

Märt Ehrenpreis

Metso DNA-automaatiojärjestelmän simulointiympäristön luominen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Automaatiotekniikan koulutusohjelma

Insinööriytyö

18.4.2015

Tekijä(t) Otsikko Sivumäärä Aika	Märt Ehrenpreis Metso DNA- automaatiojärjestelmän simulointiympäristön luominen 67 sivua + 3 liitettä 18.4.2015
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Automaatiotekniikan koulutusohjelma
Suuntautumisvaihtoehto	Energia-automaatio
Ohjaaja(t)	lehtori Markku Inkinen
<p>Opinnäytetyön aiheena oli suunnitella Metropolia Ammattikorkeakoululle Metso DNA-automaatiojärjestelmää ja Matlab Simulink simulointiohjelmistoa yhdistävä simulointiympäristö. Suunnittelutehtävä jaettiin kahteen osaan, joista ensimmäisessä rakennetaan simulaattori, jossa prosessimalli sijaitsee ohjelmassa Matlab Simulink, mutta PID-säädin puolestaan Metso DNA-automaatiojärjestelmässä. Työn toinen osio sisältää prosessisimulaattorin, jossa monimutkaisempi säätöalgoritmi on sijoitettu Matlab Simulink-prosessimallin yhteyteen. Prosessisimulaattoreiden käyttöliittymät sijaitsevat Metso DNA-ympäristössä ja ovat käytettävissä valvomo-ohjelmiston DNAuse kautta. Prosessimallin ja Metso DNA-automaatiojärjestelmän välisessä reaaliaikaisessa tiedonsiirrossa käytettiin Metson OPC-palvelinta.</p> <p>Työn teoriaosassa perehdyttiin eri OPC-määritelmiin ja simulaattorin rakentamiseen sekä virittämiseen liittyviin kysymyksiin. Simulointikokonaisuus rakennettiin siten, että seuraavat kriteerit täyttyvät: simulaattorien käyttöliittymien kautta on tarvittaessa mahdollista muuttaa prosessiohjauksen säätöparametreja, prosessimallin muokkaus muuhun käyttöön on helpposti toteutettavissa, annettiin tarvittavat ohjeet simulointiympäristön käynnistykseen ja käyttöön liittyviin yksityiskohtiin.</p> <p>Simulointiympäristö suunniteltiin ja rakennettiin onnistuneesti. Prosessisimulaattoreilla suoritettiin askelkokeet ja simulaattorit todettiin käyttökelpoisiksi.</p>	
Avainsanat	Metso DNA, OPC, Matlab, Simulink

Author Title	Märt Ehrenpreis Creation of simulation environment for Metso DNA-automation system
Number of Pages Date	67 pages + 3 appendices 18 April 2015
Degree	Bachelor of Engineering
Degree Programme	Automation Technology
Specialisation option	Energy Automation
Instructor	Markku Inkinen, Title (Principal Lecturer)
<p>The main goal of this thesis is to create a simulation environment that links together Metso DNA automation system and Matlab Simulink engineering software. This task is being performed in two steps. At first, a simulator, which has a process model in Matlab Simulink software and PID-controller in Metso DNA system, is designed. After that the main attention is paid to the process simulator where a complex regulation algorithm is integrated into the process model created in Matlab Simulink. For both simulators have been designed a human-machine interface (HMI) in Metso DNA environment. Real-time communication between Matlab Simulink engineering software and Metso DNA automation system is managed via OPC server.</p> <p>The theoretical part of this thesis studies different OPC-specifications that are used in industrial automation, and the topics related to construction and tuning of process simulators. Simulation environment was created taking into account the following criteria: the simulator HMI-interface allows the operator to edit the process parameters during the simulation. If necessary, the process model created in Matlab Simulink can be easily customized for other simulation purposes. There are useful hints how to apply and control the simulation environment.</p> <p>The process simulators were tuned and step responses were carried out to prove that the simulators are designed and built successfully.</p>	
Keywords	Metso DNA, OPC, Matlab, Simulink

Sisällys

Lyhenteet

1	Johdanto	1
1.1	Työn tavoitteet	2
1.2	Työn rajaukset	2
2	OPC	3
2.1	OPC:n historia ja tulevaisuus	3
2.2	OPC:n määrittelyt	5
2.2.1	OPC Data Access	6
2.2.2	OPC Alarm & Events	8
2.2.3	OPC Historical Data Access	9
2.2.4	OPC Batch	10
2.2.5	OPC XML-DA	10
2.2.6	OPC Data eXchange (OPC DX)	10
2.2.7	OPC Command Execution	11
2.2.8	OPC Complex Data	12
2.2.9	OPC Security	13
2.2.10	OPC Unified Architecture (OPC UA)	14
3	Metso DNA-automaatiojärjestelmä	17
3.1	Työn kohteena olevan Metso DNA-automaatiojärjestelmän yleiskuvaus	17
3.1.1	Metso DNA Explorer	18
3.1.2	Function Block CAD	19
3.1.3	DNAuseEditor Picture Designer	20
3.1.4	Metso DNA Operate ja DNA Operate TEA	21
3.1.5	Trendinäytön suunnittelu Metso DNA-järjestelmässä	22
3.1.6	Metso DNA:n OPC yhteys	23
4	Matlab Simulink	25
4.1	Matlab OPC Toolbox	25
4.2	Signal Builder	28
5	Simuloinnit	29
5.1	Simulaattori 1 (PID-säädin Metso DNA-järjestelmässä)	31
5.1.1	Säädön automaattinen viritys (simulaattori 1)	41

5.1.2	Askelvastekoe 1	47
5.1.3	Askelvastekoe 2	51
5.2	Simulaattori 2 (PID -säädin Matlab Simulink-- mallin yhteydessä)	52
5.2.1	Säädön automaattinen viritys (simulaattori 2)	58
5.2.2	Askelvastekoe 3	61
5.2.3	Askelvastekoe 4	64
	Yhteenveto	65
	Lähteet	66
	Liitteet	
	Liite 1. DNA Engineering Loop Auto Tuner:in avulla lasketut säätimen viritysarvot	
	Liite 2. FbCAD graafinen ohjelma simulaattorien 1 ja 2 kytkennöistä testaus tilassa	
	Liite 3. Metso DNA automaatiojärjestelmän laiteluettelo	

Lyhenteet

ACN	Application and Control Node
COM	Component Object Model
DCOM	Distributed Component Object Model
DDE	Dynamic Data Exchange
EAS	Enterprise Application Server
ERP	Enterprise Resource Planning
HDAC	Historical Data Access Client
HDAS	Historical Data Access Server
HMI	Human-Machine Interface
HTTP	Hypertext Transfer Protocol
I/O	Input/Output
MES	Manufacturing Execution System
MIO	Metso Input Output
NAT	Network Address Translation
OLE	Object Linking and Embedding
OPC	OLE for Process Control
OPC A&E	OPC Alarms and Events
OPC DA	OPC Data Access

OPC DX	OPC Data eXchange
OPC HDA	OPC Historical Data Access
OPC UA	OPC Unified Architecture
OPC XML DA	OPC XML Data Access
PID	Proportional - Integral - Derivative
PLC	Programmable Logic Controller
SNMP	Simple Network Management Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
XML	Extensible Markup Language

1 Johdanto

Tietotekniikan kehityksen myötä automaatiojärjestelmät ovat muuttuneet varsin monimutkaisiksi. Toisaalta he ovat tulleet verraten kalliiksi, mutta toisaalta myös tuotantoprosessien tehokkuutta parantaviksi tuotannontekijöiksi. Erityisesti vientiyrietykset ovat kiinnostuneita ottamaan käyttöön uutta teknologiaa, koska niiden on kyettävä suunnittelemaan tuotantoketju tarkasti ja kustannustehokkaasti pysyäkseen kilpailukykyisinä globaalimarkkinoilla. Globalisointi painaa tuotteiden hintaa alas, joten vaikeassa kilpailutilanteessa toimivien yritysten talous ei kestäisi pitkällä aikavälillä tuotantoprosessissa syntyviin kustannuksiin tai tuotteiden laatuun liittyvää epävarmuutta.

Prosessien ohjaamiseen kehitettyjä tietoteknisiä ohjelmia tarjoavat useat valmistajat, mutta automaatiojärjestelmät eivät ole yksinkertaisesti kilpailemassa keskenään. Jokaisella valmistajalla on useimmiten oma ydinosaamisensa ja eri ratkaisuilla on omat vahvuutensa. Näin ollen automaatioalalla pidetään tärkeänä mahdollisuutta yhdistää ohjelmia eli kytkeä yhteen eri valmistajien automaatiojärjestelmien moduulit. Tästä järjestelystä hyötyy paitsi asiakas myös automaatiojärjestelmien valmistajat, joiden ei tarvitse toimittaa asiakkaalle koko tuotantolaitoksen automatisointia, vaan jokainen valmistaja voi keskittyä kehittämään omaa ydinosaamistaan ja vahvistamaan markkina-asemaansa sillä osa-alueella. Automaatiojärjestelmien räätälöinti eri valmistajien komponenteista onnistuu avoimen OPC-standardin ansiosta.

Tämä insinööriö käsittelee mahdollisuuksia kytkeä yhteen Metropolia Ammattikorkeakoulun laboratoriossa oleva Metso DNA-automaatiojärjestelmä ja Matlab simulointiohjelmisto. Työssä perehdytään OPC-standardin luomiin edellytyksiin yhdistää eri valmistajien ohjelmia ja tarkastellaan niitä käytännössä kokeiden avulla.

1.1 Työn tavoitteet

Työn päätavoitteena on kehittää simulointiympäristö, jossa on mahdollista Metso DNA-automaatiojärjestelmän välityksellä ohjata MATLAB Simulink-ohjelmassa luotua prosessimallia. Käyttöliittymä ja tarvittavat säätöpiirit toteutetaan Metso DNA-automaatiojärjestelmässä.

Tavoitteena on sen lisäksi tarkastella Metso DNA-automaatiojärjestelmän ja MATLAB-simulointiympäristön yhteensovittamista siinä tilanteessa, jos monimutkaisemmat säätöpiirit yritetään toteuttaa MATLAB-ohjelmassa.

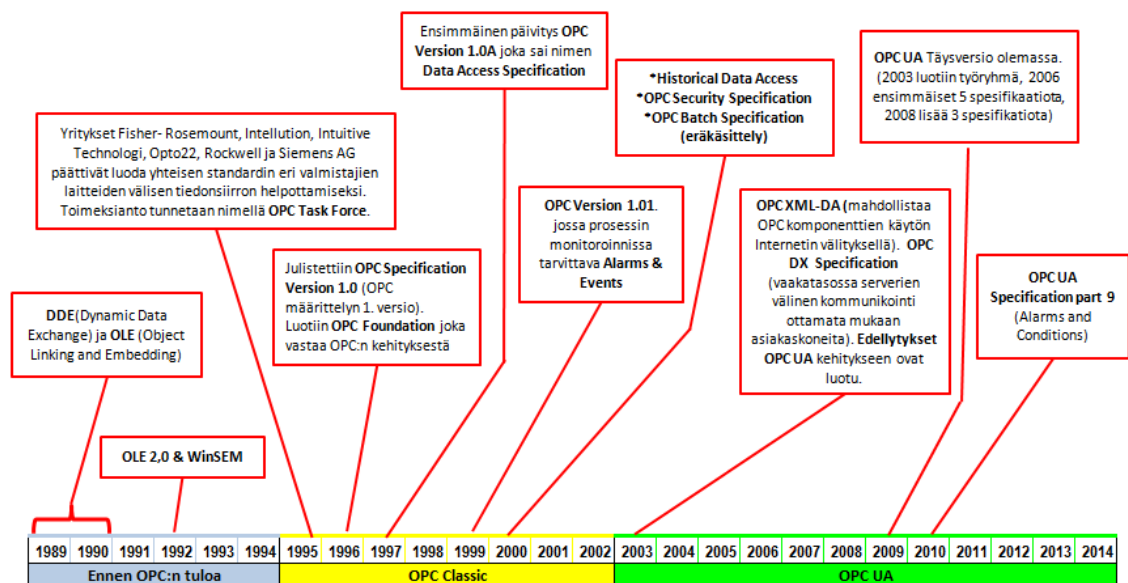
1.2 Työn rajaukset

Työn teoriaosuudessa tehdään suppea katsaus OPC:n eri spesifikaatioihin sekä simulointien osalta keskitytään vain niihin kysymyksiin, jotka liittyvät simulaattorin rakentamiseen ja simulointitulosten tulkintaan.

2 OPC

2.1 OPC:n historia ja tulevaisuus

OPC on tiedonsiirto-standardi, joka on luotu eri valmistajien automaatiokomponenttien saumattoman yhteensovittamisen helpottamiseksi. Kirjainyhdistelmä OPC muodostetaan sanoista OLE for Process Control. Lyhenne OLE eli Object Linking and Embedding viittaa Microsoftin COM tiedonsiirto- ja ohjelmistokomponentti tekniikoiden vanhaan nimeen. OPC:n läpimurto sai alkunsa vuonna 1995 (kuva 1) Microsoftin komponenttitekniikkaa hyödyntävän tiedonsiirto-standardin kehittämisestä automaatioalalle.



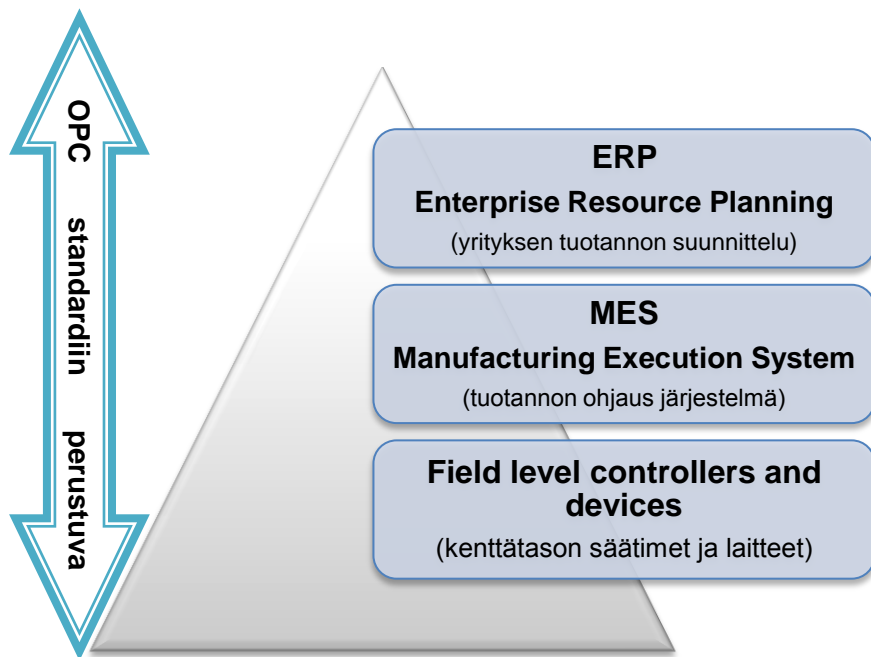
Kuva 1. OPC-standardin kehityshistoriassa merkittävimmät virstanpylväät. Värillä on ilmaistu uusien tekniikoiden käyttöönotto- ja kehitysjaksot. Käytännössä uusien toteutuksien rinnalla vanhat ratkaisut ovat myös käytössä. [1, s. 4 -7] [2]

OPC-standardin kehityksestä ja ylläpidosta vastaa voittoa tavoittelematon organisaatio nimeltään OPC Foundation. Järjestön jäsenenä ovat kaikki suuremmat automaatioinstrumenttien ja -järjestelmien valmistajat ympäri maailman.

OPC Foundation-järjestön päätehtäviin kuuluu:

- OPC spesifikaatioiden luonti ja ylläpito
- OPC spesifikaatioiden käyttökelpoisuuden varmistaminen testauksien avulla
- Yhteistyön tekeminen teollisuuden muiden standardisointiorganisaatioiden kanssa sekä OPC spesifikaatioiden käyttäjien kanssa.

Tätä nykyä OPC-standardin hyödyntäminen ei enää edellytä Windows alustaa järjestelmän toimimisen ehtona, vaan puhutaan alustasta riippumattomasta määrittelystä. OPC:n alkuaikojen Microsoftin COM/DCOM-tekniikka on osoittautunut nykyolosuhteissa osittain vanhentuneeksi, sillä verkottuminen ja Internet-tekniikat ovat vaikuttaneet automaation tietotekniseen ympäristöön. Uusia vaatimuksia järjestelmille on tullut pääasiassa luotettavuuden ja tietoturvan osalta.



Kuva 2. OPC standardiin perustuva tuotannon suunnittelutason ja kenttätason välinen tiedonsiirto. Samaa standardia hyödyntäen tapahtuu myös tiedonsiirto ns. vaakatasossa jolloin eri valmistajien automaatiokomponentit liitetään yhdeksi automaatiojärjestelmäksi. [1, s. 8]

Nykyisin OPC-standardit ovat automaatiosovellusten välisessä tiedonsiirrossa de-facto asemassa ja esimerkiksi mahdollistavat kaksisuuntaisen tiedonvälityksen yrityksen tuotannon suunnitteluportaalta kenttälaitetasolle asti (kuva 2). Kenttätason prosessitie-

dot voidaan OPC-määrittelyjen avulla koota Excel-taulukkoihin tai arkistoida tietokantoihin myöhempää analysointia varten.

Eri valmistajien tietoteknisten komponenttien yhteensovittamisen tarve esiintyy nykyään kasvavassa määrin yhä useammalla alalla, joten OPC-määrittelyt ovat löytäneet käyttöä teollisuusautomaation lisäksi myös rakennusautomaatiossa. Esimerkkinä olkoon mainittu palohälytys- ja murtohälytysjärjestelmät sekä älykkäät taloautomaatiojärjestelmät. [2]

2.2 OPC:n määrittelyt

Eniten tunnetut OPC-määrittelyt ovat DA (Data Access), A&E (Alarms and Events) ja HDA (Historical Data Access).

- **OPC DA** (määrittelee rajapinnan prosessidatan reaaliaikaiseen lukemiseen ja kirjoittamiseen)
- **OPC A&E** (määrittelee rajapinnan hälytyksien ja tapahtumatietojen monitorointiin)
- **OPC HDA** (määrittelee rajapinnan historiatietojen käsittelyyn)

Edellä mainittujen spesifikaatioiden lisäksi järjestö OPC Foundation on kehittänyt myös seuraavanlaisia määrittelyjä:

- **OPC Batch** (määrittelee rajapinnan eräluontoisen tuotannon tiedonsiirtotarpeisiin)
- **OPC XML-DA** (HTTP-protokollaa ja XML-kieltä tukevia ympäristöjä eli Web sovelluksia varten kehitetty OPC määrittely)
- **OPC DX** (OPC Data eXchange on tarkoitettu OPC-palvelimien väliseen viestintään)

- **OPC Command Execution** (OPC-palvelimen tai kenttälaitteen käytönaikainen valvonta/ohjaus käskyillä: aloita, jatka, lopeta)
- **OPC Complex Data** (määrittelee kompleksisen datan kuvailuun ja käyttöön tarvittavat käytännöt)
- **OPC Security** (määrittelee OPC-käyttöön järjestelmille tietoturvasot)
- **OPC UA** (OPC Unified Architecture)

2.2.1 OPC Data Access

OPC-määritelmä *Data Access Specification* on vanhin kaikista OPC määritelmistä. Sen avulla on luotu standardoitu tapa palvelimen (*Data Access Server*) ja asiakaskoneen tai -ohjelman (*Data Access Client*) väliseen viestintään.

OPC Data Access Server voi olla esimerkiksi yksinkertainen ohjelma, mikä mahdollistaa pääsyn PLC:n rekistereihin sarjaportin välityksellä. Monimutkaisemmassa tapauksessa kyseinen palvelin mahdollistaa pääsyn erilaisissa laitteissa sijaitseviin useisiin muuttujiin. Yksi OPC asiakas (*OPC Data Access Client*) voi olla kytketty useampaan eri valmistajan palvelimeen. *OPC Data Access Server* voi jossakin tapauksessa toimia myös isomman ohjelman osana (esim. *PC based Control*). Tällaisessa tapauksessa OPC -palvelin auttaa isompaa ohjelmakokonaisuutta tarvittavien tietojen välityksessä.

OPC Data Access Client voi olla yksinkertaisimmillaan esimerkiksi Excel-työkirja tai sitten monimutkaisemmassa tapauksessa vaikka MATLAB Simulink-simulaattori. *OPC Data Access Client* voi luoda *OPC Data Access* palvelimelle monenlaisia OPC objekteja. Objektihierarkiassa ylimmällä tasolla sijaitsee *OPCServer object*, seuraavalla tasolla on *OPCGroup object* ja sen alle kuuluvat *OPCItem objectit*. [3, s. 2] Ainoastaan *OPCServer* objekti ja *OPCGroup* objekti ovat niin sanotusti oikeat DCOM-objektit, eli niille on määritelty rajapinnat, metodit ja parametrit. *OPCItem* objekteilla ei ole omia rajapintoja, koska ohjelmissa kirjoitetaan ja luetaan useita eri arvoja samaan aikaan, eli tietojen välitys nopeutuu ilman ylimääräisiä rajapintoja. *OPC-Item*:ien luku tapahtuu *OPCGroup*:ien metodien avulla. [1, s. 36]

Tietojen välitys *Data Access Server*:in ja *Data Access Client*:in välissä voi hoitua monella eri tapaa. Tieto voi olla haettu joko palvelimelta (*Cache*) tai esimerkiksi kenttälaitteelta (*Device*). Riippuen siitä, mistä tieto haetaan tai lähetetään, käytetään erityyppisiä lukutapoja, kuten synkronoitu luku (*synchronous read*) tai a-synkronoitu luku (*asynchronous read*). Ensimmäinen tapa on käytössä, mikäli tieto löytyy heti valmiina vaikka palvelimelta ja jälkimmäinen tapa on tarpeen silloin, kun tietoa joudutaan hakemaan esimerkiksi kenttälaitteelta, jolloin esiintyy monenlaisia viiveitä. Samat säännöt tiedonvälityksessä pätevät myös tietojen kirjoitusvaiheessa. [1, s. 37]

Data Access Server mahdollistaa *Data Access Client*:in pääsyn eri tietolähteisiin kuten esimerkiksi lämpötilatunnistimiin (*datasource*) ja säätimiin (*data sink*). Tietolähteet ja säätimet voivat sijaita esimerkiksi I/O-korteilla, jotka on kytketty tietokoneeseen. Niiden sijainti voi olla myös joku laite, kuten säädin tai I/O-moduuli. Yhteys voi olla toteutettu sarjaportin kautta tai väylätekniikalla. *Data Access Client* voi olla yhtäaikaaisesti kytketty useampaan *Data Access Server*iin. [1, s. 34]

OPC-standardin mukaiseen tietopakettin rakenteeseen kuuluvat seuraavat komponentit:

- Prosessitiedon arvo (*Process Data Value*)
- Prosessitiedon aikaleima (*Time Stamp of the Value*)
- Mittajärjestelmän status (*Status of the Measuring System*)

Jotta prosessitieto saadaan kulkemaan OPC-asiakkailta palvelimelle ja palvelimelta asiakkaille, muunnetaan valmistajakohtainen tieto DCOM VARIANT tyyppiseksi. OPC asiakkaat ja palvelimet pitävät huolen siitä, että ohjelmiston rajapinnassa liikkuva tieto olisi nimenomaan OPC-standardin edellyttämässä muodossa.

Prosessin aikaleima määrittelee prosessitapahtumat aika-akselilla. Se koostuu kahdeksasta tavusta ja se kykenee ilmaisemaan ajan alkaen 01.01.1601. 100 [ns] välein.

Tilatiieto pitää sisällään laatutietoa kaksi tavua (*Quality*), statustietoa neljä tavua (*Status*) ja täydentävää diagnosointitietoa kaksi tavua (*Limites*). Laatutieto antaa signaalille arvion seuraavalla asteikolla: hyvä (*Good*), huono (*Bad*) tai epätarkka (*Uncertain*).

Palvelimen nimiavaruus (*server namespace*) selvitetään seuraavasti: *Data Access Client* lähettää kyselyn (*query*) OPC-palvelimen ylimmälle objektille, eli *OPCServer objectille* saadakseen selville OPC-palvelimen nimiavaruuden rakenteesta. [1, s. 38]

2.2.2 OPC Alarm & Events

Spesifikaatio OPC Alarm & Events (jatkossa A&E) koskee tapahtumien ja hälytyksien käsittelyä (tunnistaminen, jäsentely, seuranta, lähettäminen). A&E-palvelin kykenee tallentamaan ennalta määriteltyä tapahtumaa koskevia tietoja eri lähteistä (säätimet, anturit, I/O-kortit).

Jos verrataan DA-palvelinta A&E-palvelimeen, niin edellinen välittää prosessitietoa asiakaskoneille, mutta jälkimmäinen lähettää tietoa prosessin tapahtumista ja hälytyksistä. Usein DA-palvelimen asetuksissa määritetään näytteenottoherkkyys välitettävän prosessitiedon määrän rajoittamisen mielessä. Tieto prosessisuureesta lähetetään asiakaskoneelle vasta silloin, kun suureen muutos täyttää ennalta annetut kriteerit (esim. muutos on suurempi kuin +/- 1 yksikkö). Mahdollisuus tämäntyyppiseen skaalaukseen on ainoastaan analogia-tyyppisen tiedon kohdalla. A&E-palvelin ei lähetä asiakaskoneilleen näytteitä prosessisuureista, mutta huomauttaa prosessissa tapahtuvista muutoksista. Konfigurointivaiheessa asetetaan tapahtumille ja hälytyksille tarvittavat raja-arvot, joiden vallittaessa aktivoidaan vastaava tapahtuma tai hälytys. A&E-palvelin voidaan konfiguroida reagoimaan kaikentyyppiseen tietoon (analogia-, binääritieto jne.). Tapahtumia voidaan määrittellä myös linkitetyn tiedon perusteella eli esimerkiksi tiettyinä kellonaikana tai tietyssä ajassa lämpötila saavuttaa tietyn arvon.

A&E-palvelin saattaa olla itsenäinen palvelin joka hankkii tiedon suoraan prosessista tai DA-palvelimelta. Mahdollista on myös, että A&E-palvelin on rakennettu yhteen DA-palvelimen kanssa.

A&E-asiakas voi olla osana ohjaus- tallennus- tai hallinnointijärjestelmää. Yksinkertaisimmillaan kyseessä voi olla Excel taulukko, mutta monimutkaisemmassa tapauksessa jopa prosessiohjausjärjestelmä. [1, s. 65 - 66]

2.2.3 OPC Historical Data Access

Historical Data Access Server (HDAS) mahdollistaa asiakkaalle pääsyn prosessien historiatietoihin. Standardissa tunnetaan kahdentyyppisiä HDA-palvelimia.

- Yksinkertainen datan trendien serveri, joka tarjoaa hieman enemmän kuin raaka-datan (aikaleima, datan arvo, datan laatu) varastoinnin.
- Kehittyneempi HDA serveri, jossa edellä mainitun serverin toiminnallisuuden lisäksi tapahtuu myös datan jalostusta ja pakkausta. Esimerkiksi lasketaan datasta keskiarvoja, minimi ja maksimi arvoja. Serverillä tapahtuu myös uuden datan luontia, huomautuksien lisäämistä ja datan muutoksien lukua. [5, s. 2]

HDA-palvelimen nimiavaruus sisältää kaiken historiatiedon, johon palvelin tarjoaa pääsyn. Spesifikaatiossa edellytetään vaan hierarkkisen nimiavaruuden olemassaoloa. Semantiikka on verrattavissa OPC DA-palvelimen semantiikkaan.

Historical Data Access Client (HDAC) voi muodostaa HDA-palvelimelle erilaisia OPC objekteja, joista ylimmällä tasolla sijaitsee *OPCHDAServer object*. Mainittu objekti tarjoaa mahdollisuuden serverin datan lukuun ja kirjoitukseen tai datan ja sen attribuuttien¹ muuttamiseen.

