Tampereen yliopisto

Joona Jokipii

NOSTOSIIRTOVAUNUN HAJAUTETTU MODULAARINEN OHJAUSJÄRJESTELMÄ

Kandidaatintyö
Tekniikan ja luonnontieteiden tiedekunta
Tarkastaja: Yrjö Majanne
Tarkastaja: Heikki Saha
Tammikuu 2024

TIIVISTELMÄ

Joona Jokipii: Nostosiirtovaunun hajautettu modulaarinen ohjausjärjestelmä
Kandidaatintyö
Tampereen yliopisto
Teknisten tieteiden kandidaatin tutkinto-ohjelma, Automaatiotekniikka
Tammikuu 2024

Tässä työssä tutkitaan nostosiirtovaunun hajautetun modulaarisen ohjausjärjestelmän suunnittelu- sekä toteutusprosessia. Ohjausjärjestelmän toiminnalliset ja ei-toiminnalliset vaatimukset on asetettu projektin asiakkaan toimesta. Järjestelmän ei-toiminnallisiin vaatimuksiin on asetettu vaatimus ohjausjärjestelmän laajennettavuudesta tuoteperheen tuleviin tuotteisiin. Hajautetulla modulaarisella järjestelmäsuunnitelulla edistetään ohjausjärjestelmän uudelleenkäytettävyyttä ja laajennettavuutta. Modulaarisessa järjestelmäsuunnittelussa ohjausjärjestelmän toiminnallisuutta toteuttavat osat jaetaan osakokonaisuuksiin eli moduuleihin. Ohjausjärjestelmän moduulien välinen kommunikaatio on toteutettu CAN-väylän kautta tapahtuvan digitaalisen kommunikaation avulla. Digitaalinen kommunikaatio noudattaa CANopen viestintäjärjestelmän standardien määrittämää rakennetta.

Tässä työssä suunniteltu hajautettu modulaarinen ohjausjärjestelmä on toteutettu nostosiirtovaunun prototyypissä. Ohjausjärjestelmän suunnitteluprosessi noudattaa yleisiä hajautetun modulaarisen järjestelmäsuunnittelun periaatteita. Järjestelmän moduulit on toteutettu noudattamaan tuotemodulaation, vakiorajapintojen sekä konfiguraatiohallinnan periaatteita. Moduuleihin on myös toteutettu CANopen viestintäjärjestelmän standardien mukainen rakenne. Moduulien CANopen viestintäjärjestelmän mukainen rakenne mahdollistaa ohjausjärjestelmän moduulien välisen digitaalisen kommunikaation CAN-väylän kautta.

Työn tulokset osoittavat, että hajautetulla modulaarisella järjestelmäsuunnitelulla edistettiin järjestelmän laajennettavuutta, kehitettävyyttä, ylläpidettävyyttä, kustannustehokuutta, luotettavuutta sekä huollettavuutta. Perinteiseen järjestelmäsuunnitteluun verrattaessa hajautetulla modulaarisella järjestelmäsuunnittelulla saavutetaan useita etuja. Hajautettu modulaarinen järjestelmäsuunnittelu kuitenkin kasvattaa järjestelmän suunnittelu- sekä toteutusprosessin kuluja. Hajautetun modulaarisen ohjausjärjestelmän suuremmat kustannukset säästetään moninkertaisesti tuotteen sekä tuoteperheen elinkaaren aikana vähentyneiden ylläpito- sekä huoltokustannusten muodossa.

Avainsanat: Hajautettu ohjausjärjestelmä, Tuotemodulointi, Nostosiirtovaunu, Controller Area Network (CAN), CANopen

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ALKUSANAT

Oma kiinnostukseni liikkuvien laitteiden järjestelmäsuunnittelua kohtaan on kasvanut opintojeni sekä työkokemusteni kautta. Mielenkiintoisen tieteenalasta tekee suunnittelutyön käytännönläheisyys sekä tieteenalan nopea kehitys. Täten työyhteisöni tarjotessa minulle tutkielmani aihetta *nostosiirtovaunun hajautettu modulaarinen ohjausjärjestelmä,* otin sen mielelläni vastaan. Toteutetun projektin sekä tutkielman kirjoituksen aikana olen oppinut automaatiotekniikan alan monipuolisuudesta sekä akateemisesta tutkimuksesta.

Haluan kiittää työyhteisöäni tarjotusta mahdollisuudesta työskennellä tutkielmani aihetta käsittelevässä projektissa sekä kaikesta tuesta projektin sekä tutkielman toteuttamisessa. Erityisesti haluan kiittää ohjaajaani Heikki Sahaa rakentavasta sekä opettavaisesta yhteistyöstä projektin sekä kirjoitustyön aikana. Haluan myös kiittää kandityöni ohjaajaa Yrjö Majannetta kaikesta tuesta sekä tutkielman toteutuksen aikataulun joustavuudesta.

Tampereella, 17.1.2024

Joona Jokipii

SISÄLLYSLUETTELO

1. JOHDANTO 1
2. OHJAUSJÄRJESTELMÄN LÄHTÖKOHDAT JA VAATIMUKSET 2
2.1 Nostosiirtovaunun rakenne ja toiminnallisuus 2
2.2 Nostosiirtovaunun toimintaympäristö 4
2.3 Ohjausjärjestelmän vaatimukset 5
3. OHJAUSJÄRJESTELMÄSUUNNITTELUN TEOREETTINEN TAUSTA 7
3.1 Tuotemodulointi 7
3.2 Vakiorajapinnat 8
3.3 CANopen 9
3.4 Konfiguraatiohallinta 12
4. OHJAUSJÄRJESTELMÄN TOTEUTUS 13
4.1 Ohjausjärjestelmän rakenne 13
4.2 Moduulikohtaiset toteutukset 14
4.3 Moduulien CANopen integraatio 16
4.4 Moduulien konfiguraatiohallinta 18
5. OHJAUSJÄRJESTELMÄSUUNNITTELUN TULOKSET 19
6. YHTEENVETO JA KESKUSTELU 22
LÄHTEET 23

KUVALUETTELO

Kuva 1. *Nostosiirtovaunu [3]. 2*
Kuva 2. *Kontti- ja lauttavaunu [3]. 3*
Kuva 3. *Kasetti tyyppinen kuorma-alusta [3]. 3*
Kuva 4. *Vetomestari [5]. 4*
Kuva 5. *Nostosiirtovaunun ohjausjärjestelmän käyttötapauskaavio. 5*
Kuva 6. *CANopen laitteen yleinen rakenne. Muokattu kohteesta [12]. 10*
Kuva 7. *Ohjausjärjestelmän rakenne. 13*
Kuva 8. *Nostosiirtovaunun prototyypin toteutus. 19*
Kuva 9. *Moduulien NMT-tilanvaihdot ohjausjärjestelmän käynnistymisessä. 20*

LYHENTEET JA MERKINNÄT

CAN	Controller Area Network – Protokolla
CiA	CAN in Automation – Organisaatio
EDS	Electronic Data Sheet – CANopen tiedostotyyppi
NMT	Network Management – CANopen protokolla
PDO	Process Data Object – CANopen protokolla
PLC	Programmable Logic Controller – Ohjelmoitava logiikka
SDO	Service Data Object – CANopen protokolla
UML	Unified Modeling Language – Mallinnuskieli

1. JOHDANTO

Modulaarista järjestelmäsuunnittelua on jo pitkään hyödynnetty konesuunnittelun parissa. Modulaarisessa järjestelmässä toiminnallisuutta toteuttavat osat on jaettu omiksi osakokonaisuuksikseen eli moduuleiksi [1]. Automaatiojärjestelmien suunnittelussa modulaarista järjestelmäsuunnittelua hyödynnetään aikaisempaa enemmän. Modulaarisessa kone- sekä automaatiojärjestelmäsuunnittelussa hyödynnetään samankaltaisia suunnittelumenetelmiä sekä suunnittelumalleja. Modulaarisesti toteutetussa ohjausjärjestelmässä moduulit sijoitetaan hajautetusti toteutettavan toiminnallisuuden läheisyyteen.

Hajautettua modulaarista järjestelmäsuunnittelua käytetään keinona hallita automaatiojärjestelmien kasvavaa vaatimusten määrää sekä automaatioastetta. Ohjausjärjestelmien suurempi toiminnallisten ja ei-toiminnallisten vaatimusten määrä kasvattaa järjestelmän automaatioastetta sekä kompleksisuutta [2]. Ohjausjärjestelmän konfiguraatioiden tai vaatimusten muuttuessa modulaarisesti suunniteltu järjestelmä pystytään tehokkaasti sovittamaan uusiin vaatimuksiin.

Tässä tutkielmassa käsitellään asiakkaalle toteutetun nostosiirtovaunun prototyyppi version toteutusta. Nostosiirtovaunun toiminnalliset ja ei-toiminnalliset vaatimukset on määritetty tukemaan järjestelmän laajentamista tuoteperheen muiksi tuotteiksi tulevaisuudessa. Toteutettu nostosiirtovaunun prototyyppi sisältää hajautetun modulaarisen ohjausjärjestelmän. Tässä tutkielmassa tutkitaan, kuinka nostosiirtovaunun hajautettu modulaarinen ohjausjärjestelmä edistää järjestelmän hallittavuutta ja laajennettavuutta.

