

Tampereen ammattikorkeakoulu
Sähkötekniikan koulutusohjelma
Automaatiotekniikka
Jussi Ollila

Opinnäytetyö

**PID-säätimen virityslohkon valinta ja käyttöönotto SIMATIC
S7-300 logiikassa**

Työn ohjaaja
Työn tilaaja

Lehtori Jukka Falkman
Seilo Consulting, ohjaajana Executive Automation Consultant Anssi
Seilo

Tampere 5/2010

Tekijä	Jussi Ollila
Työn nimi	PID-säätimen virityslohkon valinta ja käyttöönotto SIMATIC S7-300 logiikassa
Sivumäärä	25 + liitteet 5 kpl
Valmistumisaika	24.5.2010
Työn ohjaaja	Lehtori Jukka Falkman
Työn tilaaja	Seilo Consulting, ohjaajana Executive Automation Consultant Anssi Seilo

TIIVISTELMÄ

Työn tavoitteena oli tehostaa voimalaitosten käyttöönottoon liittyviä tehtäviä. Työssä tutkittiin vaihtoehtoisia menetelmiä PID-säätimen automaattiselle viritykselle ohjelmoitavassa logiikassa. Menetelmistä valittiin yksi toteutettavaksi ja työn aikana tehtiin tarvittavat muutokset logiikkaohjelmistoon. Ohjelmiston toiminta varmistettiin simuloimalla erilaisia teollisuuden prosesseja. Simuloinneissa testattiin myös teollisuuden mittauksissa yleisten häiriötilanteiden vaikutuksia virityksen toimintaan.

Muutosten toteutus operointikäyttöliittymän osalta jätettiin työn aiheen ulkopuolelle. Näistä muutoksista tehtiin kuitenkin alustava suunnitelma ja työssä pyrittiin antamaan hyvät edellytykset muutosten myöhemmälle toteutukselle.

Työn aikana laadittiin viritystoiminnon käyttöohje käyttöönottohenkilökunnalle. Ohjelmisto ja ohjeet pyrittiin toteuttamaan siten, että uudelleen koulutuksen tarve olisi mahdollisimman vähäistä.

Writer	Jussi Ollila
Thesis	Choosing and implementing a PID self-tuner on SIMATIC S7-300 programmable logic controller
Pages	25 + 5 attachments
Graduation time	24.5.2010
Thesis Supervisor	Lecturer Jukka Falkman
Co-operating company	Seilo Consulting, supervisor Executive Automation Consultant Anssi Seilo

ABSTRACT

The goal of this thesis was to make the start-up of a power plant more effective. Different methods for PID auto-tuning in programmable logic controllers were investigated. One of the methods was chosen to be implemented in the logic controller software. Software was tested using simulated process models and common disturbance variables in the process industry.

Modifications for the operator's interface software weren't implemented during this thesis but specification for the changes was made. The goal was to provide preconditions for an easy implementation of the changes later.

Thesis also includes a user manual for the tuning procedure intended to be used by the commissioning personnel. Manual and software were designed in such manner that the need for retraining the personnel would be minimal.

Esipuhe

Opinnäytetyön aihe oli erittäin mielenkiintoinen ja se yhdisti tehokkaalla tavalla opiskelun aikana hankittuja tietoja ja taitoja eri oppiaineista. Tietoja pääsi myös syventämään työtä tehdessä. Opinnäytetyö liittyi olennaisesti käytännön työelämään, joka omalta osaltaan lisäsi myös työn mielekkyyttä.

Haluan kiittää Seilo Consulting:ia työn tarjoamisesta ja tuesta opinnäytetyön aikana. Lisäksi haluan kiittää työtä ohjannutta opettajaa työn sisältöön liittyvien ideoiden antamisesta.

Tampereella 24.5.2010

Jussi Ollila

Sisällysluettelo

1 Johdanto	6
2 PID-säätö	7
3 PID-säätimen viritysmenetelmät	8
3.1 Askelvastemenetelmä.....	9
3.2 Jatkuvan värähtelyn menetelmä	11
3.3 Releohjausmenetelmä	12
4 Työn toteutustavan valinta	14
5 FB50-virityslohkon toimintaperiaate	15
6 Virityslohkon käyttöönotto	17
7 Operointikäyttöliittymän muutokset	20
8 Ohjelmiston toiminnan testaus	22
9 Virityslohkon käyttöohje	24
10 Loppusanat	24
Lähdeluettelo	25

Liitteet

Liite 1: Virityksen käyttöohje

1 Johdanto

Seilo Consulting Oy on vuonna 2008 perustettu pirkkalalainen pienyritys. Yrityksen toimenkuvaan kuuluu sähkö- ja automaatioalan teknistä konsultointia ja voimalaitosten ja teollisuuden prosessien käyttöönottoja. Työkohteita yrityksellä on ollut eri puolilla maailmaa, maalla ja merellä.

Yritys työllistää tällä hetkellä kaksi henkilöä, mutta henkilöstön tarpeen odotetaan kasvavan lähikuukausina.

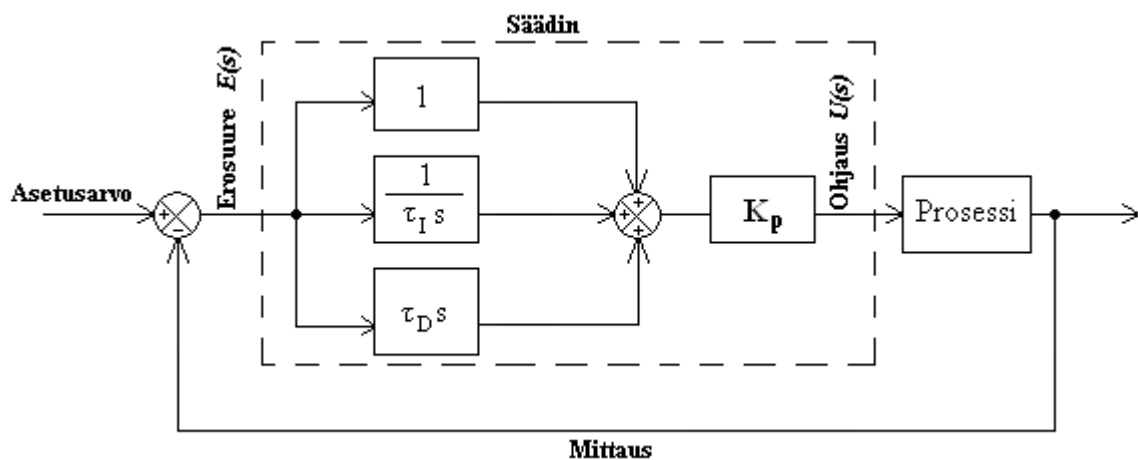
Käyttöönotto tehtävät liittyvät pääasiassa voimalaitoksiin ja niihin liitettäviin järjestelmiin. Tällaisia tehtäviä ovat mm. säätöpiirien viritykset. Erilaisia säätökohteita ovat jäähdytysvesien lämpötilasäädöt, polttoaineen paine-, viskositeetti- ja lämpötilasäädöt ja moottoreiden pakokaasujen lämmön talteenottoon liittyvät säädöt. Voimalaitoksiin voi liittyä myös ulkopuolisia järjestelmiä, kuten kaukolämpöverkosto. Kaukolämmön tuotannossa noin puolet lämpöenergiasta saadaan moottoreiden pakokaasuista. Lisäksi lämpöä saadaan lämmönvaihtimilla moottorin jäähdytysvedestä, voiteluöljystä ja ahtoilmasta, jotka jäähtyvät samalla kun kaukolämpövesi lämpiää. Käytännön toteutukset voimalaitoksilla voivat vaihdella paljonkin, mutta peruseriaatteet pysyvät enimmäkseen samoina.

Järjestelmiä ohjataan valvomo-PC:n Wonderware InTouch[®] -pohjaisella operointikäyttöliittymällä, joka on liitetty prosessia ohjaaviin ohjelmoitaviin logiikoihin. Varsinainen prosessi muodostuu voimalaitoksen moottoreista, generaattoreista ja niihin liittyvistä järjestelmistä.

2 PID-säätö

PID-säätimen nimi tulee säätimen toimintaa kuvaavista englanninkielisistä sanoista Proportional, Integral ja Derivative. Asetusarvosta vähennetään prosessin mittaussuure ja näin saatu eroosuure kerrotaan PID-säätimen termeillä. P-vaikutus aiheuttaa säätimen lähtöön erosuureeseen nähden suoraan verrannollisen muutoksen vahvistuskertoimen K_p määrittelemällä voimakkuudella. I-vaikutus integroi erosuuretta ajan suhteen, joten sen vaikutus perustuu erosuureeseen ja lisäksi myös sen keston. D-vaikutus tarkkailee erosuureen muutosnopeutta, eli se pyrkii ennakoimaan tulevia säätöpoikkeamia.

