



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Kendy Nguyen

# Virtuaalitodellisuuden hyödyntäminen teollisuusrobottien opetuksessa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

Insinöörityö

4.11.2020

Tekijä Otsikko	Kendy Nguyen Virtuaalitodellisuuden hyödyntäminen teollisuusrobottien opetuksessa
Sivumäärä Aika	43 sivua 4.11.2020
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Mobile Solutions
Ohjaajat	Yliopettaja Kari Salo Vanhempi ohjelmoija Kimmo Arovirta
<p>Insinööritöiden tarkoituksena oli suunnitella ja kehittää teollisuusrobotin työkierron toteutusta käyttäen C#-ohjelmointikieltä ja korkean tason RAPID-ohjelmointikieltä sekä erilaisia menetelmiä datan tallentamisessa tietokantoihin. Lisäksi tutkittiin virtuaalitodellisuuden hyödyntämistä teollisuusrobotin ohjaukseen.</p> <p>Insinööritöihin kuului työkiertoskriptien parantaminen, tietokannan tietojen tallennuksen suorituskyvyn parantaminen ja koordinaatiston luominen työkaluille ja työkaluille simulaatioympäristössä. Simulaatiot suoritettiin RobotStudio-ohjelman simulaatioympäristössä. RobotStudioon on mahdollista tuoda valmiit 3D-mallit teollisuusroboteista ja työkaluista. Mitat ovat todella tarkkoja ja vastaavat todellisten robottien ja työkalujen mittoja. Tavoitteena oli saada virtuaalimaailmaan realistinen työympäristö, jossa operaattori pystyy opettamaan robotille työkiertoa etänä, nopeasti ja helposti.</p> <p>Lopputuloksena saatiin parannettua tietokannan suorituskykyä ja tehtyä skriptimuutoksia robottiohjelmistoon. Lisäksi saatiin robottisoluista ja työkaluista tarkka 3D-malli, jossa työkiertoa toteutettiin aluksi tietokoneella ja myöhemmin virtuaalitodellisuuden kautta. Virtuaaliympäristössä operaattori pystyy olemaan vuorovaikutuksessa robotin ja työkalujen kanssa. Virtuaalitodellisuuden kautta toteutettu työliikkeen opetus on mahdollista esittää etänä asiakkaalle tai huoltomekaniikoille. 3D-mallin työkaluja pystyttiin purkamaan osiksi virtuaaliympäristössä ja esittämään kappaleiden yksityiskohtia. Opetetut työliikkeet on mahdollista viedä lopuksi fyysiselle robotille.</p>	
Avainsanat	teollisuusrobotti, virtuaalitodellisuus, tietokanta, työntoteutus, C#

Author Title	Kendy Nguyen Utilization of virtual reality in the teaching of industrial robots
Number of Pages Date	43 pages 4 November 2020
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Professional Major	Mobile Solutions
Instructors	Kari Salo, Principal Lecturer Kimmo Arovirta, Senior Programmer
<p>The purpose of this thesis was to design and develop implementation of an industrial robot work cycle using C# programming language and the RAPID high-level programming language and different methods for storing data in databases. In addition, the utilization of virtual reality for the control of an industrial robot was studied.</p> <p>The thesis work included improving work cycle scripts, improving the performance of database data storage, and creating frames for tools and work objects in a simulation environment. The simulations were performed in RobotStudio-program simulation environment. In RobotStudio, it is possible to import ready-made 3D modeling of industrial robots and work objects. The dimensions are really accurate and correspond to the dimension of real robot and work objects. The goal was to get realistic work environment into virtual world where the operator can teach the robot work cycle remotely, quickly, and easily.</p> <p>The result was improved software performance on the database side and changes to robot software scripts. In addition, was obtained an accurate 3D model of robot cell and work objects, in which the work cycle was initially implemented on computer and later through virtual reality. In a virtual environment, the operator can interact with the robot and work objects. Work movements and path teaching in virtual reality can be presented remotely to the customer or service mechanics. 3D modeling of work objects was able to disassemble into parts in virtual environment and present the details of the part. The taught paths can finally import to a physical robot.</p>	
Keywords	industrial robot, virtual reality, database, work cycle, C#

## Sisällys

### Lyhenteet

1	Johdanto	1
2	Teollisuusrobotit	2
2.1	Teollisuusrobotin historia	2
2.2	Flexmillin käytössä olevat teollisuusrobotit	3
3	Robottien ohjaus ja työkierron toteutus	7
3.1	RobotStudio-ohjelmointiohjelmisto	9
3.2	PowerMill-valmistusohjelma	12
3.3	MillControl-käyttöliittymä	14
3.4	FlexMarker-käyttöliittymä	15
3.5	Datan tallentamisen suunnittelu	16
3.6	Väliaikaiseen tiedostoon tallennus	20
3.7	Tietokanta	20
4	Virtuaalitodellisuuden tekniikka	24
4.1	Virtuaalilasit	25
4.2	Tukiasemat	25
4.3	Ohjaimet	26
5	Virtuaalityökierto	28
5.1	Suunnittelu	28
5.2	Komponentit ja työkalut	28
5.3	Virtuaalinen työympäristö ja työnohjaus	31
5.4	Prototyyppi	37
6	Virtuaalitodellisuus opettamisessa	38
6.1	Fyysisen ja virtuaalisen opetustavan vertailu	38
6.2	Virtuaalitodellisuuden käyttö muilla aloilla	39
7	Yhteenveto	39



## Lyhenteet

ABB	Asea Brown Boveri.
AR	Augmented Reality.
CAD	Computer-aided Design.
CAM	Computer-aided manufacturing.
CNC	Computer numerical control.
Cobots	Collaborative Robots.
CSV	Comma-separated file.
MoveJ	Move Joint.
MoveL	Move Linear.
RAM	Random Access Memory.
Solu	Suljettu robotin toiminta-alue.
SQL	Structured Query Language.
TCP	Tool Center Point.
TMP	Temporary file.
VR	Virtual Reality.

## 1 Johdanto

Teollisuusrobottien käyttö lisääntyy nopeasti nykypäivänä teollisuusosalalla, ja niiden avulla pyritään vähentämään työvoimakustannuksia ja lisäämään tuottavuutta teollisuudessa [1]. Teollisuusrobotti mahdollistaa hyvin joustavan työkierron, koska robotti pystyy suorittamaan useita eri työtehtäviä suurella nopeudella ja toistotarkkuudella. Tyypillisesti teollisuusroboteilla on kuusi liikeakselia, mikä mahdollistaa laajan joustavuuden ja hyvän ulottuvuuden. Kuuden akselin teollisuusrobotit ovat ihanteellisia mm. kaarihitsaukseen, pistehitsaukseen, materiaalinkäsittelyyn, konekäsittelyyn, hiontaan ja maalaamiseen. Teollisuusrobotilla pystytään suorittamaan työtehtäviä, jotka ovat vaikeita ja vaarallisia ihmisille, ja siten parantamaan työturvallisuutta.

Tässä insinööriyössä tutkitaan teollisuusrobottien toimintoja ja niiden työkiertoja sekä ohjelmiston puolelta tietojenkäsittelyä. Työssä suunnitellaan ohjelmiston parannuksia, kuten toimintojen lisäystä, skriptien muokkausta ja tietokantaan tallentamisen suorituskyvyn parantamista. Lisäksi työssä tutkitaan virtuaalitodellisuuden hyödyntämistä teollisuusrobottien liikeratojen opettamisessa.

Insinööriyössä tutkitaan teollisuusrobotin työkierron toteutustapoja, ohjelmistorakenteita ja liikeratojen opettamista offline-ohjelmoinnissa ja virtuaalitodellisuuden avulla. Lisäksi tutkitaan tietojenkäsittelyn suorituskyvyn ja skriptien toimintojen parantamista. Lopuksi tutkitaan myös hyviä ja huonoja menetelmiä liikeratojen opettamisessa offline-ohjelmoinnissa ja virtuaaliympäristössä.

Insinööriyö tehdään Flexmill Oy:lle, joka on erikoistunut teollisuusosalalla hionta-automaatioon ja materiaalinkäsittelyyn. Yrityksessä keskitytään pääasiassa lentokoneturbiinien siipien hiontaan käyttäen teollisuusrobotteja. Flexmill Oy sijaitsee Nurmijärvellä ja työllistää 23 henkilöä. Sen toiminta alkoi Lappeenrannan teknillisestä yliopistosta (LUT) vuonna 1994, kun opiskelijaryhmä kehitti tavan ohjelmoida teollisia 6-akselisia robotteja samalla tavalla kuin 5-akselisia jyrsintäkoneita käyttämällä perinteistä CAM-ohjelmaa (Computer-aided manufacturing). Yrityksen nimi oli tuolloin Pii Robotics. [2.]

## 2 Teollisuusrobotit

Teollisuusrobotilla tarkoitetaan robottijärjestelmää, jota käytetään tuotteiden valmistuksessa. Teollisuusrobotit ovat automatisoituja, ohjelmoitavia ja kykeneviä liikkumaan vähintään kolmella akselilla. Teollisuusroboteilla minimoidaan työvoiman tarvetta työtehtävistä, jossa on paljon toistuvia liikkeitä, jotka ovat ihmisille kuluttavia. [1.] Nykyajan robotit ovat servomootoreilla ohjattuja. Tällä hetkellä maailman suurimpia teollisuusrobottivalmistajia ovat mm. ABB, Kawasaki Heavy Industries, The Yaskawa Electric Corporation, KUKA Robotics, Comau ja Stäubli [3].

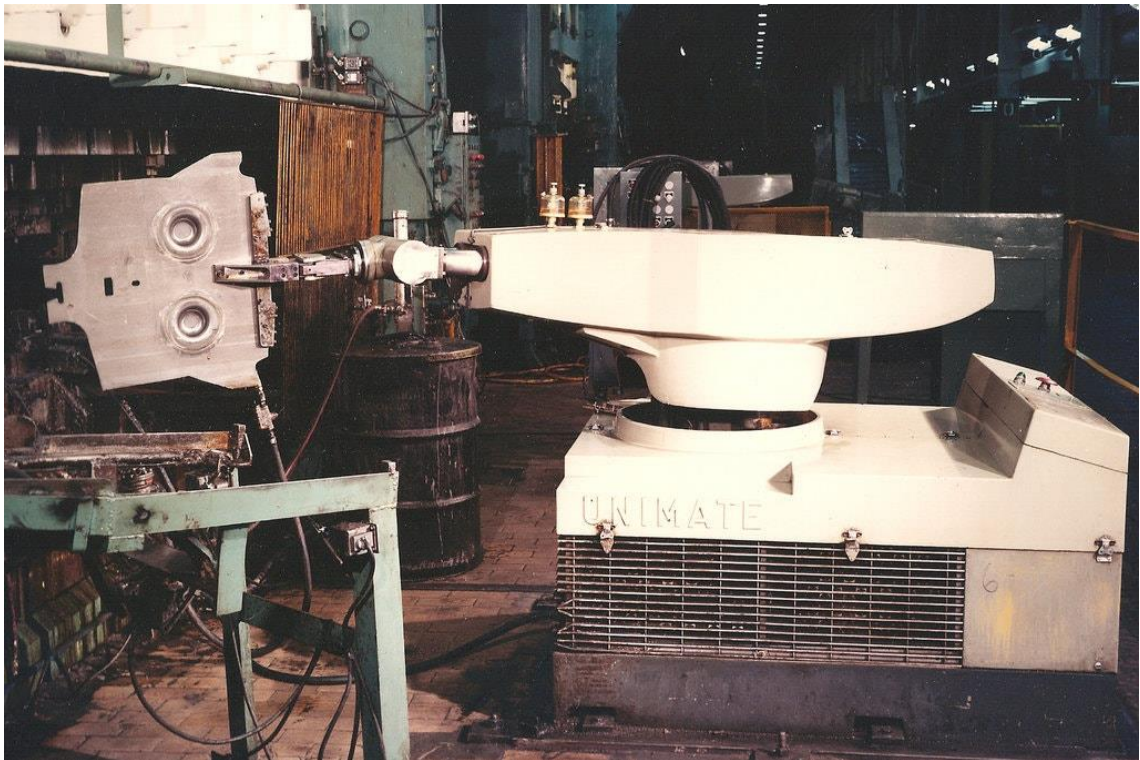
### 2.1 Teollisuusrobotin historia

Ensimmäisiä robotiikkapatentteja haki yhdysvaltalainen keksijä ja insinööri George Charles Devol Jr. vuonna 1954, ja ensimmäinen patentti myönnettiin vuonna 1961 [4; 5]. Vuonna 1956 George Devol ja yhdysvaltalainen fyysikko, insinööri ja liikemies Joseph F. Engelberger keskustelivat kirjailija Isaac Asimovin robottifilosofioista ja George Devolin patentoitavasta Programmed Article Transfer -laitteesta. Engelberger tunnisti laitteen heti robotiksi ja huomasi valtavan liiketoimintamahdollisuuden. [6.]

Devol ja Engelberger perustivat samana vuonna ensimmäisen robotteja tuottava yrityksen nimeltään Unimation, joka perustui Devolin alkuperäisiin patenteihin. Unimation-robotteja suunniteltiin siirtämään esineitä pisteestä toiseen alle kymmenen metrin etäisyydellä toisistaan. Robotit käyttivät hydraulisia toimilaitteita ja ohjelmoitiin yhteiskoordinaatteihin, eli robottien nivelten liikkeet tallennettiin opetusvaiheen aikana ja niitä toistettiin uudelleen toiminnassa. Robotit toimivat jopa 0,00254 millimetrin tarkkuudella. [4.] Vuonna 1959 kehitettiin ensimmäinen nykyistä teollisuusrobottia muistuttava prototyyppi nimeltään Unimate. Ensimmäinen Unimate-robotti asennettiin General Motorsin tuotantolinjalle New Jerseyssä Yhdysvalloissa vuonna 1961. Laitetta ohjelmoitiin pinnoittamaan ja hitsaamaan kuumia metalliosia autoihin. [7.] Kuvassa 1 esitellään Unimate-robotti.

Vuoteen 1966 mennessä Engelberger laajensi Unimationin jakelua kansainvälisesti antamalla lisenssejä suomalaiselle Nokialle ja japanilaiselle Kawasaki Heavy Industriesille (nykyään Kawasaki Robotics) ja aloitti samalla Euroopan ja Aasian robottimarkkinat. [6.]





Kuva 1. Unimate- mekaaninen varsi [7].

Teollisen robotiikan alalla tapahtui iso muutos, kun ASEA (nykyään ABB) kehitti maailman ensimmäisen mikroprosessoriohjatun sähköteollisen robotin IRB 6 vuonna 1974. IRB 6 -robotti käytti ensimmäistä Intelin 8-bittistä piirisarjaa, jossa oli 16 kt (kilotavua) RAM-muistia. Robotti oli 5-akselinen ja maksimi nostokapasiteetti oli 6 kg. [8; 9; 10].

## 2.2 Flexmillin käytössä olevat teollisuusrobotit

Tässä insinööriyössä käytetään ABB:n IRB-teollisuusrobotteja, jotka ovat käytössä Flexmill Oy:n toimitiloissa. Teollisuusrobotteja on kolme eri mallia, ja niiden käyttötarkoitukset vaihtelevat työkappaleiden mukaan. Teollisuusrobotteja käytetään pääasiassa työkappaleiden hiontaan ja materiaalikäsittelyyn, kuten laser- ja pistemerkintään. Lisäksi käytössä on virtuaalimallit IRB-teollisuusroboteista, joita käsitellään syvemmin 5. luvussa.

## S-solu

S-solu eli "Small Casing Robot" on pienikoteloinen S-sarjan teollisuusrobotti, joka käyttää ABB:n 6-akselista IRB 1600 -robottia. Solu tarkoittaa, että robotti on suljetussa tilassa ja kokonaan pois ihmiskontakteista. S-solua ympäröi Flexmillin runko, jonka vaatima pinta-ala on 5 m<sup>2</sup>, mikä tekee solusta hyvin kompaktin (kuva 2). IRB 1600 -teollisuusrobotti kykenee nostamaan työkappaletta tai työkalua, jonka kokonaispaino on 10 kg [11]. Flexmillin robottiin kiinnitettävien lisävarusteiden jälkeen nostokapasiteetti työkappaleille ja työkaluille on 3 kg [12]. Robotilla on erilaisia työkaluja, mm. hihna- ja karatyökalut. Aktiivisena robotti kykenee itse vaihtamaan työkaluja, eli operaattorin ei tarvitse manuaalisesti vaihdella työkaluja ja siten pystytään lisäämään työturvallisuutta ja saavuttamaan jopa 24 tunnin miehittämätön tuotantoajo.



