



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Mikko Koivula

Rocla Oy:n vihivaunu testauksen kehittäminen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

12.11.2019

Tekijä Otsikko	Mikko Koivula Rocla Oy:n vihivaunu testauksen kehittäminen
Sivumäärä Aika	21 sivua + 1 liitettä 12.11.2019
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	sähkö- ja automaatiotekniikka
Ammatillinen pääaine	automaatiotekniikka
Ohjaajat	lehtori Timo Tuominen testausinsinööri Mika Päivärinta
<p>Työn tarkoituksena oli keksiä tapoja, joilla voidaan nopeuttaa Roclan vihivaunujen testauksen läpimenoaikoja. Työhön valittiin neljä eri testiä. Valitut testit olivat virranmittaus, nosto- ja laskusäädöt, analogia korkeuden säätö ja hydraulikka järjestelmän vuototesti. Näistä kolmeen oli tarkoitus kehitellä parannusehdotuksia ja yhdestä tehtiin toimiva demo. Demoa varten valittiin hydraulikka järjestelmän vuototesti.</p> <p>Kaikkiin kolmeen testiin kehiteltiin parannusehdotus joko testin menetelmään tai testissä käytettäviin työkaluihin. Hydraulikan vuototestiin tehtiin OpenPCS -ohjelmaa käyttäen automaattinen vaunuohjelma, joka suorittaa vuototestin itsenäisesti.</p> <p>Demoa tehdessä kohdattiin muutamia ongelmia, mutta toimiva demo saatiin silti aikaiseksi. Demon avulla voidaan hydraulikan vuototestiä nopeuttaa ja vapauttaa testaaja tekemään testin ajaksi muita tehtäviä. Demon ohjelmaa on mahdollista jatkossa kehittää, mikäli ilmenee uusia ongelmia käytössä tai vihivaunuihin lisätään öljyn lämpötilanmittaus.</p>	
Avainsanat	testaus, kehittäminen, vihivaunu, trukki, automaatio, Rocla

Author Title	Mikko Koivula Development of AGV testing
Number of Pages Date	21 pages + 1 appendix 12 November 2019
Degree	Bachelor of Engineering
Degree Programme	Electrical and automation engineering
Professional Major	Automation Technology
Instructors	Timo Tuominen, Senior Lecturer Mika Päivärinta, Test engineer
<p>The goal of the study was to look at the Rocla's AGV testing and find ways to shorten the lead time of vehicles. Four different tests were selected to this study. The tests were measurement of current consumption, lifting adjustments, analog height adjustment and hydraulic leakage test. The hydraulic leakage test was selected for the demo, and for the other three tests, suggestions for improvements were made.</p> <p>All the three selected tests got suggestion for improvement, either for method or for the tools used for the test. An automatic program was made for the hydraulic leakage test using OpenPCS software.</p> <p>While working on the demo, some problems were encountered but still, a working demo was made. With the demo, the hydraulic leakage test can be done faster and it releases the test engineer to do other things while the test is active. It is possible to develop the demo, if new problems occur during the testing or an oil temperature meter is added to the AGV.</p>	
Keywords	AGV, testing, development, forklift, automation, Rocla

Sisällys

Lyhenteet

1	Johdanto	1
2	Vihivaunu	1
3	Testit	7
3.1	Hydrauliikan vuototesti	7
3.2	Virranmittaus	8
3.3	Nosto- ja laskusäädöt	9
3.4	ATX-mallin korkeuden säätö	9
4	Virranmittauksen parannusehdotukset	10
4.1	ATX-mallin ratkaisu	10
4.2	AWT- ja ART-mallin virranmittaus	11
5	Nosto- ja laskusäädöt	12
6	”Analog height”-työkalu	14
7	Vuototesti demo	14
7.1	ATX-mallin demo	15
7.2	ART-vaunun demon erot ATX-malliin verrattuna	17
7.3	Ongelmakohdat	18
8	Yhteenveto	19
	Lähteet	21

Liitteet

Liite 1. ATX Leak -testidemo

Lyhenteet

AGV	Automated Guided Vehicle. Vihivaunu, tarkoittaa automaattitruckia.
AWT	Automated Warehouse Truck. Roclan vihivaunumalli.
ATX	Automatic Truck X. Tuotenimi Roclan pienemmälle AGV -mallille.
ART	Automated Reach Truck. Roclan uusin vihivaunu malli, joka pystyy toimimaan kapeilla käytävillä työntömaston avulla.
CVC	Compact Vehicle Controller, Kollmorgenin valmistama vihivaunuille tarkoitettu ohjainyksikkö.
PLC	Programmable Logic Controller, Ohjelmoitava logiikka.
PPA	Physical Parameter Adjust. Toiminto, joka säätää automaattisesti laserscannerin paikan ja ohjausenkooderin ohjaamaan vaunun ajamaan suoraan.

1 Johdanto

Rocla on suomalainen yritys, joka valmistaa ja kehittää manuaalitrukkeja sekä automaattitrukkijärjestelmiä. Yritys on perustettu vuonna 1942, ja se toimi silloin nimellä Rautatyö Oy. Vuonna 1979 yhtiön nimi vaihtui Rocla Oy:ksi ja ensimmäinen vihivaunujärjestelmä toimitettiin vuonna 1983. Rocla on toimittanut yli tuhat AGV järjestelmää ja yli 7000 vihivaunua. Yrityksen omistaa Mitsubishi Logisnext. [1.] Roclan konttori ja trukkien tuotanto sijaitsevat Järvenpäässä ja vihivaunujen testaus väliaikaisesti Vantaalla.

Työn tarkoituksena on keksiä uusia työkaluja testauksen ongelmakohtien nopeuttamiseksi. Tarve muutoksille johtuu siitä, että vihivaunujen kysyntä on kasvanut viime aikoina, ja jotta toimitusajat eivät venähtäisi liian pitkiksi, on tarve pyrkiä pienentämään vaunujen läpimenoaikoja. Testauksessa on muutamia tällaisia pullonkauloja, joiden testiaikoja olisi mahdollista pienentää tai testaajien työtaakkaa helpottaa.

Työssä keskitytään muutamaa eri testiin, jotka ovat hydrauliiikan vuototesti, virranmittaukset, analogia korkeuden säätö sekä nosto- ja laskusäädöt. Näistä kaikkiin on tarkoitus keksiä parannusehdotus ja hydrauliiikan vuototestistä tehdään automaattinen demo.

2 Vihivaunu

AGV tulee sanoista Automated Guided Vehicle, joka tarkoittaa suomeksi vihivaunua. Vihivaunu on automaattinen trukki, joka navigoi ja suorittaa sille annetut tehtävät ilman kuljettajaa. Vihivaunuilla voidaan lisätä sisälogistiikan tehokkuutta sekä turvallisuutta, sillä vihivaunut eivät väsy tai niillä ei ole huonoja päiviä, kuten trukin kuljettajalla saattaa olla.

Roclalla on kolmen eri tyyppin vihivaunua, jotka ovat AWT, ATX ja ART. Kaikissa näissä on modulaarinen rakenne, joissa pystytään vaihtamaan lastinkäsittely-yksikköä asiakkaan tarpeen mukaiseksi. Tyypillisimmät lastinkäsittelyt ovat perinteinen haarukka (fork), kuljetin (conveyor) ja pihti (clamp). Vaunut käyttävät navigoimiseen pääosin laserskan-

neria ja ympäristöstä tehtyä heijastinkarttaa, jota kutsutaan layoutiksi. Vaihtoehtoisia navigointi ratkaisuja ovat piste- (spot navigation), kaapeli- (wire navigation) ja ympäristönavigointi (natural navigation). [2.]

AWT-vaunu

AWT tulee sanoista Automated Warehouse Truck. AWT (kuva 1) on Roclan tyypillisin ratkaisu varastoihin, joissa nostokorkeus tai lastin paino on suurempi.



