

Lappeenrannan teknillinen yliopisto

Tietotekniikan osasto

Kandidaatintyö

Rouhintasorvauksen säätöjärjestelmä

Kandidaatintyön aihe on hyväksytty 10.10.2007.

Työn tarkastajana toimii dosentti Ville Kyrki.

5.12.2007

Juho Ratava

Katajakatu 8D60

53810 Lappeenranta

+358 50 512 4389

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
Tietotekniikan osasto
Juho Ratava

Rouhintasorvauksen säätöjärjestelmä
Kandidaatintyö, 2007

45 sivua, 3 kuvaa, 3 liitettä.

Tarkastaja: Dosentti Ville Kyrki

Hakusanat: rouhintasorvaus, sumea logiikka, säätö

Tässä työssä kuvataan menetelmä, jonka avulla on mahdollista sorvausprosessista mitattujen signaalien perusteella muokata lastuamisprosessin parametreja siten, että prosessissa mahdollisesti esiintyvät ongelmatilanteet korjataan. Työ on tehty osana Feedchip-tutkimushanketta ja tukeutuu tutkimushankkeessa aiemmin tehtyyn työhön vaadittavien korjaustoimenpiteiden, signaaleja mittaavien antureiden instrumentoinnin sekä alustavan ongelmatilanteiden ominaispiirteiden signaaleista tunnistuksen osalta.

Tämä työ keskittyy esittelemään toiminnot, joiden avulla aiemmat tulokset voidaan koota yhteen kokonaisuuteen. Järjestelmän toiminta edellyttää sen osien toiminnan korkean tason koordinoitua. Lisäksi määritellään päättelyjärjestelmä, joka kykenee mitatuista arvoista tunnistettujen ongelmatilanteiden esiintymisasteiden perusteella määrittämään tarvittavat toimenpiteet ongelmatilanteiden poistamiseksi. Kandidaatintyön rinnalla toteutetaan ohjelmisto Lappeenrannan teknillisen yliopiston konepajatekniikan laboratorion sorvausjärjestelmän yhteyteen rakennetun prototyypilaitteiston ohjaamiseksi.

ABSTRACT

Lappeenranta University of Technology
Department of Information Technology
Juho Ratava

Control System for Turning
Bachelor's Thesis, 2007

45 pages, 3 pictures, 3 appendices.

Examiner: Docent Ville Kyrki

Keywords: turning, fuzzy logic, control

This study describes a method for controlling a turning process based on signals measured from the process. The control is applied to a lathe in order to avoid problems that may surface during the machining. The work is done as a part of the Feedchip research project and it is based on earlier work done in the project including the required corrective measures, installing of the sensors used to measure the process and the preliminary identification of the problem situations in machining based on the measured signals.

This paper concentrates to present the functionality required to integrate the earlier results into a complete system. The previously-missing parts include high level coordination of the entire system as well as an inference system able to deduct the optimal corrective action based on the identified fuzzy values of problems appearing in the measured signals. During this work there has been implemented a computer program to control the prototype machining system built around the NC-lathe of the machine shop of the Laboratory of Production Engineering of the Lappeenranta University of Technology.

ALKUSANAT

Tämä työ on tehty Lappeenrannan teknillisen yliopiston tietotekniikan laitokselle saman yliopiston konepajatekniikan laboratoriossa suoritetun harjoittelun yhteydessä. Työn ohjaajana on toiminut tohtori Tuomo Lindh ja tarkastajana dosentti Ville Kyrki.

Työ kuuluu yliopiston hankkeeseen ”Oppiva reaaliaikainen analysointi- ja säätöjärjestelmä sorvaukseen” eli Feedchip. Huolimatta siitä, että tässä työssä on käytetty joitakin uusia tekniikoita, se on onnistunut mielestäni melko hyvin. Vaikka työn painopiste onkin logiikassa ja ongelmatilanteiden mallintamisessa, se on vaatinut laaja-alaisesti tietoa eri aloilta kuten ohjelmistotekniikasta, digitaalisesta signaalinkäsittelystä ja lastuamistekniikasta. Vaikkakin oma tietotaitoni on signaalinkäsittelyn ja erityisesti lastuamistekniikan suhteen rajallinen, tämä työ on yksi Feedchip-projektin konkreettisista esimerkeistä mitä voidaan saavuttaa eri tutkimusalojen välisellä yhteistyöllä.

Tutkimushankkeen työntekijöistä haluan kiittää työni ohjaajaa tohtori Tuomo Lindhiä hänen avustaan signaalinkäsittelyn kanssa, tekniikan ylioppilas Johanna Leppästä hänen tämän työn suhteen korvaamattomasta avustaan lastuamisteknisten asioiden suhteen sekä sähköinsinööri Mikko Rikkosta hänen avustaan sorvin rajapinnan kanssa. Lisäksi haluan kiittää työn tarkastajaa Ville Kyrkiä sekä uudelleen Johanna Leppästä heidän tästä työstä antamansa palautteen vuoksi.

Lappeenrannassa 5.12.2007

Juho Ratava

SISÄLLYSLUETTELO

TIIVISTELMÄ.....	I
ABSTRACT	II
ALKUSANAT.....	III
SISÄLLYSLUETTELO.....	1
KÄYTETYT SYMBOLIT JA LYHENTEET.....	2
1 JOHDANTO	4
1.1 Tausta.....	4
1.2 Tavoitteet ja rajaukset.....	5
1.3 Työn rakenne.....	6
2 TEORIA.....	7
2.1 Signaalinkäsittelystä	7
2.2 Yleisesti sumeista joukoista, -lukuista ja -logiikasta.....	9
2.3 Sumeat konnektiivit	10
2.4 Aggregaatiosta.....	10
2.5 Ordered Weighted Average -operaattori.....	11
2.6 Sääto käyttäen useita antureita	12
3 TYÖN TOTEUTUS	13
3.1 Yleistä	13
3.2 Järjestelmänhallinta	14
3.2.1 Reaaliaikaisen kellon toteutus	15
3.2.2 Kirjanpito ja tietojen syöttö muille järjestelmän osille (output).....	16
3.3 Signaalinkäsittely	17
3.4 Päätelyjärjestelmä.....	19
3.4.1 Lastuamisprosessin ongelmatilanteiden esiintymisen määrittäminen	20
3.4.2 Jatkolastun havaitseminen.....	21
3.4.3 Erikoistapauksia.....	22
3.4.4 Sääntöjen liipaisu ja säädön oikeellisuuden tarkistaminen	23
3.4.5 Korjaustoimenpiteet.....	23
3.5 Tulokset.....	24
4 POHDINTAA	26
4.1 Tulosten hyödyntäminen.....	26
4.2 Jatkotutkimusaiheita	26
5 YHTEENVETO.....	28
LÄHDELUETTELO.....	29
LIITE 1: JATKOLASTUN ESIINTYMISDATAAN SOVITETTU MALLI JA SEN VARIANSSIANALYYSI	32
LIITE 2: KÄYTETTY PÄÄTTELYALGORITMI PSEUDOKOODINA.....	36
LIITE 3: KOETULOKSIA (SÄÄDÖN DEMO-OHJELMA).....	37

KÄYTETYT SYMBOLIT JA LYHENTEET

ACC1		Päätehon suuntainen kiihtyvyys(signaali)
ACC2		Syöttötehon suuntainen kiihtyvyys(signaali)
AE		Akustinen emissio (tai ~n signaali)
LSQ		Least Squares, pienin neliösumma
MIC		Mikrofoni
NC-sorvi tai CNC-sorvi		Numerical Control tai Numerically Controlled. Sorvi, jota ohjataan ISO-standardoidulla koodikielellä. Historiallisesti tietokoneohjatut sorvit tunnetaan myös termillä CNC eli Computer Numerical Control, joskin käytännössä nykyään kaikki NC-sorvit ovat tietokoneohjattuja.
OWA		Ordered Weighted Average tai Ordered Weighted Averaging; painotettu keskiarvo, jossa lista järjestetään ensin suuruusjärjestykseen ennen kuin painoja sovelletaan. Katso luku 2.5.
P		Teho(signaali)
PNS		Pienin neliösumma
RMS		Root Mean Square, neliöllinen keskiarvo
0		(Täydellinen) joukkoon kuulumattomuus tai epätosi.
1		(Täydellinen) joukkoon kuuluminen tai tosi.
μ		Sumean joukon jäsenyysfunktio
ω	[Hz]	Taajuus (käytetty f :n sijasta väärinkäsitysten välttämiseksi)
a	[mm]	Lastunsyvyys
b		Mallinnuksessa, mallin sovituksessa: (lineaarisen) mallin kertoimet
Agg		Aggregaatio (useamman päättelysäännön yhdiste)
f	[mm/r]	Syöttö
f_{\min}	[mm/r]	Syötön vähimmäisarvo
f_{\max}	[mm/r]	Syötön enimmäisarvo
i		Imaginääriyksikkö $\sqrt{-1}$
\Im		Kompleksiluvun imaginääriosa
j		Indeksi

m		Aggregoitavien päättelysääntöjen tai jäsenyysarvojen määrä
n		Mitattujen näytteiden määrä
R		Päättelysääntö
\Re		Kompleksiluvun reaaliosa
t	[s]	Aika
v	[m/min]	Lastuamisnopeus (terän nopeus pitkin lastuttavan kappaleen pintaa)
v_{\min}	[m/min]	Lastuamisnopeuden vähimmäisarvo
v_{\max}	[m/min]	Lastuamisnopeuden enimmäisarvo
w_i		Yksittäiset painokertoimet
w		Painokertoimet (vektoriesitys)
x	[V]	Mitatut signaalit tai niiden tehollisarvot (riippuen kontekstista).
X		Signaalien taajuusmuunnokset tai kaikkien signaalien joukko
y		Sumeiden joukkojen jäsenyysasteet (eli ”ongelmatilanteiden esiintymisarvot”) tietyillä arvoilla

1 JOHDANTO

1.1 Tausta

Tämä työ on osa Feedchip-hanketta, jonka tavoitteena on kehittää säätöjärjestelmä rouhintasorvaukseen. Tutkimushanke on osa Tekesin SISU 2010 -ohjelmaa, joka on alkanut vuonna 2005 ja on tarkoitus saattaa päätökseen viimeistään vuoden 2008 loppuun mennessä. Hankkeen yhteydessä toistaiseksi julkaistut opinnäytteet ovat Juho Pirneksen diplomityö [Pirnes 2006], Andrey Stepanovin diplomityö [Stepanov 2007] sekä Johanna Leppäsen diplomityö [Leppänen 2007].

Järjestelmän ydin on numeerisesti ohjattu sorvi eli NC-sorvi. Ajettava työstöohjelma syötetään NC-koneen muistiin, minkä jälkeen ohjelman avulla voidaan sorvata halutun muotoinen kappale. Työstöohjelman generointiin on yleensä tavallisella PC:llä ajettavia sovellustyökaluja, joista yleisimpänä Suomessa on Juho Pirneksen benchmarking-tutkimuksen [Pirnes 2006] mukaan suomalainen WinCAM. Nämä ohjelmat mahdollistavat koodin generoinnin halutuin parametrein siten, että lopputuloksena on ohjelmalle syötetyn mallin mukainen kappale. Ohjelman lähdekoodi muodostuu standardoiduista koodeista, jotka voidaan ladata sorvin muistiin. Tämä koodikieli vastaa suurin piirtein PC-koneiden konekieltä, joskin kielessä on tuki myös joillekin yksinkertaisille korkeamman tason rakenteille, kuten silmukoille ja aliohjelmille. Työstöohjelmat voi myös kirjoittaa kokonaan käsin, ja generoituja ohjelmia voi myöhemmin muokata, joskin Pirneksen tutkimuksen mukaan vielä ainakin yhdessä konepajassa ohjelma tehtiin vielä kokonaan manuaalisesti. Feedchip-hankkeen lastuamisteknisenä asiantuntijana toimiva Johanna Leppänen huomauttaa lisäksi, että kirjoitetut työstöohjelmat pitää turvallisuussyistä testata ajamalla ohjelma ”ilmassa” eli ilman työstettävää kappaletta ennen ensimmäistä työstöä, jotta varmistaudutaan sorvin liikeratojen oikeellisuudesta.

Säädön kannalta harmittavasti kerran käynnistettyä työstöohjelmaa ei voi ajon aikana muokata eikä sen suorittamisesta poiketa. Pyrittäessä maksimoimaan lastuvirtaa on olennaista, että sorvia voidaan säätää, jotta vältetään materiaalin epätasaisuuden tai väärin lastuamisparametrien aiheuttamat ongelmatilanteet, joita ovat jatkolastu, kimeät äänet ja rauhaton työstö. [Leppänen 2007] Feedchip-projektin yhteydessä käytetyssä Fanucin 18i-TB ohjauspaneelissa on kuitenkin kytkimet, jolla on mahdollista säätää kahta kolmesta eri lastuamisarvosta. Manuaalisesti säädettäviä lastuamisarvoja

ovat sorvin syöttö (eli terän etenemisnopeutta kappaletta pitkin sorvausakselin suuntaisesti) sekä lastuamisnopeus (eli terän nopeutta kiertoliikkeessä olevan työstettävän kappaleen pintaan nähden). Lastuamissyvyyttä (eli terän leikkaaman lastun paksuutta) ei voida tällä laitteistolla säätää ajonaikaisesti. Näihin kytkimiin on tehty liityntäkortti, jonka avulla ne kytketään irti sorvista ja nämä säätötoimenpiteet voidaan tehdä tietokoneelta. Ilmeisesti syötön ja lastuamisnopeuden säätöön on mahdollisuus myös lähiverkkoyhteyden yli, joskin tätä työtä tehdessä tämän rajapinnan määrittely on vielä tuntematon ja täten sitä ei voida käyttää hyödyksi.

