

Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Toni Österholm

Kunnonvalvonnan värähtelymittaus

Mittalaitteen prototyyppi ja kenttätesti

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

9.10.2020

Tekijä Otsikko Sivumäärä Aika	Toni Österholm Kunnonvalvonnan värähtelymittaus Mittalaitteen prototyyppi ja kenttätesti 51 sivua + 12 liitettä 9.10.2020
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	sähkö- ja automaatiotekniikka
Ammatillinen pääaine	automaatiotekniikka
Ohjaajat	automaatioasiantuntija Ilari Sandborg lehtori Kai Virta
<p>Insinööriyön aiheena oli kunnonvalvonnan värähtelymittaus sekä mittalaitteen prototyypin suunnittelu ja valmistaminen. Mittalaitteeseen toteutettiin lämpö- ja värähtelymittaus, joista tässä työssä keskityttiin värähtelymittaukseen. Mittalaitetta testattiin kentällä tehtävissä mittauksissa, joissa mitattiin kuuden tarjotinvaraston tarjotinhissien värähtelyä.</p> <p>Tavoitteena oli valmistaa tilaajan vaatimusmäärittelyn täyttävä mittalaitteen prototyyppi, jolla pystyy mittaamaan värähtelyä ja lämpötilaa. Kenttätestien tavoitteena oli kokeilemalla selvittää, pystyykö tarjotinhisseiltä tuottamaan värähtelytason tunnuslukuja.</p> <p>Värähtelymittauksen ja kenttätestin suunnittelussa ja toteutuksessa käytettiin apuna standardien määryksiä. Mittalaite toteutettiin Raspberry Pillä. Värähtelymittauksessa käytettiin ADXL345-kiihtyvyyssanturia. Laitteen kotelointi suunniteltiin Solidworks-mallinnusohjelmistolla ja sen tulostamiseen käytettiin Anet A8 3D-tulostinta. Datan käsittelyyn käytettiin SciDAVis-, Libre Office Calc- ja Logger Pro-ohjelmistoja.</p> <p>Insinööriyön tuloksena syntyi kunnonvalvonnan mittalaitteen prototyyppi, jolla pystyy kokeilemaan kunnonvalvontaa lämpötilamittauksella sekä värähtelymittauksella. Kenttätestien tuloksena opittiin tarjotinhissien ja yleisesti kunnonvalvonnan haasteista, kun työvälineenä käytetään värähtelymittausta. Kenttätesteissä tuotettiin tavoitteen mukaisesti värähtelytason ilmaisevia tunnuslukuja, mutta niiden hyöty kyseenalaistettiin ja esitettiin parannusehdotuksia, joilla prosessia on mahdollista jatkokehittää.</p> <p>Mittalaitteen jatkotoimenpiteinä ensisijaisena ovat prototyypin mekaanisen kestävyuden parantaminen vaihtamalla koteloinnin materiaalia sekä laitteen sisäisten johdotusten ja juottamisen minimointi suunnittelemalla piirilevy.</p> <p>Tarjotinhissien värähtelymittausten jatkotoimenpiteenä tulisi tutkia PLC:n käyttämistä mittausprosessissa, jolloin mitta-ajon pystyisi optimoimaan kunnonvalvonnan värähtelymittaukseen sopivaksi. Tämä mahdollistaisi myös prosessin automatisoinnin kiinteää järjestelmää käyttäen, jolloin mittaavaa resurssia ei tarvita. Mittauksissa havaittua häiriötä voidaan vähentää kehittämällä kiinnityspisteitä komponentteihin.</p>	
Avainsanat	kunnonvalvonta, värähtelymittaus, raspberry pi, adxl345

Author Title Number of Pages Date	Toni Österholm Condition Monitoring by Vibration Measurement Measurement Device Prototype and Field Test 51 pages + 12 appendices 9 October 2020
Degree	Bachelor of Engineering
Degree Programme	Degree Programme in Electrical and Automation Engineering
Professional Major	Automation Engineering
Instructors	Ilari Sandborg, Automation Specialist Kai Virta, Senior Lecturer
<p>The subject of this graduate thesis is condition monitoring by vibration measurement and the designing and manufacturing of a prototype of a measuring device. Thermal and vibration measurements were implemented on the measuring device. This thesis focuses on the vibration measurement. The measuring device was tested in field measurements by measuring the vibration of the Tray Cranes of six Tray Warehouses.</p> <p>The goal was to produce a prototype of a measuring device that meets the customer's specification and can measure vibration and temperature. The goal of the field test was to find out if it was possible to generate vibration level key figures from the measurements</p> <p>Specification of the standards were used to help in the designing and implementation of the vibration measurement and the field test. Raspberry Pi was used to prototype the measuring device. ADXL345-accelerometer was used in the vibration measurement. The casing was modeled with Solidworks software and it was printed with Anet A8 3D-printer. Software used to process the data were SciDAVis, Libre Office Calc and Logger Pro.</p> <p>As a result, a prototype of a condition monitoring measuring device was produced. With this device, condition monitoring by temperature and vibration measurement can be tested. As a result of the field tests, Information was gathered about the challenges of condition monitoring of Tray Cranes using vibration measurement as a tool. Vibration level key figures were produced, but the benefits of these figures were questioned and to improve the process follow-up actions were suggested.</p> <p>As a follow-up to the measuring device, firstly the mechanical durability of the prototype should be improved by changing the housing material. Next step would be to minimize the internal wiring of the device by designing a circuit board. As a follow-up to the vibration measurements of the Tray Cranes, the inclusion of the PLC in the process can be investigated, so that the measurement run could be optimized to work better for the vibration measurement. This would also allow for the measuring process to be automated using a fixed system, eliminating the need for a measuring resource. The interference in the measurement data can be lessened by improving the sensor mounting system.</p>	
Keywords	vibration measurement, prototype, condition monitoring

Sisällys

Lyhenteet

1	Johdanto	1
2	Tilaaajan kunnossapitotoiminnot	3
2.1	Ennakkohuolto-ohjelma	4
2.2	Korjaukset	5
2.3	Optimaalinen tasapaino	6
3	Kunnonvalvonnan värähtelymittaus	7
3.1	Signaalin peruskäsitteet	8
3.2	A/D-muunnos	9
3.3	Nyquistin teoreema	10
3.4	Taajuustaso	11
3.5	Aikatasovalvonta	12
3.6	Spektrivalvonta	14
3.7	Tunnuslukuvalvonta	15
3.8	Anturit	16
3.9	Kiihtyvyyssanturin kiinnitys	17
4	Mittalaitteen esiselvitykset	18
4.1	Mittalaitteen vaatimusmäärittely	19
4.2	Mikrotietokone	20
4.3	Värähtelymittaus	21
4.4	Tiedonsiirto ja ajan päivitys	23
4.5	Käyttöliittymä	24
4.6	Komponenttilista	25
5	Mittalaitteen toteutus	26
5.1	Kytkenät ja kotelointi	26
5.2	Mittalaitteen toimintakuvaus	28

5.3	Ohjelmointi	29
5.3.1	Värähtelymittaus	30
5.3.2	Automaattiliipaisu	34
5.4	Tiedonsiirto ja etäyhteys	35
6	Kenttätesti	36
6.1	Kohde	36
6.2	Suunnitelma	37
6.3	Mittapisteen valinta	38
6.4	Mittaukset	39
6.5	Tunnusluvut	43
6.6	Tulos	45
7	Yhteenveto	46
	Lähteet	48

Liite 1. LCD näytön ohjelmakoodi. *Liite vain tilaajan käyttöön.*

Liite 2. Lämpötilamittauksen ohjelmakoodi. *Liite vain tilaajan käyttöön.*

Liite 3. Ohjelmakoodi ADXL345 lukemiseen Python-ohjelmointikielellä. *Liite vain tilaajan käyttöön.*

Liite 4. Ohjelmakoodi ADXL345 lukemiseen C-ohjelmointikielellä. *Liite vain tilaajan käyttöön.*

Liite 5. TCRA002 mittausten kuvaajat.

Liite 6. TCRA003 mittausten kuvaajat.

Liite 7. TCRA004 mittausten kuvaajat.

Liite 8. TCRA005 mittausten kuvaajat.

Liite 9. TCRA006 mittausten kuvaajat.

Liite 10. TCRA007 mittausten kuvaajat.

Liite 11. Laitteen piirikaavio.

Liite 12. Esimerkki integraalista Logger Prossa.

Lyhenteet

ABS	<i>Akryylinitriilibutadieenistyreeni</i> . Kestävä muovilaatu, jota voidaan käyttää 3D-tulostamisessa.
C	Ohjelmointikieli.
FFT	<i>Fast Fourier Transform</i> . Nopea Fourier-muunnos. Tehokas algoritmi, jolla signaali saadaan esitettyä taajuustasossa.
GPIO	<i>General Purpose Input Output</i> . Mikrokontrollereissa olevat pinnit, jotka pystyy ohjelmoida sisään tai ulostuloiksi.
HDMI	<i>High Definition Multimedia Interface</i> . Digitaalinen näyttölaitteiden liitäntästandardi.
I2C	<i>Inter-Integrated Circuit</i> . Kaksisuuntainen tiedonsiirtoväylä. kahden johtimen väylä, jolla kytkettävät komponentit yksilöidään osoitteella.
MEMS	<i>Micro Electro Mechanical System</i> . Mikrosysteemi. Käytetään erittäin pienien komponenttien, kuten mikroanturien yhteydessä, kun yksittäisten rakenteiden kokoluokka on mikrometreissä.
PLA	<i>Polyaktidi</i> , 3D-tulostuksessa käytetty helppotulosteinen uusiutuvista raaka-aineista valmistettava muovilaatu.
PLC	<i>Programmable Logic Controller</i> . Ohjelmoitava logiikka. Pieni tietokone, jota käytetään automaation ohjauksessa.
PSK	Standardisointiyhdistys ry
RMS	<i>Root Mean Square</i> . Neliöllinen keskiarvo tai tehollisarvo. Kertoo aaltomuodon keskimääräisen voimakkuuden.

SSH	<i>Secure Shell</i> . Salattu tietoliikenteen protokolla, jolla voidaan muodostaa konsolikäyttöinen etäyhteys.
SPI	<i>Serial Pheripheral Interface</i> . Neljäjohtiminen nopea väylä.
TCRA	<i>Tray Crane</i> . Tarjotinhissi, siirtää tarjottimilla olevia tuotteita tarjotinvarastossa.
USB	<i>Universal Serial Bus</i> . Yleinen liitäntästandardi
VNC	<i>Virtual Network Computing</i> . Protokolla graafisen käyttöliittymän etäkäyttöön.

1 Johdanto

Insinööriyön aiheena on kunnonvalvonnan värähtelymittaus sekä mittalaitteen prototyypin suunnittelu ja valmistaminen, koska työn tilaajalla on tarve tutkia uusia menetelmiä laitteiston vikatiheyden madaltamiseksi. Värähtelymittaus oikein toteutettuna on erittäin tehokas kunnonvalvonnan keino, jolla viat ja häiriöt pystytään havaita muita menetelmiä aikaisemmin. [3, s. 17.]

Insinööriyön tilaaja WIOSS Witron On Site Services GmbH on saksalaisen WITRON Logistik & Informatik GmbH:n tytäryhtiö. WIOSS tuottaa kunnossapito-, operointi- ja asiantuntijapalveluita WITRONin suunnittelemiin ja toteuttamiin logistiikka-automaatiojärjestelmiin Australiassa, Euroopassa ja Amerikassa. Yli 90 prosenttia asiakkaista on vähittäiskauppaketjuja. [19.] WIOSS työllistää Suomessa yli 160 ihmistä.

Työn lähtökohtana on tilaajan ajatus mikrotietokoneella toteutetusta mittalaitteesta, jota voitaisiin käyttää ennakkohuollossa sekä kunnonvalvontakeinojen kehittämiseen. Mittalaitteeseen toteutetaan lämpö- ja värähtelymittaus, joista tässä työssä keskitytään värähtelymittaukseen. Mittalaitetta on tarkoitus testata kentällä tehtävissä mittauksissa.

Tavoitteena on valmistaa tilaajan vaatimusmäärittelyn täyttävä mittalaitteen prototyyppi, jolla pystyy mittaamaan värähtelyä ja lämpötilaa. Kenttätestien tavoitteena on värähtelyn tunnuslukujen tuottaminen mittauksista. Värähtelymittauksen ja kenttätestin suunnittelussa ja toteutuksessa käytetään standardien määräytyksiä. Mittalaite toteutetaan Raspberry Pillä. Värähtelymittauksessa käytetään ADXL345-kiihtyvyyssanturia. Dataa käsitellään SciDAVis-, Libre Office Calc- ja Logger pro ohjelmistoilla, joilla onnistuu datan esittämisen lisäksi myös analysointi ja kokeilut.

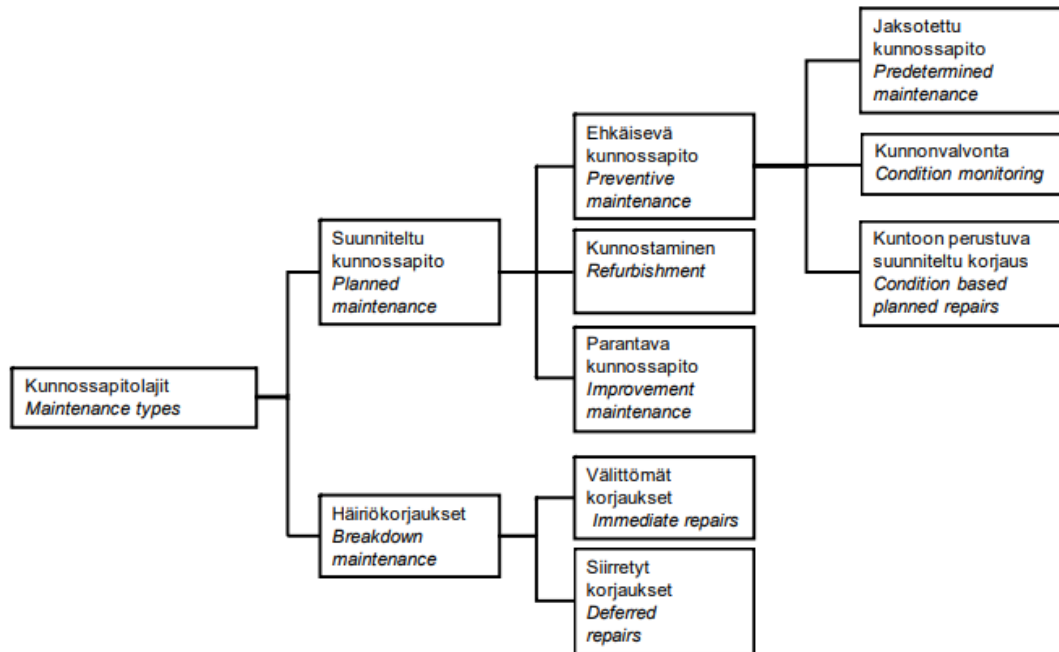
Insinööriyön kenttätesti toteutetaan Inex Partnersin rakennuttamassa Suomen suurimmassa logistiikkakeskuksessa Sipoossa, jonne saksalainen WITRON on toimittanut laajan keräysautomaatiojärjestelmän. Inex Partners tuottaa S-ryhmän käyttö- ja päivittäistavaran varastointi- ja kuljetuspalvelut. Tuotteiden varastointi, keräily ja kuljetus laitoksen vastaanoton ja lähettämön välillä on pitkälle automatisoitua. 2012 käyttöönotetun käyttötavarajakelukeskuksen ja vuonna 2016 käyttöönotetun

päivittäistavaran jakelukeskuksen yhteenlaskettu pinta-ala on noin 270 000 m². Kuljetuksia on noin 1000 ajoneuvoyhdistelmää päivässä. [1.]

Laitoksen kunnossapidosta vastaavan WIOSSin huolto-organisaation tehtävä on huoltaa ja korjata automaatiolaitteita sekä ennaltaehkäistä häiriöiden ja kulumien aiheuttamia tuotannon viivästymisiä. Yllättävät häiriöt ja kulumat pyritään havaitsemaan ennakkohuolto-ohjelman tarkastuksilla sekä статистиikkaa hyödyntämällä ennen suunnittele mattoman korjauksen tarvetta. Suunnittele mattoman pysäytyksen aiheuttaman häiriön tai rikkoutumisen tapahtuessa juurisyy selvitetään ja samankaltaisten rikkoutumisten uusiutuminen pyritään estämään.

2 Tilaajan kunnossapitotoiminnot

Tässä luvussa käydään läpi, miten tilaajan kunnossapitotoiminnot on jaettu kunnossapitolajeihin. Tilaajan jako noudattaa samaa linjaa kuin standardin PSK7501 kuvassa 1 määrittelemä jako.



Kuva 1. Kunnossapitolajit [2].

Käytännön erona on se, että SPK7501:n esittelemä "kuntoon perustuva suunniteltu korjaus" ja "siirretyt korjaukset", tehdään molemmat työlajina "suunniteltu korjaus" (planned repair).

Yrityksen kunnossapidon keskiössä on selainpohjainen Witool-huoltohallinnanjärjestelmä. Witoolista on seuraavat osiot:

- *Työmääräysjärjestelmä*, joka sisältää perusteelliset tiedot laitoksen jokaisesta kunnossapitotyöstä toiminnan mittausta, seuranta ja kehittämistä varten.
- *Varaosajärjestelmä*, josta löytyy tiedot varaosista ja niiden määristä ja hyllypaikoista.

Osa työmääräyksistä generoituu automaattisesti ennakkohoolto-ohjelman mukaisesti, muut työmääräykset lisätään tarpeen mukaan. Työmääräyksen tiedot sisältävät

- työlajin
- kohdetiedot
- tarvittavat dokumentit
- lähtötiedot
- tekijän raportti
- käytetyt resurssit
- vaihdetut osat.

Kunnossapitotyöt eli työmääräykset jaetaan järjestelmässä työlajeihin, jotka määrittävät työn tarkoituksen. Tärkeimmät työlajit ovat:

- Ennakkohoolto (Preventive maintenance).
- Tutkinta (Inspection).
- Puhdistus (Cleaning).
- Suunniteltu korjaus (Planned repair).
- Suunnittelematon korjaus (Unplanned repair).
- Parannustyö (Improvement).
- Kunnostamistyö (Spare part) [18].

2.1 Ennakkohoolto-ohjelma

Ennakkohoolto-ohjelma pitää sisällään kolme aikataulutettua toistuvaa työlajia, jotka ovat ennakkohoolto, tutkinta ja puhdistus. Järjestelmä generoi nämä työt huoltojärjestelmään automaattisesti ja ne toteutetaan ennalta määritettyjen ohjeiden mukaisesti.

Ennakkohoultotyöllä testataan ja kunnostetaan järjestelmää kulumisen varalta. Tarvittaessa osia vaihdetaan.

Tutkinnalla pyritään havaitsemaan alkavia häiriöitä ja kulumia aistinvaraisin havainnoin. Joidenkin osien kunto tarkastetaan mittauksilla - esimerkiksi osien välyksien ja hihnojen

jännitteiden mittauksilla. Tutkinta sisältää myös pienten häiriöiden korjaamiset. Pidempikestoista korjausta vaativat työt pyritään suorittamaan suunniteltuna korjauksena ilman tuotannon häiriintymistä.

Puhdistuksella varmistetaan laitteiden pitkäkestoinen toiminta ja työskentelyn turvallisuus. [18.]

2.2 Korjaukset

Ennakkohuolto-ohjelmassa tehdyssä tutkinnassa havaittujen häiriöiden ja kulumisten vaatiessa pidempikestoisia korjaustöitä käytetään työläjaina suunniteltua tai suunnittelematonta korjausta. Myös operatiivisen henkilökunnan havaintojen, статистиikan ja muissa yrityksen laitoksissa tehtyjen havaintojen kautta generoidaan korjauksia.

