

Prosessisimulaattorin suunnittelu ja toteutus Valmet DNA -ympäristöön

Juuso Välimaa

Opinnäytetyö
Lokakuu 2020
Tekniikan ala
Insinööri (AMK), automaatiotekniikka

Tekijä(t) Välimaa, Juuso	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä lokakuu 2020
	Sivumäärä 58	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Prosessisimulaattorin suunnittelu ja toteutus Valmet DNA-ympäristöön		
Tutkinto-ohjelma Automaatiotekniikka, JAMK		
Työn ohjaaja(t) Ari Kuisma, Teppo Flyktman		
Toimeksiantaja(t) Jyväskylän ammattikorkeakoulu		
<p>Tiivistelmä</p> <p>Tavoitteena oli suunnitella, toteuttaa ja käyttöönottaa Jyväskylän ammattikorkeakoulun energiatekniikan laboratorioon uusi prosessisimulaattori energiatekniikan opiskelijoiden laboratoriotöitä varten. Laboratoriossa on neljän vettä kierrättävän prosessin prosessikokonaisuus ja sinne oli tavoitteena suunnitella lisäosana fyysisen prosessin korvaava simulaattori kiinteän polttoaineen syöttöprosessille.</p> <p>Lähtötietoina toimi prosessin PI-kaavio, laitteiden toimintakuvaukset sekä laitteiden mitoitustiedot. Lisäksi lähtötietoina toimi myös Mikko Valkeejärven tekemä opinnäytetyö vastaavasta aiheesta. Simulaattori suunniteltiin ja toteutettiin tietokoneella käyttäen Valmet DNA -suunnittelu ympäristöä.</p> <p>Lopputuloksena saatiin suunniteltua toimiva prosessisimulaattori kiinteän polttoaineen syöttöprosessille. Simulaattori vastasi toimeksiantajan tarpeita ja vaatimuksia ja siten siitä saatiin Jyväskylän ammattikorkeakoulun opiskelijoille uusi laboratorioympäristö. Opiskelijat saivat mahdollisuuden ajaa ja tutkia neljän vettä kierrättävän prosessin prosessikokonaisuuden lisäksi uutta kiinteää polttoainetta syöttävää prosessia tietokonemallinnuksella ilman fyysistä prosessia.</p> <p>Raportin sisältö toteutettiin päiväkirjamuotoisella raportoinnilla. Joka päivä kirjoitettiin päiväkirjaa ja aina viikon päätteeksi kirjoitettiin viikkoanalyysi. Viikkoanalyysissä tuotiin esiin teorian tietoa viikon päiväkirjamerkintöjen mukaisesti. Joka viikolle asetettiin tavoitteet ja ne kerrottiin ja analysoitiin viikkoanalyysissä.</p>		
Avainsanat (asiasanat) Prosessisimulaattori, suunnittelu, ohjelmointi, Valmet DNA -suunnittelu ympäristö, testaus		
Muut tiedot (Salassa pidettävät liitteet)		

Author(s) Välimaa, Juuso	Type of publication Bachelor's thesis, AMK	Date October 2020 Language of publication: Finnish
	Number of pages 58	Permission for web publication: x
	Title of publication Engineering and implementing of process simulator in the Valmet DNA engineering environment	
Degree programme Bachelor of automationengineering, JAMK		
Supervisor(s) Kuisma, Ari; Flyktman, Teppo		
Assigned by Jyväskylä University of Applied Sciences		
Abstract <p>The objective was to engineer and implement new process simulator for energy engineering laboratory of Jyväskylä University of Applied Sciences and for the energy engineering student laboratory works. In this particular laboratory there is a water and steam process that consists of four different modules and the objective was to engineer simulator for the solid fuel feeding system. The simulator was supposed to be an addition that will substitute for process with physical devices.</p> <p>As an initial information of the process simulator were PI-diagram of the process, performance specification and sizing information of process devices. The bachelor's thesis of Mikko Valkeejärvi on a similar thesis topic is also the source information. The simulator was engineered and programmed by using computer and Valmet DNA engineering software.</p> <p>The result of engineering and programming was a properly working process simulator. The simulator contained all the needs and specifications which the client demanded, and that way Jyväskylä University of Applied Sciences and the energy engineering students got a new laboratory environment. In addition to the existent four-module process students got possibility to drive and resource new solid fuel feeding system process with computer simulation without process with physical devices.</p> <p>The report was written by keeping a journal. The Journal was written every day and an analysis of week was made after every week. The Analysis of the week was made by writing theoretical text according to journal entries. Goals was set in the beginning of every week and the goals was reported and analyzed in the analysis of week.</p>		
Keywords/tags (subjects) Engineering, programming, test driving, process simulator, Valmet DNA engineering environment		
Miscellaneous (Confidential information)		

Sisältö

Käsiteluettelo	3
1 Johdanto	5
2 Toimintaympäristö	6
2.1 Valmet DNA -suunnitteluympäristö	6
3 Päiväkirjaraportointi.....	10
3.1 Seurantaviikko 1	10
3.2 Seurantaviikko 2	17
3.3 Seurantaviikko 3	22
3.4 Seurantaviikko 4	25
3.5 Seurantaviikko 5	27
3.6 Seurantaviikko 6	29
3.7 Seurantaviikko 7	32
3.8 Seurantaviikko 8	35
3.9 Seurantaviikko 9	37
3.10 Seurantaviikko 10	39
4 Testaus ja tulokset.....	46
4.1 Testaussuunnitelma	46
4.2 Tulokset	47
5 Pohdinta.....	47
Lähteet	50
Liitteet.....	51
Liite 1. PI-kaavio Huom. PI-kaaviosta puuttuu lopullisessa simulaattorissa olevat virtausmittaukset FI-1, FI-2, FI-3 ja FI-4.	51
Liite 2. I/O-luettelo	52
Liite 3. Pyörintävahdin simulaattoriohjelma	53
Liite 4. Tukosvahdin simulaattoriohjelma	54
Liite 5. Polttoainesäiliön sekoittimen simulaattoriohjelma	55

Liite 6. Sulkusyöttimen simulaattoriohjelma	56
Liite 7. Magneettiventtiilin(sulkupelti) simulaattoriohjelma	57
Liite 8. I/O-signaalien tarkistuslista	58

Kuviot

Kuvio 1. Function Block CAD -näkyä polttoaineen syöttöjärjestelmän kolakuljettimen ohjauksesta.....	6
Kuvio 2. Sequence CAD -näkyä	7
Kuvio 3. Picture Designer -näkyä polttoaineen syöttöjärjestelmän valvomosta	8
Kuvio 4. DNA Explorer -näkyä.....	9
Kuvio 5. DNA Operate Client -näkyä polttoaineen syöttöjärjestelmän lopullisesta valvomosta	10
Kuvio 6. Havainnollistava piirros prosessisimulaattorin rakenteesta	13
Kuvio 7. Lead lag -lohko	14
Kuvio 8. Delay-lohko	15
Kuvio 9. Testiohjelma viiveelliselle siirtofunktiolle	17
Kuvio 10. I/O-rajapinta	21
Kuvio 11. Moottorin ohjauksen simulaattoriohjelma	29
Kuvio 12. Magneettiventtiilin(sulkupelti) ohjaus	32
Kuvio 13. Virtausmittauksen simulointi.....	34
Kuvio 14. Pinnankorkeuden laskenta funktion block cadissa.....	36
Kuvio 15. Polttoainesäiliön pinnankorkeusmittaus polttoaineen lisästoiminnon lisäyksen jälkeen.	38
Kuvio 16. Syöttötaskun pintasäädön säätöpiiri	42
Kuvio 17. Kolakuljettimen nopeusohje (sovellusohjelma)	43
Kuvio 18. Syöttötaskun pinnankorkeuden mittaus (simulaattoriohjelma)	44
Kuvio 19. Syöttöruuvien massavirta (syöttöruuveja on kaksi, molempien massavirrat ovat samanlaiset).....	45

Käsiteluettelo

Function Block CAD	Valmet DNA -suunnitteluympäristön työkalu, jolla suunnitellaan prosessin ohjaukseen tarvittavat ohjelmat.
I/O-luettelo	Luettelo, johon on koostettu prosessilaitteet ja niiden digitaaliset ja analogiset input- ja outputtiedot. Lisäksi luettelossa näkyy laitteiden mittayksiköt yms.
I/O-rajapinta	Tässä työssä rajapinta, jonka avulla sovellusohjelma kommunikoi fyysisen prosessin korvaavan simulaattoriohjelman kanssa.
Mallinnus	Mallinnuksella kuvataan prosessin yksittäistä toimintaa simuloinnissa.
Mitoitustiedot	Lähtötietojen osio, jossa kerrotaan prosessin laitteiden mittayksiköt yms.
Picture Designer	Valmet DNA -suunnitteluympäristön työkalu, jolla piirretään prosessista valvomokuva.
PI-kaavio	Piirretty kaavio prosessin rakenteesta ja ohjauslaitteista. Kaavio sisältää kaikki prosessissa olevat putkistot, säiliöt ja toimilaitteet yms. sekä niiden liitynnät. Kaavio siis kertoo prosessin rakenteellisen informaation.
Prosessisimulointi	Fyysisen prosessin toiminnan jäljittelyä tietokoneella.
Siirtofunktio	Matemaattinen malli, joka kuvaa säätöjärjestelmän tulon ja lähdön välistä riippuvuutta.

Simulaattoriohjelma	Mallinnus, joka korvaa tässä simuloinnissa fyysisen prosessin.
Sovellusohjelma	Prosessin ohjausta varten suunniteltava ohjelma.
Toimilohkokaavio	Prosessin ohjausta varten Function Block CAD -työkalulla suunniteltu yksittäinen ohjelma.
Toimintakuvaukset	Lähtötietojen osio, johon on jäsennelty, miten prosessin eri laitteiden täytyy toimia.
Toimintaympäristö	Ympäristö, missä suunnittelutyötä sekä tehdään että käytetään.
Vaativuusmäärittely	Lähtötietojen osio, jossa määritetään työn yleiset vaatimukset.
Valmet DNA - suunnittelu ympäristö	Prosessiautomaatiosovelluksille tarkoitettu ohjaus- ja suunnittelujärjestelmä.
Valvomo	Valvomossa näkyy prosessin rakenne ja laitteet ja valvomosta prosessia ohjataan.

1 Johdanto

Jyväskylän ammattikorkeakoulun energiatekniikan laboratorio toimii oppimisympäristönä automaatiotekniikan ja energiatekniikan opiskelijoille. Laboratoriossa on prosessikokonaisuus, joka tällä hetkellä koostuu neljästä pienemmästä prosessiyksiköstä. Jokaisessa prosessiyksikössä kierrätetään ja/tai käsitellään vettä.

Tämän opinnäytetyön tavoitteena on suunnitella ja käyttöönottaa laboratorioon uusi aiempiin nähden erilainen prosessiympäristö oppimisympäristöksi energiatekniikan opiskelijoita varten. Tavoitteena on suunnitella ja ohjelmoida prosessisimulaattori kiinteän polttoaineen syöttöprosessille. Tämä tarkoittaa sitä, että syöttöprosessia ei toteuteta fyysisesti, vaan siitä suunnitellaan ja ohjelmoidaan tietokonemallinnus. Tätä mallinnusta voidaan ohjata tietokoneella samalla tavoin kuin oikeaa prosessia. Simulaattorin tulee olla sellainen, että opiskelijat voivat ohjata ja ajaa prosessia valmiin sovellusohjelman avulla. Lisäksi simulaattorin on oltava sellainen, että opiskelijoilla on myös mahdollisuus ohjelmoida itse tämä sovellusohjelma tietämättä varsinaisesta simuloinnista yhtään mitään.

Työ toteutetaan käyttäen Valmet DNA -suunnittelu ympäristöä. Prosessisimulaattoria varten tulee suunnitella simulaattoriohjelma, joka korvaa fyysisen prosessin. Jotta simulaattoriohjelmaa voidaan ohjata, tulee sille suunnitella sovellusohjelma. Lopuksi tulee suunnitella valvomonäyttö, josta prosessia(simulaattoria) voidaan ohjata ja ajaa. Valmiille prosessisimulaattorille tulee tehdä koeajoja, joiden avulla se voidaan todeta toimivaksi ympäristöksi.

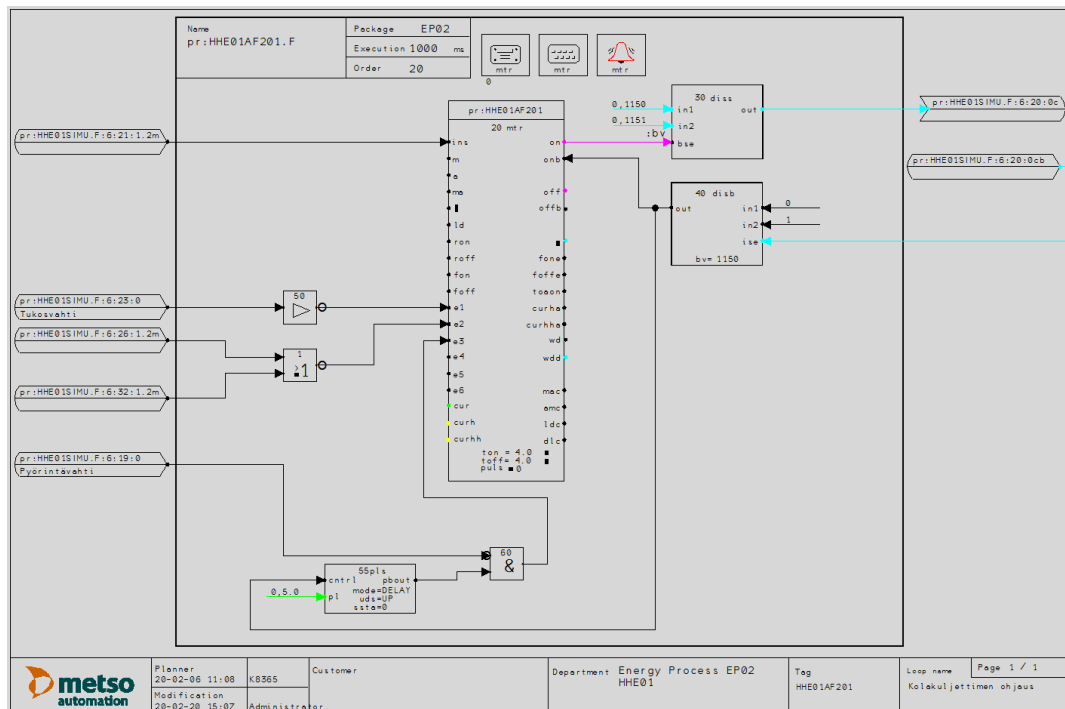
Työn lähtötietoina toimii prosessin PI-kaavio (Ks. liite 1.), laitteiden toimintakuvaukset ja laitteiden mitoitus tiedot. Lisäksi tämä työ on sikäli ainutlaatuinen, että lähtötietona toimii myös Mikko Valkeejärven tekemä opinnäytetyö vastaavasta aiheesta kuin tämä työ. Tämä työ aloitettiin tammikuun puolessa välissä ja se on tarkoitus saada valmiiksi maaliskuun lopussa.

2 Toimintaympäristö

2.1 Valmet DNA -suunnittelu ympäristö

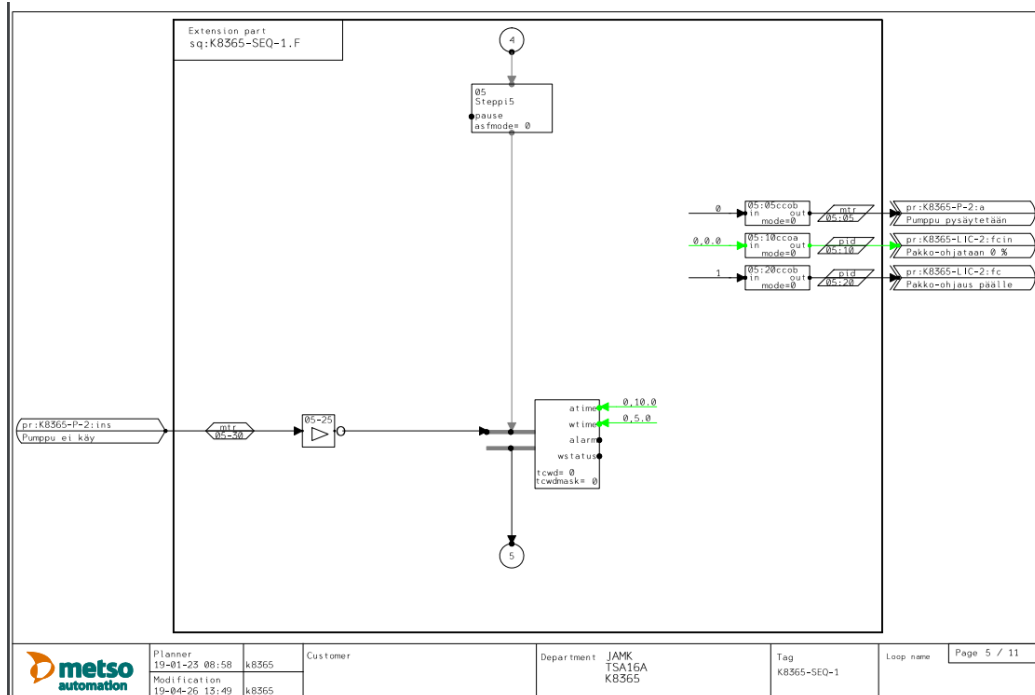
Valmet DNA on prosessiautomaatiosovelluksille tarkoitettu hajautettu ohjausjärjestelmä (Distributed Control System, DCS) ja informaatioalusta. Sillä voidaan hoitaa yksittäiseltä alustalta kaikki ohjaus prosesseja, koneita, laatua ja ominaisuutta, valvontaa, optimointia jne. varten. (Valmet DNA n.d.)

Valmet DNA -suunnittelu ympäristössä on suunnittelua varten Function Block CAD -työkalu, Sequence CAD -työkalu ja Picture Designer -työkalu. Function Block CAD -työkalulla voidaan suunnitella tarvittavat ohjelmat prosessin ohjausta ja säätämistä varten. (Valmet DNA Engineering Function Block CAD n.d.) Työkalu näyttää kuvion 1 mukaiselta.



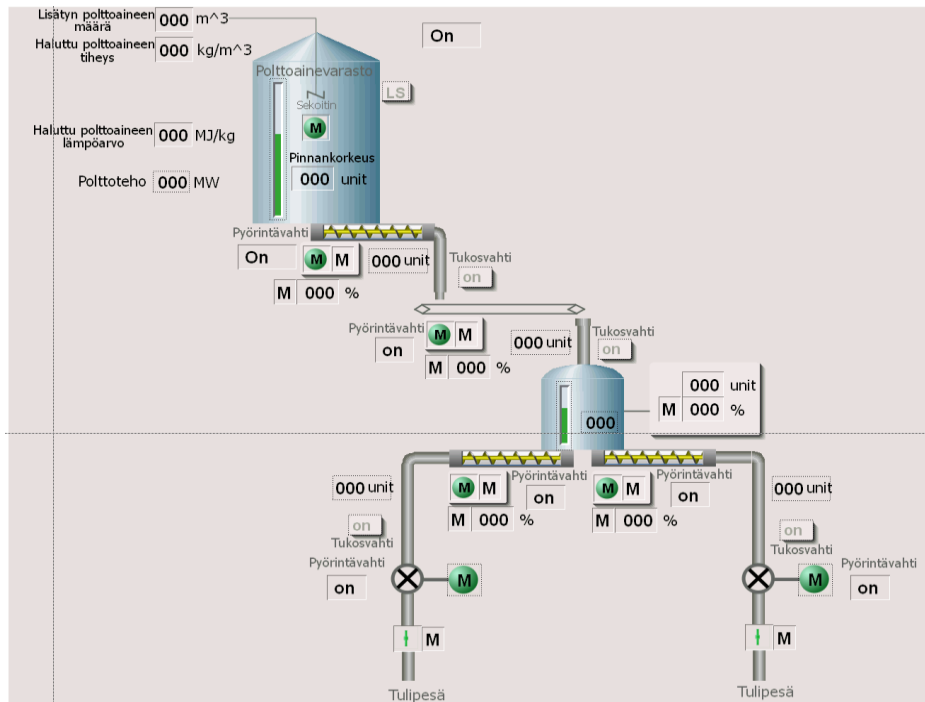
Kuvio 1. Function Block CAD -näkyä polttoaineen syöttöjärjestelmän kotikuljettimen ohjauksesta

Valmet DNA -suunnitteluympäristössä voidaan myös suunnitella prosessille sekvenssiohjelma eli ohjelma, joka käynnistyspainiketta painamalla suorittaa prosessille halutut toiminnot automaattisesti. Tähän käytetään Sequence CAD -työkalua. (Valmet DNA Engineering Function Block CAD n.d.) Tämä työkalu näyttää kuvion 2 mukaiselta.



