



# Hitsaustornin modulaarisen kehityksen määrittäminen

Ammattikorkeakoulututkinnon opinnäytetyö  
Sähkö- ja automaatiotekniikka, insinööri (AMK)

Kevät 2025

Casper Rantanen

Koulutus Sähkö- ja automaatiotekniikka  
Tekijä Casper Rantanen  
Työn nimi Hitsaustornin modulaarisen kehityksen määrittäminen  
Ohjaaja Juha Sarkula

---

Vuosi 2025

Opinnäytetyön tavoitteena oli määrittää hitsaustornin modulaarista kehitystä. Työssä määritettiin modulaarisempi ratkaisu hitsaustornin liikkeiden ohjaukselle ja hitsaustornin I/O-terminaalit vakioitiin. Lisäksi työssä huomioitiin yrityksen nykyinen tapa toteuttaa versionhallintaa ja miten sitä voitaisiin parantaa jatkossa. Opinnäytetyön toimeksiantaja on Pemamek Oy.

Työssä esitellään tuulivoimalan runkoputken valmistusprosessi, jonka yksi keskeisistä laitteista on hitsaustorni. Tämän lisäksi yrityksen tarjoamista hitsaustorneista ja niiden ominaisuuksista on kerrottu tarkemmin. Hitsaustorni kuuluu yrityksen niin kutsuttuihin vakiolaitteisiin, joille on tehty vakioituja tuoteratkaisuja. Yrityksen nykytilannetta tutkittiin toimintatapojen ja ohjelmoinnin osalta sekä hitsaustornin tuotemallien mekaanisia ja sähköisiä komponentteja kartoitettiin.

Microsoft on lopettanut tuen Windows CE-käyttöjärjestelmälle ja Beckhoff Automation on ilmoittanut lopettavansa kyseistä käyttöjärjestelmää käyttävät komponentit. Tämän takia yrityksessä on aloitettu siirtymä kohti TwinCAT 3-ympäristöä. Samalla kun siirytään kohti modernimpaa ja modulaarisempaa ohjelmointiympäristöä, laitteiden ohjelmista halutaan tehdä modulaarisempia sekä yhdenmukaisempia tulevaisuudessa.

Nykytilanteen kartoituksen pohjalta aloitettiin ensiksi vakioimaan hitsaustornin I/O-terminaaleja ja -kanavia. Vakiointi saatiin toteutettua ja siitä tehtiin dokumentaatio, jossa on listattu kaikki hitsaustornin I/O-muuttujat. Dokumentaatioissa on myös selkeästi esitetty mitkä I/O-terminaalit ovat vakioita tornin tuotemallista riippumatta ja mitkä ovat joko optioita tai tiettyyn hitsausprosessiin sidottuja. Moottorin ohjaukselle halutaan tulevaisuudessa luoda yhteinen ohjauslohko, jolla voidaan ohjata mitä tahansa moottoria. Määrittely liikeohjauslohkon rajapinnoille tehtiin ja sen perusteella modulaarisempaa ohjauslohkoa tullaan kehittämään.

Tulevaisuudessa nykyiset hitsauksen ohjausjärjestelmät tulee korvaamaan uusi WCX-järjestelmä, jonka tarkoituksena on yhdistää nykyisten eri hitsauksen ohjausjärjestelmien ominaisuudet. WCX-hitsauksen ohjausjärjestelmän käyttöliittymä tullaan toteuttamaan TwinCAT 3 HMI:n avulla. Uuden järjestelmän myötä myös hitsauksen ohjaus ja parametrien käsittely tulee muuttumaan merkittävästi.

Versionhallintaan halutaan panostaa tulevaisuudessa enemmän ja yritykselle luodaan aktiivisesti ylläpidettävää ja kehitettävää kirjastoa, johon tullaan keräämään kaikki uudelleen käytettävät ohjelmalohkot. Versionhallinnan avulla halutaan myös luoda hitsaustornille pohjaprojekti. Versionhallinnalla varmistetaan, että ohjelmien modulaarisuus säilyy. Modulaarisempi rakenne vähentää automaatiosuunnittelun työkuormaa ja projektin läpimenon pitäisi olla suoraviivaisempi.

Vasta tulevaisuudessa kuitenkin tullaan näkemään käytännöntasolla millainen vaikutus modulaarisemmalla ohjelmarakenteella on. Aika myös näyttää millainen hitsaustornin pohjaprojekti tulee olemaan.

Avainsanat Hitsaustorni, modulaarisuus, TwinCAT 3, versionhallinta  
Sivut 45 sivua

DP Electrical and Automation Engineering  
Author Casper Rantanen  
Subject Determining the Modular Development of a Welding Tower  
Supervisor Juha Sarkula

---

Year 2025

The aim of the thesis was to determine the modular development of a welding tower. The thesis defined a more modular solution for controlling the movements of the welding tower and the I/O terminals of the welding tower were standardized. Additionally, the current method of version control in the company was considered and how it could be improved in the future. The thesis was commissioned by Pemamek Oy.

The thesis presents the manufacturing process of the wind turbine main structural tube. One of the key devices in this process is the welding tower. The welding towers offered by the company and their features are described in more detail. The welding tower belongs to the company's so-called standard devices, for which standardized product solutions have been made. The current situation of the company was examined in terms of practices and programming. The mechanical and electrical components of different welding tower product models were documented.

Microsoft has ended support for the Windows CE operating system and Beckhoff Automation has announced that it will discontinue components using this operating system. As a result, the company has started transitioning towards the TwinCAT 3 environment. As the transition towards a more modern and modular programming environment progresses, the aim is to make the device software increasingly modular and standardized.

Based on the current assessment, the standardization of the welding tower's I/O terminals and channels was first initiated. The standardization was implemented, and documentation was created listing all the I/O variables of the welding tower. The documentation clearly presents which I/O terminals are standard regardless of the tower's product model and which are either optional or tied to a specific welding process. In the future, a common control block for motor controlling is desired. The interface definition for the control block was made, and a more modular control block will be developed based on this definition.

In the future, the current welding control systems will be replaced by a new WCX-system, which aims to combine the features of the current different welding control systems. The user interface of the WCX-welding control system will be implemented using TwinCAT 3 HMI. With the new system, the control and handling of welding parameters will also change significantly.

More emphasis will be placed on version control in the future, and an actively maintained and developed library will be created for the company, which will collect all reusable program blocks. Version control also aims to create a base project for the welding tower. Version control ensures that the modularity of the programs is maintained. A more modular structure reduces the workload of automation designing and the project throughput should be more straightforward.

However, the practical impact of a more modular program structure will only become evident in the future. It also remains to be seen what the base project for the welding tower will ultimately be like.

Keywords Welding tower, modularity, TwinCAT 3, version control

Pages 45 pages

# Sisällys

1	Johdanto .....	1
2	Modulaarisuus ohjelmoinnissa .....	2
2.1	Edut .....	2
2.2	Versionhallinta .....	2
3	Ohjelmitava logiikka .....	3
3.1	IEC 61131-3-standardi .....	3
3.2	Structured Text (ST).....	4
3.3	Function Block Diagram (FBD).....	5
4	Beckhoff Automation .....	6
4.1	TwinCAT 2 .....	6
4.2	TwinCAT 3 .....	7
4.3	EtherCAT-kenttäväyläprotokolla.....	10
5	Tuulivoimalan runkoputken valmistusprosessi.....	11
5.1	Levyn reunan jyrä .....	11
5.2	Levyjen hitsaus .....	12
5.3	Pituussauman hitsaus .....	13
5.4	Koontiasema ja kehäsauman hitsaus sisäpuolelta .....	14
5.5	Kehäsauman hitsaus ulkopuolelta .....	15
5.6	Laipan asennus.....	16
6	Hitsaustorni .....	16
6.1	WC100- ja WC500-hitsauksen ohjausjärjestelmä.....	17
6.2	Hitsausvarustus .....	19
6.3	Railon seuranta.....	20
6.4	Liikkeet.....	21
6.5	Anturointi.....	22
7	Nykytilanne .....	23
7.1	Uuden projektin luonti .....	23
7.2	Nykyinen versionhallinta .....	24
7.3	Ohjelman rakenne.....	24
7.4	Käytössä olevien komponenttien kartoitus .....	26
7.4.1	Vakioliikkeet .....	27
7.4.2	Optioliikkeet.....	27

7.4.3	Hitsausvarustelu .....	28
7.4.4	Yhteiset moduulit .....	29
8	Hitsaustornin modulaarinen kehitys.....	30
8.1	I/O-terminaalien ja -kanavien vakiointi.....	30
8.2	Ohjelman modulaarisuuden kasvattaminen.....	34
8.3	WCX-hitsauksen ohjausjärjestelmä.....	36
8.4	Versionhallintajärjestelmän hyödyntäminen .....	37
9	Yhteenvedo ja pohdinta .....	37
	Lähteet.....	40

## Kuvat

Kuva 1.	Esimerkki IF-lauseesta (Beckhoff Information System, n.d.-b) .....	5
Kuva 2.	Esimerkki lohkoavioista (Beckhoff Information System, n.d.-c).....	5
Kuva 3.	TwinCAT System Manager (Beckhoff Information System, n.d.-f).....	7
Kuva 4.	TwinCAT 3-kehitysympäristö (Beckhoff Information System, n.d.-i) .....	8
Kuva 5.	EtherCAT-kenttäväylätopologia (EtherCAT Technology Group, n.d.) .....	11
Kuva 6.	Tuulivoimalan runkoputken valmistuksen päävaiheet.....	11
Kuva 7.	Levyjyrsin EBS (Pemamek Oy tietopankki, n.d.) .....	12
Kuva 8.	Levyhitsausasema EHD-R (Pemamek Oy tietopankki, n.d.).....	12
Kuva 9.	LW-asema ja pituussaumajyrsin LM55 (Pemamek Oy tietopankki, n.d.) .....	13
Kuva 10.	Koontiasema (Pemamek Oy tietopankki, n.d.).....	14
Kuva 11.	Kehäsaumajyrsin CM55 (Pemamek Oy tietopankki, n.d.).....	15
Kuva 12.	WS2-hitsausasema (Pemamek Oy tietopankki, n.d.).....	15
Kuva 13.	Laipan asennusasema (Pemamek Oy, tietopankki, n.d.).....	16
Kuva 14.	WC100-käyttöliittymä .....	17
Kuva 15.	WC500-käyttöliittymä .....	18
Kuva 16.	Esimerkki moottorin ajokäskyistä .....	24
Kuva 17.	Taajuusmuuttajan rajapintalohko.....	25
Kuva 18.	MC1-pääkaapin I/O-terminaalit .....	32
Kuva 19.	CB1-kenttäkotelon I/O-terminaalit .....	33
Kuva 20.	Pääohjelman rakenteen määrittely .....	34
Kuva 21.	Moottorilohkojen rajapinnat muuhun ohjelmaan .....	35
Kuva 22.	Määrittely moottorin ohjauksen toteutukselle tulevaisuudessa .....	35

# 1 Johdanto

Tässä opinnäytetyössä käsitellään hitsaustornin modulaarisen kehityksen määrittämistä. Hitsaustornin tarkoituksena on nopeuttaa ja automatisoida hitsausprosessia raskaassa metalliteollisuudessa. Työn toimeksiantajana toimii Pemamek Oy. Yritys tarjoaa hitsausautomaattioratkaisuja raskaaseen metalliteollisuuteen.

