



Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

VILLE SIURO

Robotin laakeroinnin kunnonvalvonta

SÄHKÖ- JA AUTOMAATIOTEKNIIKAN
KOULUTUSOHJELMA

2020

Tekijä(t) Siuro, Ville	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä toukokuu 2020
	Sivumäärä 24	Julkaisun kieli Suomi
Julkaisun nimi Robotin laakeroinnin kunnonvalvonta		
Tutkinto-ohjelma Sähkö- ja automaatiotekniikan koulutusohjelma		
<p>Tiivistelmä</p> <p>Tässä työssä toteutettiin robotin laakeroinnin kulumista seuraava järjestelmä. kunnonvalvonta olemassa olevilla mekaanisilla komponenteilla ja ohjausjärjestelmillä. Tavoitteena oli toteuttaa yksinkertainen, mutta luotettava ohjelma seuraamaan muutoksia robotin akselien servovahvistimien käyttämästä tehosta.</p> <p>Työ toteutettiin ZenRobotics Oy:n Fast Picker robottilaitteistolla. Kyseinen robotti oli käytettävissä tuotekehitykseen, joten se ei ollut varsinaisessa tuotantoajossa.</p> <p>Työn suunnittelu aloitettiin tutkimalla alkuperäistä liikkeenohjausta. Tämän jälkeen luotiin testaussuunnitelma mahdollisista tulevista muutoksista ja sen avulla aloitettiin luomaan tallennettavien muuttujien valikoimista.</p> <p>Työssä kerättiin paljon dataa laitteiston akselien Homing- ajosta. Dataa tutkimalla toteutettiin varsinainen kunnonvalvontaohjelma. Kunnonvalvonta perustuu liukuvaan keskiarvoon viimeiseltä viideltä mittauskerralta, joten yksittäinen virhe ei aiheuta hälytyksiä tai varoituksia.</p> <p>Lopputuloksena onnistuttiin luomaan ohjelma, joka simuloinnissa antoi sekä varoituksia, että hälytyksiä.</p>		
Twincat, kunnonvalvonta, robotti		

Author(s) Siuro, Ville	Type of Publication Bachelor's thesis /	Date 11/20
	Number of pages 24	Language of publication: Finnish
Title of publication Condition monitoring of the robot bearing		
Degree programme Degree Programme in Electrical and Automation Engineering		
Abstract This Thesis was made for ZenRobotics Oy. The main objective was to make a condition monitoring program with existing mechanical components and softwares. The goal was to make a simple but reliable program. The planning of the work began by examining the original motion control. A hypothesis about possible future changes was made before the work start. In the work, a lot of data was collected about the Homing run of the equipment, by examining which the actual condition monitoring program was implemented. Condition monitoring is based on a moving average from the last five measurements, so a single error does not cause alarms or warnings. As a result, a program was created that gave both warnings and alerts in the simulation.		
Twincat, condition monitoring, robot		

SISÄLLYS

1 JOHDANTO.....	5
2 TOIMEKSIANTAJA.....	6
2.1 Zenrobotics Oy.....	6
2.2 Tuotteet.....	6
2.2.1 Fast Picker.....	7
2.2.2 Heavy Picker.....	7
3 OHJELMISTOT.....	8
3.1 TwinCAT 3.....	8
3.2 Te130x Scope View.....	10
4 LIIKKEENOHJAUS.....	10
4.1 PLC.....	11
4.2 Liikeohjain.....	11
4.2.1 Servomootorit.....	11
4.2.2 Servovahvistimet.....	12
5 TESTILAITTEISTO.....	13
5.1 Robotti ympäristö.....	13
5.2 Ulkoiset vaikutukset.....	13
5.3 Testaussuunnitelma.....	13
6 DATAN ANALYSOIMINEN.....	15
6.1 Mittauksien suorittaminen.....	15
6.2 Tuloksien tutkiminen.....	18
7 PLC-OHJELMAN LUOMINEN.....	21
8 YHTEENVETO.....	25

LÄHTEET

LIITTEET

1 JOHDANTO

Tämä opinnäytetyö tehdään Zenrobotics Oy:lle. Työn tavoitteena on toteuttaa portaalirobotin laakeroinnin kuntoa mittaava järjestelmä. Kunnonvalvonnan avulla voidaan ennakoida tarvittavia huoltotoimia ennakoivasti ja tällä tavalla ehkäistä osia hajoamasta yllättäen ja odottamatta. Odottamaton pysähtyminen aiheuttaa aina tuontantokatkoksen.

Työssä selvitetään ensin olemassa olevien kunnonvalvontasovelluksien soveltavuutta laakerien kunnon mittaamiseen. Laakeroinnin kuntoa on mahdollista seurata tarkemmin erilaisilla anturoinneilla, muun muassa värähtelyantureilla.

Tässä työssä tutkitaan portaalirobotin laakeroinnin kulumista testiympäristössä. Käytettävissä on Fast Picker robotti ja testiympäristö, josta pystytään lukemaan ja tallentamaan muun muassa robotin akselin: nopeutta, kiihtyvyyttä ja lämpötilaa. Työssä käytettävät ohjelmistot ovat TwinCat 3 ja Te130x Scope View. TwinCat 3 mahdollistaa varsinaisen kunnonvalvonta sovelluksen luomisen ja Te130x Scope View- ohjelman avulla mahdollistetaan tutkittavan datan tallentaminen ja siirtäminen Excel-muotoon.

