

Jimi Nisula

MONIMUUTTUJA-PI-SÄÄTÖ AUTOMAATIOJÄRJESTELMÄSSÄ

Kandidaatintutkielma
Tekniikan ja luonnontieteiden tiedekunta
Tarkastaja: Yrjö Majanne

Huhtikuu 2022

TIIVISTELMÄ

Jimi Nisula: Monimuuttuja-PI-säätö automaatiojärjestelmässä
Kandidaatintutkielma
Tampereen yliopisto
Automaatiotekniikka
Huhtikuu 2022

Monimuuttujaprosesseissa esiintyvät ristikkäisvaikutukset haittaavat prosessin säätöä, koska ne aiheuttavat säädettäviin suureisiin epätoivottavia muutoksia. Ristikkäisvaikutukset syntyvät siitä, kun muutos tai häiriö yhdessä säätöpiirissä häiritsee toisen säätöpiirin toimintaa. Tässä työssä selvitetään SISO- ja MIMO-säädön periaatteellisia eroja ja niiden käyttöä paperikoneen perälaatikkoprosessissa, jossa esiintyy vahvat ristikkäisvaikutukset. Työn tavoitteena on selvittää, kuinka MIMO-säädöllä voidaan ottaa huomioon prosessissa olevat ristikkäisvaikutukset ja näin parantaa säätöä.

Paperikoneen säädön täytyy olla tarkkaa, koska valmiin paperin on täytettävä sille asetetut laatuvaatimukset. Perälaatikko on paperikoneen alkupäässä ja sen tehtävänä on paperimassan tasainen ja oikealla nopeudella tapahtuva levittäminen paperikoneen viiralle. Paperikoneen perälaatikossa on kaksi säädettävää suuretta: paine ja pinnankorkeus. Näitä säätävillä ohjaussuureilla on voimakkaat ristikkäisvaikutukset toisiinsa, mikä vaikeuttaa perälaatikon tarkkaa säätöä.

SISO-säädön toiminta perustuu yhteen sisäänmenoon ja yhteen ulostuloon, kun taas MIMO-säädössä on useita sisäänmenoja ja useita ulostuloja. Monimuuttujaprosesseissa käytetään molempia säätömenetelmiä eri käyttökohteesta riippuen. SISO-säädössä, eli käytettäessä yksikkösäätimiä, säätimet eivät ole tietoisia toisten yksikkösäätimien toiminnasta, minkä takia säätimet kilpailevat keskenään ja huonontavat näin säätöä.

Yksikkösäätimiä käytettäessä tämä ongelma ratkaistaan niin, että tärkeämpi piiri viritetään halutuksi ja toissijainen piiri viritetään hitaaksi, jolloin sen toiminta vaikuttaa vähemmän tärkeämpään piiriin. Tällöin säätö ei voi olla lähelläkään ideaalista tilannetta. Yksikkösäätimiä käytettäessä monimuuttujaympäristöissä on myös huomioitava, miten eri sisäänmeno-ulostulo-parit on kytkettävä, jotta säätö olisi paras mahdollinen. MIMO-säätöä voidaan soveltaa monin eri tavoin, mutta sen yleisperiaatteena on säädin, joka kykenee hallitsemaan useita sisäänmeno-ulostulo-pareja. Tällöin säädin kykenee huomioimaan paremmin samanaikaisesti säädettävien suureiden vaikutukset toisiinsa kuin SISO-säädin. Tässä työssä selvitetään tarkemmin mallipohjaista monimuuttuja-PI-säädintä, jonka avulla ristikkäisvaikutuksia voidaan pienentää.

Monimuuttujasäädinrakenne paransi perälaatikkoa jäljittelevän prosessin säätöä, kun paineen säätö viritettiin nopeammaksi kuin pinnankorkeuden säätö. MIMO-säädin ei onnistunut parantamaan säätöä IAE-hyvyyskriteerin kannalta, kun pinnankorkeuden säätö viritettiin lähemmäksi paineen säädön nopeutta.

Avainsanat: MIMO, Monimuuttujaprosessi, monimuuttujasäätö, ristikkäisvaikutukset, SISO

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ALKUSANAT

Haluan kiittää työn ohjaajaa ja tarkastajaa Yrjö Majannetta työn ohjaamisesta. Suuri kiitos myös Jarille ja Mikolle järjestelmään liittyvien virhetilanteiden ratkaisemisesta. Lisäksi haluan kiittää Valmetia koulutusten tarjoamisesta liittyen heidän automaatiojärjestelmäänsä. Valmetilta erityiskiitokset ansaitsevat Arto ja Ville. Anttia haluan kiittää alkuvaiheen kokouksien järjestämisestä, jotta asiantuntevia henkilöitä löydettiin avustamaan työn kanssa.

Tampereella, 7.12.2021

Jimi Nisula

SISÄLLYSLUETTELO

1. JOHDANTO	1
2. TOIMINTAYMPÄRISTÖ	2
2.1 Perälaatikkoprosessi	2
2.2 Automaatiojärjestelmä	3
3. MENETELMÄT	5
3.1 Takaisinkytketty säätö	5
3.2 PID-säädin	5
3.3 Integraattorin windup-ilmio	7
3.4 Monimuuttujasäätö	8
3.5 IMC-menetelmä	9
4. SOVELLUS	12
4.1 Kohdeprosessiympäristö	12
4.2 Automaatiojärjestelmäympäristö	14
4.3 Viritykset	18
5. TULOKSET	19
6. YHTEENVETO	23
LÄHTEET	24
LIITTEET	25
Java-koodi	25

LYHENTEET JA MERKINNÄT

EAS	Engineering Activity Server
FbCAD	Function block CAD
IAE	Integral of the absolute error
IMC	Internal Model Control
MIMO	Multiple-Input-Multiple-Output
MPC	Model Predictive Control
SISO	Single-Input-Single-Output
TITO	Two-Input-Two-Output

1. JOHDANTO

Teollisuusprosessit ovat lähes poikkeuksetta monimuuttujaprosesseja, joissa on useita sisäänmenoja ja ulostuloja [1]. Yleisesti järjestelmää, jossa on useita säädettäviä suureita, kutsutaan monimuuttujaprosessiksi. Näissä prosesseissa jokin muutos tai häiriö, joka tapahtuu yhdessä säätöpiirissä, aiheuttaa myös muutoksen toisiin säätöpiireihin. Siten monimuuttujajärjestelmien säädössä esiintyy ristikkäisvaikutuksia, jotka häiritsevät prosessin säätämistä erillisillä yksikkösäätimillä. Ristikkäisvaikutusten merkittävyys riippuu yksikkösäätimien virityksestä. Mitä suurempia vahvistuksia käytetään, sitä suuremmat ovat ristikkäisvaikutukset. Ristikkäisvaikutuksia voidaan minimoida monin eri tavoin. Ei paras mahdollinen, mutta vaivaton keino on dekoplatattu säätö [2]. Parempia, mutta enemmän aikaa vieviä keinoja ovat esimerkiksi MPC (Model Predictive Control) [3] ja sumea logiikka -säätö [4].

Tampereen yliopiston Hervannan kampuksen systeemiteknikan laitoksella on paperikoneen lyhyttä kiertoa jäljentävä prosessi. Nykyisessä järjestelmässä kahta säädettävää suuretta säädetään erillisillä yksikkösäätimillä, jolloin yksikkösäätimet kilpailevat keskenään aiheuttaen ristikkäisvaikutuksia. Tämän opinnäytetyön tarkoituksena on tutkia monimuuttujaprosesseissa esiintyviä ristikkäisvaikutuksia ja suunnitella nämä ristikkäisvaikutukset huomioiva monimuuttuja-PI-säädin ValmetDNA-automaatiojärjestelmään. Todellisella prosessilla voidaan tehdä prosessikokeita, joiden avulla monimuuttuja-PI-säätörakenne voidaan validoida, kun automaatiojärjestelmän tallentamaa dataa verrataan yksikkösäädinrakenteella tuotettuun dataan.

Tässä työssä tullaan esittelemään toimintaympäristö, jossa esitellään monimuuttujaprosessin keskeiset ominaisuudet, perälaatikkoprosessi ja automaatiojärjestelmä. Seuraavaksi esitellään menetelmiä, jotka liittyvät monimuuttuja-aihealueeseen. Tämän jälkeen sovelletaan näitä menetelmiä kohdeprosessiympäristössä ja kerrotaan, kuinka tämä toteutetaan automaatiojärjestelmäympäristössä. Lopuksi esitellään tutkimuksen tulokset ja yhteenveto.

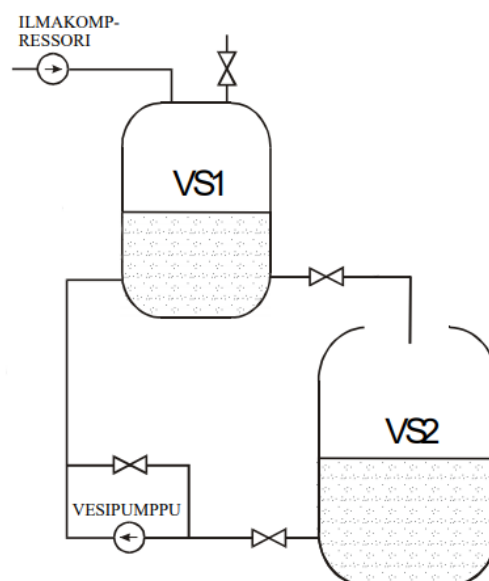
2. TOIMINTAYMPÄRISTÖ

Monimuuttujaprosessit ovat prosesseja, joissa on useita sisäänmenoja ja useita ulostuloja. Tyypillistä näille prosesseille on se, että kun muutetaan yhtä sisäänmenoa, niin se vaikuttaa useampaan kuin yhteen ulostuloon. Kun säätöpiirit kytketään suljetuiksi säätöpiireiksi, niin tuloksena on tyypillisesti värähtelevä vaste. Tämä tekee monimuuttujasäädöstä hankalaa, jolloin on tarkasteltava vaihtoehtoisia tapoja säätää järjestelmää. Yksinkertaisin monimuuttujaprosessi koostuu kahdesta sisäänmenosta ja kahdesta ulostulosta. Tällaista prosessia kutsutaan TITO-prosessiksi (Two-Input-Two-Output).

