

Antti Kangasrääsio

Adaptiivisen säädön menetelmät

Elektroniikan, tietoliikenteen ja automaation tiedekunta

Kandidaatintyö
Espoo 6.12.2010

Vastuupettaja:

TkT Ilkka Seilonen

Työn ohjaaja:

DI Petri Hänninen



Aalto-yliopisto
Teknillinen korkeakoulu

Tekijä: Antti Kangasrääsiö

Työn nimi: Adaptiivisen säädön menetelmät

Päivämäärä: 6.12.2010

Kieli: Suomi

Sivumäärä:6+25

Tutkinto-ohjelma: Automaatio- ja systeemitekniikka

Vastuupettaja: TkT Ilkka Seilonen

Ohjaaja: DI Petri Hänninen

Adaptiivisen säädön menetelmäkenttä on varsin laaja. Tämä työ antaa yleiskatsauksen siihen, mitä on adaptiivinen säätö, mitä haasteita siinä on ja millä keinoilla sitä voidaan toteuttaa. Työssä käydään aluksi lyhyesti läpi adaptiivisen säädön historia ja tehdään lyhyt katsaus adaptiiviseen säätöön olennaisesti liittyviin käsitteisiin. Jäljempänä esitellään hieman laajemmin rekursiivinen pienimmän neliösumman menetelmä, mallireferenssisäädin sekä itsevirittyvä säädin. Muista merkittävistä menetelmistä esitellään lyhyesti menetelmän idea. Työn lopuksi esitellään toteutus gain scheduling -algoritmista ja kommentoidaan sen toimintaa.

Avainsanat: adaptiivinen säätö, rekursiivinen identifiointi, rekursiivinen PNS, mallireferenssisäätö, itsevirittyvät säätimet, gain scheduling

Author: Antti Kangasrääsio

Title: Methods of adaptive control

Date: 6.12.2010

Language: Finnish

Number of pages:6+25

Degree program: Automation and systems technology

Supervisor: D.Sc. Ilkka Seilonen

Instructor: M.Sc. Petri Hänninen

Many different methods exist in the field of adaptive control. This bachelor's thesis gives a quick overview of adaptive control – what is it, what challenges does it have and how does it work. First a short overview of the history of adaptive control is presented, followed by a short introduction to some core concepts related to adaptive control. Recursive least squares method, model reference control and self-tuning regulators are analyzed more thoroughly, while with other methods only the general idea is presented. Finally, an implementation of the gain scheduling algorithm is presented and briefly analyzed.

Keywords: adaptive control, recursive identification, recursive least squares method, model reference adaptive control, self-tuning regulators, gain scheduling

Esipuhe

Älykkyys ja sen luominen on aina ollut kiinnostava aihealue. Mitä älykkäältä järjestelmältä vaaditaan? Miten älykkyyttä voidaan mallintaa? Mitä älykkyys oikeastaan on? Päätin perehtyä tähän aiheeseen automaatioon liittyvän adaptiivisen säädön kautta. Nyt asiaan perehdyttyäni olen saanut paljon uusia näkökulmia ja uusia kysymyksiä on herännyt. Mitä muita tapoja mallintaa älykkyyttä on? Mitä sovelluksia korkeamman tasoisella älykkyydellä voisi olla automaatiossa? Mitä korkeampitasoinen älykkyys oikeastaan on?

Tämän kandidaatintyön tekeminen on ollut opettavainen kokemus. Aiemmillä kursseilla asiat tuntuivat välillä etäisiltä ja abstrakteilta, eikä niiden välillä ollut välttämättä selviä yhteyksiä. Nyt kun pääsin perehtymään tähän aiheeseen tarkemmin alan kirjallisuuden kautta, ovat yhteydet avautuneet selvemmin.

Kiitän professori Heikki Koivoa mahdollisuudesta tehdä tämä kandidaatintyö oman aihevalintani mukaan, sekä työni valvojaa DI Petri Hännistä asiantuntevasta ohjauksesta kaiken kiireen keskellä. Erityiskiitos isälleni huomattavasta avusta työn oikolukemisessa.

Liittyen näihin kolmeen ja puoleen vuoteen Otaniemessä kiitän Automaatio- ja systeemitekniikan kiltaa ja tuttujani siellä ystävydestä, henkisestä tuesta ja hienoista tapahtumista. Olette parhautta. Kiitän tyttöystävääni Elliä siitä, että olet jaksanut kanssani, vaikka olen usein ollut enemmän killan tapahtumissa kuin kotona. Kuitenkin ilman sinua ja tukeasi en olisi varmasti pysynyt järjissäni. Kiitän myös perhettäni Kuopiossa kaikesta siitä tuesta mitä olen teiltä saanut. Ilman teitä olisin varmasti vararikossa.

Otaniemi, 6.12.2010

Antti O. Kangasrääsiö

Sisältö

Tiivistelmä	ii
Tiivistelmä (englanniksi)	iii
Esipuhe	iv
Sisällysluettelo	v
Määritelmät	vi
1 Johdanto	1
2 Adaptiivisen säädön tausta ja historia	3
3 Yleisiä ominaisuuksia ja haasteita	4
3.1 Säätimen jakaminen osiin	4
3.2 Epäsuora ja suora adaptiivinen säätö	4
3.3 Stabiilisuus	5
3.4 Robustisuus	5
4 Rekursiivinen identifiointi	7
4.1 Yleistä	7
4.2 Rekursiivinen PNS -menetelmä	8
5 Mallireferenssisäätimet	11
5.1 Yleistä	11
5.2 Säätoparametrien päivityssäännöt	12
6 Itsevirittyvät säätimet	14
6.1 Yleistä	14
6.2 Säätoparametrien päivitysmenetelmät	14
7 Muita menetelmiä	17
7.1 Expert system -menetelmät	17
7.2 Stokastiset menetelmät	18
7.3 Gain scheduling	19
8 Sovellusesimerkki: gain scheduling	20
9 Yhteenveto	23
Viitteet	24
A Sovellusesimerkin alijärjestelmät	25

Määritelmät

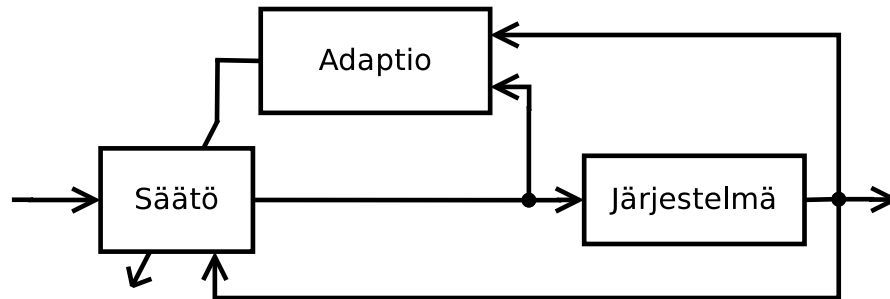
Lyhenteet

APPC	Adaptiivinen napojen asettelu (<i>adaptive pole placement control</i>)
BIBO	Rajoitettu tulo rajoitettu lähtö -stabiilisuus (<i>bounded input bounded output stability</i>)
BIBS	Rajoitettu tulo rajoitetut tilat -stabiilisuus (<i>bounded input bounded state stability</i>)
IV	Instrumenttimuuttujamenetelmä (<i>instrumental variable method</i>)
MRAC	Mallireferenssisäädin (<i>model reference adaptive control</i>)
MV	Minimivarianssisäädin (<i>minimum variance control</i>)
PID	Erosuureen arvoon, integraaliin ja derivaattaan perustuva säädin (<i>proportional-integral-derivative controller</i>)
PNS	Pienimmän neliösumman menetelmä (<i>least squares method</i>)
RLS	Rekursiivinen PNS -menetelmä (<i>recursive least squares method</i>)
SISO	Järjestelmä, jossa on yksi tulo ja yksi lähtö (<i>single input single output system</i>)
STR	Itsevirittyvä säädin (<i>self-tuning regulator</i>)

Merkinnät

$y(t), Y(s)$	Järjestelmän lähtö(vektori) ja sen Laplace-muunnos (<i>output</i>)
$u(t), U(s)$	Järjestelmän ohjaus(vektori) ja sen Laplace-muunnos (<i>input</i>)
$r(t)$	Järjestelmän referenssisignaali (<i>reference</i>)
$\hat{\theta}(t)$	Järjestelmän estimoitu parametrivektori
$\theta^*(t)$	Järjestelmän todellinen parametrivektori
$\pi(t)$	Säätimen parametrivektori
$\phi(t)$	Järjestelmän historiavektori, joka sisältää sekä y :n että u :n oleellisen historian ajanhetkellä t
$v(t)$	Häiriösignaali
V	Virhefunktio (<i>loss function</i>)
L	Lyapunov-funktio
t, τ	Aikamuuttujia
s	Laplace-avaruuden kompleksimuuttuja
\square^T	Matriisin/vektorin transpoosi
\square^{-1}	Käänteismatriisi
$ \square $	Itseisarvo
$\ \square\ $	Normi
\square	Aikaderivaatta
∇	Gradienttioperaattori

1 Johdanto



Kuva 1: Havainnekuva adaptiivisesta säädöstä

Työn taustaa

Adaptiolla tarkoitetaan sopeutumista kulloinkin vallitseviin olosuhteisiin. Säädöllä taas tarkoitetaan niitä toimia, joilla jokin järjestelmä saadaan toimimaan halutulla tavalla. Adaptiivinen säätö tarkoittaa siis säätöä, joka ottaa jollain tavalla huomioon muutokset säädettävässä järjestelmässä, ja reagoi niihin tarkoituksenmukaisella tavalla. Matemaattisemmin voisi sanoa, että adaptiivinen säädin identifioi joitain säädettävän järjestelmän parametreja, ja käyttää tätä tietoa säätimen parametrien uudelleenmäärittämiseksi.