2 TOIMEKSIANTAJA

2.1 Zenrobotics Oy

ZenRobotics Oy on perustettu vuonna 2007. Yritys on keskittynyt robotiikkaan ja sen vaatimaan sovelluksien ja ohjelmistojen luomiseen. Robottien toiminta perustuu tekoälyyn ja robotit ovat suunniteltuja kierrätyslinjastoille. Konenäön ja tunnistuksen avulla järjestelmät ovat hyvin soveltuvia erilaisille kierrätysmateriaaleille. Nykyisin yritys työllistää 50 henkilöä. (Suomen Asiakastieto Oy www-sivut 2020)

2.2 Tuotteet

Yrityksellä on kaksi järjestelmä kokonaisuutta eri tilanteisiin ja asiakkaiden tarpeisiin. Robotit mahdollistavat lajittelun ympärivuorokauden itsenäisesti ja turvallisesti. Konenäkö mahdollistaa eri materiaalien tunnistuksen. Materiaalien eri laatujen tunnistaminen on myös mahdollista. Esimerkiksi muovissa: PP, PE,PVC ja PET.

Järjestelmät on mahdollista asentaa jo olemassa oleville kuljettimille korvaamaan ihmistyövoimaa ja nostamaan lajittelun tehokkuutta. Robotteihin on mahdollista luoda etäyhteys, jonka avulla voi muun muassa tehdä PLC:n päivityksiä ja ohjelmisto muutoksia.

ZenBrain on järjestelmää hallitseva tekoäly. Sen avulla voidaan opettaa roboteille erilaisten materiaalien lajittelu. Käyttäjä pystyy valitsemaan poimittavan materiaalin käyttöliittymän avulla, joka sijaitsee sähköpääkeskuksen ovelta. ZenBrain myös määrittelee poimittaville kappaleille arvon, joka mahdollistaa arvokkaimman kappaleen poiminnan.

2.2.1 Fast Picker

Tuoteperheen pienempi malli Fast Picker mahdollistaa yhden kilogramman painoisen esineen poiminnan. Robotin tarttuja on imukupityyppinen. Se on hyvin kompaktin kokoinen ja tarvitsee kuljettimelta asennustilaa ainoastaan 1944 mm x 2670 mm (pituus x leveys). Tähän sisältyy poimittavalle tavaralle kaksi pudotuskuilua, yksi kummallakin robotin puolella.

Robotin maksimi poimintanopeus on 4000 poimintaa minuutissa ja kuljettimen hihnan nopeus voi olla jopa yhden metrin sekunnissa. Puomittavan kappaleen maksimikoko on 400 mm x 400 mm x 300 mm (pituus x leveys x korkeus). Työalue on kooltaan 1200 mm x 600 mm. Kuvassa 1 on esitetty Fast Picker robotin ulkonäkö.



Kuva 1. Fast Picker robotti (ZenRobotics [www-sivut](http://www.zenrobotics.com))

2.2.2 Heavy Picker

Suurempien kappaleiden poimintaan on kehitetty tuote nimeltä Heavy Picker. Poimitavan kappaleen maksimi paino on 30 kilogrammaa. Järjestelmässä voi olla yhdestä kolmeen robottia peräkkäin yhdellä tunnistuksella, jolloin järjestelmän vaatima tila kuljettimelta vaihtelee noin 6:sta metristä 14:sta metriin. Tarttuja on mekaaninen, jota ohjataan paineilmasylinterillä.

Kuljettimen maksimi nopeus on 0,7 metriä sekunnissa ja yksi robotti mahdollistaa 2000 poimintaa tunnissa. Työalue on kooltaan 2000 mm x 1800 mm. Jokaisella robotilla on 4 pudotuskuilua, kaksi kummallakin puolella. Järjestelmään on mahdollista lisätä syöttävien kuljettimien nopeuden hallinta nostamaan tehokkuutta. Kuvassa 2 on esimerkki Heavy Picker asennuksesta. (ZenRobotics www-sivut)

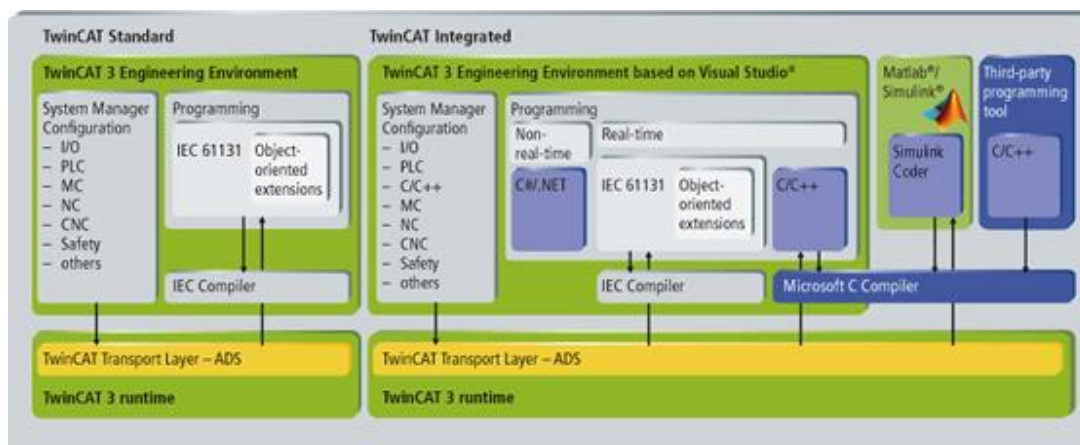


Kuva 2. Tunnistus-yksikkö ja kolme Heavy Picker robottia. (ZenRobotics www-sivut)

3 OHJELMISTOT

3.1 TwinCAT 3

TwinCAT on Beckhoff:n kehittämä ohjelmisto, joka mahdollistaa yhdellä ohjelmalla erilaisten automaatiojärjestelmien ohjauksen. Kuvassa 3 on esitetty Standart ja integrated versioiden ohjelmointi ja liitettävyydet. TwinCAT 3 tukee IEC 61131 standardia ja se on rakennettu Visual Studio:n pohjalle. Ohjelmassa on mahdollista käyttää ohjelmointikielenä vaihtoehtoja: IL, ST, FBD, LD, CFC ja SFC.