Kuvio 2. Sequence CAD -näkyvä

Jotta prosessia voidaan ohjata suunnitellulla ohjelmalla, täytyy prosessille piirtää valvomonäyttö. Valvomonäytössä näkyy prosessin rakenne ja laitteet, joita halutaan ohjata. Valvomonäytön suunnitteluun käytetään Picture Designer -työkalua. (Valmet DNA Engineering Function Block CAD n.d.) Tämä työkalu näyttää kuvion 3 mukaiselta.



Kuvio 3. Picture Designer -näkyvä polttoaineen syöttöjärjestelmän valvomosta

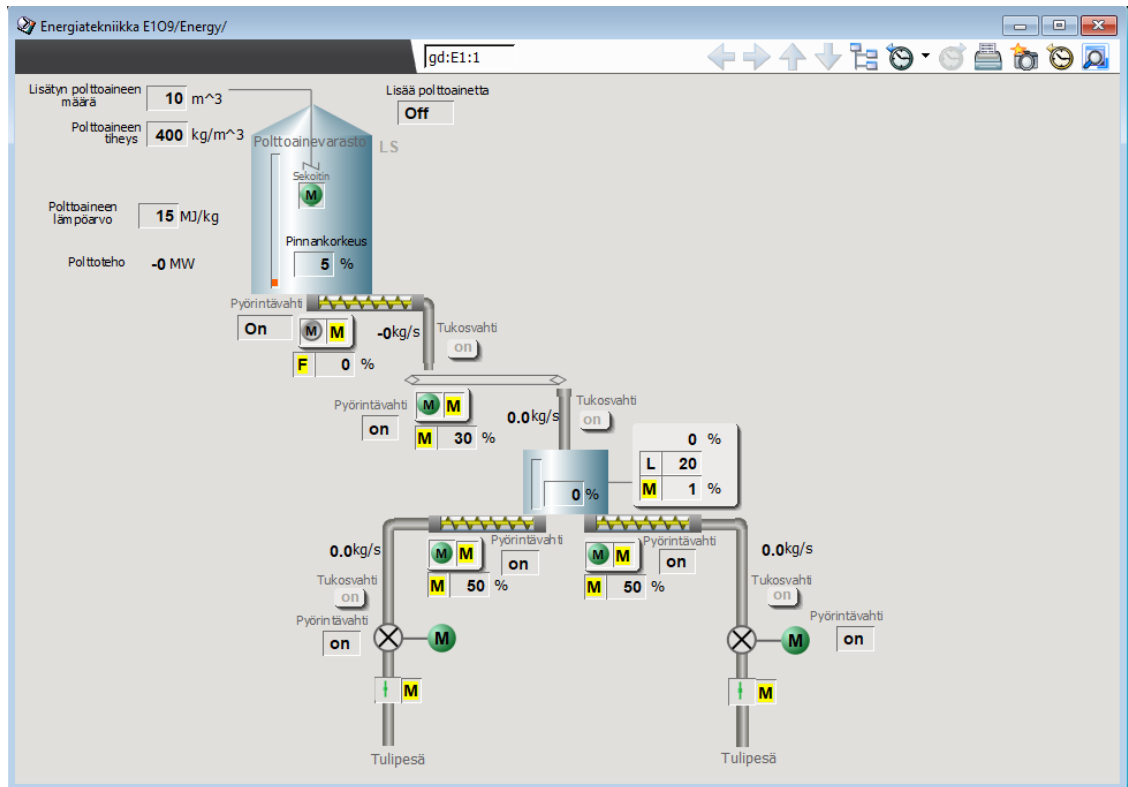
Valmiit ohjelmat ja valvomot täytyy tallentaa paikkaan, josta ne saa helposti käyttöön. Tätä varten Valmet DNA -suunnittelu ympäristöstä löytyy DNA Explorer -konfiguraatio työkalu, jonka avulla voidaan käsitellä suunnittelutietokantaa, jonne työt voidaan tallentaa (Ks. kuvio 4.). DNA Explorer toimii siis työkaluna, jolla voidaan tallentaa ja käsitellä Valmet DNA -suunnittelu ympäristössä tehtyjä ohjelmia ja valvomoja. (Valmet DNA explorer - Integrates all engineering functions n.d.)

The screenshot shows the DNA Explorer application window. The title bar indicates the user is logged in as 'Application Engineer' and is viewing a 'Process Area Hierarchy' repository. The interface includes a menu bar (Object, Edit, View, Design, Upload/Download, Logs, Window, Help), a toolbar with various icons, and a search bar with the 'metso' logo. The main area is divided into two panes. The left pane shows a tree view of the 'Process Area Hierarchy' with folders like 'Design Accessories', 'Diagnostic modules E1', 'Energy Process EP01-Standard Program', 'Energy Process EP02-Student Program', 'HHE01-SOLID Fuel System', 'LAA01-Pressurized-water Module', 'LCA01-Steam Generator', 'LFA01-Pumping Module', 'INFO', 'JAMK-For Students', 'Not Categorized', 'System Modules', 'Trends', and 'Water Process Simulator'. The right pane displays a table of function block diagrams.

Identifier	Name	Category	Description
Demo	Sirtofunktio	Function Block Diagram	
gd:E1:0		Graphic Display	
gd:E1:1		Graphic Display	
HHE01AF201	Kolakujuetimen ohjaus	Function Block Diagram	
HHE01AF301	Ruuvikujuetimen ohjaus	Function Block Diagram	
HHE01AF302	Ruuvikujuetimen ohjaus	Function Block Diagram	
HHE01AF303	Ruuvikujuetimen ohjaus	Function Block Diagram	
HHE01AM301	Sekoittimen ohjaus	Function Block Diagram	
HHE01CL201	Polttoainesäilön pinnankorkeud...	Function Block Diagram	
HHE01CS201	Ruuvikujuetimen nopeusohje	Function Block Diagram	
HHE01CS204	Kolakujuetimen nopeusohje	Function Block Diagram	
HHE01CS205	Ruuvikujuetimen nopeusohje	Function Block Diagram	
HHE01CS206	Ruuvikujuetimen nopeusohje	Function Block Diagram	
HHE01SIMU	Rajapinta	Function Block Diagram	
HHE01SIMU-AF201	Kolakujuetimen ohjaus	Function Block Diagram	
HHE01SIMU-AF301	Ruuvikujuetimen ohjaus	Function Block Diagram	
HHE01SIMU-AF302	Ruuvikujuetimen ohjaus	Function Block Diagram	
HHE01SIMU-AF303	Ruuvikujuetimen ohjaus	Function Block Diagram	
HHE01SIMU-AM301	Sekoittimen ohjaus	Function Block Diagram	
HHE01SIMU-CF101	Tukosvahti	Function Block Diagram	
HHE01SIMU-CF102	Tukosvahti	Function Block Diagram	
HHE01SIMU-CF103	Tukosvahti	Function Block Diagram	
HHE01SIMU-CF104	Tukosvahti	Function Block Diagram	
HHE01SIMU-CS101	Pyörintävahti	Function Block Diagram	
HHE01SIMU-CS102	Pyörintävahti	Function Block Diagram	
HHE01SIMU-CS103	Pyörintävahti	Function Block Diagram	
HHE01SIMU-CS104	Pyörintävahti	Function Block Diagram	

Kuvio 4. DNA Explorer -näkyvä

Varsinainen prosessinohjaus ohjelmien ja valvomönäytön avulla tapahtuu omalla työkalulla. Ohjaukseen käytetään DNA Operate Client -valvomosovellusta (Ks. kuvio 5.). Sovelluksen avulla valvomolla saadaan yhteys prosessiin. Prosessia voidaan näin ohjata ja sen tilaa voidaan seurata. Sovelluksen avulla valvomosta voidaan nähdä mittauslaitteiden mittaukset ja ohjattavien laitteiden käyntitilat ja siten koko prosessin käyntitilan. Valvomosta voidaan myös nähdä mahdolliset laiteviat ja toimintahäiriöt vikailmoitusten avulla. Lisäksi säädettävien toimilaitteiden ja mahdollisten säätöpiirien käyttäytymistä voidaan tarkemmin tutkia trendikäyrien avulla. (DNA Operate n.d.)



Kuvio 5. DNA Operate Client -näkyä polttoaineen syöttöjärjestelmän lopullisesta valvomosta

3 Päiväkirjaraportointi

3.1 Seurantaviikko 1

Tiistai 14.1.

Aloitin tämän suunnittelutyöni tekemisen. Päivä alkoi sillä, että aamulla tapasin työni ohjaajan. Ohjaajan kanssa teimme minulle Valmet DNA Explorer -konfiguraatiotyökallulla tietokannan, jonne voin ohjelmoida ja suunnitella prosessisimulaattorin. Samaan tietokantaan lisäsimme minulle ensimmäiseksi lähtötiedoksi Mikko Valkeejärven opinnäytetyönä tekemän prosessisimulaattorin toimilohkokaaviot. Loppupäivän käytinkin siihen, että tutkin näitä Mikko Valkeejärven tekemiä toimilohkokaavioita sekä hänen varsinaista opinnäytetyötään. Näin sain jo hieman käsitystä prosessisimulaattorin toiminnasta ja rakenteesta prosessin eri vaiheissa. Työni varsinaiset lähtötiedot

saan ohjaajaltani myöhemmin. Sovimme, että hän on minuun yhteydessä, kun hänellä on lähtötiedot.

Keskiviikko 15.1.

Aloitin päiväni sillä, että perehdyin tarkemmin ihan toimilohko tasolla Valkeejärven tekemiin simulaattorin toimilohkokaavioihin. Tutkin Valmet DNA -suunnittelu ympäristön manuaalin sekä Valkeejärven opinnäytetyön avulla simulaattorille oleellisen siirtofunktio-rakenteen toimintaa. Loppupäivän käytin siihen, että ohjelmoin Valmet DNA -suunnittelu ympäristössä testiohjelman siirtofunktion toiminnan hahmottamista varten. Testiohjelma toimii käytännössä niin, että valvomonäytölle tehtyyn syöttökenttään syötetään jokin numeerinen arvo kuvastamaan kuvitteellisen prosessin laitteelle tehtävää säätöä. Tämä arvo päivittyy valvomonäytölle tehtyyn lukukenttään, joka kuvastaa prosessin laitetta. Siirtofunktio-rakenne saa aikaan sen, että tämä edellä mainittu ”säätö” päivittyy lukukenttään oikean prosessilaitteen tavoin. Tein valvomonäytölle syöttökentät myös siirtofunktion parametrien säätöä varten. En kuitenkaan saanut säätöä toimimaan vielä tänään, mutta huomenna teen sen valmiiksi ja tutkin testiohjelmaa muutenkin vielä tarkemmin.

Torstai 16.1.

Koska eilen tekemäni testiohjelma jäi kesken siirtofunktion parametrien säädön osalta, päiväni alkoi sillä, että aamupäivällä ohjelmoin parametrien säädön toimintakuntoon. Loppupäivän käytin siihen, että luin säätötekniikan kurssimateriaalia ja palautin mieleeni, miten 1. kertaluvun siirtofunktio muodostuu kaavatasolla. Tutkin kaavan ja teorian avulla, kuinka siirtofunktion parametrit vaikuttavat, kun asetuservoa muutetaan. Tätä havainnollistin itselleni siten, että ajoin testiohjelmaani eri parametrisäädöillä ja katsoin trendikäyrästä vaikutusta.

Aamupäivän aikana tapasin ohjaajani ja sain ensimmäisen varsinaisen lähtötiedon prosessisimulaattoria varten. Sain prosessin PI-kaavion ja kävin sen komponenttien osalta läpi ohjaajani kanssa. Huomenna on tarkoitus alkaa hahmottelemaan valvomoa prosessin ohjausta varten.

Perjantai 17.1.

Aloitin päiväni siten, että tutkin eilen saamaani PI-kaaviota tarkemmin. Kävin PI-kaaviosta läpi, mitä kaikkia laitteita, mittauksia, kytkimiä yms. polttoaineen syöttöprosessi sisältää. Kun sain käsityksen siitä, millainen prosessin rakenne on, aloitin valvomon hahmottelemisen ensin paperille. Kun sain piirrettyä selkeän näköisen valvomon paperille, minun oli helppo aloittaa paperisen kuvan avulla valvomon suunnittelu Valmet DNA -suunnitteluympäristön Picture Designilla -työkalulla. Valvomon suunnitteluun kuluikin lähes koko loppupäivä enkä itse asiassa saanut sitä vielä valmiiksi. Sain ohjaajaltani muutaman ohjeen valvomon viimeistelyä varten ja teenkin näillä ohjeilla valvomon valmiiksi ensi viikolla. Päivän viimeisenä tehtävänä aloitin I/O-luettelon tekemisen Excel-taulukkoon. Kirjaan taulukkoon jokaisen laitteen position, mahdollisen mittausalueen ja mittausyksikön, nimen sekä I/O-signaalit. Tässä vaiheessa en tiedä vielä mitä muuta taulukkoon tulee. Tämäkin selviää, kun teen taulukon valmiiksi ensi viikolla.

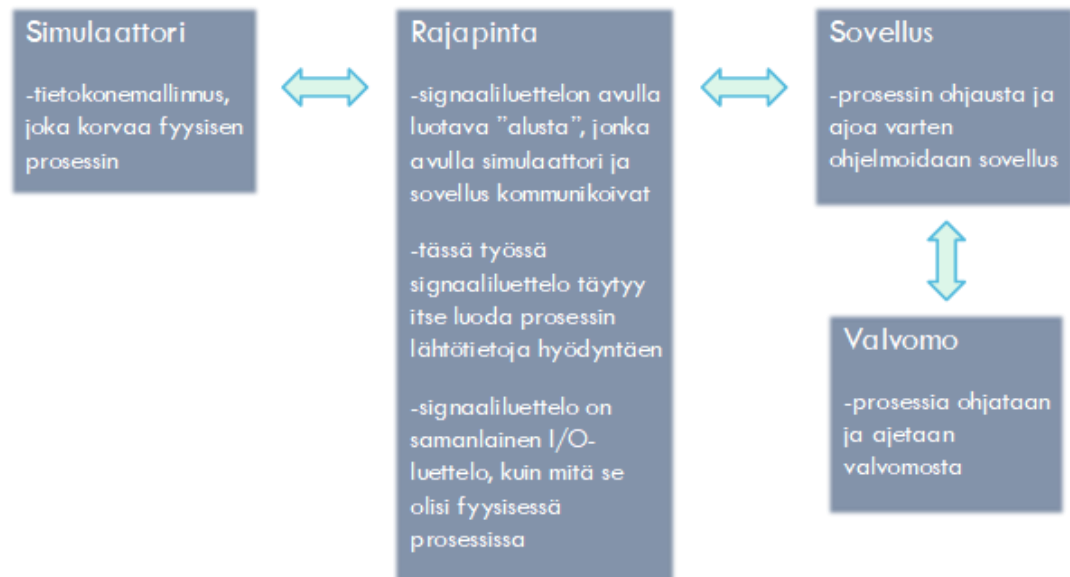
Viikkoanalyysi

Tällä ensimmäisellä viikolla tavoitteenani oli perehtyä prosessisimulaattorin suunnitteluun teoriatasolla. Tutkin työlleni oleellista käsitteistöä sekä teoriassa että käytännössä. Opin mitä tarkoittaa prosessisimulaatio ja miten siinä voidaan mallintaa prosessin vaiheita.

Prosessisimulaatio tarkoittaa sitä, että jäljitellään oikean prosessin toimintaa tietokoneella. Simulaatiota voidaan käyttää siihen, että tutkitaan suunnitteilla olevan prosessin käyttäytymistä ennen, kun prosessi toteutetaan fyysisesti. (Banks, Carson II, Nelson & Nicol 2014, 1.)

Tässä työssä prosessisimulaatio koostuu neljästä osasta, jotka ovat simulaattori, rajapinta, sovellus ja valvomo (Ks. kuvio 6.). Simulaattorina toimii tietokonemallinnus, joka korvaa fyysisen prosessin. Jotta simulaattoriin voidaan olla yhteydessä, täytyy luoda fyysiset kytkennät korvaava rajapinta. Rajapinta luodaan käyttäen samanlaista I/O-signaaliluetteloja kuin mitä fyysisessä prosessissakin käytettäisiin. Tässä työssä

myös I/O-signaaliluettelo täytyy ensin luoda prosessin lähtötietoja hyödyntäen. Rajapinnan lisäksi täytyy suunnitella ja ohjelmoida sovellus, jonka avulla simulaattoria voidaan rajapinnan välityksellä ohjata. Lopuksi täytyy vielä suunnitella valvomo, josta ohjataan prosessia.



Kuvio 6. Havainnollistava piirros prosessisimulaattorin rakenteesta

Varsinaisessa simulaatiossa prosessin käyttäytymistä ja eri toimintoja kuvataan mallinnuksien avulla. Mallinnuksien luonne ja muoto määräytyy luonnollisesti prosessin käyttäytymisen mukaan. (Banks ym. 2014, 1.) Nämä simulaatiomallit voidaan jakaa staattisiin ja dynaamisiin malleihin, deterministisiin ja stokastisiin malleihin sekä diskreetteihin ja jatkuviin malleihin (Banks ym. 2014, 13).

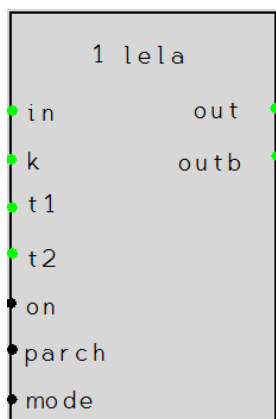
Staattinen ja dynaaminen malli: Staattinen malli kuvastaa systeemiä, jossa tapahtuu muutosta vain tiettyinä ajankohtina ja dynaaminen malli kuvastaa systeemiä, jossa tapahtuu muutosta jatkuvasti.

Deterministinen ja stokastinen malli: Deterministisessä mallissa sisään tuleva "data" on ennalta tunnettua ja stokastisessa mallissa sisään tuleva "data" on satunnaista ennalta tuntematonta.

Diskreettinen ja jatkuva malli: Diskreettinen malli kuvastaa systeemiä, jossa muutosta tapahtuu vain tietyn ajan ja jatkuva malli kuvastaa systeemiä, jossa muutosta tapahtuu äärettömän kauan. (Banks ym. 2014, 12.)

Tässä työssä tärkeänä mallinnustapana toimii matemaattinen siirtofunktio. Matemaattinen siirtofunktio on malli, jolla esitetään säätöjärjestelmän tulon ja lähdön välistä riippuvuutta. Siirtofunktiolla siis esitetään kuinka säätöjärjestelmän tuloon syötetty asetusarvo vaikuttaa säätöjärjestelmän hetkelliseen arvoon eli lähtöön. (Transfer function of control system 2019.)