Työn tarkoituksena on modulaarisen kehityksen avulla luoda hitsaustornin PLC-ohjelmointikoodista helpommin hallittava tulevissa projekteissa. Modulaarisella kehityksellä pyritään yhdenmukaistamaan laitteiden ohjelmakoodeja, sillä tällä hetkellä eri laitteiden välillä on selkeitä eroja ohjelmarakenteen osalta. Työssä keskitytään liikkeiden ohjauksen modulaarisuuden kasvattamiseen ja hitsaustornin I/O-terminaaleille sekä -kanaville tehdään vakiointi. Vakioinnin tarkoituksena on vähentää sähkö- ja automaatio suunnittelun työkuormaa. Lisäksi työssä tehdään selvitystä tämänhetkisestä versionhallinnasta ja miten sitä voitaisiin tulevaisuudessa parantaa.

Eri variaatioiden ohjelmalohkot kerätään kirjastoon, josta sopivien ohjelmalohkojen valinta onnistuu helpommin projektin määrittelyn mukaan. Tällaisten valmiiden ohjelmalohkojen lisäys projektiin ylläpidetystä kirjastosta vähentäisi työmäärää ohjelmoinnin osalta, koska toimivaa ohjauslohkoa ei tarvitse joka kerta rakentaa uusiksi. Modulaarisen rakenteen myötä ohjelmistosuunnittelun yhdenmukaisuus kasvaa. Ohjelmakoodin ollessa yhdenmukainen, on toisen automaatio suunnittelijan helpompi lukea koodia ja ymmärtää mitä sen halutaan tekevän. Yhdenmukaisten jo aiemmin toimiviksi todettujen ohjelmalohkojen käyttö vähentää tulevaisuudessa ilmeneviä bugeja. Lisäksi yhdenmukaisuus ja yhteiset suunnittelumenetelmät auttavat ongelmien ratkaisemisessa.

Aihe on tärkeä, sillä sen avulla asiakkaille toimitettavien projektien työkuormaa saadaan vähennettyä ja kokonaisuus pysyy selkeämpänä. Tämä johtaa projektin mahdollisten riskien pienenemiseen ja läpimenoajan lyhenemiseen. Tämän myötä laitteiden tuottamiseen vaadittavat kustannukset pienenevät. Kun laitteen ohjelmistosuunnitteluun kuluva aika vähenee, aikaa saadaan enemmän käyttöönottoa varten, jossa usein ilmenee asennusvirheitä ja muuta vianetsintää. Aihe on siis hyvinkin ajankohtainen yrityksessä, jonka kasvu on ollut viime vuosina suurta. Projektien hallinta ja asiakkaiden kanssa sovituisissa aikatauluissa pysyminen ovat tärkeää yritykselle.

## 2 Modulaarisuus ohjelmoinnissa

Modulaarisuus tarkoittaa ohjelman jakamista toisistaan erillisiin ja itsenäisiin moduuleihin ja komponentteihin, joista jokainen vastaa tietyistä toiminnoista. Modulaarisuus tuo joustavuutta ja nopeutta, kun uusia toiminnallisuuksia lisätään ohjelmaan. Ohjelman modulaarisuutta voidaan mitata sillä perusteella, kuinka hyvin ohjelma on jaettu pienempiin osiin, joilla on standardoidut rajapinnat. Tavoitteena on luoda tuotteita, joihin voidaan yhdistää uudelleenkäytettäviä koodikomponentteja. Ohjelmiston modulaarisuuden saavuttamiseksi tulee olla selkeä ajatus siitä, mitkä ovat tavoitteet ja mitkä osuudet ohjelmasta kuuluvat yhteen ohjelmamoduulissa. (Bråtegren, n.d.)

### 2.1 Edut

Modulaarinen koodi parantaa koodin uudelleenkäytettävyyttä, sillä se mahdollistaa toiminnallisuuksien kapseloinnin ja hyödyntämisen eri sovelluksissa. Kun koodi on kerran kehitetty ja testattu toimivaksi, ohjelmoinnin tehokkuus ja laatu paranevat. Koodi on helpommin muokattavissa ja laajennettavissa, koska uusia toimintoja voidaan lisätä ilman, että koko ohjelmakoodia tarvitsee muuttaa merkittävästi. (Donnell, 2024)

Modulaarisuus helpottaa myös vianmäärittystä ja ylläpitoa, sillä moduulikohtaiset toiminnot ovat helpommin testattavissa ja vianetsintä on suoraviivaisempaa. Tämän lisäksi modulaarinen rakenne parantaa integrointia ja yhteensopivuutta, koska hyvin määriteltyjen rajapintojen avulla voidaan varmistaa, että vain tarvittavat tiedot ovat muiden osien käytettävissä. (Donnell, 2024)

### 2.2 Versionhallinta

Versionhallinnan avulla mahdollistetaan se, että koodikantaan tehdyt muutokset tallentuvat. Ohjelmistokehittäjät voivat tarkastella kuka on tehnyt muutoksia ja milloin ne on tehty. Sen avulla voidaan myös tarvittaessa palauttaa nykyinen versio aiempaan. Versionhallinta toimii niin kutsuttuna yhtenäisen totuuden lähteenä. Versionhallinta toimii turvaverkkona, joka suojelee ohjelmistoa peruuttamattomilta vahingoilta. Kehitystiimille on turvallisempaa kokeilla uusia toiminnallisuuksia ilman pelkoa koodin rikkoutumisesta tai ristiriitojen syntyisestä. (GitLab, n.d.)

Jos useampi kehittäjä työskentelee samanaikaisesti ja tekevät keskenään ristiriidassa olevia muutoksia, versionhallinta tunnistaa nämä ongelmakohdat. Tämän myötä tiimin

jäsenet voivat nopeasti palauttaa aiemman version ja vertailla muutoksia hyödyntämällä muutoshistoriaa. Versionhallintajärjestelmän avulla ohjelmistotiimi voi ratkaista ongelmat ennen niiden vaikutusta projektin etenemiseen. (GitLab, n.d.)

Kaksi suosituinta versionhallintajärjestelmää ovat keskitetty ja hajautettu. Keskitetyssä järjestelmässä kaikki tiedostot tallentuvat yhteen keskitettyyn arkistoon. Hajautetussa järjestelmässä tiedostot tallentuvat useisiin arkistoihin. Yleensä niin, että jokaisella kehittäjällä on oma täydellinen kopionsa koko koodihistoriasta. Harvinaisempia hallintatapoja ovat lukitukseen perustuva versionhallinta ja optimistinen versionhallinta. (GitLab, n.d.)

### 3 Ohjelmoitava logiikka

Ohjelmoitava logiikka (PLC) on eräänlainen pieni tietokone, joka vastaanottaa dataa tulojen kautta ja ohjaa lähtöjä saatujen signaalien perusteella. Ohjelmoitavan logiikan tehtävänä on ohjata järjestelmän toimintoja sisäisen ohjelmointinsa avulla. Ohjelmoitavat logiikat ovat nykyään yleisesti käytettyjä tärkeiden prosessien automatisoinnissa. (Polycase, 2021)

PLC lukee syötteitä joko antureista tai ihmisten antamina, esimerkiksi kytkimien ja painikkeiden kautta. Ohjelmointinsa perusteella PLC päättää muuttuuko lähtösignaalin tila. Ohjelmoitavan logiikan lähdöt voivat ohjata monia erilaisia laitteita. Näihin kuuluu mm. oikosulkumoottorit, solenoidiventtiilit, valot, kytkinlaitteet ja turvakatkaisut. (Sharma, 2024)

Ohjelmoitavat logiikat ovat korvanneet perinteiset releohjaukset. Releohjausjärjestelmät ovat monimutkaisia, alttiita vioille ja vievät runsaasti tilaa. Nykyään PLC:t ovat keskeinen osa teollisuuden ohjausjärjestelmiä. (Sharma, 2024)

#### 3.1 IEC 61131-3-standardi

IEC 61131-3 on ensimmäinen toimittajasta riippumaton ja standardoitu ohjelmointikieli teollisuusautomaatiolle. Standardin on laatinut kansainvälinen sähkötekniikan komissio (IEC), joka on perustettu vuonna 1906. Kansainvälinen sähkötekniikan komissio on maailmanlaajuisesti tunnustettu standardointiorganisaatio, jonka ohjausalan standardit on hyväksytty yli viidessäkymmenessä maassa. IEC 61131-3 on jo vakiintunut Euroopassa ja on nopeasti yleistymässä myös Pohjois-Amerikassa ja Aasiassa teollisuus- ja prosessiohjauksen ohjelmointistandardina. Standardin käyttöönottoa teollisuudessa vauhdittaa ohjaus- ja automaatio-sovellusten ohjelmistojen kasvava monimutkaisuus. Työvoimakustannukset ja ohjelmiston kehitys- ja ylläpitoaika vaikuttavat merkittävästi

ohjausprojekteihin. Standardoidun ohjelmointikielen käyttö vaikuttaa myönteisesti koko ohjelmiston elinkaareen, johon kuuluu mm. vaatimusten määrittely, suunnittelu, käyttöönotto ja ylläpito. (Real Time Automation, n.d.)

IEC 61131-3 tukee useita ohjelmointikieliä samassa ohjausohjelmassa. Ohjelmoija voi valita tehtävään parhaiten soveltuvan ohjelmointikielen, mikä parantaa tuottavuutta merkittävästi. Lisäksi vakioitu ohjelmointirajapinta on täysin laitteistoriippumaton, mikä alentaa ohjelmiston ylläpitokustannuksia yrityksen automaatio-sovelluksissa. (Real Time Automation, n.d.)