Säätimestä voidaan tehdä P-, PI-, PD- tai PID-säädin eliminoimalla ylimääräiset termit. Yleensä I- ja D-termien eliminointi onnistuu asettamalla integrointiaika τ_I mahdollisimman suureksi ja derivointiaika τ_D nolaksi. Asetukset saattavat tosin vaihdella eri valmistajien säätimissä. PID-säätimissä voidaan käyttää erilaisia säädinalgoritmeja. Niillä voidaan vaikuttaa säätimen ominaisuuksiin ja ne tulee huomioida viritysparametreja laskettaessa. Yleisimmin käytettyjen säädinalgoritmien perusmuoto on rinnakkaismuotoinen. Tällaisen algoritmin toimintaperiaate on esitetty lohkokaaviona kuviossa 1.

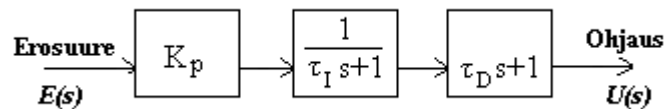


Kuvio 1: Rinnakkaismuotoisen PID-säätimen toimintaperiaate

Rinnakkaismuotoisen säädin algoritmin siirtofunktio $G_c(s)$ on seuraavanlainen:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_I s} + \tau_D s \right) \quad (1)$$

Toinen paljon käytetty algoritmi on sarjamuotoinen. Säätimen periaatteellinen lohkokkaavio on esitetty kuviossa 2 ja siirtofunktio yhtälössä (2).



Kuvio 2: Sarjamuotoisen PID-säätimen lohkokkaavio

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_I s + 1} \right) (\tau_D s + 1) \quad (2)$$

Säätimiin lisätään usein derivoivan termin suodatus tai muita poikkeamia perusalgoritmeihin nähden. Käytetty algoritmi selviää yleensä säätimen valmistajan dokumentaatioista.

3 PID-säätimen viritysmenetelmät

Säätöpiirin suunnittelussa on otettava huomioon useita tavoitteita. Yhden säädön osa-alueen parantaminen usein heikentää toista, joten täydellistä säätöä on käytännössä mahdotonta saavuttaa.

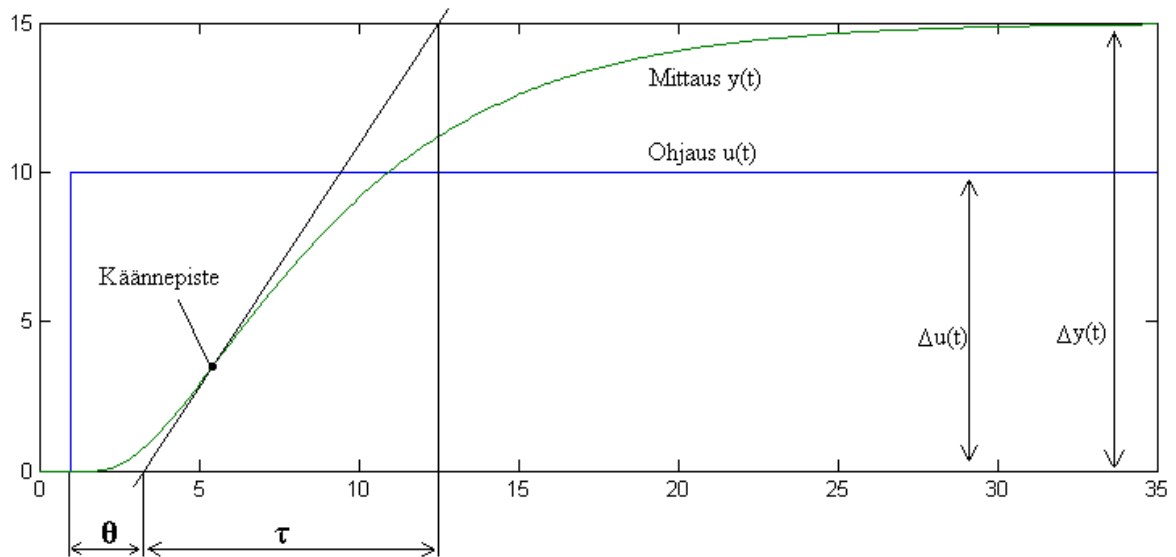
Säätöpiirejä voidaan virittää yritys ja erehdys -menetelmällä, mutta prosessista riippuen se voi olla liian työlästä ja pitkäkestoista. Erityisesti hitailla prosesseilla säädön toiminnan testaaminen useilla erilaisilla parametreilla voi kestää liian kauan ollakseen järkevää. Viritysmenetelmiä hyödyntäen on mahdollista määrittää säätimelle sopivat tai ainakin lähes sopivat viritysparametrit yhdellä prosessille tehtävällä testillä. Viritysmenetelmät mahdollistavat myös virityksen osittaisen automatisoinnin, joka voi tehostaa säätöpiirien käyttöönottoa.

Yleisesti eri vitysmenetelmissä ensimmäinen vaihe on kerätä tietoa säädettävästä prosessista, jolle voidaan tehdä tiettyjä kokeita, joiden avulla prosessista saadaan määritettyä matemaattinen malli tai yksittäisiä prosessin ominaisuuksia kuvaavia lukuarvoja. Seuraava vaihe on laskea saatujen arvojen avulla vitysparametrit säätimelle. Säädon testauksen jälkeen voi olla vielä tarvetta hienosäätää arvoja, jotta säädöstä saataisiin toivotun kaltainen. Yleensä vitysmenetelmien käyttö kuitenkin helpottaa vityttämistä huomattavasti.

John G. Ziegler ja Nathaniel B. Nichols esittelivät vuonna 1942 kaksi vitysmenetelmää PID-säätimelle. Ensimmäinen niistä on jatkuvaan värähtelyyn perustuva vitysmenetelmä (eng. Continuous Cycling Method). Jatkuvan värähtelyn menetelmää on kehitetty myöhemmin eteenpäin ja näillä uusilla menetelmillä on saatu korjattua tiettyjä ongelmia alkuperäisessä menetelmässä. Toinen Zieglerin ja Nicholsin esittelemä menetelmä on nykyäänkin paljon käytössä oleva askelvastemenetelmä.

3.1 Askelvastemenetelmä

Askelvastemenetelmässä säätöpiiri on avoimena, eli säädin on käsiajolla. Mittauksen on oltava tasapainotilassa ennen aloitusta. Ohjaussuureeseen aiheutetaan askelmainen muutos ja prosessin vasteesta voidaan lukea viiveen θ , vahvistuksen K ja aikavakion τ arvot. Näillä arvoilla prosessista voidaan muodostaa ensimmäisen kertaluvun järjestelmän viiveellinen malli ja laskea vitysparametrit. Kuviossa 2 on esitetty hypoteettisen toisen kertaluvun prosessin askelvaste. Kuviossa on piirrettynä ohjaussuure $u(t)$ sekä prosessin mittaussignaali $y(t)$.



Kuvio 2: Askelvastekoe ja apupiirroksat prosessimallin määrittämiseen

Mittauskäyrältä etsitään käänne piste eli piste, jossa käyrän kaarevuussuunta muuttuu ja piirretään sen kautta kulkeva tangentti. Prosessin viive on se aika, joka kuluu asetusarvomutoksen alkupisteestä siihen pisteeseen, jossa tangentti leikkaa X-akselin. Aikavakion määrittämiseksi piirretään apuviiva tangentin ja mittauksen huippuarvon leikkauspisteen kohdalta X-akselille. Prosessin aikavakio on se viiveen jälkeinen aika, joka kuluu tultaessa tähän pisteeseen. Vahvistus K voidaan laskea asetusarvon ja mittauksen muutosten arvoilla (3).

$$K = \frac{\Delta y(t)}{\Delta u(t)} \quad (3)$$

Sijoittamalla lukuarvot viiveellisen ensimmäisen kertaluvun järjestelmän yleiseen muotoon (4) saadaan prosessin matemaattinen malli.

$$G_p = \frac{K}{\tau s + 1} \cdot e^{-\theta s} \quad (4)$$

Viritysparemetrien laskentaan on olemassa erilaisia prosessin ja säätimen siirtofunktioihin perustuvia laskentamenetelmiä kuten λ -viritys ja IMC-menetelmä. On myös olemassa empiiristen kokeiden pohjalta määriteltyjä yksinkertaisia kaavoja. Taulukossa 1 on esitetty Ziegler-Nichols-kaavat viritysparemetrien laskentaan.

