Tampereen yliopisto



Saku Pöysäri

TEKNOLOGIADEMO: DIGITAALISET KAKSOSET TEOLLISUUSROBOTIIKASSA

2025

PIRKANMAA
COUNCIL OF TAMPERE REGION

ETELÄ-POHJANMAAN LIITTO
Regional Council of South Ostrobothnia

Euroopan unionin
osarahoittama

JOHDANTO

Digitaaliset kaksoset ovat nousseet digitaalisten teknologioiden kehittymisen ja digitaalisen siirtymän ansiosta ajankohtaiseksi aiheeksi teollisuudessa. Digitaalinen kaksonen on yksinkertaisimmillaan todellisen järjestelmän ja tästä mallinnetun virtuaalisen vastineen muodostama kokonaisuus, joka mahdollistaa todellisen järjestelmän reaaliaikaisen seurannan, analysoinnin, optimoinnin ja ohjauksen virtuaalisella vastineella. Tästä avautuu uusia mahdollisuuksia tuotannon tehokkuuden parantamiseen. Haasteeksi on kuitenkin muodostunut, kuinka digitaalisia kaksosia rakennetaan ja miten erilaiset digitaaliset kaksoset tulisi luokitella.

Kuten *Optimaalista lisäarvoa teknologiasta* -hankkeella (OPLITE) toteutetussa digitaalisia kaksosia käsittelevästä selvityksestä (*Digitaaliset kaksoset valmistavassa teollisuudessa*) on tullut ilmi, digitaaliset kaksoset voidaan jaotella tiedonsiirron reaaliaikaisuuden perusteella kolmeen ryhmään:

- Digitaalisiin malleihin, joilla ei ole reaaliaikaista automatisoitua tiedonsiirtoa reaalijärjestelmän ja tästä mallinnetun virtuaalisen järjestelmän välillä
- Digitaalisiin varjoihin, joilla on reaaliaikainen automatisoitu tiedonsiirto ainoastaan reaalijärjestelmästä tästä mallinnettuun virtuaaliseen järjestelmään
- Digitaalisiin kaksosiin, joilla on reaaliaikainen automatisoitu tiedonsiirto molempiin suuntiin reaalimaailman ja tästä mallinnetun virtuaalisen järjestelmän välillä

Teollisuudessa yritykset pyrkivät luomaan tuotantoresursseistaan digitaalisia kaksosia, mutta tyypillisesti heidän luomat digitaaliset kaksoset ovat kuitenkin ainoastaan digitaalisia malleja tai vaihtoehtoisesti digitaalisia varjoja ilman reaaliaikaista tiedonsiirtoa molempiin suuntiin. Todelliset digitaaliset kaksoset ovat harvinaisempia, koska näiden toteuttaminen on sekä teknisesti että integroimisen kannalta vaativampia toteuttaa käytännössä.

Tämän selvityksen teknologiademoissa annetaan muutamia käytännönesimerkkejä Tampereen yliopistolta, kuinka digitaalisia varjoja ja kaksosia pystytään luomaan teollisuusroboteille hyödyntäen Visual Components Premium OLP -simulointi- ja etäohjelmointiohjelmistoa. Kahdessa ensimmäisessä esimerkissä (Case 1 ja 2) luodaan digitaaliset varjot eri robottisoluista, kolmannessa esimerkissä (Case 3) osittainen digitaalinen kaksonen ja viimeisessä esimerkissä (Case 4) digitaalisille kaksosille vaihtoehtoinen toteuttamistapa etäohjaukselle ja -monitoroinnille.

CASE 1: Turvamattokokoonpano

OPLITE-hankkeella toteutetusta ihmisen ja robotin yhteistyön turvamattokokoonpano-sovelluksesta kehitettiin digitaalinen varjo. Tavoitteena oli muodostaa reaaliaikainen yhteys fyysisen robotin ja sen virtuaalisen vastineen välille siten, että reaalimaailman robotin liikkeitä voidaan seurata tarkasti virtuaalisessa ympäristössä.

Taulukko 1. *Turvamattokokoonpanosovelluksen tekniset tiedot*

Sovelluksen kuvaus	*Ihmisen ja robotin yhteistyö turvamattokokoonpanossa*
Robotti	*ABB IRB 4600-40/2.55*
Oheislaitteet	*Lineaarirata, alipainetarttuja, turva- ja ohjauslaitteet*
Digitaalisen kaksosen tyyppi	*Digitaalinen varjo* *Toteutettu Visual Components -ohjelmistolla*

Turvamatossa on erilaisia komponentteja: isoja pohja-, väli- ja päällyslevyjä, herkkiä elektroniikkakomponentteja, taipuisia sähköjohtoja ja kiinnitysruuveja. Tuotteen kokoonpanoa on vaikea täysin automatisoida, koska tuotteessa on useampia johdotuksia, jotka vaativat pujottamista ja täten operaattoria näiden työvaiheiden suorittamiseksi. Tuotteen kokoonpano on toisaalta hyvin raskas operaattorille, koska tuotteen isoja levyjä joudutaan kääntelemään johtojen asennuksen yhteydessä, jolloin täysin manuaalinen kokoonpano ei ole optimaalisin vaihtoehto. Tämän takia ihmisen ja robotin yhteistyö on välttämätöntä turvamaton kokoonpanemiseksi: robottia tarvitaan raskaiden kokoonpanovaiheiden toteuttamiseen ja ihmistä sorminäppäryyttä tarvitsevien kokoonpanovaiheiden suorittamiseen.

Kokoonpanojärjestelmä on rakennettu lineaariradan päälle asennetun ABB IRB 4600 -robotin ympärille (kts. kuva 1). Robotille on suunniteltu kustomoitu alipainetarttuja isojen pohja-, väli- ja päällyslevyjä asentamiseen. Robotti hakee kyseiset levyt robotin takana olevasta hyllyköstä. Ihminen asentaa muut elektroniikkakomponentit ja johdotukset joko asennuspöydällä tai vaihtoehtoisesti suoraan pohjalevyyn robotin pitäessä pohjalevyä ilmassa paikoillaan ihmisen suorittaessa samanaikaisesti tarvittavat kytkennät. Järjestelmässä on erillinen käyttöliittymä (ohjauskotelo) ihmisen ja robotin vuorovaikutukseen, jossa on painikkeet robotin pysäyttämiseksi sekä robotin liikkeen sallimiseksi. Järjestelmään on kytkettynä erilaisia turvalaitteita, kuten valoverho ja oven turvakytkin, joilla estetään ulkopuolisten saapuminen työskentelyalueelle.

Ihmisen ja robotin yhteistyön käytännön toteutuksesta turvamattokokoonpanossa kerrotaan tarkemmin OPLITE-hankkeelle toteutetussa erillisessä selvityksessä (*Teknologiademo: Ihmisen ja robotin yhteistyö turvamattokokoonpanossa*). Tässä raportissa keskitytään jatkossa pelkästään digitaalisen varjon toteutukseen, jolloin itse kokoonpanosovellusta on kuvattu vain tarpeellisten kokonaisuuksien osalta.