Valmet DNA -suunnitteluympäristön Function Block CAD -työkalussa siirtofunktiorakennetta kuvastaa lead lag -lohko (Metso DNA Manuals 2011). (Ks. kuvio 7.)



Kuvio 7. Lead lag -lohko

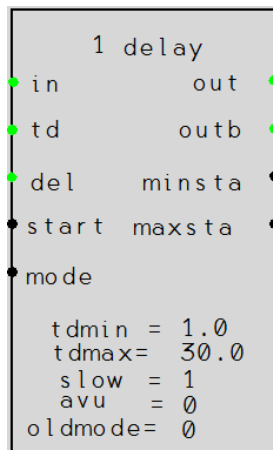
Lohko muodostaa jatkuvan ensimmäisen asteen siirtofunktion, jonka mukaisesti se muuttaa lähdössä(out) olevan arvon tuloon(in) syötettyyn arvoon. Koska tässä simulaattorissa käytetään siirtofunktiossa vain yhtä aikavakiota, on funktion kaava muotoa:

$$G(s) = \frac{k}{1 + T * s}$$

missä k=vahvistuskerroin ja T=aikavakio.

Aikavakio(T) säättää sitä, kuinka nopeasti tuloon syötetty arvo muuttuu lähtöön. Vahvistuskerroin(k) säättää sitä, kuinka paljon funktiossa on vahvistusta eli kuinka moninkertaisena tuloon syötetty arvo tulee lähtöön. Lead lag -lohkossa tuloon k syötetään vahvistuskerroin ja käytettäessä yhtä aikavakiota tuloon t2 syötetään aikavakio. (Metso DNA Manuals 2011.)

Tämän edellä mainitun siirtofunktio-käyttäytymisen lisäksi prosessilaitetta käytettäessä syntyy usein viive ennen kuin muutos alkaa tapahtua. Valmet DNA -suunnittelu-ympäristön Function Block CAD -työkalussa tämän viiveen mallinnukseen käytetään delay-lohkoa (Metso DNA Manuals 2011). (Ks. kuvio 8.)



Kuvio 8. Delay-lohko

Delay-lohko toimii niin, että se viivästä tuloon in syötetyn arvon etenemistä tuloon td syötetyn ajan verran. Syötetyn viiveen pituus rajoitetaan siten, että lohkon parametreihin syötetään minimiviive tdmin ja maksimiviive tdmax. Lisäksi tuloon del annetaan viiveelle varatun puskurin pituus, joka siten rajoittaa myös viiveen pituutta. Tarvittavan puskurivarauksen koko voidaan laskea kaavasta:

$$K = \frac{\left(\frac{M}{S}\right)}{sv+1}, \text{ missä}$$

K= Puskurivaraus

M= lohkon maksimiviive

S= lohkon jäsen slow

Sv= lohkon suoritusväli.

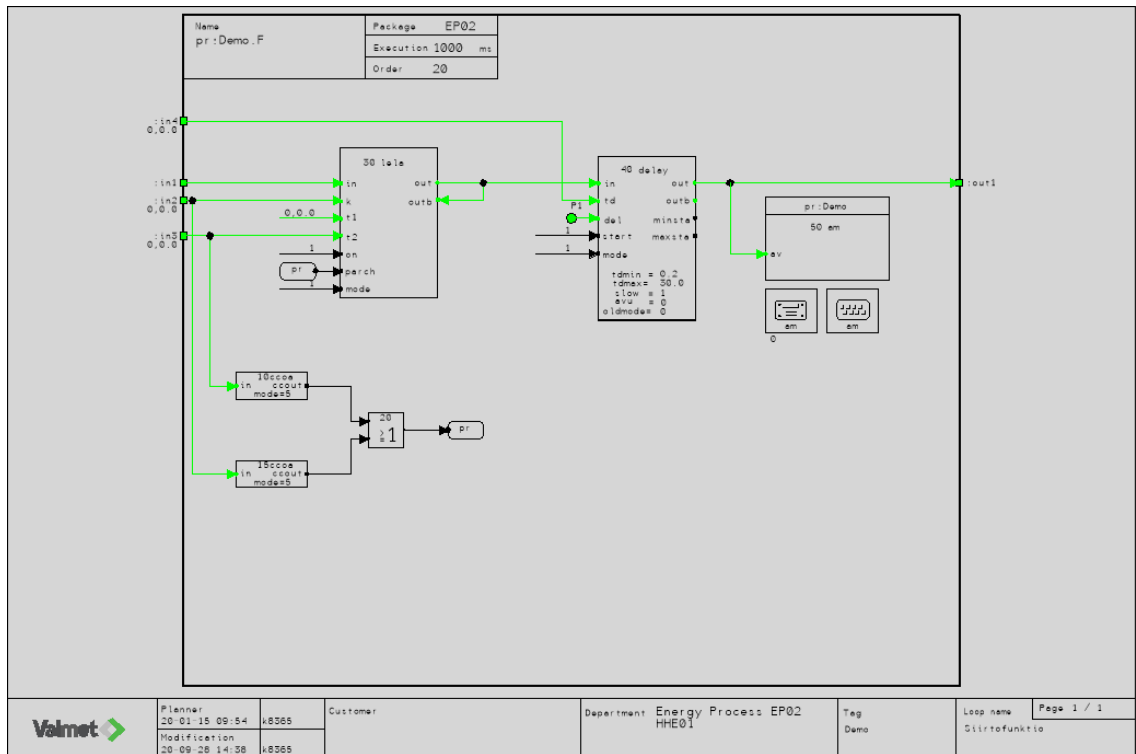
Viive saadaan siirtofunktioon mukaan kytkemällä lead lag -lohkon lähtö(out), delay-lohkon tuloon(in). (Metso DNA Manuals 2011.)

Näin saadaan muodostettua tässä työssä tärkeä ensimmäisen asteen viiveellinen siirtofunktio. Kun viive lisätään aiemmin mainittuun ensimmäisen asteen siirtofunktion kaavaan, tulee kaavasta seuraavanlainen:

$$G(s) = \frac{k}{1 + T * s} * e^{-Ls}$$

missä k=vahvistuskerroin, T=aikavakio ja L=viive (Kippo & Tikka 2008, 116.)

Hahmottaakseni tämän viiveellisen siirtofunktiorakenteen toiminnan käytännössä tein siitä testiohjelman. Tein ohjelman viiveelliselle siirtofunktiolle edellä mainittuja lead lag -lohkoa ja delay-lohkoa käyttäen. (Ks. kuvio 9.) Tekemästäni valvomosta pystyin syöttämään siirtofunktioon haluamani arvon ja näin, miten se päivittyi siirtofunktion mukaisesti delay-lohkon ulostuloon ja siten valvomossa olevaan ulostulokenttään. Tein valvomoon myös kentät siirtofunktion parametrien säätämistä varten.



Kuvio 9. Testiohjelma viiveelliselle siirtofunktiolle

Jotta sain käsityksen siitä, millainen rakenne ja millaisia toimintoja työni varsinaisessa polttoaineensyöttöprosessissa on, tutkin lähtötietona saamaani prosessin PI-kaaviota ja piirsin Valmet DNA -suunnitteluympäristön Picture Designer -työkalulla prosessille valvomon. (Ks. kuvio 3. ja kuvio 5.)

3.2 Seurantaviikko 2

Maanantai 20.1.

Aloitin päiväni sillä, että viimeistelin ja korjasin polttoaineen syöttöprosessin valvomon ohjaajaltani saamieni ohjeiden mukaisesti. Ensiksi jouduin piirtämään puolet valvomosta uudelleen. Olin aluksi piirtänyt prosessin valvomonäkymän vaakasuuntaan vasemmalta oikealle eteneväksi, koska ajattelin, että näin prosessia on helpompi tulkita ja ohjata valvomosta. En kuitenkaan oivaltanut, että tämä esitys ei vastaa todellisuutta, sillä todellisuudessa polttoaine on syötettävä ylhäältä alaspäin pu-

dottamalla, jotta se saadaan painovoiman avulla etenemään putkistossa riittävän tehokkaasti (Kuisma 2020). Kun olin piirtänyt valvomon niin, että se vastaa todellisuutta, tein sinne muut ohjeiden mukaiset korjaukset ja lisäykset. Näin sain valvomon niin valmiiksi kuin se tässä vaiheessa on mahdollista. Loppupäivän käytin siihen, että täydensin viime viikolla aloittamani I/O-luettelon I/O-signaalien osalta valmiiksi. Muita tietoja kuten mittayksiköt, laitteiden korttitiedot ym. joutuu vielä luetteloon lisäämään siinä vaiheessa, kun saan enemmän lähtötietoja.

Tiistai 21.1.

Aloitin päiväni sillä, että aloin suunnitella Valmet DNA -suunnitteluympäristössä ensimmäistä sovellusohjelmaa ja sitä kautta simulaattoriohjelmaa polttoainevarastosta kiinteää polttoainetta syöttävän ruuvikuljettimen ohjaukselle. Sain aamupäivän aikana tehtyä sovellusohjelman kuljettimen moottorin ohjaukselle. Iltapäivällä oli tarkoitus tehdä moottorin nopeusohjeelle sovellus- ja simulaattoriohjelma sekä simulaattoriohjelma moottorin ohjaukselle. En kuitenkaan päässyt ohjelmoinnissa eteenpäin, sillä ongelmaksi tuli se, että miten korvaan laitteiden korttitiedot simulaattorissa. Ohjaajani neuvoi minua tutkimaan Valkeejärven opinnäytetyötä tarkemmin ja tähän käytinkin iltapäivän. Tutkittuani opinnäytetyötä tarkemmin ymmärsin, että korttitiedot nimetään samalla tavoin kuin fyysisessäkin prosessissa. Erona fyysiseen prosessiin on se, että korttitietoa ei kuljeteta ohjelmassa profibus-lohkolla vaan jatkuvalla viittauksella. Näillä tiedoin sain aloitettua simulaattorin rajapinnan ohjelmoinnin tämän kyseisen kuljettimen osalta. Ohjaajani tuli lopuksi antamaan minulle vielä tarkemmat ohjeet korttitietojen nimeämiseen nimenomaan tässä prosessissa. Näillä ohjeilla ja opinnäytetyöstä saaduilla tiedoilla jatkan huomenna moottorin nopeusohjeen sovellusohjelman ohjelmoinnilla sekä itse simulaattorin ohjelmoinnilla.

Keskiviikko 22.1.

Tarkoitukseni oli jatkaa kuljettimen nopeusohjeen ohjelmointia sovellusohjelman osalta ja tämän jälkeen ohjelmoida simulaattori kuljettimen nopeusohjeelle sekä myös moottorin ohjaukselle. Jouduin kuitenkin ensin täydentämään saamieni lähtötietojen avulla I/O-luettelon valmiiksi korttitietojen osalta. Sen lisäksi, että täydensin

I/O-luetteloa, jouduin myös etsimään taajuusmuuttajien korttitiedoista osan ABB:n taajuusmuuttajia koskevasta manuaalista. Tämä osoittautui aikaa vieväksi työksi ja käytinkin I/O-luettelon täydentämiseen aamupäivän. Iltapäivällä jouduin kuitenkin jatkamaan I/O-luettelon täydentämistä, sillä tajusin, että en löytänyt aamupäivällä oikeita tietoja taajuusmuuttajien I/O-kortteihin. Lähes koko loppupäivä kuluikin siihen, että tutkin ABB:n manuaaleja uudelleen etsien oikeita tietoja. Koska en edelleenkään löytänyt tietoja, jouduin pyytämään ohjaajaltani apua ja siten sain I/O-luettelon valmiiksi korttitietojen osalta. Ehdin vielä tehdä sovellusohjelman kuljettimen nopeusohjeelle. Huomenna jatkan simulaattoriohjelman tekemisellä.

Torstai 23.1.

En ehtinyt aamupäivällä aloittaa kunnolla simulaattoriohjelman tekemistä, kun minulla jo olikin palaveri 1. ohjaajani kanssa. Kävimme palaverissa läpi, miten teen tähän työni raporttiin viikkoanalyysiä ja kuinka käytän lähteitä tällaisessa päiväkirjamuotoisessa raportoinnissa. Kun nämä asiat saatiin käytyä läpi, pääsin aloittamaan simulaattoriohjelman tekemisen kuljettimen moottorin ohjaukselle. Tämän ohjelman sainkin tehtyä ruokatuntiin mennessä. Iltapäivällä minun oli tarkoitus kokeilla ohjelman toimintaa valvomosta käsin, mutta ajettuani viantarkistuksen tekemilleni ohjelmille ja valvomolle, valvomosta tuli vikailmoitus. Tämä vikailmoitus osoittautui niin hankalaksi, että en saanut sitä ratkaistua itsenäisesti. Koska apua ei ollut saatavilla, en saanut vielä kokeiltua ohjelmaani. Sen kokeilu siirtyy maanantaille, sillä huomenna minulla on käytettävissä työni tekemiseen vain aamupäivä ja sen käytän tiedonhakuun.

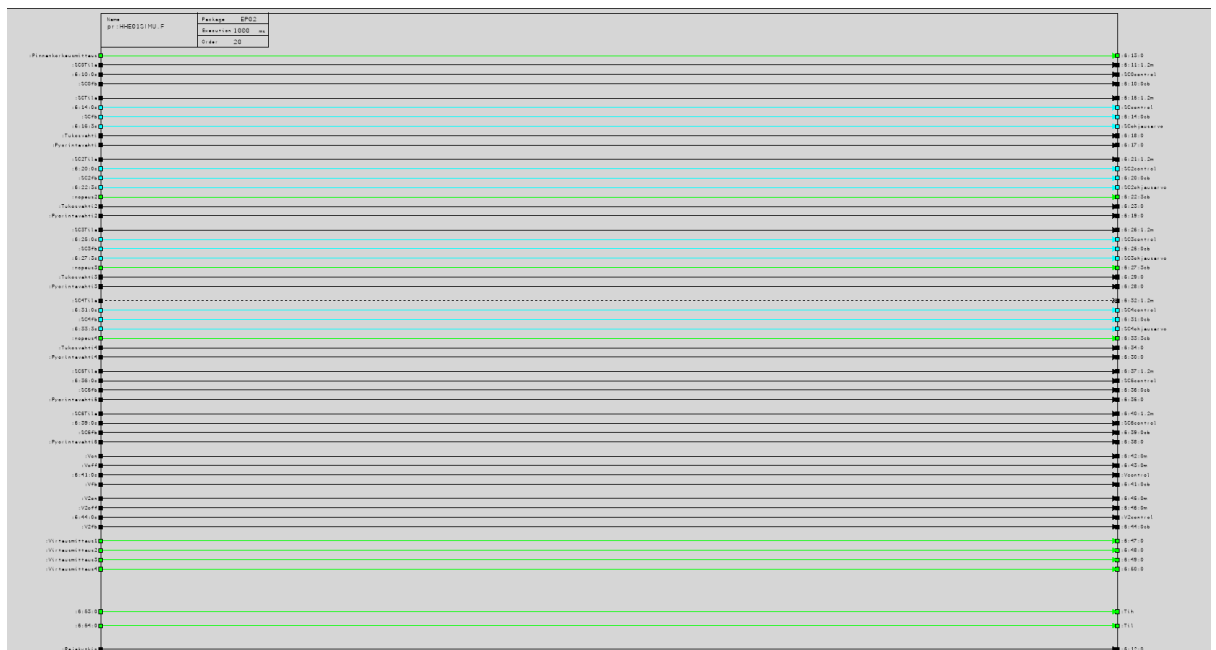
Viikkoanalyysi

Tällä viikolla tavoitteenani oli aloittaa Valmet DNA -suunnitteluympäristössä työni varsinaisen sovellusohjelman ja simulaattoriohjelman tekeminen. Tavoitteenani oli saada polttoaineen syötön ensimmäinen moottoriohjattu ruuvikuljetin toimintaan moottorin binäärisen ohjauksen osalta. Eli moottorin ohjaus päälle ja pois.

Ennen kuin pääsin aloittamaan ohjelman tekemistä moottorille, esittelin ohjaajalleni tekemääni valvomoa. Olin piirtänyt valvomon vaakasuuntaan vasemmalta oikealle hieman PI-kaaviosta poiketen. Tähän ohjaajani pyysi minua tekemään korjauksen ja sitä varten hän kertoi minulle, miten polttoaineen syöttöprosessi toimii ja etenee käytännössä. Kuten PI-kaaviokin niin myös oikea kiinteän polttoaineen syöttöprosessi etenee ylhäältä alaspäin pudottavalla liikkeellä, jotta polttoaine saadaan kulkemaan prosessissa painovoiman avulla riittävän tehokkaasti. Vaakasuunnassa ilman painovoiman avustusta kuljettimet eivät jaksaisi kuljettaa polttoainetta riittävän tehokkaasti ja se saattaisi jumiutua putkistoon. (Kuisma 2020.) Piirsinkin siis valvomon ylhäältä alaspäin eteneväksi (Ks. kuvio 3. ja kuvio 5.).

Koska prosessin sovellusohjelman ja simulaattoriohjelman tekoon tarvittiin I/O-luettelo, päätin tehdä sen ennen kuin aloitin ohjelmoinnin. (Ks. liite 2.) Minulla ei ollut lähtötietoja vielä riittävästi, joten koostin aluksi luetteloon vain laitteet ja niiden positiot sekä signaalitiedot; analogiset ja digitaaliset inputit ja outputit.

Kun sain I/O-luettelon edellä mainittujen tietojen osalta valmiiksi, aloin tekemään ensimmäistä sovellusohjelmaa ja sitä kautta simulaattoriohjelmaa ruuvikuljettimen moottorin ohjaukselle. Huomasin kuitenkin heti alussa, että ohjelmaan tarvitaan laitteiden I/O-korttitiedot. Näillä korttitiedoilla tarkoitetaan laitteiden signaalikaapeleiden kytkentätietoja eli mihin mikäkin signaali viedään sovellusohjelman avulla. Selvitin lähtötietona olevasta Mikko Valkeejärven opinnäytetyöstä, että vaikka prosessisimulaattorissa ei signaaleja viedä oikeille laitteille, täytyy korttitiedot silti olla olemassa. Korttitietojen avulla luodaan ns. I/O-rajapinta, jolla sovellusohjelma kommunikoi oikean prosessin korvaavan simulaattoriohjelman kanssa (Ks. kuvio 10.).



Kuvio 10. I/O-rajapinta

Koska korttiedoilla ei siis tässä prosessisimulaattorissa kommunikoida oikeiden laitteiden kanssa, ne voitiin keksiä itse. Tähän sovimme ohjaajan kanssa loogisen ja järkevän tavan. I/O-korttieto koostuu kolmesta numerosta, jotka ovat FBC-numero, slave-numero ja offset -numero. Sovimme ohjaajan kanssa, että kaikissa laitteissa käytetään FBC-numerona numeroa 6. Slave-numerointiin sovimme sellaisen tavan, että se tapahtuu juoksevalla numeroinnilla alkaen numerosta 10. Laitteet numeroidaan PI-kaaviota katsottaessa ylhäältä alaspäin. Offset-numeron osalta sovimme niin, että se on nolla kaikissa muissa laitteissa paitsi ABB:n taajuusmuuttajissa. ABB:n taajuusmuuttajille selvitin offset -numerot ABB:n manuaalista.

Kun olin saanut I/O-luettelon valmiiksi, pääsin kunnolla jatkamaan ohjelman tekoa moottorin ohjaukselle. Pääsinkin tavoitteeseeni ja sain tehtyä sekä sovellusohjelman että simulaattoriohjelman moottorin ohjaukselle. Ohjelma näytti päällisin puolin toimivalta, mutta viantarkistuksessa ilmeni vikailmoitus, jonka vuoksi en saanut vielä testattua ohjelmaa. Moottorinohjaus-symbolin parametreissa oli ohjelmaan liittyvässä viittauksessa jokin virhe. En osannut korjata virhettä, sillä en muistanut mikä

on oikea tapa tehdä viittaus, jolla yhdistää symboli haluttuun ohjelmaan. Tämän selvittämisen seuraavalla viikolla.

