



Hihnakuuljetinjärjestelmän etä- ohjaus

Tommi Ikonen

OPINNÄYTETYÖ
Huhtikuu 2021

Sähkötekniikan tutkinto-ohjelma
Automaatiotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkötekniikan tutkinto-ohjelma
Automaatiotekniikka

IKONEN, TOMMI:
Hihnakuuljetinjärjestelmän etäohjaus

Opinnäytetyö 50 sivua, joista liitteitä 16 sivua
Huhtikuu 2021

Työn pohjana toimi hihnakuuljetinjärjestelmä, joka oli Jarmo Kallion vuonna 2008 valmistuneen tutkintotyön tulos. Tarkoituksena oli uuden oppimisympäristön kehittämisen säätötekniikan laboratorio harjoitusta varten. Tätä varten työn tavoitteena oli laajentaa järjestelmää etäohjaustoiminnolla sekä laatia harjoitustehtävät laboratorio työtä varten. Etäohjausohjelmaan kuului PID-säädin ja valvomönäyttö.

Työ alkoi asentamalla järjestelmään kaksi uutta anturia. Laseranturi toimi säätimen mittaus tietona, ilmoittaen hihnalla olevan kappaleen aseman suhteessa anturiin ja kulma-anturi antoi tarkkaa reaaliaikaista tietoa hinnan kaltevuuskulmasta.

Etäohjausjärjestelmän ohjelmaan laadittiin kulman ohjausta varten suorat ohjaukset, ja hinnan nopeutta varten sekä suora ohjaus että säätimellä toteutettu automaattinen ohjaus. Säätö toteutettiin PID-säätimenä, jonka avainparametrit muodostivat laboratorio harjoituksen aiheen.

Valvomönäytön kautta voidaan ohjata järjestelmää ja seurata prosessin arvoja reaaliajassa. Näytöt PID-säätimen avainparametrien muokkaukseen ja prosessiarvojen historian näyttämiseen täydentävät valvomönäytön tarjonnan laboratorio harjoitusta varten.

Aikaansaatu järjestelmä täyttää asetetut tavoitteet ja harjoitustehtäviä on ennakotehtävän lisäksi kaksi. Laadittu laboratorio harjoitus pyrkii asettamaan opiskelijat keskeiseen asemaan oppimisessa antamalla heille vapaat kädet koetella järjestelmän käyttäytymistä.

Projektin budjetin ja etenkin aikataulun puitteissa järjestelmään jäi monia kehityksen kohteita. Järjestelmän koneturvallisuutta ei parannettu ja hinnan kulman ohjaus on edelleen epälineaarista. Koneturvallisuuden parantaminen edellyttäisi järjestelmän ympäröimistä läpinäkyvillä seinillä. Valvomönäyttö voitaisiin siirtää erilliselle paneelille, muuttaen sen käyttöliittymän kosketusnäytöksi.

Asiasanat: automaatio, säätötekniikka, hihnakuuljetin

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Electrical Engineering
Automation Engineering

IKONEN, TOMMI:
Remote-Control of a Transport Belt System

Bachelor's thesis 50 pages, appendices 16 pages
April 2021

This project started with a transport belt system that Jarmo Kallio finished as his thesis project in 2008. The purpose was to develop a control theory learning environment for the automation laboratory. To do this, the system needed to be expanded with a remote-control feature, including a PID controller and a control room screen. The project also involved creating laboratory exercises to go with the system.

First, two new sensors were installed. A laser sensor would provide the controller with a process value, indicating an object's position relative to the sensor. Meanwhile, an angle sensor would indicate the belt's angle of inclination.

The remote-control system could control the transport belt's speed manually or through a controller, with only manual control for the angle of inclination. The key parameters of the PID controller could be adjusted by the user, as they were the subject of the laboratory exercise. All these features are accessed from the control room screen, which also shows process variables graphically in real time.

The laboratory exercise was designed with the intent of letting the students explore the limits of the system. The remote-control system is fully capable of supporting such experiments, even if there remain many avenues of further development.

Key words: automation, control theory, transport belt

SISÄLLYS

1	JOHDANTO	6
2	ANTURIT	7
	2.1 Ultraäänianturi.....	7
	2.2 Induktioanturi.....	8
	2.3 Kulma-anturi.....	9
	2.4 Laseranturi	11
3	SÄÄTÖ.....	13
	3.1 P-säätö.....	13
	3.2 I-säätö	15
	3.3 D-säätö	16
4	OHJELMOINTI.....	17
	4.1 Laitteiston konfiguraatio	17
	4.2 Logiikan ohjelmointi.....	18
	4.2.1 Mittasignaalien käsittely.....	19
	4.2.2 PID-säädin.....	20
	4.3 Valvomonäyttö	23
5	LABORATORIO HARJOITUKSET	28
	5.1 Ennakkotehtävät	28
	5.2 Laboratoriotyö	28
6	POHDINTA	30
	6.1 Työn onnistuminen.....	30
	6.2 Kehittämisehdotuksia.....	30
	6.2.1 Kulman ohjaus.....	31
	6.2.2 Koneturvallisuus	31
	6.2.3 Valvomonäytön ominaisuudet	32
	LÄHTEET.....	33
	LIITTEET	34
	Liite 1. Antureiden kytkentäkaavio	34
	Liite 2. PLC:n muuttujaluettelo	35
	Liite 3. Logiikan ohjelma	36
	Liite 4. Valvomonäytön HMI-muuttujaluettelo.....	42
	Liite 5. Valvomonäyttö.....	43
	Liite 6. Laboratorio harjoitus.....	45
	Liite 7. Opettajan ohjeet laboratorio harjoitusta varten.....	49

LYHENTEET JA TERMIT

FBD	Function Block Diagram
GSD	General Station Description
HMI	Human Machine Interface
K_C , Gain	Säätimen vahvistuskerroin
io	Input/Output
PID	Proportional, Integral, Derivative
STL	Statement List
τ_D , Td	Säätimen derivointiaika
τ_I , Ti	Säätimen integrointiaika

1 JOHDANTO

Tämän opinnäytetyön taustalla kohtaavat TAMK:n automaatiolaboratorion tarve uudelle oppimisympäristölle ja Jarmo Kallion vuonna 2008 valmiiksi saama tutkintotyö. Tutkintotyön tuloksena syntynyt hihnakuuljetinjärjestelmä toimi pohjana, josta tämä työ lähti laajentamaan sen toimintoja. Opinnäytetyön tarkoituksena oli luoda uusi säätötekniikan prosessiympäristö ja sitä hyödyntävä laboratorioharjoitus.

Tavoitteena oli kehittää hihnakuuljetinjärjestelmää lisäämällä siihen etäohjaustoiminto. Tähän etäohjaustoimintoon kuului säädin, joka tulisi toimimaan laadittavan harjoitustehtävän aiheena. Jarmon tutkintotyön jäljiltä laboratoriossa oli saatavilla tätä järjestelmää varten hankittuja instrumentteja, joilla säädin voitaisiin toteuttaa.

Etäohjauksen toteuttaminen tarkoitti myös valvomonäytön laatimista. Oppimisympäristön käytön kannalta valvomonäyttö on tärkeässä roolissa, sillä se on pääasiallinen työkalu laboratorioharjoituksen suorittamiseen.

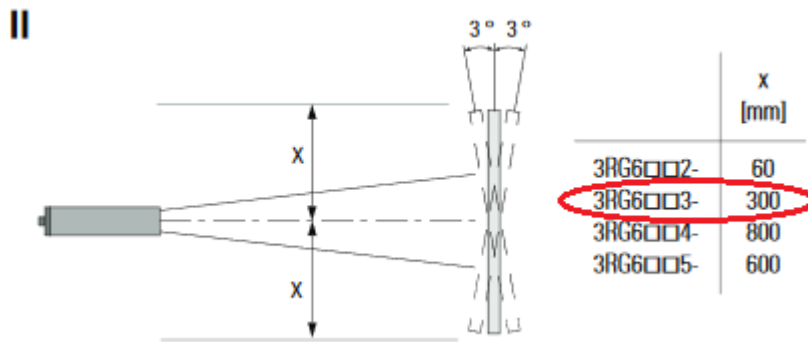
2 ANTURIT

Projektin alkaessa ainoat järjestelmän anturit olivat kaksi induktioanturia. Kappaleen paikan ja hihnan kulman mittaukset olivat selkeästi aiottu toteuttaa, mutta olivat jääneet vain aikeiksi. Käyttöön aiottu anturi oli kuitenkin jo silloin hankittu, joten ne olivat saatavilla tätä projektia varten.

2.1 Ultraäänianturi

Laboratoriossa saatavilla ollut Pepperl+Fuchs:in valmistama 3RG6-sarjan ultraäänilaite mittaa kappaleen etäisyyttä anturista ultraäänisignaalia käyttäen. Kuten monet etäisyysanturit, tämäkin anturi perustuu signaalin heijastumiseen mitattavasta kappaleesta. Säännöllisin väliajoin lähetettävä signaali heijastuu kappaleesta ja anturi havaitsee palaavan ääniaallon voimakkuuden perusteella signaalin kulkeman matkan. Anturin mittauspiiri muodostaa paluusignaaliin verrannollisen 4–20 mA virtaviestin.

Ultraäänianturi oli ensimmäinen prosessin mittaamiseen aiottu laite, sillä sen mittaussignaali on käytännössä tyystin harmiton verrattuna laseranturiin. Sen toiminta kuitenkin edellyttää riittävän esteettömän kulun anturin ja mitattavan kappaleen välillä, sillä signaalin heijastuminen muista kappaleista häiritsee mitausta. Mittaavan äänikeilan tarvitsema varoalue on suoraan verrannollinen mitattavaan etäisyyteen. Hihnakuuljettimen hihna ja kaiteet olisivat olleet häiriöksi jopa hyvin lyhyillä etäisyyksillä, mutta tämä prosessi olisi edellyttänyt jopa 30 senttimetrin varoalueen. Pepperl+Fuchs GmbH (2010) antaa eri mittausalueille soveltuvien laitteiden tarvitsemat varoalueet kuvassa 1.

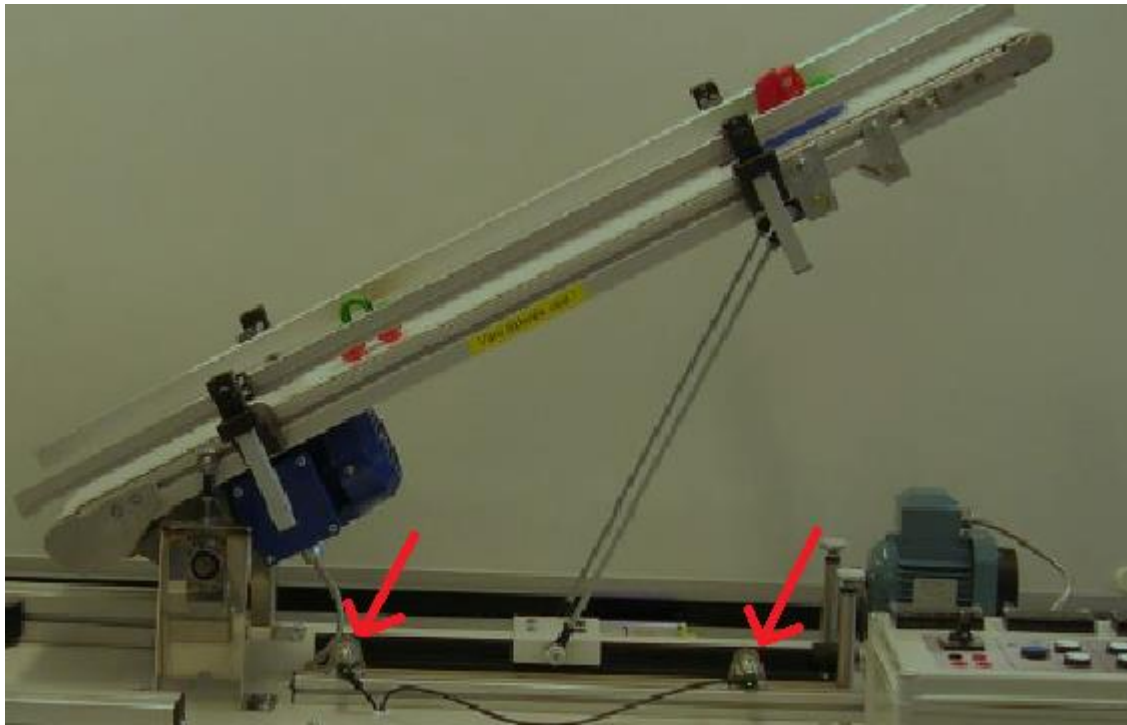


KUVA 1. Ultraäänianturin mittauskeilan edellyttämä varoalue (Pepperl+Fuchs 2010)

2.2 Induktioanturi

Järjestelmään kuului tämän työn alkaessa kaksi ifm electronic GmbH:n valmistamaa induktioanturia. Tällainen anturi koostuu kelan ja kondensaattorin muodostamasta oskillaattori- ja piiristä. Anturin tuntopintana toimii tämän piirin kela. Kun sähköä johtava kappale siirtyy kelan magneettikenttään, aiheutuu oskillaattori- ja piiriin pyörrevirtoja. Anturi havaitsee aiheutuvan kuormituksen, muuttaen lähtösignaaliaan. Tämä muutos voidaan tulkita logiikan tulokortilla binäärisenä, siis asetuksista riippuen joko avautuvana tai sulkeutuvana koskettimena. (Kuoppala 1986, 6.)

