

Titta Kemppi

PID-SÄÄTIMEN AUTOMAATTIVIRITTÄMINEN ASKELVASTEeseen SOVITTEILLA MATALAN KERTALUOKAN MALLEILLA

Kandidaatintutkielma
Tekniikan ja luonnontieteiden tiedekunta
Veli-Pekka Pyrhönen
12/2022

TIIVISTELMÄ

Titta Kemppi: PID-säätimen automaattivirittäminen askelvasteeseen sovitetuilla matalan kertaluokan malleilla
Kandidaatintutkielma
Tampereen yliopisto
Automaatiotekniikan TkK-tutkinto-ohjelma
Joulukuu 2022

Prosessiteollisuuden järjestelmien säädössä tyypillinen ongelma on matemaattisfysikaalisen prosessimallin kompleksisuus ja korkea-asteisuus. On hyödyllistä, jos malli saadaan yksinkertaistettua matalampaan asteeseen, jolloin sen käsittely ja säätimen suunnittelu helpottuu. Yksi tapa redusoida prosessimalli on sovittaa sen askelvasteeseen jokin matalan kertaluokan malli ja käsitellä prosessia kuin sen malli vastaisi tätä sovitetta. Prosessiteollisuudessa säätöön käytetään tyypillisesti PID (Proportional–Integral–Derivative) -säätöalgoritmia, jonka parametrit voidaan ratkaista prosessimallien avulla tai kokeellisilla menetelmillä. Korkean kertaluokan mallit joudutaan usein redusoidaan matalan kertaluokan malleiksi, jotta säätimen viritysparemetrit voidaan laskea. Toimivien mallireduktioiden etsimisen lisäksi työn tarkoitus on löytää PID-säätimen viritysparemetrien määrittämiseen jokin automatisoituva menetelmä.

Kaksi yleistä askelvasteeseen sovitettavaa mallia ovat FOTD (First-order system with time delay) ja SOTD (Second-order system with time delay), joiden tuottama askelvaste on monotoninen, ja sopii erityisesti hitaille ja viiveellisille järjestelmille. AMIGO (Approximate MIGO) on FOTD:tä vastaavaan KLT (vahvistus K , viive L , aikavakio T) -malliin perustuva PID-säätimen viritysalgoritmi, jolle muodostetaan automaattinen laskentafunktio. AMIGO perustuu teoreettiseen MIGO (M-constrained Integral Gain Optimization) optimointirutiiniin, jolla yritetään saada robusteja säätöratkaisuja laajalle prosessimallijoukolle. Rajoittava tekijä MIGO:n ja AMIGO:n toiminnassa on normalisoituksi viiveeksi kutsuttu τ , joka on viiveen ja aikavakion normalisoitu väli. Hyvä säätötulos AMIGOlla saadaan vain, jos $\tau \geq 0.3$.

Työn viimeisessä luvussa FOTD:hen sovittamista ja AMIGO-viritystä testataan käytännössä vedenlämmitysprosessissa. Siinä PID-säädin on kytketty tyristorisillan kautta lämmitysvastukseen, joka lämmittää läpivirtaussäilön vettä ja lämpötila pitää saada säädettyä annettuun asetusarvoon mahdollisimman tarkasti. AMIGO-viritykseen käytetään lämmitysprosessin askelvasteeseen sovitettua FOTD-mallia.

Tuloksissa todetaan FOTD- ja SOTD-malleihin sovittamisen sopivan hyvin prosesseille, joiden askelvaste on monotoninen. Säätimen suunnittelun näkökulmasta prosessin dominoiva dynamiikka säilyy, vaikka kertaluokkaa madalletaan. AMIGO-menetelmä ei sovi sellaisenaan tutkittavan lämmitysprosessin säätöön, koska lämmitysprosessin normalisoitu viive on pieni. Paremmaksi menetelmäksi tähän erityistapaukseen osoittautuu lambdaviritys, joka sopii paremmin aikavakion dominoivan dynamiikan järjestelmälle. AMIGOlla kuitenkin saadaan tyypillisille prosessiteollisuuden malleille kohtuullisen hyvin toimivien viritysparemetrien laskenta automatisoitua.

Avainsanat: FOTD, SOTD, MIGO, AMIGO, PID-säädin, säätötekniikka

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

SISÄLLYSLUETTELO

1. JOHDANTO	1
2. PROSESSIEN DYNAAMINEN MALLINNUS	2
2.1 Prosessimallin matemaattisfysikaalinen mallinnus	2
2.2 Prosessin dynaaminen malli askelvasteella	3
3. PROSESSIMALLIN PARAMETRIEN SOVITTAMINEN ASKELVASTEELLA	5
3.1 First-order system with time delay -malli	5
3.2 Second-order system with time delay -malli	8
4. PID-SÄÄTIMEN KOKEELLINEN AUTOMAATTIVIRITYS.....	12
4.1 PID-säädin	12
4.2 Ziegler–Nichols-viritysmenetelmä.....	14
4.3 MIGO	15
4.4 FOTD:n viritysparametrit AMIGOlla	16
4.5 SOTD:n viritysparametrit AMIGOlla.....	17
4.6 Integroivien prosessien viritysparametrit AMIGOlla	18
5. LÄPIVIRTAUSSÄILIÖN VEDEN LÄMPÖTILAN SÄÄTÖ	19
5.1 Lämmitysprosessin askelkoe	19
5.2 Veden lämpötilan takaisinkytketty säätö.....	21
6. YHTEENVETO.....	25
LÄHTEET	26
LIITE A: MATLAB, FOTD-PARAMETRIEN LASKENTAFUNKTIO.....	27
LIITE B: MATLAB, SOTD-PARAMETRIEN LASKENTAFUNKTIO.....	28
LIITE C: MATLAB, PID:N AUTOMAATTIVIRITYSFUNKTIOT	29

LYHENTEET JA MERKINNÄT

Lyhenteet

FOTD	engl. First-order system with time delay, ensimmäisen kertaluokan viiveellinen malli
SOTD	engl. Second-order system with time delay, toisen kertaluokan viiveellinen malli
PID-säätö	engl. Proportional–Integral–Derivative, yleinen säätöalgoritmi
AMIGO	engl. Approximate M- constrained Integral Gain Optimisation, MI-GOn approksimaatio
ZN	Ziegler–Nichols viritysmenetelmä
MIGO	engl. M- constrained Integral Gain Optimisation, M-rajoitettu integraalivahvistuksen optimointi
IE	engl. Intergrated Error, integroitu erosuure

Merkinnät

G_{FOTD}	FOTD-mallin siirtofunktio
K	DC-vahvistus
T	Aikavakio
s	Laplacen muuttuja
L	Prosessin viive
G_a	Approksimoidun FOTD:n siirtofunktio
T_{ar}	Keskimääräinen viipymäaika
T_a	Näennäisaikavakio
L_a	Näennäisviive
G_1	Viiveetön neljännen asteen testisiirtofunktio
G_2	Viiveellinen neljännen asteen testisiirtofunktio
G_{SOTD}	SOTD-mallin siirtofunktio
T_1	SOTD:n ensimmäinen aikavakio
T_2	SOTD:n toinen aikavakio
y_{SOTD}	SOTD:n askelvaste
T_{63}	Ajanhetki, jolloin askelvaste on saavuttanut 63,2 % loppuarvostaan
ω_u	Kriittisen värähtelyn kulmataajuus
K_u	Kriittinen vahvistus
G	Relekokeen prosessin siirtofunktio
y^{set}	Relekokeen prosessin asetusarvo

e	Erosuure
u	Relekokeen prosessin sisäänmeno
P_u	Relekokeen oskilloivan värähtelyn aallonpituus
a_s	Relekokeen oskilloivan värähtelyn amplitudi
K_p	PID-säätimen proportionaalivahvistus
T_i	Integrointiaika
T_d	Derivointiaika
r	Prosessin ohjaus
a_s	ZN-menetelmän askelvastemetodin viritysparameetri
M	MIGOn M-rajoite
$S(s)$	Herkkyysfunktio
$T(s)$	Komplementaarinen herkkyysfunktio
M_s	Herkkyysfunktion maksimiarvo
M_p	Komplementaarisen herkkyysfunktion maksimiarvo
c_r	M-rajoiteympyrän keskipiste
r_r	M-rajoiteympyrän säde
ω	Kulmataajuus
τ	Normalisoitu viive
G_i	Integroivan prosessin siirtofunktio
K_v	engl. Velocity gain, vasteen kasvunopeus

1. JOHDANTO

PID (Proportional–Integral–Derivate) -säädin on prosessiteollisuudessa yleisimmin käytetty säätöalgoritmi [1, s. 1]. Säädin on yksinkertainen, koska sillä on vain muutama vitysparametri ja sen toiminta perustuu nykyhetkeen, menneisyyteen ja tulevaisuuteen. Säätimen virittäminen ei kuitenkaan prosessimallista riippumatta ole täysin suoraviivaista, ja kirjallisuudessa on esitelty useita tapoja optimaalisten vitysparametrien ratkaisemiseen. Tässä työssä käsitellään automaattivirittämistä AMIGO (Approximate M-constrained Integral Gain Optimisation) -vitysalgoritmia ja sen historiaa vitysmenetelmänä.

PID-säätimen virittäminen helpottuu, jos säädettävä prosessi on matemaattiselta malliltaan yksinkertainen. Usein suoraan matemaattisfysikaalisesta mallista muodostetut prosessin siirtofunktiot ovat monimutkaisia ja korkea-asteisia. Yksinkertainen malli on mahdollista muodostaa askelvastekokeella, jossa matalan kertaluokan prosessimallin parametrit sovitetaan askelvasteen käyrämuodon ja nopeuden perusteella. Tässä työssä tutustutaan kahteen malliin, FOTD:hen (First-order system with time delay) ja SOTD:hen (Second-order system with time delay), jotka ovat yksinkertaisia mutta soveltuvat hyvin approksimoimaan monia prosessiteollisuuden systeemejä [1, s. 28].

Työn ensimmäisessä luvussa perehdytään prosessien mallintamiseen ensin matemaattisfysikaalisesta näkökulmasta, minkä jälkeen esitellään kokeellinen askelvastemenetelmä parametrien määrittämiseen. Toisessa luvussa esitellään mallien redusointi. Tässä työssä prosessimallit redusoidaan FOTD:ksi, SOTD:ksi tai yksinkertaisiksi integroiviksi prosesseiksi. Näille reduktioille kirjoitetaan automaattiset laskentafunktiot MATLABilla.

Kolmannessa luvussa tarkastellaan matalan kertaluokan mallien PID-säätöä. Esitellään kolme yleistä vitysmenetelmää, joista AMIGOlle tehdään automaattinen vitysfunktio. Neljännessä ja viimeisessä luvussa vitysmenetelmää testataan käytännössä vedenlämmityskokeella. Luvussa esitellään koejärjestelmä, AMIGOn heikkoudet, toimivampi vitysmenetelmä ja säätötulokset sekä PI- että PID-säätimille.

