

Heikki Korhonen

METSO OC TUNER -OHJELMISTON SUUNNITTELU JA
TOTEUTTAMINEN

Insinööriö
Kajaanin ammattikorkeakoulu
Tekniikka ja liikenne
Ajoneuvojen tietojärjestelmät
2013



Koulutusala Tekniikka ja liikenne	Koulutusohjelma Tietotekniikan insinööri
Tekijä(t) Heikki Korhonen	
Työn nimi Metso OC Tuner -ohjelmiston suunnittelu ja toteuttaminen	
Vaihtoehtoiset ammattiopinnot Ajoneuvojen tietojärjestelmät	Ohjaaja(t) Pentti Romppainen, Matti Törmänen Toimeksiantaja Metso Oyj
Aika 1.10.2012 - 31.1.2013	Sivumäärä ja liitteet 69 + 2
<p>Tämän insinöörytön tavoitteena oli suunnitella ja toteuttaa Metso OC -sakeusmittarille testausohjelmisto, jolla voidaan tarkistaa mittalaitteessa käytettävän kuitukaapelin liimauksen onnistuminen ja säätää kaapelissa oleva ledi optimaaliseen arvoon valovoimakkuuden perusteella. Ohjelmalla täytyi myös pystyä lukemaan mittalaitteelta sarjanumeroista mittausliikennettä virityksen yhteydessä. Virityksen lopuksi tarkastetusta ja viritetystä laitteesta täytyi tallentaa raportti, josta ilmenee laitteen viritystiedot ja sarjanumerot.</p> <p>Yksi paperi- ja selluteollisuuden haastavimmista ja kriittisimmistä säädöistä on massan sakeuden säätäminen. Sakeuden mittatarkkuutta huonontavat anturin likaantuminen, massan virtausnopeuden muutokset, näytteenotossa esiintyvät hankaluudet sekä anturin väsyminen ja kuluminen. Säätämistä haittaa mittaus- ja ohjauksen välillä oleva kuollut aika sekä sekoituskohtaan tulevan massan virtausmuutokset. Säätämisen kannalta massan sakeutta on mitattava jatkuvasti, vaikka näin toimimalla päädytään epäsuoriin mittausmenetelmiin. Sakeuden mittaamiseen on käytetty erilaisia menetelmiä jo pitkään, mutta ehdottoman hyvää menetelmää ei ole toistaiseksi löydetty.</p> <p>Metso OC -sakeusmittarissa käytetään valon siirtämiseen optista kuitukaapelia. Sen sisällä on seitsemän erillistä kuidun säätettä, joita pitkin valo lähetetään prosessiputkessa olevan yhteen läpi paperimassaan. Valo siirtyy massassa olevista kuiduista takaisin ja sironneen valon intensiteetin perusteella voidaan määrittää paperimassan sakeus. Koska kaapeli ja siinä olevien kuitujen läpimitta on todella pieni, on niitä mahdotonta havaita paljaalla silmällä katsottaessa. Kuidusta ei voi myöskään havaita valmistusvirheitä ilman lisälaitteita. Koska kaapeliin lähetettävä valo tulee yhdestä ledistä, on mahdollisia intensiteettieroja yksittäisten kuitujen kohdalla mahdotonta todeta ilman lisälaitteita.</p> <p>Työn lopputuloksena on syntynyt järjestelmä, joka vastaa tilaajaan tarpeita hyvin. Työssä luotu ohjelmisto varmistaa, että asiakas saa luotettavasti säädetyt ja kalibroidun Metso OC -sakeusmittarin ja jokaisesta lähetetystä laitteesta jää Metsolle kattava dokumentointi laitteen sarjanumeroista ja viritysarvoista. Metso OC Tuner -ohjelmisto on otettu käyttöön tammikuussa 2013.</p>	
Kieli	Suomi
Asiasanat	Sakeus, LabVIEW, konenäkö
Säilytyspaikka	<input checked="" type="checkbox"/> Verkkokirjasto Theseus <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto



School Engineering	Degree Programme Information Technology
Author(s) Heikki Korhonen	
Title Design and Implementation of Metso OC Tuner Software	
Optional Professional Studies Vehicle Information Systems	Instructor(s) Pentti Romppainen, Matti Törmänen
	Commissioned by Metso Oyj
Date 1.10.2012 - 31.1.2013	Total Number of Pages and Appendices 69 + 2
<p>The purpose of this Bachelor's thesis was to design and implement such testing software for the Metso OC consistency transmitter that enables one to check that the fiber optic cable has been glued flawlessly and to tune the LED at the end of the cable to the optimal value of brightness. In addition, the software must be able to read serial data from the device during the tuning process. Furthermore, after each check and tuning, a report indicating the device's tuning data and serial numbers needs to be created and saved.</p> <p>One of the most challenging and critical adjustments in the paper and pulp industry is the tuning of the consistency of the mass. The preciseness in the measurement of consistency is hindered by the dirtiness of the sensor, changes in the flow velocity of the mass, difficulties occurring in sampling as well as the wear and tear of the sensor. Adjustment is also hindered by the dead-time between the measurement point and the steering point, as well as changes in the flow of the mass into the mixing point. From the point of view of adjustment, the consistency of the mass must be measured continuously, although this necessitates the use of indirect measurement methods. Various methods have been used for a long time in the measurement of consistency, but so far no absolutely good method has been found.</p> <p>The Metso OC consistency transmitter utilizes a fiber optic cable for the transmission of light. The cable contains seven separate fiber threads in which light is transmitted into the paper mass through a coupling in the process pipe. The light scatters back from the fibers in the mass, and the intensity of the scattered light enables one to determine the consistency of the paper mass. Since the cable and the diameter of the fibers are very small, it is impossible to discern them with the naked eye. Neither can one detect manufacturing errors in the cable without special equipment. Since the light dispatched into the cable comes from a single LED, it is impossible to detect any intensity differences between individual fibers without additional equipment.</p> <p>The outcome of this study was a piece of software that corresponds to the commissioner's needs well. The software created by the author ensures that the customer receives a reliably tuned and calibrated Metso OC consistency transmitter, and that Metso has comprehensive documentation of the serial number and tuning values of each device dispatched. The Metso OC Tuner software was adopted into regular use in January 2013.</p>	
Language of Thesis	Finnish
Keywords	Consistency, LabVIEW, machine vision
Deposited at	<input checked="" type="checkbox"/> Electronic library Theseus <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences

ALKUSANAT

Ammattikorkeakoulun opetussuunnitelmaan kuuluu opinnäytetyön tekeminen. Sen aiheen miettiminen tuli minulle ajankohtaiseksi keväällä 2012. Ulkomailla suoritetun työharjoittelujakson jälkeen hain kesätöitä Metso Oy:n Kajaanin toimipisteestä. Jo työ haastattelun yhteydessä puhuttiin mahdollisuudesta kesätöiden jälkeen jatkaa yhteistyötä opinnäytetyön merkeissä. Kesän lopulla insinööriykseni valikoitui uuden mittalaitteen testausohjelmiston suunnittelu ja toteuttaminen.

Haluan kiittää insinööriyönaiheesta Metso Oy:tä, joka on tarjonnut hyvät puitteet kokonaisvaltaiselle ohjelmiston suunnitteluprosessille. Haluan kiittää työn ohjaajia, yliopettaja Pentti Romppaista Kajaanin ammattikorkeakoululta ja asiantuntija Matti Törmästä ja Timo Mannista Metso Oy:stä, jotka ovat ohjeistaneet minua työn eri vaiheissa. Haluan kiittää myös suomenkielen opettajaani Eero Soinista Kajaanin ammattikorkeakoululta, äitiäni Mirjami Korhosta sekä veljeäni Ilkka Korhosta kielellisestä ohjauksesta. Suuri kiitos myös muille, jotka ovat kannustaneet minua tämän työn tekemisessä.

Kajaanissa, pääsiäismaanantaina 2013

Heikki Korhonen

SISÄLLYS

1 JOHDANTO	1
2 PAPERIMASSAN VALMISTUSPROSESSI	2
2.1 Puun esikäsittely	2
2.2 Haketus	3
2.3 Hierteen valmistus	4
2.3.1 Hiertoprosessit	5
2.3.2 Hierteen saostus	6
2.4 Hiokkeen valmistus	6
2.4.1 Hiontaprosessit	6
2.4.2 Hiokkeen saostus	8
2.5 Kemiallisen massan valmistus	9
2.6 Uusiomassan valmistus	10
3 PAPERIMASSAN SAKEUDEN MITTAAMINEN	13
3.1 Kuituverkoston muodostuminen ja massan virtausolosuhteet	13
3.2 Sakeuden määritelmä ja standardin mukainen mittaus	14
3.3 Mittausmenetelmiä	14
3.3.1 Lapalähetin	15
3.3.2 Pyörivä in line -lähetin	16
3.3.3 Mikroaaltomittaus	17
3.3.4 Optinen mittaaminen	18
3.3.5 Muut menetelmät	21
4 KONENÄKÖ	23
4.1 Digitaalinen kuva	23
4.2 Digitaalisen kuvan muodostuminen	24
4.3 Kamerakennot	24
4.4 Kameran objektiivi	25
5 OHJELMOINTI	26
5.1 Ohjelmointikielten kehitys	26
5.2 Ohjelmointikielten sukupolvet	28
5.3 LabVIEW-ohjelmisto	29

5.3.1 LabVIEW 2011, graafinen tietovirtaohjelmointi.	30
5.3.2 NI Vision Builder 2011	31
6 METSO OC -SAKEUSMITTARIN TESTAUS	33
6.1 Käytettävä kamera	34
6.2 Optiikka	35
6.3 Testijigi ja muut tarvikkeet	36
7 OHJELMAN SUUNNITTELU JA TOTEUTTAMINEN	37
7.1 Vaatimusmäärittely tulevalle ohjelmistolle	37
7.2 Metso OC Sensorin optisen testerin vaatimusmäärittely	37
7.3 Kameran liittäminen tietokoneeseen ja kuvan saattaminen näytölle	39
7.4 Mittakärjen ja kuitujen paikantaminen	39
7.5 RS-232-yhteyden luominen mittakortille ja ledin virran lukeminen	40
7.6 Raportin ja sertifikaatin tallentaminen ohjelmallisesti	41
8 OHJELMISTON TOIMINTA JA TESTAUS	42
8.1 Käyttäjälle näkyvät toiminnot	42
8.1.1 Device config -välilehti	43
8.1.2 Camera view -välilehti	45
8.1.3 LED-measurement -välilehti	48
8.1.4 Settings-välilehti	51
8.2 Ohjelman toimintojen tarkempi kuvaus ja toiminnalliset lohko-kaaviot	52
8.3 Ohjelman testaaminen	65
9 TYÖN TULOKSET	66
10 YHTEENVETO	67
LÄHTEET	68
LIITTEET	

1 JOHDANTO

Metso Oy on kansainvälinen teknologia- ja palvelutoimittaja prosessiteollisuuden asiakkaille. Sen asiakaskunnan muodostaa kaivos-, maanrakennus-, öljy- ja kaasu-, voimantuotanto- sekä massa- ja paperialat. Liiketoiminta on kuitenkin jaettu kolmeen segmenttiin: kaivos ja maanrakennus, automaatio sekä massa, paperi ja voimantuotanto. [1.]

Yritys syntyi, kun Valmet ja Rauma yhdistyivät elokuussa 1999. Valmet toimi paperi- ja kartonkikonevalmistajana, kun taas Rauman toiminta keskittyi kuituteknologia-, kivenmurskaus- ja virtauksensäätöratkaisuihin. Yhdistymisen seurauksena syntyi maailmanlaajuinen prosessiteollisuutta palveleva laitetoimittaja. [1.]

Samaan aikaan Valmet Automation fuusioitui Neles Controllin kanssa, josta syntyi Neles Automation. Molemmat yritykset olivat kansainvälisesti arvostettuja prosessiautomaation informaatiohallinnan ja kenttäratkaisujen toimittajina ja kehittäjinä. Vuonna 2001 Metso-konsernin automaatio- ja säätöteknologia nimi uudistettiin Metso Automationiksi. Kajaaniin toimipiste on osa Metso Automationia ja siellä keskitytään massa- ja paperiteollisuuden erikoisanalysaattoreihin. [2.]

Yksi Metson uusista sakeusmittareista on Metso OC. Sen mittausmenetelmä perustuu optiseen mittaamiseen. Siinä mitataan valon siroamista paperimassassa ja sen perusteella voidaan laskea massan sakeus todella tarkasti riippumatta normaaleista olosuhdemuutoksista, joita prosessissa voi esiintyä. Mittalaitteessa olevan ledin tuottama valo siirretään prosessiputkeen valokuituja pitkin. Kuitujen läpimitat ovat niin pieniä, ettei niitä voi erottaa silmämääräisesti ilman lisälaitteita saati huomata niissä olevia intensiteettieroja.

Tämän insinööriyön tavoitteena oli suunnitella ja toteuttaa Metso OC -sakeusmittarille testausohjelmisto, jolla voidaan tarkistaa mittalaitteessa käytettävän kuitukaapelin liimauksen onnistuminen ja säätää kaapelissa oleva ledi optimaaliseen valovoimakkuuden arvoon. Ohjelmalla täytyi myös pystyä lukemaan mittalaitteelta sarjamuotoista mittausliikennettä virityksen yhteydessä. Virityksen lopuksi tarkastetusta ja viritetystä laitteesta täytyi tallentaa raportti, josta ilmenee laitteen viritystiedot ja sarjanumerot.

2 PAPERIMASSAN VALMISTUSPROSESSI

Suomen metsäteollisuus on nykyään yksi maailman johtavista puunjalostuksen osaajista. Menestyksen perustana on ollut vahva osaaminen konepajateollisuudessa ja nopea teknologian ja osaamisen kehittäminen. Suomen pinta-alasta yli kaksi kolmasosaa on metsän peitossa ja metsä onkin ollut jo pitkään yksi Suomen kantava vientituote. Ensimmäiset metsäteollisuuden laitokset on perustettu Suomeen jo 1860-luvulla. Jo vuosisadan loppupuolella vienti oli vilkasta. Ensimmäinen suomalaisomisteinen ulkomainen saha ostettiin 1930-luvulla Englannista. Paperiteollisuus oli todella voimakkaassa kasvussa 1980-luvun aikana rakennemuutosten ja kiihtyneen viennin ansiosta. Metsäteollisuus voidaan jakaa kahteen pääosaan, kemialliseen metsäteollisuuteen ja mekaaniseen metsäteollisuuteen. [3, s. 11.]

Vuosituhannen vaihteessa vielä 1980-luvulla toimineista kahdestakymmenestä suurimmasta paperinvalmistajasta oli fuusioiden seurauksena jäljellä enää viisi suurta yritystä. Nykyisin valtaosa suomalaisten paperiyritysten tuotantokapasiteetista tuotetaan Suomen rajojen ulkopuolella. [4.]

2.1 Puun esikäsittely

Kun kuitupuu saapuu tehtaalle, se täytyy ensiksi lyhentää sopivan mittaisiksi pätkiksi. Katkaisun pituus riippuu käytettävästä jatkokäsittelystä. Kaikkien erilaisten massojen valmistuksen edellytyksenä on kuoren irrottaminen puusta. Massatyyppistä riippuen sallittu kuoren osuus vaihtelee 0,2 % - 2 % välillä. Havupuussa kuoren osuus puuaineesta on 12 - 15 % ja koivussa kuoren osuus on noin 15 % puuaineesta. Kuorta on kuitenkin mahdollista hyödyntää irrottamisen jälkeen polttoaineena. Kuori irrotetaan samalla periaatteella puulajikkeesta riippumatta. Puu laitetaan suurina määrinä pyörivään rumpuun, jossa puut hankautuvat sekä toisiaan vasten että rummun seinämiä vasten. Irronnut kuori pääsee tipahtamaan rummussa olevista rei'istä puiden liikkuessa eteenpäin. Kuorinta voidaan suorittaa joko kuivana tai kuorinnan yhteydessä rumpuun voidaan suihkuttaa vettä. Puun kuoriutumistaseseen voidaan vaikuttaa käytetyllä täyttömäärällä, rummun kierrosnopeudella, rummun päädystä olevan portin asennolla ja läpimenoajalla. [3, s. 24.]

Kuorinnan jälkeen puut pestään vedellä. Pesu varmistaa puiden puhtauden ja poistaa vielä mahdolliset kiviainesjäämät puun pinnasta. Vettä käytetään myös talviaikaan jäisten puiden sulatukseen ennen kuin puut laitetaan kuorintarumpuun. Puun kuoresta kuitenkin liukenee uuteaineita huomattavasti enemmän kuin varsinaisesta puuaineesta. Kuoressa on jopa 10 kertaa enemmän typpeä ja fosforia kuin puuaineessa. Aineet kuormittavat vesistöjä suuresti, joten veden jatkokäsittely on välttämätöntä. Kuoriveden annetaan seistä keräilyaltaassa, jolloin hiekka painuu pohjaan, josta se on helppo erottaa. Kuoriveden pH on alle 5, joten se pyritään nostamaan kemikaalien avulla neutraalille tasolle. Vedenkäyttöä on kuitenkin pyritty vähentämään kuorimoissa. [3, s. 30.]

2.2 Haketus

Valmistettaessa sellua puu täytyy pilkkoa niin pieniksi palasiksi, että keittoliuos pääsee reagoimaan puussa olevan ligniinin kanssa kauttaaltaan. Hienontamalla puun rakenne saadaan aikaiseksi keitokseen sellaiset olosuhteet, että puunkuidut erottuvat toisistaan ja saadut kuidut soveltuvat paperin ja kartongin valmistukseen. [3, s. 30.]

Kun puu haketetaan sulfaattiprosessia varten, täytyy puun hakepaksuuden olla 4-7 mm. Keittoliuos imeytyy puuhun osittain tunkeutumalla ja osittain diffuusion avulla. Sulfaattikeitossa imeytyminen tapahtuu pääasiassa diffuusion avulla, joten hakkeen tulee olla tasapaksua. Liuos imeytyy hakkeen kuituihin nopeammin pitkittäissuuntaisten kuitujen kautta kuin poikittaisten. [3, s. 31.]

Hierrettä valmistettaessa selluhake hienonnetaan hienommaksi hakkeeksi. Hierrehakkeen tulee olla kostea, eikä siinä saa olla kuorta eikä purujaetta suurissa määrin. [3, s. 31.]

Haketuksen jälkeen hake seulotaan tarpeiden mukaan. Ylisuuri hake palautetaan murskattavaksi tai uudelleen haketettavaksi. Pienennetty hake lajitellaan uudelleen. Ylisuuri hake on keitossa ongelmallista ja aiheuttaa oksajakeen osuutta, koska se kypsyy keitossa epätasaisesti. Ylipaksu hake voidaan johtaa litistävään mankeliin tai uudelleen haketukseen. Hyväksytyin kokoinen hake johdetaan keittoon, varastointiin tai hiokkeen valmistukseen. Purujae vaikeuttaa suurina pitoisuuksina normaalia keiton valmistamista. Hienontunut puuaines pienentää keiton saantoa, joten se kannattaa keittää omassa astiassa. Purujae on mahdollista kuitenkin

palauttaa hakkeen sekaan hallittuina määrinä. Purujae voidaan myös polttaa kuoren kanssa. Hakkeen seulontaan käytetään tasoseuloja, kiekoseuloja ja rullaseuloja. [3, s. 34-35.]

Valmis hake voidaan johtaa suoraan jatkokäsittelyyn tai se voidaan varastoida tehdasalueelle. Hakkeena puuta voidaan säilyttää tehtaan alueella suurempina määrinä kuin pöleinä. Yleisin tapa hakkeen säilytykseen on ulkovarastointi, jossa hake kasataan isoiksi, 100000-200000 m³ keoiksi. Hakkeena varastoitaessa myös puuaineksen käsittely on edullisempaa, koska hakkeen kuljetus voidaan automatisoida kuljettimilla. Kekovaraston lämpö alkaa kasvaa nopeasti vuodenajasta riippuen 1-2 °C vuorokaudessa. Kasa voi lämmetä kesäaikaan 65-70 °C lämpötilaan ja talvellakin kasan lämpötila voi olla 50 °C. Pitkäaikainen varastointi ja hakkeen seassa oleva hienoaines voi nostaa kasan lämpötilan lähelle syttymispistettä. [3, s. 35.]

2.3 Hierteen valmistus

Hierteen valmistukseen käytetään valmiiksi haketettua puuraaka-ainesta yhdessä tai kahdessa vaiheessa tapahtuvalla jauhatuksella. Kun aikaisemmin haketettu puuaines saapuu jauhattimelle, se syötetään jauhimen terien väliseen tilaan. Hakepalat hienonevat yksittäisiksi kuituiksi kulkiessaan lähekkäin pyörivien terien lävitse. Kuituuntuminen tapahtuu lämmön ja mekaanisten iskujen ansiosta. Hierteen valmistuksessa voidaan käyttää apuna ligniiniä irrottavia kemikaaleja ja lämpöä. [3, s. 57.]

Hierteen raaka-aineeksi soveltuvat parhaiten havupuut, etenkin kuusi. Kemihierre vaatii laadukkaan raaka-aineen, joskin sitä voidaan valmistaa myös lehtipuusta. Kemihierre sopii sellaisenaan paperin raaka-aineeksi, mutta laadukkaampien paperien valmistukseen käytetään peroksiedeilla ja ditioniiteilla valkaistua hierrettä. [3, s. 57.]

1960-luvun alussa hierrettä valmistettiin yksinkertaisella hiertoprosessilla ilman kemikaaleja. Näin saatiin aikaan RMP-massaa. Ensimmäiset paineistetut prosessit otettiin käyttöön vuonna 1968, jolloin saatiin tuotettua TMP-massaa. [3, s. 57.]

Kemihierrettä, CMP-massaa, valmistettiin jo vuonna 1957, mutta hierteen valmistukseen soveltuva CTMP-linja käynnistettiin vasta vuonna 1978. Suomessa ensimmäinen CTMP-linja otettiin käyttöön vuonna 1986 Tampereella Lielahdessa. [3, s. 57.]

Hierteen valmistukseen kuluu energiaa jopa yli 3 MWh/t massatonna kohden, mikä on huomattavasti enemmän kuin muissa massanvalmistusmenetelmissä kuluva energia. Suurimmat jauhimet ovat teholtaan jopa 20 MW. Prosessin energiatehokkuutta on pyritty koventamaan parantamalla lämmön talteenotto- ja kierrätysjärjestelmiä. Suurimmat jauhinteräkierot ovat noin 1800 mm halkaisijaltaan. [3, s. 57.]

2.3.1 Hiertoprosessit

Hierteen valmistamiseen on käytössä kaksi toisistaan hieman poikkeavaa menetelmää.

Kuumahierrettä (TMP) valmistettaessa hake lämmitetään ennen sen hiertämistä. Lämmityksen vaikutuksesta ligniini pehmenee, jolloin kuidut erkanevat toisistaan helpommin ja kuitujen rakenne säilyy parempana. Lämmityksellä voidaan myös tasoittaa hakkeen kosteutta. Haketta voidaan lämmittää joko ei-paineistetussa tai paineistetussa astiassa. Ei-paineistetussa astiassa lämmitys tapahtuu 80-95 °C höyryllä, ja paineistetussa astiassa höyryn lämpötila on 105-120 °C. Paineistetussa astiassa on lämmityksen aikana 0,5-1 bar ylipaine. [3, s. 58.]

Lämmityksen jälkeen hake siirretään ensimmäisen vaiheen jauhimelle, jossa ylipaine on 3-5 bar ja suihkuveden lämpötila on 143-158 °C. Ensimmäisen jauhatusvaiheen jälkeen massasta otetaan lämpö talteen ja massa syötetään seuraavaan vaiheeseen. Toisessa vaiheessa käytetään suunnilleen samaa lämpötilaa ja painetta. [3, s. 58.]

Kemihierrettä (CMP) ja kemikuumahierrettä (CTMP) valmistettaessa hake käsitellään haketta pehmentävillä kemikaaleilla ennen varsinaista jauhatusprosessia. Hakkeen kemikaalinen käsittely vähentää merkittävästi jauhatukseen vaadittavaa energiaa, jopa 30-40 %. Kemikaalien käyttäminen myös mahdollistaa lehtipuiden käytön hierteen valmistuksessa. Kemikaalien avulla voidaan myös valmistaa juuri tietyille paperilaadulle soveltuvaa hierremassaa. Hakkeeseen käytettävä kemikaali on natriumsulfiitti (Na_2SO_3). [3, s. 58.]

