

Säätimen integroiminen sovelluskirjastoon

Mikko Tallgren

Opinnäytetyö

Lokakuu 2016

Tekniikan ja liikenteen ala

Insinööri (AMK), Automaatiotekniikan tutkinto-ohjelma

Tekijä(t) Tallgren Mikko	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Lokakuu 2016
	Sivumäärä 49	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Säätimen integroiminen sovelluskirjastoon		
Tutkinto-ohjelma Automaatiotekniikan tutkinto-ohjelma		
Työn ohjaaja(t) Ari Kuisma, Veli-Matti Häkkinen		
Toimeksiantaja(t) JEEC Oy		
Tiivistelmä <p>Tavoitteena oli sovittaa uusi säädin vanhan säätimen tilalle olemassa olevaan säätötoimilohkoon Siemensin TIA Portal -ympäristössä. Uudesta säätimestä haluttiin selvittää kaikki uudet ominaisuudet ja niiden mahdollinen hyödyntäminen jatkokäyttöön. Tärkeimpänä ominaisuutena säätimessä on automaattinen viritystoiminto, jonka algoritmi haluttiin selvittää, samoin se onko säätimenautomaattisesta virittämisestä hyötyä käyttöönotoissa. Opinnäytetyö tehtiin JEEC Oy:lle, joka tarjoaa korkealaatuista sähkö- ja automaatiosuunnittelua.</p> <p>Työ toteutettiin Siemens TIA Portal -ohjelmistolla. Säädin integrointiin ja testattiin ensin simuloituissa olosuhteissa ja tämän jälkeen valmis säätötoimilohko käyttöönotettiin Jyväskylän ammattikorkeakoulun laboratoriossa. Käyttöönotossa säätötoimilohkolla ohjattiin laboratorion vesiprosessin säätöventtiiliä. Toimilohkon automaattista viritystä kokeiltiin kolmella erilaisella testillä: virtaus säätö, pintasäätö sekä pintasäätö kaskadisäätimellä.</p> <p>Tuloksena saatiin kehitettyä säätötoimilohko, jossa on automaattinen viritystoiminto. Viritystoiminnosta saaduilla viritysparameetreilla pystyttiin saamaan aikaiseksi stabiileja säätöpiirejä. Toiminnon havaittiin kuitenkin antavan joissain tilanteissa huonoja parametreja, joiden johdosta ei automaattiseen virittämiseen voida täysin luottaa.</p> <p>Vaikka säätimeen automaattisen virittämisen hakemiin parametreihin ei voida täysin luottaa, voidaan sitä silti pitää suurena apuna hankalien kohteiden virittämisessä.</p>		
Avainsanat (asiasanat) PID-säädin, säätötekniikka, sovellussuunnittelu, TIA Portal -ohjelmointityökalu		
Muut tiedot		

Author(s) Tallgren Mikko	Type of publication Bachelor's thesis	Date October 2016 Language of publication: Finnish
	Number of pages 49	Permission for web publication: x
Title of publication Integration of controller to application library		
Degree programme Automation Engineering		
Supervisor(s) Ari Kuisma, Veli-Matti Häkkinen		
Assigned by JEEC Oy		
Abstract <p>The purpose of this thesis was to integrate a new controller to the place of an existing controllers in the old control function block at Siemens TIA Portal environment. Additionally, the goal of the thesis was to find out all the new features and their possible utilization for further use. The most important feature of the new controller was the auto-tuning function, the algorithm of which needed to be investigated to find out about its function as well as its benefit when using it at startup. The thesis was assigned by JEEC Oy, a company that offers high-quality electrical and automation design.</p> <p>The work was carried out using the Siemens TIA Portal software. First, the controller was integrated and tested in simulated conditions after which the ready control function block was tested in the laboratory at JAMK University of Applied Sciences. In the test, the control function block controlled water process with a control valve. The automatic tuning of the function block was tested using three different ways of testing: flow control, surface control and surface control with cascade controller.</p> <p>The thesis resulted in the control function block with the auto-tuning function. The obtained tuning parameters made stable control loops. However, auto-tuning function was found to give bad parameters in some situations; consequently, the tuning cannot be trusted completely.</p> <p>Even if the auto-tuning does not always reach the correct values can auto-tuning can still be considered a great asset for tuning on difficult objects.</p>		
Keywords/tags (subjects) PID-Controller, Control technique, Software design, TIA-Portal software		
Miscellaneous		

Sisältö

Sanasto	4
1 Johdanto	5
1.1 Toimeksiantajana JEEC Oy	5
1.2 JEEC Oy:n tavoitteet	6
1.3 Henkilökohtaiset tavoitteet.....	7
2 Sovellussuunnittelun standardisointi	7
2.1 Standardit	7
2.2 Standardit ohjelmointikielet	8
2.3 Toimilohkot.....	11
3 Siemens logiikat	14
3.1 S7-1500-Sarja.....	14
3.2 S7-1200-Sarja.....	15
3.3 Siemens TIA Portal -ohjelmointityökalu.....	15
4 Sääteotekniikka	16
4.1 Säättöpiiri.....	16
4.2 Kaskadisäätö.....	17
4.3 Säättömenetelmiä	18
4.3.1 P-säätö	18
4.3.2 Integroiva säätö	19
4.3.3 Derivoiva säätö	20
4.3.4 Ideaalisäädin	20
4.3.5 Sarjasäädin.....	21
4.3.6 Rinnakkaissäädin	21
4.4 PID_Compact-säädin	22
4.5 Viritysmenetelmät.....	24
4.5.1 Virityksen tavoite.....	24

	2
4.5.2 Ziegler-Nichols	25
4.5.3 Chien-Hrones-Reswick	29
5 Opinnäytetyön toteutus	29
5.1 Säätimen integroiminen	29
5.2 Toimilohkon toteutus	32
5.3 Piirinäytön toteutus.....	33
6 Säättötoimilohkon testaus	35
6.1 Säätimen analysointi	36
6.2 Virtaussäätö.....	38
6.3 Pinnankorkeudensäätö.....	39
6.3.1 Suorasäätö	40
6.3.2 Kaskadisäätö	43
7 Tulosten arviointi	44
7.1 Tulokset	44
7.2 Kehityskohteet.....	46
8 Pohdinta.....	47
Lähteet	49

Kuviot

Kuvio 1. Ladder Diagram -ohjelmointia.....	9
Kuvio 2. Function Block Diagram -ohjelmointia	10
Kuvio 3. Sequential Function Chart -ohjelmointia	11
Kuvio 4. Säätoimilohko	12
Kuvio 5. Toimilohko (FB) ja sen luomat Data blockit (DB)	13
Kuvio 6. Säätoipiirin komponentit (Harju & Marttinen 2000, 13.)	17
Kuvio 7. Arvojen laskenta askelvastekokeesta.....	26
Kuvio 8. Tasainen värähtely	28
Kuvio 9. Piirinäyttö	34
Kuvio 10. Vesiprosessi	36
Kuvio 11. Hienosäädön värähtelytesti	37
Kuvio 12. Virtaussäädön kuvaaja	39
Kuvio 13. Säätoikäyrät PID-slow -menetelmällä	41
Kuvio 14. Säätoikäyrät PID-rapid -menetelmällä	42
Kuvio 15. Säätoikäyrät Ziegler-Nicholsin PID-viritysmenetelmällä pintasäädössä	43
Kuvio 16. Säätoikäyrät Ziegler-Nicholsin PID-viritysmenetelmällä kaskadisäädössä ...	44

Taulukot

Taulukko 1. P, I, D -termien vaikutus	25
Taulukko 2. Ziegler-Nicholsin askelvastemenetelmästä laskettavat parametrit.....	27
Taulukko 3. Ziegler-Nicholsin kriittisenvärähtelyn menetelmästä laskettavat parametrit	28
Taulukko 4. Chien-Hrones-Reswick menetelmästä laskettavat parametrit	29
Taulukko 5. Virtaussäätöön saadut parametrit, esisäätö menetelmällä	38
Taulukko 6. Virtaussäätöön saadut parametrit hienosäätö menetelmällä	38
Taulukko 7. Pintasäätöön saadut parametrit hienosäätö menetelmällä	40

Sanasto

ATEX	Räjähdyksvaarallisia tiloja sekä niissä käytettäviä laitteita koskeva lainsäädäntö
Data Block	Tiedon tallennuslohko Siemens-ohjelmistossa
Hardware-konfiguraatio	Fyysisten laitteiden määrittely sovellukseen
HMI-paneeli	Laitteiston operointiin tarkoitettu paneeli
I/O	Input/Output, sisääntulo ja ulostulo
Move-lohko	Analogisten viestien siirtoon tarkoitettu toiminto Siemens-ohjelmistossa.
Network	Ohjelman kirjoittamiseen varattu tila Siemens-ohjelmistossa
Offset	Muistialueen aloitus osoite
Ohjelmoitava-logiikka	Pieni tietokone, jota käytetään automaation ohjaukseen
Profinet	Process Field Network, Teollisuudessa käytetty ethernet-standardi
Rampitus	Lähdön ohjaaminen portaittain
SCADA	Supervisory control and data acquisition, tietokone pohjainen valvomo
Sekvenssi	Ohjelma, joka suoritetaan tietyssä järjestyksessä, tietyin ehdoin
SIL 3	Turvallisuusluokitus
Sisääntulo	Input, logiikalle tuleva signaali
Tag	Muuttuja Siemensin ohjelmistossa
Ulostulo	Output, logiikalta lähtevä signaali

1 Johdanto

Sovellussuunnittelun osuus kasvaa koko ajan laitteiden ja ohjelmistojen monipuolistuttua. Siihen kulutettu aika kuitenkin maksaa, joten on tärkeää tehdä tehokasta sovellussuunnittelua ja käyttää tehokasta sekä monipuolista ohjausjärjestelmää. Sovellussuunnittelun lisäksi automaatioinsinöörin toimenkuvaan kuuluu ohjelmiston käyttöönotto asiakkaan kohteessa. Käyttöönotossa suunniteltu automaatio sekä siihen liittyvä sovellus testataan ja luovutetaan asiakkaalle. Tästä syystä täytyy ohjelman olla toimiva ja sovelluksen käyttöönoton helppoa. Ohjelmistot ovat monimutkaisia ja sisältävät erilaisia ohjauksia ja säätöjä. Ohjauksien ja säätöjen toteutukseen on kehitetty valmiita lohkoja, joita käyttämällä tehostetaan suunnittelua entisestään.

Tässä opinnäytetyössä paneudutaan sovellussuunnittelun standardimaiseen toteutukseen toimilohkoilla sekä säätötekniikkaan. Toimilohkoon integroidaan käyttöönottoa helpottava automaattisesti virittyvä PID-säädin. Säätimien virittäminen on hankalaa ja syvällisempi säätimien ymmärtäminen haastavaa, joten automaattinen virittäminen tulee olemaan suuri apu automaatiosovelluksen käyttöönotossa.

1.1 Toimeksiantajana JEEC Oy

Opinnäytetyön toimeksiantajana oli jyväskyläläinen insinööritoimisto JEEC Oy, jonka toimipiste sijaitsee Kuormaajantiellä. Aikaisemmin JEEC sijaitsi Kirrissä, mutta muutti uusiin ja suurempiin tiloihin vuoden 2016 keväällä. Yritys tarjoaa sekä sähkö- että automaatiotekniikan toimialoilla korkealaatuista suunnittelu- ja konsultointityötä. Automaatio suunnittelussa toiminta painottuu prosessiautomaatiojärjestelmiin ja niiden käyttöönottoihin, kun taas sähkösuunnittelussa pääpaino on prosessi- ja rakennus sähkösuunnittelussa sekä tele- ja turvajärjestelmissä. (JEEC etusivu n.d.)

JEEC Oy:n menestys tulee osaavista suunnittelijoista, joilla on pitkäaikainen ja monipuolinen kokemus suunnittelutyöstä. Laatu, osaaminen, asiakaslähtöisyys ja luottamus ovat yrityksen arvot joita painotetaan. JEEC täyttää myös ISO 9001 laadunhallintastandardin. (JEEC yritys n.d.)

1.2 JEEC Oy:n tavoitteet

JEEC Oy on kehittänyt sovellussuunnitteluaan koko ajan enemmän standardoiduksi. Standardisoinnilla on saatu lisää tehokkuutta ja minimoitu mahdollisiin muutoksiin kuluvaa aikaa. Jotta suunnittelu pysyisi tehokkaana, täytyy myös sovelluskirjaston olla ajan tasalla. Opinnäytetyössä keskityttiinkin päivittämään sovelluskirjastoon uusimmat ominaisuudet PID-säätimestä ja samalla ylläpitämään standardisoinnin tehokkuus.

Tavoitteena oli saada sovitettua uuden mallinen säädin (PID_Compact) vanhan säätimen (Cont_C) tilalle olemassa olevaan toimilohkoon TIA Portal -ympäristössä. Samassa yhteydessä tuli päivittää myös valvomo-objekti vastaamaan säätimen uusia ominaisuuksia. Varsinaista pakottavaa tarvetta ei säätimen vaihdolle ollut, mutta automaation kehittyessä ja kilpailun kovetessa on aina mentävä kehityksen mukana. Opinnäytetyössä oli tarkoituksena myös tutustua syvällisemmin tähän uuteen säätimeen ja selvittää, kuinka sen ominaisuuksia pystyttäisiin hyödyntämään uusissa projekteissa.

Uudessa säätimessä mielenkiintoisena ominaisuutena oli mm. automaattinen viritys. Mikäli automaattinen viritys saataisiin toimimaan, voitaisiin sillä helpottaa käyttöönotossa tapahtuvaa viritystä huomattavasti. Myös viritysparametrien hakeminen automaattisella viritystoiminnalla mahdollistaisi virityksen kokemattomammallekin henkilölle.

Tarkoituksena oli ensisijaisesti saada säädin toimimaan Siemens S7-1500-logiikalla ja sen jälkeen myös S7-1200-sarjassa. Olisi myös tärkeää tietää ja ymmärtää mihin Siemensin säädin perustaa säätämisensä eli millä tavalla se laskee omat parametrinsa.

Työssä tehtävää säädintoimilohkoa oli tarkoitus myös testata oikeassa ympäristössä simuloinnin lisäksi. Jotta toimilohko uskalletaan ottaa osaksi projektia, täytyy se testata laboratorioympäristössä. Testausympäristönä toimi Jyväskylän ammattikorkeakoulun automaatiotekniikan laboratoriot ja niiden laitteistot.

1.3 Henkilökohtaiset tavoitteet

Henkilökohtaisina tavoitteina oli perehtyä syvällisemmin sovellussuunnittelun standardisointiin ja säätötekniikkaan. Aikaisempaa osaamista minulla oli hieman molemmista, ja niiden taitojen pohjalta oli hyvä syventää osaamista näillä osa-alueilla.

Toimeksiantajan tavoite oli saada minulle tarvittavat taidot TIA Portal -ympäristön käyttöön, jotta minulla on valmiuksia tehdä sovellussuunnittelua itsenäisesti. Työssä pääsi myös syventymään sovellussuunnitteluun, mistä on hyötyä myös muiden järjestelmien käyttöä ajatellen.

