



Open your mind. LUT.  
Lappeenranta University of Technology

# **HAJAUTETUN AUTOMAATION KÄYTTÖ OSANA SIMULAATIOTA**

Lasse Kuparinen

Kandidaatintyö  
LUT School of Energy Systems  
Sähkötekniikka

31.5.2016

# TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto  
LUT School of Energy Systems  
Sähkötekniikka

Lasse Kuparinen

## Hajautetun automaation käyttö osana simulaatiota

2016

Kandidaatintyö  
25 sivua, 18 kuvaa

Tarkastaja: TkT Tuomo Lindh

Hakusanat: hajautettu automaatio, HIL-simulaatio, kenttäväylä, synkronointi, taajuusmuuttaja, toimintolohko

Työssä perehdytään simulaatiotekniikkaan, jossa käytetään osana fyysistä laitteistoa, ja siihen tarvittaviin komponentteihin, kuten ohjelmistorajapintoihin sekä kenttäväylään. Työssä tutustutaan myös IEC 61131-3 ja IEC 61499 -standardien mukaisiin toimintolohkomalleihin, joita käytetään automaatiassa. Näiden perusteella kehitetään järjestelmä, jonka avulla Simulink-ympäristössä voidaan oman toimintolohkon avulla käyttää taajuusmuuttajaa osana simulaatiota. Tällaisen reaaliaikaisen systeemin eri osien väliseen synkronointiin kiinnitetään myös huomiota. Järjestelmää testataan simulaatiomallilla, jossa syötetään vääntömomentti-ohje taajuusmuuttajalle, joka mittaa siihen kytketyn moottorin pyörimisnopeuden. Mallilla voidaan esimerkiksi arvioida kuorman hitausmomentti. Työssä tutustutaan myös taajuusmuuttajien ohjelmallisiin ominaisuuksiin ja niiden perusteella pohditaan esitetyn kaltaisten järjestelmien käyttöä hajautettuna automaatiojärjestelmänä. Kokeellinen työ tehtiin säätö- ja digitaalitekniikan laboratoriossa vuoden 2010 aikana.

# **ABSTRACT**

Lappeenranta University of Technology  
LUT School of Energy Systems  
Electrical Engineering

Lasse Kuparinen

## **Distributed automation as a part of simulation**

2016

Thesis for the Degree of Bachelor of Science in Technology

25 pages, 18 figures

Examiner: D.Sc. Tuomo Lindh

Keywords: distributed automation, fieldbus, frequency converter, function block, Hardware-in-the-Loop simulation, synchronization

In this thesis Hardware-in-the-Loop simulation and the required components, such as software interfaces and the fieldbus, are studied. The function block models for automation according to IEC 61131-3 and IEC 61499 standards are also introduced. On this basis, a system is developed for using a frequency converter as a function block in a Simulink simulation. The synchronization of such real-time system is also studied. The proposed system is tested with a simulation model in which a torque reference is sent to the frequency converter and the rotational speed of an attached drive is received as a result. The model can be used for estimating the inertia of the load, for instance. The programming features of frequency converters and the use of the proposed simulation system as a distributed automation system are also discussed. The experimental work was conducted in the laboratory of control engineering and digital systems during 2010.

## SISÄLLYSLUETTELO

KÄYTETYT MERKINNÄT JA LYHENTEET .....	2
1. JOHDANTO .....	3
1.1 Tausta .....	3
1.2 Tavoitteet .....	3
1.3 Työn rakenne .....	4
2. SIMULAATIO JA AUTOMAATIO .....	5
2.1 HIL-simulaatio .....	5
2.2 Kenttäväylät ja Modbus .....	6
2.3 Taajuusmuuttajien ohjelmalliset ominaisuudet .....	7
2.4 Hajautettu automaatio .....	9
3. KEHITETTY JÄRJESTELMÄ .....	11
3.1 Taajuusmuuttajan kytkentä ja tietoliikenne .....	11
3.2 Taajuusmuuttajan Simulink-lohko .....	12
3.3 Taajuusmuuttajan parametrit ja ohjelmointi .....	14
3.4 Reaaliaikaisuus ja synkronointi .....	16
4. TESTAUS .....	18
4.1 Järjestelmän käyttö ja toiminta .....	18
4.2 Testisimulaatio .....	19
4.3 Ongelmat .....	20
5. YHTEENVETO .....	22
5.1 Työn tulokset .....	22
5.2 Jatkokehitysmahdollisuudet .....	23
LÄHTEET .....	24

## KÄYTETYT MERKINNÄT JA LYHENTEET

AS-i	Actuator Sensor Interface, kenttäväylä
ASCII	American Standard Code for Information Interchange, tässä yhteydessä Modbus-protokollan toimintatila, viittaa merkistönkoodausmenetelmään
CAN	Controller Area Network, kenttäväylä
CODESYS	Controller Development System, ohjelmointiympäristö
CRC	Cyclic Redundancy Check, virheentunnistuksessa käytettävä tarkistussumma
EIA	Electronic Industries Alliance, entinen elektroniikkavalmistajien standardointijärjestö
HIL	Hardware-in-the-Loop, simulaatiotekniikka
I/O	Input/Output, laitteen tulo- ja lähtöliitännät
IEC	International Electrotechnical Commission, kansainvälinen sähkötekniikan standardointijärjestö
IP	Internet Protocol, tietoliikenneprotokolla
MATLAB	Matrix Laboratory, laskentaohjelmisto
PC	Personal Computer, henkilökohtainen tietokone
PID	Proportional Integral Derivative, säätötekniikan perussäädin
PROFIBUS	Process Field Bus, kenttäväylä
RS-232	Recommended Standard 232, standardi sarjamuotoiselle tiedonsiirrolle, myöhemmin tunnettu myös nimillä EIA-232, TIA-232 ja EIA/TIA-232
RS-485	Recommended Standard 485, standardi sarjamuotoiselle tiedonsiirrolle, myöhemmin tunnettu myös nimillä EIA-485, TIA-485 ja EIA/TIA-485
RTU	Remote Terminal Unit, tässä yhteydessä Modbus-protokollan toimintatila, viittaa etäohjattavaan päätelaitteeseen
TCP	Transmission Control Protocol, tietoliikenneprotokolla
TIA	Telecommunications Industry Association, tietoliikennetekniikoiden standardointiyhdistys
TwinCAT	The Windows Control and Automation Technology, automaatio-ohjelmisto
XML	Extensible Markup Language, merkintäkieli

# 1. JOHDANTO

Automaatio ja sähkökäytöt kehittyvät jatkuvasti älykkäämpään suuntaan. Laitteistojen suorituskyky paranee ja ohjelmallisten toiminnallisuuksien määrä kasvaa. Lisäksi tietoliikenne-ratkaisut kehittyvät ja mahdollistavat automaatiojärjestelmien hajauttamisen useille laitteille tehokkuuden ja käytettävyyden parantamiseksi. Myös simulaatioiden merkitys suunnittelu- ja kehitystyössä kasvaa ja niissä pystytään hyödyntämään monipuolisempia malleja ja menetelmiä.

## 1.1 Tausta

Taajuusmuuttajilla pystytään suorittamaan muitakin tehtäviä kuin vääntömomentin tuottamista moottorille. Niistä löytyy ohjelmallisia ominaisuuksia erilaisiin tarkoituksiin. Ohjelmien avulla voidaan saada tietoa säädettävän prosessin tilasta ja hyödyntää sitä optimointiin sekä kunnonvalvontaan. Prosessi tai sen osia voidaan myös identifioida mittaamalla sen parametreja, esimerkiksi säätöjen virittämistä varten. Hajautettavat sovellukset mahdollistavat myös etähallinnan sekä taajuusmuuttajan käytön osana simulaatiota.

## 1.2 Tavoitteet

Työssä tutkitaan taajuusmuuttajan kytkemistä simulaatio-ohjelmistoon ja sen ohjelmallisten ominaisuuksien hyödyntämistä simulaatiossa. Tavoitteena on kehittää järjestelmäkomponentteja, joiden avulla muodostetaan rajapinta simulaattorin ja taajuusmuuttajan välille. Pääasiassa erilaisin ohjelmointiratkaisuin pyritään toteuttamaan tietoliikenne siten, että koko järjestelmää pystytään ohjaamaan simulaattorina toimivalta tietokoneelta. Tarkoituksena on myös tutkia ja ratkaista muun muassa reaaliaikaisuuteen ja synkronointiin liittyviä ongelmia ja vaatimuksia. Tavoitteena on myös arvioida taajuusmuuttajan soveltuvuutta kyseiseen järjestelmään testaamalla mitä sen ohjelmilla voidaan tehdä. Lisäksi pohditaan myös mahdollisuutta käyttää järjestelmää standardin mukaisena hajautettuna automaatiojärjestelmänä lohkomallin mukaisesti.