Mainittujen toimintojen lisäksi Feedchip-hankkeessa kehitettävässä rouhintasorvauksen säätöjärjestelmässä toteutetaan terän kunnan seuranta konenäöllä, joskin tämä mainitaan kokonaisuuden takia ja ei kuulu tämän työn piiriin. Terän kulumisen seuranta on tärkeää, sillä terä kuluu työstön aikana, ja loppuun kuluneen terän rikkoontuminen työstön aikana voi mahdollisesti rikkoa työstettävän kappaleen, aiheuttaa sorvin ympäristölle vaaratilanteen sekä vaurioittaa itse työstökoneetta. Tästä syystä työntekijän tulee olla ajoissa paikalla vaihtamassa terä uuteen, jotta työstö voi jatkua. Yleisesti ei ole tapana, että koneistaja valvoo sorvausprosessia koko ajan, vaan työntekijä käynnistää sorvin, valvoo muutaman ensimmäisen kappaleen työstön ja sen jälkeen mahdollisesti jatkaa seuraavalle sorville. Sorvin lastuamisarvot valitaan yleensä siten, että lastuamisen aikana ei voi syntyä vaaratilanteita liian suurien arvojen takia. Terän vaihdot suoritetaan myös hyvissä ajoin. Tämä saattaa aiheuttaa työn laadun ja tehon heikentymistä prosessin aikana: lisää kustannuksia ja vähemmän tuottavaa työtä. [Leppänen 2007, Pirnes 2006]

1.2 Tavoitteet ja rajaukset

Tämä työ kuvaa sorvin lastuamisnopeuden ja syötön työstönaikaiseen säätöön tarkoitetun järjestelmän. Työn tavoitteena on korkeamman tason säätölogiikan määrittely, sen toteuttavan ohjelmiston suunnittelu ja toteutus. Tämä sisältää liitynnät sorviin asennettuihin antureihin ja antureilta mitattujen signaalien käsittelyn.

Erityistä huomiota on kiinnitetty järjestelmän modulaariseen rakenteeseen sekä helppoon ylläpidettävyyteen ja muokattavuuteen sen jatkokehityksen helpottamiseksi, joskin loppuvaiheessa eräät virheenkorjaukset osittain rikkovat tätä periaatetta. Työn tärkein tavoite on kuitenkin toimivan prototyypin luominen.

Huolimatta termin ”säätö” käytöstä työn nimessä, järjestelmällä ei ole huomattavia reaaliaikaisuusvaatimuksia siten kuin ne säätötekniikassa usein ymmärretään. Osittain tämä johtuu myös siitä, että säädettävä kone ei pysty reagoimaan välittömästi säätöjärjestelmän muutoksiin, ja myös siitä, että anturidatan mittauksissa ei voida luottaa hetkellisiin arvoihin, vaan mittaukset jo itsessään kestävät useamman sekunnin. Käytännössä tämä tarkoittaa hitaimmillaan noin kolmen sekunnin viivettä säätörutiinin ajamisesta säätötoiminnon suorittamiseen tämänhetkisellä toteutuksella. Säätörutiini ajetaan viiden sekunnin välein, mikä nostaa huonoimman tapauksen reagointiajan noin seitsemään tai kahdeksaan sekuntiin.

1.3 Työn rakenne

Luvussa kaksi syvennyttään hieman työn teoreettiseen perustaan, tutustuen aikaisempiin toteutuksiin ja käyden läpi yleisellä tasolla signaalinkäsittelyn ja sumean logiikan toimintaa. Tämän työn piirissä teoreettinen osuus on jokseenkin suppea, ja varsinainen teoriaosuus esitetään lähinnä pohjarakenteena mahdolliselle tulevalle kehitystyölle.

Luvussa kolme tutustutaan tarkemmin työn toteutukseen, sekä sen toteutusvaiheessa ilmenneisiin ongelmiin. Tämän luvun alussa esitellään järjestelmä kokonaisuutena, minkä jälkeen jokainen järjestelmän osa on esitelty erikseen oman aliotsikkonsa alla. Lisäksi luvun lopussa esitellään joitakin konkreettisia tuloksia järjestelmän toiminnasta.

Luvussa neljä esitetään pohdintaa työn tulosten hyödyntämisestä. Kuten yllä mainittiin, työssä esiteltävä järjestelmä on tasoltaan lähinnä pohjaratkaisu tulevalle. Feedchip-tutkimushankkeen mittakaavassa tämän työn ohessa toteutettava ohjelmisto liittyy rauhintasorvauksen ohjausjärjestelmän ensimmäiseen prototyyppiin, rakentuen aiemmin projektissa tehdyn työn laskemille perustuksille. Myös joitakin varsinaisia jatkotutkimusaiheita esitellään luvun loppupuolella.

Luku viisi esittää lyhyen ja ytimekkään yhteenvedon tämän työn tiimoilta, keskittyen yleisluontoisempaan tarkasteluun. Tavoitteena on antaa hyvä kokonaiskuva työn koko alueelta, kuitenkin tarvittaessa huomioiden tärkeimmät yksityiskohdat.

2 TEORIA

2.1 Signaalinkäsittelystä

Tässä työssä tarvittavat signaalinkäsittelyn toimenpiteet ovat melko yksinkertaisia. Erityisiä rajoitteita asettavat tarpeeksi tarkkojen mittaustulosten saaminen ja tiedostonsiirtokapasiteetti, muodostaen mittausten määrän ala- ja ylärajan. Koska signaalissa esiintyy jonkin verran satunnaisuutta, tulee mitata riittävä määrä näytteitä, jotta mittaukset analysoimalla voidaan muodostaa varmoja johtopäätöksiä. Kuitenkin on huomioitava, että suuremmat määrät mittauksia vaativat enemmän laskenta-aikaa, jolloin saatujen tulosten varmuuden kasvaessa niiden analysointiin vaadittavan laskentatehon määrä myös kasvaa. Nyquistin teoreeman mukaan jotta voidaan mallintaa tietyn taajuuksinen signaali, täytyy näytteistystaajuuden olla vähintään kaksinkertainen. Tällöinkin voi tapahtua aliasoitumista, millä puolestaan tarkoitetaan sitä, että vielä korkeampitaajuuksisten signaalien aiheuttama interferenssi häiritsee varsinaista signaalia. Tämä pystytään havaitsemaan vain pitämällä näytteistystaajuus huomattavasti pienintä teoreettisesti mahdollista taajuutta suurempana. Tämä pienimmän taajuuden raja tunnetaan myös edellä mainitun teoreeman mukaan Nyquistin taajuutena. [Huttunen 2005]

Operaatio, jossa poistetaan signaalista haluttua suuremmat taajuudet ja tämän jälkeen signaalista lasketaan matalamman näytteistystaajuuden arvot (downsampling) tunnetaan nimellä desimointi. Tätä toimenpidettä voidaan käyttää esimerkiksi käsiteltäessä jotakin signaalin hyvin matalataajuuksista komponenttia, jolloin vaadittavat suotimet voivat olla yksinkertaisempia.

Signaalinkäsittelyn kulmakiviä on niin sanottu Fourier-muunnos (kaava 1, Fourier-muunnos jatkuvalla signaalille), jonka avulla ajan suhteen esitetty signaali x voidaan esittää taajuusavaruudessa signaalina X , joka on taajuuden ω funktio. Digitaalisessa signaalinkäsittelyssä esiintyville diskreeteille signaaleille on käytettävä tämän muunnoksen diskreettiä versiota (Discrete Fourier Transform, DFT, kaava 2). [Huttunen 2005, Kreyzig 1999 luku 10]

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-i2\pi\omega t} dt \quad (1)$$

$$X(\omega) = \sum_{j=0}^n x_j e^{-\frac{2\pi i}{n}kj}, \quad \text{missä } k = 0, 1, 2, \dots, n-1 \quad (2)$$

Fourier-muunnoksessa muutetaan signaali taajuusavaruuteen, eli rekonstruoidaan signaali useamman vakiotajuuden siniaallon summana. Tämän ongelman ratkaiseminen vaatii jonkin verran laskutehoa. Erikoistapauksena signaaleille, joiden pituus n on jonkin kokonaisluvun jokin potenssi, tunnetaan algoritmeja, jotka pystyvät tekemään tämän huomattavasti nopeammin. Nimellä Fast Fourier Transform (nopea Fourier-muunnos) tunnetut algoritmit toimivat hajoita-ja-hallitse -periaatteella. Nämä algoritmit tunnetaan myös kantalukunsa perusteella nimellä *radix*, esimerkiksi $n = 2^k$ pituisille signaaleille *radix*-2. [Huttunen 2005, Kreyzig 1999 luku 10]

Tässä työssä käytetään tehollisarvon mittarina neliöllistä keskiarvoa (quadratic mean, tai tässä yhteydessä yleisemmin Root Mean Square, RMS), jota käytetään sähkötekniikassa yleisesti tähän tarkoitukseen (kaavat 3 - 5).

$$x_{rms} = \sqrt{\langle x^2 \rangle} \quad (3)$$

$$x_{rms} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}} \quad (4)$$

$$x_{rms} = \frac{\|x\|_2}{\sqrt{n}} \quad (5)$$

Signaalin tehosppektrillä voidaan laskea taajuusmuotoisena ilmoitetun signaalin eri taajuuksien tehoja. Yksinkertainen spektrin tehoestimaatti voidaan laskea kertomalla alkioittain signaalin Fourier-muunnos X omalla kompleksikonjugaatillaan \overline{X} (kaavat 6 ja 7).

$$PSD_{est,j} = X_j \overline{X_j} \quad (6)$$

$$\overline{X_j} = \Re(X_j) - i\Im(X_j) \quad (7)$$

Tehospektrin laskemiseen on kehittyneempiäkin menetelmiä, kuten Welchin menetelmä, jonka avulla pyritään pienentämään spektrin varianssia käyttämällä liikkuvaa ikkunaa, joten yksittäinen tehospektrin arvo ottaa tällä menetelmällä huomioon ympäristönsä arvot. [Welch 1967] Ohjelmiston tulevilla kehittyneemmillä versioilla lienee hyödyllisintä käyttää Welchin menetelmää, mutta käytännön toteutussyistä ohjelmiston tämä versio käyttää mainittua estimaattia.

2.2 Yleisesti sumeista joukoista, -luvuista ja -logiikasta

Sumealla joukolla ymmärretään klassisen joukko-opin yleistystä, jossa joukon jäsenyysaste y ilmaistaan jäsenyyden tai ei-jäsenyyden sijasta yksikkövälin $[0, 1]$ arvona. [Zadeh 1965] Voidaankin puhua osittaisesta tai epämääräisestä jäsenyydestä. Sumeilla luvuilla ymmärretään joko sumeaa totuusarvoa, joka on tulkittavissa jonkin sumean joukon jäsenyysasteeksi, tai varsinaista sumeaa lukua, joka on tulkittavissa sumeaksi joukoksi, esimerkiksi sumea luku ”noin kaksi”. Sumea logiikka käsittelee näitä sumeita arvoja. Tämän työn tapauksessa voidaan käsitellä tietoa vaikkapa siitä, miten paljon jokin mikrofoniilta mitattu signaali x kuuluu joukkoon *kimeä ääni* (kaavat 8 ja 9) ja tehdä tämän tiedon pohjalta päätelmiä. On huomattava, että tässä yhteydessä kaavan X tarkoittaa kaikkien signaalien joukkoa. [Mattila 2002 luku 3]

$$\mu_{kimeä} : X \rightarrow [0,1] \quad (8)$$

$$y = \mu_{kimeä}(x) \quad (9)$$

Sumeaan joukko-oppiin enemmän tutustuvan kannattaa huomioida, että vaikka monet sumeille joukoille tehtävät operaatiot muistuttavat tilastotieteessä käytettyjä todennäköisyyslaskennan vastaavia operaatioita, sumeudessa ei ole kyse satunnaisuudesta, vaan epämääräisyydestä. Käsiteltävät alkioden arvot eivät siis sisällä satunnaista komponenttia, vaan niillä on aina tietyt ominaisuudet, mutta nämä ominaisuudet eivät välttämättä vastaa täsmällisesti jonkin määritettyä ominaisuutta tai ole kokonaan siitä erillisiä. Epämääräisyyden etuna voidaan pitää helppoa analogiaa luonnollisen kielen osittaista joukkoon kuulumista ilmaiseviin termeihin. Voidaan esimerkiksi erotella tapaukset ”työstö *hieman rauhatonta*”, ”työstö *rauhatonta*”, ja ”työstö *hyvin rauhatonta*”. [Mattila 2002 luku 1 ja luku 6.2, Zadeh 1965]

2.3 Sumeat konnektiivit

Sumeat konnektiivit ovat klassisen logiikan konnektiivien yleistyksiä siten, että niiden lähdeavaruus kattaa koko yksikkövälin. Konnektiivien tulee erikoistapauksena toimia kuten klassisen logiikan konnektiivit, mikäli niiden syötteenä on vain teräviä arvoja **0** tai **1** (jotka tulkitaan arvoiksi ”epätosi” ja ”tosi” tai joukko-opillisessa tulkinnassa joukkoon kuulumattomuudeksi tai sen jäsenyydeksi) sekä täyttää konnektiiveille asetetut ehdot sovelletusti koko yksikkövälinällä. Koska näiden sääntöjen puitteissa on monia tapoja toteuttaa yksittäiset konnektiivit, on tärkeää määrittää minkä tyyppisiä konnektiivejä sumeiden sääntöjen laadinnassa on käytetty.

