

Tommi Tuunanen

# Teollisuusrobotin käyttöönotto ja ohjelmointi

Opinnäytetyö  
Sähkötekniikka


Toukokuu 2014




MAMK

University of Applied Sciences

## KUVAILULEHTI

	<b>Opinnäytetyön päivämäärä</b> 7.5.2014	
<b>Tekijä(t)</b> Tommi Tuunanen	<b>Koulutusohjelma ja suuntautuminen</b> Sähkötekniikka	
<b>Nimeke</b> Teollisuusrobotin käyttöönotto ja ohjelmointi		
<b>Tiivistelmä</b> <p>Teollisuusrobotit ovat nykyään yleisiä työvälineitä teollisuuden sovelluksissa, ja niitä löytyy kaikilta teollisuuden eri aloilta. Niiden avulla saadaan parannettua tuotteiden laatua, tehostettua tuotantoa ja tuotannosta saadaan joustavampaa, koska yksi robotti voi suorittaa useita sille määriteltyjä tehtäviä, jopa samanaikaisesti. Teollisuusrobottien käytöllä voidaan myös parantaa työturvallisuutta, koska niillä voidaan suorittaa tehtäviä, jotka ovat ihmiselle vaikeita tai vaarallisia. Teollisuusrobotteja käytetään eniten metalli- ja elintarviketeollisuudessa. Maailmalla on käytössä tällä hetkellä yli miljoona teollisuusrobotia ja vuosittain käyttöön tulee uusia robotteja noin 100 000. Uuden teollisuusrobotin hinta lähtee 20 000 eurosta ylöspäin.</p> <p>Tämän opinnäytetyön tavoitteena oli luoda uusi oppimisympäristö Mikkelin ammattikorkeakoululle. Luotu oppimisympäristö on robottisolun. Robottisolun on kokonaisuus, johon kuuluu robotti, sen ohjaukselaitteet ja muut lisä- sekä suojalaitteet. Robottisolun kuuluva robotti on ABB:n pienin teollisuusrobotimalli IRB120. Robottisolun rakentamisen lisäksi tutkin robotin käyttöä ja ohjelmointia. Tutkin robotin ohjelmointia pendantilla ja Robotstudio -simulointiohjelmalla. Robotin käytöstä ja ohjelmoinnista tein ohjeet, joilla pääsee helpommin käsiksi robotin käyttöön ja ohjelmointiin kuin lukemalla ohjekirjoja kanteesta kanteen. Robottisolua voidaan käyttää sähkötekniikan automaatio -opinnoissa. Oppilaat saavat tällöin käytännön kokemusta teollisuusrobotin käytöstä ja ohjelmoinnista.</p>		
<b>Asiasanat (avainsanat)</b> robotti, teollisuusrobotti, robottisolun, manipulaattori, automaatio		
<b>Sivumäärä</b> 93+3 liitettä	<b>Kieli</b> Suomi	<b>URN</b> <a href="http://www.urn.fi/URN:NBN:fi:amk-201405076268">http://www.urn.fi/URN:NBN:fi:amk-201405076268</a>
<b>Huomautus (huomautukset liitteistä)</b>		
<b>Ohjaavan opettajan nimi</b> Teemu Manninen	<b>Opinnäytetyön toimeksiantaja</b> Mikkelin ammattikorkeakoulu	

## DESCRIPTION

		<b>Date of the bachelor's thesis</b>  7.5.2014
<b>Author(s)</b> Tommi Tuunanen	<b>Degree programme and option</b> Electrical engineering	
<b>Name of the bachelor's thesis</b> Implementing and programming industrial robot		
<b>Abstract</b>  Nowadays industrial robots are widely used in industry. Industrial robots are most used in metal- and food industry. They carry out tasks with extreme precision and speed. This makes it possible to create better quality products with increased production rate. Industrial robots also can improve occupational safety because they can be used in hazardous tasks. There are over one million industrial robots in use all over the world and the number is increased by 100 000 every year.  The aim of this thesis was to create new studying environment for electrical engineering students. Built studying environment is a robot cell. A robot cell includes everything a robot system has in it. It includes a robot, a control system for a robot and other equipment such as safety devices. Industrial robot in this robot cell is ABB's smallest industrial robot which is IRB120. I built a robot cell and created instructions of industrial robot programming based on my studies during this thesis.		
<b>Subject headings, (keywords)</b> a robot, an industrial robot, a robot cell, manipulator, automation		
<b>Pages</b> 93+3 attachments	<b>Language</b> Finnish	<b>URN</b> <a href="http://www.urn.fi/URN:NBN:fi:amk-201405076268">http://www.urn.fi/URN:NBN:fi:amk-201405076268</a>
<b>Remarks, notes on appendices</b>		
<b>Tutor</b> Teemu Manninen	<b>Bachelor's thesis assigned by</b> Mikkeli University of Applied Sciences	

# SISÄLTÖ

1	JOHDANTO .....	1
2	ABB LYHYESTI .....	2
3	TEOLLISUUSROBOTIT .....	2
3.1	Robottien historiaa .....	2
3.2	Teollisuusrobotit tänä päivänä .....	3
3.3	Teollisuusrobotin määritelmä .....	6
3.4	Teollisuusrobottien käyttökohteita ja tehtäviä .....	6
3.5	Teollisuusrobottien hyödyt ja haitat .....	7
4	TEOLLISUUSROBOTTIEN TURVALLISUUS .....	9
4.1	Vaarat ja niiden aiheuttajat .....	9
4.2	Vaarojen ehkäiseminen .....	10
4.3	Henkilöstön koulutus ja työskentelymenetelmät .....	11
4.4	Turvalaitteet ja suojausmenetelmät .....	12
4.5	Merkkivalot ja äänimerkinantolaitteet .....	12
4.6	Turva-aluemerkinnät ja varoituskyltit .....	14
4.7	Turva-aidat ja turvakatkaisijat .....	16
4.8	Anturit .....	19
4.8.1	Turvamatto .....	20
4.8.2	Tuntoreuna .....	22
4.8.3	Valokenno ja valoverho .....	23
4.8.4	Turvalaserskanneri .....	26
5	TEOLLISUUSROBOTTIEN RAKENTEET .....	28
5.1	Suorakulmaiset robotit .....	29
5.2	Sylinterirobotit ja napakoordinaattirobotit .....	30
5.3	SCARA-robotit .....	31
5.4	Rinnakkaisrakenteiset robotit .....	31
5.5	Nivelrobotit .....	33
6	TEOLLISUUSROBOTIN OHJELMOINTITAVAT .....	34
6.1	Ohjelmointi ohjelmointikielellä .....	34
6.2	Ohjelmointi opettamalla .....	35
6.3	Ohjelmointi simulointiohjelmalla .....	35

7	ROBOTTISOLUN RAKENTAMINEN JA ROBOTIN OHJELMOINTI.....	36
7.1	Robotin hankinta ja kilpailutus.....	36
7.2	Katsaus IRB120 -teollisuusrobottijärjestelmän ominaisuuksiin.....	37
7.3	Robottisolun rakentaminen.....	38
7.4	Robottijärjestelmän käyttöönotto, kontrolleri ja pendantti.....	42
7.5	Järjestelmän sisään ja ulostulot.....	46
7.6	RAPID-ohjelmointikieli ja RAPID-ohjelman rakenne.....	47
7.7	Moduulit ja rutiinit.....	47
7.8	Ohjelmaeditori .....	49
7.8.1	Taskit ja ohjelmat -ikkuna .....	49
7.8.2	Moduulit -ikkuna .....	50
7.8.3	Rutiinit -ikkuna .....	51
7.8.4	Lisää käsky- ja muokkaa -valikko .....	52
7.8.5	Virheenkorjaus -valikko.....	53
7.9	Ohjelmadata, hotedit ja tuotantoikkuna.....	55
7.10	Käsin ajo -valikko.....	56
7.11	Robotin ohjelmointi.....	57
7.12	Työkalun kiellettyjen alueiden ja robotin liikkeiden rajojen määrittäminen.....	63
7.12.1	Työkalun alueenrajoitus.....	63
7.12.2	Robotin liikkeiden rajat.....	66
7.13	Uuden työkalun määrittäminen.....	68
7.14	Yleisiä käskyjä.....	71
7.15	Robotstudio.....	76
7.15.1	Robotin ohjelmointi robotstudiossa .....	78
7.15.2	Luodun ohjelman siirto kontrolleriin robotstudiossa.....	82
7.15.3	Työkohteen luominen .....	83
8	POHDINTA .....	87
	LÄHTEET.....	89

- 1 Kuva valmiista robottisolusta
- 2 Robotin ohjelmoinnin pikaohje

## 1 JOHDANTO

Tämä opinnäytetyö käsittelee teollisuusrobottien historiaa, asemaa nykypäivänä, käyttökohteita, turvallisuuteen liittyviä asioita, rakenteita ja ohjelmointitapoja. Työn suoritusosiossa perehdytään rakennetun robottisolun rakentamiseen ja tarkemmin robotin käyttöön sekä ohjelmointiin. Annan ohjeita, joilla pääsee hyvin perille robotin käytöstä ja ohjelmoinnista ilman aiempaa kokemusta. Ohjeiden avulla opitaan ohjelmointia ja käyttöä pendantin avulla sekä tietokoneohjelmalla.

Teollisuusrobotteja käytetään kaikilla teollisuuden aloilla ja eniten niitä on käytössä metalli- ja elintarviketeollisuudessa. Teollisuusrobotteja käytetään esimerkiksi hitsaamiseen, maalaamiseen, hiontaan ja muihin erilaisiin kappaleiden käsittelytehtäviin. Robottien avulla saavutetaan monia hyötyjä. Hyötyjä ovat tuotannon lisääntyminen ja siitä saadaan joustavampaa, koska yksi robotti voi suorittaa useita eri tehtäviä, jopa samanaikaisesti. Tuotteiden laatukin paranee, koska teollisuusrobotit suorittavat sille annetun tehtävän aina samalla tavalla ja ne kykenevät tarkkuuteen, johon ihminen ei pysty. Robottien käyttö parantaa myös työturvallisuutta, sillä niitä voidaan hyödyntää tehtävissä, jotka ovat ihmiselle vaikeita tai vaarallisia. Esimerkiksi painavien taakkojen nostot ja kemikaalien käsittely voivat aiheuttaa ihmiselle vaaroja sekä terveydellisiä haittoja, mutta käyttämällä robottia suorittamaan nämä tehtävät voidaan välttää mahdollisesti syntyvät haitat.

Teollisuusrobottien käyttö yleistyy koko ajan ja uusia robotteja otetaan vuosittain käyttöön maailmalla noin 100 000. Teollisuusrobottien määrä maailmanlaajuisesti on yli miljoona. Tärkeimpiä ja suurimpia robottien valmistajia ovat muun muassa Fanuc, Kawasaki, Motoman, Toshiba, Denso, ABB ja Kuka.

## **2 ABB LYHYESTI**

ABB on sähkövoima- ja automaatiotuotteita valmistava yritys. Se on perustettu vuonna 1988, kun ruotsalainen Asea ja sveitsiläinen Brown Boverin yhdistyivät. ABB:n pääkonttori sijaitsee Sveitsissä Zürichissä. ABB:llä on yhteensä liiketoimintaa yli sadassa maassa ja työntekijöitä on yhteensä noin 150 000. Suomessa ABB:llä on työntekijöitä noin 5500. /2./

ABB:n valmistamiin tuotteisiin kuuluvat muun muassa sähkömoottorit, taajuusmuuttajat, pehmokäynnistimet, teollisuusrobotit, tuulisähkötuotteet, laivojen sähkötuotteet, sähköverkot, automaatiojärjestelmät ja muut sähkötuotteet. /2./

## **3 TEOLLISUUSROBOTIT**

Tässä luvussa käydään läpi, miten teollisuusrobotti määritellään, robottien historiaa, robottien tämänpäiväistä tilannetta maailmalla ja Suomessa, teollisuusrobottien käyttökohteita ja tehtäviä sekä robottien tuomia hyötyjä ja haittoja.

### **3.1 Robottien historiaa**

Sana robotti tulee tšekinkielisestä verbistä robotá. Tällä sanalla tarkoitetaan orjuutta ja työntekoa. Robotti -sanana on keksinyt Josef Čapek vuonna 1923 hänen veljensä näytelmään R.U.R. (Rossum's Universal Robots). Robotilla kuvattiin näytelmässä keino-tekoista työntekijää. Esimerkiksi teollisuudessa käytettävässä käsivarsirobotissa on käytetty ihmisen käsivartta mallina, ja sille on pyritty saamaan aikaiseksi samanlaiset toiminnot, kuten oikealla kädellä. /11;1./

Joseph Engelberger kehitti Unimate-robotin vuonna 1953, joka oli ensimmäinen teollisuusrobotti. Tämä robotti aikaansai Unimationin yhtiön rungon ja vuonna 1962 ensimmäinen kaupalliseen käyttöön tarkoitettu Unimate robotti tuli autoteollisuuden käyttöön, jossa se toimi valukoneiden parissa kuumissa olosuhteissa. Vuonna 1978 kehitettiin Unimationin teollisuusrobottimalli PUMA (Programmable Universal Machine for Assembly), joka oli käsivarsirobotti ja aikansa yleisin robottimalli. Teollisuusrobottien kehityksen alussa haluttiin kehittää robotti, joka pystyisi toimimaan kaikissa sovelluksissa, mutta pian kuitenkin huomattiin, että eri sovelluksiin kannattaa

tehdä erilaisia robottimalleja. Kuvassa 1 on ensimmäinen teollisuusrobotti Unimate.  
/11./



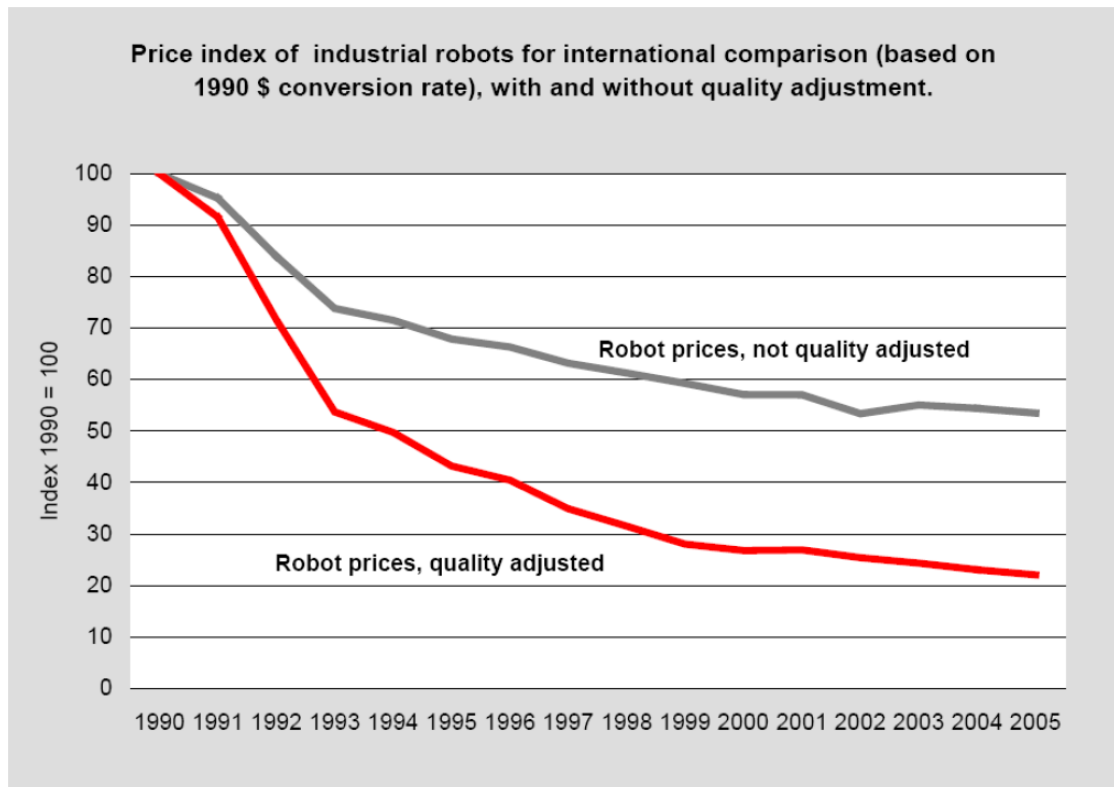
**KUVA 1. Ensimmäinen teollisuusrobotti Unimate /5/**

1970-luvulla markkinoille ilmestyivät Unimationin kanssa kilpailemaan ASEA (nykyinen ABB) ja japanilaisia robotinvalmistajia. 1990-luvulle tultaessa teollisuusrobotit olivat saavuttaneet kehityksensä huipun, kun käyttöön tulivat vaihtovirtaservomootorit. /26./

### **3.2 Teollisuusrobotit tänä päivänä**

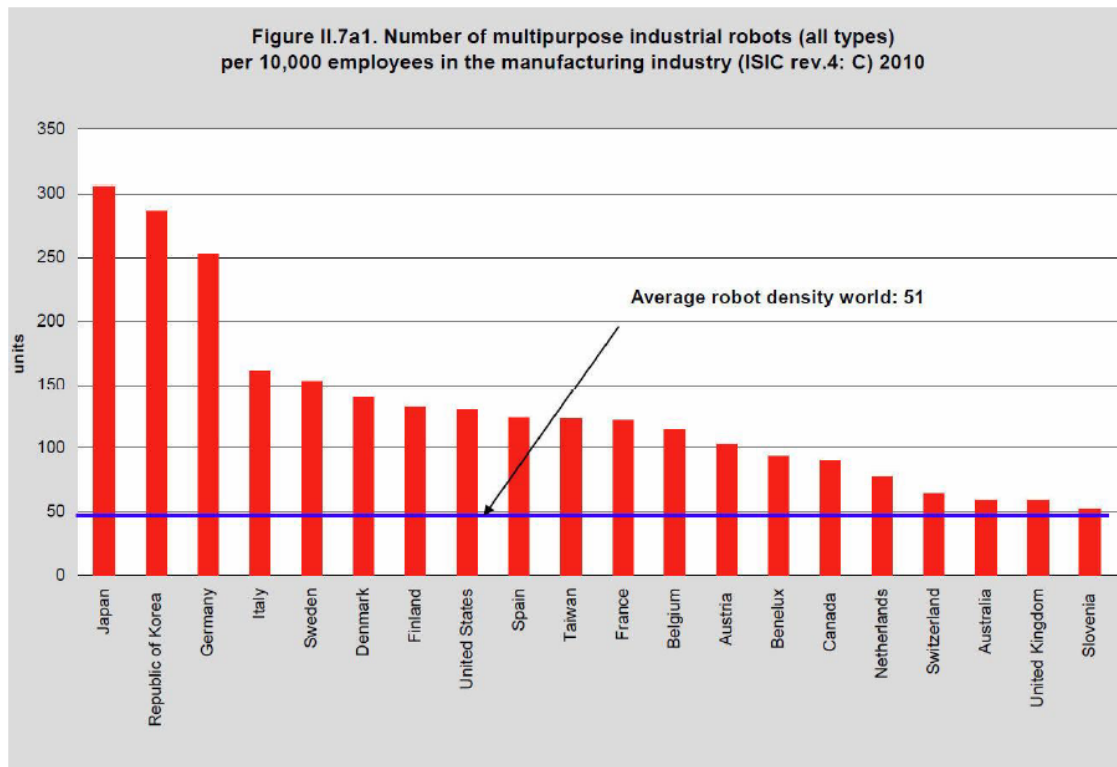
Tällä hetkellä maailmalla on teollisuusrobotteja käytössä yli miljoona ja vuosittain käyttöön tulee uusia robotteja noin 100 000. IFR:n (International Federation of Robotics) mukaan uusien käyttöönotettavien teollisuusrobottien määrä kasvaa entisestään. Tänä päivänä tärkeimpiä ja suurimpia robottien valmistajia ovat muun muassa Fanuc, Kawasaki, Motoman, Toshiba, Denso, ABB, ja Kuka. Robottien hintakehitys on ollut myös huimaa robottien tekniikan kehittymisen lisäksi. 1990-luvulta 2000-luvulle tultaessa hinnat ovat jopa puolittuneet siitä, mitä ne olivat 1990-luvun alussa. Kuvassa 2 näkyy teollisuusrobottien hintakehitys vuodesta 1990 vuoteen 2005. /12;10, s.2;11./



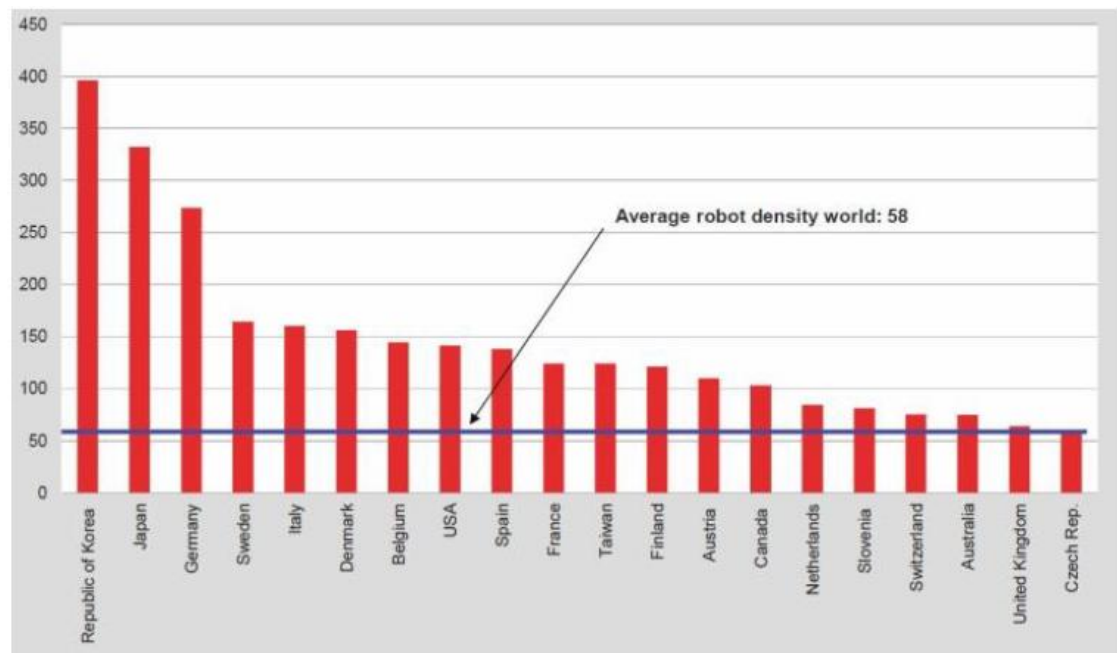


**KUVA 2. Teollisuusrobottien hintakehitys /1/**

Vaikka teollisuusrobottien määrä maailmalla kasvaa, on Suomessa sen sijaan teollisuusrobottien hankintamäärä alkanut laskea vuosikymmenen vaihteessa. Suomi onkin pudonnut robotiikan kärkikymmeniköstä, kun uusien teollisuusrobottihankintojen määrä on vähentynyt ja vanhoja robotteja poistuu käytöstä. Kuvassa 3 ja kuvassa 4 on tilasto teollisuusrobottien määrästä tuotantoteollisuudessa 10 000 työntekijää kohden vuonna 2010. /12;13./



**KUVA 3. Teollisuusrobottien määrä tuotantoteollisuudessa 2010 /12/**



**KUVA 4. Teollisuusrobottien määrä tuotantoteollisuudessa 2012 /13/**

Suomen kansainvälisesti hyvä asema tuotantotekniikassa on alkanut huonontua, ja se näkyy robotiikassa. Kun muilla tilastoiduilla teollisuusmailla robotitiheys kasvaa, niin Suomella se ainoana maana vähenee. Teollisuusrobottikannan vähentyminen Suomessa tapahtuu noin 4 % alenemalla vuosittain. /13./

### 3.3 Teollisuusrobotin määritelmä

ISO 8373 (International Organization for Specification) -standardi määrittelee robotin ja robottimaiset laitteet teollisuuden käytössä ja käytöissä teollisuuden ulkopuolella.

Teollisuusrobotin tulee täyttää standardin ISO 8373 mukaan seuraavat ominaisuudet:

- Toimii automaattisesti
- Uudelleenohjelmoitava
- Voidaan käyttää monissa eri sovelluksissa
- Pystyy tarttumaan osiin ja työkaluihin, sekä liikuttamaan niitä eri vapausasteissa eli eri suuntiin
- Vähintään kolme nivelinen
- Kiinteästi paikallaan oleva tai liikuteltava ja teollisuuden automaatio-sovellusten käytössä
- Mukaan kuuluu myös erillinen ohjauspaneeli (pendantti), jolla teollisuusrobotia voidaan ohjata ja ohjelmoida /9./

### 3.4 Teollisuusrobottien käyttökohteita ja tehtäviä

Teollisuusrobotteja löytyy nykyään kaikilta teollisuuden eri osa-alueilta. Suurimmassa käytössä niitä on konepaja-, metalliteollisuudessa ja elintarviketeollisuudessa. Konepaja- ja metalliteollisuudessa hitsaus on tärkeä käyttökohte, jossa teollisuusrobotteja käytetään. Esimerkiksi autojen tuotannossa korien kasaus ja hitsaus tehdään teollisuusroboteilla. Muita tehtäviä hitsauksen lisäksi ovat maalaaminen, hiominen, kaiveraminen ja erilaiset viimeistelyt, kuten kiillottaminen. /1;3./

Elintarviketeollisuudessa, elektroniikkateollisuudessa ja lääketeollisuudessa teollisuusroboteilla tarttumistekniikka on tärkeää, koska vaaditaan tarkkaa kappaleenkäsittelyä, koska käsiteltävänä on erikokoisia ja muotoisia kappaleita. Teollisuusrobottien tehtävänä elintarviketeollisuudessa ja lääketeollisuudessa on tuotteiden asettaminen toisiin laitteisiin, tuotteiden siirtäminen, pakkaaminen, kuljetuslavojen pakkaaminen ja niiden purkaminen. Elektroniikkateollisuudessa robotit suorittavat samankaltaisia koostamistehtäviä, kuten konepaja- ja metalliteollisuudessa paitsi, että käsiteltävät kappaleet

leet ovat paljon pienempiä, elektroniikan komponentteja, jolloin roboteilta vaaditaan suurempaa työskentelytarkkuutta. /1;3./

### 3.5 Teollisuusrobottien hyödyt ja haitat

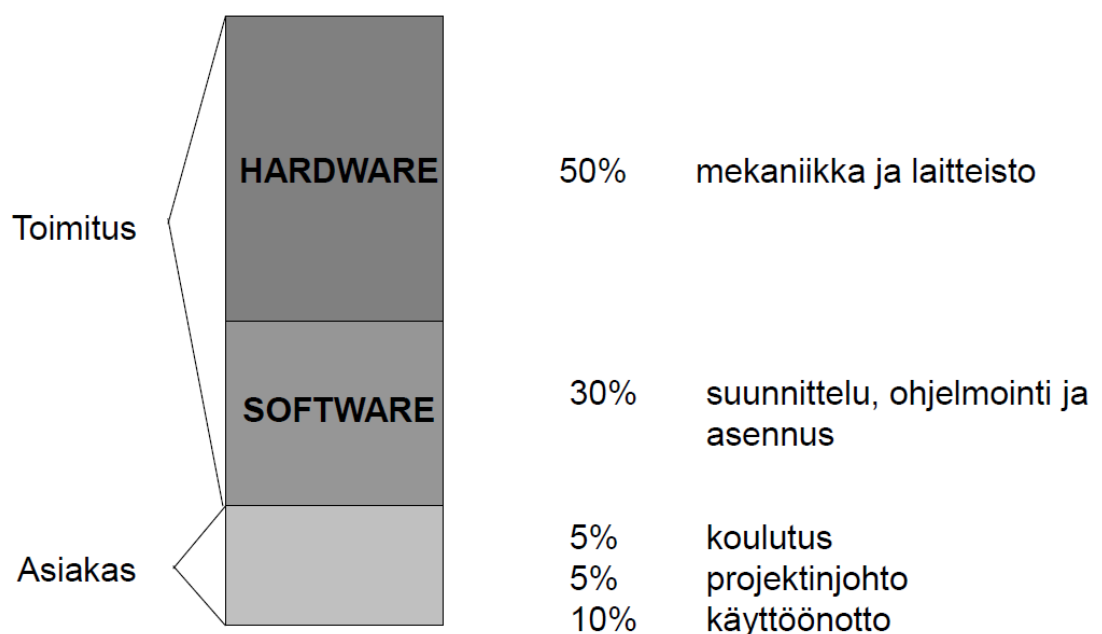
Teollisuusroboteilla voidaan saavuttaa monia hyötyjä yritykselle. Yrityksen käyttökustannukset paranevat, kilpailukyky paranee, tuotteiden laatu paranee ja on tasaisempaa, tuotannon määrä kasvaa, tuotteiden valmistuksesta tulee joustavampaa, materiaalihukka pienenee, työturvallisuus paranee, uusien työntekijöiden palkkaus vähenee sekä työskentelytilaa säästetään. Teollisuusrobotit edesauttavat yrityksen kilpailukyvyn paranemista ja tuotannon määrän kasvua, sillä robotit voivat työskennellä jatkuvasti, suurella nopeudella ilman työkatkoja, koska niiden ei tarvitse pitää taukoja. Tuotteiden laadun paraneminen ja materiaalihukan väheneminen selittyy sillä, että tietty prosessi suoritetaan aina samalla tavalla ja äärimmäisellä tarkkuudella, mihin ihminen ei pysty. Sama robotti voidaan ohjelmoida tekemään useita eri tehtäviä, joita se voi tehdä jopa samanaikaisesti, jolloin valmistuksesta saadaan joustavampaa. /4;18./

Työntekijöiden työturvallisuus ja työhyvinvointi paranee, koska robotit voivat suorittaa tehtäviä, joita ihminen ei pysty tekemään tai ne ovat hankalia ihmiselle. Näitä ovat esimerkiksi painavien taakkojen nostot, kuumat, pölyiset ja muut terveydelle haitalliset työolosuhteet. Kun ihmisen ei tarvitse itse suorittaa terveydelle haitallisia ja vaarallisia töitä, jää ihmisen tehtäväksi robotin työskentelyn valvonta vikatilanteiden varalta. Työturvallisuuden parantumisella saavutetaan tällöin myös säästöjä terveyshuollon kustannuksissa. Työskentelytilaa voidaan hyödyntää paremmin oikein asennetuilla linjastoilla, ja kun linjastoihin on lisätty robotti, säästetään tilaa entisestään, koska roboteilla on aina sama määritelty työskentelyalue, jossa se työskentelee. /4;16./

Vaikka yritys hyötyy robottien käyttöönotosta, jolloin uusien työntekijöiden palkkauksen tarve vähenee ja säästöjä tehdään palkkakustannuksissa, ei se ole hyvä asia työttömälle, kun robotti korvaa ihmisen. Työtä etsivän työnsaantimahdollisuudet vaikeutuvat, kun robotteja otetaan käyttöön ja robottien monikäyttöisyyden ansiosta niitä voidaan käyttää useiden eri tehtävien suorittamiseen. Teollisuusrobottien pääasiallisena tavoitteena ei kuitenkaan ole täysin korvata ihmistä työpaikalla, vaan vapauttaa

työntekijöitä suorittamaan toisia työtehtäviä. Työntekijöitä voidaan esimerkiksi kouluttaa käyttämään ja huoltamaan uusia robotteja. /25./

Aivan robottihankinnan alussa yritys ei hyödy suoraan teollisuusrobotin hankinnasta, koska investointikustannukset uusiin robotteihin ovat suuret ja se vie aikaa investoinnin suuruudesta riippuen, ennen kuin robotti maksaa itsensä takaisin yritykselle ja hankinnasta aletaan tehdä voittoa. Teollisuusrobotti-investointi maksaa teollisuuden käytössä itsensä takaisin noin 2-3 vuodessa ja yksittäisten robottien hinnat ovat noin 20 000 eurosta ylöspäin. Ilman tarkkaa harkintaa teollisuusrobotin hankinta yritykselle ei takaa parempaa tulosta, vaan hankinta ja sen käyttötarkoitus on suunniteltava tarkasti, jotta saataisiin haluttuja tuloksia syntymään. Ainoa asia, joka synnyttää kustannuksia ei ole itse teollisuusrobotin hankintakustannukset, vaan mukana tulevat myös suunnittelukustannukset, joihin kuuluu uusien linjastojen rakentaminen tai vanhan muokkaaminen uutta tilannetta varten. Kun suunnittelu on valmis, täytyy laitteisto asentaa, ohjelmoida ja ottaa käyttöön. Lisäkustannuksia aiheuttavat säännölliset huollot, joita robotti vaatii pysyäkseen toimintakunnossa ja myös työntekijöiden koulutus uusien laitteiden käyttöön, sekä huoltoon vie aikaa aiheuttaen lisämenoja. Jos tulevaisuudessa halutaan lisätä robotin monipuolisuutta ja käyttää sitä uusissa tehtävissä, täytyy sen ohjelmointiin tehdä muutoksia, joka ei myöskään ole ilmaista. Vaikka teollisuusrobotit edesauttavat työturvallisuuden parantumista, voivat ne kuitenkin itse aiheuttaa uusia vaaroja työpaikalle, jotka on otettava huomioon. Kuvassa 5 on teollisuusrobottiprojektin kustannusrakenne. /1;18./



## **KUVA 5. Teollisuusrobottiprojektin kustannusrakenne /1/**

### **4 TEOLLISUUSROBOTTIEN TURVALLISUUS**

Tämä luku käsittelee teollisuusrobottien turvallisuuteen liittyviä asioita, mitä vaaroja teollisuusrobotit aiheuttavat, miksi niitä syntyy ja miten niitä voidaan ehkäistä. Vuosittain sattuu työtapaturmia, joissa on osallisena teollisuusrobotti noin 33, jotka ovat lieviä onnettomuuksia. Vakavia robotitapaturmia maailmalla on sattunut kaiken kaikkiaan 25 ja niistä 23/25:stä on ollut puristumisonnettomuuksia. Kuolemaan johtaneita tapaturmia on ollut kolme, joissa on ollut osallisena teollisuusrobotti. Nämä tapaturmatiedot ovat Timo Malmin tekemän robotteihin liittyvän tapaturmatutkimuksen tuloksia. Teollisuusroboteista on vaikea löytää tapaturmatilastoja, koska tapaturmat, joissa teollisuusrobotti on ollut osallisena, luokitellaan teollisuusonnettomuuksiksi ja ne kirjataan kaikkien muiden teollisuusonnettomuuksien seassa. Seuraavissa luvuissa käsitellään erilaisia vaaratilanteita, vaarojen ehkäisytapoja ja tarkemmin suojalaitteita, joilla voidaan tehdä robotityöympäristöstä turvallisempi. /14;17./

#### **4.1 Vaarat ja niiden aiheuttajat**

Teollisuusrobottien käytöstä syntyvät onnettomuudet voidaan luokitella seuraaviin luokkiin onnettomuuden tyypin mukaan, joista ne voivat aiheutua:

- Isku ja törmäys
- Puristuksiin jääminen
- Mekaaniset viat
- Muut syyt.

