

Timi Lahtinen

Laakerien kunnonvalvonta

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Kone- ja tuotantotekniikka

Insinöörityö

20.5.2016

Tekijä Otsikko	Timi Lahtinen Laakerien kunnonvalvonta
Sivumäärä Aika	42 sivua + 7 liitettä 20.5.2016
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Kone- ja tuotantotekniikka
Suuntautumisvaihtoehto	Koneautomaatio
Ohjaaja	Lehtori Heikki Paavilainen
<p>Tämän insinööriyön aiheena oli Beckhoff TF3600 TC3 Condition Monitorin kunnonvalvontajärjestelmä ja sen käyttö laakerien kunnonvalvonnassa.</p> <p>Tavoitteena insinööriyössä oli esittää kunnonvalvontajärjestelmän ymmärtämiseksi vaadittavat taustatiedot kunnonvalvonnasta, värähtelystä, signaalinkäsittelystä ja laakerien viikaantumisesta. Tavoitteena oli myös tutustua kunnonvalvontajärjestelmään ja rakentaa pieni mittauslaite, jolla olisi voitu valvoa laakerien kuntoa.</p> <p>Työssä perehdyttiin aiheeseen liittyvään kirjallisuuteen, standardeihin, antureihin, värähtelymittauksiin ja signaalinkäsittelyyn. Työssä luotiin esimerkkiohjelma laakerin kunnonvalvontaan huippukertoimen avulla.</p> <p>Kunnonvalvontajärjestelmään tutustuessa huomattiin sen vaativan ohjelmoijalta laajan tuntemuksen aiheesta. Ohjelman luomisen huomattiin myös oleva monimutkainen prosessi, jota ei ole helppo ymmärtää.</p>	
Avainsanat	kunnonvalvonta, laakerit, Beckhoff, värähtelymittaukset

Author(s) Title	Timi Lahtinen Condition Monitoring of Bearings
Number of Pages Date	42 pages + 7 appendices 20 May 2016
Degree	Bachelor of Engineering
Degree Programme	Mechanical Engineering
Specialisation option	Machine Automation
Instructor(s)	Heikki Paavilainen, Lecturer
<p>The objective of this Bachelor's thesis was to study Beckhoff's TF3600 TC3 Condition Monitoring toolbox and its use in the condition monitoring of bearings.</p> <p>The purpose of this thesis was to provide the essential basics for understanding the operation of the toolbox and therefore, condition monitoring, vibration, signal processing and bearing defects were described. Another purpose of the study was to familiarize with the toolbox and to build a simple measuring device that can be used in the condition monitoring of bearings.</p> <p>Relevant professional literature, standards, sensors, vibration measurements and signal processing were extensively examined. While examining the toolbox it was also noticed that wide expertise in the field is required from the initialization programmer. It was discovered that the process of creating the example program is a complex process that is not easy to comprehend.</p> <p>As a result, an example program for the condition monitoring of bearings was created using the crest factor method. Furthermore, a simple measuring device was built.</p>	
Keywords	Condition Monitoring, Bearings, Beckhoff, Vibration analysis

Sisällys

Lyhenteet

1	Johdanto	1
2	Kunnossapito ja kunnonvalvonta	2
2.1	Kunnossapito	2
2.2	Kunnonvalvonta	3
2.3	Kunnonvalvonnan mittausmenetelmät	4
3	Värähtely ja aaltoliike	6
3.1	Värähtelyn syntyminen	6
3.2	Värähtelysuureet ja niiden yhteydet	7
3.3	Värähtelyn tyypit	10
3.3.1	Harmoninen värähtely	11
3.3.2	Vaimeneva värähtely	12
3.3.3	Pakkovärähtely	13
4	Anturit	14
4.1	Anturityypit	14
4.1.1	Siirtymäanturi	15
4.1.2	Nopeusanturi	15
4.1.3	Kiihtyvyyssanturi	17
4.2	Anturin paikan valinta	19
4.3	Kiihtyvyyssanturin ominaistajuus ja kiinnitystavat	21
4.4	Kiihtyvyyssanturin mittausominaisuudet	21
5	Signaalin käsittely	22
5.1	Signaalinkäsittelyn perusteet	22
5.2	Fourier'n muunnos	23
5.2.1	FFT	23
5.2.2	Welchin menetelmä	24
5.2.3	DFT:n rajoitukset	24

6	Laakerin vikaantuminen	25
6.1	Huippukerroin	26
6.2	Huipukkuus	27
6.3	Verhokäyräanalyysi	28
6.4	Laakerien vauriotaajuudet	29
7	Beckhoff TF3600 TC3 Condition Monitoring -järjestelmä	32
7.1	Käyttötarkoitus	32
7.2	Käyttöönotto	33
7.2.1	Logiikka ja liitännät	33
7.2.2	Kunnonvalvontamenetelmän päättäminen	35
7.2.3	Ohjelman kirjoittaminen	36
7.3	Käyttö	42
8	Yhteenveto	43
	Lähteet	44

Liitteet

Liite 1. MAIN (PRG)

Liite 2. MAIN_CM (PRG)

Liite 3. FB_FunctionGenerator (FB)

Liite 4. Funktiogeneraattorin apufunktiot: am, sawtooth, sine, square ja triangular

Liite 5. FB_HRErrHist (FB)

Liite 6. FB_HRErrHist:n metodit AddError ja Reset

Liite 7. Datatyyppiyksiköt (DUTs)

Lyhenteet

DFT	<i>Discrete Fourier Transform.</i> Fourier'n muunnoksen diskreettiaikainen muunnos, jossa signaali ajatellaan jaksolliseksi ja voidaan täten esittää äärellisenä Fourier'n sarjana.
FFT	<i>Fast Fourier Transform.</i> Algoritmi joka laskee diskreetin Fourier'n muunnoksen tehokkaasti.
IEPE	<i>Integrated Electronic Piezoelectric accelerometer.</i> Pietsosähköisesti toimiva kiihtyvyyssanturi, jossa on sisäinen vahvistin.
PLC	<i>Programmable Logic Controller.</i> Tietokone jota käytetään teollisuudessa sähkömekaanisten prosessien automaatioon.
RMS	<i>Root Mean Square.</i> Aaltoliikkeen neliöllinen keskiarvo, eli sen tehollisarvo.
rpm	<i>Revolutions per Minute.</i> Luku, joka kertoo kuinka monta pyörähdystä jokin kappale tai koneen osa suorittaa minuutissa.

1 Johdanto

Beckhoff on Saksassa perustettu maailmanlaajuinen avoimien automaatiojärjestelmien toimittaja. Beckhoff Suomen pääkonttori sijaitsee Hyvinkäällä ja haarakonttorit sijaitsevat Tampereella ja Seinäjoella. Yrityksen toimittamat automaatiojärjestelmät pohjautuvat PC-pohjaiseen ohjaustekniikkaan. Tuotevalikoimasta löytyy liikkeenohjaustuotteet, kenttäväyläkomponentit ohjauspaneelit ja teollisuus-PC:t sekä automaatiosovelluksien ohjelmistot.

Beckhoffin valikoiman yhtenä uutena tuotteena on TF3600 Condition Monitorin kunnonvalvontasovellus. Sovelluksella voi suorittaa signaalinkäsittelyä, taajuusanalyysijä sekä laakereiden ja vaihteistojen kunnonvalvontaa värähtelymittauksista. Kunnonvalvonnalla saadaan tärkeää tietoa koneen kunnosta, ja syntyviin vikoihin voidaan reagoida ennen kun ne kehittyvät vaurioiksi.

Tässä insinööriyössä on tarkoitus tutustua TF3600 Condition Monitoring -ohjelmistoon ja selvittää, miten ohjelmistoa voidaan ottaa käyttöä laakerin kunnonvalvontaan. Tarkoitus on perehtyä mahdollisimman hyvin laakerien kunnonvalvonnan taustatietoihin, aloittaen kunnossapidosta ja tarkemmin kunnonvalvonnasta, edeten sitten järjestyksessä värähtelyn teoriaan, antureihin, signaalin käsittelyyn, laakerin vikaantumisen tunnistamiseen ja lopuksi itse vianmääritysohjelmistoon, kun tarvittavat taustatiedot on sisäistetty.

2 Kunnossapito ja kunnonvalvonta

2.1 Kunnossapito

Kunnossapito on kohteen toimintakyvyn ylläpitoa siten, että kohdetta voidaan käyttää tehokkaasti, turvallisesti ja luotettavasti. Kohteen kunnossapitoon sisältyvät kaikki sen elinaikaiset tekniset, hallinnolliset ja johdolliset toimenpiteet, joiden tarkoituksena on ylläpitää kohde vaaditussa toimintakunnossa koko sen elinkaaren ajan tai palauttaa kohde tilaan, jossa se pystyy suorittamaan vaaditun toiminnon. [1, s. 11–29; 2, s. 2.]

Kunnossapidon keskeisiä tavoitteita ovat tuotannon kokonaistehokkuus ja hyvä käyttövarmuus. Käyttövarmuus koostuu kohteen toimintavarmuudesta, kunnossapitovarmuudesta ja kunnossapidettävyydestä. Tärkeitä tavoitteita kunnossapidolle ovat myös turvallisuus, kustannustehokkuus ja ympäristön huomioiminen. Kunnossapidon tehtäviin kuuluvat huolto, ehkäisevä kunnossapito, korjaava kunnossapito ja parantava kunnossapito. [2, s. 4; 1, s. 41.]

Huolto

Huolto on jaksotettua toimintaa, jolla pyritään ylläpitämään kohteen käyttöominaisuuksia, estämään vikojen syntymistä ja palauttamaan heikentynyt toimintakyky. Huollon toimenpiteitä ovat mm. puhdistus, voitelu, kalibrointi ja kulumien osien vaihtaminen. [1, s. 44.]

Ehkäisevä kunnossapito

Ehkäisevään kunnossapidon toimenpiteisiin kuuluvat mm. kunnonvalvonta, tarkastaminen, testaus, käynninvalvonta, määräystenmukaisuuden varmistaminen ja aikaisempien vikaantumistietojen analysointi. Ehkäisevää kunnossapitoa voidaan suorittaa kohteen toimiessa tai huoltoseisokin aikana. Se on aikataulutettua tai jatkuvaa, tai sitä voidaan tehdä tarvittaessa. Tulosten pohjalta voidaan aikatauluttaa ja suunnitella muita kunnossapidon tehtäviä. [1, s. 44–45.]

Korjaava kunnossapito

Korjaavassa kunnossapidossa vikaantunut tai vikaantuva osa tai komponentti palautetaan käyttökuntoon. Korjaavan kunnossapidon menetelmiä ovat mm. vian tunnistaminen, vian määrittäminen, vian paikallistaminen, korjaus, toimintakunnon palauttaminen ja väliaikainen korjaus. [1, s. 43–44.]

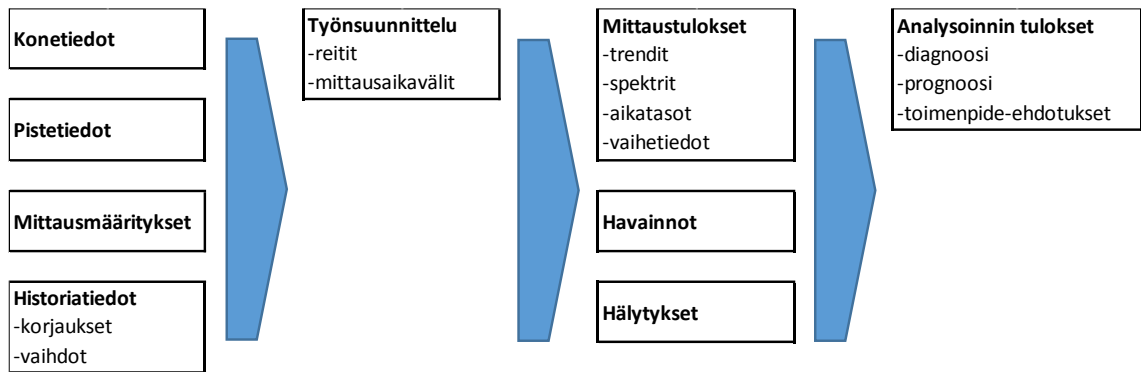
Parantava kunnossapito

Parantava kunnossapito voidaan jakaa kolmeen luokkaan. Ensimmäisessä luokassa on kohteen muuttaminen vaihtamalla osia tai komponentteja uudempiin, muuttamatta kuitenkaan olennaisesti kohteen suorituskykyä. Toisessa luokassa ovat erilaiset uudelleen suunnittelut ja korjaukset, joilla pyritään parantamaan kohteen luotettavuutta. Kolmannessa luokassa on kohteen modernisointi, jossa kohteen suorituskykyä tai lopputuotteen laatua pyritään parantamaan. Modernisaatiolla uudistetaan yleensä koko valmistusprosessi. Modernisaatio on järkevää, jos kohteella on vielä elinaikaa jäljellä, mutta sen lopputuote ei ole enää kilpailukykyistä. On edullisempaa muuttaa vanhaa tuotantolinjaa, kuin romuttaa se ja rakentaa kokonaan uusi. [1, s. 45.]

2.2 Kunnonvalvonta

Kunnonvalvontatoiminta voidaan jakaa kuvan 1 mukaisesti osatekijöihin. Mittauksen tarve ja laajuus saadaan määriteltäviä konetietojen perusteella. Pistetietoja käytetään mittauspisteen, -suunnan ja -tavan määrittämiseen. Mittausmääritykset määrittävät yksityiskohtaisesti käytettävän mittalaitteen asetukset. Työnsuunnittelussa mittauspisteet järjestetään järkeväksi reitiksi, määritetään eri mittauspisteiden mittausaikavälit ja valvotaan työtä, jotta mittaukset saadaan suoritettua oikein.

Kunnonvalvonnan mittauksista saatuja mittaustuloksia syntyy paljon. Valtaosa mittaustuloksista vaatii jatkotoimenpiteitä, jotta niitä voidaan käyttää. Mittausten ohella tärkeänä kunnonvalvonnan osana on aistinvaraisten havaintojen tekeminen. Osalle mittaustuloksista tai havainnoista on voitu antaa jokin raja-arvo, jonka ylittyminen johtaa suoraan hälytykseen, mutta useimmiten mittaustuloksia on analysoitavia, jotta ne olisivat käyttökelpoisia. Diagnoosin onnistumista varmistetaan koneiden ja tapahtuneiden vaurioiden historiatietojen kirjaamisella.



Kuva 1. Kunnonvalvontatoiminnan osatekijät [3, s. 14].

Hyvin suoritettu kunnonvalvonta lyhentää seisokkiaikaa eliminoimalla seisokin keskimääräisen odotusajan ja lyhentämällä keskimääräistä kunnossapitoaikaa. Kunnonvalvonnan ansioista seisokin odotusajan toimenpiteet voidaan suorittaa jo tuotantoaikana. Seisokin odotusajan toimenpiteitä ovat mm. vian havaitseminen, varaosien hankinta, työkalujen ja henkilökunnan varaaminen ja asiakirjojen haku. Kunnossapitoaika lyhenee, koska viat eivät pääse kehittymään vaurioiksi ja kunnossapitotyöt voidaan suunnitella paremmin vikojen ollessa tiedossa jo niiden synnyttyä. [3, s. 12–14.]

2.3 Kunnonvalvonnan mittausmenetelmät

Kunnonvalvonnan tärkeimmät mittausmenetelmät ovat värähtelymittaukset, lämpötilamittaukset, virta-analyysi epätahtimoottoreille, kulumishiukkasanalyysit ja hyötysuhdemittaukset. [3, s. 17–23.]