Tämän työn piirissä sumeana konnektiivina ”ja” käytetään funktiota ”maksimi” (kaava 10). Sumeana konnektiivina ”tai” käytetään funktiota ”minimi” (kaava 11). Tarvittaessa konnektiiveista käytetään myös useampaa kuin kaksipaikkaista versiota siten, että prioriteti osoitetaan aina käyttämällä sulkumerkkejä.

$$\text{ja}(y_1, y_2) = \max(y_1, y_2) \quad (10)$$

$$\text{tai}(y_1, y_2) = \min(y_1, y_2) \quad (11)$$

Tietenkin, koska sekä operaatiot minimi että maksimi ovat vaihdannaisia, mikäli arvoilla ei ole tarpeen esittää erityistä tärkeysjärjestystä jostakin muusta syystä, voidaan sulkumerkkien merkitseminen ohittaa. Perinteisten konnektiivien ”ja” ja ”tai” lisäksi on mahdollisuus käyttää myös konnektiivejä, jotka sijoittuvat näiden välimaastoon: Tästä lisää luvussa 2.5. [Mattila 2002]

2.4 Aggregaatiosta

Useamman sumean arvon yhdistämistä sanotaan aggregoinniksi (kaava 12). Tämän työn yhteydessä käytetään aggregaatio-operaattorina painotettua keskiarvoa (kaava 13).

$$\mathbf{R}(y) = \text{Agg}(\mathbf{R}_1(y), \dots, \mathbf{R}_m(y)) \quad (12)$$

$$\mathbf{R}(y) = \frac{\sum_{j=1}^m w_j y_j}{\sum_{j=1}^m w_j} \quad (13)$$

Tarkalleen ottaen painotettu keskiarvo ei täytä täysin aggregaation vaatimuksia, sillä aggregaation pitäisi olla riippumaton sille syötettävien arvojen indekseistä [Mattila 2002 luku 8], mutta käytännön syistä katsottiin parhaaksi käyttää painotettua keskiarvoa. Koska painot ovat lukuja yksikköväälillä, ne voisi teoriassa antaa jo sumennusvaiheessa, eli muunnettaessa jotakin reaali maailman mitattua arvoa sumeaksi jäsenysarvoksi. Tällöin määriteltäisiin, että sumennuksen huipputaso ei olisikaan **1** vaan painoarvon esittämä määrä. Tämä tulkinta ei ole täydellinen, koska siitä puuttuu painotettuun keskiarvoon kuuluva jakajan muokkaaminen, joka kuitenkin voitaisiin ymmärtää saadulle arvolle tehtäväksi muunnokseksi. Joka tapauksessa olennaisinta on sen oikeellinen toiminta, joskin mikäli tulkitaan tehdyksi edellä mainitut toimenpiteet, täyttyvät myös yleisesti aggregaatiolle määritellyt ehdot.]

2.5 Ordered Weighted Average -operaattori

Ordered Weighted Average -operaattorit eli OWA-operaattorit (kaava 14) voidaan käsittää joko keskilukuina tai loogisten konnektiivien ”ja” ja ”tai” yleistyksinä. OWA-operaattori esitetään yleisesti haluttujen aggregoitavien arvojen vektorin y kanssa samanpituisena eli m -paikkaisena vektorina. OWA-operaattorin painokertoimien summa on aina yksi (kaava 15), joskin summan rajoite voidaan tarvittaessa kiertää normalisoimalla painokertoimet ennen laskutoimituksia.

$$OWA(y, w) = \frac{\sum_{j=1}^m w_j y_{lajiteltu, j}}{\sum_{j=1}^m w_j} = \sum_{j=1}^m w_j y_{lajiteltu, j} \quad (14)$$

$$\sum w = 1 \quad (15)$$

OWA-operaattorin erottaa tavallisesta painotetusta keskiarvosta annettujen sumeiden arvojen lajitteleminen laskevaan suuruusjärjestykseen ennen keskiarvon laskemista, minkä takia OWA-operaattorilla voidaan tehdä joitakin asioita, joita (painotetulla) keskiarvolla ei voi tehdä.

Esimerkiksi OWA-operaattori $[1, 0, \dots, 0]$ vastaa max-operaatiota, ja siten myös logiikan operaatiota ”tai”. OWA-operaattori $[0, \dots, 0, 1]$ vastaa min-operaatiota, ja siten myös logiikan operaatiota ”ja”. Näiden OWA-operaattorin kahden ääripään väliin voidaan sijoittaa ääretön määrä operaatioita, jotka ovat tai- ja ja-operaattorien välimuotoja. Tietenkin OWA-operaattoreita voisi luokitella muidenkin ominaisuuksien mukaan, mutta käytettäessä niitä päättelyyn, yksi niiden mielenkiintoisimmista piirteistä on niiden kyky toimia mainittujen tavallisesti käytettyjen konnektiivien välimuotoina. Puhutaankin käsitteistä *orness* (”tai-maisuus”) ja *andness* (”ja-maisuus”), jotka ilmaisevat, miten paljon operaattori vastaa kumpaakin. *Ornessin* laskukaava on esitetty kaavassa 16 ja *andnessin* laskeminen suhteessa *ornessiin* on esitetty kaavassa 17. Täydellisen ja-operaattorin ja tai-operaattorin välimuotoihin kuuluvat esimerkiksi aritmeettinen keskiarvo $[1/n, 1/n, \dots, 1/n]$ sekä mediaani (keskimmäinen alkio 1 tai parillisella määrällä alkioita kaksi keskimmäistä $1/2$). [Mattila 2002]

$$orness(\mathbf{w}) = \frac{1}{m-1} \sum_{j=1}^n (m-j)w_j \quad (16)$$

$$andness = 1 - orness \quad (17)$$

2.6 Sääto käyttäen useita antureita

Sorvauksen säätöön on kehitetty monenlaisia automatisoituja ratkaisuja, mutta usein ne ovat keskittyneet yhden tai vain muutaman signaalin, kuten tehon, seuraamiseen ja sen pitämiseen tietyissä rajoissa. Vaikka on olemassa jo joitakin järjestelmiä, jotka hyödyntävät useampaa sensoria, usein järjestelmän eri osat säätävät yksittäisiä eri asioita. [Lin & Lee 1999, Tarnig et al. 1996] Feedchip-hankkeessa seurataan useamman anturin mittauksia ja tunnistetaan niiden yhdistelmien perusteella eri piirteitä (sensor fusion). Tämä lisää järjestelmän vikasietoisuutta ja nostaa tunnistusvarmuutta, joskin tätä järjestelyä on käytetty tähän mennessä lähinnä laadunvalvontaan, esimerkiksi [Azouzi & Guillot 1997] ja terärikköjen tunnistamiseen [Lou & Lin 1996]. Kuitenkin yhteenvetona useat tutkimukset viittaavat huomattavasti parantuneeseen tunnistusvarmuuteen käytettäessä useampaa anturia päättelyn pohjana. Monet näistä järjestelmistä hyödyntävät neuroverkkoja ongelmatilanteiden tunnistamiseen. [Liang et al. 2004]

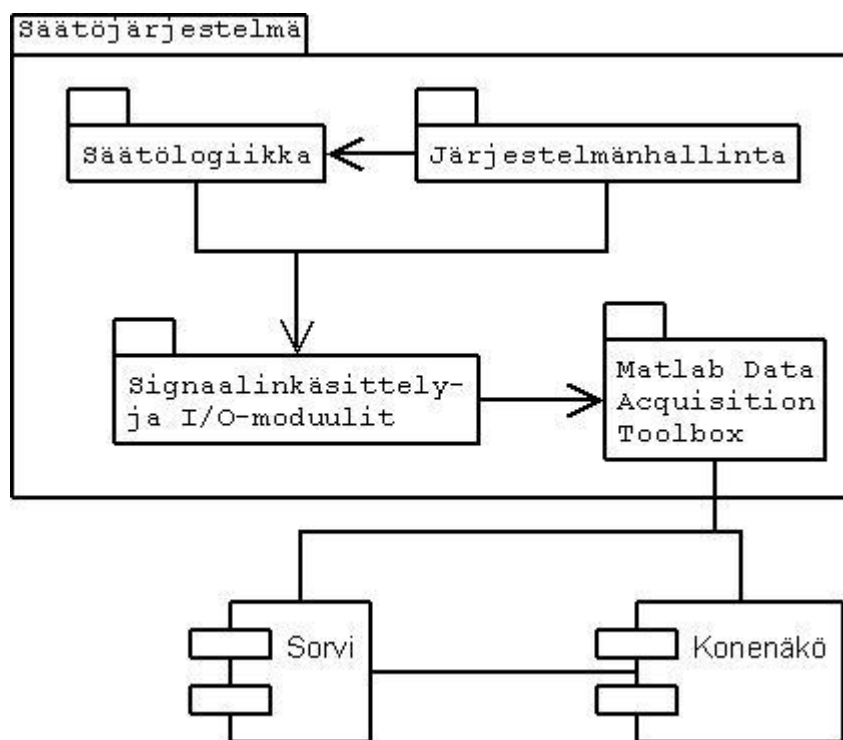
3 TYÖN TOTEUTUS

3.1 Yleistä

Työn käytännön osuus noudatti suurin piirtein vesiputousmallia, alkaen muodostamalla ensin ongelmanmäärittely (esitutkimus), vaatimusmäärittelydokumentti (vaatimusmäärittely), erilaisia luonnoksia järjestelmän toiminnasta (arkkitehtuurisuunnitelma) ja edeten lopulta toteutukseen. [Haikala & Märijärvi 2004 luku 2] Ohjelman tähänastiseen versioon tulevat ominaisuudet valmistuivat 19.11.2007 (feature freeze), minkä jälkeen testauksessa keskityttiin ohjelman parametrien säätöön ja löydettyjen vikojen korjaamiseen. Tämän takia tosin jouduttiin lisäämään vielä toiminnallisuutta, koska havaittiin lastuamisen säätötoimet jatkolastun katkaisemiseksi riittämättömäksi.

Ohjelman toteutuslupana ovat Mathworksin laskentatyökalut, siten että ohjelman korkean tason rakenne, käyttöliittymä ja tietovirrat ohjelman sisällä pääosin mallinnetaan Simulinkillä ja varsinainen toiminnallisuus Matlab-moduuleina Simulinkin level 1 S-function -rajapinnan kautta. Ohjelman ytimenä toimivat tilakoneen luomiseen käytettiin Stateflow-ohjelmistoa. Signaalinkäsittelyssä käytetään apuna Signal Processing -toolboxia (lähinnä Fourier-muunnoksen toteutus, mutta myös Welchin tehospektrimuunnosta tarvittaessa sekä kokeiltaessa suotimia). Signaalinkäsittelyssä käytetty mediaanisuodin (mukaillen esimerkiksi lähde [Reiter 2004]) on tekijän itse toteuttama. Rajapintana käytettäville antureille toimii Matlabin Data Acquisition Toolbox hyödyntäen National Instrumentsin NIDAQmx-ajuria. Anturit on yhdistetty tietokoneeseen National Instrumentsin PCI-6251 -mittauskortin kautta. Varsinainen sorvin lastuamisnopeuden ja syötön säätö tapahtuu liityntäkortin välityksellä, joka niin ikään on liitetty tietokoneeseen mittakortin kautta. Järjestelmän yleinen rakenne on esitetty kuvassa 1.

Luotettava ja tehokas lastuamisprosessin säätö toteutetaan seuraamalla antureiden signaaleita, ja laskemalla niistä tiettyjä mitta-arvoja (measure), kuten tehollisarvoja tai muita signaalin ominaisuuksia. Nämä mitatut arvot sumennetaan ja syötetään järjestelmälle, joka tunnistettujen piirteiden perusteella reagoi määriteltyihin ongelmatilanteisiin, eli liian alhaiseen lastuamisnopeuteen (kimeään ääneen), jatkolastuun ja rauhattomaan työstöön samalla pyrkien käyttämään koko käytettävissä olevan tehon. Tiettyä suurimman mahdollisen tehon rajaa ei kuitenkaan saa ylittää.

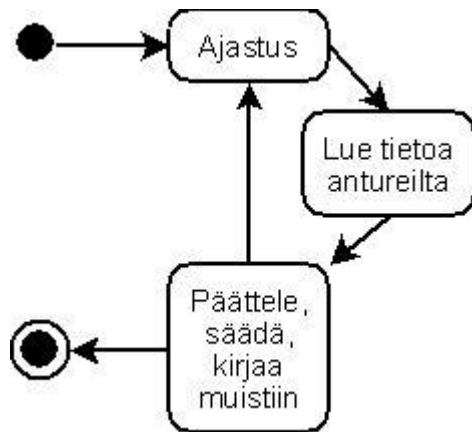


Kuva 1: Säätöjärjestelmän osat

3.2 Järjestelmänhallinta

Järjestelmänhallinnan sydän on tilakone (kuva 2), jolla säädetään mitä tapahtuu kullakin ajan hetkellä. Muut järjestelmän osat tiedottavat tarpeistaan lipuilla, mikä tässä tapauksessa tarkoittaa sorvin työstöohjelmaan sijoitettuja komentoja, joilla voidaan asettaa hetkeksi eräitä binäärimuotoisia arvoja eri tilaan. Tilakone valvoo näiden lippujen ilmaantumista sekä toimintojen ajastusta. Nykyisessä versiossa tilakoneen toiminta on sidottu tietokoneen kelloon siten, että sillä on mahdollisuus vaihtaa tilaa kahdesti sekunnissa. Pääosa muista moduuleista toimii vain tilakoneen liipaisemina, joskin osa on päällä koko ajan.

Tarvittaessa järjestelmänhallintamoduulissa on mahdollisuus liittää järjestelmään terän kulumista tarkkaileva konenäkökomponentti. Tämä komponentti toteutetaan Janne Laaksosen diplomityössä [Laaksonen 2007]. Koska terän kulumisen ei kuitenkaan liity tämän kandidaatintyön ydinalueeseen, joten konenäkökomponentti jätetään tässä työssä pääosin huomiotta. Kuitenkin, käsiteltäessä konenäöltä tulevaa informaatiota käytännön toimenpiteet sijoittautuisivat tilakoneessa erilliseen ajastuksesta alkavaan silmukkaan.



Kuva 2: Tilakone

Toiminnan ymmärtämiseksi on tärkeä ymmärtää taustalla pyörivän Simulinkin mallinnusympäristön toimintaa: Tavallisesti Simulink ei toimi reaaliajassa, vaan ajaa mallia niin nopeasti kuin mahdollista. Sitomalla tilakoneen toiminta tietokoneen kelloon ja muiden moduuleiden toiminta tilakoneeseen, on kuitenkin mahdollista ajaa mallia vain pienellä vaivalla reaaliaikaisena. Ratkaisu ei ole kuitenkaan täydellinen. Jäljelle jääneet osat toimivat edelleen niin nopeasti kuin mahdollista, joten Simulink täyttää aina kaiken vapaan suoritustehon. Tämä tarkoittaa, että mallin viemää todellista laskentatehoa voi olla vaikea määrittää, joskin joitakin osoittimia voidaan saada simulaation reaaliaikaisen kellon etenemisen vertailusta Simulinkin simulaation aikaan. Ääriarajoilleen vietyinä, yhden sekunnin aikana tulisi tapahtua yksi simulaatiokierros (kun niitä nyt tapahtuu useimmiten satoja). Tämän arvion tarkkuutta heikentää kuitenkin se, että vain simulaatiokierroksilla, joilla jotakin todella liipaistaan, tapahtuu raskaampaa laskentaa, joka joissakin tapauksessa kestää useamman sekunnin. Raskaiden jaksojen vaikutus on helppo havaita käyttöliittymässä olevan kellon lukeman epätasaisesta päivittymisestä.