Tekstissä käytän sanaa robottisolun, jolla tarkoitan robotin työskentelyaluetta. Robotisolun määrittelykokonaisuudeksi, johon kuuluu robotti, sen ohjauslaitteet ja muut lisä- sekä suojalaitteet. /20./

Vaaratilanteita teollisuusrobotin toimiessa yksin normaaleissa olosuhteissa ei tapahdu juuri ollenkaan. Vaaratilanteet sen sijaan syntyvät useimmiten silloin, kun ihminen on robotin toiminnassa mukana, joko robotin toiminta-alueella tai lähiympäristössä. Tilanteita, jotka aiheuttavat vaaroja ja onnettomuuksia ovat robotin asennus, ohjelmoin-

ti, testaus, säätö ja huolto toimenpiteet, joiden aikana voi tapahtua asioita, joita ei toivoisi tapahtuvan. Jotkin teollisuusrobotit tekevät työtä suurella nopeudella, jonka takia siitä saadut iskut voivat olla erittäin vakavia. Teollisuusrobotti voi osua robotin käyttäjään itse robotin virheellisestä toiminnasta johtuen tai käyttäjän virheestä johtuen ja robottisolun sisäpuolelle menevä henkilö saattaa jäädä esimerkiksi robotin, sekä turvaaidan väliin puristuksiin odottamattoman toiminnan seurauksena. Jos samalla alueelle toiminnassa on useampi robotti voi virheellisestä toiminnasta robotit törmätä toisiinsa tai pahimmillaan ihminen voi jäädä kahden robotin väliin puristuksiin. Kun samaan aikaan on samassa paikassa useampi henkilö työskentelemässä saattaa joku työkaverista vahingossa laittaa robotin päälle vaikka sitä ollaan parhaillaan huoltamassa, jolloin robottisolun sisäpuolella oleva robotinhuoltaja on vaarassa. /16./

Kappale, jota robotti käsittelee voi irrota tarttujasta robotin työskennellessä esimerkiksi siitä syystä, että kappaleenvalmistuksessa on tapahtunut valmistusvirhe, josta johtuen kappale ei ole muodoltaan sellainen, kuin sen kuuluisi olla ja jokin robotin mekaanisessa rakenteessa voi pettää aiheuttaen odottamattomia tilanteita. Muiksi onnettomuuksia aiheuttaviksi syiksi luokitellaan esimerkiksi viat laitteistoissa, jotka hoitavat robotin sähkönsyötön ja robotin ohjauksen. Robotissa olevien korkeapaineilmalinjojen katkeaminen on yksi vaaratekijä ja työskentely-ympäristö voi aiheuttaa robotin likaantumisen, joka aiheuttaa ajan kuluessa ongelmatilanteita sekä vaaroja, jos siihen ei puututa. Ulkopuoliset radiotaajuudet voivat myös häiritä teollisuusrobotin toimintaa. /16./

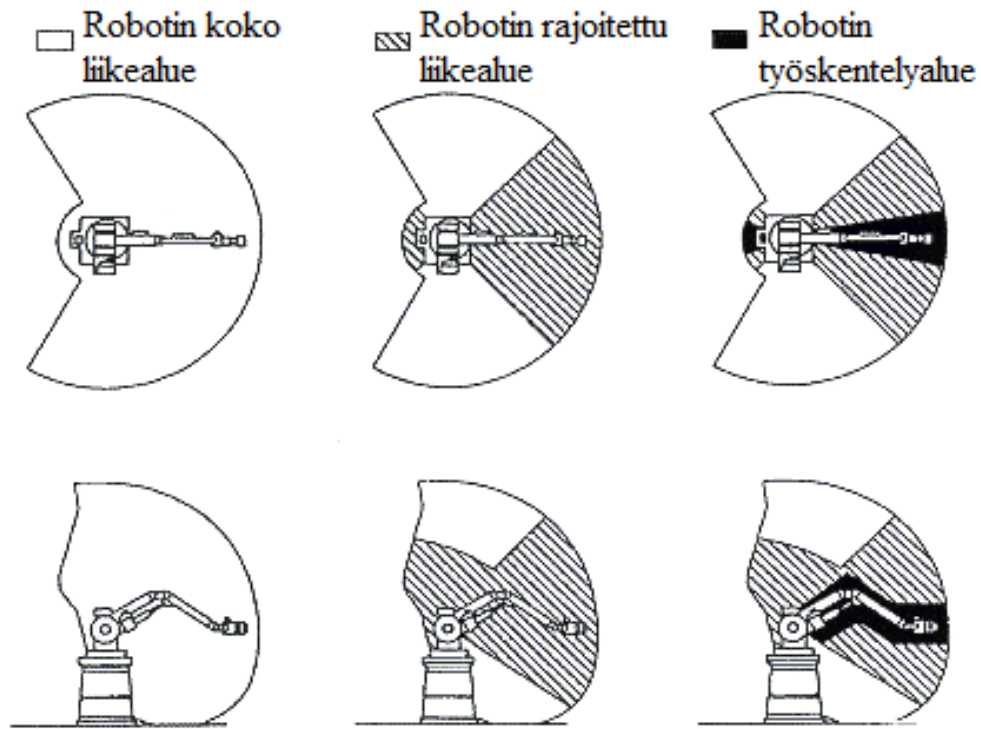
## **4.2 Vaarojen ehkäiseminen**

Teollisuusrobottiympäristössä vaaroja voidaan ehkäistä erilaisin keinoin, kuten henkilöstön perehdyttämisellä työympäristöön, kouluttamalla ne henkilöt, jotka käyttävät teollisuusrobotteja, sekä monilla suojavarusteilla ja turvalaitteilla. Säännölliset robotin huollot, mekaanisen rakenteen, toimilaitteiden sekä työkalujen tarkastukset ennaltaehkäisevät onnettomuuksia, kun mahdollisesti syntyvät viat huomataan tarpeeksi ajoissa ja niihin voidaan puuttua jo ennen, kuin mitään ehtii tapahtua. Työympäristön siistinä pitämisellä luodaan myös edellytykset turvalliselle työskentelylle, kun tavaroita ja työkaluja ei jätetä lojumaan lattialle, jolloin vältytään horjahtamisilta sekä kaatumisilta. Kun laitteiden ympärillä olevaa aluetta ei ole suojattu kunnolla, voi horjahtamisella ja kaatumisella olla vakavia seurauksia, jos alueella on vaarallisia laitteita, joiden alueelle meneminen on kielletty.

### 4.3 Henkilöstön koulutus ja työskentelymenetelmät

Henkilöt, jotka työskentelevät sellaisessa työympäristössä, jossa on teollisuusrobotteja, mutta eivät joudu olemaan tekemisissä robottien kanssa tai niiden välittömässä läheisyydessä tulisi ainakin saada perehdytys työympäristöstä ja robottien vaikutuksesta alueella työskentelyyn. Ne henkilöt, jotka käyttävät ja huoltavat tai työskentelevät robottien välittömässä läheisyydessä tulisi saada koulutus robottien käytöstä sekä turvallisuuskoulutus roboteista ja sen ympärillä olevien suojalaitteiden toiminnasta. Kun työntekijä joutuu käyttämään robottia manuaalisesti, tulisi sen käyttö suorittaa siten, että kaikki suojalaitteet ovat toiminnassa ja käyttö suoritetaan suojalaitteilla erotetun alueen ulkopuolelta. Kun joudutaan tekemään töitä, joissa joudutaan menemään robotin liikealueen rajojen sisäpuolelle (robottisolun sisäpuolelle), kuten huoltoja ja ohjelmointia, eivät suojalaitteet voi olla toiminnassa. Tällöin robotti on otettava pois automaattiohjauksesta ja laitettava se manuaaliohjaukselle tai opetustilaan, jolloin saadaan täysi hallinta robotin toiminnasta. Robotin liikenopeudetkin olisi suotavaa asettaa tarpeeksi alhaisiksi, jottei mahdollisesta törmäystilanteesta syntyisi vaaraa. Suositeltava turvanopeus on 250mm/s. Robotin käyttäjän on muistettava myös varmistaa ennen robotin käyttöä, ettei robotin lähellä ole muita työntekijöitä, joille voisi aiheutua vaaraa sen käytöstä. Henkilösuojainten käyttöä ei pidä myöskään unohtaa, sillä niillä voidaan ehkäistä vammojen syntymistä. Suojalaseilla, kuulosuojaimilla ja oikeanlaisilla työvaatteilla voidaan suojautua esimerkiksi metalliteollisuudessa, jossa teollisuusrobotteja käytetään hitsaukseen, melua ja työympäristössä syntyviä roiskeita vastaan. Kuvassa 6 esitetään esimerkki teollisuusrobotin eri liikealueista. /16./





**KUVA 6. Teollisuusrobotin liikealueet /16/**

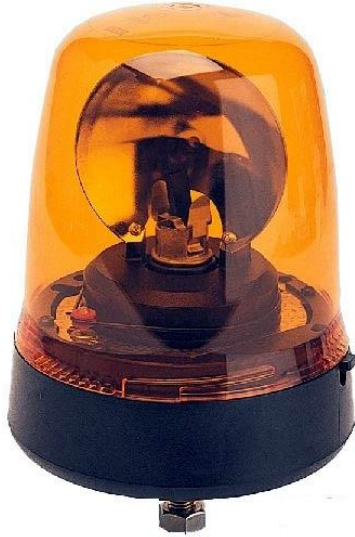
#### 4.4 Turvalaitteet ja suojausmenetelmät

Jotta työympäristöstä, jossa on robotteja toiminnassa, saadaan turvallinen, täytyy robotin työskentelyalue suojata. Robotin työskentelyalueen sisäpuolelle menoa voidaan estää kokonaan esimerkiksi turva-aidoilla, joiden ovissa on turvakatkaisijat ja suojata pysäyttämällä robotin toiminta käyttämällä erilaisia antureita, sekä havainnointilaitteita. Häätä seis painikkeita on myös sijoitettava teollisuusrobotin lähetyville, sellaisiin paikkoihin, joihin on esteetön pääsy. Muilla ulkoisilla varoituskeinoilla voidaan herättää huomiota alueella. Varoittavilla vilkkuvaloilla, sekä merkkivalotorneilla, varoitusäänillä laitteen käynnistyessä, varoituskylteillä ja lattiaan maalatuilla tai teipatuilla turva-alue merkinnöillä voidaan varoittaa työalueella liikkuvia työntekijöitä vaaratilanteiden välttämiseksi. Laitteiston vaarallisuudesta riippuen käyttämällä tarvittavia turvaetäisyyksiä laitteiston ja esimerkiksi turva-aitojen välillä voidaan estää puristuksiin jääminen.

#### 4.5 Merkkivalot ja äänimerkinantolaitteet

Vilkkuvaloilla voidaan varoittaa teollisuusrobotin toiminnasta. Teollisuusrobotin so- luun sijoitetun vilkkuvalon tulisi olla sijoitettu siten, että se näkyy kaikista suunnista

/19/. Laitteen käynnistyessä tai aloittaessa suorittaa jotakin tiettyä työvaihetta voidaan käynnistymisestä varoittaa summerilla, joka antaa kuuluvan äänimerkin. Summeri voi myös olla integroituna merkkivalotorniin. Kuvassa 7 on esimerkki vilkkuvalosta, kuvassa 8 on merkkivalotorni ja kuvassa 9 äänimerkinantolaite.



**KUVA 7. Vilkkuvalo /6/**



**KUVA 8. Merkkivalotorni /22/**

Merkkivalotornia käytettäessä voidaan ilmaista tietyllä hetkellä vallitsevan tilanteen vaarallisuus erivärisillä merkkivaloilla. Merkkivaloon voidaan liittää myös esimerkik-

si valo merkiksi siitä, että robottisolun portti on auki. Kuvassa 9 näkyy esimerkki eri valojen mahdollisista merkityksistä.

	Äärimmäinen vaara, alueella liikkuminen kielletty
	Varoitus, alueella noudatettava varovaisuutta
	Ei vaaraa
	Portti auki

**KUVA 9. Valojen merkitykset /22/**



**KUVA 10. Äänimerkinantolaite /7/**

#### 4.6 Turva-aluemerkinnät ja varoituskyltit

Kulkuväylät voidaan merkitä maalaamalla tai teippaamalla lattiaan alueet, joilla kaikki liikkuminen paikasta toiseen tulisi suorittaa. Kulkuväylämerkintöihin käytettävät värit voivat olla mitä tahansa, kunhan ne näkyvät ja selkeästi viestittävät, että niillä on merkattu alue, jolla saa kulkea. Liikkumiselta rajatut vaaralliset alueet, kuten robottisolun ympärillä oleva alue, voidaan myös merkitä samoin keinoin rajaamalla alue muusta ympäristöstä. Alueen ympärille siis merkitään maalilla tai teipillä alue, jonka sisäpuo-

lelle ei saa mennä. Molemmissa tapauksissa yleisimpänä varoitusvärinä on keltainen, sillä keltaisella viestitetään vaarasta ja varovaisuuden noudattamisesta. Lattiamerkinnät voivat myös sisältää tekstiä, jolla varoitetaan vaarasta. Vaaran luonteesta riippuen siitä voidaan myös varoittaa erilaisilla varoituskylteillä, jotka voivat sisältää kuvia ja tekstiä. Varoituskylteissä on yleensä keltainen pohja, johon kuvat ja teksti on tehty mustalla värillä. Kuvassa 11 on robottisolua ympäröivä alue merkattu lattiamerkinnällä ja kuvassa 12 on esimerkki varoituskyltistä, joka varoittaa alueesta, jossa on robotteja toiminnassa. /8./



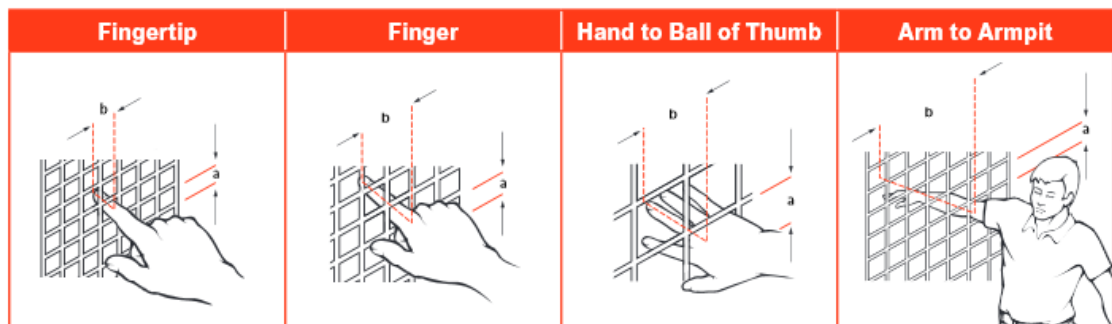
**KUVA 11. Lattiamerkintä robottisolun ympärillä /17/**



**KUVA 12. Varoituskyltti /15/**

#### 4.7 Turva-aidat ja turvakatkaisijat

Turva-aitojen avulla voidaan erottaa robottisolu muusta ympäristöstä. Aitojen avulla estetään esimerkiksi horjahtamisesta aiheutunut tahaton ruumiinosien pääsy robotisolun sisälle ja muiden asiattomien pääsy robotisoluuun. Turva-aidat ovat tehokas ja monipuolinen tapa suojata robottisolua. Turva-aidat takaavat turvallisuuden, sillä ne ovat aina toiminnassa, toisin kuin muut sähköiset suojausmenetelmät, koska niihin voi tulla vikoja ja häiriöitä, jolloin ei voida taata täyttä turvallisuutta. Turva-aidoilla voidaan aidata lähes minkä muotoisia alueita tahansa, kuten aitamallilla, jossa liittymäkohdat on saranoitu, voidaan niiden liittymäkohtia mukauttaa 0-180 asteen kulmissa. Aitoihin on saatavilla myös erilaisia ovimalleja, kuten tavallisia saranaovia, liukuovia ja nostamalla avattavia ovia. Turva-aidan verkossa olevien aukkojen koon valinnalla voidaan suojata robottisoluuun pääsyä erikokoisilta kappaleilta vaarallisuuden mukaan. Kuvassa 13 on esimerkkejä erikokoisten kappaleiden läpipääsystä verkkoaidan läpi.



**KUVA 13. Verkkoaidan aukkojen vaikutus suojauksessa /29/**

Turvallinen robottisoluuun kulkeminen turva-aidan oven kautta voidaan luoda sähkömekaanisten turvakytkimien ja rajakytkimien avulla, jolloin laite lopettaa toimintansa oven avauduttua. Sähkömekaanisten rajakytkimien lisäksi voidaan käyttää myös magneettisia rajakytкимиä, jotka toimivat ilman suoraa kosketusta. Magneettiset rajakytkimet sopivat myös paremmin likaisiin olosuhteisiin niiden liansietokyvyn ansiosta. Kuvissa 14 ja 15 on esimerkkejä turva-aidoista, sekä kuvassa 16 on turvakytkin, mekaaninen rajakytkin ja magneettinen rajakytkin. /23;28./



**KUVA 14. Turva-aita teollisuusrobotin ympärillä /28/**



**KUVA 15. Turva-aita, jossa liittymäkohdat on varustettu saranoidella /24/**



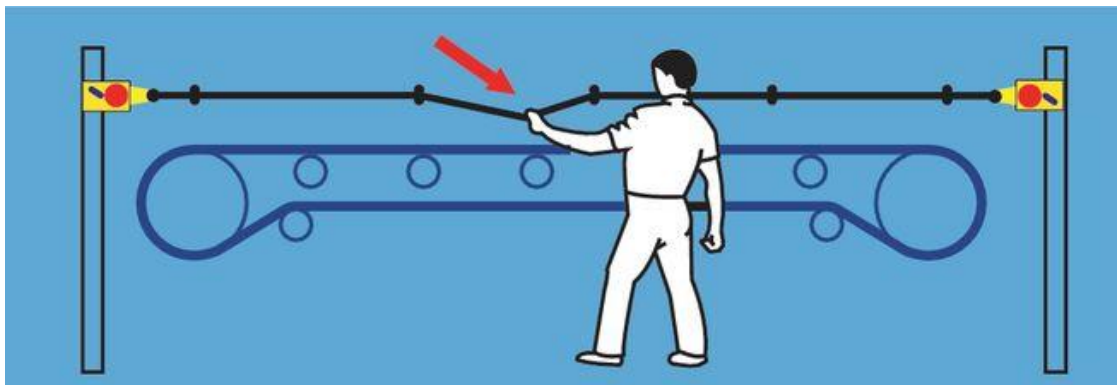
**KUVA 16 Turvakytkin, rajakytkin ja magneettinen rajakytkin /20/**

Hätä-seis -painikkeita ei luokitella pääsuojalaitteiksi, vaan ne luokitellaan muita suoja-laitteita täydentäviksi suojalaitteiksi. Tämä johtuu siitä, että ne eivät estä millään ta-valla vaara-alueelle menemistä, eivätkä ne edes valvo vaara-alueelle kulkua. Hätä-seis painikkeet on tarkoitettu vain vaaratilanteita varten, jolloin voidaan katkaista laitteen toiminta välittömästi. Painikkeen käyttö tapahtuu yleensä aivan viime hetkellä ennen onnettomuutta tai vasta sitten, kun onnettomuus on jo ehtinyt tapahtua ja laite täytyy saada äkkiä pysähtymään. Hätä-seis -painikkeella tapahtuva pysäytystoiminto voidaan tehdä kahdella tavalla. Niin sanotusti hallitsemattomalla pysäytyksellä sähkö katkais-taan välittömästi kaikilta laitteilta painiketta painettaessa, mutta tällöin esimerkiksi moottorit voivat pyöriä vielä jonkin aikaa vapaasti. Toinen tapa on niin sanottu hallittu pysäytys, jossa sähköt pidetään päällä, sen aikaa kunnes on saatu aikaiseksi jarrutus, jonka seurauksena liike pysähtyy. Liikkeen pysähtyessä sähkö katkaistaan laitteelta. Hätä-seis -painikkeet täytyy sijoittaa sellaisiin paikkoihin, joihin on esteetön pääsy, sekä niitä tulisi olla ainakin yksi paikassa, jossa voi olla välitön vaara ja sen on oltava painettavissa vaara-alueelta. Robottisolun ympärille olisi hyvä sijoittaa hätä-seis pai-nikkeita. Teollisuusrobotin ohjaukseen ja ohjelmointiin käytettävässä pendantissa olisi hyvä myös olla hätä-seis -painike, jotta saadaan äkkiä robotti pysähtymään. Liukuhih-nastoissa voidaan käyttää koko hihnaston matkalle ulottuvaa hätä-seis -laitteistoa, jos-sa hätä-seis -katkaisijat on yhdistetty toisiinsa rautavaijerilla. Kun vaijerista vedetään missä tahansa kohtaa hihnaa, katkeaa laitteen toiminta. Tällaisen hätä-seis -laitteiston on myös toimittava vaijerin löystyessä, joka varmistaa sen, ettei vaijeri ole poikki. Tällä tavoin voidaan taata laitteiston turvallinen toiminta. Hätä-seis -painikkeet ovat

väriltään punakeltaisia, jossa painike osa on punaisella ja kotelo on keltaisella värillä. Kun hätä-seis -painiketta painetaan, pitää painikkeen jäädä painetuksi, eikä se saa palautua automaattisesti. Kuvassa 17 on hätä-seis -painike ja kuvassa 18 on liukuhihnoissa käytettävä vaijerilla toimiva hätä-seis -laitteisto. /19;21./



**KUVA 17. Hätä-seis -painike, jossa painikkeen palautus tehdään painiketta kääntämällä /27/**



**KUVA 18. Hätä-seis -laitteisto vaijerilla /21/**

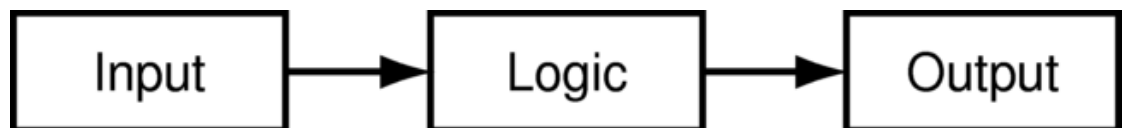
#### 4.8 Anturit

Robottisolun turvallisuuden lisäämiseksi käytetään monenlaisia antureita ja ympäristön havainnointilaitteita, joilla voidaan valvoa kulkua vaara-alueelle. Näillä suojausmenetelmillä on vähintään kaksi toimintoa, jotka niiden täytyy täyttää. Kun havaitaan,



että henkilö menee vaara-alueelle, täytyy laitteen sähköjen katketa ja toimintojen pysähtyä. Henkilön ollessa vaara-alueella täytyy myös laitteen takaisin päälle kytkentä olla estetty. Näitä menetelmiä käytetään, kun turva-aitaa ei ole robottisolun ympärillä tai niitä voidaan käyttää lisäsuojana aitojen lisäksi. Eri anturit ja havainnointilaitteet ovat hyvä suojausmenetelmä paikoissa, joissa tarvitsee usein kulkea laitteen toiminta-alueelle.

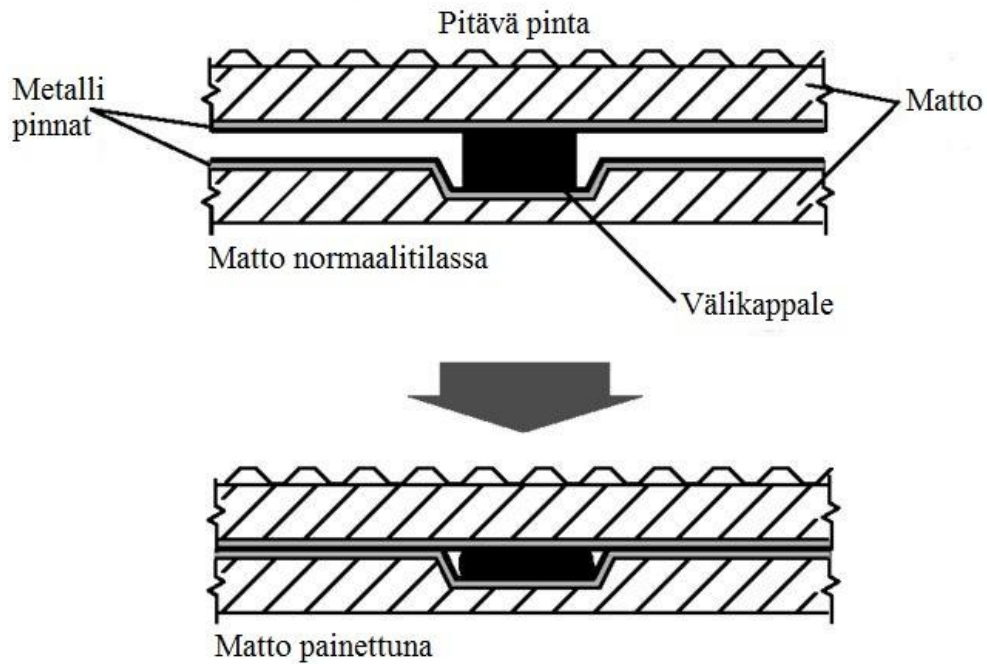
Yksittäinen anturi tai muu havainnointilaitte ei yksinään muodosta suojalaitetta, vaan se muodostuu eri laitteiden muodostamasta kokonaisuudesta. Suojalaitteisto muodostuu sisääntulolaitteesta, logiikasta ja ulostulolaitteesta. Sisääntulolaitte eli anturi havaitsee muutoksen tilassa eli vaara-alueelle menemisen. Logiikka käsittelee anturilta tulevan viestin ja sen perusteella kytkee laitteen pois päältä tai päälle. Ulostulolaitte, esimerkiksi moottori, on laite, jota logiikka ohjaa, ja se mahdollistaa teollisuusrobotin liikkeit. Luotettava valvonta vaara-alueelle kulusta saadaan aikaiseksi, kun antureiden ja muiden havaintolaitteiden toiminta perustuu jatkuvan signaalin katkaisuun. Seuraavissa kappaleissa käydään tarkemmin läpi eri anturien ja havainnointilaitteiden käyttösovelluksia. Kuvassa 19 on kaavio suojalaitteiston rakenteesta. /14;21./



**KUVA 19. Turvalaitteiston rakennekaavio: Sisääntulo, logiikka ja ulostulo /21/**

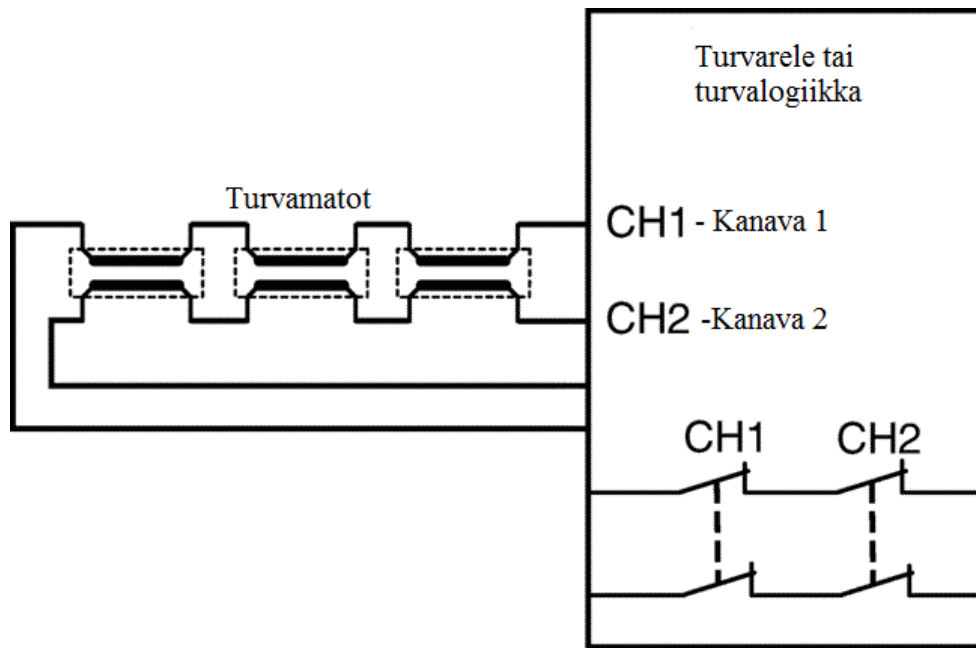
#### **4.8.1 Turvamatto**

Turvamatoilla voidaan katkaista laitteen toiminta, kun maton päälle astutaan. Mattoja voidaan liittää toisiinsa, ja niillä voidaan turvata erikokoisia- ja muotoisia alueita. Turvamaton rakenne koostuu kahdesta rinnakkaisesta metallipinnasta maton sisällä, jotka on erotettu toisistaan sähköä johtamattomilla välikappaleilla. Kuvassa 20 näkyy turvamaton rakenne.

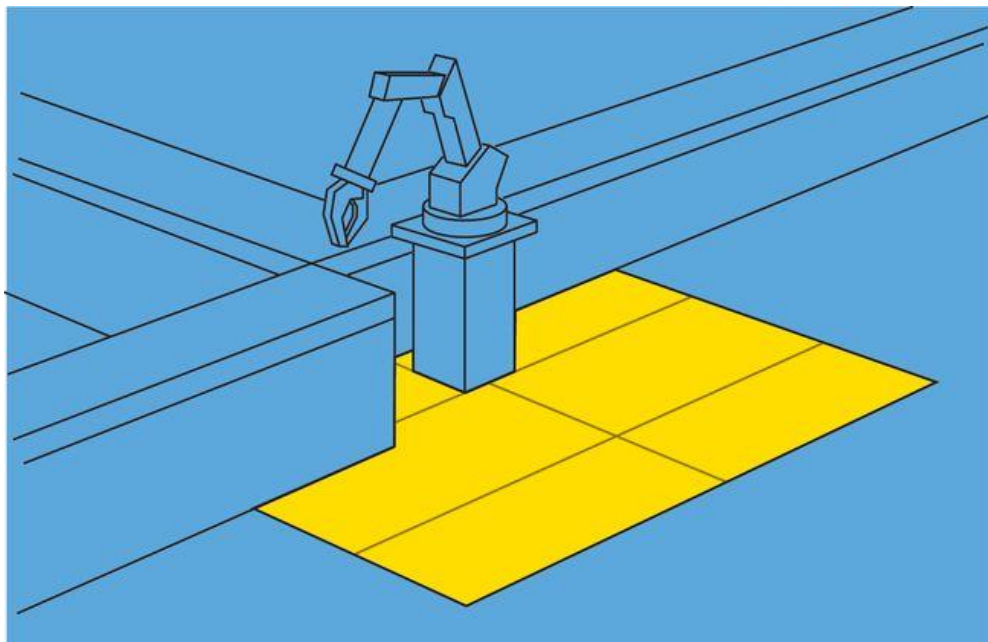


**KUVA 20. Turvamaton rakenne /21/**

Maton kytkentä voidaan tehdä kaksi-johtimisella tai nelijohtimisella kytkennällä. Kaksijohtimista kytkentää käytettäessä metallipinnat erotetaan toisistaan päätevastuksella. Nelijohtimista kytkentää käytettäessä molemmalle metallipinnalle menee kaksi johdinta. Toinen metallipinta on kytketty turvareleen tai turvalogiikan ensimmäiseen kanavaan ja toinen metallipinta kakkoskanavaan. Molempiin metallipintoihin johdetaan sähköä ja kun maton päälle astutaan aiheuttaa se oikosulun näiden pintojen välille, jotka on kytketty eri kanaviin, jonka turvarele tai turvalogiikka huomaa ja katkaisee laitteen toiminnan sen seurauksena. Kuvassa 21 on turvamattojen kytkentäkuva, ja kuvassa 22 on turvamattoja laitettuna teollisuusrobotin ympärillä olevalle vaara-alueelle. /21./



**KUVA 21 Turvamattojen kytkentä /21/**

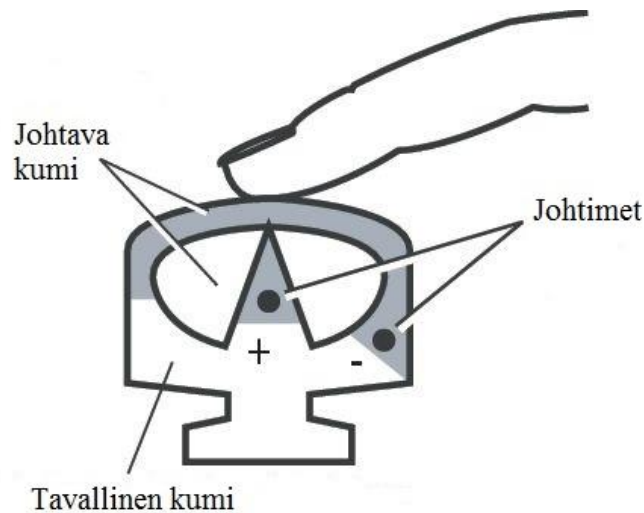


**KUVA 22. Turvamattoja teollisuusrobotin ympärillä /21/**

#### 4.8.2 Tuntoreuna

Tuntoreunoja voidaan laittaa vaarallisen laitteiston ympärillä olevien pöytätasojen reunoille. Kun pöydän reunoilla olevaa tuntoreunaa painetaan, pysähtyy laitteen toiminta. Painallus tuntoreunaa vasten voi aiheutua esimerkiksi horjahtamisesta ja tällä tavoin voidaan suojautua teollisuusrobotin liikealueelle jouduttaessa mahdollisilta robotin aiheuttamilta iskuilta. Tuntoreunoja voidaan kiinnittää myös laitteiden liikku-

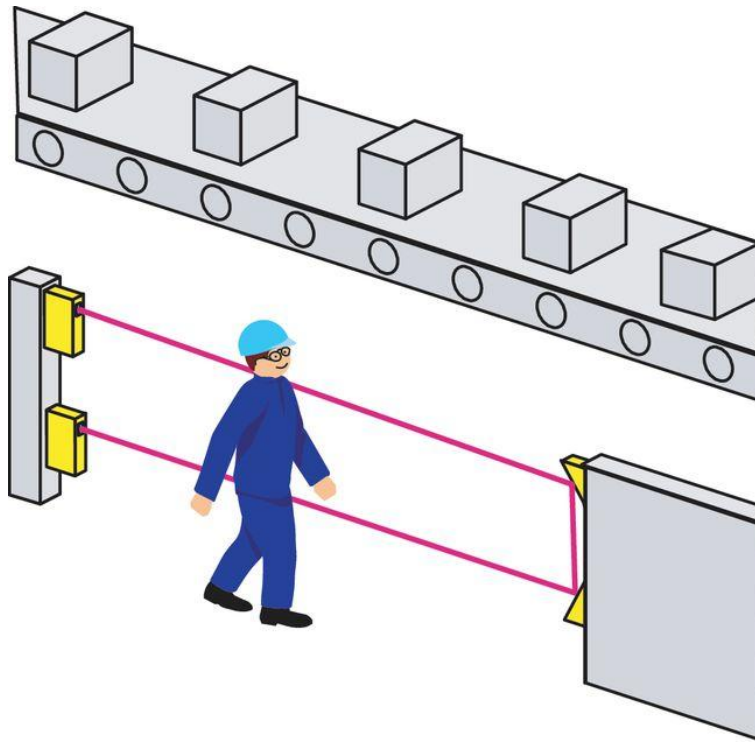
viin osiin. Nosto-ovissa tuntoreunat ovat yleisessä käytössä, ja niiden avulla estetään oven väliin jääminen. Tuntoreunan rakenne koostuu tavallisesta kumista ja sähköä johtavasta kumista. Kumirakenteen sisällä kulkee kaksi johdinta, joiden loppupäässä on päätevastus. Tuntoreunan painaminen aiheuttaa oikosulun, jonka turvarele tai turvalogiikka huomaa ja katkaisee laitteen toiminnan. Kuvassa 23 on esitetty tuntoreunan rakenne. /21./



**KUVA 23. Tuntoreunan rakenne /21/**

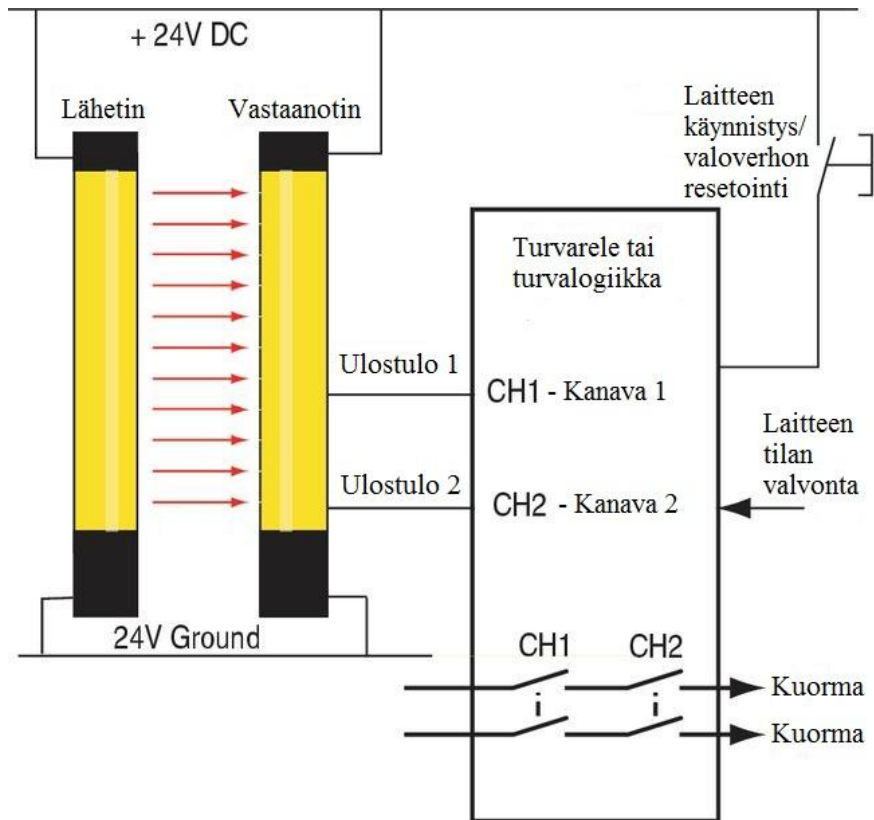
#### 4.8.3 Valokenno ja valoverho

Valokennojen ja valoverhojen avulla voidaan valvoa vaara-alueelle kulkua niillä luotujen infrapunavalosäteiden avulla. Nämä suojamenetelmät ovat hyviä sellaisissa paikoissa, joissa laitteen toiminta-alueelle joudutaan kulkemaan usein. Valokennon erona valoverhoon on se, että sillä luodaan yksi valosäde, kun taas valoverholla luodaan seinämä useilla valosäteillä. Valokennossa voidaan käyttää infrapunavalon lisäksi näkyvää valoa. Valokennolaitteisto voi muodostua kahdella tapaa. Siinä on lähetin osa, joka lähettää valosäteen ja vastaanotin osa, joka vastaanottaa valonsäteen. Lähetin ja vastaanotin osa voi olla myös samassa laitteessa, jossa lähetetty valo palaa peilin kautta takaisin. Kuvassa 24 on laitteistoon kulku suojattu valokennolla /21./



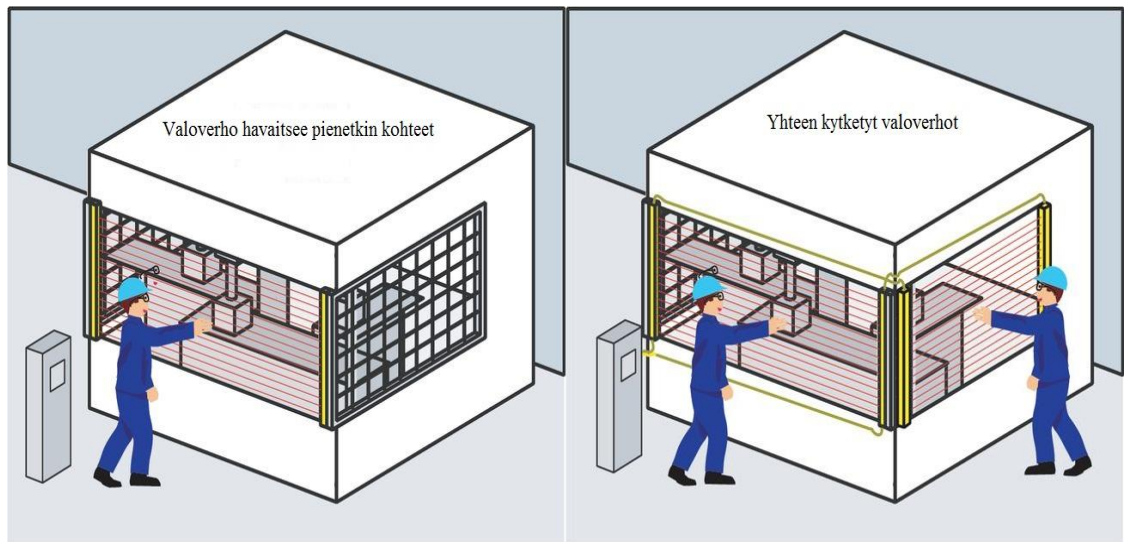
**KUVA 24. Valokennon valo heijastuu peilien kautta vastaanottimelle /21/**

