



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Alexi Kyllönen

ROBOTSTUDION KÄYTTÖ OPETUK- SESSA

Tekniikka ja liikenne
2015

VAASAN AMMATTIKORKEAKOULU
Konetekniikka

TIIVISTELMÄ

| | |
|--------------------|---------------------------------|
| Tekijä | Aleksi Kyllönen |
| Opinnäytetyön nimi | RobotStudion käyttö opetuksessa |
| Vuosi | 2015 |
| Kieli | suomi |
| Sivumäärä | 45 |
| Ohjaaja | Mika Billing |

Tämä opinnäytetyö on tehty Vaasan ammattikorkeakoulun konetekniikan opetusyksikölle. Työn yhteyshenkilönä ja ohjaajana toimi lehtori Mika Billing. Työn tarkoituksena on luoda käyttäjälle perustiedot RobotStudion käytöstä. RobotStudio-ohjelmalla luodaan ohjelmia robottisoluille ja niitä voidaan simuloida offline-tilassa. Offline-ohjelmointi eli etäohjelmointi on yleistynyt huomattavasti teollisuudessa 2000-luvulla. Etäohjelmoinnin etuna on mahdollisuus toteuttaa ohjelmointi tuotantoa pysäyttämättä. Robottia ei myöskään tarvitse pysäyttää uuden ohjelmoinnin aikana.

Opinnäytetyö alkaa teoriaosuuden esittelyllä. Tämän jälkeen esitellään tarvittavat perustiedot ohjelman käytön aloittamisesta sekä navigoinnin helpottamiseksi. Tästä lähdetään rakentamaan pohjaa robotin luomiseen, liikutteluun, työkalujen lisäämiseen, pisteiden luomiseen ja pisteiden yhdistämiseen poluksi. Työhön kuuluu myös ympäristön luomisen opettelua sekä polun luonti geometriselle kappaleelle. Opinnäytetyössä esitellään Virtual Flexpendantin käyttöä sekä simulointia. Opinnäytetyön lopussa opetetaan ohjelman ajo virtuaalirobotille ja ohjelman editointi.

VAASAN AMMATTIKORKEAKOULU
VAASA UNIVERSITY OF APPLIED SCIENCES
Mechanical Engineering

ABSTRACT

| | |
|--------------------|--|
| Author | Aleksi Kyllönen |
| Title | The usage of RobotStudio in teaching environment |
| Year | 2015 |
| Language | Finnish |
| Pages | 45 |
| Name of Supervisor | Mika Billing |

This thesis contains instructions of the usage of RobotStudio in teaching environment. ABB has developed RobotStudio for ABB robots for programming in virtual environment. This thesis is a brief version of getting started in RobotStudio.

The basic theory of robotics was studied first and then the manual for RobotStudio was created. The material for the instructions was obtained by studying various books and videos. The most important instructions were collected and they are included in this thesis.

The result of the thesis is instructions that teaches the user how to navigate and move in RobotStudio and how to create a basic program with RobotStudio. It also shows the simulation work and how to edit the RAPID code.

Keywords Robotics, Offline programming, RobotStudio

SISÄLLYS

KUVA- JA TAULUKKOLUETTELO

| | | |
|-------|---|----|
| 1 | JOHDANTO..... | 8 |
| 1.1 | Vaasan ammattikorkeakoulu Oy..... | 8 |
| 1.2 | RobotStudio | 9 |
| 1.3 | Työn tausta..... | 9 |
| 1.4 | Työn tavoitteet | 10 |
| 1.5 | Työn rakenne | 10 |
| 2 | ROBOTIIKKA | 11 |
| 2.1 | Robottityypit | 11 |
| 2.1.1 | Portaalirobotti..... | 11 |
| 2.1.2 | Scara-robotti..... | 12 |
| 2.1.3 | Nivelvarsirobotti | 12 |
| 2.1.4 | Muut robotit | 12 |
| 2.2 | Robotin koordinaatistot..... | 12 |
| 2.3 | Robotin ohjelmointi | 13 |
| 2.3.1 | ABB- teollisuusrobotin ohjelmointi..... | 14 |
| 3 | ROBOTSTUDIO-OHJELMAN KÄYTTÖ..... | 17 |
| 3.1 | RobotStudion käytön aloitus..... | 17 |
| 3.2 | Työkalun liittäminen robottiin | 21 |
| 3.3 | Pisteiden opettaminen robotille..... | 22 |
| 3.4 | Pisteiden yhdistäminen poluksi | 23 |
| 3.5 | Ympäristön luominen..... | 25 |
| 3.6 | Polun luonti geometrialle..... | 27 |
| 4 | VIRTUAL FLEXPENDANT-TYÖKALU | 34 |
| 4.1.1 | Ohjelman luominen..... | 36 |
| 4.1.2 | Ohjelmointi | 36 |
| 5 | ROBOTIN SIMULOINTI..... | 41 |
| 6 | RAPID-KOODIKIELEN MUOKKAAMINEN | 43 |
| 7 | YHTEENVETO | 44 |

LÄHTEET..... 45

KUVA- JA TAULUKKOLUETTELO

| | |
|--|----|
| Kuva 1. RAPID-ohjelmointiesimerkki (Keinänen ym. 2007) | 15 |
| Kuva 2. RAPID-ohjelmointikielen koordinaattien selitykset (Keinänen ym. 2007) | 16 |
| Kuva 3. RobotStudion aloitusnäkyvä. | 18 |
| Kuva 4. Jog Joint -toiminto..... | 19 |
| Kuva 5. Jog Joint -toiminnolla robotin nivelten liikuttaminen. | 20 |
| Kuva 6. Jog Linear -toiminnolla robotin nivelten liikuttaminen..... | 21 |
| Kuva 7. Pisteiden opetus | 23 |
| Kuva 8. Pisteet Paths&Targets-valikossa | 23 |
| Kuva 9. Empty Path-näppäin | 24 |
| Kuva 10. Move Along Path -näppäin..... | 25 |
| Kuva 11. Geometrian tuonti | 26 |
| Kuva 12. Move- ja Rotate-komento..... | 26 |
| Kuva 13. Robottisolu | 27 |
| Kuva 14. Uusi robottisolu | 28 |
| Kuva 15. Surface Selection -toiminto | 29 |
| Kuva 16. Border Around Surface -toiminto..... | 29 |
| Kuva 17. Reference Surface..... | 30 |
| Kuva 18. Polun luonti | 30 |
| Kuva 19. Robotti valitussa pisteessä Target_10 | 31 |
| Kuva 20. Rotate-valikko | 32 |
| Kuva 21. Jump to Target -valikko..... | 33 |
| Kuva 22. Auto Configuration -toiminto..... | 33 |
| Kuva 23. Käsihjain..... | 34 |
| Kuva 24. Käsihjaimen asettaminen käsiäjolle | 35 |
| Kuva 25. Jogging-valikko | 35 |
| Kuva 26. Uuden liikekomennon luominen | 37 |

| | |
|---|----|
| Kuva 27. Piste nimeäminen uudelleen | 37 |
| Kuva 28. Piste nimeäminen..... | 38 |
| Kuva 29. Lyhyt perusohjelma. | 39 |
| Kuva 30. Ohjelman ajo. | 39 |
| Kuva 31. Synchronize to VC -toiminto..... | 42 |
| Kuva 32. Simulation Setup -toiminto..... | 42 |
| Kuva 33. RAPID-koodin avaaminen | 43 |

1 JOHDANTO

Opinnäytetyön tarkoitus on luoda materiaali, jota hyödyntämällä opiskelijat voivat saada perustietoa RobotStudiosta. Kun opiskelija on tutustunut työhön, kykenee hän käyttämään RobotStudio-ohjelmaa ilman aikaisempaa kokemusta RobotStudiosta. Työ alkaa lyhyellä robotiikan teoriaosuudella ja siitä siirrytään RobotStudion käytön perusteisiin.

1.1 Vaasan ammattikorkeakoulu Oy

Vaasan ammattikorkeakoulu kouluttaa insinöörejä, tradenomeja, sairaanhoitajia, terveydenhoitajia ja sosionomeja. Tärkeä osa-alue on kansainvälinen osaaminen liike-elämän ja teollisuuden ympäristöissä. Lähialueen työelämän tarpeet vaikuttavat suuresti tarjontaan ja Vaasan ammattikorkeakoulu pyrkii vastaamaan lähialueen yritysten tarpeisiin (VAMK 2014).

Vaasan ammattikorkeakouluun kuuluu kolme koulutusalaüksikköä, jotka ovat liiketalous, sosiaali- ja terveysala sekä tekniikka. Opiskelijoita on 3300, joista aikuisopiskelijoita on 570. Vaihto-opiskelijoita on noin 400 ja Vaasan ammattikorkeakoulusta on vaihdossa useita satoja ympäri maailmaa. Henkilöstöä on noin 230, joista 137 on päätoimista opettajaa (VAMK 2014).

Ammattikorkeakoulun koulutukseen kuuluu 14 koulutusohjelmaa, koulutusta on sekä englannin että suomen kielellä. Koulutuksen kesto on 3,5-4 vuotta, riippuen tutkinnosta. Tutkinnot koostuvat perus- ja ammattiopinnoista, suuntaavista ammattiopinnoista, vapaasti valittavista opinnoista, työharjoittelusta sekä opinnäytetyöstä. Opinnäytetyö pyritään yleensä tekemään yhteistyössä Vaasan seudulla sijaitsevien yritysten kanssa. Myös Vaasan ammattikorkeakoulu tarjoaa mahdollisuuden opinnäytetyön tekemiseen (VAMK 2014).

Konetekniikan koulutusohjelma tarjoaa opiskelijalle monipuolisen mahdollisuuden työskennellä insinöörinä erilaisissa tehtävissä. Koulutusohjelmaan kuuluu nel-

jä pääaluetta: energiateollisuuden 3D-kone ja laitesuunnittelu, energiateollisuuden kansainvälinen projektijohtaminen, projektioinnit sekä kansainvälinen energia-tekniikka ja sen hallinta, jossa opetus on pääpiirteissään englannin kielellä (VAMK 2014).

1.2 RobotStudio

ABB Oy on teollisuuskonserni, jonka toiminta keskittyy sähkövoima- ja automaatiotekniikan alueille. ABB:n palveluksessa on noin 140 000 henkilöä, joista Suomessa noin 5200. ABB Oy:llä on viisi divisioonaa: sähkövoimatuotteet, sähkökäytöt ja kappaleavara-automaatio, pienjännitetuotteet ja prosessiautomaatio (ABB 2015).

ABB on kehittänyt RobotStudion omien robottimallien simulointi- ja offline-ohjelmointiohjelmisto. RobotStudio on tarkoitettu ABB:n omille roboteille ja käytettävä ohjelmointikieli on sama kuin ABB:n roboteissa. Sama ohjelmointikieli tuo sen edun, että robottiohjelma voidaan ohjelmoida etukäteen ja tuotantoa pysäyttämättä (ABB 2014).

1.3 Työn tausta

Päättötyö on tehty Vaasan ammattikorkeakoululle opinnäytetyönä. Työssä kerrotaan ensin robotiikan perusteista, jonka jälkeen perehdytään RobotStudion käyttöön.

Vaasan ammattikorkeakoulun konetekniikan insinöörien opetussuunnitelman yksi tärkeä osa-alue on robotiikka. RobotStudion kaltaisten simulointi- ja etäohjelmointiohjelmistojen käyttö on yleistynyt 2000-luvun aikana teollisuudessa. Etäohjelmointi mahdollistaa uusien ratkaisujen testauksen simuloimalla ennen käyttöönottoa, tällöin robotin toimintaa ei tarvitse keskeyttää uudelleenohjelmoinnin aikana. Yleistymisen takia etäohjelmoinnin perusteet on hyvä hallita, jotta opiskelijoita pystytään pitämään jatkuvasti kehittyvän robotisoinnin kanssa ajan tasalla.

1.4 Työn tavoitteet

Työn tavoitteena on luoda RobotStudio etäohjelmointiohjelmiston avulla oppimisympäristö, jossa robottisolusta tehdään simulointimalli ja etäoppimisympäristö. Työn tarkoitus on saada RobotStudion käyttöympäristö tutuksi, jotta käyttäjä pystyy itsenäisesti etäohjelmoimaan robotin perusasetukset. Näitä perustietoja soveltaen keskitytään RobotStudion käyttöön, eli etäohjelmointiin sekä simulointiin. Ohjeet on tehty sillä oletuksella, ettei käyttäjä tarvitse aikaisempaa kokemusta RobotStudion käytöstä.

1.5 Työn rakenne

Opinnäytetyön ensimmäinen osio sisältää lyhyen esittelyn työnantajasta, RobotStudiosta, työn taustasta sekä sen tavoitteista. Toinen osio käsittää teoriaosuuden, jossa perehdytään robotiikan perusteisiin. Kolmas osio sisältää tarvittavat perustiedot RobotStudion käyttöönotosta sekä tarpeelliset komennot. Siinä osiossa perehdytään myös ohjelman luomiseen etäohjelmointiympäristössä. Neljännessä osiossa esitellään Virtual Flexpendant -työkalun käyttöä ja luodaan lyhyt esimerkkiohjelma. Viides osio on ohjelman muokkaamisen käsittelyä. Seuraava osio sisältää siirtymistä simulointiin. Viimeinen osio on yhteenveto.