Tilakoneen lisäksi järjestelmänhallinta huolehtii sekä lastuamisarvojen rajoitteiden hallinnasta että syötteestä sorville, lokiin ja ruudulle. Käyttöliittymä on myös toteutettu rakenteellisesti samalla tasolla järjestelmänhallinnan kanssa - Simulinkissä simulaatiomalli muodostaa samalla oman käyttöliittymänsä.

3.2.1 Reaaliaikaisen kellon toteutus

Koska simulaation oma kello ei toimi halutulla taajuudella, joudutaan muodostamaan oma kellopulssi toimintojen ajastamiseksi. Tämän on toteutettu siten, että simulaatio kysyy kellonaikaa

järjestelmäkelloilta. Pulssi muodostetaan syöttämällä sekunnit sinifunktiolle, jonka muuttujan kertoimien avulla voidaan säätää pulssin tiheyttä. Esimerkiksi käytettäessä funktiota $\sin(2\pi t)$ siniaallon jakso on yksi sekunti. Kun säädetään liipaisu tapahtumaan aina, kun saapuva signaali ohittaa nollakohdan negatiiviselta puolelta positiiviselle puolelle, saadaan liipaisu tapahtumaan kerran joka aallonpituuden aikana eli samalla taajuudella kuin pulssikin. Periaatteessa tähän voisi käyttää myös Simulinkin omia työkaluja, mutta koska järjestelmän tilakone on toteutettu Stateflowlla, tätä varten tulisi hankkia lisäosa, joka mahdollistaa Stateflow:n muodostaman koodin kääntämisen reaaliaikaisessa ajossa käytetyn lisäosan ymmärtämään muotoon.

Varsinaiset aikalaskurit toimivat tilakoneen sisällä. Koska tilakone voi muuttaa tilaa sekunnin välein, näiden laskureiden tarkkuus on täten +/- 1 s olettaen, että kaikki mallin kerralla liipaistavat toiminnot voidaan suorittaa yhden sekunnin aikana. Mikäli toimintoihin kuluu enemmän aikaa, laskuri pysyy ajassaan, mutta kellon päivitys pysähtyy hetkeksi. Tällä tavoin lasketaan terän lastuamisaikaa, joka on määritelty aikana, jolloin teho on huomattavasti nolaa suurempi. Lisäksi samalla periaatteella seurataan säätöohjelman kokonaisajoaikaa. Nämä kellot päivittyvät siis aina, kun tilakone muuttaa tilaa, laskien tietokoneen kellon ilmoittaman ajan ja edellisen tilanmuutoksen ajan erotuksen, lisäten sen omaan arvoonsa. Nämä kellot sekä tietokoneen ilmoittama kellonaika tulostetaan näytölle, sekunnit pyöristettynä alaspäin lähimpään kokonaislukuun. Terän lastuamisajan laskurin yhteydessä täytyy toki muistaa, että tehon arvon mittaukset tapahtuvat pienin väliajoin, joten käytännössä teränkestolaskurin tarkkuus on +/- 5 s joka lastuamiskerralta.

3.2.2 Kirjanpito ja tietojen syöttö muille järjestelmän osille (output)

Ohjelmaan kuuluva päättelyjärjestelmä (kuvassa 1 ”säätölogiikka”) kykenee lastuamisprosessia, mutta pelkän tiedon tuottamisen lisäksi kykenee myös päättämään tarvittavista säätötoimista mahdollisesti esiintyvien ongelmatilanteiden korjaamiseksi. Kun päättelyjärjestelmä on muodostanut päätöksen, seuraavalla tilakoneen tilan vaihdolla siirrytään toteuttamaan sitä. Tässä yhteydessä kirjoitetaan tehty päätös sekä lastuamisen ongelmatilanteiden esiintymisen sumennetut arvot lokiin, joka tallennetaan levyille. Lokin otsikkotiedot (päivämäärä, kellonaika, sorvin lastuamisarvot) on kirjoitettu, kun säätöohjelman ajo on alkanut. Loki on aina nimeltään `logfile.txt`, mutta tietojen häviämättömyys (vanhan lokin ollessa vielä paikallaan, kun uuden kirjoitus alkaa) varmistetaan siten, että syöttö tiedostoon tapahtuu aina lisäävänä, eli uudet tiedot kirjoitetaan mahdollisesti tiedostossa jo olevien tietojen perään. Epätäydellinen ote tästä lokista on

liitteessä kolme. Tämän lisäksi järjestelmä kirjoittaa toisen lokin nimeltään `sensorlog.txt`, johon tallennetaan antureiden tehollisarvot.

Samassa yhteydessä ohjausarvot syötetään takaisin sorville, minkä lisäksi päätetään rajoitteista. Mikäli jompaakumpaa tai molempia säädettäviä lastuamisarvoja on ollut tarpeen muuttaa, lastuamisarvon vanha arvo määritetään rajoitteeksi. Tämä estää lastuamisarvojen poukkoilun arvosta toiseen, mikäli jo heti seuraavalla kierroksella ohjelma muuten päättäisi muuttaa jostakin syystä arvon takaisin. Rajojen kesto määritetään myös samassa yhteydessä. Mikäli mitään muutosta ei ole tapahtunut ja aikaa on kulunut tarpeeksi kauan, voidaan määritellyt rajat unohtaa, jolloin on jälleen mahdollista palata vanhoihin arvoihin. Joidenkin ongelmatilanteiden ilmettyä rajat ovat kuitenkin pysyviä.

Järjestelmään on toteutettu myös mahdollisuus tallentaa varsinaiset mittaukset, jolloin voidaan tarkastella päättelyjärjestelmän toimintaa muutetuin päättelyn parametrein ilman, että kokeet tarvitsee kokonaan uusiksi. Tämä mahdollisuus ei kuitenkaan ole vakiona käytössä näiden mittausten vaatiman suuremman levytilan vuoksi.

3.3 Signaalinkäsittely

Signaalinkäsittelyn tehtävä on muuntaa antureilta saatu raaka data helpommin ymmärrettävään ja käsiteltävään muotoon, sekä erotella signaaleista seurattavista ongelmatilanteista kertovia ilmiöitä. Anturointi on toteutettu Juho Pirneksen diplomityössä [Pirnes 2006] ja alustava signaalien tulkinta Andrey Stepanovin diplomityössä [Stepanov 2007].

Varsinaisena signaalien mittauksen matalan tason toteutuksen rajapintana käytetään Matlabin Data Acquisition Toolboxia. Säättöohjelmaan tuonnin jälkeen ensimmäisenä signaaleille suoritetaan virheenkorjaus, joka tapahtuu suodattamalla ne kolmen näytteen pituisella mediaanisuodattimella. Näennäisestä yksinkertaisuudesta huolimatta tässä työn vaiheessa oli joitakin ongelmia, johtuen käytetystä ohjelman rakenteesta, joka ei kuitenkaan sallinut tietojen lukua taustalla. Tausta-ajon mahdollisuushan on olemassa, mutta käytännössä tämä vaatii samassa (Matlab-)funktiossa pysymistä. Tämä tarkoittaa Simulink-ympäristössä sitä, että koko mallin loppuosa odottaa toiminnon saattamista loppuun, joten vaikka luku tapahtuu ”taustalla” funktion osalta, käytännössä ei muita ohjelman osia voida ajaa sillä välin. Kesken projektin vaihdettiin uudempaan Matlabin

versioon, joka tukee Simulinkia suoraan, joten ohjelman mahdollisissa tulevilla versioissa tietojen siirto Data Acquisition Toolboxin moottorista ohjelmalle onnistunee helpommin.

Kaikkein yksinkertaisin signaalinkäsittelytoimenpide on RMS- eli Root Mean Square -arvon laskeminen. Tämä tapahtuu jollekin vektorina esitetylle signaalille luvussa 2.1 kaavassa 5 mainitulla tavalla laskemalla sen 2-normi (eli vektorin tapauksessa pituus) ja jakamalla näin saatu arvo tehtyjen mittausten (vektorin ulottuvuuksien) määrän neliöjuurella. Tämä arvo lasketaan joka signaalille. Tehdyissä kokeissa havaittiin, että mediaanisuodatus vaikuttaa akustista emissiota lukuun ottamatta mitattujen arvojen RMS:ään. Tämän takia RMS:nä esitettyjä tehollisarvoja käytettäessä tulee ottaa huomioon, minkälaista suodatusta on käytetty.

Toinen, akustisen emission ja mikrofonin signaaleille tehtävä toimenpide on signaalien tehospektrin määrittäminen. Signaalin tehospektri kertoo nimensä mukaisesti miten paljon tehoa mikäkin signaalin komponentti sisältäisi, jos signaali jaettaisiin eri taajuisiin siniaaltoihin. Käytännössä tämä tapahtuu laskemalla signaalin Fourier-muunnos, joka on tässä toteutettu Matlabin valmiilla Fast Fourier Transform -implementaatiolla, ja tämän jälkeen kertomalla näin saatu kompleksivektori alkioittain omalla kompleksikonjugaatillaan. Näin saatu tehospektrin estimaatti skaalataan käytetylle taajuusakselin osalle, jotta saadaan selville yhteys vektorin indeksin ja varsinaisen taajuuden välillä, ja tämän jälkeen tarkastellaan kiinnostavampia taajuusalueita. Pohjautuen tutkimukseen [Stepanov 2007] tarkastelemme 200 – 300 Hz taajuusaluetta lastun katkeamisen toteamiseksi sekä hieman esitettyä laajempaa taajuusaluetta 3700 – 5000 Hz kimeän äänen havaitsemiseksi. Matalammalta alueelta tarkastellaan signaalin keskimääräistä tehospektriestimaatin arvoa, ja korkeammalta alueelta suurinta tehopiikkiä. Käytetyt mittaukset ovat Fast Fourier Transform -algoritmin käytetyn implementaation takia pituudeltaan 2^{16} eli 65536 näytettä, mikä tarkoittaa 40 kHz näytteistystaajuudella noin 1,6 sekunnin mittauksia.

Tavallaan taajuuskaistan valinta tehospektristä toimii kaistanpäästönä. Vaihtoehtoinen tapa olisi ollut kaistanpäästösuotimien toteuttaminen ja tehollisarvojen laskeminen kaistoista ja näiden arvojen suhteuttaminen koko signaalin tehollisarvoon, mutta tehospektriestimaatti laskemalla signaalille tarvitsee tehdä vähemmän toimenpiteitä, koska tehospektriestimaatti tarvitaan joka tapauksessa valittujen signaalien analysointimenetelmien takia. Jättämällä suodatukset pois saavutetaan hienoinen etu laskunopeudessa, joskin käytännössä kysymys on vain sekunnin kymmenyksistä. Yksittäiset suodatukset toimivat nopeammin kuin Fourier-muunnos, mutta koska Fourier-muunnos tarvitaan vain kerran koko yksittäiselle mitatulle signaalille, toimii se nopeammin

kuin sarja suodattimia. Tulevaisuudessa voi olla tarpeen muodostaa tarkempi menetelmä jatkolastun havaitsemiseksi, ja tämä menetelmä voi käyttää joitakin muita keinoja kuin tässä esiteltyjä.

Mikäli matalan eli 200 - 300 Hz kaistan spektrin keskimääräinen teho on huomattavan alhainen, voidaan päätellä, että kyseisellä taajuudella ei ole selvää huippua spektrissä ja siten lastun katkeaminen on epätasaista tai sitä ei tapahdu ollenkaan. Tätä arvoa ei voi kovinkaan yksinkertaisesti päätellä huippujen perusteella, koska myös joissakin jatkolastutapauksissa esiintyy alueella huippuja, ja joka tapauksessa alueella esiintyy vähintäänkin yksi huippu, jonka tulkittiin vastaavan sorvin ominaisvärähtelytaajuutta. Vaihtoehtoisesti voidaan jakaa signaalin alhainen kaista alueisiin, ja pyrkiä tästä tunnistamaan huippuja. Tässä tapauksessa on huomioitava, että spektrin teho laskee korkeampiin taajuuksiin päin siirryttäessä ja täten tulee tämä trendi kompensoida huippua etsittäessä. Tämä menetelmä vaatii jonkin verran jatkotutkimusta, sillä kokeissa se osoittautui riittämättömäksi.

Korkealla kaistalla puolestaan keskitytään löytämään suurin piikki tehospektrissä. Huomattavan korkean piikin löytyminen tarkoittaa kimeän äänen ja sitä vastaavien värähtelyjen esiintymistä. Alun perin tutkimustulokset [Stepanov 2007] viittasivat piikin esiintymiseen alueella noin neljä kilohertsiä, mutta koeaineiston uudelleentarkastelu paljasti, että kimeästä äänestä kertovia piikkejä esiintyy pääasiassa kahdella kaistalla, 3,7 – 4,0 kHz ja 4,5 – 5,0 kHz akustisen emission ja mikrofonin signaaleissa. Myös muilla alueilla esiintyi piikkejä, mutta tarkempi tarkastelu paljasti, että näitä piikkejä esiintyy myös muissa kokeissa kuin niissä, joissa kimeän äänen esiintyminen oli todettu, ja siten niiden pohjalta ei voida päätellä ilmiön esiintymisestä mitään. Raskaan työstön ääni peittää kimeän äänen alleen ja täten häiritsee kimeän äänen mittausta. Tämä raskas ääni tulisi myös tunnistaa, jotta sen esiintyminen voitaisiin kompensoida. Mikäli tämä tapaus pystytään tunnistamaan, voitaneen vähentää kimeiden äänien vääriä tunnistamisia korkeilla syötön arvoilla. Osittain tätä pyritään vähentämään myös suhteuttamalla mitattuja arvoja signaalin RMS:ään, jolloin tehon nousua pystytään osittain kompensoimaan.