2. PROSESSIEN DYNAAMINEN MALLINNUS

2.1 Prosessimallin matemaattisfysikaalinen mallinnus

Reaalimaailman systeemien dynamiikkaa halutaan mallintaa, jotta niiden käyttäytymiseen voidaan vaikuttaa erilaisin säätöratkaisuin. Voi esimerkiksi olla, että halutaan mallintaa painekattilan termodynamiikkaa, jolloin saadaan säädettyä sen sisälämpötila haluttuun arvoon. Tällainen *prosessimalli* voidaan selvittää joko analyttisesti tai kokeellisesti. Seuraavaksi esitellään molemmat vaihtoehdot lyhyesti.

Analyttinen mallintaminen perustuu fysiikan lakien käyttämiseen systeemiin kuvauksessa. Tarkoitus on saada aikaiseksi malli, joka koostuu systeemiin vaikuttavasta ulkoisesta vaikutuksesta, *sisäänmenosta* tai *herätteestä* ja prosessista, jonka *ulostuloa* tai *vastetta* halutaan säätää, ja niiden välisestä dynamiikasta. Teoksessaan *Modern Control Systems* Dorf ja Bishop [2, s. 52] esittelevät kuuden kohdan dynaamisten prosessimallien määrittelyyn, joka löytyy alla olevasta taulukosta 1 referoituna.

Taulukko 1. Analyttinen prosessimallin määrittely [2, s. 52]

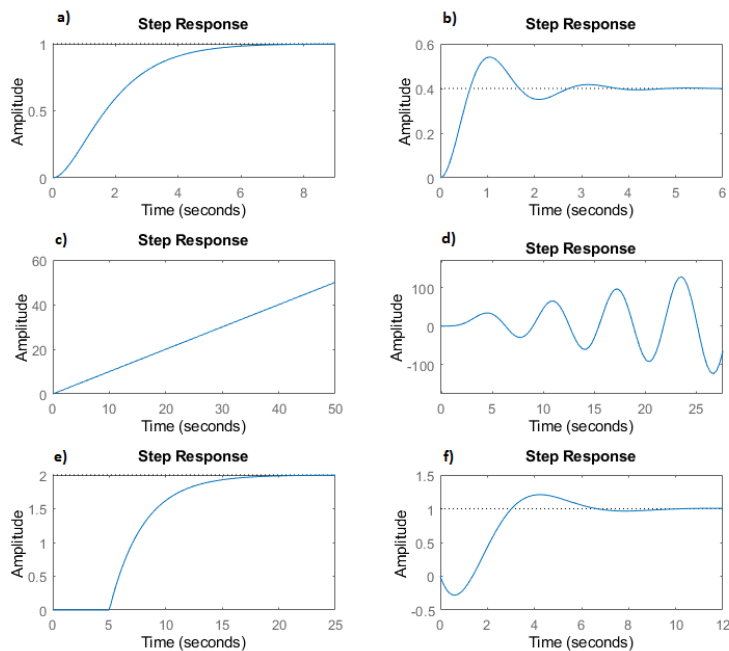
1. Määrittele järjestelmä ja sen komponentit.
2. Formuloi matemaattinen malli ja fundamentaaliset oletukset fysiikan lakeihin perustuen.
3. Kehitä matemaattista mallia kuvaavat differentiaaliyhtälöt.
4. Ratkaise yhtälöistä halutut ulostuloarvot.
5. Tutki saatuja ratkaisuja ja oletuksia.
6. Jos malli ei ole sopiva, analysoi järjestelmää uudelleen.

Matemaattisfysikaalista mallia ei ole aina mahdollista rakentaa tai se ei ole helposti ratkaistavissa vaadittujen oletuksien määrän kasvaessa tai järjestelmän monimutkaistuksessa. Tällöin on järkevämpi käyttää jotakin kokeellista menetelmää mallin identifioimiseksi.

2.2 Prosessin dynaaminen malli askelvasteella

Prosessin dynamiikka voidaan selvittää askelvastekokeella, jonka avulla saadaan selvitettyä yksinkertaistettu matemaattinen malli prosessille. Tätä mallia voidaan hyödyntää prosessin säädön suunnittelussa. Teollisuuden prosessien hitauden vuoksi mallintamiseen usein riittää yksinkertainen dynaaminen malli. Askelvastekokeessa järjestelmään syötetään vakio-ohjaus, jolloin muodostuvaa vastetta kutsutaan askelvasteeksi. Tässä työssä tutustutaan tarkemmin kokeelliseen mallintamiseen ja erityisesti askelvasteen käyttöön mallisovitteiden tekemisessä.

Åströmin ja Hägglundin [1, s. 14] mukaan askelvasteiden muotojen perusteella prosessidynamiikat voidaan jakaa karkeasti kuuteen yleisimpään kategoriaan. Kuvassa 1 on eritelty näiden askelvasteiden käyrämuodot. Tämän tyyppinen prosessien kategorisointi auttaa niiden säädössä, sillä samantyyppisen vasteen omaaville prosesseille voidaan hyödyntää samantyyppisiä säätöratkaisuja.



Kuva 1. Kuusi yleistä yksikköaskelvastetyyppiä

Kuva 1a on tyypillisin askelvaste prosessisäädössä ja -teollisuudessa ja siinä vaste siirtyy monotonisesti s-muotoisena arvosta toiseen. Kuva 1b kuvaa alivaimennettua prosessia, jonka vaste värähtelee vaimenevasti lopullisen tasapainoarvonsa läheisyydessä. Vaste on prosessiteollisuudessa harvinainen ja tyypillisempi mekaanisissa järjestelmissä, joissa on mukana esimerkiksi jousia tai muita joustavia rakenteita.

Sekä 1a että 1b ovat stabiileja prosesseja, toisin kuin kuvien 1c ja 1d prosessit, joissa ulostuloarvot kasvavat rajatta. Kuvan 1c prosessi on integroiva, ja se voi kuvastaa esimerkiksi tasaisesti kasvavaa pinnan korkeutta vesisäiliössä tai paineen kasvua suljetussa astiassa. Kuvan 1d vaste värähtelee ja kasvaa loputtomasti. Kuvan 1e järjestelmässä on paljon viivettä eli vaste ei heti muutu, vaikka ohjausta muutettiin. Pitkäviiveisiä järjestelmiä ovat esimerkiksi prosessit, joissa liikutetaan liukuhihnalla materiaa. Viimeisessä kuvassa 1f on ei-minimivaiheinen prosessi eli alussa vasteen muutos on eri suuntaan kuin sen lopullinen arvo. Myöhemmin työssä tarkastellaan kahta askelvastemallia, a:ta ja e:tä, joihin sovitetaan matalan kertaluokan malleja.

3. PROSESSIMALLIN PARAMETRIEN SOVITTAMINEN ASKELVASTEELLA

Tässä työssä keskitytään tutkimaan monotonisia askelvasteita eli vasteita, jotka eivät vaihda suuntaa missään vaiheessa eli ovat värähtelemättömiä. Tämän tyyppiin askelvasteisiin muodostetaan mallisovitteita, jotta niiden dynamiikkaa voidaan kuvata prosessin matemaattisfysikaalista mallia yksinkertaisemmalla mallilla. Työssä esitellään ja käytetään kahta prosessiteollisuudessa yleistä mallityyppiä, FOTD:tä ja SOTD:tä. Tutkittaviin askelvasteisiin sovitetaan näiden mallien parametrit ja kokeillaan, kuinka hyvin tulos vastaa alkuperäistä. Selvitetään myös, voiko alkuperäistä systeemiä säätää approksimoitujen sovitteiden avulla lasketuilla säädinparametreilla.

3.1 First-order system with time delay -malli

First-order system with time delay -malli eli FOTD on yleinen malli teollisuuden prosessissäädössä, sillä se on yksinkertainen ja sillä voidaan approksimoida erityisesti prosessiteollisuudessa tyypillisiä hitaita ja värähtelemättömiä korkean asteen prosessimalleja [1, s. 28]. FOTD-järjestelmän siirtofunktio on

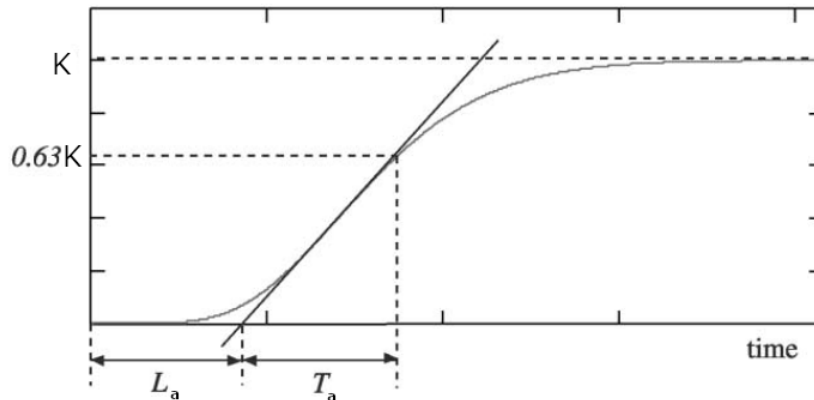
$$G_{FOTD}(s) = \frac{K}{T_S + 1} \cdot \exp(-Ls), \quad T > 0, L > 0, \quad (1)$$

jossa K on DC-vahvistus, T on aikavakio ja L on prosessin viive. Aikavakio kuvaa hetkeä, jona askelvasteen arvo on saavuttanut lähteestä riippuen 63 tai 63,2 prosenttia loppuarvostaan. Lisäksi laskennassa yleinen parametri keskimääräinen viipymäaika $T_{ar} = T + L$ on viive lisättyä aikavakioon eli aika, jona syötetty sisäänmeno aiheuttaa merkittävän muutoksen prosessin vasteeseen.

FOTD-sovitetta varten muokataan kaava (1) muotoon, johon redusoitava prosessimalli yritetään saada. Tätä mallia kuvaa kaava

$$G_a(s) = \frac{K}{T_a s + 1} \cdot \exp(-L_a s), \quad (2)$$

jossa T_a ja L_a ovat näennäisaikavakio ja näennäisviive, eli ne ovat approksimoituja arvoja. Arvot T_a ja L_a tulee valita tapauskohtaisesti ja siten, että niiden tuottaman sovittimen askelvaste vastaa mahdollisimman hyvin alkuperäisen siirtofunktion askelvastetta. Parametrit voidaan määrittää suoraan askelvestatusta asettamalla vasteen jyrkimpään kohtaan tangentti, kuten kuvassa 2.