Kuva 2. S-solun sisällä oleva IRB 1600 -teollisuusrobotti ja Hitecon karamoottori.

S-solun IRB 1600 -teollisuusrobotin työkappaleina toimivat yleensä lentokoneturbiinien siivet, hammaspyörät ja vaihdelaatikot, joita hiotaan ja viimeistellään. Robotisoidussa viimeistelyssä on tyypillisiä menetelmiä, joilla suoritetaan työajo. Ensimmäisessä tyypillisessä menetelmässä robotti pitää työkappaletta käsivarren leuoissa kiinni ja vie työkappaleen hiomatyökalun luokse, jotka ovat staattisesti solussa. Toisessa menetelmässä on päinvastainen toiminto (Tool-to-part), eli työkalu on robotin käsivarressa kiinni ja ajettava

työkappale on alustalla. Käytettävä menetelmä riippuu työkappaleen koosta, painosta ja tarvittavien työkalujen määrästä: esimerkiksi turbiinin siipi, joka on suhteellisen pieni, olisi parempi olla robotin käsivarressa kiinni, missä työkappaletta pystytään viimeistelemään joka suunnasta. Isoimmat työkappaleet, kuten vaihdelaatikko, olisi hyvä asettaa alustalle, jossa robotti ohjaa työkalua ja siten saada tarkempi lopputulos.

### Merkintäsolu

Merkintäsolu käyttää samanlaista IRB 1600 -teollisuusrobottia kuin S-solu, mutta työtehtävä on erilainen. Merkintäsolussa työtehtävä on työkappaleiden hionnan sijaan materiaalien käsittely. Merkintäsolussa on Cognexin kamerat sekä Techniforin laser- ja pistemerkintälaitteet asennettuna (kuva 3). Kirjoitushetkellä työkappaleina toimivat lentokoneen pienet turbiinien siivet.



Kuva 3. Merkintäsolu lisävarusteineen ja IRB 1600 -teollisuusrobotti.

Robotilla on erilaisia työkiertoja, joita se käy läpi. Robotti poimii hyllyistä leuat, joilla pysyy ottamaan työkappaleista kiinni ja viemään RFID-lukulaitteelle. RFID-lukulaitteen jälkeen robotti vie turbiinin siiven joko pistemerkintälaitteelle tai lasermerkintälaitteelle, missä siiven runkoon tehdään merkintöjä. Pistemerkintään verrattuna lasermerkintä on modernimpi tapa asettaa merkintöjä runkoihin [13]. Käytettävä merkintälaite riippuu siipityypistä. Työkierrossa seurataan, mikä siipi on poimittu, merkitty ja käsitelty.



Robottiin on etukäteen ohjelmoituna liikeradat, ja suoritettavat toiminnot tapahtuvat statuskoodien avulla. Työkierrossa soluohjain antaa tietyn statuksen, joka halutaan suorittaa, ja robotti vastaa statuksella. Esimerkiksi soluohjain haluaa poimia siiven paletilta ja antaa statuskoodin 10, jolloin robotti vastaa statuksella 40, että siipi on poimittu.

### L-solu

L-solu on Flexmillin toimitiloissa suurin käytössä oleva teollisuusrobotti. L-solu eli "Large Casing Robot" käyttää ABB:n IRB 6640 185/2.80 -teollisuusrobottia, jonka kuormakapasiteetti on jopa 185 kg. L-solua käytetään pääasiassa raskaiden kappaleiden hiontaan ja käsittelyyn. Solun sisällä on servolla toimiva pyörityspöytä, johon mahtuu enintään 1 800 mm pitkä ja 500 kg painava työkappale. [14.] IRB 6640 -teollisuusrobotti on 6-akselinen ja sen maksimi ulottuvuus on 2,80 m (kuva 4) [15].



Kuva 4. L-solu ja sisällä oleva IRB 6640 -teollisuusrobotti ja pyörityspöytä.

Työkappaleina L-solussa ovat toimineet isommat lentokoneen turbiinien siivet ja hammaspyörät. Kirjoitushetkellä ajettavana työkappaleena on asekaappi, jonka hitsausaumat hiotaan tasapintaiseksi.

### 3 Robottien ohjaus ja työkierron toteutus

Jokaisella teollisuusrobottien valmistajilla on käytössään oma robottiohjain. Tässä insinööriyössä tutkitaan Flexmillin toimitiloissa olevia teollisuusrobotteja ja niiden ohjausta. Kirjoitushetkellä käytössä ovat ABB:n IRB-sarjan teollisuusrobotit. IRB-sarjan teollisuusrobotteja ohjataan IRC5-ohjaimella. ABB-teollisuusrobotit ohjelmoidaan RAPID-ohjelmointikielellä. RAPID on korkean tason ohjelmointikieli, joka esiteltiin vuonna 1994 (kuva 5) [8].

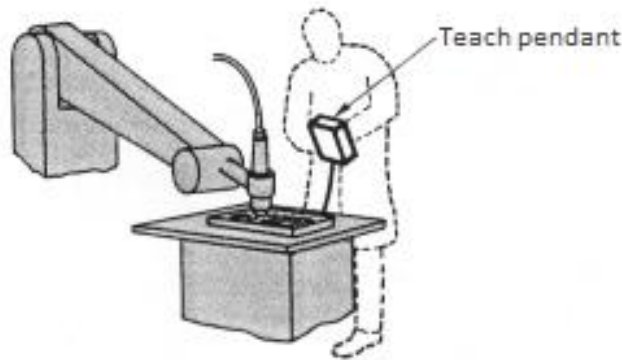


Kuva 5. Teollinen robottiohjain ja FlexPendant-ohjain.

Online-ohjelmointi tarkoittaa, että robottia ohjelmoidaan fyysisesti teollisella robottiohjaimella. Eri online-ohjelmointityyppejä ovat mm. opettaminen (engl. Teach Pendant Programming) ja johdattaminen (engl. Lead Through Programming). [16.]

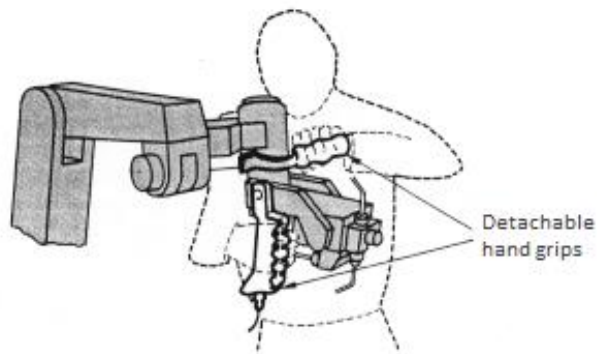
Opettamisessa operaattori ohjaa robotin haluttuun paikkaan robottiohjaimella ja tallennetaan senhetkiset liikkeet muistiin. Tätä vaihetta toistetaan, kunnes saadaan valmis lii-

kerata työkappaleelle. Turvallisuussyistä opetusvaiheessa robottia ohjataan hidastettuna ja myöhemmin ajossa ajetaan tehdasnopeudella, eli maksiminopeudella. Huono puoli opetusvaiheessa on se, että tuotanto pitää pysäyttää hetkellisesti, kun halutaan opettaa tai muokata liikeratoja. Operaattori joutuu myös työskentelemään opetusvaiheessa robotin lähietäisyydessä yleensä turvakehikon sisällä, mikä lisää työtaturman riskiä. Tällainen opetustapa on melko työlästä, puuduttavaa ja hidasta, ja muokkausmahdollisuudet ovat rajalliset (kuva 6). [16; 17.]



Kuva 6. Opetusmenetelmästä, jossa operaattori ohjaa robottia käyttäen Teach pendant -ohjainta.

Johdattaminen tarkoittaa, että operaattori manuaalisesti ja fyysisesti ohjaa robottia haluttuun pisteeseen kädestä pitäen. Johdatusvaiheessa robottien liikeakselit vapautuvat, minkä jälkeen operaattori pystyy liikuttamaan robottia vapaasti. Kunkin akselin liikkeet tallennetaan automaattisesti muistiin. Johdattamisen etuina on helppo käytettävyys yksinkertaiseen liikeratojen opettamiseen. Tätä menetelmää käytetään, kun halutaan suorittaa yksinkertaisia työtehtäviä, kuten maalin ruiskuttaminen tai liiman levittäminen epä-säännölliselle pinnalle. Huonoja puolia ovat vaikeasti muokattavat liikeradat. [16; 17.] Johdattaminen-ohjelmointityyppiä käytetään yleensä yhteistyörobottien eli cobottien (engl. Collaborative robots) kanssa. Cobotit on suunniteltu toimimaan ihmisten vuorovai- kutuksessa, ja niiden nopeus ja paino eroavat teollisista roboteista, mikä tekee cobo- teista turvallisempia (kuva 7). [18.]



Kuva 7. Johdattamisessa operaattori ohjaa robottia käsivoimin.

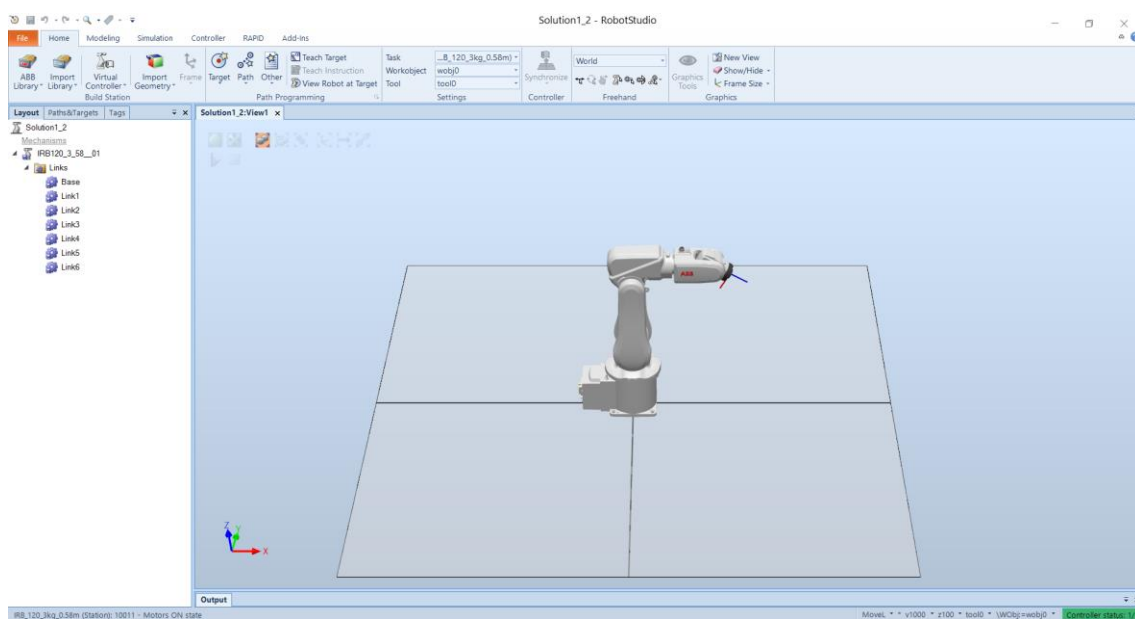
Offline-ohjelmointi tarkoittaa etäohjelmointia tietokoneella joko valmistajan oman tai kolmannen osapuolen etäohjelmointiohjelman kautta simulaatioympäristössä. Ohjelmiin voidaan tuoda 3D-malleja robotista CAD-ohjelman (engl. Computer-aided Design) kautta ja käyttää samanlaisia toimintoja, joita online-ohjelmoinnissa on käytössä. Offline-ohjelmoinnissa hyviä puolia on simulaatioympäristössä työskentely, missä voidaan kokeilla erilaisia menetelmiä ilman, että liikutetaan fyysistä teollisuusrobottia. Tuotantoa ei tarvitse myöskään pysäyttää, kun muokataan tai opetetaan uutta liikerataa. Törmäykset ovat myös havaittavissa etukäteen simulaatioympäristössä. Opetetut työkaluradat voidaan ladata robottiohjaimelle ja ajaa oikealla teollisuusrobotilla. [16; 17.]

### 3.1 RobotStudio-ohjelmointiohjelmisto

RobotStudio on ABB:n kehittämä simulointi- ja offline-ohjelmointiohjelmisto. Sitä käytetään tietokoneen kautta, missä ohjelmisto sallii operaattorin suorittaa tehtäviä, kuten harjoittelua ja ohjelmointia, häiritsemättä tuotantolinjaa. RobotStudioon on rakennettu ABB VirtualController, joka on hyvin tarkka kopio todellisesta robottiohjaimesta tuotannon puolella olevissa teollisuusroboteissa. Tämä mahdollistaa hyvin realistisen simuloinnin etäohjelmoinnissa ja liikeratojen tarkan ajon oikeassa teollisuusrobotissa. [19.]

Työn tavoitteena on luoda offline-ohjelmoinnissa simulaatioympäristö virtuaalitodellisuuden käyttöä varten. Simulointiympäristöön on myös mahdollista tuoda 3D-malleja työkaluista, joita sitten viimeistellään solussa. RobotStudiolla on valmiina ABB:n omia

IRB-sarjan teollisuusrobotteja, mutta työssä tuodaan oma 3D-malli Flexmillin toimitiloissa olevista IRB-teollisuusroboteista, jossa on valmiina solu ja tarvittavat työkalut. Robottimallin ja työkalujen tuonnissa pitää erikseen määrittää työkalut komponenteiksi, koska ne ovat tuotu erillisinä tiedostoina ohjelmaan, eikä RobotStudio näitä automaattisesti tunnista. Simulointiympäristössä suoritetaan testiskriptejä läpi, muokataan ja luodaan uusia liikeratoja. Kun uusi projekti avataan RobotStudiassa, saadaan tyhjä simulointiympäristö näkyville. Kuvassa 8 on esimerkkinä ABB:n kirjastosta tuotu IRB 120 -teollisuusrobotti simulointiympäristössä.



Kuva 8. RobotStudion simulointiympäristö ja IRB-teollisuusrobotti.

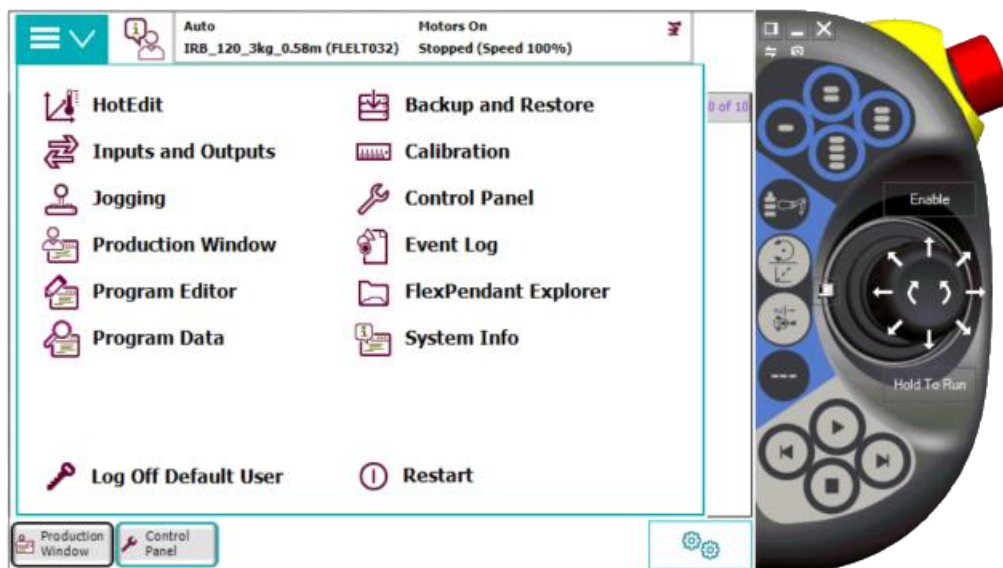
Ylävalikon Home-välilehdestä pystyy tuomaan erilaisia IRB-teollisuusrobotteja ja työkaluja. Liikeratojen opettaminen tehdään myös samasta välilehdestä luomalla tyhjä polku, ja robottia liikutetaan teach pendant -menetelmällä eli asetetaan piste kerrallaan. Robottille voi asettaa erilaisia työtehtäviä Task-osiossa, ja työkalupale ja työkalut ovat myös vaihdettavissa. Modeling-välilehdessä pystytään luomaan koordinaatistoja, työkaluja ja ryhmittelemään komponentteja. Simulation-välilehdessä voidaan synkronoida opetetut liikeradat virtuaaliohjaimeen (engl. VirtualController) ja ajaa simulaatio läpi. Controller-välilehdessä pystytään hallitsemaan virtuaaliohjainta, ja RAPID-välilehdessä pystytään tutkimaan ja muokkaamaan ohjelmistokoodia.