2.3.2 Hierteen saostus

Lajittelun jälkeen hierre täytyy saostaa ennen varastointia tai mahdollista valkaisu. Hierteen saostukseen käytetään imu-/painesaostimia tai kiekkoasaostinta. Saostuksen jälkeen hierre voidaan varastoida tai siirtää suoraan paperikoneelle. [3, s. 67.]

2.4 Hiokkeen valmistus

Hiokkeen valmistuksessa puut ladotaan pöleinä kuorinnan jälkeen kiven akselin suuntaisesti koneen hiokeuuniin. Puita painetaan uunissa pyörivää kiveä vasten säädettävissä olevalla hiontapaineella. Puussa olevat pitkittäiset kuidut irtoavat hiontasärmien liikerataa vastaan. Uuniin suihkutetaan lämmintä vettä, jonka lämpötilaa voidaan säädellä. Kuidun irtoamiseen vaaditaan paljon energiaa, joka välittyy pyörivästä hiomakivestä puuhun. [3, s. 38.]

Hiomakivessä olevat kohoumat painavat puussa olevia kuituja kasaan huippujen kohdalla, ja ohitettuaan huipun kuitu pyrkii palaamaan normaaliin tilaan. Tästä aiheutuu nopea värähtelyliike, jonka taajuus voi olla 40-50 kHz. Värähtelyn ansiosta puu lämpenee ja sen pinnan solukerrokset alkavat irrota. Vesi toimii kiven pinnalla voiteluaineena muodostaen voitelevan kerroksen puun ja kiven väliin. Se myös estää puun palamisen ja kuitujen tummumisen. Hiontaprosessissa lämpötila kasvaa ei-paineistetussa hionnassa 110-150 °C:seen ja painehionnassa 130-170 °C:seen. Suurpainehionnassa lämpötila voi olla jopa 180 °C. Korkeasta lämpötilasta johtuen kuituja sitova ligniini pehmenee ja kuidut irtoavat toisistaan helpommin. [3, s. 38.]

2.4.1 Hiontaprosessit

Hiokkeet ja hiontaprosessit voidaan jakaa käytettävän paineen ja suihkuveden lämpötilan mukaan taulukossa 1 esitettyihin pääryhmiin. Painehionnassa käytettävän suihkuveden lämpötila esitetään hiokkeen tyyppin jälkeen tulevalla numeroinnilla.

Taulukko 1. Hiontaprosessit ja hiokkeet. [3, s. 39.]

Hioke	Paine, [bar]	Suihkuvesi, [°C]	Nimityksiä
GW	0	65-75	Groundwood, kivihioke
TGW	0,02-0,03	70-80	Thermogroundwood, "kuumahioke"
PGW	2-3	70-95	Pressure groudwood, painehioke
PGW-S	4-4,5	95-120	Super pressure groudwood, superpainehioke

Kivihioketta (GW) valmistettaessa uuniin syötetään kuorittua puuta ja suihkuvettä. Jotta hiontatulos olisi hyvä, täytyy suihkutettavan veden lämpötilan olla 65-70 °C. Hiokkeen lämpötila kasvaa 10-25 °C korkeammaksi kuin syötettävän suihkuveden lämpötila on. Suihkuveden lämpötila valitaan sopivaksi, jotta prosessissa lämpötila kasvaa riittävän korkeaksi kuivumista ajatellen. Liian kuuma suihkuvesi ei jäädytä hiomakiveä tarpeeksi ja aiheuttaa puun mahdollisen palamisen. Hiomakoneesta saatavan massan sakeus on normaalisti 1-2 %. Massa johdetaan hiokekanaalin avulla karkealajitteluun. Tämän jälkeen massa pumpataan hienolajitteluun ja edelleen sakeutettavaksi. Lajittelussa suodattunut vesi johdetaan takaisin suihkuvedeksi.[3, s. 39.]

Kuumahiokkeella (TGW) tarkoitetaan kivihiontaa, jossa käytettävän suihkuveden lämpötila on normaalia korkeampi. Menetelmä on 1980-luvulla Voithin kehittämä, ja se soveltuu ketjuhiomakoneelle. Se on korvattu myöhemmin tehokkaammilla prosesseilla. Jotta lämpötilaa voidaan nostaa hionnassa, on myös prosessin painetta nostettava. Ylipaineen avulla estetään veden paikallinen höyrystyminen ja puun kuivuminen. TGW-menetelmässä tarvittava ylipaine syntyy hiontakammiossa olevan 20-30 cm korkean vesipatsaan avulla. Hiontakammion ja kiven rajapinnat on valmistettu niin tiiviiksi, ettei vesi pääse valumaan välistä pois. Kammiossa oleva vesi pyrkii kondensoimaan syntyvän vesihöyryn. Hionnasta tulevan massan jatkokäsittely on samanlainen kuin normaalilla GW-hiokkeella. [3, s. 39.]

Painehioke (PGW) on kehitetty 1970- ja 1980-lukujen vaihteessa Tampellan ja ruotsalaisen MoDo:n yhteistyön tuloksena. Ensimmäinen tuotantokone otettiin käyttöön Myllykoski Oy:ssä vuonna 1980. Menetelmä poikkeaa normaalista kivihionnasta korkeahkon paineen ja kuumemman suihkuveden takia. Ylipaine on enimmillään 3 bar ja suihkuveden lämpötila on

normaalisti 90-95 °C. Korkean lämpötilan takia massan lämpötila nousee välille 105-120 °C. Hiomakoneesta saatava noin 100 °C oleva hioke johdetaan painesakeuttimeen, jossa sakeudeltaan noin 1,5 % massa saostetaan yli 10 % kuiva-ainepitoisuuteen. Saostuksessa erotuva kuuma noin 95-100 °C oleva vesi johdetaan takaisin suihkuvesikiertoon. Menetelmän ansiosta suihkuveden lämmitykseen ei tarvita paljoa energiaa. Saostettu massa laimennetaan kiertovedellä 3-4 % sakeuteen, jolloin massa jäähtyy lähelle kiertoveden lämpötilaa ja se voidaan lajitella. [3, s. 39-40.]

Ensimmäinen ns. superpainehiomakone (PGW-S) otettiin käyttöön niin ikään Myllykoski Oy:llä vuonna 1989. Superpainehiomakoneessa paine voi olla jopa 5 bar ja suihkuveden lämpötila on 120 °C:sta. Prosessi on pääpiirteittäin samanlainen kuin tavallinen painehiontaprosessi, mutta siihen lisätään lämmönsiirtojärjestelmä, jos halutaan toimia yli 100 °C lämpötiloilla. Hiontakoneesta poistuva massa johdetaan lämmönsiirtojärjestelmään, jossa se jäähdytetään 100 °C:seen. Jäähdytyksen lämpö otetaan talteen ja sitä hyödynnetään lämmitettäessä suihkuvesi 120 °C. Jäähdytetty massa johdetaan edelleen matalapainesaostimelle, jossa se käsitellään kuten painehiontaprosessin massa. [3, s. 40-41.]

2.4.2 Hiokkeen saostus

Kun massa on lajiteltu, se täytyy saostaa. Lajittelusta tulevan massan sakeus on hyvin alhainen, noin 0,6-0,8 %. Massa on saostettava ennen valkaisua, ja saostettuna se on myös helpompi varastoida. Saostuslaitteiden toimintaperiaate on kaikilla laitteilla sama. Suodattimena toimii usein tiukka suodatinkangas tai muoviviira, jonka läpi vesi pääsee kulkemaan massan jäädessä suodattimen pinnalle. Suodatus perustuu paine-eroon suodattimen pintojen välillä. Vesi pyrkii tunkeutumaan matalamman paineen puolelle. Laitte voi olla joko paine- tai imusaostin. Saostimella saavutetaan 4-12 % kuiva-ainepitoisuus. Saostuksessa saatava vesi kerätään talteen ja kuljetetaan takaisin prosessin eri vaiheisiin, joissa tarvitaan massan laimennusta. Saostettu hioke varastoidaan säiliöihin, joihin mahtuu muutaman tunnin tai päivän hioketarve. Varastointisakeudella on suuri merkitys tilantarpeeseen. [3, s. 53.]

2.5 Kemiällisen massan valmistus

Kemiällisen massan valmistukseen on kaksi päämenetelmää: sulfiitti- ja sulfaattikeitto. Molemmat menetelmät ovat suhteellisen vanhoja, jo 1800-luvulla kehitettyjä menetelmiä, jotka eroavat toisistaan käytetyn keittokemikaalin perusteella. Sulfiittimassan lujuusominaisuudet ovat huomattavasti heikommat kuin sulfaattimassalla. Se onkin Suomessa syrjäyttänyt sulfiittimassan myös pienempien rikkipäästöjen ansiosta. Kuitenkin muualla maailmassa on vielä laajalti käytössä sulfiittitehtaita. [5, s. 31.]

Kemiällisen massan valmistus perustuu kemialliseen kuidutukseen, jossa lämmön ja kemikaalien avulla kuituja toisiinsa sitova ligniini liukenee ja kuidut irtaantuvat toisistaan. Kemikaalien avulla on mahdollista poistaa pelkästään ligniini, vaan prosessissa liukenee myös muita komponentteja, eritoten hemiselluloosaa. Koska keitoksessa ei voida liuottaa pelkästään ligniiniä, on kemiällisen keiton saanto suhteellisen heikko, tyypillisesti 50 %, eli yhtä massatonna kohden tarvitaan kaksi tonnia eli 5-6 m² puuta. Keittämällä saatu massa on kuitenkin varsin puhdasta selluloosaa. [5, s. 31.]

Eräkeittimet ja jatkuvatoimiset keittimet muodostavat kaksi keittolaitteistojen päätyyppiä. Keitetystä massasta erotetaan keittoliemi, jonka jälkeen massa lajitellaan ja tarvittaessa valkaistaan. Eroteltu keittoliuos väkevöitetään haihduttamalla, jonka jälkeen se poltetaan. Polttoprosessista kerätään talteen energia joko höyryksi tai sähköksi. Polton yhteydessä vapautuu kemikaaleja, jotka kerätään talteen ja niitä käytetään uuden keittoliuoksen valmistukseen. Keittoprosessin jälkeen valkaisematon sulfaatti sellu on väriltään ruskeaa. [5, s. 32.]

Keittoprosessissa massaan on vielä jäänyt jonkin verran ligniiniä. Massan ligniinipitoisuuden mittana käytetään kappalukua. Massan valkaisussa poistetaan jäljelle jäänyt ligniini massasta ja valkaisun tarvittava voimakkuus riippuu jäännösligniinipitoisuudesta. Aikaisemmin valkaisuun on käytetty kloorikaasua, mutta nykyään se on korvattu happivalkaisulla tai otsonivaiheella. [5, s. 32.]

Pitkäkuituisesta havupuusulfaattimassasta valmistetaan tuotteita, joiden täytyy olla lujia. Massan lujuusominaisuudet heikkenevät hieman valkaisuprosessissa. Tämän vuoksi valkaisemattomaa havupuusulfaattia käytetään pääsääntöisesti aaltopahvin pinnan eli lainerin ja voimapaperin valmistukseen. Valkaistua havupuusulfaattimassaa käytetään pääsääntöisesti kirjoitus- ja

painopaperin lujuutta antavana ns. armeerausmassana. Valkaistua massaa käytetään myös parempilaatuisten kartonkien pinta- ja taustakerroksina sekä pehmopapereissa. [5, s. 32.]

Pidemmät puukuidut muodostavat kestävämmän verkoston kuin lyhyet. Lehtipuista saatavat kuidut ovat ohuempia kuin havupuun kuidut, minkä vuoksi lehtipuusta saadaan moninkertainen määrä kuituja puutonnia kohden. Lehtipuusta saadut kuidut parantavat paperin optisia ominaisuuksia ja siitä syystä lehtipuusta saadun sellun tärkein käyttökohde on hienopaperi ja valkoiset kartonkilajit. [5, s. 31.]

2.6 Uusiomassan valmistus

Keräyspaperin merkitys on paperiteollisuuden raaka-aineena maailmanlaajuisesti jo merkittävä ja sen osuus on edelleen kasvussa. Suurin osa käytetystä paperista ja kartongista kerätään talteen uudelleen käyttöä varten. Vaikka Suomessa keräysaste on korkea, kuitenkin suurin osa Suomessa tuotetusta paperista ja kartongista menee vientiin, joten kiertokuidun osuus paperinvalmistuksessa on kohtalaisen pieni. [5, s. 35.]

Siistauksessa hajotetusta ja lajitellusta keräyspaperista poistetaan täyteaineet, väriaineet ja muut paperin valmistuksessa käytetyt lisäaineet. Kotikeräyspaperi painetaan offset-painatuksella ja tässä käytettyjen värien lisäaineet tarttuvat kuituihin kiinni tiukasti vanhetessaan ja hankaloittavat siistausprosessia. Päällystetyn paperin painoväri on tarttunut päällystyspigmenttiin ja irtoaa siksi helpommin. Kuitenkin suuret pigmenttipitoisuudet haittaavat vaahdotusprosessin hallintaa ja ne rikastavat kiertovettä. Siistaus voidaan tehdä joko pesusiistauksena tai vaahdottamalla. Suomessa käytetään vaahdotusta. [3, s. 68.]

Siistausprosessi täytyy suorittaa useassa vaiheessa, ja siihen kuuluu useiden eri kemikaalien käyttö. Yleensä siistaamo suunnitellaan käsittelemään vain tietynlaista keräyspaperia ja tehdään yhteyteen sijoitettuna siitä saatavan massan täytyy soveltua tehtaan tarpeisiin. Siistausprosessi koostuu seuraavista vaiheista: [3, s. 69.]

Pulperointi on jatkuva prosessi, jossa pulperointiin käytetään yleensä kuorimarumpua muistuttavaa rumpua, jonka halkaisija on 3,5 m ja rumpun pituus on 25 m. Kuidutusrumpu pyörii 13 kierrosta minuutissa ja massan sakeus on alkupäässä rumpua 13-15 %. Massa kulkee rummussa eteenpäin ja rumpun loppupää on rei'itetty ja sinne johdetaan laimennusvettä.

Massan sakeus on loppupäässä 4-5 %. Kuituuntunut paperimassa menee reikien läpi ja muu massa poistuu rummun avonaisesta päästä. [3, s. 69.]

Esilajittelussa massasta poistetaan kaikki kiinteä jäte, joka on mahtunut kulkemaan rummussa olevista 6 mm rei'istä. Lajittelussa poistetaan karkea sora, niitit, rautalangan pätkät ja muut mahdollisesti keräyspaperissa olleet tavarat. [3, s. 69.]

Vaahdotusprosessi suoritetaan kennoissa, joissa laimeaan massasulppuun lisätään vaahdotuskemikaaliksi saippuaa. Seokseen puhalletaan ilmaa, jolloin saippua muodostaa kuplia. Painoväri tarttuu muodostuneisiin kupliin ja nousee pintaan, josta ne poistetaan ylijuoksuna. Vaahdotus suoritetaan kahdessa vaiheessa. Toisessa vaiheessa ensimmäisen vaahdotuksen ylijuoksutusvaahto vaahdotetaan uudelleen. Poistettu vaahto saostetaan 40 % sakeuteen ja hävitetään. [3, s. 69.]

Jälkilajittelu muodostuu kolmesta lajittelujärjestelmästä. Ensimmäinen lajittelu suoritetaan pyörrepuristimella, jolla voidaan erottaa massasta hiekka ja muut raskaat jakeet, jotka eivät nousseet vaahdotuksessa. Toisessa vaiheessa massasta erotetaan pienet muovipallot, styrok-
sit, teipit ja paperissa käytettyjen liimojen tuomat epäpuhtaudet. Kolmantena vaiheena käytetään käänteistä pyörrepuhdistusta. Se erottelee massasta kuituja kevyemmät jakeet. [3, s. 69.]

Sakeutettaessa massasta poistetaan aluksi vesi kiekkosuostimella ja lopuksi puristamalla massaa ruuvi- tai kaksoisviirapuristimella. Kiekkosuostimella eroteltua vettä käytetään pulperoinnissa laimennusvetenä prosessin alkuvaiheessa. Puristuksessa eroteltu vesi puhdistetaan. Siistaamosta pyritään kuljettamaan mahdollisimman vähän vettä paperitehtaan veden kiertoon, koska siistaamon vedessä olevat vieraat aineet hankaloittavat paperikoneen ajoa. Sakeutusvaiheessa massan sakeus nostetaan noin 30 %:iin. [3, s. 69.]

Muokkausvaiheessa saostettu massa muokataan levyjauhimissa, missä painovärijäänteet irtoavat kuitujen pinnasta ja hienontuvat pieniksi, näkymättömiksi hipuiksi. Tässä vaiheessa massaan voidaan lisätä pihkatalkkia jäänteiden jälleensaostumista estämään. [3, s. 69.]

Jälkivaahdotus on samanlainen kuin aikaisempikin vaahdotusprosessi. Jälkivaahdotuksessa poistetaan loput epäpuhtaudet ja kemikaalijäänteet massasta. Vaahdotuksessa ei tarvita enää saippuaa. [3, s. 69.]

Lopullinen sakeutus tehdään kiekkosaostimella ja massa varastoidaan 10 % kuiva-ainepitoisuudessa. Massa varastoidaan varastotorniin, josta se voidaan pumpata myöhemmin paperikoneelle. [3, s. 70.]

Vedenpuhdistus suoritetaan kun siistaamolta poistuva vesi puhdistetaan kemiallisella flokkauksella ja flotaatioselkeytyksellä. Kirkastunut vesi johdetaan normaaliin jäteveden käsittelyyn. [3, s. 70.]

Rejktiä muodostuu siistaamalla vaahdotuksesta, pulperoinnista, lajittelusta ja veden käsittelystä. Suurin osa muodostuneesta siistausjätteestä tulee vaahdotuksesta ja veden käsittelystä. Kuivattu jäte sisältää noin 50 % lyhyitä kuituja, 40 % täyteaineita ja 10 % painoväriaineita, sideaineita ja siistauskemikaalien jätteitä. Siistausjäte lingotaan noin 40 % kuiva-ainepitoisuuteen ja kuljetetaan kaatopaikalle. Lajittelu rejktiosta poistetaan vesi ja jäte kuljetetaan kaatopaikalle. [3, s. 69.]

3 PAPERIMASSAN SAKEUDEN MITTAAMINEN

Yksi paperi- ja selluteollisuuden haastavimmista ja kriittisimmistä säädöistä on massan sakeuden säätäminen. Sakeuden mittatarkkuutta huonontavat anturin likaantuminen, massan virtausnopeuden muutokset, näytteenotossa esiintyvät hankaluudet sekä anturin mekaaninen kuluminen. Säätämistä haittaa mittaus- ja ohjauskohdan välillä oleva kuollut aika ja sekoituskohtaan tulevan massan virtausmuutokset. Säätämisen kannalta massan sakeutta on mitattava jatkuvasti, vaikka näin toimimalla päädytään epäsuoriin mittausmenetelmiin. Sakeuden mittaamiseen on käytetty erilaisia menetelmiä jo pitkään, mutta ehdottoman hyvää menetelmää ei ole toistaiseksi löydetty. Eräs syy tähän vaikeuteen on se, että sakeuden määritelmä ei kaikissa suhteissa kuvaa hyvin vaikutuksia paperinpainatusominaisuuksiin. [6, s. 233.]

3.1 Kuituverkoston muodostuminen ja massan virtausolosuhteet

Kun harkitaan sakeusmittarin asentamista, pitää virtausolosuhteet asennuspaikassa tuntea hyvin. Virtausolosuhteet vaikuttavat erityisesti mekaanisiin antureihin. Huomioitavia asioita asennuspaikassa on virtausnopeus, virtauskanavan muoto ja sen koko, virtausta muuttavien venttiilien ja muiden liittimien sijainti, lämpötila, kuitu-vesisuspension tyyppi ja eritoten käytettävä sakeusalue. [6, s. 233.]

Puhtaaseen veteen lisätyt kuidut muuttavat sen virtausprofiilia selvästi. Keskisakeuksinen hitaasti liikkuva massa virtaa prosessiputkessa tulppamaisena. Tulppamaisessa virtauksessa kuidut ovat muodostaneet yhtenäisen verkoston, joka liikkuu putkessa. Kun virtausnopeutta kasvatetaan, syntyy sekavirtaus. Tulpan ja putken seinämän väliin muodostuu ohut vesikerros ja kuituliete on tulpan reunalla turbulentsissa tilassa. Nopeuden kasvaessa edelleen häviää tulppa ja virtaus muuttuu kauttaaltaan turbulentsiksi. Virtaustila vaikuttaa myös massan flokkaantumiseen ja epähomogeenisuuteen. [6, s. 233.]

3.2 Sakeuden määritelmä ja standardin mukainen mittaus

Sakeudella tarkoitetaan useassa yhteydessä samaa kuin näennäisellä viskositeetilla eli konsistenssilla. Teollisuudessa sakeudella tarkoitetaan nesteen kuiva-ainepitoisuutta. Sakeutta ei voi verrata suoraan näennäiseen viskositeettiin, vaan niiden suhde muuttuu erityyppisten aineiden välillä. [7, s. 123.]

Sakeus määritellään yhtälöllä

$$Cs = 100 * \frac{F}{W} \% \quad (1)$$

missä

Cs on sakeus prosentteina

F on kuivamateriaalin paino

W on massasulpun kokonaispaino

Scan-test -menetelmää käyttäen sakeus määritellään siten, että tunnettu tilavuusmäärä massasulppua suodatetaan suodatinpaperin lävitse ja suodos papereineen kuivatetaan uunissa. Suodatinpaperin paino täytyy määrittää ennen mittauksia. Tätä menetelmää käytetään pääsääntöisesti muiden sakeusmittausten tarkastukseen ja laitteiden kalibrointiin. [6, s. 234.]

3.3 Mittausmenetelmiä

Sakeuden mittaaminen voi perustua massan leikkausvoimaan, ultraäänen vaimenemiseen, mikroaaltojen etenemisominaisuuksiin, valon absorptioon, valon sirontaan tai depolarisaatioon, tiheyteen tai virtauskitkaan. Kaikki jatkuvatoimiset mittausmenetelmät ovat approksimoivia. Mittauksissa on erilaisia vaikutusmekanismeja sakeuden lisäksi muistakin suureista. Sakeusmittarin valintaan vaikuttaa sakeuden mittausalue, massan laatu, onko massa kemiallista, mekaanista vai siistattua, täyteaineet ja niiden määrien vaihtelut, mustalipeän määrä ja sen vaihtelu, väriaineiden määrä ja vaihtelu, lämpötila-alue, virtausnopeus ja sen vaihtelu, jauhasasteen vaihtelu sekä mittauksen tarkkuusvaatimus. [6, s. 234-235.]

3.3.1 Lapalähetin

Sakeuden ollessa yli 2 % pysyvät massan virtausominaisuudet lähes vakiona muodostuneen kuituverkon ansiosta. Tällaisiin tilanteisiin mekaaninen mittaus soveltuu erinomaisesti. Kun putkessa liikkuva kuituverkostoa leikataan jollakin terällä, voidaan leikkausvoima havaita jo pienelläkin leikkausnopeudella. Tarvittava leikkausvoima on lähes riippumaton massan virtausnopeudesta, mutta massan sakeus vaikuttaa herkästi voimaan. Lapalähettimen lähtövies-
tinä käytetäänkin leikkausvoimaa. Sakeuden vaikutus lapalähettimen havaitsemaan lähtöön on neliöllinen. [6, s. 235.] Kuvassa 1 on Metso Oy:n SP-lapalähetin.



Kuva 1. Metso SP -lapalähetin [8].

Lähettimen mittausalue on yleensä 1,5-8 %. Massan virtausnopeuden vaihtelu voi olla välillä 0,5...3 m/s. Lämpötilalla on vaikutus massan viskositeettiin matalilla sakeuksilla. Sakeuden kasvaessa lämpötilan vaikutus pienenee. Lähettimen asennuspaikaksi täytyy valita kohta, jossa virtaus on laminaarinen ja lähetin on riittävän kaukana pumpuista, mutkista ja venttiileistä. Prosessiputken halkaisijan täytyy olla vähintään 100 mm. Lapalähettimen asennus on helppoa, eikä lähetin tarvitse jatkuvaa huoltoa. Lapalähetin on myös hankintahinnaltaan suhteellisen edullinen. [6, s. 235.]