OPCBrowser object tarjoaa mahdollisuuden HDA-palvelimen nimiavaruuden selailuun. HDA-palvelimella ei ole objekteja jotka olisivat verrattavissa OPC DA serverin *OPC-Group* ja *OPCItem* objekteihin. HDAC määrittelee suoraan osoitteet tietoalkioille (*data item*) niiden käsittelytavan mukaan, ilman että tarvitsisi luoda erikseen objekti serverille. [1, s. 57 - 64]

¹ Data attribuutti on yksittäiseen data-alkion kuvaamisessa käytettävä täydentävä ominaisuus. Esim. data-alkion arvo (Item Value) voi pitää sisällään attribuutteja datatyyppiä "Data Type"

Taulukko 1. HDA-spesifikaatio määrittää HDAS ja HDAC välisessä tietoliikenteessä seuraaventyypiset toiminnallisuudet

Spesifikaation objekti	Toiminnallisuus
Read	asiakas lukee tietoja palvelimelta
Update	asiakas kirjoittaa tietoja palvelimelle
Annotation	asiakas lukee tai kirjoittaa huomautuksia palvelimella olevaan tietokantaan
Playback	toistaa tietoja ajassa kyselyn perusteella tiedystä ajanhetkestä tiettyyn ajahetkeen saakka

2.2.4 OPC Batch

OPC Batch-spesifikaatio on kehitetty helpottamaan eri valmistajien automaatiokomponenttien käyttöä eräluontoisessa tuotannossa. Kyseessä on sellainen tuotantotapa, jossa esimerkiksi sama tuotantolinja valmistaa erissä erilaista tavaraa. Erätuotanto on käytössä esimerkiksi leipomoissa, lääketeollisuudessa, musteiden ja maalien valmistuksessa ja muualla samankaltaisen tuotannon ohjauksessa. *OPC Batch*-spesifikaatiossa otetaan huomioon eräluontoista tuotantoa koskevan standardin (IEC 61512 ja ANSI/ISA S88.01 1995) suositukset. Kaikki prosessialueen Item:it kirjoitetaan etuliitteellä OPCB, joka on varattu *OPC Batch*-käyttöön, eli sellaista kirjainyhdistelmää ei saa käyttää muussa tapauksessa OPC:n käytön yhteydessä. *OPC Batch*-standardissa esitetyn määrittelyn mukainen hierarkia on suositus eikä ainoa mahdollinen hyväksytty tapa muodostaa *OPC Batch*-nimiavaruutta. [6 s.14]

2.2.5 OPC XML-DA

OPC XML-DA:n määrittelyn tarkoituksena oli integroida Microsoft:in XML toimintoja OPC standardiin. Myöhemmässä kehitystyössä panostettiin muun muassa käyttöjärjestelmästä ja sijainnista riippumattomuuteen. OPC XML-DA soveltuu käytettäväksi ympäristöissä, jotka tukevat HTTP-protokollaa ja XML -kieltä. [1,s.52]

2.2.6 OPC Data eXchange (OPC DX)

OPC Foundation-järjestö on luonut määritelmät ja rajapinnat *OPC Data Access Serverille*, *Alarm and Event Serverille*, *Batch Serverille*, and *Historical Data Access Serverille*. Nämä palvelimet välittävät tietoja tehtaan alimmalta tasolta (Plant level) pysyvuoraan (kuva 2) ylimmälle tietojärjestelmätasolle (*Enterprise level ERP*). OPC DX

määritelmä on kehitetty tietojen siirtoon vaakatasossa, eli eri OPC-palvelimien väliseen tietoliikenteeseen. OPC DX mahdollistaa OPC -pohjaisten järjestelmien yhdessä toimimisen (sisältäen DCOM:iin ja XLM:iin pohjautuvat järjestelmät Ethernet ympäristössä) sisältäen PLC:n HMI/SCADA:n, toimilaitteet ja työasemat. [4, s.9]

Esimerkiksi tilanteessa, jossa *OPC Data Access Serverien* tietosisältö on siirrettävä toiselta tuotantoalueelta toiseen, tarvitaan tehokasta datasiirtoratkaisua. OPC DX-palvelimien avulla data välitetään suoraan eteenpäin ilman standardoimattomia ohjelmistorajapintoja sekä puskurointia eri välitysohjelmiin.

DX-palvelin voi hakea tietoja yhdeltä tai useammalta DA-palvelimelta sekä yhdeltä tai useammalta DX-palvelimelta.

Ennen OPC DX-määritelmän tuloa tarvittiin mainittuun tiedonsiirtoon erillinen OPC-asiakasohjelma, jolla hoidettiin eri OPC-palvelimien välinen tietoliikenne. Monet valmistajat kehittivät omia välitysohjelmia, jotka eivät olleet varustettu standardoiduilla ohjelmistorajapinnoilla, joten ne eivät sellaisinaan kelvanneet sulautettujen järjestelmien käyttöön. [2,s.55]

2.2.7 OPC Command Execution

Edellä esitetyillä OPC-standardeilla toteutetaan pelkästään tietojen välitys. Monissa sovelluksissa pelkkä tietojen lukeminen ja kirjoittaminen eivät kuitenkaan ole riittävää, mutta tarvitaan myös käskyjen suorittamista.

OPC Command Execution (jatkossa *Command*) spesifikaatio pitää sisällään toiminnallisuuksia joilla voidaan vaikuttaa palvelimen tai asiakaskoneen tilaan. *Command* on määritelty erillisessä OPC-spesifikaatiossa sen yksinkertaisuutta ja toimintanopeutta silmällä pitäen. Mikäli *Command* määrittelyn ominaisuudet on integroitu tietojen lukemisesta ja kirjoittamisesta vastaavan OPC-spesifikaation yhteyteen, käskyjen välittäminen vie huomattavasti enemmän aikaa. Määrittelyllä toteutetaan automaatiojärjestelmän käytönaikaista valvonta eli pystytään vaikuttamaan järjestelmään käskyillä: tauko, jatka, lopeta (*pause, resume, cancel*). [9]

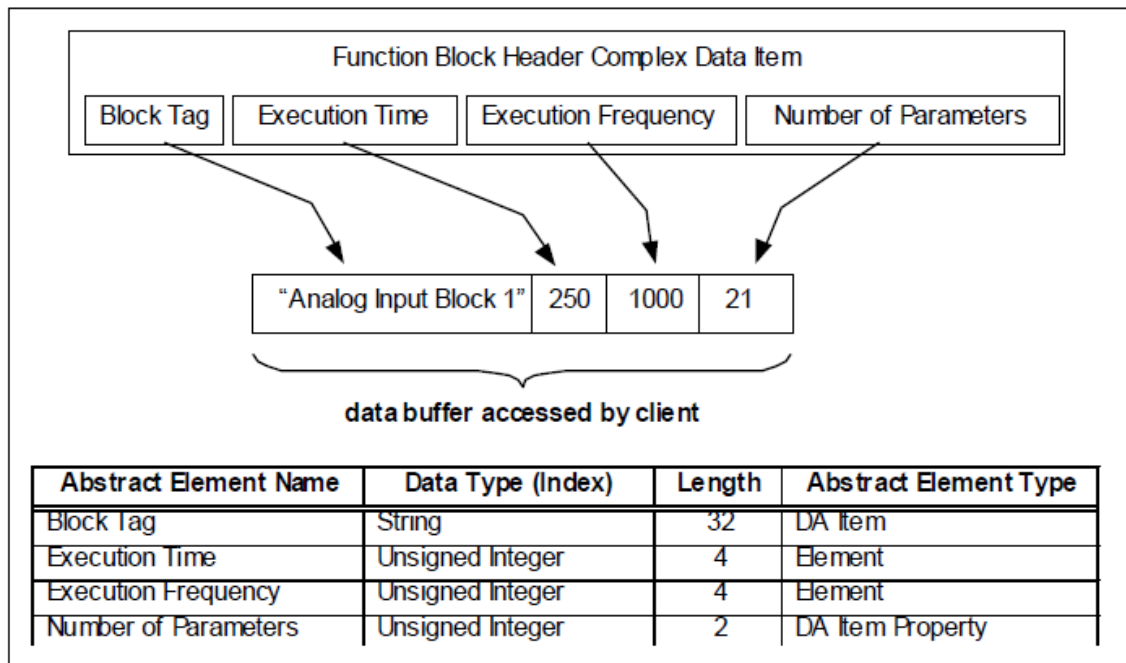
2.2.8 OPC Complex Data

OPC DA-palvelimen kautta voidaan päästää käsiksi yksinkertaiseen dataan (*simple data*), kuten lämpötilan tieto tai monimutkaisempaan tietoon (*constructed data*), kuten esimerkiksi diagnosointi tallenteet. *OPC Complex Data* määrittelee miten OPC DA:n rajapinnan kautta saadaan tehokkaasti rakenteellista tietoa (*constructed data*). Rakenteellisessa tiedossa on merkityksellinen myös yksittäisten dataelementtien välinen riippuvuus. Jos sellainen tieto vastaa *OPC Complex Data*-spesifikaation vaatimuksia, tunnetaan sitä termillä kompleksinen tieto (*Complex Data*).

OPC Complex Data määrittelyn hyödyntäminen voidaan jakaa tasoihin seuraavasti:

- OPC-asiakas ymmärtää kompleksisen tiedon rakenteen, ja pystyy arvioimaan millaista elementtiä tiedosta se hetkellä tarvitsee. Tiedon lukeminen ja kirjoittaminen hoidetaan sen elementteittäin ja tapauskohtaisesti.
- OPC-asiakas ymmärtää kompleksisen tiedon rakenteen lisäksi myös sen semantiikan (*semantics*), se kykenee tiedon elementtien erottelun lisäksi ymmärtämään niiden keskinäiset riippuvuudet.

Kuvassa 3 on näkyvässä kompleksisen datan *item:in* spesifikaation mukainen rakenne. *Block Tag* on erikseen tunnettu OPC DA-palvelimen tag:ina, mutta myös käytettävissä *OPC Complex Data*-rajapinnan kautta, jolloin se on osa isompaa rakenteellista kokonaisuutta (*data buffer accessed by client*). *OPC Complex Data*-määrittely mahdollistaa tietopisteen (*Block Tag*) ominaisuuksien räätälöidyn päivittämisen, eli kompleksisen tiedon toimilohkon elementtien valikoivan päivittämisen.



Kuva 3. Kompleksisen tiedon alkion (Complex Data Item) rakenne [10, s.13].

Vain pieni osa markkinalla olevista OPC-tuotteista tukee *OPC Complex Data*-spesifikaatiota ja syynä siihen on todennäköisesti se, että määrittely on melko uusi eikä ole vielä tullut tutuksi potentiaalisille käyttäjille sekä valmistajille. [1, s.78 - 80]

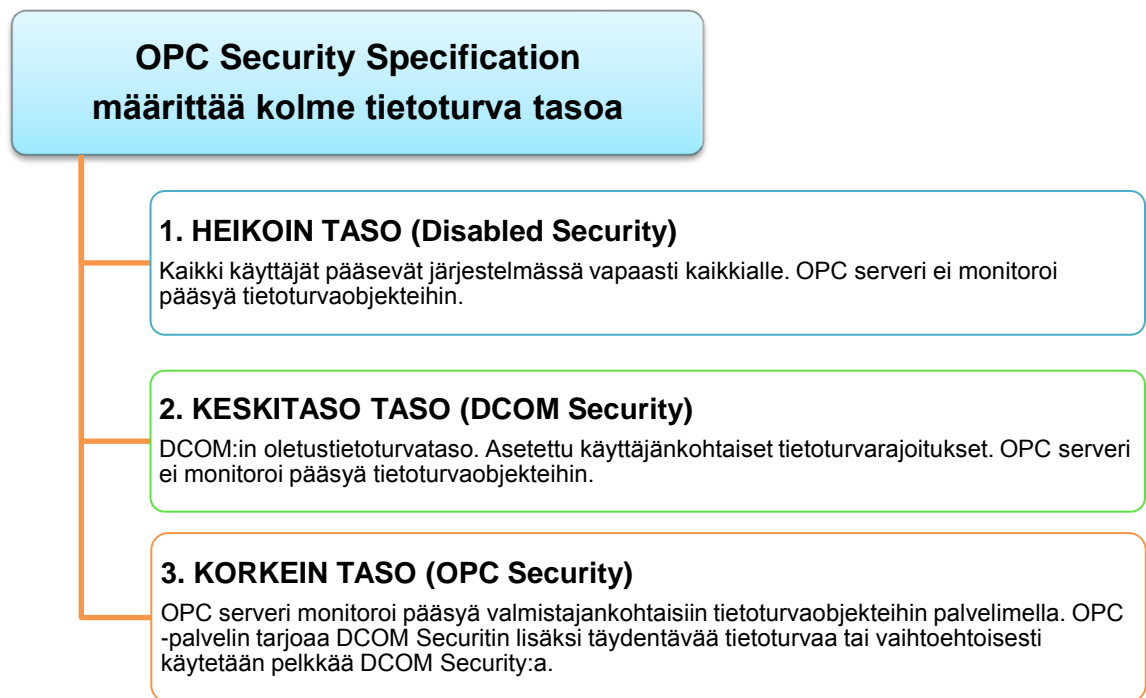
2.2.9 OPC Security

Tietoturvan tarkoitus on rajata ei-toivottua vierailua palvelimissa ja muissa verkkoon kuuluvissa laitteissa. Klassisen OPC:n kohdalla puhutaan DCOM-tietoturvasta (kuva 4) ja Web-palveluihin liittyvästä näkökulmasta tarkastellaan aihetta OPC UA:n käsittelyn yhteydessä. Tietoturva asetusten määrittämiseen on kaksi vaihtoehtoa:

Declaratiivinen tietoturva (*Declarative security*) jossa käytetään ohjelmaa DCOMCNFG. Haittapuolena tässä tapauksessa on se, että turva-asetuksia ei ole mahdollista määrittellä käyttäjäkohtaisiksi, eli muutokset vaikuttavat aina yleisesti kaikkiin käyttäjiin.

Ohjelmallinen tietoturva (*Programmatic security*) jossa käytetään *DCOM-Security-API* metodeita. Edellisestä poikkeavasti tässä tapauksessa voidaan määrittää erilaisia tietoturva-asetuksia eri käyttäjille. Mikäli tietoturvan määrittelemisessä ei noudateta *OPC*

Security Specification-suosituksia, voi myöhemmin esiintyä yhteensopivuusongelmia järjestelmän eri yksiköiden välissä.



Kuva 4. OPC Security Specifikaation määrittää DCOM:iin pohjautuville järjestelmille kolme mahdollista tietoturvasuoa. [1,s.83-86]

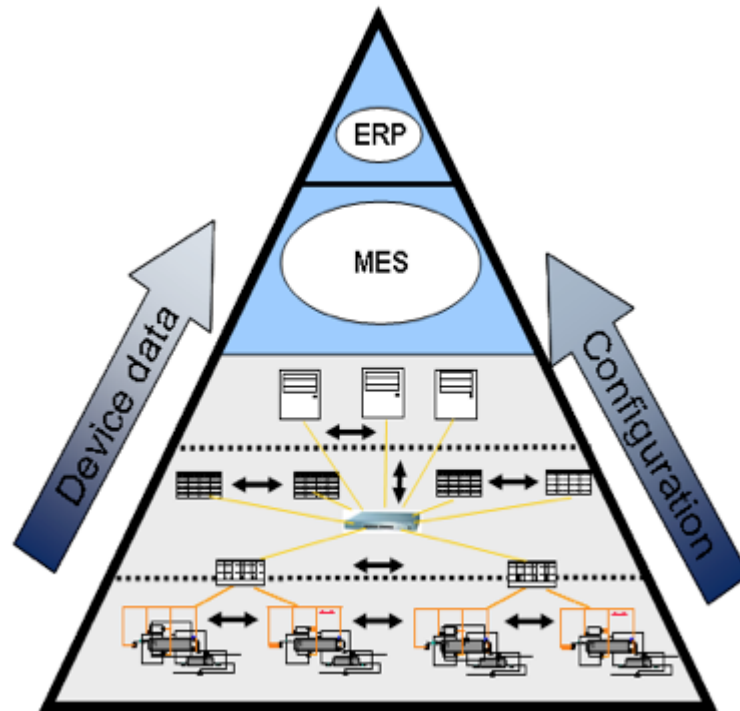
2.2.10 OPC Unified Architecture (OPC UA)

OPC UA:n kehittäminen sai alkuunsa vuonna 2003 (katso kuva 1) jolloin OPC Foundation järjestö julkaisi OPC XML-DA määrittelyn ja osoitti sillä, että OPC spesifikaatio voi olla riippumaton Windows-alustasta, sekä pystyy hyödyntämään Web Services-tekniikkaa. OPC XML-DA todettiin useammalla teollisuuden automatisointialueella liian hitaaksi ja siitä syystä käyttökelvottomaksi, mutta spesifikaatiolla oli merkitys OPC:n tulevaisuuden kannalta. Aikaisemmasta OPC Classic-määrittelystä luopuminen ja uuden spesifikaation kehittämisen tarve johtui useammasta syystä (kuva 6), mutta pääasiassa OPC Classic:in kohtaloksi koituivat DCOM-tekniikkaan ja tietoturvaan liittyvät ongelmat. [2] [11,s.16]

Viisi vuotta kestäneen OPC UA:n kehittämisen jälkeen OPC Classic-spesifikaatio korvattiin Web Service -tekniikkaan perustuvalla versiolla, joten yhteydet eivät rajoitu enää vain lähiverkkoihin, vaan ulottuvat Internetin kautta pitemmällekin. Samaan kehykseen

on koottu prosessidatan (DA), hälytyksien ja tapahtumien (A&E) sekä datahistorian (HDA) käsittelyn toiminnallisuudet. Nimiavaruudesta tuli OPC Classic-versioon verrattuna laajempi ja se esittää yleisellä tavalla kaikenlaiset mahdolliset riippuvuudet.

OPC Classic-korvaaja OPC UA kykenee valmistajasta, käyttöjärjestelmästä ja ohjelmointikielestä riippumattomaan tiedonsiirtoon mm. tehtaan kenttätasolta (*Field Level*) tuotannon suunnittelutasolle (ERP) asti (kuva 5).



Kuva 5. OPC UA mahdollistaa tehokkaan ja luotettavan tiedonsiirron tehtaan kenttätasolta tuotantosuunnittelutasolle asti. [12]

Yksittäiset UA-komponentit voivat olla jopa maantieteellisesti eri paikoissa ja erotettuna toisistaan palomureilla. Spesifikaatio on palomureiyttävällinen tukien mm. palomureissa tapahtuvia osoitemuunnoksia.

OPC UA:n tulo ei aiheuta automaatiojärjestelmien OPC Classic-laitteiden pakkoromuttamista ja suuria lisäinvestointeja. Se tukee aikaisempia spesifikaatioita ja on saumattomasti yhdistettävissä OPC Classic-mukaisiin järjestelmiin.

OPC Unified Architecturen luomiseen johtaneet syyt

1. Luopuminen COM/DCOM tekniikasta. Vuonna 2002 Microsoft lanseerasi uuden .NET ohjelmistokomponenttikirjastonsa ja ilmoitti samalla luopuvansa DCOM:in kehittämisestä. OPC Classic määritelmien käyttömahdollisuus perustuu juuri DCOM:iin.

2. DCOM -tekniikan aiheuttamat rajoitukset. DCOM tietoturvan konfigurointi on työlästä. Korkeampi tietoturvan taso saattaa hidastaa olennaisesti järjestelmien toimintaa. Mikäli tietoliikenteeseen tulee katkos esimerkiksi katkenneen yhteyden takia, menee useita sekunteja ennen kun OPC -asiaka saa siitä tiedon (long and not configurable timeouts).

3. OPC Classic - yhteys palomuurien ylitse. DCOM vaatii palomuurin useamman portin aukiolon välittämiseen tietoja palomuurin ulkopuolelle /-lta ja avoin portti on tietoturvariski. OPC Classic -yhteys intranetista ulospäin ei ole mahdollista, mikäli palomuuuri perustuu NAT -tekniikkaan (Network Address Translation).

4. OPC Classic -muussa kuin Windows ympäristössä. Perinteinen OPC vaatii Windows ympäristön toimiakseen. Esim. lääketieteellisyys ja kemiateollisuus välttää Windows pohjaisia käyttöjärjestelmiä niiden lyhytikäisyyden ja isojen versioerojen takia.

5. OPC Classic toimintatehokkuus Web Service:n kautta. OPC XML-DA mahdollistaa Windows -riippumattoman OPC yhteyden mutta tiedon välitykseen kuluva aika on sietämättömän pitkä monelle automatiojärjestelmälle. Viesti muunnetaan muodosta toiseen, pakataan ja avataan järjestelmien välisessä kommunikoinnissa, jolloin tiedon välitys on 5 - 7 kertaa hitaampi verrattuna OPC Classic sisäiseen tiedonsiirtonopeuteen.

6. Yhdistetyn tiedon malli. OPC Classic pitää sisällään DA-, AE-, HDA- määritelmät joista jokaisessa on oma nimiavaruusrakenne ja dataobjekteihin pääsyn tapa. Tällaisenaan järjestelmän käyttö on aikaa vievä joten tarvittaisiin tehokkaampi ratkaisu jossa päästäisiin helposti ja samanaikaisesti käsiksi DA-, AE-, HDA-tietoihin.

7. Kompleksisen datan tukeminen. Valtaosa nykyistä OPC Classic kalustoa ei tue kompleksisen datan käyttöä. On kyllä luotu Complex Data Specification, mutta sen mukaiset laitteet ovat markkinoilla vasta uusia. Tarvitaan ratkaisu joka nykytilanteessa tukisi kompleksisen datan rakennetta.

8. Kommunikointi ilman prosessitiedon menetyksiä. Tilanteessa jossa prosessitieto muuttuu nopeammin kun OPC Update Range mahdollistaa seurata, tai tiedon saannissa on katkos, menetetään joskus kriittistä prosessitietoa. Puuttuu OPC- ratkaisu saumattomaan ja tarkkaan prosessitiedon siirtoon. Nykyisin on käytössä korvaavia valmistajankohtaisia ratkaisuja ja odotellaan vastaavanlaisen OPC -määritelmän luontia.

9. Kasuvat tietoturvavaatimukset. Ilman valmistajankohtaisia lisätoimia OPC Classic ei täytä nykyaikaisia tietoturvavaatimuksia

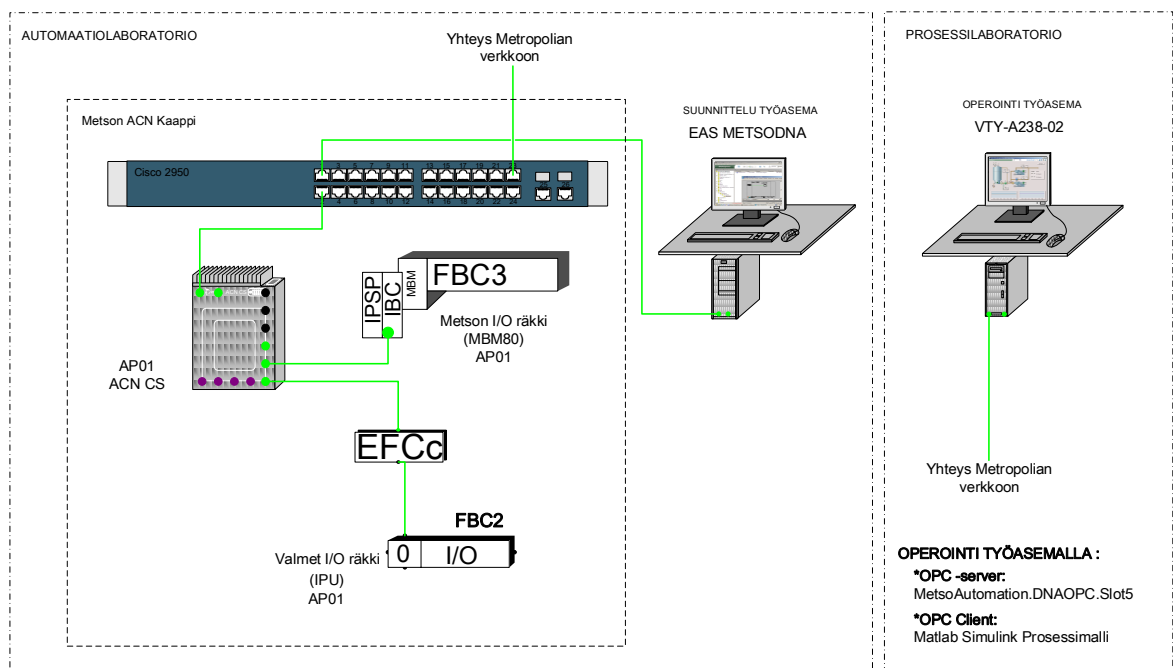
10. Metodien kutsujen tukeminen. OPC Classic sisältää määrittelyn OPC Comand Execution, mutta sen kehitys jäi osalta luonnoksen vaiheeseen. Teollisuus tarvitsee OPC:n kautta käskyjen välittämiseen täydellisemmän määrittelyn. Ominaisuuksien kehitystä jatkettiin OPC UA:n yhteydessä.

Kuva 6. Syyt jotka johtivat OPC Classic-spesifikaation kehittämisen lopettamiseen. OPC Foundation järjestö syrjäytti ongelman julkaisemalla vuonna 2008 OPC Unified Architecture määritelmän. [1, s. 86 - 92].

3 Metso DNA-automaatiojärjestelmä

3.1 Työn kohteena olevan Metso DNA-automaatiojärjestelmän yleiskuvaus

Työn kohteena oleva Metso DNA-automaatiojärjestelmä (kuva 7) sijaitsee Metropolia ammattikorkeakoulun Myyrmäen toimipisteessä. Järjestelmä koostuu Metso DNA-lähiverkosta johon kuuluu EAS-palvelin, operointitietokone, Metson ACN-kaappi jossa sijaitsee: Ciscon älykäs verkkokytin, EFC-muunnin, ACN CS solmu, Metson I/O-räkki ja Valmetin I/O-räkki (katso Liite 3 laiteluettelo). Järjestelmässä on kaksi työpistettä joista toinen (EAS METSODNA) on suunnittelijan työasema ja toinen (DNA Operate A102) on operointityöasema. Suunnittelu työasema sijaitsee automaatiolaboratoriossa ja siltä asemalta hoidetaan järjestelmän suunnittelutehtävien lisäksi myös ylläpitoon, kentälaitteiden hallintaan ja informaation hallintaan liittyviä toimia. Työpisteen käyttö on mahdollista myös koulun eri luokkatiloista, mikäli kirjaudutaan Virtuaali-ympäristöön (esim. luokka B249).



Kuva 7. Metso DNA-automaatiojärjestelmän järjestelmäkaavio

Operointityöasema sijaitsee prosessilaboratoriossa ja siltä asemalta suoritetaan operointiin, ylläpitoon tai raportointiin liittyviä tehtäviä. Operointi työasemalle on asennettu myös Metso DNA OPC palvelin (MetsoAutomation.DNAOPC.Slot5), jota tämän työn

puitteissa käytetään prosessitietojen välittäjänä Matlab Simulink-prosessimallin ja Metso DNA-prosessitietokoneen välissä.

3.1.1 Metso DNA Explorer

Automaatiosuunnittelu ohjelmisto Metso DNA Explorer on automaatioinsinöörille tehokas työkalupakki, tukien kuvan 8 mukaisia suunnittelutarpeita. Ensisijaisesti DNA Explorer on tarkoitettu erilaisien automaatiosovelluksien suunnitteluun ja ylläpitoon.

Function Block CAD	•toimilohkokaaviot
Sequence CAD	•sekvenssikaaviot
Picture Designer	•käyttöliittymät, prosessikuvat
FF Explorer	•FF-väyläkonfiguraatiot
PROFIBUS Configurator	•PROFIBUS-väyläkonfiguraatiot
Control Diagram CAD	•säätökaaviot
Field CAD	•kenttäkaaviot
Hardware CAD	•laitteistokaaviot
Loop Circuit CAD	•piirikaaviot
Motor CAD	•piiri- ja johdotuskaaviot
Logic CAD	•logiikkakaaviot
Text Editor	•automaatiokielliset konfigurointitoiminnot

Kuva 8. Metso DNA Explorerin suunnittelutyökalut kattavat kuvan mukaiset suunnittelutarpeet. [13]

Pähkinänkuoressa DNA Explorer toimii näin, että suunnittelija rakentaa kohteesta graafisen mallin suunnittelutietokantaan, jonka pohjalta ohjelma tuottaa vastaavan käyttöön sovelluksen. Ohjelma testataan suunnittelutietokannassa jolloin voidaan heti puuttua mahdollisiin suunnitteluvirheisiin. Ohjelmatestauksessa muodostuu sivutuotteena lokitiedosto, jossa raportoidaan ohjelmakomponenttien ajoympäristökelpoisuudesta eli hyväksymisestä tai hylkäämisestä. Lokitiedostossa saa myös vinkkejä mahdollisiin kor-

Työkalulla luodaan graafinen ohjelma eli toimilohkokaavio prosessin ohjauksiin ja säätöihin liittyvistä piireistä. Piirien toimintaedellytykset testataan FB CAD-suunnitteluympäristössä ennen ohjelman ajoympäristöön lataamista, jolloin voidaan korjata mahdolliset suunnitteluvirheet ajoissa. Säättöpiirin toimintaa voidaan tarkastaa myös ohjelman ajoympäristöön latauksen jälkeen Metso DNA Function test-työkalulla (kuva 9). Function test-työkalu antaa käyttäjän päättää, millaisia piirin pisteitä halutaan tarkastaa. Valituista pisteistä saadaan reaaliaikainen prosessitieto näkyviin joko numeroarvona tai trendikäyränä.

FB CAD-suunnittelutyökalu on tärkeä työväline myös automaatiojärjestelmän dokumentaation ylläpidossa, sillä piireistä luotujen graafisten ohjelmien avulla on helppoa pitää järjestelmien dokumentaatio ajan tasalla.

Työn puitteissa rakennettu FB CAD-ohjelmat ovat kytketty Metso DNA OPC palvelimen välityksellä Matlab Simulink-prosessimalliin. Function test-työkalusta oli merkittävä hyöty OPC-yhteyden luomisen aikana, jolloin seurattiin mitkä signaalit Matlab Simulink-prosessimallista päättyivät Metso DNA-ympäristöön. Myös PID-säätimen konfiguroinnissa oli hyödyllistä saada tietoa, signaalien arvoista säätimen liittimillä.