Luvussa 2 käsitellään nostosiirtovaunun yleistä toimintaa sekä ominaisuuksia. Luvussa myös esitellään asiakkaan vaatimusmäärittelystä johdetut ohjausjärjestelmän vaatimukset. Hajautetun modulaarisen ohjausjärjestelmän toteuttamiseen liittyvää teoriaa käsitellään luvussa 3. Luvussa käsitellään myös CANopen viestintäjärjestelmän toimintaa. Luvussa 4 on esitetty toteutetun ohjausjärjestelmän rakenne sekä järjestelmän moduulikohtaiset toteutukset. Työn tulokset sekä yhteenveto on esitetty luvuissa 5 ja 6.

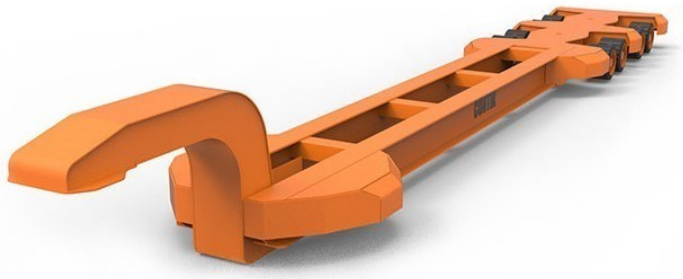
2. OHJAUSJÄRJESTELMÄN LÄHTÖKOHDAT JA VAATIMUKSET

Nostosiirtovaunun hajautettu modulaarinen ohjausjärjestelmä suunnitellaan asiakkaan sekä toimintaympäristön määrittämien toiminnallisten ja ei-toiminnallisten vaatimusten perusteella. Toteutettavaa ohjausjärjestelmää suunniteltaessa on nostosiirtovaunun yleinen toimintamalli sekä toimintatapa ymmärrettävä.

2.1 Nostosiirtovaunun rakenne ja toiminnallisuus

Nostosiirtovaunuja hyödynnetään prosessi- sekä satamateollisuuden sovelluksissa, joissa raskasta sekä suurikokoista materiaalia siirretään sijainnista toiseen. Ero nostosiirtovaunun ja perinteisen konttivaunun välillä nähdään vaunujen rakenteellisessa toteutuksessa. Nostosiirtovaunu sisältää ohjausjärjestelmän, joka on toteutettu ohjelmoitavien logiikoiden, hydrauliikan sekä erilaisten toimilaitteiden avulla. Perinteisiin kontti- sekä lauttavaunuihin ei ole sisällytetty ohjaustoiminnallisuutta mahdollistavaa ohjausjärjestelmää. Ohjausjärjestelmän avulla nostosiirtovaunun ohjauskulmaa sekä nostokorkeutta pystytään ohjaamaan. Ohjattavan nostokorkeuden sekä ohjauskulman ansiosta nostosiirtovaunuja pystyään hyödyntämään materiaalin siirtämiseen erilaisten kuormaalustojen avulla.

Nostosiirtovaunun rakenteellinen toteutus määritetään vaunun toiminnallisten vaatimusten perusteella. Kuva 1 on esitetty yleistetty nostosiirtovaunun toteutus.



Kuva 1. *Nostosiirtovaunu [3].*

Kuljetettavan materiaalin paino määrittää nostosiirtovaunun toteutuksessa käytettävien hydraulisten telien lukumäärän. Kuljetettavan kuorman painon kasvaessa vaunun telien lukumäärä kasvatetaan tarvittavan nosto- sekä kääntötehon varmistamiseksi. Kuva 1

esitetty nostosiirtovaunun toteutus sisältää yhden nosto- sekä kääntötoiminnallisuutta toteuttavan hydraulisen telin.

Nostosiirtovaunun rakenteellinen toteutus määritetään kuljettavan materiaalin ominaisuuksien mukaan. Esimerkiksi käyttötapauksessa, jossa materiaalin kuljettamiseen tarvittava tukipinta-ala on suuri, nostosiirtovaunun rakenteellinen toteutus mukautetaan lauttavaunua vastaavaksi. Kuva 2 on esitetty kontti- sekä lauttavaunun rakenteelliset toteutukset.

Kuva 2. *Kontti- ja lauttavaunu [3].*

Nostosiirtovaunujen toiminta perustuu kuljetettavan materiaalin varastointimenetelmään, jossa materiaali varastoidaan kuorma-alustojen päälle. Kuorma-alustat jaetaan ryhmiin alustojen rakenteellisten toteutusten perusteella. Erilaisia kuorma-alustoja, joita nostosiirtovaunujen kanssa hyödynnetään, ovat esimerkiksi kasetit, pukit sekä muut vastaavat jalkarakenteet. Kuva 3 on esitetty kasetti tyyppinen kuorma-alusta, jonka päälle kuljetettavaa materiaali voidaan varastoida.

Kuva 3. *Kasetti tyyppinen kuorma-alusta [3].*

Varastointimenetelmä mahdollistaa nostosiirtovaunun toimintatavan, jossa vaunu ohjataan siirrettävän kuorman alle, kuorma nostetaan ylös, siirretään haluttuun sijaintiin ja lasketaan alas.

2.2 Nostosiirtovaunun toimintaympäristö

Nostosiirtovaunuja käytetään erilaisissa käyttökohteissa, missä toimintaympäristön olosuhteet ovat usein haasteelliset. Nostosiirtovaunun toimintaympäristön sekä käyttötapausten tarkastelussa huomioidaan myös nostosiirtovaunun kanssa yhteistyössä toimivat järjestelmät sekä toimilaitteet. Nostosiirtovaunun toiminnallisuuden toteuttamiseksi tarvitaan vetomestariksi tai terminaalitraktoriksi kutsuttua ajoneuvoa.

Toimintaympäristön kosteus, pöly sekä lika aiheuttaa ohjausjärjestelmän toiminnassa oikosulkuja sekä toimintahäiriöitä johtaen järjestelmän toimintakyvyn heikkenemiseen [4]. Ohjausjärjestelmän toteutuksen toimintavarmuuden on säilyttävä ympäristön haittavaikutuksista huolimatta. Haitallisten ilmiöiden vaikutusta ohjausjärjestelmän toimintaan pyritään vähentämään toimintaympäristöön soveltuvien kaapelointi sekä komponentti valintojen avulla. Toimintaympäristön olosuhteiden tarkastelussa on myös huomioitava ympäristön lämpötilan sekä maaperän koostumuksen vaihtelut. Kokonaisuutena ohjausjärjestelmä ja sen osakokonaisuudet tulee suunnitella toimimaan nostosiirtovaunun toimintaympäristössä.

Nostosiirtovaunua ohjataan vetomestariksi kutsutulla ajoneuvolla. Vetomestariin on toteutettu nostosiirtovaunun ohjaamisen tarvittava toiminnallisuus. Kuva 4 on esitetty vetomestarin yleistetty toteutus.

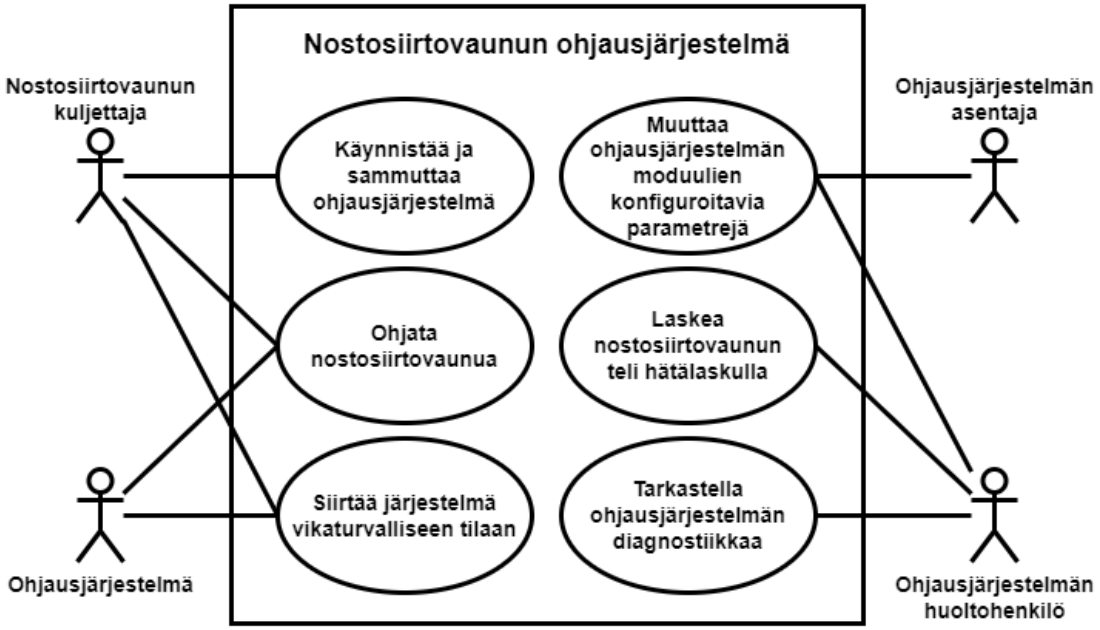
Kuva 4. *Vetomestari [5].*

Vetomestarin sekä nostosiirtovaunun liitäntärajapinta on huomioitava ohjausjärjestelmän ja sen moduulien suunnittelussa. Nostosiirtovaunu kiinnitetään vetomestariin mekaanisella, sähköisellä sekä hydraulisella liitännällä. Nostosiirtovaunu kytketään mekaanisesti vetomestarin nostopöydäksi kutsuttuun toimilaitteeseen. Nostopöydän toiminnallisuus mahdollistaa nostosiirtovaunun ja vetomestarin kiinnityksen sekä kiinnityspisteen

