



Tuukka Pennanen

SÄHKÖTYÖKONEEN MOOTTORINOHJAUSJÄRJESTELMÄN OHJELMOIN- TI

SÄHKÖTYÖKONEEN MOOTTORINOHJAUSJÄRJESTELMÄN OHJELMOIN- TI

Tuukka Pennanen
Opinnäytetyö
Kevät 2013
Automaatiotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Automaatiotekniikan koulutusohjelma

Tekijä: Tuukka Pennanen

Opinnäytetyön nimi: Sähkötyökoneen moottorinohjausjärjestelmän ohjelmointi

Työn ohjaaja: Timo Heikkinen

Työn valmistumislukukausi ja -vuosi: Kevät 2013

Sivumäärä: 51 + 1 liitettä

Tämän työn aiheena oli suunnitella ja toteuttaa saadun vaatimusmäärittelyn pohjalta logiikan ohjelmointi Hilux-sähkötyökoneeseen. Sähkötyökone on Oulun seudun ammattikorkeakoulun, Oulun teknisen liikelaitoksen ja Randax Oy:n yhteistyöprojekti, jonka tarkoituksena oli toteuttaa työkoneeseen sähköinen voimansiirto. Randax Oy on valmistanut työkoneessa käytetyn sähkömoottorin.

Työssä perehdytään logiikan ohjelmointiin yleisellä tasolla sekä selvitetään sähkötyökoneessa käytetyn CANopen-väyläprotokollan toimintaa ja ominaisuuksia. Lisäksi pohditaan ohjelmoinnin kannalta huomioitavia asioita ajoneuvokäytössä ja kerrotaan kyseisen sovelluksen ohjelmoinnista, PI-säätimen virityksestä ja ohjelmoidun sovelluksen toiminnasta. Logiikkana käytetään ABB:n AC500-sarjaa ja ohjelmointi tehdään CoDeSys-ympäristössä. Lopuksi pohditaan opittuja asioita ja mahdollisia sovelluksen jatkokehitysideoita.

Työn lopputuloksena saatiin toimiva sovellus sähkötyökoneen ohjausjärjestelmään, joten työtä voidaan pitää kokonaisuutena hyvin onnistuneena. Sovellukseen ohjelmoitiin jopa sellaisia ominaisuuksia, joita ei alkuperäisessä vaatimusmäärittelyssä ollut, vaan ne tulivat ideoina esille työtä tehtäessä. Sähkötyökone jatkaa ammattikorkeakoululla ja teknisellä liikelaitoksella testikäytössä, joten seuraavien opiskelijoiden on helppo tehdä mahdollisia parannuksia ja muutoksia sovellukseen.

Asiasanat: ohjelmoitavat logiikat, sähköautot, käyttöliittymät, ohjelmointi

ABSTRACT

Oulu University of Applied Sciences
Automation Engineering Degree Programme

Author: Tuukka Pennanen

Title of thesis: Programming of Engine Control System of Electric Vehicle

Supervisor: Timo Heikkinen

Term and year when the thesis was submitted: Spring 2013

Pages: 51 + 1 appendices

The aim of this thesis was to program the engine control system for electric maintenance vehicle. Electric Maintenance Vehicle project was carried out in co-operation with students from Oulu UAS, Oulu technical public utility company (TEKLI) and Randax Ltd. Randax Ltd. is an electrical technology company, which have made the engine of the electric maintenance vehicle.

In the thesis it is examined how PLC can be programmed and what kind of ways there is to do it. The technology behind the CANopen communication protocol is explained. Also there is thought what kind of special things has to take account of when programming PLC of the electric vehicle. Since there are different kind of controllers which are done with PI-controller, the basic principles of the PI-controller and tuning of the controller are also covered. The AC500-series, made by ABB, was used as a PLC in the project and the programming of the PLC was carried out with the CoDeSys programming system, made by 3S - Smart Software Solutions. Finally, some thought is given as what could have been done differently and what were the challenges of the thesis.

As a result of the thesis a working PLC application was commissioned, so the thesis can be considered as success. Even a few new features that were not in the original requirements were innovated and implemented during the project. The electric maintenance vehicle will continue as test platform for Oulu UAS and TEKLI, so the future students can improve the PLC application.

Keywords: electric car, electric vehicle, logic, programming, user interfaces

ALKULAUSE

Tämä työ on tehty sähkökonevalmistaja Randax Oy:lle, jossa yhteyshenkilönä on toiminut toimitusjohtaja Timo Schäfer. Työn ohjaajavana opettajana on toiminut lehtori Timo Heikkinen Oulun seudun ammattikorkeakoulusta.

Kiitokseni haluan osoittaa ohjaavalle opettajalle, työn tilaajalle, ABB:lle teknisestä tuesta sekä kaikille projektissa mukana olleille yhteistyökumppaneille ja muille opiskelijoille. Yhteishenki opiskelijoiden kesken on ollut hyvää ja projektia eteenpäin vievää.

Lisäksi haluan antaa kiitokset vaimolleni, joka on antanut tukea ja pyörittänyt arkea, vaikka allekirjoittaneen päivät ovat välillä venyneet pitkiksi opinnäytetyötä tehdessä.

Oulussa 10.5.2013

Tuukka Pennanen

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
SANASTO	8
1 JOHDANTO	9
2 SÄHKÖAJONEUVOT	10
3 LOGIIKKAOHJAUS	11
3.1 Logiikka ohjainlaitteena	11
3.2 Logiikan rakenne	11
3.3 Logiikan ohjelmointi	13
3.4 Ohjelmointikielet	14
3.4.1 Relekaavio (Ladder Diagram, LD)	14
3.4.2 Logiikkakaavio (Function Block Diagram, FBD)	14
3.4.3 Käskylistä (Instruction List, IL)	15
3.4.4 Strukturoitu teksti (Structured text, ST)	15
3.4.5 Sekvenssiohjelmointi (Sequential Function Chart, SFC)	16
3.4.6 Jatkuvatoinen ohjelmointi (Continuos Function Chart, CFC)	16
3.5 PID-säädin	17
3.5.1 P-säätö	17
3.5.2 I-säätö	18
3.5.3 D-säätö	18
3.5.4 Vuritys	19
4 CANOPEN-PROTOKOLLA	21
4.1 Toiminta	21
4.2 Protokollat	23
4.3 Solmun verkonhallintatilat	24
4.4 Viestikehyksen rakenne	25
5 HILUX-SÄHKÖTYÖKONEEN OHJAUSJÄRJESTELMÄ	26
5.1 Moottori	26
5.2 Taajuusmuuttaja	26

5.3 Logiikka	27
5.3.1 Prosessori	27
5.3.2 Kenttäväylä- ja I/O-moduulit	28
5.4 Akusto ja akkujenhallinta	29
5.4.1 Akusto	29
5.4.2 Akkujenhallintajärjestelmä (BMS)	30
5.5 Ohjauspaneeli	30
5.6 Tiedonkeruu	31
6 SÄHKÖTYÖKONEEN LOGIIKAN OHJELMOINTI	32
6.1 Ohjelmointityökalut	32
6.2 Laitteiston määrittäminen ja konfigurointi	32
6.3 Taajuusmuuttajan parametrisointi	35
6.4 Akkujenhallintajärjestelmän parametrisointi	36
6.5 Ohjelmointi	36
6.5.1 PI-säätimen viritys	36
6.5.2 Vääntömomenttipyynti	38
6.5.3 Jarruenergian talteenotto	43
6.5.4 Tiedonkeruu ja CAN-viestien lähettäminen väylään	44
6.6 Käyttöliittymä	47
6.6.1 Suunnittelu	47
6.6.2 Toteutus	47
7 POHDINTA	49
LÄHTEET	50

SANASTO

ACL	Advanced Central Logger, tiedonkeruujärjestelmä
BMS	Battery Management System, akkujenhallintajärjestelmä
CANopen	Väyläprotokolla
I/O	Input/Output, tulo- ja lähtöliitännät
Master	Kenttäväylän isäntälaitte
PLC	Programmable Logic Controller, ohjelmoitava logiikka
PWM	Pulse Width Modulation, pulssimodulaatio
RPDO	Receive Process Data Object, väylälaitteen vastaanottama prosessimuuttuja
Slave	Kenttäväylän orjalaitte
Solmu	CANopen-väylälaitte
TPDO	Transmit Process Data Object, väylälaitteen lähettämä prosessimuuttuja

1 JOHDANTO

Hilux-sähkötyökone on Oulun seudun ammattikorkeakoulun Tekniikan yksikön insinööriopiskelijoiden projekti, joka on pääsääntöisesti toteutettu yhdessä ammattikorkeakoulun opettajien, TEKLin (Oulun tekninen liikelaitos) ja Randax Oy:n kanssa. Tarkoituksena on ollut muuttaa TEKLin käytössä oleva Toyota Hilux -merkkinen työkone täysin sähkökäyttöiseksi. Työkoneesta on poistettu dieselkäyttöinen polttomoottori sekä siihen liittyvät lisälaitteet, joille ei ole tarvetta sähkömoottorikäytön kanssa. Poistettujen osien tilalle työkoneeseen on asennettu akusto, ohjauslogiikka, taajuusmuuttaja ja moottori sekä tehty muutoksia lisälaitteisiin. Sähkötyökoneen on tarkoitus toimia testikäytössä Teknisellä liikelaitoksella sekä tarjota yhteistyökumppaneille alusta laitteidensa testausta varten.

Työkoneen muutostyön ovat toteuttaneet pääosin auto- ja kuljetustekniikan insinööriopiskelijat. Koska moottorinohjaus sekä siihen liittyvien laitteiden ohjaus toteutetaan sähkötyökoneessa ohjauslogiikan avulla, tuli automaatio-opiskelijalle mahdollisuus tehdä logiikan ohjelmointi opinnäytetyönä.

Koska autot ovat kiinnostaneet allekirjoittanutta aina, oli luontevaa yhdistää sekin harrastus että ammatillisuus ja ottaa tämä opinnäytetyö tehtäväksi.

2 SÄHKÖAJONEUVOT

Sähköautoja on ollut käytössä jo paljon ennen polttomoottoriautoja, mutta ne alkoivat menettää suosiotaan kasvavan öljyteollisuuden ja halvan polttoaineen takia. Ensimmäiset sähköautot olivat käytössä jo 1800-luvun loppupuolelta aina 1920-luvulle, jolloin polttomoottoriautot alkoivat syrjäyttää sähköautoja. (1.)

Sähköajoneuvolla, tai sähköautolla, tarkoitetaan ajoneuvoa, joka liikkuu pelkästään esimerkiksi akkuun varastoidun sähköenergian voimalla. Yleensä sähköenergia ladataan akkuihin yleisestä sähköverkosta latureilla. Sähköauton rakenne on hyvin yksinkertainen verrattuna polttomoottori- tai hybridiautoon, jolloin sen vikaantumisherkkyys on huomattavasti pienempi. Lisäksi sen käyttö on paikallisesti täysin saasteetonta, ja mikäli ladattava sähkö on tuotettu päästötömästi, on auton käyttö täysin päästötöntä. Sähkömoottorit voidaan suunnitella ja valmistaa siten, että ne toimivat laajalla kierrosnopeusalueella. Tämä mahdollistaa perinteisen vaihteiston pois jättämisen, mikä puolestaan lisää ajoneuvon hyötysuhdetta sekä toimintavarmuutta. (1.)

Syitä sähköautojen vähäisyyteen on monia, yhtenä niistä akkuteknologian kehittymättömyys. Nykytekniikalla päästään akkukapasiteetin ja auton tehon mukaan jopa 300–500 kilometrin toimintasäteeseen. Tosin markkinoilla olevien autojen tyypillinen toimintasäde on noin 100–150 km. (1.)

Parhaimmillaan sähkökäyttöinen moottori onkin sellaisissa ajoneuvoissa, joissa päivittäinen ajosuorite tapahtuu esimerkiksi taajamissa. Tällöin ajoneuvon nopeudet pysyvät maltillisina, päivittäinen ajomatka on kohtuullisen lyhyt sekä jarruenergian talteenotto toimii tehokkaasti syklisen ajotavan aikana. Lisäksi sähkökäytön hyvät ominaisuudet pääsevät parhaiten oikeuksiin. Näitä ovat muun muassa korkea vääntömomentti alhaisilta nopeuksilta alkaen, nollakulutus ajoneuvon seistessä esimerkiksi liikennevaloissa ja käytön helppous vaihteettomuuden ansioista.

3 LOGIikkaOHJAUS

3.1 Logiikka ohjainlaitteena

Ohjelmoitavat logiikat (PLC, Programmable Logic Controller) kehitettiin alun perin korvaamaan releohjaukset, koska ne eivät olleet joustavia tuotemuutoksia tehdessä. Logiikalla pystytään korvaamaan järjestelmän koosta riippuen satoja releitä ja ohjelmamuutokset on helppo tehdä verrattuna johdotusten muuttamiseen. Ohjelmoitavien logiikoiden toimintojen määrä ja suorituskyky on lisääntynyt prosessoreiden kehityksen myötä. Kehitys mahdollistaa logiikoiden käyttämisen entistä monipuolisimmissa kohteissa. (2, s. 102.)

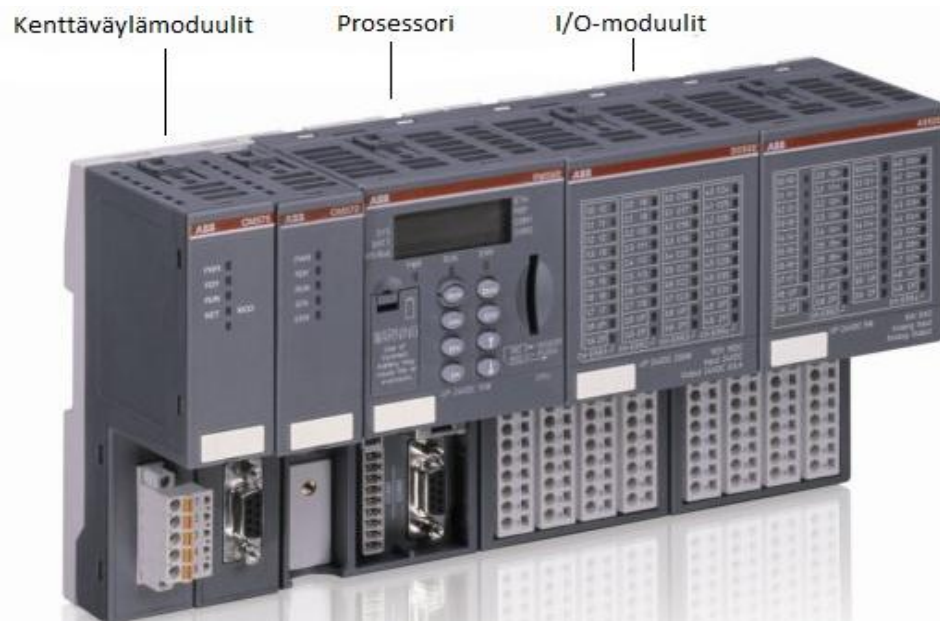
Logiikka ottaa anturilta saamansa tiedon vastaan ja siihen reagoidaan logiikan määräämällä tavalla. Se, miten anturitietoihin reagoidaan, on ohjelmoijan päätettävissä. Reagointinopeus riippuu taasen logiikan prosessorin laskentakyvystä. (2, s. 102.)