Kuva 2. Askelvasteeseen asetettu jyrkin tangenti [3, s. 2]

Kohdasta, jossa tangenti leikkaa aika-akselin, määritetään näennäsviive L_a . Näennäsviive T_a voidaan keskimääräisen viipymääajan ja viiveen T_{ar} erotuksena. Vahvistus K on se arvo, johon askelvaste asettuu. Sijoittamalla parametrit kaavaan (2), askelvasteeseen saadaan FOTD-sovite. Samaa laskentatapaa käytetään liitteen A FOTD-sovitteen laskentafunktiossa, joka on toteutettu MATLABilla.

Sovitteiden tarkkuutta tutkitaan viiveettömällä neljännen asteen siirtofunktiolla (3) ja viiveellisellä neljännen asteen siirtofunktiolla (4). Kaavat

$$G_1(s) = \frac{1}{s^4 + 7s + 15s^2 + 10s + 1} \exp(-s) \quad (3)$$

ja

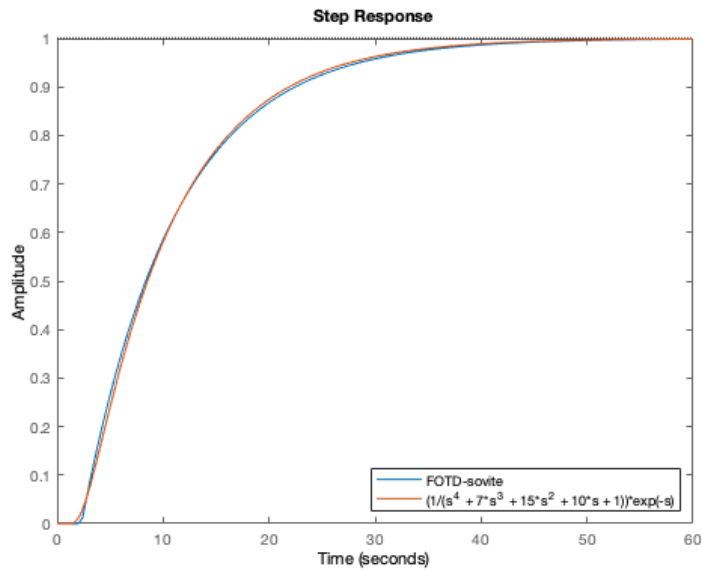
$$G_2(s) = \frac{1}{(1+s)(1+0.2s)(1+0.05s)(1+0.01s)} \quad (4)$$

ovat dynamiikaltaan monotonisia ja soveltuvat hyvin sovittavaksi FOTD- ja SOTD-malleihin. Taulukossa 1 on esitetty siirtofunktiolla tulokseksi saadut FOTD-parametrit. Sijoittamalla nämä kaavaan (1) saadaan ratkaistua siirtofunktioiden ensimmäisen asteen mallireduktiot.

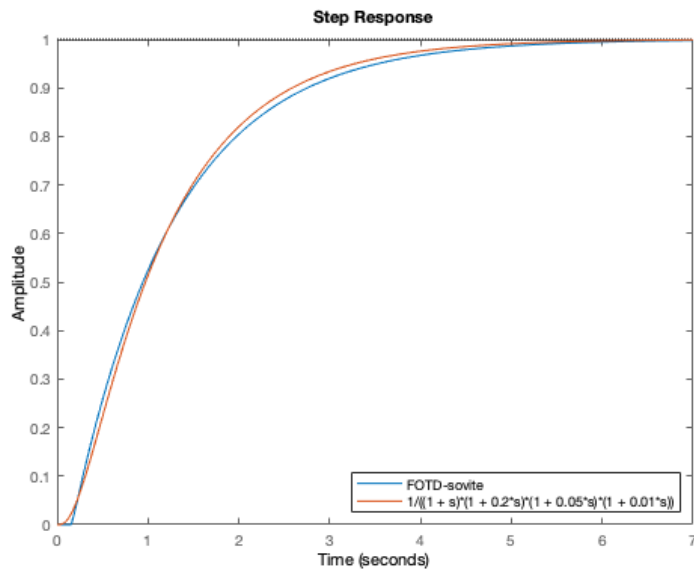
Taulukko 1. Lasketut FOTD-parametrit

FOTD	K	L_a	τ_a
G_1	1	8,7498	2,3026
G_2	1	1,1237	0,1658

Kuviin 3 ja 4 on piirretty sekä alkuperäisen että mallireduktion askelvasteet. Oranssi käyrä kuvaa alkuperäisen siirtofunktion askelvastetta ja sininen sovitetun mallin askelvastetta. Kuten nähdään, FOTD-sovite vastaa kohtuullisen hyvin alkuperäistä siirtofunktiota. Kuvan 4 sovitteessa on enemmän virhettä, ja siihen voi olla syytä käyttää korkeamman asteen approksimaatiota paremman tuloksen saamiseksi.



Kuva 3. Siirtofunktion $G_1(3)$ ja sen FOTD-sovitteen askelvasteet



Kuva 4. Siirtofunktion $G_2(4)$ ja sen FOTD-sovitteen askelvasteet

Vaikka alkuperäisten mallien astelukua vähennettiin kolmella, approksimoi yksinkertainen FOTD kyseisiä prosesseja hyvin. Mallisovitteiden sopivuudesta huolimatta tiedostettakoon, että tutkitut siirtofunktiot on valittu siten, että ne soveltuvat erityisen hyvin tämäntyyppiseen sovittamiseen.

3.2 Second-order system with time delay -malli

Second-order system with time delay, eli SOTD-malli (5) on samankaltainen FOTD-malliin verrattuna, mutta se on nimensä mukaisesti toista kertaluokkaa. Tällaisen mallin askelvasteessa on aina myös kaarevuutta, eli s-muotoisuutta. Malli on kaavaltaan muotoa

$$G_{SOTD}(s) = \frac{K}{(T_1s + 1)(T_2s + 1)} \cdot \exp(-Ls), \quad (5)$$

$$T_1 > 0, T_2 > 0, L > 0, K > 0.$$

Sovitetta varten tulee siis löytää kaksi aikavakiota T_1 ja T_2 sekä DC-vahvistus K ja viive L . Voidaan olettaa, että $T_1 \geq T_2$ [1, s. 51]. SOTD-mallin yksikköaskelvaste on muotoa

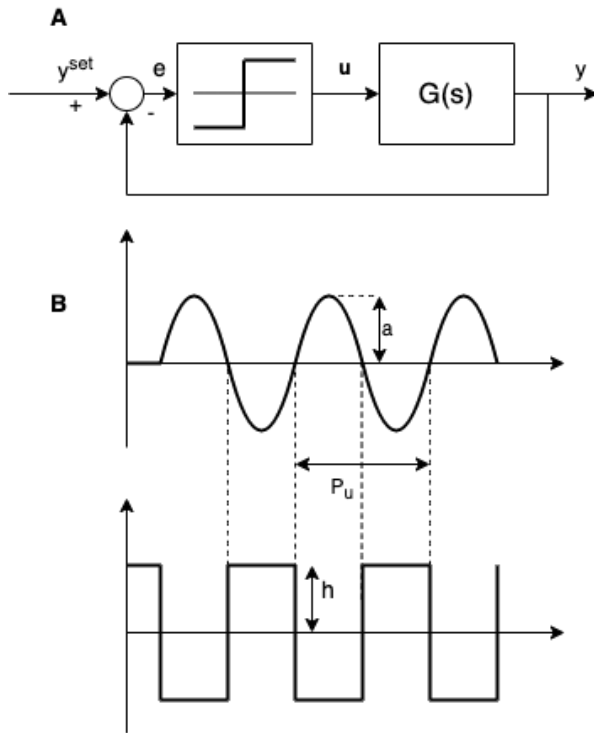
$$y_{SOTD}(t) = \begin{cases} K_p \left(1 - \frac{T_1}{T_1 - T_2} \exp\left(-\frac{t-L}{T_1}\right) - \frac{T_2}{T_2 - T_1} \exp\left(-\frac{t-L}{T_2}\right) \right), & \text{jos } T_1 \neq T_2 \\ K_p \left(1 - \exp\left(-\frac{t-L}{T_1}\right) - \frac{t}{T_1} \exp\left(-\frac{t-L}{T_1}\right) \right), & \text{jos } T_1 = T_2. \end{cases} \quad (6)$$

Askelvasteesta on laskennallisesti vaikea erottaa kahta aikavakiota, ja siksi hyvään sovitukseen tarvitaan myös taajuusvastedataa.

SOTD-sovitteen parametrit voidaan estimoida yhdistämällä askelvaste- ja taajuusvastedatat. Näin vahvistus K ja ajanhetki T_{63} saadaan askelvastedatasta ja kriittisen värähtelyn kulmataajuus $\omega_u = \omega_{180}$ sekä kriittinen vahvistus $K_u = 1/K_{180}$ taajuusdatasta. Muodostetaan kaavapari

$$\begin{cases} K^2 K_u^2 = (1 + \omega_u^2 T_1^2)(1 + \omega_u^2 T_2^2) \\ \pi = \arctan(\omega_u T_1) + \arctan(\omega_u T_2) + \omega_u L. \end{cases} \quad (7)$$

Yksi automatisoitu tapa selvittää kriittisen värähtelyn kulmataajuus sekä kriittinen vahvistus on reletakaisinkytkennältä (relay feedback) [1, s. 53]. Kuvassa 5A näkyy relekokeen lohkokaaavio, missä $G(s)$ on tutkittavan prosessin siirtofunktio, y on säädetty ulostulo, y^{set} on asetuservo, e on virhe ja u on säädetty prosessin sisäänmeno, joka on saatu releestä. Koejärjestelmä on negatiivisesti takaisinkytketty.



Kuva 5. Relekokeen lohkokaavio ja relekokeen tulokset

Kokeen ajatus on, että releen avulla saadaan systeemin ulostulo automaattisesti oskiloimaan kriittisen värähtelyn kulmataajuudella, jolloin sen arvo voidaan laskea suoraan jaksonajasta. Kuvassa 5B aallonpituutta merkitään P_u , jolloin kriittisen värähtelyn kulmataajuus ω_u saadaan kaavalla

$$\omega_u = \frac{2\pi}{P_u} \quad (8)$$

ja vahvistus K_{180} saadaan kaavasta

$$K_{180} = \frac{\pi a}{4d}, \quad (9)$$

jossa a on ulostulon amplitudi ja d releen kanttiaallon amplitudi. Yhdistämällä yksikköaskelevasteen ja taajuusvasteen kaavat (6) ja (7) saadaan kaksi yhtälöparia

$$0 = \begin{cases} 0.368 - \frac{T_1}{T_1 - T_2} \cdot \exp\left(-\frac{T_{63} - L}{T_1}\right) - \frac{T_2}{T_2 - T_1} \cdot \exp\left(-\frac{T_{63} - L}{T_2}\right), \text{ jos } T_1 \neq T_2 \\ 0.368 - \exp\left(-\frac{T_{63} - L}{T_1}\right) - \frac{T_{63}}{T_1} \cdot \exp\left(-\frac{T_{63} - L}{T_1}\right) - 0.63, \text{ jos } T_1 = T_2 \end{cases} \quad (10)$$

ja

$$\begin{cases} 0 = (1 + \omega_u^2 T_1^2)(1 + \omega_u^2 T_2^2) - K^2 K_u^2 \\ 0 = \arctan(\omega_u T_1) + \arctan(\omega_u T_2) + \omega_u L - \pi, \end{cases} \quad (11)$$

joiden avulla sovittien parametrit voidaan ratkaista.