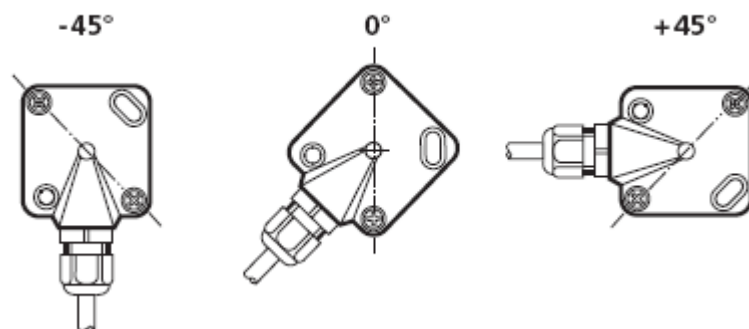
Nämä kaksi induktioanturia toimivat järjestelmässä hihnan kulman muutoksen ylä- ja alarajoina niin, että kun hihnan päätä pystysuunnassa liikuttavan varren pääty saavuttaa ääriasennon, siinä päässä oleva induktioanturi muuttaa tilaansa. Tällöin logiikan ohjelma pysäyttää liikettä ajavan taajuusmuuttajan. Kuvalla 2 Jarmo Kallio havainnollistaa hihnan kulmaa ohjaavaa järjestelmää antureineen.



Kuva 2. Induktioanturit hihnakuuljetinjärjestelmässä (Jarmo Kallio 2008, muokattu)

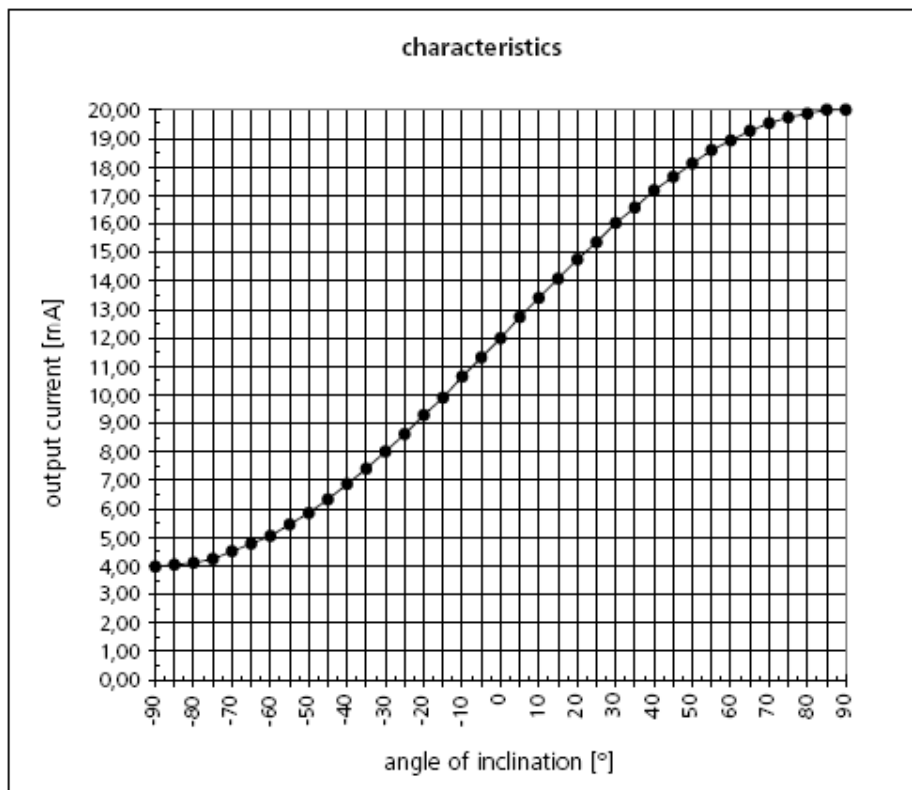
2.3 Kulma-anturi

Järjestelmään asennettiin ifm electronic GmbH:n kaltevuuskulma-anturi, joka toimii kapasitiivisen kiihtyvyyssanturin tapaan. Tämä anturi asennettiin suoraan kiinni hihnakuuljettimen runkoon, sillä ifm electronic GmbH:n (2015) mukaan anturin oma fyysinen asento vaikuttaa sen lähtösignaaliin (Kuva 3).



Kuva 3. Kulma-anturin asennon ja kulman välinen suhde. (ifm electronic GmbH 2004)

Anturin kaltevuuskulman muuttuessa anturin sisällä oleva mikromassa siirtyy suhteessa levykondensaattoriin, muuttaen sen kapasitanssia. Kapasitanssin muutos kulman suhteen tuottaa sinimuotoisen lähtösignaalin vastekäyrän ifm electronic GmbH:n (2004) mukaan (Kuva 4).



Kuva 4. Kulma-anturin signaali kulmakaltevuuden funktiona (ifm electronic GmbH 2004)

Kaltevuuskulman muutettavuus on hihnakuuljetinjärjestelmän ominainen piirre. Sellaisenaan se on keino tuottaa muutoksia kappaleen liikkeeseen hihnalla, mutta kaltevuuskulman mittaaminen mahdollistaa monia lisätoimintoja. Kulma-anturin mittaussignaali voi toimia esimerkiksi säätimen mittausravona. Anturin kytkentä on esitetty liitteessä 1 ja mittaussignaalin käsittely logiikan päässä on esitetty seuraavassa luvussa, laseranturin yhteydessä. Kuva 5 esittää kulma-anturia asennettuna hihnakuuljetinjärjestelmään.

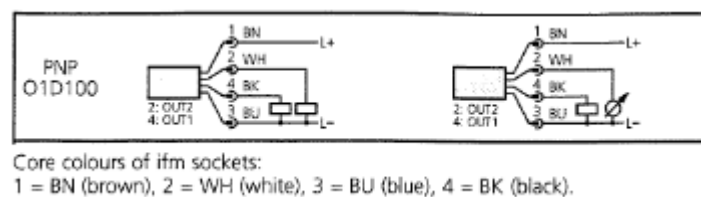


Kuva 5. Kulma-anturi

2.4 Laseranturi

Projektin kannalta järjestelmän tärkein mitattava suure oli kappaleen etäisyys hihnan päädyistä. Tätä tarkoitusta varten asennettiin ifm electronic GmbH:n valmistama 01D100 laseranturi. Käyttöohjeensa mukaan laitteen mittausalue on 0,2–10 metriä ja se ilmoittaa mittausarvonsa millimetrin tarkkuudella. Anturilla oli myös oma digitaalinen näyttönsä sen mittausarvon näyttämiseksi ja parametrien asettamiseksi.

Sekä tämä laseranturi että asennettu kulma-anturi olivat saman valmistajan laitteita, jotka molemmat käyttivät neljän johtimen kaapelia kytkentäänsä. Liite 1 näyttää, kuinka antureiden käyttöjännitteet tuotiin logiikan hajautetun osan kotelolta samalla, kun antureiden ulostulot (laseranturin OUT2 ja kulma-anturin pinni 4) kytkettiin analogiatulokortille saman kotelon sisällä. Ifm Electronics GmbH (2007) on esittänyt laseranturilta lähtevien johdinten värikoodausta ja tarkoitusta (Kuva 6).



Kuva 6. Laseranturin johdinten värikoodaus ja tarkoitus (ifm electronic GmbH 2007, muokattu)

Molempien antureiden signaalit tuotiin virtaviestinä eli 4–20 milliampeerin signaalina logiikan tulokortille. Tulokortin käyttöohjeesta selviää, että logiikka tulkitsee saapuvan virtasignaalin desimaalilukuna välillä 0–27648. Siemens (2007) esittää kuvassa 7 analogiatulokortin arvoja suhteessa saapuvaan virtaviestiin.

Measuring range 4 to 20 mA	Units		Range
	Decimal	Hexadecimal	
> 22,8142	32767	7FFF _H	Overflow
22,8142	32511	7EFF _H	Overshoot range
:	:	:	
20,0005	27649	6C01 _H	
20,0000	27648	6C00 _H	Nominal range
16,0000	20736	5100 _H	
:	:	:	
4,0000	0	0 _H	
3,9995	-1	FFFF _H	Underrange
.	:	:	
1,1852	-4864	ED00 _H	
< 1,1852	-32768	8000 _H	Underflow

Kuva 7. Analogiatulokortin arvo suhteessa saapuvaan virtasignaaliin (Siemens 2007)

3 SÄÄTÖ

Projektin tavoitteen mukainen säätötekniikan oppimisympäristö edellytti säätimen laatimista. Säädin on signaalinkäsittelylaite, joka prosessista mitattavan suureen ja käyttäjän antaman ohjeen avulla tuottaa prosessia ohjaavan signaalin. Tämä laite voi olla mekaaninen tai digitaalinen, mutta kummankinlaiset säätimet noudattavat samaa matemaattista periaatetta. Seborgin, Edgarin, Mellichampin ja Doyleen (2011, 130) esittämän yhtälön (1) mukaan projektissa käytettävän paralleelisen PID-säätimen ohjaussignaali $p(t)$ on

$$p(t) = \bar{p} + K_C \left[e(t) + \frac{1}{\tau_I} \int_0^t e(t) dt + \tau_D \frac{de(t)}{dt} \right], \quad (1)$$

jossa \bar{p} on vakio, K_C on säätimen vahvistuskerroin, $e(t)$ on erosuure, τ_I on integrointiaika ja τ_D on derivointiaika. Nämä muuttujat selitetään tarkemmin seuraavissa osioissa.

3.1 P-säätö

P-säätö tai suhteellinen säätö saa nimensä englannin kielen termistä proportional control. Se kuvaa hyvin yksinkertaista periaatetta, jossa ohjaussignaali muodostuu annetun asetusarvon ja mitatun prosessin arvon erotuksena. Yhtälö (2) kuvaa p-säätimen ohjaussignaalia $p(t)$

$$p(t) = \bar{p} + K_C e(t), \quad (2)$$

jossa erosuure $e(t)$ on

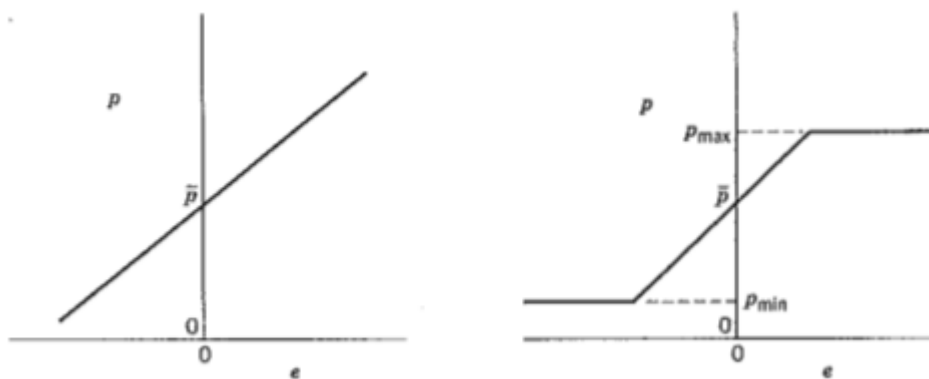
$$e(t) = y_{sp}(t) - y_m(t). \quad (3)$$

$Y_{sp}(t)$ on säätimen asetusarvo, Y_m on prosessista saatu mitta-arvo ja K_C on säätimen vahvistuskerroin. Seborg ym. (2011, 126) selittävät, että \bar{p} on vakio,

jolla voidaan määrittää P-säätimen ohjaussignaalia etenkin tilanteessa, jossa erosuure $e(t)$ saa arvon nolla.

P-säätimen avainpiirteet juontuvat vahvistuskertoimesta K_C . Nämä piirteet ovat Seborgin ym. (2011, 126) mukaan ensiksikin säätimen herkkyyden muuttaminen vahvistuskertoimen kautta, ja toiseksi negatiivisen vahvistuskertoimen aikaansaama kääntäen verrannollinen ohjaussignaali. Isompi vahvistuskertoimen tarkoittaa, että säädin reagoi voimakkaammin erosuureeseen. Kääntäen verrannollinen ohjaussignaali merkitsee, että erosuureen kasvaessa ohjaussignaali pienenee ja päinvastoin.

Suhteellista säätöä harkittaessa on myös hyvä huomata, että vaikka teoriassa tällaisen säätimen ohjaussignaali jatkaa kasvamistaan erosuureen kasvaessa, todellisuudessa ohjaussignaalia rajoittaa ohjattava laite. Jos esimerkiksi venttiilillä ohjataan virtausta, suuret ohjaussignaalin arvot lopulta tuovat venttiilin täysin aukinaiseen tilaan. Vastaavasti täysin kiinni oleva venttiili ei voi sulkeutua enempää. Seborg ym. (2011, 127) ovat esittäneet tätä ilmiötä (Kuva 8).



Kuva 8. P-säädön ideaalitoiminta verrattuna käytännön todellisuuteen (Seborg ym. 2011, 127)

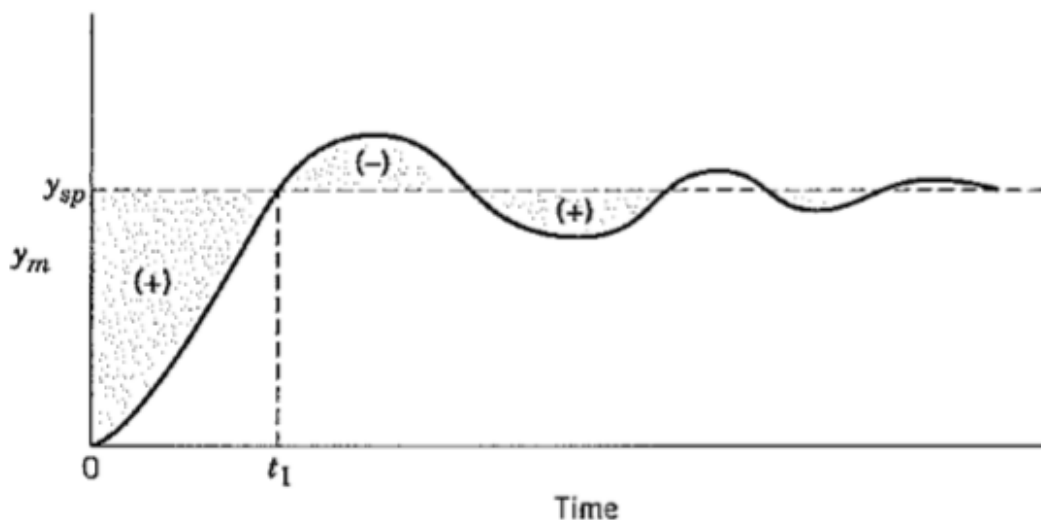
Koska P-säätimen ohjaussignaali on vakio erosuureen ollessa vakio, prosessiarvo voi vakiintua asetusarvosta poikkeavaan arvoon. Tämän poikkeaman (offset) läsnäolo ei ole riippuvainen vahvistuskertoimen arvosta. P-säätö on kuitenkin yksinkertaisuutensa takia suosittu valinta järjestelmille, joissa jonkinasteinen poikkeama on hyväksyttävää.

