



Joustavan hitsaussolun ohjelmointi

Aapo Eränummi

OPINNÄYTETYÖ
Toukokuu 2022

Konetekniikka
Koneautomaatio

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Konetekniikka
Koneautomaatio

ERÄNUMMI, AAPO:
Joustavan hitsaussolun ohjelmointi

Opinnäytetyö 69 sivua, joista liitteitä 9 sivua
Toukokuu 2022

Opinnäytetyön tarkoituksena oli suunnitella joustavan robottihitsaussolun ohjelmointi ja toteutus Emeca Oy:n osatuotantoon. Joustavan ohjelmoinnin avulla voidaan mahdollistaa nopea tuotannon mukautuminen asiakkaan tarpeeseen. Yritys oli hankkinut kaksi ABB-teollisuusrobottia, joilla oli tarkoitus aloittaa paalujatkosten jatkoslaatikoiden valmistus. Työn tavoitteena oli selvittää hitsaussolun joustavan ohjelmoinnin mahdollisuutta yrityksen tuotannossa ja soveltaa saatuja tuloksia käytäntöön. Opinnäytetyö rajattiin hitsaussolun mallin sekä robottiohjelmien suunnitteluun ja testaukseen. Ohjelmien käyttöönotto ei kuulunut opinnäytetyöhön.

Opinnäytetyö sisältää teoriaosuuden sekä käytännön suunnittelun. Teoriaosuuksessa käsitellään teollisuusrobottien käyttökohteita, robottihitsausta, ABB-teollisuusrobottien ohjelmointia ja erilaisia ohjelmointitapoja. Käytännön osuus sisältää hitsaussolun mallintamisen ja robottiohjelmien suunnittelun sekä testauksen RobotStudio-ohjelmistolla. Opinnäytetyössä lähdemateriaalina käytettiin verkkolähteitä, dokumentteja ja manuaaleja.

Opinnäytetyö toteutettiin toiminnallisena eli projektityyppisenä työnä, jonka tuloksena saatiin suunniteltua robottihitsaussolun malli ja sen joustava ohjelmointi. Opinnäytetyöstä saatujen tulosten perusteella voidaan valmistaa joustava robottihitsaussolu Emeca Oy:n osatuotantoon. Tuloksia voidaan hyödyntää nykyisen tuotannon kehittämiseen sekä uusien robottisolujen suunnitteluun. Haasteena työssä oli tiedon löytäminen joustavasta ohjelmoinnista ja sen mahdollisuuksista teollisuuden robottisovelluksissa.

Asiasanat: robottisolu, joustava ohjelmointi, teollisuusrobotti, ABB-robotti

ABSTRACT

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

ERÄNUMMI, AAPO:
Programming of a Flexible Welding Cell

Bachelor's thesis 69 pages, appendices 9 pages
May 2022

The purpose of this thesis was to design the programming and implementation of a flexible welding cell for Emeca Oy's partial production. Flexible programming allows production to adapt quickly to customers' needs without long production breaks. The company had bought two ABB industrial robots for pile joint part production. The aim was to research possibilities of a flexible robot programming in Emeca's production and apply the results in practice. The subject area of the thesis was limited to designing the welding cell and creating and testing the robot programs. Deployment of the programs was not included in this thesis.

This thesis includes a theoretical and a practical part. The theoretical part goes through industrial robots in general, robot welding, programming of ABB industrial robots and different types of programming methods. The practical part includes the designing of the welding cell and robot programs as well as testing the programs with RobotStudio software. Websites, documents, and manuals were used as a source material.

This thesis was carried out as a practice-based study and as a result of the project the welding cell model was designed, and the flexible robot programs were created. With the results of this thesis a flexible robot welding cell can be implemented in Emeca Oy's partial production. The results of this thesis can be used to develop existing production cells and to create new ones. The main challenge was to find information about flexible programming in industrial applications.

Key words: robot cell, flexible programming, industrial robot, ABB-robot

SISÄLLYS

1	JOHDANTO	7
2	EMECA OY	8
	2.1 Yritys	8
	2.2 Tuotteet	8
3	TEOLLISUUSROBOTIT	10
	3.1 Teollisuusrobotin määritelmä	10
	3.2 Teollisuusrobottien asennukset vuosittain	10
	3.3 Robottityypit	11
	3.4 Robottihitsaus	16
	3.5 Sykli aika	17
	3.6 Robottiturvallisuus	18
4	ABB TEOLLISUUSROBOTIT	20
	4.1 ABB robottivalmistajana	20
	4.2 ABB robottijärjestelmä	21
	4.2.1 IRC5-Ohjausjärjestelmä	21
	4.2.2 FlexPendant-käsiohjain	22
	4.3 Koordinaatistot	23
	4.4 Ohjelmointi	25
5	TEOLLISUUSROBOTTIEN OHJELMOINTITAVAT	30
	5.1 Opettamalla ohjelmointi	30
	5.2 Johdattamalla ohjelmointi	31
	5.3 Offline-ohjelmointi	32
	5.4 Joustava ohjelmointi	33
6	JOUSTAVAN HITAUSSELUN SUUNNITTELU	35
	6.1 Lähtötilanne	35
	6.2 Hitsattava tuote	38
	6.3 Hitsaussolun malli	38
	6.3.1 Hitsaussolun vaatimukset	39
	6.3.2 Kaulustaivutin	41
	6.3.3 Hitsausjigit	41
	6.3.4 Valmistuotekuljetin	43
	6.3.5 Hitsaussolun kokoonpano	43
	6.4 Hitsaussolun turvallistaminen	44
7	JOUSTAVAN HITAUSSELUN OHJELMOINTI JA TESTAUS	46
	7.1 Ohjelmoinnin suunnittelu	46
	7.1.1 Ohjelmoinnin vaatimukset	46

7.1.2 Ohjelmointitavan valinta	47
7.1.3 Työkiertojen suunnittelu.....	47
7.2 Ohjelmointi RobotStudiolla.....	48
7.3 Simulointi ja testaus RobotStudiolla	52
7.3.1 Työkiertojen simulointi ja testaus.....	53
7.3.2 Jatkoslaatikoiden valmistus	54
7.3.3 Ohjelmien käyttöönotto.....	55
8 POHDINTA	56
LÄHTEET	57
LIITTEET	61
Liite 1. Kappaleenkäsittelyrobotin ohjelmat.....	61
Liite 2. Hitsausrobotin ohjelmat.....	66
Liite 3. Hitsaussolun työkierron simulointi	68
Liite 4. Joustavan hitsausohjelman simulointi	69

LYHENTEET JA TERMIT

IFR	International Federation of Robotics
SCARA	Selective Compliance Articulated Robot Arm
COBOT	Collaborative Robot, yhteistyörobotti
IRC5	ABB:n viidennen sukupolven teollisuusrobottien ohjausjärjestelmä
RAPID	ABB:n kehittämä teollisuusrobottien ohjelmointikieli
TASK	RAPID-sovelluksen yksittäinen tehtävä
TCPF	Tool Center Point Frame, työkalukoordinaatisto
TCP	Tool Center Point, työkalupiste
I/O	Tulo- ja lähtötieto (input/output)
BOOL	Looginen tuusarvomuuuttuja
NUM	Numeerinen muuttuja
DNUM	Tarkka numeerinen muuttuja
STRING	Merkkijono
VAR	Muuttuva data
CONST	Vakio data
PERS	Pysyvä Data

1 JOHDANTO

Opinnäytetyön toimeksiantaja oli suomalainen paalutarvikkeiden valmistaja ja toimittaja Emeca Oy. Opinnäytetyön aihe syntyi yrityksen hankittua kahden ABB IRB 2400/10 -robotin järjestelmä, joka suunniteltiin käyttöönotettavan yrityksen tuotannossa. Roboteilla oli tarkoitus aloittaa paalujatkosten jatkoslaatikoiden valmistus. Opinnäytetyön tarkoituksena oli suunnitella laatikoiden valmistukseen soveltuva joustava robottihitsaussolu ja sen ohjelmointi.

Työn tavoitteena oli selvittää, kuinka hitsaussolu ja sen ohjelmointi voidaan toteuttaa mahdollisimman joustavaksi. Tavoitteena oli suunnitella erikokoisille jatkoslaatikoille mukautuvat robottiohjelmat sekä niiden testaus RobotStudiolla. Joustavan robottihitsaussolun avulla voidaan valmistaa tuotteita nopealla mukautumiskyvyllä ilman uusien ohjelmien luomista. Tämä mahdollistaa pienienkin erien valmistuksen asiakkaan tarpeeseen menettämättä tuotannon tehokkuutta. Opinnäytetyön aihe rajattiin hitsaussolun mallin ja robottiohjelmien suunnitteluun sekä testaukseen. Ohjelmien käyttöönotto ei kuulunut opinnäytetyöhön.

Projektin suunnittelulle asetettiin lisäksi seuraavat tavoitteet. Jigien ja asetusten säätäminen sekä muokkaaminen tuli sujua mahdollisimman joustavasti ja nopeasti, hitsaussolun ja robottiohjelmien tuli olla yksinkertaisia sekä helppoja muokata. Robottien ohjelmointi täytyi suunnitella selkeäksi ja käyttäjäystävälliseksi, koska lukuisten monimutkaisten ohjelmien ylläpitäminen ja päivittäminen olisi haastavaa, eikä vastaa nykypäivän joustavaa tuotantoa.

2 EMECA OY

2.1 Yritys

Emeca Oy on vuonna 2001 perustettu suomalainen paalutarvikkeiden valmistaja ja toimittaja Säkylän Köyliössä. Tuotevalikoimaan kuuluu paalujatkokset, kalliojärjet ja useita erilaisia paalutuotteita sekä työkaluja niin valutyöhön, kuin paalujen asennukseenkin. Emecan vahvuuksia ovat laadukkaat, kestävät ja monipuoliset tuotteet. Lisäksi vahvuuksiin kuuluu jatkuva ja aktiivinen tuotekehitys, moderni tuotantolaitos sekä osaava henkilökunta. (Emeca 2021d.)

Emeca toimittaa tuotteitaan kotimaan lisäksi useisiin maihin ympäri maailmaa. Yhdysvalloissa toimiva Emeca/Spe USA vastaa tuotteiden valmistuksesta Amerikan markkinoille (Emeca 2021.) Yrityksessä työskentelee noin 16 henkilöä ja liikevaihto on noin 3–4 miljoonaa euroa. (Kauppalehti 2021).

2.2 Tuotteet

Emeca Oy:n päätuotteita ovat teräsbetonipaaluihin asennettavat paalujatkokset ja kalliojärjet. Yrityksen tuotteet ja tuotantolinjat ovat korkeasti koulutetun ja luovan henkilökunnan suunnitteleimia ja toteuttamia. Tuotteet valmistetaan pitkälle automatisoidussa tehtaassa Köyliössä, jossa teollisuusroboteilla varustetut robotisolut hitsaavat ja kokoonpaneavat suurimman osan Emecan tuotteista. (Emeca 2021c.)

Paalujatkokset

Emecan paalujatkokset (kuva 1) ovat vankkoja ja helposti asennettavia. Niitä tarvitaan kohteissa, joissa vaadittavan pituista teräsbetonipaalua ei voida valaa tai kuljettaa yksimittaisena. Emeca-paalujatkokset siirtävät kaiken paalulle aiheutuvan kuorman osapaalulta toiselle heikentämättä paalun kapasiteettia. (Emeca 2021a.)



KUVA 1. Emeca-Paalujatkos (Emeca 2021a.)

Paalukärjet

Emeca-paalukärjen (kuva 2) tehtävä on keskittää paaluun kohdistuvat voimat ja suojata paalun päätä rikkoutumiselta. Kun betonipaalu lyödään vinoon kalliopintaan, kiviseen tai lohkareiseen maakerrokseen, se varustetaan aina kalliojärjellä. (Emeca 2020.)



KUVA 2. Emeca-Kalliojärki (Emeca 2021b.)

3 TEOLLISUUSROBOTIT

3.1 Teollisuusrobotin määritelmä

Robotit toimivat nykyään teollisuuden kaikilla osa-alueilla. Teollisuusrobotit koostavat elektronisia laitteita, pakkaavat elintarvikkeita, leikkaavat metallia ja hitaavat autoja. Useat yritykset käyttävät teollisuusrobotteja säästääkseen rahaa, aikaa ja materiaaleja parantaakseen tuotannon tehokkuutta sekä tuotteiden laatua. Robottien erinomainen tarkkuus, kestävyys ja toistettavuus mahdollistaa työvaiheiden tekemisen, jotka ovat ihmiselle liian yksitoikkoisia, raskaita tai vaarallisia. (Robots 2021c.)

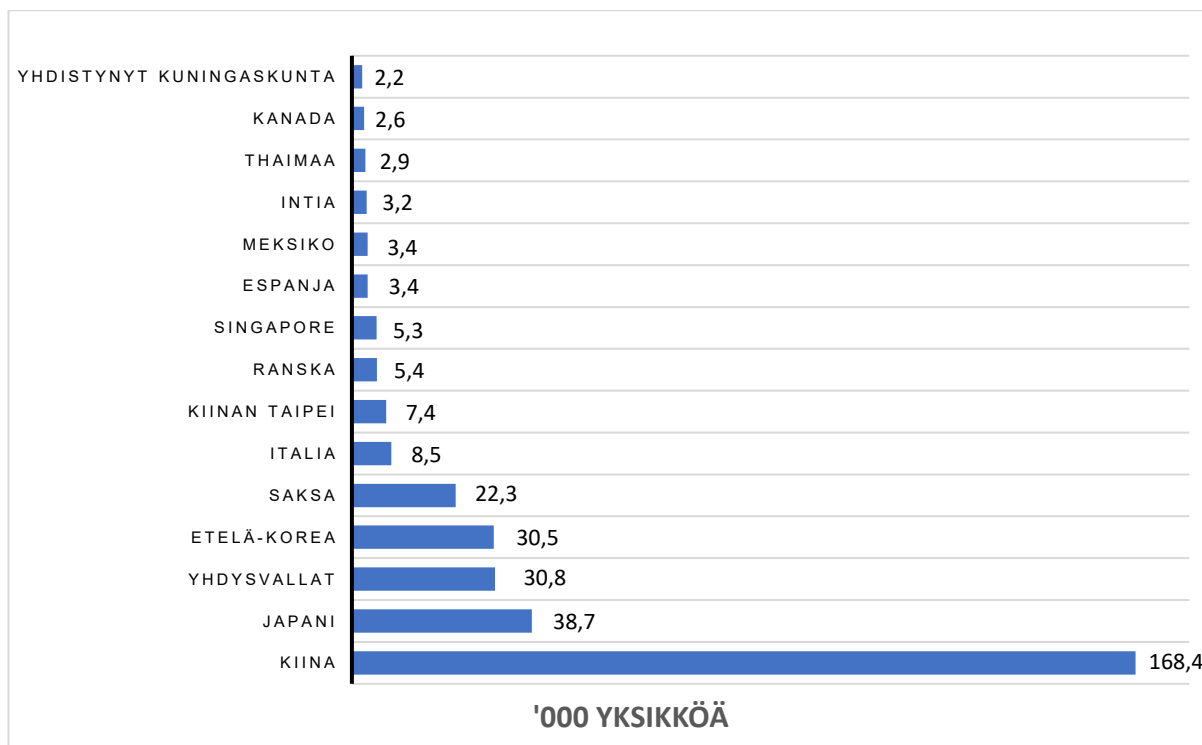
Teollisuusrobotti terminä perustuu kansainvälisen standardointijärjestön määritelmään ISO 8373. Se tarkoittaa automaattisesti ohjattua, uudelleenohjelmoitavaa monikäyttöistä manipulaattoria, joka on ohjelmoitavissa kolmella tai useammalla akselilla. Teollisuusrobotti voi olla joko kiinteä tai liikuteltava teollisuusautomaatio sovelluksissa käytettävä laite. (SFS ISO 8373 2021, 2.)

Teollisuusrobotin akseleita ohjataan servomootoreilla, joiden käyttö robotiikassa on yleistä niiden pienen koon, suuren väännön ja korkea tarkkuuden vuoksi. Servomootori tuottaa halutun vääntömomentin ja nopeuden, syötetyn virran ja jännitteen perusteella. Tavallisesti moottorin perässä oleva takaisinkytkentälaitte esimerkiksi pulssianturi palauttaa akselin sijaintitietoa servo-ohjaimelle, joka vertaa sitä moottorin tavoiteltuun kohdesijaintiin ja pyrkii korjaamaan virheen reaaliajassa syöttämällä moottorille sopivan määrän virtaa ja jännitettä. (How to Mechatronics 2021.)

3.2 Teollisuusrobottien asennukset vuosittain

Industrial Federation of Robotics (IFR) kerää vuosittain tilastotietoja teollisuusrobottien asennuksista maailmalla. IFR:n mukaan vuonna 2020 toimitettiin yhteensä 384 000 teollisuusrobottia, joista hieman alle puolet käyttöönotettiin Kiinassa (kuvio 1, muokattu). Kiinassa käyttöönotettujen robottien määrä (168 000)

on yli nelinkertainen toisena listalla tulevaan Japaniin, jossa robotteja käyttöön otettiin noin 38 000 kappaletta. IFR:n mukaan vuoden 2020 loppuun mennessä käytössä olevia teollisuusrobotteja oli yhteensä yli 3 miljoonaa kappaletta koko maailmassa. (IFR 2021a.)



KUVIO 1. Teollisuusrobottien asennukset vuonna 2020 (IFR 2021a)

Suomessa vuosittain käyttöön otetaan alle 1000 teollisuusrobottia, mikä on huomattavan pieni määrä verrattuna johtaviin robottimaihin. Suomessa on kuitenkin paljon korkeatasoista koulutusta robotiikan alalla, joka parantaa Suomen mahdollisuuksia robotiikan kehityksessä. (Suomen Robotiikkayhdistys 2019.)

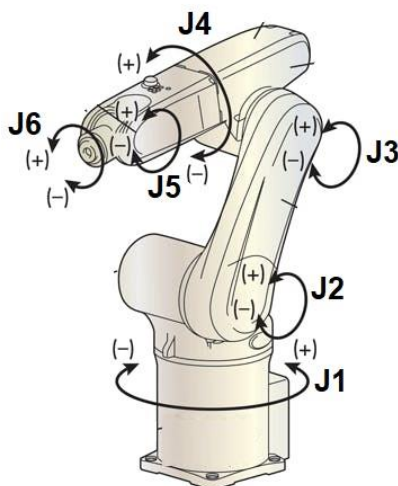
3.3 Robottityypit

Teollisuusrobotit voidaan luokitella kuuteen eri pääkategoriaan niiden mekaanisen rakenteen mukaan. Kategorioita ovat kiertyväniveliset-, suorakulmaiset-, sylinteri-, napakoordinaatisto-, SCARA-, ja rinnakkaisrakenteiset robotit. (Robots 2021b.)

Kiertyväniveliset robotit

Kiertyväniveliset robotit ovat yleisin teollisuusympäristöissä käytetty robottityyppi. Nivelrobotit määritellään roboteiksi, joissa on pyörivät nivelet (kuva 3). Niveleitä kutsutaan yleisesti robotin akseleiksi. Nivelrobotilla voi olla kolmesta yli kymmeneen akselia, mutta tavallisesti teollisuudessa käytetyissä roboteissa on kuitenkin neljästä kuuteen akselia, joista kuusiakseliset ovat tyypillisimpiä. (Robots Done Right 2021e.)

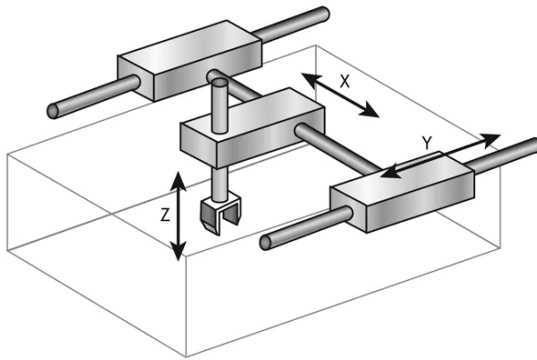
Kiertyväniveliset robotit muistuttavat rakenteeltaan ihmisen käsivartta ja ne pyrkivät jäljittelemään tarkasti ihmisen liikettä. Tämä tekee niistä ihanteellisia ratkaisuja tuotantolinjoille, joissa tarvitaan laajoja työskentelyalueita. Suurimpia robotivalmistajia maailmassa ovat Fanuc, ABB, Yaskawa Motoman ja KUKA. Valmistettavista robottityypeistä yleisin on kuusiakselinen nivelrobotti. (Robots Done Right 2021e.)



KUVA 3. Kiertyvänivelinen robotti, jossa on kuusi akselia J1-J6 (Gonzales 2016)

Suorakulmaiset robotit

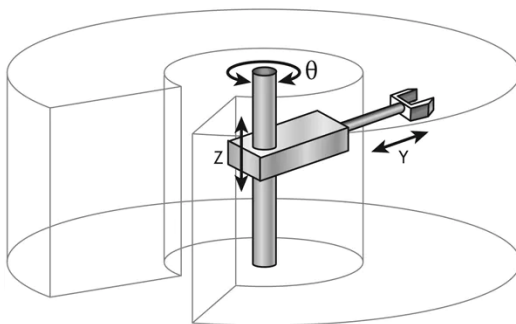
Suorakulmaisilla roboteilla (kuva 4) on kolme lineaarista niveltä, jotka noudattavat cartensian-koordinaattijärjestelmää (X, Y ja Z). Suorakulmaisia robotteja kutsutaan myös portaaliroboteiksi ja niillä voi olla lisäksi pyörivän liikkeen mahdollistava ranne. Suorakulmaiset robotit liikkuvat koordinoitusti yhteisen liikeohjaimen avulla ja niitä käytetään esimerkiksi materiaalin käsittelyssä ja numeerisesti ohjatuissa laitteissa. Portaalirobotteja voidaan rakentaa käytännössä mistä tahansa lineaarisista toimilaitteista, joiden käyttömekanismeja voi olla esimerkiksi hihnakäyttö, kuularuuvi, pneumaattinen toimilaite tai lineaarimoottori. (Collins 2018.)