IEC on määritellyt joukon vakioituja tietotyyppejä, joita käytetään yhdenmukaisesti kaikissa IEC 61131-3-standardin mukaisissa ohjelmointiympäristöissä. Standardissa määritellään, kuinka muuttujan sisältö tulee tulkita. Tietyille tietotyypille sallitaan vain sen mukaiset operaatiot, esimerkiksi matemaattisia operaatioita voidaan suorittaa vain numeerisilla tietotyypeillä, ei bittijonoilla. (Devasia, 2021)

### 3.2 Structured Text (ST)

Structured Text (ST) on IEC 61131-3-standardiin kuuluva ohjelmointikieli, joka on verrattavissa muihin korkean tason ohjelmointikieliin, kuten C tai PASCAL. ST-kielen avulla voidaan toteuttaa monimutkaisia algoritmeja. Ohjelmakoodi koostuu lausekkeista ja komennoista, jotka voivat olla esimerkiksi kuvan 1 mukaisia ehtolauseita (IF...THEN...ELSE) tai toistolauseita (FOR...DO). Lauseke on rakenne, joka palauttaa arvon sen arvioinnin jälkeen. Komennot määrittelevät, kuinka lausekkeitä käsitellään. (Beckhoff Information System, n.d.-a)

ST-kielen etuihin kuuluu koodin helppo muokattavuus, sillä ST-kieli on muodoltaan tekstipohjainen. Tämä helpottaa koodin kopiointia, liittämistä ja koodin jakamista. Suuria tekstimääriä voidaan muokata tekstinkäsittelyohjelmalla ja liittää takaisin ohjelmaan. Tiivis ohjelmakoodin muoto mahdollistaa sen, että ohjelmoija voi nähdä suuremman määrän koodia yhdellä sivulla. Esimerkkinä 4 riviä tikapuulogiikkaa voi viedä näytön verran tilaa, kun ST-kielellä sama toiminnallisuus voidaan toteuttaa kahdella koodirivillä. (AutomationDirect, n.d.)

Kuva 1. Esimerkki IF-lauseesta (Beckhoff Information System, n.d.-b)

```

IF temp<17

THEN heating_on := TRUE;

ELSE heating_on := FALSE;

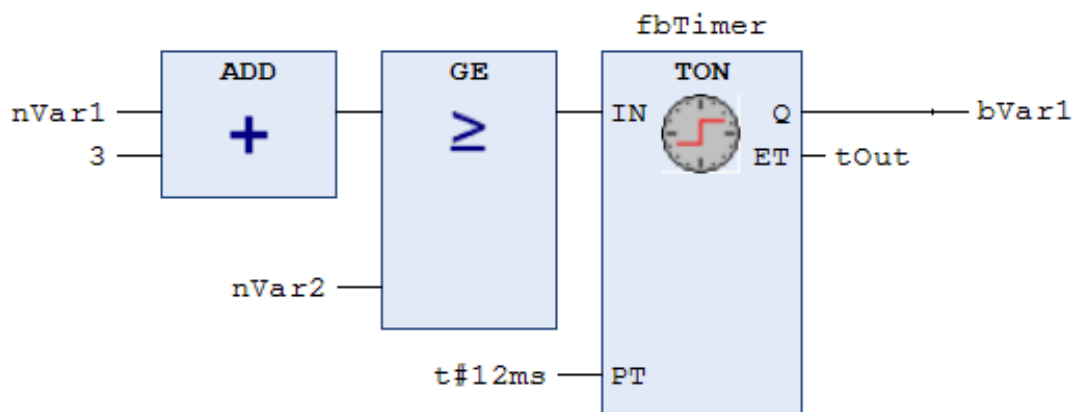
END_IF;

```

### 3.3 Function Block Diagram (FBD)

Function Block Diagram (FBD) on graafinen ohjelmointikieli, joka kuuluu IEC 61131-3-standardiin. FBD-ohjelmakoodi koostuu verkoista (network), jotka sisältävät kukin oman rakenteen. Rakenne voi sisältää loogisia ja aritmeettisiä lausekkeita, funktiolohkokutsuja, hyppykäskyjä tai palautuslauseita. Funktiolohkot perustuvat Boolean algebraan. Funktiolohkot ja muuttujat yhdistetään viivoilla kuvan 2 tapaisesti. Signaalien kulkusuunta verkossa on vasemmalta oikealle. (Beckhoff Information System, n.d.-c)

Kuva 2. Esimerkki lohkokaaviosta (Beckhoff Information System, n.d.-c)



FBD-kieli toimii hyvin esimerkiksi prosessiautomaatiossa, koska lohkokaavioita käytetään prosessien selittämiseen. Esimerkiksi yksi askel voi tapahtua vasta, kun edeltävä prosessi on valmis. FBD-kielellä toteutettu prosessin ohjaus eroaa sekvenssiohjauksesta, jossa eteneminen perustuu boolean-antureihin tai tapahtumiin. Turvaohjelmat käyttävät usein FBD-kieltä, koska ne ovat yksinkertaisia ja nopeita lukea. (Dietrich, 2023)

## 4 Beckhoff Automation

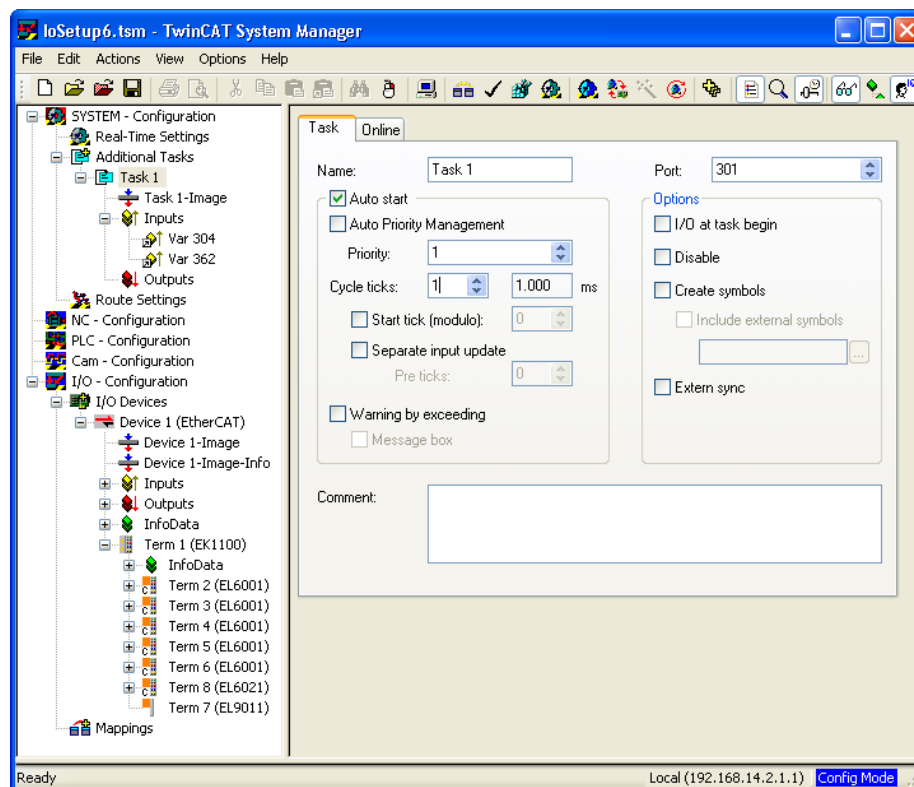
Beckhoff Automation toimittaa avoimia automaatiojärjestelmiä, jotka perustuvat PC-pohjaiseen ohjausteknologiaan. Beckhoffin tuotevalikoimaan kuuluu mm. teollisuus-PC:t, I/O- ja kenttäväyläkomponentit, liikkeenohjaustuotteet ja automaatio-ohjelmistot. Näillä tuotealueilla on kattava valikoima erilaisia tuotteita, joiden avulla voidaan luoda suuriakin kokonaisuuksia. Automaatoratkaisujen kehitys alkoi vuonna 1980 ja moni nykyäänkin käytössä oleva standardi on yrityksen kehittämä. Beckhoffin PC-pohjainen automaatio-ohjaus sekä TwinCAT-ohjelmisto ovat yrityksen merkittäviä tuotteita. (Beckhoff, n.d.-a)

### 4.1 TwinCAT 2

Vuonna 1996 julkaistu ensimmäisen sukupolven TwinCAT 2 käyttää kahta käyttöliittymää. TwinCAT System Manageria käytetään sähkömekaanisten eli sähkökaapissa olevien fyysisten komponenttien kanssa kommunikointiin ja TwinCAT PLC Control on tarkoitettu ohjelmoitavan logiikan kehittämiseen. TwinCAT tukee kaikkia yleisimpiä kenttäväyliä, esimerkiksi EtherCAT, Beckhoff Lightbus, Profibus DP, Interbus, CANopen ja DeviceNet. (Beckhoff Information System, n.d.-d)

TwinCAT System Manager on järjestelmän konfigurointikeskus. PLC-järjestelmien ohjelmat, akseliohjaukset sekä liitetyt I/O-kanavat liittyvät toisiinsa. System Manager yhdistää kaikki komponentit ja niiden tiedot. System Manager mahdollistaa I/O-kanavien yhdistämisen bittikohtaisesti sekä tarjoaa työkaluja, joilla esimerkiksi 100 kanavaa voidaan liittää peräkkäin yhden komennon avulla. Kenttäväylä- ja prosessikuvatasoilla käyttöönottoa helpottavat online-näkymä ja "Write and Force"-toiminnot. Watch-ikkunasta voidaan tarkastella valittuja muuttujia ja tiedot esitetään kaikille I/O-laitteille yhtenäisessä muodossa. Kuvassa 3 nähdään System Managerin käyttöliittymä. (Beckhoff Information System, n.d.-e)

Kuva 3. TwinCAT System Manager (Beckhoff Information System, n.d.-f)



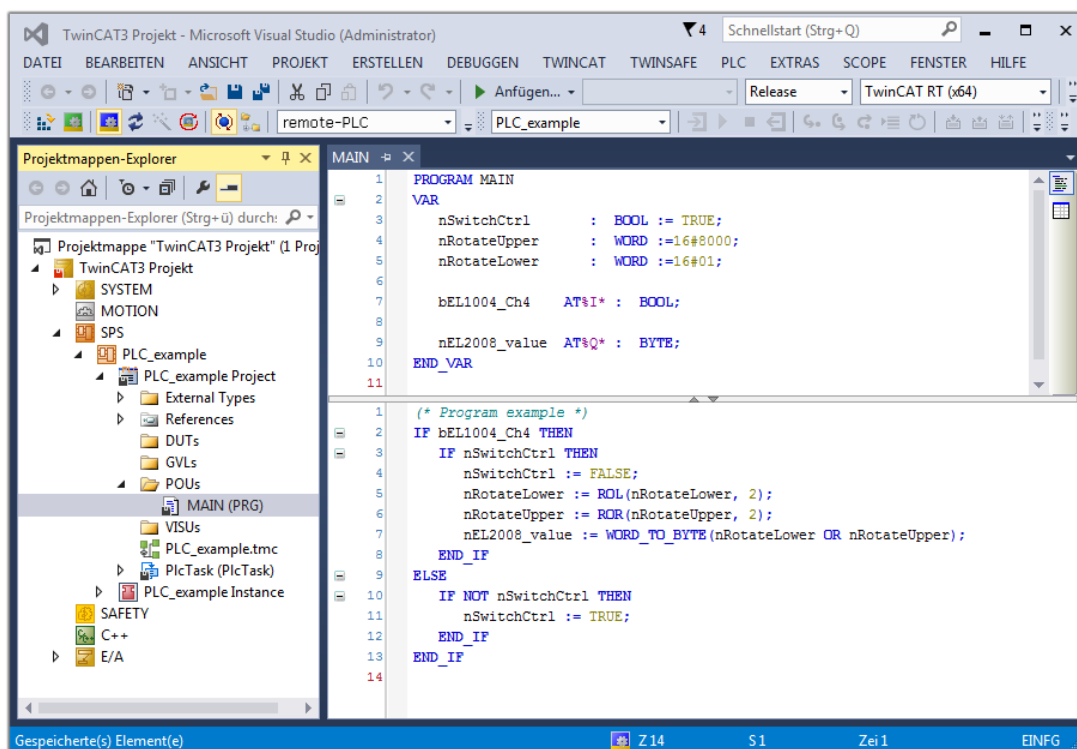
TwinCAT PLC Controllin puolella tehdään itse ohjelma, jota PLC suorittaa. PLC Control tukee kaikkia ohjelmointikieliä, jotka ovat määritelty IEC 61131-3-standardissa.