Logiikalla voidaan toteuttaa joko askeltavaa ohjausta tai kriteeriohjausta. Askeltavan ohjauksen ideana on, että ohjelma koostuu askelista ja askelissa siirrytään eteenpäin haluttujen ehtojen toteuduttua. Yleensä askeltavaa ohjausta käytetään esimerkiksi jonkin prosessin ylös- tai alasajosekvensseissä. Kriteeriohjaus on tavanomainen ohjaus, jossa ohjelma reagoi anturitietojen muutokseen määrättyllä tavalla. Esimerkiksi sähkötyökone on hyvä esimerkki kriteeriohjauskohteesta. Tuloja seurataan ja niiden muutoksiin reagoidaan ohjelmassa määrättyllä tavalla. (2, s. 102–103.)

3.2 Logiikan rakenne

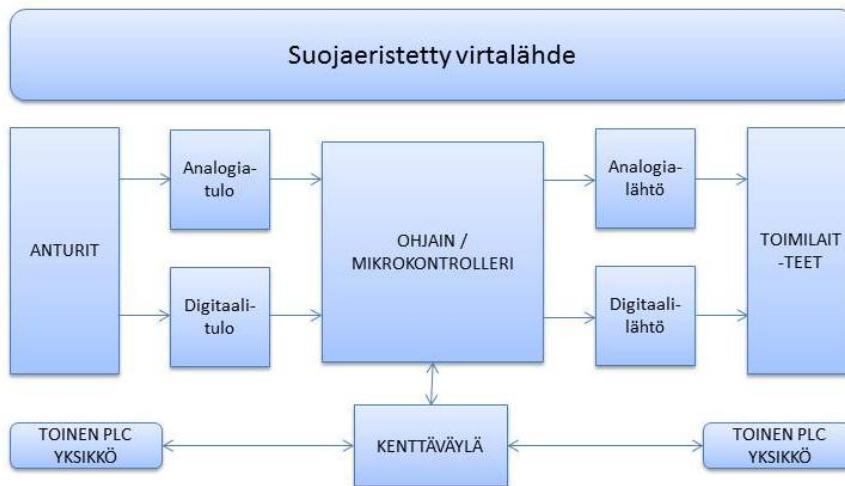
Perinteisesti logiikka koostuu virtalähteestä, prosessorista, erilaisista I/O-moduuleista sekä mahdollisista kenttäväylälaajennusmoduuleista. I/O-moduulien määrä ja tyyppi riippuu ohjattavan prosessin luonteesta ja siinä käytettävistä mittauksista ja ohjauksista. Käytetyimpiä I/O-moduuleja ovat analogia- ja binäärimoduulit. Kaikki moduulit kiinnitetään korttikehikoihin tai takalevyihin, jolloin ne ovat sisäisesti kytkettynä toisiinsa eikä niitä tarvitse keskenään yhdistää johtimilla. (2, s. 106.)

Kuvassa 1 on tyypillinen ohjelmitava logiikka I/O-moduuleineen.



KUVA 1. Tyypillinen ohjelmitava logiikka I/O-moduuleineen (3)

Kuvassa 2 on esitetty logiikan sisäinen rakenne. Tieto luetaan antureilta joko analogia- tai digitaalitulon kautta prosessorille. Prosessori käsittelee tiedot ja ohjaa analogia- tai digitaalilähtöihin liitettyjä toimilaitteita ohjelman mukaisesti. Tuloja ja lähtöjä voidaan lukea ja kirjoittaa myös kenttäväylän kautta. Esimerkiksi taajuusmuuttajaa voidaan kytkeä logiikkaan kenttäväylän kautta, jolloin kaikki I/O-viestit kulkevat yhtä väyläjohdinta pitkin. Jos kytkentä toteutettaisiin perinteisellä I/O-johdotuksella, jokaiselle mittaukselle pitäisi olla oma parikaapelinosa. Luvussa 4 on käsitelty tarkemmin CANOpen-kenttäväylän toimintaperiaatetta.



KUVA 2. Tyypillinen logiikan sisäinen rakenne

3.3 Logiikan ohjelmointi

Ohjelmoinnin lähtökohtana on usein toimintakaavio tai -selostus, jossa kuvataan ohjattavan prosessin haluttu toiminta. Selostuksen pohjalta suunnittelija luo logiikkakaaviot tai toimintadiagrammit ja näiden perusteella tehdään varsinainen ohjelma. (1, s. 117.)

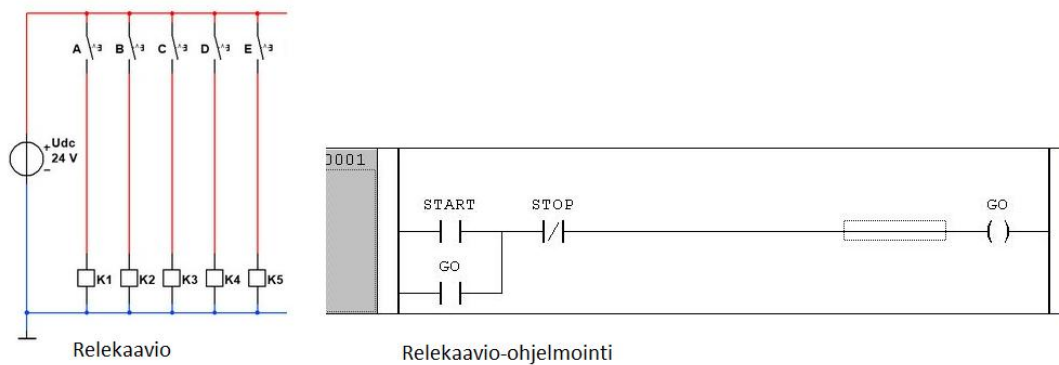
Yleensä ohjelma kirjoitetaan logiikkavalmistajan toimittamalla ohjelmistolla tai laitteistolla. Ohjelmointitapoja on useita erilaisia. Eri laitevalmistajien tuotteissa käytössä olevien ohjelmointikielten määrä vaihtelee eri logiikoiden välillä. Yleisimpiä ohjelmointikieliä ovat logiikkakaavio-ohjelmointi (FBD, Function Block Diagram), relekaavio-ohjelmointi (LD, Ladder Diagram), sekvenssiohjelmointi (SFC, Sequential Function Chart), käskylista (IL, Instruction List) ja strukturoitu teksti (ST, Structured Text). (1, s. 117.)

Monilla valmistajilla on omia logiikkakohtaisia ohjelmistoja ja laitteistoja. Tosin monet logiikkavalmistajat pyrkivät tekemään logiikoistaan tietyn standardin mukaisia, jotta yrityksen logiikoita voidaan ohjelmoida kolmannen osapuolen tekemillä ohjelmistoilla. Tällainen järjestely on järkevää siltä kannalta, että hallitessaan yhden ohjelmiston ohjelmoija pystyy ohjelmoimaan eri valmistajien laitteita. (1, s. 117.)

3.4 Ohjelmointikielet

3.4.1 Relekaavio (Ladder Diagram, LD)

Ladder Diagram eli relekaavio-ohjelmointi on logiikan perinteinen ohjelmointitapa. Relekaavio-ohjelmointia käytettiin alun perin, koska perinteinen releohjaus on helppo muuttaa relekaavio-ohjaukseksi. Helppous tulee siitä, että relekaavio-ohjelmoinnin asettelu muistuttaa hyvin paljon perinteistä relekaaviota (kuva 3). Ohjelma koostuu virtapiireistä (network), jotka rakentuvat erilaisista toimilohkoista sekä JA- ja TAI- piireistä. Toimilohkot voivat olla esimerkiksi Set-Reset-piirejä, ajastimia, datamuunnoksia tai matemaattisia toimilohkoja. (1, s. 121.)

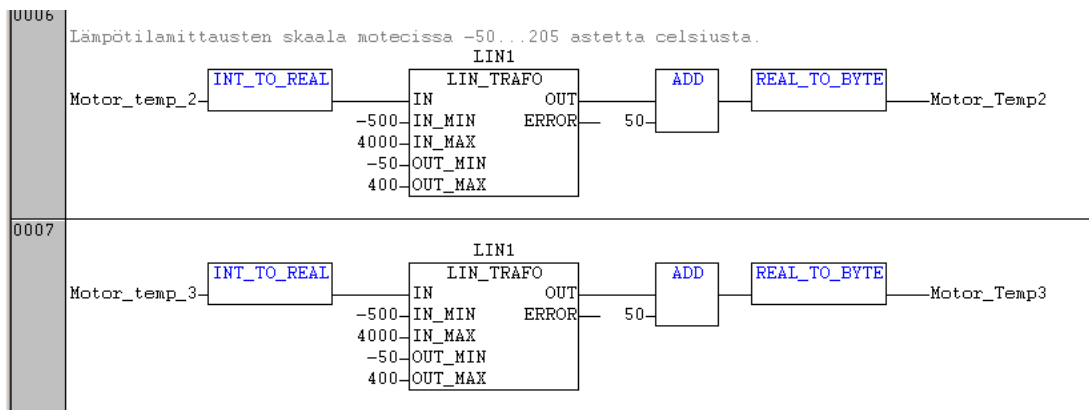


KUVA 3. Esimerkki relekaaviosta ja relekaavio-ohjelmointi sovelluksesta

3.4.2 Logiikkakaavio (Function Block Diagram, FBD)

Logiikkakaavio-ohjelmointi on graafinen ohjelmointitapa, jossa pystytään käsittelemään sekä binääri- että analogiatietoja. FBD:n rakenne koostuu networkeista, eikä siinä ole takaisinkytkentämahdollisuutta. Ohjelmointitapa perustuu IEC 61131-3 -standardiin ja siinä määriteltyihin logiikkasymboleihin. (4.)

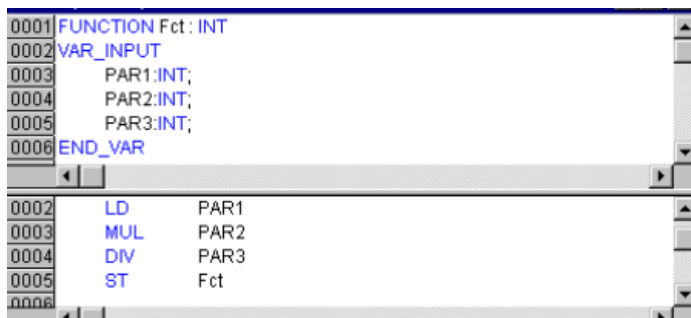
Kuvassa 4 on esimerkki ohjelmasta, joka on tehty FBD:llä. Esimerkissä on kaksi niin sanottua virtapiiriä.



KUVA 4. Esimerkki Function Block Diagram -ohjelmointitavasta

3.4.3 Käskylista (Instruction List, IL)

Käskylistaohjelmointi muistuttaa Assembly-kieltä. Käskyrivit koostuvat käskystä sekä siihen liittyvistä operandeista. Virtapiiri aloitetaan latauskäskyllä (LD), jota seuraa muut tarvittavat toimenpiteet. Lopuksi virtapiiri päättyy jonkin lähdön tai muistipaikan ohjaamiseen. (Kuva 5.) (1, s. 122.)



KUVA 5. Esimerkki ohjelmoinnista IL-kielillä

3.4.4 Strukturoitu teksti (Structured text, ST)

ST on korkean tason ohjelmointikieli. Siinä voidaan käyttää valintausekkeitä, kuten IF-ELSEIF-END-IF sekä FOR-silmukoita. ST-kieli tukee myös REPEAT-UNTIL- ja WHILE-DO-silmukoita ja erilaisia toimintoja, kuten SQRT() ja SIN(). (Kuva 6.) (1, s. 123.)

```

0001 (*taajuusmuuttajan ohjaus*)
0002
0003     (*Taajuusmuuttaja voidaan käynnistää*)
0004     IF Inverter_control.Ready THEN
0005         CW :=6;
0006     ELSE
0007         CW:=2;
0008     END_IF;
0009
0010     (*Taajuusmuuttajan "enablointi"*)
0011     IF (Unpack_SW.Ready_to_switch_on AND Inverter_control.ENABLE) THEN
0012         CW :=7;
0013     END_IF;
0014
0015     (*Taajuusmuuttajan käynnistys*)
0016     IF Unpack_SW.Switched_on AND Inverter_control.START_STOP THEN
0017         CW:=15;
0018
0019     END_IF;
0020
0021     (*Taajuusmuuttajan pysäytys*)
0022     IF (Unpack_SW.Operation_enabled AND NOT Inverter_control.START_STOP) THEN
0023         CW :=7;
0024     END_IF;
0025

```

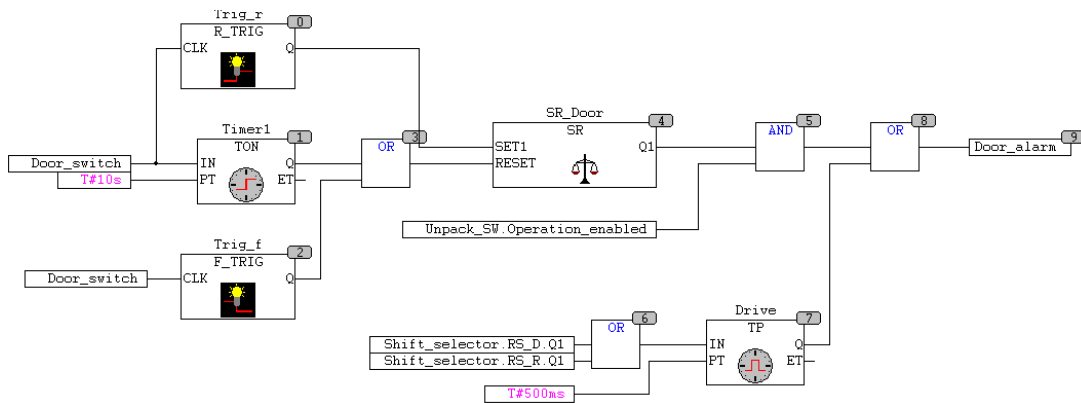
KUVA 6. Esimerkki ohjelmoinnista ST-kielillä

3.4.5 Sekvenssiohjausohjelmointi (Sequential Function Chart, SFC)

Sekvenssiohjelmointia käytetään silloin, kun halutaan jonkun prosessin ohjauksen etenevän askelmaisesti. Tällöin ohjelma koostuu askeleista ja siirtoehdoista. Jokin toimenpide suoritetaan, jonka jälkeen tarkistetaan, onko jokin haluttu ehto toteutunut. Toteutuneen ehdon jälkeen siirrytään seuraavaan askeleeseen. Tämä ohjelmointitapa soveltuu hyvin vaikkapa prosessin ylösajoon. (1, s. 123.)

3.4.6 Jatkuvatoiminen ohjelmointi (Continuos Function Chart, CFC)

CFC-ohjelmointi on FBD:n tyyppinen ohjelmointitapa, mutta FBD:stä poiketen siinä ei ole virtapiirejä, vaan toimilohkot voidaan sijoittaa vapaasti. CFC:llä on helppo toteuttaa takaisinkytkennät sekä tarvittaessa voidaan muuttaa toimilohkojen suoritusjärjestystä. Ohjelmointitapa ei ole niin nopea kuin FBD, mutta on paljon joustavampi (4). Kuvassa 7 on esimerkki CFC-ohjelmointitavalla tehdystä sovelluksesta.



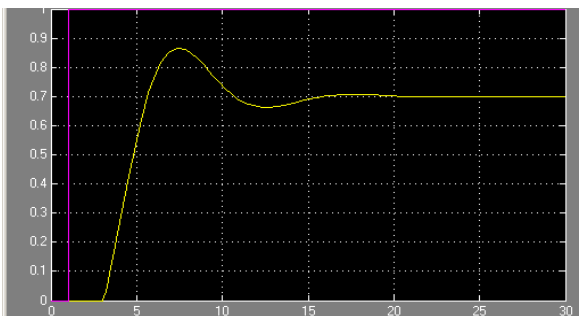
KUVA 7. Esimerkki Continuous Function Chart -ohjelmointitavalla tehdystä sovelluksesta

3.5 PID-säädin

PID-säädin (Proportional-Integral-Derivate) on yleisimmin käytetty säädin. Säädin koostuu kolmesta osasta, P, I ja D. Säädettävästä prosessista riippuen säätimestä voidaan käyttää eri variaatioita. Niitä ovat P-, PI-, PID- tai PD-säätimet. Sopiva säädin valitaan ohjattavan prosessin mukaan. Säätimen tulona on asetusrvon ja mittauksen eroosuure. Säätimen ulostulo on puolestaan P-, I- ja D-osien summa. (5, s. 44.)