Isossa järjestelmässä, joissa on useita säädettäviä suureita, toimilaitteita ja säätimiä, on paljon säätösilmukoita. Järjestelmää voidaan hallita käsittelemällä yksittäisiä säätösilmukoita, mutta toisinaan yhdellä säätösilmukalla on vaikutus myös toiseen silmukkaan. Tätä kutsutaan ristikkäisvaikutukseksi. Säätöjärjestelmiä, joissa esiintyy ristikkäisvaikutuksia ovat esimerkiksi höyrykattilat, paperikoneet ja kemialliset prosessit [1]. Ristikkäisvaikutuksien takia järjestelmää on vaikea hallita säätämällä yksittäisiä silmukoita, koska yhden suureen muuttaminen vaikuttaa muihin suureisiin ja ne edelleen vaikuttavat toisiin suureisiin "luoppimaisesti".

2.1 Perälaatikkoprosessi

Tässä kandidaatintyössä prosessina on paperikoneen lyhyttä kiertoa jäljentävä prosessi.



Kuva 1 – Havainnollistava kuva perälaatikkoprosessista [5]

Paperikoneen lyhyt kierto muodostuu perälaatikosta VS1 ja viirakaivosta VS2. Lyhyen kierron toimintaperiaate on seuraava: Perälaatikosta syötetään paperimassaa tasaisesti viiralle. Paperimassasta viiralla poistunut vesi ohjataan viirakaivoon, josta se pumpataan takaisin perälaatikkoon. Perälaatikossa säädettävänä suureina ovat paine ja pinnankorkeus ja ohjauksina ovat peränsyöttöpumpun ja paineilmakompressorin pyörimisnopeudet. Koska perälaatikossa on kaksi sisäänmenoa ja kaksi ulostuloa, niin kyseessä on monimuuttujaprosessi. Perälaatikkoprosessissa on vahvat ristikkäisvaikutukset säädettävien suureiden välillä.

2.2 Automaatiojärjestelmä

ValmetDNA on hajautettu automaatiojärjestelmä (eng. DCS). Säättöpiirit ovat tällöin hajautettu useammalle mikroprosessipohjaiselle prosessiasemalle. ValmetDNA:ssa yhdistetään sertifioitu tietoturva, tiedonhallinta, suunnittelu, huolto, prosessinohjaus ja I/O yhden järjestelmän sisään. [6] DNA on monipuolinen järjestelmä, jossa on paljon erilaisia työkaluja eri tarkoituksiin. Tässä työssä tullaan käsittelemään vain työn kannalta oleelliset sovellukset.

Yksi keskeisimmistä työkaluista on Function block CAD (FbCAD), jonka avulla järjestelmään tehdään sovelluksia, esimerkiksi ohjauspiirejä, jotka ovat samalla graafisia dokumentteja. Ohjauspiirit tehdään yleensä eri sisäänmeno ja ulostulo yhdistelmien välille sisältäen jonkinlaista laskentaa ja logiikkaa. FbCAD:ssä on laaja joukko erilaisia lohkoja käytettäväksi sovelluksiin. Näitä ovat esimerkiksi Fuzzy- ja MPC-säätölohkot sekä vapaasti ohjelmoitava Java-lohko.

DNA Explorer on suunnittelu ja ylläpitotyökalu, joka tarjoaa visuaalisen esityksen DNA:n tiedostojärjestelmästä [7]. Explorerin avulla voidaan tarkastella suunnittelupalvelimen (EAS) ja tiettyjen prosessien sisältämiä ohjauspiirejä, valvomonäyttöjä ja laitekonfiguraatioita.

DNA Operate on prosessin operoimiseen tarkoitettu operointityökalu, jossa on graafinen navigoitavuus ja selkeä toimintaperiaate [7]. Tuplaklikkaamalla valittuja lohkoja saadaan auki laitteen toimintaan liittyvää tietoa ja hiiren oikealla klikkauksella voidaan tutkia tarkemmin lohkon toimintaa esimerkiksi mittaustrendien perusteella.

Picture designer on työkalu, jonka avulla luodaan graafinen käyttöliittymä sovellukselle. Sen avulla graafiset elementit voidaan yhdistää säättöpiiriin TAG-koodeilla merkittyihin muuttujiin.

Valmet DNA Report on raportointiin ja analyysin tarkoitettu verkkopohjainen työkalu, jonka avulla voidaan muun muassa tarkastella järjestelmän trendejä ja viedä ne esimerkiksi Excel-tiedostoon.

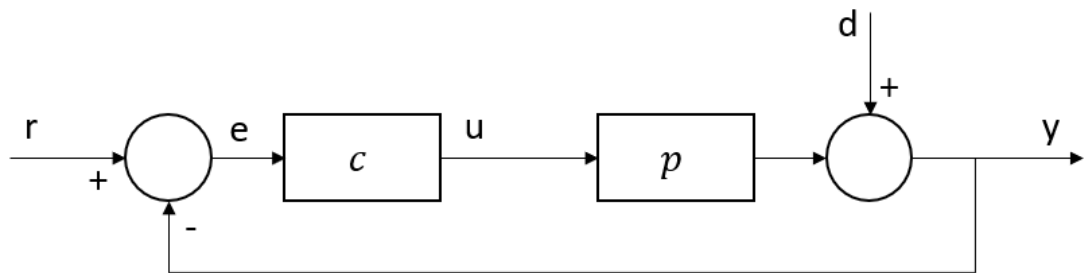
Tyypillinen sovellusentekoprosessi on yksinkertaistetusti seuraava: FbCAD-ohjelmalla tehdään säätöpiiri. Picture Designer työkalulla kytketään säätöpiirin jotakin toiminnallisuutta operointikuvaan. Seuraavaksi ohjauspiiri ja operointikuva ladataan järjestelmään DNA Explorerin avulla. Tämän jälkeen säätöpiiriä voidaan hyödyntää järjestelmässä operointikuvasta DNA Operaten kautta.

3. MENETELMÄT

Tässä luvussa tullaan esittelemään yleisiä menetelmiä, jotka liittyvät läheisesti tähän työhön.

3.1 Takaisinkytketty säätö

Yksi keskeisimmistä säädön periaatteista on takaisinkytketty säätö.



Kuva 2 – Takaisinkytketty järjestelmä [8].

Tavallinen negatiivisesti takaisinkytketty järjestelmä on kuvan 2 mukainen. Siinä r on asetuservo, e on erosuure, u on ohjaus, d on häiriö, y on ulostulo, c on säädin ja p on prosessi. Säätörakenteen tarkoituksena on pitää ulostulo mahdollisimman lähellä asetuservoa seuraamalla mitattua ulostuloa ja vertaamalla sitä asetuservoon.

3.2 PID-säädin

Proportional-Integral-Derivate-säädin eli PID-säädin on teollisuudessa yleisesti käytetty säädin, jolla säädetään takaisinkytkettyä järjestelmää. PID-säädin koostuu kolmesta osasta: suhdeosasta, integrointiosasta ja derivointiosasta. PID-säädin käyttää toiminnassaan menneisyyden, nykyhetken ja tulevaisuuden ennustettua säätövirhettä [8].

Suhdeosalla tarkoitetaan säätimen toimintoa, jossa säätimen ulostulo reagoi suoraan verrannollisesti säätimeen sisäänmenoon, joka on yleensä erosuure e . Erosuure on asetuservon ja mittauservon erotus, mikä on siten poikkeama asetetusta arvosta. P-osan vahvistusermi K_p säätimen parametreissa kuvastaa, kuinka voimakkaasti säätö reagoi säätimeen sisäänmenoon. P-osa voidaan esittää seuraavasti:

$$u_p(t) = K_p e(\tau). \quad (3.1)$$

Mitä suurempi termi K_p on, sitä suurempi on sen vaikutus säätöön. P-osa ei kykene usein saavuttamaan asetusrvoa, jolloin säätöpiirissä on tasapainotilan virhettä [8]. Tämän takia säätimessä on mukana integrointiosa, joka pystyy ratkaisemaan tämän ongelman. Integrointiosa on säätimen toiminto, joka integroi erosuuretta. I-osan ulostulo on suhteessa erosuuruuden suuruuteen P-osan tavoin, mutta se myös ottaa huomioon erosuureen kestoajan. I-osan parametri on integrointiaika T_i , jonka avulla määritetään integroinnin kesto. [8] I-osa voidaan esittää muodossa:

$$u_i(t) = \frac{K_p}{T_i} \int_0^t e(\tau) d\tau. \quad (3.2)$$

Kaavan (3.2) perusteella integrointiajan pienentäminen vahvistaa I-osan merkitystä säätöön, jolloin liiallinen pienennys voi aiheuttaa värähtelevän vasteen. Toisaalta, jos T_i asetetaan äärettömäksi, niin PI-säätimen tapauksessa ohjaus perustuu vain P-osaan.

Jos järjestelmässä on tasapainotila, ideaalitapauksessa integrointiosan takia säätöpiiri saavuttaa aina tasapainotilan ja tasapainotilassa lopullinen säätövirhe on siinä nolla. P-osan nopean reagoinnin erosuureeseen ja I-osan säätövirheen pienenemisen takia näistä kahdesta osasta muodostuva PI-säädin on suosittu säätöratkaisu. [8]

Derivointiosa derivoi erosuuretta eli se reagoi erosuuruuden muutosnopeuteen. D-osa esitetään seuraavasti:

$$u_d(t) = K_p T_d \dot{e}(t), \quad (3.3)$$

jossa uusi parametri on derivointiaika T_d , jonka kautta voidaan määrittää derivoinnin kesto. Kaavasta (3.3) nähdään, että PID-säätimessä D-osa saadaan kytkettyä pois asettamalla derivointiaika nolllaksi.