KUVA 4. Suorakulmainen cartensian-robotti (Gonzales 2016)

Sylinterirobotit

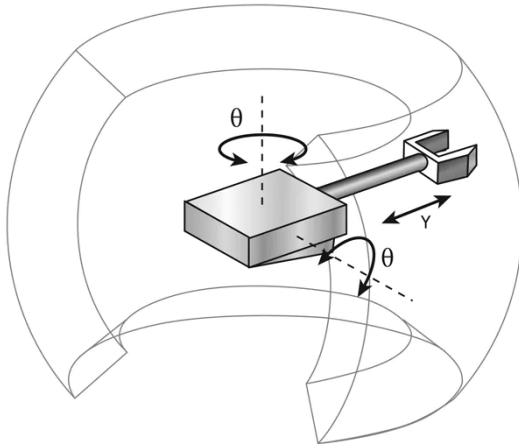
Sylinteriroboteissa (kuva 5) on yksi akselin suuntainen pyörivä nivel kiertoliikettä varten ja toinen nivel lineaarista liikettä varten. Nivelakselia pitkin pyörivä nivel tekee kiertoa liikettä ja toinen akseli liikkuu lineaarisesti. Sylinterirobotin liikkeet tapahtuvat sylinterin muotoisella liikealueella. Sylinterirobotteja käytetään yleisesti kokoonpanotehtävissä, työstökoneiden kappaleenkäsittelyssä sekä pistehitsauksessa. (Gonzales 2016.)



KUVA 5. Sylinterirobotti (Gonzales 2016)

Napakoordinaatistorobotit

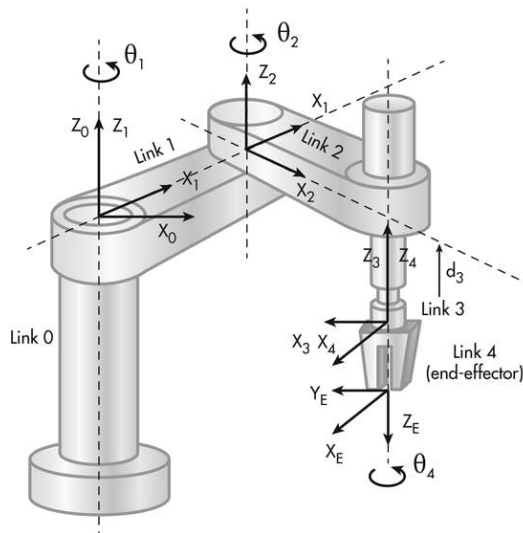
Napakoordinaatistorobotti (kuva 6) toimii kahdella pyörivällä nivelellä ja yhdellä lineaarisella nivelellä pallonmuotoisen liikealueen muodostamiseksi. Naparobotin varsi on yhdistetty alustaan kiertoa liikettä varten, ja sen akseleiden yhdistelmä muodostaa napakoordinaattijärjestelmän. Napakoordinaatistorobotteja käytetään esimerkiksi pistehitsaukseen, painevaluun, sahauskoneisiin ja kaarihitsaukseen. (Gonzales 2016.)



KUVA 6. Napakoordinaatistorobotti (Gonzales 2016)

SCARA-robotit

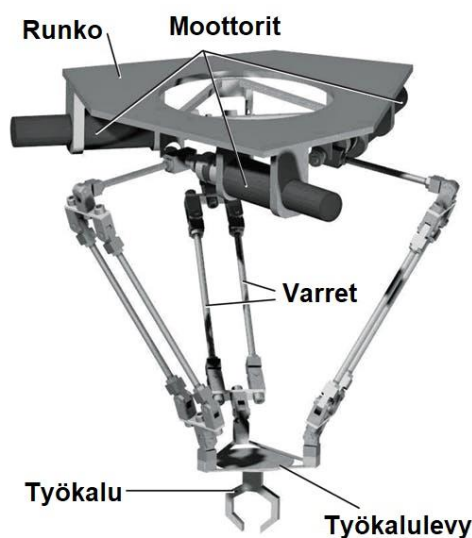
SCARA-robottia käytetään pääasiassa poraus- tai kierrekokoonpanoja vaativissa töissä sen liikkeen luonteen vuoksi (Gonzales 2016). SCARA-robotilla on kaksi yhdensuuntaista niveltä, jotka mahdollistavat liikkumisen yhdessä tasossa todella tarkasti ja nopeasti (kuva 7). SCARAN mekaniikka on yleensä melko vahva ja kestää kovia rasituksia ja odottamattomia törmäyksiä. Tästä syystä sitä käytetään usein voimakkaissa paineissa, kuten pystysuorissa rei'ityksissä ja porauksissa. (FlexiBowl 2021.)



KUVA 7. SCARA-robotti (Gonzales 2016)

Rinnakkaisrakenteiset robotit

Rinnakkaisrakenteisissa- eli deltaroboteissa on tyypillisesti kolmesta neljään kevyttä vartta, jotka suuntaavat alaspäin päärobotin rungosta (kuva 8). Delta-robotin varret on liitetty keskeltä, jolloin ne taipuvat sisäänpäin ja yhdistyvät pieneen työkalulevyyn. Jokainen varsi on kytketty moottoriin, joka sijaitsee päärobotin rungossa. Moottorit koordinoivat käsivarsien liikkeitä, joiden avulla työkalulevy saadaan liikkumaan X-, Y- ja Z-akselien suuntaisesti todella nopeasti. Rinnakkaisrakenteisia robotteja käytetään usein poimintaoperaatioissa, kuten elintarvikkeiden lajittelutöissä ja annosteluissa sen nopeiden liikkeiden takia. (Robots Done Right 2021.)



KUVA 8. Rinnakkaisrakenteinen robotti (Gonzales 2016)

Yhteistyörobotit

Yhteistyössä toimivat teollisuusrobotit eli Cobotit (kuva 9) on suunniteltu suorittamaan tehtäviä yhteistyössä työntekijöiden kanssa (IFR 2021b). Ne mahdollistavat ihmisten ja robottien perinteisiin teollisuusroboteihin verrattuna turvallisemman työskentelyn samassa ympäristössä ilman vammojen/vaurioiden riskiä. Yhteistyöroboteille voidaan opettaa pisteitä yksinkertaisesti liikuttamalla robottia manuaalisesti haluttuun suuntaan. Yhteistyörobotti pystyy siten toistamaan tarkan reitin, joka sille opetettiin. (Robots 2021a.)



KUVA 9. ABB-yhteistyörobotteja (ABB 2021a)

Cobotit ovat ulkomuodoltaan tavallista robottia pyöreämpiä ja niiden kaikki johdot on sijoitettu robotin rungon sisälle, eikä tällöin puristumisvaara ole yhtä suuri. Cobotit voidaan ohjelmoida reagoimaan välittömästi pysähtymällä tai vaihtamalla asentoa, kun ne joutuvat ihmiskontaktiin. Yhteistyörobotit säästävät tilaa ja parantavat turvallisuutta, mikä voi tarjota huomattavia etuja yrityksille. (Robots 2021a.)

3.4 Robottihitsaus

Robottihitsauksessa hitsausprosessi on automatisoitu robottien avulla (kuva 10). Se on pitkälle kehittynyt versio automaattisesta hitsauksesta, jossa robotit suorittavat hitsauksen ohjelmaan perustuen. Siinä robotit suorittavat hitsausprosessin, mutta ihmiset valvovat ja ohjaavat sitä. (Kemppi 2021.)



KUVA 10. Robottihitsausta Emecan tuotannossa

Robottiteknologia tuo hitsausprosessiin lukuisia etuja. Hitsausprosessissa tulokset ovat nopeita ja tarkkoja. Robottihitsauksen avulla pystytään vähentämään jätettä ja takamaan parempi turvallisuus. Teollisuusrobotit yltyvät haastaviinkin

paikkoihin, ja ne voivat suorittaa manuaalista hitsausta tarkempia ja monimutkaisempia hitsauslinjoja. Robottihitsauksen ansiosta valmistukseen kuluu vähemmän aikaa ja se on joustavampaa. (Kempfi 2021.)

Hitsauksessa käytetään usein termiä kaariaikasuhde, joka tarkoittaa hitsauskaaren palamiseen ja hitsaustyön suorittamiseen kuluvan ajan suhdetta (SFS 3054 2020, 15). Robottihitsauksessa päästään yli 80 % kaariaikasuhteeseen ja se voidaan säilyttää koko tuotantoprosessin ajan. Manuaalisesti hitsattaessa kaariaikasuhde on yleensä vaan noin 35 prosenttia, koska robotit pystyvät työskentelemään huomattavasti tehokkaammin, tasaisemmin ja pidempään kuin ihmiset. (Wilson 2014, 78.)

Korkean tuottavuuden ansiosta robottihitsauksesta on tullut tärkeä rooli metalliteollisuudessa ja erityisesti piste- ja laserhitsausta käyttävässä autoteollisuudessa. Vaikka robottihitsausta käytetään yleensä suurissa massatuotannoissa, joissa tuotantomäärät ja tehokkuus ovat avainasemassa, voidaan robotiikkaa hyödyntää myös pienempiin sarjoihin ja jopa yksittäisten kappaleiden valmistukseen kustannustehokkaasti. (Kempfi 2021.)

3.5 Sykli aika

Robotiikassa yhteen työkiertoon kuluvaa aikaa kutsutaan yleensä sykli ajaksi. Se on olennainen osa teollisuusrobottien ohjelmointia ja suunnittelua. Sykli aikaa pidetään keskeisenä indikaattorina tuotannon tehokkuudesta ja siksi yritykset pyrkivätkin jatkuvasti optimoimaan työkierron keston kuluvaa aikaa. (Robots Done Right 2022.)

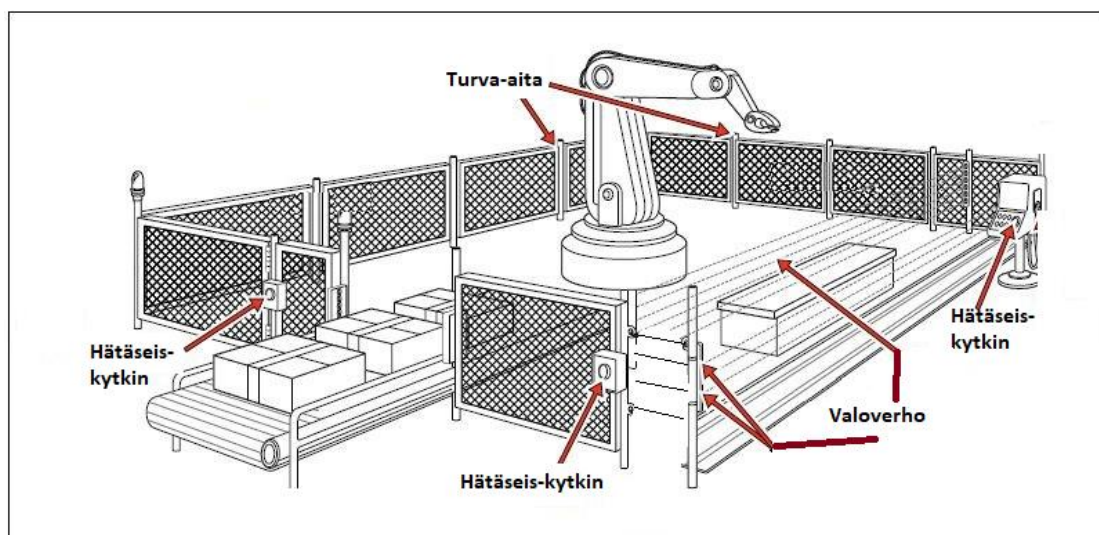
Koska teollisuusrobotit ovat nopeita, tarkkoja ja luotettavia, ovat ne suuressa roolissa sykli ajan parantamisessa. Robottien lisäämisellä tuotantoon on huomattu olevan positiivinen vaikutus sykli aikoihin ja sitä kautta myytävien tuotteiden määrään. (Robots Done Right 2022.)

3.6 Robottiturvallisuus

Teollisuusrobottien ja teollisuusrobottijärjestelmien turvallisuusvaatimukset määritetään standardeissa ISO 10218-1 ja ISO 10218-2. Standardit määrittävät teollisuusrobottien suunnittelussa ja rakenteessa huomioitavat riskit ja vaarat sekä robottijärjestelmien turvallisen käytön. (SFS-EN ISO 10218-1 2011, 8.)

Teollisuusrobottien ominaisuudet voivat poiketa merkittävästi muiden laitteiden ja koneiden ominaisuuksista ja siksi vaarojen tunnistaminen on erityisen tärkeää. Robotit voivat liikkua suurella energialla, laajalla toiminta-alueella ja niiden liiketapa voi olla vaikeasti ennustettava. Robotin käyttäjän on usein myös tarpeellista työskennellä robotin tai robottijärjestelmän välittömässä läheisyydessä, mikä kohoaa turvallisuusriskiä huomattavasti. (SFS-EN ISO 10218-2 2011, 10.)

Turvalaitteet ovat olennainen osa kaikkia robottijärjestelmiä tai -soluja, jotta työntekijä voi työskennellä turvallisesti robotin ollessa liikkeellä. Nykyään markkinoilla on useita turvallisuusvaihtoehtoja. Robottijärjestelmä voidaan turvallistaa esimerkiksi hätäseis-kytkimillä, turva-aidoilla tai muilla turvalaitteilla (kuva 11). Yleisin turvavaihtoehto on turva-aita, jolla suojataan robottijärjestelmän ympärillä oleva alue. Turva-aidan lisäksi kulkuväylien ja muiden avointen alueiden suojaaminen voidaan toteuttaa turvaloverhoilla tai turvalaserskannereilla. Näiden avulla varmistetaan robottien turvallinen käyttö, koska valoverhon tai laserskannerin tunnistessa se laukaisee välittömästi hätäseis-piirin, jolloin robotti pysähtyy. Alueskanneri on samanlainen kuin valoverho, mutta sen sijaan, että sillä olisi yksi tietty alue, jolla valo siirtyy lähettimestä vastaanottimeen, alueskanneri pystyy skannaamaan laajan alueen. (Robots 2022.)



KUVA 11. Robottijärjestelmän turvallistaminen (Kishan 2016)

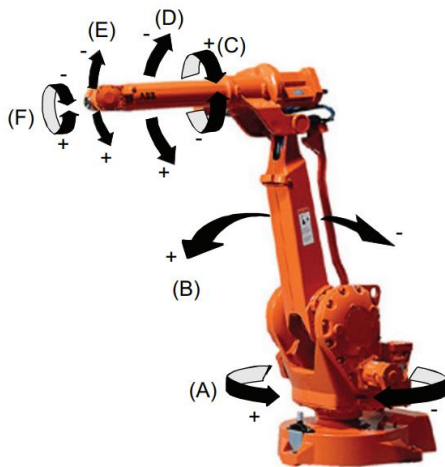
Hätäseis-kytkimiä käytetään hätätilanteissa. Robottisolun tai -järjestelmän ympärillä on yleensä yksi tai useampi hätäseis-painike, jotka pysäyttävät robotin liikkeen välittömästi. Toisin kuin valoverhot ja laserskannerit, hätäseis-kytkimen aktivoinnin jälkeen robottia ei voi liikuttaa ollenkaan, edes manuaalisesti käsiohjaimella ennen kuin kytkin on vapautettu. (Robots 2022.) Edellä mainitut robottijärjestelmien turvalaitteet ovat pakollisia ja ne on suunniteltu suojaamaan henkilöitä robottijärjestelmän vaaroilta. (SFS-EN ISO 10218-2 2011, 20.)

4 ABB TEOLLISUUSROBOTIT

4.1 ABB robottivalmistajana

ABB on robotiikan, koneautomaation ja digitaalisten palveluiden edelläkävijä, joka tarjoaa innovatiivisia ratkaisuja autoteollisuudesta elektroniikkaan ja logistiikkaan. ABB on toimittanut Suomeen jo yli 3000 robottia ja maailmanlaajuisesti yli 400 000 robottia (ABB 2021). Yhtenä maailman johtavista robotiikan ja koneautomaation toimittajista ABB omaa laajan valikoiman nivelvarsirobotteja, jotka kattavat satoja teollisia sovelluksia (Robots Done Right 2021).

ABB valmistaa nivelvarsi-, yhteistyö-, maalaus-, SCARA- ja rinnakkaisrakenteisia robotteja, joista käytetyimpiä robottimalleja ovat suuret nivelrobotit (kuva 12). Niitä käytetään tyypillisesti autotehtailla pistehitsaukseen erinomaisen tarkkuuden ja luotettavuuden vuoksi. (Robots Done Right 2021.)



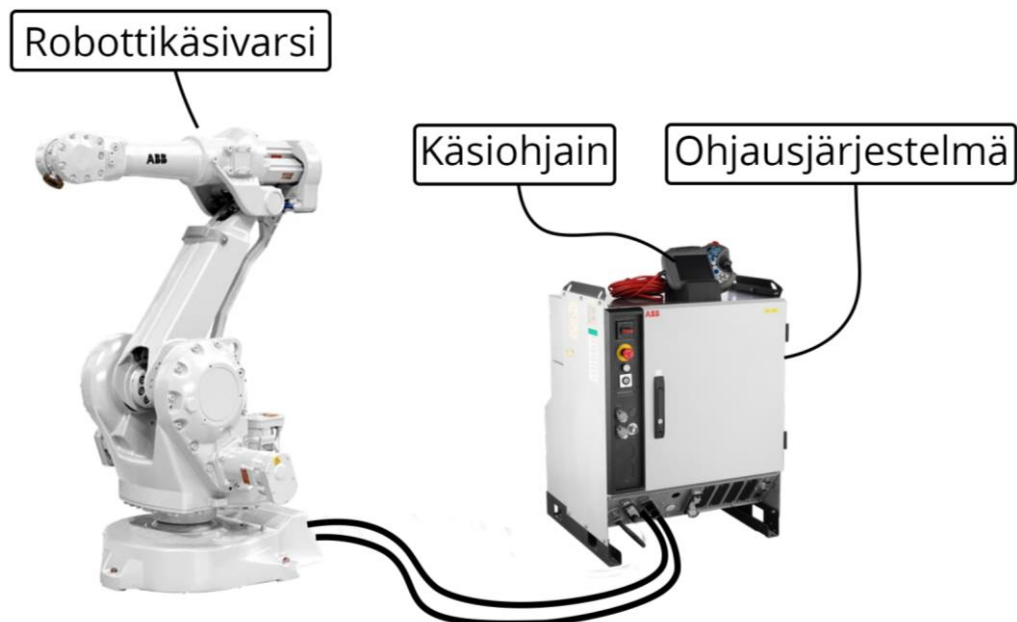
KUVA 12. ABB-teollisuusrobotti (ABB 2021c, 10)

ABB:n laaja robottivalikoima sisältää useita erikokoisia robotteja, joista suurimpien nostokyky on jopa 1000 kg. Robotteja on saatavana yli 4 metrin yltämällä, mikä takaa laajat ja monipuoliset käyttömahdollisuudet. (ABB 2022a.)

4.2 ABB robottijärjestelmä

ABB-robottijärjestelmä (kuva 13) koostuu robottikäsiarresta, FlexPendant-käsiohjaimesta ja ohjausjärjestelmästä. ABB valmistaa erinomaisella liikkeenhallinnalla varustettuja ohjausjärjestelmiä, joihin lukeutuu IRC5- ja OmniCore-ohjaimet. Teollisuudessa yleisimmin käytetty ohjausjärjestelmä on ABB:n IRC5. (ABB 2022c.)

ABB robottijärjestelmä omaa korkean suorituskyvyn robottisovelluksiin, jotka vaativat erittäin suurta tarkkuutta ja luotettavuutta. ABB-robottien vahvuutena on niiden vankka rakenne, joka takaa pitkät huoltovälit ja erinomaisen luotettavuuden. (ABB 2022b.)



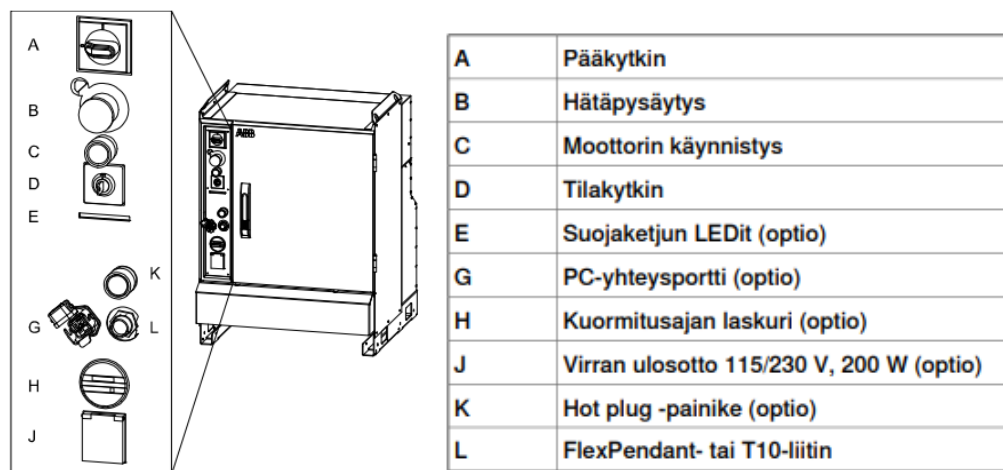
KUVA 13. ABB IRC5- robottijärjestelmä (ABB 2022b, 2022c)

4.2.1 IRC5-Ohjausjärjestelmä

IRC5-ohjain (kuvio 2) on ABB:n teollisuusrobottien viidennen sukupolven ohjausjärjestelmä, joka on kehittynyt vuosikymmenten robotiikkakokemuksen tuloksena edellisestä S4C-järjestelmästä. IRC5 takaa korkean IP-suojauksen ja laajennettavuuden. Sen avulla pystytään ohjaamaan jopa neljää erillistä robottia lisäksi leineen MultiMove-asetuksella. (ABB 2021.)

IRC5-ohjain pitää sisällään kaikki robotin liikuttamiseen ja ohjaamiseen vaadittavat toiminnot. IRC5-ohjaimen standardimalli on yksikoteloinen. Ohjain voidaan myös jakaa kahteen ohjauskaappiin tai sisällyttää ulkoiseen koteloon (ABB 2021b, 16.)

Robotin ohjauskaapista voidaan kytkeä virrat päälle, valita haluttu operointitila ja käynnistää robottien moottorit. Tarvittaessa robotin liike voidaan pysäyttää hätäpysäytyspainikkeella. IRC5-ohjaimen voidaan myös kytkeä RJ-45 portin kautta tietokone, RobotStudio-sovelluksen online- tai offline-ohjelmointia varten. (ABB 2021b, 32.)