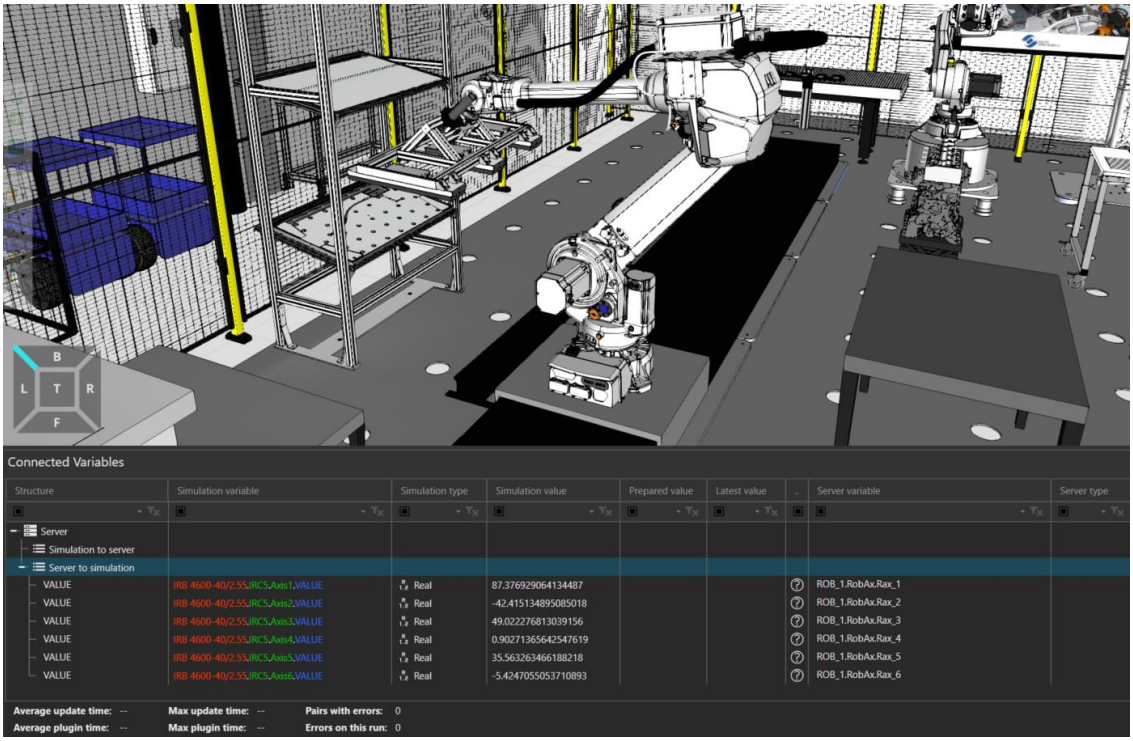


Kuva 1. *Turvamattokokoonpanosolu*

Turvamattokokoonpanosolusta mallinnettiin Visual Components Premium OLP -simulointi- ja etäohjelmointiohjelmistolla kolmiulotteinen virtuaalinen vastine digitaalisen varjon luomiseksi. Digitaalisen varjon vaatiman reaaliaikaisen ja automatisoidun tiedonsiirron muodostamiseksi reaalimaailman ja virtuaalisen vastineen välille käytettiin Visual Componentsin Connectivity-lisäosaa, joka mahdollistaa reaalimaailman robotin liikkeiden visualisoimisen virtuaalimallissa lähes reaaliaikaisesti. Connectivity-lisäosa tukee ainoastaan valittujen robottivalmistajien reaalimaailman robottien yhdistämisen Visual Componentsiin, kuten esimerkiksi ABB, Fanuc ja Universal Robots.

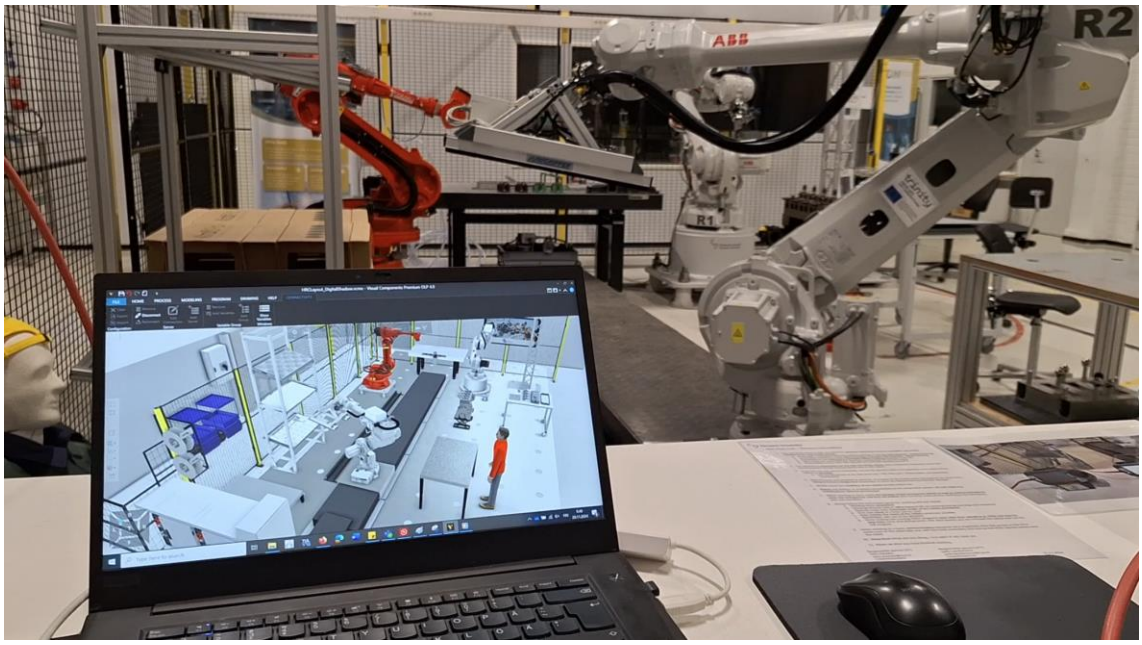
Käytännössä reaalimaailman robotin yhdistäminen digitaaliseen vastineeseen tarkoittaa, että ensin kytketään fyysisen robotin ohjain Ethernet-kaapelilla tietokoneeseen, jossa Visual Components on käynnissä. Tämän jälkeen Connectivity-lisäosassa valitaan robottivalmistaja, jonka robotille luodaan Visual Componentsiin serveri, jolla Visual Componentsin virtuaalisen robotin ohjain saadaan yhdistettyä fyysisen robotin ohjaimeen. Näiden kahden varsinainen yhdistäminen toteutetaan Connectivity-lisäosalla fyysisen robotin URL-osoitteen syöttämisellä luotuun serveriin.