korkeuden ohjaamisen. Vetomestariin on myös toteutettu vaunun tarvitsema virran- sekä hydraulipainesyöttö. Ohjausjärjestelmän toimintaan tarvittu ohjaussignaalia tuottava sauvaohjain on yleisesti sijoitettu vetomestariin.

2.3 Ohjausjärjestelmän vaatimukset

Ohjausjärjestelmän tehtävä on muuttaa sauvaohjaimelta saatu ohjaussignaali nostosiirtovaunun hydrauliseksi ohjaukseksi sekä samanaikaisesti tarkkailla ohjausjärjestelmän sisäistä tilaa. Kuva 5 on esitetty nostosiirtovaunun ohjausjärjestelmän käyttötapaukset Unified Modeling Language:n (UML) mukaisessa käyttötapauskaaviossa. Ohjausjärjestelmän vaatimukset on määritetty kaavion perusteella.



Kuva 5. *Nostosiirtovaunun ohjausjärjestelmän käyttötapauskaavio.*

Käyttötapauskaaviossa on esitetty nostosiirtovaunun ohjausjärjestelmän käyttötapaukset sekä sidosryhmät. Järjestelmän kanssa toimiva taho saattaa edustaa yhtä tai useampaa sidosryhmää samanaikaisesti. Nostosiirtovaunun kuljettajan on pystyttävä ohjaamaan nostosiirtovaunua ohjausjärjestelmän avulla. Nostosiirtovaunua ohjataan vetomestariin sijoitetulla sauvaohjaimella kuljettajan toimesta. Ohjaustoiminnallisuuden mahdollistamiseksi kuljettajan on pystyttävä käynnistämään ohjausjärjestelmä. Ohjausjärjestelmä tulee myös pystyä sammuttamaan. Kuljettajan on myös pystyttävä siirtää järjestelmä vikaturvalliseen tilaan. Kuljettaja siirtää ohjausjärjestelmän vikaturvalliseen tilaan painamalla hätä-seis-painiketta. Ohjausjärjestelmän on pystyttävä ohjaamaan nostosiirtovaunua teliä automaattisen keskitys toiminnallisuuden toteuttamiseksi. Ohjausjärjestelmän on myös pystyttävä siirtää järjestelmä vikaturvalliseen tilaan. Ohjausjärjestelmä

siirretään vikaturvalliseen tilaan virhetilaanteen tapahtuessa. Nostosiirtovaunun ohjausjärjestelmän asentajan on pystyttävä muuttamaan järjestelmän moduulien konfiguroitavia parametrejä. Järjestelmän moduulien konfiguroitavat parametrit asetetaan ohjausjärjestelmän kokoonpanon yhteydessä. Ohjausjärjestelmän huoltohenkilön on pystyttävä laskemaan nostosiirtovaunun teli alas hätälaskun avulla. Hätälasku mahdollistaa nostosiirtovaunun telin laskemisen tapauksessa, jossa ohjausjärjestelmän toiminta on vikaantunut. Huoltohenkilön on myös pystyttävä tarkastelemaan ohjausjärjestelmän diagnostiikkaa järjestelmän virhetilanteiden selvittämiseksi. Korvattaessa ohjausjärjestelmän vikaantunut moduuli huoltohenkilön on pystyttävä muuttamaan korvaavan moduulin konfiguroitavia parametrejä järjestelmään toteutukseen sopiviksi.

Ohjausjärjestelmälle on myös asetettu vaatimus laajennettavuudesta tuoteperheeksi tulevaisuudessa. Ohjausjärjestelmän laajennettavuutta tuetaan hajautetulla modulaarisella ohjausjärjestelmäsuunnittelulla. Tämä asettaa ohjausjärjestelmän toteuttamiselle vaatimuksen modulaarisesta rakenteesta. Ohjausjärjestelmän moduulit hajautetaan järjestelmän toiminnallisuutta toteuttavien osien läheisyyteen. Ohjausjärjestelmän laajennettavuutta tuetaan myös moduulien uudelleenkäytettävyyden kasvattamisella. Uudelleenkäytettävyyttä edistetään toteuttamalla ohjausjärjestelmän moduulit alustariippumattomina toteutuksina. Tämä asettaa ohjausjärjestelmälle vaatimuksen hajautetusta rakenteesta ja alustariippumattomasta toteutuksesta.

Ohjausjärjestelmän hajautettu modulaarinen rakenne muodostaa tarpeen järjestelmän moduulien väliselle kommunikaatiolle. Moduulien välinen kommunikaatio on mahdollista toteuttaa analogisen tai digitaalisen kommunikaation avulla. Ohjausjärjestelmän moduulien välisen kommunikaation toteuttaminen digitaalisen kommunikaatio avulla sisältää useita etuja perinteiseen analogiseen kommunikaatioon verrattuna. [6] Moduulien välinen digitaalinen kommunikaatio luo tarpeen viestintäjärjestelmälle. Kommunikaation toteuttamiseen on kannattavaa hyödyntää standardoitua viestintäjärjestelmää, kuten CANopen protokollaperhettä [4]. Tämä asettaa järjestelmälle vaatimuksen digitaalisen kommunikaation sekä viestintäjärjestelmän käytöstä.

3. OHJAUSJÄRJESTELMÄSUUNNITTELUN TEOREETTINEN TAUSTA

Tässä luvussa käsitellään tuotemodulaation, vakiorajapintojen sekä konfiguraationhallinnan menetelmiä ohjausjärjestelmäsuunnitellussa. Luvussa esitellään myös CANopen viestintäjärjestelmän toimintaa.

3.1 Tuotemodulointi

Tuotemodulaatio on järjestelmäsuunnittelun toimintatapa, jossa suunniteltava järjestelmän jaetaan osakokoonpanoihin sekä komponentteihin [1]. Modulaarisessa järjestelmäsuunnittelussa suunniteltava järjestelmä jaetaan osakokonaisuuksiin, eli moduuleihin, suunniteltavan järjestelmän toiminnallisten vaatimusten perusteella. Järjestelmän samaa toiminnallisuutta tai toiminnallisuusryhmää toteuttavat osat yhdistetään omiksi moduuleikseen.

Marshallin ja Leaneyn [7] mukaan modulaarisesti toteutetussa järjestelmässä moduulien tulisi täyttää seuraavat ehdot:

1. Moduulit ovat yhteistyössä toimiva alijärjestelmiä, joista muodostetaan koko järjestelmän toteutus
2. Moduulin päätoiminnallisuus on toteutettu kokonaan moduulin sisällä, eikä muiden moduulien välillä
3. Moduulilla on yksi tai useampi tarkasti määritelty toiminnallisuus, joka voidaan testata erillään muusta järjestelmästä
4. Moduulit ovat itsenäisesti toimivia kokonaisuuksia, joita voidaan yhdistää sekä konfiguroida samanlaisten yksiköiden kanssa erilaisen lopputuloksen saavuttamiseksi [7].

Modulaarisesti toteutetun järjestelmän jokaisen moduulin tulee täyttää yllä esitetyt ehdot. Jos moduuli ei täytä kaikkia yllä esitettyjä ehtoja, tulee moduulin toteutus yhdistää toisen moduulin kanssa tai hajottaa yhdeksi tai useammaksi moduuliksi yllä esitettyjen ehtojen täyttämiseksi.

Modulaarisen järjestelmäsuunnittelun avulla suunnittelua toteuttava taho pystyy hallitsemaan astetta, jolla toiminnallisten vaatimusten muutokset vaikuttavat järjestelmään. Edistämällä moduulien vaihdettavuutta tuotemodulaatio antaa suunnittelijalle enemmän joustavuutta muuttuvien toiminnallisten vaatimusten toteuttamiseksi. Tämä lisääntynyt joustavuus mahdollistaa moduulien toteutusten suunnittelupäätösten viivyttämisen ta-

pauksissa, joissa lisää informaatiota moduulin toiminnasta tarvitaan suunnittelupäätösten aikaansaamiseksi. [1] Lisääntynyt joustavuus mahdollistaa myös järjestelmän moduulien rinnakkaisen kehitystyön, sillä moduulien toiminta on toisistaan riippumatonta.