Valoverholaitteisto koostuu lähettinosasta ja vastaanotinosasta. Valoverhon toiminta perustuu infrapunavalon katkeamiseen. Kun yhdenkin valosäteen pääsy lähettimeltä vastaanottimelle estyy, huomaa turvarele tai turvalogiikka muutoksen ja katkaisee laitteen toiminnan. Kun este lähettimen ja vastaanottimen väliltä häviää, ei laite saa jatkaa toimintaansa ennen, kuin valoverho resetoidaan. Valoverhoissa on kaksi ulostuloa, jotka menevät pois päältä havaittaessa este lähettimen ja vastaanottimen välillä. Jos toinen näistä ulostuloista ei vaihda tilaa, katkeaa laitteen toiminta, mutta laitetta ei saa käyntiin ennen, kuin valoverholaitteisto on korjattu. Valoverhossa on myös laitteen tilan valvonta, jolla voidaan valvoa, onko laite päällä vai pois päältä. Kun valoverho katkaisee laitteen toiminnan, valoverhon resetointi ei käynnistä laitetta uudelleen ennen, kuin laite on ensin pysähtynyt. Lähettimen lähettämät valonsäteet vilkkuvat vuorotellen tietyllä taajuudella, jotka lähettimelle määritetty vastaanotin havaitsee. Tällä tavoin voidaan estää muualta ympäristöstä aiheutuvat valohäiriöt ja muiden samanlaisten suojalaitteiden aiheuttamat häiriöt. Kuvassa 25 on valoverhon kytKentä. /21./

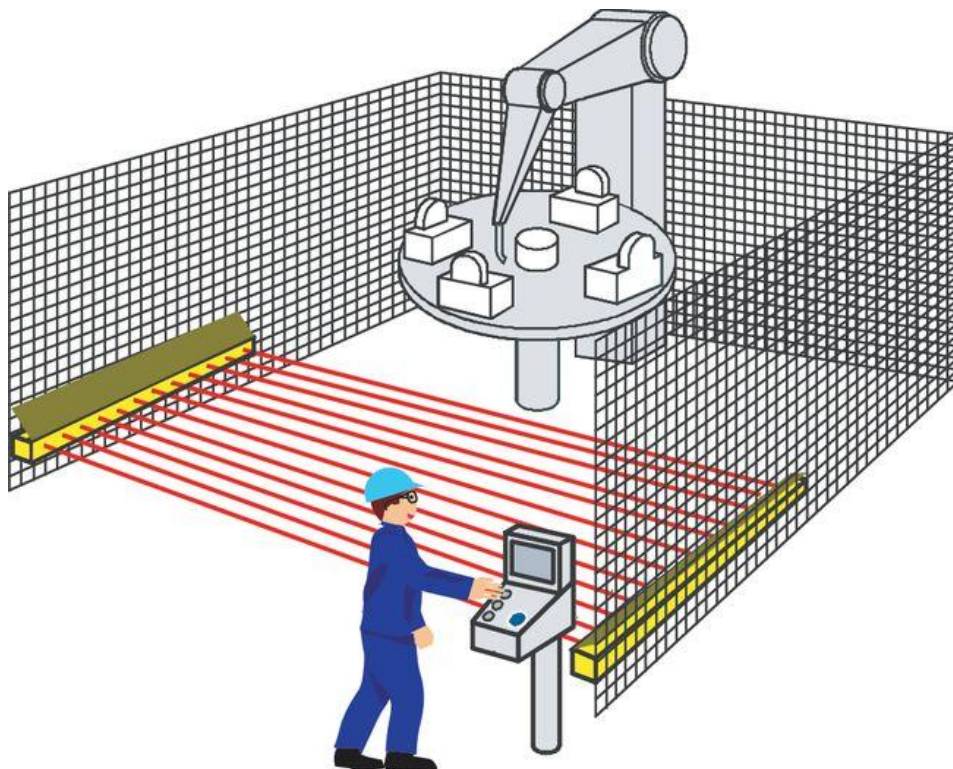


**KUVA 25. Valoverhon kytkentä /21/**

Valokennoilla voidaan havaita suurien kohteiden kulkua vaara-alueelle, kuten ihmisen koko vartalo. Valoverhoilla sen sijaan voidaan havaita pienempien kohteiden, kuten käden joutumista vaara-alueelle. Valoverhoja voidaan myös kytkeä yhteen, jolloin saadaan aikaiseksi yhtenäinen valoverho, vaikkapa kohteen ympärille. Osa valoverhon valosäteistä voidaan ottaa pois päältä ja havaintoaluetta muokata tällä tavoin sallien esimerkiksi prosessissa syntyvien kappaleiden läpikulku valoverhon ohi. Kuvassa 26 on pienen kappaleen tunkeutuminen vaara-alueelle suojattu valoverholla ja kuvassa 27 erotetaan työntekijä teollisuusrobotin vaara-alueesta valoverholla. /21./



**KUVA 26. Suojaus valoverholla ja yhteen kytketyillä valoverhoilla /21/**

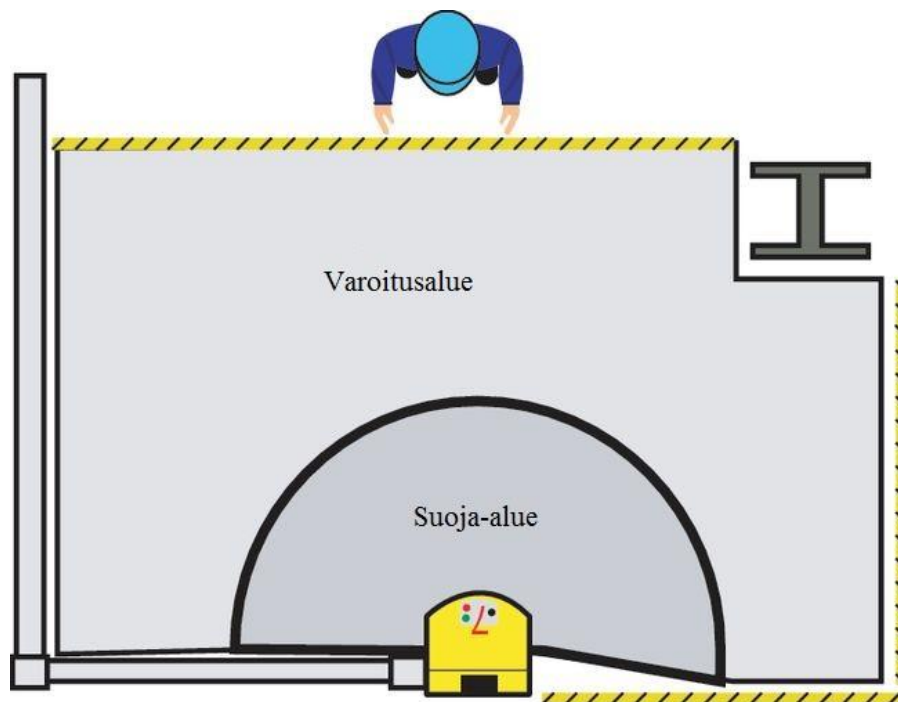


**KUVA 27. Valoverho erottaa työntekijän teollisuusrobotin vaara-alueesta /21/**

#### 4.8.4 Turvalaserskanneri

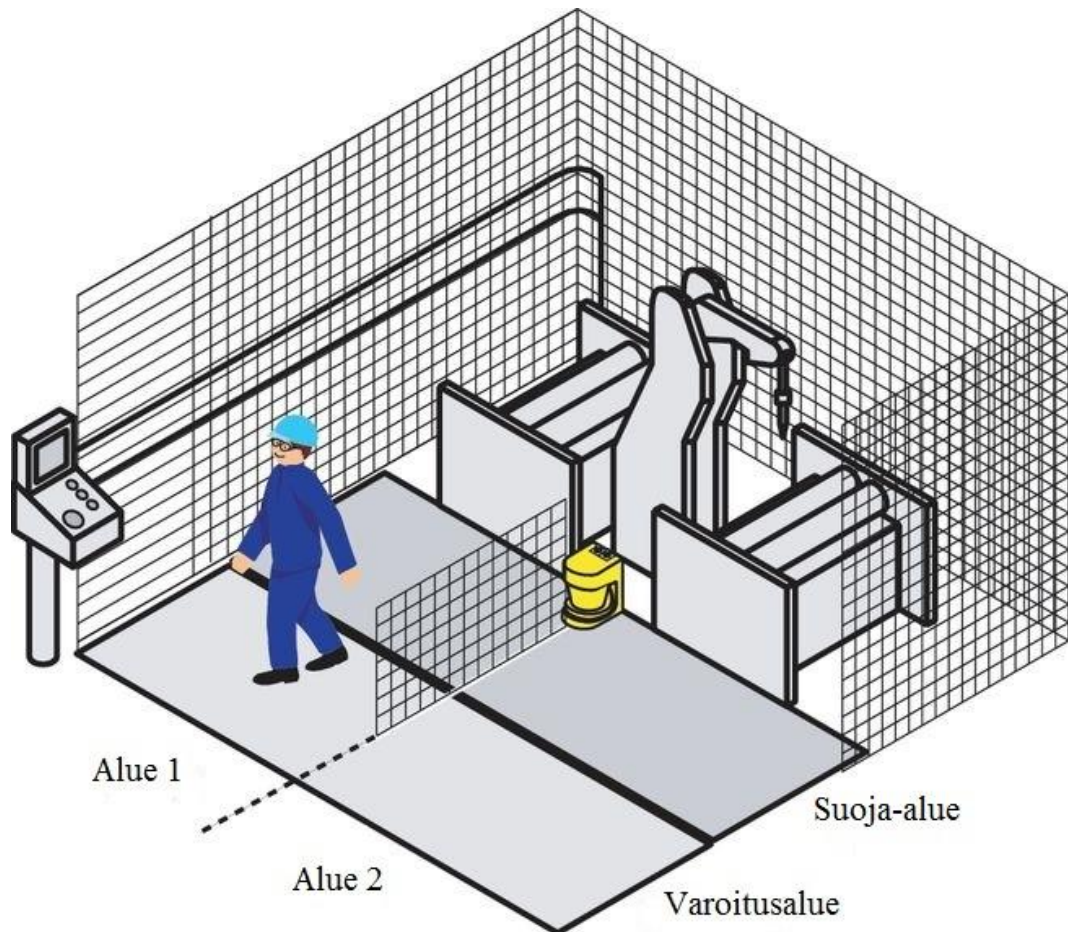
Turvalaserskanneri pystyy määrittämään tarkasti kohteen etäisyyden ja paikan sen valvomallaan alueella. Turvalaserskannerin toiminta perustuu, sen lähettämän valopulssin kulkemisajan mittaukseen kohteeseen ja takaisin skannerille. Skannerissa on pyörivä peili, joka heijastaa valonsäteet ja luo kaarenmuotoisen valvonta alueen. Peilin kulman avulla määritetään kohteen paikka ja valosäteen kulkuajalla määritetään

kohteen etäisyys. Turvalaserskanneri muodostaa kaksi valvottavaa aluetta, varoitusalueen ja turva-alueen. Kohteen mennessä varoitusalueelle, laitteen toimintaa ei katkaista vaan vaara-aluea lähestyvää henkilöä varoitetaan esimerkiksi äänimerkillä tai vilkkuvalolla. Kun henkilö menee suoja-alueen sisään, katkaisee turvalaserskanneri laitteen toiminnan, koska ollaan jo lähellä vaara-aluea. Poistuttaessa vaara-alueelta, laite ei saa jatkaa automaattisesti toimintaansa, vaan skanneri pitää resetoida, jonka jälkeen laite lähtee uudelleen päälle. Yhdellä turvalaserskannerilla voidaan valvoa samanaikaisesti useampaa aluetta. Kuvassa 28 näkyy varoitusalue ja suoja-alue. Kuvassa 29 on esimerkki kahden eri alueen valvonnasta. /21./



**KUVA 28. Turvalaserskannerin valvonta-alueet /21/**





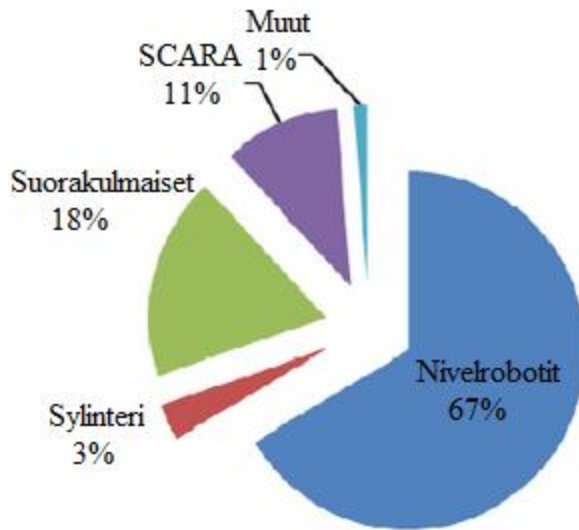
**KUVA 29. Kahden alueen valvonta samanaikaisesti: teollisuusrobotin työskennellessä alueella 2 voidaan alueelle 1 kulkea /21/**

## 5 TEOLLISUUSROBOTTIEN RAKENTEET

Teollisuusrobotit voidaan rakenteensa perusteella jakaa neljään pääkategoriaan:

- Suorakulmaiset robotit (nosturi robotit)
- Sylinterirobotit, napakoordinaattirobotit ja SCARA (Selective Compliance Assembly Robot Arm) robotit
- Rinnakkaisrakenteiset robotit (delta robotti)
- Nivelrobotit (käsivarsirobotti).

Kuvassa 30 on jaoteltu teollisuusrobottien määrä rakenteen mukaan.

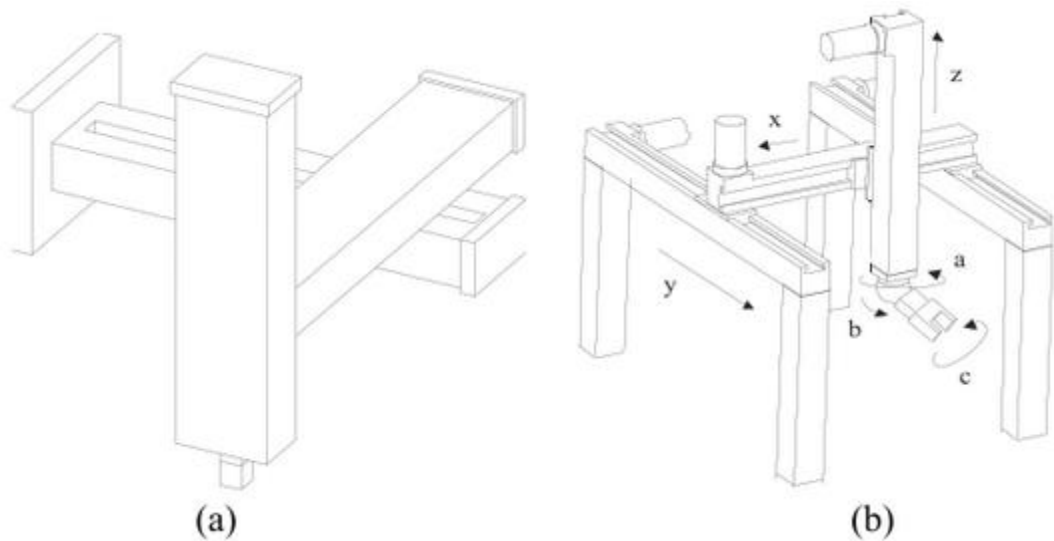


**KUVA 30. Teollisuusrobottien jakautuminen rakenteen mukaan /10, s.5/**

Teollisuusrobotin jaon johonkin näistä kategorioista määrittelee sen kolmen päänive-  
len liikeasettelu. Robotin nivelen käsittää itse kääntyvä nivelosa ja sen liikkeen ai-  
kaansaava moottori. Seuraavissa kappaleissa käsitellään näiden teollisuusrobottien  
ominaisuuksia. /10, s.4-5;26./

### 5.1 Suorakulmaiset robotit

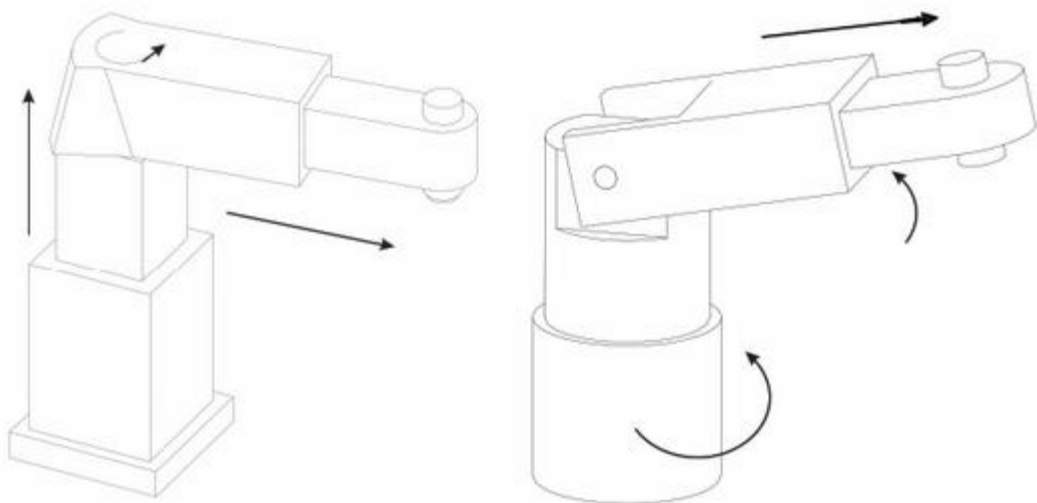
Suorakulmaisissa roboteissa on lineaarisesti liikkuvat nivelet, jotka liikkuvat x, y ja z  
koordinaattisuunnissa. Lisäksi niissä voi olla erikseen nivelet pyörivää liikettä varten.  
Kuvassa 27 b), nivelet a, b ja c pyörivät. Suorakulmaisista robotteja käytetään teollisuu-  
dessa kappaleiden työstötehtäviin, ja ne ovat erittäin tarkkoja, koska niiden rakenne on  
jäykkä. Suuria suorakulmaisista robotteja käytetään myös satamissa konttien siirtoon ja  
laivojen siirtoon. Kuvassa 31 on pienen ja suuren suorakulmaisen robotin rakenneku-  
va. /10, s.5./



**KUVA 31. a) Pieni suorakulmainen robotti ja b) suuri suorakulmainen robotti (nosturirakenne) /10, s.5/**

## 5.2 Sylinterirobotit ja napakoordinaattirobotit

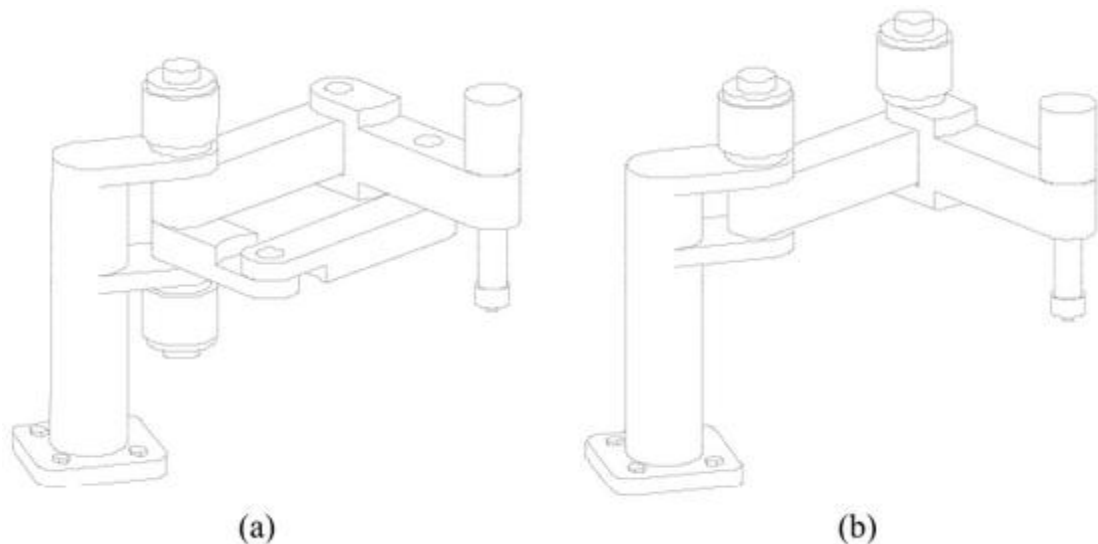
Sylinterirobotissa on yksi kääntyvä nivel, joka kääntää robotin suuntaa sivuttain ja kaksi niveltä, joiden liikkeet ovat lineaarisia. Napakoordinaattirobotissa on kaksi kääntyvää niveltä, joista toinen kääntää robotin suuntaa sivuttain ja toinen pystysuunnassa, sekä yksi lineaarinen nivel. Napakoordinaattirobotissa lineaarisen nivelen päässä voi olla lisäksi myös nivel pyörivää liikettä varten. Näitä teollisuusrobotteja käytetään kappaleenkäsittely-, kokoamis- ja hitsaustehtävissä. Kuvassa 32 on sylinterirobotti vasemmalla ja napakoordinaattirobotti oikealla. /10, s.6-7;5./



**KUVA 32. Sylinterirobotti ja napakoordinaattirobotti /10, s.6/**

### 5.3 SCARA-robotit

SCARA-robotin rakenne muodostuu kiinteästä alustasta, johon yksi tai useampi kääntyvä nivel on kiinnitetty. Tässä robottimallin rakenteessa on kääntyviä niveliä, jotka tekevät robotin sivusuuntaiset liikkeet ja lineaarinen nivel, joka liikkuu pystysuunnassa. Lineaarisen nivelen päässä voi olla nivel, joka mahdollistaa pyörivän liikkeen. Tavallisimpia tehtäviä SCARA-robotilla ovat nopeat kappaleenkäsittely, pakkaustehtävät lääke- ja elintarviketeollisuudessa sekä elektroniikkalaitteiden kokoamistehtävät. Kuvassa 31 on kaksi erilaista SCARA-robotin rakennetta. Kuvan 33 kohdassa a) molemmat pyörivät nivelet on kiinnitetty kiinteään alustaan ja kohdassa b) toinen kääntyvä nivel on kiinteässä alustassa, sekä toinen on kääntyvän akselin päässä. /10, s.6-7./

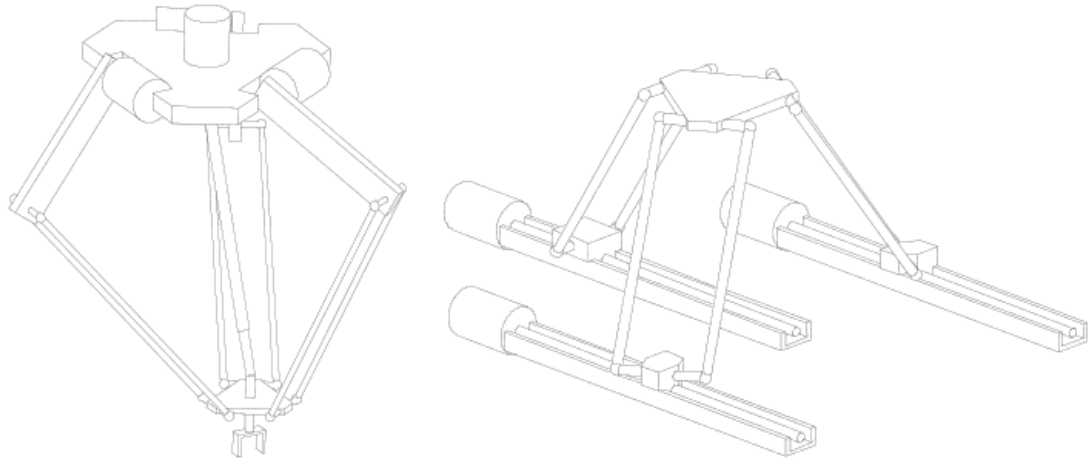


**KUVA 33. Esimerkkejä SCARA-robotin rakenteesta /10, s.7/**

### 5.4 Rinnakkaisrakenteiset robotit

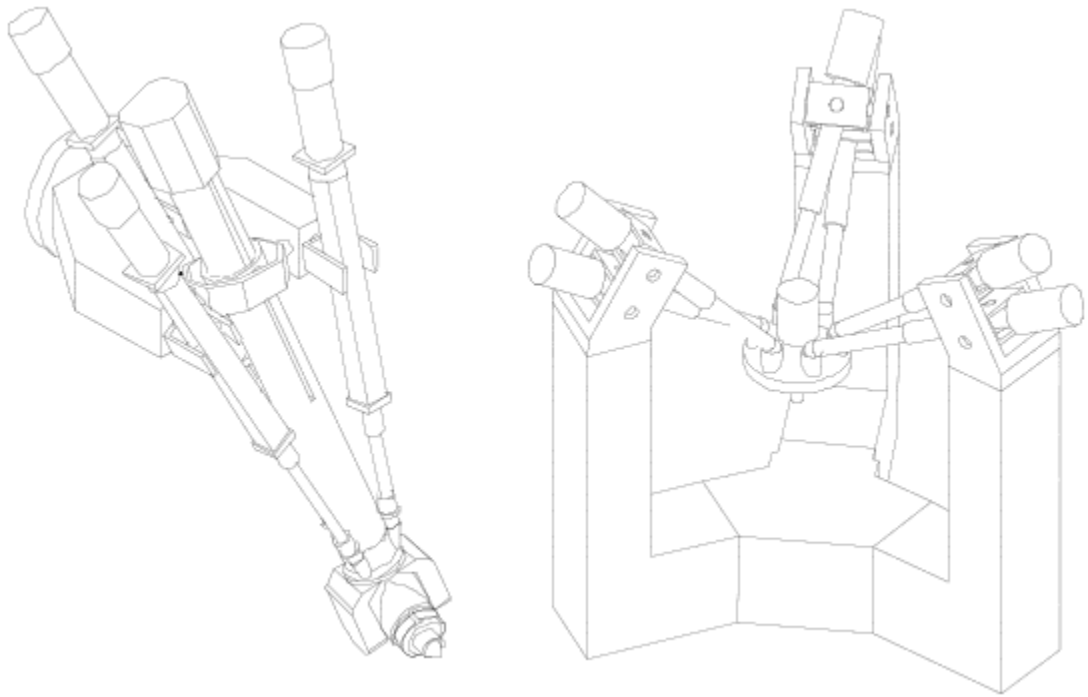
Tämän robottikategorian nimi tulee siitä, että siinä robotin raajat ovat toisiinsa nähden rinnakkain. Yleisin malli tässä robotti kategoriassa on niin sanottu delta-robotti. Siinä on kiinnitetty kolme tankoa kiinteään alustaan, ja ne voivat liikkua alustaan nähden pystysuunnassa. Näiden tankojen päässä on pyöreät liitosnivelet, joista lähtee kustakin kaksi tankoa, jotka ovat samalla tavalla kiinnitettynä liikkuvaan alustaan, jossa on kiinni robotin tarttuja tai jokin muu työkalu. Neljännellä tangolla voidaan mahdollistaa liikkuvassa alustassa olevalle työkalulle pyörivä liike. Nämä robotit ovat erittäin nopeita ja niitä käytetään nopeissa kappaleenkäsittely tehtävissä elintarvike- ja lääketieteellisuudessa. Tässä kategoriassa on muitakin malleja, kuten Triaglide, jonka työs-

Liukunivel voi olla melko suuri, siinä olevien liukunivelten ansiosta. Liukunivel mahdollistaa pitkiä liikkeitä eteen ja taakse. Liukunivel voi myös kääntyä. Kuvassa 34 on vasemmalla yleisimmän rinnakkaisrakenteisen robotin Delta-rakenne ja oikealla on Triaglide-robotin rakenne. /10, s.7-8./



**KUVA 34. Delta-robotti ja Triaglide-robotti /10, s.8/**

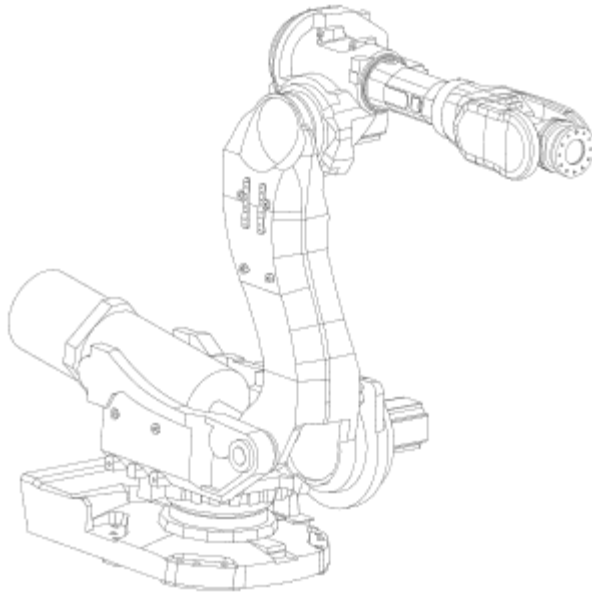
Kolmiomaisen rakenteen omaava rinnakkaisrakenteinen robottimalli Tricept soveltuu työstötehtäviin sen jäykkyyden ja tarkkuuden ansiosta. Tätä mallia on vasta viime aikoina alettu ottaa enemmän käyttöön. Yhtenä mallina on vielä Stewart-Gough alustaan perustuva malli, joka on nimetty sen kehittäjien mukaan. Siinä on kuusi tankoa kiinnitetty alustaan, jossa työkalu on, pallonivelin. Tämän mallin käyttö teollisuudessa on erittäin vähäistä. Kuvassa 35 on vasemmalla rakenteeltaan kolmiomainen robottimalli Tricept ja oikealla Stewart-Gough robotti. /10, s.8-9./



**KUVA 35. Tricept-robotti ja Stewart-Gough robotti. /10, s.8-9/**

### **5.5 Nivelrobotit**

Nivelrobotti on kaikkein menestynein robottimalli. Siinä on jäljitely ihmisen kehoa ja tästä esimerkkinä on käsivarsirobotti. Nivelrobotteja on saatavilla paljon erikokoisina ja erikokoisille kuormille teollisuuden vaatimuksiin. Nivelrobotti pystyy kääntymään mihin asentoon tahansa siinä olevien monien pyörivän, sekä pystysuuntaisen liikkeen mahdollistavien nivelten ansiosta, joka tekee siitä monipuolisen ja joustavan. Nivelrobotteja voidaan käyttää kaikissa teollisuuden tehtävissä, joista esimerkkeinä tehtävät kappaleenkäsittelyssä, valimoissa, hitsauksessa ja maalauksessa. Kuvassa 36 on käsivarsirobotti. /10, s.9./



**KUVA 36. Käsivarsirobotti /10, s.9/**

## **6 TEOLLISUUSROBOTIN OHJELMOINTITAVAT**

Teollisuusrobotin ohjelmoinnilla tarkoitetaan sitä, että sille opetetaan ne tehtävät, mitä sen halutaan suorittavan, sekä muut sen toimintaominaisuudet. Robotin pääohjelmointitapoja on kolme erilaista:

- Ohjelmointi ohjelmointikielellä (tietokoneella online- tai offline-tilassa)
- Ohjelmointi opettamalla (online-tilassa)
- Ohjelmointi simulointiohjelmalla (offline-tilassa)

Online -tilalla tarkoitetaan tilaa, jolloin ohjelmoitaessa robottia se ei voi olla toiminnassa. Offline-tilalla tarkoitetaan ohjelmointia robotista erillisellä tietokoneella, ja sillä tehty ohjelma ladataan myöhemmin robotin kontrolleriin eli robotin logiikan sisältävään laitteistoon. Offline -ohjelmoinnin aikana robotti voi olla toiminnassa. /10, s.19./

### **6.1 Ohjelmointi ohjelmointikielellä**

Kaikilla teollisuusrobottien valmistajilla on robottejaan varten omat ohjelmointikielensä. Ne on joko mukautettu yleisestä C-kielestä tai sitten on luotu täysin omia ohjelmointikieliä. Kaikilla kielillä on kuitenkin samankaltaisuuksia. Offline-tilassa komennot kirjoitetaan tietokoneella joko manuaalisesti tai sitten haluttuja komentoja

voidaan valita valikosta ja arvoja muuttaa halutulla tavalla. Online-tilassa ohjelmoitaessa komennot voidaan valita pendantin valikoista, ja niitä voidaan muokata, kuten tietokoneella. /10, s.19-20./

## **6.2 Ohjelmointi opettamalla**

Opettamalla ohjelmoimalla robotin liikkeitä ohjataan pendantin avulla, joko näppäimiä painamalla tai ohjaus-sauvalla. Pendantti on liitetty robotin kontrolleriin. Kun robotti on ohjattu haluttuun paikkaan ja asentoon sekä silmämääräisesti on tarkastettu paikka, voidaan paikkatieto tallentaa pendantilla. Tämän jälkeen voidaan robotti ohjata toiseen paikkaan ja asentoon, sekä samalla tavalla tallentaa paikkatieto. Paikan ja asennon lisäksi määritellään robotin siirtymätapa paikkojen välillä. Siirtymät voidaan määritellä esimerkiksi lineaarisiksi tai kaareviksi. Myös robotin liikenopeudet ja halutut tarttujan tai muun työkalun toiminnot voidaan määritellä. Edellä mainitut askeleet tehdään niin monelle pisteelle, kuin halutaan ja kun robotin ohjaus laitetaan automaatti-tilaan, tekee robotti edellä määritellyt liikkeet paikkojen ja asentojen välillä automaattisesti. /10, s.21-22./

## **6.3 Ohjelmointi simulointiohjelmalla**

Simulointiohjelmalla ohjelmoitaessa ohjelmointi suoritetaan erillisellä tietokoneella offline-tilassa, jolloin robotti voi olla toiminnassa. Simulointiohjelmalla simuloidaan robottisolun ja kaikki sen sisältämät kappaleet 3-D mallinnuksena graafisesti. Simulointiohjelmassa on myös virtuaalinen pendantti, jolla voidaan simuloida oikean pendantin toiminnot tietokoneella. Simulointiohjelmassa määritellyt robotin liikkeet muutetaan automaattisesti ohjelmassa robotin ohjelmointikielelle. Kun halutun lainen ohjelma on valmis, voidaan se testata tietokoneella ja ladata tämän jälkeen robotin kontrolleriin. Ohjelmointi simulointiohjelmalla on turvallista, koska silloin ei tarvitse olla robotin lähellä. /10, s.22-24./



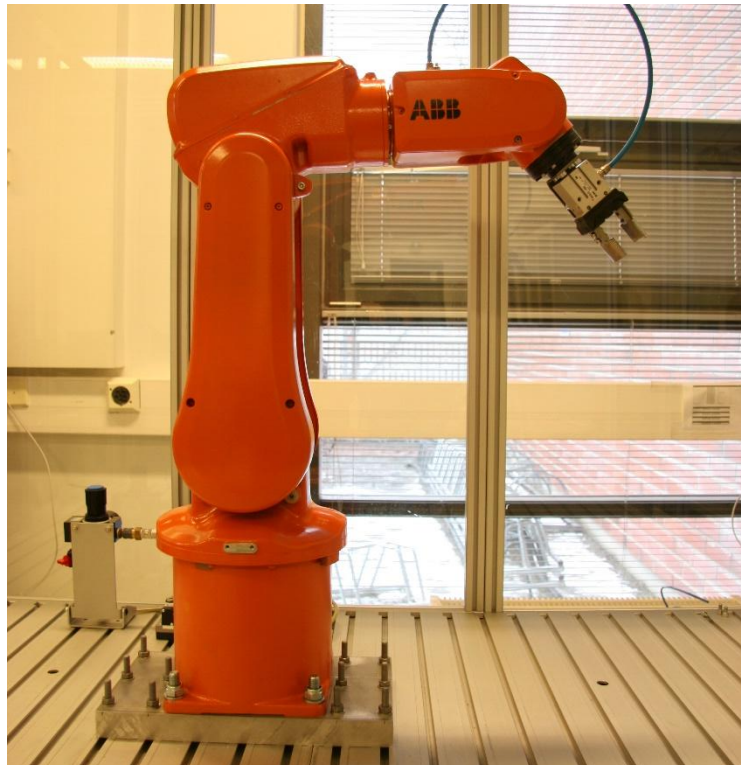
## **7 ROBOTTISOLUN RAKENTAMINEN JA ROBOTIN OHJELMOINTI**

Tämä luku käsittelee työn suoritusta, jossa käyn läpi asioita, joita teollisuusrobotti oppimisympäristön luomisessa on tehty. Vaiheita, joita työ sisältää, ovat robotin hankinta, robottisolun rakentaminen, robotin käyttöön, ohjelmointiin tutustuminen, sekä ohjeiden luominen käytöstä ja ohjelmoinnista robotin kanssa tulevaisuudessa työskenteleville.

### **7.1 Robotin hankinta ja kilpailutus**

Työ aloitettiin tekemällä kilpailutus eri robottivalmistajien välillä. Kilpailutus suoritettiin luomalla tarjouskysely eri robottivalmistajille. Tarjouskyselyssä määriteltiin robotille ominaisuudet, joita siltä halutaan, kuten työstövoima eli kuinka suurta massaa robotti pystyy käsittelemään, ulottuvuus, toistotarkkuus ja I/O tulojen ja lähtöjen määrät. Tarjouskyselyä tehtäessä robottien myyjiin oltiin yhteydessä sähköpostilla ja puhelimitse.