Sivuvalikossa olevassa asettelu (engl. Layout) -ikkunassa ovat robotin tiedot, kuten nivelten arvot ja erilaiset työkalut ja komponentit. Työkaluja pystytään liittämään robottiin raahaa ja pudota -menetelmällä (engl. drag and drop) 6-akselin alle komponenttipuussa. Kuvassa 8 on esitettyä 6-akseli nimeltään Link6.

Päänäkymässä ovat simulaatioympäristö, teollisuusrobotti ja mahdolliset työkappaleet. Operaattori pystyy valitsemaan simuloinnin kuvakulman ja tarvittaessa tutkimaan työkalun mittoja.

RobotStudiassa saa virtuaalisen FlexPendant-ohjaimen, jolla voi ladata ohjelmistotietoja (kuva 9). Ohjelmistotietoja ovat mm. erilaiset työpolut, RAPID-ohjelmointikieli ja parametrit. Virtuaaliohjaimessa on täsmälleen sama toiminnallisuus kuin todellisessa ohjaimessa, joten sen avulla saadaan hyvin tarkka toteutus. Skriptipuolen toteutukset suoritetaan virtuaalisella FlexPendant-ohjaimella, jossa syötetään statusparametriarvoja ohjelmiston dataan.



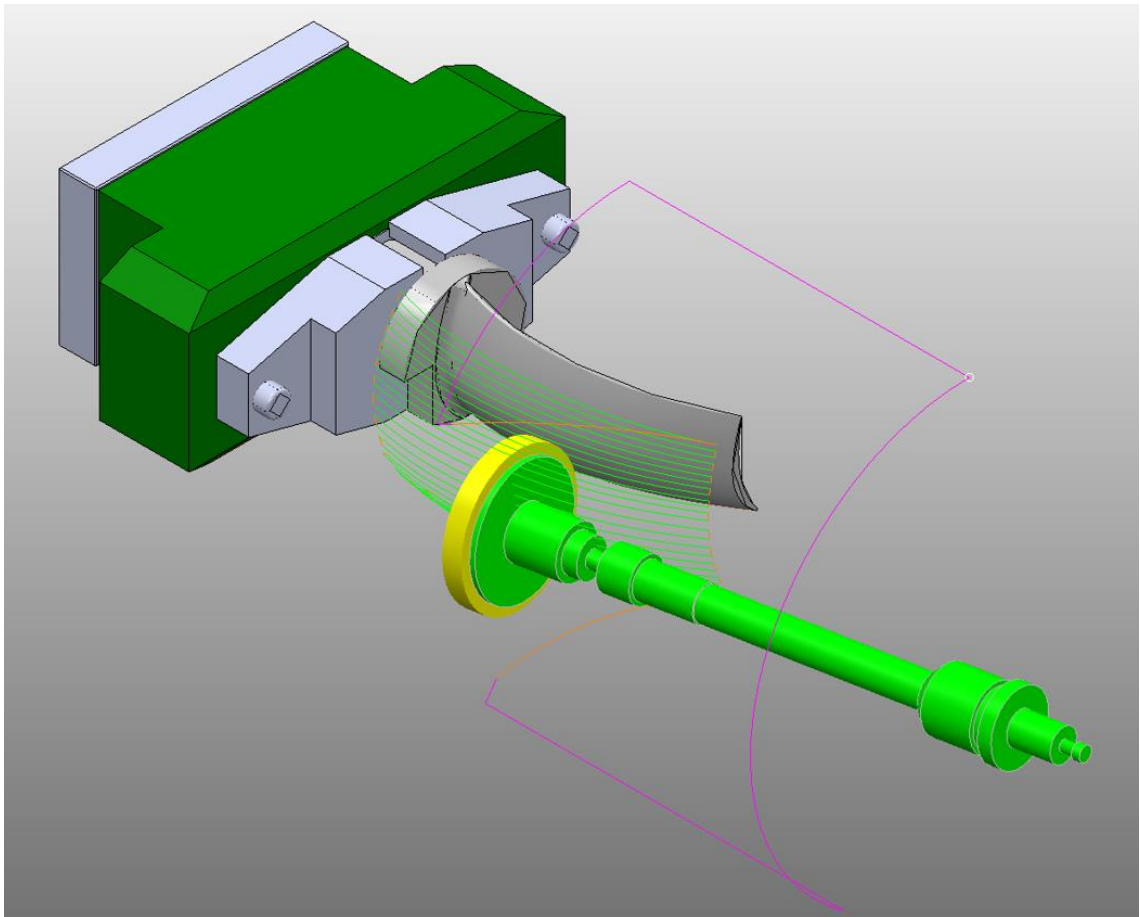
Kuva 9. ABB FlexPendant-virtuaaliohjain.

### 3.2 PowerMill-valmistusohjelma

PowerMill on Autodeskin kehittämä 3D-CAM (Computer-aided manufacturing) -tietokoneavusteinen valmistusohjelma. PowerMill on suunniteltu käytettäväksi CNC-jyrsinkoneille. PowerMill-ohjelmistoa käytetään myös useilla teollisuuden aloilla määrittämään liikeratoja teollisuusroboteille. Sen avulla saadaan vähennettyä työaikaa ja valmistuskustannuksia. PowerMilliä käytetään pääasiassa 3- ja 5-akseliseen koneistukseen. PowerMillin käyttöjärjestelmänä toimii Microsoft Windows. [20.]

PowerMill-ohjelmaa käytetään offline-ohjelmoinnissa. Ohjelma pystyy tunnistamaan pintoja ja reunoja, minkä avulla työkaluradat saadaan. Luoduille radoille PowerMill-ohjelma pystyy laskemaan optimaalisen ajotavan robotille, jotta robotti välttää törmäykset, singulariteetit ja akselien äärirajat. PowerMill-ohjelmalla saadaan hyvin tasainen pinta työkappaleille, koska ohjelma tarjoaa hyvin erilaisia tasoja työkappaleen pinnalle. Pinnanviimeistelyn aikaansaaminen robotilla vaatii kuitenkin melko paljon prosessituntemusta. Ohjelma työstää CAD-kappaleen pintaa joko suoraa viivaa seuraten tai siksak-liikkeillä. Tämä antaa todella tarkan ja sulavan pinnan työkappaleille. Robottien työstöliike muistuttaa hyvin paljon 3D-tulostimien liikkeitä.

Autodesk on kehittänyt PowerMilliin laajennuksen, joka mahdollistaa CAM-ratojen laskennan 6-akseliselle robotille ja robottikoodin generoinnin eri robottivalmistajalle. Esimerkiksi pinnanviimeistelyradalle PowerMillissä pystyy määrittämään radan tiheyttä (step over), viimeistelyaluetta, työkalun orientaation pinnan normaalin suhteen, väliliikkeet, törmäysmäärytykset, lähestymis- ja poistumisliikkeet ja liikkeiden väliset linkit ja aloitus- ja lopetuspisteet (kuva 10). [21.]



Kuva 10. Työstöstrategiat PowerMillin avulla.

Laajennuksien avulla pystytään määrittämään robotin liikkeitä ja sitä, miten se seuraa liikerataa. Ajetut liikeradat voidaan muuntaa RAPID-ohjelmointikieleksi, joilla saadaan suoritettava liikerata ABB:n teollisuusrobotteihin (kuva 11).

```

1 MODULE
2 PROC Main()
3   !Offline program created: 09:41:31 / 20.08.2020
4   !-----ROBOT CONFIGURATIONS-----!
5   AccSet 80, 80; ConfJ \On;
6   SingArea \Wrist; Confl \Off;
7   jHome := [[90,-30,-(-30),0,-90,90],[0,0,0,0,0,0]];
8   !-----TOOL CONFIGURATIONS-----!
9   tCurrent := t3;
10  tCurrent.tframe := PoseMult(tCurrent.tframe,[[0,0,0],[1,0,0,0]]);
11  tCurrent.tframe.rot := OrientZYX(EulerZYX(\Z,tCurrent.tframe.rot)+0, EulerZYX(\Y,tCurrent.tframe.rot)+0, EulerZYX(\X,tCurrent.tframe.rot)+0);
12  !-----WORKOBJECT CONFIGURATIONS-----!
13  wCurrent := wRobotFlange;
14  wCurrent.uframe := PoseMult(wCurrent.uframe,[[0,0,14],[1,0,0,0]]);
15  wCurrent.uframe.rot := OrientZYX(EulerZYX(\Z,wCurrent.uframe.rot)+0, EulerZYX(\Y,wCurrent.uframe.rot)+0, EulerZYX(\X,wCurrent.uframe.rot)+0);
16  !-----SPINDLE/BELT CONFIGURATIONS-----!
17  nSpinValueCurrent := 7500; bSpindleRotBWD := FALSE;
18  nCompliancePressure := 2; bComplianceNegDirection := FALSE;
19  !-----SPEED CONFIGURATIONS-----!
20  vHome.v_tcp := 400; vLead.v_tcp := 50;
21  vApproach.v_tcp := 200; vFeed.v_tcp := 5;
22  vPlunge.v_tcp := 100; zCurrent := z1;
23  vHome.v_ori := 400; vLead.v_ori := 50;
24  vApproach.v_ori := 200; vFeed.v_ori := 5;
25  vPlunge.v_ori := 100;
26
27  MoveAbsJ jHome \NoEOffs,vHome,zCurrent,tCurrent\WObj:=wCurrent;
28  SetCompliancePressure nCompliancePressure, bComplianceNegDirection;
29  StartNakanishi \bBwdDirection:=bSpindleRotBWD;
30
31  Proc_Vane7_LSG_2_1;
32
33  StopNakanishi;
34  bSpindleRotBWD := TRUE;
35  StartNakanishi \bBwdDirection:=bSpindleRotBWD;
36
37  Proc_Vane7_LSG_2_1;
38
39  StopNakanishi;
40  MoveAbsJ jHome \NoEOffs,vHome,zCurrent,tCurrent\WObj:=wCurrent;
41  ENDPROC
42
43  !PROC Proc_Vane7_LSG_2_1()
44  !Toolpath starts
45  !Plunge in move starts
46  MoveJ [[-226.77,-20.51,334.63],[0.500757,0.500371,-0.500924,-0.497942],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vApproach,zCurrent,tCurrent\WObj:=wCurrent;
47  MoveL [[-226.60,-20.72,275.29],[0.500755,0.50037,-0.500928,-0.497941],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vPlunge,zCurrent,tCurrent\WObj:=wCurrent;
48  !Lead in move starts
49  MoveJ [[-233.36,-20.78,264.21],[0.500758,0.500366,-0.500928,-0.497941],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vLead,zCurrent,tCurrent\WObj:=wCurrent;
50  !Feed move starts
51  MoveL [[-223.23,-20.75,264.68],[0.500756,0.500367,-0.500928,-0.497944],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
52  MoveL [[-213.26,-20.73,263.66],[0.500755,0.500369,-0.500927,-0.497942],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
53  MoveL [[-189.30,-20.69,261.99],[0.500755,0.50037,-0.500925,-0.497943],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
54  MoveL [[-173.55,-20.65,262.63],[0.500755,0.50037,-0.500928,-0.497941],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
55  MoveL [[-170.62,-20.64,262.94],[0.500753,0.500369,-0.500931,-0.497942],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
56  MoveL [[-164.27,-20.59,272.62],[0.500754,0.500369,-0.500929,-0.497942],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
57  MoveL [[-157.54,-20.58,272.64],[0.500755,0.50037,-0.500927,-0.497943],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
58  MoveL [[-157.51,-20.61,263.41],[0.500753,0.50037,-0.500929,-0.497942],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
59  WaitTime 2;
60  MoveL [[-157.54,-20.58,273.88],[0.500754,0.500371,-0.500928,-0.497942],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
61  MoveL [[-151.40,-20.56,273.71],[0.500753,0.500367,-0.500929,-0.497945],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
62  MoveL [[-146.25,-20.55,272.86],[0.500752,0.50037,-0.500928,-0.497944],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
63  MoveL [[-144.48,-20.57,267.88],[0.500754,0.500369,-0.500927,-0.497943],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
64  MoveL [[-141.05,-20.58,264.38],[0.500752,0.50037,-0.500927,-0.497945],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
65  MoveL [[-134.70,-20.56,264.40],[0.500753,0.50037,-0.500927,-0.497944],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vFeed,zCurrent,tCurrent\WObj:=wCurrent;
66  !Feed move ends
67  MoveL [[-134.73,-20.52,276.00],[0.500752,0.500369,-0.500928,-0.497945],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vLead,zCurrent,tCurrent\WObj:=wCurrent;
68  !Lead out move ends
69  MoveL [[-134.82,-20.41,305.84],[0.500754,0.500371,-0.500927,-0.497942],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vPlunge,zCurrent,tCurrent\WObj:=wCurrent;
70  MoveL [[-135.11,-20.06,405.32],[0.50075,0.500371,-0.500927,-0.497946],[1,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vApproach,zCurrent,tCurrent\WObj:=wCurrent;
71  !Plunge out move ends
72  !Toolpath ends
73  ENDPROC
74  ENDMODULE

```

Kuva 11. Esimerkkikoodi ajetusta liikeradasta RAPID-ohjelmointikielellä.

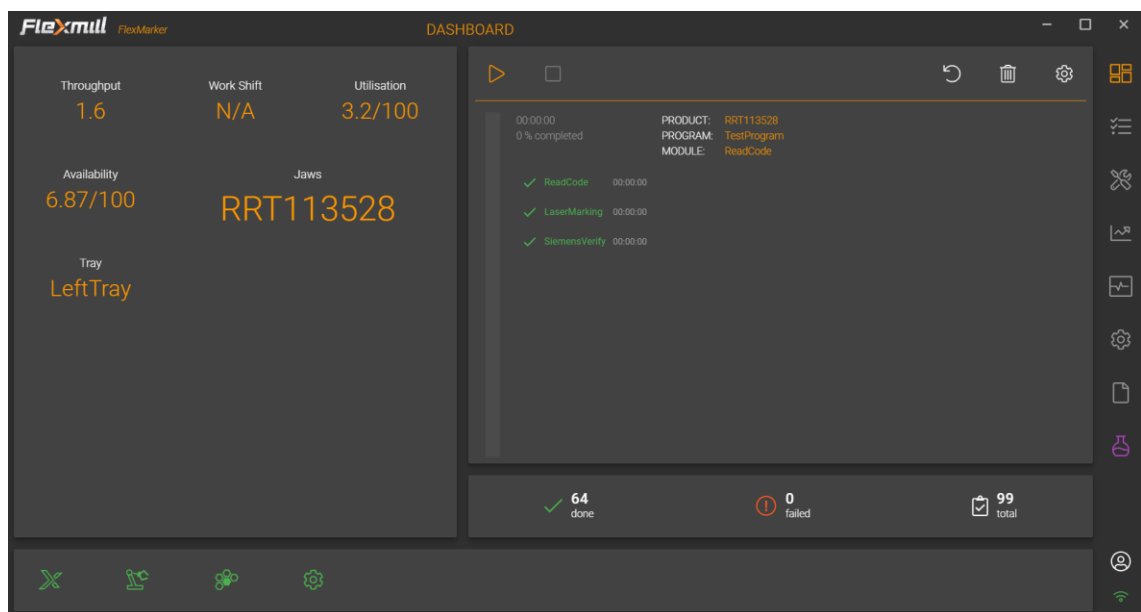
### 3.3 MillControl-käyttöliittymä

MillControl on Flexmill Oy:n kehittämä graafinen käyttöliittymä ja soluohjainohjelmisto robottien prosessien seurantaan ja työjonojen hallintaan. Operaattori pystyy syöttämään parametriarvoja ohjelmaan ja luomaan uusia työkaluja ja skriptejä. MillControl-käyttöliit-

tymässä näkyy, mikä työkalu robotilla on aktiivisena, ja työkalun kierrosnopeus työstäessä. Ohjelmassa erilaisia ominaisuuksia ovat mm. virheiden käsittely, prosessidatan kerääminen, tiedonkeruu koodinlukijoilta kuten RFID ja työkalutietojen hallinta. MillControllerin ohjelmointirajapintana on VBScript. [13.]