3.5.1 P-säätö

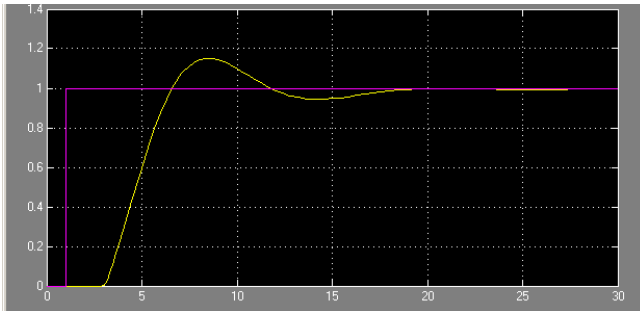
P-säädössä ohjaussignaalin arvo riippuu suoraan erosuureen (mittaus - asetusrvon) arvosta. Eroosuure kerrotaan vahvistusarvolla K_p , jolloin saadaan P-osan ulostulo. P-säädölle on tyypillistä, että se ei pysty kokonaan kompensoimaan eroosuuretta vaan mittauksen ja asetusrvon välille jää yleensä poikkeama (kuva 8). (5, s. 45.)



KUVA 8. P-säätimellä varustetun prosessin reagointi askelmuutokseen

3.5.2 I-säätö

Integraaliosaa käytetään yleisesti P-osan yhteydessä. Koska P-säädin ei yksinään pysty poistamaan kokonaan säätövirhettä mittauksen ja asetusravon väliltä, sopii integraali osa siihen hyvin. I-säädin integroi erosuuretta (säätövirhettä), jolloin säätimen lähtö muuttuu niin kauan, kuin erosuuretta on olemassa. Kuvassa 9 on esimerkki I-säädön vaikutuksesta säädön toimintaan. (5, s. 47.)

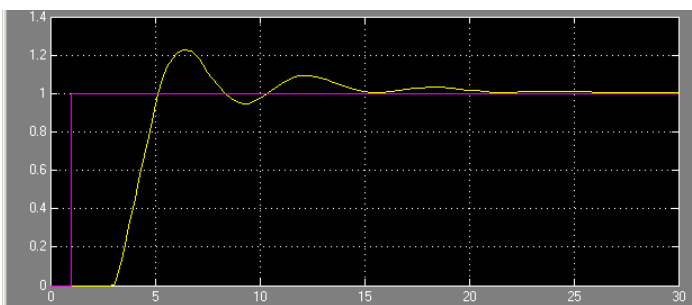


KUVA 9. PI-säätimellä varustetun prosessin reagointi askelmuutokseen

3.5.3 D-säätö

Derivoivaa säätöä käytetään P- tai PI-säätimen yhteydessä. Derivoiva säätö reagoi erosuureen muuttumisnopeuteen. Mitä suurempi muutosnopeus (kulma-kerroin) on, sitä suurempi on D-säädön ulostulo. D-osan vaikutusta voidaan jakaa pitemmälle ajalle käyttämällä derivoinnin suodatusta. (5, s. 48–49.)

Kuvassa 10 on esimerkki D-säädön vaikutuksesta säädön toimintaan. Kuvien 8, 9 ja 10 prosessit ovat samanlaiset, joten kuvista on helppo vertailla, miten eri säätimet vaikuttavat prosessin ulostuloon.

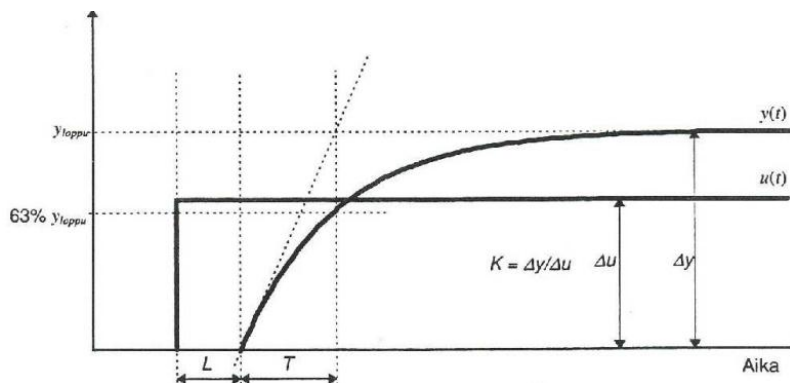


KUVA 10. PID-säätimellä varustetun prosessin reagointi askelmuutokseen

3.5.4 Viritys

Säätimen viritys on periaatteessa kolmen viritysparametrin (K_p , T_i , T_d) hakemista siten, että prosessi saadaan käyttäytymään halutulla tavalla. Parametreilla on riippuvuussuhde toisiinsa, joten niitä ei yleensä voida virittää yksi kerrallaan. Tietyissä tapauksissa säädintä voidaan kuitenkin virittää yksi parametri kerrallaan, jolloin säätimen parametreja yritetään saada oikeaksi kokeilemalla. Näin ollen säätöpiireille riittää joskus vähemmän analyttinen säätötapa, kunhan se tehdään silti huolellisesti. Kriittisten prosessien kohdalla järjestelmän dynamiikkaa ja säätörakennetta on tarkasteltava tarkemmin. (5, s. 80.)

Säätimen viritystä varten pitää määrittää prosessin askelvasteesta viive, nousuaika ja vahvistus. Se aika, joka kuluu yksikköaskeleen antamisesta ulostulossa näkyvään muutokseen, on viive (L). Nousuaika (T) on se aika, joka kuluu, kun ulostulo on saavuttanut 63,2 % ulostulon loppuarvosta. Vahvistus (K) on ulostulon ja yksikköaskeleen suhde. (Kuva 11.) (5, s. 80.)



KUVA 11. Yhden aikavakion prosessin askelvaste, josta määritetään parametrit PID-säätimen viritämistä varten (5, s. 80)

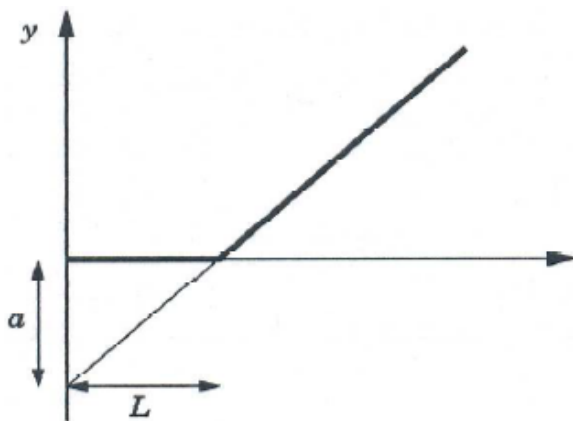
Säätimen viritukseen on olemassa useita eri kaavoja ja tapoja. Yksi käytetyistä menetelmistä on Ziegler-Nichols-menetelmä, joka on esitelty jo vuonna 1942. Menetelmä antaa tyypillisesti hieman liian suuren ylityksen, joten tätä menetelmää käytetäänkin lähinnä suuntaa antavana lähtökohtana. Ziegler-Nicholsin menetelmässä on kaksi tapaa virittää säädin, joko avoimen piirin askelvasteeseen (kuva 11) perustuen tai suljetun piirin taajuusvasteeseen perustuen.

Viritystaulukossa (taulukko 1) on laskukaavat ensimmäisen kertaluvun viiveellisen prosessimallin säätimen virittämiseen.

TAULUKKO 1. Ziegler-Nichols-menetelmän laskukaavat (6)

	K_p	T_i	T_d
P	$T/(KL)$		
PI	$0,9T/(KL)$	$3L$	
PID	$1,2T/(KL)$	$2L$	$L/2$

Ziegler-Nicholsin askelvastemenetelmässä pitää integroivalle prosessille määrittää integroivan prosessin askelvasteesta omat parametrit (kuva 12).



KUVA 12. Integroivan prosessin askelvaste, josta määritetään parametrit PID-säätimen virittämistä varten (6)

Taulukossa 2 on esitetty laskukaavat PID-säätimen viritysarvojen laskentaan integroivalle prosessille.

TAULUKKO 2. Integroivan prosessin laskukaavat (6)

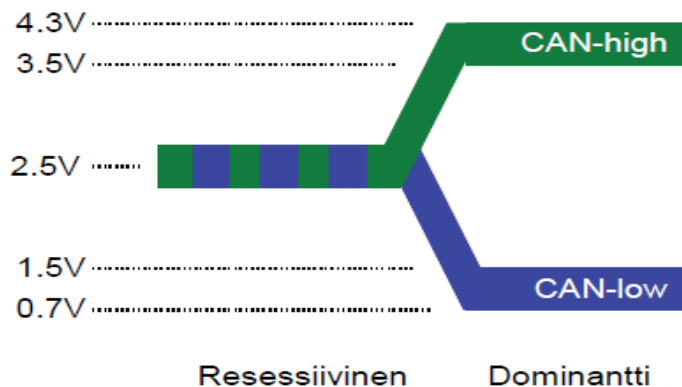
	a/K_p	T_i/L	T_d/L
P	1		
PI	0,9	3	
PID	1,2	2	$L/2$

4 CANOPEN-PROTOKOLLA

4.1 Toiminta

CANopen on korkean tason tiedonsiirtoprotokolla, minkä siirtoyhteys- ja fyysinen kerros perustuu perinteisesti CAN-väylään. CAN-väylä (Controller Area Network) on suunniteltu alun perin autojen hajautettujen ohjausjärjestelmien tiedonsiirtoon, esimerkiksi moottorinohjausyksikön ja vaihteistonohjausyksikön väliseen kommunikointiin. Nykyään väylää sovelletaan laajasti erilaisissa ympäristöissä kuten ajoneuvoissa, hisseissä, maatalouskoneissa, lääketieteellisissä laitteissa, kappaletavara-automaatiossa, roboteissa, tekstiilikoneissa, mittausjärjestelmissä, ohjelmoitavissa logiikoissa ja työkoneissa. (7.)

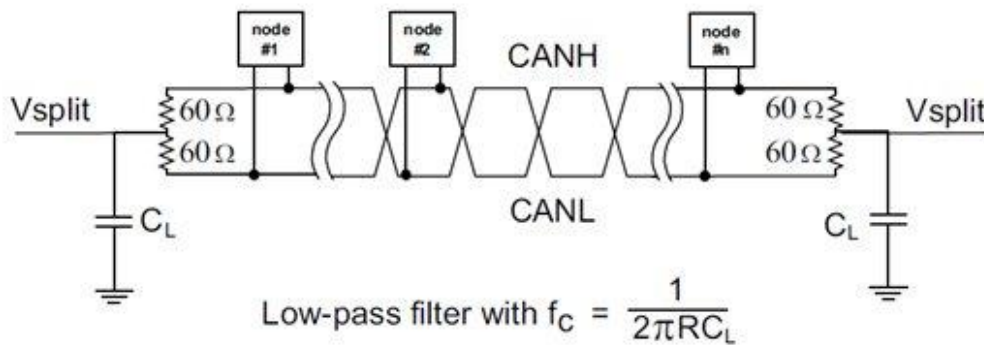
CANopen käyttää CAN-väylän High Speed CAN -tyyppistä fyysistä kerrosta, joka on yleisin fyysinen kerros, jota CAN-väyläsovelluksissa käytetään. Väyläsignaalien lukeminen perustuu väyläjohtimien väliseen jännite-eroon ja sen mittaamiseen. Dominantissa tilassa väyläjohtimien välinen jännite-ero voi vaihdella sallitun toleranssin rajoissa. (Kuva 13). (7.)



KUVA 13. CAN-väylän tilat (7)

Resessiivinen tarkoittaa tilannetta, jossa mikään väylän solmuista ei aktiivisesti lähetä väylälle mitään tai kaikki solmut lähettävät loogista ykköstä. Dominanttila vallitsee väylällä silloin, kun vähintään yksi solmu lähettää loogista nollaa. (Kuva 13.) (7.)

CANopen-väylä koostuu erilaisista solmuista (Node), joista yksi on master-tyyppinen ja muut slave-tyyppisiä. Jokaisella solmulla on oma tunniste (node-id). Solmut liitetään toisiinsa kierretyllä parikaapelilla ja väylä terminoidaan 120 ohmin päätevastuksilla molemmista päistä. Normaalin terminoinnin lisäksi voidaan käyttää ns. split-termination-tapaa, jossa kaksi 60 ohmin vastusta kytetään sarjaan ja niiden välinen kytkentäpiste maadoitetaan kondensaattorin läpi maihin (kuva 14). Split-termination vastaa alipäästösuodatinta, jolla väylässä olevia häiriöitä saadaan vähennettyä. (8.)



KUVA 14. Split-termination (8)

Yleensä väylä kestää häiriöitä paljon, koska ulkoiset häiriöt kytkeytyvät saman merkkisiin molempiin väyläjohtimiin. Koska väylän signalointi perustuu kahden johtimen väliseen jännite-eroon, kumoutuu niihin yhteismuotoisesti kytkeytynyt häiriö automaattisesti pois. (8.)

Väyläkaapelina käytetään tyypillisesti häiriösuojattua kierrettyä parikaapelia. Kaapelissa tulee olla kierretyn parikaapelin lisäksi johdin yhteistä maapotentiaalia varten. Kaapelin suojavaippa toimii häiriönsuojana ja yleensä se maadoitetaan vain toisesta päästä väyläkaapelia. Yhteinen signaalimaajohto sitä vastoin pitää kytkeä kaikille solmuille. (8.)

4.2 Protokollat

CANopen tarjoaa peruspalveluiden toteuttamiseen useita protokollia. Seuraavassa on lueteltuna yleisempiä protokollia, joita väylän kanssa tarvitaan. Seuraavassa esiintyvällä väylän käynnistämällä tarkoitetaan sitä, kun kenttäväylään liitettyihin laitteisiin kytketään virrat ja ne alkavat kommunikoida keskenään.

- **BOOT**-protokollan avulla CANopen-solmu ilmoittaa käynnistymisestään muille solmuille samassa väylässä. Tietoa käyttää lähinnä master-laite käynnistäessään väylän toimintaa.
- **Heartbeat**-protokollan avulla solmu voi tuottaa väylään tietoa omasta tilasta muiden solmujen käytettäväksi.
- **SDO (Service Data Object)** -protokollan avulla luetaan ja kirjoitetaan CANopen-solmun objekti kirjaston arvoja. Luettavia arvoja voivat olla esimerkiksi toimilaitteen parametrintiasetukset. SDO-osoituksia käytetään pääsääntöisesti vain master- ja slave-solmujen välillä.
- **NMT (Network Management)** -protokollan avulla master-solmu siirtää samaan väylään kytkettyjä slave-solmuja verkonhallintatilasta toiseen. NMT-protokolla käyttää apuna slave-solmuilta tulevaa BOOT-protokollan tietoa siirtäessään slave-solmua toimintatilaan.
- **SYNC**-protokollan avulla voidaan PDO-kehysten lähetys tahdistaa. Synkroniset PDO-kehykset voidaan asettaa lähtemään jokaisen, joka toisen, joka kolmannen jne. vastaanotetun synkronointikehysten jälkeen.
- **PDO (Process Data Object)** on se protokolla, jolla siirretään prosessisignaaleja, esim. lämpötila-anturin tietoa, kahden solmun välillä. TPDO (Transmit Process Data Object) on lähettävää tietoa ja RPDO (Receive Process Data Object) vastaanottavaa tietoa. Lähetys ja vastaanotto katsotaan aina siitä suunnasta, minkä solmun PDO:sta puhutaan. Esimerkiksi akuston jännite on akkujenhallintajärjestelmällä TPDO ja vastaanottavalle solmulle (PLC) kyseinen tieto on RPDO.
- **EMCY (Emergency)** -protokollan avulla voidaan verkossa siirtää virheilmoituksia. Mikäli johonkin väylän solmuista tulee vika, se lähettää väylään EMCY-viestin. (9.)