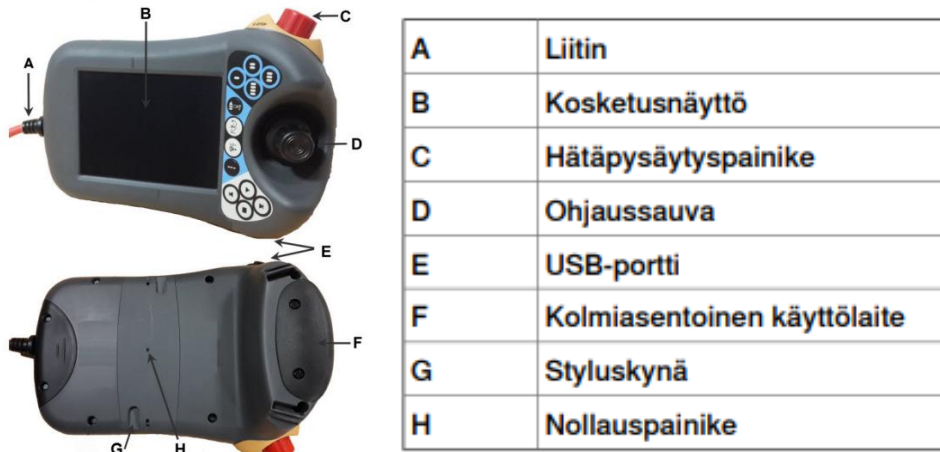
KUVIO 2. IRC5-Ohjain (ABB 2021b, 32)

IRC5-ohjausjärjestelmä koostuu kahdesta eri moduulista, Ohjausmoduuli ja Käyttömoduuli. Ohjausmoduuli pitää sisällään kaiken ohjauselektronikan päätietokoneesta I/O-korttiin (tulo- ja lähtötieto) ja Käyttömoduuli sisältää robotin moottoreiden tarvitseman sähköelektronikan, kuten servovahvistimet. Jos samalla ohjaimella halutaan ohjata useampaa robottia, on lisäoptiona asennettava toinen Käyttömoduuli. (ABB 2021b, 16.)

4.2.2 FlexPendant-käsiohjain

FlexPendant (kuvio 3) on käyttölaite, jonka avulla voidaan suorittaa monia robotijärjestelmän käyttöön liittyviä toimintoja kuten luoda ja muokata ohjelmia, mää-

rittää muuttujia ja liikuttaa robotin akseleita. Käsiohjain sisältää laitteiston ja ohjelmiston, ja on itsessään jo täysimittainen tietokone. FlexPendant on osa IRC5-järjestelmää, ja se on kytketty robotin ohjausjärjestelmään ohjauskaapelin avulla. (ABB 2021b, 17.)

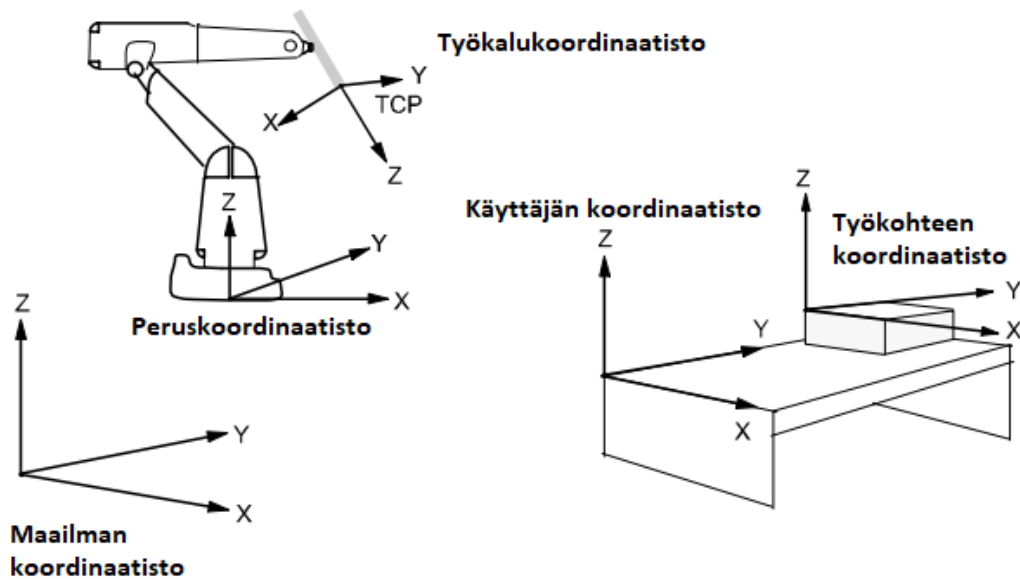


KUVIO 3. FlexPendant-käsiohjain (ABB 2021b, 18)

4.3 Koordinaatistot

Koordinaattijärjestelmä määrittää robotin akseleiden suhteen avaruuden tai tason origoksi kutsutusta kiinteästä pisteestä. Koordinaattijärjestelmien akseleiden suuntaisten mittausten avulla voidaan paikantaa robotin sijainnit ja asennot. Robotin käytössä on monia koordinaattijärjestelmiä, jotka soveltuvat kukin tietynlaiseen ohjelmointiin tai siirtymiseen. Robotin koordinaatistoja ovat peruskoordinaatisto, työkohteen koordinaatisto, työkalukoordinaatisto, maailman koordinaatisto ja käyttäjän koordinaatisto. (ABB 2021b, 105.)

Peruskoordinaatiston (kuva 14) origo eli nollapiste on robotin alustassa. Ohjelmointi peruskoordinaatistossa on kätevää, jos robotti on kiinteästi paikallaan. Peruskoordinaatistoa voidaan käyttää suurten välimatkojen ja nopeiden siirtymien määrittämisessä. (ABB 2021b, 107.)



KUVA 14. Koordinaattijärjestelmät (ABB 2010, 1225)

Työkohteen koordinaatisto määrittää sijainnin suhteessa maailman koordinaatistoon tai käyttäjän koordinaatistoon. Se voidaan määrittää haluamalle tasolle tai kappaleelle. Robotilla voi olla monia työkoordinaatistoja esimerkiksi useammalle työkappaleelle. (ABB 2021b, 108.)

Työkalun koordinaatiston origo on työkalulaipan keskipisteessä ja se määrittää työkalun asennon ja suunnan. Työkalukoordinaatistosta käytetään lyhennettä TCPF (Tool Center Point Frame). Työkalukoordinaatiston keskipistettä kutsutaan työkalukeskipisteeksi TCP (Tool Center Point). Robottia liikutettaessa TCPF:n avulla työkalun suunnan voi pitää samana koko liikkeen ajan. (ABB 2021b, 109.)

Maailman koordinaatistossa nollapiste sijaitsee kiinteässä kohdassa. Se on hyödyllinen, jos on tarve käsitellä useampia robotteja samassa koordinaatistossa. Alkuoletuksena maailman koordinaatisto vastaa peruskoordinaattijärjestelmää. (ABB 2021b, 110.)

Käyttäjän koordinaatisto voidaan määrittää esimerkiksi kahteen eri työtasoon. Mikäli työtasot liikkuvat, voidaan koordinaatiston uudelleenosoituksella päivittää liikepisteiden sijainnit. Käyttäjän koordinaatisto on silloin hyödyllinen, jos esimerkiksi useita työkohteen koordinaatistoja halutaan siirtää käyttäjänkoordinaatistossa. (ABB 2021b, 177.)

4.4 Ohjelmointi

ABB-robottien ohjelmointi voidaan toteuttaa FlexPendant-käsiohjaimella tai RobotStudio-sovelluksella. FlexPendant soveltuu parhaiten yksinkertaisempien ohjelmien luontiin sekä sijaintitietojen ja liikeratojen muuttamiseen. RobotStudiolla voidaan luoda ja simuloida vaativampia kokonaisuuksia nopeammin ja yksinkertaisemmin. (ABB 2021b, 129)

Tuottavuuden parantamiseksi ABB on kehittänyt ohjelmistotuoteperheen, joka tarjoaa laajan valikoiman helppokäyttöisiä ohjelmistosovelluksia, jotka auttavat optimoimaan tuotantoa, lisäämään tuottavuutta ja vähentämään riskejä. Ohjelmisto tarjoaa helpon ja joustavan ohjelmoinnin sekä suoraviivaisen konfiguroinnin. ABB:n tarjoamia sovelluspaketteja on esimerkiksi 3D-tulostuksen PowerPac, Hitsaukseen Arc Welding-sovellus, Lavaukseen Palletizing-sovellus ja laserleikkaukseen Cutting-sovellus sekä lukuisia muita valmiita sovelluksia. Näiden avulla on huomattavasti helpompi luoda ja päivittää uusia ohjelmia riippumatta onko käyttäjänä robotin operaattori vai ohjelmoija. (ABB 2022d.)

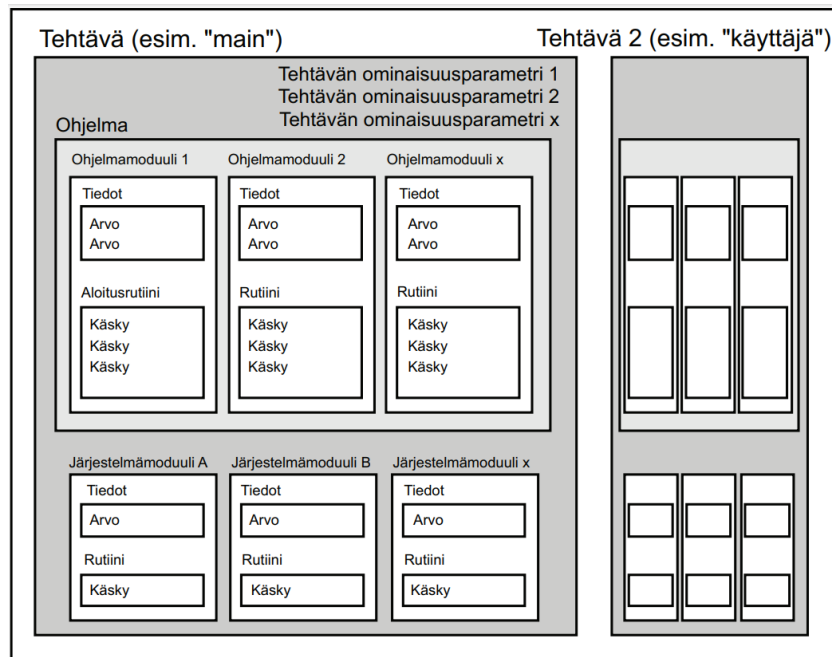
RAPID-ohjelmointikieli

ABB-robotteja ohjelmoidaan korkean tason ohjelmointikielellä RAPID. Ohjelma koostuu lukuisista käskyistä (liitteet 1 ja 2), joilla ohjataan robottia ja sen toimilaitteita. Ohjelmassa voidaan esimerkiksi suorittaa liikekäskyjä, kutsua aliohjelmiä, asettaa lähtöjä ja määrittää muuttujia sekä tallentaa dataa. (ABB 2013, 13.)

RAPID-ohjelmointikielen ominaisuuksiin lukeutuu esimerkiksi modulaarinen ohjelmointi tehtävillä ja moduuleilla, aliohjelmat, funktiot ja aritmetiikka. Lisäksi ohjelmointikieli sisältää virheen korjauksen, ohjelman kumoamiskäsittelyn, keskeytyksen käsittelyn ja käänteisen suorittamisen tuen. (ABB 2021d, 11.)

Ohjelmarakenne

RAPID-sovellus (kuva 15) koostuu tehtävästä "TASK", joka pitää sisällään ohjelmia ja järjestelmämoduuleja. RAPID-sovellus sisältää normaalisti yhden tehtävän. Mikäli IRC5-ohjaimella halutaan kontrolloida useampaa robottia, tarvitaan Multitasking-lisäoptio. Sen avulla tehtäviä voi olla useampia. (ABB 2013a, 158.)



KUVA 15. RAPID-sovelluksen rakenne (ABB 2013a, 158)

Ohjelmat sisältävät ohjelmamoduuleja, joissa on tietoja ja rutiineja käytettäväksi tiettyihin tarkoituksiin. Ohjelmamoduulit selkeyttävät ja helpottavat ohjelman rakenteen ymmärtämistä ja käsittelyä. Järjestelmämoduuleissa määritellään yleisiä, järjestelmäkohtaisia tietoja ja rutiineja, kuten erilaisten työkalujen tietoja. Järjestelmämoduulit ovat yleensä robotin valmistajan luomia. (ABB 2013a, 159.)

Ohjelman rutiineja voidaan kutsua aliohjelmiksi, sillä ne sisältävät käskyjä, jotka robotti toteuttaa aina kyseistä rutiinia suoritettaessa. Yksi ohjelmamoduuleista sisältää rutiinin "main", joka on ohjelman aloitusrutiini. Se on pakollinen päärutiini, jota ilman robotin ohjelmaa ei voida suorittaa. Tavallisesti pääruutiinista kutsutaan muita rutiineja tiettyjen ehtojen täytyttyä. (ABB 2021e, 18.)

Yksinkertaisissa sovelluksissa yksi moduuli saattaa riittää koko ohjelman toteuttamiseen, mutta suurissa robottisovelluksissa voi olla päämoduuli, joka viittaa rutiineihin ja tietoihin, jotka sisältyvät yhteen tai useampaan eri moduuliin. (ABB 2021e, 17.)

Muuttujat

RAPID-ohjelmointikielessä yleisimmin käytetyt muuttujat ovat totuusarvomuttujat (bool), numeeriset arvot (num ja dnum) sekä merkkijonot (string). Totuusarvomuttuja ilmaisee loogista totuusarvoa TRUE tai FALSE. Sitä voidaan käyttää

esimerkiksi ilmaisemaan, kun ohjelman tietty rutiini on suoritettu. Numeeriset arvot voivat olla kokonais- tai desimaalilukuja ja string-merkkijonoja käytetään määrittäessä muuttuja vastaamaan ISO 8859-1 standardin mukaisia merkkejä, kuten kirjaimia. (ABB 2021d, 31.)

Ohjelmassa kaikki muuttujat voidaan nimetä halutulla tavalla, lukuun ottamatta RAPID-kielen varattuja sanoja kuten IF, THEN ja ELSE. Varatuilla sanoilla on jo ennalta määritetty funktio ohjelmakielessä eikä niitä voida korvata. (ABB 2021d, 13.)

Ohjelmadata

RAPID-ohjelmointikielessä dataa voidaan tallentaa joko muuttuvana (VAR), vakiona (CONST) tai pysyvänä (PERS) kuvan 16 mukaisesti. Data voidaan myös määrittää tallennettavaksi globaalisti, paikallisesti tai tehtävän sisäisesti. Globaali data näkyy moduulista ja rutiinista riippumatta kaikkialla, mutta paikallinen data pysyy tietyn moduulin sisällä. (ABB 2021d, 29.)

```
PERS robtarget P_Paikotus_piste:=[[757.03,77.60,-442.00]
CONST num Laatikon_koko:=350;
VAR num Laatikon_korkeus:=100;
PERS num Riveja_lavalla:=3;
PERS num Sarakkeita_lavalla:=2;
```

KUVA 16. RAPID datan tallennusmuodot

Muuttuvana (VAR) tallennettu data säilyy, jos ohjelma pysähtyy ja käynnistetään uudelleen, mutta häviää jos ohjelma aloitetaan alusta (ABB 2013b, 14). Pysyvänä tallennettu data (PERS) vastaa muuttuvaa dataa, mutta se säilyttää arvonsa, vaikka ohjelma pysäytetään ja aloitetaan alusta. Pysyvän muuttujan arvo voidaan muuttaa ainoastaan uudelleenmäärityksellä. (ABB 2013b, 16).

Vakio data (CONST) määritetään ohjelman alussa, eikä sitä pystytä sen jälkeen enää muuttamaan. Määrittämällä muuttuja vakioksi voidaan varmistaa, ettei sen arvo muutu missään vaiheessa kesken ohjelman. (ABB 2013b, 17.) Sitä voidaan käyttää esimerkiksi piin likiarvon tai robotin kotipisteen määrittämisessä.

Operaattorit

RAPID-ohjelmointikieli sisältää operaattoreita, joita voidaan käyttää esimerkiksi muuttujien vertailussa, laskemisessa sekä ehtolauseissa kuvan 17 mukaisesti. Käytettävissä on loogisia-, numeerisia- ja merkkijono- eli string-operaattoreita. Numeeriset operaattorit palauttavat numeerisen arvon, loogiset operaattorit palauttavat loogisen tosi tai epätosi arvon ja string-operaattorit palauttavat arvon merkkijonona.

(ABB 2013b, 18.)

```
PROC Aihioopinon_sijainnin_laskenta()
  Pinoja_Lavalla := Riveja_lavalla * Sarakkeita_lavalla;
  Sarake_nro := Pinon_numero MOD Sarakkeita_lavalla;
  Rivi_nro := Pinon_numero DIV (Riveja_lavalla - 1);
  Pinon_haku_offs_X := R_Aihio_X_Offset * Sarake_nro;
  Pinon_haku_offs_Y := R_Aihio_Y_Offset * Rivi_nro;
ENDPROC
```

KUVA 17. Operaattorien käyttö RAPID-ohjelmassa

Ohjelmakäskyt

RAPID-ohjelmointikielessä on monia käskyjä ja funktioita. Käsky voi esimerkiksi asettaa lähtösignaalin päälle, odottaa muuttujan arvoa tai esittää käyttäjälle ilmoituksen. Ohjelmakäskyjen avulla voidaan suorittaa haluttuja toimintoja ja rutiineja tietyin reunaehdoin. Kuvassa 18 on esitetty yleisten ohjelmakäskyjen käyttö RAPID-ohjelmointikielessä.

```
WaitRob\InPos;
WaitDI diTarttutaraja, 1;
IF diTarttutaraja = 1 THEN
  Reset doMagneettitarttuja;
  WaitTime 0.5;
```

KUVA 18. RAPID-ohjelmointikäskyjen käyttö

Ehtolauseita voidaan muodostaa RAPID ohjelmointikielessä esimerkiksi IF-THEN käskyillä, jolloin ohjelma toteuttaa halutun toiminnon, mikäli ehdot käskylle toteutuvat. Odotuskäskyjä, kuten WAIT, käytetään esimerkiksi anturitiedon odottamiseen. Robotin lähtöjä ohjataan SET/RESET-komennoilla, joiden avulla saadaan asetettua haluttu lähtö ON/OFF-tilaan.

(ABB 2021e, 46,63.)

Liikekäskyt

RAPID-ohjelmointikielen etuna on, että se on erityisesti suunniteltu ohjaamaan robotin liikkeitä. Robotin liikkeet ohjelmoidaan sijainnista sijaintiin -liikkeiksi, joiden välisen reitin robotti laskee automaattisesti. (ABB 2021e, 55.)

RAPID ohjelmassa on monia liikkeen ohjaamiseen tarkoitettuja käskyjä, joista yleisimpinä MoveL, MoveJ ja MoveC, Näiden käskyjen perässä oleva kirjain kertoo liikkeen tyyppin. (ABB 2021e, 55.) Liikekäskyt esitetty taulukossa 1.

TAULUKKO 1. RAPID-ohjelmointikielen liiketyypit (ABB 2021e, 55)

Instruction	Type of movement
MoveC	TCP moves along a circular path.
MoveJ	Joint movement.
MoveL	TCP moves along a linear path.

Liikekäskyt sisältävät tavallisesti sijaintitiedon, nopeuden, liiketarkkuuden, työkalun tiedot ja työkohteen koordinaatiston tiedot (taulukko 2). Liikekäskyihin voidaan lisätä myös erilaisia komentoja, kuten Offset tai RelTool, mikäli liikepisteen sijaintia halutaan siirtää tietyssä koordinaatistossa x, y tai z suuntaan ilman uuden pisteen opettamista. (ABB 2021e, 55,61.)

TAULUKKO 2. Liikepisteen tiedot RAPID ohjelmassa (ABB 2021e, 55.)

MoveL P_lavaus, v100, fine, gripdon\WObj:=WO_lavaus;	
MoveL	Liikuttaa robottia lineaarisesti suorassa linjassa määritettyyn pisteeseen
P_lavaus	Piste johon robotti siirtyy
v100	Robotin nopeus mm/s
fine	Robotti liikkuu haluttuun pisteeseen oikaisematta kulmia
gripdon	Robotin määritetty työkalu
WObj:=WO_Lavaus	Työkohteen koordinaatisto

5 TEOLLISUUSROBOTTIEN OHJELMOINTITAVAT

5.1 Opettamalla ohjelmointi

Teollisuusrobottien ohjelmointitavoista yleisin on opettamalla ohjelmointi (kuva 24). Yli 90 prosenttia teollisuusroboteista on ohjelmoitu tällä tavalla. Opettamalla ohjelmoinnin pääperiaate on robotin liikuttaminen tiettyyn sijaintiin valitussa koordinaatistossa käsiohjaimen avulla. Opetetut sijainnit tallennetaan robotin ohjelmaan liikekäskyinä, joihin robotti voi siirtyä ohjelmaa suoritettaessa. Tämä ohjelmointimenetelmä on ollut suosituin jo useiden vuosien ajan sen helppouden takia. (Wilkins 2021.)

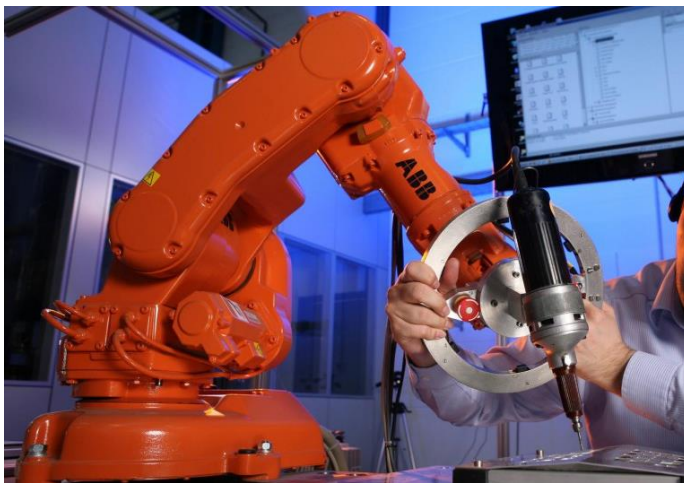
Opettamalla ohjelmoimisen hyöty perustuu sen yksinkertaisuuteen. Suurin osa markkinoilla olevista nykyaikaisista teollisuusroboteista on varustettu käsiohjaimella, jonka avulla jopa tavallisen käyttäjät voivat ohjata ja ohjelmoida robottia suorittamaan haluttuja toimintoja. Tämä mahdollistaa nopeammat asennus- ja ohjelmointiajat. Käsiohjaimella ohjelmoitaessa ei ole tarvetta ostaa lisäohjelmia tai välineitä, jotka vaativat erikoisosaamista. (Robots Done Right 2021c.) Koska opettamalla ohjelmoitaessa ollaan varsin lähellä robottia, on virheiden ja törmäysten määrää helpompi hallita ja ennakoida.

Opettamalla ohjelmoimisen yksi suurimmista haittapuolista on ajankäyttö, varsinkin kun ohjelmoidaan monimutkaisia sovelluksia. Mitä enemmän komentoja sovelluksella on, sen kauemmin ohjelmointi kestää, koska jokainen ohjelmoinnin vaihe on syötettävä manuaalisesti.