Taulukko 1: Säädinparametrien laskenta

	K_p	τ_I	τ_D
P	$\tau / (K \theta)$	-	-
PI	$0,9 \tau / (K \theta)$	3θ	-
PID	$1,2 \tau / (K \theta)$	2θ	$\theta / 2$

3.2 Jatkuvan värähtelyn menetelmä

Ziegler-Nicholsin jatkuvan värähtelyn menetelmässä säädin asetetaan P-säätimeksi, eli I- ja D-termien vaikutus eliminoidaan. Prosessi on aluksi tasapainotilassa, säädin automaattilla ja K_p asetettuna pieneksi. Asetusarvomuutoksella prosessi saadaan värähtelemään. K_p :n ollessa pieni värähtely vaimenee ja suurella K_p :n arvolla prosessi muuttuu epästabiiliksi. Tarkoitus on löytää kokeilemalla sellainen K_p :n arvo, jolla prosessi värähtelee harmonisesti. Tätä vahvistuksen arvoa kutsutaan kriittiseksi vahvistukseksi ja sitä merkitään K_{pu} :lla. Prosessin värähtelytaajuutta tällä K_p :n arvolla kutsutaan vastaavasti kriittiseksi taajuudeksi ja sitä merkitään P_u :lla.

Taulukon 2 kaavoilla lasketaan viritysparametrit eri säädintyypeille. Ziegler-Nichols-kaavat perustuvat $1/4$:n vaimennussuhteeseen, joten niillä viritetty säätö on usein aggressiivinen sisältäen suurehkon ylityksen ja värähtelyä. Vaihtoehtoisten Tyreus-Luyben-kaavojen (taulukko 2) kehityksessä on hyödynnetty Ziegler-Nichols-kaavoja. Tyreys-Lyuben-kaavoilla säädöstä saadaan yleensä hieman rauhallisempi kuin Ziegler-Nichols-kaavoilla. Kirjallisuudessa on esitelty myös monia muita kaavoja, kuten Cohen-Coon-kaavat.

Taulukko 2: Säädinparametrien laskenta

(Seborg, Edgar, Mellichamp 2004, 318)

Ziegler-Nichols	K_p	τ_I	τ_D
P	$0,5 K_{pu}$	-	-
PI	$0,45 K_{pu}$	$P_u / 1,2$	-
PID	$0,6 K_{pu}$	$P_u / 2$	$P_u / 8$
Tyreus-Luyben	K_p	τ_I	τ_D
PI	$0,31 K_{pu}$	$2,2 P_u$	-
PID	$0,45 K_{pu}$	$2,2 P_u$	$P_u / 6,3$

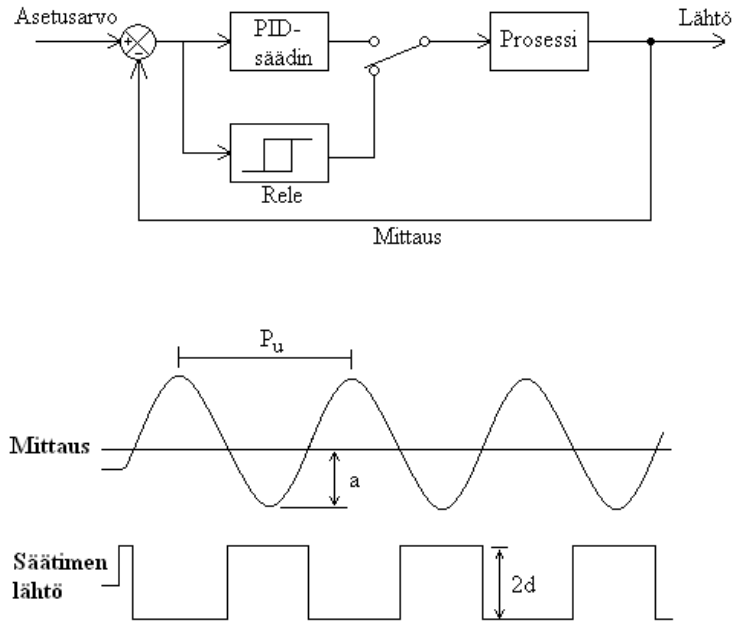
Viritysparametrien määrittämisen jälkeen ne syötetään säätimeen ja testataan säädön toiminta. Tarvittaessa viritysparametreihin tehdään vielä korjauksia.

Ziegler-Nicholsin jatkuvan värähtelyn menetelmä on käytännössä hankala automatisoida ja manuaalisesti tehtynäkin se aiheuttaa usein ongelmia tai vaaratilanteita. Menetelmässä prosessi viedään stabiilisuusrajalle ja monissa prosesseissa rajan ylitys on vaarallista. Lisäksi värähtelyn amplitudin suuruutta ei välttämättä voida päätellä etukäteen, ja on täysin operaattorin tarkkaavaisuudesta ja ennakoitavuudesta kiinni, aiheutuuko siitä prosessille haittaa.

3.3 Releohjausmenetelmä

Releohjausmenetelmän (eng. Relay Feedback Method) kehittivät Ruotsalaiset Åström ja Hägglund vuonna 1984. Se on samankaltainen kuin jatkuvan värähtelyn menetelmä, mutta releohjausmenetelmässä prosessia ei tarvitse viedä stabiilisuusrajalle ja värähtelyn amplitudia voidaan rajoittaa merkittävästi.

Menetelmässä säädin korvataan digitaalisella releellä kuvion 3 mukaisesti. Releellä aiheutetaan prosessin ohjaussuureksi kantiaaltomuotoista signaalia, jonka tila vaihtuu mittauksen ohittaessa asetusarvon. Releessä kannattaa käyttää sopivaa erottelukynnystä, jotta mittauksen häiriöt eivät aiheuttaisi säätimen nopeaa edestakaista liikettä asetusarvon läheisyydessä. Kuvio 3 havainnollistaa periaatteellisella tasolla asetusarvon ja mittauksen liikkeitä virityksen aikana.



Kuvio 3: Releohjausmenetelmän toimintaperiaate

Mittauksen ja asetusarvon värähtelyjen taajuuudet ovat yhtä suuret ja niiden taajuus on sama kuin kriittinen taajuus P_u . Testistä saadaan luettua myös kuvion 3 mukaiset amplitudit a ja d , joista voidaan laskea arvio kriittiselle vahvistukselle K_{pu} Åströmin ja Hägglundin johtamalla kaavalla (5) (Seborg ym. 2004, 321). Kaavassa käytettävät arvot tulee lukea sellaisesta kohdasta, jossa häiriöt ja releen erottelukynnys eivät ole aiheuttaneet merkittävää virhettä mittaussignaalin muotoon.

$$K_{pu} = \frac{4d}{\pi \cdot a} \quad (5)$$

K_{pu} :n ja P_u :n arvoilla sekä taulukon 2 kaavoilla voidaan laskea viritysparametrit eri säädintyypeille.

Edellä esiteltyihin menetelmiin verrattuna releohjausmenetelmä on suhteellisen helppo automatisoida ja käyttäjä voi määrittää amplitudin d prosessille sopivaksi. Lisäksi viritys voidaan suorittaa prosessin tavanomaisen toimintapisteen välittömässä läheisyydessä, toisin kuin esimerkiksi askelvastemenetelmää käytettäessä.

Viritysparametrien tarkkaan määrittämiseen tarvitaan yleensä kahdesta neljään värähtelyjaksoa. Tämä voi tehdä virityksestä liian pitkäkestoisen erityisen hitailla

prosesseilla ja silloin voi olla parempi käyttää askelvastemenetelmää. Tosin on olemassa releohjausmenetelmiin perustuvia kaupallisia automaattiviritysohjelmaa, joille viritykseen riittää jopa yksi pulssi.

4 Työn toteutustavan valinta

Työn alkuvaiheessa oli jo selvillä haluttu lopputulos, mutta toteutustapaa ei oltu valittu. Vaihtoehtoina oli oman virityslohkon suunnittelu tai valmiin kaupallisen toteutuksen integroiminen yrityksen olemassa olevaan ohjelmistoon. Alusta loppuun tehtävä virityslohko olisi saattanut olla toteutuskelpoinen vaihtoehto, mutta käytännössä opinnäytetyön laajuus olisi tullut rajaavaksi tekijäksi eikä virityslohkosta olisi tullut yhtä monipuolista kuin kaupalliset vaihtoehdot. Lohkon jatkokehitys olisi ollut siten kallista. Yksi merkittävä hyöty SCL-kielellä ohjelmoidusta virityslohkosta olisi ollut sen mahdollinen käyttöönotto muilla logiikoilla. SCL on standardoitu ohjelmointikieli logiikoilla ja sillä ohjelmoidut ohjelmat voidaan ottaa käyttöön tietyillä logiikoilla suhteellisen helposti. Kaupalliset toteutukset ovat tarkoitettu vain tietyille logiikalle ja eri valmistajien logiikoille on erillisiä tuotteita.