3.3.2 Pyörivä in line -lähetin

Pyörivän sakeuslähettimen mittaus perustuu myös leikkausvoiman mittaamiseen. Laitteessa on tunnustelija ja potkuri, jotka pyörivät prosessiputken sisällä massan seassa. Sähkömoottori pyörittää akselijärjestelmää kartiovaihteen välityksellä. Mittausakseli, jonka päässä tunnustelija on, on kytketty vetoakseliin elastisesti. Kun tunnustelija pyöriessään leikkaa massaa, leikkausvoima aiheuttaa mittausakseliin momentin. Väjäntömomentti saa aikaan mittausakselin ja vetoakselin väliin kulmaeron. Syntynyt kulmaero mitataan ja säädetään nolllaksi magneettikentän avulla. Magnetoivien kelojen virtaa mitataan, ja niiden vaatimaa virtaa käytetään mittarin lähtöviestinä. [6, s. 235.] Kuvassa 2 on Metso Oy:n Rotary-sakeusmittari, joka on pyörivä sakeuslähetin.



Kuva 2. Metso Rotary -pyörivä sakeuslähetin [9].

Sakeuden mittausalue on yleensä 1...10 %. Massan virtausnopeuden vaihtelu voi olla välillä 0,5...5 m/s. Pyörivä anturi voidaan asentaa välittömästi sekoituspumpun jälkeen, jolloin saadaan pienennettyä säätöä haittaavaa matkaviivettä. Monimutkaisen rakenteensa vuoksi laite on herkempi mekaanisille häiriöille, kuin lapalähetin. Pyörivä sakeusmittari vaatii myös enemmän huoltoa ja sen asentaminen on hieman hankalampaa kuin lapamallin lähettimen. [6, s. 236.]

3.3.3 Mikroaaltomittaus

Sakeusmittauksissa käytettyjen mikroaaltolähettimien toiminta voi perustua mm. korkeataajuisen radioaallon vaimenemiseen, heijastuksen muutokseen, vaiheen muutokseen, kantotaajuuden muutokseen tai kulkuajan- tai aallonpituuden muutokseen. Tässä luvussa perehdytään sakeuslähettimeen, jonka toiminta perustuu mikroaaltosignaalien kulkuajan mittaamiseen. [6, s. 236.]. Kuvassa 3 on Metso Oy:n MCA-sakeusmittari, joka perustuu mikroaaltomittaukseen.



Kuva 3. Metso MCA-FT ja TCU [10].

Lähetin lähettää n. 3 GHz taajuudella olevaa mikroaaltosignaalia massan lävitse vastaanottimelle. Kun signaali kulkee massa-vesisuspension läpi, signaalin kulku hidastuu. Vertaamalla viivästynyttä signaalia alkuperäiseen referenssisignaaliin voidaan määrittää signaalin kulku-aika t . Kulku-aika riippuu lähettimen ja vastaanottimen välisestä matkasta sekä massan sähköisistä ominaisuuksista, joita kuvaa dielektrisyysvakio ϵ . Suhteellinen dielektrisyysvakio on vedellä noin 80, kun vastaava arvo puuainekselle lajikkeesta riippumatta on noin 3. Massaan lisättävien erilaisten lisäaineiden dielektrisyysvakio on tältä väliltä, mutta niiden vaikutus mittaukseen pyritään kompensoimaan syöttämällä laitteelle laboratorion saadut kompensointiarvot. Massan dielektrisyysvakio on lineaarisessa suhteessa massan sakeuden kanssa.

Massan keskimääräinen dielektrisyysvakio määritellään yhtälöllä

$$v = \frac{c}{\sqrt{\epsilon_r}} = \frac{l}{t} \rightarrow \epsilon_r = \frac{c^2 * t^2}{l^2} \quad (2)$$

missä

v on nopeus väliaineessa

t on signaalin kulkuaika

ϵ_r on suhteellinen dielektrisyysvakio

c on valon nopeus tyhjiössä

l on signaalin kulkema matka

Mikroaaltotekniikkaan perustuvalla sakeusmittarilla päästään 0,1 % mittausepävarmuuteen ja 0,01 % toistettavuuteen Cs-arvoina mitattuna. Mikroaaltomittarin teoreettinen mittausalue on 0...75 %. Kuitenkin suurilla sakeuksilla massasuspension johtokyky kasvaa suureksi ja mikroaaltoihin alkaa syntyä heijastuksia, jotka aiheuttavat häiriöitä. Yleinen mittausalue on 1,5...40 %. [6, s. 238.]

Mikroaaltotekniikkaan perustuva mittalaite on tarkoitettu prosessiteollisuudessa sovelluksiin, joissa tarvitaan tarkkaa kokonaissakeuden mittaamista. Mittaustekniikka soveltuu myös erinomaisesti säätöjärjestelmiin, joissa ongelmia ovat vaihtelevat prosessiolosuhteet, kuten virtausnopeuden muutokset, massalajin muutokset tai kuitutyyppien muutokset. Mikroaaltoon perustuva mittari soveltuu noin 15 %:iin paperi- ja selluprosessien sakeusmittauksista, kun otetaan huomioon laitteen suorituskyky ja hinta. Laitteen hinta on noin kymmenkertainen verrattuna mekaaniseen sakeuslähettimeen. [6, s. 238.]

3.3.4 Optinen mittaaminen

Optiset mittausmenetelmät soveltuvat erittäin hyvin matalasakeusmittauksiin, joissa mitattavat sakeusalueet vaihtelevat 0...2 % välillä. Paperikoneen viirakaivon ja perälaatikon sakeusalue on tyypillisesti 0,1...1 % ja niiden mittaus on merkittävin sovellusalue matalasakeusmittauksissa. [6, s. 247.]

Massan optisiin ominaisuuksiin vaikuttavat puukuidut, vesi ja lisäaineet. Prosessiputkessa kulkevassa massassa olevat puukuidut vaimentavat valoa voimakkaammin kuin vesi. Puukuidut muodostavat suspensioon optisia rajapintoja, joista valo heijastuu osittain takaisin tai sirotaan satunnaisesti. Valon intensiteetti siis vaimenee, kun valo kulkee kuitususpensiossa absorptio- ja sironnan vaikutuksesta. [6, s. 248.]

Kuitulajin jauhatustapa ja valmiusaste vaikuttavat suspension optisiin ominaisuuksiin, jolloin massan vaimennusolosuhteet muuttuvat. Tämä asia täytyy ottaa huomioon mittauksissa tai käyttää samaa kuitulajia, jolloin valon kokonaisintensiteetin heikkeneminen on suoraan verrannollinen massan sakeuteen. Puukuidut ovat myös optisesti aktiivisia, jolloin ne kiertävät lineaarisesti polarisoidun valon tasopolarisaatiota. Puukuidun optinen aktiivisuus liittyy sen epäsymmetriseen molekyyliarakenteeseen. [6, s. 248.]

Retentiolla voidaan tarkoittaa yleisesti hyötysuhdetta. Paperinvalmistuksessa se tarkoittaa sitä osaa syötettävästä raaka-ainevirrasta, joka jää valmistettavaan lopputuotteeseen. Retentio-käsite voidaan määritellä sen mukaan, miten prosessi rajataan ja mikä raaka-ainekomponentti on tarkasteltavana. Koko paperintekoprosessin hyötysuhdetta kuvaa kokonaisretentio. Viiraretentiolla tarkoitetaan viiralle jäävää osuutta perälaatikosta purkautuvan kokonaiskiintoainevirrasta. Viiraretentio lasketaan yhtälöstä,

$$Retentio = \frac{Perälaatikon\ massavirta - Viiran\ läpaisevä\ massavirta}{Perälaatikon\ massavirta} * 100\% \quad (3)$$

Yhtälö voidaan muuttaa matemaattiseen muotoon,

$$R = \frac{C_{hb}Q_{hb} - C_{ww}Q_{ww}}{C_{hb}Q_{hb}} * 100\% \quad (4)$$

missä

R on retentio

C_{ww} on viiraveden sakeus

C_{hb} on perälaatikon sakeus

Q_{ww} on tilavuusvirta viirakaivoon

Q_{hb} on tilavuusvirta perälaatikosta.

Kun otetaan huomioon, että huulisuihkun ja viiraveden virtaukset ovat lähes yhtä suuret, saadaan yhtälö muutettua retentiolle yleisemmin käytettyyn muotoon,

$$R = \left(1 - \frac{C_{ww}}{C_{hb}}\right) * 100\% \quad (5).$$

Kiintoaineretentio saadaan laskemalla kaksi eri sakeusmittausta yhtälöstä (5). Retentiomittauksiin tarvitaan siis matalasakeusmittareita. [6, s. 244-245.]

Ensimmäisen sukupolven matalasakeusmittarit perustuivat useimmiten vain yhden optisen ominaisuuden mittaamiseen. Mittaustavat olivat depolarisaatio, valon vaimeneminen tai valon sironta. Näillä mittareilla kyettiin mittaamaan vain massan kokonaissakeutta. Niiden heikkoutena oli herkkyys massalajien muutoksille, täyteaineiden määrän vaihtelulle, lämpötilan vaihtelulle ja retentio- sekä väriaineiden vaihtelulle. Esimerkkeinä ensimmäisen sukupolven laitteista voidaan mainita Kajaani Elektroniiikan LC-100 (kuva 4), joka perustuu depolarisaatioon, Eur-Controlin Opticon, jossa mitataan sirontaa ja Valmetin Cons-EI, joka perustuu absorptioon. [6, s. 248-249.]



Kuva 4. Kajaani Elektroniiikan LC-100 matalasakeusmittari [11].

Toisen sukupolven matalasakeusmittaukset mittaavat samaan aikaan useita optisia ominaisuuksia massasta. Mittaamalla valon vaimeneminen, depolarisaatio ja vaimennus yhdellä tai useammalla aallonpituudella tuloksista voidaan laskea aikaisemmin mainituista tekijöistä

riippumatta anturin sakeusviesti. Mittarit pystyvät myös erottamaan kokonais- ja täyteenesakeudet toisistaan. Esimerkkeinä toisen sukupolven mittalaitteista voidaan mainita Kajaani Elektroniiikan RM-200C ja RM-200M (kuva 5) ja Eka Nobel Chemtronics 400. Monet nykyaikaiset mittaukset perustuvat RM-200-mallin periaatteeseen, joten sen toimintaperiaatetta tarkastellaan seuraavaksi tarkemmin. [6, s. 249.]



Kuva 5. Kajaani Elektroniiikan RM-200 -matalasakeusmittari [13].

Mittausperiaatteena on, että puolijohdelaserin tuottama polarisoitu valonsäde johdetaan mittakennossa virtaavan massan läpi. Massan lävitse päässyt valo kootaan linssillä ja suunnataan erityiselle aukkolevyille. Sen läpäissyt valo kootaan toisen linssin avulla taas valokimpuksi, joka johdetaan polarisoivalle prismalle. Prisma jakaa valon risti- ja yhdensuuntaispolarisoiduiksi komponenteiksi. Tällaisella järjestelmällä saadaan mitattua näytteen läpäisseen valon vaimennusta ja depolarisaatiota. Valon sirontaominaisuuksia saadaan mitattua vertaamalla valon intensiteetin vaimennusta eri aukoilla. Mittaussignaaleista lasketaan matemaattisten mallien perusteella kokonais- ja tuhkasakeuspitoisuudet massasta. Mittauksessa ei voida mitata koko prosessiputken massaa, vaan prosessiputkesta järjestelmään tulee erillinen näytelinja, jota laite analysoi. [6, s. 250.]

3.3.5 Muut menetelmät

Aikaisemmin mainittujen periaatteiden lisäksi on olemassa muita menetelmiä, joita käydään lyhyesti läpi tässä luvussa.

Ultraäänen absorboitumista massaan voidaan mitata antamalla lähettimen lähettämän ultraäänisignaalin kulkea massan läpi prosessiputken seinämään. Ääniaalto heijastuu seinämästä takaisin lähettimen suuntaan. Mittaamalla heijastuvan signaalin voimakkuutta saadaan signaalien suhde selville, ja sen perusteella on mahdollista mitata massan sakeutta. Lähetetty signaali vaimenee massassa. [7, s. 124.]

Radioaktiivinen mittaus perustuu gammasäteilyn vaimenemiseen massassa. Menetelmä pohjautuu Comptonin sirontaan gammasäteillä, jotka ovat peräisin cesiumin tai koboltin radioisotoopeista. Menetelmän etuna on helppo asennus, koska antureiden ei tarvitse olla kosketuksessa massaan, vaan laite voidaan asentaa prosessiputken ympärille. Radioaktiivisten antureiden käyttö on luvanvaraista. [11, s. 16.]

Virtauskitkamittauksessa massan sakeutta mitataan sileässä virtausputkessa liikkuvan massasulpun luomien paine-erojen avulla. Edellytyksenä mittaukselle tarvitaan riittävä virtausnopeus ja riittävän suuri putken halkaisija. [7, s. 125.]

4 KONENÄKÖ

Konenäön tärkein tavoite suurimmalle osalle sen sovelluskohteista on löytää kuvista oleellisia piirteitä, joiden tulkinta tai ymmärtämys voidaan suorittaa tietokoneella. Konenäköjärjestelmä voi erotella esimerkiksi osia kokoonpanolinjalla ja määrittää niiden ominaisuudet visuaalisella tarkastamisella. Konenäköjärjestelmä voi kuvan muodostamisen ja esittämisen lisäksi myös analysoida kuvaa ja ymmärtää kuvan merkityksen. [14, s.14.]

4.1 Digitaalinen kuva

Ensimmäiset digitaaliset kuvat ovat peräisin sanomalehtiteollisuudesta jo 1920-luvulta. Vuonna 1920 kuvia alettiin lähettää Lontoon ja New Yorkin välillä merikaapelia pitkin laivakuljetuksen sijaan. Kuvien lähettäminen vaati lähetettävän kuvan koodauksen, ja vastaanotavassa päässä kuvan koodaus täytyi purkaa. Ensimmäisten mannerten välisten kuvien taso oli kovin vaatimaton, mutta vuotta myöhemmin tekniikka oli kehittynyt sellaiselle tasolle, että oli mahdollista lähettää kuvia, jotka sisälsivät viisi harmaan sävyä. Vuosikymmenen lopulla kuvat sisälsivät 15 harmaan sävyä. [15.]

Kuitenkaan tuohon aikaan lähetettyä kuvaa tai kuvan käsittelytekniikkaa ei voida pitää digitaalisena kuvan käsittelynä. Digitaalisen kuvan käsittely on määritelmän mukaan kuvan numerista käsittelyä. Tästä johtuen digitaalinen kuvankäsittely on alkanut vasta tietokoneiden kehityksen jälkeen 1940-luvulla. [15.]

1960-luvun alussa tietokoneiden tiedonkäsittelykapasiteetti oli vasta mielekkäällä tasolla kuvankäsittelyn kannalta. Samoihin aikoihin Yhdysvalloissa suunniteltiin miehitettyä kuulentoa, ja ohjelman alkuvaiheessa miehittämättömistä kuulennoista saatuja vääristyneitä kuvia täytyi korjata digitaalisesti. Avaruusohjelmalla ja sotateollisuudella on ollut suuri vaikutus digitaalisen kuvankäsittelyn kehitykseen. [15.]

Avaruusohjelman lisäksi digitaalista kuvankäsittelyä käytettiin tähtitieteeseen, satelliittikartoitukseen ja lääketieteeseen jo 1960- ja 1970-luvulla. Erityisen merkittävä sovellus oli lääketieteen kannalta digitaalisen kuvauksen mahdollistama kerroskuvaus. Sen avulla oli mahdollista saada tietoa kohteista, joita ei voinut kuvata perinteisellä röntgen-kuvauksella. Myöhemmassä

vaiheessa kerroskuvausmenetelmänä on alettu soveltaa MRI-tekniikkaa eli magneettikuvausta. [15.]

1990-luvulla digitaalisen kuvankäsittelyn soveltaminen on kasvanut todella voimakkaasti. Suurin syy huimaan kehitykseen on tietokoneiden jatkuva kehitys. Samaan aikaan kuvankäsittelyyn vaadittavan laitteiston hinta on tullut alas. Myös kuvankäsittelyn kannalta oleelliset ohjelmat ovat kehittyneet voimakkaasti. [15.]

4.2 Digitaalisen kuvan muodostuminen

Digitaalinen kuva koostuu kuva-alkioista eli pikseleistä. Yleisin digitaalisen kuvan tunniste on sen resoluutio, joka ilmaisee pikseleiden määrän vaakasuorassa ja pystysuorassa suunnassa tai kuvan pikseleiden kokonaislukumäärän. Jokaisella pikselillä on paikkatiedon lisäksi sen väri tai mustavalkoisessa kuvassa harmaasävy. [15, s. 9].

4.3 Kamerakennot

Nykyaikaisesta digitaalikamerasta löytyy joko CCD- tai CMOS-kuvakenno. Pohjimmiltaan tekniikat ovat samantyyppisiä. Molemmissa valon määrää mitataan pinnalla, joka on valoherkkä ja se tallennetaan varaukseksi. Varaukset voidaan lukea pikseli kerrallaan ja muuttaa numeeriseksi arvoksi. Suurin ero kennojen välillä on käytetty puolijohdetekniikka. [16.]

CCD-kenno (Charged Coupling Devices) on kehitetty jo 60-luvulla. Aluksi se oli tarkoitettu vain muistivälineeksi, koska valoherkkiä pintoja ei ollut vielä saatavilla. Kuitenkin videokuvauksen myötä niitä alettiin tuottaa videoputkien korvaajiksi. Kenno toimii siten, että yksittäisessä pikselissä olevaa varausta voidaan siirtää seuraavaan pikseliin. Tällä tavoin kaikkia kennossa olevia pikseleitä voidaan siirtää reunaan kohti ja reunimmaisissa pikseleissä olevat varaukset kerätään talteen pitkään rekisteriin. Tyhjennyksen aikana varaukset vahvistetaan ja ne kuljetetaan A/D-muuntimelle. CCD-kennon heikkoutena voidaan pitää sen hitautta. [16.]

CMOS-kennon (Complementary Metal Oxide Semiconductor) tarpeelliset toiminnot voidaan valmistaa yhdelle piirille. CMOS-tekniikkaa käytetään kaikissa yleiskäyttöisissä piireissä, kuten muisteissa ja mikroprosessoreissa. Toiminnot luodaan rakentamalla piirille tarvittavia

komponentteja, joilla saadaan aikaan halutunlainen toiminta. Varsinaisen kennon ulkopuolelle, kuitenkin samalle piirille voidaan rakentaa myös muita komponentteja, kuten A/D-muuntimet ja vahvistimet, CMOS-kenno kuluttaa vähemmän virtaa verrattaessa CCD-kennoon. Tekniikkaa kehitetään edelleen ja pikseleiden koko pienenee samalla, kun niiden tiheys kasvaa. [16.]

4.4 Kameran objektiivi

Yleensä objektiivissa on useita linsejä, mutta suurin osa kameran optiikoista voidaan pelkistää yhden kuperan linssin toimintaan. Sen tarkoitus on kerätä kuvattavan kohteen yhden pisteen väri ja valoisuus linssin toiselle puolelle yhdeksi pisteeksi. [17.]

Linssin olennaisin ominaisuus on polttoväli, joka ilmaisee yleisesti aikaisemmin mainitun pisteen etäisyyden linssitasosta. Tuttu efekti lienee suurennuslasilla ja auringolla. Polttovälin etäisyys riippuu vain linssin hionnasta ja materiaalista. Se ilmaistaan usein millimetreinä. [17.]

Muita ominaisuuksia ovat yleensä linssin valovoima, joka on verrannollinen linssin pinta-alaan. Valovoima esitetään usein f-luvulla, joka lasketaan jakamalla linssin polttoväli sen läpimitalla. Mitä suurempi luku on, sitä heikompi on linssin valovoima. [17.]

Linssin pinta-alaa voidaan kuitenkin rajoittaa himmentimellä. Se on säädettävä aukko, joka muodostuu ohuista kaarevista levyistä, jotka liukuvat limittäin muodostaen säädettävän reiän. Sen kuvaukseen käytetään nimitystä aukko. Himmennin sijoitetaan linssien keskelle tai taakse siten, ettei se kuitenkaan häiritse varsinaista kuvan muodostumista. [17.]

5 OHJELMOINTI

Ohjelmointikielien historia alkaa jo 1950-luvun alkupuolelta. Symbolisten konekielien siirrettävyysohjelmien vuoksi alettiin etsiä ratkaisua automaattisesta ohjelmointikielestä. Se perustuu siihen, että ohjelmoija kirjoittaa ongelman ratkaisun kuvauksen jollain formaalilla kielellä, jonka jälkeen jokin erityinen ohjelma muuttaa sen konekieliseksi ohjelmaksi, joka ratkaisee ongelman. Nykyisin formaalikielillä tarkoitetaan jotain korkean tason kieltä, ja konekielen tuottajaa nimitetään kääntäjäksi. Kun ensimmäiset konekielen kääntäjät julkaistiin, ihmisten oli vaikea uskoa, että automaattinen konekielen ohjelmointi olisi mahdollista. [18, s. 18.]

5.1 Ohjelmointikielten kehitys

Nykyisten ohjelmointikielien lähtökohtana voidaan pitää Fortran-kieltä, joka myös edisti luottamusta automaattisen ohjelmoinnin mahdollisuuksiin. Ensimmäisen Fortran-kääntäjän luominen oli tosin haasteellista, sillä sen kehittämiseen kului kolme vuotta aikaa ja 18 ihmistyövuotta. Kielen pääkehittäjänä toimi John Backus, joka sai kehitystyöhön tukea IBM:ltä. Fortran muistutti jo aikanaan nykyäänkin käytettäviä ohjelmointikieliä joidenkin piirteiden perusteella. Kielessä oli mukana taulukkomuuttujat, toistolause ja valintalause. Fortran sai laajalti suosiota ja sen uudemmat versiot ovat edelleen käytössä. [18, s. 18.]

1960-luvun alussa kehitys johti Algol60:n syntymiseen. Tarkoituksena oli tuottaa laitevalmistajasta riippumaton ohjelmointikieli algoritmien toteuttamiseen. Kielen kehitystyö tapahtui saksalais-amerikkalaisena yhteistyönä. Algol60 oli ensimmäinen lohkorakenteinen kieli, ja siinä oli mukana nykyisten kielten kaltaiset kontrollirakenteet ja rekursio. Sen suunnittelussa tarkoituksena oli luoda kieli, joka on mahdollista kääntää mekaanisesti konekielelle. Suunnittelun tuloksena saatiin uusi notaatio ohjelmoinnin syntaksin kuvaamiselle: BNF (Backus-Naur form). Ohjelmointikielenä Algol60 ei kuitenkaan saanut suurta levikkiä akateemisen maineensa vuoksi. Kielellä on silti ollut tärkeä merkitys myöhemmin kehitettyjen ohjelmointikielien suunnittelussa. [18, s. 20.]

Basic oli yksi 1960-luvulla syntyneistä ohjelmointikielistä. Sen tarkoituksena oli luoda kieli, jonka käyttö olisi helppo opettaa kenelle tahansa. Sen pohjana oli Algol60, mutta siitä jätet-

tiin monia tärkeitä piirteitä pois, esimerkiksi lohkorakenne. Lohkorakennetta pidettiin liian vaikeana asiana opetella. [18, s. 20.]

Ohjelmistotuotanto oli voimakkaassa kasvussa 1970-luvulla, kun laitekustannukset pienenevät. Niiden halventuessa ohjelmistotuotannon kustannusten merkitys tuli hyvin oleelliseksi. Ohjelmointi oli kuitenkin virhealtista ja tarvittavien resurssien arviointi oli hankalaa. Ohjelmointiin käytettiin paljon erilaisia kieliä ja niiden variaatioita, ja ohjelmien siirto oli hankalaa. Tästä seurasi lopulta niin kutsuttu ohjelmistokriisi. Sen ratkaisuna oli rakenteellinen ohjelmointi, jota monet 1970-luvulla kehitetyt kielet tukisivat. [18, s. 21.]

Dennis Ritchie kehitti C-kielen 1970-luvulla. Se sai vaikutteita Algol68:sta, vaikkei se sitä syntaksiltaan muistutakaan. Se oli tarkoitettu alun perin raakaan järjestelmäohjelmointiin ja Unix-käyttöjärjestelmä kirjoitettiin kokonaan uudestaan sillä. Siinä olevat tyyppitarkastukset eivät ole täydellisiä, jonka takia sillä tehdyistä ohjelmista saadaan joustavia ja tehokkaita, mutta samalla myös virhealttiita. Varsinaisesti C-kieli ei tuonut mitään uutta, mutta siitä tuli hyvin suosittu ja on siksi merkittävä ohjelmointikieli. [18, s. 21.]

C++-kieli syntyi 1980-luvulla, kun Bjarne Stroustrup laajensi C-kieltä oliorakenteella. Bertrand Meyer puolestaan kehitti Eiffel-kielen, joka on puhdas oliokieli. Eiffelin syntaksi muistuttaa paljon Adaa. [18, s. 22.]