3.4 Päättelyjärjestelmä

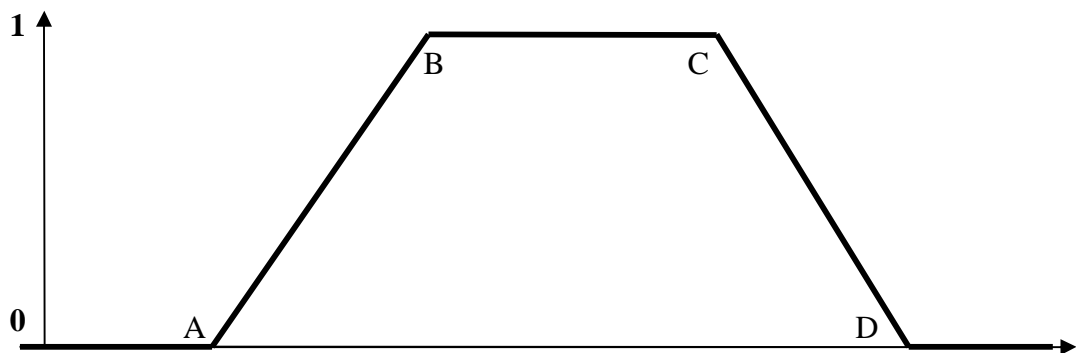
Päättelyjärjestelmän tehtävä on erottaa mitatuista ja käsitellyistä signaaleista tietyt lastuamisen ongelmatilanteet ja pyrkiä suorittamaan määritellyt toimenpiteet, joilla nämä ongelmatilanteet

voidaan poistaa. Säännöstö pohjautuu Johanna Leppänen diplomityöhön [Leppänen 2007] siten, että diplomityössä esitetty säännöstö on pyritty yleistämään sumeaksi säännöstöksi.

Jotta päättelyjärjestelmä pystyy laskemaan jonkin päätöksen, täytyy sille syötetyt arvot ensin nimellisarvostaan muuntaa totuusarvoksi. Tätä prosessia kutsutaan sumean logiikan yhteydessä sumentamiseksi, ja totuusarvo voi olla mikä tahansa arvo yksikkövälillä. Matemaattisesti ilmaistuna, ensimmäiseksi tarkistetaan kuuluuko syötetty signaali johonkin sumeaan joukkoon.

3.4.1 Lastuamisprosessin ongelmatilanteiden esiintymisen määrittäminen

Tässä järjestelmässä kaikki sumeat joukot ovat mitattavan suureen suhteen suunnikkaan muotoisia. Tämä tarkoittaa, että kun joukon jäsenyysarvot alkavat nousta pisteessä a, ne nousevat lineaarisesti kunnes saavuttavat täyden jäsenyysasteen pisteessä b. Välillä [b, c] kaikille signaaleille jäsenyysaste on 1. Jälleen välillä [c, d] jäsenyysaste laskee yhdestä takaisin nolnaan. Muodostunut kuvio (kuvassa 3) on siis nelikulmio, jonka mitattavan suureen ylä- ja alasivut ovat yhdensuuntaisia, korkeus totuusarvojen suunnassa yksi yksikkö, ja akselista kauempi sivu lyhyempi kuin akselin päällä oleva sivu. On myös mahdollista, että b ja c ovat sama luku, jolloin kyseessä on kolmio, jonka yksi sivu on vasten akselia.



Kuva 3: Sumennus suunnikas- eli trapetsoidimenetelmällä.

Kun mitatun signaalin jäsenyysaste jossakin joukossa on todettu, voidaan sitä edelleen käsitellä. Tässä järjestelmässä jatkokäsittely tapahtuu painotetun keskiarvon avulla. Periaatteessa keskiarvon sijasta voisi käyttää jotakin (sumeaa) loogista operaattoria kuten JA tai TAI, tai jotakin OWA-operaattoria (Ordered Weighted Average). OWA-operaattoreilla on mahdollista toteuttaa erilaisia keskilukuja sekä myös mainitut loogiset operaattorit, mutta painotettuun keskiarvoon päädyttiin,

koska on todettu, että jotkin signaalit ovat luotettavampia kuin toiset. Tällöin voidaan painotettu keskiarvo suoraan säätää käyttämään enemmän jotakin tiettyä arvoa. Lisäksi tällä menetelmällä voidaan tarvittaessa kytkeä jokin signaali kokonaan pois päältä antamalla sille painoksi nolla. Tämä painotettu yhdiste eri signaalien jäsenyysarvoista jossakin sumeassa joukossa välitetään edelleen varsinaiselle päättelyjärjestelmälle.

On myös mahdollista, että jotakin signaalia käytetään useamman sumean arvon laskemiseen. Esimerkiksi pätehon suuntaista kiihtyvyyttä (ACC1) käytetään kaikkien arvojen laskemiseen, minkä lisäksi liian matalan lastuamisnopeuden tapauksessa lasketaan kaksi päättelyjärjestelmälle välitettävää arvoa, joista toinen ottaa huomioon akustisen emission ja mikrofonin signaalin, ja toinen huomioi vain pätehon suuntaisen kiihtyvyyssignaalin.

Sumennuksen vaatimat parametrit ovat siis neljä arvoa joka sumentajalle (a, b, c, d) sekä painoarvo jokaisen yhdisteen jokaiselle osalle (eli sumennetulle signaalille). Tietenkin mikäli jokin arvo päätellään vain yhden signaalin perusteella, ei painoarvoa tarvita.

3.4.2 Jatkolastun havaitseminen

Poikkeukseen edellä mainittuun menetelmään muodostaa tällä hetkellä jatkolastun havaitseminen. Alustavissa testeissä jatkolastua ei pystytty merkitsevällä tavalla havaitsemaan mitatusta datasta tai valitun sumennusmenetelmän rajoissa sumentamaan mitattuja vaihteluita, joten yritettiin työlle varatun ajan puitteissa muodostaa malli jatkolastun esiintymiselle. Malli (kaava 18) käyttää tässä vaiheessa syöteinään akustista emissiota, mikrofonia sekä kiihtyvyyssantureita, ja on muodostettu pienimmän neliösumman sovituksena (kaavat 19 ja 20). Mallin oikeellisuutta testattiin vertailemalla sen mitatun syötteen x perusteella muodostamia tuloksia $\mu(x)$ jatkolastun esiintymiseen y . Regressioanalyysin keinoin todettiin mallin selittävän noin 93 % siihenastisissa kokeissa esiintyneistä vaihteluista (ks. liite 1).

$$y(x) \approx \mu(x) = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 \quad (18)$$

$$LSQ(\vec{y}, \vec{x}) = \sum_{j=1}^n ((y)_j - \mu((x)_j))^2 \quad (19)$$

$$\min_b LSQ(\vec{y}, \vec{x}) \quad (20)$$

Koska tämänkaltainen tilastollinen ”black box” -malli ei ole hyvin yleistettävissä, malli tulisikin korvata signaalinkäsittelyllisillä toimenpiteillä, jotta toiminto voidaan yleistää. Käytännössä malli toimi hyvin osalla kokeista, mutta vaatii kuitenkin jonkin verran (vähintään kymmeniä) kokeita kalibrointiaan varten, minkä lisäksi tämänkin jälkeen malli toimii vain kokeillulla alueella (interpolointi). Jouduttaessa kokeillun alueen ulkopuolelle (ekstrapolointi) malli hyvin antaa virheellisiä tuloksia, kuten jouduttiin toteamaan. Lopuksi päädyttiin muodostamaan terävä joukkoon kuulumisen tarkistus pelkän akustisen emission perusteella.

Varsinainen päättelyjärjestelmä jakaa sille syötetyt sumeat totuusarvot (eli jäsenyysarvot sumeassa joukossa) lastuamisnopeuden ja syötön säätöön tarvittaviin, ja tämän jälkeen laskee painotetut totuusarvot. Kertoimina käytetyt totuusarvot ovat myös yksikkövälin arvoja, jotta varmistetaan, että lopullinen painotettu arvo on myös yksikkövälin totuusarvo, ja sitä voidaan näin tarvittaessa jatkokäsittelä sumean logiikan menetelmillä. Nämä painotetut totuusarvot järjestetään sen mukaan, kumpaa säädettävistä kahdesta lastuamisparametrasta pitäisi niiden perustella säätää. Tämän jälkeen kummastakin listasta valitaan suurin arvo. Lopuksi suoritetaan havaitulle ja valitulle ongelmatilanteelle määritetyt korjaustoimenpiteet. Painoarvojen käyttö varmistaa, että tärkeämpiin ongelmatilanteisiin reagoidaan ensin, mutta mikäli tärkeämpi tilanne ei ole kovinkaan vakava, voidaan tarvittaessa reagoida myös vakavampaan mutta vähemmän tärkeään tilanteeseen. Mikäli samalla painotetun totuusarvon lukuarvolla on useampia ongelmatilanteita, reagoidaan siihen, jonka painoarvo on suurempi. Algoritmi on esitetty liitteessä kaksi.

3.4.3 Erikoistapauksia

Erikoistapauksena matalaan lastuamisesta mitattuun tehoon perustuvaa syötön nostoa käsitellään hieman eri tavalla kuin muita. Syöttöä ei kannata nostaa, mikäli värähtelyitä esiintyy, joten painoarvoa sovelletaan totuusarvoon aliteho ja ei paljoa värähtelyjä. Alitehoon reagoidaan siis vain tapauksissa, joissa on havaittu vain hieman tai ei ollenkaan rauhatonta työstöä.

Toisena erikoistapauksena on liian pieni lastuamisnopeus (joka aiheuttaa kimeän äänen sekä mittausten kannalta haitallisia värähtelyjä työstettävään kappaleeseen) tunnistetaan säännöllä kimeä

ääni ”ja” kiihtyvyydsarvot koholla. Konnektiivi ”ja” on lainausmerkeissä, koska käytetään korkean *andness*-arvon OWA-operaattoria, eikä varsinaista sumeaa ja-operaattoria eli minimiä.

3.4.4 Sääntöjen liipaisu ja säädön oikeellisuuden tarkistaminen

Jotta vältetään rauhaton säätöjen poukkoilu, ennen varsinaista toteutussuositusta kuitenkin tarkistetaan, ovatko lastuamisnopeuden ja syötön ongelmatilanteiden painottamattomat totuusarvot tarpeeksi suuria. Käytännössä niitä verrataan johonkin annettuun liipaisuarvoon, ja arvoa suuremmat tai sen kanssa yhtä suuret totuusarvot liipaisevat toiminnon. Painottamatonta totuusarvoa joudutaan käyttämään, koska muuten selkeästi havaittu ongelmatilanne, jolla on pieni painoarvo, ei välttämättä ikinä liipaisisi järjestelmää.

Lopuksi tarkistetaan toteutussuositusten oikeellisuus. Tämä tapahtuu vertaamalla toteutettavaa toimintoa käyttäjän syöttämiin järjestelmän rajoihin ja järjestelmän ennalta tunnistamiin rajoitteisiin. Näitä rajoja ovat syötön rajat f_{\min} ja f_{\max} sekä lastuamisnopeuden rajat v_{\min} ja v_{\max} seuraavissa muodoissaan: Käyttäjän määrittelemät rajat, edellisten havaintojen perusteella asetetut rajat ja sorvin rajoitteet. Käyttäjän määrittelemät rajat tulisi syöttää käytettävän terän mukaan. Sorvin rajat varmistavat, että käytettävät säädöt ovat sorvin asetuksissa sallitulla alueella. Lopuksi seurataan järjestelmän säätöhistoriaa ja pyritään pysymään poissa aiemmin huonoiksi havaituista säädöistä. Näiden rajoitteiden lisäksi tehon suurin sallittu arvo toteutetaan omana sääntönään. Jos sorvin rajoitteet estävät vaaditut säädöt, järjestelmä ilmoittaa virheestä. Tässä tapauksessa tulisi työstö lopettaa, mutta käytännön vaikeuksien takia tämä toiminnallisuus puuttuu toistaiseksi säätöjärjestelmästä. Mikäli kyseessä on vain tavanomainen virhetilanne, jossa todetaan säädön aiheuttavan toisen, jo aiemmin korjatun virhetilanteen, työstön annetaan jatkua, mutta ilmoitetaan kuitenkin virhetilanteesta.

3.4.5 Korjaustoimenpiteet

Mikäli lastuamisen ongelmatilanne on havaittu, ja päättelyjärjestelmä on määrittänyt, että niihin tulee puuttua, suoritetaan seuraavat korjaustoimenpiteet. Korjaustoimenpiteet ovat määritelleet Feedchip-tutkimushankkeen lastuamisasiantuntijat Johanna Leppänen ja Juha Turku suorittamiensa lastuamiskokeiden perusteella. Mikäli kyseessä on kimeä ääni, nostetaan lastuamisnopeutta 10 %. Mikäli kyseessä on jatkolastu, suoritetaan hieman monimutkaisempia säätöjä. Ensin syöttö

kytketään pois puoleksi sekunniksi, minkä jälkeen sitä nostetaan. Mikäli syöttö on sallittujen rajojen puolivälin alapuolella, sitä nostetaan puoliväliin. Mikäli se on puolivälin yläpuolella, sitä nostetaan 10 %. Mikäli se on maksimiarvossaan, syötön katkaisun jälkeen se pidetään siellä. Rauhattoman työstön yhteydessä taas puolestaan lasketaan syöttöä 10 %.

Kun jokin lastuamisen ongelmatilanne on havaittu, järjestelmä merkitsee sen esiintymisen muistiin ja korjaa rajoja siten, että kyseisillä lastuamisarvoilla ei lastuta enää samassa ajossa uudestaan. Koska jotkin virhetilanteet voivat ajan myötä poistua, tulee järjestelmän unohtaa tunnistamansa virhetilanteet tietyn ajan kuluttua. Rajoitteiden käsittely on kuitenkin sijoitettu järjestelmänhallintamoduulille ja siten ei ole päättelyjärjestelmän piirissä. Huomattavaa on, että lisäksi jatkolastun korjaustoimenpiteet eivät välitä näistä rajoitteista, koska jatkolastun esiintyminen voi aiheuttaa vaaratilanteen, mikäli jatkolastu takertuu työstökoneeseen.

