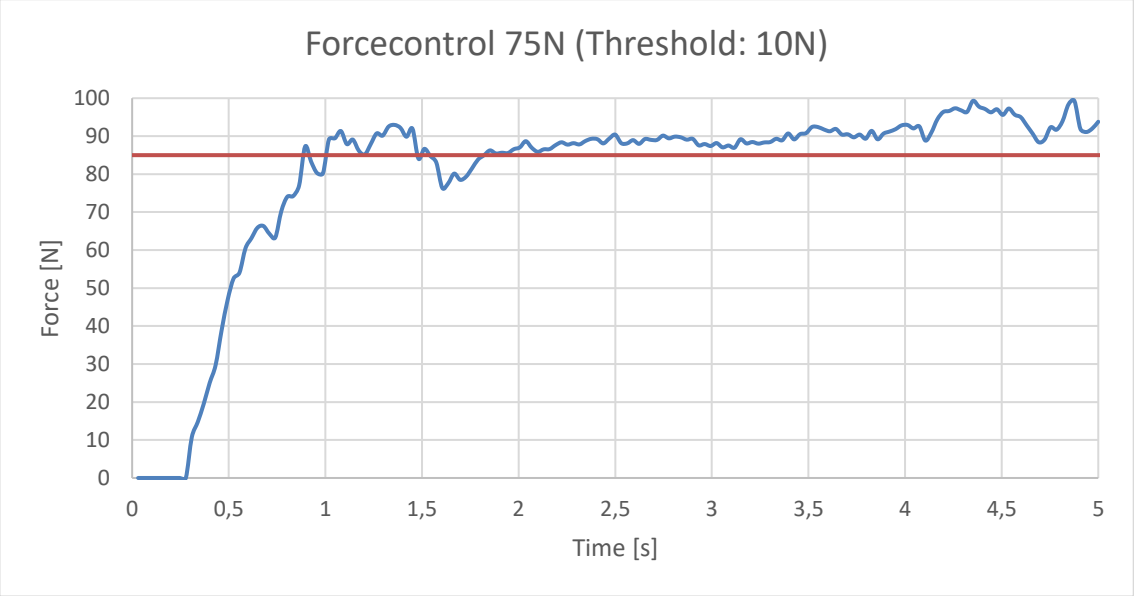
rempi kuin asetettu tavoite, joka herätti mielenkiintoa voima-anturin toimintaan. Vain yksittäisissä tapauksissa robotti laski voimaa niin, että voima laski alle asetetun tavoitteen. Kuvioista 1 – 4 huomattiin myös, että suuremmilla voimilla robotti ei nostanut voiman ylittävää osuutta suhteessa asetettuun voimaan, vaan sen toiminta oli täysin satunnaista. Tästä pääteltiin, että ongelma ei ollut itse funktiossa vaan jossakin ulkoisessa tekijässä. Ongelmasta huolimatta seuraavassa vaiheessa lähdettiin suorittamaan kokeellista työstörataa käyttäen contouring -funktiota.



KUVIO 2. Voiman kuvaaja voimalla 25 N.



KUVIO 3. Voiman kuvaaja voimalla 50 N.



KUVIO 4. Voiman kuvaaja voimalla 75 N.

5.4.2 Contouring -funktion kokeet

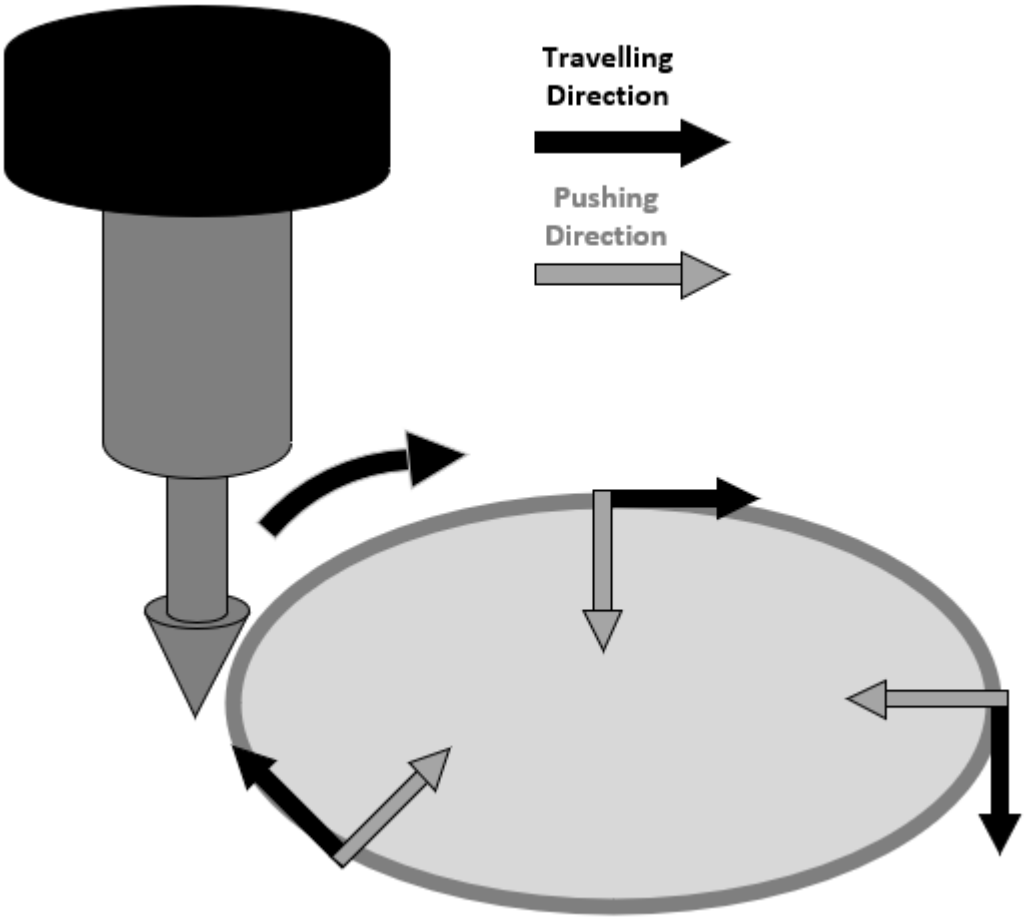
Contouring -funktion käyttö aloitettiin kokeilemalla sitä samalla kappaleella, jota käytettiin constant push -funktiolla. Suorakulmaisessa kappaleessa oli noin 300 mm pitkä suora pinta, jolle luotiin kaksi pistettä. Näiden pisteiden välillä robotin oli tarkoitus pitää voima vakiona asetetussa arvossa työkalun ja kappaleen pinnan välillä.

