

Juho Visuri

Teollisuusrobotin tuotantosolun mallintaminen ja simulointi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Kone- ja tuotantotekniikka

Insinöörityö

28.1.2016

Tekijä Otsikko	Juho Visuri Teollisuusrobotin tuotantosolun mallintaminen ja simulointi
Sivumäärä Aika	39 sivua + 2 liitettä 28.1.2016
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Kone- ja tuotantotekniikka
Suuntautumisvaihtoehto	Tuotantotekniikka
Ohjaaja	Lehtori Markku Saarnio
<p>Tässä insinööriyössä käsitellään tuotantojärjestelmiä ja niiden osia, sekä RobotStudio-simulointiohjelmaa. Työn teoriaosassa käydään läpi tuotantojärjestelmää yleisesti, sen historiaa ja rakennetta, sekä tuotantojärjestelmätyyppejä. Varsinainen työ tarkastelee RobotStudio-ohjelmaa, ja sen avulla teollisuusrobotin tuotantosolun mallintamista sekä simulointia.</p> <p>Insinööriyön tavoitteena oli perehtyä RobotStudio-ohjelman käyttöön ja ominaisuuksiin, sekä mallintaa ja simuloida teollisuusrobotin tuotantosolu tällä simulointiohjelmalla. Ohjelma hankittiin Metropolia Ammattikorkeakoulun konepajatekniikan laboratorioon tulevaa teollisuusrobotia varten.</p> <p>Työn tuloksena syntyi mallinnettu teollisuusrobotin sisältävä tuotantosolu ja siitä saatu simulointimalli. Simulointimallin robotin liikkeistä saatiin synkronoimalla RAPID-koodikielen koodi robotin liikeradalle.</p> <p>Insinööriyössä opittiin käyttämään RobotStudio-simulointiohjelmaa perustasolla. Ohjelmaan perehdyttäessä, sen käyttöön liittyvistä huomioista pyryttiin tiedottamaan eteenpäin Metropolia Ammattikorkeakoulun henkilökunnalle. Insinööriyöstä saatuja tietoja voidaan hyödyntää oppilaitoksen kursseilla sekä konepajatekniikan laboratorioon tulevan teollisuusrobotin käytössä.</p>	
Avainsanat	tuotantojärjestelmä, RobotStudio, simulointi

Author Title	Juho Visuri Modeling and Simulation of an Industrial Robot Production Cell
Number of Pages Date	39 pages + 2 appendices 28 January 2016
Degree	Bachelor of Engineering
Degree Programme	Mechanical Engineering
Specialisation option	Production Technology
Instructor	Markku Saarnio, Senior Lecturer
<p>This Bachelor's thesis discusses various production systems, its parts and the RobotStudio simulation program. In the theoretical part of this thesis, the production system, its history, structure and different production system types are examined. The main target was to analyze the RobotStudio program and how to model and simulate an industrial robot production cell with the program.</p> <p>The objective of this Bachelor's thesis was to study the usage and features of the RobotStudio program and in addition, to create a simulated production cell model for an industrial robot. The simulation program was acquired for the machining laboratory of Metropolia University of Applied Sciences.</p> <p>As a result of this thesis, a simulated model of the industrial robot production cell was made. The trajectories of the simulated model were synchronized to the RAPID-code language.</p> <p>In this Bachelor's thesis, the RobotStudio program was studied in a basic level. In the future, the information of this thesis can be utilized in the courses of Metropolia University of Applied Sciences and also in the practical use of the industrial robot in the machining laboratory.</p>	
Key words	production system, RobotStudio, simulation

Sisällys

Lyhenteet ja sanasto

1	Johdanto	1
1.1	Työn taustaa	1
1.2	Työn tavoite	1
2	Tuotantojärjestelmä	2
2.1	Yleistä	2
2.2	Historia ja kehitys	3
2.3	Rakenne	4
2.3.1	Tuotantokoneet ja koneistuskeskukset	4
2.3.2	Teollisuusrobotti	5
2.3.3	Siirto- ja kuljetusjärjestelmät	6
2.3.4	Varastointi	7
2.3.5	Ohjelmointi ja ohjausjärjestelmät	9
3	Tuotantojärjestelmätyypit	10
3.1	FMU - joustava automaattinen valmistusyksikkö	10
3.1.1	FTU - joustava automaattinen sorvausyksikkö	10
3.1.2	MCU - automaattinen koneistuskeskuskysikkö	10
3.2	FTL - joustava automaattinen transferlinja	11
3.3	FMS - joustava automaattinen valmistusjärjestelmä	11
3.4	FMF - joustava automatisoitu tehdas	12
4	Simulointi	12
4.1	Yleistä	12
4.2	RobotStudio	12
5	Tuotantosolun mallintaminen ja simulointi	13
5.1	Robotin työkalun mallintaminen ja määrittäminen	13
5.2	Robotin työaseman luonti ja tartuntaominaisuuden ohjelmointi työkaluun	18
5.3	Sorvimekanismin luominen	25
5.4	Liukuhihnamekanismin luominen	27
5.5	Robotin liikeratojen määrittäminen	32
5.6	Materiaalin tuonti soluun	34

6	Tulokset	36
7	Yhteenveto	38
	Lähteet	39
	Liitteet	
	Liite 1. Imukuppitarraimen työpiirros	
	Liite 2. Simulaation RAPID-koodi	

Lyhenteet ja sanasto

CAD	<i>Computer-aided design.</i> Tietokonemallinnus.
FMF	<i>Flexible Manufacturing Factory.</i> Joustava automatisoitu tehdas.
FMS	<i>Flexible Manufacturing System.</i> Joustava valmistusjärjestelmä.
FMU	<i>Flexible Manufacturing Unit.</i> Joustava valmistusyksikkö.
FTL	<i>Flexible Transfer Line.</i> Joustava transferlinja.
FTU	<i>Flexible Turning Unit.</i> Joustava sorvausyksikkö.
Layout	Pohjapiirros.
MCU	<i>Machining Center Unit.</i> Koneistuskeskusyksikkö.
NC	<i>Numerical Control.</i> Numeerisesti ohjattu.
Offline-ohjelmointi	Etäohjelmointi, tuotannon ulkopuolella suoritettavaa robotin ohjelmointia.
RAPID-koodikieli	Tekstipohjainen robotin ohjelmointikieli.
SCARA	<i>Selective Compliance Assembly Robot Arm.</i> Valikoiva, totteleva ja kokoonpaneva robottikäsi.

1 Johdanto

1.1 Työn taustaa

Tämän insinööriyön tilaajana on Metropolia Ammattikorkeakoulu. Työn taustalla on se, että konepajatekniikan laboratorioon hankitaan teollisuusrobotti. Robotti liikkuu noin kahdeksan metrin pituisella servoradalla, ja sen tehtävänä on työstettävien kappaleiden käsittely ja liikuttelu. Robottikäsivarsi palvelee laboratorion Mazak Mazatrol T32-3 NC-sorvia ja Quaser MV154P -koneistuskeskusta. Robotin liikkeiden ohjelmointia ja simulointia varten koulun tietokoneille asennetaan RobotStudio-simulointiohjelma, jolla näitä robotin suorittamia tehtäviä voidaan ennalta suunnitella ja simuloida. Lisäksi ohjelmaa käytetään hyväksi opetuksessa ja opiskelijat pääsevät perehtymään ohjelmaan Tuotantojärjestelmät- sekä Teollisuusrobotit tuotannossa -kursseilla.

1.2 Työn tavoite

Tavoitteena on perehtyä eri tuotantojärjestelmätyyppeihin ja yleisesti tuotantojärjestelmän rakenteeseen sekä RobotStudio-simulointiohjelmaan. Tarkoituksena on tutustua ohjelman käyttöön ja suunnitella sekä simuloida ohjelman avulla tuotantosolu.

2 Tuotantojärjestelmä

2.1 Yleistä

Tuotantojärjestelmässä materiaali on tarkoitus jalostaa tilatuksi tuotteeksi. Tuotantojärjestelmä koostuu kahdesta perustoiminnosta: suunnittelu- ja valmistusjärjestelmästä. Suunnitteluvaiheessa luodaan valmiudet valmistusvaiheelle, jossa varsinainen tilauksen jalostus toimitukseksi tapahtuu. Suunnittelu ja valmistus kulkevat käsi kädessä ja suunnittelua toteutetaan myös valmistusvaiheen aikana. Tuotantojärjestelmä rakentuu useista yksiköistä ja toiminnan tasoista. Näitä yksiköitä ovat esimerkiksi erilaiset osavalmistus- ja kokoonpanoyksiköt (kuva 1). Valmistusjärjestelmä voi koostua useista valmistusyksiköistä. Tuotantojärjestelmä muodostuu verkko-, tehdas-, solu- ja työasematasoista. [1, s. 15–16.]



Kuva 1. Esimerkki tuotantojärjestelmästä [2].

Tuotantojärjestelmästä puhuttaessa tulevat vastaan termit *miehittämättömyys* ja *joustavuus*. Miehitettömiin jaksoihin päästään riittäväällä järjestelmän ja sen osien automatisoinnilla ja valvontatoiminnoilla. Tässä yhteydessä joustavuudella tarkoitetaan tuotejoustavuutta, operatiivista joustavuutta sekä järjestelmän muunneltavuutta. Tuotejoustavuus tarkoittaa mahdollisuutta laajan osaperheen sekä samankaltaisten kappaleiden eri variaatioiden valmistamiseen. Operatiivisella joustavuudella tarkoitetaan järjestelmän kykyä pienten eräkokojen valmistukseen, hyvään ohjattavuuteen sekä reservikapasiteetin hallintaan. [1, s. 62–63.]

2.2 Historia ja kehitys

1950-luvulta alkaen numeerinen ohjaus mahdollisti automatisoinnin tuotannossa ja 1960-luvulla koneistuskeskukset joustavien tuotantojärjestelmien kehityksen. Tämän automatisoidun ja joustavan järjestelmän yleistymiseen meni kuitenkin vielä 10–15 vuotta. 1970-luvulla alettiin kiinnostua tuotannossa apuna käytettävistä teollisuusroboteista. Alkuaikoina robotteja käytettiin lähinnä hyllystöjärjestelmissä, mutta myöhemmin niiden käyttömahdollisuuksia hyödynnettiin myös esimerkiksi työkappaleiden panostuksessa ja kokoonpanon automatisoinnissa. Saksan demokraattinen tasavalta (Itä-Saksa, DDR) oli merkittävä tuotantojärjestelmien kehittäjä alkuvaiheessa, joskin sen merkitys sittemmin väheni. Japanissa joustavat tuotantojärjestelmät yleistyivät vuoden 1975 tietämillä ja Euroopassa 1980-luvun alussa. Suomeen ensimmäiset joustavat automatisoidut järjestelmät tulivat samoihin aikoihin. [3, s. 8–9.]

Tekniikan kehittyessä 1980- ja 1990-luvuilla kiinnostus tuotannon autonomiaa kohtaan kasvoi. Miehittämätöntä tuotantoa alettiin tutkia laajemmin ja erilaisia kokeiluja käynnistettiin Japanin johdolla. Miehittämätön tuotanto ei kuitenkaan vielä tuolloin ollut varmatoimista eikä kannattavaa taloudellisesti. Näistä syistä ja muutamien 1980-luvun kokeilujen takaiskuista johtuen joustaviin valmistusjärjestelmiin suhtauduttiin alussa varauksella, ennen kuin tekniikka kehittyi ja järjestelmät yleistyivät. [3, s. 9–10.]

2.3 Rakenne

2.3.1 Tuotantokoneet ja koneistuskeskukset

Joustavissa tuotantojärjestelmissä käytettävät laitteet ja koneet ovat numeerisesti ohjattuja. Tällaisia laitteita ovat muun muassa erilaiset pora- ja jyrsinkoneet sekä koneistuskeskukset. NC-porakoneissa automatiikka toteutuu kahdella akselilla, työkalun paikoitus tapahtuu numeerisesti ohjaamalla ja rajakatkaisimet hoitavat poran liikkeitä. NC-jyrsinkoneet puolestaan käyttävät kolmen akselin numeerista ohjausta, mutta työkalunvaihto suoritetaan vielä manuaalisesti. Kun jyrsinkoneen työkalunvaihto automatisoitiin, syntyivät pystykaraiset koneistuskeskukset. Vaakakaraiset koneistuskeskukset kehittyivät avaruskoneista. Koneistuskeskuksille ominaisia piirteitä ovat työkalunvaihdon ja työkalumakasiinin automatiikka ja vähintään kolmella akselilla tapahtuva samanaikainen numeerinen ohjaus. Koneistuskeskuksiin saatavilla erityyppisillä pyöröpöydillä saadaan työstettävää kappaletta käännettyä ja paletinvaihtajilla kappaleita vaihdettua, jolloin keskuksen käyttöympäristöä pystytään muokkaamaan. Koneistuskeskuksella on mahdollista porata, jyrsiä, avartaa, upottaa, kierteittää, kalvia ja väljentää. [4, s. 25–27.]

Kehitys sorveista sorvauskeskuksiin on ollut samankaltainen. Sorveihin laadittiin numeerinen ohjaus ja työkalut asetettiin revolveriin. Perinteisellä NC-sorvilla ei kuitenkaan voida suorittaa mahdollisia poraus- ja jyrsintätyövaiheita, jolloin kehitettiin pyöriviä työkaluja käyttävät sorvauskeskukset. Sorvauskeskuksella voidaan moni kappale työstää alusta loppuun saakka samalla koneella. Sorvauskeskuksessa on jakava kara ja mahdollisesti useampi kuin yksi työkalurevolveri. Käännettävät kappaleet asettavat sorvauskeskuksille ja niiden kappaleenvaihtajille haasteita. Osittain sorvattu työstökappale saattaa tarvita väliaikaisen varaston. Ratkaisuna tähän kehitettiin kaksikarainen sorvi, jossa karat ovat vastakkain. Tällaisella sorvilla, kun kappaleen yksi pää on työstetty, voidaan kappaleen toinen pää työstää heti apukaralla, ja pääkara voi siirtyä työstämään uutta kappaletta. Sorveille tavallisimpia lisälaitteita ovat kappaleenvaihtaja sekä tangonsyöttölaite. Lisäksi löytyy kouruja ja kaappareita, joilla valmis kappale voidaan poistaa sorvista. Työstöstä syntyvät lastut poistuvat koneesta kuljetinta pitkin. On olemassa myös pystysorveja, jotka voidaan niin ikään laajentaa sorvauskeskuksiksi. Tällaisella sorvilla voidaan myös jyrsiä ja porata kappaletta sorvaamisen lisäksi. [4, s. 27–29.]

NC-tekniikka on kehittynyt myös levytyötuotannossa, minkä seurauksena levytyökeskukset ovat syntyneet. Levytyökeskuksissa työkalut sijaitsevat työkalukaseteissa, jotka toimivat automatiikalla. Keskuksissa myös työkalujen kierto on mahdollinen. Levyihin työstettävät kappaleet voidaan valmiina jättää levyyn kiinni ja irrottaa ne myöhemmin tai pudottaa heti pöydän läpi keräävään kasettiin. Levytyökeskuksella muotojen leikkaus voidaan suorittaa laser- tai plasmaleikkaamalla. Kun keskukselle määrätty levytyö on käynnissä, itse levy liikkuu ja kappaleiden lävistysyksikkö on paikoillaan. Nykyään osalla levytyökeskuksista on jo mahdollista leikkaamisen lisäksi tehdä erilaisia nakerrus-, kierteitys- ja taivutus- sekä rullaustöitä. [4, s. 30–31.]

2.3.2 Teollisuusrobotti

Yksittäisellä teollisuusrobotilla tarkoitetaan itsenäisesti toimivaa vähintään kolminivelistä laitetta, joka voidaan ohjelmoida uudelleen ja uudelleen. Robotin liikerata voi olla määriteltä täysin ennalta, se voi määräytyä robotin työympäristön toiminnan mukaan tai olla esimerkiksi liikkeiden aikana antureiden mukaan valittava. Robotin rakenteeseen kuuluu jalusta, nivelillä toisiinsa liittyvät tukivarret ja kiinnityslaippa, johon robotin käyttämät työkalut kiinnitetään. Nivelet liikkuvat ohjattavilla servotoimilaitteilla. Teollisuusrobotit voidaan jakaa eri luokkiin niiden rakenteen ja liikealueen perusteella. Luokat jakautuvat suorakulmaisiin, rinnakkaisrakenteisiin, sylinterimäisiin, napakoordinaatistisiin sekä SCARA-luokkaan ja kiertyvänivelisiin. [5, s. 110–112.]

Robottiikalla osana tuotantojärjestelmää tarkoitetaan erilaisia teollisuusrobottisovelluksia ja robottijärjestelmiä. Joustavassa järjestelmässä robottien rooli on tuotantosolujen kappaleiden käsittelyssä. Erilliset solut muodostavat joustavan valmistusjärjestelmäkokonaisuuden. Robotteihin liitettävät erilaiset työkalut kuten tarraimet ja hitsausvälineet sekä niiden muunneltavuus mahdollistavat robottien käytön useissa eri tuotantotehtävissä ja näin ollen nopeat ja laajatkin layout-muutokset tuotantojärjestelmässä. [6, s. 15, 160–161.]

2.3.3 Siirto- ja kuljetusjärjestelmät

Tuotantojärjestelmässä kappaleiden ja materiaalin siirtoihin ja kuljetuksiin käytetään erilaisia kuljettimia, rullaratoja, kisko-, vihi- sekä siirtovaunuja ja portaalirobotteja. *Kuljettimet ja rullaradat* ovat tyypillisesti mekanisoituja, toisinaan myös automatisoituja. Ne voidaan sijoittaa lattialle tai kattoon, materiaalivirta on niissä yleensä yksisuuntaista ja niitä käytetään myös varastoinnissa. Kuljettimilla ja rullaradoilla on mahdollista toteuttaa niin yksittäisten kappaleiden kuin myös jatkuvien kuormien siirtoja. [6, s. 228–229.]