Olio-ohjelmoinnin yleistyessä 1990-luvulla kehitettiin C++:n pohjalta Java-kieli. Sen tarkoituksena oli säilyttää C-kielestä tuttu joustavuus, mutta kuitenkin parantaa ohjelman turvallisuutta. Javassa osoittimet on korvattu viitteillä. Kieleen on myös lisätty automaattinen roskien keruu. Javan ja C++:n pohjalta on myöhemmässä vaiheessa kehitetty vielä C#. C#-kieleen on otettu takaisin Javassa poistettuja C++:n ominaisuuksia, joskin hieman eri muodossa. Myös Internetin käyttö yleistyi 1990-luvulla, mikä vaikutti ohjelmistojen kehitykseen. Internetin yleistymisen johdosta erityisesti script-kielet lisääntyivät, koska ne sopivat HTML-dokumenttien luomiseen ja käsittelemiseen. [18, s. 22.]

5.2 Ohjelmointikielten sukupolvet

Tavallisesti ohjelmointikielten kehittyminen jaetaan viiteen sukupolveen. Ohjelmointikielen jako kuvaa sitä, miten lähellä kieli on suorittavaa laitetta ja toisaalta sitä, miten kaukana se on ohjelmoinnin kohteesta eli sovellusalueista. [18, s. 22.]

Konekielet ovat ensimmäisen sukupolven kieliä. Ohjelmointi on tällöin sananmukaisesti koodaamista. Ratkaisualgoritmi pitää muuttaa binaarikoodiksi, muistiosoitteet ovat viittauksia absoluuttisiin muistiosoitteisiin. Jos ohjelmaan pitää lisätä keskelle uusi käsky, ohjelman loppuosan sijainti muuttuu muistissa, joten myös kaikki loppuosaan kohdistuvat viittaukset täytyy osoittaa uudelleen. [18, s. 22.]

Toisen sukupolven kielillä eli symbolisilla konekielillä suoritettava ohjelmointi on huomattavasti helpompaa. Käskyt on nimetty selkeästi ja myös muistiosoitteet ovat symbolisia. Ohjelma voidaan kääntää erillisiksi moduuleiksi, ja ne voidaan myöhemmin linkittää suoraan ohjelmaksi. Kuitenkin luotujen ohjelmien luku on vaikeaa, ja ne ovat koneriippuvaisia. [18, s. 22.]

Korkean tason lausekieliä kutsutaan kolmannen tason ohjelmointikieliksi. Useimmat nykyisistä ohjelmointikielistä kuuluvat korkean tason ohjelmointikieliin. Niiden tyypillisiä tunto-merkkejä ovat abstraktit kontrollirakenteet ja tietoalkioiden tyypittäminen. Aliohjelmat mahdollistavat uusien abstraktien käskyjen rakentamisen. Ohjelmien luku on suhteellisen helppoa, ja ne ovat helposti siirrettävissä. [18, s. 22.]

Neljännellä sukupolvella tarkoitetaan yleensä sovelluskehittäjäkieltä, jota voidaan ajatella ohjelmajäsenä, jolla on mahdollista generoida suhteellisen helposti jonkin tietyn sovellusalueen ohjelmia. 4GL-lyhenne tarkoittaa neljättä sukupolvea, ja lyhenne esiintyy usein sovelluskehittäjien yhteydessä. Neljännen sukupolven kielet eivät kuitenkaan ole yhtenäisiä, vaan ne ovat esimerkiksi jonkin tietyn kaupallisen valmistajan sovelluksia. Usein sovelluskehittäjän kieli on liitetty graafiseen käyttöliittymään. [18, s. 22.]

Viidennen sukupolven kielillä tarkoitetaan asiantuntijajärjestelmiä, joilla pyritään mallintamaan ihmisen päätöksentekoa ja ajattelumallia. [18, s. 22.]

Edellä mainittua sukupolvijakoa voidaan pitää keinotekoisena, koska useimmat nykyiset ohjelmointikieliset kielet kuuluvat kolmanteen sukupolveen. Neljäs ja viides sukupolvi nähdään vain

usein laitevalmistajien mainoslauseina. Tämän lisäksi funktionaalinen ohjelmointi ei kuulu mihinkään sukupolveen yksiselitteisesti. Jako onkin järkevämpi esittää pidemmälle menevänä kolmannen sukupolven jaotteluna:

1. koneriippumattomat kielet
2. tyyppi- ja kontrollirakenteita sisältävät kielet
3. aliohjelmia tukevat kielet
4. moduuleja ja olioita tukevat kielet
5. geneerisyyttä ja monimuotoisuutta tukevat kielet. [18, s. 22.]

5.3 LabVIEW-ohjelmisto

Kuten jo aikaisemmin mainittiin, Fortran oli ensimmäinen korkean tason ohjelmointikieli. Kun kieli julkaistiin, sen oli vastaanotto ohjelmoijien keskuudessa kovin skeptinen. Ohjelmoijat eivät uskoneet aluksi, että tietokoneella luotu konekoodi voisi olla nopeampi ja parempi, kuin ihmisen kirjoittama konekoodi. Fortran onnistui kuitenkin vähentämään tarvittavien komentojen määrän murto-osaan assembly-komennoista, ja ohjelman suoritus aika ei juuri poikennut perinteisestä ohjelmointimenetelmästä. [19, s. 1.]

Viisikymmentä vuotta myöhemmin voimme edelleen ottaa oppia tästä hullunkurisesta uskomuksesta. Ensinnäkin, yli viidenkymmenen vuoden ajan insinöörit ovat etsineet helpompia ja nopeampia menetelmiä koneelliseen ongelmanratkaisuun. Toiseksi, tutkijat valitsevat tehtävien suorittamiseen yhä useammin kielen, joka on korkeantason ohjelmointikieli. Tämä selittää graafisen ohjelmoinnin saaman suuren suosion, kun se ensimmäisen kerran esiteltiin vuonna 1986. Graafinen G-kieli edustaa todella korkeantason ohjelmointikieltä, jonka tarkoituksena on kasvattaa ohjelmoijien tuottavuutta samalla toimien lähes yhtä nopeasti, kuin matalamman tason ohjelmointikielien kuten Fortran tai C++. [19, s. 1.]

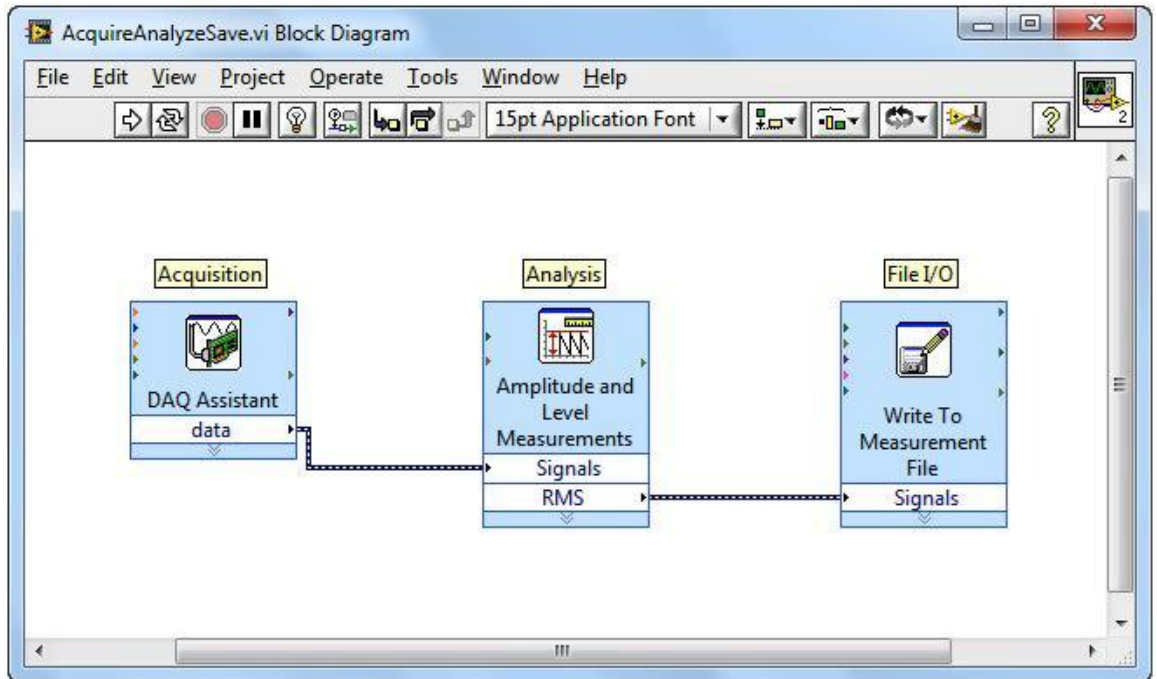
5.3.1 LabVIEW 2011, graafinen tietovirtaohjelmointi

LabVIEW poikkeaa valtaosasta muista yleistarkoitusohjelmointikielistä kahdella suurella erotuksella. Ensiksi, G-kielissä ohjelmointi tapahtuu yhdistelemällä graafisia kuvakkeita eli ikoneja toisiinsa. Näistä ikoneista muodostuu sitten kokonaisuuksia, aivan kuten elektronikan piirikaavioita, joista voidaan kääntäjän avulla muodostaa konekielinen ohjelma, jonka tietokoneen prosessori voi suorittaa. Vaikkakin G-kielen esitys on tekstin sijasta graafinen, löytyy siitä paljon yhteisiä tekijöitä perinteisten ohjelmointikielten kanssa. Esimerkiksi G-kielessä on kaikki normaalit rakenteet, kuten silmukkarakenteet, tapahtumien hallinta, muutujat ja olio-ohjelmointi. [19, s. 1.]

Toinen eroavaisuus perinteisiin tekstipohjaisiin ohjelmointikieliin on se, että LabVIEW toteuttaa suoritettavat toiminnot tietovirran sääntöjen perusteella. Perinteisessä ohjelmoinnissa suoritusjärjestys tapahtuu riviriviltä edeten, kuten C ja C++ -kielissä. Tietovirtaan perustuvat graafiset kielet kuten G-kieli, Agilent VEE, Microsoft Visual Programming Language ja Apple Quartz Composer pitävät tietoa ohjelman tärkeimpänä osana. Tietovirtaan pohjautuvassa kielessä suoritus on tieto-ohjattua tai tietoriippuvaista. Ohjelmassa siirtyvät tietovirrat määrittävät ohjelman suoritusjärjestyksen erillisten rivien sijaan. [19, s. 2.]

Tämä erotus voi tuntua aluksi pieneltä, mutta sen vaikutus on todella suuri, koska tietovirtojen reitit ohjelmassa on ohjelmistosuunnittelijan päähuomiona. LabVIEW:ssä tehdyt liitokset funktioihin, aliohjelmiin ja muihin rakenteisiin vaikuttavat ohjelman suoritusrutiiniin. Jokaisella ohjelman ikoneilla on herätteitä, joiden perusteella ne muodostavat vasteita. Kun jokin ikoni tai funktio on saanut tiedon kaikkiin siihen liitettyihin tietovirtoihin, suoritetaan toiminto ja ikoni tai funktio lähettää tiedon eteenpäin seuraavalle ikonille tai funktiolle. Tästä määräytyy tarkka suoritusjärjestys ohjelmalle. [19, s. 2.]

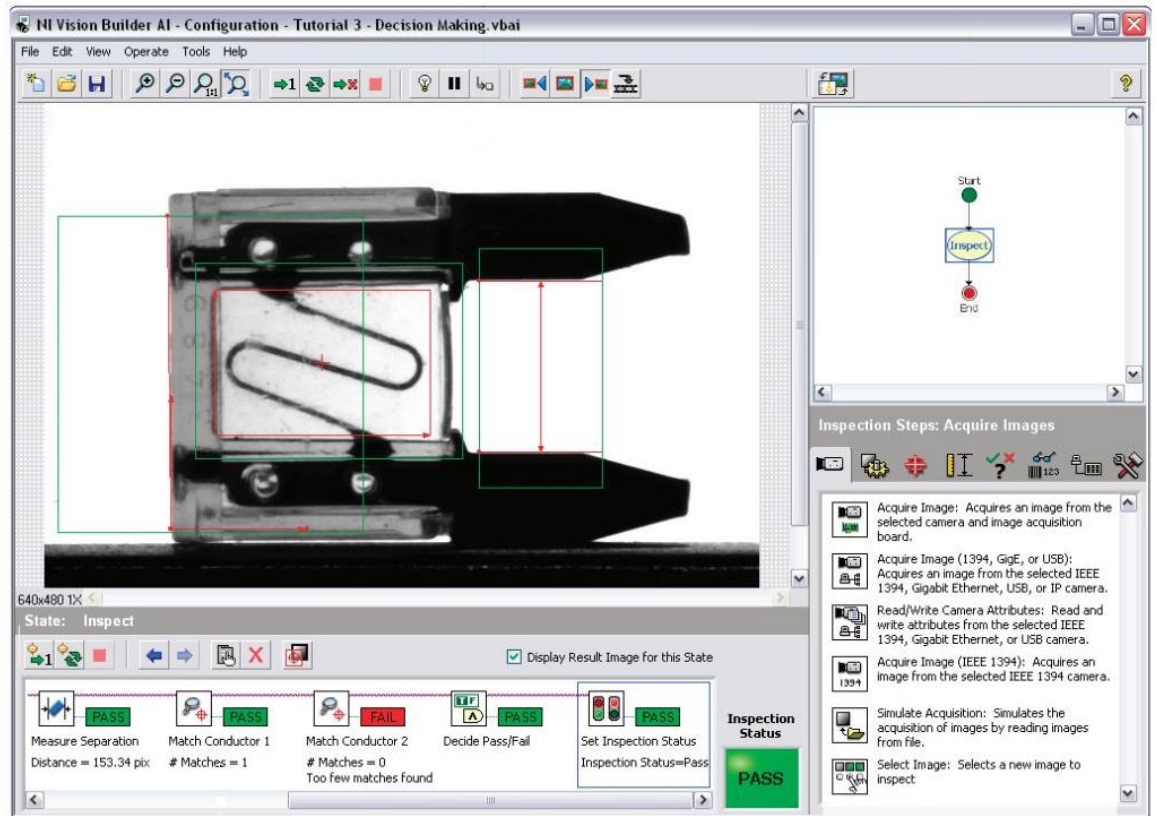
G-koodia on tyypillisesti helppo ymmärtää, koska koodin rakenne muistuttaa paljon oikeaa mittaustapahtumaa. Koodia luetaan vasemmalta oikealle ja tiedot siirtyvät loogisesti ikonista toiseen. Esimerkiksi yksinkertainen ja usein käytetty G-kielillä tehty ohjelma voi olla tiedonkeruuohjelma. Ohjelman rakenne ja vaiheet näkyvät selkeästi koodista, jossa ensin hankitaan tiedot tiedonkeruukortilta, tiedot siirretään analysoitavaksi ja analysoinnin jälkeen tallennetaan tulokset tiedostoon (kuva 6). [19, s. 3.]



Kuva 6. Yksinkertaisen tiedonkeruuohjelman lohkokkaavio [19, s. 3].

5.3.2 NI Vision Builder 2011

National Instruments Vision Builder -ohjelma on suunniteltu helpottamaan automaattisten konenäkösovellusten luomista. Siinä voi luoda konenäkö tarkastuksia ilman varsinaista ohjelmointia. Ohjelmalla luotua tarkastusta voidaan siis ajaa semmoisenaan. Siinä on mahdollista toteuttaa mittauksia oikeilla mitta-asteikoilla, etsiä erilaisia muotoja kappaleista, lajitella tarkasteltavia kohteita värin tai ulkomuodon perusteella ja tarvittaessa tehdä päätös visuaalisen tarkastelun hyväksymisestä tai hylkäämisestä. [20, s.1.]



Kuva 7. Sulakkeen visuaalinen tarkastus Vision Builder -ohjelmassa [21].

Kuvassa 7 on Vision Builder -ohjelman käyttöliittymä. Esimerkkitarkastuksessa ohjelma tarkistaa sulakkeen jalkojen välisen välimatkan sekä onko sulakkeen lanka kunnossa. Vision Builderilla rakennetut tarkastukset on mahdollista siirtää LabVIEW-ohjelmaan, jossa niitä voi muokata vapaasti.

6 METSO OC -SAKEUSMITTARIN TESTAUS

Metso OC -sakeusmittarissa (kuva 8) käytetään valon siirtämiseen optista kuitukaapelia. Kaapelissa on sisällä seitsemän erillistä kuitua, joista kuutta kuitua pitkin valo lähetetään prosessiputkessa olevan yhteen läpi paperimassaan. Valo siroaa massassa olevista kuiduista takaisin ja sironneen valon intensiteetin perusteella voidaan määrittää paperimassan sakeus.



Kuva 8. Metso OC -sakeusmittari.

Kuvassa 8 on eroteltuna sakeusmittarin toimitussisältö:

1. Operointiyksikkö TCU (Transmitter Central Unit) on laitteen paikallinen ohjauksen keskus. Siitä on mahdollista lukea hetkellisiä mittauksia, ja sen voi kytkeä tehtaalla olevaan säätöjärjestelmään. Lähettintä voi siis operoida paikallisesti tai etäyhteydellä.

2. Optinen yksikkö (Optical Unit). Sen sisällä on lähettimen valonlähteenä toimiva led sekä tarpeellinen mittaus- ja ohjauselektroniikka. Ledin virittäminen tapahtuu sisällä olevaa lediä kääntämällä sekä kiertämällä piirilevyllä olevaa potentiometriä.
3. Mittakärki, jonka avulla ledin lähettämä valo voidaan siirtää prosessiyhteen (4) läpi mitattavaan massaan.
4. Prosessiputkeen liitettävä läpivientiyhde.

Optisen yksikön sisällä kaapelin päähän on liimattu holkki, jossa ledi on kiinni kierteellä. Holkin jälkeen, noin 10 cm päässä valonlähteestä liitos jossa kaapelista on erotettu keskimäinen kuitusäie. Säie on katkaistu ja sitä on jatkettu siten, että ledin tuottamaa valon intensiteettiä on mahdollista mitata detektorin avulla. Näin saadaan valon intensiteetin referenssiarvo. Vastaavasti katkaistun säikeen toinen pää jatketaan myös, ja siihen on lisätty detektori, jonka avulla voidaan mitata massasta takaisin siroavaa valoa. Tästä syystä mittapäässä olevista kuiduista vain kuusi ulommaista on valaistu. Keskimäinen kuitu toimii siis valon kulkureitinä takaisin kaapelin alkuun.

Koska kaapelissa olevat kuidut ovat läpimitaltaan todella pieniä, on niitä mahdotonta havaita paljaalla silmällä katsottaessa. Kuidusta ei voi myöskään havaita valmistusvirheitä ilman lisälaitteita. Koska kaapeliin lähetettävä valo tulee yhdestä ledistä, on mahdollisia intensiteettieroja yksittäisten kuitujen välillä mahdotonta todeta ilman lisälaitteita. Intensiteettierojen havaitsemiseen on poikkeuksetta käytettävä digitaalista kameraa. Sen käyttämiseen tarvitaan kuitenkin ohjelmisto.

6.1 Käytettävä kamera

Tarkoitukseen sopiva kamera ja optiikka löytyivät jo valmiiksi Metsolta. Alkuperäisessä ideassa oli tarkoitus käyttää Ethernet-kameraa. Kamera olisi kuitenkin vaatinut erillisen virtalähteen, joten kamera korvattiin USB-väylään liitettävällä kameralla. Kameraksi valikoitui IDS:n UI-1545LE-M-kamera. Siinä on 1,3 megapikselin 1/2" CMOS-kenno. Kameran resoluutio oli riittävä tähän sovellukseen ja harmaasävykamerana siitä saatavien kuvien käsittely ei vaadi niin paljon laskentatehoa kuin värillisten kuvien käsittely vaatisi. Värikamera ei toisi sovelluk-

seen ollenkaan lisäarvoa. Kamera on kompaktin kokoinen ja sen rungossa on jo valmiiksi C-tyypin kiinnitys objektiiville. Kamera on esitelty kuvassa 9.



Kuva 9. IDS UI-1545LE-M -harmaasävykamera [22].

6.2 Optiikka

Työssä käytetty linssi (kuva 10) on ns. makrolinssi, joka on tarkoitettu kohteen kuvaamisen lähietäisyydeltä. Linssin polttoväli on 28 mm ja siinä on säädettävä aukko. Loittorenkaiden avulla kuva saadaan suurennettua riittävän suureksi kameran kennolle.



Kuva 10. Schneider 2.8/28 Makro-linssi 66 mm loittorenkailta.

6.3 Testijigi ja muut tarvikkeet

Kuvassa 11 oleva testijigi on valmistettu yhtenäisestä eloksoidusta alumiinista. Siinä oleva kirkas osa on mittakärjen tukemista varten. Mittakärki asetetaan kirkkaan osan mukaisesti siten, että se osoittaa valoa kohti kameraa. Kameran etäisyyttä mittakärkeen on mahdollista muuttaa liikuttamalla sitä lähemmäs tai kauemmas mittakärjestä. Kamera lukitaan paikoilleen pitoruuvilla.



Kuva 11. Testijigi ja kalibrointilaatta.

Kuvassa oleva musta lieriö on kalibrointilaatta, joka asetetaan mittakärjen päähän virityksen aikana. Sen sisällä olevavasta materiaalista valo siroaa aina samalla tavalla, ja siitä sironneen valonmäärää käytetään referenssiarvona. Laatan avulla jokainen laite normalisoidaan tarkasti.

Lisäksi järjestelmään kuuluu RS-232-sarjakaapeli, jonka avulla voidaan lukea sarjaliikennettä TCU:lta. Kaapelissa on myös ohjelmointiliitin, joka mahdollistaa TCU:n ohjelmiston päivittämisen tarvittaessa.

7 OHJELMAN SUUNNITTELU JA TOTEUTTAMINEN

Tässä luvussa käsitellään teknisiä ratkaisuja, joiden avulla ohjelma on suunniteltu ja toteutettu.

7.1 Vaatimusmäärittely tulevalle ohjelmistolle

”Vaatimusmäärittely (Requirements Engineering) on yksi ohjelmistojärjestelmien kehitystyön perustehtävistä” [23]. Yleensä ohjelmiston suunnittelu alkaa yksityiskohtaisella vaatimusmäärittelydokumentilla. Ohjelmiston suunnittelu ja toiminnot pohjautuvat vaatimusmäärittelyssä oleviin toiminnallisiin ja ominaisuuksiin. Luvussa 7.2 on esiteltyä tilaajalta saatu vaatimusmäärittely tulevalle ohjelmistolle.

7.2 Metso OC Sensorin optisen testerin vaatimusmäärittely

Testauksen tarkoitus

Testauksen tarkoitus on varmistaa, että kuitukoannon liimaus on onnistunut ja kuidut ovat ehjiä. Testausta varten tarvitaan testausjigi, johon anturi kiinnitetään ja jossa on kiinteästi asennettu kamera. Kameran kuva siirretään PC:lle, jolla kuva käsitellään. Testaus- ja viritysvaiheessa tarvitaan valolähde ja kuidut.

Tarvittavat työkalut

- Testausjigi
- Kamera
- PC
- Testeriohjelmisto.