Tekstipohjaisia kieliä ovat Structured Text (ST) ja Instruction List (IL). Graafisiin kieliin kuuluu Function Block Diagram (FBD), Ladder Diagram (LD) ja Sequential Function Chart (SFC). Kun ohjelma on saatu luotua ja tallennettua ilman virheitä, voidaan PLC Controllin luoma TPY-tiedosto linkata System Managerin PLC-konfiguraatioon. System Manager lukee tiedostosta I/O-muuttujat osoitepaikkoineen. (Beckhoff Information System, n.d.-g)

## 4.2 TwinCAT 3

Vuonna 2010 julkaistun TwinCAT 3:n yksi keskeisistä lähestymistavoista on yksinkertaistaa ohjelmistosuunnittelua. TwinCAT 3:n isona etuna on TwinCAT PLC Controllin integrointi suoraan TwinCAT System Manageriin. Tämän ansiosta kehittäjä tarvitsee vain yhden kehitysympäristön ohjelman tekemiseen (kuva 4). Tämän avulla säästytään kahden eri kehitysympäristön välillä hyppimiseltä ja yksinkertaistaa työkalun oppimista. (Beckhoff Information System, n.d.-h)

Kuva 4. TwinCAT 3-kehitysympäristö (Beckhoff Information System, n.d.-i)



TwinCAT 3 eXtended Automation Engineering (XAE) ohjelmistoarkkitehtuuri tukee modulaarisempaa ohjelmointia. Jotta nykyaikaisten koneiden monimutkaisuutta voitaisiin hallita ja vähentää tarvittavaa suunnittelutyötä, siirrytään kohti modulaarisempaa ohjausohjelmistoa. Tämä johtaa siihen, että yksittäisiä toimintoja ja kokoonpanoja tarkastellaan moduuleina. Moduulien tulisi olla mahdollisimman itsenäisiä ja hierarkkisesti jäsenneltyjä. Rakenteen tulisi olla järjestetty niin, että hierarkiassa alimmalla tasolla olevat moduulit ovat yksinkertaisimpia ja uudelleenkäytettävissä olevia peruselementtejä. Vakiintuneet rajapinnat mahdollistavat sen, että ylempään tason moduuleista voidaan yhdistellä monimutkaisempia konemoduuleja aina kokonaisuksi koneisiin asti. Ihanteellisesti yksittäiset moduulit voitaisiin ottaa käyttöön, laajentaa, skaalata ja käyttää uudelleen toisistaan riippumattomasti. (Beckhoff Information System, n.d.-h)

Modulaarisuutta voidaan toteuttaa esimerkiksi olio-ohjelmoinnin avulla. Olio-ohjelmoinnissa ohjelmisto jaetaan olioihin. Kaikki kyseiseen olioon liittyvät kuvaukset yhdistetään yhteen elementtiin, esimerkiksi funktiolohkoon. Kuvauksiin sisältyvät olion tiedot ja siihen liittyvät toiminnot. Lisäksi voidaan määrittellä rajapinta metodien ja ominaisuuksien avulla. Olio-ohjelmoinnin ansiosta voidaan kehittää olioita, joita voidaan käyttää uudelleen itsenäisesti ilman erityisiä ehtoja. Elementtejä voidaan hyödyntää sellaisenaan yhdessä tai useammassa sovelluksessa. (Beckhoff Information System, n.d.-j)

Olio-ohjelmointimenetelmä tarjoaa monia etuja. Jakamalla ohjelma olioihin, voidaan kehittää selkeä ja hyvin jäsennelty rakenne. Näin mahdollistetaan rakenteen yksittäisten osien helppo ymmärrettävyys ja laajennettavuus. Olioiden uudelleenkäytettävyys säästää aikaa ja kustannuksia sovellusten kehityksessä ja ylläpidossa. (Beckhoff Information System, n.d.-j)

- **Kapselointi (encapsulation):** Kapseloinnin periaatteen mukaan kaikki olion tärkeä tieto on sen sisällä ja vain valitut tiedot ovat ulkopuolelle näkyvissä. Jokaisen olion toteutus ja tila pidetään yksityisinä sille määritellyssä luokassa. Muiden olioiden suora pääsy tähän luokkaan tai oikeus tehdä muutoksia on estetty. Muut oliot voivat kutsua luokan tarjoamia julkisia funktioita tai metodeja. Tietojen piilottaminen parantaa ohjelman turvallisuutta ja estää tietojen korruptoitumisen.
- **Abstraktio (abstraction):** Abstraktiolla tarkoitetaan sitä, että oliot paljastavat vain ne sisäiset mekanismit, jotka ovat tarpeellisia muille olioille. Johdettu (peritty) luokka voi laajentaa toiminnallisuuttaan. Tämä konsepti helpottaa kehittäjää tekemään lisäyksiä ja muutoksia ohjelmaan ajan myötä.
- **Periytyminen (inheritance):** Periytyminen on sitä, että luokat voivat hyödyntää uudelleen koodia ja ominaisuuksia muista luokista. Eri olioiden välille voidaan määritellä suhteita ja aliluokkia. Periytyminen vaatii perusteellisempaa tietojen analysointia. Periytymisen avulla kehitysaikaa ja tarkkuutta voidaan parantaa.
- **Monimuotoisuus (polymorphism):** Monimuotoisuudella tarkoitetaan olioiden kykyä jakaa käyttäytymismalleja ja saada useita eri muotoja. Tämän avulla voidaan vähentää tarpeettoman koodin toistoa. Aliluokka laajentaa kantaluokan toiminnallisuutta. Monimuotoisuuden avulla mahdollistetaan erilaisten olioiden käyttäminen saman rajapinnan kautta.
- **Syntaksi (syntax):** Syntaksi on sääntökokoelma, joka määrittelee sanojen ja välimerkkien järjestyksen ohjelmointikielessä.
- **Kytkenä (coupling):** Kytkenällä kuvataan, kuinka vahvasti eri ohjelmaelementit ovat yhteydessä toisiinsa. Esimerkiksi jos luokan ominaisuudet muuttuvat toisen luokan ollessa siihen sidottuna, myös tämän sidotun luokan ominaisuudet muuttuvat.
- **Assosiaatio (association):** Assosiaatio tarkoittaa yhteyttä yhden tai useamman luokan välillä. Assosiaatiot voivat olla yksi yhteen (1:1), yksi moneen (1:M), monta yhteen (M:1) tai monta moneen (M:M). (Gillis, n.d.)

TwinCAT 3 eXtended Automation Runtime (XAR) tarjoaa reaaliaikaisen käyttöympäristön, jossa TwinCAT-moduuleja voidaan ladata, suorittaa ja hallita. Yksittäisten moduulien ei tarvitse olla tehty samalla kääntäjällä, joten ne voidaan ohjelmoida toisistaan riippumattomina. Useita taskeja voi pyöriä yhdellä PC:llä. Koska eri moduulit esimerkiksi

SPS, C/C++ tai MATLAB voivat kutsua toisiaan TwinCAT 3 XAR-ympäristössä, ohjelmistoarkkitehtuurin mahdollisuudet laajenevat huomattavasti. Tämän ansiosta itsenäisiä toimintoja sisältäviä moduuleja voidaan yhdistää yhdeksi kokonaiseksi koneratkaisuksi. Taskien kautta kutsuttavien moduulien määrä ei ole rajoitettu, mutta lopullinen rajoitus riippuu kyseisen järjestelmän resurssien riittävydestä. TwinCAT 3 tukee moniydinsuorittimia. Yksittäisiä taskeja voidaan jakaa eri suorittimen ytimille. Tämän ansiosta uusimpien teollisuuteen tarkoitettujen PC-laitteiden suorituskykyä voidaan hyödyntää paremmin. (Beckhoff Information System, n.d.-h)

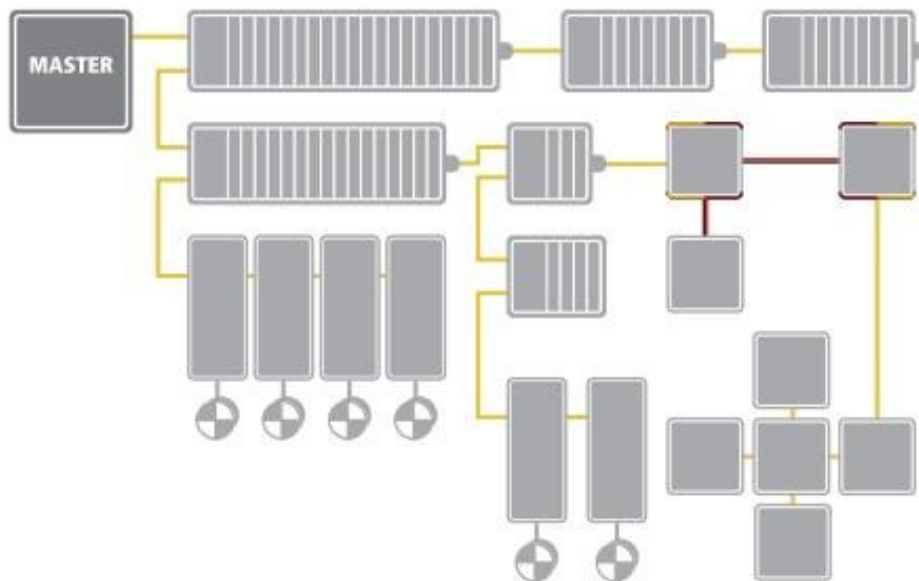
### 4.3 EtherCAT-kenttäväyläprotokolla

Beckhoffin kehittämä kenttäväyläprotokolla EtherCAT (Ethernet for Control Automation Technology) yhdistää Ethernetin hyödyt ja perinteisten kenttäväyliä yksinkertaisuuden. EtherCAT ja Safety over EtherCAT ovat kansainvälisiä IEC-standardeja (IEC 61158 ja IEC 61784). Protokollaa voi käyttää kuka tahansa yhteensopivassa muodossa. EtherCAT Technology Group on vuonna 2003 perustettu yhtiö, joka on vastuussa EtherCAT-protokollan kehittämisestä ja ylläpitämisestä. EtherCAT mahdollistaa joustavan kuvan 5 mukaisen kenttäväylätopologian, johon voidaan liittää 65535 laitetta. (Beckhoff, n.d.-b)