Kehitettävän järjestelmän elinkaaren kustannuksia sekä tuotekehityksen toistuvia prosesseja vähennetään tuotemodulaation avulla [7]. Kehitettävän järjestelmän modulaarinen rakenne mahdollistaa moduulien monistamisen sekä uudelleenkäytön järjestelmän sekä tuoteperheen muiden tuotteiden kehityksessä. Moduulien monistaminen sekä uudelleenkäyttö vähentää työmäärää, joka järjestelmäprojektien toteuttamiseen vaaditaan. Vähentynyt työmäärä pienentää tuotekehityksen kuluja.

Järjestelmän jakaminen moduuleihin mahdollistaa komponenttien ja osakokoonpanojen toiminnan standardoinnin [1]. Moduulin toiminnan sekä sen käyttämän rajapinnan standardointi mahdollistaa moduulin sisäisen toteutuksen muuttamisen vaikuttamatta muun järjestelmän toimintaan. Komponenttien tapauksessa standardointi mahdollistaa komponentin vaihtamisen vastaavaan markkinoilta löytyvään komponenttiin, joka toteuttaa vastaavaa vakiorajapintaa.

3.2 Vakiorajapinnat

Modulaarisessa järjestelmäsuunnittelussa rajapinnalla kuvataan moduulien väliseen kommunikaation käytettävää rajapintaa. Modulaarisesti toteutetussa järjestelmässä moduulien välinen kommunikaatio toteutetaan moduulien jakamien rajapintojen kautta. Vakioimalla moduulien väliseen kommunikaatioon käytetyt rajapinnat edistetään moduulien hallittavuutta ja monistettavuutta. [8] Järjestelmän moduulien käyttämät kommunikaatiorajapinnat voidaan määrittää järjestelmäsuunnittelijan toimesta tai käyttää olemassa olevaa tietoliikenneprotokollaa. Moduulin vakiorajapinta edistää moduulin konfiguraatiohallintaa määrittämällä standardoidun tavan konfiguroitavien parametrien hallitsemiseen.

Järjestelmän moduulien välisen kommunikaation toteuttamiseen on kannattavaa hyödyntää standardoitua tietoliikenneprotokollaa. Tietoliikenneprotokollat määrittävät standardoidun tavan toteuttaa moduulien välisen kommunikaation. Moduulien välisen kommunikaation toteuttamiseen on myös kannattavaa hyödyntää tietoliikenneprotokollan päälle rakennettua viestintäjärjestelmää. Viestintäjärjestelmä, kuten CANopen, määrittää kommunikaatiolle standardoidun protokolla pohjaisen viestintä rakenteen. [4]

Modulaarisessa järjestelmäsuunnitellussa moduulin toiminnallisuus piilotetaan moduulin rajapinnan taakse. Vakioimalla moduulin rajapinnan toteutus mahdollistetaan moduulin sisäisen toteutuksen muokattavuus vaikuttamatta muun järjestelmän toimintaan. Vakiorajapinnan säilyessä muuttumattomana voidaan moduulin sisäistä toteutusta muokata

vapaasti. Kompleksien moduulien tapauksessa moduulien monimutkainen toteutus pystytään piilottamaan vakiorajapinnan taakse. Kompleksin toteutuksen piilottaminen rajapinnan taakse helpottaa moduulien välisen kommunikaation toteuttamista sekä edistää järjestelmän hallittavuutta.

Moduulien välisen kommunikaation toteuttamiseen käytetty tietoliikenneprotokolla määrittää moduulien rajapintojen toteutukset. CANopen viestintäjärjestelmän mukaisessa kommunikaatiossa erilaisten toimilaitteiden sekä moduulien rajapintojen toteutukset on määritetty toimilaitteiden toimintaa määrittävissä laiteprofiileissa ja standardeissa. Erilaisia laiteprofiileita on määritetty erilaisille yleisille teollisuuden toimilaitteille kuten esimerkiksi sauvaohjaimille, antureille ja venttiileille. Tapauksessa, jossa toteutettavalle toimilaitteelle ei ole määritetty yksilöivää laiteprofiilia voidaan sen rajapinnan toteutukseen käyttää yleistä laiteprofiilia. [9] Vakiorajapinnat mahdollistavat markkinoilta saatavilla olevan komponentin tehokkaan integroimisen toteuttavaan järjestelmään. Toteutetun järjestelmän toimilaitteen tai komponentin elinkaaren tai toimituksen keskeytyessä voidaan toimilaite tai komponentti korvata vastaavalla markkinoilta löytyvällä toimilaitteella tai komponentilla.

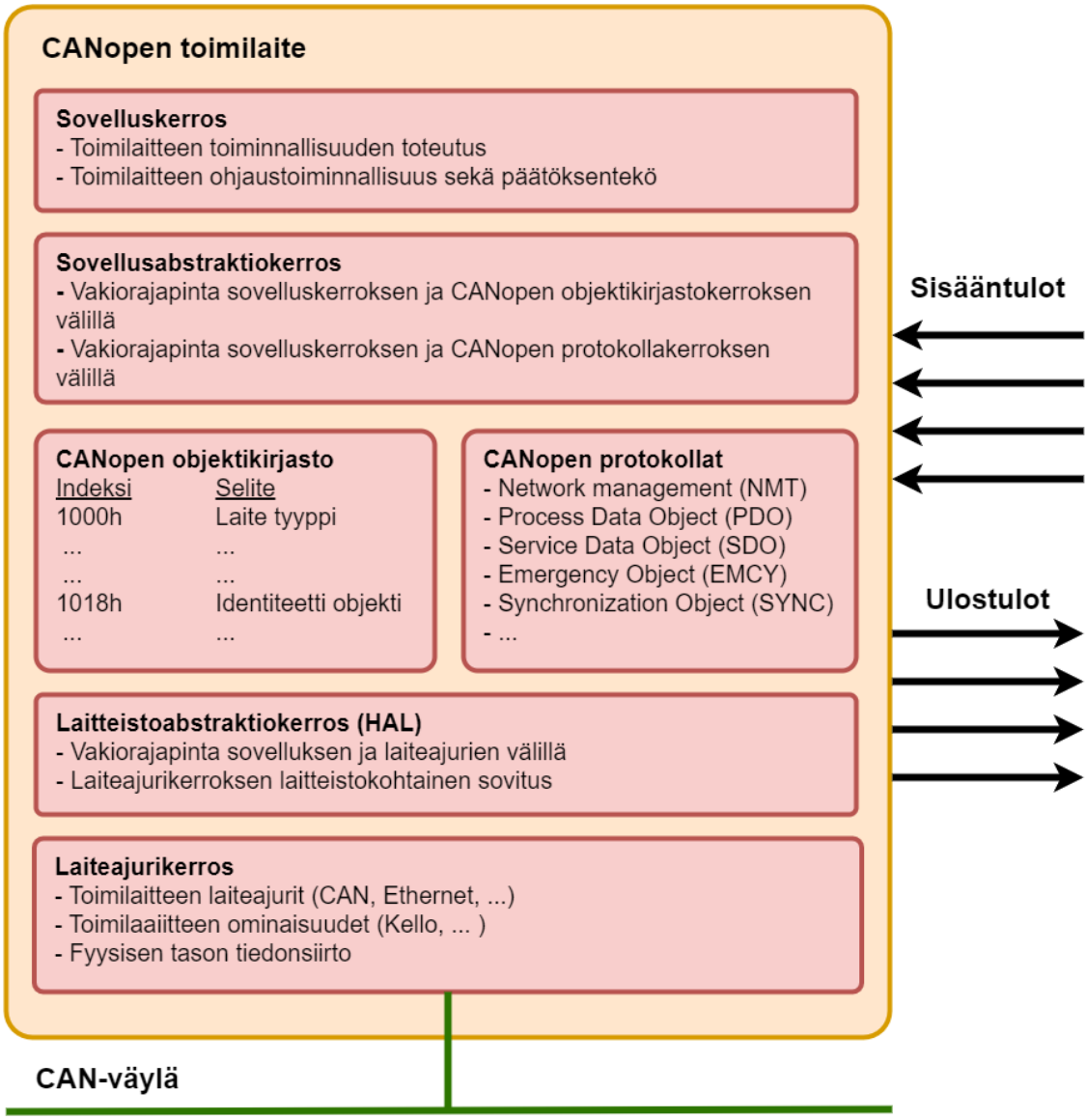
Moduulien vakiorajapinnoilla edistetään myös moduulien kehitettävyyttä sekä testattavuutta. Moduulin vakiorajapinta mahdollistaa moduulin määritetyn toiminnallisuuden tehokkaan testaamisen vakiorajapinnan kautta. Vakiorajapinta vähentää tarvetta uudelleenkirjoittaa moduulitestejä [10]. Moduulin rajapinnan muuttuessa moduulitestit täytyy päivittää aina uuden rajapinnan mukaisiksi.