Kuva 3. TwinCAT 3 Standard ja Integrated.

Työn pohjana käytetään valmista Fast Picker projektia, johon lisätään kunnonvalvontaan tarvittavat ohjelmaosa-alueet. Projekti on kirjoitettu pääosin ST-kielellä eli Structured Text:nä. Kuvassa 4 on esimerkki yksinkertaisesta ohjelmasta ST-kiellä. Kyseisen ohjelmointikielen yhtenä hyvänä puolena voidaan pitää asioiden esittäminen lyhyesti ja ytimekkäästi. Tarkoituksena on toteuttaa mahdollisimman yksinkertainen ja vähän prosessoritehoa kuormittava ohjelma.

IF .. THEN .. ELSE

```
IF      condition= true
THEN   action1
ELSE   action2   (condition not true)
```

```
IF raining THEN Stay_In   ELSE Go_Out
```

Kuva 4 Esimerkki ST-kielestä (PLCopen www-sivut)

3.2 Te130x Scope View

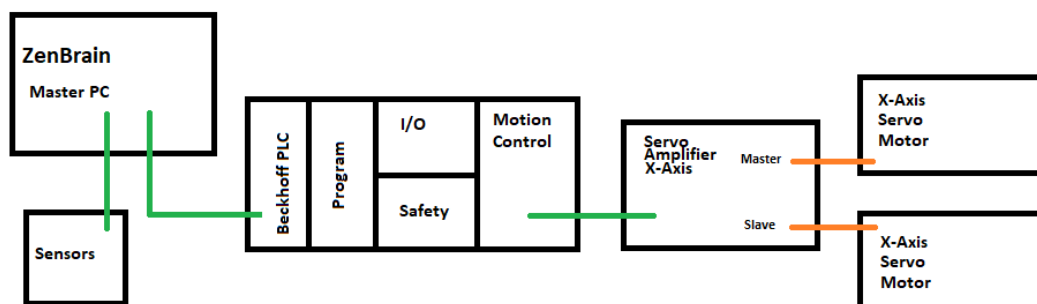
TwinCat 3 ympäristöön on kehitetty oskilloskooppi-sovellus Scope View. Sen avulla on mahdollista monitoroida PLC-puolelle tuotuja muuttujia ja nauhoittaa niitä myöhempiä tutkimuksia varten. Sovellus on rakennettu Visual Studion pohjalle, joten käytettävyys on luotu samankaltaiseksi kuin TwinCat:n.

Yhdistäminen projektiin tapahtuu PLC logiikan IP-osoitteen perusteella. Yhdistämisen jälkeen luodaan mittausprojektin muuttujien linkitys. Nauhoituksen voi aloittaa joko itse tai määrittää nauhoituksen aloittavan liipaisevan bitin ohjelmasta. Scope View mahdollistaa suuren datamäärän käsittelyn tehokkaasti ja selkeästi graafien avulla.

Tässä työssä hyödynnetään ohjelman ominaisuutta, jolla nauhoitetut muuttujien arvot on mahdollista tuoda Excel: iin .csv-muodossa. Tämän ansiosta tietynlaisen robotin liikkeen etsiminen on mahdollista nopeasti hyödyntämällä Excelin hakutoimintoa.

4 LIIKKEENOHJAUS

ZenBrain on liikeohjauksen tekoäly, joka kertoo robotille koordinaatit. Liikeradat luodaan kuljettimella olevien poimittavien kappaleiden sijainnin perusteella. Järjestelmässä oleva tietokone on yhdistettynä logiikkaan EtherCat-väylällä ja sen avulla tapahtuu mm. liikkeenohjaukseen tarvittava tiedonsiirto. Kuvassa 5 on esitetty yksinkertaistettu kaavio x-akselin ohjauksesta.



Kuva 5. EtherCat-väylä ja servokaapelointi.

4.1 PLC

Järjestelmässä päälogiikkana käytetään Beckhoff:n CX5130-logiikkaa. Se on 2-ytimisellä prosessorilla, joten sen avulla on mahdollista suorittaa aikakriittiset prosessit nopeammalla ohjelmakierrolla ja samanaikaisesti taustalla hitaammalla nopeudella muita prosesseja. Liikkeenohjaus suoritetaan nopealla ohjelmankierrolla.

Järjestelmän turvalogiikka on toteutettu Beckhoff:n EL6910 Twinsafe I/O-kortilla, joka on liitetty suoraan logiikkaan. Robottien työalue on suojattu kiinteillä rakenteilla, joten työaluetta valvotaan ovilukoilla ja rajoilla.

4.2 Liikeohjain

4.2.1 Servomoottorit

FastPicker robotissa x-akselilla käytetään kahta servomoottoria. Ne ovat Beckhoff:n AM8042-mallisia. Servomoottoreiden kytkentä servovahvistimeen tapahtuu yhdellä kaapelilla, joka sisältää sekä virransyötön, että takaisinkytkennän. Kuvassa 6 on käytetty servomoottori

Servomoottoreiden tekniset tiedot ovat:

- Vääntö: 3,70nm
- Teho: 1,94kW
- Kierrosnopeus: 5000rpm
- Jatkuva virta: 4,10A



Kuva 6. AM8042 Servomoottori (Beckhoff [www-sivut](http://www.beckhoff.com))