Kilpailutus tehtiin kolmen merkittävän teollisuusrobottivalmistajan, ABB:n, Fanucin ja Denson, robottien välillä. Kun kaikilta valmistajilta oli saatu tarjoukset, tehtiin päätös hankinnasta saatujen tarjousten välillä, mikä robottimalli hankitaan. Valintaan vaikuttavia tekijöitä olivat kilpailutuksessa määritetyt ominaisuudet, joita robotilta haetaan. Voittajaksi kilpailutuksessa nousi ABB:n robottimalli IRB120, joka oli muun muassa robottimalleista tarkin toistotarkkuudeltaan ja se pystyy paikoittamaan itsensä jopa millin sadasosan tarkkuudella. Kuvassa 37 on IRB120 teollisuusrobotti.



**KUVA 37. IRB120 -teollisuusrobotti, jossa työkaluna kiinni tarttuja**

## **7.2 Katsaus IRB120 -teollisuusrobottijärjestelmän ominaisuuksiin**

IRB120 on ABB:n pienin valmistuksessa oleva teollisuusrobottimalli. Se on kuusiakselinen, sillä on ulottuma 580 mm päähän sen alustan keskikohdasta, se pystyy käsittelemään kolmen kilon painoista työkalua ja se pystyy paikoittamaan itsensä millin sadasosan tarkkuudella. Robotin akselit luetaan alhaalta ylöspäin käsivarren päähän siten, että jokainen robotin kääntyvä kohta on yksi akseli. Robotin sisällä kulkee paineilmaletkut, joista saadaan tuotua paineilmaa työkaluja varten. Maksimi ilmanpaine, jonka robotille voi tuoda, on viisi baaria. Painemaliittimien vieressä on myös liitin, josta saadaan myös tarvittaessa sähköä työkalulle. Robottijärjestelmään kuuluu itse robotin lisäksi kontrolleri ja pendantti. Kontrolleri on robotin tietokone ja siinä on kaikki se elektroniikka, joka mahdollistaa robotin toiminnot. Kontrolleri muodostuu control module osasta ja drive module osasta. Control module sisältää robotin päätietokoneen, I/O piirilevyt ja muistin. Control module suorittaa kaikki robotin ohjelmistot, joita robottijärjestelmässä on, kuten törmäyksen valvonta. Drive module antaa robotin kuudella eri akselilla oleville moottoreille niiden tarvitseman sähkön. Kontrolleriin on kiinnitetty pendantti, joka on robotin ohjaukseen ja ohjelmointiin käytettävä opetusohjain. Sen kosketusnäytöllä voidaan navigoida koskettamalla sitä sormella

tai pendantin mukana olevalla kynällä. Pendantilla voidaan hallita kaikkia robotin toimintoja, kuten luoda robotille ohjelmia sekä muokata niitä.

### 7.3 Robottisolun rakentaminen

Rakensin robottisolun, jonka runko on tehty metallipalkeista ja rungon päälle on tehty pöytätaaso, jonka päälle robotti on pultattu kiinni. Pöytätaason ympärille on sijoitettu suojaseinät, jotka ympäröivät koko solua siten, että niillä estetään pääsy robotin työskentelyalueelle. Pöytätaason alapuolella on toinen taso, joka on valmistettu vanerista. Sen päälle on sijoitettu kontrolleri, ja siinä voidaan myös säilyttää muita robotin tarvikkeita. Koko robottisolusolu on tehty liikuteltavaksi ja siinä on pyörät jokaisessa kulmassa, jotka voidaan lukita liikkumisen estämiseksi. Pyörien lukkojen lisäksi solu voidaan tukea lattiaa vasten vielä kuudella säädettävällä tassulla. Robotin työkaluja, kuten tarttujaa varten, soluun on tuotu myös paineilma ja magneettiventtiileitä voidaan ohjata kontrollerin ulostuloilla päälle ja pois. Robottisolua rakensin oppilaitoksella ja rakentamisessa hyödynnettiin oppilaitokselta löytyviä rakennustarvikkeita.

Rungossa käytetyt kaikki metallipalkit ovat mitoiltaan 50mm x 50mm, paitsi kulmissa olevat metallipalkit ovat 100mm x 50mm. Paksuudeltaan metallipalkit ovat 4mm. Kokonaismitat rungolle ovat 1463mm x 1163mm. Koska runko on tehty tällaisesta metallipalkista, on sille saatu tarpeeksi paljon massaa, jotta se ei heilu ja pääse liikkumaan robotin tehdessä nopeita liikkeitä. Kuvassa 38 näkyy rungon rakenne.

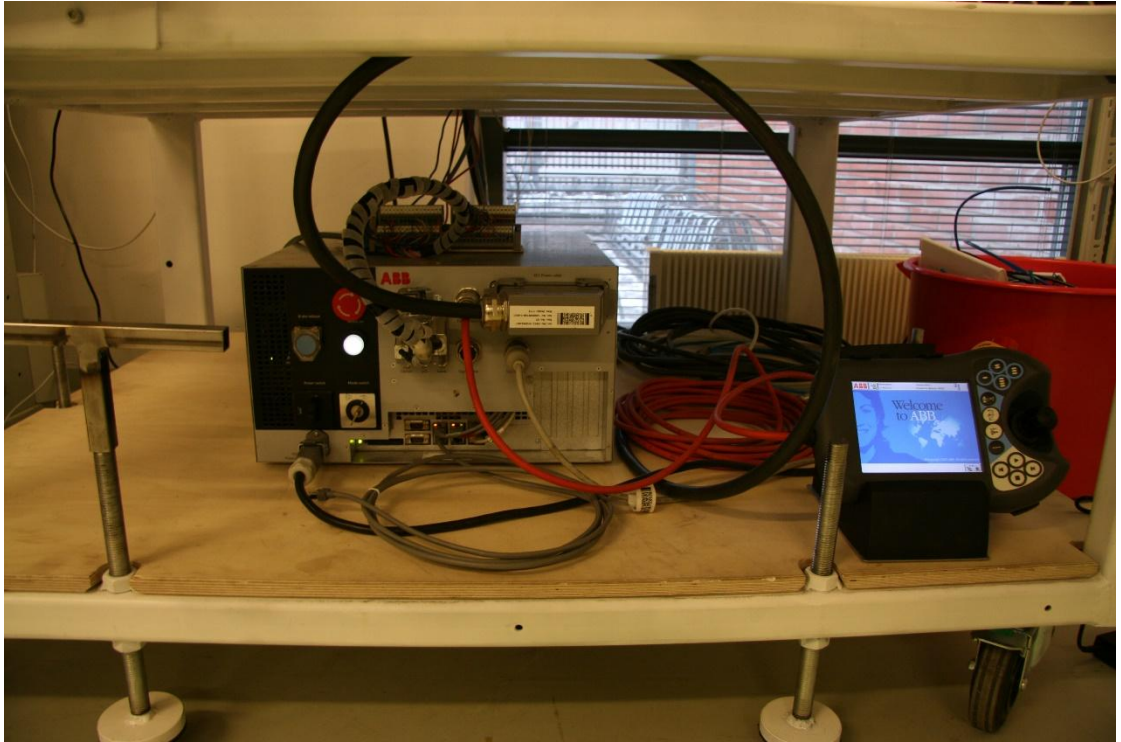


**KUVA 38. Robottisolun rungon rakenne**

Pöytätaaso, joka on rungon päällä, on Feston pöytätaaso. Näitä taasoja on laitettu kaksi vierekkäin. Yhden pöytäpalan koko on 1100mm x 700mm, joten kokonaisuudessaan pöydän koko on siis 1100mm x 1400mm. Pöytäpalat on kiinnitetty runkoon alapuolelta kulmarauodoilla ja suoraan päältä päin pulteilla. Pöytätaason alapuolella oleva taaso toimii kontrollerin ja muiden robotin tarvikkeiden säilytystilana, jolloin kaikki tarvikkeet kulkevat robottisolun mukana siirrettäessä sitä. Kuvassa 39 on pala Festo -pöytätaaso ja kuvassa 40 on alataaso, jossa voidaan säilyttää robotin tarvikkeita.



**KUVA 39. Festo -pöytäpala**

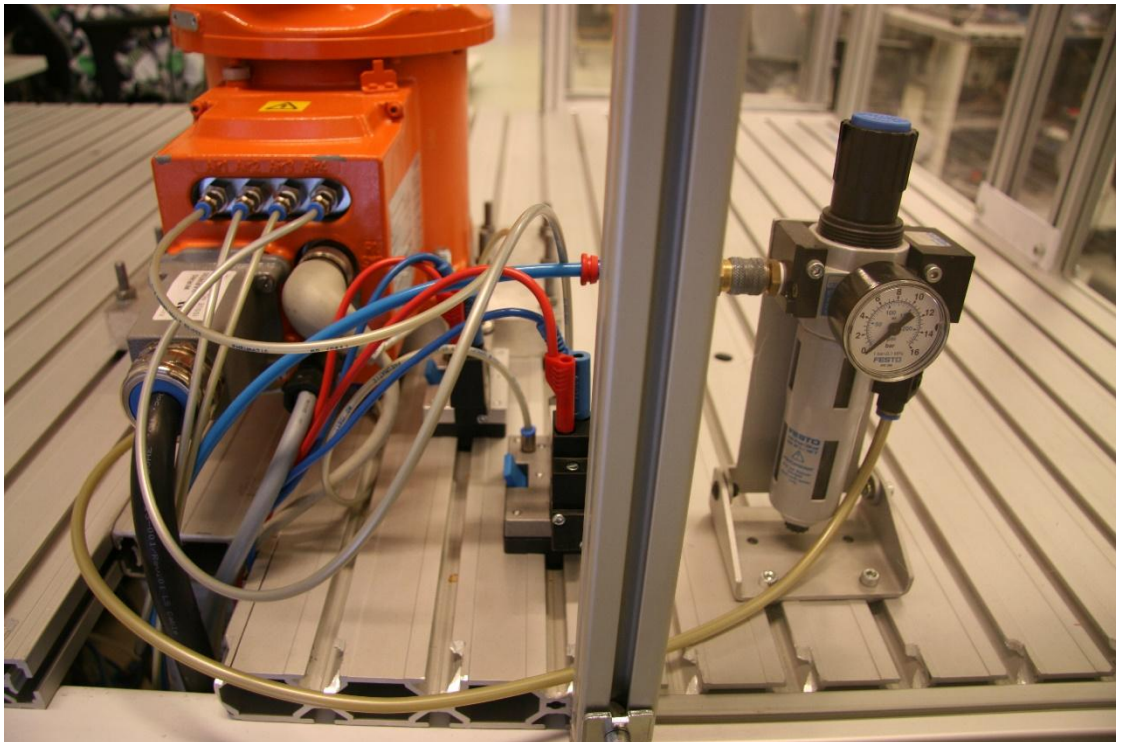


**KUVA 40. Alataso, jossa on vasemmalla robotin kontrolleri ja oikealla pendantti**

Pöytätasoa ympäröivät suojaseinät ovat joskus aiemmin oppilaitoksella olleen robotisolun vanhoja suojaseiniä, ja ne on otettu uusiokäyttöön. Korkeutta suojaseinillä on 1145mm. Etummainen suojaseinä taittuu auki kahdesta kohtaa, jolloin robotisolun sisälle on mahdollista päästä. Takana oleva suojaseinä avautuu keskeltä, josta pääsee käsiksi robotin liittimiin ja paineilmatarvikkeisiin. Paineilmatarvikkeet ovat Feston tarvikkeita, ja niiden kiinnitykset käyvät suoraan pöytätasoon, joten niitä on helppo irrottaa ja laittaa takaisin kiinni. Robottijärjestelmässä olevat kaksi magneettiventtiiliä on kytketty robotin ulostuloihin, jolloin niitä voidaan ohjata. Ulostulo viisi ohjaa magneettiventtiiliä, joka ohjaa ilman robotin ensimmäiseen paineilmaliihtimeen. Ulostulo kuusi ohjaa magneettiventtiiliä, joka ohjaa ilman robotin neljänteen paineilmaliihtimeen. Maksimi ilmanpaine, jonka robotille saa tuoda, on viisi baaria. Kuvassa 41 on suojaseinän osia ja kuvassa 42 näkyy robotin paineilmaliihtännät ja Feston paineilmatarvikkeita.



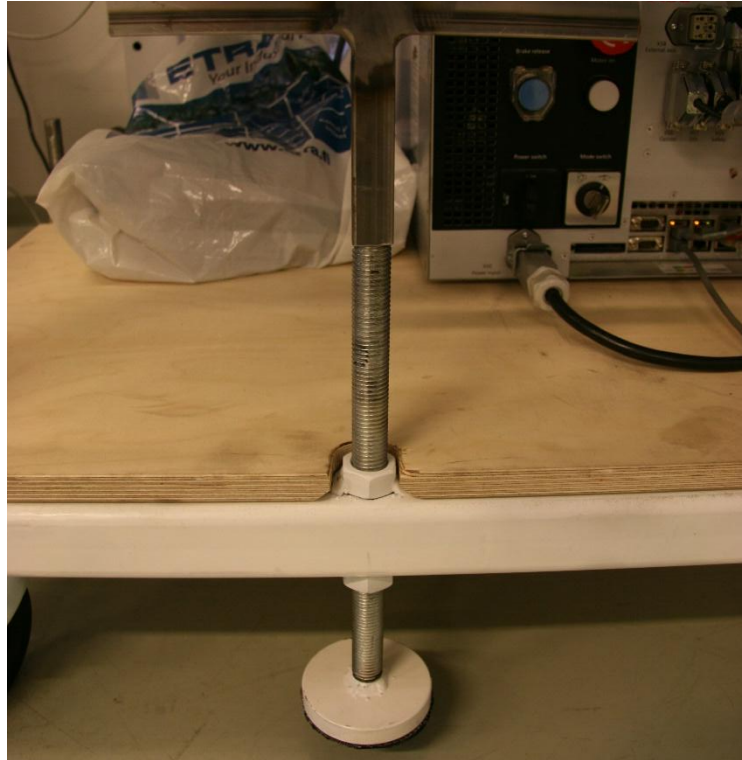
**KUVA 41. Suojaseinän osia**



**KUVA 42. Kaksi magneettiventtiiliä ja painemittari, josta voidaan säätää ilmapainetta**

Tuentatassuja on robottisolun etu- ja takaosissa kaksi kappaletta, ja molemmilla sivuilla on yhdet tassut. Tassujen korkeutta voidaan säätää tassuille tehdyllä säätötyöka-

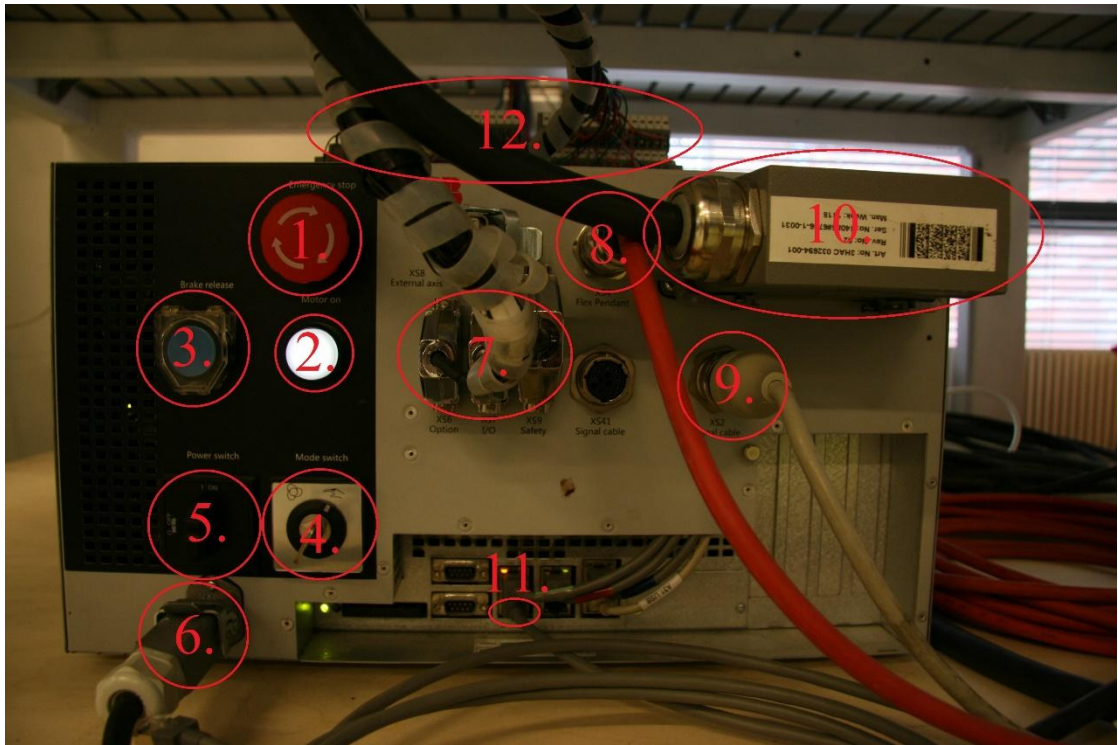
lulla. Tassujen pohjissa on vielä liimattuna kiinni tuennan pitoa parantavat matonpalat. Kuvassa 43 on tuentatassu, jossa on kiinni säätötyökalu.



**KUVA 43. Tuentatassu, jossa on kiinni säätötyökalu**

#### **7.4 Robottijärjestelmän käyttöönotto, kontrolleri ja pendantti**

Robottijärjestelmä saadaan toimintakuntoon, kun kontrolleriin on kytketty oikeat johdot oikeisiin liittimiin. Kun johdot ovat paikallaan ja kontrollerin virtakatkaisija on käännetty päälle asentoon, voidaan robottia alkaa liikuttamaan pendantin avulla. Käyn seuraavaksi läpi kontrollerin johtoliittimet ja painikkeet sekä pendantin painikkeet. Kerron myös painikkeista tapahtuvia toimintoja. Kuvassa 44 on kontrolleri, jonka eri kohdat on numeroitu ja selitetty. Kuvassa 45 on pendantti, jonka kohdat numeroitu. Numeroitujen kohtien selitykset löytyvät kuvien alapuolelta.

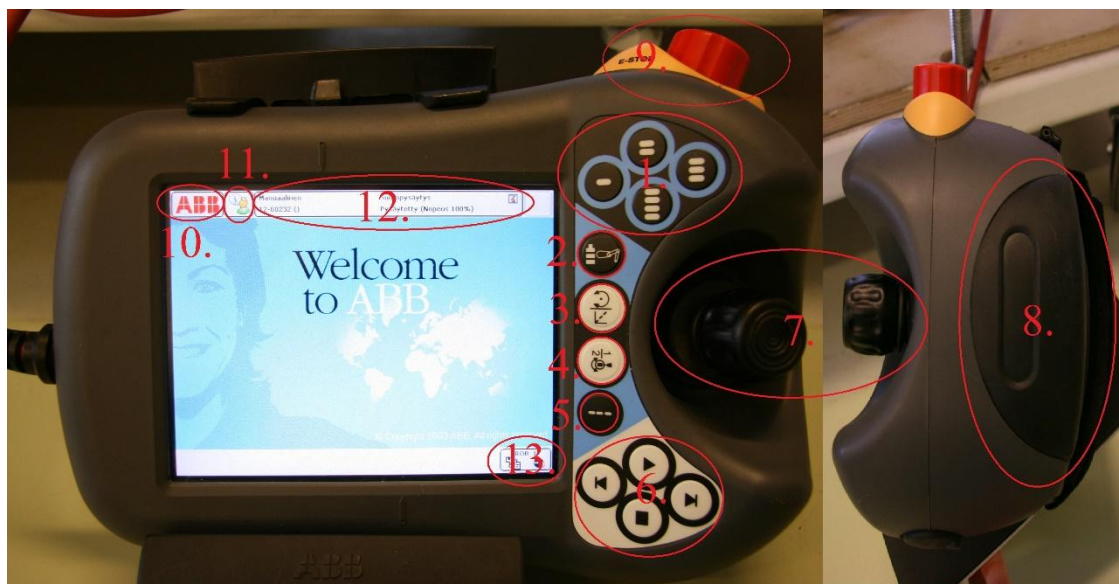


**KUVA 44. Kontrolleri**

1. Kontrollerin hätä-seis -painike: Painamalla hätä-seis -painiketta saadaan robotin liikkeet pysäytettyä.
2. Moottorit päälle -painike: Kun robotti on automaatti/tuotantotilassa, täytyy moottorit kytkeä päälle ennen, kuin robotti voi alkaa suorittamaan ohjelmaa.
3. Jarrujen vapautus -painike: Tällä painikkeella voidaan vapauttaa robotin aksleilla sijaisevien moottorien jarrut, jolloin robottia voidaan liikuttaa vapaasti. Kun painike pidetään pohjassa, ovat jarrut auki, ja kun painike vapautetaan, jarrut lukkiutuvat. Huomio: älä paina tätä painiketta, jos robottia ei ole tuettu.
4. Robotin tila -kytkin: Tämä kytkimen avulla voidaan vaihtaa robotti opetustilaan tai automaatti/tuotantotilaan. Oikealle käännettynä, jossa näkyy käden kuva, on robotti opetustilassa. Tällöin sitä voidaan liikuttaa pendantilla ja sille voidaan luoda ohjelmia. Vasemmalle käännettynä robotti on automaatti/tuotantotilassa. Tällöin robotti suorittaa siihen ajettua ohjelmaa.
5. Virta -kytkin: Virta kytkimellä kytketään kontrollerin virta päälle ja pois.



6. Kontrollerin virtaliitin: Tähän kytketään kontrollerin virtakaapeli.
7. XS6/XS7/XS9 liittimet: Näihin liittimiin kytketyt kaapelit ovat yhteydessä muun muassa jarrujen vapautukseen, I/O tuloihin ja lähtöihin, sekä hätä-seis-toimintoihin.
8. Pendantin liitin: Pendantti kytketään kiinni tähän liittimeen.
9. XS2 -signaalikaapeli: Tähän kytketään kontrollerin ja robotin välillä oleva signaalikaapeli.
10. Robotin virtakaapeli: Tähän kytketään kontrollerin ja robotin välillä oleva virtakaapeli.
11. Service -liitäntä: Tähän kytketään tietokoneesta ethernetkaapeli, jonka avulla voidaan siirtää ja luoda ohjelmia tietokoneen simulointiohjelmalla.
12. Riviliittimet: Riviliittimet I/O tuloille ja lähdöille, sekä ulkoisille pysäytyslaitteille. Sisääntulot 1-4 ja ulostulot 1-4 ovat käytössä työkaluja varten, jos niitä tarvitsee ohjata sähköllä tai niistä tarvitsee tuoda tietoa kontrollerille. Ulostulot viisi ja kuusi ohjaavat magneettiventtiileitä, joilla ohjataan paineilmaa robotin paineilimaliittimille yksi ja neljä. Sisääntuloon 16 on kytketty induktiivinen anturi, joka valvoo robottisolun ovea.



**KUVA 45. Pendantti**

1. Ohjelmoitavat käyttöpainikkeet: Näille painikkeille voidaan itse määrätä toiminnot.
2. Robotin valinta painike: Jos robottijärjestelmässä olisi useita eri robotteja, voitaisiin tällä painikkeella valita ohjattava robotti.
3. Liiketavan valinta -painike: Tällä painikkeella voidaan valita robotin liiketapa. Painettaessa painiketta voidaan valita tila, jolloin robotti pitää työkalun pään samassa asennossa liikuttaen sitä lineaarisesti. Toisella painaluksella tulee aktiiviseksi tila, jolloin työkalun asentoa voidaan muuttaa.
4. Liikuteltavien akselien valinta: Tällä painikkeella voidaan valita akselit, joita liikutetaan. Painettaessa painiketta voidaan valita akselit akseliryhmien 1-3 ja 4-6 välillä.
5. Askellustilan valinta: Painikkeella voidaan valita miten robotti liikkuu. Jos valittuna on jokin askellustiloista, robotti liikkuu sen askelluksen verran siihen suuntaan mihin ohjaus-sauvaa käännetään. Esimerkiksi pieni askellustila valittuna liike on 0,05mm. Jos ohjaus-sauvaa pidetään sekunnin ajan pohjassa, liikkuu robotti kymmenen kertaa valitun askellustilan verran, tässä ajassa. Kun mitään askellustilaa ei ole valittuna liikkuu robotti valitun nopeuden mukaisesti ohjaus-sauvan liikkeen mukaan, kuitenkin maksimissaan 250mm/s.
6. Ohjelman toimintopainikkeet: Näillä painikkeilla käynnistetään ohjelma ja pysäytetään se. Ohjelmassa voidaan myös hyppiä askel kerrallaan eteen tai taaksepäin.
7. Ohjaus-sauva: Ohjaus-sauvaa kääntämällä ja kiertämällä voidaan hallita robotin liikkeitä.
8. Turva -painike: Jotta robottia voidaan ohjata opetustilassa pitää turva -painike olla painettuna puoleen väliin. Jos sitä ei paina, tai se on painettuna kokonaan pohjaan, ei robotti liiku. Jos siitä päästetään irti, tai se painetaan pohjaan kesken robotin liikkeiden, pysähtyy robotti.

9. Häätä-seis -painike: Painettaessa painike pohjaan saadaan kaikki robotin liikkeet pysäytettyä.

10. ABB -valikko: ABB -valikon sisältä löytyvät kaikki robotin valikot, kuten ohjelmaeditori ohjelmien luomista varten ja ohjelmadata ohjelmatietojen muokkausta ja tarkastelua varten.

11. Operaattori-ikkuna: Tähän ikkunaan tulee näkyville kaikki viestit, joita ohjelma mahdollisesti voi sisältää. Tämä valikko toimii niin sanotusti kommunikointivälineenä robotin käyttäjän kanssa.

12. Tilarivi: Tästä näkyy robottijärjestelmän tila ja nimi. Tilarivistä painettaessa tulee näkyviin järjestelmän ilmoitukset.

13. Pikavalikko: Tästä valikosta päästään nopeasti robotin eri asetuksiin. Tässä valikossa voidaan valita esimerkiksi työkalu, työkohde ja robotin nopeus.

## **7.5 Järjestelmän sisään ja ulostulot**

Robotin kontrollerissa on 16 sisääntuloa ja 16 ulostuloa, joita voidaan käyttää laitteiden, kuten robotin työkalun ohjaukseen tai niitä voidaan käyttää robotin ohjaukseen ulkoisesti. Robotin ulostuloihin viisi ja kuusi on kytketty magneettiventtiilit, joilla voidaan ohjata robotin työkaluja. Ulostulo viisi ohjaa paineilman robotin ilmaliittimeen numero 1 ja ulostulo kuusi ohjaa paineilman ilmaliittimeen numero 4. Sisääntuloon 16 on kytketty induktiivinen anturi, joka valvoo robottisolun ovea. Kun robotti suorittaa automaattitilassa jotakin ohjelmaa ja ovi avataan, pysähtyy robotin toiminta. Robotti ei jatka liikkeitään oven ollessa auki, vaikka painettaisiin pendantin käynnistä painiketta. Kun ovi on suljettu ja painetaan pendantin käynnistä painiketta, jatkaa robotti ohjelmaa siitä, mihin se pysähtyi oven auetessa.

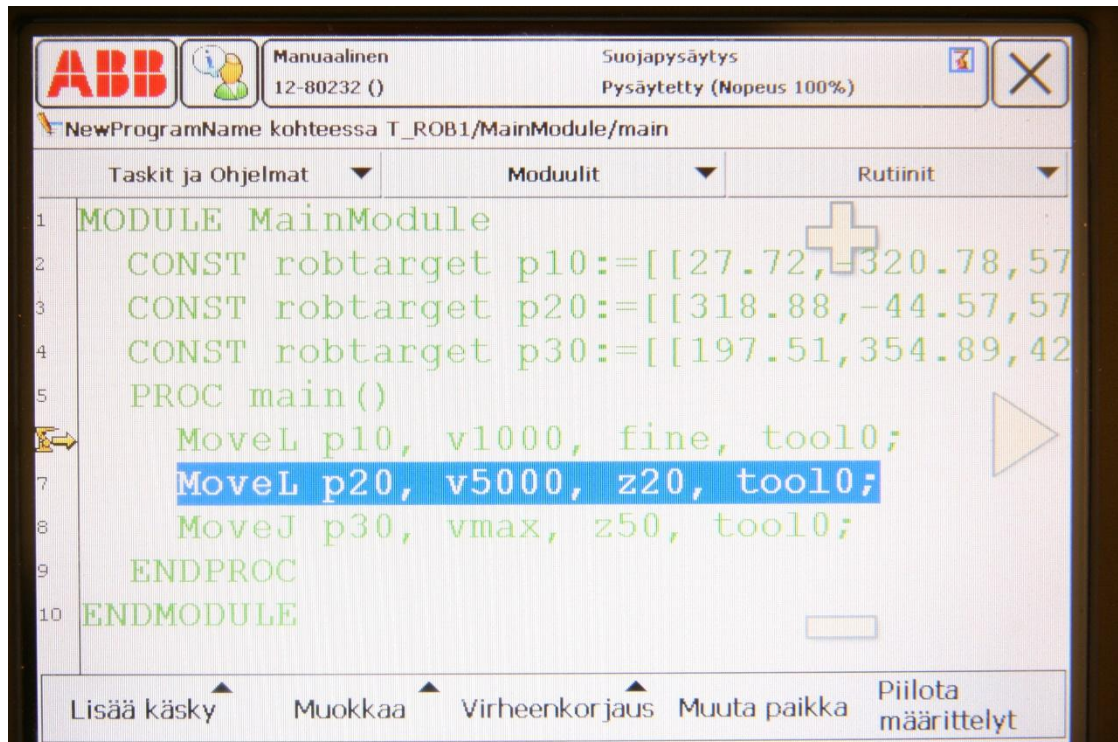
Sisääntulon 16 on toimintaa ja määrittelyä on muutettu ABB -valikon ohjauspaneelin konfiguraatiosta. Konfiguraatio valikon system input valikkoon on lisätty sisääntulo 16 toiminnolla stop. Sisääntulon 16 määrittelyä on myös muutettu konfiguraation signal valikosta. Signal valikosta on muutettu sisääntulo 16:sta invert physical valuen arvoksi yes. Vakiona tämä arvo olisi no.

## 7.6 RAPID-ohjelmointikieli ja RAPID-ohjelman rakenne

RAPID-ohjelmointikieli koostuu englanninkielisiä käskyistä, joilla ohjataan robotin liikkeitä ja suoritetaan muita ohjelman toimintoja, kuten asetetaan digitaalinen ulostulo päälle tai siirrytään ohjelmaa suoritettaessa rutiinista toiseen. Ohjelma rakentuu moduuleista, joiden sisällä on rutiineja, joihin itse ohjelman käskyt on kirjoitettu. Moduuleita voi olla ohjelmassa useita ja yksi moduuli voi sisältää useita rutiineja. Kun suurempi ohjelmakokonaisuus sisältää useita moduuleita, tulee ohjelman luettavuudesta ja rakenteesta helpommin ymmärrettävä, kun eri robotin toiminnot voidaan tehdä eri moduuleihin. Ohjelman moduuli ja rutiinirakenne voidaan ajatella esimerkiksi seuraavalla tavalla. Ajatellaan, että vasen käsi on moduuli yksi ja oikea käsi on moduuli kaksi. Näiden moduulien sisällä on rutiineja, joita ovat molemmissa käsissä olevat viisi sormeä, eli kummassakin moduulissa on viisi rutiinia. Ohjelman aikana voidaan sitten siirtyä näissä kahdessa moduulissa olevien rutiinien välillä. Ohjelmat voivat sisältää ohjelmamoduuleja, joihin on kirjoitettu robotin ohjelmat, ja järjestelmämoduuleja, joissa on tallennettuna tietoja, jotka mahdollistavat toimintojen suorittamista. Esimerkiksi user -niminen järjestelmämoduuli, joka on aina ohjelmassa mukana, sisältää viisi numeerisen datan rekisteriä, joissa on ennalta määritettyjä muuttujia, joita voidaan käyttää ohjelmoinnissa hyödyksi.

## 7.7 Moduulit ja rutiinit

Kuvassa 46 on ohjelman rakennekuva ohjelmaeditorista pendantin näytöltä. Kuvan jälkeen on kuvassa näkyviä kohteita selitetty.



**KUVA 46. Ohjelman rakenne ohjelmaeditorissa**

Ohjelmaeditorissa vasemmassa laidassa näkyvät tekstirivien rivinumerot. Ohjelmassa kaikista ylimpänä rivillä yksi näkyy moduulin nimi sanan MODULE perässä. Tässä tapauksessa se on nimeltään MainModule, ja se on moduuli, jossa ollaan työskentelemässä. Tämä moduuli luodaan aina tehtäessä uusi ohjelma. MODULE -tekstin alapuolella on tietonäkymä, johon tulee tietoja näkyviin ohjelmassa käytetyistä käskyistä ja toiminnoista. Tässä tiedot ovat robotin eri paikkojen p10, p20 ja p30 tiedot. CONST tarkoittaa, että arvo on vakio ja robtarget tarkoittaa, että se on datatyypiltään robotin paikkatietoja. P10, p20 ja p30 ovat paikkatiedoille määritellyt nimet. Paikkamerkintöjen perässä näkyy tarkemmat tiedot paikan sijainnista, joka on robotissa käytettävän työkalun keskipisteen sijainti, z, y, ja z koordinaateissa robotin alustan keskipisteestä. Tietonäkymä saadaan pois näkyvistä oikeassa alakulmasta olevasta piilota määrittelyt painikkeesta. Rivillä yksi oleva MODULE -teksti ja sen perässä oleva moduulin nimi kertoo kohdan, mistä kyseinen moduuli alkaa. Rivillä kymmenen oleva teksti ENDMODULE näyttää kohdan, johon moduuli loppuu. Moduuli loppuu aina tekstiin ENDMODULE.

Tietonäkymän alapuolella rivillä viisi oleva teksti PROC main() kertoo rutiinin alkamiskohdan ja rutiinin nimen. PROC on lyhenne englanninkielisestä sanasta procedure ja rutiinin nimi on tässä tapauksessa sanan PROC perässä oleva main. Tämä on pääru-

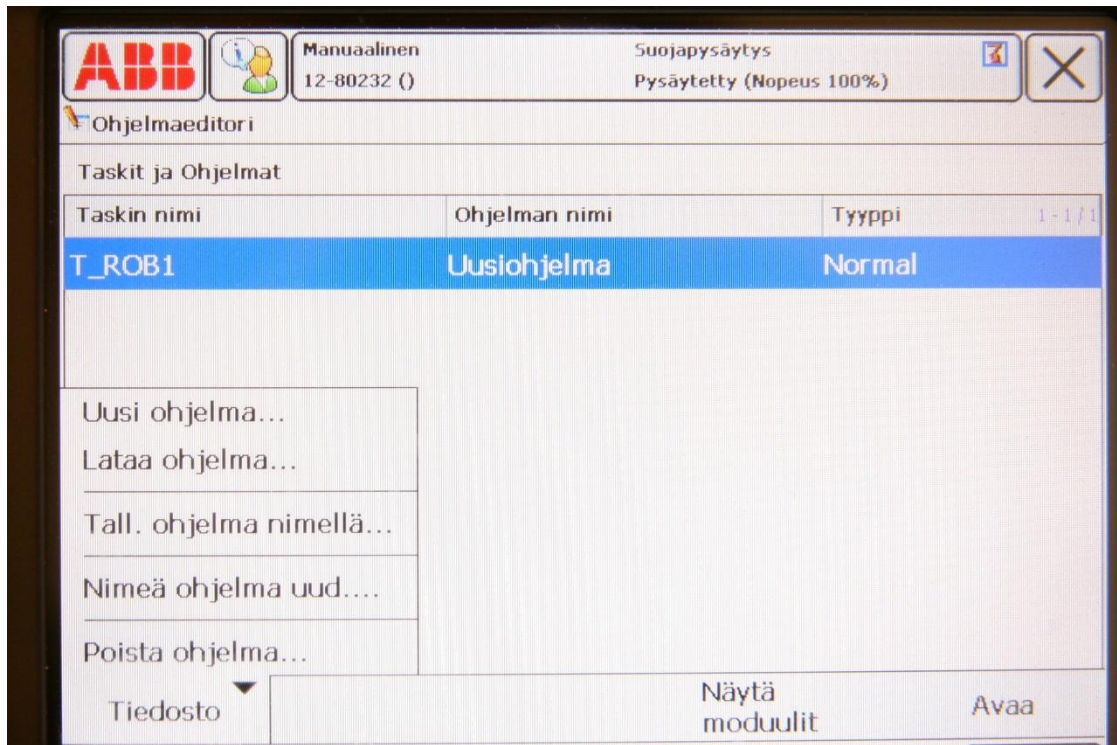
tiini, joka luodaan aina tehtäessä uusi ohjelma. Ohjelmassa on oltava aina päärutiini main, jotta ohjelmaa voidaan ajaa automaattitilassa. Rivillä yhdeksän oleva teksti ENDPROC näyttää kohdan, johon rutiini loppuu. Rutiini loppuu aina tekstiin ENDPROC. Robotin ohjelman käskyt kirjoitetaan rutiinin alun ja lopun välissä olevaan tilaan. Ohjelma on tässä esimerkissä riviltä kuusi riville kahdeksan. Jos moduulissa on useita rutiineja, tulevat ne edellisen rutiinin perään jatkoksi. Ne alkavat edellä olevan rutiinin lopun ENDPROC jälkeen seuraavalta riviltä. Toinen rutiini alkaa edellisen tavoin tekstillä PROC ”rutiinin nimi”. Jos painiketta piilota määrittelyt on painettu, näky näytössä ainoastaan valittuna oleva rutiini.