Värähtelymittaus

Värähtelymittaus on kunnonvalvonnassa yleisimmin käytetty menetelmä. Tämä perustuu siihen tosiasiaan, että koneet on tehty metallista. Koneet ovat täten kovia, mutta elastisesti kimmoisia rakenteita, jotka ovat lähes aina alttiita jaksollisille voimille. Oikein tulkituna, pyörivästä koneesta mitatusta värähtelystä voidaan luotettavasti määrittää eri koneosien kuntoa.

Värähtelymittausmenetelmät voidaan jakaa kahteen luokkaan. Ensimmäinen luokka kattaa koneiden yleistärinän valvontaan tai vierintälaakereiden kunnonvalvontaan käytetyt yksinkertaiset menetelmät. Ensimmäisen luokan mittalaitteita tarvitaan yleensä kaksi,

toinen mittaamaan kohteen kokonaistärinää ja toinen mittaamaan vierintälaakerien värinää. Toinen luokka pitää sisällään monimutkaisemmat menetelmät laakereiden kunnonvalvontaan tai koneiden tärinän yksityiskohtaiseen valvontaan. Toisen luokan mittalaitteita käytetään, jos valvottavassa kohteessa on voimansiirtolaitteita ja eri pyörimisnopeuksilla pyöriviä akseleita. [3, s. 18–40; 8.]

Lämpötilanmittaus

Lämpötilanmittausta kannattaa hyödyntää muita mittauksia täydentävänä menetelmänä, sillä lähes kaikkien vikojen muuttuessa vaurioiksi, vikakohtaan syntyy huomattava lämmönkehitys. Lämpötilan mittauslaitteita ovat kosketukselliset käsimitarit, infrapunamittalaitteet ja infrapunakamerat. [3, s. 20.]

Virta-analyysi epätahtimoottoreille

Virta-analyysillä epätahtimoottoreille voidaan havaita näiden moottoreiden vikoja. Mittaus suoritetaan spektrianalyysinä analyysointilaitteella, jolle saadaan mittaussignaali sähkömoottorin yhden vaiheen sähköjohdon ympärille asetettavalla virtapihdillä. Virta-analyysillä voidaan luotettavasti havaita useita sähköisiä ja mekaanisia vikoja sähkömoottorista. [3, s. 22.]

Kulumishiukkasanalyysi

Kulumishiukkasanalyysi tehdään yleensä voiteluöljynäytteestä. Voiteluöljyssä olevien kulumishiukkasten koko ja määrä lisääntyvät huomattavasti koneen kulumisen lisääntyessä. Yksittäisiä kulumishiukkasia tutkimalla saadaan tietoa kulumismekanismista ja kulumavasta komponentista. [3, s. 22–23.]

Hyötysuhdemittaukset

Hyötysuhdemittauksilla tarkoitetaan kohteen suorituskyvyn selvittämiseksi suoritettavia mittauksia. Hyötysuhdemittausten kohteina voivat olla mm. turbiinit, venttiilit, pumput, lämmönsiirtimet ja prosessikokonaisuudet. Suoritettavat mittaukset voivat olla esimerkiksi virtausmäärien, lämpötilojen ja paineiden mittauksia prosessin eri kohdissa. [3, s. 23.]

3 Värähtely ja aaltoliike

3.1 Värähtelyn syntyminen

Mekaaninen värähtely, eli värähtely, on jonkin systeemin jaksoittain toistuvaa liikettä tietyn tasapainoaseman ympärillä. Värähtelevä systeemi sisältää kyvyn varastoida potentiaalienergiaa (joussi), kyvyn varastoida kineettistä energiaa (massa, inertia) ja kyvyn hävittää energiaa vähitellen (vaimennin). Värähtelevässä systeemissä energiamuoto vaihtelee kineettisen ja potentiaalienergian välillä. Vaimennetussa systeemissä osa energiasta häviää jokaisessa värähtelyn jaksossa. Vaimennetussa systeemissä värähtelyn ylläpitoon (pakkovärähtelyyn) tarvitaan ulkoinen voima (heräte), jonka suuruus tai suunta vaihtelee. Koneenosien vioittuminen tai rakenteelliset muutokset aiheuttavat muutoksia värähtelykuvioon. [3, s. 40; 4, s. 2.1; 5, s. 66; 8.]

Värähtelyn vaikutukset prosesseihin, koneisiin ja rakenteisiin ovat pääosin haitallisia. Joissakin tiivisteissä ja kuljettimissa värähtelyä käytetään hyväksi. Värähtelyn haittavaikutuksia ovat

- jännityksien lisääntyminen
- väsymismurtumat rakenteissa
- löystyneet liitokset
- energiahäviöt
- melu
- epävarma käynti
- alentunut käyttöikä
- heikentynyt lopputuotteen laatu. [3, s. 40; 5, s. 79.]

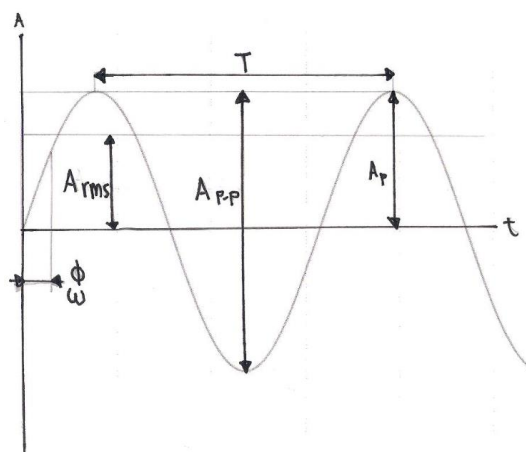
3.2 Värähtelysuureet ja niiden yhteydet

Värähtelymittauksissa ja niihin liittyvissä laskutoimituksissa käytetään taulukon 1 mukaisia suureita ja mittayksiköitä.

Taulukko 1. Värähtelymittauksissa ja laskutoimituksissa käytettyjä suureita ja mittayksiköitä [3, s. 41–52; 4, s. 2.2–2.5; 5, s. 66–79; 7, s. 1].

Suure	Merkki	Mittayksikkö	SI-mittayksikkö
Siirtymä	s	μm	m
Nopeus	v	mm/s	m/s
Kiihtyvyys	a	m/s^2 ; $g \approx 9.81\text{m/s}^2$	m/s^2
Taajuus	f	1/s	Hz
Pyörimisnopeus	n	1/min(rpm), 1/s	1/s
Kulmataajuus	ω	rad/s	rad/s
Vaihekulma	ϕ	°	rad
Jaksonaika	T	ms	s
Huippuarvo	A_p	μm	m
Huipusta-huippuun -arvo	A_{p-p}	μm	m
Tehollisarvo	A_{rms}	μm	m
Jousivakio	k	N/m	N/m
Vaimennuskerroin	c		
Aika	t	ms	s

Kuvassa 2 selvennetään taulukon 1 niitä suureita, jotka liittyvät suoraan aaltoliikkeeseen.



Kuva 2. Värähtelyyn liittyvät peruskäsitteet [3, s. 51–52; 7, s. 1].

Massa

Massa (m) on jäykkä kappale. Massa hitausvoima F_I lasketaan seuraavasti

$$F_I = ma, \quad (1)$$

jossa a on kiihtyvyys. [3, s. 41; 4, s. 2.2; 6, s. 91.]

Siirtymä

Siirtymä (s) on kappaleen poikkeama tasapainoasemasta ajan funktiona. Siirtymän suuruus on pienimmillään $-A_P$ ja suurimmillaan A_P . Siirtymä voidaan laskea seuraavasti, jos hetkellä $t = 0$ on $s = 0$ ja $v = +A_P\omega$

$$s = A_P * (\sin 2\pi f t), \quad (2)$$

suluissa oleva lauseketta ei tarvitse ottaa mukaan laskuihin, mikäli värähtelyn taajuus tunnetaan, sillä se kuvaa hetkellisiä arvoja. [3, s. 41–52; 5, s. 68–69; 6, s. 95.]

Nopeus

Nopeus (v) on siirtymän muutosnopeus ajan funktiona, eli siirtymän aikaderivaatta. Nopeus voidaan laskea seuraavasti, jos hetkellä $t = 0$ on $s = 0$ ja $v = +A\omega$

$$v = A_P * 2\pi f * (\cos 2\pi f t), \quad (3)$$

suluissa olevaa lauseketta ei tässäkään laskutoimituksessa tarvitse ottaa mukaan jos värähtelyn taajuus tunnetaan, syy on sama kuin siirtymää laskettaessa. [3, s. 41–52; 5, s. 68–69; 6, s. 95.]

Kiihtyvyys

Kiihtyvyys (a) on nopeuden muutosnopeus ajan funktiona. Kiihtyvyys on siirtymän toinen aikaderivaatta. Kiihtyvyys voidaan laskea seuraavasti, hetkellä $t = 0$ on $s = 0$ ja $v = +A\omega$

$$a = 4\pi^2 * f^2 * s. \quad [3, \text{s. } 41\text{--}52; 5, \text{s. } 68\text{--}69; 6, \text{s. } 95.] \quad (4)$$

Pyörimisnopeus

Pyörimisnopeus kertoo, kuinka monta kierrosta kappale pyörii tietyssä ajassa. Pyörimisnopeus ilmoitetaan yleensä kierrosta sekunnissa tai kierrosta minuutissa (rpm). [5, s. 66–79.]

Jaksonaika ja taajuus

Jaksonaika (T) ja taajuus (f) ovat suoraan yhteydessä toisiinsa. Jaksonaika kertoo, kuinka pitkä aika kuluu värähtelyn yhteen jaksoon. Taajuus kertoo, kuinka monta värähdystä tapahtuu sekunnissa. Jaksonaikaa voidaan mitata, mutta se voidaan myös laskea seuraavasti

$$T = 2\pi * \sqrt{\frac{m}{k}} = \frac{1}{f}, \quad (5)$$

ja taajuus

$$f = \frac{1}{2\pi} \sqrt{\frac{k}{m}} = \frac{1}{T}. \quad [5, \text{s. } 67; 6, \text{s. } 95.] \quad (6)$$

Vaihekulma ja kulmataajuus

Vaihekulma ja kulmataajuus kertovat siniaallon syntymistä kuvaavan suuntavektorin suunnasta (vaihekulma φ) ja suunnan muutosnopeudesta (kulmataajuus ω). Vaihekulman ja kulmataajuuden laskukaavat ovat seuraavanlaiset

$$\varphi = 2\pi * f * t, \quad (7)$$

$$\omega = \sqrt{\frac{k}{m}} = 2\pi * f. \text{ [6, s. 95.]}$$

Huippuarvo, huipusta huippuun -arvo ja tehollisarvo

Huippuarvo, huipusta huippuun -arvo ja tehollisarvo ovat suureita, joilla voidaan kuvata mittaussignaalin suuruutta. Huippuarvo kertoo mittausajan korkeimman hetkellisen amplitudin suuruuden. Huipusta huippuun -arvo ilmaisee mittausajan suurimman ja pienimmän amplitudiarvon erotuksen. Tehollisarvo (RMS) kuvaa hyvin värähtelyn keskimääräisen energiasisällön. Tehollisarvo ei kuitenkaan muutu merkittävästi hetkellisistä korkeista arvoista, joten huippuarvon mittausmenetelmällä havaitaan ensin esimerkiksi laakeriviat tai rattaiden vauriot. Tehollisarvon matemaattinen laskukaava on

$$A_{rms} = \sqrt{\frac{1}{T} \int_0^T s^2 dt}, \quad (8)$$

Sinimuotisen värähtelyn tapauksessa pätevät seuraavat yhteydet

$$A_{p-p} = 2 * A_p, \quad (9)$$

$$A_{rms} = 0,7 * A_p. \text{ [3, s. 51–52.] } \quad (10)$$

3.3 Värähtelyn tyypit

Värähtelevällä systeemillä on todellisuudessa vähintään kuusi vapausastetta. Tulevissa kappaleissa käsitellään yhden vapausasteen värähtelijöitä. Värähtelijä voi liikkua kolmeen keskenään kohtisuoran akselin suuntaisesti tai kiertyä kunkin akselin ympäri. Kuva 3 selventää värähtelijän vapausasteita. [3, s. 40–41; 4, s. 2.3–2.30.]

3.3.1 Harmoninen värähtely

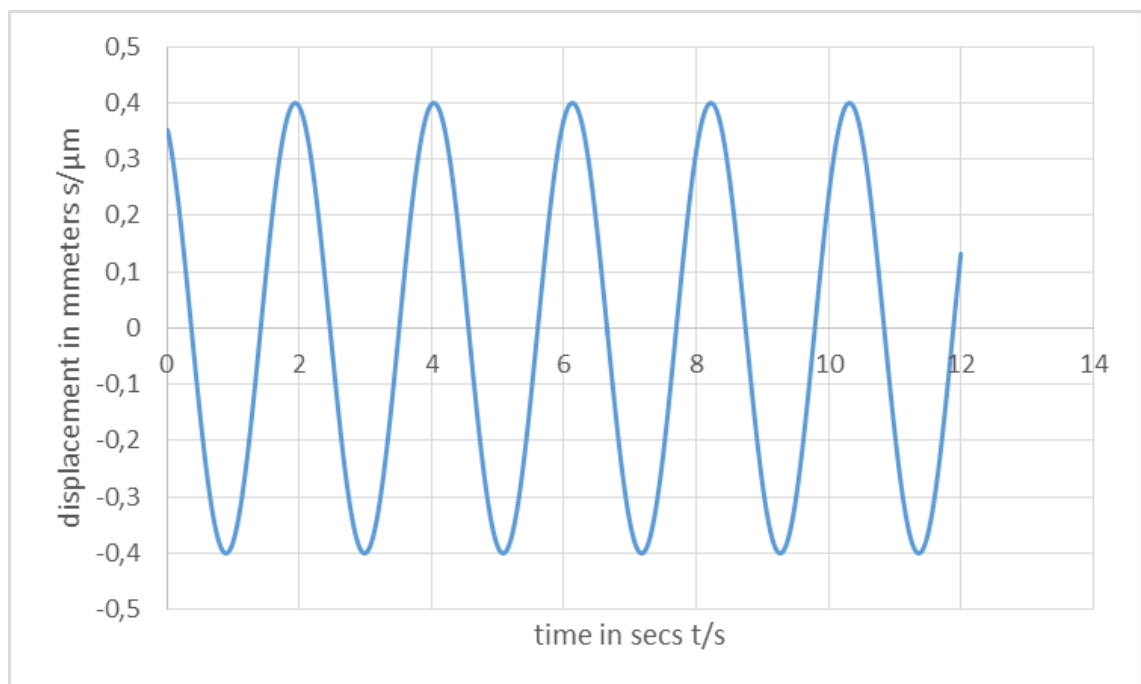
Värähtelijä, johon ei vaikuta ulkoista herätettä eikä vaimennusta, värähtelee harmonisesti tasapainoasemansa suhteen kuvan 3 mukaisesti. Harmonista värähtelyä voidaan myös kutsua vapaaksi värähtelyksi. Harmonisen värähtelijän liikeyhtälö voidaan kirjoittaa muotoon

$$F_I + F_S = 0 \quad (11)$$

$$ma + ks = 0. \quad (12)$$

Harmoninen värähtelijä värähtelee ominaistajuuudellaan (ω_n)

$$\omega_n = \sqrt{\frac{k}{m}}. \quad [3, \text{ s. } 41\text{--}42; 4, \text{ s. } 2.2\text{--}2.3; 5, \text{ s. } 66\text{--}69.] \quad (13)$$



Kuva 3. Harmoninen värähtely [3, s. 43].