Toinen merkittävä haittapuoli on se, että robotit on pysäytettävä ja tuotanto keskeytettävä ohjelmoinnin ajaksi. (Robots Done Right 2021c.) Haittapuolena on myös, että ohjelmointi voidaan aloittaa vasta, kun robottisolu on valmiina ja robotit ovat oikeilla paikallaan. Tämä saattaa hidastaa huomattavasti prosessia, jos robottisolu ei vastaa ohjelmoijan ennakko-odotuksia tai mikäli sitä on tarvetta muuttaa jälkikäteen.

5.2 Johdattamalla ohjelmointi

Johdattamalla ohjelmointi (kuva 19) oli suosittu menetelmä kymmenen vuotta sitten, mutta sen jälkeen se on kadonnut lähes kokonaan. Johdattamalla ohjelmoinnin avulla robottia liikutetaan fyysisesti käyttäjän toimesta, mikä on tullut entistä haastavammaksi suurten teollisuusrobottien myötä. (Wilkins 2021.) Tätä menetelmää käytetään nykyään pääasiassa yhteistyörobottien ohjelmointiin, koska niiden käsivarret ovat pienempiä ja kevyempiä, mikä tekee niistä helpommin liikuteltavia (Robots Done Right 2021c.)



KUVA 19. ABB-robotin johdattamalla ohjelmointi (Eu Robotics 2014)

Johdattamalla ohjelmointi on hyödyllinen menetelmä yksityiskohtaisiin tehtäviin, jotka vaativat useita rivejä ohjelmoitaessa offline-sovelluksella. Johdattaminen ei kuitenkaan ole paras vaihtoehto esimerkiksi yksinkertaiseen keräilyyn ja pakkaamiseen. (Wilkins 2021.)

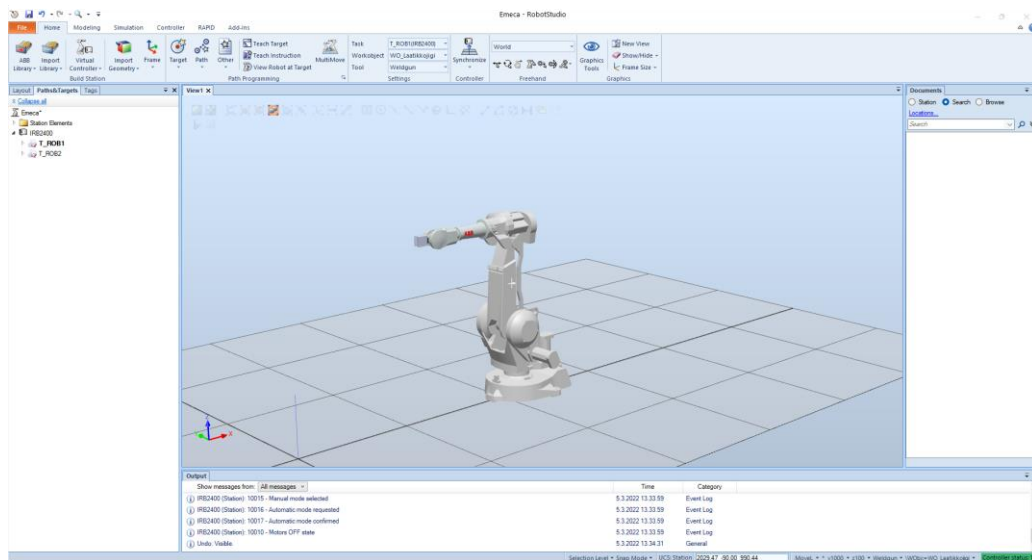
Johdattamalla ohjelmoinnin tärkein etu on sen yksinkertaisuus. Kuka tahansa voi ohjelmoida robottia johdattamalla, koska siihen ei tarvita lisälaitteita eikä käyttäjien tarvitse opetella erityistä ohjelmointia. Johdattaminen on myös nopeampaa kuin muut ohjelmointimenetelmät. Käyttäjät säästävät aikaa johdattamalla robottia sen sijaan, että syöttäisivät sovelluksen jokaisen liikekäsken manuaalisesti. (Robots Done Right 2021c.)

Johdattamalla ohjelmoinnin haittapuolina on sen monipuolisuuden ja tarkkuuden puute. Robotin liikuttaminen käsin on huomattavasti epätarkempaa kuin opettamalla- ja etäohjelmointi. Samoin kuin opettamalla ohjelmoinnissa on tuotannon seisakkien syntyminen väistämätöntä ja ohjelmien teko hidasta. (Robots Done Right 2021c.)

5.3 Offline-ohjelmointi

Offline- eli etäohjelmointi tarkoittaa robottiohjelmien luomista tietokoneen kautta ohjelmointiin tarkoitettujen ohjelmien ja simuloidun työympäristön avulla. Toisin kuin muissa ohjelmointimenetelmissä, offline-ohjelmointi ei vaadi robotin fyysistä läsnäoloa. (Robots Done Right 2021b.)

Offline-ohjelmointisovellukset ovat yleensä robottivalmistajien kehittämiä. ABB on luonut offline-sovelluksen RobotStudio (kuva 20), FANUCilla on oma Offline-ohjelmointisovellus ROBOGUIDE ja Yaskawa Motomanin robotteja voidaan ohjelmoida MotoSim-ohjelmalla. (Robots Done Right 2021b.)



KUVA 20. ABB RobotStudio

On olemassa myös universaaleja offline-sovelluksia, jotka tukevat lähes kaikkia robottivalmistajia. Esimerkiksi yksi maailman johtavista offline-ohjelmointisovelluksista on Delfoi valmistama ja se on suunniteltu toimimaan yleisimmillä robotibrändeillä, kuten ABB, FANUC, KUKA ja Yaskawa. (Delfoi n.d.)

Offline-ohjelmointia ja simulointia käytetään useimmiten robotiikassa edistyneiden ohjausalgoritmien toimivuuden varmistamiseksi ennen kuin niitä testataan oikeilla roboteilla. Ohjelmat luodaan offline-tilassa tietokonesovelluksella, eikä tuotannossa olevaa robottia tarvitse pysäyttää, kuin uuden ohjelman lataamisen ja testaamisen ajaksi. (Wilkins 2021.)

Offline-ohjelmoinnin hyödyt perustuvat uusien ohjelmien luomiseen ja simulointiin ilman tuotannon seisakkeja. Monimutkaisissa robottisovelluksissa etäohjelmoinnin ja simuloinnin avulla voidaan välttää ongelmat, kuten törmäykset ja akseleiden rikkoutumiset. (Delfoi n.d.)

Etäohjelmoinnin avulla käyttäjä voi suunnitella ja testata erilaisia konfiguraatioita 3D-simulaatiossa ennen robotin asentamista. Käyttäjä voi suunnitella, tunnistaa ja ratkaista ongelmia jo ennen robotin saapumista. (Robotics Tomorrow 2021.)

Offline-ohjelmoinnin heikkoutena on siitä aiheutuvat lisäkustannukset ja ohjelmoinnin haasteellisuus. Siihen tarkoitettujen ohjelmointisovellukset myydään yleensä lisävarusteena ja niiden käyttäjän mahdolliset koulutus- ja palkkakulut tuovat huomattavan määrän lisäkustannuksia verrattuna muihin ohjelmointitapoihin. Virtuaalisesti luodut mallit eivät luultavasti koskaan pysty täysin vastaamaan todellista maailmaa ja ohjelmia on usein vielä muutettava sen jälkeen, kun ne on jo otettu käyttöön oikeissa roboteissa. (Robotiq n.d.)

Vaikka offline-ohjelmointi vähentää robotin seisakkiaikaa, on jonkun käytettävä ylimääräistä aikaa simulaation kehittämiseen sekä testaukseen. On mahdollista myös hukata runsaasti aikaa simulaatioiden ongelmien selvittämiseen tuotanto-haasteiden ratkaisemisen sijaan. (Robotiq n.d.)

5.4 Joustava ohjelmointi

Joustava automaatio tarkoittaa saman robotin tai järjestelmän kykyä suorittaa erilaisia tehtäviä. Tehtävästä toiseen vaihdettaessa riittää yksinkertainen ohjelmistomuutos tai uuden arvon syöttäminen robotin ohjaimeen. (Robotpark 2015.)

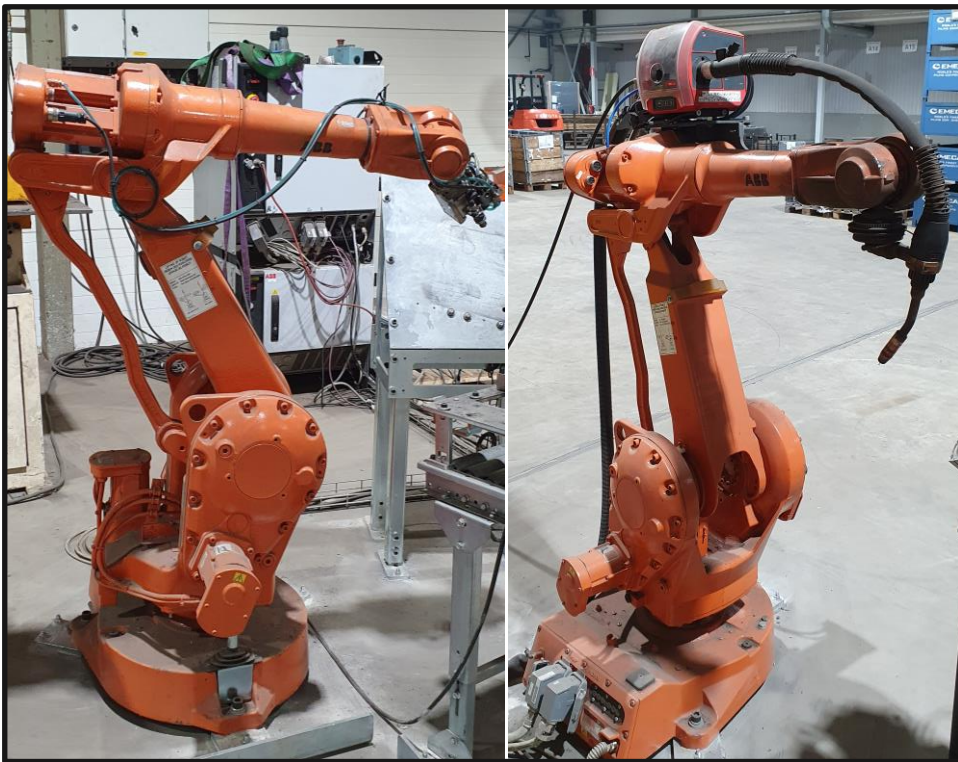
Joustava ohjelmointi on nopeaa prosessinvaihtoa varten. Sen avulla tuotantoaika ei menetetä muutettaessa valmistettavaa tuotetta, koska tarvittavat ohjelmien muutokset voidaan suorittaa nopeasti nappia painamalla. (Miller 2017.)

Esimerkki joustavasta automaatiosta on robotti, joka voidaan ohjelmoida asentamaan ruuveja, poraamaan reikiä ja hiomaan sekä hitsaamaan kokoonpanolinjan esineitä. Robottien kehittyessä, on mahdollista tehdä monia tehtäviä ainoastaan yhden ohjelman avulla. (Robotpark 2015.)

6 JOUSTAVAN HITSAUSSOLUN SUUNNITTELU

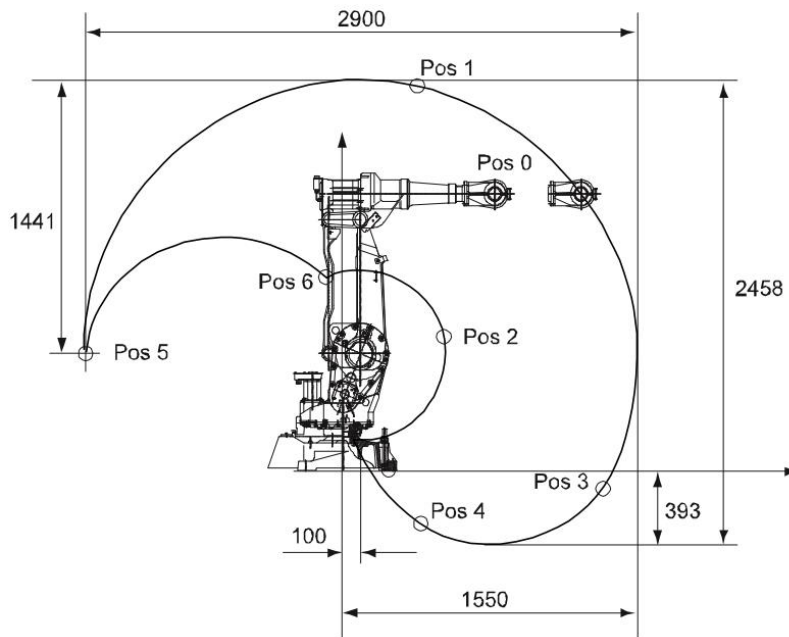
6.1 Lähtötilanne

Opinnäytetyö lähti liikkeelle Emecan hankittua kaksi kuusiakselista ABB IRB 2400/10 kiertyvänivelistä robottia (kuva 21). Robotit hankittiin paalujatkosten jatkoslaatikoiden valmistusta varten. Tarkoituksena oli suunnitella jatkoslaatikoiden valmistukseen soveltuvan hitsaussolun malli ja joustava robottien ohjelmointi. Hitsaussolun tavoitteena oli valmistaa erikokoisia jatkoslaatikoita nopealla mukautumiskyvyllä.



KUVA 21. ABB IRB 2400/10 robotit

Emecan hankkimat robotit saapuivat suoraan tuotannosta poistettuina eli ne olivat valmiiksi varusteltuja. Robottien kantokyky oli 10 kg ja yltämä 1550 mm (kuva 22). Toinen roboteista oli varustettu hitsauspolttimella ja toinen kappaleenkäsittelyyn soveltuvalla pneumaattisella magneettitarttujalla.



KUVA 22. ABB IRB 2400/10 robotin liikealue (ABB 2021c, 35)

Robottien ohjaus oli toteutettu IRC5 ohjaimella kahden ohjauskaapin yhdistelmällä eli Dual Cabinet controllerilla (kuva 23), yhdellä FlexPendant-käsiohjaimella ja MultiMove-optiolla. Robotin ohjelmisto sisälsi molemmille roboteille oman tehtävän, joten kummankin robotin ohjelmoiminen ja liikuttaminen voitiin suorittaa yhdellä käsiohjaimella.



KUVA 23. IRC5 Dual Cabinet controller

Hitsausrobotti oli varustettu Fronius TPS 400i hitsausvirtalähteellä (kuva 24), langansyöttökoneistolla ja poltinkaulalla. Hitsausvirtalähde oli nestejäähdytetty ja sen maksimivirta oli 400 A (ampeeria).



KUVA 24. Fronius TPS 400i

Kappaleenkäsittelijän tarttujana toimi IXTUR MRP-46 paineilmamagneetti (kuva 25). Magneetin lisäksi tarttujassa oli induktiivinen anturi sekä laseretäisyysanturi, joiden avulla voitiin tunnistaa kappale sitä lähestyttäessä.



KUVA 25. Kappaleenkäsittelyrobotin tarttuja

6.2 Hitsattava tuote

Hitsaussolu suunniteltiin neliön muotoisen jatkoslaatikon valmistukseen. Jatkoslaatikko on osa Emecan paalujatkoksen valmistusta. Laatikko valmistettiin hitsaamalla taivutetun kauluksen päät yhteen, jonka jälkeen valmis kaulus hitsattiin pohjalevyyn kiinni neljältä sivulta kuvan 26 mukaisesti.

Hitsaussolussa valmistettävien tuotteiden koko päätettiin rajata 235 mm ja 500 mm välille toimeksiantajan puolesta. Hitsattavat jatkoslaatikot olivat tällöin riittävän suuria yrityksen tarpeeseen, muttei kuitenkaan liian painavia robotin käsitte-lyyn.



KUVA 26. Jatkoslaatikko ja sen osat

6.3 Hitsaussolun malli

Tässä kappaleessa käydään läpi hitsaussolun suunnittelu sekä mallinnus. Hitsaussolun suunnittelu aloitettiin selvittämällä jatkoslaatikoiden valmistukselle asetetut tavoitteet, jonka jälkeen aloitettiin varsinainen suunnittelutyö.

Mekaniikkasuunnittelu toteutettiin SolidWorks 3D-CAD-ohjelmistolla ja se sisälsi hitsausjigien, valmistuotekuljettimen ja solun kokoonpanon 3D-mallinnuksen. Suunnittelussa otettiin huomioon joustava ohjelmointi tekemällä ratkaisuja sen tukemiseksi.

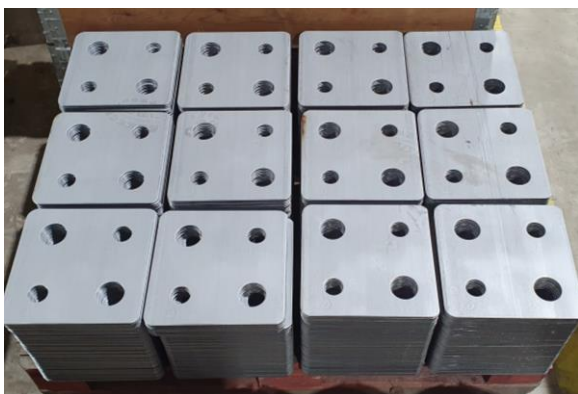
6.3.1 Hitsaussolun vaatimukset

Hitsaussolulle määritettiin vaatimukset projektin alussa. Pää tavoitteena oli suunnitella hitsaussolusta mahdollisimman joustava, jotta pienienkin erien valmistus olisi mahdollista menettämättä tuotannon tehokkuutta. Tarkoituksena oli suunnitella hitsausjigleistä helposti ja nopeasti säädettäviä. Asetusten tekoon ja hitsausjigien säätämiseen kuluva aika pyrittiin myös optimoimaan.

Hitsaussolun vaatimukset:

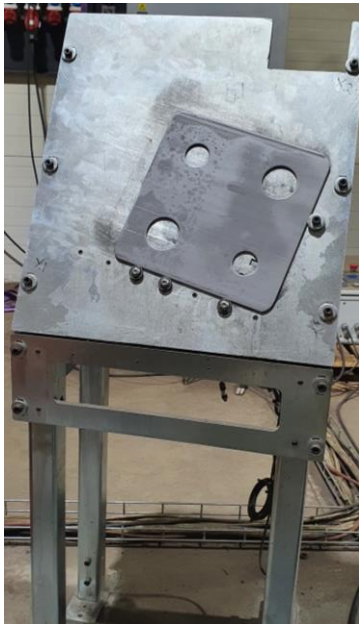
- Jatkoslaatikoiden koko 235–500 mm (pakollinen)
- Hitsausjigit mahdollisimman nopeasti ja helposti säädettäviä (pakollinen)
- Jatkoslaatikoiden pohjalevyt haetaan lavalta yksitellen robotilla (pakollinen)
- Pohjalevyt paikoitetaan valmiissa paikoitustelineessä (pakollinen)
- Valmiit tuotteet pinotaan robotilla EUR-lavalle (pakollinen)
- Kaulus haetaan robotilla taivuttimelta ja sen päät hitsataan yhteen (pakollinen)
- Kaulus hitsataan pohjalevyyn kiinni sisäpuolelta noin 35 % (pakollinen)

Pohjalevyaihiot tuli noutaa robotilla EUR-lavalta (kuva 27). Nouto suunniteltiin toteutettavaksi robotin magneettitarttujalla. Lavalla olevien pinojen määrä ei ollut vakio, vaan muuttui pohjalevyn koon mukaan. Levyjen tunnistamiseen suunniteltiin käytettäväksi laseretäisyysanturia ja induktiivista anturia, jotka olivat osa robotin tarttujan varustelua. Laseranturin avulla olisi mahdollista havaita levypinojen sijainti sekä korkeus ja induktiivisella anturilla tartuttavan levyn pinta.



KUVA 27. Pohjalevyaihiot EUR-lavalla

Lavalta noudettaessa pohjalevy tuli viedä kuvan 28 mukaiseen paikoitustelineeseen, joka varmisti tarkan tarttumiskohdan robotille sekä pudotti ylimääräiset levyt pois. Paikoitusteline oli yrityksellä jo valmiina, eikä sitä ollut tarvetta suunnitella uudelleen.



KUVA 28. Paikoitusteline

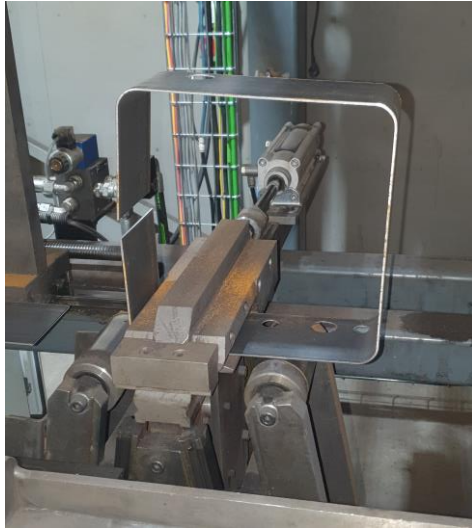
Valmiit jatkoslaatikot oli tarkoitus pinota robotilla EUR-lavalle kuvan 29 mukaisesti. Haluttujen pinojen ja rivien määrä sekä korkeus tuli voida määrittää käyttäjän toimesta robotin ohjelmassa.



KUVA 29. Jatkoslaatikoita EUR-lavalla

6.3.2 Kaulustaivutin

Jatkoslaatikoiden ”kaulus” valmistettiin erillisessä prosessissa Emecan tuotannossa olevalla kaulustaivuttimella. Kaulustaivuttimen tarkoitus oli ilmoittaa kappaleenkäsittelyrobotille, milloin kaulus voitiin noutaa, sekä vastaanottaa robotilta lupa aloittaa uuden kappaleen taivutus. Kaulustaivutin on esitetty kuvassa 30.



KUVA 30. Kaulustaivutin

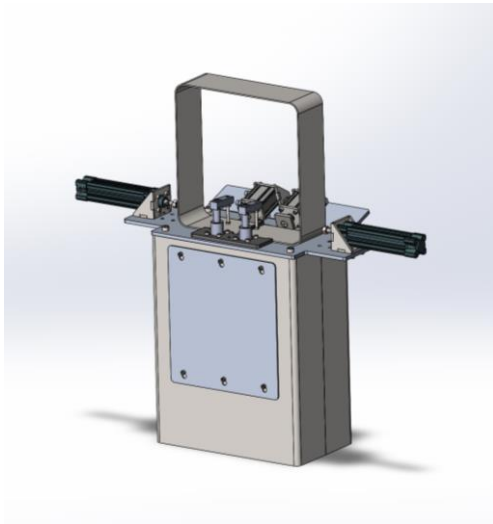
Kaulustaivuttimen toiminta pyrittiin pitämään erillisenä tuotantoprosessina, eikä sen toimintaan otettu kantaa projektin aikana. Taivuttimelta annettiin yksittäinen I/O- signaali robotille ja päinvastoin, jolloin robotin ja taivuttimen välinen kommunikointi voitiin pitää mahdollisimman yksinkertaisena.