### **1.3 Työn rakenne**

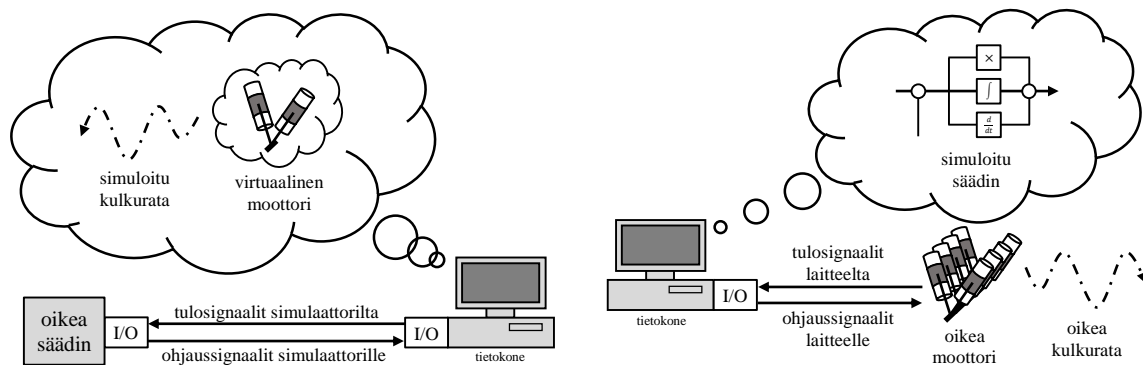
Työ rakentuu siten, että käsiteltäviä aiheita esitellään ensin teoriatasolla ja siltä pohjalta laajennetaan käytännön toteutuksiin ja testaustuloksiin. Luvussa 2 perehdytään muutamiin automaatio- ja simulaatiojärjestelmien teknisiin käytäntöihin sekä yleisten standardien sisältöön. Luvussa 3 esitellään työn aikana kehitetyn järjestelmän osien toteutusta ja teknisiä ratkaisuja. Luvussa 4 tarkastellaan järjestelmän toimintaa ja testaustuloksia. Luvussa 5 kootaan yhteen koko työn tulokset sekä arvioidaan tavoitteiden saavuttamista ja tulevaisuuden kehittymismahdollisuuksia.

## 2. SIMULAATIO JA AUTOMAATIO

Simulaatiossa ja automaatiassa on käytössä useita vakiintuneita ratkaisuja. Perinteisen ohjelmistosimulaation ja prototyyppilaitteiston välimuotona voidaan yhdistellä laitteita simulaatioon tai myös ihminen voi toimia osana antamalla syötteitä. Automaatiossa käytetään erilaisia tietoliikenne- ja ohjelmointitekniikoita, jotka soveltuvat eri tarkoituksiin. Teollisuudessa yleistynyt käytäntö on myös hajautettu automaatio.

### 2.1 HIL-simulaatio

HIL-simulaatio (Hardware-in-the-Loop) on simulaatiotekniikka, jossa osa testattavista komponenteista on korvattu fyysisellä laitteistolla. Tyypillisesti sitä käytetään säätösuunnittelussa, jolloin simuloimalla testattu säädin suoritetaan tarkoitukseen valitulla laitteistolla, kuten ohjelmoitavalla logiikalla tai sulautetulla järjestelmällä. Toinen vaihtoehto on toimilaitteen simulaatiomallin korvaaminen oikealla laitteella, kuten moottorilla (Baracos, 2001). Molemmilla, kuvassa 2.1 esitetyillä, menetelmillä on omat käyttötarkoituksensa ja etunsa verrattuna tavalliseen simulaatioon.



Kuva 2.1 Laitteistopohjainen säädin osana simulaatiota ja oikean toimilaitteen käyttö säätösimulaatiossa. (Baracos, 2001)

Merkittävin HIL-simulaatioon liittyvä asia on reaaliaikaisuus. Koska simulaatioon kytketty laitteisto toimii luonnollisesti reaaliajassa, on koko simulaatio synkronoitava toimimaan samassa tahdissa. Säätösuunnittelussa pelkkä säädinmallin ohjelmistopohjainen simulointi ei vielä riitä vahvistamaan säätimen toimintaa käytännössä. Laitteiston ja tietoliikenteen aiheuttamat suorituskykyrajoitteet reaaliaikaympäristössä voidaan todentaa HIL-simulaatiolla (Baracos, 2001). Syy HIL-simulaation käyttämiseen on usein säästäminen. Prototyyppien ja



kokonaisten järjestelmien rakentaminen ja testaaminen on aikaa vievää ja kallista. HIL-simulaatio mahdollistaa säätöjärjestelmän jatkuvan testaamisen eri kehitysvaiheissa, jolloin pystytään minimoimaan virheiden ja tarvittavien prototyyppien määrä. Suunnitteluun käytettävää työaikaa saadaan lyhennettyä ja järjestelmän toimivuus on jo paremmalla tasolla, kun varsinaista prototyyppiä päästään testaamaan ja riski kalliin laitteiston tuhoamisesta pienenee (Halvorsen, 2016; Kleijn, 2016). Lisäksi HIL-simulaatioon voidaan syöttää häiriöitä ja siten testata säätimen toimintaa erikoistilanteissa, jotka olisivat fyysisessä järjestelmässä harvinaisia (Ogan, 2015).

Mikäli säädettävä toimilaitte löytyy valmiina, HIL-simulaation avulla säätösuunnittelu voidaan aloittaa heti simuloimalla säädintä eikä toimilaitteelle tarvita simulaatiomallia. Tämä säästää aikaa, jos kyseessä on monimutkainen järjestelmä eikä mallia ole saatavilla valmiina. Esimerkiksi auto- ja lentokoneteollisuudessa, joita varten HIL-simulaatio alun perin on kehitetty, simulaatiot koostuvat lukuisista toimilaitteista ja sensoreista, jolloin mallintaminen voi olla erittäin haastavaa (Baracos, 2001). Säätösuunnittelun lisäksi HIL-simulaatiota voidaan käyttää järjestelmän identifioimiseen. Mittaamalla laitteiston toimintaa voidaan määrittää sen parametreja, esimerkiksi moottorin pyörimisnopeuden avulla kuorman hitausmomentti.

## **2.2 Kenttäväylät ja Modbus**

HIL-simulaatiossa tarvitaan tietoliikennettä simulaatiota suorittavan tietokoneen ja simulaatiosta ulkoistetun laitteiston välille. Teollisuudessa yleisiä tietoliikenteeseen käytettäviä kenttäväyliä ovat mm. AS-i (Actuator Sensor Interface), CAN (Controller Area Network), PROFIBUS (Process Field Bus), Modbus sekä monet Ethernet-verkkoihin perustuvat väylät. Alun perin ohjelmoitavien logiikoiden käyttöön suunniteltu Modbus on yksi vanhimmista ja yksinkertaisimmin toteutettavista kenttäväylistä.

Modbus on avoin ja ilmainen kenttäväyläratkaisu ja siksi hyvin yleisesti tuettu eri valmistajien laitteistoissa. Väylässä yksi laite toimii isännänä ja sen ohjaamia orjalaitteita voi olla maksimissaan 247. Kaikki käskyt tulevat isännältä ja orjat voivat pyydettyäessä lähettää sille takaisin dataa. Tällöin isännän on säännöllisesti kyseltävä orjilta tietoja, sillä ne eivät voi

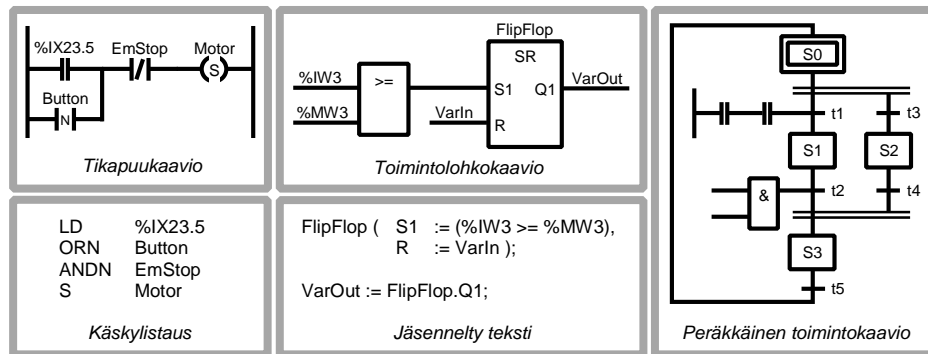
itsenäisesti raportoida mahdollisista muutoksista. Protokollan toiminta perustuu funktiokoodeihin, joiden perusteella laitteet joko lukevat tai kirjoittavat datarekistereitään tai I/O (Input/Output) -porttejaan.

