

Markku Nuolimäki

## **AUTOMATISOIDUN VARASTON VIRTUAALINEN KÄYTTÖNOTTO**

# AUTOMATISOIDUN VARASTON VIRTUAALINEN KÄYTTÖNOTTO

Markku Nuolimäki  
Opinnäytetyö  
Syksy 2023  
Automaatiotekniikka  
Oulun ammattikorkeakoulu

## TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Sähkö- ja automaatiotekniikan tutkinto-ohjelma, automaatiotekniikka

---

Tekijä: Markku Nuolimäki  
Opinnäytetyön nimi: Automatisoidun varaston virtuaalinen käyttöönotto  
Työn ohjaaja(t): Manne Tervaskanto (OAMK), Jyri-Jussi Torvinen (OAMK)  
Työn valmistumislukukausi ja -vuosi: Syksy 2023  
Sivumäärä: 29

---

Työn tarkoituksena oli simuloida varastoautomaatiojärjestelmää. Työssä käytettiin Beckhoff Twin-cat 3 sekä Visual Components -ohjelmistoa. Työ tehtiin Oulun ammattikorkeakoululle, jossa järjestelmää on tarkoitus laajentaa ja käyttää opetuksessa ja tutkimuksessa. Työssä käsiteltiin ohjausjärjestelmän sekä simulaatioympäristön luomista ja liittämistä toisiinsa.

Työssä tutustuttiin PLC- ohjelmointiin sekä digitaalisiin kaksosiin ja simulointiympäristöihin. Logiikkaohjelmointi toteutettiin Beckhoff TwinCat 3 -ohjelmointiympäristöllä, joka liitettiin Visual Components simulointiympäristöön. Simulointiympäristöön tuotiin varastoautomaatiojärjestelmän 3D CAD-malli.

Tulokseksi tästä työstä saatiin ymmärrys digitaalisista kaksosista ja simuloinnista sekä PLC-ohjelmoinnista. Lisäksi huomattiin mahdollisuus liittää kumpikin järjestelmä sulavasti yhteen ja testata toimivuutta uudella tavalla.

Yhteenvetona voidaan mainita, että Visual Components on erinomainen ohjelmistokokonaisuus, joka voidaan tarvittaessa yhdistää eri alustoille ja luoda digitaalisia kaksosia tai simuloida laitteistoja ja prosesseja. Työssä saatiin toteutettua demoympäristö simuloinnille ja PLC-ohjelmalle, jota on mahdollista jatkokehittää kattamaan haluttu simulaatiokokonaisuus.

---

Asiasanat: automaatiojärjestelmä, simulointiympäristö, varastoautomaatio, TwinCat 3, Visual Components

## ABSTRACT

Oulu University of Applied Sciences  
Degree Programme in Electrical and Automation Engineering, Option of Automation Engineering

---

Author: Markku Nuolimäki

Title of thesis: Virtual deployment of the automatic storage system

Supervisor(s): Manne Tervaskanto (OAMK), Jyri-Jussi Torvinen (OAMK)

Term and year when the thesis was submitted: Autumn 2023

Number of pages: 29

---

This thesis dealt with the creation and connection of the control system and the simulation environment. The purpose of the work was to simulate an automatic storage system. On this project we used Beckhoff TwinCAT 3 and Visual Components software. This work was made to Oulu University of Applied Sciences they have interest to expand and use this with in education and research.

The work introduced PLC programming and digital twins and simulation environments. The logic programming in this work was implemented with the Beckhoff TwinCAT 3 programming environment, which was connected to the Visual Components simulation environment. A 3D CAD model of the automatic storage system was brought into the simulation environment. With that model there was possible to examine how to link two systems to a getter and how to use this kind of system.

In summary, Visual Components is an excellent software package that can, if necessary, be connected to different platforms and create digital twins or simulate hardware and processes. I was able to create a demo version of the simulation and PLC programming and if needed there is possible to continue developing those demo programs to make a full digital twin.

---

Keywords: automation system, simulation environment, automatic storage, TwinCAT 3, Visual Components

# SISÄLLYS

TEKNILLINEN SANASTO.....	6
1 JOHDANTO.....	7
2 AUTOMAATION OHJAUSJÄRJESTELMÄ.....	8
2.1 Automaattivaraston järjestelmästä .....	8
2.2 PLC- Programmable Logic Controller.....	8
2.3 Twincat 3.....	9
3 LAITTEISTON SIMULOINTI .....	12
3.1 Digitaaliset kaksoset.....	12
3.2 Simulointiympäristö-ohjelmistot.....	13
4 TYÖN TOTEUTUS .....	15
4.1 Simulointiympäristön luominen.....	15
4.2 PLC-ohjelman luominen .....	20
4.3 PLC:n ja simuloimisen linkittäminen .....	23
5 TULOKSET.....	26
6 POHDINTA.....	27
LÄHTEET.....	28

## TEKNILLINEN SANASTO

Twincat 3	Beckhoff Automation:in valmistama PC- pohjainen ohjelmointisovellus
PLC	Programmable Logic Controller
CAD	Computer Aided Design
CPU	Tietokoneen prosessori
I/O-kortti	Input/output- kortti
Input	Logiikalle sisään tuleva tieto
Output	Logiikalta lähtevä tieto
IF-lause	Ehtolause, jota käytetään ohjelmoinnissa
Python	Tekstipohjainen ohjelmointikieli
C ja C++	Tekstipohjainen ohjelmointikieli
IoT	Internet of Things

# 1 JOHDANTO

Digitaalisen kaksosen sekä simulointiympäristöjen tarve teollisuus 4.0:ssa on kasvava trendi. Suurin osa sensoritiedosta liikkuu IoT- palveluiden kautta, ja tämä tieto voidaan suoraan liittää digitaaliseen kaksoseen. Näistä saadut hyödyt vähentävät kustannuksia ja parantavat tuottavuutta, niiden avulla voidaan havaita pullonkaulat prosessissa tai parantaa niiden tehokkuutta. (1.)

Automaattivarastojen määrä on lisääntynyt 2000-luvulla, sillä niiden avulla voidaan vähentää kustannuksia sekä parantaa tehokkuutta. Nämä järjestelmät voidaan liittää digitaalisiin kaksosiin ja älykkäisiin ohjausjärjestelmiin, jolloin saadaan tehokas ja tuottoisa varastointijärjestelmä.

Tämän opinnäytetyön tarkoituksena on kartoittaa ja toteuttaa simulaatioympäristö Beckhoff:n loogikkaa käyttävälle automaatiovarastolle. Toimeksiantajana toimii Oulun ammattikorkeakoulu, jolla on fyysinen varastoautomaatiolaitteisto, jonka pohjalle simulointi luodaan. Tämän opinnäytetyön tuloksia tullaan hyödyntämään varastoautomaation käyttöönotossa, ja fyysinen laitteisto on tarkoituksena ottaa käyttöön myöhemmin syksyllä 2023.

Opinnäytetyössä perehdytään automaation PLC-ohjelmointiin sekä simulointiympäristöön. Näihin tarkoituksiin teollisuudesta löytyy useita vaihtoehtoja, jotka sopivat eri tarkoituksiin. Tähän opinnäytetyöhön on valikoitunut Beckhoffin:n Twincat 3 sekä Visual Components -ohjelma.

Työn tarkoituksena on luoda virtuaalinen simulointiympäristö ja PLC-ohjelma, jolla ohjataan varastoautomaatiota simulointiympäristössä. Simulointiympäristön avulla voidaan havaita teknologian hyötyjä ja käyttötarkoituksia. PLC-ohjelmaa puolestaan voidaan hyödyntää fyysisen laitteiston käyttöönotossa ja konfiguroinnissa.

## 2 AUTOMAATION OHJAUSJÄRJESTELMÄ

Tässä kappaleessa käydään läpi automaation ohjausjärjestelmän toimintaperiaatetta. Opinnäytetyössä keskitytään Beckhoff:n tarjoamaan TwinCAT 3 -ohjelmistokokonaisuuteen ja sen toimivuuteen.