Työssä keskeisessä osassa oli myös säätötekniikka, jonka syvempi hallitseminen oli yksi tärkeimmistä tavoitteista. Säätötekniikan syvällisemmällä ymmärtämisellä pystyy virittämään säätöpiirejä hankalissakin olosuhteissa. Säätötekniikka tuo myös opinnäytetyöhön monipuolisuutta, eikä työ keskity pelkästään ohjelmointiin

2 Sovellussuunnittelun standardisointi

Sovellussuunnittelun standardisoinnilla pyritään tekemään sovellus järjestelmällisemmäksi eli pyritään yhdenmukaistamaan eri sovelluksia valmiilla toimilohkoilla. Sovellussuunnittelun standardisointi vähentää virheiden määrää ja vähennetään muutoksiin kuluva aikaa. Jotta suunnittelua pystyttäisiin tekemään mahdollisimman tehokkaasti, täytyy käytettävän sovelluskirjaston ja toimilohkojen olla ajan tasalla.

2.1 Standardit

Ohjelmoitaville logiikoille on olemassa oma standardi IEC 61131. Standardi on kahdeksanosainen ja käsittelee laajasti vaatimuksia ohjelmistokielistä ohjelmistoympäristöön. Standardi on laitevalmistajien hyväksymä, ja ohjelmistoja on toteutettu standardin puitteissa. Standardin käyttöä valvoo PLCOpen-yhteisö, joka toimii maailmanlaajuisesti. Standardeista tärkein ohjelmoinnin kannalta on kolmas osa, jossa käsitel-

lään ohjelmointikieliä, arkkitehtuuria, tietotyyppejä sekä valmiita toimilohkoja. Standardi kuitenkin määrittelee vain ohjausjärjestelmän ja ohjelmoijan rajapinnan, eikä ota kantaa järjestelmien sisällä tehtyihin ratkaisuihin. Tästä syystä ohjelmointitavat voivatkin poiketa laitevalmistajien välillä. Tämän lisäksi standardissa ei määritellä tiedostoformaattia eikä I/O-osoitteistusta, joten avoimuus ja ohjelmien siirto eri valmistajien välillä ei onnistu. (Automaatiosovellusten ohjelmistokehitys 2005, 14.)

Standardin tarkoituksena on yhdenmukaistaa logiikkaohjelmointia. Ohjelmistojen laajetessa on hyvä olla suuntaa antava ohje, jota noudattamalla saadaan ohjelmistoista samankaltaisia. Tällöin laitteistoa voivat ohjelmoida muutkin henkilöt kuin ohjelman tekijä. Esimerkiksi samoilla ohjelmointikielillä ohjelmointi helpottaa eri laitevalmistajien ohjelmistojen käyttämistä. (John & Tiegelkamp 2010, 12.)

2.2 Standardit ohjelmointikielet

IEC 61131-3 määrittelee viisi erilaista ohjelmointikieltä, näistä kaksi on tekstipohjaisia ja kolme graafisia.

Tekstipohjaisia ovat

- Instruction list, IL
- Structured Text, ST.

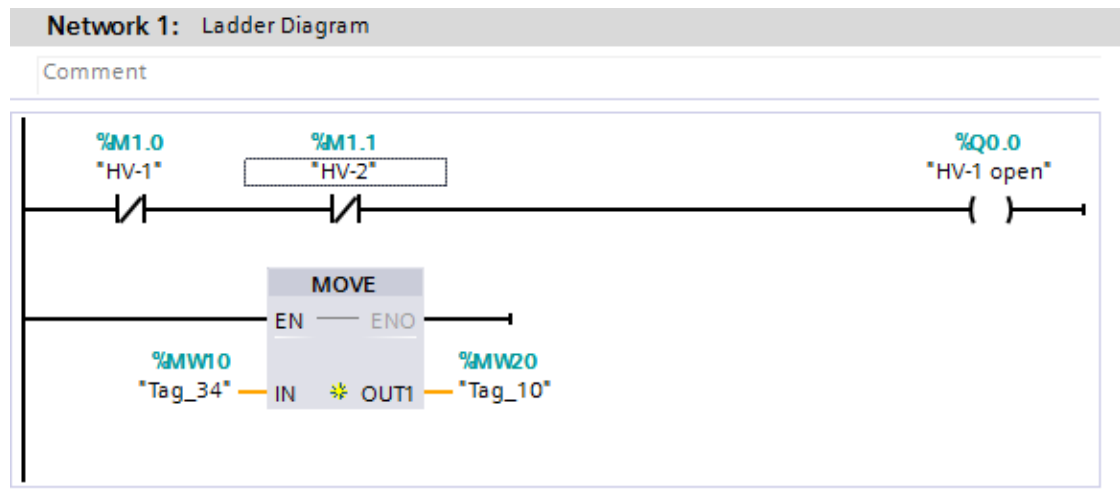
Graafisia ovat

- Ladder Diagram, LAD
- Function Block Diagram, FBD
- Sequential Function Chart, SFC.

(IEC 61131-3 2013, 195)

Instruction list -ohjelmointikieltä voidaan verrata Assembly-ohjelmointikieleen. Se on monipuolinen kieli, johon monet graafiset kielet voidaan kääntää. Structured Text-ohjelmointikieli muistuttaa eniten ulkoasultaan tietokonepohjaisia ohjelmointikieliä kuten C :tä ja PASCAL :a. Structured Text on Instruction listiin verrattuna hieman vapaamuotoisempi ja sopii paremmin matemaattisiin ja tiedonsiirtoon liittyviin toimintoihin. (John & Tiegelkamp 2010, 100-116.)

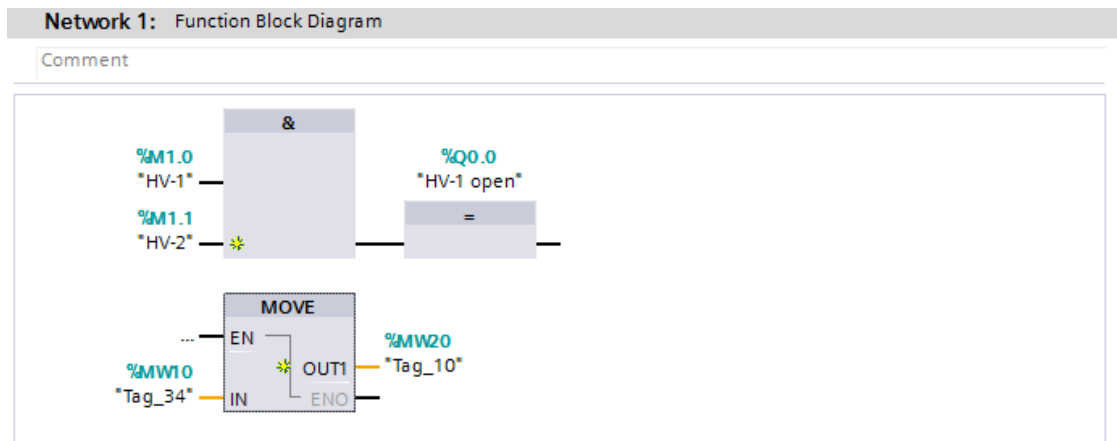
Graafiset kielet ovat nimensä mukaisesti helpommin havainnollistettavia ja ohjelman tulkitseminen onnistuu kokemattomaltakin ohjelmoitsijalta. Ladder Diagramissa on tikapuumainen esitystapa, jossa ohjelmaa luetaan vasemmasta laidasta oikealle. Tätä tapaa voidaan kutsua virtapiiritoteutukseksi, sillä vasemmassa laidassa voidaan kuvitella olevan jännite, jota ohjataan oikealle maata kohti. Kuviossa 1 on esitetty Ladder Diagram -ohjelmaa.



Kuvio 1. Ladder Diagram -ohjelmointia

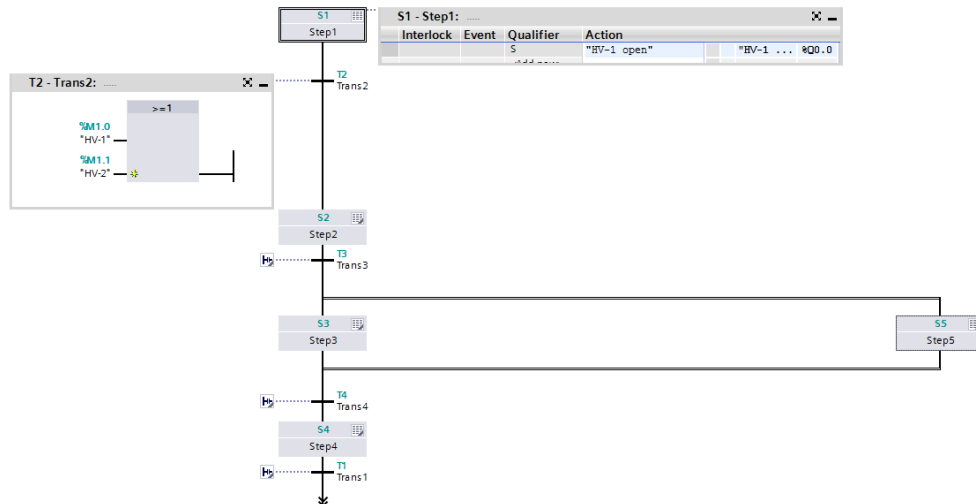
Function Block Diagram -ohjelmointikieli rakentuu laatikoista, lohkoista, joita yhdistelemällä saadaan aikaiseksi ohjaus. Kielen luettavuus on helppoa, ja se muistuttaa

Ladder Diagram -kieltä, koska siinäkin käytetään lohkoja jonkin verran. Kuten kuviossa 1 nähdään, on MOVE –lohkon toteutus hyvin saman näköinen kuin kuviossa 2. Funktion Block Diagram -ohjelman suoritusjärjestys on myös samanlainen kuin Ladder Diagrammissa, vasemmalta oikealle.



Kuvio 2. Function Block Diagram -ohjelmointia

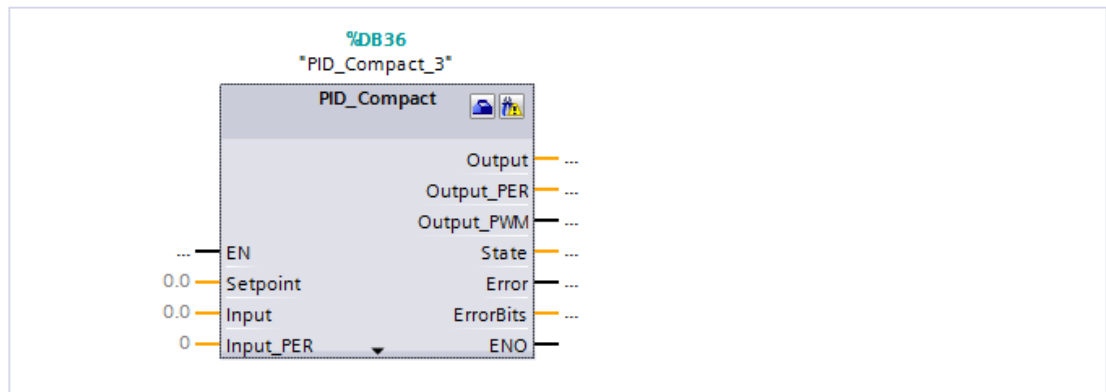
Viimeistä graafista ohjelmointikieltä, Sequential Function Chartia, käytetään pääasiassa sekvenssien ohjelmointiin. Ohjelma etenee ylhäältä alas ja suorittaa ohjelman askel kerrallaan (Step). Jokaisella askeleella on suoritusehtoja (Transition), joiden täytyessä voidaan ohjata laitteita (Action) ja siirtyä seuraavaan askeleeseen. Kuviossa 3 on Sequential Function Chart -ohjelmointikielellä toteutettu havainnollistava esimerkki.



Kuvio 3. Sequential Function Chart -ohjelmointia

2.3 Toimilohkot

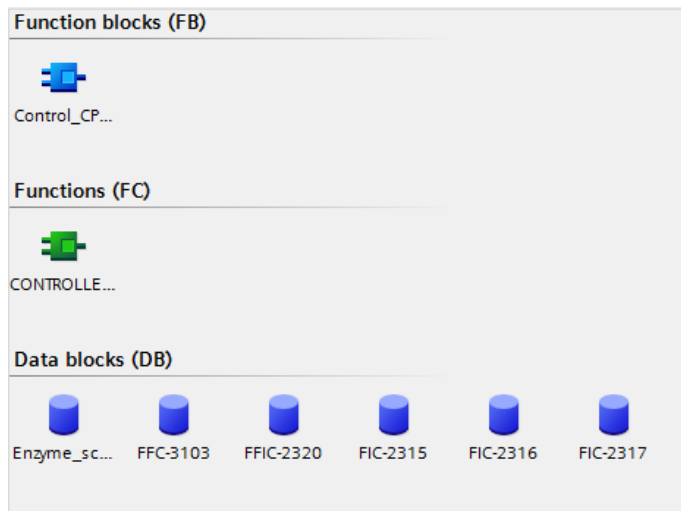
Toimilohko voidaan kuvitella laatikkona, jossa on sisään- ja ulostuloja. Lohkon sisällä käsitellään sisääntulosignaaleita, joista muodostetaan ulostulevia signaaleja. Lohkon sisällä tapahtuvat toiminnot sisääntuloille riippuvat toimilohkon ohjelmoinnista, lohko voi käytännössä sisältää minkä tyypisiä toimintoja tahansa, ks. kuvio 4. IEC-standardi määrittelee myös yleisen toimilohkomallin. Lohkomalli koostuu myös sisään- ja ulostulevista tiedoista sekä sisällä tapahtuvista algoritmeista. (Automaatiosovellusten ohjelmistokehitys 2005, 17.)



Kuvio 4. Säättötoimilohko

Toimilohkon edut tulevat esiin tilanteissa, joissa halutaan ohjata useampaa laitetta. Yhdellä toimilohkolla pystytään ohjaamaan montaa laitetta eikä jokaiselle laitteelle tarvitse suunnitella omaa ohjelmaa. Samalla toimilohkolla toteutettuja ohjauksia on myös nopea muuttaa, koska muutettaessa toimilohkon toimintaa, muuttuvat myös kaikkien ohjattavien laitteiden toiminnot. Suurempia ohjelmakokonaisuuksia tehdessä virheiden määrä kasvaa, mutta toimilohkototeutuksella sitäkin pystytään vähentämään.

Käytettäessä samaa toimilohkoa useammalle laitteelle tulee laitteen tiedot tallentaa johonkin. Tietojen tallennukseen käytetään Data block -lohkoja (DB-lohkoja). Tällöin yhtä toimilohkoa kutsumalla pystytään luomaan jokaiselle laitteelle oma DB, johon tallennetaan laitteen tietoja. Kuviossa 5 on toimilohko ja sen DB-lohkot TIA Portal -ohjelmistoympäristöstä



Kuvio 5. Toimilohko (FB) ja sen luomat Data blockit (DB)

Toimilohkon on tarkoitus olla mahdollisimman monipuolinen ja parametroitavissa. Yhdellä loholla tulisi pystyä toteuttamaan erityyppisiä ohjauksia parametreja vaihtamalla, esimerkiksi ulostulon tietotyyppiä voidaan muuttaa parametrilla. Hyvän toimilohkon tulisi muuntautua eri tilanteisiin eikä erilaisille variaatioille tarvitsisi tehdä uutta sovellusta ja samalla mahdollisia suunnitteluvirheitä.