3.2 I-säätö

I-säätö pohjautuu integraalilaskentaan. Käytännössä tämä tarkoittaa, että I-säätimen ohjaussignaali jatkaa kasvamista niin kauan, kuin erosuureen arvo poikkeaa nolasta. Seborg ym. (2011, 127) toteavatkin, että I-säädin on laajassa käytössä johtuen sen kyvystä poistaa poikkeama. Integraalisäädin sellaisenaan kuitenkin reagoi erosuureeseen hitaasti P-säätimeen verrattuna. Siksi nämä säätömenetelmät usein yhdistetäänkin PI-säätimeksi.

Integraalisäätimen integraaliaika τ_I kuvastaa aikaa, jonka kuluttua integraalisäädön osa PI-säädintä on tuottanut yhtä ison vaikutuksen koko ohjaussignaaliin kuin p-säädön osa (Seborg ym. 2011, 128). Suurempi τ_I :n arvo siis aiheuttaa heikomman integraalisäädön komponentin.

Integraalisäädön eräs mahdollinen haittapuoli on, että se voi jättää prosessiarvon värähtelemään aset SARVON ympärille aset SARVON muutoksen jälkeen. Erosuureen saavuttaessa nol lan integraalisäädön osa on edelleen ohjaamassa samaan suuntaan, johtaen erimerkkiseen erosuureen arvoon. Vasta tässä kohdassa i-säätö alkaa korjata toiseen suuntaan. Tämä ilmiö voi toistua useammankin kerran, ilmentyen värähtelynä. Seborgin ym. (2011, 128) mukaan tarkka integraalisäädön määrittely kuten myös derivointitoiminto voivat ehkäistä tämän integraalisäädön haitan. Kuva 9 esittää tällaista järjestelmän epävakautta.



Kuva 9. Integraalisäädön värähtelyilmiö (Seborg ym. 2011, 128)

I-säädön kanssa voidaan myös törmätä ilmiöön, jota kutsutaan integraalikyllästykseksi (integral windup). Erosuureen säilyessä pidempään integraalisäädön osa voi kasvaa huomattavan suureksi, saattaen ohjaussignaalin maksimiarvoonsa (Seborg ym. 2011, 128). Integraalikyllästys tarkemmin ottaen viittaa ohjaussignaalin kasvuun vielä sen jälkeen, kun ohjaimen parhaat yritykset korjata virhe eivät riitä.

3.3 D-säätö

D-säädössä on kyse derivaatilaskennasta, eli tarkemmin sanottuna erosuureen muutosnopeudesta. Tämä säätötoiminto ei siis reagoi ollenkaan erosuureeseen, joka pysyy vakiona. Derivointisäädön rooli on kuitenkin reagoida äkillisiin muutoksiin, mihin p- ja i-säädöt eivät kykene, tukien näitä säätökeinoja. Derivointiaika τ_D toimii D-säädön omana vahvistuskertoimena.

Monissa prosesseissa derivointiosan lisääminen säätöön parantaa säätimen vasteaikaa eli aikaa, joka prosessiarvolta menee asetusarvon saavuttamiseksi, kun asetusarvossa tapahtuu askelmuotoinen muutos. Prosesseissa, joissa esiintyy paljon satunnaisia prosessiarvon muutoksia eli korkeataajuisia kohinaa D-säätö aiheuttaa kuitenkin epävakautta. Seborg ym. (2011, 129) huomauttavatkin, että D-säätöä käytetään vain harvoin virtauksen säätämiseen, sillä virtausprosesseilla on taipumusta sisältää kohinaa.

4 OHJELMOINTI

Projektin lähtökohtana toimiva paikallisohjauksella varustettu hihnakuuljetinjärjestelmä oli koottu Siemensin logiikkalaitteita käyttäen ja ohjelmoitu Siemensin ohjelmistolla. Tästä syystä Siemensin ohjelmisto oli ilmeinen valinta tätä projektia varten. Käytössä oli siis TIA Portal-sovelluksen versio 16. Tavoitteena oli toteuttaa järjestelmän etäohjaus, johon kuului hihnan nopeuden ja kulman suora ohjaus sekä automaattisen ohjauksen vaihtoehto, johon kuului PID-säädin. Näitä toimintoja varten piti laatia myös valvomonäyttö.

4.1 Laitteiston konfiguraatio

Ohjelmaprojektin aluksi oli välttämätöntä määrittellä käytössä oleva laitteisto ja konfiguroida laitteiden väliset yhteydet. Siemensin laitteet saatiin valmiina ohjelmiston omasta katalogista, mutta muiden valmistajien laitteita varten oli hankittava ja asennettava GSD-tiedostot TIA Portalin käytettäväksi. GSD-tiedosto (General Station Description) on laitteen valmistajan tarjoama kuvaus io-laitteesta, sisältäen tietoa konfigurointiasetuksista, parametroidista, moduuleista, diagnostiikasta, hälytystoiminnoista, laitteen myyjästä sekä tuotetunnisteesta (Profinet University n.d.).

Alkuperäiseen laitteistoon kuuluneiden analogialähtökorttien tuki oli lakkautettu muutama vuosi alkuperäisen projektin valmistumisen jälkeen. Toisin sanoen näitä kortteja ei ollut TIA Portalin katalogissa. Analogialähtökortteja ei kuitenkaan ole käytetty edeltäneessä projektissa eikä tällä projektilla ollut myöskään niille tarvetta, joten ne voitiin purkaa laitteistosta.

Taajuusmuuttajien konfigurointi oli edelleen voimassa edeltäneen projektin jäljiltä, joten yhteys niihin muodostui helposti, kunhan TIA Portaliin oli asennettu tarvittavat tiedostot. Loput laitteet olivat Siemensin, joten analogialähtökorttien poiston jälkeen yhteydet järjestyivät kivuttomasti.

4.2 Logiikan ohjelmointi

Edeltävä projekti oli ohjelmoitu käyttäen tekstimuotoisia komentoja eli STL-formaatissa (Statement List). Koska etäohjauksen ohjelmaan kuului samoja toimintoja kuin paikallisohjaukseen, oli järkevää käyttää samaa formaattia. Formaattista riippumatta Siemensin ohjelmoinnissa ohjelman muuttujat nimetään (englanniksi näistä nimistä käytetään termiä tag). Kaikki PLC:n (PLC = Programmable Logic Computer) ohjelmassa käytetyt muuttujat ovat nähtävissä liitteessä 2.

Etäohjauksen kutsu oli ohjelmoitu jo edeltäneessä projektissa. Tämä kutsu oli asiallisesti ohjelmoitu, mutta ymmärrettävästä syystä ei ottanut huomioon etäohjelman tarpeita ohjelmakutsunsa suhteen. Etäohjauksen ohjelma FC20 päättyi sisältämään useita tilapäisiä muuttujia, jotka piti määritellä ohjelmakutsussa. Tämä tarkoitti myös, että kutsun ehdollisuus piti ohjelmoida uudelleen. Kuva 10 esittää pääohjelman osaa, jossa etäohjausta kutsutaan.

Network 3: Etäohjauksen kutsu

```

0001      A      %M10.1
0002      A      "setRemoteControl"
0003      JCN    M003
0004      CALL   "FC20"
0005      AngleInt    := "intKulma"
0006      AngleScaled := "skaalattuKulma"
0007      AngleFinal  := "korjattuKulma"
0008      PositionInt := "intPaikka"

```

Kuva 10. Etäohjauksen kutsu pääohjelmassa OB1

Hihnan kulman ohjaus toteutettiin identtisesti paikallisohjaukseen nähden sillä erotuksella, että käskybitit tuotiin valvomonäytöltä logiikan muistialueelle sen sijaan, että signaalit tuotaisiin digitaalisääntulokorteilta. Hihnan nopeuden ohjaus päätettiin toteuttaa eri tavalla. Paikallisohjaus ohjasi hihnan nopeutta kahdella painikkeella, toinen nopeuden nostamista ja toinen nopeuden laskemista varten. Etäohjaus korvasi nämä yhdellä nopeusohjeen syöttökentällä, jolloin taajuusmuuttajaa voitiin ohjata tarkasti. Hihnan kulman ja nopeuden ohjauksen ohjelmointi on nähtävissä liitteessä 3.

4.2.1 Mittasignaalien käsittely

Kappaleen paikan ja hinnan kaltevuuskulman mittasignaalit tulivat logiikalle analogisina viesteinä. Sisääntulokortilta saapuva tieto oli desimaaliluvun muodossa, mutta kortin luontainen mitta-alue (0–27648) olisi ollut epäkäytännöllinen.

Kappaleen paikan mittaus

Kappaleen paikka haluttiin näyttää valvomonäytöllä, mutta paikan mittasignaali oli myös olennainen osa PID-säätimen toimintaa. Sekä näytön suunnitteluohjelmisto WinCC että PID-säätimen parametrit kykenevät signaalinkäsittelyyn, mutta tämä olisi tarkoittanut skaalauksen määrittelyä kahteen kertaan. Skaalaus tehtiin siis logiikan ohjelmassa.

Skaalausta varten ohjelmassa kutsuttiin lohkoa SCALE. Tämä lohko muuttaa kokonaisluvun reaaliluvuksi määriteltyjen ylä- ja alarajojen mukaan. Kuva 11 esittää paikan mittauksen skaalausta FC20-ohjelmassa.

Network 1: Kappaleen paikan mittauksen skaalaus

Skaalaus tapahtuu näytön ohjelmassa

```

0001      L      "laserMittaus":P      //Laseranturin viesti
0002      T      #PositionInt
0003      CALL   SCALE                  //Muutos millimetreiksi
0004      IN      :=#PositionInt
0005      HI_LIM  :=10000.0
0006      LO_LIM  :=-30.0
0007      BIPOLAR :="vakioNolla"
0008      RET_VAL :="paikkaTarpeeton"
0009      OUT     :="mittausPaikkaSkaalattu"

```

Kuva 11. Paikan mittauksen skaalaus

Skaalauksen alaraja muuttui käytännön testauksen seurauksena. Laseranturin nimellinen mitta-alue alkaa arvosta 200 mm ja todellinen mitta-alue alkaa arvosta 150 mm. Sen parametrit kuitenkin määrittelevät nollasignaalin tarkoittavan arvoa nolla. Anturin oman näytön ja ohjelman skaalatun tuloksen vertailu johti yllänähtävään alarajan arvoon.

Kulma-anturin mittaus

Hihnan kaltevuuskulmaa päätettiin näyttää valvomonäytöllä. Tavoitteena oli näyttää kaltevuuskulma asteina niin, että kun hihna on vaakatasossa, kulma on 0° . WinCC tarjoaa omat signaalinkäsittelytoimintonsa, mutta sillä oli vaikeuksia käsitellä negatiivisia lukuja. Tämän takia skaalaus suoritettiin logiikan ohjelmassa.

Signaali skaalattiin ensin asteiksi anturin mittausalueen mukaan. Tämän jälkeen alueen nolla-arvoa siirrettiin, sillä anturin aseman mukainen kulma-arvo hihnan ollessa vaakatasossa oli 90° . Kuva 12 esittää kaltevuuskulman mittasignaalin käsittelyä PLC:n ohjelmassa.

Network 2: Hihnan kulman mittauksen skaalaus

Skaalaukseen kuuluu negatiivisten lukejn mahdollisuus, joten skaalaus logiikassa.

```

0001      L      "kulmaMittaus":P      //Kulma-anturin viesti
0002      T      #AngleInt
0003      CALL   SCALE                //Muutos asteiksi
0004      IN     :=#AngleInt
0005      HI_LIM :=0.0
0006      LO_LIM :=-180.0
0007      BIPOLAR :="vakioNolla"
0008      RET_VAL :="kulmaTarpeeton"
0009      OUT    :=#AngleScaled
0010      L      90.0
0011      -R
0012      T      "hihnanKulma"

```

Kuva 12. Kaltevuuskulman mittauksen skaalaus

4.2.2 PID-säädin

Koska PID-säätimen toiminta edellyttää toistuvaa säännöllistä suorittamista, se asetettiin erityiseen paikkaan logiikan ohjelmassa. Logiikan ohjelmalohko OB35 on piiri, jonka logiikka suorittaa säännöllisesti (cyclic interrupt). Piirin ominaisuuksien tarkastelu osoitti, että sen suoritussykli on 100 millisekuntia.

Säätimen ohjelmaa varten tehtiin poikkeus ohjelmointiformaatissa. Funktiolohko-ohjelmointi (FBD) antoi graafisen esityksen ohjelmasta, mikä oli luonteva ratkaisu PID-säädön monimutkaista ohjelmalohkoa varten.

Ohjelmalohko toteutettiin ohjelmakatalogin loholla CONT_C. Tämä lohko ei sisältänyt parametrejä asetusarvon rajoittamiseksi, joten tämä toiminto piti ohjelmoida erikseen. Käytännössä tämä toteutettiin vertailemalla asetusarvoa, ja käyttäen vertailujen tuloksia ehtona säädön ohittamiselle. Kuva 13 esittää asetusarvon vertailua ohjelmassa.