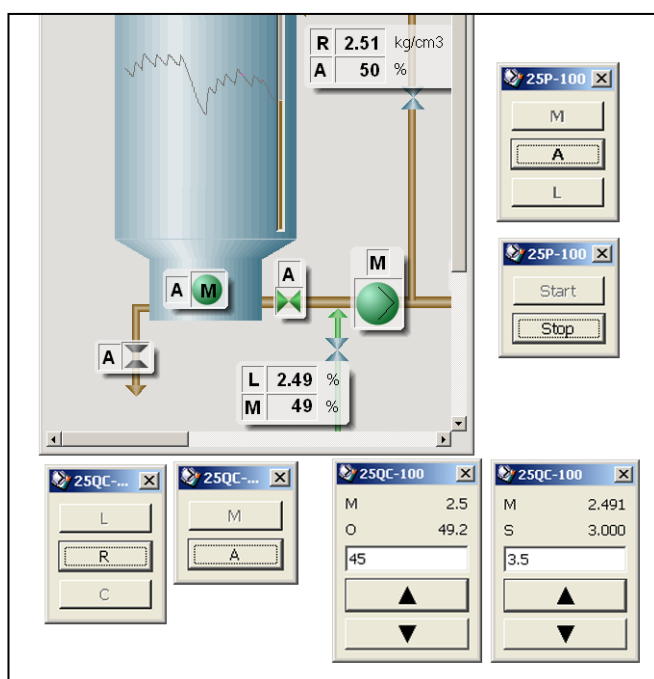
3.1.3 DNAuseEditor Picture Designer

Metso DNAuseEditor Picture Designer (jatkossa Picture Designer) on suunnittelutyökalu jolla luodaan prosessikuvia ja käyttöliittymiä. Picture Designer-ohjelmaan konfiguroidaan yhteydet muualle Metso DNA-automaatiojärjestelmään ja esimerkiksi OPC-palvelimen välityksellä mahdollisesti jopa pidemmällekin.

Picture Designer-ohjelmistolla suunniteltu käyttöliittymä ladataan Metso DNA-automaatiojärjestelmän ajoympäristöön jonka jälkeen suunnittelutuotos voidaan avata Metso DNA Operate-ohjelmassa laitoksen operaattorin käyttöön.

3.1.4 Metso DNA Operate ja DNA Operate TEA

Picture Designer ei toimi operaattorin työkaluna, vaan siihen on tarkoitettu erillinen ohjelmisto nimeltään Metso DNA Operate (kuva 10). Kyseessä on käyttäjätunnuksella ja salasanalla suojattu ohjelma, jonka avulla operaattori pääsee, riippuen käyttöoikeuksien laajuudesta, muokkaamaan tai selaamaan prosessin tietoja. Käyttöoikeudet on jaettu kolmeen tasoon: *display mode* (ei muokkaus-oikeutta), *control mode* (rajoitettu muokkaus-oikeus), *service mode* (ylläpitäjän oikeudet).



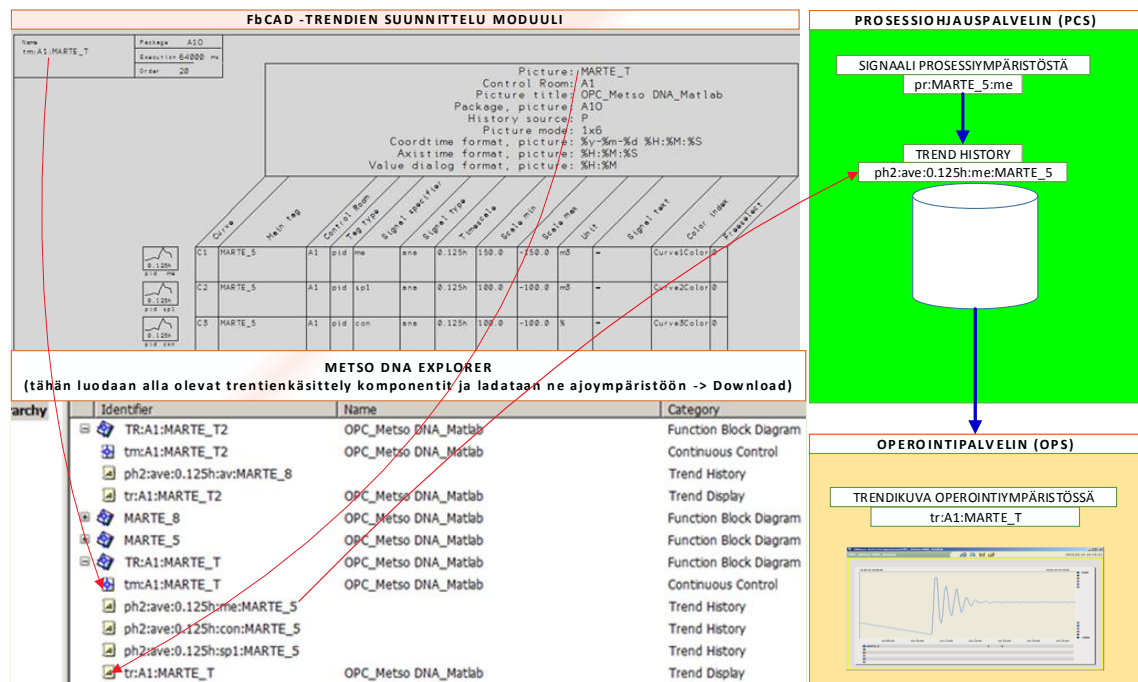
Kuva 10. Metso DNA Operate-näytön esimerkki [16 s.4]

Klikkaamalla prosessikuvan komponentteja pääsee tarkistamaan erilaisia prosessin alanäyttöjä ja saamaan tietoa kyseisen komponentin tilasta tai muista ominaisuuksista.

Operaattorin työkalu DNA Operaten lisäosio on tunnettu nimellä DNA Operate TEA (eng. *The Trend and Event Archive*). Tämä ohjelmisto on suunniteltu kuvaamaan trendien, tapahtumien ja hälytysten historiaa. Samassa ympäristössä onnistuu sekä prosessin historiatiedon analysointi että raportointi. Hälytystiedot voidaan esittää prosessin trendien ja tapahtumatietojen rinnalla samassa näytössä niin, että voidaan mahdollisesti tehdä johtopäätöksiä hälytyksiin johtaneista syistä prosessissa. [17 s. 1 - 4]

3.1.5 Trendinäytön suunnittelu Metso DNA-järjestelmässä

Alla olevassa kuvassa 11 on esitetty trendinäytön tietojen hankinnan periaate. Aluksi luodaan Function Block CAD-ohjelman avulla Metso DNA Exploreriin trendien seurannan moduuli. Moduuli löytyy valmiina mallipohjana DNA Explorerin mallipohjakirjastosta, mutta sinne täytyy konfiguroida meitä kiinnostavien signaalien tiedot. Kuvassa on konfiguroituna näkyvissä Metson PID-säätimen suorasaantiportin osoitteet me, sp1 ja con.



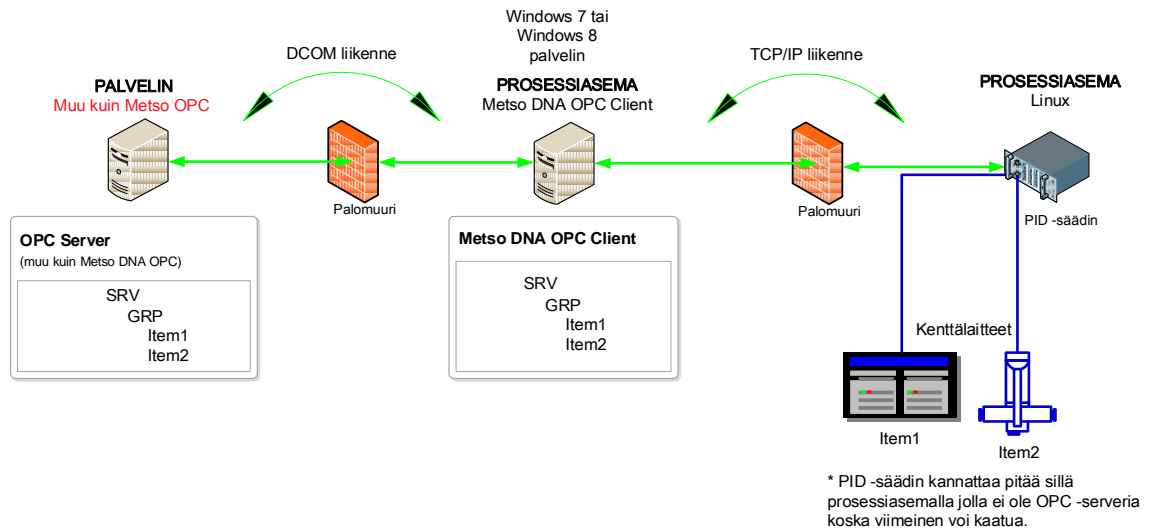
Kuva 11. Trendien suunnittelu Metso DNA-automaatiojärjestelmässä. FbCAD-trendien suunnittelumoduuli muodostaa konfiguroinnin aikana automaattisesti DNA Explorer:iin tarvittavat ohjelmisto-oliot, joten suunnittelijan tehtävänä on ladata ne ajoympäristöön [22 s. 1]

Sen jälkeen, kun seurattavien signaalien tiedot on syötetty ja ohjelmamoduulin toimintakelpoisuus testattu (*check*), voidaan se ladata Metso DNA-ajoympäristöön prosessiohjauspalvelimelle (PCS).

Valvomon Metso DNA Operate-ympäristössä luotu trendikuva saadaan näkyviin syöttämällä navigointikenttään trendikuvan kokonainen nimi (tr:A1:MARTE_T). Trendikuva tarjoaa mahdollisuuden piilottaa ja skaalata signaaleja tarpeen mukaan. Monien muiden ominaisuuksien ohella trendikuvissa voidaan esittää signaalien arvot tietyllä ajanhetkellä ja myös tarkastaa signaalien arvot menneisyydessä.

3.1.6 Metso DNA:n OPC yhteys

Työn alkuvaiheessa oli tarkoitus käyttää tiedon välitykseen MatrikonOPC² Simulation Server:ia tai OPC UA-palvelinta, mutta käytännön kokeet osoittivat, että muun kuin Metson valmistaman OPC-palvelimen käyttö ei ole nykytilanteessa mahdollista. Konsultaatio Metso Automaation systeemiasiantuntijan kanssa auttoi saamaan selville syyt jotka johtivat OPC-yhteyden epäonnistumiseen. Jotta Metso DNA-järjestelmä voitaisiin liittää ulkopuoliselle OPC-palvelimelle, tarvitaan rajapinnaksi täydentävä prosessiohjauspalvelin, joka voi tuetuissa Windows-käyttöjärjestelmissä toimia OPC DA- ja /tai OPC A&E-asiakkaana (Metso DNA OPC Client) [23].



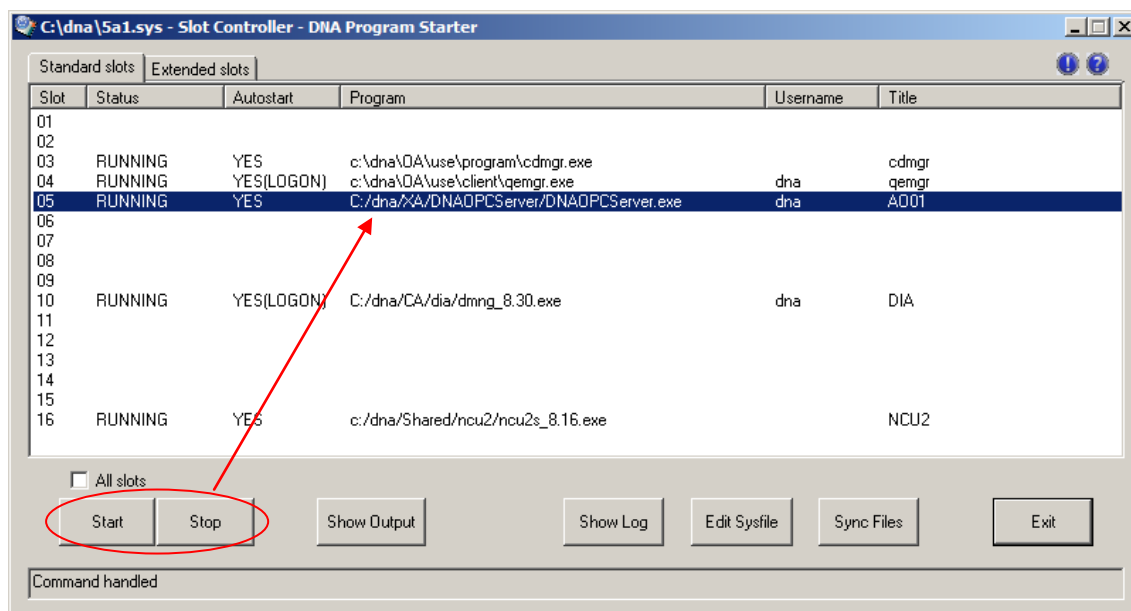
Kuva 12. Metropolian automaatiolaboratorioon olisi hankittava kaupallinen sovellus Metso DNA OPC Client, jos tarkoituksena on muodostaa yhteys muulta kuin Metson omalta OPC palvelimelta Metson Linux-käyttöiselle prosessiasemalle. [23]

Asiakasohjelma Metso DNA OPC Client tekee ulkopuolisesta kommunikaatioprotokollasta (DCOM, Web Service, tms.) Metso DNA-järjestelmälle käyttökelpoisen TCP/IP-liikenteen (kuva 12). Metson systeemiasiantuntijan Jussi Oikarin mukaan Metropolia Ammattikorkeakoulun käytössä olevan Metso DNA-automaatiojärjestelmän ohjelmistokokoonpanosta puuttuu edellä mainittu asiakasohjelma ja siitä syystä ei ole mahdollista käyttää FbCAD:in OPC-yhteyden muodostamisen toimilohkoja, jotka lisäävät selvyttä

² MatrikonOPC Simulation Server - ilmainen OPC-simulointiohjelmisto jolla voidaan simuloida OPC -asiakkaan yhteys OPC DA, OPC HDA tai OPC A&E palvelimeen testaustarkoituksessa. [24]

dokumentaatioon. Simulointiympäristössä OPC-yhteys palvelimen ja Metso DNA-automaatiojärjestelmän välissä saatiin toimimaan käyttäen FbCAD ohjelman toimilohkojen suorasaantiportteja.

Matlab Simulink-prosessimallin käynnistyksessä tapahtuu automaattisesti OPC-yhteyden muodostaminen Metso DNA-järjestelmään. Simulointiympäristön moitteettoman toimivuuden kannalta on tärkeää, että operointityöasemalle ja suunnittelutyöasemalle (kuva 7) olisi kirjaututtu saman tason käyttäjätunnuksella (esim. käyttäjätunnuksella "dna"). Operointityöasema on tällä hetkellä konfiguroitu sillä tavalla, että sen käynnistyessä saattaa automaattisesti käynnistyä kaksi kappaletta Metso OPC-palvelimia. Mikäli Metso OPC palvelin käynnistyy ns. tuplasti, tiedonsiirto Matlabin ja Metso DNA:n välissä ei tule onnistumaan.



Kuva 13. Ainoa oikea tapa käynnistää Metso OPC - palvelin on tehdä se Metso DNA- Slot Controller:in kautta. [23]

Ongelmatilanteessa avataan työaseman Windows Task Manager, klikataan auki Processes-välilehti ja sammutetaan toinen Metso OPC-palvelin käskyllä End Process. Jos yhteysongelma ei korjaannu, voidaan käynnistää uudelleen myös toinenkin Metso OPC-palvelin. Tässä työssä meneteltiin näin, että sammutettiin Task Manager-ikkunassa toinen OPC-palvelimista ja toiselle tehtiin Metso Slot Controllerin kautta uudelleenkäynnistys (Stop → Start).

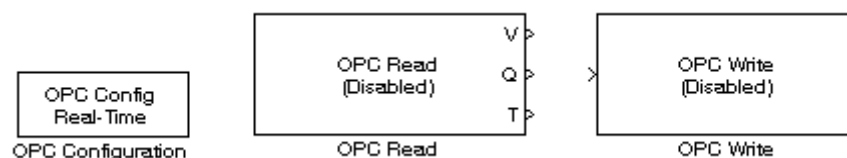
4 Matlab Simulink

4.1 Matlab OPC Toolbox

Alun perin Matlab oli tarkoitettu insinööri työkaluksi erilaisten laskutoimituksien tekemiseen jolloin ohjelmointi tapahtui komentoikkunan kautta. Matlabia on vuosien saatossa kehitetty ja ohjelmaan on liitetty erilaisia lisäosioita eli työkalupakkeja (*eng. toolbox*) jolloin siitä on tullut hyödyllinen simulointiväline. Myös ohjelmointi on helpottunut, sillä tämän päivän Matlab sisältää myös graafisen ohjelmoinnin ympäristön, joka tekee ohjelmoinnista entistä havainnollisemman. Nykyisin Matlabia voidaan hyödyntää simuloinneissa yhdessä teollisuusautomaatiojärjestelmien kanssa. Tällaisen käytön tekee mahdolliseksi Matlabin työkalupakki OPC Toolbox.

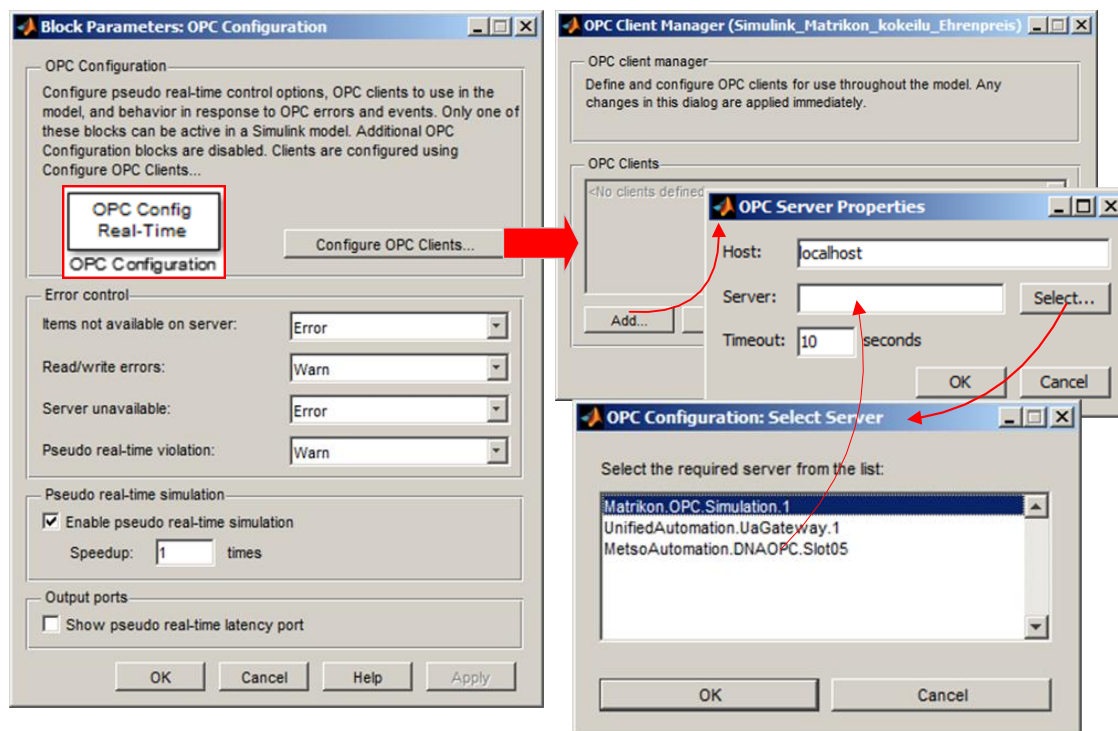
OPC Toolbox mahdollistaa Matlab Simulink:in yhteyden OPC DA- ja HDA-palvelimeen ja tukee OPC Foundationin kehittämiä standarteja Data Access v2.05 ja Historical Data Access v1.2. Lisäksi OPC Toolbox:illa on mahdollinen samanaikainen tietojen lukeminen ja käsitteleminen sekä yhteys useampaan OPC-palvelimeen. Yhteys OPC-palvelimelle voi olla määritelty synkroniseksi (tietomakasiini / älykäs kenttälaite) tai asynkroniseksi (älykäs kenttälaite).

Yhteyden luontiin tarvitaan kuvan 14 mukaisia OPC Toolbox-toimilohkoja.



Kuva 14. OPC Toolbox-toimilohkot: OPC Configuration; OPC Read; OPC Write.

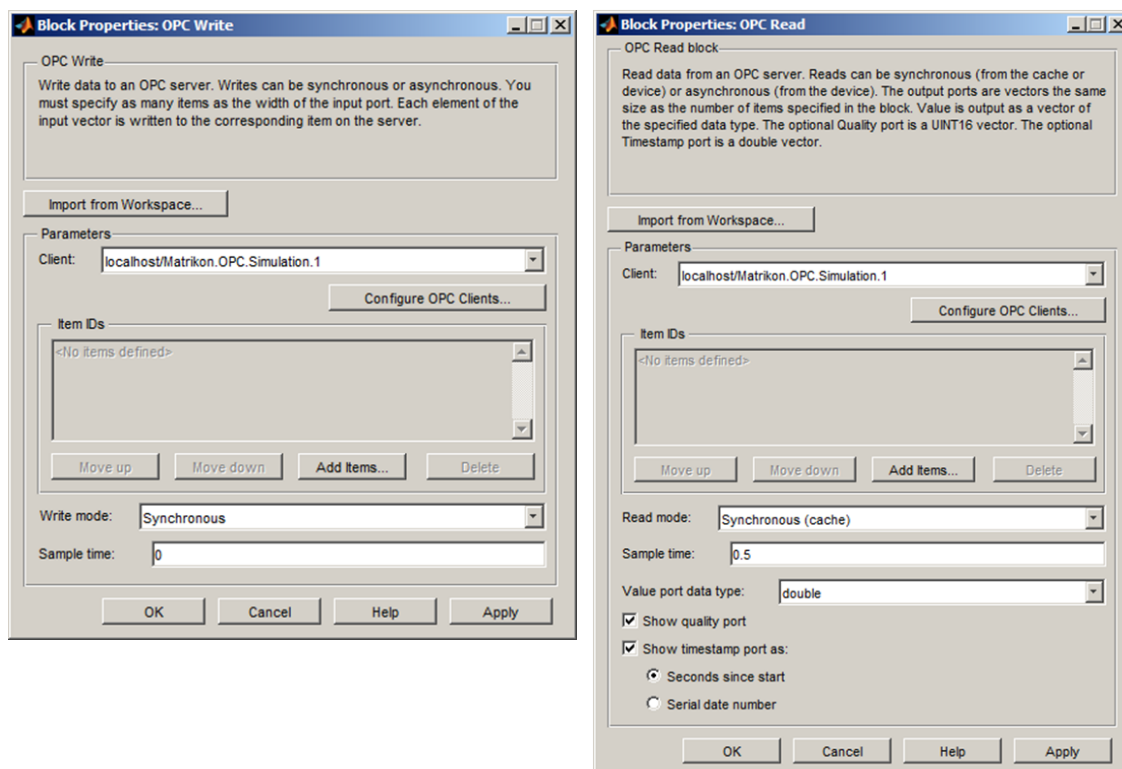
Jotta Simulink-malli voisi olla yhteydessä OPC-palvelimeen, täytyy määrittellä konfigurointilohkoissa (*OPC Configuration*) OPC-palvelimen asetukset (kuva 15). Konfiguroinnin yhteydessä täytyy määrittää millä koneella OPC palvelin sijaitsee, eli onko se Matlab Simulink:in kanssa samalla koneella (*localhost*), tai sitten etäkoneella (*remote*). OPC Configuration-lohko määrittää käytettävän OPC-palvelimen myös muille OPC-lohkoille.



Kuva 15. OPC Config-lohkon asetuksien määrittäminen. Työssä tämä lohko määrittää Matlab Simulink-prosessimallia varten OPC palvelimeksi MetsoAutomation.DNAOPC.Slot05-palvelimen. Host: localhost tarkoittaa, että kyseinen palvelin on asennettu Matlab:in kanssa samalle työasemalle.

Tietojen lukemiseen OPC-palvelimelta tarvitaan OPC Read-lohkoa, johon määritellään miltä palvelimelta ja millaista tietoa (tag) luetaan. Lohkolla on kolme ulostuloa: prosessisuure (V), laatu (Q), aikaleima (T). Lohkon tehtävänä on lukea OPC-palvelimelta tietoja Matlab Simulink-ympäristöön. Lohkon asetuksiin (kuva 16) määritellään mm. näytteenottotaajuus eli päivitystaajuus (*Sample time*) jonka perusteella lohko "tietää", kuinka usein pitää arvojaan päivittää. *Client*-kenttään merkitään minkä palvelimen asiakas on kyseinen Matlab prosessimalli (työn alkuvaiheessa tutkittiin *Matrikon.OPC.Simulaation*-palvelimen käyttömahdollisuuksia, mutta tämä ratkaisu osoittautui käyttökelvottomaksi). *Read mode* määrittää sen, mistä ja miten tietoja luetaan, eli luku voi tapahtua synkronisesti / epäsynkronisesti laitteelta tai tietokoneen muistista.

Prosessitietoja kirjoitetaan OPC-palvelimelle *OPC Write*-lohkon välityksellä. Samaan lohkoon voidaan määritellä useampia tag:ejä (Item IDs) kirjoitettavaksi palvelimelle. OPC-palvelimelle kirjoitettava tieto on toimenpiteen jälkeen muitten OPC-asiakkaiden käytettävissä.



Kuva 16. Matlab Simulonk OPC Toolbox:in OPC Write- ja Read- lohkojen konfigurointi-ikkunat.

Luku- ja kirjoitustilan (*Read mode*) valinnassa täytyy huomioida tiedon luonne, eli onko se aina luettavissa ja kirjoitettavissa tietyin aikaväleihin tai riippuuko tiedon saanti vallitsevista olosuhteista. Luku- ja kirjoitustila voi olla joko synkroninen tai asynkroninen. Siinä voidaan välittää tietoja palvelimen ja Matlab:in kesken (*cache*) tai Matlab:in ja laitteen kesken palvelimen kautta (*device*).

Synkroninen tila soveltuu käytettäväksi, mikäli tieto on jatkuvasti saatavilla ja sen välittäminen eteenpäin on järkevää. Portti pidetään konfiguroinnin mukaisesti auki tietyn ajan ja tietyin aikaväleihin tai siihen asti kun tiedon välitys on tapahtunut onnistuneesti. Tämänkin työn yhteydessä on valittu käytettäväksi synkroninen kirjoitus- ja lukutila, koska halutaan seurata signaalin muutosta ajassa jatkuvasti. Lukutila on tarkennettu termillä *cache*, jolloin luetaan OPC-palvelimelta eikä kenttälaitteelta.

Oletetaan, että meitä kiinnostava suure muuttuu hitaasti tai epäsäännöllisesti ja sen muutoksella on merkitystä vasta silloin kun se ylittää tietyn rajan (esim. lämpötilan mittaust). Siinä tapauksessa voidaan käyttää asynkronista luku- tai kirjoitustilaa, sillä signaalin pienet muutokset eivät välttämättä vaikuttaa olennaisesti prosessin kulkua ja jatkuva tiedonsiirto vaan rasittaa järjestelmää sekä alentaa mahdollisesti sen suoritus-

kykyä. Matlab lähettää palvelimelle suuren arvoa koskevan pyynnön ja mikäli palvelin hyväksyy kyseisen pyynnön, Matlab siirtyy seuraavien komentojen täyttämisen pariin. Asynkroninen luku tapahtuu aina suoraan kenttälaitteelta asti ja jos palvelin saa pyynnön kriteerit täyttävän tiedon haltuunsa, se lähettää signaalin arvon eteenpäin Matlab:iin. [25]

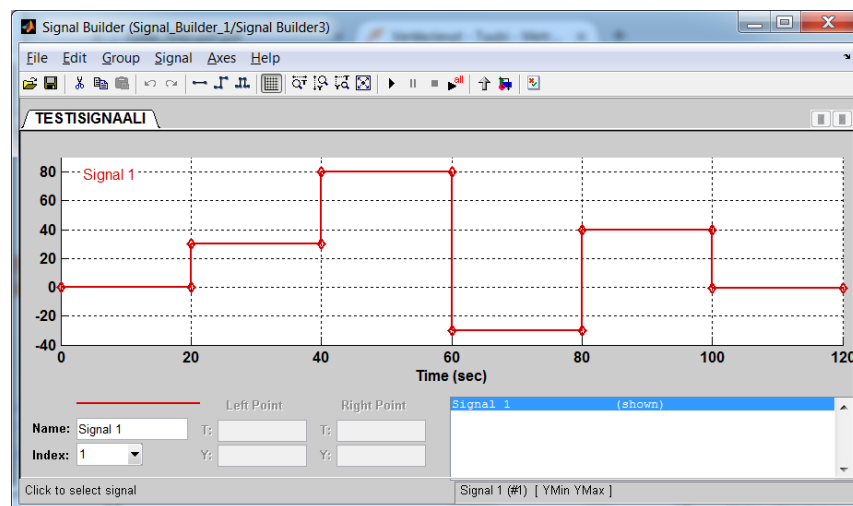
4.2 Signal Builder

Hyödyllinen apuväline askelvastekokeiden suunnittelussa on Matlab Simulinkin tarjoama Signal Builder-toimilohko. Työkalu sisältää monenlaisia vakio signaaleja joita voidaan summata keskenään tai käyttää simuloinneissa yksi kerrallaan.