3.5 Tulokset

Tähänastisissa testeissä (liite kolme) ohjelmisto on osoittanut pystyvänsä tunnistamaan haluttuja piirteitä sekä aiemmin tehdyistä mittauksista että työstönaikaisesti antureiden avulla meneillään olevasta työstöprosessista, huolimatta ohjelman päättelyjärjestelmän hyvin yksinkertaisesta loogisesta rakenteesta. Virhemarginaali on tosin suuri, mutta kuitenkin on käytännössä osoitettu, että kuvatus kaltainen järjestelmä on toimintakykyinen. On huomautettava, että tämän raportin kirjoittamisen aikana järjestelmää ei ole vielä täysin testattu, mutta tähänastisissa testeissä järjestelmä on toiminut hyvin. Kimeän äänen tunnistus on toiminut, minkä lisäksi jatkolastun tunnistusta on yritetty muutamilla eri menetelmillä. Lopulta jatkolastun tunnistuksessa on päädytty seuraamaan akustisen emission arvoa, muodostaen terävä raja-arvo signaalille. Jatkokehitystä ajatellen tulisi löytää yleistettävämpi menetelmä. Tässä menetelmässä käytetään todennäköisesti hyödyksi signaalin tehospektriä.

Keskittyminen säädön vaatimuksiin on nostanut esiin joitakin havaintoja signaalien luonteesta ja ominaisuuksista, kuten mikrofonin- ja AE -signaalien spektrin hyödyntäminen jatkolastun ja kimeän äänen tunnistamiseksi. Erityisesti mikrofonisignaalin käyttäytymisessä on tosin havaittu suuria häiriöitä, joita ei kokonaan ymmärretä. Kokeet on kuitenkin toteutettu (suhteessa konepajan normaalitilanteeseen) kohtuullisen häiriöttömässä ympäristössä.

Itse säätöjärjestelmä on toimintakykyinen hyvin yksinkertaisenakin, ja lisäämällä säätöjä ja muokkaamalla varsinaista säännöstöä monipuolisemmaksi pystytään reagoimaan hyvin monipuolisesti työstössä mahdollisesti ilmeneviin ongelmatilanteisiin. Kimeä ääni pystytään tunnistamaan piikkinä akustisen emission ja mikrofonin spektrissä, joskin tämä tunnistus muuttuu hieman epävarmemmaksi suuremmilla lastuamisarvoilla. Suhteuttamalla piikin korkeus vastaavan suureen tehollisarvoon päästään kuitenkin luotettavampiin tuloksiin. Jatkolastu ilmenee spektrin 200 - 300 Hz alueella tavallisesti havaittavan trendiä korkeampina tehospektrin arvoina. Lisäksi jatkolastu voidaan päätellä tietyinä tehollisarvojen yhdistelmänä. Tehollisarvoissa jatkolastun tunnusomaiset piirteet ovat hieman koholla olevat kiihtyvyyssarvot sekä matalat akustisen emission ja äänisignaalin arvot. Jatkolastun esiintyessä akustisen emission keskimääräinen arvo signaalin RMS:n mukaan mitattuna on aina alle 5 V käytössä olevalla laitteistolla. Rauhaton työstö näkyy jokaisessa signaalissa, joskin sen tarkkaa tunnistusta ei vielä ole saavutettu.

Jotkin lastuamisen ongelmatilanteiden joukot ovat päällekkäisiä hyvän lastuamisen joukon kanssa, mikä saattaa aiheuttaa vääriä ongelmatilanteiden tunnistamisia. Tunnistuksen varmentamiseksi tulee käyttää useaa anturia yhteistyössä ja vertailla, miten niistä tunnistetut ongelmatilanteet suhtautuvat toisiinsa. Ohjelmiston testausvaiheessa havaittiin ominaisuuksien useista signaaleista tunnistamisen (feature extraction) lisäävän huomattavasti järjestelmän vikasietoisuutta, joskin edelleen joitakin ongelmia on jäänyt järjestelmään, huomattavimpana väärät kimeän äänen havainnot työstettäessä tavallista suuremmalla (0,7 mm/r) syötöllä. Tämä johtuu siitä, että työstön raskaus nostaa tavallisen työstön tuottamat mittausarvot tasolle, jossa normaalitilanteessa esiintyisi kimeää ääntä. Lisäksi kuten mainittua, jatkolastun tunnistaminen oli erityisen ongelmallista, ja lopulta tämän kandidaatintyön rajoissa jouduttiin luopumaan heikon lastunmurron tunnistamisesta ja tyydyttiin jatkolastun terävään tunnistamiseen akustisen emission signaalista. Erityisesti tämä alue kaipaa lisäkehitystä.

Mainittujen ongelmien lisäksi havaittiin rauhattoman työstön aiheuttamia signaaleita mitattaessa ilmeisesti vain raskaan työstön yhteydessä esiintyvä kimeän äänen kaltainen viheltävä ääni. Ilmeisesti tämä ääni voitaisiin havaita ainakin kohonneina kiihtyvyyssarvoina. Vaikka ääni tulee saada poistettua, äänen tarkka tunnistaminen ei mahdu tämän opinnäytetyön piiriin.

4 POHDINTAA

4.1 Tulosten hyödyntäminen

Feedchipin yhteydessä Lappeenrannan teknillisen yliopiston konepajatekniikan NC-sorviin instrumentoidun prototyypilaitteiston toiminta osoittaa, että tässä ja muissa Feedchip-hankkeeseen liittyvissä töissä kuvatun kaltainen järjestelmä on mahdollista toteuttaa. Prototyypin jatkokehittäminen kaupallisen tasoiseksi tuotteeksi on mahdollista, joskin vaatii vielä huomattavasti töitä. Projektissa on löydetty mitattujen signaalien joukosta mikrofonisignaali, akustisen emission signaali ja kiihtyvyyden signaalit, joita voidaan käyttää edelleen. Signaalinkäsittelyn ja päättelyjärjestelmän rakenteen suhteen on vielä runsaasti kehitettävää, joskin jo nykyisellään järjestelmä pystyy tekemään päätöksiä ja selviytymään omin avuin joistakin ongelmatilanteista tai niiden yhdistelmistä.

4.2 Jatkotutkimusaiheita

Toistaiseksi on hieman epäselvää miten pystytään tunnistamaan päällekkäiset tapaukset, joiden aiheuttamat muutokset signaaleihin häiritsevät toisiaan. Lisäksi häiriöiden tai lastuamisessa ilmenevien satunnaisten vaihteluiden aiheuttamista vääristä tunnistuksista tulisi päästä eroon.

Päättelyjärjestelmä on huomattavan yksinkertainen. Muodostamalla rakenteeltaan monipuolisempi säännöstö on mahdollista tehdä säädöistä huomattavasti kehittyneempiä. Näin voidaan myös turvallisemmin varmistaa, että vakavimpiin tuloksiin reagoidaan ensin, ja että tuloksiin reagoidaan oikeassa järjestyksessä. Lisäksi järjestelmään ei välttämättä vielä voida täysin liittää kuvausta ”oppiva”, joskin järjestelmä oppii säädöt, joihin sen ei kannata palata. On mahdollista kehittää järjestelmän osaksi oppiva järjestelmä, joka asettaa kohdalleen sumean logiikan käyttämät parametrit, jolloin järjestelmä on helpompi siirtää toimimaan eri terälastuamismateriaaliyhdistelmille.

Joissakin signaaleissa on lisäksi huomattavaa epätarkkuutta. Samoilla lastuamisarvoilla toistetuissa kokeissa esiintyi jatkolastun tunnistuksessa huomattavaa vaihtelua kokeilluilla sumeilla tunnistimilla. Mikäli pystytään määrittämään eri signaalien virheiden jakaumat, voidaan määrittää

tehokkaammin näiden signaalien tunnistusrajat. Tämä edellyttää signaalien luottamusvälien tunnistusta, sekä kokeita joissa pyritään löytämään eri tunnistettavien ongelmatilanteiden yleistettävät piirteet. Itse lastuamistapahtumaa lienee myös mahdollista mallintaa siten, että havaitaan mahdolliset yhteydet lastuamisarvojen, ongelmatilanteiden ja signaalien välillä, tai mahdollisesti jopa edellä mainittujen ja terä-lastuamismateriaaliyhdistelmien välillä. Myös terän kuluneisuuden aiheuttamat muutokset signaaleihin tulisi selvittää, jotta voidaan lisätä tunnistuksen luotettavuutta. Lisäksi havaittiin raskaan työstön yhteydessä esiintyvä viheltävä ääni, joka tulee poistaa. Tämä häiriö ei näy riittävän yksiselitteisesti nyt kehitetyissä tunnistusjärjestelmissä, joten se tulee erikseen löytää signaaleista ja kehittää säännöstö, joka ottaa ilmiön esiintymisen huomioon ja kykenee poistamaan sen.

Järjestelmän käytännön jatkokehityksessä on monia asioita, jotka voidaan tehdä paremmin. Nyt kun järjestelmän toimintaperiaate tunnetaan paremmin kuin projektin alussa, voidaan järjestelmän rajapintoja siistiä siten, että tarpeettomat parametrit poistetaan sekä uudet kehityksen loppuvaiheessa asetetut parametrit lisätään. Yhteen liittyvät osat voidaan siirtää arkkitehtonisesti samaan paikkaan, jolloin vähennetään tarvittavien rajapintojen määrää. Käyttöliittymän suunnitteluun tulee kiinnittää hieman enemmän huomiota, esimerkiksi osan palautteesta voi antaa äänenä. Anturien luku tulisi vihdoin saada taustalle (tarkoitukseen sopivien Simulink-blokkien avulla). Lisäksi monet ominaisuudet, kuten rajoitteiden asettaminen, voidaan vaihtoehtoisesti kiertää asettamalla parempi säännöstö sekä sumennus, joka tunnistaa juuri ilmenemässä olevat tapaukset siten, että kyetään ennakoimaan ongelmatilanteet ennen kuin ne esiintyvät jäykkien rajoitusten sijaan.

5 YHTEENVETO

Työn tuloksena on syntynyt ohjelmisto, joka kykenee tunnistamaan sorvauksessa esiintyviä ongelmatilanteita. Ohjelmisto testattiin projektissa aiemmin instrumentoidun NC-sorvin avulla, johon liitettiin ohjelmaa suorittava tietokone. Järjestelmä kykenee toimimaan häiriöiden poistamiseksi pohjautuen kiihtyvyyden, akustisen emission ja äänen signaaleihin sekä niistä tehtyihin tarkempiin analyyseihin.

Varsinaisen signaalienkäsittelytoimintojen ja päättelymoottorin lisäksi työn yhteydessä on toteutettu Matlab-rajapinta antureille sekä toiminnoiltaan yksinkertainen, mutta käytännön syistä rakenteeltaan hieman monimutkainen käyttöliittymä ja analyyseja varten kattavat lokitoiminnot. Kaikesta huolimatta täytyy todeta, että työlle on tyypillisesti käytössä rajallisesti aikaa, eikä tämäkään opinnäytetyö ole poikkeus. Tätä kirjoitettaessa rauhattoman työstön tunnistus on testaamatta, joten ohjelmistotuotannon realiteetit huomioiden ei voida sanoa kaikkien työn aloituksessa asetettujen tavoitteiden täytyneen. Yksittäisten toimintojen tasolla kaikki järjestelmän moduulit on tosin testattu. Koko järjestelmän tasolla tähän mennessä ajettujen testien perusteella on kuitenkin todettu, että pelkkään olemassa olevaan aineistoon perustuvassa testauksessa ei luonnollisesti voida huomioida työstössä mukana olevaa sattuman elementtiä.

Toisaalta kaikki tärkeimmät tavoitteet on saavutettu minkä lisäksi järjestelmä kykenee nyt ja tämän työn aikana tehtyjen havaintojen pohjalta tulevaisuudessa ainakin antamaan lähes reaaliaikaista palautetta sekä tunnistamaan entistä tarkemmin eri ongelmatilanteiden ominaispiirteitä. Työn aikana on myös kokeiltu eri menetelmiä työstössä esiintyvien ongelmatilanteiden piirteiden erottamiseksi työstön aikana syntyvistä signaaleista, mitä pystytään toivottavasti käyttämään hyödyksi tulevaisuudessa pyrittäessä ymmärtämään työstössä esiintyviä ilmiöitä ja tämän ymmärtämyksen perusteella tekemään uusia, entistä tehokkaampaan ja laadukkaamman työstön mahdollistavia järjestelmiä.

