



TAMPEREEN
AMMATTIKORKEAKOULU

MECHATRONICS CONCEPT DESIGNER OHJELMISTO OPETUSKÄYTTÖÖN

Asko Pakkala

Opinnäytetyö
Joulukuu 2019
Koneautomaatio
Teollisuustalous



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Kone- ja tuotantotekniikka
Koneautomaatio

PAKKALA, ASKO:
Mechatronics Concept Designer ohjelmisto opetuskäyttöön

Opinnäytetyö 73 sivua, joista liitteitä 2 sivua
Joulukuu 2019

Tämä opinnäytetyö tehtiin Tampereen ammattikorkeakoulun kone- ja tuotantotekniikan insinööriopintotoiminnan tueksi koneautomaation opintopolulle. Työn tavoitteena oli hyödyntää Tampereen ammattikorkeakoulun virtuaalista 3D-mallia kuljettimesta, johon yhdistetään automaatio-ohjelmistolla tehty logiikkaohjelma, jonka jälkeen simuloidaan kuljettimen toiminta virtuaalisesti.

Kuljetin on tällä hetkellä käytössä oppilaitoksen ohjausjärjestelmän kurssilla, mutta ongelmaksi on muodostunut kuljettimien vähäinen määrä, jolloin jokaiselle opiskelijalle ei ole riittänyt omaa kuljetinta ohjelmointiharjoituksia varten. Tämän vuoksi on ollut tarve kehittää kuljetin virtuaaliseen muotoon ja määrittää kuljettimen toiminta mekaniikan ja erillisen logiikkaohjelman avulla.

Kuljettimen 3D-malli viedään Siemens NX 3D Mechatronics Concept Designer (MCD) -ohjelmaan simulointia varten. Simulaatioon kytketään OPC-rajapintatekniikan avulla TIA Portal -ohjelmalla tehty logiikkaohjelma ja suoritetaan virtuaalinen käyttöönotto kuljettimesta.

Opinnäytetyössä tutustutaan virtuaalisen suunnittelun, simuloinnin ja digitaalisen kaksonen määrittelmään sekä perehdytään yleisesti mekatronisen laitteen tuotekehitysmenetelmiin ja sen prosesseihin. Tämän lisäksi työssä käsitellään mekatronisen laitteen kehityksessä käytettäviä yleisiä projektinhallintamenetelmiä.

Työn tuloksena syntyi kattavat käyttöohjeet harjoitustyön muodossa, jossa opiskelijat määrittävät virtuaalisen kuljettimen toiminnallisuudet ja kytkevät siihen automaatio-ohjelmistolla tehdyn logiikkaohjelman. Tämän avulla opiskelijan on mahdollista syventää osaamistaan mekaniikka- ja automaatio-suunnittelussa ja niihin vaikuttavista asioista.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Mechanical and Production Engineering
Machine Automation

PAKKALA ASKO:

Mechatronics Concept Designer Software for Educational Use

Bachelor's thesis 73 pages, appendices 2 pages
December 2019

This thesis was carried out for the Degree Programme in Mechanical and Production Engineering at Tampere University of Applied Sciences (TAMK). The purpose of this thesis was to connect PLC automation software with mechanical design software and simulate a whole 3D model of the dynamics of the conveyor in the TAMK Mechanical Engineering laboratory.

The real conveyor is used in the programming course, but due to the lack of conveyors, students did not have enough exercises to master PLC programming. This thesis provides guidelines on how to combine a virtual 3D conveyor with logic programming and gives future students a deeper understanding of the software as well as the design processes in general.

In this study the 3D model of the conveyor were exported to a Siemens NX Mechatronics Concept Designer (MCD) module for the usage simulation. TIA portal automation design software was used for logic programming. Furthermore, the OPC interface was used to integrate the software and simulations.

The theoretical part of this thesis provides definitions of virtual design, simulation, commissioning and digital twin. Furthermore, typical mechatronics product development steps and project management processes are also discussed.

As a product of this study, guided exercise was created to deepen mechanical engineering student's understanding of mechanical and automation design in conjunction with virtual simulation.

Key words: virtual design, simulation, commissioning, digital twin

SISÄLLYS

1	JOHDANTO.....	7
2	MEKATRONISEN JÄRJESTELMÄN SUUNNITTELU.....	8
2.1	Mekatroniikka yleisesti.....	8
2.2	Mekatronisen järjestelmän suunnittelu.....	9
2.3	Tuotekehitysprosessin yleisiä vaiheita.....	9
2.3.1	Markkinakysynnän selvittäminen.....	10
2.3.2	Vaatimustenmäärittely.....	11
2.3.3	Luonnostelu ja mallinnus.....	13
2.3.4	Testaaminen ja varmentaminen.....	19
3	PROJEKTINHALLINTAMENETELMÄT.....	21
3.1.1	Perinteinen vesiputousmalli.....	21
3.1.2	Rinnakkainen tuotekehitysprosessi.....	22
3.1.3	V-malli tuotekehityksen tukena.....	23
3.1.4	Ketterän kehittämisen malli.....	24
3.1.5	Scrum-viitekehitys ketterässä kehityksessä.....	25
4	VIRTUAALINEN SUUNNITTELU.....	28
4.1	Simulointi suunnittelun yhteydessä.....	30
4.2	Perinteinen käyttöönotto.....	32
4.3	Virtuaalinen käyttöönotto.....	34
4.4	Digitaalinen kaksonen.....	35
5	KÄYTETTÄVÄT OHJELMISTOT.....	38
5.1	Siemens NX -suunnitteluohjelmisto.....	38
5.2	Mechatronics Concept Designer.....	39
5.2.1	MCD-simuloinnin rajoitukset.....	40
5.3	TIA Portal.....	41
6	MECHATRONICS CONCEPT DESIGNER TOIMINNALLISUUDET.....	43
6.1	Aloittaminen.....	43
6.1.1	Aloituserävalikot.....	44
6.2	Simuloinnin aloittaminen.....	46
6.2.1	Työkalut kappaleiden määrittämiseen.....	47
6.2.2	Liitoksien määrittäminen.....	51
6.2.3	Antureiden lisääminen.....	52
7	YHTEYS OHJATTAVAAN ERILLISEEN JÄRJESTELMÄÄN.....	54
7.1	OPC.....	54
7.2	MCD tiedonvälitys TIA Portal V15.1 ohjelmaan.....	55
7.2.1	PLCSIM Advanced.....	56

7.3	OPC-palvelimen asettaminen TIA Portal V15.1 ohjelmistoon	57
7.3.1	OPC UA -valinta	58
7.3.2	Yhteys PLCSIM Advanced -ohjelmistoon	60
8	HARJOITUSTEHTÄVÄN LAADINTA	64
8.1	Kuljettimen määrittäminen	64
8.1.1	Liikkuvien osien ja kosketuspintojen määrittäminen	64
8.1.2	Kuljetinhihnan liikkeen määrittäminen	65
8.1.3	Liikuteltavan kappaleen monistaminen	66
8.1.4	Anturin lisääminen	66
8.1.5	Kappaleiden poistaminen kuljettimesta	67
8.1.6	Mekanismin animointi	67
8.1.7	MCD -ohjaussignaalien luominen	68
8.1.8	Kuljetinhihnan signaalin määrittäminen	69
8.1.9	Signaalien yhdistäminen ohjelmistojen välille	70
8.1.10	Kuljettimen toiminnan simuloiminen	71
9	POHDINTA	73
	LÄHTEET	74
	Liite 1. MECHATRONICS CONCEPT DESIGN WORKFLOW	77
	Liite 2. S7-PLCSIM Advanced V2.0 SP1 täyttöohjeet (Siemens 2018)	78

LYHENTEET JA TERMIT

NX	Siemens PLM softwären kehittämä 3D-suunnitteluohjelmisto
CAD	Computer-Aided Design. Tietokoneavusteinen suunnittelu.
CAM	Computer-Aided Manufacturing. Tietokoneavusteinen valmistus.
MCD	Mechatronics Concept Designer. Siemens NX -lisäosa
CAE	Computer Aided Engineering. Tietokoneavusteinen laskenta.
FEM	Finite Element Method. Elementtimenetelmä.
CFD	Computational Fluid Dynamics. Tietokoneavusteinen virtausdynamikka.
PLM	Product Lifecycle Management. Tuotetiedon hallinta
PLC	Programmable Logic Controller. Ohjelmoitava logiikka

1 JOHDANTO

Markkinatilanteen jatkuva muutos ja asiakkaiden vaatimustason nousu asettaa uusia haasteita valmistavalle teollisuudelle. Globaaleilla markkinoilla pärjätäkseen yritysten tulee tuottaa laadukkaita tuotteita lyhyessä ajassa ja rajallisella budjetilla. Virtuaalista suunnittelua ja digitaalista kaksosta hyödyntämällä voidaan tehostaa tuotekehitysprosessia, jonka avulla voidaan saavuttaa säästöjä kustannuksissa.

Digitaalisen kaksosen avulla voidaan testata ja simuloida prototyyppisiä tietokoneavusteisten ohjelmien avulla ennen sen tuotantoon laittamista, jolloin virheellisten tuotteiden määrä vähenee ja saavutetut hyödyt ovat näin suuremmat.

Opinnäytetyön teoriaosuudessa käsitellään yleisiä vaiheita, jotka tulee ottaa huomioon mekatronisen laitteen virtuaalisessa suunnittelussa. Tämän lisäksi esitellään yleisiä projektinhallintamenetelmiä, jotka ovat käytössä erilaisissa tuotekehityshankkeissa, joihin hyvin moni työntekijä uransa aikana osallistuu.

Tässä opinnäytetyössä on tarkoitus hyödyntää virtuaalista kaksosta yhdistämällä 3D -suunnitteluohjelmisto ja automaatio-suunnitteluohjelmisto yhteen ja simuloida Tampereen ammattikorkeakoulun opetuskäytössä olevaa kuljetinta. Opetuksen haasteena on ollut, ettei kuljettimia ole riittänyt kaikille opiskelijoille harjoitustöiden tekemistä varten. Tämän lisäksi konetekniikan koulutuksessa on useita aikuisopiskelijoita, jotka eivät aina pääse osallistumaan lähiopetus tunneille. Työn tarkoitus on selvittää, kuinka Siemens NX Mechatronics Concept Designer (MCD) ja TIA Portal -ohjelmistojen yhdistäminen ja simulointi tapahtuu, jotta opiskelijat voisivat tehdä ohjausjärjestelmä kurssin harjoitustöitä omalla koneellaan. Koululle saapuessaan opiskelija voi ladata virtuaalisena mallina luodun logiikan fyysiselle kuljettimelle ja todentaa osaamisensa kurssilla.

Tämän opinnäytetyön tavoitteena on tutustua virtuaaliseen suunnitteluun ja simuloimiseen luomalla käyttöohjeet harjoitustehtävän muodossa konetekniikan opiskelijoille. Harjoitustehtävän avulla opiskelija pystyy määrittämään virtuaalisen simulointimallin toiminnallisuudet ja kytkemään siihen automaatio-ohjelmistolla tehdyn logiikkaohjelman.

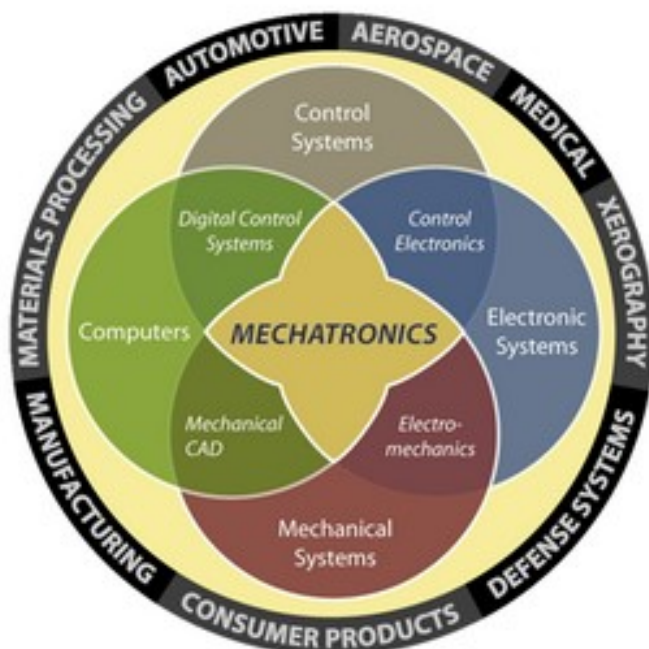
2 MEKATRONISEN JÄRJESTELMÄN SUUNNITTELU

2.1 Mekatroniikka yleisesti

Mekatroniikka on monitieteellinen insinöörialala, joka käsittää tieto-, sähkö-, kone- ja elektronisen ohjaustekniikan yhteensovittamista ja näiden keskinäisiä riippuvuuksia. Mekatroniikka käsitteenä on monitulkintainen, eikä se tarkoita vain mekaanisen-, sähköisen –ja ohjausjärjestelmien yhteyttä. Vaan näiden kaikkien järjestelmien integroimiseksi yhdeksi isoksi kokonaisuudeksi (Bolton 2003).

Bishop (2006) mukaan mekatroninen järjestelmä voidaan jakaa viiteen erikoisosa-alueeseen:

- Fyysisten järjestelmien mallintamiseen (Physical Systems Modeling)
- Anturi ja toimilaitteisiin (Sensors and Actuators)
- Signaali ja sen erilaisiin järjestelmiin (Signals and Systems)
- Tietokone-ja logiikkajärjestelmiin (Computer and Logic Systems)
- Ohjelmisto-ja datan keräämiseen (Software and Data Acquisition)



Kuva 1. Mekatronisen järjestelmän pääsisältö (Craig 1995)

2.2 Mekatronisen järjestelmän suunnittelu

Mekatronisen järjestelmän ja laitteen suunnittelussa voidaan käyttää hyväksi perinteisiä tuotekehityksen menetelmiä. Mekatronisen laitteet tuotekehitys ei tapahdu suoraviivaisesti tehtävänasettelusta viimeistelyyn. Tehtyä työtä voidaan joutua tarkastelemaan kriittisesti sekä tekemään muutoksia tarvittaessa, jos hankkeen edistyminen on tuottanut uutta edistyneempää tietoa. Aikaisempia päätöksiä ja valintoja voidaan joutua muuttamaan uuden tiedon pohjalta eli iteroimaan oikeanlaisen mekatronisen järjestelmän kehittämiseksi. Moniteknisyyden ja haasteelliset tavoitteet niin aikataulun kuin vaatimusmäärittelyn osalta edellyttävät, että mekatronisen laitteen tuotekehitys tulisi tehdä tiiviissä yhteistyössä eri suunnittelijoiden ja sidosryhmien kanssa.

Tämän lisäksi erinäisten suunnitteluohjelmistojen optimaalinen käyttö on tärkeässä asemassa lopullisen tuotteen aikataulussa ja hinnassa (Iserman 2005, 24-35; Airila 2011,13).

2.3 Tuotekehitysprosessin yleisiä vaiheita

Tuotekehitys on uuden tuotteen elinkaaren ensimmäinen vaihe, jolle havaintaan asiakas-tarve tai kysyntä uudelle tuotteelle. Alkuvaiheessa tuotekehitystä pyritään selvittämään asiakkaan tarpeet, markkinat, tuotteen yleiset vaatimukset sekä tuotteen valmistamiseen tarvittavat työkalut. Kehitettävä tuote tulee olla teknisesti toteutettavissa ja se tulee arvioida markkinoihin nähden kannattavaksi, jonka jälkeen tuote suunnitellaan, toteutetaan ja testataan.

Digitaalisessa tuotekehitysprosessissa suunnittelu -ja testausvaiheissa pyritään hyödyntämään mahdollisimman paljon virtuaalisen suunnittelun ohjelmia (Ahola, Hovila, Karhunen, Nevala, Schäfer & Nevala 2011).

Ahola ym. (2011) kuvaa tuotekehitysprosessia iteratiivisena kehityksenä. Tuotekehitysprosessi koostuu viidestä päävaiheesta, jotka ovat markkinaselvitys ja projektisuunnitelma, vaatimusmäärittely (sisältää turvallisuusvaatimukset), virtuaalisuunnittelu sekä testaus.

Riippuen projektinhallintamenetelmästä tuotekehityksen päävaiheet ja sisältö ovat usein samansuuntaisia. Tietotekniikan ohjelmistokehityksessä on käytössä iteratiivisia ketteränkehityksen projektinhallintamenetelmiä. Konetekniikan puolella voidaan käyttää perinteistä vesiputousmallia, jossa tuotekehitys etenee lineaarisesti prosessin vaiheesta toiseen.

Mekatroniikan tuotekehitysprosessiin voidaan soveltaa erilaisia tuotekehitysmenetelmiä, mutta pääsisältö on kuitenkin menetelmästä riippumatta samansuuntainen alkaen aina tuotekehityksen vaatimustenmäärittelystä.

Tuotekehitysprosessi katsotaan päättyneeksi, kun tuotteen ensimmäinen kaupallinen versio saadaan markkinoille, joka täyttää asiakas -ja standardien asettamat vaatimukset.

2.3.1 Markkinakysynnän selvittäminen

Ennen tuotekehityksen aloittamista tulee selvittää, onko tuotteelle kysyntää markkinoilla. Tuotekehitys tulee perustua olemassa olevaan asiakastarpeeseen, jossa asiakas, markkinointi tai myynti määrittelee ongelman, jonka kehitettävä tuote ratkaisee. Asiakastarve pitää sisällään ongelmanasettelun lisäksi tuotteen yleiset ominaisuudet ja vaatimukset. Tämän lisäksi tuotteen toimintaympäristö, määräykset ja siihen vaikuttavat standardeista koota kokonaisvaltainen dokumentaatio.

Tarvekartoitus sisältää kuusi pääkohtaa:

Lähtötilanteen määrittämisessä pyritään löytämään vastauksia asiakasryhmän ja kilpailijoiden löytämiseen sekä mikä on tuotekehitystä tekevän yrityksen valtti kilpailutilanteessa. Tuotekehityksen laajuus, asiakastarpeiden ja kilpailijoiden tietämys markkinoista ennakkoon sekä tämän lisäksi oman lähtötilanteen tunteminen auttavat päätöksenteossa, tuleeko tuotekehitysprojekti aloittaa vaiko ei.

Tiedon kerääminen asiakkaan tarpeista, markkinoilta tai muilta sidosryhmiltä määrittellään tiedonvaihtamiseksi kahden henkilön välillä. Tietojenkeruun aikana tarkentuu vaatimusten tärkeysaste, iteraatiokierrosten avulla pyritään tarkentamaan mitkä vaatimuksista on vähemmän tai enemmän tärkeämpiä. Vaatimusmäärittelyn lopulla saadaan selville alkuperäiset asiakkaantarpeet, jotka toteutetaan tuotekehitysvaiheessa.

Asiakastiedon jäsentäminen ja analysointi on syytä jakaa tuotesuunnittelun avulla erilaisiin pienempiin osa-alueisiin esimerkiksi mekaniikan, tietotekniikan, elektroniikan, hydraulikan tai pneumatiikan mukaan, sekä tehdä näistä alustavia luonnoksia, tämä selkeyttää havaitsemaan laajemmin asiakkaantarpeita.

Tiedon kerääminen yrityksen kilpailutilanteen avulla pyritään varmistumaan, että kehitettävän tuotteen uutuusarvo tai markkinat kiinnostavat asiakkaita, eikä tuote ole samanlainen kuin kilpailijoilla.

Tuotetta koskevien tavoitteiden asettaminen ohjaa sekä selkeyttää suunnittelutyön aloittamista. Tavoitetta määriteltäessä on asiakkaan tarpeet tiedostettava, koska vaatimuksiin nähden liian korkeita tavoitteita ei nähdä tarpeellisena. Ilman tavoitteiden ja rajojen asettamista kehitysprojekti voi suistua väärään suuntaan ja tällöin ei pystytä vastaamaan asiakkaan tarpeisiin.

Tavoitteiden saavuttamiseen ohjaaminen on oleellista, jotta suunnittelun ja tavoitteiden saavuttamisessa onnistutaan. Kehitystyötä on pyrittävä säännöllisesti seuraamaan, jonka ansiosta palautteen antaminen henkilöstölle helpottuu.

Näiden pääkohtien läpikäynnin jälkeen tuotteesta on tehty sen ensimmäinen konsepti.

Ensimmäinen konsepti on dokumentti, joka pitää sisällään edellä mainittujen asioiden lisäksi projektin aikataulun, resurssit sekä vastaa määrittelyjen, analyysien ja laskelmien avulla käynnistetäänkö tuotekehitysprojekti, hankitaanko lisäselvityksiä vai hylätäänkö koko idea (Ahola ym. 2011).

2.3.2 Vaatimustenmäärittely

Vaatimustenmäärittely on tuotekehitysprosessin ensimmäisiä vaiheita, jolla on tärkeä rooli kehitettävän tuotteen onnistumisessa. Heikosti määritelty tuote vaikeuttaa suunnitteluprosessia, koska tuotteen eri osa-alueiden suunnitteluryhmät eivät tiedä tarkalleen työnsä tavoitetta. Tämä voi johtaa epävarmuuteen suunnittelutyössä ja puutteellisiin teknisiin ratkaisuihin, joita voidaan joutua korjaamaan useasti.

Onnistuneessa vaatimusmäärittelyssä tuotetta voidaan kehittää tehokkaammin ja vähemmällä iteraatiokierroksilla, mikä tarkoittaa säästöä tuotekehitykseen kuluvassa ajassa ja kustannuksissa. Vaatimusmäärittely tulee suorittaa vähintään kerran projektin alussa, mutta siihen sisältyy myös sen jatkuva päivittäminen tuotekehitysprosessin aikana täydennettävistä ja tarkentavista vaatimuksista.

Vaatimusmäärittely pitää sisällään tuotteen tavoitteen, tärkeimmät toiminnallisuudet, komponentit, tekniset rajoitukset sekä ympäristön ja loppukäyttäjän vaatimukset.

Vaatimusmäärittely on usein organisaation eri toimintojen vaatimusten yhteensovittamista, jossa tulee huomioida myynnin ja markkinoinnin, asiakkaan, työympäristön, standardien ja määräysten sekä tuotannon erityispiirteet.

Vaatimusmäärittelyn ymmärrettävästi dokumentoidut vaatimuslistat ja käyttötilannekuvaukset määrittelevät lähtöarvot virtuaalisuunnittelulle ja simulointilähtöiselle tuotekehitykselle. Kun tuotteen tärkeimmät toiminnallisuudet ja vaatimukset ovat selvillä ja kohdennettu helposti ymmärrettäviksi suunnittelutehtäviksi, helpottaa tämä virtuaalisten mallien vaatimusten määrittämistä sekä oikeanlaisten mallinnusmenetelmien ja työkalujen valintaa, joka voi vaikuttaa tuotekehitysprojehtin kestoon ja kustannuksiin.

Vaatimusmäärittely voidaan jakaa karkeasti vaatimusten käsittelyyn ja niiden hallintaan.

Vaatimusten käsittely sisältää neljä pääkohtaa:

Vaatimusten keruulla ja kehittämisellä pyritään tarkentamaan asiakkaan vaatimuksia, jotka helpottavat suunnittelutehtävien tekemistä. Vaatimuksien keräämisellä ja tarkentamisella ei määritellä tuotteen ominaisuuksia, jotka eivät lisää tuotteen arvoa asiakkaalle. Asiakkaan pitämistä mukana tuotekehityksen vaiheissa helpottaa suunnittelutyön kohdentamista sekä asiakkaan ymmärrys lisääntyy tuotteelle, jota hän on ostamassa. Vaatimusten määrittelyn päätavoite on muuttaa asiakastarpeen selvitykset toiminnalliseksi vaatimuksiksi.

Vaatimusten luokittelulla, kohdentamisella ja resursoinnilla on tarkoitus varmistaa, että kaikki tiedossa olevat vaatimukset on jaettu erillisiksi suunnittelutehtäviksi. Vaatimukset priorisoidaan välttämättömiin, hyödyllisiin ja käyttötarkoituksen perusteella kriittisiin ja ei-kriittisiin vaatimuksiin. Moniteknisen tuotteen vaatimusten ominaisuudet tulee hajauttaa suunnittelun eri osa-alueiden perusteella jokaiselle suunnittelutiimille erikseen. Tämä auttaa kohdistamaan työpanosta enemmän olennaisimpiin osa-alueisiin. Resursoinnilla pyritään määrittämään asiakastarpeen kartoitukset yksityiskohtaisemmin, jolla selvitetään mitkä vaiheista voidaan ostaa alihankintana ja mitkä tehdään itse.

Vaatimusten analysoinnin ja spesifioinnin avulla pyritään selkeyttämään tehtäväalueen sisältöä suunnittelijoiden ymmärtäviksi työtehtäviksi. Yksittäisten vaatimusten, sekä suunnittelutehtävien riippuvuuksien varmentaminen selvittää tuotekehityksen vaatimuksia. Tämän pohjalta luodaan dokumentaatio, joka helpottaa tuotteen suunnittelua.

Vaatimusten analysoinnissa tulee miettiä tuotteen konfiguroitavuutta, jos tuotteen vaatimuslista ja toiminnallisuudet ovat suuria. Tällöin tulee harkita, mikä on tuotteen ominaisuuksien taso, jota lähdetään toteuttamaan yhteen ja samaan tuotteeseen.

Vaatimusten todentamisella ja hyväksymisellä on tarkoitus käydä läpi, mitkä tuotteen tekniset vaatimukset vastaavat mitäkin asiakastarpeen ominaisuutta. Vaatimusten todentamisen yhteydessä luodaan dokumentti, josta käy ilmi vaatimusten riippuvuussuhteet. Vaatimusmäärittelyn päätökselle haetaan lopulta hyväksyntä, jonka viime kädessä tekee asiakas (Ahola ym. 2011).

2.3.3 Luonnostelu ja mallinnus

Luonnosteluvaiheessa tuotteesta luodaan spesifikaation avulla kokonaiskuva eli konsepti. Luonnostelussa pyritään luovaan toimintaan, jossa eri teknologioiden mahdollisuudet pyritään ottamaan mahdollisimman hyvin huomioon. Luonnosteluvaiheen tärkeimmät vaiheet ovat järjestelmän sisältämien toimintojen sekä materiaali-, energia-, ja tietovirtojen mallintamiset. Mekatronisessa järjestelmässä tietovirtojen mallinnus on ratkaisevassa asemassa (Airila 2011, 16).

Mekatronisen järjestelmän suunnitteleminen vaatii systemaattista kehitystyötä ja oikeiden virtuaalisten suunnitteluohjelmistojen käyttämistä. Mekatroniikan tietokoneavusteinen suunnittelu sisältää tuotteen rakenteen suunnittelemista CAD- ja CAE- ohjelmistojen avulla. Lisäksi se voi sisältää staattisten -ja dynaamisten mallien luomista, simulointia, sekä ohjelmistojen kehittämistä mekatroniselle järjestelmälle sopivaan muotoon (Bishop 2006, 2-9).