4.2.2 Servovahvistimet

Servomoottoreita ohjaavana servovahvistimena toimii Beckhoff:n 2-kanavainen AX5206-malli, joten yhdellä servovahvistimella voidaan ajaa x-akselin kahta servomoottoria. (Beckhoff [www-sivut](http://www.beckhoff.com))

AX5206 tekniset tiedot:

- Jännite: 3 x 100...480V AC tai 1 x 100...240V AC
- Jatkuva virta: 2 x 6A
- EtherCat-liitäntä

5 TESTILAITTEISTO

5.1 Robotti ympäristö

Työssä käytetään Fast Picker-robottia. Robotti on testikäytössä ja sillä ajetaan niin sanottuja haamupoimintoja, joten kuljettimella ei ole materiaalivirtaa. ZenBrain luo robotille työtä, jolla robotti saadaan rasiin. Testauksessa robotin haamuajoa seurataan noin neljä kuukautta. Laitteiston lähtötaso määritellään seuraamalla kaksi viikkoa robotin toimintaa ja nauhoittamalla Scope View-ohjelmalla muuttujia.

5.2 Ulkoiset vaikutukset

Testauksen luotettavuuteen on otettava huomioon myös sellaiset muuttujat, jotka eivät ole robotista johtuvia, mutta saattavat vaikuttaa tuloksiin. Hyvänä esimerkkinä voidaan pitää lämpötila muutoksia. Lämpötilan muuttuessa robotin samanlainen liike voi vaatia hyvinkin eri lailla energiaa. Huomioon on myös otettava ympäristön yleinen siisteys, huomattava määrä esimerkiksi pölyä voi nopeuttaa kulumista ja luoda kitkaa joutuessaan robotin laakerointiin.

5.3 Testaussuunnitelma

Testauksen mahdollistamiseksi kehitetään Scope View- projekti, johon lisätään liikeohjaimella käytössä olevia muuttujia. Käytössä on molempien servomoottoreiden muuttujat. Työssä tallennetaan molempien servomoottoreiden muuttujia. Master – muuttujiin vaikutetaan kulumista lisäävillä ja nopeuttavilla tekijöillä ja Slave - muuttujat toimivat referenssinä. Slave - akselin laakerointiin ei testauksessa vaikuteta, mutta mahdollisesti suoritetaan huolto mittauksien jälkeen. Testauksessa on tavoitteena kehittää ohjelma, mitä voitaisiin hyödyntää sekä Fast Picker, että Heavy Picker roboteissa ja robottien eri akseleilla.

Tallennetut muuttujat:

- SetVelocity - Nopeusohje
- SetAcceleration – Asetettu kiihtyvyys
- ActualVelocity – Todellinen nopeus
- ActualAbsoluteCurrent – Virta-arvo
- ActAcceleration – Todellinen kiihtyvyys
- VelocityFeedbackValue – Takaisinkytkennästä saatava nopeus
- TorqueFeedbackValue – Takaisinkytkentä momentista
- MotorTemperature – Moottorin lämpötila

Dataa tallennetaan robotin haamuajolla ja Homing-tilanteissa. Tallennetut liikkeet siirretään Exceeliin. Mittauksia tallennetaan 1 millisekunnin päivitystaajuudella, joten mittausdataa syntyy runsaasti. Yksittäinen Homing- ajo tuottaa noin 3000-riviä Excel-tiedostoon. Haamuajoa tallennetaan 2–5 minuutin aikajaksoja. Mittauspäivinä tallennetaan 20 kappaletta Homing- ajoja ja lisäksi noin 40 minuuttia haamuajoa.

Erilaisten tilanteiden vertailemiseksi tallennetaan muuttujia robotin moottoreiden ollessa vallitsevan ulkolämpötilan lämpöisiä sekä huomattavasti ulkolämpötilaa lämpöisempiä. Tämän avulla saadaan niin kutsuttuja kylmä ja kuuma arvoja, joiden avulla voidaan vertailla lämpötilan vaikutusta muuttujiin.

Robotin laakeroinnin kuluman oletetaan näkyvän virta- ja momentti arvoista, joten niiden muuttumista seurataan mittauspäivinä. Akseleiden välisiä eroja vertaillaan Exceeliin kerättävien keskiarvojen avulla. Virheen mahdollisuuden pienentämiseksi käytetään liukuvaa keskiarvoa, jonka avulla yksi virheellinen mittauskerta ei aiheuta varoituksia tai hälytyksiä. Robotissa käytettävä laakerointi on rakenteeltaan kestävä, joten hypoteesina pidetään suurta kestävyyttä myös ilman laakereiden voitelua ajaessa. Suurin osa kunnonvalvonta ohjelmasta tullaan toteuttamaan ST-kielillä.

6 DATAN ANALYSOIMINEN

Lähtökohtana mittauksien suorittamiseen otetaan virta-arvojen tutkiminen. PLC- ohjelmaan tehdään muutos, mikä mahdollistaa servovahvistimien muuttujien tallentamisen. Muuttujista tallennetaan sellaisia, jotka testaussuunnitelman mukaan ovat mahdollisesti muutoksille alttiita. Tallennukseen sisällytetään myös mukaan muuttujia, joita voidaan pitää referenssinä ulkoisista vaikutuksista. Referenssinä pidetään moottorista saatavaa lämpötilatietoa.

6.1 Mittauksien suorittaminen

Tuloksia tutkiessa on hyödynnettävä keskiarvoa, joten myös mittauksia suoritettiin runsaasti. Yksittäinen hetki yhden millisekunnin tarkkuudella aiheuttaa liikaa virheiden mahdollisuuksia ja tulevaisuudessa kunnonvalvonnassa turhia hälytyksiä. Mittauksien alkaessa selvitettiin ensin robotin laakeroinnin kunto silmämääräisesti. Kuvassa 7 on laakeroinnin rakenne.