### 2.1 Automaattivaraston järjestelmästä

Automaatiovarastojen tarve tämän päivän logistiikkatarpeissa kasvaa jatkuvasti. Ennen varastoista tavarantoimituksen tai tuomisen on kuljettanut ihminen. Älykkäällä varstoautomaatiolla voidaan vähentää työntekijöistä aiheutuvia kuluja sekä vähentää virheiden määrää varastoinnissa. Samalla esimerkiksi korkeavaraston avulla saadaan samalle lattiapinta-alalle laajempi säilytyskapasiteetti, koska hyllyjen väliin ei tarvitse jättää leveitä kulkuväyliä, jotta trukki mahtuisi noutamaan tavaraa. (2.)

Automaatiovarastoon voidaan helposti liittää myös varastohallintajärjestelmä, ja voidaan ohjata varaston ohjauslogiikkaa. Hyvällä varastohallintajärjestelmällä voidaan hallita ja ohjata tavarantoimituksia, keräilyä, materiaalien ja tuotteiden siirtelyä. (3.)

### 2.2 PLC- Programmable Logic Controller

Ohjelmoitava logiikka eli PLC on teollisuudessa paljon käytössä oleva automaatio-ohjausjärjestelmä. Ohjattavat järjestelmät voivat vaihdella kooltaan yksinkertaisista moottoriohjauksista aina kokonaisten toimilaitteista koostuvien järjestelmien ohjaukseen. Rakenteeltaan PLC koostuu seuraavista osista: CPU, IO- moduuli sekä PLC ohjelmointiohjelman. Ohjelmointi suoritetaan PC:llä, josta ohjelma siirretään PLC:hen. Tämän jälkeen PLC:n CPU suorittaa ohjelmaa. (4.)

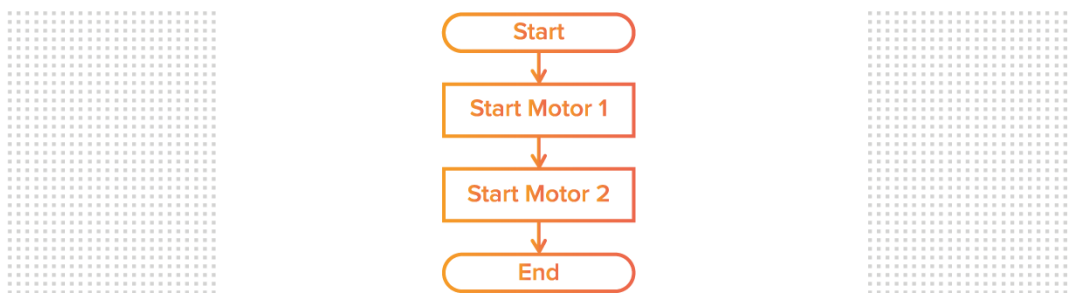
Markkinoilla on monien valmistajien malleja sekä versioita PLC-logiikoista, useaan eri käyttötarkoitukseen. Pohjimmiltaan logiikan ohjelmointi on hyvin pitkälle samanlainen, riippumatta valmistajasta. Tämän takia ohjelmointityyli voidaan jakaa kahteen kategoriaan, tekstipohjaiseen ohjelmointiin ja visuaaliseen ohjelmointiin. (4.)

Tekstipohjainen ohjelmointi pitää sisällään instruction list - sekä structured text -tyylit. Instruction list on vanhempi ohjelmointikieli, jossa ohjelmalla annetaan hyvin yksinkertaisia käskyjä. Structured text on yksi yleisimmistä ohjelmointikielistä ja se pohjautuu C ja C++ ohjelmointikieliin. Kuvassa 1 on esimerkki structured text -ohjelmoinnista, jossa käytetään IF-lausetta suorittamaan haluttu toiminto. (5.)

```
3 IF Sensor THEN
4     Motor1 := FALSE;
5     ajastin.IN := TRUE;
6 END_IF
```

KUVA 1 IF- lause Structered text Twincat 3 -ohjelmassa.

Visuaaliset ohjelmointikielät ovat ladder diagram, function block diagram sekä kuvan 2 esimerkissä sekvenssien ohjelmointiin tarkoitettu sequential function chart. Yleensä visuaaliset ohjelmointikielät ovat helpompia ymmärtää sekä oppia kuin tekstipohjaiset. Ladder-diagrammi perustuu relelogiikkaohjelmointiin ja se on suosittu käyttäjäystävällisestä ulkoasusta ja helposti ymmärrettävästä visuaalisuudesta johtuen. Function block diagram -ohjelmointi perustuu lohkoihin, jotka pitävät sisällään muuttujia. Näihin tuodaan tuloja, joista ohjelma muodostaa halutut lähtösignaalit. Sequential function chart on tarkoitettu sekvenssien ohjelmointiin, jossa seuraavaan vaiheeseen voidaan siirtyä vasta kun edellisen askeleen vaiheet ovat suoritettu ja ehdot täytetty. (5.)



KUVA 2. Sequential function chart kaaviokuva. (5.)

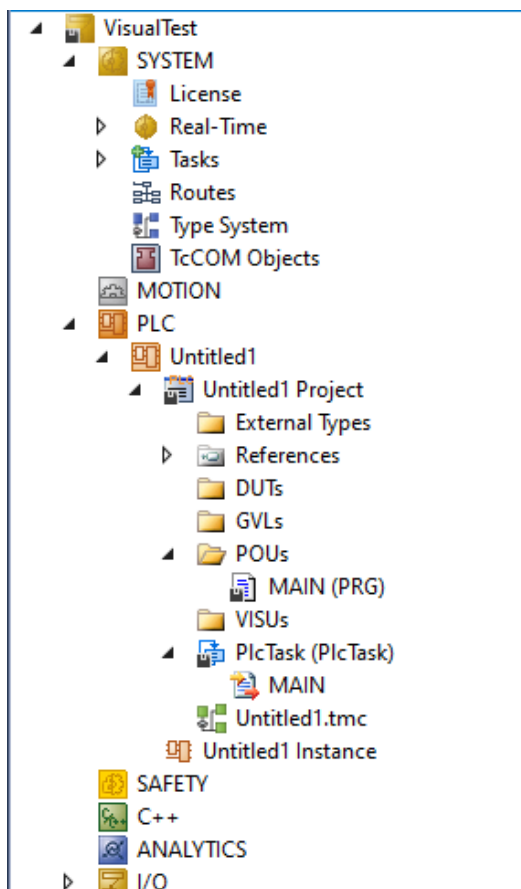
## 2.3 Twincat 3

Twincat 3 on Beckhoff:n suunnittelema ohjelmointiympäristö, joka on muokattavissa modulaarisuutensa ansiosta jokaiselle projektille sopivaksi. Pohjimmiltaan Twincat 3 on PLC ohjelmointiohjelma,

jossa voidaan käyttää useita edellä mainittuja ohjelmointikieliä. Tässä opinnäytetyössä käytetään Structured text-ohjelmointikieltä. TwinCAT 3 -ohjelma julkaistiin 2011 ja se korvasi edeltäjänsä TwinCAT 2-ohjelmointiohjelman. (6.)

TwinCAT 3:ssa projektirakenne koostuu kuvan 3 mukaisista moduuleista, joita ovat esimerkiksi PLC-ohjelma, turvalogiikka sekä konenäkömoduuli Vision. Tässä projektissa edellä mainituista moduuleista käytössä on vain PLC-ohjelma. Ohjelmoinnin hierarkia koostuu projektista, jonka alle muodostuu projektin moduulit sekä loput tarvittavista työkaluista.

TwinCAT 3:sta otetussa kuvassa 3 havaitaan myös I/O-moduuli, joka pitää sisällään kaikki liitettävät laitteet ja sensorit. Nämä ovat fyysisen laitteen kannalta tärkeitä ja niitä voidaan tarpeen mukaan ottaa käyttöön. Simuloidussa ympäristössä näistä otetaan käyttöön vain servomoottorinot, ja tämän jälkeen servomoottoria voidaan käyttää toivotulla tavalla. TwinCAT 3 pystyy skannaamaan siihen liitetyt laitteet, jonka avulla se tuo ohjelmaan löydettyt laitteet ja niiden parametroidin.