3.3 Seurantaviikko 3

Maanantai 27.1.

Päiväni alkoi sillä, että jatkoin kuljettimen moottorin ohjauksen ja nopeusohjeen sovellusohjelman tekoa. Tein sekä nopeusohjeeseen että moottorin ohjaukseen jo pari lukitusta. Tässä yhteydessä tein myös pari korjausta I/O-luetteloon moottorien ja taajuusmuuttajien positioiden osalta. Iltapäivällä aloin tekemään kuljettimen nopeusohjetta varten simulaattoriohjelmaa. Se jäi kuitenkin ihan alkutekijöihin, sillä sain vihdoin kokeiltua moottorin ohjausta valvomosta käsin, kun ohjaajani saapui auttamaan minua korjaamaan aiemmin mainitsemani vikailmoituksen. Ohjaus toimi kuitenkin vain osittain, sillä sain muutettua moottorin käyntiluvan, mutta en saanut muutoksesta tietoa takaisinkytkentään. Eli sain annettua käskyn, mutta en saanut tietoa siitä, menikö käsky perille. Jatkan ongelman selvittelyä keskiviikkona, sillä huomisen käytän raportin kirjoittamiseen.

Keskiviikko 29.1.

Aloitin päiväni sillä, että jatkoin kuljettimen moottorin ohjauksessa ilmenneen ongelman selvittelyä. Tutkin siis miksi moottorille antamani käyntilupa ei mene perille eli en saa siitä tietoa takaisinkytkentään. Kun tutkin aikani tekemiäni ohjelmia, sain kun sainkin ratkaistua ongelman ja lopulta moottorin ohjaus binäärisesti alkoi toimia. Tämän myötä pystyin kokeilemaan, että tekemäni lukitus tukosvahdilta toimii myös. Koska olin tehnyt lukituksen vain sovellusohjelman puolelle, käytin iltapäivän siihen, että tein lukituksen niin, että se tulee simulaattoriohjelmalta. Sain aamupäivällä kokeiltua myös pyörintävahdilta tulevan lukituksen. Se ei toiminut vielä oikein ja sekin täytyy tehdä simulaattoriohjelmana niin, että sitä ei ohjata painikkeella kuten nyt, vaan se toimii automaattisesti oikean laitteen tavoin.

Torstai 30.1.

Aloitin päiväni sillä, että tein eilen kesken jääneelle pyörintävahdilta tulevalle lukitukselle simulaattorihjelman. Sain ohjelman valmiiksi ja sen pitäisi olla toimiva. En kuitenkaan saanut kokeiltua ohjelmaa, koska valvomoon testaamista varten jätetty painike ei suostunut toimimaan. Yritin selvittää ongelmaa koko aamupäivän, mutta se jäi kesken, koska ohjaajan apua ei ollut saatavilla. Iltaapäivän käytin siihen, että tein kuljettimen nopeusohjeen sovellusohjelmalle simulaattorihjelman. Senkin sain valmiiksi ja päällisin puolin katsottuna sen pitäisi toimia. Kun yritin syöttää nopeutta valvomoon, se ei onnistunut. Koska en saanut aamupäivällä pyörintävahdin painiketta toimimaan, se jäi off-tilaan ja siten aiheutti lukituksen ja pysäytti moottorin ohjelmani mukaisesti. En siis päässyt edes syöttämään nopeutta, koska tämä lukitus esti sen. En saanut ongelmaa itse selvitettyä, joten nopeusohjeenkin testaus jäi kesken. Selvitän sekä lukituksen että nopeusohjeen kanssa ilmenneet ongelmat ensi viikolla, sillä huomenna kirjoitan raporttia.

Viikkoanalyysi

Tällä viikolla tavoitteenani oli selvittää binäärisen moottorinohjauksen testauksessa ilmennyt valvomoon liittyvä vikailmoitus ja siten saada moottorin ohjaus toimintaan. Lisäksi tavoitteenani oli saada tehtyä myös ruuvikuljettimen nopeusohjeelle sekä sovellus- että simulaattorihjelma.

Heti maanantaina aamupäivällä minun oli tarkoitus korjata ohjaajan avustuksella moottorin ohjauksen testauksessa ilmennyt valvomon vika. En kuitenkaan saanut heti apua, joten jatkoin moottorin ohjauksen ja ruuvikuljettimen nopeusohjeen ohjelmien tekoa lisäämällä niihin jo kaksi lukitusta.

Lukitus on ehto, jonka mukaan määrittäyty tai ohjelmallisesti määritetään missä tilanteessa prosessilaitte voi olla päällä tai sitä voi ohjata. Lukitus voi olla anturin mittaus-tiedon mukaisesti muodostuva tai ohjelmallisesti muodostettava ehto prosessilaitteen käymiselle tai käynnissä olemiselle. Lukitusehto voidaan ohjelmallisesti muodostaa myös jonkin toisen prosessilaitteen käyntitiedon avulla.

Ruuvikuljettimen moottorin ohjaukselle minun piti tehdä lukitukset kahden anturin avulla. Nämä anturit ovat pyörintävahti ja tukosvahti. (Ks. liite 3. ja liite 4.) Koska en tiennyt, että tässä simulaattorissa näiden oletetaan olevan oikeita antureita, tein näille vain sovellusohjelmat ja siten ohjelmalliset kytkimet valvomoon. Sain kuitenkin pian ohjaajalta tiedon, että pyörintävahti ja tukosvahti oletetaan olevan oikeita antureita. Siten tein antureille niiden mittausta mallintavat simulaattoriohjelmat.

Tukosvahdin aiheuttama lukitus toimii niin, että jos putkistossa on tukos, moottori sammuu automaattisesti, muuten se pyörii normaalisti. Pyörintävahdin lukitus toimii siten, että vahti mittaa pyöriikö moottori, kun sille on annettu käyntilupa. Jos moottorissa on jokin vika eikä se lähde pyörimään käyntiluvasta huolimatta, täytyy tieto näkyä myös valvomossa. Myöhemmin minun pitää tehdä moottorille myös sellainen lukitus, joka estää moottoria pyörimästä, jos seuraava kolakuljetin ei pyöri.

Tehtyäni lukitukset sainkin ohjaajalta apua moottorin binäärisen ohjauksen kanssa ilmenneeseen valvomo-ongelmaan. Ohjaajani neuvoi, miten moottorinohjaus objektille tulee syöttää parametrit, jotta se yhdistyy tekemääni ohjelmaan. Tämän jälkeen sain moottorin ohjauksen toimimaan. Huomasin kuitenkin heti, että ohjelmani ei vielä toiminut aivan täydellisesti. Sain annettua käyntiluvan ja se meni perille, mutta en saanut käyntitietoa eli tietoa siitä, että tila oikeasti muuttui. Tämän sain helposti itse korjattua, sillä huomasin, että yritin syöttää sovellusohjelman takaisinkytkentään käyntilupaa output- lohkon avulla. Se ei tietenkään onnistunut, sillä käyntilupa on sovellusohjelmaan tuotuna input- tietoa. Vaihdoin output- lohkon tilalle input- lohkon ja sain moottorin binäärisen ohjauksen toimimaan täydellisesti. Tässä yhteydessä huomasin, että minun pitää varmistaa vielä myöhemmin ohjaajalta olenko tehnyt pyörintävahdin lukituksen toimintakuvauksen mukaisesti.

Kun sain moottorin binäärisen ohjauksen toimintaan, jatkoin ruuvikuljettimen nopeusohjeen sovellusohjelman ja simulaattoriohjelman tekoa. Sovellusohjelman olin oikeastaan saanut jo aiemmin tehtyä, joten aloin tekemään nopeusohjeelle simulaattoriohjelmaa. Tässä nopeusohjeessa pääsin soveltamaan ensimmäisen kerran Valmet DNA -suunnittelu ympäristön siirtofunktiorakennetta. Minun piti mallintaa nopeuden syöttö siten, että se vastaa oikean kuljettimen käyttäytymistä. Kuten ensimmäisellä

viikolla opin, tähän käytin lead lag -lohkoa ja delay-lohkoa. Tein siis ohjelman niin, että haluttu nopeus syötetään lead lag -lohkoon ja siitä delay-lohkoon, jolla saadaan aitoa tilannetta vastaava viive. Delay-lohkon ulostulosta(out) saadaan hetkellinen nopeus. Tässä vaiheessa minun ei tarvinnut tuoda hetkellistä nopeutta valvomoon, mutta sen pystyy katsomaan ohjelmasta testitilassa. Jotta sain kokeiltua, että ohjelma toimii, säädin lead lag - ja delay-lohkoon sopivat parametrit ohjaajalta saamieni lähtötietojen avulla.

3.4 Seurantaviikko 4

Maanantai 3.2.

Tein vain puolikkaan mittaisen päivän, sillä aamupäivällä minulla oli palaveri, jossa keskusteltiin muiden opiskelijoiden ja opinnäytetyön ohjaajien kanssa päiväkirjamallisen opinnäytetyön sisällöstä ja rakenteesta. Iltapäivällä jatkoin varsinaisen työni tekoa. Kirjoittelin moottorin ohjauksen simulaattoriohjelmaan kommentteja sen eri vaiheisiin. Lisäksi tein sovellusohjelman polttoainesäiliön pinnankorkeuden mittaukselle. Lopuksi sain ohjaajaltani apua pyörintävahdin lukituksessa ilmenneeseen ongelmaan. Ensiksi korjasin lukitukselle tekemiäni sovellus- ja simulaattoriohjelmaa, sillä olin tehnyt ne hieman toimintakuvauksen vastaisesti. En kuitenkaan ehtinyt saamaan apua varsinaiseen lukituksessa ilmenneeseen ongelmaan, joten jatkan tutkimista huomenna.

Tiistai 4.2.

Koska sain apua lukituksen ongelmaan vasta iltapäivällä, käytin aamupäivän raportin kirjoittamiseen. Tämän jälkeen sain lopulta ohjaajani avustuksella moottorin ohjauksen ja nopeuden säätämisen toimimaan.

Keskiviikko 5.2.

Aloitin päiväni sillä, että tutkin kuljettimelle tekemäni nopeusohjeen toimintaa tarkemmin. Syötin valvomosta kuljettimelle nopeuden ja tutkin miten tieto etenee ohjelmassa. Huomasin kuitenkin, että syöttämäni nopeus ei muuttunut kuljettimen nopeudeksi. Se jäi pelkäksi asetusarvoksi, sillä tekemäni asetusarvon skaalaus oli virheellinen ja siten arvo ei edennyt moottoria mallintavan siirtofunktion läpi oikein. Koska apua ei ollut saatavilla, nopeuden skaalaus jäi vielä kesken. Iltapäivällä tein binaarisen moottorin ohjauksen polttoainesäiliön sekoittimelle. (Ks. liite 5.) Sain myös testattua valvomosta, että tekemäni ohjelma toimi.

Torstai 6.2.

Tämän päivän käytin siihen, että tein polttoaineen syöttöjärjestelmän seuraavalle kuljettimelle sovellusohjelman ja simulaattoriohjelman sekä moottorin ohjaukselle että kuljettimen nopeusohjeelle lukituksineen. Huomisen käytän raportin kirjoittamiseen.

Viikkoanalyysi

Tämän viikon tavoitteena oli tehdä ruuvikuljettimen moottorille pyörintävahdin lukitus oikein toimintakuvauksen mukaisesti. Lisäksi tavoitteena oli tutkia tarkemmin ruuvikuljettimen nopeusohjeen toimintaa. Tämän jälkeen oli tarkoitus tehdä jo seuraavan kolakuljettimen ohjaukselle ja nopeusohjeelle sekä sovellus- että simulaattoriohjelmat lukituksineen.

Tämä viikko oli tarkoitus aloittaa maanantaina sillä, että tarkistan ohjaajalta, olenko tehnyt pyörintävahdin lukituksen oikein. En kuitenkaan saanut apua heti, joten tein jo sovellusohjelman polttoainesäiliön pinnankorkeuden mittaukselle. Tämä oli helppo tehdä automaatiojärjestelmät-kurssilta saamieni oppien avulla. Lähtötietojen puutteen vuoksi en voinut tehdä mittaukselle vielä simulaattoriohjelmaa.

Tämän jälkeen sainkin ohjaajan tarkistamaan pyörintävahdin viiveen. En ollut tehnyt lukitusta ihan toimintakuvauksen mukaisesti. Olin ymmärtänyt toimintakuvauksessa määritetyn viiden sekunnin viiveen väärin. Olin tehnyt sen niin, että se ei ollut viive vaan pulssi. Käytännössä pulssi toimi niin, että pyörintävahdilta piti tulla pyörintätieto viiden sekunnin sisällä siitä, kun moottori yritettiin käynnistää. Kun viisi sekuntia oli kulunut, pyörintävahdin pyörintätiedon muuttaminen testauspainikkeella ei vaikuttanut moottorin käyntitilaan lainkaan. Tämä minun piti korjata. Minun piti tehdä lukitus niin, että pyörintätieto tuli viiden sekunnin kuluttua moottorin käynnistämisestä. Näin pyörintätilan muuttaminen testauspainikkeella vaikutti moottorin käyntitilaan, kun viisi sekuntia oli kulunut. Sain siis pyörintävahdin toimimaan oikein.

Kun pyörintävahti oli korjattu, aloin tutkia tarkemmin ruuvikuljettimelle tekemääni nopeusohjetta. Olin katsonut mallia skaalaukseen Valkeejärven opinnäytetyön Function Block CAD -kuvista. Huomasin, että tämä skaalaus ei minun nopeusohjeessani toiminut kuten toimintakuvauksessa määritettiin. Nopeus piti syöttää prosentteina 0-100 %, ja hetkellinen nopeus piti näkyä samassa yksikössä ja kokoluokassa. Tekemäni skaalaus muutti asetusarvon väärän suuruiseksi ja desimaaliluvuksi. Nopeus ei myöskään muuttunut hetkelliseksi nopeudeksi vaan sain muutettua vain asetusarvoa. Nämä ongelmat selvitän seuraavalla viikolla.

Tämän viikon lopussa ehdin vielä tehdä seuraavalle kolakuljettimelle sekä sovellus- että simulaattorihjelmat kuljettimen moottorin ohjaukselle ja kuljettimen nopeusohjeelle lukituksineen. Tämä oli helppo tehdä edellisen kuljettimen ohjelmaa mallina käyttäen.

3.5 Seurantaviikko 5

Tiistai 12.2. ja keskiviikko 13.2.

Käytin maanantain raportin kirjoittamiseen, joten jatkoin simulaattorin ohjelmointia vasta tänään. Huomasin aamulla, että oikein tehdyistä ohjelmistani huolimatta valvomo ei kommunikoinut niiden kanssa. Koska en voinut kokeilla tekemiäni ohjelmia, käytin aamupäivän raportin kirjoittamiseen ja iltapäivällä sain ohjaajaltani apua sekä

valvomo-ongelmaan että siihen, miten nopeuden skaalaus tehdään oikein. Itse asiassa tässä simulaattorissa nopeudelle ei tarvinnutkaan tehdä skaalausta lainkaan. Minun piti vain tehdä sen kokoluokalle rajoitus välille 0-100 %. Koska sain ohjeen vasta iltapäivällä, tein tämän muutoksen nopeusohjeisiin vasta tänään keskiviikkona.

Torstai 13.2.

Tämän päivän aikana tein prosessin lopuille kuljettimille ohjelmat. Tein siis kahden ruuvikuljettimen moottorin binääriselle ohjaukselle sekä sovellus- että simulaattoriohjelman. Sain myös testattua ohjelmat ja ne toimivat. Koska en saanut vielä lähtötietoja riittävästi, tein vain binääriset ohjaukset, joihin sain suoraan mallia aiempien kuljettimien ohjelmista. Huomisen käytän raportin kirjoittamiseen ja ensi viikolla jatkan alkuviikosta moottorien ohjausten tekoa sen mukaan, miten saan lähtötietoja.

Viikkoanalyysi

Tämän viikon tavoitteena oli korjata ruuvikuljettimen ja siten myös kolakuljettimen nopeusohjeen skaalaus oikeanlaiseksi. Lisäksi tavoitteena oli tehdä lopuille kahdelle ruuvikuljettimelle sovellus- ja simulaattoriohjelmat moottorin binääristä ohjausta varten.

Aloitin viikkoni sillä, että tutkin ohjaajani kanssa, miten nopeusohjeen skaalaus tehdään oikein. Minulle selvisi, että olin tehnyt skaalauksen niin, että prosentteina annettu nopeus muunnettiin hertseiksi. Hertseistä nopeuden olisi voinut skaalata yksiköön kierrosta/ minuutti eli RPM. Minun työssäni tätä skaalausta ei tarvinnut tehdä ollenkaan, koska nopeus syötetään prosentteina ja näytetään prosentteina. Minun tarvitsi tehdä ohjelmaan rajaus, joka määrittä nopeuden kokoluokaksi 0-100 %. Tämän jälkeen nopeus syötettiin siirtofunktion läpi hetkelliseksi nopeudeksi. (Ks. kuvio 11.)

Torstai 20.2.

Tämän päivän käytin siihen, että tein sulkusyöttimien binäärisille moottorin ohjauksille sekä sovellus- että simulaattoriohjelmat lukituksineen. Lisäksi tein aiemmin tehdyn kolakuljettimen ja kahden ruuvikuljettimen ohjaukseen loput lukitukset. Lopuksi tein kolakuljettimen ja ensimmäisen ruuvikuljettimen nopeusohjeisiin rajoitukset nopeuden ylärajalle.

Perjantai 21.2.

Tämän päivän käytin siihen, että tein sulkupeltien ohjauksille sekä sovellus- että simulaattoriohjelmat lukituksineen. Lisäksi lisäsin eilen tehtyihin sulkusyöttimien ohjauksiin pyörintävahtien lukitukset.

Viikkoanalyysi

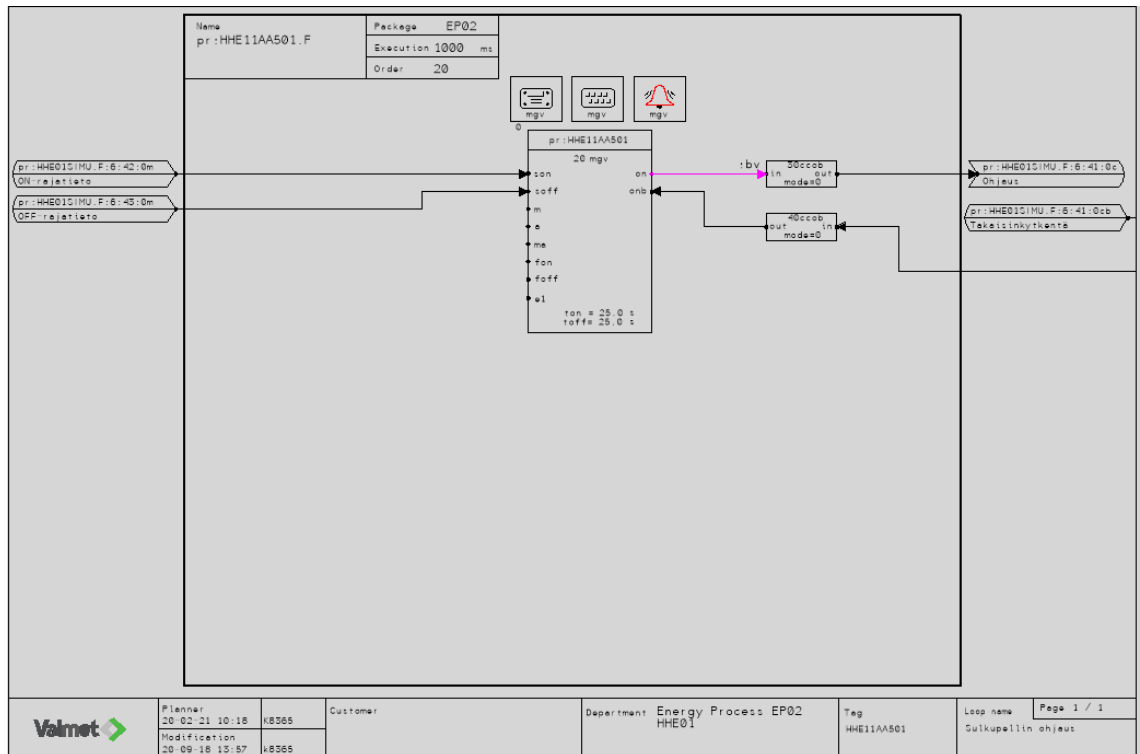
Tämän viikon tavoitteena oli tehdä kahden syöttöruuvikuljettimen nopeusohjeille sovellus- ja simulaattoriohjelmat. Lisäksi tavoitteena oli tehdä sekä sovellus- että simulaattoriohjelmat kahdelle sulkusyöttimelle ja kahdelle sulkupellille.