3.3.2 Vaimeneva värähtely

Värähtely on vaimenevaa, kun värähtelevään kappaleeseen vaikuttaa vaimennus, mutta ei herätettä. Vaimenevan värähtelyn liikeyhtälö on

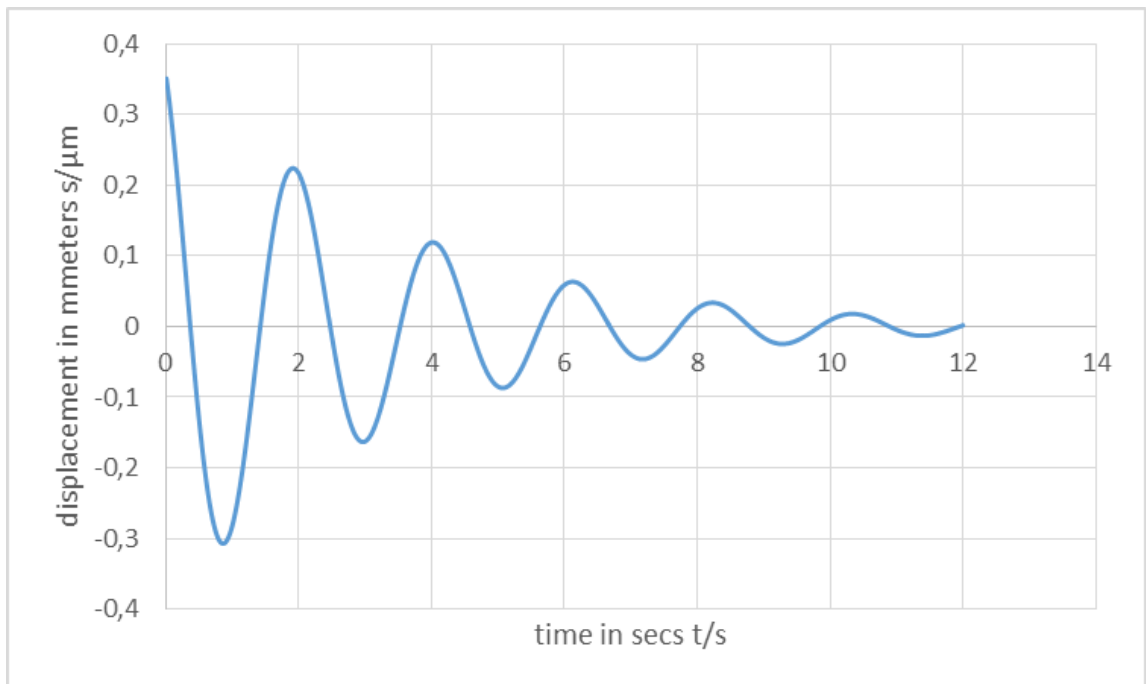
$$F_l + F_D + F_S = 0 \quad (14)$$

$$ma + cv + ks = 0. \quad (15)$$

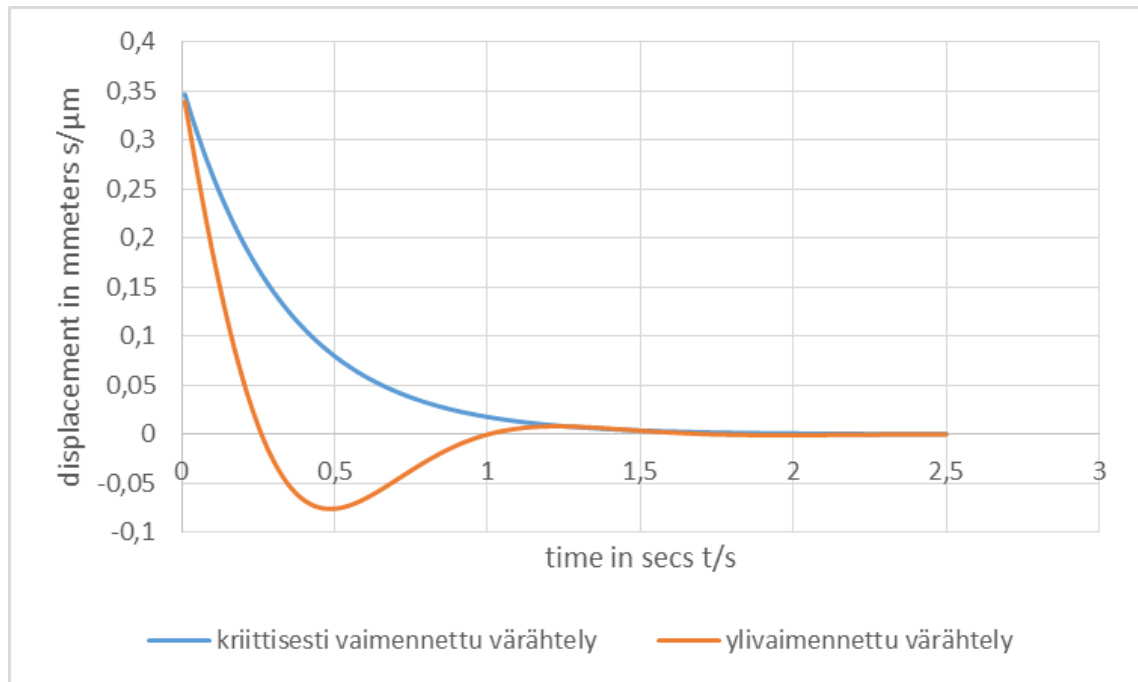
Vaimenevan värähtelyn todellinen amplitudi pienenee ajan mukaan. Vaimennustekijä (ζ) lasketaan kaavasta

$$\zeta = \frac{c}{2\sqrt{k*m}}, \quad (16)$$

ja sen arvo määrittää vaimennetun värähtelyn luonteen. Jos $\zeta < 1$, kyseessä on **alivaimennettu värähtely** (kuva 4), jos $\zeta = 1$ kyseessä on **kriittinen vaimennus** (kuva 5) ja jos $\zeta > 1$, kyseessä on **ylivaimennettu värähtely** (kuva 5).



Kuva 4. Alivaimennettu värähtely [3, s. 43].



Kuva 5. Kriittisesti vaimennettu ja ylivaimennettu värähtely [3, s. 43; 5, s. 76].

Vaimennetun värähtelyliikkeen **ominaistaajuus** (ω_d) on pienempi kuin vastaavan **vaimentamattoman** värähtelyn **ominaistaajuus** (ω_n) ja se lasketaan kaavasta

$$\omega_d = \omega_n \sqrt{1 - \zeta^2}. \quad (17)$$

Vaimennustekijä on kuitenkin yleensä niin pieni, että se ei vaikuta merkittävästi värähtelyn ominaistaajuuteen. [3, s. 40–43; 4, s. 2.5–2.7; 5, s. 75–76.]

3.3.3 Pakkovärähtely

Värähtelyä, jonka energiaa ylläpidetään ulkoisen tekijän aiheuttaman voiman (herätteen, F) avulla, kutsutaan pakkovärähtelyksi. Heräte on harmoninen voima, joka muuttaa suuruuttaan tai suuntaansa jaksollisesti. Pakkovärähtelijän liikeyhtälöksi saadaan

$$F_I + F_D + F_S = F \quad (18)$$

$$ma + cv + ks = F \sin(\omega_F * t) \quad (19)$$

ω_F on herätteen kulmataajuus.

Pienellä vaimennuskertoimella värähtelyn amplitudi saavuttaa maksimiarvonsa herätteen kulmataajuuden ollessa lähes yhtä suuri kuin värähtelyjen ominaistaajuus. Amplitudin arvo voi nousta hyvinkin suureksi. Kyseistä tilannetta, jossa värähtelijän siirtymä (amplitudi) saadaan maksimiinsa, kutsutaan resonanssiksi. Resonanssitaajuuden laskukaava on

$$f_{res} = \frac{1}{2\pi} \sqrt{\omega_n^2 - \frac{c^2}{2m^2}}. \quad [3, \text{ s. } 44\text{--}45; 4, \text{ s. } 2.7\text{--}2.19; 5, \text{ s. } 77\text{--}78.] \quad (20)$$

4 Anturit

4.1 Anturityypit

Anturi on laite, joka muuntaa värähtelyliikkeen optiseksi, mekaaniseksi tai yleisimmin sähköiseksi signaaliksi. Kohteen värähtelyä voidaan mitata kolmella eri anturityypillä, joista jokainen mittaa eri värähtelyn suuretta. Anturityypit ovat siirtymäanturi, nopeusanturi ja kiihtyvyydentanturi. Ne eroavat rakenteeltaan ja toimintaperiaatteeltaan toisistaan. Antureilla mitattuja suureita voidaan derivoida ja integroida, jotta suureet saa muutettua toisiksi suureiksi.

$$v = \dot{s}, \text{ nopeus on siirtymän ensimmäinen aikaderivaatta,} \quad (21)$$

$$a = \ddot{s}, \text{ kiihtyvyys on siirtymän toinen aikaderivaatta,} \quad (22)$$

$$s = \int v \, dt, \text{ siirtymä on nopeuden integraali,} \quad (23)$$

$$v = \int a \, dt, \text{ nopeus on kiihtyvyyden integraali.} \quad (24)$$

[3, s. 40–45; 4, s. 10.1; 5, s. 68–69.]

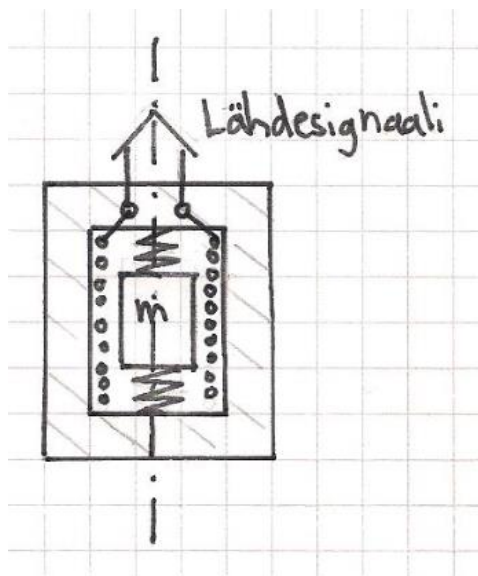
4.1.1 Siirtymäanturi

Siirtymäanturi mittaa nimensä mukaisesti kohteen sijaintia suhteessa vertailupisteeseen. Sillä voidaan havaita asemaa, akselin ratakäyriä, pinnan muodonmuutoksia ja värähtelyä. Tavallisesti siirtymäantureita käytetään liukulaakeroitujen koneiden kunnonvalvonnassa.

Pyörrevirta-anturi on tyypillinen siirtymäanturi. Sillä mitataan läheltä mittauspistettä. Anturin päässä oleva kela muodostaa magneettikentän. Magneettikenttä indusoi pyörrevirtoja kohteen ferromagneettiseen pintaan. Muodostuneet pyörrevirrat aiheuttavat muutoksen anturin päässä olevan kelan jännitteeseen. Kohteen liikuessa, anturin ja kohteen välimatka muuttuu ja aiheuttaa muutoksia pyörrevirtoihin. Pyörrevirtojen muuttuminen vaikuttaa edelleen kelan jännitteeseen, Anturista saadaan ulos jännite, joka kertoo anturin ja mittauspisteen välisen etäisyyden. [3, s. 49.]

4.1.2 Nopeusanturi

Nopeusanturin sisällä on käämi, jonka sisällä on jousien varaan kiinnitetty magneettinen massa (m), kuten kuvassa 6. Kohteeseen kiinnitetty anturi värähtelee kohteen mukana. Anturin sisällä oleva massa seuraa muun anturin liikettä viiveellä. Magneettisen massan värähdellessä käämiin suhteen, indusoi käämiin jännite, joka voidaan johtaa mittaussignaaliiksi (lähdesignaali).



Kuva 6. Nopeusanturin rakenne [3, s. 47].

Parhaimmillaan nopeusanturin mittausalue on 5 Hz–2000 Hz. Alataajuus määräytyy anturin sisäisen resonanssin mukaan. Resonanssi voi olla 3 Hz–12 Hz riippuen magneetin massasta ja jousen jäykkyydestä. Mittausalueen alataajuuden tulee olla muutamia hertsejä resonanssin yläpuolella.

Nopeusanturin hyviä puolia on sen helppokäyttöisyys, herkkyys ja signaali/häiriösuhteen erinomaisuus. Nopeusanturin ei tarvitse ulkoista virtalähdettä. Nopeusanturi mittaa värinänopeutta, joka on kunnonvalvonnassa yleisimmin käytetty mittaussuure.

Nopeusanturin huonoja puolia on sen häiriöherkkyys lämmönvaihtelulle ja sähkömagneettisille kentille. Nopeusanturin ovat usein rakenteeltaan isokokoisia ja raskaita, rakenne on myös herkkä vaurioitumaan. Mittaustaajuus ja käyttölämpötila nopeusantureilla ovat myös rajoitettuja. Anturin asento ja suuntaisuus voivat vaikuttaa mittaustulokseen. Usein anturille ilmoitetaan, mitataan pysty- vai vaakasuorassa.

Nopeusmittauksessa uutena menetelmänä on lasermittaus, joka ei tarvitse kosketusta kohteeseen. Kyseisellä menetelmällä pystytään mittaamaan 0 Hz–20 kHz taajuuskaisalta, joka on erittäin laaja. Menetelmän ongelmana on lasersäteen takaisinheijastuminen mitattavasta kohteesta. Ratkaisuja takaisinheijastumiseen on esimerkiksi heijastavan teipin kiinnittäminen kohteen pintaan, tai mittauslaitteen sijoittamisen huomattavan lähelle kohdetta. [3, s. 47–48.]

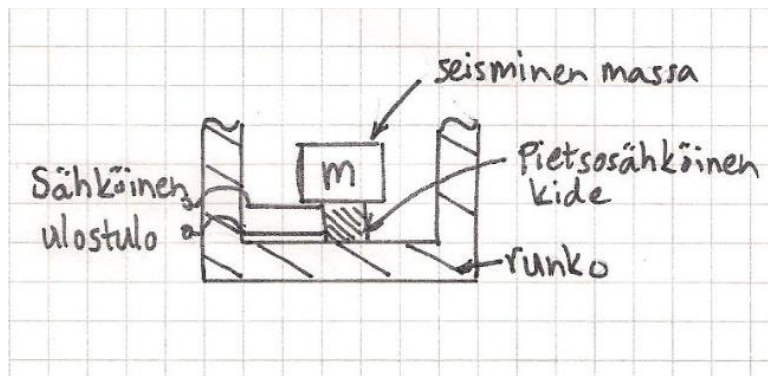
4.1.3 Kiihtyvyyssanturi

Kiihtyvyyssanturin ulostulosignaali (virta) on suoraan verrannollinen syötettyyn kiihtyvyyteen (värinään/tärinään). Yksinkertaisimmillaan kiihtyvyyssanturin rakenne on kuvan 7 mukainen. Pietsosähköisen kiteen päälle on asennettu seisminen massa. Anturin altistuessa kiihtyvyydelle, seisminen elementti puristaa pietsosähköistä kidettä (hitausvoimien vaikutuksesta), jolloin kiteeseen syntyy varaus. Näin yksinkertainen anturi ilmaisee kiihtyvyyden vain yhteen suuntaan (+ merkkinen varaus).