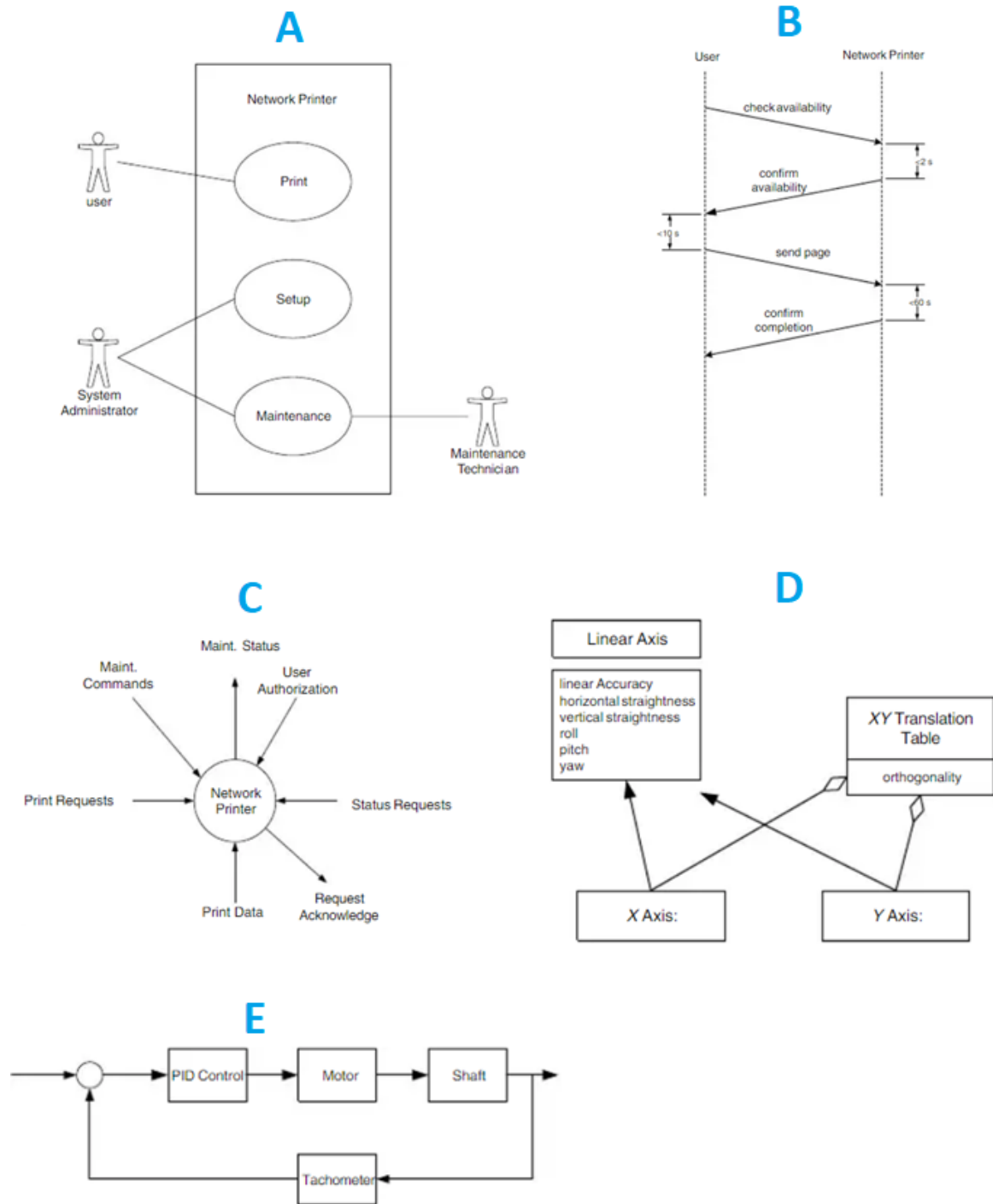
Mallinnuksen avulla voidaan palvella monia eri käyttötarkoituksia suunnitteluprosessin aikana. Pää tarkoituksena on parantaa suunnittelijoiden tuottavuutta tai kuvata tekniikoita, jotka parantavat tuotteen laatua ja pienentävät kustannuksia (Bishop 2006, 13.1 - 13.2).

Mallinnukseen on olemassa erinäisiä standardisoituja kaavioiden tuhrustustekniikoita sekä erilaisia mallinnusmenetelmiä ja kieliä, jolla voidaan helpottaa suunnitteluprosessia. Useimmat niistä ovat nykyisin standardisoituja ja ovat osa suosittua UML (Unified

modeling language) kieltä. UML -kieli laajalle levinnyt standardisoitu yleiskäytännöllinen graafinen mallinnuskieli, joka ei ole riippuvainen mallinnusohjelmistosta tai toimittajasta. Se sisältää runsaasti erilaisia kaavioita eri mallinuksia varten. Tämän lisäksi UML -kieli sisältää sääntöjä mallien luomiselle ja tarjoaa mahdollisuuden laajentaa niitä (Lahtinen n.d).

Bishop (2006) mukaan mekatronisen tuotteen kehitykselle ominaisia UML -kielen kaavioita ovat:

- Käyttötapauskaavio (*use case diagram*) (kuva 2A) kuvaa järjestelmän ja käyttäjän välistä vuorovaikutusta. Käyttötapauskaaviota käyttämällä on syytä huomioida, ettei se pidä sisällään ympäristön vuorovaikutuksia, vaan se on pelkästään järjestelmän korkeimman tason mallinnus.
- Sekvenssikaavio (*sequence diagram*) (kuva 2B) kuvaa yksittäisten käyttötapauskaavioiden järjestystä selventää vuorovaikutusta järjestelmän ja useiden ympäristö elementtien välillä. On kuitenkin huomioitava, että sekvenssikaavioiden dokumentit ovat erilliset eri käyttötapauskaavioille ja siihen vaikuttavat tietovirrat voivat olla molempisuuntaiset järjestelmän ja ympäristön välillä.
- Luokkakaavio (*class diagram*) (kuva 2C) kuvaa kuinka järjestelmä koostuu sen alijärjestelmistä ja mitkä sen komponentit ovat samankaltaisia ja mitkä niistä puolestaan taas eroavat toisistaan.
- Kontekstikaavio (*context diagram*) (kuva 2D) kuvaa järjestelmän ja sen ympäristön välillä vaikuttavia tietovirtoja. Huomionarvoista kontekstikaaviossa on sen yhteenveto eri vuorovaikutuksissa olevista tietovirroista kaikissa mahdollisissa käyttötapauksissa.
- Lohkokaavio (*block diagram*) (kuva 2E) kuvaa komponenttien välisiä yhteyksiä, jolla voidaan havainnollistaa laitteiden toimintaa.



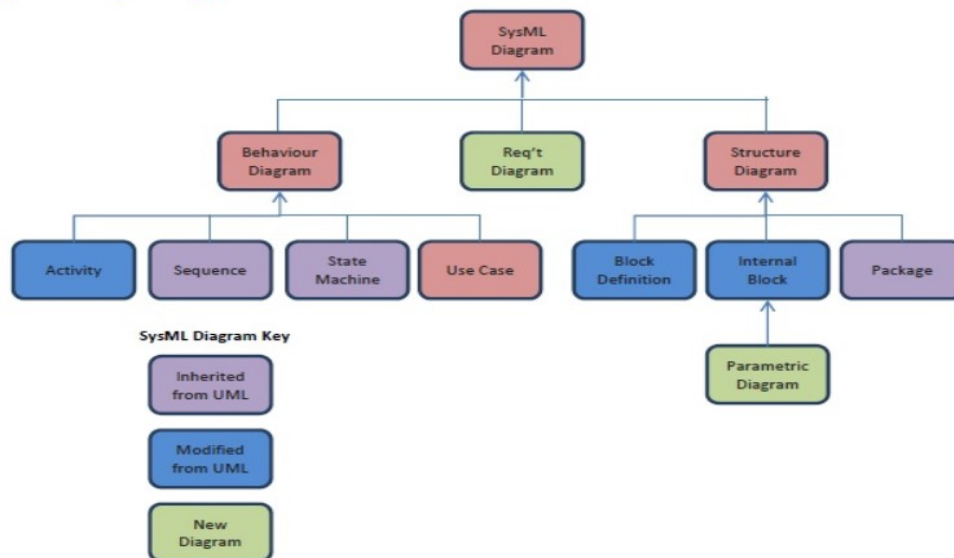
Kuva 2. UML-kielen kaaviotyypit (Bishop 2006)

Järjestelmäsuunnittelu (*System engineering*) on laajasti käytössä ilmailu -ja puolustusteollisuudessa, jossa pyritään usein ratkaisemaan haastavia -ja kriittisiä ongelmia. Järjestelmäsuunnittelu sisältää laitteisto-, ohjelmisto-, data-, ihmisten- ja palveluiden suunnittelua. Teknisten prosessien osalta järjestelmäsuunnittelu pyrkii analysoimaan, suunnittelemaan ja varmentamaan, että järjestelmän moduulit toimivat toistensa kanssa.

Järjestelmäsuunnittelun avulla tuotetaan lisäarvoa monimutkaisuusien ja riskien hallitsemiseen. Lisäksi järjestelmäsuunnittelulla on kyetty parantamaan tuotteiden laatua ja tuottavuutta esimerkiksi autoteollisuudessa, tietoliikenne alalla ja lääketieteellisten laitteiden parissa (Friedenthal, Moore & Steiner 2014, 4-5).

Malliperusteinen systeemis suunnittelu (*Model based system engineering*) on osa järjestelmäsuunnittelua, jonka tarkoitus on tukea analyysin, määrittelyn, suunnittelun ja varmentamisen avulla järjestelmien kehittämistä. Pää tarkoitus malliperusteisessa systeemis suunnittelussa on saada johdonmukainen malli kehitettävästä järjestelmästä (Friedenthal ym. 2014, 15-17).

SysML on UML-kielen kaltainen graafinen mallinnuskieli, jonka tarkoitus on mallintaa järjestelmiä, jotka voivat sisältää malliperusteisen systeemis suunnittelun periaatteita järjestelmistä ja niihin vaikuttavista tekijöistä. Nämä voivat olla laitteistojen, ohjelmistojen, datan, henkilöstöjen, toimenpiteiden, toimitilojen ja muiden ihmisen luomia elementtejä tai luonnollisia järjestelmiä. SysML-kielen on tarkoitus varmentaa, suunnitella järjestelmien arkkitehtuuria ja määrittellä komponentteja, kun puolestaan UML-kieli keskittyy pääasiassa ohjelmistojen toiminnallisuuksien suunnitteluun (Friedenthal ym. 2014, 31-33).



Kuva 3. SysML-kielen kaaviolajit (Dunstan n.d)

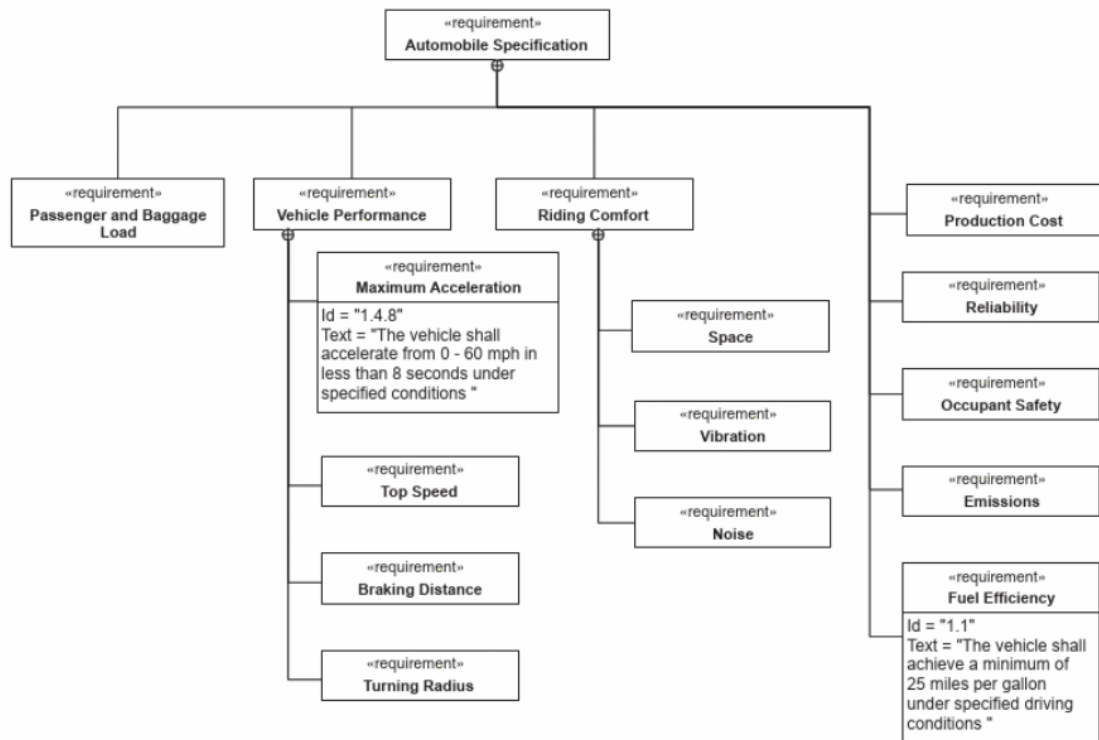
Kuvasta 3 huomataan, että SysML sisältää useita kaavioita. Osa kaavioista ovat identtisiä UML-kielen kanssa tai ovat muunnelmia niistä.

UML-kielestä SysML kieleen muunnetut kaaviot ovat:

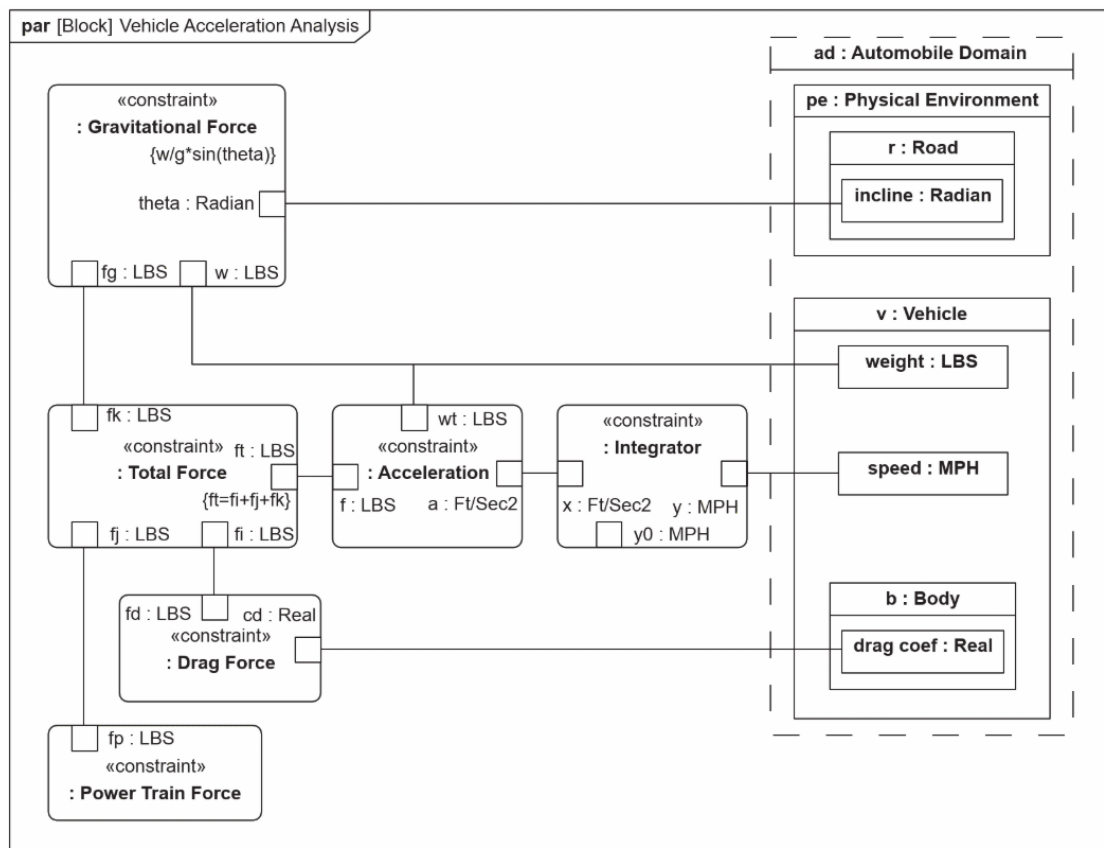
- Aktiiviteettikaavio (*Activity diagram*) kuvaa yksittäisiä toiminnallisuuksia järjestelmässä ja niiden suorittamisjärjestystä tietovirtojen perusteella.
- Lohkon määritelmäkaavio (*Block definition diagram*) kuvaa lohkon sisäisen rakenteen muodostamista ja luokitusta.
- Sisäinen lohkokaavio (*Internal block diagram*) kuvaa lohkon osien yhdistämistä ja rajapintoja.

Seuraavat kaaviot ovat pelkästään SysML kielessä:

- Vaatimuskaavio (*Requirement diagram*) (kuva 4) kuvaa tekstipohjaisten vaatimusten suhdetta toisiin vaatimuksiin, suunniteltuihin elementteihin sekä testaus tapauksiin, jotka tukevat vaatimusten jäljitettävyyttä.
- Parametrinen kaavio (*Parametric diagram*) (kuva 5) kuvaa järjestelmän ominaisuuksien rajoituksia. Parametrisen kaavion avulla voidaan määrittellä tarkempia suunnitelmia, joiden avulla pystytään varmentamaan järjestelmiä. Kuvassa 5 on havainnollistettu parametrinen kaavio käyttötarkoitusta, jossa ajoneuvon kiihtyvyyksien analyysiä on kuvailtu erillisten fysikaalisten lohko yhtälöiden avulla ja niiden riippuvuutta toisistaan.



Kuva 4. Ajoneuvon vaatimuskaavio (Friedenthal ym. 2014)



Kuva 5. Ajoneuvon kiihtyvyyden analyysi parametrisessä kaaviossa (Friedenthal ym. 2014)

Mekatroniikan kehityksessä malleilla voidaan palvella sen montaa eri käyttötarkoitusta. Mallien avulla dokumentoidaan käsitteitä, oletuksia ja vaatimuksia sekä sallitaan kommunikointi sidosryhmien välillä. Mallit tarjoavat hierarkkisen kehyksen, joka mahdollistaa työnjaon suunnitteluprosessissa sekä rinnakkaisen työntekemisen sen eri tasoilla. Mallit voivat tuottaa tietoa järjestelmän käyttäytymisestä, joka saattaa jäädä huomaamatta, kun järjestelmää tarkastellaan suurena kokonaisuutena. Mallit auttavat käsittämään yhtäläisyydet toisten järjestelmien kanssa ja auttaa ratkaisemaan nykyiset ilmenevät ongelmat. Lisäksi mallit voivat myöskin auttaa ymmärtämään ja testaamaan järjestelmän odottamatonta käyttäytymistä tuottamalla oletuksia testausten avulla (Bishop 2006, 6-7).

2.3.4 Testaaminen ja varmentaminen

Malliperusteiset suunnittelun vaatimuksena on kyetä simuloimaan mekatronisessa järjestelmässä olevia komponentteja. Simulointia voidaan toteuttaa matemaattisten yhtälöiden avulla tai käyttämällä erikseen erillisiä tietokoneavusteisia simulointiohjelmistoja. Tässä opinnäytetyössä keskitytään pelkästään tietokoneavusteiseen tekemiseen (Iserman 2005, 24-35).

Mekatroniikan digitaalisen tuotteen kehityksessä testaus ja varmentaminen eli verifiointi on vaiheistettu virtuaaliseen testaukseen, HIL- ja SIL- simulointeihin ja todellisilla prototyypeillä suoritettaviin testauksiin. Tuotekehitysprosessin loppupuolella suoritettavaa testausta ja verifiointi prosessia pyritään vaiheistamaan, jonka tavoitteena on minimoida fyysisten prototyyppien rakentamiseen liittyvät riskit ja kustannukset. Tekniset tuotteet ja ratkaisut kehitysprosessin aikana pyritään testaamaan mahdollisimman pitkälle virtuaalisten työkalujen avulla, jolloin ilmenevät puutteet ja suunnitteluvirheet voidaan korjata edullisesti. Testauksen ja verifiointin tavoitteena vahvistaa mekatronisen tuotteen ohjelmiston, osajärjestelmän tai koko tuotteen virheetön toiminta sekä parempi laatu.

Virtuaalisuunnittelun aikana tapahtuvan testauksen avulla pyritään varmistamaan ohjelmistojen virheettömyys, mekaanisten rakenteiden ja toimilaitteiden mitoitus sekä oikeanlaiset vaatimukset komponenttivalinnoille.

Testaus voidaan jakaa suunnittelutyön aikana tapahtuvaan testaukseen ja kokonaisen virtuaalimallin testaukseen. Suunnittelutyön aikana tapahtuvassa testauksessa keskitytään pienempiin osa-alueisiin kerralla. Kokonaisvirtuaalimallin testauksessa keskitytään tuotteen eri vaatimuksien testaamiseen.

HIL- ja SIL- simulointeja on käsitelty opinnäytetyön myöhemmässä vaiheessa.

Virtuaalisen testauksen jälkeen testataan määriteltyjen suunnittelutietojen pohjalta valmistettu ja kokoonpantu prototyyppi. Fyysisen prototyyppi testauksen tavoitteena on varmistaa, että tuotteelle asetetut vaatimukset täyttyvät todellisessa ympäristössä. Tämän lisäksi testataan suunnitellut järjestelmät ja valitut komponentit todellisella prototyypillä erilaisissa käyttöympäristöissä esimerkiksi kylmässä, kosteassa ja tärinässä (Ahola ym. 2011).

3 PROJEKTIHALLINTAMENETELMÄT

Projektit voivat olla suuria ja monimutkaisia kokonaisuuksia, millä voidaan tarkoittaa haastavaa, kertaluontoista ja tiimityönä tehtävää ponnistusta. Projektinhallintamenetelmät antavat ohjeita ja suuntaviivoja, miten työskennellä oikein, jotta projekti onnistuu. 1950-luvulla on alettu kehittämään projektinhallintatekniikoita ja onkin arvioitu, että projektinhallintaroolit tulevat kasvamaan merkittävästi vuosien 2010-2020 aikana. Arvion mukaan noin yksi neljäsosa maailman bruttokansantuotteesta käytetään erilaiseen projektityöhön. Tämän vuoksi projektihallinnallinen osaaminen on entistä tärkeämpää erilaisten projektitöiden onnistumisen kannalta, mihin hyvin moni työntekijä uransa aikana osallistuu (Pulkanen n.d).

Mekatronisen laitteen suunnittelua voidaan toteuttaa useilla erilaisilla tuotekehitys -ja projektinhallintamenetelmillä. Yleisempiä näistä menetelmistä ovat perinteinen vesiputousmalli (kuva 2), rinnakkainen suunnitteluprosessi (kuva 3), V-malliin perustuva tuotekehitysprosessi (kuva 4) sekä ketterään kehitykseen kuuluvat Agilen prosessit (kuva 5 ja 6). Näistä menetelmistä erityisesti V-mallia ja rinnakkaista suunnitteluprosessia käytetään mekatronisten laitteiden suunnittelussa. (Iserman 2005, 23-25).

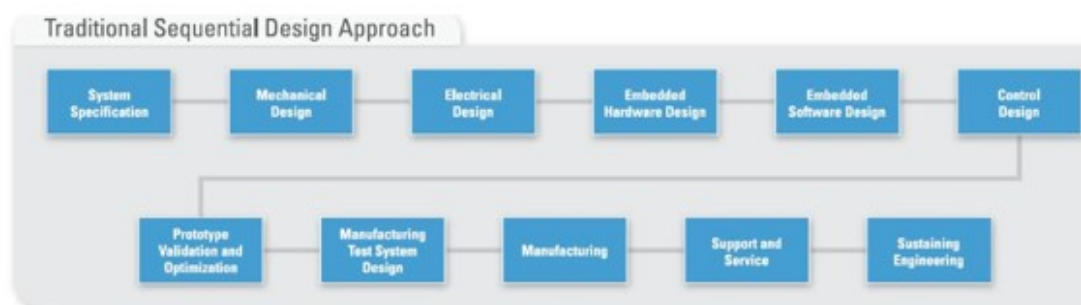
3.1.1 Perinteinen vesiputousmalli

Vesiputousmalli (waterfall model) on kehitetty alun perin vuonna 1970-luvulla Yhdysvalloissa ohjelmistokehityksen tarpeisiin. Perinteisen vesiputousmallin toimintatapa on tehdä tuotteeseen liittyvät tuotekehitysprosessit peräkkäin toistensa jälkeen, ennalta määrättyssä järjestyksessä (kuva 6).

Vesiputousmallin edut astuvat esiin suurissa projekteissa ja organisaatioissa, jossa pyritään täsmällisiin vaiheisiin ja aikatauluihin. Vesiputousmallin hyödyt voivat esimerkiksi olla projektin ylläpitoon ja tarkkuuteen liittyviä, eivätkä henkilöstön vaihtuvuus vaikuta projektin lopputulokseen merkittävästi. Lisäksi tavoitekeskeisyys jokaisessa vaiheessa antaa selkeyttä ja auttaa projektin jäseniä valmistautumaan seuraaviin vaiheisiin.

Tuotteen määrittelyn tai vaatimusten muuttuessa esimerkiksi virhe suunnittelussa tai markkinatilanteen äkillinen muuttuminen voi johtaa koko tuotekehitysprosessin keskeyttämiseen ja lopulta jopa projektin alusta aloittamiseen. Mekatronisen järjestelmän suunnittelussa perinteistä vesiputousmallia ei voida pitää kuitenkaan tehokkaana prosessin

etenemisvälineenä, johtuen mekatronisen laitteen iteratiivisesta kehitysprosessista, jossa voidaan joutua palaamaan prosessissa ja vaiheissa taaksepäin ja validoimaan asioita uudelleen (Powell-Morse 2016).