6.3.3 Hitsausjigit

Tarvittavat hitsausjigit suunniteltiin ”kauluksen” päiden yhdistämiseen sekä valmiin jatkoslaatikon kokoonpanohitsausta varten. Jigit suunniteltiin helposti ja nopeasti säädettäväksi huomioiden robottien joustava ohjelmointi.

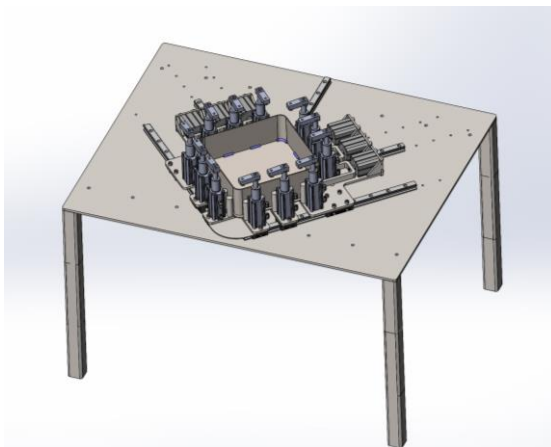
Kauluksen hitsausjigi (kuva 31) suunniteltiin ilman säätöjä siten, että valmiiksi taivutettu kaulus voitiin noutaa kaulustaivuttimelta ja viedä robotin magneettitartujan avulla hitsausjigiin sen koosta riippumatta. Jigi suunniteltiin paikoittamaan kauluksen sauma aina keskelle jigiä, hitsausta varten. Tällä mahdollistettiin tasa-laatuinen tarkkuus ja toistettavuus kauluksen hitsauksessa. Paikotus toteutettiin

alhaalta viistossa nousevalla johdesylinterillä, jonka päässä olevaa levyä vasten kaulus voitiin asettaa robotilla. Kauluksen lukitseminen paikoilleen toteutettiin sivuilta ja ylhäältä puristavilla paineilmasylintereillä. Hitsausjigi suunniteltiin 235–500 mm kokoisille kauluksille sopivaksi.



KUVA 31. SolidWorks-ohjelmalla mallinnettu kauluksen hitsausjigi

Jatkoslaatikoiden kokonpanojigi (kuva 32) suunniteltiin asennettavaksi jigipöydälle lineaarijohteiden ja johdekelkkojen päälle. Lineaarijohteiden avulla mahdollistettiin jigin joustava ja portaaton säätö valmistettavan laatikon koon mukaan.

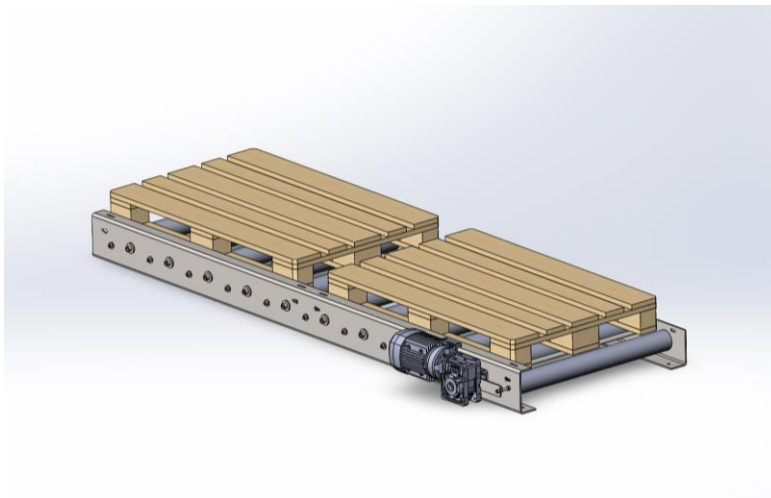


KUVA 32. SolidWorks-ohjelmalla mallinnettu jatkoslaatikon kokonpanojigi

Kokoonpanojigi suunniteltiin paikoittamaan jatkoslaatikoiden kaulus ja pohjalevy kohdakkain paineilmasylintereillä sivuilta sekä yläpuolelta. Paikoittamalla laatikon osat aina yhteen kiinteään kulmaan, hitsaus voitiin suorittaa luotettavasti robotilla.

6.3.4 Valmistuotekuljetin

Hitsaussolussa valmistuvien laatikkolavojen siirtämiseen suunniteltiin oikosulkumoottorilla varustettu ketjukäyttöinen rullakuljetin (kuva 33). Rullakuljetin mitoitettiin kahdelle EUR-lavalle sopivaksi mitoilla 2700 mm x 900 mm. Hitsaussoluun suunniteltiin asennettavaksi kaksi valmistuotekuljetinta peräkkäin, jolloin jatkoslaatikoita voitaisiin valmistaa puskuriiin useampi lava, mikä mahdollistaisi tuotannon jatkumisen pidempään miehittämättömästi.

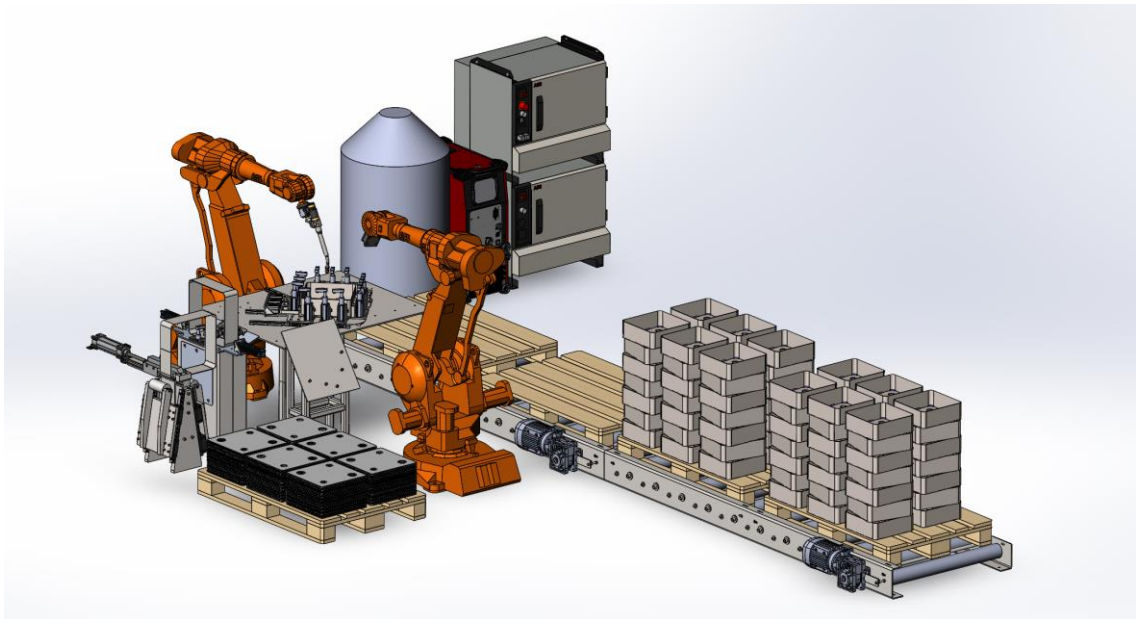


KUVA 33. SolidWorks-ohjelmalla mallinnettu rullakuljetin

Rullakuljettimen ohjaus suunniteltiin itsenäiseksi Siemens LOGO-logiikkaohjaimella ja lavan tunnistavilla optisilla antureilla. Ohjaus suunniteltiin siten, että lavan valmistuessa robotti antaa käskyn kuljettimelle, joka siirtää valmiin lavan eteenpäin seuraavan anturin näköpiiriin. Tyhjän lavan saapuessa robotti saa logiikalta tiedon, jolloin laatikoiden lavaus voidaan taas aloittaa. Kuljettimelle suunniteltiin myös mahdollisuus manuaaliseen ajoon painonapeilla.

6.3.5 Hitsaussolun kokoonpano

Solun kokoonpano (kuva 34) suunniteltiin mahdollisimman selkeäksi ja johdonmukaiseksi huomioimalla robottien työkiertojen toteutus. Hitsaussolu suunniteltiin sijoitettavaksi Emecan tuotannossa olevan vannelinjan perään, jossa kaulustavutin sijaitisi. Tällä ratkaisulla mahdollistettiin kauluksen nouto robotilla suoraan taivutuksesta. Kaulustavutin kuvan 34 vasemmassa laidassa.



KUVA 34. SolidWorks-ohjelmalla luotu hitsaussolun kokoonpano

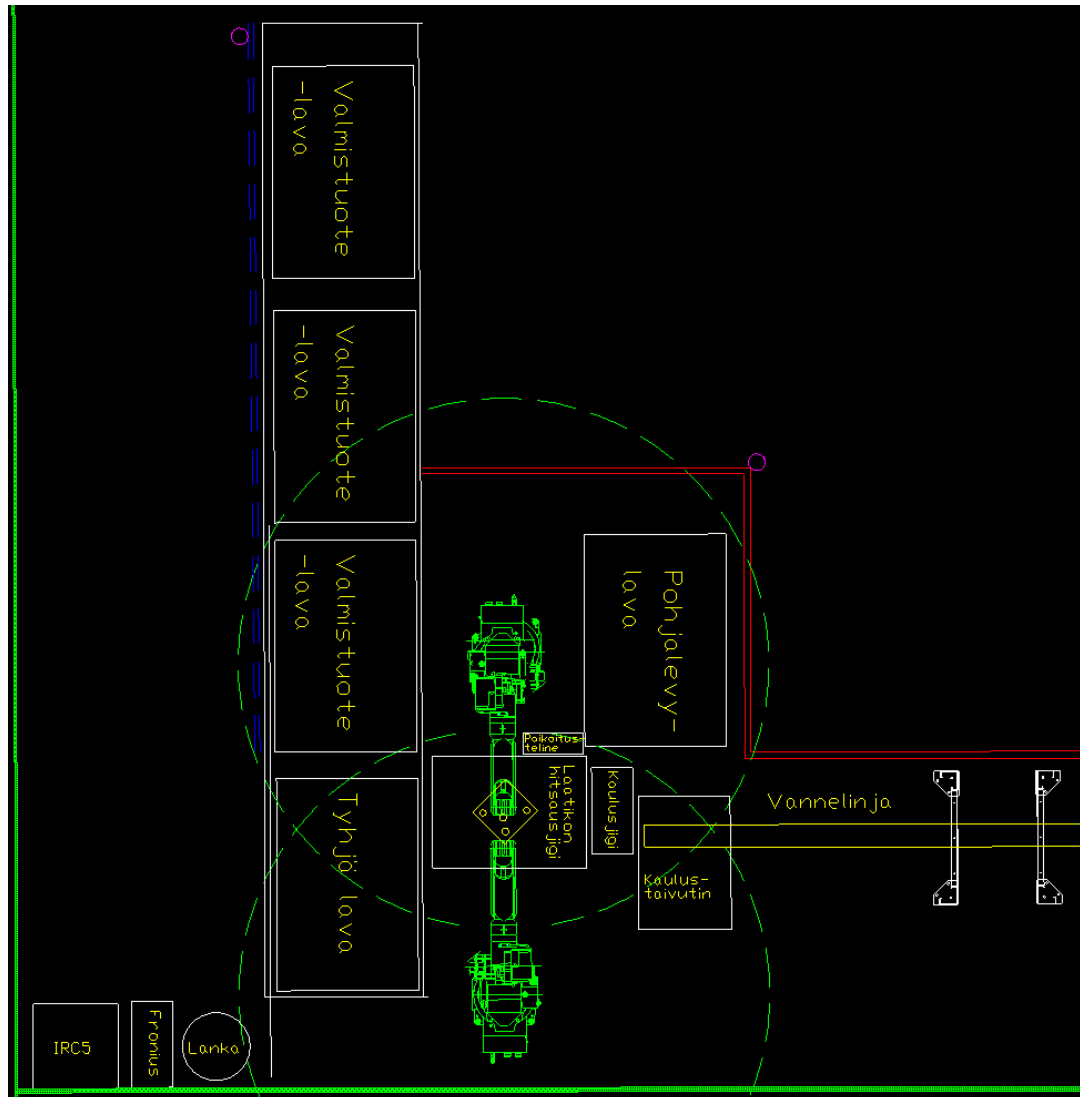
Robotit asetettiin vastakkain tilan säästämisen vuoksi ja jaetun työalueen muodostamiseksi. Pohjalevyt ja rullakuljettimet suunniteltiin sijoitettavan kappaleenkäsittelyrobotin sivuille, jolloin robotin yltämä saatiin hyödynnettyä tehokkaasti. Hitsausjigit sijoitettiin robottien väliin, jolloin kumpikin robotti voisi toimia samalla jaetulla työalueella.

Pohjalevyjen paikoitustelineen sijainti määritettiin kuvan 34 mukaisesti kokoonpanojigin ja hitsausrobotin väliin. Hitsausvirtalähde, IRC5-ohjauskaappi, hitsauslankatynnyri ja suojakaasupullon sijoitus suunniteltiin hitsaussolun ulkopuolelle, seinän viereen. Tällä mahdollistettiin vapaa kulku robotin ohjauskaapille tuotteiden valmistuksen aikana ja samalla suojattiin robotin ohjauskaappi ja käsiohjain hitsausroiskeilta.

6.4 Hitsaussolun turvallistaminen

Hitsaussolua suunnitellessa otettiin huomioon myös turvallisuustekijät. Solun turvallistaminen suunniteltiin turvaloverhojen, turva-aidan ja hätäseis-kytkimien avulla (luku 3.6). Kuvassa 35 on esitetty Hitsaussolun turvallistaminen. Turvaaidat on merkitty sinisellä katkoviivalla, valoverhot punaisella viivalla ja Hätäseis-kytkimet vaaleanpunaisina ympyröinä.

Molemmat robotit sekä valmistuotekuljetin suunniteltiin kytkettäväksi samaan hätäseis-piiriin, turvavaloverhojen ja hätäseis-kytkimien kanssa. Tällä varmistetaan työntekijöiden turvallinen liikkuminen hitsaussolun ympäristössä.



KUVA 35. Hitsaussolun turvallistaminen

7 JOUSTAVAN HITSAUSSOLUN OHJELMOINTI JA TESTAUS

7.1 Ohjelmoinnin suunnittelu

Ohjelmoinnin suunnittelu oli olennainen osa opinnäytetyötä. Sillä varmistettiin riittävän kattava käsitys ohjelmien vaatimuksista. Selkeät vaatimukset mahdollistivat johdonmukaisen ohjelmoinnin ja sen suunnittelun.

Suunnittelu aloitettiin selvittämällä hitsaussolun ohjelmoinnille asetetut vaatimukset, joiden perusteella valittiin sopivin ohjelmointitapa. Vaatimusmäärittelyn jälkeen suunniteltiin hitsaussolun työkierrat, joiden mukaan robottien ohjelmointi voitiin toteuttaa.

7.1.1 Ohjelmoinnin vaatimukset

Vaatimusmäärittelyn avulla voitiin varmistaa ohjelmoinnissa vaadittujen ominaisuuksien toteutuminen ja sen avulla pystyttiin säästämään ohjelmointiin käytettävää aikaa. Vaatimukset jaettiin kategorioihin pakollinen ja hyödyllinen. Kategorioiden avulla voitiin tunnistaa kriittisimmät vaatimukset ja tarpeet ohjelmien suunnittelussa. Vaatimukset esitetty taulukossa 3.

TAULUKKO 3. Ohjelmoinnin vaatimukset

Ohjelmoinnin vaatimukset:	Tärkeys: Pakollinen/Hyödyllinen
1. Selkeä ja yksinkertainen ohjelmarakenne	Pakollinen
2. Joutoajan minimointi	Pakollinen
3. Ohjelmat tarpeeseen mukautuvia ja uudelleenkäytettäviä	Pakollinen
4. Ohjelmat luotettavia ja turvallisia	Pakollinen
5. Modulaarinen ohjelmarakenne	Pakollinen
6. Mahdollisimman vähän opetettavia pisteitä	Hyödyllinen
7. Itsenäisiä ongelmanratkaisuja	Hyödyllinen

7.1.2 Ohjelmointitavan valinta

Hitsaussolun ohjelmointitapa valittiin asetettujen vaatimusten perusteella. Koska ohjelmat tuli suunnitella nopeasti tarpeeseen mukautuviksi, oli joustava ohjelmointi selkeä valinta (luku 5.4). Joustavan ohjelmoinnin myötä erikokoisten tuotteiden valmistus voitiin toteuttaa yhdellä ohjelmakokonaisuudella. Tämä mahdollisti selkeän ja yksinkertaisen ohjelmarakenteen sekä uudelleenkäytettävyyden.

Osa robotin sijainneista täytyi kuitenkin määrittää opettamalla, jonka myötä toiseksi ohjelmointitavaksi valikoitui opettamalla ohjelmointi. Koska ohjelmat suunniteltiin pääasiassa joustaviksi, oli opetettavien pisteiden määrä huomattavasti pienempi verrattuna perinteiseen opettamalla ohjelmointiin.

7.1.3 Työkiertojen suunnittelu

Hitsaussolun työkierto (luku 3.5) suunniteltiin siten, että kummankin robotin odotusaika saatiin minimoitua. Kappaleenkäsittelyrobotin työskennellessä, tuli hitsausrobotin voida hitsata joko kaulusta tai jatkoslaatikkoa.

Hitsaussolun työkierron rakenne:

1. Pohjalevy paikoitustelineestä laatikkojigiin, jos laatikkojigi on tyhjä
2. Kaulus hitsausjigistä laatikkojigiin, jos kaulus on hitsattu ja pohjalevy on jigissä -> Hitsausrobotti hitsaa laatikon
3. Kaulus taivuttimelta hitsaukseen, jos hitsausjigi on tyhjä ja kaulus on taivutettu -> Hitsausrobotti hitsaa kauluksen
4. Pohjalevyn nouto lavalta paikoitustelineeseen -> jos paikoitusteline on tyhjä
5. Valmiin laatikon vienti lavalle, jos laatikko on hitsattu

Työkierto alkaa, kun kappaleenkäsittelijä noutaa pohjalevyn paikoitustelineestä ja vie kokoonpanojigiin. Seuraavaksi noudetaan edellisellä työkierrolla hitsattu kaulus, joka viedään pohjalevyn päälle kokoonpanojigiin. Tässä vaiheessa hitsausrobotti aloittaa laatikon hitsauksen. Hitsauksen aikana kappaleenkäsittelijä vie valmiiksi taivutetun kauluksen hitsattavaksi ja noutaa pohjalevyn lavalta ja vie sen paikoitustelineeseen. Laatikon valmistuessa hitsausrobotti siirtyy kauluksen

hitsaukseen ja kappaleenkäsittelijä vie sillä välin valmiin laatikon lavalle, jonka jälkeen työkierto voi alkaa alusta.

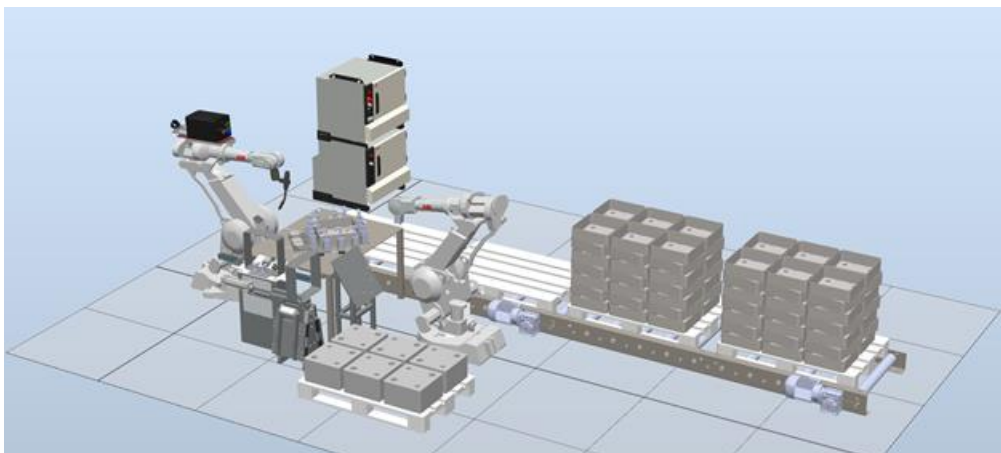
7.2 Ohjelmointi RobotStudiolla

Hitsaussolun ohjelmointi suunniteltiin aluksi toteutettavan robottien käsiohjaimella, mutta siitä luovuttiin, koska hitsaussolua ei ehditty fyysisesti rakentamaan opinnäytetyön aikana. Siksi ohjelmointi päätettiin toteuttaa ABB:n RobotStudiolla (luku 5.3). RobotStudiolla oli mahdollista testata ohjelmien toimivuus virtuaalisesti ja sillä saatiin testattua erilaisia ohjelmaratkaisuja ja robottien konfiguraatioita, vaikka hitsaussolua ei ollut vielä fyysisesti rakennettu. Tämä edesauttoi havaitsemaan mahdollisia epäkohtia ja haasteita ohjelmoinnista.

Virtuaalinen malli

Ohjelmointi aloitettiin luomalla RobotStudioon virtuaalinen malli hitsaussolusta. Koska robotit olivat fyysisesti jo olemassa, voitiin robotit luoda todellisen IRC5-ohjaimen varmuuskopion pohjalta. RobotStudioon luotiin uusi robottiasema varmuuskopiosta, jolloin kaikki ohjaimen sisältämä data robotit mukaan lukien voitiin avata RobotStudiolla.

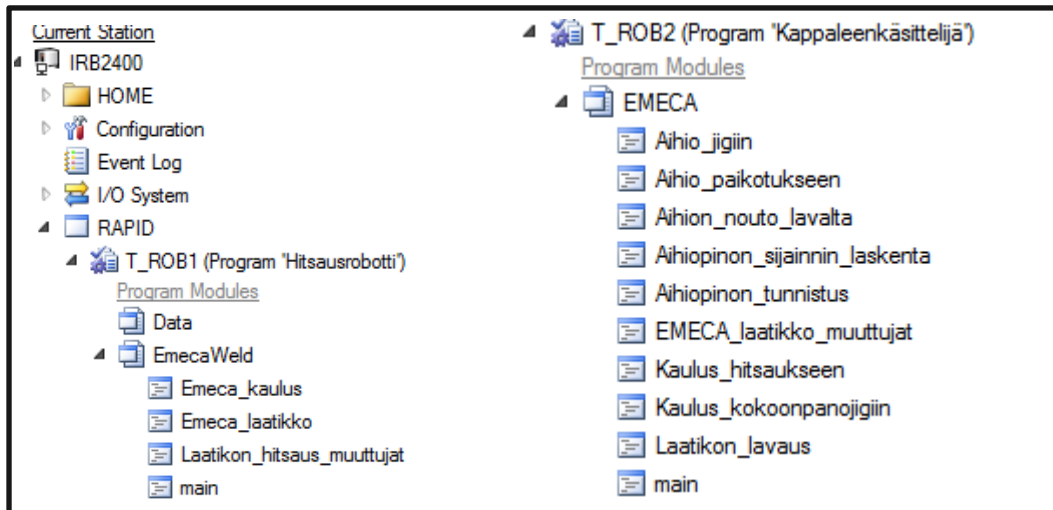
Hitsaussolun jigat ja kuljettimet tuotiin robottiasemaan step-tiedostoina SolidWorks-ohjelmasta. Step-tiedostot sijoitettiin robottien kanssa todellisuutta vastaaville etäisyyksille. RobotStudiolla luotu virtuaalinen hitsaussolun malli kuvassa 36.