Siemensin logiikoiden standardikirjastoissa on kaksi PID-säädintä. Pelkän säätimen sisältävä FB41, joka oli käytössä yrityksen ohjelmistossa ja FB58 jossa on integroituna viritystoiminto. FB58 olisi ollut kustannuksiltaan paras vaihtoehto koska standardikirjaston lohkot ovat ilmaisia. Sen ominaisuudet kuitenkin eroavat siinä määrin FB41:n ominaisuuksista, että vaihto olisi ollut hankala eikä kaikkia ennen käytössä olleita ominaisuuksia olisi välttämättä saatu käyttöön. FB58:lla tehtiin kuitenkin testejä ja näin saatiin alustavaa tietoa Siemensin automaattiviritysten toiminnasta. FB58 sisälsi askelvastemenetelmän tapaisen viritystoiminnon ja lisäksi releohjausmenetelmän tapaisen yhteen pulssiin perustuvan optimointitoiminnon.

Siemensillä on erikseen myytävä tuote SIMATIC PID Self-Tuner V5, joka voidaan liittää mm. FB41-säädinlohkoon. Sellainen tilattiin koekäyttöön ja ensimmäinen tehtävä oli selvittää miten liitokset viritys- ja säädinlohkojen välillä tehdään. Tuotteen mukana tulleissa esimerkkiohjelmissa ja ohjekirjan esimerkeissä oli käytetty erästä maksullista säädinlohkoa, mutta ohjekirjoja tutkimalla ja kokeilemalla Self-Tuner saatiin toimimaan FB41-säätimen kanssa. Tästä työvaiheesta on kerrottu tarkemmin luvussa 6.

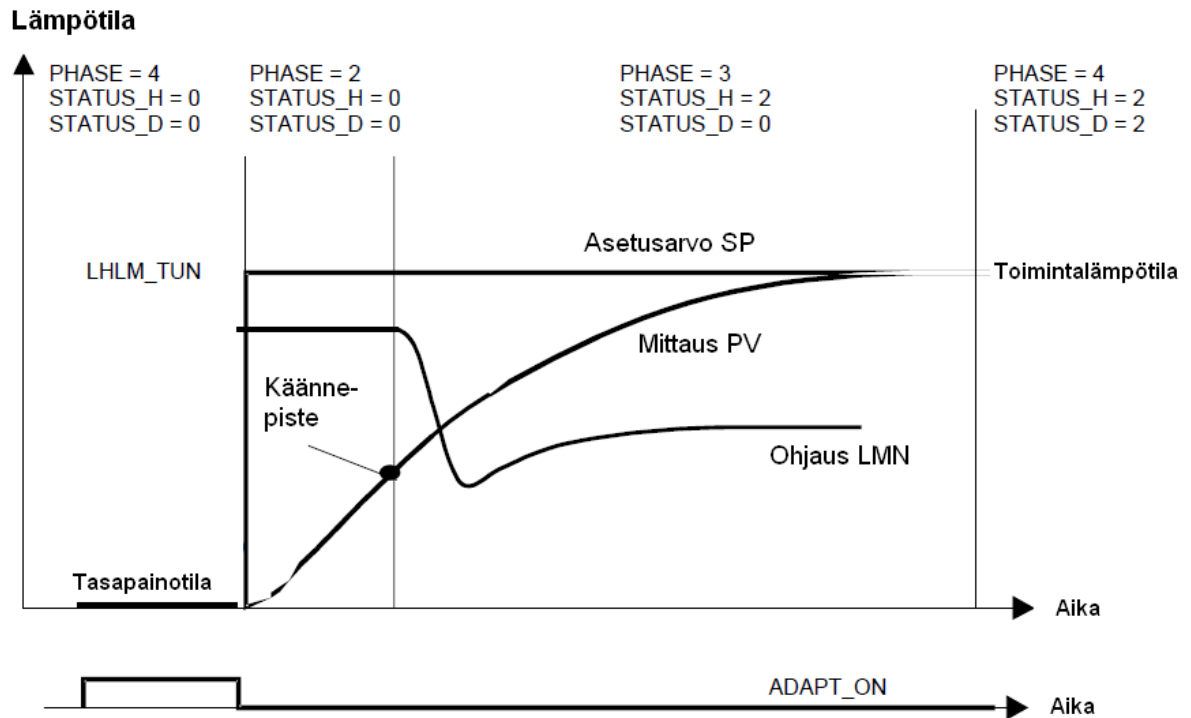
Toteutukseen valittiin valmis kaupallinen ohjelmisto SIMATIC PID Self-Tuner V5. Merkittäviä kriteerejä valintaa tehtäessä olivat virityslohkon hyvä toiminta testeissä, suhteellisen helppo käyttöönotto, pienet lisenssikustannukset ja hyvä integraatio vanhan ohjelmiston kanssa.

5 FB50-virityslohkon toimintaperiaate

Työhön valittu virityslohko käyttää säätimen viritykseen muokattua askelvastemenetelmää. Kuviossa 4 on esitetty virityksen vaiheet (PHASE). Viritys otetaan käyttöön asettamalla bitti ADAPT_ON tai ADAPT_1ST ja viritys alkaa käyttäjän tekemästä asetusarvon askelmaisesta muutoksesta. Askelmuutoksen voi tehdä kun säädin on automaatilla ja mittaus on tasapainotilassa tai lähestymässä tasapainotilaa. PHASE-muuttuja vaihtaa arvoaan tilasta 4 tilaan 2 ja virityslohko asettaa säätimen manuaalille. LHLM_TUN-muuttuja on käyttäjän asetettavissa ja se määrittää ohjausarvon suuruuden askelvastekoetta tehtäessä. PHASE-muuttujan arvo vaihtuu tilaan 3, kun virityslohko havaitsee käännepisteen mittauksessa.

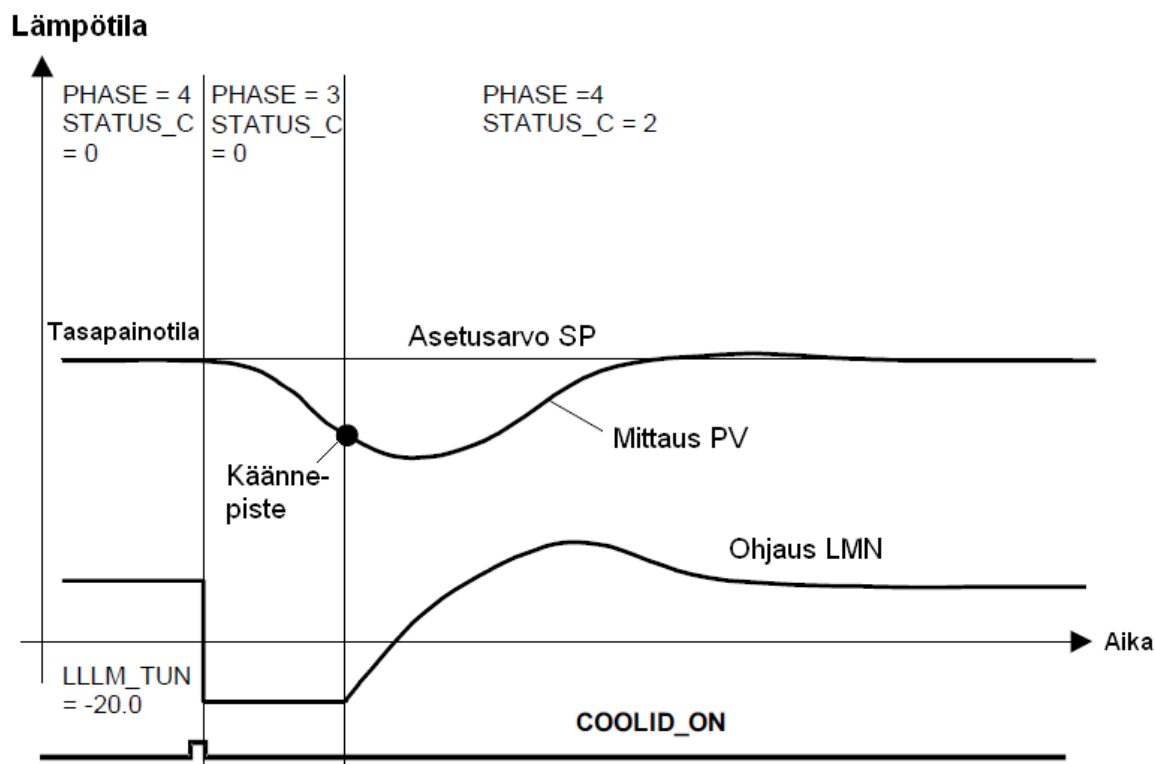
Virityslohko muodostaa prosessista viiveellisen ensimmäisen kertaluvun järjestelmän mallin ja laskee viritysparametrit sen avulla. Säätimen tila vaihdetaan automaatille ja virityslohko asettaa viritysparametrit säätimeen. Mittaus ohjataan asetusarvoon ja virityslohko tekee varsinaisen virityksen tänä aikana. Säädin tekee pieniä muutoksia asetusarvoon ja tarkkailee erosuuretta. Näillä tiedoilla virityslohko parantaa viritysparametreja edelleen. Viritysparametrit voivat muuttua tässä vaiheessa merkittävästi, varsinkin jos prosessin dynamiikka ei ole todellisuudessa ensimmäisen kertaluvun prosessi tai jos mittauksessa on ollut häiriöitä lämmityksen aikana.