Kuva 17. Matlab Simulinkin toimilohko Signal Builder.

Signal Builder:illa on mahdollista, työkalun nimensä mukaisesti, myös luoda itse tarpeen mukaan erilaisia signaaleja. Askelvastekokeiden suunnittelussa voidaan etukäteen muodostaa sopiva signaalipätkä, joka sisältää kaikki prosessin tutkimisen kannalta olennaiset ominaisuudet.

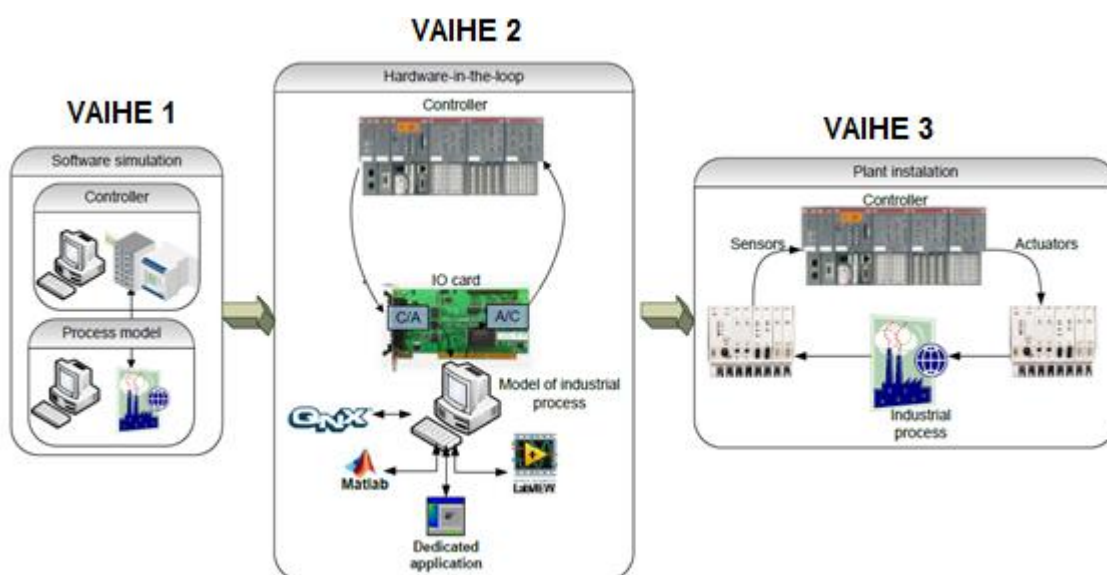


Kuva 18. Matlab Simulinkin toimilohkolla Signal Builder voidaan itse rakentaa monenmuotoisia testisignaaleja.

Valmiiksi ohjelmoitu Signal Builder-toimilohko kytketään simulaattoriin ja käynnistetään simulointi, jolloin luotu signaali syötetään prosessiin.

5 Simuloinnit

Simuloinneilla tähdätään automatisointiprojektien kustannusten pienentämiseen, suunnitteluvirheiden vähentämiseen ja käyttöönoton sujuvuuden takaamiseen. Automatisointitilasta seuraa suunnitteluprosessi, jonka tarkoituksena usein on luoda simulointiohjelmassa prosessimalli asiakkaan asettamien vaatimusten pohjalta. Suunnitteluvaiheessa valitaan sopivat säätömenetelmät ja suunnitellaan säätöpiirit, hälytykset, turvajärjestelmät, varajärjestelmät ym. (kuva 19 vaihe 1).



Kuva 19. Vaiheet automaatiojärjestelmän simuloinnista sen tehdasasennukseen [7,s.50].

Mikäli simulointiohjelmalla luotu prosessimalli on valmis testaukseen, voidaan siirtyä seuraavaan vaiheeseen (kuva 19 vaihe 2), eli kytketään simulointiohjelmalla luotu prosessimalli fyysiseen prosessiohjausjärjestelmään. Sillä tavalla saadaan hyödyllistä tietoa mahdollisista viiveistä ja muista fyysisen laitteiston asettamista rajoitteista. Myös tämän insinööriyden simulointiosuus kuuluu kuvan 19 vaiheeseen 2, koska simulointiohjelmassa Matlab Simulink luotu prosessimalli on kytketty fyysiseen prosessiohjausjärjestelmään Metso DNA. Tässä suunnitteluportaassa on mahdollista kerätä prosessista tietoa, jota käytännön tuotanto-olosuhteissa olisi vaikeaa tai jopa mahdotonta hankkia.

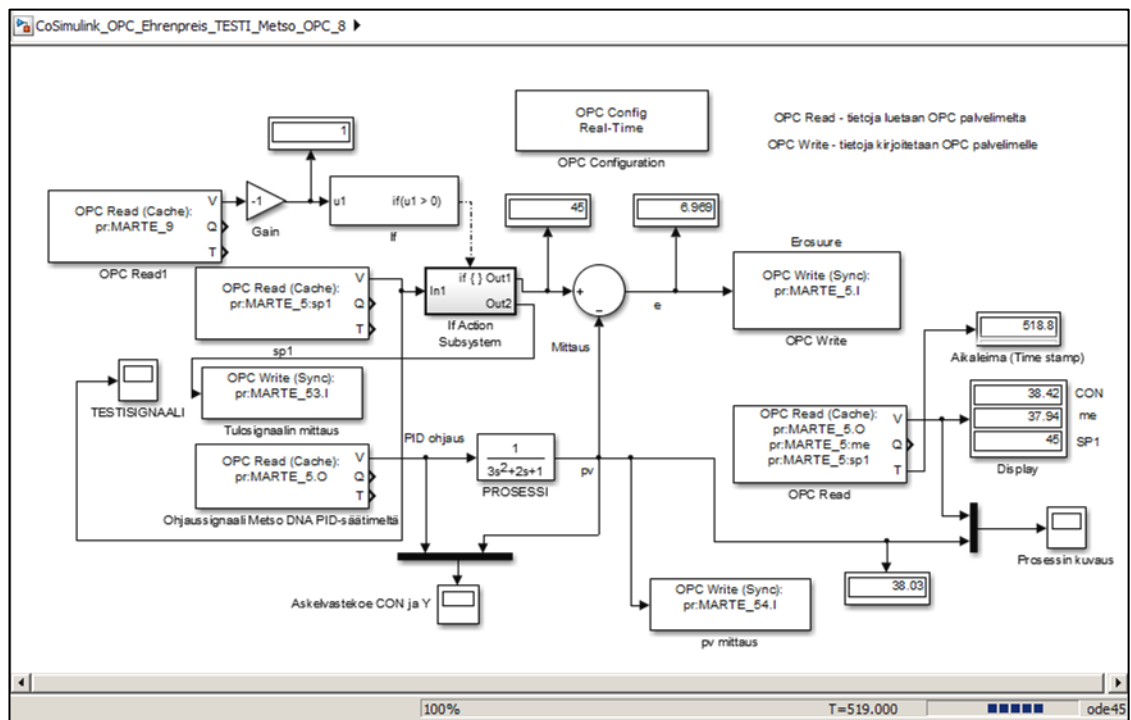
Ohjelmassa voidaan simuloida tilanteita jotka käytännön testeissä tulisivat kohtuuttoman kalliiksi tai osoittautuisivat vaarallisiksi.

- hälytysrajojen toiminta (esim. korkea lämpötila tai paine)
- laitoksen käynnistys ja alasajo
- turva- ja varajärjestelmien toiminta

Mikäli automaatiojärjestelmän suunnittelussa simuloinnin vaiheet 1 ja 2 ovat tuottaneet hyviä tuloksia, voidaan siirtyä järjestelmän tehdasasennuksiin (kuva 19 vaihe 3). Jos automaatiojärjestelmän ensimmäiset kaksi suunnitteluvaihetta on menestyksekkäästi läpäisty, sujuu todennäköisesti myös viimeinen vaihe eli käyttöönotto ilman suurempia ongelmia.

5.1 Simulaattori 1 (PID-säädin Metso DNA-järjestelmässä)

Prosessimalli luodaan Matlab Simulinkissa, mutta PID-säädin sijaitsee Metso DNA-automaatiojärjestelmässä. Prosessimallin ohjaus tapahtuu Metso DNA:n PID-säätimen käyttöliittymän kautta. Yhteys Matlab Simulink:in ja Metso DNA:n välissä hoidetaan Metson OPC serverin avulla. Prosessin trenditiedot kerätään talteen Metso DNA-automaatiojärjestelmän toimesta ja kuvataan Metso DNA Operate käyttöliittymän prosessitiedot ikkunassa.



Kuva 20. Ulkoisen PID-säätimen ohjattavaksi tarkoitettu Matlab Simulink-prosessimalli, jota ohjataan Metso DNA:n PID-säätimellä Metson käyttöliittymän kautta.

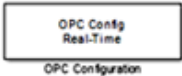
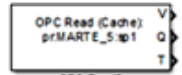
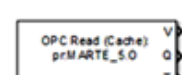
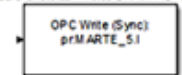
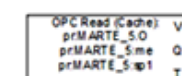
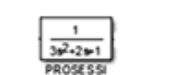

Prosessisimulaattori koostuu jo yläpuolelta tutuista Matlab Simulink-ohjelmiston OPC-yhteyden luomiseen tarvittavista moduuleista: *OPC Config*, *OPC Read* ja *OPC Write* (katso myös taulukko 2). Prosessimalli (kuva 20) on esitetty yleisessä muodossa, eli OPC-palvelimelle linkitetty kaikki ulkoisen säätimen mahdollisesti tarvitsemat portit. Myös prosessin erosuure lasketaan, jotta se olisi tarvittaessa ulkoisen säätimen käytettävissä. Erosuureen laskennassa simulaattorin (kuva 20) asetusarvo $r(t)$ luetaan OPC-palvelimen välityksellä ulkopuolisesta tietolähteestä Simulink-malliin (esimerkin tapauksessa Metso DNA-järjestelmän PID-säädinlohkon *sp1* liittimeltä). Summalohkossa ase-

tusarvosta vähennetään prosessin ulostulo $y = pv(t)$ eli mittaus jolloin saadaan eroosuure $e(t)$.

$$e(t) = r(t) - pv(t) \quad (1)$$

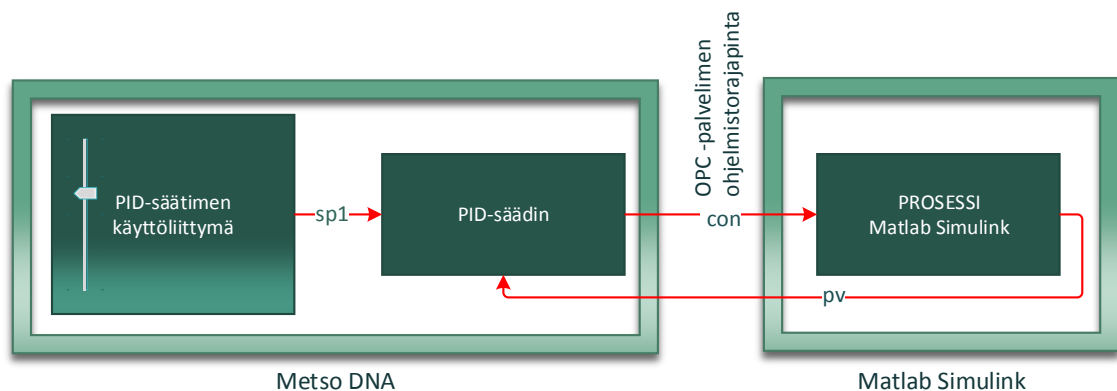
Metso DNA:n PID-säädin ei todellisuudessa tarvitse prosessimallin laskemaa erosuuretta, koska säädin laskee sen itse sen käyttöliittymästä syötetyn asetusarvon (pr:MARTE_5:sp1) ja mitatun prosessiulostulon (pr:MARTE_54.I) perusteella.

Taulukko 2. Askelvastekokeessa käytetyn simulaattorin komponentit

Askelvastekoe jossa Metso DNA PID säädin automaattilla.	
Toimilohko	Toimilohkon tehtävä
 OPC Configuration	OPC Config Lohkossa tapahtuu Matlab Simulink mallin ja OPC -palvelimen välisen yhteyden määrittely
 OPC Read2	OPC Read (Cache) Lukee OPC palvelimelta tag:in pr:MARTE_5:sp1 tietoja ja syötää signaalin summaelimeen. (asetusarvo)
 Ohjussignaali Metso DNA:n PID-säätimeltä	OPC Read (Cache) Lukee OPC palvelimelta tag:in pr:MARTE_5:0 tietoja ja syötää signaalin ohjattavaan prosessiin. (ohjussignaali Metso DNA:n PID-säätimeltä)
Erosuure lähetetään Metso DNA:n PID-säätimelle  OPC Write	OPC Write (Sync) Kirjoittaa OPC palvelimelle tag:in pr:MARTE_5:1 tietoja. (Metso DNA järjestelmään menevä prosessin eroosuure -signaali.)
 OPC Read	OPC Read (Cache) Lukee OPC palvelimelta tag:it pr:MARTE_5:0 pr:MARTE_5:me pr:MARTE_5:sp1 (simulinkkiin luettuja tietoja kuvataan simulaattorin numeronäytössä ja esitetään tunnuskäyrinä)
 PROSESSI	Kuvitteellinen prosessi
	Summaelin prosessin asetusarvon ja ulostulon välisen erosuureen laskentaan.

Tieto prosessin ulostulon arvosta lähetetään OPC Write lohkon välityksellä OPC-palvelimelle, josta se luetaan Metso DNA-automaatiojärjestelmän suorasaantiportin pr:MARTE_54.I kautta (kuva 23) FbCAD-sovelluksen PID-säätimen mittausliittimeen pr:MARTE_5:me. PID-säädin vertaa mittausliittimelle pr:MARTE_5:me tulevan signaa-

lin arvoa valvomosta syötettyyn asetusarvoon pr:MARTE_5:sp1 ja laskee niiden perusteella ohjaussignaalin arvon pr:MARTE_5:con. Viimeinen kopioidaan ccoa toimilohkon välityksellä suoransaantiporttiin pr:MARTE_5.O, josta signaali kirjoitetaan OPC-palvelimelle vastaavaan tag:iin. Palvelimelta ohjaussignaali pr:MARTE_5.O luetaan OPC Read (Cache) lohkon kautta Matlab Simulink-prosessimalliin. Ohjaussignaali päättyy prosessilohkoon (PROSESSI) jonka ulostulosta saadaan prosessisuure eli $y = pv(t) = mittaus$.



Kuva 21. Simulaattorin (kuva 20) periaatekuva josta käy yksinkertaistettuna ilmi signaalin kulku. Prosessin ulostulo (pv) luetaan PID säätimen mittausliittimeen (me). Ohjausviesti menee PID säätimen con-liittimeltä Simulink:in prosessilohkoon.

Signaalin laatu, kulkiessaan rajapinnasta toiseen, heikkenee, signaaliin tulee vaiheen siirto ja signaalin muotokin vääristyy eli tapahtuu ns. laskostuminen³. Signaalin laatua voidaan parantaa kasvattamalla ohjelmistorajapinnoissa signaalista näytteenottotaajuutta (*sampling rate*). Nyquistin teoreeman (2) mukaan A/D-muunnoksessa näytteenottotaajuuden on oltava vähintään kaksinkertainen digitalisoitavan analogiasignaalin taajuuteen verrattuna, jotta signaali ei menettäisi muunnoksessa huomattavasti ominaisuuksiaan. [20]

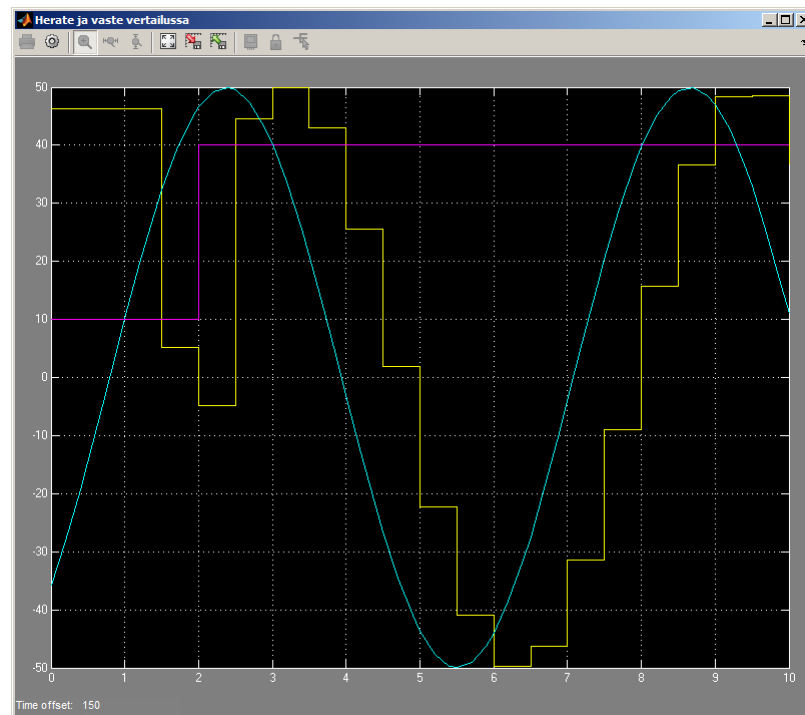
$$f_{\text{näytteenotto}} \geq 2f_{\text{signaali}}$$

(2)

Kuvasta 22 huomataan miten paljon *sini*-signaalin muoto muuttuu, kun se kulkee suhteellisen matalalla näytteenottotaajuudella Matlab Simulinkista Metso OPC-

³ Laskostuminen - myös aliasoituminen on signaalinkäsittelyssä signaalin kaistaleveyteen nähden liian pienestä näytteenottotaajuudesta johtuva signaalin vääristyminen. [20]

palvelimelle, palvelimelta Metso DNA FbCAD-sovellukseen ja sieltä takaisin OPC-palvelimen kautta Matlab Simulinkiin. Signaalissa tapahtuu myös vaihesiirto eli Metso DNA-järjestelmästä takaisin tuleva signaali on hieman viivästynyt alkuperäiseen testisignaaliin nähden. Näytteenottotaajuus täytyy valita tapauskohtaisesti riippuen signaalin laatuvaatimuksista ja mitattavan kohteen luonteesta. Toisaalta täytyy myös muistaa, että tarpeettoman suuri näytteenottotaajuus kuormittaa turhaan järjestelmiä. Matlab Simulinkin OPC-konfiguraatiolohkoissa määritellään signaalin näytteistystaajuus OPC Read ja OPC Write lohkoissa termillä Sampling Rate.

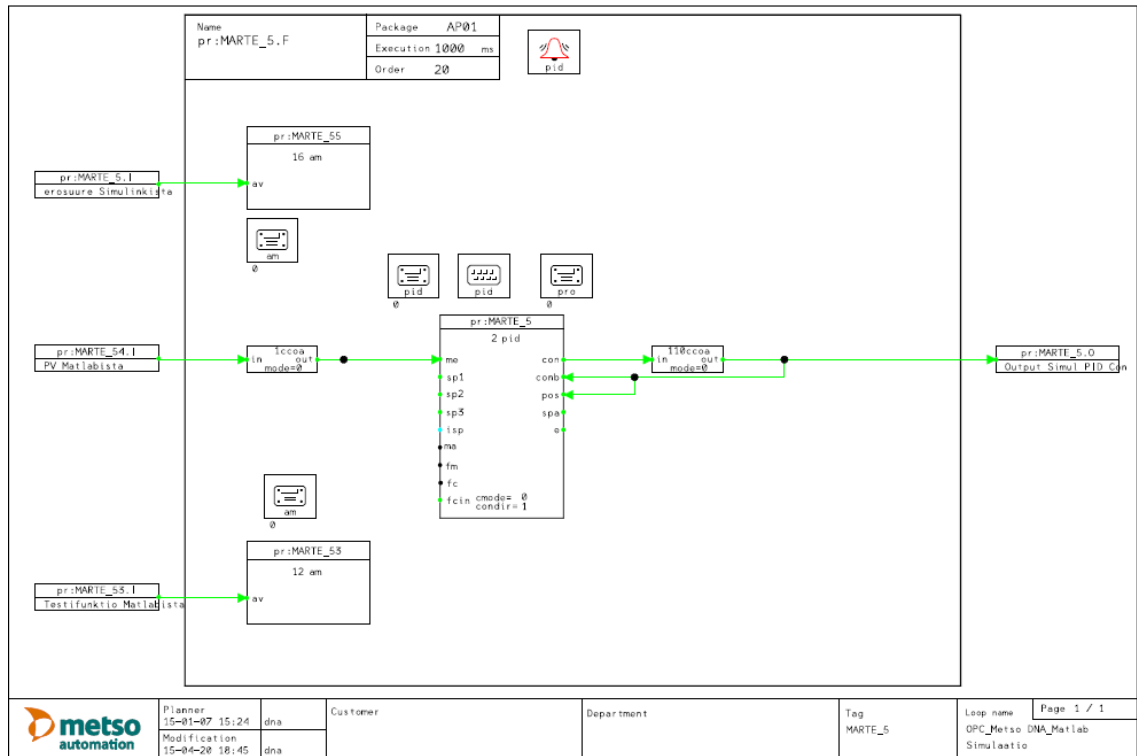


Kuva 22. Matlab Simulinkin syöttämä testisignaali (sininen), Metso DNA-järjestelmästä takaisin Matlab Simulinkiin luettu testisignaali joka on alkuperäiseen signaaliin nähden hieman viivästynyt (keltainen), asetuservo setpoint (violetti).

Alla olevassa kuvassa (kuva 23) on Metso DNA:n graafisen suunnittelutyökalun FbCAD avulla luotu automaatiomoduli. Kuvan vasemmassa reunassa sijaitsevat prosessinohjauspalvelimen tulo- ja oikeassa reunassa lähtömoduulit. Keskialueelle kehyksen sisälle jäävät prosessinohjauspalvelimen toimintamoduulit, valvomon positio-, tapahtuma- ja operointimoduulit.

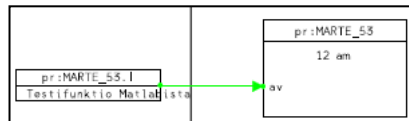
Toimintamoduuleina on käytössä analogiamittaus toimilohkot pr:MARTE_53 ja pr:MARTE_55, joilla mitataan ohjelmassa Matlab Simulink sijaitsevasta prosessimallista vastaavasti testifunktion ja prosessin erosuureen arvoja.

PID-säätimen liittimien tausta on koottu taulukkoon 3, mutta kytkennöistä olkoon vielä mainittu, että säätimen ohjaussignaali on takaisin kytketty säätimen conb ja pos porttiin. Nämä portit ovat normaalisti kytketty I/O korttien vastaavanimisiin portteihin, jolloin conb on ohjaussignaalin takaisinkytkentä ja pos on portti jonka kautta säädin saa tiedon toimilaitteen asennosta.



Kuva 23. FbCAD sovellus simulaattorissa jonka PID - säädin sijaitsee Metso DNA- ympäristössä.

Suorasaantiportti (Direct Access Port) - portin nimi on yksikäsitteinen kaikkialla sovellusverkossa, eli sillä nimellä ei voi esiintyä muita portteja. Suorasaantiportiin voidaan kytkeä joko yksittäinen tietopiste tai koko toimilohko (kuva 24).



Kuva 24. Suorasaantiportti (pr:MARTE_53.I) jonka kautta Metso DNA-automaatiojärjestelmä saa käyttöönsä Matlab Simulinkissa olevan prosessimallin testisignaalin tiedot. Vaihtoehtoisesti voitaisiin käyttää myös am-lohkoon integroitua suorasaantiporttia (pr:MARTE_53).

Jos porttiin kytketään tietopiste, saadaan sen tieto moduulin ulkopuolelle. Koko toimilohkoon kytkettävän suorasaantiportin välityksellä saadaan kaikki porttiin kytketyn toimilohkon jäsenet muiden moduulien käytettäväksi. Työssä käytetään suorasaantiportilla varustettuja PID-säätimen ja analogiamittauksen (am) toimilohkoja.

Seuraavaksi esitellään PID-toimilohko jossa on integroituna suorasaantiportti. Portin välityksellä on koko verkon käytettävissä kaikki lohkon jäsenet (taulukko 3). Tämän työn yhteydessä on käytetty suorasaantiportteja ns. tupla määrä signaalien kulun havainnollistamisen tarkoituksessa. Käytännössä oltaisi voitu luopua tulo- ja lähtöportteista kokonaan koska kaikki tarvittavat signaalit on mahdollista toimittaa suorasaantiporttien välityksellä toimilohkoille ja myös ohjelmasta ulospäin. [21 s.15 - 16]



Kuva 25. PID-toimilohko jonka yhteydessä on suorasaantiportti pr:MARTE_5 (suorakaide lohkon yläosassa).

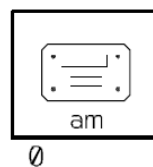
Edellä mainitun kanssa samantyyppisiä suorasaantiportteja voidaan huomata myös kuvan 23 analogiamittaus toimilohkoissa. Toimilohkojen suorasaantiporttien käyttöä havainnollistetaan tämän työn puitteissa vielä prosessin trendinäytön suunnittelun sekä Matlab Simulink-prosessimallin konfigurointimoduulien yhteydessä.

Taulukko 3. PID-säätimen suorasaantiportin välityksellä ovat säätimen kaikki jäsenet koko ohjelmaverkon käytettävissä.

PID -säätimen suorasaantiportin käyttö	
pr:MARTE_5:me	mittaus me
pr:MARTE_5:sp1 :sp3	asetusarvot sp1.....sp3
pr:MARTE_5:isp	asetusarvon valintaindeksi
pr:MARTE_5:ma	toimintatila MANU /AUTOM
pr:MARTE_5:fm	pakko käsiohjaus
pr:MARTE_5:fc	pakko-ohjaus (tulon arvo lähtöön)
pr:MARTE_5:fcin	pakko-ohjauksen suuruus
pr:MARTE_5:con	ohjaus
pr:MARTE_5:conb	takaisin luettu ohjaus
pr:MARTE_5:pos	toimilaitteen asento
pr:MARTE_5:spa	asetusarvo (aktiivinen eli säätimen näkemä arvo)
pr:MARTE_5:e	eroarvo = asetuservo - mittaus

Positio-, operointi- ja tapahtumatoiminnoilla siirretään tietoa valvomon ja prosessiase-
man välissä.

Positiotoiminto lohkot (kuva 26) ovat yhteisiä saman valvomon hälytyskäsittelijöille ja operointipalvelimille. Ne ovat käytössä erilaisia ilmoituksia varten ja esimerkiksi mahdollistaa tapahtumaselaimessa hälytyksen vieressä suureen nykyisen elävän arvon ilmaisemisen. Itse positiotoiminnon symboli on aina samanlainen, mutta sen alaosassa on ko. toimilohkon tunnus ja livepositiotoiminnon symbolissa vielä lisäksi tapahtumaselaimen kuva oikeassa yläkulmassa. [14 s. 119; 131]



Kuva 26. Positiotoiminto lohkoa käytetään erilaisten ilmoitusten esittämiseen

Positiotoiminnot hallitsevat yhden luodun prosessipiirin tapahtumapisteitä binäärityyppisillä signaaleilla. Positiotoimintoa käytetään yleensä trendi-, piirturi- hälytys-, piiri- ja ryhmäkuvissa.

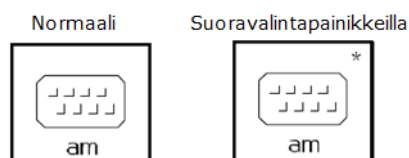
Tässä työssä positiotoiminto on käytössä PID-toimilohkon kytkennässä DNAuseEditor Picture Designer kuvaan (pidpos). Kuva avataan Metso DNA Operate ympäristössä jolloin se tarjoaa automaatiojärjestelmän operaattorille monipuolisen PID-säätimen käyttöliittymän.