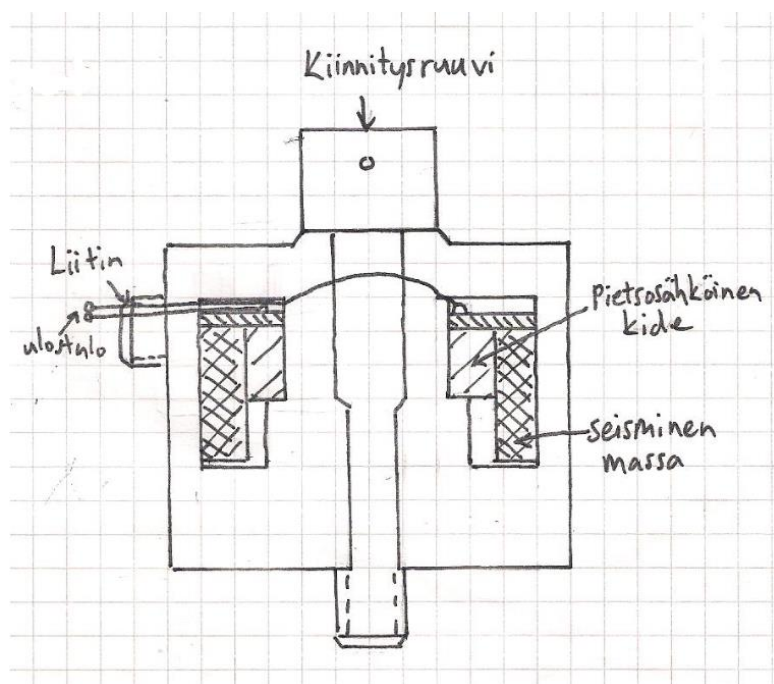
Todellisuudessa kiihtyvyyssanturi, jossa pietsosähköinen kide on kiinnitetty seismiseen massaan, on parempi (kuva 8). Tämän tyyppisessä anturissa seismisen elementin liikuminen voi joko puristaa tai venyttää pietsosähköistä kidettä, jolloin syntyvän varauksen etumerkki on kiihtyvyyden suunnan mukainen (+, -). Värähtelyssä pietsosähköiseen kiteeseen syntynyt varaus muuttaa jatkuvasti suuruuttaan ja suuntaansa.

Kiihtyvyyssanturin hyviä puolia on rakenteen keveys, pienikokoisuus, edullisuus ja taajuusalueen laajuus. Kiihtyvyyssanturin mittausalue (taajuusalue) on 2 Hz–14 kHz, mutta anturin ominaisuudet ja kiinnitystapa kohteeseen vaikuttavat huomattavasti niin ylätaajuuksiin kuin alataajuuksiinkin.

Huonoja puolia kiihtyvyyssanturilla ovat rakenteen herkkyys, matalahko signaali/häiriösuhde, tuloksen alttius sähköstaattisille häiriöille ja signaalien matalien taajuuksien häviäminen korkeataajuisen osan alle. Kiihtyvyyssanturilla on myös 1 s–600 s asettumisaika, joka johtuu kahdesta häiriöstä. Yksi häiriöistä syntyy siirrettävään anturiin asennusvaiheessa, kun anturiin kohdistuva isku lamauttaa pietsosähköisen kiteen. Toinen häiriö syntyy, kun anturiin kytketään jännite. Toisen häiriön kesto riippuu anturin vahvistimen rakenteesta ja mittausalueen alataajuudesta, mitä matalampi, sitä pidempään toipuminen kestää.



Kuva 7. Yksinkertainen kiihtyvyyssanturi [4, s. 10.16].



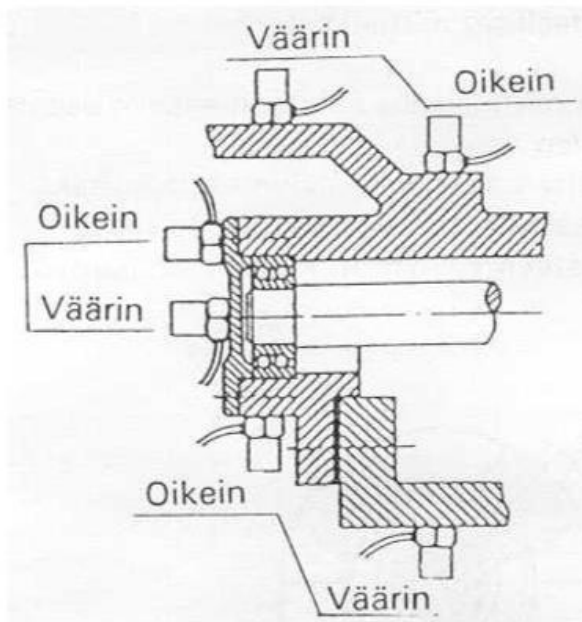
Kuva 8. Kiihtyvyyssanturi [3, s. 46; 5. s. 10.19].

Kiihtyvyyssanturi voidaan kiinnittää kohteen mittauspisteeseen ruuvilla, magneetilla, mehiläisvahalla tai käsin painamalla. Suurin mittausalue saadaan ruuvikiinnityksellä, joka voi sopivalla anturilla ylittää jopa 20 kHz. Magneetilla mittausalue riippuu magneetin voimakkuudesta ja kiinnityspinnan tasaisuudesta, mittausalue on 800 Hz–4 kHz. Käsin painamalla mittausalueen ylärajataajuudeksi saadaan vain 600 Hz. Kiihtyvyyssanturin alataajuus määrittyy anturin suunnittelusta. Taajuusvasteen lineaarinen alue päättyy yleensä 0,1 Hz–3,0 Hz välillä. [3, s. 46–56; 4. s. 10.1–10.38.]

4.2 Anturin paikan valinta

Anturin paikan valintaa käsitellään yleisemmin standardissa PSK 5702 [9] ja tarkemmin vierintälaakerien osalta standardissa ISO 15242 [13; 14; 15; 16]. Mittauspisteet valitaan siten, että anturi ja värähtelyn lähde ovat mahdollisimman lähellä toisiaan. Korkeataajuisen värähtelyn menettäessä energiaa kohdatessaan rajapinnan (liitokset). Mittauspiste tulee siis valita niin, että anturin ja värähtelylähteen välillä on mahdollisimman vähän rajapintoja. Pyörivän massan värähtely siirtyy laakerin välityksellä runkoon. Mittauspisteet on siis hyvä valita laakerointikohdista.

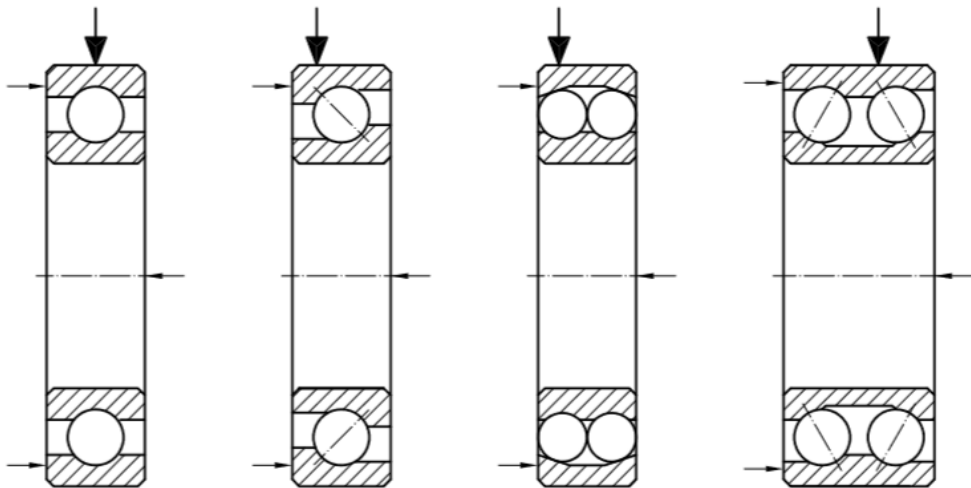
Värähtelymittaus suoritetaan yleensä säteissuuntaisesti, mutta tarvittaessa akselin suuntaisesti. Kuva 9 selventää anturin paikan valinnan ehtoja.



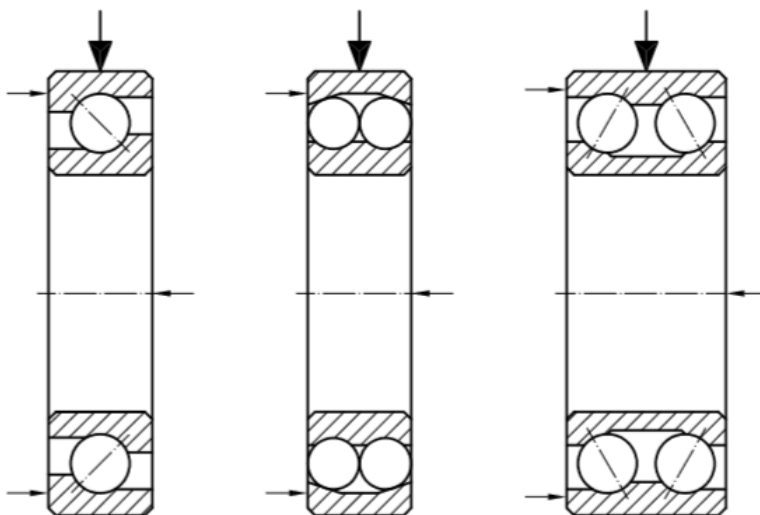
Kuva 9. Anturin paikan valinta [3, s. 53; 9, s. 2].

Kohteesta on valittava riittävän monta mittauspistettä niin, ettei mikään vika eri koeneelimissä jää havaitsematta. Mittauspisteiden minimimäärä on yksi laakeria kohden. Käytännössä anturia ei aina pysty asentamaan kaikkien laakerien kohdalle ahtauden tai työturvallisuuden vuoksi. Mittauspiste täytyy valita siten, että se päästään myös asentamaan tai vaihtamaan turvallisesti. Anturin kaapelointi ja suojaus täytyy pystyä suorittamaan luotettavasti. Anturin sijainti ei saa vaikeuttaa käyttö- ja huoltotoimenpiteitä. [3, s. 53; 9, s. 2.]

Vierintälaakerien värähtelymittauksissa anturin sijainti laakerin suhteen on määritelty standardissa ISO 15242-2 kuvan 10 mukaisesti. Anturi tulee sijoittaa uloimman renkaan ulkopintaan, uloimman uran ja kuulun kontaktipinnan keskikohtalle säteissuuntaisesti. Laakerin valmistaja toimittaa tarkemmat tiedot. Vaihtoehtoisesti anturi voidaan asentaa ulomman renkaan keskelle, paitsi syväuraisten kuulalaakereiden tapauksessa (kuva 11). Anturin ja laakerin säteissuunnan aksiaalinen poikkeama on 5° joka suuntaan. [14, s. 6–7].



Kuva 10. Anturin sijainti säteissuuntaisessa mittauksessa [14, s. 6].



Kuva 11. Anturin vaihtoehtoinen sijainti [14, s. 6].

4.3 Kiihtyvyyssanturin ominaistaajuus ja kiinnitystavat

Kiinnitystapa vaikuttaa kiihtyvyyssanturin rakenteelliseen ominaistaajuuteen. Irrallisen kiihtyvyyssanturin ominaistaajuus on tyypillisesti 5 kHz–30 kHz. Kiihtyvyyssanturin kiinnitystavat ja niiden mahdollistamat taajuusvasteen ylärajat sulkeissa:

- vaarnaruuvikiinnitys (~40 kHz)
- metyylysiyaniakrylaattiliima (~30 kHz)
- ohut kaksipuolinen teippi (~25 kHz)
- pehmeä liima (~10 kHz)
- magneettikiinnitys (~7 kHz)
- kiinnitysnippa (pikaliitin) (~4 kHz)
- koetinpuikko (ei kiinnitys) (~1,9 kHz)
- paksu kaksipuolinen teippi (~1,8 kHz). [3, s. 57–60.]

4.4 Kiihtyvyyssanturin mittausominaisuudet

Anturin kykyä erottaa samanaikaisesti voimakkaita ja heikkoja signaaleja kuvaa signaali/kohinasuhde. Tyypillinen signaali/kohinasuhde kiihtyvyyssanturille on 60 dB, joka tarkoittaa, että 1000 mittausyksikön suuruisesta tärinästä voidaan erottaa yli yhden mittausyksikön suuruinen eritaajuinen tärinä.

Anturin herkkyys ilmaisee mitattua tärinää kuvaavan volttimäärän suuruuden mittausignaaliissa. Tavallinen herkkyys kiihtyvyyssanturissa on 100 mV/g. Mitä suurempi tämä luku on, sitä herkempi anturi on kyseessä. Vaimeiden värähtelyjen mittauksessa on aiheellista käyttää erittäin herkkää anturia (1000 mV/g) ja voimakasta tärinää mitattaessa on taas aiheellista käyttää vähemmän herkkää anturia.

Anturin mittauskiteeltä (pietsosähköinen kide) saatava mittausignaali on erittäin heikkoa varausta. Varausta on vahvistettava ennen sen viemistä mittalaitteeseen. Tavallisesti kiihtyvyyssantureissa on riittävän hyvä sisäinen vahvistin. Joidenkin anturien kanssa on käytettävä ulkoista vahvistinta. Varaus muutetaan vahvistimessa jännitteeksi, joka on yleisin mittalaitteissa käytetty sisäänmeno. [3, s. 62–63.]

5 Signaalin käsittely

5.1 Signaalinkäsittelyn perusteet

Kohteen mitattu värähtelykuvio koostuu monen yksilöllisen värähtelyn lineaarisesta suodattumisesta ja päällekkäisyydestä (superimpositio). Tämä johtuu siitä, että koneet koostuvat usein monesta osasta, jotka välittävät elastisesti muilta osilta tullutta värähtelyä ja värähtelevät itsekin. Päällekkäisyydet luovat niin sanotun poimun, jossa eri viiveet tai eri suuntariippuvaiset vaimennukset omaavat signaalin osatekijät ovat kasautuneet. Alkuperäiset signaalin komponentit voivat olla hankalia tunnistaa aikasignaalista.

Signaalin käsittelyn vahvuutena on kyky pilkkoa päällekkäinen signaali osatekijöihinsä. Osatekijät on helpompi yhdistää tiettyyn koneen tilaan, osaan tai prosessiin. [3, 62–97; 4, s 19.1–19.27; 8.]

Näytepituus

Näytepituus määrittää aikatason kuvaamiseen käytettyjen näytepisteiden määrän. Näyte on otos mitatusta signaalista. Korkeilla mitattavilla taajuuksilla tarvitaan lyhyt näytepituus. Sen on oltava sitä pidempi, mitä parempi mittauksen erotuskyky on.

Analogi/digitaali muunnos

Analogi/digitaali (A/D) -muunnos on prosessi, jossa analoginen näytesignaali muutetaan digitaaliseksi. Näyte viipaloidaan hetkellisiksi arvoiksi, jotka muutetaan digitaaliseksi biittimuunnoksen sallimalla tarkkuudella. Näytteen viipalointi suoritetaan yleensä 1024 näytepisteeksi. Mikäli näyte viipaloidaan useampaan näytepisteeseen, saadaan tarkempi mittaustulos tai voidaan käyttää pidempää näytettä. Digitointi tapahtuu yleensä 12 tai 14 bitin muunnoksena. Mitä enemmän bittejä on käytössä, sitä parempi muunnoksen signaali/kohinasuhde saadaan. [3, s. 65].

5.2 Fourier'n muunnos

Tärkeimpänä signaalinkäsittelyn keinona on Fourier'n muunnos FFT (Fast Fourier Transform) -muodossa. Fourier-analyysin olennaisena merkityksenä on osoittaa minkä tahansa signaalin olevan eri amplitudisten ja taajuuksisten sinimuotoisten harmonisten signaalien summa. Tässä esitysmuodossa on sama tiedollinen sisältö kuin alkuperäisessä signaalissa. Jos kaksi eri koneenosaa aiheuttaa eri jaksolliset värähtelyt, jotka ovat superimposoituneet yhdeksi summa-aalloksi, näyttää Fourier'n muunnos kaksi eri komponenttia.

Käytännössä oikea Fourier'n muunnos ja Fourier'n sarja eivät sovellu oikeiden mitattujen signaalien tarkasteluun, sillä signaalin mittaamiseen ei ole äärettömästi aikaa, eikä järjestelmien muistista löydy äärettömästi muistipaikkoja äärettömän korkeiden taajuuksien Fourier-kertoimille. Tästä syystä käytännön sovelluksissa käytetään diskreettiä Fourier-muunnosta (DFT). DFT on määritelty diskreetille, jaksolliselle signaalille, jolla on äärellinen määrä diskreettejä taajuuden komponentteja. "Diskreetti" tarkoittaa sitä, että signaalia tarkastellaan tasaisin aikavälein, yleensä suoraan A/D -muuntajalla (I/O -terminaalissa). [3, s. 62–78; 4, s. 19.1–19.27; 8; 17, s. 170–217.]