Modbus-protokolla on yksinkertainen ja siitä on olemassa muutama eri versio, joiden toimintamallit ovat kuitenkin samanlaiset. Sarjamuotoista tietoliikennettä varten on kaksi variaatiota. Modbus ASCII (American Standard Code for Information Interchange) lähettää datan tekstimuotoisena ja sallii jopa yhden sekunnin mittaisen viiveen merkkien välillä ilman virheitä. Modbus RTU (Remote Terminal Unit) käsittelee datan binäärimuodossa tiiviimmin kuin ASCII-tila, jolloin raakadataa pystytään välittämään enemmän käyttäen samaa lähetysnopeutta. Data täytyy kuitenkin lähettää jatkuvana virtana, jolloin järjestelmältä vaaditaan usein puskurointia katkosten välttämiseksi. Sarjamuotoisten väylien lisäksi on olemassa myös Modbus TCP (Transmission Control Protocol), joka toimii TCP/IP (Internet Protocol) -protokollapinon päällä, esimerkiksi Ethernet-verkoissa. Tässä variaatiossa ei tarvita erillistä virheentarkastelua, koska se sisältyy alempien protokollien rakenteeseen. Sen sijaan sarjamuotoisissa versioissa tarvitaan erilaisten tarkistussummien laskentaa. (Modicon, 1996; Modbus.org, 2006)

### **2.3 Taajuusmuuttajien ohjelmalliset ominaisuudet**

Ohjelmoitavat logiikat ovat automaatiojärjestelmien perustyökaluja. Niiden tarkoituksena on ohjata mitta- ja toimilaitteita niihin ohjelmoidun toimintaperiaatteen mukaan. Nykyään vastaavia ohjelmallisia ominaisuuksia löytyy myös taajuusmuuttajista, jolloin erillisen logiikan käyttö taajuusmuuttajaohjatussa järjestelmässä ei ole välttämätöntä. Logiikoiden ohjelmoimiseen on olemassa lukuisia eri valmistajien menetelmiä, mutta IEC 61131-3 (International Electrotechnical Commission) -standardi määrittelee ensimmäiset valmistaja- ja laitteistoriippumattomat ohjelmointikielet. Standardin tarkoituksena on yksinkertaistaa ohjelmoitavien logiikoiden käyttöä yhdenmukaistamalla käytettävät ohjelmointikielet ja mahdollistamalla siten ohjelmistojen luomisen sekä siirtämisen eri laitteille. Se käsittää kuitenkin niin valtavan määrän yksityiskohtia, että sen noudattamista pidetään enemmän ohjenuorana kuin tarkkana vaatimuksena. Standardi itsessään sisältääkin vaatimustaulukot sekä testimenetelmän, joiden avulla valmistajien täytyy määritellä miltä osin tuotteet toteuttavat standardia.

IEC 61131-3 sisältää viisi kuvan 2.2 mukaista ohjelmointikieltä. Tekstimuotoisia kieliä ovat käskylistaus (Instruction List), joka muistuttaa matalan tason konekieltä ja on jo standardin uusimmassa painoksessa määritelty vanhentuneeksi, sekä jäsennelty teksti (Structured Text), joka on korkean tason lausepohjainen kieli. Graafisia kieliä ovat tikapuukaavio (Ladder Diagram), jossa signaalitie kulkee vasemmasta kiskosta oikeaan muodostaen kytkentöjä lähinnä binäärilogiikkaan perustuen, sekä toimintolohkokaavio (Function Block Diagram), jonka toiminta perustuu erilaisten lohkojen välisiin yhteyksiin ja signalointiin. Viides kieli on pääasiassa graafinen peräkkäinen toimintokaavio (Sequential Function Chart), jonka avulla voidaan luoda peräkkäisiä tai rinnakkaisia prosesseja muilla kielillä toteutetuista ohjelmaosuuksista. Muuttuja- ja funktiomäärittelyt ovat yhteisiä elementtejä, joten eri kieliä voidaan käyttää sekaisin saman ohjelman sisällä. (John, 2010)

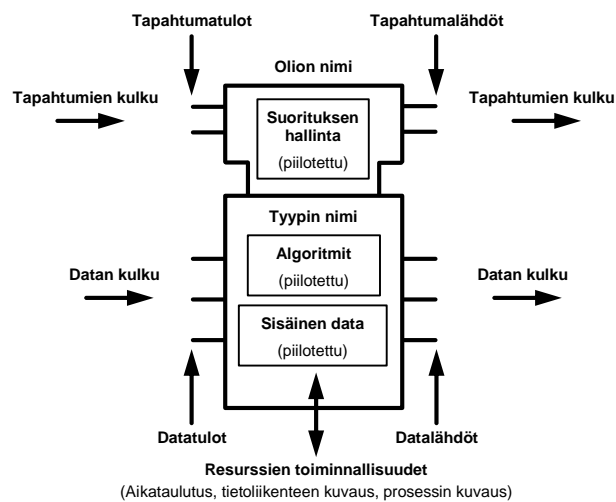


Kuva 2.2 IEC 61131-3 sisältää viisi keskenään yhteensopivaa ohjelmointikieltä, joista osa on tekstimuotoisia ja osa graafisia. (John, 2010)

Suurimmista suomalaisista taajuusmuuttajavalmistajista mm. ABB:llä ja Vacon:lla on omat kehitysokalunsa IEC 61131-3 -standardin mukaisten ohjelmien tekemiseen ja suorittamiseen taajuusmuuttajillaan. Visedo sen sijaan on ensimmäinen suomalainen valmistaja, joka on ottanut käyttöön tuotteilleen CODESYS (Controller Development System) -ohjelmointiympäristön (Visedo, 2011). CODESYS on yksi tehokkaimmista IEC 61131-3 -ohjelmointiyökaluista ja se tukee kaikkia standardin ohjelmointikieliä sekä sisältää automaatioon tarkoitettuja laajennuksia. Ohjelmistolla voidaan kääntää ohjelmakoodi suoraan alkuperäiskäskyiksi hyvin suurelle määrälle eri alustoja ja siitä löytyy tuki useille kenttäväylille. Lisäksi ohjelmiston lisenssi on ilmainen, joten sadat laitteistovalmistajat ovat ottaneet CODESYS:n käyttöön tuotteissaan. (CODESYS, 2016)

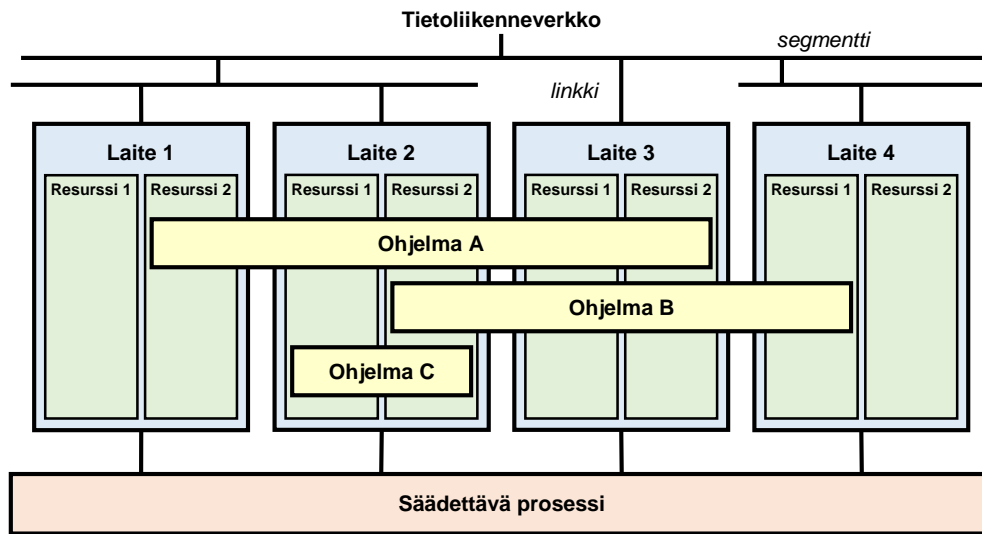
## 2.4 Hajautettu automaatio

IEC 61131-3 -standardin ohjelmointikielistä toimintolohkokaavio vastaa simulaatioissa yleisimmin käytettyä lohkokaaviota, joten se on looginen tapa ohjelmoida esimerkiksi HIL-simulaatioissa ulkoista laitetta. Toimintolohkoja suoritetaan yleensä synkronisesti tietyin aikavälein, mutta myös asynkroninen toiminta olisi tarpeellista HIL-simulaation kaltaisessa reaaliaikaisessa järjestelmässä. Muiden muassa tähän tarpeeseen on kehitetty IEC 61499 -standardi. Se käsittää kuvan 2.3 mukaisen lohkomallin, jonka sisällä voidaan suorittaa IEC 61131-3 -standardin mukaisia ohjelmia. Lohkon toimintaa ohjataan tapahtumapohjaisesti erillisillä signaaleilla. Lohkoa käsitellään yksittäisenä oliona ja sen sisäinen toiminta, kuten algoritmit, muuttujat ja suorituksen hallinta, on piilotettu. (Yoong, 2015)



Kuva 2.3 IEC 61499 -standardin lohkomallissa lohkon sisäinen rakenne on piilotettu ja sen toimintaa ohjataan tapahtuma- ja datasiinaaleilla. (IEC, 2005)

Toinen IEC 61499 -standardin tärkeä ominaisuus on hajautettu automaatio. Kuvan 2.4 mukaisesti ohjelmien suoritus voidaan jakaa eri laitteille ja niiden sisäisille resursseille. Tarkoituksena on mahdollistaa laajojen ja monimutkaisten automaatiojärjestelmien hallinta vaivattomasti. Samalla järjestelmän tehokkuus paranee, kun useiden laitteiden suorituskapasiteetti saadaan hyötykäyttöön. Hajautettu malli soveltuu myös HIL-simulaatioon, jossa osia simulaatiosta voidaan suorittaa eri laitteistoilla.