2 ROBOTIIKKA

Standardi ISO 8373 määrittää robotin olevan ohjelmoitava, monipuolisesti toimiva mekaaninen laite. Vähintään kolme robotin akseleista on ohjelmoitavissa. Riippuen teollisuusautomaation käyttötarkoituksista, akselit voivat olla joko kiinteästi paikallaan tai liikkuvina (SFS 2014).

Robotteja voidaan käyttää teollisuudessa moneen eri tarkoitukseen, kuten koonpanoon, hitsaukseen, koneistuksen kappaleenkäsittelyyn, ruiskupuristuksen kappaleenkäsittelyyn, paketointiin ja pakkaukseen. Myös vaaralliset tai raskaat työtehtävät voidaan antaa robottien tehtäväksi. Tuotannon muuttuessa robotti on myös vaihdettavissa nopeasti käsittelemään uutta tuotetta. Robotin avulla pystytään saavuttamaan tuotannon laatu sekä poistamaan käyttökatkokset tuotantolinjoilla (Keinänen, T. Kärkkäinen, P., Lähetkangas, M. & Sumujärvi, M. 2007., 259).

2.1 Robottityypit

Teollisuusrobotteja on monia erilaisia ja niiden tehtävät vaihtelevat robottityypin mukaan. Toiset robottityypit liikkuvat tasolla tai suorakulmaisessa koordinaatistossa, kun toiset taas kykenevät liikkumaan nivelien avulla. Tässä on muutama esimerkki yleisimmistä käytetyistä teollisuusroboteista.

2.1.1 Portaalirobotti

Portaalirobotissa eli suorakulmaisessa robotissa on kolme liikkuvaa vapausastetta, jotka liikkuvat suorakulmaisessa koordinaatistossa. Joissain tapauksissa tarttuja on mahdollistettu kääntymään tai kiertymään eri asentoihin. Portaalirobotin rakenteen etuina ovat yksinkertaisuus ja kyky kestää hyvin kuormitusta sekä sen vaihteiluita. Yleisiä käyttökohteita ovat erilaiset lavaus- ja pakkaustehtävät sekä erilaiset työstökoneiden palvelutehtävät (Keinänen ym. 2007, 259).

2.1.2 Scara-robotti

Scara-robotissa on kaksi tai kolme kiertyvää niveltä, jotka ovat samassa tasossa. Scara-robotissa on myös yleensä pystysuunnassa kiertyvä nivel, jolla suoritetaan lineaariliike. Robotti kykenee työskentelemään vain kohtisuoraa XY -tasoa vastaan. Yleisin käyttökohde on erilaiset kokoonpanotehtävät, kuten matkapuhelimet ja muut elektroniset laitteet. Scara-robotti soveltuu hyvin näihin tehtäviin sen tarkkuuden ja nopeuden takia (Keinänen ym. 2007, 259-260).

2.1.3 Nivelvarsirobotti

Nivelvarsirobotti on yleisimmin käytetty teollisuusrobotti. Nivelvarsiroboteilla on yleensä viisi tai kuusi vapaasti ohjelmoitavaa niveltä ja näiden avulla kappale tai työkalu voidaan asettaa kaikkiin mahdollisiin kulmiin, joihin robotti ulottuu. Nivelvarsirobotin pallomainen työskentelyalue ja kyky asemoida robotti mihin tahansa työskentelyasentoon tekevät siitä juostavan ja monikäyttöisen teollisuusrobotin. Nivelvarsirobotin rakenne mahdollistaa laajan kokovalikoiman (1 kg... 500 kg), jolloin löytyy robottimalli lähes kaikkiin käyttötarkoituksiin (Keinänen ym. 2007, 260).

2.1.4 Muut robotit

Yllämainitut robotit ovat tavallisimpia ratkaisuja teollisuusroboteista. Teollisuudessa on käytössä myös monenlaisia erikoisrobotteja, mutta niiden osuus on jäänyt vähäiseksi. Suurin vaikuttava syy on yleisien robottityyppien edullinen valmistaminen suurina sarjoina (Keinänen ym. 2007, 259).

2.2 Robotin koordinaatistot

Teollisuusrobotti liikkuu koordinaattien mukaan, jotka ovat määriteltäviä numeroarvoja XYZ-suunnissa. Yleisesti tunnetut koordinaatistot ovat:

- maailmankoordinaatisto
- peruskoordinaatisto
- työkalukoordinaatisto.

Maailmankoordinaatisto on robotin toimintaympäristöön sidottu koordinaatisto. Toimintaympäristö määräytyy robotin ympäristöstä olevista laitteista ja rakennuksista (Keinänen ym. 2007, 260).

Peruskoordinaatisto on robotin jalustaan sidottu koordinaatisto. Vaakasuoran tason robotin jalustan tasolle muodostavat X- ja Y-akselit. Ensimmäisen nivelen keskikohdan kautta muodostuu Z-akseli (Keinänen ym. 2007, 261).

Työkalukoordinaatisto on suorakulmainen koordinaatisto, joka on sidottu työkalun tai käsiteltävän kappaleen koordinaatistiksi. Liikeratojen ohjelmointia helpottaa se, että koordinaatistojen akseleiden suunta muuttuu kappaleen asennon mukaan. Tämä helpottaa esimerkiksi hitsaus- ja kokoonpanon tehtävissä liikeratojen määrittely (Keinänen ym. 2007, 261).

2.3 Robotin ohjelmointi

Robotin ohjelmointi voidaan jakaa kahteen pääluokkaan: online- ja offline-ohjelmointiin. Online-ohjelmoinnissa robotti ei pysty jatkamaan tuotantoa, vaan se on pysäytettävä ohjelmoinnin ajaksi. Online-ohjelmointiin käytetään robotin käsiohjainta. Offline-ohjelmointi eli etäohjelmointi tehdään ilman robottia ulkoisella tietokoneella. Etäohjelmointi voi olla tekstipohjainen tai mallipohjainen. Tekstipohjainen etäohjelmointi vaatii ohjelmointikielen tuntemista, koska tässä ohjelmointitavassa robottiohjelma kirjoitetaan tekstinä. Mallipohjaisessa etäohjelmoinnissa käytetään 3D-simulointimalleja robotista ja oheislaitteista ohjelman tuottamiseen. Tyypilliset mallipohjaisen ohjelmiston vaiheet ovat: mallintaminen, kalibrointi, ohjelmointi, simulointi, siirto robotille ja testaus (Malm 2008, 95–98).

Etäohjelmoinnissa kalibrointi on tärkeä osa ohjelmointia, jotta robotti ja etäohjelma saadaan vastaamaan toisiaan. Jos robotille ja etäohjelmaan ohjelmoidut pai-

koitukset eivät vastaa toisiaan, on tarkkojen töiden tekeminen erittäin haastavaa. Tämä johtuu siitä, etteivät esimerkiksi työkappaleen koordinaattipisteet ole täsmälleen samassa kohdassa ohjelmassa ja robotissa.

2.3.1 ABB- teollisuusrobotin ohjelmointi

ABB:n roboteissa käytetään ohjauskielenä RAPID-kieltä. Päärutiini (pääohjelma), alirutiini (aliohjelma) ja ohjelmadata muodostavat RAPID-kielen. Pääohjelma aloittaa ohjelman ja kutsuu aliohjelman toimintoja käyttöön. Käsky määrittää toiminnon, joka tapahtuu käskyä suorittaessa, esimerkiksi robotin liikuttaminen. Robotti suorittaa käskyt yksi kerrallaan ohjelmoidussa järjestyksessä (Keinänen ym. 2007, 263).

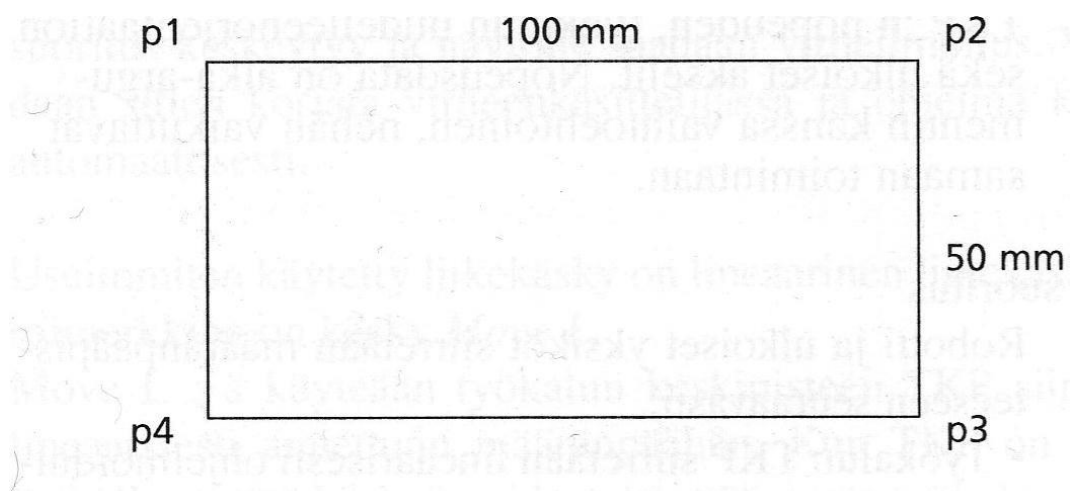
Yksi yleisimmin käytetyistä liikekäskyistä on lineaariliikkeellä tapahtuva MoveL-komento. Kuvassa 1 esimerkkinä on ohjelma, jossa työkalu ohjataan pisteeseen p1 ja sen jälkeen annetaan uuden pisteen koordinaatit p1 pisteen suhteen. Työkalun käytyä kaikissa pisteissä, se palaa pisteeseen p1. Toisinaan on helpompi määrittää piste annetun pisteen suhteen eli offsettina. Liikekäskynä käytetään komentoa MoveL. Tällöin robotille pitää vain opettaa yksi piste ja seuraavat pisteet voidaan antaa offsettina. Ympyräliikkeisiin käytetään MoveC-komentoa ja nivelliikkeissä käytetään MoveJ-komentoa (Keinänen ym. 2007, 264). Kuva 2 selittää RAPID-ohjelmointikielen MoveL-komennon koordinaattien selitykset.

Liikekäskyjen lisäksi robotti voi suorittaa muitakin käskyjä. Erilaisten käskyjen lisääminen saattaa helpottaa ohjelman tekoa, jos kyseessä on vaikka toistosilmukka. Tällöin ohjelma voi suorittaa halutun toiminnon niin kauan kuin silmukka käsklee. Tässä on muutama esimerkki hyödyllisistä komennoista:

- IF-lause, jossa robotti suorittaa silmukkaa niin kauan kunnes ehtolause on tosi.
- while-silmukka, jossa robotti suorittaa silmukkaa niin kauan kuin ehtolause pitää paikkaansa.

- ProcCall-toiminto, joka kutsuu halutun rutiinin, esimerkiksi alirutiinin, joka hioo kappaleen pinnan.
- WaitTime-toiminto, joka odottaa annetun sekuntimäärän.
- Label-toiminto, jolla voidaan nimetä ohjelman osa ja kutsua haluttu osa myöhemmin GOTO-toiminnolla.

Yllä olevat esimerkit ovat vain murto-osa käytettävistä komennoista ja näiden esimerkkien on tarkoitus antaa käsitys siitä, millaisia komentoja robottiohjelmassa voidaan mahdollisesti käyttää liikekäskyjen lisäksi.



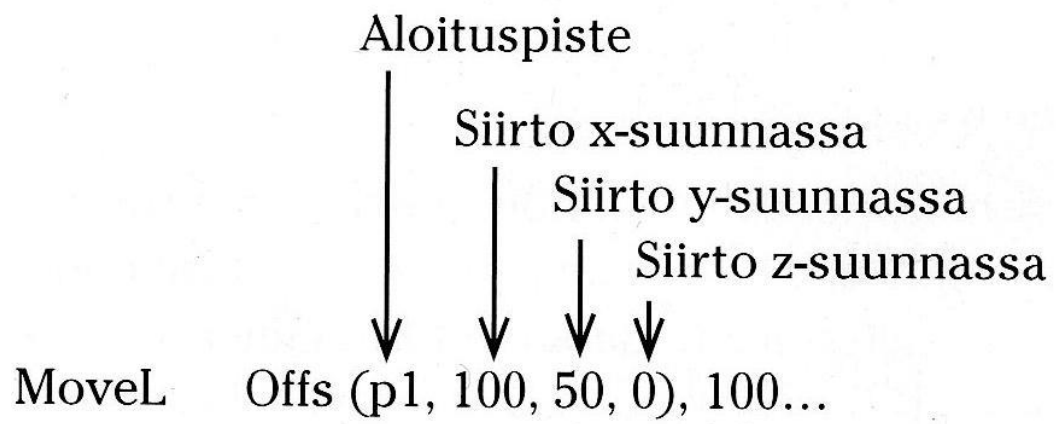
MoveL p1.....

MoveL p2 MoveL offs (p1,100,0,0)...

MoveL p3 MoveL offs (p1,100,50,0)...

MoveL p4 MoveL offs (p1,0,50,0)...