5.2.1 FFT

FFT on DFT:lle luotujen algoritmien joukko, jotka toteutetaan erityisen tehokkaasti ja johtavat samaan numeeriseen ratkaisuun kuin DFT. FFT:llä muunnos aikatasosta taajuustasoon (spektriin) suoriutuu erittäin nopeasti.

Mikäli tarvitaan pitkän signaalin FFT-analyysiä, signaalia ei voida yksinkertaisesti jakaa osiin ja muuntaa DFT:llä. Jos jonkin osan viimeinen arvo ei vastaa sen ensimmäistä arvoa, FFT tulkitsee sen epäjatkuvuudeksi säännöllisessä signaalissa. Tämä epäjatkuvuus ilmenee selkeästi spektristä (spektrivuoto). Tämä voidaan korjata Welchin menetelmällä. [3, s. 62–78; 4, s. 19.1–19.27; 8; 17, s. 170–217.]

5.2.2 Welchin menetelmä

Welchin menetelmä on pitkien signaalien käsittelyyn tarkoitettu menetelmä, jossa signaali hajotetaan limittäisiin osiin. Hajottamisen jälkeen signaali kerrotaan ikkunafunktiolla (window function) ja muunnetaan DFT:llä. Menetelmä välttää DFT:n vaatimuksena olevasta jatkuvuudesta johtuvat virheet. Yleensä käytetty ikkunafunktio on "Hann window". Sisääntuloarvojen häviämisen estämiseksi signaalin osien limittäisyydeksi asetetaan 50 %. [3, s. 62–78; 4, s. 19.1–19.27; 8; 17, s. 170–217.]

5.2.3 DFT:n rajoitukset

Suurin käytännön rajoitus DFT:lle on yksilöllisesti kuvattujen taajuuksien rajallisuus. Nyquistin teoreeman mukaan signaali voidaan kuvata selvästi, jos korkein taajuus on enintään puolet mittaustaajuudesta. Suurempien taajuuksien läsnäolo johtaa niin sanottuun laskostumiseen (aliasing). Korkeat taajuudet suodatetaan signaalista yleensä ennen A/D-muunnosta muunneltavilla suodattimilla. [3, s. 62–78; 8; 17, s. 170–217.]

6 Laakerin vikaantuminen

Laakeri voi vioittua monella tavalla. Luonnollinen tapa laakerin vioittumiselle on materiaalin väsyminen kontaktipinnoilla kuormituksen aikaisien kovien rasituksien johdosta. Ajan kanssa väsyminen johtaa halkeamiin rakenteessa ja lopulta murtumiin. Materiaalin väsyminen on hyvin tunnettu ilmiö teoriassa, ja sitä voidaan selittää tilastollisesti. Oikein mitoitetulle ja hyvin huolletulle laakerille voidaan ennustaa hyvinkin pitkää käyttöikää. Todellisuudessa käyttöikä on lyhempi, mutta sitä ei voida tarkasti ennustaa seuraavien syiden johdosta:

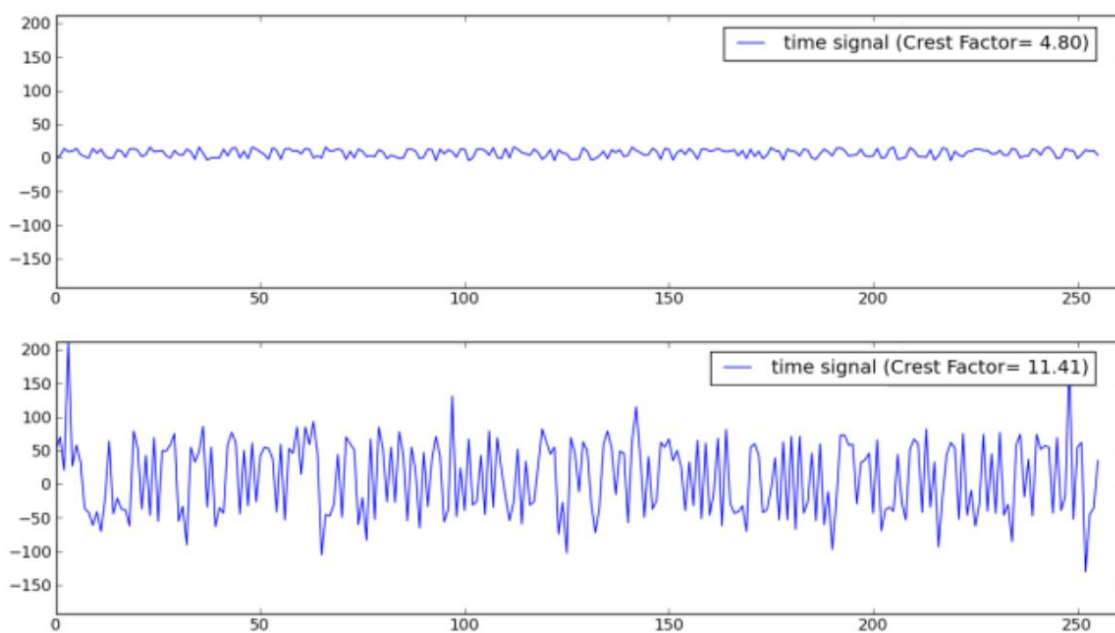
- jännitykset pyörimiselementeissä ja kulkupinnoilla kasvavat merkittävästi väärin suoritettun voitelun vuoksi, sillä voiteluaine jakaa jännityksiä ja estää ylikuumenemista
- lian ja ylimääräisten aineiden päätyminen laakerin sisään huonojen tiivisteiden vuoksi, veden pääseminen laakeriin heikentää myös voitelua
- laakerin asennon virheet ja asennusvaiheen virheet aiheuttavat varioivia jännityksiä
- liialliset jännitykset johtavat plastisiin muodonmuutoksiin
- sähköisissä sovelluksissa virran kulku voi tuhota kulkupinnat
- korroosio voi aiheuttaa alustavan vaurion pintaan.

Viat laakerissa liikkuvat jaksollisesti alueiden läpi, joissa niihin kohdistuu suurempia jännityksiä. Tämä aiheuttaa erittäin lyhyitä iskuja, joiden amplitudi ylittää huomattavasti normaalin värähtelyn amplitudin. Iskuja ei ole helppo huomata Fourier'n spektrissä, sillä sykäksen Fourier'n muunnos on leveäkaistainen spektri. Parametrit, jotka arvoivat impulsin kohinasisältöä, johtavat tulkittaviin tuloksiin. Tällaisia parametreja ovat huippukerroin ja huipukkuus. [8].

6.1 Huippukerroin

Huippukerroin (Crest factor) on yleisin kohinatason arviointiin käytetty parametri. Sen arvo on maksimiampplitudi A_P jaettuna tehollisarvolla A_{rms} . Huippukerroin ilmoitetaan desibeleissä (dB), ja sen arvo on nollaa suurempi numero. Iskut alkavasta viasta kasvattavat huippukerrointa.

Huippukerroin voidaan laskea hyvin tehokkaasti ja sitä on helppo tulkita. Esimerkiksi kuvassa 12 ylempi kuvaaja kuvaa hieman kulunutta laakeria ja alempi kuvaaja kuvaa pahasti vioittunutta laakeria. Vaurioista johtuvat huiput ovat helposti tunnistettavissa. Ehjän laakerin huippukerroin on 4,8 dB ja alemman vastaava parametri on 11,4 dB, joka kertoo vaurioiden olemassaolosta. Huippukerroin voidaan myös asettaa helposti kaavioon, jolla seurataan laakerin kunnan kehittymistä. [3, s. 62–97; 4, s 19.1–19.27; 8.]



Kuva 12. Hieman kuluneen (ylempi) ja pahasti vaurioituneen laakerin (alempi) huippukertoimet [8].

Oikein käyttämisen vuoksi on hyvä ymmärtää myös huippukertoimen rajoitukset. Maksimiampplitudit vaikuttavat paljon huippukertoimeen, eikä se ole tästä syystä vakaa muuttuja tilastollisesti. Kun signaaliampplitudi ylittää A/D -muuntajan ylärajan, maksimiarvo ei enää kasva. Tästä syystä huippukertoimen arvo laskee taas kun vauriot edistyvät laakerissa, koska tehollisarvo kasvaa. Tämä ilmiö voidaan estää ohjaamalla A/D -muuntajaa niin, että amplitudille jää tilaa kasvaa.

Huippukerroin on kuitenkin hyvä vioittumisen aikaiseen havaitsemiseen. kertoimen alenemista ei kannata tulkita niin, että laakerin kunto on paranemassa. Arvon alentuminen voi myös johtua tasaisen värinän kasvusta resonanssien tai irrallisten osien johdosta. [3, s. 62–97; 4, s 19.1–19.27; 8.]

6.2 Huipukkuus

Joissain tapauksissa huippukertoimen rajattu vakaus, joka johtuu sen riippuvuudesta maksimiampplitudiin, luo heikkouden jota tehokas laskenta ei kompensoi. Näissä tapauksissa voidaan vaihtoehtoisesti käyttää huipukkuutta.

Jos tilastollisen muuttujan arvot ovat x_n ja arvojen kokonaismäärä on N , määrittyy huipukkuus K seuraavasti keskiarvosta μ , varianssista ρ ja arvojen taajuuden jakautumisen neljännessä keskeismomentista m_4

$$\text{Keskiarvo } \mu = \frac{(\sum x_n)}{n}, \quad (25)$$

$$\text{Varianssi } \rho = \sum (x_n - \mu)^2, \quad (26)$$

$$\text{Neljäs momentti } m_4 = \sum (x_n - \mu)^4, \quad (27)$$

$$\text{Huipukkuus } K = \frac{m_4}{\rho^2} - 3. \quad (28)$$

Huipukkuus kuvailee jakauman äärimmäisien arvojen suhdetta keskiarvoon ja keskiarvon heilahteluun tai amplitudien varianssia. Huipukkuuden arvot, jotka ylittävät 7, ovat vahva merkki vaurioista. [3, s. 62–97; 4, s 19.1–19.27; 8.]

6.3 Verhokäyräanalyysi

Huippukerroyin ja huipukkuus tuottavat aikaisen merkin vioittumisesta hyvin pienellä investoinnilla. Koneiden purku ja osien tutkiminen fyysisesti aiheuttavat aina lisäkuluja. Joissain tapauksissa, ottaen huomioon, että laakereita voi olla hyvinkin monia, on hyvä olla keino lisädiagnostiikkaan. Lisädiagnostiikalla tulee voida tunnistaa tarkemmin vioittuneet laakerit tai jopa laakerin osat.

Vikojen havaitseminen tehdään iskusykäyksistä, jotka johtuvat vaurioista kontaktipinnoissa. Pyörivän kappaleen vaurioituessa sykäykset esiintyvät jaksollisesti. Jakson pituus riippuu taajuudesta jossa vaurio koskettaa kontaktipintaa. Tämä iskusykäyksen jakso riippuu laakerin pyörimisnopeudesta ja koneenosan rakenteesta.

Sykäykset koostuvat kahdesta päällekkäisestä osasta: korkeataajuisesta osasta, joka johtuu värähtäneen koneenosan värähtelystä, ja matalataajuisesta osasta, joka sisältää sykäyksien jaksollisia toistumia. Korkeataajuinen osa on poimuuntunut ja saattaa olla myös muuntunut matalataajuisen osan vaikutuksesta. Matalataajuiset osat voidaan määrittää laskemalla verhokäyrä. [8.]

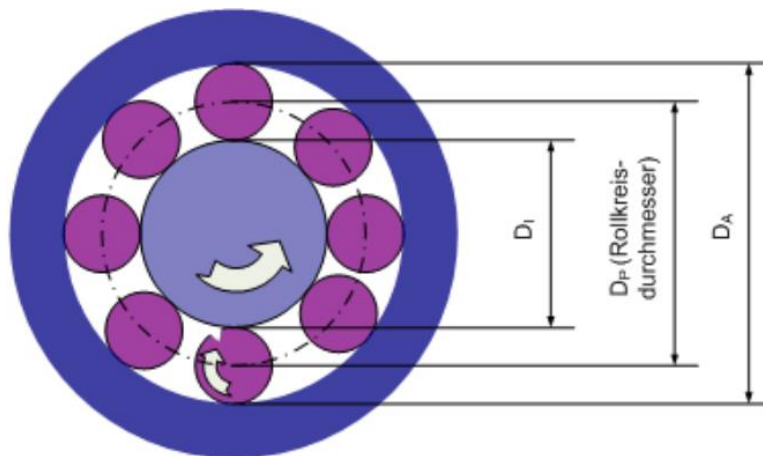
Verhokäyrä voidaan laskea suorittamalla Hilbertin muutos värähtelyalueelle (Hilbert Transformation). Hilbertin muutos määrittää tehokkaasti värähtelysignaalista 90 astetta vaihesiirretyn signaalin.

Verhokäyräanalyysi on erityisen hyödyllinen, kun määritetään mitkä koneenosat tai laakerin komponentit ovat viallisia. Tiettyjen signaalinosien arvioiminen ja interferoivien osien poisjättäminen edesauttavat vaurioiden aikaista havaitsemista. Mikäli verhokäyräanalyysiä halutaan käyttää vaurioiden aikaiseen havaitsemiseen, seurattavat vauriotaajuudet täytyy määrittää laakerin rakenteesta.

Verhokäyrä on rakenteeltaan samanlainen kuin tavallinen tehosppektri, jolloin sitä voidaan jatkoanalysoida FFT:llä. Iskusykäykset ovat FFT:n jälkeen havaittavissa selkeänä kampamaisena sekvenssinä, jossa on terävät maksimit ja tasaiset välit. Perustaajuus on maksimien välinen etäisyys. Sen käänteisluku on jaksonaika, jonka yksikkö on aikaero. Akselin pyörimisnopeuden, vauriotaajuuksien nopeussuhteiden ja huippujen jaksonajan avulla voidaan määrittää, mitkä komponentit ovat vaurioituneet. [8.]

6.4 Laakerien vauriotaajuudet

Kuvassa 13 näkyy yksinkertaisen laakerin rakenne. Laakerin alaosassa olevan laakerinosan yläreunassa on vaurio. Joka kerta kun vauriokohta pyörähtää jonkin muun laakerinosan kontaktipinnan yli, syntyy iskusysäys. Sysäyskohta liikkuu laakerinosien liikkussa toistensa suhteen. Oikein toimivassa laakerissa ei ole juuri lainkaan luistoa, vaan osat pyörivät toisiinsa kohden lähes täydellisesti, joten kontaktipinnan kulmanopeus voidaan määrittää. [8.]



Kuva 13. Laakerin rakenne ja laskutoimituksissa käytetyt nimikkeet [8].

Oletetaan, että akselin nopeus f_A on mitattu ja laakerin osien mitat ovat

- sisäkehän halkaisija D_I
- ulkokehän halkaisija D_A
- kuulien halkaisija D_B
- nousun halkaisija D_P .