Adaptiivisen säädön määritelmästä on käyty paljon keskustelua. Vaikka myös normaalit ja robustit säätimet kykenevät säätämään järjestelmiä, joiden parametrit vaihtelevat, ei niitä silti luokitella adaptiivisiksi. Verrattuna tavalliseen tai robustiin säätöön, adaptiivinen säätö tarkoittaa sitä, että säätimessä on muutettavia parametreja ja mekanismi näiden parametrien muuttamiseksi. Adaptiivisessa säätimessä on siis kaksi silmukkaa: prosessin ja säätimen välinen normaali takaisinkytkentä sekä identifioitujen parametrien kautta tapahtuva säätimen parametrien säätö. Tätä on havainnollistettu kuvassa 1. [1, s. 1] [2, s. 23] [3, s. 1-2]

Adaptiiviseen säätöön liittyvät toisaalta läheisesti myös älykäs, oppiva ja itseorganisoituva säätö. Nämä säätötekniikat ottavat huomioon adaptiivista säätöä korkeamman tasoisia muutoksia, ja perustuvat usein tekoälypohjaisiin ratkaisuihin. Nämä tekniikat pystyvät tietysti myös toteuttamaan adaptiivista säätöä, mutta ne eivät ole tässä työssä keskeisessä asemassa. [4, s. 5-6]

Työn rajaus ja tavoitteet

Tämän kandidaatintyön tarkoituksena on luoda katsaus adaptiivisen säädön menetelmäkenttään, eli tehdä lyhyt yhteenveto yleisimmistä tavoista toteuttaa adaptiivista säätöä. Tämän lisäksi paneudutaan syvemmin muutamaan yleisimmin käytettyyn menetelmään. Identifiointimenetelmistä esitellään tarkemmin rekursiivinen

PNS (pienimmän neliösumman) -menetelmä, ja adaptiivisten säädinten osalta perehdytään mallireferenssisäätöön sekä itsevirittyviin säätimiin. Lisäksi työhön sisältyy lyhyt soveltava osio, jossa demonstroidaan gain scheduling -menetelmällä toteutettu yksinkertainen adaptiivinen säädin.

Tutkimusmenetelmä

Työn teoriaosa on tehty täysin kirjallisuustutkimuksena. Adaptiivista säätöä sekä siihen läheisesti liittyvää rekursiivista identifointia on ennen pidetty hajanaisena kokoelmana erinäisiä kikkoja ja teknisiä temppuja [2, s. 25] [5, s. 9]. Tästä syystä työssä on pyritty käyttämään erillisten artikkelien sijaan tunnettuja kokoelmateoksia, jotta pystyttäisiin tarjoamaan mahdollisimman yhtenäinen katsaus alaan.

Perusteoksista on valittu oppikirjamaisia sekä paljon viitattuja kirjoja, jotka käsittelevät adaptiivista säätöä ja rekursiivista identifointia. Lisäksi lähteinä on käytetty tiettyihin aihepiireihin keskittyneitä teoksia syventämään analyysiä tarvittavissa määrin.

Luettavuuden parantamiseksi on työssä yhdenmukaistettu eri lähteiden toisistaan poikkeavia merkintätapoja.

Työn rakenne

Aluksi tarkastellaan adaptiivisen säädön taustaa ja historiaa. Tämän jälkeen perehdytään tarkemmin adaptiivisen säädön yleisiin ominaisuuksiin ja haasteisiin. Työn jälkimmäisessä osassa käydään läpi tarkemmin adaptiivisen säädön eri menetelmiä ja esitellään esimerkkisovellus. Lopuksi tehdään yhteenveto.

2 Adaptiivisen säädön tausta ja historia

Motivaatio adaptiiviseen säätöön

Alunperin tarve adaptiiviseen säätöön syntyi, kun 1950-luvun alkupuolella alettiin tutkia keinoja rakentaa automaattisia ohjausjärjestelmiä uusia kehittyneitä lentokoneita varten. Nämä lentokoneet lensivät useilla eri korkeuksilla ja nopeuksilla, jolloin niiden dynaamiset ominaisuudet vaihtelivat merkittävästi riippuen toimintapisteestä. Huomattiin, että tavalliset vakiovahvisteiset lineaariset säätimet (*constant-gain linear feedback control*) toimivat kyllä tietyissä olosuhteissa, mutta eivät kaikissa. Tarvittiin siis kehittyneempi tapa ohjata lentokonetta, mikä ottaisi huomioon erilaiset olosuhdevaikutukset. [1, s. 2] [2, s. 23] [3, s. 2]

Adaptiivisen säädön kehityskaari

Ensimmäiset adaptiiviset säätöjärjestelmät olivat lentokoneen autopilotteja, jotka oli toteutettu gain scheduling -menetelmällä. 50- ja 60-lukujen vaihteessa alettiin tähän ongelmaan soveltaa mallireferenssisäätöä (*model reference adaptive control, MRAC*) sekä itsevirittyviä säätimiä (*self-tuning regulators, STR*). [2, s. 23] [3, s. 2]

60-luvulla saavutettiin adaptiivisen säädön kannalta useita merkittäviä tuloksia yleisen säätöteorian alalla. Näistä mainittakoot tilaesityksen ja stabiilisuusteorian kehittäminen, merkittävät edistykset stokastisen säätöteorian sekä identifioinnin alalla, sekä kaksoissäädön (*dual control*) ja dynaamisen ohjelmoinnin kehittyminen. [1, s. 2] [2, s. 24] [3, s. 2-3]

70-luvulla saatiin viimein todistettua useiden adaptiivisten menetelmien stabiilisuus, joskin vahvoin oletuksin. 80-luvun alussa kuitenkin osoitettiin, että 70-luvulla stabiileiksi todetut järjestelmät eivät olleet kuitenkaan robusteja, eli ne saattoivat menettää stabiilisuutensa häiriöiden tai mallintamattoman dynamiikan takia. Tämä taas johti runsaaseen tutkimukseen adaptiivisten säätimien stabiilisuuden ja stabiilisuusehtojen alalla. Tämän tutkimuksen pohjalta kehitettiin 80-luvun keskivaiheilla robusti adaptiivinen säätö. [1, s. 2-3] [2, s. 24-25] [3, s. 3]

90-luvun puoliväliin mennessä oli menetelmien robustisuutta kehitetty huomattavasti. Myös lineaarisen aikariippuvan prosessin säätäminen saatiin vihdoinkin ratkaistua. Lisäksi epälineaaristen järjestelmien tutkimuksen edistymisen auttoi ymmärtämään adaptiivisia järjestelmiä vieläkin paremmin. Adaptiivisen säädön menetelmiä kyettiin laajentamaan myös tiettyjen epälineaaristen prosessien säätämiseen. [1, s. 3] [2, s. 25]

90-luvun alusta lähtien tutkimuskohteina ovat olleet muun muassa neuroverkkojen soveltaminen adaptiiviseen säätöön, sekä sellaiset menetelmät, jotka eivät nojautu samalla tavoin identifointiin kuin nykyiset järjestelmät. Tämän tyyppisten säätimien tutkimus jatkuu edelleen. [6, s. 10-11]

3 Yleisiä ominaisuuksia ja haasteita

3.1 Säätimen jakaminen osiin

Adaptiivinen säädin koostuu yleisesti ottaen kahdesta osasta: identifioivasta ja adaptiivisesta osasta. Ehkä selvimmän jako näkyy itsevirittyvissä säätimissä. Säätimen identifioiva osa tekee havaintoja järjestelmästä, ja välittää nämä havainnot eteenpäin adaptiiviselle osalle. Adaptiivinen osa taas päättelee järjestelmästä saadun informaation perusteella miten järjestelmää tulisi säätää. Se, miten nämä kaksi osaa on toteutettu, ja millä tavalla ne on integroitu toisiinsa vaihtelee eri menetelmissä.

Adaptiivisessa säädössä käytetään usein niin sanottua varmuusperiaatetta (*certainity equivalence principle*), joka tarkoittaa sitä, että säätimen adaptiivinen osa ei ota huomioon järjestelmästä estimoitujen parametrien virheitä, vaan käyttää estimoituja parametreja kuin ne olisivat järjestelmän tarkat parametrit. Tämä mahdollistaa sen, että säätimen eri osat voidaan suunnitella toisistaan riippumatta. Tällöin suunnittelukustannukset ovat todennäköisesti myös pienemmät, koska huomioon otettavia asioita on vähemmän. Varmuusperiaatteessa on kuitenkin se huono puoli, ettei parametrien estimointivirheitä voida käyttää hyödyksi säädössä.

On kuitenkin myös mahdollista ottaa estimointivirheet huomioon säädössä, ja jopa pyrkiä pitämään ne pieninä. Eräs ratkaisu tähän on epälineaarisen stokastisen säätöteorian pohjalta kehitetty kaksoissäätö (*dual control*), joka pyrkii yhtä aikaa hyvään säätötulokseen ja pitämään estimointivirheet mahdollisimman pieninä. Kaksoissäädön suunnittelu on kuitenkin huomattavan monimutkaista, ja siinä vastaan tulevat matemaattiset ongelmat ovat usein ratkaistavissa vain numeerisesti. Aina ne eivät ole edes ratkaistavissa. Stokastisesta säädöstä kerrotaan tarkemmin luvussa 7.2. [1, s. 22-24]

3.2 Epäsuora ja suora adaptiivinen säätö

Usein adaptiiviset säätimet voidaan jakaa epäsuoriin (*indirect, explicit*) ja suoriin (*direct, implicit*) adaptiivisiin säätimiin. Tämä jako perustuu säätimen suunnitteluideologiaan.