4.3 Solmun verkonhallintatilat

Jokaisella väylässä olevalla solmulla on kolme kiinteää verkonhallintatilaa, pre-operational, operational ja stopped. Lisäksi jokaisella on kaksi hetkellistä verkonhallintatilaa, reset node ja reset communication. Väylän master-solmu ohjaillee slave-solmuja eri tiloihin NMT-protokollan avulla. Kiinteän ja hetkellisen verkonhallintatilan erot ovat siinä, että solmu jää kiinteään tilaan odottamaan uutta käskyä, kun taas hetkellisessä tilassa käydään hetkellisesti. Master-solmu voi NMT-komennolla siirtää slave-solmun mistä tahansa kiinteästä tilasta toiseen kiinteään tai hetkelliseen tilaan. (9.)

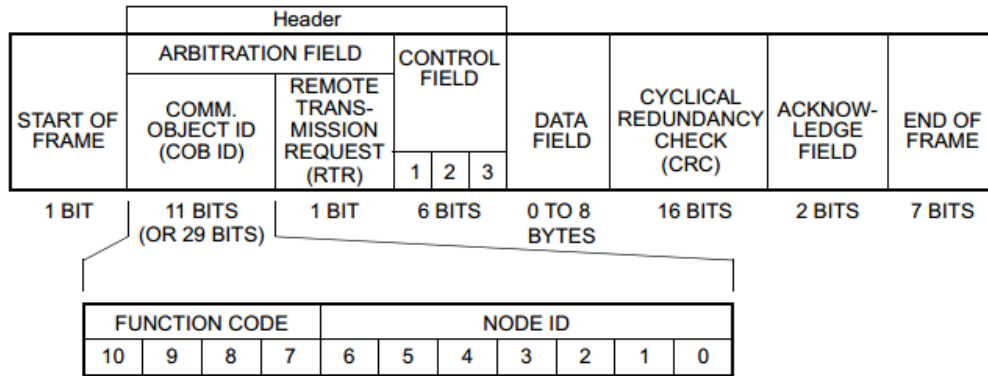
- **Reset node** -tilassa koko solmu resetoidaan eli käynnistetään uudelleen.
- **Reset communication** -tilassa alustetaan ja käynnistetään solmun protokollapino. Muutettaessa esimerkiksi solmun PDO-parametrejä pitää solmulle suorittaa Reset communication -käsky, jotta muutokset tulevat voimaan.
- **Pre-operational**-tilassa solmu odottaa masterilta hallittua väylän käynnistyskäskyä sekä mahdollisia tarkistuksia sekä parametrin muutoksia. Tässä tilassa prosessisignaalien siirtäminen ei ole sallittua, joten mahdollisista vääristä signaaliliikenteen asetuksista päivitysvaiheessa ei aiheudu vaaraa prosessille. Mahdolliset solmun virhetilanteet voidaan siirtää EMCY-kehyksellä pre-operational-tilassa.
- **Operational**-tilassa CANopen-solmu on normaalissa toimintatilassa jolloin myös prosessisignaalien (PDO) päivitys on sallittua. Myös parametrimuutokset onnistuvat SDO-kehyksillä sekä tapahtumailmoitukset EMCY-kehyksillä.
- **Stopped**-tilassa solmu ei voi ottaa väylältä muita kuin NMT-komentoja ja se saa lähettää väylälle vain omasta tilasta kertovia node guarding- tai heartbeat-viestejä. Jos solmuun tulee jokin vakavampi häiriö, se ohjataan stopped-tilaan. (9.)

4.4 Viestikehyksen rakenne

CANopen-väylässä tiedonsiirto suoritetaan viestikehyksillä (kuva 15).

CAN data frame

CAN employs data frames for transferring data between the host (controller) and the nodes on the bus. The following figure presents the structure of the data frame.



KUVA 15. CAN-viestikehyksen rakenne (9)

Viestikehyksen COB ID-kentän 11 bittiä määrittävät, minkä tyyppistä tietoa datakehys sisältää ja miltä solmulta ne ovat tulossa. Väylässä liikkuvat viestit ovat siis kaikkien nähtävillä ja vastaanottava solmu tunnistaa itselleen tulevat viestit käyttäjän konfiguroinnin perusteella. Esimerkiksi jos solmu, jonka node-id on 1, lähettää prosessimuuttujien arvoja (TPDO) väylään, on sen kehyksen COB ID 00110000001 eli 181h. Kuusi vähiten merkitsevää bittiä kertoo node id:n ja neljä eniten merkitsevää kommunikointiobjektin tyyppin. Mikäli solmu 2 lähettäisi saman ryhmän (TPDO) muuttujia väylään, olisi sen viestin COB ID 00110000010 eli 182h. (10.)

Control field sisältää kuusi bittiä ja niistä neljä vähiten merkitsevää bittiä kertoo datakentässä olevan datan pituuden. Control fieldin jälkeen tulee varsinainen datalle varattu osuus (data field), joka voi olla aina 8 tavun mittainen eli 64 bittiä. Datakenttä voi sisältää samalta solmulta useita eri arvoja. Väylää konfiguroitaessa pitää määrittää, mitä muuttujia on missäkin siirtokehyksessä siirretään. Lopuksi viestikehyksessä on virheentarkistus- ja hyväksyntäbittejä. Viestikehyksen vastaanottava solmu pakottaa hyväksyntäbitin dominanttiin tilaan vastaanotettuaan viestikehyksen. (10.)

5 HILUX-SÄHKÖTYÖKONEEN OHJAUSJÄRJESTELMÄ

5.1 Moottori

Moottori on Randax Oy:n valmistama, nestejäähdytetty kestopagnetoitu tahtikone. Moottorin suoritusarvot ovat suhteellisen isot verrattuna tämänhetkisen taajuusmuuttajan ja akuston kapasiteettiin, joten moottori ei käy pieneksi päivitettäessä akustoa ja taajuusmuuttajaa isommaksi. Taulukossa 3 on esitetty moottorin tekniset tiedot.

TAULUKKO 3. Moottorin tekniset tiedot (11)

Suurin vääntömomentti	300 Nm
Nimellisvääntömomentti	160 Nm
Nimellisteho	70,4 kW
Nimellistaajuus	280 Hz
Nimellisottovirta	120 A
Nimellisjännite	400 V
Nimellispyörintänopeus	4200 rpm
Suurin pyörintänopeus	5300 rpm

5.2 Taajuusmuuttaja

Taajuusmuuttajalla saadaan ohjattua moottorin vääntömomenttia ja nopeutta portaattomasti. Taajuusmuuttajan avulla saadaan moottoria ajettua yli nimellispyörintänopeuden, mikä mahdollistaa moottorin paremman hyödyntämisen laajan kierrosnopeusalueen ansiosta. Sähkötyökoneessa käytetty taajuusmuuttaja koostuu tasasuuntaajasta, välipiiristä ja vaihtosuuntaajasta. Tasasuuntaajan tehtävän on tasasuunnata vaihtojännite tasajännitteeksi. Välipiiri koostuu latausvastuksesta ja kondensaattoreista. Latausvastuksen avulla rajoitetaan kondensaattoreiden latausvirtaa kytkettäessä jännite taajuusmuuttajalle. Konden-

saattorit pyrkivät pitämään välipiirin jännitteen tasaisena. Vaihtosuuntaajan tehtävä on vaihtosuunnata tasasähkö moottorille meneväksi vaihtosähköksi. Muunnos tehdään IGBT (Insulated Gate Bipolar Transistor) -kytkimien avulla (12). Koska sähkötyökoneessa saadaan akustolta suoraan tasajännitettä, ei tässä sovelluksessa ole tarvetta käyttää taajuusmuuttajan tasasuuntausta vaan jännite voidaan kytkeä suoraan taajuusmuuttajan välipiiriin. Taulukossa 4 on taajuusmuuttajan ominaisuudet.

TAULUKKO 4. Taajuusmuuttajan ominaisuudet (12)

Nimellisteho	45 kW
Nimellinen lähtövirta	90 A
Hetkellinen maksimilähtövirta	150 A
Jatkuva lähtövirta, 4 kHz kytkentätaajuus	93 A

5.3 Logiikka

Ajoneuvon toimintaa ohjataan ABB:n AC500-sarjaan kuuluvalla PM 573-ETH-XC -ohjelmoitavalla logiikalla. Laitteisto koostuu PM 573 -prosessorista, AI531-, DA501- ja CD522-I/O-moduuleista sekä CM578-master-moduulista.

5.3.1 Prosessori

PM 573-ETH-XC-prosessoriin voidaan kytkeä yhteensä neljä erilaista kenttäväylää sekä yhteensä kymmenen analogista tai digitaalista I/O-moduulia. Jokaiselle kenttäväylälle pitää olla oma liityntämoduulinsa. Prosessorin liitynnät on lueteltuna seuraavassa. (13.)

Sarjaportti 1

9-pinnisellä riviliittimellä varustetun sarjaportin kautta voidaan suorittaa

- ohjelmointi, on-line-seuranta
- kommunikointi COM_SEND- ja COM_REC-toimilohkoilla

- liityntä Modbus RTU -väylään master- tai slave-laitteena
- liityntä CS31 järjestelmäväylään master-laitteena. (13.)

Sarjaportti 2

9-pinnisen D-liittimellä varustetun sarjaportin kautta voidaan suorittaa

- ohjelmointi, on-line-seuranta
- kommunikointi COM_SEND- ja COM_REC-toimilohkoilla
- liityntä Modbus RTU -väylään, master- tai slave-laitteena. (13.)

Ethernet

Ethernet-liitynnän kautta voidaan suorittaa

- ohjelmointi, on-line-seuranta
- kommunikointi ETH_UDP_SEND- ja ETH_UDP_REC-toimilohkoilla
- liityntä Modbus TCP/IP -väylään, master- tai slave-laitteena. (13.)

Arcnet

Arcnet liittimellä voidaan kytkeytyä prosessorin sisäiseen arcnet-tietoliikennemuoduuliin. (13.)

FBD

FBD-liittimen kautta prosessori voidaan liittää slave-laitteena kenttäväylän master-laitteeseen. (13.)

Prossessorissa on myös muistikorttipaikka, jonka avulla muistikortille voidaan tallentaa varmuuskopiot ja ohjelmat sekä suorittaa laiteohjelmiston (firmware) päivitykset.

5.3.2 Kenttäväylä- ja I/O-moduulit

CM578 CANopen-kenttäväylämoduuli

CM578-laajennusmoduulilla saadaan ohjelmoitava logiikka kytkettyä CANopen-väylään. Moduuli kykenee toimimaan 10 Kbit/s – 1Mbit/s nopeudella. Taajuus-

muuttaja, akkujenhallintajärjestelmä ja tiedonkeruulaitteisto on liitetty CANopen-kenttäväylän avulla logiikkaan.

CD522

CD522 laajennusmoduulissa on seuraavat ominaisuudet:

- 2 enkooderituloa
- 2 pulssimodulaatiolähtöä
- 2 digitaalista tuloa 24 VDC
- 8 ohjelmoitavaa digitaalista tuloa tai lähtöä. (13.)

AI531

AI531 moduulissa on kahdeksan ohjelmoitavaa analogista tuloa. Tuloihin voidaan kytkeä joko jännite- tai virtaviestejä sekä erilaisia lämpötila-antureita. Tulojen voidaan käyttää myös digitaalisina tuloina. (13.)

DA501

DA501 moduuli koostuu seuraavista ominaisuuksista:

- 16 digitaalituloa, 24 VDC
- 8 ohjelmoitavaa digitaalituloa tai -lähtöä, 24 VDC
- 4 analogiatuloa
- 2 analogialähtöä. (13.)

5.4 Akusto ja akkujenhallinta

5.4.1 Akusto

Työkoneeseen on asennettu European Batteries Oy:n valmistamat LiFePo₄-akut. LiFePo₄ tulee sanoista Lithium iron phosphate. Käyttöönottovaiheen akut on sijoitettu lavan etuosaan heti matkustamon taakse. Käyttöönottovaiheen akusto käsittää seitsemän sarjaan asennettua 3,5 kWh:n akkua, jolloin kokonaiskapasiteetiksi tulee 24,5 kWh. Nimellisjännite akustolla on 537,6 V. Taulukossa 5 on esitetty käyttöönottovaiheen akuston tekniset tiedot.

TAULUKKO 5. Akuston tekniset tiedot

Nimellisjännite	537,6 V
Kapasiteetti	45 Ah
Energiasisältö	24,5 kWh
Jatkuva kuormitusvirta	45 A
Maksimivirta 30 s	135 A
Maksimivirta 10 s	160 A

Loppuvuoteen 2013 mennessä sähkötyökoneen akkukapasiteetti kaksinkertaistetaan, jolloin kokonaiskapasiteetiksi tulee 49 kWh. Lisäkapasiteetin akut asennetaan auton konetilaan entisen polttoainekäyttöisen moottorin tilalle.

5.4.2 Akkujenhallintajärjestelmä (BMS)

Akuston toimintaa ohjaa akkujenhallintajärjestelmä. Jokaisessa akussa on moduuli, joka tuottaa erilaista informaatiota, muun muassa jännite- ja lämpötilatietoa, BMS:n käyttöön. Akkumoduulit on kytketty sisäisellä CAN-väylällä akkujenhallintajärjestelmään. Akkujenhallintajärjestelmä liitetään muihin laitteisiin ulkoisen CANopen-väylän kautta. Akkujenhallintajärjestelmää ohjataan väylän kautta ohjaussanoilla. Näiden käskyjen perusteella järjestelmä ohjaa sekä pää- että latauskontaktoreita.

5.5 Ohjauspaneeli

Työkoneeseen asennettiin ABB:n valmistama CP 620 -ohjauspaneeli (kuva 16). Paneelin välityksellä ei ole tarkoitus ohjata järjestelmän toimintaa, vaan ainoastaan antaa tietoa kuljettajan käyttöön. Paneelissa on kaksi Ethernet-porttia. Toisella kytkeydytään logiikkaan ja toiseen voidaan kytkeä tietokone. Tietokoneella voidaan tehdä ohjauspaneelin ja logiikan ohjelmointi sekä on-line-seuranta.