Säädettävän suureen saavuttaessa asetusarvon PHASE-muuttujan arvoksi tulee jälleen 4 ja virityslohko asettaa lopulliset viritysparametrit säätimelle. STATUS_H ilmaisee lämmitysvaiheessa tapahtuneet mahdolliset ongelmat ja STATUS_D antaa diagnostiikkatietoa virityksestä. Virityksen käyttöä ja muuttujien arvojen merkityksiä on selitetty tarkemmin liitteen 1 käyttöohjeessa.



Kuvio 4: Automaattisen virityksen vaiheet
(SIMATIC PID Self-Tuner V5 käyttöohje, 3-8)

Virityksparametrit voidaan lisäksi optimoida jäähdytyksen suhteen, jos prosessissa on lämmityksen lisäksi jäähdyttävä toimilaite. Kuviossa 5 on esitetty toiminnon vaiheet. Edellisen kohdan LHLM_TUN-muuttujaa vastaa nyt muuttuja LLLM_TUN, joka määrittää ohjaussuureen arvon jäähdytyksen aikana. Säädin pysyy nyt koko ajan automaattilla ja virityslahko etsii käänne-pisteen. Tuloksena on kerroin, joka muuttaa virityksparametreja mittaussuureen pienentyessä. Jäähdytysoptimointi ei siis ole erityisen hyödyllistä, jos prosessin dynamiikka on samanlainen lämmityksessä ja jäähdytyksessä. Tätä toimintoa ei otettu käyttöön työn lopullisessa versiossa, koska sille ei ollut erityistä tarvetta niissä käyttökohteissa, joihin virityslahkoa suunniteltiin käytettäväksi.

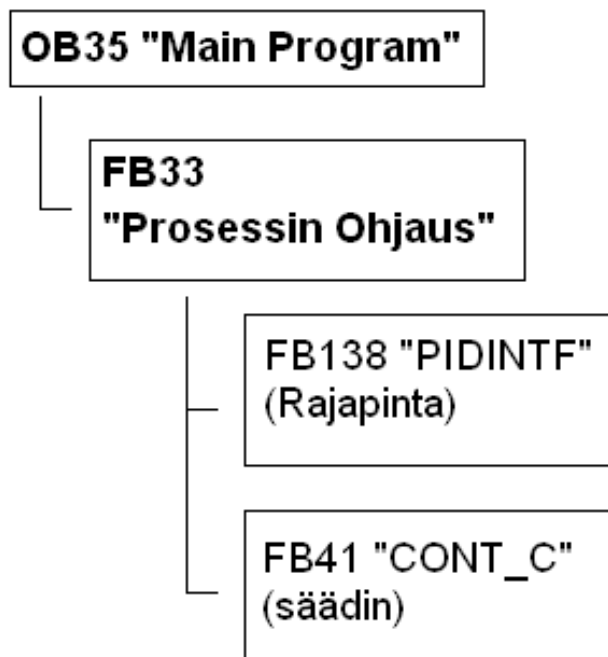


Kuvio 5: Jäähdytysoptimoinnin vaiheet
(SIMATIC PID Self-Tuner V5 käyttöohje, 3-10)

6 Virityslohkon käyttöönotto

Logiikkaohjelmiston perusrakenne säätöpiirien osalta ennen muutoksia oli kuvion 6 mukainen. OB35 on keskeytysjaksollinen lohko, joka suoritetaan jatkuvasti määritellyin väliajoin. Käytetyt säätimet ovat jatkuvatoimisia, joten niitä pitää kutsua tasaisin väliajoin. Kutsut voidaan täten sijoittaa OB35:een.

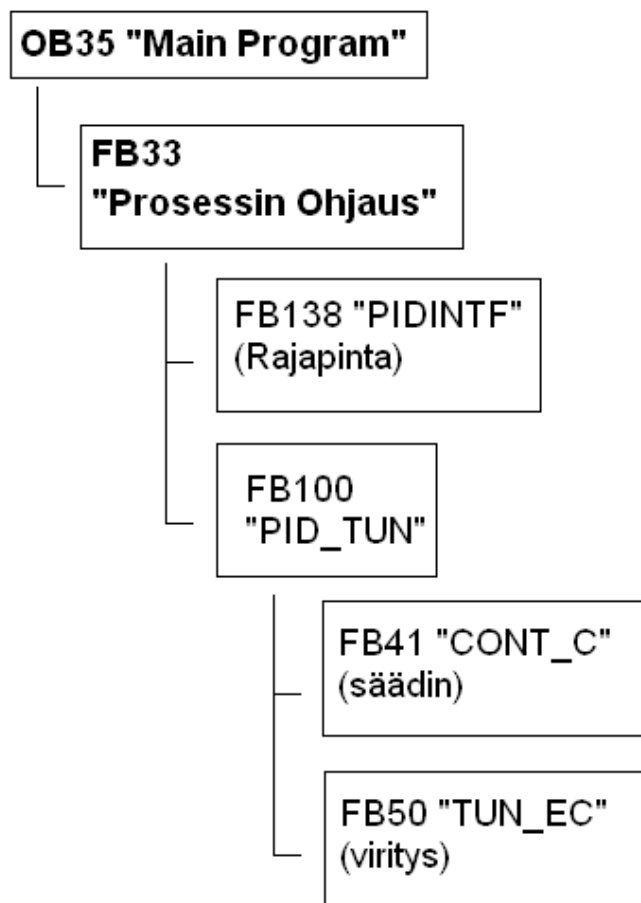
Ohjelmassa on funktiolohko FB33, joka sisältää tietyn prosessikonaisuuden ohjaukset. Se voi sisältää useita säätimiä ja muita lohkoja. Tärkeimmät osat ovat valvomon ja logiikan välistä kommunikointia hoitava rajapintalohko FB138 ja säädinlohko FB41.



Kuvio 6: Ohjelman rakenne yhden säätimen osalta ennen muutoksia

Funktiolohkot tarvitsevat instanssidatalohkon, johon ne tallentavat tulo-, lähtö- ja sisäiset muuttujansa. Datalohkojen määrää ja ohjelman muistin tarvetta voidaan vähentää määrittelemällä funktiolohkoja staattisiksi muuttujiksi toisten funktiolohkojen sisällä. Kuvion 6 ohjelmassa FB33:een on määritelty lohkot FB138 ja FB41 staattisiksi muuttujiksi, jolloin kaikki funktiolohkot käyttävät FB33:n datalohkoa.

Ohjelmaan tehtiin kuvion 7 mukaiset muutokset. FB41-säädinlohko korvattiin funktiolohkolla, joka sisältää säädinlohkon ja virityslahkon sekä näiden väliset kytkennät. Uusi säädinlohko on ulkoasultaan samanlainen FB41-lohkon kanssa sillä poikkeuksella, että se sisältää myös virityksen muuttujia. Lisäksi rajapintalohkoon tehtiin tarvittavat lisäykset viritystoiminnon osalta.



Kuvio 7: Ohjelman rakenne yhden säätimen osalta virityslohkon lisäyksen jälkeen

Yhdistelmälohkoon määriteltiin staattisiksi muuttujiksi säädinlohko FB41 ja virityslohko FB50. Yhdistelmälohko oli tarkoitus toteuttaa siten, että sen ulkoasu poikkeaisi mahdollisimman vähän aiemmin käytössä olleen FB41:n ulkoasusta. Tämä vähentää käyttöönottohenkilökunnan uudelleenkoulutuksen tarvetta. Yhdistelmälohkon tuloiksi ja lähdöiksi määriteltiin säädin- ja virityslohkon tuloja ja lähtöjä vastaavat muuttujat. Osa virityslohkon tarvitsemista muuttujista jätettiin kokonaan pois ja osa voitiin muodostaa säätimen tulosuureista yhdistelmälohkon sisällä. Muutama virityslohkon toiminto kuitenkin vaati tulo- ja lähtömuuttujien lisäämistä yhdistelmälohkoon.