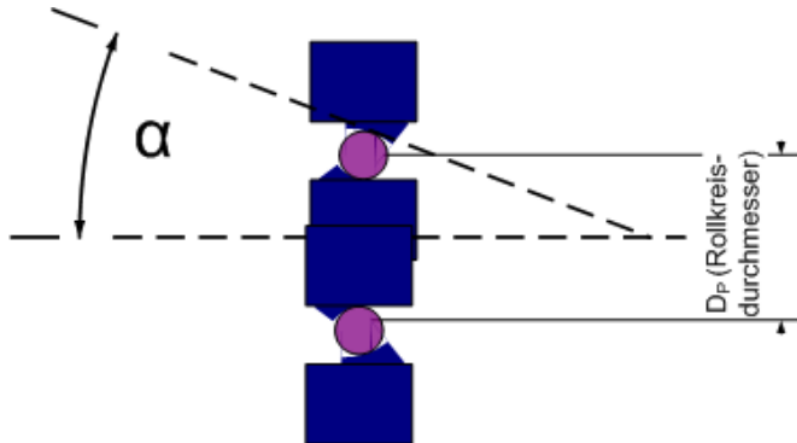
Nousun halkaisija lasketaan kaavan 29 mukaan

$$D_P = \frac{D_I + D_A}{2}. \quad (29)$$

Laakerin kuulien lukumäärä on Z . Niiden laakerien kohdalla joihin kohdistuu aksiaalisia kuormia, tulee pallojen halkaisija korjata pallojen ja kulkupinnan kohtauskulmalla α (kuva 14), laskukaava on

$$D_b = \cos(\alpha) * D_B. \quad (30)$$

Radiaalilaakereissa kohtauskulma on 0° .



Kuva 14. Laakerin rakenne ja laskutoimituksissa käytetyt nimikkeet [8].

Laakerin sisäkehän pyöriessä nopeudella f_A , voidaan sysäystaajuudet laskea. [8.]

Näistä taajuuksista voidaan käyttää seuraavia lyhenteitä:

- **BPFO** (Rolling element pass frequency outer race): taajuus, jolla pyörimiselementit pyörähtävät ulkokehän ohi
- **BPFI** (Rolling element pass frequency inner race): taajuus, jolla pyörimiselementit pyörähtävät sisäkehän ohi
- **BSF** (Bearing spin frequency rolling elements): taajuus, jolla pyörimiselementit pyörähtävät kehän suhteen
- **BPF** (Ball pass frequency): taajuus, jolla pyörimiselementissä oleva vika ohittaa kehän
- **FTF** (Fundamental train frequency): häkin pyörimisnopeus tai laakerinosan modulointitaajuus.

Taajuudet voidaan laskea seuraavien kaavojen avulla

$$BPFO = Z * \frac{f}{2} * (1 - \frac{D_b}{D_P}), \quad (31)$$

$$BPFI = Z * \frac{f}{2} * (1 + \frac{D_b}{D_P}), \quad (32)$$

$$BSF = \frac{f}{2} * \frac{D_P}{D_B} * (1 - (\frac{D_b}{D_P})^2), \quad (33)$$

$$BPF = 2 * BSF, \quad (34)$$

pyörivälle sisäkehälle:

$$FTF = \frac{f}{2} * (1 - \frac{D_b}{D_P}), \quad (35)$$

pyörivälle ulkokehälle:

$$FTF = \frac{f}{2} * (1 + \frac{D_b}{D_P}), \quad (36)$$

$$\frac{BPFI+BPFO}{f} = Z. \quad (37)$$

Käytännössä kaavoista seuraa pieniä poikkeamia, sillä esimerkiksi kohtauskulma voi vaihdella kuorman vaikutuksessa. Hyvänä muistisääntönä on, että arvoa

$$f_{BPFI} = 0,6 * f * Z, \quad (38)$$

käytetään usein sisäkehän vaurioita ilmaisevana taajuutena ja

$$f_{BPFO} = 0,4 * f * Z, \quad (39)$$

ulkokehän vaurioille. Laakereiden rakenteen varmistamiseksi, kannattaa turvautua laakerin valmistajan tietoihin. Voi myös olla hyödyllistä käyttää laskentaohjelmia, joita laakerien valmistajat tarjoavat internetissä. [8.]

7 Beckhoff TF3600 TC3 Condition Monitoring -järjestelmä

7.1 Käyttötarkoitus

Beckhoff tarjoaa laitteisto- ja ohjelmistojoukkoa, joka muodostaa kunnonvalvontajärjestelmän (Beckhoff TF3600 TC3 Condition Monitoring). Järjestelmää voidaan käyttää esimerkiksi värähtelytaajuuksien analysointiin, laakerien tarkkailuun tai vaihteistojen tarkkailuun. Järjestelmä suorittaa tarvittavan signaalinkäsittelyn. Eri algoritmit ovat saatavissa PLC -funktioblokkeina. Funktio blokit ovat koodinkatkelmia, joiden sisäänmenoon syötetään jokin arvo, ja ulostulosta saadaan jokin toinen arvo.

Beckhoff TF3600 TC3 Condition Monitoring -kunnonvalvontajärjestelmän hyvänä puoleena on kunnonvalvontapuolen integraatio normaaliin koneen ohjaimen. Tällöin ei tarvita alisysteemejä eikä ristiinviestintää. Ohjelmointi tapahtuu yhdellä päätteellä. EtherCAT:llä tarvitaan vain yksi kenttäväylätermiinaali molempiin, ohjaamiseen ja mittaamiseen. Kuvassa 15 näkyy, mitä eri toimintoja järjestelmä sisältää. [24.]

Algorithms/Features:	TF3600 Condition Monitoring Level 1
Power Spectrum	✓
Magnitude Spectrum	✓
Envelope	✓
Envelope Spectrum	✓
Power Cepstrum	✓
Analytic Signal	✓
Crest Factor	✓
Moment Coefficients (Mean, Standard Deviation, Skew, Kurtosis)	✓
Histogram	✓
Time based RMS	✓
Integrated RMS	✓
Multiband RMS	✓
Quantiles	✓
Discrete Classification	✓
Watch Upper Thresholds	✓
ArgSort	✓
Downsampling	✓

V1.5

Kuva 15. TF3600 Condition Monitoring -järjestelmän tason 1 toiminnot [8].

7.2 Käyttöönotto

Käyttöönotossa tulee tuntea Beckhoffin TwinCat3 PLC -ohjelma, sekä tietää hieman Beckhoffin kenttäväylälaitteiston käyttämisestä.

7.2.1 Logiikka ja liitännät

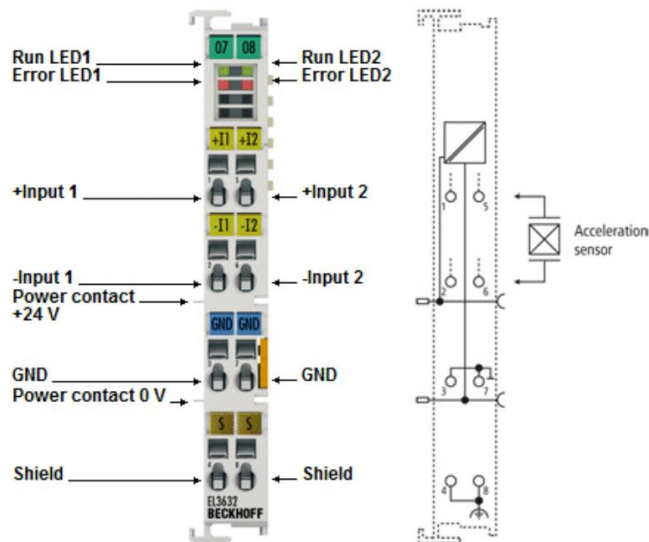
Kiihtyvyyssanturi tulee valita käyttökohteen mukaan, mutta kun varsinaista käyttökohdetta ei ole, valitaan mahdollisimman edullinen ja laajan mittausalueen omaava pietsosähköisesti toimiva kiihtyvyyssanturi. Valitaan MIP:n valmistama KS78.10 IEPE -kiihtyvyyssanturi, jonka mittausalue on ± 500 g, taajuusalue 0,6–18 000 Hz ja herkkyys 10 mV/g. Valittu anturi näkyy kuvassa 16.



Kuva 16. Valittu anturi KS78.10.

Kiihtyvyyssanturilta tuleva signaali johdetaan EL3632 EtherCAT -terminaaliin (kuva 17). Terminaaliin voidaan liittää kaksi anturia, joilta tulee + ja – merkkistä signaalia. Kiihtyvyyssanturi, jossa on IEPE -käyttöliittymä, eli pietsosähköisellä kiteellä toimiva, voidaan liittää suoraan terminaaliin.

Tarvitaan EK1100 EtherCAT -kytkin, jolla EK3632 -terminaali voidaan liittää tietokoneeseen, jossa kunnonvalvonta suoritetaan. Kaikki Beckhoffin kytkimet ja terminaalit kiinnittyvät samaan standardi-kiskoon ja toisiinsa. Kun osat liitetään toisiinsa, syntyy automaattisesti virtapiiri, joka syöttää kaikille osille kytkimeen syötetyn 24 V:n jännitteen.



Kuva 17. EL3632 2-kanavainen analoginen sisääntulotermiäli [23].

Anturilta tulevat + ja – merkkiset tulot liitetään kaapeleilla EL3632:ssa sijaitseviin +Input1 ja –Input1 kohtiin. Kaikki tarvittava liitännät on nyt tehty ja logiikkaan voidaan syöttää virtaa. Valojen vilkkumisen päätyttyä EL3632 -terminaaliin jää palamaan vain Error LED2 (ei liitetty mitään), eli liitännät on suoritettu onnistuneesti. Valmista logiikkapakettia esitellään kuvassa 18. [23.]



Kuva 18. Käyttövalmis PLC-rauta.

7.2.2 Kunnonvalvontamenetelmän päättäminen

Logiikan ohjelmointi kunnonvalvontaan tapahtuu Beckhoffin TwinCAT 3 -ohjelmalla, jossa itse ohjelman kirjoittaminen voidaan suorittaa eri ohjelmointikielillä, joita ovat

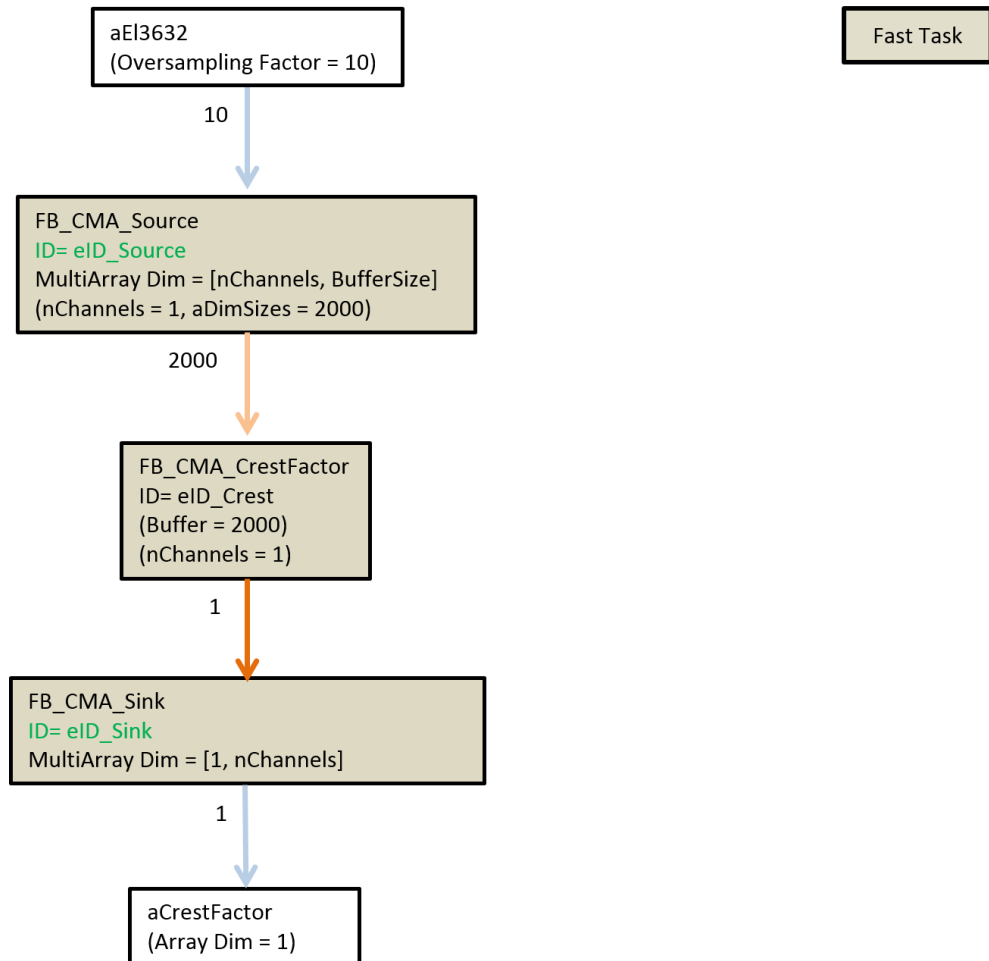
- strukturoitu teksti (ST)
- funktioblokkidiagrammi (FBD)
- tikapuulogiikkadiagrammi (LD)
- sekvenssikaavio (SFC).

TwinCAT 3 sisältää kaikille tarvittaville muutoksille ja funktioille valmiit funktioblokit. Funktioblokkeja voidaan käyttää myös muissa ohjelmointikielissä, kuin funktioblokkidiagrammeja. Strukturoitu teksti on yleispätevin ja toiminnaltaan hyvin lähellä paremmin tunnettuja ohjelmointikieliä, kuten C++, joten valitaan käytettäväksi strukturoitu teksti.

Ennen kun ohjelmointi voidaan aloittaa, on syytä miettiä, minkälaista kunnonvalvontaa halutaan suorittaa. Tässä tapauksessa kun antureita on vain yksi, on syytä valita huippukertoimen tai huippukkuuden tarkastelu. Tarkempi vian sijainnin määrittely vaatii useamman kiihtyvyyssanturin kohteessa. Jos ei odoteta suuria iskusykäyksiä tai värähtelyamplitudin kasvua, on huippukertoimen tarkastelu riittävä kunnonvalvonnan menetelmä. [8; 20; 21; 22.]

7.2.3 Ohjelman kirjoittaminen

Yksinkertaistettu ohjelman rakenne on kuvan 19 mukainen.



Kuva 19. Yksinkertaistettu ohjelman rakenne

Seuraavaksi esitetään ohjelman tärkeimpiä osia. Aluksi tärkeät parametrit kuten datapuskurin pituus ja kanavien lukumäärä määritetään globaaleihin muuttujiin (esimerkkikoodi 1).

```
VAR_GLOBAL CONSTANT
cOversamples: UDINT := 10; //Oversamplejen lukumäärä
cChannels: UDINT := 1; //kanavien määrä, 1 anturi
cBufferLength: UDINT := 2000; //datapuskurin pituus
END_VAR
```

Esimerkkikoodi 1. GVL_constants tiedoston sisältö.

Pääohjelmassa MAIN (PRG) esitellään muuttujia esimerkkikoodin 2 mukaisesti. MAIN (PRG) ohjelma on kokonaisuudessaan liitteenä 1.

```

PROGRAM MAIN
VAR CONSTANT
cInitSource: ST_MA_MultiArray_InitPars

:= ( eTypeCode := eMA_TypeCode_LREAL, nDims := 2,
aDimSizes := [cChannels, cBufferLength]);

END_VAR

VAR

nInputSelection: UDINT := 1;
aCrestFactor: ARRAY[1..cChannels] OF LREAL;
nSampleIdx: UDINT;
nChannelIdx: UDINT;
aEl3632 AT %I*: ARRAY[1..cChannels] OF ARRAY[1..cOversamples] OF
INT; // EL3632 tuleva sisääntulo
aBuffer: ARRAY[1..cChannels] OF ARRAY[1..cOversamples] OF LREAL;

fbSource: FB_CMA_Source := (stInitPars := cInitSource, nOwnID :=
eID_Source, aDestIDs := [eID_Crest]); // Lähteen alustus

fbSink: FB_CMA_Sink := (nOwnID := eID_Sink);

aFunctionGenerator: ARRAY[1..cChannels] OF FB_FunctionGenerator;

fSetFrequency: LREAL := 20; // Hz
fSetAmplitude: LREAL := 5; // sini-signaalin korekin amplitudi

fbErrHist: FB_HRErrHist; // virheiden seuranta
bNewResult: BOOL;
END_VAR

```

Esimerkkikoodi 2. Muuttujien esittely pääohjelmassa.