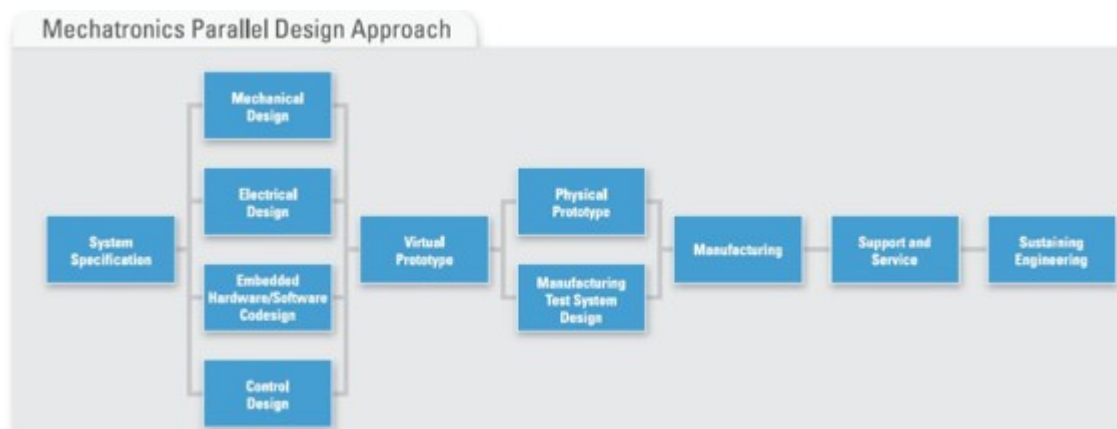
Kuva 6. Vesiputousmalli mekatroniikan näkökulmasta (National Instruments 2012)

3.1.2 Rinnakkainen tuotekehitysprosessi

Rinnakkainen tuotekehitysprosessi (Concurrent engineering) syntyi Yhdysvalloissa erinäisten asejärjestelmien kehityshankkeissa. Rinnakkaisella tuotekehityksellä voidaan tarkoittaa kahta asiaa. Suunnittelutyön tiedonkulun parantamista, kytkemällä kaikki sidosryhmät mukaan tai tuotekehitystä nopeuttamalla rinnakkaistamalla prosessin vaiheita ja hallinnoimaan paremmin prosessissa muuttuvia vaiheita siten, ettei koko suunnitteluprosessia tarvitse aloittaa alusta. Näiden lisäksi virheet kyetään paikantamaan nopeammin ja muutokset kyetään implementoimaan tehokkaammin kuin perinteisellä vesiputousmallilla.

Rinnakkaisen suunnittelumenetelmän käyttämistä projekteissa edellyttää, että projektinhallinta ja päätökset tulisi pystyä tekemään kollektiivisesti yhteistyössä eri tiimien kesken, vaikka haasteena voi olla yhteisen kielen puuttuminen, joka saattaa hankaloittaa projektinhallintamenetelmän käyttöönottoa.

Rinnakkaisella tuotekehitysmenetelmällä voidaan parantaa tuottavuutta, laatua ja koneiden käyttöastetta. (Leppälä 2000; Künzel 2005).



Kuva 7. Rinnakkainen tuotekehitysprosessi mekatroniikan näkökulmasta (National Instruments 2012)

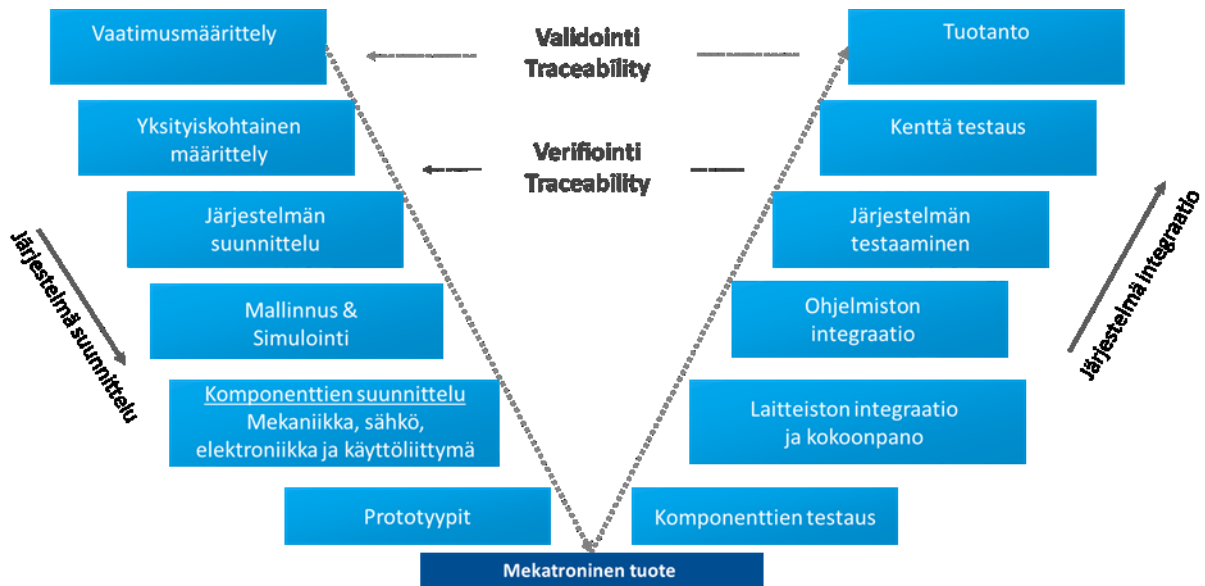
3.1.3 V-malli tuotekehityksen tukena

V-malli on suoraviivainen tuotekehityksen menetelmä, joka on kehitetty alun perin ohjelmistokehityksen tarpeisiin. Kuvassa 8 huomataan, että v-mallin tuotekehityksen prosessi rakentuu v-kirjaimen muotoon alkaen aina vaatimustenmäärittelystä valmiin laitteen tuotantoon saattamiseen. Vasen puoli kuvaa systeemin vaiheiden määrittelyä ja eri järjestelmien suunnittelua. Oikea puoli kuvaa järjestelmien, laitteiston, komponenttien ja ohjelmistojen testausta sekä näiden integrointia yhdeksi kokonaiseksi mekatroniseksi laitteeksi. V-mallin puutteena on sen muokattavuus kesken kehitysprosessin. Esimerkiksi järjestelmätestauksen aikana havaitun ohjelmistovirheen vuoksi voidaan joutua palaamaan järjestelmäsuunnittelun vaiheeseen ja korjaamaan havaittu bugi.

V-mallia on haasteellista soveltaa pitkän aikavälin projekteihin, jotka sisältävät useita uudistuksia ja muutoksia kesken tuotekehitysprosessin. Tämän vuoksi suunniteltu aikataulu ja budjetti voivat ylittyä projektin aikana.

V-mallin etuna on sen lineaarinen tuotekehitysprosessi ja siinä olevat selkeät vaiheet, jotka helpottavat pitämään tiukan aikataulun ja projektin tavoitteen ryhmän jäsenten mielessä.

V-malli soveltuu hyvin projekteihin, joiden kesto ja laajuus on tarkoin määritelty. Työskentelytavat ovat vakaat ja dokumentointi sekä eri kehityskaari on selkeät (Powell-Morse 2016).



Kuva 8. V-mallin tuotekehitysprosessi (Iserman 2005, muokattu)

3.1.4 Ketterän kehittämisen malli

Ketterän kehittämisen malli on alun perin kehitetty ohjelmistokehityksen tarpeisiin vuonna 2001 USA:ssa. Ketteryyden alkuperänä pidetään Agilen Manifestia eli toisoin sanoen ketterän ohjelmistokehityksen julistusta arvoista, joita sovelletaan, kun menetelmä tai käytäntö ei suoranaisesti vastaa ratkottavaan ongelmaan.

Ketterällä kehittämisellä tavoitellaan parempaa ja kattavampaa näkyvyyttä kokonaisprojektin tilasta, hyvällä teknisellä näkyvyydellä helpotetaan työntekijöiden työn organisoinnista ja projektin etenemistä kohti yhteistä tavoitetta.

Ketterässä tuotekehitysprosessissa projektia ei suunnitella täysin valmiiksi ennakkoon, eikä projektitiimit etene lineaarisesti vaiheesta toiseen niin kuin perinteissä vesiputousmallissa. Ketterän kehityksen eli agilen perusideaan kuuluu, että projektin kehityksen aikana on jo varauduttu muutoksiin iteratiivisilla toimintatavoilla.

Prosessin toimintatapa on tehdä lyhyitä vaiheita iteraatioissa eli sprinteissä suunnittelusta toteutukseen ja arviointiin kerta toisensa jälkeen ikään kuin miniprojekteina varsinaisen isomman projektin sisällä (kuva 5). Tämän avulla projektin suuntaan voidaan ketterästi muuttaa ja tarkastella aina uutta silmukan kierrosta aloittaessa. Projektin onnistumisen taustalla on tiivis ja jatkuva yhteistyö tiimin, että sen tärkeimpien sidosryhmien kanssa, jotta projektin aikana tekemät oikeat päätökset johtavat lopulta projektin onnistumiseen.

Agilea voidaan käyttää erityisesti pienempiin ja luovuutta vaativiin projekteihin, joissa on tiukka kehitysrytmi ja toteutustapoja voidaan muuttaa kesken projektin.

Agilesta on useita erilaisia sovelluksia, joilla voidaan tarkastella tarkemmin projektin määrittelyjä. Näistä sovelluksista erityisesti SCRUM viitekehystä voidaan soveltaa mekatronisen tuotteen kehityksessä. (Turun yliopisto & Työterveyslaitos: Sulautettujen järjestelmien ketterä käsikirja 2014; Pulkanen n.d)



Kuva 9. Ketterän kehityksen toimintaperiaate (Suomidigi n.d)

3.1.5 Scrum-viitekehys ketterässä kehityksessä

Scrum-viitekehys kuuluu ketterän kehittämisen menetelmiin. Scrum kehitetty ohjelmistokehityksen tarpeisiin 90-luvulla. Scrum soveltuu hyvin myös mekatronisen tuotteen projektinhallintamenetelmäksi.

Scrumin keskiössä ovat itseohjautuvat tiimit, päivittäisen työn etenemisen seuranta ja tarkat ennalta määrätyt lyhyet tapaamiset. Scrumin periaatteet ovat yksinkertaisia, mutta sen käytännöt vaikuttavat tiimin toimintaan ja työskentelytapoihin. Scrum koostuu prosessista, tiimistä ja erilaisista rooleista, joiden avulla työskennellään ketterästi.

Tuotetta kehitetään 1-4 viikon iteraatiojaksoissa eli *sprinteissä* kerrallaan. Sprintin pituuden päättää kehitystiimi ja sen aikana tapahtuu kehitystyö, testaus ja valmiiden toiminnallisuuden julkaiseminen.

Kuvasta 10 huomataan, että Scrum koostuu useista erilaisista vaiheista.

Ensimmäinen vaihe on määritellä projektin vaatimukset kehitysjonoon *Product Backlogille*, joka sisältää projektin vaatimukset listattuna. Jokaisen sprintin alussa järjestetään suunnittelupalaveri *Sprint planning*, jossa kehitystiimi valitsee ja suunnittelee työtehtävät iteraatiojakson kehitysjonolle *Sprint backlogille*, joka listaa sprintin aikana tehtävät toiminnallisuudet ja tehtävät.

Sprintin aikana kehitystiimi pitää päivittäin tilannepalavereita *Daily Scrumeja*, joiden tarkoitus on päivittää työnsä tilannetta edellisen ja tämän päivän osalta, sekä poistaa mahdolliset esiintyvät ongelmat.

Sprintin päättyessä järjestetään katselmointipalaveri *Sprint review*, jonka tarkoitus on käydä läpi, miten iteraatiojakso kokonaisuutena onnistui ja saatiinko kaikki sprintin kehitysjonossa olevat tehtävät tehdyksi. Tämän lisäksi katselmointipalaverissa voidaan esitellä valmiita toiminnallisuuksia asiakkaalle.

Jokaisen sprintin päätteeksi pidetään kehityskeskustelu *retrospective*, jossa tiimin tehtävä on käydä läpi edeltävää iteraatiojaksoa ja miettiä kuinka tiimien työskentelytapoja voitaisiin kehittää.

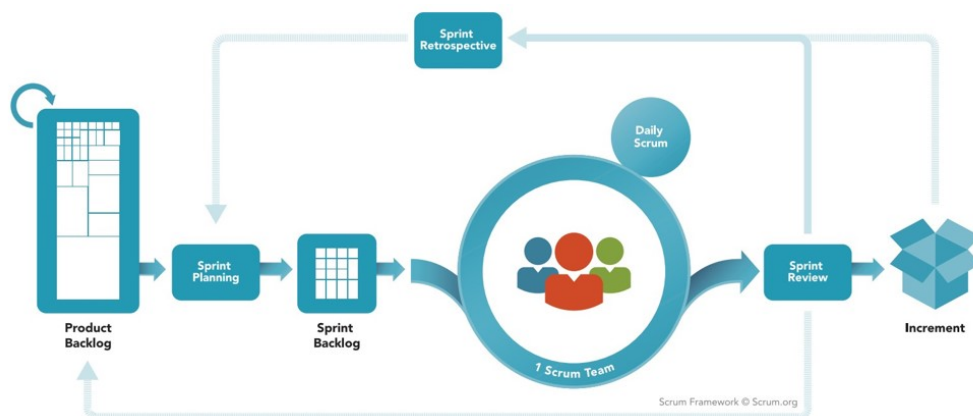
Scrumiin liittyy olennaisesti tiimit, jotka työskentelevät itsenäisesti ja itseorganisoituvasti. Tiimi päättää itse myös työskentelymenetelmistään sekä työnjaostaan.

Scrumissa koko kehitystiimi ottaa vastuun projektista ja tiimi tulee rakentaa siten, että sillä on tarvittava osaaminen vaadittujen työtehtävien tekemiseen.

Scrumissa tiimiin kuuluu tiimijäsenten lisäksi yleisesti fasilitoija ja tuoteomistaja.

Fasilitoija eli *Scrum Masterin* tehtävä on toimia projektihoitajan roolissa ja huolehtia, että tiimi noudattaa Scrumin periaatteita ja käytänteitä. Lisäksi hänen tehtävänänsä on varmistaa tiimille työrauha sprintin ajaksi ja poistaa työn etenemiseen liittyviä esteitä sekä yleensä organisoida päivittäiset pika -, iterointi- ja katselmointipalaverit sekä retrospektiivin.

Scrum -tiimiin kuuluu myös tuoteomistaja eli *Product Owner*, jonka tehtävä on tuotteen arvon maksimoiminen kehitystiimin tekemästä työstä. Tuoteomistaja hallinnoi ja on vastuussa tuotteen kehitysjonon *Product Backlogin* luomisesta, ylläpidosta ja priorisoinnista.



Kuva 10. Scrum kehitysprosessi (Scrum framework poster n.d)

Sulautettu järjestelmä, johon mekatronikka lukeutuu ohjaa vastuunkantamiseen eri osa-alueiden tiimien kesken ja pilkkomaan suuret kokonaisuudet tuotteen teknisten osaamis-alueiden sijaan toiminnallisuuksien kehittämiseen Scrumin avulla.

Ketterässä kehityksessä suunnitelmat tarkentuvat projektin edetessä, tämä voi tuoda haasteita fyysisen testattavan laitteen kehitykseen, koska laitteen kehitystyö saattaa vaatia usean viikon työpanoksen. Laiteprojektin kehittämistä aloittaessa tulee selvittää, mitkä asiat tehdään missäkin vaiheessa projektia ja muodostaa näistä projektin sisäiset vaiheet, joiden puitteissa voidaan edetä Scrumin periaatteiden mukaisesti.

Sulatettujen järjestelmien kehityksessä tiimit muodostuvat ohjelmisto- ja laitteistopuolen suunnittelijoista. Laitteistopuolen suunnittelijat käsittävät elektroniikka -ja mekaniikka-suunnittelun.

Haasteena sulautettujen järjestelmän ketterälle kehittämiselle on pitää tiivistä yhteyttä asiakkaan kanssa koko Scrum kehitysprosessin aikana. Tämän vuoksi kehittäjien tulee toimia tiiviissä yhteistyössä myyntiorganisaation kanssa, tunnistaakseen asiakkaidensa muuttuvat tarpeet ja vaatimukset. Myyntiorganisaation tehtävänä on kerätä asiakkailtansa palautetta säännöllisesti tuotteen toiminnallisuuksien esittelyiden aikana.

Tämän lisäksi haasteena voi olla laitteistojen ja ohjelmistojen väliset riippuvuudet. Järjestelmien väliset rajapintojen määrittäykset kehitysprosessin alussa on hyvä tehdä yhdessä kehittäjien kanssa, jotta kokonaisuus suunnitella saataisiin kaikkien osalta käyntiin ja pienetkin muutokset jaettaisiin tuotekehityksessä mukana oleville tahoille. Tämän avulla minimoidaan ylimääräisen ja päällekkäisen työn tekeminen (Turun yliopisto & Työterveyslaitos: sulautettujen järjestelmien ketterä käsikirja 2014).

4 VIRTUAALINEN SUUNNITTELU

Virtuaalisuunnittelulla tarkoitetaan suunnittelua, joka perustuu fysikaalisten ilmiöiden ja moniteknisten järjestelmien mallintamiseen ja simuloimiseen tietokoneohjelmistojen avulla. Virtuaalisuunnittelulla tavoitellaan tuotekehitysprosessin nopeuttamista, kehityskustannusten ja riskien pienentämistä sekä suunnittelulaadun parantamista.

Virtuaalisen suunnittelun päätavoite on saada tekniset määritykset olennaisimmille ja välttämättömille komponenteille, joita yleisesti tarvitaan fyysisen tuotteen tai osajärjestelmän prototyypin rakentamiseen.

Virtuaalisessa suunnittelussa onnistunut tuotekehitysprojekti edellyttää, että kaikki suunnittelutiimit ja sidosryhmät (asiakas, loppukäyttäjä, myynti ja tuotanto) osallistuvat yhdessä aktiivisesti kehitystyöhön sen alusta alkaen.

Kolmiulotteisen virtuaalimallin luonnosteluvaiheessa kyetään näkemään suunnitteilla olevan mallin toiminnallisuus, kun se on visuaalisesti nähtävissä ja puutteet sen suunnittelussa tai tuotteen vaatimuksissa ovat helpommin havaittavissa. Lisäksi virtuaalimallin demonstraatio on oiva tilaisuus koota yhteen eri sidosryhmät projektin vuorovaikutuksen lisäämiseksi ja palautteen keräämiseksi ketterien menetelmien mukaisesti.

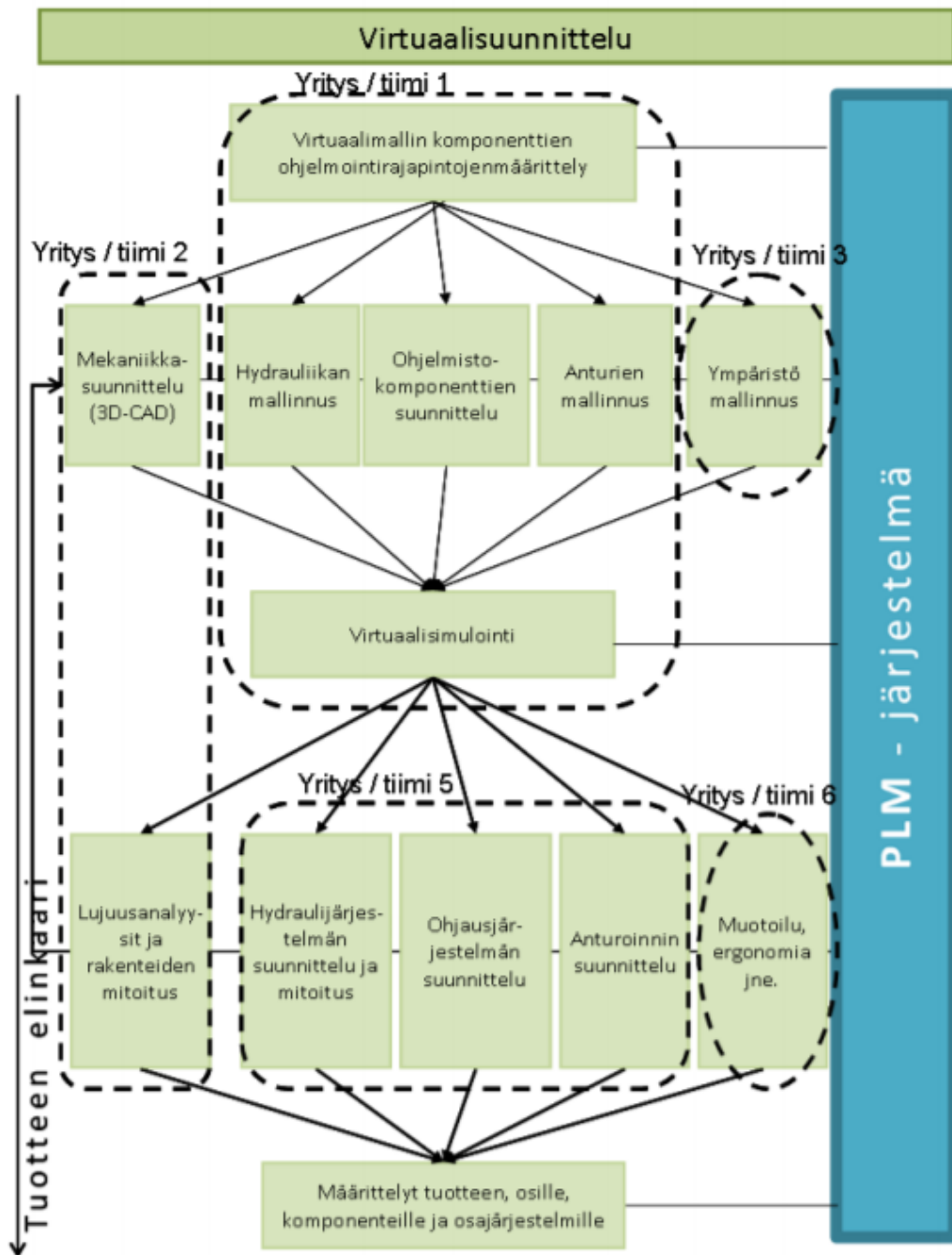
Suunnitteluprosessissa simulointipohjainen mallinnus on mukana tuotekehityksen alusta alkaen ja iteraatiokierrosten aikana malli rakentuu teknisesti toteuttavissa olevaksi lopputuotteeksi. Kuvassa 11 huomataan, että moniteknisen tuotteen modulaarinen virtuaalimalli muodostuu erilaisista suunnittelun osa-alueista. Erilliset osa-alueensa muodostavat mekaaninen-, hydraulinen-, anturi-, ympäristömallit ja ohjelmistokomponentit.

Mallinnusprojekteissa modulaarinen mallinnus on todettu joustavimmaksi ja helpommin hallittavaksi kuin, että yritys suunnittelisi ja kokoaisi yhden suuren virtuaalimallin koko projektista yrityksen omilla resursseillaan. Osa mallinuksesta voidaan hankkia palveluna yrityksen ulkopuolelta ja suunnittelijoiden eri sidosryhmien välinen tiedonvälitys voidaan hoitaa PLM-järjestelmän avulla (kuva 11).

Kuvassa 11 nähdään virtuaalisuunnittelun vastuualueiden jaoista eri sidosryhmien kesken verkostomaista tuotekehitystä hyödyntämällä. *Yritys/tiimi 1* tehtävä on koordinoida virtuaalimallia ja yhteensovittaa osamallit yhteen sekä suorittaa näiden simuloinnit ja analysoida tulokset. *Tiimi 2* vastuulla on tehdä mekaniikkasuunnittelu ja toimittaa tuotteen

3D -mallit simulointiympäristöön. *Tiimi 3* tehtävä on vastata toimintaympäristön mallinnuksesta esimerkiksi muodostamalla virtuaalinen pistepilven 3D -mallina ja viemällä se simulointiympäristöön.

Tiimi 5 tehtävänä on puolestaan suunnitella ja mitoittaa hydraulijärjestelmään liittyviä malleja ja toiminnallisuuksia. *Tiimi 6* vastaa tuotteen muotoilusta ja käyttäjäystävällisyydestä (Ahola, Hovila, Karhunen, Nevala, Schäfer & Nevala 2011).

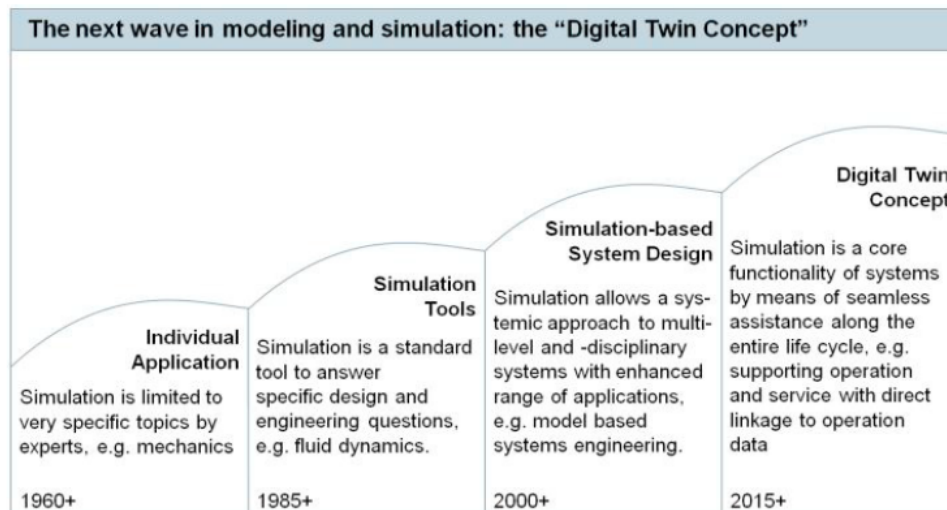


Kuva 11. Moniteknisen tuotteen verkottuneisuus (Ahola ym. 2011)

4.1 Simulointi suunnittelun yhteydessä

Vuosien saatossa simulointi on kehittynyt matemaattisista malleista tietokoneohjelmistoihin, joilla pyritään ratkaisemaan suunnittelijoiden eteen tulleita haasteita ja ongelmia. Tietokoneiden laskentatehon kasvu on tehostanut simulointeja edelleen, jonka johdosta pystytään simuloimaan kokonaisia järjestelmiä, kolmiulotteisia tuotteita ja niihin liittyviä laajempia toimilaittekokonaisuuksia. Digitaalisen kaksosen simulointi on seuraava vaihe mallinnuksen, simulaation ja optimoinnin teknisessä kehityksessä (kuva 12). Digitaalisen kaksosen mukaista simulointia pyritään hyödyntämään sen koko tuotteen elinkaaren ajan. (Rosen, Wichert, Lo & Betterhausen 2015).