Epäsuorat säätimet on suunniteltu siten, että säädin identifioi säädettävän järjestelmän luonnollisia parametreja. Esimerkiksi lineaarisen järjestelmän tapauksessa siirtofunktion osoittajan ja nimittäjän polynomien kertoimien arvoja. Tämän jälkeen säädin käyttää näitä järjestelmäparametreja laskeakseen säätimen parametrien arvot. Tämä tarkoittaa sitä, että säädin joutuu suorittamaan identifioinnin lisäksi ylimääräisiä laskentaoperaatioita. [2, s. 9]

Suorat säätimet taas on suunniteltu siten, että säädin identifioi järjestelmää, joka on johdettu säädettävästä järjestelmästä parametrisoimalla se säätimen parametreilla. Säädettävä järjestelmä on siis sama, identifikaatio on vain suunniteltu eri tavalla. Tällöin saadut identifioidut parametrit ovat suoraan säätimen uudet parametrit. Suorien adaptiivisten säätimien suunnittelu on monimutkaisien järjestelmien tapauksessa hankalaa, ja ne soveltuvatkin lähinnä lineaaristen aikariippumattomien SISO (*single input single output*)-prosessien säätöön. [2, s. 9-10]

3.3 Stabiilisuus

Stabiili järjestelmä kykenee toimimaan hyväksyttävällä tavalla sen kohdatessa häiriöitä toimintapisteessään. Käytännössä tämä tarkoittaa sitä, että järjestelmä palaa joko luonnostaan tai säädön vaikutuksesta takaisin toimintapisteeseensä. Stabiilisuudelle on olemassa erilaisia formaaleja määritelmiä, kuten Lyapunov-stabiilisuus.

Lyapunov-stabiilisuuden määritelmän oletukset eivät kuitenkaan aina toteudu adaptiivisessa säädössä. Eräs vaihtoehtoinen lähtökohta on vaatia järjestelmän, identifikaattorin ja säätimen tilojen pysyminen rajoitettuna kaikilla rajoitetuilla tuloilla, eli BIBS (*bounded input bounded state*)-stabiilisuus. [3, s. 130]

Varsinkin robustin adaptiivisen säädön tapauksessa osoittautuu, että edellisiä parempi vaihtoehto stabiilisuuden määrittämiseksi on kuitenkin stabiilisuus L_p -normin mielessä. L_p -stabiilisuus määritellään seuraavasti. Olkoon meillä järjestelmä, jonka siirtofunktioesitys on

$$Y(s) = G(s)U(s), \quad (1)$$

missä $G(s)$ järjestelmän $g(t)$ siirtofunktio. Järjestelmä on L_p -stabiili, jos

$$(u \in L_p \Rightarrow y \in L_p) \text{ ja } (\forall u \in L_p, \exists c \geq 0 : \|y\|_p \leq c\|u\|_p), \quad (2)$$

missä merkintä

$$f(t) \in L_p \Leftrightarrow \|f\|_p = \left(\int_0^\infty |f(\tau)|^p d\tau \right)^{1/p} < \infty. \quad (3)$$

Tämä tarkoittaa, että jos järjestelmän tulon L_p -normi on äärellinen niin myös lähdön L_p -normin täytyy olla äärellinen. Jos $p = \infty$, vastaa määritelmä BIBO (*bounded input bounded output*)-stabiilisuutta. [2, s. 79-80]

Adaptiivisen säädön stabiilisuuden todistaminen on yleisesti ottaen haastavaa. Stabiilisuuteen vaikuttavat ainakin järjestelmän luontainen stabiilisuus sekä identifikaation ja säätimen stabiilisuudet. Kokonaisjärjestelmän stabiilisuuden osoittamiseksi onkin kyettävä analysoimaan järjestelmän kaikkien osien toimintaa kokonaisuutena. On kuitenkin mahdollista todistaa joidenkin adaptiivisten säätinten olevan stabiileja tietyin edellytyksin [2] [3].

3.4 Robustisuus

Robustisuus tarkoittaa säätimen kykyä säilyttää hyväksyttävä toimintakyky ulkoisten tai sisäisten olosuhteiden muuttuessa. Säätimen tulisi toimia esimerkiksi erilaisista häiriöistä, järjestelmän mallintamattomasta käyttäytymisestä ja järjestelmän parametrien vaeltamisesta huolimatta. Adaptiivinen säädin, joka on suunniteltu ottamatta huomioon mahdollisia häiriötekijöitä, saattaa kuitenkin olla pahimmassa tapauksessa epästabiili sen säätäessä todellista järjestelmää [3, s. 215-219].

Robustin säätimen suunnittelun lähtökohtana on oltava jonkinlainen malli siitä, minkä tyyppiset epäideaalisuudet kyseisessä prosessissa ovat oleellisia. Eri tyyppisten virheiden huomioimiseksi on kehitetty erilaisia menetelmiä [2] [3].

Robustisuuden kannalta on myös tärkeää, että ohjattavan järjestelmän olennaiset parametrit saadaan identifioidua mallinnusvirheistä huolimatta. Tästä syystä ohjaussignaalin laatu on erityisen tärkeä. Esimerkiksi olkoon meillä järjestelmä, jonka malli on

$$y = G(s)u + \Delta(s)u, \quad (4)$$

missä $G(s)$ on järjestelmän mallinnettu dynamiikka ja $\Delta(s)$ on mallintamaton dynamiikka. Tavoitteena on identifioida $G(s)$:n parametrit. Tämä onnistuu, mikäli ohjaussignaali u on olennaisen rikas (*dominantly rich*), eli jos se sisältää riittävän määrän taajuuskomponentteja identifikaatiota varten, ja jos sen sisältämät taajuu-
det aiheuttavat suuremman vasteen järjestelmän mallinnetussa ($G(s)$) kuin mallintamattomassa osassa ($\Delta(s)$). Toinen usein käytetty vaatimus ohjaussignaalille $u(t)$ on, että se täyttää jatkuvan aktivoinnin ehdon (*persistent excitation, PE*)

$$\forall t \geq 0, \exists \alpha_0, t_0 > 0 : \int_t^{t+t_0} u(\tau)^2 d\tau \geq \alpha_0 t_0, \quad (5)$$

missä α_0 kuvaa signaalin aktivointitasoa. Tämä tarkoittaa, että u :n neliön integraalin on oltava kaikilla aikaväleillä nolaa suurempi. [2, s. 150, 177, 638-639]

Yleisesti ohjaussignaalin rikkaus voi olla hankala taata. Suunnittelemalla järjestelmän parametrien identifikaatio sekä säätimen parametrien päivitys robustiksi oleellisten epäidealiteettien suhteen voidaan adaptiivisen säätimen kokonaisrobustisuutta kuitenkin parantaa. [2, s. 634-635] [3, s. 248-256]

4 Rekursiivinen identifiointi

4.1 Yleistä

Määritelmä

Identifiointilla tarkoitetaan yleisesti järjestelmän mallin parametrien estimointia järjestelmän tulojen ja lähtöjen avulla. Identifiointimenetelmät voidaan jakaa karkeasti rekursiivisiin ja ei-rekursiivisiin.

Ei-rekursiivisessa (*off-line*) identifiointissa parametrien estimaatit lasketaan jonkin kertyneen historian perusteella. Estimaatti lasketaan aina kokonaan uudestaan kun dataa tulee lisää. Ei-rekursiivisessa identifiointissa voidaan käyttää monia erilaisia menetelmiä. [5, s. 7-8]

Rekursiivinen (*on-line*) identifiointi eroaa ei-rekursiivisesta identifiointista siten, että parametrien estimaatteja päivitetään sitä mukaa, kun uutta dataa tulee käytettäväksi. Estimaatit määrittyvät siis rekursiivisesti ja ovat ajan funktioita. Menetelmät käyttävät yleensä aikaisempaa estimaattia hyödyksi seuraavan laskemisessa. Vanhaa dataa ei tällöin tarvitse käsitellä enää uudestaan, koska sen sisältämä informaatio sisältyy jo edelliseen estimaattiin. [5, s. 7-8]

Rekursiivisen identifiointin menetelmät ovat yleensä rajoittuneempia kuin ei-rekursiivisen identifiointin. Reaaliaikaisuuden takia myös muisti- ja laskentaresurssit asettavat usein omia rajoitteitaan rekursiivisille menetelmille. [5, s. 8]

Matemaattinen perusta

Identifiointi perustuu siihen, että verrataan järjestelmän lähtöä $y(u, \theta^*)$ mallin lähtöön $\hat{y}(u, \hat{\theta})$, missä $\hat{\theta}$ on mallin estimoitu parametrivektori ja θ^* mallin todellinen parametrivektori. Identifikaation tarkoituksena on saada estimoidut parametrit suppenemaan kohti järjestelmän todellisia parametreja. Identifioitava järjestelmä esitetään yleensä muodossa

$$y = \theta^{*T} \phi(y, u), \quad (6)$$

missä ϕ sisältää järjestelmän oleellisen ohjaus- ja mittaushistorian. Identifikaatio voidaan tämän jälkeen toteuttaa yhtälöillä

$$\hat{y} = h(\hat{\theta}, \phi), \quad (7)$$

$$\epsilon = g(y, \hat{y}), \quad (8)$$

$$\dot{\hat{\theta}} = f(\epsilon, \phi), \quad (9)$$

missä \hat{y} on järjestelmän lähdön estimaatti, ϵ estimaatin virhe ja $\hat{\theta}$ estimoitu parametrivektori. Yhtälöistä johdetaan usein käytännön sovelluksia varten rekursiivinen algoritmi. [2, s. 146-148]

Käytännön modifikaatiot

Identifiointin teoriassa on yleensä oletettu, että järjestelmän kaikki tilat voidaan mitata, järjestelmä on stabiili ja että järjestelmän lähtö on rajoitettu. Näin ei kui-

tenkaan aina adaptiivisen säädön tapauksessa ole. Käytännön tilanteita varten onkin kehitetty useita identifikaation suorituskykyä parantavia menetelmiä.