EtherCAT käyttää tavallisia Ethernet-datapaketteja suorittaakseen viestintää kaikkien kenttäväylään liitettyjen laitteiden välillä. Ohjaus tapahtuu master-slave-järjestelmällä. EtherCAT-master lähettää tavoitetietoa slave-laitteille ja vastaanottaa niiltä todellista tietoa. Tavoitetieto voi olla esimerkiksi ohjauskomentoja toimilaitteille. Todellinen tieto voi olla puolestaan esimerkiksi mittauslaitteiden keräämää tietoa, esimerkiksi nopeutta tai lämpötilaa. (Kithara Software, n.d.)

EtherCAT-master kokoaa datagrammit ja lähettää ne kaikkien väylään kytkettyjen slave-laitteiden läpi. Datagrammien kulkiessa slave-laitteiden läpi, kukin laite ottaa siitä oman tavoitetietonsa ja antaa samalla oman todellisen tietonsa. Kaikki muu tieto, joka ei kuulu kyseiselle slave-laitteelle, viedään eteenpäin muuttumattomina. Kun datagrammi on kulkenut kaikkien slave-laitteiden läpi, se palaa takaisin EtherCAT-masterille. Tämä prosessi on erittäin nopea, vaikka väylään olisi liitettynä paljon laitteita, koska yhden slave-laitteen läpimenoviive on vain muutamia nanosekunteja. (Kithara Software, n.d.)

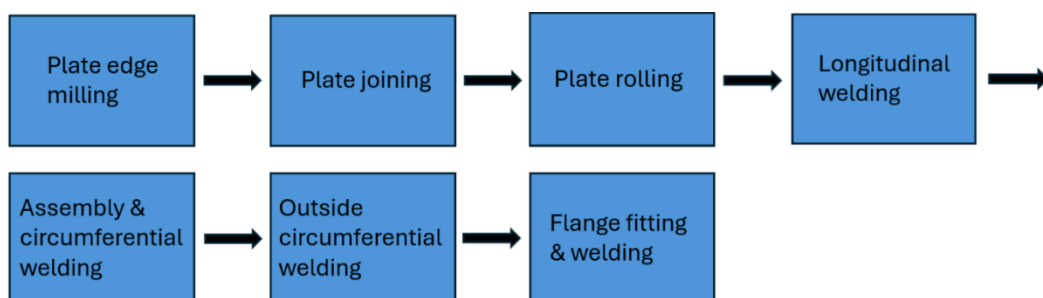
Kuva 5. EtherCAT-kenttäväylätopologia (EtherCAT Technology Group, n.d.)



## 5 Tuulivoimalan runkoputken valmistusprosessi

Tuulivoimalan runkoputken valmistus voidaan pääpiirteittäin jakaa seitsemään vaiheeseen. Alla olevassa kuvassa 6 on esitetty tuulivoimalan runkoputken valmistuksen päävaiheet. Tuulivoimalan runkoputken valmistusprosessi ja siihen suunnitellut laitteet voivat hieman poiketa toisistaan eri valmistajien välillä, riippuen millainen on juuri heidän tarpeensa.

Kuva 6. Tuulivoimalan runkoputken valmistuksen päävaiheet

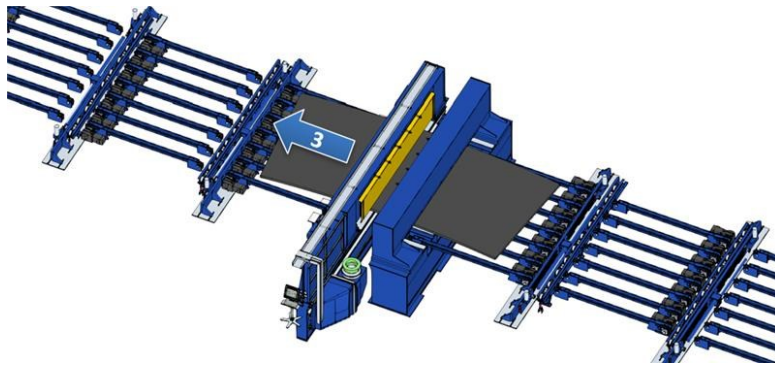


### 5.1 Levyn reunan jyrshintä

Tuulivoimalan runkoputken valmistus alkaa suorista metallilevyistä, jotka leikataan sopivan kokoisiksi. Levyn leikkaamisen jälkeen ne viedään jyrsimelle (plate edge milling). Levyn päät jyrsitään sopivan hitsausrailon aikaansaamiseksi. Levyn reunan jyrshintä tapahtuu esimerkiksi kuvan 7 EBS-jyrsimellä. Levyn reuna tuodaan kuljettimen avulla koneelle ja

asetetaan oikeaan kohtaan. Tämän jälkeen jyrsimen puristimet painetaan kiinni, jotta levy pysyy mittauksen ja jyrsinnän ajan paikallaan. Jyrsimelle syötetään tiedot levystä ja näiden annettujen tietojen avulla jyrsin hoitaa ensiksi mittauksen levyille. Kun mittaustyökierto on tehty onnistuneesti, voidaan aloittaa jyrsintä. (Pemamek Oy, henkilökohtainen tiedonanto, n.d.)

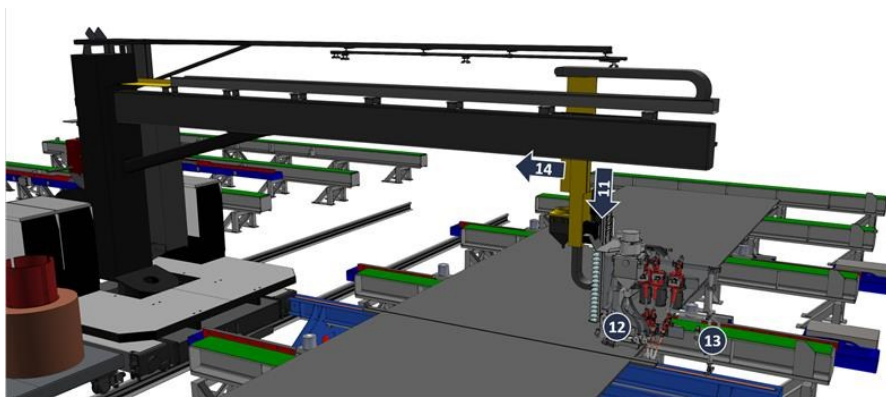
Kuva 7. Levyjyrsin EBS (Pemamek Oy tietopankki, n.d.)



## 5.2 Levyjen hitsaus

Kun levy on jyrsitty, viedään se kuljetinta pitkin levyn hitsausasemalle (plate joining). Levyjä voidaan joutua esitaivuttamaan esitaivutuskoneilla, jotka ovat kuljetinlinjastossa. Levyjä hitsataan yhteen, jotta saadaan tavoiteltu runkoputken halkaisija. Ensiksi levyt asetetaan kohdakkain kuljettimen ja sivusiirtimien avulla. Kun levyt on saatu kohdakkain, voidaan aloittaa hitsausprosessi. Jokaisen hitsaustyökierron jälkeen hitsauspalon päälle jää kuonakerros, joka tulee kaapia pois. Levyt voivat olla jopa 150 mm paksuja, joten hitsauspaloja tulee useita. Alla olevassa kuvassa 8 on levynhitsausasema.

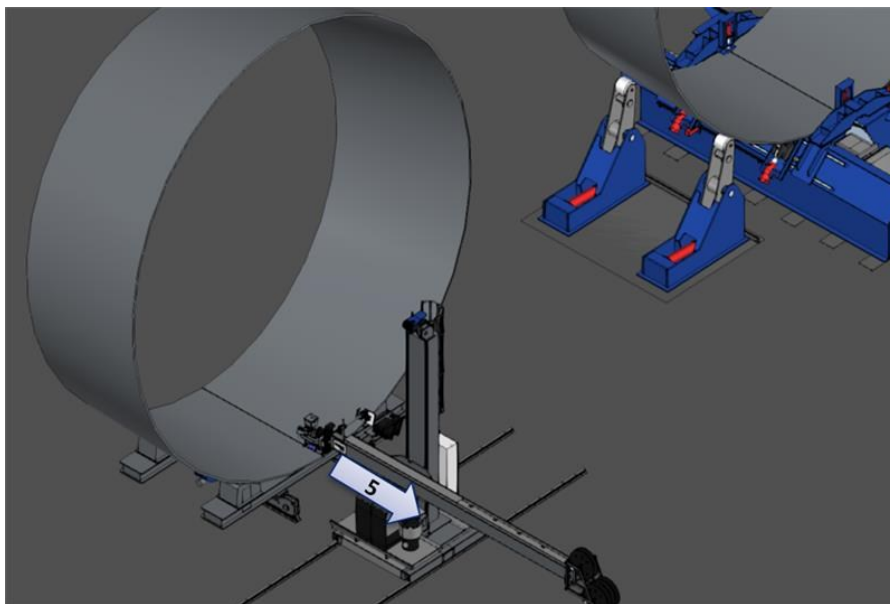
Kuva 8. Levynhitsausasema EHD-R (Pemamek Oy tietopankki, n.d.)



### 5.3 Pituussauman hitsaus

Kun metallilevyjä on hitsattu sopiva määrä yhteen tavoitellun halkaisijan saavuttamiseksi, seuraavaksi levy viedään mankeliin, joka mankeloi levystä pyöreän työkappaleen. Kun työkappale on mankeloitu, viedään se pituussauman hitsausasemalle, jossa työkappaleen päät hitsataan yhteen (longitudinal welding). Työkappale lasketaan hitsausaseman rullastojen päälle (kuva 9), jonka jälkeen pituussauma kohdistetaan. Kun sauma on saatu kohdistettua oikeaan kohtaan, voidaan aloittaa hitsaus. Kun kaikki hitsauspalot on saatu hitsattua, voidaan työkappale siirtää pituussauman jyrsimelle.