3.3 CANopen

CANopen on Controller Area Network (CAN) protokollan avulla toteutettu viestintäjärjestelmä. CANopen viestintäjärjestelmä aluksi kehitettiin liikkuvien laitteiden ohjausjärjestelmien toteuttamiseen. Myöhemmin CANopen viestintäjärjestelmää on myös alettu soveltamaan erilaisissa tekniikan alan sovelluskohteissa, kuten bioteknologiassa sekä kiinteistöautomaatiossa. [11] CANopen viestintäjärjestelmä mahdollistaa standardoidun tavan toteuttaa sulautettujen järjestelmien digitaalista kommunikaatiota. CANopen viestintäjärjestelmän mukainen kommunikaatio edistää järjestelmän digitaalisen kommunikaation luotettavuutta ja tehokkuutta [4]. CANopen viestintäjärjestelmää kehittää sekä valvoo kansainvälinen CAN in Automation (CiA) organisaatio.

CANopen verkko sisältää CAN-protokollan mukaista kommunikaatiota toteuttavia CANopen toimilaitteita. CANopen toimilaitteiden standardoitu kommunikaatio sekä loogiset rajapinnat on toteutettu CANopen standardien sekä laiteprofiilien mukaisesti. Yleisesti

CANopen toimilaite on toteutettu pinomaisen rakenteen avulla, missä sovelluksen ja laitteen toiminnallisuus on jaettu omiksi kerroksikseen [12]. Kuva 6 on esitetty CANopen toimilaitteen yleinen rakenne CANopen toiminnallisuuden näkökulmasta.



Kuva 6. *CANopen laitteen yleinen rakenne. Muokattu kohteesta [12].*

CANopen toimilaitteen rakenne jaetaan eri toiminnallisuutta toteuttavien kerroksien ohjelmistopinoksi. Kommunikaatio ohjelmistopinon kerrosten välillä on toteutettu määritettyjen vakiorajapintojen avulla. Laiteajurikerros on ohjelmistopinon alin kerros, joka sisältää toimilaitteen laiteajurien sekä fyysisen tason tiedonsiirron toteutuksen. Ohjelmistototeutuksen seuraava kerros on laitteistoabstraktiokerros, joka toteuttaa rajapinnan toimilaitteen laiteajurikerrokselle. Laitteistoabstraktiokerroksen tehtävänä on yhdistää toimilaitteesta riippuva laiteajurikerros sekä toimilaitteen ohjelmistosovellus. Toimilaitteen toteutusalustan vaihtuessa laitteistoabstraktiokerros toteutetaan uudestaan vaikuttamatta

muun sovelluksen toimintaan. CANopen toimilaitteeseen on toteutettu CANopen viestintäjärjestelmän mukainen objektikirjasto sekä tarvittavat protokollat. Sovellusabstraktiokerros toteuttaa CANopen viestintäjärjestelmän kerroksille standardoidun rajapinnan. Ohjelmistopinon ylin kerros sisältää toimilaitteen sovelluksen toiminnallisuuden. Sovelluskerros sisältää toimilaitteen toiminnallisuuden toteutuksen. Toimilaitteen ohjaustoiminnallisuus sekä päätöksenteko on toteutettu tähän kerrokseen.

CANopen viestintäjärjestelmän mukainen objektikirjasto mahdollistaa CANopen toimilaitteiden konfiguroinnin. Objektikirjasto sisältää toimilaitetta sekä sen kommunikaatiota määrittäviä objektialueita. Objektikirjaston alueet on määritetty heksadesimaali muotoisten objekti-indeksien avulla [13]. Taulukko 1 on esitetty objektikirjaston CANopen viestintäjärjestelmän mukainen rakenne.

Taulukko 1. *CANopen objektikirjaston rakenne. Muokattu lähteestä [12].*

Indeksi	**Selite**
0000h	Varattu
0001h – 025Fh	Data tyypit
0260h – 0FFFh	Varattu
1000h – 1FFFh	Kommunikaatio objekti alue
2000h – 5FFFh	Laitevalmistajakohtainen alue
6000h – 9FFFh	Laiteprofiilikohtainen alue
A000h – BFFFh	Prosessikuva
C000h – FFFFh	Varattu

Objektikirjaston sisältämän informaation standardoitu rakenne edistää CANopen pohjaisten toteutuksien kehitettävyyttä sekä ylläpidettävyyttä. Määritetty rakenne edistää tiedon löytämistä sekä sijoittamista osana tuotekehitystä sekä ylläpitoa.

CANopen standardien mukainen objektikirjasto sisältää määritettäviä sekä konfiguroitavia objekteja. Konfiguroitavia objekteja käytetään CANopen toimilaitteiden konfiguraatiohallintaan. Objektikirjaston objektien arvoja muutetaan CANopen Service Data Object (SDO) protokollan mukaisen kommunikaation avulla [13]. Protokolla mahdollistaa objektikirjaston objektien arvojen lukemisen sekä kirjoittamisen CAN-väylän kautta, jos objektin tyyppi sen sallii.

CANopen viestintäjärjestelmä määrittää erilaisia CANopen profiileja. Profiileissa määritetään CANopen toimilaitteiden toteutuksien sisältämiä prosessitietoja, konfigurointiparametreja ja diagnostiikkatietoja. Profiileissa toimilaitteille määritetään standardoidut loogiset rajapinnat CiA 301 standardin mukaisesti [13]. Profiileissa määritetään erilaisia laiteprofiileja, kuten CiA 408 laiteprofiili, joka määrittää CANopen toimilaitteiden toimintaa [14].

3.4 Konfiguraatiohallinta

Konfiguraatiohallinnalla tarkoitetaan toimintatapaa, missä moduulin toiminnallisuutta muutetaan erilaisten konfiguroitavien parametrien avulla. Moduulin konfiguroitavia parametrejä muuttamalla moduulin toiminnallisia ominaisuuksia sekä laitetietoja pystytään hallitsemaan. Konfiguraatiohallintaa käytetään esimerkiksi venttiiliohjaimen ulostulojen määrittämiseen konfiguroitavien parametrien avulla. Konfiguraatiohallinta on usein toteutettu sarjaporttien, kenttäväylien tai fyysisten painikeiden avulla. Konfiguraatiohallinta edistää moduulin hallittavuutta, uudelleenkäytettävyyttä sekä huollettavuutta [15].

Moduulien konfiguroitavuus mahdollistaa moduulin tehokkaan integroimisen suunniteltavaan järjestelmään. Konfiguraatiohallinta vähentää tarvetta moduulien sisäisten toteutuksien muutoksille edistäen järjestelmän tuotekehitys- sekä ylläpitoprosessia. [15] Konfiguraatiohallinnan avulla ohjausjärjestelmän moduuleja pystytään monistamaan tehokkaasti tuoteperheen muihin tuotteisiin. Esimerkiksi nostosiirtovaunun tapauksessa venttiiliohjaimia monistetaan ohjausjärjestelmän toteutukseen telien lukumäärän mukaan. Konfiguraatiohallinta mahdollistaa moduulin integroimisen sovitettavaan järjestelmään ilman muutoksia laitteen ohjelmistosovellukseen.

CANopen viestintäjärjestelmä määrittää standardoidun tavan konfiguroitavien parametrien hallitsemiseksi. Toimilaitteen konfiguroitavia ominaisuuksia hallitaan objektikirjaston konfiguroitavien objektien arvoja muuttamalla. Määritetty tapa hallita ohjausjärjestelmän moduulien konfiguraatioita edistää järjestelmän hallittavuutta.

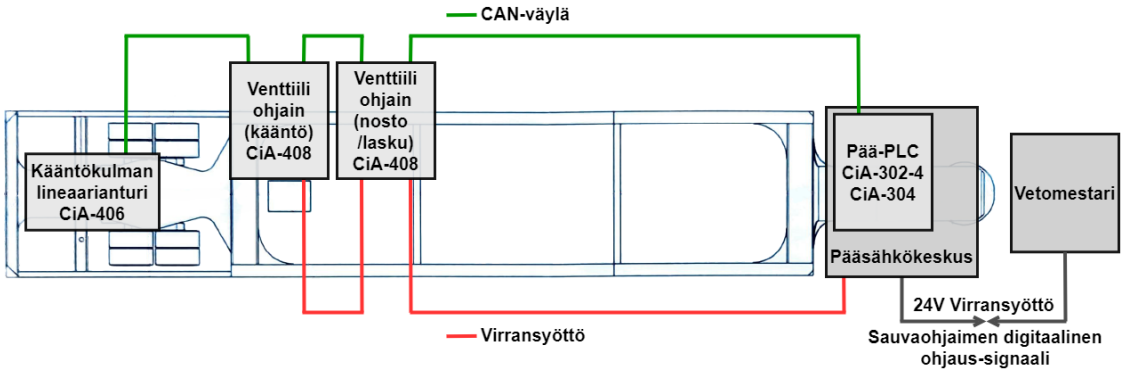
4. OHJAUSJÄRJESTELMÄN TOTEUTUS

Tässä luvussa käsitellään nostosiirtovaunun hajautetun modulaarisen ohjausjärjestelmän toteutusta. Luvussa on myös esitetty toteutettujen moduulien toiminnallisuus sekä ominaisuudet.