Kuva 2.4 Hajautetussa automaatioissa sovelluksia ja niiden osia voidaan suorittaa rinnakkain eri laitteilla ja niiden resursseilla. (IEC, 2005)

IEC 61499 sisältää myös vaatimuksia laitteille ja ohjelmille. Standardin mukaisten laitteiden täytyy valmistajasta riippumatta pystyä toimimaan yhdessä ja niiden on oltava konfiguroitavissa kaikilla IEC 61499 -yhteensopivilla työkaluilla. Ohjelmien täytyy olla siirrettävissä eri valmistajien ohjelmointiympäristöjen välillä. Lisäksi standardi sisältää ylläpitoa varten hallintamallin, jonka avulla laitteet ovat konfiguroitavissa ja ohjelmistot päivitettävissä ajonakaisesti ilman käyttökatkoja. (Rooker, 2009)

### 3. KEHITETTY JÄRJESTELMÄ

HIL-simulaatiota varten kehitettiin testausjärjestelmä, jossa Microsoft Windows -käyttöjärjestelmällä toimivalla tietokoneella suoritetaan MathWorks:n MATLAB (Matrix Laboratory) -laskentaohjelmistoa ja sen Simulink-simulaatioympäristöä. Tietokoneeseen on kytketty ABB:n valmistama ACSM1-taajuusmuuttaja Modbus-kenttäväylän avulla. Järjestelmän tarkoituksena on testata tiedonsiirtoa simulaattorin ja taajuusmuuttajan välillä sekä tutkia taajuusmuuttajan ohjelmallisten ominaisuuksien käyttöä.

#### 3.1 Taajuusmuuttajan kytkentä ja tietoliikenne

Tavallisten PC (Personal Computer) -tietokoneiden tietoliikennesovutuksista yksinkertaisin käyttää on sarjaportti. Teollisuuden laitteistoista löytyy yleensä suoraan useita portteja erilaisia väyliä varten, mutta kotitietokoneista löytyvä sarjaportti käyttää RS-232 (Recommended Standard 232) -standardin mukaista signalointia. ACSM1-taajuusmuuttajan ohjausyksikön porteilla on mahdollista jossain määrin ohjata laitteen toimintaa, mutta kenttäväyläratkaisua varten siihen on liitettävä erillinen kenttäväylämoduuli. Testijärjestelmässä kenttäväylänä on Modbus RTU, jonka käyttämää RS-485 (Recommended Standard 485) -standardin mukaista signalointia varten taajuusmuuttajassa käytetään FSCA-01-moduulia, kuten on esitetty kuvassa 3.1.



Kuva 3.1 ACSM1-taajuusmuuttaja ja siihen liitettävä FSCA-01-kenttäväylämoduuli, joka käyttää Modbus RTU -kenttäväylää. (ABB, 2007; ABB, 2011)

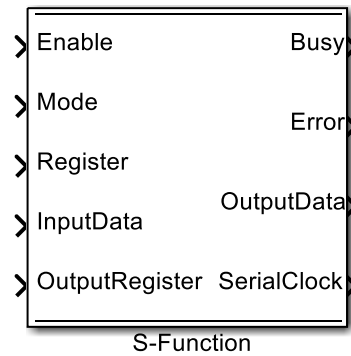
RS-232-signaalit ovat balansoimattomia eli yhteismuotoisia ja RS-485-signaalit balansoituja eli eromuotoisia. Tämän vuoksi tarvitaan erillinen muunnin, jotta tietokoneen ja taajuusmuuttajan sarjaportteja voidaan käyttää keskenään. Tiedonsiirtonopeus ja -asetukset täytyy asettaa kummassakin päässä vastaamaan toisiaan. ACSM1 tukee yleisimpiä nopeuksia välillä 9,6 - 115,2 kb/s. Asetusten muokkaamiseen voidaan käyttää ABB:n DriveStudio-ohjelmaa.

Modbus RTU -protokollan toteutukseen Windows-käyttöjärjestelmässä käytetään erillistä Visual C++ -kielellä koodattua ohjelmaa. Sarjaportin käyttöön löytyy valmiita luokkia ja Modbus RTU -ominaisuuksien lisääminen vaatii lähinnä binääridatan käsittelyn, funktio-koodien ja CRC (Cyclic Redundancy Check) -tarkistussumman laskennan toteutusta. Windows-sovellus toimii väylässä isäntänä ja taajuusmuuttaja orjana. Taajuusmuuttajalle täytyy DriveStudio:n avulla määrittää laiteosoite, johon viestit lähetetään. Modbus-käskyillä pystytään lukemaan ja kirjoittamaan taajuusmuuttajan rekistereitä, joiden avulla sen toimintaa voidaan ohjata.

### **3.2 Taajuusmuuttajan Simulink-lohko**

HIL-simulaation toteutuksessa taajuusmuuttajan ohjaamiseen tarvittavia Modbus-käskyjä täytyy pystyä lähettämään Simulink-ympäristöön luodusta simulaatiomallista käsin. Tätä varten käytetään taajuusmuuttajan rekistereitä kuvaavaa lohkoa, joka on toteutettu S-funktiona. S-funktio on Simulink-lohko, jonka toiminta on ohjelmoitu käyttäen MATLAB:n omaa ohjelmointikieltä tai C, C++ tai Fortran-kieltä. Jälkimmäiset vaihtoehdot täytyy kääntää MEX-tiedostoiksi ennen kuin niitä voidaan suorittaa simulaatiossa. S-funktiolle voidaan määrittää tulo- ja lähtösignaaleja, kuten muillekin Simulink-lohkoille, ja sen toimintaa suoritetaan määritetyn näytteistysajan välein.

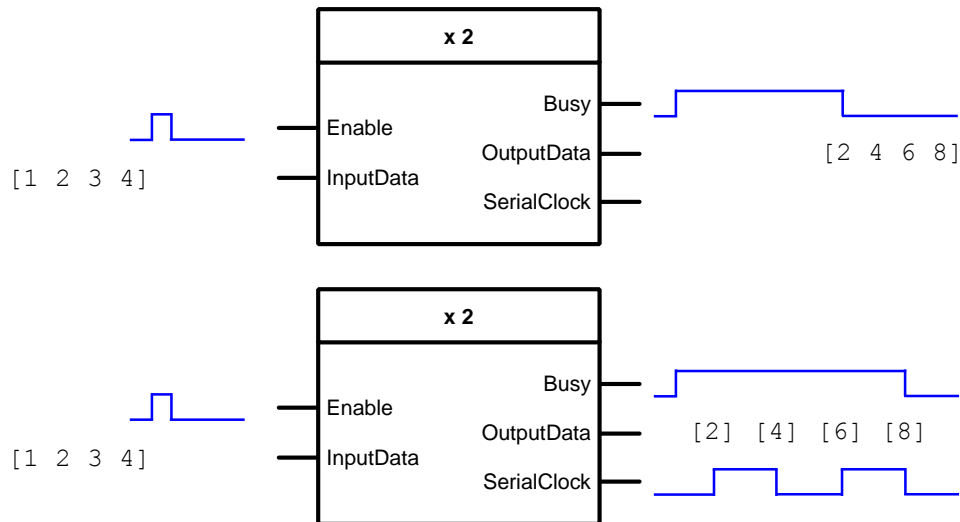
Taajuusmuuttajan loholla on yhteys erikseen suoritettavaan Modbus-sovellukseen, jotta se pystyy käyttämään kenttäväylää. Yhteyden muodostamiseen on useita tapoja ja yksinkertaisen toteutuksen vuoksi siihen käytetään erillistä tiedostoa. Sekä Simulink-lohko että Modbus-sovellus suorittavat aikavälein tilakonetta sekä lukevat ja kirjoittavat tarvittaessa yhteistä tiedostoa. Menetelmä ei ole kovin tehokas, mutta se on hitaassa suorituksessa riittävän toimiva ja samalla voidaan helposti myös kirjata suorituskäytännön toimintaa talteen tiedostoon.



Kuva 3.2 S-funktiona toimiva Simulink-lohko, jolla voidaan kontrolloida taajuusmuuttajan rekistereitä.

Kuvassa 3.2 esitetty Simulink-lohko piilottaa sisäänsä kenttäväylän toiminnan ja toimii kuin simulaatiolla olisi suora yhteys taajuusmuuttajan rekistereihin. Tulosignaaleilla voidaan kontrolloida lohkon toimintaa. Liipaisemalla Enable-signaali ylös käynnistetään lohko suorittamaan Mode-signaalilla valittua toimintaa. Eri toimintatiloilla voidaan joko lukea tai kirjoittaa rekistereitä tai tehdä molemmat sekvenssinä. Register- ja InputData-tuloja käytetään määrittämään operoitava rekisteri ja kirjoitettava data. Tulosignaaleina voidaan käyttää myös vektoreita, jolloin useita rekistereitä voidaan lukea tai kirjoittaa yhdellä käskyllä. Sekvenssikirjoituksessa syötetään enemmän dataa, joka yksitellen ensin kirjoitetaan valittuun rekisteriin ja luetaan OutputRegister-tulon määrittelemästä rekisteristä dataa taajuusmuuttajalta. Tätä toimintatilaa voidaan käyttää, esimerkiksi jos tiettyjen rekisterien tilat vaikuttavat toisiinsa ja halutaan tarkkailla vaikutuksia jokaisella yksittäisellä muutoksella. Saman toiminnan voi toteuttaa peräkkäisillä kirjoitus- ja lukukäskyillä, mutta toimintatilan tarkoitus on mahdollistaa suoritus yhdellä käskyllä ja nopeammin. Sekvenssikirjoitus voidaan tehdä rinnakkaismuotoisesti, jolloin suoritus tapahtuu yhtenäisesti ja sen päätyttyä OutputData-lähdöstä voidaan lukea vektori, joka sisältää vektorina yhtä monta lukutulosta kuin InputData-tulo sisältää kirjoitusarvoja. Toinen vaihtoehto on suorittaa sekvenssikirjoitus sarjamuotoisesti, jolloin jokaisen yksittäisen kirjoitus- ja lukuoperaation jälkeen OutputData-lähdöstä on luettavissa luetun rekisterin arvo ja valmiin tuloksen merkinä SerialClock-lähtösignaali vaihtaa tilaansa. Esimerkit sekvenssikirjoituksista on esitetty kuvassa 3.3.