Testeriohjelmiston vaatimukset

1. Itsenäisesti toimiva sovellus.

2. Testeri sovelluksessa 2 osiota
 - 2.1. Kameraperustainen kuitutesti.
 - 2.2. Anturikortin viritystesti.
3. Kameraperustainen kuitutesti
 - 3.1. Lukee kuvan käsittelyikkunaan, jossa kuvan analysointi.
 - 3.2. Näyttää elävää kuvaa kuidun päästä. (ei still-kuva).
 - 3.3. Kuitujen osalta tarkastellaan kuvasta harmaasävyjä, muuttamalla harmaasävyt pseudovärikartaksi esim. rainbow.
 - Vurityksessä pyritään saamaan mahdollisimman tasainen ja voimakas valo kuitujen päähän. Viritys tehdään pyörittämällä LED-liitintä kuituliittimen suhteen.
 - Optimaalisin (korkea ja tasainen intensiteetti) kohta lukitaan lukitteella.
 - 3.4. Sovelluksen tunnistettava kuidut (6 kpl).
 - 3.5. Määritettävä kuidun keskipiste ja sen ympäriltä tietty tarkastelualue.
 - 3.6. Laskettava jokaisen kuidun keskimääräinen valointensiteetti tarkastelualueelta.
 - 3.7. Säädetessä minimoidaan suhteellinen keskihajonta-arvo (lasketaan: keskihajonta/keskiarvo). Näille tehdään näytölle arvokentät.
 - 3.8. Suhteelliselle keskihajonnalle näytössä pitää olla myös "mittari", jonka avulla on helpompi määritellä pienin mahdollinen keskihajonnan arvo. Testaaja lukitsee ledin aseman kohtaan, jolla pienin arvo saavutettiin.
 - 3.9. Yläraja-arvo keskihajonnalle määritetään myöhemmin.
 - 3.10. Mikäli kuitu on viallinen, testaaja keskeyttää testin. Sovelluksen ei tarvitse tulkita kuiturikkoa.
 - 3.11. Testiraporttiin tallennetaan:
 - Kuidun Item tunnus + versio
 - Kuidun sarjanumero
 - Testauspäivämäärä
 - Kuva, jossa on määritetyt intensiteetit ja lasketut arvot
4. Anturikortin viritystesti
 - 4.1. Testin aikana anturin kärkeen liitetään kalibrointilaatta.
 - 4.2. Lukee TCU:n kautta anturikortilta: LED-virta, Meas-arvo ja VComp-arvo. Nämä arvot myös talletetaan raporttiin.
 - 4.3. Sovellus käynnistää Factory Testin toiminnon TCU:n debugger-portin kautta.

- 4.4. Sovellus seuraa Meas-arvoa. Arvo isohkolla fontilla näytöllä, fontin väri esim punainen, kun arvo poikkeaa asetusarvosta yli ± 10 mV.
- 4.5. Testiparametri luetaan Testi.ini tiedostosta. Oletusarvo 0,450 mV
- 4.6. Muut testi.ini parametrit:
 - Meas-arvon asetusarvo
 - Käytettävä COM-portti
 - COM-portin parametrit

Vaatimusmäärittelyn pohjalta suunniteltiin toteutettava ohjelmisto ja sen eri vaiheet. Toteutuksen aikana ohjelmistoon tuli lisävaatimuksia, kun uusia, tarpeellisia toimintoja havaittiin.

Ohjelman suunnittelu- ja toteutusvaiheet on eritelty seuraaviin kappaleisiin.

7.3 Kameran liittäminen tietokoneeseen ja kuvan saattaminen näytölle

Kameran liitetään tietokoneeseen USB-kaapelilla, ja se saa myös käyttöjännitteen sitä kautta. Kameran valmistaja tarjoaa ajureiden lisäksi ohjelmiston, jossa sen toimintoja ja ominaisuuksia voi testata. Ajureiden asentamisen ja kameran liittämisen jälkeen kameran muodostama kuva saatiin tietokoneelle.

Valmistaja tarjoaa myös muutaman esimerkkikoodin LabVIEW-ympäristöön, mutta ne eivät olleet käyttökelpoisia puutteellisen kuvankäsittelyn takia. Seuraavaksi täytyi siis saada kameran muodostama kuva tuotua LabVIEW-ympäristöön siten, että kuvia olisi mahdollista käsitellä ja analysoida.

7.4 Mittakärjen ja kuitujen paikantaminen

Jokaisen sakeusmittarin mittakärki on yksilöllinen, joten kärjen keskipisteen määrittäminen on välttämätöntä ohjelman toiminnan kannalta. Ennen kuin ohjelman varsinainen keskipisteenmäärittämysosio suunniteltiin, suoritettiin yksittäisille kuiduille testauksia ja niiden absoluuttinen asema kameran kennolla tarkistettiin. Testauksien perusteella havaittiin, että keskipisteen poikkeama toisistaan oli kymmeniä pikseleitä mittakärkien kesken, joten ohjelmaan oli välttämätöntä suunnitella automaattinen keskipisteen tunnistusosio.

Ohjelmaan on myös lisätty varmuuden vuoksi manuaalinen keskipisteen määrittämismahdollisuus, jolla voidaan varmistaa, että ohjelma etsii valokuituja oikeista kohdista. Manuaalinen määrittäminen onnistuu suoraan ohjelman näyttämästä kamerankuvasta osoittamalla hiirellä haluttu keskipiste.

LabVIEW Vision Builder -lisäosan avulla ohjelmaan luotiin sekvenssi, jossa kuvasta määritetään ensiksi anturin keskipiste. Keskipiste etsitään kuvasta funktiolla "Find Circular Edge". Funktiolle luodaan seuraavaksi "ROI" (Region of Interest), joka sisältää etsittävän alueen keskipisteen, alueen sisä- ja ulkohalkaisijan sekä asetukset. Keskipisteenmäärityksessä kuvasta lähdetään ROI:n ulkoreunoilta lähestymään keskipistettä ja kun kuvassa tapahtuu intensiteettimuutos kirkkaammaksi, määritetään ympyrän reuna sille kohdalle. Ohjelma osaa nyt tulkita kuidun aseman suhteessa kameran kennoon.

Kun kuitukaapelin keskipiste on määritetty, etsitään keskipisteen ympäriltä tietyn säteen päästä yksittäisiä kuituja. Koska kuidut sijaitsevat tasaisin välein kaapelissa, on niiden etsintä hieman yksinkertaisempaa kuin satunnaisesti sijoitettujen kuitujen etsintä. Kuvasta etsitään kuituja kuuden erillisen ROI:n avulla, joilla jokaisella on oma keskipiste ja halkaisija. Koska kuidut sijaitsevat tietyssä asemassa suhteessa koko kaapelin keskipisteeseen, aloitetaan kuitujen reunojen etsiminen intensiteetin heikentymisen perusteella. Kun etsintä loittonee yksittäisen kuidun keskipisteestä tarpeeksi, alkaa kuidun intensiteetti laskea ja näin löydetään kuidun reuna. Kuiduista mitataan valon intensiteettiä, jota käytetään hyödyksi laskuissa myöhemmässä vaiheessa.

7.5 RS-232-yhteyden luominen mittakortille ja ledin virran lukeminen

Kuidun testaamisen ja virittämisen lisäksi ohjelman täytyy pystyä lukemaan ja kirjoittamaan laitteen tietoja käyttämällä RS-232 -sarjaliikennettä. Yhteyden luominen TCU:lle on suhteellisen selkeä, koska LabVIEW tarjoaa tarkoitukseen helppokäyttöisiä ja suoraviivaisia funktioita, joiden avulla sarjaliikenteen toteuttaminen on mutkatonta. Ohjelmiston on tarkoitus kommunikoida Metso OC:n TCU-yksikön kanssa. TCU:ssa on valmiiksi toiminnot sarjaliikennöintiä varten ja yhteyden luominen onnistui mutkattomasti. Alkuvaiheessa ohjelmistolla oli vain tarkoitus lukea laitteelta tietoja tietyillä komennoilla. Myöhemmässä vaiheessa ohjelmistoon lisättiin toimintoja, joilla voidaan kirjoittaa tietoja TCU:lle.

7.6 Raportin ja sertifikaatin tallentaminen ohjelmallisesti

Ohjelmisto tallentaa testien jälkeen jokaisesta laitteesta sarjanumeron perusteella yksilöidyn testausraportin, jossa on kuva kuidunpäästä, kamerasta saadut intensiteettimittaukset, ledistä saadut mittaukset ja laitteen tunnistetiedot. Tiedot tallennetaan Excelillä tehtyyn raporttipohjaan. Raportoinnin jälkeen ohjelma tallentaa myös laitteen sarjanumerot erilliseen taulukkoon, josta löytyvät kaikki ohjelmalla raportoidusti testatut laitteet. Liitteessä 1 on esimerkkituloste ohjelman tallentamasta raportista.

Myös jokaisesta laitteesta tallennetaan sertifikaatti, joka saadaan sarjaliikenteen välityksellä TCU:lta. Sertifikaatti tallennetaan myös sarjanumeron perusteella. Liitteessä 2 on ohjelman tallentama esimerkkisertifikaatti laitteesta.

8 OHJELMISTON TOIMINTA JA TESTAUS

Tässä osiossa käydään läpi ohjelmiston toiminnallisuutta yksityiskohtaisesti. Ohjelmisto on luotu osittain "Event"-pohjaiseksi. Ohjelmassa tapahtumat lisäävät halutun toiminnon jonoon, josta ne suoritetaan sinne laitetussa järjestyksessä. Sen liiallinen käyttö on kuitenkin estetty poistamalla painikkeet käytöstä suoritusten ajaksi ja tällä tavoin on myös pienennetty käyttäjän mahdollisuutta saattaa ohjelmisto virheelliseen tilaan. Ohjelmassa olevia tapahtumia voi pitää omina aliohjelmina, vaikka ne eivät varsinaisesti ole niitä. Tällainen rakenne mahdollistaa muuttujien kuljettamisen yksinkertaisesti ohjelman vaiheesta toiseen.

8.1 Käyttäjälle näkyvät toiminnot

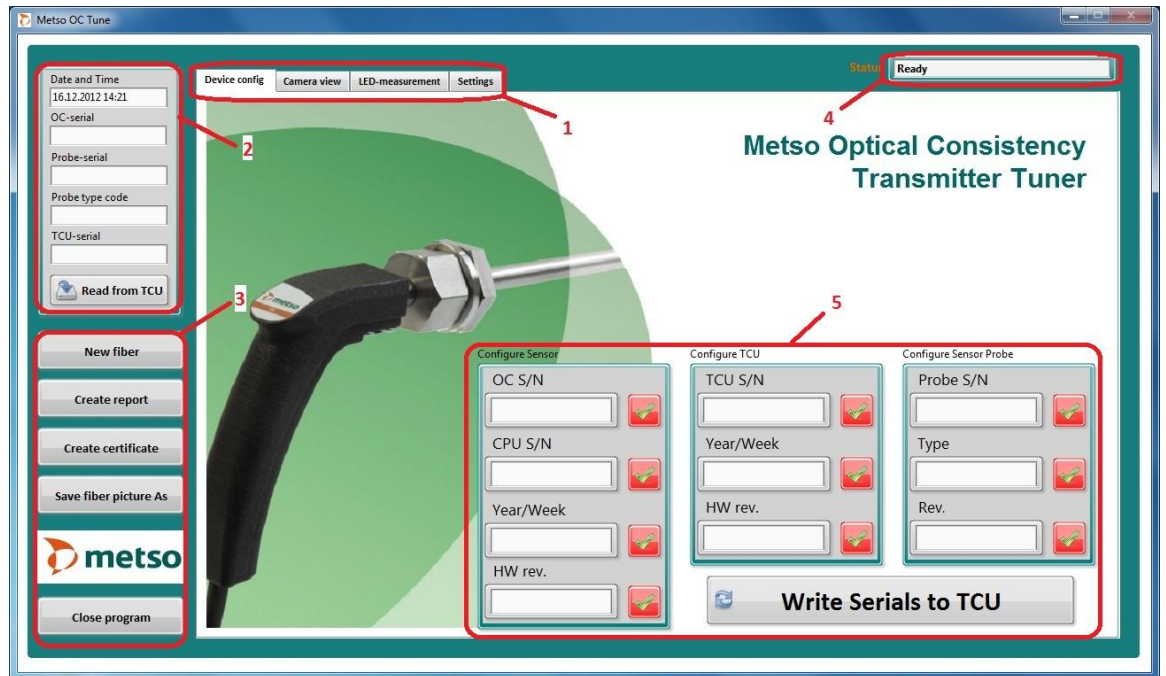
Ohjelmassa on neljä varsinaista näkymää, joista kolmea käytetään Metso OC -laitteen testauksessa. Näkymät ovat:

- Päänäkymä, joka avautuu, kun ohjelma käynnistetään.
- Kameranäkymä
- Mittausnäkyvä
- Asetukset

Seuraavat neljä osiota käsittelevät näitä näkymiä ja niiden toimintoja.

8.1.1 Device config -välilehti

Kun ohjelma käynnistetään, tulee näytölle kuvan 12 mukainen ikkuna.



Kuva 12. Metso OC Tuner -ohjelmiston päänäkymä numeroviitteillä.

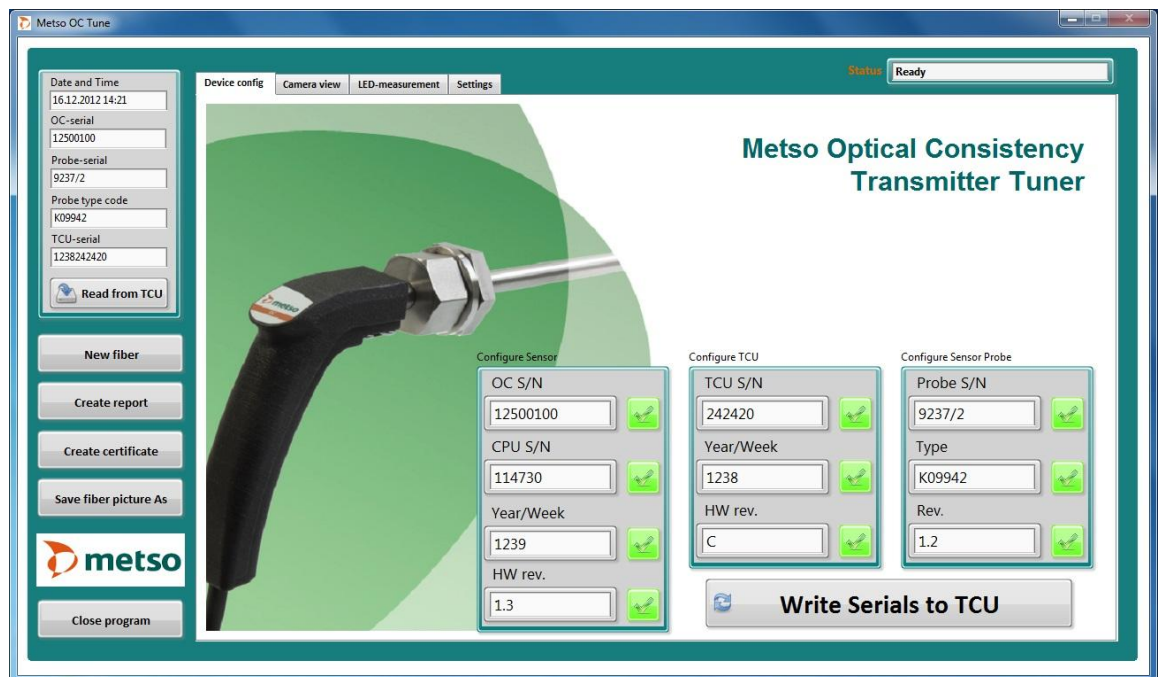
Kuvaan 12 on eroteltu ohjelman kannalta oleellisia kohtia seuraavasti:

1. Välilehtien hallinta, ohjelman näkymää vaihdetaan valitsemalla oikea välilehti.
2. Viritettävän laitteen tiedot, kellonaika tulee automaattisesti kun ohjelma käynnistetään, "OC-serial" määrää ohjelman tallentamien raporttien tiedostonimen, "Probe-serial" määrää kuidusta tallentuvan kuvan nimen. "Type code" ja "TCU-serial" ovat tarkastuskenttiä. Painikkeella "Read from TCU" luetaan tiedot niille varattuihin kenttiin. Normaalisti näitä tietoja ei kirjoiteta itse.
3. Ohjelman hallintapainikkeet, "New fiber" -painike tyhjentää kentät ja asettaa kellonajan oikein kun testataan uusi laite sekä vaihtaa näkymän ensimmäiselle välilehdelle. "Create report" ja "Create certificate" painikkeet luovat testatun laitteen raportit. "Save fiber picture As" tallentaa kuvan kuidusta esim. reklamaatiota varten. "Close program" -painikkeella suljetaan ohjelma.

4. "Status"-ikkuna näyttää ohjelman toimintavaiheen. Kun status kentässä on "Ready", voi ohjelma ottaa vastaan uuden komennon käyttäjältä.
5. TCU:lle kirjoitettavat tunnistetiedot, jotka lähetetään "Write Serials to TCU"-painikkeella

"Device config" -välilehdellä aloitetaan laitteen tarkistaminen syöttämällä TCU:lle sen tiedot sarjaliikenteen välityksellä. "Write Serials to TCU" -painikkeella ohjelmisto kirjoittaa käyttäjän syöttämät tunnistetiedot TCU:lle. Jos jokin syötteistä puuttuu, ei kyseistä tietoa kirjoiteta sinne ollenkaan. Kirjoituksen jälkeen ohjelma tarkistaa, että uusi tieto on mennyt perille oikein. Jos vastauksena tuleva viesti on sama kuin sinne lähetetty viesti, tulkitsee ohjelma kirjoituksen onnistuneen ja vaihtaa sen kohdalla olevan kuvakkeen vihreäksi. Vastaavasti kirjoituksen epäonnistuneessa kuvake vaihtaa väriään punaiseksi ja käyttäjän tulee tarkistaa ongelma. Tiedot on myös mahdollista kirjoittaa TCU:lle suoraan operointiyksiköltä. Kun TCU:n kaikki tiedot on kirjoitettu, tarkistetaan tiedot vielä klikkaamalla vasemmalla olevaa "Read from TCU" -painiketta, jolloin ohjelma tulostaa TCU:lta saadut tiedot oikeisiin kenttiin. Kun tiedot on varmistettu, voidaan siirtyä seuraavalle välilehdelle.

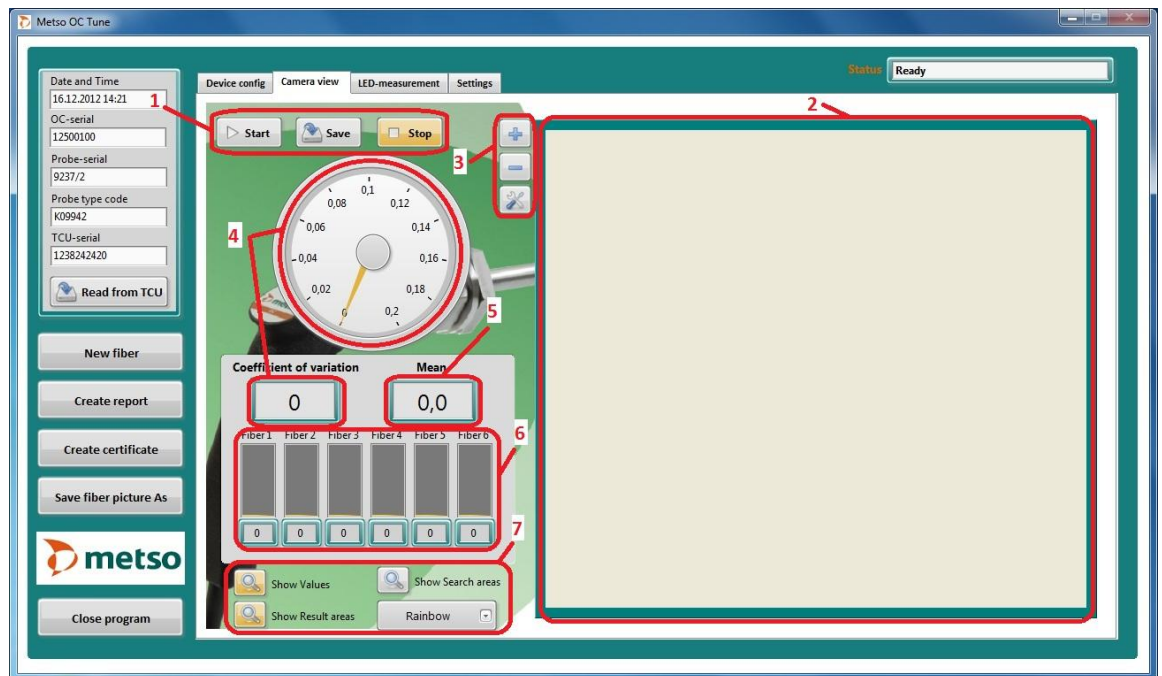
Kuvassa 13 on ohjelman näkymä, kun tiedot on kirjoitettu TCU:lle onnistuneesti ja ne on vielä tarkastettu lukemalla laitetiedot "Read from TCU" -painikkeella.



Kuva 13. Metso OC Tuner, päänäkymä kirjoituksen ja lukemisen jälkeen.

8.1.2 Camera view -välilehti

Kameravälilehti on esitetty kuvassa 14.



Kuva 14. Metso OC Tuner, kameranäkymä numeroviitteillä.

Kuvaan 14 on eroteltu ohjelman kannalta oleellisia kohtia seuraavasti:

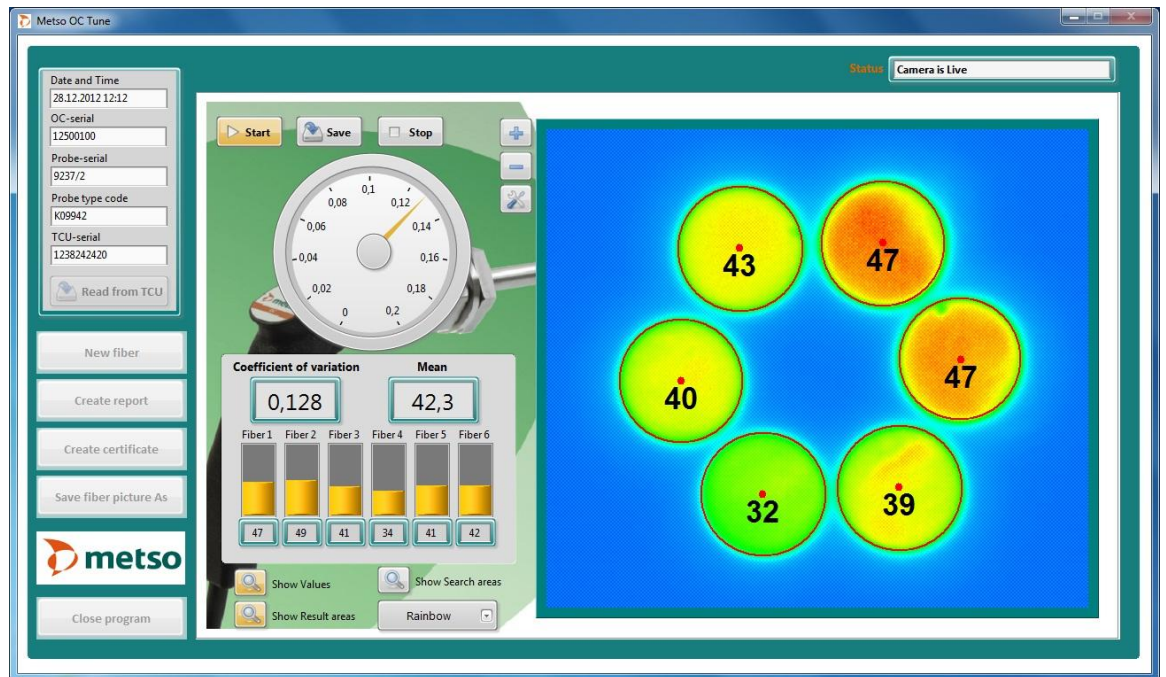
1. "Start"-painikkeella aloitetaan kuvan näyttäminen, "Save"-painikkeella tallennetaan kuidunkuva, kun säätö on suoritettu" ja "Stop"-painikkeella se lopetetaan.
2. Kun kamera on käynnistetty, näkyy sen muodostama kuva ikkunassa (2).
3. Ikkunassa olevan kuvan suurennustasoa on mahdollista muuttaa + ja - -painikkeilla. "Työkalu"-kuvakkeella voi käyttäjä määrittää kuitukaapelin keskipisteen manuaalisesti, jos automatiikka ei tunnista kuitua.
4. Mittari, joka näyttää kuitujen suhteellisen keskihajonnan, lukuarvo pyritään säätämään testin aikana mahdollisimman pieneksi.
5. Kuitujen intensiteetin keskiarvo. Mittausten aikana kameran objektiivissa oleva aukko pidetään arvossa 5. Jos kuva on ylivalottunut tai himmeä, voi aukkoa tarvittaessa muuttaa. Liian suuri aukko ylivalottaa kuvan.

6. Yksittäisten kuitujen intensiteettiarvot pylväinä ja numeroina.
7. Ikkunassa (2) näytettävän kuva ominaisuudet. "Show values"-painike näyttää kuitujen intensiteetin kuvassa kuidun keskellä. "Show Result areas"-painike näyttää kuitujen reunat ja keskipisteen kuvassa. "Show Search areas"-painike näyttää kuvassa alueet, joista ohjelma etsii kuituja. Alasvetovalikosta voi muuttaa käytettävää kuvapalettia.