Uuden toimilohkon toteutuksessa on tärkeää lohkon testaaminen. Toimilohkon suunnitteluvaiheessa on tehty uutta ohjelmaa ja uudessa ohjelmassa on aina vaarana mahdolliset virheet. Näiltä vältytään vain toimilohkon järjestelmällisellä testaamisella. Testauksessa toimivaksi todettua lohkoa voidaan tämän jälkeen käyttää huolletta osana ohjelmaa, eikä tarvitse pelätä toimilohkon virheellistä käyttäytymistä.

3 Siemens logiikat

Opinnäytetyössäkin käytetyn Siemensin logiikkojen tuoteperheeseen kuuluu suuri määrä erilaisia logiikoita. Logiikoita on suunniteltu kaikkiin tarpeisiin, ja laajasta valikoimasta löytyy niin turva- kuin myös ATEX-standardit täyttäviä logiikoita. Pienimmät logiikkaohjaukset voidaan toteuttaa kaikista kevyimmällä logiikalla LOGO:lla. Vaativampiin kohteisiin löytyy S7-sarjalaisia, kuten S7-300, -400, -1200 ja -1500. Vanhemmat S7-300- ja S7-400-logiikat ovat korvautumassa S7-1500-logiikalla, mutta vanhempia malleja on saatavissa vielä pitkään. (Ohjelmoitavat logiikat n.d.)

3.1 S7-1500-Sarja

Siemensin S7-1500 on vuonna 2013 julkaistu sarja, joka on tarkoitettu käytettäväksi samantyyppisissä kohteissa, kuten aikaisemmin on käytetty S7-300- ja S7-400-logiikoita. S7-1500-logiikkamallisto on kasvanut tasaisesti julkaisu vuodesta lähtien, malliston laajuus mahdollistaa logiikan käytön kaikissa kohteissa eikä enää välttämättä tarvita vanhempia malleja. (Tehokasta automaatio-ohjelmointia S7-1500-logiikalla n.d.)

Uusimpana julkaisuna S7-1500 tuoteperheeseen on tullut turvalogiikat. Kaikista S7-1500 standardimalleista löytyy vastaava turvalogiikka, mallit on suunniteltu täyttämään SIL 3 turvallisuusvaatimukset. Tämä mahdollistaa standardi- ja turvasovelluksen tekemisen samalla logiikalla. (Tehokasta automaatio-ohjelmointia S7-1500-logiikalla n.d.)

S7-1500-logiikkojen paremman suorituskyvyn lisäksi niissä on myös pieninäyttö, josta nähdään suoraan esimerkiksi vikadiagnostiikka. Diagnostiikkaa pystyy lukemaan ilman ohjelmointia ja se kattaa koko järjestelmän hajautetusta I/O:sta käyttöihin ja kytkimiin. Logiikkaan lisätty pieni näyttö helpottaa diagnostiikan lukemista ja nopeuttaa vian paikantamista. Kaikki sarjan logiikat sisältävät myös vakiona profinet liitännän. (Tehokasta automaatio-ohjelmointia S7-1500-logiikalla n.d.)

3.2 S7-1200-Sarja

S7-1200-logiikat on tarkoitettu projekteihin, joissa LOGO käy pieneksi ja S7-1500 liian järeäksi. 1200-mallisto on suorituskykyinen, mutta se ei kuitenkaan pysty käsittelemään isompia kokonaisuuksia tehokkaasti ja soveltuukin siksi keskisuuriin projekteihin. Logiikalla voi esimerkiksi maksimissaan ohjata 16:ta PID-säädintä. Samoin kuin 1500-sarjassa on myös 1200-sarjassa turvalogiikka mahdollisuus, joka mahdollistaa standardi- ja turvasovelluksen tekemisen samalla logiikalla. (Helppoa automaatio-ohjelmointia S7-1200-logiikalla n.d.)

3.3 Siemens TIA Portal -ohjelmointityökalu

Siemens TIA Portal eli Totally Integrated Automation Portal on Siemensin kehittämä ohjelmointityökalu. TIA Portal on itsessään iso ohjelmistokehys, joka pitää sisällään logiikkaohjelmoinnin (SIMATIC STEP 7), käyttöliittymäsuunnittelun (SIMATIC WinCC) sekä taajuusmuuttajasuunnittelun (SINAMICS StartDrive). Parannusta aikaisempaan on huomattavasti, sillä ennen kaikki työkalut olivat omina sovelluksinaan. Niiden yhdistämisen myötä ohjelmien käytettävyys on parantunut huomattavasti. Ohjelman visuaalisen puoleen on myös kiinnitetty huomiota. (Totally Integrated Automation Portal n.d.)

Vanhojen Step 7 -projektien kääntäminen onnistuu TIA Portaaliin, joten ohjelman vaihtaminen uudempaan on helppoa. Ohjelman käyttäminen on myös tehty helpoksi yksinkertaisilla ja avustetuilla toiminnoilla. Esimerkiksi Hardware-konfiguraatiossa laitteet on tehty fyysisten ominaisuuksien näköisiksi sekä niiden johdotus onnistuu hiirellä piirtämällä. Logiikan ohjelmointiin on mahdollista käyttää aikaisemmissakin versioissa käytettyjä LAD-, FBD-, STL-, SCL- ja Graph-ohjelmointikieliä. (Totally Integrated Automation Portal n.d.)

Kommunikaatio TIA Portaalisissa toimii ohjelmien välillä ilma käyttäjän määrittelyä, sillä ohjelmilla on yhtenäinen tiedonhallinta. Samalla työkalulla pystytään myös tekemään visualisoinnin kaikille Siemensin tuotteille isosta SCADA-järjestelmästä pieneen HMI-paneeliin. (Totally Integrated Automation Portal n.d.)

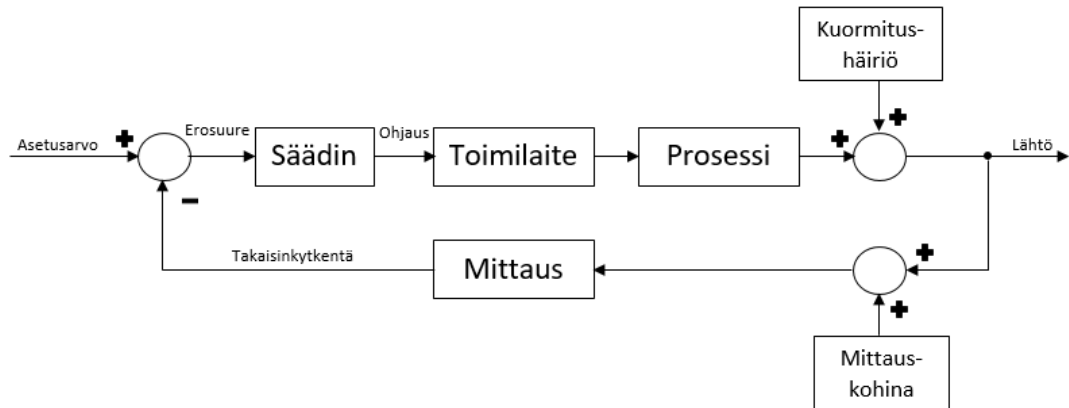
4 Sääntöteknikka

Sääntöteknikkaa käsitellään monesti matemaattisissa lausekkeissa ja ymmärtääkseen sääntöteknikkaa tulisi matematiikasta osata ainakin integroinnin ja derivoinnin teoria. Yksinkertaisimmillaan sääntöteknikkaa voidaan ajatella huoneen lämpötilan säätämällä patterin avulla. Tavoitteena olisi saada huoneen lämpötila pysymään sopivana huoneessa asuville ihmisille. Mikäli lämmitysteho olisi vakio läpi vuoden, olisi talvella mahdollisesti liian kylmä tai kesällä liian lämmin. Paras tulos saavutetaan mittaamalla sisälämpötilaa ja säätämällä lämmitystehoa. Tuloksen aikaan saamiseksi tarvitaan kuitenkin säätöpiiri. (Harju & Marttinen 2000, 9.)

4.1 Sääntöpiiri

Sääntöpiiriin kuuluu säädettävä prosessi, toimilaitte, mittaus ja säädin. Säädettävä prosessi voi olla käytännössä mitä vain, ja mitattava suure voi olla esimerkiksi painetta, virtausta tai lämpötilaa. Toimilaitte, jota säädin ohjaa, voi olla esim. moottori, venttiili tai lämpövastus. (Harju & Marttinen 2000, 9.) Aikaisemmassa esimerkistä patteri olisi toimilaitte ja huoneisto prosessi, josta mitattaisiin lämpötilaa. Huoneen lämpötilaa voitaisiin säätää ohjaamalla patteria päälle ja pois. Mikäli patterin lämpötehoa pysyttäisiin säätämään, voisi patterin tehoa säätää 0-100 %.

Säädettävästä prosessista mitattu suure kytetään takaisinkytkennällä takaisin säätimelle, jossa sitä verrataan asetusarvoon (setpoint). Näiden erotuksesta saadaan eroarvo, jonka perusteella tehdään toimilaitteelle ohjaus. Teoriassa asia menisi näin yksinkertaisesti, mutta todellisuudessa oikeissa ympäristöissä täytyy huomioida vielä mittauksessa syntyvää kohinaa ja kuormitushäiriöitä. (Harju & Marttinen 2000, 13.) Sääntöpiiriä voidaan kuvata lohkokaaviolla, kuten kuviossa 6.



Kuvio 6. Säätöpiirin komponentit (Harju & Marttinen 2000, 13.)

Takaisinkytkennän ollessa käytössä puhutaan säätöpiirin olevan automaattilla, kun taas sen ollessa pois on piiri manuaalilla. Takaisinkytkennän avulla säädin tietää, miten prosessi käyttäytyy, ja sillä pyritään eliminoimaan häiriöiden vaikutusta. Kytkennällä on kuitenkin huonojakin puolia, kuten epästabiliin tilanteeseen ohjautuminen. Epästabiililla käyttäytymisellä tarkoitetaan tilannetta, jossa ohjaus alkaa vahvistamaan säätöpiirissä tapahtuvaa värähtelyä. Värähtelyn kasvaessa koko ajan alkaa jossain vaiheessa laitteisto tai prosessi pettää. (Harju & Marttinen 2000, 13.)

4.2 Kaskadisäätö

Kaskadisäätö on kuvion 6 tilanne, mutta siihen täytyy lisätä uusi säädin vanhan säätimen rinnalle. Voidaan puhua sisemmästä ja ulommasta säätöpiiristä. Sisemmällä piirillä säädetään toimilaitetta, kun taas ulommalla piirillä säädetään itse prosessia. Kaskadisäädön hyöty tulee esiin järjestelmissä, joissa järjestelmän ohjaukseen summautuu virheitä. Sisemmällä säädöllä pyritäänkin poistamaan häiriöt, ennen kuin ne vaikuttavat säädettävään järjestelmään. (Savolainen & Vaittinen 2003, 48.)

4.3 Säätoimenetelmiä

Yleisin säädin teollisuudessa on PID-säädin. Säätimen kirjainyhdistelmä PID tulee sanoista proportional, integral ja derivative (vahvistus, integrointi ja derivointi). Säätimen toimintaan käytetään yleisenä pohjana ideaalisäädintä, mutta valmistajasta riippuen eroavaisuuksia säätimien välillä on. Tärkein on kuitenkin säätimen perusrakenne, jossa vahvistuksesta, integroinnista, derivoinnista sekä eroarvosta muodostetaan säätöarvo. Säätimellä on siis eroarvo prosessista ja kolme parametria, joiden suhteita muuttamalla pystytään vaikuttamaan säätämiseen. Se, kuinka säädin parametrit käsittelee, riippuu säätimen rakenteesta. Yleisimmät säädinrakenteet ovat ideaalinen, sarjamuotoinen ja rinnakkaismuotoinen. (Harju & Marttinen 2000, 44.)

Tärkeää PID-säätimessä on myös ymmärtää, mihin ja miten P, I ja D -termit vaikuttavat. Riippuen säädettävästä prosessista tulee valita tilanteeseen sopiva säädinyhdistelmä (P, PI, PID, PD). Yleisimmin käytetty yhdistelmä on PI-säädin, sillä pystytään hallitsemaan yleisimmät prosessit teollisuudessa. (Harju & Marttinen 2000, 15.)

4.3.1 P-säätö

P-säätimessä ei ole ollenkaan integrointia eikä derivointia mukana. Säädin muuttaa verrannollisesti toimilaitteelle menevää ohjaussignaalia erosuureeseen nähden. P-säädin kuitenkin jättää pysyvän säätöpoikkeaman, joten säädintä valittaessa tulee miettiä, haittaako pieni säätöpoikkeama prosessia. Pienestä säätöpoikkeamasta huolimatta P-säädin on hyvin toimiva ratkaisu monessa paikassa ja sen virittäminen yhdellä parametrilla on hyvin yksinkertaista. (Savolainen & Vaittinen 2003, 35.) P-säädin algoritmi on esitetty yhtälössä 1.

$$U(t) = K_p e(t) \quad (1)$$

missä K_p = vahvistus

e = eroarvo

U = säätimen lähtö

(Savolainen & Vaittinen 2003, 36.)

4.3.2 Integroiva säätö

Integroivassa säädössä P-termin lisäksi mukana on myös I-termi eli integrointi osa. Integroiva osuus laskee eroarvon pinta-alaa kumulatiivisesti yhteen. Mikäli eroarvo on negatiivinen, laskenta alkaa vähentämään kumulatiivista summaa ja säätimen lähtö alkaa pienentyä. Tästä muodostuu P-termin kanssa PI-säädön ohjaus. Useimmissa säätimissä integrointi termi ilmoitetaan aikavakiona eli kuinka kauan säätöpoikkeaman poistamiseen kuluu aikaa. Kirjain lyhenteenä käytetään T_i ja yksikkönä yleisesti sekunnit, mutta tämäkin valmistaja kohtaisesti. (Savolainen & Vaittinen 2003, 38.) Yhtälössä 2 on esitetty PI-säätimen algoritmi.

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

missä K_p = vahvistus

T_i = integrointi aika

e = eroarvo

U = säätimen lähtö

(Savolainen & Vaittinen 2003, 38.)

4.3.3 Derivoiva säätö

Derivoivaa termiä D voidaan käyttää P ja PI -termien kanssa. Yleisempi näistä kahdesta on PID-yhdistelmä. D-termin vaikutus säädössä kohdistuu nopeisiin muutoksiin erosuureessa, puhutaan ennakoivasta säädöstä. Derivoiva säätö ikään kuin ennakoiki kun se laskee hetkellisen kulmakertoimen eroarvolle. Kuten integroivassa säädössä myös derivoivassa säädössä derivoivaa termiä kuvataan aikavakioina. Aikaa jolloin säätö saa enemmän vahvistusta mitä normaalisti se saisi eroarvon muutokseen nähden. (Savolainen & Vaittinen 2003, 38.) Yhtälössä 3 on esitetty PID-säädin, jonka lopussa on derivoiva osa.

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

missä K_p = vahvistus

e = eroarvo

U = säätimen lähtö

T_i = integrointi aika

T_d = derivointi aika

(Savolainen & Vaittinen 2003, 59.)