### 3.4 FlexMarker-käyttöliittymä

FlexMarker on vastaavanlainen graafinen käyttöliittymä ja soluohjain-ohjelmisto kuin MillControl, mutta parannetuilla ominaisuuksilla. FlexMarkeria käytetään kirjoitushetkellä merkintäsolussa. Ohjelmointirajapintana on VBScriptin lisäksi C#-ohjelmointikieli (kuva 12). [13.]



Kuva 12. FlexMarker käyttöliittymä.

FlexMarkerissa on sisäänrakennettu skriptien muokkausmahdollisuus. Lisälaitteiden liittäminen on paljon helpompaa verrattuna MillControlliin, koska FlexMarkerin suunnittelussa on alusta alkaen otettu huomioon lisälaitteiden liittäminen. Merkintäsolussa FlexMarkeriin on kytkettynä Cognexin kamerat, Techniforin laser- ja pistemerkintälaitteet, ABB IRC5 -robotiohjain, Siemens Simatic ET 200SP PLC ja Simativ MV440 -kamera.

### 3.5 Datan tallentamisen suunnittelu

Datan tallennuksessa tallennetaan robottien prosessidataa sekä sykli- ja kontekstitiedot. Suunnitteluvaiheessa on otettu huomioon robottien työkalut ja niiden ominaisuudet, kuten osien tunnuksset, sarja- tai eränumerot, operaation tunnus, operaattorin tunnus, moduulin nimi, aktiivisen työn tunnus, pyörimisnopeus ja aktiivisen työkalun tunnus. Tallennuksessa on myös tärkeä tallentaa työn aloitus- ja lopetusaika, jotta ylläpito pystyy seuraamaan työnkulkua. Datat tallennetaan CSV-tiedostomuotoon (Comma-separated values), joka on tekstitiedostossa esiintyvä taulukkomuotoista dataa [22]. Datan tallennus on toteutettu back-end-puolella C#-ohjelmointikielellä ja Microsoft Visual Studio -ohjelmankehitysympäristöllä. Datan testaamiselle luotiin Visual Studio WinFormin (engl. Windows Forms App) avulla käyttöliittymä, johon dataa pystyttiin syöttämään manuaalisesti (kuva 13).

The screenshot shows a Windows Forms application window. On the left, there is a vertical list of input fields. To their right are labels: Part ID, Serial number, Batch number, Operation, MachineID, Operator ID, Module Name, Run ID, strCycleStartTim, strCycleENDTim, Spindle Number, and ActiveToolID. In the center, there are three buttons: 'RANDOM SERIAL-OR BATCH NUMBER' (yellow), 'ADD KPVDATA' (green), and 'PROCESS READY' (green). On the right, there are two main sections: 'CYCLE DATA' and 'CONTEXT DATA', both in green boxes. Each section has an 'ADD' button (yellow) followed by input fields and labels. The 'CYCLE DATA' section includes labels for Cycle ID, Trav, and Trav Place. The 'CONTEXT DATA' section includes labels for RESOURCE ID, USER ID, PLANT, and ID TYPE. At the bottom left, there is a red 'CLEAR DATA' button and an empty input field.

Kuva 13. Ohjelmiston käyttöliittymä on toteutettu Visual Studio WinFormia käyttäen.

Projektihakemistossa on C#-ohjelmointikielen ja WinForm-käyttöliittymän lisäksi INI-tiedosto, johon pystyttiin asettamaan tiedostojen polkuja ja otsikko- ja oletusarvot. INI-tiedosto on konfiguraatiotiedosto, joka on tekstipohjaista sisältöä [23]. Tiedostoon on valmiiksi määritettyä otsikkoarvot sisääntulodatalle ja arvot erotetaan toisistaan puolipisteellä. C#-ohjelmassa käydään läpi, mikä sisääntulodata asetetaan mihinkin otsikkoarvoihin (kuva 15).

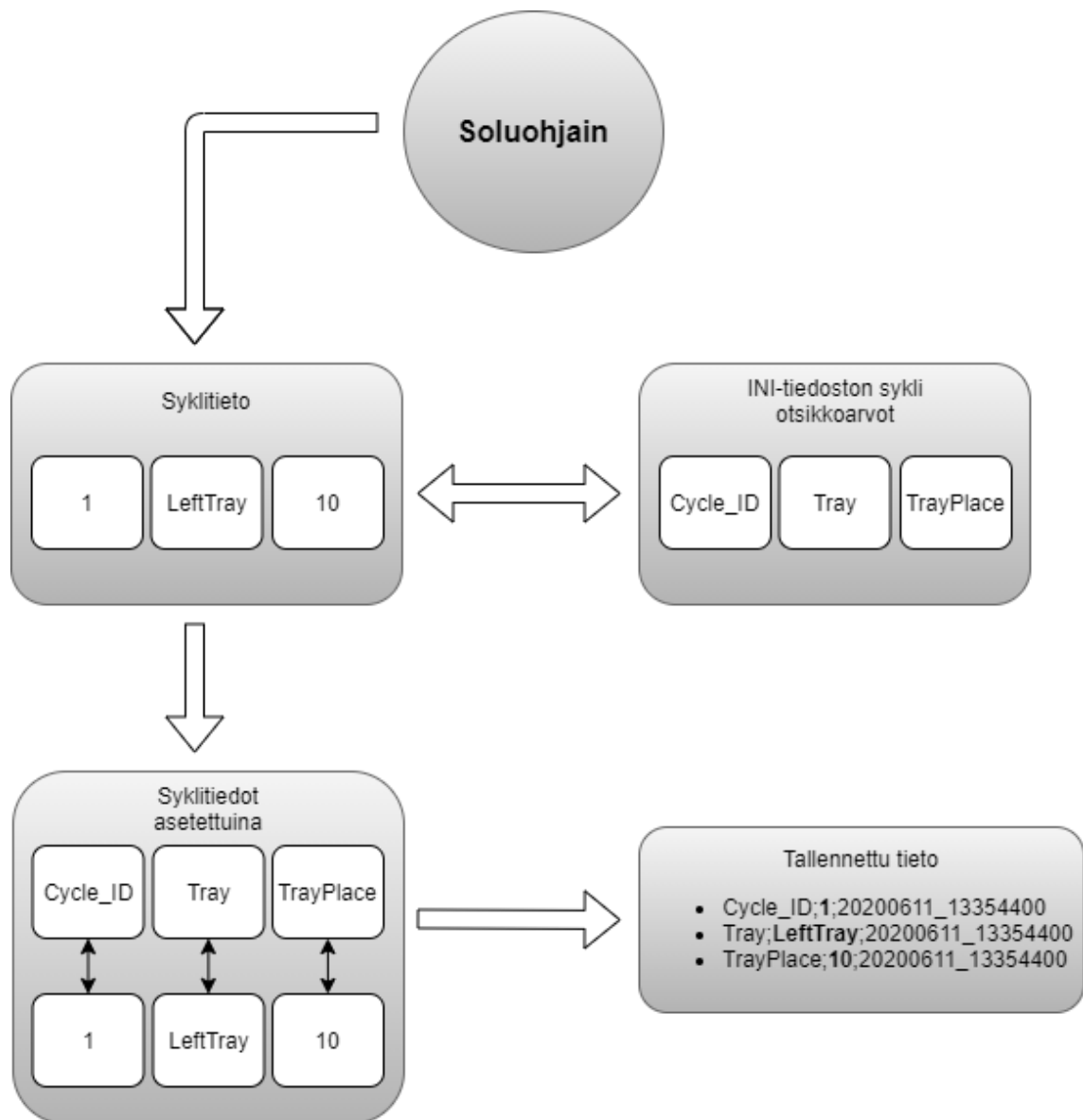


Prosessidatassa tallennetaan operaattorin tiedot, työkappaleen ja työkalujen tiedot sekä päivämäärät. Soluohjaimesta tuleva data tallennetaan CSV-tiedostoon, kunnes työkierto on suoritettu tai soluohjain kutsuu ohjelman lopetusta. Dataa verrataan INI-tiedostossa oleviin otsikkoarvoihin, ja tiedot täytetään niiden mukaisesti. Jos soluohjaimesta tulee tyhjää tietoa, jätetään sen kohta taulukosta tyhjäksi ja siirrytään seuraavaan arvoon. Tallennetussa datassa verrataan lisäksi aina sarja- tai eränumeroita keskenään. Jos työkappaleessa vaihtuvat sarja- tai eränumerot työn aikana, ohjelma tulostaa CSV-tiedoston ja aloittaa uuden työn tallentamisen uudella sarja- tai eränumerolla. Esimerkiksi aktiiviset työkalut ja osien tunnukset voivat vaihdella työn aikana, mutta sarjanumero pysyy aina samana (kuva 14).

<PROCESS DATA>										
Part_ID	Serial_No	Operation	MachineID	OperatorID	ModuleName	Run_ID	strCycleStartTime	strCycleEndTime	SpindleNumber	ActiveToolID
PART_1	SERIALNO1	PART_1	MACHINE1	OPERATOR1	MODULE1	RUNID1	20200611_09384500	20200611_09384500	SPINDLE1	ACTIVETOOL1
PART_1	SERIALNO1	PART_1	MACHINE1	OPERATOR1	MODULE1	RUNID1	20200611_09384500	20200611_09384500	SPINDLE1	ACTIVETOOL1
PART_1	SERIALNO1	PART_1	MACHINE1	OPERATOR1	MODULE1	RUNID1	20200611_09384500	20200611_09384500	SPINDLE1	ACTIVETOOL1
PART_2	SERIALNO1	PART_2	MACHINE1	OPERATOR1	MODULE2	RUNID2	20200611_09384500	20200611_09384500	SPINDLE2	ACTIVETOOL2
PART_2	SERIALNO1	PART_2	MACHINE1	OPERATOR1	MODULE2	RUNID2	20200611_09384500	20200611_09384500	SPINDLE2	ACTIVETOOL2
PART_2	SERIALNO1	PART_2	MACHINE1	OPERATOR1	MODULE2	RUNID2	20200611_09384500	20200611_09384500	SPINDLE2	ACTIVETOOL2
PART_2	SERIALNO1	PART_2	MACHINE1	OPERATOR1	MODULE2	RUNID2	20200611_09384500	20200611_09384500	SPINDLE2	ACTIVETOOL2
</PROCESS DATA>										

Kuva 14. CSV-tiedoston prosessidata esitettynä taulukkomuodossa, jossa näkyy operaattorin, työkappaleen ja työkalujen tiedot.

Syklitiedoissa tallennetaan seuraavat tiedot: syklitunnus, paletti, paletin paikka, päivämäärä ja aika. Paletti ja paletin paikka kertovat robottisolun hyllypaikat. Esimerkiksi sisääntulodata "1, LeftTray, 10" tarkoittaa ensimmäinen sykli, vasemmanpuoleinen paletti ja kymmenes paikka paletista. Soluohjaimesta tulevat tiedot muotoillaan oikeaan muotoon lopulliseen CSV-tiedostoon. Datan loppuun lisätään myös aikaleima, joka on muotoa vuosi-kuukausi-päivä-tunti-minuutit-sekunnin kymmenesosa (kuva 15 ja esimerkkikoodi 1). Syklitiedoissa data tallennetaan ainoastaan kerran työkierron aikana.



Kuva 15. Soluohjaimesta tuleva sykldata ja sen tallennus CSV-tiedostoon.

Kontekstitiedossa halutaan tallentaa laitteen tunnus, käyttäjänimi, tehdasalue ja ID-tyyppi. Laitteen tunnuksena toimivat merkintäsolumen lisälaitteet, eli joko "marking" tai "laser", ja käyttäjänimeen tulee laitteen operaattori. Tehdasalueena toimii laitteen sijainti ja ID-tyyppinä on joko sarja- tai eränumero (engl. serial number or batch number). Kuten sykkitiedoissa, kontekstitietojen data tallennetaan ainoastaan kerran työkierron aikana. Kaava on samanlainen kuin kuvassa 15 esitelty sisääntulodata, mutta INI-tiedoston otsikkoarvot ovat erilaiset.



	A	B	C	D	E	F	G	H	I	J	K	L
1	<CONTEXT>											
2	RESOURCE_ID	Marking										
3	USER_ID	Matti										
4	PLANT	Nurmijarvi										
5	ID_TYPE	Serial										
6	RunID	MarkingMattiNurmijarviSerial										
7	</CONTEXT>											
8	<CYCLE DATA>											
9	Cycle_ID	1	20200611_14244700									
10	Tray	LeftTray	20200611_14244700									
11	TrayPlace	10	20200611_14244700									
12	RunID	1LeftTray10										
13	</CYCLE DATA>											
14	<PROCESS DATA>											
15	Part_ID	Serial_No	Operation	MachineID	OperatorID	ModuleNa	Run_ID	strCycleStartTime	strCycleENDTime	SpindleNu	ActiveToolID	
16	part1	SerialNO1	part1	machine1	operator	module1	runId1	20200611_14234900	20200611_14234900	spindle1	tool1	
17	part1	SerialNO1	part1	machine1	operator	module1	runId1	20200611_14234900	20200611_14234900	spindle1	tool1	
18	part1	SerialNO1	part1	machine1	operator	module1	runId1	20200611_14234900	20200611_14234900	spindle1	tool1	
19	part1	SerialNO1	part1	machine1	operator	module1	runId1	20200611_14234900	20200611_14234900	spindle1	tool1	
20	part1	SerialNO1	part1	machine1	operator	module1	runId1	20200611_14234900	20200611_14234900	spindle1	tool1	
21	part1	SerialNO1	part1	machine1	operator	module1	runId1	20200611_14234900	20200611_14234900	spindle1	tool1	
22	</PROCESS DATA>											
23												

Kuva 16. CSV-tiedostossa olevat tiedot.

Suoritettun työkierron jälkeen soluohjain kutsuu ohjelman lopetusta ja tulostaa CSV-tiedoston. CSV-tiedostossa on prosessidata, sykli- ja kontekstitiedot ja data on erotettu toisistaan <OTSIKKO></OTSIKKO>-tyylillä (kuva 16). Tulostettavan CSV-tiedoston nimi on määritettynä valmiiksi INI-tiedostossa. INI-tiedostossa on asetettu NameData-nimi ja sille on määritetty arvoksi "Part\_ID;Operation;TimeStamp;Serial\_No" (esimerkkikoodi 1). NameDatan arvoissa määritetyt arvot ovat samat kuin prosessidatan otsikkoarvoissa. Ohjelmiston puolella korvataan NameDatan arvojen puolipisteet alaviivalla eli tiedoston-nimeksi muodostuu part1\_part1\_20200611\_14234900\_Serial\_NO1.csv.

```
[Settings]
Line=Part_ID;Serial_No;Operation;MachineID;OperatorID;ModuleName;Run_ID;strCycleStartTime;strCycleENDTime;SpindleNumber;ActiveToolID
Source=ProgramName;CompUniqNum;ProgramName;MachineID;OperatorID;ModuleName;Part_ID|ModuleName|MachineID|DATETIME;strCycleStartTime;strCycleENDTime;SpindleNumber;ActiveToolID
CycleDataHeaders=Cycle_ID;Tray;TrayPlace;RunID
CycleDataSources=ID;Tray;Slot;ID|Tray|Slot
ContextDataHeaders=RESOURCE_ID;USER_ID;PLANT;ID_TYPE;RunID
ContextDataSources=MachineID;UserName;PlantName;IdType;MachineID|UserName|PlantName|IdType
TimeStamp=yyyyMMdd_HHmss
NameData=Part_ID;Operation;TimeStamp;Serial_No
```

Esimerkkikoodi 1. INI-tiedoston otsikko- ja oletusarvot.

### 3.6 Väliaikaiseen tiedostoon tallennus

Suunnitteluvaiheessa on otettu huomioon asiat, jotka aiheuttavat teollisuusrobotille ja soluohjaimelle ongelmia työkierrossa, kuten sähkökatkos tai laitteen jumiutuminen. Datat menetykset näissä tapauksissa pystytään estämään luomalla väliaikaisia tiedostoja, joihin saadaan tallennettua soluohjaimesta tulevaa dataa ennen varsinaisen CSV-tiedoston tulostamista. Tämän menetelmän avulla työtä pystyttiin myös jatkamaan keskeytyneestä kohdasta. Väliaikaisen tiedoston formaattina toimi .TMP-tiedostotyyppi (Temporary files).