Kuva 1. AWT-vastapainovihivaunu.

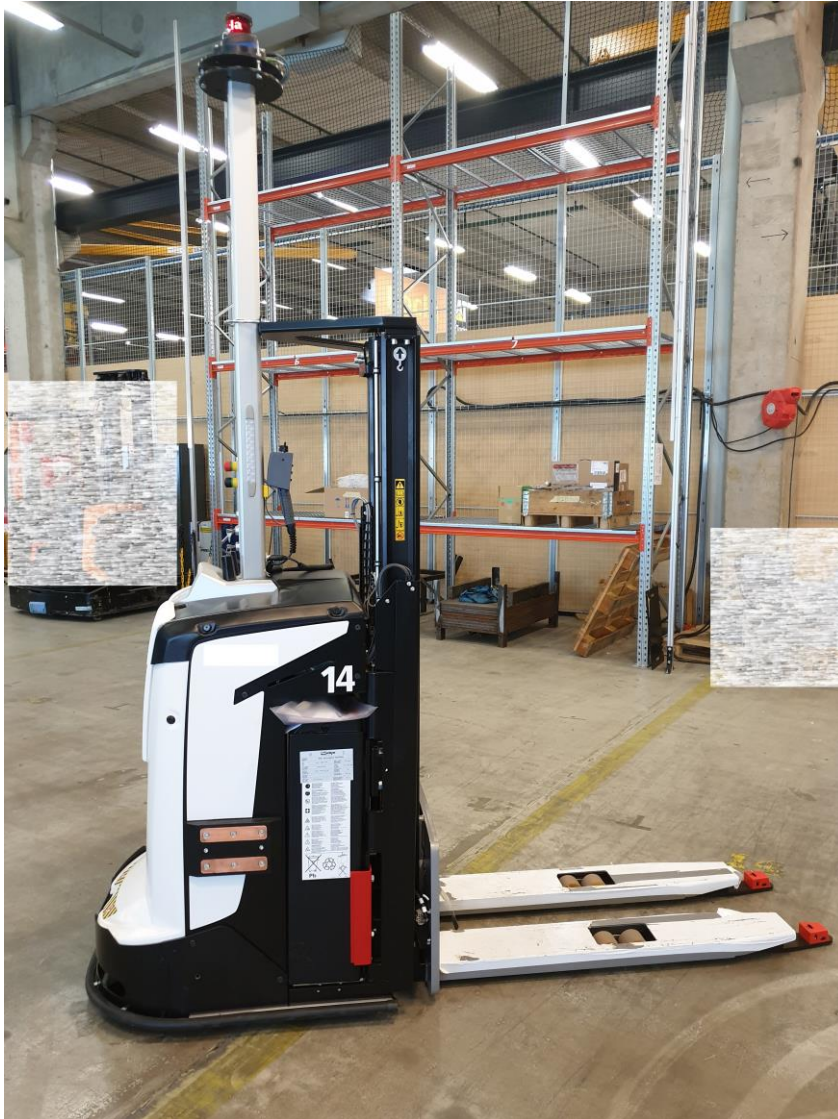
Kuten muissakin vaunutyypeissä, voidaan AWT:n vastapainomallin lastinkäsittely-yksikköä vaihtaa asiakkaan tarpeisiin sopivaksi, kuten vaihtamalla haarukoiden tilalle kuljetin-yksikkö (kuva 2) tai lisäämällä tukihaarukat vaunuun.



Kuva 2. ATX-vaunu, jossa lastinkäsittely vaihdettu kuljettimiksi

ATX-vaunu

ATX (kuva 3) on tarkoitettu pääosin matalammille korkeuksille kuin AWT. ATX on oiva ratkaisu lavojen siirtämiseen. ATX tulee sanoista Automatic Truck X. Ensimmäinen ATX tuli markkinoille 2010 [1].



Kuva 3. Perinteinen ATX-vaunu.

ART-vaunu

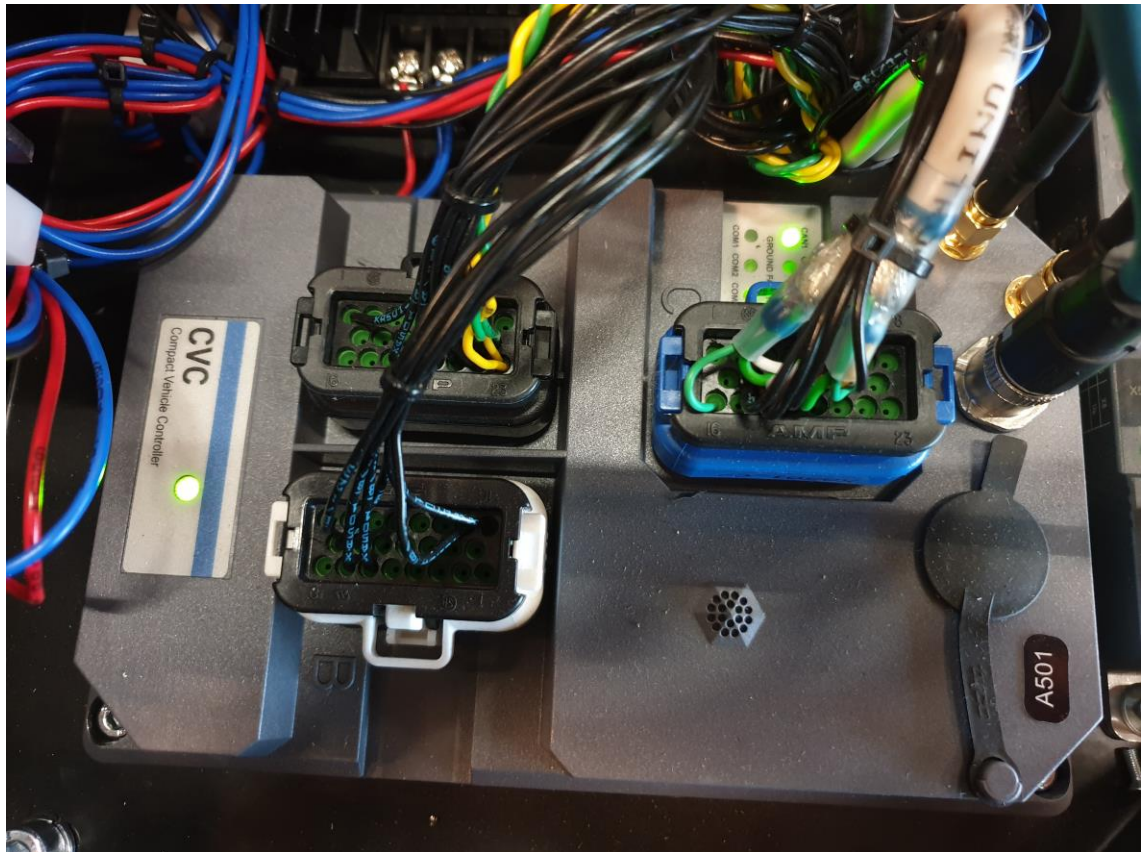
ART (kuva 4) eli Automated Reach Truck on Roclan uusiin AGV-malli. Vuonna 2019 markkinoille tullut ART eroaa AWT:sta sen työntömassan takia. Tämän ansiosta ART pystyy toimimaan jopa vain kolme metriä leveillä käytävillä ja nostamaan jopa kymmeneen metriin asti. Se pystyy käsittelemään 1600 kilogramman lastia. [4.]



Kuva 4. AGV työntömastolla eli ART.