Rajapintalohko FB138:ssa on yhtenä tulomuuttujana datasana CNW, joka tulee valvomosta. Datasanan biteillä voidaan muuttaa säätimen boolean-tyyppisten muuttujien tiloja. Valvomo-ohjelmisto asettaa CNW:n bitit yhden ohjelmakierron ajaksi ja CNW nollataan rajapintalohkossa. Rajapintalohkossa on lähtömuuttujina datasanat OPW ja OPWD, joihin kopioidaan tietyt CNW:llä ohjattujen muuttujien tilat. OPWD lähettää

valvomoon tietoja, joista nähdään että muutokset ovat tapahtuneet logiikassa, kuten oli tarkoitus. OPW:n kautta valvomoon lähetetään tapahtumätietoja, joiden avulla käyttäjä voi etsiä tiettyjä tapahtumia valvomon historiatiedoista.

CNW-sanaan lisättiin kaikki virityksen käyttöön liittyvät boolean-muuttujat. ADAPT_ON-muuttuja oli ainoa, jonka tulee pitää tilansa kauemmin kuin yhden ohjelmakierron ajan. CNW nollataan jokaisella ohjelmakierrolla ja silloin myös ADAPT_ON nollaantuu, joten yhdistelmälohkosta tehtiin takaisinkytkentä rajapintalohkoon. Takaisinkytkentä pitää ADAPT_ON-muuttujan tilassa 1, jos se on 1 myös virityslohkossa. Virityslohko nollaa kyseisen muuttujan automaattisesti virityksen alkaessa ja käyttäjä voi myös nollata sen käyttämällä säätimen käsiajolla.

Monimutkaisempien datatyyppeiden kuten aikojen, reaalityyppien ja kokonaislukujen siirto logiikan ja valvomo-ohjelmiston välillä tapahtuu datalohkojen kautta. Datalohkoissa kaikki tällaiset muuttujat ovat DINT-tyyppiä, joten muuttujia luettaessa datalohkoista ja kirjoittaessa datalohkoihin tehdään tyyppimuunnoksia. Reaalityyppien pyöristetään kokonaisluvuiksi ja kerrotaan sadalla, jotta tarkkuus pysyy riittävänä pyöristyksestä huolimatta. Ajat ovat logiikassa käytännössä DINT-tyyppisiä, mutta ne pitää kertoa kymmenellä, jotta millisekunnit muuntuvat oikein. Rajapintalohkoon lisättiin MIN_STEP- ja LHLM_TUN-muuttujat ja yhdistelmälohkoon tehtiin lähdöt virityslohkon määrittämille PID-parametreille. Virityslohko kirjoittaa parametrit suoraan säätimelle, mutta ne pitää kirjoittaa lisäksi valvomo-ohjelmalle tarkoitettuun datalohkoon, jotta uudet viritysparemetrit eivät ylikirjoittuisi sitä kautta seuraavalla ohjelmakierrolla.

7 Operointikäyttöliittymän muutokset

Virityslohkoa tullaan käyttämään valvomon InTouch-pohjaisesta käyttöliittymästä, joten virityslohkon tarvitsevat muuttujat tulee lisätä myös sinne. Varsinaisten muutosten ohjelmointi jätettiin opinnäytetyöaiheen ulkopuolelle, mutta alustava suunnitelma muutoksista päätettiin kuitenkin tehdä.

Operointikäyttöliittymää ei päästy testaamaan työn aikana, mutta kuvakaappausten perusteella saatiin riittävästi tietoa sen toiminnasta. PID-säätimen käyttöliittymässä on

Dynamics-painike, jolla avataan PID-parametrien muuttamiseen tarkoitettu uusi ikkuna. Ikkunassa on painikkeet kullekin parametrille ja lisäksi painike oletusparametrien palauttamiseen. Ikkunaan tulee lisätä painikkeet boolean-tiedoille: ADAPT1ST, ADAPT_ON ja STEADY. Muuttujien tilat viedään logiikalle CNW-sanalla. ADAPT1ST ja STEADY ovat päällä vain yhden ohjelmakierron ajan, jonka jälkeen logiikka nolaa niiden arvot. ADAPT_ON on IN/OUT-tyyppinen muuttuja, jonka logiikka nolaa kun viritys aloitetaan. Se asetetaan CNW-sanalla, mutta sen tilaa tarkkaillaan OPWD-sanalla.

Taulukossa 3 on esitetty CNW- ja OPWD-sanojen sisältö. Ensimmäisellä rivillä on sanan yksittäisiä bittejä vastaavat kokonaislukuarvot. Logiikka vertaa CNW-sanan sisältöä kokonaislukuihin, joten vaikka ohjaukset ovat bittitietoja, kahden muuttujan asettaminen kerralla on mahdotonta. OPWD lähetetään sellaisenaan valvomoon, joten siinä voi olla useita bittejä asetettuna kerralla. Taulukon toiselta ja kolmannelta riviltä selviää CNW:n ja OPWD:n toiminnot.

Taulukko 3: CNW:n ja OPWD:n toiminnot

Sanan bitit	128	64	32	16	8	4	2	1
CNW	STEADY	ADAPT_ON	ADAPT1ST	Reset WOIS_CV	Set WOIS_CV	Load defaults	Reset WOIS_CTRL	
OPWD	ADAPT_ON	D_SEL	I_SEL	P_SEL			WOIS_CV	WOIS_CTRL
								Set WOIS_CTRL

OPW-sanassa ADAPT_ON:ia vastaa sama bitti kuin OPWD-sanassa. OPW puretaan logiikassa ”Word to Bit”-lohkolla yksittäisiksi biteiksi, jotka tallennetaan tiettyihin muistipaikkoihin ja lähetetään valvomoon.

Käyttöliittymään lisätään reaalityypit MIN_STEP ja LHLM_TUN. Ne lähetetään PLC:lle DINT-tyyppisinä, eli reaalityypit tulevat kertot sadalla, jotta kokonaistyyppi pyöristämisestä huolimatta säilyy kahden desimaalin tarkkuus.

Logiikalta tulee tieto virityksen onnistumisesta ja säätimen tilasta. Ne ovat kokonaistyyppiä ja lukujen arvoja vastaavat selitykset löytyvät liitteestä 1. Valvomon käyttöliittymässä voidaan näyttää joko lukuarvot tai sanallinen selitys. Taulukossa 4 on esitetty valvomoon lisättävät muuttujat ja niiden sijainnit logiikassa.

Taulukko 4: Valvomo-ohjelmistoon lisättävien muuttujien I/O-luettelo

Muuttuja	Tyyppi	Kuvaus	Sijainti logiikassa
IN			
PHASE	DINT	Virityksen vaihe ja säätimen tila. Kts. Liite 1 Taulukko 1	DB41
STATUS_H	DINT	Virityksen lämmitysvaiheen diagnostiikkatieto. Kts. Liite 1 Taulukko 2	DB41
STATUS_D	DINT	Viritysparametrien suunnittelun diagnostiikkatieto. Kts. Liite 1 Taulukko 2	DB41
OUT			
ADAPT1ST	BOOL	Ensimmäisen virityksen käyttöönotto.	DB42 (CNW = 32)
STEADY	BOOL	Tasapainotila saavutettu tai paluu viritykseen.	DB42 (CNW = 128)
LHLM_TUN	DINT	Ohjauksen arvo ensimmäisen virityksen aikana.	DB42
MIN_STEP	DINT	Pienin asetusarvomuutos, jolla viritys voidaan käynnistää.	DB42
IN/OUT			
ADAPT_ON	BOOL	Virityksen käyttöönotto / viritystoiminto odottaa asetusarvon askelmuutosta.	DB42 (CNW = 64) ja DB41 (OPWD = 128)

8 Ohjelmiston toiminnan testaus

Automaattista viritystä testattiin työn aikana simuloituilla prosesseilla, joiden dynamiikka oli muunneltavissa prosessilohkon tulomuuttujilla. Simuloituissa prosesseissa ei ole luontaisia häiriöitä mittauksissa, joten häiriöiden vaikutusta virityksen toimintaan testattiin lisäämällä häiriösignaaleja prosessin säädettävään suureeseen.