KUVA 36. Hitsaussolun virtuaalinen malli

Moduulit ja rutiinit

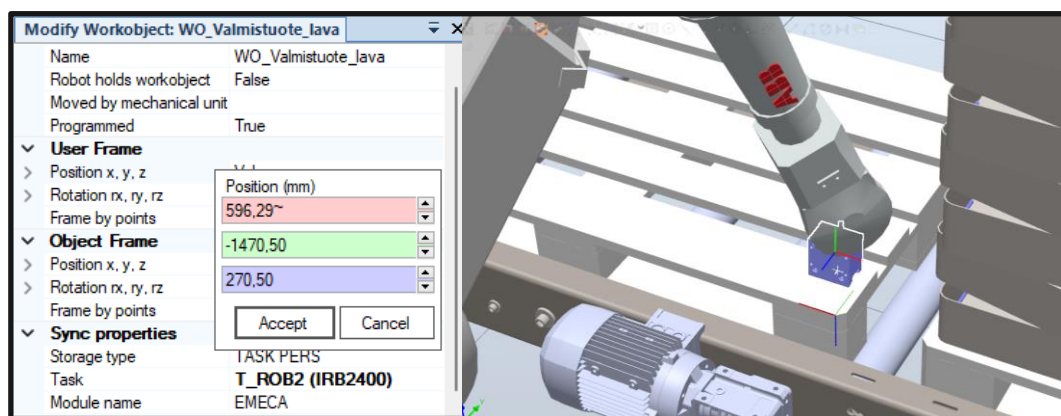
Ohjelmointi aloitettiin määrittämällä ja nimeämällä molemmille roboteille RobotStudioissa omat ohjelmamoduulit EmecaWeld ja EMECA (kuva 37). Moduulien sisälle luotiin aiemmin suunniteltujen työkiertojen mukaiset rutiinit. Yksi kummankin robotin rutiineista nimettiin pakolliseksi päärutiiniksi "main", joka toimi ohjelman aloitusrutiinina. Koska kappaleenkäsittely- ja hitsausrobotille luotiin rutiinit jokaiselle eri työvaiheelle, saatiin ohjelmarakenteesta selkeä ja helppolukuinen.



KUVA 37. Moduulit ja rutiinit

Koordinaatistojen määrittäminen

Seuraavaksi määritettiin työkohteen koordinaatistot (luku 4.3) hitsaussolun osille, kuten jigille ja lavoille (kuva 38). Robottien työkalukoordinaatistot ja työkalupisteet saatiin varmuuskopioiden mukana, eikä niitä ollut tarvetta määrittää uudelleen. Luoduilla koordinaatistoilla varmistettiin, että oikeat sijainnit säilyvät, vaikka jigien tai robottien fyysistä paikkaa muutettaisiin.



KUVA 38. Koordinaatistojen määrittäminen RobotStudiolla

Muuttujien määrittäminen

Ohjelmaan määritettiin useita numeerisia- ja totuusarvomuuttujia esimerkiksi valmistettavan tuotteen koon määrittämiseen sekä valmiiden tuotteiden pinoamiseen lavalle (liitteet 1 ja 2). Muuttujat määritettiin pysyviksi (PERS), koska niiden ei haluttu nollaantuvan aloitettaessa ohjelma alusta.

Työkierron suorittaminen tietyssä järjestyksessä varmistettiin luomalla ohjelmiin totuusarvomuuttujat, joiden loogista arvoa muutettiin rutiinin alkaessa ja päättyessä ohjelman toimesta. Jokaisen rutiinin alussa tarkistettiin, toteutuuko vaaditut ehdot ja onko edellinen vaihe suoritettu, jotta ohjelmaa voidaan jatkaa. Tällä varmistettiin robottien turvallinen eteneminen ohjelmassa.

Hitsausrobotin ja kappaleenkäsittelyrobotin toimiminen yhteisellä työalueella otettiin myös huomioon ohjelmien suunnittelussa. Roboteille suunniteltiin kättelyt, joiden avulla voitiin varmistaa, etteivät robotit päässeet törmäämään toisiinsa. Kättelyt toteutettiin määrittämällä globaalit muuttujat robottien välille. Muuttujien avulla voitiin koordinoita, milloin hitsausrobotti saa aloittaa hitsauksen ja koska hitsaus tulee valmiiksi. Kappaleenkäsittelyrobotin tuli odottaa kappaleen noutoa niin pitkään, kunnes hitsausrobotti oli turvallisesti pois työalueelta ja sama toisinpäin.

Tulo- ja lähtösignaalit

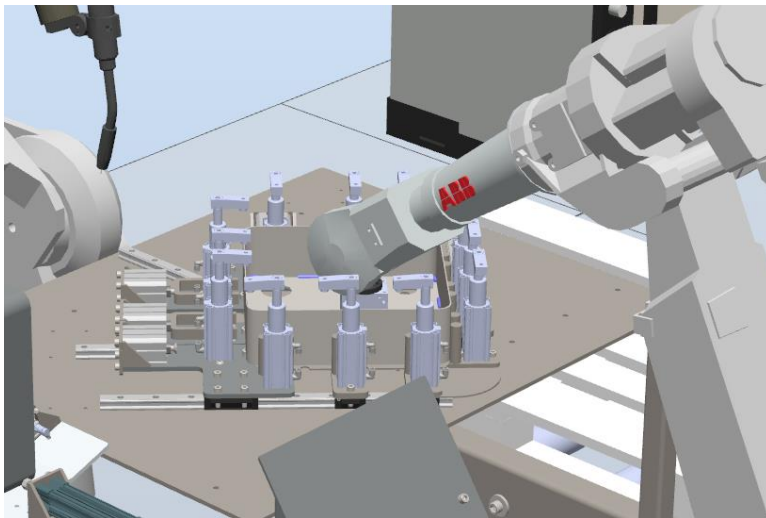
Roboteille tuli määrittää myös digitaaliset tulo- ja lähtösignaalit, jotta tarvittavien toimilaitteiden ohjaaminen onnistuisi. Osa I/O-signaaleista saatiinkin tuotua valmiina varmuuskopion pohjalta luodun virtuaalisen kontrollerin mukana. Valmiiksi konfiguroidut signaalit olivat kappaleenkäsittelyrobotin magneettitarttujan lähtösignaali sekä induktiivisen anturin ja laseretäisyysanturin tulosignaalit. Loput tarvittavat signaalit konfiguroitiin RobotStudiolla taulukon 4 mukaisesti.

TAULUKKO 4. Robottien digitaaliset tulo- ja lähtösignaalit

Tulosignaali:	Lähtösignaali:
diKaulusTaitettu	do_Kauluksen_puristajat
diLavausAloitus	do_Painajat_ltk
diLaseranturi	doLavanvaihto
diTarttutaraja	doLupataittaa
diSaaLähteä	doMagneettitarttuja
diTyökierronlopetus	doTarttujakiinni

Joustava ohjelmointi

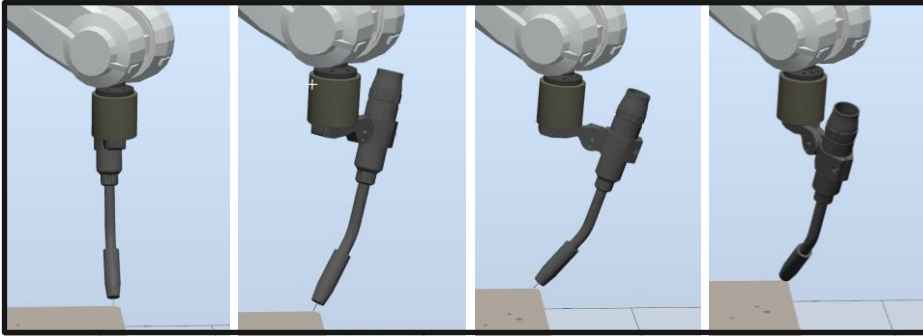
Hitsaussolun ohjelmointi toteutettiin joustavaksi opettamalla roboteille tietyt kiinteät paikkapisteen, joita voitiin käyttää referenssinä riippumatta valmistettavan tuotteen koosta. Esimerkiksi kuvan 39 poimintapiste pysyi samana, vaikka hitsattavan laatikon koko muuttui. Ohjelmat rakennettiin siten, että opettavien pisteiden määrä pyrittiin pitämään mahdollisimman vähäisenä. Tämä toteutettiin käyttämällä samoja paikkapisteitä useissa eri ohjelman vaiheissa.



KUVA 39. Kappaleenkäsittelyrobotille opetettu muuttumaton piste

Joustavat ohjelmaratkaisut toteutettiin numeeristen muuttujien ja Offset-funktioiden avulla, jolloin robotti saatiin siirtymään haluttuun sijaintiin ilman uusien pisteiden luomista. Tätä hyödynnettiin esimerkiksi luomalla hitsausrobotille nollapisteen kuvan 40 mukaisesti. Pisteet luotiin kaikille hitsaukseen vaadittaville astekulmille, joilla mahdollistettiin laatikon sivujen sekä kulmien hitsaus. Näiden nolla-

pisteiden yhdistäminen liikekäskyjen OffSet- ja RelTool-komentoihin, voitiin pisteitä siirtää halutun koordinaatiston x, y ja z- suunnissa ennalta määriteltyjen muuttujien lukuarvoilla. Tämä mahdollisti jatkoslaatikon hitsauspisteiden luonnin perustuen pelkästään nollapisteisiin ja käyttäjän määrittämiin numeerisiin muuttujiin.



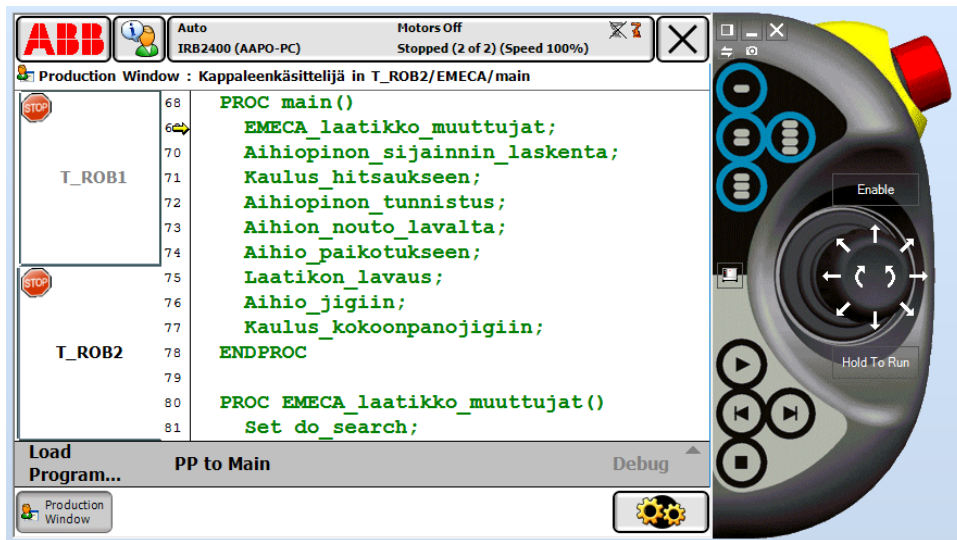
KUVA 40. Hitsausrobotin nollapisteet

Valmistettavan tuotteen kokoa muutettaessa, käyttäjä pystyi asettamaan numeeriselle muuttujalle uuden lukuarvon robotin käsiohjaimesta, jolloin hitsauspisteet päivittyivät automaattisesti ohjelmassa muuttujan arvon mukaisille paikoille. Tämä mahdollisti saman hitsausohjelman käytön erikokoisille jatkoslaatikoille.

Käyttäjän määritettävissä olevat numeeriset lukuarvot olivat valmistettavien laatikoiden koko, korkeus, kauluksen paksuus, lavalla olevien pohjalevyjen rivit ja sarakkeet sekä valmistuotekuljettimelle pinottavien jatkoslaatikoiden rivit, sarakkeet ja kerrokset. Loput muuttujien määrytyksistä suoritettiin automaattisesti erillisillä laskentarutiineilla, jotka luettiin ohjelman jokaisella työkierrolla (liitteet 1 ja 2).

7.3 Simulointi ja testaus RobotStudiolla

Ohjelmien toimivuus testattiin simuloimalla ohjelmat RobotStudion virtuaalisella FlexPendant-käsiohjaimella. Virtuaalinen käsiohjain vastasi täysin todellista käsiohjainta, joten ohjelmien testaus oli luotettavaa ja todenmukaista. Kuvassa 41 on esitetty virtuaalinen käsiohjain RobotStudiolla.



KUVA 41. Virtuaalinen käsihjain

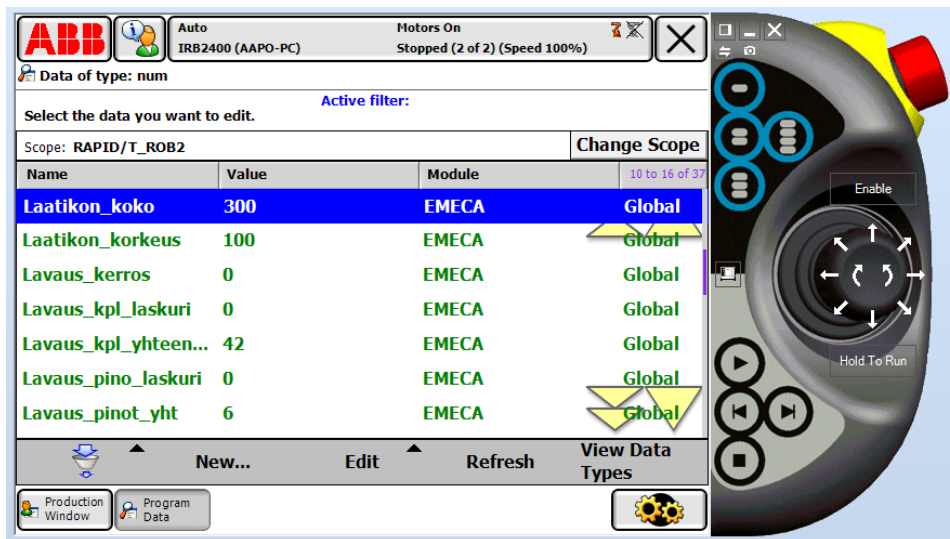
Simuloinnin avulla saatiin selkeämpi käsitys työkiertoista sekä mahdollisista ongelmista ohjelmakoodissa. Simulaatioon perustuen voitiin myös testata ja laskea työkiertojen sykliajat erikokoisten tuotteiden valmistukselle.

7.3.1 Työkiertojen simulointi ja testaus

Simulointi aloitettiin asettamalla virtuaaliseen käsihjaimeen tarvittavat muuttujien arvot valmistettavalle tuotteelle, jonka jälkeen varmistettiin, että ohjelmien aloituspiste on kummankin robotin pääohjelmassa. Tämän jälkeen valittiin manuaalinen käsiajo, jolla ohjelmien simulointi voitiin aloittaa. Ohjelmat simuloitiin aluksi 350 mm kokoisella jatkoslaatikolla, jolla saatiin yleiskäsitys ohjelmien toiminnasta.

Ongelmaksi havaittiin heti ensimmäisellä työkierrolla työkohteen koordinaatistojen suunnat. Pohjalevyllä ja laatikon hitsausjigin koordinaatistoissa x- ja y- akselit olivat väärinpäin, mitä alun perin oli suunniteltu. Tämän johdosta robotin Offset- pisteet olivat väärissä sijainneissa. Ongelma ratkaistiin koordinaatistojen uudelleenmäärittämisellä. Koska kumpikin koordinaatisto sisälsi vain kiinteitä nollapisteitä, ei paikkapisteiden uudelleenmäärittäminen ollut tarpeellista. Korjausten jälkeen ohjelma saatiin testattua ja simuloitua onnistuneesti alusta loppuun, liitteen 3 videon mukaisesti.

Onnistuneen simuloinnin myötä testattiin vielä joustavien ohjelmataratkaisujen toimivuus. Valmistettava tuote määritettiin 50 mm pienemmäksi eli 300 mm kookseksi (kuva 42), jonka jälkeen ohjelman simulointi aloitettiin alusta. Kokoa muutettaessa ohjelma laski liikepisteiden sijainnit suunnitellusti päivitettyjen muuttujien arvoilla, jolloin esimerkiksi hitsauspisteet vastasivat valmistettavan jatkoslaatikon uutta kokoa. Liitteen 4 videolla on esitetty joustavan hitsausohjelman simulointi 350 ja 300 mm kokoisilla tuotteilla. Onnistuneilla simuloinneilla saatiin varmistettua ohjelmakoodien toimivuus.



KUVA 42. Valmistettavan tuotteen koon määrittäminen käsiohjaimesta

7.3.2 Jatkoslaatikoiden valmistus

Jatkoslaatikoiden valmistukseen kuluva aika selvitettiin ja laskettiin perustuen simulointeihin. Työkiertojen ja hitsausaikojen laskenta toteutettiin kolmella erikoisella jatkoslaatikolla ja luvut taulukoitiin Microsoft Exceliin. Tarkasteluun otettiin koot 235 mm, 270 mm ja 350 mm. Hitsausprosentiksi asetettiin vaadittu 35, mikä tarkoitti hitsausauman osuutta laatikon kehän pituudesta. Hitsausajan laskentaan kertoimeksi asetettiin 0,8, joka vastasi 80 % kaariaikasuhdetta (luku 3.4).

Hitsausaika määritettiin hitsausnopeuden 12 mm/s mukaan, josta saadut tulokset on esitetty taulukossa 5. Tuloksista voitiin huomata, että valmistettavan tuotteen koon kasvattaminen lisäsi hitsausrobotin työkierron pituutta, mikä johti kappaleenkäsittelyrobotin odotusajan kasvuun.

TAULUKKO 5. Työkiertojen laskenta

Jatkoslaatikoiden valmistus		Hitsaus %	kerroin	Kauluksen hitsaus	
		35	0.8	15.3	
235	936	270	1076	350	1368
Hitsausaika s/kpl Laatikko	mm/s	Hitsausaika s/kpl Laatikko	mm/s	Hitsausaika s/kpl Laatikko	mm/s
49.5	12	54.6	12	65.2	12
Hitsausaika s/kpl Kaulus	mm/s	Hitsausaika s/kpl Kaulus	mm/s	Hitsausaika s/kpl Kaulus	mm/s
8.3	12	8.3	12	8.3	12
Hitsausrobotin työkierto		Hitsausrobotin työkierto		Hitsausrobotin työkierto	
57.8		62.9		73.5	
Kappaleenkäsittelijän työkierto	Sekuntia	Kappaleenkäsittelijän työkierto	Sekuntia	Kappaleenkäsittelijän työkierto	Sekuntia
Valmis laatikko lavalle	10	Valmis laatikko lavalle	10	Valmis laatikko lavalle	10
Kauluksen haku taivutuksesta	5	Kauluksen haku taivutuksesta	5	Kauluksen haku taivutuksesta	5
Kauluksen vienti hitsaukseen	8	Kauluksen vienti hitsaukseen	8	Kauluksen vienti hitsaukseen	8
Pohjalevy paikoituksesta laatikkojigiin	6	Pohjalevy paikoituksesta laatikkojigiin	6	Pohjalevy paikoituksesta laatikkojigiin	6
Kaulus laatikkojigiin	7	Kaulus laatikkojigiin	7	Kaulus laatikkojigiin	7
Pohjalevy lavalta paikoitukseen	8	Pohjalevy lavalta paikoitukseen	8	Pohjalevy lavalta paikoitukseen	8
Yht.	44	Yht.	44	Yht.	44
Kappaleenkäsittelijän odotusaika /s		Kappaleenkäsittelijän odotusaika /s		Kappaleenkäsittelijän odotusaika /s	
13.8		18.9		29.5	
Valmiita laatikoita per lava	Kpl	Valmiita laatikoita per lava /kpl	Kpl	Valmiita laatikoita per lava /kpl	Kpl
Rivejä	3	Rivejä	3	Rivejä	2
Sarakkeita	5	Sarakkeita	4	Sarakkeita	3
Kerroksia	7	Kerroksia	7	Kerroksia	7
Yht.	105	Yht.	84	Yht.	42
Valmis laatikkolava (min)		Valmis laatikkolava (min)		Valmis laatikkolava (min)	
86.55208333		76.3875		45.64583333	

Lisäksi taulukossa laskettiin täyteen laatikkolavaan kuluva aika. Aika laskettiin lavalle pinottavien laatikoiden määrään ja hitsaussolun työkiertoon perustuen. Tällä laskennalla pystyttiin arvioimaan kauanko hitsaussolu pystyisi teoriassa tuottamaan valmiita jatkoslaatikoita miehittämättömästi.

7.3.3 Ohjelmien käyttöönotto

Hitsaussolua ei valmistettu fyysisesti vielä opinnäytetyön aikana, joten suunniteltujen ohjelmien käyttöönotto siirtyi myöhempään ajankohtaan. Käyttöönottovaiheessa tulee työkohteen koordinaatit osoittaa roboteilla fyysisen hitsaussolun jigeihin ja lavoihin, jotta robottien liikeradat saadaan varmistettua oikeiksi. Koska ohjelmat luotiin ja simuloitiin offlinessa, voitiin tarvittavat ohjelmamuutokset käyttöönottovaiheessa tehdä RobotStudiolla.

8 POHDINTA

Opinnäytetyön tavoitteena oli suunnitella joustava robottihitsaussolu ja sen ohjelmointi Emeca Oy:n osatuotantoon. Tarkoituksena oli pyrkiä suunnittelemaan robottiohjelmista mahdollisimman joustavia ja asiakkaan tarpeisiin mukautuvia menettämättä tuotannon tehokkuutta.