Kamerasta saatavan kuvan näyttäminen aloitetaan "Start"-painiketta. Kun kamera on saanut tarvittavat komennot, tulee sen muodostama kuva näkyville kuvassa 12 oikealla puolella näkyvään ikkunaan (2). Kuva keskitetään ja rajataan automaattisesti siten, että mittakärki näkyy suurennettuna kuvassa. Oletuksena kuvassa kuituihin on lisätty niiden yksittäiset intensiteettiarvot niiden keskelle. Kuidusta mitattavat intensiteettiarvot keskiarvoistetaan kymmenen edellisen mittauksen perusteella, jolla voidaan pienentää kohinan vaikutusta. Jokaisen kuidun intensiteetit näytetään pylväinä ja numeroina näkymän vasemmassa laidassa (6). Keskiarvoistetuista intensiteettiarvoista lasketaan vielä erikseen keskiarvo, joka näkyy käyttäjälle "Mean"-ruudussa.

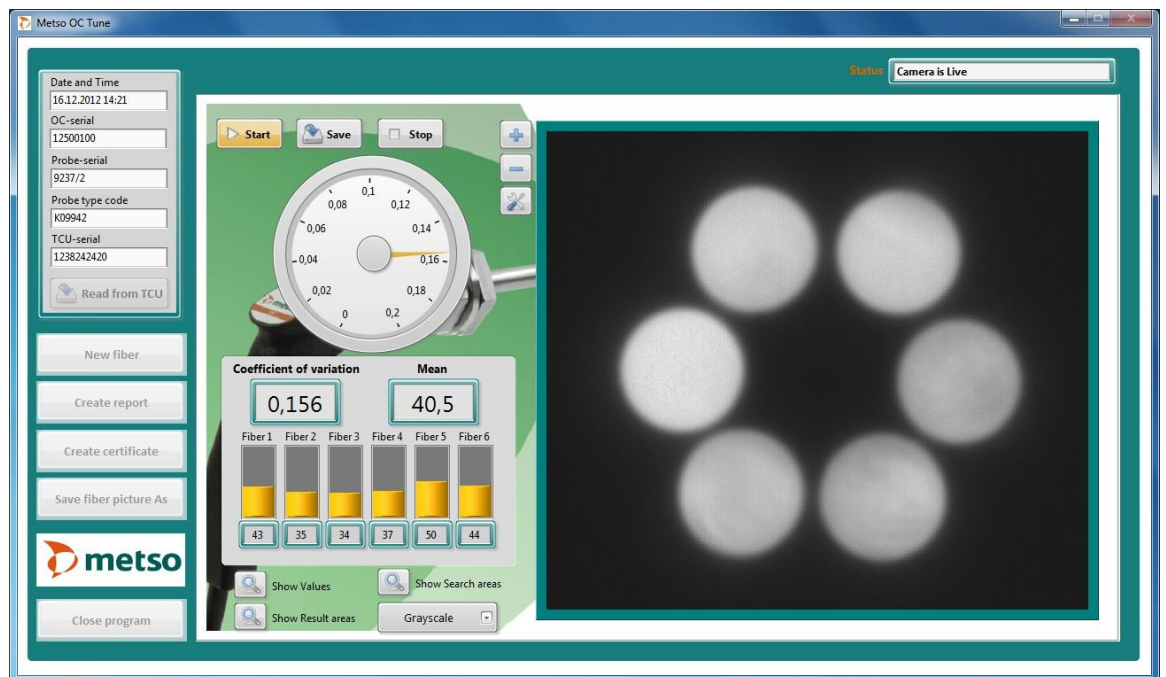
Intensiteettiarvoista lasketaan myös suhteellinen keskihajonta, joka pyritään saamaan mahdollisimman pieneksi viritysvaiheessa. Nämä tiedot näkyvät sekä analogisella mittarilla että "Coefficient of variation" -ruudussa. Kameran muodostama kuva näkyy oletusarvoisesti "Rainbow"-väreillä, josta intensiteettierot näkyvät huomattavasti selkeämmin kuin harmaasävykuvasta. Kuvan värikarttaa on mahdollista muuttaa vaihtamalla alasvetovalikon tilaksi joku muu kuvapaletti. Kuvasta saa myös poistettua mittausarvot ja keskipisteet, jos käyttäjä haluaa ottaa kuvan pelkästä kuidusta esimerkiksi reklamaatiota varten. Kun led on viritetty kääntämällä sitä TCU:n päässä parhaaseen mahdolliseen hajonnan ja intensiteetin tasoon, tallennetaan kuva "Save"-painikkeella ja kameran muodostaman kuvan näyttäminen lopetetaan "Stop"-painikkeella. Kun toiminto on loppunut, käyttäjän on mahdollista vaihtaa ohjelman näkymää.

Kuvassa 15 on "Camera view"-välilehti, kun ohjelma mittaa säädettävää kuitua. Säädön ajaksi kaikki painikkeet, joista voi olla haittaa ohjelman toiminnan kannalta, esimerkiksi raportointi tai muut välilehdet, on asetettu tilaan, jossa niitä ei voi käyttää tai ne on piilotettu kokonaan näkyvistä.



Kuva 15. Metso OC Tuner, kamerakuva kuiduista.

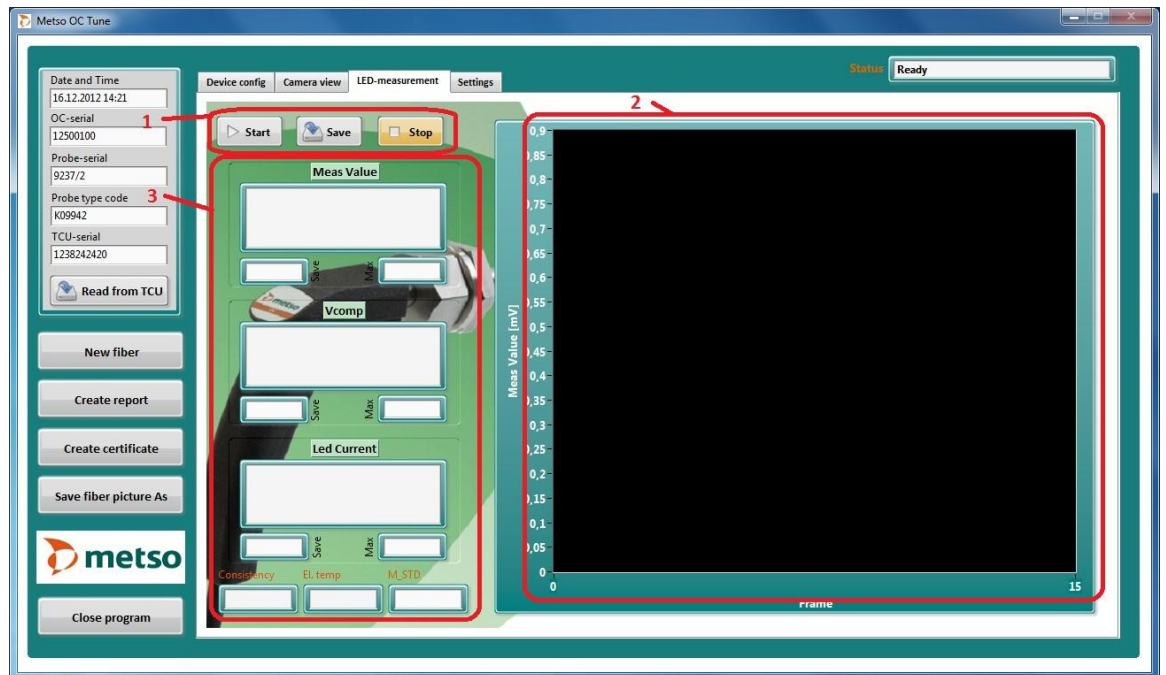
Kuitujen intensiteettierot näkyvät selkeästi "Rainbow"-paletilla, jossa sininen sävy edustaa pientä intensiteettiä ja punainen sävy edustaa suurta intensiteettiä. Harmaasävykuvasta intensiteettieroja on hankalampi havaita (kuva 16).



Kuva 16. Metso OC Tuner, kamerakuva muokkaamattomana.

8.1.3 LED-measurement -välilehti

Kun kuitukaapelissa oleva ledi on säädetty ja lukittu sekä kuitukuva tallennettu, valitaan seuraavaksi "LED-measurement" -välilehti (kuva 17).



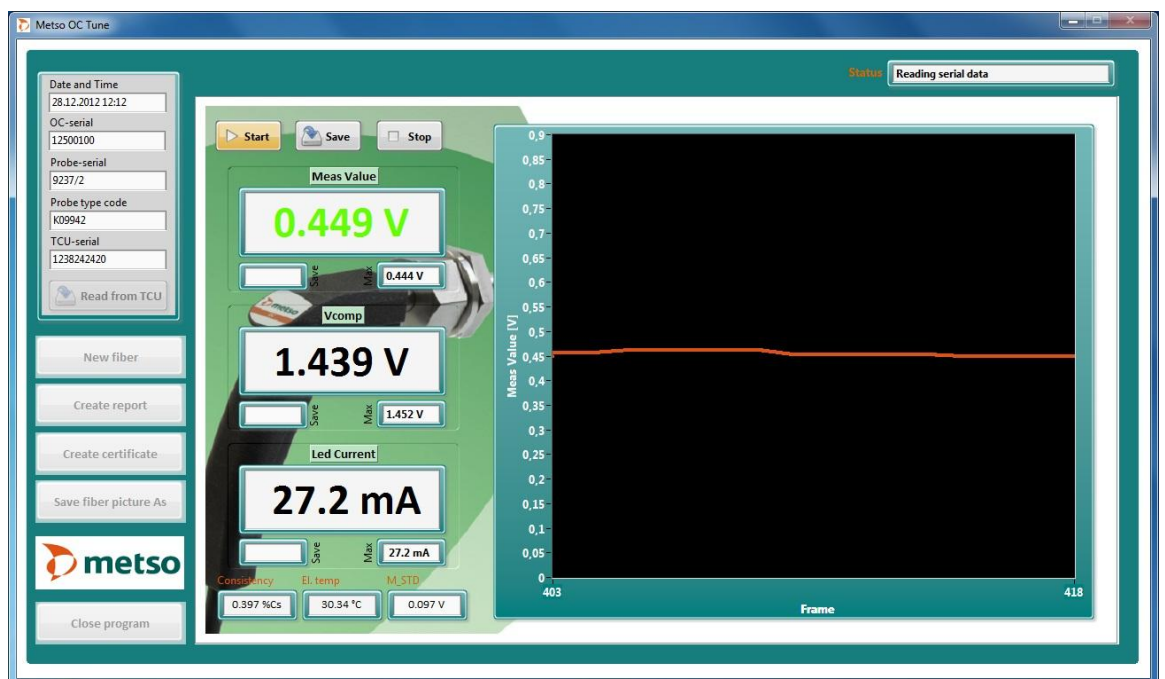
Kuva 17. Metso OC Tuner, mittausnäkyä numeroviitteillä.

Kuvaan 17 on eroteltu ohjelman toiminnan kannalta oleellisia kohtia seuraavasti:

1. "Start"-painikkeella käynnistetään sarjaliikenne, jolloin ohjelma alkaa lukea mittaus-tietoja TCU:lta. Tiedot luetaan n. 600 ms välein ja ne sijoitetaan oikeille paikoilleen (3). "Save"-painikkeella tallennetaan säädetty "Meas Value" ja muut sen hetkiset ar-vot raportointia varten. Tallennuksen jälkeen aukeaa käyttäjälle normalisointi-ikkuna. "Stop"-painike sulkee sarjaliikenteen.
2. Ohjelma näyttää aluksi "Meas Valuen" graafissa, josta käyttäjä näkee helposti mitat-tavien jännitteiden (Meas Value ja Vcomp) muutokset. Tallennuksen jälkeen kuvaa-jan asteikko muuttuu kompensointiarvolle sopivaksi ja näyttää "Vcomp"-arvoa.
3. Laitteelta saatavat mittaukset näkyvät niille varatuissa kentissä yksiköineen. "Save"-kenttiin tallentuu senhetkiset arvot, kun tiedot tallennetaan "Save"-painikkeella.

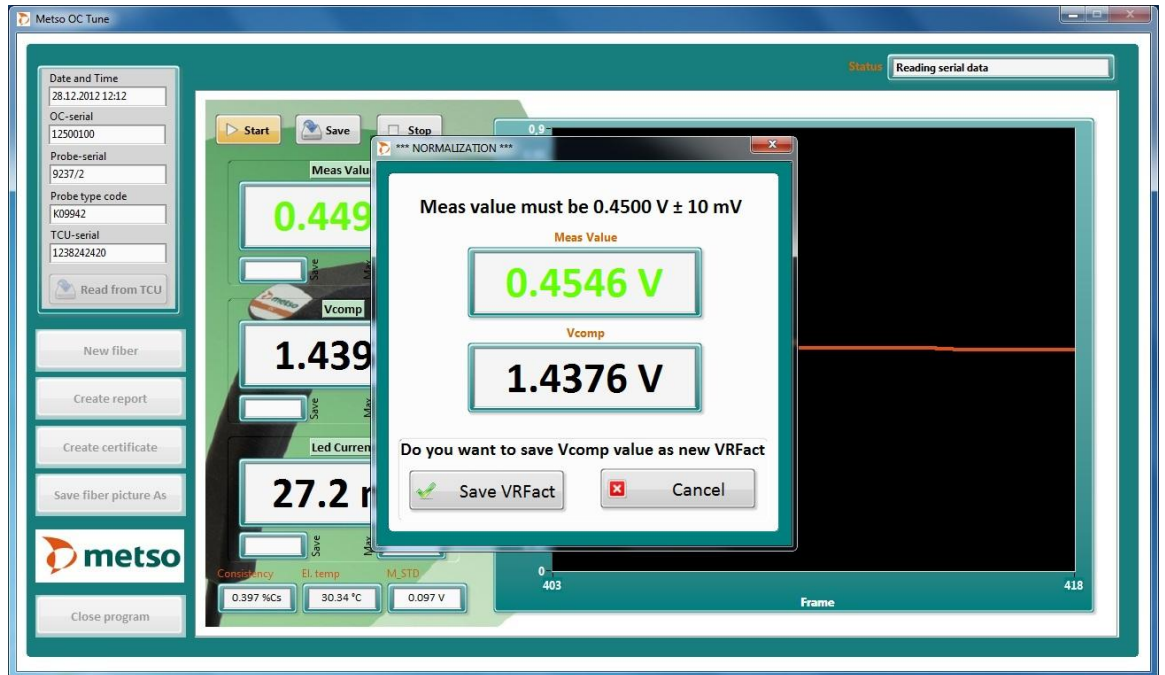
”Max”-kenttiin tallentuvat arvot aina, kun ”Vcomp” saavuttaa aikaisempaa suuremman arvon.

Kun ledin asento on säädetty kameran näyttämän kuvan perusteella optimaaliseksi, täytyy seuraavaksi säätää ledin intensiteetti oletusarvoon. Ennen säätöä mittakärki poistetaan kamerajigistä ja mittakärjen päähän asetetaan kalibrointilaatta. Tämän jälkeen voidaan aloittaa ledin säätäminen. Säädön aikana ohjelma lukee TCU:lta saatavia mittaustietoja ja näyttää ne käyttäjälle. Kuvassa 18 on ledin intensiteetti säädetty oikein ja meas-fontin väri on muuttunut vihreäksi. Graafissa näkyy ”Meas Value”-arvon muutokset säädön aikana.



Kuva 18. Metso OC Tuner, ”Meas Value” säädetty oletusarvon (0,450 V \pm 10 mV) mukaiseksi.

Kun säätö on suoritettu, tallennetaan säädettävät arvot raportointia varten ”Save”-painikkeella. Mittausarvot tallentuvat ”Save”-kenttiin. Tallentamisen jälkeen käyttäjälle avautuu normalisointi-ikkuna (kuva 19).



Kuva 19. Metso OC Tuner, normalisointi-ikkuna.

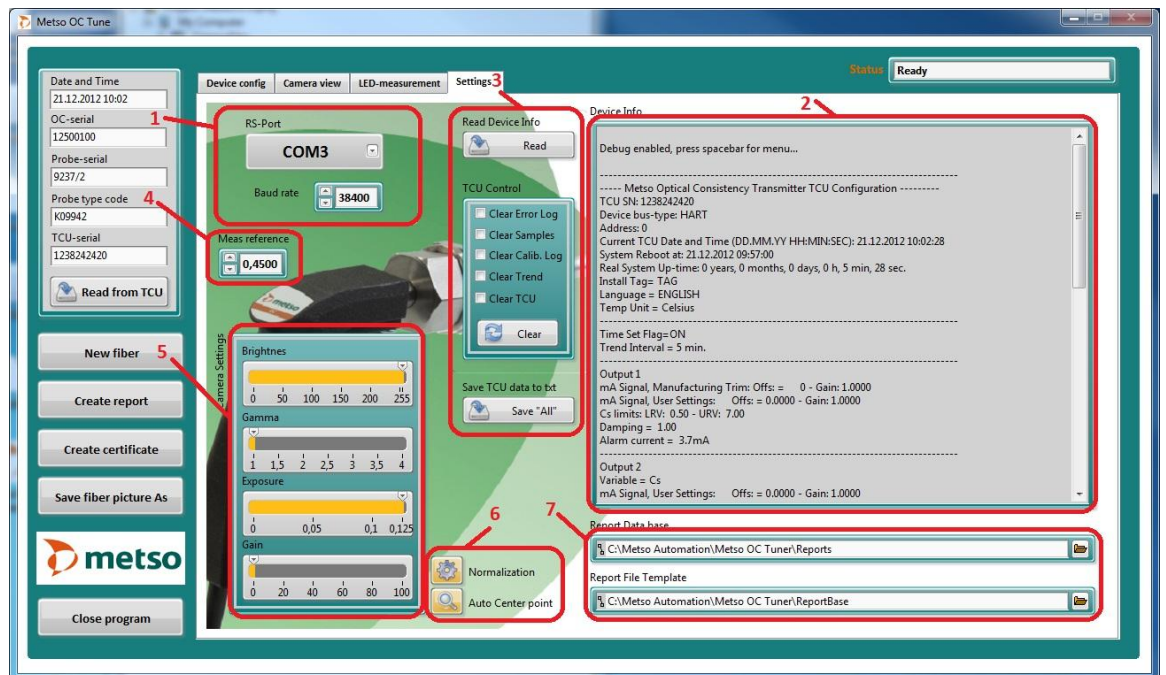
Normalisointi-ikkunassa varmistetaan vielä, että säätö on onnistunut ja "Meas Value" on oikea. Kun tiedot tallennetaan "Save VRFact" -painikkeella, lähettää ohjelma TCU:lle normalisointikomennon, jolloin TCU tallentaa senhetkisen kompensointiarvon uudeksi lämpötilakompensointikertoimeksi. Kun uusi arvo on tallennettu, ikkuna sulkeutuu.

Koska ledi on puolijohdekomponentti, sen sähköiset ominaisuudet muuttuvat lämpötilan muuttuessa. Kompensoinnilla pyritään pitämään ledin intensiteetti vakiona lämpötilasta riippumatta, jolloin mittauksiin ei tule virhettä lämpötilavaihteluiden seurauksena. Sen toiminta testataan suihkuttamalla lediin n. -40 °C olevaa kylmäsprayta, jolloin "Vcomp"-arvo alkaa kohota. Ohjelma tallettaa aina ledin hetkelliset mittausarvot, kun "Vcomp"-arvo saavuttaa aikaisempaa suuremman arvon. Maksimiarvot tallennetaan myöhemmin viritysraporttiin.

Kun ledi alkaa taas lämmetä, "Vcomp"-arvo rupeaa laskemaan ja testaus voidaan lopettaa "Stop"-painikkeella. Kun toiminto on loppunut, käyttäjän on mahdollista vaihtaa ohjelman näkymää.

8.1.4 Settings-välilehti

”Settings”-välilehdellä on korvattu vaatimusmäärityksissä ollut testi.ini-tiedosto. Välilehdellä käyttäjä voi visuaalisesti muuttaa asetuksia ja siihen on myös liitetty TCU:n ohjauksen kannalta oleellisia toimintoja. ”Settings”-välilehti esitetty kuvassa 20.



Kuva 20. Metso OC Tuner, asetukset numeroviitteillä.

Kuvaan 20 on eroteltu ohjelman kannalta oleellisia kohtia seuraavasti:

1. Sarjaportin valinta ja baudinopeuden muuttaminen.
2. ”Device Info”-ikkunassa näkyy kaikki TCU:lta tulevat sarjaliikenneviestit pois lukien ledin mittaukseen liittyvät viesti.
3. ”Read”-painikkeella voi lukea TCU:n tiedot ikkunaan(2). Se helpottaa oikean portin valintaa. ”TCU Control”-valinnoilla voi nollata TCU:ssa olevia tietoja. ”Clear”-painike lähettää komennot TCU:lle. ”Save All”-painike tallentaa laitteen kaikki tiedot ASCII-tekstitiedostoksi raporttikansioon.
4. ”Meas reference” määrittää oletusarvoisen referenssijännitteen, jonka mukaan meas-arvo vaihtaa väriään mittauksen aikana (vaihteluväli on ± 10 mV) ”Meas reference” -arvosta.

5. Kameran parametreja voi muuttaa liikuttamalla liukukontrolleja.
6. "Auto Center point"-valinta voidaan ottaa pois käytöstä, jos automatiikka ei tunnista kuidun keskipistettä. "Normalization"-valinta voidaan poistaa käytöstä, jolloin testauksen yhteydessä normalisointia ei suoriteta.
7. Ohjelman käyttämien polkujen vaihtaminen. "Report file Template" -kansiossa täytyy olla raporttien mallipohjat. Myös taulukko, johon testatut laitteet tallentuvat, sijaitsee tässä kansiossa. "Report Data base" -kansioon ohjelma tallentaa valmiit raportit ja sertifikaatit.

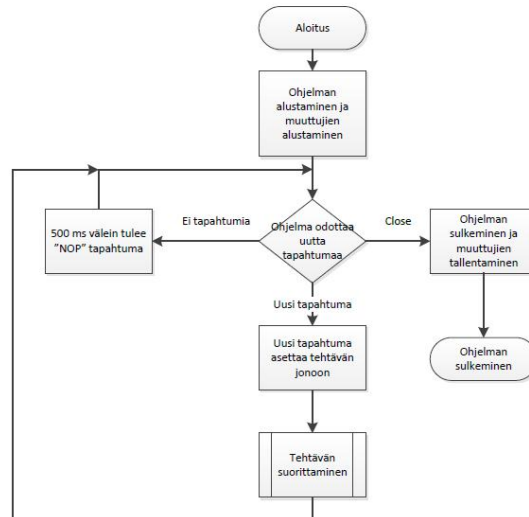
8.2 Ohjelman toimintojen tarkempi kuvaus ja toiminnalliset lohkokaaaviot

Ohjelmointi on sinänsä erikoinen ala, että jonkin toiminnon toteuttamiseen on yhtä paljon vaihtoehtoja kun on tekijöitäkin. Tämä ohjelmisto on toteutettu "Event"-rakenteella. Lähes jokainen uusi tapahtuma lisää toimintapyynnön jonoon, josta pyynnöt toteutetaan saadussa järjestyksessä. Näistä toiminnoista on piirretty kaaviot selventämään ohjelman suoritusta. Poikkeuksellisia tapahtumia, josta ei ole toiminnallista kaaviota, mutta jotka ovat kuitenkin "Event"-rakenteessa ovat sarjaportin valinta, kuvan tallentaminen ja "New fiber" -painike. Näissä tapahtumissa ei tarvitse erillistä jonotusosiota vähäisen toiminnan takia.

Seuraavassa käydään läpi yksityiskohtaisesti ohjelman suorittamat tapahtumat, joita käyttäjä tai ohjelma voi laukaista.

Aloitus, No Operation (NOP) ja lopetus

Kuvassa 21 on esitetty ohjelman toiminnallinen lohkokkaavio.



Kuva 21. Metso OC Tuner, pääohjelman toiminnallinen lohkokkaavio.