Network 1: Asetusarvon rajoitus

```

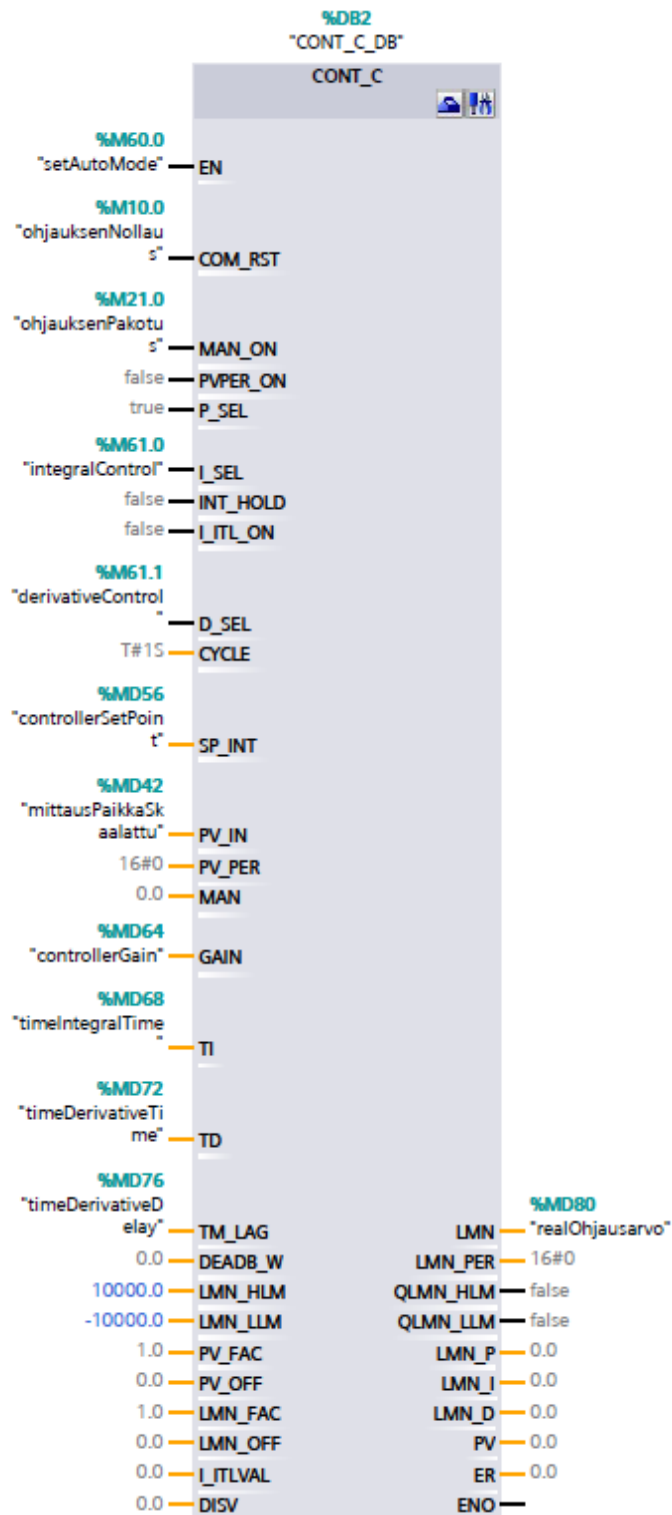
0001      L      "controllerSetPoint"
0002      L      1050.0
0003      >R
0004      JC      M001
0005      L      "controllerSetPoint"
0006      L      200
0007      <R
0008      JC      M001

```

Kuva 13. Asetusarvon rajoitus

Säätimen matemaattisten operaatioiden parametrejä liitettiin ohjelmamuuttujiin, jotta niitä voitaisiin muuttaa valvomonäytöltä jopa kesken ohjelman suorituksen. Myös COM_RST (säätimen uudelleenkäynnistys) kytkettiin muuttujaan, jotta säätimen ohjaus voitaisiin nollata käyttäjän toimesta. Säätimessä ei ollut sisäänrakennettua integraalikyllästyksen vastaista toimintaa, joten COM_RST:n käyttö valjastettiin tähän tarkoitukseen.

Säätimen ohjausarvo luettiin ulostulosta LMN. Tällekin asetettiin rajat (LMN_HLM, LMN_LLM), jotta ohjausarvo pysyisi taajuusmuuttajan nopeusohjeen lukualueella. Kuva 14 esittää PID_CP lohkoa sisään- ja ulostuloineen.



Kuva 14. CONT_C-lohko ohjelmassa

Lohkoon OB35 sisällytettiin myös STL-komento, jolla muuttuja ohjauksenNollaus palautettiin nolnaan. Tällä tavoin vältettiin säätimen jumituminen uudelleenkäynnistystilaan.

Käytännön testaus osoitti, että ohjausarvon siirto reaalitylukuna taajuusmuuttajan nopeusohjeeksi aiheutti tarpeettoman korkeita nopeusohjeen arvoja. STL-komentojen luettelosta löytyi kuitenkin yksinkertainen ratkaisu. RND-komento pyöristää viimeisimmän muistiin ladatun luvun reaalityluvusta lähimpään kokonaislukuun. Kuva 15 esittää säätimen ohjausarvon pyöristystä ja siirtoa taajuusmuuttajalle.

```
0011 L "realOhjausarvo" //Nopeuden ohjaus säätimeltä
0012 RND
0013 T "hihnaMoottoriNopeus":P
```

Kuva 15. Ohjausarvon pyöristys ja siirto taajuusmuuttajalle

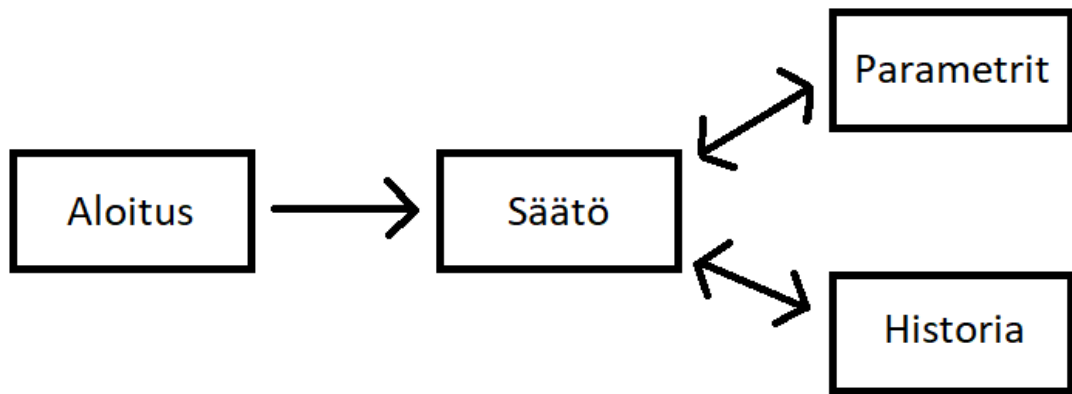
4.3 Valvomonäyttö

PC:ltä käytettävän valvomonäytön toiminta edellytti ensimmäisenä HMI-yhteyden (Human Machine Interface) luomista. Tätä varten PC:n IP-asetukset ja PG/PC-asetukset piti saada kuntoon. Käytännössä tämä tarkoitti Windowsin verkkoasetuksien käyttöä niin, että IPv4-osoite oli määritelty yhteneväksi TIA Portaliin annettujen tietojen kanssa, ja että PG/PC-yhteys käytti tietokoneen verkkokorttia.

Valvomonäytön tavoite projektissa oli antaa käyttäjälle graafinen esitys prosessin muuttujista sekä prosessin ohjaukseen tarvittavat työkalut. Näytön suunnitteluohjelma WinCC:n tarjonnasta ohjauksen työkaluiksi valittiin joukko painikkeita, kytkimiä ja io-kenttiä. Näiden työkalujen lisäksi tarvittiin kuvaajia prosessimuuttujien arvojen näyttämiseksi.

Näytön toiminnot, jotka vaikuttivat logiikan muistialueelle tai logiikan io-korttien arvoihin, tarvitsivat omat HMI-muuttujansa. HMI-muuttuja yhdistettiin vastaavaan PLC-muuttujaansa valitsemalla luettelosta. Kaikki valvomonäytön HMI-muuttujat ja niitä vastaavat PLC-muuttujat ovat nähtävissä liitteessä 4.

Kaikkien toimintojen asettaminen yhdelle näytölle ei olisi ollut mielekäästä monestakaan syystä. Liian täysi valvomonäyttö olisi jättänyt kuvaajat hyvin pieniksi, ja oikeiden io-kenttien tai painikkeiden löytäminen olisi ollut kuormittavaa. Kuvio 1 esittää valvomonäytön eri näyttöjä ja niiden välistä navigointia.

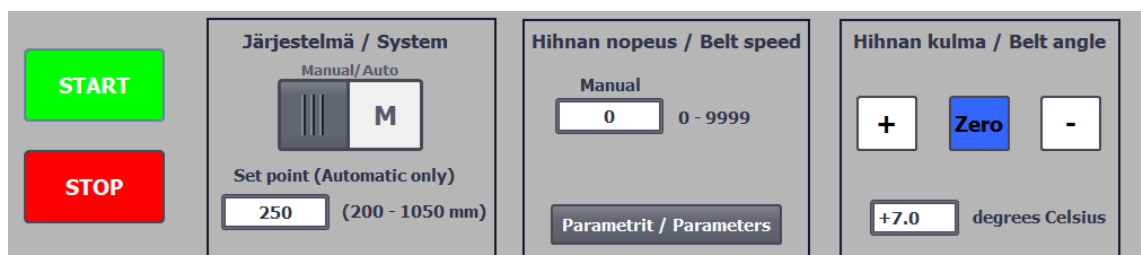


Kuvio 1. Valvomonäytön näyttörakenne

Aloitussäätö näytön tarkoitus on tarjota ikään kuin orientoiva sisääntulo. Siinä käyttäjä toivotetaan tervetulleeksi ja esitetään varoitus laseranturin muodostamasta vaarasta. Näytöllä oleva painike avaa Säätö-näytön, eikä "Aloitus"-näyttöön enää pääse palaamaan käynnistämättä näytön ohjelmaa uudelleen.

"Säätö"-näyttö

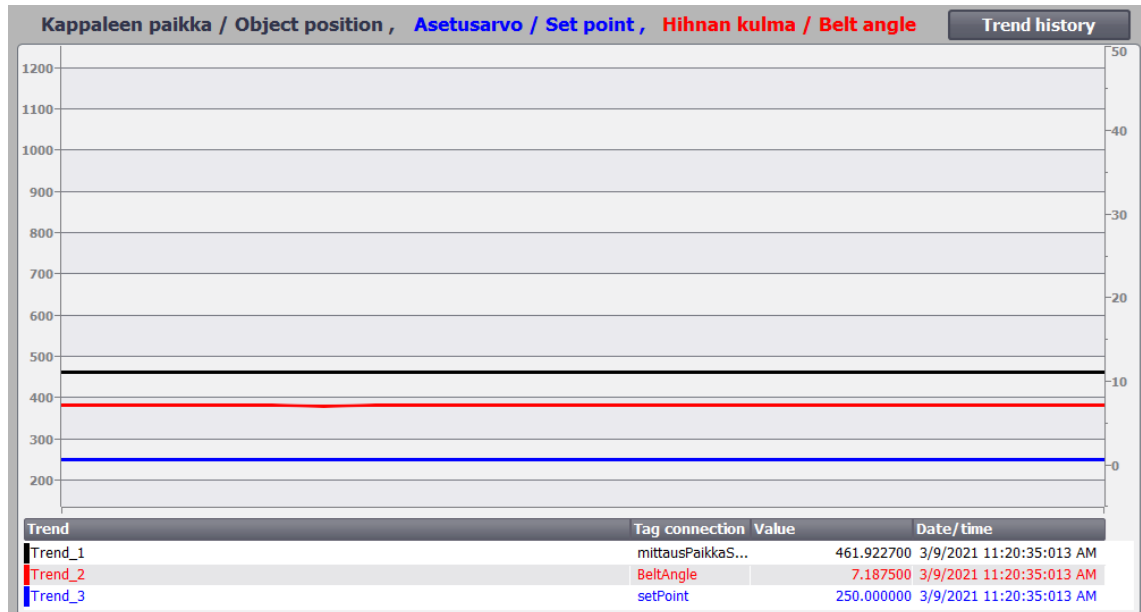
Ohjelman pääasiallisena näyttönä tämä näyttö onkin kaikista ruuhkaisin näyttö. Tämän näytön yläreunasta löytyvät kaikki järjestelmän ohjaamiseen tarvittavat työkalut. Kulman ohjauksen osassa näyttöä on myös io-kenttä näyttämässä kaltevuuskulman nykyistä arvoa. Kuva 16 on otos "Säätö"-näytöstä.



Kuva 16. "Säätö"-näytön yläosa

Kulman ohjauksen painikkeet ohjelmoitiin toimimaan kuten fyysisetkin painikkeet, eli niin kauan kuin hiiren näppäintä pidetään pohjassa painikkeen päällä, bitti pysyy arvossa 1. Manual/Auto-kytkin toimii kaksiasentoisena valitsimena, eli se säilyttää asentonsa, kunnes sitä vaihdetaan. Säätimen asetusarvon ja hihnan nopeuden ohjauskentät antavat käyttäjän syöttää luvun.

Näytön alempi osa on lähes kokonaan omistettu kuvaajanäytölle. Kuvaajan ot-sikko toimii myös redundanttina käyrien tunnistimena, sillä käyrät nimetään myös kuvaajan alareunassa. Kuva 17 esittää ”Säätö”-näytön kuvaajaa.