Kun työstöä lähdettiin suorittamaan, ensimmäinen piste oli noin 10 mm päässä kappaleesta. Tähän pisteeseen saavuttaessa robotti aloitti funktion suorittamisen. Robotti aloitti funktion lähestymällä kappaletta asetetulla 10 mm/s lähestymisnopeudella. Kun robotti oli saanut kosketuksen kappaleeseen ja saavuttanut asetetun kosketusvoiman kynnyksen, se aloitti liikkeen kohti kappaleen sivun toisessa päädyssä olevaa pistettä. Tänä aikana robotti työnsi koko ajan työkaluaan kohti kappaletta. Robotti oli asetettu painamaan työkaluaan käyttäjäkoordinaatiston suuntaan -X, joka osoitti robotin lähtöpisteestä suoraan kappaleen pintaa kohti. Tässä vaiheessa robotin liikenopeus oli asetettu 100 mm/s. Robotti ei kuitenkaan saanut pidettyä kontaktia työkalun ja kappaleen välillä, koska nopeus oli liian suuri. Tällöin robotti aloitti työntämisen ollen kiinni kappaleessa, mutta työntönopeuden ollessa liian hidas se lähti liikkumaan suoraan kohti kappaleen toista pistettä. Korjatakseen ongelman robotin liikenopeuksia laskettiin 40 mm/s, joka vaikutti huomattavasti tuloksiin. Kun nopeutta oltiin laskettu, robotti sai kontaktin kappaleeseen, mutta

liikkeen aikana huomattiin, että voima, jolla robotti työnsi kappaletta, kasvoi eksponentiaalisesti. Lopulta voima kasvoi niin suureksi, että robotti pysäytti ohjelmansa ja antoi virheen voimarajojen ylityksestä. Voimien suurimmaksi sallituksi rajaksi oltiin asetettu 150 Newtonia, joten asetettu 25 Newtonia oli ylitetty kuusinkertaisesti.

Tässä vaiheessa vaikutti siltä, että robotin nopeus ja voima vaikuttivat toisiinsa niin radikaalisti, että nopeuden ollessa alhainen robotti kasvatti voimaansa, mutta ei laskenut sitä. Voiman laskeminen olisi edellyttänyt robotilta, että se olisi liikkunut kappaleesta poispäin, mutta kokeiden hetkellä tällaista tilannetta ei syntynyt. Koska valmistajalta ei saatu ongelmaan ratkaisua, alettiin muuttamaan voima-anturin voimaa nopeuden suhteen. Tälle toimenpiteelle oli contouring-funktiolla eri ominaisuuksia. Kaikkia ominaisuuksia kokeiltaessa huomattiin kuitenkin, että jokaisen toiminnon tarkoituksena oli laskea robotin liikenopeutta, kun voima laski asetetun arvon alapuolelle. Kyseisessä tapauksessa nopeutta olisi pitänyt nostaa, kun voima osoittautui liian suureksi. Tällaista ominaisuutta ei kuitenkaan ollut käytettävissä. Näitäkin toimintoja käyttämällä robotti nosti voimansa niin suuriksi, että se pakotti ohjelman pysähtymään.

Samat testit haluttiin suorittaa myös toisella kappaleella vertailun vuoksi. Tähän kokeeseen otettiin käyttöön kappale, jonka ulkokehä oli pyöreä. Tarkoituksena oli kokeilla luoda funktio uudelleen ja yrittää ajaa robotilla pyöreää liikerataa. Tästä haluttiin tutkia, miten robotti ja sen voimat käyttäytyvät erilaisella liikkeellä, kun voima-anturin toiminto automaattinen työntösuunnan kääntö oli käytössä. Automaattisen työntösuunnan käännön tarkoitus oli kääntää työntösuuntaa sen perusteella, mihin robotti kullakin hetkellä liikkui. Tästä toiminnosta on esitetty havainnollistava kuva (Kuva 23). Kuvassa robotti kääntää työntösuuntaansa ympyrällä liikeradalla aina ympyrän keskiosaa kohti. Näin saatiin varmistettua, että työstövoima saatiin pidettyä vakiona oikeaan suuntaan.



KUVA 23. Automaattinen työntösuunnan kääntö

Sama ongelma ilmaantui myös tällä kappaleella. Suuremmilla nopeuksilla robotti ei pystynyt pitämään kosketusta kappaleeseen. Vaikka voima-anturi antoi voiman arvoksi 0 Newtonia, ei robotti hidastanut vauhtia. Laskettaessa nopeutta robotti nosti voimaa, kunnes voima ylitti asetetun voiman ylärajan, joka oli 150 Newtonia.

Koska voima-anturin käyttöohjeista ei löydetty vastausta ongelmaan, oltiin yhteydessä voima-anturin valmistajaan. Valmistajan pohjoismaiden pääkonttorilla ei ollut kuitenkaan ketään, joka olisi käyttänyt ennen kyseistä voima-anturia, joten ratkaisua ongelmaan ei saatu. Odottaessa valmistajan vastausta ongelmaan päätettiin, että robotille ohjelmoidaan itse vaihtoehtoinen ohjelma, jonka suorittamisen oli tarkoitus kääntää työkalukoordinaatistoa niin, että automaattista työntösuunnan kääntö toimintoa ei tarvittu. Tämän toiminnon oletettiin häiritsevän ohjelman toimintaa.