Datan kerääminen pääohjelmassa tapahtuu esimerkkikoodin 3 mukaisesti. Koodissa on rakennettu funktioblokki strukturoidulla tekstillä. Kuvassa 20 on sama koodi funktioblokkidiagrammi -muodossa (ylempi).

```

fbSource.Input2D(
pDataIn:= ADR(aBuffer),
nDataInSize:= SIZEOF(aBuffer),
eElementType:= eMA_TypeCode_LREAL,
nWorkDim0:= 0,
nWorkDim1:= 1,
pStartIndex:= 0,
nOptionPars := 0 );

```

Esimerkkikoodi 3. datan kerääminen lähteestä.

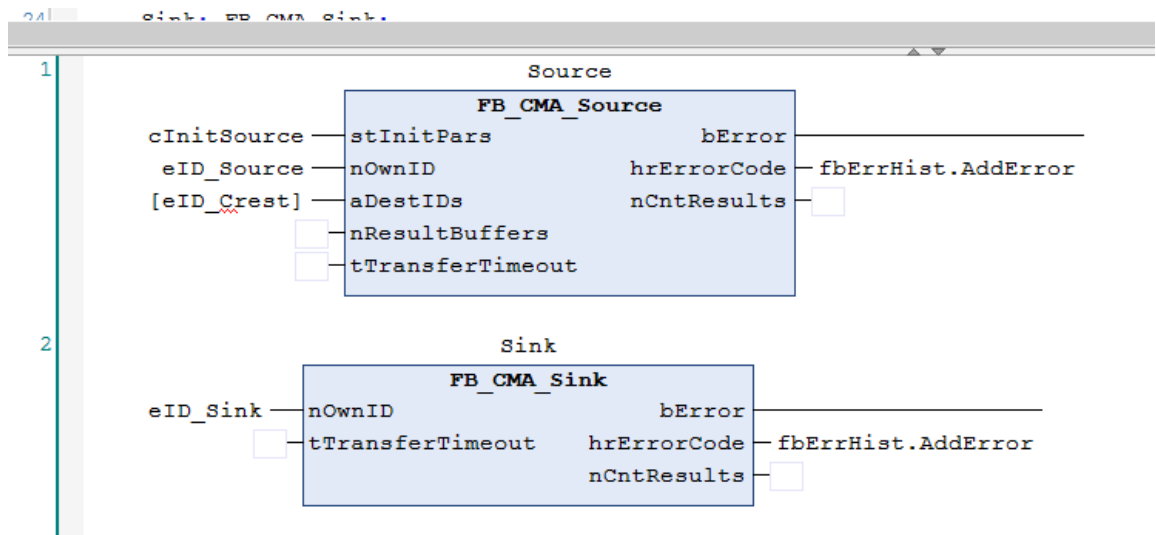
Tulosten ilmoittaminen tapahtuu esimerkkikoodin 4 mukaisesti. Tässäkin tapauksessa on rakennettu funktioblokkii strukturoidulla tekstillä. Funktioblokkidiagrammi -muodossa kyseinen funktioblokki on nähtävissä kuvassa 20 (alempi).

```

fbSink.Output1D(
pDataOut:= ADR(aCrestFactor),
nDataOutSize:= SIZEOF(aCrestFactor),
eElementType:= eMA_TypeCode_LREAL,
nWorkDim:= 0,
nElements:= 0
pStartIndex:= 0,
nOptionPars:= 0,
bNewResult => bNewResult);

```

Esimerkkikoodi 4. Datan ilmoittaminen Sink:llä.



Kuva 20. Source ja sink funtioblokit funktioblokkidiagrammi muodossa.

Vianmäärityksen pääohjelmassa MAIN_CM määritetään muuttujia esimerkkikoodin 5 mukaisesti. MAIN_CM -ohjelma on kokonaisuudessaan liitteenä 2.

```

PROGRAM MAIN_CM
VAR CONSTANT
    cInitCrest: ST_CM_CrestFactor_InitPars
    := ( nChannels := cChannels, nBufferLength := cBuffer-
        Length );
END_VAR

VAR
    fbCrest: FB_CMA_CrestFactor
    := (stInitPars := cInitCrest, nOwnID:= eID_Crest, aDes-
        tIDs:= [eID_Sink]); // Huippukertoimen alustus

    fbErrHist: FB_HRErrHist; // virheiden valvonta
END_VAR
  
```

Esimerkkikoodi 5. Muuttujat MAIN_CM -ohjelmassa.

MAIN_CM -ohjelmassa huippukertoimen arvoa kutsutaan komennolla "fbCrest.Call();".

Kunnonvalvontaohjelman luominen on hyvin monivaiheinen ja vaikea prosessi, jossa tarvitaan TwinCAT 3:n tuntemusta ja hyvää ohjelmointitaitoa. Ohjelman kirjoittajan täytyy myös ymmärtää erittäin hyvin värähtelyn teoriaa ja signaalinkäsittelyä. Toimivan ohjelman luominen edellyttää, että kirjoittaja luo itse funktioblokkit virheiden historiatietojen kirjaamiselle ja funktioiden luomiselle. Kirjoittajan täytyy myös luoda datatyyppeiksi virhehistoriatiedoille, funktiotyypeille ja analyysin tunnisteille.

Yksinkertaistettuna ohjelmassa ensin otetaan EL3632 -terminaalilta tuleva data. Sen jälkeen dataa käsitellään FB_CMA_Source -funktioblokillä. Blokin ulostulo käsitellään FB_CMA_CrestFactor -funktioblokkissa, siitä siis luodaan huippukerroinanalyysi. Analyysiltä saatu arvo asetetaan historiatietoihin. Jonossa tuleva seuraava signaali käsitellään samalla tavalla, verrataan aiempaan ja asetetaan historiatietoihin. Jos huippukertoimen arvossa havaitaan muutoksia, saadaan tieto vikatilasta.

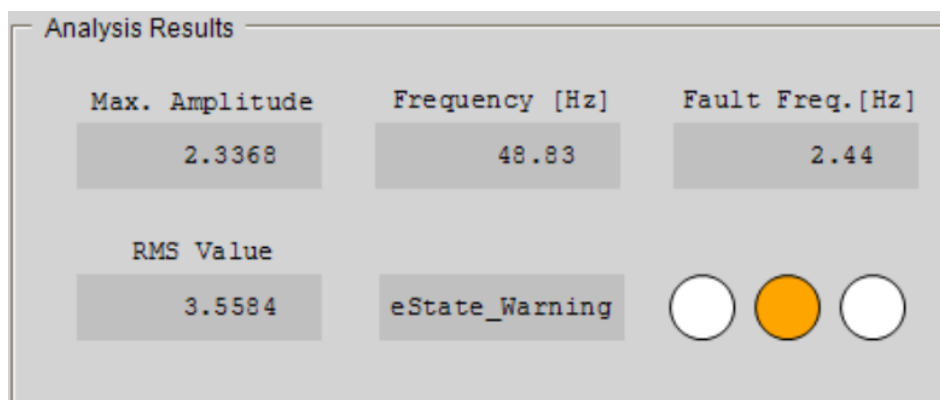
Funktiogeneraattorin funktioblokin koodi on nähtävissä kokonaisuudessaan liitteessä 3. Liitteestä 4 löytyvät funktiogeneraattorin käyttämien apufunktioiden koodit kokonaisuudessaan.

Historiatietoja kirjaavan funktioblokin koodi löytyy liitteestä 5 ja sen käyttämät metodit liitteestä 6.

Määritellyt datatyyppeiksi (DUTs) löytyvät liitteestä 7. [8; 20; 21; 22.]

7.3 Käyttö

Käyttäjälle, kuten esimerkiksi operaattori, voidaan kunnonvalvonta tehdä hyvinkin helppoksi. Ohjelma voi luoda esimerkiksi monitorille kuvan 21 mukaisen visualisoinnin, jossa on värähtelyn tärkeimmät arvot ja varoitusvalot, jotka kertovat kohteen ehjyydestä (vihreä), vikaantumisesta (keltainen) ja vaurioitumisesta (punainen). Vaihtoehtoisesti Beckhoffin kunnonvalvontasovellutuksen ollessa yhteensopiva yrityksen ohjaussysteemien kanssa, voidaan vikatilojen ilmetessä ohjata esim. fyysisiä valoja ja hälytyksiä. Vaurioitumisen ilmetessä voidaan esimerkiksi ajaa prosessi hallitusti alas ennen kuin vaurio aiheuttaa vaaratilanteita.



Kuva 21. Esimerkki visualisoinnista [20].

8 Yhteenveto

Tässä työssä tutustuttiin Beckhoff TF3600 TC3 Condition Monitoring kunnonvalvontajärjestelmään ja sen käyttämisessä tarvittaviin taustatietoihin. Työn alkuosassa käsiteltiin kunnonvalvontaa ja sen menetelmiä sivuten hieman kunnossapitoa. Seuraavaksi työssä käsiteltiin värähtelyn teoriaa kohtuullisen laajasti. Työn keskiosassa keskityttiin antureihin, ja niistä tarkimmin kiihtyvyyssantureihin, sekä signaalinkäsittelyn pääkeinoihin. Lopuosassa tutustuttiin laakerien vikaantumiseen ja vikaantumisen vaikutukseen signaaliin sekä niihin signaalinkäsittelyn keinoihin, jotka soveltuvat parhaiten laakerivian tunnistamiseen. Työn lopussa käsiteltiin itse kunnonvalvontajärjestelmää ja koottiin yksinkertainen mittauslaite Beckhoffin komponenteista.

Jo värähtelymittauksista sekä signaalinkäsittelystä on valtavasti tietoa tarjolla, eikä aika vain yksinkertaisesti riittänyt kaikkien aiheiden kokonaisvaltaiseen käsittelyyn. Työn tavoitteena oli tarjota tarvittavat taustatiedot mahdollisimman niukasti, jotta työ pysyisi sulavana ja virtaviivaisena. Taustatiedot kuitenkin kattavat suurimman osan työn sisällöstä ja vielä olisi voinut kertoa paljon enemmän.

Työssä ongelmana oli testattavan sovelluksen puuttuminen. Kunnonvalvontajärjestelmän konkreettinen käyttäminen olisi luonut lisää ajatuksia ja mielipiteitä järjestelmästä. Päällimmäiseksi työstä jäi mieleen järjestelmän käyttöönoton monimutkaisuus.

Värähtelyteorian ja signaalinkäsittelyn vankka tuntemus on ehtona Beckhoffin kunnonvalvontajärjestelmän käyttöönotolle. Käyttäjälle voidaan tosin luoda yksinkertainen käyttöliittymä, josta vikaantuminen ilmenee selvästi. Hyväksi ominaisuudeksi työssä todettiin kunnonvalvontajärjestelmän yhteensopivuus Beckhoffin ohjausjärjestelmien kanssa.

Tätä työtä voidaan hyödyntää kunnonvalvonnan, värähtelyteorian ja signaalinkäsittelyn taustatietojen omaksumisessa, sekä TF3600 TC Condition Monitorin kunnonvalvontajärjestelmään tutustumisessa.

Lähteet

- 1 Järviö, Jorma – Piispa, Taina – Parantainen, Timo – Lappalainen, Markku – Åström, Thomas. 2006. *Kunnossapito*. 3. uudistettu painos. Hamina. KP-Media Oy.
- 2 PSK 6201. 2011. *Kunnossapito. Käsitteet ja määritelmät*. 3. painos. PSK Standardisointiyhdistys ry.
- 3 Nohynek, Petri – Lumme, Veli Erkki. 2004. *Kunnonvalvonnan värähtelymittaukset*. 2. täydennetty painos. Hamina. KP-Media Oy.
- 4 Piersol, Allan G. – Paez, Thomas L. 2010. *Harris' shock and vibration handbook*. McGraw-Hill Co.
- 5 Peltonen, Hannu – Perkkiö, Juha – Vierinen, Kari. 2004. *Insinöörin (AMK) fyysiikka, osa II*. 6. painos. Saarijärvi. Saarijärven OFFSET Oy.
- 6 Mäkelä, Mikko – Soininen, Lauri – Tuomola, Seppo – Öistämö, Juhani – Kulmala, Marko. 2010. *Tekniikan kaavasto*. 9. painos. Hämeenlinna. Amk-Kustannus Oy Tammertekniikka.
- 7 PSK 5701. 2015. *Kunnonvalvonnan värähtelymittaus. Käsitteet ja määritelmät. Käytettävät suureet ja mittayksiköt*. 7. painos. PSK Standardisointiyhdistys ry.
- 8 Beckhoff Automation GmbH & Co. 2016. *TF3600 TC3 Condition Monitoring, Frequency analysis* [verkkodokumentti]. Saatavissa: https://infosys.beckhoff.com/english.php?content=../content/1033/tf3600_tc3_condition_monitoring/1162492299.html&id=. [Luettu 13.5.2016].
- 9 PSK 5702. 2007. *Kunnonvalvonnan värähtelymittaus. Mittauspisteen valinta ja tunnistaminen*. 3. painos. PSK Standardisointiyhdistys ry.
- 10 PSK 5705. 2006. *Kunnonvalvonta. Värähtelymittaus. Mittaustoiminnan suunnittelu*. 5. painos. PSK Standardisointiyhdistys ry.
- 11 PSK 5707. 2011. *Kunnonvalvonnan värähtelymittaus. Vianmääritys*. 5. painos. PSK Standardisointiyhdistys ry.
- 12 PSK 5708. 2003. *Kunnonvalvonta. Värähtelymittaus. Rakenteelliset värähtelyominaisuudet*. 2. painos. PSK Standardisointiyhdistys ry.
- 13 SFS-ISO 15242-1. 2012. *Rolling bearings. Measuring methods for vibration. Part 1: Fundamentals*. Metalliteollisuuden Standardisointiyhdistys ry.

- 14 SFS-ISO 15242-2. 2012. *Rolling bearings. Measuring methods for vibration. Part 2: Radial ball bearings with cylindrical bore and outside surface*. Metalliteollisuuden Standardisointiyhdistys ry.
- 15 SFS-ISO 15242-3. 2012. *Rolling bearings. Measuring methods for vibration. Part 3: Radial spherical and tapered roller bearing with cylindrical bore and outside surface*. Metalliteollisuuden Standardisointiyhdistys ry.
- 16 SFS-ISO 15242-4. 2012. *Rolling bearings. Measuring methods for vibration. Part 4: Radial cylindrical roller bearing with cylindrical bore and outside surface*. Metalliteollisuuden Standardisointiyhdistys ry.
- 17 SFS-ISO 15243. 2012. *Rolling bearings. Damage and failures. Terms, characteristics and causes*. Metalliteollisuuden Standardisointiyhdistys ry. Launonen, Eero – Sorvali, Esko – Toivonen, Pertti. 2006. *Sarjat ja F-, L- ja Z-muunnos*. 1. Painos. Helsinki. WSOY Oppimateriaalit Oy.
- 18 Hemilä, Simo – Utriainen, Juha. 1991. *Värähtelyt ja aallot*. 3. muuttumaton painos. Helsinki. Otatieto Oy.
- 19 Kivioja, Seppo. 2009. *Konetekniikka*. 7. painos. Helsinki. Otatieto Oy.
- 20 Beckhoff Automation GmbH & Co. 2016. *TF3600 TC3 Condition Monitoring with Frequency Analysis* [verkkodokumentti]. Saatavissa: http://infosys.beckhoff.com/english.php?content=../content/1033/tf3600_tc3_condition_monitoring/9007200417328139.html&id=. [Luettu 13.5.2016].
- 21 Beckhoff Automation GmbH & Co. 2016. *TF3600 TC3 Condition Monitoring Crest Factor - Sample 2* [verkkodokumentti]. Saatavissa: http://infosys.beckhoff.com/english.php?content=../content/1033/tf3600_tc3_condition_monitoring/1162590219.html&id=17711341208151324514. [Luettu 13.5.2016].
- 22 Beckhoff Automation GmbH & Co. 2016. *TF3600 TC3 Condition Monitoring Multi-Channel Spectrum - Sample 3* [verkkodokumentti]. Saatavissa: http://infosys.beckhoff.com/english.php?content=../content/1033/tf3600_tc3_condition_monitoring/18014399672073739.html&id=24224868552542723514. [Luettu 13.5.2016].
- 23 Beckhoff Automation GmbH & Co. 2016. Beckhoff Information System [verkkodokumentti]. Saatavissa: http://infosys.beckhoff.com/english.php?content=../content/1033/tf3600_tc3_condition_monitoring/1162492299.html&id=23221917910915512316. [Luettu 13.5.2016].
- 24 Beckhoff Automation GmbH & Co. 2016. Beckhoff [verkkodokumentti]. Saatavissa: <http://www.beckhoff.fi/>. [Luettu 13.5.2016].