Kuva 7. Portaalirobotin x-liikkeen laakerointi.

Lähtötason määrittämisen jälkeen laakeroinnin kulumisen nopeuttamiseksi laakereista poistettiin voitelu. Laakerit ovat varustettu automaattivoitelulla, laakereiden päädyissä on öljysäiliöt. Laakerit irrotettiin, poistettiin öljysäiliöt ja pestiin öljyt pois kitkan kasvattamiseksi ja kulumisen nopeuttamiseksi. Tämä suoritettiin vain robotin toisen puolen laakeroinnille, joten oli mahdollista tutkia ja vertailla servomoottoreiden välisiä eroja. Mittauksia suoritettiin ajotuntien perusteella. Öljyttömällä laakereilla ajettiin noin 60 tuntia, jonka jälkeen tallennettiin lisää dataa.

Kulumisen nopeuttamiseksi suunniteltiin uusi metodi. Seuraavassa vaiheessa laakereista poistettiin päädyissä sijaitsevat pölysuojat. Tämä mahdollisti lian päätyminen laakerien rulliin. Kyseisessä testiympäristössä ilmassa ei ole paljon pölyä ja epäpuhtauksia, sillä varsinaista materiaalivirtaa ei ole kuljettimella. Robotin ollessa tuotantokäytössä voitaisiin havaita suurempia muutoksia poistamalla voitelu ja pölysuojat.

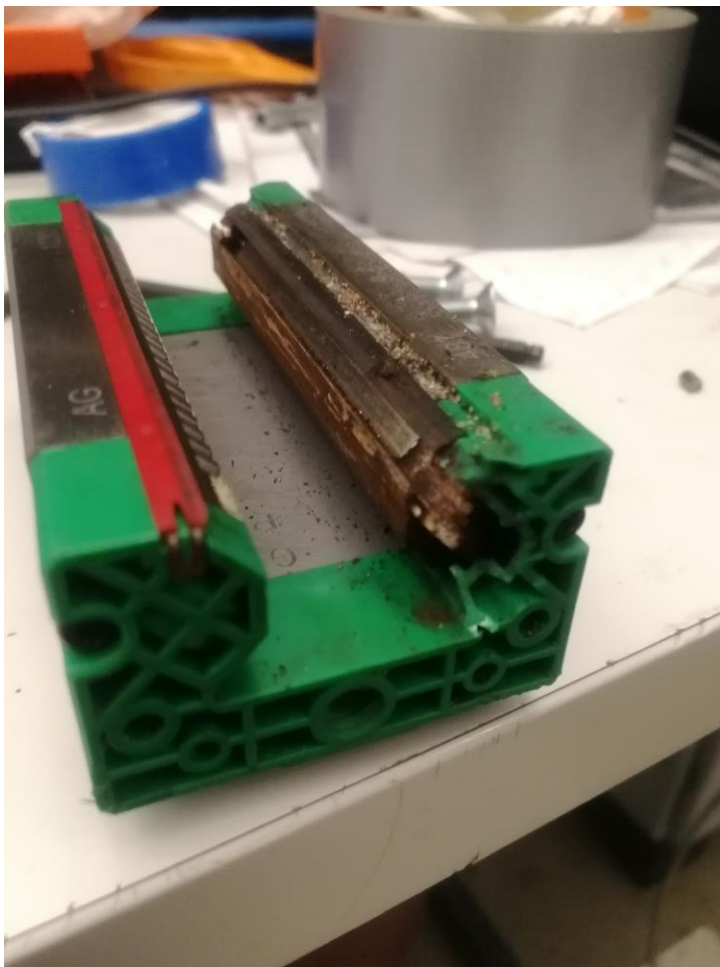
Pölysuojien poistamisen jälkeen tehtiin päätös lisätä kitkaa ja nopeuttaa kulumista ulkoisilla tekijöillä, joten päätettiin lisätä johteille hienojakoista hiekkapuhallushiekkaa. Tavoitteena oli hiekan päätyminen laakerin rullien kanssa kontaktiin, joka kuluttaisi rullia ja aiheuttaisi kitkaa.

Hiekkapuhallushiekka aiheutti halutun kaltaista kulumista, joka johti laakerin hajoamiseen. Kuvassa 8 hajonnut laakeri, sekä kulunut laakerin johde. Tämän jälkeen suoritettiin robotin huolto. Ensimmäisenä vaihdettiin robotin laakeri uusiin ja puhdistettiin johteet. Laakereihin myös asennettiin uudet öljysäiliöt ja voitelu otettiin jälleen käyttöön. Uusilla laakereilla nauhoitettiin muutokset, jonka jälkeen myös laakerin johde vaihdettiin uuteen. Laakerijohteessa oli havaittavissa kulumista ja myös suurempi vaurio laakerin hajoamiskohdassa.



Kuva 8. Robotti pysäytettyä hajoamisen jälkeen.

Vertailuun otettiin myös dataa tilanteesta, kun robottiin on tehty perusteellisempi huolto. Huollossa vaihdettiin robotin x-akselille uudet vetohihnat ja hihnojen päätyrullat. Tämän avulla voitiin suorittaa vertailua huollosta saatavista hyödyistä ja muutoksista datasta. Laakerin hajoamista tutkittiin kuvassa 9. Kulumisesta ja hajoamisesta voitiin päätellä kuluneisuuden kohdistuneen pääosin laakeroinnin yläpuolelle, johon kertyy huomattavasti helpommin kulumista edistäviä ulkoisia tekijöitä.



Kuva 9. Vaurioitunut laakeri.