Väliaikaiset tiedostot toteutettiin ohjelmiston puolella. Väliaikaiset tiedostot luodaan Temp-hakemistoon, joka sijaitsee oletuksena Windows-hakemistossa, ellei tiedostopolkua erikseen määritetä. Väliaikaisen tiedoston luonnin jälkeen tiedostopolku tallennetaan INI-tiedostoon ja ohjelmiston puolella luetaan soluohjaimesta tuleva data ja vertaillaan dataa keskenään. Jos esimerkiksi keskeytetyn työn jälkeen soluohjaimesta tulee erilainen sarja- tai eränumero kuin väliaikaisessa tiedostossa oleva, niin väliaikaisessa tiedostossa olevat arvot tulostetaan CSV-tiedostoon ja aloitetaan uuden työn tallentaminen. Suoritettua työtä jälkeen ohjelma poistaa väliaikaisen tiedoston automaattisesti.

### 3.7 Tietokanta

#### SQL Server Compact

SQL CE (SQL Server Compact) on kompakti relaatiotietokanta, jonka on kehittänyt Microsoft [24]. Ohjelmiston puolella SQL CE -tietokantaan tallennetaan soluohjaimesta tulevat ID-arvot työjonoista, hyllyjen paikoista, tilakoodeista ja parametreista. SQL CE -tietokannassa esiintyvät tiedot sisältävät osittain samoja tietoja kuin tulostetussa CSV-tiedostossa. Erona ovat kuitenkin aktiiviset työjonojen ID-arvot, jotka eivät näy CSV-tiedostossa. Aktiivisilla työjonoilla tarkoitetaan työkierron statusta. Erilaisia statuksia ovat mm. uusi työ, keskeneräinen työ, käynnissä oleva työ ja suoritettu työ. Käyttöliittymän puolella kuvakkeet päivittyvät työjonojen statusten mukaisesti.

## SQLite

Ohjelmiston puolelle pohdittiin SQLite-tietokannan lisäämistä, koska sen avulla pystytään parantamaan suorituskykyä. SQLite-tietokannassa on ominaisuuksia, jotka nähtiin hyödyllisiksi työkiertojen datan tallentamisessa, kuten tallennusnopeus ja tiedostokoot. SQLite on ohjelmistokirjasto, joka on hyvin kevyt, palvelimeton ja itsenäinen, eli erillisiä laajennuksia ei tarvita [25]. Tietokanta luodaan paikallisesti tietokoneen kiintolevyille, johon säilytetään dataa. SQLite mahdollistaa transaktion käytön (engl. Transaction), eli useat SQL-lausekkeita ryhmitellään yhdeksi työyksiköksi, joka on sitoutunut tietokantaan yhtenä atomiyksikkönä. Lausekkeiden aiemmin tehdyt muutokset voidaan palauttaa, jos jokin tapahtuman lausekkeista epäonnistuu. Transaktion avulla suorituskykyä voidaan parantaa SQLite-tietokannassa, kun tehdään useampia muutoksia kerralla. [26.]

## Nopeusvertailu

Tietokantojen nopeuden vertailun testaamiseksi luotiin uusi käyttöliittymä Visual Studiosissa WinFormia käyttäen. Käyttöliittymässä luodaan tekstikentät, joihin asetetaan tietokannan ja taulukon nimi. Käyttöliittymässä on painikkeita, joilla voi lähettää parametrisarvoja ohjelmistoon ja siten luoda tietokanta hakemistoon (kuva 17). Ohjelmiston puolella on valmiiksi määritettynä oletusarvot ja foreach-silmukka, jossa dataa lisätään automaattisesti. Testausvaiheessa oletusarvoksi asetettiin "12345".

Form1

**Database Speed tester**

**SQL tester**

MyDBSql Create SQL

SampleSQLTable Create Table

66 ms RunNonQuery

RunQuery

**SQLite tester**

Create SQLite MyDB

Create Table SampleLiteTable

RunSQLiteNonQuery 18 ms

RunSQLiteQuery

Kuva 17. Tietokantojen nopeuden testaamiseksi luotiin uusi käyttöliittymä Visual Studiossa Win-Formia käyttäen.

Tietojen nopeuden vertailua toteutettiin syöttämällä aluksi 100 riviä dataa ja jatkettiin 1 000 000 riviin asti. Testauksessa mitattiin SQL CE:n, SQLiten ilman transaktiota ja SQLiten transaktion käytön kanssa tallennusnopeutta millisekunteina.

## 1 000 000 line of data

```
Total Execution Time for SQL: 94365 ms
Total Execution Time for SQLite Bulk insert: 3190 ms
Total Execution Time for SQLite: 3693826 ms
```

## 100 000 line of data

```
Total Execution Time for SQL: 5211 ms
Total Execution Time for SQLite: 256624 ms
Total Execution Time for SQLite Bulk insert: 185 ms
```

## 10 000 line of data

```
Total Execution Time for SQL: 532 ms
Total Execution Time for SQLite: 26445 ms
Total Execution Time for SQLite Bulk insert: 38 ms
```

## 1 000 line of data

```
Total Execution Time for SQL: 60 ms
Total Execution Time for SQLite: 2422 ms
Total Execution Time for SQLite Bulk insert: 6 ms
```

## 100 line of data

```
Total Execution Time for SQL: 9 ms
Total Execution Time for SQLite: 265 ms
Total Execution Time for SQLite Bulk insert: 3 ms
```

Kuva 18. Ohjelmisto tulostaa mitatut ajat tietojen tallentamisesta.

Kuten kuvassa 18 näkyvistä tietojen tallennusnopeuksista voi päätellä, SQLite transaktion käytön kanssa on huomattavasti nopeampi kuin SQL CE:tä käyttäen. Miljoona riviä dataa kesti SQLiten ja transaktion käytön kanssa vain 3 190 millisekuntia, kun taas SQL CE:n kanssa se kesti 94 365 millisekuntia. Testauksessa kokeiltiin myös SQLiten tallennusnopeutta ilman transaktion käyttöä, ja se osoittautui olevan hitain menetelmä tiedonsiirtonopeudeltaan. Suoritetun nopeusvertailun jälkeen vertailtiin tiedostokokoja. SQLiten tiedostokoko oli myös huomattavasti pienempi kuin SQL CE:n tietokanta, noin puolet siitä. Testauksesta voidaan päätellä, että SQLiten ja transaktion käyttö osoittautui parhaaksi vaihtoehdoksi suorituskyvyltään ja pienen tiedostokoon ansiosta.

## 4 Virtuaalitodellisuuden tekniikka

Virtuaalitodellisuus (VR) on tietokonesimulaatio, jossa pyritään simuloimaan keinote-koista tai todellista ympäristöä käyttäjälle. Virtuaaliympäristössä käyttäjät pystyvät kat- somaan ympärilleen, liikkumaan ja olemaan vuorovaikutuksessa 3D-objektien kanssa. [27.] VR-laitteiston tunnistaa helposti virtuaalilaseista, jotka puetaan päähän (engl. Head-mounted display (HMD)) ja peittävät käyttäjän näkökentän kokonaan. Kuvassa 19 esitetään laaja valikoima virtuaalilaseja erilaisille laitteille.

### MOBILE HEADSETS



GOOGLE DAYDREAM



SAMSUNG GEAR VR



GOOGLE CARDBOARD

### DESKTOP HEADSETS



HTC VIVE



OCULUS RIFT

Kuva 19. Erityyppisiä virtuaalitodellisuuslaitteita.

Sellaiset laitteet kuin HTC Vive-, Oculus Rift- ja Valve Index -virtuaalilasit kytketään tie- tokoneeseen kiinni, kun taas esimerkiksi Sony PlayStation VR kytketään pelikonsoliin kiinni. Tällä hetkellä suosituimmat virtuaalilasit ovat HTC Vive -sarja, Oculus Rift, Sony PlayStation VR ja Valve Index. [28.]

Insinööriyössä valittiin HTC Vive Pro -virtuaalilasit käyttöön, koska tiimin jäsenillä oli ai- kaisempaa kokemusta HTC Vive -sarjan virtuaalilaseista. Virtuaalilaseissa olevat tuki- asemat mahdollistavat tarkan seurannan liikkuvuusalueella. Lisäksi HTC Vive Pro on

suorituskyvyltään tehokas ja näytön resoluutio oli tehtävään sopivan tarkka. Ohjainten avulla pystytään liikkumaan virtuaaliympäristössä ilman, että liikuttaisiin todellisessa ympäristössä.

#### 4.1 Virtuaalilasit

Kuvassa 20 esitetyssä HTC Vive Pro -virtuaalilaseissa on kaksois-Amoled 3.5"-näyttö, jonka resoluutio on 1440 x 1600 pikseliä silmää kohden (2880 x 1600 pikseliä yhdistettynä). Näytön virkistystaajuus on 90 Hz ja näkökenttä on 110 astetta. Virtuaalilasissa on mm. SteamVR-seuranta, G-sensori (kiihtyvyyssanturi), gyroskooppi ja silmien mukavuusasetus (IPD). [29.] Vive Pro -virtuaalilaseissa on myös kaksi etuosakameraa, joiden avulla laite pystytään muuttamaan lisätty todellisuusversioksi (engl. Augmented Reality (AR)). Uusien työkalujen avulla kamerat voivat havaita syvyyttä ja paikkatietoa ja siirtyä saumattomasti VR- ja AR-maailmojen välillä. [30.] Kameroiden avulla pystytään myös näkemään todellinen työalue, jos käyttäjä on astumassa virtuaalisen rajan ulkopuolelle.



Kuva 20. HTC Vive Pro -paketti.

#### 4.2 Tukiasemat

HTC Vive Pro -laitteessa on kaksi tukiasemaa (kuva 20). Tukiasemat asennetaan pään korkeuden yläpuolelle, mieluiten yli kahden metrin korkeuteen. Tukiasemien tulee olla

asennettuna tiukasti seinään tai telineeseen kiinni, mistä ne eivät voi helposti pudota tai liikkua. Tukiasemilla on 120 asteen näkökenttä ja niiden kulmia säädetään 30–45 asteen välillä, jotta saadaan mahdollisimman iso tunnistettava liikkuvuusalue. Kahdella tukiasemalla voidaan saada maksimissaan 7 m x 7 m:n alue ja minimissään 2 m x 1,5 m liikkuvuustilaa. Neljällä tukiasemalla on mahdollista suurentaa aluetta jopa 10 m x 10 m:n kokoiseksi. [31.]



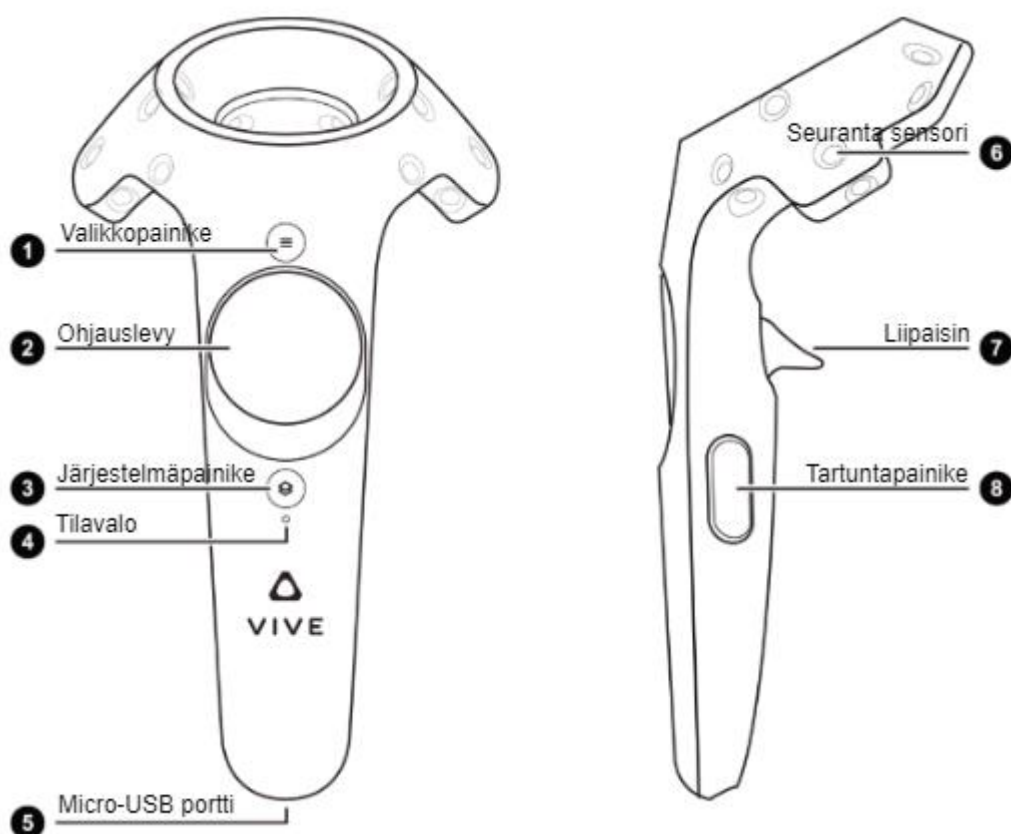
Kuva 21. Kalibroitu alue, jossa tukiasemat tunnistavat käyttäjän liikettä.

Aluetta pitää kalibroida, kun tukiasemia on asennettu paikoilleen. Kalibrointi tehdään asettamalla ohjaimet maahan kahden tukiasemien välille. Kalibroinnin jälkeen piirretään tunnistettava alue ohjaimen liipaisinta painaen (kuva 21). Kalibrointiin käytettiin SteamVR-ohjelmaa.

#### 4.3 Ohjaimet

HTC Vive Pro -laitteeseen kuuluu myös kaksi ohjainta. Ohjaimen avulla käyttäjä pystyy olemaan vuorovaikutuksessa virtuaaliympäristön 3D-objektien kanssa ja liikkumaan virtuaaliympäristössä ilman, että liikkuu oikeassa ympäristössä. Ohjaimen päällä on sensoreita, joita tukiasemat seuraavat jatkuvasti (kuva 22).





Kuva 22. HTC Vive Pro -ohjaimen painikkeet ja toiminnot.

Kuvassa 22 nähdään ohjaimen painikkeiden toimintoja. Valikkopainikkeessa päästään HTC Vive Pron yleisiin asetuksiin. Asetuksissa pystytään säätämään näytön resoluutiota, kameran asetusta ja ääniä. Ohjauslevyn tehtävänä on liikuttaa käyttäjää virtuaalissa työympäristössä. Vasemmasta ohjauslevystä ylöspäin painamalla käyttäjä pääsee virtuaaliympäristössä liikkumaan eteenpäin ja taaksepäin pääsee painamalla alaspäin. Järjestelmäpainikkeella voidaan käynnistää ja sammuttaa ohjain painamalla kolme sekuntia. Tilavallo kertoo ohjaimen tilan. Liipaisinpainike on ohjaimen tärkein painike, sillä liipaisinpainikkeella pystytään valitsemaan valikosta erilaisia tiloja ja toimintoja. Liipaisinpainikkeella pystytään myös tarttumaan 3D-objekteista kiinni, kuten työkalusta ja työkalupaleesta. Tämä toiminto tehdään viemällä ohjain objektien päälle ja painamalla liipaisinpainiketta pohjassa. Objekti muuttuu virtuaaliympäristössä siniseksi, kun käyttäjä vie ohjaimen objektien päälle. Viimeisenä on tartuntapainike, jolla pystytään ns. kaukosiirtymään. Painiketta pohjassa painamalla ohjaimen päähän ilmestyy kaari, jolla maahan osoittaessa ja liipaisinpainiketta painettaessa käyttäjä pääsee siirtymään tähdättyyn pisteeseen.