KUVA 3. TwinCAT 3 -ohjelman rakenne.

Twincat 3 on todella kattava PLC-ohjelmointikonaisuus. Tässä työssä käytetään vain pientä osaa siitä mihin Twincat 3 on kykenevä. Työn myöhemmässä kappaleessa käydään läpi ohjelman luomista.

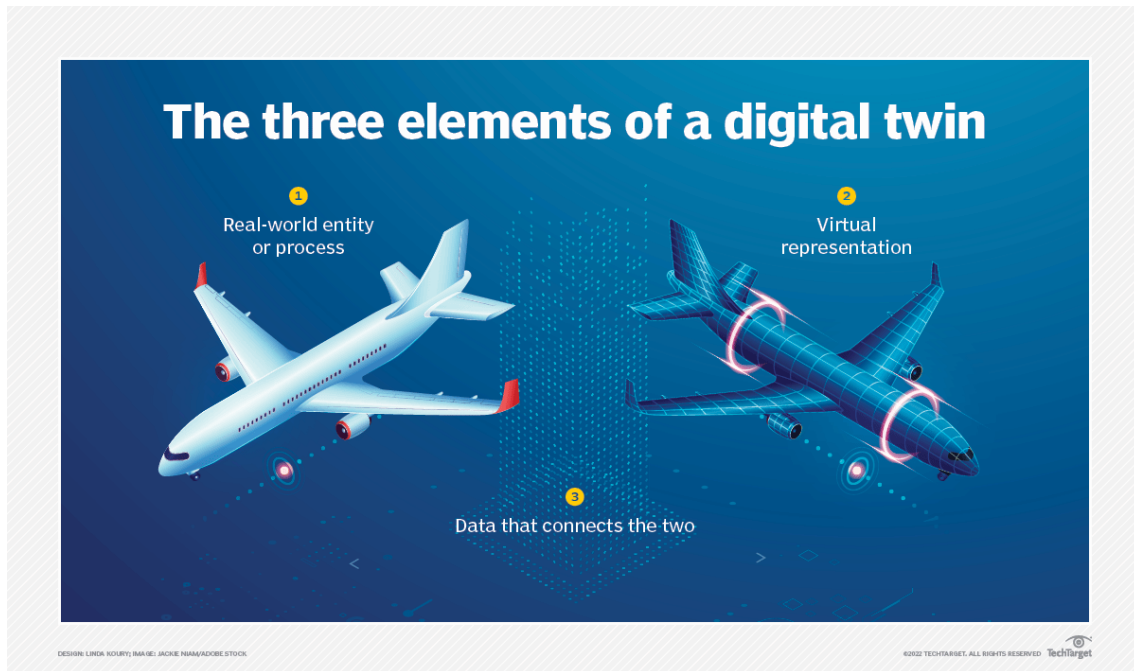
### 3 LAITTEISTON SIMULOINTI

Tässä kappaleessa käsitellään simuloimisen sekä digitaalisten kaksosten hyötyjä. Pääosin opinnäytetyössä keskitytään simulointiin Visual Components -ohjelmalla, mutta käydään läpi myös digitaalisten kaksosten tarjoamia hyötyjä.

#### 3.1 Digitaaliset kaksoset

Tässä opinnäytetyössä digitaalinen kaksonen ja simulointi luodaan Visual Components -ohjelmiston avulla ja tutkitaan teknologian hyötyjä virtuaalisessa käyttöönotossa. Digitaalisten kaksosten merkitys kasvaa päivä päivältä teollisuudessa. Niitä otetaan käyttöön sekä paremman optimoinnin vuoksi että säästösyistä. Näistä esimerkkinä voidaan sanoa uusien komponenttien testaus, sillä digitaalisella kaksosella voidaan helposti ja kustannustehokkaasti kokeilla useita muutoksia laitteistoon. Näin saadaan nopeasti tuloksia laitteiston uusimisesta ja miten se vaikuttaa tuotantoon, ilman, että fyysinen laitteisto joudutaan ottamaan pois käytöstä. (7.)

Digitaalinen kaksonen koostuu kolmesta osasta: fyysinen laite, virtuaalinen malli ja rajapinta, joka yhdistää kummatkin keskenään. Tämä malli näkyy kuvasta 4. Yleensä virtuaalinen laitteisto muodostetaan fyysisen laitteiston CAD-mallista tai luodaan uusi malli käyttäen samoja ohjelmia, joita suunnittelijat käyttivät CAD-malliin. Fyysinen laitteisto siirtää tiedon yleensä IoT:n avulla digitaaliselle kaksoselle. Tämän avulla voidaan toistaa reaaliaikaisesti tai halutulla viiveellä fyysisen laitteiston toimivuutta virtuaalisessa ympäristössä. (8.)



KUVA 4. Digitaalisen kaksosen rakenne (8.)

Digitaalisen kaksosen muodostaminen alkaa suunnitelmasta, jossa käydään tarkasti läpi, mitä mallilla halutaan toteuttaa. Nämä tulevat määrittävät pitkälti mallin suunnittelua sekä tiedonsiirtämisen toteutusta. Toinen tärkeä huomio on selvittää, mitä mallilla halutaan tehdä tulevaisuudessa. Tämä mahdollistaa sen, että mallilla nähdään selvä hyöty tulevaisuudessa, eikä yhtä käyttötarkoitusta esimerkiksi vain virtuaalisen käyttöönoton testauksessa. (9.)

### 3.2 Simulointiympäristö-ohjelmistot

Simuloinnin hyödyt kasvavat jatkuvasti rahallisen hyödyn sekä vihreän siirtymän takia. Simulointi on nopea tapa testata kokonaisuuksia ilman fyysisiä laitteistoja, jotka voivat nostaa testauksen hintaa ja hidastaa prosessia. Digitaalisten simulaattoreiden lisääntyminen on parantanut tuotannon tehokkuutta ja nopeuttanut päätöksentekoa. Simulaattoreissa voidaan testata useita eri toimilaitteita nopealla testauksella. (10.)

Simulointiympäristöissä voidaan testata helposti eri laitteistojen toimivuutta sekä yhteensopivuutta. Näin voidaan välttää kalliit testaukset fyysisen laitteiston kanssa sekä löytää mahdolliset ongelmatilanteet. Kokonaisuuksien saaminen simulointiympäristössä on helpompaa, ja sen avulla voidaan tehdä parempia investointi-ideoita. (11.)

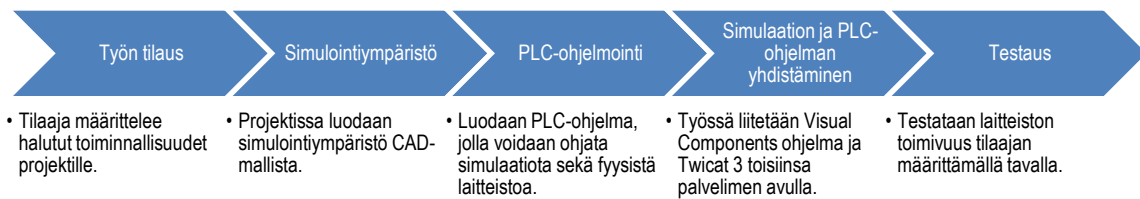
Tässä työssä käytetään CAD-mallia hyväksi luomaan 3D-simulointiympäristöä Visual Components -ohjelmaan, jotta voidaan testata laitteiston toimivuutta turvallisesti simulointiympäristössä. Visual Components valikoitui käytettäväksi tässä työssä, koska siitä on toimeksiantajaorganisaatiossa kokemusta. Valintaan vaikutti myös se, että Visual Components -ohjelmasta löytyi suoraan liitännät Twincat 3 -ohjelmalle, ja näillä liitännöillä saadaan ohjelmat helposti keskustelemaan keskenään. (11.)

Muita simulointiohjelmistovalmistajia ovat esimerkiksi Honeywell ja Siemens. Jokaisella simulointiohjelmistolla on oma käyttötarkoitus ja ympäristö. Simulointiohjelma tulee valita tarpeen projektin mukaan. Tällöin simuloitavasta prosessista ja simulaatiosta saadaan täysi hyöty irti.