LÄHDELUETTELO

- [Azouzi & Guillot 1997] Azouzi, R. ja Guillot, M.: On-line Prediction of Surface Finish and Dimensional Deviation in Turning Using Neural Network Based Sensor Fusion
International Journal of Machine Tools and Manufacture, Volume 37, Issue 9. Elsevier Science Ltd., 1997
- [Haikala & Märijärvi 2004]
Haikala, Ilkka ja Märijärvi, Jukka: Ohjelmistotuotanto
Kymmenes, uudistettu painos, Talentum, 2004. ISBN 952-14-0850-2
- [Huttunen 2005] Huttunen, Heikki: Signaalinkäsittelyn menetelmät
Tampereen teknillinen yliopisto, 2005. ISSN 1459-4609, ISBN 952-15-1432-9
- [Kreyzig 1999] Kreyzig, Erwin: Advanced Engineering Mathematics, 8th Edition
John Wiley & Sons, Inc., 1999. ISBN 0-471-15496-2
- [Laaksonen 2007] Laaksonen, Janne: Toistaiseksi nimeämätön diplomityö.
Diplomityö, Lappeenrannan teknillinen yliopisto, 2007
- [Leppänen 2007] Leppänen, Johanna: Adaptiivisen rouhintasorvausjärjestelmän säätörakenteen määrittäminen
Diplomityö, Tampereen teknillinen yliopisto, 2007
- [Liang et al. 2004] Liang, Steven Y.; Hecker, Rogelio L.; Landers, Robert G.: Machining Process Monitoring and Control: The State-of-the-Art
Journal of Manufacturing Science and Engineering, Volume 126, Issue 2. ASME, 2004
- [Lin & Lee 1999] Lin, Lih-Chang ja Lee, Gwo-Yuh: Hierarchical Fuzzy Control for C-Axis of CNC Turning Centers Using Genetic Algorithms

Journal of Intelligent and Robotic Systems 25, Kluwer Academic Publishers, 1999

- [Lou & Lin 1996] Lou, Kang-Ning ja Lin, Cheng-Jen: An intelligent sensor fusion system for tool monitoring on a machining center
Multisensor Fusion and Integration for Intelligent Systems, 1996.
IEEE/SICE/RSJ International Conference on 8-11 Dec. 1996 sivut 208 - 214
- [Mattila 2002] Mattila, Jorma K.: Sumean logiikan oppikirja – Johdatus sumean matematiikkaan
Kolmas, uusittu painos. Art House, 2002. ISBN 951-884-300-7
- [Pirnes 2006] Pirnes, Juho Mikael: Automaattiseen mittaus- ja säätöjärjestelmään liittyvän laiteympäristön rakentaminen NC-sorville
Diplomityö, Lappeenrannan teknillinen yliopisto, 2007
- [Reiter 2004] Reiter, Cliff: With J: Image Processing 1: Smoothing Filters
ACM SIGAPL APL Quote Quad archive, Volume 34, Issue 2, sivut 9 - 15.
Association for Computing Machinery, 2004. ISSN 0163-6006
- [Stepanov 2007] Stepanov, Andrey: Adaptive Real-Time Control of Machinery Operation Using Multiple Sensors Input
Diplomityö, Lappeenrannan teknillinen yliopisto 2007
- [Tarng et al. 1996] Tarng, Y. S.; Lin, C. Y. ja Nian, C. Y.: An Optimization Approach for the Fuzzy Control of Turning Operations
Fuzzy Sets and Systems, Volume 83, Issue 3, Elsevier Science B.V., 1996
- [Welch 1967] Welch, P. D.: The use of fast Fourier transforms for the estimation of power spectra: A method based on time averaging over short modified periodograms

IEEE Transactions on Audio and Electroacoustics, vol. 15, pp. 70-73,
1967

[Zadeh 1965]

Zadeh, Lotfi A.: Fuzzy Sets
Information and Control, Issue 8, 1965

LIITE 1: JATKOLASTUN ESIINTYMISSÄ DATAAN SOVITETTU MALLI JA SEN VARIANSSIANALYYSI

Analyysissä käytetty StatGraphics-ohjelmistoa.

Kahden ensimmäisen jatkolastukokeen perusteella:

Multiple Regression Analysis

Dependent variable: Jatkolastu

Parameter	Estimate	Standard Error	T Statistic	P-Value
CONSTANT	3,7334	0,156486	23,8577	0,0000
AE	-0,392727	0,101002	-3,88832	0,0002
MIC	-8,37633	1,80107	-4,65075	0,0000
ACC1	0,348658	0,126229	2,76209	0,0070
ACC2	1,86915	0,48998	3,81475	0,0003

Analysis of Variance

Source	SSQ	Df	MSQ	F-Ratio	P-Value
Model	19,5488	4	4,88719	283,18	0,0000
Residual	1,4842	86	0,0172582		
Total (Corr.)	21,033	90			

R-squared = 92,9434 percent

R-squared (adjusted for d.f.) = 92,6152 percent

Standard Error of Est. = 0,13137

Mean absolute error = 0,0872027

Durbin-Watson statistic = 1,63602

$$\text{Jatkolastu} = 3,7334 - 0,392727 \cdot \text{AE} - 8,37633 \cdot \text{MIC} + 0,348658 \cdot \text{ACC1} + 1,86915 \cdot \text{ACC2}$$

Data

Koe 1: Johanna Leppäsen aiemmat kulumiskokeet, ei jatkolastua, kimeää ääntä tai rauhatonta työstyä

Koe 2: Ensimmäinen jatkolastukoe

Koe 3: Toinen jatkolastukoe

AE: AE-signaalin RMS

AElo: AE-signaalin tehospektriestimaatin keskiarvo kaistalla 200 - 300 Hz

MIC: MIC-signaalin RMS

MIClo: MIC-signaalin tehospektriestimaatin keskiarvo kaistalla 200 - 300 Hz

ACC1: Päätehon suuntaisen kiihtyvyyden signaalin RMS

ACC2: Syöttötehon suuntaisen kiihtyvyyden signaalin RMS

Jatkolastu: Jatkolastun esiintyminen

AE	AElo	MIC	MIClo	ACC1	ACC2	Jatkolastu	Koe
5,31	198601,22	0,24	227,26	0,18	0,17	0	1
5,43	214102,4	0,24	231,13	0,22	0,21	0	1
4,41	110654,14	0,19	204,18	0,15	0,14	0	1
5,36	188091,36	0,24	227,86	0,24	0,18	0	1
5,36	185425,75	0,24	221,42	0,26	0,19	0	1
5,40	188851,31	0,24	259,92	0,38	0,16	0	1
5,33	210881,44	0,24	218,54	0,28	0,17	0	1
5,46	159165,86	0,24	246,01	0,37	0,17	0	1
5,46	163944,91	0,24	243,76	0,37	0,17	0	1
5,63	165159,73	0,25	101,22	0,21	0,18	0	1
5,50	203573,6	0,25	198,71	0,37	0,17	0	1
5,44	185800,04	0,24	218,91	0,36	0,17	0	1
5,47	190359,1	0,24	240,98	0,38	0,18	0	1
5,51	196981,2	0,25	222,4	0,33	0,19	0	1
5,47	180466,15	0,24	219,21	0,37	0,18	0	1
5,54	222139,59	0,25	248,12	0,38	0,19	0	1
5,47	189867,61	0,24	228,93	0,30	0,18	0	1
5,50	195294,38	0,25	194,17	0,37	0,18	0	1
5,33	166210,12	0,24	207,08	0,31	0,18	0	1
5,48	181835,86	0,25	196,11	0,37	0,19	0	1
5,35	187930,48	0,24	203,96	0,33	0,18	0	1
5,47	209657,37	0,24	211,93	0,36	0,19	0	1
5,46	224010,67	0,24	185,55	0,38	0,17	0	1
5,47	223779,72	0,24	219,64	0,38	0,19	0	1
5,47	180595,43	0,24	183,82	0,36	0,17	0	1
5,44	165500,96	0,24	180,8	0,35	0,17	0	1
5,55	203292,08	0,25	120,35	0,21	0,2	0	1

AE	AElo	MIC	MIClo	ACC1	ACC2	Jatkolastu	Koe
5,53	209048,09	0,25	186,07	0,38	0,18	0	1
5,52	206371,81	0,25	128,81	0,21	0,2	0	1
5,53	205484,44	0,25	100,55	0,24	0,18	0	1
5,49	230692,25	0,25	162,78	0,26	0,2	0	1
5,32	170536,23	0,24	177,83	0,35	0,17	0	1
5,26	171370,32	0,23	178,22	0,35	0,17	0	1
5,45	208915,51	0,24	164,62	0,38	0,17	0	1
5,30	198353,73	0,24	176,62	0,36	0,17	0	1
5,43	206039,77	0,24	163,21	0,38	0,16	0	1
5,45	210912,15	0,24	158,54	0,38	0,16	0	1
5,34	222213,66	0,24	154,22	0,38	0,15	0	1
5,50	224925,94	0,25	113,23	0,27	0,17	0	1
5,32	159670,65	0,24	163,5	0,36	0,17	0	1
5,29	193479,71	0,24	164,3	0,36	0,17	0	1
5,40	220017,86	0,24	145,56	0,38	0,15	0	1
5,50	215019,24	0,25	171,05	0,37	0,2	0	1
5,45	198252,52	0,24	146,52	0,38	0,16	0	1
5,34	187427,93	0,24	153,89	0,36	0,17	0	1
5,38	197643,4	0,24	152,54	0,37	0,18	0	1
5,34	202146,29	0,24	150,43	0,37	0,17	0	1
5,36	172389,4	0,24	131,42	0,38	0,15	0	1
5,50	209632,33	0,25	98,2	0,29	0,18	0	1
5,51	159007,12	0,25	118,08	0,33	0,17	0	1
5,51	196368,1	0,25	93,16	0,32	0,17	0	1
5,36	187926,78	0,24	138,62	0,37	0,18	0	1
5,47	206097,93	0,24	106,12	0,31	0,18	0	1
5,44	197989,81	0,24	126,67	0,39	0,17	0	1
5,51	216044,86	0,25	142,96	0,34	0,21	0	1
5,27	178183,78	0,23	135,96	0,37	0,17	0	1
5,52	209628,9	0,25	135,72	0,37	0,22	0	1
5,53	213926,85	0,25	132,09	0,31	0,22	0	1
3,86	10423,73	0,17	230,29	0,14	0,1	1	2
4,01	8063,99	0,18	248,84	0,17	0,1	1	2
4,04	6490,44	0,18	271,11	0,23	0,1	1	2
4,70	119750,84	0,21	187,78	0,75	0,25	1	2
3,99	6758,15	0,18	102,13	0,14	0,09	1	2
4,08	7425,24	0,18	247,72	0,42	0,09	1	2
4,80	38745,3	0,21	227,26	1,18	0,32	1	2
4,66	142903,18	0,21	141,41	0,72	0,26	1	2
3,84	12521,12	0,17	86,05	0,27	0,11	1	2
4,24	16561,98	0,19	268,81	0,29	0,09	1	2
3,91	8714,98	0,17	91,69	0,35	0,09	1	2
3,87	20189,06	0,17	90,47	0,38	0,13	1	2
4,66	29944,79	0,21	181,8	0,25	0,3	1	2
4,43	17513,14	0,2	105,63	0,32	0,12	1	2
4,3	11341,64	0,17	34,78	0,16	0,14	1	3
4,01	10522,75	0,16	41,11	0,25	0,15	1	3
4,29	13314,05	0,17	44,37	0,29	0,14	1	3
4,08	10491,55	0,16	219,23	0,18	0,15	1	3

AE	AElo	MIC	MIClo	ACC1	ACC2	Jatkolastu	Koe
4,13	31337,58	0,16	227,64	0,2	0,12	1	3
4,03	2017,77	0,16	193,49	0,21	0,14	1	3
3,92	27901	0,15	41,49	0,23	0,1	1	3
4,86	41885,75	0,19	87,2	0,26	0,19	1	3
4,61	6916,4	0,18	41,31	0,37	0,15	1	3
4,32	11385,98	0,17	73,48	0,37	0,11	1	3
4,31	6980	0,17	101,37	0,38	0,15	1	3
4,31	8000,4	0,17	156,6	0,38	0,14	1	3
4,26	6337,33	0,17	178,84	0,15	0,12	1	3
4,32	7610,16	0,17	195,23	0,19	0,09	1	3
4,4	10898,95	0,17	191,82	0,26	0,1	1	3
4,3	27982,37	0,17	187,72	0,31	0,12	1	3
4,28	13293,27	0,17	188,87	0,32	0,1	1	3
4,41	10988,55	0,17	187,05	0,34	0,1	1	3
4,25	4983,32	0,17	171,39	0,37	0,11	1	3

LIITE 2: KÄYTETTY PÄÄTTELYALGORITMI PSEUDOKOODINA

```
# kommentti          \jatkuu
program päättele(signaalit, painoarvot, rajoitteet, liipaisuarvot) \
  returns ongelmatilanne, päättelydata
begin päättele
  for j := 1 to ongelmatilanteiden_määrä
    ongelmatilanteen_esiintyminen(j) := sumenna(ongelmatilanne(j), signaalit);
  endfor
  ongelmatilanne := 0, 0;
  päättelydata := [ongelmatilanteen_esiintyminen, ei_virheitä];
  if(max(ongelmatilanteen_esiintyminen)<min(liipaisuarvot))
    return;
  elseif(signaalit(teho)<lastuamisen_vähimmäisteho)
    päättelydata := [ongelmatilanteen_esiintyminen, virhe_ei_lastuamista];
    # ei lastuamista, ei säätöä
    return;
  else
    for j := 1 to ongelmatilanteiden_määrä
      painotettu_ongelmatilanteen_esiintyminen = \
        painoarvot(j)*ongelmatilanteiden_esiintyminen(j);
    endfor
    painotettu_ongelmatilanteiden_esiintyminen(matala_teho) := \
      painoarvot(matala_teho)*(ongelmatilanteiden_esiintyminen(matala_teho) AND \
        NOT ongelmatilanteiden_esiintyminen(rauhaton_työstö));
    painotettu_ongelmatilanteiden_esiintyminen(kimeä_ääni) := \
      painoarvot(kimeä_ääni)*OWA(ongelmatilanteiden_esiintyminen(kimeä_ääni), \
        ongelmatilanteiden_esiintyminen(kiihtyvyyks_koholla);
    # Tässä käytetyn OWA:n andness on 0.9 - muistuttaa hyvin paljon minimiä
    ongelmatilanne(1) := \
      maxval_index(painotettu_ongelmatilanteiden_esiintyminen(syöttö));
    ongelmatilanne(2) := \
      maxval_index(painotettu_ongelmatilanteiden_esiintyminen(lastnop));
    if ongelman_esiintyminen(ongelmatilanne(1))<liipaisuarvot(ongelmatilanne(1));
      ongelmatilanne(1) := 0;
    endif
    if ongelmatilanteiden_esiintyminen(ongelmatilanne(2)) \
      < liipaisuarvot(ongelmatilanne(2));
      ongelmatilanne(2) := 0;
    endif
    if(tarkasta_rajat(rajoitteet, ongelmatilanne(1))=raja_ylitetty)
      syöttö_rajat_ylitetty := 1;
    endif
    if(tarkasta_rajat(rajoitteet, ongelmatilanne(2))=raja_ylitetty)
      lastuamisnopeus_rajat_ylitetty := 1;
    endif
    if(syöttö_rajat_ylitetty AND NOT lastuamisnopeus_rajat_ylitetty)
      virhe := virhe_syötön_rajat_ylitetty;
    elseif(NOT syöttö_rajat_ylitetty AND lastuamisnopeus_rajat_ylitetty)
      virhe := virhe_lastuamisnopeuden_rajat_ylitetty;
    elseif(syöttö_rajat_ylitetty AND lastuamisnopeus_rajat_ylitetty)
      virhe := virhe_molemmat_rajat_ylitetty;
    else
      virhe := ei_virheitä
    endif
    päättelydata := [ongelman_esiintyminen, virhe];
  endif
  return;
endprogram.
```


LIITE 3: KOETULOKSIA (SÄÄDÖN DEMO-OHJELMA)

Tämän liitteen koetulokset on saatu aamupäivällä 30.11. järjestetyissä kokeissa, joissa testattiin järjestelmän kykyä tunnistaa ja korjata jatkolastua ja kimeää ääntä. Liitteessä on vain ote koetuloksista, ei kaikkia tuloksia. Tästä otteesta on karsittu joitakin toistoja sekä epäolennaisia sarakkeita (mm. alhaisen tehon tunnistus ja diagnostiikkaa). Täysi loki on muutamia kertoja tämän työn varsinaisen osan pituinen.