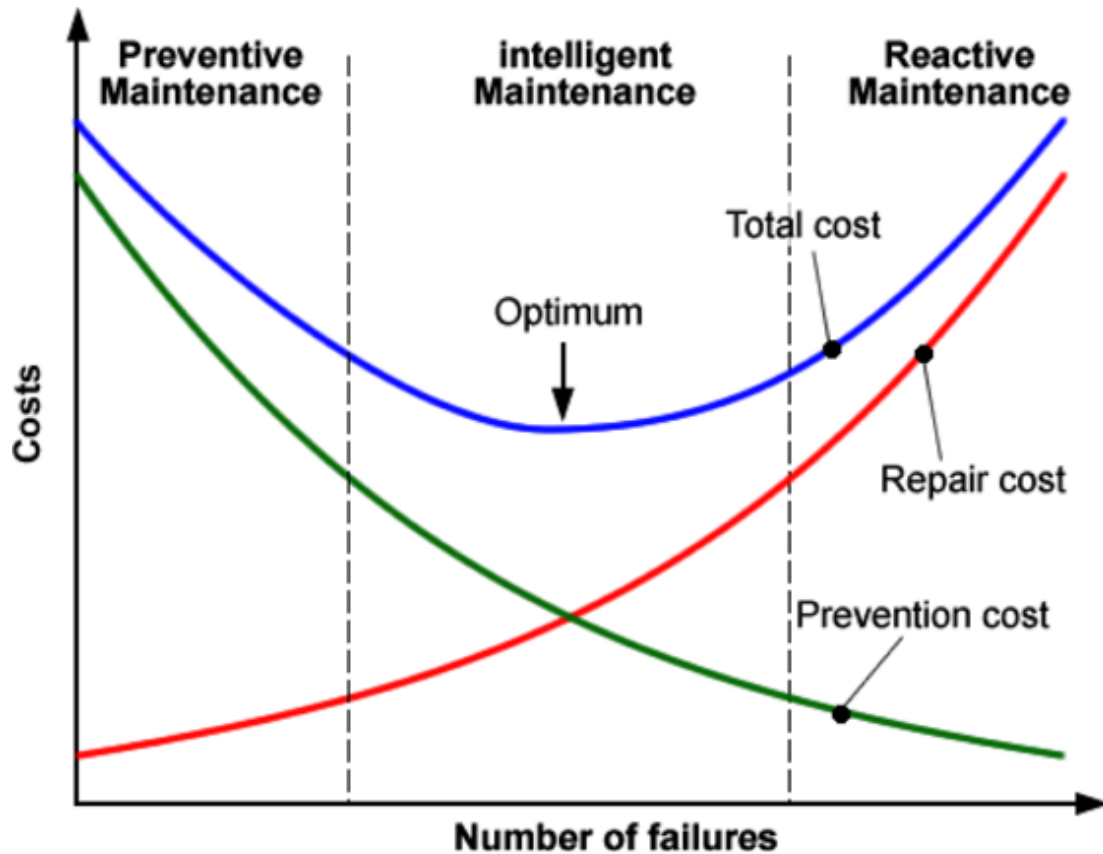
Korjaustoimenpiteet pyritään ensisijaisesti järjestämään suunniteltuna korjauksena niissä tapauksissa, kun laite vielä toimii mutta vaatii korjauksen. Suunniteltu korjaus aikataulutetaan priorisoiden niin, että tuotannolle ei aiheudu keskeytystä.

Jos laite on rikkoutunut niin, että sitä ei pystytä pitämään ajossa, tehdään laitteelle suunnittelematon korjaus, jossa tarvittavat osat vaihdetaan, korjataan tai suoritetaan muut tarvittavat toimenpiteet, joilla laite saadaan mahdollisimman nopeasti takaisin toimintakuntoon. Tarvittaessa töitä jatketaan myöhemmin suunnitellusti.

Suunnittelematon rikkoutuminen aiheuttaa viivästyä tuotannolle. Näistä töistä tehdään juurisyyanalyysi, jonka pohjalta tarvittavat toimenpiteet toteutetaan samankaltaisten häiriöiden tai rikkoutumisten välttämiseksi. Analyysin pohjalta ennakkohuolto-ohjelmaa muokataan tai suoritetaan IT, PLC, sähkö tai mekaaninen parannustoimenpide, jolla juurisyy häiriön toistuminen estetään. [18.]

2.3 Optimaalinen tasapaino

Kunnossapito-ohjelman tavoitteena on löytää kuvan 2 mukainen optimaalinen tasapaino ennakkohuollon ja korjausten väliltä, jolloin kustannukset pysyvät alhaisimmalla tasolla.

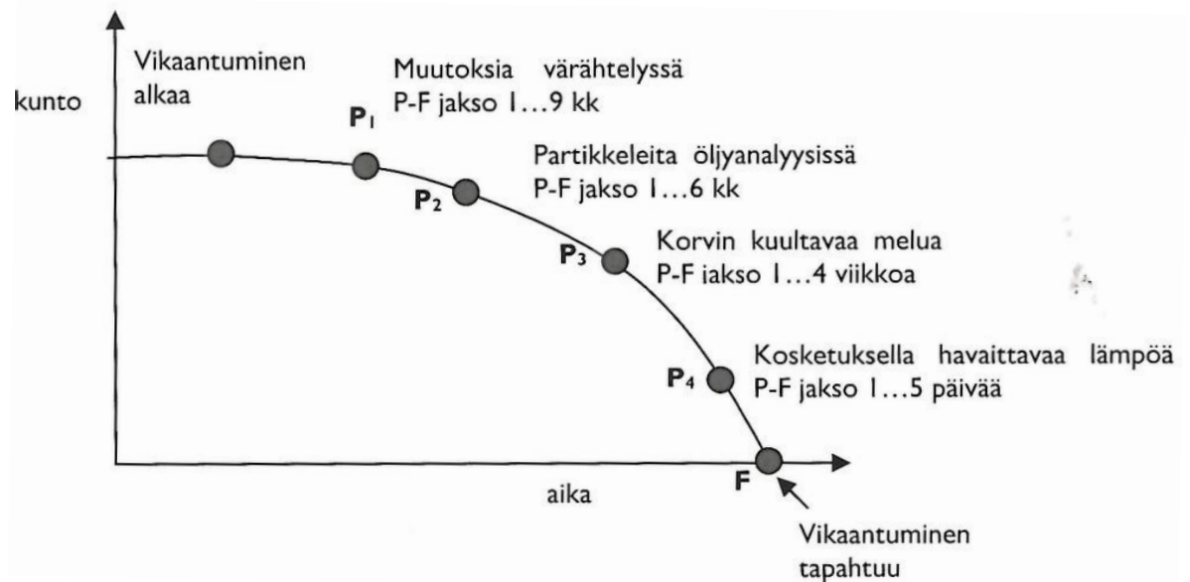


Kuva 2. Optimaalinen kunnossapito [18].

3 Kunnonvalvonnan värähtelymittaus

Tässä luvussa esitellään työn kannalta oleellista tietoa ja käsitteitä kunnonvalvonnan värähtelymittauksesta, joita käytettiin mittalaitteen ja kenttätestien toteutuksessa.

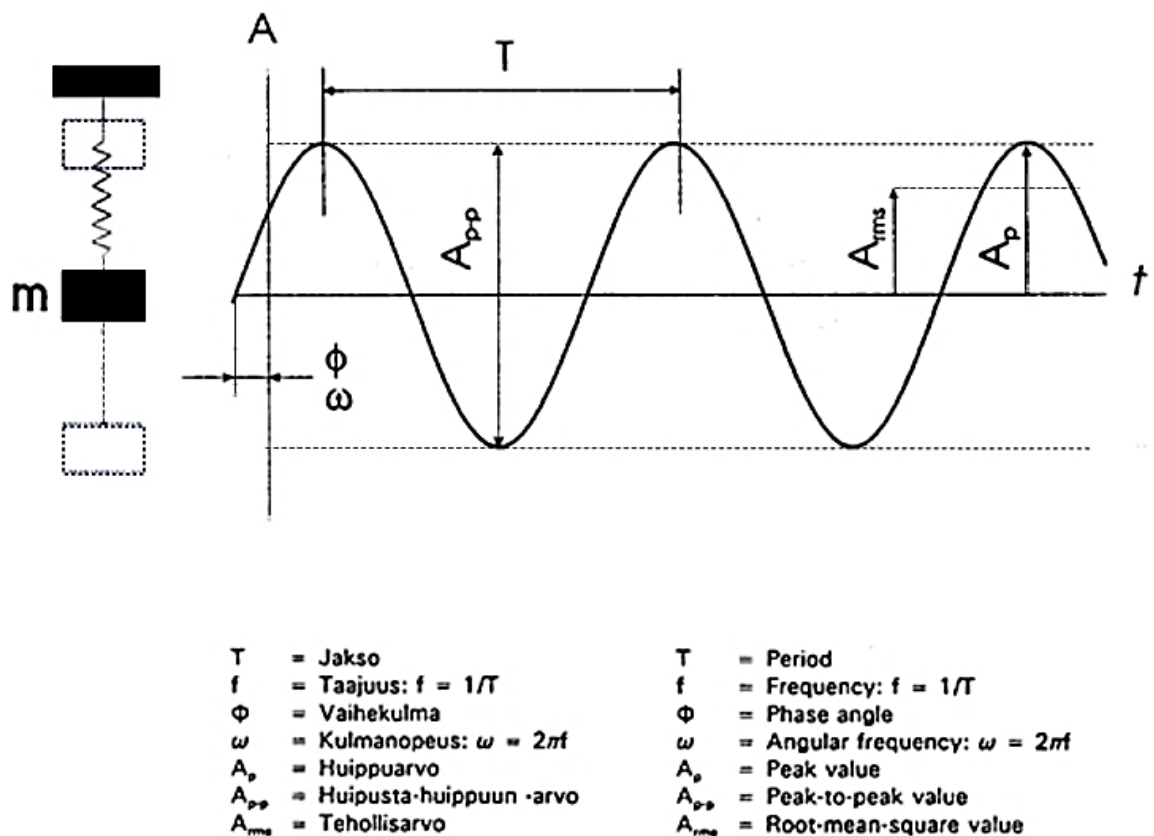
Kunnonvalvonnessa värähtelymittaus on yleisin menetelmä laitteen suorituskyvyn ja jäljellä olevan luotettavan käyttöajan seurannassa. Lisäksi sitä käytetään käytönvalvonnessa ja vikaselvityksissä. Oikein sovellettuna menetelmällä on mahdollista havaita alkava häiriö ennen muita menetelmiä. [3, s. 17.] Värähtelyä analysoimalla voidaan löytää esimerkiksi epätasapaino, mekaaniset välykset, rakenteen resonanssitaajuudet, taipunut akseli ja asennusvirheitä. [11, s. 13.] Kuvassa 3 P-F jakso tarkoittaa sitä ajanjaksoa, joka oireen havaittavuudesta kuluu vikaantumiseen.



Kuva 3. Vierintälaakerin vikaantumisen oireet [5, s. 141].

3.1 Signaalin peruskäsitteet

Värähtelymittauksen suurena voidaan käyttää siirtymää, nopeutta tai kiihtyvyyttä. Tyypillisesti mitattava taajuusalue määrittää käytettävän suureen. Tässä työssä värähtelymittaukseen käytetään kiihtyvyyssanturia ja värähtelyn suurena kiihtyvyyttä. Signaalin peruskäsitteet pätevät muillekin suureille, joita värähtelymittauksissa käytetään. Värähtelymittauksia mitataan siirtymänä, nopeutena ja kiihtyvyytenä. Karkeasti voidaan sanoa, että hidasta värähtelyä mitataan siirtymänä, keskinopeuksista nopeutena ja suurinopeuksista kiihtyvyytenä. Kuvassa 4 nähdään miten aikatason yleiset tunnusluvut A_p , A_{p-p} ja A_{rms} muodostetaan. Pystyakselilla on kiihtyvyyys, mutta se voisi siis olla myös nopeus tai siirtymä riippuen käytetystä mittausmenetelmästä.



Kuva 4. Aikatasosignaalin peruskäsitteet [4, s. 6].

3.2 A/D-muunnos

Analogia-digitaalimuunnoksella signaali muutetaan jatkuvamuotoisesta analogisignaalista diskreetiksi numerojonoksi eli binääriarvoksi. Digitaalisessa muodossa dataa pystytään mm. analysoimaan ja jalostamaan tietokonelaskennalla.

Digitaalisessa anturissa A/D muunnos tehdään paikallisesti anturissa ja data lähetetään eteenpäin digitaalisesti binäärilukuina. Käytännössä A/D-muunnin mittaa analogisen signaalin jännitearvon tasaisin väliajoin ja muuttaa sen binääriarvoksi. [5, s. 185.] Tavoitteena on saada signaali muutettua analogisesta digitaaliseen muotoon, niin että sen alkuperäiset ominaisuudet ovat tallella sovelluksen vaatimalla tarkkuudella. Oleellisimpia arvoja ovat taajuus ja amplitudi.

A/D-muuntimen nopeus eli näytteenottotaajuus määrittää sen, kuinka korkeita taajuuksia anturilla pystyy mitata. Muuntimen bittimäärä taas kertoo erottelukyvyn eli resoluution, jolla mitattavan signaalin amplitudimuutokset saadaan muutettua digitaaliseksi arvoksi. Käytännössä suurempi bittimäärä tarkoittaa, että anturin jännitemuutokset saadaan talteen pienemmällä jänniteportaalla, koska jännitealueen osiin jakamiseksi on käytettävissä enemmän bittejä. [5, s. 187.] Jänniteportaan suuruuden voi laskea kaavalla 1,

$$V_{ref} = \frac{(V_{ref+}) - (V_{ref-})}{2^N} \quad (1)$$

jossa V_{ref} on AD- muuntimen jänniteporras, N on bittimäärä, V_{ref+} ja V_{ref-} ilmaisee jännitealueen suurimman ja pienimmän jännitteen, joiden erotus on anturin jännitealue. [5, s. 187.]

Kaavaa voidaan soveltaa ilmoittamaan anturin resoluutio kiihtyvyyden yksikössä. Esimerkiksi, ADXL345-anturissa on 13 bitin muuntaja ja 16G:n amplitudialue. Kaavan 2 mukaisesti voidaan laskea

$$A_{ref} = \frac{2 * 16}{2^{13}} \quad (2)$$

jolloin ADXL345-anturin erottelukyvyyksi Aref saadaan 0,00390625 g eli n. 0,038 m/s²

3.3 Nyquistin teoreema

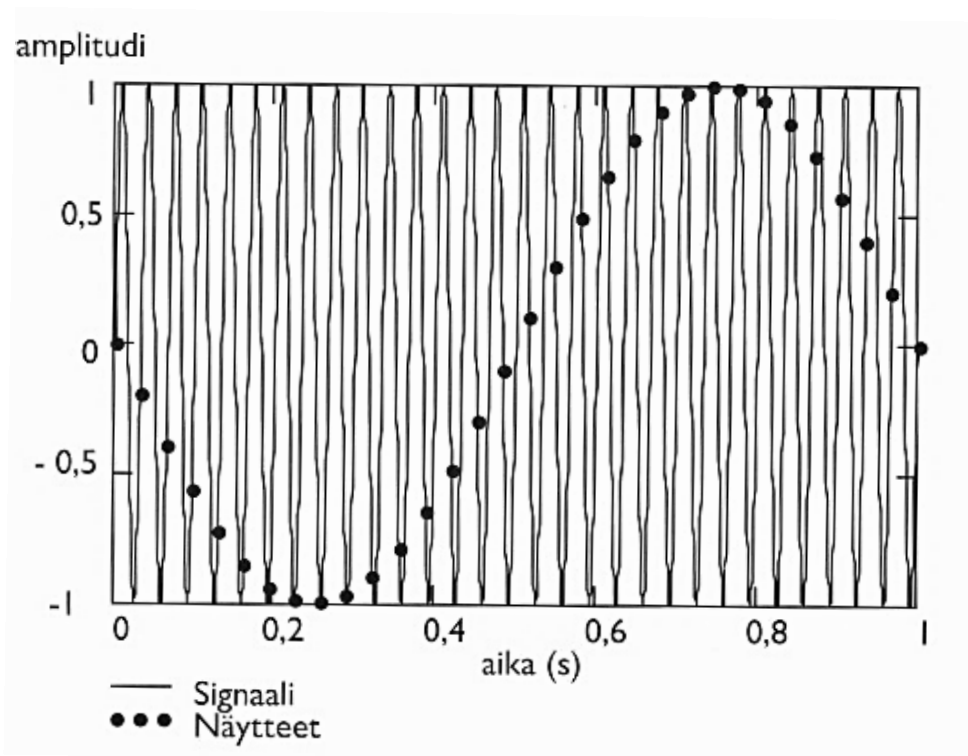
Nyquistin teoreema esittää kaavan 3 mukaisesti, että signaalista tulee ottaa näytteitä kaksi kertaa suuremmalla taajuudella, kuin mitä signaalin suurin esitettävä taajuus on suuruudeltaan. [6, s. 39.]

$$f_N = \frac{f_S}{2} \quad (3)$$

jossa f_N on Nyquistin taajuus eli suurin esitettävä taajuus ja f_S on näytteenottotaajuus.

Teoreema määrittää siis AD-muunnoksen vähimmäisnopeuden, eli vaaditun näytteenottotaajuuden, jolla värähtely saadaan muutettua digitaaliseen muotoon niin, että signaalin ominaisuudet säilyvät.

Nyquistin taajuutta käytetään kuvan 5 laskostumisilmiön välttämiseksi.



Kuva 5. Laskostumisen vaikutus signaaliin [5, s. 270].

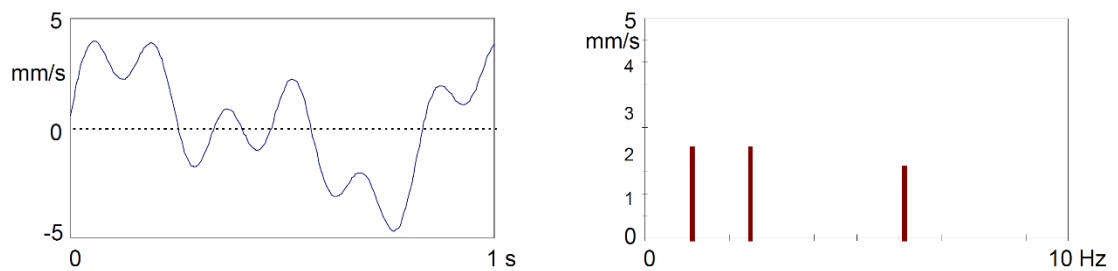
Laskostumisen seurauksena signaalin kuvaaja vääristyy, koska näytteenottopisteet eivät riitä toistamaan suurinta värähtelyn taajuutta.

Nyquistin taajuus pätee teoriassa ihannetilanteessa sinimuotoisella tai muulla tiedetyllä jaksollisella aaltomuodolla, jolloin muutamilla näytteillä voidaan päätellä aallon muoto. Käytännön mittauksissa tarvitaan usein suurempaa näytteenottonopeutta laskostumisen välttämiseksi.

3.4 Taajuustaso

Aikatason signaalia voidaan tarkastella myös taajuustasossa, jolloin kuvaaja piiryy ajan sijasta taajuuden funktiona. Tällöin värähtelyn osataajuuksien voimakkuudet saadaan yksilöityä. Signaali muutetaan taajuustasoon Fourier-muunnoksella. Fourier-muunnokseksi kutsuttuja menetelmiä on useampia erilaisia. Nykyisin yleinen menetelmä Fast Fourier Transformation (FFT), eli nopea Fourier-muunnos, toimii lähes reaaliajassa.

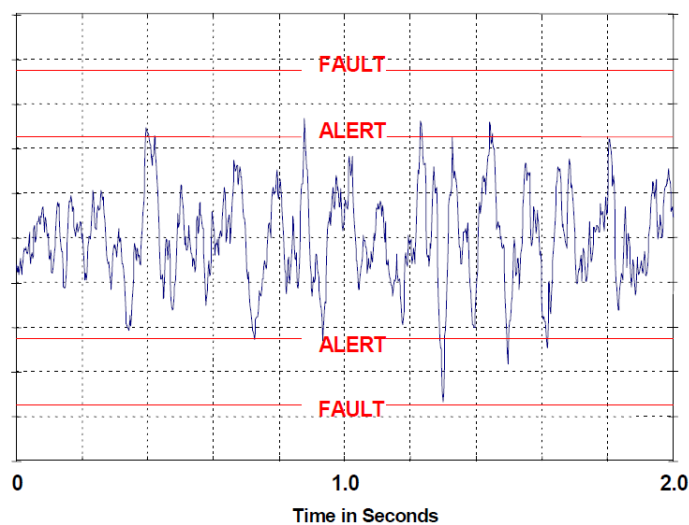
[5, s. 190–195.] Kuvassa 6 vasemmalla näkyy signaali aikatasossa ja oikealla taajuustasossa.