Vaihtoehtoinen ohjelma rakennettiin niin, että robotin suorittaessa liikerataa se tulkitsi edellistä-, nykyistä- ja seuraavaa pistettä, jotka saatiin tallennettua ohjelmaan. Ohjelma kirjoitettiin Fanuc-robottien kielellä KAREL, joka on robotin tekstipohjainen ohjelmointikieli. Vektorilaskennan avulla voitiin luoda laskentakaava, jota käyttämällä saatiin selville robotin nykyisen ja seuraavan pisteen välinen kulma Z-akselin ympärillä. Kun tarvittava kulma oltiin laskettu, se tallennettiin robotin positiorekisteriin. Positiorekisterissä tätä kulmaa voitiin käyttää kirjoittamaan työkalukoordinaatisto uudelleen aina, kun robotti saavutti uuden pisteen. Ohjelmasta tehtiin monta kokeellista versiota ja niitä kokeiltiin simuloimalla ohjelmaa HandlingPRO-simulointiohjelmassa. Valmis ohjelma on esitetty opinnäytetyön raportin liitteessä 5. Ohjelmassa piti myös ottaa huomioon ensimmäinen ohjelman piste, koska tässä tapauksessa robotilla ei ollut edellistä pistettä. Tässä tapauksessa robotti käsitteli ohjelmaa toisin. Tässä kyseisessä ohjelmassa ei voitu kirjoittaa muistiin kaikkia tarvittavia pisteitä, vaan pisteiden tiedot tallennettiin positiorekisteriin erillisissä ohjelmissa. Nämä ohjelmat suoritettiin ohjelman kierrossa sitä mukaa, kun robotti saavutti tallennettavat pisteet.

5.4.3 Ratkaisut

Kun vaihtoehtoinen ohjelma oltiin saatu valmiiksi ja onnistuneet simuloinnit oltiin tehty, oli tarkoitus suorittaa ohjelma robottisolun robotilla. Ohjelman valmistumisen jälkeen voima-anturin valmistaja otti yhteyttä ja ehdotti korjaustoimenpiteitä, jotka tekemällä voima-anturin toiminta olisi pitänyt korjaantua.

Valmistajan lähettämä lista käytiin läpi yhdessä muiden projektissa työskentelevien henkilöiden kanssa. Valmistaja kiinnitti kuitenkin suurta huomiota robotin työkalulaipan ja voima-anturin välissä käytettyyn sovitteeseen, joka oli koneistettu sovitelaippa. Tämän kappaleen tilalla olisi pitänyt valmistajan sanojen mukaisesti käyttää valmistajan omaa standardisovitetta. Sovitteiden ollen lähes identtiset ei asiassa nähty ongelmaa. Kuitenkin voima-anturin työssä päätettiin suorittaa käyttöönoton osana tehty konfigurointi uudestaan. Konfiguraatiossa huomattiin, että kun sovitteen tyyppi muutetaan, muuttui myös voima-anturin koordinaatisto. Kun järjestelmä asetettiin käyttämään valmistajan valmistamaa sovitetta, voima-anturin koordinaatisto oli 180 astetta väärässä asennossa. Voima-anturin koordinaatisto luotiin manuaalisesti oikeaan asemaan asettaen koordinaatiston suunta oikein. Tämän jälkeen suoritettiin muu voima-anturin konfiguraatio loppuun ja

varmistettiin, että voimien suunnat olivat oikein. Konfiguraation jälkeen luotiin uusi ohjelma samalle ulkokehältään ympyrän muotoiselle kappaleelle ja suoritettiin ohjelma käyttäen contouring -funktiota. Ohjelman suorituksessa havaittiin, että voima pysyi vakiona koko työkierron ajan. Toimintaa kokeiltiin vielä aiemmin käytetyllä suorakulmion mallisella kappaleella ja ohjelman suorituksen jälkeen todettiin ohjelman toimivan odotetulla tavalla. Ratkaisuna ongelmaan oli oletuksena asetetun koordinaatiston kääntö.

Ratkaistaessa ongelmaa yritettiin voima-anturin epävakaaseen toimintaan vaikuttavat asiat selvittää. Voima-anturin toimintaan vaikutti muun muassa työkalun ominaisuudet. Työn aikana huomattiin, että mikäli työssä käytettiin joustavaa työkalua, jollaista esimerkiksi ensimmäisissä testeissä käytettiin, voima-anturin lukemat eivät olleet paikkaansa pitäviä. Myös voima-anturin päivitystaajuuden todettiin vaikuttavan merkittävästi toimintaan. Päivitystaajuus oli muutettavissa ja se tuli asettaa kappalekohtaisesti. Työssä käytetyissä kappaleissa oletuspäivitystaajuus oli sopiva työhön.

6 POHDINTA

Tässä kappaleessa käsitellään opinnäytetyön tuloksia ja sitä miten asetetut työn tavoitteet saavutettiin. Pohdinnassa mietitään myös sitä, miten työtä lähdetään jatkossa kehittämään ja mikä on robottisolun kehityksessä otettava huomioon tulosten seurauksena.

6.1 Tulokset

Voima-anturi saatiin työn lopuksi toimimaan halutulla tavalla. Voima-anturia pystyttiin käyttämään voiman hallinnassa sekä käyttämään sen ominaisuuksia kappaleen muodon seurannassa. Alussa suoritetut testit osoittivat, että voima-anturin käyttö ja sen käyttöönotto eivät kuitenkaan olleet täysin mutkattomia. Voima-anturin contouring -funktio on robottisolun tarkoitukseen varmasti oikea työkalu, mutta sitä tukevia ominaisuuksia tarvitaan työn laadun ja toimivuuden varmistamiseksi.

Ohjelmoinnin osuutta jouduttiin osittain karsimaan työn aikana. Tämä johtui voima-anturin käyttöönotossa ilmenneistä ongelmista, jotka toivat työlle odotettua enemmän kuormitusta. Ohjelmallisesti robotin ja PLC:n välinen kommunikointi saatiin kuitenkin ohjelmoitua siten, että se oli valmis opinnäytetyn jälkeistä työtä varten. Ohjelmoinnin ja automatisoinnin osuus jäi siis odotetusta tavoitteesta, mutta voima-anturin toiminnan katsottiin olevan tärkeämpi työn onnistumisen kannalta, joten automatisoinnin prioriteettia piti muista syistä laskea.