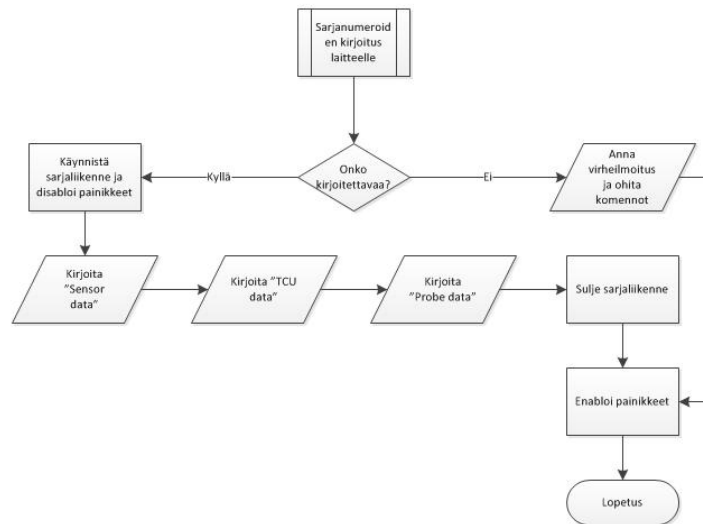
Kun ohjelma käynnistyy, ohjelma aloittaa muuttujien alustustoiminnot. Tässä vaiheessa nollataan kuvankäsittelyyn liittyvät toiminnot, luetaan aikaisempien, tärkeiden muuttujien tilat config.txt-tiedostosta ja asetetaan kytkimet luettujen muuttujien perusteella oikeaan tilaan. Sen jälkeen tarkistetaan käytettävissä olevat sarjaportit ja listataan ne valintakytkimelle. Ohjelmassa oleva kello ja päiväys asetetaan tietokoneen kalenterin ja kellon mukaan.

Alustusten jälkeen ohjelma on niin sanotussa lepotilassa, jossa se odottaa uutta tapahtumaa toteutettavaksi. Jos uusia tapahtumia ei tule, laukaisee ohjelma aina NOP-tilan 500 ms välein. NOP-tilassa laite kirjoittaa statuskenttään olevansa valmis uutta tapahtumaa varten. Ohjelma tarkkailee samalla, onko käyttäjä valinnut ohjelman sulkemiseen johtavan painikkeen. Jos käyttäjä on klikannut "Close program" -painiketta, ohjelma vielä varmistaa sulkemisen avaamalla ponnahdusikkunan, jossa käyttäjä vahvistaa sulkemisen. Jos käyttäjä päättää sulkea ohjelman, tallennetaan tärkeät muuttujat config.txt tiedostoon ja ohjelma suljetaan.

Ohjelma havahtuu uuteen tapahtumaan ja toteuttaa sen sisällön. Lähestulkoon kaikki tapahtumat laukaisevat jonkin suuremman tehtävän jonoon, joita näissä kaavioissa on kuvattu ali-ohjelmilla.

Sarjanumeroiden kirjoittaminen TCU:lle

Kuvassa 22 on esitetty tapahtuman toiminnallinen lohkokaavio.



Kuva 22. Metso OC Tuner, sarjanumeroiden kirjoituksen toiminnallinen lohkokaavio.

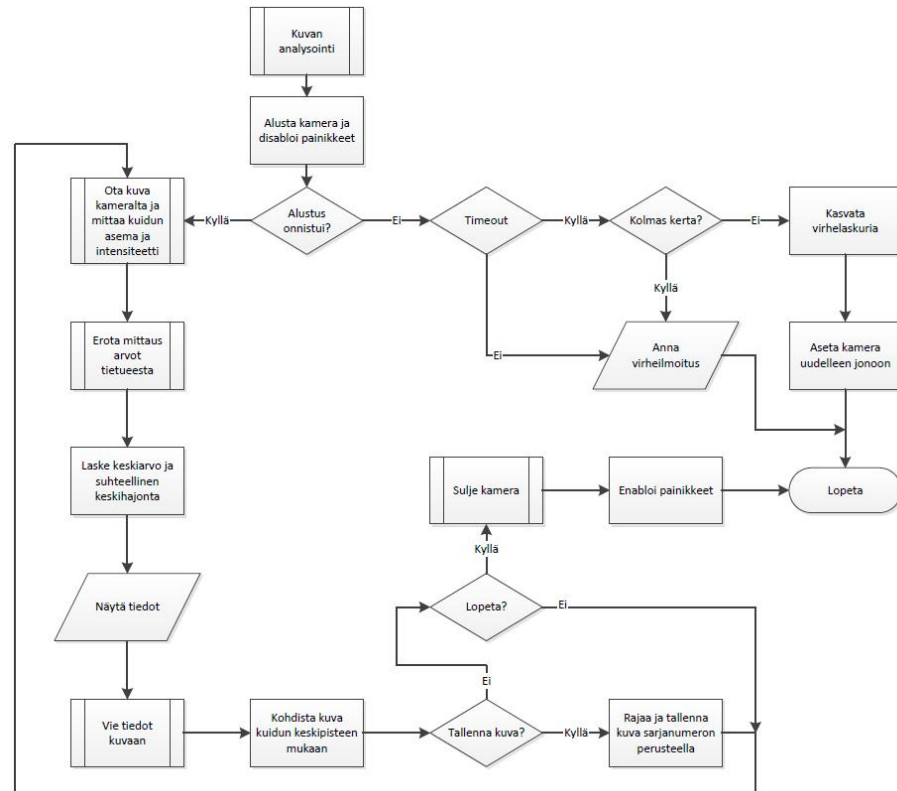
Sarjanumeroiden kirjoittaminen aktivoidaan klikkaamalla "Write Serials to TCU" -painiketta päänäkyvässä. Kun tapahtuma aktivoituu, ohjelma tarkistaa ensiksi, onko mitään kirjoitettavaa. Jos kaikki tunnistetietokentät ovat tyhjiä, ohjelma antaa siitä virheilmoituksen ja ohittaa varsinaiset tapahtuman toiminnot.

Jos tunnistetietokentässä on kirjoitettavia merkkejä, aloitetaan prosessi käynnistämällä sarjaliikenne ja disabloimalla kaikki painikkeet, joista voi aiheutua haittaa ohjelman toiminnalle. Sarjaliikenne käynnistetään kirjoittamalla TCU:lle debugkaus-komento, jonka jälkeen kirjoitetaan erikoiskomento, jolla TCU siirtyy kirjoitustilaan. Kun se on kirjoitustilassa, kirjoitetaan sinne kaikki tunnistetietokentät, joissa on merkkejä. Jos merkkejä ei ole, ohitetaan sen tunnistetiedon kirjoitus. Jokaisesta kirjoitetusta merkkijonosta tulee vastauksena sama merkkijono, jos kirjoitus on mennyt oikein. Onnistunut kirjoitus näkyy käyttäjälle vihreänä merkkivälönä tunnistetietokentän vierellä. Vastaavasti epäonnistunut kirjoitus näkyy käyttäjälle punaisena kuvakkeena.

Kun merkkijonot on kirjoitettu TCU:lle, kirjoitetaan vielä komento, jolla poistutaan kirjoitustilasta. Kun laite on kirjoitustilassa, toimii siinä vain tietyt komennot, joten siitä poistuminen on välttämätöntä. Kun TCU on normaalitilassa, enableoidaan kaikki painikkeet, ja ohjelma palaa takaisin lepotilaan.

Kameran käynnistys ja kuvan analysointi

Kuvassa 23 on esitetty tapahtuman toiminnallinen lohkokaavio.



Kuva 23. Metso OC Tuner, kameraosion toiminnallinen lohkokaaavio.

Kamera ja sen toiminnot käynnistetään "Camera view" -välilehdellä olevasta "Start"-painikkeesta. Tapahtuma disabloi kaikki painikkeet, joista voi aiheutua haittaa ohjelman toiminnalle. Seuraavaksi alustetaan kamera ja valmistellaan kuvankäsittelytoiminnot. Kameran alustuksessa on ollut jonkin verran ongelmia, joten siihen on tehty automaattinen uusinta, jos kamera antaa "timeout"-virheen. Ohjelma yrittää kolmesti käynnistää kameran korottaen aina virhelaskuriaan. Jos kamera ei käynnisty kolmannellakaan yrittämällä, ohjelma antaa käyttäjälle virheilmoituksen ja palaa lepotilaan.

Jos kamera käynnistyy oikein, nollataan virhelaskuri ja aloitetaan kuvan prosessointi. Aluksi kuvasta tutkitaan kuitukimpun keskipiste, jonka perusteella kuva keskitetään käyttäjälle näkyvään ikkunaan. Keskipiste keskiarvoistetaan, jotta vältetään kohinan aiheuttamalta kuvan tärinältä. Kun kuitukaapelin keskipiste on määritetty, siirretään kuitujen oletusarvoisten keskipisteiden asemat korjauskertoimien avulla oikeille paikoille. Näin toimittaessa ohjelma löytää

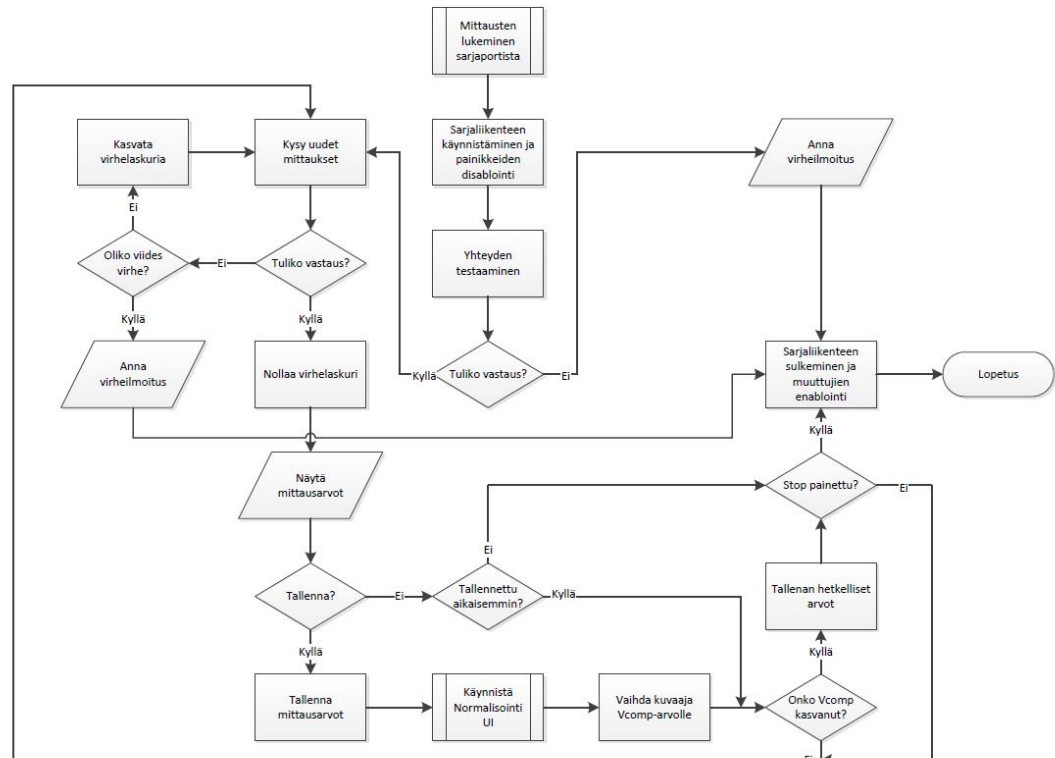
kuitujen keskipisteet, vaikka kaapelin asema suhteessa kamerakennoon muuttuisikin. Kun ohjelma on löytänyt kaikki kuidut, se mittaa niiden intensiteettiarvot ja määrittää niiden aseman.

Seuraavaksi kuvaan lisätään intensiteettiarvot oikeiden kuitujen kohdille, että ne näkyvät käyttäjälle kuvassa ja tulostuvat siten myös raporttiin. Mitatuista kuuden kuidun arvoista lasketaan keskiarvo ja suhteellinen keskihajonta, jotka näytetään käyttäjälle mittarilla ja numeronäytöillä.

Kun käyttäjä on säätänyt kaapelissa olevan ledin aseman kuvan ja mittarin perusteella, tallennetaan kuva klikkaamalla "Save"-painiketta. Kuva tallentuu kuidun sarjanumeron perusteella oletusraportointikansioon, josta se haetaan raportointivaiheessa. Kamera suljetaan klikkaamalla "Stop"-painiketta. Ohjelma sulkee kameran, poistaa kuvan ja enablei taas painikkeet, jotta käyttäjä voi siirtyä seuraavaan vaiheeseen.

Ledi-mittaukset, sen virittäminen ja normalisointi

Kuvassa 24 on esitetty tapahtuman toiminnallinen lohkokaavio.



Kuva 24. Metso OC Tuner, ledi-mittausten toiminnallinen lohkokaavio.

Viritystapahtuma käynnistetään "LED-measurement" -välilehdellä olevasta painikkeesta. Aluksi ohjelma käynnistää sarjaliikenteen, disabloi kaikki painikkeet, joista voi aiheutua haittaa ohjelman toiminnalle ja lähettää TCU:lle debuggaus-komennon. Seuraavaksi ohjelma odottaa TCU:n vastausta sarjaliikenteeseen. Jos vastausta ei tule määräaikaan mennessä, ohjelma antaa virheilmoituksen, jonka jälkeen sarjaliikenne suljetaan ja painikkeet enabloidaan.

Jos TCU:n vastaus on oikeanlainen, ohjelma siirtyy osioon, jossa se kysyy uudet mittaukset TCU:lta n. 600 ms välein. TCU lähettää mittausviestin joka kerta, kun ohjelma lähettää kyselyn. TCU:lta saadusta viestistä erotetaan ledin "Meas Value" -arvo, "VComp"-arvo, ledin virta, sakeus ja lämpötila. Tiedot tuodaan käyttäjälle näkyviin kenttiin ja alkutilanteessa ledin "Meas Value" -arvo piirretään myös kuvaajaan, josta sen muutoksia on helpompi seurata. Mittauskyselyprosessiin on lisätty yhteyden katkeamisen varalle laskuri, joka pyrkii saamaan

oikeanlaista viestiä viisi kertaa, ennen kuin se antaa virheilmoituksen. Sarjakaapelin voi siis hetkellisesti irrottaa ja laittaa takaisin, eikä ohjelman toiminnot häiriinny siitä.

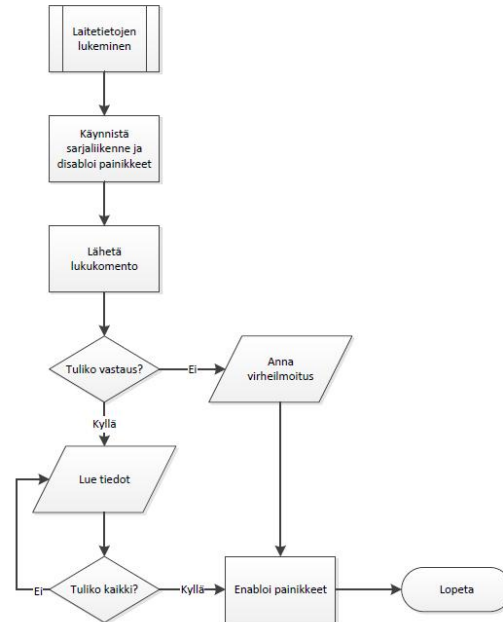
Kun ledin "meas"-arvo on viritetty 0,450 V:iin, tallennetaan sen hetkiset mittaustulokset "Save"-painikkeella muistiin. Painallus tuo ruudulle uuden ikkunan, joka mahdollistaa ledin korjauskertoimen tallentamisen TCU:lle. Korjauskerroin pyrkii pitämään ledin meas-arvon stabiilina lämpötilamuutoksissa. Ohjelma näyttää vielä normalisointi-ikkunassa mittausarvot tarkastamista ja mahdollista säätöä varten. Kun käyttäjä on hyväksynyt uuden kompensointikertoimen, lähettää ohjelma tallennuskomennon TCU:lle ja ikkuna sulkeutuu. Avautuva ikkuna on sen tyyppinen, että käyttäjän on pakko tallentaa kompensointiarvo "Save VRfact" -painikkeella, tai hylätä arvon muuttaminen painamalla "Cancel"-painiketta. Kun ikkuna on avoinna, käyttäjä ei voi tehdä ohjelmassa muita toimintoja.

Normalisoinnin jälkeen ohjelma vaihtaa graafin näyttämään TCU:lta saatavaa kompensointiarvoa "Vcomp". Nyt ohjelma tallentaa mittausarvot aina, kun kompensointiarvo on saavuttanut suurimman arvonsa. Seuraavaksi käyttäjä jäädyttää lediä suihkuttamalla siihen kylmäsprayta noin viiden sekunnin ajan. Kun led jäähtyy voimakkaasti, lähtee kompensointiarvo kasvamaan normalisoinnissa määritellyn kertoimen perusteella. Kompensointiarvo saavuttaa huippunsa noin minuutin päästä jäähdytyksestä. Ohjelma tallentaa saadut mittausarvot raportointia varten.

Ledin viritys lopetetaan painamalla "STOP"-painiketta, jolloin ohjelma sulkee sarjaliikenteen ja enablei taas painikkeet, jotta käyttäjä voi siirtyä seuraavaan vaiheeseen.

Laitetietojen lukeminen

Kuvassa 25 on esitetty tapahtuman toiminnallinen lohkokaavio.



Kuva 25. Metso OC Tuner, laitetietojen lukemisen toiminnallinen lohkokaavio.

Laitetietojen lukemista käytetään pääsääntöisesti yhteyden tarkistamiseen, kun määritellään oikeaa sarjaporttia.

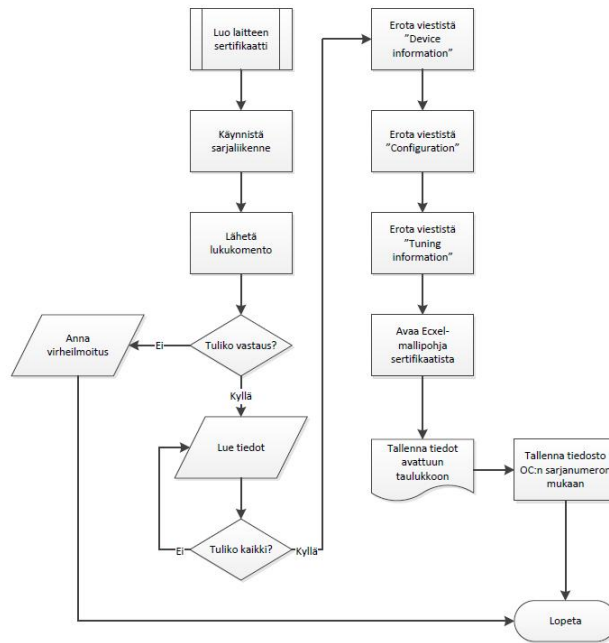
Lukeminen aloitetaan painamalla "Settings"-välilehdellä "Read"-painiketta. Aluksi ohjelma käynnistää sarjaliikenteen, disabloi kaikki painikkeet, joista voi aiheutua haittaa ohjelman toiminnalle ja lähettää TCU:lle debugkaus-komennon. Seuraavaksi ohjelma odottaa TCU:n vastausta sarjaliikenteeseen. Jos vastausta ei tule määräaikaan mennessä, näyttää ohjelma virheen välilehdellä olevassa ikkunassa, jonka jälkeen sarjaliikenne suljetaan ja painikkeet enableidaan.

Jos porttivalinta on oikea, alkaa TCU komennon saatuaan lähettää laitetietoja sarjaliikenteenä. Ohjelmisto ei erittele saatavista tiedoista mitään, vaan tulostaa ne välilehdellä olevaan "Device info" -ikkunaan.

Kun lähetys loppuu ja sarjaportti on tyhjä, enablei ohjelma taas kaikki painikkeet ja käyttäjä voi suorittaa seuraavan vaiheen.

Sertifikaatin luominen

Kuvassa 26 on esitetty tapahtuman toiminnallinen lohkokaavio.



Kuva 26. Metso OC Tuner, sertifikaatin luomisen toiminnallinen lohkokaavio.

Laitteen yksilöllinen sertifikaatti tallennetaan tietokoneelle virityksen loppuvaiheessa.

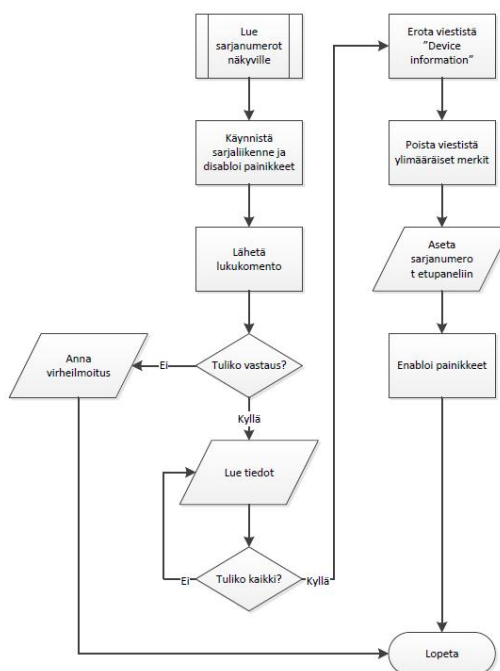
Sertifikaatin luominen aloitetaan painamalla "Create certificate" -painiketta, kun mittaukset on suoritettu. Aluksi ohjelma käynnistää sarjaliikenteen ja lähettää TCU:lle debuggus-komennon. Seuraavaksi ohjelma odottaa TCU:n vastausta sarjaliikenteeseen. Jos vastausta ei tule määräaikaan mennessä, antaa ohjelma virheilmoituksen, jonka jälkeen sarjaliikenne suljetaan.

Jos debuggus-komentoon tulee vastaus, lähetetään TCU:lle sertifikaatin tulostuskomento. Kun TCU on vastaanottanut viestin, lähettää se sertifikaattitiedot ohjelmalle. Kun lähetys loppuu ja sarjaportti on tyhjä, ohjelma alkaa erotella saadusta viestistä tallennettavaan sertifikaattiin oleelliset tiedot. Eroteltavia tietoja ovat laitteen tunnistetiedot, laitteen mittaussasetukset ja laitteen viritystiedot.

Kun ohjelma on erottanut tiedot, avaa se Excel-mallipohjan sertifikaatista ja siirtää tiedot siihen. Ohjelma tallentaa sertifikaatin OC:n sarjanumeron perusteella oletusraportointikansioon. Tallennuksen jälkeen ohjelma palaa normaaliin tilaansa.

Laitetietojen lukeminen raporttiin

Kuvassa 27 on esitetty tapahtuman toiminnallinen lohkokaavio.



Kuva 27. Metso OC Tuner, raportointia varten tapahtuvan laitetietojen lukemisen toiminnallinen lohkokaavio.

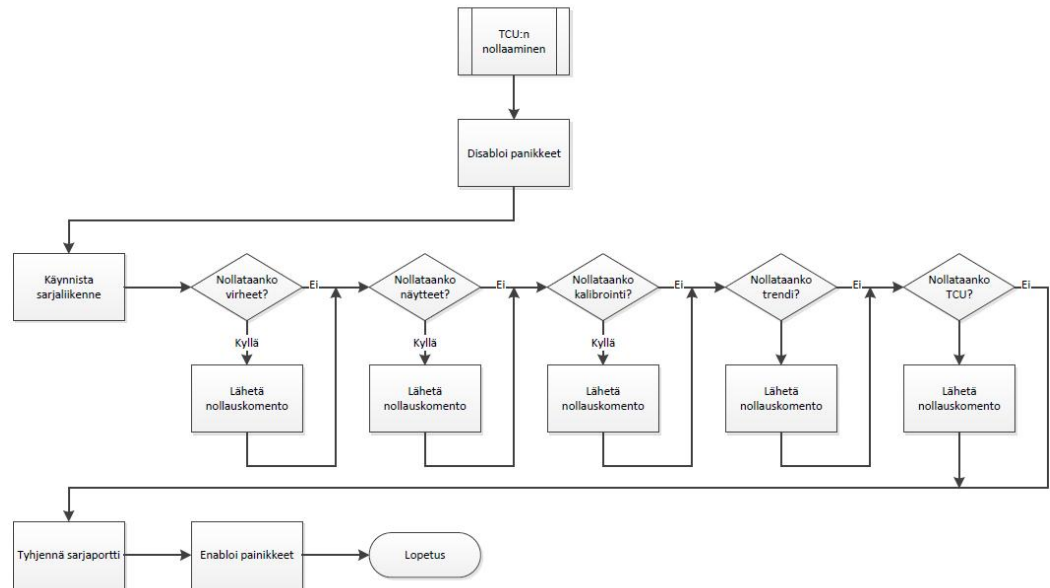
Raportointia varten tapahtuva laitetietojen lukeminen tehdään jo sarjanumeroiden kirjoituksen jälkeen. Samalla voidaan varmistaa, että tiedot ovat menneet laitteelle oikein ja kuvien, raportin ja sertifikaatin nimet menevät oikein.