Reaaliaikaisia simulointeja hyödynnetään mekatronisen tuotteen tai järjestelmän kehityksessä, koska tuotteen tai laitteen tuotekehitykseen suunniteltua aikaa, että sen markkinoille tuontiaikataulua (*time to market*) pystytään lyhentämään simulointien avulla.



Kuva 12. Simuloinnin kehityshistoria (Rosen ym. 2015)

Kuvan 13 perusteella reaaliaikainen simulointi voidaan jakaa HIL (*Hardware in the loop*), SIL (*Software in the loop*) ja *control prototyping* -malleihin.

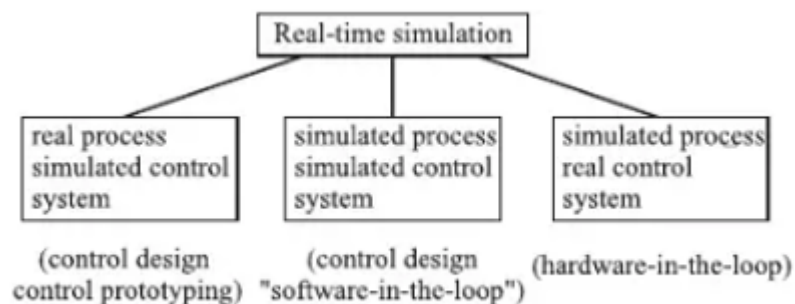
HIL- simuloinnissa käytetään hyväksi aitoja komponentteja, jotka ovat yhdistetty reaaliaikasisimuloiuihin elementteihin. Simuloinnissa automaatiojärjestelmän laitteisto ja ohjelmisto ovat todellisia, jolloin sillä kyetään varmentamaan automaatiojärjestelmän toiminta. HIL-simuloinnin ohjausprosessi sisältää toimilaitteet, fyysisen prosessin ja anturit, jotka voidaan sisällyttää joko reaaliaikasisimuloiuihin taikka todellisiin komponentteihin.

Vaikeasti mallinnettavat komponentit esimerkiksi toimilaitteet ovat todellisia ja kaikki muut ovat simulointikäyttöisiä (Bishop 2015, 2.13-2.15).

SIL-simuloinnissa hyödynnetään puolestaan ainoastaan virtuaalista ympäristöä ilman fyysisiä laitteita. Virtuaalisen ympäristössä ohjaukseen tarkoitettut sovellukset esimerkiksi PLC-ohjelmat ovat yhteydessä IP/TCP yhteyden avulla mekatroniseen laitteeseen ja ohjelmistoemulaattorin välille. Ohjelmistoemulaattorilla tarkoitetaan esimerkiksi tässä työssä käytettyä virtuaalista simulointiympäristö Siemens NX Mechatronics Concept Designer -ohjelmistoa.

SIL-simulointi ei edellytä fyysisiä komponenttia, mutta haasteena on oikeanlaisen ohjelmoitavan logiikan ohjelmistoversion saatavuudet. Tämän vuoksi on hyvä tiedostaa, että simuloitava PLC -ohjelmistoversio tai itse ohjelma eroaa tehtaan omasta PLC -ohjelmistoversiosta, voi tämä antaa epäluotettavia simulointituloksia (Chryssolouris, Makris & Michalos 2012).

HIL- ja SIL -simuloinnin lisäksi reaaliaikaisiin simulointeihin kuuluu *Control prototyping*, joka vaatii fyysisten laitteiden lisäksi todellisen prosessin, jonka päätarkoituksena on testata ja optimoida ohjausjärjestelmän toiminnallisuutta (Isermann & Schaffnits 2000).



Kuva 13. Reaaliaikaisten simulointien jako (Bishop 2015)

Virtuaalinen simulointi suoritetaan tietokoneavusteisesti suunnittelun jälkeen, jossa osamallit yhdistetään yhteen yhdeksi isoksi kokonaissimulointimalliksi eli minikappalesysteemimalliksi, johon on sisällytetty hydraulikka-, ohjelmisto- ja anturimallit toisiinsa. Osamallien yhteensopivuus määritellään projektin alussa käyttämällä yhteensopivia mallinnusohjelmistoja ja määrittelemällä osamallien rajapinnat ennen niiden toteuttamista.

Useimmissa nykyaikaisissa CAD-ohjelmistoista löytyvät tarvittavat työkalut mekanismien mallintamiseen ja ohjausten simuloimiseen tiettyjen reunaehtoja puitteissa. (Ahola ym. 2011).

HIL -ja SIL-simulointien avulla pystytään testaamaan osa prototyypistä virtuaalisena simulointimallina. HIL-simuloinnin avulla tehdään mekaniikkatestausta, kun SIL-simuloinnilla keskitytään enemmän ohjainlaitteiden hardwaren testaamiseen. HIL- ja SIL-simulointitestaamisella helpotetaan ohjelmistojen ja ohjaimien testaamista, sekä parannetaan testausturvallisuutta, kun mahdolliset ilmenevät virhetilanteet eivät aiheuta prototyyppien rikkoutumista tai vaaratilanteita. Haasteena simulointilähtöisessä testaamisessa on virtuaalisen mallin luotettavuus, mallinnusongelmat esimerkiksi todenmukaisten lämpötila anturisignaalien tuottaminen simulointimallista. HIL- ja SIL-simulointitestaamisen jälkeen voidaan kokoonpanna todellinen prototyyppi (Ahola ym. 2011 94-97).

Tässä opinnäytetyössä käytetään virtuaaliseen simulointiin Siemens NX Mechatronics Concept Designer -ohjelmistoa ja TIA Portal -ohjelmaa automaatiologiikan tekemiseen ja testaamiseen.

4.2 Perinteinen käyttöönotto

Perinteisellä käyttöönotolla tarkoitetaan koneen tai laitteen ensimmäistä käynnistystä ja sitä edeltäviä laitteeseen liittyviä toimenpiteitä ja tarkastuksia.

Perinteinen käyttöönotto vaatii fyysisen järjestelmän ja tuotteen olemassaolon paikalla koottuna asiakkaan määrittämälle paikalle ennen kuin voidaan aloittaa itse käyttöönotto ja tarkastus. Käyttöönotto on suunnitteluprosessin viimeisiä vaiheita ennen kuin asiakas vastaanottaa tuotteen käyttöönsä.

Perinteinen käyttöönoton prosessi on vesiputousmallin tyylinen, jossa vaiheet etenevät lineaarisesta toiseen, ilman iteratiivista takaisinkytkentää prosessinvaiheissa taaksepäin. Tämän vuoksi testausta kyetään tekemään vain prosessin lopussa käyttöönoton aikana. Testauksen aikana ilmenneet ongelmat ovat pääasiassa kahdenlaisia, bugit ohjelmistoissa ja kehitysvaiheessa ilmenneiden ongelmien kertaantumiset käyttöönotoissa.

Zäh ja Wunsch:n (2005) mukaan käyttöönotto vie noin 15-20% koko projektiin suunnitellusta ajasta, josta 90% siitä kuluu sähköistykseen ja ohjausjärjestelmien (PLC) käyt-

töönottoihin. Tästä PLC:n käyttöönotosta menee jopa noin 70% ohjelmistobugien selvittämiseen (kuva 14). Ohjelmistobugien testaus on yleisesti aikaa vievää, joka voi pahimmillaan rikkoa fyysistä laitteistoa. Lisäksi käyttöönotot on yleisesti tiukasti aikataulutettu, jonka vuoksi aikataulun venyminen voi tuoda taloudellisia sanktioita yrityksille. Tämän vuoksi on suotavaa tehdä testaus jo projektin aikaisemmissa vaiheissa esimerkiksi virtuaalisena käyttöönottona (Liu, Suchhold, & Diedrich 2012).



Kuva 14. Ohjausjärjestelmien osuus projektin viivästyisestä (Wünsch 2008)

Toinen merkittävä ilmenevä ongelma käyttöönotoissa on kehitysvaiheissa ilmenneiden virheiden kertaantuminen laitteiston tai projektin käyttöönotto vaiheessa. Tämä voi aiheuttaa projektin budjettiin merkittäviä lisäkustannuksia. Kuvasta 15 nähdään, että mitä aikaisemmassa vaiheessa projektia virhe havaitaan, sen edullisemmaksi se tulee yritykselle (Liu ym. 2012).



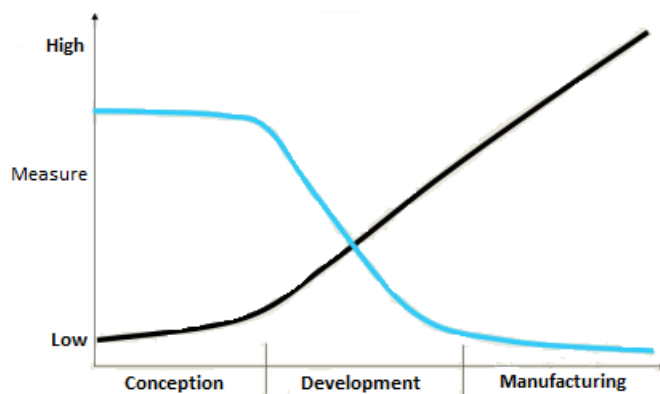
Kuva 15. Virheiden korjaamisesta aiheutuvat kustannukset sen havaitsemisajan mukaan (Kiefer 2007)

4.3 Virtuaalinen käyttöönotto

Virtuaalinen käyttöönotto eroaa perinteisestä käyttöönotosta sillä, että automaatiojärjestelmän ja mekaniikkasuunnitelman prosessien vaiheet testataan virtuaalisella mallilla.

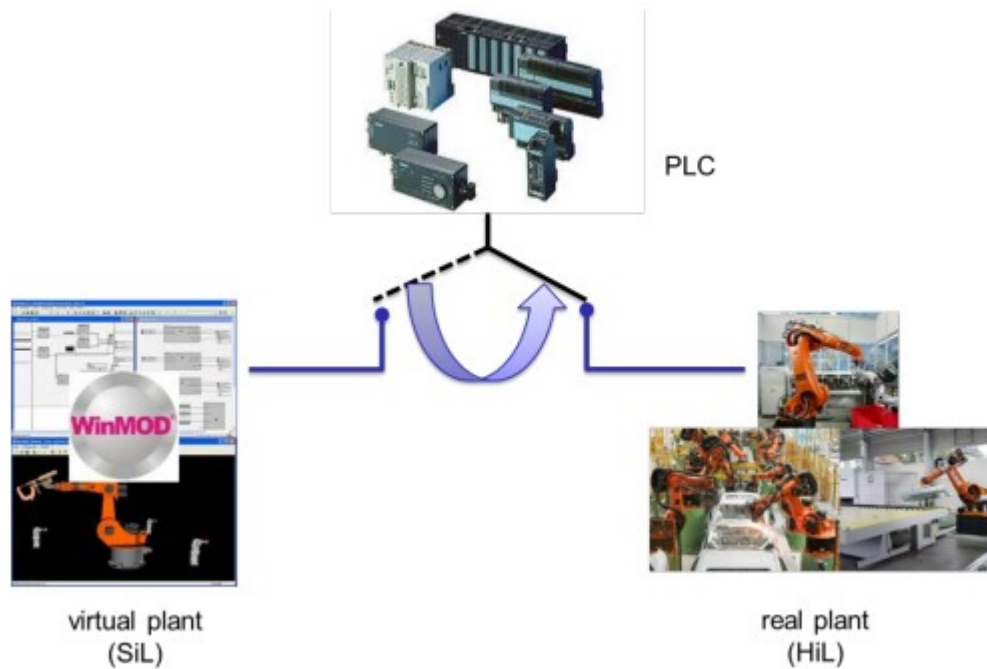
Virtuaalisella käyttöönotolla pyritään todentamaan mallien ja järjestelmien toiminnallisuudet ja niistä ilmenevät virheet havaittua entistä aikaisemmassa vaiheessa, ennen laitteen tai yksikön lopullista kokoamista. Tämän ansiosta tuotteen valmistus ja suunnittelukustannukset vähenevät (kuva 16), suunnitteluprosessi nopeutuu sekä kyetään näyttämään asiakkaalle tilaamansa tuotteen tai sen yksikön toiminnallisuus virtuaalisesti ennen sen toimittamista.

Tuotteen tai sen yksikkökokonaisuuden fysikaalisia ilmiöitä voidaan tarkastella virtuaalisesti tarkemmin, esimerkiksi kappaleen asentoa, nopeutta ja pyörimistä pystytään tarkastelemaan simuloinnin aikana. Tämän lisäksi virtuaalisessa käyttöönotossa tuotteen tai sen yksikkökokonaisuus on ohjattavissa ja siihen vaikuttavat suoritusarvot säädettävissä ennen kuin valmistusmateriaaleihin on sidottu yhtään pääomaa (Liu, Suchhold & Diedrich 2012; Siemens: pikakäyttöopas 2010).



Kuva 16. Suunnittelukustannusten vaikutus tuotekehityksessä (Siemens: pikakäyttöopas 2010, muokattu)

Kuvassa 17 on pyritty havainnollistamaan virtuaalisen käyttöönoton toiminnallisuutta, jossa PLC -ohjelmistolla tehty logiikkaohjelmalla voidaan ohjata joko virtuaalista tai todellista tehdasta. Päätavoitteena on kuitenkin simuloida, testata ja validoida ohjelmiston toiminnallisuudet virtuaalisesti ensin ennen sen tuotantoon siirtämistä.



Kuva 17. Virtuaalisella käyttöönotolla järjestelmän toiminnallisuudet kyetään siirtämään todelliseen ympäristöön (Liu, Suchhold & Diedrich. 2012)

4.4 Digitaalinen kaksonen

Globaalin valmistavan teollisuuden tilanne, vaatimukset ja nykypäivän nopeat muutokset ajavat yritykset kehittämään tuotteitaan entistä nopeammin ja kustannustehokkaammin kuin aikaisemmin. Nykypäivän valmistus- ja tuotantojärjestelmien toimintaa pyritään autonomisoimaan entistä enemmän. Autonomisella järjestelmillä tarkoitetaan älykkäitä koneita, jotka kykenevät suoriutumaan vaativammista tehtävistä entistä itsenäisemmin ilman ihmisen työpanosta. Tähän päästäkseen järjestelmälle luodaan digitaalinen kaksonen (*Digital Twin*), jolla voidaan tarkoittaa fyysisen koneen, laitteen tai kokonaisen tehtaan digitaalista kopiota virtuaalisessa muodossa. Virtuaalisella mallilla ja sen reaali maailman fyysisellä vastineella välillä tulee olla yhteys, jonka avulla informaatiota voidaan siirtää molempiin suuntiin (Rosen, Wichert, Lo & Betterhausen 2015; Boschert, Heinrich & Rosen 2018).

Digitaalisen kaksosen etuna on, että sen avulla tuote tai sen valmistusprosessi voidaan virtuaalisesti testata, optimoida ja ottaa käyttöön ennen laitteen fyysisestä kasaamista. Virtuaalisella simuloinnilla ja siinä digitaalista kaksosta hyödyntämällä laitteen tai tuotantolinjaston laatu poikkeamat tai suunnitteluvirheet voidaan havaita entistä aikaisemmassa vaiheessa.

Tämä ansiosta tuotekehitystä kyetään nopeuttamaan sekä tuottamaan hyödyllisempää informaatiota strategisten päätösten tuoksi. Tämän lisäksi, kun digitaalinen kaksonen on yhdistetty fyysiseen tuotantolaitokseen tai sen laitteeseen, voidaan siihen suunniteltuja modifikaatioita testata ensin virtuaalisesti ja tämän jälkeen viedä testatut muutokset tuotantoon. Näiden pohjalta kyetään tehostamaan ja sujuvoittamaan valmistusteollisuuden tuottavuutta (Siemens: Digitaalinen kaksonen simuloi täydelliseksi n.d).

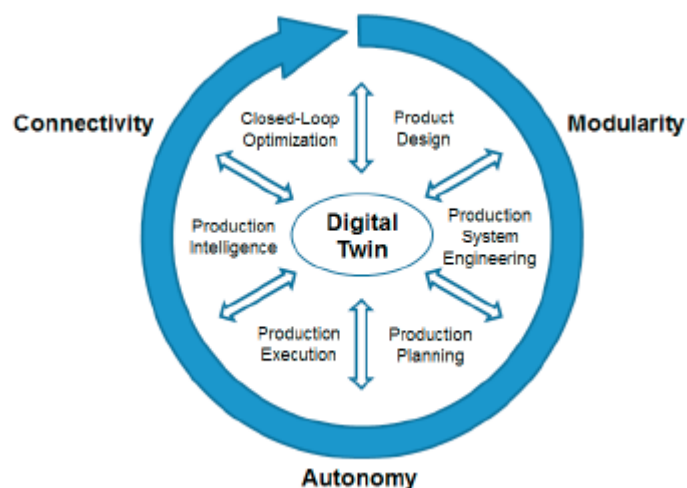
Tuotteen elinkaaren näkökulmasta digitaaliseen kaksonen pitää sisällään modulaarisuuden suunnittelussa (*Modularity*), autonomian tuotantojärjestelmissä (*Autonomy*) sekä näiden välillä vallitsevan yhdistettävyyden (*Connectivity*) (kuva 18).

Modulaarisella eli verkostomaisella suunnittelulla ja tuotantomoduuleilla saavutetaan parempi tehokkuus erilaisten tuotantojärjestelmien suunnittelussa.

Autonomian avulla tuotantojärjestelmät pystyvät vastaamaan yllättäviin muutoksiin ennalta ohjelmoitulla tavalla ilman, että järjestelmiä täytyy uudelleen konfiguroida.

Yhdistettävyyys IOT:n avulla kyetään välittämään tietoa prosessien välillä ja tehostamaan näiden välistä suorituskykyä.

Tämä kolminaisuus on elinehto sille, että digitaalista kaksosta hyödynnetään oikeaoppisesti, eikä vain yhtä osa-aluetta kolmesta (Rosen, Wichert, Lo & Betterhausen 2015; Boschert, Heinrich & Rosen 2018).

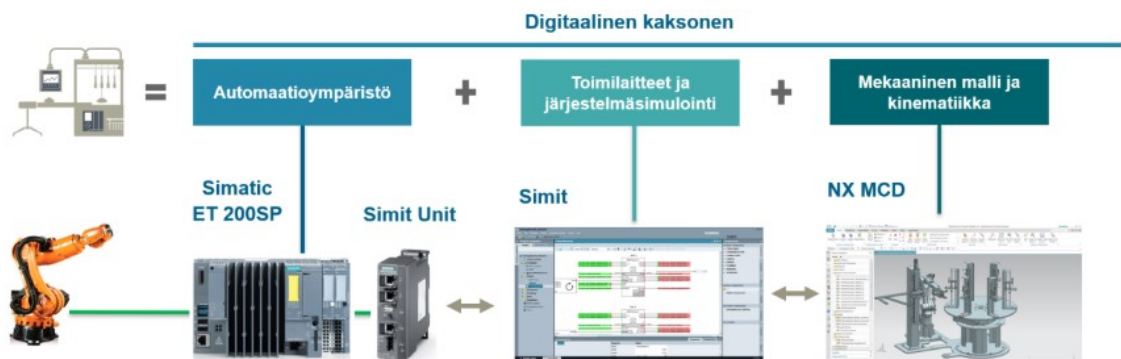


Kuva 18. Digitaalinen kaksonen tuotteen koko elinkaareissa (Rosen, Wichert, Lo & Betterhausen 2015)

Siemens tarjoaa digitaalisen kaksosen virtuaaliseen käyttöönottoon ja sen simuloimiseen omaa tuotepakettiaan (kuva 19). Paketti koostuu automaatioympäristön, toimilaitte -ja järjestelmäsimulointi paketeista sekä mekaniikka -ja kinematiikasta päämoduuleista.

Toimilaitte ja järjestelmäsimulointi moduulin ohjelmistona on käytössä SIMIT, joka sisältää työkalut Profinet-väylän, turvan, toimilaitteiden (venttiilit, ja käytöt) ja erilaisten fysikaalisten suureiden esimerkiksi paineilman ja lämpötilan simuloimiseen. SIMIT ohjelmiston ympäristöllä pystytään testaamaan virhetilanteita ja analysoimaan laitteen toimintaa. Tämän lisäksi ohjelmistolla on mahdollista simuloida prosesseja sekä yhdistää SIMIT-ohjelmalla tehty automaatiologiikka MCD -moduuliin.

Mekaniikka ja kinematiikka moduulin ohjelmistona on kolmiulotteinen NX modulaarinen suunnitteluohjelmisto, johon asennettu mekaniikan simuloimiseen MCD (Mechanics Concept Designer) ympäristö (Siemens: kilpailuetua virtuaalisella käyttöönotolla n.d).



Kuva 19. Siemensin tarjoamat ohjelmistot digitaalisen kaksosen hyödyntämiseen (Siemens 2019)

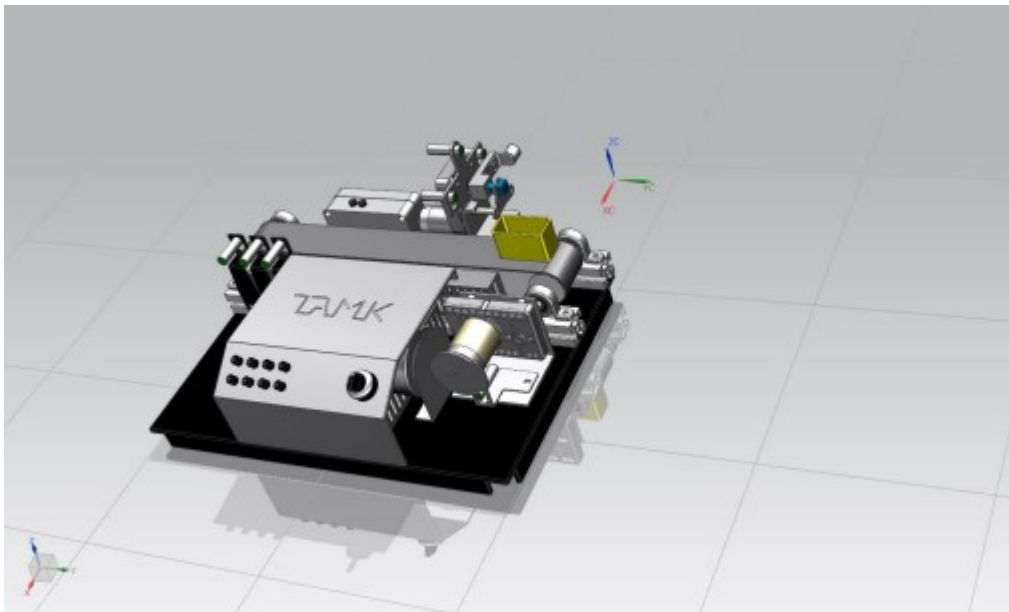
5 KÄYTETTÄVÄT OHJELMISTOT

5.1 Siemens NX -suunnitteluohjelmisto

Siemens PLM softwarin kehittämä NX on tarkoitettu tietokoneavusteiseen suunnitteluun (CAD, CAE) ja tietokoneavusteiseen valmistamiseen (CAM). Tämän lisäksi NX -ohjelmisto voidaan integroida suoraan PDM-tuotetiedonhallinta ohjelmaan esimerkiksi Siemensin omaan Teamcenteriin.

NX on laajalle levinnyt 3D-suunnitteluohjelmisto, jolla suunnittelun ja valmistuksen lisäksi pystytään suorittamaan simulointeja erilaisille NC työstökoneen ohjelmille sekä tekemään analyysejä, jotka koskevat lujuuksia (FEM), kinemaattisia liikkeitä, lämpötiloja ja virtauksia (CFD).

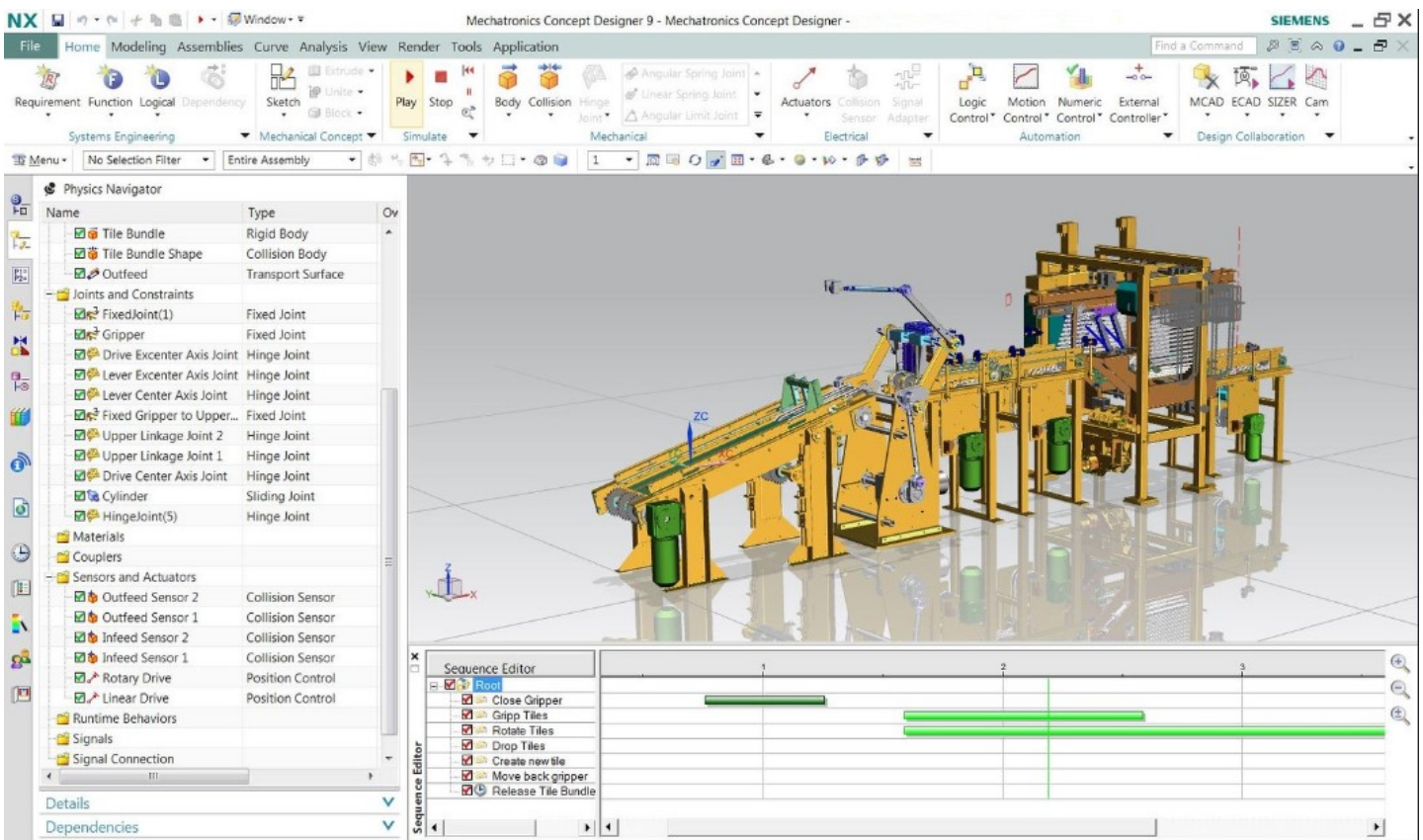
Modulaarisuuden ansiosta NX-ohjelmaan saadaan helposti lisättyä tarvittavat lisäohjelmat ja ominaisuudet, esimerkiksi mekatronisen tuotteen tai järjestelmän virtuaaliseen simuloimiseen ja käyttöönottoon. Tässä opinnäytetyössä käsitellään NX Mechatronics Concept Designer (MCD) lisäosan toiminnallisuuksia (Siemens 2019).