6.2 Tuloksien tutkiminen

Datan analysoimisen alussa oli havaittavissa se, että helpointa ja samankaltaisen tilanteen luomiseksi on helpointa tutkia robottia sen Homing-vaiheen aikana. Tällöin robotti ajaa olemassa olevasta asemastaan päätyrajalle. Tuloksia suodatettiin vielä sellaiseen vaiheeseen, jolloin robotti on saavuttanut annetun nopeuden (nopeusohje on yhtä suuri kuin todellinen nopeus). Tässä tilanteessa kiihtyvyys on luonnollisesti nolla. Pääteltiin kyseisen tilanteen olevan helposti vertailtavissa.

Nauhoitetusta datasta tutkittiin ensin muuttujia sen avulla, että mitkä muuttuisivat ja miten helppoa muutos olisi havaita. Alussa seurattiin momentin takaisinkytkennän ja virta- arvojen muutoksia Homing- ajossa. Näistä arvoista selkeästi muutokset olivat havaittavissa, mutta virta-arvoa oli helpompi tutkia, sillä se oli jokaisessa tilanteessa

positiivinen- arvo ja akselien pyörimissuunta ei vaikuttanut tuloksiin. Momentti muuttui akselien kesken kääntäen verrannollisesti rajoissa 0–65535. Momentti- arvon hyödyntäminen vaatisi huomattavasti monimutkaisemman laskentakaavan, joten se päätettiin hylätä. Virta-arvot vaihtelivat asteikolla 0–160 tuloksien kesken ja olivat suoraan verrannollisia keskenään eri mittaustilanteissa.

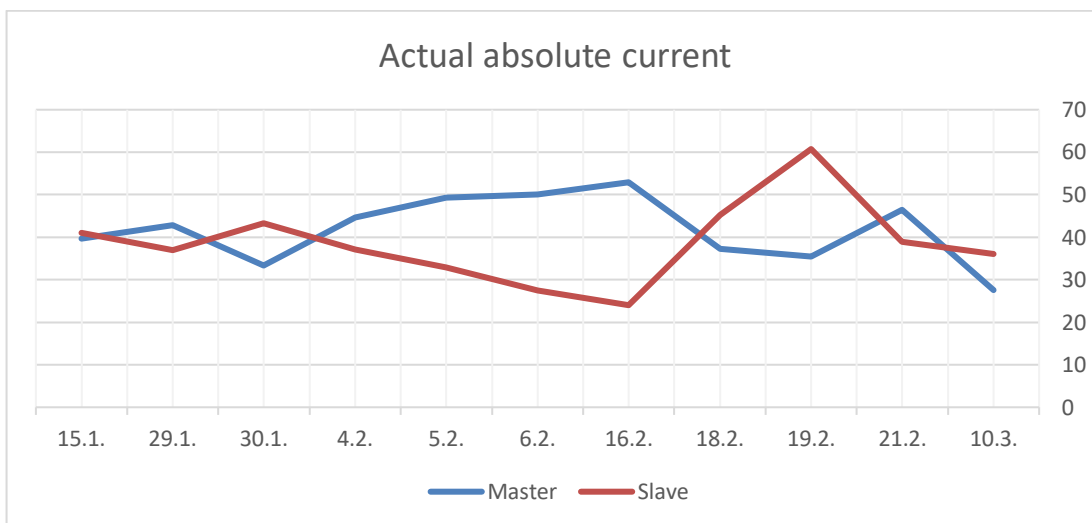
Datasta aloitettiin tutkimaan ennalta mainittuja vakionopeudella ajettuja Homing- hetkiä Excelin avulla. Nauhoituksia suoritettiin mittauspäivinä vähintään 3 kertaa robotin ollessa kylmä ja vähintään 10 kertaa robotin servomoottorin ollessa 40–50 asteinen. Kylmänä mittaukset suoritettiin servomoottorin ollessa noin 10–20 asteinen. Esimerkiksi mittauspäivänä lämpöisenä mitatuista arvoista tehtiin lopulta päiväkohtainen keskiarvo. Tämän avulla saatiin virhemarginaalia pienennettyä.

Lämpötilan muutos oli selkeästi havaittavissa. Kylmänä servojen käyttämä virta kasvoi noin 35 %, mikäli robotin laakerointi on hyvässä kunnossa. Suurimmillaan laakeroinnin ollessa rikkiäinen saatiin kylmänä servon käyttämä virta-arvo nousemaan jopa 79 % verrattuna laitteiston lämpöisenä käyttämään virtaan.

Robotin servomoottoreiden ollessa 40–50 asteisina muutokset akselien käyttämässä virrassa oli eroa vain 3 % suuruinen ehjillä laakereilla. Vastaavana lukuna akselikohdainen ero on kylmällä laitteistolla pahimmallaan noin 20 %. Tästä voitiin päätellä, että testaussuunnitelman mukaisesti lämpötilan vaikutus on huomattava myös mitatuissa arvoissa.

Tallennetuista arvoista pystyi havaitsemaan erikoisen tilanteen, sillä hetkellä kun robotin laakerointiin lisättiin hiekkaa niin yllättäen Master-akselin virta-arvo alkoi pienemään ja Slave – akselin virta-arvot nousemaan. Tästä on pääteltävissä, että laakerit aiheuttivat suurta kitkaa ja vastustivat robotin liikettä tasaisesti, joten Slave – akseli joutui huomattavasti suuremmalle rasitukselle. Robotin liikkeistä tätä ei ollut mahdollista huomata, sillä automaatiojärjestelmä mukautui tilanteeseen. Kuluneisuutta voidaan korjata liikkeenohjauksella niin kauan, kunnes akselien kesken alkaa muodostumaan jättämää. Tämän jälkeen liikeohjain antaa virheen ja vaaditaan uusi Homing-

ajo. Kuvassa 10 esiintyy päivämäärien mukaan keskiarvot servomoottoreiden kuluttamasta virrasta. Master kuvastaa hajonnutta puolta x-liikkeen laakeroinnista ja Slave on referenssinä alkutilanteen mukaisessa kunnossa oleva laakerointi.