6.2 Jatkokehitys

Robottisolulla tehdään lisää testejä, joissa tullaan käyttämään työhön soveltuvia työkaluja sekä kappaleita, joita robottisolussa oli tarkoitus käyttää. Opinnäytetyön jälkeen kappaleen kuvausta konenäköä käyttäen muutetaan, mikä saattaa vaikuttaa myös voima-anturin käyttöön. Tarkoituksena on miettiä, onko työn kannalta järkevämpää kuvata kappale eri paikassa, jolloin kuvauksen valotus on helpompi suunnitella. Voima-anturille tehdään mahdollisimman automaattinen ohjelma, joka pystyy hyödyntämään kappaleen parametreja.

Robotin ohjelmaa kehitetään, jotta ohjelma saadaan kommunikoimaan robottisolun PLC:n kanssa. Ohjelmaan tulisi myös tehdä järjestelmien välinen käskytys, jonka avulla voitaisiin käyttöliittymän avulla hallita koko robottisolua ja sen toimintoja. Myös käyttöliittymä on tarkoitus tehdä työssä tehdylle robottisolulle, jota käyttämällä muun muassa kappaleiden parametrit saadaan syötettyä yksilöllisesti järjestelmään.

LÄHTEET

JTA Connection. WWW-sivu. Luettu 11.3.2019. http://www.jtaconnection.fi/

Beckhoff. CX9020. Esite. Luettu 20.1.2019. https://www.beckhoff.com/english.asp?embedded_pc/cx9020.htm

Fanuc. WWW-sivu. M-20iB/25. Luettu 12.3.2019 http://www.fanuc.eu/fi/en/robots/robot-filter-page/m-20-series/m-20ib-25

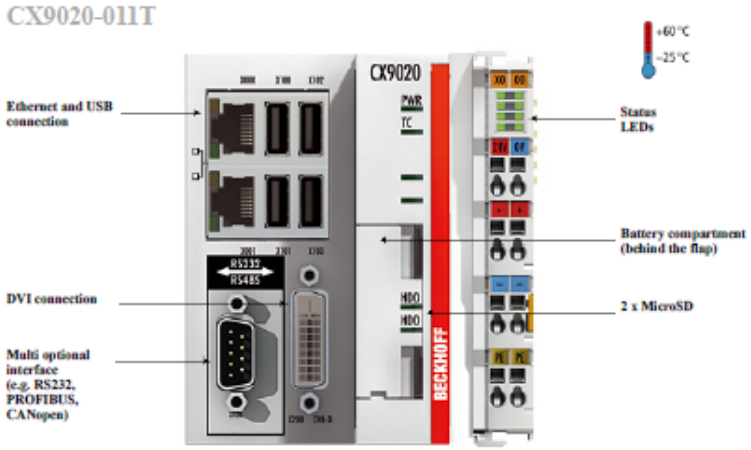
Fanuc. 2019. Tekniset tiedot -dokumentti. M-20iB/25. Luettu 12.3.2019

Fanuc. 2019. Tekniset tiedot -dokumentti. FS-40iA Force Sensor. Luettu 12.3.2019

B&R. 2019. Tekniset tiedot -dokumentti. PPC2100. Luettu 12.3.2019

LIITTEET

Liite 1. Beckhoff CX9020 esite



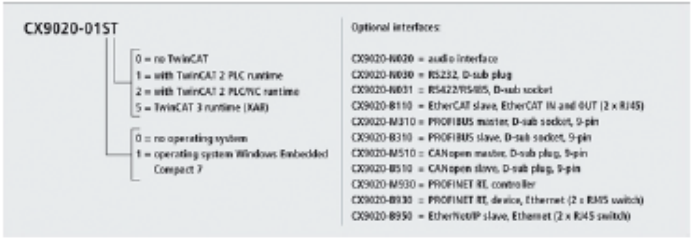
CX9020 | Basic CPU module

The CX9020 is a compact, DIN rail-mountable Ethernet control system with 1 GHz ARM Cortex™-A8 CPU. The connection for the Beckhoff I/O systems is directly integrated into the CPU module. The unit offers automatic bus system identification (K-bus or E-bus) and independently switches in the corresponding mode. The CX9020 comprises the CPU with two microSD card slots, the internal RAM and 128 kB NOVRAM as non-volatile memory. The basic configuration also includes two switched Ethernet RJ45 interfaces, four USB 2.0 interfaces and a DVI-D interface. The RJ45 interfaces are connected to an internal switch and offer a simple option for creating a line topology without the need for additional Ethernet switches. The operating system is Microsoft Windows Embedded Compact 7. TwinCAT automation software transforms a CX9020 system into a powerful PLC and Motion Control system that can be operated with or without visualisation. Optionally, the unit can be ordered with a fieldbus, serial or audio interface.

The extended operating temperature range between -25 and +60 °C enables application in climatically demanding situations.

Order identifier

The order identifier is derived as follows:



Technical data	CX9020
Processor	ARM Cortex™-A8, 1 GHz
Number of cores	1
Flash memory	2 x slot for microSD card, 512 MB included (expandable)
Main memory	1 GB DDR3 RAM (not expandable)
Persistent memory	128 KB NOVRAM integrated
Interfaces	2 x RJ45 10/100 Mbit/s (internal switch), 1 x DVI-D, 4 x USB 2.0, 1 x optional interface
Diagnostics LED	1 x power, 1 x TC status, 2 x flash access, 2 x bus status
Clock	internal battery-backed clock for

	time and date (battery exchangeable)
Operating system	Microsoft Windows Embedded Compact 7, English
Control software	TwinCAT 2 runtime TwinCAT 3 runtime (XAR)
I/O connection	E-bus or K-bus, automatic recognition
Power supply	24 V DC (-15 %/+20 %)
Current supply E-bus/K-bus	2 A
Max. power loss	5 W (including the system interfaces)
Dimensions (W x H x D)	84 mm x 99 mm x 91 mm
Weight	approx. 590 g
Operating/storage temperature	-25…+60 °C/-40…+85 °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protection class	IP 20
Approvals/Markings	CE, UL, ATEX, GL, IECEx
Ex-Marking	II 3 G Ex nA IIC T4 Gc II 3 D Ex tc IIIC T135 °C Dc Ex nA IIC T4 Gc Ex tc IIIC T135 °C Dc
TC3 performance class	Economy Plus (30); please see here for an overview of all the TwinCAT 3 performance classes