Mitä tuloksista näkyy ja mitä ei näy? Tuloksista näkyy reagointi jatkolastun ilmenemiseen (syötön säätö) ja reagointi kimeään ääneen (lastuamisnopeuden säätö). Kuitenkin, jatkuvasti havaitsematta kimeään äänen tapauksessa jäi pieni vihellys. Tämä viheltävä ääni jäi jäljelle kovemman kimeään äänen korjauksen jälkeenkin, vaikka se olisi pitänyt korjata.

Selite:

Time: Aika

Parameters: Lastuamisparametrit (alkuarvo ja rajat)

Fs: Näytteistystaajuus

Time: Aika säätöjärjestelmän käynnistämisestä

FR: Syöttö, FRx10 % alkuperäisestä

Rule: Säätö, jonka perusteella syöttöä on säädetty. Jatkolastu on säätö 1.

SP: Lastuamisnopeus, SPx10 % alkuperäisestä

WN: Kimeään äänen arvioitu taso (sumeaa totuusarvo)

ACC: Päätehon suuntaisen kiihtyvyyden oleminen liian koholla (sumeaa totuusarvo)

NCB: Jatkolastun esiintyminen (terävä totuusarvo)

LogicError: Virhe päättelyssä. Virhe 4 tarkoittaa sitä, että sorvi ei lastua (hyvin alhainen teho).

Time: 9:59:01

Parameters:

vmin: 105

v: 140

vmax: 170

a: 3

fmin: 0,3

f: 0,3

fmax: 0,7

Fs: 20000

Time	FR	Rule	SP	Rule	WN	ACC	NCB	LogicError
8	10	0	10	0	0,3333	0	1	4
13	10	0	10	0	0,3333	0,1732	1	4
18	15	1	10	0	0	0,1068	1	0
23	15	0	10	0	0	0,0546	0	0
28	15	0	10	0	0	0,0524	0	0
33	15	0	10	0	0	0,16	0	0
38	15	0	10	0	0	0,2345	0	0
43	15	0	10	0	0	0,2871	0	0
48	15	0	10	0	0	0,319	0	0
53	15	0	10	0	0	0,3401	0	0
58	15	0	10	0	0	0,3584	0	0
63	15	0	10	0	0	0,3675	0	0
68	15	0	10	0	0	0,3741	0	0

Time: 10:10:46

Parameters:

vmin: 105

v: 105

```

vmax: 170
a: 3
fmin: 0,3
f: 0,5
fmax: 0,7
Fs: 20000
Time FR Rule SP Rule WN ACC NCB LogicError
8 10 0 10 0 0,3333 0 1 4
13 10 0 11 2 0,8498 0,8215 0 0
18 10 0 12 2 0,7198 0,7467 0 0
23 10 0 12 0 0 0,1524 0 0
28 10 0 12 0 0 0,232 0 0
33 10 0 12 0 0 0,3056 0 0
38 10 0 12 0 0 0,3466 0 0
43 10 0 12 0 0 0,383 0 0
48 10 0 12 0 0 0,4017 0 0
53 10 0 12 0 0 0,4109 0 0
58 10 0 12 0 0 0,4162 0 0
63 10 0 12 0 0 0,4287 0 0
68 10 0 12 0 0 0,4242 0 0

```

Time: 10:12:08

Parameters:

```

vmin: 105
v: 105
vmax: 170
a: 3
fmin: 0,3
f: 0,5
fmax: 0,7
Fs: 20000
Time FR Rule SP Rule WN ACC NCB LogicError
8 10 0 10 0 0,3333 0 1 4
13 10 0 10 0 0,3333 0,0976 1 4
18 10 0 11 2 0,9647 0,8855 0 0
23 10 0 12 2 0,6667 0,7576 0 0
28 10 0 12 0 0,0109 0,1095 0 0
33 10 0 12 0 0,0072 0,1894 0 0
38 10 0 12 0 0 0,2725 0 0
43 10 0 12 0 0,0006 0,3227 0 0
48 10 0 12 0 0 0,3565 0 0
53 10 0 12 0 0 0,3917 0 0
58 10 0 12 0 0 0,4012 0 0
63 10 0 12 0 0 0,4104 0 0
68 10 0 12 0 0 0,4294 0 0
73 10 0 12 0 0 0,4276 0 0
78 10 0 12 0 0 0,4234 0 0
83 10 0 12 0 0 0,4304 0 0

```

Time: 10:13:53

Parameters:

```

vmin: 105
v: 140
vmax: 170
a: 3
fmin: 0,3
f: 0,3
fmax: 0,7
Fs: 20000
Time FR Rule SP Rule WN ACC NCB LogicError
8 10 0 10 0 0,0358 0 1 4

```

13	10 0	10 0	0,1841	0,1617	1	4
18	15 1	10 0	0	0,2413	1	0
23	15 0	10 0	0	0	0	0
28	16 1	10 0	0	0	1	0
33	16 0	10 0	0	0,1353	0	0
38	16 0	10 0	0	0,2424	0	0
43	16 0	10 0	0	0,3306	0	0
48	17 1	10 0	0	0,3746	1	0
53	18 1	10 0	0	0,4273	1	0
58	18 0	10 0	0	0,4949	0	0
63	19 1	10 0	0	0,4919	1	0
68	20 1	10 0	0	0,6921	1	0
73	10 0	10 0	0,2609	0,2335	1	4

Time: 10:18:28

Parameters:

vmin: 105
v: 140
vmax: 170
a: 3
fmin: 0,3
f: 0,3
fmax: 0,7
Fs: 20000

Time	FR Rule	SP Rule	WN	ACC	NCB	LogicError
8	10 0	10 0	0,0313	0	1	4
13	15 1	10 0	0	0,0151	1	0
18	15 0	10 0	0	0,0288	0	0
23	15 0	10 0	0	0,0909	0	0
28	15 0	10 0	0	0,1882	0	0
33	15 0	10 0	0	0,2705	0	0
38	15 0	10 0	0	0,3146	0	0
43	15 0	10 0	0	0,3424	0	0
48	15 0	10 0	0	0,3612	0	0
53	15 0	10 0	0	0,3748	0	0
58	15 0	10 0	0	0,3824	0	0
63	15 0	10 0	0	0,3863	0	0
68	15 0	10 0	0	0,3824	0	0
73	15 0	10 0	0	0,3912	0	0
78	16 1	10 0	0,0073	0,3969	1	0

Time: 10:20:07

Parameters:

vmin: 105
v: 105
vmax: 170
a: 3
fmin: 0,3
f: 0,5
fmax: 0,7
Fs: 20000

Time	FR Rule	SP Rule	WN	ACC	NCB	LogicError
8	10 0	10 0	0,3333	0	1	4
13	10 0	11 2	1	0,739	0	0
18	10 0	12 2	0,7732	0,7399	0	0
23	10 0	12 0	0,1103	0,1827	0	0
28	10 0	12 0	0	0,2445	0	0
33	10 0	12 0	0,0404	0,3033	0	0
38	10 0	12 0	0	0,3575	0	0
43	10 0	12 0	0	0,3831	0	0
48	10 0	12 0	0,0284	0,3999	0	0

53	10 0	12 0	0	0,4111	0	0
58	10 0	12 0	0	0,4174	0	0
63	10 0	12 0	0	0,4297	0	0

Time: 10:21:21

Parameters:

vmin: 105
v: 105
vmax: 170
a: 3
fmin: 0,3
f: 0,5
fmax: 0,7
Fs: 20000

Time	FR Rule	SP Rule	WN	ACC	NCB	LogicError
8	10 0	10 0	0,3333	0	1	4
13	10 0	11 2	1	0,8012	0	0
18	10 0	12 2	0,9339	0,7895	0	0
23	10 0	12 0	0,192	0,177	0	0
28	10 0	12 0	0	0,237	0	0
33	10 0	12 0	0	0,3164	0	0
38	10 0	12 0	0	0,3476	0	0
43	10 0	12 0	0	0,3939	0	0
48	10 0	12 0	0	0,4034	0	0
53	10 0	12 0	0	0,4242	0	0
58	10 0	12 0	0	0,4462	0	0
63	10 0	12 0	0	0,4488	0	0
68	10 0	12 0	0	0,4568	0	0
73	11 1	12 0	0	0,4804	1	0

Time: 10:23:43

Parameters:

vmin: 105
v: 140
vmax: 170
a: 3
fmin: 0,3
f: 0,3
fmax: 0,7
Fs: 20000

Time	FR Rule	SP Rule	WN	ACC	NCB	LogicError
8	10 0	10 0	0,3333	0	1	4
13	15 1	10 0	0	0,0572	1	0
18	16 1	10 0	0	0	1	0
23	16 0	10 0	0	0,0581	0	0
28	16 0	10 0	0	0,1915	0	0
33	16 0	10 0	0	0,2643	0	0
38	16 0	10 0	0	0,3093	0	0
43	16 0	10 0	0	0,3487	0	0
48	16 0	10 0	0	0,3732	0	0
53	16 0	10 0	0	0,3866	0	0
58	16 0	10 0	0	0,3983	0	0

Time: 10:24:48

Parameters:

vmin: 105
v: 140
vmax: 170
a: 3
fmin: 0,3
f: 0,3

```

fmax:    0,7
Fs:      20000
Time     FR Rule  SP Rule  WN      ACC      NCB  LogicError
8        10 0     10 0     0,3333  0        1    4
13       10 0     10 0     0,3333  0,223    1    4
18       10 0     10 0     0,3333  0,4925   1    4
23       10 0     10 0     0,3333  0,6604   1    4
28       10 0     10 0     0,3333  0,7301   1    4
33       15 1     10 0     0        0,4814   1    0
38       15 0     10 0     0        0,0256   0    0
43       15 0     10 0     0        0        0    0
48       15 0     10 0     0        0,0819   0    0
53       15 0     10 0     0        0,1837   0    0
58       15 0     10 0     0        0,2612   0    0
63       15 0     10 0     0        0,3052   0    0
68       15 0     10 0     0        0,3325   0    0
73       15 0     10 0     0        0,3573   0    0
78       15 0     10 0     0        0,3628   0    0
83       15 0     10 0     0        0,3732   0    0
88       15 0     10 0     0        0,3785   0    0
93       10 0     10 0     0,0192  0,2331   1    4
98       10 0     10 0     0,2778  0        1    4
103      10 0     10 0     0,3333  0,232    1    4

```

Time: 10:26:54

Parameters:

```

vmin:    105
v:       115
vmax:    170
a:       3
fmin:    0,3
f:       0,5
fmax:    0,7
Fs:      20000

```

```

Time     FR Rule  SP Rule  WN      ACC      NCB  LogicError
8        10 0     10 0     0,3333  0        1    4
13       10 0     10 0     0,3333  0,2957   1    4
18       10 0     10 0     0,3333  0,5407   1    4
23       10 0     10 0     0,3333  0,691    1    4
28       10 0     10 0     0,3333  0,7853   1    4
33       10 0     10 0     0,3333  0,8411   1    4
38       10 0     10 0     0,3333  0,8287   1    4
43       10 0     10 0     0,3333  0,783    1    4
48       10 0     11 2     0,6667  1        0    0
53       10 0     11 0     0        0,2842   0    0
58       10 0     11 0     0,1215  0,1631   0    0
63       10 0     11 0     0        0,217    0    0
68       10 0     11 0     0        0,2916   0    0
73       10 0     11 0     0        0,3582   0    0
78       10 0     11 0     0        0,3989   0    0
83       10 0     11 0     0        0,422    0    0
88       10 0     11 0     0        0,4384   0    0
93       10 0     11 0     0        0,4555   0    0

```

Time: 10:29:12

Parameters:

```

vmin:    105
v:       115
vmax:    170
a:       3
fmin:    0,3

```

```

f:      0,5
fmax:   0,7
Fs:     20000
Time    FR Rule  SP Rule  WN      ACC      NCB  LogicError
8       10 0     10 0     0,3333  0        1    4
13      10 0     11 2     0,7307  0,7303  0    0
18      10 0     11 0     0        0,1456  0    0
23      10 0     11 0     0,0291  0,2079  0    0
28      10 0     11 0     0        0,2819  0    0
33      10 0     11 0     0        0,3484  0    0
38      10 0     11 0     0        0,3984  0    0
43      10 0     11 0     0        0,4358  0    0
48      10 0     11 0     0        0,4568  0    0
53      10 0     11 0     0        0,4834  0    0
58      10 0     11 0     0        0,492   0    0
63      10 0     11 0     0        0,4923  0    0

```

Time: 10:30:37

Parameters:

```

vmin:   105
v:      140
vmax:   170
a:      3
fmin:   0,3
f:      0,3
fmax:   0,7
Fs:     20000

```

```

Time    FR Rule  SP Rule  WN      ACC      NCB  LogicError
8       10 0     10 0     0,2703  0        1    4
13      10 0     10 0     0,197   0,151   1    4
18      15 1     10 0     0        0,0507  1    0
23      15 0     10 0     0        0        0    0
28      15 0     10 0     0        0,0532  0    0
33      15 0     10 0     0        0,1664  0    0
38      15 0     10 0     0        0,2538  0    0
43      15 0     10 0     0        0,295   0    0
48      15 0     10 0     0        0,3259  0    0
53      15 0     10 0     0        0,3462  0    0
58      15 0     10 0     0        0,359   0    0

```