Kuva 10. Excel kaavio virtojen keskiarvoista tutkimus ajanjaksolla.

Kaaviosta on x-akselilta luettavissa päivämäärät:

- 30.1 Laakereista on poistettu voitelu
- 16.2 Laakereihin on lisätty hienoa hiekkaa
- 19.2 Master – puolen laakerointi hajosi
- 10.3 Huolto suoritettu

Huoltamisen jälkeen uusilla laakereilla ja uudella johteella varustettu servomoottorin tarvitsema virta on pienempi kuin toisella puolella. Tästä voidaan päätellä myös, että laakeroinnin kuluneisuus havaitaan virta-arvoista. Slave-akselille ei vaihdettu laakereita, eikä johdetta.

7 PLC-OHJELMAN LUOMINEN

Plc-ohjelmaan luodaan Fast Picker-robotin käyttämään kirjastoprojektiin. Projektiin luodaan POU (a program organization unit). Tässä tapauksessa ohjelma luodaan function block:n sisään, joka mahdollistaa sisäisen muistin käyttämisen. Ohjelma on nimetty FB_Condition- nimellä.

Aiempien tutkimuksien avulla selvitettyt muuttujat tuodaan tulopuolelle VAR_INPUT otsikon alle. Lähdöt ohjelmasta määritetään VAR_OUTPUT:ksi. Ohjelman sisäisessä laskennassa käytetään VAR-muuttujia, mitkä eivät näy lohkon kutsussa päätasolla. Kuvassa 11 on esitetty ohjelman vaatimat muuttujat.

```

1 | FUNCTION_BLOCK FB_Condition
2 | VAR_INPUT
3 |     iAbsoluteCurrentRaw1 AT %I*: INT;           //set axis absolute current parameter
4 |     iAbsoluteCurrentRaw2 AT %I*: INT;           //set axis absolute current parameter
5 | END_VAR
6 | VAR_OUTPUT
7 |     OverCurrentWarning: BOOL;                   //Warning
8 |     OverCurrentAlarm: BOOL;                     //Alarm
9 |     MovingAvg: LREAL;                           //The average of the last five measurements
10| END_VAR
11| VAR
12|     NC1: AXIS_REF;
13|     fbMovingAvg: FB_MovingAverage;
14|     NewValue: BOOL;
15|     MovingAvgLimitWarning: INT:= 120;           //Warning Limit
16|     MovingAvgLimitAlarm: INT:= 135;            //Alarm Limit
17|     iAbsoluteCurrent1:LREAL;
18|     iAbsoluteCurrent2:LREAL;
19|     CurrentSummary: LREAL;
20|     CurrentAvg: LREAL;
21|     Calculator: LREAL;
22|
23| END_VAR

```

Kuva 11. Ohjelmalohkon muuttujat.

Tuloihin linkitetään ainoastaan suoraan servovahvistimelta saatava AbsoluteCurrent arvo. Tietotyyppinä Beckhoff käyttää kyseisessä muuttujassa integeriä.

Function Blockin ulkopuolelle valvonnasta saadaan kolme erilaista muuttujaa, OverCurrentWarning, OverCurrentAlarm ja MovingAvg. Varoitus- ja hälytysrajat määritellään ohjelman sisällä, joten niitä ei ole mahdollista muokata ulkopuolisesti. Mo-

vingAvg- muuttujalla on mahdollista esittää toteutunutta mittausta, jota verrataan määritettyihin rajoihin. MovingAvg on mahdollista esittää esimerkiksi käyttöliittymässä huoltonäkymässä.

VAR-muuttujia käsitellään ainoastaan FB_Condition:n sisällä. Näissä kutsutaan Beckhoff:n valmista AXIS_REF- tietorakennetta ja ZenRobotics:n luomaa FB_MovingAverage- ohjelmaa. Varoitus- ja hälytysrajat alustetaan muuttujan datatyyppin määrittämisen jälkeen.

Tarvittavien muuttujien määrittämisen jälkeen aloitetaan varsinainen ohjelman luominen. Ohjelman ensimmäisessä vaiheessa eli kuvassa 12 muutetaan AbsoluteCurrent arvot integer:stä real- arvoiksi, joka mahdollistaa niiden käyttämisen FB_MovingAverage- ohjelmassa.

```

1 // Convert all to real
2   IF iAbsoluteCurrentRaw1 < 0 THEN
3     iAbsoluteCurrent1:= 0;
4   ELSE
5     iAbsoluteCurrent1:= INT_TO_LREAL(iabsoluteCurrentRaw1);
6   END_IF
7
8
9   IF iAbsoluteCurrentRaw2 < 0 THEN
10    iAbsoluteCurrent2:= 0;
11  ELSE
12    iAbsoluteCurrent2:= INT_TO_LREAL(iabsoluteCurrentRaw2);
13  END_IF