Kuva 1. RAPID-ohjelmointiesimerkki (Keinänen ym. 2007)



Kuva 2. RAPID-ohjelmointikielen koordinaattien selitykset (Keinänen ym. 2007)

3 ROBOTSTUDIO-OHJELMAN KÄYTTÖ

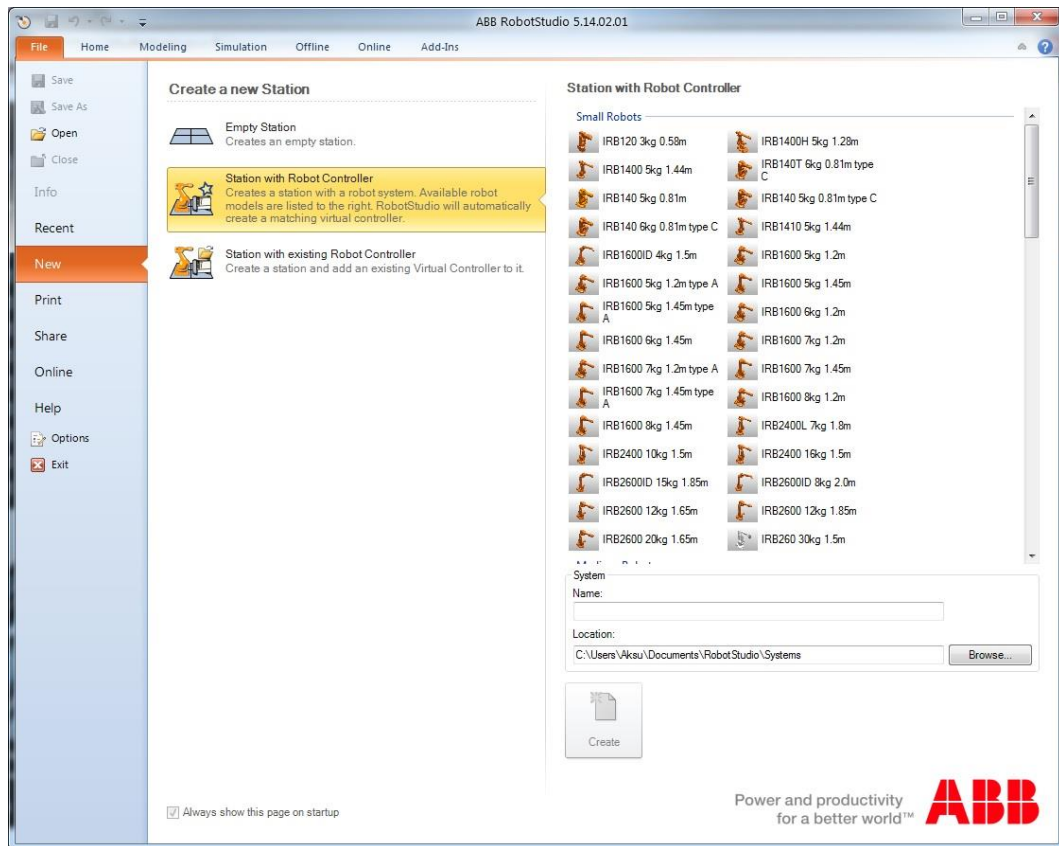
RobotStudio-ohjelman käyttö aloitetaan tutustumalla ohjelmankäytön perusteisiin sekä keskeisiin komentokäskyihin. Kun nämä ovat halussa, perehdytään robotin määrittelyyn ja lopulta opetellaan tekemään lyhyt ohjelma. Kaiken tämän jälkeen käyttäjän on tarkoitus osata itse luoda perusohjelma ja saada etäohjelma toimimaan robotilla.

Ennen ohjelman käytön aloittamista on hyvä tietää muutama hyödyllinen komento navigoinnin helpottamiseksi.

- Kuvakulmaa voidaan säätää pitämällä pohjassa samanaikaisesti CTRL-, SHIFT-näppäintä sekä hiiren vasenta painiketta ja hiirtä liikuttamalla kuvakulma muuttuu.
- Pitämällä Ctrl-näppäintä ja hiirtä liikuttamalla voidaan liikuttaa kuvaa.
- Kuvaa pystytään zoomaamaan pyörittämällä hiiren keskinäppäintä.

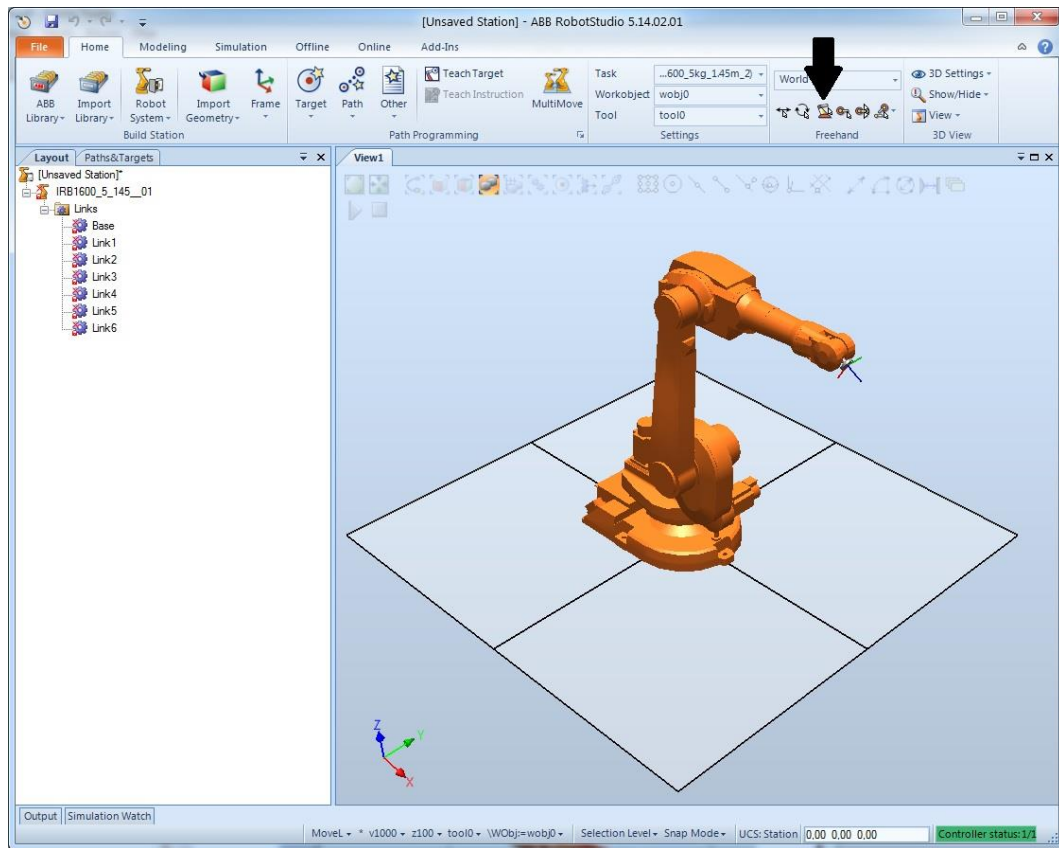
3.1 RobotStudion käytön aloitus

Ohjelman käyttö aloitetaan valitsemalla uusi asema Create a New Station -välilehden alta (**Kuva 3.**). Luodaan uusi asema valitsemalla Station with Robot Controller. Tässä työssä käytettävä robottimalli on IRB1400 5kg 1.44m.

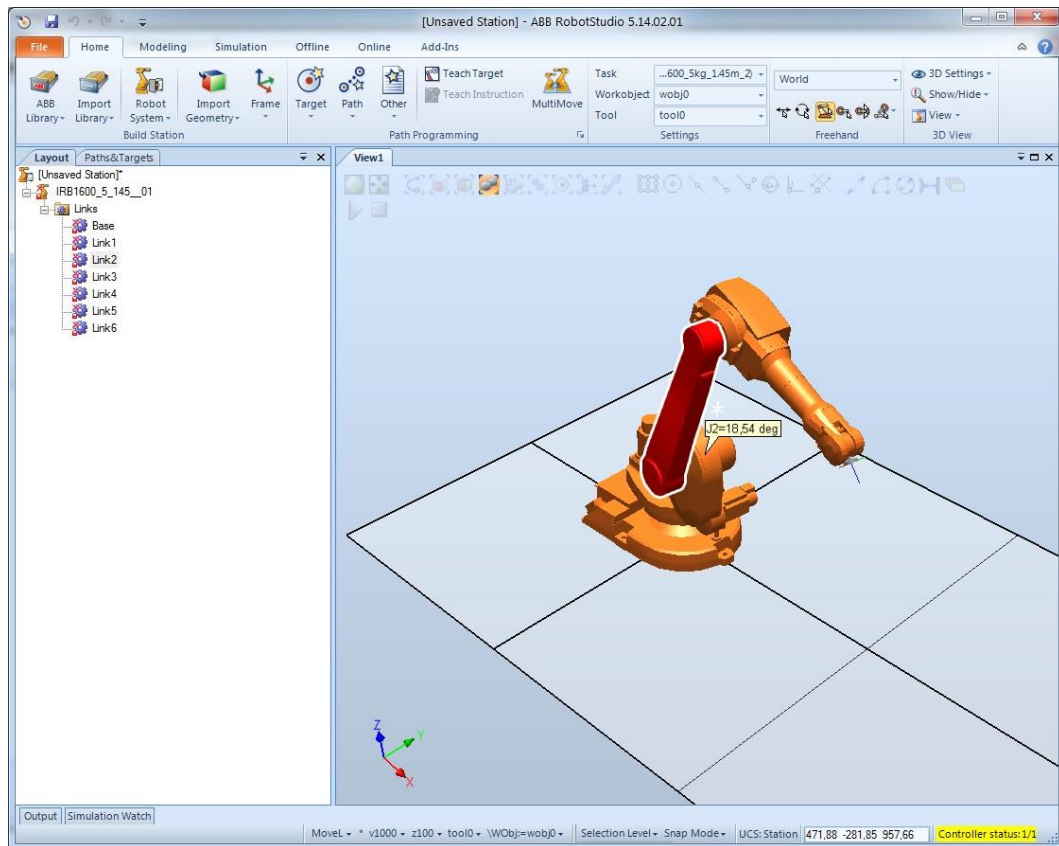


Kuva 3. RobotStudion aloitusnäky.

Robotin niveliä voidaan liikutella Jog Joint -toiminnolla (**Kuva 4.**). Ensin valitaan liikuteltavaksi haluttu nivel, jonka jälkeen niveltä liikutetaan hiiren vasemmalla näppäimellä. Valittu nivel muuttuu punaiseksi ja seuraa hiiren liikkeitä (**Kuva 5.**).

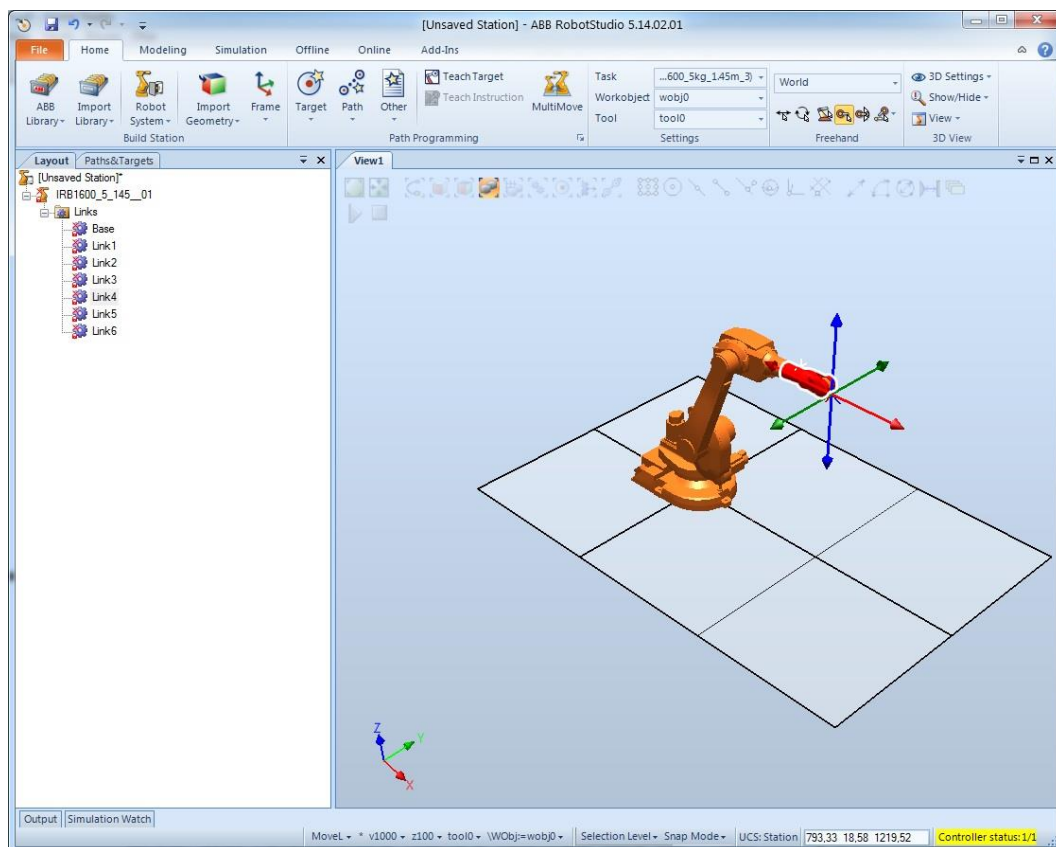


Kuva 4. Jog Joint -toiminto.