Kuva 9. LW-asema ja pituussaumajyrsin LM55 (Pemamek Oy tietopankki, n.d.)



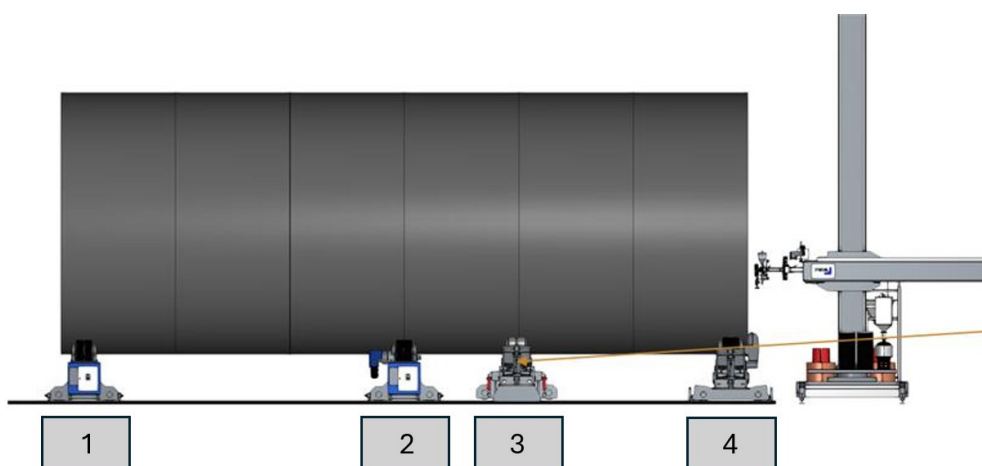
Pituussaumajyrsin jyrsii työkappaleen pituussauman ulkopuolelta, jotta saadaan haluttu V-railo hitsausta varten. Jyrsittävä työkappale painetaan kiinni hydraulisten puristimien avulla jyrtimeen. Jyrsimen terä säädetään kohdalleen kameran ja laserin avulla. Kun jyrsin on saatu säädettyä sauman kanssa kohdakkain, voidaan aloittaa työkappaleen jyrshintä. Kun jyrshintä on valmis, työkappale siirretään takaisin rullastojen päälle, jossa jyrshinty sauma hitsataan ulkopuolelta. (Pemamek Oy, henkilökohtainen tiedonanto, n.d.)

## 5.4 Koontiasema ja kehäsauman hitsaus sisäpuolelta

Kun työkappaleiden pituussaumat on saatu hitsattua, työkappaleet tuodaan koontiasemalle, jossa työkappaleiden väliset kehäsaumat hitsataan sisäpuolelta (assembly & circumferential welding). Yhteen hitsatuista työkappaleista muodostuu tuulivoimalan runkoputki. Kuvassa 10 on esitetty koontiasema. Koontiasema koostuu hitsaustornista ja neljästä rullastosta:

1. Anti-creep-rullastosta, joka valvoo, että runkoputki pysyy paikallaan pyöriksen aikana. Anti-creep-rullastossa on laseretäisyysanturi, joka valvoo runkoputken reunaa. Jos runkoputken reuna liikkuu, niin saadaan myös tieto kumpaan suuntaan. Analogisen signaalin avulla rullaston telejä säädetään hydraulisesti, jotta runkoputki saadaan pysymään paikallaan. Runkoputken paikallaan pysyminen on tärkeää, jotta hitsattava railo ei liiku hitsauksen aikana.
2. Power-rullasto, jossa on runkoputken taajuusmuuttajaohjatut pyöritysmoottorit ja kiskoliikemoottorit. Moottoreiden avulla runkoputkea pyöritetään ja siirretään. Power-rullastoon yhdistetään Jig- ja Fit-up-rullasto.
3. Jig-rullastosta, jonka päälle runkoputken ja siihen hitsattavan uuden työkappaleen reunat tuodaan ja asetetaan kohdakkain erilaisten hydraulisten liikkeiden avulla. Jig-rullastossa on lisäksi taajuusmuuttajaohjatut kiskoliikemoottorit rullaston liikuttamista varten.
4. Fit-up-rullasto, jonka hydraulisten liikkeiden avulla työkappale saadaan suoraan runkoputkeen nähden. Fit-up-rullastoa voidaan lisäksi siirtää kuten Jig-rullastoa.

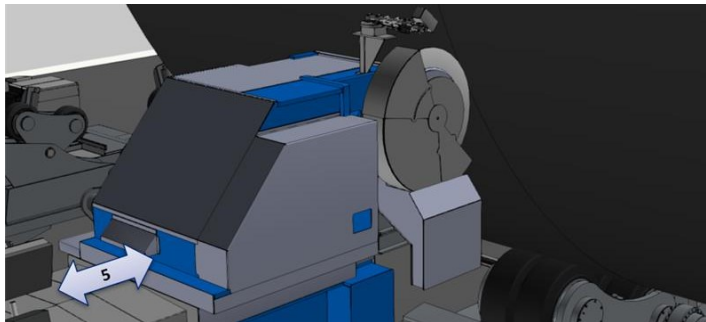
Kuva 10. Koontiasema (Pemamek Oy tietopankki, n.d.)



## 5.5 Kehäsauman hitsaus ulkopuolelta

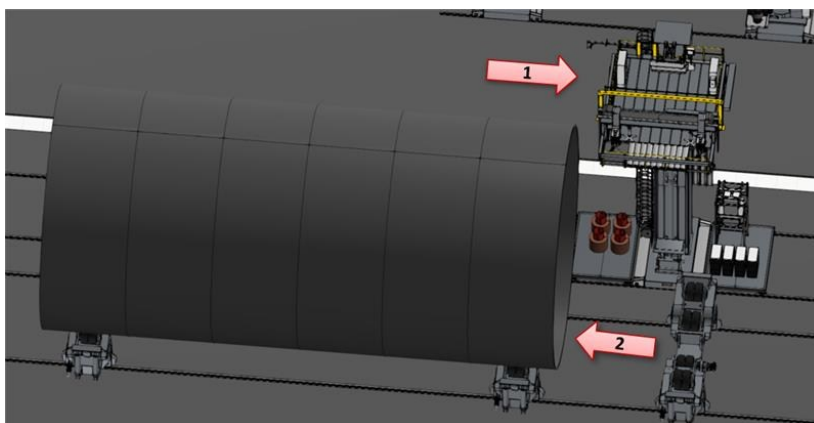
Kun runkoputken kehäsaumat on saatu hitsattua sisäpuolelta, siirretään runkoputki CM55 kehäsaumajyrsimelle (kuva 11). Kehäsaumajyrsin jyrsii kehäsaumat auki ulkopuolelta, jotta saadaan haluttu V-railo hitsausta varten. Jyrsin asetetaan kehäsauman kohdalle kameran ja laserin avulla. Kun jyrsimen terä on oikeassa kohdassa, tarkastetaan parametrit ja käynnistetään automaattinen jyrsintä. (Pemamek Oy, henkilökohtainen tiedonanto, n.d.)

Kuva 11. Kehäsaumajyrsin CM55 (Pemamek Oy tietopankki, n.d.)



Kun kehäsauman jyrsintä on valmis, viedään runkoputki rullastojen päällä kuvan 12 mukaiselle WS2-kehäsauman hitsausasemalle (outside circumferential welding). WS2-hitsausasemassa on kaksi hitsauspäättä, joten sillä voidaan hitsata kahta kehäsaumaa samaan aikaan nopeuttaen hitsausprosessia huomattavasti. WS2-hitsausaseman tasoa voidaan nostaa ja laskea hitsattavan runkoputken halkaisijan mukaan. Hitsausaseman tason nosto ja lasku tapahtuu hydraulisesti. WS2 hitsausasemalla voidaan hitsata kehäsaumaa ilman, että hitsausprosessi katkaistaan hitsauspalojen välissä, eli kehäsauman railo saadaan halutessaan hitsattua umpeen ilman pysäytyksiä.

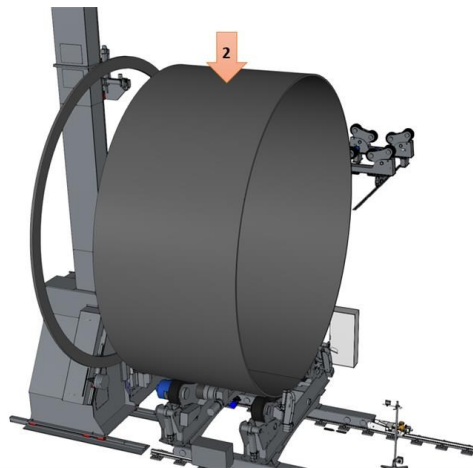
Kuva 12. WS2-hitsausasema (Pemamek Oy tietopankki, n.d.)



## 5.6 Laipan asennus

Viimeiseen työkappaleeseen hitsataan asennuslaippa laipan asennusasemalle (kuva 13) ennen sen hitsaamista runkoputkeen. Ensiksi laippa asetellaan asemaan paikalleen hydraulisten liikkeiden avulla, jonka jälkeen työkappale tuodaan laippaa vasten rullastojen avulla. Kun työkappale on saatu aseteltua oikeaan kohtaan laippaa, se hitsataan kiinni. Riippuen asiakkaasta laippa voidaan joko pelkästään pistehitsata laipan asennusasemalla tai asemalla voi olla oma hitsaustorni, joka on varusteltu pitkillä polttimilla railon umpeen hitsaamiseksi. Kun laippa on saatu asennettua, voidaan työkappale nostaa pois asemalta ja hitsata kiinni runkoputkeen. Tämän jälkeen tuulivoimalan runkoputki on valmis jatkokäsittelyä varten. (Pemamek Oy, henkilökohtainen tiedonanto, n.d.)

Kuva 13. Laipan asennusasema (Pemamek Oy, tietopankki, n.d.)



## 6 Hitsaustorni

Hitsaustorni on oleellinen osa tuulivoimalan runkoputken valmistusprosessia. Hitsaustornilla saadaan aikaan tasalaatuista hitsausseamaa ja sen avulla paksunkin levyin sauma saadaan täytettyä huomattavasti nopeammin kuin käsin hitsaamalla. Hitsaustornin avulla työkappaleen sauma saadaan hitsattua sisä- ja ulkopuolelta. Yleisimpiä hitsausprosesseja, joita käytetään hitsaustornissa ovat jauhekaarihitsaus (submerged arc welding, SAW) ja GMAW (gas metal arc welding). Jauhekaarihitsaus on kuitenkin näistä kahdesta hitsausprosessista yleisempi, sillä se mahdollistaa nopeamman sauman täytön, joka korostuu etenkin paksuja levyjä hitsatessa. Teollisuuden yleisimpiä käyttökohteita hitsaustornille ovat tuulivoima- ja telakkateollisuus.