D-osan tarkoitus on tuoda säätöpiiriin lisää stabiiliutta. Ilman D-osaa prosessin dynamiikan takia perinteinen PI-säädin ei ehdi reagoimaan tarpeeksi nopeasti säätövirheeseen, jolloin järjestelmä on myöhässä ohjausten kanssa. Derivointiosan toiminta perustuu ennustamiseen ekstrapoloimalla virhettä erosuureen tangentin kulmakertoimen avulla. [8]

Säätimen parametrejä ovat edellisten kappaleiden perusteella vahvistus K_p , integrointiaika T_i ja derivointiaika T_d . Näistä parametreistä voidaan muodostaa PID-säätimelle matemaattinen esitys

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \dot{e}(t) \right), \quad (3.4)$$

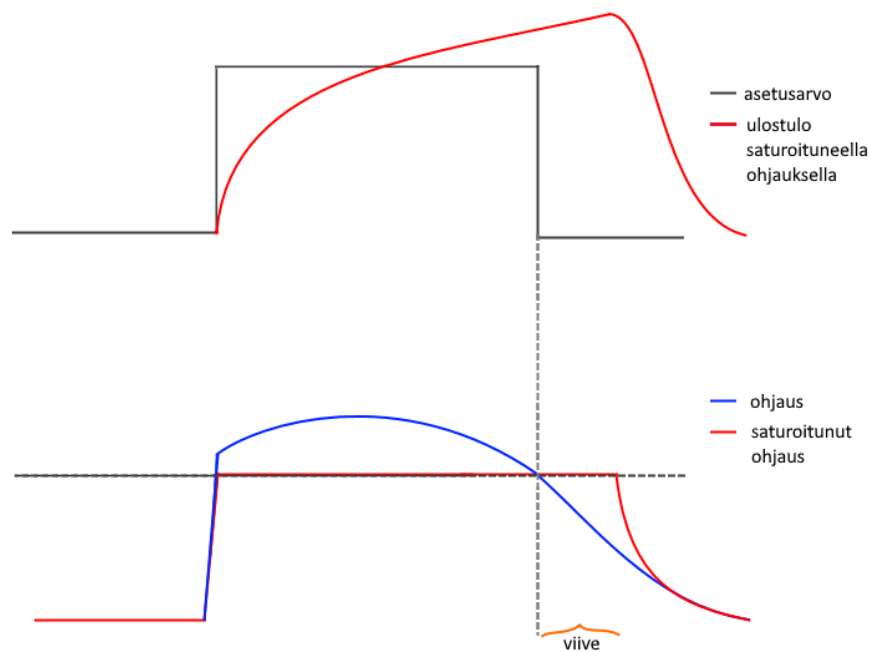
jossa u on säätimen ulostulo eli ohjaus, e on erosuure ja \dot{e} on erosuureen derivaatta. Siirtofunktiona esitettyä kaava (3.4) saadaan muotoon

$$u = \left(1 + \frac{1}{T_i s} + T_d s \right) K_p e. \quad (3.5)$$

PID-säätimessä ei aina hyödynnetä kaikkia näitä osia, koska ne eivät ole aina tarpeellisia tai ne voivat häiritä säädettävää prosessia [8].

3.3 Integraattorin windup-ilmio

Kaikki säätöjärjestelmän osat eivät ole pelkästään lineaarisia, vaan on huomioitava myös epälineaariset vaikutukset säätöjärjestelmään. Toimilaitteilla on äärelliset liikealueet: moottorilla on rajattu kierrosnopeus, venttiili ei voi avautua enää enempää, kun se on täysin auki, pumppu pystyy siirtämään vain tietyn tilavuusvirran ja niin edelleen. [8] Joskus säätöjärjestelmän ohjaus voi päätyä tilanteeseen, jossa säätöjärjestelmä vaatii toimilaitteelta enemmän kuin se pystyy suorittamaan. Jos säätimessä on käytössä integrointi, niin se integroi erosuuretta virheellisesti aiheuttaen häiriötä säätöjärjestelmän toimintaan. Windup-ilmiotä tapahtuu usein, kun järjestelmään tehdään suuria asetusarvomutoksia joko itse aiheuttaen tai isojen häiriöiden sekä laiterikkojen takia [8]. Windup-ilmiotä esiintyy myös silloin, kun toimilaitteet satureituvat äärirajoilleen tavallisessa käytössä.



Kuva 3 – Esimerkki windupin aiheuttamasta häiriöstä säätöön

Kuvassa 3 on esitetty yksi mahdollinen skenenaario tapauksesta, kun järjestelmän säätö häiriintyy windup-ilmiöstä. Järjestelmässä, jossa ohjaus on 100 % ja mittaus on asetusarvon alapuolella, on positiivista erosuuretta. Kun tilanne jatkuu pidemmän aikaa, niin järjestelmä integroi erosuuretta, jolloin ohjaukseen kerääntyy virheellistä ohjausta. Kun järjestelmän pitäisi pudottaa ohjausta 100 %:sta alaspäin eli erosuure on negatiivinen, järjestelmän ohjaus pysyykin 100 %:ssa. Jotta tilanteesta päästäisiin eroon, erimerkkisen erosuuren täytyy olla säädön kannalta pitkän aikaa voimassa, kunnes ohjaus on taas toimilaitteen rajojen sisällä. Kuvasta 3 nähdään, että järjestelmä ei kykene reagoimaan asetusarvon pudottamiseen, minkä seurauksena ulostulo jatkaa nousuaan.

Windup-ilmiötä vastaan on kehitetty monia menetelmiä. Näitä ovat esimerkiksi: Set-Point Limitation, Back-Calculation ja integrator clamping. Yksi tapa on pitää integraattorin arvoa vakiona, jos ohjaus on saturoitunut eli se on ylittänyt toimilaitteen rajat. Parempi tapa on kuitenkin, että integrointi lopetetaan, jos ohjaus on saturoitunut ja seuraava integraattorin arvon päivitys veisi ohjausta vielä kauemmas toimilaitteen rajoista. [8]

3.4 Monimuuttujasäätö

Monimuuttujatapauksissa on oleellista tietää, mitä ulostuloa ohjaa mikäkin sisäänmeno. Prosessin väliset vuorovaikutussuhteet on selvitettävä, jotta tiedetään, mitkä ovat optimaaliset sisäänmeno-ulostulo-parit. Tätä varten on kehitetty RGA-menetelmä, jonka avulla näitä vuorovaikutuksia voidaan arvioida. Ristikkäisvaikutuksia voidaan tutkia tarkastelemalla yhden silmukan staattisista vahvistusta ja sen aikaansaamaa vaikutusta muihin silmukoihin [8]. RGA-menetelmän avulla voidaan selvittää myös minkälainen säätöratkaisu voisi olla kelvallinen tietyissä tilanteissa. Jos ristikkäisvaikutukset ovat pienet, monimuuttujaprosessin säädöstä voidaan selvittää SISO-säädöllä eli jokaiselle säädettävälle suurelle on oma säätimensä. Jos ristikkäisvaikutukset ovat suuret, SISO-säädöllä (Single-Input-Single-Output) ei enää pärjää, jolloin on keksittävä muita tapoja.

Tällöin yksi vaihtoehto on ottaa käyttöön monimuuttujasäätöön kehitetyjä menetelmiä. Monimuuttujasäädön yleisperiaate on kytkeä säätimen rakenteeseen rakenteita, jotka pyrkivät eliminoimaan ristikkäisvaikutukset. Tämä on mahdollista, koska MIMO-säädin (Multiple-Input-Multiple-Output) kykenee samanaikaisesti ohjaamaan useaa sisäänmenoa. MIMO-säätimen algoritmeissa voidaan kytkeä eri sisäänmenoja niin, että ristikkäisvaikutukset pienenevät.

Yksinkertaisin monimuuttujaprosessi koostuu kahdesta sisäänmenosta ja kahdesta ulostulosta. Järjestelmä voidaan esittää siirtofunktioyhtälöillä:

$$Y_1(s) = p_{11}(s)U_1(s) + p_{12}U_2(s) \quad (3.6)$$

$$Y_2(s) = p_{21}(s)U_1(s) + p_{22}U_2(s), \quad (3.7)$$

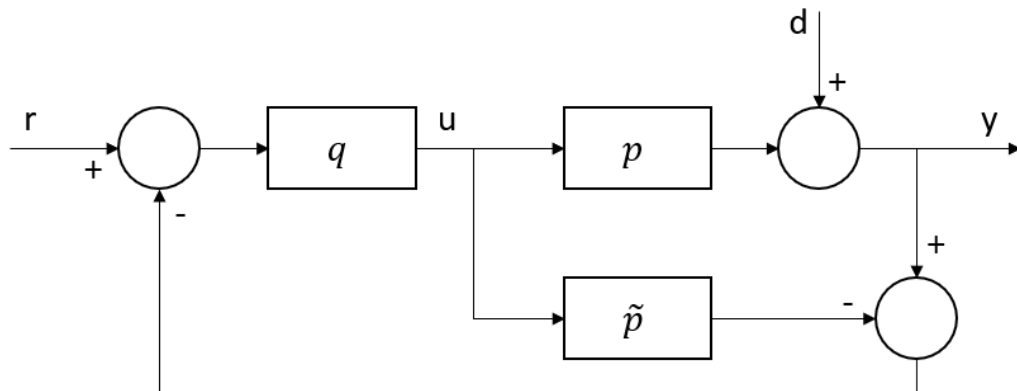
jossa $p_{nm}(s)$ on siirtofunktio m:nnestä sisäänmenosta n:nteen ulostuloon. Siirtofunktiot voidaan koota yhteen matriisiin, jota kutsutaan järjestelmän siirtofunktioksi.

$$P(s) = \begin{bmatrix} p_{11}(s) & p_{12}(s) \\ p_{21}(s) & p_{22}(s) \end{bmatrix} \quad (3.8)$$

Siinä $P_{nm}(s)$ edustaa yksittäistä siirtofunktiota m:nnestä sisäänmenosta n:nteen ulostuloon.

3.5 IMC-menetelmä

Internal Model Control -menetelmä on yleinen teollisuusprosesseissa käytetty menetelmä, jonka tavoitteena on tuottaa säätimen rakenne ja viritysparametrit PID-tyyppistä säädintä varten. IMC-menetelmän nimitys on peräisin siitä, minkälainen rakenne muodostuu säätimelle. Säätimen rakenteessa on mukana prosessin mallin rinnan kytkettynä oikean prosessin kanssa. [8] Yksi IMC-menetelmän vahvuuksista on sen takaama stabiilius takaisinkytketylle järjestelmälle: valitsemalla stabiili IMC-säädin. Suljetun järjestelmän suorituskykyä voidaan muuttaa yksinkertaisesti muuttamalla säätimen parametrejä [1].