Kohinan kaltaista häiriötä simuloitiin funktiolohkolla, joka tuotti satunnaislukuja. Lukujen vaihteluväli oli määriteltävissä funktiolohkon tulomuuttujalla. Virityslohko onnistui virityksessä varsin hyvin, vaikka jatkuvan kohinamaisen häiriön amplitudi oli 2 % mittausalueesta. Sitä suuremmat jatkuvat häiriöt ovat teollisuuden prosesseissa yleensä harvinaisia ja kertovat lähinnä siitä, että mittausten sijoitus tai toteutustavan valinta on ollut väärä.

Satunnaiset häiriöpiikit vaikuttavat virityksen onnistumiseen samalla tavalla kuin jatkuva kohina. Yleensä viritys onnistuu hyvin, mutta se voi myös epäonnistua häiriöiden ollessa liian suuria

Jos viritys epäonnistuu edellä mainituista syistä, diagnostiikkatiedot STATUS_H ja STATUS_D yleensä ilmoittavat virityksen epäonnistumisesta. Lisäksi käyttäjän on suhteellisen helppo havaita tällaiset ongelmat. Näin ollen edellä mainitut häiriöt eivät todennäköisesti aiheuta virityksen uusimisesta johtuvaa ajanhukkaa suurempia ongelmia.

Merkittävää haittaa säätöpiirin toiminnan kannalta virityksen aikana voi aiheuttaa säädettävään suureeseen summautuvat vakiosuuruiset jatkuvat häiriöt. Käytännön prosesseissa tällaisia voi esiintyä esimerkiksi silloin, jos muut lähellä olevat prosessit luovuttavat hukkalämpöä. Tällöin viritettävän prosessin dynamiikka voi muuttua hetkellisesti hyvinkin erilaiseksi kuin normaalitilanteessa ja viritys tehdään vääränlaisen prosessimallin pohjalta. Toinen vastaavanlainen tilanne on prosessin kuormituksen muuttuminen kesken virityksen. Tällaisia ongelmia virityslohko ei välttämättä havaitse ja viritys voi olla epäonnistunut, vaikka diagnostiikkatiedot viittaisivat onnistuneeseen viritykseen.

Käyttäjällä tulee siis olla tuntemusta säätöpiirien virityksestä, vaikka virittäminen on automatisoitu. Automaattiseen viritykseen pätee samat vaatimukset kuin käsin tehtävään viritykseen. Prosessista pitää eliminoida sen toimintaan normaalitilanteessa kuulumattomat muuttujat ja virittäminen tulee tehdä säädin kerrallaan, jos prosessiin kuuluu useita säätimiä. Virityksen kulkua tulee tarkkailla, jotta pystytään puuttumaan virheisiin ja havaitaan viritykseen mahdollisesti vaikuttavat häiriöt.

9 Virityslohkon käyttöohje

Viritystoiminnosta laadittiin käyttöohje käyttöönottohenkilökunnalle. Käyttöohje on liitteessä 1. Ohjeessa oletetaan lukijalla olevan perustietämys säätötekniikasta ja logiikkaohjelmistosta sellaisena kuin se oli ennen työn aikana tehtyjä muutoksia. Käyttöohje on laadittu työn aikana tehdyille logiikkaohjelmistolle eikä siinä ole vielä huomioitu valvomokäyttöä. Käyttöohjeen päivitys valvomokäytölle on kuitenkin suhteellisen yksinkertaista kun valvomoon tehtävät muutokset ovat valmistuneet.

10 Loppusanat

Työn tavoitteena oli tehostaa voimalaitosten käyttöönottoja. Tavoite luultavasti saavutettiin, vaikka työn aikana valmistunut ohjelmisto ei vielä ole käytössä. Viritystoiminto toimi testauksessa pääosin erinomaisesti ja todennäköisesti se tulee toimimaan myös todellisilla prosesseilla. Kaikkien prosessien viritys sillä ei välttämättä onnistu, mutta osittainenkin viritystoiminnon käyttö tehostaa käyttöönottoa.

Työn jatkojalostukseen tulee kuulumaan valvomokäyttöliittymän muokkaaminen niin, että viritystoiminnon käyttö onnistuu valvomosta käsin. Käytännön kokemusten karttuessa ohjelmaan voidaan tehdä vielä parannuksia, kuten muutoksia joidenkin muuttujien oletusarvoihin ja mahdollisesti ottaa käyttöön ominaisuuksia, jotka työtä tehtäessä koettiin tarpeettomiksi.

Lähdeluettelo

Seborg, D.; Edgar, T.; Mellichamp, D. (edited) 2003; 2004. Process dynamics and control 2nd edition. John Wiley & Sons, Inc.

SIMATIC PID Self-Tuner V5 käyttöohje, [online-pdf] [viitattu 6.4.2010].

Saatavissa: <http://support.automation.siemens.com/WW/view/en/1939235>

Liitteet

Liite 1: Virityksen käyttöohje

Viritystoiminto soveltuu käytettäväksi erityisen hyvin lämpötilasäädöissä, mutta myös pinnankorkeuden ja virtauksen säädöissä. Viritystä ei voi käyttää virtauksen säätöihin, joissa prosessin ominaisuudet koostuvat pelkästään venttiilin liikkeestä. Venttiilin jälkeen tulee olla aikavakiollinen prosessi.

Prosessin askelvasteen tulee olla asymptoottinen, eli sen tulee asettua tasapainotilaan käsiöjhajauksella. Viritystoiminto on suunniteltu käytettäväksi prosesseille, joiden viive on noin kymmenesosa prosessin aikavakiosta. Viritys toimii kuitenkin kohtuullisen hyvin viiveen ollessa korkeintaan kolmasosa prosessin aikavakiosta. Suuren viiveen tai vahvistuksen sisältävät prosessit saattavat aiheuttaa ylityksiä ja tällaisessa tilanteessa on suositeltavaa suorittaa askelvaste mahdollisimman matalassa toimintapisteessä ja nostaa asetusrvo tavoitteeseen vasta virityksen jälkeen.

Virityksen näytteenottotaajuutta määrittäessä tulee ottaa huomioon prosessin aikavakiot. Hitaiden prosessien virityksessä saattaa loppua muisti kesken, jos näytteenottotaajuus on liian suuri. Näytteenottotaajuus tulisi kuitenkin määrittää niin, että prosessin pienimmästä aikavakiosta otetaan vähintään 10-20 näytettä.

Viritystoiminto sietää melko suurta kohinamaista häiriötä signaaleissa, kunhan näytteenottotaajuus on riittävän suuri. Yksittäiset suuremmat häiriöpiikit saattavat kuitenkin keskeyttää virityksen liian aikaisin ja tällöin viritys tulee uusiksi. Viritys ei havaitse ulkopuolisia häiriöitä, kuten toisista prosesseista aiheutuvia satunnaisia lämpötilanvaihteluita. Tällaiset prosessin normaaliin toimintaan kuulumattomat muuttujat tulee eliminoida virityksen aikana.

FB ”PID_TUN”

Funktiolohko FB ”PID_TUN” eli FB100 sisältää säädinlohkon FB41 ja viritysolhkon FB50. Säätimen toiminnot ja liitännät ovat samat kuin FB41 -lohkossa, joten niiden toimintaa tai kytkentöjä ei ole käsitelty tässä ohjeessa.

Kytkenät

Viritystoiminnon boolean-tyyppisiä muuttujia ohjataan rajapintalohkon FB138 CNW-datasanan kautta. Rajapintalohkon boolean-tyyppiset IN/OUT-muuttujat kytketään STAT-muuttujilla vastaaviin muuttujiin FB100-lohkossa. Viritystoiminnon tarvitsemat reaalitylvut tulevat valvomosta DINT-tyyppisinä rajapintalohkoon, jossa tehdään tyyppimuunnos DINT → REAL. Rajapintalohkon reaalitylvulähdöt kytketään FB100:n vastaaviin tuloihin TEMP-tyypin muuttujilla.

Viritystoiminto kirjoittaa säätimelle uudet PID-parametrit virityksen aikana ja parametrit päivitetään samalla valvomon datalohkoon. Lohkossa FB100 on lähdöt tätä varten ja ne voidaan liittää suoraan datalohkon DB42 viritysparetreille tarkoitettuihin tietueisiin. Virityksen vaihetta ja diagnosointia ilmaisevat lähtömuuttujat kirjoitetaan suoraan datalohkoon DB41.

Virityksen toiminnot lyhyesti

PRED_ON

Käsiajolta automaatille siirryttäessä ohjaussuureen muutos voidaan tehdä yhdellä tai useammalla askeleella. Jos PRED_ON = TRUE, viritystoiminto määrittää askelten suuruuden ja määrän. Toiminto on käytettävissä vasta onnistuneen virityksen jälkeen. PRED_ON = TRUE nopeuttaa säädettävän suureen tasaantumista asetusarvoon erityisesti PI-säädöissä.