Lohkon asetuksissa määritellään mm. seuraavat tiedot:

- mittausalueen ylä- ja alaraja
- ohjauksen ylä- ja alaraja
- position operoitavuus (kyllä tai ei)
- mittaus- ja asetusarvon mittausalue
- suureiden yksiköt
- ohjaus ja mittausrajojen ilmoitukset

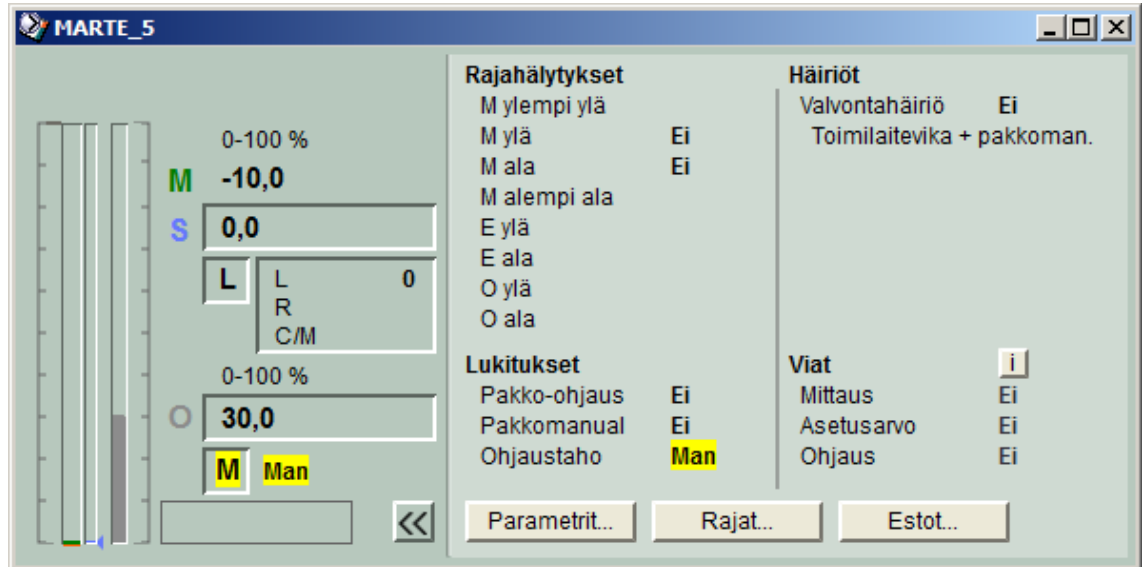
Käyttöliittymän kautta operaattori voi, riippuen käyttöoikeuksista, muuttaa PID-säätimen parametreja, kytkeä säätimen automaatti- tai manuaalitilaan, muuttaa ohjauksen arvoa (O), asetusarvoa (SL, SR, SC/M), asettaa suureilla hälytysrajoja ja estoja. Rajapinta informoi käyttäjää myös rajahälytyksistä, lukituksista, häiriöistä, ja vioista.

Operointitoiminto lohko. Operointitoiminnot välittyvät prosessinhoitajien valvomo-ohjelmistosta laitteille OPS- aseman kautta. Operointitoiminto lohkoja (kuva 27) on kah- ta tyyppiä: normaali operointitoiminto ja operointitoiminto, jossa on suoravalintapaini- keet.



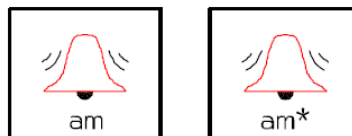
Kuva 27. Operointitoiminnon lohkot

Operointipalvelimen kuvamoduuleissa tapahtumatiedot liitetään jokaiseen kuvatoimilohkoon erikseen ja kytkentä tapahtuu operointimoduulin kautta. Kuvassa 28 on esitetty PID-säätimen käyttöliittymä joka hyödyntää tiedonvälityksessä operointitoimilohkoa.



Kuva 28. PID-säätimen ohjaukseen DNAuseEditor Picture Designerissa luotu kuva joka on avattuna Metso DNA Operate ympäristössä.

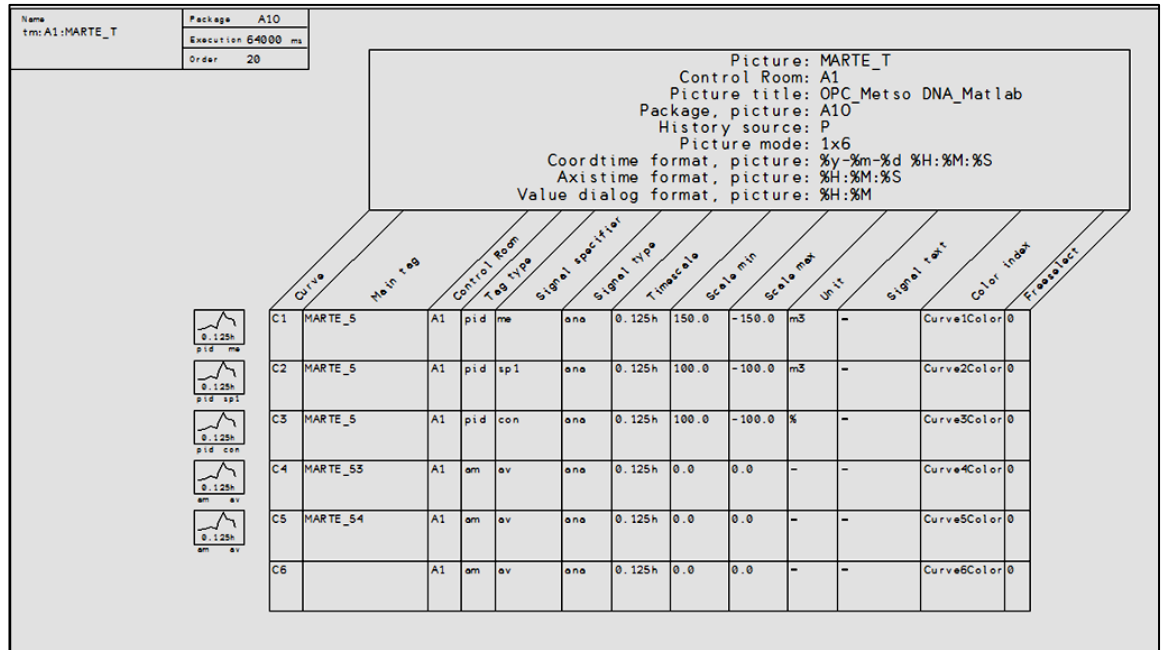
Tapahtumatoiminto lohko (kuva 29) välittää hälytyksiä ja ilmoituksia valvomoon ennalta asetettujen kriteerien perusteella. Piirikohtaisesti voidaan hälytyksiä ja tapahtumia priorisoida.



Kuva 29. Tapahtumatoiminnon lohkot.

Esimerkiksi mittauksesta voidaan muodostaa tapahtuma suureen ylä- tai alarajan ylitykseen tai tulosignaalin laatuun liittyen.

Trenditietojen tallennus Metso DNA-järjestelmässä tapahtuu kuvan 11 osoittamalla tavalla. Signaalitietojen tallennus koskee kuvan 30 mukaisessa järjestyksessä seuraavia prosessisuureita: PID-säätimen MARTE_5 mittaustietoa (me), asetustietoa (sp1), ohjaussignaali (con), Matlab Simulinkin toimilohkon prosessiin syöttämää signaalia MARTE_53 (input) ja Matlab Simulink-prosessimallin ulostulua MARTE_54 (pv).



Kuva 30. Signaalien trendinäytön suunnittelu Metso DNA FbCAD työkalulla.

FbCAD:in trendien suunnittelu ikkunaan voidaan syöttää jokaiselle trendille signaalin vapaaehtoisia tietoja. Nämä ovat tietoja, jotka eivät ole tietoteknisesti välttämättömiä signaalin kuvaamisen kannalta (esimerkiksi: skaalauksetiedot, suureiden yksiköt, signaalin teksti, käyrän väri).

5.1.1 Säädön automaattinen viritys (simulaattori 1)

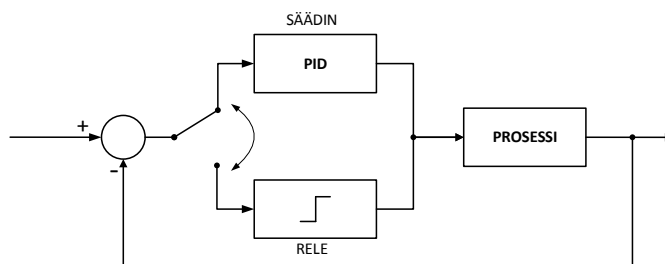
Nykyaikaisissa automaatiojärjestelmissä on yleensä integroituna säädön automaattisen virituksen työkalu. Ideaalilanteessa tällainen työkalu kerää prosessin käyttäytymisestä tietoja ja simuloi prosessin säädintä varten optimaaliset parametrit, joihin on mahdollista tehdä vielä muutoksia ennen parametrien käyttöönottoa prosessin ajoympäristössä. Säädön automaattisessa virityksessä noudatetaan pääsääntöisesti taulukossa 4 tuotua työjärjestystä. Työssä käytetään säätimen parametrien määrittämiseen kahta automaattista viritystyökalua: DNA Engineering Loop Auto Tuner:ia ja Matlab Simulink-PID Tuner:ia.

Taulukko 4. Virittämisen vaiheet käytettäessä viritystyökaluohjelmistoa [8, s.138]

1.	Suoritetaan prosessikokeet jolloin kerätään dataa
2.	Mallinnetaan järjestelmä
3.	Lasketaan PID -säätimen parametrien mallin perusteella
4.	Analysoidaan järjestelmä esim. simuloimalla
5.	Ladataan järjestelmään uudet PID-säätimen parametrit
6.	Raportoidaan tuloksista

DNA Engineering Loop Auto Tuner on tehokas työkalu, joka ideaalilanteessa vaatii käyttäjältä viritysprosessin aloittamiseen vaan säätöpiirille sopivan tavoitenopeuden määrittämisen ja lopuksi ehdotettujen säädinparametrien sijoittamisen säätimen aseuksiin. Säätöpiirin virituksen ensimmäisessä vaiheessa prosessimalli identifioidaan ja seuraavaksi lasketaan prosessimallille uudet parametrit.

Automaattinen viritystyökalu toimii normaalitapauksessa siten että, se kytkee säätimen rinnalle releen, joka saa prosessin värähtelemään prosessin ominaistajuudella (kuva 31), ja kerää sen jälkeen prosessista tarvittavaa tietoa säätimen viritysparametrien laskentaan (katso Liite 1).



Kuva 31. Relevitetyksessä PID-säädin korvataan releellä prosessikokeen ajaksi [8,s.116].

Loop Auto Tuner:in avulla voidaan suorittaa myös passiivinen viritys, jossa ei suoriteta ollenkaan koetta vaan viritystyökalu kerää ainoastaan prosessista dataa. [9, s.9]

Metson automaattinen viritystyökalu huomautti heti viritysprosessin alkutaipaleella, että säätö on saturoitunut⁴. Tässä tapauksessa Metso DNA:n PID-säätimen optimaalisen toimintapisteen löytäminen DNA Engineering Loop Auto Tuner avulla muuttuu mahdolliseksi.

Ilmiö on tyypillinen digitaalisille säätimille, koska näiden tapauksissa ei ole järjestelmässä niin sanottuja fyysikaalisia rajoituksia. Voidaan päätyä integraalin *wind-up*⁵ tilanteeseen jossa toimilaitteen ääriasento on saavutettu, mutta nolasta poikkeavan erosuureen ansiosta säätimen integroiva termi jatkaa kasvuaan rajatta (laskettu ohjaus ryömii). Kaikissa toimilaitteissa on omat fyysikaaliset rajoituksensa, kuten moottorin pyörintänopeus ei voi kasvaa rajattomasti ja myös säätöventtiilillä on omat ääriasentonsa. Teollisuudessa käytetään yleensä säätimiä joissa on oltava *AntiWind-Up*-rakenne, jolla estetään säätimen *Wind-Up*-käyttäytyminen. [8, s. 58 - 59]

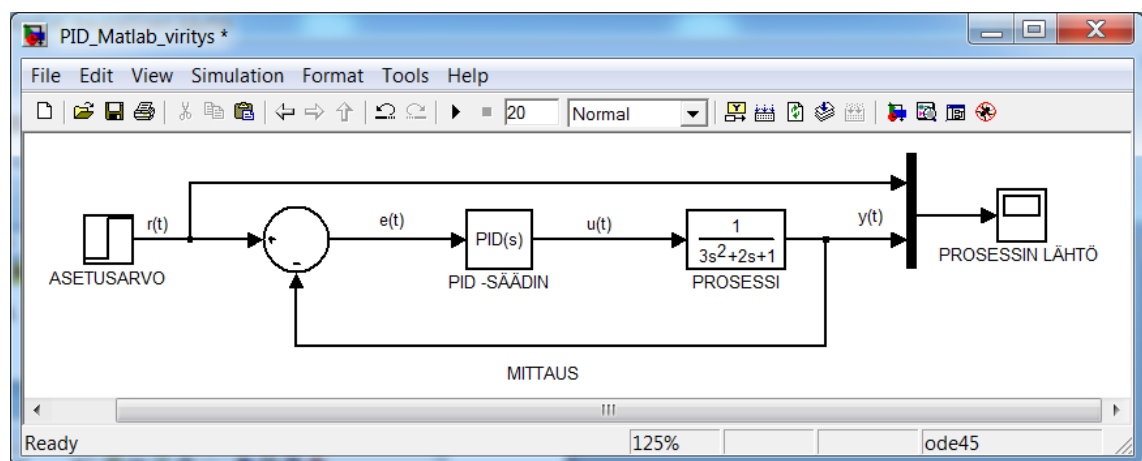
Simulaattorissa käytettävän Metso DNA:n PID-säätimen saturoitumiseen liittyy ilmeisesti säätimen ulostulon (con) rajoittaminen tiettyyn kapeaan arvoväliin. Säätimen tuottama laskennallinen ohjaussignaali ylittää mahdollisesti asetettujen signaalirajojen ulkopuolelle, mutta säätimen varsinainen ulostulo pysyy ääriarvojen comi ja coma sisällä. Säätimen saturoituminen huomataan siitä, ettei säädin reagoi enää asetusarvomutoksiin eikä pysty pienentämään erosuuretta, säätimen tuottama ohjaussignaali on ajautunut ääriasentoon. (kuva 32). Ohjaus ei toimi ja sen huomaa myös erosuureen (sininen tunnuskäyrä) loppuarvosta. Ideaalitilanteessa erosuureen pitäisi säädön rauhoittuessa päätyä nolaa. Säätimen saturoitumiselta voidaan välttyä estämällä integrointi silloin, kun laskennallinen ohjaussignaali on säätimeen konfiguroitujen rajojen ulkopuolella.

⁴ saturoitunut säätö - säätimet jossa on integroiva termi voivat käytön aikana saturoitua jolloin säädin menettää väliaikaisesti toimintakykynsä. Saturoitumista (*integraalin wind-up*) esiintyy kun säätimen lähtö (ohjaussignaali con) on rajoitettu, säätimen laskema ohjaussignaali on todellisten rajojen ulkopuolella.[18 s.4]



Kuva 32. Säättö on saturoitunut eli ohjaussignaali (oranssi) ajautuu asetusarvon askelmuutoksessa (vihreä) nopeasti ääriasentoon 100 % (rajoittimeen).

Koska säädettävä prosessi sijaitsee Matlab Simulink-ympäristössä, voidaan vertailun vuoksi PID-säätimen viritysparametrit laskea myös Matlab:in tarjoamalla viritystyökalulla. Muodostetaan kuvan 33 mukainen prosessimalli, mikä sisällöltään vastaa prosessikokeissa käytettyä mallia (kuva 20).



Kuva 33. Luodaan sisällöltään kuvaa 20 vastaava prosessimalli Matlab Simulink-ympäristössä, jotta voitaisiin vaihtoehtoisesti laskea PID-säätimen parametrit Metso DNA-järjestelmän PID-säädintä varten.

Matlab Simulink-ohjelman PID-säätimen viritystyökalu PID Tuner pitää sisällään ominaisuuden (anti-windup) joilla voidaan tarvittaessa estää säädön saturoituminen.

PID-lohkon oletusparametreina on käytössä (kuva 34):

$$P = 1; \quad I = 1; \quad D = 0$$

Virityksen jälkeen (kuva 34 ja 35) nähdään, että uusilla parametreilla prosessi saavuttaa huomattavasti nopeammin asetusrvon ja värähtelyn amplitudi sekä värähtelyyn kulunut aikakin pienenevät olennaisesti.

PID-säätimen toiminnan paremmaksi ymmärtämiseksi esitetään seuraavaksi lyhyt yhteenveto säätimen osien vaikutuksesta säätimen käyttäytymiseen.

P-säädössä jatkuva-aikainen ohjaussignaalin arvo $u(t)$ riippuu suoraan erosuureen arvosta $e(t)$. Ohjaussignaalin vakiotaso u_0 asetetaan sellaiseksi, että valitussa säätimen valitussa toimintapisteessä ei esiinny säätövirhettä. Matemaattisessa mielessä kyseessä on pelkkä kerto- ja yhteenlasku. Säätimen vahvistuksen K_p etumerkki määrittää säätimen toimintasuunnan. Jos vahvistus on positiivinen, säätimen toimituunta on suora, eli jos erosuure kasvaa niin myös säätimen vahvistus kasvaa.

$$u(t) = K_p e(t) + u_0 \quad (1)$$

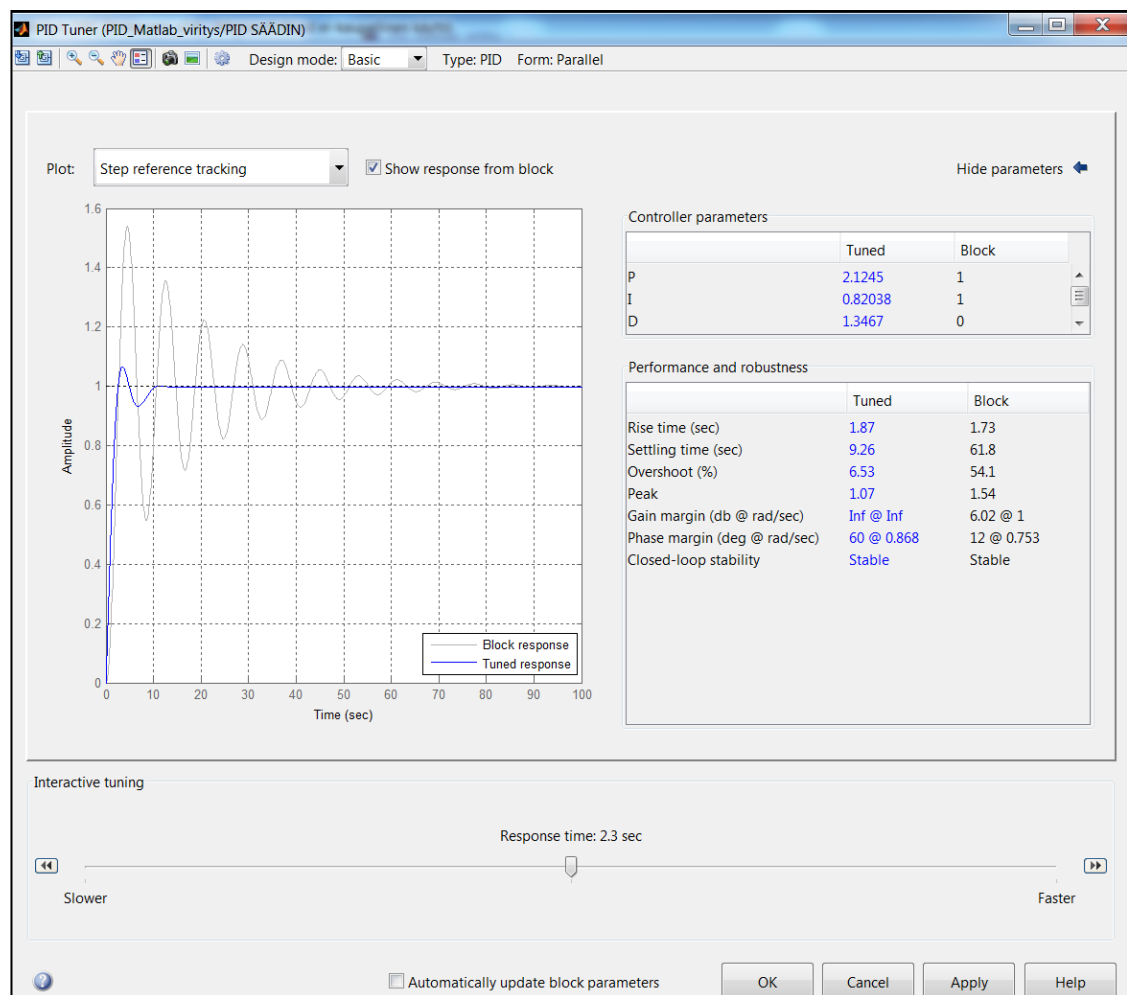
I-integraiva säätö on yleensä käytössä P-säädön kanssa ja se poistaa viimeiselle niin tyypillisen säätövirheen. Säätimen lähtö muuttuu niin kauan kunnes erosuuren arvo on nolla. Suure T_i on integrointiaika ja vahvistuskäyrän kulmakerroin riippuu integrointitermin vahvistuksesta $\frac{K_p}{T_i}$. Integrointi tarkoittaa matemaattisesti pinta-alan laskua ja säätimen integraiva termikin laskee erosuureen ja nolla-akselin välisen pinta-alan. Tavoitteena on pienentää kyseinen pinta-ala nollaan. Mitä pienempi on säätimen integrointiaika sitä suurempi I -termin vaikutus $\left(\frac{K_p}{T_i}\right)$. PI-säädin on teollisuudessa yleisimmin käytetty säädintyyppi ja sen matemaattinen ulkoasu on seuraava:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int e(t) dt = K_p \left(e(t) + \frac{1}{T_i} \int e(t) dt \right) \quad (2)$$

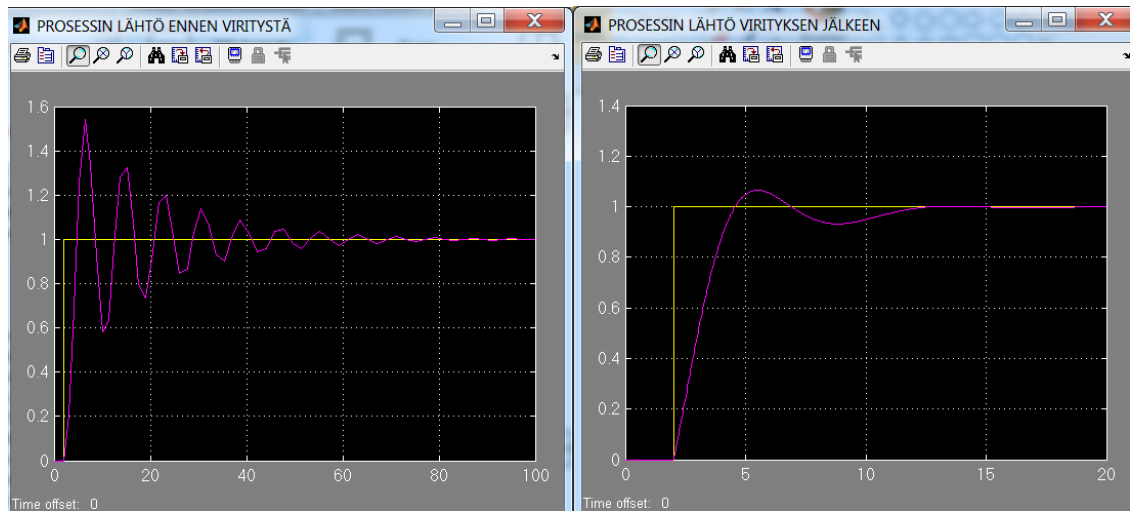
D-derivoivaa säätöä käytetään yleensä P-säätimen kanssa (PD) tai PID-säätimen kanssa jolloin ohjaussignaali seuraa kaavaa:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int e(t) dt + K_p T_d \frac{de(t)}{dt} = K_p \left(e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (3)$$

Säätimen derivoiva osa reagoi erosuuren muutosnopeuteen, eli jos viimeinen on nolla, niin D-osan lähtökin on nolla. Säätimen derivoiva osa saa erosuuren muutoshetkellä aikaan ohjauspiikin. Derivointi tarkoittaa matemaattisesti, että käyrään sovitetaan derivointipisteessä suora joka sivuaa käyrää siinä pisteessä. Derivaatan avulla kuvitellaan, että erosuure kehitty tulevaisuudessakin samansuuntaisesti kun tarkastuspisteessä. Mitä suurempi on derivointiaika, sitä suurempi on D-termin vaikutus. Yleensä käytetään derivoivan termin yhteydessä myös alipäästösuodatinta, jolla pienennetään korkeataajuisen kohinan vaikutusta ohjaussuureeseen. [8 s. 45 - 51]



Kuva 34. Matlab Simulink-viritystyökalulla PID Tuner saatu viritystulos. Vertailun vuoksi ovat vieressä näkyvissä myös alkuperäiset PID-lohkossa oletusasetuksina olleet parametrit.



Kuva 35. Prosessin lähtö ennen ja jälkeen PID-Tuner virityksen.

Eri automaattisilla viritystyökaluilla ja PID-säätimillä voidaan saada saman prosessin ohjaukseen hyvinkin toisistaan poikkeavia lukuja (kuva 36). Poikkeamat johtuvat todennäköisesti Metso DNA:n ja Matlab Simulinkin PID-säätimien rakenteellisista eroista. Matlab PID Tuner ehdottaa yli 10 kertaa pienempää integrointiaikaa, joka vaatii säätimeltä hyviä Anti Windup-ominaisuuksia. Metson PID-säätimen testauksessa oli säädön saturoitumisen vaara, joten tästä syystä viritystyökalu suosittelee varovaisempia integrointiaikoja.

DNA Engineering Loop Auto Tuner				Matlab PID -Tuner		
Positio:	MARTE_5			Controller parameters		
Kuvaus:	OPC_Metso DNA_Matlab				Tuned	Block
Säätimen tyyppi:	PID			P	2.1245	1
Tavoitenopeus:	Normaali			I	0.82038	1
				D	1.3467	0
Säätimen parametrit				Performance and robustness		
	■ Vanhat	■ Uudet	Prosessimalli:		Tuned	Block
Kp:	3	0,47	Vahvistus (K)	2,01	1.87	1.73
Ti:	20	12,62	Aikavakio (T)	11	9.26	61.8
Td:	0	1,41	Viive (L)	3,24	6.53	54.1
					Peak	1.54
					Gain margin (db @ rad/sec)	6.02 @ 1
					Phase margin (deg @ rad/sec)	12 @ 0.753
					Closed-loop stability	Stable

Kuva 36. Vertailu automaattisilla PID-säätimen viritystyökaluilla Metso DNA Engineering Loop Auto Turner ja Matlab PID Tuner lasketuista säätimen parametreista (katso Liite 1).

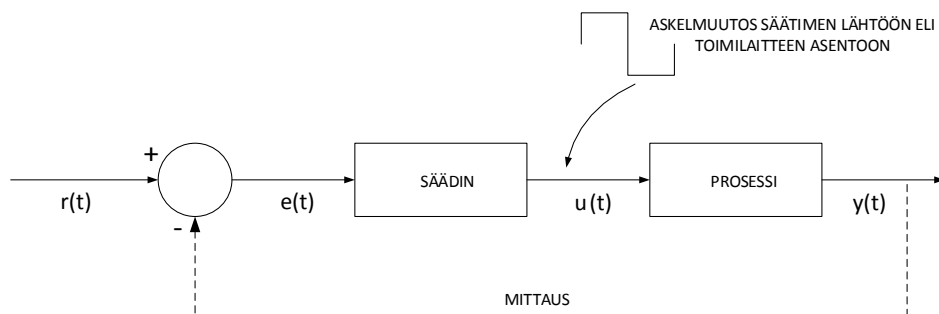
koskevassa lisäikkunassa (kuva 38) voidaan skaalata prosessin tunnuskäyrät tarpeen mukaan tai valita automaattinen skaalaus Auto: min-max. Voidaan myös prosessisuurien välisen riippuvuuden paremman havainnollistamisen nimissä vaihtaa esimerkiksi pysty akselin suunta signaali kohtaisesti.

Skaalaus/Käyrätiedot										
	Näytä käyrä	Merkki	Tarkennin	Yksikkö	Moodi	Alaraja	Yläaraja	Keskipiste	Alue	
MARTE_5	<input type="checkbox"/>	<input type="checkbox"/>	me	m3	Auto: min-max	-25	2	-11	28	
MARTE_5	<input type="checkbox"/>	<input type="checkbox"/>	sp1	m3	Auto: min-max	-0	0	0	1	
MARTE_5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	con	%	Auto: min-max	18	42	30	25	
MARTE_53	<input type="checkbox"/>	<input type="checkbox"/>	av	%	Auto: min-max	19	21	20	2	
MARTE_54	<input checked="" type="checkbox"/>	<input type="checkbox"/>	av	%	Auto: min-max	18	45	31	28	

Kuva 38. Trendi-ikkunassa skaalaus työkalussa aktivoidaan vaan ohjaus (con) ja mittaussignaalit (MARTE_54).