Kuva 6. Signaali aika- ja taajuustasossa [11, s. 10].

3.5 Aikatasovalvonta

Aikatasosignaalia voidaan analysoida silmämääräisesti muotoa tarkastelemalla tai automatisoidulla rajaseurannalla, jolloin ensin määritetään kuvan 7 mukaisesti varoitus- ja vikahälytysrajat, joihin tulevia mittauksia verrataan.

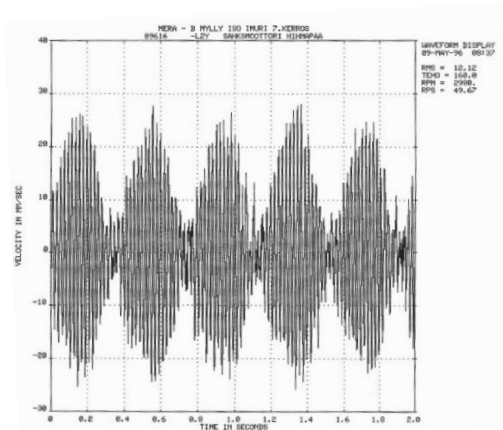


Kuva 7. Aikatasovalvonnan hälytysrajat [9, s. 4].

Normaalista poikkeavaa signaalin muotoa voidaan käyttää myös vikojen havaitsemiseen, jolloin analysoija etsii värähtelystä normaalista poikkeavia muotoja. Seuraavassa on esimerkkejä muodon poikkeamista:

Huojunta kuvassa 8 on tavallisesti merkki

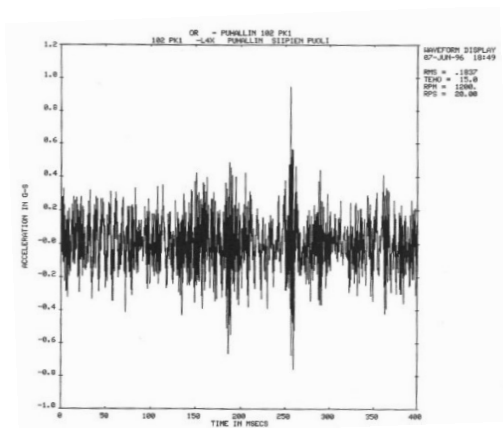
- sähkömoottorin sähköisestä viasta
- rakenteita pitkin kantautuvasta ulkoisesta tärinästä
- nivelakselin linjaus virheestä.



Kuva 8. Huojunta [3, s. 85].

Isku kuvassa 9 on tavallisesti merkki

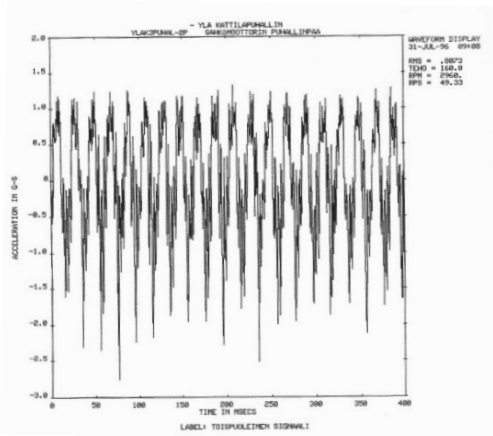
- laakeriviasta
- hammaspyöräviasta
- roottorin siiven osumisesta pesään.



Kuva 9. Aikatason isku [3, s. 85].

Toispuoleinen signaali kuvassa 10 on tavallisesti merkki

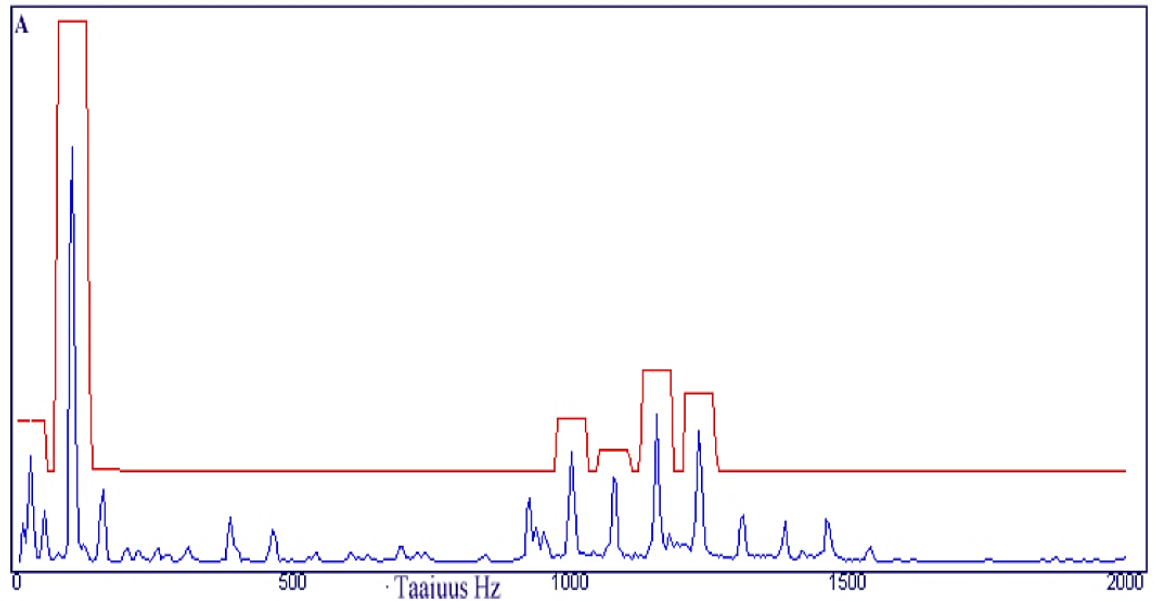
- linjausvirheestä
- hankauksesta
- löysästä liitoksesta
- irtonaisuudesta



Kuva 10. Toispuoleinen signaali [3, s. 86].

3.6 Spektrivalvonta

Spektrivalvonnassa koneen yksittäisten osien kuntoa pystytään valvomaan tarkastelemalla signaalin taajuustason komponenttien voimakkuuksia. Kun aikataason signaali muutetaan taajuustasoon, voidaan yhdellä mittauksella tarkastella myös useamman värähtelyakselin omaavien koneiden osavärähtelyjen suuruuksia. Mittauksen vaatimuksena on, että kohteen akselien kierroslukunopeudet ovat tiedossa. [3, s. 88.] Kuvassa 11 punainen viiva kuvaa aikaisempien mittausten perusteella määritettyä hälytysrajaa saman koneen useammille eri vikaantumismekanismeille.



Kuva 11. Spektrivalvonnan hälytysraja [9, s. 4].

3.7 Tunnuslukuvalvonta

Aikatasoin analysoinnin tehokkuutta voidaan lisätä käyttämällä hyödyksi tunnuslukuja, jolloin voidaan paremmin kohdentaa resursseja niihin kohteisiin, joissa poikkeamaa on havaittu. Tunnuslukulaskennan voi automatisoida ja hälytysrajojen käyttäminen on tavallista. [3, s. 83-84.]

Yleisimmät aikatasosta laskettavat tunnusluvut ovat

- tehollisarvo
- huipusta huippuun arvo
- huippuarvo
- huippukerroin eli crest factor
- kurtosis. [9, s. 3].

Aikatasoin tunnusluvuilla tarkkaillaan yleistärinän tasoa. Taajuustason tunnusluvuilla pystytään kohdentamaan valvonta tiettyyn vikataajuuskomponenttiin. Standardin PSK5706 määrittelemät taajuustason tunnusluvut ovat

- pyörimistaajuudella tapahtuvan värähtelyn voimakkuus
- pyörimistaajuudella tapahtuvan värähtelyn vaihekulma
- pyörimistaajuuden monikerroilla tapahtuvan värähtelyn voimakkuus
- värähtelyn voimakkuus vierintälaakerin sysäystaajuuksilla ja niiden monikerroilla
- värähtelyn voimakkuus lapataajuudella ja sen monikerroilla
- värähtelyn voimakkuus hammaspyöräparin ryntötaajuudella, sen monikerroilla ja niiden sivunauhoilla
- rakenteen valitun ominaistaajuuden sisältävän taajuuskaistan tehollisarvo
- harmonisten värähtelykomponenttien tehollisarvo
- ei harmonisten värähtelykomponenttien tehollisarvo
- aliharmonisten värähtelykomponenttien tehollisarvo
- valitun taajuuskaistan tehollisarvo [9, s. 3].

3.8 Anturit

Anturin valintaan vaikuttaa valvottavan koneen tai sen osien vikaantumisen havaitsemiseen liittyvät taajuudet ja värähtelyn amplitudi, jotka määrittävät anturin vaaditun taajuusalueen ja herkkyyden. Kunnonvalvonnan värähtelymittauksissa yleisimmät anturityypit ovat kiihtyvyydenturit, siirtymäanturit ja iskusysäysanturit. Jos kohteesta halutaan tietoa laajalla taajuusalueella, voidaan käyttää useampia antureita. [10, s. 2.]

Siirtymäanturina käytetään tavallisesti induktiivista pyörrevirta-anturia, jonka toiminta perustuu anturin muodostamaan magneettikenttään, joka indusoi pyörrevirtoja ferromagneettisille pinnoille. Induktiivisen mittauksen haasteena on, että anturia ei voi kiinnittää suoraan kohteeseen vaan sen pitää olla tukevasti paikallaan mittaamassa akselia. Jos anturi altistuu värähtelylle mittauksen aikana, on tällä välitön vaikutus anturin suhteelliseen lukemaan. [3, s. 49-50.] Tyypillinen käyttökohte on liukulaakeroidun koneen kunnonvalvonta. [5, s. 235.]

Nopeusanturit ovat tyypillisesti kiihtyvyydentureita kookkaampia. Niiden toiminta perustuu magneettiseen massaun, joka on kiinnitetty anturin päihin jousilla. Massa

aiheuttaa liikkeessaan jännitteen muutoksen anturin käämiin. Jousien aiheuttaman viiveen ansiosta nopeusanturien jännite on verrannollinen nopeuteen. [3, s. 47-48.]

Kiihtyvyyssanturin vahvuus on monipuolisuus. Anturin pietsosähköinen kide liikkuu yhtenäisesti mitattavan kohteen kanssa venyttäen tai puristaen kidettä aiheuttaen kiihtyvyyteen verrannollisen jännitemuutoksen. Laajan taajuuskaistansa ansiosta kiihtyvyyssanturit sopivat laaja-alaisesti värähtelymittauksiin, jolloin ei tarvitse käyttää esimerkiksi erillistä nopeusanturia nopeussuuretta käyttävään mittaukseen. Kiihtyvyyssanturilla voidaan suorittaa nopeusmittaus integroimalla yhden kerran ja siirtymämittaus integroimalla kaksi kertaa. [3, s. 46-47.]

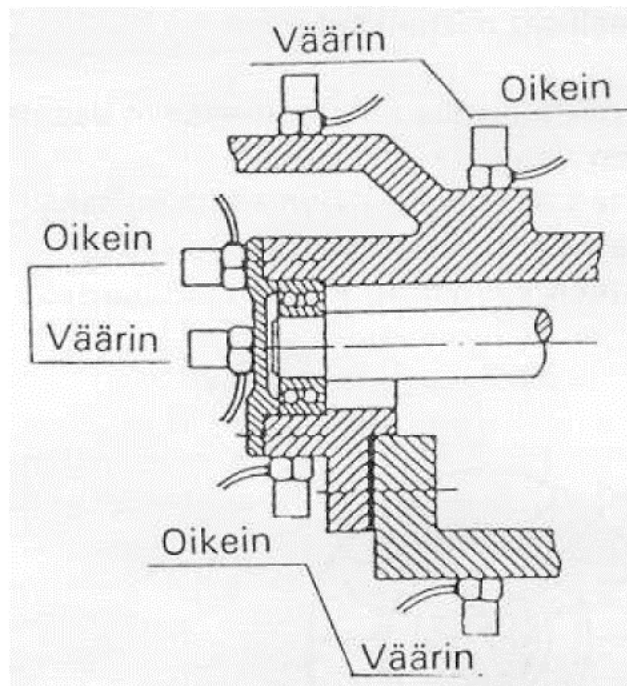
PSK 5706 suosittaa valvottavat suureet jaettavaksi taulukon 1 mukaisesti värähtelyn taajuuden mukaan. Iskusysäysantureina käytetään kiihtyvyyssantureita, joiden ominaisvärähtelytaajuutta käytetään hyväksi mittauksessa.

Taulukko 1. Suositeltava valvontasuure. [9, s. 2.]

Taajuus [Hz]	Valvontatapa
<10	Siirtymä
10-1000	Nopeus
>1000	Kiihtyvyys

3.9 Kiihtyvyyssanturin kiinnitys

Kiihtyvyyssanturin kiinnitys on kriittinen mittauksen onnistumisen kannalta. Kiinnityksessä tulee huomioida mm. anturin kiinnityskohta ja kiinnityksen lujuus. Kuvassa 12 on esitetty anturin oikeaoppinen kiinnittäminen.



Kuva 12. Kiithyvyysaturin oikeaoppinen kiinnitys [8, s. 2].

Yleisesti halutaan päästä mahdollisimman lähelle laakereita säteissuunnassa, mutta tarvittaessa mittaus suoritetaan akselin suuntaisesti. [8, s. 2.] Mitä korkeampia taajuuksia halutaan mitata, sitä lujempi tulee liitoksen olla. Lujimmat kiinnitykset ovat kiinteitä asennuksia, joissa anturi kiinnitetään joko suoraan kohteeseen tai käyttäen välissä nippaa tai kierrealuslevyä. Tällöin anturia ei irroteta mittausten välissä. Reittimittauksissa käytetään pikaistukoita tai magneetteja. Magneettia pystyy käyttämään myös ei ferromagneettisissa kohteissa liimaamalla kohteeseen aluslevyn, johon anturi kiinnitetään. [10.]

4 Mittalaitteen esiselvitykset

Insinööriyön lähtökohtana oli tilaajan ajatus mittalaitteen prototyypistä, jota asentajat voisivat käyttää ennakkohuollon yhteydessä kohteen kunnan määrittämiseen. Laitteella haluttiin mitata lämpötilaa ja värähtelyä.

4.1 Mittalaitteen vaatimusmäärittely

Mittalaitteen prototyypille tehtiin tilaajan kanssa seuraavanlainen vaatimusmäärittely, jonka perusteella laite toteutettiin:

1. Laitteen tuli olla mobiili eli kannettava ja akkukäyttöinen.
2. Laitteessa tuli olla näytöllinen käyttöliittymä, jolla mittaaja valitsee ja käynnistää halutun mittauksen.
3. Laitteella piti voida mitata kiihtyvyyttä ja lämpötilaa.
4. Antureiden tuli olla ulkoisia ja vaihdettavissa.
5. Mittausdata piti saada tallennettua tiedostoon.

Vaatimusmäärittely ei antanut laitteelle ja itse mittauksille tarkempia teknisiä vaatimuksia. Esiselvityksessä selvitettiin mittalaitteen pääpiirteet ja komponenttien vaatimukset.

Mittalaite hahmoteltiin kädessä pidettäväksi laitteeksi, jossa olisi näyttö ja painikkeet käyttöliittymää varten, sekä liitin antureita varten. Todettiin, että laitteeseen tarvitaan tärkeimpinä mikrotietokone, kiihtyvyyssanturi, lämpötila-anturi, näyttö ja varavirtalähde. Lisäksi tarvittaisiin kaapeleita, ruuveja, liittimiä ja painikkeita. Kotelointi suunniteltaisiin Solidworks ohjelmistolla ja tulostettaisiin 3D-tulostimella. Ennen osien tilaamista suunniteltiin laitteen pääpiirteet ja varmistettiin komponenttien laitteisto- ja ohjelmistotason ominaisuudet riittäviksi ja yhteensopiviksi.

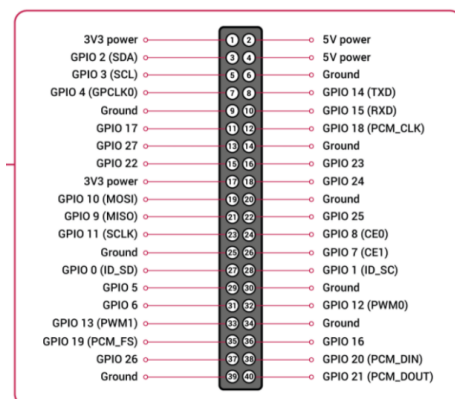
4.2 Mikrotietokone

Mittalaite toteutettiin Raspberry Pi 3 B+ -mikrotietokoneella. Se valittiin monipuolisuutensa takia. Projektin ohjelmisto-osuuden pystyi toteuttamaan täysin Raspberry Pillä. Sitä pystyy käyttämään kuten tietokonetta liittämällä siihen näytön, hiiren ja näppäimistön. Laitteesta löytyy myös Wifi ja Bluetooth sekä Ethernet-portti yhteyksiä varten, sekä tässä projektissa vaadittavat I2C, SPI-väylät komponenttien kanssa kommunikointiin. Käyttöjärjestelmäksi asennettiin Debian Busteriin pohjautuva Raspberry Pi OS linux-käyttöjärjestelmä. Kyseisessä käyttöjärjestelmässä on suora tuki Python-ohjelmointikielelle. Kuvassa 13 on esitetty Raspberry Pi 3 B+ -malli.



Kuva 13. Raspberry Pi 3 B+ [12].

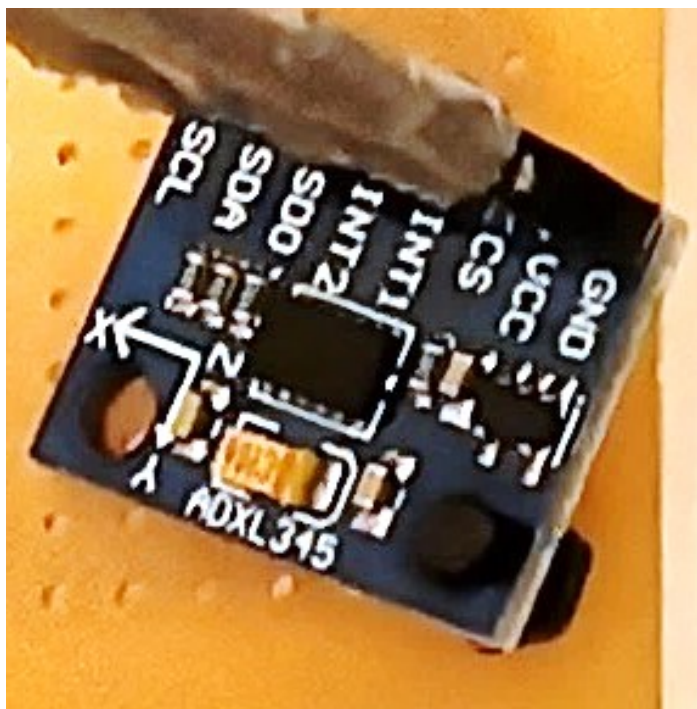
HDMI, Ethernet, ääni, USB-liityntöjen lisäksi laitteesta löytyy GPIO (General Purpose Input/Output), jonka pinnit pystyy määrittämään sisään- tai ulostuloiksi. Määrättyjä pinnejä voidaan käyttää myös UART, I2C tai SPI -väylinä. Kuvassa 14 on esitelty Raspberry Pin GPIO. [12]



Kuva 14. Raspberry Pin GPIO [12].