Kuva 5. Jog Joint -toiminnolla robotin nivelten liikuttaminen.

Jog Linear -toiminnolla robottia voidaan liikuttaa maailmankoordinaatiston mukaisesti. Kun nivel muuttuu punaiseksi, sitä voi liikuttaa koordinaatistossa (**Kuva 6.**).



Kuva 6. Jog Linear -toiminnolla robotin nivelten liikuttaminen

3.2 Työkalun liittäminen robottiin

Seuraavaksi robottiin liitetään työkalu, joka kiinnitetään robotin laippaan ja seuraa robotin liikkeitä. ABB:n omasta kirjastosta löytyy erilaisia valmiita työkaluja haluttuihin toimintoihin.

Työkalu lisätään valitsemalla Home-välilehdeltä Import Library -painike, sieltä valitaan Equipment ja aukeavasta valikosta Tools- tai Training objects-valikon alta haluttu työkalu. Tässä opinnäytetyössä käytetään Pen-työkalua.

Työkalu ilmestyy Layout-välilehdelle. Sieltä se valitaan hiiren vasemmalla näppäimellä ja siirretään valittu työkalu robottimallin päälle. Näin työkalu voidaan kiinnittää robottiin. Tämän jälkeen ohjelma kysyy, säilytetäänkö työkalun nykyi-

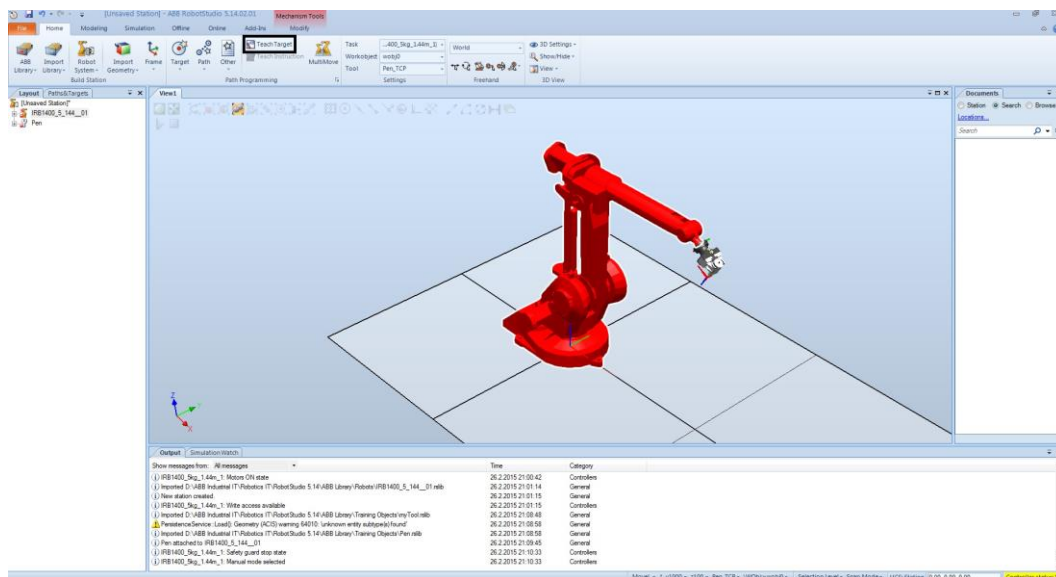
nen asema. Tähän vastataan ei, koska työkalun halutaan seuraavan robotin liikkeitä ja työkalu pitää kiinnittää robotin laippaan.

Toinen tapa on valita työkalu ja painaa hiiren oikeaa painiketta. Aukeavasta valikosta valitaan Attach to -toiminto. Uudesta aukeavasta valikosta valitaan robotti, mihin työkalu halutaan liittää. Ohjelma esittää saman kysymyksen kuin ensimmäisessä toimintatavassa, eli pidetäänkö valitun työkalun nykyinen asema. Tähänkin kysymykseen vastataan ei.

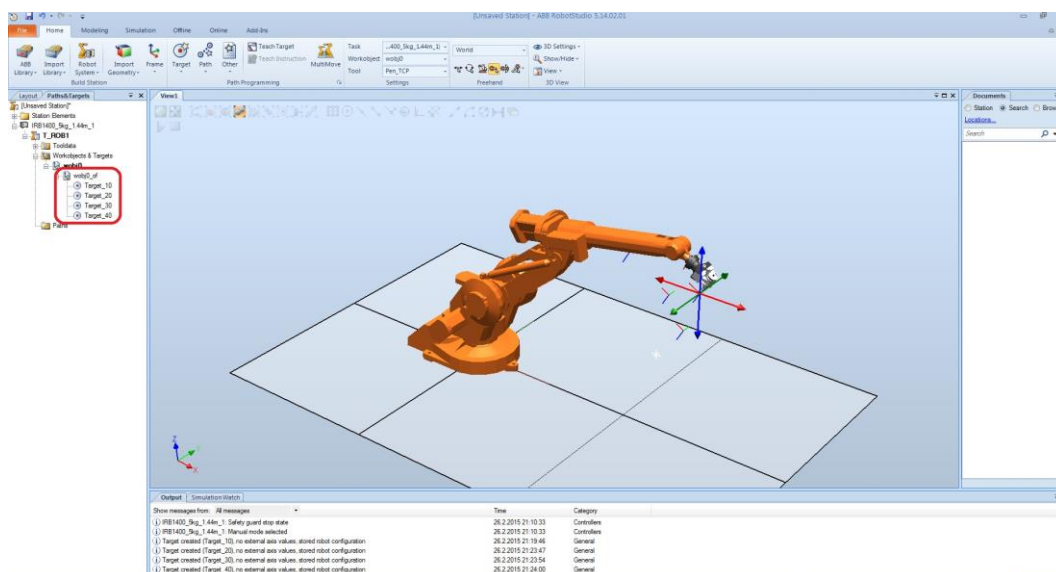
3.3 Pisteiden opettaminen robotille

Tässä luvussa käsitellään pisteiden opettamista robotille. Robotti tarvitsee pisteet polun muodostamiseksi, jotta se tietää mitä kautta polku kulkee. Aluksi luodaan ne pisteet, jotka muodostavat robotille kuljettavan polun.

Seuraavaksi jatketaan aiemmin luodusta robottiasemasta, johon on kiinnitetty Pen-työkalu. Valitaan Home-valikon alta Teach Target -painike (**Kuva 7.**). Ohjelma tallentaa pisteen siihen kohtaan, missä robotin työkalukoordinaatisto on tällä hetkellä. Piste ilmestyy Paths&Targets-valikkoon nimellä Target_10. Nyt liikutetaan robottia haluttuun kohtaan ja opetetaan uusi piste. Tämä onnistuu samalla tavalla kuin ensimmäisen pisteen valinta. Yhteensä luodaan neljä pistettä eri kohtiin. Pisteet ilmestyvät Paths&Targets-alivalikkoon (**Kuva 8.**). Valikkoon ilmestyy kotipiste (Target_10), välipisteet 1 ja 2 (Target_20 ja Target_30) sekä päätepiste (Target_40).



Kuva 7. Pisteiden opetus

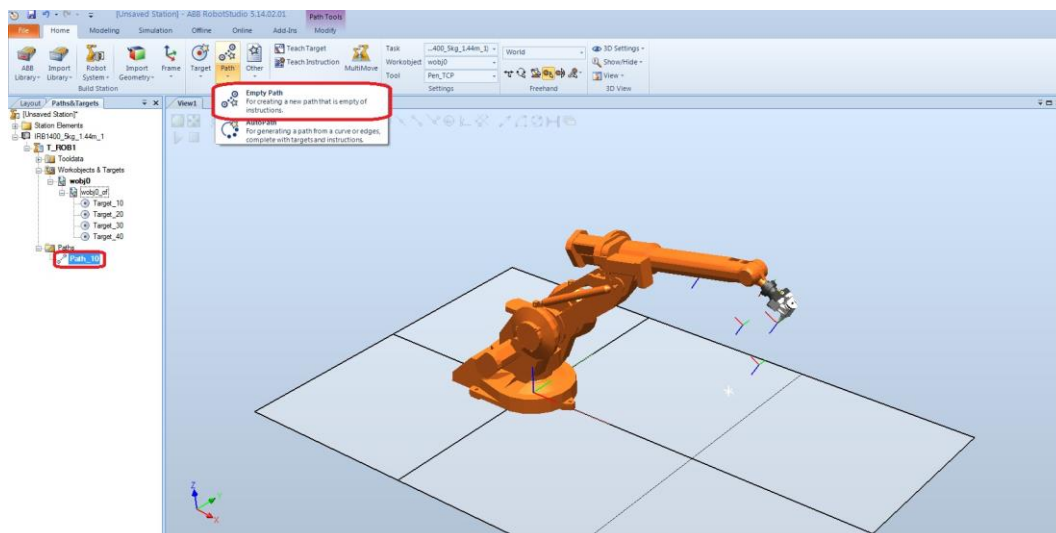


Kuva 8. Pisteet Paths&Targets-valikossa

3.4 Pisteiden yhdistäminen poluksi

Kun pisteet on luotu, voidaan ne yhdistää poluksi. Valitaan Home-valikon alla oleva Empty Path -näppäin (**Kuva 9.**) sekä siirretään jokainen piste yksitellen hii- ren vasemmalla näppäimellä Path_10-hakemistoon. Pisteiden siirtojärjestyksen tulee olla sama kuin järjestys, jossa robotti käy pisteet läpi. Aloitetaan siis raa-

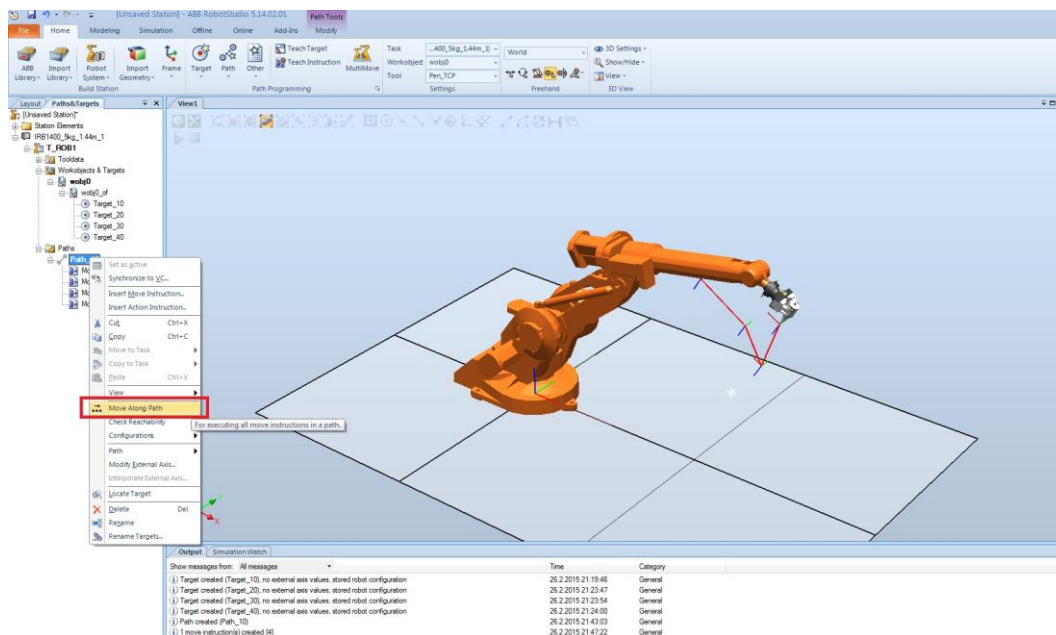
haamalla ensin kotipiste (Target_10), sitten välipisteet 1 ja 2 (Target_20 ja Target_30) sekä viimeiseksi päätepiste (Target_40).



Kuva 9. Empty Path-näppäin

Kun halutut pisteet on lisätty Path_10-hakemistoon, aktivoidaan Path_10 ja klikataan sitä hiiren oikealla näppäimellä. Aukeavasta valikosta valitaan Move Along Path (**Kuva 10.**). Robotti laskee valitun radan ja hetken kuluttua robotti liikkuu valitun polun.

Jos robotti ei ala liikkua valittua polkua, johtuu se todennäköisesti siitä, ettei robotin ulottuvuus riitä tai polku kulkee robotin sisältä. Robotin rajoitukset tulee siis ottaa huomioon. Jos robotti ei yletä annettuun koordinaattipisteeseen, se ei voi liikkua. Robotin liikekäsken muuttaminen lineaariliikkeestä nivelliikkeeksi saattaa auttaa.