Yhteyden muodostamisen ja testaamisen jälkeen fyysisen robotin jokaisen nivelen nivelkulman signaali yhdistetään näiden vastinsignaaleihin Visual Componentsin virtuaaliselle robotille (kts. kuva 2). Reaalimaailman robotin liikkeiden visualisoimisen ohella Connectivity-lisäosalla voitaisiin yhdistää myös muita fyysisen robottisolun komponentteja ja näiden toimintojen visualisoiminen tämän virtuaaliseen vastineeseen. Esimerkiksi robotin tarttujien, kuljettimien ja turvalaitteiden toiminnallisuuksia voidaan visualisoida, mikäli näitä ohjataan robotilla. Turvamattokokoonpanossa on ainoastaan yksinkertaisuuden vuoksi yhdistetty robotin nivelkulmien signaalit keskenään, mutta myös robotin alipainetarttujan sekä turva- ja ohjauslaitteiden signaalit voitaisiin yhdistää simuloituun järjestelmään, koska nämä on yhdistetty fyysiseen robottiin.



Kuva 2. *Turvamattokokoonpanosolun virtuaalinen vastine ja robotin nivelkulmien signaalien yhdistäminen fyysisen robotin signaaleihin*

Luodun digitaalisen varjon käyttäminen on hyvin yksinkertaista yhteyden luomisen jälkeen. Visual Componentista tarvitsee ainoastaan käynnistää simulointi, jonka jälkeen fyysisen robotin työohjelma voidaan käynnistää reaalimaailmassa. Tämän jälkeen reaalimaailman robotin liikkeitä voidaan visualisoida melkein reaaliaikaisesti virtuaalisessa vastineessa. Viiveen suuruus riippuu monesta eri tekijästä, mutta tyypillisesti se on 100ms-500ms välillä ilman suurempia ongelmia tiedonsiirrossa. Reaalimaailman robotin liikkeiden monitoroimisen ohella Visual Componentissa voidaan luoda statistiikkatyökaluilla erilaisia kaavioita robottisolun toiminnasta, esimerkiksi robotin käyttöasteista ja tuotantomääristä. Turvamattokokoonpanolle luotu digitaalinen varjo on esitetty kuvassa 3.



Kuva 3. *Turvamattokokoonpanosolun digitaalinen kaksonen*

On kuitenkin syytä muistaa, että Connectivity-lisäosalla muodostettu yhteys on kuitenkin ainoastaan yksisuuntainen: fyysisen robotin liikkeet heijastuvat digitaaliseen malliin, mutta ei päinvastoin eli Visual Componentsilla ei pysty ohjaamaan fyysistä robottia. Tästä syystä kyseessä on digitaalinen varjo, ei varsinainen digitaalinen kaksonen. Lisäosalla muodostettu yhteys on kuitenkin helppokäyttöinen ja kustannuksellisesti edullinen ratkaisua, mikäli tavoitteena on ainoastaan visualisoida ja monitoroida reaalijärjestelmän toimintaa simuloidussa ympäristössä.

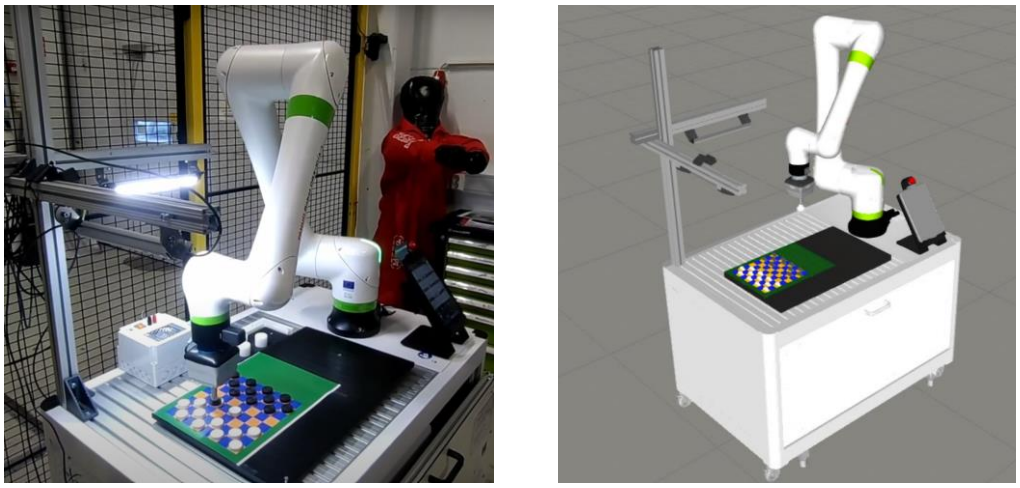
CASE 2: Konenäkösovellus

Opiskelijaprojektissa kehitettiin digitaalinen varjo yhteistyörobottisolusta, jossa robotti pelaa tammea operaattoria vastaan. Projektin keskeisenä tavoitteena oli muodostaa reaaliaikainen yhteys fyysisen robotin ja sen digitaalisen vastineen välille siten, että reaalimaailman robotin liikkeitä voidaan visualisoida tarkasti virtuaalisessa ympäristössä.

Taulukko 2. *Konenäkösovelluksen tekniset tiedot*

Sovelluksen kuvaus	*Konenäöllä varustettu yhteistyörobotti pelaamassa tammea operaattoria vastaan*
Robotti	*Fanuc CRX -koulutusrobottisolu*
Oheislaitteet	*Imukuppitarttuja, webkamera ja valaistus*
Digitaalisen kaksosen tyyppi	*Digitaalinen varjo* *Toteutettu Visual Components ja OpenCV -ohjelmistoilla*

Järjestelmä on rakennettu Fanuc CRX -koulutusrobottisolun ympärille (kts. kuva 4). Fanucin käsivarsirobotti on varustettu OnRobotin imukuppitarttujalla pelinappuloiden siirtelemiseksi. Pelinappuloiden tunnistamista varten pelialusta on valaistu kahdella suunnattavalla valolla ja pelilaudan yläpuolelle asennetulla webkameralla seurataan pelin kulkua pelinappuloiden sijaintia seuraamalla. Konenäkösovellus on toteutettu OpenCV:n (engl. Open Source Computer Vision Library) avoimen lähdekoodin kirjastolla. Pelipöydästä otetut kuvat analysoidaan OpenCV:n avulla ja erillisellä Python-skriptillä pelinappuloiden sijaintitiedot siirrettiin robottisolun digitaaliseen vastineeseen, joka on toteutettu Visual Components Premium OLP -simulointi- ja etäohjelmointiohjelmistolla.



Kuva 4. *Konenäkösovelluksen digitaalinen varjo*

Simulointiympäristön ja fyysisen robotin välinen reaaliaikainen yhteys toteutettiin Visual Componentsin Connectivity -lisäosalla, joka mahdollistaa todellisen robotin liikkeiden visualisoimisen virtuaalimallissa lähes reaaliaikaisesti kuten Case 1:ssä esiteltiin. Käytännössä tämä tarkoittaa sitä, että fyysisen robotin jokaisen nivelen nivelkulman signaali yhdistetään näiden vastinsignaaleihin Visual Componentsin virtuaaliselle robotille. Yhteys on kuitenkin ainoastaan yksisuuntainen: fyysisen robotin liikkeet heijastuvat digitaalimalliin, mutta ei päinvastoin. Tästä syystä kyseessä on digitaalinen varjo, ei varsinainen digitaalinen kaksonen.