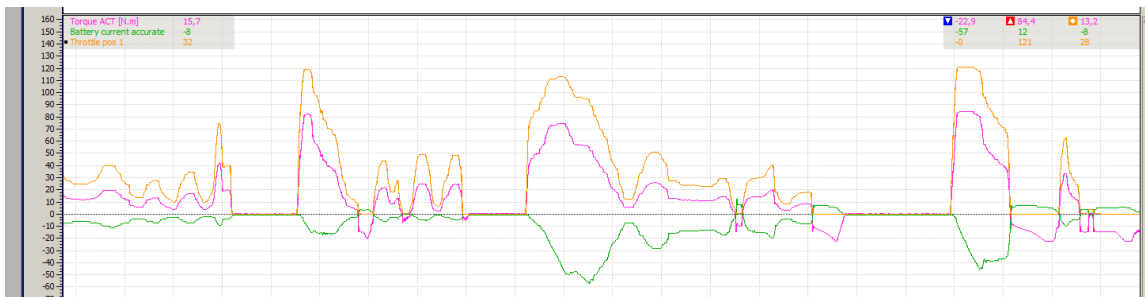


KUVA 16. CP 620 näyttöpaneeli (14)

5.6 Tiedonkeruu

Tiedonkeruulaitteisto on työkoneen toiminnan analysoinnin kannalta tärkein järjestelmä. Laitteistoksi asennettiin Motecin valmistama ACL (Advanced Data Logger). Laitteiston sisäiseen muistiin voidaan tallentaa dataa yhteensä 4 GB:n verran. Tallennuskapasiteetin suuruus riippuu datan näytteenottotaajuudesta. Kriittisiä suureita, kuten virta ja nopeus, pitää tallentaa suhteellisen nopealla näytteenottotaajuudella, kun taas esimerkiksi akuston lämpötilaa tai varaustilaa voidaan tallentaa huomattavasti harvemmin. Toisaalta dataa ei ole mitään järkeä tallentaa yhtään suuremmalla näytteenottotaajuudella kuin millä sitä mitataan.

Tallennettua tietoa on suhteellisen helppo analysoida ja käsitellä Motecin omalla i2 Pro -analysointityökalulla. Tallennetusta datasta on hyvä analysoida esimerkiksi virtarajoittimen toimintaa, mikä helpottaa virtarajoittimen säätämistä. (Kuva 17.)



KUVA 17. Näkymä i2 Pro -analysointityökalusta

6 SÄHKÖTYÖKONEEN LOGIIKAN OHJELMOINTI

6.1 Ohjelmointityökalut

PS 501 Control Builder Plus

Laitteiston määrittäminen ja konfigurointi tehtiin Control Builderilla. Ohjelma perustuu 3S- Smart Software Solutions -yhtiön tekemään CoDeSys-ympäristöön.

CoDeSys

CoDeSys on 3S- Smart Software Solutions -yhtiön tekemä laitteistoriippumaton ohjelma, jolla ohjelmoidaan logiikoita. Ohjelmalla voidaan ohjelmoida kuudella aikaisemmin esitetyllä ohjelmointikielellä ja se on yhteensopiva useiden logiikkavalmistajien kanssa. Ohjelma perustuu IEC-61131-3-standardiin. (15.)

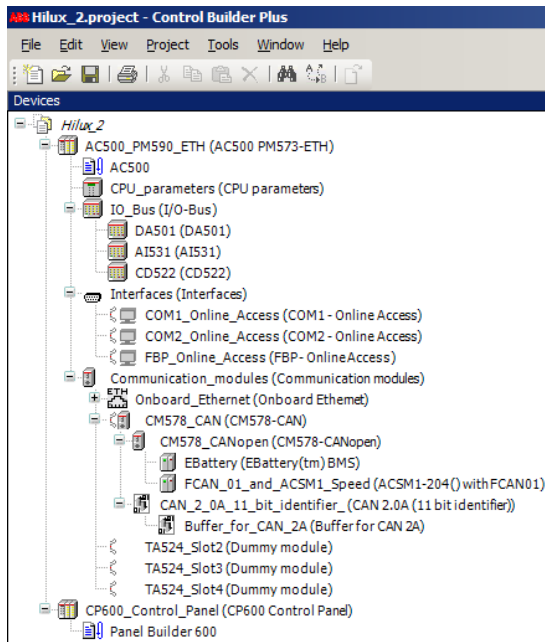
Panel Builder 600

Panel Builder 600 -ohjelmistolla tehtiin käyttöliittymä. Ohjelma on täysin yhteensopiva CoDeSys-ympäristön kanssa, jolloin esimerkiksi tagien (muuttujien) yhdistäminen käyttöliittymään käy vaivattomasti.

Lisäksi käytettiin Matlab -ohjelmistoympäristöä muun muassa PI-säätimien viritysarvojen laskentaan sekä simulointiin. Motec i2 Pro -ohjelmistolla analysoitiin kerättyä dataa.

6.2 Laitteiston määrittäminen ja konfigurointi

Aloitettaessa uutta työtä pitää Control Builder -ohjelmistolla luoda uusi projekti sekä määrittää projektissa käytetty laitteisto. Laitteiston määrittäminen on tehty suhteellisen yksinkertaiseksi. Yleensä projektissa käytettävien laitteiden mukana tulee sähköiset ajurit (EDS, Electronic Data Sheet), joiden avulla laite lisätään laitehierarkiaan. Ohjelman kirjastosta löytyy myös valmiina ajureita useille prosessoreille ja I/O-moduuleille. Kuvassa 18 näkyy projektin laitehierarkia.



KUVA 18. Sähkötyökoneen logiikan laitehierarkia

Laitteiden lisäämisen jälkeen tulee niille tehdä tarvittavat konfiguroinnit. I/O-moduuleille pitää antaa osoitteille muuttujanimet eli ne nimet, joilla niihin viitataan ohjelmaa tehtäessä. Lisäksi pitää määrittää, minkätyyppisinä tuloja ja lähtöjä käytetään sekä minkätyyppistä viestiä viedään esimerkiksi analogiatuloon. Esimerkissä (kuva 19 ja kuva 20) on määritetty DA501-moduulin kanavat ja annettu kanaville nimet (muuttujat).

Parameter	Type	Value	Default Value
Ignore module	Enumeration of BYTE	No	No
Check supply	Enumeration of BYTE	On	On
Input delay	Enumeration of BYTE	1 ms	8 ms
Fast counter	Enumeration of BYTE	0-No counter	0-No counter
Detect short circuit at outputs	Enumeration of BYTE	On	On
Behaviour outputs at comm. error	Enumeration of BYTE	Off	Off
Substitute value	BYTE(0..255)	0	0
Input 0, channel configuration	Enumeration of BYTE	Pt100 (2-wire) -50...+400 °C	Not used
Input 0, check channel	Enumeration of BYTE	Not used	Plausib, Cut wire, Short circuit
Input 1, channel configuration	Enumeration of BYTE	Pt100 (2-wire) -50...+400 °C	Not used
Input 1, check channel	Enumeration of BYTE	Not used	Plausib, Cut wire, Short circuit
Input 2, channel configuration	Enumeration of BYTE	Pt100 (2-wire) -50...+400 °C	Not used
Input 2, check channel	Enumeration of BYTE	Not used	Plausib, Cut wire, Short circuit
Input 3, channel configuration	Enumeration of BYTE	Not used	Not used
Input 3, check channel	Enumeration of BYTE	Plausib, Cut wire, Short circuit	Plausib, Cut wire, Short circuit
Output 0, channel configuration	Enumeration of BYTE	Not used	Not used
Output 0, check channel	Enumeration of BYTE	Plausib, Cut wire, Short circuit	Plausib, Cut wire, Short circuit
Output 0, substitute value	WORD(0..65535)	0	0
Output 1, channel configuration	Enumeration of BYTE	Not used	Not used
Output 1, check channel	Enumeration of BYTE	Plausib, Cut wire, Short circuit	Plausib, Cut wire, Short circuit
Output 1, substitute value	WORD(0..65535)	0	0

KUVA 19. DA501 I/O -moduulin kanavien konfigurointi

Kuten kuvasta 19 näkyy, tuloihin 0, 1 ja 2 on kytketty PT100-anturit kaksijohdin-kytkennällä sekä niiden määritetty mittaskaalaksi -50+400 celsiusastetta. Sa-

malla määritetään, käytetäänkö kanavan vikadiagnostiikkaa. Vikadiagnostiikka kertoo esimerkiksi oikosulusta kanavan napojen välillä.

		Inputs 0-15	%IW0	WORD	Digital inputs 0-15
Handbrake		Input 0	%DX0.0	BOOL	Input 0
Brake_pedal_switch		Input 1	%DX0.1	BOOL	Input 1
Start_key		Input 2	%DX0.2	BOOL	Input 2
Door_switch		Input 3	%DX0.3	BOOL	Input 3
Emergency_switch		Input 4	%DX0.4	BOOL	Input 4
		Input 5	%DX0.5	BOOL	Input 5
		Input 6	%DX0.6	BOOL	Input 6
		Input 7	%DX0.7	BOOL	Input 7
		Input 8	%DX1.0	BOOL	Input 8
		Input 9	%DX1.1	BOOL	Input 9
		Input 10	%DX1.2	BOOL	Input 10
		Input 11	%DX1.3	BOOL	Input 11
		Input 12	%DX1.4	BOOL	Input 12
		Input 13	%DX1.5	BOOL	Input 13
		Input 14	%DX1.6	BOOL	Input 14
		Input 15	%DX1.7	BOOL	Input 15
Motor_temp_1		Analog input 0	%IW1	INT	Analog input 0
Motor_temp_2		Analog input 1	%IW2	INT	Analog input 1
Motor_temp_3		Analog input 2	%IW3	INT	Analog input 2
		Analog input 3	%IW4	INT	Analog input 3
		Analog output 0	%QW0	INT	Analog output 0
		Analog output 1	%QW1	INT	Analog output 1
		Inputs 16-23	%IW5	WORD	Digital In/Outputs - Inputs 16-23
		Outputs 16-23	%QW3	WORD	Digital In/Outputs - Outputs 16-23
Coolant_fan		Output 16	%QX6.0	BOOL	Output 16
Reverse_light		Output 17	%QX6.1	BOOL	Output 17
Battery_heater		Output 18	%QX6.2	BOOL	Output 18
Door_alarm		Output 19	%QX6.3	BOOL	Output 19
		Output 20	%QX6.4	BOOL	Output 20

KUVA 20. Tulojen ja lähtöjen nimeäminen

Väylään liitettäville laitteille pitää muun muassa valita tunnus (node-id), väylänopeus, laitteen käynnistyssekvenssi sekä muuttujat (TPDO, RPDO), joita kyseiset laitteet lähettävät tai vastaanottavat (kuva 21). Lopuksi muuttujat on nimettävä (kuva 22), jotta niitä voidaan kutsua ohjelmassa.

Select receive PDO (RPDO)				Select send PDO (TPDO)			
Name	Index	SubIndex	Bitlength	Name	Index	SubIndex	Bitlength
<input checked="" type="checkbox"/> Receive PDO 1 Parameter	16#1400			<input checked="" type="checkbox"/> Transmit PDO 1 Parameter	16#1800		
Control Word	16#6040	16#00	16	1.01 SPEED ACT [rpm/100]	16#4001	16#01	32
Target torque	16#6071	16#00	16	Status Word	16#6041	16#00	16
<input type="checkbox"/> Receive PDO 6 Parameter	16#1405			8.01 ACTIVE FAULT [-]	16#4008	16#01	16
Control Word	16#6040	16#00	16	<input checked="" type="checkbox"/> Transmit PDO 6 Parameter	16#1805		
<input type="checkbox"/> Receive PDO 21 Parameter	16#1414			Torque actual value	16#6077	16#00	16
				1.15 TEMP INVERTER [C]	16#4001	16#0F	16
				01.17 EXT TEMP [C]	16#4001	16#11	16
				<input checked="" type="checkbox"/> Transmit PDO 21 Parameter	16#1814		
				01.22 INVERTER POWER [kW]	16#4001	16#16	32
				01.07 DC-VOLTAGE [V]	16#4001	16#07	32

KUVA 21. Taajuusmuuttajan muuttujat

Variable	Mapping	Channel	Address	Type	Unit	Description
CW		Control Word	%QW1.4	UINT		
Torque_ref		Target torque	%QW1.5	INT		
Speed_act		1.01 SPEED ACT [rpm/100]	%ID1.10	UDINT		
SW		Status Word	%IW1.22	UINT		
Active_fault		8.01 ACTIVE FAULT [-]	%IW1.23	UINT		
Torque_act		Torque actual value	%IW1.24	INT		
Inverter_temp		1.15 TEMP INVERTER [C]	%IW1.25	INT		
Motor_temp		01.17 EXT TEMP [C]	%IW1.26	INT		
Inverter_power		01.22 INVERTER POWER [kW]	%ID1.14	DINT		
DC_volt		01.07 DC-VOLTAGE [V]	%ID1.15	UDINT		

KUVA 22. Muuttujien nimet

6.3 Taajuusmuuttajan parametrusointi

Taajuusmuuttajan parametrit pitää asettaa omaan sovellukseen sopivaksi. Parametrusointi suoritetaan Drive Studio -ohjelmalla. Keskeisimpiä parametrejä ovat

- kenttäväylän asetukset
- taajuusmuuttajan toimintaprofiilin valinta
- rajoitukset ja turvallisuus (toiminta häiriötilanteessa)
- moottorin parametrit.

Esimerkiksi kenttäväylähäiriön yhteydessä taajuusmuuttaja on määritetty pysähtymään vapaasti pyörien. Muita valintoja voi olla esimerkiksi, että taajuusmuuttaja voidaan valita jatkamaan pyörimistä viimeksi voimassa olleella momenttipyyntöllä häiriötilanteessa. Ajoneuvokäytössä tällainen tilanne ei ole kovin toivottavaa, joten oikeat parametrit on valittava huolella.

Taajuusmuuttajan muuttajat (PDO) voidaan määrittää joko Drive Studio -ohjelmalla, tai kuten tässä projektissa, kenttäväylän kautta. Muuttujien määrittäminen tehdään Control Builder -ohjelmassa ja logiikka lähettää määritetyt muuttajat kenttäväylää pitkin taajuusmuuttajalle solmun käynnistyksen yhteydessä. (10.)

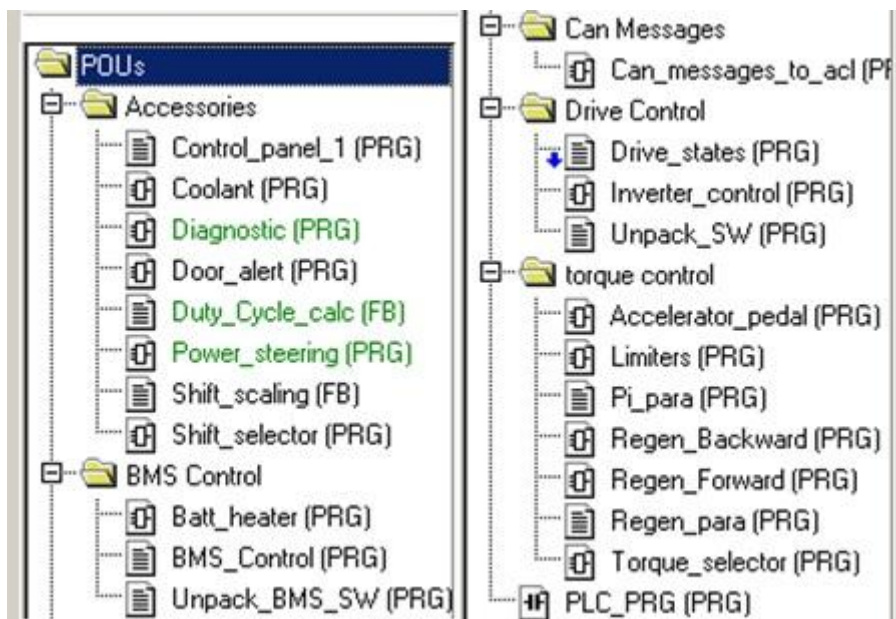
6.4 Akkujenhallintajärjestelmän parametrisointi

BMS:n parametrisointi suoritetaan Hyper Terminal -ohjelmalla. BMS:lle pitää määrittää tunnus (node-id) ja väylänopeus. BMS:n muuttujat (PDO) määritetään Control Builder -ohjelmassa samalla tavalla kuin taajuusmuuttajalle.