Näistä yhtälöistä SOTD-parametreja ei pysty ratkaisemaan suoraan niiden epälineaarisuuden vuoksi, joten parametrit pitää ratkaista iteratiivisesti. Tätä vaikeuttaa tilanne jossa aikavakiot ovat yhtä suuret. Toinen ratkaisutapa on iteroida suhdetta $a = \frac{T_2}{T_1}$. Tämä valinta muuttaa yhtälöparin (7) ensimmäisen yhtälön muotoon (12):

$$(1 + \omega_u^2 T_1^2)(1 + a^2 \omega_u^2 T_1^2) = K^2 K_u^2. \quad (12)$$

Tästä voidaan ratkaista T_1 ja edelleen T_2 sekä L , kuten kaavat

$$T_1 = \frac{1}{a\omega_u\sqrt{2}} \sqrt{\sqrt{4a^2 K^2 K_u^2 + (1 - a^2)^2} - 1 - a^2}, \quad (13)$$

$$T_2 = aT_1 \quad (14)$$

ja

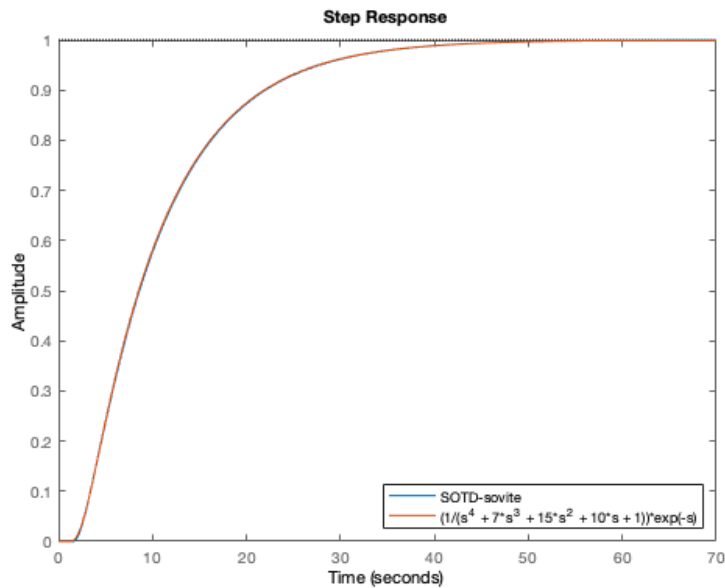
$$d = \frac{\pi - \arctan(\omega_u T_1) - \arctan(\omega_u T_2)}{\omega_u} \quad (15)$$

havainnollistavat.

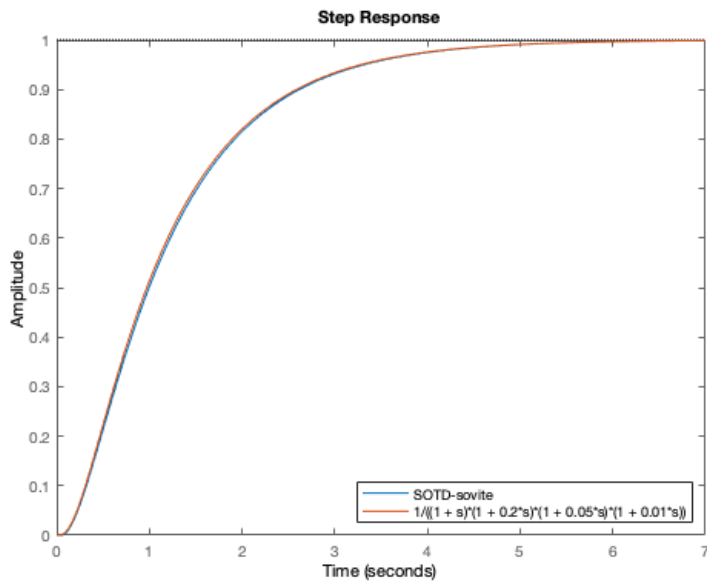
Askelvastefunktio kaavassa (6) voidaan ratkaista a :n funktiona ja a :ta voidaan iteroida niin että tulos vastaa hetken T_{63} arvoa. Sovitteen kaavojen johto on tehty mukaillen Åströmin ja Haglundin vastaavaa [1, s. 54]. Liitteessä B on SOTD-parametrien automaattinen määrittäminen toteutettuna MATLAB-funktiona. Funktiolla lasketut parametriarvot löytyvät taulukosta 2, ja kuvat 6 ja 7 esittävät tulokset kaavoissa 3 ja 4 esitetyille korkean asteen siirtofunktioille.

Taulukko 2. Lasketut SOTD-parametrit

	K	L_a	τ_1	τ_2
G_1	1	1.5854	8.3297	1.1662
G_2	1	0.0509	1.0010	0.2302



Kuva 6. Siirtofunktion $G_1(3)$ ja sen SOTD-sovitteen askelvasteet



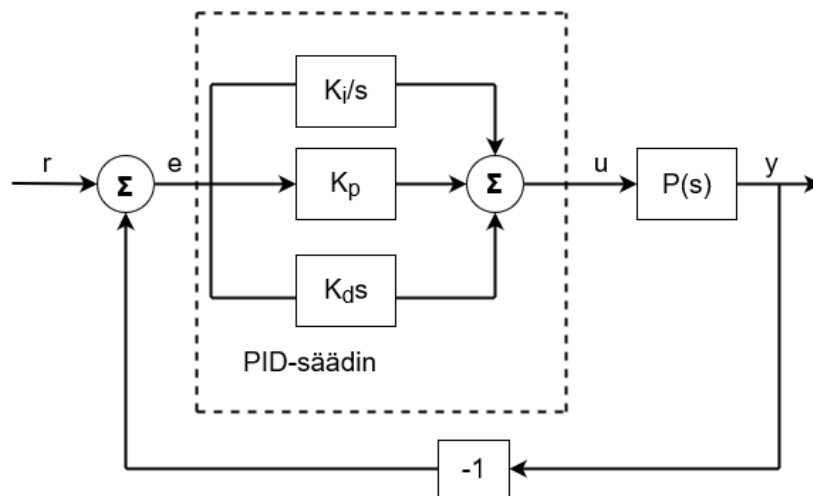
Kuva 7. Siirtofunktion $G_2(4)$ ja sen SOTD-sovitteen askelvasteet

Kuten kuvista nähdään, vastaavat SOTD-sovitteet varsin hyvin alkuperäisiä siirtofunktioita. Toisen asteen sovite antaa mahdollisuuden kaarevuudelle ja erityisesti siirtofunktiolle G_2 tämä sovite sopii FOTD:tä paremmin. Näin siirtofunktiomalli on saatu redusoitua toiseen asteeseen ja voidaan edetä keskustelemaan mallien säätömahdollisuuksista.

4. PID-SÄÄTIMEN KOKEELLINEN AUTOMAATTIVIRITYS

4.1 PID-säädin

PID-säädin on ylivoimaisesti yleisimmin käytetty säätöalgoritmi, jota käytetään takaisinkytkettyihin järjestelmiin sellaisenaan ja varioituna [4, s. 59]. PID-säädin soveltuu erinomaisesti myös FOTD- ja SOTD-mallien säätöön, ja sen viritysparametrien määrittämistä AMIGO-menetelmällä esitellään seuraavissa luvuissa. Nyt tutustutaan PID-säätimen toimintaperiaatteeseen.

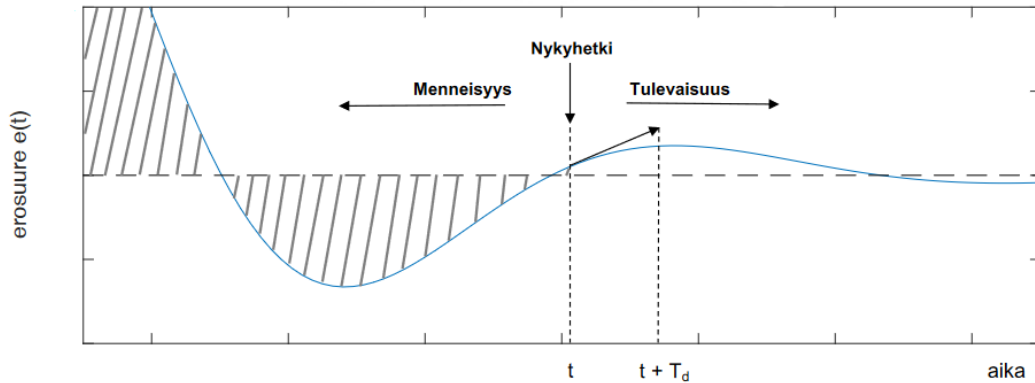


Kuva 8. PID-säädetty järjestelmä

Tyypillinen PID-säädin koostuu kolmen termin summasta, ja sen lohkokaaevio yhdistettynä takaisinkytkettyyn järjestelmään on esitetty kuvassa 8. Näitä kolmea termiä kutsutaan P-, I- ja D-haaroiksi, joista jokaisella on oma vaikutuksensa säätötulokseen. Ideaalin PID-säätimen ulostulo voidaan siten esittää kaavalla

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right), e(t) = r(t) - y(t). \quad (16)$$

P-haaraa kutsutaan proportionaalisäätimeksi, ja sen viritysparametri on proportionaalivahvistus K_p . I-haarassa on integroiva säädin, jonka viritysparametri on integraalivahvistus K_i . D-haarassa on derivoiva säädin, jonka viritysparametri on vastaavasti K_d , eli derivaatan vahvistus [4, s. 293]. Tässä tutkielmassa säätöparametrien K_i ja K_d tilalla käytetään integraaliaikaa $T_i = K_p/K_i$ sekä derivointiaikaa $T_d = K_p/K_d$.