Kuva 4 – IMC-rakenne [8]

Kuvassa 4 on esitettyä yksinkertainen IMC-rakenne, jossa on lähes sama rakenne kuin edellä esitetyssä takaisinkytketyssä rakenteessa. Ainoa ero on siinä, että IMC-rakenteessa on mukana prosessin malli \tilde{p} ja säädin on nyt q .

Tavallisen takaisinkytketyn järjestelmän ja IMC-rakenteen välillä on yhteys [1]:

$$c = \frac{q}{1 - \tilde{p}q} \quad (3.9)$$

$$q = \frac{c}{1 + \tilde{p}c}. \quad (3.10)$$

IMC-menetelmän tuottamat viritysparametrit eivät ole suoraan käytettävissä säätimessä, vaan niitä joudutaan muuttamaan. IMC:llä tehty säätäjän siirtofunktio on muotoa

$$u = \left(K_p + K_I \frac{1}{s} + K_D s \right) e, \quad (3.11)$$

jossa K_p on P-osan vahvistus, K_I on I-osan vahvistus ja K_D on D-osan vahvistus. Verrataan tätä siirtofunktio mallia kaavaan (3.5), jolloin havaitaan yhteys näiden siirtofunktio mallien välillä. IMC-parametrit voidaan muuntaa takaisin standardi PID-parametreiksi, jolloin

$$K_P = K_p, T_i = \frac{K_P}{K_I} \text{ ja } T_D = \frac{K_D}{K_P}. \quad (3.12)$$

Nyt IMC-menetelmällä saadut viritysparametrit voidaan ottaa käyttöön tavallisessa PID-säätimessä.

IMC-menetelmässä säädintä voidaan vielä erikseen virittää muuttamalla säätimessä käytetyn alipäästösuodattimen f aikavakiota λ . λ on kääntäen verrannollinen säätäjän vahvistukseen, joten mitä pienempi se on, sitä nopeampi säädin on. Luonnollisesti virituksen takia säätöpiirin stabiilius on huomioitava.

MIMO-säädin suunnitellaan, niin että ensiksi tehdään säätimet jokaisella yksittäiselle sisäänmeno-ulostulo-silmukalle. Tämän jälkeen säätimet voidaan muuntaa MIMO-yhteensopivaksi säätimeksi seuraavalla kaavalla: [9]

$$C(s) = \begin{bmatrix} \frac{1}{c_{11}(s)} & \frac{1}{c_{12}(s)} \\ \frac{1}{c_{21}(s)} & \frac{1}{c_{22}(s)} \end{bmatrix}^{-1}. \quad (3.13)$$

Monimuuttuja PI-säätimen tapauksessa säädin on $C(s) = K_p + K_i s^{-1}$ [3], jossa

$$K_p = \begin{bmatrix} \frac{1}{K_{p11}(s)} & \frac{1}{K_{p12}(s)} \\ \frac{1}{K_{p21}(s)} & \frac{1}{K_{p22}(s)} \end{bmatrix}^{-1} \text{ ja} \quad (3.14)$$

$$K_i = \begin{bmatrix} \frac{1}{K_{i11}(s)} & \frac{1}{K_{i12}(s)} \\ \frac{1}{K_{i21}(s)} & \frac{1}{K_{i22}(s)} \end{bmatrix}^{-1}. \quad (3.15)$$

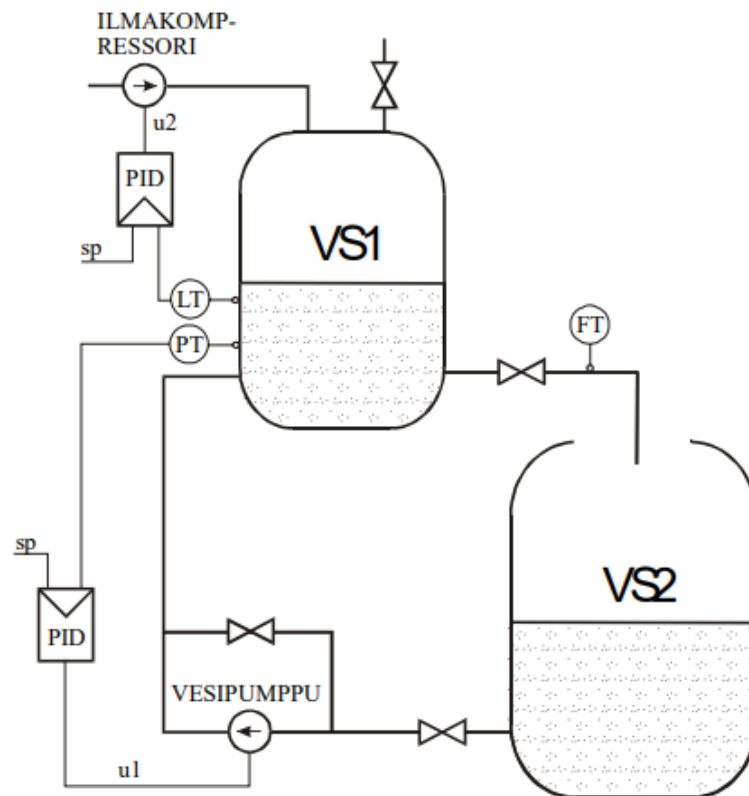
Näitä vahvistuksien arvoja voidaan käyttää suoraan monimuuttujasäätimessä.

4. SOVELLUS

Tässä kappaleessa käsitellään, miten teoria yhdistetään kohdeprosessiin ja miten se käytännössä implementoidaan automaatiojärjestelmään.

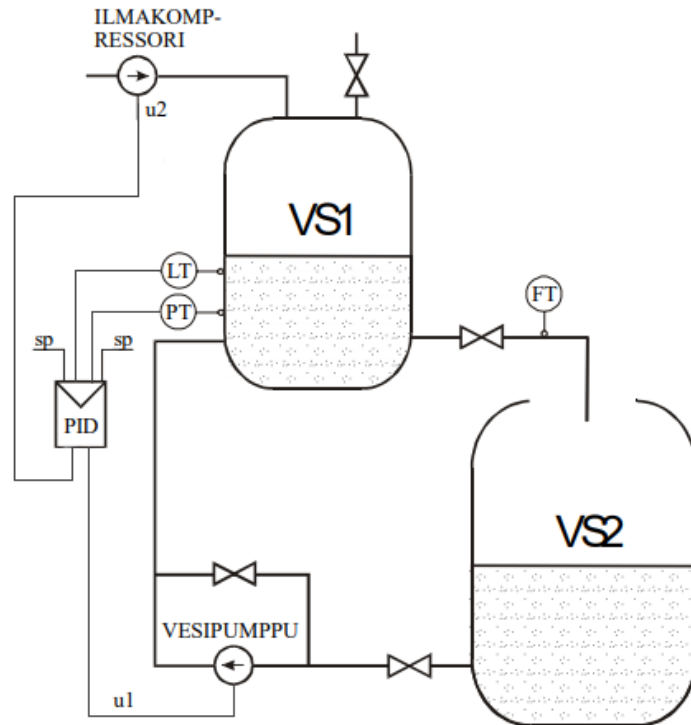
4.1 Kohdeprosessiympäristö

Nykyisessä järjestelmässä on käytössä yksikkösäätimet, joilla voidaan säätää järjestelmää kahdella eri konfiguraatiolla. Vesipumpulla voidaan säätää joko painetta tai pinnankorkeutta ja kompressorilla joko pinnankorkeutta tai painetta.



Kuva 5 – Perälaatikon SISO-säätö

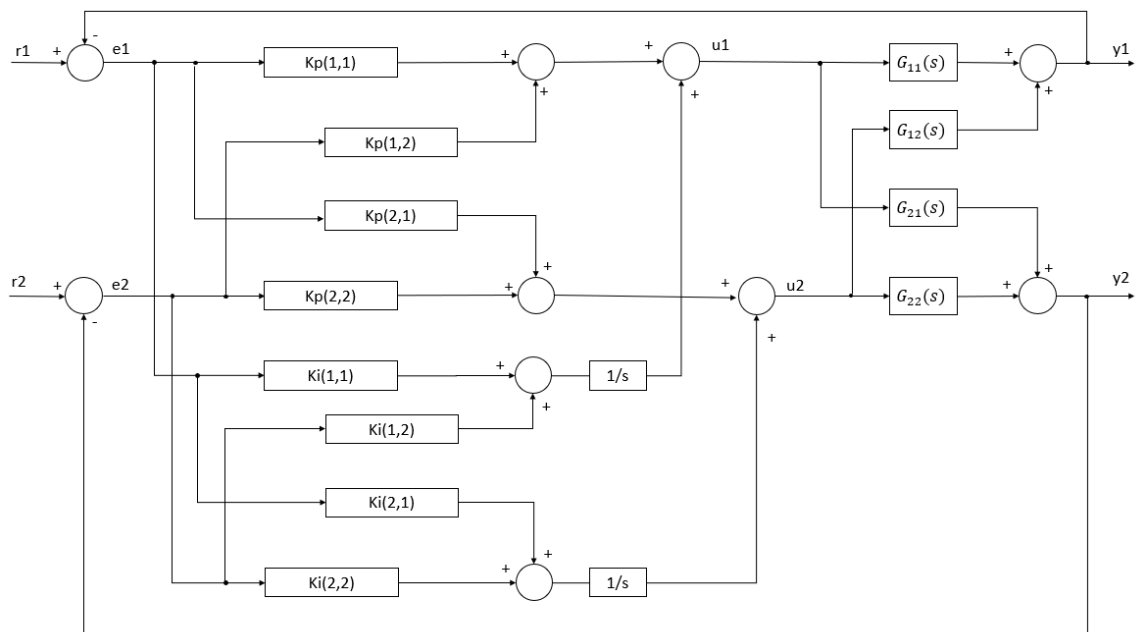
Kuvassa 5 on esitettyä Perälaatikon SISO-säätö konfiguraationa, jossa painetta säädetään pumpulla ja pinnankorkeutta kompressorilla.



Kuva 6 – Perälaatikon MIMO-säätö

Kuvassa 6 on esitettyä perälaatikon MIMO-säätö. Sisäänmenoina ovat pinnankorkeuden mittausta ja paineen mittausta, ja näille molemmille on omat asetusarvonsa. MIMO-säätimen säätöalgoritmeissa lasketaan ohjaukset u_1 ja u_2 .