Esimerkiksi jos järjestelmä ei ole luontaisesti stabiili tai lähtöjen ei voida taata olevan rajoittuneita, voidaan identifikaatioyhtälöt normalisoida, eli yleensä skaalata jollain normalisoivalla signaalilla n_s . Tämä tarkoittaa esimerkiksi estimointivirheen lausekkeen muuttamista muotoon

$$\epsilon = g(y, \hat{y}, n_s, \epsilon), \quad (10)$$

missä n_s on normalisoiva signaali, joka varmistaa sen, että virheen L_∞ -normi on äärellinen. Normalisoinnilla voidaan taata parametrien estimaattien rajallisuus sekä suppeneminen. [2, s. 162-202]

4.2 Rekursiivinen PNS -menetelmä

Määritelmä

Rekursiivinen PNS (*recursive least squares, RLS*) -menetelmä voidaan johtaa lineaarisen differentiaaliyhtälön tapauksessa seuraavasti. Olkoon meillä järjestelmää kuvaava yhtälö

$$y(t) + a_1 y(t-1) + \dots + a_n y(t-n) = b_1 u(t-1) + \dots + b_m u(t-m) + v(t), \quad (11)$$

missä $v(t)$ on nollakeskiarvoista valkoista kohinaa. Yhtälö voidaan esittää muodossa

$$y(t) = \hat{\theta}^T \phi(t) + v(t), \quad (12)$$

missä $\hat{\theta}$ on estimoitu parametrivektori

$$\hat{\theta}^T = (a_1 \dots a_n \ b_1 \dots b_m), \quad (13)$$

ja ϕ on historiavektori

$$\phi(t) = (-y(t-1) \dots -y(t-n) \ u(t-1) \dots u(t-m))^T. \quad (14)$$

Mallin todelliset parametrit ovat θ^* , jotka eivät ole tiedossa ja joita siis haluamme estimoida.

Nyt olkoon tiedossa $y(t)$ ja historia $\phi(t)$, kun $t = 1, 2, \dots, N$. Pyritään minimoimaan mallin ulkopuolelle jäävä käyttäytyminen $v(t) = y(t) - \hat{\theta}^T \phi(t)$ koko aikajakson ajalta, jolloin saadaan minimoitava virhefunktio (*loss function*)

$$V_N(\hat{\theta}, t) = \frac{1}{N} \sum_{t=1}^N [y(t) - \hat{\theta}^T \phi(t)]^2, \quad (15)$$

Virhefunktio $V_N(\hat{\theta}, t)$ voidaan minimoida analyttisesti, jolloin ratkaisuna on

$$\hat{\theta}(N) = \left[\sum_{t=1}^N \phi(t) \phi^T(t) \right]^{-1} \sum_{t=1}^N \phi(t) y(t), \quad (16)$$

olettaen että käänteismatriisi on olemassa. Kaava (16) voidaan kirjoittaa rekursiivisessa muodossa

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{1}{t}R^{-1}(t)\phi(t)[y(t) - \hat{\theta}^T(t-1)\phi(t)], \quad (17)$$

$$R(t) = R(t-1) + \frac{1}{t}[\phi(t)\phi^T(t) - R(t-1)]. \quad (18)$$

Tämän algoritmin ongelmana on kuitenkin se, että $\hat{\theta}(t)$:n laskemiseksi täytyy joka askeleella kääntää matriisi $R(t)$, mikä on laskennallisesti raskas toimitus. Algoritmille löytyy kuitenkin myös ekvivalentti muoto, joka saadaan, kun määritellään

$$P(t) = \frac{1}{t}R^{-1}(t), \quad (19)$$

eli lasketaan $R(t)$:n sijasta matriisi $P(t)$, joka tarvitsee kääntää siis vain ensimmäisellä askeleella. Tällöin saadaan algoritmille lopulta muoto

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)[y(t) - \hat{\theta}^T(t-1)\phi(t)], \quad (20)$$

$$K(t) = \frac{P(t-1)\phi(t)}{1 + \phi^T(t)P(t-1)\phi(t)}, \quad (21)$$

$$P(t) = (I - K(t)\phi^T(t))P(t-1), \quad (22)$$

joka tunnetaan yleisesti rekursiivisena PNS -algoritmina. $K(t)$ on algoritmin 'vahvistus', eli se kuvaa sitä, miten nopeasti algoritmi sallii $\hat{\theta}$:n muuttua. [5, s. 17-20]

Algoritmin johdossa tehtiin oletus, että kaavan (16) käänteismatriisi voidaan laskea. Tämä tarkoittaa sitä, että mittaushistorian pitää olla vähintään yhtä pitkä kuin on $\hat{\theta}$:n dimensio. Tämän takaamiseksi on yleensä kaksi tapaa. Ensimmäinen vaihtoehto on aloittaa rekursio vasta riittävän pitkän ajan kuluttua, jolloin mittaushistoriaa on kertynyt riittävästi. Toinen vaihtoehto on keksiä mittaushistorialle jotkin alkuarvot, joilla varmistetaan matriisin kääntyvyys. Nämä alkuarvot aiheuttavat todennäköisesti jonkinlaisen alkuhäiriön (transientin), mutta sen merkitys häviää ajan kuluessa. [5, s. 20-21]

Modifikaatioita

Rekursiivinen PNS -menetelmä on johdettu edellä melko laajoin oletuksin, varsinkin todellisia järjestelmiä silmälläpitäen. Algoritmiin onkin olemassa erilaisia käytännön toimintakykyä parantavia modifikaatioita, joista seuraavaksi kuvataan muutama yleisin.

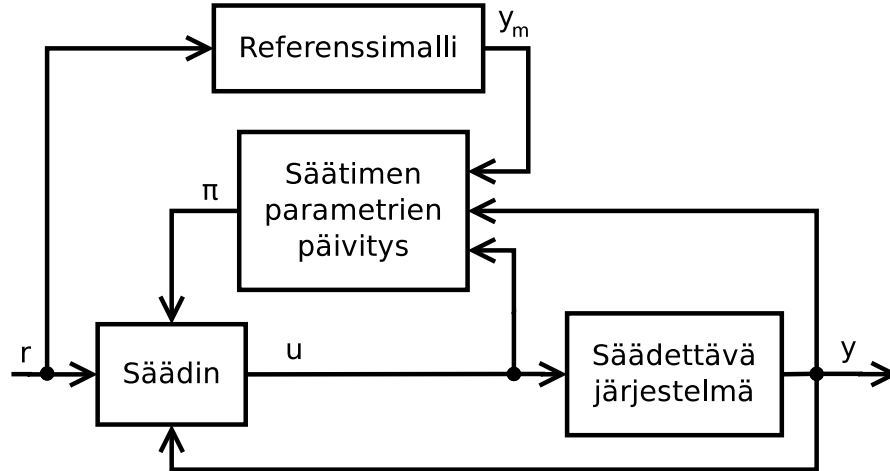
Instrumenttivektori Algoritmin johdossa on oletettu, että häiriö $v(t)$ on nollakeskiarvoista valkoista kohinaa, eikä siten korreloi $u(t)$:n tai $y(t)$:n kanssa. Mikäli näin ei ole, ei voida taata estimoidun parametrivektorin $\hat{\theta}$ yleisesti suppenevan kohiti todellista parametrivektoria θ^* . Mikäli suppenemisen kanssa on ongelmia, algoritmia voidaan modifioida korvaamalla $\phi(t)$ osittain instrumenttivektorilla (*instrumental variable vector, IV-vector*) $\zeta(t)$, jolla on se ominaisuus, että se korreloi $\phi(t)$:n kanssa, mutta ei $v(t)$:n kanssa. [5, s. 21-24]

Nollaus Algoritmin johdossa on myös oletettu, että mallin todelliset parametrit θ^* ovat vakioita. Näin ei kuitenkaan aina ole, vaan ne voivat vaihdella ajan myötä. Mikäli parametrit muuttuvat harvoin, mutta suurin hyppäyksiin, voidaan algoritmia modifioida siten, että matriisi P nollataan tietyillä ehdoilla matriisiksi $I\alpha$, missä α on suuri positiivinen luku. Tällöin myös estimoidut parametrit voivat muuttua suurin hyppäyksiin. Tätä menetelmää kutsutaan nollaukseksi (*resetting*). [1, s. 52]

Unohtaminen Mikäli taas järjestelmän parametrit muuttuvat vähän ja hitaasti, voidaan algoritmia modifioida antamalla eri-ikäisille mittauksille erilaiset painoarvot siten, että uusimpien mittausten painoarvot ovat suurimmat. Tällöin myös uusien mittausten vaikutus $\hat{\theta}$:n on suurempi. Tätä menetelmää kutsutaan unohtamiseksi (*forgetting*). [1, s. 52-53]

5 Mallireferenssisäätimet

5.1 Yleistä



Kuva 2: MRAC-säätimen lohkokaaavioesitys

Mallireferenssisäätimellä (*model reference adaptive control, MRAC*) tarkoitetaan säädintä, joka vertaa järjestelmän käyttäytymistä annettuun referenssimalliin, ja muuttaa säätimen parametreja sen perusteella, miten hyvin järjestelmä noudattaa mallia. Säädin koostuu kahdesta silmukasta: sisempi takaisinkytketty silmukka vastaa normaalia takaisinkytkettyä säätöä, kun taas ulompi silmukka muuttaa säätimen parametreja siten, että järjestelmän ja mallin lähtöjen välinen ero pysyy pienenä. Säätimen lohkokaaavioesitys on esitetty kuvassa 2. [1, s. 20]

MRAC-säädin voidaan nähdä ikään kuin adaptiivisena servojärjestelmänä, sillä mallireferenssisäätimen tavoitteena on saada säädetty järjestelmä käyttäytymään samalla tavalla kuin annettu malli. Tämä vaatii tietysti sen, että suunnittelijalla pitää olla hyvä ymmärrys säädettävästä prosessista, jotta mielekäs malli on mahdollista luoda. [1, s. 185]

Koska tavoitteena on saada järjestelmä käyttäytymään mallin tavoin, on säätimen käytännössä identifioitava säädettävän järjestelmän parametreja tavalla tai toisella. Identifikaatio on kuitenkin säädön kannalta yleensä toissijaisesta, sillä säätövaatimusten toteutumiseksi identifioitujen parametrien ei välttämättä tarvitse edes olla täsmälleen oikeat. MRAC-säädin on luonnostaan lähempänä suoraa kuin epäsuoraa adaptiivista säätöä, mutta se voidaan silti toteuttaa kummalla tahansa menetelmällä. [2, s. 318, 396-397]

Säätimen suunnittelussa suurin haaste on luoda sellainen säätimen parametrien säätöalgoritmi, joka varmistaa, että kokonaisjärjestelmä pysyy stabiilina, ja että mallin ja säädettävän järjestelmän lähtöjen erotus saadaan suppenemaan nolliin. Tämä ei yleensä ole helppoa. [1, s. 20]

5.2 Säätoparametrien päivityssäännöt

Säätimen parametrien päivittämiseen on kaksi yleisesti käytettyä tapaa: MIT-sääntö ja Lyapunov-teoria.