Kuva 9. PID-säätimen haarojen toiminta hetkellä t [5, s. 5]

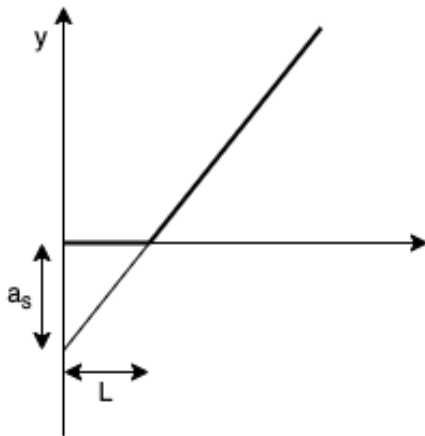
Kuva 9 havainnollistaa PID-haarojen toimintaa. P-säädin muodostaa ohjauksen nykyhetkessä kertomalla erosuuretta e vakiovahvistuksella K_p . Siten säätötulos on suoraan verrannollinen erosuureen kanssa, ja ohjaus lähestyy nollaa, kun erosuure lähestyy nollaa. Käytännössä P-säädöllä harvoin saadaan tarkkaa asetusravon seurantaa, koska se vaatisi tarkan säädön nollaohjauksella. Tämän vuoksi pelkkä P-säätö harvoin riittää hyvän säätötuloksen saamiseksi, ja järjestelmään pitää lisätä integrointi. Kuvassa P-haaran vaikutusta esittää nykyhetki, sillä sen ulostulo $u_p(t) = K_p e(t)$ riippuu vain hetken t arvosta erosuureessa.

I-haarassa pieni säätövirhe aiheuttaa aina vaikutuksen säätösignaaliin riippumatta siitä, miten pieni virhe on. Integrointihaaran ohjaus muodostuu hetkeen t asti kertyneen erosuureen historia-arvojen integraalista, kuvassa 9 viivoin väritetty alue. Integroivan haaran vaikutus säätötulokseen on niin merkittävä, että suurin osa PID-säätimestä toimii PI-moodissa ilman derivointia [6, s. 293]. Oikein viritetyllä derivointihaaralla on joskus mahdollista parantaa asetusravovastetta. Prosessidynamiikan takia vie aikaa, että säätimen ulostulo näkyy prosessin ulostulossa, mistä johtuen on säädin aina myöhässä virheen korjauksessa. Derivaatalla pyritään ennustamaan prosessia siten, että säädetään vasta odotettavaa prosessin ulostuloa ekstrapoloimalla virhettä tulevaisuuteen, jota on havainnollistettu kuvassa 9 ekstrapoloitulla tangentilla. Derivointiaika T_d kuvaa, kuinka pitkän ajan päähän ekstrapolointi on tehty, ja mitä korkeampi derivointiaika, sitä epätarkempi ennustus ja värähtelevämpi vaste. Kaavaa (16) kutsutaan PID-säätimen standardimuodoksi ja säätimen siirtofunktiota kuvaa kaava

$$C(s) = K_p \left(1 + \frac{1}{sT_i} + sT_d \right). \quad (17)$$

4.2 Ziegler–Nichols-viritysmenetelmä

PID-säätimen viritykseen on olemassa monia analyttisiä ja kokeellisia menetelmiä. Yksi klassinen ja suosittu [7, s. 111] PID-säätimen viritysmenetelmä on vuonna 1942 Zieglerin ja Nicholsin esittelemä kokeellinen algoritmi [8], sittemmin nimetty Ziegler–Nichols-menetelmäksi, eli ZN-menetelmäksi. Siitä on olemassa kaksi erilaista versiota, askelvaste- ja taajuusvastemetodi [4, s. 135], esitellään molemmat lyhyesti. Askelvastemetodi perustuu avoimen järjestelmän askelvasteeseen, josta kuvataan kahdella parametrilla a_s ja L , jotka määritetään kuvassa 10. Kuvassa näkyvä tangentti piirretään askelvasteen jyrkimpään kohtaan. Tällöin prosessin askelvastetta käsitellään kuin prosessi koostuisi pelkästä viiveestä ja integraattorista. Tällä muutoksella Ziegler ja Nichols määrittivät PID-parametrit, jotka löytyvät taulukosta 3.



Kuva 10. Askelvasteesta määritetyt parametrit.

Taulukko 3. Askelvastemetodin PID-parametrit.

Säädin	K	T_i	T_d
P	$1/a_s$	–	–
PI	$0,9/a_s$	$3L$	–
PID	$1,2/a_s$	$2L$	$L/2$

Taajuusvastemenetelmän parametrien määrittäminen perustuu kriittisen värähtelyn kulmataajuuden ja kriittisen vahvistuksen selvittämiseen, mikä tehdään kappaleessa 3.3 määritellyllä relekokeella. Ziegler ja Nichols määrittivät PID-säätimen parametreille yksinkertaiset kaavat näiden suhteen, mitkä löytyvät taulukosta 4.

Taulukko 4. Taajuusvastemetodin PID-parametrit

Säädin	K	T_i	T_d
P	$0,5K_u$	–	–
PI	$0,4K_u$	$0,8\omega_u$	–
PID	$0,6K_u$	$0,5\omega_u$	$0,125\omega_u$

ZN-menetelmällä saadaan aikaiseksi prosessille nopea askelvaste, mutta ylitystä tulee paljon ja häiriövaste on huono viiveellisille järjestelmille. Näistä syistä ZN-menetelmää on koettu liian aggressiiviseksi. Viritysmenetelmä on yksinkertainen, mutta tuottaa harvoin hyvän säätötuloksen. Se on kuitenkin edelleen laajassa käytössä johtuen sen intuitiivisuudesta ja siitä, että prosessia ei tarvitse tuntea ennalta kovin hyvin [4, s. 142].

4.3 MIGO

Julkaisussaan *Revisiting the Ziegler–Nichols step response method for PID control* [8] Åström ja Hägglund tutkivat ZN-menetelmän askelvastemetodia ja esittelevät siitä johdetun, kehittyneemmän menetelmän MIGO (M- constrained Integral Gain Optimisation) ja sitä approksivoivan menetelmän AMIGO (Approximate MIGO).

MIGOSSa optimoidaan säädön häiriövastetta yksinkertaisella Integrated error (IE) -tunnusluvulla [10, s. 2]. Tunnusluku tarkoittaa integroitua erosuuretta, joka määritetään kaavalla

$$IE = \int_0^{\infty} e(t)dt = \frac{1}{k_i}. \quad (18)$$

Sen suuruutta käytetään säädön suorituskyvyn mittarina, ja mitä pienempi arvo, sitä parempi on säätö. Minimoiminen tässä tapauksessa tarkoittaa integrointivahvistuksen k_i kasvattamista mahdollisimman suureksi. Optimointifunktiota rajoittaa termi M, jossa yhdistyy herkkyysfunktion

$$S(s) = \frac{1}{1 + G(s)C(s)} \quad (19)$$

ja komplementaarisen herkkyysfunktion

$$T(s) = \frac{G(s)C(s)}{1 + G(s)C(s)} \quad (20)$$

amplitudivahvistusten suurimmat arvot. Merkitään näitä arvoja $M_s = |S(s)|_{max}$ sekä $M_p = |T(s)|_{max}$. Nämä kaksi termiä määräävät robustin, eli vakaan takaisinkytkennän

[10, s. 3], jossa M_s pitää järjestelmän stabiilina ja M_p rajoittaa resonanssipeikkien suuruutta. Niiden molempien vaikutus yhdistetään ehdolla, jossa $M \geq M_p$ sekä $M \geq M_s$.

Muuttuja M muodostaa taajuustasossa esitettävälle Nyquist-käyrälle ympyrän. Åström ja Hägglund [9, s. 3] esittävät, että takaisinkytketyn järjestelmän Nyquist-käyrän tulee asettua tämän ympyrän ulkopuolelle, aiheuttaen vaatimuksen M -ympyrälle, jonka keskipiste on c_R ja säde r_R . Nämä vaatimukset löytyvät esitettynä kaavassa

$$c_R = \frac{2M^2 - 1}{2M(M - 1)}, r_R = \frac{2M - 1}{2M(M - 1)}. \quad (21)$$

Yhdistämällä M -vaatimuksen sekä optimoitava funktion, saadaan ongelma formuloitua neljän muuttujan funktioksi (22),

$$f(k_p, k_i, k_d, \omega) = |c_R + GC|^2 \geq r_R^2 \quad (22)$$

jossa on uutena parametrina ω eli kulmataajuus. Ongelma voidaan kehittää eteenpäin geometrisesti tulkittavaan muotoon, josta lopulliset optimaaliset viritysparametrit pystytään löytämään. Kyseinen mallinnus ei kuitenkaan ole tämän työn kannalta oleellista.

4.4 FOTD:n viritysparametrit AMIGOlla

FOTD-mallille voidaan määrittää parametri $\tau = \frac{L}{T_{ar}} = \frac{L}{L+T}$, jota kutsutaan normalisoiduksi viiveeksi [1, s. 26], sekä myös *säädettävyyssuhteeksi* (controllability ratio). Sen arvot vaihtelevat välillä $0 \leq \tau \leq 1$. Karkeasti sanottuna, mitä lähempänä arvoa yksi normalisoitu viive on, sitä vaikeammin säädettävä prosessi on kyseessä.

Lähteessä [11] esitellään AMIGOn käyttö PI-parametrien ratkaisemiseen. Herkkyyden M arvona käytetään $M = 1.4$, sillä sen on todettu olevan hyvä kompromissi säätötuloksen ja robustiuden välillä. PI-säännöt

$$\begin{cases} K_p = \frac{0.15}{K} + \left(0.35 - \frac{LT}{(L+T)^2}\right) \frac{T}{KL} \\ T_i = 0.35L + \frac{13LT^2}{T^2 + 12LT + 7L^2} \end{cases} \quad (23)$$

perustuvat prosessien approksimointiin FOTD-mallina (myös KLT-malli) ja integroivien prosessien mallina (tästä lisää kappaleessa 4.6). Sääntöjen määrittämiseksi käytettiin ennalta määrätyle prosessimallien testierälle MIGO-menetelmää ja saatiin PI-parametrit K_p ja T_i . AMIGO-säännöt ratkaistiin etsimällä korrelaatiota PI-säätöparametrien ja prosessin KLT-parametrien väliltä.

MIGOn jälkeen Åström ja Hägglund jatkavat lähteessä [9] pohtimaan, josko FOTD-mallia voisi hyödyntää MIGOlla löydettävien PID-parametrien approksimointiin. Tästä nimi *approximate* MIGO. He toteavat heti, että ei ole mahdollista löytää optimaalisia PID-parametreja, mutta on kuitenkin mahdollista löytää robustit, ei-optimaaliset laskusäännöt. He ehdottavat PID-säätimen parametrien laskemiseksi konservatiivista kaavakokoelmaa

$$\begin{cases} K_p = \frac{1}{K} \left(0.2 + 0.45 \frac{T}{L} \right) \\ T_i = \frac{0.4L + 0.8T}{L + 0.1T} L \\ T_d = \frac{0.5LT}{0.3L + T}, \end{cases} \quad (24)$$

joka on käyttökelpoinen laajalle määrälle prosesseja, jos $\tau \geq 0.5$. M-arvo saa heitellä määritetystä 1.4:stä vain 15 %.