4.3.4 Ideaalisäädin

Ideaalirakenteinen säädin on yleisin tapa toteuttaa säädin, mutta säätimillä on valmistaja kohtaisia eroja toiminnoissa ja laskentatavoissa. Ideaalisäätimessä vahvistuksella kerrotaan integrointi, eroarvo sekä derivointi. (Harju & Marttinen 2000, 59.) Ideaalisäädin on esitetty derivoiva säätö, kappaleen alla yhtälössä 3.

4.3.5 Sarjasäädin

Sarjasäädin on harvinaisimmin käytetty rakenne. Sarjasäädintä on käytetty mekaanisesti toteutetuissa säätimissä kuten pneumaattisissa säätimissä. Jos T_d/T_i suhde on pieni on sarjasäädin verrattavissa ideaalisäätimeen tai jos D-termiä ei käytetä on kyseessä ideaalisäädin. (Harju & Marttinen 2000, 60.) Sarjasäädin on esitetty alla yhtälössä 4.

$$U(t) = K_p \left[Z(t) + \frac{1}{T_i} \int Z(t) dt \right] \quad (4)$$

$$Z(t) = e(t) + T_d \frac{de(t)}{dt}$$

Missä K_p = vahvistus

e = eroarvo

U = säätimen lähtö

T_i = integrointi aika

T_d = derivointi aika

(Harju & Marttinen 2000, 60.)

4.3.6 Rinnakkaissäädin

Rinnakkaissäätimellä on kaikki parametrit rinnakkain ja ne lisätään toisiinsa. Tässä tilanteessa vahvistuksella ei kerrota muita arvoja. (Harju & Marttinen 2000, 9.) Rinnakkaissäädin on esitetty alla yhtälössä 5.

$$U(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (5)$$

missä K_p = vahvistus

e = eroarvo

U = säätimen lähtö

K_i = integrointi vahvistus

K_d = derivointi vahvistus

(Harju & Marttinen 2000, 60.)

4.4 PID_Compact-säädin

Siemens esittää omassa manuaalissaan PID_Compact säätimen algoritmin Laplace-tasossa, toisin kuin aiemmat esimerkit, jotka on esitetty aikajatkuvina funktioina. Tämän takia yhtälössä 6 on Laplace termi S . Laplace-tasossa $1/S$ tarkoittaa aikajatkuvan funktion integrointia ja $S/(S+1)$ derivointia. Yhtälöstä voidaan kuitenkin nähdä säätimen olevan ideaalisäätimen kaltainen, pienillä lisäyksillä. Yhtälössä kaikki termit ovat auki eikä esimerkiksi eroarvoa näydetä pelkkänä kirjaimena vaan se on purettu auki, jotta suunnittelija tietää miten eroarvo muodostetaan kyseisessä säätimessä. Yhtälöä täytyy kuitenkin tarkastella lähemmin, jotta pystytään sanomaan, miten esimerkiksi integrointi ajan lisääminen vaikuttaa säätöön.

Eroarvo on kaavassa $w-x$. Säätimessä on kuitenkin kerrottu asetusarvoa w , joko b tai c -termillä. B ja c -termit ovat painokertoimia joidenka vaikutus vaihtelee välillä 0-1. Painokerroin vaikuttaa siihen kuinka voimakkaana asetusarvon muutos näkyy ohjauksessa. Toisin sanoen voidaan valita asetusarvon muutoksessa P ja D -termit pois käytöstä. Painokertoimet eivät kuitenkaan vaikuta, mikäli mittauksessa tapahtuu muutos, vain jos käyttäjä antaa uuden asetusarvon. (Siemens TIA Portal V13 SP1 n.d.)

$$U = K_p \left[(b * w - x) + \frac{1}{T_i * S} (w - x) + \frac{T_d * S}{a * T_d * S + 1} (c * w - x) \right] \quad (6)$$

missä U = Säätimen lähtö

K_p = Vahvistus

S = Laplace muuttuja

b = Vahvistuksen painotuskerroin

w = Asetusarvo

x = Mittaus

T_i = Integrointi aika

T_d = Derivointi aika

a = derivoinnin viivekerroin

c = derivoinnin painokerroin

(Siemens TIA Portal V13 SP1 n.d.)

Yhtälön tarkemmasta tarkastelusta saadaan selville miten eri termit vaikuttavat säätimien. Säätimen ollessa ideaalisäädin tiedetään vahvistuksella kerrottavan kaikki termit, jonka jälkeen kaikki lasketaan yhteen. Säätimen toiminnassa eniten kiinnostaa integroinnin ja derivoinnin vaikutuksen suunta, joten niiden kaavat on hyvä avata tarkemmin. Yhtälössä 7 on kerrottu integrointi ja derivointi termien yhtälöt vahvistuksella K_p .

$$I = \frac{K_p * 1}{T_i * S} \quad D = \frac{K_p * T_d * S}{a * T_d * S + 1} \quad (7)$$

Missä I = integrointi osuus

D = derivointi osuus

K_p = vahvistus

T_i = integrointi aika

T_d = derivointi aika

a = derivoinnin viivekerroin

S = Laplace muuttuja

Yhtälöstä 7 nähdään vahvistuksen suhde integrointi- ja derivointiaikaan. Kasvattamalla vahvistusta suurennetaan osoittajassa olevaa K_p -termiä, toisaalta taas integrointi-aikaa T_i pienentämällä voidaan myös suurentaa vahvistuksen vaikutusta I-termiin. T_d -termiä taas pienentämällä saadaan suurennettua D-termin vaikutusta derivointi ajan sijaitessa osoittajassa. Termien arvot pystytään määrittämään vasta säätimen viritysvaiheessa, mutta ennen sitä on tärkeää tietää viritettävän säätimen toiminta.

4.5 Viritysmenetelmät

4.5.1 Virityksen tavoite

Usein säätöpiirien huonotoiminta johtuu huonosta virityksestä. Suurena tekijänä on myös toimilaitteen kulumisesta aiheutuvat muutokset säätöpiiriin ja sen seurauksena parametrien huononeminen. Säätöpiirin virittäminen saattaa tapahtua vain käyttöönotto vaiheessa ja silloinkin valmiilla parametreilla ilman varsinaista virittämistä. Kunnollista viritystä ei tehdä välttämättä ollenkaan ja tämän seurauksena automaatiojärjestelmästä ei saada täyttä hyötyä irti. (Harju & Marttinen 2000, 112.)

Jokaisen säädettävän kohteen ominaisuudet poikkeavat toisistaan ja samalla myös säädön tavoitteet. Virittämisessä on lähdettävä ensin liikkeelle siitä mitä halutaan ja millaisissa toleransseissa. Toisessa kohteessa voi tavoitteena olla nopea säätötulos pienellä tarkkuudella, kun taas toisaalla hitaampi säätö tarkemmalla tuloksella. Virittämistä ennen tulee myös ottaa huomioon järjestelmän dynamiikka, epälineaarisuudet, häiriöt ja muut epävarmuudet. Hyvin viritetyn säätöpiirin pitäisi kuitenkin antaa riittävä:

- Tarkkuus säätöön
- Nopeus uuteen asetusarvoon
- Kuormitushäiriöiden kompensointikyky
- sietokyky mittauskohinaan

Jotta kaikki vaatimukset täyttyisivät ja säädöstä tulisi täydellinen täytyy tuntea prosessin ominaisuudet ja vaatimukset. Esimerkiksi viiveellinen tai integroiva prosessi

olisi hyvä tunnistaa jo ennen virittämisen aloittamista. (Harju & Marttinen 2000, 107.)

Säätimen virittämiseen on suuri määrä erilaisia matemaattisia menetelmiä kuten Ziegler-Nichols, Chien-Hrones-Reswick, Cohen-Coon, Lamda, jne. Tässä työssä esitellään tarkemmin kaksi ensimmäistä, sillä ne ovat työn kannalta oleelliset. Monet piirit kuitenkin viritetään yritys ja erehdys- tyyliä. Tyyli voi toimia joissakin tilanteissa, mutta joissain prosesseissa se vie liian kauan aikaa ja on myös liian työläs. Viritysmenetelmillä päästäänkin parempiin tuloksiin ja parametrit pystytään laskemaan prosessin ominaisuuksia mittaavalla testillä. (Harju & Marttinen 2000, 114.)

Testeillä päästään hyviin tuloksiin, mutta on hyvä ymmärtää miten P, I, D -termit vaikuttavat säätimeen, jotta pystytään tekemään viimeinen hienosäätö. Karkeasti termien vaikutus prosessiin on kuvattuna taulukossa 1. (Harju & Marttinen 2000, 114.)

Taulukko 1. P, I, D -termien vaikutus

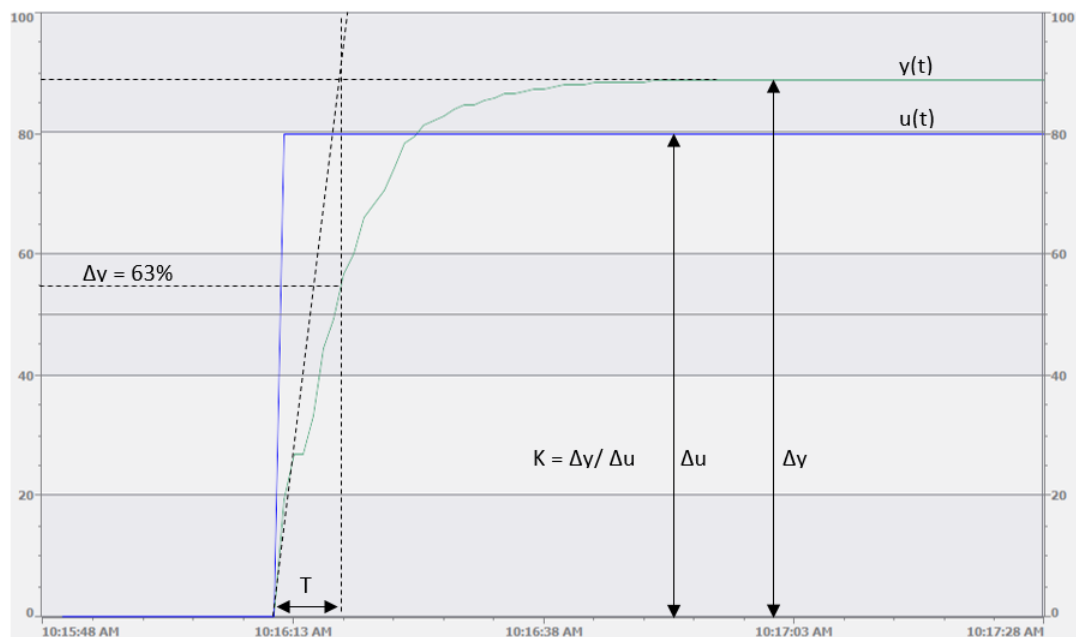
Termi	Muutos suunta	Nopeus	Stabiilius
K_p	Kasvaa	Kasvaa	Vähenee
T_i	Kasvaa	Vähenee	Kasvaa
T_d	Kasvaa	Kasvaa	Kasvaa

(Harju & Marttinen 2000, 114.)

4.5.2 Ziegler-Nichols

Ziegler-Nicholsin viritysmenetelmät on ensimmäistä kertaa esitelty vuonna 1942. Vaikka menetelmät ovat vanhoja, ovat ne silti hyvin tunnettuja säätötekniikassa ja niitä käytetään edelleen. Menetelmiä on kaksi, ensimmäinen perustuu askelmaiseen muutokseen järjestelmässä ja toinen jatkuvaan värähtelyyn. Jälkimmäistä menetelmää käytetään hieman muunneltuna monissa automaattisissa virityksissä. (Harju & Marttinen 2000, 115.)

Askelvastemenetelmässä säädin on manuaalilla eli se ei saa takaisinkytkentää, prosessin tulee myös olla stabiili ennen kuin askelvaste voidaan toteuttaa. Tämän jälkeen voidaan toteuttaa askelmainen muutos säätimen ohjaukseen. (Harju & Marttinen 2000, 115.) Askelmäisestä muutoksesta kuten kuviossa 7, voidaan laskea viive L , aikavakio T ja prosessin vahvistus K . Viive on kuviossa 7 nolla, koska asetusarvon ja mittauksen välillä ei ole viivettä, laskennassa kuitenkin on käytettävä arvoa yksi, jotta laskennasta saadaan tulokseksi nollaa. Kuviossa 7 on malliesimerkki millainen askelmainen muutos voisi olla, mutta jokainen prosessi muodostaa omanlaisensa käyrät. Jotta päästäisiin parhaimpaan tulokseen tulisi säätimen piirtämiä käyriä pystyä tutkimaan mahdollisimman tarkasti. Mallintamiseen on olemassa erilaisia ohjelmia ja esimerkiksi tässä työssä käyrät pystytiin piirtämään kuvitteelliseen valvomoon TIA Portal -ohjelmalla.



Kuvio 7. Arvojen laskenta askelvastekokeesta

Kokeen tuloksena saadut arvot voidaan sijoittaa taulukkoon 2, josta tämän jälkeen saadaan säätimelle viritysparametrit. Saadut parametrit eivät välttämättä ole täydelliset ja niitä tulee vielä hienosäätää tilanteen mukaan.

Taulukko 2. Ziegler-Nicholsin askelvastemenetelmästä laskettavat parametrit

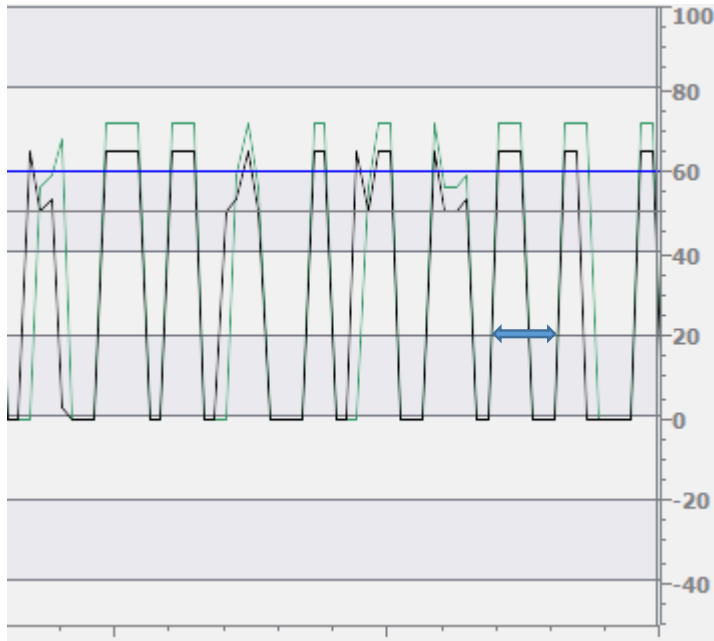
	K_p	T_i	T_d
P	$T/K*L$		
PI	$0,9*T / (K*L)$	3L	
PID	$1,2*T / (K*L)$	2L	L / 2

(Harju & Marttinen 2000, 115.)

Toinen Ziegler-Nicholsin viritysmenetelmä on kriittisenvahvistuksen menetelmä.