Kuva 20. Siemens NX -ohjelmistolla mallinnettu kuljetin (Tampereen ammattikorkeakoulu 2018)

5.2 Mechatronics Concept Designer

Mechatronics Concept Designer (MCD) on Siemens NX -suunnitteluohjelmistoon sisältyvä lisäosa. MCD on tarkoitettu vuorovaikutteiseen suunnitteluun ja sillä voidaan simuloida monimutkaisia mekatronisia järjestelmiä ja niiden malleja suunnittelusta lähtien (kuva 21). MCD avulla saadaan kehitettyä mekatronisen järjestelmän suunnitteluprosessia ja valmista järjestelmää voidaan testata virtuaalisesti ennen sen lopullista valmistamista. MCD toimii sähkömekaanisen suunnittelun periaatteita käyttäen. Suunniteltuun 3D-malliin voidaan luoda ja yhdistää anturi- ja toimilaitte tietoja, PLC logiikkaohjelmia sekä ulkoisten ohjelmien välillä kulkevaa tiedonvälitystä. Tämä helpottaa mekaniikka-, sähkö- ja automaatiosuunnittelijat näkemään suunnitelmiansa toimivuutta, parantamaan yhteistyötä toisiensa välillä ja mekatronisen laitteen markkinoille saaminen nopeutuu suunnittelun osalta (Siemens PLM Software; Siemens: pikakäyttöopas 2010).



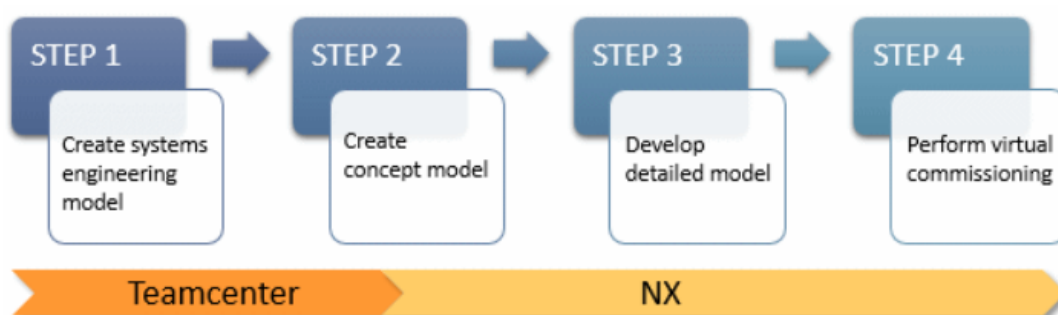
Kuva 21. MCD avulla pystytään esittämään kokonaisen tuotantolinjan toiminta (Ada n.d).

Kuvasta 22 havaitaan MCD -moduulin työnkulku ja vaiheet. Ensimmäisenä vaiheena on luoda järjestämästä malli yleensä PDM-järjestelmään ja sieltä MCD -ohjelmistoon osana suunnitteluprosessia.

Toisena vaiheena on luoda konseptimalli, jolloin luodaan mallista kokoonpano, johon lisätään fyysiset ominaisuudet ja anturit. Tähän vaiheeseen kuuluu myös kokoonpanomallissa olevien liikkeiden ja ehtojen määrittäminen ja tarvittaessa testaaminen.

Kolmantena vaiheena on määrittellä yksityiskohtaisempi mallin, johon voidaan sisällyttää geometrian muodot, sisäiset signaalit, realistiset liikealueet sekä yhdistää ulkoisien ohjelmistojen rajapinnat.

Viimeisenä vaiheena on tehdä virtuaalinen käyttöönotto ja dokumentoida prosessi (Siemens: Learning Advantage n.d).



Kuva 12. MCD:n moduulin työnkulku (Siemens: Learning Advantage n.d)

5.2.1 MCD-simuloinnin rajoitukset

MCD -simuloimiseen sisältyy tiettyjä rajoituksia, jotka käyttäjän on syytä huomioida ohjelmaa käyttäessä. Ohjelman simulointi kykenee ratkaisemaan yksinkertaistettuja yhtälöitä, eikä se kykene analysoimaan monikappaleisia simulointeja kuten taajuusanalyyseja. Lisäksi MCD:tä ei tule käyttää monimutkaisiin törmäysanalyyseihin kappaleisiin, jotka ovat monikulmaisia.

MCD ei tue joustavien materiaalien simulointeja eikä muodonmuutoksia kappaleissa pysyttyä havaitsemaan. (Siemens: pikakäyttöopas 2010)

5.3 TIA Portal

PLC ohjelmointiin tarkoitettu TIA Portal on Siemensin tuottama ohjelmisto automaatio-suunnitteluun. Siemensin TIA (Totally Integrated Automation) on täysin integroitu automaatio ohjelmisto, johon on yhdistetty logiikkaohjelmoinnin (SIMATIC STEP 7), käyttöliittymäsuunnittelun (Simatic WinCC) ja taajuusmuuttajien (Sinamics StartDrive) ohjelmoinnit. Näiden kolmen yhdistettyjen ominaisuuksien ansiosta Tia Portal on monipuolinen ja tehokas työkalu automaatio-suunnitteluun, kun sillä kyetään suorittamaan suunnittelu- ja tuotantoprosessit kokonaisuudelle tuotantoketjulle.

Automaatio-suunnittelun konfigurointia, diagnostiikkaa ja ylläpitoa voidaan toteuttaa TIA Portal ohjelmistolla yksinään. Ohjelmiston optimaalisella käytöllä saadaan mahdollistettua tehokkaampi työskentely sekä parannettua tuottavuutta ja kilpailukykyä. (Siemens: TIA portal n.d.)

SIMATIC STEP 7 on logiikkaohjelmointiin käytetty ohjelmisto, jonka tehtävä on kattaa tarpeet yksittäisestä koneenohjauksesta aina laajoihin turvatoimintoihin sisältyviin järjestelmiin. Ohjelmiston tuotteisiin sisältyy kolme erilaista versiota.

Basic, Professional ja Safety versiot käsittävät kaikkiaan S7-300, S7-400 S7-1200, S7-1500 ja WinAC- logiikkaohjaimet. (Siemens: SIMATIC STEP 7 n.d.)

SIMATIC WinCC on tarkoitettu visuaalisten käyttöliittymien luomiseen. Ohjelmistolla voidaan toteuttaa visualisointisovellukset pienistä HMI paneeleista aina graafisesti suurempiin valvomosovelluksiin esimerkiksi tehtaiden valvomoiden käyttöliittymät voidaan toteuttaa tällä tavalla. (Siemens: SIMATIC WinCC TIA n.d)

SINAMICS StartDrive ohjelmistolla integroidaan SINAMICS-taajuusmuuttajat automaatio järjestelmään sekä näiden käyttöönotto tehdään TIA Portal ohjelmistoa käyttämällä. Ohjelmisto tukee taajuusmuuttajien lisäksi pumppu-, tuuletin ja kompressorisovelluksia. (Siemens: SINAMICS StartDrive -ohjelmisto n.d.)

Project tree

5K00BH32-3005-Harjoitus-Kuljetin-v1_V13_SP1 ▸ Kuljetin-PLC [CPU 314C-2 PN/DP] ▸ Program blocks ▸ Main [OB1]

Devices

- 5K00BH32-3005-Harjoitus-Kuljetin-v1_V13_SP1
 - Add new device
 - Devices & networks
 - Kuljetin-PLC [CPU 314C-2 PN/DP]
 - Device configuration
 - Online & diagnostics
 - Program blocks
 - Add new block
 - Main [OB1]
 - System blocks
 - Technology objects
 - External source files
 - PLC tags
 - PLC data types
 - Watch and force tables
 - Online backups
 - Device proxy data
 - Program info
 - PLC alarms
 - Text lists
 - Local modules
 - HMI_1 [KTP600 Basic color PN]

Main

	Name	Data type	Offset	Default value	Comment
1	Temp				
2	OB1_EV_CLASS	Byte	...		Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
3	OB1_SCAN_1	Byte	...		1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)

& >=1

Block title: "Main Program Sweep (Cycle)"

Kuljetin harjoitus

Network 1: Siirto tunnistukseen

SET: Kappale saa lähtää liikkeelle, kun se on paikalla ja kytkin S1 käännetään "1" asentoon...

Kuva 23. TIA Portalin käyttöliittymä (Pakkala,2019).

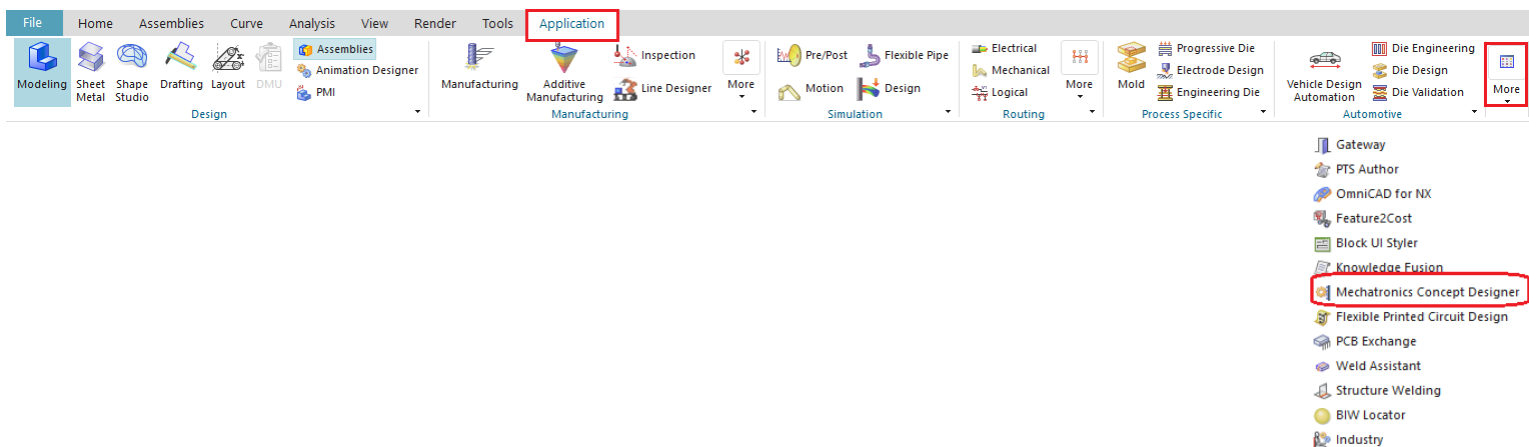
6 MECHATRONICS CONCEPT DESIGNER TOIMINNALLISUUDET

Seuraavaksi esitellään MCD:n yleisimmät työkalut ja niiden toimintaperiaatteet. Osuuden materiaali on koottu Siemens Learning Advantage –palvelun internetsivuilta. Learning Advantage on Siemensin sähköinen oppimispalvelu, jonka avulla käyttäjät voivat kehittää ja kasvattaa osaamistaan Siemensin tarjoamillaan materiaaleillaan ja PLM –ohjelmisto kursseillaan. Pääsyn oppimismateriaaleihin mahdollisesti Tampereen ammattikorkeakoulu.

6.1 Aloittaminen

Suunniteltu 3D-malli saadaan vietyä MCD moduuliin *Application* välilehdeltä ja napsauttamalla *More* valikon alta *Mechatronics Concept Designer* (kuva 24).

Ohjelmiston aloittamisen jälkeen määritellään mallille halutut liikkeet ja toiminnallisuudet. Mallin toimintojen määrittämisessä on suositeltavaa valita vain mallin toiminnalle tarpeelliset komponentit. Mikäli kaikki osat määritellään tarkasti, tulee mallista helposti raskas ja hidas mallia simuloimassa.



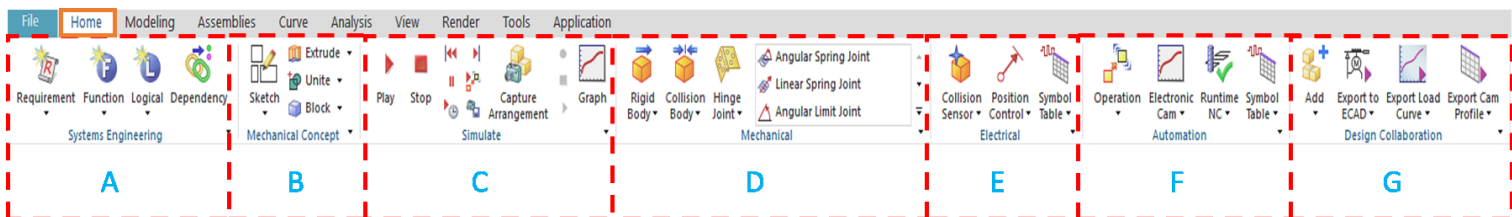
Kuva 24. 3D-ohjelman käyttövalikot

6.1.1 Aloitusvalikot

Käynnistämisen jälkeen aukeaa MCD:n lisämoodulin työkalut (kuva 25), jotka on jaettu niiden toiminnallisuuden mukaan vastaaviin osakokonaisuuksiin.

Kategoriat ovat jaettu seitsämään seuraavaan kohtaan:

- **Systems engineering:** Systeemitekniikan työkalut, joita voidaan käyttää toiminnallisten -ja loogisten mallien suunnitteluun (kuva 25A)
- **Mechanical Concept:** 3D-mallintamisen työkalut mallien luontiin ja muokaamiseen (kuva 25B).
- **Simulate:** Sisältää työkalut toiminnallisuuden määrittämiseen. Tämän kategorian toiminnot ovat MCD:n käytössä oleellisia (kuva 25C).
- **Mechanical:** Sisältää työkalut 3D-malleihin liittyvien fysikaalisten ominaisuuksien määrittämiseen. Nivelet, liitokset, pintojen sekä kappaleiden törmäysominaisuuksien määrittäminen kuuluvat tähän kategoriaan. Tämän kategorian toiminnallisuudet ovat MCD:n käytössä oleellisia (kuva 25D).
- **Electrical:** Sisältää työkalut sähköisten ominaisuuksien määrittämiseen. Antureiden ja toimilaitteiden määrittäminen tehdään tässä kategoriassa (kuva 25E).
- **Automation:** Sisältää työkalut kinematiikan ja fysikaalisten ominaisuuksien ohjaukseen simulaation aikana. Lisäksi ulkoisen automaatiojärjestelmän yhdistämisen työkalut löytyvät tästä kategoriasta (kuva 25F).
- **Design Collaboration:** Sisältää työkalut ulkoisten ohjelmistojen yhteenliittämiseen ja lisäämiseen MCD:n kanssa (kuva 25G).



Kuva 25. Yläpalkin työkalut kuvattuna toiminnallisuuden mukaan.

Kuvassa 26 on esitetty ohjelmiston sivupalkin työkalut.



Kuva 26. Sivupalkin työkalut.

System Navigator: Toiminnallisuus, jolla tarkastellaan kokoonpanon toiminnallisia ja loogisia malleja.



Kuva 27. System Navigator-työkalu

Physics Navigator: Toiminnallisuus, josta nähdään kokoonpanon fyysiset ja loogiset ominaisuudet.



Kuva 28. Physics Navigator-työkalu

Runtime Inspector: Toiminnallisuus, jolla nähdään kappaleiden fysikaaliset ominaisuudet simulaation aikana.



Kuva 29. Runtime inspector-työkalu

Runtime Expression: Toiminnallisuus, jolla nähdään kappaleiden liikkeisiin sekä viitauksiin liittyvät suhteet ja ominaisuudet.



Kuva 30. Runtime Expression työkalu

Assembly Navigator: Toiminnallisuus, josta nähdään kokoonpano ja sen eri osat sekä näiden muokkaustoiminnot.



Kuva 31. Assembly Navigator-työkalu

Sequence Editor: Toiminnallisuus, jolla säädetään aika- ja tapahtumatyyppisiä toimintoja.



Kuva 32. Sequence Editor-työkalu

6.2 Simuloinnin aloittaminen

3D-mallin suunnittelun jälkeen määritellään kokoonpanon toiminnallisuus ja tämän jälkeen aloitetaan simulointi. MCD sisältää useita työkaluja, jolla voidaan määrittellä ja kuvata kappaleen käyttäytymistä sen eri tilanteissa. Tärkeimmät ohjelmistosta löytyvät työkalut ovat liikkeen, nopeuden, suunnan sekä pintakosketuksen määrittämiseen.

Kokoonpanon simulointi on suositeltava aloittaa suunnitteluvaatimusten määrittämisellä, esimerkiksi miettimällä, kuinka järjestelmän kuuluu toimia ja minkälaisia vaatimuksia se sisältää. Määritysten jälkeen voidaan luoda funktio -ja looginen malli, josta tulee selvitä

järjestelmän toiminnallisuudet ja niiden prioriteettijärjestys sekä niiden välillä ilmene yhteys. Tämän jälkeen määritellään kokoonpanomallin mekaaniset toiminnallisuudet ja lisätään siihen tarvittaessa liike ja dynamiikka. Mekaanisten määritysten malliin voidaan lisäksi lisätä fyysiset ja nopeudelliset rajoitukset sekä aikaperusteiset toiminnot esimerkiksi toimilaitteiden ohjaukset. Kokoonpanoon voidaan lisätä anturit ja niiden kytkentä sekä määrittää niille tapahtumat. Käynnistyvä toiminnallisuus voi esimerkiksi olla kuljetinhinnan pysähtyminen, kun kuljettimeen määritelty anturi saa tunnistusherätteen.

Lopuksi simulaatioon voidaan yhdistää ulkoinen PLC automaatio-ohjelma ja testata OPC-yhteys ennen simuloimisen aloittamista.

Liitteestä 1. Löytyy tarkempi työnkulku ja muistilista alkaen määrittämisestä simuloimisen aloittamiseen.

6.2.1 Työkalut kappaleiden määrittämiseen

Rigid body: Työkalun avulla (kuva 33) valitaan mallista pelkästään liikuteltavat osat. Osan määrittelyn jälkeen se saa fysikaaliset ominaisuudet (paikka, massa, inertia ja nopeuden).



Kuva 33. Rigid Body-työkalun avulla valitaan liikuteltavat osat

Object Source: Työkalun avulla (kuva 34) luodaan useita kappaleita yhdestä mallista. Tätä toiminnallisuutta voidaan käyttää esimerkiksi liukuhihnalla matkaavien kappaleiden monistamiseen.



Kuva 34. Object Source-työkalun avulla monistetaan kappaleita

Object Sink: Työkalun avulla (kuva 35) poistetaan *Object Source*:lla monistetut kappaleet tai niiden osat. Tätä toiminnallisuutta voidaan käyttää yhdessä antureiden määrittämisen kanssa, kun kappale kulkeutuu liukuhihnalla määräsemaansa, voidaan se poistaa liukuhihnalta *Object Sink*-työkalun avulla.

Antureiden määrittäminen tapahtuu *Collision Sensor*-työkalun avulla, jota käsitellään myöhempanä.



Kuva 35. Object Sink-työkalun avulla poistetaan kappaleita

Transport surface: Työkalun avulla (kuva 36) voidaan määrittää malli olemassa olevaksi. Määritelty pinta muuttuu kuljetinhihnaksi, jolle voidaan antaa suunta- ja nopeusvektori.

Liukuhinnan määrittämiseen on kuitenkin otettava huomioon kuljetettava kappale, jonka pohjalle täytyy määrittää *Collision Body*-työkalun avulla kosketuspinta kuljetinhihnaa vasten ja määrittellä kuljetettava kappale liikkuvaksi *Rigid Body*-työkalun avulla. Tämän jälkeen kuljetinhihna ja sen päällä liikkuva kappale liikkuvat toisiinsa nähden.



Kuva 36. Transport surface-työkalun avulla voidaan luoda kuljetin

Prevent collision: Työkalun avulla (kuva 37) määritellään kaksi kappaletta, joita estetään törmäämästä toisiinsa. Tätä toiminnallisuutta voidaan käyttää esimerkiksi liukuhinnan päällä kulkevien kappaleiden lajittelemiseen yhdessä antureiden määrittämisen *Collision Sensor*-työkalun avulla.



Kuva 37. Prevent Collision-työkalu estää kappaleita törmäämästä toisiinsa

Change material: Työkalun avulla (kuva 38) määritellään kaksi kappaletta ja niiden törmäysominaisuudet. Tarkempi määrittely saavutetaan yhdessä *Collision material*-työkalun avulla, jolla mahdollistetaan esimerkiksi erilaisten anturityyppien luomisen simulointimalliin.



Kuva 38. Change material-työkalun avulla törmäysominaisuudet

Collision Material: Työkalun avulla (kuva 39) määritellään komponentin materiaali ja sen käytös törmäystilanteessa.



Kuva 39. Collision material-työkalun avulla määritellään komponentin materiaali

Spring Damper: Työkalun avulla (kuva 40) pystytään määrittämään jousen ominaisuudet. Määrittämisellä voidaan säätää jousivoimaa ja vääntömomenttia sekä näiden sijaintia.



Kuva 40. Spring Damper-työkalun avulla määritellään jousen parametrejä

Force/Torque Control: Työkalun avulla (kuva 41) kyetään asettamaan voima tai vääntömomentti niveleen. Nivelkiinnikkeisiin käytetään positiivista arvoa, kun liitospohjiin negatiivisia arvoja.



Kuva 41. Force/Torque Control-työkalun avulla säädetään niveliin vaikuttavia suureita

Display Charger: Työkalun avulla (kuva 42) pystytään säätämään liikkuvan kappaleen asetuksia simulaation aikana. Työkalu voidaan liittää *collision sensor*-, *rigid body* -työkalujen sekä geometrian tueksi säätämään ominaisuuksia, esimerkiksi värejä ja näkyvyyttä.



Kuva 42. Display Changer-työkalun avulla pyritään helpottamaan kappaleiden määrittämistä.

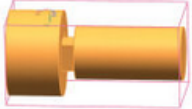
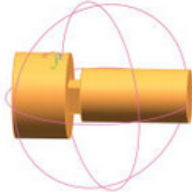
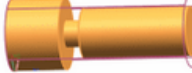

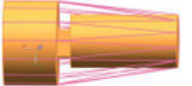
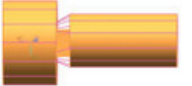

Collision Body: Työkalun avulla (kuva 43) määritellään kappaleiden törmäyspinta ja liikkeen suunta. Tämän toiminnallisuuden määrittäminen MCD:ssä on oleellinen osa simuloimisen toteuttamista. Kappaleiden törmäysominaisuudet on syytä määrittää välttääkseen niiden tippumisen ulos mallista, koska vain määritellyt kappaleet vaikuttavat toisiinsa. Taulukossa 1 on esitetty *Collision body*-työkalun sisältämät erilaiset törmäysmuodot.



Kuva 43. Collision body-työkalulla valitaan kappaleen törmäyspinnat.

Törmäysmuotojen avulla saadaan valittua kappaleen tarkkuus törmäyksen aikana. Suuremmalla geometrisella tarkkuudella (tiheämpi verkko) saadaan tarkempia törmäystarkkuuksia, mutta kappaleen runko on tällöin alttiimpi virheille, värinälle ja erinäisille muutoksille. Törmäysmuotojen määrittämisessä on suotavaa käyttää yksinkertaistettua muotoa, sen paremman suorituskyvyn vuoksi.

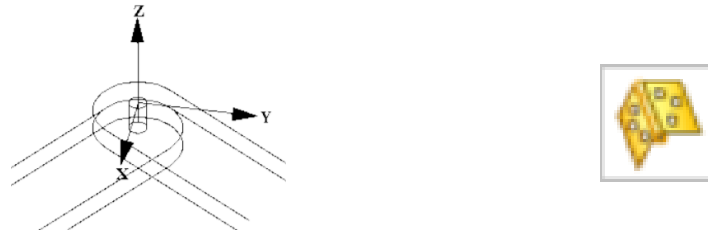
TAULUKKO 1. Törmäysmuodot ja niiden määrittäykset (Siemens 2018).

Type	Geometric Accuracy	Reliability	Simulation Performance
 Box	Low	High	High
 Sphere	Low	High	High
 Cylinder	Low	High	High
 Capsule	Low	High	High
 Convex	Medium	High	Medium
 Multi Convex	Medium	High	Medium
 Mesh	High	Low	Low

6.2.2 Liitoksien määrittäminen

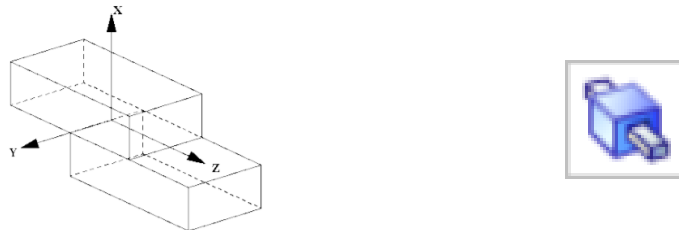
MCD sisältää lukuisia erilaisia liitostyyppjä kappaleiden liitoksien määrittämiseen. Näistä käytetyimmät liitostyypit ovat *Hinge Joint*-saranaliitos, *Sliding Joint*-liukupintaliitos, *Ball Joint*-palloliitos ja *Fixed joint*-lukittuliitos.

Hinge Joint: Kuvassa 44 on saranaliitoksen määrittästyökalu, jonka avulla määritellään kahden kappaleen välillä olevan nivelen liike akselinsa suhteen.



Kuva 44. Hinge Joint-saranaliitos (vasemmalla) ja työkalu (oikealla)

Sliding Joint: Kuvassa 45 on liukupintaliitoksen määrittästyökalu, jonka avulla määritellään kappaleen liike vain tiettyyn suuntaan.



Kuva 45. Sliding Joint-liukupintaliitos (vasemmalla) ja työkalu (oikealla)

Ball Joint: Kuvassa 46 on pallonivelen määrittästyökalu, jonka avulla nivelen liikkumissuunta on yhden pisteen ympärillä, sallien nivelen vapaanmuotoisen liikkumisen sen ympärillä.