MIT-sääntö

MIT-sääntö sai nimensä siitä, että se on kehitetty MIT:ssä (*Massachusetts Institute of Technology*). Sääntö johdetaan siten, että olkoon meillä virhefunktio V , joka kuvaa säätoparametrien virhettä. V on muotoa

$$V(\pi) = f(\epsilon), \quad (23)$$

missä π on säätimen parametrivektori ja ϵ on järjestelmän ja referenssimallin lähtöjen erotusvektori. Koska tavoitteena on saada V pieneksi, ideana on muuttaa säätimen parametreja V :n negatiivisen gradientin suuntaan, eli matemaattisesti

$$\dot{\pi} = -\gamma \nabla V(\pi), \quad (24)$$

missä γ on vahvistusvektori, joka kuvaa sitä miten herkästi parametrit muuttuvat. Jos esimerkiksi kaikki termit ovat skaalaareja, ja

$$V(\pi) = \frac{1}{2}\epsilon^2, \quad (25)$$

saadaan säännöksi

$$\dot{\pi} = -\gamma \epsilon \frac{\partial \epsilon}{\partial \pi}, \quad (26)$$

joka on alkuperäinen MIT-sääntö. Termiä $\partial \epsilon / \partial \pi$ kutsutaan herkkyysderivaataksi, joka kuvaa sitä, kuinka virhe riippuu säätimen parametrissa π . [1, s. 186-187]

MIT-säädössä vahvistusvektorin γ valinta on tärkeä osa suunnittelua. γ :n valitsemiseksi löytyy suoraviivaisia menetelmiä, kun säädettävästä järjestelmästä on riittävästi tietoa. [1, s. 194-195]

Vahvistusvektorin γ valintaan vaikuttaa myös signaalitaso, joka voi kuitenkin vaihdella suuresti. Tämä ongelma voidaan ratkaista normalisoimalla parametrien päivitysyhtälö (24). Esimerkiksi yhtälö (26) voidaan normalisoida muotoon

$$\dot{\pi} = \frac{-\gamma \epsilon \frac{\partial \epsilon}{\partial \pi}}{\alpha + \left(\frac{\partial \epsilon}{\partial \pi}\right)^2}, \quad (27)$$

missä α on (valittavissa oleva) positiivinen luku, jonka tehtävä on estää ongelmia syntymästä kun $\partial \epsilon / \partial \pi$ on lähellä nollaa. [1, s. 196-198]

Lyapunov-teoria

MIT-säännössä on se ongelma, että kokonaisjärjestelmän stabiilisuuden todistaminen on yleensä vaikeaa. Vaihtoehtoinen tapa johtaa säätimen parametrien päivitysyhtälö perustuu Lyapunovin stabiilisuusteoriaan. Näin johdettujen päivityslakien stabiilisuustarkastelu on yleensä yksinkertaista.

Lyapunov-teoriaa käytettäessä lähtökohtana on löytää ensin säädettävän järjestelmän Lyapunov-funktio L , eli funktio joka on positiividefiniitti ($L(0) = 0$, muutoin $L(x) > 0$) ja jonka aikaderivaatta järjestelmän ratkaisuja pitkin on negatiivisemidefiniitti ($\dot{L}(0) = 0$, muutoin $\dot{L}(x) \leq 0$). Tämä tehdään esimerkiksi johtamalla virheelle ϵ differentiaaliyhtälö

$$\dot{\epsilon} = f(\pi, \theta_m, \theta^*, y, u), \quad (28)$$

missä θ_m on annetun mallin parametrivektori. Yhtälön perusteella voidaan yleensä valita jokin sopiva ehdokas Lyapunov-funktioksi. Derivoimalla tätä funktiota ja vaatimalla negatiivisemidefiniittisyys voidaan johtaa parametrien päivityslait, mikäli valittu funktio oli sopiva. Lyapunov-funktion löytäminen voi olla kuitenkin monimutkaisten järjestelmien tapauksessa hankalaa. Esimerkiksi olkoon meillä on referenssimalli

$$\dot{y}_m = -a_m y_m + b_m r, \quad (29)$$

missä $a_m > 0$ ja r on rajoitettu, sekä järjestelmä

$$\dot{y} = -ay + bu. \quad (30)$$

Valitaan säätölaiksi yhtälö

$$u = \pi_1 r - \pi_2 y, \quad (31)$$

ja virheeksi

$$\epsilon = y - y_m. \quad (32)$$

Nyt voimme johtaa virheelle differentiaaliyhtälön

$$\dot{\epsilon} = -a_m \epsilon - (b\pi_2 + a - a_m)y + (b\pi_1 - b_m)r. \quad (33)$$

Muokataan tästä Lyapunov-funktio valitsemalla $b\gamma > 0$ ja

$$L(\epsilon, \pi) = \frac{1}{2} \left(\epsilon^2 + \frac{1}{b\gamma} (b\pi_2 + a - a_m)^2 + \frac{1}{b\gamma} (b\pi_1 - b_m)^2 \right), \quad (34)$$

joka on siis nolla, kun virhe ϵ on nolla ja säätöparametrit π ovat 'oikeissa arvoissaan' sisempien sulkulausekkeiden ollessa nollia. Muutoin L on positiivinen. L :n aikaderivaatta on

$$\dot{L}(\epsilon, \pi) = \epsilon \dot{\epsilon} + \frac{1}{\gamma} (b\pi_2 + a - a_m) \dot{\pi}_2 + \frac{1}{\gamma} (b\pi_1 - b_m) \dot{\pi}_1 \quad (35)$$

$$= -a_m \epsilon^2 + \frac{1}{\gamma} (b\pi_2 + a - a_m) (\dot{\pi}_2 - y\gamma\epsilon) + \frac{1}{\gamma} (b\pi_1 - b_m) (\dot{\pi}_1 + r\gamma\epsilon), \quad (36)$$

joka saadaan valinnalla

$$\dot{\pi}_1 = -r\gamma\epsilon, \quad (37)$$

$$\dot{\pi}_2 = y\gamma\epsilon, \quad (38)$$

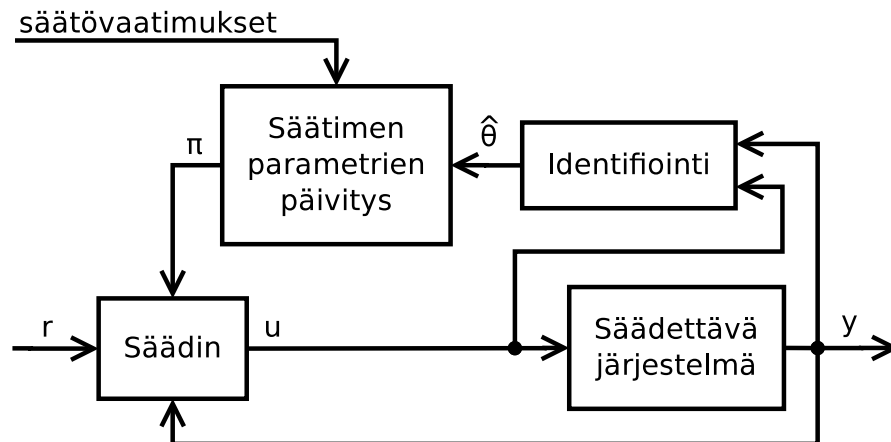
muotoon

$$\dot{L}(\epsilon, \pi) = -a_m \epsilon^2, \quad (39)$$

joka on siis negatiivisemidefiniitti, koska vaadittiin $a_m > 0$. Voidaan osoittaa, että näillä valinnoilla järjestelmän lähtö y pysyy rajoitettuna ja virhe ϵ menee ajan kuluessa nolnaan. [1, s. 206-209]

6 Itsevirittyvät säätimet

6.1 Yleistä



Kuva 3: STR-säätimen lohkokaavioesitys

Itsevirittyvällä säätimellä (*self-tuning regulator, STR*) tarkoitetaan säädintä, joka identifioi järjestelmän parametreja, ja päivittää säätimen parametreja näiden estimoitujen parametrien perusteella. Itsevirittyvän säätimen suunnittelun ongelmana on sellaisen algoritmin luominen, joka pystyy laskemaan säätimen parametrit järjestelmän estimoitujen parametrien ja annettujen säätövaatimusten avulla. Tätä ongelmaa kutsutaan myös pohjimmaisiksi suunnitteluongelmaksi (*underlying design problem*). Säätimen lohkokaavioesitys on esitetty kuvassa 3. [1, s. 21-22, 90-91]

Itsevirittyvän säätimen toteuttamiseen tarvitaan sekä säätimen että identifioinnin suunnittelu. STR-säädin on luonnostaan lähellä epäsuoraa adaptiivista säätöä, mutta se voidaan hyvin toteuttaa myös suorana adaptiivisena säätönä. Suoraa säätöä voidaan kuitenkin käyttää lähinnä yksinkertaisissa järjestelmissä. [2, s. 435]

Säätimen eri osat voidaan toteuttaa täysin erillään ja monilla eri tavoilla. Tämän vuoksi STR-säädin on erittäin monipuolinen säädin, jota voidaan soveltaa moniin eri tarkoituksiin. STR-säätimen toteutuksessa on myös vähemmän reunaehtoja kuin MRAC-säätimen toteutuksessa. Esimerkiksi mikäli säädettävä järjestelmä on ei-minimivaiheinen, voi MRAC-säätimien kanssa tulla ongelmia, jotka STR-säädintä käytettäessä voidaan välttää. [1, s. 90-91] [2, s. 434]

6.2 Säätöparametrien päivitysmenetelmät

Itsevirittyvä säädin tarvitsee aina jonkin algoritmin, jolla se laskee säätimen parametrit identifioitujen järjestelmäparametrien perusteella. Menetelmiä on monia, ja niistä esitellään seuraavaksi deterministinen napojen asettelu ja stokastinen minimivarianssisäätö.