## **7.8 Ohjelmaeditori**

Seuraavissa kappaleissa käsittelen pendantin valikkoa ohjelmaeditori ja sen sisältämiä valikoita, sekä toimintoja. Ohjelmaeditoriin pääsee avaamalla vasemmassa ylälaudassa olevan ABB -valikon ja valitsemalla sieltä ohjelmaeditori

### **7.8.1 Taskit ja ohjelmat -ikkuna**

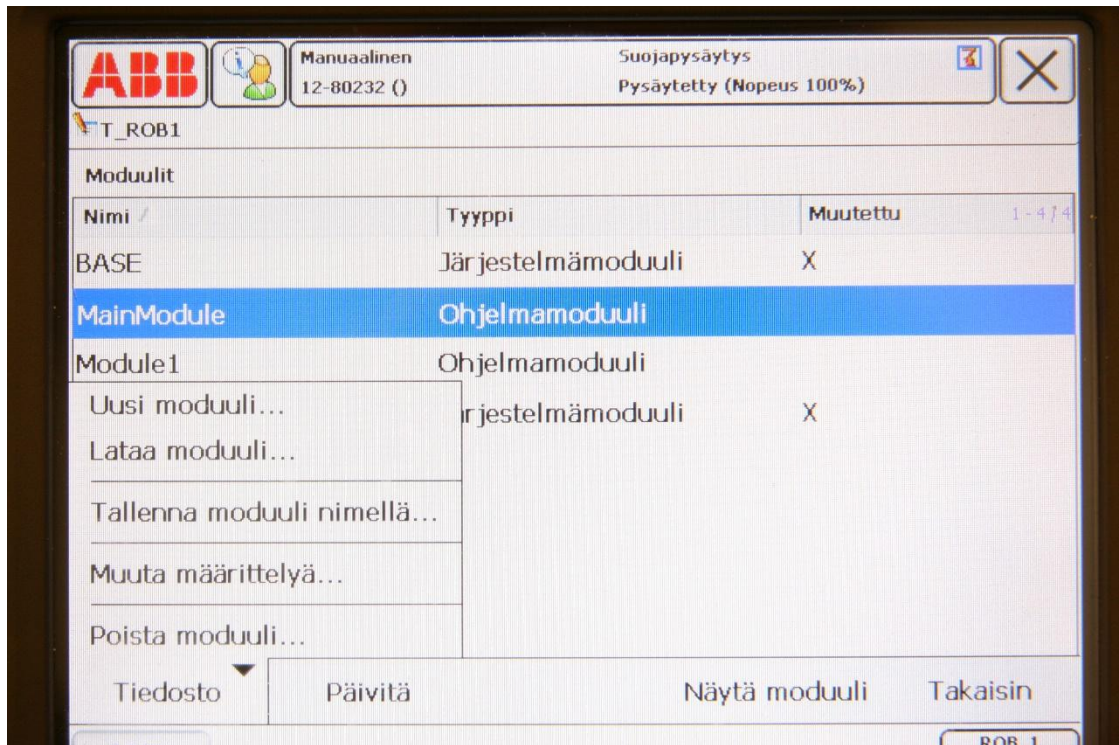
Ohjelmaeditorin yläpalkissa sijaitsevasta taskit ja ohjelmat -painikkeesta avautuu ikkuna, jossa näkyvät taskit eli käytössä olevat robotit ja sen käytössä olevan ohjelman nimi. Tässä robottijärjestelmässä taskin nimi on T\_ROB1 ja, koska useampia robotteja ei ole, ei muita taskeja näy. Taskit ja ohjelmat ikkunan vasemmassa alalaidassa olevasta tiedosto painikkeesta avautuu valikko, josta voidaan luoda uusi tyhjä ohjelma, ladata toinen ohjelma robotin käyttöön ja tallentaa ohjelma järjestelmän muistiin. Kuvassa 47 on taskit ja ohjelmat -ikkuna.



**KUVA 47. Taskit ja ohjelmat -ikkuna**

### 7.8.2 Moduulit -ikkuna

Ohjelmaeditorin yläpalkissa olevasta moduulit -painikkeesta avautuu ikkuna, jossa näkyvät ohjelmassa olevat moduulit. Vasemmassa alalaidassa olevasta tiedosto -painikkeesta avautuu valikko, josta voidaan luoda uusi moduuli, ladata ohjelmaan järjestelmään jo aiemmin tallennettu moduuli, poistaa moduuleita ja tallentaa moduuleita järjestelmän muistiin. Näytä moduuli -painikkeella voidaan avata valittuna oleva moduuli. Kuvassa 48 on moduulit -ikkuna.



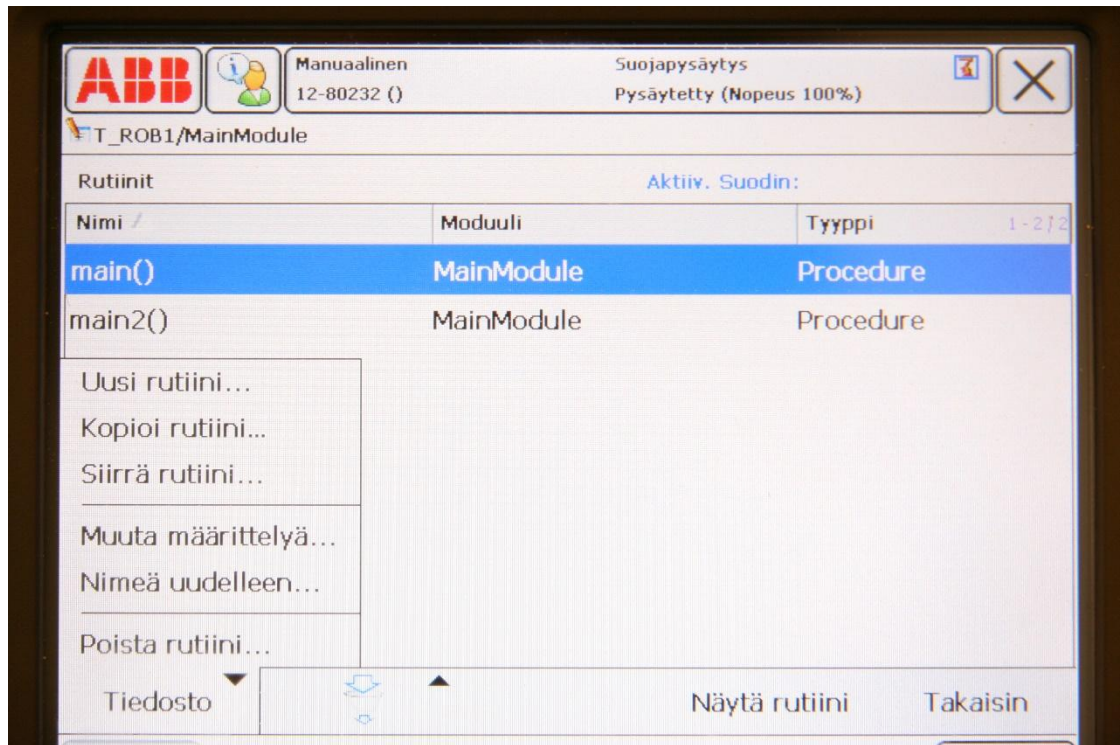
**KUVA 48. Moduulit -ikkuna**

Kun luodaan uusi moduuli, avautuu ikkuna, josta voidaan kirjoittaa moduulille haluttu nimi ja sille voidaan valita tyyppi. Tyypiksi valitaan program. Kun nämä asetukset ovat kunnossa, painetaan ok -painiketta, jolloin uusi moduuli ilmestyy moduulit valikkoon.

### 7.8.3 Rutiinit -ikkuna

Ohjelmaeditorin yläpalkissa olevasta rutiinit -painikkeesta avautuu ikkuna, jossa näkyy valittuna olevan moduulin rutiinit. Vasemmassa alalaidassa olevasta tiedosto -painikkeesta avautuu valikko, josta voidaan luoda uusi rutiini, nimetä rutiini ja poistaa rutiineita. Näytä rutiini -painikkeella voidaan avata valittuna oleva rutiini. Kuvassa 49 on rutiinit -ikkuna.



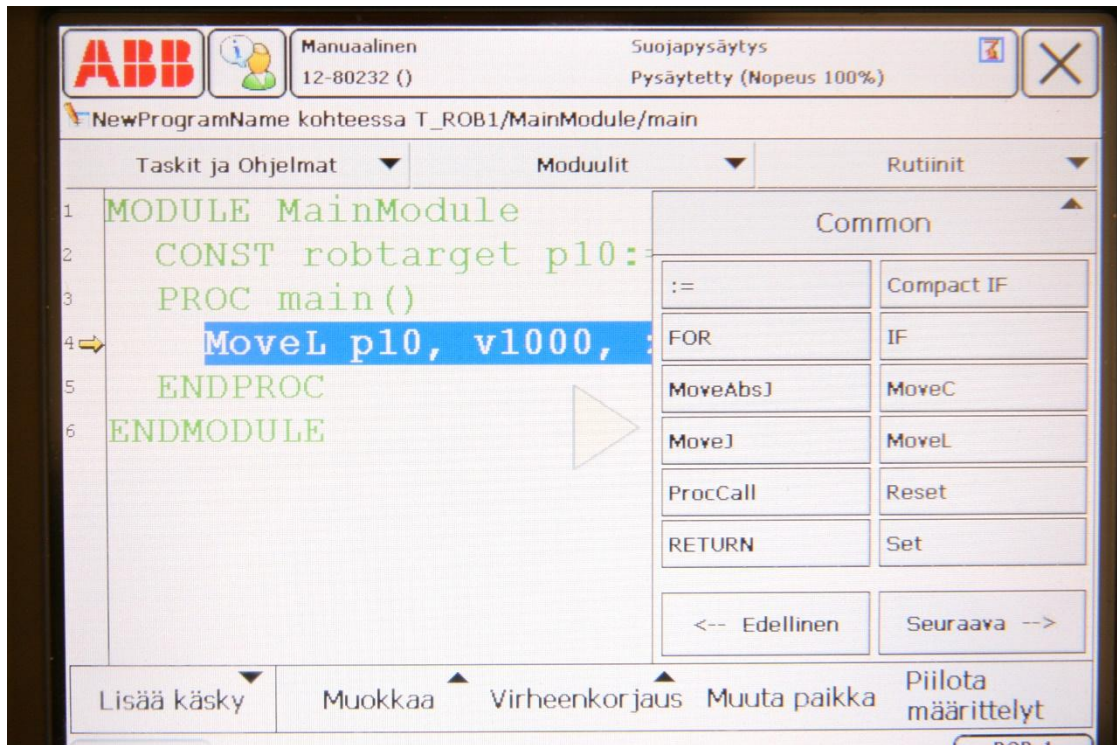


**KUVA 49. Rutiinit -ikkuna**

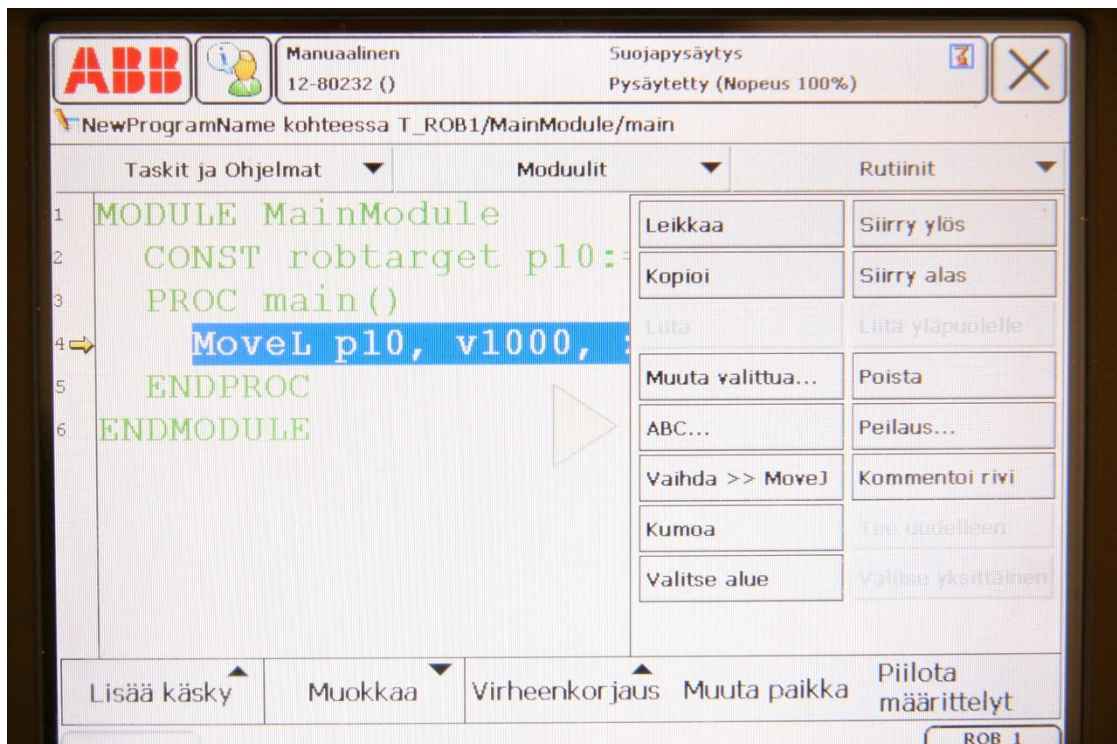
Kun luodaan uusi rutiini, avautuu ikkuna, jossa voidaan kirjoittaa rutiinille haluttu nimi ja valita rutiinin tyyppi. Rutiinin tyyppiksi valitaan proseduuriksi. Moduuli kohtaan valitaan, mihin moduuliin rutiini tehdään. Tämän jälkeen painetaan ok painiketta, jonka jälkeen luotu rutiini ilmestyy valittuna olevan moduulin rutiinit valikkoon.

#### 7.8.4 Lisää käsky- ja muokkaa -valikko

Lisää käsky -valikosta aukeaa lista robotin ohjelmointiin käytettävistä käskyistä ja toiminnoista. Muokkaa -valikosta aukeaa valinnat, joilla lisättyjä käskyjä voidaan muun muassa muokata, poistaa, kopioida, leikata, liittää ja kumota. Lisää käsky -valikon yleisimpiä tarvittavia käskyjä ja niiden ominaisuuksia käsitellen edempänä ohjelmointiohjeessa. Kuvassa 50 on avattuna lisää käsky -valikko auki ja kuvassa 51 on muokkaa -valikko avattuna.



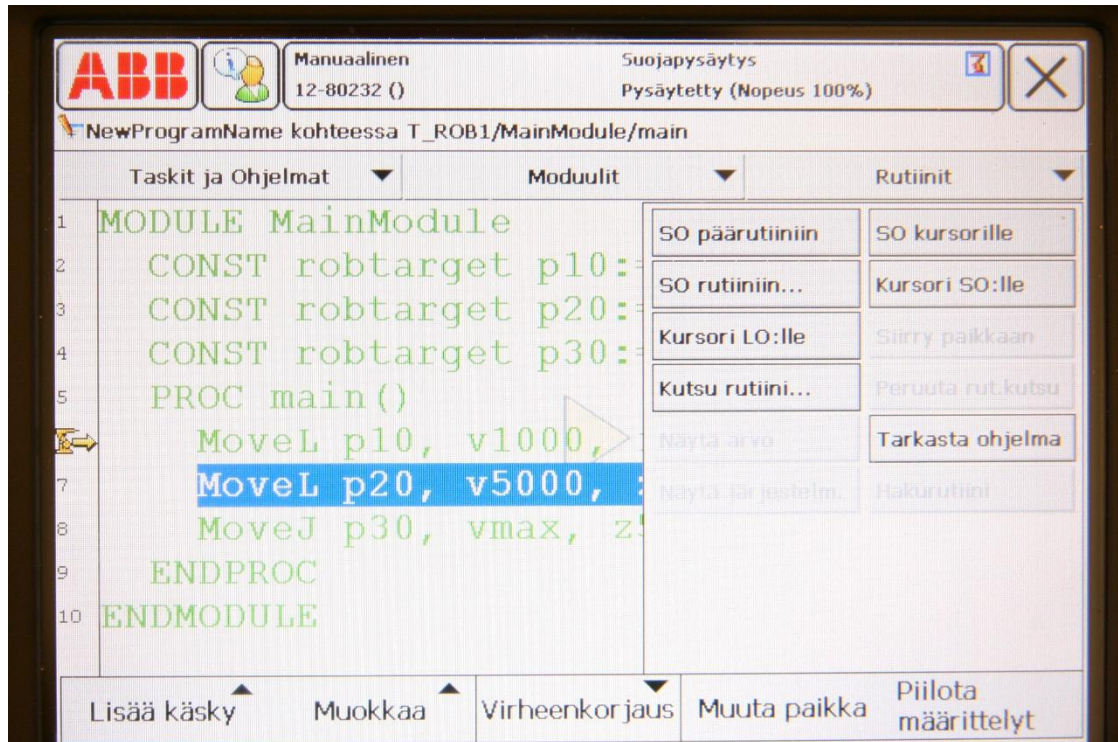
KUVA 50. Lisää käsky -valikko



KUVA 51. Muokkaa -valikko

### 7.8.5 Virheenkorjaus -valikko

Virheenkoroausvalikosta saadaan auki valinnat, joilla voidaan ohjelman osoittimen, liikeosoittimen ja kursorin paikkaa liikuttaa. Kuvassa 52 on avattuna ohjelmaeditorin valikko virheenkoroaus.



**KUVA 52. Virheenkoroaus -valikko ohjelmaeditorissa**

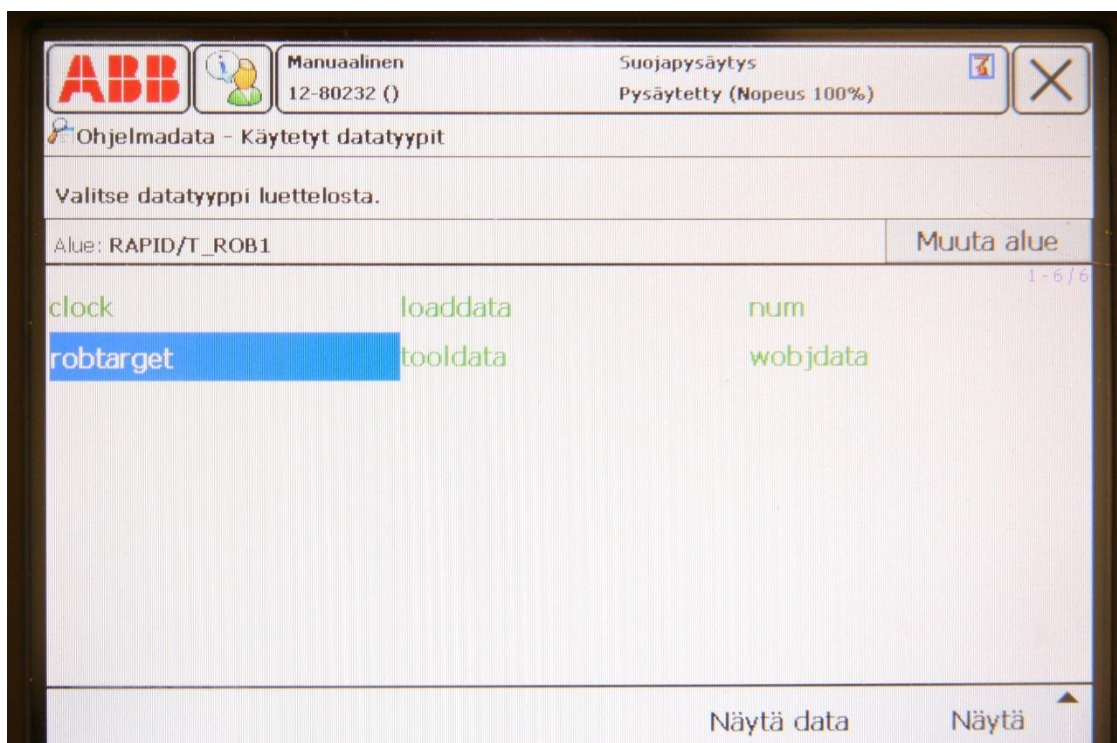
Vasemmalla olevien rivinumeroiden päällä näkyvät ohjelman osoitin ja liikeosoitin. Ohjelman osoitin on nuoli ja liikeosoitin on robotin kuva. Kohdistin on sininen korostusväri ohjelmakoodin päällä. Ohjelman osoitin ja liikeosoitin tulevat näkyviin, kun ohjelma käynnistetään. Ohjelman osoitin näyttää mistä kohtaa ohjelman suoritus alkaa ja ohjelman kulkiessa se näyttää, missä kohtaa kontrolleri suorittaa käskyjä. Ohjelman suoritin on yleensä yhden käskyn verran edellä liikeosoitinta. Liikeosoitin ilmaisee kohdan, jota roboti parhaillaan suorittaa. Kohdistin osoittaa koodiriviltä tehdyt valinnat.

SO päärutiniin siirtää ohjelman osoittimen päärutiniin ensimmäiselle käskyriville. SO kursorille siirtää ohjelman osoittimen kohdistimella korostetun rivin kohdalle. Kursori SO:lle toimii vastakkain siirtäen kursorin ohjelman osoittimen kohdalle. Jos ohjelmassa on useita moduuleita, joissa kaikissa on rutiineja tai jos yhdessä moduulissa on useita rutiineja, voidaan ohjelma osoitin siirtää halutun moduulin rutiinin ensimmäiselle käskyriville painamalla painiketta SO rutiiniin. Kursori LO:lle siirtää kohdisti-

men liikeosoittimen kohdalle. Kutsu rutiini painikkeella voidaan valita suoritettava huoltorutiini. Huoltorutiini on suoritettava esimerkiksi silloin, kun määritetään robotille uusi työkalu. Tarkasta ohjelma painikkeella voidaan tarkistaa ohjelma virheiden varalta.

## 7.9 Ohjelmadata, hotedit ja tuotantoikkuna

ABB -valikosta löytyvä ohjelmadata valikko sisältää kaikki ohjelman sisältämien käskyjen ja toimintojen tiedot. Ohjelmadata valikosta voidaan näitä tietoja muokata ja poistaa. Esimerkiksi jos robotille luodaan liikerata, tallentuvat robotin paikkakäskyjen tiedot ohjelmadata valikon rotarget datatyyppiin. Jos ohjelmaeditorissa muokataan jotain paikkakäskyä, muuttuvat käskyn tiedot myös ohjelmadata valikossa. Jos paikkakäsky poistetaan ohjelmaeditorissa, ei paikkatiedon määrittäminen poistu ohjelmadata valikosta, vaan se jää sinne. Tämä vaikuttaa uusien paikkakäskyjen lisäessä siten, että poistetun paikan numero jää välistä pois. Jos ohjelmaeditorissa poistettu paikkakäsky halutaan poistaa kokonaan, täytyy se poistaa myös ohjelmadatasta. Kun paikan tiedot on poistettu ohjelmadatasta, jatkuu ohjelmaeditorissa uusien paikkakäskyjen numerointi järjestyksessä. Ohjelmadata valikko toimii muillekin ohjelman toiminnoille samalla tavalla ja niiden tietoihin pääsee käsiksi ohjelmadata -valikon kautta. Kuvassa 53 on ohjelmadata -valikko, jossa on luettelo ohjelman datatyypeistä.



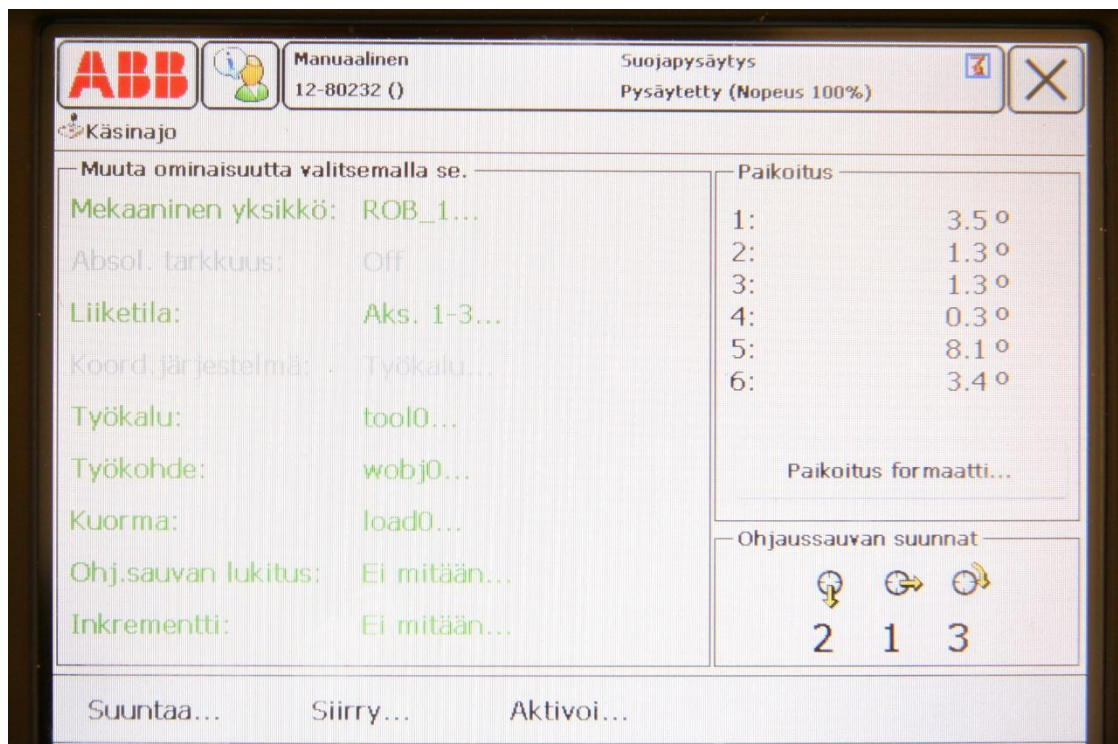
### KUVA 53. Ohjelmadata -valikko

ABB -valikosta löytyvän hotedit valikon avulla voidaan hienosäätää ohjelman paikkakäskeyjen paikoituksia. Tuotantoikkuna on tarkoitettu robotille luodun ohjelman kulun seurantaan. Kun järjestelmä laitetaan automaattiseen tilaan, aukeaa tuotantoikkuna automaattisesti ja siinä näkyy robotille luotu ohjelma. Kun ohjelma käynnistetään, liikkuvat ohjelman osoitin ja liikeosoitin ohjelman mukaisesti tuotantoikkunan vasemmassa laidassa.

### 7.10 Käsin ajo -valikko

Käsin ajo -valikosta nähdään robotin akseleiden paikoitus asteina. Kun robotin valittuna liiketapana on lineaarinen tai työkalun asennon muuttaminen, näkyy paikoitus työkalun keskipisteen paikoituksena millimetreinä robotin alustan keskikohdasta x, y ja z koordinaateissa. Tässä valikossa voidaan tehdä samanlaisia valintoja, kuten pika-valikossa ja pendantin painikkeilla. Uudet työkalun, työkohteen ja kuorman määrittämiset suoritetaan käsin ajo -valikossa.

Kuvassa 54 on käsin ajo -valikko.

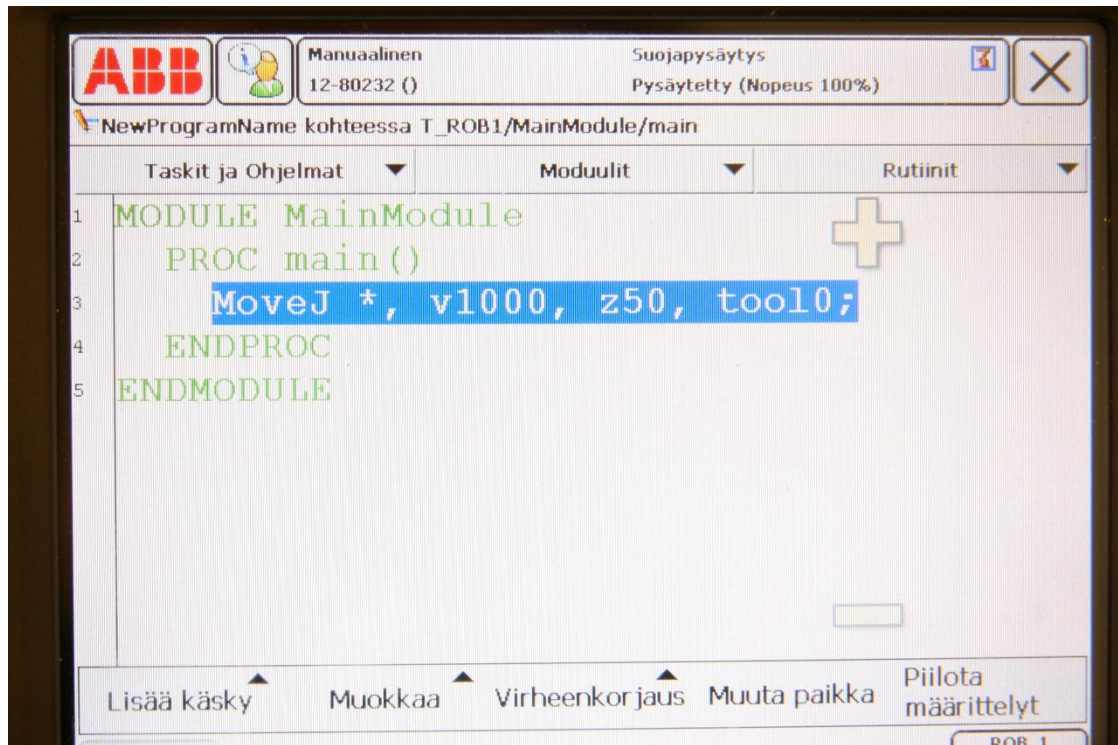


KUVA 54. Käsin ajo -valikko

## 7.11 Robotin ohjelmointi

Robotin ohjelmointi aloitetaan menemällä ABB -valikkoon ja valitsemalla sieltä ohjelmaeditori. Kun ohjelmaeditori on auennut, valitaan taskit ja ohjelmat valikko. Sieltä avataan tiedosto -valikko ja uusi ohjelma. Tämän jälkeen ohjelmaeditoriin aukeaa tyhjä ohjelma, jossa ohjelmamoduulina on automaattisesti MainModule. MainModulen rutiinina on automaattisesti main rutiini. Tämä rutiini on ohjelman päärutiini. Tämä tarkoittaa sitä, että, jos virheenkorjaus -valikosta valitaan SO päärutiiniin, siirtyy ohjelman osoitin tämän main rutiinin ensimmäiselle käskyriville, vaikka ohjelmassa olisi muitakin rutiineja. Main rutiinin voidaan ajatella olevan myös se kohta, josta luotu ohjelma alkaa. Jos jostakin toisesta rutiinista halutaan tehdä ohjelman päärutiini, onnistuu se muuttamalla rutiinin nimi main rutiiniksi. Ohjelmassa ei voi olla kahta saman-nimistä rutiinia.

Käskyjen lisääminen aloitetaan valitsemalla lisää käsky -valikko. Jotta halutun käskyn voi lisätä, täytyy PROC main() ja ENDPROC rivien välissä oleva teksti <SMT> olla korostettuna. Ennen käskyn lisäämistä liikutetaan robotti haluttuun paikkaan ja asentoon, johon määritetään ensimmäinen liikekäsky. Opetus- eli ohjelmointitilassa robotti saadaan liikkumaan painamalla pendantin päässä oleva turvakytin ensin puoleen väliin ja pitämällä se siinä. Kun turvakytin on painettuna puoleen väliin, voidaan robotin liikkeitä hallita ohjaus-sauvalla. Kun robotti on halutussa paikassa ja asennossa valitaan lisää käsky -valikosta MoveJ. Tämän jälkeen tekstin <SMT> tilalle ilmestyy valittu käsky, kuten kuvassa 55 näkyy.



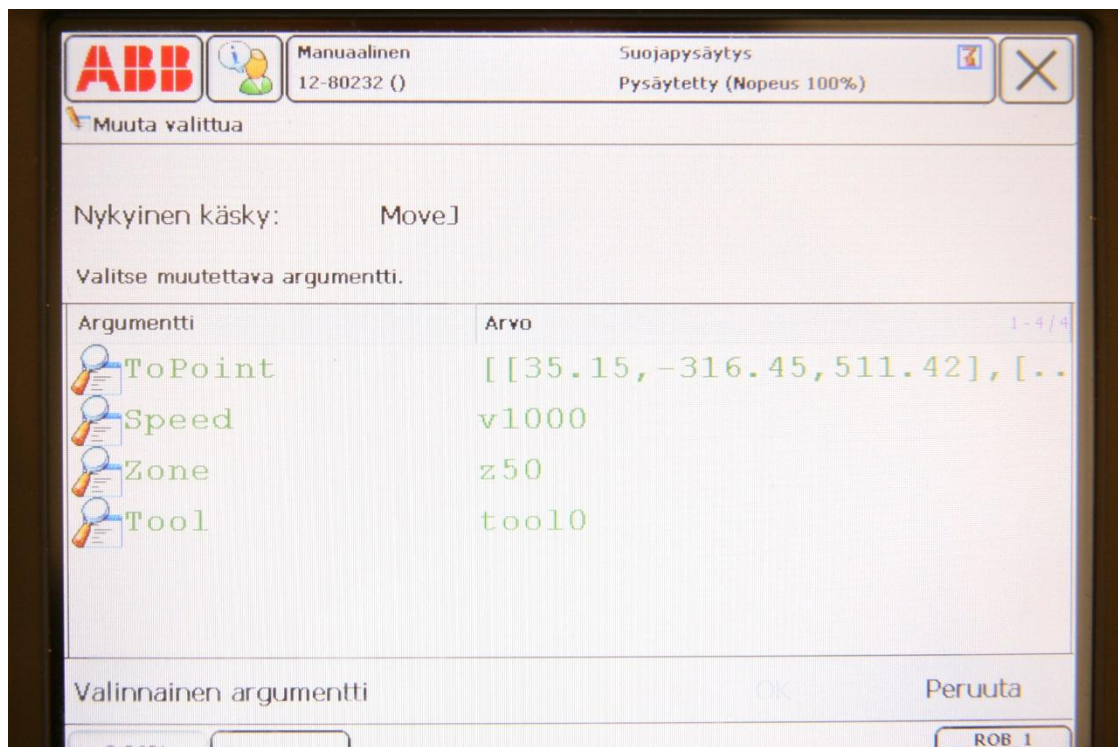
**KUVA 55. Ensimmäisen käskyn lisääminen**

Lisätty käsky MoveJ tarkoittaa liikettä, jossa robotti pystyy kääntämään kaikkia aksleita siten, että määrättyyn pisteeseen päästään. MoveL sen sijaan tarkoittaa liikettä, jossa pisteeseen siirrytään lineaarisesti eli suoraa viivaa pitkin. Jos halutaan siirtyä robotin toiselta puolelta toiselle, ei voida käyttää käskyä MoveL, koska robotti ei voi kulkea itsensä läpi. Tällöin täytyy käyttää käskyä MoveJ. Tämä ensimmäinen lisätty piste on se piste, josta ohjelman suoritus alkaa ja johon palataan ohjelman kierron päättyessä, jos ohjelman suoritustilana on valittuna jatkuva kierto. Kun ensimmäiseen pisteeseen on palattu, jatketaan siitä eteenpäin seuraavalle määritetylle paikalle. Jos valittuna on yksi kierto, jää robotti viimeisen käskyn määrittämään paikkaan.

Lisätyn käskyn perässä näkyy käskylle määritetyt argumentit. Näitä argumentteja ovat \*, v, z ja tool0. Argumentti \* tarkoittaa äsken robotille määritettyä paikkaa ja sille voidaan määrittää uusi käytettävä nimi/määrittely, jolla se ilmoittaa määritetyn paikan. V argumentti on suoritettavan liikkeen nopeus, joka kuvaa robotissa olevan työkalun keskipisteen kiihtyvyyttä millimetreinä sekunnissa. Argumentti z on tapa, miten pisteeseen kuljetaan. Jos valittuna on z50, kuten äsken lisätyssä käskyssä, tällöin robotti ei mene määrättyyn paikkaan, vaan kulkee paikan ohi pyörivällä liikkeellä. Pyöritys aloitetaan luvun määrittämän mitan verran millimetreinä ennen paikkaa ja pyöritys loppuu saman mitan verran pisteen toiselle puolelle. Tällä tavoin saadaan robotin

liikeradoista sulavampia, jos määrityssä pisteessä ei ole pakko käydä. Jos z50:n tilalla on fine, tällöin robotti käy määrityssä pisteessä Tool0 on automaattisesti robotille määritelty työkalu, jos muita työkaluja ei ole määritetty. Tämän työkalun keskipiste on robotin kuudennessa akselissa olevan levyn keskikohta. Robotille annettavien paikkakäskeyjen paikkatiedot määritetään tähän pisteeseen millimetreinä x, y ja z koordinaateissa robotin alustan keskikohdasta käytettäessä työkalua tool0.