Lisätietoa sovelluksesta löytyy oheisesta linkistä:

- https://www.youtube.com/watch?v=Pz1iB5tfgLs

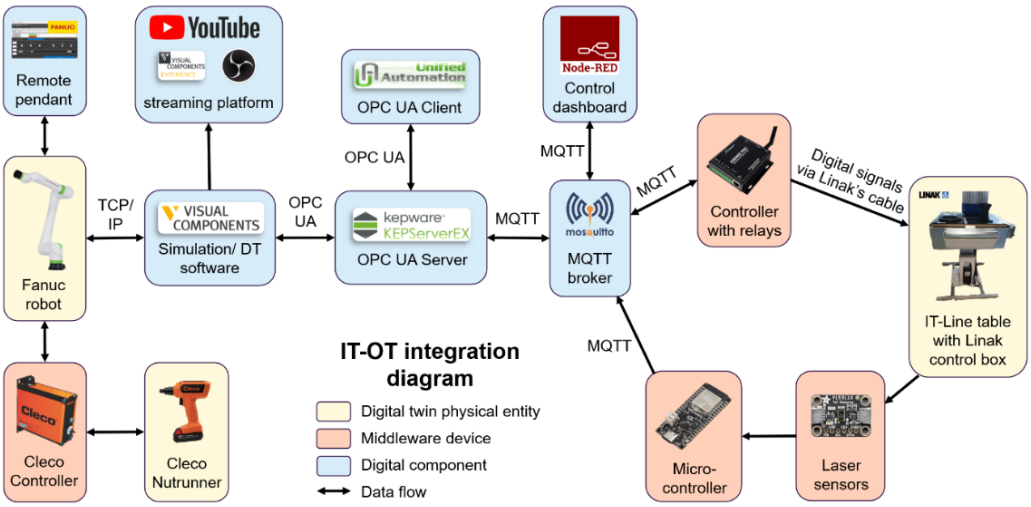
CASE 3: Ruuvaussovellus

Techboost-hankkeella toteutetussa diplomityössä toteutettiin osittainen digitaalinen kaksonen ruuvaussovelluksessa käytettävästä teollisuusrobottisolusta (kts. kuva 5). Diplomityön perimmäisenä tavoitteena oli osoittaa tapaustutkimuksen muodossa, mitä kaikkea vaaditaan teollisen sovelluksen digitaalisen kaksosen luomiseksi pk-yrityksissä. Tyypillisesti digitaalisten kaksosten toteutukset jäävät fyysisen järjestelmän peilaamiseen virtuaalisessa järjestelmässä eikä reaaliaikaista kytkentää ole virtuaalisesta järjestelmästä reaalijärjestelmään. Yhtenä haasteena on teollisuudessa käytössä oleva vanha ja monimuotoinen laitekanta, joka vaikeuttaa eri järjestelmien integroimista keskenään, sekä turvallisuusnäkökulmat. Tässä tutkimuksessa edellä olevaa ongelmaa lähdettiin ratkaisemaan.



Kuva 5. *Ruuvaussovelluksen digitaalinen kaksonen (vasemmalla järjestelmän virtuaalinen vastine Visual Componentissa, oikealla fyysinen järjestelmä)*

Ruuvaussovelluksessa on kolme pääkomponenttia: FANUC CRX -yhteistyörobotti, Cleco-ruuvinväännin erillisellä mPro200GC-ohjaimella, sekä IT-Line korkeussäädettävä kokoonpanopöytä LINAK-ohjaimella. Tämän lisäksi tarvitaan erilaisia välilaitteita (esimerkiksi ohjaimia, mikrokontrollereita ja antureita), jotta pystyttäisiin muodostamaan yhteys korkeussäädettävän kokoonpanopöydän ja tämän virtuaalisen vastineen välille. Kokoonpanopöytää ei voida integrointi virtuaaliseen vastineeseen ilman lisälaitteita kokoonpanopöydässä käytettävän vanhan teknologian ja eri tiedonsiirtoväylien takia. Ruuvaussovelluksen virtuaalinen vastine on toteutettu Visual Components Premium OLP -ohjelmistolla. Eri laitteiden integrointi toisiinsa havainnollistetaan kuvassa 6.

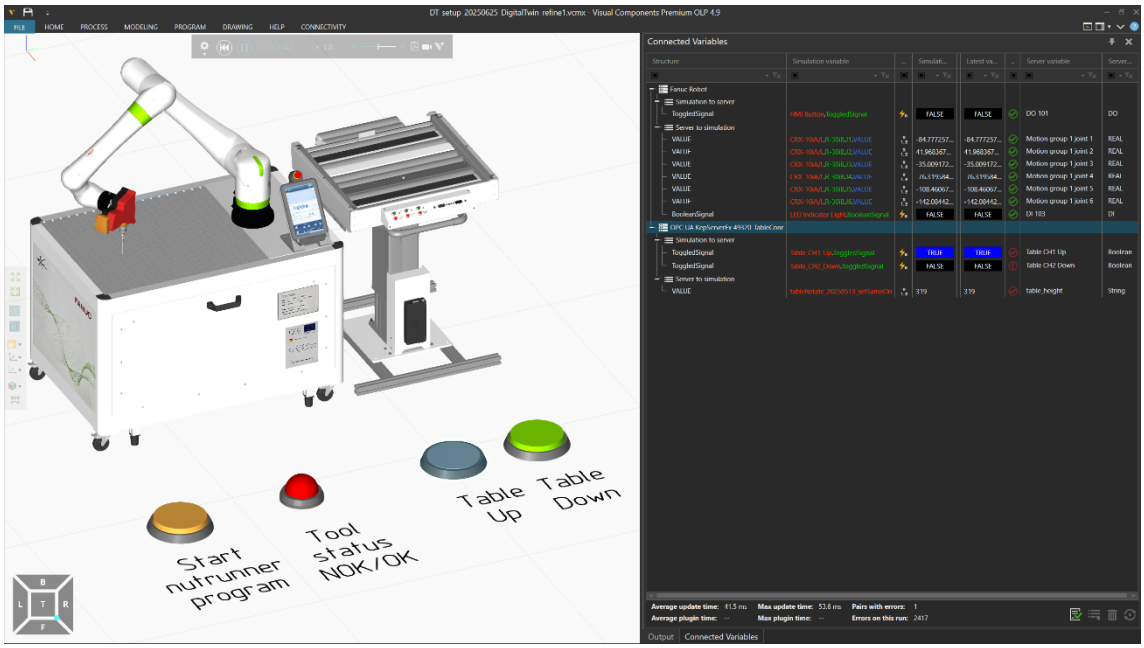


Kuva 6. *Ruuvaussovelluksen integrointikaavio*

Cleco-ruuvinväännin yhdistyy suoraan mPro200GC-ohjaimen kautta Fanucin yhteistyörobotiin I/O-signaalien kautta. Visual Componentsin Connectivity-lisäosan avulla fyysisen robotin I/O-signaalit sekä robotin nivelten nivelkulmat saadaan yhdistettyä reaalijärjestelmän virtuaaliseen vastineeseen Visual Componentsissa. Kytkentä on kuitenkin ainoastaan yksisuuntainen robotin liikeohjauksen osalta turvallisuussyistä eli Visual Componentsilla ei voida ohjata fyysistä robottia. Tämän kiertämiseksi diplomityössä hyödynnettiin Fanucin iRProgrammer-ohjelmistoa, mikä mahdollisti robotin etäohjauksen samalta koneelta, jolla Visual Components on käynnissä.