Kuva 46. Ball Joint-pallonivel (vasemmalla) ja työkalu (oikealla)

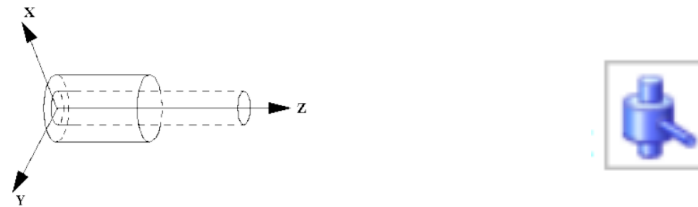
Fixed Joint: Kuvassa 47 on lukittuliitoksen määrittästyökalu, jolla voidaan lukita koko kappale paikalleen.

Lukitseminen tapahtuu *Rigid body*-työkalun avulla, jolla kappale pitää määritellä jonkin pinnan tasolle. Tämän jälkeen voidaan yhdistää kaksi *Rigid body*-työkalulla määriteltyä elementtiä toisiinsa lukitusliitoksen avulla, jolloin yhdistetty kappale liikkuu kokonaisuutena.



Kuva 47. Fixed Joint-lukittuliitos (vasemmalla) ja työkalu (oikealla)

Cylindrical Joint: Kuvassa 48 on sylinteriliitoksen määritystyökalu, jonka avulla kappaleet voivat liikkua pituus- ja pyörimissuuntaansa vapaasti.



Kuva 48. Cylinder Joint-sylinteriliitos (vasemmalla) ja työkalu (oikealla)

6.2.3 Antureiden lisääminen

Kokoonpanomallin mekaanisten määrittelyjen jälkeen voidaan simulaatioon määritellä antureita ja niihin erilaisia toiminallisuuksia. MCD sisältää useita erilaisia anturimalleja nopeuden, kiihtyvyyden, asennon, pituuden, kaltevuuden ja törmäyksen mittaamiseen. Tämän lisäksi ohjelmisto sisältää releohjaus toiminnot ja rajakytkimet, jolla saadaan määriteltyä edistyneempiä simulointimalleja. Anturimalleista yleiskäyttöisin on *Collision Sensor*-törmäysanturi, joka on yleisessä käytössä erilaisissa kuljettimissa.

Collision Sensor: Antureiden määrittämiseen tarkoitettu työkalu (kuva 49), joka tunnistaa sen tielle tulevia kappaleita. Kappaleen tunnistamiseen ja sen jälkeiseen tapahtumaketjuun käyttäjä voi määritellä, mitä kappaleelle tapahtuu, sen osuessa anturin tunnistusalueelle. Anturin asetusvalikosta voidaan määrittää tarkemmat parametrit sen geometrisen määrittelyyn. Anturin tunnistusalueen sijainti on helposti määriteltävissä kokoonpanomallissa ja sitä voidaan siirtää halutessaan.



Kuva 49. Collision Sensor-työkalu

Change material: Kappaleiden määrittämiseen tarkoitettu työkalu (kuva 50), mitkä ovat toisiinsa nähden vuorovaikutuksessa. Työkalun avulla pystytään määrittämään kapasitiivisia ja induktiivisia anturimalleja, jotka ovat suosittuja erilaisissa kuljettimissa.



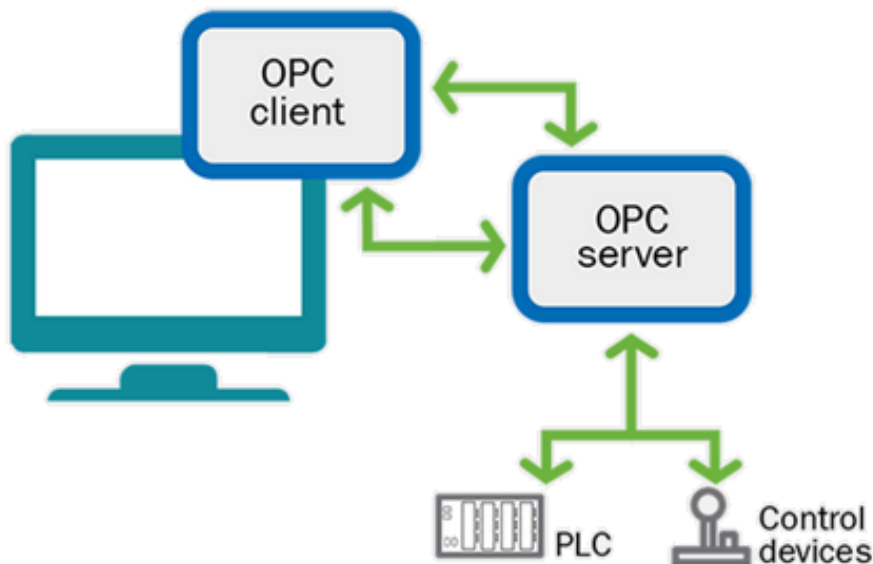
Kuva 50. Change Material-työkalu

7 YHTEYS OHJATTAVAAN ERILLISEEN JÄRJESTELMÄÄN

7.1 OPC

OPC (*Ole for Process Control*) on avoin tiedonsiirron standardi, jota käytetään yleisesti automaatiosovelluksissa. OPC:n avulla voidaan luoda yhteys ohjelmoitavan logiikan ja automaatiojärjestelmän välille. OPC sisältää useita erilaisia määrittelyjä, jolla jokaisella eri käyttötarkoituksena. OPC DA (*Data Access*) on tarkoitettu reaaliaikaisen datan siirtoon prosessilaitteista ja ohjausjärjestelmistä. OPC A&E (*Alarms and Events*) on tarkoitettu hälytys- ja tapahtumatietojen välittämiseen. OPC HDA (*Historical Data Access*) on tarkoitettu datankeruuseen.

OPC-palvelin on pääasiassa tarkoin määritelty ohjelma, joka on tehty tietyille automaatiolaitteistolle. Ohjelman avulla kyetään välittämään mittaus -ja ohjaustietoja rajapinnan avulla laitteen ja ulkopuolisen järjestelmän välillä. OPC sisältää asiakas (*Client*) ja palvelin (*Server*) pohjaisen kommunikoinnin (kuva 51), joka pyytää tietoja palvelimelta tai palvelin toimittaa tiedot asiakkaalle kyselyjen avulla (Novotek: OPC n.d; Heikkinen n.d).



Kuva 51. Laitteiden välinen OPC-yhteys (Novotek n.d, muokattu).

7.2 MCD tiedonvälitys TIA Portal V15.1 ohjelmaan

MCD käyttää hyväkseen OPC-tekniologiaa yhdistämään rajapinnat ulkoisen ohjelmoitavan logiikan ja OPC-palvelimen välillä. Tämän avulla MCD voi vaihtaa tietoja ulkoisesta rajapinnasta sekä käyttää sen ohjaussignaaleja simuloinnin ja visualisoinnin aikana.

Ulkoiset ohjaussignaalit voidaan yhdistää MCD:ssä käyttämällä *Signal mapping*-työkalua (kuva 52). MCD tukee OPC DA-protokollan lisäksi MATLAB, PLCSIM adv, Profinet, SHM ja UDP- tyyppisiä protokollia järjestelmien välisessä tiedonvälityksessä. (Siemens 2018).



Kuva 52. Signal Mapping-työkalu

Signal Adapter: työkalun avulla (kuva 53) määritellään MCD:n puolella toimilaitteiden signaalit ja toiminnallisuudet asetettujen ehtojen mukaan. Voit sisällyttää useita signaaleja suoritusajakaavioiden sisään ja määrittää niille ehtoja, esimerkiksi kuljettimen hihnan ja antureiden signaalien toiminnallisuudet määritellään tämän avulla.



Kuva 53. Signal adapter-työkalu

Symbol Table: työkalun avulla (kuva 54) voidaan nimetä signaaleja ja tuoda niitä erillisestä ohjelmasta. Symbolitaulukko tukee .asc -ja txt-formaatteja, jotka sopivat yhteen STEP7 ja Teamcenter ohjelman kanssa.



Kuva 54. Symbol Table-työkalu

Ennen ohjaussignaalien kytkemistä tulee MCD malliin luoda liitännät antureille ja toimilaitteille, jotka yhdistetään ulkoisen järjestelmän kanssa. Tiedonvälitykseen anturit tulee määrittellä ulostulo (*output*) signaalien muotoon, koska signaalien tulee kommunikoida ulkoisen järjestelmän avulla. Toimilaitteet määritellään sisääntulo (*input*) signaalien muotoon.

Siemens NX MCD ja TIA Portal ohjelmiston yhteen liittämiseen on erilaisia vaihtoehtoja ulkoisen OPC-palvelimien ja SIMATIC NET-ohjelmistojen välillä.

Tässä opinnäytetyössä yhdistäminen toteutetaan PCLSIM Advanced -ohjelmiston avulla, joka on tarkoitettu käyttöön Tampereen ammattikorkeakoulun koneautomaation ope- tuksessa

7.2.1 PLCSIM Advanced

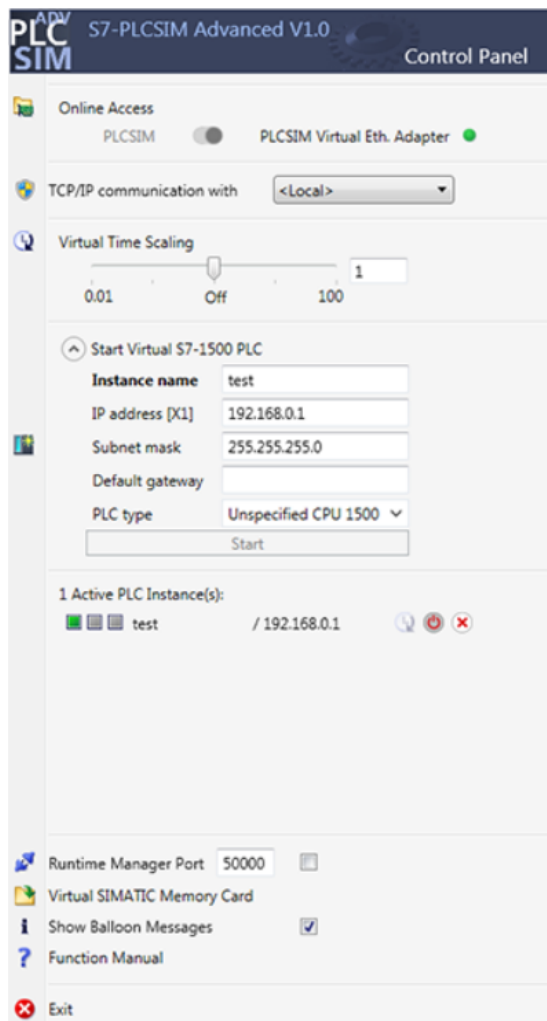
PLCSIM Advanced on TIA Portal ohjelmistoon kuuluva lisäosa, jonka avulla pystytään suorittamaan laitteen virtuaalinen käyttöönotto, ilman erillistä fyysistä laitetta. Ohjel- misto tukee S7-1500 ja ET200SP-sarjan ohjelmoitavia logiikoita ja on yhteensopiva TAULUKKO 2 versioiden mukaan.

TAULUKKO 2. Yhteensopivuus eri ohjelmistoversioiden kanssa (Siemens 2019)

PLCSIM Advanced	TIA Portal	Supported CPU firmware version
V1.0 SP1	V14 to V15.1	V1.8, V2.0
V2.0	V14 to V15.1	V1.8 to V2.5
V2.0 SP1	V14 to V15.1	V1.8 to V2.6

PLCSIM Advanced -ohjelmiston avulla virtuaalisten ohjaimien simulointi ja varmenta- minen voidaan toteuttaa rajapintatekniikan (API) avulla erillisessä mallin virtuaalisessa ympäristössä esimerkiksi MCD:ssä tai eri ohjelmistosovelluksissa.

PLCSIM Advanced -ohjelmiston etuna on, että TIA Portal-ohjelmistosta voidaan ladata ohjelmat ja laiteasetukset suoraan PLCSIM Advanced -ohjelmiston sisältämään virtuaa- liseen muistikorttiin, jolloin logiikan käyttö sujuu ilman logiikkaohjelmiston käynnistä- mistä uudelleen. Lisäksi ohjelmiston avulla voidaan simuloida monia logiikkainstansseja samanaikaisesti. Kommunikaation osalta PLCSIM Advanced -ohjelmisto mahdollistaa Softbus kommunikaation lisäksi TCP/IP protokollan mukaisen kommunikaation ohjel- mistossa olevalla virtuaalisella Ethernet adapterilla, joka tukee tiedonvälitystä OPC UA ja Web-palvelimen välillä. Kuvassa 55 on PCLSIM Advanced -ohjelmiston käyttöliit- tymä ja liitteessä 3 on kerrottu tarkemmin käyttöliittymän määrittämisestä (Siemens 2018).



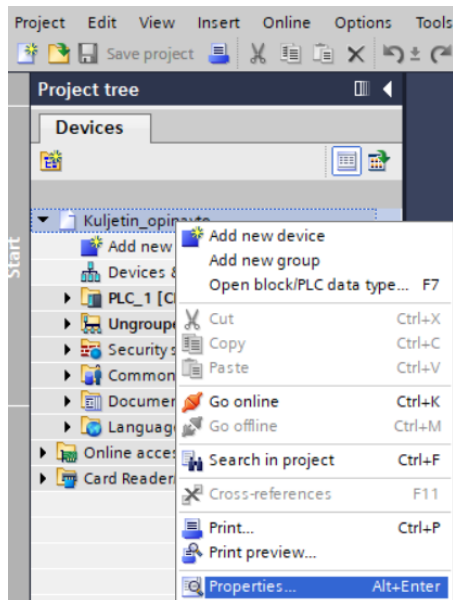
Kuva 55. PLCSIM Advanced -käyttöliittymä

7.3 OPC-palvelimen asettaminen TIA Portal V15.1 ohjelmistoon

Jotta PCLSIM Advanced-ohjelmisto löytää TIA Portal-ohjelman, määritä OPC-palvelin aktiiviseksi. Tee virtuaalisen käyttönoton simulointi S7-1500 tai ET200SP -sarjan ohjelmitavien logiikoiden avulla.

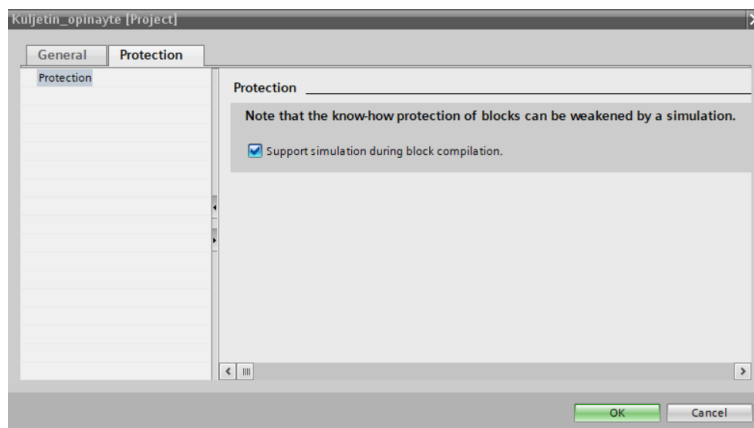
Luo OPC-palvelin seuraavasti:

1. *Devices* -osiossa napsauta hiiren oikealla painikkeella projektin nimeä ja valitse **properties** (kuva 56). Projekti-ikkuna aukeaa.



Kuva 56. Projektin *Properties*-valinta

2. Valitse *Protection* välilehti.
3. Napsauta **Support simulation during block compilation** -valintaruutua ja napsauta **OK** -painiketta (kuva 57).

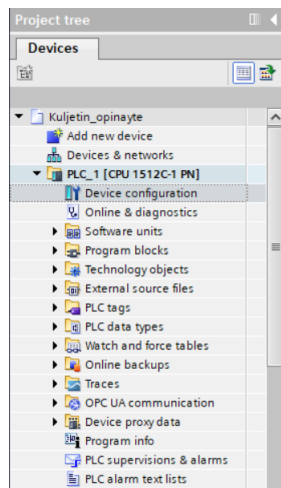


Kuva 57. Properties-ikkuna

4. Jatka määrittämistä 7.3.1 kohdassa.

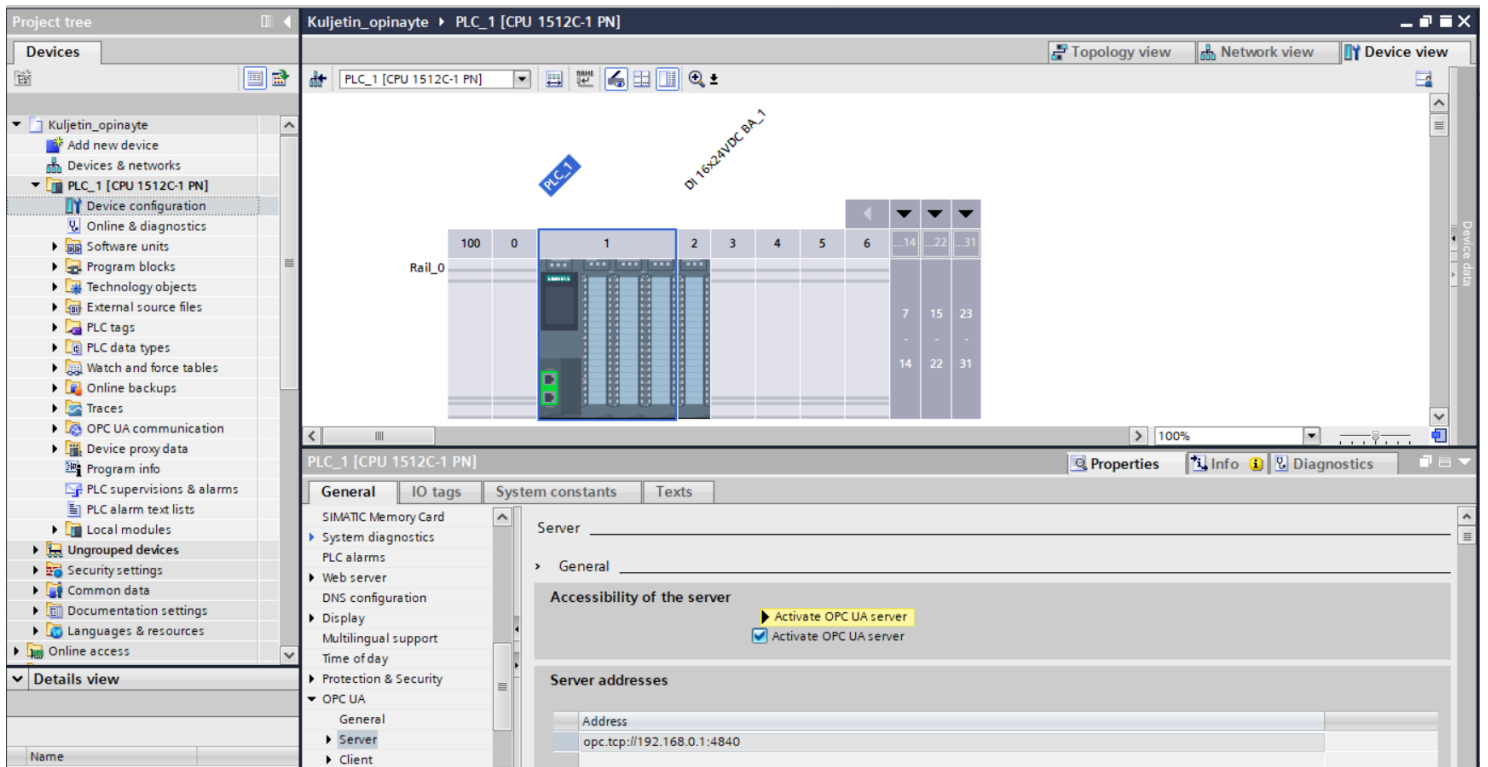
7.3.1 OPC UA -valinta

1. Avaa *Project tree* -näkyvä (kuva 58).
2. Napsauta **PLC_1** -kohtaa. PLC:n valikko aukeaa.



Kuva 56. TIA Portalin piirrepuu

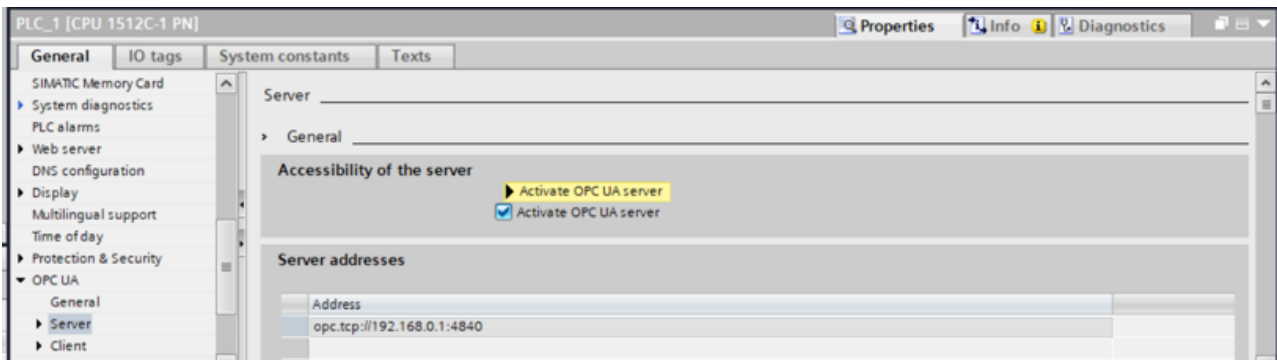
3. Napsauta **Device configuration** -kohtaa. *Device view* -ikkuna aukeaa (kuva 59).



Kuva 57. Device view-näkymä

4. Napsauta *General*-välilehdeeltä **OPC UA** -kohtaa. Ikkuna aukeaa.

5. Napsauta **Server** -kohtaa. Ikkuna aukeaa (kuva 60).



Kuva 60. OPC-palvelimen asettaminen

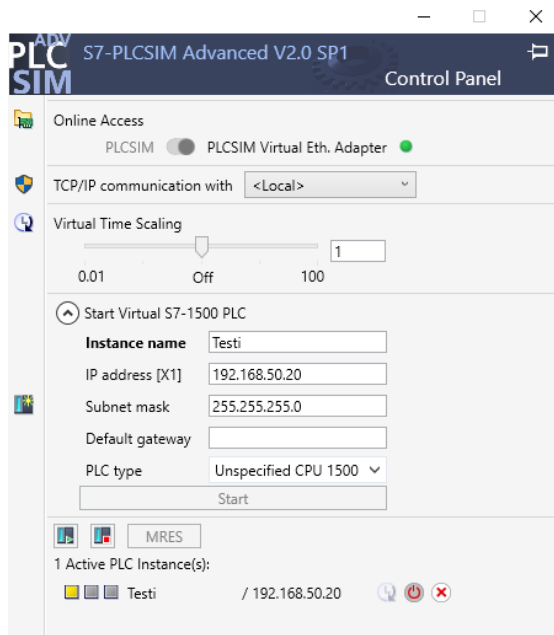
6. Napsauta **Active OPC UA Server** -valintaruutua. Varoitusikkuna aukeaa.
7. Napsauta **OK** -painiketta.
8. Jatka kohdasta 7.3.2.

7.3.2 Yhteys PLCSIM Advanced -ohjelmistoon

Ennen yhteyden muodostamista virtuaaliselle kontrollerille, tulee PLCSIM Advanced -ohjelmistoon luoda instanssi. Instanssin luominen on kerrottu liitteessä 2 ja kappaleessa 7.3.

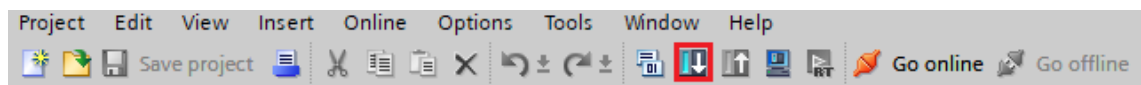
Kytke yhteys PLCSIM Advanced -ohjelmistoon seuraavasti:

1. Luo TIA Portal ohjelmaan logiikkaohjelma.
2. Luo PLCSIM Advanced -ohjelmistoon instanssi. Kohta *Active PLC Instance(s)* on keltainen, kun ohjelma on löytänyt rajapinnan TIA Portalista (kuva 61).



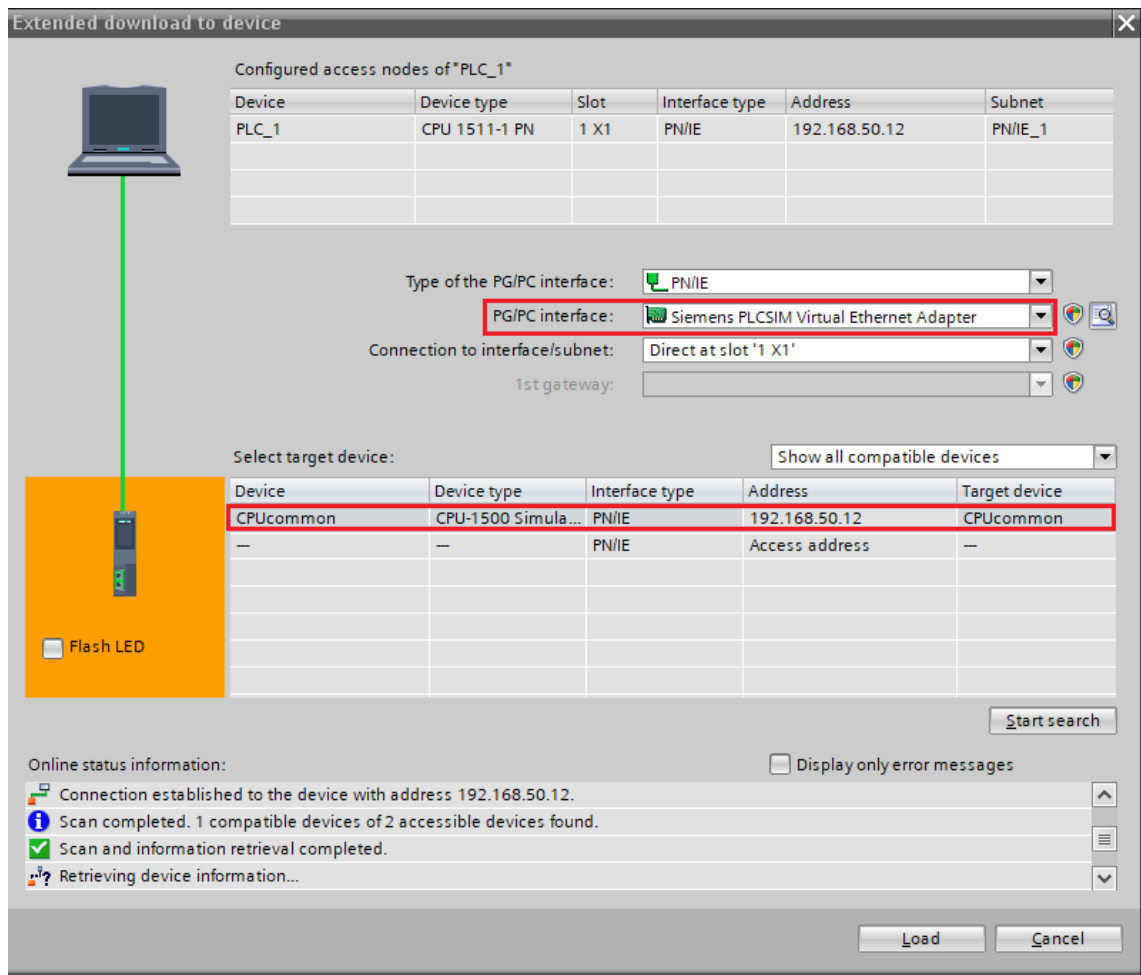
Kuva 8. Instanssin luominen.