4.3 Värähtelymittaus

Värähtelymittauksessa käytettävän anturin valinnassa tutustuttiin olennaiseen värähtelymittauksen teoriaan ja tietoon, sekä otettiin huomioon saatavuus ja ohjelmoinnin haastavuus. Värähtelymittauksen anturiksi valittiin kuvassa 15 näkyvä yleinen Analog Devicen ADXL345 MEMS-kiihtyvyyssanturi, jonka ohjelmoinnin avuksi löytyi avoimen lähdekoodin ohjelmointikirjastoja GitHub.com sivustolta. [22; 23; 24.]



Kuva 15. ADXL345 3-akselinen MEMS-kiihtyvyyssanturi.

Kiihtyvyyssanturin valinnassa tärkeintä oli selvittää mitattavan kohteen värähtelyn taajuusalue ja amplitudi. Riittävän näytteenottotaajuuden arvioimiseen käytettiin myös yleisten vikojen ilmenemistaajuuksia.

Vaadittavan näytteenottotaajuuden määrittämisen esimerkki

Peruspyörimistaajuutta kuvataan kirjaimella n . Sen kerrannaisia käytetään eri vikataajuuksien esittämiseen. Taulukossa 2 on esimerkkejä vikataajuuksista.

Taulukko 2. Esimerkkejä vikojen suhteesta peruspyörimistaajuuteen. [3, s. 81.]

Vika	Taajuus [Hz]
Epätasapaino	n
Liukulaakerin öljykalvon pyörteily	0,45 ... 0,5 x n
Linjausvirhe ja kiinnitysten löystyminen	2 ... 6 x n

Esimerkkimoottorin tyypikilven nimellisyörimisnopeus on 2935 rpm. Kuvassa 16 on tarjotinvaraston tarjotinhissin moottorin tyypikilpi.

Hans-Lenze-Straße 1 31855 Aerzen GERMANY				DE	D	S
Lenze				Hz	100	58
EAC UL File No. E210321				kW	38.4	22.7
3~MOT	Type: MDFQABS112-22U			HP	51.5	30.4
I.CL. F	IP 23s	Ta 40°C	KTY/TKO	RPM	2935	1670
Brake	24 V	A	170 Nm	V	360	360
Feedback RS1				A	81.9	49.1
MAT-No	MOT-No			cos φ	0.83	0.85
13502078	10000174099659			C86	41	33

Kuva 16. Tarjotinhissin moottorin tyypikilpi.

Tästä voidaan laskea näytteenottonopeuden vaatimus em. esimerkkivikojen pohjalta. Esimerkin moottorin nimellisyörimistaajuus n on 48,9 Hz. Tällöin kerrannaistaajuudet ovat

- $2 \times n = 97,8 \text{ Hz}$
- $3 \times n = 195,7 \text{ Hz}$
- $4 \times n = 391,4 \text{ Hz}$
- $5 \times n = 782,7 \text{ Hz}$
- $6 \times n = 1565 \text{ Hz}$.

Nyquistin teoreeman (ks. alaluku 3.3) mukaan näytteenottotaajuuden pitää olla kaksi kertaa suurempi kuin korkeimman esitettävän taajuuden, eli esimerkin vähimmäisvaatimus olisi 3130,7 Hz. Tämän työn prototyyppiin ja testimittauksiin ADXL345:n näytteenottotaajuus 3200 Hz katsottiin riittäväksi kohteesta riippuen yleisistä linjauksista, asennus ym. ongelmista aina pyörivien laakerien valvontaan.

ADXL345 tärkeimmät ominaisuudet ovat seuraavat:

- Amplitudialue [G] (valittavissa) +- 2, +-4, +-8, +-16.
- Näytteenottotaajuus 3200 Hz.
- Resoluutio 13 bittiä.
- Väylät I2C, SPI [13].

Värähtelymittauksen käynnistys

Toimintaympäristön laitteista useat liikkuvat turva-alueella tavalla, joka tekee mittauksen käynnistämisestä haasteellisen, koska laitteen läheisyydessä ei pysty ollen laitteen ajaessa täydellä nopeudella. Mittauksen käynnistykseen tuli tehdä vaihtoehto etä- tai automaattikäynnistykseksi, niiden tilanteiden varalle, jolloin mittauksista ei itse voida käynnistää. Etäkäynnistykseen vaihtoehtotekniikoina olivat viive, Bluetooth, Direct Wifi tai automaattinen liipaisu käyttäen kiihtyvyyssanturin liikkeentunnistusta.

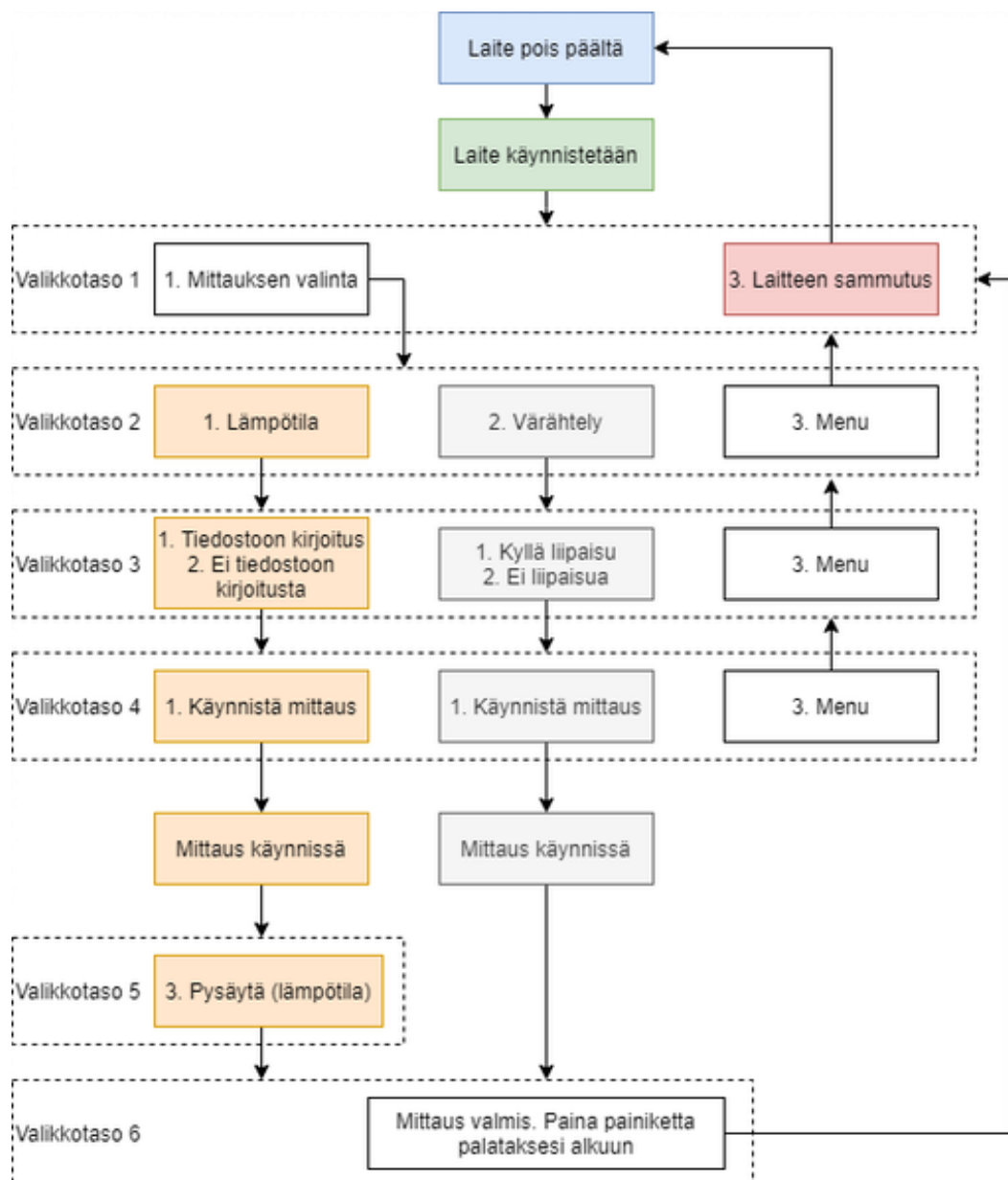
Päätettiin toteuttaa automaattinen liipaisu, koska sitä käyttämällä eliminoitaisiin käynnistykseen ihmisvaikutus toistettavuuden parantamiseksi.

4.4 Tiedonsiirto ja ajan päivitys

Toimintaympäristön tuoma haaste oli internetyhteyden puute, joten kellonaikaa ei pysty päivittämään eikä tiedonsiirtoa toteuttamaan verkon yli. Raspberry Pi-mikrotietokoneesta puuttuu paristovarmenteinen kellonajan säilytys. Kellonaika päivitetään aina kun internet yhteys on saatavilla. Raspberry Pi:hin tuli yhdistää paristovarmenteinen reaaliaikakellomoduuli, jolloin kello ja kalenteri pysyvät ajan tasalla, kun virta on katkaistu. Toiminto oli tarpeen varsinkin mittausten aikaleimojen takia. Reaaliaikakelloksi valittiin DS1307, joka käyttää I2C-väylää.

4.5 Käyttöliittymä

Käyttöliittymä suunniteltiin toteutettavaksi kolmella painonapilla ja 16x2 -kokoisella LCD-näytöllä, johon tulostetaan valikon vaihtoehdot. Ensimmäinen ja toinen painike ovat valikkorakenteessa eteenpäin liikkumista varten. Kolmas painike on cancel- tai menupainike, jolla valikkorakenteessa liikutaan taaksepäin. Kuvan 17 kaavio esittelee käyttöliittymän toiminnan, jonka mukaan myös pääohjelman rakenne tehtiin.



Kuva 17. Käyttöliittymän toimintakaavio.

4.6 Komponenttilista

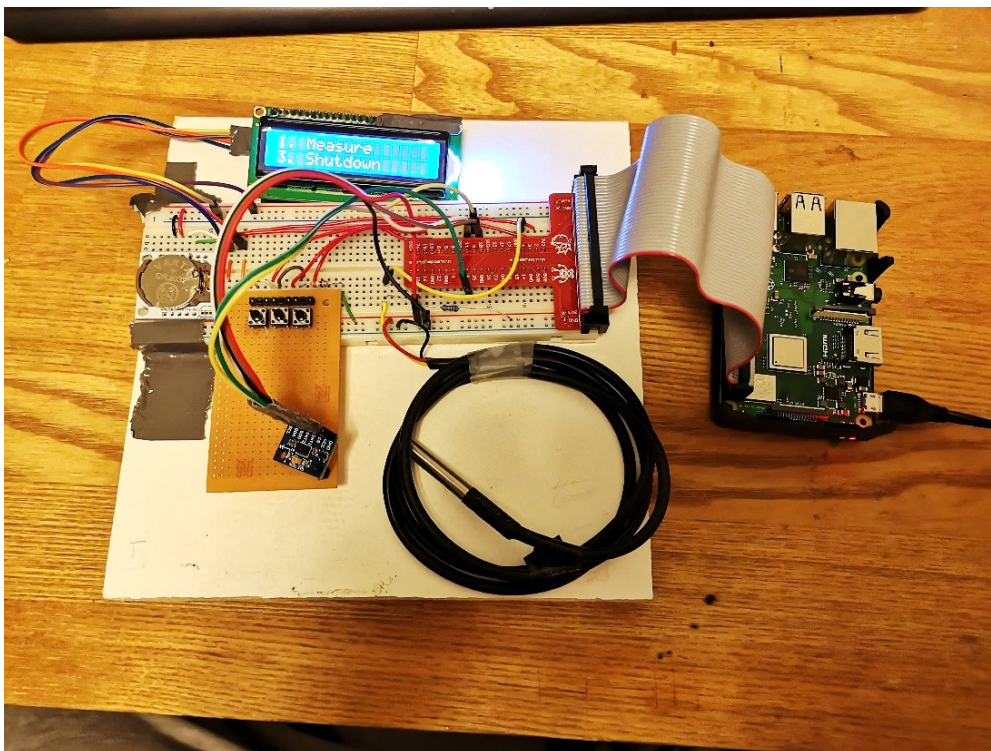
Taulukossa 3 on listattu suunnitelman mukaan tilatut komponentit.

Taulukko 3. Valitut komponentit ja niiden käyttämät väylät.

Komponentti	Malli	Väylä
Mikrotietokone	Raspberry Pi 3 B+	SPI, I2C, UART, 1wire
Reaaliaikakello	DS 1307	I2C
Kiihtyvyyssanturi	ADXL-345	I2C, SPI
Lämpötila-anturi	DS18B20	1wire
16x2 LCD-näyttö	HJ1602A	
LCD näytön I2C muuntaja	PCF8574T	I2C
Varavirtalähde	Anker PowerCore - varavirtalähde, 10400 mAh, musta	

5 Mittalaitteen toteutus

Komponenttien saavuttua ne testattiin ensin yksitellen, jonka jälkeen ne kytkettiin kuvan 18 mukaisesti koekytkentälevyn avulla Raspberry Pihin. Toimintojen ohjelmoinnin ja värähtelymittauksen testausten jälkeen laitteelle suunniteltiin kotelointi Solidworks-ohjelmistolla. Kotelo tulostettiin Anet A8 3D-tulostimella.



Kuva 18. Laitteen pääkomponentit.

5.1 Kytkennät ja kotelointi

Kytkeäntöjen tekeminen varsinkin GPIO:n osalta oli haastavaa tilan puutteen takia. Kytkentäjohtojen piti lähteä 90 asteen kulmassa GPIO:lta. Juottamisen apuna käytettiin nauhakuparoitua protopiirilevyä, joka helpotti prosessia. Anturien liittämiseen käytettiin 7-pinnisiä MIC-liittimiä. Liitteen 11 piirikaaviosta näkee, miten komponentit kytkettiin Raspberry Pihin. Kuvassa 19 näkyy mittalaitteen prototyyppi ennen sulkemista.



Kuva 19. Mittauslaite ennen kotelon sulkemista.

Koteloinnin prototyyppi suunniteltiin solidworks-ohjelmistolla, Koteloon suunniteltiin aukot tarpeellisille Raspberry Pin liitännöille. Kotelo tulostettiin 3D-printterillä PLA-muovista, jonka käyttö kestopuolesta rajoittuu prototyyppihin ja koriste-esineisiin. PLA-muovia on helppo työstää, koska se sulaa alhaisessa lämpötilassa, ei tuota käytännössä tuulettamista vaativia myrkyllisiä kaasuja ja tarttuu alustaan helposti. 3D-printtereillä pystyy tulostamaan myös esimerkiksi kestäväällä ABS-muovilla, mutta tällöin tulee käyttää koteloitinta ja tuuletusjärjestelmää, jota ei ollut käytettävissä tämän työn puitteissa.

Anturin kiinnitys ja koteloitinta on kriittinen osa kiihtyvyyssmittausta. Anturille lähdettiin suunnittelemaan metallista koteloitinta, mutta tämän työn puitteissa suunnitelmasta piti luopua, koska tarvittavia resursseja ei ollut käytettävissä. Kenttätestiä varten kiihtyvyyssanturiin tehtiin metallinen sovitepala, jotta anturi pystyttiin kiinnittämään mahdollisimman jyrkästi mitattavaan pintaan. Anturille ei tehty täydellistä koteloitinta, vaan sen kytkentöjen suojaus sekä vedonpoisto toteutettiin teippauksin koneruuveja hyödyntäen. Kuvassa 20 on kiihtyvyyssanturi kytkettynä mittalaitteeseen.



Kuva 20. ADXL345 kiihtyvyyssanturi kiinnitettynä mittalaitteeseen.

5.2 Mittalaitteen toimintakuvaus

Laite käynnistetään kytkemällä virtalähteen USB-johto laitteeseen. Laite käynnistyy alkuvalikkoon odottamaan käyttäjän syötettä. Käyttäjä navigoi laitteen valikossa kolmella painonapilla. Ensimmäinen ja toinen painonappi on valikkorakenteessa eteenpäin liikkumista varten. Kolmas painonappi on valikkorakenteessa taaksepäin liikkumista ja laitteen sammuttamista varten. Valikkorakenne on kuvattu luvussa 4.5.

Lämpötilamittauksessa lämpötila-anturin tieto tulostetaan LCD-näytölle. Käyttäjä valitsee, tulostetaanko data tiedostoon vai ei. Lämpötilamittaus lopetetaan painamalla painiketta "3" pohjassa.

Värähtelymittauksessa käyttäjä valitsee mitattavan kohteen mukaisen mittauksen valinnoista, jotta mittauksen kesto ja näytteenottoaajuus on sopiva. Käyttäjä valitsee, käynnistetäänkö mittaus välittömästi, vai käytetäänkö automaattista liipaisua. Mittauksen jälkeen laite kirjoittaa ensin datan tiedostoon, jonka jälkeen ilmoitetaan mittauksen valmistumisesta ja jäädään odottamaan painonapin painallusta. Painalluksen saatua palataan alkunäyttöön.

Laite sammutetaan painamalla alkunäytössä painiketta "3" pohjassa kolme sekuntia. Kun laite on sammunut, laitteen virtajohdon irrottaminen on turvallista.

5.3 Ohjelmointi

Ohjelmointi tehtiin pääasiassa Python-ohjelmointikielellä. Lopullisessa versiossa ainoastaan kiihtyvyydsmittaus tapahtuu C-ohjelmointikielellä. Ohjelman käynnistäminen toteutettiin niin, että Raspberry Pi käynnistyy komentoliittymään ja mittausohjelma käynnistetään samalla rinnalle. Graafinen käyttöliittymä poistettiin käytöstä. Ohjelmointiympäristöinä käytettiin Raspberry Pin komentokehoitteen Nano-ympäristöä sekä graafisia THONNY IDEä sekä Geanya.

Pääohjelma sisälsi valikkorakenteen, joka tehtiin luvun 4.5 kuvan 17 kaavion mukaan. Valikkorakenteen eri tasoilla liikuttiin painonappien syötteellä. Komponenttien tiedonvaihtoon käytetyt kirjastot sijaitsevat omissa tiedostoissa, joiden metodeja kutsutaan pääohjelmasta. Kiihtyvyydsmittauksesta tehtiin "executable" eli käynnistettävä tiedosto, jota kutsutaan valinnaisilla parametreilla pääohjelmasta. Näytön ohjaamiseen käytetty ohjelmakoodi löytyy liitteestä 1. Lämpötila-anturin lukemiseen käytetty ohjelmakoodi löytyy liitteestä 2 [25; 26].

5.3.1 Värähtelymittaus

Tässä luvussa käydään läpi havaintoja ja kokeiluja, joita värähtelymittausta ohjelmoidessa tehtiin.

ADXL345-kiihtyvyyssanturin ohjelmointi aloitettiin Python-ohjelmointikielellä käyttäen I2C-väylää, joka toimii kahdella johtimella. Tämä väylä valittiin sen takia, että muutkin komponentit käyttävät sitä ja sen käyttöönottaminen on helppoa. Järjestelmä tunnistaa automaattisesti väylään kytketyt laitteet ja niiden osoitteet. [20.] Kuvassa 21 näkyy esimerkki I2C osoitevaruudesta, jossa osoite 27 on varattu komponentille.

```
pi@raspberrypi:~ $ i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  27  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi:~ $
```

Kuva 21. I2C osoitevaruus.

Github-sivustolta löytyi ohjelmakirjasto, jolla anturin toimintaa pystyi testaamaan. Anturin todellisen näytteenottonopeuden selvittämiseksi kirjoitettiin ohjelma, jossa anturilta haetaan mittatieto ja tallennetaan se tiedostoon. Tämä toistettiin ohjelmassa 1000-kertaa näytteenottotaajuuden keskiarvon selvittämiseksi. Anturin nopeimmalla näytteenottotaajuudella mittauksen kesto oli 2,512 sekuntia. Todellinen näytteenottotaajuus oli siis 464,7 Hz. Näytteenottotaajuutta pystyi nostamaan tästä hieman kirjoittamalla mittausdata ensin tauluun ja tekemällä tiedostoon kirjoituksen koodissa jälkeinpäin, mutta ero näillä kahdella ei ole merkittävä. I2C-väylänopeus 100 kb/s, rajoitti näytteenottonopeuden liian alhaiseksi. [20.] Tämän perusteella hylättiin I2C-väylän käyttäminen kiihtyvyyssanturin kanssa.