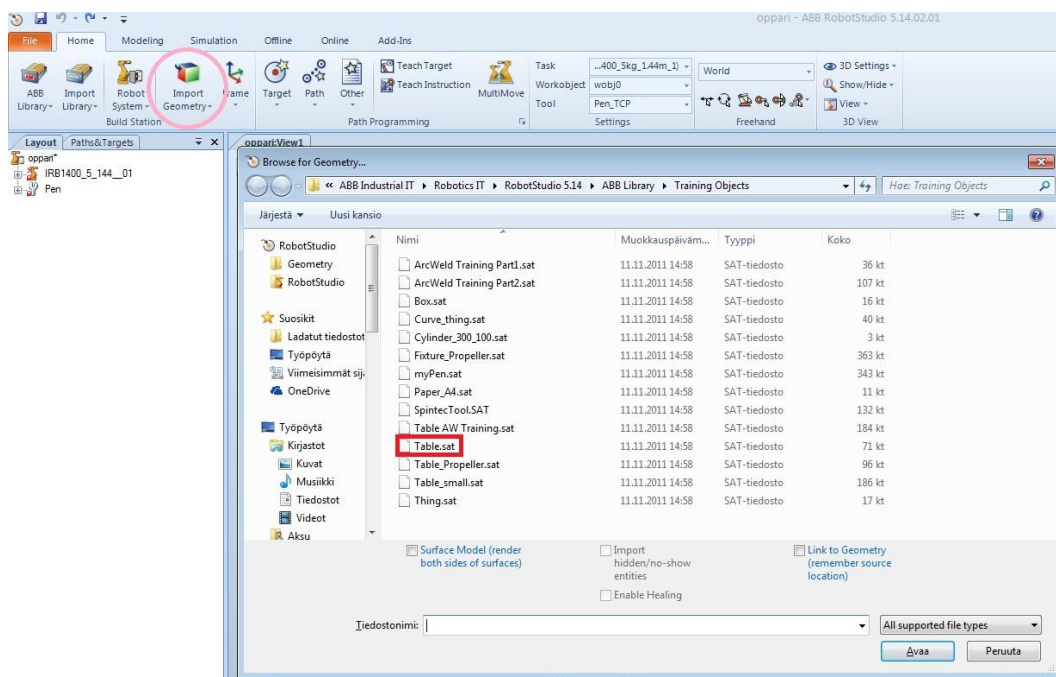
Kuva 10. Move Along Path -näppäin

3.5 Ympäristön luominen

ABB:n kirjastosta löytyy erilaisia valmiiksi mallinnettuja työtasoja ja -kappaleita. ABB Library -valikosta voidaan tarpeen mukaan lisätä esimerkiksi kuljettimia, kaappeja, turva-aitoja sekä työalustoja. Uusia esineitä on mahdollista mallintaa itse lisää 3D-mallinnusohjelmilla, mikäli siihen on tarve.

Käytetään RobotStudio kirjastosta löytyvää valmista työtasoa. Painetaan Home-valikosta löytyvää Import Geometry -näppäintä. Avautuvasta hakemistosta aktiivoidaan esimerkiksi Table.sat-tiedosto ja painetaan Open (**Kuva 11.**). Uusi työtaso ilmestyy robottisoluun.

Lisättyjen esineiden paikkaa voidaan vaihdella Home-valikosta valitsemalla Move- tai Rotate-käsky Freehand-toiminnosta (**Kuva 12.**). Näiden avulla esineet voidaan asetella haluttuun paikkaan niin, että simulointiympäristö kyetään mallintamaan todellista ympäristöä vastaavaksi.

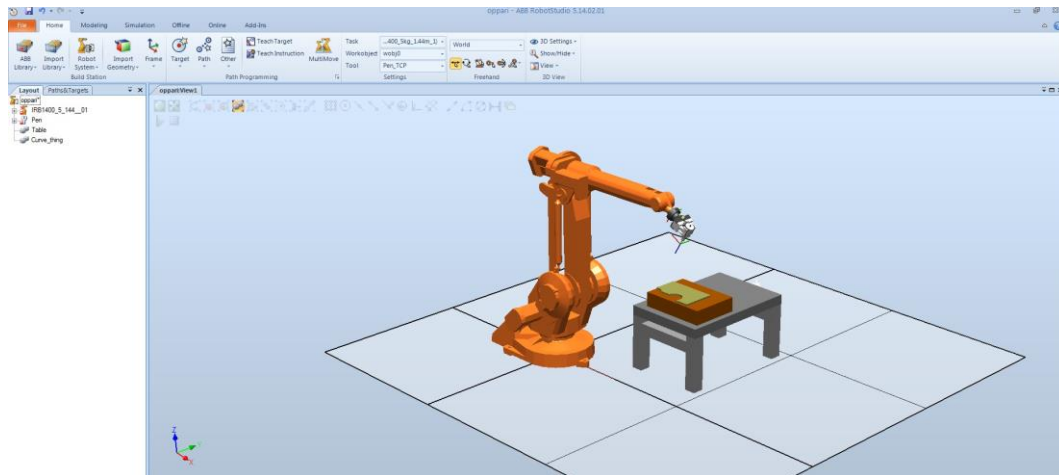


Kuva 11. Geometrian tuonti



Kuva 12. Move- ja Rotate-komento

Lisätään robottisoluuun vielä yllä olevan ohjeen mukaan työkappale. Liikutetaan molemmat geometriat haluttuun paikkaan, niin että robotti ylettyy työkappaleeseen. Asetetaan työkappale työtason pinnalle. Paikoitukseen käytetään Move-toimintoa. Paikoituksen käyttämisessä voidaan myös käyttää muita tapoja, mutta tässä työssä käytetään yllämainittua tapaa. Kuvassa 13 on esitelty yksinkertainen robottisoluu.

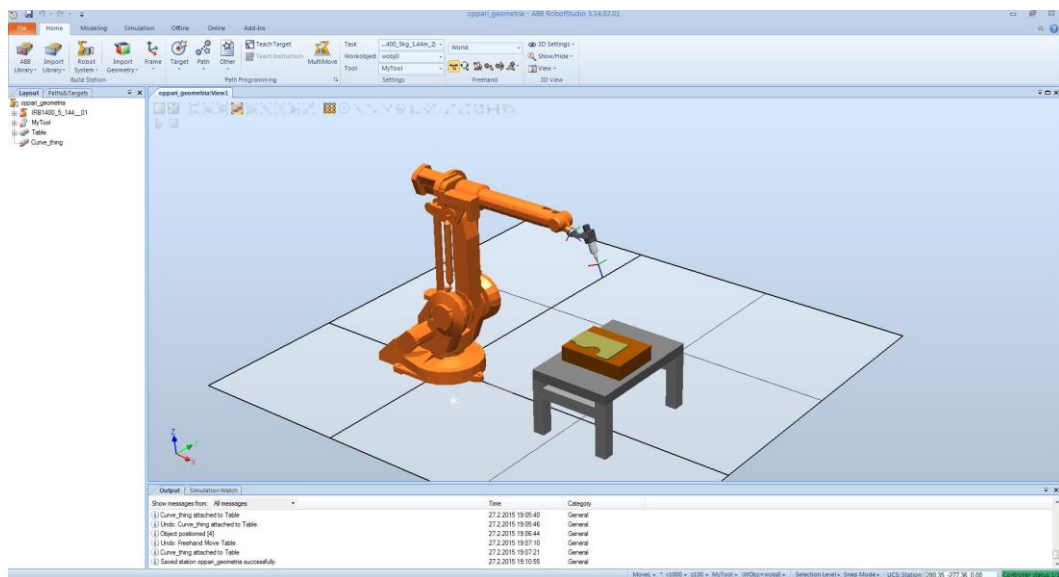


Kuva 13. Robottisolu

3.6 Polun luonti geometrialle

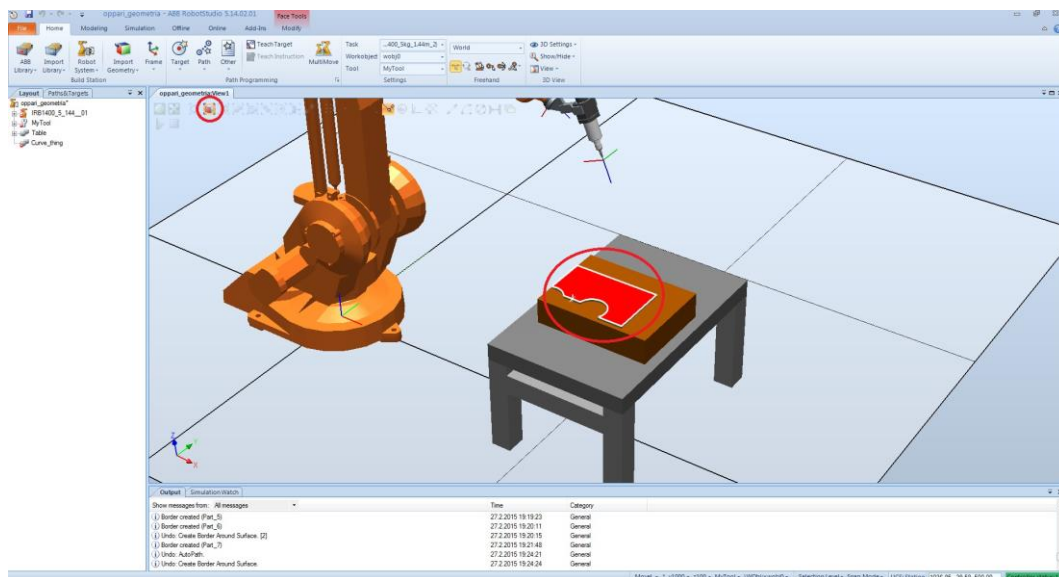
RobotStudioon avulla voidaan luoda ohjelma, joka seuraa kappaleen muotoja. Tämä mahdollistaa tarkan työn tekemisen helposti. Geometrialle polun luonnissa on tärkeä varmistaa, ettei robotti törmää työstäessä itseensä. Pisteiden on myös oltava robotin työskentelyalueen sisällä ja liikeradan ääripää ei saa ylittyä.

Luodaan uusi robottisolu. Tässä esimerkissä käytetään robottina IRB1400_5kg_1.44mm__01 ja Mytool-työkalua. Lisätään työtaso Table.sat ja työkalu Curve Thing. Siirretään geometriat niin, ettei robotti törmää työstäessä itseensä. Varmistetaan myös, että työkalun asento on sopiva robottiin nähden (Kuva 14.).

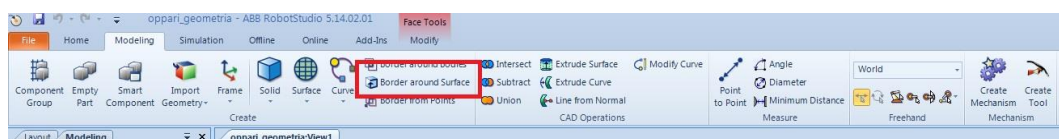


Kuva 14. Uusi robottisolu

Osien ja robotin ollessa oikeilla paikoillaan, valitaan työkappaleen pinta Surface Selection -toiminnolla (**Kuva 15.**). Pinnan ollessa aktivoitu siirrytään Modeling-valikkoon ja valitaan Border Around Surface -toiminto (**Kuva 16.**). Valitaan aukeavasta valikosta Create-toiminto, joka luo reunuksen. Tämän jälkeen Border Around Surface -toiminnon voi sulkea.