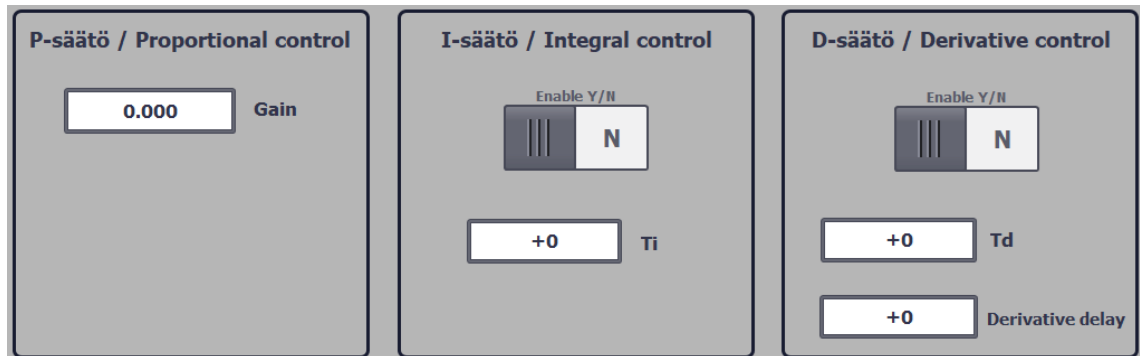


Kuva 17. ”Säätö”-näytön alaosa

”Säätö”-näytöltä löytyvät painikkeet kahdelle eri sivulle siirtymiseen, toinen säätimen parametrejä ja toinen kuvaajan historiaa varten. Näyttöjen ohjelmoinnista enemmän myöhemmässä osiossa.

”Parametrit”-näyttö

PID-säätimen vapaasti muokattavat parametrit ovat koottuna ”Parametrit”-näytölle. P-säätimen vahvistuskerroin, integraalisäädön integrointi aika, derivointisäädön derivointi aika ja derivointitoiminnon viive on jaettu asiamukaisten otsikoiden alle. I- ja D-säädöt on myös mahdollista yksitellen ottaa pois käytöstä tai käyttöön kytkintöiminnolla. Kuva 18 esittää ”Parametrit”-näytön asetusosaa.



Kuva 18. "Parametrit"-näytön asetusosa

Tämän näytön alareunassa on myös osio ohjausarvon näyttämiseksi, jonka vierestä löytää ohjauksen nollauspainikkeen. "Parametrit"-näytön alaosa on esitetty kuvassa 19.



Kuva 19. "Parametri"-näytön alaosa

Kuvaajanäytöt

"Säätö"-näytön kuvaaja esittää prosessin muuttujien arvoja reaaliajassa, viimeisen 20 sekunnin ajalta. Kuvaajan vasen reuna antaa paikan asteikon millimetreissä, arvosta 150 arvoon 1250. Kuvaajan oikean reunan asteikko esittää kaltevuuskulmaa asteikolla, joka alkaa arvosta -5° ja päättyy arvoon 50° .

"Kuvaajien historia"-näyttö sisältää vain kuvaajanäytön, jonka tarkoitus on näyttää prosessimuuttujien arvoja pidemmältä ajalta. Tätä varten piti luoda erillinen data log-tiedosto, johon tietokone tallentaa määrättyjen PLC-muuttujien arvoja. Data log-tiedoston luomisessa määritellään myös, kuinka usein lokiin kirjataan arvoja. Tämän projektin data log päivittyy kerran sekunnissa.

Liitteestä 5 voidaan nähdä, että "Kuvaajien historia"-näytöllä kuvaajan asteikot pystysuunnassa ovat samat kuin "Säätö"-näytölläkin, mutta aika-akseli on tällä

100 sekuntia. Tällä näytöllä myös kuvaajan omat selaustyökalut ovat näkyvissä, jotta käyttäjä voisi paremmin selata menneitä prosessiarvoja.

5 LABORATORIO HARJOITUKSET

Harjoitustehtävien kohdeyleisönä olivat automaatioon suuntautuneet kolmannen vuoden opiskelijat heidän kevätkaudellaan. PID-säädön pitäisi jo olla heille tuttu käsite. Sisällöltään harjoituksen tavoite oli pureutua säädön ilmiöihin kuten värähtelyyn tai integraalikyllästyksen. Tarkoitus oli myös luoda mielekäs, jopa innostava oppimisympäristö.

5.1 Ennakkotehtävät

Laboratorioharjoitusten ennakkotehtävien ensimmäinen tarkoitus on varmistaa, että opiskelijoilla on riittävät ennakkotiedot harjoitusta varten. Nämä tehtävät voivat myös auttaa suuntaamaan opiskelijoiden odotuksia laboratoriotyön suhteen. Opiskelijoiden itsenäinen tiedonhankinta ja mahdollisesti myös sosiaalisemmat tiedon jakamiseen perustuvat keinot korostuvat näissä tehtävissä.

Jotta siis PID-säätö ei olisi täysin unohtunutta tietoa laboratorioon tultaessa, ennakkotehtävät asettavat opiskelijat selvittämään säätimen perusedellytyksiä. Näiden pelkkä nimeäminen ei kuitenkaan riitä, vaan ne pitäisi myös selittää sanoin tai matemaattista kaavaa osoittaen. Hihnakuuljetinjärjestelmän käytössä olevan digitaalisen säätimen parametrilistan tuntemus olisi kuitenkin aivan liikaa pyytää kohdeyleisöltä. Ennakkotehtävä on nähtävissä liitteessä 6.

5.2 Laboriatoriötyö

Ensimmäinen harjoitustehtävä

Laboratoriotehtävän ensimmäisen osan tarkoitus on asettaa opiskelija ottamaan suurempaa vastuuta oppimisestaan tutkivan kokeilun kautta. Tällaista oppimista tukee myös konstruktivistinen oppimiskäsitys. Karjalainen (2006, 12) toteaa että konstruktivistisessä oppimisessä korostuu oppiminen opettamisen sijaan ja oppija opettajan sijaan.

Tehtävänanto tarjoaa ehdotuksia siitä, millaisista parametrien arvoista aloittaa kokeilu ja muistuttaa, että erilaiset kappaleet hinnalla tuottavat erilaisia tuloksia. Tarkoitus ei kuitenkaan ole antaa selkeästi määriteltyä kokeiden sarjaa, vaan opiskelijoiden vastuulle jää löytää sellaiset parametrit ja olosuhteet, jotka tuottavat kiinnostavia ilmiöitä. Tehtävänanto on kokonaisuudessaan nähtävissä liitteessä 6.

Hihnakuljetinjärjestelmä on luonteeltaan sellainen prosessi, jossa säätöilmiöt tulevat nopeasti näkyviin. Kokeiden suorittaminen vie vain suhteellisen vähän aikaa, ja valvomonäytön kautta on helppo muuttaa parametrejä seuraavaa koetta varten. Etäohjausjärjestelmä laadittiin tarkoituksellisesti niin, että opiskelijoilla olisi vapaat kädet kokeilla mitä järjestelmällä voi saada aikaan.

Toinen harjoitustehtävä

Jälkimmäisessä harjoituksessa opiskelijat pääsevät miettimään hihnakuljetinjärjestelmän ohjelman toteutusta. Tämä jakautuu kahteen osaan, jossa ensimmäisessä opiskelijoiden tehtävä on asettaa ohjelman päälohkon osat keskinäiseen järjestykseen. Liitteestä 6 nähdään, että lisäkysymyksenä esitetään kysymys PID-säätimen piirin suhteesta näihin muihin.

Tehtävän toisessa osassa opiskelijoiden tehtävä on yrittää luoda järjestelmän etäohjaustoiminto nykyaikaisemmassa FBD-formaatissa. Lähtökohtana toimivassa projektitiedostossa laitteisto on jo määritelty, mutta ohjelmasta on jäljellä vain hätäseis-toiminto ja muuttujaluettelo. Myös valvomonäytön ruudut ja HMI-muuttujaluettelo ovat jäljellä. Opettajan ohjeissa tarkennetaan tehtävänannon varoitusta kulman ohjauksen vaaroista, suositellen, että kulman ohjaus laaditaan myöhemmin.

6 POHDINTA

Työn tuloksena on saatu aikaan etäohjausjärjestelmä, johon kuuluvat toimiva säädin ja valvomonäyttö. Valitut mitta-anturit toimivat nekin oikein ja tarjoavat sekä järjestelmän että oppimisympäristön kannalta olennaista tietoa. Laboratorioharjoituksia kehitettiin ennakkotehtävän ja kahden laboratoriotehtävän verran.

6.1 Työn onnistuminen

Yleisesti ottaen voidaan todeta, että työ onnistui tavoitteissaan. Antureiden asennuksessa ei törmätty ongelmiin ja antureiden mittasignaalit kertovat sitä mitä niiltä odotettiin. Etäohjausjärjestelmä kykenee niihin toimiin, joita siltä vaadittiin. Valvomonäytön ja PLC:n välinen kommunikaatio toimii ja näyttö kykenee kiitettävällä tavalla näyttämään harjoitustehtävän kannalta olennaiset tiedot.

Moniin olemassa oleviin harjoitustyöpisteisiin kuuluu sähköpiirien muodostusta. Piirien kytkeminen ja sitä seuraava tarkastus vievät usein merkittävän määrän aikaa. Tätä työtä varten harjoitustöitä piti laatia enemmän, sillä harjoitustehtävän käytännön valmistelut lyhenevät huomattavasti, kun käytössä on valmiiksi laadittu tietokoneohjelma. Prosessin ohjaaminen tietokoneen näytöltä tarjoaa myös tilaisuuden näyttää prosessimuuttujien arvoja graafisesti ilman piirturia.

6.2 Kehittämisehdotuksia

Järjestelmään asennetut anturit asennettiin sillä ajatuksella, että vähällä vaivalla ne toimisivat riittävän hyvin. Kyseessä on kuitenkin vain prototyypilaitteisto. Koko projekti myös suoritettiin suhteellisen tiukalla aikataululla, mikä vaikutti ainakin järjestelmän ohjelmointiin. Toisin sanoen järjestelmässä on monia kehityksen kohteita.

6.2.1 Kulman ohjaus

Kuten kulma-anturin osiossa mainittiin, hihnan kaltevuuskulmaa voitaisiin käyttää säätimen mittausarvona. Kulman ohjaus kuitenkin toteutuu olennaisesti eri tavalla kuin hihnan nopeus. Nykyisellään kulman ohjaus pitäisi luultavasti toteuttaa pelkkänä suunnan ohjauksena, niin että säädin määrää vain mihin suuntaan kulmaa säädetään.

Kulman ohjaus ei myöskään ole lineaarista. Kulman arvo muuttuu nopeasti kulman arvon ollessa matala, ja hitaasti kun arvo on korkea. Yksinkertaisin tapa korjata tätä olisi kääntää ohjausjärjestelmä pystysuuntaan, edellyttäen ainakin reiän poraamista pöytään. Myös ohjaavaa moottoria saattaisi vaihtaa erityyppiseen. Toisaalta halvin ratkaisu kulman ohjaukseen saattaisi olla ohjelmallinen lisäys, jossa kulmaa ohjataan eri tavalla riippuen kulmakaltevuuden arvosta.

6.2.2 Koneturvallisuus

Järjestelmän turvallisuutta oli tarkasteltu etenkin sähköturvallisuuden osalta Jarmo Kallion tutkintotyön osana. Kallion (2008, 55) mukaan kuljetinlaitteiston sähkö- ja ohjausjärjestelmien toteutuksessa noudatettiin SFS-EN-60204-1 standardia. Hihnakuljetinjärjestelmässä on kuitenkin edelleen mainittavia huomionaiheita koneturvallisuuden kannalta.

Hihnan kaltevuuskulmaa nostettaessa hihnan pääty nousee suhteessa pöydän pintaan. Alas tullessaan huolimattoman käyttäjän käsi voisi jäädä hihnan ja hihnan tukirakenteen väliin puristuksiin. Myös itse hihna voi aiheuttaa vahinkoa varomattomalle käyttäjälle, joka onnistuu saamaan jotain hihnan ja laitteen väliin.

Yleisemmin järjestelmän koneturvallisuus paranisi sillä, että hihnalaitteiston ympärille asetettaisiin läpinäkyviä seiniä. Ohjauskotelon puolella tällaisessa seinässä pitäisi olla joko luukku tai valoverho, joilla tunnistettaisiin hihnaa lähestyvä kappale. Moottorit voitaisiin tällaisella kytkinsignaalilla sammuttaa.

6.2.3 Valvomonäytön ominaisuudet

Siemensin WinCC-ohjelmistolla voi toteuttaa monenlaisia asioita. Näytön elementtiin voi esimerkiksi yhdistää tooltip-tekstin, joka ilmestyy, kun elementtiä osoitetaan kursorilla. Tällä tapaa voitaisiin tarvittaessa selittää näytön toimintoja tarkemmin.

Tarkempi kehitysehdotus koskee nopeuden käsiohjausta näytöltä. Säätimen ohjausarvo kykenee lähettämään taajuusmuuttajalle kahdensuuntaisia ohjeita. Käsiohjauksen io-kenttä ei kuitenkaan tähän pysty. Tämän korjaaminen edellyttäisi uuden muuttujan luomista etäohjauksen ohjelmapiiriin.

Valvomonäytön voisi myös siirtää tietokoneelta erilliselle Siemensin valmistamalle paneelinäytölle. Tällainen oli hankittu Jarmo Kallion tutkintotyön yhteydessä. Se muuttaisi näytön toimintatavan kosketusnäytöksi, jolloin io-kenttää painettaessa näytölle avautuu näppäimistö.

LÄHTEET

ifm electronic GmbH. 2004. Mounting and installation instructions. EC2082.

ifm electronic GmbH. 2007. Operating instructions. Optical distance sensor 01D100.

Kallio J. 2008. Logiikkaohjattu hihnakuuljetin. Tampereen ammattikorkeakoulu. Tutkintotyö.