Lähdettiin toteuttamaan mittausta SPI-väylällä, jonka nopeus on jopa yli 50 MHz. SPI-väylän heikkoutena mainitaan, että siihen ei voi kytkeä useita laitteita samanaikaisesti I2C-väylän tapaan ja kahden johtimen sijaan käytetään neljää johdinta. [21.] Liitteessä 3

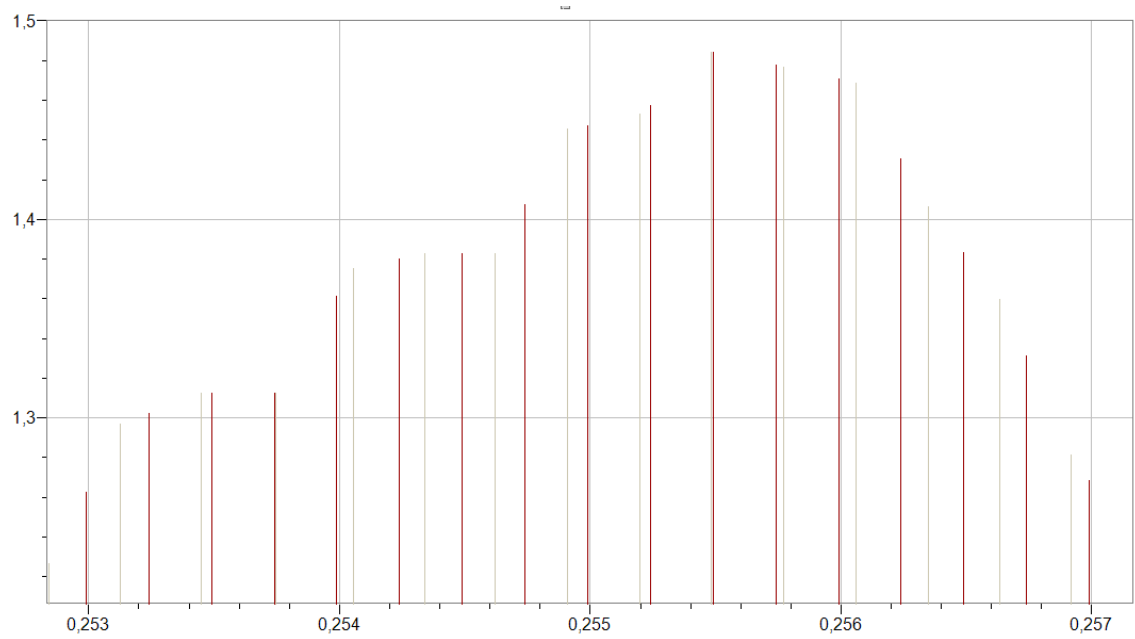
olevalla Python-koodilla kiihtyvyyssmittaus onnistuu, mutta mittauksen näytteenoton tasaisuuteen liittyy ongelmallista epätarkkuutta, joka kertautui suhteessa mittauksen pituuteen. Epätarkkuus syntyy siitä, että ohjelmakierron nopeus vaihtelee, jolloin aikaleimat eivät ole tasaisin välein. Tämän seurauksena ensinnäkin aikataason kuvaaja vääristyy. Toinen ongelma tulee vastaan taajuustason muunnoksessa. Jotta FFT-operaatio pystytään suorittamaan, tulee näytteiden kestojen olla tasapituisia. Jos tässä tapauksessa näytteitä kohdeltaisiin tasavälisinä, kuvaaja ja näin ollen sen taajuusvaste vääristyisi.

Kokeiltiin ratkaista ajoitusongelma interpoloimalla data. Interpoloimalla voidaan tietyn datajoukon perusteella tuottaa uusi datajoukko muuttaen näytteiden aikaintervallia. Kuvassa 22 näkyy LibreOfficen Calc-ohjelmistolla tehty taulukko, joka tehtiin hyödyntäen ulkoisia laajenteita. [14; 15.]

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	data		--	interpolaaatio		--fft (ctrl+shift+enter)		-- spektri						
2	t	g	smpl#	t	g	real	img	freq	imabs					
3	0.000590093	-1.484375	1	0	-1.499340773	-849.37723	0	0	849.37723					
4	0.000898136	-1.4765625	2	0.000249888	-1.493000665	-362.946076	103.9698433	1.65092596	258.9762326					
5	0.001184249	-1.484375	3	0.000499976	-1.486660527	-198.6634124	22.01460392	3.30185192	176.6488085					
6	0.001466665	-1.515625	4	0.000749964	-1.480320404	-204.177426	4.966865209	4.95277788	199.2105608					
7	0.00175007	-1.515625	5	0.000999952	-1.479342641	-78.11654173	518.930233	6.60370384	440.8136913					
8	0.002036288	-1.53125	6	0.00124994	-1.49164381	-852.8590165	431.2116041	8.254629799	421.6474123					
9	0.002315995	-1.5078125	7	0.001499927	-1.515625	-1397.998744	-788.7138129	9.90555759	2186.712557					
10	0.002597682	-1.484375	8	0.001749915	-1.515625	-3585.208764	600.4737133	11.55648172	2984.73505					
11	0.002879368	-1.5859375	9	0.001999903	-1.529263708	2370.085225	-741.8635396	13.20740768	1628.221685					
12	0.003164023	-1.5390625	10	0.002249881	-1.513351546	554.3850323	516.0074558	14.85833364	1070.392488					
13	0.003443939	-1.5859375	11	0.002499879	-1.492512605	682.0068637	198.8524432	16.5092596	880.8593069					
14	0.003724583	-1.546875	12	0.002749867	-1.539245599	434.0694114	-165.5289295	18.16018556	268.5404818					
15	0.004005332	-1.5703125	13	0.002999855	-1.566096573	252.2581867	120.5283559	19.81111152	372.7865426					
16	0.004288425	-1.5703125	14	0.003249843	-1.55343395	233.3474878	-58.65359966	21.46203748	174.6938881					
17	0.004567455	-1.578125	15	0.003499831	-1.578158014	-73.85883177	92.11221253	23.11296344	18.25338076					
18	0.004846277	-1.6171875	16	0.003749819	-1.548981712	901.2009469	367.0684367	24.7638894	1268.269384					
19	0.00512765	-1.65625	17	0.003999806	-1.568951212	94.52671899	327.9904332	26.41481536	422.5171522					
20	0.00541017	-1.6484375	18	0.004249794	-1.5703125	327.0620969	327.713265	28.06574132	654.7753619					
21	0.00569045	-1.671875	19	0.004499782	-1.576230244	245.8019426	39.93286078	29.71666728	285.7348034					
22	0.00596948	-1.640625	20	0.00474977	-1.603667045	-13.60394897	202.705685	31.36759324	189.101736					
23	0.006249552	-1.6875	21	0.004999758	-1.638494989	145.9417785	62.94050237	33.0185192	208.8822809					
24	0.006529884	-1.6875	22	0.005249746	-1.652873693	19.04272602	208.8884709	34.66944516	227.9311969					
25	0.006809279	-1.6875	23	0.005499734	-1.655926981	122.30197	214.802733	36.32037112	337.104703					
26	0.007087006	-1.6953125	24	0.005749722	-1.665236857	-111.596274	176.0129601	37.97129708	64.41668615					
27	0.007368172	-1.75	25	0.00599971	-1.64568446	-24.01129968	129.3304291	39.62222304	105.3191294					
28	0.007647775	-1.765625	26	0.006249698	-1.6875	-15.28145918	72.08436236	41.273149	56.80290318					

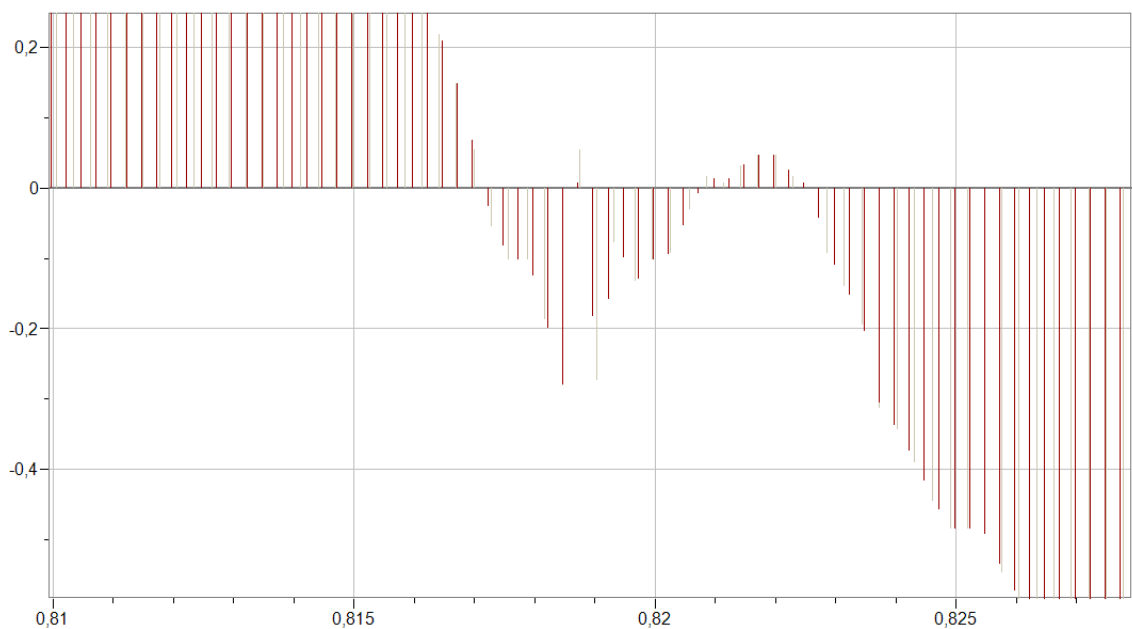
Kuva 22. Datan interpolointi ennen FFT operaatiota.

Taulukossa vasemmalla näkyy alkuperäinen data, jolle suoritettiin interpoloinnin jälkeen FFT-operaatio. Tällä tavalla interpoloimalla pystytään tekemään FFT-operaatio datalle, joka ei alun perin ole tasavälein näytteistettyä. Kuvassa 23 on esitetty kuvaaja esimerkkitiedon interpoloinnista. Kuvaajan harmaat pylväät esittävät raakadataa ja punaiset pylväät tasavälein interpoloitua dataa.



Kuva 23. Interpoloinnin vaikutus signaaliin.

Kuvasta näkee, että lineaarinen interpolointi seuraa suurimmilta osin hyvin raakadatan taajuus- ja aikatietaa. Interpolointi aiheuttaa jonkin verran häiriötä tulokseen. Kuvassa 24 näkyy esimerkkidatan kohta, jossa tietoa häviää interpoloinnin seurauksena.



Kuva 24. Interpoloinnilla voidaan menettää tietoa.

Interpoloinnilla toteutus tuntui kompromissilta ja se aiheutti ylimääräisen toimenpiteen datan analysoinnissa, joten tutkittiin vielä vaihtoehtoja tälle ratkaisulle.

Optimaalisin ratkaisu oli toteuttaa kiihtyvyyssmittaus C-ohjelmointikielellä tehdyllä mittausohjelmalla, joka käynnistetään Python-pääohjelmasta. Liitteessä 4 oleva koodi on optimoitu tiedostoon kirjoitukseen anturin suurilla näytteenotonopeuksilla. Esimerkkikoodissa 1 pääohjelmasta käynnistetään kiihtyvyyssmittaus `os.system` -komennolla.

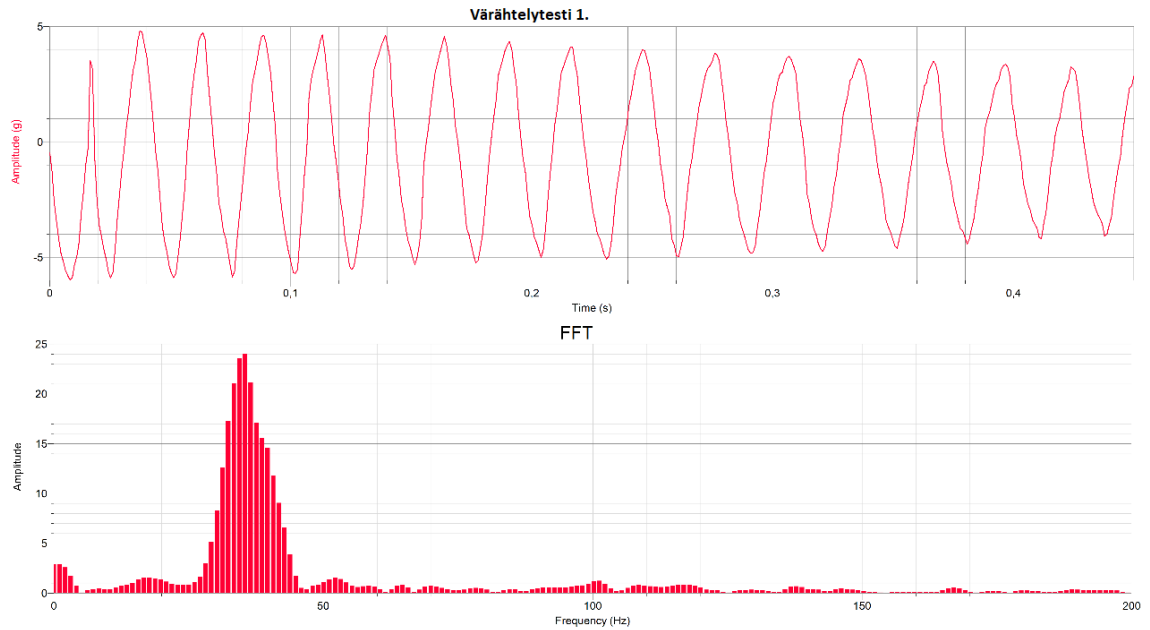
```
if GPIO.input(35) == GPIO.HIGH: #Älä käytä triggeriä

    sleep(1)
    lcd("Measuring          ",LCD_LINE_1)
    lcd("Vibration...        ",LCD_LINE_2)
    os.system ("sudo /home/pi/Desktop/Mittalaite/adxl345spi -s 1 -t 1 -f
3200")

    lcd("Finished! press ",LCD_LINE_1)
    lcd("a key to go menu",LCD_LINE_2)
```

Esimerkkikoodi 1. Kiihtyvyyssmittauksen käynnistys.

Kiihtyvyyssmittauksen dataa haluttiin tarkastella taajuustasossa. Data muutettiin aikatasosta taajuustasoon FFT-muunnoksella. Anturin datan FFT-muunnos todettiin toimivaksi mittaamalla mm. soittimen kielen värähtelyä ja tarkastelemalla, kuinka hyvin taajuusvaste vastaa kielen todellista taajuutta, joka mitattiin TC Polytune 2 -virittimellä. Kuvassa 25 näkyy värähtelytestin aika- ja taajuustason kuvaajat, joiden perusteella FFT-muunnos todettiin toimivaksi.



Kuva 25. Värähtelytestissä mitattiin kielen värähtelyä.

5.3.2 Automaattiliipaisu

Useimmat laitoksen tärkeimmistä kohteista sijaitsevat turva-alueilla, jonne ei ole pääsyä laitteiden ollessa automaattitilassa. Tämän takia mittaus piti pystyä käynnistämään etänä. Erilaisista tavoista pohdinnassa olivat

- aikaviive
- Bluetooth -etäkäynnistys
- Direct Wifi -etäkäynnistys
- liipaisu havaittaessa liikettä.

Näistä keinoista liipaisu anturin liikkeestä oli käytännöllisin ja ainoa, jossa ihmistekijän saa poistettua toistettavan tuloksen saavuttamiseksi. Tätä varten muokattiin C-ohjelmasta versio, jossa laite jää odottamaan ja käynnistää mittauksen liikettä havaitessa.

5.4 Tiedonsiirto ja etäyhteys

Laitteesta siirrettiin tiedostoja ja muuta tietoa ulkoisiin laitteisiin käyttäen Ethernet-yhteyttä ja SSH:ta eli salattuun tietoliikenteeseen tarkoitettua protokollaa, jolla saadaan yhteys Raspberry Pin komentokehoitteeseen. Komentokehoitteesta voidaan myös käynnistää virtuaalinen etätyöpöytäsessio (VNC) vncserver palvelulla, jos ulkoisella laitteella halutaan käyttää Raspberry Pin graafista työpöytää. [12.]

6 Kenttätesti

6.1 Kohde

Inex PTDC (Päivittäistavara Distribution Center) -logistiikkakeskus sisältää erilliset osastot kuivatavara-, hedelmä ja vihannes-, liha ja kala- ja pakastetuotteille, joissa keräily tapahtuu automaattisesti. Työn mittaukset tehtiin hedelmä ja vihannes osastolla, jossa on 26 kappaletta kuvan 26 mukaisia tarjotinvarastokäytäviä.



Kuva 26. Tarjotinvarasto ja tarjotinhissi.

Tarjotinhissit siirtävät tarjottimille purettuja tuotteita varastoon ja pois sieltä. Tarjotinvarastot syöttävät automaattisia ja puoliautomaattisia keräilyjärjestelmiä, joissa tarjotinvarastossa olevat tuotteet kerätään lavoille ja rullakoihin, joten niiden suunnittelemattomasta pysäytyksestä aiheutuu häiriötä tuotannolle.

Tarjotinhissien komponentit ja mitat ovat samoja, joten tämän perusteella voisi olettaa samansuuntaisia värähtelyn tasoja. Mitatut tarjotinhissit (engl. Tray Crane) olivat

- TCRA002
- TCRA003
- TCRA004
- TCRA005
- TCRA006
- TCRA007

6.2 Suunnitelma

Kenttätestillä testattiin laitteen toimintaa ja vastattiin kysymykseen: Pystyykö kiihtyvyyksmittauksella tuottamaan värähtelytason tunnuslukuja tarjotinhisseiltä? Ennakkoon tarjotinhissi tiedettiin haastavaksi ja mielenkiintoiseksi mittauskohteeksi, koska sen on pakko liikkua varastokäytävällä mittauksen aikana, josta aiheutuu häiriötä mittaukseen. Suunnitelmana oli kiinnittää mittalaite ja anturi tarjotinhissiin, jonka jälkeen liikkuvalla tarjotinhissille suoritettaisiin värähtelymittaus mitta-ajon tasaisen nopeuden aikana. Erityisesti kiinnosti se, pystyykö kohteen useiden pyörivien komponenttien eri pyörimistaajuuksia erottamaan taajuustasossa toisistaan.