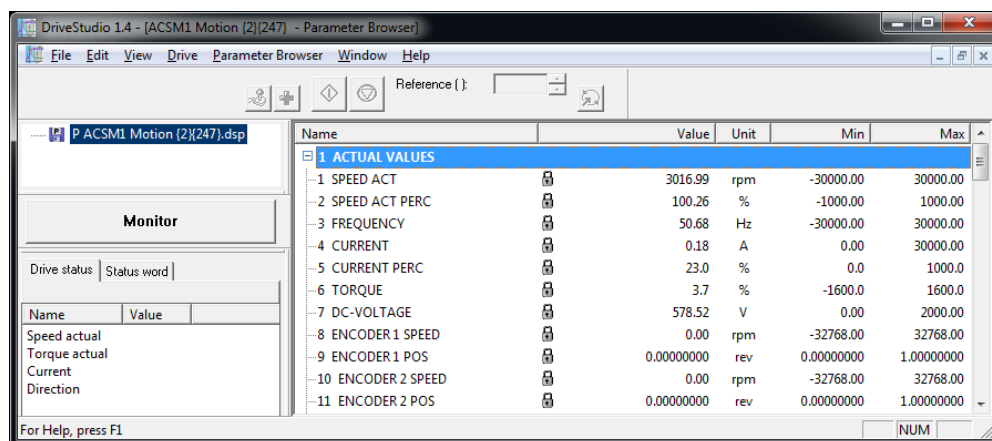




Kuva 3.3 Sekvenssikirjoituksen toiminta rinnakkaismuotoisena ja sarjamuotoisena.

### 3.3 Taajuusmuuttajan parametrit ja ohjelmointi

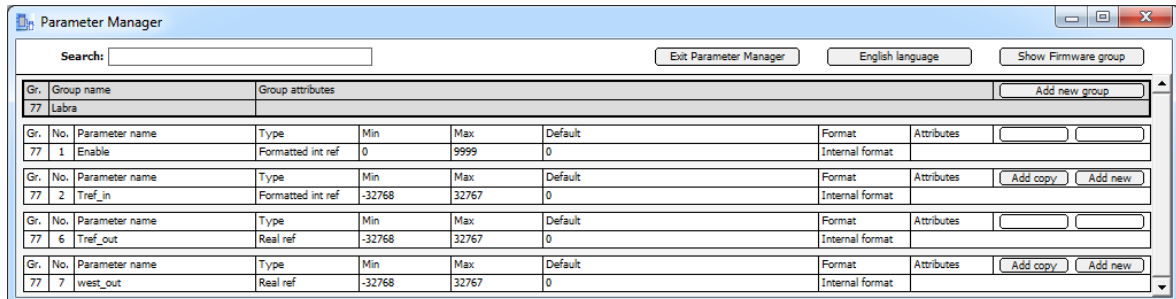
ACSM1-taajuusmuuttajan rekisterit sisältävät runsaasti erilaisia parametreja, joita voidaan tarkastella ja muokata DriveStudio-ohjelmalla. Osa parametreista sisältää asetuksia taajuusmuuttajan toiminnan ohjaamiseen ja osasta voidaan lukea järjestelmän arvoja, kuten moottorin pyörimisnopeus tai tietoliikenteessä tapahtuneiden virheiden määrä. Kuvassa 3.4 on näkymä ohjelman parametrilistauksesta.



Kuva 3.4 DriveStudio-ohjelman parametrinäkömä. Eri parametriryhmistä löytyy parametreja toiminnan ohjaamiseen tai järjestelmän arvojen lukemiseen.

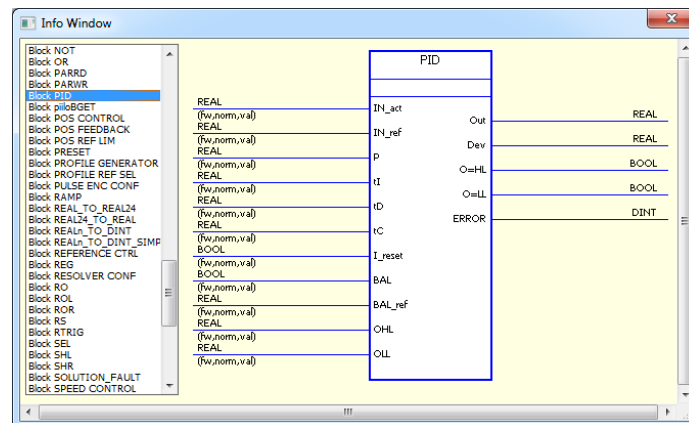
Valmiiseen parametrilistaukseen pystyy kuvan 3.5 mukaisella editorilla myös lisäämään omia parametriryhmiään, joita voi käyttää ohjelmoinnissa. Esimerkiksi kenttävyöhyk-

nessä ACSM1 sisältää tietoliikennettä varten 16-bittisiä tulo- ja lähtörekistereitä 12 kappaletta kumpaakin. DriveStudio:n avulla näiden rekisterien arvot pystytään ohjaamaan suoraan taajuusmuuttajan suorittamien ohjelmien käyttöön.



Kuva 3.5 Uusien parametrien ja parametriryhmien luominen Parameter Manager:illa. Parametreille voi määrittää tietotyyppin, raja-arvot sekä oletusarvon.

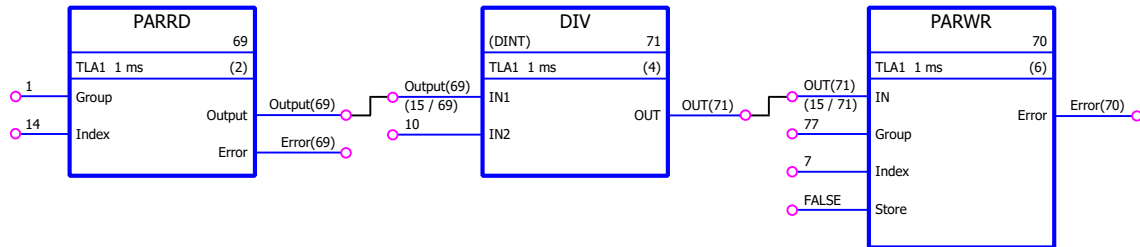
DriveSPC on ABB:n ohjelmointiympäristö, joka toimii tiiviisti DriveStudio:n kanssa. Se tukee IEC 61131-3 -standardin korkeamman tason ohjelmointikieliä eli jäsenneltyä tekstiä sekä toimintolohkokaavioita. Taajuusmuuttajan parametrit ja niiden yhteydet näkyvät graafisessa muodossa laiteohjelmiston toimintolohkoina. Kuvassa 3.6 on esitetty ohjelman toimintolohkovalikko, josta löytyy esimerkiksi PID (Proportional Integral Derivative) –säädin toteutettuna yhdellä lohkoilla.



Kuva 3.6 DriveSPC-ohjelman toimintolohkovalikko. Valikoimasta löytyy laiteohjelmiston toimintolohkoja, vakio toimintolohkoja sekä teknologisia toimintolohkoja.

Ohjelmisto sisältää runsaasti standardin mukaisia vakiolohkoja, joita yhdistelemällä pystyy helposti muodostamaan omia ohjelmakokonaisuuksia. Kuvassa 3.7 on esitetty yksinkertainen lohko-ohjelma, jossa parametri luetaan taajuusmuuttajan rekisteristä, suoritetaan laskutoimenpide ja kirjoitetaan tulos toiseen rekisteriin. Jokaiselle lohkolle valitaan suorituksen

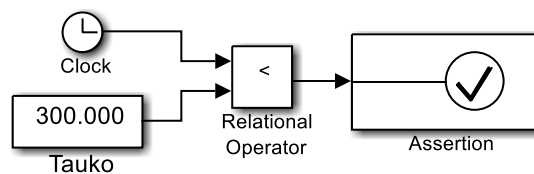
aikataso, joka on kuvan esimerkissä kaikilla yksi millisekunti. Ohjelman suoritusaikaa voi arvioida ohjelmointioppaassa (ABB, 2015) jokaiselle lohkolle annetun suoritusajan perusteella. Esimerkiksi jakolaskun laskeva DIV-lohko suoritetaan 2,55 mikrosekunnissa, kun taas parametrien luku ja kirjoitus tapahtuu vastaavasti 6,00 ja 14,50 mikrosekunnissa.