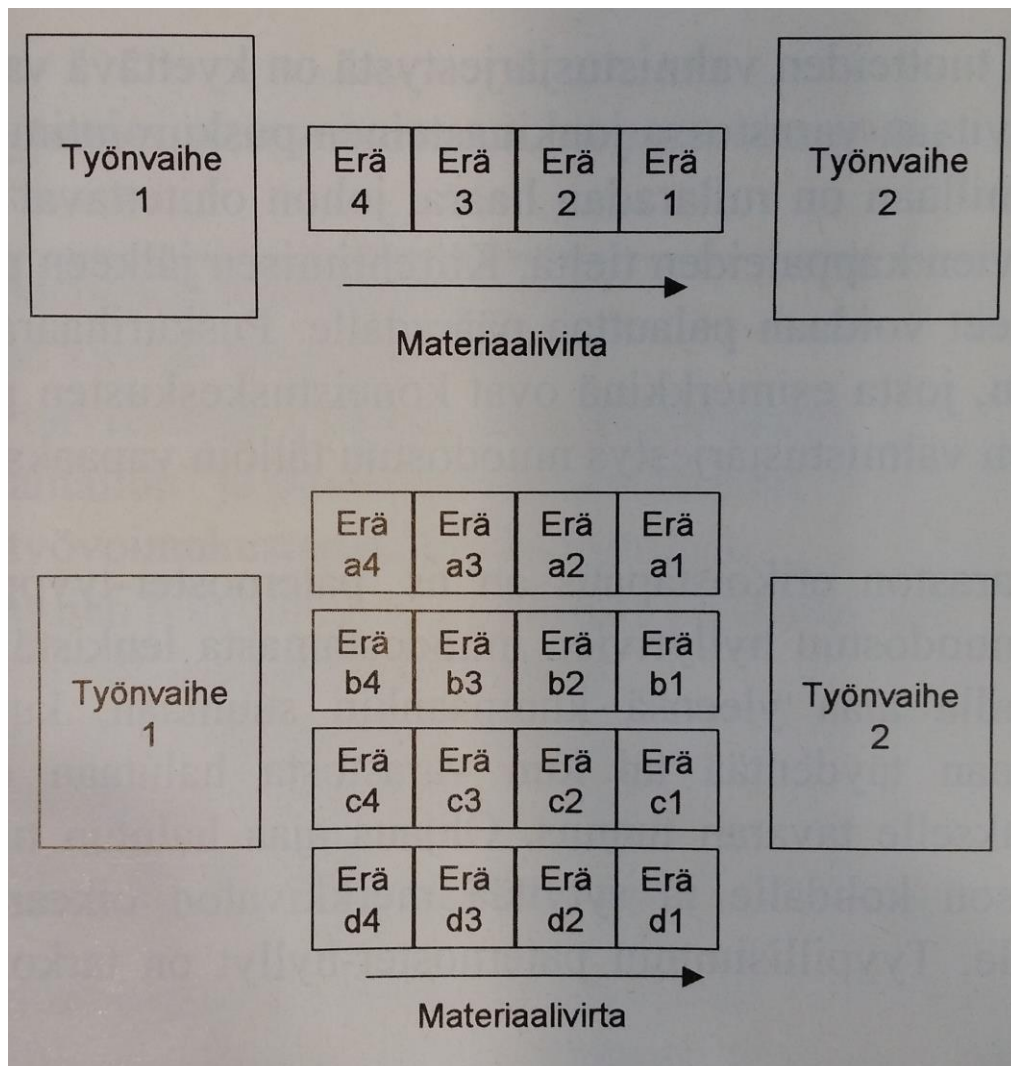
Kiskovaunuja hyödynnetään kuljetusalustojen ja palettien siirroissa valmistusyksiköstä toiseen. *Vihivaunu* ei tarvitse toimiakseen pysyviä kiskoja tai ratoja, vaan sen ohjaus perustuu esimerkiksi lattiaan merkätun reitin seuraamiseen optisesti tai lattiaan upotetun johtimen seuraamiseen antennin avulla. Vihivaunuilla voidaan kuljettaa raskaita kuormia pitkiäkin etäisyyksiä ja työvaiheita pystytään suorittamaan kappaleen ollessa vaunun kyydissä. Trukkimaiset vihivaunut kykenevät myös kuormien nostoihin. [6, s. 230–233.]

Jos järjestelmän toimivuuden edellytyksenä ovat kappaleiden nopeat ja tehokkaat siirrot ja kuljetukset, suositaan pinoavia *siirtovaunuja*. Siirtovaunut mahdollistavat korkeat nostot, jolloin tilojen käyttö tehostuu. Vaunuissa hyödynnettävät numeerinen ohjaus ja digitaalinen tekniikka puolestaan takaavat siirroille täsmällisen paikoituksen. Siirtovaunusysteemi koostuu kahdesta kiskosta, joista toinen sijaitsee lattiassa ja toinen siirtovaunun yläpäässä, ja varsinainen vaunu liikkuu näiden kiskojen avulla. Vaunulla kappaleita ja materiaaleja voidaan liikutella pysty-, pituus- ja poikittaissuunnassa. Noudettava kohde haetaan ja jätetään halutulle paikalle vaunun teleskooppihaarukoilla. Jotta siirtovaunulla pystytään operoimaan jokaista varastosta löytyvää palettia ja kuljetusalustaa, on vaunun teleskooppihaarukassa oltava ohjelmointimahdollisuus. Teleskooppihaarukan tilalle löytyy joitakin vaihtoehtoisia ratkaisuja, kuten esimerkiksi suunnikasmekanismi, joka soveltuu hyvin pienemmän mittakaavan automaattisiin järjestelmiin. [6, s. 235–237.]

Muutaman tuotantokoneen järjestelmissä, joissa tilat on rajattu ja työstökoneisiin voi olla vaikeaa päästä käsiksi, voidaan kappaleiden käsittelyyn ja siirtoihin käyttää *portaalirobotteja*. Tällainen portaalisysteemi on irti lattiatasosta, se on nopealiikkeinen ja paikoitustarkkuudeltaan hyvä sovellus. Portaaliratkaisut on aidattava, mutta huoltotoimenpiteille saadaan tilaa ohjaamalla portaalit syrjään huollon tieltä. [6, s. 237–238.]

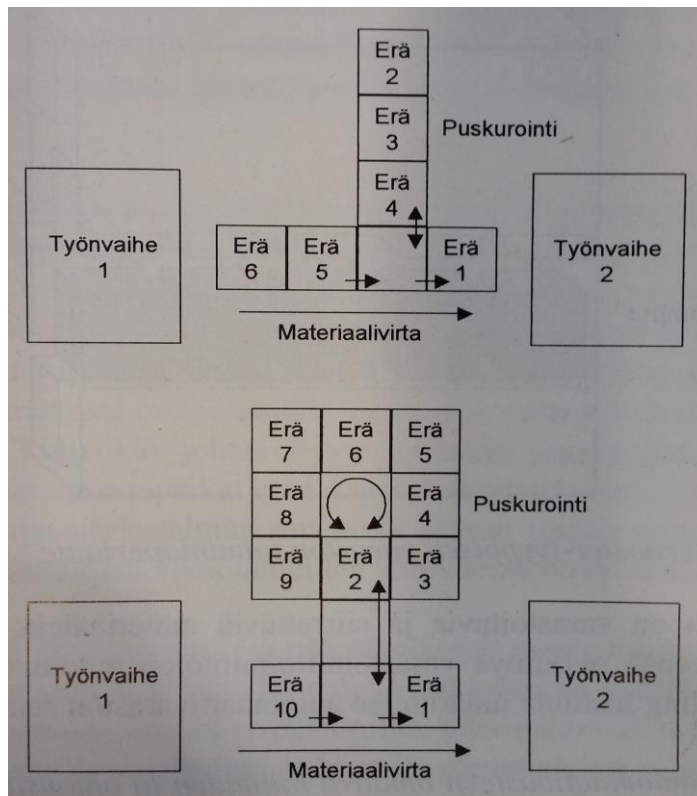
2.3.4 Varastointi

Osana tuotantojärjestelmää varastointia tarvitaan niin yksittäisillä tuotantokoneilla kuin myös työvaiheiden ja tuotantosolujen välillä. Varastoitavia materiaalityyppejä ovat muun muassa raaka-aineet, keskeneräiset tuotannon tuotteet, työkalut, varaosat ja lopullinen tuote. Varastot voidaan järjestää helpoimmillaan hyllyillä tai rullaradoilla toteutettaviksi läpivirtausvarastoiksi (kuva 2). [6, s. 223–225.]



Kuva 2. Esimerkki läpivirtausvarastosta [6, s. 225].

Varastot voivat olla myös puskuroivia (kuva 3), mikä voidaan toteuttaa esimerkiksi rullaradan haaralla, jolloin kiireelliset kappaleet saadaan ohittamaan kiireettömät kappaleet. Koneistuskeskusten työkalumakasiinit ovat esimerkki puskuroivasti varastosta. Puskuvoiva varasto voi olla myös paternoster-tyyppinen, jossa hyllyrivit ovat lenkkimäisessä ketjussa pystyvarastossa. Varastoitu materiaali saadaan varastosta, kun ohjaukselle annetaan materiaalin tunnus ja ohjaus ajaa materiaalin ulosottoaukole. [6, s. 224–226.]



Kuva 3. Esimerkki puskuroivasta varastosta [6, s. 225].

Varaston automatisointi on kannattavaa, kun varastoitavaa ja siirrettävää on paljon. Automatisoitu varasto on tarkka ja nopeatoiminen. Suurissa automatisoiduissa varastoissa vapaa reititys on merkittävää. Tällöin kiinteistä rullaradoista ja kuljettimista voidaan luopua ja tilalla käytetään siirto- ja kuljetusratkaisuja. Joustavan tuotantojärjestelmän varastojärjestelmä pystyy noutamaan tarvittavat materiaalit ja kappaleet toivotussa järjestyksessä ja siirtämään ne määrättyihin paikkoihin. Varastojärjestelmissä materiaalin siirtoihin käytettävä aika pyritään minimoimaan, jolloin varastoitavat kohteet voidaan siirtää mahdollisimman lähelle seuraavaa työvaihetta lähimpään vapaaseen paikkaan varastossa. Tämä saattaa kuitenkin sekoittaa varastojärjestelmän, ja tilanne varastossa on tällöin vain ohjauksen muistin varassa. Tämä puolestaan voi hankaloittaa varaston inventointia ja päivitystä joissakin tilanteissa. [6, s. 226–227.]

2.3.5 Ohjelmointi ja ohjausjärjestelmät

Työstökoneiden NC-ohjelmaan vaikuttavat työkappaleen geometria, kiinnitin, kappaleen materiaali, työstössä käytettävät työkalut ja tietysti käytettävä numeerisesti ohjattu tuotantokone. Hyvä NC-ohjelma on moduulirakenteinen, helppolukuinen, helposti korjattavissa ja tehokas. Jos ohjelma on heikosti tehty, se tuhlaa koneaikaa turhaan. [4, s. 103–105.]

Ohjelmointitapoja on useita. Ne voidaan jakaa *manuaaliseen, vuorovaikutteiseen ja tietokoneavusteiseen ohjelmointiin*. Ohjelmoijan on tunnettava perusteellisesti ohjelmoitava työstökone ja sen ohjausjärjestelmä. Manuaalista ohjelmointitapaa tulisi käyttää silloin, kun kappaleet ovat yksinkertaisia ja erä koko on suuri, kun voidaan soveltaa aliohjelmiä ja kun NC-koneet ovat yksinkertaisia ja niitä on vähän. Vuorovaikutteinen konepajaohjelmointi tuo mukanaan simuloinnin ja helpottaa siten ohjelmointia. Tietokoneavusteisella ohjelmoinnilla käsitellään monimutkaisia kappaleita ja tätä tapaa käytetään, kun työkappaleesta on saatavilla CAD-malli sekä kun ohjelmoitavia kappaleita on suuri määrä ja NC-koneita on useampia. [4, s. 106–111.]

Tuotantojärjestelmässä koneet, siirto- ja kuljetusjärjestelmät, varastot sekä työväline- ja tietovirta on koottava yhdeksi toimivaksi kokonaisuudeksi. Valmistustoiminta on pelkistettynä siirtymis- ja liikesarjojen sekä niiden tietovirtojen hallintaa. Numeerinen ohjaus luo perustan tuotantoautomaatiolle ja ohjausjärjestelmillä hoidetaan niin rataohjaukset, pisteohjaukset ja interpolaatiot kuin myös janaohjauksetkin hallittavasta työstökoneesta riippuen. [6, s. 85–88.]

3 Tuotantojärjestelmätyypit

3.1 FMU - joustava automaattinen valmistusyksikkö

3.1.1 FTU - joustava automaattinen sorvausyksikkö

Joustava automaattinen sorvausyksikkö koostuu numeerisesti ohjatusta sorvista ja sen työkalujärjestelmästä, varastoivista kuljettimista ja valvontatoiminnoista miehittämättömiä työkaksoja varten. Työkappaleiden varastoinnissa käytetään kuormalavoja ja kappaleiden siirroissa portaalirobotia. Järjestelmän ohjaukselta vaaditaan kykyä työjonon ja kappalemäärien hallintaan sekä työstöohjelmien hakemiseen järjestelmän muistista. Uusien numeeristen ohjainten joustavuus perustuu niiden tietokonepohjaan. Sorvausyksikölle on ominaista kappaleiden erävalmistus työaikojen lyhyiden ja asetusten seurauksena. [1, s. 155.]

3.1.2 MCU - automaattinen koneistuskeskysyksikkö

Automaattiseen koneistuskeskysyksikköön kuuluu olennaisena osana palettijärjestelmä. Paletit voidaan varastoida koneistuskeskuksen vaihtoasemien kanssa samalla tasolla tai monitasoisessa hyllystössä. Palettien siirto tapahtuu palettiradalla, kiskovaunuilla tai hyllystöhissillä. Palettirataa voidaan käyttää varastona, siirroissa, panostus- tai purkauspaikkana ja odotusasemana. Palettirataratkaisu soveltuu kapean osaperheen tuottamiseen. Kiskovaunut puolestaan sopivat yhden tason palettivaraston yhteyteen. Hyllystöhissiä käytetään pienemmissä yksiköissä kiskovaunun tavoin, mutta se sopii myöskin laajempiin systeemeihin. [1, s. 156–158.]

3.2 FTL - joustava automaattinen transferlinja

Transferlinjalla numeerisesti ohjattujen koneyksiköiden työnkulku on kiinteä. Yksiköt ovat kuitenkin joustavia ja omaavat mahdollisesti usean akselin. Yksittäisen kappaleen erä-tuotantokin on mahdollinen riippumatta linjamuodosta. Työkappaleet kulkevat paleteilla työstökoneiden työasemien läpi. NC-ohjelma kutsutaan työstökoneelle paletissa olevalla koodilla. Koneiden välissä on puskurivarastoja, jotka huolehtivat tuotannon tahdituksesta. Transferlinja ei ole yhtä joustava kuin varsinainen FMS, ja tuotanto vaatii suurehkoja kappalemääriä yhdenmukaisia kappaleita. Kuitenkin variaatioita löytyy paljon ja yhden kappaleen erä kulkee linjan läpi siinä missä isompikin eräkokko. Transferlinjalla voidaan tuottaa niin pyörähdys- kuin myös paleteille kiinnitettäviä kappaleita. [1, s. 159–160.]

3.3 FMS - joustava automaattinen valmistusjärjestelmä

Joustava automaattinen valmistusjärjestelmä koostuu tyypillisesti useasta numeerisesti ohjatusta tuotantokoneesta, esimerkiksi hitsausrobotista, levytyökoneesta ja koneistuskeskuksesta sekä erilaisista kappaleiden ja palettien kuljettimista, varasto- ja ohjausjärjestelmästä sekä muista laitteista, esimerkiksi robottiasemasta, joka hoitaa materiaalien käsittelyä. Valmistusjärjestelmässä valmistettavalle kappalevalikoimalle asettaa rajoituksia kappaleiden muoto ja koko, materiaali sekä vaaditut toleranssit. Järjestelmässä kappale tulee paletille lastausasemassa esimerkiksi korkeavarastosta, kun työstökone tilaa sen sieltä ohjelmoidun työstöohjelmansa mukaan. Ohjelman jälkeen työstetty kappale palautuu paletissa varastoon kuljettimia pitkin tai esimerkiksi vihivaunulla ja siitä nosturilla tai hyllystöhissillä omalle varastopaikalleen. Joustavuus merkitsee valmistusjärjestelmässä sitä, että siinä valmistettavat kappaleet voivat olla erilaisia, mutta järjestelmän asettamissa rajoissa. Tämä antaa pelivaraa valmistuksessa, kun koko tuotantoerää varten ei tarvita yhdenlaisia kiinnittimiä ja eri kappaleita voidaan tuottaa eri määrät. Haluttuja kappaleita ei tarvitse tuottaa ylimäärin ja välivarastoida, vaan sen sijaan valmistaa kysynnän mukaan. [4, s. 192–195.]

3.4 FMF - joustava automatisoitu tehdas

Joustavassa automatisoidussa tehtaassa käytetään nykyaikaista valmistus- ja automaatiotekniikkaa, jota tuotannon työntekijät ohjaavat ja valvovat. Materiaalivirrat ja informaatio on integroitu tehtaan automaatiojärjestelmään. Tehtaan tuotantokokonaisuus koostuu valmistuksen, kokoonpanon ja muiden tarvittavien järjestelmien automaatiosta. Tällainen joustava automatisoitu tehdas hyödyntää pitkiä miehittämättömiä jaksoja ja on laaja kokonaisuus. [7.]

4 Simulointi

4.1 Yleistä

Tuotantojärjestelmien yhteydessä simuloinnista puhuttaessa tarkoitetaan tietokoneella tehtävää järjestelmän mallintamista. Järjestelmän mallia voidaan muunnella ja manipuloida ja saatuja tuloksia analysoida. Tuloksia ovat animaatiomalli ja numeeriset sekä graafiset tulokset, ja niistä voidaan ennustaa järjestelmän käyttäytymistä. Kuviteltuja ongelmatilanteita ja ideoita voidaan testata simulointimallissa, jolloin saadaan käsitys järjestelmän käyttäytymisestä eri tilanteissa. Käytännössä kokeilemisen sijaan simuloinnilla vältetään turhia muutoksia tuotannossa eikä tuotannon toiminta häiriinny tai keskeydy. [1, s. 319–322.]