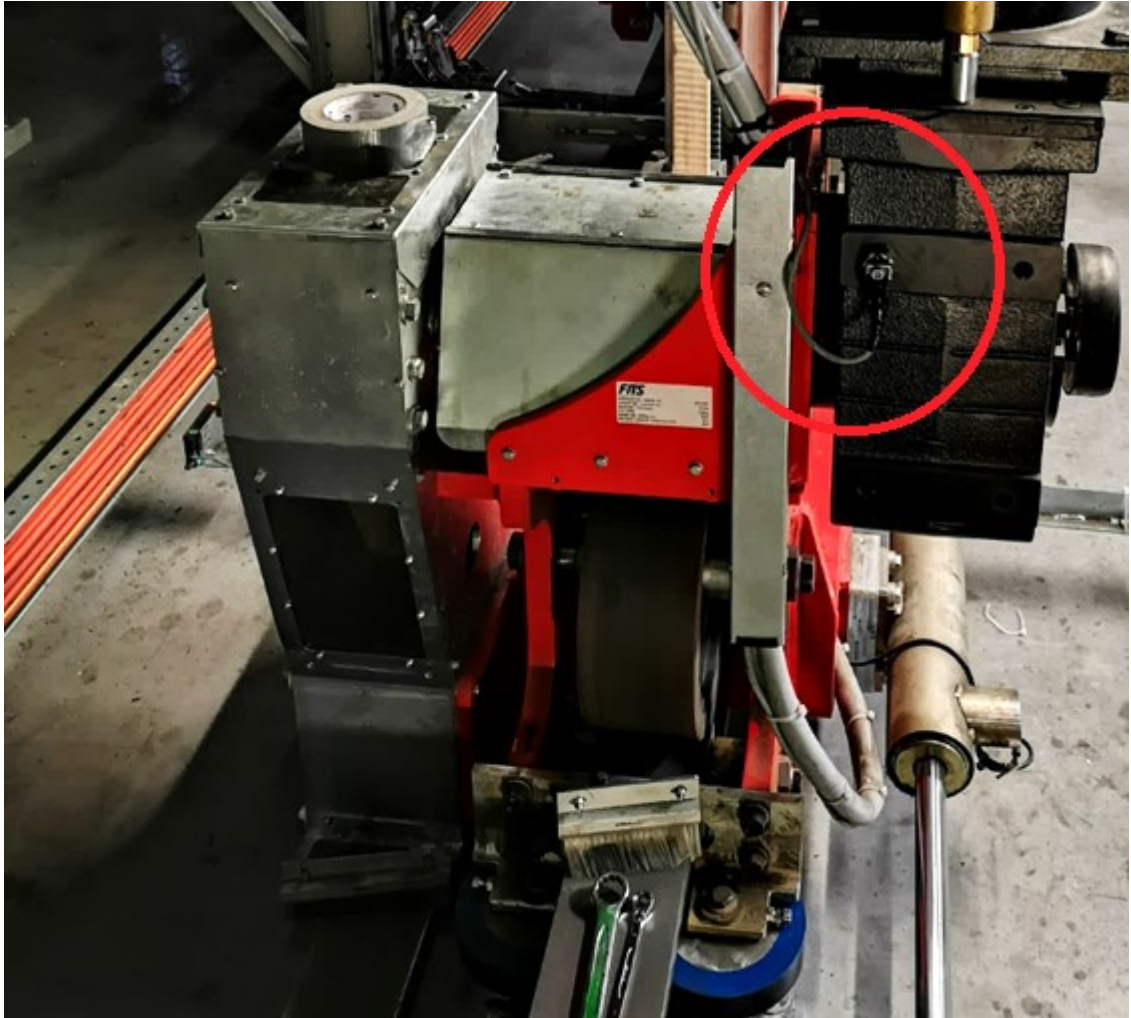
6.3 Mittapisteen valinta

Tarjotinvarastoissa oli ajankohtaisesti tapahtunut omegapyörien rikkoutumisia, joten mittapisteen valinnassa tutkittiin ensin omegapyörän kunnonvalvonnan mahdollisuutta. [27.] Kuvasta 27 nähdään, että pyörien laakereiden säteissuuntainen mittaaminen ei tulisi kyseeseen. Akselisuuntainen anturin kiinnitys vaatisi mittatilauksena valmistettavat kiinnikkeet.



Kuva 27. Omegapyörän rakenne [16, s. 16 muokattu]-

Tarjotinhississä ei juurikaan ole valmiina kiinnitysmahdollisuuksia antureille. Mittalaitteen prototyypin testimittausta varten kiinnityspisteeksi valittiin moottorin vaihdelaatikossa oleva kierreikä. Vaihdelaatikko oli ainoa kiinnityspisteitä sisältävä komponentti, johon anturi oli mahdollista kiinnittää tämän työn puitteissa. Anturin kiinnittämistä varten sopivan kokoiseen isoon pulttiin hitsattiin 5 mm:n kierre toiseen suuntaan anturin kiinnitystä varten. Mittauspisteeseen välittyi mm. moottorin, vaihteen, omegapyörän sekä hissien liikkeen ja muiden osien aiheuttamia värähtelyitä. Anturin kiinnitys on esitetty kuvassa 28.



Kuva 28. Anturin kiinnitys.

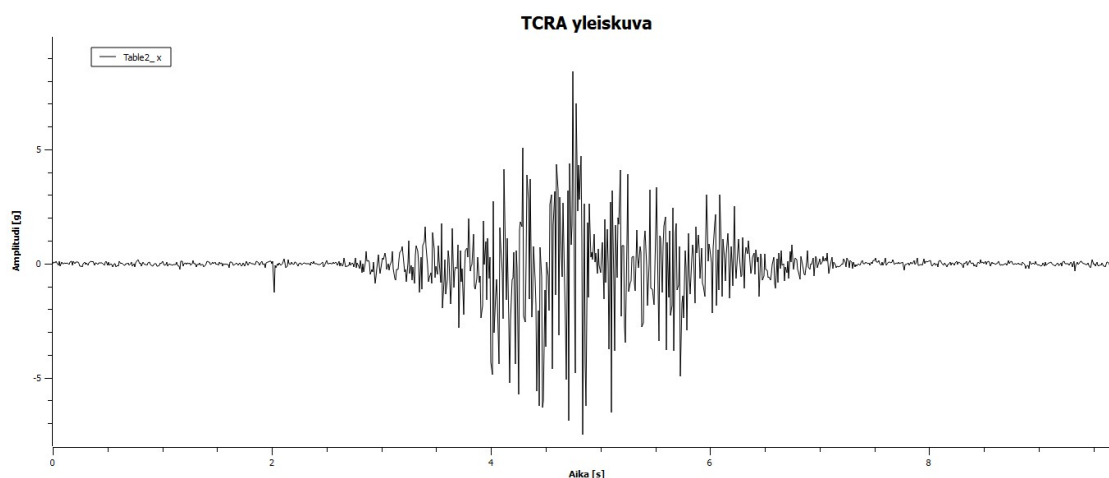
6.4 Mittaukset

Kenttätestin mittaukset suoritettiin huoltoseisokkiyönä. Tarkoituksena oli testata ja saada aikaan toimiva mittaustapa ensin yhdelle tarjotinhissille, jonka jälkeen kaikille kuudelle tarjotinhissille tehtäisiin mittaus samalla tavalla.

Mittauksessa käytettiin manuaaliajtoa. Se oli ainut ajotila, jolla mittauksen pystyi toteuttamaan. Tarjotinhissin eri ajotilat ovat:

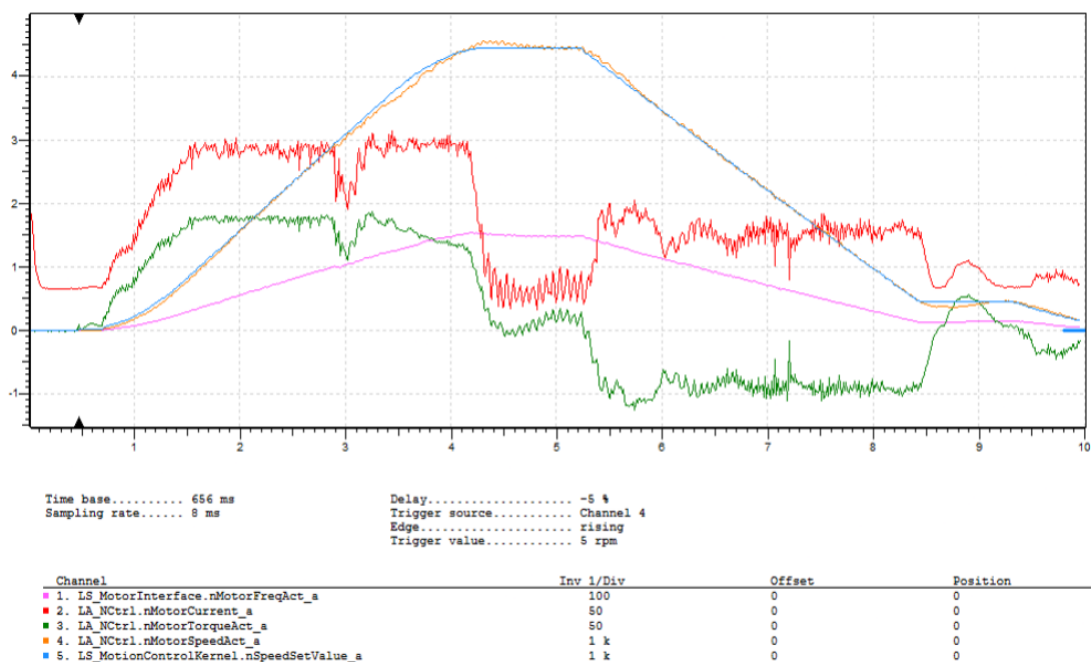
1. *Automaatti*: Laite täysin automaattisessa tilassa.
2. *Manuaali*: Laitteelle annetaan liikekäsky, laite suorittaa liikkeen, jos logiikan ehdot täyttyvät. Pystyy ajamaan täydellä nopeudella käytävän päästä päähän.
3. *Huolto*: Ajaa erittäin hitaalla nopeudella. Käytetään turva-alueen sisältä ajamiseen.

Mitta-ajossa tarjotinhissi ajettiin manuaaliajolla käytävän toisesta päästä toiseen. Käytännössä laitteelle annetaan ajokäsky, jolloin laite suorittaa automaattisesti liikkeen. Ensimmäisellä mittauskerralla tehtiin pitkäkestoinen mittaus alhaisella näytteenottotaajuudella yleiskuvan saamiseksi. Huomattiin, että anturi havaitsee moottorin magnetoitumisen. Kaikkein selkeimmin tämä oli nähtävissä x-akselilta, joka oli 90 asteen kulmassa mittausten kohteena olevaan akseliin nähden. Kuvassa 29 kahden sekunnin kohdalla näkyvä negatiivinen piikki johtuu moottorin magnetoitumisesta. Tätä kiihtyvyyksiä käytettiin mittausten automaattisessa liipaisussa.



Kuva 29. Mitta-ajon värähtelyn yleiskuva SciDAVis-ohjelmistossa.

Mittauksesta heräsi epäily, että mitta-ajossa ei olisi hetkeä, jolloin pyörimistaajuus pysyisi samana, eli liike olisi pääosin kiihtyvää tai hidastuvaa. Tämä varmistettiin vielä tallentamalla taajuusmuuttajan tiedot ajon ajalta. Kuvan 30 keltainen viiva kuvaa moottorin todellista pyörimisnopeutta.



Kuva 30. Mitta-ajon suureet taajuusmuuttajalta.

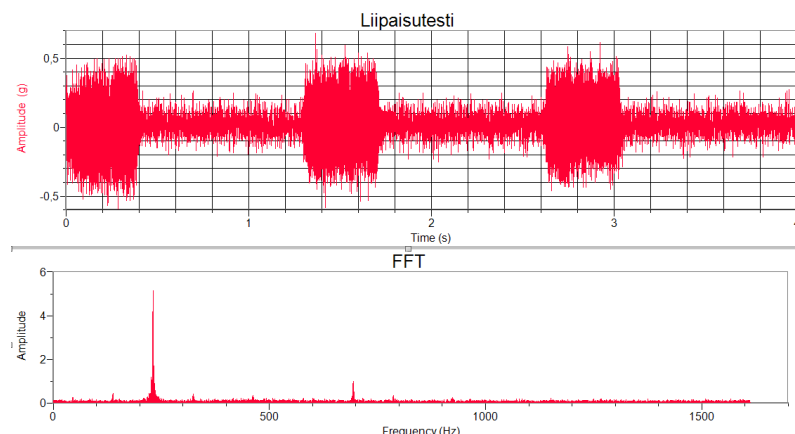
Pyörimistaajuuden puute oli tärkeä havainto, sillä se tarkoitti, että taajuustason valvontakeinot eivät olisi käytettävissä tällä mittaustavalla.

Tarjotinhissi ajaa mitta-ajossa 22 metriä pitkän tarjotinvaraston käytävän päästä päähän noin 6 sekunnissa. Liike on ensin kiihtyvä ja lähes heti hidastuva. Ajon keskivaiheessa on noin puolen sekunnin vaihe, jolloin taajuusmuuttaja ohjaa nopeutta vakioksi, mutta todellinen pyörimisnopeus ei tasaannu. Kokeiltiin myös hitaampaa ”huolto” -ajoa, jolloin hissiä ajetaan käsiajolla hitaalla nopeudella. Tässä ajossa kierrosnopeudet olivat tasaiset, mutta niin alhaiset, että hyödyllistä tietoa ei saatu.

Edellä mainittujen seikkojen perusteella todettiin, että spektrianalyysiä, taajuustaso- ja taajuustason tunnuslukuvalvontaa ei pystyisi toteuttamaan, koska ne pystyttäisiin toteuttamaan vain, jos pyörimistaajuus olisi tiedossa. Samasta syystä myöskään tehollisarvoa (rms) ei pystytty selvittämään aikatasosta.

Alkuperäisen ajatuksen mukaan mittaus olisi tehty tasaisen nopeuden vaiheesta. Sen sijaan toteutettiin toistettava mittaustapa, jossa mitataan kaikki liikkeen vaiheet sisältäen liikkeelle lähdön, kiihdytyksen, hidastuksen ja pysähtymisen. Ongelmaksi muodostui automaattinen liipaisu, joka oli omassa C-ohjelmätiedostossa, joka laukaistiin

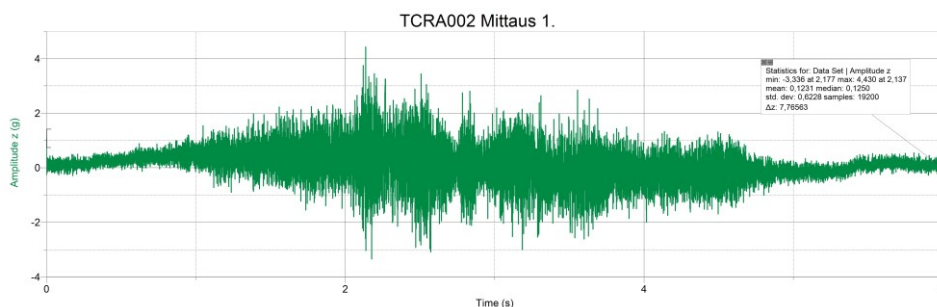
pääohjelmasta. Tästä toteutuksesta aiheutui viivettä mittauksen käynnistymiseen, jonka takia menetettiin liikkeellelähtö ja osa kiihdytyksestä. Ratkaisuna liipaisu ja itse mittaus yhdistettiin samaan C-ohjelmamedioseen. Näin viive saatiin käytännössä poistettua kokonaan. Kuvassa 31 näkyy parannellun liipaisun toiminnan viive.



Kuva 31. Parannellun automaattiliipaisun testin aika- ja taajuustason kuvaajat.

Testaus suoritettiin mittaamalla älypuhelimella generoitua värähtelyä. Mittaus alkaa nyt välittömästi, kun värähtelyn amplitudi ylittää asetetun raja-arvon.

Mittauksen käynnistävänä liipaisuarvoksi valittiin 400 mg. Mittausajaksi asetettiin kuusi sekuntia. Näytteenottotaajuus oli 3200 Hz. Toimivalla mittaustavalla mitattiin kaikki kuusi tarjotinhissiä. Jokaiselle tarjotinhissille tehtiin neljä peräkkäistä mittausta, joten kuvan 32 mukaisia kuvaajia tuotettiin yhteensä 24 kpl. Liitteissä 5-11 on nähtävissä kaikkien mittausten kuvaajat.

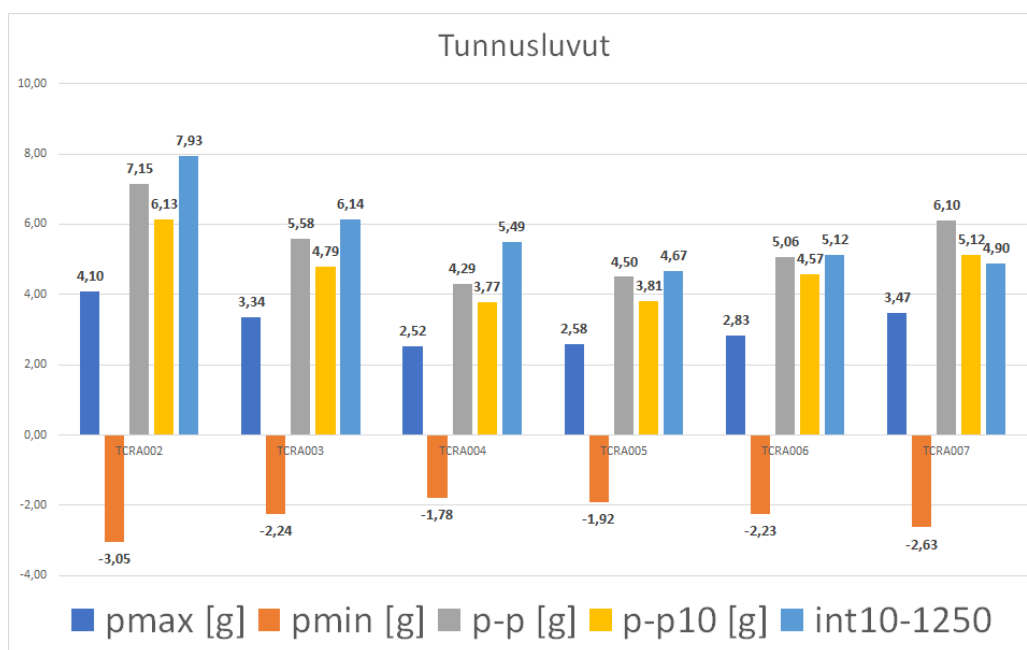


Kuva 32. TCRA002:n ensimmäisen mittauksen kuvaaja.

6.5 Tunnusluvut

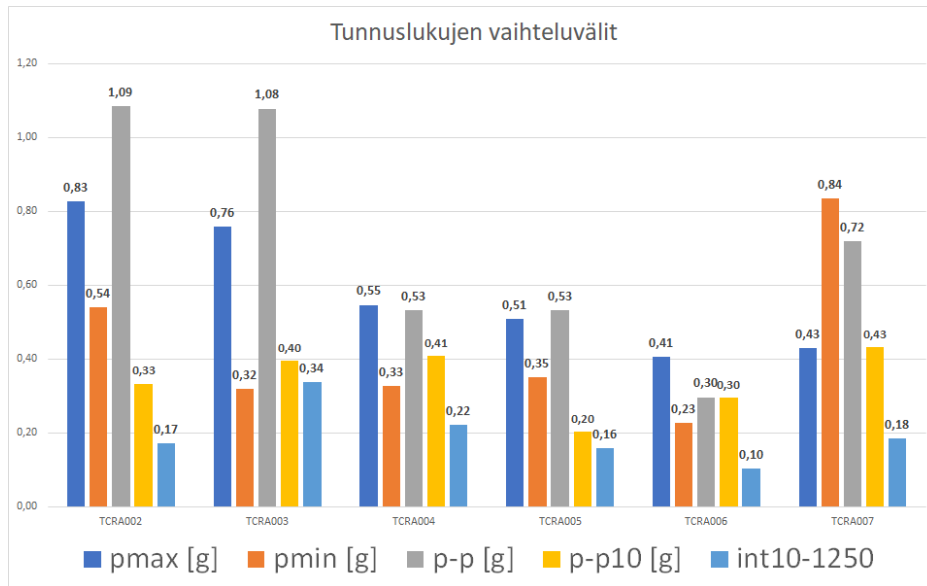
Mitattuja tarjotinhissejä oli kuusi. Niistä käytetään tunnuksia TCRA002, TCRA003, TCRA004, TCRA005, TCRA006 ja TCRA007. Jokaiselle tarjotinhissille tehtiin neljä mittausta vaihteluvälien tarkastelua ja keskiarvoistamista varten. Mittauksista tuotettiin kuvassa 33 näkyvät seuraavat tunnusluvut:

1. pmax: Mittauksen suurin huippu.
2. pmin: Mittauksen pienin huippu.
3. p-p: Suurimman ja pienimmän huipun erotus.
4. p-p10: 10 suurimman huipun keskiarvon ja 10 pienimmän huipun keskiarvon erotus.
5. int10-1250: Kuvaa mittauksen kokonaisenergiaa taajuvälillä 10 ... 1250 Hz. Skaalattu jakamalla luvulla 100.



Kuva 33. Värähtelymittausten tunnusluvut.

Tavallisesti huipusta huippuun arvoja voidaan käyttää luotettavasti kokonaisvärähtelyn arvioimiseen. Tässä tapauksessa huippuarvot eivät antaneet luotettavaa kuvaa kokonaisvärähtelytasosta, johtuen kohteen liikkumisesta, anturin kiinnityssuunnasta, mitta-ajon kiihtyvyydestä ja pitkän mittausajan aiheuttamasta satunnaisuudesta. Kuvassa 34 näkyy tunnuslukujen vaihteluvälit.