4.1 Ohjausjärjestelmän rakenne

Tässä luvussa käsitellään hajautetun modulaarisen ohjausjärjestelmän rakennetta järjestelmätasolla. Järjestelmän rakennetta käsiteltäessä järjestelmän toiminnallisuutta toteuttavat moduulit on esitetty omina toiminnallisina kokonaisuuksinaan. Järjestelmän moduulikohtaisia toteutuksia käsitellään seuraavassa luvussa.

Ohjausjärjestelmä on toteutettu hajautettuna modulaarisena rakenteena, missä järjestelmän toiminnallisuutta toteuttavat osat on jaettu osakokonaisuuksiin. Ohjausjärjestelmän osakokonaisuudet, eli moduulit, on sijoitettu hajautetusti nostosiirtovaunun toteutukseen lähelle moduulin toteuttamaa toiminnallisuutta. Ohjausjärjestelmän hajautettu ja modulaarinen rakenne on esitetty Kuva 7.



Kuva 7. *Ohjausjärjestelmän rakenne.*

Kuva 7 on esitetty ohjausjärjestelmän moduulien sijoittelu nostosiirtovaunun toteutuksessa. Nostosiirtovaunun hydraulisen telin kääntökulmaa mittaava lineaarianturi on sijoitettu kääntökulmaa ohjaavaan hydraulisylinteriin. Kääntökulmaa sekä nostokorkeutta ohjaavat venttiiliohjaimet on sijoitettu nostosiirtovaunun hydrauliikkalohkon läheisyyteen. Ohjausjärjestelmän ohjausta toteuttava ohjelmoitava logiikka on sijoitettu nostosiirtovaunun pääsähkökeskukseen. Ohjausjärjestelmän moduulien välinen digitaalinen kommunikaatio on toteutettu CANopen viestintäjärjestelmän avulla. Virransyöttö useimmille moduuleille on toteutettu erillisenä virransyöttönä. Lineaarianturin virransyöttö on toteutettu CAN-väylän virransyötön avulla. Järjestelmän moduulien virransyötöt on esitetty Kuva 7.

4.2 Moduulikohtaiset toteutukset

Tässä luvussa käsitellään ohjausjärjestelmän moduulien toiminnallisia ominaisuuksia sekä toteutuksia. Tässä luvussa käsitellään myös alustariippumattoman toteutuksen ominaisuuksia ja etuja.

Ohjaustoiminnallisuus

Nostosiirtovaunun ohjausjärjestelmän ohjaustoiminnallisuus on toteutettu ohjausjärjestelmän päätoiminnallisuutta hallitsevaan ohjelmoitavaan logiikkaan. Ohjausjärjestelmän ohjausta toteuttavan ohjelmoitavan logiikan pääasiallinen tehtävä on muodostaa venttiiliohjaimien ohjaussignaalit sekä valvoa ohjausjärjestelmän sisäistä tilaa. Ohjelmoitava logiikka muodostaa venttiiliohjaimien ohjaussignaalit sauvaohjaimen ohjaussignaalin sekä lineaarianturin mittaussignaalin perusteella. Ohjelmoitava logiikka valvoo järjestelmän sisäistä tilaa virhetilanteiden havaitsemiseksi. Virhetilaanteeksi ohjausjärjestelmän toiminnassa nähdään kommunikaatiovirheet sekä hätä-seis-painikkeen aktivointi. Virhetilanteen tapahtuessa ohjelmoitava logiikka siirtää ohjausjärjestelmän vikaturvalliseen tilaan. Vikaturvallisessa tilassa ohjausjärjestelmän toiminnallisuus on lukittu. Ohjausjärjestelmän sisäinen tila esitetään käyttäjälle nostosiirtovaunun valomajakan avulla.

Ohjelmoitava logiikka muodostaa nostokorkeutta sekä kääntökulmaa ohjaavien venttiiliohjaimien ohjaussignaalit eri tavoin. Nostosiirtovaunun nostokorkeutta ohjataan avoimen silmukan säädöllä, missä sauvaohjaimelta saatu ohjaussignaali välitetään suoraan venttiiliohjaimelle. Nostosiirtovaunun telin kääntökulman ohjaus sisältää kaksi eri ohjaustilaa. Ohjaustila valitaan sauvaohjaimen ohjaussignaalin arvon perustella. Sauvaohjaimen ollessa keskitettynä, eli ohjaussignaalin ollessa nolla, nostosiirtovaunun teli keskitetään automaattisesti suljetun silmukan säädöllä. Ohjaussignaalin erotessa nollasta telin kääntökulmaa ohjataan avoimen silmukan säädöllä.

Ohjelmoitavan logiikan toiminnallisuus on toteutettu IEC 61131-3 [16] standardin mukaisesti alustariippumattomana ohjelmistosovelluksena. Ohjelmistosovelluksen alustariippumattomuus mahdollistaa ohjelmistosovelluksen siirtämisen standardia tukevien laitealustojen välillä. Ohjelmistosovelluksen siirrettävyys edistää toteutuksen joustavuutta ja hallittavuutta mahdollistamalla toiminnallisuuden toteuttamisen laajemmalla komponenttivalikoimalla [17]. Ohjelmistosovelluksen toteutuksen ollessa riippumaton alkuperäisistä komponenteista toteutuksen elinikä määritetään tuoteomistajan toimesta eikä alkuperäisten komponenttien saatavuuden mukaan.

Venttiiliohjaimet

Ohjausjärjestelmän venttiiliohjaimien tehtävä on muuttaa ohjausta toteuttavalta ohjelmoitavalta logiikalta saatu ohjaussignaali hydraulisylinterien ohjaukseksi. Venttiiliohjaimien toiminnallisuus mahdollista nostosiirtovaunun nostokorkeuden sekä kääntökulman ohjaamisen. Ohjausjärjestelmä sisältää kaksi venttiiliohjainta. Yksi ohjaa nostosiirtovaunun nostokorkeutta ja toinen kääntökulmaa. Venttiiliohjaimiin ei ole toteutettu ohjaukseen liittyviä laskennallisia ominaisuuksia. Kaikki ohjaukseen liittyvä päätöksenteko on toteutettu ohjausta toteuttavaan ohjelmoitavaan logiikkaan. Venttiiliohjain toteuttaa ohjausta suoraviivaisesti vastaanotetun ohjaussignaalin perusteella. Venttiiliohjaimen ohjelmistosovellus on toteutettu IEC 61131-3 [16] standardin mukaisesti alustariippumattomana toteutuksena.

Venttiiliohjaimen monimutkainen sisäinen toteutus on piilotettu määritetyn rajapinnan taakse. Venttiiliohjaimen toteutuksen avulla hydraulisylinterin asentoa pystytään ohjaamaan yhden ohjaussuureen avulla. Moduulin monimutkaisen toteutuksen piilottaminen rajapinnan taakse helpottaa moduulien yhdistämistä ohjausjärjestelmäksi.

Lineaarianturi

Ohjausjärjestelmän lineaarianturin tehtävä on mitata nostosiirtovaunun telin kääntökulman suuruutta. Telin kääntökulman arvoa käytetään nostosiirtovaunun ohjaustoiminnallisuuden toteuttamiseksi. Nostosiirtovaunun toteutuksessa on käytetty magnetostriktiiviseen periaatteeseen pohjautuva lineaarianturia, joka integroidaan hydraulisylinterin asennukseen. Lineaarianturin toteutukseen hyödynnettiin valmista markkinoilta löytyvää komponenttia.

Virransyöttö lineaarianturille on toteutettu CAN-väylän kautta. Virransyöttö CAN-väylässä toimiville toimilaitteille voidaan toteuttaa suoraan väylän avulla. CAN-väylän kautta toteutetun virransyötön jännite on 3,3 V tai 5 V [18].

4.3 Moduulien CANopen integraatio

Ohjausjärjestelmän moduulien toteuttama toiminnallisuus yhdistetään nostosiirtovaunun ohjausjärjestelmäksi CAN-väylän kautta toteutetun digitaalisen kommunikaation avulla. Ohjausjärjestelmän CAN kommunikaatio noudattaa CANopen viestintäjärjestelmän mukaista standardoitua kommunikaatiota.

Ohjausjärjestelmän moduulit on toteutettu noudattamaan CANopen viestintäjärjestelmän mukaisia standardeja ja laiteprofiileja. Venttiiliohjaimet on toteutettu noudattamaan CiA 408 laiteprofiilia. CiA 408 laiteprofiili määrittää hydrauliikka komponenttien toimintaa

[14]. Lineaarianturi noudattaa CiA 406 laiteprofiilia, mikä määrittää anturien toimintaa [19]. Ohjausjärjestelmän kaikki moduulit on toteutettu noudattamaan CiA 301 ja CiA 302 kommunikaatio standardeja [13] [20]. Ohjausta toteuttava ohjelmoitava logiikka noudattaa myös CiA 304 standardia [21].