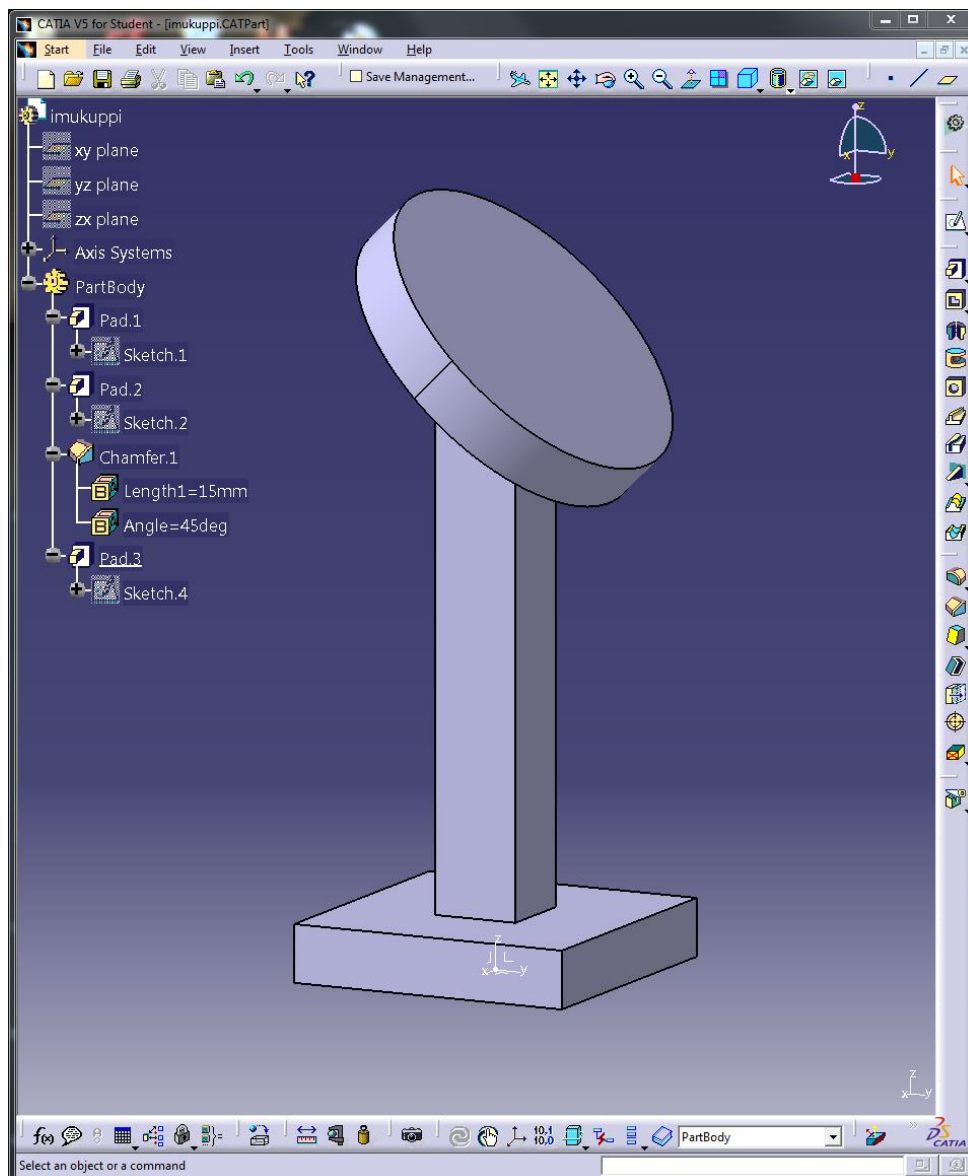
4.2 RobotStudio

RobotStudio on ABB Oy:n kehittämä offline-ohjelmointi- ja simulointiohjelma, jolla teollisuusrobotin ohjelmointia voidaan toteuttaa tietokoneella vaikkapa toimistosta käsin, häiritsemättä tuotantoa. RobotStudio-ohjelmasta löytyvät työkalut, joilla voidaan harjoitella, ohjelmoida ja optimoida robottisysteemiä. RobotStudio käyttää ABB:n VirtualController-ohjelmaa, joka on tarkka kopio ohjelmistosta, jolla tosielämässä tuotannossa olevia robotteja ohjelmoidaan. Tämän ansiosta RobotStudio-ohjelmalla saadaan tehtyä realistisia simulointeja käyttäen todenmukaisia robottisovelluksia ja rakennetiedostoja. [8.]

5 Tuotantosolun mallintaminen ja simulointi

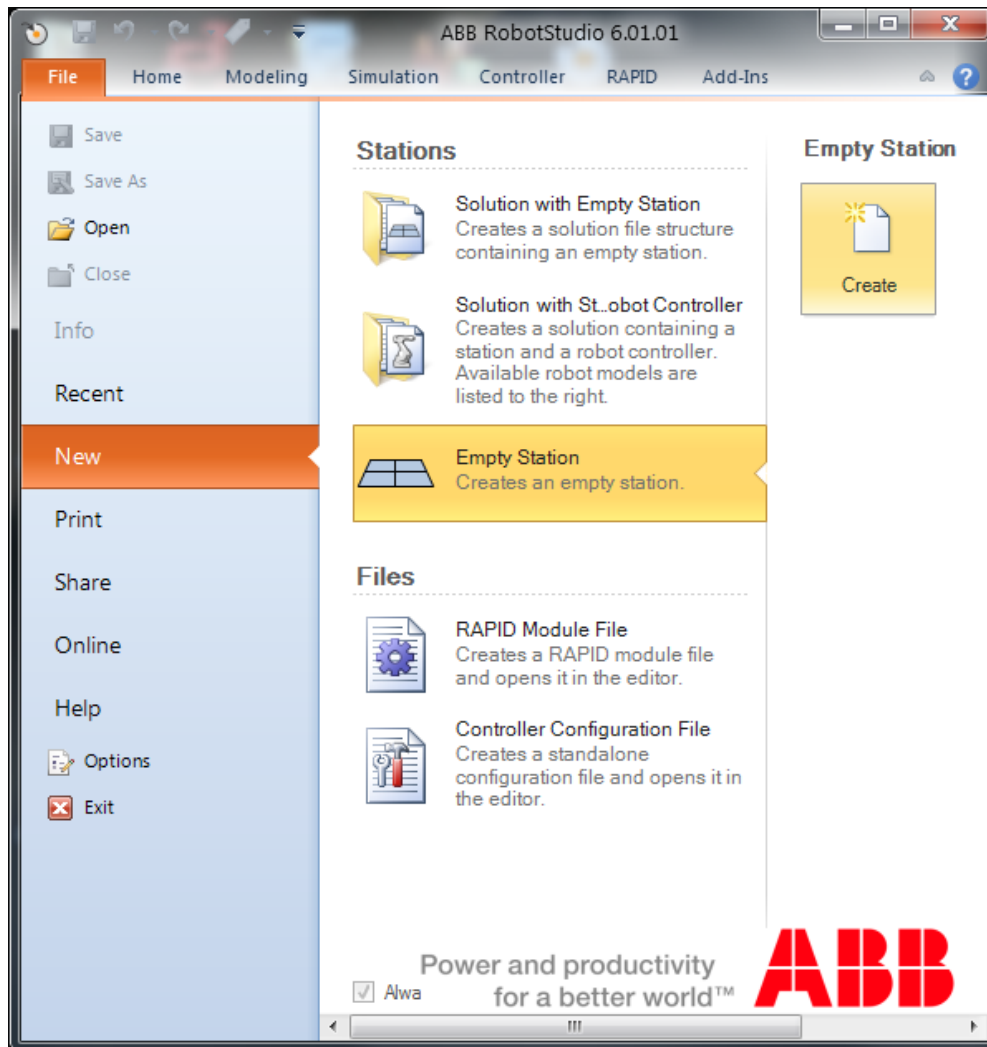
5.1 Robotin työkalun mallintaminen ja määrittäminen

Insinööriyössä Robotstudio-simulointiohjelmalla tehtävässä tuotantosolun mallintamisessa ja simuloinnissa käytettiin teollisuusrobotin työkaluna CATIA-mallinnusohjelmalla mallinnettua imukupitarrainta. Kuvassa 4 on mallinnetun työkalun 3D-malli, ja sen työpiirros on liitteenä (liite 1).



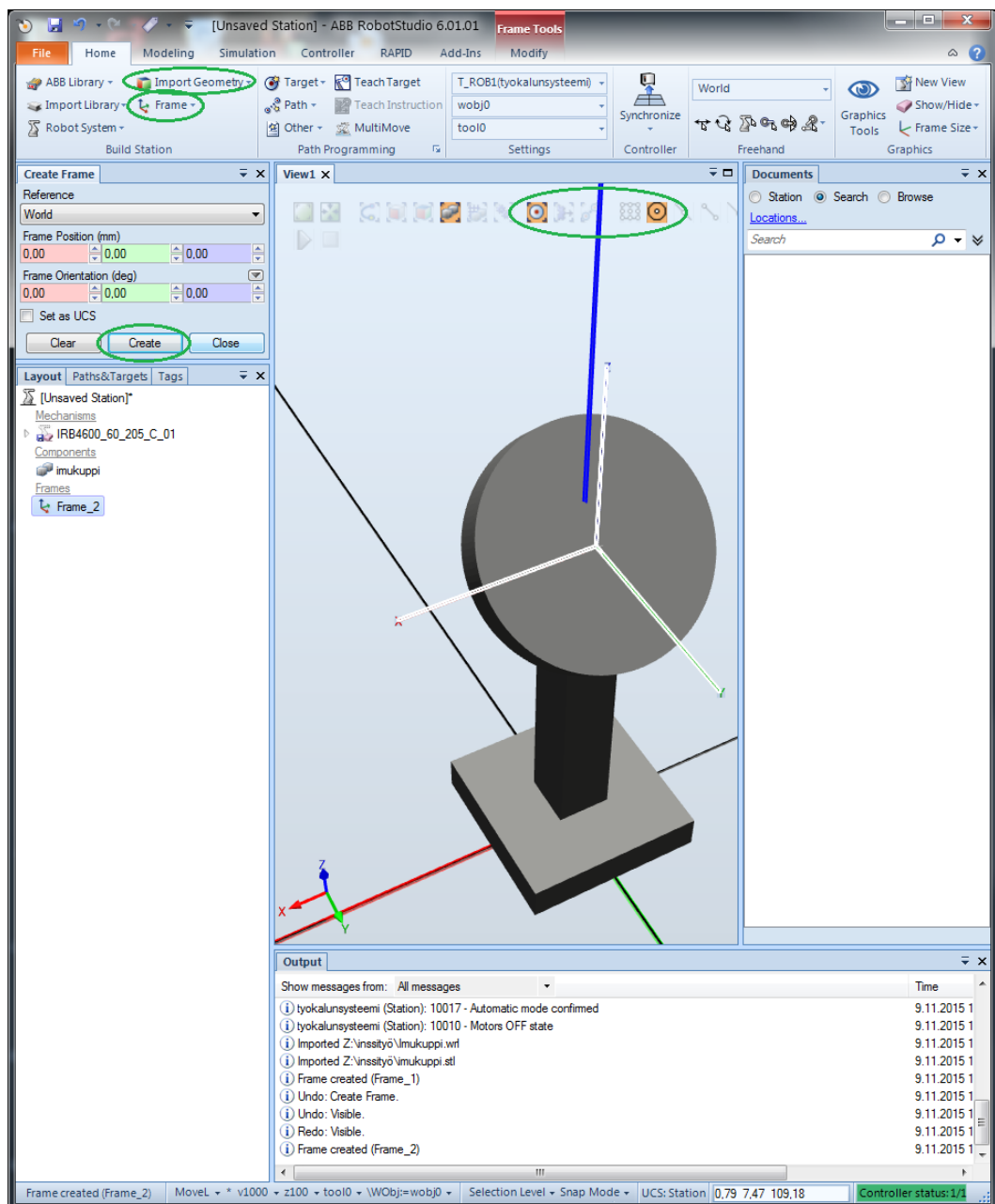
Kuva 4. Imukupitarraimen 3D-malli.

Robotstudio-ohjelmalla robotin työkalun määrittäminen aloitetaan avaamalla tietokoneen työpöydältä ohjelman pikakuvake. Aloitusnäky on kuvan 5 mukainen. Tyhjä työasema luodaan valitsemalla File-välilehden New-valikosta Empty Station -valinta ja painamalla Create-näppäintä.



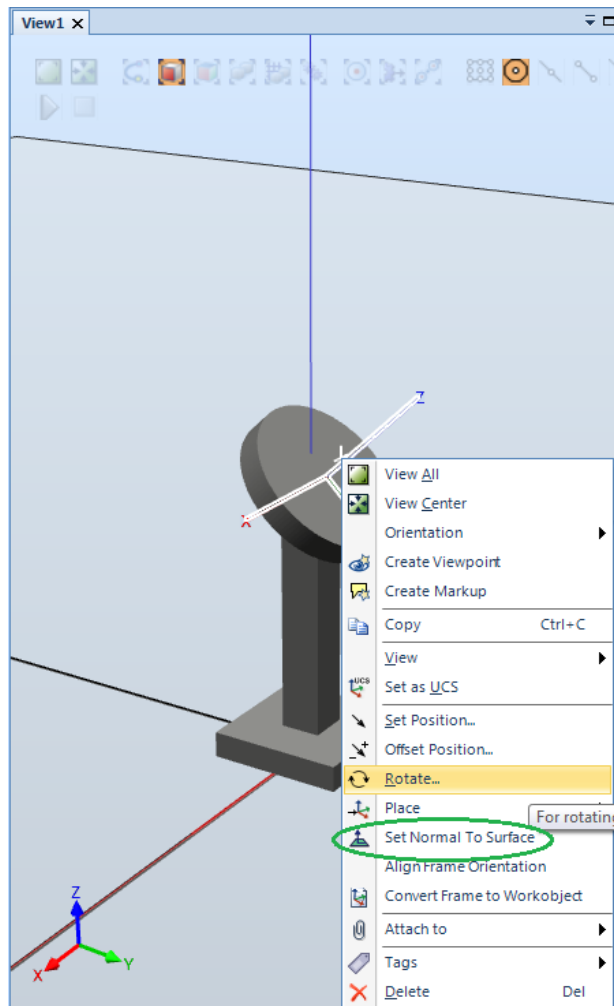
Kuva 5. Tyhjän työaseman luominen.

Ohjelmaan aukeaa työaseman näkymä, johon 3D-mallinnettu työkalu tuodaan. CATIA-mallinnusohjelmalla mallinnettu imukuppitarraintyökalu noudetaan Home-välilehden Import Geometry -valikosta, johon se tallennettiin CAD-tiedostona. Työkalun tarrainpintaan luodaan koordinaatisto valitsemalla Frame-valikosta Create Frame -komento. Koordinaatisto saadaan keskelle tarrainpintaa valitsemalla työaseman View1-ikkunan työkaluriviltä Target/Frame Selection - sekä Snap Center -työkalut ja valitsemalla työkalun tarrainpinnan keskipiste. Koordinaatiston origon koordinaatit ilmestyvät Create Frame -kenttään, ja ne tallennetaan painamalla Create-näppäintä (kuva 6).



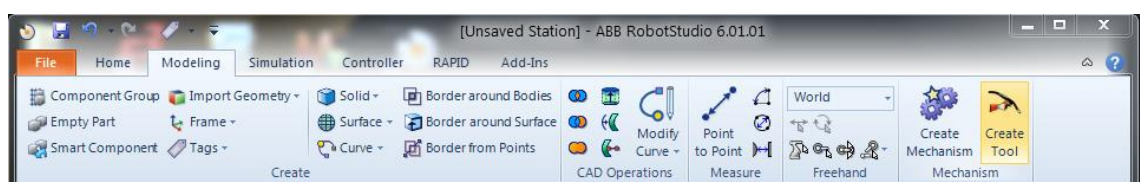
Kuva 6. Imukuppitarraimen koordinaatiston luonti.

Seuraavaksi työkalun koordinaatisto ja tarrainpinnan normaali asetetaan samansuuntaiseksi valitsemalla Rotate- ja Set Normal to Surface -komennot kuvan 7 mukaisesti. Tämän jälkeen työkalun koordinaatisto on halutussa kohdassa ja suunnassa.



Kuva 7. Työkalun koordinaatiston suunnan ja kulman asetus.

Imukuppitarrain määrittellään vielä työkaluksi valitsemalla Modeling-välilehdeltä Create Tool -toiminto (kuva 8). Tällöin avautuvat kaksi ikkunaa, joissa määrittellään työkalulle viimeiset asetukset.



Kuva 8. Create Tool -toiminto.

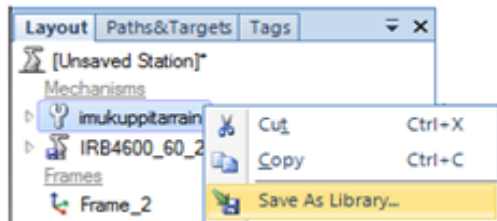
Ensimmäisessä ikkunassa (kuva 9) työkalu nimetään ja valitaan mallinnettu osa, jota käytetään työkaluna. Lisäksi työkalulle voidaan määrittää massa ja painopiste sekä hi-tausmomentti. Seuraavaan ikkunaan siirrytään Next-näppäimellä.

Kuva 9. Tool Information -ikkuna.

Toisessa ikkunassa (kuva 10) imukuppitarraimelle nimetään ja määritetään työkalun keskipiste (TCP, Tool Center Point). Valitaan käytettävä koordinaatisto, joka määritettiin aiemmin työkalulle. Asetukset tulevat voimaan painamalla Done-näppäintä.

Kuva 10. TCP Information -ikkuna.

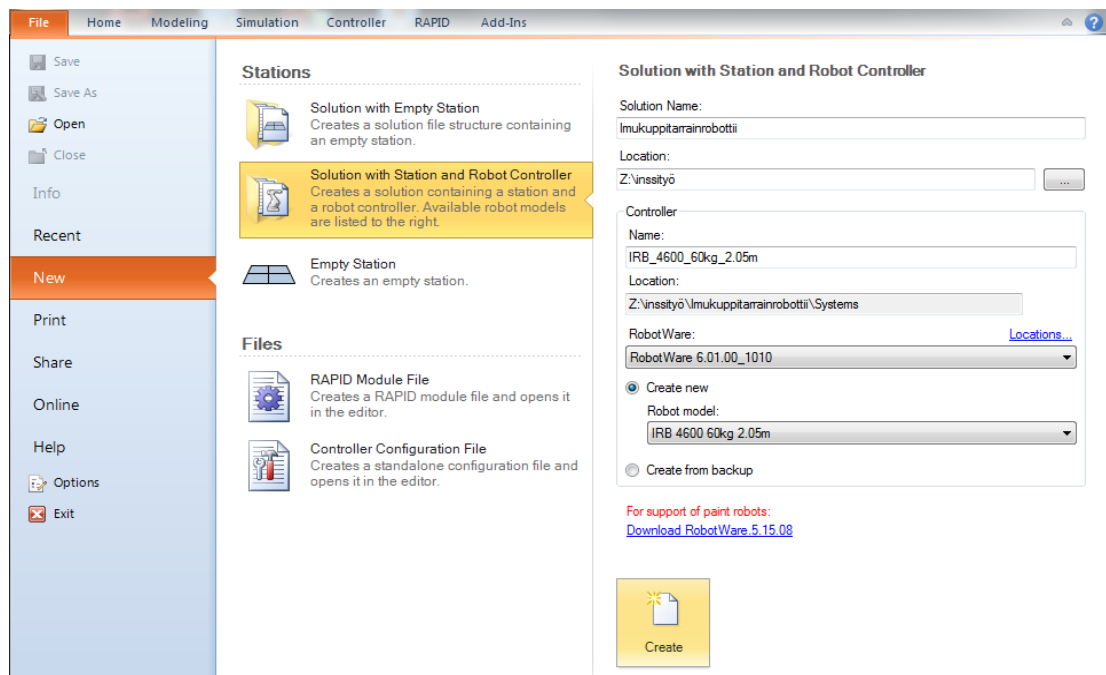
Luotu imukuppitarraintyökalu voidaan tallentaa haluttuun kansioon kirjastotiedostoksi tulevia projekteja varten valitsemalla Layout-kentässä imukuppitarraimen Save As Library -komento kuvan 11 mukaisesti.



Kuva 11. Työkalun tallentaminen kirjastotiedostoksi.