Ordering information	no operating system	Windows Embedded Compact 7	no TwinCAT	TwinCAT 2 PLC runtime	TwinCAT 2 NC PTP runtime	TwinCAT 3 runtime (XAR)
CX9020-0100	x	–	x	–	–	–
CX9020-0110	–	x	x	–	–	–
CX9020-0111	–	x	–	x	–	–
CX9020-0112	–	x	–	–	x	–
CX9020-0115	–	x	–	–	–	x

Accessories	
CX9020-U900	internal, capacitive 1-second UPS to ensure secure backup of persistent application data on the microSD card
CX2900-0107	Device modification for CX5120, CX5130, CX5140 and CX9020 Embedded PCs according to the requirements for ATEX and IECEx certification. The modification is mandatory for the usage of CX5120, CX5130, CX5140 and CX9020 in hazardous areas, Zone 2/22. It includes the modification and repositioning of the device label as well as a mounting bracket installed ex works for mechanical locking of the connectors. Product labeling: ATEX: II 3 G Ex nA IIC T4 Gc and II 3 D Ex tc IIIC T135 °C Dc IECEx: Ex nA IIC T4 Gc and Ex tc IIIC T135 °C Dc Read the device documentation for use in hazardous areas carefully.

Optional interfaces

CX9020-N020	audio interface, 3 x 3.5 mm jack sockets, Line In, Mic In, Line Out
CX9020-N030	RS232 interface, D-sub plug, 9-pin
CX9020-N031	RS485 interface, D-sub socket, 9-pin, configuration as an end point, without echo, termination on
CX9020-N031-0001	RS485 interface, D-sub socket, 9-pin, configuration as an end point, with echo, termination on
CX9020-N031-0002	RS485 interface, D-sub socket, 9-pin, configuration as drop point, without echo, termination off
CX9020-N031-0003	RS485 interface, D-sub socket, 9-pin, configuration as drop point, with echo, termination off
CX9020-N031-0004	RS422 interface, D-sub socket, 9-pin, configuration as full duplex end point, termination on
CX9020-B110	EtherCAT slave interface, EtherCAT IN and OUT (2 x RJ45)
CX9020-M310	PROFIBUS master interface, D-sub socket, 9-pin
CX9020-B310	PROFIBUS slave interface, D-sub socket, 9-pin
CX9020-M510	CANopen master interface, D-sub plug, 9-pin
CX9020-B510	CANopen slave interface, D-sub plug, 9-pin
CX9020-M930	PROFINET RT, controller interface, Ethernet (2 x RJ45)
CX9020-B930	PROFINET RT, device interface, Ethernet (2 x RJ45 switched)
CX9020-B950	EtherNet/IP slave interface, Ethernet (2 x RJ45 switched)
Product announcement	**CX9020-011T, CX9020-N0xx, CX9020-x310, CX9020-x510, CX9020-x930, CX9020-B110: available** CX9020-B950: estimated market release on request

Liite 2. B&R PPC 2100

Preliminary material number:

Automation Panel 24.0" Full HD TFT
- 1920 x 1080 Pixel (16:9)
- Multi-touch (projected capacitive)
- Support arm mounted
- Landscape format
- Extention Option
- For PPC2100/PPC2200/link modules

AP5000 Swing arm mounting unit

AP5000 Heatpipe
- For PPC2100 (5PPC2100.BYxx-002)
- For swing arm mounting unit

AP5000 Flange
- Swing Arm Flange Adapter
- For Rittal
- For Console Swing Arm

AP5000 Swing Arm Extension Option
- Expansion unit
- 1x Emergency stop
- 3x pushbuttons (red, green, blue)
- 1x Key switch
- 1x USB interface on front
- For 5AP5230.240C-000 panel

PPC2100 System unit
- Intel Atom E3827 1.75 GHz
- Dual core
- 4 Gbyte SDRAM
- For Automation Panel 5000

CFast 32 GByte MLC

Connector 24 VDC
- 3-pin female
- Cage clamps 3.31 mm²

Windows 10 IoT Enterprise 2016 LTSB
- 64-bit
- Entry
- Multilingual

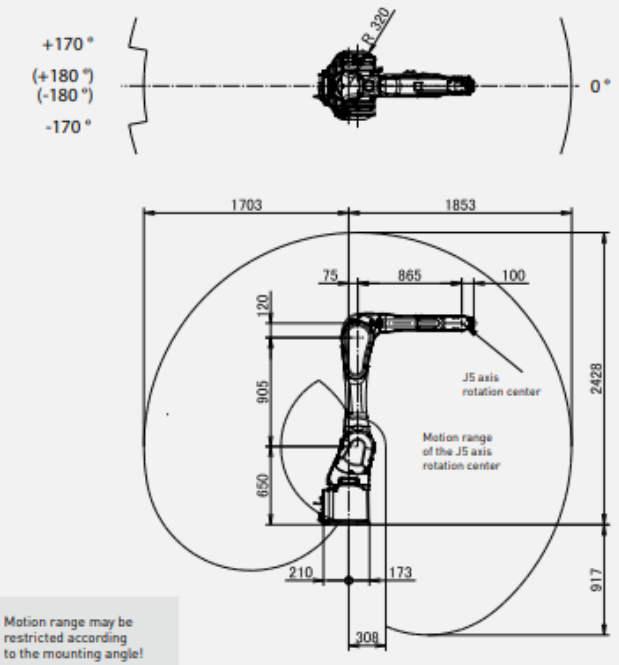
M-20*i*B/25

Max. load capacity at wrist: **25 kg**

Max. reach: **1853 mm**

Controlled axes	Repeatability (mm)	Mechanical weight (kg)	Motion range (°)						Maximum speed (°/s)						J4 Moment/ Inertia (Nm/kgm²)	J5 Moment/ Inertia (Nm/kgm²)	J6 Moment/ Inertia (Nm/kgm²)
			J1	J2	J3	J4	J5	J6	J1	J2	J3	J4	J5	J6			
6	± 0.02*	210	340(360)	240	303	400	290	540	205	205	260	415	415	880	51/2.2	51/2.2	31/1.2

Working range



Motion range may be restricted according to the mounting angle!