3. Napsauta TIA Portal -ohjelman **Download to Device** -kohtaan (kuva 62). Ikkuna aukeaa.



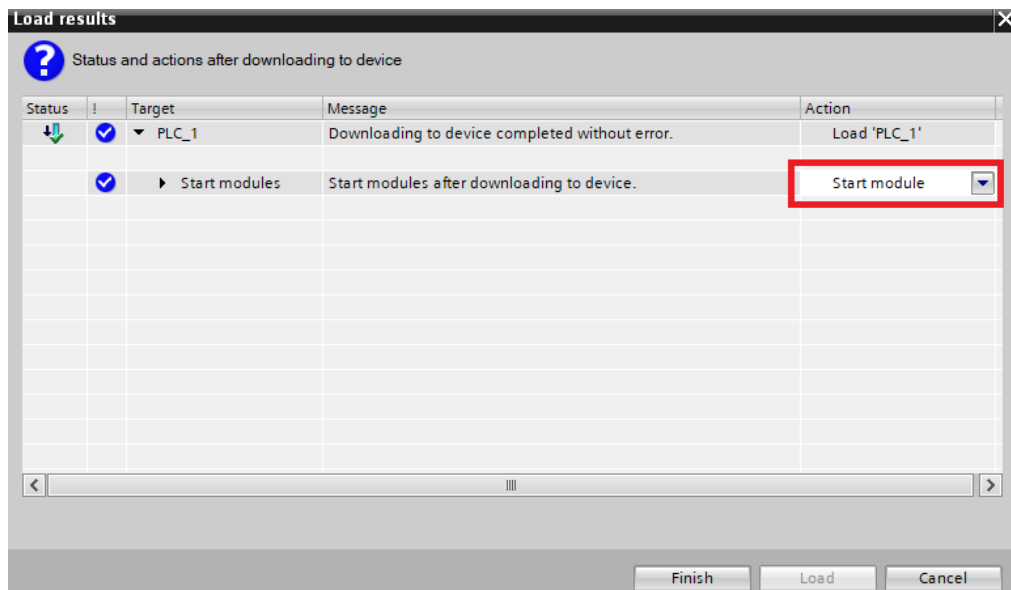
Kuva 62. TIA Portalin ohjelman lataaminen logiikkaan

4. Määritä säätöikkunaan seuraavat parametrit kuvan 63 perusteella:
 - a. Valitse *PG/PC interface* kohtaan **Siemens PLCSIM Virtual Ethernet Adapter**.
 - b. Napsauta **Start search** -painiketta. Jos ohjelma ei löydä logiikkaa, määrittökset ovat pielessä.
 - c. Valitse *Select target device* -taulukosta oikea CPU-logiikka.
 - d. Napsauta **Load** -painiketta. Ikkuna aukeaa.



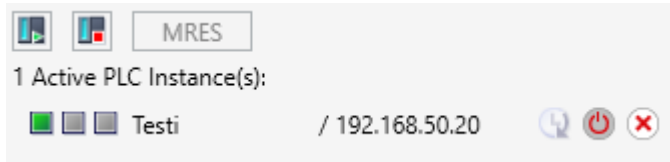
Kuva 63. Ohjelman lataaminen logiikkaan.

5. Napsauta **Load-** painiketta uudestaan. Ikkuna aukeaa (kuva 63).
6. Valitse **Start module** pudostusvalikosta.



Kuva 64. Load results-valintaikkuna

7. Napsauta **Finish**.
8. Tarkasta, että PCLSIM Advanced- ohjelmistoon yhteys on päällä (kuva 65). Jos ohjelma ei ole yhdistänyt ohjelmistoja, määriykset ovat pielessä.



Kuva 65. Instanssi on kytkettynä päälle.

8 HARJOITUSTEHTÄVÄN LAADINTA

Opinnäytetyön tarkoitus on tuottaa Tampereen ammattikorkeakoululle harjoitustyö koneautomaatio opiskelijoiden käyttöön. Harjoitustyön tavoitteena on parantaa opiskelijoiden osaamista MCD:n käytöstä ja ymmärtää paremmin kokonaista suunnitteluprosessia ja siihen liittyviä vaiheita.

8.1 Kuljettimen määrittäminen

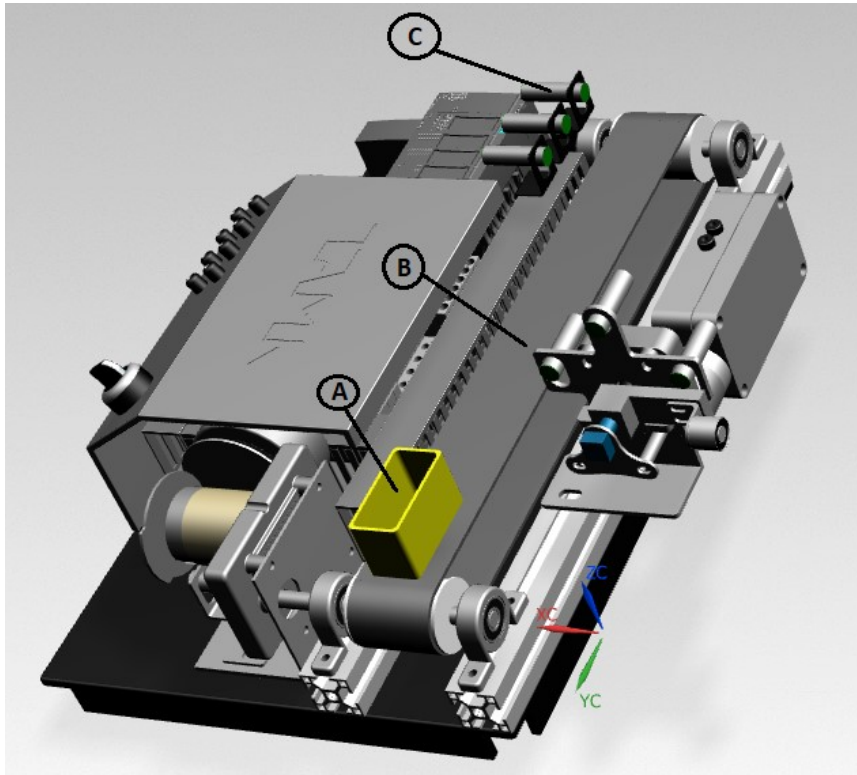
Harjoitustyön komponenteille määritellään toiminnallisuudet, anturointi sekä 3D-malliin kytketään TIA Portal automaatio-ohjelmisto ja simuloidaan kuljettimen toiminta.

8.1.1 Liikkuvien osien ja kosketuspintojen määrittäminen

Määritä **Rigid body**- ja **Collision body**-työkalujen avulla kuljettimen hihnan päällä olevat kappaleet **A** ja **B**. (kuva 66):

1. Valitse **Rigid body**-työkalu.
2. Napsauta kappaletta **A**.
3. Määritä aukeavaan ikkunaan haluamasi parametrit.
4. Valitse **Collision body**-työkalu.
5. Napsauta kappaleen **A** pohjaa, jolloin se määrittyy kosketuspinnaksi.
6. Määritä aukeavaan ikkunaan haluamasi parametrit.
7. Valitse **Collision body**-työkalu.

8. Napsauta kappaletta **B**, jolloin se määrittyy kosketuspinnaksi.
9. Määritä aukeavaan ikkunaan haluamasi parametrit.
10. Jatka määrittelyä kohdasta 8.1.2.



Kuva 66. Tampereen ammattikorkeakoulun kuljetin 3D-ohjelmassa ja sen pääkomponentit A B ja C (Tampereen ammattikorkeakoulu 2018, muokattu)

8.1.2 Kuljetinhihnan liikkeen määrittäminen

Määritä kuljetinhihnan liike **Transport surface** -työkalun avulla:

1. Valitse **Transport surface**-työkalu.
2. Napsauta kappaletta **B**, jolloin se aktivoituu.
3. Määritä aukeavaan ikkunaan seuraavat arvot:
 - a. Valitse **Velocity**-taulukosta haluamasi nopeus ja syötä se **Parallel**-kenttään.
 - b. Valitse **Specify vector** -osiossa kuljetinhihnan liikesuunta napsauttamalla haluamaasi suuntavektoria.
 - c. Jatka määrittelyä 8.1.3.

8.1.3 Liikuteltavan kappaleen monistaminen

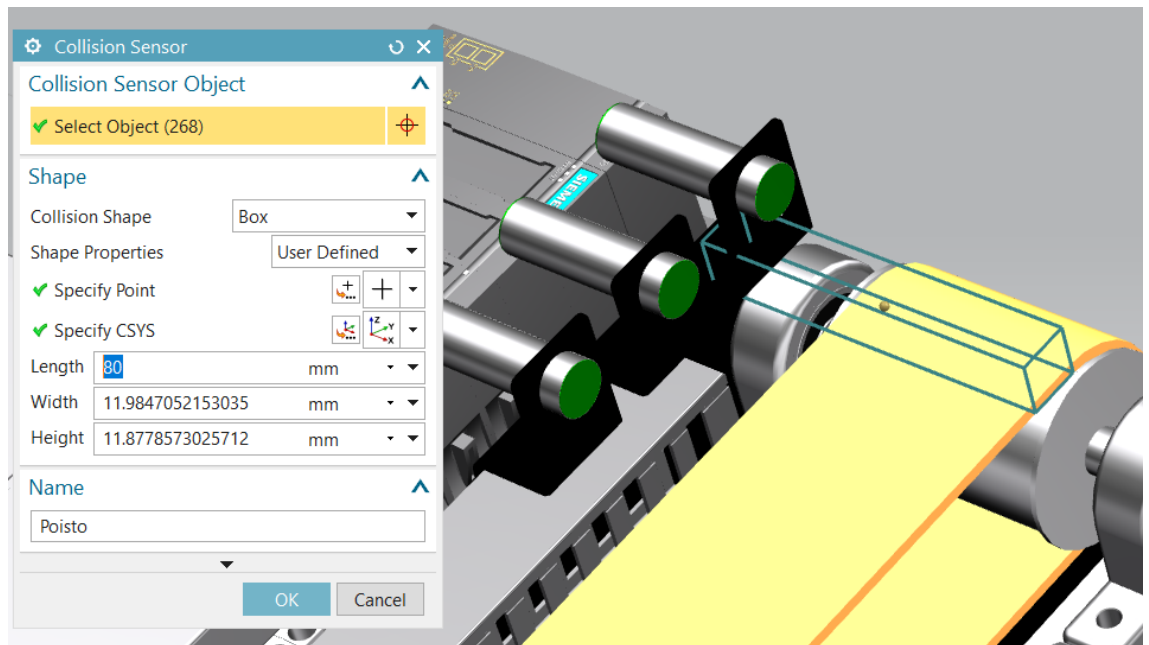
Monista kuljetinhihnan päällä kulkevia kappaleita käyttämällä **Object source** -työkalua:

1. Valitse **Object source** -työkalu.
2. Napsauta kappaletta A.
3. Määritä aukeavaan ikkunaan haluamasi parametrit, kuten monistuvien kappaleiden lisäysaika.
4. Jatka määrittelyä kohdasta 8.1.4.

8.1.4 Anturin lisääminen

Lisää anturi **Collision sensor**-työkalun avulla:

1. Valitse **Collision sensor**-työkalu.
2. Napsauta kohtaa, johon anturi lisätään (kuva 67).
3. Määritä aukeavaan ikkunaan anturin muoto ja sijainti (kuva 67). Käytä anturin määrittelyssä **User Defined**-asetusta.
4. Jatka määrittelyä kohdasta 8.1.5.

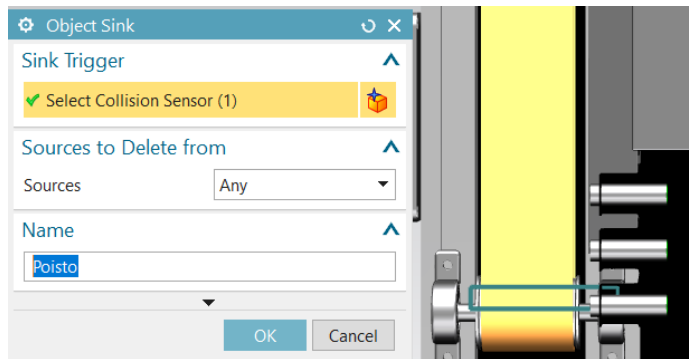


Kuva 67. Collision Sensor-työkalun valintaikkuna

8.1.5 Kappaleiden poistaminen kuljettimesta

Poista kappaleita **Object sink**-työkalulla:

1. Valitse **Object sink** -työkalu. Ikkuna aukeaa (kuva 68).
2. Napsauta **Sink trigger** -kohtaa ja valitse aiemmin määrittelemäsi anturi. Nyt ohjelmisto tunnistaa ne kappaleet, jotka pitää poistaa.
3. Jatka määrittelyä kohdasta 8.1.6.

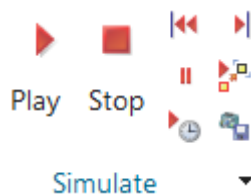


Kuva 68. Object Sink-työkalun määrittelyikkuna antureineen.

8.1.6 Mekanismin animointi

Kun määrittelykset on tehty, simuloi kuljettimen toimintaa *Simulate*-työkalulla (kuva 69):

1. Valitse **Home > Simulate > Käytä** painikkeita.
2. Paina **Play**, jotta näet, liikkuvatko kappaleet. Jos kappaleet eivät liiku, määrittelykset ovat virheellisiä.
3. Jatka määrittelyä kohdasta 8.1.7.



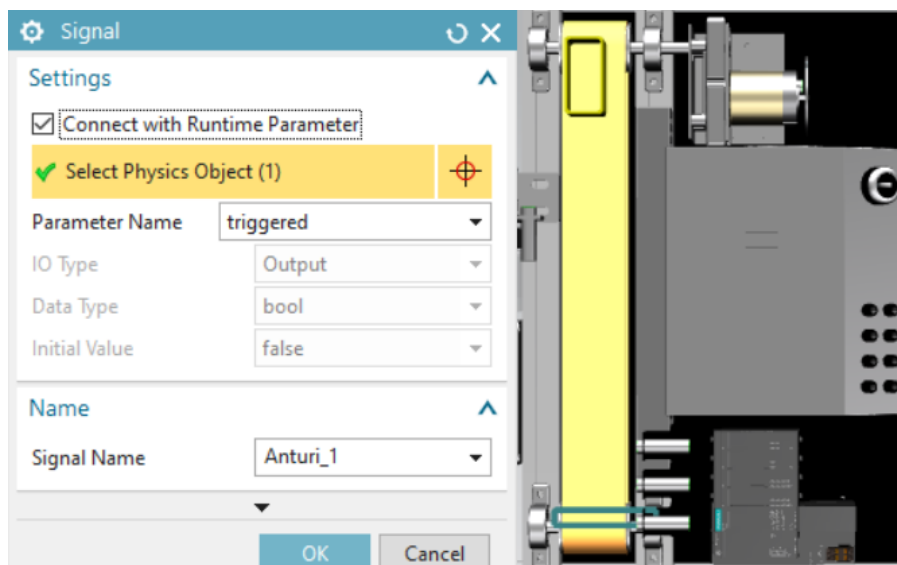
Kuva 69. Mekanismin simulointipainikkeet.

8.1.7 MCD -ohjaussignaalien luominen

Määritä **Signal Adapter**-, **Signal**- ja **Symbol Table** -työkalujen avulla kuljettimen ohjaussignaalit kappaleille **B** ja **C**.

Määrittele anturisignaali **Signal**-työkalulla:

1. Valitse **Signal**-työkalu. Ikkuna aukeaa (kuva 70).
2. Napsauta **Select Physics Object** -kohtaa ja valitse aiemmin määrittelemäsi anturi.
Nyt ohjelmisto luo anturille signaalin.
3. Napsauta **Connect with Runtime Parameter** -valintaruutua.
4. Valitse *Parameter Name* -pudotusvalikosta **Triggered** -kohta.
5. Valitse *Signal Name* -kohdassa anturille nimi.
6. Paina **OK**.
7. Jatka määrittelyä kohdassa 8.1.8.

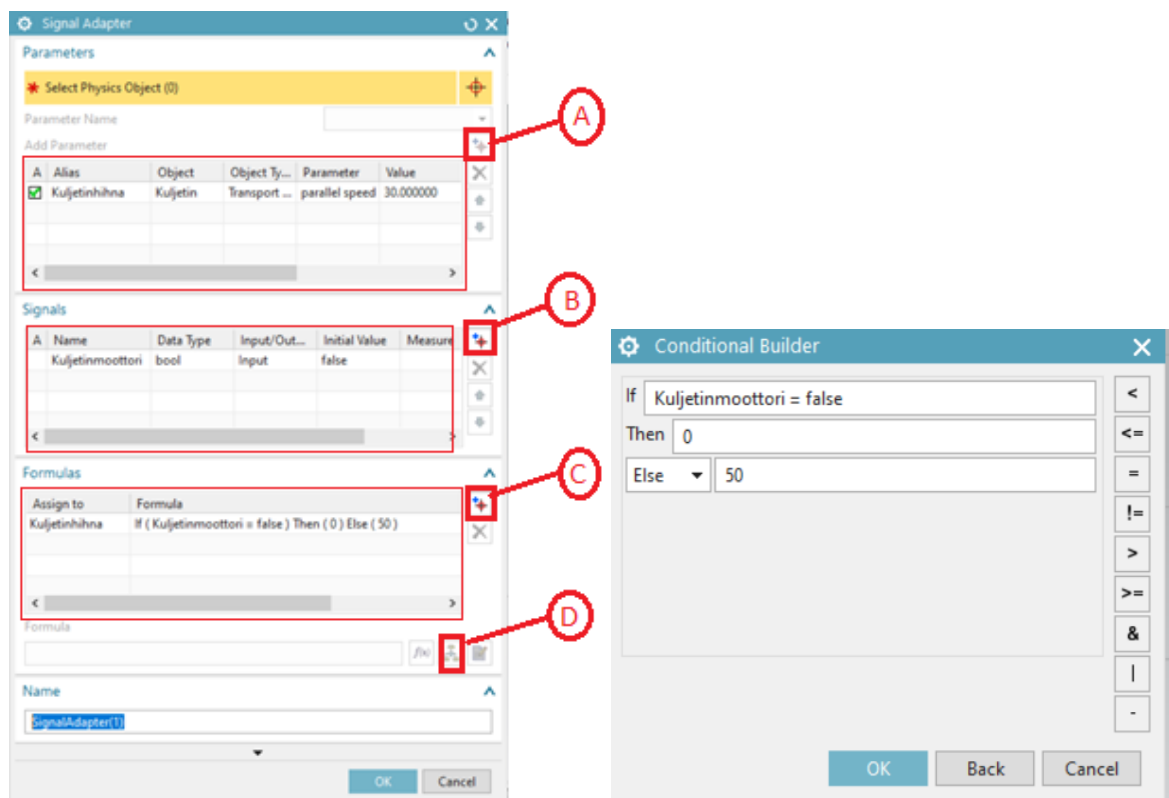


Kuva 70. Signal-työkalun määritysikkuna.

8.1.8 Kuljetinhihnan signaalin määrittäminen

Määritä kuljetinhihnan ohjaussignaalin (kappale **B**) toiminnallisuus käyttämällä **Signal Adapter** -työkalua:

1. Valitse **Signal Adapter**-työkalu. Ikkuna aukeaa (kuva 71).
2. Napsauta **Select Physics Object** -kohtaa ja napsauta kuljetinhihnaa. Hihna aktivoituu.
3. Napsauta **A** -kohtaa (kuva 71). Edellisen kohdan määrittäminen ilmestyy *Add Parameter* -taulukkoon.
4. Napsauta **B** -kohtaa (kuva 71). *Signals* -taulukkoon ilmestyy ohjaussignaali. Vaihda signaalin nimi ja lähdeksi *Input*.
5. Napsauta **Kuljetinhihna** valintaruutua (kuva 70).
6. Napsauta **D** -kohtaa. Ikkuna aukeaa.
7. Määritä kuvan 70 mukaiset ehdot.
8. Määritä *Name* -kohtaan nimi.
9. Napsauta **OK**. Jos *Symbol Table*-ikkuna aukeaa. Napsauta **OK**.
10. Jatka määrittämistä kohdassa 8.1.9.



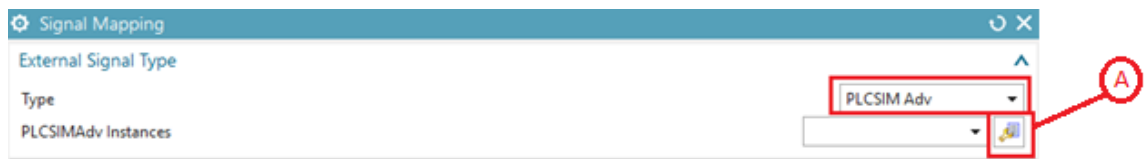
Kuva 71. Signal Adapter-ikkuna (vasemmalla) ja ehtojen määrittämissikkuna (oikealla)

8.1.9 Signaalien yhdistäminen ohjelmistojen välille

MCD ja TIA Portal ohjelmiston rajapinnat yhdistetään PLCSIM Advanced -ohjelmisto avulla. Kohdassa 8.1.10 on määritelty instanssin luominen ja yhdistäminen TIA Portaliin.

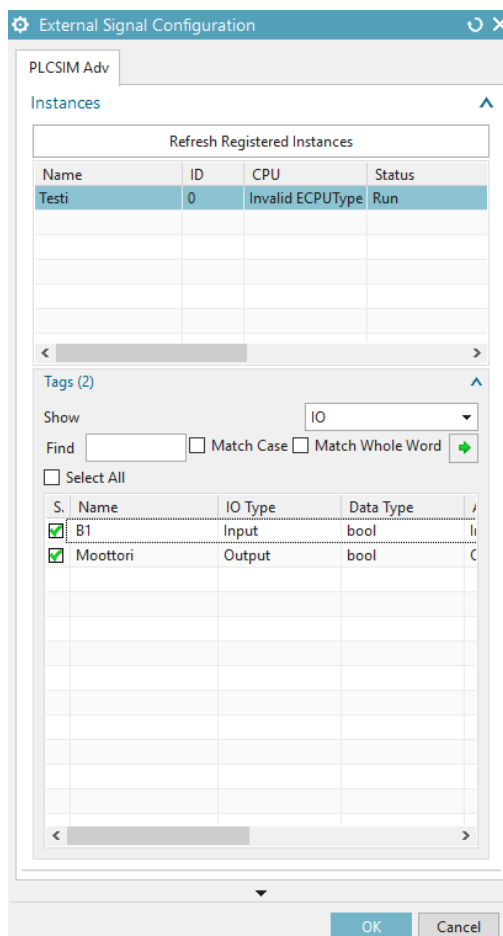
Yhdistä MCD:n ja TIA Portal ohjelmistojen signaalit **Signal Mapping** -työkalulla:

1. Napsauta **Signal Mapping** -työkalua. Ikkuna aukeaa.
 - a. Valitse *Type* -kohtaan **PLCSIM Adv** (kuva 72).
 - b. Napsauta **A**-kohtaan. Ikkuna aukeaa.



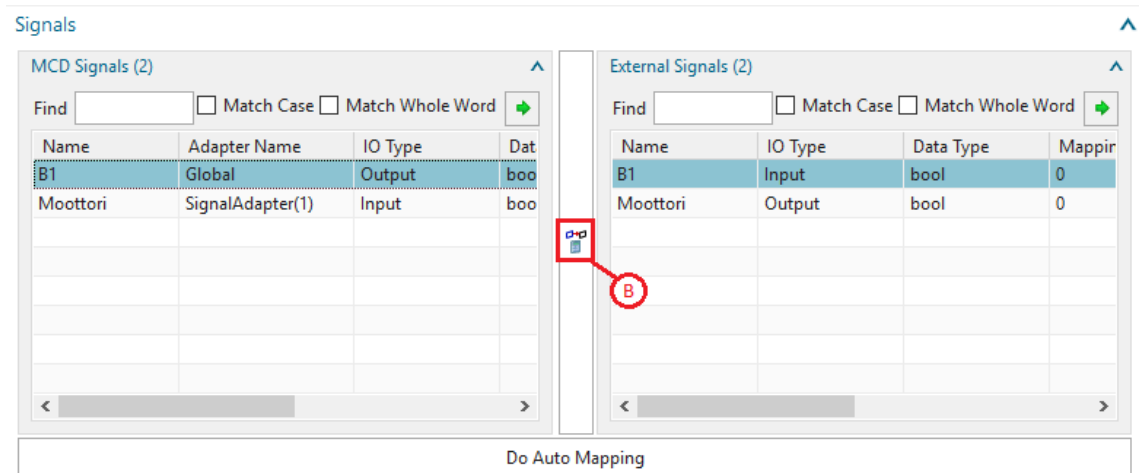
Kuva 72. Ulkoisen ohjelman liitettä

2. Napsauta **Refresh Registered Instances** (kuva 73).
3. Valitse *Tags (2)* taulukosta molemmat tagit.
4. Napsauta **OK**.