Kappaletavaran sekä prosessien simulointiin tarvitaan ohjelmistolta kattavia ominaisuuksia sekä joustavuutta. Tämän takia jokaiselle prosessille on omat ohjelmistot. Visual Components on vain yksi tällaisista ohjelmistoista. Honeywell:n suurimmat etuudet muihin verrattuna ovat muun muassa heidän suunnittelemansa simulaattorimoottori Honeywell UniSim Desing ja kattava datankäsittely, joka on erityisesti luotu nestemäisiin prosesseihin sekä kaivosten simulointiin. (12.) Simit-ohjelmisto on Siemensin suunnittelema digitaalisen kaksosen ohjelmisto. Ohjelmalla voidaan laajasti luoda eri käyttötarkoitukseen simulointiympäristöjä, näiden tarkkuutta voidaan muokata käyttötarkoitusten mukaan. Tämän avulla saadaan toivottu tarkkuus simuloitavalle prosessille. Tämän ansiosta simulointi voidaan tarvittaessa jättää yksinkertaiseksi tai luoda todella kattava simulointi (13.).

## 4 TYÖN TOTEUTUS

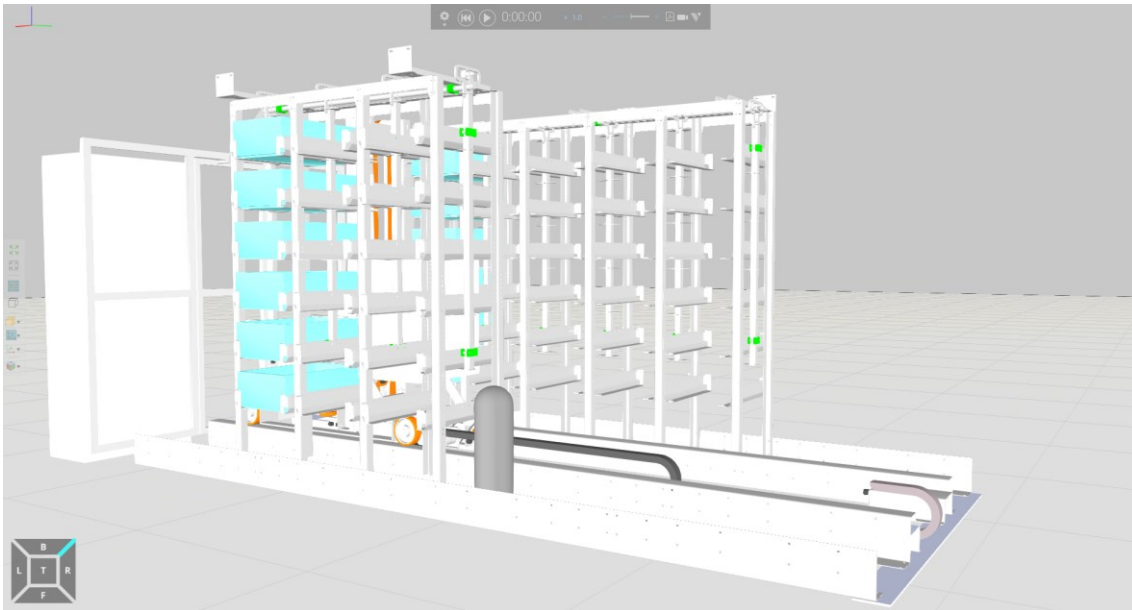
Työ toteutettiin toimeksiantajan tiloissa Oulussa keväästä kesään vuonna 2023. Toteutus koostui kolmesta vaiheesta: simulointiympäristön tekeminen, PLC-ohjelman luominen sekä PLC:n ja simuloimisen linkittäminen. Toteutusvaiheessa ohjelmointia sekä simulointia tehtiin samanaikaisesti ja varsinainen testaus suoritettiin laitteistojen yhdistämisellä. Kuviossa 1 nähdään työn eri välivaiheet, ja välivaiheita ja toteutusta käydään tarkemmin läpi tässä kappaleessa.



KUVIO 1 Työn toteutuksen vaiheet.

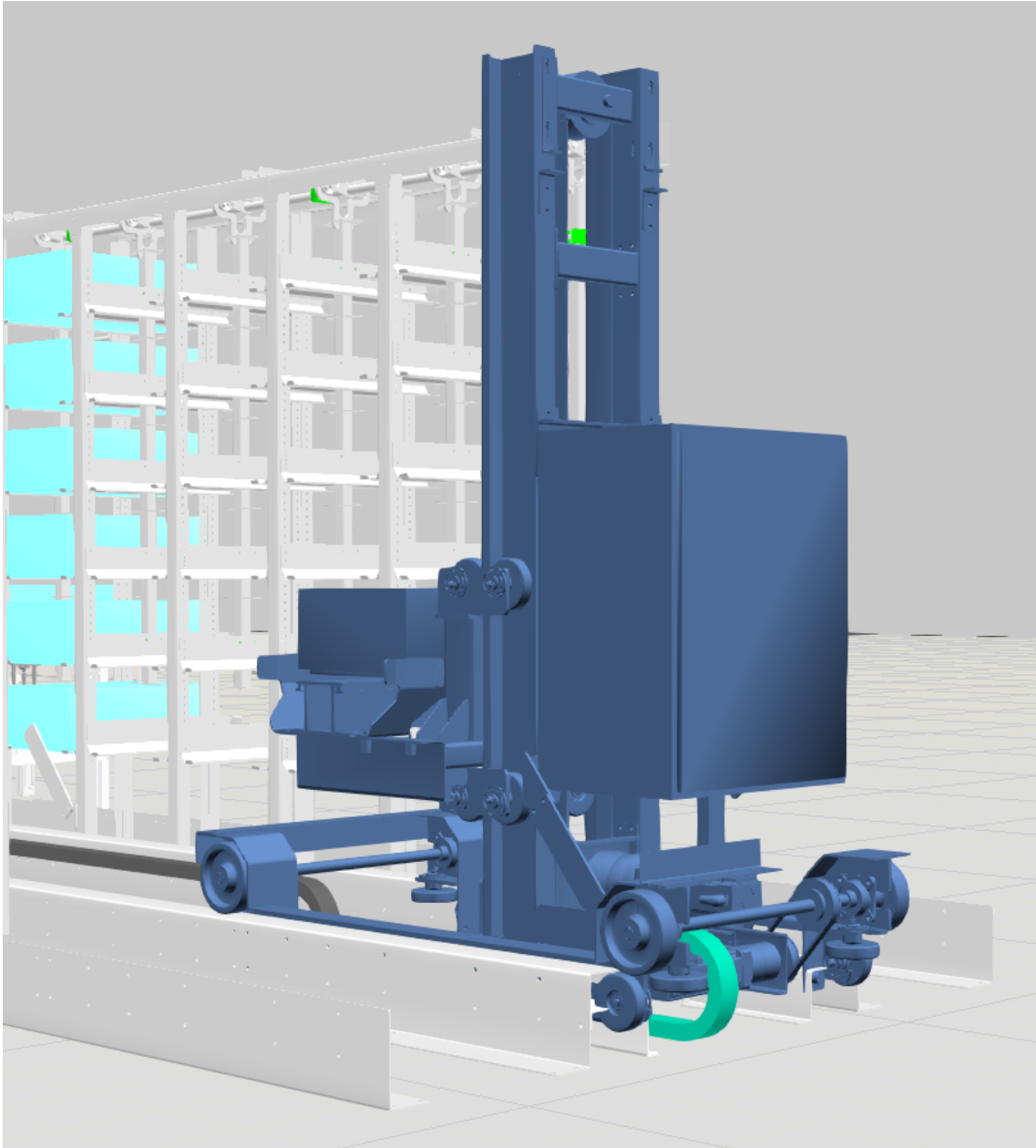
### 4.1 Simulointiympäristön luominen

Visual Components -ohjelmistolla luodaan uusi projekti, johon tuodaan CAD-malli varastoautomaatiojärjestelmästä. Tämän jälkeen malliin luodaan X, Y, Z- koordinaatistolle simuloitua servomoottoriakselit, joita voidaan ohjata PLC-ohjelman avulla. Kuvassa 5 Visual Components -ohjelmaan on tuotu varastoautomaation malli, josta lähdetään luomaan servomoottorien akseleita.