Lukeminen aloitetaan painamalla "Read from TCU" -painiketta. Ohjelma käynnistää sarjaliikenteen, disabloi kaikki painikkeet, joista voi aiheutua haittaa ohjelman toiminnalle ja lähettää TCU:lle debuggaus-komennon. Seuraavaksi ohjelma odottaa TCU:n vastausta sarjaliikenteeseen. Jos vastausta ei tule määräaikaan mennessä, antaa ohjelma virheilmoituksen, jonka jälkeen sarjaliikenne suljetaan ja painikkeet enabloidaan.

Jos debuggaus-komentoon tulee vastaus, lähetetään TCU:lle sertifikaatin tulostuskomento. Kun TCU on vastaanottanut viestin, lähettää se sertifikaattitiedot ohjelmalle. Kun lähetys on loppunut ja sarjaportti on tyhjä, erottelee ohjelma raportin kannalta oleelliset tiedot ja tuo ne käyttäjän nähtäville viritettävän laitteen tietoihin. Tämän jälkeen painikkeet taas enabloidaan ja voidaan siirtyä seuraavaan vaiheeseen.

TCU:n tietojen nollaaminen

Kuvassa 28 on esitetty tapahtuman toiminnallinen lohkokkaavio.



Kuva 28. Metso OC Tuner, TCU:n tietojen nollaamisen toiminnallinen lohkokkaavio.

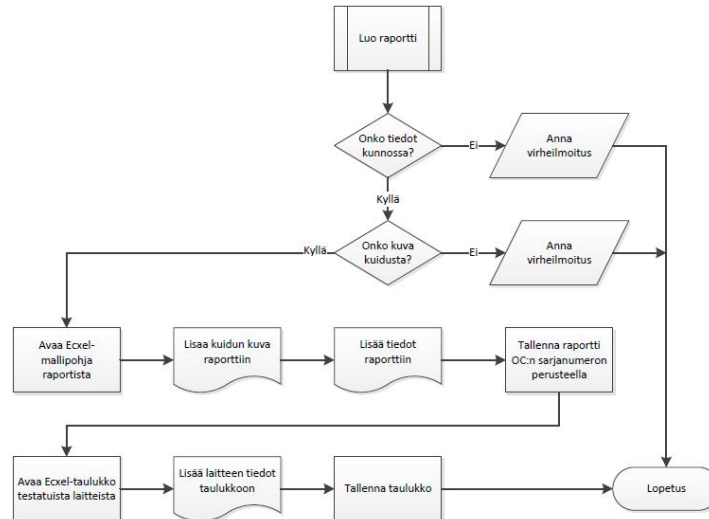
Ohjelman avulla voi myös tyhjentää sakeusmittarissa olevia tietoja. Tyhjennysvaihtoehdot ovat "Settings"-välilehdellä.

Ennen toiminnon aloittamista käyttäjä voi valita tyhjennettävät tiedot painamalla valintaruudun aktiiviseksi. Kun halutut toiminnot on valittu, aloitetaan tyhjennys "Clear"-painikkeella. Ohjelma disabloi kaikki painikkeet, joista voi aiheutua haittaa ohjelman toiminnalle ja käynnistää debuggauksen TCU:lla.

Seuraavaksi ohjelma lähettää jokaisen valittuna olleen toiminnon tyhjennyskomennon TCU:lle. Lähetysten jälkeen ohjelma ottaa vastaan kuittaukset tyhjennetyistä tiedoista ja enablei painikkeet.

Raportin luominen

Kuvassa 29 on esitetty tapahtuman toiminnallinen lohkokaavio.



Kuva 29. Metso OC Tuner, raportoinnin toiminnallinen lohkokaavio.

Jokaisesta testatusta laitteesta tallennetaan raportti, johon liitetään tiedot suoritetusta virittämisestä kuvineen, ledin säädetyt arvot ja kylmähokkiarvot.

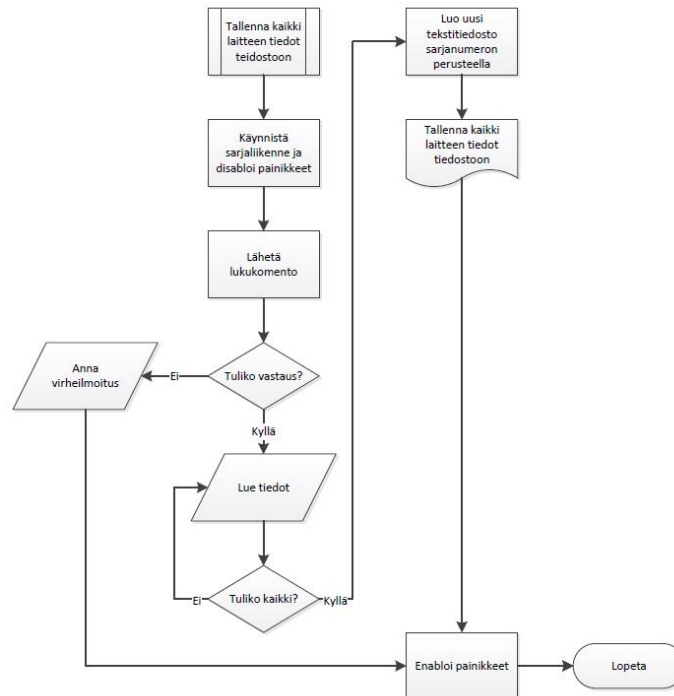
Raportoiminen aloitetaan painamalla "Create report" -painiketta. Aluksi ohjelma tarkistaa, että tarpeelliset tiedot on kirjoitettu tai ne on luettu laitteelta. Jos jokin tieto puuttuu, antaa ohjelma virheilmoituksen ja raportointi keskeytyy.

Seuraavaksi ohjelma tarkistaa, onko kyseiselle kuidun sarjanumerolle olemassa virityskuva. Jos sarjanumeron mukaista kuvaa ei löydy, antaa ohjelma virheilmoituksen ja raportointi keskeytyy.

Jos tiedot ovat oikein ja kuva löytyy, avaa ohjelma raporttipohjan Exceliin ja siirtää tiedot siihen. Kun tiedot on siirretty, tallentaa ohjelma raportin OC:n sarjanumeron perusteella oletusraportointikansioon. Tallentamisen jälkeen ohjelma avaa taulukon, johon kerätään kaikki raportoidut laitteet viritystietoineen. Ohjelma etsii avatusta taulukosta seuraavan vapaan rivin, siirtää laitteen tiedot taulukkoon ja tallentaa taulukon.

Kaikkien laitetietojen tallentaminen

Kuvassa 30 on esitetty tapahtuman toiminnallinen lohko-kaavio.



Kuva 30. Metso OC Tuner, kaikkien TCU tietojen tallentamisen toiminnallinen lohko-kaavio.

TCU:lla on myös komento, joka lähettää kaikki laitteen tiedot sarjaportille. Komennon avulla voidaan tallentaa sarjanumeroiden lisäksi laitteeseen tallennetut reseptit, näytteet, trendit ja virheilmoitukset.

Toiminto käynnistetään "Settings"-välilehdellä olevalla "Save All" -painikkeella. Ohjelma käynnistää sarjaliikenteen, disabloi kaikki painikkeet, joista voi aiheutua haittaa ohjelman toiminnalle ja lähettää TCU:lle debuggaus-komennon. Seuraavaksi ohjelma odottaa TCU:n vastausta sarjaliikenteeseen. Jos vastausta ei tule määräaikaan mennessä, antaa ohjelma virheilmoituksen, jonka jälkeen sarjaliikenne suljetaan ja painikkeet enabloidaan.

Jos yhteys on kunnossa, lähettää TCU kaikki sen tiedot sarjaporttiin. Viesti luetaan loppuun, jonka jälkeen tiedot tallennetaan raporttikansioon sarjanumeron perusteella ASCII-tekstitiedostoon. Kun tiedosto on tallennettu, enabloidaan kaikki painikkeet ja ohjelma siirtyy valmiustilaan.

8.3 Ohjelman testaaminen

LabVIEW-kieli osaa se ilmoittaa välittömästi, jos koodi ei ole ajettavissa. Tämän toiminnon perusteella ohjelmasta on helppo korjata yksinkertaiset ohjelman syntaksivirheet.

LabVIEW ei kuitenkaan osaa korjata toiminnallisia virheitä. Kuitenkin koodia voi testata heti, kun siihen tekee muutoksia tai lisäyksiä. Siinä on mahdollista testata yksittäisiä osioita tai suurempia kokonaisuuksia myös debuggauksella. Sen aikana ohjelma näyttää jokaisen yksittäisen ohjelmassa olevan tietovirran-arvon. Sen avulla ohjelmasta näkee välittömästi yksittäisten muuttujien arvot ja ohjelmaa on mahdollista ajaa vaihe kerrallaan.

Debuggauksen lisäksi ohjelmassa on mahdollista lisätä probeja, eli yksittäisiä tarkistuspisteitä ympäri koodia vapaasti valittaviin paikkoihin. Probe-ikkunasta näkee reaaliajassa tarkastuspisteessä olevan arvon. Toiminto on todella käytännöllinen esimerkiksi sarjaliikenteessä kulkevien viestien tarkastamiseen.

Ohjelmistoa testattiin siis jatkuvasti, kun sitä kehitettiin. Aina kun ohjelmaan rakennettiin uusi osio, se testattiin välittömästi. Jos lisätty toiminto toimi halutulla tavalla, siirryttiin ohjelmassa seuraavan vaiheen toteuttamiseen. Jokainen ohjelman osio on aina testattu huolellisesti, ennen kuin seuraavaan vaiheeseen on siirrytty. Myös eri osioiden yhteensopivuutta testattiin eri vaiheissa kehitystä.

LabVIEW-ohjelmistossa on lisätoimintona VI-analyzer, jonka avulla ohjelman toimintoja ja koodia testattiin järjestelmällisesti. Analyysin perusteella ohjelman koodiin tehtiin tarpeellisia muutoksia ja siitä poistettiin turhia osioita. Analyysi tutkii koodin rakennetta erilaisten testien perusteella ja ilmoittaa havaitsemansa virheet testauksen jälkeen. Analysoinnin perusteella ohjelmassa ei ilmennyt virheitä ja kaikki sen yksittäiset aliohjelmat avautuvat normaalisti.

Ohjelmistoa testattiin huolellisesti ennen kuin se asennettiin tuotantoon pilottijaksoa varten. Se aloitettiin noin 2 kk työn aloittamisesta. Tuotantotestauksissa on selvinnyt jotain korjaustarpeita ohjelmasta, mutta periaatteellinen toiminta oli heti alussa halutulla tasolla.

9 TYÖN TULOKSET

Työn lopputuloksena on syntynyt ohjelmisto, joka vastaa työn tilaajan tarpeita erinomaisesti. Vaatimusmääritelmän mukainen ohjelma oli jo valmiina aikaisemmin, mutta 9.11.2012 käydyssä palaverissa ohjelmistoon lisättiin tarpeelliseksi katsottuja toimintoja. Ohjelmaan lisättiin laitetietojen kirjoittamismahdollisuus nopeuttamaan testausprosessia. Aikaisemmin testauksessa laitteen tiedot on syötetty TCU:ssa olevilla näppäimillä.

Työssä luotu ohjelmisto varmistaa, että asiakas saa luotettavasti säädetyin ja kalibroidun Metso OC -sakeusmittarin ja jokaisesta lähetetystä laitteesta jää Metsolle kattava dokumentointi laitteen sarjanumeroista ja viritysarvoista. Metso OC Tuner -ohjelmisto on otettu käyttöön tammikuussa 2013.

Ohjelmiston ja insinööriyön kirjallisen osion lisäksi ohjelmiston suunnitteluun ja valmistukseen on liittynyt varsinainen ohjelmistokuvaus Metsolle (40 sivua), asennusohjeen laatiminen (18 sivua) ja ohjelman käyttöohjeen laatiminen (22 sivua).

Työn laajuutta on hankala tuoda esille työn kirjalliseen osioon. LabVIEW:ssa on mahdollista katsoa koko ohjelman hierarkia (rakenne) ja valmis ohjelmisto koostuu yli 500 erillisestä osasta. Toki enemmistö aliohjelmista on valmiita funktioita LabVIEW:ssä.

Toistaiseksi ohjelmisto tukee vain yhtä kameratyyppiä/mallia, joten tarvittaessa ohjelmaa voisi laajentaa siten, että se osaisi käyttää muitakin kameroita. Tätä ei kuitenkaan ollut laitettu vaatimusmäärittelyyn, ja se olisi lisännyt työn kuormittavuutta. Pelkän kameran tunnistamisen lisäksi täytyisi kuidusta tehtävät mittaukset skaalata uudelleen, jos kennon koko muuttuu. Kuidun mittaukset suoritetaan kuitenkin absoluuttikoordinaateilla. Tilanne on sama jos käytettävää optiikkaa muutetaan, ei kuva enää näy samankokoisena kameran kennolla.

Toinen kehittämiskohta on lisätä nykyiseen järjestelmään objektiivivalo tai jokin muu valonlähde, jonka avulla voitaisiin valaista mittakärjen pää. Tällainen mahdollisuus helpottaisi huollettavien laitteiden tutkimista, koska sovellus toimisi mikroskoopin tavoin, eikä varsinaista mikroskooppia tarvittaisi. Kuvattavan kohteen valaisu on hankalaa perinteisellä lampulla, koska linssi on lähellä kuvattavaa kohdetta. Valonlähteen tulisi olla pieni ja sen pitäisi valaista kohde tasaisesti.

10 YHTEENVETO

Teollisuuden kehittyminen on ollut huimaa viimeisen vuosisadan aikana. Erityisesti tietotekniikan osuus on kasvanut merkittävästi teollisuuden eri aloilla. Suomi on ollut kansainvälisesti kehityksen kärjessä jo pitkään, ja innovatiivisilla tulevaisuuden ratkaisuilla voimme taata tämän kehityksen jatkuvuuden.

Testausohjelmat ovat yleensä laitekohtaisia, ja siksi kullekin laitteelle on oltava olemassa oma testausympäristö. Työssä syntyneen ohjelmiston modulaarisuus mahdollistaa ohjelman käytettävyyden pienillä muutoksilla samankaltaisille mittalaitteille. Insinööriyössä luotuu ohjelmistoon on jo lisätty uusi toiminnallisuus, jonka avulla on mahdollista kirjoittaa myös Metso LC:n tunnistetiedot mittalaitteelle.

Insinööriyö on ollut kokonaisvaltainen prosessi, jonka aikana on perehdytty erilaisiin sateidenmittaustekniikoihin ja sovelluskohteisiin. Työssä pääpaino on ollut luotettavan testausjärjestelmän suunnittelussa ja käytäntöön panemisessa. Ohjelmiston suunnittelussa on pyritty ottamaan huomioon ohjelman helppokäyttöisyys, testausten toistettavuus ja tulosten helppo tarkastelu sekä kattava raportointi Metso Oy:lle.

Työssä suunniteltu testausjärjestelmä takaa nopean ja luotettavan testausmenetelmän uusien Metso OC -sateusmittareiden viritykseen ennen tuotteen toimitusta asiakkaalle. Työn lopputulos täyttää hyvin sille asetetut tavoitteet ja vaatimukset.

LÄHTEET

1. Metso yrityksenä, Metso lyhyesti, [WWW-dokumentti]
<http://www.metso.com/fi/corporation/about_fin.nsf/WebWID/WTB-041026-2256F-55957?OpenDocument> (Luettu 1.12.2012)
2. Tervetuloa taloon opas 2012 PAS, Sisäinen materiaali, [PDF-dokumentti]
3. Klemetti Ursula - Kortelainen Veli-Antti - Lyytikäinen Jorma - Siitonen Heikki - Siironen Raimo, 1999, Paperimassan valmistus, ISBN 952-13-0604-1.
4. Suomen metsäteollisuuden historia tiivistetysti, [WWW-dokumentti]
<<http://www.metsateollisuus.fi/Infokortit/historialyhyesti/Sivut/default.aspx>>
(Luettu 1.12.2012)
5. Häggblom-Ahnger Ulla - Komulainen Pekka, 2001, Paperin ja kartongin valmistus, ISBN 952-13-1280-7.
6. Aumala Olli, Teollisuusprosessin mittaukset, 4. painos 1998, ISBN 952-9835-31-0.
7. Pikala Juhani, Prosessisuureiden mittaustekniikka, 2-1. painos 2008, ISBN 978-952-13-2219-8.
8. Metso Product training 2010, Sisäinen materiaali, [PowerPoint-esitys]
9. Metso Performance Solutions for Continuous cooking, analyzers, economic benefits, Sisäinen materiaali, [PowerPoint-esitys]
10. Metso Microwave Consistency Transmitter, Sisäinen materiaali, [PowerPoint-esitys]
11. Tuomaranta Matti, 2006, Diplomityö, Sellun ilmapitoisuuden määrittäminen mikroaaltomittalaitteen signaaleista, Oulun yliopisto
12. Consistency Management Presentation, Sisäinen materiaali, [PowerPoint-esitys]
13. It's Time to Upgrade Your Old Kajaani RM-200 to Metso RM3, Sisäinen materiaali, [PowerPoint-esitys]
14. Kämäräinen Joni, 1999, Diplomityö, Teräksen vakuumikäsittelyn visuaalinen laadunarviointi, Lappeenrannan teknillinen korkeakoulu
15. Romppainen Pentti, Konenäön opetusmateriaali, Kajaanin ammattikorkeakoulu
16. Rinne Olli, Digikameroiden kennotyyppien eroja, [WWW-dokumentti]
<http://digifaq.info/digi_omat/kennot.html> (Luettu 12.1.2013)

17. Rinne Olli, Optiikka valokuvaajalle, [WWW-dokumentti]
<http://digifaq.info/digi_omat/optiikka> (Luettu 12.1.1013)
18. Harsu Maarit. Ohjelmointikielet: periaatteet, käsitteet ja valintaperusteet, Tammer-Paino Oy, 2005. ISBN 952-14-0981-9.
19. Graphical Programming, The Benefits of Programming Graphically in NI LabVIEW, [PDF-dokumentti]
<ftp://ftp.ni.com/evaluation/labview/whatislv_kit.zip>
20. Configurable Machine Vision Software , NI vision Builder for Automated Inspection, [PDF-dokumentti]
<http://www.ni.com/pdf/products/us/vbai_datasheet.pdf>
21. NI Vision Builder for Automated Inspection Tutorial, [PDF-dokumentti]
<<http://www.ni.com/pdf/manuals/373379h.pdf>>
22. Industrial Cameras for digital imaging and visualization, [WWW-dokumentti]
<http://www.ids-imaging.com/frontend/products.php?cam_id=159> (Luettu 1.12.2012)
23. Paakki Jukka, Ohjelmistojen vaatimusmäärittely, Helsingin yliopisto Tietojenkäsittelytieteen laitos, [PDF-dokumentti] <<http://www.cs.helsinki.fi/u/paakki/Vaatimus-11-Luentokalvot-1.pdf>>

LIITTEET

LIITE 1, Esimerkkituloste ohjelman tallentamasta raportista.

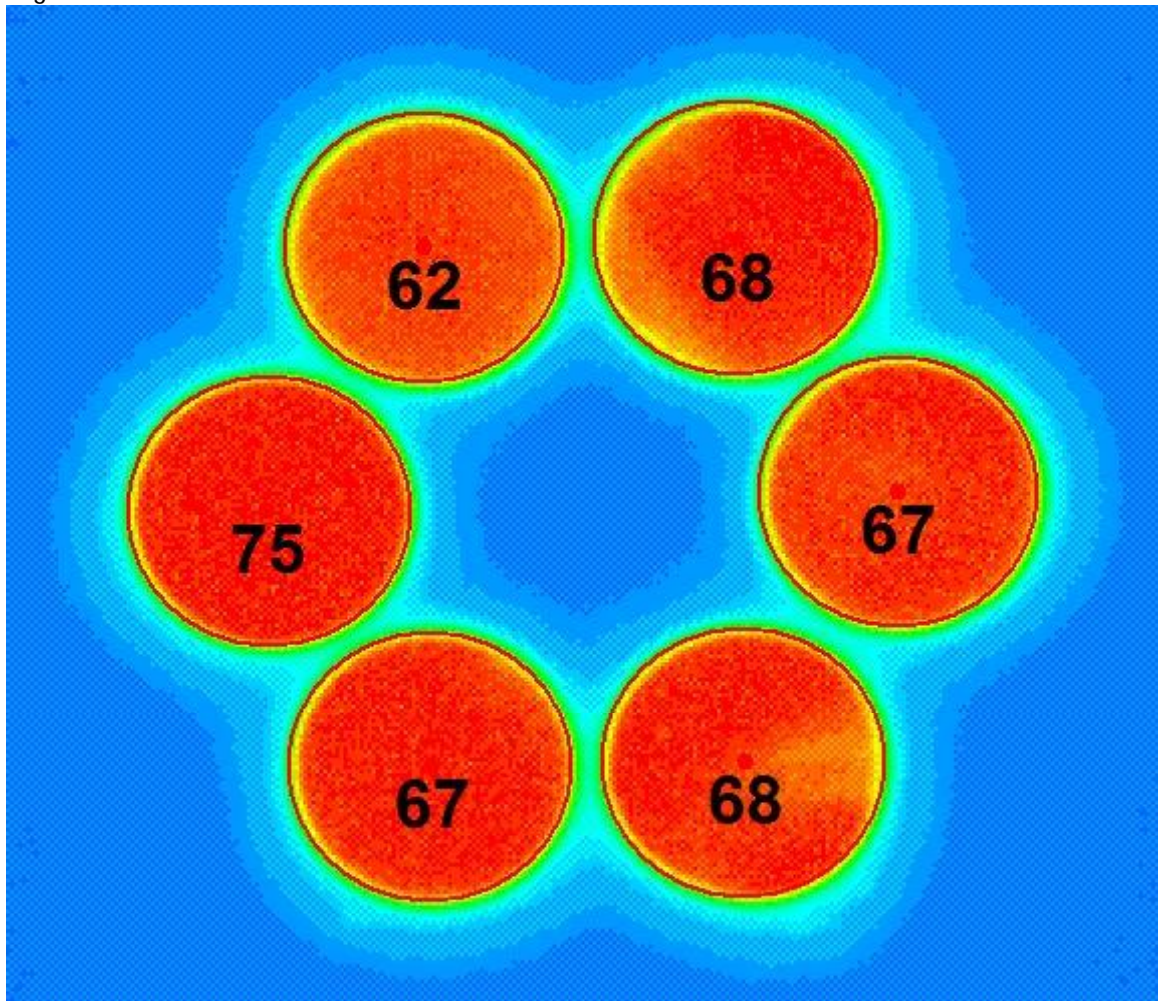
LIITE 2, Esimerkkituloste ohjelman tallentamasta sertifikaatista.

METSO OC Test report

Device information:

OC S/N	12341234
Probe code:	K11006
Probe S/N:	9684/4
TCU S/N:	1124241352

Fiber image:



Tuning Information:

Mean	68	Coefficient of variation	0,060
Meas	0,449 V	Meas reference	0,450 V
Vcomp	1,400 V	LED Current	47,2 mA

Cold shock values: (Values after cooling LED with -40°C air for 5 sec)

Meas	0,458 V	LED Current	45,9 mA
Vcomp	1,427 V		

Tested by:

METSO OC Test report

Order no, and Customer:	
Note:	
Device information: Current TCU Date and Time (DD.MM.YY HH:MIN:SEC): 07.01.2013 13:46:16 TCU SW Version= 1.1(1712.10) TCU HW Version= C TCU SN: 1124241352 Sensor SW Version = 1.10 Sensor HW Version = 1.30 OC Serial Number = 12341234 CPU Serial Number = 1239114730 Probe code : K11006 Probe version : 1.0 Probe S/N : 9684-4	
Configuration: Output 1 mA Signal, Manufacturing Trim: Offs: = 0 - Gain: 1.0000 Cs limits: LRV: 0.50 - URV: 7.00 Damping = 1.00 Alarm current = 3.7mA Recipe 1: Pulptype GW (A)Gain: 2.2800, (B)Exp: 1.2500, (C)Offset: 0.0000 Recipe 2: Pulptype GW (A)Gain: 2.2800, (B)Exp: 1.2500, (C)Offset: 0.0000 Recipe 3: Pulptype GW (A)Gain: 2.2800, (B)Exp: 1.2500, (C)Offset: 0.0000 Recipe 4: Pulptype GW (A)Gain: 2.2800, (B)Exp: 1.2500, (C)Offset: 0.0000 Output 2 Variable = Cs Cs limits: LRV: 0.50 - URV: 7.00 Damping = 1.00 Alarm current = 3.7mA	
Tuning Information: VRFact = 1.3965 LED Current = 47.2 Tuning date:07.01.2013 13:42:14	
Tested by:	