Korkeussäädettävä kokoonpanopöytä käyttää LINAK-ohjainta, jolla ei ole teollisuusprotokollaa, mikä vaikeuttaa digitaalisen kaksosen rakentamista. Jotta kokoonpanopöytä voitaisiin yhdistää virtuaaliseen vastineeseen Visual Components lähettää kokoonpanopöydän ohjauskomennot OPC UA -serverille, joka muuntaa ne MQTT-viesteiksi. MQTT-palvelin välittää ohjauskomennot edelleen erillisellä ohjaimella, joka on kytketty LINAK-ohjaimeen I/O:n kautta. Kokoonpanopöydän asematietoa (eli tässä tapauksessa korkeustietoa) varten lasermittausanturi mittaa millä korkeudella kokoonpanopöytä on ja siirtää tiedot takaisin Visual Componentsille käänteisessä järjestyksessä (MQTT → OPC UA → Visual Components).

Digitaalisen kaksosen rakentamista ja ylläpitoa varten toteutettiin erillisiä lisätoimintoja. Node-RED:lla luotiin erillinen selainpohjainen käyttöliittymä kokoonpanopöydän ohjaamiseen ilman Visual Componentsia, mikä osoittautui hyödylliseksi digitaalisen kaksosen käyttöönotossa. UAExpert ja MQTT Explorer tukevat puolestaan vianetsintää järjestelmästä. Visual Components Experience ja OBS (engl. Open Broadcaster Software) -ohjelmistot mahdollistavat digitaalisen kaksosen virtuaalisen vastineen tilan jakamisen ja striimauksen esimerkiksi YouTubessa.



Kuva 7. *Ruuvaussovelluksen virtuaalinen vastine käyttöliittymällä*

Kuten edellä jo mainittiin, fyysisen robotin ohjaaminen virtuaalisen mallilla ei ollut mahdollista turvallisuussyistä takia. Kuitenkin ruuvinvääntimen ja korkeussäädettävän kokoonpanopöydän ohjaaminen onnistuu virtuaalisen mallin kautta sinne luotujen painikkeiden avulla (kts. kuva 7). Visual Componentsin kautta on mahdollista säätää pöydän korkeutta ylöspäin tai alaspäin sekä käynnistää ruuvinvääntimen työohjelma ja näyttää ruuvaimen tilatieto (status).

Diplomityön tuloksena saatiin toteutettua osittainen digitaalinen kaksonen, jossa pystyttään heijastamaan fyysisen järjestelmän tila kokonaisuudessaan virtuaalisen järjestelmään. Virtuaalisella järjestelmällä voidaan kuitenkin ainoastaan ohjata reaalimaailman ruuvinväännintä ja kokoonpanopöydän korkeutta, mutta ei suoraan fyysistä robottia turvallisuussyiden vuoksi. Tutkimuksessa huomattiin, että digitaalisen kaksosen rakentaminen voi olla haastavaa, kun erilaisia laitteita ollaan integroimassa keskenään. Erityisen haasteen asettavat vanha laitekanta ja vaihtelevat tiedonsiirtoprotokollat. Tällöin voi olla tarvetta lisälaitteille ja -sovelluksille, jotta integrointi voidaan toteuttaa reaalijärjestelmän ja tämän virtuaalisen vastineen välillä.

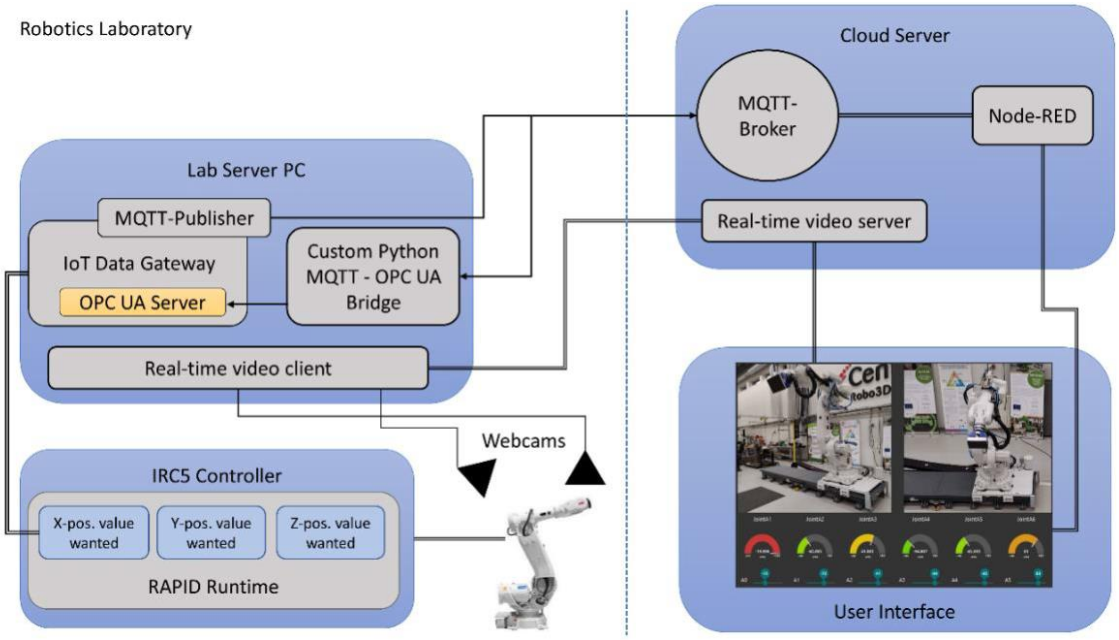
Lisätietoa sovelluksesta löytyy oheisesta linkistä:

- Phan L. Digital Twin In Discrete Manufacturing: Benefits, Challenges, And Case Study. Tampere University; 2025. Saatavilla: https://trepo.tuni.fi/handle/10024/229544

CASE 4: Teollisuusrobotin etäohjaus

Tampereen yliopisto toteutti Centria-ammattikorkeakoulun kanssa yhteistyönä teollisuusrobottien etäohjausalustan yliopiston *Raskaan yhteistyörobotiikan pilottilinjalle* (engl. *Human-Robot Collaboration Pilot Line, HRC Pilot Line*), joka on yliopiston tutkimus- ja opetuslaboratorio. Etäohjausalustan verkkopohjainen käyttöliittymä mahdollistaa teollisuusrobottien helpon ja yksinkertaisen etäohjauksen erilaisilla liukusäätimillä ja painikkeilla sekä etämonitoroinnin webkameroilla, jotka välittävät melkein reaaliaikaista kuvaa robottisolusta etäohjausalustalle. Alustalla voidaan esimerkiksi suorittaa yksinkertaisia Pick&Place -tehtäviä etäohjauksella ilman operaattorin fyysistä läsnäoloa laboratoriossa. Kehitetty sovellus on vielä tutkimuksen tasolla, mutta tulevaisuudessa teknologioiden kehittyessä mahdollisesti sovellettavissa myös valmistavassa teollisuudessa tarjoten mahdollisuuden reaaliaikaiseen etämonitorointiin ja -ohjaukseen.