Tämän viikon aloitin siis sillä, että jatkoin viime viikolla tekemiäni syöttöruuvikuljettimien ohjelmia. Tein syöttöruuvien nopeusohjeille sovellus- ja simulaattoriohjelmat. Nämä minun oli jälleen helppo tehdä edellisten kuljettimien ohjelmia mallina käyttäen. Myös syöttöruuveille nopeusohjeen simulaattoriohjelma tehtiin aiempien kuljettimien tavoin siirtofunktorakenteella lead lag -lohkoa ja delay-lohkoa käyttäen. Lisäksi tein nopeusohjeille tarvittavat lukitukset. Kuljettimien nopeusohjeisiin liittyen sain tällä viikolla vaatimusmäärittelyyn ohjeistuksen, jossa ohjaaja haluaa, että jo aiemmin tekemäni purkuruuvin ja kolakuljettimen nopeudet täytyy rajoittaa. Purkuruuvin nopeusohjeeseen minun täytyi tehdä sellainen ehto, että ruuvin nopeus voi maksimissaan olla yhtä suuri kuin kolakuljettimen nopeus. Kolakuljettimen nopeusohjeeseen minun täytyi tehdä sellainen ehto, että kuljettimen nopeus voi maksimissaan olla puolet syöttöruuvien yhteenlasketusta nopeudesta. Rajoitukset täytyi tehdä

siksi, että kolakuljetin ja purkuruuvit eivät voi ottaa vastaan polttoainetta enempää, kuin mikä niiden oma kapasiteetti on.

Seuraavaksi tein sovellus- ja simulaattorihjelmat prosessin kahdelle viimeiselle kuljettimelle eli sulkusyöttimille. (Ks. liite 6.) Myös sulkusyöttimet ovat aiempien kuljettimien tavoin moottorihjattavia. Ne eroavat kuitenkin hieman aiemmista kuljettimista, sillä ne pyörivät aina vakionopeudella, kun ne laitetaan päälle. Toisin sanoen niille ei tarvitse tehdä nopeusohjetta. Ohjaus tapahtuu binäärisesti eli ne joko pyörivät tai eivät pyöri. Minun oli jälleen helppo tehdä tämä binäärinen moottorinohjaus aiempia ohjelmia mallina käyttäen.

Viikon viimeisenä työnä tein sovellus- ja simulaattorihjelmat prosessin viimeisille komponenteille eli sulkupelleille. (Ks. liite 7.) Sulkupeltejä ohjataan on/off-magneettiventtiileillä. Pystyin jälleen käyttämään tämän ohjauksen tekemiseen aiempia ohjelmia mallina. Tämä onnistui siksi, että magneettiventtiilin ohjauksessa on käytännössä samanlainen rakenne kuin binäärisessä moottorinohjauksessa. Sovellusohjelman puolella minun piti vain käyttää moottorinohjauslohkon sijaan magneettiventtiilin ohjaukseen tarkoitettua mgv-lohkoa. Lisäksi tilatietoja on magneettiventtiilin ohjauksessa kaksi, ON-rajatieto ja OFF-rajatieto. (Ks. kuvio 12.)



Kuvio 12. Magneettiventtiin(sulkupelti) ohjaus

3.7 Seurantaviikko 7

Tiistai 25.2. ja keskiviikko 26.2.

Käytin tiistain ja keskiviikon siihen, että tein polttoaineen syöttöjärjestelmään kaikki virtausmittaukset sovellus- ja simulaattorihjelmineen. Sain ne myös toimintaan. Otin huomioon sen, että kolakuljettimen ja syöttöruuvien jälkeen virtausmittauksissa voi olla arvot vain, jos purkuruuvi pyörii jollain nopeudella eli polttoaine kulkee prosessissa.

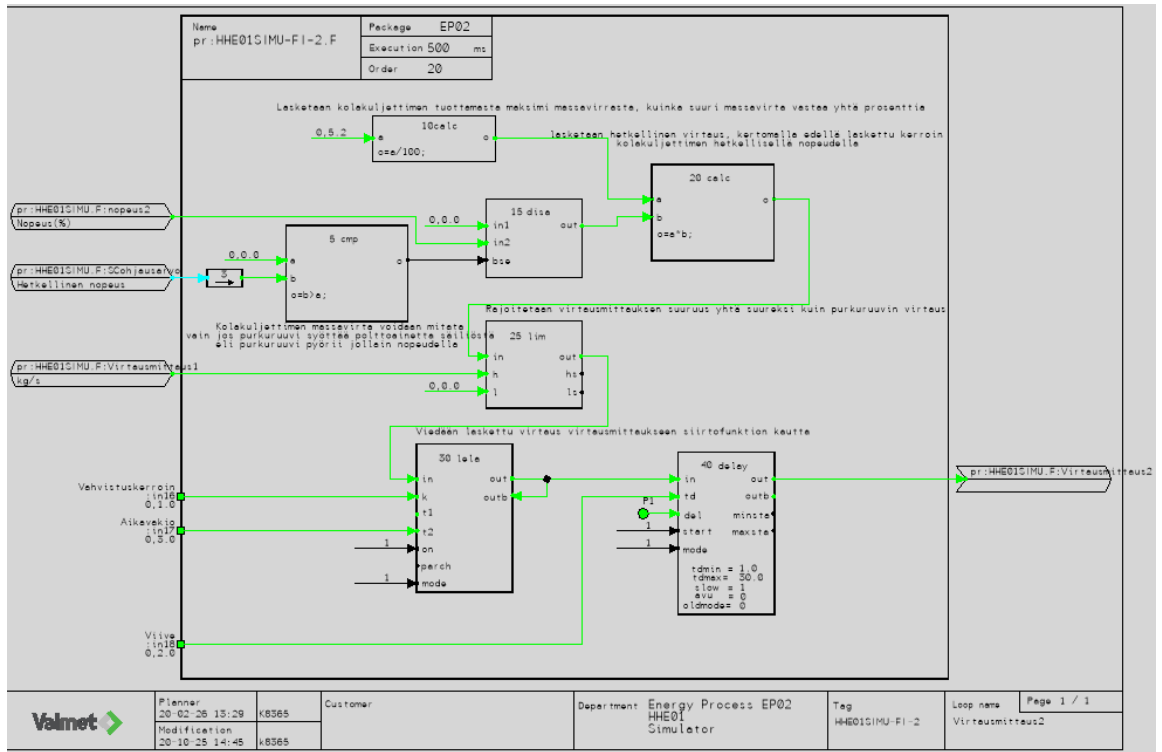
Torstai 27.2.

Käytin tämän päivän siihen, että tein ohjelman ja valvomo-objektit polttoaineen poltotehon laskemiseksi. Ohjelma näytti päällisin puolin toimivalta, mutta en saanut sitä vielä toimintaan. Ongelman ratkaisu siirtyy ensi viikkoon.

Viikkoanalyysi

Tämän viikon tavoitteena oli tehdä prosessiin tarvittavat virtausmittaukset sovellus- ja simulaattorihjelmineen. Lisäksi tavoitteena oli tehdä vaatimusmäärittelyssä vaadittu prosessin polttotehon laskenta.

Aloitin siis viikon tekemällä virtausmittaukset. Minun piti tehdä prosessiin jokaisen kuljettimen jälkeen virtausmittaus, jotta valvomosta voidaan prosessia ajettaessa nähdä, että se oikeasti käy ja siellä tavara liikkuu. Koska kyseessä on kiinteän polttoaineen kuljetus, virtausmittaukset näyttävät polttoaineen massavirran yksikössä kg/s. Virtausmittaukset minun piti mallintaa simulaattorihjelmaan laskennan avulla. Mitoitustiedoissa minulle annettiin jokaiselle kuljettimelle maksimikapasiteetti eli massavirta, jonka kuljetin kykenee maksiminopeudellaan (100 %) tuottamaan. Tämän massavirran avulla pystyin tekemään simulaattorihjelmaan laskennan. Ohjelmassa lasketaan ensin mikä on massavirta yhden prosentin nopeudella eli jaetaan maksimikapasiteetti 100 %:lla. Seuraavaksi tällä saadulla kertoimella kerrotaan kuljettimen hetkellinen nopeus. Näin saatu hetkellinen massavirta viedään virtausmittaukseen siirtofunktorakenteen kautta. Sain virtausmittaukset toimimaan, mutta huomasin testauksen yhteydessä, että minun pitää lisätä toimintaehto purkuruuvien jälkeistä mittausta lukuun ottamatta kaikkiin muihin mittauksiin. Minun piti tehdä kolakuljettimen ja syöttöruuvien jälkeisiin mittauksiin sellainen ehto, että virtausmittaukseen saadaan tieto vain, jos prosessissa kulkee polttoaine. Käytännössä, jos purkuruuvien jälkeisessä virtausmittauksessa näkyy hetkellinen massavirta, polttoaine kulkee. Siten muut massavirrat voidaan "mitata" eli simulaattorissa voidaan suorittaa massavirran laskenta. (Ks. kuvio 13.)



Kuvio 13. Virtausmittauksen simulointi

Viikon toisena työnä tein vaativuusmäärittelyssä vaaditun prosessin polttotehon laskennan. Polttoteho lasketaan alla olevan vaatimusmäärittelyssä annetun kaavan mukaisesti.

$$Teho(MW) = LA * qm, missä$$

LA= Lämpöarvo (MJ/kg), alkuarvona 19.3 MJ/kg, qm= hetkellinen massavirta.

Lämpöarvoksi voidaan maksimissaan syöttää 20.6 MJ/kg (Alakangas, E. Hurskainen, M. Korhonen, J. Laatikainen-Luntama, J. 2016, 64). Tein laskennalle sovellusohjelman ja valvomo-objektit. Tein valvomoon kaksi kenttää, joista toiseen voidaan syöttää haluttu lämpöarvo ja toiseen saadaan laskennasta polttoteho. En kuitenkaan saanut vielä ohjelmaa toimintaan, mutta selvitän ongelman ensi viikolla.

3.8 Seurantaviikko 8

Tiistai 3.3., keskiviikko 4.3. ja torstai 5.3.

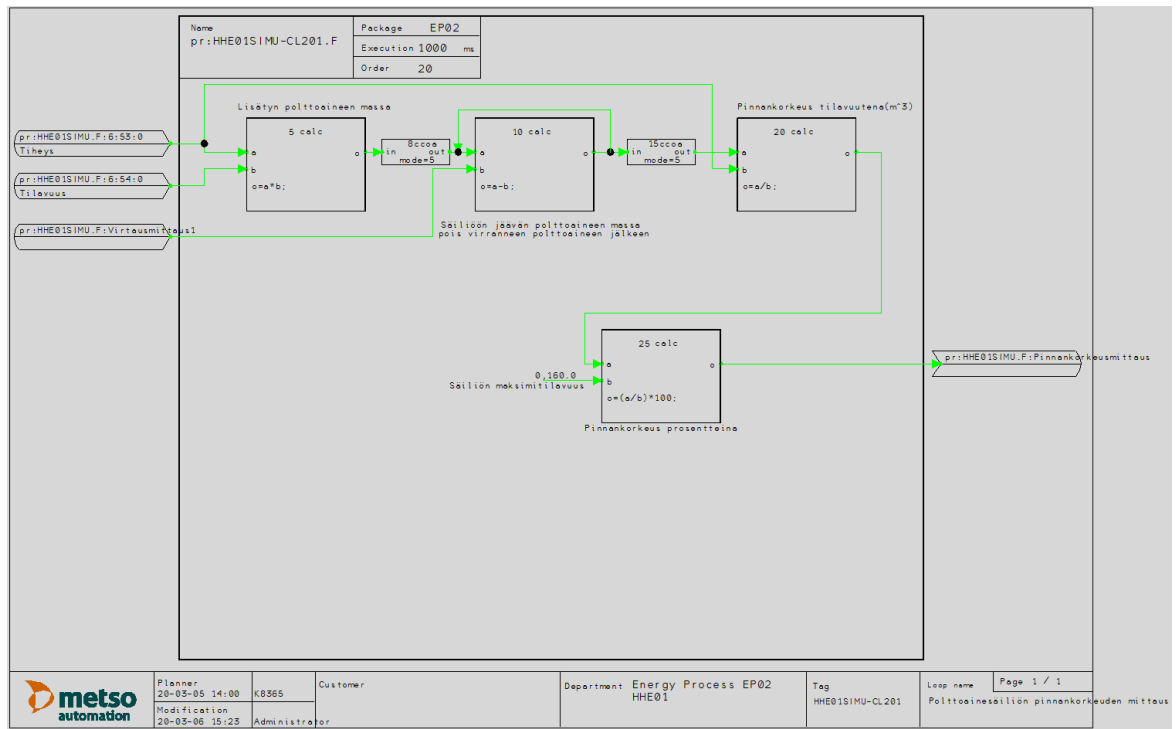
Tiistain, keskiviikon ja torstain aikana sain polttotehon laskennan ohjelmat toimintakuntoon. Lisäksi sain myös polttoainesäiliön pinnankorkeusmittauksen ohjelmat toimintaan.

Viikkoanalyysi

Tämän viikon tavoitteena oli saada viime viikolla tehty polttotehon laskenta toimintaan. Lisäksi tavoitteena oli saada tehtyä simulaattoriohjelma ja saada sitä kautta toimintaan polttoainesäiliön pinnankorkeusmittaus.

Aloitin viikon sillä, että selvitin polttotehon laskennan testauksessa ilmenneen ongelman. Huomasin ilman ohjaajan apua, että olin asettanut parametrit laskentaa varten tekemiini valvomo-objekteihin virheellisesti. Korjattuani parametrit, laskenta alkoi toimia täydellisesti.

Viikon toisena työnä aloin suunnitella mallinnusta polttoainesäiliön pinnankorkeuden mittaukselle. Lähtötiedoissa minulle annettiin polttoainesäiliön maksimikapasiteetti 160 m^3 ja polttoaineen tiheyden alkuarvo 406 kg/m^3 . Polttoaineen tiheydeksi voidaan maksimissaan syöttää 500 kg/m^3 (Alakangas ym. 2016, 65). Näiden sekä aiemmin tekemäni purkuruuvien aikaan saaman massavirran avulla minun piti suunnitella laskenta säiliön pinnankorkeuden mallintamiseksi. Sainkin hahmoteltua paperille laskennan, jolla sain laskettua säiliössä olevan polttoaineen määrän. En kuitenkaan vielä hahmottanut, miten saan tehtyä laskennan siten, että pinnankorkeus muuttuu reaaliajassa sitä mukaan, kun polttoainetta otetaan säiliöstä purkuruuvilla. Lopulta ohjaajan avustuksella sain tehtyä laskennan täydelliseksi. Tein Function Block CAD -työkälulla kuvion 14 mukaisen simulaattoriohjelman laskennalle.



Kuvio 14. Pinnankorkeuden laskenta funktion block cadissa

Ensimmäisessä lohossa lasketaan säiliössä olevan polttoaineen massa kilogrammoina kaavalla $\text{tiheys} * \text{tilavuus}$. Tiheyden alkuarvona on edellä mainittu 406 kg/m^3 ja tilavuuden alkuarvona säiliön maksimikapasiteetti 160 m^3 . Seuraavassa lohossa lasketaan vähennyslaskulla säiliöön jäävän polttoaineen määrä (kg), kun purkuruuvi on poistanut ainetta. Jotta polttoaineen määrä saadaan muuttumaan reaaliajassa, tehdään lohkoon jatkuva silmukka. Näin polttoaineen massasta voidaan vähentää eri yksikössä olevaa massavirtaa. Jatkuva silmukka vähentää säiliöstä ohjelman suoritusvälin eli sekunnin välein massavirran (kg/s) mukaisen kilogrammamäärän polttoainetta. Lopuksi skaalataan pinnankorkeus alkuperäiseen mittausyksikköön kuutiometriin. Koska valvomossa korkeus ilmoitetaan prosentteina, skaalataan kuutiometrit vielä prosenteiksi.

Näillä laskuilla sain pinnankorkeuden mittauksen toimimaan oikein. Ohjelmaa ja valvomoa pitää vielä muokata niin, että valvomosta voidaan lisätä säiliöön polttoainetta ja aineelle voidaan antaa haluttu tiheys.

3.9 Seurantaviikko 9

Maanantai 9.3.

Muokkasin pinnankorkeuden mittausta niin, että valvomosta voidaan lisätä säiliöön polttoainetta ja aineelle voidaan antaa tiheys. Sain mittauksen myös toimintaan näiden ominaisuuksien kanssa.

Tiistai 10.3.

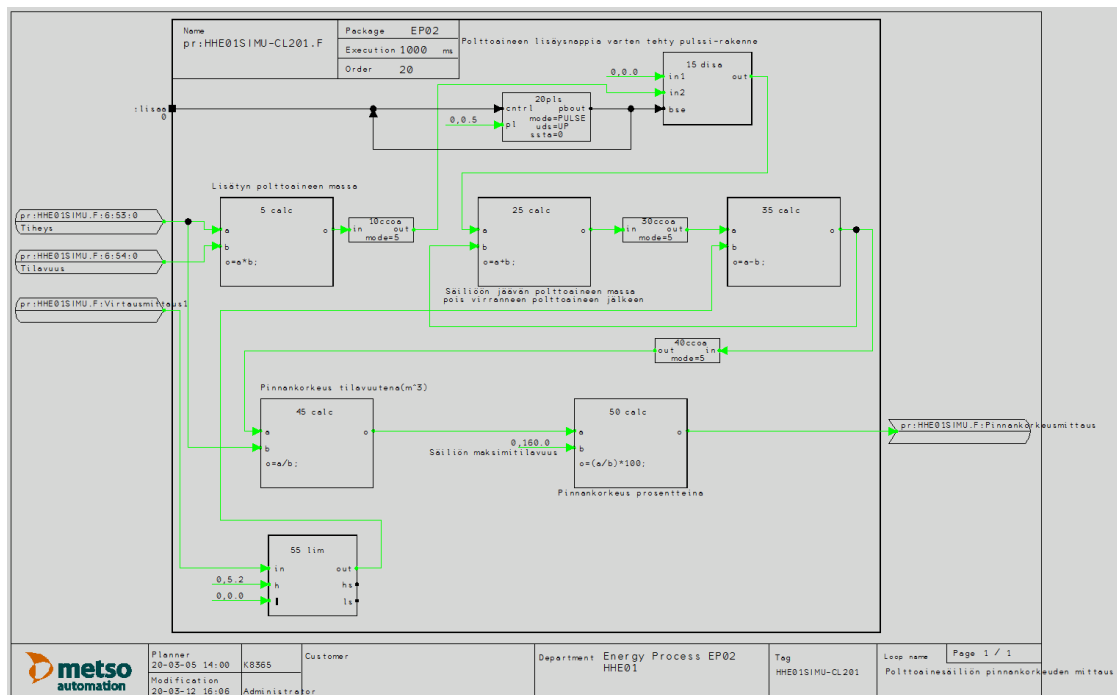
Tein kolakuljettimen ja syöttöruuvien aikaan saamille massavirroille rajoitukset: kolakuljettimen aikaansaama massavirta ei voi olla suurempi kuin purkuruuvien aiheuttama massavirta ja syöttöruuvien aikaan saama yhteenlaskettu massavirta ei voi olla suurempi kuin kolakuljettimen aikaansaama massavirta. Tein myös säiliön pinnankorkeuskytkimelle sovellus- ja simulaattoriohjelman, mutta en saanut kytkimen aiheuttamia lukituksia toimintaan. Tämä johtuu siitä, että kytkimelle tarvitsee todennäköisesti tehdä kaksi I/O-tietoa; ylempi yläraja ja alempi alaraja. Tämän varmistan myöhemmin ohjaajaltani.