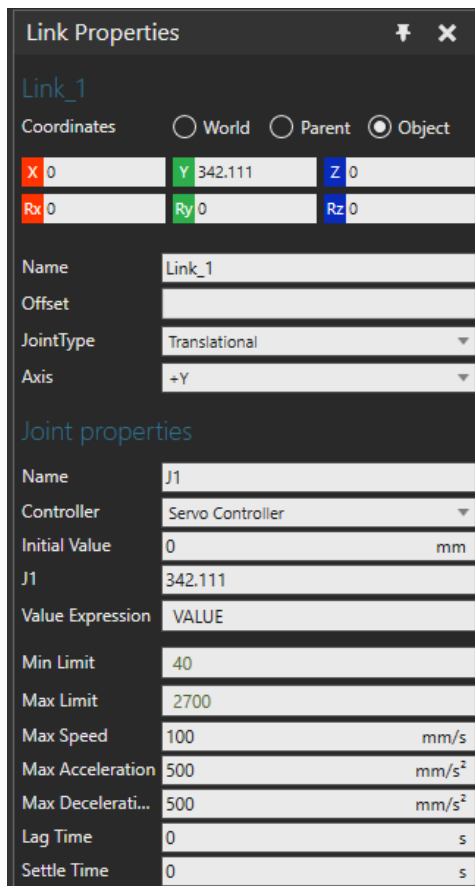
*KUVA 5. Varastoautomaation 3D CAD-malli tuotuna Visual Components -ohjelmaan.*

Kuvassa 6 on Visual Components -mallista irrotettu nostin, johon luodaan akselit simuloituille servomootoreille sekä tarvittavat liikeradat. Twincat 3 -ohjelma antaa ohjausarvoja akseleille, jotka liikuttavat akseliin liitettyjä komponentteja simulaatioympäristössä. Visual Components ympäristössä servo-ohjain kirjoittaa ja lukee tiedon millimetreinä, mutta Twincat 3 ei tue tätä muotoa suoraan. Twincat 3:sta lähtevälle tiedolle tulisi tehdä tietotyyppimuutos ja tämän avulla saataisiin Visual Components lukemaan oikeaa tietotyyppiä. Ongelma ratkaistiin syöttämällä määrättyjä paikkoja Twincat 3 -ohjelmalla tavalla, jota Visual Components pystyi lukemaan ongelmitta. Tämä ratkaisu ei kuitenkaan toimi, mikäli simulointia haluttaisiin käyttää rinnan fyysisen laitteiston kanssa.



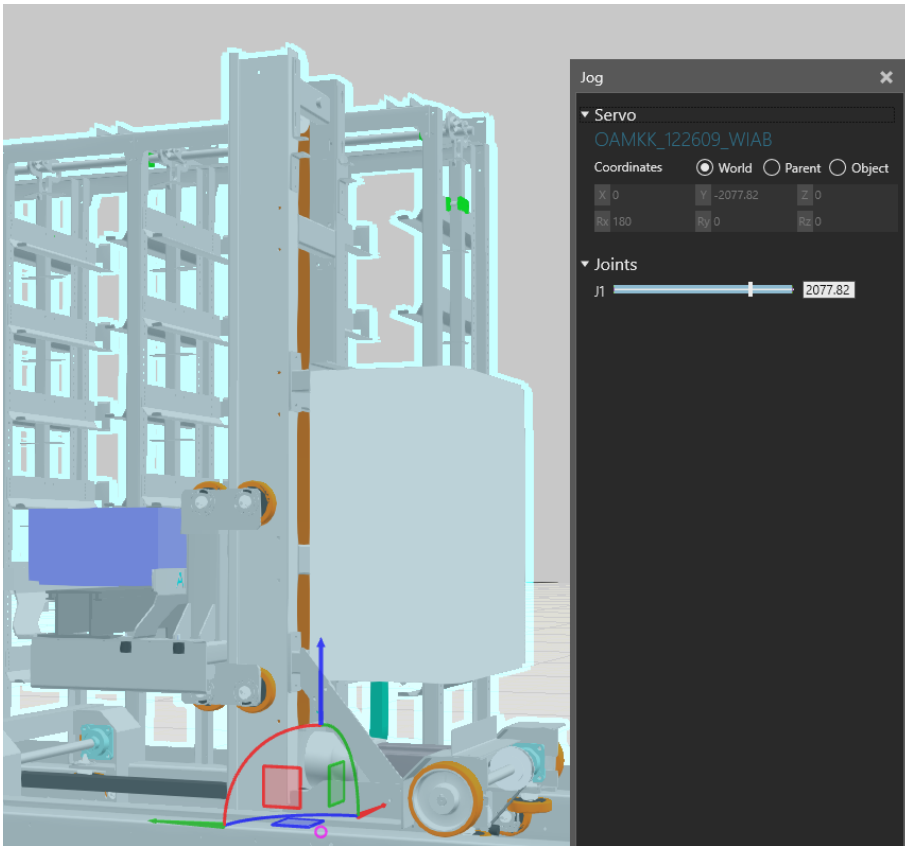
Kuva 6. Nostin eritelty mallista Visual Components -ohjelmassa.

Nostimelle on luotu liikerata sekä minimi- ja maksimiarvot, minkä sisällä nostin voi liikkua ohjauksen mukaan. Kuvasta 7 nähdään myös koordinaatiston suunnat, minkä mukaan nostinta tullaan ohjaamaan. Lisäksi akselille luodaan simuloitu servo-ohjain, jonka avulla nostinta voidaan liikuttaa sisäisellä simuloinnilla tai ulkoisella ohjauksella, kuten Twincat 3:lla. Kuvassa 6 voidaan myös huomata, että nostinta on ajettu liian pitkälle liikeradalla. Tämä voidaan huomata nostimesta, joka on liikkunut yli kiskostansa.



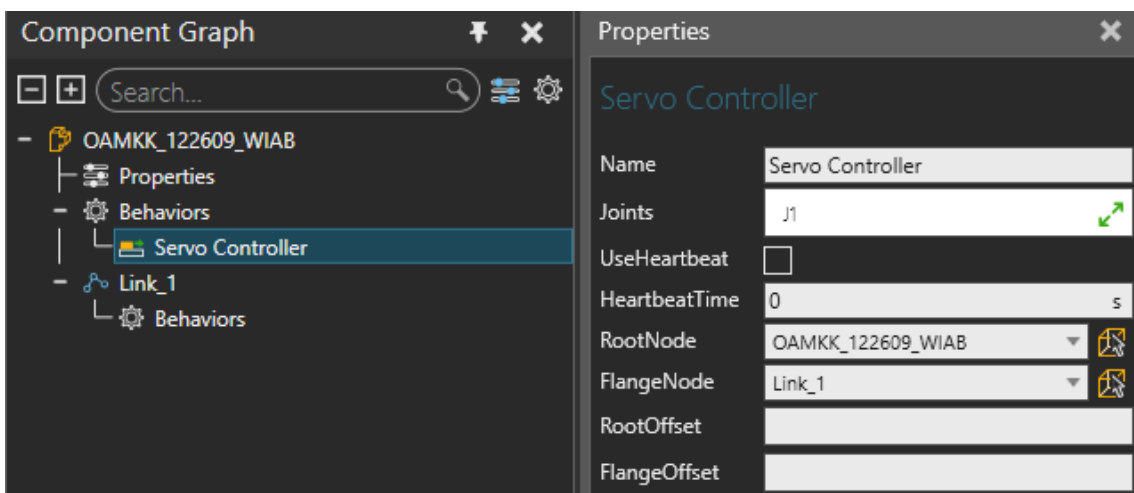
KUVA 7. Nostimen liikeradat Visual Components -ohjelmassa.