## 5 Virtuaalityökierto

### 5.1 Suunnittelu

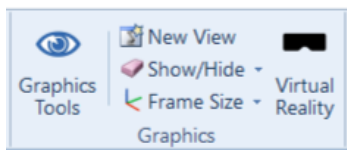
Työssä pohdittiin teollisuusroboteille uusia opetusmenetelmiä, jotka olisivat käyttäjäystävällisempi ja helpompi tapa esitellä työnohjausta asiakkaille. Ideoita olivat lisätyn todellisuuden ja virtuaalitodellisuuden käyttö. Suunnitteluvaiheessa huomioitiin 3D-mallien pienet yksityiskohdat, ja työssä haluttiin tarkkaa suoritusta 3D-mallien kanssa. Simulointiympäristössä työkappaleissa esiintyvien pienten osien, kuten jouset, mutterit ja ruuvit, tulisi olla helposti purettavia kappaleista, jos suoritettaisiin huolto-opastusta virtuaalisesti. Lisäksi huomioitiin robottisolujen koot simulointiympäristössä. Virtuaalitodellisuuden avulla pystyttiin liikkumaan virtuaalitilassa joustavasti ja tutkimaan pieniä yksityiskohtia objekteista ohjainten avulla, kun taas lisätyn todellisuuden tilassa ei pystytty olemaan yhtä tarkasti vuorovaikutuksessa pienten objektien kanssa. Lisäksi virtuaalitodellisuudessa pystyttiin liikkumaan virtuaalisesti ylöspäin ja alaspäin riippuen robottisolun koosta virtuaaliympäristössä. RobotStudio tuki myös työskentelyä virtuaalitilassa, kun taas lisätyn todellisuuden kanssa ei ollut tukia saatavilla.

Työssä päädyttiin virtuaalitodellisuuteen, koska se mahdollistaa tarkan vuorovaikutuksen 3D-objektien kanssa ja joustavan liikkuvuuden virtuaalitilassa ohjainten avulla. Ideana on luoda virtuaaliympäristö ja tuoda siihen robottisolun 3D-malli, jonka sisällä on kaikki tarvittavat työkalut työajaja varten. Virtuaaliympäristöön tuodaan myös 3D-malli ajettavasta työkappaleesta. Työnohjauksessa simuloidaan työkappaleen pinnan hiontaa. Suoritetun työnohjauksen jälkeen otetaan RAPID-koodi talteen ja siirretään se fyysiseen teollisuusrobottiin.

### 5.2 Komponentit ja työkalut

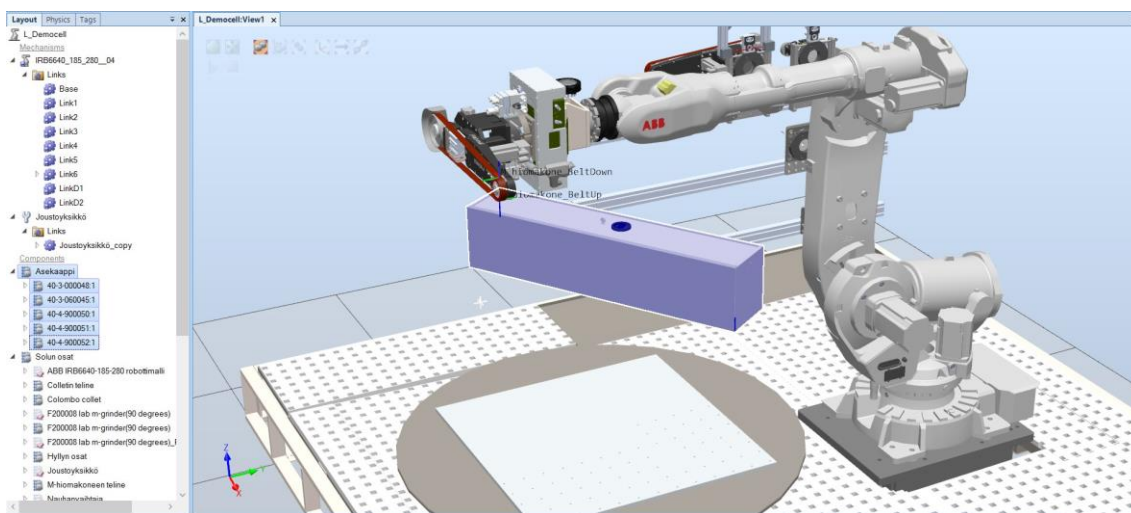
Virtuaalitodellisuuden toteutus tehtiin RobotStudio-ohjelmassa. RobotStudioon on mahdollista tuoda omia 3D-malleja työympäristöön. Toteutuksessa käytettiin 3D-malleja S-solusta ja L-solusta. Ajettavina työkappaleina toimivat lentokoneen turbiinien siivet, vaihdelaatikko ja asekaappi. RobotStudion kotivalikkoon ilmestyy virtuaalitodellisuuspainike,

kun luodaan uusi työtila (kuva 23). Virtuaalilasien ja ohjainten pitää olla kalibroituina, ennen kuin pystytään käyttämään virtuaalitilaa.



Kuva 23. RobotStudion valikko, jossa on virtuaalitodellisuuspainike.

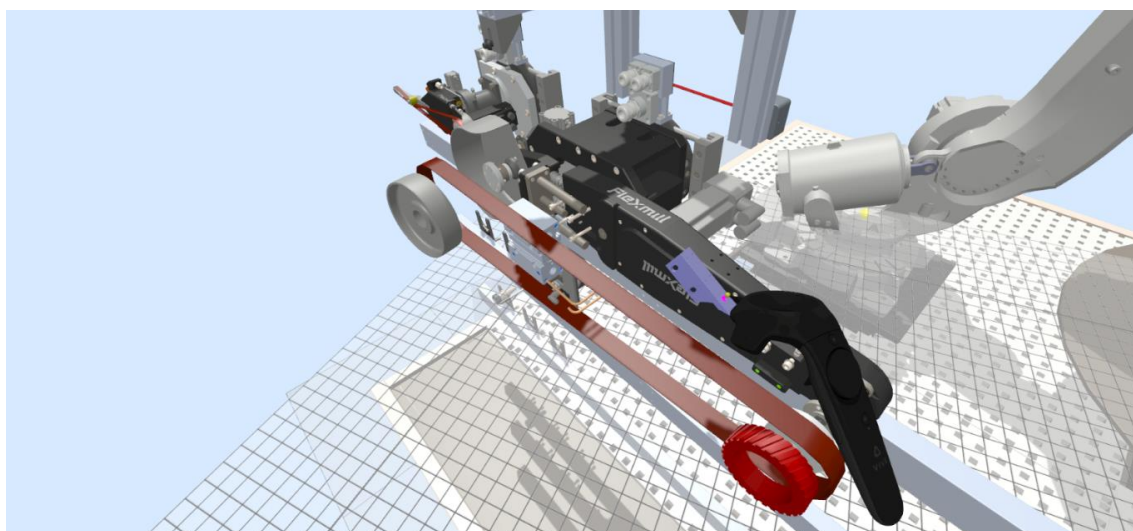
RobotStudion kirjastossa on valmiiksi ABB:n omia IRB-teollisuusrobotteja ja työkaluja, jotka voidaan suoraan linkittää robottiin raahaa ja pudota -menetelmällä. Tässä työssä kuitenkin käytettiin omia 3D-malleja robottisoluista ja työkaluista, joita RobotStudio ei välttämättä tunnista. Tuodut mallit näkyvät RobotStudiassa Layout-ikkunassa irrallisina komponentteina, jotka kootaan yhtenäiseksi esineeksi ennen työnohjausta. Yhdistämällä komponenteiksi saadaan mallin irralliset osat yhtenäiseksi ryhmäksi. Tämän avulla saadaan siistimpi kirjasto, jonka kanssa on myöhemmin helpompi työskennellä. RobotStudion yläpalkista valitaan Modeling-välilehti, jossa pystytään luomaan komponenttiryhmä (engl. Component Group). Komponenttiryhmälle annetaan kuvaava nimi, ja irralliset komponentit liitetään siihen ryhmään (kuva 24).



Kuva 24. RobotStudiassa korostettuna komponenttiryhmä nimeltä Asekaappi.

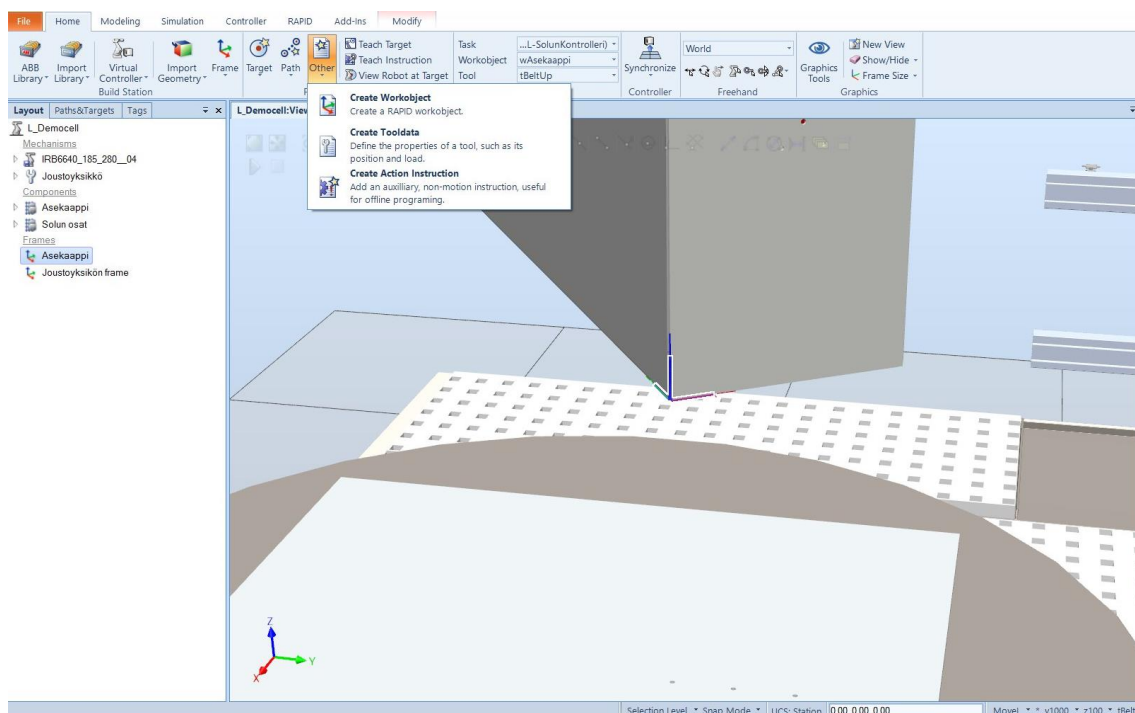
Työkalu luodaan samalla menetelmällä, mutta komponenttiryhmän sijaan luodaan uusi työkalu. Komponenteissa on paljon erilaisia työkaluja, jotka eivät ole yhdistettynä toisiinsa. Työkaluja ovat esim. erilaiset hiomapäät, jatkovarret ja liittimet. Työkalujen luonnilla yhdistetään nämä komponenttityökalut yhtenäiseksi työkaluksi, jotta saadaan kappale, jonka kanssa pystytään työskentelemään virtuaalillassa. Yhdistetty työkalu mahdollistaa työkalujen liikuttelun kokonaisena kappaleena.

Kuvassa 25 on esitettyä 3D-mallin hiomakone, joka ei ole koottu yhtenäiseksi työkaluksi. Käyttäjä pystyy yksitellen purkamaan ja tarkastelemaan osia koneesta. Tätä menetelmää käytetään, kun halutaan esitellä työkalun huoltoa etänä. RobotStudiassa on mahdollista kutsua toinen käyttäjä mukaan samaan virtuaaliympäristöön.



Kuva 25. RobotStudio virtuaaliympäristössä, missä työkalukomponentit eivät ole yhdistettynä yhtenäiseksi työkaluksi. Työkalukomponentit voidaan purkaa osiksi ja saada räjäytyskuva.

Komponenttiryhmien ja työkalun luonnin jälkeen luodaan uusi työkohte (engl. Workobject) työkappaleelle. RobotStudiassa on oletuksena wobj0-niminen työkohte, mutta tässä tapauksessa luodaan uusi. Työkohte tarkoittaa koordinaatiston luontia, jota käytetään kuvaamaan työkappaleen sijaintia. Työkohte luodaan kotivalikon other-painikkeen kautta. Other-painikkeesta ilmestyy määrittämisvalikko, josta valitaan "Create Workobject" (kuva 26). Työkohteelle sitten annetaan nimi ja määritetään koordinaatisto käyttäjän ja maailman suhteen. [33, s. 83.] Kuvassa 26 on esiteltynä asekaapin työkohte ja sen kolmiulotteinen koordinaatisto, joka on asetettuna asekaapin alanurkkaan. Työkohteen luonnissa halutaan Z-akseli osoittamaan työkappaleesta työkalua päin.



Kuva 26. Työkohteen luominen ja sen vertaaminen suhteessa maailmaan.

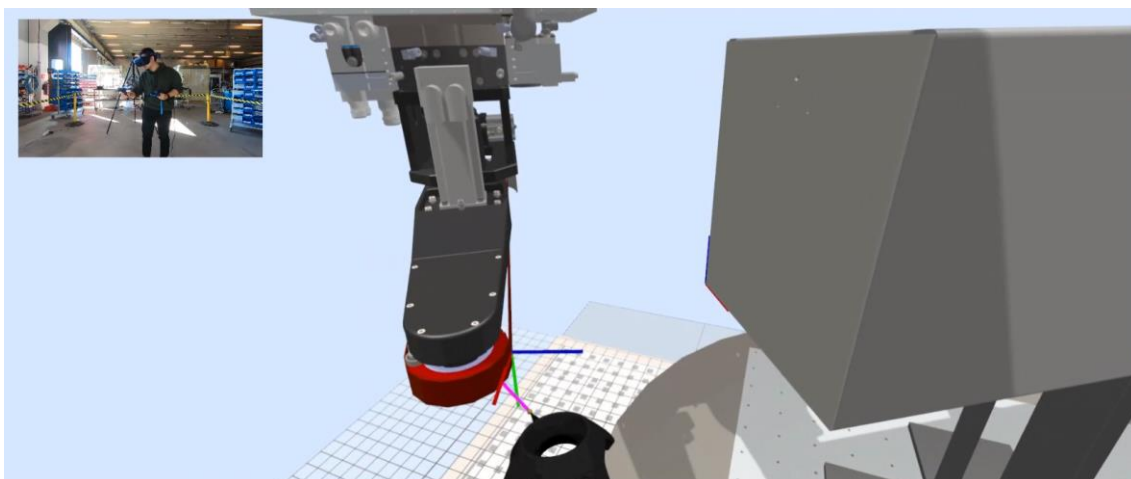
Työkohteen lisäksi luodaan työkalulle keskipiste (engl. Tool Center Point (TCP)). Kuvasssa 26 näkyy "Create Tooldata" -painike, jonka avulla pystytään luomaan keskipiste työkalulle. RobotStudioissa työkalun nimi on oletuksena tool0, mutta luodaan uusi keskipiste kuvaavalla nimellä. Keskipiste luodaan aluksi asettamalla työkalun pohjaan (engl. base) koordinaatisto. Työssä haluttiin kuitenkin keskipiste työkalun kärkeen. Keskipiste voidaan asettaa työkalun kärkeen luomalla erillinen koordinaatisto (engl. frame) työkalun kärkeen, jonka X-, Y- ja Z-arvot kopioidaan luodulle keskipisteelle. Erillisen koordinaatiston luonnin yhteydessä pystytään tunnistamaan työkalun pinta, minkä takia on helpompaa kopioida arvot keskipisteelle. Työkalun keskipisteen koordinaatisto määritellään siten, että Z-akseli osoittaa työkalua päin. Työssä asetettiin hiomakoneen kärkeen keskipiste, joka ottaa kontaktia työkalupäähän ajon aikana (kuva 27, s. 32). Kotivalikosta vaihdetaan määritetty työkohte ja keskipiste ennen työnohjausta.

### 5.3 Virtuaalinen työympäristö ja työnohjaus

Virtuaalisessa työympäristössä käyttäjä näkee tilan, jossa on robottisolu ja työstettävä työkalupala. Virtuaaliympäristössä käyttäjä pääsee vapaasti liikkumaan ohjainta käyttäen

tai kävellen tunnistettavan alueen sisällä. Liikkuvuusalue on n. 3 m x 3 m, joka on eristettynä muusta liikenteestä. Työskennellessään virtuaaliympäristössä käyttäjä ei ole täysin tietoinen siitä, mitä ympärillä tapahtuu, koska virtuaalilasit peittävät näkyvyyden. Asetuksista on säädetty niin, että virtuaalilasien kamerat käynnistyvät, kun käyttäjä on astumassa kalibroidun rajan reunalle. Näin käyttäjä näkee todellisen ympäristön ja pystytään lisäämään työturvallisuutta.