MAIN (PRG)

```
PROGRAM MAIN
VAR CONSTANT
    cInitSource: ST_MA_MultiArray_InitPars
    := ( eTypeCode := eMA_TypeCode_LREAL, nDims := 2, aDimSizes :=
[cChannels, cBufferLength]);

END_VAR

VAR
    nInputSelection: UDINT := 1;
    aCrestFactor: ARRAY[1..cChannels] OF LREAL;
    nSampleIdx: UDINT;
    nChannelIdx: UDINT;
    aEl3632 AT %I*: ARRAY[1..cChannels] OF ARRAY[1..cOversamples] OF
INT; // EL3632 tuleva sisääntulo
    aBuffer: ARRAY[1..cChannels] OF ARRAY[1..cOversamples] OF LREAL;

    fbSource: FB_CMA_Source := (stInitPars := cInitSource, nOwnID :=
eID_Source, aDestIDs := [eID_Crest]); // Lähteen alustus

    fbSink: FB_CMA_Sink := (nOwnID := eID_Sink);

    aFunctionGenerator: ARRAY[1..cChannels] OF FB_FunctionGenerator;

    fSetFrequency: LREAL := 20; // Hz
    fSetAmplitude: LREAL := 5; // sini-signaalin korekin amplitudi

    fbErrHist: FB_HRErrHist; // virheiden seuranta
    bNewResult: BOOL;

END_VAR



---


FOR nChannelIdx := 1 TO cChannels DO
    CASE nInputSelection OF
    0: (* input from EL3632 *)
        FOR nSampleIdx := 1 TO cOversamples DO
```

```
        aBuffer[nChannelIdx][nSampleIdx]:=
        INT_TO_LREAL(aE13632[nChannelIdx][nSampleIdx]);
    END_FOR

    1: (* signal generator *)
        aFunctionGenerator[nChannelIdx](bStart:= TRUE,

        fFrequency:= fSetFrequency,

        fAmplitude:= fSetAmplitude,

        eFunction := E_FunctionType.eSine,

        aSignal:= aBuffer[nChannelIdx] );
    END_CASE
END_FOR

// Datan keruu lähteeltä
fbSource.Input2D(pDataIn:= ADR(aBuffer),
nDataInSize:= SIZEOF(aBuffer),
eElementType:= eMA_TypeCode_LREAL,
nWorkDim0:= 0,
nWorkDim1:= 1,
pStartIndex := 0,
nOptionPars := 0 );
IF fbSource.bError THEN
fbErrHist.AddError(bErr:=TRUE,          hrErrCode:=fbSource.hrErrorCode,
sErrSource:='MAIN.fbSource'); // virheiden valvonta
```

```
//Tulosten syöttäminen sinkkiin
fbSink.Output1D( pDataOut := ADR(aCrestFactor),
                nDataOutSize := SIZEOF(aCrestFactor),
                eElementType:= eMA_TypeCode_LREAL,
                nWorkDim:= 0,
                nElements:= 0,
                pStartIndex:= 0,
                nOptionPars:= 0,
                bNewResult => bNewResult);

IF fbSink.bError THEN
    fbErrHist.AddError(bErr:=TRUE,          hrErrCode:=fbSink.hrErrorCode,
                      sErrSource:='MAIN.fbSink'); // virheiden valvonta
END_IF

IF bNewResult THEN
END_IF
```

MAIN_CM (PRG)

```
PROGRAM MAIN_CM
```

```
VAR CONSTANT
```

```
    cInitCrest: ST_CM_CrestFactor_InitPars  
    := ( nChannels := cChannels, nBufferLength := cBufferLength );
```

```
END_VAR
```

```
VAR
```

```
    fbCrest: FB_CMA_CrestFactor := (stInitPars := cInitCrest, nOwnID  
    := eID_Crest, aDestIDs:= [eID_Sink]); // Huippukertoimen alustus
```

```
    fbErrHist: FB_HRErrHist; // virheiden valvonta
```

```
END_VAR
```

```
// Kutsutaan huippukerroin laskentaan
```

```
fbCrest.Call();
```

```
IF fbCrest.bError THEN
```

```
    fbErrHist.AddError(bErr:=TRUE,      hrErrCode:=fbCrest.hrErrorCode,  
    sErrSource:='MAIN_CM.fbCrest'); // virheiden valvonta
```

```
END_IF
```

FB_Function Generator (FB)

```
FUNCTION_BLOCK FB_FunctionGenerator

VAR_INPUT

    bStart: BOOL;
    fAmplitude: LREAL;      // huipusta-huippuun
    fFrequency: LREAL;     // Vaihteleva
    fOffset: LREAL;
    nDutyCycle: UINT;
    eFunction: E_FunctionType;

END_VAR

VAR_IN_OUT

    aSignal: ARRAY[1..cOverSamples] OF LREAL;

END_VAR

VAR_OUTPUT

    bError: BOOL;

END_VAR

VAR

    bInit: BOOL;
    bEnable: BOOL;
    fLimitMinCount: LREAL := 1;
    nSamples: UDINT;
    fCycleTime: LREAL;
    fDelta: LREAL;
    fLastFrequency: LREAL;
    fPhase: LREAL;
    nSign: INT := 1;
    fDefaultCarrier: LREAL := 700;
    fModulationSignal: LREAL := 1;
    fModulationSignalIncrement: LREAL;
    fLastSignalValue: LREAL;
    fPeriod: LREAL;
    fAngularIncrement: LREAL;
    fAngularIncrementPerCycle: LREAL;
```



```

fAngularIncrementPerSample := fAngularIn-
crementPerCycle/cOversamples;

bInit := FALSE;

bEnable := TRUE;

ELSIF eFunction = E_FunctionType.eSquare THEN

    fPeriod := 1/fFrequency;

    fAngularIncrementPerCycle := 0;

    fAngularIncrementPerSample:=
    (1/UDINT_TO_LREAL(cOversamples));

    // Vaiheen muuttaminen

    IF nDutyCycle <> 0 THEN

        fPhase:=(UDINT_TO_LREAL(nDutyCy-
        cle)/100)*(fLim-
        itMinCount/(fCycleTime*fFrequency));
        // (fLim-
        itMinCount/(2*fCycleTime*fFrequency))* (UIN
        T_TO_LREAL(nDutyCycle)/100);

        ELSE

            fPhase := (fLim-
            itMinCount/(2*fCycleTime*fFrequency));

        END_IF

        bInit := FALSE;

        bEnable := TRUE;

    ELSIF eFunction = E_FunctionType.eTriangular THEN

        fPeriod := 1/fFrequency;

        fAngularIncrementPerSample := (fLim-
        itMinCount);

        // Vaiheen muuttaminen

        // fPhase := (fLim-
        itMinCount/(4*fCycleTime*fFrequency));

        fPhase:=fLimitMinCount*(UDINT_TO_LREAL(cO-
        versamples)/(4*fCycleTime*fFrequency));

        fDelta:=-1*ABS((fAmplitude-fOff-
        set)/fPhase);

        bInit := FALSE;

        bEnable := TRUE;

    ELSIF eFunction = E_FunctionType.eAM THEN

        fModulationSignal := fFrequency;

        fAngularIncrementPerCycle := 1;

```

```
fAngularIncrementPerSample := (fAngularIncrementPerCycle/cOversamples)*fLimitMinCount;

// Vaiheen muuttaminen
fPhase:=(fLimitMinCount/(fCycleTime*fDefaultCarrier));

bInit := FALSE;
bEnable := TRUE;

ELSE
    bError := TRUE;
END_IF

ELSE
    bError := TRUE;
END_IF

ELSE
    bError := TRUE;
END_IF

ELSIF fLastFrequency <> fFrequency THEN
    bInit := TRUE;
END_IF

// Aktivoidaan funktiogeneraattori
IF bEnable THEN
    CASE eFunction OF
        E_FunctionType.eSine: sine();

        E_FunctionType.eSquare: square();

        E_FunctionType.eSawtooth: sawtooth();

        E_FunctionType.eTriangular: triangular();

        E_FunctionType.eAM: am();

    END_CASE
END_IF

// Automaattinen alustus
fLastFrequency := fFrequency;
```

Funktiogeneraattorin apufunktiot: am, sawtooth, sine, square ja triangular

AM

```
FOR nSamples := 1 TO cOversamples DO
    IF (LREAL_TO_INT(fAngularIncrement) = fPhase) THEN
        fAngularIncrement := 0;
    END_IF
    aSignal[nSamples] := SIN(2*PI*(fModulationSignalIncrement
        + (nSamples-1)*(360*(fCycleTime*fModulationSignal)/cOversamples)/360)*fAmplitude*SIN(2*PI*(fAngularIncrement/fPhase));
    fAngularIncrement := fAngularIncrement + fAngularIncrementPerSample;
END_FOR

fModulationSignalIncrement := fModulationSignalIncrement +
360*(fCycleTime*fModulationSignal);
```

Sawtooth

```
FOR nSamples := 1 TO cOversamples DO
    IF (fAngularIncrement = fPhase) THEN
        fAngularIncrement := 0;
        fLastSignalValue := 0;
    END_IF
    aSignal[nSamples] := fLastSignalValue + (fDelta/cOversamples);
    fAngularIncrement := fAngularIncrement + fAngularIncrementPerSample;
    fLastSignalValue := aSignal[nSamples];
END_FOR
```

Sine

```

FOR nSamples := 1 TO cOversamples DO
    aSignal[nSamples] := fAmplitude*SIN(2*PI*(fAngularIn-
        crement+(nSamples-1)*fAngularIncrementPerSample)/360
END_FOR
fAngularIncrement := fAngularIncrement + fAngularIncrementPerCycle;

```

Square

```

FOR nSamples := 1 TO cOversamples DO
    IF nDutyCycle = 0 THEN
        IF (fAngularIncrement < fPhase) THEN
            fAngularIncrement := fAngularIncrement +
                fAngularIncrementPerSample;

            ELSE
                nSign := -1*nSign;
                fAngularIncrement := 0;
                fAngularIncrement := fAngularIncrement +
                    fAngularIncrementPerSample;
            END_IF
        ELSE
            IF (fAngularIncrement < fPhase) THEN
                nSign := 1;
                fAngularIncrement := fAngularIncrement +
                    fAngularIncrementPerSample;

                ELSIF (fAngularIncrement >= fPhase) AND
                    (fAngularIncrement < fPeriod*1000) THEN
                    nSign := 0;
                    fAngularIncrement := fAngularIncrement +
                        fAngularIncrementPerSample;
                ELSE
                    fAngularIncrement := 0;
                END_IF
            END_IF
            aSignal[nSamples] := nSign*fAmplitude;
        END_FOR
        nCount := nCount + 1;

```

Triangular

```
FOR nSamples := 1 TO cOversamples DO
    IF (ABS(fDebug) = 0) AND fDelta < 0 THEN
        nSign := ABS(nSign);
        fDelta := ABS(fDelta);
    ELSIF (ABS(fDebug) = LREAL_TO_INT(fPhase)) AND fDelta >
    0 THEN
        nSign := -1*ABS(nSign);
        fDelta := ABS(fDelta);
    ELSIF (ABS(fDebug) = LREAL_TO_INT(fPhase)) AND fDelta <
    0 THEN
        nSign := ABS(nSign);
        fDelta := -1*ABS(fDelta);
    ELSIF (ABS(fDebug) = 0) AND fDelta > 0 THEN
        nSign := -1*ABS(nSign);
        fDelta := -1*ABS(fDelta);
    END_IF
    aSignal[nSamples] := fLastSignalValue +
    nSign*(ABS(fDelta));
    fAngularIncrement := fAngularIncrement + nSign*fAngu-
    larIncrementPerSample;
    fDebug := fAngularIncrement;
    fLastSignalValue := aSignal[nSamples];
END_FOR
```

FB_HRErrHist (FB)

```
FUNCTION_BLOCK FB_HRErrHist

VAR_INPUT
    bEdgeDetect : BOOL;
END_VAR
VAR_OUTPUT
    nErrSum      : UINT;
    nErrIdx      : UINT;
    aErrList     : ARRAY[1..cHistLen] OF ST_HRErrHist;
END_VAR

VAR
    bNewErr      : BOOL;
    fbErrTrig    : R_TRIG;
END_VAR

VAR CONSTANT
    cHistLen     : UINT := 100;
END_VAR
```

FB_HRErrHist:n metodit AddError ja Reset

AddError

```
METHOD AddError
VAR_INPUT
    bErr                : BOOL;
    hrErrCode           : HRESULT;
    sErrSource          : STRING(31);
END_VAR


---


fbErrTrig(CLK:=bErr);
IF bEdgeDetect THEN
    bNewErr := fbErrTrig.Q;
ELSE
    bNewErr := bErr;
END_IF

IF bNewErr THEN
    nErrSum := nErrSum + 1;
    IF nErrIdx < cHistLen THEN
        IF nErrIdx = 0 THEN
            nErrIdx:= nErrIdx + 1;
        ELSIF aErrList[nErrIdx].hrErrCode <>
            hrErrCode OR aErrList[nErrIdx].sErrSource
            <> sErrSource THEN
            nErrIdx := nErrIdx + 1;
        END_IF
        aErrList[nErrIdx].hrErrCode := hrErrCode;
        aErrList[nErrIdx].sErrSource := sErrSource;
        aErrList[nErrIdx].nErrCount:=
            aErrList[nErrIdx].nErrCount + 1;
    END_IF
END_IF
```


Reset

```
METHOD Reset
VAR_INPUT
END_VAR


---


nErrSum := 0;
nErrIdx := 0;
MEMSET(ADR(aErrList),0,SIZEOF(aErrList));
```

Datatyyppiyksiköt (DUTs)

ST_HRErrHist (STRUCT)

```
TYPE ST_HRErrHist :  
STRUCT  
{attribute 'displaymode':='hex'}  
    hrErrCode    : HRESULT;  
    sErrSource   : STRING(31);  
    nErrCount    : UDINT;  
END_STRUCT  
END_TYPE
```

E_FunctionType (ENUM)

```
TYPE E_FunctionType :  
(  
    eSine,  
    eSquare,  
    eTriangular,  
    eSawtooth,  
    eAM  
);  
END_TYPE
```

E_AnalysisIDs (ENUM)

```
TYPE E_AnalysisIDs :  
(  
    (* analyysi ketju *)  
    eID_Source := 1,  
    eID_Crest,  
    eID_Sink  
  
) UDINT;  
END_TYPE
```