Robot	**M-20*i*B/25**
Robot footprint [mm]	383 x 343
Mounting position Floor	•
Mounting position Upside down	•
Mounting position Angle	•
Controller	**R30*i*B**
Open air cabinet	-
Mate cabinet	o
A-cabinet	•
B-cabinet	o
*i*Pendant Touch	•
Electrical connections	
Voltage 50/60Hz 3phase [V]	380-575
Voltage 50/60Hz 1phase [V]	-
Average power consumption [kW]	1
Integrated services	
Integrated signals on upper arm In/Out	8/2
Integrated air supply	1
Environment	
Acoustic noise level [dB]	< 70
Ambient temperature [° C]	0-45
Protection	
Body standard/optional	IP67
Wrist & J3 arm standard/optional	IP67

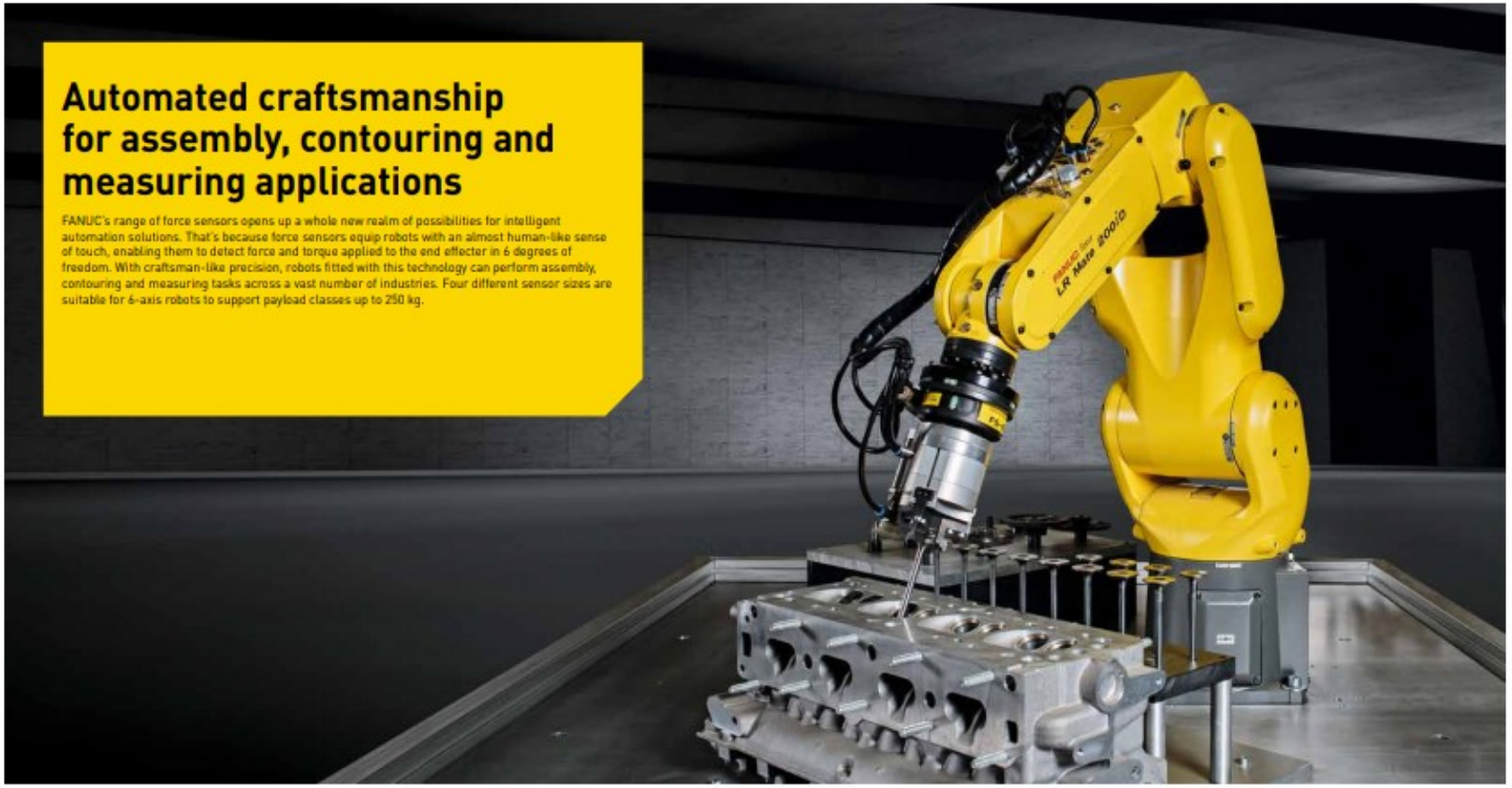
● standard o on request - not available ( ) with hardware and/or software option *Based on ISO9283

 Technical information subject to changes without prior notice. All rights reserved. ©2017 FANUC Europe Corporation

Liite 4. Fanuc FS-40iA Force Sensor

1 (7)

2 (7)



Automated craftsmanship for assembly, contouring and measuring applications

FANUC's range of force sensors opens up a whole new realm of possibilities for intelligent automation solutions. That's because force sensors equip robots with an almost human-like sense of touch, enabling them to detect force and torque applied to the end effecter in 6 degrees of freedom. With craftsman-like precision, robots fitted with this technology can perform assembly, contouring and measuring tasks across a vast number of industries. Four different sensor sizes are suitable for 6-axis robots to support payload classes up to 250 kg.