Opinnäytetyö onnistui asetettujen tavoitteiden mukaisesti. Työssä selvitettiin kattavasti teollisuusrobottien käyttöä, ohjelmointia ja ohjelmointitapoja, joiden pohjalta onnistuttiin suunnittelemaan joustavaa ohjelmointia tukevat robottiohjelmat. Työstä saatujen tulosten perusteella voidaan todeta joustavan ohjelmoinnin hyödyttävän jatkoslaatikoiden valmistusprosessia huomattavasti. Jos ohjelmointia ei olisi toteutettu joustavaksi, ohjelmia olisi kerääntynyt valtava määrä, eikä niitä olisi ollut enää mahdollista ylläpitää järkevästi. Joustavan ohjelmoinnin avulla ohjelmakanta onnistuttiin luomaan yksinkertaiseksi ja selkeäksi.

Robottiohjelmien simuloinnilla varmistettiin suunniteltujen ohjelmien toimivuus ja sen avulla saatiin todellista dataa työkiertojen sykliajoista ja mahdollisista ongelmakohtista. Simuloinnilla voidaankin todeta olevan suuri merkitys koko projektin kannalta, koska ilman sitä olisi ohjelmoinnin testaus jäänyt hitsaussolun käyttöönottovaiheeseen ja mahdollisesti viivästyttänyt koko projektin valmistumista.

Opinnäytetyö laajensi tietämystä ABB-teollisuusrobottien ohjelmoinnista RobotStudioilla ja joustavan ohjelmoinnin mahdollisuuksista nykypäivän teollisuudessa. Opinnäytetyössä käytettiin runsaasti lähteitä, joista kerättyä tietoa voidaan pitää pääosin luotettavana. Lähteet pyrittiin hankkimaan tunnettujen tekijöiden verkkosivuilta ja manuaaleista. Suurin osa lähteistä sisälsi ajankohtaista tietoa, mutta vanhempia julkaisuja käytettäessä lähteitä arvioitiin kriittisemmin.

Joustavan ohjelmoinnin merkitys kasvaa entisestään tulevaisuuden konepajateollisuudessa. Joustavilla ohjelmointiratkaisuilla mahdollistetaan tehokkaampi tuotantoprosessi ja nopeampi mukautumiskyky asiakkaan tarpeeseen. Opinnäytetyöstä saatuja tuloksia voidaan hyödyntää uusien robottisolujen suunnitteluun ja ohjelmointiin sekä vanhojen tuotantolinjojen kehittämiseen Emeca Oy:n tuotannossa.

LÄHTEET

ABB 2010. Technical reference manual. RAPID Instructions, Functions and Data types. Luettu 20.11.2021

https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf

ABB. 2013a. Käyttäjän opas IRC5 ja FlexPendant. ABB Robotics. Luettu 13.11.2021

<https://docplayer.fi/510174-Abb-robotics-kayttajan-opas-irc5-ja-flexpendant.html>

ABB. 2013b. Operating manual. Introduction to RAPID. Luettu 20.11.2021

<http://rovar.cimr.pub.ro/docs/OplIntroRAPID.pdf>

ABB. 2019. IRC5 Industrial Robot Controller. Luettu 20.11.2021

<https://search.abb.com/library/Download.aspx?DocumentID=ROB0295EN&LanguageCode=en&DocumentPartId=&Action=Launch>

ABB. 2021a. Cobots. Luettu 14.11.2021

<https://cobots.robotics.abb.com/en/>

ABB. 2021b. Käyttäjän opas. IRC 5 ja Flex Pendant. Luettu 8.1.2021

<https://abb.sluzba.cz/Pages/Public/IRC5RoboticsDocumentationRW6/Controllers/IRC5/FlexPendant/fi/3HAC050941-013.pdf>

ABB. 2021c. Product specification IRB 2400. Luettu 20.11.2021

<https://search.abb.com/library/Download.aspx?DocumentID=3HAC042195-001&LanguageCode=en&DocumentPartId=&Action=Launch>

ABB. 2021c. Robotics. Luettu 14.11.2021

<https://new.abb.com/products/robotics>

ABB. 2021d. Technical reference manual. RAPID kernel. Luettu 20.11.2021

<https://abb.sluzba.cz/Pages/Public/IRC5RoboticsDocumentationRW6/Controllers/RobotWare/RAPID/en/3HAC050946-001.pdf>

ABB. 2021e. Technical reference manual. RAPID Overview. Luettu 20.11.2021

<https://abb.sluzba.cz/Pages/Public/OmniCoreRoboticsDocumentationRW7/Controllers/RobotWare/RAPID/en/3HAC065040-001.pdf>

ABB. 2022a. Industrial robots. Luettu 08.01.2022

<https://new.abb.com/products/robotics/industrial-robots>

ABB. 2022b. IRB 2400 Luettu 08.01.2022

<https://new.abb.com/products/robotics/industrial-robots/irb-2400>

ABB. 2022c. IRC5 - Industrial Robot Controllers. Luettu 08.01.2022

<https://new.abb.com/products/robotics/controllers/irc5-overview>

ABB. 2022d. Robot Application Software. Luettu 08.01.2022

<https://new.abb.com/products/robotics/application-software>

ABB. n.d. Product Specification. IRB 2400. Luettu 21.11.2021

<https://library.e.abb.com/public/fa8324850004a97fc125766d003e863c/Product%20specification%202400%20M98A%20BWOS3.2.pdf>

Collins, D. 2018. What is a Cartesian robot? Linear Motion Tips. Julkaistu 05.10.2016. Luettu 14.11.2021

<https://www.linearmotiontips.com/what-is-a-cartesian-robot/>

Delfoi. n.d. Offline programming. Luettu 28.11.2021

<https://www.delfoi.com/delfoi-robotics/offline-programming/>

Emeca. 2020. Jatkoksen käyttöohje. Luettu 01.11.2021

<https://emeca.fi/wp-content/uploads/2020/07/200609-Jatkoksen-kayttoohje.pdf>

Emeca. 2021a. Paalujatkokset. Luettu 01.11.2021.

<https://emeca.fi/paalujatkokset/>

Emeca. 2021b. Paalukärjet. Luettu 01.11.2021.

<https://emeca.fi/paalukarjet/>

Emeca. 2021c. Tuotanto. Luettu 01.11.2021.

<https://emeca.fi/tuotanto/>

Emeca. 2021d. Yritys. Luettu 01.11.2021.

<https://emeca.fi/yritys/>

Eu Robotics. n.d. From 'SMERobot' to 'SMERobotics': Robots for SME manufacturing. Julkaistu 31.05.2014. Luettu 28.11.2021

<https://www.eu-robotics.net/sparc/success-stories/from-smerobot-to-smerobotics-robots-for-sme-manufacturing.html?changelang=2>

FlexiBowl. 2021. SCARA Robot. Julkaistu 11.05.2021. Luettu 14.11.2021

<https://www.flexibowl.com/scara-robot.html>

Gonzalez, C. 2016. What's the Difference Between Industrial Robots? Machine Design. Julkaistu 01.12.2016. Luettu 14.11.2021

<https://www.machinedesign.com/markets/robotics/article/21835000/whats-the-difference-between-industrial-robots>

Hanson, K. 2021. Welding Productivity. Taking it offline. Luettu 28.11.2021

<https://weldingproductivity.com/article/taking-it-offline/>

How to Mechatronics. 2021. How Servo Motor Works & How to Control Servos using Arduino. Luettu 13.11.2021

<https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/>

IFR. 2021a. IFR presents World Robotics 2021 reports. Luettu 13.11.2021

<https://ifr.org/ifr-press-releases/news/robot-sales-rise-again>

IFR. 2021b. Industrial Robots. Luettu 13.11.2021

<https://ifr.org/industrial-robots>

Kauppalehti. 2021. Emeca Oy. Luettu 01.11.2021

<https://www.kauppalehti.fi/yritykset/yritys/emeca+oy/16481355>

Kemppi. 2021. Robottihitsaus. Luettu 13.11.2021

<https://www.kemppi.com/fi-FI/tuki/hitsausaapinen/robottihitsaus/>

Kishan, J. 2016. Standard safety practices for Industrial Robots. Research Gate. Julkaistu 01.7.2016. Luettu 08.01.2021

https://www.researchgate.net/publication/310799255_Standard_safety_practices_for_Industrial_Robots

Lehtinen, N. 2019. Suomen teollisuusrobotit lukuina ja Suomen aseet robotiikan vallankumouksessa. Suomen Robotiikkayhdistys. Julkaistu 10.04.2019. Luettu 13.11.2021

<https://www.merinoiva.fi/wp-content/uploads/2019/04/Nina-Vaasa-10.4.2019.pdf>

Miller, J. 2017. The drivers of flexible automation. Control Engineering. Julkaistu 08.03.2017. Luettu 08.01.2022

<https://www.controleng.com/articles/the-drivers-of-flexible-automation/>

Robotics Tomorrow. 2021. Why Use Offline Robot Programming Software and How to Get Started. Julkaistu 09.09.2021. Luettu 28.11.2021

<https://www.roboticstomorrow.com/article/2021/09/why-use-offline-robot-programming-software-and-how-to-get-started/17418>

Robotiq. n.d. What Are the Different Programming Methods for Robots? Julkaistu 24.03.2016. Luettu 28.11.2021

<https://blog.robotiq.com/what-are-the-different-programming-methods-for-robots>

Robotpark. 2015. Flexible automation Julkaistu 19.3.2013. Luettu 28.11.2021

<http://www.robotpark.com/academy/flexible-automation-21052/>

Robots Done Right. 2021a. ABB Robots. Luettu 14.11.2021

<https://robotsdoneright.com/ABB-Robots.html>

Robots Done Right. 2021b. Online Robot Programming vs Offline Robot Programming. Luettu 28.11.2021

<https://robotsdoneright.com/Articles/online-robot-programming-vs-offline-robot-programming.html>

Robots Done Right. 2021c. Teach Pendants vs Lead Through Programming. Luettu 28.11.2021

<https://robotsdoneright.com/Articles/teach-pendants-vs-lead-through-programming.html>

Robots Done Right. 2021d. What is a Delta Robot? Luettu 14.11.2021

<https://robotsdoneright.com/Articles/what-is-a-delta-robot.html>

Robots Done Right. 2021e. What is an Articulated Robot? Luettu 13.11.2021
<https://robotsdoneright.com/Articles/what-is-an-articulated-robot.html>

Robots Done Right. 2022. Reducing Cycle Times with Industrial Robots. Luettu 19.2.2022
<https://robotsdoneright.com/Articles/reducing-cycle-times-with-industrial-robots.html>

Robots. 2021a. Applications. Collaborative Robots. Luettu 13.11.2021
<https://www.robots.com/applications/collaborative>

Robots. 2021b. What are the main types of robots. Luettu 13.11.2021
<https://www.robots.com/faq/what-are-the-main-types-of-robots>

Robots. 2021c. Why should my company use industrial robots? Luettu 13.11.2021
<https://www.robots.com/faq/why-should-my-company-use-industrial-robots>

Robots. 2022. Safety Options for Robotic Systems. Luettu 08.01.2022
<https://www.robots.com/articles/safety-options-for-robotic-systems>

SFS-EN ISO 10218-1. 2011. Robotit ja robotiikkalaitteet. Turvallisuusvaatimukset. Osa 1: Teollisuusrobotit. Helsinki: Suomen Standardoimisliitto SFS. Luettu 08.01.2022. Vaatii käyttöoikeuden
<https://online.sfs.fi/fi/index/tuotteet/SFS/CENISO/ID2/1/235190.html.stx>

SFS-EN ISO 10218-2. 2011. Robotit ja robotiikkalaitteet. Turvallisuusvaatimukset. Osa 2: Robottijärjestelmät ja niiden yhdistelmät. Helsinki: Suomen Standardoimisliitto SFS. Luettu 08.01.2022. Vaatii käyttöoikeuden
<https://online.sfs.fi/fi/index/tuotteet/SFS/CENISO/ID2/1/540732.html.stx>

SFS ISO 8373. 2021. Robotics - Vocabulary. Helsinki: Suomen Standardoimisliitto SFS. Luettu 13.12.2021. Vaatii käyttöoikeuden
<https://online.sfs.fi/fi/index/tuotteet/ISO/ISO/ID9998/8/1043935.html.stx>

SFS 3054. 2020. Hitsaussanasto. Kaarihitsaus. Helsinki: Suomen Standardoimisliitto SFS. Luettu 13.12.2021. Vaatii käyttöoikeuden
<https://online.sfs.fi/fi/index/tuotteet/SFS/SFS/ID2/3/943799.html.stx>

Wilkins, J. 2021. Comment: understanding the methods of robot programming. manufacturingglobal. Julkaistu 16.05.2020. Luettu 28.11.2021
<https://manufacturingglobal.com/technology/comment-understanding-methods-robot-programming>

Wilson, M. 2014. Implementation of Robot Systems. Butterworth-Heinemann Inc. Luettu 08.01.2022

LIITTEET

Liite 1. Kappaleenkäsittelyrobotin ohjelmat

1 (5)

```

1
2  MODULE EMECA
3  TASK PERS wobjdata WO_Pohjalevylava:=[FALSE,TRUE,"",[
4  TASK PERS wobjdata WO_Valmistuote_lava:=[FALSE,TRUE,'
5  TASK PERS wobjdata WO_Laatikkojigi:=[FALSE,TRUE,"",[[
6  TASK PERS wobjdata WO_Kaulusjigi:=[FALSE,TRUE,"",[[6
7  TASK PERS wobjdata WO_Paikotusteline:=[FALSE,TRUE,"",
8  CONST robtarget P_etsimispiste:=[[1261.96,-236.52,-:
9  CONST robtarget P_Pinon_Pohja10:=[[-57.29,66.54,589.1
10 CONST robtarget P_koti:=[[426.91,22.24,795.11],[0.01:
11 PERS robtarget P_Pinon_Keskipiste:=[[1261.33,-228.02
12 PERS robtarget P_move_lavaus:=[[505,140,-5.27],[0.99
13 PERS robtarget P_hakupiste:=[[1087.02,-402.337,-291.
14 PERS robtarget P_Pinon_Pohja:=[[1261.33,-228.031,15.:
15 PERS robtarget P_lavaus_laatikko:=[[140,140,-5.27],[
16 PERS robtarget P_Kauluksen_haku:=[[1474.63,175.89,-71
17 PERS robtarget P_Kauluksen_vienti:=[[1159.29,697.95,-
18 PERS robtarget P_Paikotus_piste:=[[757.03,77.60,-442.
19 PERS num Laatikon_koko:=350;
20 PERS num Laatikon_korkeus:=100;
21 PERS num Riveja_lavalla:=3;
22 PERS num Sarakkeita_lavalla:=2;
23 PERS num Laatikko_halutut_rivit:=3;
24 PERS num Laatikko_halutut_sarakkeet:=2;
25 PERS num Laatikko_halutut_kerrokset:=7;
26 PERS num Laatikko_koko_minus_235:=115;
27 PERS num Pinoja_Lavalla:=6;
28 PERS num R_Aihio_X_Offset:=380;
29 PERS num R_Aihio_Y_Offset:=380;
30 PERS num Pinon_numero:=0;
31 PERS num Pinon_haku_offs_X:=0;
32 PERS num Pinon_haku_offs_Y:=0;
33 PERS num Sarake_nro:=0;
34 PERS num Rivi_nro:=0;
35 PERS num Lavaus_X_offs:=365;
36 PERS num Lavaus_Y_offs:=0;
37 PERS num Lavaus_Z_offs:=0;
38 PERS num Lavaus_rivi:=1;
39 PERS num Lavaus_sarake:=0;
40 PERS num Lavaus_kerros:=0;
41 PERS num Lavaus_pino_laskuri:=3;
42 PERS num Laatikko_x_vali:=365;
43 PERS num Laatikko_y_vali:=365;
44 PERS num Laatikko_z_vali:=105;
45 PERS num Lavaus_pinot_yht:=6;
46 PERS num Lavaus_kpl_yhteensa:=42;
47 PERS num Lavaus_kpl_laskuri:=3;
48 PERS num lavauskerros_pariton:=0;

```

```

49 PERS num Aihio_tunnistus_x:=175;
50 PERS num Aihio_tunnistus_y:=175;
51 PERS bool Hitsaus_paalla:= FALSE;
52 PERS bool Lupa_hitsata_laatikko:= FALSE;
53 PERS bool Lupa_hitsata_kaulus:= FALSE;
54 PERS bool Seuraava_ahiopino:=FALSE;
55 PERS bool Paikotus_ok:=TRUE;
56 PERS bool Aihio_jigiin_ok:=TRUE;
57 PERS bool Aihio_paikotukseen_ok:=FALSE;
58 PERS bool Ltk_jigi_tyhja:=FALSE;
59 PERS bool Kauluksen_haku_ok:=FALSE;
60 PERS bool Laatikko_valmis:=FALSE;
61 PERS bool Aihion_nouto_ok:=TRUE;
62 PERS bool Kaulus_jigiin_ok := FALSE;
63 PERS bool Kauluksen_hitsaus_tyhja := FALSE;
64 PERS bool Uusi_Lava := FALSE;
65 PERS bool Kauluksen_hitsaus_päällä:=FALSE;
66
67
68 □ PROC main()
69     EMECA_laatikko_muuttujat;
70     Aihioopinon_sijainnin_laskenta;
71     Kaulus_hitsaukseen;
72     Aihioopinon_tunnistus;
73     Aihion_nouto_lavalta;
74     Aihio_paikotukseen;
75     Laatikon_lavaus;
76     Aihio_jigiin;
77     Kaulus_kokoonpanojigiin;
78 ENDPROC
79
80 □ PROC EMECA_laatikko_muuttujat()
81     Set do_search;
82     Laatikko_x_vali := Laatikon_koko + 15;
83     Laatikko_y_vali := Laatikon_koko + 15;
84     Laatikko_z_vali := Laatikon_korkeus + 5;
85     R_Aihio_X_Offset := Laatikon_koko + 30;
86     R_Aihio_Y_Offset := Laatikon_koko + 30;
87     Aihio_tunnistus_x:= Laatikon_koko / 2;
88     Aihio_tunnistus_y:= Laatikon_koko / 2;
89     Laatikko_koko_miinus_235 := Laatikon_koko - 235;
90     !Lavaus_sijainnin_laskenta
91     Lavaus_rivi := Lavaus_pino_laskuri DIV Laatikko_halutut_sarakkeet;
92     Lavaus_sarake := Lavaus_pino_laskuri MOD Laatikko_halutut_sarakkeet;
93     Lavaus_kerros := Lavaus_kpl_laskuri DIV Lavaus_pinot_yht;
94     !Offsetin_laskenta
95     Lavaus_X_offs := Laatikko_x_vali * Lavaus_rivi;
96     Lavaus_Y_offs := Laatikko_y_vali * Lavaus_sarake;
97     Lavaus_Z_offs := Laatikko_z_vali * Lavaus_kerros;
98 ENDPROC
99
100 □ PROC Aihioopinon_sijainnin_laskenta()
101     Pinoja_Lavalla := Riveja_lavalla * Sarakkeita_lavalla;
102     Sarake_nro := Pinon_numero MOD Sarakkeita_lavalla;
103     Rivi_nro := Pinon_numero DIV (Riveja_lavalla - 1);
104     Pinon_haku_offs_X := R_Aihio_X_Offset * Sarake_nro;
105     Pinon_haku_offs_Y := R_Aihio_Y_Offset * Rivi_nro;
106 ENDPROC
107
108 □ PROC Kaulus_hitsaukseen()
109 □ IF diKaulusTaitettu = 1 AND Kauluksen_hitsaus_tyhja = TRUE AND Kauluksen_hitsaus_päällä = FALSE THEN
110     MoveJ [[925.54,-50.14,-902.71],[0.999629,-0.00147957,0.00391064,0.0269054],[0,0,-3,0],[9E+09,9E+09,9E+
111     MoveJ [[1470.16,172.66,-907.29],[0.999629,-0.00147954,0.00391077,0.0269052],[0,0,-3,0],[9E+09,9E+09,9E+
112     MoveL RelTool(P_Kauluksen_haku,0,0,(-Laatikko_koko_miinus_235)), v1500, fine, gripdon\WObj:=W0_Kaulusji
113     WaitRob\InPos;
114     WaitDI diLaseranturi, 1;
115 □ IF diLaseranturi = 1 THEN
116     Reset doMagneettitarttuja;
117     WaitTime 0.5;
118     PulseDO\PLength:=1, doTarttujakiinni;
119     WaitDI diSaalähteä, 1;
120     MoveL [[1470.75,173.31,-937.21],[0.999583,-0.0055867,0.00889301,0.0269085],[0,0,-3,0],[9E+09,9E+09
121     MoveJ [[1172.99,413.47,-868.22],[0.507472,-0.490197,-0.514963,-0.486818],[0,-2,-3,0],[9E+09,9E+09,
122     MoveL RelTool(P_Kauluksen_vienti,0,0,(-Laatikko_koko_miinus_235)), v1000, fine, gripdon\WObj:=W0_Ka
123     WaitRob\InPos;