Laitteisto

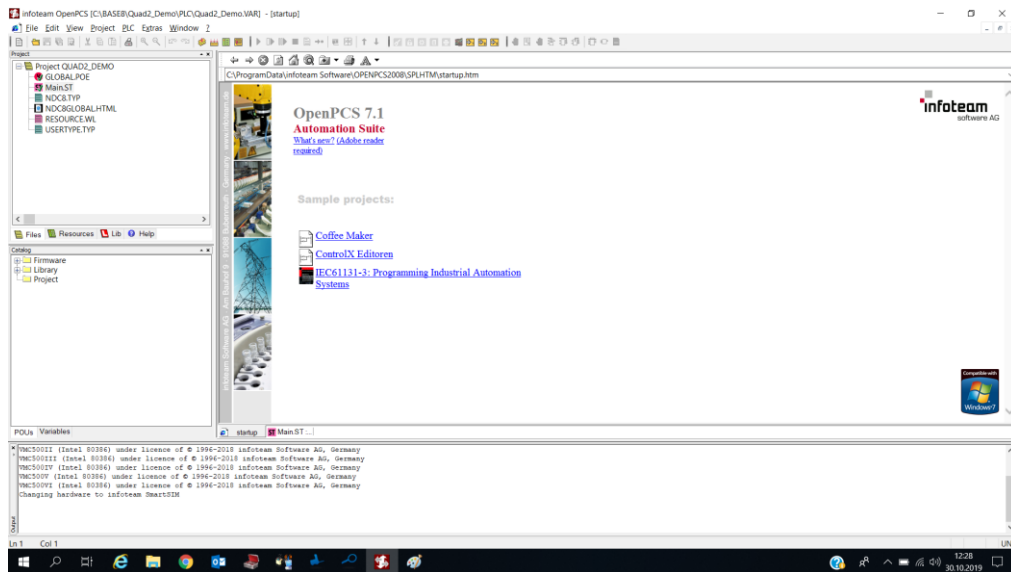
Vihivaunujen tietokoneena toimii Kollmorgenin valmistama CVC600 (kuva 5), joka on suunniteltu AGV-käyttöön. Lyhenne CVC tulee sanoista Compact Vehicle Controller. Muita vaunujen oleellisia komponentteja ovat LS2000-laserskanner, Sickin turvaskanerit, joita on vaunun mallista riippuen yksi tai useampia, enkooderit, kosketusnäyttö, käsiohjain, moottoriohjaimet, sekä Beckhoffin PLC [5.]. Vaunun laitteet ovat yhdistettynä CAN-väylään.



Kuva 5. Vaunun ohjain eli CVC.

Ohjelmisto

Vihivaunun ohjelmointi tehdään Infoteamin OpenPCS (kuva 6) ohjelmaa käyttäen. OpenPCS ohjelma tukee standardin IEC 61131-3 ohjelmointikieliä, jotka ovat ladder diagram (LD), function block diagram (FBD), structured text (ST), instructiuon list (IL) ja sequential function chart (SFC) [8]. Vihivaunuissa tyypillisin ohjelmointikieli on ST eli Structure Text. Valmis ohjelma käännetään ja siirretään vaunuohjaimelle käyttämällä Kollmorgenin Vehicle application designeria, jolla pystyy muuttamaan myös vaunun parametrejä.



Kuva 6. OpenPCS ohjelman aloitusnäyttö.

3 Testit

3.1 Hydrauliiikan vuototesti

Hydrauliiikan vuototestissä (hydraulic leakage test) on tarkoitus testata, ettei hydrauliiikkajärjestelmässä ole sisäisiä tai ulkoisia vuotoja. Testi alkaa sillä, että öljy lämmitetään liikuttamalla trukin haarukoita asiakkaan vaatiman maksimikuorman kanssa. Kun öljy on saavuttanut normaalin käyttölämpötilan, joka on yli 40 °C, haarukat jätetään yläasentoon kuorman kanssa kymmeneksi minuutiksi. Kymmenen minuutin jälkeen tarkistetaan, kuinka paljon haarukoiden korkeus on laskenut kyseisessä ajassa. Standardin EN ISO 3691-1:2015 4.6.3.1 mukaan haarukat eivät saa tulla yli 100 millimetriä alemmas lähtökorkeudesta [3]. Seuraavassa on lainaus kyseisestä standardista:

The hydraulic lifting system shall be designed such that, with the hydraulic fluid at normal operating temperature, the most substantially vertical and carrying rated capacity load, the descent of the load caused by internal leakage in the first 10 min shall not exceed

- 100 mm for trucks up to and including 10 000 kg rated capacity,
- 200 mm for trucks over 10 000 kg rated capacity

Testi vaatii yhdeltä testaajalta yllättävän pitkän ajan. Siihen nähden, kuinka yksinkertainen testi on. Tarkoituksena olisi automatisoida kyseinen prosessi, jotta se ei vaatisi henkilöä käyttämään käsiohjainta lämmittääkseen hydrauliiikan öljyn. Tärkeää olisi myös, ettei toiminto saisi vahingossakaan päälle kukaan testauksen ulkopuolinen, kuten asiakas. Tämä voidaan välttää esimerkiksi tekemällä ulkoinen laite tai sisältämällä koodiin useamman parametrin, joiden pitää olla päällä, jotta ohjelma lähtisi käyntiin. Esimerkiksi laite tulisi olla automaattilla tai parametri PPAModeEnable tulisi olla päällä samaan aikaan kun käsketään Op_Commands2-parametrillä toiminto päälle.

PPAModeEnable on parametri, jota käytetään, kun halutaan vaunulle tehdä physical parameter adjust (fyysisten parametrien säätö) eli PPA. PPA:ssa vaunu tekee sarjan liikkeitä automaattijossa ja laskee siitä laserskannerin oikean fyysisen position piirustuksiin verrattuna ja säätää ohjauksen kooderin niin, että vaunu ajaa suoraan, kun käsiohjaimesta niin käsketään. Kyseinen parametri myös ohittaa kuolleenmiehenkytkimen AWT- ja ART-vaunuista.

OP_Commands2 on niin sanottu huoltoparametri. Esimerkiksi kyseiselle parametrille arvolla yksi on määritetty lampputesti, jolla testaaja pystyy tarkistamaan vihivaunun vilkkujen toimivuuden.

3.2 Virranmittaus

Virranmittauksessa tarkistetaan, että vihivaunun virrat ovat normaalilla tasolla ja ettei saman projektin vaunujen kesken ole suuria eroja. Virtaa mitataan kuudesta eri tilanteesta, jotka ovat, kun vaunu seisoo paikallaan, hätä-seis-tilassa, ajaa eteenpäin ja taaksepäin lastin kanssa sekä lastinkäsittelyt lastin kanssa ja ilman lastia.

Näistä hankalinta on katsoa, kuinka paljon vaunu kuluttaa virtaa sen ajaessa. Tällä hetkellä virranmittaus suoritetaan laittamalla Fluken pihtimittari vaunuun kiinni ja yritetään samalla katsoa virran keskiarvoa mittarista, kun ajetaan vaunua. Tähän sopisi muutama eri ratkaisu. Toinen olisi langaton mittari, jolloin ei tarvitsisi ”juosta” vaunun perässä, kun yrittää katsoa mittarin näyttämää arvoa pieneltä Fluken näytöltä. Vaihtoehtoinen ratkaisu voisi olla se, että mittarissa olisi ominaisuus, jolla pystyisi tallentamaan tuloksia, jotta voisi tuloksia tarkastella rauhassa, vaikka jälkeinpäin kun on vaunuja ajanut. AWT- ja

ART-malleissa on myös vaunuissa itsessään virranmittausominaisuus, jota voisi näissä mittauksissa hyödyntää.