More productivity

- complete integration into the robot for faster processing
- supports practice-proven standard functions
- robust hardware and intelligent software for reduced cycle times – all 100% FANUC
- combines Force Control and iRVision for unique functionality

Faster setups and maximum uptime

- fast setups thanks to complete integration into the robot (no interface to external devices)
- ready to use software library of addition features
- seamless integration with FANUC vision systems
- fewer parts and FANUC's reliability for maximum uptime

Quality control

- complete production data tracking and logging
- high precision on assembly and contouring applications
- automated force operation ensures consistent quality for process
- full range of measurement options

Opening up your options

- assembling – high-precision face matching, positioning, fitting, feeding, insertion up to clearance H7
- contouring – accurate deburring, grinding, sanding and polishing
- measuring – versatile mass measurement, weight and gravity centre calculation

Deburring and grinding

Deburr and grind parts precisely by compensating for dimensional deviations in castings and ensuring the tool remains on the contour despite changes to the effective path.

Polishing

Polish metal parts by measuring the force applied and controlling the robot's motion path in real time to keep the tool pressing with constant force to the part.

Assembling

Assemble gearboxes and other mechanical devices using coordinated robots to hold the base unit and phase match cogs and bearings.

Fitting and inserting

Fit parts and insert workpieces vertically or horizontally into a chuck or tool holder to very exact tolerances without jamming.

Contact us to find the perfect solution for your industry: **www.fanuc.eu**

Constant Push
This enables the robot to push with constant force in one direction. After it reaches a predefined "Contact Force Threshold", the robot starts to push with "Pushing Force" for a specified time.

Face Match
On assembly and contouring operations, this provides the robot with the ability to align and match the faces of components or workpieces.

Hole Search
Enables the robot to search for a hole by moving the shaft or object to be inserted at right angles to the hole. This can then be followed by an insertion.

Shaft insert
Allows loose fitting by aligning the angle and position of a shaft with a hole before insertion.

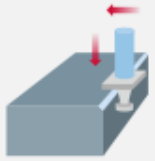
Phase match
This function performs phase matching of teeth, such as key shaft insertion and gear engagement. It aligns gears or key shafts by rotating and meshing them before final insertion is executed.

Fixture Mount Force Sensor

Ideally suited to precise fitting, phase matching and contouring operations, this allows the force sensor to be mounted on a table. Table mounting the sensor also reduces the weight of the end effector and enables the robot to carry different tools.

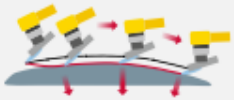
TP Program Auto Generation

On processes requiring accurate contoured edges, this feature records the path of the workpiece and automatically generates a TP program.



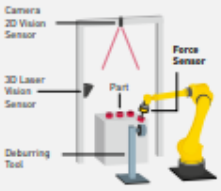
Two-Direction Push

On deburring and contouring operations this enables the robot to push down and in at the same time.



Contouring

On sanding, grinding and polishing operations involving uneven or irregular surfaces, this feature measures the force applied to the tool and surface of a workpiece and applies constant force. The robot follows the uneven surface even if it differs from the surface it has been taught to follow.

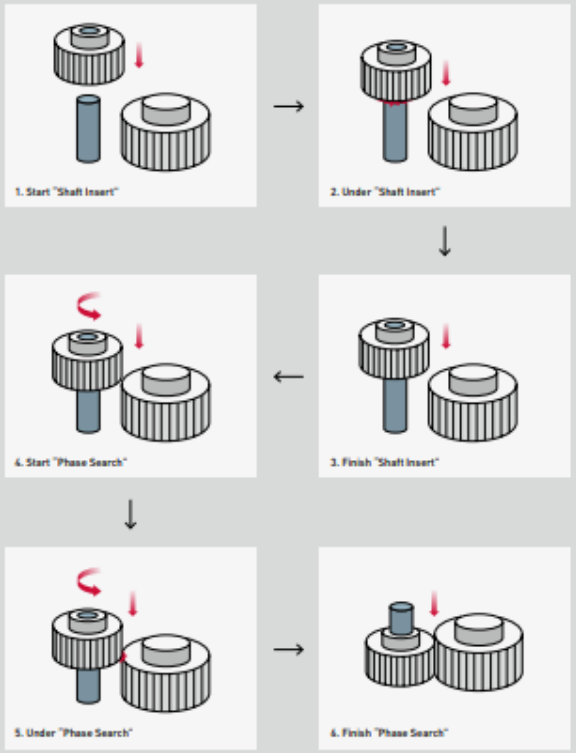


Deburring Path Auto Generation using iRVision

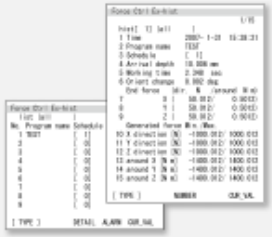
Automatically generates a TP program using positional information acquired with FANUC iRVision's Image to Points feature. With absolutely no need to teach, deburring can now be carried out using the optional Force Control Contouring function.

Combining features for more versatility

Force control commands are connected on a master/slave basis for easy execution



Complete quality control



Execution History Screen

To provide a complete overview of your processes, the Execution History Screen displays both key and detailed data on different screens. Parameters such as insertion depth, duration of insertion and force applied can all be checked and recorded. (image?)

Data Viewer

In order to provide a process archive and make adjusting and fine tuning Force Control easy, this function logs force sensor data and allows you to output it graphically.

Force Control features

- 5 different types of sensors available: 15, 40, 100 and 250 kg
- Easy to use predefined force functionalities:
  - Insert functions (insert shafts, bearings etc. for almost every conceivable part)
  - Contouring function (follow a contour and apply defined force to the surface)
  - Search functions (find holes, grooves or clutch disks)
  - Match function (match gripper/tool to surface orientation)
  - Push function (for adhesive bonding of retainers in an airplane body for example)
- Data logging with visualisation as Force graph
- Tool weight and centre of gravity calculation
- TP auto generation by tracing path when contouring
- The sensor can be used in 2 ways:
  - robot attached (as in this brochure pictures)
  - fix mounted (on table or jig)