Pemamek Oy:n hitsaustorneista on saatavilla kolmea eri tuotemallia; MD (Medium Duty), HD (Heavy Duty) ja EHD (Extra Heavy Duty). MD-torneissa on useita eroavaisuuksia

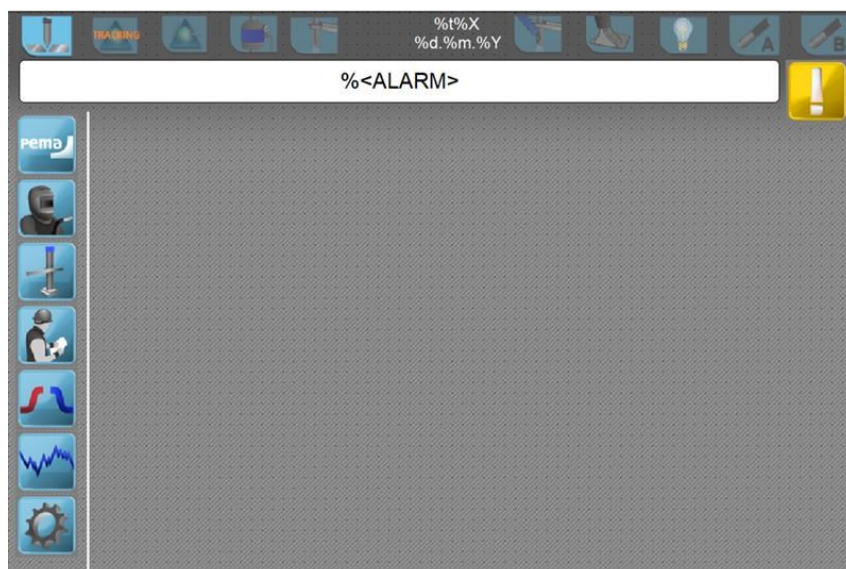
mahdollisten varustusten osalta verrattuna HD- ja EHD-torneihin. MD-tornia saa ainoastaan WC100-hitsauksen ohjausjärjestelmällä, joka ei mahdollista kaikkia toimintoja. HD- ja EHD-tornissa on pitkälti samat mahdolliset varustukset, poikkeuksena EHD-torneille ominaisena vaakapuomin pystyliike kahdella nostomoottorilla. HD- ja EHD-torneihin on saatavilla WC500-hitsauksen ohjausjärjestelmä, joka mahdollistaa lisäominaisuuksia WC100 verrattuna.

Hitsaustorniin voidaan myös yhdistää rullastoja tai pyörityspöytiä, jolloin kannun pyörittämistä voidaan hoitaa tornista käsin. Tämä helpottaa operaattorin työtä, sillä jos esimerkiksi kannua hitsataan ulkoa, operaattori voi olla useiden metrien korkeudessa tornin puomin päässä operaattorituolissa valvomassa hitsausprosessia. Torni saa myös rullaston enkooderilta tiedon kannun pyörimisestä, jonka avulla torni tietää milloin kierros kehäsaumaa on hitsattu. Tämä mahdollistaa automaattisen hitsauspalon vaihdon.

## 6.1 WC100- ja WC500-hitsauksen ohjausjärjestelmä

WC100 on hitsauksen ohjausjärjestelmä, jota käytetään MD-torneissa ja osassa HD- ja EHD-torneista. WC100-pohjaisissa hitsaustorneissa käytetään Beckhoff CP7711-näyttöpaneelia, jossa on PLC. Näyttöpaneelissa on Windows CE-käyttöjärjestelmä. WC100-käyttöliittymä (kuva 14) on tehty TwinCAT 2 Visualization-työkalulla. Käyttöliittymästä voidaan mm. muuttaa hitsausparametreja, määrittää hitsauksessa käytettäviä liikeakseleita ja nähdä hitsausdataa. WC100-hitsauksen ohjausjärjestelmän kanssa voidaan käyttää railonseurantaan optista korkeuden seuranta tai GMD-seurantasormeaa.

Kuva 14. WC100-käyttöliittymä



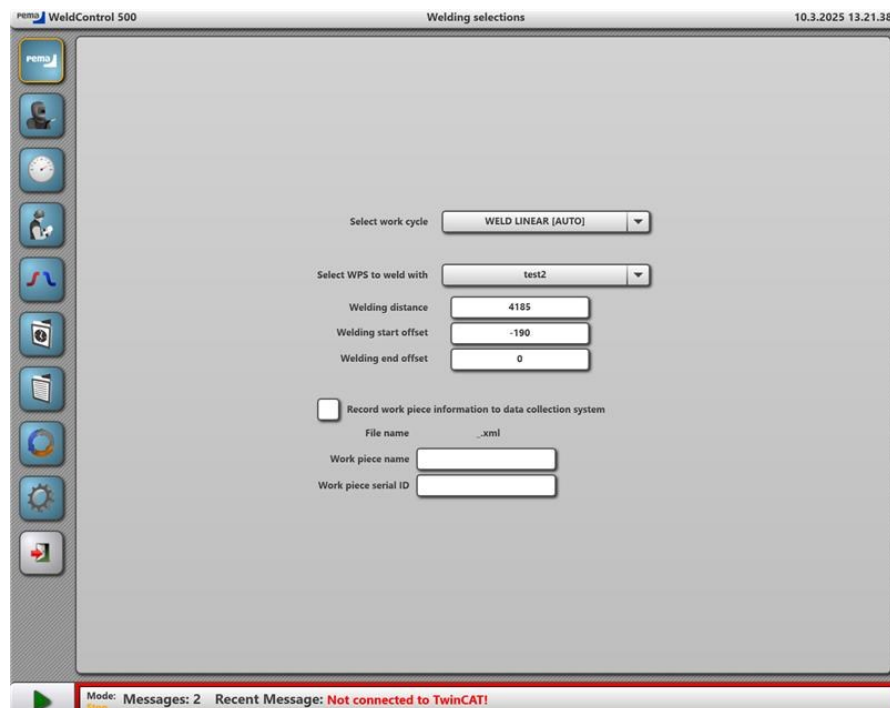
WC500-hitsauksen ohjausjärjestelmällä mahdollistetaan lisäominaisuuksia vaativampiin käyttötarkoituksiin. WC500 on Pemamek Oy:n kehittelemä ohjelmisto, joka kommunikoi PLC:n kanssa ADS-väylän yli. WC500-järjestelmän XML-tiedostoihin tehdään konfigurointi hitsaustornin ominaisuuksien mukaan. WC500-pohjaisissa hitsaustorneissa käytetään Beckhoff CP7213-näyttöpaneelia, jossa on PLC. Näyttöpaneelissa on Windows 10-käyttöjärjestelmä.

WC500 ominaisuuksiin kuuluu laserrailonseurantajärjestelmä, joka on toteutettu Quelltech Q4-laserskannereilla. Laserskannereille on tornin pääkaapissa oma PC, johon on asennettu Pemamek Oy:n kehittämä SeamTracking-ohjelmisto. Laserskannereiden PC:n ja tornin PLC:n välille luodaan ADS-väylä, jonka yli laitteet kommunikoivat.

Laserskannereiden havaitseman railon pistetiedot lähetetään logiikalle. Pistetietojen avulla WC500 osaa piirtää railon käyttöliittymään ja tehdä korjausliikkeet poltinta liikuttaville liikeakseleille.

Lisäksi WC500 mahdollistaa muita ominaisuuksia, joita ei voida toteuttaa WC100-järjestelmän kanssa. Näitä ovat mm. polttimien kallistus, automaattinen hitsauspalon vaihto hitsatessa ja adaptiivinen hitsaus, jonka avulla hitsaustorni voi tehdä pieniä muutoksia asetettuihin hitsausparametreihin hitsauksen aikana. Alla olevasta kuvan 15 WC500-käyttöliittymästä löytyy edistyneempien lisäominaisuuksien lisäksi samat perustoiminnot kuin WC100-käyttöliittymästä.

Kuva 15. WC500-käyttöliittymä



## 6.2 Hitsausvarustus

Jauhekaarihitsausvarustuksesta on saatavilla useampaa eri variaatiota. Polttimia on saatavilla 1–3 kappaletta ja hitsausvirtalähteiden määrä riippuu polttimien määrästä. GMAW-hitsausvarustuksessa on yksi poltin. Yritys käyttää Lincoln Electric-hitsausvirtalähteitä ja niihin liittyviä oheislaitteita. Hitsausvirtalähteiden kanssa kommunikoidaan DeviceNet-kenttäväylän yli. Hitsausvirtalähteille tuodaan tornin pääkaapista Ethernet-kaapeli, jotta virtalähteitä voidaan parametroida ilman, että tietokoneen täytyy olla suoraan kiinni virtalähteessä. Hitsausvirtalähteiden hitsauksen aikaisia arvoja ja muuta diagnostiikkaa voidaan tarkastella Power Wave Manager-ohjelmiston avulla.

Jos hitsaustornissa on jauhekaarihitsausvarustus, tarvitaan lisäksi jauhepainetankki ja hitsauslankakelat tornin tasolle tai erilliseen kiskovaunuun. Hitsauspäässä on pienempi jauhesäiliö, johon saadaan imuroitua palamaton jauhe ja se voidaan käyttää uudelleen. Lankakeloille tarvitaan langansyöttimet, jotka syöttävät lankaa polttimille. Päälangansyötin on hitsauspäässä polttimen yhteydessä. Lisäksi voi olla apulangansyötin hitsaustornin tasolla, jonka avulla saadaan uusi lanka syötettyä hitsauspäähän. Hitsaustornin vaaka- tai pystypalkkiin asennetaan langansyöttökontrolleri, johon tuodaan hitsausvirtalähteeltä kontrollikaapeli. Hitsausvirtalähde ohjaa langansyöttimiä kontrollerin kautta. Jauhekaarihitsauksessa voidaan käyttää kaikkia tarjolla olevia railonseurantamenetelmiä. Jauhekaarihitsauksen kanssa voi olla käytössä WC100- tai WC500-hitsauksen ohjausjärjestelmä. Jauhekaarihitsausvarustuksen kanssa usein on lisänä kamera ja näyttö hitsauspäässä, joiden avulla polttimien asettelu hitsausrailoon helpottuu.