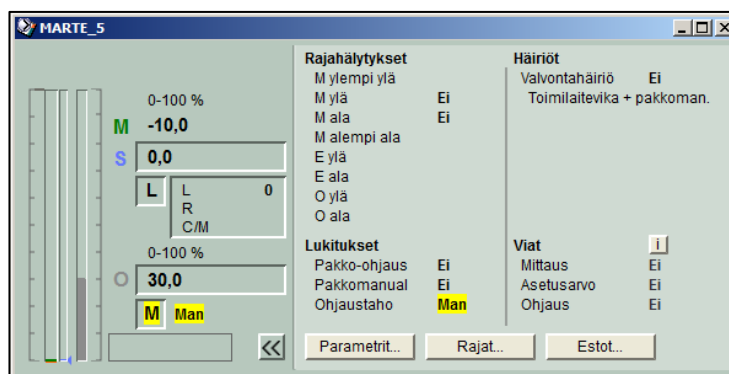
Askelkoe suoritetaan seuraavaa tehtäväjärjestystä noudattaen [8 s. 89]:

- säädin laitetaan käsiajolle
- piiristä pyritään poistamaan kuormitushäiriöt (koskee enemmän käytännön kohteita)
- tehdään askelmäinen muutos säätimen lähtöön eli toimilaitteen asentoon (katso kuvat 39 ja 41)
- otetaan talteen sekä ohjaus, että mittaussignaali kunnes prosessi on asettunut uuteen toimintapisteeseen



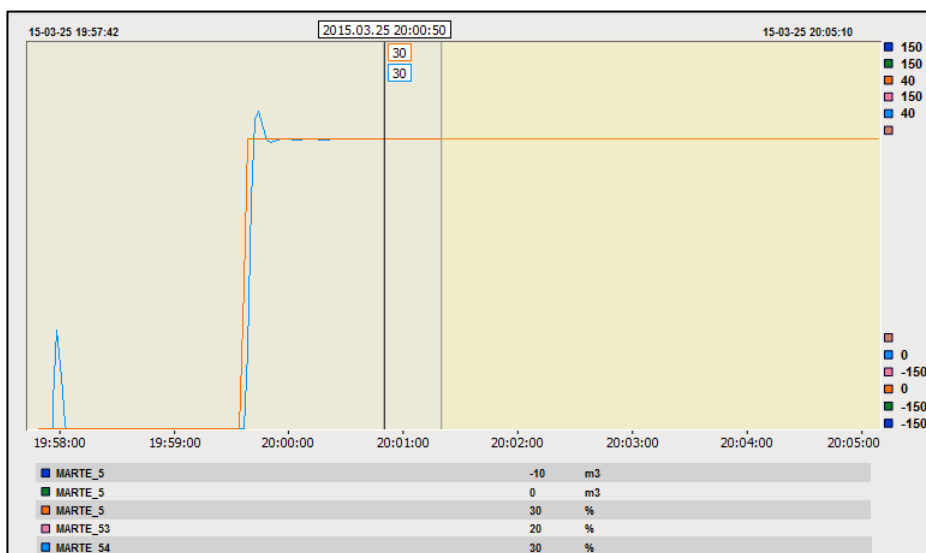
Kuva 39. Askelkokeessa tehdään askelmäinen muutos säätimen lähtöön eli toimilaitteen asentoon. [8 s. 89]

Askelmaisen muutoksen tekeminen säätimen lähtöön on hyvin yksinkertaista Metso DNA- Operate ikkunassa PID-säätimen käyttöliittymän kautta. Klikataan säätimen käyttöliittymässä auki ohjauksen (O) ikkuna ja syötetään siihen uusi ohjauksen arvo (kuva 42). Heti, kun prosessi on rauhoittunut, voidaan Enter-painikkeella aktivoida ohjaussignaalin uuden arvon.



Kuva 40. PID-säätimen käyttöliittymä Metso DNA- Operate ympäristössä. Tehdään säätimen lähtöön (O) askelmainen muutos (ohjausmuutos 0 --► 30 %)

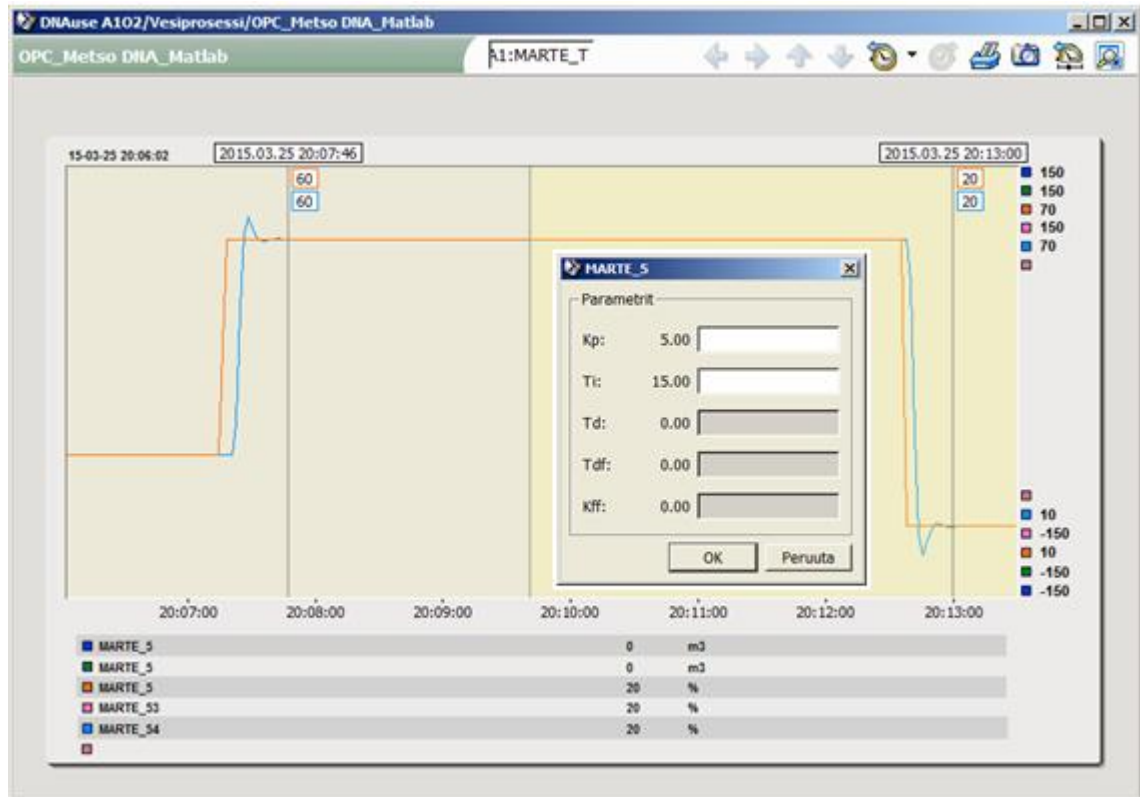
Kuvassa 42 nähdään Metso DNA- järjestelmän PID-säätimen HMI, jonka kautta tehdään askelmuutos säätimen lähtöön (O).



Kuva 41. Askelkokeen tulokset Metso DNA-trendinäytöllä. Kuvataan ohjaussignaalin muutoksen vaikutus mittaukseen.

Askelkokeen tuloksena nähdään Metso DNA Operate trenditiedot ikkunassa ohjaussignaalin muutos ja pienellä viiveellä sitä seuranneen mittaussignaalin muutos. Signaa-

lit asettuvat nopeasti uudelle tasolle eikä suurempaa värähtelyä synny. Signaalien arvot ajanhetkellä 20:00:50 asettumisen jälkeen nähdään hiusviivan kohdalla ikkunassa.



Kuva 42. Toimilaitteen asentoon tehdään vielä kaksi muutosta, joista toisessa lisätään ohjausta ja toisessa pienennetään.

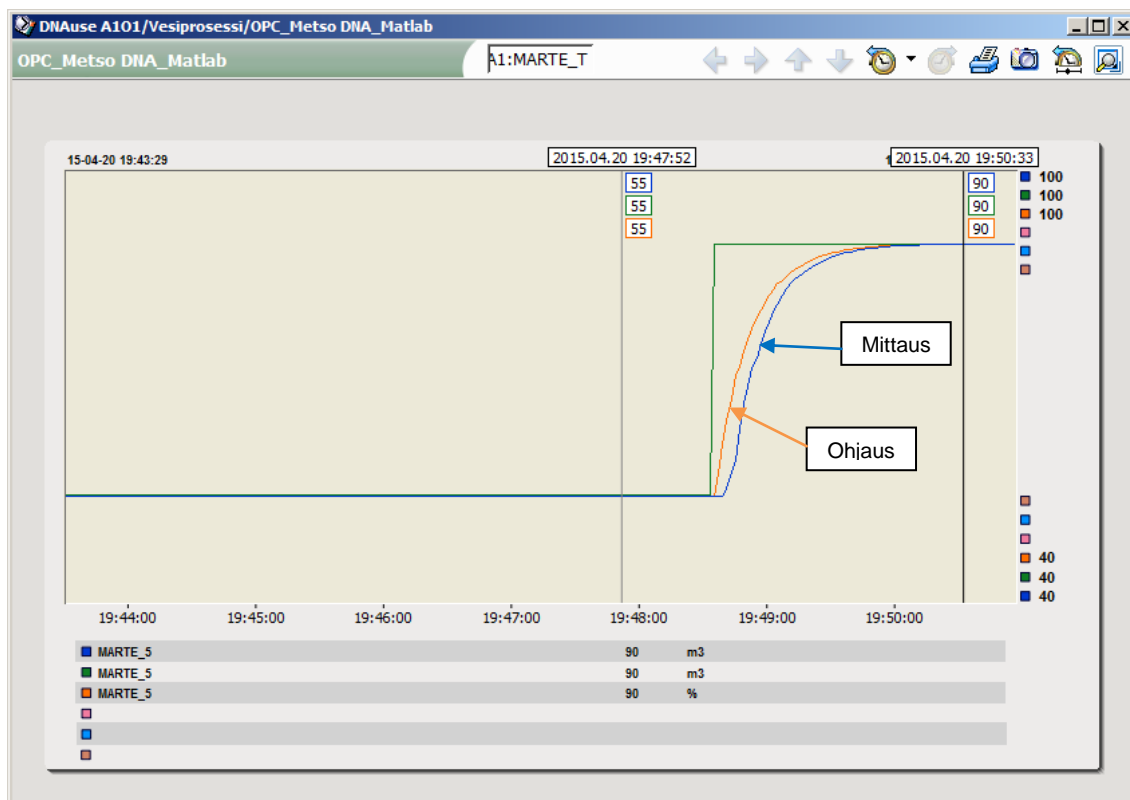
Askelkoe jatkuu ja seuraavaksi tehdään toimilaitteen asentoon vielä kaksi muutosta vastakkaisiin suuntiin (kuva 42). Aluksi lisätään ohjausta 30 - 60 % ja sen jälkeen tiputetaan signaalin arvo 60 - 20 %. Samassa kuvassa esitetään myös säätimen viritysparametrit ($K_p = 5$ ja $T_i = 15$)

Askelkokeen tuloksena nähdään, että mittaus seuraa nopeasti ja tarkasti ilman suurempia signaalipiikkejä ohjaussignaalin muutoksia.

5.1.3 Askelvastekoe 2

Seuraavaksi suoritetaan askelkoe (kuva 43) jossa Metso DNA:n PID-säädin on automaattilla. Säätimen käyttöliittymän kautta muutetaan askelmaisesti säätimen asetusarvoa (sp1) ja kerätään talteen ohjaus (con) sekä mittaussignaali (me). Säätimelle laskettiin parametrit Metson DNA:n automaattisella viritystyökalulla Loop Auto Tuner.

Olkoon mainittu, että Loop Auto Tuner mahdollistaa PID-säätimen eri termien virityksen lisäksi, myös niiden käyttöönoton tai käytöstä poiston. Samalle säätimelle voidaan viritellä parametrit esimerkiksi PI-muodossa tai PID-muodossa (Liite 1). Säätimen tyyppin valinta tehdään Metso DNA:n viritystyökalun käyttöliittymässä ennen virityksen alkua.



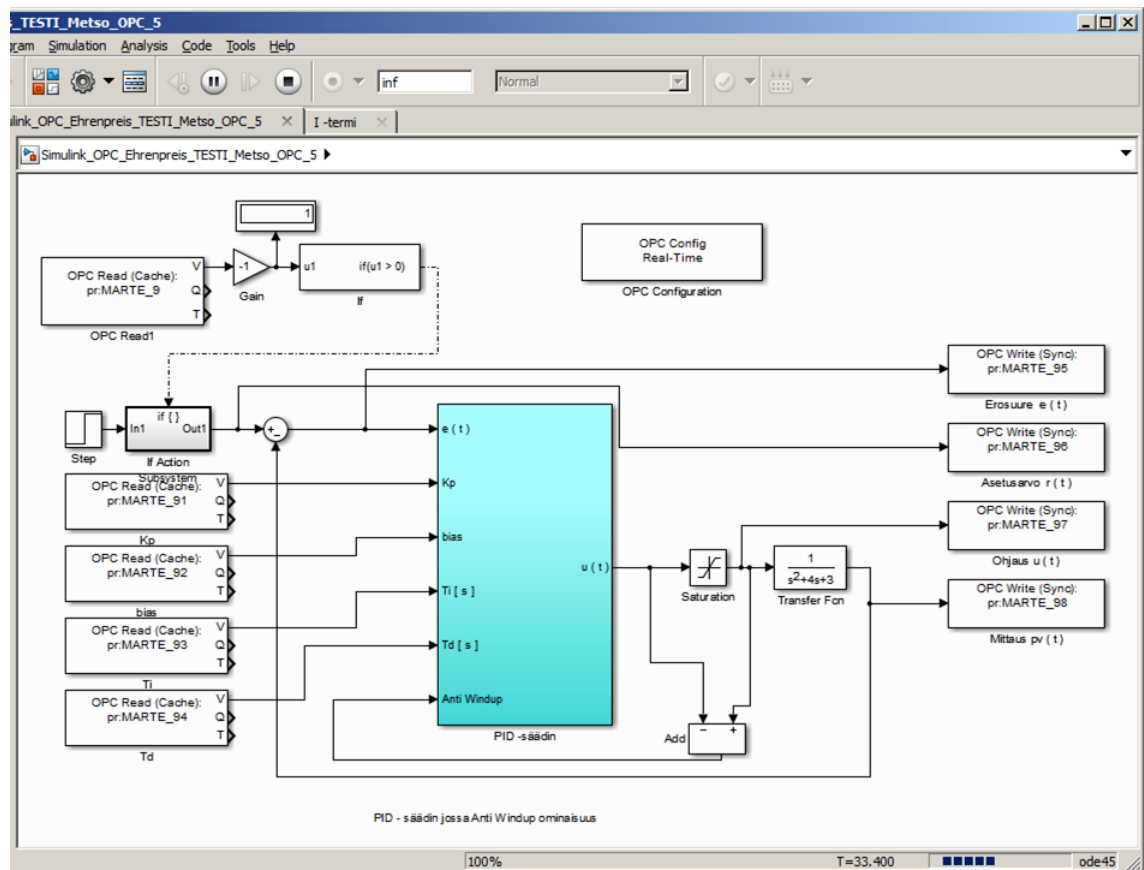
Kuva 43. Askelkokeessa käytetyt PID-säätimen parametrit $K_p=0,47$; $T_i=12,62$; $T_d=1,41$. (Liite 1 s. 2)

Askelvasteesta nähdään, että mittaus saavuttaa asetusarvon ilman suurempia värähteilyjä ja myös ohjaussignaalin muutos tapahtuu rauhallisesti.

5.2 Simulaattori 2 (PID -säädin Matlab Simulink-- mallin yhteydessä)

Simulaattori on rakennettu siten, että prosessimalli varustettuna PID-säätimellä sijaitsee Matlab Simulinkissa, mutta käyttöliittymä puolestaan on Metso DNA-järjestelmässä (kuva 44). Signaalien välityksessä käytetään Metso OPC -palvelinta ja ohjelmarajapinnoissa Matlab Simulinkin OPC-toolbox:in toimilohkoja OPC Config, OPC Write ja OPC Read sekä Metso DNA-automaatiojärjestelmän FbCAD:in suorasaantiporteja.

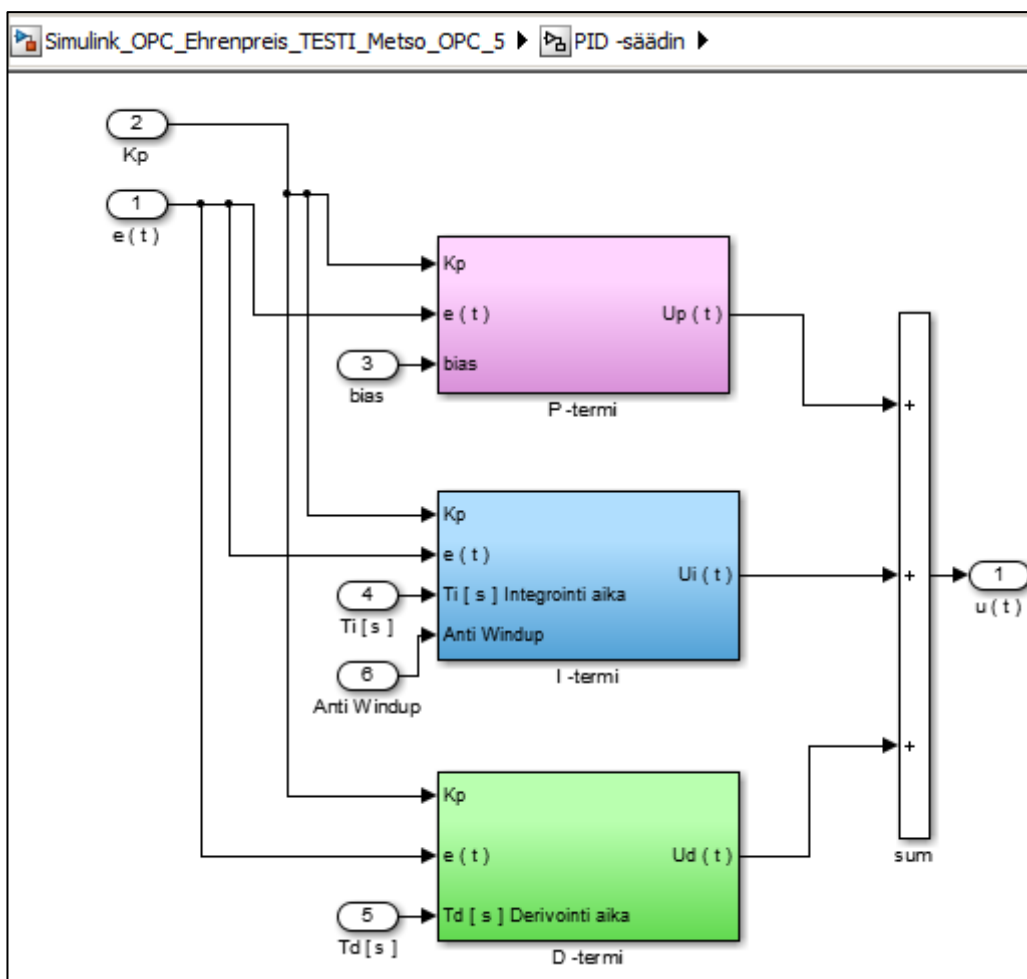
Simulaattorin 1 yhteydessä törmättiin ensimmäisissä simuloinneissa ongelmaan jossa säätö satureitui, eli säätimen integroivan termi kasvatti ohjaussignaalia rajattomasti. Simulaattorin 2 suunnittelussa vältetään ohjaussignaalin mainitulta käyttäytymiseltä, asettamalla Saturation-lohkon avulla ohjaussignaalin alle- ja yläraja (-100 ja +100).



Kuva 44. Simulaattorin Matlab Simulink-osa, jossa PID-säädin merkitsee Metso DNA-automaatiojärjestelmän ulkopuolella sijaittavaa monimutkaisempaa ohjausalgoritmia.

Matlab-simulaattorin vasemmassa yläkulmassa on näkyvissä if-toimilohkoilla toteutettu simuloinnin käynnistysmekanismi. Metso DNA-käyttöliittymässä oleva ON/OFF painike

lähettää OPC Read lohkon välityksellä Simulink-prosessimalliin binäärisignaalin joka avaa tai sulkee simulaattorin portti tulosignaali (1 = ON ja 0 = OFF). Jos Metson käyttöliittymän painike laitetaan ON-asentoon, pääsee Simulink-prosessimallissa signaali if-toimilohkon kautta prosessimallin summaelimeen. Kuten edellisessäkin prosessimallissa, lasketaan summaelimellä asetusarvosta ja mittausviestistä säädintä varten prosessin ero suure $e(t)$. Säätimen tehtävä on ohjata prosessia näin, että ero suure olisi nolla.

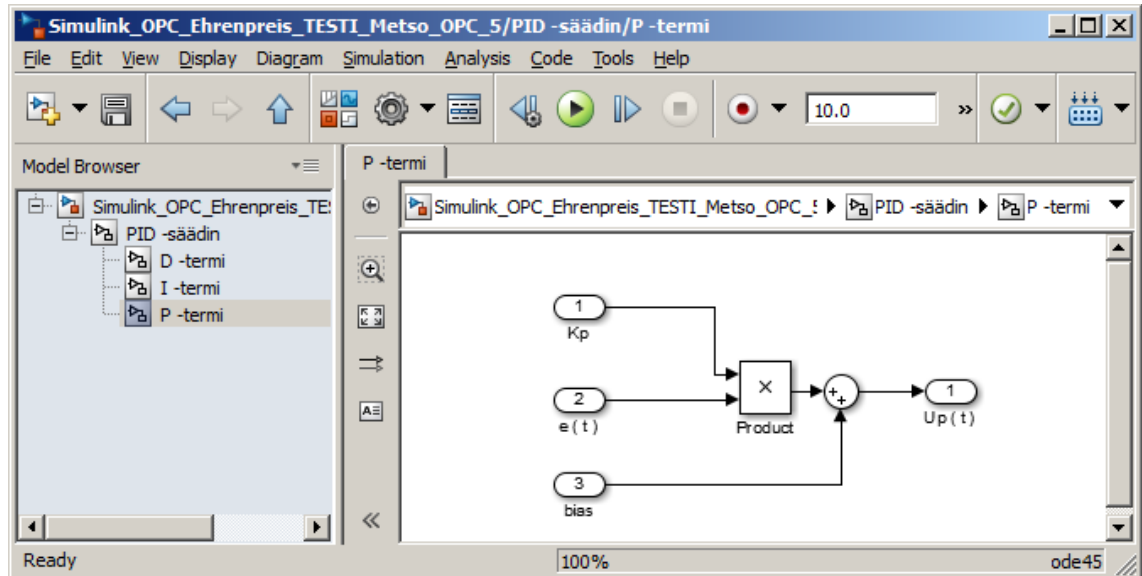


Kuva 45. Simulaattorin 2 prosessimallissa PID-säädin on Simulink-ohjelmassa *subsystem*-lohkona joka koostuu kolmesta alemman tason *subsystem*-lohkosta P-termi, I-termi, ja D-termi.

Matlab Simulink-ohjelma mahdollistaa mallisuunnittelussa alempien kytkentäkokoisuuksien luomisen (*subsystems*). Tämän ominaisuuden ansiosta suunnittelu helpottuu, sillä voidaan tarvittaessa prosessimallin monimutkaisempiakin kokonaisuuksia pilkkoa pienempiin ja yksinkertaisempiin osajärjestelmiin (kuva 45). On mahdollista myös jär-

jestelmän olioiden uudelleen käyttäminen muissa suunnittelukohteissa. Myös alisysteemien väliset kytkennät muotoutuvat helppolukuisemmiksi ja mahdollisuus suunnitteluvirheisiin pienenee.

Myös tämän simulaattorin PID-säätimen eri komponentit (P-, I-, D-termi) ovat jaettu alisysteemeiksi, jolloin niitä voidaan käyttää helposti uudelleen tai tehdä niihin tarvittaessa muutoksia koskematta muihin kytkentöihin (kuvat 46 - 48).

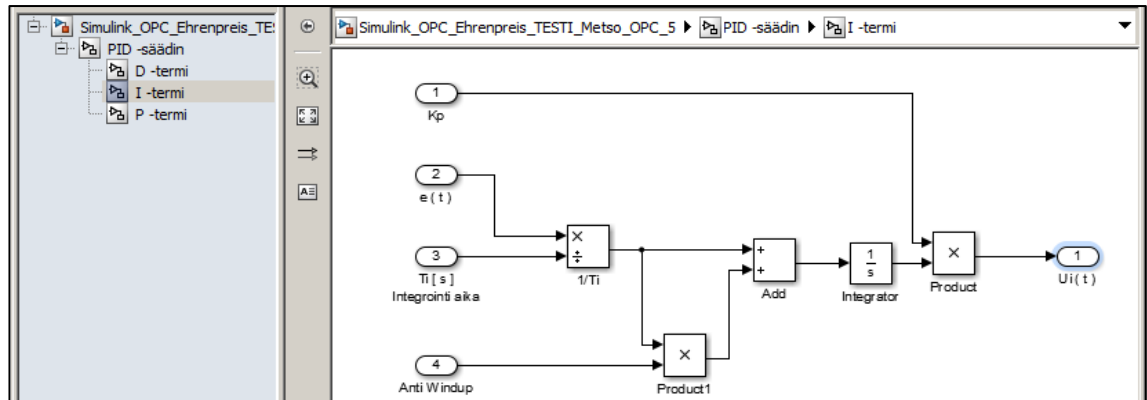


Kuva 46. Alisysteemi joka kuvaa PID-säätimen P-termiä.

PID -säätimen P-termi (kuva 46) on lohkokaaevioesitys kaavasta 1. Alisysteemin lohkokossa tulosaaleina ovat säätimen vahvistus (K_p) ja erosuure ($e(t)$). Usein käytettäessä pelkkää P- säädintä jää säädettävään systeemiin asentovirhe, eli *asetusarvo – mittaus* $\neq 0$. Sitä varten on P-säätimessä kolmas tulo nimeltään bias, johon voidaan tarvittaessa syöttää ohjauksen vakiotaso, joka viritetään prosessin tyypillistä toimintapistettä vastaavaksi. P-säätimen tuottama ohjaussignaali eli säätimen ulostulo $U_p(t)$ summataan PID-säätimessä muiden säätimen termien tuottaman ohjauksen kanssa yhteen summalohkon (sum) avulla. Kuvasta 45 nähdään, että säätimen vahvistus (K_p) ja prosessin eroarvo ($e(t)$) ovat kaikille säädinkomponenteille (P, I, T) yhteisiä, joten ne kytketään PID-lohkossa saman tuloliittimen alle ja jaetaan sitten lohkon sisäisesti jokaiselle komponentille.

Kuvassa 47 alisysteemeistä keskimäinen on PID-säätimen I-termi, jossa on yllä mainittujen lisäksi liittimet integrointiajan ($T_i[s]$) ja PID-säätimen tuottaman ohjaussignaalin

rajoitustason (Anti Windup) syöttämiseen. Ohjaussignaalin rajoitustaso lasketaan rajoitinlohkon (Saturation) tulo- ja lähtösignaalien vertailusta (kuva 44).

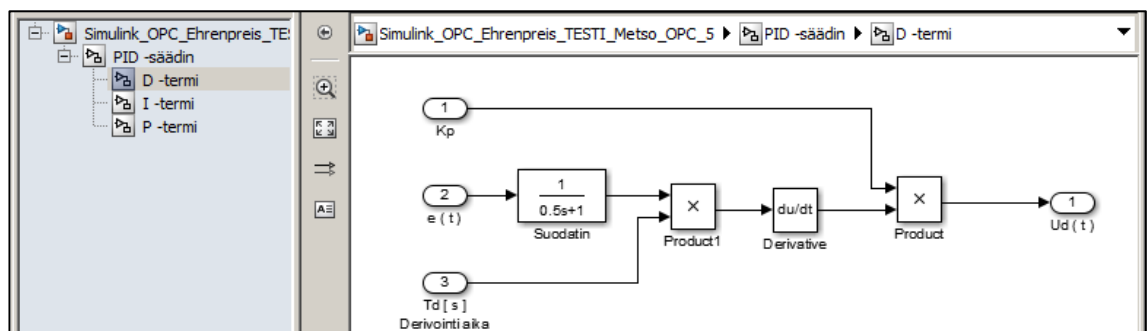


Kuva 47. Alisysteemi joka kuvaa PID-säätimen I-termiä.