Viikkoanalyysi

Viikon tavoitteena oli muokata säiliön pinnankorkeuden mittauksen simulaattoriohjelma sellaiseksi, että säiliöön voidaan tarvittaessa lisätä polttoainetta. Lisäksi tavoitteena oli muokata valvomoa niin, että sieltä voidaan tarvittaessa lisätä polttoainetta säiliöön.

Aloitin siis viikkoni sillä, että rupesin tutkimaan viime viikolla tekemääni pinnankorkeusmittauksen simulaattoriohjelmaa. Mietin, miten minun pitää muokata ohjelmaa, jotta siitä saa sellaisen, että säiliöön voidaan tarvittaessa lisätä painonapilla polttoainetta. Pienellä ohjaajan avustuksella hahmotin, mitä minun täytyy ohjelmalle tehdä. Ensimmäisenä minun piti muokata jatkuvaa silmukkaa, jolla mahdollistin pinnankorkeuden reaaliaikaisen muuttumisen. Jatkuvassa silmukassa oli se ongelma, että se ei ottanut huomioon polttoaineen mahdollista lisäämistä. Käytännössä se siis vähensi

massavirran aikaan saaman muutoksen pinnankorkeudesta, jonka pystyi itse syöttämään valvomosta. Pinnankorkeus oli siis tässä vaiheessa ikään kuin vain lisätyn polttoaineen määrä. (Ks. kuvio 14.) Minun piti muokata jatkuva silmukka sellaiseksi, että se ei vain vähennä massavirran aikaan saamaa muutosta pinnankorkeudesta, vaan laskee reaaliajassa hetkellisen pinnankorkeuden ottaen huomioon mahdollisen polttoaineen lisäämisen. Tämä tapahtui siten, että lisäsin jatkuvaan silmukkaan ennen massavirran vähennystä lohkon, joka lisää lisäyspainiketta painamalla halutun määrän polttoainetta pinnankorkeuteen. Tämän jälkeen rakenne vähentää pinnankorkeudesta mahdollisen massavirran aikaan saaman muutoksen. Lopuksi rakenne palauttaa näin lasketun pinnankorkeuden hetkelliseksi pinnankorkeudeksi aiemmin mainittuun polttoaineen lisäyslohkoon. Jotta lisäyslohkolla voidaan vain tarvittaessa lisätä polttoainetta, minun piti tehdä vielä lisäyspainike tätä lisäämistä varten. Tein lisäyspainiketta varten pulssirakenteen, jolla lisättävä polttoaineen määrä viedään pulssina lisäyslohkolle vain silloin, kun lisäyspainiketta painetaan. (Ks. kuvio 15.)



Kuvio 15. Polttoainesäiliön pinnankorkeusmittaus polttoaineen lisäystoiminnon lisäyksen jälkeen.

Samassa yhteydessä, kun sain ohjeita tuohon pinnankorkeusmittauksen muokkaamiseen, ohjaajani antoi minulle lisää vaatimusmäärittelyjä kuljettimien toimintaan liittyen. Minun piti tehdä kuljettimien tuottamiin kapasiteetteihin eli massavirtoihin rajoituksia, jotta ne ovat mahdollisimman realistisia. Minun piti tehdä massavirroille sellaiset rajoitukset, että kolakuljettimen aikaansaama massavirta ei voi olla suurempi kuin purkuruuvin aikaansaama massavirta. Myös syöttöruuvien aikaansaamat massavirrat minun piti rajoittaa. Tämä minun piti tehdä siten, että syöttöruuvien aikaansaama yhteenlaskettu massavirta ei voi olla suurempi kuin kolakuljettimen aikaansaama massavirta. Näin sain kapasiteetit eli massavirrat realistisiksi.

Viikon viimeisenä työnä aloin tekemään polttoainesäiliön pinnankorkeuskytkimelle sovellus- ja simulaattorihjelmaa. Sain kyllä tehtyä toimivat ohjelmat, mutta havaitsin kuitenkin yhden ongelman. Pinnankorkeuskytkin toimii nimittäin siten, että se aktivoituu, jos pinnankorkeuden ylempi yläraja saavutetaan tai jos taas korkeuden alempi alaraja saavutetaan. Tekemäni ohjelmat toimivat nyt siten, että kytkimeltä saadaan yksi tilatieto. Kytkimen tila on aktiivinen, jos kumpi tahansa yllä mainitusta kahdesta eri ehdosta täyttyy. Tämän yhden tilatiedon perusteella minun pitäisi kuitenkin tehdä toimintakuvauksen mukaisesti kaksi eri lukitusta. Jos säiliön pinnankorkeuden ylempi yläraja saavutetaan, säiliön polttoaineen sekoitin ei voi käydä. Jos taas säiliön pinnankorkeuden alempi alaraja saavutetaan, säiliön purkuruuvi ei voi käydä. Tulin siihen tulokseen, että mainitsemani yhdellä tilatiedolla ei voi tehdä kahta eri lukitusta. Arvelen, että tarvitsen kytkimeltä kaksi I/O-tietoa, jotta saan oman tilatiedon siitä, jos ylempi alaraja saavutetaan ja oman tilatiedon siitä, jos alempi alaraja saavutetaan. Tämä varmistuu myöhemmin, kun ohjaajani on tavoitettavissa.

3.10 Seurantaviikko 10

Keskiviikko 12.8. ja torstai 13.8.

Keskiviikon käytin siihen, että tein valmiiksi simulaattori- ja sovellusohjelman polttoainesäiliön pinnankorkeuden ylärajalukitukselle ja alarajalukitukselle. Sain lukitukset toimintaan edellä mainittujen ohjelmien osalta, mutta valvomossa ilmeni testauksen

yhteydessä ongelma. Pinnankorkeuden ylärajakytkintä varten tehty objekti ei suostunut toimimaan, vaikka sitä varten tehdyt ohjelmat toimivat. Tämän ongelman selvittelyyn menikin torstaipäivä ja en silti saanut objektia vielä toimintaan. Jatkan ongelman selvittelyä myöhemmin ohjaajan kanssa.

Keskiviikko 19.8.

Tämän päivän käytin siihen, että aloin suunnittelemaan sovellusohjelmaa prosessin viimeiselle toiminnolle, syöttötaskun pintasäädölle. Tämän pintasäädön tarkoituksena on pitää syöttötaskussa sopivaa pinnankorkeutta kolakuljettimen massavirtaa rajoittamalla. Sain tehtyä Function Block CAD -työkalulla hahmotelmat pintasäätöä varten tarvittavan säätöpiirin sovellusohjelmalle sekä syöttötaskun pinnankorkeuden mittauksen sovellusohjelmalle.

Tiistai 25.8.

Tämän päivän käytin siihen, että hahmottelin Function Block CAD -työkalulla säätöpiiriä varten tarvittavan pinnankorkeusmittauksen simulaattoriohjelman. En kuitenkaan saanut simulaattoriohjelmaa vielä toimimaan, sillä se näytti pinnankorkeudeksi negatiivista lukua.

Keskiviikko 2.9.

Tänään sain pienellä ohjeistuksella säätöpiiriä varten tarvittavan pinnankorkeusmittauksen simulaattoriohjelman toimintakuntoon. Ohjelma oli tehty muuten täysin oikein, mutta pinnankorkeuden mittaus piti vielä rajoittaa välille 0-100 %. Näin mittaus alkoi toimia ja näyttää minimissään nollaa eikä negatiivista lukemaa. Tässä yhteydessä jouduin muokkaamaan syöttöruuvien massavirtauksia. Massavirrat piti rajoittaa siten, että niiden summa ei voi olla suurempi kuin kolakuljettimen massavirta, jos

syöttötaskussa ei ole pintaa. Muuten syöttöruuvien massavirrat ovat ruuvien pyörimisnopeuksien mukaiset. Lopulta sain myös säätöpiirin toimintakuntoon.

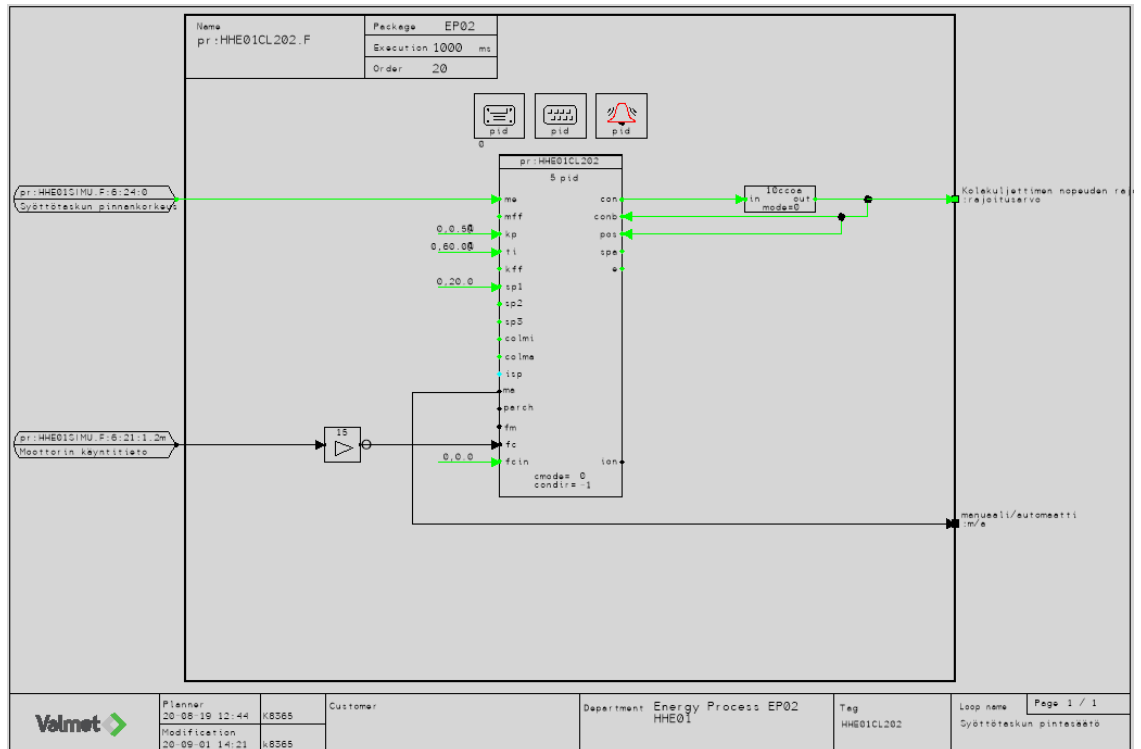
Viikkoanalyysi

Tämän viikon tavoitteena oli saada tehtyä valmiiksi sovellus- ja simulaattorihjelma polttoainesäiliön pinnankorkeuden yläraja- ja alarajalukitukselle. Lisäksi tavoitteena oli ohjelmoida prosessin viimeinen toiminto, syöttötaskun pintasäätö.

Aloitin viikon sillä, että tein loppuun polttoainesäiliön pinnankorkeuden ylä- ja alarajalukitukset. Ylärajalukitus piti tehdä pinnankorkeuden rajakytkimen tilatiedon avulla. Jos tilatieto on 1, polttoainesäiliön sekoitin ei saa pyöriä. Vielä viime viikolla luulin, että myös alarajalukitus pitää tehdä rajakytkimen avulla. En kuitenkaan ollut varma ajatuksestani, joten selvitin asian tällä viikolla. Sain pienen ohjeen alarajalukituksen tekoon ja se pitikin tehdä huomattavasti yksinkertaisemmin eli pinnankorkeuden mittauksen avulla. Jos pinnankorkeus on pienempi tai yhtä suuri kuin toimintakuvauksessa annettu lukitusraja eli 5 %, polttoaineen purkuruuvi ei saa pyöriä. Näin sain ylä- ja alarajalukituksen sovellus- ja simulaattorihjelmat toimintakuntoon.

Loppuviikon käytin siihen, että tein sovellus- ja simulaattorihjelmat prosessin viimeiselle toiminnolle, syöttötaskun pintasäädölle. Syöttötaskun pintasäädön on tarkoitus toimia siten, että se pitää sopivan määrän polttoainetta syöttötaskussa eli välisäiliössä ennen syöttöruuveja. Polttoaineen määrä eli syöttötaskun pinnankorkeus pidetään sopivana siten, että rajoitetaan kolakuljettimen pyörimisnopeutta ja siten kolakuljettimelta tulevaa massavirtaa. Kun säädin on automaattilla, sille annetaan asetusarvoksi syöttötaskun pinnankorkeus (0-100 %). Kun säädin on manuaalilla, sille annetaan suoraan ohjausarvo eli rajoitusarvo (0-100 %) kolakuljettimen nopeudelle. Aloiti-

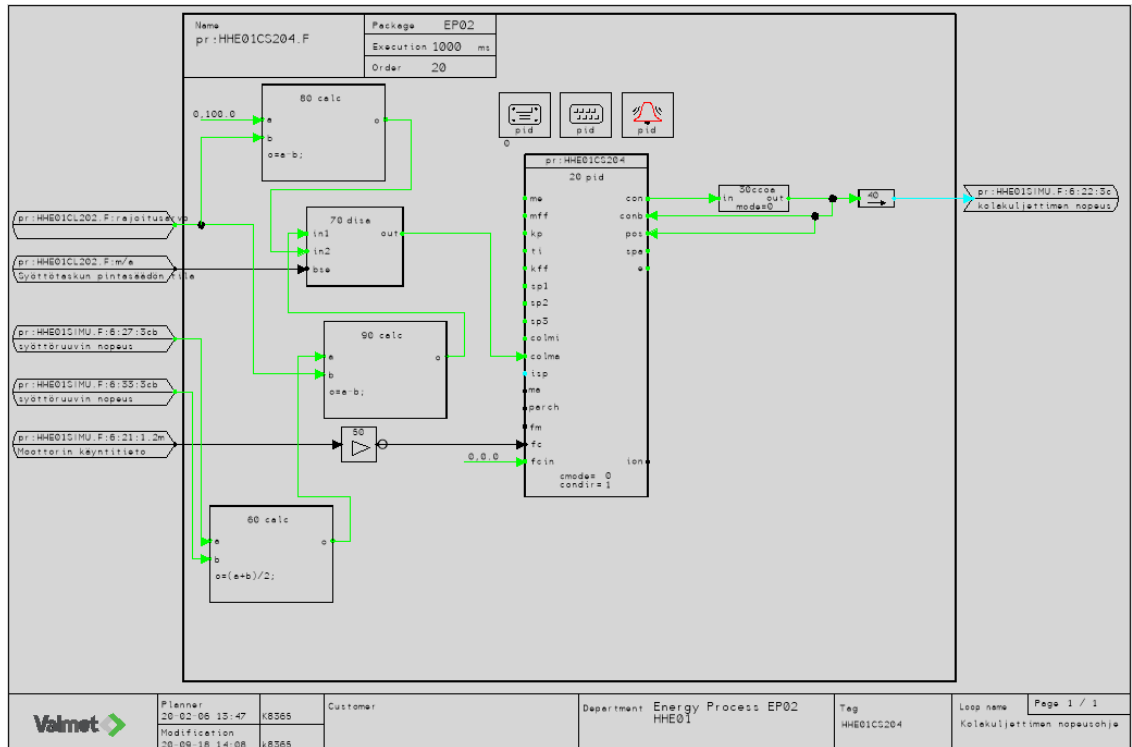
tin pintasäädön tekemisen sillä, että tein sille tarvittavat sovellusohjelmat eli syöttötaskun pinnankorkeusmittauksen ja säätöpiirin. Pinnankorkeusmittauksen toteutin analogisella mittauksella ja säätöpiirin toteutin kuvion 16 mukaisella PID-säätimellä.



Kuvio 16. Syöttötaskun pintasäädön säätöpiiri

PID-säätimen mittaustietona toimii syöttötaskun pinnankorkeus. Säätimen laskema lähtöarvo on rajoitusarvo kolakuljettimen nopeudelle. Rajoitusarvoa ei viedä suoraan kolakuljettimen maksiminopeudeksi vaan arvo vähennetään 100 %:sta. Näin saatu arvo viedään kolakuljettimen maksiminopeudeksi säätimen ollessa automaattimoodissa. Jos säädin on manuaalilla, maksiminopeus määräytyy jo aiemmin tehdyn rajoi-

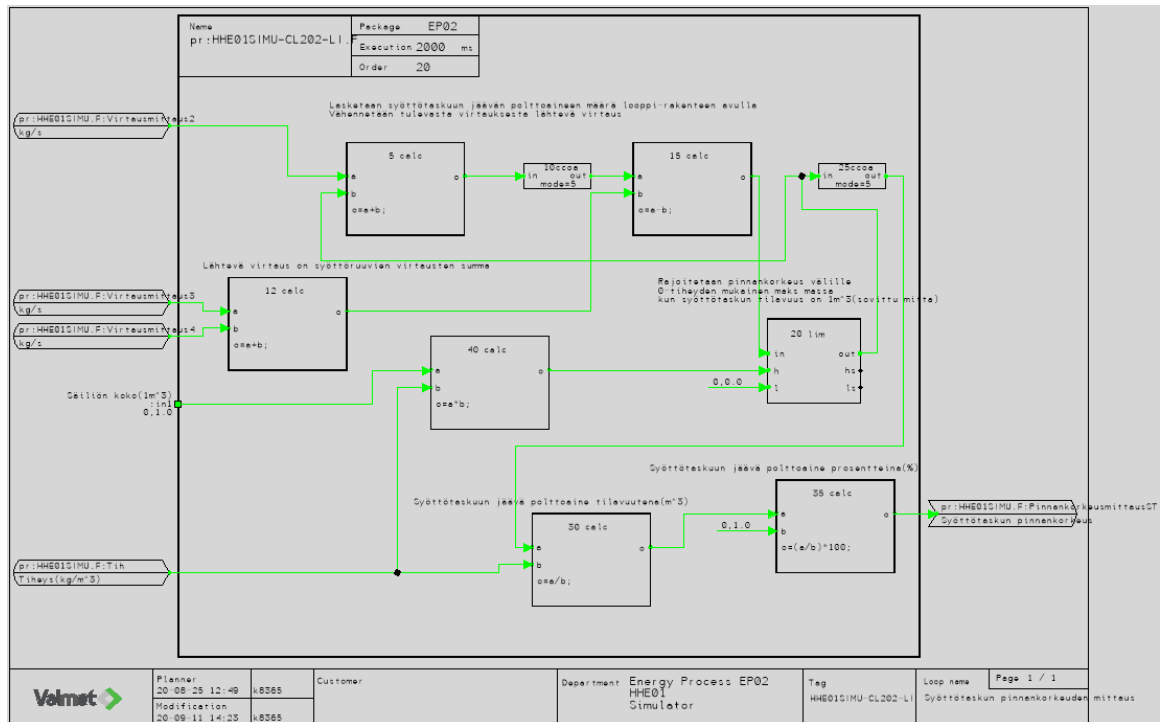
tuksen mukaan eli syöttöruuvien nopeuksien summa jaettuna kahdella. Jos manuaalimoodissa annetaan kolakuljettimelle rajoitusarvo, se vähennetään edellä mainitusta rajoituksesta. (Ks. kuvio 17.)