Etäohjausalustan tekninen toteutus pohjautuu Tero Kaarlelan (2023) väitöskirjaan: *A common digital twin platform for education, training, and collaboration in robotics*. Tavoitteena oli skaalata Kaarlelan kehittämä ratkaisu raskaille teollisuusroboteille. Etäohjausalustan teknisen toteutuksen periaate on esitetty kuvassa 8.



Kuva 8. *Etäohjausalustan teknisen toteutuksen periaate (Pöysäri et al. 2023)*

Raskaan yhteistyörobotiikan pilottilinjalla on käytössä lineaariradalle asennettu ABB IRB 4600 sarjanrobotti, johon voi olla asennettuna joko sormitarttuja tai alipainetarttuja. Robotin etäohjaamista varten tarvitaan käyttäjältä ohjaussignaali tarttujalle (sormien tai alipaineen avaaminen/sulkeminen) sekä riippuen robotin ohjaustavasta joko robotin yksittäisten nivelten ohjausarvot tai vaihtoehtoisesti robotin työkalupisteen X-, Y- ja Z-koordinaatit robotin ja tämän työkalun liikuttamiseksi lineaarisesti robotin peruskoordinaatistossa. Robotin IRC5 -ohjaimelle luotu RAPID-työohjelma vastaanottaa ohjaussignaalit ja -arvot etäohjausalustalta erilaisten sovellusten ja tiedonsiirtoväylien kautta ja suorittaa tämän perusteella robotin työohjelmaa. Näin pystytään liikuttamaan joko robotin yksittäisiä niveliä tai työkalua lineaariliikkeillä sekä operoimaan robotin tarttujaa.

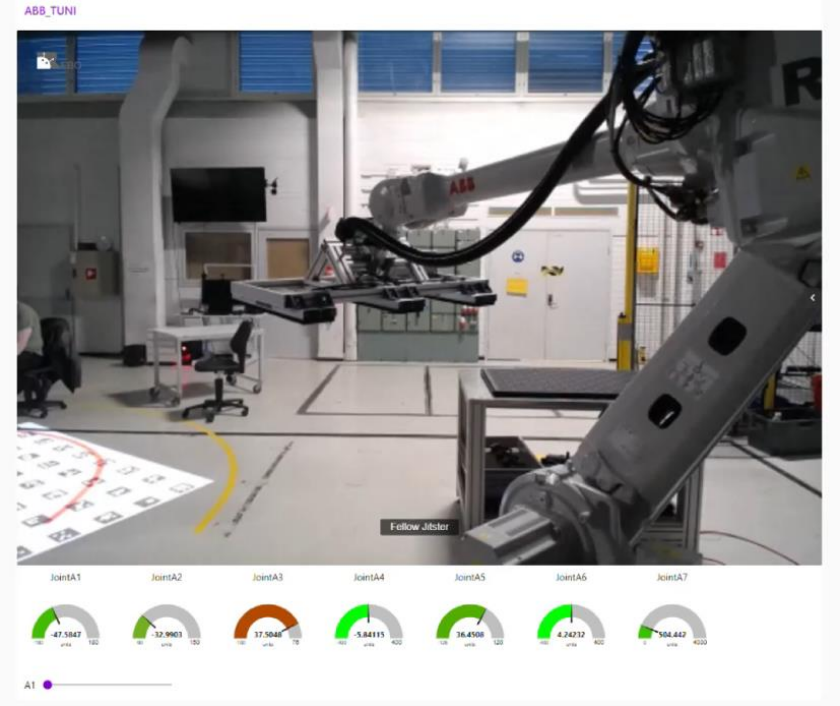
Turvallisuus on keskeinen huolenaihe etäohjauksessa, koska robotit eivät ole luonnostaan turvallisia. Riskinarvioinnin perusteella etäohjauksen suurin riski on, jos joku päätyy robotin työskentelyalueelle etäohjauksen aikana. Raskaan yhteistyörobotiikan pilottilinjalla turvallisuudesta on huolehdittu fyysisten turva-aitojen lisäksi käytössä olevilla valoverhoilla, ovikytkimillä ja turvaskannereilla, jotka varmistavat turvallisen työskentelytilan. Nämä laitteet tunnistavat, mikäli joku tulee robotin työskentelyalueelle ja pysäyttävät robotin. Käytössä on myös valomajakoita, jotka ilmaisevat järjestelmän tilan laboratoriossa työskenteleville henkilöille. Turvalaitteiden lisäksi robotilla on käytössä ABB:n SafeMove-lisäoptio, jolla ohjelmallisesti rajataan robotin liikealuetta törmäysten välttämiseksi sekä rajoitetaan robotin liikenopeuksia.

Laboratoriossa olevilla USB-pohjaisilla webkameroilla pystytään monitoroimaan lähes reaaliaikaisesti robottisolua ja robotin liikkeitä etäohjausalustan käyttöliittymän kautta. Olennaista on käyttää riittävä määrä hyvin sijoiteltuja kameroita, jotta

a) robotin työskentelyalue nähtäisiin riittävän laajalta alueelta
b) robotin työskentelyalueelle ei jäisi katvealueita
c) robotin varsinaista tehtävää, esimerkiksi kappaleen poimintaa pöydältä, kuvataan riittävän läheltä eri kuvakulmista

Käytettävien webkameroiden tulee olla korkearesoluutioisia (ja mielellään automaattisesti tarkentavia) USB-kameroita, joilla saadaan aikaiseksi hyvä kuvanlaatu verkkopohjaiseen käyttöliittymään sekä mahdollisimman pienet viiveet kuvayhteyksissä (noin 10-20 ms), mikä on ensiarvoista etäohjauksessa esimerkiksi törmäyksien välttämiseksi, kun robottia ohjataan käyttöliittymässä videokuvan perusteella. Lisäksi täytyy kiinnittää huomiota kameroiden sijoitteluun laboratoriossa, jotta vältytään pitkiltä tiedonsiirtokaapeleiden vedoilta (maks. 5 m), jotka lisäävät latenssiviiveitä ja aiheuttavat yhteysongelmia.

Mikäli tarvitaan useampia kameroita eri puolella laboratoriota, nämä kannattaa kytkeä eri tietokoneisiin yhden tietokoneen sijasta, koska etäohjausalustaan voi liittyä useammalta tietokoneelta ja selaimelta samanaikaisesti. Tässä sovelluksessa käytettiin kahta laboratorion etu- ja takaseinille sijoiteltua kameraa, jotka valvoivat koko laboratoriota, sekä kahta kokoonpanopöydille sijoiteltua kameraa, jotka kuvaavat kokoonpanopöytiä, joilla työkappaleita käsitellään.