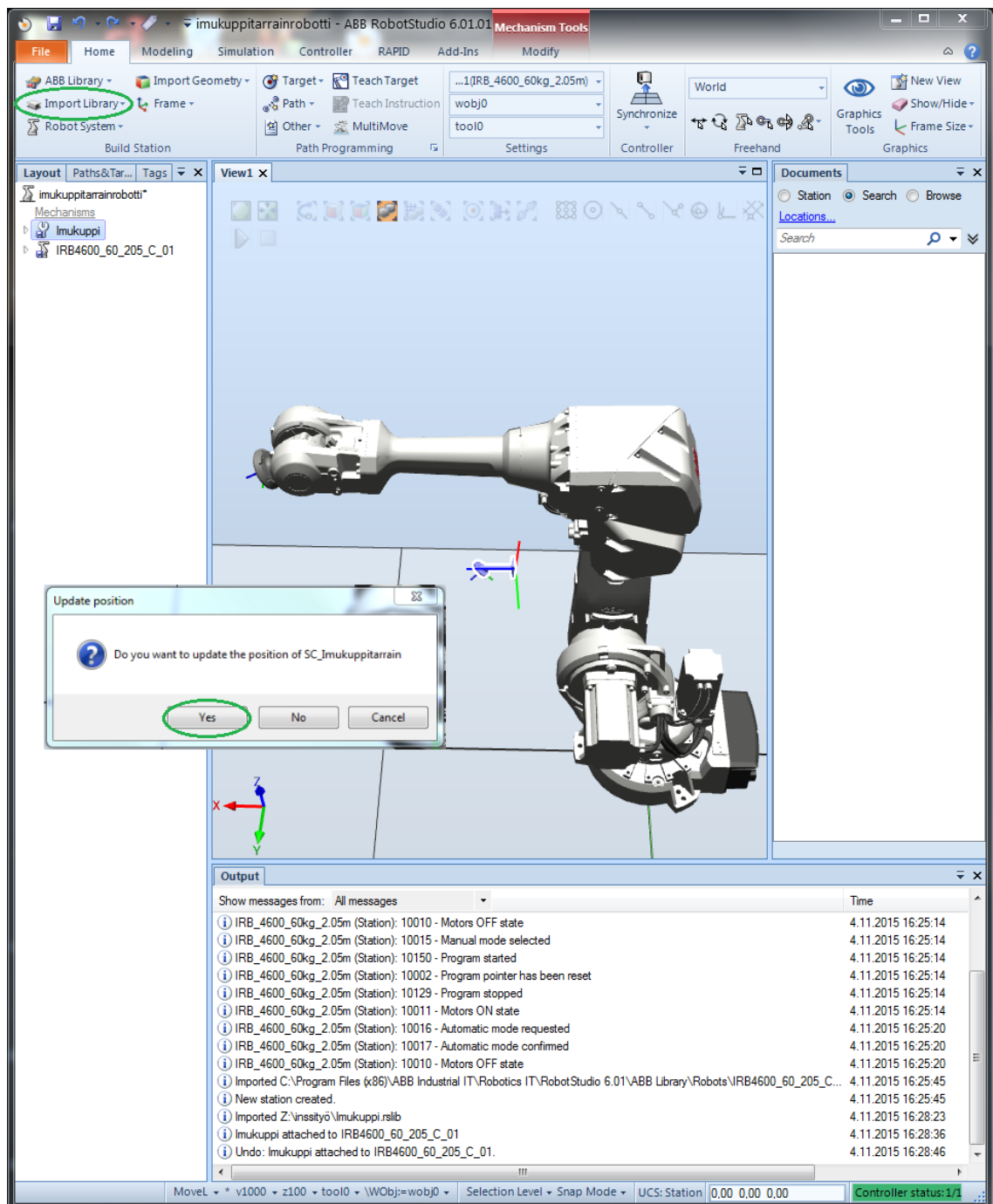
5.2 Robotin työaseman luonti ja tartuntaominaisuuden ohjelmointi työkaluun

Tuotantosolun mallintaminen aloitetaan valitsemalla RobotStudio-ohjelman aloitusnäytön File-välilehden New-valikosta vaihtoehto Solution with Station and Robot Controller. Tämä valinta luo kokonaisuuden, joka sisältää työaseman ja virtuaalisen robottiohjaimen. Avautuvassa kentässä työasema ja robottiohjain nimetään ja niille valitaan tallennuskansiot sekä valitaan listasta haluttu teollisuusrobottimalli. Työasema ja robottiohjain aktivoidaan painalla Create-painiketta (kuva 12).



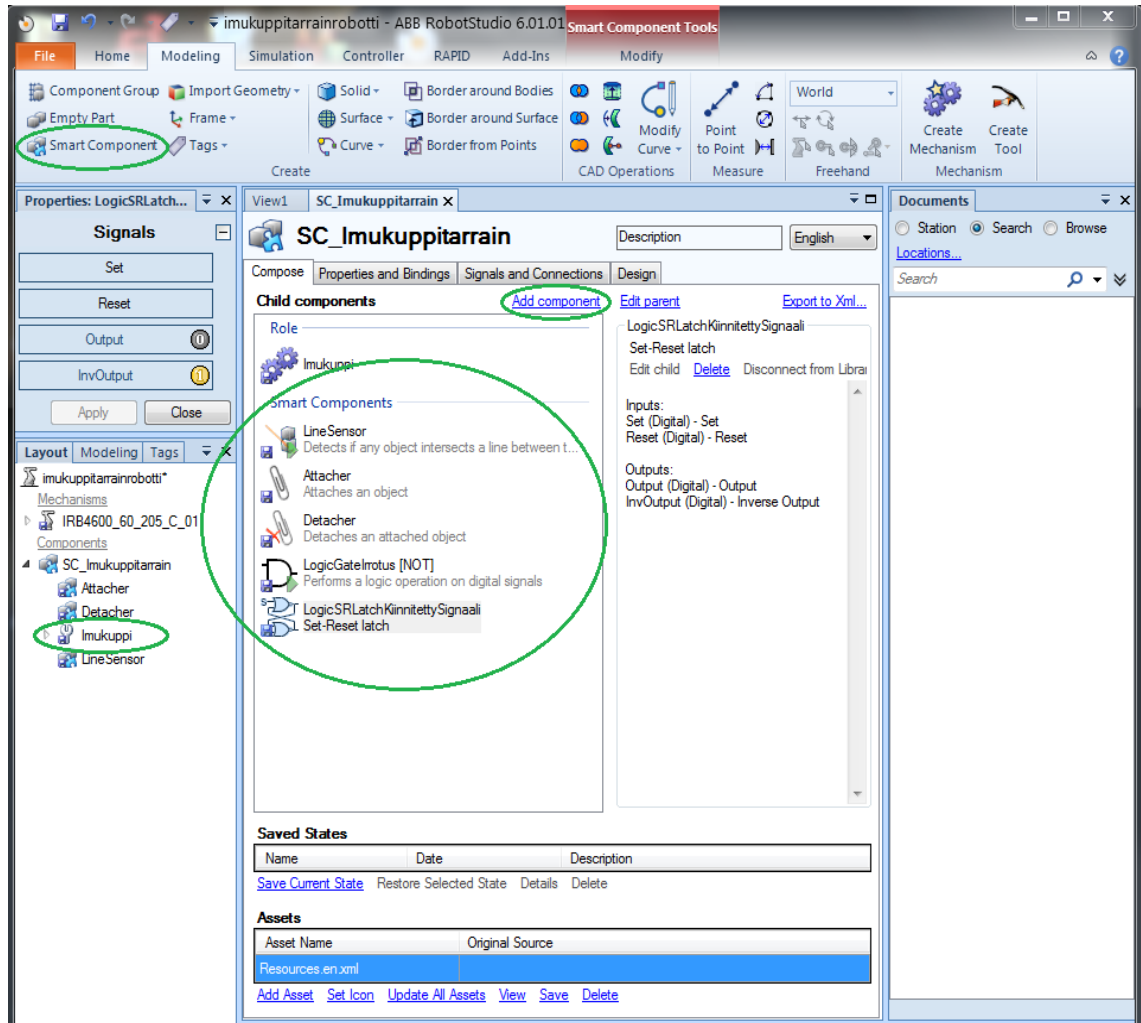
Kuva 12. Työaseman ja robottiohjaimen luonti.

Valittu teollisuusrobotti avautuu työasemanäkymään ja robottiohjain aktivoituu, kun ikkunan oikeassa alakulmassa oleva Controller status -palkki muuttuu vihreäksi. Kun ohjain on aktivoitunut, haetaan aiemmin luotu imukupitarraintyökalu Home-välilehden Import Library -kansiosista. Työkalu liitetään robottiin viemällä työkalun kuvake Layout-kentän robotin kuvakkeen päälle ja valitsemalla avautuvassa ikkunassa Yes-painike, jolloin työkalun paikka päivittyy ja työkalu kiinnittyy robotin työkalulaippaan (kuva 13).



Kuva 13. Työkalun kiinnittäminen robottiin.

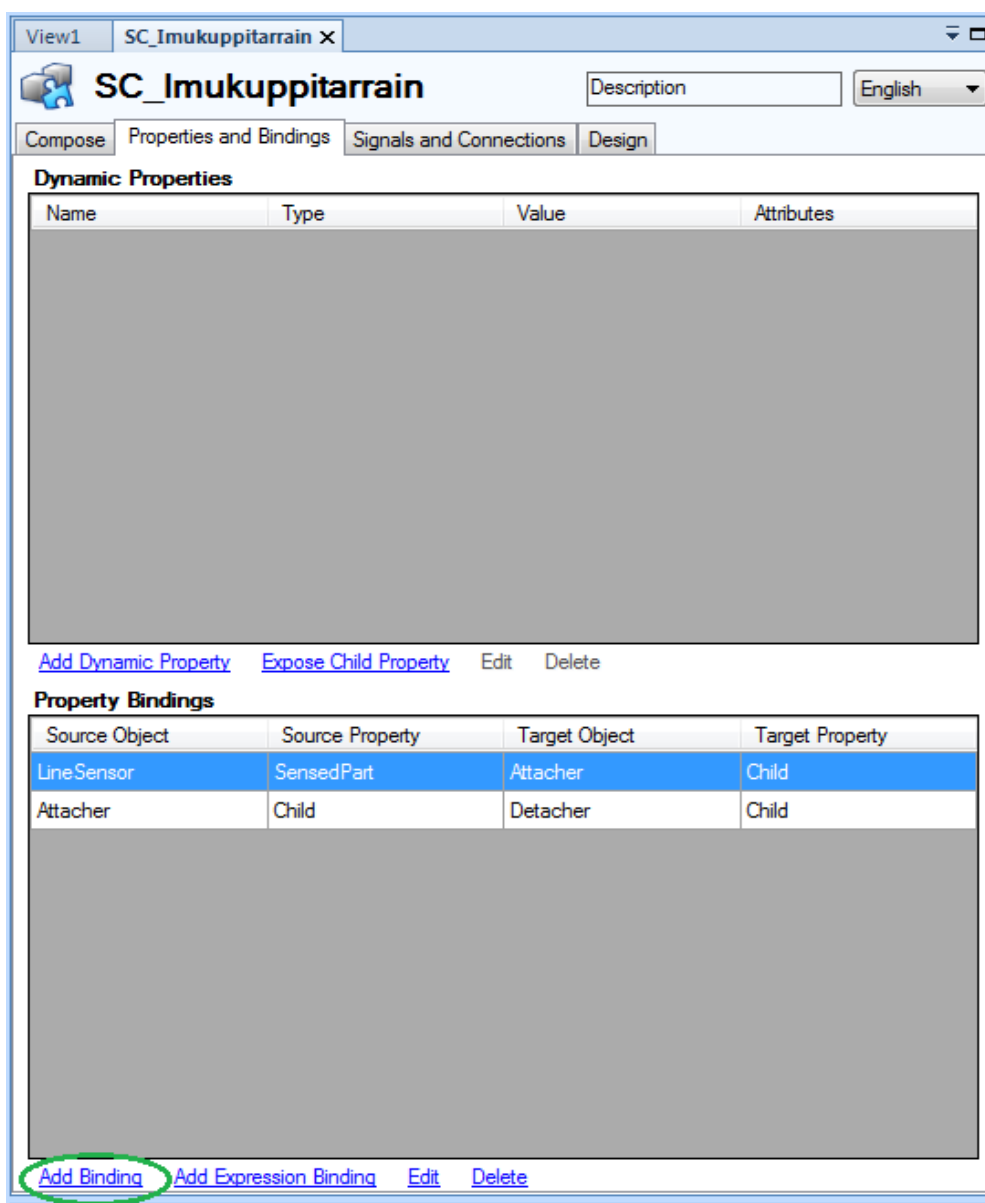
Imukupittarraimeen luodaan tartuntaominaisuus käyttämällä Modeling-välilehden Smart Component -toimintoa. Ohjelmaan avautuu Smart Component -ikkuna ja Layout-kenttään Smart Component -kuvake, joka nimettiin SC_Imukupittarraimeksi. Aiemmin tehdyn imukupittarraimen kuvake viedään tähän Smart Component -kuvakkeeseen, jolloin siitä tulee käytettävä komponentti (kuva 14).



Kuva 14. Smart Component -toiminto.

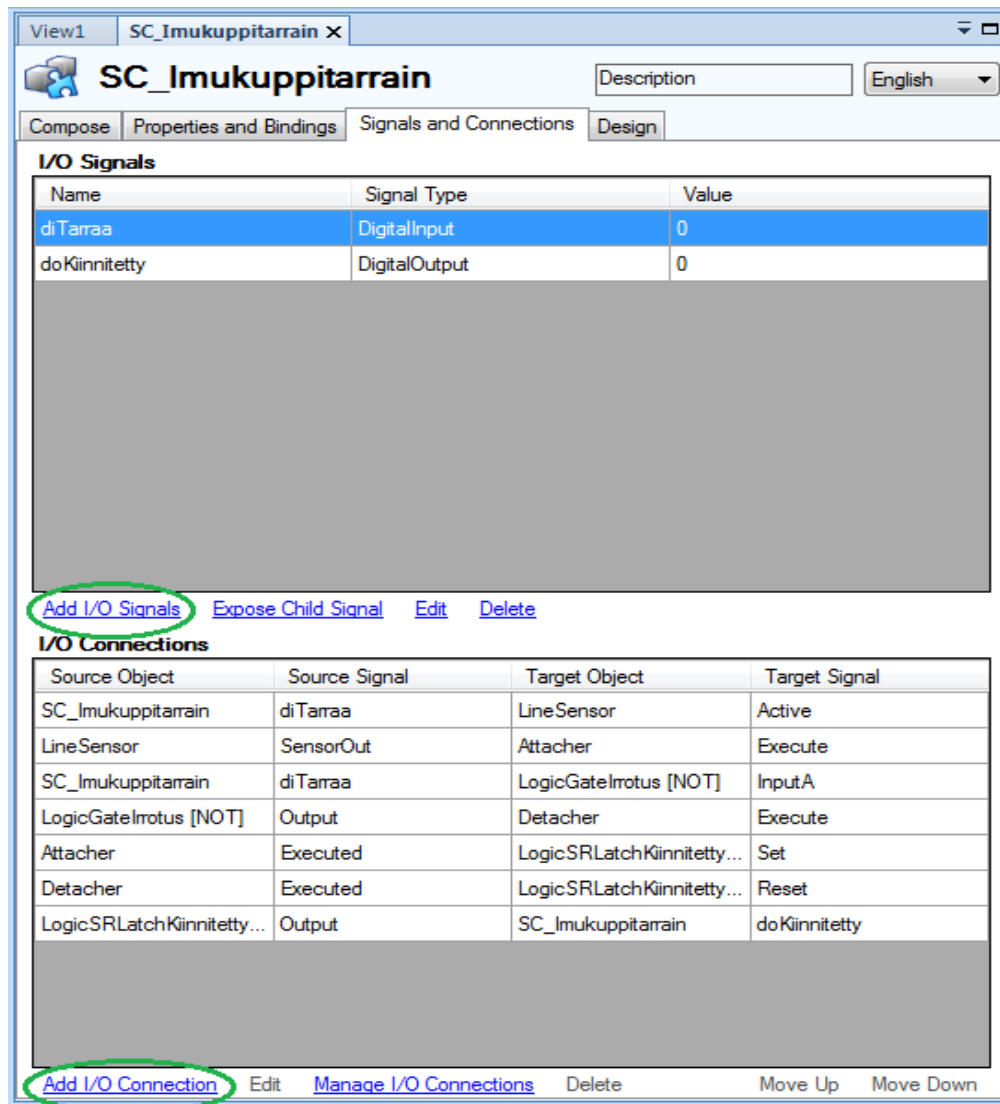
Compose-välilehden Add component -valikosta valitaan muut työkalulle tarvittavat komponentit. Imukupittaraimelle valittiin LineSensor-, Attacher-, Detacher-, LogicGate- sekä LogicSRLatch-komponentit. LineSensor-komponentti tunnistaa kappaleen, johon työkalulla halutaan tarrata. Attacher-komponentilla kappale kiinnittyy imukupittarraimeen, Detacher-ominaisuus puolestaan irrottaa kappaleen tarraimesta. LogicGate-toiminnolla luodaan toimintalogiikka digitaalisille signaaleille. LogicSRLatch-komponentilla logiikkaan tehdään kierto.

Properties and Bindings -välilehdellä Add Binding -valikossa muodostetaan sidokset toimintojen välille. Sidokset saadaan muodostettua valitsemalla komponentille Source Object, Source Property, Target Object sekä Target Property. Source Object -kenttään valitaan varsinainen lähdekomponentti tai toiminto, Source Property -kenttään komponentin ominaisuus, Target Object -kenttään kohdekomponentti ja Target Property -kenttään kohdekomponentin ominaisuus (kuva 15).



Kuva 15. Properties and Bindings -välilehti.

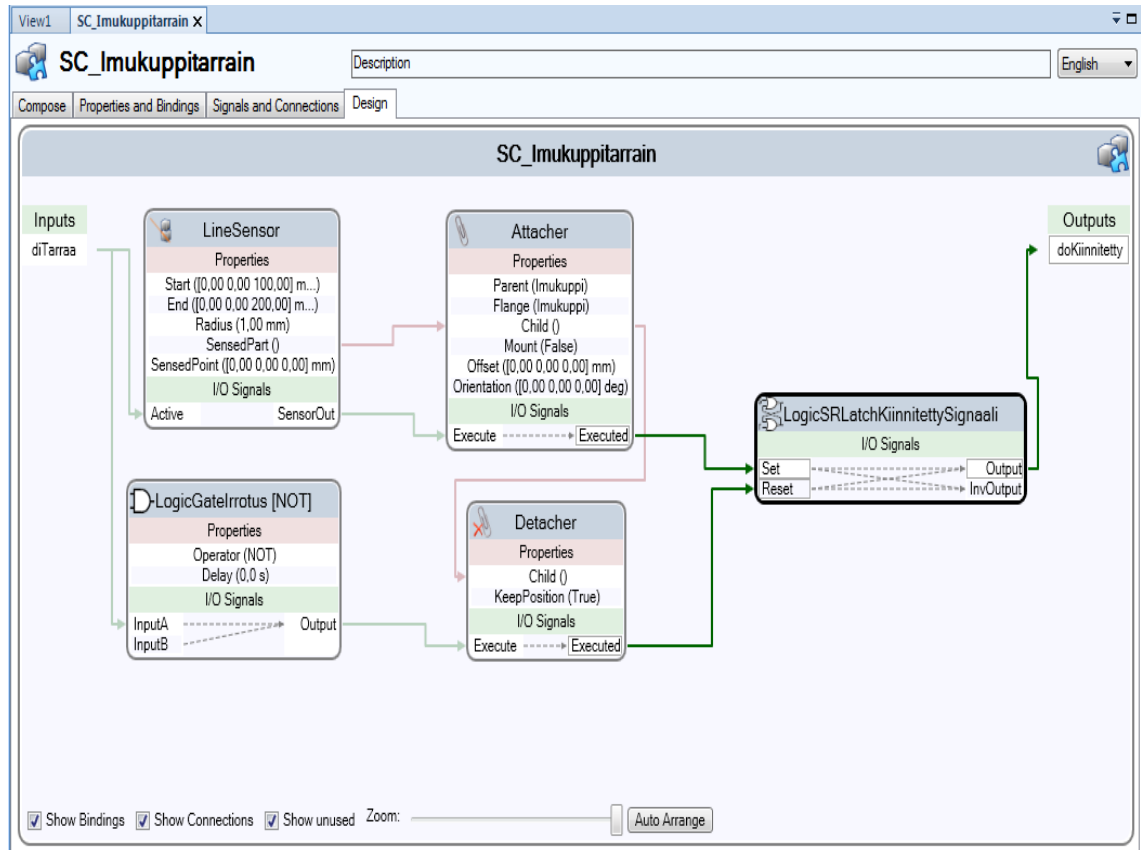
Signals and Connections -välilehdellä Add I/O Signals -valikossa imukupittaraimelle tehdään ohjaussignaali, nimetään ne ja määritellään signaalityyppi sekä arvo. Insinöö- rityössä signaalit nimettiin diTarraa- ja doKiinnitetty-signaaleiksi. DiTarraa-signaali mää- ritettiin DigitalInput-tyyppiseksi. Tällöin signaali toteuttaa halutun toiminnon, tässä ta- pauksessa tarttumisen. DoKiinnitetty-signaali puolestaan valittiin DigitalOutput-tyyp- piseksi. Tämä osoittaa DigitalInput-signaalin, tässä tapauksessa diTarraa-signaalin seu- rauksen eli tarttumisen seurauksena kappale kiinnittyy imukupittaraimen (kuva 16).