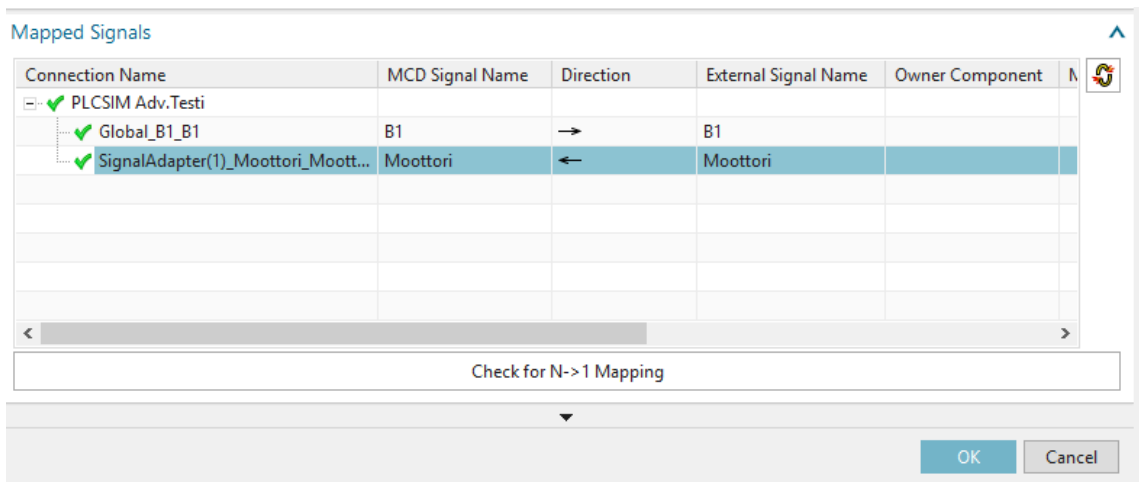
Kuva 73. External Signal-instanssin haku

5. Valitse *Signals* -taulukosta yhteen kytkettävät tulot ja lähdöt (kuva 74).
6. Napsauta **B** -kohtaan. Signaalit siirtyvät *Mapped Signals* -taulukkoon (kuva 75).



Kuva 74. Signals-näkymä

7. Napsauta **OK**.
8. Jatka määrittämistä 8.1.10.

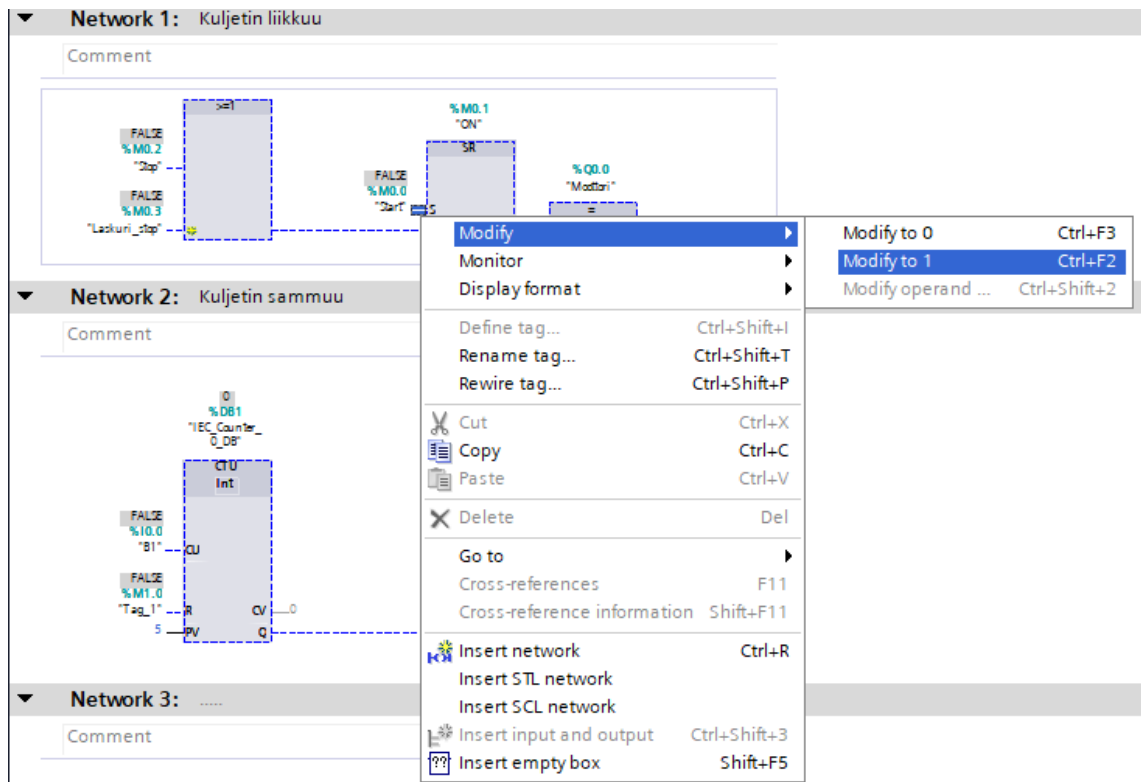


kuva 75. Mapped Signals-näkymä

8.1.10 Kuljettimen toiminnan simuloiminen

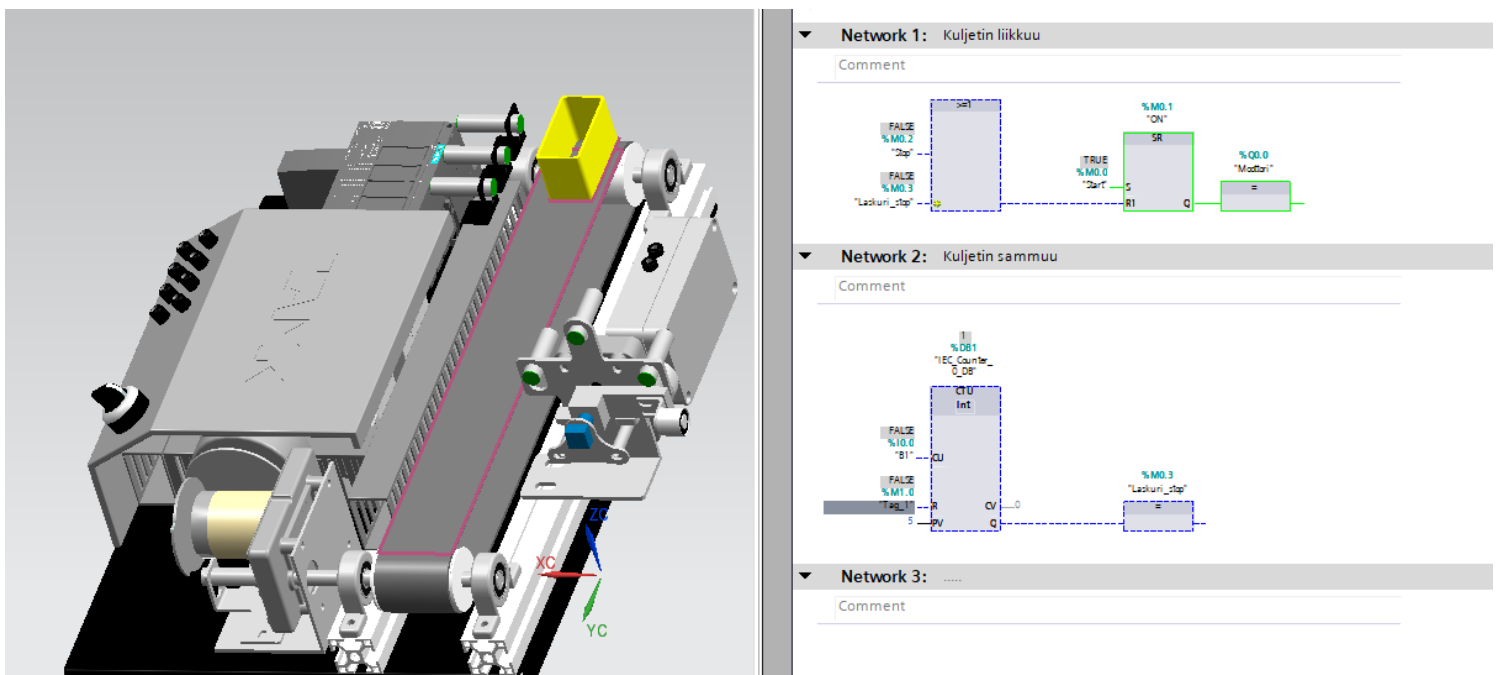
Signaalien määrittysten jälkeen testaa kuljettimen toiminta seuraavasti:

1. Kytke TIA Portalissa *Simulate* -tila päälle.
2. Napsauta hiiren oikealla logiikkaohjelman käynnistysfunktiota (kuva 76).
3. Valitse **Modify>Modify to 1**. Logiikkaohjelma käynnistyy.



Kuva 76. TIA Portal simulaatio näkymä

4. Paina **Play** MCD:ssä, jotta näet, liikkuvatko kappaleet. Jos kappaleet eivät liiku, määrytykset ovat virheellisiä.
5. Tarkastele kuljettimen toiminta kuvan 77 mukaan.



Kuva 97. Kuljettimen simulaation MCD:ssä (vasemmalla) ja TIA Portalissa (oikealla)

9 POHDINTA

Tämän opinnäytetyön tarkoitus oli kehittää Tampereen ammattikorkeakoulun kone -ja tuotantotekniikan ohjausjärjestelmän opintojakson käyttöön tulevaa 3D-mallia kuljettimesta. Haasteena opintojaksolla on ollut kuljettimien vähäinen määrä sekä monimuoto-opiskelijoiden haasteelliset aikataulut päästäkseen lähiopetukseen.

Työn tavoitteena oli pystyä yhdistämään mekaniikkaohjelmisto Siemens NX MCD ja automaatio-ohjelmisto TIA Portal V15.1 ja simuloida kuljettimen mekaaniset ja automaatiopohjaiset määritykset. Ohjelmistojen liitännän avulla tulevat opiskelijat pystyvät tekemään opintojaksolla tulevia harjoitustehtäviä omalla koneellaan ensin ja myöhemmin laaamaan tekemänsä automaatiologiikan oikealle kuljettimelle ja testaamaan sen toiminta.

Työn haastavin osuus oli oikeanlaisten ohjelmistoversioiden yhdistäminen toisiinsa. Työn alkuvaiheessa yritettiin yhdistää TIA Portal V13 -ohjelma MCD:n kanssa yhteen, mutta Tampereen ammattikorkeakoulun koneautomaation luokassa oli väärienlaiset ohjelmistopakettit, josta puuttuivat esimerkiksi OPC-palvelimen asettaminen. Työ saatiin kuitenkin tehtyä, kun koneautomaatioluokkaan asennettiin uudempi versio TIA Portalista. TIA Portal V15.1 on tarkoitus ottaa opetuskäyttöön myöhemmin Tampereen ammattikoulussa. Ohjelmiston etuna vanhaan ohjelmaversioon V13 on sen helpompi liitettävyys MCD:n kanssa yhteen PLCSIM Advanced -lisäohjelman avulla.

Kuljettimen 3D-mallin yhdistämisestä ulkoisen automaatio-ohjelmiston kanssa on luovutettu koululle käyttöohjeet harjoitustehtävän muodossa. Harjoitustehtävä opettaa käytännön työn kautta opiskelijoille, kuinka Siemens NX MCD:ssä olevalle 3D-mallin komponenteille määritellään toiminnallisuudet ja signaalit sekä kuinka niiden kytkentä tapahtuu TIA Portalin kanssa. Lisäksi käyttöohjeisiin kuuluu osio, jossa logiikkaohjelmalla ohjataan kuljettimen 3D-mallia.

Siemens NX MCD sisältää useita erilaisia työkaluja. Tässä opinnäytetyössä käsiteltävät ovat vain yksi osa siitä. Aiheesta on suositeltava tehdä toinen opinnäytetyö, joka käsittelee Siemens NX MCD ja Matlab SIMULINK-ohjelman yhdistämistä ja 3D-mallien dynamiikan määrittämistä. Lisäksi erilaisten harjoituspohtien luonti esimerkiksi robottisolun ja tuotantolinjan operoiminen MCD:n ja PLC-ohjelman avulla voisi olla seuraava vaihe opetuksen kehittämisessä.

LÄHTEET

Bolton, W. 2003. Mechatronics: Electronic control systems in mechanical and electrical engineering. 3 painos. Harlow: Pearson, 0-3.

Bishop, R. H. 2006. Mechatronics: An introduction. Boca Raton: Taylor & Francis, 1-3

Collins, D. 2018. Can mechatronics engineering help alleviate the shortage of skilled manufacturing worker. Luettu 23.7.2019. <https://www.motioncontroltips.com/can-mechatronics-engineering-help-alleviate-the-shortage-of-skilled-manufacturing-workers/>

Laitinen, S. n.d. Lyhyt UML-opas. PDF-tiedosto. Luettu 23.10.2019. <http://www.cs.tut.fi/~ohjsuun/luennot/2015/UMLpikaopas.pdf>

Friedenthal, S. Moore, A & Steiner, R. 2014. Practical Guide to SysML. The Systems Modeling Language. Elsevier Science & Technology. Luettu 25.10.2019.

Dunstan, T. n.d. What is SysML, The Systems Modeling language Explained. Luettu 26.10.2019. <http://dthomas-software.co.uk/resources/frequently-asked-questions/what-is-sysml-2/>

Powell-Morse, A. 2016. Waterfall Model: What is it and when should you use it. Blogi. Luettu 24.7.2019. <https://airbrake.io/blog/sdlc/waterfall-model>

Leppälä, K. 2000. Kohti systemaattisesta tuotekehitystä. VTT Elektroniikka. PDF-tiedosto. Luettu 4.7.2019. [http://virtual.vtt.fi/virtual/proj1/projects/finesse/tuotekehitys\(finesse\).pdf](http://virtual.vtt.fi/virtual/proj1/projects/finesse/tuotekehitys(finesse).pdf)

Künzel, G. 2005. The mechatronics design process. PDF-tiedosto. Luettu 6.7.2019 <https://pdfs.semanticscholar.org/11a7/2e14ddd855ab926ea4342efe886ea22f773b.pdf>

Powell-Morse, A. 2016. V-model: what is it and how do you use it. Blogi. Luettu 24.7.2019. <https://airbrake.io/blog/sdlc/v-model>

SuomiDigi. n.d. Kettärän kehittämisen johtamine. Luettu 24.7.2019. <https://suomidigi.fi/menetelmat/ketteran-kehittamisen-johtaminen/>

Pulkkanen, A. n.d. Sinunkin kannattaa valita: 6 yleistä menetelmää projektityöhön (sis. Agile, Waterfall ja Kanban). Luettu 24.7.2019 <https://www.agendum.com/post/agile-waterfall-kanban-6-projektinhallintamenetelmaa>

Turun yliopisto & Työterveyslaitos, 2014. Sulautettujen järjestelmien ketterä käsikirja. Luettu 15.8.2019. https://tech.utu.fi/embedded_kasikirja/index.html

Schwaber, K & Sutherland, J. 2017. The Scrum Guide. Pdf-tiedosto. Luettu 20.7.2019. <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>

Scrum, N.d. The Scrum Framework poster. Luettu 26.7.2019 <https://www.scrum.org/resources/scrum-framework-poster>

Ahola, Hovila, J. Karhunen, E. Nevala, K. Schäfer, T. & Nevala, T. 2011. Monitekkinen piensarjatuotteet digitaalinen tuoteprosessi. Pdf-tiedosto. Luettu 9.9.2019. <https://www.vtt.fi/inf/pdf/tiedotteet/2011/T2592.pdf>

Liu, Z. Suchhold, N. & Diedrich, C. 2012. Virtual Commissioning of Automated Systems. Luettu 9.9.2019. <https://www.intechopen.com/books/automation/virtual-commissioning-of-automated-systems>

Rosen, R. Wichert, G. Lo, G. Bettenhausen, K. 2015. About the importance of Autonomy and Digital Twins for the Future of Manufacturing. Pdf-tiedosto. Luettu 10.9.2019 <https://www.sciencedirect.com/science/article/pii/S2405896315003808>

Boschert, S. Heinrich, C. & Rosen, R. 2018. Next Generation Digital Twin. Pdf-tiedosto. Luettu 11.9.2019. https://www.researchgate.net/publication/325119950_Next_Generation_Digital_Twin

Siemens. Digitaalinen kaksonen simuloi täydelliseksi. Luettu 15.9.2019. <https://new.siemens.com/fi/fi/tuotteet/teollisuus/referenssit/digitaltwin.html>

Siemens. Kilpailuetua virtuaalisella käyttönotolla. Luettu 15.9.2019. <https://new.siemens.com/fi/fi/tuotteet/teollisuus/referenssit/kilpailuetua-virtuaalisella-kayttoonotolla.html>

Chryssolouris, G. Makris, S. Michalos, G. Virtual Commissioning of an Assembly Cell with Cooperating Robots. Pdf-tiedosto. Luettu 17.9.2019. https://www.researchgate.net/publication/258384416_Virtual_Commissioning_of_an_Assembly_Cell_with_Cooperating_Robots

Isermann, R. Schaffnits, J. Hardware-in-the-loop simulation and rapid prototyping for the Development of Control Functions for Diesel Engines. Pdf-tiedosto. Luettu 18.9.2019. <https://www.sciencedirect.com/science/article/pii/S1474667017392455>

Siemens. Products. Luettu 15.4.2019. <https://www.plm.automation.siemens.com/global/en/products/nx/>

Siemens PLM Software. Learning Advantage. Mechatronics Concept Designer. Luettu 20.4.2019. <http://training.plm.automation.siemens.com/mytraining/home.cfm>

Siemens, Mechatronics Concept Designer. 2010. Quick start and User Manual. PDF-tiedosto. Luettu 21.4.2019. http://www.alvarestech.com/temp/PDP2011/Siemens-NXMechatronicsConceptDesigner/MCD_Quick_Start.pdf

Mechatronics Concept Designer. 2010. A functional approach to machine design. PDF-tiedosto. Luettu 21.4.2019. https://www.ideal-plm.ru/uEditor/files/4/NXCAD01_16_en_mcd_FS.pdf

Siemens. n.d. TIA portal – teollisuusautomaation ohjelmistoalusta. Luettu 13.7.2019. http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/tia_portal.php

Johdanto SCADA sähköisten insinöörien – aloittelijoille. Luettu 13.7.2019

<https://fi.electronics-council.com/an-introduction-scada-67445>

Ada The PLM specialist. n.d. Mechatronics Concept Designer. Luettu 28.10.2019.

<https://www.adacomputers.ro/en/nx-solutions/>

Zabala.J. n.d. NX 10 – Mechatronics Cocept Desgin. Pdf-tiedosto. Luettu 4.11-2019.

https://www.academia.edu/20062592/NX_10_-_Mechatronics_Concept_Design_-_Gu%C3%ADa

Heikkinen, T. n.d. OPC. Luettu 29.10.2019.

https://www.oamk.fi/~timohei/?p=20opintojaksot/0130T115303&t=10opc_fi.html

Novotek. n.d. OPC ja OPC UA. Luettu 29.10.2019.

<https://www.novotek.com/fi/ratkaisut/kepware-kommunikointialusta/opc-ja-opc-ua/>

BCT. n.d. Mechatronics Product Development and Virtual Commissioning. Luettu

20.10.2019. <https://www.bct-technology.com/en/support/tips-tricks/virtual-commissioning-mechatronics-concept-designer.html>

Siemens. 2018. Getting started with S7-PLCSIM Advanced and simulation tables. PDF-

tiedosto. Luettu 21.11.2019. https://cache.industry.siemens.com/dl/files/047/109759047/att_962042/v3/109759047_PLCSIMAdv_SimTable_DOC_V10_en.pdf

Liljanto, L. tekninen viestijä. 2019. Haastattelu 23.11.2019. Haastattelija Pakkala, A. tekninen konsultointi. Tampere.

Vento, T. 2018. NX Mechatronics Concept Designer. Konetekniikan koulutusohjelma. Oulun yliopisto. Kandidaatintyö.

Penttilä, M. 2018. Harjoituskuljettimen luonti ja Siemens Mechatronics Concept designer ohjelmiston käyttöönotto. Konetekniikan koulutusohjelma. Tampereen ammattikorkeakoulu. Opinnäytetyö.

Lantta, J. 2016. Automaatiolinjan virtuaalikäyttöönotto. Automaatiotekniikan diplominsinöörin tutkinto-ohjelma. Tampereen teknillinen yliopisto. Diplomityö.

Taipalus, J. 2015. 3D-Virtualization of a Conveyor Machine. Automation Engineering. Seinäjoki University of Applied Sciences. Thesis.

Simeon, S. 2019. Porausaseman virtuaalinen käyttöönotto. Automaatiotekniikan tutkinto-ohjelma. SeAMK Tekniikka. Opinnäytetyö.

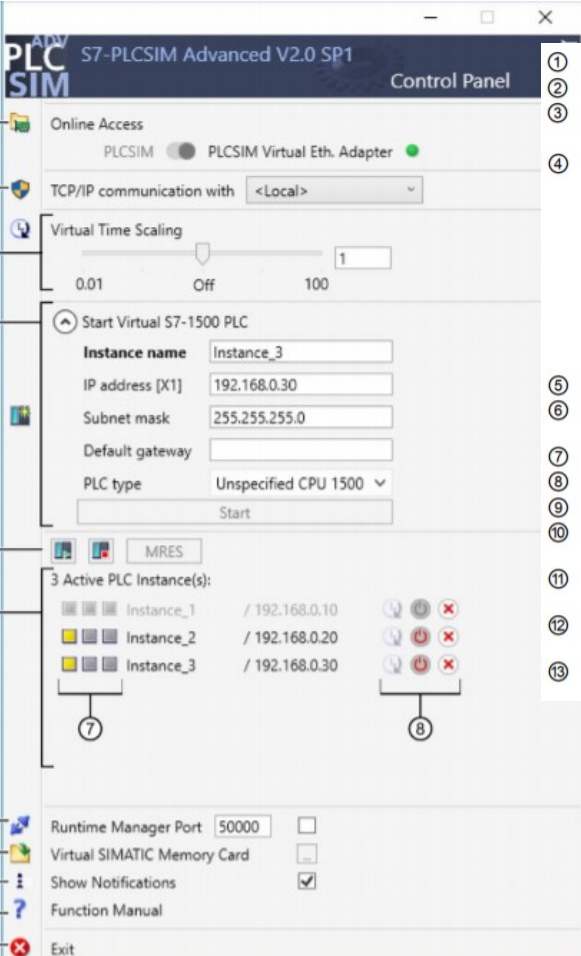
LIITTEET

Liite 1. MECHATRONICS CONCEPT DESIGN WORKFLOW

The following represents a typical machine design workflow using Mechatronics Concept Designer:

1. Define and manage design requirements.
 - Gather and structure requirements such as system response time and cost.
 - Add secondary requirements derived from the main requirements.
 - Link requirements to each other.
 - Add more details to the requirements using embedded tools like Microsoft Word.
2. Create a functional model.
 - Define basic functions of the system.
 - Create a hierarchy based on a functional decomposition.
 - Create and maintain alternatives for the functional design.
 - Reuse functional units.
 - Parameterize the output of functions and associate them with requirements.
3. Create a logical model.
 - Define the basic logical model of the system.
 - Create a hierarchy based on a logical decomposition.
 - Reuse logical blocks.
 - Parameterize the blocks and associate them with functions.
4. Create trachelinks to show the dependencies between functional units and the logical blocks.
5. Define the mechanical concept.
 - Define a rough 3D outline of the basic machine.
 - Assign mechanical objects to functional units and logical blocks.
 - Add kinematics and dynamics.
6. Add basic physics and signals.
 - Add basic physics and speed constraint and position constraint actuators.
 - Add signal adaptors.
 - Assign signal adaptor object to functional units and logical blocks.
7. Define time based operations.
 - Define how the actuators are controlled by operations.
 - Arrange the sequence of operation with a time based notion.
 - Assign operations to the corresponding functions in the function tree.
 - Assign operations to the corresponding logics in the logical tree.
8. Add sensors.
 - Add sensors that are triggered by collisions of system elements with sensor objects, or sensors that are defined by a signal adapter.
9. Define event based operations.
 - Define operations that are triggered by events that are generated by sensors or other objects in the mechatronics systems, such as the position of an actuator.
 - Assign operations to the corresponding functions in the function tree.
10. Replace concept model with detailed model and transfer physics objects from the rough geometries to the detailed ones.
11. Align sensors and actuators with ECAD.
12. Export sequence of operation in PLCOpen XML format to a PLC engineering tool like STEP 7.
13. Test PLC program via OPC connection (Siemens:learning advantage 2018).

Liite 2. S7-PLCSIM Advanced V2.0 SP1 täyttöohjeet (Siemens 2018).



The screenshot shows the S7-PLCSIM Advanced V2.0 SP1 Control Panel. The interface includes sections for Online Access, TCP/IP communication, Virtual Time Scaling, Start Virtual S7-1500 PLC, Active PLC Instance(s), Runtime Manager Port, Virtual SIMATIC Memory Card, Show Notifications, Function Manual, and Exit. Numbered callouts (1-13) point to specific elements: 1. Online Access; 2. TCP/IP communication with <Local>; 3. Virtual Time Scaling slider; 4. Start Virtual S7-1500 PLC section with fields for Instance name (Instance_3), IP address (192.168.0.30), Subnet mask (255.255.255.0), Default gateway, and PLC type (Unspecified CPU 1500); 5. MRES button; 6. Active PLC Instance(s) list showing Instance_1, Instance_2, and Instance_3; 7. LED displays; 8. Instance list; 9. Runtime Manager Port (50000); 10. Virtual SIMATIC Memory Card; 11. Show Notifications (checked); 12. Function Manual; 13. Exit button.

①	Online access	Switch to select the communication interface
②	TCP/IP communication	Selection of network adapter for distributed communication
③	Virtual time	Slider to adjust the scaling factor
④	Start Virtual S7-1500 PLC	Opens and closes the text boxes for creating the instance (virtual controller). Here you enter a unique name for the instance. Enter a minimum of 3, a maximum of 64 characters. If the name is unique in the network, the button "Start" is enabled.
	• Name of the instance	
	• IP address	The input boxes are visible when you switch the communication interface to "PLCSIM Virtual Ethernet Adapter". The IP address is entered automatically.
	• Subnet mask	
	• Standard gateway	
	• CPU type	Here you select the type of CPU to be simulated.
	• "Start" button	Create with the button and start the instance.
⑤	Buttons	Buttons for operating the selected instances.
⑥	Instance list	The list shows the available local instances. The instances can be resorted using the mouse cursor.
⑦	LED displays	The meaning of the LED is displayed when you move the mouse over it.
⑧	Icons	Icons for operating the instance
⑨	Runtime Manager Port	Here you open a port on the local PC.
⑩	Virtual SIMATIC Memory Card ...	Open an Explorer window here in which you select the path to the virtual memory card.
⑪	Display messages	Here you disable the PLCSIM Advanced messages in the Windows task bar for the duration of the operation.
⑫	Function manual	This is where you open the S7-PLCSIM Advanced Function Manual in a standard PDF viewer.
⑬	Exit	Exit logs off all instances and closes the Control Panel.