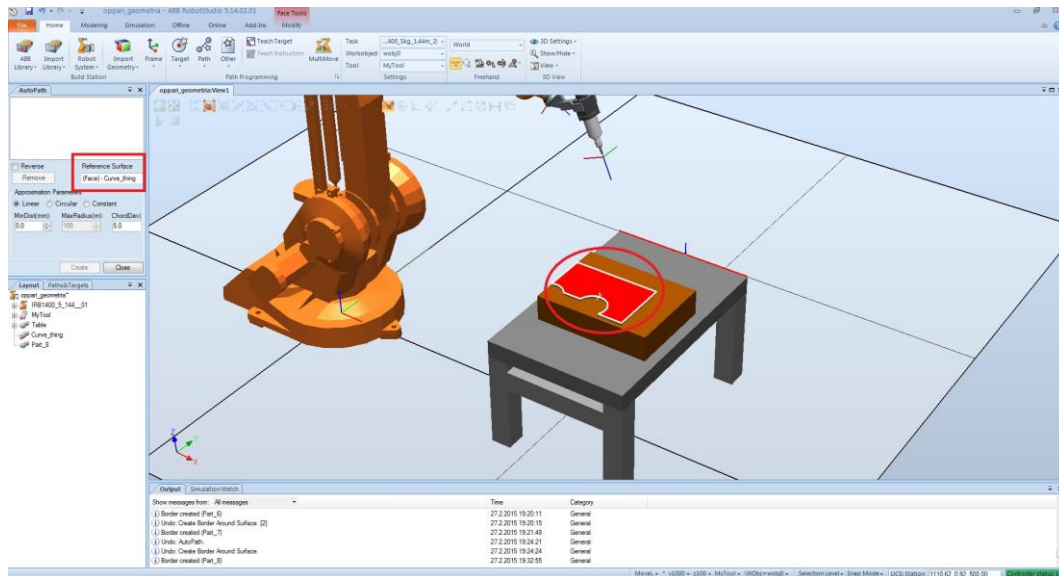


Kuva 15. Surface Selection -toiminto

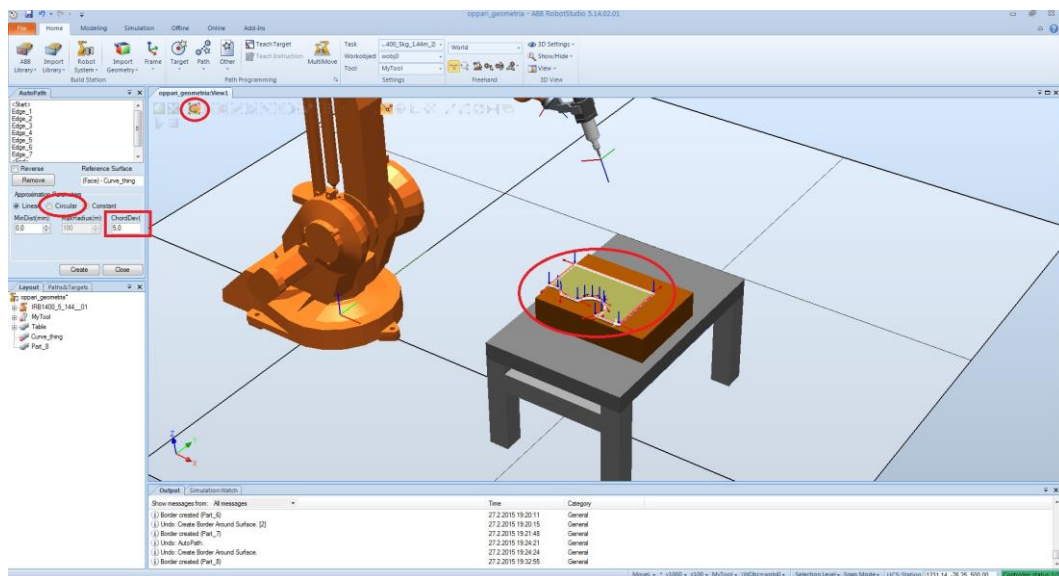


Kuva 16. Border Around Surface -toiminto

Seuraavaksi valitaan Home-valikosta Path-painikkeen alla olevasta nuolesta AutoPath-painike. Avautuvasta valikosta aktivoidaan Reference Surface ja painetaan hiiren vasemmalla näppäimellä kappaleen pintaa (**Kuva 17**). Valitaan Surface Selection -painikkeen vasemmalta puolelta Curve Selection -painike ja painetaan hiiren vasenta näppäintä työkappaleen reunaan. Tämä luo polun työkappaleen reunoihin. Vaihetaan Approximation Parameters -valikosta Max Chord Dev (mm) 1,0 lukemaan. Tämä ohjearvo määrää, kuinka monta pistettä kyseiselle reunalle tehdään. Vaihetaan vielä samasta valikosta pyöreät muodot painamalla Circular-kohtaan pallo. Viimeistellään vielä painamalla Create-toimintoa, jonka jälkeen Autopath-valikko voidaan sulkea Close-toiminnosta (**Kuva 18**).



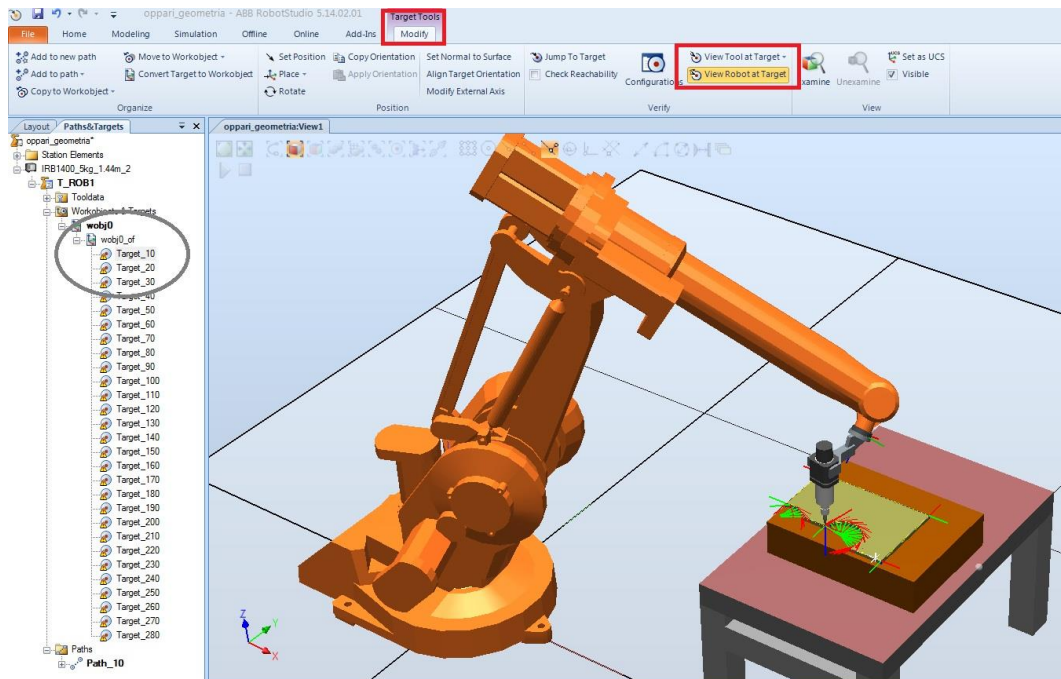
Kuva 17. Reference Surface



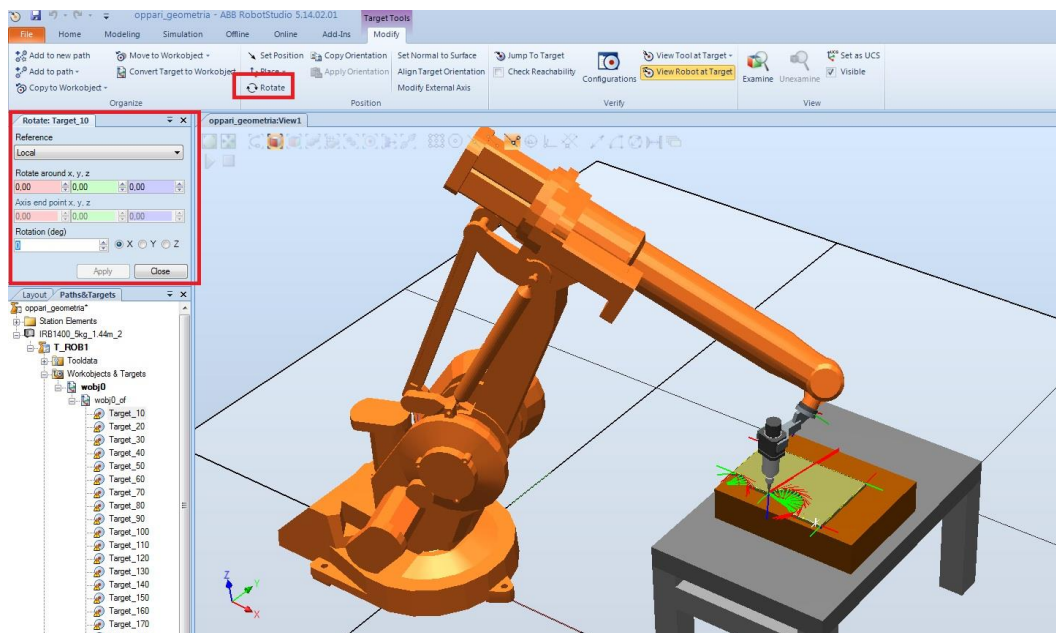
Kuva 18. Polun luonti

Robotille on nyt luotu pisteet sekä alustava polku reunalle. Siirrytään valikkoon, josta luodut pisteet löytyvät (ks. 3.3 pisteiden luonti robotille). Aktivoidaan ensimmäinen piste, avataan Modify-valikko ja valitaan View Robot At Target- sekä View Tool At Target-painikkeet. Tämän valinnan avulla robotti ja työkalu näkyvät valitussa pisteessä Target_10 (**Kuva 19**). Seuraavaksi valitaan Rotate-

näppäin. Tästä valikosta voidaan valita kuinka monta astetta työkalua käännetään: Rotation (deg) ja minkä akselin ympäri: X-, Y- ja Z-akselit (**Kuva 20.**)

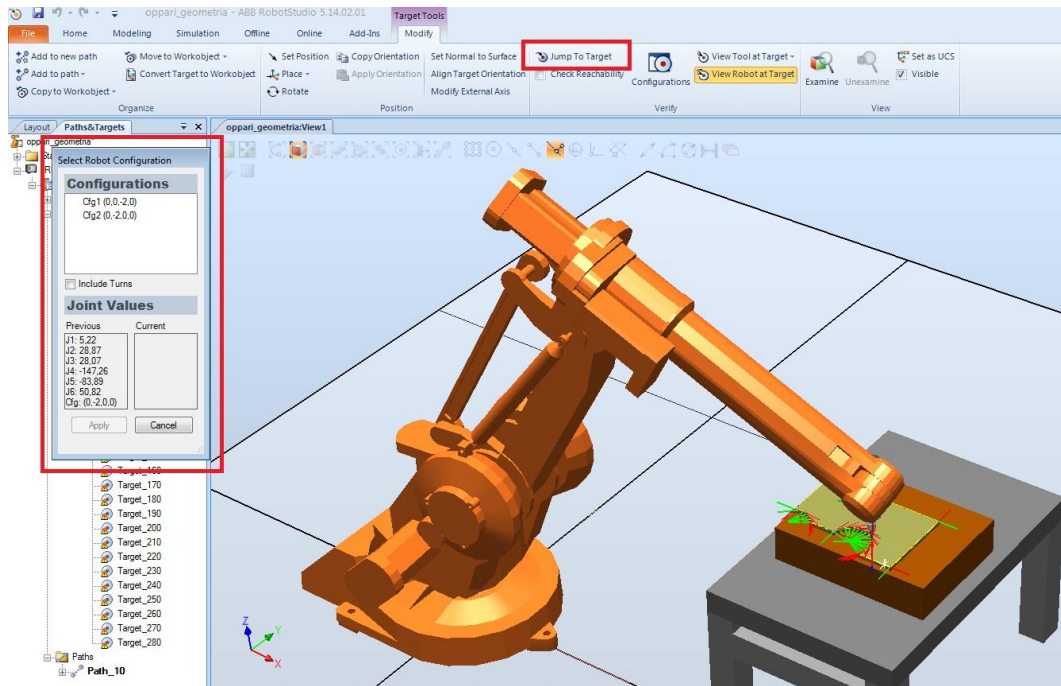


Kuva 19. Robotti valitussa pisteessä Target_10

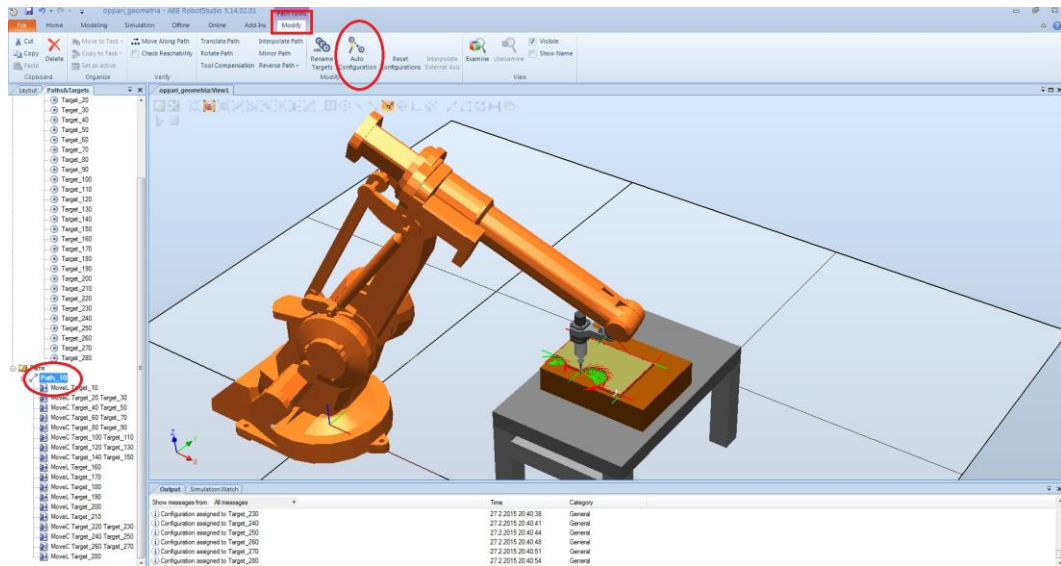


Kuva 20. Rotate-valikko

Kun työkalu on siirretty hyvään asentoon Rotate-valikon avulla, painetaan Jump to Target -näppäintä. Avautuvasta Select Robot Configuration -valikosta (**Kuva 21.**) valitaan ylin ja painetaan Apply. Ensimmäinen piste on luotu. Kun kaikille pisteille on asetettu oikea työkalun asento samalla tavalla kuin ensimmäiselle pisteelle, aktivoidaan Paths-valikko. Paths-valikon ollessa aktivoitu siirrytään Modify-valikkoon ja sieltä valitaan Auto Configuration-toiminto (**Kuva 22.**). Ohjelma laskee automaattisesti pisteiden kautta tehdyn liikeradan reunaan pitkin. Testataan uutta rataa painamalla Modify-valikosta Move Along Path -toimintaa.