Napojen asettelu

Napojen asettelun idea on yksinkertainen. Kun säädettävän järjestelmän ja säätimen siirtofunktiot ovat tiedossa, ja järjestelmän parametrit on identifioitu, on tehtävänä vain asettaa säätimelle sellaiset parametrit, että suljetun järjestelmän navat ovat tietyissä ennalta määritetyissä kohdissa. Tällaisista menetelmistä käytetään myös nimitystä adaptiivinen napojen asettelu (*adaptive pole placement control, APPC*). [2, s. 16]

Johdetaan seuraavaksi napojen asettelun yhtälöt yksinkertaiselle jatkuvalla järjestelmälle

$$\dot{y} = ay + bu, \quad (40)$$

missä a ja b ovat tuntemattomia vakioita, mutta b :n etumerkki on tiedossa. Tavoitteena on saada tämän järjestelmän napa pisteeseen $-a_m$, missä $a_m > 0$ on annettu. Nyt olkoon säätölaki

$$u = -ky + r, \quad (41)$$

$$\hat{k} = \frac{\hat{a} + a_m}{\hat{b}}, \quad (42)$$

missä \hat{a} ja \hat{b} ovat todellisten parametrien estimaatit. Mikäli estimaatit ovat tarkkoja, johtaa tämä säätölaki selvästi säätövaatimuksen täyttymiseen, jolloin kokonaisjärjestelmä on muotoa

$$\dot{y} = -a_my + br. \quad (43)$$

Tämän jälkeen täytyy vielä suunnitella identifiointi parametreille \hat{a} ja \hat{b} (epäsuora menetelmä) tai parametrille \hat{k} (suora menetelmä). Tämä voidaan tehdä monella tavalla, kuten käyttämällä rekursiivista PNS -menetelmää (kappale 4.2) tai Lyapunov-menetelmää (kappale 5.2). [2, s. 436-440]

Minimivarianssisäätö

Edellä kuvattu napojen asettelu on deterministinen menetelmä, eli säätöparametrit määräytyivät suoraviivaisesti identifioitujen järjestelmäparametrien perusteella. Minimivarianssisäätö (*minimum variance control, MV control*) taas on stokastinen menetelmä ja perustuu säätösuureen varianssin minimointiin.

Menetelmässä tavoitteena on minimoida virhekkriteeri

$$V = E(y^2(t)), \quad (44)$$

missä E tarkoittaa matemaattista odotusarvoa. Säätötavoitteena on siis pitää y nollassa (regulointitehtävä). Olkoon meillä esimerkiksi yksinkertainen diskreetti järjestelmä

$$y(t+1) = -ay(t) + bu(t) + v(t+1) + cv(t). \quad (45)$$

Nyt y :n varianssi on funktio sekä ohjauksista että häiriöstä. Ohjaukset voidaan valita, mutta häiriöön ei voida vaikuttaa. y :n varianssi on siis aina vähintään yhtä suuri

kuin häiriön varianssi. Koska kuitenkin vanhat ohjaukset ja mittaukset, sekä parametrien a , b ja c estimaatit tunnetaan, niin häiriöt voidaan laskea hetkeen t asti. Näiden perusteella taas voidaan johtaa säätölaki

$$u(t) = \frac{\hat{a}y(t) - \hat{c}v(t)}{\hat{b}}. \quad (46)$$

Jos parametrit ovat tarkat, saadaan säädetty järjestelmä

$$y(t+1) = v(t+1). \quad (47)$$

Nyt y :n varianssi on sama kuin v :n varianssi, mikä on siis pienin saavutettavissa oleva varianssi ja samalla yhtälön (44) haluttu minimi. Nyt säädetylle järjestelmälle pätee myös tietysti, että

$$y(t) = v(t), \quad (48)$$

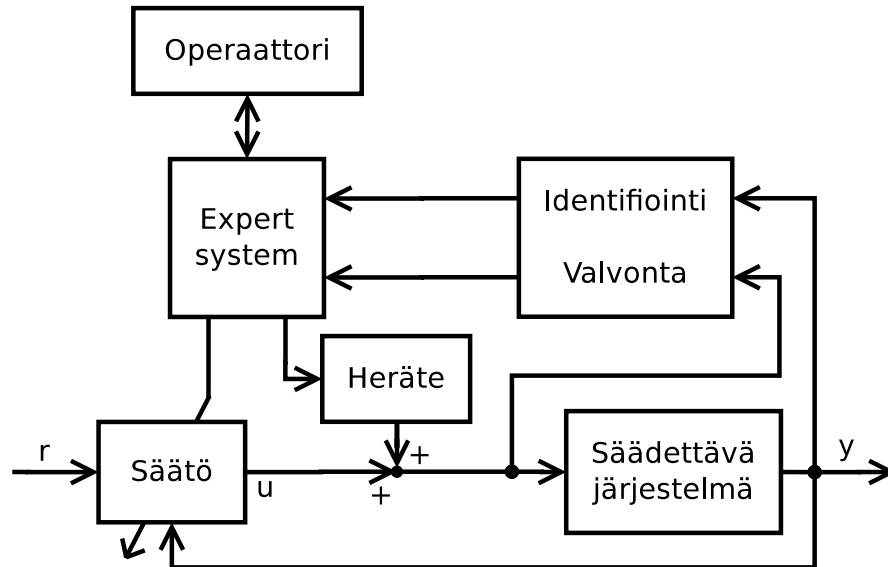
jolloin säätölaki saa muodon

$$u(t) = \frac{\hat{a} - \hat{c}}{\hat{b}}y(t), \quad (49)$$

joka on siis tämän järjestelmän minimivarianssisäädin. Taaskin tehtäväksi jää vielä suunnitella parametrien identifiointi. [1, s. 140]

7 Muita menetelmiä

7.1 Expert system -menetelmät



Kuva 4: Expert system -säätimen lohkokaaavioesitys

Kuten aikaisemmin on todettu, adaptiivisten säädintien suunnittelu vaatii runsaasti ymmärrystä säädettävästä järjestelmästä. Tämä tieto voidaan kuitenkin hankkia ainakin osittain loogisten päättelyketjujen avulla, jonka jälkeen voidaan valita kyseiseen tilanteeseen hyvin soveltuva säädin. Tähän päättelyyn voidaan käyttää expert system -menetelmiä. [3, s. 328-329]

Expert system -menetelmät ovatkin tietyssä mielessä adaptiivisten säätimien 'yläpuolella'. Menetelmä perustuu tietämuskantaan (*knowledge base*), johon on kerätty tiettyyn aihealueeseen (tässä tapauksessa säätöön) liittyviä loogisia päättelyketjuja. Näitä päättelyketjuja käyttäen säätöjärjestelmä voi tehdä johtopäätöksiä siitä, miten järjestelmää tulisi säätää. Johtopäätökset tekevää osaa järjestelmästä kutsutaan päättelijäksi (*inference engine*). Järjestelmässä on myös käyttäjärajapinta (*interface*), jonka avulla järjestelmän tekemistä ratkaisuista voidaan saada informaatiota, ja jonka avulla tietämuskannan sääntöjä pystytään muokkaamaan tarvittaessa. Säätimen lohkokaaavioesitys on esitetty kuvassa 4. [1, s. 554-556] [3, s. 328-329] [7, s. 37-39]

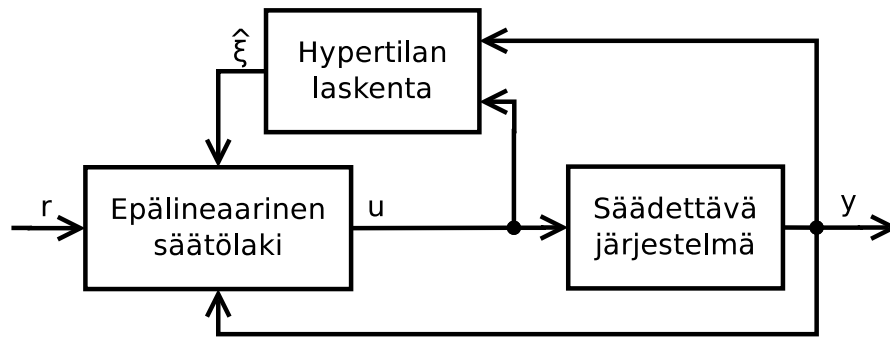
Säädin kerää säädettävästä järjestelmästä informaatiota esimerkiksi identifioimalla järjestelmän parametreja. Säädin voi myös valvoa järjestelmää ja reagoida eri tyyppisiin hälytystilanteisiin johdonmukaisella tavalla. Expert system -säätimet soveltuvatkin hyvin tilanteisiin, jossa tarvitaan turvalogiikkaa. [1, s. 554-556] [3, s. 328-329]

Expert system -säädin pystyy myös käyttämään säätöön kulloinkin tarkoitukseenmukaista säätömenetelmää. Se voi siis esimerkiksi tietyissä tilanteissa säätää jär-

jestelmää pelkällä PID-säätimellä ja toisessa tilanteessa adaptiivisella säätimellä. Säätöjärjestelmä voi myös päätellä, millä menetelmällä käytetty säädin kannattaa virittää ja mitä modifikaatioita säätimen algoritmiin kannattaa tehdä. Säätöjärjestelmä voi myös antaa säädettävälle järjestelmälle herätteitä, mikäli esimerkiksi parametrien identifiointin kanssa on ongelmia. [1, s. 554-556] [3, s. 328-329]

Säätöjärjestelmän toiminta tietysti edellyttää sen, että järjestelmän tietämyksistä löytyy riittävä määrä sääntöjä päättelyä varten ja että järjestelmään on toteutettu riittävä määrä säätöalgoritmeja niihin liittyvine sääntöineen. Tämä tietysti nostaa suunnittelukustannuksia huomattavasti. [1, s. 556]