```

```

124 PulseDO\PLength:=1, doLupataittaa;
125 WaitDI diLaseranturi, 1;
126 IF diLaseranturi = 1 THEN
127     Set doMagneettitarttuja;
128     WaitTime 0.4;
129     MoveL [[1165.96,554.31,-706.09],[0.507473,-0.490197,-0.514962,-0.486817],[0,-2,-3,0],[9E+09,9E+09,9E+09,9E+09],0,0,0,0];
130     MoveJ [[1050.33,324.70,-913.95],[0.684486,-0.219919,-0.232797,-0.65492],[0,-1,-4,0],[9E+09,9E+09,9E+09,9E+09],0,0,0,0];
131     Set do_painajat_ltk;
132     WaitTime 0.1;
133     Set do_kauluksen_puristajat;
134     PulseDO\PLength:=0.1, do_painajat_ltk;
135     MoveJ P_koti, v5000, z200, gripdon\WObj:=wobj0;
136     Aihion_nouto_ok := TRUE;
137     Kauluksen_hitsaus_tyhja := FALSE;
138     Lupa_hitsata_kaulus := TRUE;
139     ENDIF
140 ENDIF
141 ENDIF
142 ENDPROC
143
144 PROC Aihionpinon_tunnistus()
145 IF Seuraava_ahiopino = TRUE OR Uusi_Lava = TRUE THEN
146     MoveJ [[1261.96,-236.52,-522.79],[0.685965,0.00609614,0.00204865,0.727606],[0,0,-2,0],[9E+09,9E+09,9E+09,9E+09],0,0,0,0];
147     MoveJ Offs(P_etsimispiste1,Pinon_haku_offs_X,-Pinon_haku_offs_Y,0), v1000, fine, gripdon\WObj:=wobj0;
148     Set do_search;
149     SearchL\Stop, diLaseranturi, P_hakupiste1, Offs(P_etsimispiste1,Pinon_haku_offs_X,Pinon_haku_offs_Y,0), v100, fine, gripdon\WObj:=wobj0;
150     MoveL Offs(P_hakupiste1,0,0,50), v100, fine, gripdon\WObj:=wobj0;
151     SearchL\Stop, diLaseranturi\NegFlank, P_hakupiste1, Offs(P_hakupiste1,-500,0,50), v100, fine, gripdon\WObj:=wobj0;
152     MoveL Offs(P_hakupiste1,20,0,0), v100, fine, gripdon\WObj:=wobj0;
153     SearchL\Stop, diLaseranturi\NegFlank, P_hakupiste1, Offs(P_hakupiste1,20,-500,0), v100, fine, gripdon\WObj:=wobj0;
154     MoveL Offs(P_hakupiste1,Aihio_tunnistus_x,Aihio_tunnistus_y,0), v100, fine, gripdon\WObj:=wobj0;
155     P_Pinon_Keskipiste := CROBT(\Tool:=gripdon\WObj:=wobj0);
156     P_Pinon_Pohja := CROBT(\Tool:=gripdon\WObj:=wobj0);
157     P_Pinon_Pohja.trans.z := 15.3;
158     Reset do_search;
159     Aihion_nouto_ok := TRUE;
160     Seuraava_ahiopino := FALSE;
161     Uusi_Lava := FALSE;
162     ENDIF
163 ERROR
164 IF ERRNO = ERR_WHLSEARCH THEN
165     Incr Pinon_numero;
166     Set doMagneettitarttuja;
167     IF Pinon_numero >= Pinoja_Lavalla THEN
168         Pinon_numero := 0;
169     ENDIF
170     Seuraava_ahiopino := TRUE;
171     RETURN;
172 ENDIF
173 ENDPROC
174
175 PROC Aihion_nouto_lavalta()
176 aihion_haku_alusta:
177 IF Aihion_nouto_ok = TRUE AND Aihio_paikotukseen_ok = FALSE THEN
178     MoveJ Offs(P_Pinon_Keskipiste,0,0,-500), v1000, z50, gripdon\WObj:=wobj0;
179     MoveL Offs(P_Pinon_Keskipiste,0,0,-10), v200, z50, gripdon\WObj:=wobj0;
180     Set do_search;
181     Confl\Off;
182     SearchL\Stop, diTarttutaraja, P_Pinon_Keskipiste, P_Pinon_Pohja, v60, gripdon\WObj:=wobj0;
183     WaitTime 0.2;
184     IF diTarttutaraja = 1 THEN
185         Reset doMagneettitarttuja;
186     ENDIF
187     WaitTime 0.2;
188     MoveL RelTool(P_Pinon_Keskipiste,0,0,-50), v100, fine, gripdon\WObj:=wobj0;
189     MoveJ RelTool(P_Pinon_Keskipiste,0,0,-200), v1000, z50, gripdon\WObj:=wobj0;
190     IF diTarttutaraja = 0 THEN
191         GOTO aihion_haku_alusta;
192     ENDIF
193     Aihio_paikotukseen_ok := TRUE;
194     MoveJ [[1284.36,-239.52,-476.66],[0.68591,0.00138396,-0.010994,0.727602],[0,0,-2,0],[9E+09,9E+09,9E+09,9E+09],0,0,0,0];
195     MoveJ P_koti, v5000, z200, gripdon\WObj:=wobj0;
196     Aihion_nouto_ok := FALSE;
197     ENDIF
198 ERROR

```

```

199 IF ERRNO = ERR_WHLSEARCH THEN
200   Incr Pinon_numero;
201   Set doMagneettitarttuja;
202   IF Pinon_numero >= Pinoja_Lavalla THEN
203     Pinon_numero := 0;
204   ENDIF
205   AihioPinon_sijainnin_laskenta;
206   Seuraava_ahiofino := TRUE;
207   AihioPinon_tunnistus;
208   TRYNEXT;
209 ENDIF
210 ENDPROC
211
212 PROC Aihio_paikotukseen()
213 IF Aihio_paikotukseen_ok = TRUE AND Seuraava_ahiofino = FALSE AND Aihion_nouto_ok = FALSE THEN
214   MoveJ [[830.91,3.71,-507.80],[0.86794,-0.451574,0.0435542,0.202149],[0,0,-4,0],[9E+09,9E+09,9E+09,
215   MoveL RelTool(P_Paikotus_piste,Laatiko_koko_minus_235,0,-50), v100, fine, gripdon\WObj:=WO_Paikot
216   MoveL RelTool(P_Paikotus_piste,Laatiko_koko_minus_235,0,0), v1000, z50, gripdon\WObj:=WO_Paikotu
217   WaitTime 0.1;
218   Set doMagneettitarttuja;
219   WaitTime 0.1;
220   MoveL [[751.50,53.67,-472.11],[0.849849,-0.447592,0.112485,0.254488],[0,0,-3,0],[9E+09,9E+09,9E+09,
221   Paikotus_ok := TRUE;
222   Aihio_paikotukseen_ok := FALSE;
223   MoveJ P_koti, v5000, z200, gripdon\WObj:=wobj0;
224   ENDIF
225 ENDPROC
226
227 PROC Laatikon_lavaus()
228 IF Laatikko_valmis = TRUE AND Hitsaus_paalla = FALSE THEN
229   Reset do_kauluksen_puristajat;
230   Reset do_painajat_ltk;
231   MoveJ [[435.23,448.85,-931.46],[0.914784,-0.0123184,-0.00304699,0.403744],[-1,-1,-3,0],[9E+09,
232   MoveJ [[430.43,455.06,-622.13],[0.914783,-0.0123184,-0.00304696,0.403745],[-1,-1,-3,0],[9E+09,
233   MoveL [[429.33,456.49,-551.16],[0.914783,-0.0123185,-0.00304692,0.403745],[-1,-1,-3,0],[9E+09,
234   WaitDI diTarttutaraja, 1;
235   WaitRob\InPos;
236   WaitTime 0.1;
237   IF diTarttutaraja = 1 THEN
238     Reset doMagneettitarttuja;
239     WaitTime 0.5;
240   ENDIF
241
242   ENDIF
243   MoveL [[434.25,450.13,-867.78],[0.914783,-0.0123185,-0.00304671,0.403745],[-1,-1,-3,0],[9
244   MoveL [[450.94,157.90,-899.93],[0.914783,-0.0123185,-0.00304672,0.403745],[-1,-1,-3,0],[9
245   MoveJ P_koti, v4000, z200, gripdon\WObj:=wobj0;
246   Lavaus_pilot_yht := (Laatikko_halutut_sarakkeet * Laatikko_halutut_rivit);
247   Lavaus_kpl_yhteensa := (Laatikko_halutut_sarakkeet * Laatikko_halutut_rivit * Laatikko_ha
248   !Laskurien_nollaus lavan tullessa täyteen
249   IF Lavaus_pino_laskuri >= Lavaus_pilot_yht THEN
250     Clear Lavaus_pino_laskuri;
251   ENDIF
252   IF Lavaus_kpl_laskuri >= Lavaus_kpl_yhteensa THEN
253     Clear Lavaus_kpl_laskuri;
254     PulseDO\PLength:=1, doLavanvaihto;
255     WaitDI diLavausAloit, 1;
256     EMECA_laatikko_muuttajat;
257   ENDIF
258   !lavaus alkaa
259   P_lavaus_laatikko.trans.x := Laatikon_koko / 2.5;
260   P_lavaus_laatikko.trans.y := Laatikon_koko / 2.5;
261   P_move_lavaus := Offs(P_lavaus_laatikko,Lavaus_X_offs,Lavaus_Y_offs,Lavaus_Z_offs);
262   lavauskerros_pariton := Lavaus_kerros MOD 2;
263   IF lavauskerros_pariton = 1 THEN
264     P_move_lavaus := RelTool(P_move_lavaus,0,0,0\Rz:=15);
265   ELSE
266     P_move_lavaus := RelTool(P_move_lavaus,0,0,0\Rz:=0);
267   ENDIF
268   MoveJ [[616.51,-120.02,-655.81],[0.999573,-0.0238357,0.0113868,0.0124663],[-1,0,-3,0],[9E
269   MoveJ Offs(P_move_lavaus,0,0,-200), v2500, z200, gripdon\WObj:=WO_Valmistuote_lava;
270   MoveL Offs(P_move_lavaus,0,0,-50), v300, z50, gripdon\WObj:=WO_Valmistuote_lava;
271   MoveL P_move_lavaus, v100, z50, gripdon\WObj:=WO_Valmistuote_lava;
272   WaitRob\InPos;
273   WaitTime 0.2;
274   Set doMagneettitarttuja;
275   WaitTime 1;
276   MoveL Offs(P_move_lavaus,0,0,-50), v300, z50, gripdon\WObj:=WO_Valmistuote_lava;
277   MoveL Offs(P_move_lavaus,0,0,-200), v2500, z200, gripdon\WObj:=WO_Valmistuote_lava;
278   MoveJ [[616.51,-120.02,-655.81],[0.999573,-0.0238357,0.0113868,0.0124663],[-1,0,-3,0],[9E
279   MoveJ P_koti, v5000, z200, gripdon\WObj:=wobj0;
280   Incr Lavaus_pino_laskuri;
281   Incr Lavaus_kpl_laskuri;
282   P_move_lavaus := RelTool(P_move_lavaus,0,0,0\Rz:=0);
283   Laatikko_valmis := FALSE;
284   Aihion_nouto_ok := TRUE;
285   Aihio_jigiin_ok := TRUE;
286   WaitDI diTyokierroLopetus, 0;
287 ENDIF
288 ENDPROC

```

```

289 PROC Aihio_jigiin()
290     IF Paikotus_ok = TRUE AND Aihio_jigiin_ok = TRUE AND Hitsaus_paalla = FALSE THEN
291         MoveJ [[800.72,12.51,-410.31],[0.849849,-0.447592,0.112485,0.254488],[0,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
292         MoveL [[799.79,33.15,-395.83],[0.849849,-0.447592,0.112485,0.254488],[0,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
293         WaitRob\InPos;
294         WaitDI diTarttutaraja, 0;
295         MoveL [[799.51,39.43,-391.42],[0.849849,-0.447592,0.112485,0.254488],[0,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
296         WaitRob\InPos;
297         WaitDI diTarttutaraja, 1;
298         IF diTarttutaraja = 1 THEN
299             Reset doMagneettitarttuja;
300             WaitTime 0.5;
301             MoveL [[823.64,37.33,-426.55],[0.849849,-0.447592,0.112485,0.254488],[0,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
302             MoveL [[796.49,10.45,-502.60],[0.849849,-0.447592,0.112485,0.254488],[0,0,-4,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
303         ENDIF
304         MoveJ P_koti, v2500, z200, gripdon\WObj:=wobj0;
305         MoveJ [[448.10,417.48,-896.68],[0.915191,-0.0159408,0.00507081,0.402672],[-1,-1,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
306         MoveL [[423.02,461.05,-566.02],[0.915192,-0.0159414,0.00507099,0.402671],[-1,-1,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
307         WaitRob\InPos;
308         IF diTarttutaraja = 1 THEN
309             Set doMagneettitarttuja;
310             WaitTime 0.5;
311         ELSE
312             WaitDI diTarttutaraja, 1;
313         ENDIF
314         MoveL [[424.07,451.18,-862.79],[0.915191,-0.0159413,0.00507095,0.402672],[-1,-1,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
315         MoveJ P_koti, vmax, z200, gripdon\WObj:=wobj0;
316         Kauluksen_haku_ok := TRUE;
317         Paikotus_ok := FALSE;
318         Aihio_jigiin_ok := FALSE;
319     ENDIF
320 ENDPROC
321
322 PROC Kaulus_kokoonpanojigiin()
323     IF Kauluksen_haku_ok = TRUE AND Kauluksen_hitsaus_paalla = FALSE THEN
324         MoveJ [[995.72,114.24,-808.08],[0.507473,-0.490197,-0.514962,-0.486818],[0,-2,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
325         MoveJ [[1089.73,504.07,-826.09],[0.507473,-0.490197,-0.514962,-0.486818],[0,-2,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
326         MoveL RelTool(P_Kauluksen_vienti,0,0,(-Laatiko_koko_minus_235)), v100, fine, gripdon\WObj:=WO_Kaulusjigi;
327         WaitRob\InPos;
328         WaitDI diLaseranturi, 1;
329         IF diLaseranturi = 1 THEN
330             Reset doMagneettitarttuja;
331             WaitTime 0.5;
332             MoveL [[1166.10,580.90,-968.63],[0.509346,-0.482032,-0.514046,-0.493932],[0,-2,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
333             MoveJ [[495.47,386.86,-793.58],[0.680212,-0.606993,-0.248618,0.327198],[0,-2,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
334             WaitRob\InPos;
335             WaitDI diLaseranturi, 1;
336             IF diLaseranturi = 1 THEN
337                 Set doMagneettitarttuja;
338                 WaitTime 0.4;
339                 MoveL [[530.84,344.62,-931.45],[0.680212,-0.606993,-0.248619,0.327198],[0,-2,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09]]
340                 Set do_painajat_ltk;
341                 WaitTime 0.1;
342                 Set do_kaulusken_puristajat;
343                 PulseDO\PLength:=0.1, do_painajat_ltk;
344                 Lupa_hitsata_laatikko := TRUE;
345                 MoveJ P_koti, v5000, z200, gripdon\WObj:=wobj0;
346                 Kauluksen_hitsaus_tyhja := TRUE;
347                 Kaulus_jigiin_ok := FALSE;
348                 Kauluksen_haku_ok := FALSE;
349             ENDIF
350         ENDIF
351     ENDIF
352 ENDPROC
353 ENDMODULE

```

Liite 2. Hitsausrobotin ohjelmat

1 (2)

```

1  MODULE EmecaWeld
2  TASK PERS wobjdata WO_Laatikkojigi:=[FALSE,TRUE,"",[[1254.81,28.9943,725],[8.3
3  TASK PERS wobjdata WO_Kaulusjigi:=[FALSE,TRUE,"",[[605.872,-539.969,701.2],[0
4  CONST robtarget P_kotip:=[[421.21,-6.58,1023.16],[0.107769,2.48579E-09,-0.994:
5  PERS robtarget P_1:=[[0.01,1.10,-8.68],[0.652552,0.279637,0.275467,-0.648149].
6  PERS robtarget P_2:=[[7.25,29.02,-11.28],[0.784205,0.361481,0.00295353,-0.504:
7  PERS robtarget P_3:=[[11.26,15.51,-11.11],[0.783911,0.361302,0.00273006,-0.50:
8  PERS robtarget P_4:=[[29.24,7.02,-10.82],[0.784205,0.361482,0.0029533,-0.5043:
9  PERS num reg1:=0;
10 PERS num Laatikon_koko:=350;
11 PERS num Kauluksen_paksuus:=4;
12 PERS num OffsSivu:=342;
13 PERS num Offs5:=150;
14 PERS num Offs6:=200;
15 PERS num Offs7:=320;
16 PERS num Offs8:=336;
17 PERS num Offs9:=30;
18 PERS num Offs10:=14;
19 PERS num Kiertokulma:=0;
20 PERS bool Hitsaus_paalla:= FALSE;
21 PERS bool Lupa_hitsata_laatikko:= FALSE;
22 PERS bool Lupa_hitsata_kaulus:= FALSE;
23 PERS robtarget P_5_45dg:=[[0.01,1.10,-7.78],[0.83896,0.147645,0.344895,-0.394:
24 PERS robtarget P_6_90deg:=[[0.01,1.10,-6.98],[0.926771,0.00777406,0.375428,-0
25 PERS robtarget P_7_135deg:=[[0.01,1.10,-7.73],[0.857964,-0.131678,0.345303,0.:
26 PERS robtarget P_8_180deg:=[[0.01,1.10,-8.13],[0.654395,-0.257947,0.264238,0.
27 PERS robtarget P_9_225deg:=[[0.01,1.10,-8.75],[0.364164,-0.337346,0.150187,0.
28 PERS robtarget P_10_270deg:=[[0.34,1.45,-7.47],[0.0213886,-0.368639,0.0149118:
29 PERS bool Laatikko_valmis:=FALSE;
30 PERS bool Aihio_jigiin_ok:=TRUE;
31 PERS bool Kauluksen_hitsaus_päällä:=FALSE;
32
33 PROC main()
34     Laatikon_hitsaus_muuttujat;
35     IF Lupa_hitsata_laatikko = TRUE THEN
36         Emeca_laatikko;
37     ELSEIF Lupa_hitsata_kaulus = TRUE THEN
38         Emeca_kaulus;
39     ENDIF
40 ENDPROC
41
42 PROC Laatikon_hitsaus_muuttujat()
43     OffsSivu := Laatikon_koko - (Kauluksen_paksuus * 2);
44     Offs5 := (Laatikon_koko / 7) * 3;
45     Offs6 := (Laatikon_koko / 7) * 4;
46     Offs7 := OffsSivu - 22;
47     Offs8 := OffsSivu - 6;
48     Offs9 := 22 + (Kauluksen_paksuus * 2);
49     Offs10 := 6 + (Kauluksen_paksuus * 2);
50 ENDPROC

```

2 (2)

```

52 PROC Emeca_laatikko()
53   CONST robtarg P_5_45deg:=[[0.01,1.10,-0.34],[0.83896,0.147645,0.344895,-0.394201],[0,-1,-1,0],[9E+09
54   Hitsaus_paalla := TRUE;
55   MoveJ [[177.62,210.74,-264.79],[0.784204,0.361481,0.00295365,-0.504327],[-1,0,-2,0],[9E+09,9E+09,9E+09
56   !Laatikon Hitsaus
57   MotionSup\On;
58   ConfL\Off;
59   MoveL RelTool(P_2,0,0,-25), v1000, z50, Weldgun\WObj:=WO_Laatikkojigi;
60   ArcLStart P_2, v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
61   ArcL P_3, v1000, seam1, weld1, z10, Weldgun\WObj:=WO_Laatikkojigi;
62   ArcLEnd P_4, v1000, seam1, weld1, z10, Weldgun\WObj:=WO_Laatikkojigi;
63   MoveL RelTool(P_4,0,0,-50), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
64   MoveL Offs(P_1,Offs5,25,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
65   ArcLStart Offs(P_1,Offs5,0,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
66   ArcLEnd Offs(P_1,Offs6,0,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
67   MoveL Offs(P_1,Offs6,25,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
68   MoveL Offs(P_1,Offs7,25,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
69   ArcLStart Offs(P_1,Offs7,0,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
70   ArcL Offs(P_5_45deg,Offs8,Offs10,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
71   ArcLEnd Offs(P_6_90deg,OffsSivu,30,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
72   MoveL Offs(P_6_90deg,OffsSivu - 25,30,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
73   MoveL Offs(P_6_90deg,OffsSivu - 25,Offs5,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
74   ArcLStart Offs(P_6_90deg,OffsSivu,Offs5,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
75   ArcLEnd Offs(P_6_90deg,OffsSivu,Offs6,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
76   MoveL Offs(P_6_90deg,OffsSivu - 25,Offs6,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
77   MoveL Offs(P_6_90deg,OffsSivu - 25,Offs7,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
78   ArcLStart Offs(P_6_90deg,OffsSivu,Offs7,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
79   ArcL Offs(P_7_135deg,Offs8,Offs8,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
80   ArcLEnd Offs(P_8_180deg,Offs7,OffsSivu,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
81   MoveL Offs(P_8_180deg,Offs7,OffsSivu - 25,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
82   MoveL Offs(P_8_180deg,Offs6,OffsSivu - 25,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
83   ArcLStart Offs(P_8_180deg,Offs6,OffsSivu,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
84   ArcLEnd Offs(P_8_180deg,Offs5,OffsSivu,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
85   MoveL Offs(P_8_180deg,Offs5,OffsSivu - 25,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
86   MoveL Offs(P_8_180deg,Offs9,OffsSivu - 25,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
87   ArcLStart Offs(P_8_180deg,Offs9,OffsSivu,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
88   ArcL Offs(P_9_225deg,Offs10,Offs8,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laatikkojigi;
89   ArcLEnd Offs(P_10_270deg,Kauluksen_paksuus,Offs7,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laati
90   MoveL Offs(P_10_270deg,Kauluksen_paksuus + 25,Offs7,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
91   MoveL Offs(P_10_270deg,Kauluksen_paksuus + 25,Offs6,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
92   ArcLStart Offs(P_10_270deg,Kauluksen_paksuus,Offs6,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laa
93   ArcLEnd Offs(P_10_270deg,Kauluksen_paksuus,Offs5,0), v1000, seam1, weld1, fine, Weldgun\WObj:=WO_Laati
94   MoveL Offs(P_10_270deg,Kauluksen_paksuus + 25,Offs5,-25), vmax, fine, Weldgun\WObj:=WO_Laatikkojigi;
95   MoveJ [[101.20,146.89,-108.87],[0.215395,-0.363595,-0.0625733,0.904151],[0,-1,1,0],[9E+09,9E+09,9E+09,
96   MoveJ P_kotip, vmax, fine, Weldgun\WObj:=wobj0;
97   WaitRob\InPos;
98   Lupa_hitsata_laatikko := FALSE;
99   Laatikko_valmis := TRUE;
100  Hitsaus_paalla := FALSE;
101  ENDPROC
102
103 PROC Emeca_kaulus()
104 IF Lupa_hitsata_kaulus = TRUE THEN
105   Kauluksen_hitsaus_päällä := TRUE;
106   !Kauluksen hitsaus
107   MoveJ [[225.44,7.07,-244.87],[0.464937,-0.0691736,-0.23045,-0.852022],[-1,-1,0,0],[9E+09,9E+09,9E+09,9
108   MotionSup\On;
109   MoveL [[195.19,241.84,-8.10],[0.437734,-0.15957,-0.278681,-0.8398],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+
110   ArcLStart [[197.24,244.73,3.96],[0.437734,-0.15957,-0.278681,-0.8398],[-1,-2,1,0],[9E+09,9E+09,9E+09,9
111   ArcLEnd [[197.24,145.22,13.87],[0.464937,-0.0691738,-0.230449,-0.852022],[-1,-1,0,0],[9E+09,9E+09,9E+
112   MoveL [[201.58,124.68,-25.87],[0.464937,-0.0691737,-0.23045,-0.852022],[-1,-1,0,0],[9E+09,9E+09,9E+09,
113   MoveL [[205.92,76.23,-111.80],[0.466319,-0.0602103,-0.232308,-0.851442],[-1,-1,0,0],[9E+09,9E+09,9E+09,
114   Kauluksen_hitsaus_päällä := FALSE;
115   MoveJ P_kotip, vmax, fine, Weldgun\WObj:=wobj0;
116   Lupa_hitsata_kaulus := FALSE;
117   ENDIF
118   ENDPROC
119 ENDMODULE

```

Liite 3. Hitsaussolun työkierron simulointi

Youtube video: <https://youtu.be/aBIMgD2MwOQ>

Liite 4. Joustavan hitsausohjelman simulointi

Youtube video: https://youtu.be/4tui_T3wl2g