Kuva 34. Kaikkien tunnuslukujen vaihteluvälit.

Mittausdatasta luotiin taajuustasosta integroimalla oma kokeellinen tunnusluku, jonka tavoitteena oli kuvata mittauksen kokonaisvärähtelyä. Datalle tehtiin FFT eli nopea Fourier-muunnos, jonka jälkeen dataa pystyttiin tarkastelemaan taajuustasossa. Integroitava taajuusalue valittiin standardin PSK5721 määrittämisen perusteella:

"Aikatasosignaalin näytteenottotaajuuden tulee olla vähintään 2,56 kertaa mittausalueen ylärajataajuus. Esimerkiksi spektrin ylärajataajuuden ollessa 1000 Hz näytteenottotaajuus on 2560 Hz". [7, s. 7.]

Näin tarkasteltava ylärajataajuus käytetyllä näytteenottotaajuudella 3200 Hz on 1250 Hz. Tällä tavoin otettiin huomioon taajuusalueen ylärajaa lähestyttäessä lisääntyvä häiriö. Vaikka mittauksesta ei ollut mahdollista erottaa pyörimistaajuuksien komponentteja, oli taajuustasoa mahdollista käyttää näin hyödyksi. Liitteessä 12 on esimerkki mittausdatan integraalista Logger Prolla.

6.6 Tulos

Kenttätestien tuloksena saatiin testattua mittalaitetta ja kehitettyä tarjotinhisseille mittaustapa. Pyörimistaajuuden määrittäminen oli mittausten isoin ongelma. Mittauspisteeseen tulevia useiden eri komponenttien värähtelytaajuuksia ei saatu eroteltua toisistaan, joten taajuustason valvontakeinoja ei voitu kokeilla.

Mittauksesta tuotettiin aikataulun tunnuslukuja, mutta mittausta ei voi suositella tästä pisteestä suoritettavaksi, jos pyörimistaajuutta ei pystytä määrittämään. Syynä on se, että värähtelyn komponenttien erottelun puuttuessa värähtelytason muutosta ei pysty kohdistamaan varmuudella mihinkään komponenttiin, jolloin mittauksen hyödyllisyys täytyy kyseenalaistaa.

Kenttätestin perusteella tilaajalle suositeltiin tarjotinhissien mittauksien jatkotoimenpiteeksi:

1. Tehdään anturikiinnitykset mitattaviin komponentteihin laakerin kohdalle säteissuunnassa.
2. Mittauksen satunnaisuus voidaan minimoida anturin kiinnityssuunnalla (90 astetta liikkeeseen nähden).
3. Tarjotinhissin PLC:n hyödyntämismahdollisuus voidaan selvittää, jolloin mitta-ajo saadaan optimoitua kunnonvalvonnan värähtelymittaukseen sopivaksi. Lisäksi mittausprosessi olisi mahdollista automatisoida.

7 Yhteenveto

Tässä Insinööriyössä tutustuttiin kunnonvalvonnan värähtelymittaukseen ja toteutettiin kunnonvalvontamittalaitteen prototyyppi. Mittalaitteeseen toteutettiin lämpö- ja värähtelymittaus, joista tässä työssä keskityttiin värähtelymittaukseen. Mittalaitetta testattiin kentällä tehtävissä mittauksissa, joissa mitattiin kuuden tarjotinvaraston tarjotinhissien värähtelyä.

Insinööriyön tavoite oli toteuttaa tilaajan vaatimusmäärittelyn täyttävä mittalaitteen prototyyppi, jolla pystyy mittaamaan värähtelyä ja lämpötilaa. Kenttätestien tavoitteena oli värähtelyn tunnuslukujen tuottaminen mittauksista.

Värähtelymittauksen ja kenttätestin suunnittelussa ja toteutuksessa käytettiin apuna alan kirjallisuutta ja standardien määrytyksiä. Mittalaitteeseen toteutettiin Raspberry Pi. Värähtelymittauksessa käytettiin ADXL345-kiihtyvyyssanturia. Laitteen koteloitiin suunniteltiin Solidworks-mallinnusohjelmistolla ja sen tulostamiseen käytettiin Anet A8 3D-tulostinta. Datan käsittelyyn käytettiin SciDAVis-, Libre Office Calc- ja Logger ohjelmistoa.

Työn tuloksena saatiin mittalaitteen prototyyppi, joka mahdollistaa uusien kunnonvalvontakeinojen kokeilemisen ja jatkokehittämisen. Tuloksena saatiin myös selvitys tarjotinhissin kunnonvalvonnan värähtelymittauksen haasteista ja kehitysmahdollisuuksista.

Tarjotinhissien värähtelymittauksen suurin ongelma oli pyörimistaajuuden määrittäminen, jonka takia mittauspisteeseen tulevia useiden eri komponenttien värähtelytaajuuksia ei saatu eroteltua toisistaan, joten taajuustason valvontakeinoja ei voitu kokeilla. Syynä tähän oli mitta-ajon luonne, jossa tasaisen pyörimisnopeuden ajovaihe ei ollut riittävän pitkä, jotta ajomoottorin todellinen pyörimisnopeus olisi tasaantunut.

Mittauksesta tuotettiin aikataason tunnuslukuja, mutta mittausta ei voi suositella tästä pisteestä suoritettavaksi. Syynä on se, että värähtelyn taajuuskomponenttien erottelun

puuttuessa värähtelytason muutosta ei pysty kohdistamaan varmuudella mihinkään komponenttiin, jolloin mittauksen hyödyllisyys täytyy kyseenalaistaa.

Jatkotoimenpiteenä satunnaisuutta pystyttäisiin vähentämään suunnittelemalla anturikiinnikkeet jokaiselle mitattavalle pyörivälle akselille, jolloin värähtely saataisiin mitattua mahdollisimman läheltä lähdeettä. Anturin kiinnityssuunnan valinnalla voidaan minimoida tarjotinhissin liikkeestä aiheutuva häiriö. Laitteen PLC voitaisiin ottaa mukaan mittausprosessiin, jolloin mitta-ajo voidaan optimoida värähtelymittaukseen sopivaksi. Muina hyötyinä prosessin voisi automatisoida käyttäen kiinteää järjestelmää, jolloin mittauksen suorittavaa resurssia ei tarvita.

Mittalaitteen jatkotoimenpiteinä ensisijaisena ovat prototyypin mekaanisen kestävyuden parantaminen vaihtamalla koteloinnin materiaalia sekä laitteen sisäisten johdotusten ja juottamisen minimointi suunnittelemalla piirilevy.

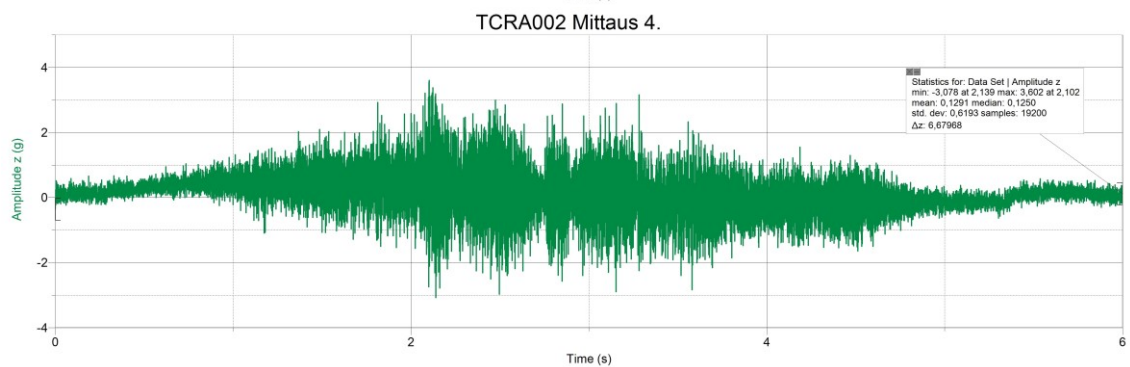
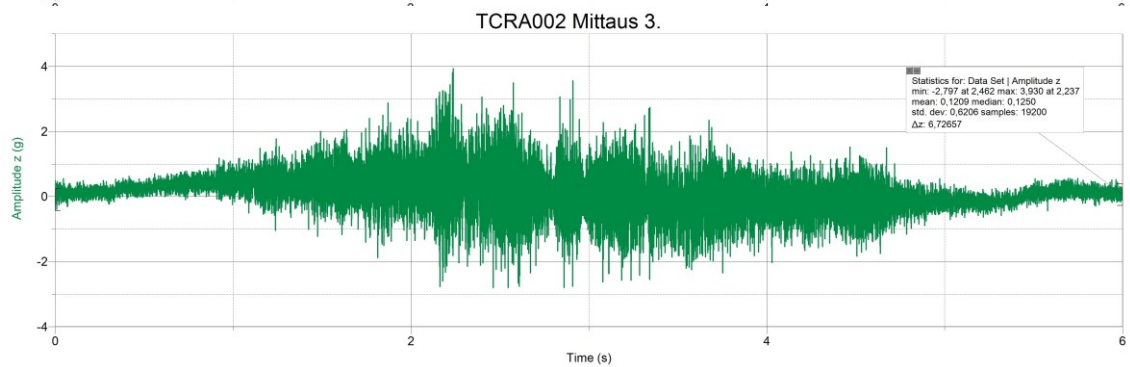
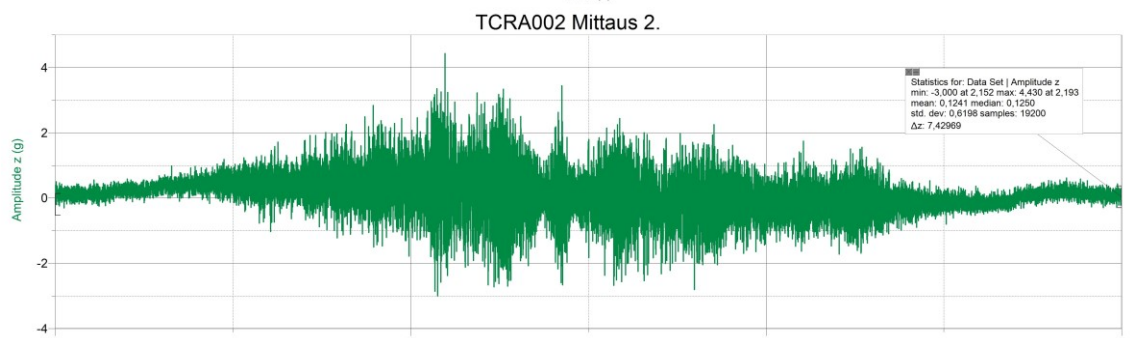
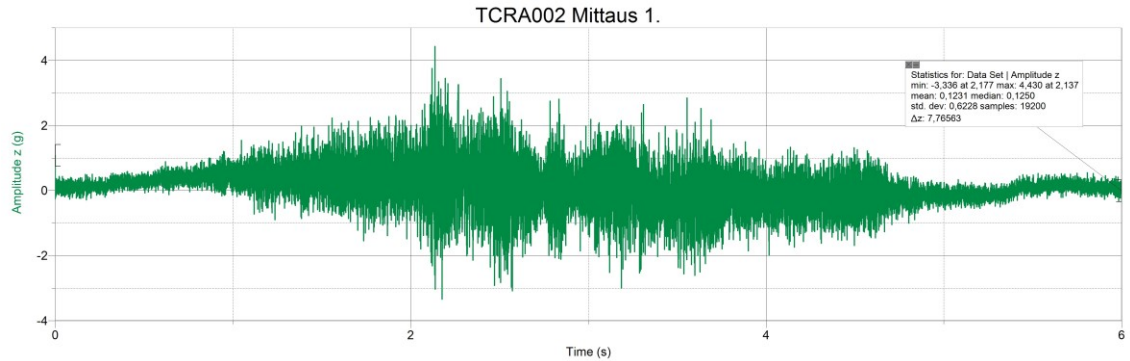
Insinööriyön aihe oli mielenkiintoinen ja melko laaja. Kehitystyön edetessä uusia näkökulmia ja toteutusvaihtoehtoja tuli usein esille, jolloin haasteena oli työn pitäminen raameissaan. Työssä opittiin, kuinka Raspberry Pillä valmistetaan usean komponentin laitekokonaisuus. Kunnonvalvonnan värähtelymittauksen toteuttamisesta opittiin, että se on monimutkainen ja kehittyvä operaatio, jossa värähtelymittauksen lisäksi tärkeää on se, että datan ja tuloksien käsittely on järjestelmällistä. Vain siten päästään lähemmäksi tilannetta, jossa koneen vikaantumista pystytään ennustamaan.

Lähteet

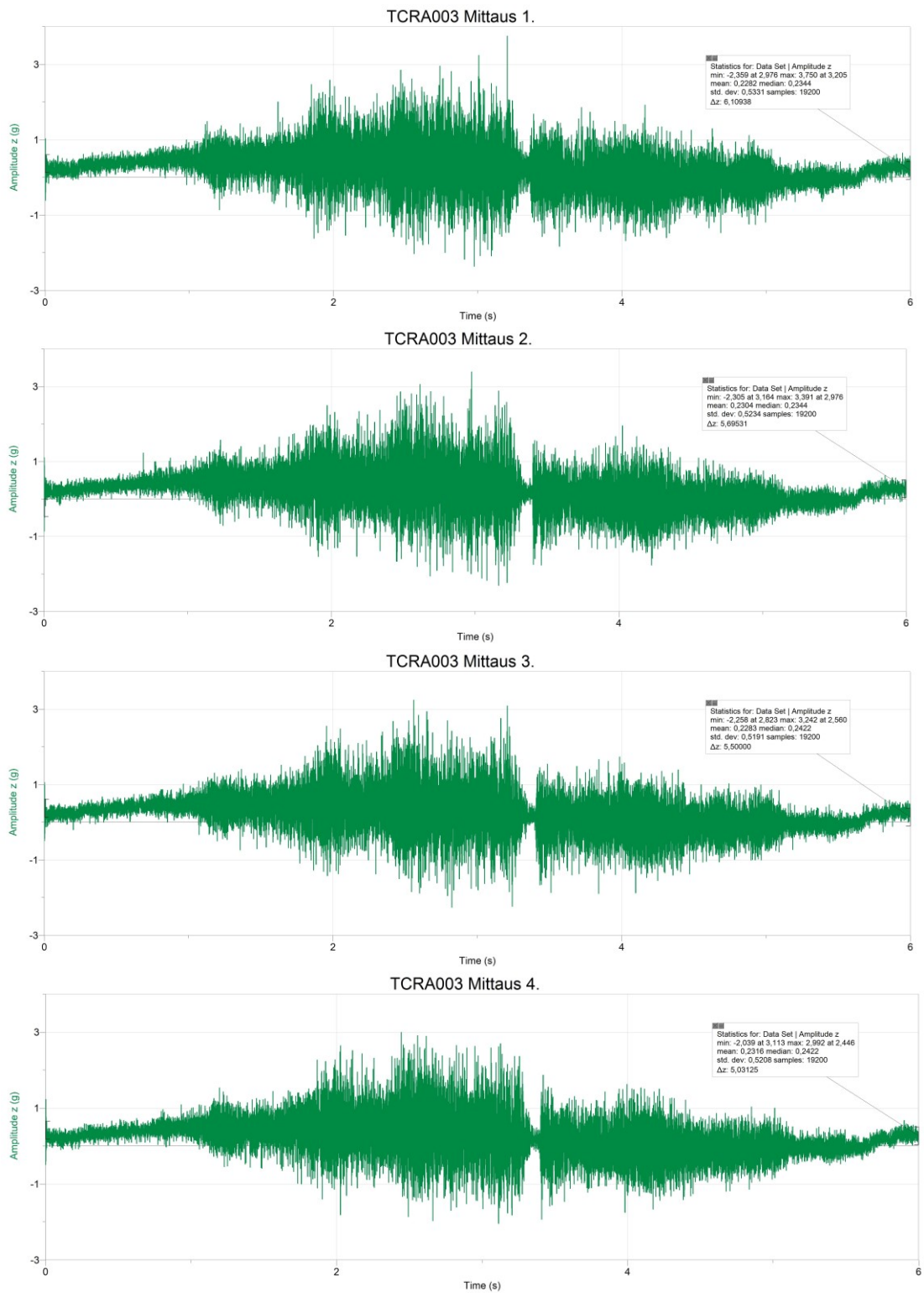
- 1 Inex lukuina. 2020. Verkkoaineisto. Inex Partners Oy. <<https://www.inex.fi/inex-yrityksena/inex-lukuina>> Luettu 15.1.2020.
- 2 Standardi PSK 7501 Prosessiteollisuuden kunnossapidon tunnusluvut. 2. painos. 32 s. 2010 Kunnossapitoyhdistys, Mikkonen, H., Miettinen, J., Leinonen, P., Jantunen, E., Kokko, V., . . . Mäkeläinen, R. 2009. Kuntoon kunnossapito: Käsikirja. Helsinki: KP-Media.
- 3 Nohynek, P. & Lumme, V. E. 2004. Kunnanvalvonnan värähtelymittaukset (2. täyd. p.). Rajamäki: KP-Media.
- 4 Standardi PSK 5701. Kunnanvalvonnan värähtelymittaus. Käsitteet ja määritelmät. Käytettävät suureet ja mittayksiköt. 8. painos. 2017.
- 5 Mikkonen H., Miettinen, J., Leinonen, P., Jantunen, E., Kokko, V., Riutta, E., Sulo, P., Komonen, K., Lumme, V., Kautto, J., Heinonen, K., Lakka, S., Mäkeläinen, R. Kunnossapitoyhdistys 2009. Kuntoon perustuva kunnossapito: käsikirja. Helsinki. KP-Media Oy.
- 6 Standardi SFS-ISO 2041:2019:en Mechanical vibration, shock and condition monitoring.
- 7 Standardi PSK 5721 Kunnanvalvonnan värähtelymittaus. Mittausasetukset. 12 s. 2013.
- 8 Standardi PSK5702 Kunnanvalvonnan värähtelymittaus. Mittauspisteen valinta ja tunnistaminen. 3. painos. 4 s. 2007.
- 9 Standardi PSK 5706 Kunnanvalvonnan värähtelymittaus. Valvontamenetelmät. 3. p. 6 s. 2002.
- 10 Standardi PSK 5703 Kunnanvalvonnan värähtelymittaus. Anturin, liittimen ja kaapelin valinta sekä asennus. 5. painos. 9 s. 2018
- 11 ABB. 2000. TTT-käsikirja.
- 12 Raspberry Pi dokumentaatio. Verkkoaineisto. Raspberry Pi Foundation. <<https://www.raspberrypi.org/documentation/>>. Luettu 15.1.2020.
- 13 Datasheet ADXL345 accelerometer. Verkkoaineisto. <<https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf>> . Luettu 15.1.2020.
- 14 Fast Fourier Transform for LibreOffice Calc v1.1. Ohjelmistolaajenne. Andrew Que. 2017. Saatavissa: <<http://fft4loc.drque.net/>>. Luettu 25.3.2020.
- 15 Interpolation for Calc. Ohjelmistolaajenne v0.4. Manuel Astros. 2018 Saatavissa: <<https://extensions.libreoffice.org/en/extensions/show/interpolation-for-calc/>>. Luettu 25.3.2020.