Kuva 9. *Verkkoselaimen näkymä etäohjausalustan kehitysversiosta, jossa yläpuolella reaaliaikainen kuvayhteys robotilta ja etäohjauksen käyttöliittymä alapuolella*

Robotin, robotin ohjaimen, turvalaitteiden ja webkameroiden lisäksi laboratoriossa on palvelintietokone (engl. Lab server PC, kuvassa 8), jolla muodostetaan yhteys Centrian toteuttamaan pilvipalvelimeen, joka toimii laboratorion laitteiden ja etäohjauksen käyttöliittymän välisenä linkkinä. Tässä selvityksessä ei syvennytä pilvipalvelimen toteutukseen, koska tätä hyödynnettiin tässä demonstraatiossa sellaisenaan ilman, että siihen tehtiin suurempia muutoksia. Pilvipalvelimen tarkempi toteutus on luettavissa edellä mainitussa Kaarlelan (2023) väitöskirjasta.

Palvelintietokone on kytkettynä Ethernet-yhteydellä robotin ohjaimeen. Jotta robotin ohjaimelta voidaan lukea ja sille kirjoittaa tietoa, kuten esimerkiksi robotin nivelten nivelkulmat, palvelintietokoneelle täytyy asentaa ABB:n IoT Data Gateway -ohjelmisto, jolla mahdollistetaan robotin liitettävyys muihin järjestelmiin. Kyseisellä ohjelmistolla pystytään välittämään esimerkiksi edellä mainitut robotin nivelten nivelkulmat eteenpäin

MQTT-protokollalla etäohjausalustan käyttöliittymään, mutta vastaavia tietoja ei pystytä vastaanottamaan käyttöliittymästä MQTT-protokollalla vaan ainoastaan OPC UA-protokollalla johtuen IoT Data Gateway -ohjelmiston rajoitteista. Näin ollen kehitettiin kustomoitu MQTT/OPC UA -silta Pythonilla, joka mahdollistaa ohjausarvojen lähettämisen etäohjausalustan käyttöliittymältä robotille.

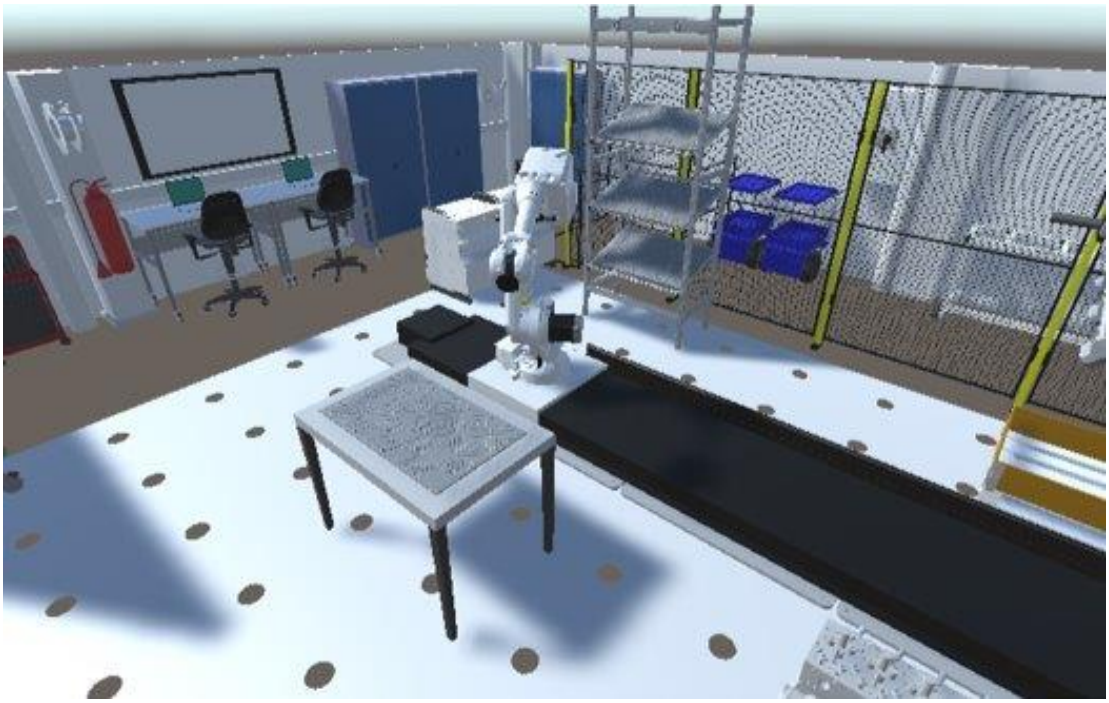
Robotin työohjelmaa (RAPID-ohjelmaa) suoritetaan robotin ohjaimella jatkuvassa silmukassa. Ohjelma lukee käyttöliittymästä lähetetyt robotin nivelten nivelkulmat tai robotin työkalupisteen X-, Y- ja Z-koordinaatit ja tämän perusteella ohjataan joko robotin yksittäisiä niveliä tai siirretään robotin työkalua lineaariliikkeillä haluttuun asentoon. Samaan tapaan ohjelma lukee robotin tarttujan ohjausarvon ja avaa/sulkee sormitarttujan sormet tai laittaa alipainetarttujan imun päälle tai pois.

Etäohjausalustan käyttöliittymä toimii verkkoselaimella (kts. kuva 9) ja se toteutettiin Node-RED kehitystyökalulla. Käyttöliittymä näyttää lähes reaaliaikaista videokuvaa robottisolusta, esittää robotin yksittäisten nivelten nivelkulmat sekä robotin työkalupisteen sijainnin robotin peruskoordinaatistossa X-, Y- ja Z-koordinaateilla. Robotin nivelkulmat esitetään käyttöliittymässä mittarielementeillä, jolloin käyttäjä voi helpommin seurata nivelten asentoja ja havaita, ovatko ne lähestymässä äärirajojaan. Käyttöliittymässä on painikkeet robotin tarttujan ohjaamiseen (avaaminen/sulkeminen) sekä liukusäätimet robotin yksittäisten nivelten ohjaamiseksi sekä työkalupisteen lineaariliikkeiden hallintaan X-, Y- ja Z-koordinaateilla robotin peruskoordinaatistossa. Mahdollisen väärinkäytön estämiseksi työkalupisteen ohjaamiseksi on käytössä erillinen TCP-ohjausvalitsin, jolla estetään robotin yksittäisten nivelten ohjaaminen silloin, kun liikutetaan robotin työkalupistettä lineaariliikkeillä. Sama toimii toisinpäin eli kun TCP-ohjausvalitsin ei ole aktivoituna, voidaan ainoastaan ohjata robotin yksittäisiä niveliä. Lopullisen käyttöliittymän mittareita ja painikkeita on havainnollistettu kuvassa 10.