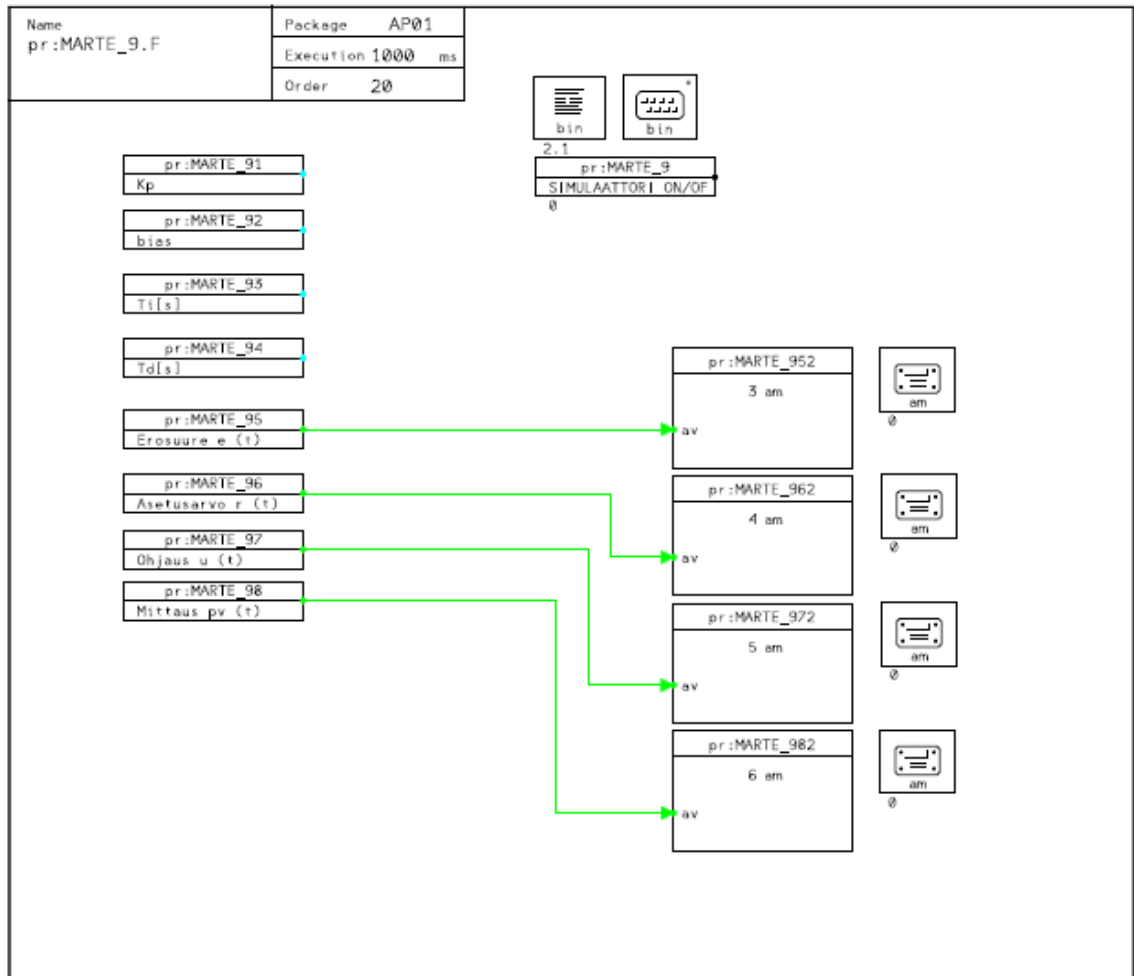
PID-säätimen I-termin rakenne seuraa kaavaa 4 sillä erolla, että säädinkomponenttiin on lisätty Anti Windup -ominaisuus jolla rajoitetaan ohjaussignaalin kasvua.

$$u_i(t) = \frac{K_p}{T_i} \int e(t) dt \quad (4)$$

Säätimen D-termi (kuva 48) eli derivoiva termi on kuvassa 45 alin lohko. Yllä käsiteltävien signaalien lisäksi tähän lohkoon syötetään säätimen derivointi-aika sekunneissa ($T_d[s]$). PID-säätimen derivoivaan komponenttiin kuuluu myös alipäästösuodatin. Sen tehtävä on parantaa säädön laatua leikkaamalla P-osan vahvistusta korkeammilla signaalien taajuuksilla. Ilman alipäästösuodatinta pääsisi signaaliin mahdollisesti summautunut korkeataajuinen kohina vaikuttamaan ohjaussuureeseen eikä säätö enää toimisi-kaan. [26 s.9]



Kuva 48. Alisysteemi joka kuvaa PID -säätimen D-termiä.

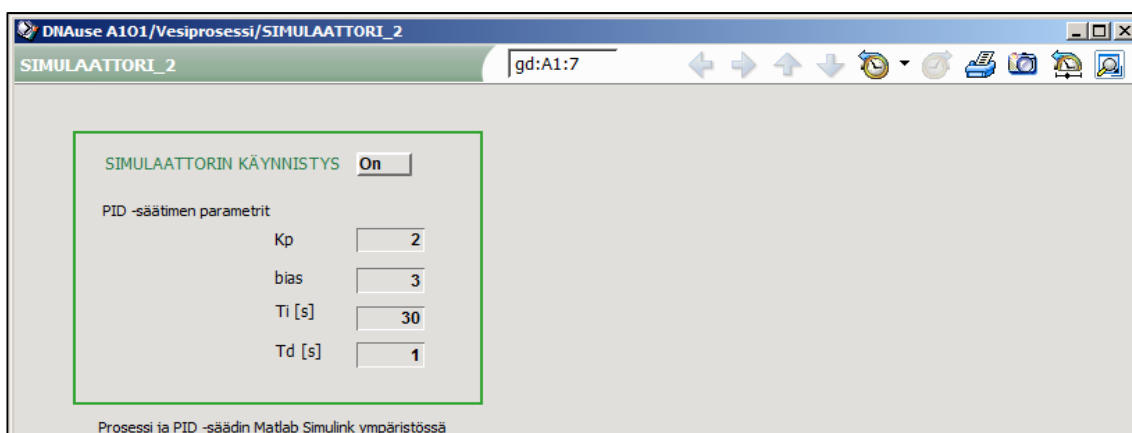


Kuva 49. Metso DNA- FbCAD -ohjelma simulaattorin käyttöliittymän rakentamista varten. Kuvan suorasaantiportit ovat käytössä myös trendien seurannassa. Suorasaantiportteja on käytetty syöttämään signaaleja analogiamittaus toimilohkoille, koska näin saadaan kuvasta lukijalle helpommin ymmärrettävä. Myös analogiamittaus toimilohkon yhteydessä on käytettävissä suorasaantiportti.

Simulaattorin kytkentä Metso automaatiojärjestelmään edellyttää vastaavan FbCAD-ohjelman tekemistä. Ohjelmaan määritellään meitä kiinnostavien suureiden tunnuksia (tag:it) joilla signaalit tunnistetaan ohjelmistojen rajapinnoissa. Kuvassa 49 on simulaattori 2 Metso DNA-kytkentään luotu FbCAD -ohjelma graafisessa muodossa. Siihen kuuluvat suorasaantiportit ja analogiamittauslohkot sekä tarvittavat valvomomoduulit.

Ohjelmakaavion yläosan keskialueella on binääri- signaalin välittämiseen tarkoitettu suorasaantiportti (pr:MARTE_9) joka on kytketty kuvassa 50 olevaan Metso DNAuse käyttöliittymän ON/OFF -painikkeeseen. Painiketta käytetään Matlab Simulink-- simulaattorin käynnistämiseen ja sammuttamiseen. Matlab -simulaattorin IF -toimilohko saa, Metson käyttöliittymän painikkeen asennosta riippuen, joko signaalin arvostaan 1 jolloin

IF -toimilohko päästää tulosignaalin $r(t)$ simulaattoriin tai 0 jolloin tulosignaali katkaistaan.



Kuva 50. Simulaattori 2 käyttöliittymä Metso DNA- järjestelmässä.

Suorasaantiportit pr:MARTE_91 - 94 kuvassa 49 ovat tyyppiä INT ja niiden välityksellä syötetään PID -säätimen konfiguraatioparametrit Metso DNA-use käyttöliittymästä Matlab Simulink-- prosessimalliin. Klikkaamalla käyttöliittymän ikkunassa säätimen parametria, ilmestyy pieni lisäikkuna johon voidaan syöttää säätimen uusi asetusarvo tai ikkunan nuolinäppäimellä muuttaa parametrin nykyistä arvoa.

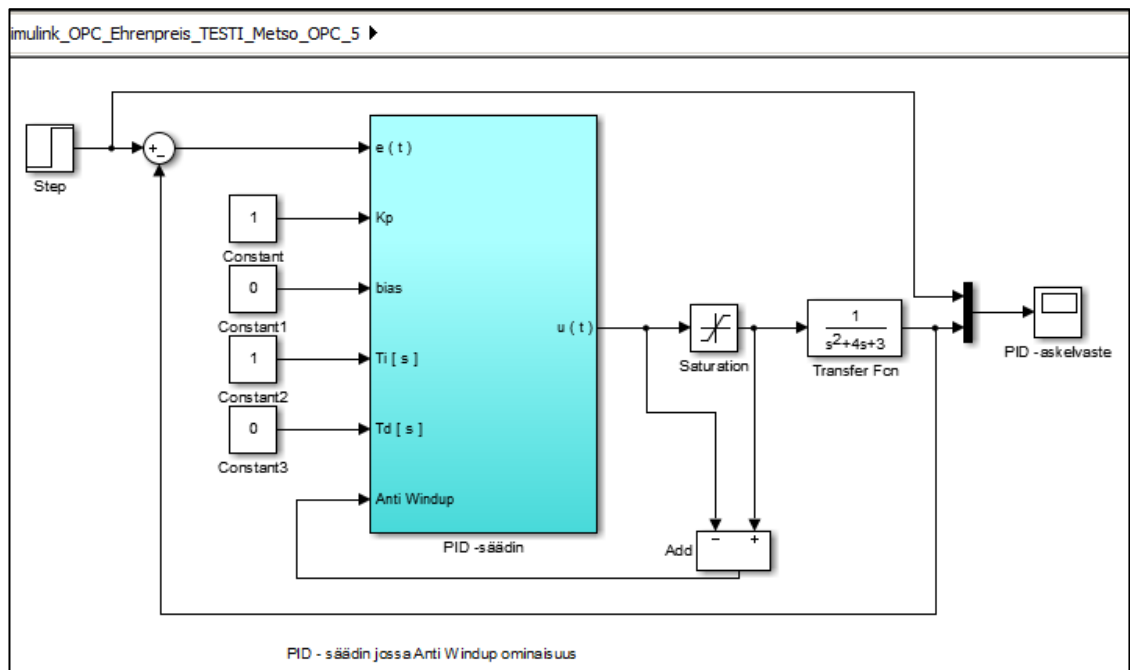
Suorasaantiportit pr:MARTE_95 - 98 kuvassa 49 ovat tyyppiä ANA ja niitä kautta lue-taan Matlab-prosessimallista mitattavat suureet Metso DNA-ympäristöön, jotta ne voitaisiin kuvata Metson trendinäytölle. Vaihtoehtoisesti olisi voitu käyttää mitattavien signaalien lukemiseen pelkkiä analogiasignaalin mittauslohkoihin integroituja suorasaanti-portteja pr:MARTE_951-981, mutta ulkopuoliselle järjestelmän toiminnan ymmärtämi-sen tarkoituksessa käytettiin erillisiä suorasaantiportteja.

Simulaattorin 2 käyttöliittymän suunnittelussa tuli testattu myös Metso DNA- FbCAD-ohjelmassa sijaitsevan PID-säätimen hyödyntäminen ns. käyttöliittymämoodissa. Säädin saadaan siihen tilaan asettamalla sen parametreissa cmode arvoksi 128. Säädin ei tässä tilassa yritä säätää prosessia vaan esimerkiksi toimii vaihtoehtoisena käyttöliittymänä sen kautta kulkeville signaaleille. Säätimen tällainen käyttömahdollisuus on kehitetty tilanteisiin joissa varsinainen ohjausalgoritmi sijaitsee Metso FbCAD:in ulkopuolella (esim. Matlab Simulinkissa), mutta dokumentaatioissa tai raportoinnissa on silti tarvetta esittää piirissä PID-säätimen olemassaolo myös Metso DNA-ympäristössä. [19 s.450]

Tätä HMI vaihtoehtoa ei ole työssä sen tarkemmin käsitelty koska se ei tarjonnut mahdollisuutta syöttää Matlab Simulink-ympäristöön PID-säätimelle kaikkia tarvittavia konfigurointiparametreja.

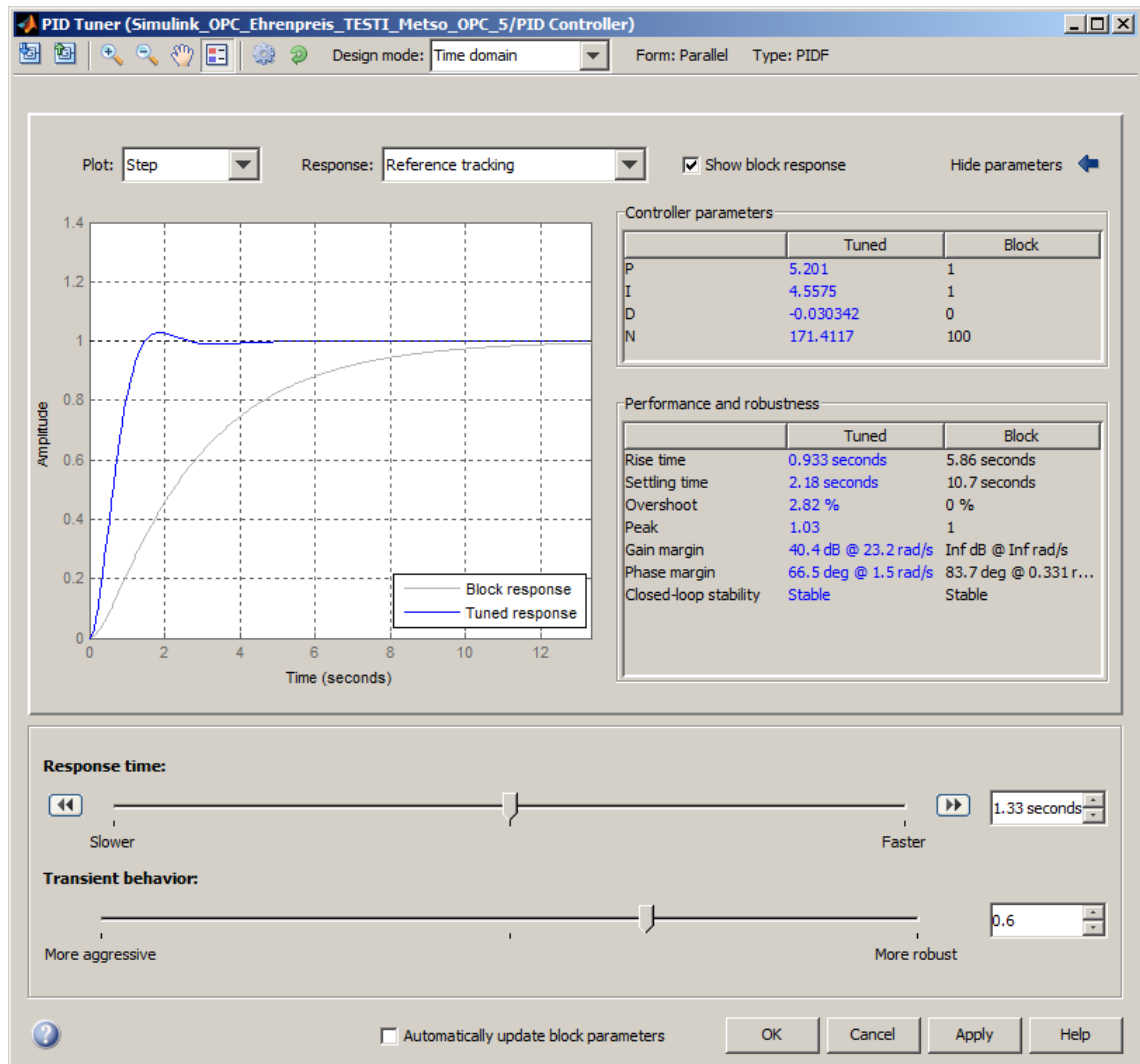
5.2.1 Säädön automaattinen viritys (simulaattori 2)

Rakennetun säätimen varsinainen automaattinen virittäminen ei PID-säätimen viritystyökalulla PID Tuner onnistu, mutta viritetään saman prosessin ohjaukseen Matlab Simulinkin vakio PID -säätimen toimilohko ja käytetään saatuja parametreja rakennetun säätimen konfiguroinnissa. Syötetään samanlaiset viritysparametrit molempiin säätimiin ja suoritetaan askelkoe muuttamalla asetusarvo nolasta yhteen.



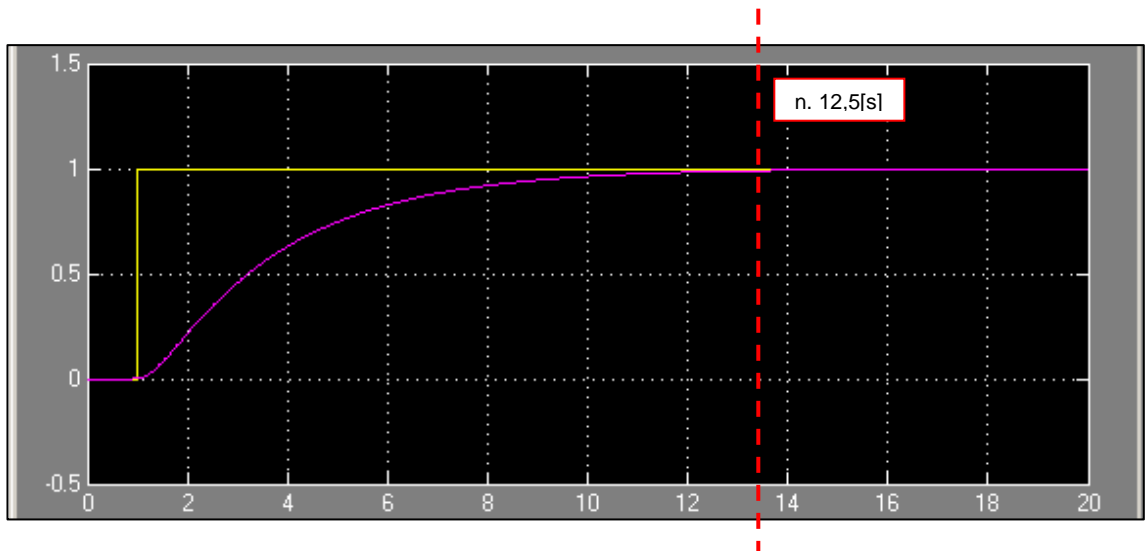
Kuva 51. PID -säätimen Matlab Simulink-mallin testaus ($K_p=1$, $bias=0$, $T_i(t)=1$, $T_d(t)=0$).

Säädinkokeessa syötetään viritysparametrit säätimen liittimiin käyttäen Constant-toimilohkoja. Prosessista mitattavat asetusarvo $r(t)$ ja prosessin lähtö $p_v(t)$ kuvataan graafisesti Scope toimilohkon näytölle. Testauksien onnistuttua, voidaan ottaa säädin simulaattorikäyttöön eli kytkeä sen liittimiin Matlab OPC Toolbox:in signaalivälitys toimilohkot OPC -yhteyttä varten (kuva 45).



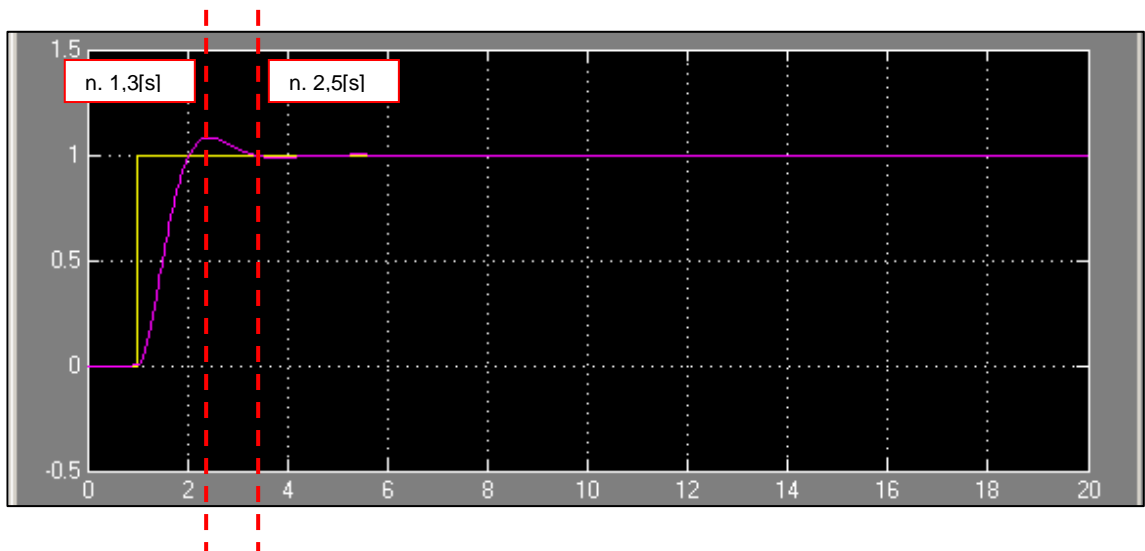
Kuva 52. Prosessin ohjaukseen käytettiin testimielessä Matlab'in PID -säädinlohkoa johon haettiin ohjelman omalla viritystyökalulla optimaaliset parametrit.

PID Tuner ikkunassa (kuva 52) nähdään automaattisen virityksen parametrit (Controller parameters Tuned) ja alkuperäiset toimilohkon parametrit (Controller parameters Block). Nähdään myös molempia viritysparametreja vastaavat askelvastekäyrät. Työssä rakennetun PID-säätimen kohdalla ei ole mahdollista käyttää alipäästösuodattimen parametria N (Filter Coefficient), koska alipäästösuodatin on toteutettu siirtofunktion loholla [27]. Kun käytettiin Matlab PID Tuner'in ehdottamia viritysparametreja simulaattorissa 2 rakennettuun PID-säätimeen, todettiin kumminkin täydentävän virityksen olevan tarpeen. Käytettiin yritys-erehdys viritysmenetelmää [8 s.114] ja haettiin simulaattorin 2 säätimelle uudet parametrit, joilla päästiin lähes samaan viritystulokseen kun Matlab Simulink'in valmiilla PID-säätimellä (kuva 52).



Kuva 53. Simulaattori 2 rakennetun PID -säätimen askelvaste ($K_p=1$, $\text{bias}=0$, $T_i=1$, $T_d=0$)

Simulaattorin säätimen ensimmäisessä kokeessa päästiin kuvan 53 mukaiseen askelvasteeseen. Säättö toimii eikä prosessisuureeseen jää pysyvää poikkeamaa eli säätimen bias-termin käyttöön ei ole tarvetta.



Kuva 54. Simulaattori 2 rakennetun PID -säätimen askelvaste ($K_p=7$, $\text{bias}=0$, $T_i=1.1$, $T_d=0.01$)

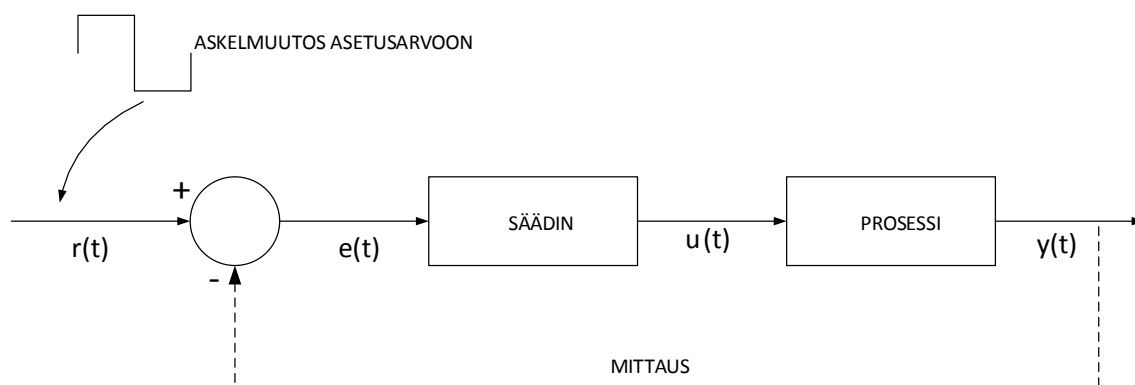
Kuvassa 53 prosessin ulostulokäyrän nousuaika ja asettumisen aika ovat likimain samoja (noin 12,5[s]). Tarvitaan siis täydentävää viritystä saadakseen säädön toimimaan tehokkaammin. Esimerkiksi Matlab:in omalla vakio PID -toimilohkolla (kuva 52) sama prosessi asettui 2,18 [s] kuluttua asetusravon $r(t)$ askelmuutoksesta ja prosessin ulostulon nousuajaksi mitattiin 0,933 [s]. Simulaattorin 2 virityksessä saadut säätöparametrit antoivat kohtuullisen hyvän askelvastetuloksen, jossa prosessisuureen $p_v(t)$ nousu-

ajaksi mitattiin noin 1,3 [s] ja prosessi asettuu uudelle tasolle noin 2,5 [s] kuluttua asetusarvon muutoksesta.

Tällaisenaan Simulaattori 2 rakennettu PID-säädin voidaan ottaa simulointikäyttöön ja kytkeä Matlab Simulink-prosessimalliin sekä OPC -palvelimen kautta Metso DNA- -automaatiojärjestelmään.

5.2.2 Askelvastekoe 3

Mikäli askelkoe tehdään säätimen ollessa automaattilla, on askel tehtävä asetusarvoon. Tätä ei mallintamisen teoriassa kuitenkaan suositella, sillä malli on aina rakennettava säätimen ohjaussignaalin ja prosessisuuren mittaussignaalin välille. Isommissa säätöpiireissä on vaara, että asetusarvoon tehty askel menettää askelmaisen muodon sillä matkalla. [8 s. 90]

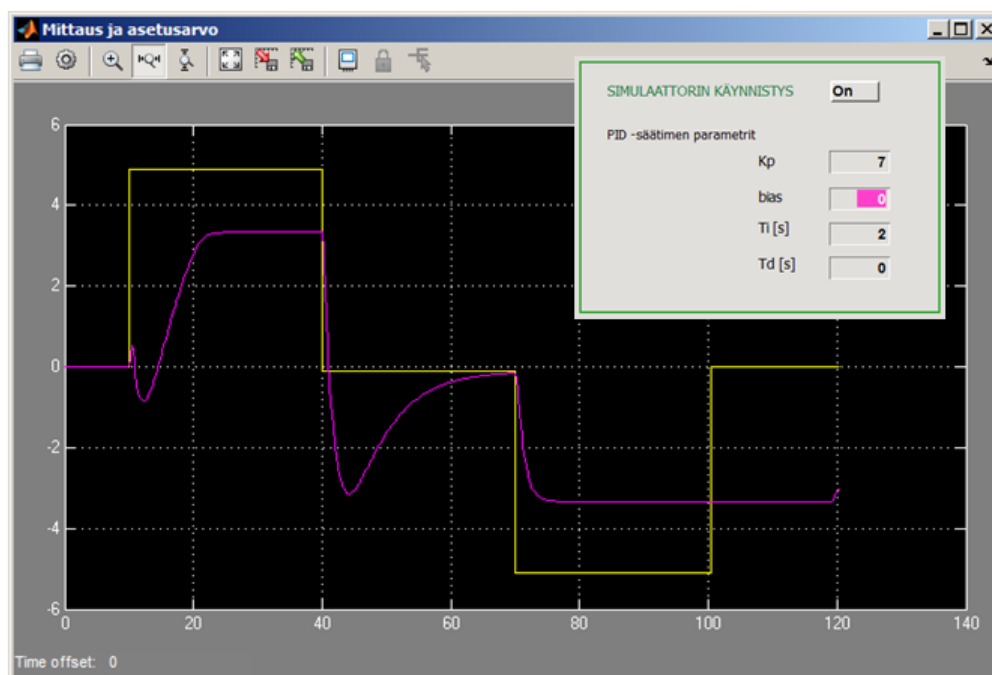


Kuva 55. Askelvastekoe jossa askelmainen muutos tehdään prosessimallin asetusarvoon $r(t)$. [8 s.90]

Testataan nyt kuitenkin myös tämä mallinnuksen vaihtoehto ja otetaan huomioon riski, ettei tämä laajahkoissa säätöpiireissä välttämättä ole paras tapa säädön viritykseen.

Metso DNA-Operate käyttöliittymän kautta syötettiin PID-säätimelle kuvan 56 mukaiset parametrit. Prosessimallin (kuva 44) tuloliittimeen In1 kytketään Signal Builder toimilohko ja käynnistetään simulointi. Kuten huomataan, prosessin ulostulo ei saavuta asetusarvoa ja jättää säätötulokseen pysyvän poikkeaman. Prosessi asettuu uudelle johon se jää rauhallisesti pysymään kunnes suoritetaan seuraavaan asetusarvomuuutos. Asetusarvon muutoshetkellä nähdään mittauksessa piikki. Säätimen Anti Wind-Up ominai-

suus rajoittaa ohjaussignaalia asetusarvon kasvaessaan, mutta asetusarvon pienentyessä tapahtuukin prosessisuureessa paljon jyrkempi muutos, eli ohjaussignaalin rajoitus on epäsymmetrinen. Säädettävään systeemiin jäänyttä asentovirhettä kokeiltiin poistaa myös säätimen *bias*-toiminnon avulla, mutta sillä ei ollut ääriarvoissa vaikutusta prosessisuureeseen.