Säätöjen toimivuuden rajoittavana terminä toimii derivointitermi T_d . Säännöt approksimoivat P- ja I-termejä hyvin, jos $0.3 \leq \tau \leq 0.5$. Tällä välillä parametri T_d tulisi olla jopa kaksi kertaa suurempi kuin AMIGO-säännöllä laskettava arvo. Tätä vielä matalammilla normalisoidun viiveen alueilla derivointiaika saa vaihtelevasti liian suuria tai pieniä arvoja verrattuna MIGO-menetelmän tuloksiin. Kun $\tau < 0.2$ AMIGO antaa ajalle T_i liian suuren arvon, jos tutkittava prosessi ei ole KLT. Vastaavasti jos $\tau < 0.3$, saadaan muille kuin KLT-prosesseille liian pieni proportionaalivahvistuksen arvo. Näistä rajoitteista huolimatta tulokset antavat konservatiiviset PID-arvot kaikille lähteessä [9] testatuille prosesseille. MATLABin funktiototeutus kaavakokoelmalle (24) löytyy liitteestä C.

4.5 SOTD:n viritysparametrit AMIGOlla

AMIGOon perustuvat PID-parametrit SOTD-mallille voidaan määrittää virityssäännöllä

$$\begin{cases} KK_p = \alpha_1 + \frac{\alpha_2 T_1}{L} + \frac{\alpha_3 T_2}{L} + \frac{\alpha_4 T_1 T_2}{L} \\ KK_i = \frac{\beta_1}{d} + \frac{\beta_2 T_1}{L^2} + \frac{\beta_3 T_2}{L^2} + \frac{\beta_4 T_1 T_2}{L^3} \\ KK_d = \left(\gamma_1 L + \gamma_2 T_1 + \gamma_3 T_2 + \gamma_4 \frac{T_1 T_2}{L} \right) \frac{T_1 + T_2}{T_1 + T_2 + L} \end{cases} \quad (25)$$

joihin Hägglund ja Åström ehdottavat virityssäätöjen puuttuviksi vakioparametreiksi seuraavia:

$$\begin{aligned} \alpha_1 &= 0.19, & \alpha_2 &= 0.37, & \alpha_3 &= 0.18, & \alpha_4 &= 0.02 \\ \beta_1 &= 0.48, & \beta_2 &= 0.03, & \beta_3 &= -0.0007, & \beta_4 &= 0.0012 \\ \gamma_1 &= 0.29, & \gamma_2 &= 0.16, & \gamma_3 &= 0.20, & \gamma_4 &= 0.23. \end{aligned}$$

Näiden sääntöjen käyttäminen ja SOTD-sovitteen tekeminen ovat tyypillisiä hitaissa (*lag-dominated*) järjestelmissä. Tällaisissa järjestelmissä sekä viive että pienet aikavakiot sisältyvät termiin L , jolloin säätötulos saattaa parantua merkittävästi, jos mallissa käytetään kahta aikavakiota [1, s. 264].

AMIGO-virityssäännöt SOTD-prosesseille antaa samankaltaisen tuloksen kuin säännöt (24) dynaamisesti hitaille ja matalan aikavakion prosesseille, mutta kasvattaa proportionaalivahvistusta suuren viiveen (*delay-dominated*) prosesseille. Tämäkin parametrisointi löytyy toteutettuna MATLAB-funktiona liitteestä C.

4.6 Integroivien prosessien viritykset AMIGOlla

Vastaavia AMIGO-sääntöjä voidaan käyttää myös integroiville prosesseille. Tällaisen prosessin kaava on

$$G_i(s) = \frac{K_v \exp(-ds)}{s}, K_v = \frac{K}{T}. \quad (26)$$

Tämäntyyppinen prosessi koostuu viiveestä ja lineaarisesta vasteen arvon noususta. Parametri K_v kuvaa vasteen kasvunopeutta (*velocity gain*) [9]. Integroivan prosessin parametrit pystytään määrittämään samalla periaatteella askelvastekokeella kuin FOTD-parametrit.

AMIGOon perustuva PI-säätimen virityssääntö integroiville prosesseille on kaavakokoelma

$$\begin{cases} K_p = \frac{0.35}{K_v L} \\ T_i = 13.35L, \end{cases} \quad (27)$$

jonka perustelu löytyy määritettynä lähteestä [11]. Periaate siinä on täysin FOTD-sääntöjen määrittästä vastaava. PID-säätimen viritykset saadaan integroiville prosesseille kaavasta

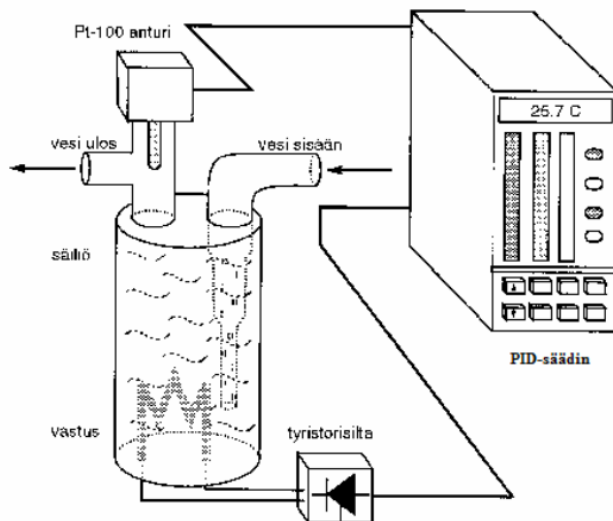
$$\begin{cases} K_p = \frac{0.45}{K_v} \\ T_i = 8L \\ T_d = 0.5L. \end{cases} \quad (28)$$

Tämän kokoelman perustelu löytyy myös lähteestä [11]. Näiden viritykset laskentafunktiot löytyvät liitteestä B. Käytännön säätöratkaisuihin integroiville prosesseille ei tässä tutkielmassa perehdytä, mutta laskusäännöt on kattavuuden takia hyvä dokumentoida.

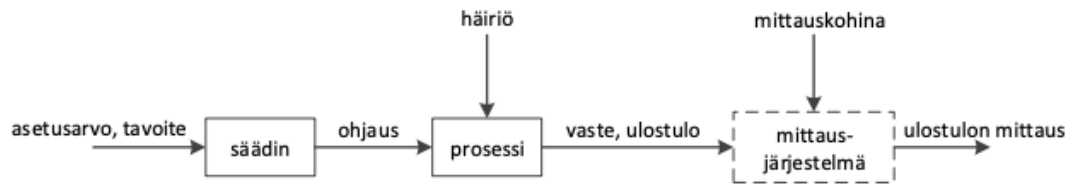
5. LÄPIVIRTAUSSÄILIÖN VEDEN LÄMPÖTILAN SÄÄTÖ

5.1 Lämmitysprosessin askelkoe

PID-säätimen virittämistä testattiin käytännön kokeella veden lämpötilan säätämisessä. Koejärjestelmän komponentit ovat kuvan 11 rakennekaaviossa. Järjestelmä koostuu vesisäiliöstä, johon virtaa noin litra minuutissa kylmää vettä hanasta ja josta lämmitetty vesi poistuu sen ulostuloputkesta. Veden lämpötilaa mitataan Pt100-anturilla, joka on kytketty PID-säätimeen. PID-säädin säätelee säiliössä olevan lämmitysvastuksen tehoa, ja säätimen ja vastuksen välissä on tyristorisilta. Kuvassa 12 on esitettyä avoimen järjestelmän lohko-kaavio, jossa säädin on luonnollisesti PID-säädin, prosessi lämmitettävä vesisäiliö sisään- ja ulosvirtauksineen ja mittausjärjestelmä Pt100-anturi.



Kuva 11. Veden jatkuvatoimisen lämmityksen laitteiston rakennekaavio [12, s.1].



Kuva 12. Avoimen järjestelmän lohkokkaavio [13, s. 2].

Järjestelmän askelkoe toteutettiin nostamalla avoimen järjestelmän askelmaisen ohjauksen asetuservo 25 %:sta 50 %:iin. Kyseinen muutos ohjauksessa vastaa 750 W:n tehon muutosta, eli lämmitysteho nostettiin 750 W:sta 1500 W:iin. Askelvastekokeen tulos on kuvassa 13. Siinä musta viiva kuvaa sisäänmenosignaalia prosenteissa ja punainen tankista ulosvirtaavan veden lämpötilaa. Vaaka-akselilla on aika minuuteissa ja sekunneissa.

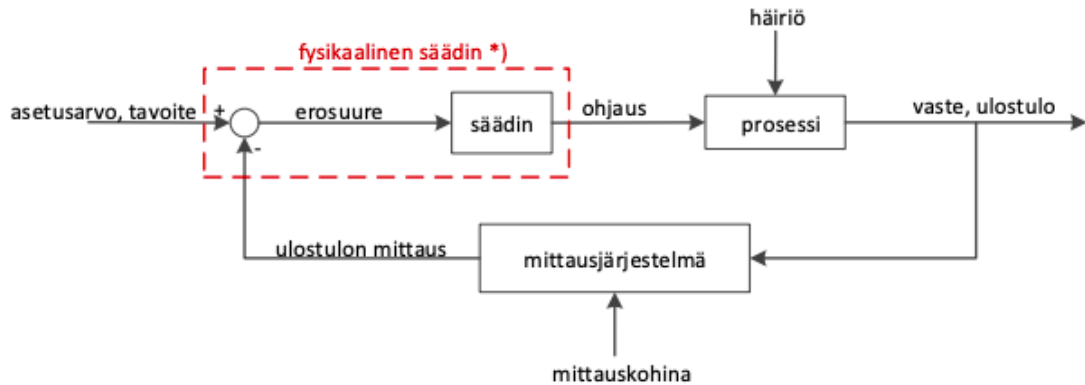


Kuva 13. Lämmitysprosessin askelvaste, kun ohjaus muutettiin 25 %:sta 50 %:iin.