Monimuuttajasäätöjärjestelmän ideana on huomioida prosessissa esiintyvät ristikkäisvaikutukset. Järjestelmään tehdään seuraava MIMO-säätöalgoritmi.



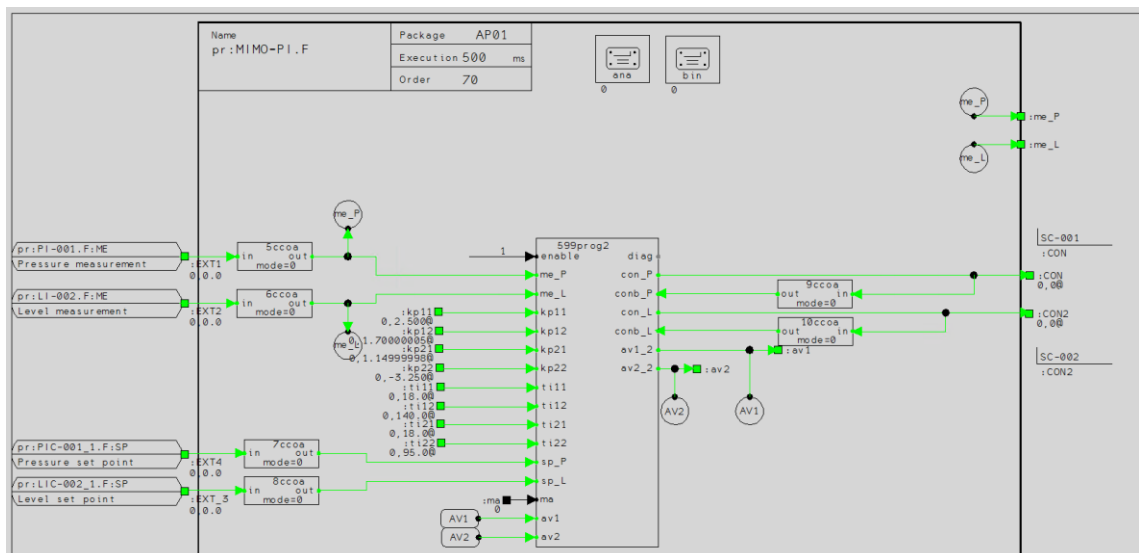
Kuva 7 – Monimuuttajasäätöjärjestelmä

Kuvasta 7 nähdään algoritmiin tuleva ristikytkentä molempien ohjauksien ja erosuureiden välillä. Ristiin kytkennän tarkoituksena on kompensoida prosessin ulostulojen y ja ohjauksien u välillä esiintyvää ristikkäisvaikutusta, joka syntyy prosessin siirtofunktioista G_{12} ja G_{21} . Esimerkiksi ylemmän paineen ohjauksen u_1 P-osa muodostuu suorasta ohjauksesta $Kp_{11} \cdot e_1$ sekä epäsuorasta ohjauksesta $Kp_{12} \cdot e_2$. Samaan tapaan ohjauksen u_1 I-osan ohjauksen päivitys muodostuu suorasta ohjauksesta $Ki_{11} \cdot e_1$ ja epäsuorasta ohjauksesta $Ki_{12} \cdot e_2$.

4.2 Automaatiojärjestelmäympäristö

ValmetDNA:ssa ei ole valmista toimilohkoa monimuuttujasäätimen toteuttamiseen, joten toimilohko tehdään vapaasti ohjelmoitavalla Prog2-toimilohkolla. Prog2-toimilohko on toimilohko, joka suorittaa Java-ohjelmointikielellä kirjoitettua koodia. Prog2-toimilohkoon ohjelmoidaan säätöalgoritmi ja säätimen toiminta.

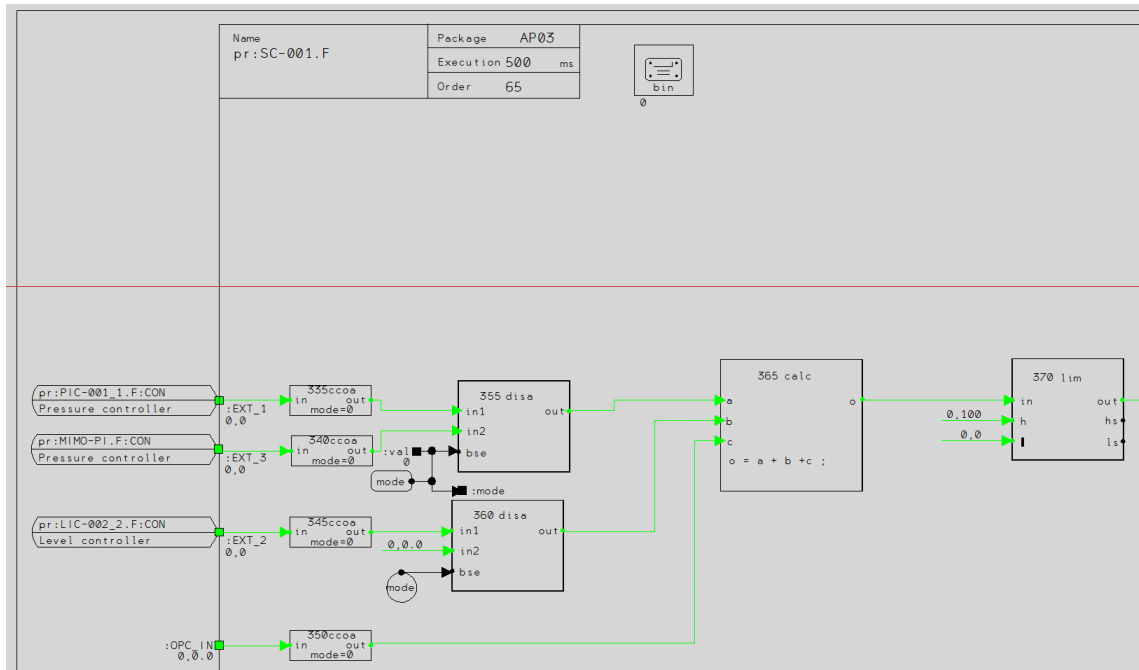
Seuraavaksi tarkastellaan säätöpiirin toimintaperiaatetta. Ensiksi järjestelmästä luetaan mittauksista paine ja pinnankorkeus. Sen jälkeen mittaukset menevät prog2-lohkolle, jossa tapahtuu säätimen toiminta. Prog2-lohkoon tulevat tämän lisäksi viritysarvot, asetusarvot, binäärinen ma (manuaali/automaatti) ja ohjauksien arvot. Asetusarvojen asettaminen tapahtuu piirin alaosassa näkyvillä lohkoilla. Prog2-lohkon merkitykselliset ulostulot ovat pelkästään ohjaukset, jotka menevät toimilaitteiden ohjauspiireille luettavaksi.



Kuva 8 – MIMO-säätöpiiri

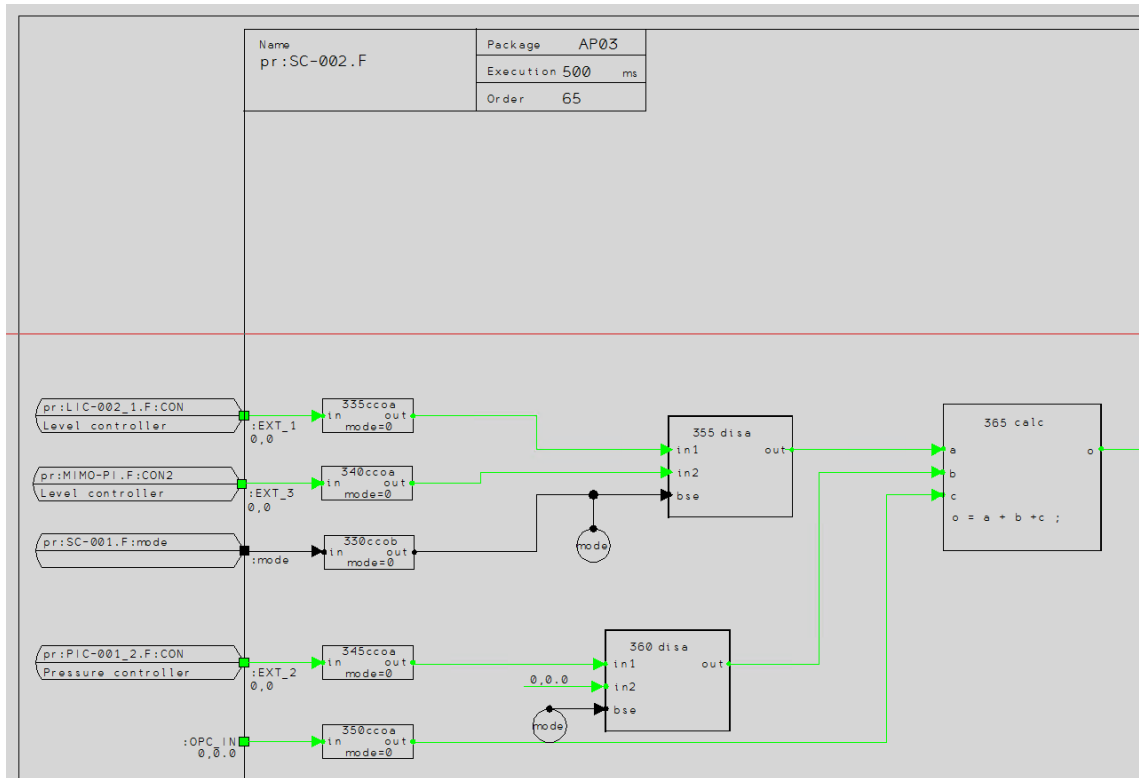
Kuvasta 8 nähdään, että järjestelmästä luetaan mittauksista paine ja pinnankorkeus. Sen jälkeen ne menevät kuvan 8 keskellä olevalle prog2-lohkolle, jossa tapahtuu säätimen

toiminta. Säädin laskee ohjaukset pumpulle ja kompressorille. Nämä asetetaan luettavaksi pumpun ja kompressorin ohjauspiireille.