Kuva 3.7 Yksinkertainen IEC 61131-3 -standardin mukainen toimintolohko-ohjelma DriveSPC-ohjelmassa. Ohjelma lukee parametrin 1.14, jakaa sen arvon kymmenellä ja kirjoittaa tuloksen parametriin 77.07.

### 3.4 Reaaliaikaisuus ja synkronointi

HIL-simulaation reaaliaikavaatimuksen vuoksi Simulink:ssä ja taajuusmuuttajassa suoritettava laskenta täytyy synkronoida. Muussa tapauksessa Simulink suorittaisi simulaationsa hetkessä alusta loppuun eikä taajuusmuuttajan toiminta ehtisi päivittyä siihen. Simulink:iin on olemassa laajennuksia, kuten Simulink Real-Time sekä RT-LAB, joiden avulla simulaatioita voi ajaa reaaliajassa erityisesti HIL-simulaatioita varten. HIL-simulaation toteuttaminen on kuitenkin mahdollista ilman oikeaa reaaliaikasisimulaatiota ja kolmansien osapuolien lisäohjelmistoja. Yksi tapa on synkronointi jaksottamalla simulaatiota. Tässä menetelmässä laskenta tapahtuu simulaatioajassa eli yleensä nopeammin kuin reaaliajassa, mutta simulaation suoritus tauotetaan tietyin aikavälein. Toteutus on pääpiirteissään yksinkertainen, sillä kuvan 3.8 mukaisilla Simulink-lohkoilla voidaan tarkkailla simulaation suoritusaikaa ja tauottaa suoritus, kun tietty aikaraja saavutetaan.

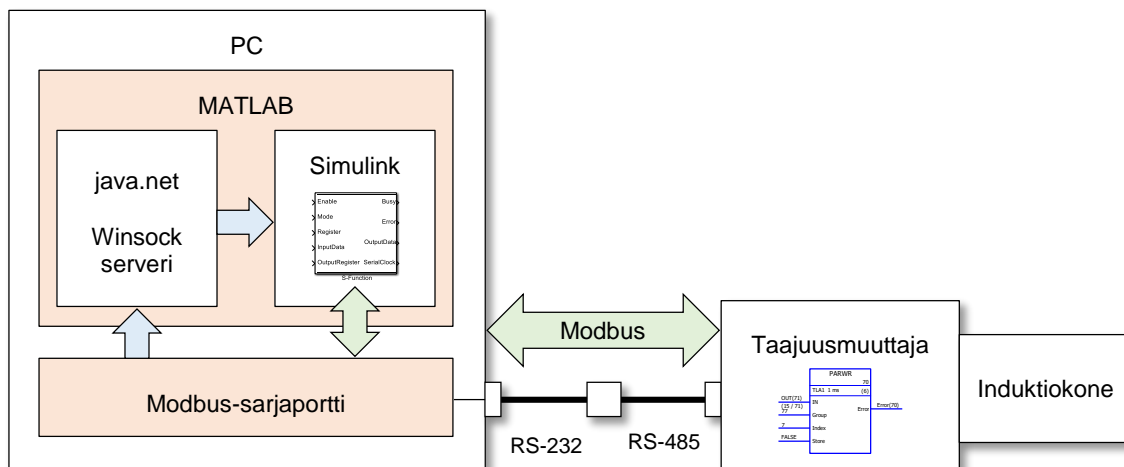


Kuva 3.8 Simulink-simulaation tauotus peruslohkojen avulla. Simulaatio keskeytetään, kun simulaatioaika saavuttaa Tauko-lohkoksa määritetyn ajankohdan.

Simulaation jatkaminen voidaan tehdä manuaalisesti Simulink:stä tai MATLAB-komennoilla. Jaksottamisen täytyy kuitenkin tapahtua automaattisesti tietyin väliajoin reaaliajassa. Toteutuksessa on hyödynnetty taustalla suoritettavaa Modbus-sovellusta. Samalla, kun se huolehtii kenttäväyläliikenteestä, se käyttää reaaliaikaista ajastinta, jonka avulla se käynnistää Simulink:n jatkamaan simulaatiota tietyin väliajoin. Aina ennen kuin uusi simulaatiojakso aloitetaan, seuraavan tauotuksen aika päivitetään kuvan 3.8 esimerkissä Tauko-lohkoon. MATLAB tukee Java-luokkien käyttöä ja niiden avulla se suorittaa simulaation aikana Winsock-serveriä, joka kuuntelee tietoliikenneporttia. Modbus-sovellus ottaa yhteyden kyseiseen porttiin ja ohjaa siten simulaation suoritusta MATLAB-komennoin.

## 4. TESTAUS

Kehitettyä järjestelmää testattiin laboratoriossa liittämällä tietokone kenttäväylän välityksellä taajuusmuuttajaan, johon oli kytketty induktiokone. Koko järjestelmän rakenne laitteistoinen ja ohjelmistoinen on esitetty kuvassa 4.1. Ensimmäisillä testiajoilla tarkasteltiin tietoliikenteen toimintaa eri osien välillä sekä määritettiin sopivat asetukset. Kun järjestelmä oli toimintakunnossa, sillä suoritettiin yksinkertainen simulaatio. Testit suoritettiin säätö- ja digitaalitekniikan laboratoriossa vuoden 2010 aikana.



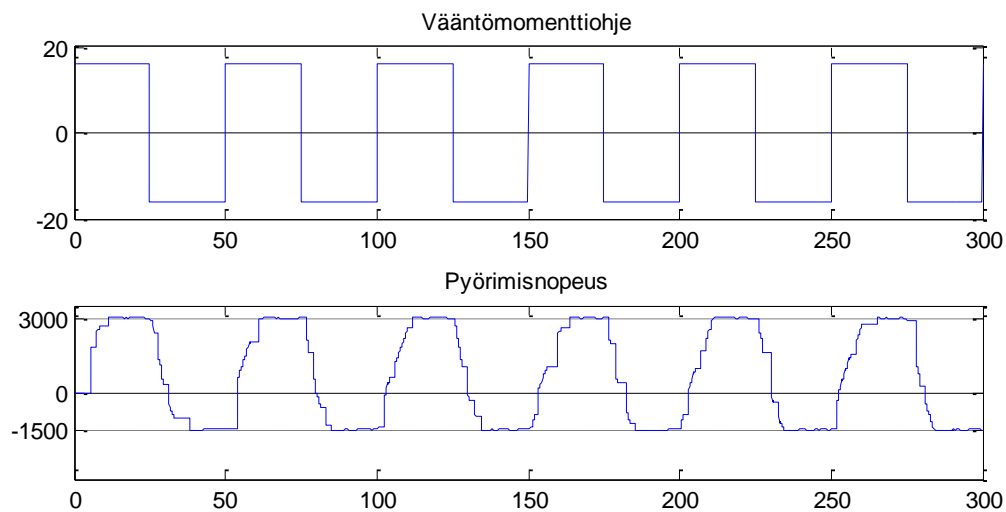
Kuva 4.1 Testijärjestelmän laitteisto koostuu tietokoneesta sekä taajuusmuuttajasta, johon on kytketty induktiokone. Kenttäväylässä tietokoneen RS-232-signaalit ja taajuusmuuttajan RS-485-signaalit muunnetaan yhteensopiviksi. Tietokoneella suoritetaan MATLAB-ohjelmistoa, Simulink-ympäristöä sekä Modbus-sovellusta.

### 4.1 Järjestelmän käyttö ja toiminta

Taajuusmuuttaja asetetaan DriveStudio:n avulla käyttämään moottorin ohjaukseen kenttäväylän välityksellä saatavaa vääntömomenttiohjetta. Ohje annetaan prosentiosuutena moottorin nimellisvääntömomentista ja arvo skaalataan valitusta kenttäväyläprofiilista riippuvalla tavalla. (ABB, 2015)

Järjestelmän käyttäminen on hieman kankeaa johtuen sen useista komponenteista. Simulaatorina toimivan tietokoneen ja taajuusmuuttajan ollessa toimintavalmiudessa tietokoneella käynnistetään MATLAB, Simulink-malli sekä Modbus-sovellus. Modbus-sovelluksella määritellään sarjaportin asetukset sekä simulaation jaksonaika ja kokonaisaika. MATLAB:n komentoriviltä käynnistetään Winsock-serveri ja Modbus-sovellus yhdistetään siihen nappia

painamalla. Tämän jälkeen simulaatio voidaan käynnistää Modbus-sovelluksen napista. Kuvassa 4.2 on esitetty viiden minuutin pituisen testiajon tulokset. Vääntömomenttiohjeena käytetään askelmaisesti suuntaansa vaihtavaa vakioarvoa. Taajuusmuuttajalta vastaanotetusta pyörimisnopeudesta havaitaan, että moottori kiihtyy ohjearvon mukaan tasaisesti kunnes se rajoitetaan taajuusmuuttajan asetuksista määritettyyn raja-arvoon, joka on 3000 kierrosta minuutissa positiiviseen pyörimissuuntaan ja 1500 kierrosta minuutissa vastakkaiseen suuntaan.