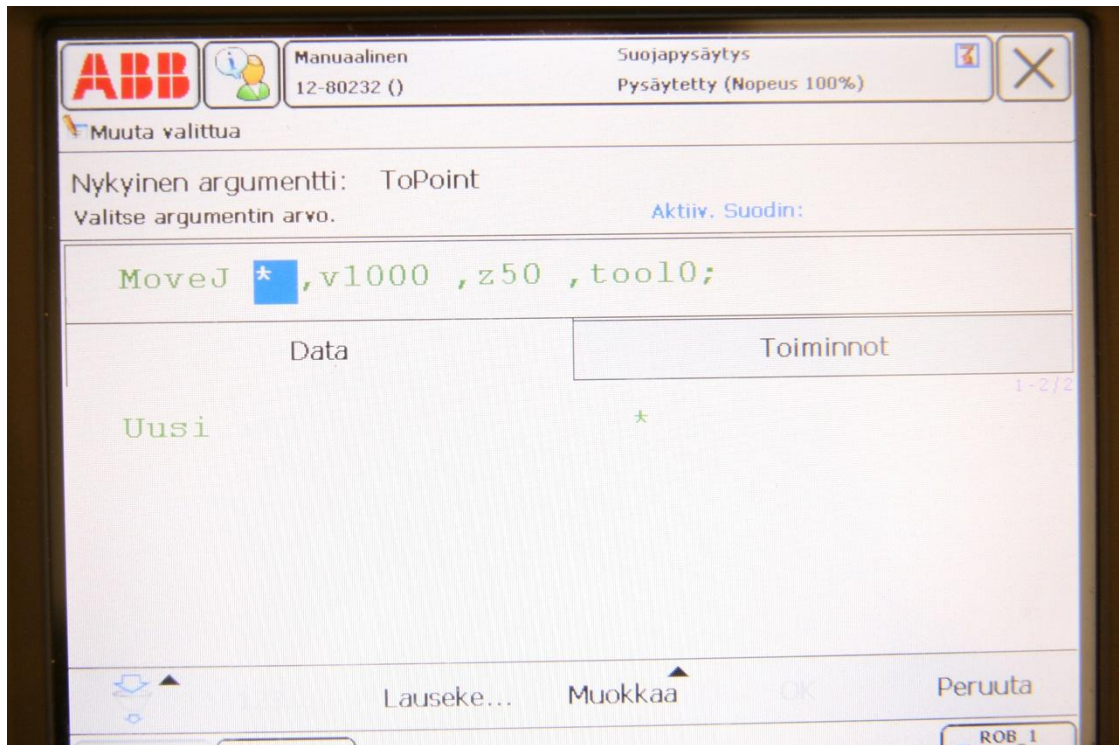
Seuraavaksi muokataan äsken annetun käskyn argumentteja. Argumenttien muokkaukseen pääsee kolmella eri tapaa. Voidaan valita koko käskyrivi napauttamalla käskyn MoveJ nimeä, jolloin koko käsky korostuu sinisellä värillä, eli kohdistin on sillä kohdalla. Sitten valitaan muokkaa -valikosta muuta valittua, jolloin aukeaa uusi ikkuna, jossa voidaan valita muutettava argumentti. Kuvassa 56 on muokattavan argumentin valintaikkuna.



**KUVA 56. Muokattavan argumentin valinta**

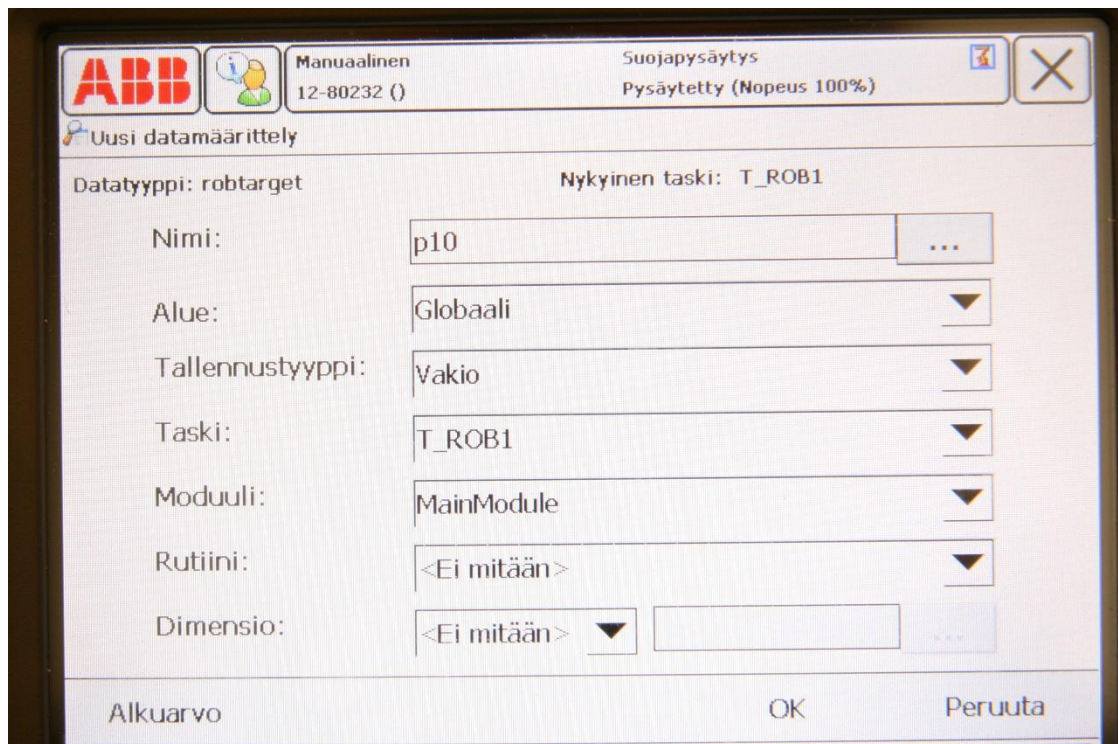
Valitaan muokattava argumentin ikkunasta kohta To Point, joka on robotin paikan \* määrittely. Tämän jälkeen aukeaa uusi ikkuna, jossa muokataan argumentin määrittelyä. Kuvassa 57 on argumentin määrittelyn muokausikkuna.





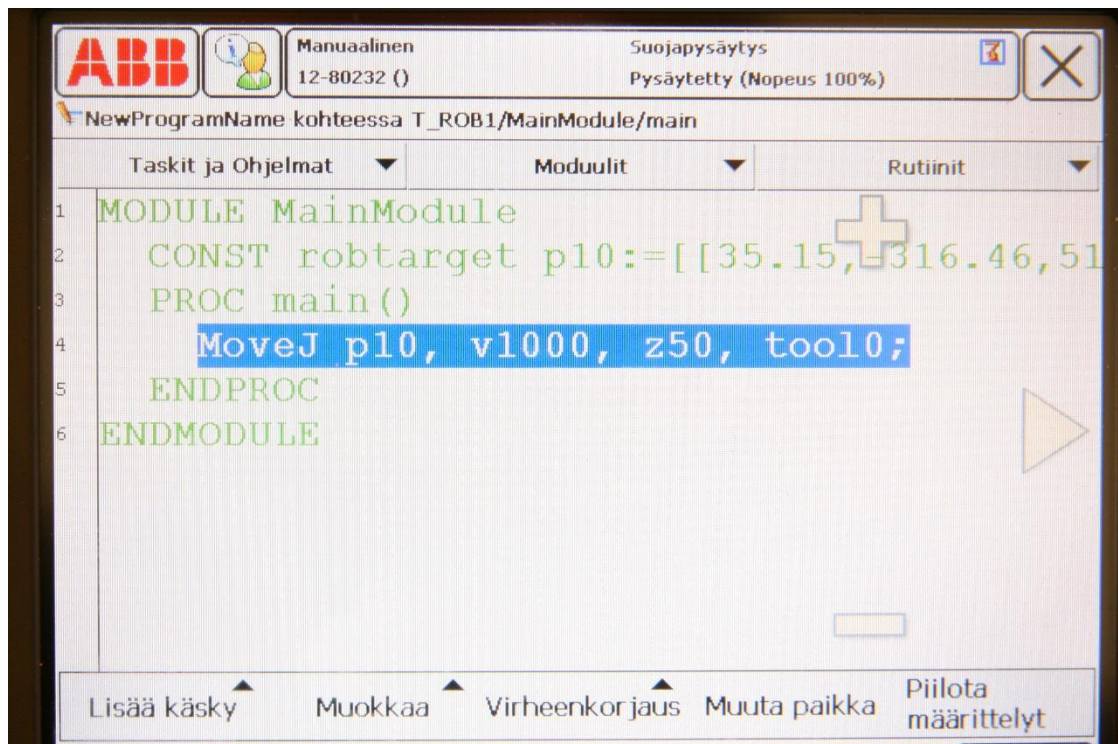
**KUVA 57 Argumentin muokkausikkuna**

Tästä ikkunasta valitaan uusi, joka avaa edelleen uuden ikkunan, jossa voidaan määrittää argumentille halutut tiedot/arvot. Jos painetaan alalaidassa olevaa lauseke - painiketta aukeaa lausekkeen muokkausikkuna, jossa voidaan samalla tavalla muokata argumentteja. Erona on se, että sitä kautta voidaan myös vaihtaa datatyyppiä, lisätä argumentille ehtoja ja muuten muokata argumentin lauseketta. Kuvassa 58 on argumentin tietojen muokkausvalikko.



**KUVA 58. Argumentin tietojen muokkausvalikko**

Annetaan paikkatiedolle järjestelmän ehdottama nimi p10. Kaikki muutkin asetukset ovat valmiiksi kunnossa ja eteenpäin voidaan jatkaa painamalla ok. Tämän jälkeen näyttöön ilmestyy uudelleen argumentin muokkausikkuna, johon on ilmestynyt paikanmäärittelyn nimi p10. Paikan p10 pitäisi olla automaattisesti valittuna ja eteenpäin voidaan jatkaa painamalla ok niin monta kertaa, kunnes ollaan takaisin ohjelmaeditorin pääikkunassa. Nyt robotille määritetyn ensimmäisen paikan nimen pitäisi olla p10 ja sen tietojen pitäisi myös ilmestyä rutiinin yläpuolella olevaan tietonäkymään, kuten kuvasta 59 näkyy.



**KUVA 59. Uusi määritetty paikkatiedon argumentti p10**

Toinen tapa muokata argumenttia on näpäyttää käskyn nimeä, jolloin käskyrivi tulee aktiiviseksi. Sitten näpäytetään uudelleen käskyn nimeä, jolloin aukeaa äsken läpi käydyt ikkunat. Kolmas tapa on tuplanäpäyttää suoraan argumenttia, jota halutaan muokata. Muilla argumenteilla on määriteltynä jo valmiina arvoja, joita voidaan valita argumenttia muokatessa. Käskylle määriteltyjen argumenttien arvot määrittelevät sen miten robotti liikkuu tähän paikkaan, eli tässä tapauksessa paikkaan p10 kuljetaan kuvassa 59 näkyvien argumenttien määrittämällä tavalla.

Toinen haluttu asento ja paikka robotille määritetään samalla tavoin kuin ensimmäinen paikka. Kun toinen paikka on määritetty, on jo yksinkertainen liikeohjelma tehty. Jos paikkoja halutaan muuttaa, voidaan käyttää ohjelmaeditorin painiketta muuta paikka. Kun tätä painiketta painetaan, päivittyy valittuna olevan paikan tiedot siihen paikkaan, jossa robotti sillä hetkellä on. Ohjelmaa voi kokeilla valitsemalla virheenkorjaus -valikosta SO päärutiniin, jolloin ohjelman osoitin ilmestyy main rutiinin ensimmäiselle käskyriville. Opetus- eli ohjelmointitilassa robotti saadaan liikkumaan painamalla pendantin päässä oleva turvakytkin ensin puoleen väliin ja pitämällä se siinä. Sitten painetaan pendantissa olevaa ohjelman käynnistys painiketta ja robotti alkaa liikkumaan äsken määriteltyjen paikkojen väliä. Robottia ohjelmoitaessa solun ovi voi olla auki, mutta kun ohjelmaa halutaan ajaa, täytyy oven olla suljettuna. Ope-

tustilassa robotti ei kuitenkaan liiku kovemmalla nopeudella, kuin 250mm/s vaikka käskyssä määritelty nopeus olisikin suurempi. Robotin nopeutta ja ohjelman suoritus-tilaa voidaan säätää pikavalikosta. Aina ennen automaattitilaan siirtymistä kannattaa kokeilla ohjelman toiminta opetustilassa. Automaattitilassa solun oven on oltava suljettuna, kuten opetustilassakin ohjelmaa ajettaessa. Kun on todettu ohjelman toimivan opetustilassa, voidaan sitä kokeilla automaattitilassa. Tällöin käännetään kontrollerissa oleva tilanvalitsin automaattiasentoon. Tällöin pendantin näytölle tulee ilmoitus automaattitilaan siirtymisestä ja jos nopeus on asetettu alle 100:n prosentin, tulee myös ilmoitus, että nopeus asetetaan 100:n prosenttiin. Kun näytöllä olevat ilmoitukset on kuitattu, aukeaa tuotantoikkuna, josta voidaan seurata ohjelman kulkua. Ennen ohjelman käynnistämistä asetetaan ohjelman osoitin pääruutiiniin. Tämä voidaan suorittaa automaattitilassa tuotantoikkunasta. Kun ohjelman osoitin on pääruutiinissa, painetaan kontrollerin moottorit päälle painikkeesta, jolloin moottorit menevät päälle. Kun moottorit on laitettu päälle, voidaan ohjelma käynnistää painamalla pendantissa olevaa ohjelman käynnistyspainiketta.

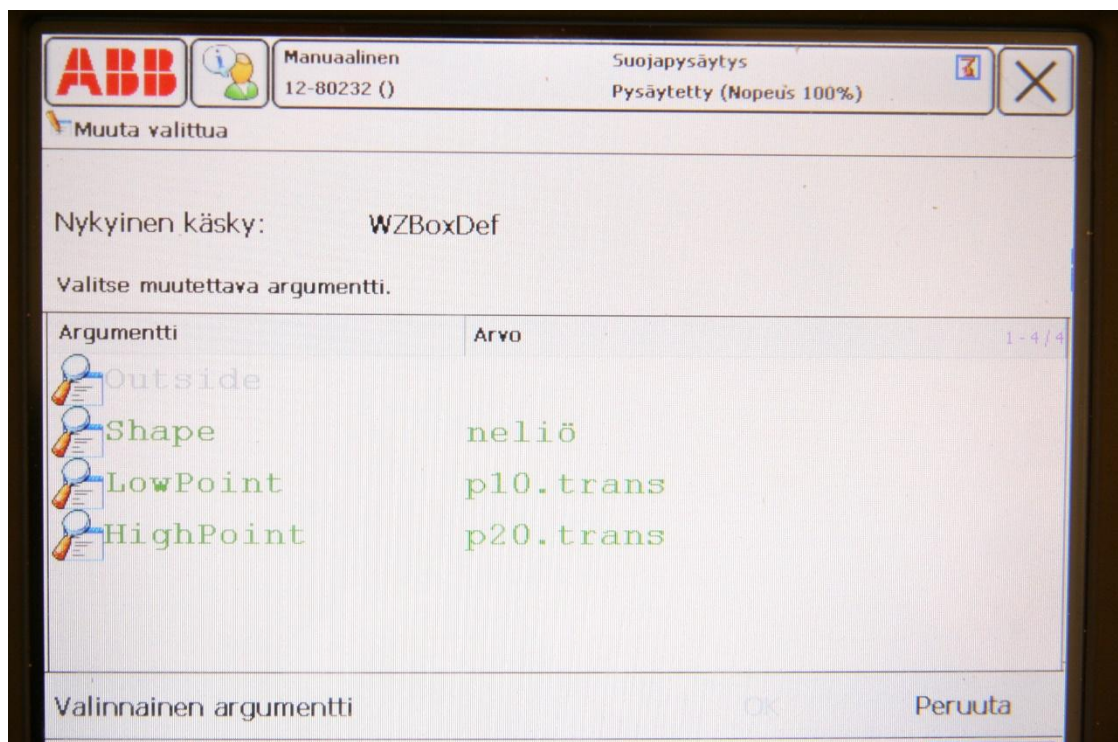
## **7.12 Työkalun kiellettyjen alueiden ja robotin liikkeiden rajojen määrittäminen**

Robotin liikkeitä voidaan rajoittaa kahdella eri tavalla. Ensimmäinen tapa rajoittaa robotin liikkeitä on valvomalla työkalun keskipisteen paikkaa määritetyllä alueella. Alueen muoto voi olla neliö, sylinteri tai pallo. Toinen tapa rajoittaa robotin liikkeitä on valvomalla robotin akseleita, eli kuinka paljon mikäkin akseli on kääntynyt. Molemmissa liikkeen rajoitustavoissa on mahdollista määrittää, lopetetaanko liike, kun robotti meinaa mennä määritetyn alueen sisälle tai lopetetaanko liike, kun robotti meinaa mennä rajoitetulta alueelta ulos. Kun määritetyn alueen rajalle on menty, ainoa tapa saada robotti takaisin sallitulle alueelle on opetustilassa liikuttamalla tai vapauttamalla robotin jarrut ja siirtämällä se sinne käsin.

### **7.12.1 Työkalun alueenrajoitus**

Työkalun alueenrajoitus valvoo ainoastaan, missä työkalun keskipiste on. Kun on luotu rajoitettu alue, tarkoittaa se sitä, että rajoitettu alue ei vaikuta robotin muihin osiin millään tavalla. Tässä esimerkissä käytän rajoitetun alueen muotona neliötä. Kun halutaan rajoittaa aluetta, jossa työkalu saa liikkua, käytetään käskyä WZBoxDef. Neliöstä täytyy antaa käskylle tiedot neliön vastakkaisista kulmista. Shape -argumentille voi-

daan antaa haluttu nimi, mieluiten sellainen, joka kuvaa aluetta selvästi. Kun shape -argumentti on luotu, luo järjestelmä datatyyppin nimeltään shapedata, jota käytetään määritettyjen lowpoint- ja highpoint -pisteiden tietojen siirtämiseen määritetyn alueen eli worldzone -alueen valvonnalle. Lowpoint -argumentille annetaan piste neliön toisessa alakulmassa ja highpoint -argumentille annetaan piste neliön toisessa yläkulmassa, joka on vastakkainen kulma lowpointille. Käskeylle voidaan määrittää valinnainen argumentti, joka määrittää, pysäytetäänkö liike, kun robotti meinaa mennä rajoitetun alueen sisälle, tai pysäytetäänkö liike, kun robotti meinaa mennä rajoitetusta alueesta ulos. Tässä esimerkissä on valittu valinnaiseksi argumentiksi outside eli, jos robotti meinaa mennä rajoitetun alueen ulkopuolelle, liike pysäytetään. Jotta rajoitetun alueen valvonta saadaan päälle, täytyy vielä käyttää käskyä WZLimSup. Tämän käskyn argumenteissa määritetään valvonnalle haluttu nimi, joka on worldzone -argumentti, josta luodaan ohjelmadataan datatyyppi wztemporary. Sille määritetään myös shape -argumentti, joka on sama kuin jo aiemmin määritetyllä WZBoxDef -käskyllä. Esimerkissä olen nimennyt shape -argumentin nimellä neliö ja worldzone -argumentin nimellä rajavalvonta. Kuvassa x on käskeylle WZBoxDef määritetyt argumentit.



**KUVA 60. Käskyn WZBoxDef argumentit ja arvot**

Ohjelmakoodi näyttää tältä ohjelmaeditorissa käytettäessä käskyä WZBoxDef:

```

MODULE MainModule
VAR shapedata neliö;
CONST robtarget p10:= [...];
CONST robtarget p20:= [...];
VAR wztemporary rajavalvonta:= [0];
PROC main()
WZBoxDef\Outside, neliö, p10.trans, p20.trans;
WZLimSup\Temp, rajavalvonta, neliö;
MoveJ p30, v100, fine, tool0
MoveJ p40, v100, fine, tool0
ENDPROC
ENDMODULE

```

Jos robottia halutaan käyttää näyttämään pisteet lowpos- ja highpos -argumenteille, voidaan paikkatiedot ottaa niihin robotille määritetyistä paikoista. Robotin paikat voidaan luoda normaalisti käyttäen jotain liikekäskyistä tai luomalla itse ohjelmadataan paikan tiedot datatyyppeihin robtarget. Jotta paikkatieto robotin paikasta saadaan siirrettyä lowpoint- tai highpoint -argumenteille, tehdään seuraavalla tavalla. Näpätetään argumenttia kahdesti, jolloin aukeaa argumentin muokkausikkuna. Tästä ikkunasta valitaan datatyypiksi robtarget, jolloin näkyviin tulee robotin paikat. Argumentiksi valitaan luotu robotin paikkatieto, kuten yllä olevassa ohjelmakoodissa p10 ja sitten lauseketta muokataan valitsemalla alalaidasta muokkaa ja kaikki. Nyt voidaan argumentin tekstiä muokata. Paikkatiedon perään kirjoitetaan .trans, jolloin argumentin nimi on kokonaisuudessaan p10.trans ja paikkatietoa voidaan käyttää osoittamaan rajoitetun neliöalueen kulma. Sitten painetaan niin monta kertaa ok, kunnes näytössä on ohjelmaeditorin pääikkuna.

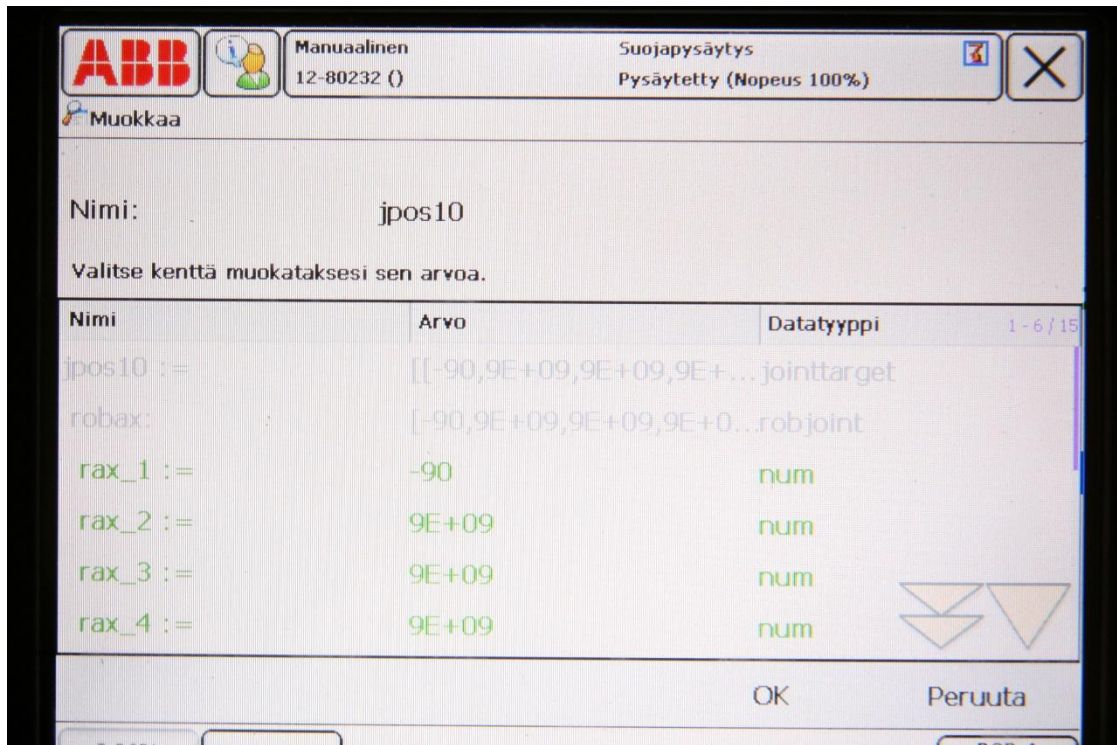
Jos luotuja rajoja halutaan käyttää opetustilassa, täytyy ohjelman osoitin siirtää pääruutiiniin ja käynnistää ohjelma kerran. Jos suoritustila on jatkuvalla kierolla, tulee ensimmäisen kierron jälkeen virheviesti, jossa sanotaan, että worldzone on jo käytössä, se on määritetty ja aktivoitu. Jotta tätä virheviestiä ei tule, täytyy suoritustila olla yhdellä kierolla. Kun yksi kierto on suoritettu ja ohjelmanosoitin on palannut ohjelman alkuun tai ohjelma on keskeytetty kesken ohjelman suorituksen, jää määritetyn alueen valvonta päälle. Kun ohjelma käynnistetään, menee worldzone -alueen valvonta päälle ja se ei resetoitu ennen, kuin ladataan kokonaan uusi ohjelma tai asetetaan itse ohjel-

manosoitin rutiinin alkuun. Jos halutaan, että rajoitetun alueen valvonta toimii opetus-tilassa ja automaattitilassa jatkuvalla kierrolla, on ohjelman loppuun lisättävä käsky WZFree, joka resetoii worldzone -alueen valvonnan, jolloin valvonta lähtee pois päältä. Kun ohjelma siirtyy sen jälkeen takaisin alkuun, aktivoituu worldzone -alueen valvonta uudelleen.

### 7.12.2 Robotin liikkeiden rajat

Robotin liikkeiden rajat määritellään jokaiselle akselille erikseen. Liikealueiden rajoitus tapahtuu oikeastaan muutoin samalla tavalla, kuin työkalun alueenrajoitus, paitsi että käsitellään eri datatyyppejä. Liikealueen rajat määritellään käskyllä WZLimJointDef ja liikealueen valvonta otetaan käyttöön kuten edellä käyttäen käskyä WZLimSup. Halutut liikkeiden rajat määritetään datatyyppiin jointtarget, johon annetaan jokaiselle akselille haluttu sallittu kääntökulma. Datatyyppi jointtarget voidaan luoda itse ohjelmadatassa tai se saadaan luotua suoraan näpäyttämällä argumenttia käskyssä. Argumentit, joille täytyy antaa arvot määrittäessä liikerajoja ovat lowjointval ja highjointval. Tässä esimerkissä argumentille lowjointval on annettu nimi jpos10 ja argumentille highjointval nimi jpos20.

Liikerajat on määritelty siten, että robotin ensimmäinen akseli voi liikkua -90 asteesta +90 asteeseen eli kokonaisuudessaan se voi kääntyä 180 astetta toiselta puolelta toiselle. Jotta argumentille saadaan annettua halutut arvot ohjelmadatassa, valitaan halutun niminen argumentti, painetaan alalaidassa olevaa painiketta muokkaa ja valitaan muuta arvoa. Jos argumenttia näpäytetään suoraan käskystä, argumentin tietojen muokausvalikossa painetaan vasemmassa alalaidassa olevaa alkuarvo painiketta. Tämän jälkeen päästään ikkunaan, josta voidaan muuttaa argumentin arvoja. Argumentille määritetään arvot rax\_1-rax\_6. Rax\_1 on robotin ensimmäinen akseli ja niin edespäin. Arvoja määrittäessä voidaan robotin akseleiden kulmia katsoa käsin ajo -valikosta. Annettaessa argumentille arvoja on otettava myös huomioon, että highposval miinus lowjointval arvon on oltava suurempi kuin nolla kaikilla akseleilla, jotta liikerajojen valvontaa voidaan käyttää. Jos halutaan, että toista tai molempia akseleiden rajoista ei valvota, annetaan sen kohdan arvoksi 9E+09. Kuvassa 61 on ikkuna, jossa voidaan antaa argumentille arvoja.



**KUVA 61. Argumentin lowjointval arvojen määrittely**

Ohjelmakoodi näyttää tältä ohjelmaeditorissa käytettäessä käskyä WZLimJointDef:

```

MODULE MainModule
VAR shapedata rajat;
CONST jointtarget jpos10:= [-90...];
CONST jointtarget jpos20:= [90...];
VAR wztemporary rajavalvonta:= [0];
PROC main ()
WZLimJointDef\Outside, rajat, jpos10, jpos20;
WZLimSup\Temp, rajavalvonta, rajat;
MoveJ p30, v100, fine, tool0
MoveJ p40, v100, fine, tool0
ENDPROC
ENDMODULE

```

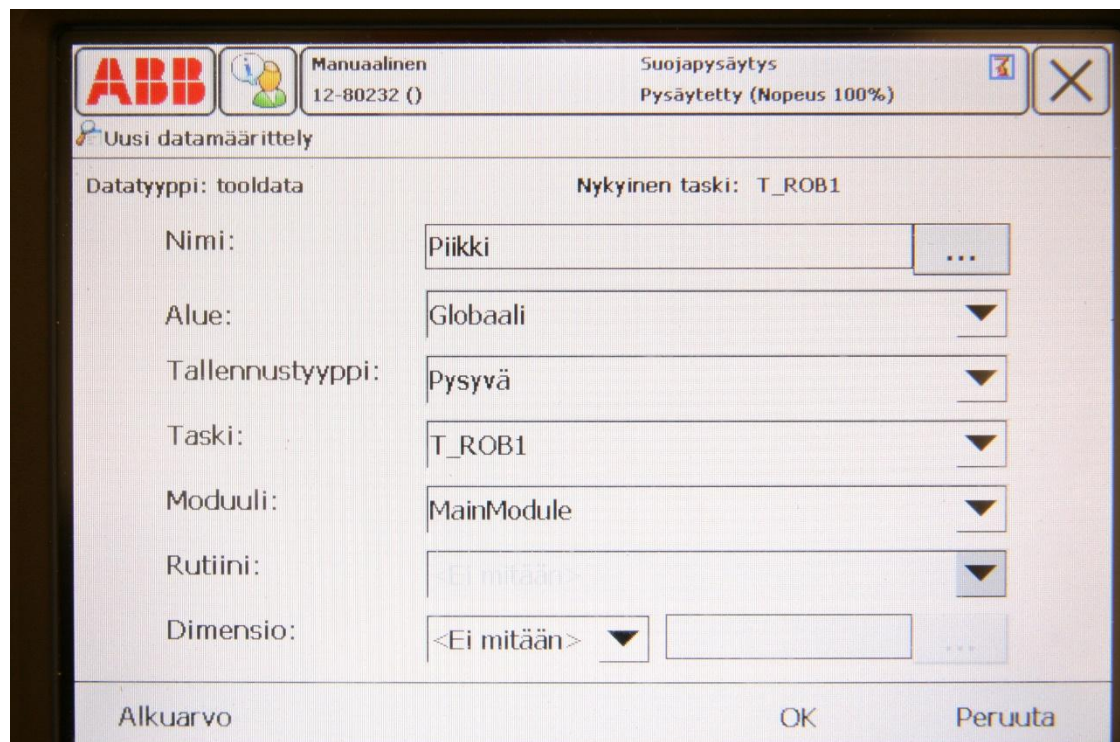
Robotin työkalulle ja liikkeille voi olla luotuna kaiken kaikkiaan kymmenen valvottua worldzone -aluetta samaan aikaan. Jos valvottavia alueita on useita, voidaan käyttää kaikissa alueissa samaa shapedataa, jolla siirretään määritetyn rajoitetun alueen tiedot



worldzone -alueen valvonnalle. Tällöin shapedata kannattaa nimetä esimerkiksi nimellä rajat.

### 7.13 Uuden työkalun määrittäminen

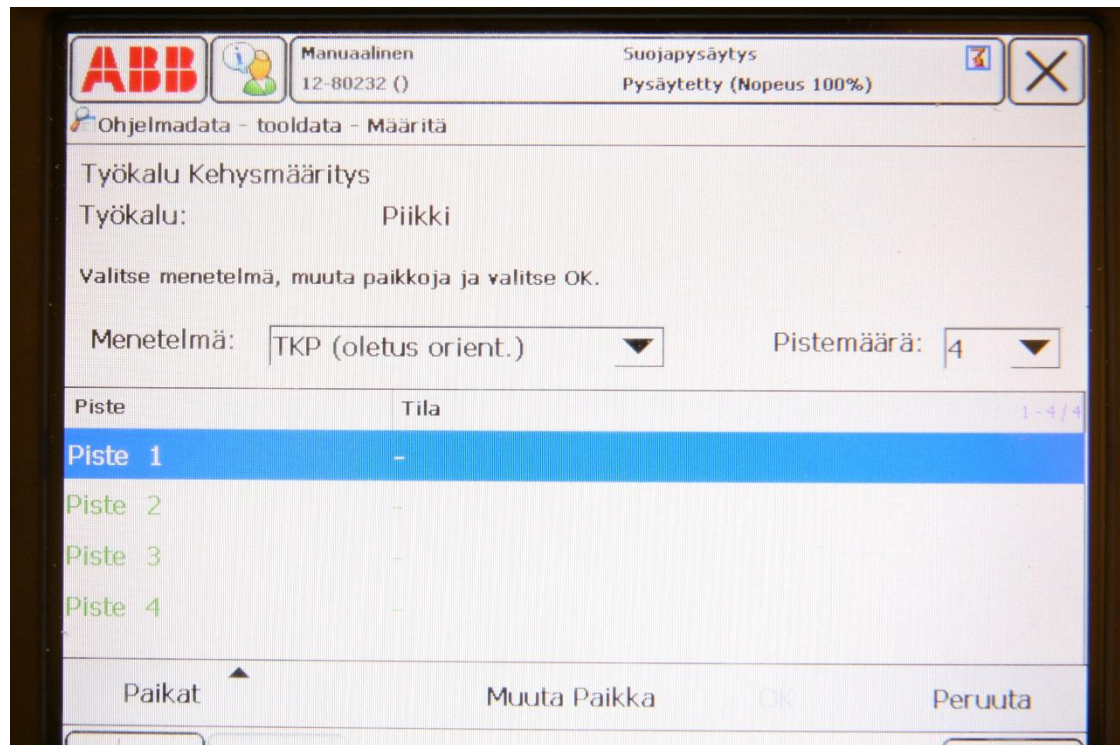
Kun robotille halutaan määrittää uusi työkalu, mennään ABB -valikosta käsin ajoon ja käsin ajo -valikosta valitaan kohta työkalu. Kun kohta työkalu on valittu, aukeaa luettelo robotin työkaluista. Tässä luettelossa on aina työkalu tool0, joka on robotin oletustyökalu, jonka keskipiste on kuudennessa akselissa kiinni olevan kiinnityslevyn keskipiste. Muut työkalut luodaan siirtymänä tästä keskipisteestä. Työkalu luettelon vasemmasta alalaidasta painetaan painiketta uusi, jolloin aukeaa työkalun tietojen muokkausvalikko. Tässä työkalun määrittelyssä annetaan työkalulle nimi ja työkalun alueeksi valitaan globaali, jotta sitä voidaan käyttää kaikissa moduuleissa. Tallennustyyppiltään työkalujen pitää olla aina pysyviä. Kuvassa 62 on työkalun tietojen muokkausvalikko.



**KUVA 62. Työkalun tietojen muokkausvalikko, esimerkissä työkalu on nimetty piikiksi**

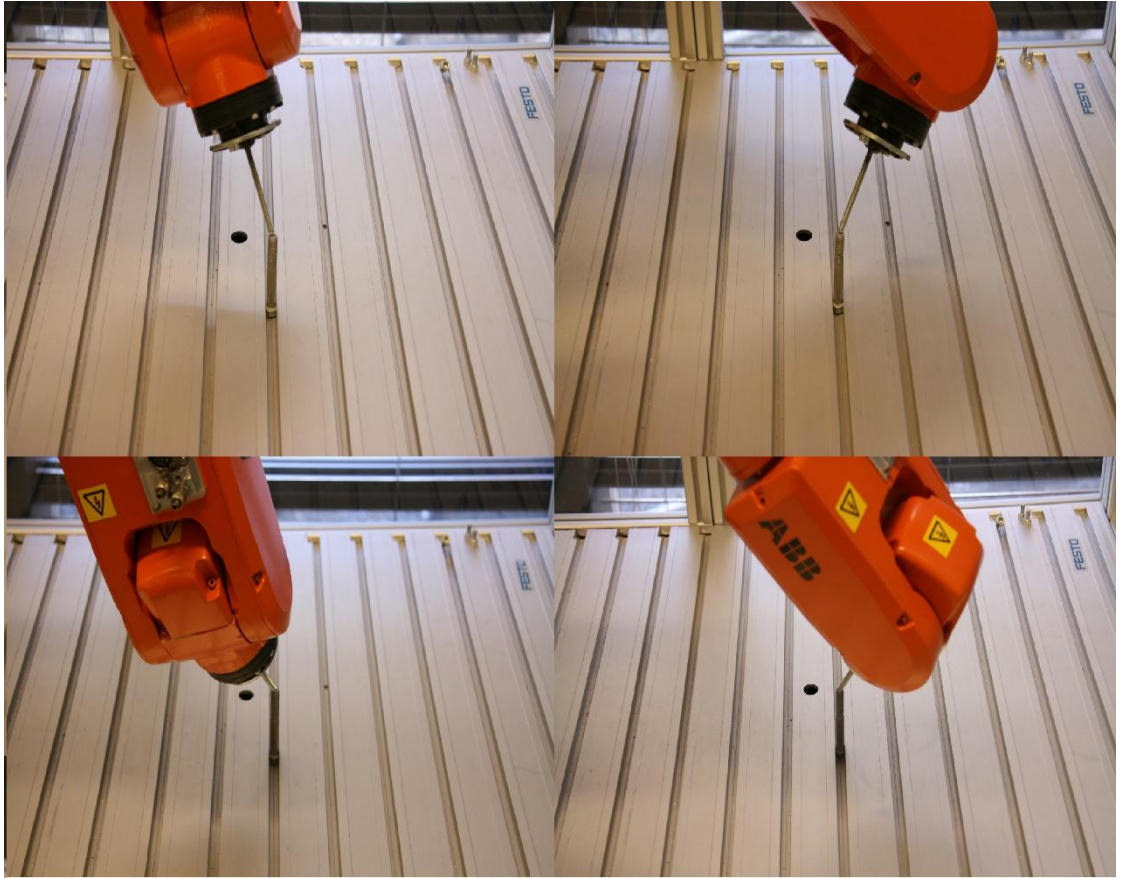
Kun nämä asetukset ovat kunnossa, painetaan ok. Uusi työkalu ilmestyy työkaluluetteloon ja sen jälkeen sen koordinaatisto on määriteltävä. Määrittely suoritetaan valitse-

malla työkalu luettelosta ja painamalla muokkaa ja määritä. Tämän jälkeen aukeaa uusi ikkuna, josta valitaan menetelmä, jolla työkalun koordinaatisto määritetään. Menetelmää TKP, eli työkalun keskipiste, voidaan käyttää, kun työkalun suunta on sama, kuin robotin kiinnityslevyn suunta. Kuvassa 63 on työkalun koordinaatiston määrittämisvalikko.



**KUVA 63. Työkalun koordinaatiston määrittämisvalikko**

Tämän jälkeen määritetään pisteet uudella työkalulla. Tämä tehdään valitsemalla robotin työalueelta jokin kiinteä viitepiste, johon määritetään neljä pistettä mahdollisimman tarkasti. Robotin työkalu ohjataan mahdollisimman lähelle haluttua pistettä, ja kun kohtaan ollaan tyytyväisiä, valitaan piste yksi ja painetaan painiketta muuta paikka. (Katso kuva 63) Tässä esimerkissä kiinteänä viitepisteenä on käytetty pultin kärkeä. Kun muuta paikka painiketta on painettu, muuttuu valitun pisteen tilaksi muutettu ja valinta siirtyy seuraavaan pisteeseen. Robotti ohjataan uudestaan samaan viitepisteeseen eri asennoissa ja pisteille suoritetaan sama muuta paikka toiminto. Kuvassa 64 on esimerkki robotin asennoista.