ADAPT1ST

Mahdollistaa virityksen aloittamisen positiivisesta asetusarvon askelmuutoksesta. Käytetään yleensä ensimmäisessä käyttöönotossa tai jos prosessin ominaisuudet ovat muuttuneet merkittävästi.

ADAPT_ON

Mahdollistaa virityksen aloittamisen positiivisesta asetusarvon askelmuutoksesta.

Voidaan käyttää esimerkiksi prosessin toimintapistettä muutettaessa tai jos prosessin ominaisuuksissa tapahtuu pieniä muutoksia. Jos virityslohkolla ei ole aiempaa tietoa prosessista, suoritetaan ADAPT1ST automaattisesti ADAPT_ON:n sijaan.

Viritystoiminnon käyttö

Säätimen tila ja virityksen vaiheet ilmaistaan muuttujalla PHASE. Taulukossa 1 on esitetty muuttujan tilojen merkitykset.

Taulukko 1: PHASE-muuttujan tilat

PHASE = 0	Säädin automaattilla. Käyttöönottovaihe, josta poistumiseen säädin tulee käyttää käsiajolla.
PHASE = 2	Viritys on käynnissä, säädin käsiajolla ja haetaan käännepistettä.
PHASE = 3	Viritys on käynnissä, säädin automaattilla ja viritysparametreja optimoidaan.
PHASE = 4	Säädin automaattilla normaalitilassa.
PHASE = 7	Säädin käsiajolla normaalitilassa.

Viritystoiminto otetaan käyttöön asettamalla bitti ADAPT1ST tai ADAPT_ON tilaan 1.

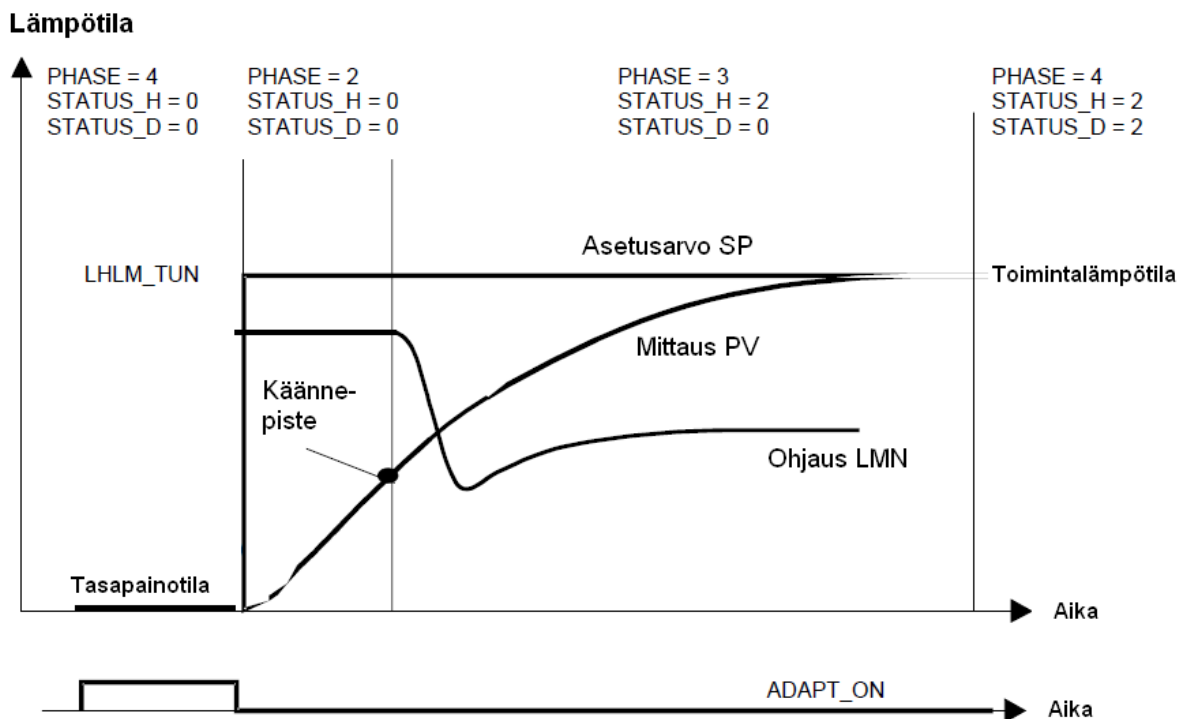
Viritystoiminto käynnistyy, kun käyttäjä tekee asetusarvomuutoksen positiiviseen suuntaan.

Muutoksen tulee olla suurempi kuin MIN_STEP-muuttujan arvo. MIN_STEP:in tulisi olla vähintään 10 % asetusarvon ja prosessisuureen toiminta-alueesta.

LHLM_TUN määrittää maksimiarvon ohjaussuureelle ensimmäisen virityksen (ADAPT1ST) aikana. Se kannattaa asettaa alussa pieneksi ja suurentaa tarvittaessa, jos viritys epäonnistuu. LHLM_TUN-muuttujan tarkoitus on estää tavoitteena olevan toimintalämpötilan ylitys virityksen aikana. LHLM_TUN ei ole käytössä, jos viritys aloitetaan ADAPT_ON-muuttujalla. Silloin viritystoiminto määrittää ohjauksen ylärajan automaattisesti. ADAPT_ON voi kuitenkin käynnistää ADAPT1ST-toiminnon, jos viritystoiminnolla ei ole riittävästi tietoa prosessista.

Viritys etenee kuvion 1 mukaisesti. Virityksen ensimmäisessä vaiheessa (PHASE=2) säädin on käsiajolla. Säätimelle kirjoitetaan alustavat virityksetparametrit ja säädin asetetaan automaatille, kun viritystoiminto löytää mittauskäyrän käännepisteen. Virityksen toiseen vaiheeseen (PHASE=3) siirrytään käännepisteen löydyttyä tai mittauksen noustua 70 %:iin asetusravosta.

Virityksen toisessa vaiheessa (PHASE=3) säädin on automaatilla ja mittaus ohjataan asetusravoon. Tänä aikana tapahtuu lopullisten virityksetparametrien määrittely. Viritys päättyy (PHASE=4) ja uudet parametrit asetetaan säätimelle, kun mittaus saavuttaa asetusravon tai aikaa on kulunut $30 * \tau_I$ vastaava aika. Käyttäjä voi myös lopettaa virityksen aiemmin STEADY-muuttujalla, jos mittaus on tasapainotilassa eikä viritystoiminto havaitse sitä esimerkiksi häiriöiden vuoksi. STEADY-muuttujalla voidaan myös tehdä paluu PHASE=4-tilasta PHASE=3-tilaan, jos viritys päättyy liian aikaisin.



Kuvio 1: Automaattisen virityksen vaiheet

STATUS_H on diagnostiikkatieto lämmityksen onnistumisesta ja STATUS_D virityksetparametrien suunnittelun onnistumisesta. Taulukossa 2 on esitetty näiden muuttujien mahdolliset arvot.

Taulukko 2: Diagnostiikkatietojen selitykset

Diagnostiikka-arvo	STATUS_H (lämmitys)	STATUS_D (suunnittelu)
0	Käyttöönottovaihe	Käyttöönottovaihe
1	Käännepiste liian matalalla. Viritys onnistuu yleensä kohtuullisen hyvin. Parantaaksesi tulosta nosta LHLM_TUN:ia tai tee pienempi asetusarvomutos.	Viritys onnistui hyvin. PI-säädin soveltuu prosessille.
2	Ideaalinen tilanne.	Viritys onnistui hyvin. Sekä PI-, että PID-säätimen käyttö mahdollista.
3	Käännepiste liian korkealla. Viritys onnistuu yleensä kohtuullisen hyvin. Parantaaksesi tulosta laske LHLM_TUN:ia tai tee suurempi asetusarvomutos.	PID-säädin suositeltu. Prosessi monikapasiteettinen tai aikavakiot/viiveet suuria.
11, 12, 13	Aikavakiota ei tunnistettu. Viritystä jatketaan, mutta tulos on todennäköisesti huono. Aloita viritys uudelleen ja varmista, ettei prosessisuureessa ole suuria häiriöitä virityksen aikana..	
4		Viritys ei onnistunut. PHASE=3:n aikaiset viritysparametrit ovat käytössä. Aloita viritys uudelleen ja varmista, ettei prosessisuureessa ole suuria häiriöitä virityksen aikana.