7.2 Stokastiset menetelmät



Kuva 5: Kaksoissäätimen lohkokaaavioesitys

Stokastisten menetelmien lähtökohtana on tehdä järjestelmästä stokastinen malli, jonka jälkeen suunnitteluongelmana on muodostaa sellainen säätölaki, joka minimoi annetun tilastollisen virhefunktion. Kaksoissäätimen lohkokaaavioesitys on yleensä kuvan 5 mukainen. Tällaista säätöä kutsutaan myös kaksoissäädöksi (*dual control*), sillä säätölakien on havaittu pitävän sekä säätö- että identifiointivirheet pieninä. Tietysti molempia ei voida pitää yhtä aikaa absoluuttisessa minimissä, mutta optimaaliset säätimet löytävät tasapainon näiden kahden usein ristiriitaisen vaatimuksen väliltä. [1, s. 22-24] [3, s. 10-11]

Matemaattisesti esitettynä prosessista identifioidaan prosessin mallin parametrien $\hat{\theta}$ lisäksi parametrien epävarmuudet σ , jolloin saadaan uusi tilavektori

$$\hat{\xi} = [\hat{\theta}^T \sigma^T]^T, \quad (50)$$

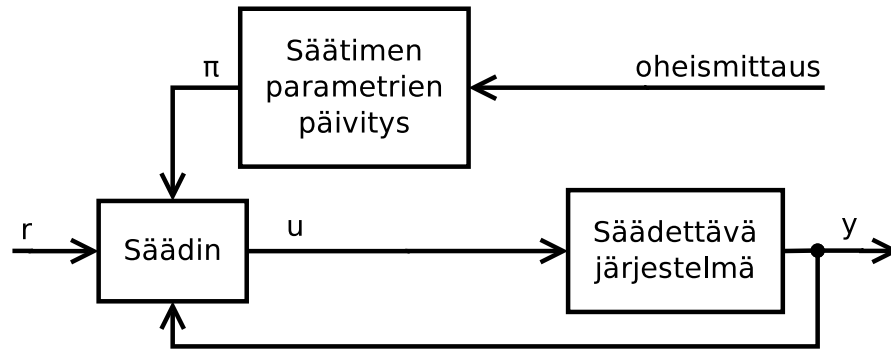
eli niin kutsuttu hypertila (*hyperstate*). Tämän jälkeen haluttu säätölaki saadaan siten, että minimoidaan funktio

$$V = E\left(H(\hat{\xi}(t), u(t)) + \int_0^t h(\hat{\xi}(\tau), u(\tau))d\tau\right), \quad (51)$$

sallittujen ohjaussuureiden u suhteen siten, että u :n tulee olla funktio ainoastaan menneistä ja nykyisistä mittauksista ja epävarmuuksista. E tarkoittaa matemaattista odotusarvoa järjestelmän alkuarvojen todennäköisyysjakauman sekä kaikkien

mallissa olevien häiriöiden suhteen. H ja h ovat järjestelmästä riippuvia skalaarifunktioita. Tämä minimointitehtävä on hankala, ja johtaa yleensä Bellmanin yhtälöön, joka on ratkaistavissa usein vain numeerisesti jos ollenkaan. [1, s. 22-24]

7.3 Gain scheduling



Kuva 6: Gain scheduling -säätimen lohkokaaavioesitys

Gain scheduling -säädin perustuu siihen, että tiedetään joidenkin mitattavissa olevien suureiden korreloivan vahvasti järjestelmän dynamiikan muutoksien kanssa. Tätä tietoa hyväksi käyttäen voidaan taulukoida säätimen parametreja, kuten vahvistus (*gain*), joita sopivasti vaihtamalla mittausarvojen mukaan (aikataulutus, *scheduling*) säädin toimii hyvin eri toimintapisteissä. Säätimen lohkokaaavioesitys on esitetty kuvassa 6. Säätimestä esitellään esimerkksiovellus luvussa 8. [1, s. 19] [2, s. 7-8]

Gain scheduling -säätimen hyvänä puolena on se, että säädin voi reagoida järjestelmän dynamiikan muutoksiin heti kun olosuhteiden havaitaan muuttuneen, eikä vasta sitten, kun järjestelmän lähtöjen havaitaan muuttuneen. Ongelmana on kuitenkin se, että nopea säätöparametrien edestakainen vaihtaminen voi johtaa järjestelmän epästabiilisuuteen, jolloin on olemassa rajataajuus sille, miten usein parametreja voi vaihtaa. Toinen ongelma on se, että säätimen parametritaulukot ovat kiinteät, eikä niitä voi vaihtaa, vaikka huomataan, että järjestelmä ei käyttäydykään odotetulla tavalla. Kolmas ongelma on se, että toimintapisteiden määrän kasvaessa säätimen suunnittelukustannukset voivat nousta korkeiksi. [2, s. 8]

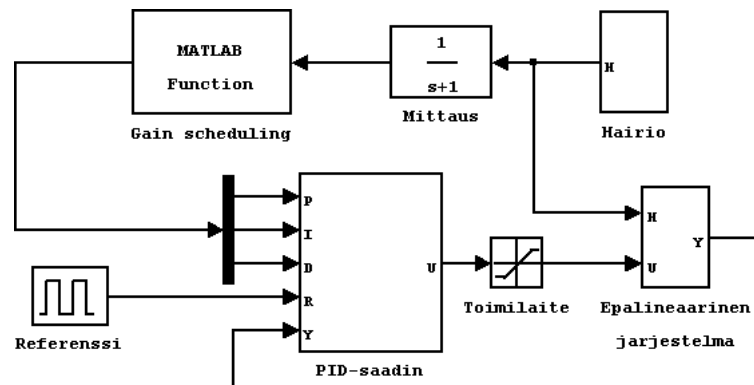
8 Sovellusesimerkki: gain scheduling

Seuraavaksi havainnollistetaan yksinkertaisen adaptiivisen säätimen toimintaa. Esi-merkinä on gain scheduling (GS) -säädin, ja ohjattavana on epälineaarinen järjestelmä

$$\ddot{y} = h\dot{y}^2 + y + u, \quad (52)$$

missä h on mitattavissa oleva häiriö.

Säätimenä toimii ideaalinen PID-säädin sillä modifikaatiolla, että derivoitava suure on erosuureen sijasta järjestelmän lähtö. Säätimen parametrit on viritetty käsin kolmessa eri toimintapisteessä. Säätimen käytössä on lineaarinen toimilaite, jonka lähtö on rajoitettu välille $[-5, 5]$. Ohjauksen referenssiarvo on kanttiaaltoa, jotta voidaan tarkkailla askelvasteen muutoksia. Mittaus on toteutettu yksinkertaisesti alipäästösuodattimena. Järjestelmän lohkoakaavioesitys on kuvassa 7 ja varsinainen algoritmi on esitetty kuvassa 8. Lohkokaavion alijärjestelmät on esitetty liitteessä A.



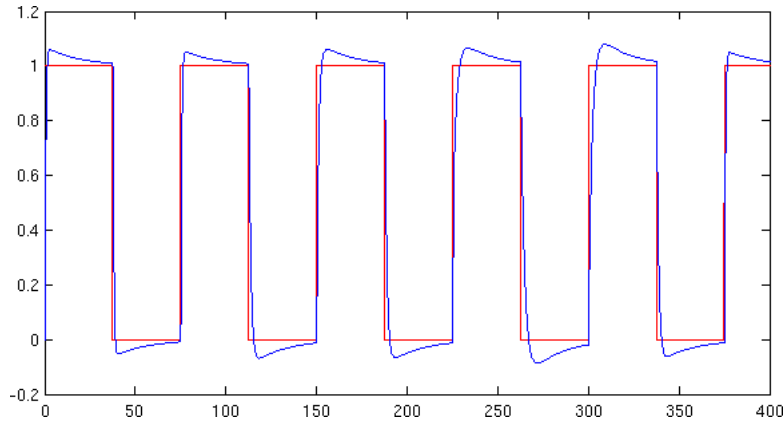
Kuva 7: Gain scheduling -säätimen toteutus Simulink-ympäristössä

```
function PID = gainscheduling(h)

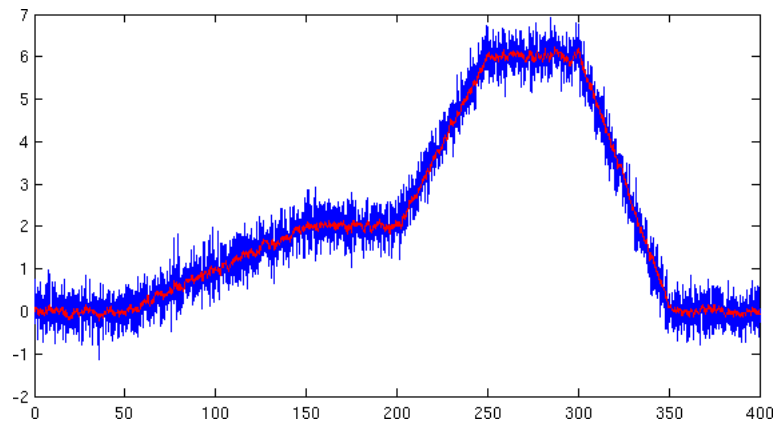
if h < 1
    PID = 1.5 * [20, 1, 10]; % p1
elseif h < 4
    PID = 1.5 * [20, 1, 20]; % p2
else
    PID = 1.5 * [20, 1, 35]; % p3
end
```

Kuva 8: Gain scheduling -algoritmin toteutus Matlab-ympäristössä

Kuvassa 9 nähdään järjestelmän käyttäytyminen normaalitilassa GS-algoritmin muuttaessa PID-säätimen parametreja häiriömittauksen mukaisesti. Voidaan todeta, että järjestelmä käyttäytyy hyväksyttävästi koko aikajakson ajan. Häiriön ollessa pieni järjestelmä käyttäytyy nopeammin kuin silloin, kun häiriö on suuri. Häiriö on esitetty kuvassa 10.