3.3 Nosto- ja laskusäädöt

ATX-mallissa on ominaisuus, joka säätää automaattisesti lastinkäsittelyn nosto- ja laskunopeuksia, kun vaunu on automaattiajossa. AWT- ja ART-malleissa nopeuksiin vaikuttavat parametrit joudutaan käsin säätämään, jotta trukin lastinkäsittelyn liikkeet olisivat sulavia. Tämä toimenpide vaatii manuaalisesti haarukoiden liikuttamista useaan kertaan ylös ja alas. Tässä säätämistä helpottaisi, jos trukki hoitaisi liikkeet automaattisesti ja testaaja voisi vain koneelta säätää parametrejä, kunnes nosto- ja laskuliike näyttää hyvältä. Tässäkin tulee huomioida, ettei tätä automaattitilaa tulisi pystyä mitenkään vahingossa kytkeä päälle.

Tämän säätämisen voisi myös yhdistää vuototestin yhteyteen AWT- ja ART-vaunuilla, jolloin kun vaunu tekee automaattisesti nosto- ja laskuliikkeitä öljyn lämmittämiseksi, voisi silloin samalla testaaja säätää nosto parametrejä.

3.4 ATX-mallin korkeuden säätö

ATX-mallin haarukoiden korkeus säädetään eri tavalla kuin isompien vaunumallien. Vaunussa on Sickin etäisyysanturi. Anturi antaa analogia-arvon, joka pitää skaalata oikein. Tämä tapahtuu niin, että aluksi säädetään korkeuden offset-arvo sellaiseen lukemaan, jotta saadaan nykyiselle haarukoiden korkeudelle realistinen tulos. Tämän jälkeen suoritetaan vähintään neljä mittausta, joista kirjataan muistiin vehicle application designerin antama arvo korkeudelle sekä lasermittarilla mitattu arvo. Arvot syötetään laskukaavaan, johon lisätään myös offset- ja scale-parametrien arvot. Kaavasta saadaan tuloksena uudet arvot parametreille. Sitten tulos vielä tarkistetaan tekemällä yksi mittaus, jolloin fyysisen mittauksen arvon tulisi olla hyvin lähellä vaunuohjelman antamaa arvoa.

Isommissa vaunumalleissa korkeuden mittaukseen käytetään absoluuttienkooderia, jonka seurauksena niiden mallien korkeuden säätö vaatii vain offset parametrin säätämisen yhdellä mittauksella. Näille malleille korkeuden säätäminen ei juurikaan vie aikaa.

Tämän takia olisi hyvä, jos ATX-malleille saisi myös kehitettyä jonkin nopeamman tavan tehdä korkeuden säätö.

4 Virranmittauksen parannusehdotukset

4.1 ATX-mallin ratkaisu

Etenkin ATX-mallin virranmittauksia voitaisiin helpottaa ostamalla uusi virtamittari. Nykyiset pihtivirtamittarit ovat jo melko vanhoja ja uudemmissa malleissa on testiä helpottavia toimenpiteitä. Esimerkiksi Flukella on myynnissä mittareita, joissa voi virtamittarin näytön irrottaa (kuva 7) ja seurata mittarin antamia lukemia etäältä tai toisessa mallissa pystyy lataamaan puhelimeen sovelluksen (kuva 8), jolla mittarin lukemia voi seurata. Puhelin sovelluksessa on myös ominaisuus, jolla voi tallentaa mittaustuloksia. Molemmat olisivat hyviä ratkaisuja nykyiseen ongelmaan, jossa on hankala seurata vaunua, kun sillä ajaa ja mittari heiluu sen kyydissä. Pihtivirtamittareihin on myös tarjolla joustava pihtejä, joilla helpottuisi virtamittarin kiinnittäminen, sillä joskus virranmittauskohta on melko ahdas perinteiselle pihtivirtamittarille.



Kuva 7. Fluke 381 irrotettavalla näytöllä ja iFlex virta anturilla. [6.]



Kuva 8. Fluke 376 virtamittari puhelin sovelluksella. [7.]

4.2 AWT- ja ART-mallin virranmittaus

Koska AWT ja ART ovat rakenteiltaan hyvin samanlaisia, on molemmissa malleissa sisäänrakennettu virtamittari. Virtamittarina käytetään LEM:in Dhab S/06 -mittaria. Kyseinen mittari on kaksikanavainen, ja sen kanavat on kytketty analogiatuloihin. Toinen kanava mittaa pienempää virta-aluetta ja toinen 20 ampeerista suurempia virtoja. Myös vaunuohjelmassa kyseiselle anturille on tehty jo valmiiksi parametrit sen käyttöä varten. Tulojen arvot skaalataan CurrentScale-, CurrentScaleFactor- ja CurrentOffset-parametreilla, jolloin Vehicle application designerillä pystyy mittarin arvon näkemään BatteryCurrent-parametrissa.

Kyseessä on siis jo toimiva tapa mitata virtaa, mutta koska kyseisiä parametrejä ei ole skaalattu oikein, ei kyseistä mittaria voi vielä käyttää. Pitäisi tutkia tarkemmin, riittäisikö parametrien säätäminen kerran vai ovatko kyseiset parametrit vaunu- tai projektikohtaisia. Mikäli arvot eivät ole vaunukohtaisia, saisi tästä tavasta näppärän työkalun AWT- ja ART-mallien virranmittaamiseen, koska virran arvon näkisi suoraan kannettavan tietokoneen näytöltä, eikä tarvitsisi erikseen kiinnittää virtamittaria vaunuun. Toisaalta näiden

vaunumallien virranmittauksen saisi myös helposti hoidettua samanlaisella virtamittarilla kuin ATX-mallien, jos parametrien säätäminen koituisi liian hankalaksi.

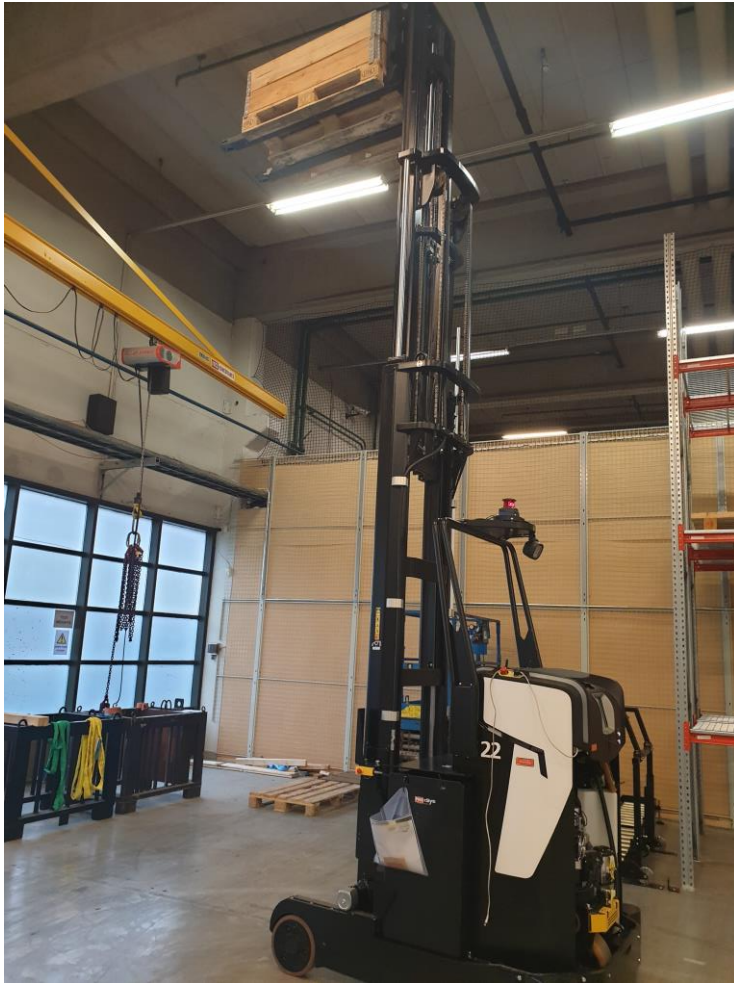
5 Nosto- ja laskusäädöt

Pääosin nosto ja laskusäädöt ovat tehtävä AWT- ja ART-malleille, koska ATX-malleissa on automaattinen säätö nosto- ja laskunopeuksille. Joskus kuitenkin saattaa olla tilanteita, jolloin automaattinen säätö ei anna optimaalista tulosta. Esimerkiksi jos projekti-kohtainen testikuorma on erittäin kevyt, saattaa näissä tilanteissa automaattinen säätö antaa tuloksen, jota pitää vielä käsin hieno säätää. Näissä tilanteissa voisi myös ATX:ssä automaattista nosto- ja laskutoimintoa hyödyntää.