Virtuaalimaailmassa käyttäjä seisoo keinotekoisessa ympäristössä, jossa on robottisolu ja työkappale. Käyttäjä pystyy virtuaalimaailmassa olemaan vuorovaikutuksessa teollisuusrobotin ja työkappaleen kanssa ohjainten avulla. Robotin käsivarresta pystytään tarttumaan kiinni ja raahaamaan se haluttuun paikkaan ja robotin nivelet seuraavat mukana (kuva 27). Työnohjaus tehdään asettamalla opetuspisteitä virtuaaliympäristöön. Robotti seuraa asetettuja pisteitä ja tallentaa koordinaatistot muistiin.



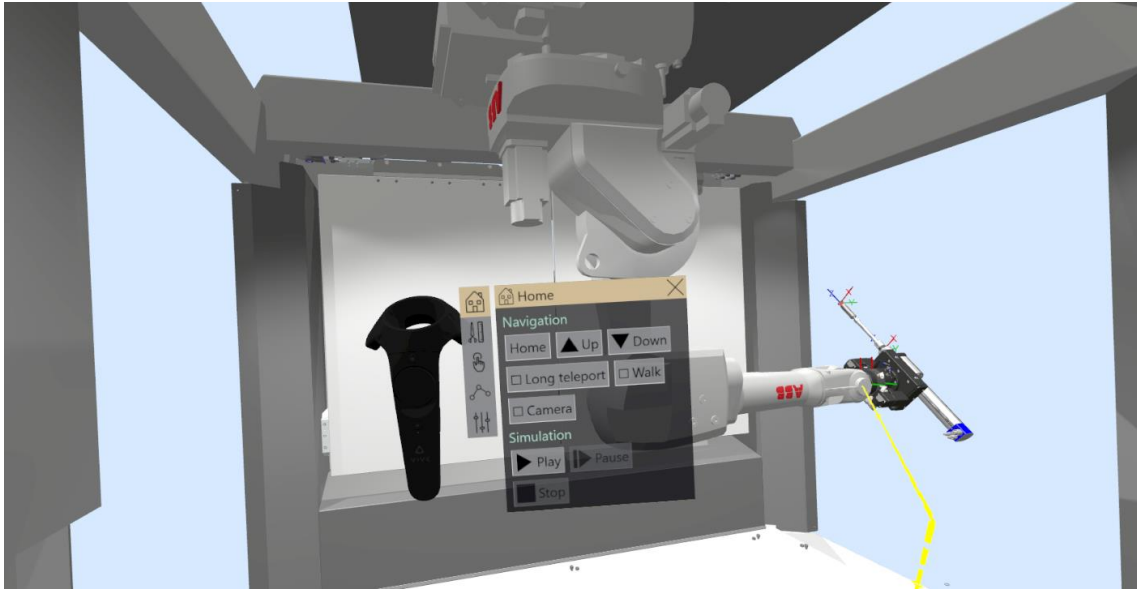
Kuva 27. Käyttäjä operoimassa virtuaaliympäristössä.

Virtuaalimaailmassa on valikko integroituna vasempaan ohjaimeen. Valikko koostuu home-, edit-, manipulate- ja path-painikkeista. Oikean ohjaimen päässä on integroituna pieni osoitin, jonka avulla voidaan valita vasemman valikon painikkeita. Painikkeisiin pääsee osoittamalla ja liipaisupainiketta painamalla.

Kotivalikossa käyttäjä pystyy säätelemään liikkuvuusasetuksia. Navigation palkissa on mahdollista aktivoida joko Long teleport -komennolla eli kaukosiirtymällä tai Walk-komennolla, jolla pystytään liikkumaan ohjauslevystä objektien yli. Up- ja Down-painikkeilla

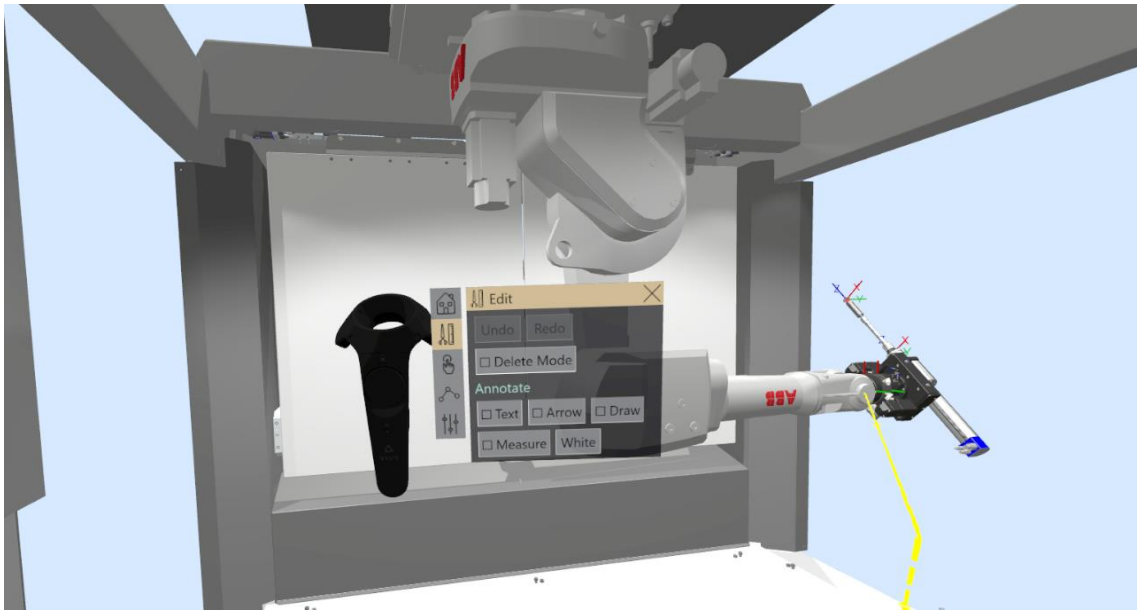


käyttäjä pystyy virtuaalisesti liikkumaan ylös ja alas. Tarvittaessa on kamerapainike, jonka avulla voidaan nähdä virtuaalilasien kautta reaaliaikaista kuvaa oikeasta ympäristöstä. Simulation-palkissa pystytään ajamaan suoritettut työkierrot suoraan virtuaaliympäristössä (kuva 28).



Kuva 28. Virtuaalimaailman kotivalikon painikkeet.

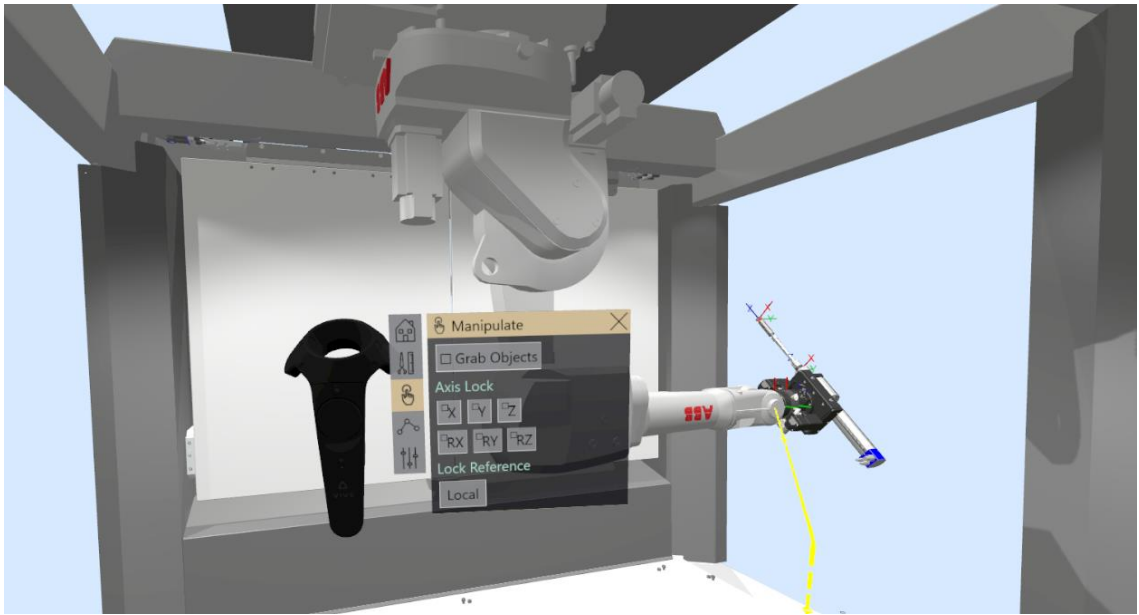
Muokkausvalikossa pystytään kumoamaan ja toistamaan robottien liikkeitä. Muokkausvalikossa on Annotate-palkki, jossa on mahdollista asettaa tekstilaatikoita, piirrellä, lisätä nuolen suuntia ja mitata objekteja. Delete-painikkeella voidaan poistaa ylimääräiset mittaukset, tekstit ja piirrokset virtuaaliympäristöstä (kuva 29).



Kuva 29. Muokkausvalikon painikkeet.

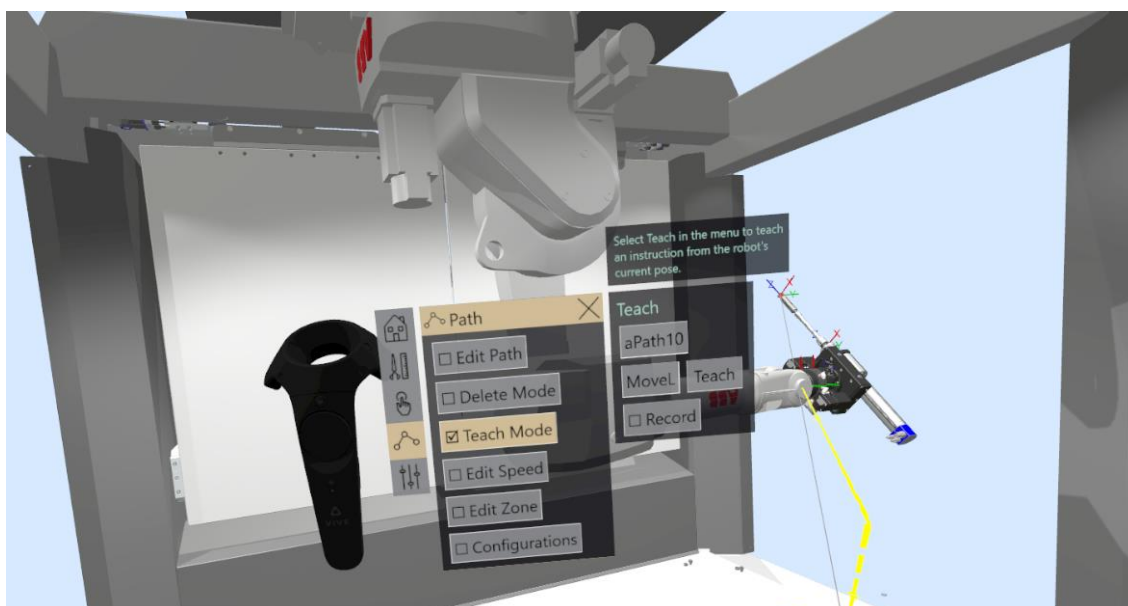
Käsittelyvalikossa pystytään lukitsemaan robotin akselit ja lukitusviittausta. Käsittelyssä pystytään kytkemään esineen tarttuminen päälle, jolloin pystytään tarttumaan objekteista kiinni. Akselien lukituksessa on mahdollista lukita X-, Y- ja Z-akselit, jolloin pystytään rajaamaan robottien liikkuvuutta maailman suhteen. RX-, RY- ja RZ-akselit sallivat robotin liikkumista kaarevasti, esim. työkalun kärki on lukittuna työkappaleeseen kiinni, mutta robotin käsivarsi liikkuu. Työkaluilla on omat fysiikkansa, ja sen takia "Grab Objects" -painikkeella ei ole vaikutusta niihin (kuva 30).





Kuva 30. Käsittelyvalikon painikkeet.

Viimeisenä on robotin työnohjaus, joka tehdään polkuvalikosta. Polkuvalikossa pystytään luomaan, muokkaamaan ja poistamaan opetettuja työreittejä sekä muokkaamaan työreittien nopeuksia ja kulmia. Valikossa tärkein painike on opetuspainike (engl. Teach Mode). Opetuspainikkeessa pystytään valitsemaan työpolku, johon tallennetaan opetus-pisteet. Työnohjausta pystytään suorittamaan kahdella tavalla, joista ensimmäinen tapa on Teach-menetelmä. Siinä liikutellaan työkalua ja asetetaan manuaalisesti piste kerrallaan virtuaaliympäristöön. Toinen menetelmä on Record-painike, joka mahdollistaa automaattisen pistetallennuksen, kun työkalua liikutetaan (kuva 31).



Kuva 31. Polkuvalikon painikkeet.

Opetuspainikkeessa pystytään määrittelemään työkalupolkua ja liikettä. Kuvassa 31 on esitettynä Teach-taulukon sisällä olevat aPath10- ja MoveL-painikkeet. Painikkeista aPath10 määrittää senhetkisen työpolun ja MoveL on robotin liike. Painikkeesta on mahdollista vaihtaa seuraavia liikkeitä:

- lineaarinen liike (MoveL) lyh. Move Linear
- nivelen liike (MoveJ) lyh. Joint Movement
- absoluuttinen liike (MoveAbsJ) lyh. More Absolute Joint
- ulkoinen liike (MoveExtJ) lyh. More External Joints.

Työn alussa robotti on asetettuna nolla-asentoon, jolloin kaikki akselien nivelten arvot ovat nollia. Kun robottia liikutetaan työkappaleen pinnalle, käytetään MoveJ-liikettä, kun nolla-asennon ja työkappaleen välillä ei ole suoraa linjaa ja halutaan lähestyä työkappaletta täydellä nopeudella. Robotti ja ulkoiset akselit siirtyvät määränpäähän epälineaarista polkua pitkin, jolloin akselit pääsevät määränpäähän samaan aikaan [32, s. 253]. Liikettä vaihdetaan MoveL-liikkeeksi, kun työkalu on tarpeeksi lähellä työkappaletta ja halutaan lineaarinen liike. MoveL-liikettä käytetään, kun työkalun keskipistettä siirretään lineaarisesti annettuun määränpäähän eli tässä tapauksessa työkappaleen pinnalle. [32, s. 264.]

## 5.4 Prototyyppi

Virtuaalitodellisuudessa kokeiltiin erilaisia työkiertoja erilaisilla työkappaleilla. Työssä suoritettiin asekaapille työajo, joka olisi potentiaalinen työ suorittaa fyysisellä teollisuusrobotilla. Virtuaalitodellisuudessa oli käytössä L-solu ja asekaappi, jonka tehtävänä oli pinnanhiointa työkappaleen reunalle. Työtä ohjattiin kokonaan virtuaalitodellisuutta käyttäen. Työssä saatiin suoritettua työnohjaus ja pystyttiin muokkaamaan liikkeiden nopeuksia ja kulmia. Suoritetun työajon liikkeet ladattiin muistitikulle ja siirrettiin fyysiselle L-solun robotille. L-solussa tarkistettiin vielä, täsmäävätkö koordinaatit toisiinsa. Työssä vertailtiin RobotStudion arvoja ja L-solun FlexPendant-ohjaimen arvoja toisiinsa. Lopputuloksena oli parinkymmenen millimetrin eroja, mutta ne saatiin korjattua nopeasti (kuva 32).



Kuva 32. Virtuaalinen ja fyysinen työajo L-solussa.

Opetettu työliike suoritettiin aluksi hiomakone pois päältä kytkettynä, jolloin oli mahdollista seurata robotin liikettä solun sisällä. Robotin nopeus oli säädetty alhaiseksi, jolloin seuraaminen vieressä oli turvallista. Työn tarkistuksessa robotin liikkeet olivat hyvin rauhallisia ja tarkkoja virtuaalitoteutuksessa. Liikeradat näyttivät kiitettäviltä, eikä suuria eroja havaittu. Tarkistuksen jälkeen sama liikerata toteutettiin hiomakone päälle kytkettynä ja tuotantonopeudella.