Kuva 16. Signals and Connections -välilehti.

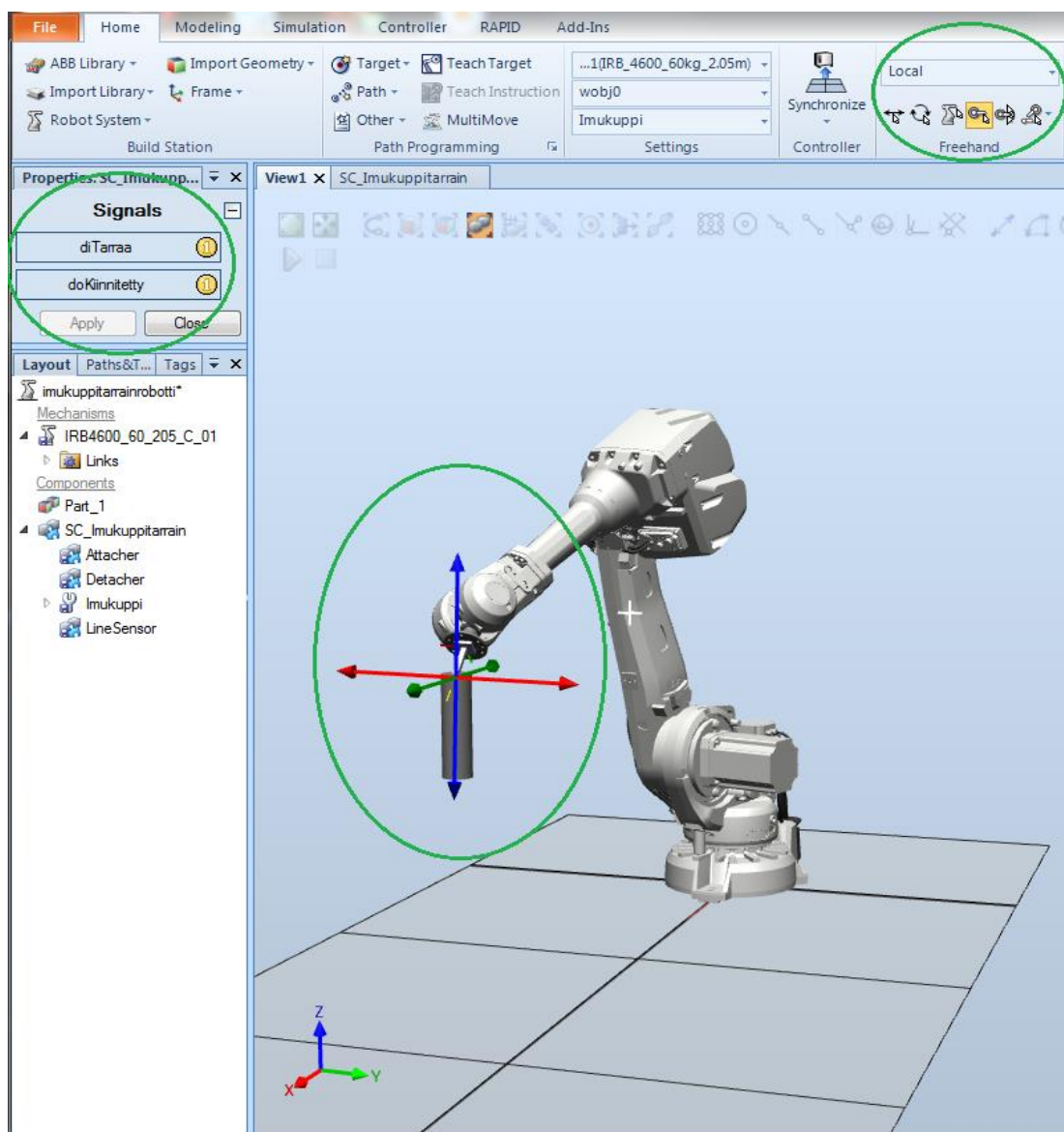
Add I/O Connection -valikosta valitaan Source Object -, Source Signal -, Target Object - sekä Target Signal -kenttiin lähdekomponenttien/-toimintojen ja kohdekomponenttien sekä näiden signaalien väliset kytkennät.

Design-välilehdelle saadaan näkyviin työkalun komponenttien rakenne graafisena näkymänä. Näkymä sisältää komponentit ja niiden väliset sisäiset kytkennät, ominaisuudet ja sidokset. Komponenttien paikkaa voidaan järjestellä näkymässä, ja niiden paikat tallentuvat aseman mukana (kuva 17).



Kuva 17. Design-välilehti.

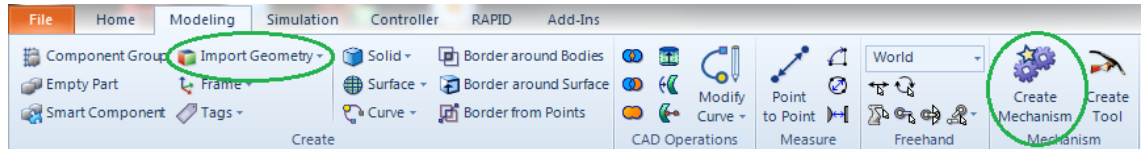
Työkalan toimintaa päästään testaamaan View1-työasemanäkymässä luomalla työkalupale Modeling-välilehden Solid-valikosta. Insinööriyössä valittiin kappale sylinterimäiseksi. Kappaleelle annetaan halutut mitat eli halkaisija ja pituus. Robottikäsiä ohjataan Freehand-valikon toiminnoilla. Valikosta valitaan käytettävä koordinaatisto, minkä mukaan robotti liikkuu, ja se, onko liike lineaarista koordinaatiston akseleiden suuntaisesti vai akseleiden ympäri pyörivää. Robotin imukupittarrain viedään kappaleen luo niin, että se koskettaa kappaleen pintaa. Aktivoidaan aiemmin luoduista ohjussignaaleista diTarraa-signaali, jolloin tarraintyökalu tarttuu kappaleeseen. Nyt robottia voidaan liikuttaa niin, että kappale pysyy kiinni imukupityökalussa. Kun diTarraa-signaali-painikkeesta painetaan uudestaan, tarraintyökalu päästää irti työkalupaleesta (kuva 18).



Kuva 18. Tarttuminen kappaleeseen ja robotin liikuttelu.

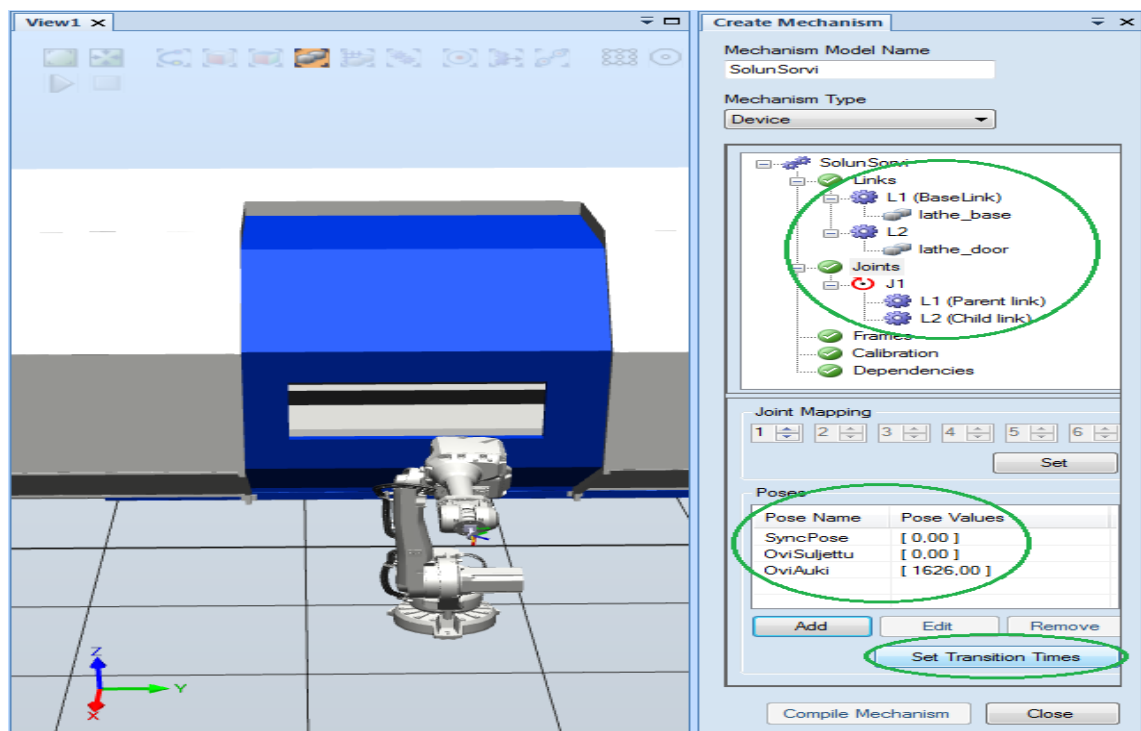
5.3 Sorvimekanismin luominen

Tuotantosolussa käytettävän NC-sorvin ovimekanismi luodaan hakemalla työasemaan sorvin runko ja ovi erillisinä tiedostoina Modeling-välilehden Import Geometry -kansioista ja valitsemalla Create Mechanism -toiminto (kuva 19).



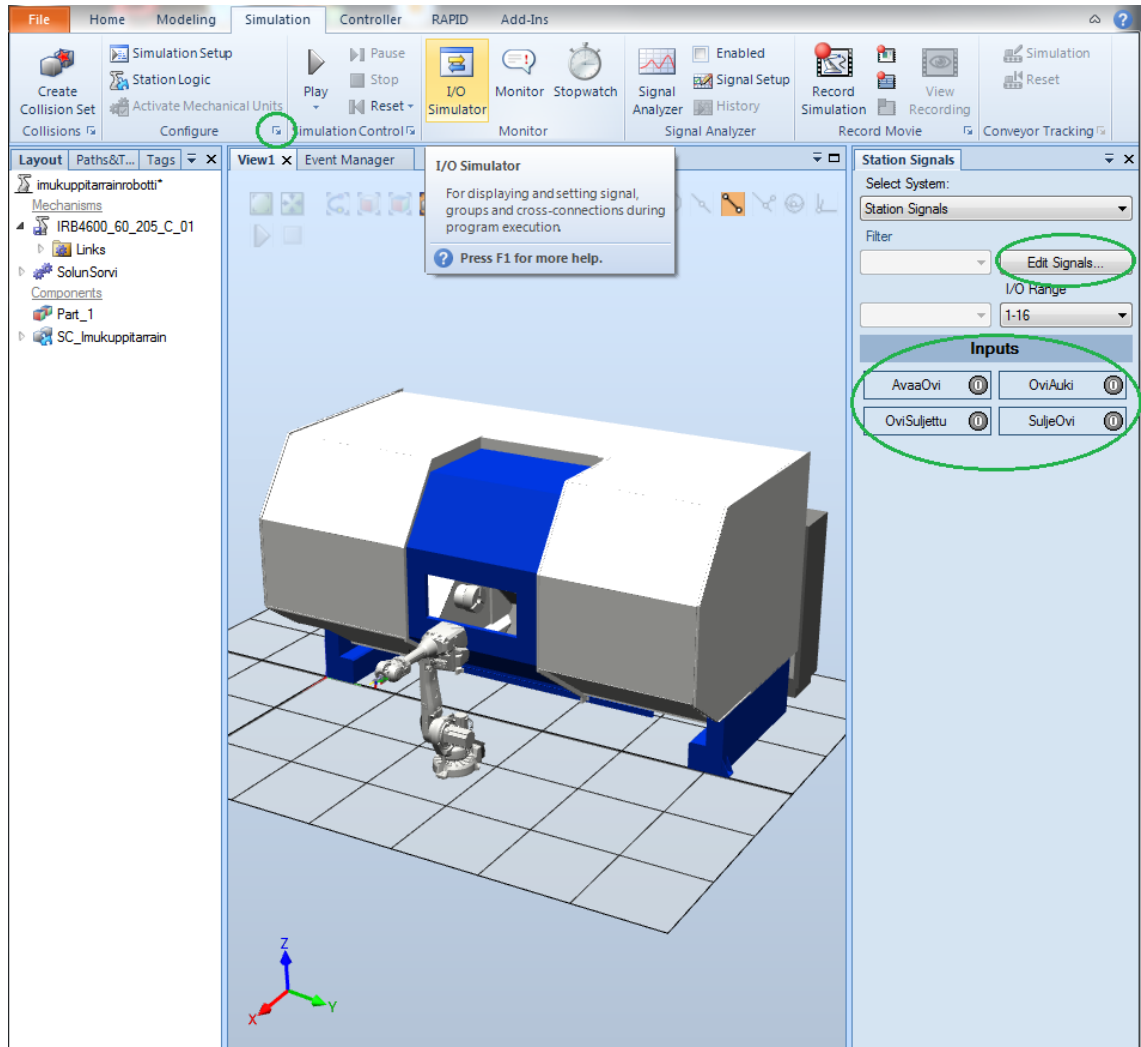
Kuva 19. Create Mechanism -toiminto.

Create Mechanism -ikkunassa valitaan mekanismin tyyppi ja nimetään se. Sorvimekanismille lisätään linkit sorvin ovelle ja rungolle Add Link - ja Add Joint -toiminnoilla. Sorvin oven asennot nimetään ja niille annetaan arvot Poses-ikkunan Add-painikkeella. Insinööriyössä OviSuljettu-asennossa ovi on suljettu ja sen arvo nolla. Ovi on auki arvolla 1 626 mm, joka on sorvin oviaukon leveys, ja asento on nimetty OviAuki-asennoksi. Set Transition Times -painikkeella sorvin ovelle määritetään siirtymäajat asennosta toiseen. Insinööriyössä ajat määritettiin OviAuki-asennosta OviSuljettu-asentoon (kuva 20).



Kuva 20. Create Mechanism -ikkuna.

Simulation-välilehden I/O Simulator -toiminnolla tehdään sorvin ovimekanismille ohjaussignaaleja Edit Signals -painikkeella. Signaalien asetuksia editoidaan vielä Event Manager -toiminnolla, joka löytyy Configure-valikosta (kuva 21).



Kuva 21. I/O Simulator -toiminto ja ohjaussignaaleja.

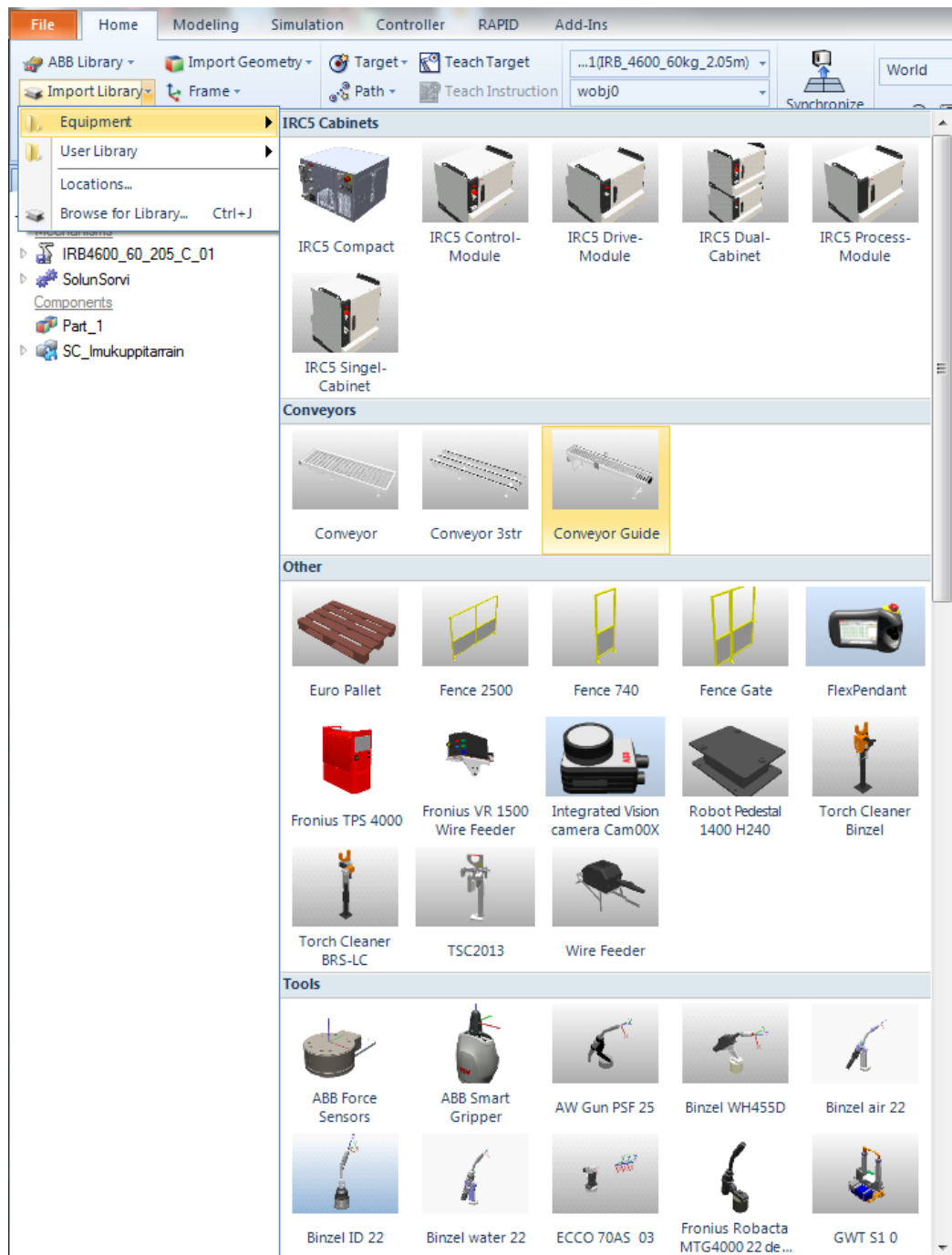
Event Manager -ikkunassa Add-painikkeella signaaleille annettiin kuvan 22 mukaiset asetukset. Signaalikomennoille ja toiminnoille määritettiin aktivointi, tyypit, systeemit, nimet ja parametrit.