Kyseisen parametrien säädön helpottamiseksi olisi hyvä tehdä ohjelma, joka hoitaisi automaattisesti vihivaunun haarukoiden liikuttamisen. Pohjana voisi käyttää samanlaista ohjelmakoodia kuin hydrauliiikan vuototestiin tehdyssä demossa, mutta pienillä muutoksilla. Parametrejä voi toki myös vuototestin yhteydessä säätää, mutta myös oma ohjelma tälle toiminnolle olisi hyvä, jotta testaaja voisi nähdä, miten vaunu laskee kuorman lattiatasolle asti.

Ohjelmassa pitäisi olla myös vuototesti demon tapaan aloitusehdot, joilla sitä ei saisi vahingossa päälle. Ohjelma voisi suorittaa tietyn määrän nosto- ja laskuliikkeitä. Testaajan työn helpottamiseksi voisi jokaisen liikkeen jälkeen olla muutaman sekunnin tauko, jolloin testaaja ehtii rauhassa miettiä ja vaihtaa vaunun parametrejä ennen kuin vaunu aloittaa uuden liikkeen. Kyseiseen ohjelmaan voisi esimerkiksi asentaa parametrin, jolla testaaja voisi itse määrittää toistojen lukumäärän ja tarvittaessa helposti käynnistää ohjelman uudelleen.

Duplex mast eli teleskooppimasto (kuva 9) on yleinen AWT- ja ART-vaunuilla, ja se mahdollistaa kuorman nostamisen korkeammalle kuin tavallisella mastolla lisäämättä vaunun fyysistä korkeutta. Vihivaunussa kyseiselle toiminnolle on parametri, joka kertoo vaunulle korkeuden, jolla masto vaihtuu teleskooppi osaan. Parametreillä pystytään myös rajoittamaan nosto- ja laskunopeutta maston vaihtokohdassa sekä tarvittaessa säätää, kuinka aikaisin vauhti alkaa hidastumaan.



Kuva 9. ART-vaunu duplex-mastolla.

Jos vaunussa on teleskooppimasto, tulisi testaajan varmistaa ennen automaattisen toiminnon käyttöä, että maston vaihtokohdan parametrit ovat vaunuun laitettu oikein ja nopeudet eivät ole liian suuret. Tällöin vältetään mahdollisilta vaurioilta, joita saattaisi syntyä, jos vaunu liikuttaisi mastoa liian nopeasti vaihtokohdassa.

Mikäli ohjelmassa käytettäisiin nosto- ja laskukorkeuksien määrittämiseen Min- ja Max-Height arvoja, tulisi nekin olla määritelty oikeiksi ja tarkistaa, että vaunu pystyy ne saavuttamaan. Vaihtoehtoisesti ohjelma voisi alussa kysyä käyttäjältä, että minkä korkeuksien välillä haarukat liikkuvat.

6 ”Analog height”-työkalu

Tällä hetkellä testaajan täytyy suorittaa vähintään neljä mittausta haarukoiden eri korkeuksilta, kirjata fyysinen korkeus ja vaunuohjelman antama korkeus ylös laskukaavaan. Kätevä ratkaisu olisi, jos voitaisiin vähentää testaajan liikkumista tietokoneen ja vaunun välillä. Vaunu voisi esimerkiksi suorittaa ohjelman, jossa se käy neljän eri korkeuden välillä, jotka ovat määritelty ennalta. Kyseiset korkeudet voisi vielä tallentua parametreihin, jotta ne voisi jälkeempään tarkistaa ja syöttää sieltä laskukaavaan. Tämä jättäisi testaajalle vain fyysisten korkeuksien mittaamisen, eikä tarvitsisi käydä koneella välissä kirjaamassa molempia tuloksia. Näin välttyisi myös manuaaliselta haarukoiden liikuttamiselta. Uudemmissa laseretäisyysmittareissa on vielä muisti- tai tallennustoimintoja, jolloin ei testaajan tarvitse edes erikseen mittaustuloksia ulkoa muistaa.

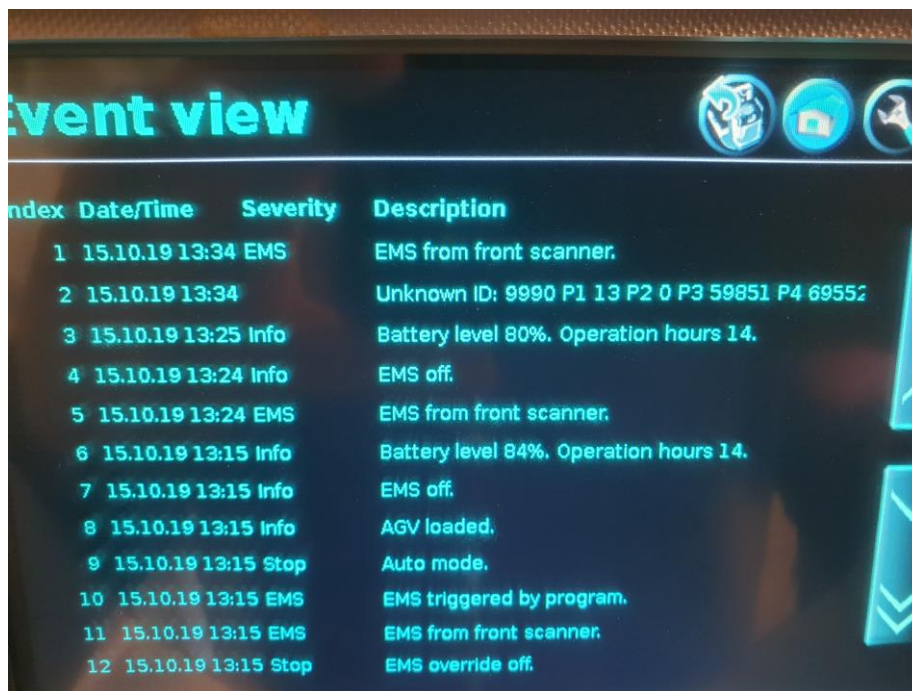
Riskinä kyseisessä tavassa olisi se, että testaajalla tulisi olla turvallista suorittaa mittaukset ja vaunu ei saisi kesken mittauksen alkaa liikuttaa haarukoita. Liikkeiden välissä tulisi olla reilusti aikaa, joka voitaisiin esimerkiksi ajastimella asettaa, ja vaunu voisi antaa vaikka äänimerkin vielä ennen liikkeen aloittamista. Toinen mahdollisuus olisi, että ennen jokaista liikettä vaunu vaatisi ”reset” -painikkeen painamista ennen kuin siirtyisi seuraavalle korkeudelle.

7 Vuototesti demo

Vuototestin ohjelmassa käytettiin ohjelmointikielenä structured textiä. Vuototestin demo on aliohjelma, jota kutsutaan pääohjelmassa, kun sille asetetut ehdot täyttyvät. Ohjelman rakenteessa päädyttiin state-rakenteeseen, joka on tyypillinen vihivaunujen ohjelmoinnissa. State-rakenteessa ohjelma suorittaa sille määriteltyjä asioita niin kauan kunnes sen ehdot siirtyä seuraavan tilaan täyttyvät. State-rakenteessa voidaan myös palata aiempiin tiloihin ja siten tehdä toistotoiminto eli niin sanottu loop (silmukka). Aliohjelman alussa on kommentteilla kerrottu, mitä kyseinen ohjelmalohko tekee ja kuinka se voidaan aloittaa, sekä lisätty englanniksi kommentteja selkeyttämään parametrejä ja jokaisen eri Staten toimintoa.