Technical overview grid

	DOF	Dim Ø (mm)	Weight (kg)	Rated load		Static overload tolerance		Resolution		Accuracy [%]
				Fx, Fy, Fz [N / kgf]	Mx, My, Mz [Nm / kgfcm]	Fx, Fy, Fz [N / kgf]	Mx, My, Mz [Nm / kgfcm]	Fx, Fy, Fz [N / kgf]	Mx, My, Mz [Nm / kgfcm]	
FS-15uAe										
	3 axis	90 x 36	0.31	147 / 15	11.8 / 120	1570 / 160	125 / 1280	0.39 / 40	0.016 / 0.16	less than 3
FS-15uA										
	6 axis	94 x 43	0.57	147 / 15	11.8 / 120	1570 / 160	125 / 1280	0.39 / 0.04	0.016 / 0.16	less than 2
FS-40uA										
	6 axis	105 x 47	0.87	392 / 40	39.2 / 400	3920 / 400	392 / 4000	1.0 / 0.1	0.029 / 0.29	less than 2
FS-100iA										
	6 axis	155 x 59	3.2	980 / 100	156 / 1600	9800 / 1000	1560 / 16000	2.0 / 0.2	0.08 / 0.8	less than 2
FS-250iA										
	6 axis	198 x 85	6.9	2500 / 255	500 / 5100	25000 / 2550	5000 / 51000	4.9 / 0.5	0.25 / 2.5	less than 2

MANUFACTURED EFFICIENCY
MANUFACTURED EFFICIENCY: 5 PRODUCT GROUPS – ONE COMMON SERVO AND CONTROL PLATFORM
FA
ROBOTS
ROBOCUT
ROBODRILL
ROBOSHOT
WWW.FANUC.EU

Liite 5. Pushing_Dir_Auto_Change_2 – ohjelma


```
--------------------------------------------------
PROGRAM Pushing_Dir_Auto_Change_2
--------------------------------------------------

%NOLOCKGROUP
%DEFGROUP = 1
--------------------------------------------------
VAR
--------------------------------------------------

CurrentPos:             XYZWPR
NextPos:                XYZWPR
PreviousPos:                    XYZWPR
Kiertokulma:                    XYZWPR

Kierros:                BOOLEAN

status:                 INTEGER
stat:                   INTEGER
int_val:                INTEGER
Nollaus:                INTEGER

A_X:                    REAL
A_Y:                    REAL
B_X:                    REAL
B_Y:                    REAL
A:                      REAL
B:                      REAL
AB:                     REAL
cosin:                  REAL
alfa:                   REAL
real_val:               REAL
kulma:                  REAL
--------------------------------------------------
BEGIN
--------------------------------------------------


                        --Kirjoitetaan tarvittavat positiot muistiin
                        CurrentPos = GET_POS_REG(7, status)
                        PreviousPos = GET_POS_REG(8, status)
                        NextPos = GET_POS_REG(9, status)

                        --Luoetaan rekisteristä onko kyseessä ensimmäinen
liikekäsky (P1)
                        GET_REG(50,FALSE,int_val,real_val,status)
```

```
                --Jos ensimmäinen liikekäsky, niin A.y =B.y & A.x = 0
--              IF int_val = 0 THEN
--              A_X = 0
--              B_X = NextPos.x - CurrentPos.x
--              B_Y = NextPos.y - CurrentPos.y
--              A_Y = B_Y
--              ELSE
                --Asetetaan laskennallisille vektoreille X ja Y arvot
                edellisestä-, nykyisestä- ja seuraavasta pisteestä
                A_X = CurrentPos.x - PreviousPos.x
                A_Y = CurrentPos.y - PreviousPos.y
                B_X = NextPos.x - CurrentPos.x
                B_Y = NextPos.y - CurrentPos.y
--              ENDIF

                --Määritellään vektorien pituudet
                A = SQRT(A_X*A_X+A_Y*A_Y);
                B = SQRT(B_X*B_X+B_Y*B_Y);

                --Lasketaan vektorien skalaaritulo
                AB = A_X*B_X+A_Y*B_Y;

                --Lasketaan kulma
                cosin=AB/(A*B);

                alfa = ACOS(cosin);

                kulma = Kiertokulma.r+alfa



                --Asetellaan positiolle "kiertokulma" Z-akselin ympäri
kieryminen, jota käytetää offsettina
                Kiertokulma.x= 0.0
                Kiertokulma.y= 0.0
                Kiertokulma.z= 0.0
                Kiertokulma.w= 0.0
                Kiertokulma.p= 0.0
                Kiertokulma.r= Kiertokulma.r+alfa
                Kiertokulma.config_data.cfg_turn_no1=0
                Kiertokulma.config_data.cfg_turn_no2=0
                Kiertokulma.config_data.cfg_turn_no3=0
                Kiertokulma.config_data.CFG_FLIP=FALSE
                Kiertokulma.config_data.CFG_LEFT=FALSE
                Kiertokulma.config_data.CFG_UP=TRUE
                Kiertokulma.config_data.CFG_FRONT=TRUE

                SET_POS_REG(10, Kiertokulma, stat,1)
```

```
                    --Kirjoitetaan rekisteriin, että seuraavalla luennalla ei
ole enää kyseessä ensimmäinen piste
                    SET_INT_REG(50,1,status)

                    SET_REAL_REG(70,kulma,status)

                    --NOLLAUS

                    GET_REG(51,FALSE,Nollaus,real_val,status)

                    IF Nollaus = 1 THEN
                              SET_INT_REG(50,0,status)
                              A_X = 0
                              B_X = 0
                              B_Y = 0
                              A_Y = 0
                              SET_INT_REG(51,0,status)

                    Kiertokulma.x= 0.0
                    Kiertokulma.y= 0.0
                    Kiertokulma.z= 0.0
                    Kiertokulma.w= 0.0
                    Kiertokulma.p= 0.0
                    Kiertokulma.r= 0.0
                    Kiertokulma.config_data.cfg_turn_no1=0
                    Kiertokulma.config_data.cfg_turn_no2=0
                    Kiertokulma.config_data.cfg_turn_no3=0
                    Kiertokulma.config_data.CFG_FLIP=FALSE
                    Kiertokulma.config_data.CFG_LEFT=FALSE
                    Kiertokulma.config_data.CFG_UP=TRUE
                    Kiertokulma.config_data.CFG_FRONT=TRUE

                    SET_POS_REG(10, Kiertokulma, stat,1)

                    ENDIF
          ------------------------------------------------
          END Pushing_Dir_Auto_Change_2
          ------------------------------------------------
```