Kuva 21. Jump to Target -valikko

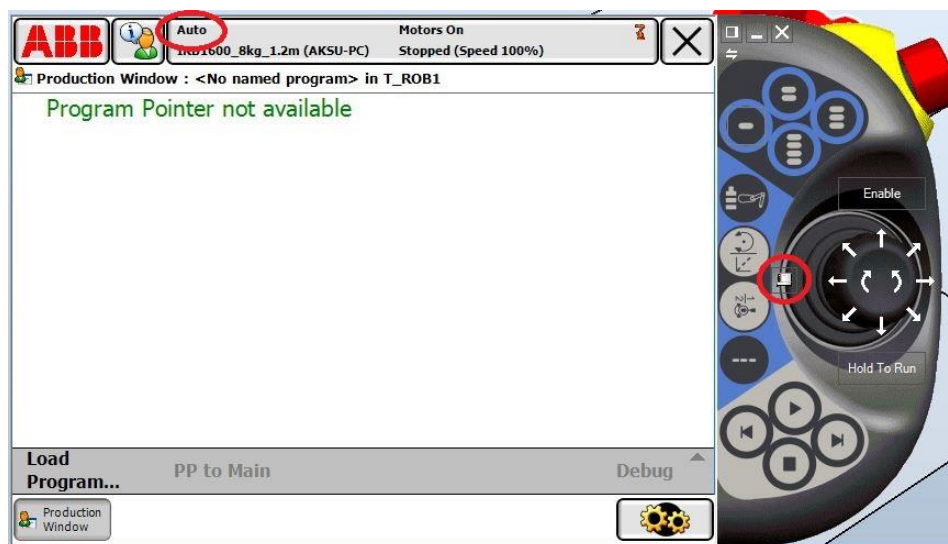


Kuva 22. Auto Configuration -toiminto

4 VIRTUAL FLEXPENDANT-TYÖKALU

Virtual flexpendant -työkalu vastaa online-ohjelmoinnin käsiohjainta. Ohjaimessa on sama ulkonäkö ja käyttö on samanlaista. Oleellisin ero on, ettei ohjainta voi fyysisesti pitää kädessä.

RobotStudion käsiohjain löytyy offline-välilehdeltä ja sieltä valitaan Virtual flexpendant. Käsiohjaimen auettua se on oletuksena automaattisessa tilassa, joka pitää muuttaa manuaaliseen tilaan, jotta akseleita päästään liikuttamaan. Ohjaimen yläpalkissa lukee ”auto”; tämä asetus saadaan vaihdettua painamalla ympyröityä painiketta (**Kuva 23.**).



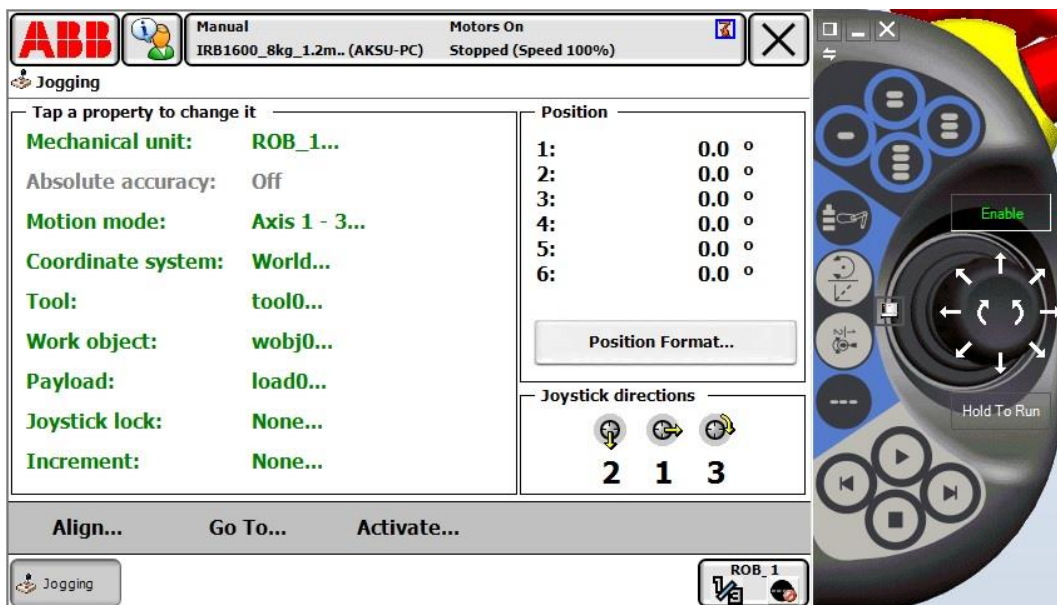
Kuva 23. Käsiohjain

Aukeavasta ikkunasta valitaan käsiajo, joka on ympyröity kuvassa 24. Vilkkuva valo tarkoittaa, että moottoreille on kytketty virta ja sitä voidaan ohjata. Käsiohjaimen yläpalkissa ollut auto muuttuu manuaaliksi. Painamalla Enable-näppäintä pystytään robottia liikuttamaan painamalla ainoastaan suuntanuolia. Tämä ilman, että ns. kuolleenmiehenkytkintä pitää painaa koko ajan. Näiden toimenpiteiden jälkeen robotti on valmiina virtuaalijoon.



Kuva 24. Käsihjaimen asettaminen käsiäjolle

Robotin asetuksista tulee usein vaihtaa, mitä robotin osia liikutellaan. Esimerkiksi, tuleeko robotin liikkua käyttäen akseleita 1-3 vai halutaanko robotin liikkuvan lineaarisesti. Näihin asetuksiin päästään painamalla ABB-painiketta ja valitsemalla valikosta Jogging (**Kuva 25.**).



Kuva 25. Jogging-valikko

4.1.1 Ohjelman luominen

Uuden ohjelman luominen aloitetaan painamalla ABB-logoa ja valitsemalla avautuvasta valikosta Program Editor. Avautuvaan ikkunaan vastataan, että halutaan tehdä uusi ohjelma.

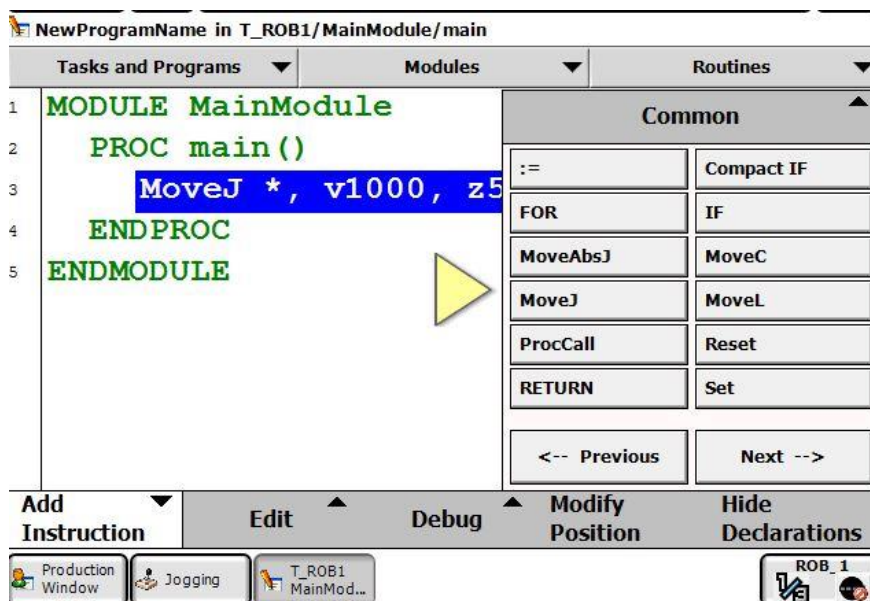
Uuteen ohjelmaan voidaan lisätä käskyjä Add instruction -valikon kautta. Yleisimmin käytetyt liikekomennot ovat Common-valikossa. Vaihtamalla valikkoa saadaan esille erikoiskäskyt, joita tarvitaan monimutkaisemmissa ohjelmissa.

4.1.2 Ohjelmointi

Aloitettaessa uuden ohjelman tekoa on hyvä luoda robotille kotipiste, johon se palaa suoritettuaan liikekomennot. Piste kannattaa valita niin, että siihen palaaminen ei saa robottia törmäämään vaikkapa työkohteeseen tai turvarajoihin.

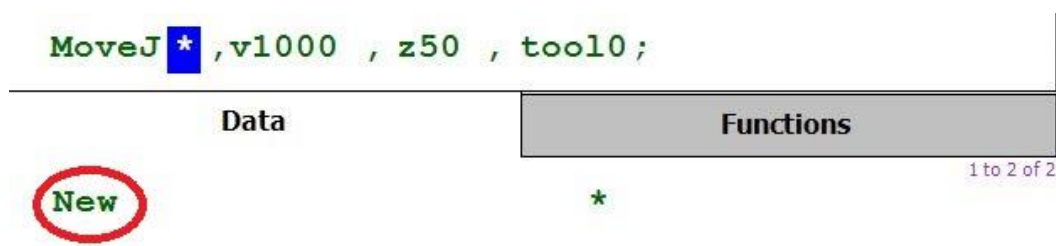
Luodaan uusi piste valitsemalla MoveJ-liikekomento (**Kuva 26.**). Nivelliikkeen avulla robotti pääsee liikkumaan vapaammin kuin lineaariliikkeellä, jolloin sen on helpompi palata kotipisteeseen. Suoraviivaiset liikkeet kannattaa suorittaa MoveL-liikekomennolla, mutta vaativimmat liikkeet, joissa niveletkin liikkuvat, kannattaa suorittaa MoveJ-liikekomennolla.

Ennen kuin luodaan uusi liikekomento, pitää robotti liikuttaa haluttuun paikkaan. Tämän jälkeen luodaan liike halutulla liikekomennolla, kuten MoveL tai MoveJ.



Kuva 26. Uuden liikekomennon luominen

Uuden ohjelman alussa on aina hyvä nimetä koordinaattipiste "*" toisella nimellä, koska tämä helpottaa tulevien koordinaattipisteiden erottamista toisistaan. Valitaan "*" ja kaksoisklikkaamalla sitä saadaan auki valikko, josta piste voidaan nimetä uudelleen valitsemalla "New" (Kuva 27.). Valitaan ohjelman ehdottama nimi "p10", josta tulee nyt uusi kotipiste (Kuva 28.). Seuraavat pisteet ohjelma nimeää automaattisesti p10:stä eteenpäin eli p10, p20, p30 jne.



Kuva 27. Piste nimeäminen uudelleen

New Data Declaration

Data type: robtarget Current Task: T_ROB1

Name: p10

Scope: Global

Storage type: Constant

Task: T_ROB1

Module: MainModule

Routine: <None>

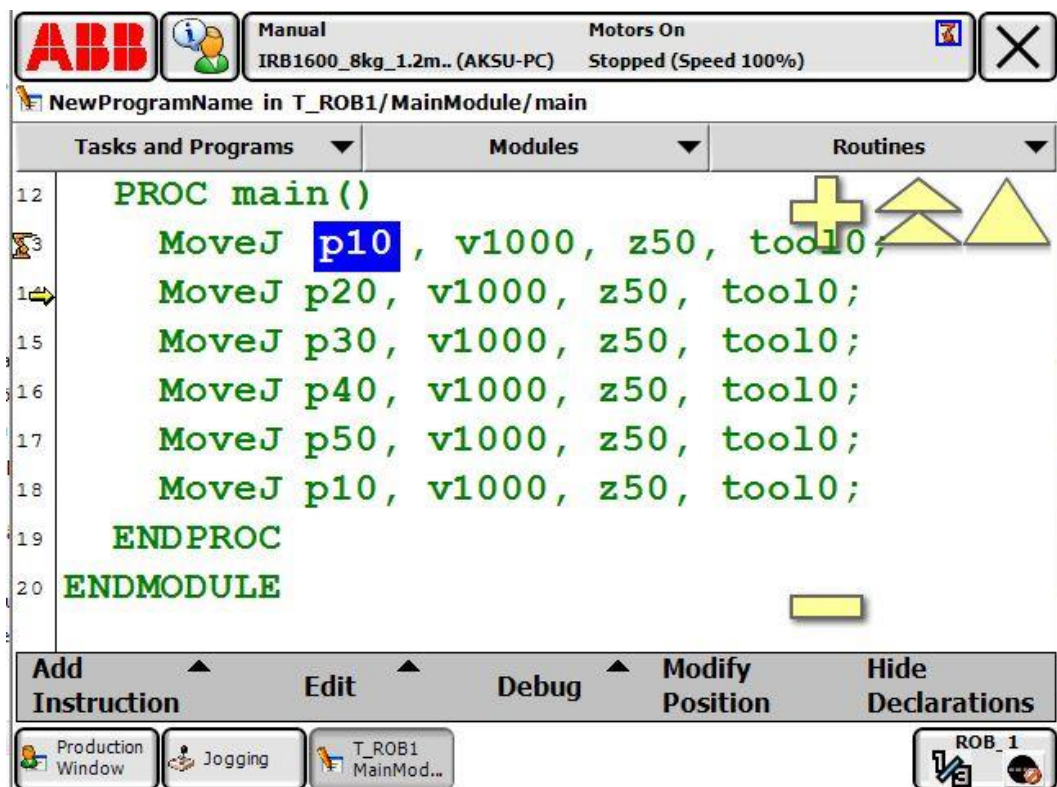
Dimension: <None>

Initial Value OK Cancel

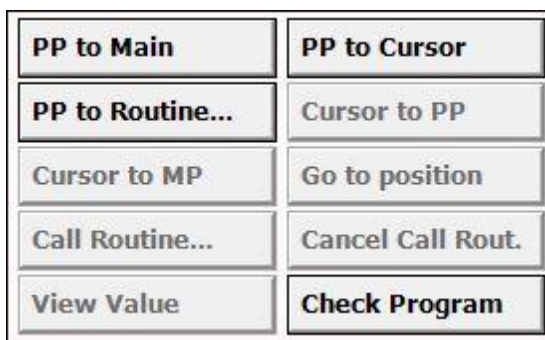
Kuva 28. Piste nimeäminen.