7.1 ATX-mallin demo

Ohjelma liikuttaa vaunun haarukoita kuorman kanssa automaattisesti minimi ja maksimi arvojen välillä. Hyödyntäen loop toimintoa, ohjelma toistaa tilaa 2–4 kunnes ehto siirtyä tilasta 4 tilaan 5 täyttyy, joka tässä tapauksessa on ajastin. Tämän jälkeen haarukat jäävät yläasentoon kymmeneksi minuutiksi ja ohjelma ottaa haarukoiden senhetkisen korkeuden muistiin parametriin StopHeight. Ajastimen loputtua ohjelma lähettää eventin (kuva 10), jossa on testin tulos. Event ID on tällä hetkellä 9990, ja P1 kertoo testin tuloksen millimetreissä. P1 on parametrin Result-arvo, joka on kymmenen minuutin jälkeinen, haarukoinen korkeus vähennetty arvosta StopHeight, joka otettiin aiemmin talteen. Valmiin testin merkiksi palavat keltaiset merkkivalot. Sitten ohjelma jää odottamaan testaa- jaa joko sammuttamaan vaunun tai Reset-toiminnolla palaamaan testin alkutilanteeseen.



Index	Date/Time	Severity	Description
1	15.10.19 13:34	EMS	EMS from front scanner.
2	15.10.19 13:34		Unknown ID: 9990 P1 13 P2 0 P3 59851 P4 69552
3	15.10.19 13:25	Info	Battery level 80%. Operation hours 14.
4	15.10.19 13:24	Info	EMS off.
5	15.10.19 13:24	EMS	EMS from front scanner.
6	15.10.19 13:15	Info	Battery level 84%. Operation hours 14.
7	15.10.19 13:15	Info	EMS off.
8	15.10.19 13:15	Info	AGV loaded.
9	15.10.19 13:15	Stop	Auto mode.
10	15.10.19 13:15	EMS	EMS triggered by program.
11	15.10.19 13:15	EMS	EMS from front scanner.
12	15.10.19 13:15	Stop	EMS override off.

Kuva 10. Vaunun event log kun testi on valmis. Testin tulos P1 = 13.

Öljyn lämmittäminen

Testien perusteella keskimäärin öljyn lämmittäminen normaalilla ATX-vaunulla ja normaali painoisella kuormalla meni noin 6,5 minuuttia. Tämän perusteella oletusajaksi on

asetettu kahdeksan minuuttia, jotta öljy lämpenee varmasti haluttuun lämpötilaan. Testaajan kannattaa kuitenkin lisätä uusien vaunujen kohdalla lämpömittari öljyyn ja varmistaa riittääkö kyseinen aika öljyn lämmittämiseksi. Testin pituutta voidaan lisätä muuttamalla Test_TIME-parametria.

Käynnistysehdot

Käynnistysehtoina ovat parametri boolean_10 on arvolla 1 ja MinHeight-parametrin pitää olla arvolla 300. Boolean_10 parametrin ollessa arvolla 1, se vaimentaa vaunun äänimerkin. Tämä tekee testaajien työstä mukavampaa, kun ei vaunut jatkuvasti pidä ääntä, kuten ne normaalisti automaattilla ajaessa tekevät. MinHeight-parametri asetetaan arvoon 300, jotta ei kuorma kolahda lattiaan asti vaunun laskiessa kuormaa alaspäin. Koska boolean_10 on asiakkaiden tiloissa oltava arvossa 0 ja MinHeight-parametri on yleensä alle 100, on tällöin hyvin epätodennäköistä, että kyseistä toimintoa saisi kukaan vahingossa päälle. Vaunu tulee olla myös automaattitilassa kyseisen testin ajan, koska tällöin vaunun AutomaticHeightControl parametri mahdollistaa haarukoiden liikuttamisen CommandHeight parametrilla. Vaunun ollessa automaattitilassa ei tarvitse myöskään huolehtia AWT- ja ART-mallien käsiohjaimen kuolleen miehen kytkimestä.

Demon turvallisuus

Testin ollessa käynnissä pätee kaikki normaalit automaattiajan turvallisuus toiminnotkin. Jos joku kävelee kesken testin vaunun Sickin turva alueelle, pysähtyvät vaunun haarukat, kunnes alue vapautuu. Myös vaunun käyttäjä-seis- ja hätä-seis-painikkeet toimivat normaaliin tapaan. Jos testikuorma sattuu kesken testin luisumaan ja lastinkäsittely-yksikön kuorman tunnistuslappä sattuu vapautumaan, aiheuttaa tämä myös testin pysähtymisen ja vaunun keltaiset valot alkavat vilkkumaan.

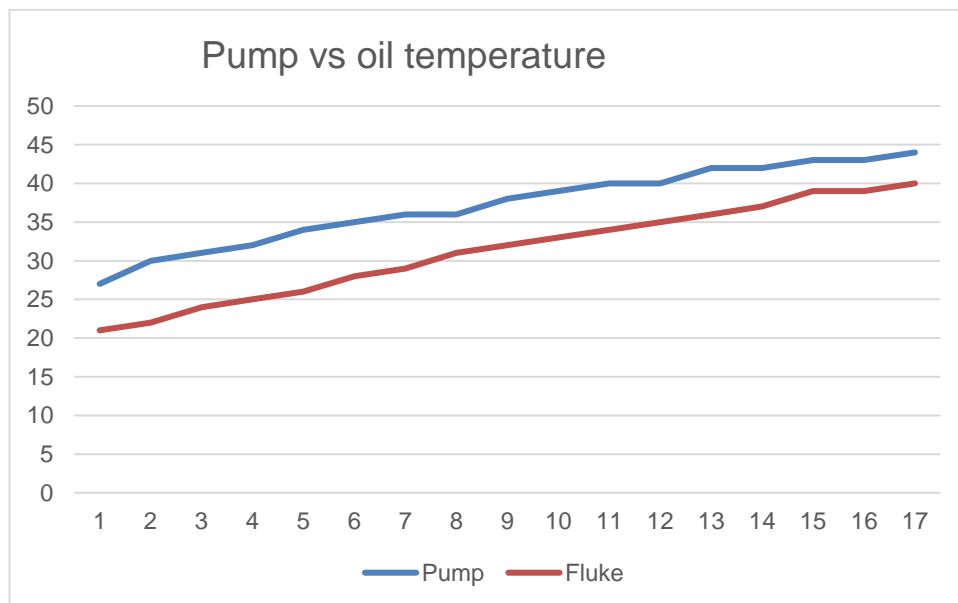
Hydrauliikan vuototestiä tehdessä on haarukoiden alle pääsy estettävä ja alue merkittävä esimerkiksi ketjulla tai muulla huomiota herättävällä tavalla, erityisesti silloin, jos jättää vaunun suorittamaan testiä ilman valvontaa.

Vaunun liikuttaessa haarukoita lähtee käyntiin ajastin, jonka aikana haarukoiden tulee päästä minimi- tai maksimikorkeuksiin. Jos ajastin saavuttaa sille määritetyn ajan ennen

kuin haarukat ovat päässeet ääriasentoon, siirtyy vaunu "Error/Stop test" -tilaan. Tällä pyritään välttämään mahdollisista hydraulikkavuodoista aiheutuvat vahingot.