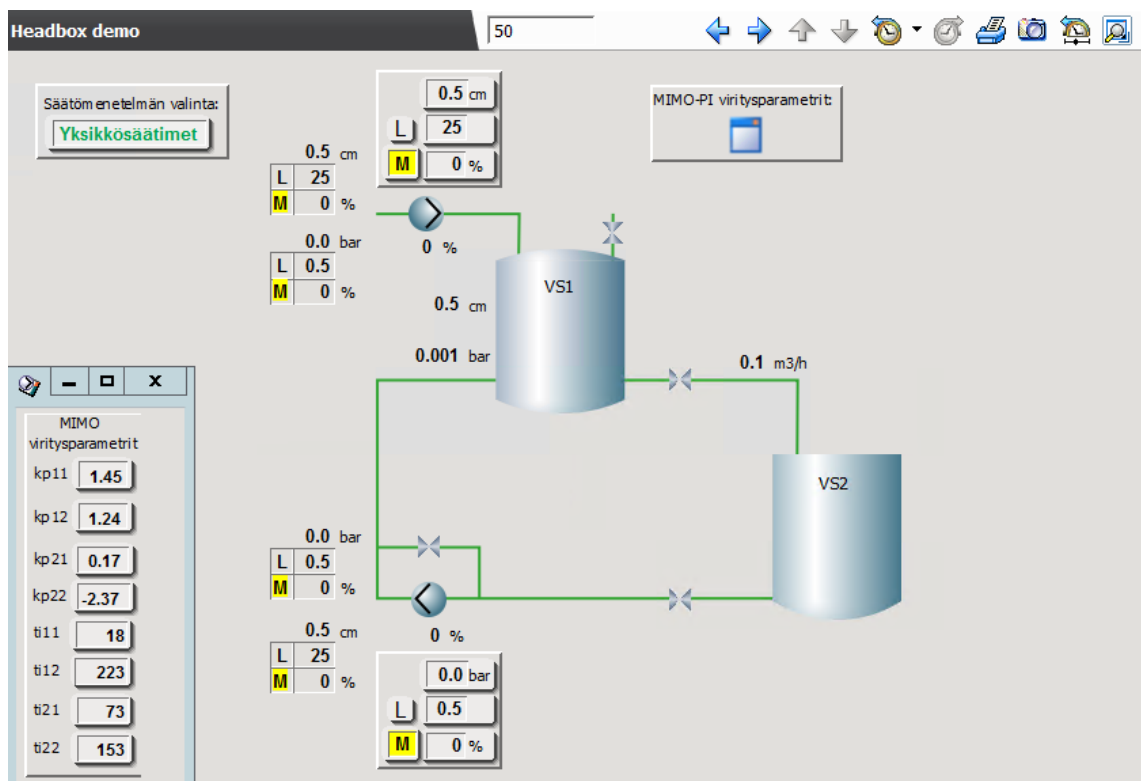
Kuva 9 – Vesipumpun pyörimisnopeuden ohjauspiiri

Kuvasta 9 voidaan havaita, että pumpua ohjataan samalla ohjauspiirillä huolimatta säätötavasta. Tämä tarkoittaa, että samalla ohjauspiirillä luetaan sekä SISO-säätimestä, että MIMO-säätimestä tulevia ohjauksien arvoja. disa-lohkoilla suoritetaan valinta, ohjataanko pumpua SISO- vai MIMO-säädöllä. calc-lohkolla lasketaan ohjauksien summa ja lim-lohkolla rajoitetaan ohjaus välille (0,100).



Kuva 10 – Ilmakompressorin pyörimisnopeuden ohjauspiiri

Kuvasta 10 nähdään, että ilmakompressorin ohjauspiiri on toiminnaltaan samankaltainen kuin vesipumpun ohjauspiiri.



Kuva 11 – Ohjausnäköymä-ikkuna

Kuvassa 11 esiintyvään, järjestelmässä olleeseen SISO-säätimillä toteutettuun kuvaan lisättiin dialogi säätömenetelmän valinnalle, MIMO-säätimen laatikot ja –viritysparameetri-ikkuna.

Java-koodi on nähtävillä tämän kandidaatintyön liitteissä. Java-koodin runko on tehty generoimalla pohja koodille VarmetDNA:n ohjeissa esitettyllä tavalla. Liitteen koodista on poistettu rakentaja ja käyttämättömät funktiot koodin selkeyttämiseksi. Java-koodin runkoon on tehty kaksi ehto rakennetta sen mukaan, käytetäänkö säädintä manuaalisesti vai automaattiohjauksella. Java-lohkon sisääntulojen lukemiseen ja ulostulojen asettamiseen käytetään DNA-muuttujien metodeja sisääntulo.`[get/set][tyyppi]Value([arvo])`. Esimerkiksi paineen mittauksen Double arvon lukemiseen voidaan käyttää seuraavaa kutsua `me_P.getDoubleValue()`. Ulostulojen asettaminen tapahtuu vastaavasti eli esimerkiksi paineen ohjauksen asettaminen onnistuu komennolla `con_P.setDoubleValue(u_P)`, jossa `u_P` on laskettu paineen ohjauksen arvo.

Integraattorien anti-windup on tehty seuraavasti: 1. paineen ja pinnankorkeuden ohjaukset tarkastetaan positiivisen ja negatiivisen saturaation osalta. 2. Jos saturaatiota on, ohjaus sidotaan joko ylärajalle tai alarajalle. 3. Tarkastetaan, onko ero suure saman merkinen kuin saturaatio. 4. Jos on, niin integraattorien arvoja ei päivitetä, jolloin integraattorin arvo pysyy samana seuraavallakin koodikierröksellä. 5. Jos ei ole, integraattorin arvot päivitetään.

Manuaaliohjauksen rakenne on yksinkertainen: Jos ohjauksien arvoja muutetaan, niin Java-lohko asettaa uudet ohjaukset järjestelmään.

4.3 Viritykset

IMC-menetelmällä saadut viritykset saatiin käyttämällä λ_1 arvoa 10 ja λ_2 arvoa 80, kun painetta säädetään pumpulla ja pintaa kompressorilla. Lambdan arvot havaittiin hyviksi simuloinneilla ja prosessin testaamisella. Kun, painetta säädettiin kompressorilla ja pintaa pumpulla, niin lambda λ_1 oli 10 ja λ_2 oli 20. Näillä lambdan arvoilla P-osan vahvistukseksi ja I-osan integrointiajaksi saatiin

$$K_{p_siso} = \begin{bmatrix} 1,54 & 2,94 \\ 22,11 & -2,52 \end{bmatrix} \text{ ja} \quad (4.1)$$

$$T_{i_siso} = \begin{bmatrix} 19 & 60 \\ 13 & 60 \end{bmatrix}. \quad (4.2)$$

Kun nämä vahvistukset muunnettiin MIMO-säätimelle sopiviksi yhtälöiden 3.14 ja 3.15 avulla, saatiin MIMO-säätimen vahvistukset ja integrointiajat

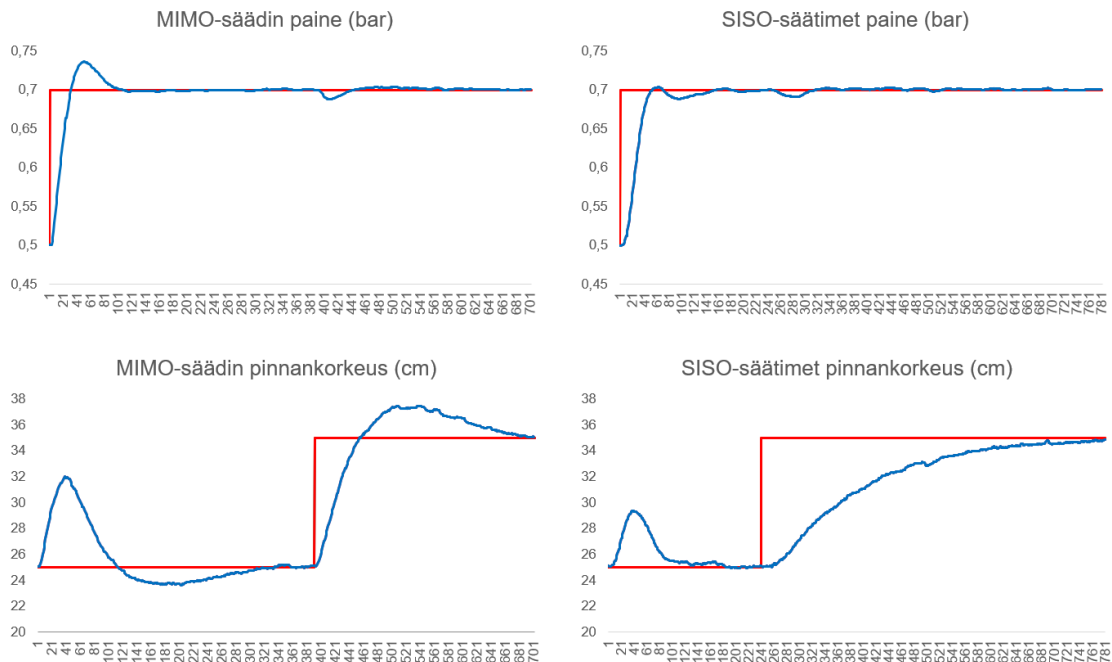
$$K_{p_mimo} = \begin{bmatrix} 1,45 & 1,24 \\ 0,17 & -2,37 \end{bmatrix} \text{ ja} \quad (4.3)$$

$$T_{i_mimo} = \begin{bmatrix} 18 & 223 \\ 72 & 153 \end{bmatrix}. \quad (4.4)$$

Kun SISO-säätimien ja MIMO-säätimen vahvistuksia vertaillaan, niin huomionarvoista on, että diagonaalielementit pysyvät lähes samoina.

5. TULOKSET

Säädön hyvyystarkastelu tehdään vertailemalla yksikkösäätimillä ja MIMO-säätimellä tehtyjä prosessikokeiden vasteita. Prosessikokeet tehdään seuraavalla tavalla: Prosessi ajetaan ylös 25 senttimetrin pinnankorkeuteen ja 0,5 barin paineeseen. Seuraavaksi paineen asetusarvo asetetaan 0,7 bariin, jolloin säädin ajaa prosessin 0,7 bariin, ja samalla yrittää pitää pinnankorkeuden 25 senttimetrissä. Kun prosessi on asettanut uuteen tilaan, jossa pinnankorkeus on vanhassa ja paine on uudessa asetusarvossaan, asetetaan pinnankorkeuteen uusi asetusarvo 35 senttimetriin. Nyt vastaavasti säädin ajaa pinnankorkeuden 35 senttimetriin ja samalla yrittää pitää paineen 0,7 barissa. Tämä koe tehdään sekä yksikkösäätimillä että MIMO-säätimellä.