6.5 Ohjelmointi

Varsinainen ohjelmointi suoritettiin vaatimusmäärittelyn (liite 1) pohjalta. Valmis ohjelma tosin poikkeaa joltakin osin alkuperäisestä määrittelystä, koska ohjelmaa tehtäessä huomattiin joillekin sovelluksille parempia ratkaisuja kuin alkuperäisesti oli ajateltu.

Ohjelman runko (kuva 23) koostuu useista sovelluksista, joilla kaikilla ohjataan jotain tiettyä osakokonaisuutta sähkötyökoneesta.



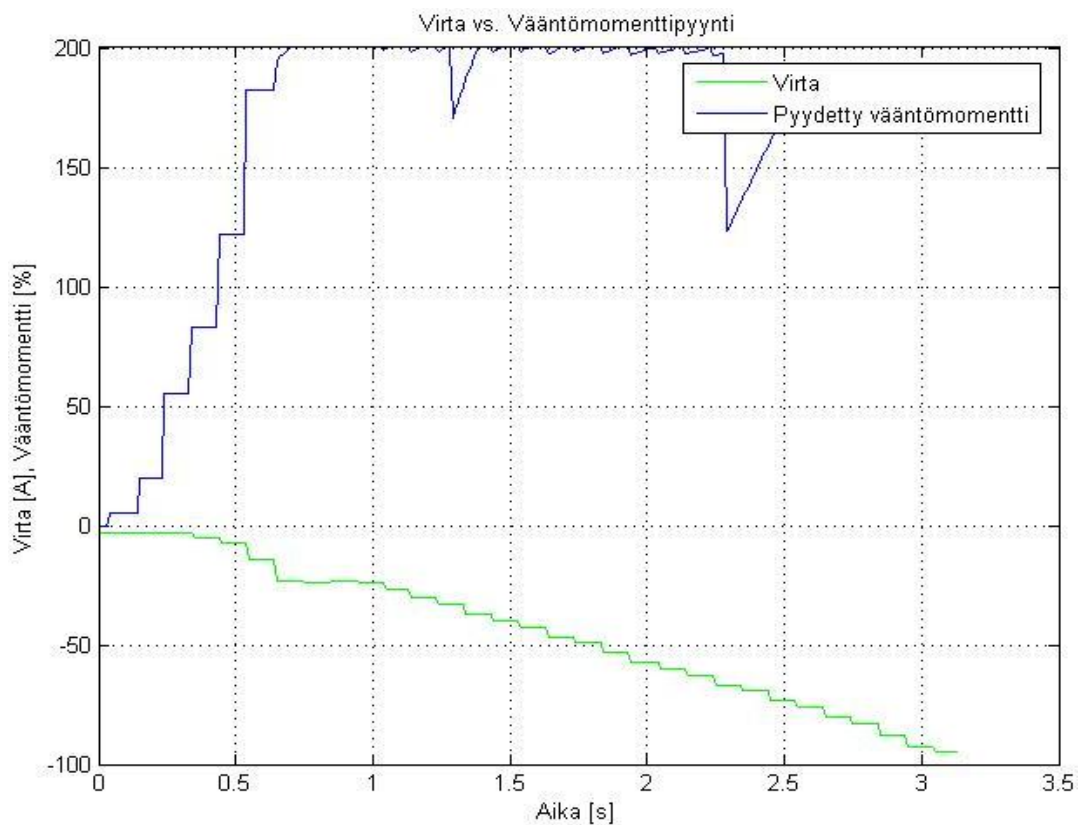
KUVA 23. Logiikkaan ohjelmoidut sovellukset

6.5.1 PI-säätimen viritys

PI-säädintä käytettiin muun muassa akustolta otettavan virran rajoittamiseen. Säätimen hyvä toiminta on edellytys ajoneuvon jouhevalle toiminnalle. Mikäli sallittu virtamäärä ylitetään riittävän kauan, aukaisee BMS automaattisesti pääkontaktorit. Turha kontaktoreiden aukominen kuormitustilanteessa kuluttaa kon-

taktoreiden pintoja, eikä äkillinen ajoneuvon sammuminen liikenteessä ole tarkoituksenmukaista. Säätimen virittäminen oli mielenkiintoista, koska en ollut aikaisemmin suorittanut virittämistä todelliseen prosessiin.

Ajoneuvolla tehtiin kiihdytyksiä useista eri nopeuksista suljetulla radalla sekä tehomittauspenkissä. Kiihdytyksistä saaduista mittaustuloksista (kuva 24) analysoitiin PI-säätimen virityksessä tarvittavia parametrejä sekä laskettiin säätimelle sopivat viritysparametrit. Lopulliset viritysparametrit haettiin kokeilemalla. Lasketut arvot kertoivat kuitenkin parametrien suuruusluokan ja ne olivatkin lähellä lopullisia arvoja. Koska virran käyttäytyminen riippuu hyvin paljon moottorin pyörintänopeudesta, käytetään eri pyörintänopeuksilla eri viritysarvoja.

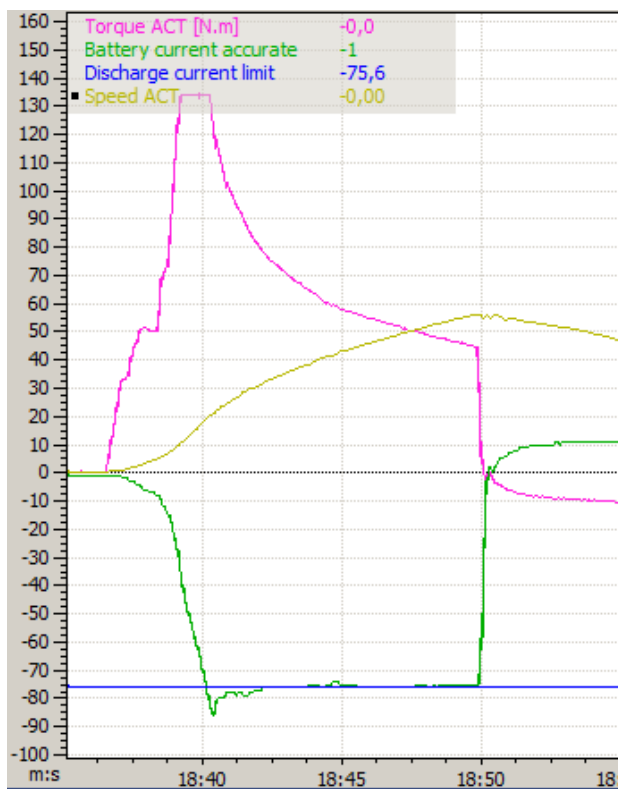


KUVA 24. Virran käyttäytyminen suhteessa vääntömomenttipyyntiin ajan funktiona

BMS laskee erinäisten parametrien pohjalta sallitun maksimikuormitusvirran. Tätä virtarajaa käytettiin ohjelmassa PI-säätimen asetusarvona. Säätimen mitausarvona oli akustolta otettu virta ja ulostulona sallittu maksimivääntömomentti.

Säätimen viritys onnistui kohtuullisen hyvin ja virta pysyy hyvin rajoitettuna oikeaan arvoon (kuva 25) työkonteen ajettavuuden pysyessä miellyttävänä. Kuvassa olevat muuttujat ovat seuraavat:

- vihreä = akuston virta [A]
- lila = toteutunut momentti [%]
- keltainen = auton nopeus [km/h]
- sininen = akuston maksimivirta [A].



KUVA 25. Virtarajoittimen toiminta

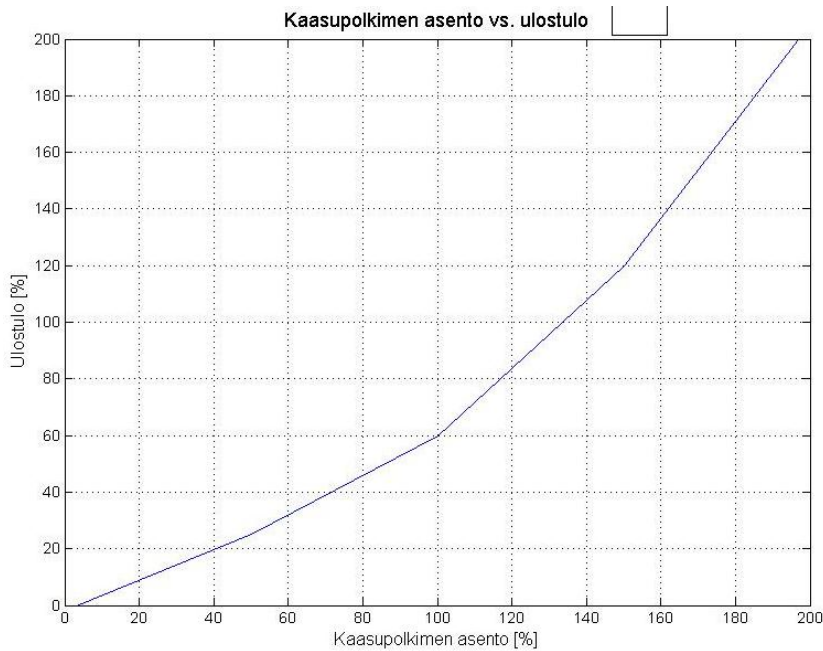
6.5.2 Vääntömomenttipyynti

Yksi mielenkiintoisimmista kokonaisuuksista oli vääntömomenttipyyntiin kokonaisuuden hallitseminen. Kaasupolkimen asentotieto tulee kaksiosaiselta potentiometriltä jännitetietona (0–5 V) logiikkaan. Asentotieto tuodaan kahdennettuna käytännöllisyyden ja turvallisuuden takia. Anturin toiminta-alue on noin 0,25 voltin ja 4,75 voltin välillä. Mikäli anturin johtimet menevät jostakin syystä oikosulkuun tai poikki, tiedetään että anturissa tai johtimissa on vikaa ja sovelluksessa

jätetään viallisen anturin tieto huomioimatta. Näin estetään esimerkiksi viasta johtuva kaasun pohjaan hirttäytyminen.

Moottorin pyörimissuunta riippuu momenttipyyntistä. Positiivinen pyynti pyörittää moottoria eteenpäin ja negatiivinen pyynti takaisinpäin. Suunnanvaihtimella valitaan vaihde joko eteen tai taakse. Suunnanvaihtimen tilan mukaan ohjataan negatiivista tai positiivista momenttipyyntiä taajuusmuuttajalle. Suunnanvaihto on rajoitettu nopeuden mukaan, jottei vahingossa anneta vastakkaista momenttipyyntiä. Mikäli nopeus on positiivinen ja annettaisiin negatiivista momenttipyyntiä, moottori jarruttaa ja tämä voi aiheuttaa talvella renkaiden lukkiutumisen ja työkoneen hallinnan menettämisen.

Analogiatulot näkyvät ohjelmointiympäristössä digitaalilukuina, ja siksi ne täytyy skaalata halutulla tavalla. Taajuusmuuttajan parametreissa on määritetty, että maksimimomenttipyynti on 200 %. Täten kaasupolkimen asentotieto skaalattiin välille 0–200 %. Mikäli maksimi- tai minimimomenttipyyntiä muutetaan taajuusmuuttajaan, joudutaan logiikkaan vastaavasti muuttamaan samat arvot. Ohjelmointi on tehty siten, että käytetään apumuuttujaa `Max_torque_ref`, jolloin riittää pelkästään tämän arvon muuttaminen eikä tarvitse manuaalisesti käydä joka toimilohkoon muuttamassa maksimiarvoa. Asentotiedon lineaarisuudesta haluttiin eroon, joten anturitiedon käyrän vastetta muutettiin hieman toimilohkon avulla, jotta kaasupolkimen käyttö olisi jouhevampaa (kuva 26). Arvot ovat prosentteina, koska taajuusmuuttajan momenttipyynti on prosenttia moottorin nimellismomentista (169 Nm).



KUVA 26. Kaasupolkimen ulostulo asentotiedon funktiona

Kaasupolkimen asentotietoa ei sellaisenaan voi viedä taajuusmuuttajalle momenttipyyntinä, koska pitää ottaa huomioon erilaisia rajoittavia tekijöitä. Sellaisia ovat muun muassa

- akustolta otettava virta sekä akuston lämpötila
- moottorin ja taajuusmuuttajan lämpötila
- yksittäisten akkusolujen jännitetaso.

Näiden rajoittavien tekijöiden perusteella on luotu erilaisia rajoittimia. Rajoittimien ulostuloja ja kuljettajan momenttipyyntiä vertaillaan keskenään ja näistä sillä hetkellä pienin arvo menee momenttipyyntinä taajuusmuuttajalle. Näin estetään kuljettajaa rikkomasta tahattomasti laitteistoa esimerkiksi lämpenemisen yhteydessä.

Ajoneuvon sulavan toiminnan kannalta akustolta otettava virta on keskeisessä roolissa. Taajuusmuuttajan tehon kaava on

$$P = 2 * \pi * n * M \Rightarrow U * I = 2 * \pi * n * M , \quad \text{KAAVA 1.}$$

jossa U on taajuusmuuttajan välipiirin jännite [V], I on akustolta otettava virta [A], n on moottorin kierrosnopeus [1/s] ja M vääntömomentti [Nm]. Sähkötyöko-

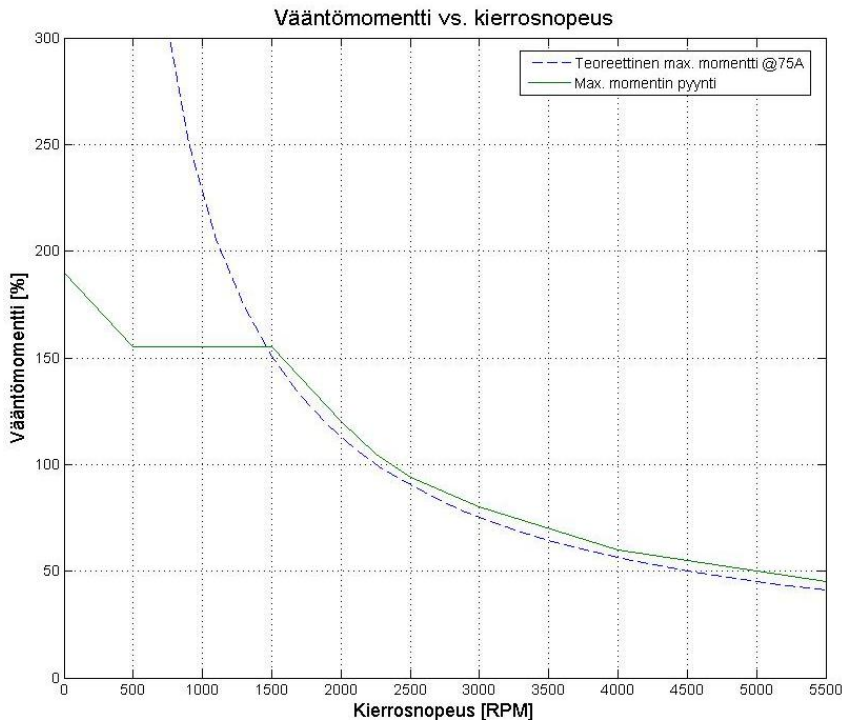
neen BMS kertoo parametrilla Discharge_current_limit akuston maksimi kuormitusvirran. Kerätyn datan perusteella tiedetään, että virtaraja on noin 75 A, ja lisäksi tiedetään, että akuston (taajuusmuuttajan välipiirin) jännite ilman kuormitusta on noin 550 V.