7.2 ART-vaunun demon erot ATX-malliin verrattuna

ART-mallin ohjelma on samasta pohjasta tehty kuin ATX-mallin ohjelma. Suurimpana erona on se, että koodissa on pystytty hyödyntämään vuototestissä oleellista asiaa, eli lämpötilaa. AWT- ja ART-vaunuissa on mahdollista seurata pumpun lämpötilaa. Yhdeksi ehdoksi lastinkäsittelyn lopettamiseksi lisättiin tähän versioon pumpun lämpötila, joka on verrannollinen öljyn lämpötilaan. Työn edetessä huomattiin, että aina kun haarukat liikkuvat ylös, lähti pumpun tuuletin myös päälle. Tämä viilensi pumppua, jolloin pumpun alkuperäinen lämpötila kriteeri testin lopettamiseksi, joka oli 44 celsiusastetta (kuva 11), osoittautui liian korkeaksi öljyn kannalta. Pumpun lämpötila ehdoksi riitti 41 celsiusastetta, jolloin öljy lämpeni yli testiin vaadittavan minimi lämpötilan.



Kuva 11. Pumpun ja öljyn lämpötila vertailu. Pumpun lämpötila tarkistettu, kun haarukat olleet hetken paikallaan.

ART- ja AWT-malleissa on keltaisten vilkkujen lisäksi myös punaiset vilkut. Tätä ominaisuutta pystytään hyödyntämään merkiksi siitä, että ohjelma on suoritettu loppuun. Ohjelman ollessa valmis, vaunun punaiset merkkivalot syttyvät päälle ja pysyvät päällä, kunnes vaunusta otetaan virrat pois tai käytetään parametria `Reset_B`, joka palauttaa vaunun takaisin tilaan nolla.

7.3 Ongelmakohdat

Työn edetessä huomattiin, että alun perin suunniteltuja käynnistysehtoja ei voitu käyttää, koska esimerkiksi `PPAModeEnable`-parametrinä ilmeni ominaisuus, joka estää lastinkäsitteily-yksikön käyttämisen kyseisen parametrin ollessa päällä. Myös `OP_Commands2` parametri keskeytti testin jostain syystä.

Automaattiajon ansiosta ohjelmaan sai paljon hyviä ominaisuuksia, mutta myös ilmeni yksi uusi ongelma. `AutomaticHeightControl` tarkkailee haarukoiden korkeutta automaattiajossa. Muutamissa tilanteissa, joissa haarukoiden korkeus putosi vuototestin aikana yli kymmenen millimetriä, saattoi vaunu korjata korkeutta aloituslukemiin. Tämä ongelma saatiin korjattua lisäämällä komento, jossa uudeksi `CommandHeightiksi` käsketään nykyistä korkeutta. Tällöin vaunu ei tee korjausliikkeitä, jos haarukoiden korkeus putosi.

Koska pohjana käytettiin jo valmista ja toimivaa ohjelmaa, ilmeni myös ongelmia saada vaunun merkkivalot toimimaan. Tämä voi johtua siitä, että ohjelmassa komentojen päälle kirjoitetaan myöhemmin ohjelmassa, jolloin valot eivät pysy päällä halutulla tavalla, tai niille on asetettu ohjelmassa ehto, joka pitäisi olla myös aktiivinen samaan aikaan.

Myös `ControlUT`-parametri tulee olla pakotettu arvoon 1, jotta demo toimii. Ellei kyseinen parametri ole arvossa 1 testin aikana, alkaa vaunu laskea haarukoita `DefaultHeight` korkeuteen liikkeiden loputtua ja jättää haarukat siihen korkeuteen tarkoitetun yläasennon sijaan. `DefaultHeight` on ohjelmassa määriteltä haarukoiden korkeus, jota vaunu käyttää automaattiajossa.

Työn edetessä tehtiin päätös, että päällekkäisyyksien ja ehtojen etsiminen ei kuulu työhön, jotta siitä ei tulisi liian haastava. Demon lisääminen base-vaunuohjelmiin jää tuotekehityksen vastuulle.

8 Yhteenveto

Työn tarkoituksena oli keksiä parannusehdotuksia testien nopeuttamiseksi ja testauksen läpimenoaikojen pienentämiseksi. Insinööriyön keskeisin tavoite oli saada tehtyä automaattinen demo hydrauliiikan vuototestistä, jota voitaisiin testauksessa hyödyntää. Työn tavoitteet saavutettiin ja saatiin uusia työkaluja testauksen nopeuttamiseksi.

Työn aikana selvisi myös uusia ominaisuuksia muutamista parametreista. Esimerkiksi PPAModeEnable-parametrin vaikutus lastinkäsittely-yksikköön tuli osalle yllätyksenä ja selvittämättä jäi syy, miksi OP_Commands2-parametri pysäytti testin. AWT- ja ART-mallien sisäänrakennettu virranmittaus on ollut myös käyttämättömänä, mutta asiaa lisää tutkimalla, siitä voi saada hyvän työkalun testauksen avuksi.

Erityisesti haasteita demon ohjelmoinnin suhteen aiheutti jo valmis vaunuohjelma ja parametrien suuri määrä. Työn alussa oli haastavaa löytää oikeat parametrit haluttujen toimintojen toteuttamiseksi. Haasteista huolimatta toimiva vuototesti demo saatiin aikaiseksi.

Tarkoituksena on, että vuototestidemo lisätään Roclan base-vaunuohjelmiin, jolloin aina kun aloitetaan tekemään uutta projektikohtaista ohjelmaa, on kyseinen demo jo ohjelmassa sisällä ja testauksen käytettävissä. Tuotekehityksen tehtäväksi jää demon integrointi base-ohjelmiin ja selvitettäväksi, missä kohtaa ohjelmassa merkkivalojen ja ControlUT-parametrien päälle kirjoitetaan. Myös Vehicle application designerin puolelle oleellisten parametrien lisääminen helpottaisi testaajien taakkaa ja poistaisi OpenPCS:n tarpeen testiin liittyen.

Vuototestin ohjelmaa pystyttäisiin parantamaan ja tekemään varmemmaksi lisäämällä vaunuihin lämpömittarit öljyn lämpötilan mittaamiseksi. Vaunuihin voitaisiin lisätä joko ulkopuolinen mittari, joka antaisi tulon vaunuun input-signaalina tai lisäämällä vaunun luonnostaan olemassa olevalle Beckhoffin logiikalle lämpötilanmittaus input-kortti [9]. Jos kyseisiä muutoksia vaunuihin tehdään tai testauksessa ilmenee ohjelmasta uusia vikoja, on mahdollista tehdä ohjelmaan tarvittavat muutokset.

Osa parannusehdotuksista vaatii enemmän työtä joko ohjelmistopuolelta tai sitten muutoksia vaunuihin. Lisäämällä jokaiseen vaunuun lämpötilamittari lisää jonkin verran kustannuksia ja saattaa vaatia myös suunnittelutyötä mekaniikkasuunnittelijoilta. Mutta jos hankkimalla uudet työkalut virranmittauksen nopeuttamiseksi ja ottamalla käyttöön vuototesti demo voidaan testaajien urakkaa helpottaa huomattavasti.