Kuva 12 – Säätötulokset prosessikokeista

Kuvassa 12 on esitettyä sekä MIMO-säätimellä että SISO-säätimellä suoritettujen prosessikokeiden tulokset. SISO-säätimien tulokset ovat konfiguraatiosta, jossa painetta säädetään pumpulla ja pinnankorkeutta kompressorilla. Tämän havaittiin tuottavan parempia tuloksia kuin tilanteen, jossa säädetään painetta kompressorilla ja pinnankorkeutta pumpulla. Kuvassa oleva punainen käyrä on kunkin suureen asetusarvo ja sininen käyrä on kyseisen suureen mitattu arvo.

Asetettaessa paineen asetusarvo 0,7 bariin säätimien vasteet olivat erilaisia. MIMO-säätimellä ylitys oli suurempi, mutta paine pysyy lähempänä asetusarvoa sen jälkeen. SISO-säätimillä paine värähtelee enemmän. Pinnankorkeus pääsee kasvamaan

isommaksi MIMO-säätimellä ja palautuminen 25 senttimetrin korkeuteen kestää kauemmin kuin SISO-säätimillä.

Kun järjestelmään asetetaan pinnankorkeuden asetusarvoksi 35 senttimetriä, MIMO-säädin lähtee nopeasti kohti sitä. SISO-säädinratkaisu on huomattavasti hitaampi, jolloin ylitystä ei tapahdu. Nopeuden takia MIMO-säätimellä pinnankorkeus ylittyy. MIMO-säädin pääsee nopeammin asetusarvoon kuin SISO-säätimet.

Säätimien tuloksia vertaillaan käyttämällä hyvyyskriteerinä itseisarvollisen erosuureen integraalia (IAE). Erosuuresta eli asetusarvon ja mittauksen erotuksesta otetaan siis itseisarvo ja sitä summataan jokaisella mittausvälillä. Mitä pienempi IAE arvo on, sitä paremmin säätö on onnistunut. IAE-arvoksi MIMO-säädin sai paineelle 491,4 ja pinnankorkeudelle 344,7. SISO-säätimet saivat paineelle 546,7 ja pinnankorkeudelle 415,8. SISO-säätimien tulokset olivat paineelle 11,3 % ja pinnankorkeudelle 20,6 % suurempia kuin MIMO-säätimen tulokset. Erilliset IAE:t summattuna SISO-säätimien tulokset olivat 15,1 % suurempia kuin MIMO-säätimen.

Taulukko 1 – SISO-säätimien ja MIMO-säätimien järjestelmän tunnusluvut paineen asetusarvon muutokselle

Tunnus	Selitys	Yksikkösäätimet	Monimuuttujasäädin
t_r (s)	nousuaika	29	21
t_p (s)	huipun aika	62	51
PO (%)	ylitys	0,48	5,15
t_s (s)	asettumis aika (2%)	44	100

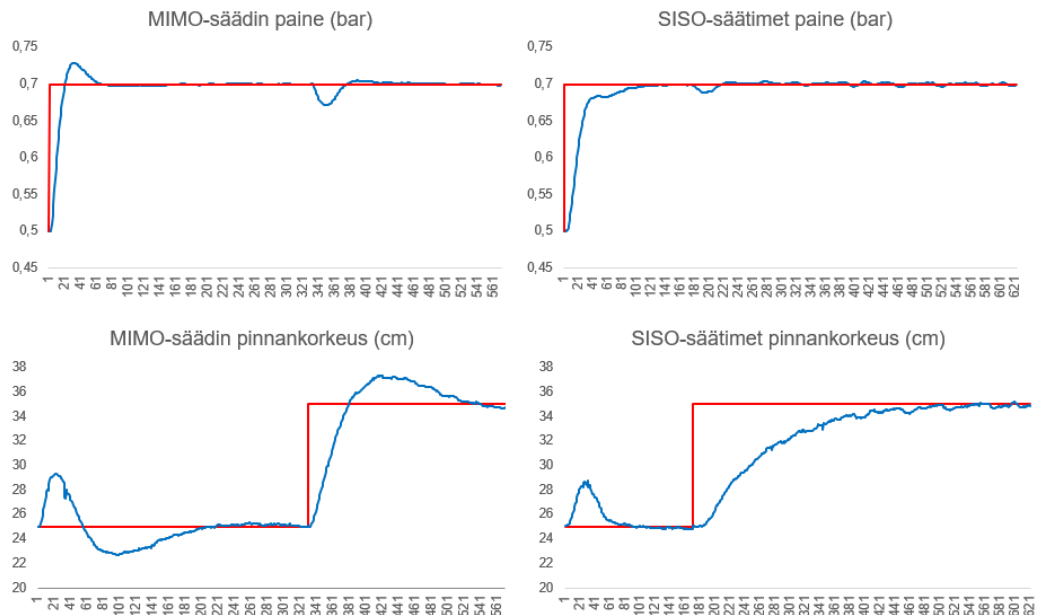
Taulukosta 1 nähdään yksikkösäätimien suoriutuvan servotehtävästä jopa paremmin kuin monimuuttujasäätimen. Kuitenkin yksikkösäätimien IAE-arvo oli suurempi kuin monimuuttujasäätimellä, koska servotehtävän jälkeen paine jää värähtelemään asetusarvon ympäristössä ristikkäisvaikutusten takia enemmän kuin monimuuttujasäätimen tapauksessa.

Taulukko 2 – SISO-säätimien ja MIMO-säätimien järjestelmän tunnusluvut pinnankorkeuden asetusarvon muutokselle

Tunnus	Selitys	Yksikkösäätimet	Monimuuttujasäädin
t_r (s)	nousuaika	485	45
t_p (s)	huipunaika	540	118
PO (%)	ylitys	0	1,21
t_s (s)	asettumisaika (2%)	533	288

Taulukosta 2 nähdään monimuuttujasäätimen suoriutuvan pinnankorkeuden asetusarvon servotehtävästä huomattavasti paremmin kuin yksikkösäätimien.

Viritetään säätimiä nyt nopeammaksi käyttämällä λ_1 :lle arvoa 5 ja λ_2 :lle arvoa 50.

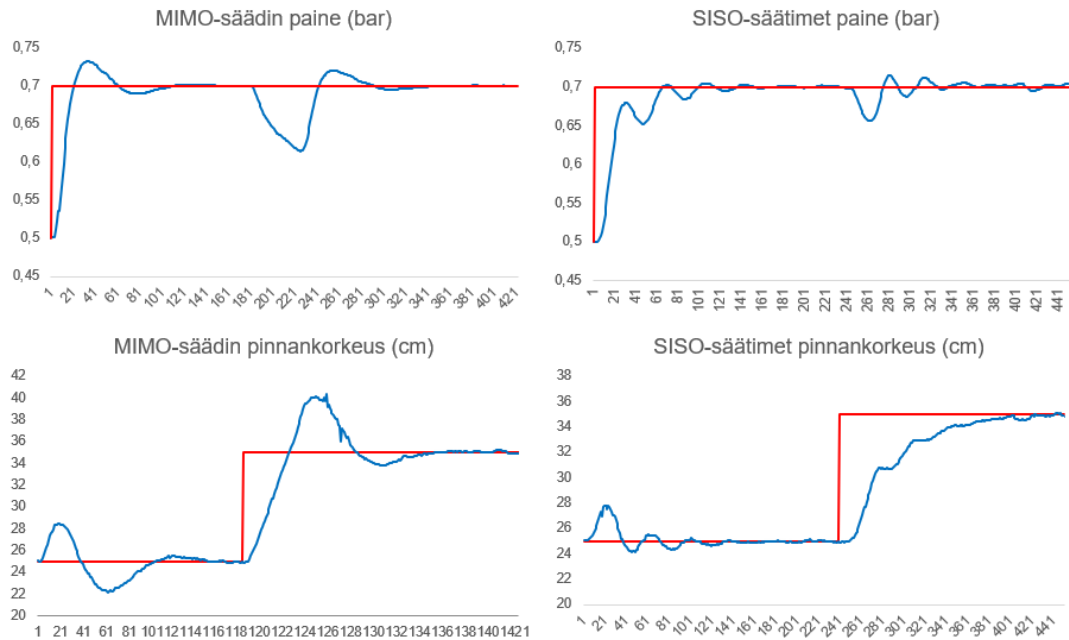


Kuva 13 – Säätötulokset prosessikoikeista

Kuvasta 13 nähdään, että vasteet ovat hyvin samanlaisia kuvan 12 vasteiden kanssa. MIMO-säätimen IAE:t paineelle on 4,2 ja pinnankorkeudelle 13,4. SISO-säätimien IAE:t ovat paineelle 5,6 ja pinnankorkeudelle 19,0. Nyt SISO-säätimien tulokset olivat paineelle 33,3 % ja pinnankorkeudelle 41,8 % suurempia. Erilliset IAE:t summattuna SISO-säätimien tulokset olivat 40,5 % suurempia kuin MIMO-säätimen.

Monimuuttujasäädin haluttiin saada järjestelmässä käyttöön, koska ristikkäisvaikutukset olivat suuret prosessissa. Tämän takia pinnankorkeuden ohjaus viritettiin hitaaksi. Seuraavaksi esitetään tulokset, kun pinnankorkeuden ohjausta viritetään nopeammaksi

käyttämällä λ_1 :lle arvoa 5 ja λ_2 :lle arvoa 25 eli pinnankorkeuden virituksen nopeus kaksinkertaistetaan.



Kuva 14 – Säättötulokset prosessikokeista

Kuvasta 14 havaitaan molempien säätimien suoriutuvan prosessikokeista. SISO-säätimien paineen vasteesta nähdään, että nyt sen säätö häiriintyy huomattavasti ristikkäisvaikutuksista, kun pinnankorkeuden säätö on viritetty nopeaksi. SISO-säätimien molemmat säädettävät suuret värähtelevät asetusrvomutoksen läheisyydessä. MIMO-säätimien vasteetkin värähtelevät, ja ylitykset ovat jälleen suuremmat kuin SISO-säätimellä.