Kuvio 17. Kolakuljettimen nopeusohje (sovellusohjelma)

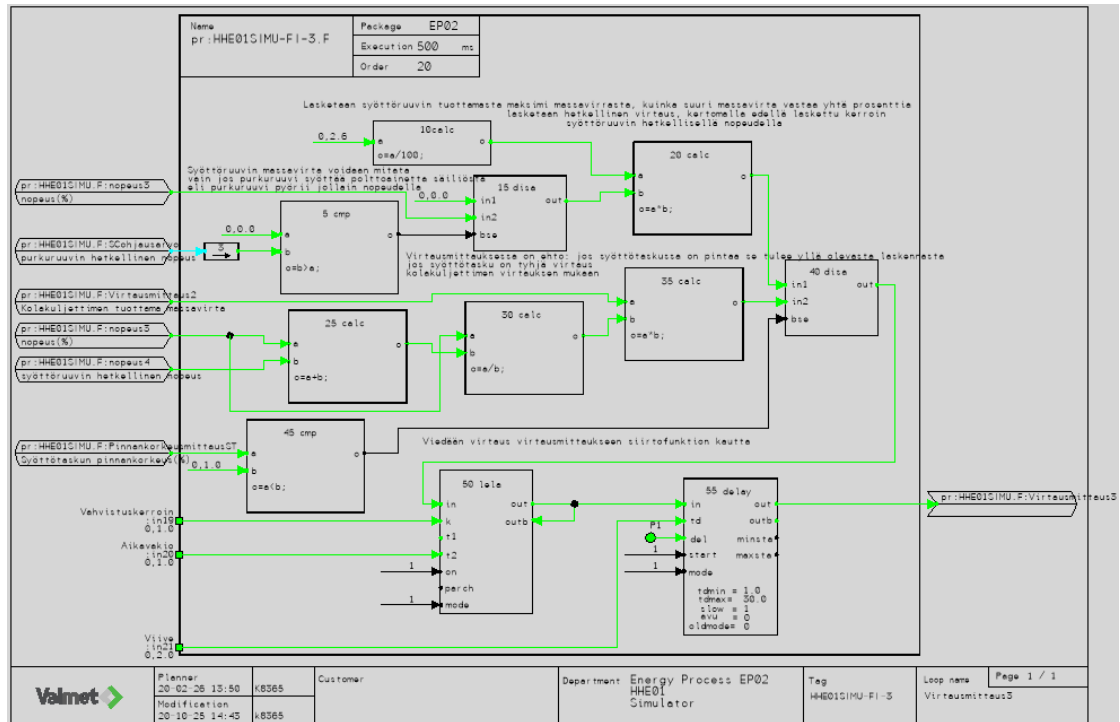
Pinnankorkeusmittaukselle piti vielä tehdä simulaattoriohjelma ja sen tein edellä mainittujen sovellusohjelmien jälkeen. Simulaattoriohjelman toteutin samalla tavalla kuin jo aiemmin tein polttoainesäiliön pinnankorkeusmittauksessa. Tulevasta massavirrasta vähennetään lähtevä massavirta. Syöttötaskun tapauksessa kolakuljettimen massavirrasta vähennetään syöttöruuvien massavirtojen summa. Jotta sain tämän laskennan toimimaan oikein, tein sille loop-rakenteen. Rakenne toimii siten, että se laskee yhteen syöttötaskun hetkellisen pinnankorkeuden ja kolakuljettimen massavirran. Tästä se vähentää lähtevän massavirran eli syöttöruuvien massavirtojen summan. Näin sain pinnankorkeusmittauksen toimintaan. Jotta pinnankorkeusmittaus ei

näytä negatiivisia arvoja, vähennyslaskun tuloksen minimiarvo rajoitetaan nollaan.
(Ks. Kuvio 18.)



Kuvio 18. Syöttötankin pinnankorkeuden mittaus (simulaattoriohjelma)

Tässä yhteydessä huomasin, että syöttöruuvien massavirrat piti rajoittaa. Rajoitus piti tehdä niin, että syöttötaskun ollessa tyhjä massavirtojen summa ei voi olla suurempi kuin kolakuljettimen massavirta. Jos syöttötaskussa on pintaa, massavirrat voivat olla syöttöruuvien pyörimisnopeuksien mukaiset. (Ks. kuvio 19.)



Kuvio 19. Syöttöruuvien massavirta (syöttöruuveja on kaksi, molempien massavirrat ovat samanlaiset)

Kun syöttötaskussa ei ole pintaa, syöttöruuvien massavirta lasketaan siten, että syöttöruuvien nopeus jaetaan sen nopeuden ja toisen syöttöruuvien nopeuden summalla. Näin saatu suhdeluku ja kolakuljettimen hetkellinen massavirta kerrotaan keskenään. Saatua tulosta toimii syöttöruuvien massavirtana. (ks. Kuvio 19)

4 Testaus ja tulokset

4.1 Testaussuunnitelma

Kun prosessisimulaattori on valmis, täytyy se toimeksiantajan kanssa tarkistaa ja tehdä tarvittavat testiajot. Näin saadaan selville, onko simulaattori vaatimusmäärittelyn ja toimintakuvausten mukainen.

Ensimmäisenä käydään läpi simulaattoriohjelma. Simulaattoriohjelmasta tarkistetaan, että siinä on mallinnettu kaikki lähtötiedoissa annetun prosessin vaatimat toiminnot ja laitteet. Seuraavaksi tarkistetaan sovellusohjelma, jolla prosessia on tarkoitettu ohjata. Sovellusohjelma käydään läpi ja testataan, että onko prosessin laitteita ja siten koko prosessia mahdollista ajaa toimintakuvausten mukaisesti. Jotta tähän saadaan selvyys, täytyy samalla tarkistaa, että sovellusohjelman ja simulaattoriohjelman välinen kommunikaatio on kunnossa. On siis tarkistettava ja käytävä läpi kommunikaatiota varten tehty I/O-rajapinta. Täytyy selvittää, meneekö sovelluksesta annetut ohjaukset rajapinnan avulla oikeisiin osoitteisiin eli oikeille laitteille. Tätä varten täytyy myös tarkistaa rajapintaa varten tehty I/O-signaaliluettelo. Eli tarkistetaan, onko I/O-signaaliluettelossa kaikki tarvittavat signaalit ja onko niitä käytetty oikein rajapinnassa. I/O-luettelo käydään läpi signaali signaalilta ja tarkistetaan, että yhteydet ovat kunnossa. Tähän käytän apuna tarkistuslistaa, jonka pohjana on aiemmin tekemäni I/O-luettelo. Luettelossa on sarake nimeltä ”yhteys kunnossa”. Tähän sarakkeeseen laitetaan rasti, jos tarkistettavana oleva signaali todetaan toimivaksi. (ks. Liite 8)

Lopuksi tarkistetaan valvomo. Katsotaan, löytyykö valvomosta kaikki PI-kaavion ja vaatimusmäärittelyn mukaiset laitteet ja toiminnot. Samalla katsotaan myös, onko valvomossa kaikki toimintakuvausten vaatimat toiminnot.

Kun kaikki edellä mainittu on tarkistettu, tehdään prosessisimulaattorille tarvittavat testiajot, jotta nähdään, että koko kokonaisuus toimii toimeksiantajan vaatimusten mukaisesti.

4.2 Tulokset

Valmis prosessisimulaattori tarkistettiin ja testattiin toimeksiantajan kanssa. Testaus suoritettiin testaussuunnitelman mukaisesti. Prosessisimulaattorin valvomo todettiin vaatimusmäärittelyn mukaiseksi ja se sisälsi kaikki tarvittavat toiminnot. Prosessin kaikki toiminnot testattiin valvomosta käsin ja samalla saatiin myös tehtyä IO-testi tarkistuslistan mukaisesti. (Ks. liite 8.) Näin tultiin siihen tulokseen, että sovellusohjelmien ja simulaattoriohjelmien välinen kommunikaatio on kunnossa. Samalla todettiin, että kommunikaatiota varten tehty rajapinta on kunnossa ja siten myös sitä varten tehty IO-lista. Kun prosessin toiminnot testattiin, pystyttiin myös näkemään, että sovellusohjelmat ja simulaattoriohjelmat toimivat toimintakuvausten mukaisesti.

Lopuksi tehtiin vielä testiajoja koko prosessikokonaisuudelle ja todettiin, että se on toimiva kokonaisuus. Polttoainesäiliöön voidaan lisätä polttoainetta ja säiliön pinnankorkeusmittaus toimii. Prosessia ajettaessa kaikki kuljettimet toimivat oikein ja niiden tuottamat massavirrat näkyvät oikein. Kaikki hälytykset ja lukitukset toimivat oikein. Säättöpiiri toimii oikein automaatti- ja manuaaliohjauksella. Myös sitä varten tehty pinnankorkeusmittaus toimii oikein. Myös toimeksiantajan haluamat laskennat toimivat oikein.

5 Pohdinta

Tämän opinnäytetyön toimeksianto oli käytännönläheinen ja hyvin selkeä. Toimeksiantoa eli prosessisimulaattorin suunnittelua varten annettiin selkeä vaatimusmäärittely ja selkeät toimintakuvaukset. Näillä lähtötiedoilla oli helppo aloittaa suoraan työn käytännön toteutuksen eli prosessin suunnittelu ilman mitään ennakkotutkimuksia. Minulle mieluista on tehdä nimenomaan käytännönläheisiä suunnittelutöitä, jotka eivät vaadi mitään ennakkotutkimuksia. Siksi tämän opinnäytetyön toimeksianto oli minulle täydellinen.

Tätä suunnittelutyötä oli pääsääntöisesti melko helppo tehdä eikä siinä ilmennyt mitään suurempia ongelmia. Työn tekoa helpotti merkittävästi se, että prosessia varten suunnitellut ohjelmat sisälsivät paljon samankaltaisuutta. Tämän vuoksi jo tehtyjä ohjelmia pystyi työn edetessä kopioimaan myöhempisiin toimintoihin. Suunnitteluympäristön eli Valmet DNA -ympäristön käyttö oli hyvässä muistissa, sillä viimeisimmät automaatio suunnittelukurssit käsittelivät Valmet DNA -ympäristöllä suunnittelua. Siksi toimilohkokaavioiden ja valvomon teko oli melko nopeaa ja helppoa eikä siinä tarvinnut oikeastaan apua. Apua tarvitsi välillä lähinnä siinä, että toimilohkokaaviot saatiin toimimaan oikein.

Lopputuloksena saatiin suunniteltua Jyväskylän ammattikorkeakoululle toimiva prosessisimulaattori, joka vastasi toimeksiantajan tarpeita ja vaatimuksia. Näin saatiin energiatekniikan opiskelijoille uusi oppimisympäristö, jossa he voivat ajaa ja tutkia kiinteää polttoainetta kuljettavaa prosessia, jota ei fyysisesti ole koululla.

Prosessisimulaattoria testattaessa siinä ilmeni myös jatkokehitysideoita. Syöttötaskun pintasäätöä tullaan myöhemmin muokkaamaan siten, että sen käytettävyys paranee. Pintasäädön oli siis tarkoitus toimia siten, että se pitää syöttötaskussa sopivaa pinnankorkeutta rajoittamalla kolakuljettimen nopeutta. Käytännössä säädin itsessään saatiin toimimaan eli se kyllä rajoittaa kolakuljettimen nopeutta, kun asetusarvo eli haluttu pinnankorkeus ylitetään. Säätimen käytettävyys ja hyödynnettävyys ei kuitenkaan ole vielä järkevä. Tämä johtuu siitä, että säädin ei pysty säätämään kolakuljettimen nopeutta siten, että syöttötaskuun saataisiin polttoainetta ja siten pinnankorkeutta, jota pitää asetusarvon mukaisena. Kolakuljettimen nopeus joudutaan säätämään käsin sellaiseksi, että syöttötaskuun alkaa tulla pintaa. Siten pintasäädin voi rajoittaa kolakuljettimen nopeutta asetusarvon mukaisesti. Jotta säädin toimisi järkevästi, syöttötaskussa pitäisi pitää jatkuvasti pintaa. Tätä varten pintasäätö tullaan myöhemmin muokkaamaan sellaiseksi, että se rajoittaa kolakuljettimen nopeuden sijasta syöttöruuvien nopeutta. Testiajojen yhteydessä havaittiin, että myös valvomoa voisi hieman kehittää. Sitä voisi muokata niin, että sieltä olisi mahdollista nähdä laitteiden toimintakuvaukset. Samalla tavoin olisi hyvä saada myös laitteita koskevien lukitusten kuvaukset näkyville.

Alkuperäisen aikataulusuunnitelman mukaan työ oli tarkoitus saada valmiiksi maaliskuussa 2020. Työn valmistuminen venyi kuitenkin syksylle 2020, sillä maailmalla ja Suomessa vallitseva covid-19 pandemia keskeytti työn käytännön toteutuksen tekemisen maaliskuun alkupuolella ja jatkamaan pystyi vasta syksyllä.

Lähteet

Alakangas, E. Hurskainen, M. Korhonen, J. Laatikainen-Luntama, J. 2016. Suomessa käytettävien polttoaineiden ominaisuuksia, 64-65. Viitattu 18.9.2020.

<https://www.vttresearch.com/sites/default/files/pdf/technology/2016/T258.pdf>

Banks, J. Carson II, J. Nelson, B. Nicol, D. 2014. Discrete-Event System Simulation. Käsikirja. Pearson Education Limited

Kippo, A. Tikka, A. 2008. Automaatiotekniikan perusteet. Oppikirja. Helsinki: Edita Prima Oy.

Kuisma, A. 2020. Automaatiotekniikan lehtori. Jyväskylän ammattikorkeakoulu. Haastattelu 20.1.2020.

Metso DNA Manuals, Collection 2011 Fi V.14.1 build 1, Viitattu 29.1.2020

Valmet DNA. N.d. Tietoteksti Valmetin WWW-sivuilla. Viitattu 19.2.2020

<https://www.valmet.com/automation/control-systems/valmet-dna/>

Valmet DNA Engineering Function Block CAD. N.d. Tietoteksti Valmetin WWW-sivuilla. Viitattu 19.2.2020

<https://www.valmet.com/automation/control-systems/valmet-dna/engineering-maintenance-tools/valmet-dna-engineering-function-block-cad/>

Valmet DNA explorer - Integrates all engineering functions. N.d. Tietoteksti Valmetin WWW-sivuilla. Viitattu 19.2.2020

<https://www.valmet.com/automation/control-systems/valmet-dna/engineering-maintenance-tools/valmet-dna-explorer/>

DNA Operate. N.d. Tietoteksti Valmetin WWW-sivuilla. Viitattu 19.2.2020

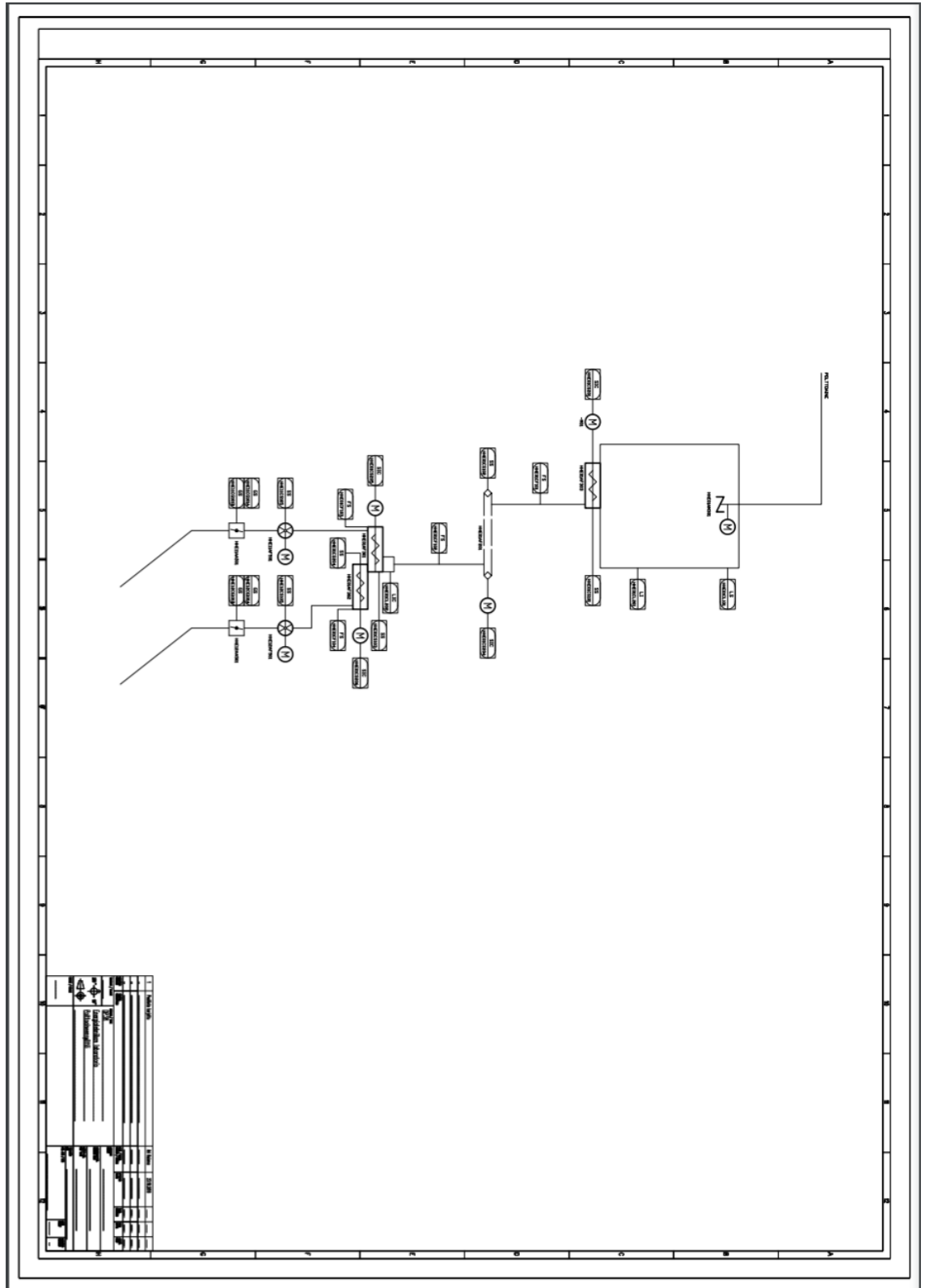
<https://www.valmet.com/automation/control-systems/valmet-dna/user-interface/operate/>

Transfer function of control system. 2019. Artikkelij Electrical 4 U:n WWW-sivulla. Viitattu 22.10.2020

<https://www.electrical4u.com/transfer-function/>

Liitteet

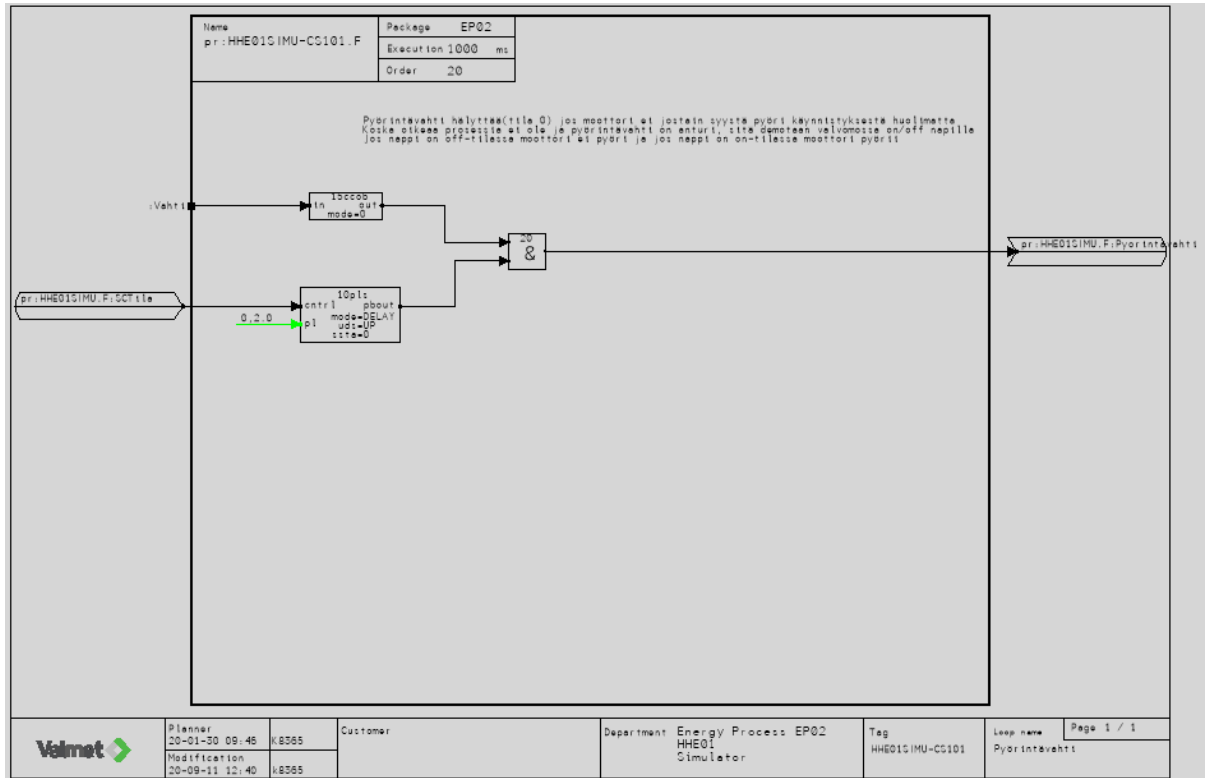
Liite 1. PI-kaavio Huom. PI-kaaviosta puuttuu lopullisessa simulaattorissa olevat virtausmittaukset FI-1, FI-2, FI-3 ja FI-4.