Karjalainen H. 2006. Oppimiskäsityksistä opettamiseen. Fenomenografinen tutkimus ammatillisten aikuisopettajien oppimis- ja opettamiskäsityksistä. Kasvatustieteiden tiedekunta. Tampereen Yliopisto. Pro gradu.

Kuoppala R. 1986. Anturit koneautomaatiossa. Tuotesuunnittelu 21/86. Helsinki: Metalliteollisuuden Kustannus Oy

Pepperl+Fuchs GmbH. 2010. ULTRASONIC SENSORS M30. Betriebsanleitung/Operating instructions.

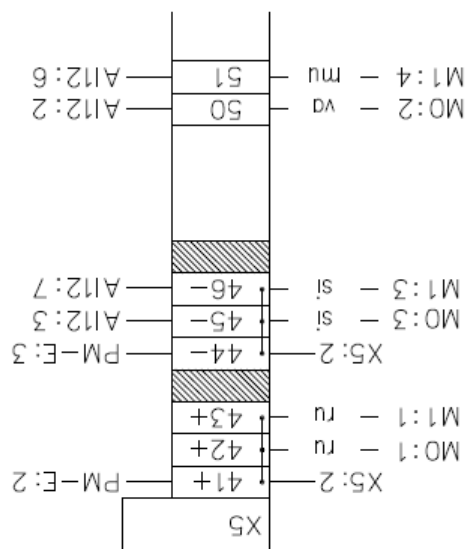
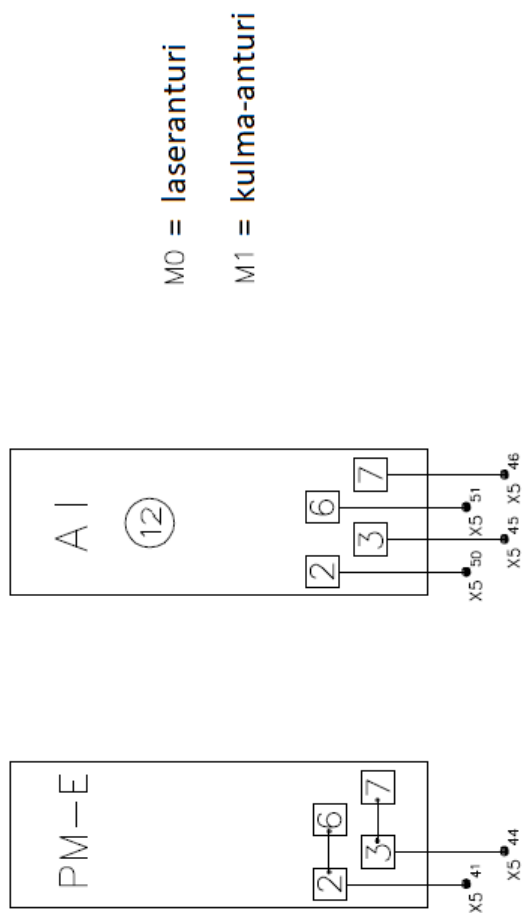
Seborg D., Edgar T., Mellichamp D., Doyle F. 2011. Process Dynamics and Control. International Student Version. Third Edition. Asia: John Wiley & Sons

Siemens. 2007. SIMATIC ET 200S distributed I/O 2AI 2WIRE ST analog electronic module (6ES7134-4GB01-0AB0). Manual.

Profinet University. n.d. PROFINET GSD File Basics. Luettu 10.3.2021. <https://profinetuniversity.com/profinet-basics/profinet-gsd-file-basics/>

LIITTEET

Liite 1. Antureiden kytkentäkaavio



Liite 2. PLC:n muuttujaluettelo

Totally Integrated Automation Portal									
Hihnakuljetinettä / PLC_1 [CPU 315-2 PN/DP] / PLC tags									
Default tag table [49]									
PLC tags									
Name	Data type	Address	Retain	Accessi-ble from HMI/OPC UA/Web API	Writab-le from HMI/OPC UA/Web API	Visible in HMI engi-neering	Supervision	Comment	
laserMittaus	Int	%W284		True	True	True		Laseranturin mittaviesti	
setRemoteControl	Bool	%I.1		True	True	True		Etsiöjauksen valinta	
localHihnaRunning	Bool	%M20.1		True	True	True			
hihnaMoottoriStart	Word	%QW268		True	True	True		Kuljettimen moottorin käynnistys-osoite	
localHihnaNopeusPlus	Bool	%I2.0		True	True	True			
localHihnaNopeusMinus	Bool	%I2.1		True	True	True			
localHihnaNopeusohje	Word	%MW50		True	True	True			
hihnaMoottoriNopeus	Word	%QW270		True	True	True		Kuljettimen moottorin nopeusohjeen osoite	
remHihnaRunning	Bool	%M20.2		True	True	True			
remHihnaStart	Bool	%M25.0		True	True	True		Kuljetin käyntiin (remote-tilassa)	
remHihnaStop	Bool	%M25.1		True	True	True		Kuljetin seis (remote-tilassa)	
remHihnaNopeusohje	Word	%MW34		True	True	True			
kulmaMittaus	Word	%W286		True	True	True		Kulma-anturin mittaviesti	
kulmaTarpeeton	Word	%MW28		True	True	True			
hihnanKulma	Real	%MD210		True	True	True		Tilapiinen	
inKulma	Int	%MW100		True	True	True		Tilapiinen	
skaalattuKulma	Real	%MD200		True	True	True		Tilapiinen	
korjattuKulma	Real	%MD220		True	True	True		Tilapiinen	
remKulmaPlus	Bool	%M22.0		True	True	True		Kaltevuuskulma (+) (remote-tilassa)	
rajaKulmaHigh	Bool	%I4.0		True	True	True		Yläraja saavutettu	
remKulmaMinus	Bool	%M22.1		True	True	True		Kaltevuuskulma (-) (remote-tilassa)	
rajaKulmaLow	Bool	%I4.1		True	True	True		Alaraja saavutettu	
remKulmaZero	Bool	%M22.2		True	True	True		Ajo nollakulmaan (remote-tilassa)	
kulmanNollaus	Bool	%M30.1		True	True	True			
kulmaMoottoriStart	Word	%QW256		True	True	True		Kulmasäätimen moottorin käynnistysoite	
kulmaMoottoriNopeus	Word	%QW258		True	True	True		Kulmasäätimen moottorin nopeusohjeen osoite	
localKulmaPlus	Bool	%I2.2		True	True	True		Kaltevuuskulma (+) (local-tilassa)	
localKulmaMinus	Bool	%I2.3		True	True	True		Kaltevuuskulma (-) (local-tilassa)	
localKulmaZero	Bool	%I3.0		True	True	True		Ajo nollakulmaan (local-tilassa)	
paikkaTarpeeton	Word	%MW40		True	True	True			
mittausPaikkaSkaalattu	Real	%MD42		True	True	True		Kuvaajille ja säätimeen menevä paikan mittaus-tieto	
inPaikka	Int	%MW38		True	True	True			
setLocalControl	Bool	%I1.0		True	True	True		Paikallisohjauksen valinta	
integralControl	Bool	%M61.0		True	True	True		Integraalisäädin valinta	
derivativeControl	Bool	%M61.1		True	True	True		Derivointisäädin valinta	
controllerGain	Real	%MD64		True	True	True		Vahvistuskertoin	
timeIntegralTime	Time	%MD68		True	True	True		Integrointi-aika	
timeDerivativeTime	Time	%MD72		True	True	True		Derivointi-aika	
timeDerivativeDelay	Time	%MD76		True	True	True		Derivoinnin viive	
realOhjausarvo	Real	%MD80		True	True	True		Säätimen ohjausarvo	
setAutoMode	Bool	%M60.0		True	True	True		Automaattisen ohjauksen valinta	
controllerSetPoint	Real	%MD56		True	True	True		Säätimen asetusarvo	
ohjauksenNollaus	Bool	%M10.0		True	True	True		Säätimen reset-käsky	
systemStart	Byte	%QB0		True	True	True			
localHihnaStart	Bool	%I1.2		True	True	True		Kuljetin käyntiin (local-tilassa)	
localHihnaStop	Bool	%I1.3		True	True	True		Kuljetin seis (local-tilassa)	

Liite 3. Logiikan ohjelma

1 (6)

Totally Integrated Automation Portal							
Hihnakuljetinetä / PLC_1 [CPU 315-2 PN/DP] / Program blocks							
OB1 [OB1]							
OB1 Properties							
General							
Name	OB1	Number	1	Type	OB	Language	STL
Numbering	Manual						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					
OB1							
Name	Data type	Offset	Default value	Comment			
▼ Temp							
Temp_0	Byte	0.0					
Temp_1	Byte	1.0					
Temp_2	Byte	2.0					
Temp_3	Byte	3.0					
Temp_4	Byte	4.0					
Temp_5	Byte	5.0					
Temp_6	Int	6.0					
Temp_7	Int	8.0					
Temp_8	Int	10.0					
Temp_9	Date_And_Time	12.0					
Constant							
Network 1: Turvapiirin tila							
0001	A	"turvaTarkistus"					
0002	=	"valoGreen"					
0003	=	"ohjelmaSallittu"					
0004	AN	"turvaTarkistus"					
0005	=	"valoRed"					
0006	JC	M002					
Network 2: Paikallisohjauksen kutsu							
0001	A	"ohjelmaSallittu"					
0002	A	"setLocalControl"					
0003	CC	"FC10"					
Network 3: Etäohjauksen kutsu							
0001	A	"ohjelmaSallittu"					
0002	A	"setRemoteControl"					
0003	JCN	M003					
0004	CALL	"FC20"					
0005		AngleInt := "intKulma"					
0006		AngleScaled := "skaalattuKulma"					
0007		PositionInt := "intPaikka"					
Network 4: Moottoreiden sammutus							
0001	M003: AN	"setLocalControl"					
0002	AN	"setRemoteControl"					
0003	JCN	M001					
0004	L	0					
0005	T	"kulmaMoottoriStart":F					
0006	T	"hihnaMoottoriStart":F					
0007	R	"localHihnaRunning"					
0008	S	"ohjauksenNollaus"					
Network 5: Turvapiiri päästännyt							
0001	M002: R	"ohjelmaSallittu"					
0002	L	0					
0003	T	"kulmaMoottoriStart":F					
0004	T	"hihnaMoottoriStart":F					
0005	T	"kulmaMoottoriNopeus":F					
0006	T	"hihnaMoottoriNopeus":F					
0007	R	"localHihnaRunning"					
0008	S	"ohjauksenNollaus"					
0009	M001: BE						

(jatkuu)

Totally Integrated Automation Portal					
Hihnakuljetinettä / PLC_1 [CPU 315-2 PN/DP] / Program blocks					
FC10 [FC10]					
FC10 Properties					
General					
Name	FC10	Number	10	Type	FC
Numbering	Manual			Language	STL
Information					
Title		Author		Comment	
Version	0.1	User-defined ID		Family	
FC10					
Name	Data type	Offset	Default value	Comment	
Input					
Output					
InOut					
Temp					
Constant					
▼ Return					
FC10	Void				
Network 1: Kulman säätö					
0001	A		"localKulmaPlus"		
0002	AN		"rajaKulmaHigh"		
0003	JC		M001		
0004	A		"localKulmaMinus"		
0005	AN		"rajaKulmaLow"		
0006	JC		M002		
0007	A		"localKulmaZero"		
0008	AN		"rajaKulmaLow"		
0009	S		"kulmanNollaus"		
0010	A		"rajaKulmaLow"		
0011	R		"kulmanNollaus"		
0012	A		"kulmanNollaus"		
0013	JC		M017		
0014	AN		"localKulmaPlus"		
0015	JC		M003		
0016	AN		"localKulmaMinus"		
0017	JC		M003		
0018	M001: L		16#047F		
0019	T		"kulmaMoottoriStart":P		
0020	L		-5000		
0021	T		"kulmaMoottoriNopeus":P		
0022	JU		M013		
0023	M002: L		16#047F		
0024	T		"kulmaMoottoriStart":P		
0025	L		5000		
0026	T		"kulmaMoottoriNopeus":P		
0027	JU		M013		
0028	M017: L		16#047F		
0029	T		"kulmaMoottoriStart":P		
0030	L		5000		
0031	T		"kulmaMoottoriNopeus":P		
0032	JU		M013		
0033	M003: L		0		
0034	T		"kulmaMoottoriNopeus":P		
0035	R		"kulmanNollaus"		
0036	M013: NOP		0		
Network 2: Nopeuden säätö					
0001	A		"localHihnaStart"		
0002	S		"localHihnaRunning"		
0003	A		"localHihnaStop"		
0004	R		"localHihnaRunning"		
0005	A		"localHihnaRunning"		
0006	JCN		M005		
0007	L		16#047F		
0008	T		"hihnaMoottoriStart":P		
0009	A		"localHihnaRunning"		
0010	A		"localHihnaNopeusPlus"		
0011	JC		M008		
0012	A		"localHihnaRunning"		
0013	A		"localHihnaNopeusMinus"		
0014	JC		M009		
0015	AN		"localHihnaNopeusPlus"		
0016	JC		M011		
0017	AN		"localHihnaNopeusMinus"		
0018	JC		M011		
0019	M008: L		"localHihnaNopeusohje"		
0020	L		-10000		

Totally Integrated Automation Portal			
0021	==I		
0022	JC	M015	
0023	L	"localHihnaNopeusohje"	
0024	+	-1	
0025	T	"localHihnaNopeusohje"	
0026	T	"hihnaMoottoriNopeus":P	
0027	M015: JU	M006	
0028	M009: L	"localHihnaNopeusohje"	
0029	L	0	
0030	==I		
0031	JC	M016	
0032	L	"localHihnaNopeusohje"	
0033	+	1	
0034	T	"localHihnaNopeusohje"	
0035	T	"hihnaMoottoriNopeus":P	
0036	M016: JU	M006	
0037	M005: L	0	
0038	T	"hihnaMoottoriStart":P	
0039	L	0	
0040	T	"localHihnaNopeusohje"	
0041	JU	M006	
0042	M011: L	"localHihnaNopeusohje"	
0043	T	"hihnaMoottoriNopeus":P	
0044	M006: BE		