- 16 C2-2-303 Crane without anti-pendulum. Tilaajayrityksen sisäinen dokumentti. FoerderAnlagen Systeme GmbH. 2016.
- 17 Standardi PSK 5707 Kunnonvalvonnan värähtelymittaus. Vianmääritys. 4. p. 20 s. 2004.
- 18 WIOSS Maintenance principles. 2020. Tilaajayrityksen sisäinen dokumentti. WITRON Logistik + Informatik GmbH.
- 19 Overview of the WITRON Group. Tilaajayrityksen sisäinen dokumentti. WITRON Logistik + Informatik GmbH. 2017.
- 20 I2C-bus. What is that?. Verkkoaineisto. Telos Systementwicklung GmbH. <<http://www.i2c-bus.org/>>. Luettu 2.3.2020
- 21 Raspberry Pi SPI-bus. Verkkoaineisto. Raspberry Pi foundation. <<https://www.raspberrypi.org/documentation/hardware/raspberrypi/spi/RE-ADME.md>>. Luettu 2.3.2020
- 22 ADXL-345 I2C. Python ohjelmointikirjasto. <https://github.com/adafruit/Adafruit_CircuitPython_ADXL34x>. Luettu 15.1.2020
- 23 ADXL345-SPI. Python ohjelmointikirjasto. <<https://github.com/dzheng256/adxl345spi-py>>. Luettu 15.2.2020
- 24 ADXL345-SPI. C-Ohjelmointikirjasto. <<https://github.com/nagimov/adxl345spi>>. Luettu 15.03.2020
- 25 DS18B20 Temperature sensor tutorial. Verkkoaineisto. Scott Campbell. <<https://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/>>. Luettu 15.1.2020
- 26 Lcd_i2c. Python ohjelmointikirjasto. <https://github.com/muhammedessa/lcd_i2c.py>. Luettu 15.1.2020
- 27 Teemu Kauppi. 2020. Reactive Maintenance Shift Leader. WIOSS Witron On Site Services GmbH. Sipoo. Keskustelu 1.5.2020.

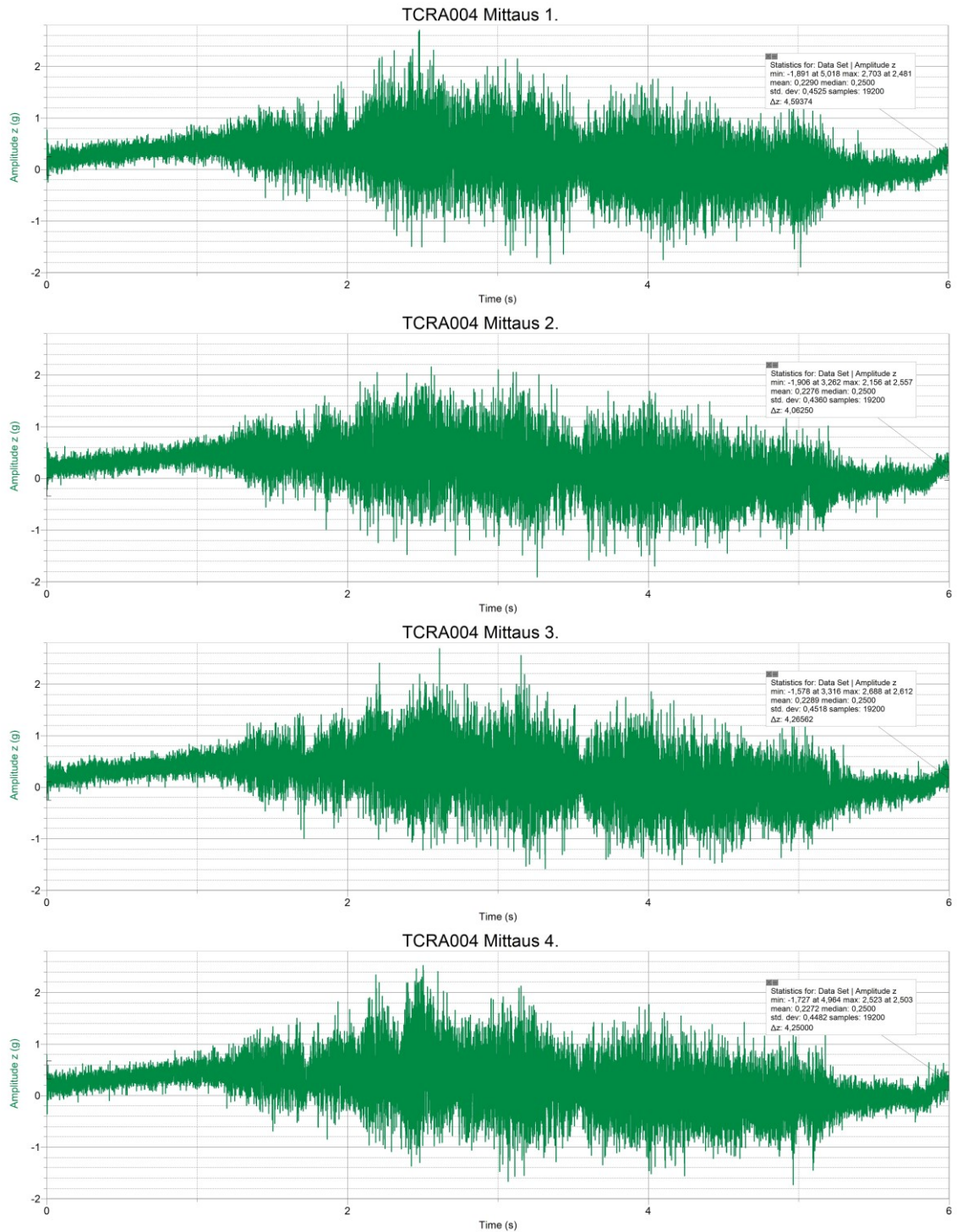
Liite 5. TCRA002 kuvaajat. $F_s = 3200$ Hz, $t = 6$ s.



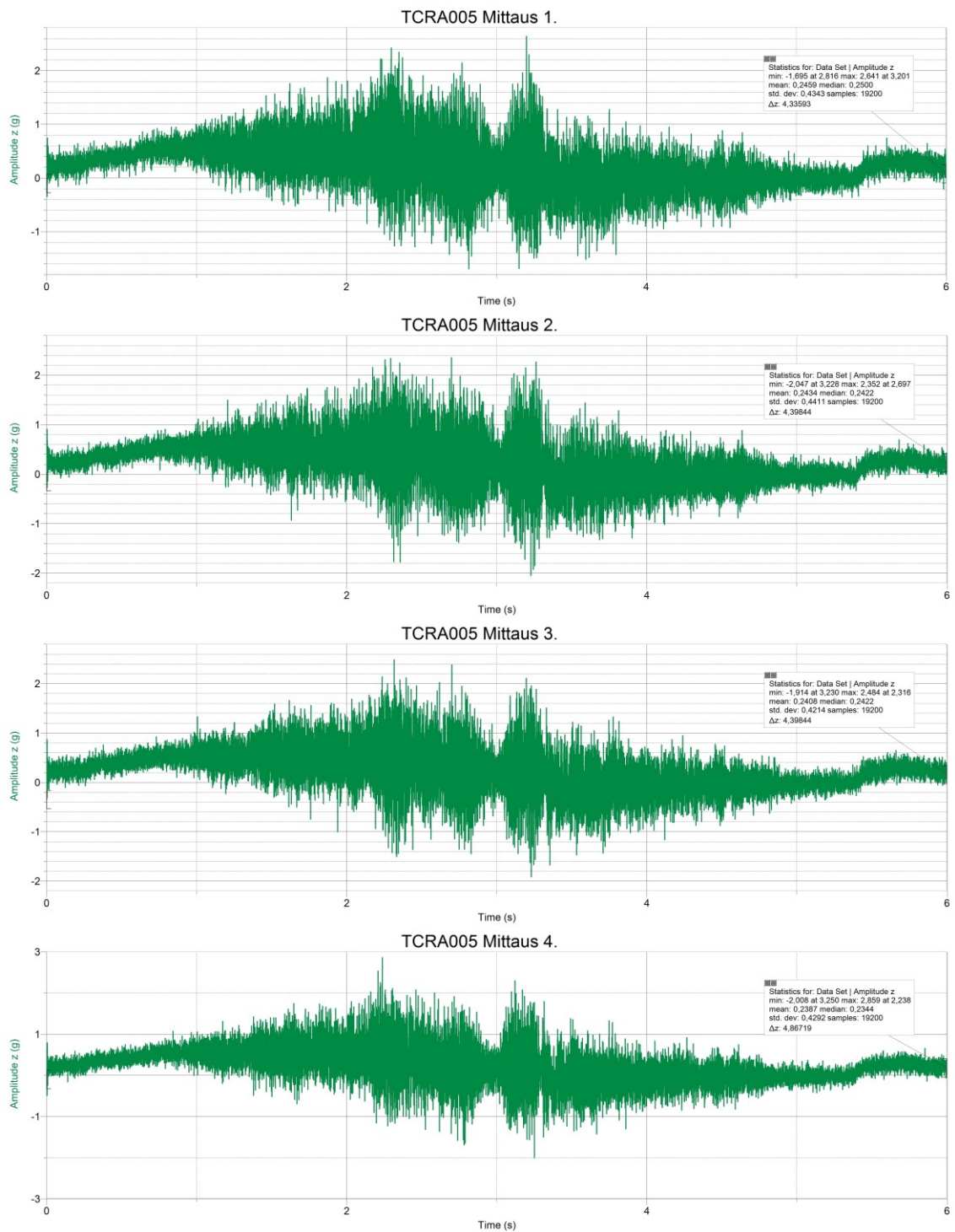
Liite 6. TCRA 003 kuvaajat. Fs = 3200 Hz, t = 6 s



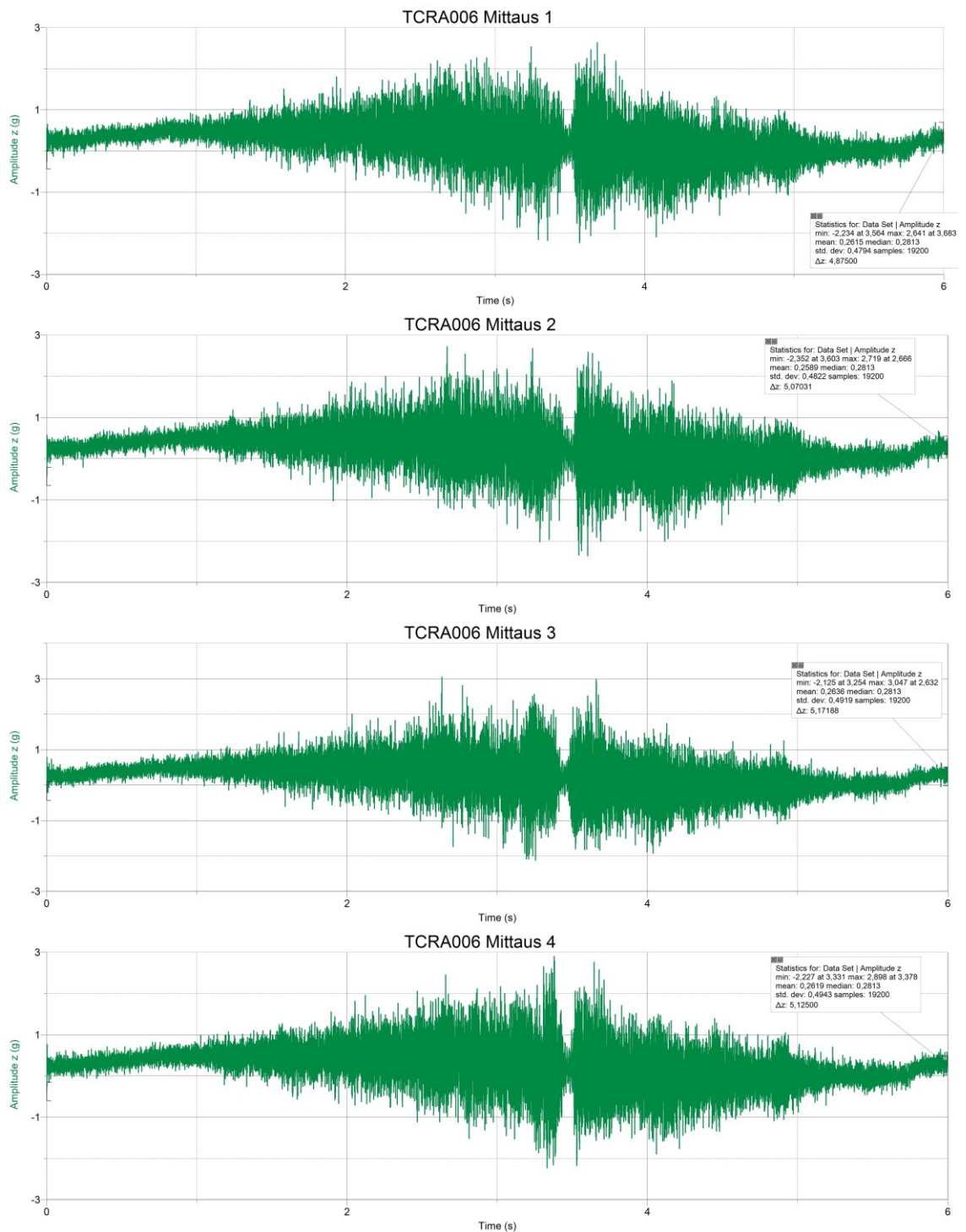
Liite 7. TCRA004 mittaukset. Fs = 3200 Hz, t = 6 s.



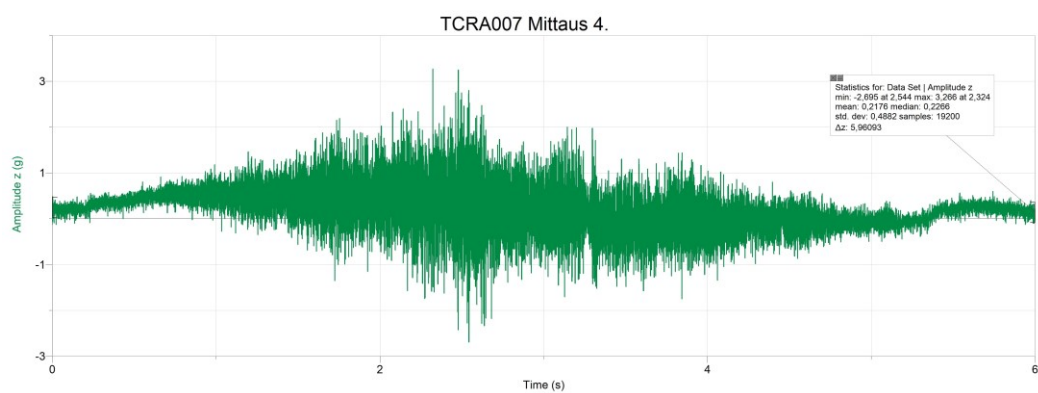
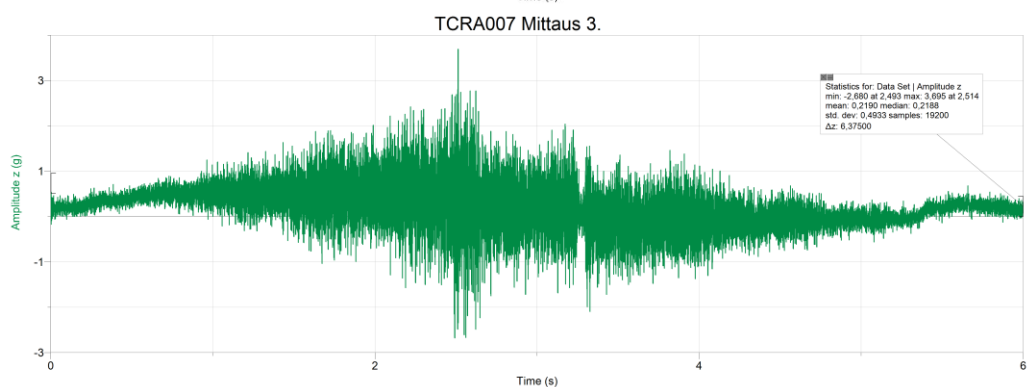
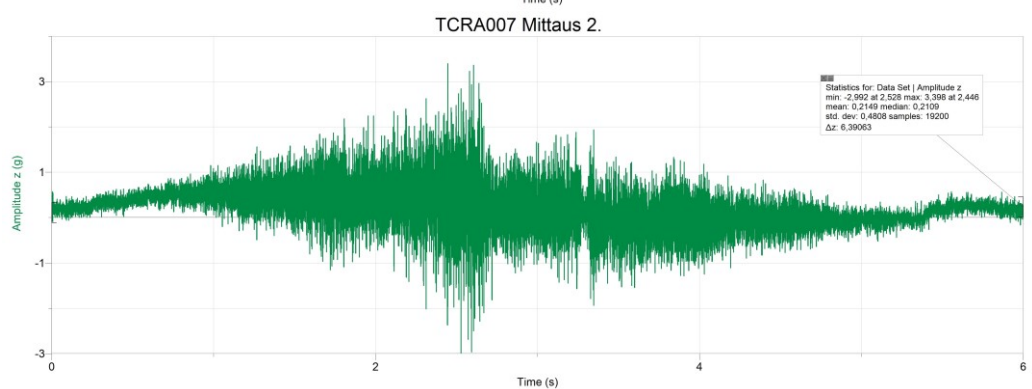
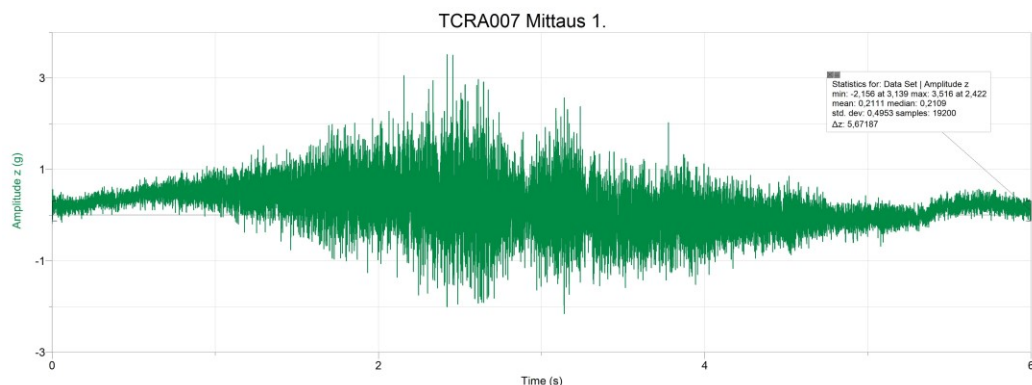
Liite 8. TCRA005 mittausket. Fs = 3200 Hz, t = 6 s.



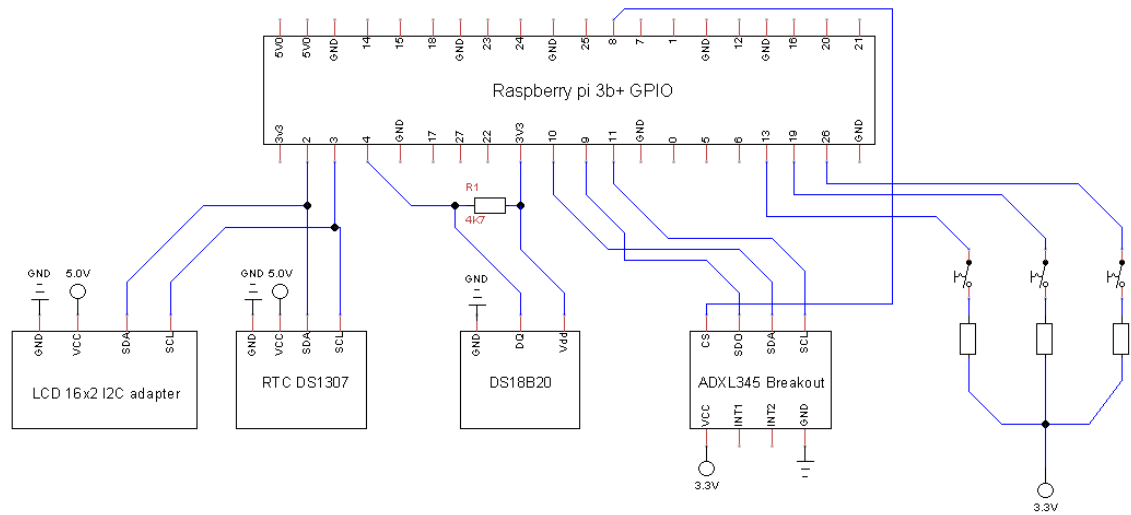
Liite 9. TCRA006 mittaukset. Fs = 3200 Hz, t = 6 s.



Liite 10. TCRA007 mittaukset. $F_s = 3200$ Hz, $t = 6$ s.



Liite 11. Laitteen piirikaavio.



Liite 12. Esimerkki integraalista Logger Prossa.