Ohjausjärjestelmän moduulien välinen prosessidatan välitys on toteutettu CANopen viestintäjärjestelmän Process Data Object (PDO) protokollan avulla [13]. Prosessidatan välitys ohjausjärjestelmän moduulien välillä on toteutettu laitteiden lähettämien ja vastaanottamien prosessidataobjektien avulla. Prosessidataobjektin data pakataan ja lähetetään moduulien välillä CAN-viestikehyksillä. Nostosiirtovaunun ohjausjärjestelmässä esimerkiksi venttiiliohjaimien ohjaussignaalit sekä lineaarianturin mittausarvo välitetään moduulien välillä prosessidataobjektien avulla. Prosessidataobjektien syklistä lähetystä hyödynnetään myös ohjausjärjestelmän kommunikaation virhetilanteiden havaitsemiseen. Virhetilanne prosessidataobjektien lähetyksessä tai vastaanotossa siirtää ohjausjärjestelmän vikaturvalliseen tilaan.

CANopen viestintäjärjestelmän mukaisella digitaalisella kommunikaatiolla ratkaistaan useita perinteisen analogisen kommunikaation ongelmia. Analogisen kommunikaation ongelmina nähdään toimilaitteiden mittaustarkkuuden heikentyminen, analogi-digitaalimuuntimista syntyvä mittausvirhe sekä digitaalimuuntimien korkeat kustannukset. Digitaalisen kommunikaation avulla analogisen kommunikaation yleisimmät ongelmakohdat pystytään ratkaisemaan. [6]

Ohjausjärjestelmän digitaalinen kommunikaatio mahdollistaa järjestelmän diagnostiikan tarkastelun moduulien välisestä kommunikaatiosta. Moduulien välinen kommunikaatio pitää sisällään ohjausjärjestelmän sekä moduulien eri tila- sekä prosessitietoja. Ohjausjärjestelmän kommunikaatiota on mahdollista tutkia reaaliajassa sekä tallentaa lokitiedostoon myöhäisempää tarkastelua varten.

Ohjausjärjestelmän moduuleihin on toteutettu CANopen viestintäjärjestelmän mukainen objektikirjasto. Objektikirjastoon on toteutettu laitteen hallintaan, kommunikaatioon ja konfiguraatioon tarvittavat objektit. Ohjausjärjestelmän moduulien objektikirjastot on toteutettu CiA 301 [13] standardin mukaisesti. Kaikki ohjausjärjestelmän moduulit on toteutettu noudattamaan CANopen viestintäjärjestelmän mukaista objektikirjasto rakennetta.

4.4 Moduulien konfiguraatiohallinta

Nostosiirtovaunun ohjausjärjestelmän toteutuksessa moduulien konfiguraatiohallinta on toteutettu CANopen viestintäjärjestelmän SDO protokollan avulla. Protokollan avulla moduulien konfiguroitavia parametrejä hallitaan CAN-väylän kautta tapahtuvan digitaalisen kommunikaation avulla. Konfiguroitavien parametrien arvoja muuttamalla moduulien toiminnallisia ominaisuuksia hallitaan ilman tarvetta muuttaa moduulien sisäisiä toteutuksia.

Ohjausta toteuttavan ohjelmoitavan logiikan konfiguraatiohallinta mahdollistaa moduulin ohjausnopeuden muuttamisen ja kommunikaatio parametrien asettamisen. Venttiiliohjaimien konfiguroitavien parametrien muuttaminen mahdollistaa ohjaimen fyysisten ulostulojen määrittämisen, moduulin kommunikaatio parametrien asettamisen sekä ohjaustilojen valinnan. Moduulien konfiguroitavat parametrit on asetettu ohjausjärjestelmän konfiguraation mukaisiksi.

Lineaarianturin konfiguraatioiden hallittavuus on määritetty komponentti valmistajan toimesta. Komponentin toteutuksen laajuus määrittää, mitkä ominaisuudet lineaarianturin toiminnassa ovat muutettavissa. Lineaarianturin konfiguroitavat parametri on määritetty CiA 406 laiteprofiilissa [19]. Toimilaitteen konfiguroitavat parametrit kuvataan Electronic Data Sheet (EDS) dokumentissa [13]. Laiteprofiili määrittää toimilaitteelle pakollisia sekä vallinnaisia konfiguroitavia parametrejä. Lineaarianturi sovitetaan ohjausjärjestelmään konfiguraatiohallinnan avulla.

5. OHJAUSJÄRJESTELMÄSUUNNITTELUN TULOKSET

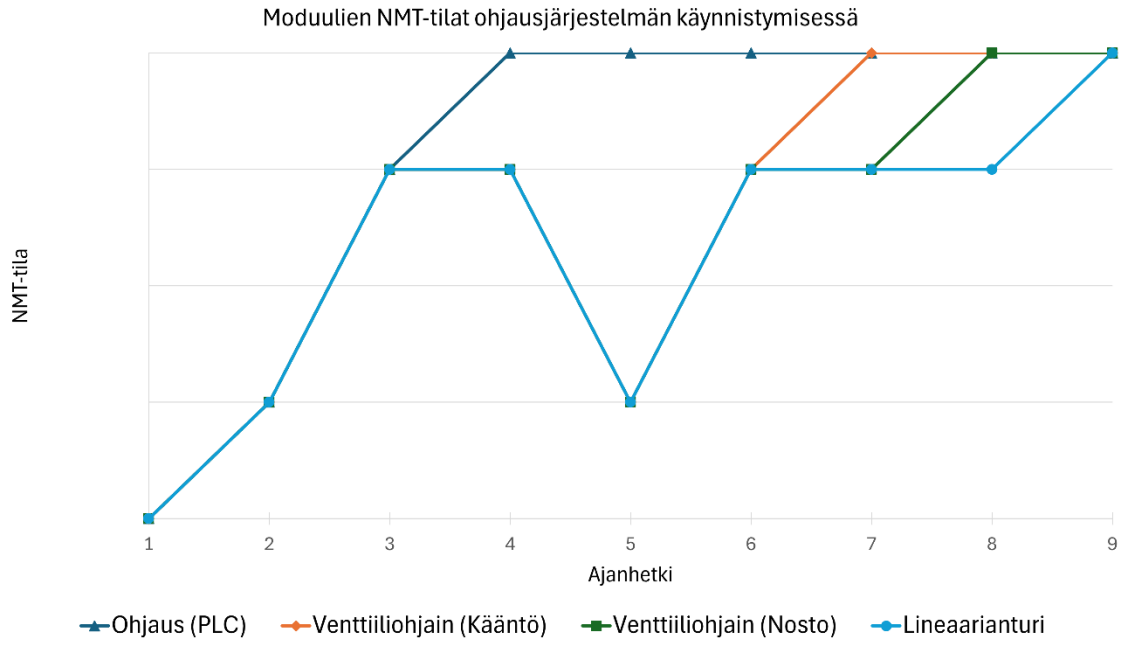
Tässä luvussa käsitellään nostosiirtovaunun hajautetun modulaarisen ohjausjärjestelmän toteutuksen tuloksia. Ohjausjärjestelmän toteutuksen tuloksia tarkastellaan tutkielman tutkimuskysymyksen pohjalta, eli miten hajautetulla modulaarisella järjestelmäsuunnittelulla edistetään tuotteen hallittavuutta ja laajennettavuutta.

Nostosiirtovaunun ohjausjärjestelmän prototyyppi kehitettiin tutkielmassa esitetyn teorian pohjalta. Ohjausjärjestelmän prototyyppiä testattiin asiakkaan toimesta teollisuuden olosuhteissa. Ohjausjärjestelmä täytti kaikki asiakkaan määrittämät toiminnalliset ja ei-toiminnalliset vaatimukset. Kuva 8 on esitetty nostosiirtovaunun prototyypin toteutus.

Kuva 8. *Nostosiirtovaunun prototyypin toteutus.*

Nostosiirtovaunua ohjattiin vetomestarilla ja toiminnallisuutta testattiin todellisen kuorman avulla.

Hajautetun modulaarisen ohjausjärjestelmän toiminnassa moduulien tiloja hallitaan CANopen Network Management (NMT) protokollan avulla. Ohjausjärjestelmän käynnistyessä moduulien tilat asetetaan toiminnalliseen tilaan NMT-protokollan avulla. Ohjausjärjestelmän moduulien tilanvaihdot on esitetty Kuva 9.