Totally Integrated Automation Portal			
Hihnakuljetineta / PLC_1 [CPU 315-2 PN/DP] / Program blocks			
FC20 [FC20]			
FC20 Properties			
General			
Name	FC20	Number	20
Type	FC	Language	FBD
Numbering	Manual		
Information			
Title		Author	
Version	0.1	User-defined ID	
FC20			
Name	Data type	Offset	Default value
Input			
Output			
▼ InOut			
AngleInt	Int		
AngleScaled	Real		
PositionInt	Int		
▼ Temp			
Temp0	Bool	0.0	
Temp1	Word	2.0	
Constant			
▼ Return			
FC20	Void		
Network 1: Kappaleen paikan mittauksen skaalaus			
Skaalaus tapahtuu näytön ohjelmassa			
0001	L	"laserMittaus":P	//Laseranturin viesti
0002	T	#PositionInt	
0003	CALL	SCALE	//Muutos millimetreiksi
0004	IN	:=#PositionInt	
0005	HI_LIM	:=10000.0	
0006	LO_LIM	:= -30.0	
0007	BIPOLAR	:= "vakioNolla"	
0008	RET_VAL	:= "paikkaTarpeeton"	
0009	OUT	:= "mittausPaikkaSkaalattu"	
Network 2: Hihnan kulman mittauksen skaalaus			
Skaalaukseen kuuluu negatiivisten lukejn mahdollisuus, joten skaalaus logiikassa.			
0001	L	"kulmaMittaus":P	//Kulma-anturin viesti
0002	T	#AngleInt	
0003	CALL	SCALE	//Muutos asteiksi
0004	IN	:=#AngleInt	
0005	HI_LIM	:=0.0	
0006	LO_LIM	:=180.0	
0007	BIPOLAR	:= "vakioNolla"	
0008	RET_VAL	:= "kulmaTarpeeton"	
0009	OUT	:=#AngleScaled	
0010	L	90.0	
0011	-R		
0012	T	"hihnanKulma"	

Network 3: Hihnan kulman ohjaus

```

0001      A      "remKulmaPlus"
0002      AN     "rajaKulmaHigh"
0003      JC     M001
0004      A      "remKulmaMinus"
0005      AN     "rajaKulmaLow"
0006      JC     M002
0007      A      "remKulmaZero"
0008      AN     "rajaKulmaLow"
0009      S      "kulmanNollaus"
0010      A      "rajaKulmaLow"
0011      R      "kulmanNollaus"
0012      A      "kulmanNollaus"
0013      JC     M017
0014      AN     "remKulmaPlus"
0015      JC     M003
0016      AN     "remKulmaMinus"
0017      JC     M003
0018 M001: L    16#047F      //Kulma ylös
0019      T      "kulmaMoottoriStart":P
0020      L      -5000
0021      T      "kulmaMoottoriNopeus":P
0022      JU     M013
0023 M002: L    16#047F      //Kulma alas
0024      T      "kulmaMoottoriStart":P
0025      L      5000
0026      T      "kulmaMoottoriNopeus":P
0027      JU     M013
0028 M017: L    16#047F      //Kulma nolllaan
0029      T      "kulmaMoottoriStart":P
0030      L      5000
0031      T      "kulmaMoottoriNopeus":P
0032      JU     M013
0033 M003: L    0          //Stop
0034      T      "kulmaMoottoriNopeus":P
0035      R      "kulmanNollaus"
0036 M013: NOP 0
0037
0038

```

Network 4: Nopeuden ohjaus

```

0001      A      "remHihnaStart"
0002      S      "remHihnaRunning"
0003      A      "remHihnaStop"
0004      R      "remHihnaRunning"
0005      A      "remHihnaRunning"
0006      JCN   M005      //Hihnan käynnistys
0007      L      16#047F
0008      T      "hihnaMoottoriStart":P
0009      AN     "setAutoMode"
0010      JC     M009
0011      L      "realOhjausarvo"      //Nopeuden ohjaus säätimeltä
0012      RND
0013      T      "hihnaMoottoriNopeus":P
0014      JU     M006
0015 M009: L    "remHihnaNopeuschje" //Nopeuden manuaalinen asetus
0016      L      "vakioMiinusYksi"
0017      *I
0018      T      "hihnaMoottoriNopeus":P
0019      JU     M006
0020 M005: L    0          //Stop
0021      T      "hihnaMoottoriStart":P
0022      L      0
0023      T      "remHihnaNopeuschje"
0024      JU     M006
0025 M006: BE
0026
0027

```

Hihnakuljetinettä / PLC_1 [CPU 315-2 PN/DP] / Program blocks

OB35 [OB35]

OB35 Properties

General							
Name	OB35	Number	35	Type	OB	Language	FBD
Numbering	Manual						
Information							
Title	"Cyclic Interrupt"	Author		Comment		Family	
Version	0.1	User-defined ID					

OB35

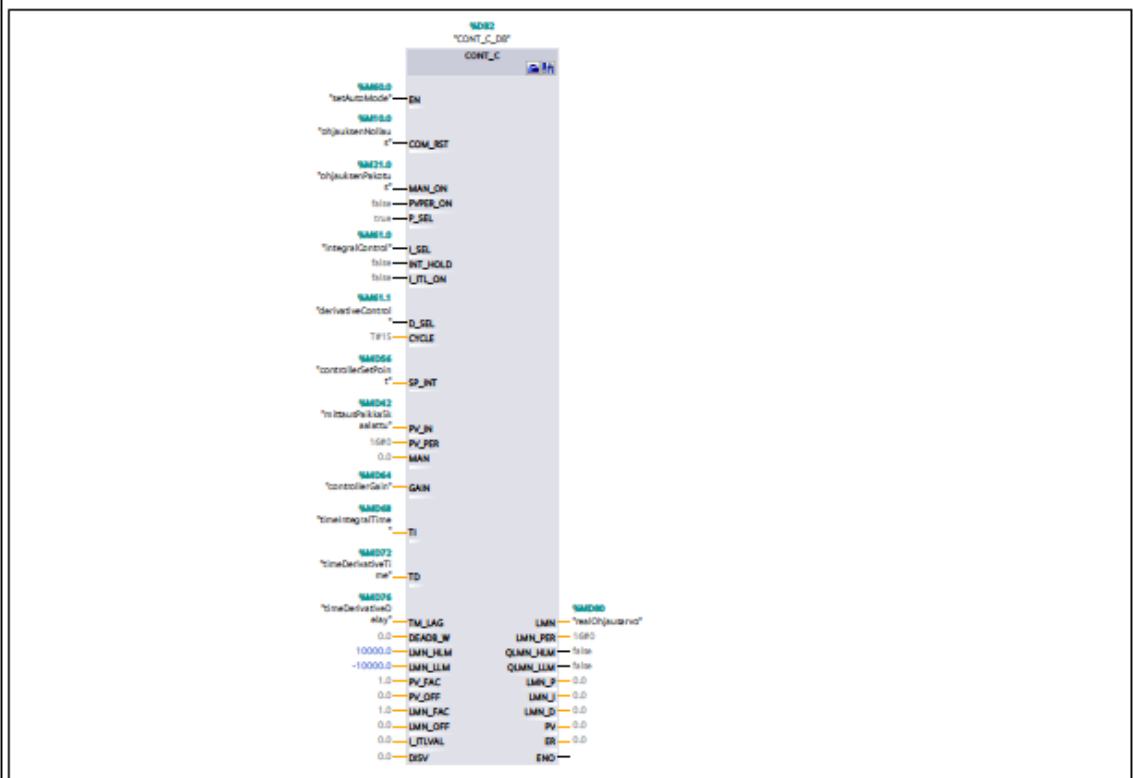
Name	Data type	Offset	Default value	Comment
▼ Temp				
OB35_EV_CLASS	Byte	0.0		Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB35_START_INF	Byte	1.0		16#36 (OB 35 has started)
OB35_PRIORITY	Byte	2.0		Priority of OB Execution
OB35_OB_NUMBR	Byte	3.0		35 (Organization block 35, OB35)
OB35_RESERVED_1	Byte	4.0		Reserved for system
OB35_RESERVED_2	Byte	5.0		Reserved for system
OB35_PHASE_OFFSET	Word	6.0		Phase offset (msec)
OB35_RESERVED_3	Int	8.0		Reserved for system
OB35_EXC_FREQ	Int	10.0		Frequency of execution (msec)
OB35_DATE_TIME	Date_And_Time	12.0		Date and time OB35 started
Constant				

Network 1: Asetusarvon rajoitus

```

0001      L      "controllerSetPoint"
0002      L      1050.0
0003      >R
0004      JC      M001
0005      L      "controllerSetPoint"
0006      L      200
0007      <R
0008      JC      M001
    
```

Network 2: Säädin



Network 3: Säätimen reset-bitin nollaus

```

0001      R      "ohjauksenNollaus"
0002
    
```

Network 4: Piirin loppu

```

0001 M001: BE
    
```

Totally Integrated Automation Portal							
Hihnakuljetinettä / PLC_1 [CPU 315-2 PN/DP] / Program blocks OB100 [OB100]							
OB100 Properties							
General							
Name	OB100	Number	100	Type	OB	Language	STL
Numbering	Manual						
Information							
Title		Author		Comment		Family	
Version	0.0	User-defined ID					
OB100							
Name		Data type	Offset	Default value	Comment		
▼ Temp							
Temp_0		Byte	0.0				
Temp_1		Byte	1.0				
Temp_2		Byte	2.0				
Temp_3		Byte	3.0				
Temp_4		Byte	4.0				
Temp_5		Byte	5.0				
Temp_6		Word	6.0				
Temp_7		DWord	8.0				
Temp_8		Date_And_Time	12.0				
Constant							
Network 1:							
0001	L	1					
0002	T	"systemStart"					
0003	S	"ohjauksenNollaus"					

Liite 4. Valvomonäytön HMI-muuttujaluettelo

1 (2)

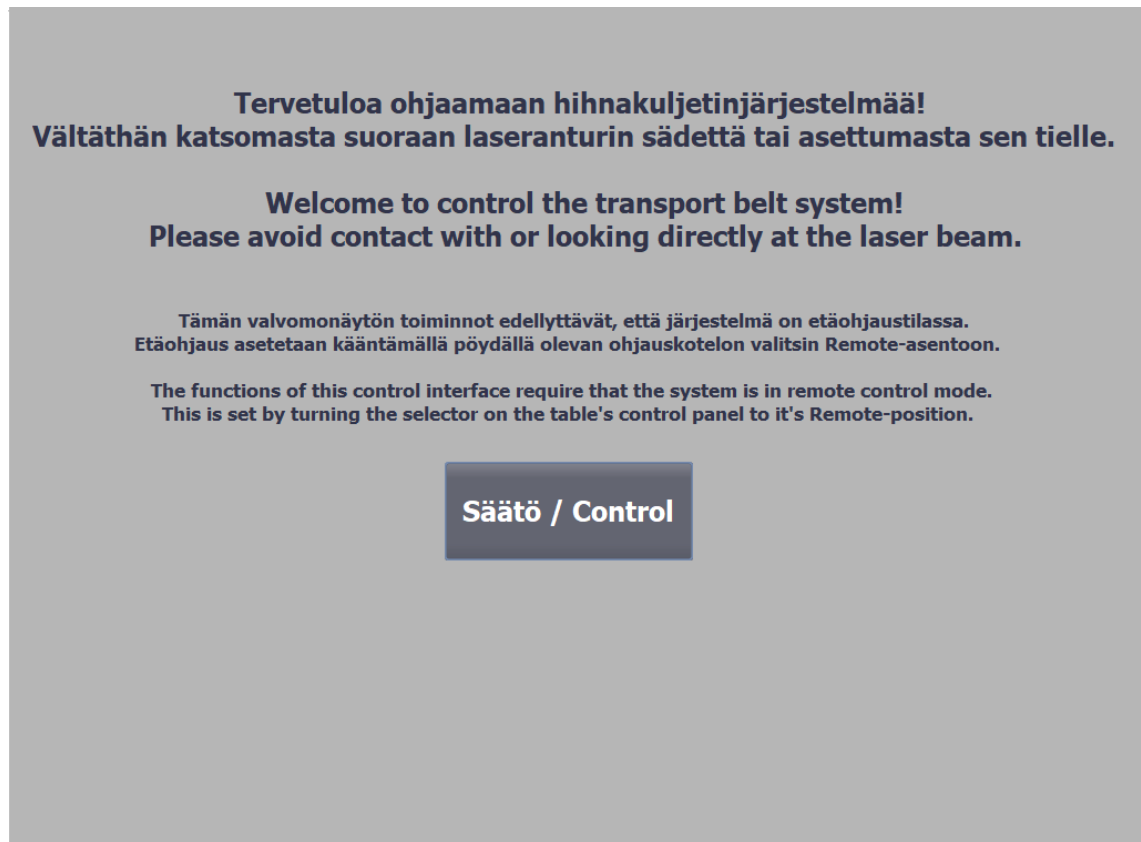
Totally Integrated Automation Portal					
Hihnakuljetinettä / PC-System_1 [SIMATIC PC station] / HMI_RT_1 [WinCC RT Advanced] / HMI tags					
Default tag table [22]					
BeltStart					
Name	BeltStart	Display name		Address	%M25.0
Connection	HMI_Connection_1	Data type	Bool	Length	1
remHihnaStart					
BeltStop					
Name	BeltStop	Display name		Address	%M25.1
Connection	HMI_Connection_1	Data type	Bool	Length	1
remHihnaStop					
BeltStatus					
Name	BeltStatus	Display name		Address	%M20.2
Connection	HMI_Connection_1	Data type	Bool	Length	1
remHihnaRunning					
ObjectDistance					
Name	ObjectDistance	Display name	ObjectDistance	Address	%MD42
Connection	HMI_Connection_1	Data type	Real	Length	4
mittausPaikkaSkaalattu					
BeltSpeed					
Name	BeltSpeed	Display name	BeltSpeed	Address	%MW34
Connection	HMI_Connection_1	Data type	Word	Length	2
remHihnaNopeusohje					
BeltAngle					
Name	BeltAngle	Display name	BeltAngle	Address	%MD210
Connection	HMI_Connection_1	Data type	Real	Length	4
hihnanKulma					
AnglePlus					
Name	AnglePlus	Display name		Address	%M22.0
Connection	HMI_Connection_1	Data type	Bool	Length	1
remKulmaPlus					
AngleMinus					
Name	AngleMinus	Display name		Address	%M22.1
Connection	HMI_Connection_1	Data type	Bool	Length	1
remKulmaMinus					
AngleZero					
Name	AngleZero	Display name		Address	%M22.2
Connection	HMI_Connection_1	Data type	Bool	Length	1
remKulmaZero					
laserMittaus					
Name	laserMittaus	Display name		Address	%IW284
Connection	HMI_Connection_1	Data type	Int	Length	2
laserMittaus					
mittausPaikkaSkaalattu					
Name	mittausPaikkaSkaalattu	Display name		Address	%MD42
Connection	HMI_Connection_1	Data type	Real	Length	4
mittausPaikkaSkaalattu					
switchManualAuto					
Name	switchManualAuto	Display name		Address	%M60.0
Connection	HMI_Connection_1	Data type	Bool	Length	1