MIMO-säätimen IAE:t paineelle on 7,2 ja pinnankorkeudelle 10,7. SISO-säätimien IAE:t ovat paineelle 6,5 ja pinnankorkeudelle 10,1. Nyt MIMO-säätimen tulokset olivat paineelle 10,8 % ja pinnankorkeudelle 5,9 % suurempia. Erilliset IAE:t summattuna MIMO-säätimen tulokset olivat 8,1 % suurempia kuin SISO-säätimien.

Monimuuttajasäädinrakennetta voidaan pitää onnistuneena, koska se onnistui pienentämään sekä paineen että pinnankorkeuden IAE-arvoa ottamalla huomioon prosessissa esiintyvät ristikkäisvaikutukset tapauksessa, jossa paine viritetään pinnankorkeutta paljon nopeammaksi. Monimuuttajasäädinrakenne ei kuitenkaan pystynyt parempiin tuloksiin, kun pinnankorkeuden nopeus asetettiin lähemmäksi paineen nopeutta. Tällaisessa tilanteessa täytyisi hienosäätää viritysarvoja ja harkita vielä edistyneimpien menetelmien käyttöönottoa.

6. YHTEENVETO

Tässä työssä suunniteltiin ja toteutettiin paperikoneen lyhyttä kiertoa jäljittelevän prosessin säätö käyttäen yksikkösäätimiä ja monimuuttujasäädintä. Yksikkösäätimillä toteutettu säätöratkaisu ei pärjää hyvin monimuuttujaprosessin säädössä, jossa esiintyy voimakkaita ristikkäisvaikutuksia. Monimuuttujasäädinrakenne kykenee huomioimaan prosessissa esiintyvät ristikkäisvaikutukset, jolloin ristikkäisvaikutukset eivät vaikuta enää yhtä paljon säätöön.

Säätö tehtiin molemmissa tapauksissa PI-säädöllä, joka viritettiin yksikkösäätimille IMC-menetelmällä. Yksikkösäätimien viritykset muunnettiin käytettäväksi monimuuttujasäätimelle. Prosessilla tehtiin eriaikaisesti asetusarvomutokset paineeseen ja pinnankorkeuteen. Tuloksia vertailtiin käyttämällä IAE-suorituskykykriteeriä. Käytettäessä monimuuttujasäätöä vasteista havaittiin, että ristikkäisvaikutukset eivät haitanneet säätöä enää samalla tavalla kuin yksikkösäätimien tapauksessa.

LÄHTEET

- [1] M. Morari, E. Zafiriou (1989). Robust process control, Prentice Hall Englewood Cliffs.
- [2] F. Morilla, F. Vázquez ja J. Garrido (2008). Centralized PID Control by Decoupling for TITO Processes, IEEE International Conference on Emerging Technologies and Factory Automation.
- [3] E. F. Camacho, C. Bordons (2007). Model Predictive Control, Springer.
- [4] A. Konar (2005). Fuzzy logic in Process control, Computational intelligence.
- [5] ASE-4020 Laboratoriotyö 1, Opetusmateriaali.
- [6] Verkkosivu. Valmet Automation, Process controls with Valmet DNA automation system: <http://www.valmet.com/automation-solutions/valmet-dna-dcs/>. Viitattu 15.04.2021.
- [7] Valmet Automation, Opetusmateriaali.
- [8] K.J. Åström, T. Hägglund (2006). Advanced PID Control, Research Triangle Park, NC: ISA.
- [9] J. Lieslehto (1996). MIMO controller design using SISO controller design methods, 13th Triennial World Congress, San Francisco, US.

LIITTEET

Java-koodi

```

import Valmet.AITypes.*;

public Pcontrol() throws AITableReferenceException {
    try {

        me_P =      new Ana("me_P");
        me_L =      new Ana("me_L");
        kp11 =      new Ana("kp11");
        kp12 =      new Ana("kp12");
        kp21 =      new Ana("kp21");
        kp22 =      new Ana("kp22");
        ti11 =      new Ana("ti11");
        ti12 =      new Ana("ti12");
        ti21 =      new Ana("ti21");
        ti22 =      new Ana("ti22");
        sp_P =      new Ana("sp_P");
        sp_L =      new Ana("sp_L");
        con_P =     new Ana("con_P");
        conb_P =    new Ana("conb_P");
        con_L =     new Ana("con_L");
        conb_L =    new Ana("conb_L");
        ma =        new Bin("ma");
        av1 =       new Ana("av1");
        av2 =       new Ana("av2");
        av1_2 =     new Ana("av1_2");
        av2_2 =     new Ana("av2_2");

        max = 100;
        min = 0;
        i_P1 = 0;
        i_P2 = 0;
        i_L1 = 0;
        i_L2 = 0;
        u_P = 0;
        u_L = 0;
        man_auto = 0;
        AV1 = 0;
        AV2 = 0;
        e_P = 0;
        e_L = 0;
        u1_suora = 0;
        u1_epasuora = 0;
        u2_suora = 0;
        u2_epasuora = 0;

        Kp11 = 0;
        Ti11 = 0;
        Ki11 = 0;
        Kp12 = 0;
        Ti12 = 0;
        Ki12 = 0;
    }
}

```

```

Kp21 = 0;
Ti21 = 0;
Ki21 = 0;
Kp22 = 0;
Ti22 = 0;
Ki22 = 0;

alustus_i = true;

update_i_P1 = false;
uusi_i_P1 = 0;
update_i_L1 = false;
uusi_i_L1 = 0;

private void runMe() {

// manuaali- vai automaattiohjaus
// 0 = manuaali
// 1 = automaatti
man_auto = ma.getIntValue();

// nykyohjaukset
AV1 = av1.getDoubleValue();
AV2 = av2.getDoubleValue();

// automaattiohjaus mahdollinen vain, jos ensin kompressori tai pumppu on asetettu
// manuaalisesti paalle

if ((man_auto == 1) & ((AV1 > 0) | (AV2 > 0))) {

    // integraattorin alkuarvot, kun
    // pinnankorkeus 25 cm ja paine 0.5 bar

    if(alustus_i) {
        i_P1 = 74;
        i_L1 = 32;
        alustus_i = false;
    }

    // erosuure = asetusarvo – mittaus
    e_P = sp_P.getDoubleValue() - me_P.getDoubleValue();
    e_L = sp_L.getDoubleValue() - me_L.getDoubleValue();

    // erosuureen skaalaus

    // paine 0-1 -> 0-100
    e_P = e_P * 100;
    // pinta 0-60 -> 0-100
    e_L = e_L/60*100;

    Kp11 = kp11.getDoubleValue();
    Kp12 = kp12.getDoubleValue();
    Kp21 = kp21.getDoubleValue();
    Kp22 = kp22.getDoubleValue();

    Ti11 = ti11.getDoubleValue();
    Ti12 = ti12.getDoubleValue();
    Ti21 = ti21.getDoubleValue();
    Ti22 = ti22.getDoubleValue();

```

```

// muunnetaan intergointiajat takaisin vahvistuksiksi

Ki11 = Kp11 / Ti11;
Ki12 = Kp12 / Ti12;
Ki21 = Kp21 / Ti21;
Ki22 = Kp22 / Ti22;

// Koska ohjelma suoritetaan 500ms välein,
// niin skaalataan viritysparametrit sen mukaan

Kp11 = Kp11 * 0.5;
Kp12 = Kp12 * 0.5;
Kp21 = Kp21 * 0.5;
Kp22 = Kp22 * 0.5;

Ki11 = Ki11 * 0.5;
Ki12 = Ki12 * 0.5;
Ki21 = Ki21 * 0.5;
Ki22 = Ki22 * 0.5;

// lasketaan uusi arvo integraattorille

uusi_i_P1 = i_P1 + Ki11*e_P + Ki12*e_L;
uusi_i_L1 = i_L1 + Ki22*e_L + Ki21*e_P;

// lasketaan uudet ohjaukset

// paineen ohjaus
u1_suora = (Kp11 * e_P + Ki12 * e_L);
u1_epasuora = uusi_i_P1;
u_P = (u1_suora + u1_epasuora);

// pinnankorkeuden ohjaus
u2_suora = (Kp22 * e_L + Kp21 * e_P);
u2_epasuora = uusi_i_L1;
u_L = (u2_suora + u2_epasuora);

update_i_P1 = true;
update_i_L1 = true;

// tarkistetaan, onko ohjaus saturoitunut

// paineelle

// onko positiivista saturaatiota
if (u_P > max) {
    // sidotaan ohjaus
    u_P = max;

    // onko virhe saman merkinen?
    if (e_P > 0) {
        update_i_P1 = false;
    }
}

// onko saturoitunut alarajalla?
else if (u_P < min) {
    u_P = min;
    // onko virhe saman merkinen
    if (e_P < 0) {

```



```

        update_i_P1 = false;
    }
}

// pinnankorkeudelle

// onko positiivista saturaatiota?
if (u_L > max) {
    // sidotaan ohjaus
    u_L = max;

    // onko virhe saman merkkinen?
    if (e_L > 0) {
        update_i_L1 = false;
    }
}

// onko saturoitunut alarajalla?
else if (u_L < min) {
    u_L = min;
    // onko virhe saman merkkinen
    if (e_L < 0) {
        update_i_L1 = false;
    }
}

// asetetaan ohjaus analogiaulostuloon
con_P.setDoubleValue(u_P);
av1_2.setDoubleValue(u_P);

// asetetaan ohjaus analogiaulostuloon
con_L.setDoubleValue(u_L);
av2_2.setDoubleValue(u_L);

// integraattoreiden päivittäminen

if (update_i_P1) {
    i_P1 = uusi_i_P1;
}

if (update_i_L1) {
    i_L1 = uusi_i_L1;
}
}

// manuaaliohjaus

else if (man_auto == 0) {
    alustus_i = true;

    // muokataanko ohjauksia
    if ( (AV1 != con_P.getDoubleValue()) | (AV2 != con_L.getDoubleValue()) ) {
        if ( (AV1 >= 0) & (AV1 <= 100) ) {
            con_P.setDoubleValue(AV1);
        }
        if ( (AV2 >= 0) & (AV2 <= 100) ) {
            con_L.setDoubleValue(AV2);
        }
    }
}
}
}

```