Visual Components -ohjelmalla voidaan testata simulointia ilman Twincat 3-integrointia. Simuloinnissa voidaan antaa akselleille esimerkkiarvoja ja seurata, toimiiko akseli simulaatiossa toivotulla tavalla. Kuvassa 8 nostimelle annetaan akselille arvo ja seurataan reaaliaikaisesti, että kappale liikkuu vain halutulla liikeradalla.



KUVA 8. Nostimen liikuttaminen Visual Components -ohjelmassa.

Visual Components tukee samanaikaisesti liikkuvia komponentteja ja näitä voidaan ohjata omilla ohjaimilla tai yhdellä ohjaimella useampaa. Tässä työssä kuvan 9 mukaan ohjataan vain yhtä akselia. Tälle akselille on määritetty yksi ohjain.

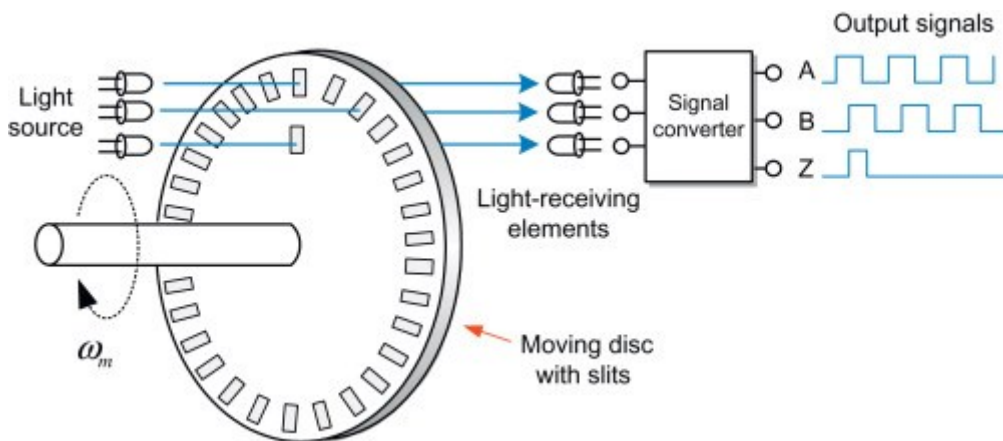


KUVA 9. Servo-ohjain simuloinnissa Visual Components -ohjelmasta.

## 4.2 PLC-ohjelman luominen

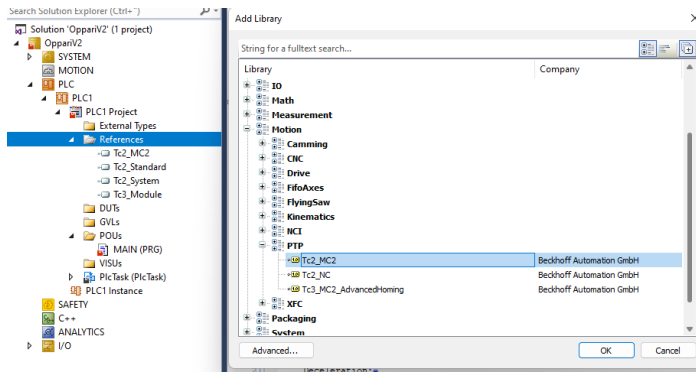
PLC-ohjelmointi aloitetaan Twincat 3:lla samalla tietokoneella, mihin simulointiympäristömalli on luotu. Twincat 3 -ohjelmalla voidaan joko ohjata simuloitua servomootoria tai fyysisen laitteiston moottoreita.

Servomootoreilta saadaan takaisinkytkentätieto pulssianturilta, jonka avulla saadaan tarkkaa tietoa moottorin pyörimissijainnista. Takaisinkytkentätieto muodostuu nopeus- ja kiihtyvyyssparametreihin. Näiden parametrien avulla servomootorin ohjausta säädetään, jotta moottoriliikkeen määrän pääksi saadaan haluttu sijainti. (14.) Kuvassa 10 nähdään pulssianturin toimintaperiaate. Anturissa valoa syötetään pyörivän kiekon läpi ja valo läpäisee kiekon siinä olevien reikien kautta. Kiekon vastapuolella on valoanturi, joka muuttaa tiedon signaaliksi.



KUVA 10. Pulssianturin toimintaperiaate. (15.)

Moottoriohjauksessa käytetään Twincat 3:n sisäänrakennettuja kirjastoja ja näitä kirjastoja voidaan tuoda tarvittaessa projektiin. Kirjasto pitää sisältää tarvittavat ohjainkäskyt ja kirjasto otetaan käyttöön projektissa, jonka avulla voidaan tuoda tarvittavat muuttujat ohjelmaan. Kuvassa 11 on esitettyä kirjaston tuominen projektiin Twincat 3 -ohjelmalla.



KUVA 11. Kirjasto Twincat 3 -ohjelmasta.

Kuvassa 12 on esimerkki servomoottorin ohjauksesta Twincat 3 -ohjelmassa. Tässä esimerkissä moottorille on annettu sijainti, johon moottori kääntyy ja tämän jälkeen pysähtyy. Muuttujaa voidaan myös käyttää sijaintina, jolloin se voidaan määrittää jokaisella ohjausliikkeellä uudestaan. Tässä opinnäytetyössä päädyttiin käyttämään määritettyjä sijainteja moottoreille ohjelman drive mode -komennolla. Tämän ansiosta ohjelma pysyi yksinkertaisena ja helposti ymmärrettävänä, eikä se vaikuttanut kokeilun onnistumiseen. Tämä toiminto koostuu määrätystä sijainnista, johon moottori ajaa ja jonka jälkeen drive mode määritellään nolaksi (0) eli odottamaan uutta ohjausta. Ohjauksia tässä työssä on kaksi piste akselilla ja nolapiste eli origo. Näitä voidaan ohjata vuorotellen, minkä avulla voidaan todeta simuloinnin toimivuus.

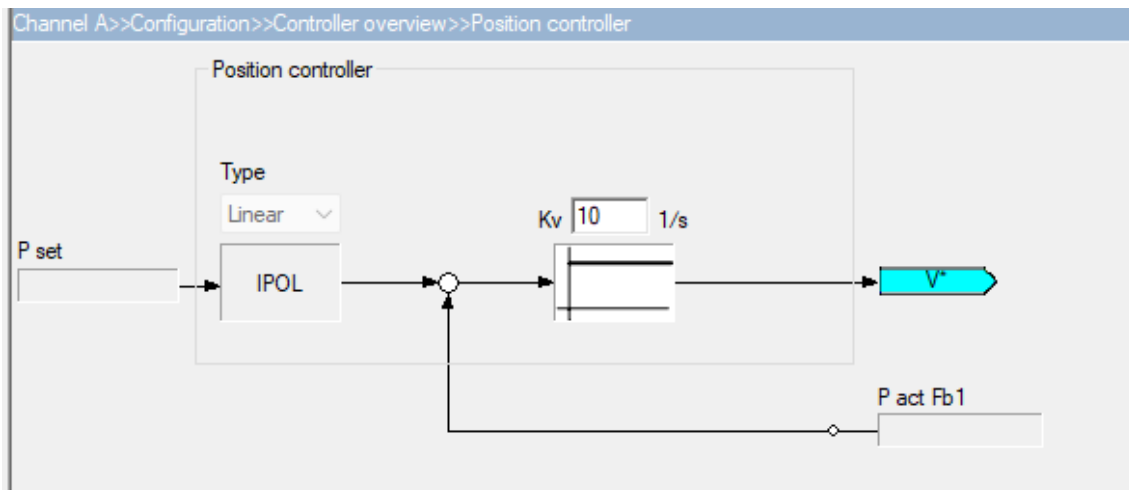
```

1 PROGRAM MAIN
2 VAR
3     servoREF      : AXIS_REF;
4     servoPower    : MC_Power;
5     servoMoveAbsolute : MC_MoveAbsolute;
6
7     driveMode     : INT :=0;
8 END_VAR
9
10
11
12
13
14
15 servoPower(
16     Axis:= servoREF,
17     Enable:= TRUE,
18     Enable_Positive:= TRUE,
19     Enable_Negative:= TRUE,
20     Override:= 100,
21     BufferMode:= ,
22     Options:= ,
23     Status=> ,
24     Busy=> ,
25     Active=> ,
26     Error=> ,
27     ErrorID=> );
28
29
30
31
32 servoMoveAbsolute(
33     Axis:= servoREF,
34     Execute:= ,
35     Position:= ,
36     Velocity:= 1000,
37     Acceleration:= ,
38     Deceleration:= ,
39     Jerk:= ,
40     BufferMode:= ,
41     Options:= ,
42     Done=> ,
43     Busy=> ,
44     Active=> ,
45     CommandAborted=> ,
46     Error=> ,
47     ErrorID=> );
48
49
50 IF servoPower.Status THEN
51     CASE driveMode OF
52         1: servoMoveAbsolute.Position := 500;
53            servoMoveAbsolute.Execute := TRUE;
54            driveMode := 0;
55         2: servoMoveAbsolute.Position := 0;
56            servoMoveAbsolute.Execute := TRUE;
57            driveMode := 0;
58     END_CASE
59
60 IF servoMoveAbsolute.Done THEN
61     servoMoveAbsolute.Execute :=FALSE;
62 END_IF
63 END_IF