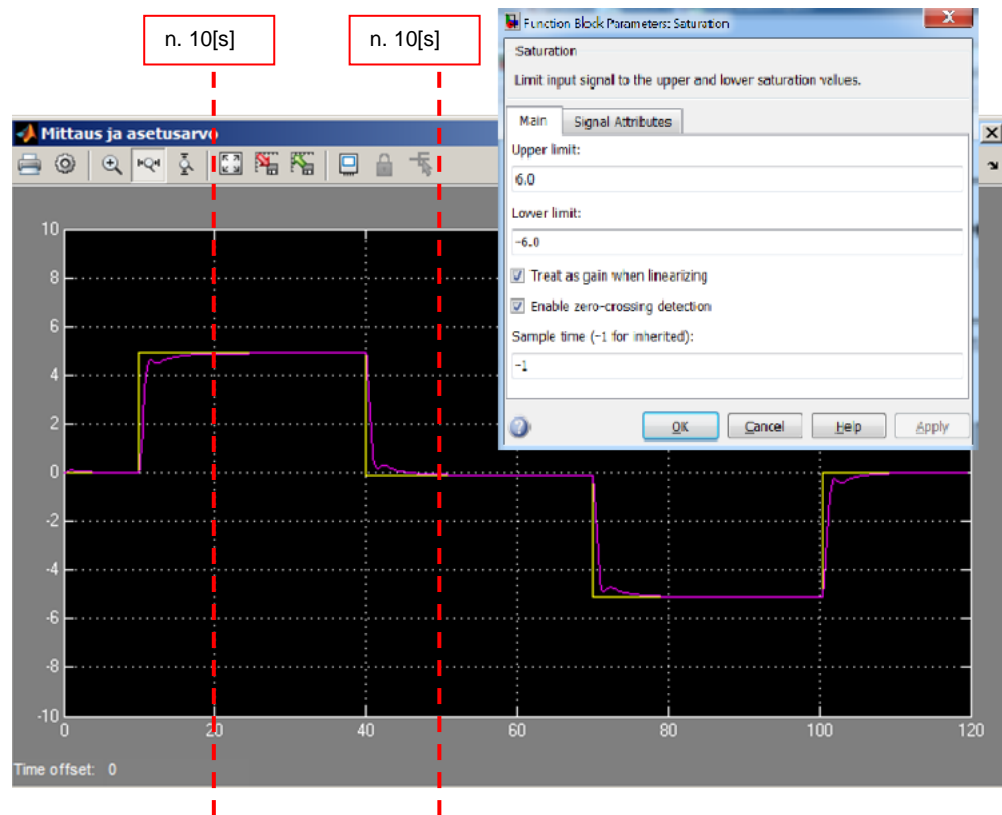


Kuva 56. Askelkoe jossa tutkitaan asetusarvomuutoksen ($\Delta r(t)$) vaikutuksia prosessisuureeseen $p_v(t)$. Saturation-toimilohko rajoittaa PID-säätimestä tulevaa ohjaussignaalia ja siitä johtuen myös prosessiin jää pysyvä säätövirhe.

Säätöpoikkeaman syyksi osoittautui Saturation-toimilohkon (kuva 44) asetusten virheellinen määrittely. Lohkon tehtävä on rajoittaa sen läpi kulkeva signaali asetetun ylä- ja alarajan mukaisesti. Määriteltiin Saturation-lohkolle kuvan 57 mukaiset asetukset, eli ylärajaksi 6.0 ja alarajaksi -6.0 eli säätimeltä tuleva ohjaussignaali pääsee näissä rajoissa lohkon läpi. Kuten seuraavassa kuvassa nähdään, tämän muutoksen jälkeen ohjauksen laatu parani ja säätövirhe poistui kokonaan.

Saturation-lohkon käyttö edellyttää säädettävän prosessin tuntemusta sen verran, että pystytään arvioimaan ohjaussignaalin tarpeellinen vaihteluväli. Jos tunnetaan prosessin normaali toimintapiste, voidaan simulointien avulla selvittää, kuinka voimakas ohjaussignaali prosessin hallintaan tarvitaan ja rajata se Saturation-lohkon avulla siihen väliin.

Vaikka prosessiin ei asetusarvomuutoksien jälkeen jää säätöpoikkeamaa, on säätö silti suhteellisen hidasta.

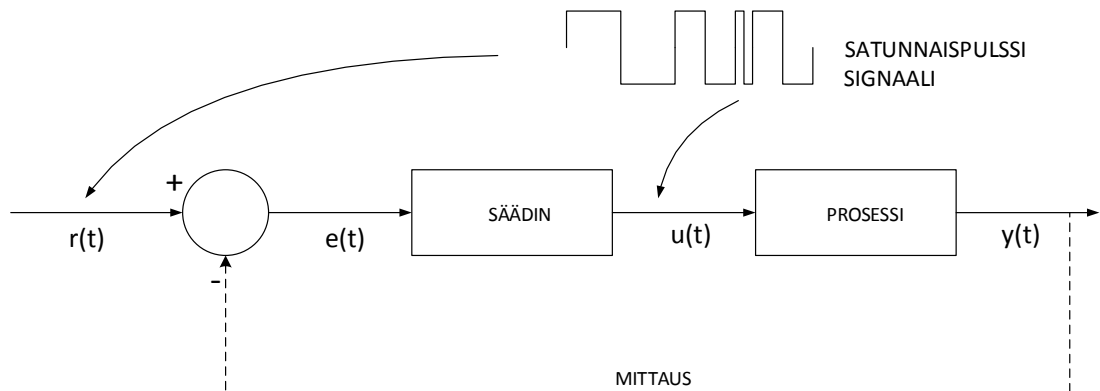


Kuva 57. Saturation lohkon sopivilla parametreilla saavutettiin laadukas askelvaste.

Prosessi asettuu uudelle tasolle, kun asetusarvomuutoksesta on kulunut n. 10[s]. Tämä ei välttämättä ole aina huono asia, sillä on prosesseja jossa nimenomaan tarvitaan rauhallista säätöä. Riippuen käyttökohteesta, voidaan silti tarvittaessa kasvattaa säätönopeutta suurentamalla P -osan vahvistusta (K_p) ja pienentämällä I -osan integrointiaikaa (T_i).

5.2.3 Askelvastekoe 4

Satunnaispulssikoetta (kuva 58) käytetään yleensä nopeiden säätöpiirien virittämiseen, kuten esimerkiksi virtauksen tai paineen säätö. Menetelmä antaa myös tilastollisesti luotettavamman tuloksen, koska pulsseja on montaa peräkkäin. Tämä viritystapa soveltuu, suhteellisen pieneen pulssiampplitudin ansiosta, erinomaisesti käytännön prosessikokeisiin, sillä se ei häiritse merkittävästi tutkittavaa prosessia.



Kuva 58. Askelvastekoe jossa satunnaispulssisignaali syötetään joko toimilaitteen asentoon tai prosessin asetusarvoon [8 s. 90-92, 97].

Satunnaispulssikokeen haittapuolena on sen kesto. Sen hyödyt saadaan käyttöön vasta silloin, kun on kerätty minimissään 500 - 1000 näytettä. [8 s. 90-92, 97] Aikarajoitteiden takia ei tätä koetta lähetä kumminkaan suorittamaan. Prosessikokeista paremman kuvan luomisen tarkoituksessa, mainittakoon sellaisen vaihtoehdoisen menetelmän olemassaolo.

Yhteenveto

Työssä kehitettiin simulointiympäristö jossa Matlab Simulink-ohjelmassa sijaitsevaa prosessimallia ohjataan Metso DNA-automaatiojärjestelmällä. Signaalien reaaliaikainen välitys prosessimallin ja automaatiojärjestelmän välissä tapahtuu Metso DNA OPC-palvelimen kautta.

Päästiin työn alussa asetettuihin tavoitteisiin, eli rakennettiin kaksi erilaista prosessimulaattoria. Toisessa sijaitsee PID -säädin Metso DNA- automaatiojärjestelmässä, mutta toisessa on ohjausalgoritmi ulkopuolisessa simulointiohjelmassa Matlab Simulink. Molempien simulaattorin käyttöliittymä on toteutettu Metso DNAuse-ohjelmistolla ja signaalien trendien keruu hoidetaan Metso DNA-ympäristössä.

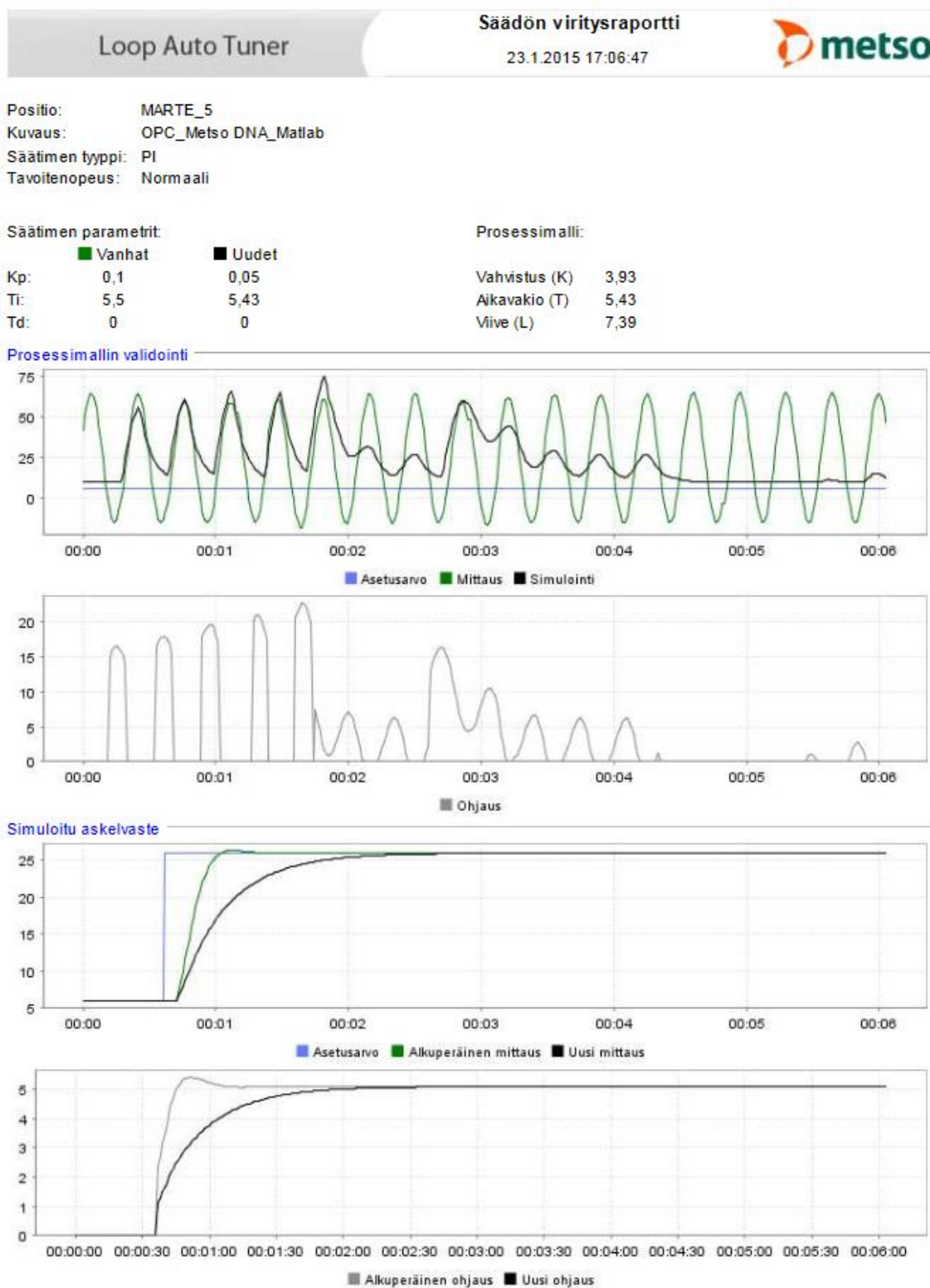
Työn puitteissa käytiin kattavasti läpi myös asiaan liittyvä teoreettinen tausta, jotta lukijalle muodostuisi kuva miten ja mihin tarkoitukseen simulointiympäristö on kehitetty.

Lähteet

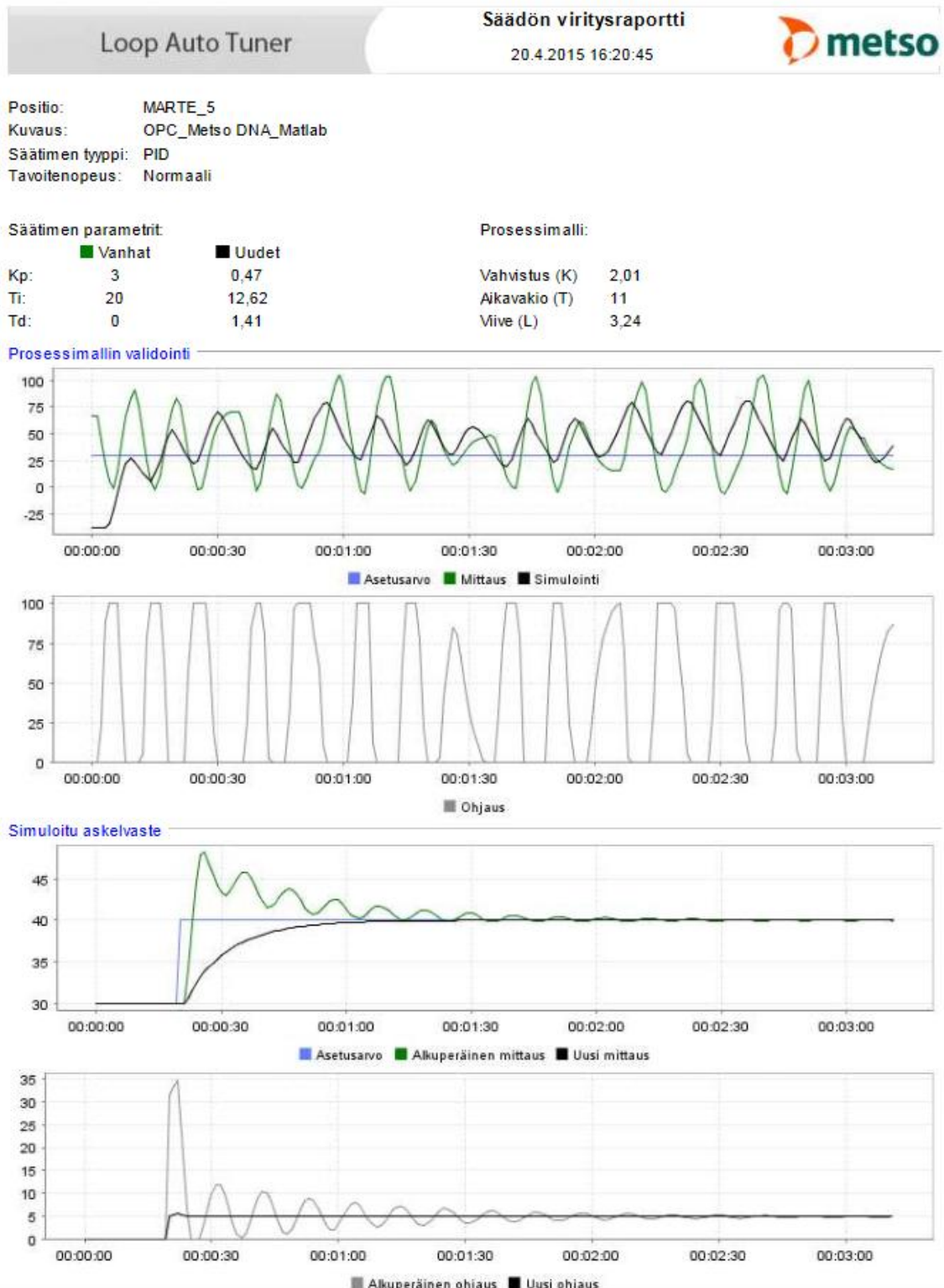
- 1 Jürgen Lange, Frank Iwanitz, Thomas J. Burke. 2010. OPC From Data Access to Unified Architecture. 4th rev. Ed. VDE VERLAG GMBH, Berlin.
- 2 OPC Foundation verkkosivusto. History. <<https://opcfoundation.org/about/opcfoundation/history/>> Luettu 7.12.2014.
- 3 OPC Data Access Custom Interface specification Version 3.00; Opc Foundation. March 4, 2003
- 4 OPC Data eXchange Specification Version RC 1.10; Opc Foundation. January 23, 2003
- 5 OPC Historical Data Access Specification Version 1.20; Opc Foundation. December 10, 2003
- 6 OPC Batch Custom Interface Specification Version 2.00; Opc Foundation. July 19, 2001
- 7 Marcin Bajer. 2008. Master's Thesis. Control Systems Integration using OPC Standard. AGH University of Science and Technology, Antwerp.
- 8 Timo Harju, Arto Marttinen. 2000. Säättöpiirin virityksen perusteet. 1. painos. Paino. Otamedia Oy, Espoo.
- 9 OPCconnect.com verkkosivusto. OPC Commands. <<http://www.opcconnect.com/commands-intro.php>> Luettu 24.2.2015.
- 10 OPC Complex Data Specification Version 1.00; Opc Foundation. December 10, 2003
- 11 Wolfgang Mahnke, Stefan-Helmut Leitner, Matthias Damm. 2009. OPC Unified Architecture. Ed. Springer-Verlag, Berlin.
- 12 automationworld.com verkkosivusto. New automation concepts with OPC Unified Architecture. <<http://www.automationworld.com/new-automation-concepts-opc-unified-architecture>> Luettu 2.3.2015.
- 13 Metso DNA- Engineering DNA Explorer -käyttöohje, DNA Explorer versio 14.1. Rev 3/2011, Metso Automation Oy, Tampere
- 14 Metso DNA- Engineering Function Block CAD -käyttöohje, Collection 2011 rev. 5. G2010_FI_05, Metso Automation Oy, Tampere

- 15 [www.metso.com verkkosivusto. Metso DNA- Engineering.<http://www.metso.com/Automation/ip_prod.nsf/WebWID/WTB-110922-2256F-1FC2A/\\$File/E8730_EN_04-DNA%20Engineering%20environment.pdf>](http://www.metso.com/Automation/ip_prod.nsf/WebWID/WTB-110922-2256F-1FC2A/$File/E8730_EN_04-DNA%20Engineering%20environment.pdf) Luettu 16.3.2015.
- 16 [www.metso.com verkkosivusto. Metso DNA- Operate.<http://www.metso.com/Automation/ip_prod.nsf/WebWID/WTB-111003-2256F-8F0EE/\\$File/E8383_EN_03-DNA%20Operate.pdf>](http://www.metso.com/Automation/ip_prod.nsf/WebWID/WTB-111003-2256F-8F0EE/$File/E8383_EN_03-DNA%20Operate.pdf) Luettu 22.3.2015.
- 17 [www.metso.com verkkosivusto. Metso DNA- Operate TEA.<http://www.metso.com/automation/docs2.nsf/0/0FE87D41894A4759C2257790002BEC47/\\$file/E8767_EN_2-DNA%20OperateTEA.pdf>](http://www.metso.com/automation/docs2.nsf/0/0FE87D41894A4759C2257790002BEC47/$file/E8767_EN_2-DNA%20OperateTEA.pdf) Luettu 22.3.2015.
- 18 [www.aalto.fi verkkosivusto <https://noppa.aalto.fi/noppa/kurssi/as-74.2111/luennot/AS-74_2111_luento_10.pdf>](https://noppa.aalto.fi/noppa/kurssi/as-74.2111/luennot/AS-74_2111_luento_10.pdf) Luettu 30.3.2015.
- 19 pid - SÄÄDINTOIMILOHKO. Metso DNA- manuaalin osio 47. Metso Automation Oy, Tampere
- 20 [http://library.kiwix.org verkkosivusto. Laskostuminen <http://library.kiwix.org/wikipedia_fi_all/A/Laskostuminen.html>](http://library.kiwix.org/wikipedia_fi_all/A/Laskostuminen.html) Luettu 30.3.2015.
- 21 metsoDNA CR Automaatiokieli, Collection 2010 rev. 4 G2047_FI_04, Metso Automation Oy, Tampere
- 22 Metso DNA- Engineering Trendien suunnittelu -käyttöohje, Collection 2011 rev. 3. G2115_FI_03-I, Metso Automation Oy, Tampere
- 23 Jussi Oikari, Automation Engineer, System Specialist - Metso Automation. Konsultointi 7.1.2015
- 24 [www.matrikonopc.com verkkosivusto. <https://www.matrikonopc.com/products/opc-drivers/opc-simulation-server.aspx>](https://www.matrikonopc.com/products/opc-drivers/opc-simulation-server.aspx) Luettu 19.9.2014.
- 25 [http://se.mathworks.com verkkosivusto <http://se.mathworks.com/help/opc/ug/reading-and-writing-data.html#f6-6077>](http://se.mathworks.com/help/opc/ug/reading-and-writing-data.html#f6-6077) Luettu 5.4.2015.
- 26 [https://noppa.aalto.fi verkkosivusto< https://noppa.aalto.fi/noppa/kurssi/as-0.2230/materiaali/AS-0_2230_tyo_11__tyoohje_2.pdf>](https://noppa.aalto.fi/noppa/kurssi/as-0.2230/materiaali/AS-0_2230_tyo_11__tyoohje_2.pdf) Luettu 10.4.2015.
- 27 [http://www.mathworks.com verkkosivusto <http://www.mathworks.com/matlabcentral/fileexchange/23965-anti-windup-pid-controller-Simulink---block >](http://www.mathworks.com/matlabcentral/fileexchange/23965-anti-windup-pid-controller-Simulink---block) Luettu 10.4.2015.

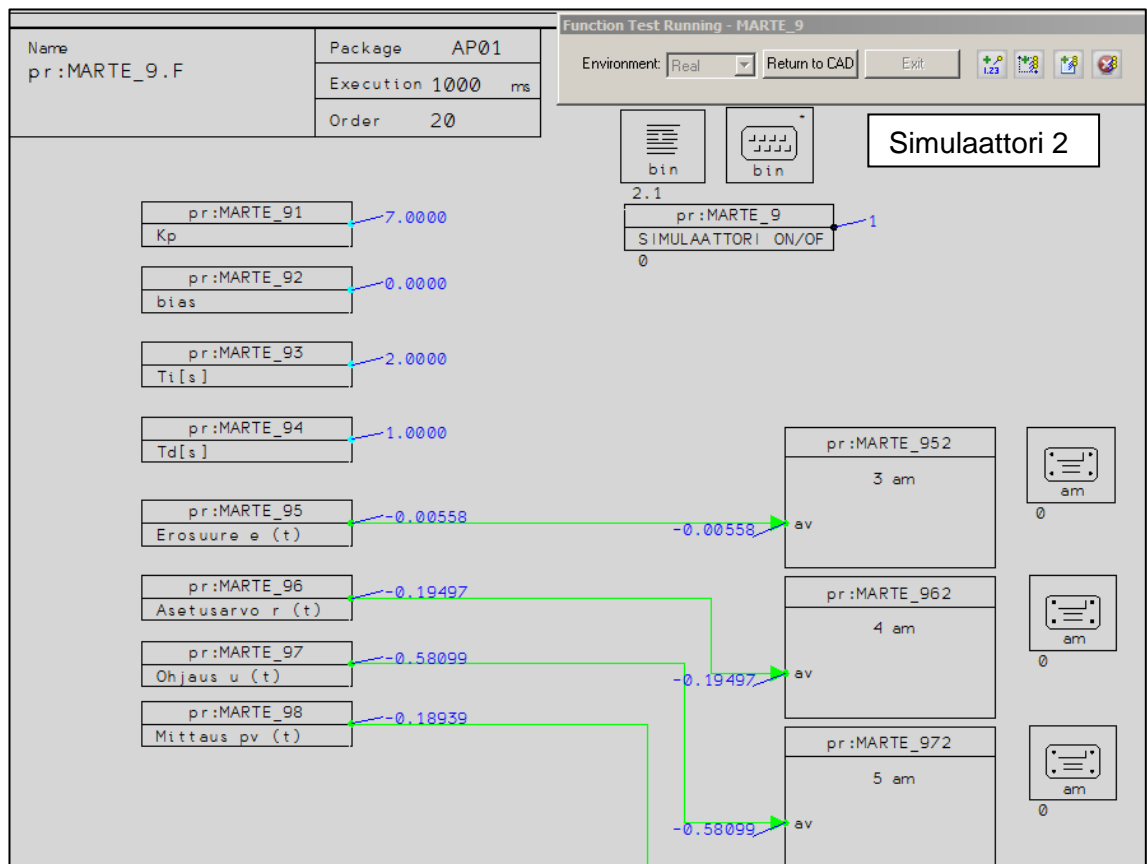
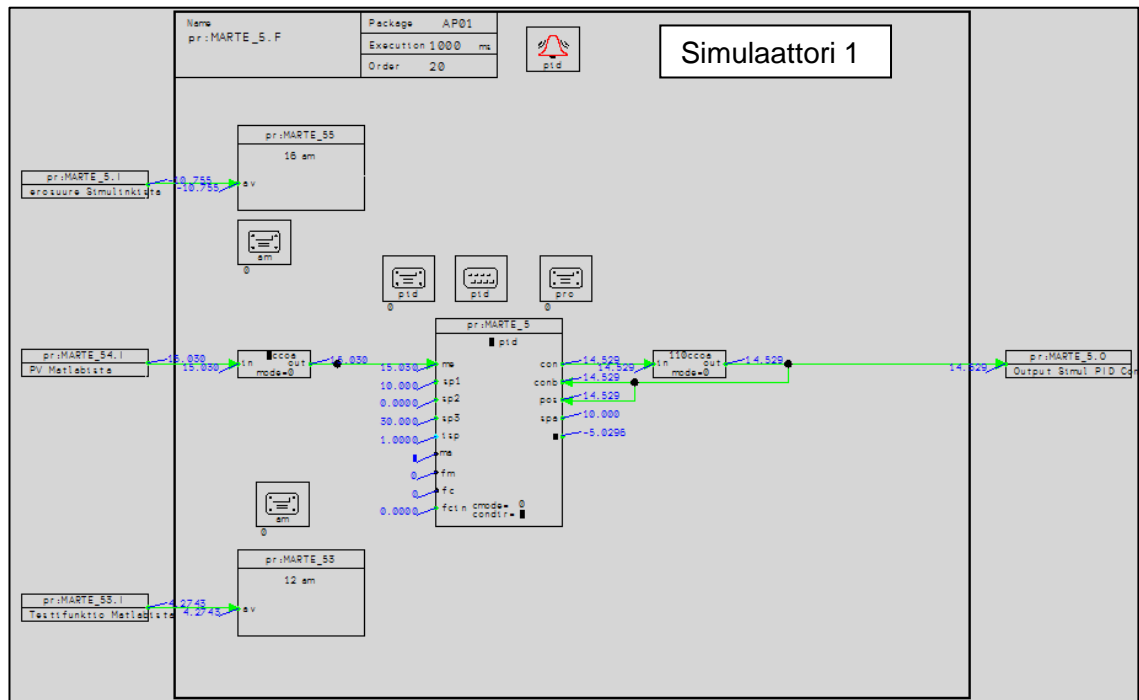
DNA Engineering Loop Auto Tuner avulla lasketut säätimen viritysarvot (PI -säädin)






Engineering Loop Auto Tuner avulla lasketut säätimen viritysarvot (PID -säädin)



FbCAD graafinen ohjelma simulaattorien 1 ja 2 kytkennöistä testaus tilassa.



Laiteluettelo ACN -kaappi

Nro.	Laite	Valmistaja	Mallinimi	Mallinumero	Kuva	Lisää tietoa
1	Switch älykäs verkkokytin	Cisco Systems	Cisco Catalyst 2950 Series	WS-C2950G-24-EI	 Cisco Catalyst 2950 24 Switch	Kotelossa lisäksi merkintä: E-E011-02-1257(A)
2	ACN CS prosessinohjausyksikkö	Metso		D201925 ACN CS s/n: CEF1205001 Ver 1.3		Tarra 1 AP01 Tarra 2 AP01 IP:10.83.86.177 HW: 161 Token: 11
3	Ethernet to Fieldbus Converter Ethernet Fieldbuss muunnin	Metso	EFC			
4	ACN I/O ja virransyöttö ym.	Metso	MBM80 Samassa modulissa: IPSP IBC I/O kortit: AI8H 2kpl AO4H 2kpl DI8P 8kpl			Tarra: AP01 FBC 3
5	I/O räkki	VALMET	VALMET IPU I/O kortit: BIU-82 1kpl BIU-84 2kpl BOU-8 2kpl BOU-82 1kpl AIU-8 2kpl AIU-8 SIMUL 2kpl AOU-4 2kpl AOU-4 SIMUL 1kpl TIU-6 1kpl BIC 1kpl			Tarra: AP01 FBC 2
6	Serveri		EAS METSO DNA			Tarra 1: EAS METSODNA IP: 10.83.86.176 MC: 239.0.171.0/A HW: 101 Token: 8 Tarra 2: A1A1 / AB01 IP: - MC: - HW: - Token: -