```

Kuva 12. Muuttujien muuntaminen Integer:stä Real:ksi

Mittauksen kohdistamiseen oikeaan ajanjaksoon, käytetään AXIS_REF- tietorakenteen muuttujia. Mittauksen aloituksen tarkastukseen käytetään Homing, SetAcc ja SetVelo- arvoja. SetAcc- arvon on oltava 0 eli ei ole asetettua kiihtyvyyttä. SetVelo- arvon avulla tarkastellaan sitä, että robotti on liikkeessä ja annetussa Homing- nopeudessa. Mittauksessa lasketaan yhteen akselien virrat. Tämän lisäksi jokaisen mittauksen jälkeen kasvatetaan laskuria yhdellä, jonka avulla saadaan kirjattua

näytteiden määrä keskiarvon laskemiseksi. Nc1.Status.Homed- bitin asetettua ja laskurin ollessa > 0 , lasketaan uusi keskiarvo. Seuraavaksi asetetaan NewValue- bitti ja nollataan laskuri. Kuvassa 13 on esitetty mittauksen aloittaminen.

```

14
15 //Enable measurement
16
17 IF ncl.Status.Homing = TRUE AND ncl.NcToPlc.SetAcc = 0 AND ncl.NcToPlc.SetVelo > 0.09 THEN
18     CurrentSummary:= iAbsoluteCurrent1 + iAbsoluteCurrent2+ CurrentSummary;
19     Calculator:= Calculator + 1;
20 END_IF
21
22 IF ncl.Status.Homed = TRUE AND Calculator > 0 THEN
23     CurrentAvg:= CurrentSummary / Calculator;
24     NewValue:= TRUE;
25     Calculator:=0;
26
27 ELSE NewValue:= FALSE;
28
29 END_IF

```

Kuva 13. Nauhoittamisen päälle kytkentä ja mittauskertojen laskenta.

fbMovingAvg- ohjelman kutsuun määritetään siihen vaadittavat muuttujat tuloiksi ja lähdeiksi. NrOfSamples mahdollistaa määrittelyn, kuinka monesta viimeistä mittauksesta liukuvakeskiarvo luodaan. Tässä tapauksessa sopivaksi arvoksi määriteltiin viimeiset viisi Homing- ajoa. Ohjelman kutsu on esitetty kuvassa 14.

```

30
31 fbMovingAvg(
32     Enable:= ncl.Status.Homed,
33     Value:= CurrentAvg,
34     Tick:= NewValue,
35     NrOfSamples:= 5,
36     Avg=> );
37
38 MovingAvg:=(fbMovingAvg.Avg);
39

```

Kuva 14. Liukuvan keskiarvo-ohjelman kutsu.

Varoitus- ja hälytysrajojen vertailuun on käytetty yksinkertaista vertailua. Tallennetuista datoista määriteltiin sopivat raja-arvot, jolla testilaitteisto oli alkanut kulumaan. Hälytysrajaksi muodostui arvo, joka on hieman ennen laakerin lopullista hajoamista. Kuvassa 15 esittää ohjelmaan kirjoitetut vertailut.

```
40 // Limits
41 IF MovingAvg > MovingAvgLimitWarning THEN
42     OverCurrentWarning:= TRUE;
43 ELSE
44     OverCurrentWarning:= FALSE;
45 END_IF
46
47 IF MovingAvg > MovingAvgLimitAlarm THEN
48     OverCurrentAlarm:= TRUE;
49 ELSE
50     OverCurrentAlarm:= FALSE;
51 END_IF
```

Kuva 15. Raja-arvojen määrittely.

8 YHTEENVETO

Opinnäytetyössä onnistuttiin luomaan ohjelma, jolla voidaan tutkia robotin laakeroinnin kulumaa täysin ohjelmallisesti ilman mekaanisia muutoksia tai anturoinnin lisäämistä laitteistoon. Lämpötila ja muut ulkoiset tekijät vaikuttavat huomattavasti tuloksiin, joten erilaisiin ympäristöolosuhteisiin asennettujen robottien vertailu ei ole luotettavaa. Testiympäristössä oli mahdollista vaikuttaa robottia ympäröivän rakennuksen lämpötilaan välillä 5–30 astetta celsiusta.

Valvonnassa on hyödyllistä käyttää liukuvaa keskiarvoa, joten yksittäinen tulos ei aiheuta hälytyksiä. Asetetut varoitus- ja hälytysrajat tulee asettaa aina kyseisen laitteiston virta-arvojen mukaisesti. Esimerkiksi käyttöönoton yhteydessä määriteltävä lähtötaso. Kunnonvalvonta-ohjelma käyttää tällä hetkellä Master- ja Slave-akseleiden virtojen summaa. Fast Picker ei haamuajolla saavuta kuin muutamaksi millisekunniksi vakionopeuden, joten kunnonvalvonnassa on suositeltavaa käyttää Homing-ajotilanteita.

Opinnäytetyössä uutena oppimisena tuli Scope View- ohjelman käyttäminen ja sen liittäminen laajaan TwinCat-projektiin. Työssä hyödynnettiin myös etäyhteyden käyttämistä testilaitteistolle, joten ohjelmamuutoksien lataaminen ei edellyttänyt laitteiston luona käymistä. Datan tallennusta myös testattiin etäyhteyden avulla. Projektin ohjelmoinnissa ST-kielen käyttäminen oli suuressa ja sen myötä siihen perehtyminen. Oppimista tuli myös paljon Excelin käyttämisestä tehokkaasti. Tallennettua dataa opinnäytetyöstä kertyi .csv formaatissa noin 550 Mt.

Fast Picker ja Heavy Picker oli aikaisemman työkokemuksen puolesta mekaanisesti tuttuja kokonaisuuksia, mutta työssä sai hyvän käsityksen myös liikkeenohjauksen ratkaisuksista Fast Picker:ssä.

LÄHTEET

Suomen Asiakastiedon www-sivut viitattu 20.11.2020 <https://www.asiakastieto.fi>

Beckhoff www-sivut. Viitattu 20.5.2020

ZenRobotics www-sivut. Viitattu 10.10.2020 <https://zenrobotics.com/>

PLCopen www-sivut. Viitattu 24.11.2020 <https://plcopen.org/downloads/introduction-structured-text-st>