```

KUVA 12. Twincat 3 servomoottorin ohjaus sekä asetusarvot.

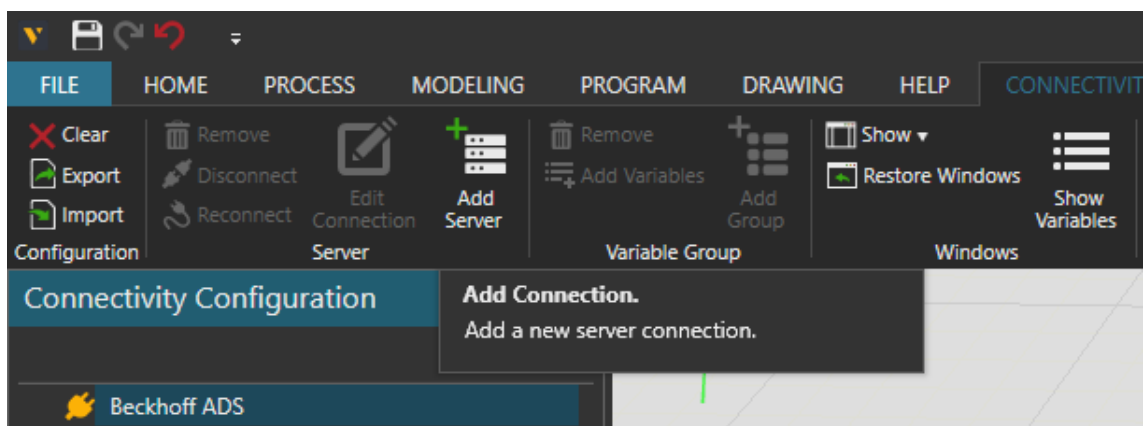
Ilman simulointia moottorilta saataisiin parametreilla takaisinkytkentätietoa, jolla voidaan määrittää tarvittava muutos moottorin ohjaamiseen. Simuloinnissa oikeaa takaisinkytkentätietoa ei ole mahdollista saada. Visual Components lähettää omaa sijaintitietoa moottorin pyörimisestä, mutta tätä tietoa ei pysty suoraan lukemaan Twincat 3 -ohjelmalla, tässä projektissa jätettiin selvitystyö tekeväksi. Takaisinkytkentä on tärkeä osa ohjausautomaatiota, mutta simuloinnissa laitteisto toimii halutusti ilman takaisinkytkentää, koska reaali maailman ulkopuoliset tekijät eivät vaikuta nostimen liikkeeseen. Kuvassa 13 on moottorin takaisinkytkentä Twincat 3 -ohjelmasta, jonka avulla ohjain säätää parametria ja tämä määrittää sijaintia liikkeelle.



KUVA 13. Moottorin sijainti ohjauksessa, Twincat 3 -ohjelmassa.

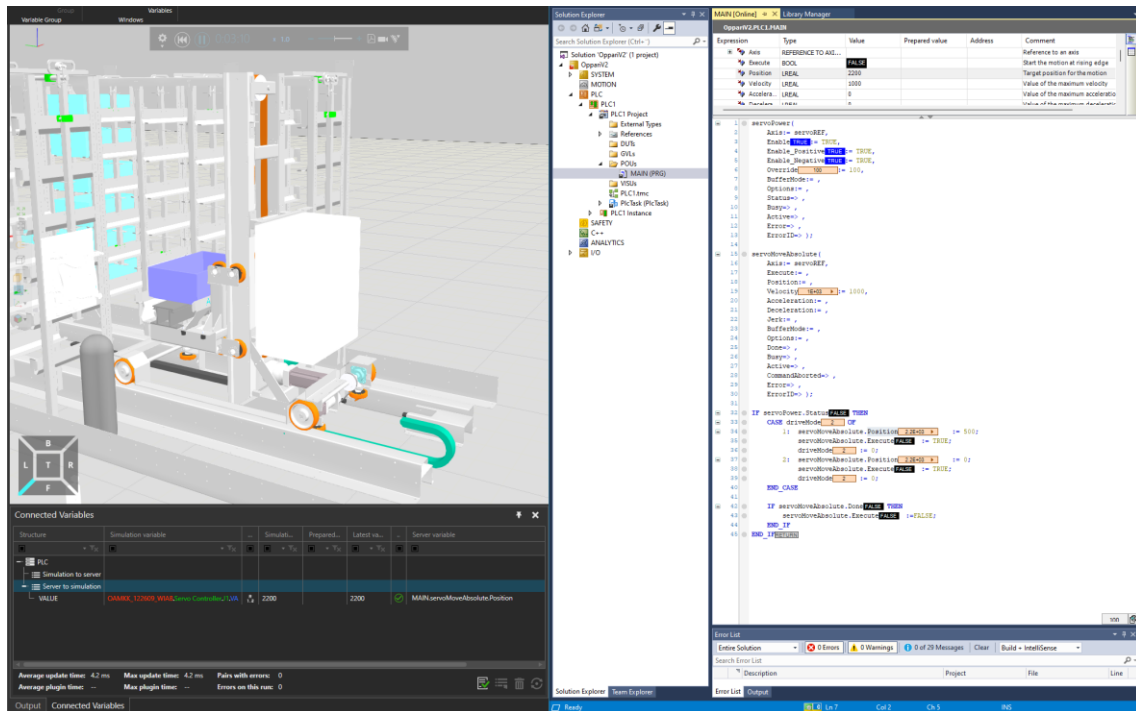
### 4.3 PLC:n ja simuloimisen linkittäminen

Twincat 3 -ohjelma sekä Visual Components saadaan jakamaan dataa toistensa välille luomalla palvelin. Palvelin luodaan Visual Components:iin ja näin saadaan ohjelmat keskustelemaan toistensa kanssa. Kuvan 14 avulla havaitaan, miten tietoa jakava palvelin luodaan Connectivity-väli-lehdeltä kahden ohjelman välille. Twincat 3 sekä Visual Components voivat jakaa dataa kyseisen palvelimen kautta. Valmis sisäänrakennettu palvelin johti Visual Componentsin valikoitumiseen tähän projektiin.



KUVA 14. Visual Components:in sisäänrakennettu palvelin.





KUVA 16. Simuloinnin testaus, vasemmalla Visual Components ja oikealla Twincat 3.

Kuvasta 16 voidaan havaita, että nostin on liikunut drive mode 2:lle asetettuun arvoon origosta. Tämän avulla voidaan todeta projektin testauksesta, että laitteisto toimii toivotulla tavalla ja siinä on halutut toiminnallisuudet. Tämän projektin tuloksia käydään tarkemmin läpi tulokset- luvussa.