Liite 2. I/O-luettelo

Laitteen tyyppi	Mittausalue	Mittausyksikkö	Laitteen nimi	AI	AO	BI	BO	FBC	Slave	Offset	Korjiteho
HHE01AM301			Polttoainesäiliön sekoitin käyntilupa				1	6	10	0	
HHE01AM301			Polttoainesäiliön sekoitin käyntitieto				1	6	11	12	
LS HHE01CL101	ONDOFF		Polttoainesäiliön pinnankorkeuden rajaustykin				1	6	12	0	
LI HHE01CL201	0-100	%	Polttoainesäiliön pinnankorkeuden mittaus				1	6	13	0	
SIC HHE01AF303			Taajuusmuuttajan käyntilupa				1	6	14	0	
SIC HHE01AF303	0-100	%	Taajuusmuuttajan käyntitieto				1	6	15	12	
SIC HHE01CS201			Ruuvikulkulaitimen nopeuden asetus				1	6	16	3	
SS HHE01CS101			Puorintävähti				1	6	17	0	
FS HHE01CF101			Tukosvähti				1	6	18	0	
SIC HHE01AF201			Taajuusmuuttajan käyntilupa				1	6	19	0	
SIC HHE01AF201			Taajuusmuuttajan käyntitieto				1	6	20	0	
SIC HHE01CS204	0-100	%	Kolikkulähtimen nopeuden asetus				1	6	21	12	
FS HHE01CF102			Tukosvähti				1	6	22	3	
LI HHE01CL202	0-100	%	Syöttöastakan pinnankorkeuden mittaus				1	6	23	0	
SIC HHE01AF301			Taajuusmuuttajan käyntilupa				1	6	24	0	
SIC HHE01AF301			Taajuusmuuttajan käyntitieto				1	6	25	0	
SIC HHE01CS205	0-100	%	Ruuvikulkulaitimen nopeuden asetus				1	6	26	12	
SS HHE01CS103			Puorintävähti				1	6	27	3	
FS HHE01CF103			Tukosvähti				1	6	28	0	
SS HHE01CS104			Puorintävähti				1	6	29	0	
SIC HHE01AF302			Taajuusmuuttajan käyntilupa				1	6	30	0	
SIC HHE01AF302			Taajuusmuuttajan käyntitieto				1	6	31	0	
SIC HHE01CS206	0-100	%	Ruuvikulkulaitimen nopeuden asetus				1	6	32	12	
FS HHE01CF104			Tukosvähti				1	6	33	3	
SS HHE01CS105			Puorintävähti				1	6	34	0	
SS HHE11AF501			Sulkusyötin käyntilupa				1	6	35	0	
SS HHE11AF501			Sulkusyötin käyntitieto				1	6	36	0	
SS HHE11AF501			Puorintävähti				1	6	37	12	
SS HHE12CF5105			Sulkusyötin käyntilupa				1	6	38	0	
SS HHE12AF501			Sulkusyötin käyntitieto				1	6	39	0	
SS HHE12AF501			Sulkusyötin käyntitieto				1	6	40	12	
HHE11AA501	ONDOFF		Sulkupelti ohjaus				1	6	41	0	
SS HHE11CG101A	KIINNI		Sulkupelti rajatieto				1	6	42	0	
SS HHE11CG101B	KIINNI		Sulkupelti rajatieto				1	6	43	0	
HHE12AA501	ONDOFF		Sulkupelti ohjaus				1	6	44	0	
SS HHE12CG101A	ALUKI		Sulkupelti rajatieto				1	6	45	0	
SS HHE12CG101B	KIINNI		Sulkupelti rajatieto				1	6	46	0	
FI-1	0-10	kg/s	Virtausmittaus				1	6	47	0	
FI-2	0-10	kg/s	Virtausmittaus				1	6	48	0	
FI-3	0-10	kg/s	Virtausmittaus				1	6	49	0	
FI-4	0-10	kg/s	Virtausmittaus				1	6	50	0	
Tilveys	0-500	kgm ³	Polttoaineen tilveys				1	6	53	0	
Tilavuus	0-160	m ³	Säiliön lisäyksen polttoaineen tilavuus				1	6	54	0	

Liite 3. Pyörintävahdin simulaattoriohjelma



Liite 4. Tukosvahdin simulaattoriohjelma

Name pr:HHE01GIMU-CF101.F	Package EP02	Execution 1000 ms	Order 20
------------------------------	-----------------	----------------------	-------------

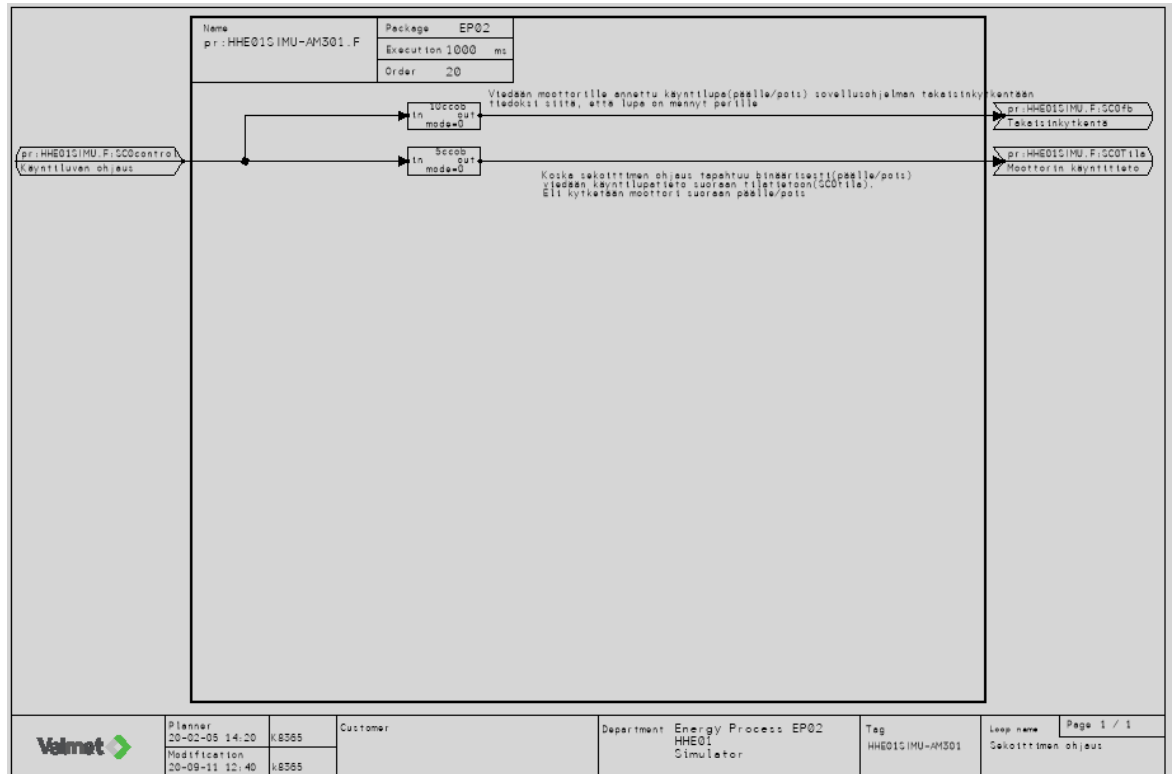
Tukosvahti on anturi joka hälyttää jos prosessin pöytätoissa on tukos.
Koska oltiin prosessissa ei ole anturia demoteen valvomissa on/off napille
kun napit on on-tilassa(tulo on 1) anturi on normaali tilassa ei ilme tukosta ei ole
kun napit on off-tilassa(tulo on 0) anturi hälyttää tukoksesta

pr:HHE01GIMU.F:Tukosvahti

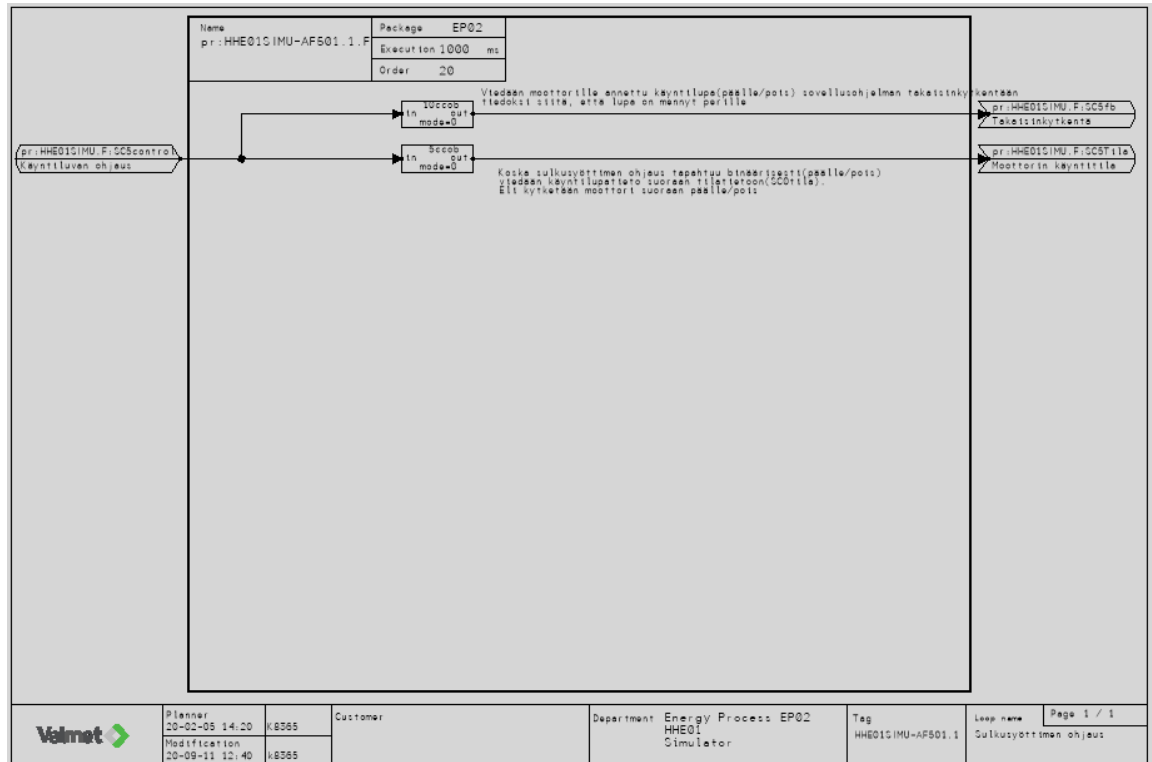
Valmet

Planner 20-01-29 14:10 K8285	Customer	Department Energy Process EP02 HHE01 Simulator	Tag HHE01GIMU-CF101	Loop name Tukosvahti	Page 1 / 1
Modification 20-09-11 12:40 K8285					

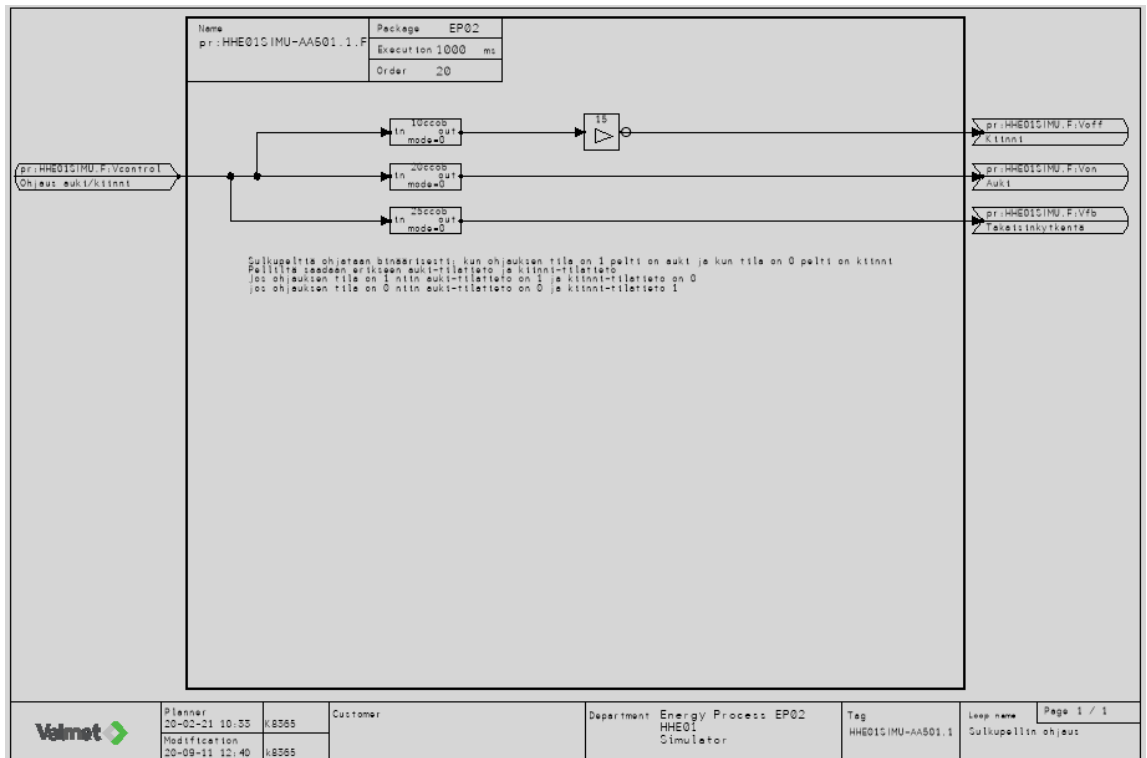
Liite 5. Polttoainesäiliön sekoittimen simulaattoriohjelma



Liite 6. Sulkusyöttimen simulaattoriohjelma



Liite 7. Magneettiventtiilin(sulkupelti) simulaattoriohjelma



Liite 8. I/O-signaalien tarkistuslista

Laitteen laaji	Mittausalue	Mittausyksikkö	Laitteen nimi	AI	AO	BI	BO	FBC	Slave	Korttiteho	Offset	Yhteys kunnossa
HHE01AM301			Polttoainessaation sekoinin käyntilupa				1	6	10	0		X
HHE01AM301			Polttoainessaation sekoinin käyntitieto			1		6	11	12		X
LS.HHE01CL101	ON/OFF		Pinnankorkeuden rajaustykin			1		6	12	0		X
LI.HHE01CL201	0-100	%	Pinnankorkeuden mittaus				1	6	13	0		X
SIC.HHE01AF303			Taajuusmuuttajan käyntilupa				1	6	14	0		X
SIC.HHE01AF303			Taajuusmuuttajan käyntitieto			1		6	15	12		X
SIC.HHE01CS201	0-100	%	Ruuvikulkuleitimen nopeus			1		6	16	3		X
SS.HHE01CS101			Pyoöntävähti			1		6	17	0		X
FS.HHE01CF101			Tukosvähti			1		6	18	0		X
SS.HHE01CS102			Pyoöntävähti			1		6	19	0		X
SIC.HHE01AF201			Taajuusmuuttajan käyntilupa				1	6	20	0		X
SIC.HHE01AF201			Taajuusmuuttajan käyntitieto			1		6	21	12		X
SIC.HHE01CS204	0-100	%	Kolkakuljettimen nopeus			1		6	22	3		X
FS.HHE01CF102			Tukosvähti			1		6	23	0		X
LI.HHE01CL202	0-100	%	Syötötaskun pinnankorkeuden mittaus			1		6	24	0		X
SIC.HHE01AF301			Taajuusmuuttajan käyntilupa				1	6	25	0		X
SIC.HHE01AF301			Taajuusmuuttajan käyntitieto			1		6	26	12		X
SIC.HHE01CS205	0-100	%	Ruuvikulkuleitimen nopeus			1		6	27	3		X
SS.HHE01CS103			Pyoöntävähti			1		6	28	0		X
FS.HHE01CF103			Tukosvähti			1		6	29	0		X
SS.HHE01CS104			Pyoöntävähti			1		6	30	0		X
SIC.HHE01AF302			Taajuusmuuttajan käyntilupa				1	6	31	0		X
SIC.HHE01AF302			Taajuusmuuttajan käyntitieto			1		6	32	12		X
SIC.HHE01CS206	0-100	%	Ruuvikulkuleitimen nopeus			1		6	33	3		X
FS.HHE01CF104			Tukosvähti			1		6	34	0		X
SS.HHE01CS105			Pyoöntävähti			1		6	35	0		X
SS.HHE01AF501			Sulkusyvöin käyntilupa				1	6	36	0		X
SS.HHE01AF501			Sulkusyvöin käyntitieto			1		6	37	12		X
SS.HHE01CS105			Pyoöntävähti			1		6	38	0		X
SS.HHE01AF501			Sulkusyvöin käyntilupa				1	6	39	0		X
SS.HHE01AF501			Sulkusyvöin käyntitieto			1		6	40	12		X
HHE01AA501	ON/OFF		Sulkupelti ohjaus				1	6	41	0		X
GS.HHE01CG101A	AUKI		Sulkupelti rajatieto			1		6	42	0		X
GS.HHE01CG101B	KIINNI		Sulkupelti rajatieto			1		6	43	0		X
HHE01AA501	ON/OFF		Sulkupelti ohjaus				1	6	44	0		X
GS.HHE01CG101A	AUKI		Sulkupelti rajatieto			1		6	45	0		X
GS.HHE01CG101B	KIINNI		Sulkupelti rajatieto			1		6	46	0		X
FI-1	0-10	kg/s	Virtausmittaus				1	6	47	0		X
FI-2	0-10	kg/s	Virtausmittaus			1		6	48	0		X
FI-3	0-10	kg/s	Virtausmittaus			1		6	49	0		X
FI-4	0-10	kg/s	Virtausmittaus			1		6	50	0		X
Tiheys	0-500	kg/m ³	Polttoaineen tiheys			1		6	53	0		X
Tilavuus	0-160	m ³	Säiliöön lisäyryn polttoaineen tilavuus			1		6	54	0		X