Tämä eroaa suuresti ensimmäisestä menetelmästä, sillä askelmaisen muutoksen sijasta haetaan prosessille pistettä, jossa se alkaa värähdellä. Menetelmä suunniteltiin ensin kuormitushäiriöiden kompensointiin ja oikeassa prosessissa asetusarvon muutoksesta tuleekin 20-30 % ylitystä. Menetelmää hyödynnetään nykyisin monissa automaatti virityksissä. Automaattinen viritystoiminto tekee pulssimaista muutosta säätimen lähtöön ja saa aikaan värähtelyä, josta taas pystytään laskemaan viritysparametrit. (Harju & Marttinen 2000, 115-116.)

Jotta kriittisenvahvistukseen perustuva viritys voidaan tehdä, täytyy ensimmäiseksi ottaa käyttöön pelkkä P-säätö, säätimeen voi kuitenkin valita muutkin termit lopuksi, mutta testi tulee tehdä pelkällä P-termillä. Tämän jälkeen aletaan vahvistuksen arvoa kasvattaa. Vahvistusta voidaan kasvattaa siihen asti, kunnes piiri värähtelee tasaisella amplitudilla ja taajuudella. Mikäli piiri alkaa värähtelemään hallitsemattomasti täytyy vahvistusta pienentää, tarkoitus on löytää kriittinen vahvistus K_{kr} ja kriittinen jaksonaika T_{kr} . (Harju & Marttinen 2000, 115.) Kun saavutetaan tilanne, jossa värähtely on tasaista kuten kuviossa 8, voidaan kuviosta laskea jaksonaika.



Kuvio 8. Tasainen värähtely

Nyt kun tiedetään kriittinen vahvistus ja jaksonaika pystytään arvot sijoittamaan taulukkoon 3. Sijoittamalla taulukkoon 3 saadut arvot voidaan laskea viritysparametrit.

Taulukko 3. Ziegler-Nicholsin kriittisenvärähtelyn menetelmästä laskettavat parametrit

	K_p	T_i	T_d
P	$K_{kr} / 2$		
PI	$K_{kr} / 2,2$	$T_{kr} / 1,2$	
PID	$K_{kr} / 1,7$	$T_{kr} / 2$	$T_{kr} / 8$

(Harju & Marttinen 2000, 115.)

Menetelmät toimivat parhaiten välillä $0,15 < (L / T) < 0,6$. Monesti Ziegler-Nicholsin viritysmenetelmällä saadut parametrit alkavat värähdellä liian helposti, varsinkin viiveellisissä prosesseissa tulokset ovat huonoja. (Harju & Marttinen 2000, 116.)

4.5.3 Chien-Hrones-Reswick

Chien-Hrones-Reswick viritysmenetelmä on saman tyylinen kuin Ziegler-Nicholsin askelvastemenetelmä. Erona on vahvempi ote säätämiseen ja sillä haetaan nopeaa vastetta ilman ylitystä tai ylityksen kanssa. Kuviosta 7 voitaisiin hakea parametrit piirtämällä tangentti askelvasteen jyrkimpään kohtaan, tangentin ja aika-akselin leikkauspistettä merkitään kirjaimella L, kun taas amplitudiakselin leikkauspistettä kirjaimella a. Taulukossa 4 on esitetty säätöparametrien laskentakaavat tilanteeseen, jossa ei haluta ylitystä ja tilanteeseen jossa halutaan 20 % ylitys. (Savolainen & Vaitinen 2003, 63.)

Taulukko 4. Chien-Hrones-Reswick menetelmästä laskettavat parametrit

	Ei ylitystä			20%:n ylitys		
P	$0,3*a$			$0,7*a$		
PI	$0,3*a$	$4*L$		$0,7*a$	$2,3*L$	
PID	$0,3*a$	$2,4*L$	$0,42*L$	$1,2*a$	$2*L$	$0,42*L$

(Savolainen & Vaitinen 2003, 63.)

5 Opinnäytetyön toteutus

5.1 Säätimen integroiminen

Uuden säätimen PID_Compactin sovittaminen eli integroiminen toimilohkoon vanhan säätimen tilalle alkoi säätimeen ja ohjelmistoympäristöön tutustumisella. Testiympäristönä toimi vanha projekti, jossa oli käytetty säädintoimilohkoja. Säätimen integroiminen tuli saada simuloinnissa toimimaan, jotta sitä voitaisiin lähteä testaamaan oikeaan ympäristöön. Projektiin tuli luoda kuvitteellinen mittausta, jotta säädin pystyisi säätämään edes kokeellisesti. Kuvitteellinen mittausta toteutettiin kopioimalla säätimen lähdön arvo takaisin säätimen mittaukseen. Lisäämällä mittaussignaalin viivettä saatiin aikaan tilanne, jossa säädin saa kuvitteellisen mittaustuloksen ilman fyysistä mittausta.

Säätimen luomisen yhteydessä ohjelma luo automaattisesti Tegnology objectin, jota ei projektiin haluttu. Objekti mahdollistaa säätimen virittämisen ohjelmointitilassa, valmiin Siemensin kehittämän parametri-ikkunan välityksellä. Parametri-ikkuna oli visuaalisesti helppokäyttöisen oloinen ja sen kautta pystyttäisiin virittämään säädin ja simuloimaan signaaleja samalla. Ongelmaksi tulee kuitenkin heti tilanne, jossa säätimiä olisi useampi. Jos jokainen säädin loisi yhden ylimääräisen DB-lohkon, olisi muistia mennyt kaksinkertainen määrä. Toimilohkon kaikki tieto haluttiin myös säilyttää yhdessä DB:ssä toiminnan ja käytettävyyden takia. Objektin luominen pystyttiin välttämään tekemällä säätimestä multi-instanssi eli säädin käyttäisi toimilohkon omaa DB-lohkoa. Säätimen parametrien syöttämien tapahtui tämän jälkeen suoraan toimilohkon DB-lohkoon.

Itse toimilohkoon säätimen sijoittaminen onnistui ilman hankaluuksia, jos ei alussa mukana ollut Tegnology objektia lasketa. Säädin ei varaa offset-arvoa itselleen ollelleen toimilohkon DB:sta, joten muiden toimintojen osoitteistus ei mennyt sekaisin, vaikka uusi säädin sisälsikin enemmän ominaisuuksia kuin vanha säädin.

Aikaisemmalla Cont_C-säätimellä tila valitaan manuaali- ja automaattiohjauksen välillä. Tilan vaihtaminen tapahtuu tuomalla säätimelle 1- tai 0-arvon, kun taas uudessa säätimessä on tiloja yhteensä kuusi erilaista ja niiden valinta tapahtuu syöttämällä säätimelle arvo välillä 0-5. Tämän jälkeen tulee myös tuoda tilan aktivointitieto, minkä jälkeen tila vasta vaihtuu.

Säätimellä on yhteensä kuusi erilaista tilaa:

- inactive (Ei aktiivinen)
- pretuning (Esisäätö)
- finetuning (Hienosäätö)
- automatic mode (Automaattitila)
- manual mode (Manuaalitila)
- substitute output value with error monitoring (Vikatila).

Inactive-tila, suomennettuna ei aktiivinen tila, tarkoittaa säätimen tilaa, jossa lähtö asetetaan aina nolnaan. Tähän tilaan siirrytään, mikäli säädin havaitsee virheen eikä automaattipalautusta ole asetettu päälle. (Siemens TIA Portal V13 SP1 n.d.)

Pretuning, suomennettuna esisäätö, tekee askelmaisen muutoksen säätimen lähtöön ja laskee siitä tulleesta "heijastumasta" säätöarvot. Säätöarvot lasketaan maksimi

nousunopeudesta ja prosessin kuolleesta ajasta. Pretuning on mahdollista asettaa päälle vain, jos seuraavat ehdot täyttyvät (Siemens TIA Portal V13 SP1 n.d.):

$$|Asetusarvo - Mittaus| > 0,3 * |Mittausalue| \text{ ja}$$

$$|Asetusarvo - Mittaus| > 0,5 * |Asetusarvo|$$

Esisäädön laskenta perustuu Chien-Hrones-Reswick säätöalgoritmiin, joka tunnetaan paremmin nimellä askelvastekoe. Valittavissa on joko PID- tai pelkästään PI-säädin. (Siemens TIA Portal V13 SP1 n.d.)

Finetuning, suomennettuna hienosäätö, luo vakiovärähtelyä lähdön avulla ohjattavaan prosessiin. Säätimen parametrit lasketaan syntyneen värähtelyn taajuudesta ja amplitudista. Hienosäädöllä saaduilla parametreilla on parempi häiriöiden sietokyky sekä ne sopivat paremmin pääsäätimelle. Hienosäädöllä pystytään hakemaan säätöparametrit, mikäli piiri on stabiili eikä värähtele. (Siemens TIA Portal V13 SP1 n.d.)

Käyttäjä määrittää parametrien laskentaan käytettävän tavan kuudesta erilaisesta menetelmästä:

- PID automatic
- PID rapid
- PID slow
- PID Ziegler-Nichols
- PI Ziegler-Nichols
- P Ziegler-Nichols

Menetelmät laskevat erilaisia parametreja nopeasta hitaaseen ja PID-säätimestä P-säätimeen. Kaikki menetelmät kuitenkin käyttävät samaa värähtelyyn perustuvaa viritysmenetelmää, laskemalla siitä vain erisuuruisia parametreja.

Kuudes säätimen tila on vikatila, mihin säädin siirtyy, mikäli automaattiajon aikana havaitaan virheitä. Virheen tullessa voidaan tilanteeseen reagoida jättämällä viimeisin ohjaus voimaan tai vaihtamalla ennalta määrätty arvo ohjaukseen. Virheestä palaututaan automaattisesti virheen poistuttua säätimeltä. (Siemens TIA Portal V13 SP1 n.d.)

Uusi säädin poikkesi vanhasta eniten uusilla ominaisuuksillaan: Vanha säädin toimi säätimenä, jolle tuli itse antaa parametrit, kun uudessa säätimessä oli perinteisen käsin virittämisen tilalle tuotu virityksen avustus. Säätimen virittäminen tapahtuu valitsemalla säätimen tilaksi esisäätö tai hienosäätö. Näillä haetaan sopivat säätöarvot automaattitilaan. Automaattitilaan pääsee myös ilman virittämistä, jolloin säätöparametrit täytyy syöttää käsin, jotta säädin olisi stabiili.

5.2 Toimilohkon toteutus

Toimilohko oli tehty vuosia aikaisemmin ja sitä oli jalostettu projektien aikana paremmaksi. Mitään varsinaista pohjaa ei ole ollut, vaan toimilohko oli muokattu edellisistä projekteista ja tehty sovitukset tarpeen vaatiessa. Toimilohkoon oli kuitenkin lisätty jokaisen projektin yhteydessä projektiin liittyviä toimintoja, joten toimilohko sisälsi myös turhia ohjelmaosuuksia. Tarkoituksena olikin karsia kaikki ylimääräinen ja tehdä toimilohkosta mahdollisimman yksinkertainen ja tehokas. Toimilohkon on tarkoitus toimia mittausloimilohkon kanssa yhdessä ja saada sieltä säädettävä mittauservo, hälytykset sekä muita tietoja mittauksesta.

Koska säädin pystyy itse skaalamaan mittauksensa, syntyi idea mittaus- ja säätötoimilohkon yhdistämisestä. Toimilohkojen yhdistäminen vähentäisi toimilohkoja ohjelmasta ja samalla suunnittelijan työtä. Toimilohkojen yhdistäminen onnistuikin, ja myös valvomosta saatiin vähennettyä sivuja. Yhdistetystä toimilohkosta päätettiin kuitenkin luopua, sillä säätimen skaalaamaa mittausta ei pystytty käsittelemään järkevästi. Mittausloimilohko oli paljon monipuolisempi ja mahdollisesti skaalauksen ohella mittauksen suodatuksen ja häiriönpoiston. Tämän johdosta säätö- ja mittausloimilohkot pysyivät erillään, säätöloimilohko omaa kuitenkin edelleen skaalauksen.

Toimilohkon rakenne pysyi hyvin saman näköisenä kuin alun perin, toiminnot vain päivittyivät vastaamaan nykyistä tilannetta. Suurimpina lisäyksinä lohkoon tuli vahvistus, integrointi ja derivointi parametrien, säätimeen kirjoitettavien arvojen tallentamisominaisuus sekä automaatti- ja manuaalitilan käsittely.

Säätimessä itsessään on parametrien tallentamisominaisuus jokaisen uuden virityksen jälkeen. Toiminto oli kuitenkin puutteellinen, sillä arvojen palautus ei onnistunut

kuin hetken ajan virittämisen jälkeen. Tästä johtuen parametrien käsittelyyn joutui tekemään uusia ratkaisuja, jotta parametrit saatiin palautettua milloin vain ohjelman aikana. Myös parametrien kirjoitusjärjestystä täytyi muuttaa, jotta käyttäjä sekä säädin pystyivät syöttämään molemmat uusia parametreja sekä palauttamaan niitä.

Koska toimilohko on multi-instanssi, täytyi valvomosta kirjoittaa haluttu arvo valitun säätimen DB-lohkoon. Säätimellä tulisi olla jokin offset-arvo, jotta tiedetään muisti-alueen alku, jonne halutaan kirjoittaa tai lukea. Kun tätä ei kuitenkaan säätimellä ollut, täytyi säätimelle kirjoittaminen ja lukeminen rakentaa muistipaikoilla, joilla on offset-arvo. Siirto olisi onnistunut MOVE-lohkoilla, mutta lohkon ulkoasu haluttiin pitää siistinä ja yksinkertaisena, päätettiin siirto kirjoittaa SCL-kielellä. Tällä tavalla kaikki siirrot tapahtuivat yhdessä Networkissa, eikä sitä avattaessa sivu täyty MOVE-lohkoista.

Toimilohkon sisältämät rampitukset, eroarvon laskenta ja pakko-ohjaus pysyivät ennallaan. Säätimentilan valintaa päivitettiin vastaamaan uutta säädintä, valinta oli hieman monimutkaisempi kuin aikaisemmin, kun valittavia tiloja oli puolet enemmän. Operoijan ei kuitenkaan tarvitse muuttaa säätimen tilaa kuin automaatin ja manuaalin välillä, joten muihin tiloihin siirtymiseen ei tarvinnut tehdä automaattisia toimintoja.