Vasteeseen sovitettiin FOTD-parametrit silmämääräisesti. Lämpötila nousee 32 asteesta 64 asteeseen, eli K :n arvo on $(62\text{ }^{\circ}\text{C} - 32\text{ }^{\circ}\text{C})/750\text{ W} \approx 0.0427\text{ }^{\circ}\text{C}/\text{W}$. Viive ohjauksen lähettämistä vasteen reaktioon on 8 sekuntia ja aikavakio on noin 52 sekuntia. Näillä arvoilla voidaan laskea PI(D)-parametrit käyttämällä liitteen C AMIGO-funktioita. Laskematta prosessin normalisoitu viive tulokseksi saadaan $\tau = 8\text{ s}/(8\text{ s} + 52\text{ s}) = 0,133\text{}$

5.2 Veden lämpötilan takaisinkytketty säätö

PI(D)-säätöä varten järjestelmä muutettiin takaisinkytketyksi, ks. kuva 14. Laskemalla AMIGO-parametrit määritetyillä FOTD-parametreilla saadaan PI-säätöparametreiksi $K_p = 2.2876e + 03$ ja $T_i = 37.3305$. Tässä vaiheessa laboratoriokokeessa todettiin, että proportionaalivahvistuksen arvo on aivan liian suuri.

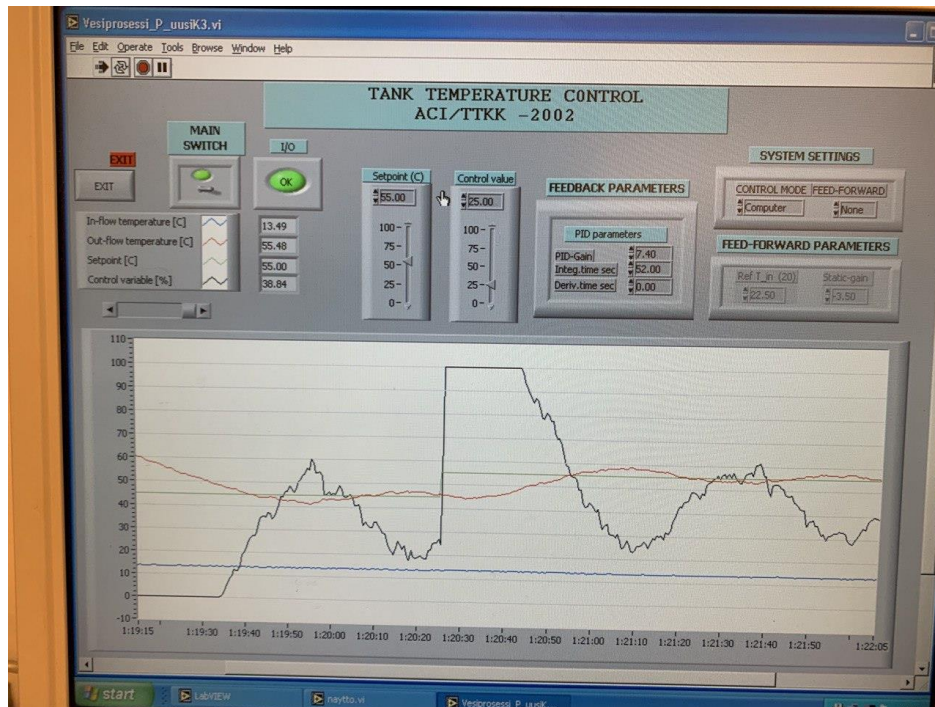


Kuva 14. Negatiivisesti takaisinkytketyn järjestelmän lohkokaavio [12, s. 2].

Huonon parametrisoinnin selittää kappaleessa 4.4 läpikäyty AMIGOn rajoitteet. Laskemalla normalisoitu viive on liian matala, että AMIGO tuottaisi hyviä säätötuloksia. Proportionaalivahvistus on niin suuri, että ei ole kannattavaa edes kokeilla sitä tälle systeemille, sillä säätötulos olisi epästabiili. On käytettävä jotain toista ja paremmin soveltuvaa menetelmää. Tähän prosessiin sopivammaksi säätömenelmäksi osoittautui lambdaviritys, *lambda tuning* [1, s. 186]. Menetelmä perustuu takaisinkytketyn säätöpiirin tavoiteaikavakion T_{cl} valintaan ja prosessimalliin. Lambdavirityksen PI-säännöt ovat esitelty kaavaparissa

$$\begin{cases} K_p = \frac{1}{K} \frac{T}{L + T_{cl}} \\ T_i = T. \end{cases} \quad (29)$$

Parametri T_{cl} on virityksen suunnitteluparametri. Yhtenä vaihtoehtona T_{cl} :n valintaan on, että se on kolme kertaa aikavakion suuruinen, $T_{cl} = 3T$, eli tässä tapauksessa $T_{cl} = 3 * 52 s = 156 s$. Hyvin parametrisoidulle mallille aggressiivisen säädön saisi valinnalla $T_{cl} = T$. Lambda-kaavaparilla PI-parametreiksi saadaan $K_p = 7,4$ ja $T_i = 52$. Näitä parametreja käytettiin vedenlämmityksen säätöön ja saatiin vasteeksi kuvan 15 tulos.

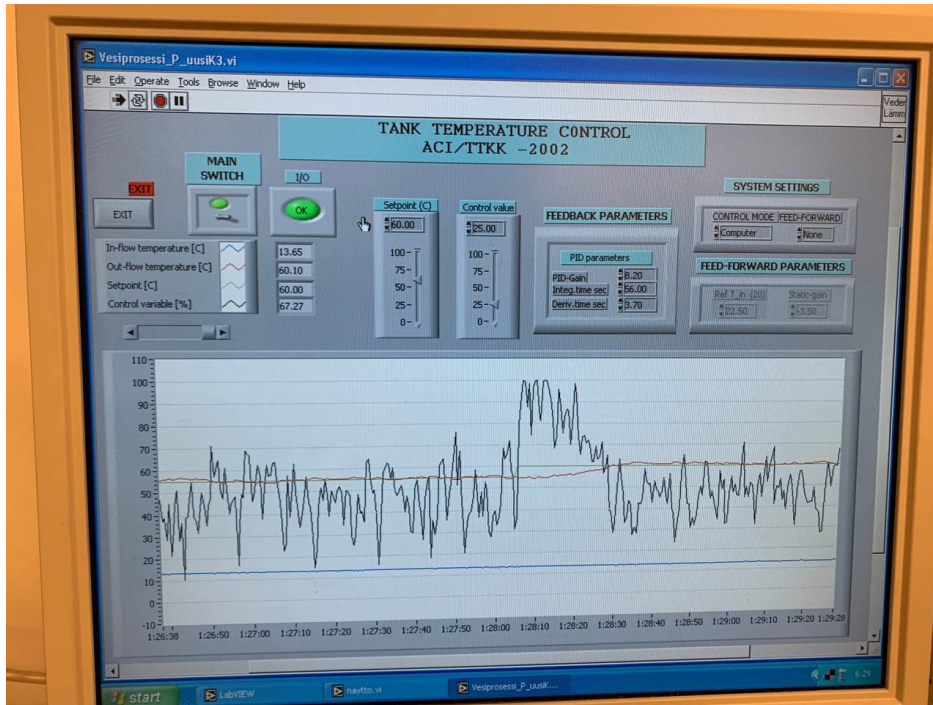


Kuva 15. PI-säädetyin prosessin vaste, kun tavoitearvoa on kasvatettu 10 astetta.

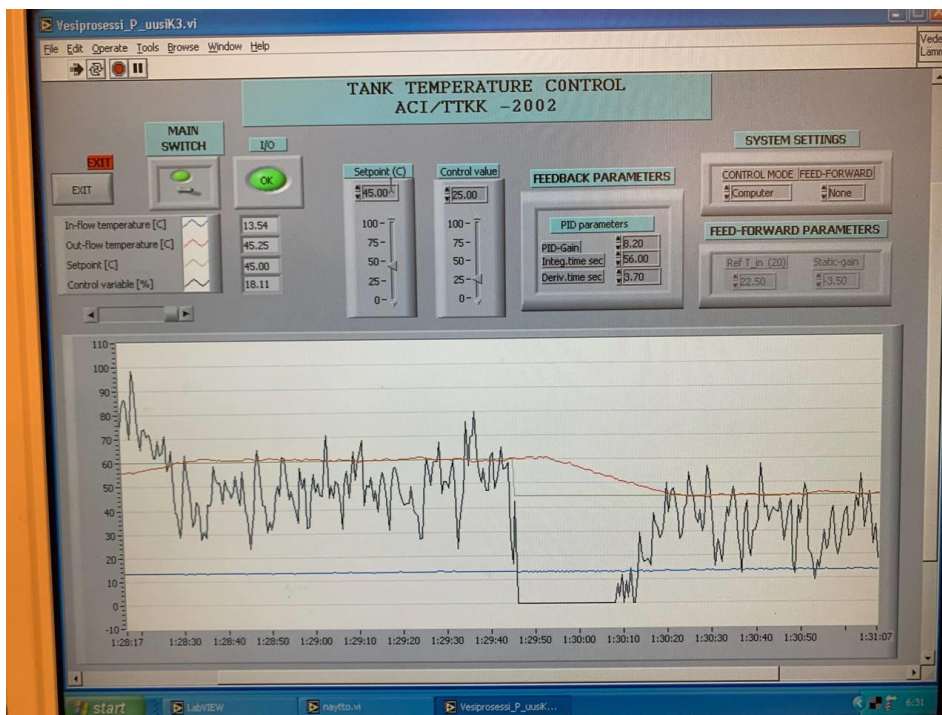
Kuvassa 15 vihreä viiva kuvaa tavoitearvoa, joka nostetaan 45 asteesta 55 asteeseen, musta viiva on säätimen tuottama ohjaus ja punainen viiva on putkesta poistuvan veden lämpötila. Vaste seuraa kohtuullisen hyvin tavoitearvoa, mutta jää lievästi värähtelemään. Lisäämällä säätimeen derivointiparametrin säätötulosta voi kehittää paremmaksi. Lambdavirityksen säännöt PID-säätimelle on esitelty kaavakokoelmassa

$$\begin{cases} K_p = \frac{1}{K} \frac{L}{\frac{L}{2} + T} \\ T_i = T + \frac{L}{2} \\ T_d = \frac{TL}{L + 2T} \end{cases} \quad (30)$$

Käyttämällä näitä sääntöjä saadaan PID-parametreiksi $K_p = 8,2$, $T_i = 56$ ja $T_d = 3,7$. Tämän säädön tulokset ovat kuvissa 16 ja 17, joissa tavoitearvoa oli nostettu ensin 5 astetta ja sitten laskettu 15 astetta.



Kuva 16. PID-säädetyin prosessin vaste, kun tavoitearvoa on kasvatettu 5 astetta.



Kuva 17. PID-säädetyin prosessin vaste, kun tavoitearvoa on laskettu 15 astetta.