## 5 TULOKSET

Tämän opinnäytetyön tarkoituksena oli tutustua automaatiovaraston ohjausjärjestelmään, simuloimiseen sekä digitaaliseen kaksoseseen. Pystyin luomaan tarvittavan moottoriohjauksen sekä luomaan yhteyden simulointiympäristön ja PLC- ohjelman välille. PLC- ohjelman toteuttamisessa onnistuin toivotulla tavalla, ohjelma saatiin testattua ja toiminnallisuus todistettua. Parannettavaa ohjelmassa sekä simuloinnissa silti on, sillä nostin esimerkiksi liikkui heti haluttuun paikkaan, eikä liikkeen nopeutta ollut mahdollista parametroida.

Opinnäytetyössä ei päästy toteuttamaan simulointia täysin omien toiveiden mukaan. Projektin aikataulu alkoi loppumaan, ja mikäli simuloinnista olisi haluttu täysin toimiva versio, olisi aikaa pitänyt olla enemmän. Aikataulun puitteissa saatiin tehtyä esiselvitystä PLC-ohjelmointiin sekä simulointiin tarvittavaan konfiguroimiseen ja simulointiympäristön rakentamiseen. Valmiiksi saatiin myös demoympäristö PLC:n ohjauksesta simulaatioalustalle. Työssä oli käytössä vain yksi akseli ja sitä ohjattiin kahden pisteen välillä.

Projektin eteenpäin vieminen jatkuisi luomalla täydellinen digitaalinen kaksosen varastoautomaatiojärjestelmästä sekä luomalla toimivampi simulaatioympäristö. Simulaatioympäristö sallisi laitteiston testaamisen sekä opettamisen ilman riskiä fyysisen laitteiston vaurioitumiselle. Tämä vaatisi ratkaisuja Twincat 3:n ja Visual Componentsin väliseen tiedonsiirtoon. Tämä olisi mahdollista, mutta se vaatisi enemmän aikaa ja suunnittelua.

## 6 POHDINTA

Suoriuduin omasta mielestäni opinnäytetyön tehtävästä hyvin ja taitojeni mukaisesti. Pystyin laajan aihepiirin takia haastamaan itseäni ja kehittämään tietotaitoani digitaalisista kaksosista sekä simulointiympäristöistä. Laaja aihepiiri antoi mahdollisuuden tutustua uusiin tekniikoihin sekä näkemään niiden hyötyjä tulevilla projekteilla.

Vaikeimpana työssä koin tutustumisen itselleni uuteen ohjelmistoon, Visual Components:iin. Ohjelmalle löytyi onneksi paljon materiaalia, joka edesauttoi projektin etenemistä ja aikataulussa pysymistä perustoiminnallisuuden toteuttamisessa. Omat tavoitteeni täydellisestä varastoautomaation simuloinnista jäi kesken Visual Components -ohjelmiston hyödyntämiseen liittyvien taitojen ja rajallisen aikataulun vuoksi. Suurimpaan osaan ongelmistani sain apua toimeksiantajaorganisaatiosta, jolta löytyi entuudestaan kokemusta Visual Components -ohjelmistosta.

Aiheesta löytyi paljon materiaalia useilta eri alustoilta, joista lähteeksi valikoitui luotettavia artikkeleita ja muita verkkolähteitä. Lähteiden luotettavuus osoittautui osittain haasteelliseksi, mutta lähdemateriaalia tuli vertailtua eri lähteiden välillä. Lähteet ovat monipuolisia ja ajantasaisia, vaikka ne ovatkin laajalta aikaväliltä. Tämän ansioista sain määritettyä luotettavat lähteet käyttöön.

Oulun ammattikorkeakoululla on halua jatkokehittää digitaalisen kaksosen mallia tästä opinnäytetyöstä. Olen ollut työsuhteessa Oulun ammattikorkeakoululla opinnäytetyö prosessin aikana ja tulen mahdollisesti jatkamaan digitaalisen kaksosen kehitystä varastoautomaatiosta ammattikorkeakoulun tarpeisiin. Minulta löytyy kiinnostusta perehtyä syvemmin digitaalisiin kaksosiin ja opetella hyödyntämään niitä teollisuuden tarpeissa.

## LÄHTEET

1. Behrtech. Digital Twins for industry 4.0: Applications benefits, and considerations. Hakupäivä 12.7.2023 <https://behrtech.com/blog/digital-twins-for-industry-4-0/>.
2. MECALUX. Advantages of automated storage and retrieval systems. Hakupäivä 10.8.2023. <https://www.mecalux.com/blog/advantages-of-automated-storage-and-retrieval-systems>.
3. Logistiikan maailma. Varastohallintajärjestelmät. Hakupäivä 10.8.2023. <https://www.logistiikanmaailma.fi/logistiikka/ohjausjarjestelmat/varastonhallintajarjestelmat/>.
4. WatElectronics. Step by Step Procedure of PLC Programming in Industries. Hakupäivä 3.6.2023. <https://www.watelectronics.com/how-to-program-the-programmable-logic-controllers/>.
5. Inductive automation. PLC Programming Languages: Go Beyond Ladder Logic. Hakupäivä 3.6.2023. <https://inductiveautomation.com/blog/plc-programming-languages-go-beyond-ladder-logic>.
6. Beckhoff. TwinCAT automation software. Hakupäivä 1.6.2023. <https://www.beckhoff.com/en-en/products/automation/twincat/>.
7. MathWorks. What is a Digital Twin. Hakupäivä 1.6.2023. <https://se.mathworks.com/campaigns/offers/next/digital-twins-for-predictive-maintenance.html>.
8. Techtarget. Defintion digital twin David Essex. Hakupäivä 9.6.2023. <https://www.techtarget.com/searcherp/definition/digital-twin>.
9. Suomen Tieyhdistys. Digitaaliset kaksoset virtuaalista ja todellista elämää rinnakkain. Hakupäivä 1.6.2023. <https://www.tieyhdistys.fi/tie-ja-liikenne/artikkelit/digitaaliset-kaksoset-virtuaalista-ja-todellista-elamaa-rinnakkain/>.
10. Visual Components. Developing Industrial Processes Through 3D Simulation. Hakupäivä 1.6.2023. <https://www.visualcomponents.com/resources/blog/developing-industrial-processes-through-3d-simulation/>.
11. Visual Components. Why You Should Seriously Consider 3D Manufacturing Simulation. Hakupäivä 3.6.2023. <https://www.visualcomponents.com/resources/blog/why-you-should-seriously-consider-3d-manufacturing-simulation-videos-included/#top>.
12. Honeywell Forge. Honeywell Process Digital Twin Go Beyond Visualization. Hakupäivä 10.8.2023. [https://www.honeywellforge.ai/us/en/products/industrials/honeywell-process-digital-twin?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=23-q2-hce-ci-ww-](https://www.honeywellforge.ai/us/en/products/industrials/honeywell-process-digital-twin?utm_source=google&utm_medium=cpc&utm_campaign=23-q2-hce-ci-ww-)

[process dig twin product cmpn-7016s000002wdbhqa4-europe&utm\\_ad-group=&utm\\_term=&gclid=Cj0KCQjwldKmBhCCARIsAP-0rfyc36Rzj00S-gdhc8u-6oN-Pw4t2F5-E0NGYM350jnceTYB1e06VnoaAuKcEALw\\_wcB.](https://www.siemens.com/global/en/products/automation/simatic/simatic-ipc/simatic-ipc-7016s000002wdbhqa4-europe&utm_ad-group=&utm_term=&gclid=Cj0KCQjwldKmBhCCARIsAP-0rfyc36Rzj00S-gdhc8u-6oN-Pw4t2F5-E0NGYM350jnceTYB1e06VnoaAuKcEALw_wcB)

13. Siemens. Simulation with SIMIT and SIMATIC PCS 7 using a practical example. Hakupäivä 10.8.2023. <https://support.industry.siemens.com/cs/document/77362399/simulation-with-simit-and-simatic-pcs-7-using-a-practical-example?dti=0&lc=en-US>.
14. Motion control tips. What is a servo drive. Hakupäivä 3.6.2023. <https://www.motioncontrol-tips.com/faq-what-is-a-servo-drive/>.
15. ScienceDirect. Incremental Encoder. Hakupäivä 7.6.2023. <https://www.sciencedirect.com/topics/engineering/incremental-encoder>.