Kaavasta 1 johtamalla saadaan kaava 2, jolla voidaan laskea eri kierrosnopeudella maksimi vääntömomenttipyynti ilman, että 75 ampeerin virtaa ylitetään.

$$M = \frac{U \cdot I}{2 \cdot \pi \cdot n},$$

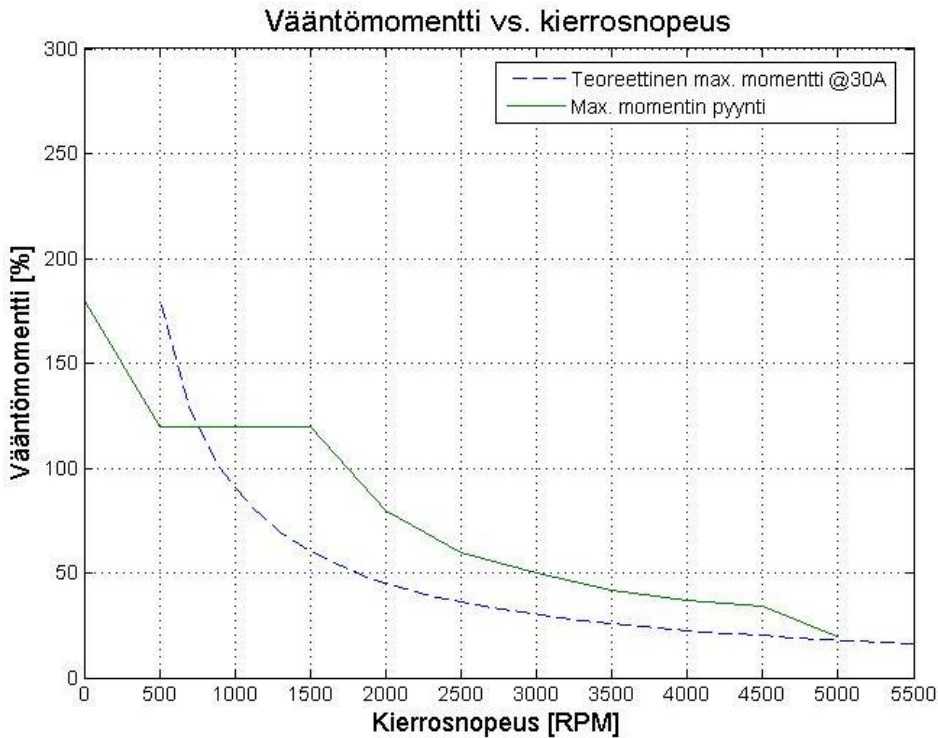
KAAVA 2.

Koska taajuusmuuttajassa on häviötä sekä välipiirin jännite laskee kuormittaessa, ei tällä saada täysin tarkkaa arvoa. Ideana onkin saada karkea arvio maksimivääntömomentista kullakin kierrosnopeudella, jolloin PI-säädin huolehtii hienosäädön. Mikäli tällaista esiasetusta (kuva 27) ei olisi, PI-säädin ei pystyisi millään pitämään virtaa rajoitettuna oikeaan tasoon työkoneen ajettavuuden kärsimättä. Jos säädin viritetään todella nopeaksi, se aiheuttaa epämukavaa nykimistä ajettaessa. Liian hidas säädin taas tekee ajosta jouhevaa, mutta virta kasvaa liian suureksi. Kuvassa momenttipyynti on esitetty prosentteina moottorin nimellismomentista. Lisäksi on huomioitava, että vaikka teoriassa matalilla kierroksilla voitaisiin ottaa moottorista korkeita vääntömomentteja, todellisuudessa taajuusmuuttajan lähtövirran rajoitus rajoittaa vääntömomentin noin 130–140 prosenttiin moottorin nimellismomentista. Maksimilähtövirta taajuusmuuttajalla on 150 ampeeria. Tämän takia momentin pyynti on rajoitettu matalilla kierroksilla alle 200 prosenttiin.



KUVA 27. Akun varaustila yli 20 %. Momentin rajoitus kierrosnopeuden funktiona.

Akuston lataustason laskiessa alkaa myös maksimikuormitusvirta laskea. Täten on tehtävä oma momentinrajoitus (kuva 28) alhaiselle varaustilalle. Käytännössä tämä taso saavutetaan lataustason ollessa 20 %, jonka jälkeen virta laskee lähes lineaarisesti 75 A:sta ja on 5 %:n lataustasolla noin 30 A. Koska sallitussa kuormitusvirrassa on noin paljon eroa alle 20 %:n varaustilalla, on alle 20 %:n varaustilan momentinrajoittimeen jätetty enemmän pelivaraa, jotta vääntömomenttia ei turhaan rajoiteta sallitun kuormitusvirran ollessa esimerkiksi 70 A:n tasolla.



KUVA 28. akun varaustila alle 20 %. Momentin rajoitus kierrosnopeuden funktiona.

6.5.3 Jarruenergian talteenotto

Jarrutettaessa auton omilla jarruilla liike-energia muuttuu jarrupaloissa ja levyissä lämmöksi. Kaikki liike-energia menee hukkaan eikä sitä voida hyödyntää millään tavalla. Sähkötyökoneessa on vaihtovirralla toimiva kestopäinetoimiva tahtikone, jota voidaan käyttää myös generaattorina. Halutussa tilanteessa moottoria käytetään generaattorina, jolloin virtaa syötetään akustolle eli akkuja ladataan. Tällöin moottori ottaa energiansa akselilta, toisin sanoen käyttää sähkötyökoneen liike-energiaa, jonka moottori muuttaa sähköenergiaksi. Tämä energia syötetään taajuusmuuttajan kautta takaisin akustolle. Mikäli moottorin (taajuusmuuttajan) nopeustieto on positiivinen, tällöin annettaessa negatiivinen momenttipyynti taajuusmuuttajalle moottori alkaa toimia generaattorina ja lataa akkuja. Järjestelmä toimii myös ajoneuvolla peruutettaessa, eli nopeustiedon ollessa negatiivinen annetaan positiivinen momenttipyynti.

Sähkötyökoneessa jarrutusenergian talteenotto on toteutettu siten, että kuljettajan nostettua jalka kaasupolkimelta aloitetaan mieto jarrutusenergian talteenot-

to. Tarkoitus on mallintaa normaalin polttomoottorin moottorijarrutusta. Kuljettajan painaessa jarrupoljinta aloitetaan energian maksimaalinen talteenotto, jolloin jarrutusteho voimistuu. Maksimilatausvirta on kuitenkin kohtuullinen, jolloin jarrutustehokaan ei pääse muodostumaan äärettömän suureksi.

BMS:lle täytyy kertoa, että jarrutusenergian talteenotto otetaan käyttöön. Tämä tapahtuu asettamalla tietyt BMS:n parametrit TRUE-tilaan. BMS puolestaan kertoo tietyillä parametreillä, voiko akusto ottaa energiaa vastaan. Näitä parametrejä on ohjelmallisesti valvottava, jotta vältytään akkujen lataamiselta silloin, kun se ei ole mahdollista. BMS myös laskee erinäisten parametrien perusteella maksimilatausvirran ja tämä tieto on saatavilla parametrillä Charge Current Requested. Suositeltavaa latausvirtaa sekä suositeltavaa maksimikenojännitettä (3,65 V) ei saa ylittää, joten näitä on logiikan sovelluksen valvottava ja tarvittaessa vähennettävä momenttipyyntiä. (16.)

Jännite- ja virtarajoja valvotaan logiikan ohjelmassa PI-säätimillä. Esimerkiksi latausvirran rajoittimessa PI-säätimen mittaustieto on latausvirta, asetusarvo on BMS:n ilmoittama maksimilatausvirta ja säätimen lähtönä on taajuusmuuttajalle menevä maksimimomenttipyynti. Lähtö ei siis suoraan ole momenttipyynti taajuusmuuttajalle vaan toimii rajoittimena. Säätimen lähtö on rajoitettu sopivaan arvoon, ettei momenttipyynti kasva hitailla pyörintänopeuksilla liian suureksi.

6.5.4 Tiedonkeruu ja CAN-viestien lähettäminen väylään

Tiedonkeruuta varten autoon asennettiin Motecin ACL-tiedonkeruulaitteisto. Koska väylässä liikkuva tieto ei ole salattua eikä viestiä ole varsinaisesti osoitettu millekään solmulle, voidaan väylässä liikkuva tieto ottaa talteen. Tätä varten pitää tietää kyseisen viestikohdan kommunikointi-id (COB-ID) sekä se, missä kohdin datakenttää talteen otettava data sijaitsee. Nämä tiedot pitää syöttää tiedonkeruulaitteistoon ACL manager -ohjelmistolla. Väylässä olevien laitteiden välisen tiedonsiirron talteen ottaminen ei siis vaatinut logiikan ohjelmalta mitään.

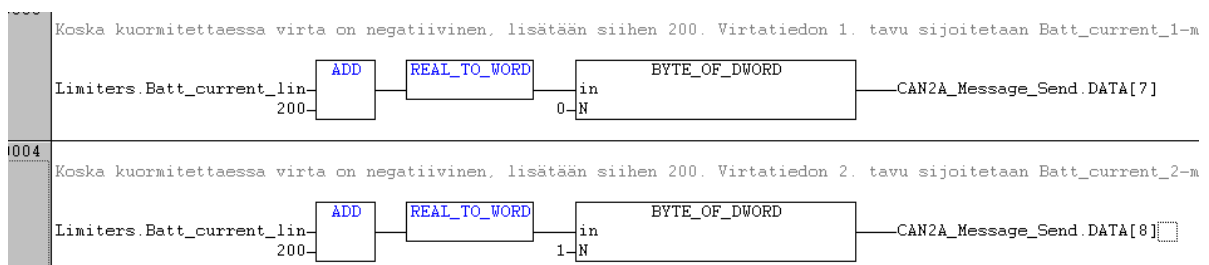
Talteen haluttiin kerätä myös sellaista dataa, joka ei valmiiksi liikkunut väylän kautta. Tätä varten piti logiikkaan ohjelmoida sovellus, joka lähettää halutut tiedot väylän kautta tiedonkeruulle. Tietojen lähettämiseen käytettiin CAN 2.0A

-protokollan viestikehystä, joka on jo esitelty luvussa 4.4. Viestien lähettämistä varten CoDeSysissä on oma toimilohko, CAN2A_SEND. Toimilohkoon pitää määrittää muutamia parametrejä, muun muassa COB-ID, sekä tietysti sijoittaa lähetettävä data lohkon omiin muistipaikkoihin.

Alla kerrotussa esimerkissä akustolta tulevaa virtaa mitataan virtapihtianturilla. Anturi on kytketty logiikan I/O-tuloon. Tieto pitää lähettää tiedonkeruulaitteeseen. Akkujenhallintajärjestelmä toki mittaa virtatietoa koko ajan, mutta sen päivitysnopeus on vain 1 Hz, joten se on liian hidas käytettäväksi virran rajoituksessa.

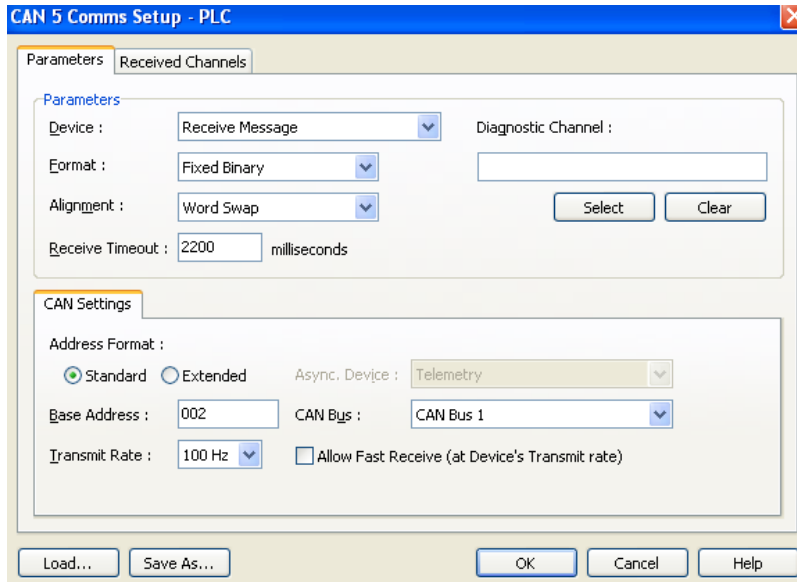
CAN2A_SEND-toimilohkolla voidaan lähettää kahdeksan yhden tavun mittaista tietuetta. Koska tavulla eli 8 bitillä voidaan osoittaa desimaalina korkeintaan 255, ja virta-anturin skaala on välillä –150–350 A, pitää virtatieto lähettää kahdessa tavussa. Nämä kaksi tavua yhdistetään tiedonkeruulaitteessa jälleen yhdeksi tiedoksi.

Virtatieto on skaalattu valmiiksi vastaamaan todellista virtaa toisessa ohjelma-ohkossa. Kuormittaessa virtatieto menee negatiiviseksi ja koska se halutaan tuoda positiivisena väylässä, täytyy siihen lisätä 200. Tämän jälkeen tieto muutetaan reaalityyppisestä tuplasanaksi (DWORD). Tuplasanasta otetaan ensimmäinen tavu ja sijoitetaan se CAN-viestikehyksen seitsemänteen tavuun. Virtatiedon toinen tavu sijoitetaan toimilohkon kahdeksanteen tavuun. Näin virtatieto on siis CAN-viestikehyksen 7. ja 8. tavu (kuva 29).



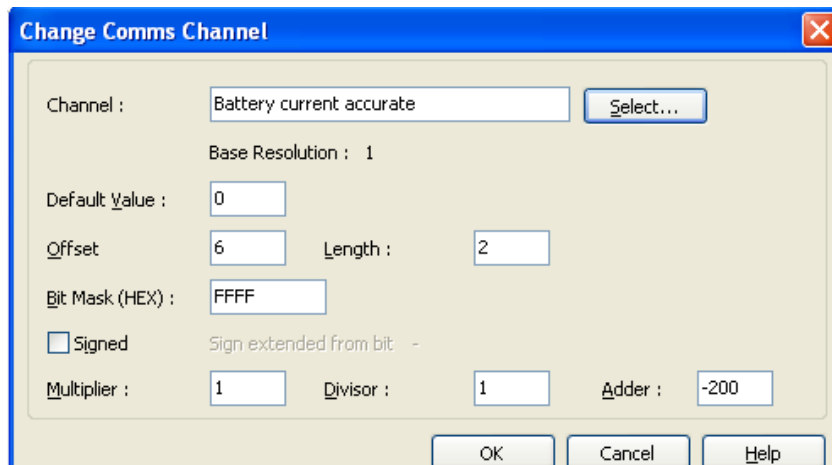
KUVA 29. Virtatiedon käsittely ennen lähettämistä väylään

Tiedonkeruujärjestelmään (Motec) syötetään viestikehyksen osoite (COB-ID) sekä muut asetukset (kuva 30). COB-ID:n on oltava sama, joka annettiin CAN2A_SEND-toimilohkolle.



KUVA 30. CAN-viestikehyksen asetusten määrittäminen Motec ACL -managerilla

Koska yhdessä viestikehyksessä voidaan viedä dataa 0–8 tavun verran, pitää tiedonkeruun asetuksiin määrittää, mitä tietoja tuodaan ja missä kohdin datakenttää ne sijaitsevat (kuva 31).



KUVA 31. Datan paikan ja pituuden määrittäminen viestikehyksessä

Virtatieto tuodaan siis 7. ja 8. tavussa lisättyä 200:lla, joten tiedonkeruun asetuksiin määritetään offsetiksi 6 ja pituudeksi 2. Lukuarvosta vähennetään 200,

joka lisättiin sovelluksessa ennen väylään lähettämistä. Näin saadaan todellinen virtatieto.