Kuvista näkee, että säätötulos paranee merkittävästi. Värähtelyä tavoitearvon ympärillä ei ole käytännössä ollenkaan lukuun ottamatta kevyttä mittauskohinaa. Derivointihaara aiheuttaa ohjaukseen hyvin terävää ja nopeaa vaihtelua, mikä ei tässä kuitenkaan ole haitallista, koska säätimellä ohjataan lämmitysvastuksia. Piikikkäästi vaihteleva ohjaus

lisäksi suodattuu itse lämmitysprosessissa. Sääto toimii erinomaisesti myös tilanteeseen, jossa tavoitearvoa lasketaan. Tämä on säätötekniisesti vaikeampi tilanne, sillä säätimen pitää kyetä ennustamaan koska veden lämmitys pitää aloittaa uudestaan ilman, että lämpötilaan tulee alitusta.

Vaikka AMIGO-iritys ei tähän prosessiin sopinutkaan, saatiin aikaiseksi mallikas säätötulos vedenlämmitysprosessille. AMIGOn toimivuus olisi vaatinut systeemin, jossa normalisoitu viive olisi suurempi eli viive olisi suurempi tai aikavakio pienempi. Lambdairitys ei ole absoluuttisesti AMIGOa parempi iritysmenetelmä, vaan se sopii tietyn tyyppisten prosessien säätöön. Tällä iritysmenetelmällä hitaat järjestelmät ovat herkkiä kuormitushäiriölle, sillä yksi prosessimallin napa kumotaan säätimen nollalla [1, s. 189].

6. YHTEENVETO

Monotoniseen askelvasteeseen, jossa on viivettä, voidaan helposti sovittaa sekä FOTD-että SOTD-mallien parametrit. Sovittamalla saatavat siirtofunktiot vastaavat alkuperäistä mallia niin hyvin, että prosessia voidaan säätää alemman kertaluokan mallilla. SOTD-malli on tyypillisesti tarkempi, koska siinä on kaksi aikavakiota yhden sijaa, mistä johtuen se mahdollistaa kompleksisemmän mallireduktion, jossa on FOTD-mallin tasaisen nousun sijaan s-muotoisuutta. Mallireduktiota kokeiltiin viiveelliselle ja viiveettömälle neljännen asteen siirtofunktiolle, joista molemmille sovittaminen onnistui todella hyvin.

ZN-menetelmästä kehitetty MIGO laskee PID-säätimelle sellaiset viritysparametrit, että takaisinkytketty järjestelmä kestää häiriötä hyvin. AMIGO, eli approksimoiva MIGO antaa samankaltaiset parametrit yksinkertaisella kaavakokoelmalla. MIGO- ja AMIGO-viritysmenetelmien toimivuus on rajoittunut normalisoidun viiveen $\tau = L/(L + T)$ arvoon, eikä hyviä säätötuloksia saada jos $\tau < 0,3$. Tämä rajoite huomattiin käytännössä luvun 6 laboratorionkokeessa, missä AMIGO osoittautui käyttökelvottomaksi vedenlämmitysprosessin matalan viiveen ja aikavakion vuoksi. Matalilla normalisoidun viiveen arvoilla paremmaksi viritysmenetelmäksi osoittautui lambdaviritys, joka antoi erittäin hyvä asetusarvovasteen veden lämmitysprosessin lämpötilan muutoksille. Normalisoidun viiveen antaman vaatimuksen täytyessä AMIGO antaa kuitenkin robustin ja häiriötä hyvin sietävän säädön.

LÄHTEET

- [1] K. J. Åström ja T. Hägglund, *Advanced PID Control*, Isa, 2006.
- [2] R. Dorf, ja R. Bishop, *Modern Control Systems*, Thirteenth edition, Harlow: Pearson, 201, Print.
- [3] G. Silva, D. Aniruddh ja S.P. Bhattacharyya, "PI Stabilization of First-Order Systems with Time Delay." *Automatica (Oxford)* 37.12 (2001): 2025–2031, Web.
- [4] K. J. Åström ja T. Hägglund, *PID Controllers: Theory, Design, and Tuning*. N.p., 1995, Print.
- [5] V.-P. Pyrhönen, *Johdatus systeemien hallintaan: Luento 6*, Tampereen yliopisto, Tampere, 2021.
- [6] K. J. Åström ja R. Murrey, *Feedback Systems : an Introduction for Scientists and Engineers*, Princeton, NJ: Princeton University Press, 2008, Print.
- [7] G. Ellis, *Control System Design Guide: Using Your Computer to Understand and Diagnose Feedback Controllers*, St. Louis: Elsevier Science & Technology, 2012, Print.
- [8] J.G Ziegler ja N.B. Nichols, "Optimum Settings for Automatic Controllers." *Journal of dynamic systems, measurement, and control* 115.2 B (1993): 220–222, Web.
- [9] T. Hägglund ja K. J. Åström, "Revisiting the Ziegler–Nichols Step Response Method for PID Control." *Journal of process control* 14.6 (2004): 635–650, Web.
- [10] K. J. Åström, H. Panagopoulos ja T. Hägglund, "Design of PI Controllers Based on Non-Convex Optimization." *Automatica (Oxford)* 34.5 (1998): 585–601, Web.
- [11] T. Hägglund ja K. J. Åström, "REVISITING THE ZIEGLER-NICHOLS TUNING RULES FOR PI CONTROL - PART II THE FREQUENCY RESPONSE METHOD." *Asian journal of control* 6.4 (2004): 469–482, Web.
- [12] ASE-1130 Automaatio: Laboratoriotyö, Veden lämpötilan säätö, Tampereen teknillinen yliopisto, Tampere, 2014
- [13] V.-P. Pyrhönen, *Automaatio: Laboratoriotyö*, Tampereen yliopisto, Tampere, 2020

LIITE A: MATLAB, FOTD-PARAMETRIEN LASKENTAFUNKTIO

```

function [K,tau_a, d_a] = fofd(G)
%FOFD Funktio laskee annetulle siirtofunktiolle FOTD-sovitteen parametrit.
if(isstable(G))
    K = dcgain(G); S = allmargin(G);
    [GY, GT] = step(G);

    %Määritetään hetki jolla funktio on saavuttanut 63.2 % loppuarvostaan
    for i = 1 : length(GT)
        if GY(i) > 0.632*K
            %Otetaan 63.2% arvoa yhtä pienempi ja suurempi arvo
            opt = [GY(i-1), GY(i)];
            break
        end
    end
    %Valitaan lähempi arvo
    if abs(opt(1)-0.632*K) < abs(opt(2)-0.632*K)
        T_ar = GT(i-1);
    else
        T_ar = GT(i);
    end

    %Derivointi
    dy = diff(GY)./diff(GT);

    %Jyrkimmän kohdan indeksi
    [~,idx] = max(dy);

    %Tangentti
    tang = (GT-GT(idx))*dy(idx)+GY(idx);

    %Etsitään tangentin indeksi, jonka arvo on lähimpänä 0:aa
    for j = 1: length(tang)
        if tang(j) > 0
            d_a_idx = [tang(j-1), tang(j)];
            break
        end
    end
    %Valitaan viiveelle lähin arvo
    if abs(d_a_idx(1)) < abs(d_a_idx(2))
        d_a = GT(j-1);
    else
        d_a = GT(j);
    end

    %Lasketaan aikavakio
    tau_a = T_ar-d_a;

end
end

```

LIITE B: MATLAB, SOTD-PARAMETRIEN LASKENTAFUNKTIO

```

function [tau1, tau2, K, d] = sotd(G)
%SOTD Funktio laskee annetulle siirtofunktiolle SOTD-sovitteen parametrit.
%Toleranssiparametri
tolerance = 0.01
%Tarkistetaan, että siirtofunktio on stabiili
if(isstable(G))
    s = tf('s');
    K = dcgain(G);
    [GY, GT] = step(G);

    %Määritetään hetki jolla funktio on saavuttanut 63.2 % loppuarvostaan
    for i = 1 : length(GT)
        if GY(i) > 0.632*K
            %Otetaan talteen 63.2% arvoa yhtä pienempi ja yhtä suurempi
            %arvo
            opt = [GY(i-1), GY(i)];
            break
        end
    end
    %Valitaan lähempi arvo
    if abs(opt(1)-0.632*K) < abs(opt(2)-0.632*K)
        t63 = GT(i-1);
    else
        t63 = GT(i);
    end
    S = allmargin(G);
    Kkr = S.GainMargin(1);
    wkr = S.GMFrequency(1);

    syms tau1 tau2 d
    %Iteroidaan parametreille arvot
    for a = 0.01:0.01:1
        tau1 = (1/(a*wkr*sqrt(2)))*sqrt(sqrt(4*a^2*K^2*Kkr^2+(1-a^2)^2)-1-a^2);
        tau2 = a*tau1;
        d = (pi-atan(wkr*tau1)-atan(wkr*tau2))/(wkr);

        %Lasketaan askelvaste näillä parametreilla
        if(a ~= 1)
            f = 0.368-(1/(1-a))*exp((d-t63)/tau1)-(a/(a-1))*exp((d-t63)/(tau1*a));
            if(abs(f) < tolerance)
                break
            end
        else
            f = 0.368-exp(-(t63-d)/tau1)-(t63/tau1)*exp(-(t63-d)/tau1)-0.63;
            if(abs(f) < tolerance)
                break
            end
        end
    end
end
end

```

LIITE C: MATLAB, PID:N AUTOMAATTIVIRITYSFUNKTIOT

```
function [Kp, Ti] = PI_FOTD(K, tau, d)
%Funktio määrittää PI-säätimen parametrit FOTD-mallille
Kp = (0.15/K)+(0.35-(d*tau/(d+tau)^2))*(tau/K*d);
Ti = 0.35*d + 13*d*tau^2/(tau^2+12*d*tau+7*d^2);
end
```

```
function [Kp, Ti, Td] = PID_FOTD(K,tau,d)
%Funktio määrittää PID-säätimen parametrit FOTD-mallille
Kp = 1/K*(0.2+0.45*(tau/d));
Ti = (0.4*d+0.8*tau)/(d+0.1*tau)*d;
Td = (0.5*d*tau)/(0.3*d+tau);
end
```

```
function [Kp, Ki, Kd] = PID_SOTD(K, tau1, tau2, d)
%Funktio määrittää PID-parametrit SOTD-mallista

alfa1 = 0.19; alfa2 = 0.37; alfa3 = 0.18; alfa4 = 0.02;
beta1 = 0.48; beta2 = 0.03; beta3 = -0.0007; beta4 = 0.0012;
gamma1 = 0.29; gamma2 = 0.16; gamma3 = 0.20; gamma4 = 0.28;

Kp = (1/K)*(alfa1 + alfa2*(tau1/d)+alfa3*(tau2/d)+alfa4*(tau1*tau2/d^2));
Ki = (1/K)*(beta1*(1/d)+beta2*(tau1/d^2)+beta3*(tau2/d^2)+beta4*(tau1*tau2)/d^3);
Kd =
(1/K)*(gamma1*d+gamma2*tau1+gamma3*tau2+gamma4*(tau1*tau2/d))*((tau1+tau2)/(tau1+tau2+d));
end
```