Toimilohkon päätoiminnot ovat:

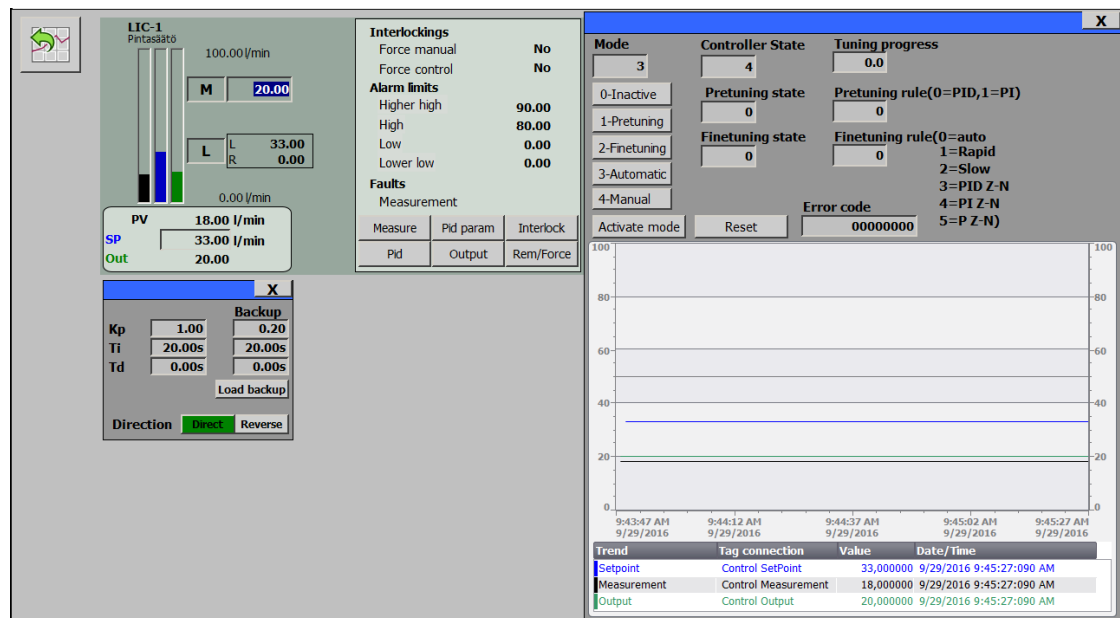
- Tietojen kopiointi mittauslohkosta
- Paikallinen/kauko-ohjauksen valinta
- Manuaali/automaattiohjauksen valinta
- Säädön rampitus
- Toimilaitteen säätö
- Pakko-ohjaus
- Eroarvon laskenta
- Parametrien käsittely
- Vika- ja häiriökäsittely

5.3 Piirinäytön toteutus

Kuten toimilohkokin oli myös piirinäyttö jo olemassa eikä sitä tarvittu rakentaa alusta asti. Valmiiseen piirinäyttöön tuli kuitenkin lisätä kaikki tarvittavat ominaisuudet,

jotta uutta säädintä pystyttäisiin hyödyntämään mahdollisimman tehokkaasti. Kaikkien objektien lisäämisessä kuitenkin tuli pyrkiä jäljittelemään mahdollisimman tarkasti alkuperäistä ulkoasua ja ohjelmarakennetta.

Ominaisuuksien suuren määrän vuoksi päätin tehdä kokonaan uuden "Pid param" ikkunan. Kuviossa 9 oikeassa laidassa on luotu ikkuna, joka avautuu Pid param -painikkeen alta. Kuvion vasemmassa laidassa pienemmällä nähdään PID-säätimen parametrien asetus-ikkuna, tänne lisättiin säätimen toimitusvalinta ja parametrien varmuuskopiointi. Toimisuunta oli aiemmin valittu negatiivisella vahvistuksella, mutta uudessa säätimessä se on valittavissa toimilohkon parametreista.



Kuvio 9. Piirinäyttö

Uuden ikkunan ominaisuuksiin kuuluu mm.

- Automaattisen virityksen valinta
- Virityksen edistymisen seuranta
- Kaikkien säätimen tilojen hallinta
- Säätimen nollaus
- Vikakoodien lukeminen
- Säädön graafinen esittäminen

Varsin kattavien ohjaustoimintojen sisältyessä yhden ikkunan alle, todettiin paremmaksi lukita ikkuna salasanan taakse. Sääntötekniikasta ja säätimen toiminnasta tietämätön käyttäjä saattaisi saada liikaa vahinkoa aikaiseksi, mikäli ikkunaan olisi esteetön pääsy. Ikkunan päätarkoitus oli myös toimia käyttöönotossa säätimen virittämisessä, joten sinne ei olisi enää tarvetta päästä normaalissa prosessiajossa.

Piirinäytön aikaisempia ominaisuuksia myös yhdistettiin uuteen säätimeen, näkymän kuitenkin pysyen muuttumattomana. PID-parametrien asetus-ikkuna pysyi vanhan näkymän mukaisena, näkymää jouduttiin kuitenkin hieman laajentaa, jotta varmuuskopiointi ja säätimen suunnan valinta saatiin mahtumaan ikkunaan. Säätimen yhtenä ominaisuutena oli vanhojen säätöparametrien tallentaminen automaattisen virityksen yhteydessä, joten nämä tallennetut parametrit oli hyvä tuoda käyttäjän nähtäväksi. Nyt vanhoja ja uusia parametreja voidaan verrata keskenään. Myös tilanteessa jossa virityksen seurauksena saadaan värähtelevät parametrit, toimii varmuuskopioiden palautus virittäjän turvana.

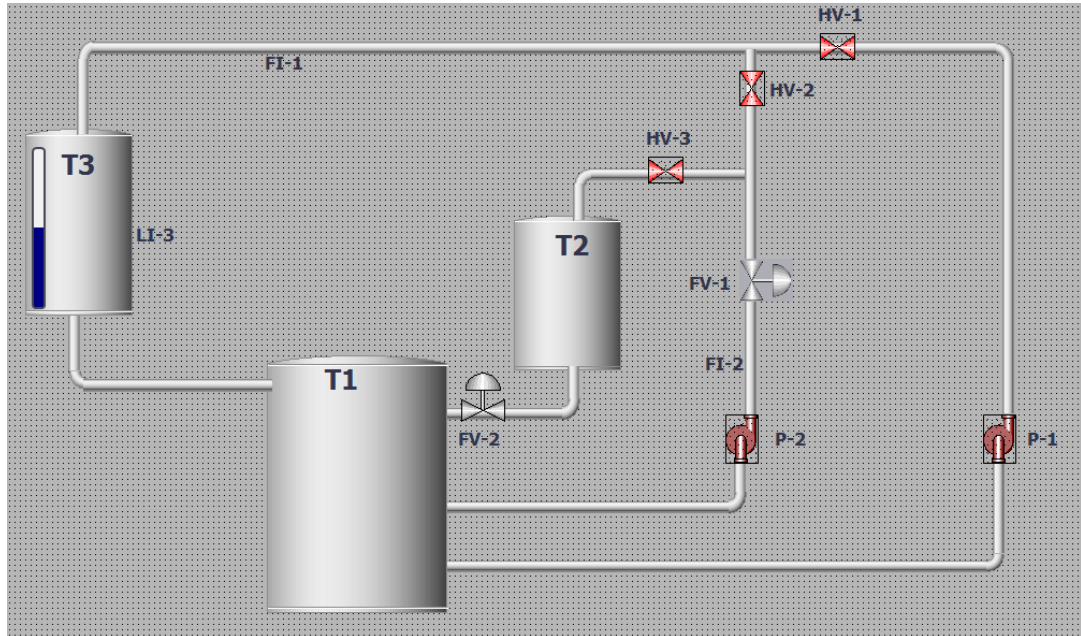
6 Säättötoimilohkon testaus

Säätimen testausympäristönä toimi Jyväskylän ammattikorkeakoulun vesiprosessi. Vesiprosessi koostuu teollisuudessa käytetyistä komponenteista ja sillä voidaan jäljitellä oikeita tilanteita pienemmässä koossa. Prosessissa kierrätetään vettä päävesialtaasta kahteen vesisäiliöön kolmen magneettiventtiin sekä kahden taajuusmuuttajaohjatun pumpun ja säätöventtiilien avulla. Prosessissa mitataan veden virtausta ja pinnan korkeutta.

Kuviossa 10 on säätimen testausta varten tehty valvomonäkymä. Testauksessa keuhkettiin, miten säätimen virittäminen onnistuu erilaisissa tilanteissa. Tilanteita oli kolme: virtauksensäätö, pinnankorkeudensäätö ja pinnansäätö kaskadisäädöllä. Virtaussäätö toteutettiin kuvion 10 venttiilillä FV-1 ja vettä ajettiin säiliöön T3 pumpulla P2. Säiliöstä T3 vesi poistuu käsiventtiin kautta takaisin vesialtaaseen T1. Jotta tilanne pysyisi samanlaisena kaikkien testien aikana, säädettiin pumppu P-2 jokaisessa testissä pumppaamaan 100 % ja käsiventtiili samaan asentoon. Tämän lisäksi tehtiin myös testejä muuttamalla pumpun tehoa ja lisäämällä toinenkin pumppu toimimaan

rinnalle. Käsiventtiilin asennolla myös saatiin aikaan häiriöitä sekä pystyttiin simuloimaan eri tavalla käyttäytyviä prosesseja.

Kokeen tärkein tavoite oli saada uusi toimilohko toimimaan halutulla tavalla sekä selvittää automaattisen virityksen jatkohyödyntämisen mahdollisuudet. Tämän lisäksi testissä nähtiin myös valvomon toimintojen riittävyys sekä muut puutteet.

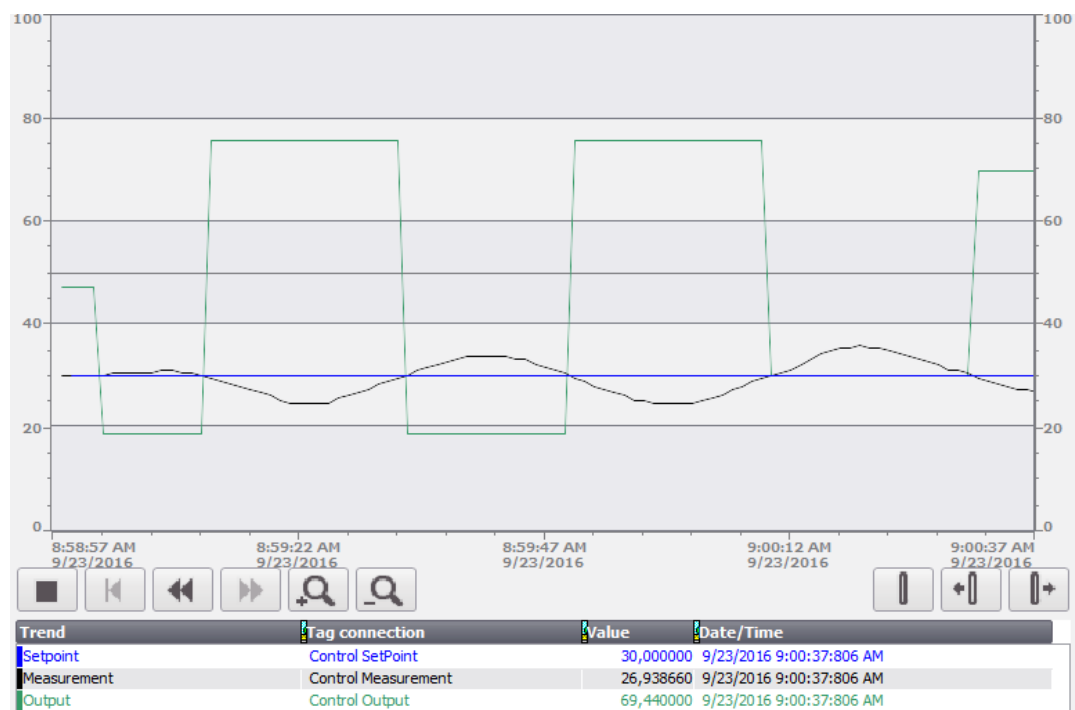


Kuvio 10. Vesiprosessi

6.1 Säätimen analysointi

Testauksen aloittamisessa ongelmia tuotti aluksi säätimen saaminen esisäätötilaan. Esisäädön ehdot on esitetty kappaleessa ”Säätimen integroiminen”, ehtojen toteuttaminen kuitenkin tuotti ongelmia, sillä esimerkiksi asetusarvolla 50 ei mittaus olisi saanut olla yli 20. Antaessa asetusarvon 50 nousee ohjaus väkisin ja samalla kasvaa mittaus, joten jotta esisäätö toimisi järkevästi täytyisi kyseessä olla hidas prosessi. Esisäätö perustui Chien-Hrones-Reswick viritysmenetelmään, jossa tehdään askelmainen muutos, säätimen muutos lähtöön oli kuitenkin 100 % mikä voi olla joillekin laitteille liian raju äkillinen heilautus. Onnistuneen esisäädön jälkeen saadut vahvistuksen parametrit olivat kuitenkin hyvin suuret ja piiri alkoi värähdellä joka kerta. Vahvistuksen puolittamisella saatettiin saada värähtely vaimenemaan, mutta parametrit olivat silti heikot.

Hienosäädön aloittamisessa ei ollut yhtä tarkkoja ehtoja kuin esisäädössä, ainut ehto oli, ettei prosessi saanut värähdellä. Prosessin värähdellessä ennen säädön aloittamista sai lasketut parametritkin epätarkkuutta eikä niillä saatu aikaiseksi stabiilia tilaa. Virittämisen aloittaminen onnistuikin joka kerta, jos ei jokin muu lukitus ollut aktiivisena. Hienosäädön viritys perustui kriittisenpisteen viritysmenetelmään. Vahvistusta ei kuitenkaan tässä kasvateta värähtelypisteeseen vaan säädin toimii releenä ja saa aikaa prosessissa aaltoimaista värähtelyä, josta se laskee aallon taajuutta ja amplitudia. Kuviossa 11 esitetty hienosäädön virityksen aaltomainen testi.



Kuvio 11. Hienosäädön värähtelytesti

Hienosäätö toimi huomattavasti paremmin kuin esisäätö ja sitä onkin käytetty kaikissa testeissä toisin kuin esisäätöä, joka on vain virtaussäädössä mukana. Säätimen säätökäyrissä (kuvioissa 6-11) nähdään säätimen asetusarvo sinisellä(Setpoint), mitausarvo mustalla(Measurement) ja säätimen lähtö eli toimilaitteen ohjaussignaali vihreällä(Output).

6.2 Virtaussäätö

Virtaussäädössä suljettiin kuviossa 10 olevat venttiilit HV-1 ja HV-3 ja käynnistettiin pumppu P-2. Virtauksen mittaukseen käytettiin virtausmittausta FI-2. Tällä tilanteella kokeiltiin, kuinka säädin hakee oikeat viritysparametrit. Säätimen viritykseen kokeiltiin ensin esisäätöä, jonka jälkeen kokeiltiin hienosäätöä. Esisäädöllä saadut parametrit alkoivat kaikki värähdellä, ne ovat esitetty taulukossa 5. Taulukosta nähdään vahvistuksen olevan aivan liian suuri verrattuna integrointi-aikaan.