Kuva 9. *Moduulien NMT-tilanvaihdot ohjausjärjestelmän käynnistymisessä.*

Kuva 9 on muodostettu ohjausjärjestelmän CAN-väylästä mitatusta lokitiedostosta. Ohjausjärjestelmän käynnistyessä jokainen CAN-väylässä toimiva toimilaite käynnistyy ja siirtyy esitoiminnalliseen tilaan. Ohjausta toteuttava ohjelmoitava logiikka toimii ohjausjärjestelmän toiminnassa NMT-protokollan mukaisena NMT-masterina. Ohjelmoitava logiikkaa siirtyy toiminnalliseen tilaan ja lähettää verkkoon NMT-resetoinnin. Tämän seurauksena muut verkossa toimivat laitteet käynnistyvät uudelleen. Uudelleenkäynnistymisien jälkeen NMT-master siirtää muut verkossa toimivat laitteet toiminnalliseen tilaan yksi kerrallaan.

Ohjausjärjestelmän hajautetulla modulaarisella toteutuksella edistettiin järjestelmän laajennettavuutta tuoteperheen muiksi tuotteiksi. Ohjausjärjestelmän modulaarinen toteutus mahdollistaa moduulien monistamisen muihin toteutettaviin tuoteperheen tuotteisiin tulevaisuudessa. Esimerkiksi venttiiliohjaimia voitaisiin monistaa toteuttavaan tuotteeseen tarvittava määrä telien lukumäärän mukaan. Useampi telisessä järjestelmässä jokaista kääntyvää teliä kohden lisättäisiin yksi venttiiliohjain. Konfiguraatiohallinnan avulla moduuli sovitettaisiin toteutettavaan järjestelmään.

Ohjausjärjestelmän modulaarisella rakenteella ja alustariippumattomuudella edistettiin järjestelmän hallittavuutta. Moduulit toteutettiin alustariippumattomina ohjelmistosovelluksina mahdollistaen moduulien toteuttamisen millä tahansa markkinoilta löytyvillä komponenteilla. Moduulien konfiguraatiohallinnan avulla ohjausjärjestelmän huollettavuutta ja ylläpidettävyyttä lisätiin.

Automaatioväylä pohjaisella toteutuksella ohjausjärjestelmän kommunikaatioon tarvittavan kaapeloinnin määrää vähennettiin. Pienempi määrä kaapelointia vähentää mahdollisia kaapeloinnista johtuvia ongelmia edistäen järjestelmän toimintavarmuutta ja hallittavuutta. Automaatioväylä pohjaisen ratkaisun alkukustannukset ovat perinteistä analogista ratkaisua suuremmat. Suuremmat alkukustannukset kuitenkin säästetään moninkertaisesti järjestelmän elinkaaren aikana vähentyneiden huolto- sekä ylläpitokustannusten muodossa.

6. YHTEENVETO JA KESKUSTELU

Työn tavoitteena oli suunnitella ja toteuttaa asiakkaan vaatimukset täyttävä nostosiirtovaunun ohjausjärjestelmän prototyyppi. Ohjausjärjestelmän yhdeksi vaatimukseksi määritettiin tuotteen laajennettavuus tuoteperheeksi tulevaisuudessa. Vaatimus tuotteen laajennettavuudesta asettaa ohjausjärjestelmän rakenteelle vaatimuksen hajautetusta modulaarisesta arkkitehtuurista. Modulaarista järjestelmäsuunnittelua on jo pitkään hyödynnetty koneenrakennuksessa, mutta samat modulaarisen suunnittelun periaatteet ovat myös sovellettavissa automaatiojärjestelmiin.

Tutkielman teoria osiossa tarkasteltiin hajautettuun modulaariseen järjestelmäsuunnittelun teoreettisia menetelmiä. Teoria osiossa osoitettiin, miten modulaarisella järjestelmäsuunnittelulla pystytään edistämään järjestelmän hallittavuutta ja laajennettavuutta. Osiossa myös käsiteltiin digitaalisen kommunikaation etuja perinteiseen analogiseen kommunikaatioon verrattuna.

Nostosiirtovaunun ohjausjärjestelmän prototyyppi toteutettiin tutkielmassa esitetyn teorian pohjalta. Ohjausjärjestelmän toimintavaatimusten sekä toimintavarmuuden täyttymisen arviointi suoritettiin asiakkaan toimesta todellisissa teollisuuden olosuhteissa. Testien pohjalta määritettiin, että nostosiirtovaunun ohjausjärjestelmän toteutus täytti kaikki sille asetetut vaatimukset.

Tutkielman tulos kappale osoittaa hajautetun modulaarisen järjestelmäsuunnittelun hyödyn perinteiseen järjestelmäsuunnitteluun verrattuna. Modulaarisella järjestelmäsuunnittelulla edistettiin ohjausjärjestelmän laajennettavuutta, uudelleenkäytettävyyttä, kustannustehokuutta sekä tulevaisuudenkestävyyttä. Hajautetun modulaarisen järjestelmäsuunnittelun suuremmat kehityskustannukset säästetään moninkertaisesti tuotteen sekä tuoteperheen elinkaaren aikana vähentyneiden huolto-, ylläpito- sekä kehityskustannusten muodossa.

Tutkielma avaa erilaisia mahdollisuuksia aiheen jatkotutkimukselle. Tutkielman pohjalta jatkotutkimusta voidaan tehdä nostosiirtovaunun prototyypin laajentamista tuoteperheen muiksi tuotteiksi seuraavien projektien yhteydessä. Tutkimuksessa voitaisiin käsitellä tuotemodulaation onnistumista sekä tuoteperheen hallintaprosesseja. Tutkielma mahdollistaa myös hajautetun modulaarisen järjestelmäsuunnittelun kustannustehokkuuden tarkastelun.

LÄHTEET

[1] Gershenson JK, Prasad GJ, Zhang Y. Product modularity: Definitions and benefits. Journal of engineering design. 2003;14(3):295–313.

[2] Hansen PK, Sun H. A comprehensive view on benefits from product modularization. In: 11th International Design Conference, DESIGN 2010. 2010.

[3] Contik-mallisto – Contik Group Oy [Internet]. [viitattu 10. syyskuuta 2022]. Saatavissa: https://contik.fi/fi/contik-mallisto/

[4] Hietikko M, Malm T, Saha H. Comparing performance level estimation of safety functions in three distributed structures. Reliability engineering & system safety. 2015;134:218–29

[5] Heavy-Duty-Tractor HD 445 - Tractors / MAFI Transport-Systeme GmbH [Internet]. [viitattu 22. toukokuuta 2022]. Saatavissa: https://www.mafi.de/en/products/tractors/heavy-duty-tractor-hd-445/

[6] Maurice Bildstein, Stefan Heusel. Digital transmission in pressure sensors; 2015.

[7] Marshall R, Leaney PG. A systems engineering approach to product modularity. Proceedings of the Institution of Mechanical Engineers Part B, Journal of engineering manufacture. 1999;213(8):847–51

[8] Mikkola J. Modularity and interface management of product architecture. In: PICMET ’01 Portland International Conference on Management of Engineering and Technology Proceedings Vol1: Book of Summaries (IEEE Cat No01CH37199). IEEE; 2001. p. 359 vol.1–.

[9] CAN in Automation (CiA): CANopen profiles [Internet]. [viitattu 18. syyskuuta 2022]. Saatavissa: https://www.can-cia.org/can-knowledge/canopen/canopen-profiles/

[10] Beningo J. Reusable Firmware Development: A Practical Approach to APIs, HALs and Drivers. 1st ed. Berkeley, CA: Apress L. P; 2017.

[11] CAN in Automation (CiA): CANopen [Internet]. [viitattu 24. syyskuuta 2022]. Saatavissa: https://www.can-cia.org/canopen/

[12] CAN in Automation (CiA): CANopen internal device architecture [Internet]. [viitattu 24. syyskuuta 2022]. Saatavissa: https://www.can-cia.org/can-knowledge/canopen/device-architecture/

[13] CiA 301 CANopen application layer and communication profile Version: Version: 4.2.0. CAN in Automation; 2011.

[14] CiA 408 Draft Specification Proposal Version 2.1.0. CANopen device profile for fluid power technology. CAN in Automation; 2021.

[15] Choi M, Erikstad SO. A module configuration and valuation model for operational flexibility in ship design using contract scenarios. Ships and offshore structures. 2017;12(8):1127–35.

[16] IEC 61131-3:2003(E). Programmable controllers – Part 3: Programming languages. International Electrotechnical Commission; 2003.

[17] Feldmann K, Wolf W, Weber M. Development of an open, event-based and platform independent architecture for distributed and intelligent control systems. In: 2nd IEEE International Conference on Industrial Informatics, 2004 INDIN '04 2004. IEEE; 2004. p. 560–6.

[18] Corrigan S. Controller Area Network Physical Layer Requirements. 2008

[19] CiA 406 Draft Specification Proposal Version 4.1.0. CANopen device profile forencoders. CAN in Automation; 2019.

[20] CiA 302 Draft Standard Proposal Version: 4.1.0. Additional application layer functions. CAN in Automation; 2009.

[21] EN 50325-5. Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces - Part 5: Functional safety communication based on EN 50325-4. European Standard; 2010.