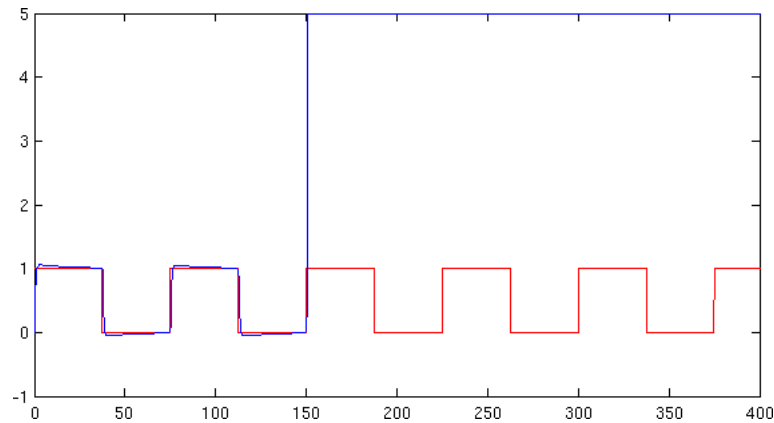


Kuva 9: Säädetyin järjestelmän lähtö (sininen) ja referenssi (punainen)



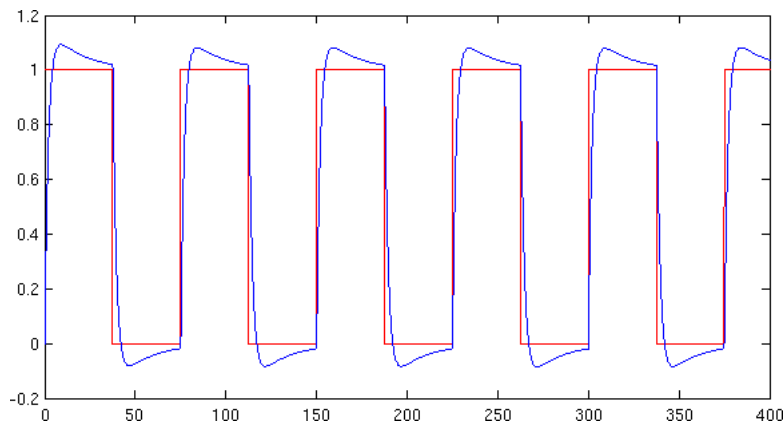
Kuva 10: Häiriösignaali (sininen) ja mitattu häiriö (punainen)

Verrataan seuraavaksi säätimen toimintaa pelkkään staattiseen PID-säätimeen. Kuvassa 11 on esitetty järjestelmän käyttäytyminen, kun säätimen parametrit ovat samat kuin edellä häiriön ollessa pieni (parametrit p1 kuvassa 8). Havaitaan, että säädin ei pysty stabiloimaan järjestelmää tällä virityksellä, jolloin lähtö siis karkaa äärettömyyteen.



Kuva 11: Järjestelmän lähtö, kun sitä säädetään parametreilla $p1$

Toinen vertailukohde on valita säätimelle varovaisimmat parametrit (parametrit $p3$ kuvassa 8), joiden voidaan taata toimivan järjestelmän ollessa vahvasti epälineaarinen. Kuvassa 12 on esitetty järjestelmän käyttäytyminen tällöin. Havaitaan, että säädin pystyy pitämään järjestelmän stabiilina, mutta häiriön ollessa pieni ei päästä yhtä hyvään säätötulokseen kuin adaptiivisella säädöllä (kuva 9).



Kuva 12: Järjestelmän lähtö, kun sitä säädetään parametreilla $p3$

Voidaan siis todeta, että tämän esimerkin tapauksessa adaptiivisesta säädöstä oli hyötyä, ja että sen ansiosta säätötulos oli parempi kuin käytettäessä pelkkää staattista säädintä. Adaptiivinen säädin pystyi mukauttamaan itseään vallitsevaan tilanteeseen, ja sen vuoksi sai pidettyä säädön virheet pienempinä.

9 Yhteenveto

Adaptiivinen säätö kehitettiin 50-luvulla lentokoneiden automaattisen ohjauksen tarpeisiin. Nykyään sitä voidaan soveltaa erittäin monipuolisesti erilaisten säätö-ongelmien ratkaisemiseen.

Adaptiivisen säädön toteuttamisessa on huomioitava useita seikkoja, kuten stabiilisuus ja robustisuus. Näiden todistaminen on yleensä haastavampaa kuin perinteisten säätimien tapauksessa.

Adaptiivinen säätö jakautuu yleensä kahteen osaan: järjestelmän parametrien identifiointiin ja säätimen parametrien päivitykseen. Kummankin osan toteutukseen on olemassa erilaisia menetelmiä.

Rekursiivinen PNS -menetelmä on parametrien identifioinnissa yleisesti käytetty menetelmä. Se on suhteellisen yksinkertainen suunnitella ja toteuttaa, ja siihen on olemassa monia käytännön toimintaa parantavia modifikaatioita.

Adaptiivisista säätömenetelmistä käsiteltiin MRAC-, STR- ja GS-menetelmät, ja sivuttiin stokastisia- sekä expert system -menetelmiä.

MRAC-säätimet perustuvat siihen, että säätimen parametreja päivitetään sen perusteella miten hyvin säädettävä järjestelmä noudattaa annetun referenssijärjestelmän käyttäytymistä. Parametrien päivitykseen on kaksi käytettyä tapaa: MIT-säätö ja Lyapunov-teoria.

STR-säätimet perustuvat siihen, että järjestelmästä identifioituja parametreja ja annettuja säätövaatimuksia käyttäen voidaan säätimen parametrit päivittää järjestelmän ollessa käynnissä. Suunnittelumenetelmistä esiteltiin napojen asettelu ja minimivarianssisäätö.

GS-säätimet perustuvat siihen, että säätimelle on taulukoitu parametreja, joita valitaan käyttöön jonkin kriteerin mukaan. Tämä on eräänlaista avointa adaptiivista säätöä, sillä itse adaptaatiolla ei ole omaa takaisinkytkentää.

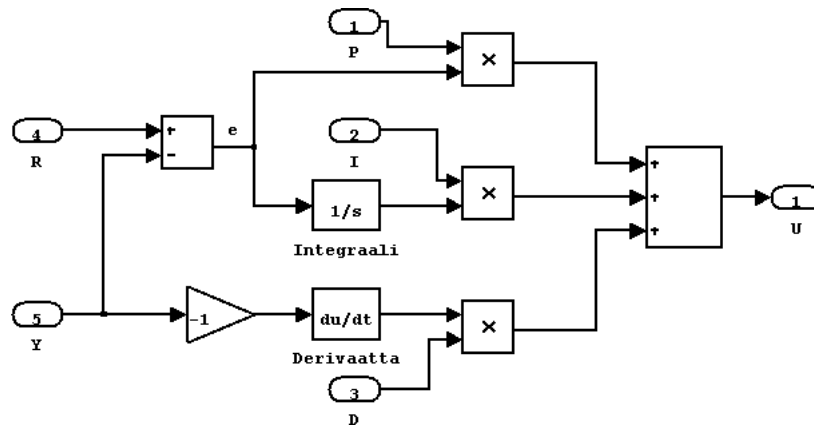
Kaikki nämä säätömenetelmät voidaan toteuttaa monilla eri tavoilla kuitenkin samaa perusideaa käyttäen. Erilaisia toteutustapoja sekä modifikaatioita on lukematon määrä.

Työssä on käsitelty adaptiivista säätöä eri osa-alueilta. Aihe on kokonaisuudessaan kuitenkin valtavan laaja, joten tämä työ toimii lähinnä johdantona asiaan.

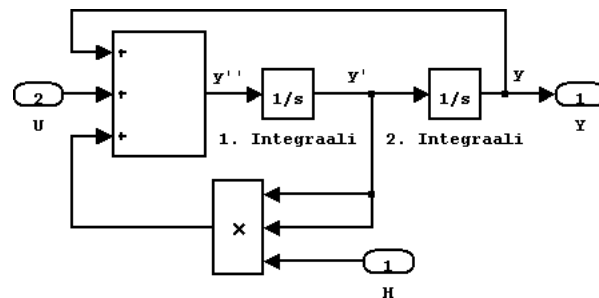
Viitteet

- [1] K. J. Åström ja B. Wittenmark, *Adaptive Control*. Addison-Wesley, 2. painos., 1995. ISBN 0-201-55866-1.
- [2] P. A. Ionnanu ja J. Sun, *Robust Adaptive Control*. Prentice-Hall, 1996. ISBN 0-13-439100-4.
- [3] S. Sastry ja M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*. Prentice-Hall, 1989. ISBN 0-13-004326-5.
- [4] P. J. Antsaklis ja K. M. Passino, *An Introduction to Intelligent and Autonomous Control*. Kluwer Academic Publishers, 1993. ISBN 0-7923-9267-1.
- [5] L. Ljung ja T. Söderström, *Theory and Practice of Recursive Identification*. MIT Press, 1983. ISBN 0-262-12095-X.
- [6] P. A. Ioannou ja B. Fidan, *Adaptive control tutorial*. Cambridge University Press, 2006. ISBN 0-89871-615-2.
- [7] J. P. Ignizio, *An introduction to expert systems: the development and implementation of rule-based expert systems*. McGraw-Hill, 1991. ISBN 0-07-909785-5.

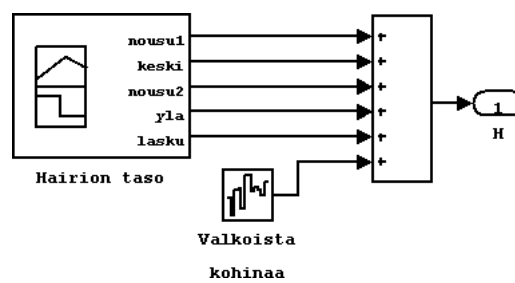
A Sovellusesimerkin alijärjestelmät



Kuva 13: Käytetyn PID-säätimen toteutus Simulink-ympäristössä



Kuva 14: Säädetyn järjestelmän toteutus Simulink-ympäristössä



Kuva 15: Häiriön toteutus Simulink-ympäristössä