Events		Activation	Trigger Type	Trigger System	Trigger Name	Trigger Parameter	Action Type	Action System	Action Name	Action Parameter	Time (s)
Add...	On	I/O	Station Signals	SuljeOvi	1	Multiple			Multiple		
	On	I/O	Station Signals	AvaaOvi	1	Multiple			Multiple		

Kuva 22. Event Manager -ikkuna.

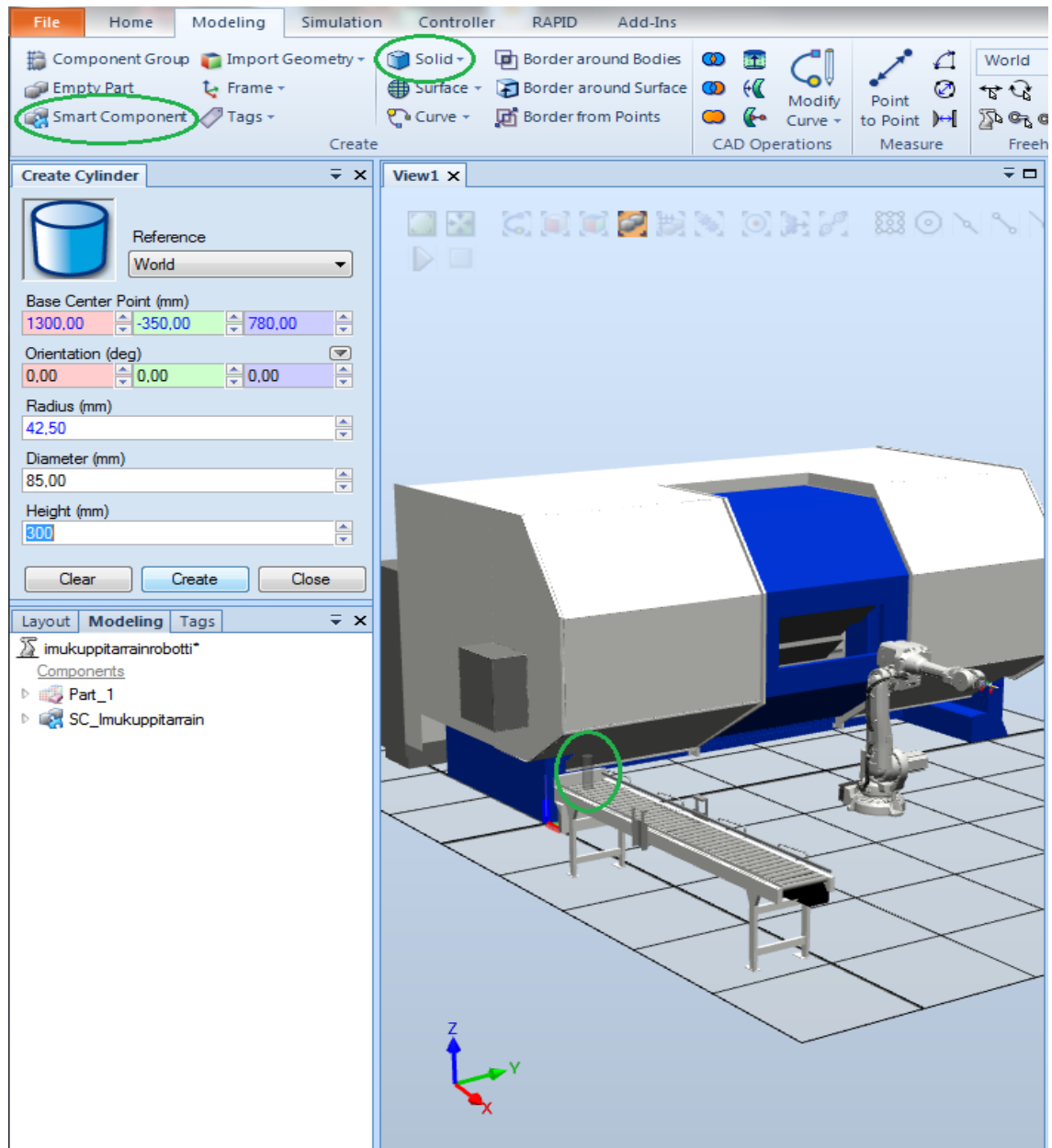
5.4 Liukuhihnamekanismin luominen

Liukuhihnamekanismin luonti aloitetaan hakemalla liukuhihna Import Library -kansion Equipment-valikosta (kuva 23). Liukuhihnalle tehdään työkappale Modeling-välilehden Solid-toiminnolla ja liukuhihnan mekanismi Smart Component -toiminnolla, jolla aiemmin tehtiin robotin imukupittaraintyökalu.



Kuva 23. Import Library -kansio sekä Equipment-valikko.

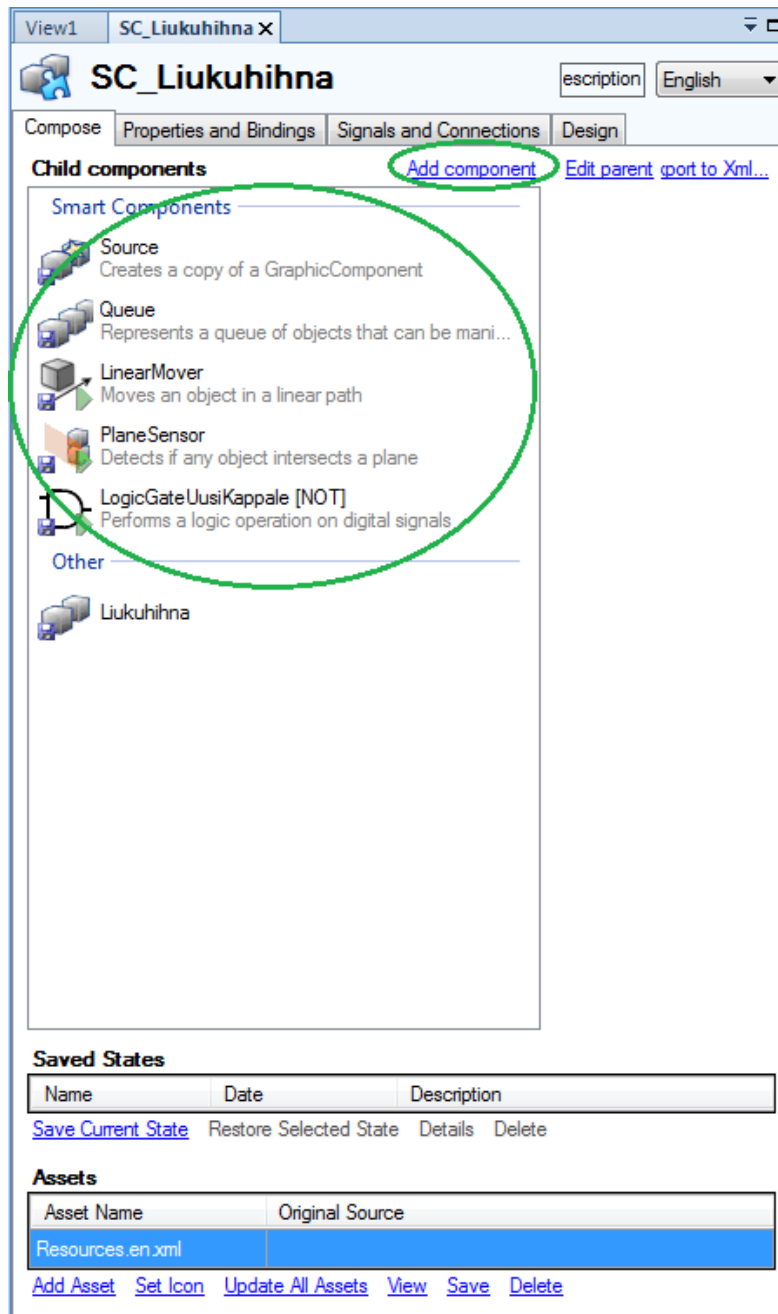
Valittiin Solid-valikosta sylinterimäinen kappale. Avautuvassa Create Cylinder -ikkunassa asetettiin koordinaatit niin, että kappale on liukuhihnalla. Työkappaleelle annettiin kuvan 24 mukaiset mitat eli halkaisija sekä korkeus. Create-näppäintä painamalla luodaan kappale liukuhihnalle.



Kuva 24. Create Cylinder -ikkuna.

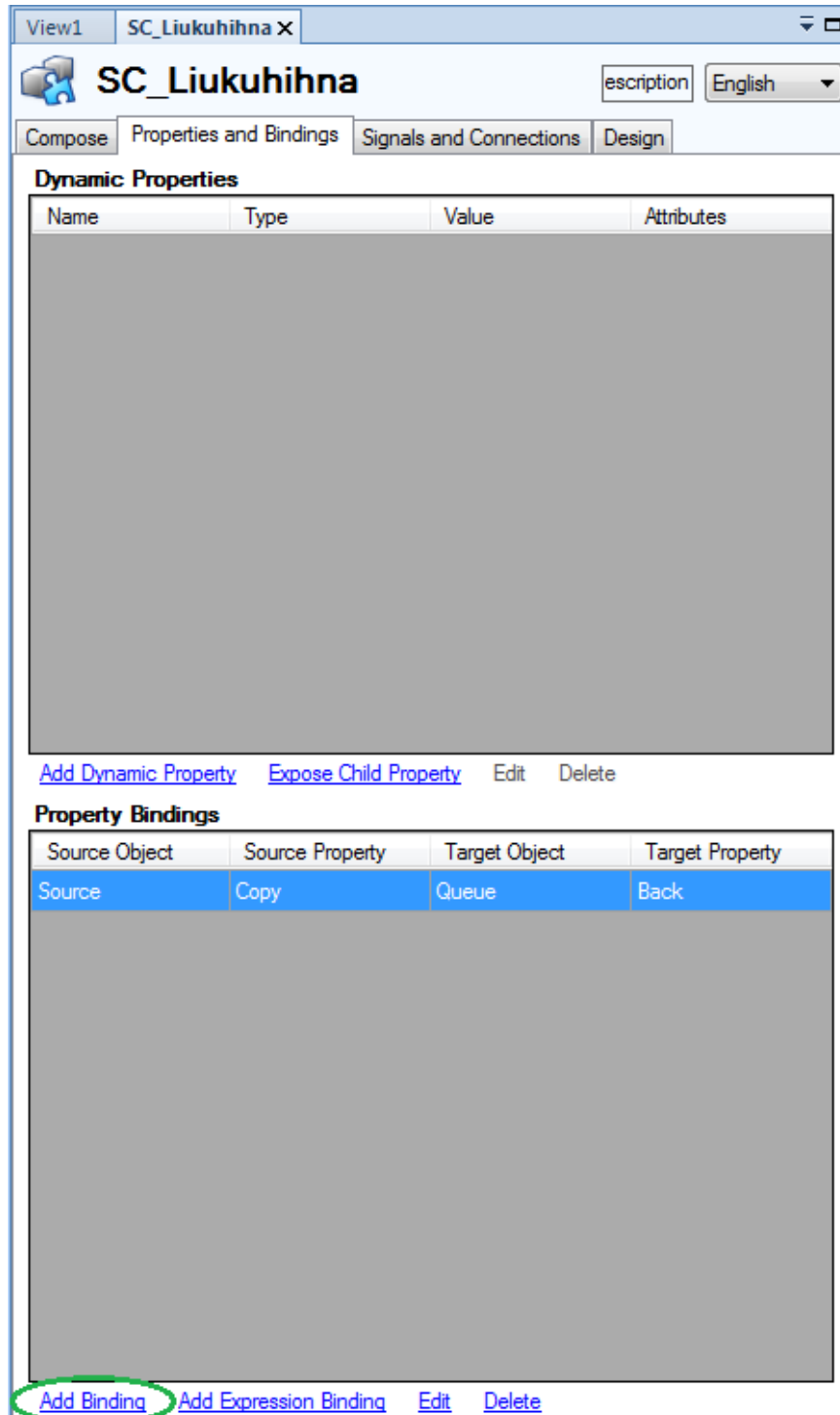
Liukuhihnamekanismi tehdään Modeling-välilehden Smart Component -toiminnolla, jolla aiemmin luotiin tartuntaominaisuus robotin imukupittaraimelle. Liukuhihnan mekaniisin komponentit, sidokset, signaalit ja kytkennät eroavat kuitenkin täysin työkalun vastaavista.

Compose-välilehden Add component -valikosta valitaan liukuhihnassa tarvittavat komponentit eli Source-, Queue-, LinearMover-, PlaneSensor- sekä LogicGate-toiminnot. Source-toiminto monistaa kappaleen liukuhihnalla ja Queue-toiminto edustaa kappalejonoa, jota voidaan käsitellä ryhmänä. LinearMover-komponentilla työkappaleet saadaan liikkumaan liukuhihnalla suoraviivaista rataa, PlaneSensor-komponentti puolestaan tunnistaa ja pysäyttää hinnan, jos kappale osuu sensoripintaan. LogicGate-komponentti esittää signaaleiden logiikan (kuva 25).



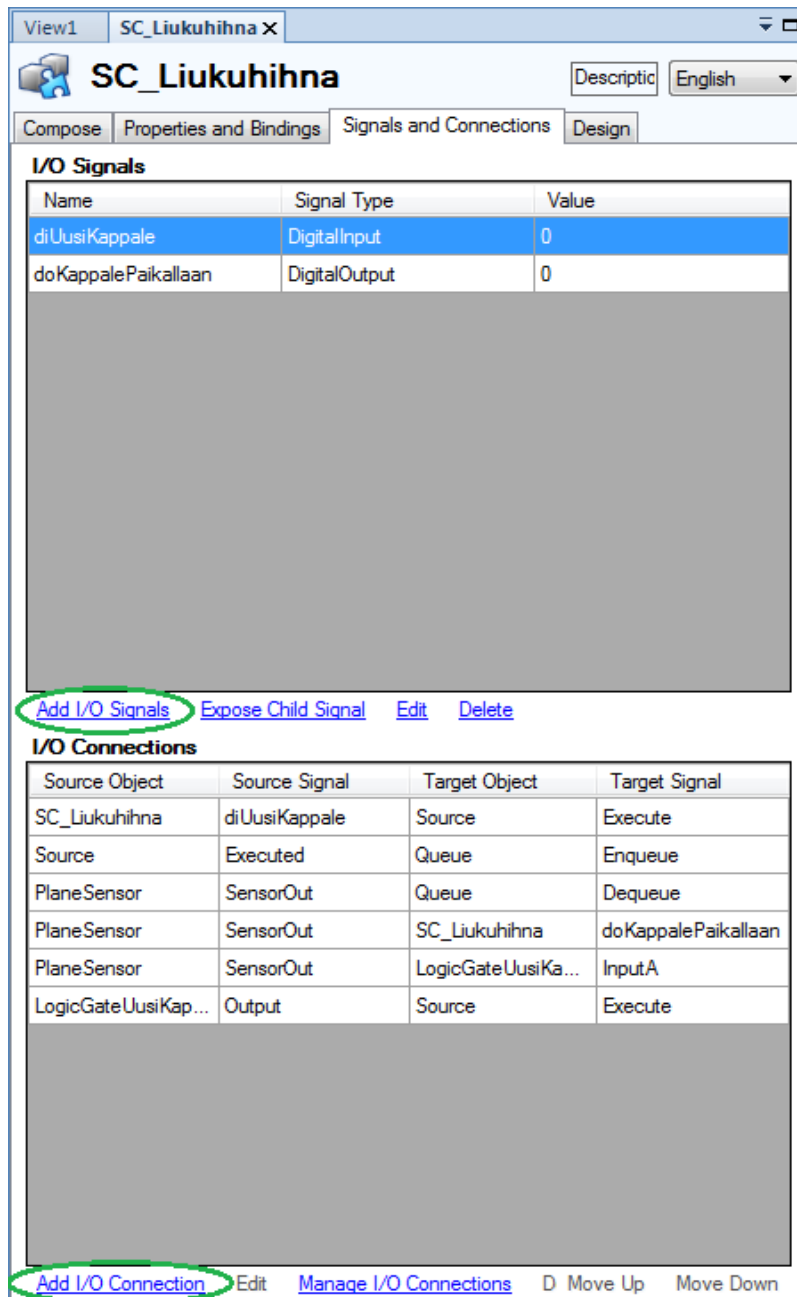
Kuva 25. Liukuhinnan komponentit.

Properties and Bindings -välilehdellä Add Binding -valikosta lisätään tarvittavat sidokset liukuhinnan komponenttien välille. Insinööriyössä liukuhihnalle muodostettiin vain yksi sidos kuvan 26 mukaisesti.



Kuva 26. Liukuhinnan komponenttien väliset sidokset.

Signals and Connections -välilehdessä Add I/O Signals -valikossa määritetään liukuhinnan ohjaussignaalit. Insinööriyössä liukuhinnaa ohjaavat signaalit nimettiin diUusiKappale- ja doKappalePaikallaan-signaaleiksi. Add I/O Connection -valikosta valitaan signaalien ja liukuhinnan komponenttien väliset kytkennät (kuva 27).