Kuva 10. *Lopullisen käyttöliittymän painikkeita, säätimiä ja mittareita*

Käyttöliittymään pystytään yhdistämään reaalijärjestelmän lisäksi robottisolun simulointimalli ja ohjaamaan sitä samalla tavalla käyttöliittymän avulla. Tämän toteuttamiseksi robottisolu mallinnettiin ensin Visual Componentsilla ja siirrettiin UNITY-ohjelmistoon, jossa robotille määriteltiin esimerkiksi nivelrajoitukset Articulation Body -ominaisuuksilla. Virtuaalisen robotin ohjaamiseksi kirjoitettiin kaksi erillistä skriptiä robotin nivelarvojen lukemiseen ja muuttamiseen simuloinnissa. Simulaatio julkaistiin WebGL-alustalla ja integroitiin Node-RED:iin verkkokehyksenä muiden käyttöliittymäelementtien rinnalle. Halutessaan simulointimalli pystytään yhdistämään reaalimaailman robottiin, jolloin kyseessä olisi digitaalinen varjo. Robottisolun virtuaalinen vastine on havainnollistettu kuvassa 11.



Kuva 11. *Robottisolun virtuaalinen vastine Unity-ohjelmistossa*

Kehitetty etäohjausalusta mahdollistaa helpon ja yksinkertaisen tavan teollisuusrobottien etämonitorointiin ja -ohjaukseen, kun järjestelmä on kerran saatu rakennettua. Ratkaisu mahdollistaa robotin etäohjauksen liukusäätimillä ja painikkeilla varustetulla verkkopohjaisella käyttöliittymällä, jossa on reaaliaikainen videoyhteys fyysiseen robotiin. Etäohjaus on riippumaton robottimerkkikohtaisista ohjaus- ja ohjelmointikäytännöistä tehden etäohjauksesta helposti lähestyttävän. Käyttöliittymään voidaan myös yhdistää robottisolun simulointimalli ja ohjata tätä reaalimaailman robotin arvoilla luoden digitaalisen varjon.

Etäohjausalustan nykyiset sovellusmahdollisuudet valmistavassa teollisuudessa ovat osin rajalliset johtuen turvallisuustekijöistä sekä teknisistä rajoitteista. Videoyhteydessä on aina latenssiviiveitä eikä etäohjausalusta mahdollista tällä hetkellä esimerkiksi robottien varsinaista etäohjelmointia, vaan ainoastaan rajallisen etäohjauksen. Etäohjaus on tällä hetkellä myös melko kömpelöä ja epätarkkaa, koska robottia ohjataan videokuvan perusteella rajallisin ohjauskeinoin, jolloin tarkan asennon saavuttaminen on haastavaa. Suurimpana ongelmana on kuitenkin mahdollisten ongelmatilanteiden ratkaiseminen etäohjauksella: mitä tehdään jos robotti saavuttaa liikealueensa rajat, ajautuu singulariteettiin, törmää tai tiputtaa käsiteltävän työkappaleen. Miten näistä ongelmatilanteista toivutaan ja palaudutaan normaaliin toimintaan ilman, että jonkun täytyy käydä fyysisellä robotilla korjaamassa ongelma? Näiden ongelmien ratkaisemiseksi etäohjausalustaa tulee kehittää ennen kuin siitä saataisiin tulevaisuuden mahdollinen tuotannollinen ratkaisu.

Lisätietoa sovelluksesta löytyy oheisista linkeistä:

- Pöysäri S, Kaarlela T, Dianatfar M, Lanz M. Educational Teleoperation Platform for Heavy Industrial Robotics as a Learning Environment. 2023. Saatavilla: https://trepo.tuni.fi/handle/10024/205547
- Kaarlela T. A Common Digital Twin Platform for Education, Training and Collaboration. Tampere University; 2023. Saatavilla: https://trepo.tuni.fi/handle/10024/149112

YHTEENVETO

Case-esimerkeistä 1 ja 2 nähdään, että pelkkien digitaalisten varjojen luominen Visual Components Premium OLP -ohjelmiston Connectivity-lisäosalla onnistuu helposti ja vaivattomasti, mikäli vain robottimerkkiä tuotetaan ko. lisäosassa. Valitettavasti tällä hetkellä lisäosa tukee ainoastaan vain valittuja robottimerkkejä, jolloin digitaalisia varjoja ei tällä tavalla pystytä luomaan suurimmalle osalle eri robottivalmistajista. On myös tärkeä huomioida, että pelkän yhteyden luominen reaalimaailman ja tämän virtuaalisen vastineen välillä ei takaa onnistunutta digitaalista varjoa. Tarvitaan huolellisesti mallinnettu virtuaalimalli robottisolusta, joka vastaa todellisuutta ja jossa on luotuna tarvittavat vastinsignaalit reaalimaailman robottisoluun.

Aitojen digitaalisten kaksosten luominen on puolestaan haastavaa, kuten Case-esimerkistä 3 havaitaan. Case 3:ssa pystyttiin muodostamaan reaaliaikainen ja kaksisuuntainen tiedonsiirtoyhteys ainoastaan korkeussäädettävälle kokoonpanopöydälle ja ruuvinvääntimelle eikä robotille turvallisuussyiden takia. Käytännössä tämä tarkoittaa sitä, ettei virtuaalisella mallilla pystytä ohjaamaan todellista robottia, mikä on edellytyksenä aidolle digitaaliselle kaksoselle. Case 3:ssa havaittiin myös, että omat haasteensa digitaalisten kaksosten luomiseen tuovat erilaiset tiedonsiirtoväyläteknologiat sekä vanha ja monipuolinen laitekanta. Kyseiset integrointihaasteet toimivat kompastuskivenä toimivien digitaalisten kaksosten luomisessa.

Case 4:ssa esiteltiin vaihtoehtoinen tapa, kuinka toteuttaa reaalimaailman robotin etäohjaus digitaalisen kaksosen sijasta. Tässäkin esimerkissä havaitaan, kuinka moniulotteinen järjestelmä tarvitaan tämän toteuttamiseksi huomioiden sen, että etäohjaus esitellyllä järjestelmällä ei ole vielä käytännöllistä eikä turvallista teollisuudessa huomioiden etäohjauksen tarkkuus, latenssiviiveet sekä alttius ongelmatilanteisiin, joita ei pystytä etänä ratkaisemaan.

Jatkoselvitettäväksi jää, kuinka luotuja digitaalisia varjoja ja digitaalisia kaksosia pystyttäisiin hyödyntämään (kustannus)tehokkaasti teollisuusrobotiikassa. Nykyisellä monitoroinnilla pystytään seuraamaan ja visualisoimaan reaalijärjestelmän toimintaa virtuaalisessa vastineessa ja saamaan tätä kautta tietoa järjestelmän tilasta ja hyödyntämään sitä robottisolun tehostamiseksi esimerkiksi robotin työohjelmien optimoimisen kautta. Kysymyksenä kuitenkin on: onko digitaalisten varjojen ja kaksosten rakentaminen kannattavaa ja onko siitä saavutettavat hyödyt kaiken rakentamisen tuoman vaivan arvoista.