**KUVA 64. Robotin eri asennot kiinteän viitepisteen ympärillä**

Kun kaikki paikat on määritetty, painetaan ok. Näytöllä kysytään, halutaanko paikat tallentaa myöhempää käyttöä varten, johon vastataan kyllä, jolloin järjestelmä luo uuden moduulin. Tämän jälkeen tulee näyttöön laskentataulukko, jossa on laskettu äsken määritettyjen paikkojen tarkkuudet kiinteästä viitepisteestä. Jos tuloksiin ollaan tyytyväisiä, painetaan ok.

Viimeinen vaihe työkalun määrittämisessä on sen massan määrittäminen. Annetaan työkalulle arvioitu massa, esimerkiksi 0,2 kilogrammaa. Tämä arvo voidaan antaa työkalulle valitsemalla työkalu työkaluluettelosta, painamalla muokkaa ja muuta arvoa. Massa kirjoitetaan kohtaan mass, ja se annetaan kilogrammoina. Kun arvioitu massa on annettu, mennään ohjelmaeditoriin, siirretään ohjelman osoitin päärutiiniin ja valitaan virheenkorjaus -valikosta kutsu rutiini. Rutiini valikosta valitaan huoltorutiini nimeltään loadidentify ja painetaan oikealla alhaalla olevaa siirry -painiketta. Huoltorutiini käynnistetään painamalla pendantin turvapainike puoleen väliin ja painamalla käynnistä -painiketta. Tämän jälkeen seurataan näytöllä olevia ohjeita. Kun näytöllä kysytään, tunnistetaanko työkalu vai kuorma, valitaan työkalu. Tunnistusmenetelmäksi valitaan myöhemmin numero kaksi eli tuntematon massa kohteessa työkalu ja ko-

koonpanon kulmaksi plus tai miinus 90 astetta. Loput huoltorutiinista suoritetaan näytön ohjeiden mukaan. Kun robotti on suorittanut huoltorutiinin liikkeet, ilmestyy näytölle tulokset työkalulle määritetystä painosta ja painopisteestä. Huoltorutiini kysyy lopuksi myös päivitetäänkö työkalun tiedot ja kun vastataan kyllä, tiedot päivittyvät ja huoltorutiini päättyy. Kannattaa tallentaa kyseinen ohjelma, johon työkalu on määritetty, jotta sitä voidaan käyttää uudelleen. Tällöin määritetyn työkalun käyttö onnistuu muissakin ohjelmissa.

### 7.14 Yleisiä käskyjä

Seuraavat käskyt ovat yleisiä käskyjä, joita robotin ohjelmoinnissa käytetään.

**MoveJ:** Robotti liikkuu määrättyyn paikkaan siten, että se pystyy kääntämään kaikkia aksleita tarvitsemallaan tavalla, jotta paikkaan pääsee.

Esimerkki:

```
PROC main ()
MoveJ p10, v100, fine, tool0
MoveJ p20, v100, fine, tool0
ENDPROC
```

**MoveL:** Robotti liikkuu määrättyyn pisteeseen lineaarisesti. Tämä komento ei toimi, jos yritetään siirtyä robotin toiselta puolelta toiselle puolelle, koska robotti ei voi kulkea itsensä läpi.

Esimerkki:

```
PROC main ()
MoveL p10, v100, fine, tool0
MoveL p20, v100, fine, tool0
ENDPROC
```

**MoveC:** Robotti tekee määrättyjen kolmen pisteen välisen liikkeen kaarevana liikkeenä. Ensimmäinen on aiemman käskyn paikka, toinen on kaaren piste, eli kuinka kauas robotti tekee kaaren ja kolmas on kaaren loppupiste, eli paikka, johon robotti menee tehtyään kaaren.

Esimerkki:

```
PROC main ()
MoveJ p10, v100, fine, tool0
MoveJ p20, v100, fine, tool0
MoveC p30, p40, v100, fine, tool0
ENDPROC
```

**FOR:** Tätä käskyä käytettäessä robotti toistaa tämän käskyn käskyriveille määritetyn ohjelman halutun määrän verran, eli se on counteri, joka numeroita ylöspäin tai alaspäin riippuen sille määritetyistä arvoista. Käsky alkaa tekstillä FOR ja päättyy tekstiin ENDFOR. ENDFOR:in jälkeen olevat käskyt suoritetaan normaaliin tapaan.

Käsky näyttää ilman määritettyjä arvoja tältä:

```
FOR <ID> FROM <EXP> TO <EXP> DO
<SMT>
ENDFOR
```

<ID> on counterille määritettävä nimi, <EXP> kohtiin määritetään, mistä aletaan laskemaan ja mihin asti lasketaan. DO -tekstin jälkeen <SMT> kohdassa olevat käskyt suoritetaan niin monta kertaa kuin counterin arvoissa on määritetty.

Esimerkki: FOR -käskyn sisältämä liikerata tehdään kolme kertaa.

```
PROC main ()
FOR i FROM 1 TO 3 DO
MoveJ p10, v100, fine, tool0
MoveJ p20, v100, fine, tool0
ENDFOR
MoveJ p30, v100, fine, tool0
```

```
MoveJ p40, v100, fine, tool0
ENDPROC
```

**Käskey** := on muuttujien luomista ja sen arvojen määrittämistä varten. Muuttujalle valitaan datatyyppi. Datatyyppi voi olla esimerkiksi num, eli numeerinen arvo, string, joka on tekstiä, tai bool, jonka arvo on true tai false. Esimerkissä muuttujalle a on määritetty arvo 5 ja b:lle 10. C:n arvoksi on määritetty lauseke, joka laskee a:n ja b:n arvot yhteen, eli c:n arvo on 15.

Esimerkki:

```
PROC main ()
a := 5
b := 10
c := a+b
ENDPROC
```

**WaitTime:** Tällä käskyllä voidaan asettaa aika, kuinka kauan robotti odottaa tekemättä mitään, ennen kuin se suorittaa seuraavan käskyn.

Esimerkki: Robotti odottaa paikassa p20 kolme sekuntia ennen, kuin se jatkaa pisteeseen p30.

```
PROC main ()
MoveJ p10, v100, fine, tool0
MoveJ p20, v100, fine, tool0
WaitTime 3
MoveJ p30, v100, fine, tool0
ENDPROC
```

**Set ja Reset:** Käskyllä Set voidaan asettaa haluttu ulostulo päälle ja käskyllä Reset haluttu ulostulo voidaan laittaa pois päältä.

Esimerkki:

```

PROC main ()
MoveJ p10, v100, fine, tool0
MoveJ p20, v100, fine, tool0
WaitTime 1
Set DO5
WaitTime 1
MoveJ p30, v100, fine, tool0
WaitTime 1
Reset DO5
WaitTime 1
MoveJ p40, v100, fine, tool0
ENDPROC

```

**ProcCall ja RETURN** käskyä käytetään, kun halutaan siirtyä rutiinien välillä. ProcCall -käskyn tilalle tulee sen rutiinin nimi, johon siirrytään. Kun käskyt on suoritettu, jotka ovat rutiinissa, johon on siirrytty, jatkaa ohjelma takaisin rutiiniin, jossa alun perin oltiin. Ohjelman suoritus jatkuu normaalisti eteenpäin ProcCall -käskyn jäljessä olevien käskyjen mukaisesti. RETURN -käskyä käytetään, kun halutaan siirtyä takaisin rutiiniin, josta siirryttiin toiseen rutiiniin. Tätä voidaan käyttää esimerkiksi silloin, kun ei haluta suorittaa koko rutiinia loppuun asti.

Esimerkki

```

PROC main ()
MoveJ p10, v100, fine, tool0
MoveJ p20, v100, fine, tool0
Routine1;
MoveJ p50, v100, fine, tool0
MoveJ p70, v100, fine, tool0
ENDPROC
PROC Routine1 ()
MoveJ p30, v100, fine, tool0
MoveJ p40, v100, fine, tool0
RETURN;
MoveJ p60, v100, fine, tool0

```

ENDPROC

Robottijärjestelmän kanssa voidaan kommunikoida käyttämällä käskyjä, joiden sisältämät tekstit tulevat näkyviin operaattori-ikkunaan. Näitä käskyjä ovat esimerkiksi **TPWrite** ja **TPReadNum**.

Esimerkki TPWrite käskystä:

```
PROC main ()
TPWrite ‘ ’
ENDPROC
```

Käskyssä oleviin hakasulkuihin voidaan kirjoittaa teksti, jonka halutaan ilmestyvän operaattori-ikkunaan.

Esimerkki TPRReadNum käskyn käytöstä:

```
PROC main ()
TPReadNum <EXP>, ‘ ’ ’;
FOR <ID> FROM <EXP> TO <EXP> DO
<SMT>
ENDPROC
```

TPReadNum -käskyn <EXP> kohtaan voidaan esimerkiksi laittaa numeerinen muuttuja reg1, joka on user -järjestelmämoduulissa ennalta määritetty muuttuja, jonka arvo voidaan antaa pendantilla. Heittomerkkien väliin voidaan kirjoittaa vaikka kysymys, johon sitten voidaan vastata antamalla arvo pendantilla. Tämä vastaus on muuttujan reg1 arvo.

```
PROC main ()
TPReadNum reg1, ’Monta kertaa kierros tehdään? ’;
FOR i FROM 1 TO reg1 DO
MoveJ p10, v100, fine, tool0
MoveJ p20, v100, fine, tool0
ENDPROC
```

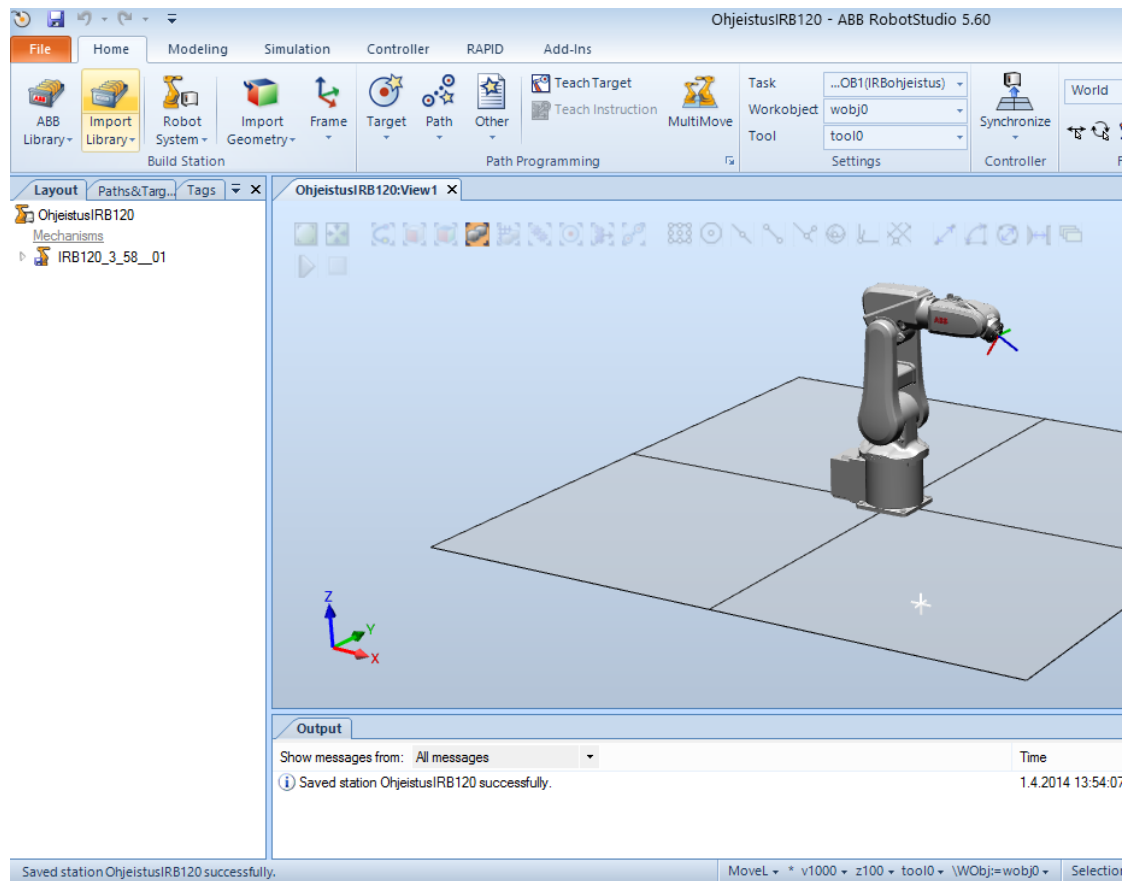


Kun tämän ohjelmakoodin mukainen ohjelma suoritetaan, tapahtuu seuraavaa. Ope-  
raattori-ikkuna tulee esille, ja siinä näkyy kysymys, monta kierrosta ajetaan. Kysy-  
mykseen voidaan vastata pendantin näytössä olevalla numeronäppäimistöllä. Kun  
kysymykseen on vastattu, suorittaa ohjelma liikekäskyt yhtä monta kertaa kuin vasta-  
uksessa on annettu.

### **7.15 Robotstudio**

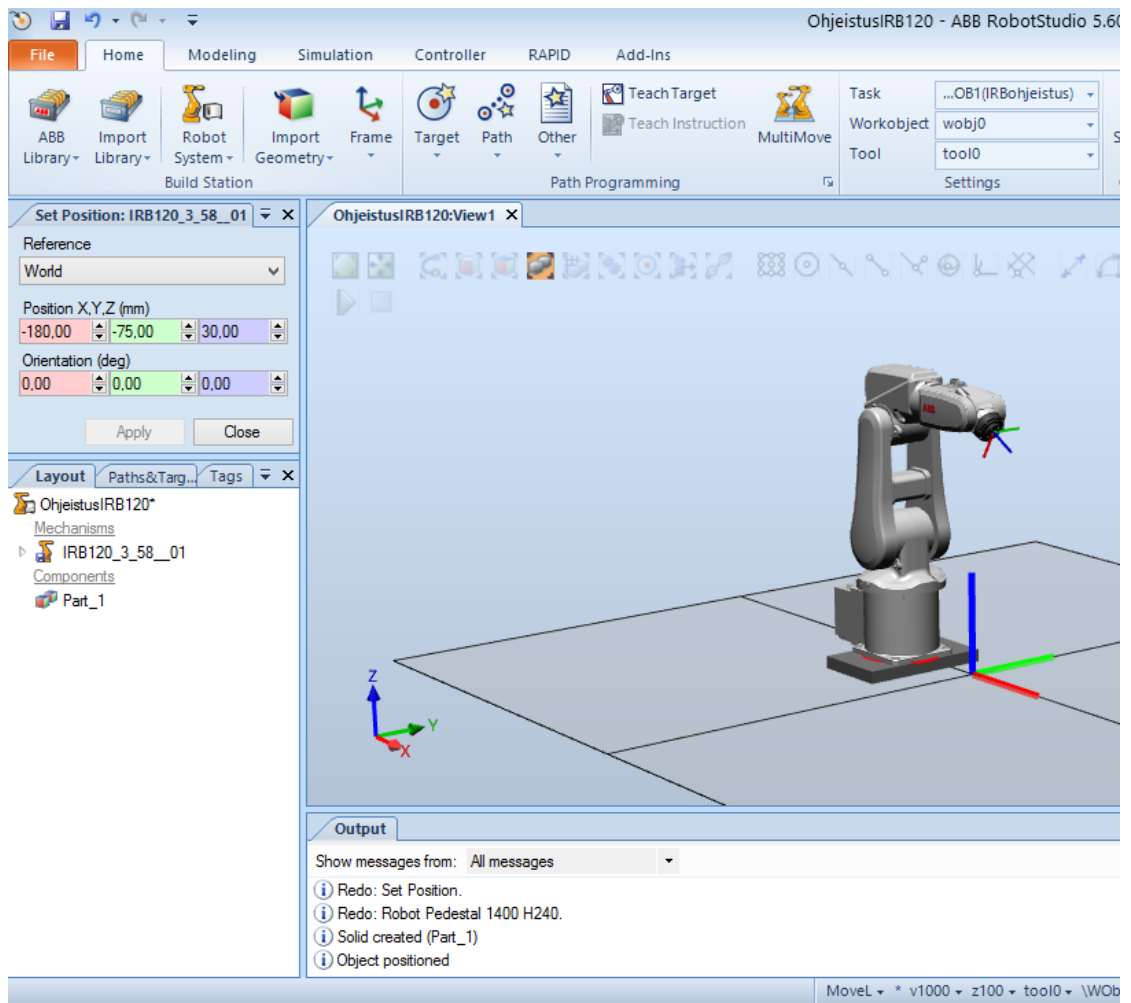
Robotstudio -simulointiohjelma aukeaa työpöydällä olevasta pikakuvakkeesta Robot-  
studio 5.60. Uusi asema luodaan valitsemalla file valikosta new, sitten empty station  
ja create. Tämän jälkeen home -valikko avautuu ja näytöllä näkyy tyhjä työaseman  
pohja. Aseman ympärillä voidaan liikkua sivuttain ja pystysuunnassa painamalla ctrl +  
hiiren vasen painike ja raahaamalla. Painamalla ctrl + shift +hiiren vasen painike ja  
raahaamalla voidaan asemaa pyörittää. Zoomaus toimii hiiren rullasta tai painamalla  
ctrl + hiiren oikea painike ja raahaamalla. Robotti voidaan lisätä luotuun työasemaan  
painamalla home valikossa olevaa ABB library painiketta, josta avautuu luettelo eri  
robottimalleista. Luettelosta valitaan robotti IRB 120, jonka jälkeen robotti ilmestyy  
työasemaan.

Virtuaalinen kontrolleri voidaan lisätä painamalla home -valikosta robot system paini-  
kettä, josta valitaan system from layout. Tämän jälkeen aukeaa ikkuna, jossa annetaan  
järjestelmälle nimi ja voidaan valita tallennus-sijainti, jos sitä halutaan muuttaa. Kun  
nimi ja tallennus-sijainti on annettu, painetaan next. Sitten valitaan mekaaniset yksi-  
köt, joita järjestelmä sisältää. Näitä ei ole kuin yksi, joka on aiemmin valittu IRB 120 -  
robotti. Painetaan taas next ja uudessa ikkunassa näkyy järjestelmä yhteenveto. Järjes-  
telmään voidaan lisätä options -painikkeen kautta samat robotware -optiot, joita oike-  
assakin järjestelmäsää on. Kun järjestelmän kokoonpanoon ollaan tyytyväisiä, paine-  
taan finish, jonka jälkeen virtuaalinen kontrolleri käynnistyy ja sen tila näkyy oikeassa  
alakulmassa. Kuvassa 65 on edellä olevien ohjeiden mukaan luotu työasema.



## KUVA 65. Luotu työasema

Vasemmalla ohjelmaikkunassa näkyy pudotusvalikko, johon aukeaa eri valikkojen, kuten home -valikon sisältämät kohteet, jotka on luotu. Robotin paikkaa työasemassa voidaan siirtää klikkaamalla oikealla hiiren painikkeella mekaanisen yksikön eli robotin nimeä ja valitsemalla set position. Modeling -valikossa voidaan luoda erilaisia kappaleita työasemaan. Esimerkiksi robotin alla oleva kiinnitysalusta voidaan tehdä modeling -valikon kautta. Kappaleita voidaan siirtää samalla tavalla kuin robotin paikkaa. Kuvassa 66 robotti on kiinnitysalustan päällä ja siirretty paikkaan, jossa se on oikeassa järjestelmässä.



## KUVA 66. Robotti samassa paikassa kuin oikeassa järjestelmässä

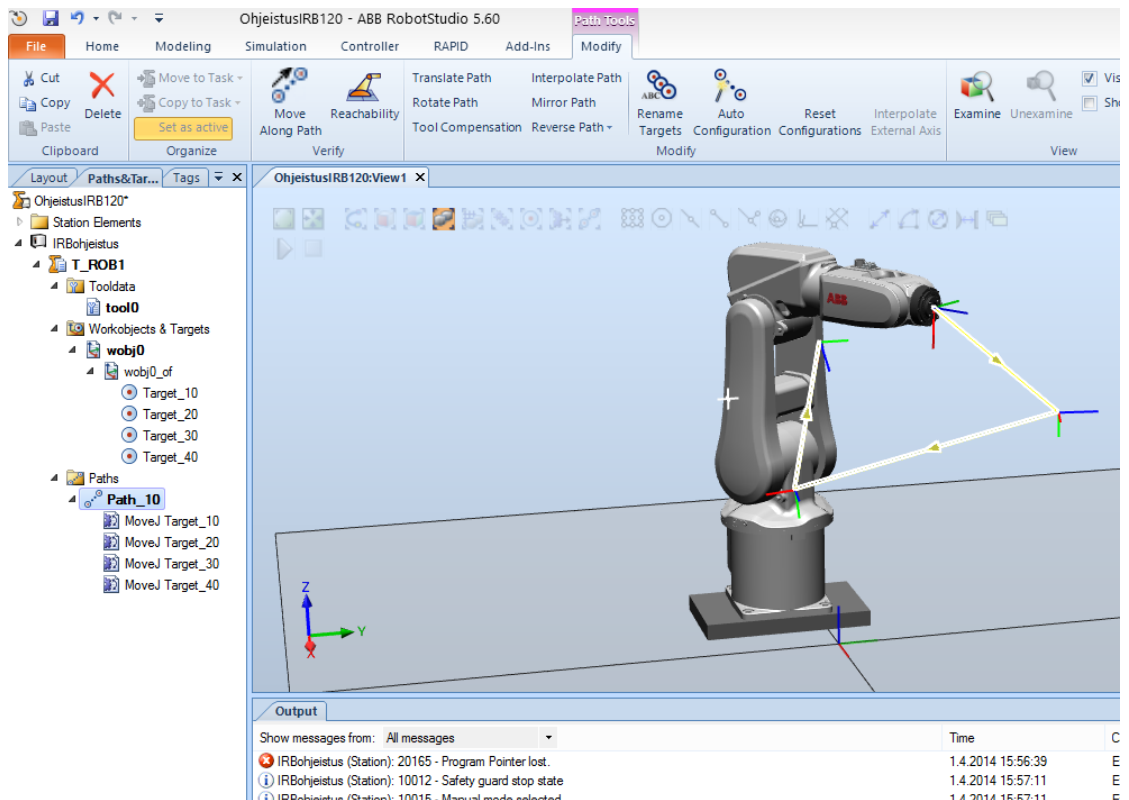
Robotin asentoa kappaleita voidaan liikuttaa home valikossa olevasta freehand -kohdasta. Robottia voidaan liikuttaa vapaasti, tai liiketavaksi voidaan valita lineaarinen, jolloin työkalun asento pysyy samana. Controller -valikosta kohdasta flexpendant saadaan näkyville virtuaalinen pendantti, jolla on samat ominaisuudet kuin oikeallakin pendantilla, ja sillä voidaan muun muassa liikuttaa robottia simulointiohjelmassa.

### 7.15.1 Robotin ohjelmointi robotstudiassa

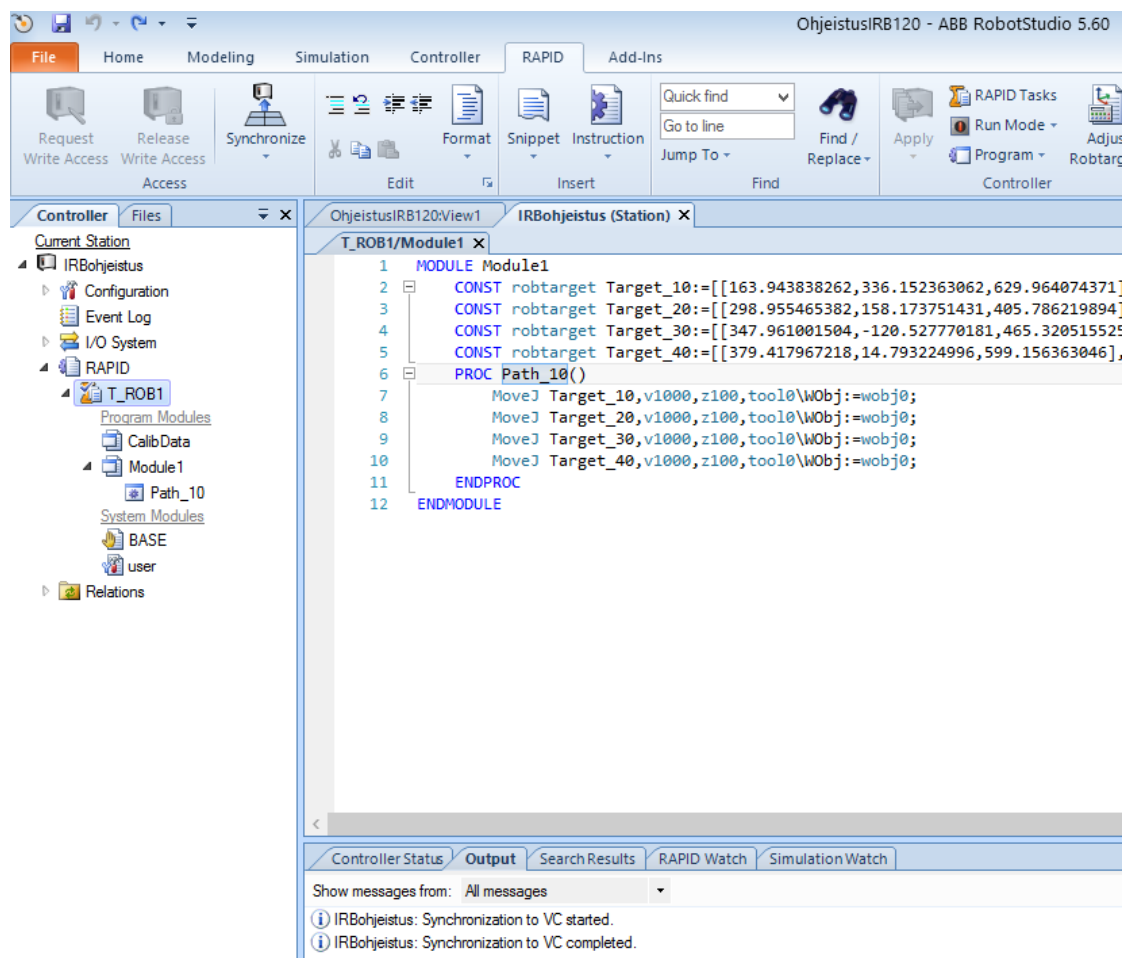
Robotille voidaan opettaa paikat simulointiohjelmassa, kuten oikeassakin järjestelmässä. Robotti ohjataan haluttuun paikkaan ja valitaan home -valikosta kohta teach target, jolloin paikka tallentuu. Oletuksena käytössä on työkalu tool0 ja työkohteena wobj0. Opetetut paikat ilmestyvät vasemmalla olevaan pudotusvalikkoon kohtaan paths and targets nimellä target\_10 ja target\_20 ja niin edelleen. Paikkoja opetetaan haluttu määrä, jonka jälkeen niistä luodaan robotille reitti. Reitti luodaan painamalla

home -valikosta kohtaa path ja valitsemalla empty path, jolloin pudotusvalikon path kohtaan tulee uusi reitti nimeltään path\_10. Kaikki opetetut paikat valitaan, ja ne voidaan raahata luotuun reittiin, jolloin näistä pisteistä muodostuu robotille reitti/ohjelma rutiini. Reitin viivat piirtyvät samalla työasemaan näkyville. Jos valitaan jokin luoduista paikoista, esimerkiksi target\_10, voidaan modify -valikosta muun muassa tarkastella robottia kyseisessä pisteessä ja valita robotille eri asentoja, joissa se pisteessä on, configuration -painikkeesta. Samat toiminnot löytyvät myös klikkaamalla paikan nimeä hiiren oikealla painikkeella.

Luodun reitin paikkakäskeyjä voidaan muokata klikkaamalla hiiren oikealla haluttua käskeyä ja valitsemalla modify instruction. Pudotusvalikkoon ilmestyy valikko, josta voidaan muokata käskeyn argumentteja, kuten oikeassakin järjestelmässä. Rapid valikossa voidaan muokata ja kirjoittaa RAPID -koodia. Luotu reitti saadaan rapid -valikkoon seuraavalla tavalla. Mennään rapid valikkoon ja valitaan kohdasta synchronize, synchronize to VC, jolloin valitut asemassa luodut reitit siirretään virtuaaliseen kontrolleriin. Tästä aukeaa valikko, jossa voidaan valita, mitä reittejä siirretään virtuaaliseen kontrolleriin ja mihin moduuliin se tehdään. Luodun reitin RAPID -koodi saadaan näkyviin ollessa rapid -valikossa ja avaamalla vasemmalla olevasta pudotusvalikosta luotu moduuli tuplaklikkaamalla. Kuvassa 67 on pudotusvalikossa näkyvillä paikat ja niistä muodostetut reitin käskeyt ja kuvassa 68 on reitistä RAPID -koodi rapid -valikossa.



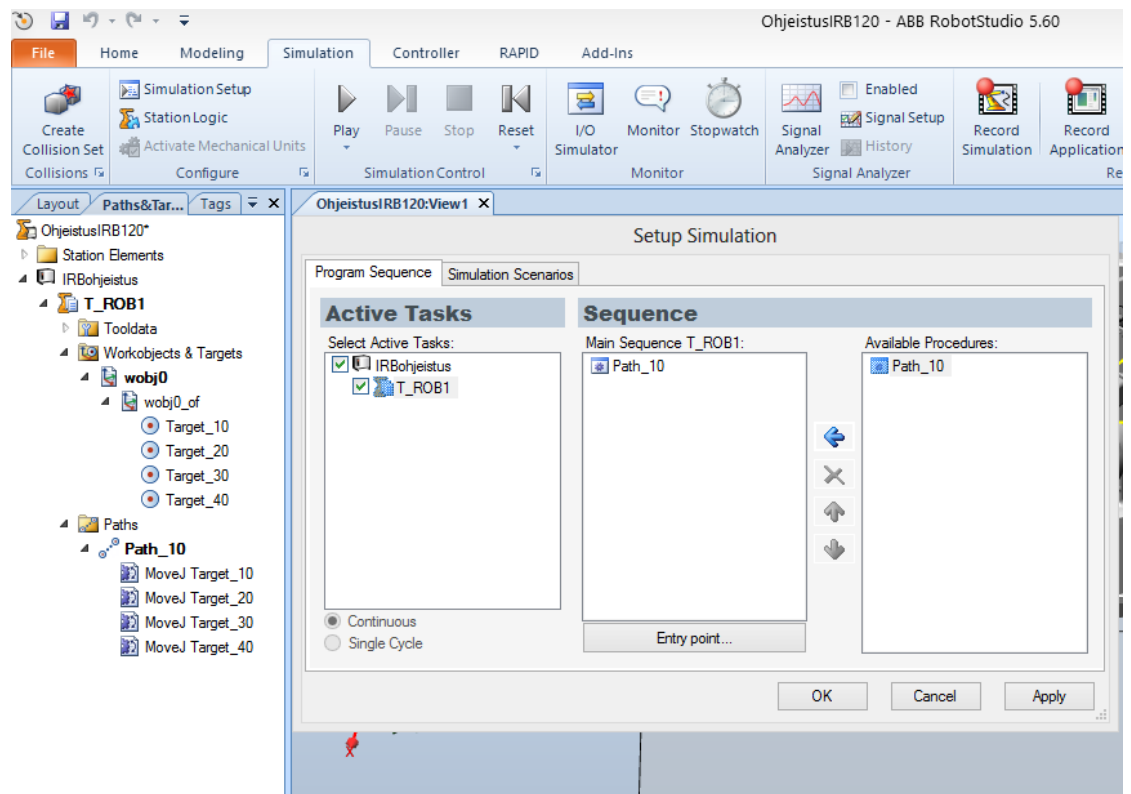
**KUVA 67. Luotu reitti**



**KUVA 68. Luodun reitin RAPID -koodi**

Rapid -valikossa rapid -koodia voidaan kirjoittaa ja muokata aivan vapaasti. Kun koodia on muokattu, painetaan rapid -valikon painiketta format, jolloin ohjelma tasaa koodirivit oikeanlaisiksi. Kun painikkeesta apply valitaan apply to all, päivitetään kaikkiin muokattuihin moduuleihin tehdyt muutokset. Uusia moduuleita voidaan lisätä ohjelmaan rapid -valikossa ollessa klikkaamalla taskin nimeä eli T\_ROB1 oikealla hiiren painikkeella ja valitsemalla new module.

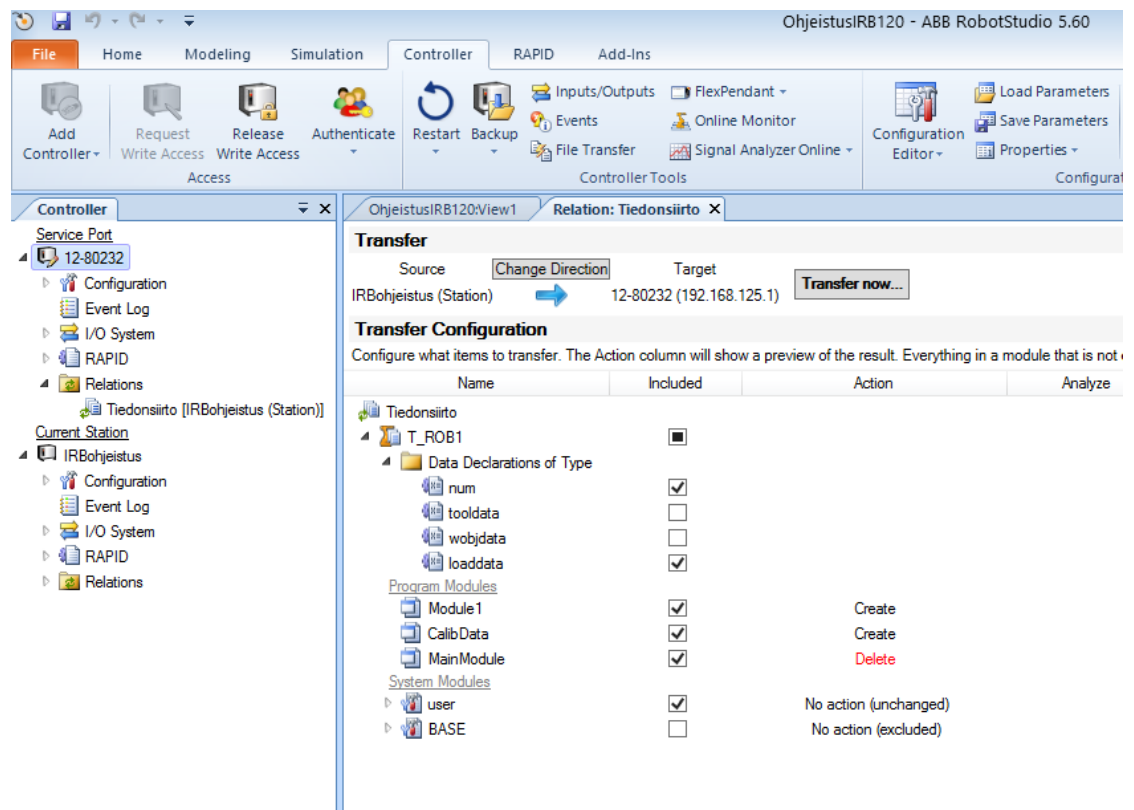
Luotua reittiä voidaan simuloida klikkaamalla luodun reitin nimeä ja menemällä valikkoon modify. Sieltä valitaan kohta move along path. Reitin nimeä voidaan myös klikata hiiren oikealla painikkeella ja sieltä valita sama toiminto. Toinen tapa simuloida robotin toimintaa on simulation -valikossa. Sieltä valitaan simulation setup, valitaan, onko simulointi yhdellä kierrolla vai jatkuvalla kierrolla, ja valitaan reitit/rutiinit, joita simuloidaan. Kun reitit on valittu ja painetaan ok, luo ohjelma rapid -koodiin main -rutiinin, johon valituista rutiineista luodaan siirtymäkäskeyt. Simulation -valikon play -painikkeella voidaan sitten laittaa rapid -koodissa oleva ohjelma päälle ja simuloida sen liikkeitä. Kuvassa 69 on simulation -valikon reittien valinta.



**KUVA 69. Simuloitavien reittien/rutiinien valinta**

### 7.15.2 Luodun ohjelman siirto kontrolleriin robotstudioissa

Jotta ohjelma voidaan siirtää oikeaan kontrolleriin, täytyy tietokoneen olla yhdistettynä ethernetkaapelilla kontrollerin service porttiin. Robotstudioissa mennään controller-valikkoon ja vasemmasta laidasta valitaan add controller, jolloin pudotusvalikkoon ilmestyy oikean järjestelmän kontrolleri, joka nimeltään 12-80232. Kun kontrolleri on lisätty, luodaan virtuaalisen ja oikean kontrollerin välille yhteys painamalla controller-valikon oikeassa laidassa olevaa create relation painiketta. Tämän jälkeen aukeaa ikkuna, jossa annetaan yhteydelle nimi, valitaan ensimmäinen kontrolleri eli virtuaalinen, joka on se, josta siirretään. Toiseksi kontrolleriksi valitaan oikea kontrolleri, johon siirretään. Kun painetaan ok, on siirtoyhteys luotu ja aukeaa tiedonsiirtoikkuna, josta voidaan valita, mitä ohjelman osia siirretään. Kuvassa 70 on tiedonsiirtoikkuna.