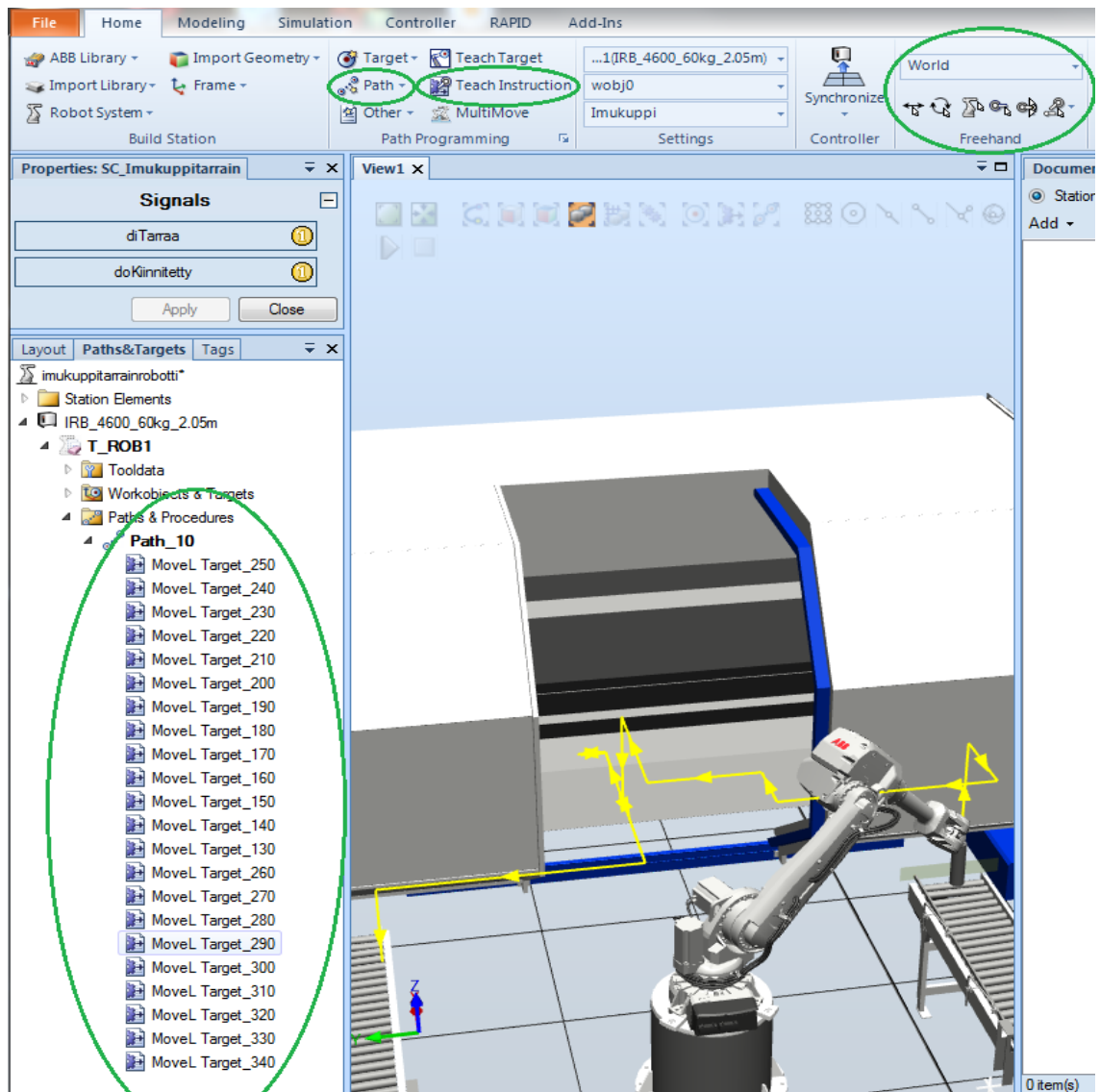


Kuva 27. Liukuhinnan ohjaussignaalit ja signaalien kytkennät.

Tämän jälkeen työkappale saadaan liikkumaan liukuhinnaa pitkin diUusiKappale-signaalin painikkeella ja kappale pysähtyy sensorin osoittamaan pintaan, kunnes kappale nostetaan hinnalta, jolloin hihna syöttää uuden kappaleen liikkeelle.

5.5 Robotin liikeratojen määrittäminen

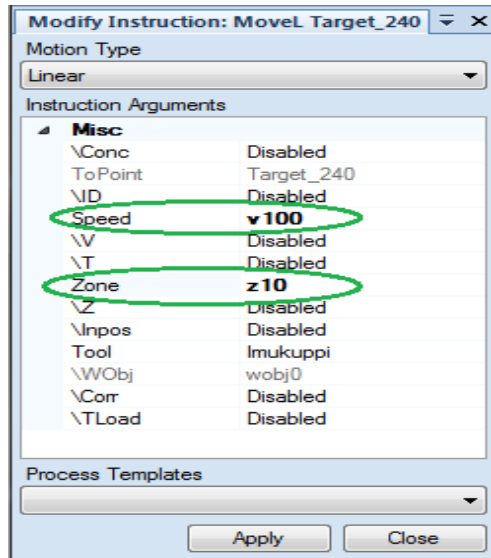
Teollisuusrobotin liikkeiden määrittäminen tehdään valitsemalla Home-välilehden Path Programming -valikosta Path-toiminto ja sieltä Empty Path -vaihtoehto. Tällä saadaan luotua tyhjä polku robotin liikerataa varten. Robottia ohjataan haluttuihin pisteisiin työasemassa Freehand-valikon toiminnoilla. Jokainen piste saadaan tallennettua osaksi liikeradan polkua Path Programming -valikon Teach Instruction -toiminnolla (kuva 28).



Kuva 28. Liikeradan polun ja polun pisteiden määrittäminen.

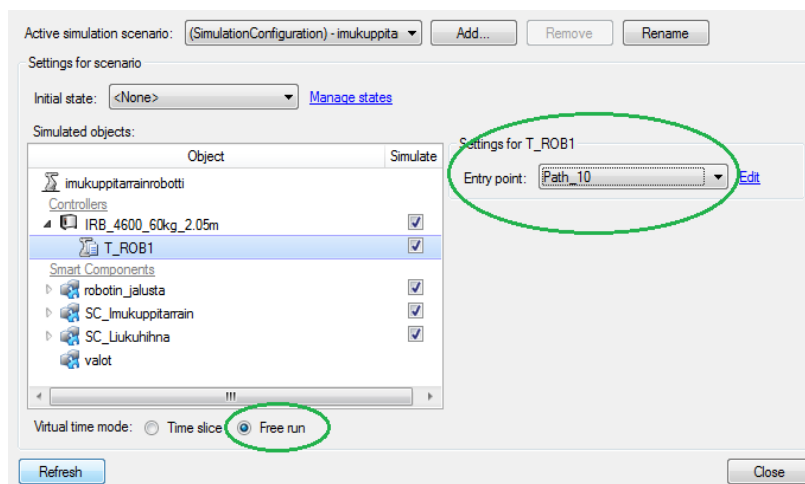
Insinööriyössä robotin liikeradan polku kulkee liukuhihnalta NC-sorvin karalle ja sieltä toiselle liukuhihnalle. Robotin liikkeille määritetty polku näkyy työasemassa keltaisina nuolina.

Robotin liikkeiden nopeutta ja sulavuutta voidaan säätää valitsemalla halutut pisteet ja hiiren oikealla näppäimellä avautuvasta valikosta Modify Instruction -toiminto. Modify Instruction -ikkunassa valitaan kohtiin Speed ja Zone halutut arvot ja sen jälkeen Apply-painikkeella otetaan uudet arvot käyttöön (kuva 29).



Kuva 29. Modify Instruction -ikkuna.

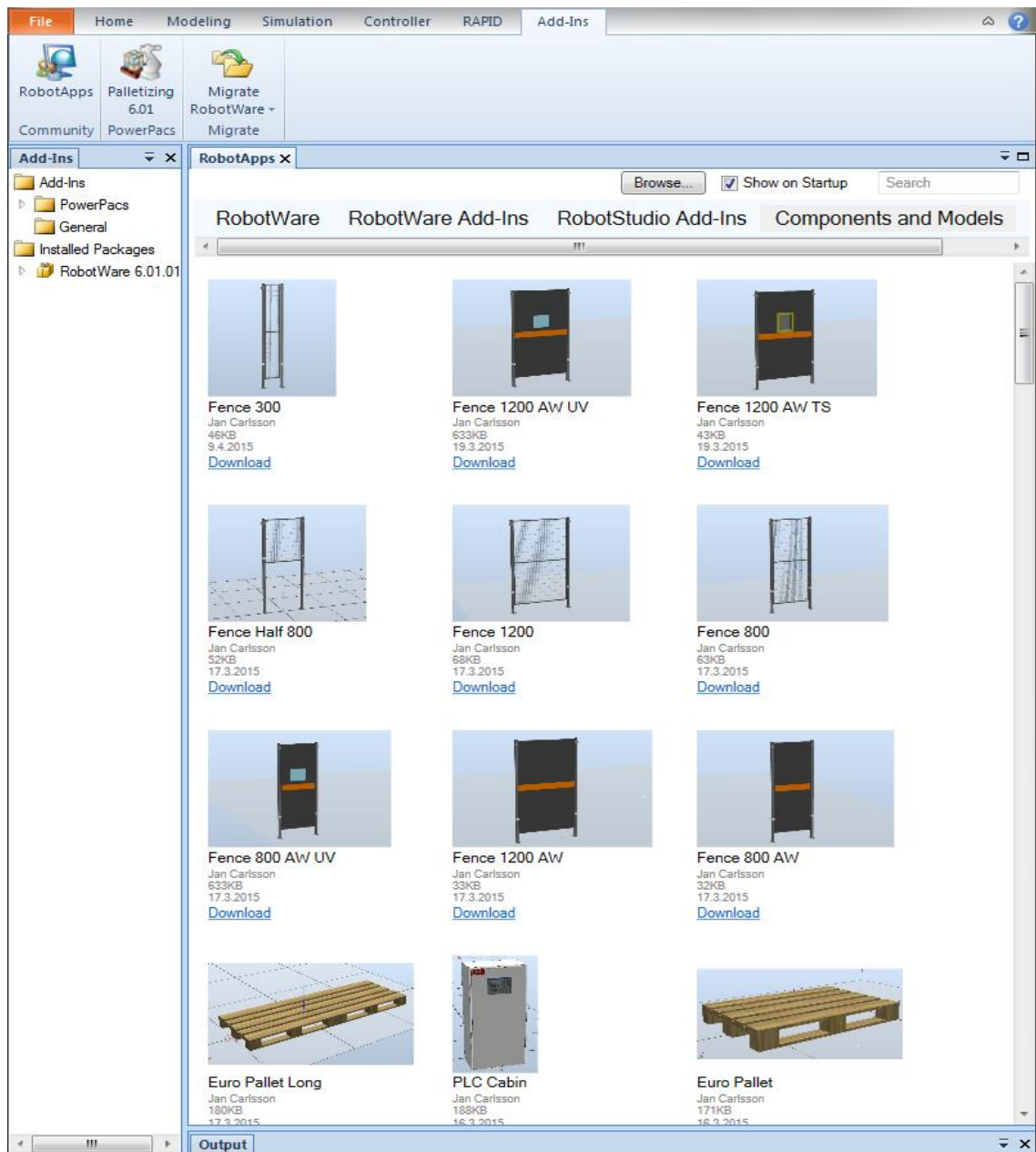
Simulaatiota varten asetuksia muutetaan Simulation-välilehden Configure-valikon Simulation Setup -toiminnolla. Simulation Setup -ikkunassa valitaan Virtual time mode -tilaksi Free run -tila ja Entry point -valikosta aiemmin robotin liikkeille luotu polku. Insinööriyössä tuo luotu polku nimettiin Path_10 -poluksi. Muutetut asetukset tulevat voimaan painamalla lopuksi Refresh-näppäintä (kuva 30).



Kuva 30. Simulation Setup -ikkuna.

5.6 Materiaalin tuonti soluun

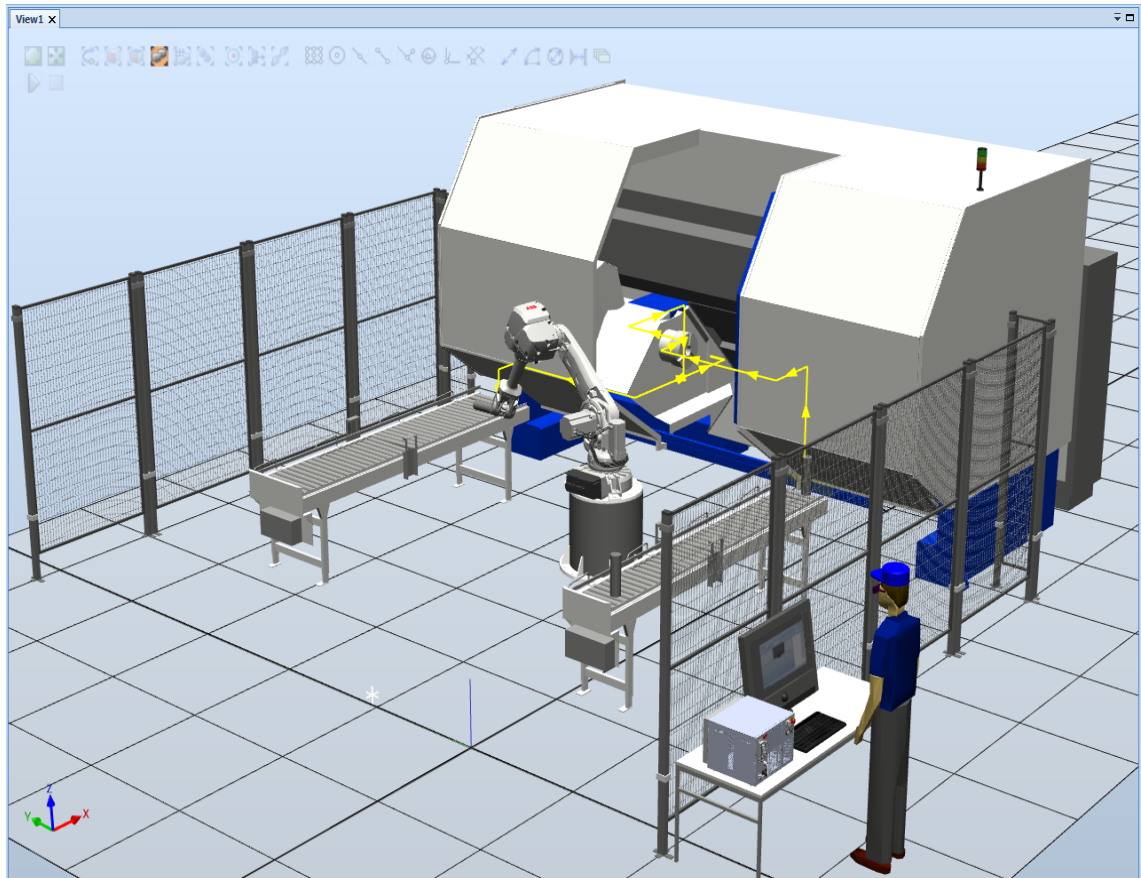
Tuotantosoluun voidaan tuoda komponentteja kuvan 31 mukaisesti Add-Ins-välilehden Components and Models -valikosta. Valikosta löytyy muun muassa turva-aitoja, kuormalavoja, robotin jalustoja ja ohjausyksiköitä sekä työstökoneiden malleja. Halutut mallit ja komponentit tallennetaan kansioon Download-painikkeella.



Kuva 31. Add-Ins-välilehti.

Tallennetut komponentit haetaan työasemaan Home-välilehden Import Library -kansion kautta. Komponentteja voidaan liikutella työasemassa halutuille paikoille Freehand-valikon toiminnoilla.

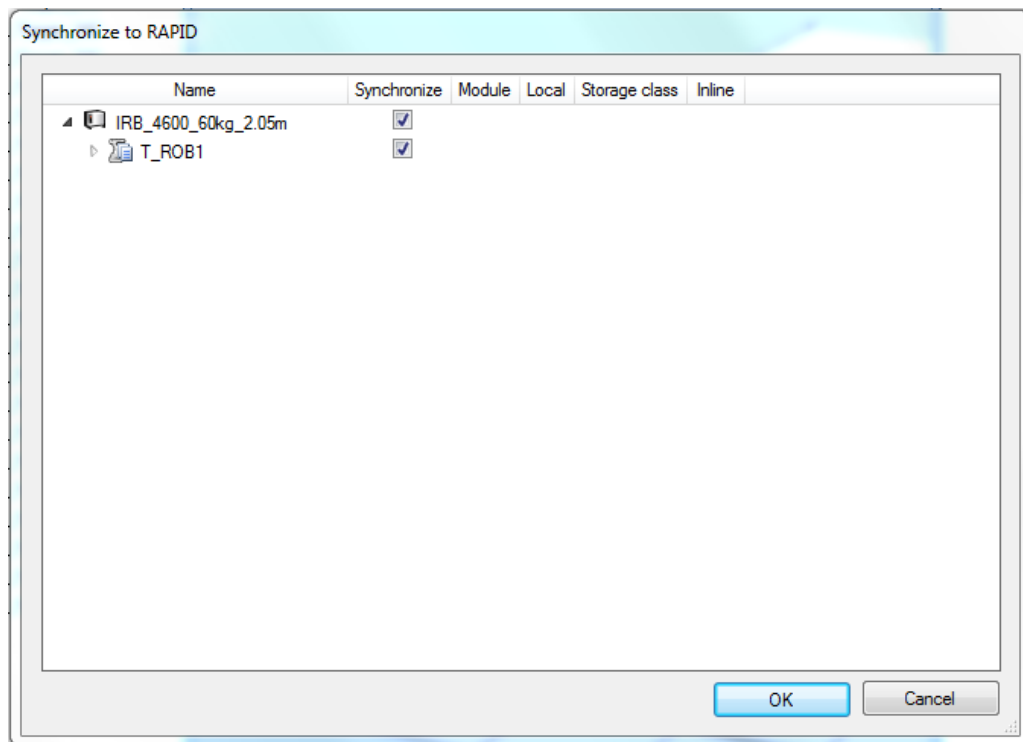
Insinöörityössä tuotantosoluun tuotiin suoja-aidat solun ympärille, robotille jalusta ja ohjauksyksikkö, pöytä tietokonemonitoria varten sekä tuotantotyöntekijä. Valmis tuotantosolu on kuvan 32 mukainen.



Kuva 32. Valmis tuotantosolu.

6 Tulokset

Insinööriyön tuloksena saatiin simulaatiosta RAPID-koodikielen koodi ja nauhoitettu simulointivideo. Jotta simuloinnista saadaan RAPID-koodi, täytyy robotin liikeradoille luotu polku synkronoida koodikieleen. Synkronointi tehdään valitsemalla polun kohdalta hiiren oikealla näppäimellä avautuvasta valikosta Synchronize to RAPID -toiminto ja valitsemalla avautuvasta ikkunasta synkronoitavat kohteet kuvan 33 mukaisesti.



Kuva 33. Synchronize to RAPID -ikkuna.

RAPID-koodi saadaan näkyviin RAPID-välilehdelle. Koodista näkyy robotin liikeradan polun pisteiden paikkakoordinaatit, siirtymänopeudet pisteestä pisteeseen, käytettävä työkalu sekä valittu työkappale (kuva 34). RAPID-koodi on kokonaisuudessaan liitteessä 2.