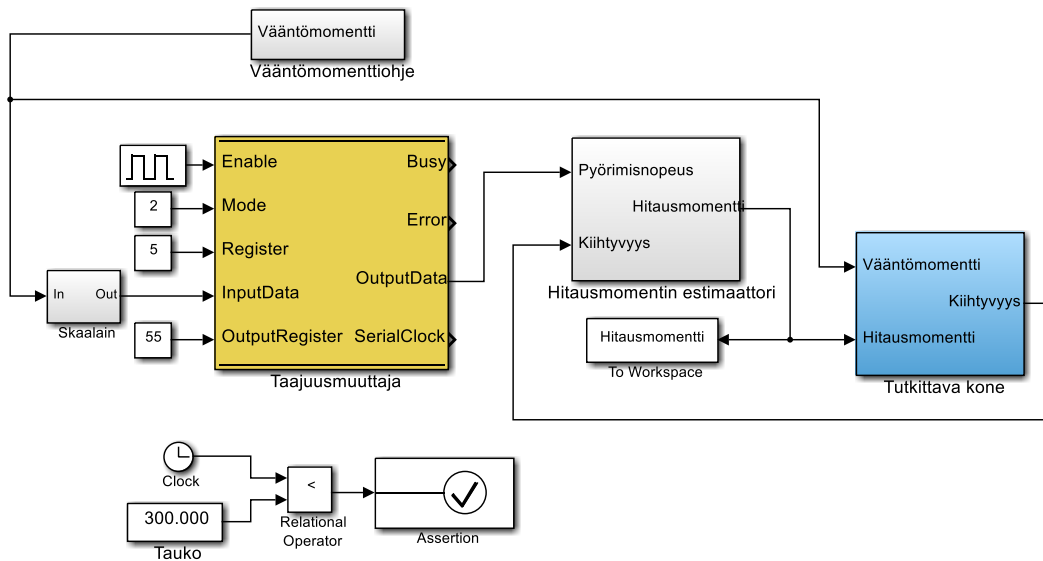


Kuva 4.2 Taajuusmuuttajalle syötetty vääntömomenttiohje ja sitä vastaava moottorin pyörimisnopeus.

Simulink-lohkossa käytetään sekvenssikirjoitusta, jossa jokaisella toimintakäskyllä syötetään vääntömomenttiohje määrättyyn rekisteriin ja vastaavasti luetaan pyörimisnopeus toisesta rekisteristä.

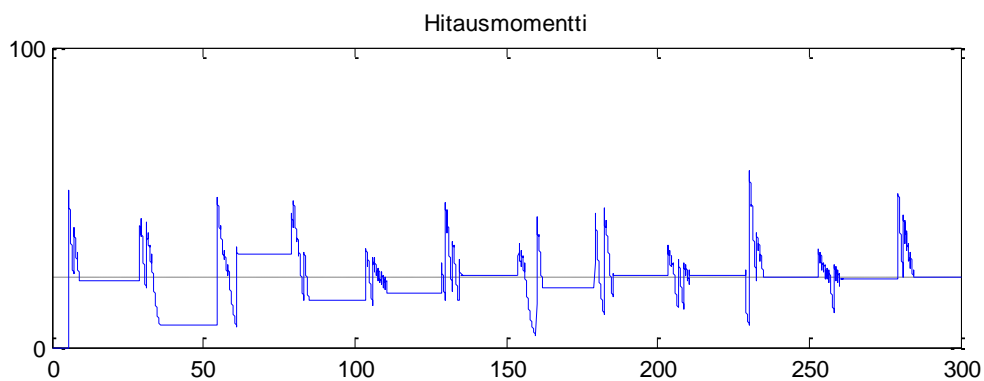
## 4.2 Testisimulaatio

Kehitettyä testijärjestelmää voi käyttää esimerkiksi taajuusmuuttajaan kytketyn induktiokoneen ja siihen asennetun kuorman hitausmomentin määrittämiseen. Testisimulaatiossa koneesta käytetään mallia, joka laskee vääntömomentin avulla moottorin kiihtyvyyden. Laskettua kiihtyvyyttä verrataan taajuusmuuttajalla mitattuun, todelliseen pyörimisnopeuteen perustuvaan kiihtyvyyteen ja siten estimoidaan kuorman hitausmomentti. Yksinkertaistettu simulaatiomalli on esitetty kuvassa 4.3.



Kuva 4.3 Induktiokoneen ja siihen asennetun kuorman hitausmomentin laskentaan käytettävä simulaatiomalli.

Vakiovääntömomentti aiheuttaa vakiokiihtyvyyden, mutta taajuusmuuttajalle asetettujen nopeusrajoitusten vuoksi suurilla pyörimisnopeuksilla moottorin kiihtyvyys vaimenee nolnaan. Tämän vuoksi hitausmomentin estimaatin laskenta pyritään rajaamaan nopeusalueelle, jolla pyörimisnopeuden muutos on lineaarista. Lisäksi kuvassa 4.2 havaittavia, tiedonsiirtovirheistä johtuvia häiriöitä pyörimisnopeudessa suodatetaan pois. Simulaation avulla laskettu hitausmomentti on esitetty kuvassa 4.4.



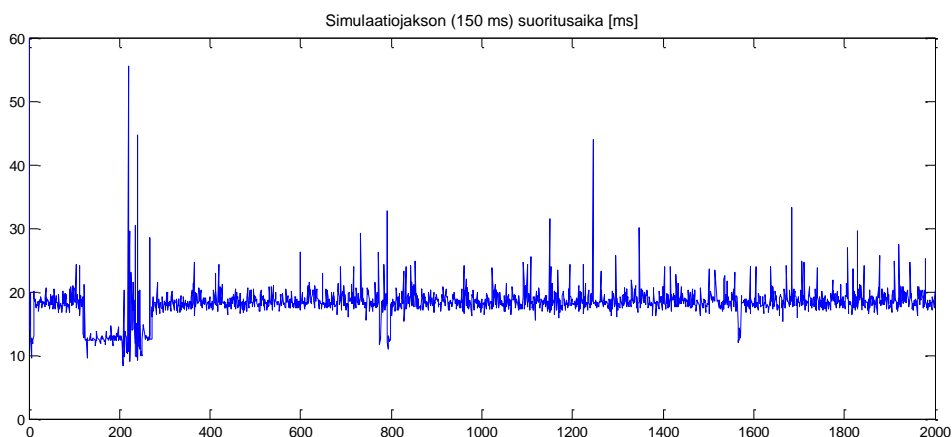
Kuva 4.4 Simulaation perusteella laskettu kuorman hitausmomentti.

### 4.3 Ongelmat

Vaikka kehitetyllä testausjärjestelmällä pystyy suorittamaan onnistuneita simulaatioita, se ei silti ole kovin varmatoiminen. Lukuisissa testiajoissa kone ei lähtenyt pyörimään ollenkaan

tai pyörimisnopeudeksi luettiin pelkkää nollaa, vaikka kone pyöri. Ongelmat johtuivat virheistä joko kenttäväylällä tai kenttäväylän ja Simulink:n välillä. Modbus-sovelluksen ja Simulink:n taajuusmuuttajalohkon välisessä tietoliikenteessä tapahtuu paljon virheitä, joten protokollassa ja koko yhteysmuodossa olisi parannettavaa.

Eräs ongelma on myös simulaation suoritusnopeuden epädeterministisyys. Reaaliaikaisuuden kannalta jaksotettavassa simulaatiossa olisi ideaalia, jos jokaisen simuloitavan jakson laskenta kestäisi reaaliajassa yhtä kauan kuin on jakson pituus. Tällöin simulaatio säilyisi jatkuvana. Simulointinopeus riippuu kuitenkin muun muassa simuloitavan mallin monimutkaisuudesta sekä simulaattorin laskentatehosta. Kuvan 4.3 testisimulaatio on melko yksinkertainen, joten simulaattorina käytetty tietokone ehtii suorittamaan jaksot selvästi riittävän nopeasti, kuvan 4.5 mukaisesti. Keskimääräinen jakson suoritus-aika on alle 20 ms, joten 150 ms jaksoilla suoritustehosta käytetään vain noin 13 %. Erilaisilla simulaatiomalleilla testatessa suoritusajat olivat kuitenkin huomattavasti pidempiäkin.



Kuva 4.5 Simulaation jaksojen laskentaan kulunut aika. Keskimääräinen suoritus-aika on hieman alle 20 ms.

Suoritustehoon voidaan vaikuttaa muuttamalla simulaatioaskelien pituutta. Testisimulaatiossa käytetään 10 ms askelia, joten monimutkaisemmalla mallilla askelien pituutta saatetaan joutua kasvattamaan eikä järjestelmä enää sovellu esimerkiksi nopeiden säätöjen simulointiin. Tällöin taajuusmuuttajaa voidaan hyödyntää hajauttamalla nopeutta vaativat osat sen suoritettavaksi.