2 (2)

setAutoMode					
setPoint					
Name	setPoint	Display name		Address	%MD56
Connection	HMI_Connection_1	Data type	Real	Length	4
controllerSetPoint					
screenControllerGain					
Name	screenControllerGain	Display name		Address	%MD64
Connection	HMI_Connection_1	Data type	Real	Length	4
controllerGain					
screenIntegralTime					
Name	screenIntegralTime	Display name		Address	%MD68
Connection	HMI_Connection_1	Data type	Time	Length	4
timeIntegralTime					
screenDerivativeTime					
Name	screenDerivativeTime	Display name		Address	%MD72
Connection	HMI_Connection_1	Data type	Time	Length	4
timeDerivativeTime					
screenDerivativeDelay					
Name	screenDerivativeDelay	Display name		Address	%MD76
Connection	HMI_Connection_1	Data type	Time	Length	4
timeDerivativeDelay					
setIntegralControl					
Name	setIntegralControl	Display name		Address	%M61.0
Connection	HMI_Connection_1	Data type	Bool	Length	1
integralControl					
setDerivativeControl					
Name	setDerivativeControl	Display name		Address	%M61.1
Connection	HMI_Connection_1	Data type	Bool	Length	1
derivativeControl					
screenOhjausarvo					
Name	screenOhjausarvo	Display name		Address	%MD80
Connection	HMI_Connection_1	Data type	Real	Length	4
realOhjausarvo					
controlReset					
Name	controlReset	Display name		Address	%M10.0
Connection	HMI_Connection_1	Data type	Bool	Length	1
ohjauksenNollaus					

Liite 5. Valvomonäyttö

1 (4)



(jatkuu)



Palaa säätöön / Return to control **Parametrit / Parameters**

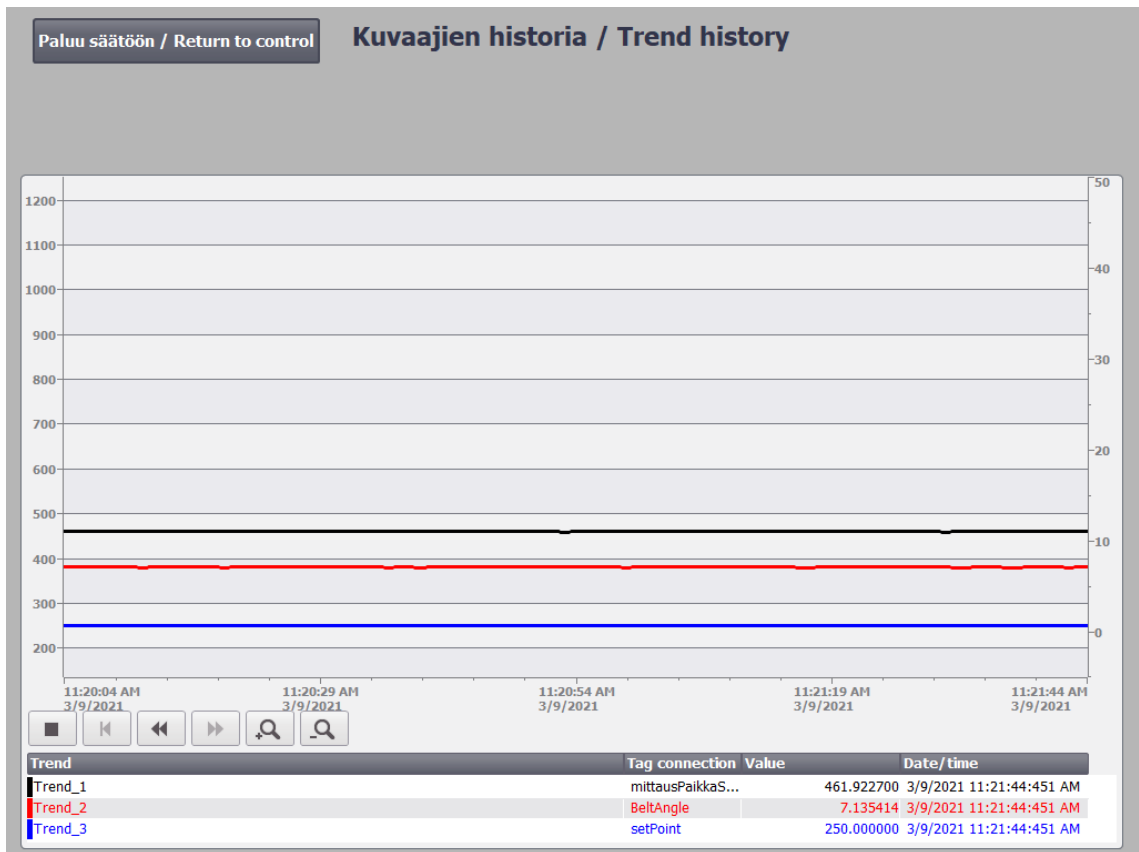
Huomaathan, että nämä parametrit koskevat vain automaattista ohjausta.
Please note that these parameters concern only the automatic mode of control.

I- ja D-säätöjen muuttujat annetaan ajan yksiköissä, tarkemmin sanottuna millisekunneissa.
The variables of I and D control are given in units of time, specifically milliseconds.

P-säätö / Proportional control	I-säätö / Integral control	D-säätö / Derivative control
<input type="text" value="0.000"/> Gain	Enable Y/N <input type="checkbox"/> <input checked="" type="checkbox"/> N	Enable Y/N <input type="checkbox"/> <input checked="" type="checkbox"/> N
	<input type="text" value="+0"/> Ti	<input type="text" value="+0"/> Td
		<input type="text" value="+0"/> Derivative delay

Säätimen ulostulo / Controller output

<input type="text" value="+0.000"/> Ohjausarvo / Control signal	<input type="button" value="Ohjauksen nollaus / Control reset"/>
---	--



Liite 6. Laboratorio harjoitus

SÄÄTÖTEKNIIKAN LABORATORIO HARJOITUS**Ennakkotehtävä**

Mitä sisääntuloja tai parametreja itsenäinen PID-säädin tarvitsee (5 asiaa)? Mitä nämä tarkoittavat? Voit käyttää kaavioita ja/tai matemaattisia kaavoja selityksen tukena.

Laboratoriotehtävät

Tehtävä 1:

Hihnakuljetinjärjestelmässä mitattavana suureena toimii kappaleen etäisyys hihnan päästä. Ohjaus tapahtuu hihnan nopeuden kautta. Etäisyyteen vaikuttavat säätimen parametrien lisäksi hihnan kaltevuuskulma ja hihnalla olevan kappaleen ominaisuudet.

Tutkkaa prosessin käyttäytymistä: mitä kappaleen etäisyydelle tapahtuu ja miksi? Kokeet voi aloittaa leluveneellä, säätimen parametriksi $K_P = 5$.

Tehtävä 2:

- a) Järjestelmän logiikan ohjelman päälohko jakautuu useampaan ohjelmapiiriin, jotka kuitenkin suoritetaan järjestyksessä. Näiden osien nimet ovat listattu alle satunnaiseen järjestykseen.

Mikä olisi mielekäs järjestys näille ohjelmapiireille? Miksi?

- Paikallisohjauksen kutsu
- Turvapiiri päästänyt
- Turvapiirin tila
- Moottoreiden sammutus
- Etäohjauksen kutsu

PID-säätimellä on ohjelmassa oma lohkonsa. **Miten PID-säätimen ohjelmalohko sijoittuu suhteessa päälohkon piireihin?**

- b) PLC:n ohjelma on tehty STL-formaatissa. **Laatikkaa ohjelma nykyaikaisemmaksi FBD-formaattia käyttäen.** Järjestelmän ohjelmamuuttajat on listattu muuttujaluetteloihin (PLC- ja HMI default Tag Table). Taajuusmuuttajat käynnistyvät hexadesimaaliluvulla 16#047F, ja niiden nopeusohjeet annetaan kokonaislukuna: minimi -10000, maksimi 10000. **Huom!** Kulman ohjauksen kanssa on mahdollista vaurioittaa laitteistoa, mikäli ohjausta ei rajoiteta rajakytkimillä.

Liite 7. Opettajan ohjeet laboratorio harjoitusta varten

1 (2)

Laboratoriotehdävät

Kun hihnakuljetinjärjestelmän käynnistää, turvapiiri aloittaa estävästä tilasta, jota punainen valo indikoi. Turvapiiri resetoidaan painamalla hätäseis-painiketta, vetämällä se auki ja painamalla sitten ”turvapiirin resetointi”-painiketta. Indikaattorin pitäisi vaihtua vihreäksi.

Tehtävä 1:

Hihnakuljetinjärjestelmässä mitattavana suureena toimii kappaleen etäisyys hihnan päästä. Ohjaus tapahtuu hihnan nopeuden kautta. Etäisyyteen vaikuttavat säätimen parametrien lisäksi hihnan kaltevuuskulma ja hihnalla olevan kappaleen ominaisuudet.

Tutkikaa prosessin käyttäytymistä: mitä kappaleen etäisyydelle tapahtuu ja miksi? Kokeet voi aloittaa leluveneellä, säätimen parametriksi $K_P = 5$.

Opettajan rooli on esitellä järjestelmän ohjauksen komponentit ja tarjota ehdotuksia ilmiöistä, joita toteuttaa järjestelmällä. Järjestelmän ohjauksen komponentit ovat

- pääkytkin (logiikan kaapin ovessa)
- hätäseis (logiikan kaapin kyljessä)
- turvapiirin resetointipainike (hätäkatkaisimen alapuolella)
- ohjauskotelo, etenkin ohjaustavan valitsin
- tietokone, jolta valvomonäyttöä käytetään (HihnakuljetinEtä.ap).

Etsittäviä ilmiöitä ovat ainakin pysyvä värähtelytila ja integraaliohjauksen yhteydessä esiintyvä väliaikainen värähtely. On myös hyvä tutkia eri säätömenetelmien välisiä hienovaraisempia eroja.

Leluauton kanssa voi myös etsiä niitä kaltevuuskulman ja hihnan nopeuden arvoja, joilla auton asema muuttuu sen sijaan, että auto polkee paikallaan.

Tehtävä 2:

- a) Järjestelmän logiikan ohjelman päälohko jakautuu useampaan ohjelmapiiriin, jotka kuitenkin suoritetaan järjestyksessä. Näiden osien nimet ovat listattu alle satunnaiseen järjestykseen.

Mikä olisi mielekäs järjestys näille ohjelmapiireille? Miksi?

- Paikallisohjauksen kutsu
- Turvapiiri päästänyt
- Turvapiirin tila
- Moottoreiden sammutus
- Etäohjauksen kutsu

Piirit voivat toki olla useammassa eri järjestyksessä ja silti toimia. Ainakaan paikallisohjauksen ja etäohjauksen kutsujärjestyksellä ei ole olennaista merkitystä. Ohjelman piirien väillä on mahdollista hyppiä, mutta tämä on rajoitettu eteenpäin menemiseen, joten järjestyksellä on merkitystä.

(jatkuu)

PID-säätimellä on ohjelmassa oma lohkonsa. Miten PID-säätimen ohjelmalohko sijoittuu suhteessa päälohkon piireihin?

Tämä saattaa hämmentää opiskelijoita, mikäli sykliset toimilohkot eivät ole tuttuja. Ohjelman suorittamiseen menevä aika vaihtelee, joten säätimen kutsuminen ohjelman sisältä voisi aiheuttaa epäsäännöllisiä säätötoimia.

- b) PLC:n ohjelma on tehty STL-formaatissa. **Laatkaa ohjelma nykyaikaisemmaksi FBD-formaattia käyttäen.** Järjestelmän ohjelmamuuttajat on listattu muuttujaluetteloihin (PLC- ja HMI default Tag Table). Taajuusmuuttajat käynnistyvät hexadesimaaliluvulla 16#047F, ja niiden nopeusohjeet annetaan kokonaislukuna: minimi -10000, maksimi 10000. **Huom!** Kulman ohjauksen kanssa on mahdollista vaurioittaa laitteistoa, mikäli ohjausta ei rajoiteta rajakytkimillä.

On hyvä suositella, että kulman ohjaus ohjelmoidaan myöhemmin. Tällä tapaa ensimmäiset ohjausyritykset eivät kohdistu riskialttiiseen laitteeseen.