Taulukko 5. Virtaussäätöön saadut parametrit, esisäätö menetelmällä

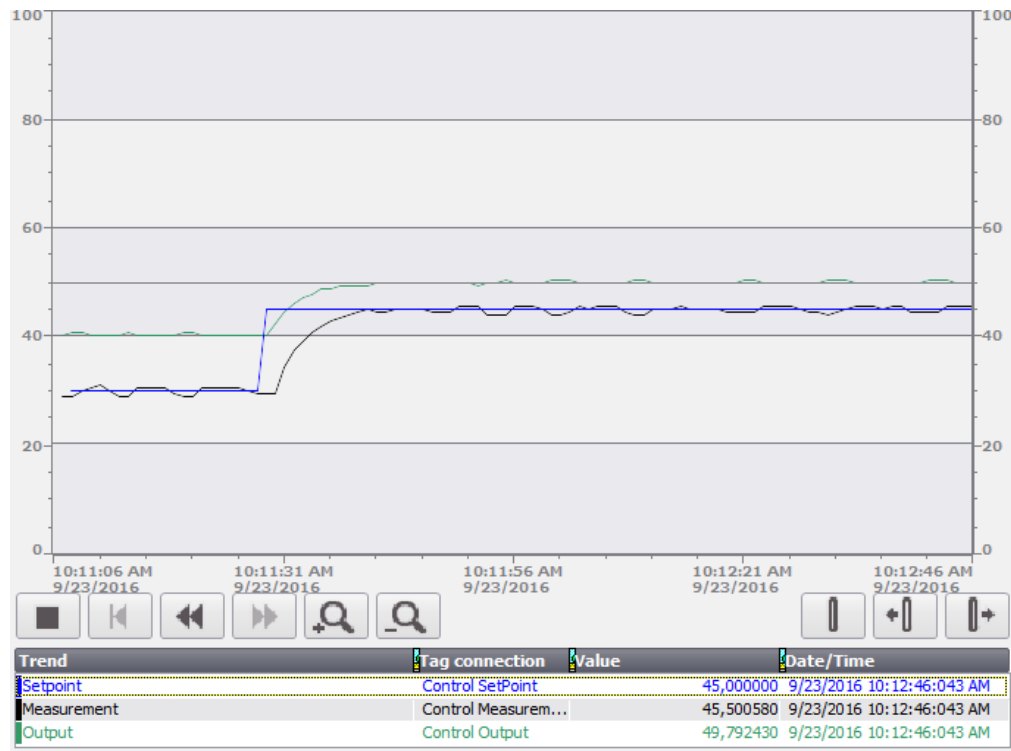
P	T_i	T_d	Menetelmä	Toiminta
9,88	0,48	0,08	C-H-R, PID	Värähteli
5,56	0,8	-	C-H-R, PI	Värähteli

Hienosäädöllä saavutettiin todella hyviä tuloksia verrattuna esisäätöön, josta ei käytännössä ollut mitään hyötyä. Virityksessä kokeiltiin kaikkia kuutta erilaista viritysmenetelmää ja niiden tulokset on esitetty taulukossa 6.

Taulukko 6. Virtaussäätöön saadut parametrit hienosäätö menetelmällä

P	T_i	T_d	Menetelmä	Toiminta
0,17	0,76	0,12	PID automatic	Ok
1,15	1,86	0,47	PID rapid	Värähteli
0,38	1,39	0,37	PID slow	Ok
0,18	0,81	0,15	Z-N, PID	Ok
0,57	4,32	-	Z-N, PI	Värähteli
0,77	-	-	Z-N, P	Ei sovellu

Taulukosta 6 nähdään hienosäädönkin epäonnistuneen nopeassa PID- ja Ziegler-Nichols PI-säädössä. Näissä tilanteissa kuitenkin vahvistuksen pienentäminen rauhoitti tilanteen. Prosessin ollessa nopea ei parametreilla saavutettu näkyvää eroa eri laisten säätöjen välillä. Säätöparametrien ollessa hyvin lähellä toisiaan olivat myös säätökäyrät hyvin identtisiä, kuviossa 12 on esitetty viritettyyn virtaussäätöön tehty asetusarvon muutos. Kuviossa 12 on Ziegler-Nicholsin PID-säädön kuvaaja parametreilla $P=0,18$, $T_i=0,81$, $T_d=0,15$.



Kuvio 12. Virtaussäädön kuvaaja

6.3 Pinnankorkeudensäätö

Pinnankorkeuden säätöä kokeiltiin kahdella eri menetelmällä. Ensimmäisenä kokeiltiin ohjata säätimellä venttiiliä suoraan ja tämän jälkeen kaskadisäätöä, jossa pinnan korkeudensäädin ohjaa virtaussäädintä.

6.3.1 Suorasäätö

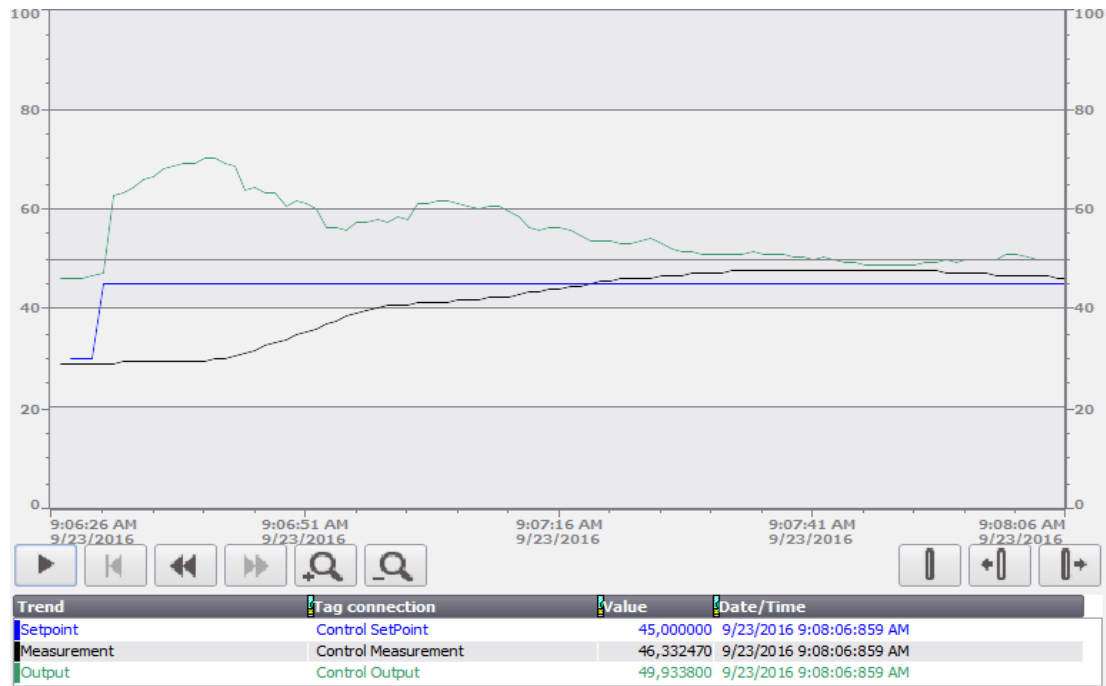
Pintasäädössä tilanne laitteiston osalta pysyi muuten samana, mutta mitattava suure vaihdettiin virtausmittauksesta pinnankorkeusmittaukseen LI-3. Pintasäädöllä ei esisäätö antanut järkeviä tuloksia, joten esisäätö jätettiin kokonaan pois pintasäädöstä. Pinnankorkeusmittaus on huomattavasti hankalampi säätää kuin virtaussäätö, sillä prosessilla on pitkä viive ennen kuin se reagoi. Pinnan korkeudensäädössä kokeiltiin myös kaikkia kuutta erimenetelmää saman lailla kuin virtaussäädössä, tulokset on esitetty taulukossa 7.

Taulukko 7. Pintasäätöön saadut parametrit hienosäätö menetelmällä

P	T_i	T_d	Menetelmä	Toiminta
3,87	22,07	5,58	PID automatic	Ok
3,35	24,48	6,2	PID rapid	Ok
1,89	28,19	6,7	PID slow	Ok
3,65	19,5	4,88	Z-N, PID	Ok
2,64	34,03	-	Z-N, PI	Ok
2,89	-	-	Z-N, P	Ok

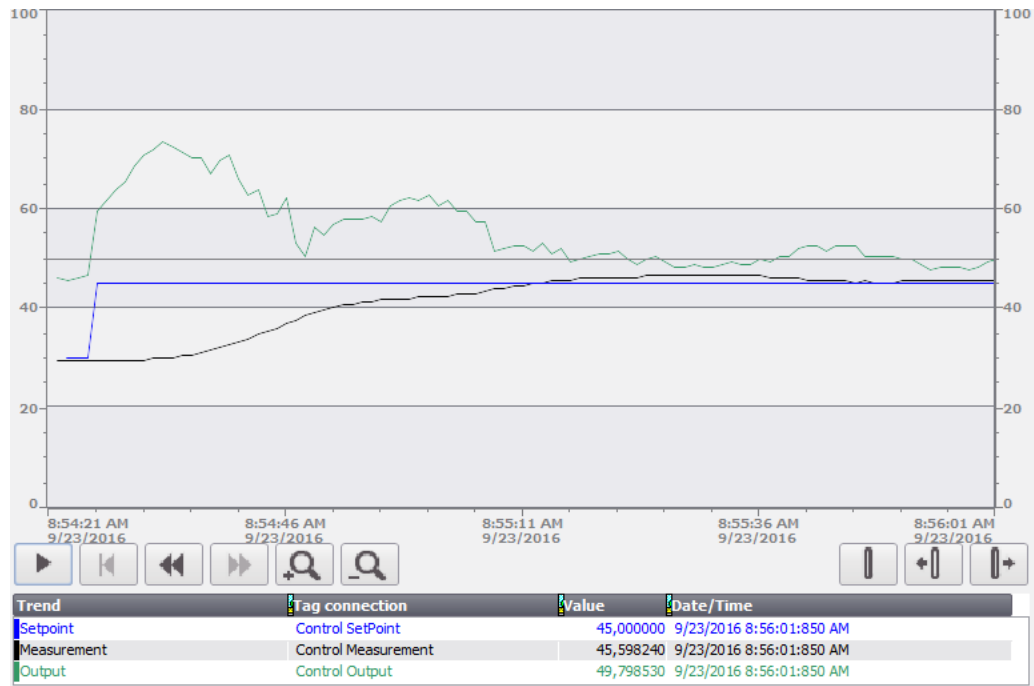
Pinnansäädössä yksilläkään parametreilla ei piiri alkanut värähdellä. Taulukon 7 parametreja tutkimalla huomataan isompia eroja vahvistuksissa ja integroinneissa ja derivaivoineissa kuin aikaisemmassa virtaussäädössä. Esimerkiksi PID-slow menetelmällä on saatu noin puolet pienempi vahvistus kuin muilla tavoilla sekä suuri integrointi aika, näin ollen säätö on siis oikeasti hitaampi.

Säätöparametreja on kuitenkin helpoin verrata prosessiin ja kokeilla miten prosessi käyttäytyy kyseisillä arvoilla. Kuviossa 13 on PID-slow menetelmällä saanut arvot ja kuviossa 14 PID-rapid menetelmällä saadut arvot. Kuvioden pitäisi siis olla toistensa vastakohtat, kuvion 13 säädin on hidas ja kuvion 14 säädin nopea.



Kuvio 13. Säätekäyrät PID-slow -menetelmällä

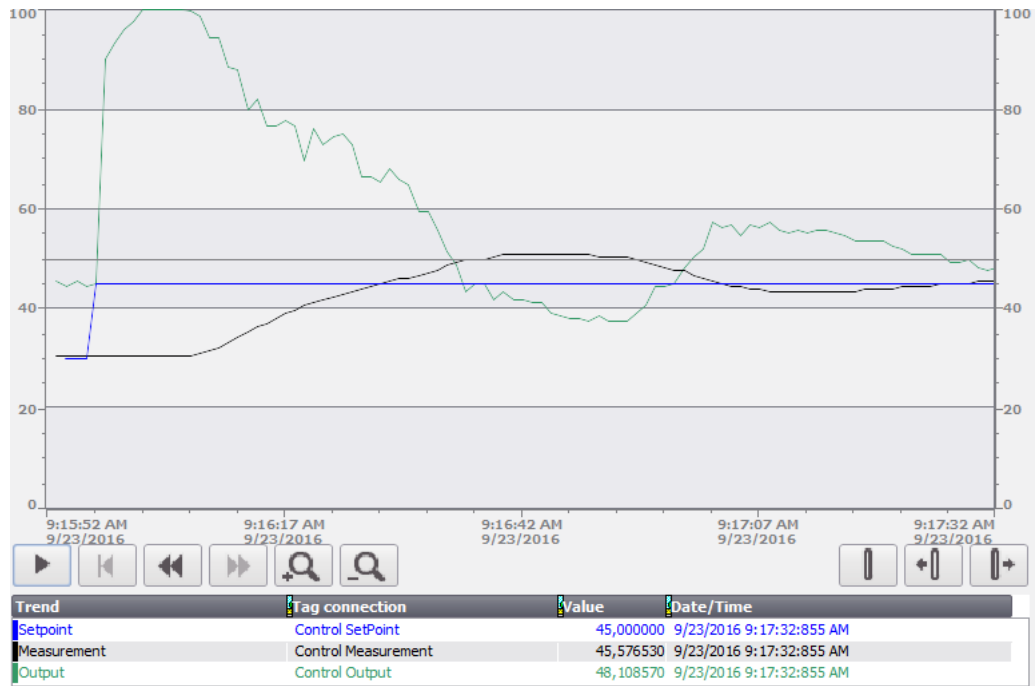
Kuviossa 13 ja 14 on säätimelle tehty asetusrvon muutos 30-45 välillä (sininen setpoint). Molemmilla arvoilla saavutetaan asetusrvo suurin piirtein samaan aikaan, nopeampi säätö vain rauhoittuu nopeammin, kun taas hitaammassa kuviossa 13 asetusrvon ylitys on hyvin loiva ja pitkä. Tutkiessa taulukkoa 7 huomataan vahvistuksen ja integrointiajan olleen paljon pienempi. Derivointiaika taas on hitaammalla säätimellä hieman suurempi. PID-rapid arvoilla ei saavutettu asetusrvoa yhtään nopeammin kuin PID-slow saaduilla arvoilla, mutta säädön nopeus huomataankin paremmin seuratesa vihreää käyrää (Output). Vihreä käyrä reagoi huomattavasti nopeammin mittauksen muutoksiin ja käyrän malli onkin paljon terävämpi.



Kuvio 14. Säätekäyrät PID-rapid -menetelmällä

Ziegler-Nicholsin-PID viritysmenetelmällä saatu käyrä poikkeaa huomattavasti nopean ja hitaansäädön käyrästä. Kuviossa 15 on esitetty edellä mainittu käyrä, johon on tehty samanlainen askelmainen muutos kuin kuvioissa 13 ja 14. Kuviossa 15 nähdään, kuinka asetusarvo saavutetaan aikaisemmin kuin nopealla ja hitaalla säädöllä. Tämä voidaan todeta myös taulukon 7 arvoista, joissa vahvistus on suuri ja integrointi pieni.