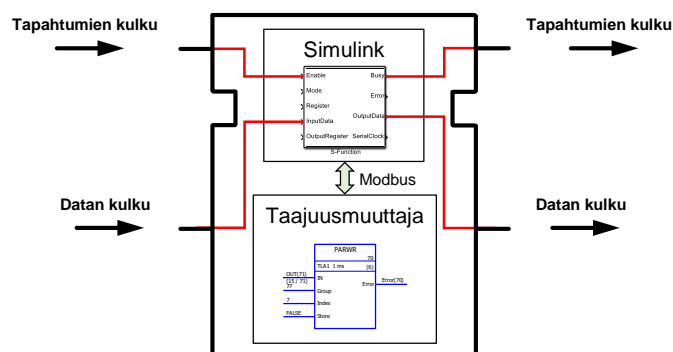


## 5. YHTEENVETO

Tavoitteiden mukaisesti työssä kytkettiin taajuusmuuttaja MATLAB- ja Simulink-ohjelmistoon muodostamalla rajapinta tarvittavaa tietoliikennettä varten. Tiedonsiirto tapahtuu Modbus RTU –kenttäväylän välityksellä ja siten taajuusmuuttajan rekistereitä pystytään käsittelemään suoraan Simulink:stä. Tämä mahdollistaa taajuusmuuttajan käyttämisen osana simulaatiota muodostaen HIL-järjestelmän. Simulaation synkronointi reaaliaikaiseksi tapahtuu jaksottamalla suoritusta. Taajuusmuuttaja välittää simulaatioon informaatiota käyttäen hyväksi omaa IEC 61131-3 –standardin mukaista ohjelmaa.

### 5.1 Työn tulokset

Kaikki kehitetyn järjestelmän osat toimivat riittävän hyvin toivotun toiminnallisuuden saavuttamiseksi. HIL-simulaation toteutus onnistui ja sen avulla myös taajuusmuuttajan ohjaaminen. Testisimulaatiossa taajuusmuuttajaa voidaan ohjata vääntömomenttiohjeella ja siltä saadaan vastauksena moottorin pyörimisnopeus, jonka avulla voidaan määrittää kuorman hitausmomentti. Simulink-mallia simuloidaan 10 ms aikavälein ja jaksotetaan 150 ms välein, jolloin esimerkiksi nopeiden säätöjen simulointi ei välttämättä ole mahdollista. Taajuusmuuttajan lohko-ohjelmaa sen sijaan suoritetaan 1 ms aikatasoilla, joten nopeutta vaativien osien hajauttaminen taajuusmuuttajalle mahdollistaa monipuolisempien simulaatioiden toteuttamisen. Kehitetystä järjestelmästä voidaan muodostaa kuvan 5.1 kaltainen, IEC 61499 –standardin mukainen lohkomalli, jossa sisäinen toiminta ja suorituksen hajautus on piilotettu ja toimintaa ohjataan tapahtumapohjaisesti.



Kuva 5.1 IEC 61499 –standardin mukainen lohkomalli, joka sisältää taajuusmuuttajan käytön Simulink-lohkon kautta.

## 5.2 Jatkokehitysmahdollisuudet

Järjestelmän sisäisessä tietoliikenteessä tapahtuu paljon virheitä, joten protokollaa tulisi jatkossa kehittää luotettavamman ja nopeamman toiminnan mahdollistamiseksi. Myös erilaisia, esimerkiksi Ethernet-pohjaisia, kenttäväyliä kannattaisi testata järjestelmässä ja tehdä nopeusmittauksia tiedonsiirron optimoimiseksi. IEC 61131-3 –standardin mukaisten ohjelmien tutkiminen ja testaaminen jäi työssä hieman suunniteltua pienemmälle osalle. Testeissä lohko-ohjelmia käytettiin vain yksinkertaisiin skaalausoperaatioihin tarkoituksena vain testata niiden toiminta ja käyttömahdollisuudet. Jatkossa olisi hyödyllistä rakentaa lohkoista laajempia ohjelmakokonaisuuksia, kuten esimerkiksi kokonainen säätöjärjestelmä, ja testata taajuusmuuttajan resursseja ja suoritusnopeutta. Lisäksi kannattaisi tutkia myös useiden erilaisten taajuusmuuttajien ominaisuuksia ja muun muassa valmiuksia kunnonvalvontaan ja etähallintaan. Työn tuloksia voisi laajentaa myös IEC 61499 –standardin osalta. Standardi sisältää XML (Extensible Markup Language) –kielen mukaisen kuvauksen lohkoista. Kuvan 5.1 lohkomallista voisi muodostaa XML-kielisen kuvauksen, jonka avulla voisi olla mahdollista luoda lohkoja suoraan Simulink-malleihin.

Simulink:in uusimpiin versioihin on myös tullut jo ominaisuuksia, jotka mahdollistavat reaaliaikaisen simulaation, HIL-simulaation sekä kenttäväylien käytön. Niitä hyödyntämällä järjestelmää saisi yksinkertaistettua ja tehostettua huomattavasti. Lisäksi monilla automaatiovalmistajilla on tarjolla Simulink:iin yhdistettäviä tuotteita, kuten Siemens:n SIMATIC-logiikat, Beckhoff:n TwinCAT (The Windows Control and Automation Technology) –ohjelmisto sekä dSPACE:n reaaliaiksimulaattori. Niiden avulla voisi tutkia tarkemmin HIL-simulaatiota ja laajempia hajautettuja järjestelmiä.

## LÄHTEET

- (ABB, 2007) ABB Oy. 2007. ACSM1-04 Drive Modules 0.75 to 45 kW - Hardware Manual. Verkkolähde: [https://library.e.abb.com/public/e77c7dbb873624cbc12572f8002dd548/EN\\_ACSM1\\_04\\_HWMan\\_C.pdf](https://library.e.abb.com/public/e77c7dbb873624cbc12572f8002dd548/EN_ACSM1_04_HWMan_C.pdf). [Viitattu: 31.5.2016]
- (ABB, 2011) ABB Oy. 2011. FSCA-01 RS-485 adapter module - User's manual. Verkkolähde: [https://library.e.abb.com/public/45986ebef18e3a5cc1257996002a484f/EN\\_FSCA\\_01\\_UM\\_A\\_screenres.pdf](https://library.e.abb.com/public/45986ebef18e3a5cc1257996002a484f/EN_FSCA_01_UM_A_screenres.pdf). [Viitattu: 31.5.2016]
- (ABB, 2015) ABB Oy. 2015. ACSM1 Speed and Torque Control –ohjelma – Ohjelmointiopas. Verkkolähde: [https://library.e.abb.com/public/8cd0b7b87fad49e2b5b03640fbbe7e35/FI\\_ACSM1\\_Speed\\_and\\_Torque\\_Control\\_FW\\_I\\_A4.pdf](https://library.e.abb.com/public/8cd0b7b87fad49e2b5b03640fbbe7e35/FI_ACSM1_Speed_and_Torque_Control_FW_I_A4.pdf). [Viitattu: 31.5.2016]
- (Baracos, 2001) Baracos Paul, Murere Guillaume, Rabbath Camille Alain, Jin Wensi. 2001. Enabling PC-Based HIL Simulation for Automotive Applications
- (CODESYS, 2016) CODESYS, 3S-Smart Software Solutions. 2016. Verkkolähde: <https://www.codesys.com>. [Viitattu 31.5.2016]
- (Halvorsen, 2016) Halvorsen Hans-Petter. 2016. Hardware-in-the-Loop Simulation. Verkkolähde: <http://home.hit.no/~hansha/documents/lab/Lab Work/HIL Simulation/Background/Introduction to HIL Simulation.pdf>. [Viitattu 31.5.2016]
- (IEC, 2005) IEC. 2005. IEC 61499-1:2005 – International Standard
- (John, 2010) John Karl-Heinz, Tiegelkamp Michael. 2010. IEC 61131-3: Programming Industrial Automation Systems. Springer-Verlag Berlin Heidelberg. ISBN 978-3-642-12014-5
- (Kleijn, 2016) Kleijn Christian. 2016. Introduction to Hardware-in-the-Loop Simulation. Verkkolähde: <http://www.hil-simulation.com/images/stories/Documents/Introduction to Hardware-in-the-Loop Simulation.pdf>. [Viitattu 31.5.2016]

- (Modbus.org, 2006) Modbus.org. 2006. MODBUS over Serial Line - Specification and Implementation Guide V1.02. Verkkolähde:  
[http://www.modbus.org/docs/Modbus\\_over\\_serial\\_line\\_V1\\_02.pdf](http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf).  
[Viitattu 31.5.2016]
- (Modicon, 1996) Modicon, Inc. 1996. Modicon Modbus Protocol Reference Guide. Verkkolähde:  
[http://web.eecs.umich.edu/~modbus/documents/PI\\_MBUS\\_300.pdf](http://web.eecs.umich.edu/~modbus/documents/PI_MBUS_300.pdf).  
[Viitattu 31.5.2016]
- (Ogan, 2015) Ogan Ron T. 2015. Hardware-in-the-Loop Simulation. Modeling and Simulation in the Systems Engineering Life Cycle. Springer-Verlag London. ISBN 978-1-4471-5633-8
- (Rooker, 2009) Rooker Martijn N., Ebenhofer Gerhard, Strasser Thomas. 2009. Re-configurable Control in Distributed Automation Systems.
- (Visedo, 2011) Visedo Oy. 2011. CoDeSys chosen as programming platform for Visedo PowerMASTER. Verkkolähde:  
<http://www.visedo.com/en/news/codesys-chosen-programming-platform-visedo-powermaster>. [Viitattu 31.5.2016]
- (Yoong, 2015) Yoong Li Hsien, Roop Partha S., Bhatti Zeeshan E., Kuo Matthew M.Y. 2015. Model-Driven Design Using IEC 61499. Springer International Publishing Switzerland. ISBN 978-3-319-10520-8