**KUVA 70. Tiedonsiirtoikkuna**

Oikealta kontrollerilta on vielä pyydettävä kirjoituslupa ennen, kuin tietoja voidaan ruveta siirtämään. Tämä pyydetään klikkaamalla controller-valikossa olevaa request write access painiketta. Tällöin pendantin näytölle ilmestyy viesti, jossa kysytään, halutaanko kirjoitusoikeus myöntää. Kun painetaan pendantin näytöstä myönnä, voidaan tietoja siirtää painamalla robotstudion tiedonsiirtoikkunasta transfer now. Jos

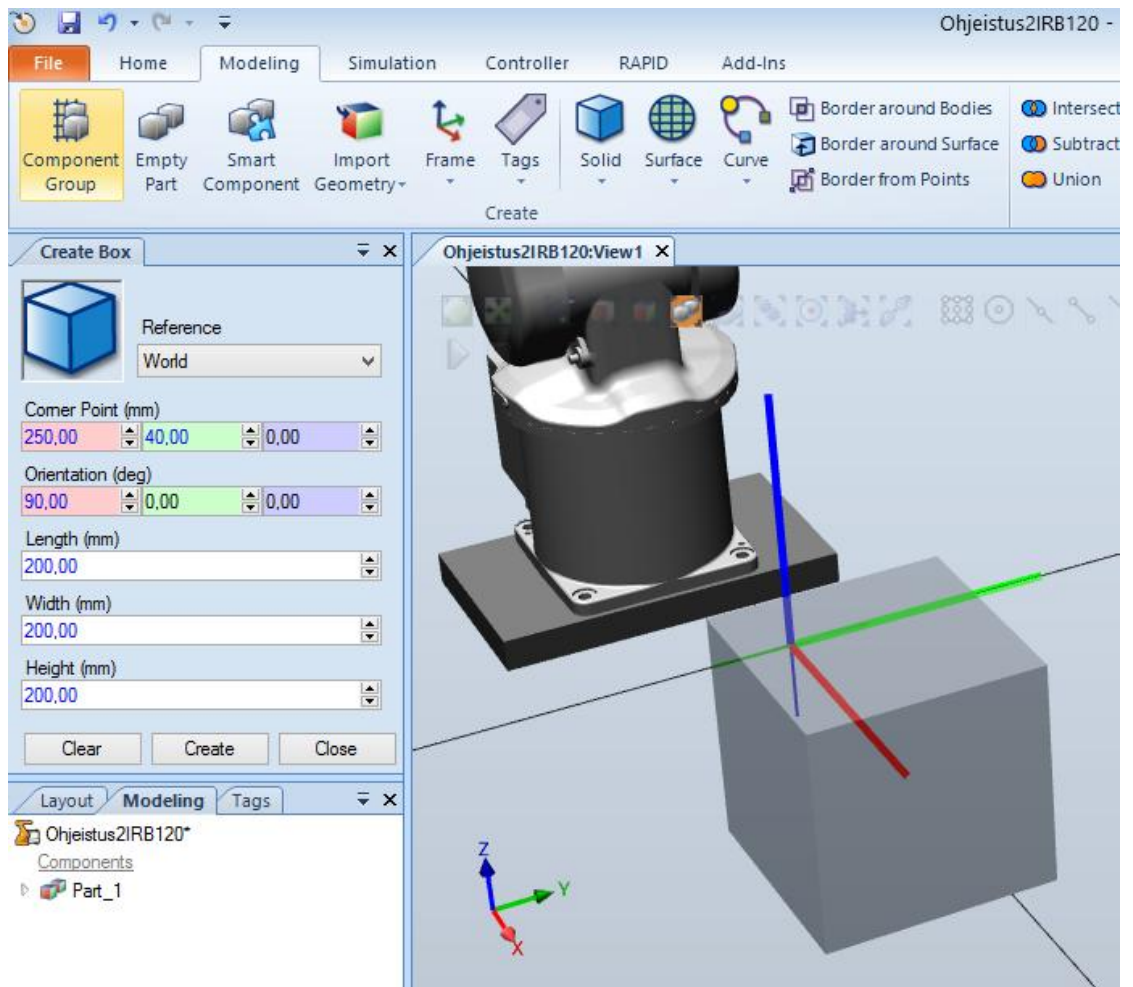
tiedonsiirtoikkuna suljetaan välillä robotstudiosta, saadaan se uudelleen auki tupla-klikkaamalla controller -valikon pudotusvalikosta jommankumman kontrollerin relations -kohdan alla olevasta yhteyden nimestä. Oikeassa kontrollerissa kannattaa olla tyhjä ohjelma, kun robotstudiosta siirretään ohjelma, jotta ohjelmien osat eivät ole samassa ohjelmassa sekaisin ja samalla myös vältetään ristiriidoilta.

### **7.15.3 Työkohteen luominen**

Työkohteen luominen ja sen käyttäminen robotin ohjelmoinnissa on kätevä tapa, jos esimerkiksi työkappaleen paikkaa tarvitsee siirtää työasemassa. Tällöin kappaleeseen liitetty työkohde ja siinä olevat robotin paikkakäskyt ja toiminnot siirtyvät kappaleen mukana ja siihen tehty ohjelma toimii paikan päivytyksen jälkeen uudessa paikassa samalla tavalla kuin työkappaleen alkuperäisessäkin paikassa. Tällöin samalle työkappaleelle ei tarvitse tehdä samaa ohjelmaa uudelleen sen uudessa sijainnissa.

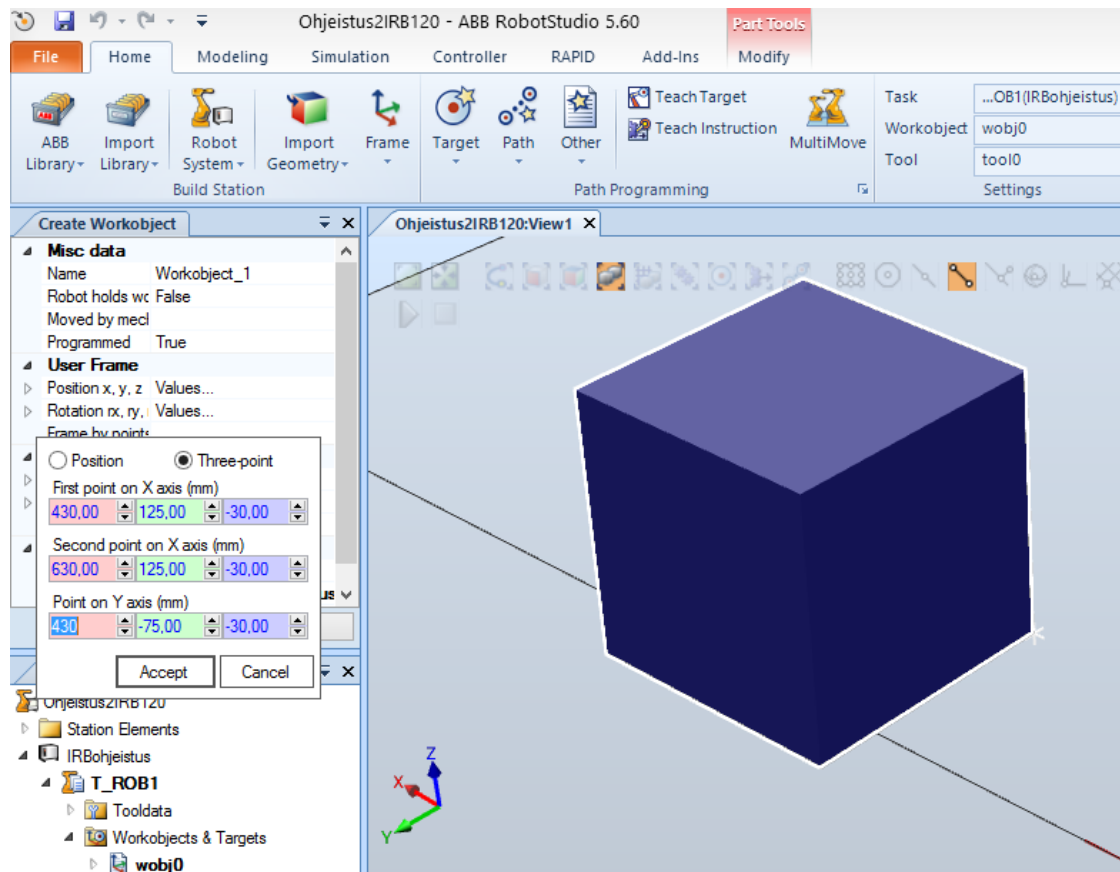
Työkohde luodaan menemällä home -valikkoon ja valitsemalla kohdasta other, create workobject. Pudotusvalikkoon aukeaa työkohteen tietojen määrittämisvalikko. Siinä annetaan työkohteelle haluttu nimi ja määritetään sen koordinaatisto käyttäjän koordinaatistossa ja työkohteen koordinaatistossa. Nämä koordinaatistot voidaan määrittää samalla tavalla. Aloitetaan aluksi luomalla työasemaan jokin kappale, esimerkiksi kuutio ennen työkohteen luomista. Kuvassa 71 on kuution luominen työasemaan.





**KUVA 71. Kuution luominen työasemaan**

Koska kuutio luodaan suhteessa maailman koordinaatistoon, eli robotstudiossa työaseman keskipisteeseen, käännetään kuutiota sitä luodessa x akselin suhteen 90 astetta, joka helpottaa työkohteen ja siihen luotujen paikkakäskeyjen saamista oikealle kohdalle kuutiota, kun ne liitetään kuutioon. Kuution luomisen jälkeen luodaan työkohde. Työkohteen koordinaatistot luodaan käyttäen kolmea pistettä, jotka valitaan kuution kulmista. Käyttäjän ja työkohteen koordinaatistoa määritettäessä molemmissa valitaan kaksi pistettä x-akselilta ja yksi piste y-akselilta. Kuvassa 72 on työkohteen koordinaatiston määrittäminen.

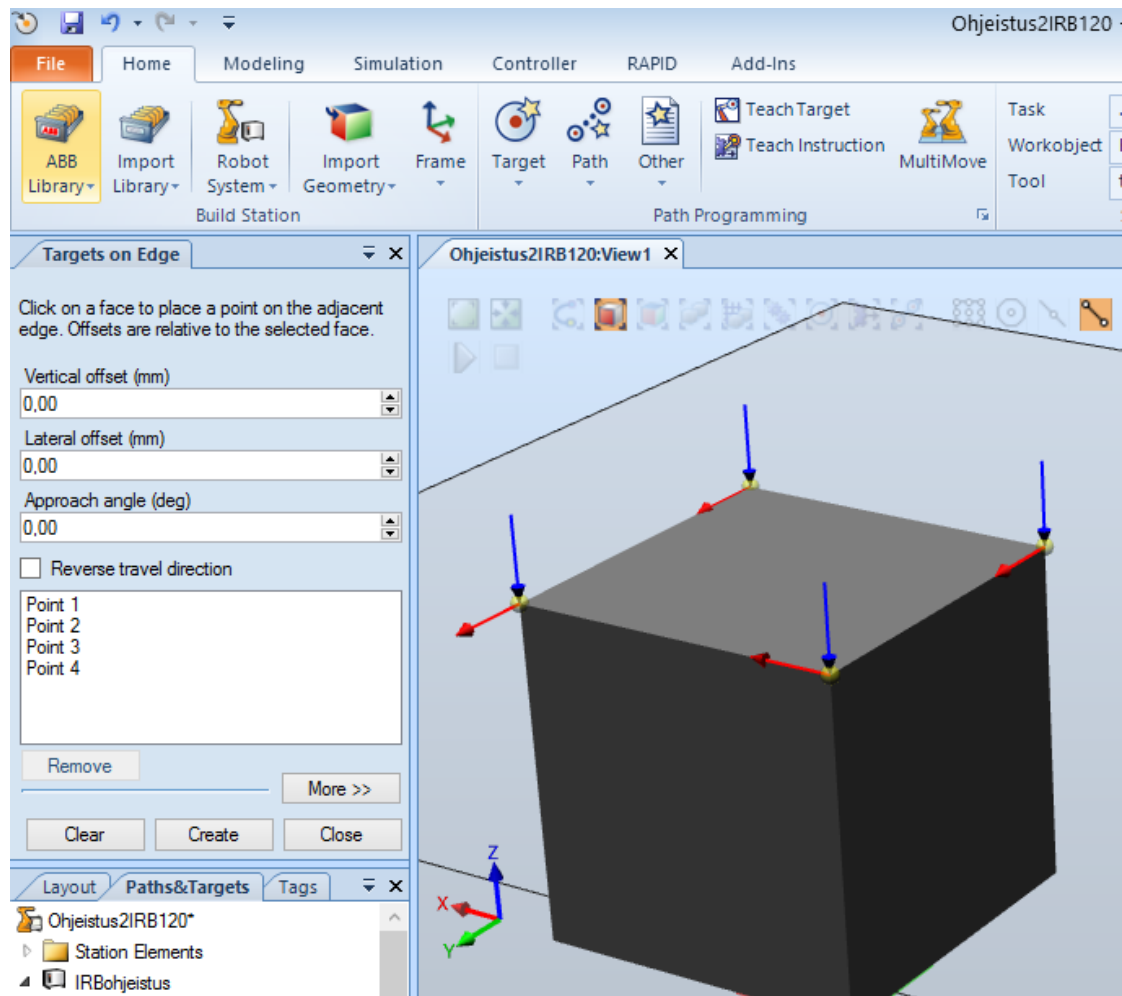


**KUVA 72. Työkohteen koordinaatiston määrittäminen**

Kun työkohteen koordinaatistot on määritetty ja työkohde luotu painamalla create -painiketta, ilmestyy pudotusvalikkoon nimetyn mukainen työkohde ja sen koordinaatisto näkyy työasemassa. Jos robotin paikkakäskyt halutaan luoda tähän työkohteeseen, täytyy sen olla valittuna kuvan 72 oikeassa ylälaudassa olevassa workobject -kohdassa. Liitetään robottiin vielä työkalu menemällä home -valikon import libraryyn ja valitsemalla equipment -kohdasta my tool. Työkalu ilmestyy työasemaan, ja se voidaan liittää robottiin klikkaamalla työkalun nimeä home -valikon layout -kohdassa hiiren oikealla painikkeella pudotusvalikossa valitsemalla attach to sekä valitsemalla robotin nimi.

Seuraavaksi tehdään luotuun työkohteeseen robotille paikkakäskyjä kulkemaan kuution ylälaitojen ympäri. Nyt käskyjä ei opeteta robotille ohjaamalla robotti oikeille paikoille, vaan pisteet merkataan suoraan kuutioon valitsemalla home -valikosta target ja create targets on edge. Erialaisten valintojen tekemistä helpottaa valitsemalla sopiva snap mode, jonka voi valita kuvassa 72 kuution yläpuolella olevasta läpinäkyvästä valikosta tai ohjelman alalaidassa olevista valikoista. Kun pisteitä lisätään, näkyy pisteessä x- ja z-akselien suunnat, eli suunnat, jossa robotin kärki menee kyseiseen pisteeseen.

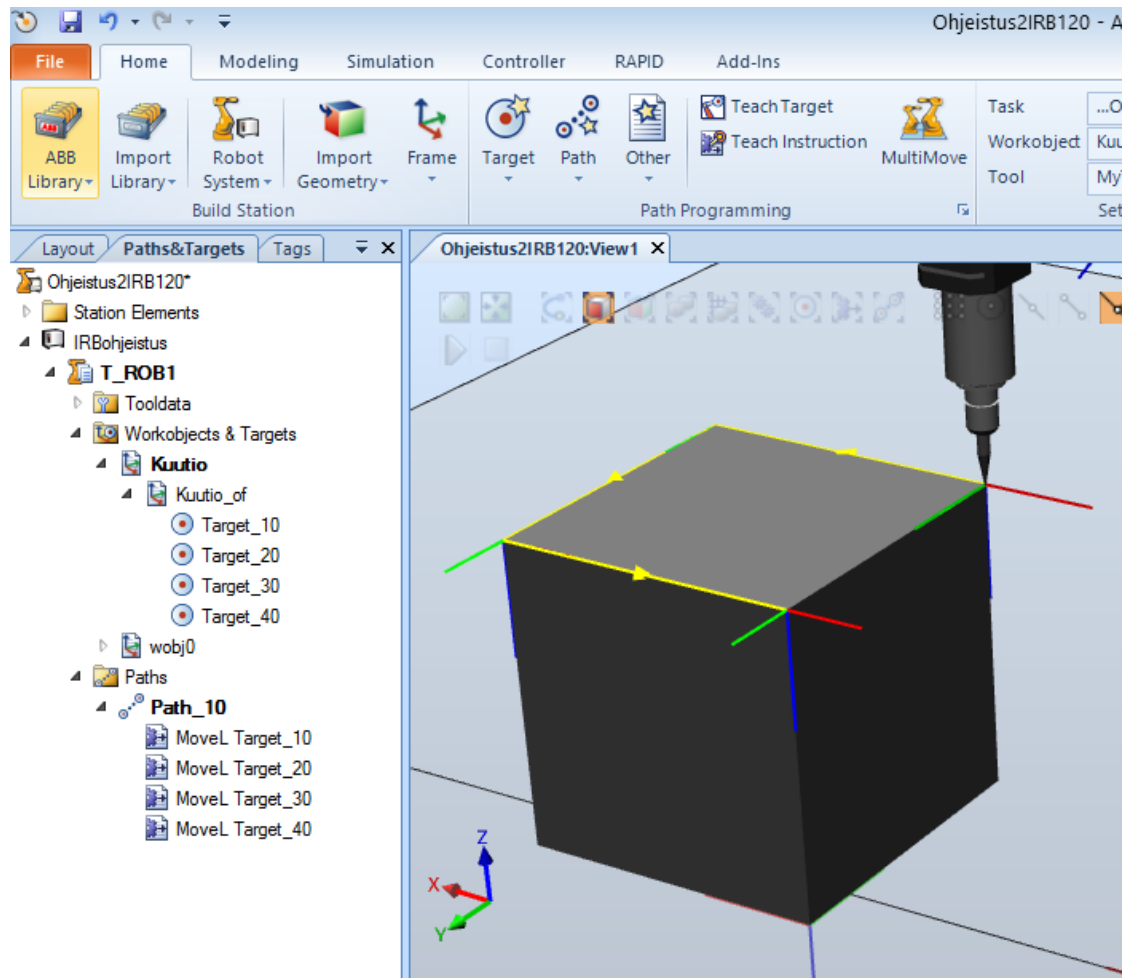
teeseen. Kun valittuna on oikea työkohde, kääntyy z akseli oikeinpäin sen mukaisesti. Näitä pisteitä kannattaa säätää, kun ne on luotu painamalla create -painiketta, jotta robotin liikkeistä saadaan mahdollisimman helppoja ja sulavia. Kuvassa 73 on kuution reunoille merkityt paikat.



**KUVA 73. Kuution ympärille merkityt paikat**

Pisteiden suuntia saadaan säädettyä menemällä halutun pisteen kohdalle pudotusvalikossa, klikkaamalla sitä oikealla hiiren painikkeella ja valitsemalla modify target -valikosta haluttu toiminto. Valitsemalla useita pisteitä voidaan useampien pisteiden säädöt suorittaa yhdellä kerralla. Jokaiselle pisteelle täytyy määrittää robotin konfiguraatio eli asento, jossa robotti menee kyseiseen pisteeseen. Tämä tehdään valitsemalla piste, menemällä modify -valikkoon ja valitsemalla jump to target, jolloin ilmestyy ikkuna, jossa voidaan valita robotin asento. Toinen tapa on valita jump to target klikkaamalla hiiren oikealla painikkeella pisteestä. Kaikille pisteille kannattaa valita sellainen asento, joka on lähimpänä edellistä asentoa. Kun pisteiden säädöt on tehty, luo-

daan niistä robotille reitti aiempien ohjeiden mukaisesti. Kuvassa 74 on luotu reitti kuution reunoilla.



**KUVA 74. Luotu reitti kuution reunoilla**

Viimeisenä vaiheena tehdään työkohteen liittämisen kuution. Tämä tehdään klikkaamalla työkohteen nimeä oikealla hiiren painikkeella, valitsemalla attach to ja valitsemalla kuutio. Liittämisen seurauksena luotu reitti siirtyy maailmankoordinaatiston kohdalle ja kääntyy sen mukaisesti, mutta se voidaan kääntää takaisin samaan asentoon, missä se oli aiemminkin. Sen paikkakin saadaan kohdalleen kuution nähden siirtämällä se alussa määritettyyn kuution paikkaan. Myös robotin asennot on määritettävä pisteisiin uudestaan, kun työasema liitetään kappaleeseen. Kun työkohde on saatu takaisin paikalleen, liikkuu työkohde kuution mukana, kun kuutiota liikutetaan.

## 8 POHDINTA

Pidän opinnäytetyötä hyvin onnistuneena ottaen huomioon, että minulla ei ollut aiempaa kokemusta teollisuusroboteista. Työn aihetta opiskelemalla robotin oppaista pääsin asioista hyvin perille ja sain käsityksen robotin ohjelmoinnista ja sen ohjelmointikielen (RAPID) lukemisesta ja ymmärtämisestä. Oppimieni asioiden pohjalta loin ohjeita robotin käytöstä, joilla saa helpommin käsityksen järjestelmän toiminnasta ja robotin ohjelmoinnista kuin tutkimalla ohjekirjoja kannesta kanteen. Opinnäytetyön aikataulutus onnistui myös mielestäni hyvin. Rakensin robottisolua sillä välin, kun robotin hankintapäätös oli tehty ja robotti oli laitettu tilaukseen. Kun robotti saatiin koululle, siirryin tutkimaan sen käyttöä ja ohjelmointia pendantilla sekä robotstudio -simulointiohjelmalla.

Ongelmana työtä tehdessä oli alussa robotin ohjelmointikielen lukeminen, koska en heti ymmärtänyt, kuinka ohjelman ja käskyjen rakenne muodostuu. Tutkiessani erilaisia käskyjä ja niiden toimintoja opin pikkuhiljaa ymmärtämään ohjelman ja käskyjen rakennetta. Kun opin ymmärtämään ohjelman ja käskyjen rakenteen, aukesi ohjelmointikielen lukeminen kokonaan uudella tavalla. Kun ohjelman rakenteen oppii ymmärtämään, helpottuu robotin ohjelmointi suuresti. Työssä pyrinkin kertomaan yksityiskohtaisesti, kuinka ohjelma muodostuu, jolloin robotin ohjelmoinnin pystyy sisäistämään paljon helpommin.

Työn lopputuloksena syntyneitä toimivia robottisolua voidaan käyttää tulevaisuudessa sähkötekniikan automaatio-opinnoissa, jolloin opiskelijat saavat käsityksen teollisuusroboteista ja niiden toiminnasta. Robottisolun liikuteltavuuden ansiosta se ei ole myöskään sidottuna ainoastaan yhteen paikkaan, vaan sitä voidaan myös tarvittaessa siirtää eri paikkaan.

## LÄHTEET

1. Aalto-yliopisto. Automaatiokurssin luentokalvot. Verkkodokumentti.  
[https://noppa.aalto.fi/noppa/kurssi/as-0.1502/luennot/AS-0\\_1502\\_2013\\_luentokalvot.pdf](https://noppa.aalto.fi/noppa/kurssi/as-0.1502/luennot/AS-0_1502_2013_luentokalvot.pdf). Luettu 7.1.2014. Päivitetty 7.2.2013
2. ABB. Verkkosivut. <http://new.abb.com/fi>. Luettu 20.3.2014. Päivitetty 20.3.2014
3. ABB. ABB teknologiat: teollisuusrobotit. Verkkodokumentti  
<http://www.abb.fi/cawp/seitp202/eb0f9942b4ed6301c125784d0053bb94.aspx>. Päivitetty 7.1.2014. Luettu 7.1.2014
4. ABB. 10 hyvää syytä robottiautomaatioon. Videoklippii.  
<http://www.abb.at/product/ap/seitp327/4ae61ec5f302b56dc1257949001d7722.aspx>.  
Katsottu 7.1.2014. Päivitetty 2014
5. All on robots. Verkkodokumentti. <http://www.allonrobots.com/>. Luettu 21.1.2014.  
Päivitetty 13.10.2013
6. Chameleon light bars & beacons. Verkkodokumentti. [http://www.flashing-beacons.co.uk/index.php?main\\_page=product\\_info&cPath=491\\_522&products\\_id=2081](http://www.flashing-beacons.co.uk/index.php?main_page=product_info&cPath=491_522&products_id=2081).  
Luettu 13.1.2014 Päivitetty 13.1.2014
7. Direct industry. Verkkodokumentti.  
<http://www.directindustry.com/prod/werma/electronic-multi-sound-sirens-58262-540322.html>. Luettu 13.1.2014. Päivitetty 13.1.2014.
8. Duralabel. OSHA floor marking requirements. Verkkodokumentti.  
<http://www.duralabel.com/articles/osha-floor-marking.php>. Luettu 14.1.2014. Päivitetty maaliskuu 2013.
9. ISO. Verkkodokumentti. <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>.  
Luettu 8.1.2014. Päivitetty 2012

10. Leglani, Giovanni ja Fassi, Irene. Computer Science, Technology and Applications: Robotics: State of the Art and Future Trends. New York. Nova Science Publishers, Inc. 2012.

<http://site.ebrary.com.ezproxy.mikkeliyamk.fi:2048/lib/mikkeli/docDetail.action?docID=10683193&p00=robot%20industry%20manipulator>

11. Lehtinen, Hannu. Robotit. Suomen automaatioseura ry. Verkkodokumentti. [www.automatioseura.fi/index/tiedostot/Robotit.doc](http://www.automatioseura.fi/index/tiedostot/Robotit.doc). Luettu 7.1.2014

12. Lempiäinen, Juha. Teollisuusrobottitilastot 2011. Deltatron Oy. Verkkodokumentti.

[http://www.roboyhd.fi/index.php?option=com\\_docman&task=doc\\_download&gid=161&Itemid=66](http://www.roboyhd.fi/index.php?option=com_docman&task=doc_download&gid=161&Itemid=66). Luettu 8.1.2014. Päivitetty 2012

13. Lempiäinen, Juha. Teollisuusrobottitilastot 2012. Deltatron Oy. Verkkodokumentti.

[http://www.roboyhd.fi/index.php?option=com\\_docman&task=doc\\_download&gid=168&Itemid=66](http://www.roboyhd.fi/index.php?option=com_docman&task=doc_download&gid=168&Itemid=66). Luettu 8.1.2014. Päivitetty 2013

14. Malm, Timo. Robottijärjestelmien uudet turvallisuustekniikat. Verkkodokumentti. <http://www.vtt.fi/inf/julkaisut/muut/2008/RobUudetTurv.pdf>. Luettu 13.1.2014 Päivitetty 17.9.2008

15. My safety sign. Verkkodokumentti. <http://www.mysafetysign.com/Safety-Signs/Hazard-Robot-Area-Do-Not-Enter-Sign/SAF-SKU-S-0196.aspx>. Luettu 14.1.2014 Päivitetty 14.1.2014

16. OSHA. OSHA technical manual (OTM) section IV: chapter 4, industrial robots and robot system safety. Verkkodokumentti.

[https://www.osha.gov/dts/osta/otm/otm\\_iv/otm\\_iv\\_4.html](https://www.osha.gov/dts/osta/otm/otm_iv/otm_iv_4.html). Luettu 10.1.2014. Päivitetty 20.1.1999

17. Robotics online. Robot safety and you: Examples of real-life accidents we can learn from. Verkkodokumentti.

[http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Featured-Articles/Robot-Safety-and-You-Examples-of-Real-Life-Accidents-We-Can-Learn-From/content\\_id/1104](http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Featured-Articles/Robot-Safety-and-You-Examples-of-Real-Life-Accidents-We-Can-Learn-From/content_id/1104). Luettu 13.1.2014. Päivitetty 2.8.2004

18. Robotworx. Advantages and disadvantages of automating with industrial robots Verkkodokumentti. <http://www.robots.com/blog/viewing/advantages-and-disadvantages-of-automating-with-industrial-robots>. Luettu 9.1.2014. Päivitetty 2014

19. Robotworx. Robot safety. Verkkodokumentti <http://www.robots.com/articles/viewing/robot-safety>. Luettu 10.1.2014. Päivitetty 10.1.2014

20. Robotworx. What is a robot cell. Verkkodokumentti. <http://www.robots.com/faq/show/what-is-a-robot-cell>. Luettu 14.1.2014. Päivitetty 14.1.2014

21. Rockwell Automation. Protective measures and complementary equipment. Verkkodokumentti. <http://www.ab.com/en/epub/catalogs/3377539/5866177/3378076/7131359/print.html>. Luettu 16.1.2014. Päivitetty 2014

22. RS. Verkkodokumentti. <http://uk.rs-online.com/web/generalDisplay.html?id=werma>. Luettu 13.1.2014. Päivitetty 13.1.2014

23. SICK. [www.sick.com](http://www.sick.com). Verkkosivusto. Luettu 15.1.2014. Päivitetty 2014

24. Steel guard. Verkkodokumentti. <http://www.steelguardsafety.com/machine-equipment-guarding.htm>. Luettu 15.1.2014 Päivitetty 17.6.2012

25. Teollisuusrobotit kehittyvät huimaa vauhtia. Verkkodokumentti. <http://www.yleistieto.net/teollisuusrobotit-kehittyvat-huimaa-vauhtia/> Luettu 7.1.2014. Päivitetty 28.6.2011

26. Teollisuusrobotti. Verkkodokumentti.



<http://fi.wikipedia.org/wiki/Teollisuusrobotti>. Luettu 7.1.2014. Päivitetty 8.12.2014.

27. TLC. Verkkodokumentti. <http://www.tlc->

[di-](http://www.tlc-di-)

[rect.co.uk/Main\\_Index/Distribution\\_and\\_Switchgear\\_Index/Emergency\\_Stop\\_Buttons](http://www.tlc-di-rect.co.uk/Main_Index/Distribution_and_Switchgear_Index/Emergency_Stop_Buttons)

[/](http://www.tlc-di-rect.co.uk/Main_Index/Distribution_and_Switchgear_Index/Emergency_Stop_Buttons). Luettu 16.1.2014. Päivitetty 28.2.2013

28. Wirecrafters. Machine guarding systems. Verkkodokumentti.

<http://www.wirecrafters.com/products/machine-guarding/machine-guarding-systems/>.

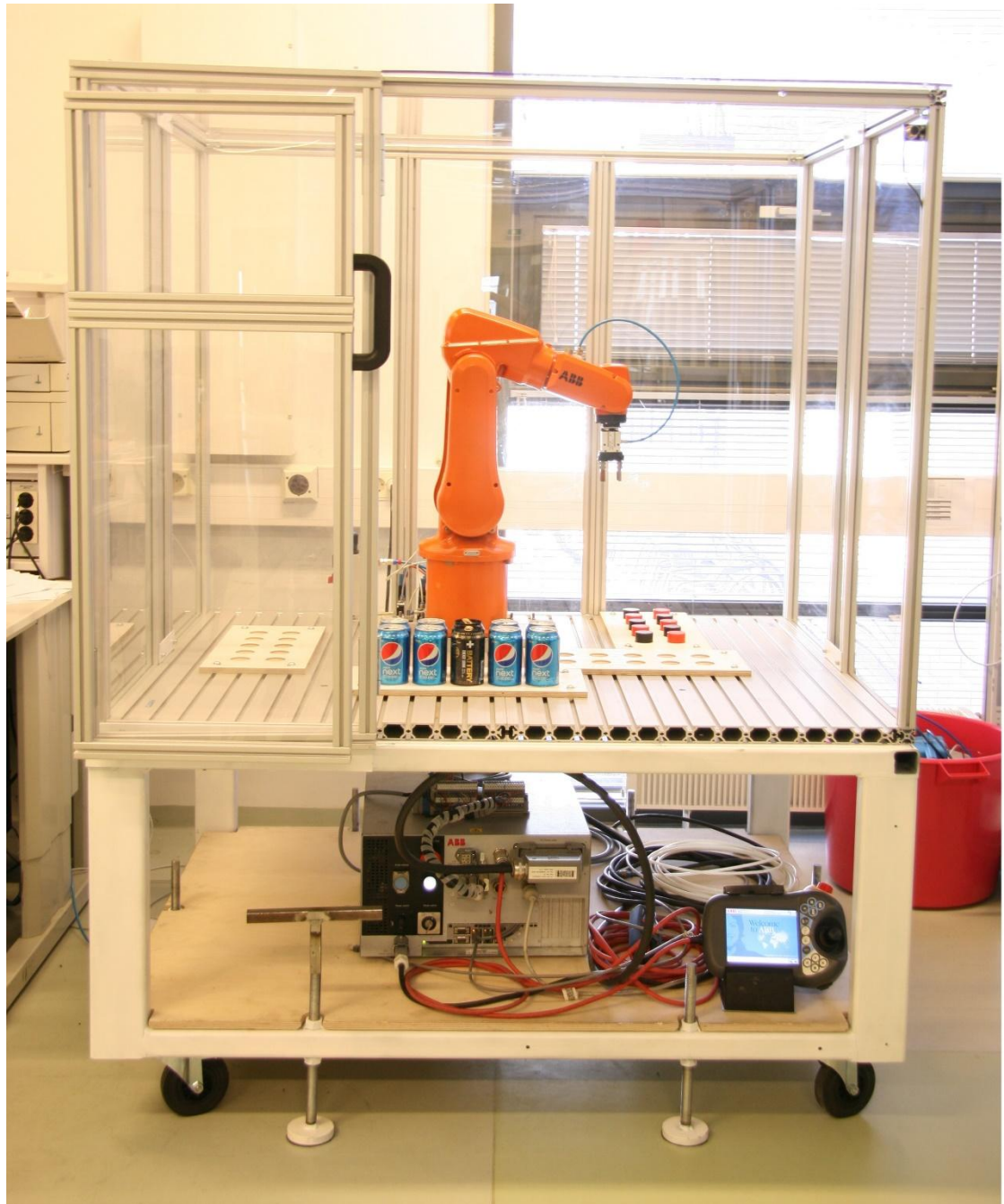
Luettu 15.1.2104. Päivitetty 2014

29. Worksafebc. Verkkodokumentti.

[http://www.worksafebc.com/publications/health\\_and\\_safety/by\\_topic/assets/pdf/bk10](http://www.worksafebc.com/publications/health_and_safety/by_topic/assets/pdf/bk10)

[1\\_chapter3design\\_selection\\_criteria.pdf](http://www.worksafebc.com/publications/health_and_safety/by_topic/assets/pdf/bk10). Luettu 16.1.2014. Päivitetty 2013

Kuva valmiista robottisolusta



Robotin ohjelmoinnin pikaohje:

1. Varmista, että kontrollerin tila -kytkin on opetustilassa eli käännettynä oikealle
2. Avaa ABB valikosta ohjelmaeditori
3. Luo uusi ohjelma menemällä ja taskit ja ohjelmat -valikkoon. Valitse tiedosto ja uusi ohjelma.
3. Liikuta robotti haluamaasi paikkaan ja asentoon. Robotti saadaan liikkumaan painamalla pendantissa oleva turvakytkin puoleen väliin ja kääntämällä ohjaus-sauvasta. Liiketapa ja työkalun asento voidaan valita liikuteltavien akselien valintapainikkeella ja liiketavan valintapainikkeella.
4. Lisää haluamasi liikekäsky lisää käsky valikosta. Käytä käskyä MoveJ tai MoveL. MoveJ -käskyllä robotti voi liikkua mihin asentoon tahansa, koska tällä käskyllä se voi kääntää kaikkia akseleita sen mukaan, mitä määrätty paikka vaatii. MoveL -käskyn käyttö sen sijaan on rajallisempi, koska sitä käytetään suoraviivaisissa liikkeissä määrättyjen paikkojen ja asentojen välillä. Liikekäskyjen argumenteilla on seuraavat merkitykset. Tähti on määritetty paikka ja se näkyy tähtenä, kun sille ei ole määritetty muuta arvoa. Kirjain v ja sen perässä oleva numero tarkoittaa nopeutta mm/s. Kirjain z ja sen perässä oleva numero tarkoittaa, kuinka paljon robotti oikaisee ennen määritettyä pistettä ja kuinka paljon se tulee määrätyn pisteen toiselle puolelle. Kun jokin näistä argumenteista on valittu, tulee robotin liikkeistä sulavampia, koska se ei kulje suoraan pisteeseen vaan kulkee sen ohi pyöristäen liikeratansa. Jos tämän argumentin paikalla on teksti fine, käy robotti määrättyssä pisteessä.
5. Liikuta robotti toiseen haluamaasi paikkaan ja asentoon.
6. Lisää toinen liikekäsky toiselle paikalle ensimmäisen paikan tapaan.
7. Tee haluamasi määrä käskyjä robotille edellisten ohjeiden mukaisesti.
8. Luotua ohjelmaa voidaan kokeilla opetustilassa sulkemalla robottisolun ovi, valitsemalla virheenkorjaus -valikosta SO päärutiniin, jolloin ohjelman osoitin siirtyy main rutiinin ensimmäiselle käskylle. Tämän jälkeen painetaan pendantin turvakytkin puoleen väliin ja painetaan pendantissa olevaa käynnistä -painiketta. Robotin nopeus ei ole opetustilassa missään vaiheessa yli 250mm/s.
9. Ohjelma saadaan pysähtymään painamalla pendantin pysäytä -painiketta, painamalla turvakytkin kokonaan pohjaan tai päästämällä siitä irti.
10. Automaattitilassa robotin ohjelmaa voidaan kokeilla suuremmilla nopeuksilla riippuen käskyssä määritetystä nopeudesta. Jotta ohjelmaa voidaan ajaa automaattitilassa,

## **LIITE 2(2).**

### **Monisivuinen liite**

täytyy kontrollerin tila-kytkin olla käännettynä automaattitilaan, eli vasemmalle ja solun oven olla suljettuna. Myös robotin moottorit täytyy laittaa päälle moottorin käynnistä painikkeesta. Kun nämä asiat ovat kunnossa, voidaan ohjelma käynnistää painamalla pendantin käynnistä -painiketta ja pysäyttää painamalla pysäytä -painiketta.