## 6 Virtuaalitodellisuus opettamisessa

Virtuaalitodellisuuden avulla oli ajatuksena saada käyttäjäystävällisempi tapa ohjata teollisia robotteja ja esitellä työnohjausta asiakkaille virtuaalisesti. Virtuaalitodellisuuden avulla käyttäjä pääsee kokeilemaan erilaisia menetelmiä robotin ohjattavuuteen. Vaikka virtuaalitodellisuuden kautta opetettu työliike on helppoa ja nopeaa, siinä on myös huonot puolensa verrattuna perinteiseen työliikkeeseen robottiohjainta käyttäen.

### 6.1 Fyysisen ja virtuaalisen opetustavan vertailu

Hyvä puoli virtuaalitodellisuudessa on se, että käyttäjä pystyy vapaasti työskentelemään robottisolussa, koska teollisuusrobotti ei aiheuta käyttäjälle hengenvaaraa. Virtuaaliympäristössä teollisuusrobottia pystytään ohjaamaan niin kuin cobotit, eli johdattamisella, mikä ei olisi mahdollista oikeilla teollisuusroboteilla. Työtä pystytään ohjaamaan todella helposti ja nopeasti ja opetettu liikerata pystyttiin näkemään toistonappia painamalla. Käyttäjä ei tarvitse aikaisempaa kokemusta robottiteollisuudesta, koska virtuaalitilassa robottia pystytään ns. rikkomaan eli robotin nivelet voi vetää ääriasentoon tai työkappaleeseen törmätään ilman, että aiheutuisi vahinkoja laitteelle tai käyttäjälle. Virtuaaliympäristössä pystytään purkamaan työkappaleet ja työkalut osiksi ja saamaan räjäytyskuva, jonka avulla on helpompi tutkia ja esitellä kappaleen osia (kuva 25, s. 30).

Huonoja puolia virtuaalitodellisuudessa kuitenkin on myös jonkin verran. Virtuaalitodellisuudessa ei pystytä havaitsemaan törmäyksiä. Tämä tarkoittaa, että haastavampien työkappaleiden kanssa, kuten lentokoneen turbiinin siipi, on vaikeaa työskennellä virtuaaliympäristössä. Myös työkappaleiden pintojen havainnointi ei ole mahdollista virtuaalitodellisuudessa. Virtuaaliympäristössä ei ole huomioituina robotin johdotuksia, eli tietyissä asennoissa voisi aiheutua vahinkoa todelliselle robotille. Virtuaaliympäristössä solumalissa eivät välttämättä näy seinät kuten oikeassa solussa. Tämän takia käyttäjän pitää olla varuillaan, että opetetussa liikkeessä robotti ei törmää seiniin todellisessa työajossa. Virtuaalilasit voivat aiheuttaa myös pahoinvointia riippuen ihmisestä. Virtuaalilasit ovat suurikokoiset päässä ja johdot voivat roikkua jalkojen välissä, mikä lisää kompastumisen riskiä.

Fyysisessä opetustavassa pystytään opettamaan robottia robottiohjaimella tai tietokoneen kautta. Tietokoneessa suoritettu työliike onnistuu robotin omalla robottiohjelmalla. Tietokoneen kautta suoritettussa työnohjauksessa on mahdollista havaita työkappaleen pinnat ja reunat. Lisäksi tietokoneessa on enemmän asetuksia tarjolla verrattuna virtuaalitodellisuuteen. Työnohjaus on haastavampaa kuin virtuaalitodellisuudessa, ja käyttäjältä vaaditaan kokemusta ohjelman käytössä, mikä on aikaa vievä prosessi. Teollisuusrobotin käytössä robottiohjaimella vaaditaan taas työhön perehdytystä ja kokemusta.

## 6.2 Virtuaalitodellisuuden käyttö muilla aloilla

Virtuaalitodellisuutta on käytetty myös muilla teollisuudenaloilla opettamisessa ja esittelyssä. Suomalainen Metso Oyj käyttää virtuaalitodellisuutta MX-kartiomurskaimen esittelyyn ja koulutukseen. Virtuaalitodellisuuden avulla on mahdollista pitää myyntikoulutuksia, jotta saataisiin paremmin esiteltäviä tuotteen tekniset ja teknologiset ominaisuudet mukaansatempaavien kokemusten avulla. Myös laitteiden asennus-, kokoonpano-, korjaus- ja huoltoprosessia pystytään visualisoimaan virtuaalitodellisuuden avulla ja esittämään näyttelyissä ja messuilla. Virtuaalitoteutuksella on mahdollista saada palautetta asiakkaalta jo suunnittelun varhaisessa vaiheessa. [34.]

OpenAI-yritys käyttää myös virtuaalitodellisuutta robottien kouluttamiseen. Robottia ohjataan kerran suorittamaan toiminto virtuaalitodellisuudessa, minkä jälkeen robotti kykenee toistamaan opetetun toiminnon. Tehtäviä ovat esim. värikuutioiden järjestäminen tai kokoaminen torniksi. Toteutuksessa on käytetty robotin kameraa ja virtuaalitodellisuuden avulla on tallennettu työnohjaus, joita koneoppiminen vertailee keskenään ja saadaan suoritettava työkierto. [35; 36.]

## 7 Yhteenveto

Insinööriyön tarkoituksena oli tutkia ja kehittää asiakasyrityksen tämänhetkistä tietojenkäsittelyä ja tutkia erilaisia menetelmiä robottien ohjaukseen. Työssä suunniteltiin ja toteutettiin uusia ominaisuuksia ohjelmiston puolella, kuten tietojenkäsittelyn suorituskyvyn

parantamista ja erilaisia ohjelmiston muutoksia. Työssä päästiin tutkimaan ja toteuttamaan teollisuusrobotin työnohjausta offline-ohjelmoinnin ja virtuaalitodellisuuden avulla. Lisäksi työssä päästiin tutkimaan myös erilaisia robottisoluja ja niiden ominaisuuksia ja toimintoja. Työ koostui ohjelmistotuotannosta robottiteollisuuteen saakka, ja työtä tehtiin sekä toimisto- että tehdaspuolella.

Ohjelmiston puolella onnistui suorituskyvyn parantaminen tietojenkäsittelyssä tietokantaan. Testauksessa saatiin iso määrä dataa tallennettua kiitettävällä nopeudella. Suoritusta tehtiin viikon mittaisella työjaksolla, johon sisältyivät testaukset. Myös ohjelmistomuutokset onnistuivat. Kirjoitushetkellä saatiin soluohjaimesta tietoa tallennettua väliaikaistiedostoihin, CSV-tiedostoon ja tietokantaan. Työn määrä oli suuri, mutta työ saatiin toteutettua aikataulussa. Lopputuloksena saatiin parannettua suorituskkyä tietojenkäsittelyssä tietokantaan, paljon lisää toimintoja ohjelmistoon ja työnohjaus kokonaan virtuaalitodellisuutta käyttäen.

Insinööritö oli haastava, koska työtehtävään liittyi ohjelmistotuotannon lisäksi paljon robottiteollisuutta, josta minulla ei ollut aikaisempaa kokemusta. Robottiohjelmisto, kuten RobotStudio ja RAPID-ohjelmointikieli, olivat aivan uutta. Työssä pääsin ensimmäistä kertaa etäohjelmoimaan teollisuusrobottia tietokoneen ja virtuaalitodellisuuden kautta. Pääsin myös kokeilemaan robottiohjaimella työskentelyä, jotta pystyin vertailemaan työn menetelmiä keskenään. Haasteellisessa insinööritöössä pääsin monipuolisiin tehtäviin ja oppimaan paljon uutta.

Jatkokehitystä on vielä paljon, ja uusia menetelmiä robotin ohjaukseen on jo suunnitteilla. Kirjoitushetkellä saatiin toteutettua työnohjaus virtuaalitodellisuutta käyttäen, mutta ajatuksena olisi tehdä samanlainen menetelmä huollon puolelle käyttäen lisättyä todellisuutta.

## Lähteet

- 1 The Rise of Collaborative Industrial Robots in Advanced Manufacturing. 2020. Verkkoaineisto. Interesting Engineering. <<https://interestingengineering.com/the-rise-of-collaborative-industrial-robots-in-advanced-manufacturing>>. Luettu 24.8.2020.
- 2 Flexmill History. 2018. Verkkoaineisto. Flexmill Oy. <<https://www.flexmill.fi/company/history/>>. Luettu 20.3.2020.
- 3 World's Top 10 Industrial Robot Manufacturers. 2019. Market Research Reports. Verkkoaineisto. <<https://www.marketresearchreports.com/blog/2019/05/08/world%E2%80%99s-top-10-industrial-robot-manufacturers>>. Luettu 31.3.2020.
- 4 Industrial Robots History. 2018. Verkkoaineisto. Engineering Channel. <<https://engineers-channel.blogspot.com/p/industrial-robots-history.html>>. Luettu 27.8.2020.
- 5 Programmed article transfer. 1961. Verkkoaineisto. Google Patents. <<https://patents.google.com/patent/US2988237>>. Luettu 29.3.2020.
- 6 Joseph Engelberger the Father of Robotics. 2020. Verkkoaineisto. Robotic Industries Association. <<https://www.robotics.org/joseph-engelberger/about.cfm>>. Luettu 29.8.2020.
- 7 Pearce, Jeremy. 2011. George C. Devol, Inventor of Robot Arm, Dies at 99. Verkkoaineisto. <[https://www.nytimes.com/2011/08/16/business/george-devol-developer-of-robot-arm-dies-at-99.html?\\_r=0](https://www.nytimes.com/2011/08/16/business/george-devol-developer-of-robot-arm-dies-at-99.html?_r=0)>. Luettu 29.8.2020.
- 8 ABB Robotics Historical milestones. 2020. Verkkoaineisto. ABB. <<https://new.abb.com/products/robotics/about-us/historical-milestones>>. Luettu 11.8.2020.
- 9 Rise of the robot. Verkkoaineisto. ABB Robotics. <[https://library.e.abb.com/public/c1ecdae9855e5452c1257d200053f273/24-31%202m433\\_EN\\_72dpi.pdf](https://library.e.abb.com/public/c1ecdae9855e5452c1257d200053f273/24-31%202m433_EN_72dpi.pdf)>. Luettu 30.8.2020.
- 10 IntelC8008-1. 2005. Verkkoaineisto. The cpushack museum. <<http://www.cpushack.com/chippics/Intel/8008/IntelC8008-1.html>>. Luettu 31.8.2020.
- 11 IRB 1600. 2020. Verkkoaineisto. ABB. <<https://new.abb.com/products/robotics/industrial-robots/irb-1600>>. Luettu 2.9.2020.

- 12 S-series. 2018. Verkkoaineisto. Flexmill Oy. <<https://www.flexmill.fi/products/s-series/>>. Luettu 2.9.2020.
- 13 Arovirta, Kimmo. 2020. Vanhempi ohjelmoija. Flexmill Oy, Nurmijärvi. Merkintäso-lun toiminnon koulutus 2.9.2020.
- 14 L-series. 2018. Verkkoaineisto. Flexmill Oy. <<https://www.flexmill.fi/products/l-series/>>. Luettu 7.9.2020.
- 15 Technical data for the IRB 6640 industrial robot. 2020. Verkkoaineisto. ABB. <<https://new.abb.com/products/robotics/industrial-robots/irb-6640/irb-6640-data>>. Luettu 7.9.2020.
- 16 Programming Robots. Verkkoaineisto. thnet. <<https://thnet.co.uk/thnet/ro-bots/33.htm#On-Line>>. Luettu 10.9.2020.
- 17 Robotics: Online Programming – Teach Pendant & Lead-through. 2015. Verkko-aineisto. YouTube. <[https://www.youtube.com/watch?v=EA6pWwNI\\_wg](https://www.youtube.com/watch?v=EA6pWwNI_wg)>. Kat-sottu 10.9.2020.
- 18 Nichols, Megan R. 2019. How to Train Employees to Work With Cobots Safely. Verkkoaineisto. <<https://www.roboticsbusinessreview.com/cro/how-to-train-em-ployees-to-work-with-cobots-safely/>>. Luettu 10.9.2020.
- 19 RobotStudio. 2020. Verkkoaineisto. ABB. <<https://new.abb.com/products/robot-ics/robotstudio>>. Luettu 10.9.2020.
- 20 Create complex 3D geometry for multi-axis, high—speed CNC machining. 2020. Verkkoaineisto. Autodesk. <<https://www.autodesk.com/products/powermill/over-view>>. Luettu 14.9.2020.
- 21 Nousiainen, Lauri. 2020. Experimental investigation of robotic surface finishing using design of experiments. Diplomityö. Aalto-yliopisto. Aaltodoc.
- 22 CSV – Comma Separated Values. 2018. Verkkoaineisto. Data Hub. <<https://data-hub.io/docs/data-packages/csv>>. Luettu 20.9.2020.
- 23 .INI File Extension. 2020. Verkkoaineisto. FileInfo. <<https://fileinfo.com/exten-sion/ini>>. Luettu 22.9.2020.
- 24 Microsoft SQL Server Compact 4.0 is available for download and use with Web-Matrix. 2011. Verkkoaineisto. Microsoft. <<https://cloudblogs.mi-crosoft.com/sqlserver/2011/01/13/microsoft-sql-server-compact-4-0-is-available-for-download-and-use-with-webmatrix/>>. Luettu 28.9.2020.



- 25 SQLite - Overview. 2020. Verkkoaineisto. Tutorialspoint. <[https://www.tutorialspoint.com/sqlite/sqlite\\_overview.htm](https://www.tutorialspoint.com/sqlite/sqlite_overview.htm)>. Luettu 29.9.2020.
- 26 Transactions. 2020. Verkkoaineisto. Microsoft Documentation. <<https://docs.microsoft.com/en-us/dotnet/standard/data/sqlite/transactions>>. Luettu 29.9.2020.
- 27 Bardi, Joe. 2019. What is Virtual Reality? [Definition and Examples]. Verkkoaineisto. <<https://www.marxentlabs.com/what-is-virtual-reality/>>. Luettu 30.9.2020.
- 28 Stein, Scott. 2020. Best VR headset for 2020. Verkkoaineisto. <<https://www.cnet.com/news/best-vr-headset-for-2020/>>. Luettu 30.9.2020.
- 29 Vive Pro. 2020. Verkkoaineisto. Vive. <<https://www.vive.com/eu/product/vive-pro/>>. Luettu 30.6.2020.
- 30 Lang, Ben. 2018. Vive Pro Gets AR Capabilities Thanks to HTC's New Tools for Front-facing Cameras. Verkkoaineisto. <<https://www.roadtovr.com/vive-pro-turns-into-an-ar-dev-kit-thanks-to-htcs-new-tools-for-front-facing-cameras/>>. Luettu 30.6.2020.
- 31 Asennusohje. 2018. Käyttöopas. HTC Vive Pro.
- 32 Technical reference manual RAPID Instructions, Functions and Data types. 2004. Asennusohje. ABB Robotics. <[https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual\\_RAPID\\_3HAC16581-1\\_revJ\\_en.pdf#page=257&zoom=100,93,628](https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf#page=257&zoom=100,93,628)>. Luettu 7.8.2020.
- 33 Operating manual RobotStudio. 2009. Verkkoaineisto. ABB. <[https://library.e.abb.com/public/792b4432c1402c40c1257b4b00521525/3HAC032104-001\\_RevB\\_en.pdf#page=195&zoom=100,93,126](https://library.e.abb.com/public/792b4432c1402c40c1257b4b00521525/3HAC032104-001_RevB_en.pdf#page=195&zoom=100,93,126)>. Luettu 13.10.2020.
- 34 Kasat, Rashmi. 2019. Realizing the true potential of Virtual Reality in industrial applications. Verkkoaineisto. <<https://www.metso.com/blog/mining/virtual-reality-in-industrial-applications/>>. Luettu 21.10.2020.
- 35 Nichols, Greg. 2018. How to train a robot with virtual reality (no coding required). Verkkoaineisto. <<https://www.zdnet.com/article/how-to-train-a-robot-with-virtual-reality-no-coding-required/>>. Luettu 21.10.2020.
- 36 Melnick, Kyle. 2017. Elon Musk-Backed Company Using VR To Train Robots. Verkkoaineisto. <<https://vrscout.com/news/elon-musk-vr-train-robots/>>. Luettu 21.10.2020.