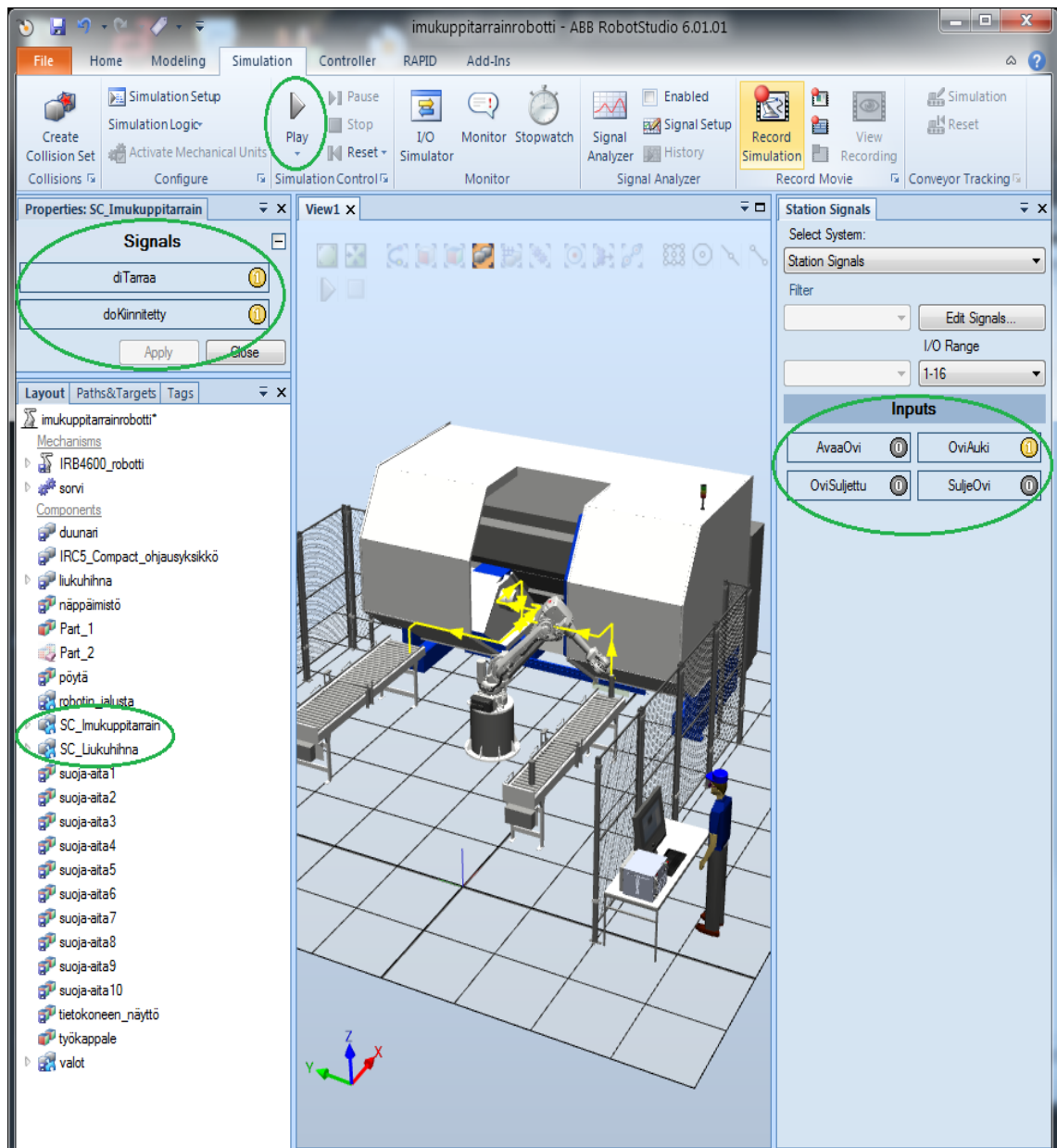
```

21 | CONST robtarget Target_320:=[[701.068579558,1718.582020583,511.990079545],[0.299106226,-0.300347883,0.643897055,0.636964047]
22 | CONST robtarget Target_330:=[[494.28548148,1718.582372969,511.989547734],[0.299105831,-0.300347147,0.643897279,0.636964354],
23 | CONST robtarget Target_340:=[[494.285657466,1718.583172108,143.399659737],[0.299106186,-0.300347624,0.643896695,0.636964552]
24 | PROC Path_10()
25 |     MoveL Target_250,v10,fine,Imukuppi\WObj:=wobj0;
26 |     MoveL Target_240,v10,fine,Imukuppi\WObj:=wobj0;
27 |     MoveL Target_230,v10,fine,Imukuppi\WObj:=wobj0;
28 |     MoveL Target_220,v10,fine,Imukuppi\WObj:=wobj0;

```

Kuva 34. Osa RAPID-koodia.

Simuloinnista voidaan nauhoittaa video kuvan 35 mukaisesti Simulation-välilehdellä. Simulaatio käynnistetään Play-näppäimestä ja nauhoitetaan Record Simulation -toiminnolla. Robotin imukuppiyksikön tarrausominaisuuksia ja NC-sorvin ovimekanismeja ohjataan signaalinäppäimillä. Simuloinnin päätyttyä nauhoitettu video voidaan tallentaa haluttuun kansioon VLC media file -tiedostona.



Kuva 35. Simulaatiovideon nauhoitus.

7 Yhteenveto

Insinööriyössä mallinnettiin ja simuloitiin tuotantosolumalli RobotStudio-ohjelmalla, joka hankittiin Metropolia Ammattikorkeakoululle konepajatekniikan laboratorioon tulevaa teollisuusrobotia varten. Työn aikana perehdyttiin RobotStudio-simulointiohjelmaan ja sen eri ominaisuuksiin.

Insinööriyölle asetetut tavoitteet saavutettiin hyvin. RobotStudio-ohjelmaan perehdyttiin niin, että sen perustoiminnot tulivat tutuiksi, ja ohjelman käyttö onnistuu perustasolla. Lisäksi ohjelmalla saatiin luotua teollisuusrobotin tuotantosolusta toimiva yksinkertainen simulointimalli. Ohjelman laajuudesta ja monipuolisuudesta johtuen syvempi osaaminen jäi kuitenkin vielä puuttumaan, ja toimintoja käyttämättä. Insinööriyössä perehtyminen rajattiinkin tuotantosolun simulointimallin mallintamiseen.

Työn tuloksina simulointiohjelmalla saatiin valmis simulaatiomalli tuotantosolusta, sekä solun teollisuusrobotin liikeradoista synkronoitu RAPID-koodi. Mallinnettaessa tuotantosolun simulointimallia, ongelmaksi muodostuivat eri komponenttien yhtäaikainen toiminta, sekä robotin liikealue ja siihen liittyvät rajoitukset.

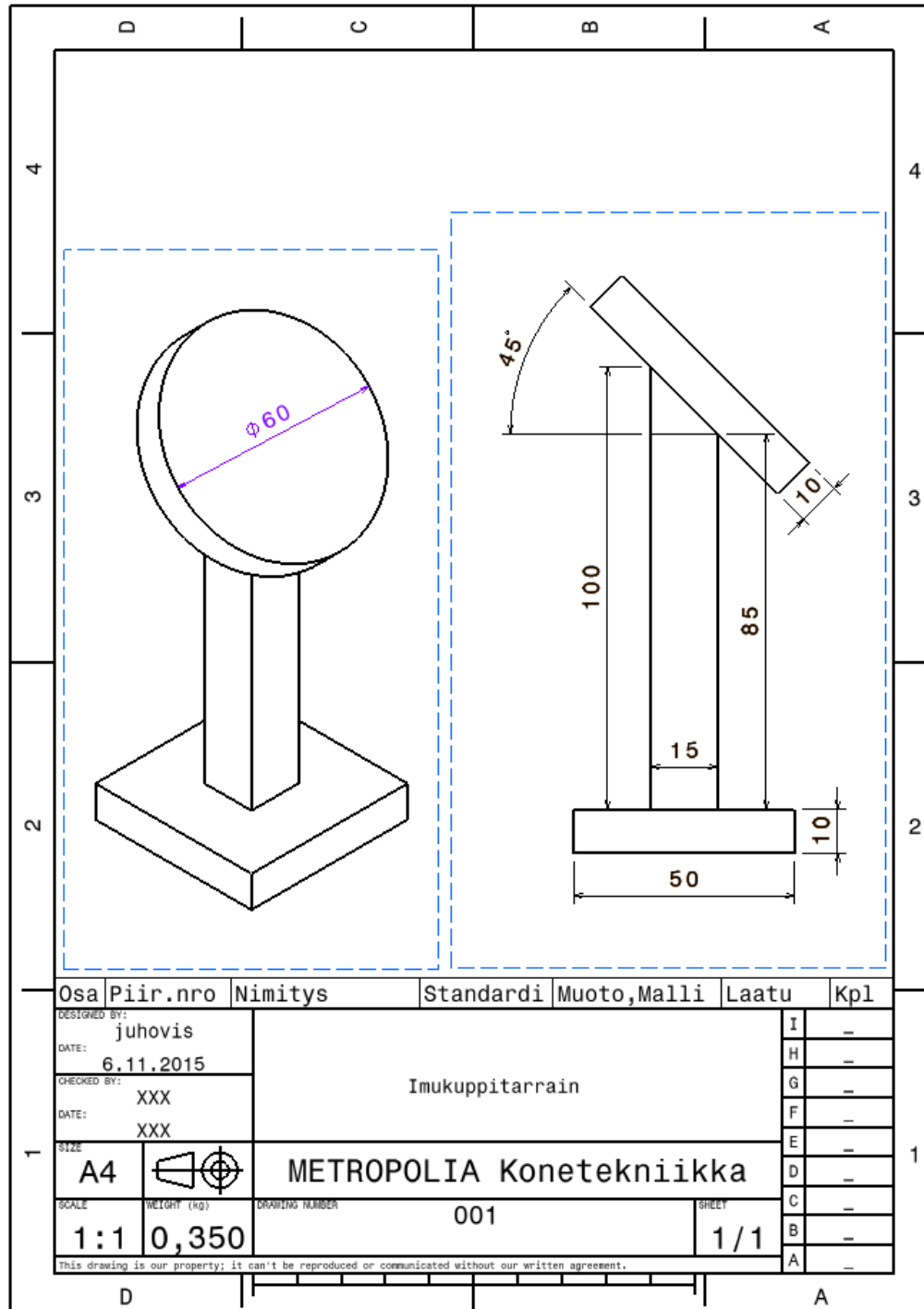
Insinööriyön aikana opittiin RobotStudio-ohjelman käytöstä ja sen ominaisuuksista. Opittua tietoa voidaan hyödyntää työelämässä ja -ympäristössä, jossa ohjelmointi ja teollisuusrobotit ovat osana tuotantoa. Metropolia Ammattikorkeakoulu voi hyödyntää insinööriyössä saatuja oppeja teollisuusrobotin käyttöönotossa, sekä oppitunneilla, joilla RobotStudio-ohjelmaan ja teollisuusrobotteihin on tarkoitus perehtyä.

Lähteet

- 1 Lapinleimu, I., Kauppinen, V. & Torvinen, S. 1997. Kone- ja metallituoteteollisuuden tuotantojärjestelmät. Porvoo: WSOY.
- 2 Tuotantojärjestelmä. 2016. Verkkodokumentti. <<https://webhotel2.tut.fi/isam2011/conferenceprogram.html>>. Luettu 11.1.2016.
- 3 Ahokas, J., Auer, J., Hyvönen, J., Naskali, E., Nenonen, M., Niemi, J., Peltonen, K., Rantalainen, J., Somppi, A., Tammisto, M. & Välikoski, A. 2013. Fastems FMS - Täydellä teholla. Tampere: Fastems Oy Ab.
- 4 Pikkarainen, E. & Mustonen, M. 2010. Numeerisesti ohjatut työstökoneet. Tampere: Opetushallitus.
- 5 Heinonkoski, R., Asp, R. & Hyppönen, H. 2008. Automaatio – helppoa elämää? Vammala: Opetushallitus.
- 6 Aaltonen, K. & Torvinen, S. 1997. Konepaja-automaatio. Porvoo: WSOY.
- 7 Tuotantoautomaatio. 2015. Verkkodokumentti. <www.oamk.fi/~eeroko/Opetus/Tuotantoautomaatio/Kappaletavaratuotannon%252520ja%252520metalliteollisuuden%252520tuotantoj%2584rjestelm%2584t%255B1%255D.ppt+%&cd=1&hl=fi&ct=clnk&gl=fi>. Luettu 11.11.2015.
- 8 Robotstudio-ohjelma. 2015. Verkkodokumentti. <<http://new.abb.com/products/robotics/robotstudio>>. Luettu 26.10.2015.

Imukuppitarraimen työpiirros

Liitteessä 1 on robotille mallinnetun työkalun työpiirros. Imukuppitarraintyökalu mallinnettiin CATIA-mallinnusohjelmalla.



Simulaation RAPID-koodi

Liitteessä 2 on luodun tuotantosolun robotille määritettyjen liikkeiden RAPID-koodi kokonaisuudessaan.

```

1  MODULE Module1
2  ▢ CONST robtarget Target_250:=[[740.319379989,-1618.283060406,399.43367544],[0.00201296,0.423873222,-0.905689631,0.007330169],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9]];
3  CONST robtarget Target_240:=[[723.299675195,-1618.283329195,399.43327624],[0.002013079,0.423873655,-0.905689428,0.007330167],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9]];
4  CONST robtarget Target_230:=[[723.299778688,-1618.283302314,1129.622608109],[0.002013086,0.423874014,-0.905689264,0.007329783],[-1,0,2,0],[9E9,9E9,9E9,9E9,9E9]];
5  CONST robtarget Target_220:=[[255.602793269,-1618.284243844,1129.622357065],[0.002013043,0.423873686,-0.905689421,0.007329303],[-1,0,2,0],[9E9,9E9,9E9,9E9,9E9]];
6  CONST robtarget Target_210:=[[255.602760032,-1142.782052967,1129.622545956],[0.002013004,0.423873963,-0.905689291,0.007329374],[-1,0,2,0],[9E9,9E9,9E9,9E9,9E9]];
7  CONST robtarget Target_200:=[[255.602400959,-1142.781808538,1129.622613621],[0.299106595,-0.300346491,0.643897276,0.636964307],[-1,0,2,0],[9E9,9E9,9E9,9E9,9E9]];
8  CONST robtarget Target_190:=[[255.6020883094,-396.98372701,1129.622019326],[0.299106751,-0.300345311,0.643897098,0.63696497],[-1,0,3,0],[9E9,9E9,9E9,9E9,9E9]];
9  CONST robtarget Target_180:=[[601.968599488,-396.983341573,1129.621837128],[0.299106651,-0.300345279,0.643897098,0.636965032],[-1,0,3,0],[9E9,9E9,9E9,9E9,9E9]];
10 CONST robtarget Target_170:=[[601.968374076,233.529212557,1129.621547138],[0.299106459,-0.300345165,0.643897117,0.636965156],[0,0,3,0],[9E9,9E9,9E9,9E9,9E9]];
11 CONST robtarget Target_160:=[[1442.538176796,233.529078989,1129.622226752],[0.299106727,-0.300344487,0.643897216,0.636965081],[0,0,3,0],[9E9,9E9,9E9,9E9,9E9]];
12 CONST robtarget Target_150:=[[1442.538558868,233.529177794,511.99179295],[0.299106437,-0.300344733,0.643897175,0.636965312],[0,1,2,0],[9E9,9E9,9E9,9E9,9E9]];
13 CONST robtarget Target_140:=[[2086.504082918,233.530266837,511.990859137],[0.299105528,-0.300344946,0.643897519,0.636965291],[0,0,3,0],[9E9,9E9,9E9,9E9,9E9]];
14 CONST robtarget Target_130:=[[2086.504345883,392.329978726,511.990896572],[0.299105253,-0.300345256,0.64389751,0.636965283],[0,0,3,0],[9E9,9E9,9E9,9E9,9E9]];
15 CONST robtarget Target_260:=[[2086.50483637,232.709706397,511.991299992],[0.2991054,-0.300345485,0.64389729,0.636965329],[0,0,3,0],[9E9,9E9,9E9,9E9,9E9]];
16 CONST robtarget Target_270:=[[701.567596137,232.709815239,511.991417553],[0.299105552,-0.300346782,0.643897264,0.636964672],[0,1,2,0],[9E9,9E9,9E9,9E9,9E9]];
17 CONST robtarget Target_280:=[[2103.451203431,232.709877777,511.990154081],[0.299105686,-0.30034633,0.643897117,0.63696497],[0,0,3,0],[9E9,9E9,9E9,9E9,9E9]];
18 CONST robtarget Target_290:=[[2103.451025501,408.557915962,511.989756379],[0.299105798,-0.30034631,0.643897108,0.636964936],[0,0,3,0],[9E9,9E9,9E9,9E9,9E9]];
19 CONST robtarget Target_300:=[[2103.451055913,233.304422226,511.989399872],[0.299105813,-0.300346094,0.643897346,0.63696479],[0,0,3,0],[9E9,9E9,9E9,9E9,9E9]];
20 CONST robtarget Target_310:=[[701.067996022,233.304391871,511.990278163],[0.299106089,-0.300347626,0.643896993,0.636964295],[0,1,2,0],[9E9,9E9,9E9,9E9,9E9]];
21 CONST robtarget Target_320:=[[701.068579558,1718.582020583,511.990079545],[0.299106226,-0.300347883,0.643897055,0.636964047],[0,2,1,0],[9E9,9E9,9E9,9E9,9E9]];
22 CONST robtarget Target_330:=[[494.285481148,1718.582372969,511.989547734],[0.299105831,-0.300347147,0.643897279,0.636964354],[0,2,1,0],[9E9,9E9,9E9,9E9,9E9]];
23 CONST robtarget Target_340:=[[494.285657466,1718.583172108,143.399659737],[0.299106186,-0.300347624,0.643896695,0.636964552],[0,2,2,0],[9E9,9E9,9E9,9E9,9E9]];
24 ▢ PROC Path_10()
25     MoveL Target_250,v10,fin,Imukuppi\Wobj:=wobj0;
26     MoveL Target_240,v10,fin,Imukuppi\Wobj:=wobj0;
27     MoveL Target_230,v10,fin,Imukuppi\Wobj:=wobj0;
28     MoveL Target_220,v10,fin,Imukuppi\Wobj:=wobj0;
29     MoveL Target_210,v10,fin,Imukuppi\Wobj:=wobj0;
30     MoveL Target_200,v10,fin,Imukuppi\Wobj:=wobj0;
31     MoveL Target_190,v10,fin,Imukuppi\Wobj:=wobj0;
32     MoveL Target_180,v10,fin,Imukuppi\Wobj:=wobj0;
33     MoveL Target_170,v10,fin,Imukuppi\Wobj:=wobj0;
34     MoveL Target_160,v10,fin,Imukuppi\Wobj:=wobj0;
35     MoveL Target_150,v10,fin,Imukuppi\Wobj:=wobj0;
36     MoveL Target_140,v10,fin,Imukuppi\Wobj:=wobj0;
37     MoveL Target_130,v10,fin,Imukuppi\Wobj:=wobj0;
38     MoveL Target_260,v10,fin,Imukuppi\Wobj:=wobj0;
39     MoveL Target_270,v10,fin,Imukuppi\Wobj:=wobj0;
40     MoveL Target_280,v10,fin,Imukuppi\Wobj:=wobj0;
41     MoveL Target_290,v10,fin,Imukuppi\Wobj:=wobj0;
42     MoveL Target_300,v10,fin,Imukuppi\Wobj:=wobj0;
43     MoveL Target_310,v10,fin,Imukuppi\Wobj:=wobj0;
44     MoveL Target_320,v10,fin,Imukuppi\Wobj:=wobj0;
45     MoveL Target_330,v10,fin,Imukuppi\Wobj:=wobj0;
46     MoveL Target_340,v10,fin,Imukuppi\Wobj:=wobj0;
47  ENDPROC
48 ENDMODULE

```