Nyt kun kotipiste on luotu, voidaan lisätä uusia pisteitä robotin liikuttamiseksi. Alla olevassa esimerkissä on lyhyt perusohjelma, jossa robotti suorittaa 5 liikekäskeyä (**Kuva 29.**). Ohjelman kotipiste on p10. Se suorittaa annetut liikekäskeyt ja palaa kotipisteeseen. Liikekomennon muuttujat tarkoittavat seuraavaa:

- Ohjelmassa liikekäskeyt suoritetaan MoveJ-liikekomennolla.
- Arvo ”v1000” kertoo, kuinka nopeasti robotti liikkuu.
- Arvo ”z50” kuvaa, miten robotti lähestyy pistettä. Jos robotin on tarkoitus kulkea täsmälleen määrätyn koordinaattipisteen kautta, pitää vaihtaa arvoksi z0 tai fine.
- ”tool10” on käytetty työkalu. Tässä valittu työkalu ”tool10” on kynä.



Kuva 29. Lyhyt perusohjelma.



Kuva 30. Ohjelman ajo.

Ohjelman ajo suoritetaan painamalla ”Debug”. Kuvan 30 mukaisesti valikosta voidaan valita, mistä ohjelman ajo alkaa:

- Pp to main aloittaa pääohjelma alusta.
- PP to cursor aloittaa siitä kohtaa, missä osoitin on kyseisellä hetkellä.

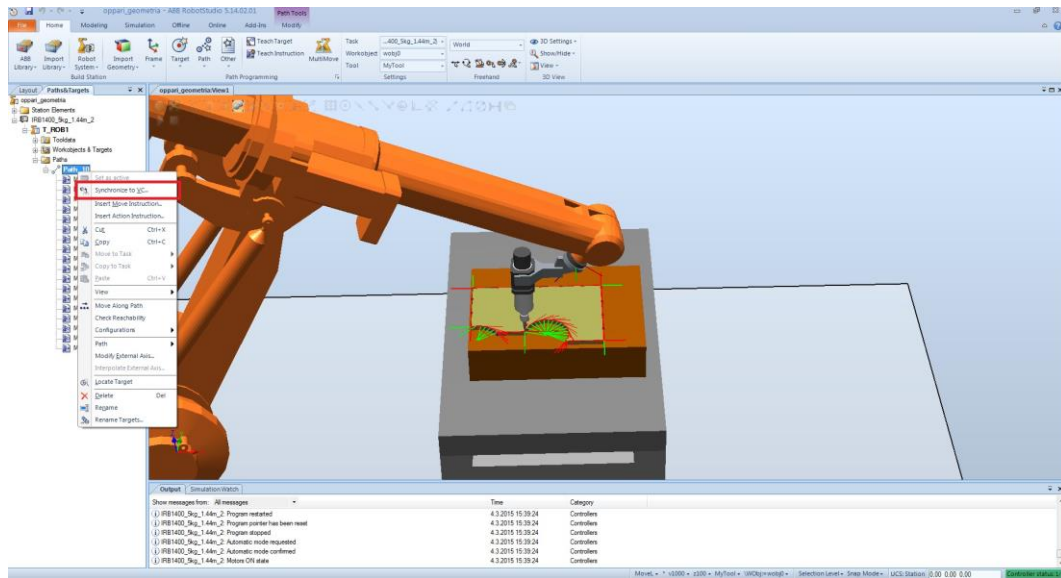
- PP to Routine ottaa esille rutiinivalikon, josta voidaan valita haluttu rutini. Tässä ohjelmassa on vain päärutini. Monimutkaisimmissa ohjelmissa pääohjelma on hyvä jakaa rutiineihin niiden tunnistamisen helpottamiseksi.

Kun haluttu ohjelma on valittu, voidaan ohjelman ajo aloittaa joko toistamalla koko ohjelma Play-painikkeella tai edetä askeleittain käyttämällä Forward-painiketta.

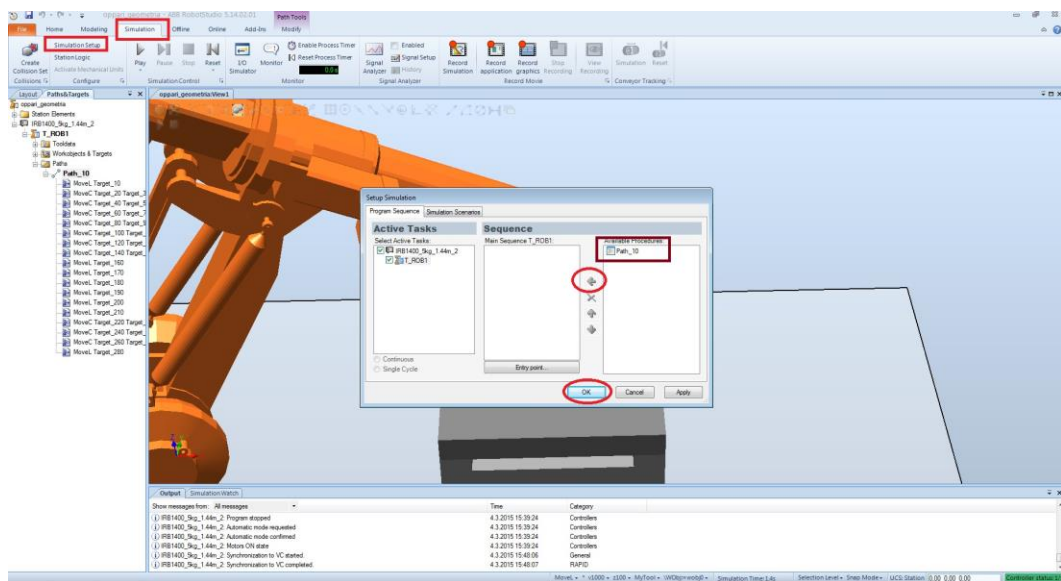
5 ROBOTIN SIMULOINTI

Robottiohjelman ollessa valmis, voidaan se kääntää koodiksi. Tämä koodikieleksi muuttaminen tapahtuu valitsemalla haluttu polku ja hiiren oikeasta näppäimestä aukeavasta valikosta valitsemalla Synchronize to VC -näppäin (**Kuva 31.**). Valitaan aukeavasta ikkunasta OK. Synchronize to VC -toiminto tarkoittaa ohjelman kääntämistä Virtual Controlleriin RAPID-koodikieleksi. Jos polkuja on useampi, pitää jokainen kääntää erikseen.

Kun kaikki halutut polut ovat siirretty Virtual Controlleriin, avataan yläpalkista Simulation. Aukeavasta valikosta valitaan Simulation Setup -näppäin. Uusi ikkuna avautuu näytölle. Tästä ikkunasta valitaan, mitkä polut syötetään simulaatio-ohjelmaan. Valitaan Path_10 ja painetaan nuolta vasemmalle. Nyt ikkuna voidaan sulkea painamalla OK (**Kuva 32.**). Polkuja ollessa useampi, pitää jokaiselle toistaa toiminto erikseen. Polkujen siirtojärjestys pitää olla sama kuin millä niitä olisi tarkoitus suorittaa. Näin robotti suorittaa ne oikeassa järjestyksessä. Kun halutut polut ovat siirretty, painetaan Play-näppäintä, joka liikuttaa robottia polkujen mukaan.



Kuva 31. Synchronize to VC -toiminto

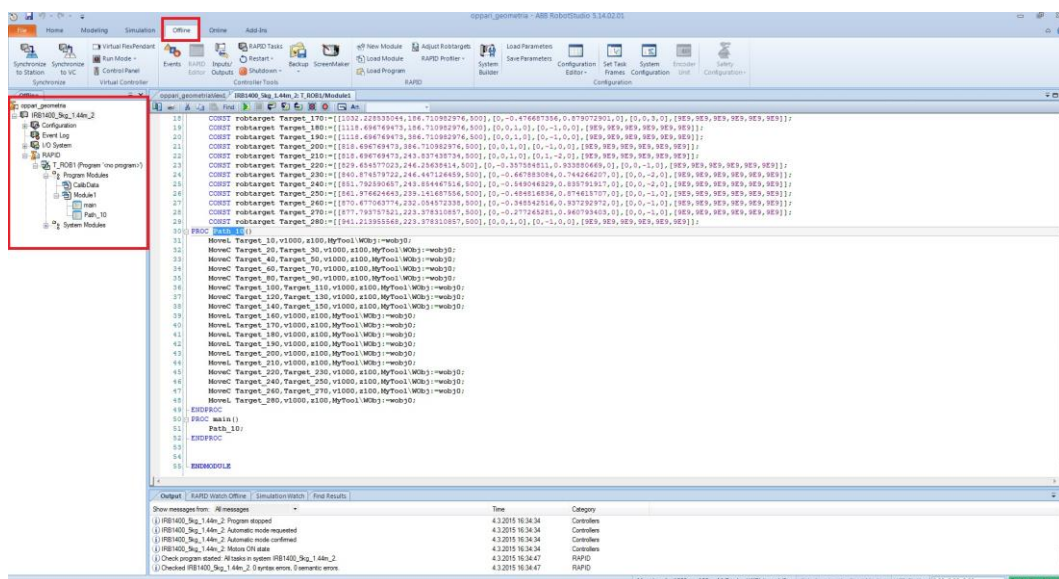


Kuva 32. Simulation Setup -toiminto

6 RAPID-KOODIKIELEN MUOKKAAMINEN

RobotStudio-ohjelma kääntää ohjelman automaattisesti RAPID-koodikielelle, jonka mukaan robotti liikkuu. Koodia voidaan tarpeen mukaan lisätä, poistaa ja muokata.

Avataan kappaleessa 5 luotu työ. Valitaan yläpalkista Offline. Vasemmalla olevassa ikkunassa on palkki, jossa lukee työn nimi. Laajennetaan palkkia painamalla robotin nimen vieressä olevaa plus-merkkiä. Avataan vielä lisää alle avautuvasta RAPID-valikosta. Avataan Program Modules -valikko ja sieltä Module1. Tämän avautuu nyt robotille käännetty ohjelma. Valitaan Path_10 ja kaksoisklikataan sitä. Kuvaruudulle ilmestyy ohjelma RAPID-koodikielellä. Sieltä ohjelmien muokkaaminen on mahdollista (**Kuva 33.**).



Kuva 33. RAPID-koodin avaaminen

7 YHTEENVETO

Tämän opinnäytetyön tavoitteena on saada ABB RobotStudion käyttöperusteet tutuksi, jotta etäohjelman käyttöä voidaan käyttää hyödyksi oppimisympäristössä. Opinnäytetyössä esitellään tarvittavat perustiedot, joiden avulla käyttäjä pystyy luomaan perusohjelman, testaamaan sen ja siirtämään ohjelman RobotStudiosta robotille.

Työ painottuu perustietojen opettamiseen, sillä RobotStudio on erittäin laaja kokonaisuus. Mahdollisen syventävän opiskelun kannalta ohjelman toimintoja olisi hyvä avata lisää ja esitellä ohjelman edistyneempiä toimintoja.

LÄHTEET

Vamk 2014. Vaasan ammattikorkeakoulun kotisivu. Tietoja Vamk:sta. Viitattu 10.12.2014. <http://www.puv.fi/fi/about/>

ABB 2015. ABB:n suomenkielinen kotisivu. ABB lyhyesti. Viitattu 3.3.2015. <http://new.abb.com/fi/abb-lyhyesti/yhtyma>

ABB 2014. RobotStudio-ohjelmiston yleiskatsaus. Viitattu 4.12.2014. <http://www.abb.com/product/seitp327/a3aae74d6d9cb7dcc1257559003458bb.aspx?productLanguage=fi&country=FI>

SFS 2014. Robottistandardi. Viitattu 4.12.2014. <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>

Keinänen, T. Kärkkäinen, P., Lähetkangas, M. & Sumujärvi, M. 2007. Automaatiojärjestelmien logiikat ja ohjaustekniikat. Helsinki: WSOY Oppimateriaalit Oy.

Malm, T. 2008. Vuorovaikutteisen robotiikan turvallisuus. Helsinki: Hakapaino Oy.

Opetusvideoita. Käytetty itse opiskelun apuna. <http://new.abb.com/products/robotics/robotstudio/tutorials>