Lähteet

- 1 Tietoa Roclasta. 2019. Verkkoaineisto. Rocla Oy. <<http://www.rocla.com/fi/tietoa-roclasta>> Luettu 4.9.2019.
- 2 Maximizing your logistic performance. 2019. Verkkoaineisto. Rocla Oy. <http://www.rocla.com/sites/rocla.com/files/page/fields/field_sample_files/rocla-agvgenbr_eng_web.pdf> Luettu 4.9.2019.
- 3 Industrial trucks - Safety requirements and verification. 2016. Standardi. EN ISO 3691-1:2015 4.6.3.1. Luettu 4.9.2019
- 4 Reach Mast AGV. 2019. Verkkoaineisto. Rocla AGV. <<https://www.rocla-agv.com/en/products/reach-mast-agv>> Luettu 4.9.2019.
- 5 Building AGVs. 2019. Verkkoaineisto. Kollmorgen. <<https://www.kollmorgen.com/en-us/solutions/automated-guided-vehicles/building-agvs/>> Luettu 9.9.2019
- 6 Fluke virtamittari puhelin sovelluksella. 2019. Verkkoaineisto. Fluke Store. <<https://www.fluke.com/en/product/electrical-testing/clamp-meters/fluke-376-fc>> Luettu 22.10.2019
- 7 Fluke virtamittari irtonäytöllä. 2019. Verkkoaineisto. Fluke Store. <<https://www.fluke.com/en/product/electrical-testing/clamp-meters/fluke-381>> Luettu 22.10.2019
- 8 PLC programming system. 2019. Verkkoaineisto. Infoteam Software AG <<https://infoteam.de/en/standards-compliant-software/plc-programming-systems/>> Luettu 25.10.2019
- 9 Beckhoff järjestelmä. 2019. Verkkoaineisto. Beckhoff Automation Ltd. <<https://www.beckhoff.fi/fi/default.htm?applicat/jarjestelma.html>> luettu 29.10.2019

ATX Leak -testidemo

```

VAR
    State_INT                : INT;
    ForkHeight_INT           : INT;      (*Current mast height*)
    MaxHeight_INT            : INT;      (*Max lift height*)
    MinHeight_INT            : INT;      (*Min lift height*)
    StartLeakTest_B         : Bool;     (*Test starts*)
    WaitTime_B               : Bool;     (*Ten minute wait time starts*)
    TON_leak                 : TON;
    TON_leak_time_out1      : TON;
    TON_leak_time_out2      : TON;
    TON_ready                : TON;
    Test_TIME                : TIME := T#8m; (*Parameter to adjust the test
time*)
    TON_leak_time_out1_IN    : Bool;
    TON_leak_time_out2_IN    : Bool;
    StopHeight_INT          : INT;      (*Height before ten minute wait
time*)
    Result_INT              : INT;      (*Test result*)
    Reset_B                 : Bool;     (*Parameter to reset the test*)
    StopTest_B              : Bool;     (*Parameter to stop test early*)
    TestReady_B             : Bool;
    TON_WaitForStop         : TON;
    WaitForStop_IN          : Bool;
    Pl_DINT                 : DINT;
    SendEvent_FB            : Event_ToBuffers_FB_S;

```

```
END_VAR
```

```

MaxHeight_INT := DINT_TO_INT(NDC8.UserDefined.MaxHeight);
MinHeight_INT := DINT_TO_INT(NDC8.UserDefined.MinHeight);
ForkHeight_INT := NDC8.BC_Tractor_1_21.Term3_AnalogInput1_Height_INT;

```

```

If NDC8.UserDefined.Boolean_10 = true and MinHeight_INT = 300 then
Case State_INT of

```

```
0: (*Start*)
```

```

    StartLeakTest_B := True;
    WaitTime_B := false;
    Reset_B := false;
    StopTest_B := false;
    TestReady_B := false;
    Mi.Mast.Command.Lh.AutomaticHeightControlActive_B := true;
    Features.Lift.Command.Loadhandling.Control_Ut := Automatic;

```

```
if
```

```

    Features.Lift.Command.Loadhandling.Control_Ut = Automatic then
    State_INT := 2;

```

```
end_if;
```

```
2: (*Mast up movement*)
```

```

    TON_Leak(IN := StartLeakTest_B,PT := Test_Time);

```

```
If
```

```

    ForkHeight_INT > 200 then Mi.Mast.Command.Lh.Height_INT := MaxHeight_INT
    - 5;

```

```
end_if;
```

```

    TON_leak_time_out1(IN := TON_leak_time_out1_IN,PT := T#30s);
    TON_leak_time_out1_IN := true;

```

```

If
    ForkHeight_INT > MaxHeight_INT - 25 then
        TON_leak_time_out1_IN := false;
        State_INT := 3;
elsif
    TON_leak_time_out1.Q=true then TON_leak_time_out1_IN := false;
    StartLeakTest_B := false;
    State_INT := 11;
end_if;

3: (*Mast is up. If test is ready then state 5, otherwise test continues by going to state 4*)

    TestReady_B := TON_Leak.Q;
If
    ForkHeight_INT > MaxHeight_INT - 25 and TestReady_B then
        State_INT := 5;
elsif
    ForkHeight_INT > MaxHeight_INT - 25 and StopTest_B then
        State_INT := 5;
Else
    State_INT := 4;
end_if;

4: (*Mast down movement and then state 2 to start up movement again*)
    Mi.Mast.Command.Lh.Height_INT := MinHeight_INT + 5;
    TON_Leak(IN := StartLeakTest_B,PT := Test_Time);
    TON_leak_time_out2(IN := TON_leak_time_out2_IN,PT := T#30s);
    TON_leak_time_out2_IN := True;
if
    TON_leak_time_out2.Q then TON_leak_time_out2_IN := false;
    StartLeakTest_B := false;
    State_INT :=11;
Elsif
    ForkHeight_INT<MinHeight_INT+20 then TON_leak_time_out2_IN := false;
    State_INT := 2;
end_if;

5: (*Wait time for mast movement to stop*)

    TON_WaitForStop(IN :=WaitForStop_IN,PT := T#3s);
    WaitForStop_IN := true;

if
    TON_WaitForStop.Q then
        Mi.Mast.Command.Lh.MoveStop_B := true;
        StopHeight_INT := ForkHeight_INT;
        State_INT := 6;
end_if;

6: (*Ten minute wait time*)

    WaitForStop_IN := false;
    WaitTime_B := true;
    StartLeakTest_B := false;
    Mi.Mast.Command.Lh.Height_INT := ForkHeight_INT;
    TON_ready(IN := WaitTime_B,PT := T#10m);
    TON_Leak(IN := StartLeakTest_B,PT := Test_Time);
    TON_WaitForStop(IN :=WaitForStop_IN,PT := T#3s);
If (*Reset*)
    Reset_B Then

```

```
        WaitTime_B := false;
        StartLeakTest_B := false;
        WaitForStop_IN := false;
        State_INT := 0;
    end_if;
    Result_INT := StopHeight_INT-ForkHeight_INT;
if
    TON_ready.Q then
        State_INT := 7;
end_if;

7: (*Sending event*)

    P1_DINT := INT_TO_DINT(Result_INT);
    SendEvent_FB(EventCode_DINT := 9990,Parameter1_DINT := P1_DINT);(*Result
of leak test*)
    State_INT := 8;

8: (*Reset/Wait for user*)

    TON_ready(IN := WaitTime_B,PT := T#10m);
    TON_Leak(IN := StartLeakTest_B,PT := Test_Time);
    TON_WaitForStop(IN := WaitForStop_IN,PT := T#3s);

    Tractor.IO.Output.Light.BL_Yellow_B := true;
    Tractor.IO.Output.Light.BR_Yellow_B := true;
If
    Reset_B Then
        WaitTime_B := false;
        StartLeakTest_B := false;
        WaitForStop_IN := false;
        Tractor.IO.Output.Light.BL_Yellow_B := false;
        Tractor.IO.Output.Light.BR_Yellow_B := false;
        State_INT := 0;
end_if;

11: (*Stop test/error*)

    Mi.Mast.Command.Lh.MoveStop_B := true;
    TON_ready(IN := WaitTime_B,PT := T#10m);
    TON_Leak(IN := StartLeakTest_B,PT := Test_Time);
    TON_WaitForStop(IN := WaitForStop_IN,PT := T#3s);
If
    Reset_B Then
        WaitTime_B := false;
        StartLeakTest_B := false;
        WaitForStop_IN := false;
        State_INT := 0;
end_if;
end_case;
end_if;
```