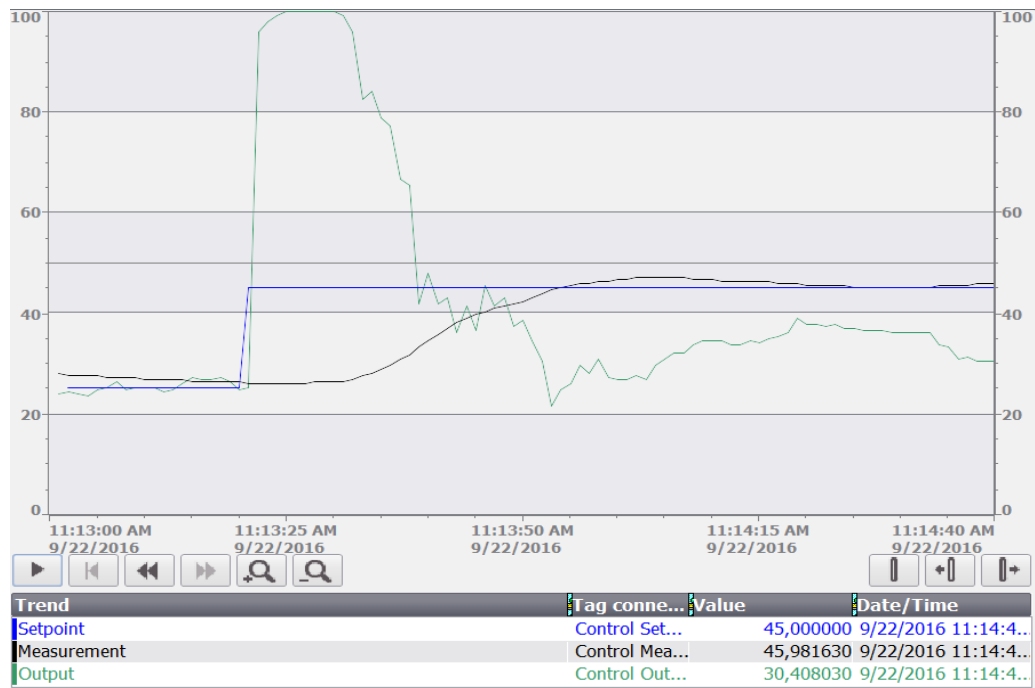
Kappaleessa Ziegler-Nichols on selitetty, kuinka parametrit muodostetaan ja todetaan menetelmän antavan 20-30 % ylityksen asetusarvoon. Niin suurta ylitystä ei kuitenkaan saavuteta vaan ylitys jää 10 %, joka voi olla joissakin tilanteissa kuitenkin melko iso. Kuviossa 15 on nähtävissä tämä ylitys, samanlainen ylitys on myös kuvioissa 13 ja 14, mutta pienempänä.



Kuvio 15. Säätkäyrät Ziegler-Nicholsin PID-viritysmenetelmällä pintasäädössä

6.3.2 Kaskadisäätö

Yhtenä testinä käytettiin kaskadisäätöä pinnankorkeudensäätöön. Kaskadisäätö toteutettiin virtaussäädön tilanteella. Virtaussäädin toimi alasäätimenä ja pinnankorkeudensäädin virtaussäätimen yläsäätimenä. Virtaussäädin oli viritettyä automaattisen virityksen parametreilla menetelmällä PID-automatic. Tämän jälkeen kokeiltiin kuinka säädin osaa virittää itsensä tilanteessa, jossa säädin ohjaa toista säädintä. Kuviossa 16 on kaskadisäädön säätkäyrät Ziegler-Nicholsin PID-viritysmenetelmällä, parametreina: $P=4,32$, $T_i=24,9$ ja $T_d=6,22$. Kuvioista voidaan todeta säätimen pystyvän virittämään tilanteita, joissa on myös kaskadisäätöjä. Parametreja ei voi suoraan verrata pelkkään pintasäätöön, mutta parametrit olivat kuitenkin tilanteeseen sopivat ja säätö käyttäytyi saman lailla kuin pintasäätö paremmalla häiriöiden siedolla. Pitkän virityksajan takia ei ollut mielekästä testata samalaista tilanne kahta kertaa, ja kaskadisäädön virittämisen testaamisessa haluttiin vain tietää toimiiko se.



Kuvio 16. Säätkäyrät Ziegler-Nicholsin PID-viritysmenetelmällä kaskadisäädössä

7 Tulosten arviointi

Työn tuloksena saatiin uusittu säädintoimilohko, joka sisältää uusimman Siemensin tarjoaman PID-säätimen. Toimilohkon lisäksi on valvomo-objektit päivitetty uudelle säätimelle sopiviksi. Päätavoitteeseen, säätimen integroimiseen, siis päästiin ja tuloksena on vielä järkevästi toimiva kokonaisuus.

7.1 Tulokset

Säätimen integroiminen sovelluskirjastoon onnistui suunnitellusti. Hankaluuksina tuli säätimeen viestien kirjoittaminen ja lukeminen sekä mittaussignaalien järkevä käsittely. Säätimelle kirjoittaminen ja lukeminen jouduttiin toteuttamaan muistipaikkojen välityksellä, mikä lisäsi yhden turhan vaiheen ohjelmaan. Tämä ei kuitenkaan vaikuta ohjelman toimintaa tai mihinkään muuhun, joten tätä voidaan pitää yhtenä integrointiin tarvittavana osana.

Mittaustoimilohko ja säätötoimilohko jäivät omiin lohkoihin, vaikka ensiksi niitä aiottiin yhdistää. Säätimen mittauksen skaalaus olisi toiminut, mutta signaalia ei olisi pystynyt muokkaamaan enää skaalauksen jälkeen. Tämän johdosta yhdistämisessä ei olisi lopulta saavutettu konkreettista hyötyä ollenkaan. Toimilohkot olisi silti voinut yhdistää yhdeksi lohkoksi, mutta toisaalta kun jokainen yksittäinen mittausta on oma toimilohkonaan, olisi säätöpiirin ja joidenkin mittausten yhdistäminen saattanut vain aiheuttaa ongelmia sovelluksen teossa. Näin ollen ohjelman rakennekin pysyy johdonmukaisena.

Valmis säädintoimilohko haluttiin ensisijaisesti saada S7-1500-logiikalle toimivaksi, mutta myös selvittää mahdollisuus 1200-sarjan logiikkoihin, joita käytetään myös paljon projekteissa. Säädintoimilohko on suoraan yhteensopiva ja siirrettävissä 1200-sarjan logiikkoihin ilman muutoksia.

Työn yhtenä tavoitteena oli selvittää, miten säädin toimii ja kuinka sillä pystytään virittämään erilaisia prosesseja. Säätimen toiminta on kuvattu luvussa viisi yksityiskohteisesti. Tiedot on yhdistetty Siemensin manuaalista yhdeksi kattavaksi selvitykseksi ja käännetty suomen kielelle. Säätötekniikan teoria tukee työn teoriaosuutta, joten näiden tietojen pohjalta pystyy säätimen toiminnan ymmärtämään ja tämän jälkeen virittämään.

Automaattisen virittämisen osalta työn voidaan sanoa onnistuneen vaikkakin esisäädön jatkokäyttö jäi hieman olemattomaksi. Hienosäädöllä säädin osaa hakea säätöparametreja ainakin kahteen yleiseen säätöön: virtaus- ja pintasäätöihin. Mikäli säädin osaa hakea säätöparametrit kyseisiin prosesseihin, voidaan olettaa säätimen toimivan myös muissakin säätötehtävissä, joissa on samankaltaisia prosesseja säädettävänä. Testien perusteella säädin uskalletaan ottaa osaksi projekteja. Pelkoina uuden säätimen vaihtamisessa olivat ohjelmalliset virheet ja huonosti virittyvä säädin. Nämä virheet pystyttiin välttämään oikeasta projektista, testaamalla säädin laboratoriooloissa.

Automaattisen virittämisen tekemät testit järjestelmään olivat nopeassa prosessissa kuten virtauksessa hyvin räväköitä, joten prosesseissa joissa toimilaitteet ovat suuria täytyy käyttää toimilohkon rampitus-toimintoa. Rampituksen ollessa päällä virtauksen virittäminen onnistui edelleen eikä venttiili joutunut tekemään nopeita liikkeitä,

säätötulos kuitenkin hieman heikkeni. Säätöön jäi pysyvää heiluntaa, jota täytyi poistaa muuttamalla käsin parametreja.

Säätimen automaattinen viritystoiminto tulee jatkossa olemaan käyttöönoton apuna. Toiminto antaa hyviä tuloksia ja sen avulla pystytään virittämään säätöpiirejä, on myös muistettava tulosten olleen välillä huonompiakin, joten säätötekniikan tunte-
musta tarvitaan edelleen. Säätimen laskemat parametrit voidaankin nähdä suuntaa antavina ja niiden perusteella voidaan lähteä vielä virittään lisää, mikäli ne eivät anna tarpeeksi tarkkaa tulosta. Vaikka automaattista virittämistä ei haluttaisi säädettävissä kohteessa käyttää toimii säädin saman lailla kuin vanhakin säädin.

7.2 Kehityskohteet

Kehityskohteita ja ideoita erilaisiin toteutuksiin tuli jo työn toteutusvaiheessa ja työn testaamisen jälkeen. Säätötoimilohko tulee tietenkin kehittymään uusien projektien myötä ja mahdollisia uusia ominaisuuksia tullaan vielä lisäämään. Toimilohkoon on kuitenkin saatu sisällytettyä tämänhetkiset vaatimukset ja uusien ominaisuuksien lisääminen on helppoa uuteen pohjaan.

Säätimen testaus vaiheessa kehityskohteena huomattiin säätimen säätökäyrien menevän samaan tallennusosoitteeseen eli mikäli vaihdetaan säätimen 1 näkymästä säätimeen 2, näkyy säätimellä ensimmäisen säätimen käyrät. Viimeisimpänäpä valittu säädin alkaa piirtää omia käyriään kuvaajaan, mutta tällöin säätimen 1 käyrät eivät piirry enää muistiin. Säätimiä onnistuu virittää yhtä aikaa, mutta mittauksen ja ohjauksen käyttäytymistä ei pystytä valvomaan kuin yhdeltä säätimeltä kerrallaan. Säätimen käyrän tallentaminen vaatisi kuitenkin muistikortin paneelille ja ohjelmaa tulisi lisää, jotta säätimien kuvaajat saataisiin tallennettua. Säätimen virittäminen tapahtuu kuitenkin pääsääntöisesti yksitellen, joten kuvaajien tallentaminen tulee ajan-kohtaiseksi vasta kun asiakas haluaa tallentaa prosessin käyttäytymistä.

Toiminnassa havaittavia ongelmia muodostui esisäädön kanssa. Sen jatkokehitys ei ole välttämätön, sillä hienosäädöllä päästiin hyviin tuloksiin. Esisäädön tekemä räväkkä askelmainen heilautus lähtöön ei myöskään sovi kovin moneen paikkaan. Mi-

käli iso toimilaite ohjataan sekunnissa kiinni ja auki, ei siitä saada mitään järkevää tulosta, eikä mikään prosessi kestä sellaista ilman hajoamista. Mikäli tilanteeseen kehiteltäisiin rampitusta ei sekään antaisi järkevää tulosta testille, sillä esisäättö käytti Chien, Hrones and Reswick-algoritmia ja siinä lasketaan nousunopeutta, joka olisi tässä tilanteessa rajoitettu rampituksella vakio arvoon.

8 Pohdinta

Säätötekniikan osaaminen ja sen yhdistäminen toimivaksi automaatiojärjestelmän kanssa voi olla haastavaa. Säätötekniikasta löytyy paljon kirjallisuutta ja sitä on mallinnettu matemaattisesti monilla eri tavoilla. Hankaluutta säätötekniikan ymmärtämiseen tekee aiheen laajuus sekä vaadittava matemaattinen taito. Säätötekniikan lisäksi tulisi hallita vielä ohjelmointia, jotta säätötekniikka voitaisiin siirtää toiminnoksi automaatiojärjestelmään.

Automaattisesti virittyvä säädin on esimerkki siitä, kuinka kehitys menee eteenpäin ja sovellussuunnittelijalle tuodaan helpotusta virittämiseen. Virittämiseen ei enää vaadita niin paljon ymmärrystä säätötekniikasta kuin ennen. Tätä voidaan pitää hyvänä kehityssuuntana, mikäli tällä saadaan enemmän hyvin viritettyjä säätöpiirejä.

Opinnäytetyössä tutkinnan alla ollut Siemensin säädin voidaan sanoa toimivan ja virittyvän automaattisesti vaadituissa olosuhteissa. Kaikilla viritysmenetelmillä ei saavutettu stabiilia lopputulosta, mutta tähän voidaan todeta, etteivät kaikki viritysmenetelmät sovi kaikkiin tilanteisiin. Eri menetelmiä kokeilemalla saavutettiin kuitenkin hyviä tuloksia. Säätimen automaattisenvirityksen kanssa tulee kuitenkin olla hieman skeptinen eikä luottaa täysin sen toimintaan. Automaattinen viritys toimii käyttöönotossa virittämisen apuna ja sen avulla voidaan saada suoraan parametrit säätimelle tai sen avulla saadaan suuntaa antavat arvot, joita hiomalla paremmaksi saadaan haluttu säätö aikaiseksi. Testien perusteella monessakin tilanteessa saatiin ajoon sopivia parametreja ilman virittäjän koskemista parametreihin.

Työn aihetta jouduttiin rajaamaan, sillä työ sisälsi niin sovellussuunnittelua kuin säätötekniikkaa ja ilman aihealueen rajaamista olisi työmäärä paisunut liian suureksi.

Säätötekniikasta ja sovellussuunnittelusta käsiteltiin kuitenkin työn kannalta oleelliset asiat sekä asia yhteyteen tärkeät osuudet. Työn teoriaosuus ja toteutus olivat työn kannalta kattavat ja toteutuksen yhteydessä suoritettu käyttöönotto on todiste työn toimivuudesta.

Lähteet

Automaatiosovellusten ohjelmistokehitys. 2005. Suunnittelun työtavat, välineet ja sovellusarkkitehtuurit. Helsinki: Painomerkki.

Harju, T. & Marttinen, A. 2000. Säättöpiirin virityksen perusteet. Espoo: Otamedia.

IEC 61131-3. 2013. Programmable controllers – Part 3: Programming languages.

JEEC etusivu. N.d. JEEC Oy:n internetsivu. Viitattu 18.8.2016. <http://www.jeec.fi/index.htm>

JEEC yritys. N.d. JEEC Oy:n internetsivu. Viitattu 18.8.2016. <http://www.jeec.fi/yritys.htm>

John, K.-H. & Tiegelkamp, M. 2010. IEC 61131-3: Programming Industrial Automation Systems: Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids. New York: Springer.

Kuisma, A. 2015. Säättötekniikan perusteita. Viitattu 20.10.2016. Jyväskylän ammattikorkeakoulu, Optima. Oppimateriaali Jyväskylän ammattikorkeakoulun tietojärjestelmässä.

Savolainen, J. & Vaittinen, R. 2003. Säättötekniikan perusteita. Helsinki: Gummerus.

Siemens TIA Portal V13 SP1. N.d. Manuaali. Viitattu 19.8.2016

Ohjelmoitavat logiikat. N.d. Artikkelisiemensin internetsivuilla. Viitattu 13.9.2016. http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/ohjelmoitavat_logiikat_simatic.php

Tehokasta automaatio-ohjelmointia S7-1500-logiikalla. N.d. Artikkelisiemensin internetsivuilla. Viitattu 13.9.2016 http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/ohjelmoitavat_logiikat_simatic/s7_1500.php

Helppoa automaatio-ohjelmointia S7-1200-logiikalla. N.d. Artikkelisiemensin internetsivuilla. Viitattu 13.9.2016 http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/ohjelmoitavat_logiikat_simatic/s7_1200.htm

Totally Integrated Automation Portal. N.d. Siemensin esite. Viitattu 13.9.2016 http://www.siemens.fi/pool/products/industry/iadt_is/tuotteet/tia_portal/tia_portal_brochure.pdf