6.6 Käyttöliittymä

6.6.1 Suunnittelu

Käyttöliittymää suunniteltaessa on hyvä tietää ihmisen luontainen käyttäytymisen kuva ja tekstiä katsottaessa. Kuvaa katsotaan ylhäältä alaspäin, ja tekstin katsominen ja lukeminen aloitetaan luonnollisesti vasemmasta yläkulmasta. Tietyt asiat ärsyttävät silmiä ja häiritsevät käyttäjää, joten niitä on syytä välttää. Yhdelle näytölle ei tule laittaa liian paljon informaatiota, vaan käyttöliittymät kannattaa rakentaa selkeiksi. Myös käyttöliittymän taustakuva tulee valita siten, että se ei ole liian sekava. (17.)

Väreillä saadaan luotua mielikuvia ja viestittyä erilaisia asioita. Esimerkiksi jonkin laitteen normaalitoimintatila voidaan osoittaa vihreällä värillä ja epänormaali toimintatila punaisella värillä. Tämä toimii muuten hyvin, mutta punavihervärisokeuden yleisyyden takia väriyhdistelmää on syytä välttää. Väriä valittaessa on kuitenkin tärkeää muistaa taustaväriin vaikutus valittavaan väriin. Värien käytössä on hyvä muistaa käyttöliittymän selkeys. Suositeltavaa onkin käyttää värejä siten, että yksi väri tarkoittaa aina samaa asiaa ja että käyttöliittymässä käytetään vain muutamia eri värejä. (17.)

6.6.2 Toteutus

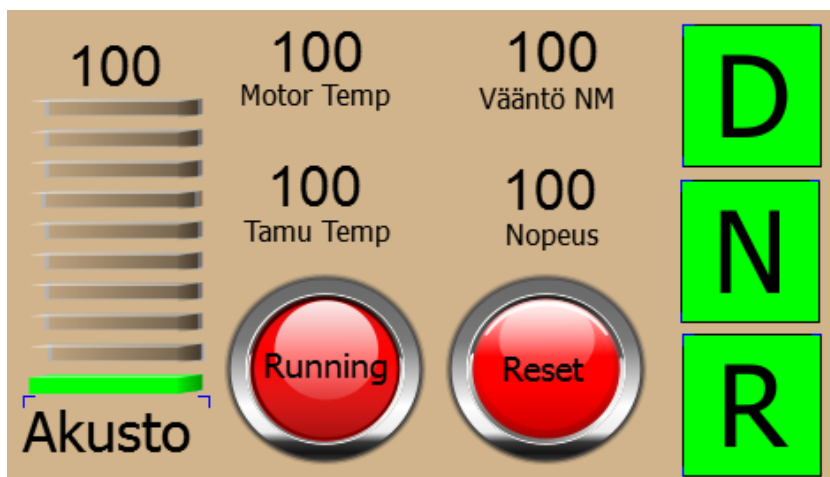
Käyttöliittymä toteutus on tehty vaatimusmäärittelyn (liite 1) pohjalta. Ohjauspaneelista ei ole tarkoitus ohjata auton toimintaa, vaan sen tehtävänä on kertoa kuljettajalle tärkeimmät tiedot autosta. Myöskään tästä syystä käyttöliittymän tekoon ei ole uhrattu loputtomasti aikaa, vaan on pyritty saamaan aikaiseksi toimiva järjestelmä. Käyttöliittymä tehdään Panel Builder 600 -ohjelmalla. Ennen kuin varsinaista käyttöliittymää aletaan tehdä, pitää määrittää näyttötyyppi, kommunikointiprofiili ja sen asetukset. Asetusten lisäksi CoDeSysistä tuodaan tagit Panel Builderiin. Tagit ovat kaikkia niitä sisäisiä ja ulkoisia muuttujia, joita ohjelmassa käytetään. Vaikkapa jonkun PI-säätimen mittaustieto voidaan tuoda

tagina Panel Builderiin. Tämä tieto voidaan näyttää esimerkiksi numerotietona ohjauspaneelin näytöllä.

Ohjauspaneelista (kuva 32) pyrittiin saamaan mahdollisimman selkeä, jotta sitä on helppo ajon aikana seurata. Testikäyttöä varten näyttöön laitettiin muutama mittaustieto näkyviin, jotta moottorin toimintaa on helpompi seurata esimerkiksi tehonmittauspenkissä ajettaessa. Näytössä on tällä hetkellä näkyvissä seuraavat tiedot:

- akun varaustila palkki- ja numerotietona
- suunnanvaihtimen tila (D, N, R)
- moottorin lämpötila
- taajuusmuuttajan lämpötila
- moottorin nopeus
- moottorin vääntömomentti
- taajuusmuuttajan moduloinnin indikointi.

Taajuusmuuttajan viat voidaan kuitata Reset-painikkeella.



KUVA 32. Käyttöpaneeli

Käyttöpaneelista jätettiin vaatimusmäärittelystä poiketen poissa taajuusmuuttajan ja akkujenhallintajärjestelmän vikakoodien näyttäminen sekä kesä- ja talviasetuksien valitseminen. Tiettyjä asioita ei ehditty testata riittävän erilaisissa talviolosuhteissa, joten omia asetuksia kesälle ja talvelle ei vielä tässä vaiheessa tehty.

7 POHDINTA

Kun syksyllä 2012 minulle tuli mahdollisuus aloittaa tällaisen opinnäytetyön tekeminen, mietin jonkin aikaa, alanko työtä tekemään. Näin jälkikäteen voi todeta, että on ollut erittäin opettavainen syksy ja kevät sähkötyökoneen parissa. Työtunteja ei ole laskettu ja paljon on tullut tehtyä sellaista, mikä ei suoranaisesti omaan opinnäytetyöhön liity. Projekti on ollut kuitenkin sen laatuinen, ettei jokaista työvaihetta ole voinut miettiä sen kautta, kenelle se kuuluu. Muuten olisi tämä sähkötyökone jäänyt valmistumatta. Työn haasteellisuutta lisäsi alkuvaiheessa se, että laitteisto ja ohjelmistoympäristö oli minulle ennestään täysin tuntematon. Ohjelmassa oli kuitenkin hyvät ohjeet sen toiminnasta sekä sain ABB:ltä hyvää teknistä tukea vikatilanteissa.

Kokonaisuutena oma osuuteni, eli logiikan ohjelmointi, onnistui hyvin. Logiikan sovellus toimii odotusten mukaisesti ja sähkötyökoneella on päästy ajamaan testiajoa keväällä 2013. Tiedonkeruulaitteiston lisääminen järjestelmään aiheutti aluksi hieman haasteita, koska laitteen liittämistä tällaiseen järjestelmään ei ole olemassa yksiselitteisiä ohjeita. Tietoja järjestelmistä joutui keräilemään sieltä täältä ja päättellemään itse, miten tiedonkeruujärjestelmän saisi järjestelmään kytkettyä. Tätä ongelmaa ratkottiin yhdessä Miika Kajulan kanssa, joka suoritti myös oman opinnäytetyönsä sähkötyökoneeseen, aiheenaan moottorinohjausjärjestelmän suunnittelu ja toteutus.

Sovelluksessa on toki myös parannuskohteita. Momenttipyyntiin kokonaisuutta voi parannella lisää esimerkiksi siten, että maksimivääntömomentilla pitäisi laittaa useampi portaita suhteessa akuston varaustasoon, jolloin sovelluksen mahdollinen maksimivääntömomenttiraja olisi hyvin lähellä teoreettista maksimia. Akuston varaustason laskeminenhan aiheutti akuston maksimikuormitusvirran laskun. Lisäksi kaasupolkimen vasteesta voisi tehdä 3D-kartan, jolloin kaasupolkimen ulostuloon vaikuttaisivat sekä kierrosnopeus että polkimen asentotieto. Näin saataisiin korkeilla kierroksilla polkimesta järkevämpi vaste.

LÄHTEET

1. Ajoneuvon vaihtoehtoiset polttoaineet. 2013. Saatavissa:
http://fi.wikipedia.org/wiki/Ajoneuvon_vaihtoehtoiset_polttoaineet. Hakupäivä 1.4.2013
2. Fonselius, Jaakko – Pekkola, Kari – Selosmaa, Seppo – Ström, Markku – Välimaa, Taisto. 1996. Automaatiolaitteet. Helsinki: Oy Edita Ab.
3. New scalable PLC AC 500. Saatavissa:
http://www.industrialpartners.eu/uploads/tx_ipprojects/ctl110104_ABB_A_C500_PLC_1_01.jpg. Hakupäivä 1.4.2013
4. 3S-Smart Software Solutions, CoDeSys Programming System. Saatavissa:
http://extern.codesys.com/index.shtml?en_oem1. Hakupäivä 10.03.2013.
5. Harju, Timo – Marttinen, Arto. 2000. Säättöpiirin virityksen perusteet. Espoo: Control CAD Oy.
6. Tervaskanto, Manne. 2011. PID-säädön perusteet. Saatavissa:
http://cc.oulu.fi/~ylikoant/PID/PID_saadon_perusteet_2011.pdf. Hakupäivä 5.2.2013
7. Saha, Heikki 2005. CAN-väylä. Saatavissa:
<http://www.canopen.fi/artikkelit/CAN.pdf>. Hakupäivä 20.11.2012
8. Corrigan, Steve. 2008. Controller Area Network Physical Layer Requirements. Saatavissa: <http://www.ti.com/lit/an/slla270/slla270.pdf>. Hakupäivä 15.1.2013
9. Saha, Heikki. 2006. CANopen perusteet. Saatavissa:
<http://www.canopen.fi/artikkelit/CANopen.pdf>. Hakupäivä 20.11.2012
10. Fieldbus Control with FCAN-01 CANopen Adapter Module and ABB AC500 PLC. 2011. Saatavissa:
<http://www05.abb.com/global/scot/scot201.nsf/veritydisplay/c0bf8ec2cc9>

- [55cdfc12576f80039bced/\\$file/en_acsm1_fb_ctrl%20fcan01_appl_guide_a.pdf](#). Hakupäivä 20.11.2012.
11. Hilux-sähkötyökoneen moottorin tyyppikilpi. 2013. Randax Oy.
12. ACSM1 laiteopas. Saatavissa:
[http://www05.abb.com/global/scot/scot201.nsf/veritydisplay/ec6b1bf5cfd4458dc1257379002b7848/\\$file/FI_ACSM1_04_HW_C_screenres.pdf](http://www05.abb.com/global/scot/scot201.nsf/veritydisplay/ec6b1bf5cfd4458dc1257379002b7848/$file/FI_ACSM1_04_HW_C_screenres.pdf).
Hakupäivä 20.11.2012
13. AC500 Control Builder PS501 - Complete English Documentation V2.2.0. 2012. Saatavissa: [http://search-ext.abb.com/LibraryDownloadManager/Default.aspx?resource=http://www05.abb.com/global/scot/scot209.nsf/veritydisplay/2e32586cc621067ac1257a990024f707/\\$file/3ADR025078M0201.zip](http://search-ext.abb.com/LibraryDownloadManager/Default.aspx?resource=http://www05.abb.com/global/scot/scot209.nsf/veritydisplay/2e32586cc621067ac1257a990024f707/$file/3ADR025078M0201.zip). Hakupäivä 21.11.2012
14. CP600 and CP400 series. 2013. Saatavissa:
<http://abbib.cloudapp.net/public/v2/e9dce309-49c3-430c-864c-e682bc40a116/presentation.jpg>. Hakupäivä 1.4.2013
15. CODESYS - The Complete Software Suite for Automation Technology. 2013. Saatavissa: <http://www.codesys.com/the-system.html>. Hakupäivä 10.03.2013
16. European Batteries Oy. 2012. BMS 1.2 CANopen interface description for I/O device profile 401. Sisäinen dokumentti.
17. Metsämäki, Markku. 1995. Näytön graafinen suunnittelu. Helsinki: Oy Edita Ab (Painatuskeskus Oy)

Hilux-sähkötyökoneprojektin moottorinohjaus

Laitteet:

- ABB ACSM1 taajuusmuuttaja FCAN-01 CAN-moduulilla
 - Ohjaa Randaxin kestopagneettitahtikonetta
- ABB AC500-XC PLC
 - DA501 I/O -moduuli
 - AI531 I/O -moduuli
 - CD522 Encoder & PWM -moduuli
 - CM578 CAN-moduuli
- ABB CP620 4" kosketusnäyttö
- MoTec ACL tiedonkeruulaitteisto
- European Batteries BMS (Battery Management System)

Sisääntulot:

- 5 kpl PT100 lämpötila-anturia
 - 3x Moottorin lämpötila
 - Jäähdytysnesteen lämpötila
 - Ohjaamon lämmitysnesteen lämpötila
- Jarruvalokatkaisin
- 12v järjestelmän akkujännite
- Seisontajarru
- Kaasupolkimen asento (Kaksiliukuinen potentiometri)
- 3 kpl suunnanvaihtaja (eteen, taakse, vapaa)

Ulostulot:

- Jäähdyttimen tuuletin
- Ohjaustehostimen pumppu
- Ohjaamon lämmitin

CANopen väylä:

- PLC, taajuusmuuttaja, akkujenhallinta ja tiedonkeruu liitetään samaan väylään
- PLC ohjaa tamua väylällä
- PLC:n tarvitsemat tiedot väylän kautta:
 - Akuston laskennallinen varaustila(BMS)
 - Akuston virheilmoitukset (BMS)
 - Tamun oloarvot ja virheilmoitukset
- Tiedonkeruun tarvitsemat tiedot väylän kautta:
 - Moottorin lämpötilat (PLC)
 - Jäähdytysnesteen lämpötila (PLC)
 - Kaasupolkimen asento (PLC)
 - Tamun lähtöteho moottorille (TAMU)
 - Akkujännite (TAMU)

- Tamun lämpötila (TAMU)

Moottorin ohjaaminen

- Kaasupolkimella pyydetään vääntömomenttia kaasupoljinkäyrän mukaisesti
 - Vääntömomentin rajoitus akuilta otettavan virran mukaan(Max. virta 90A)
 - Vääntömomentin rajoitus akuston varaustilan mukaan
 - Vääntömomentin rajoitus moottorin ja tamun lämpötilan mukaan
 - Vääntömomenttia ei anneta, jos painetaan jarrua kaasun painamisen jälkeen
 - Vääntömomentin rajoitus ennen kuin tietty pyörintänopeus saavutetaan
- Kesä- ja talviasetukset
- Moottorin pyörintäsuunta ei saa vaihtua eikä se saa tehdä äkkipysähdystä, mikäli se pyörii suunnanvaihdinta käytettäessä vastakkaiseen suuntaan
- Seisontajarru kytkettynä vaihde menee automaattisesti vapaalle
- Peruutusvaihteen nopeudenrajoitus

Muut ohjaukset

- Jäähdyttimen tuulettimen ohjaus jäähdytinnesteen lämpötilan mukaan
- Ohjaustehostimen pumpun ohjaus PWM:llä ajonopeuden mukaan
 - Vaihteen oltava valittuna, että pumppu pyörii
- Ohjaamoon polttonestekäyttöisen lämmittimen ohjaus lämmitysneesteen lämpötilan mukaan
 - Minimi käyttöaika säädettävä

Käyttöliittymä/Kuljettajainformaatio

- Normaalitilassa näytössä näkyy:
 - Vaihde D, R, N tai P
 - Akuston laskennallinen varaustila
- Hälytykset(“moottorin häiriövalo”):
 - Tamulta tulevat vikakoodit
 - BMS:itä tulevat vikakoodit
 - Moottorin lämpötila
 - Jäähdytysjärjestelmän lämpötila
 - Anturiviat
 - Alijännite 12v järjestelmässä
- Näytöstä valittavissa kesä ja talviasetukset