Jos hitsaustornissa on GMAW-hitsausvarustus, niin polttimia on hitsauspäässä yksi. Hitsausvirtalähteeltä tuodaan kontrollikaapeli kontrollerille, jossa on itsessään langansyötin. Lankakela tulee lähelle hitsauspäätä kontrollerin viereen. Koska hitsausvirtalähde on vesijäähdytetty, virtalähteelle tarvitaan lisäksi vesijäähdytin. Lisäksi hitsauspäässä voi olla prosessikamera ja näyttö, joista nähdään reaaliaikaisesti hitsausprosessi. GMAW-hitsausvarustuksen kanssa voidaan käyttää optista korkeudensäättöä tai GMD-seurantasormeaa. Hitsauksen ohjausjärjestelmänä toimii WC100.

## 6.3 Railonseuranta

Hitsaustornilla hitsatessa railonseurantajärjestelmä helpottaa hyvän ja tasalaatuisen hitsaussauman aikaansaamista. Railonseurannan avulla hitsaustorni osaa tehdä korjaavia liikkeitä hitsauksen aikana. Tämä helpottaa ja nopeuttaa hitsausprosessia huomattavasti, kun esimerkiksi suoria levyjä hitsatessa levyjen välisen sauman ei tarvitse olla tismalleen suorassa linjassa torniin nähden. Pemamek Oy:n hitsaustorniin on saatavilla kolmea erilaista railonseurantajärjestelmää.

Laserrailonseuranta on toteutettu kahdella Quelltech Q4-laserskannerilla. Kun käytössä on laserrailonseuranta, on hitsauksen ohjausjärjestelmän oltava WC500. Skannerit kalibroidaan niin, että ne skannaavat samaa kohtaa railossa. Laserrailonseurantaan käytetään kahta skanneria, koska kumpikin skanneri näkee puolet hitsausrailosta. Skannerit kommunikoivat Ethernetin yli, joten niille on oma PC hitsaustornin pääkaapissa. Pääkaapissa olevaan PC:hen asennetaan SeamTracking-ohjelmisto, jonka avulla skannerit kalibroidaan ja parametroidaan. Laserskannereiden antaman datan ja WC500-käyttöliittymään määriteltyjen parametrien perusteella torni korjaa polttimien sijaintia railossa hitsauksen aikana.

GMD-seurantasormi on sähkömekaaninen anturi, joka vaatii fyysisen kosketuksen hitsausrailoon. GMD-seurantasormelle määritetään ennen hitsausta referenssipaiikka, jossa se pyrkii pysymään. Seurantasormi antaa analogista jännitesignaalia sen mukaan, kuinka paljon sormen asento muuttuu hitsauksen aikana. Tällä jännitesignaalilla ohjataan hitsauspään ristiluistin liikkeitä hitsauksen aikana. Operaattori voi itse ohjauspaneelistä säätää railonseurannan herkkyyttä. Jos hitsaustornissa railonseurantamenetelmänä on GMD-seurantasormi, niin hitsauksen ohjausjärjestelmänä on tyypillisesti WC100, mutta tämän railonseurantatavan käyttö onnistuu myös WC500-järjestelmällä.

Optinen korkeuden seuranta toteutetaan Keyence IL300-laseranturilla. Laseranturi valvoo korkeutta railon pohjasta. Ennen hitsauksen aloittamista laserille määritetään referenssikorkeus, jossa sen tulisi pysyä. Tällä railonseurantajärjestelmällä ei voida tehdä korjausliikkeitä vaakasuunnassa, eli operaattorin tulee tarvittaessa korjata polttimien sijaintia hitsauksen aikana. Tämän railonseurantatavan kanssa on käytössä aina WC100.

## 6.4 Liikkeet

Hitsaustornin vakioliikeakseleihin kuuluu vaakapuomin vaaka- ja pystyliike, sekä hitsauspäässä olevat ristiluistin vaaka- ja pystyliike. Hitsaustornin vaakapuomin vaakaliike on voitu toteuttaa joko servomootorilla tai taajuusmuuttajaohjatulla oikosulkumootorilla. Vaakapuomin liike toteutetaan yleisemmin servomootorilla, jos tornilla on tarkoitus hitsata kannun pituussaumaa. Pituussauman hitsauksessa yhtenä liikeakselina käytetään vaakapuomin vaakaliikettä, jolloin tarvitaan puomin paikkatieto ja tarkka nopeus. Vaakapuomin pystyliike toteutetaan aina taajuusmuuttajaohjatulla oikosulkumootorilla, jossa riippuen hitsauksen ohjausjärjestelmästä voidaan tarvita enkooderi. Jos hitsaustorni toteutetaan WC500-hitsauksen ohjausjärjestelmällä, silloin enkooderi on pakollinen. Pystyliikkeelle saadaan myös toteutettua softarajat tässä tapauksessa.

Hitsauspäässä olevan ristiluistin vaaka- ja pystyliike on voitu toteuttaa joko 24 V DC-moottoreilla tai servomoottoreilla. Jos hitsaustorni on WC100-pohjainen, silloin ristiluistin liikkeitä ohjataan DC-moottoreilla. Jos taas torni on WC500-pohjainen, niin ristiluistin liikkeet ovat servo-ohjatut. Hitsauspäässä voi lisäksi olla polttimien kallistusliikkeet, jotka ovat saatavilla WC500-hitsauksen ohjausjärjestelmän kanssa. Kallistuksen liikkeisiin kuuluu itse polttimien kallistus sekä vaakasuuntainen offset-liikeakseli, jotta polttimien päät pysyvät samassa pisteessä hitsausrailoon nähden.

Näiden lisäksi torneihin on saatavilla vaakapuomin pyöritys ja kiskoliike. Vaakapuomin pyörityksellä ja hitsaustornin kiskoliikkeellä saadaan lisää joustavuutta prosessiin. Kaikkia isoja liikkeitä, eli vaakapuomin pysty-, vaaka- ja pyöritysliikettä sekä tornin kiskoliikettä voidaan ohjata tornin pääkaapin ovesta.

Hitsaustornin ulkoisiin liikeakseleihin kuuluvat torniin yhdistettävien rullastojen tai pyörityspöytien liikkeet. Hitsaustornin kaapin kyljessä on kolme sync-liitintä, joihin voidaan liittää kolme eri ulkoista laitetta. Rullastojen ja pyörityspöytien pyöritystä voidaan ohjata valitsemalla tornin ohjauspaneelista haluttu ulkoinen akseli. Laitteen tulee olla yhdistetty torniin, jotta ohjaus on mahdollista. Kaikkia ulkoisia liikeakseleita ohjataan omanaan, niitä ei voida ohjata yhtäaikaisesti.

Hitsaustorneihin on saanut hitsauspäähän tulevan induktiivisen esilämmittimen noston ja laskun ohjauksen lisäoptiona, mutta tämä on ollut projektikohtainen erikoisominaisuus eikä yleinen lisäoptio. Jauhekaarihitsauksessa käytettävän jauheen imurointiin tarkoitettu imurin suulake on normaalisti vakioetäisyydellä polttimista, mutta pituussaumaa hitsaaviin torneihin on joissain projekteissa lisätty servo-ohjattu suulake. Tämän avulla hitsaustornin

vaakapuomin ei tarvitse liikkua niin paljoa yli hitsattavan kappaleen reunasta, jotta imusuulake saa imettyä loputkin jauheet hitsausauaman jälkeen. Jos tornissa on GMAW-hitsausprosessi, niin hitsauspäässä sijaitsevalla polttimella on vaaputusliike.

## 6.5 Anturointi

Jos hitsaustornin hitsausprosessina on jauhekaarhitsaus, niin hitsaustornin jauhepainetankissa on yksi kapasitiivinen anturi, joka valvoo jauheen pinnan alarajaa. Jos tornissa on lisätoiminnallisuutena automaattinen jauheentäyttö, on ylärajalle samanlainen anturi. Jauhetankissa on myös usein lämpötila-anturi, jonka avulla voidaan valvoa ja säätää jauheen lämpötilaa painetankissa.

### Isot liikkeet:

- Vaakapuomin vaakaliikkeen päätyrajat on voitu toteuttaa joko induktiivisilla antureilla tai suunnan tuntevalla mekaanisella rajakytkimellä. Suunnan tunteva rajakytkin antaa ajaa liikettä toiseen suuntaan tullessaan rajalle. Jos vaakaliikettä ohjataan servomootorilla, saadaan myös softarajat, jolloin päätyrajat ovat siltä varalta, että jostain syystä softaraja ei pysäytä liikettä.
- Vaakapuomin pystyliikkeen päätyrajat toteutetaan suunnan tuntevalla mekaanisella rajakytkimellä. Jos nostomootorissa on enkooderi, saadaan pystyliikkeelle lisäksi softarajat. Koska vaakapuomin pystyliikkeessä vaakapuomia nostetaan ketjun avulla, on ketjulle oltava ketjurikkoanturi ketjun katkeamisen varalta. Ketjun katketessa vaakapuomi jää kynsien varaan, jotka tarttuvat kiinni pystypuomiin. Kun ketjurikkoanturi havaitsee katkenneen ketjun, hitsaustornin hätäseis aktivoituu.
- Vaakapuomin pyöryksen ja hitsaustornin kiskoliikkeen päätyrajoihin käytetään suunnan tuntevaa mekaanista rajakytkintä.

### Hitsauspää:

- Ristiluistin vaaka- ja pystyliikkeen päätyrajat on toteutettu induktiivisilla antureilla. Jos ristiluistin liikkeet ovat servomootoreilla, saadaan lisäksi softarajat.
- Polttimien kallistuksen päätyrajat on toteutettu induktiivisilla antureilla. Koska kallistuksen liikkeet on tehty servomootoreilla, niin lisäksi saadaan softarajat.
- Jos optiona hitsauspäässä on servo-ohjattu jauheenimusuulake, on liikkeelle magneettikytkimet päätyrajoina. Lisäksi servon ansiosta saadaan softarajat.
- Hitsauspäähän voidaan myös asentaa lisäksi infrapunalämpötila-antureita, joiden avulla nähdään kappaleen lämpötila ohjauspaneeliilta.

## 7 Nykytilanne

WC100- ja WC500-hitsaustorneille on omat ohjelmansa, sillä ne eroavat toisistaan huomattavasti WC500-järjestelmän tarjoamien lisätoiminnallisuuksien vuoksi. Koska WC500-käyttöliittymästä voidaan ohjata hitsaustornin tiettyjä toiminnallisuuksia, niin sille vaaditaan ohjelmaan oma rajapintansa. WC500-rajapintaa ja lisätoiminnallisuuksia lukuun ottamatta hitsaustornien ohjelmarakenne on hyvin samanlainen ja sieltä löytyy pitkälti samat perustoiminnot. Koska kummallekin hitsauksen ohjausjärjestelmälle on oma ohjelmansa, ei koodista ole tullut liian sekava. Jos taas kaikki toiminnallisuudet olisivat samassa koodissa, olisi se paljon raskaampi muokata ja tulkita. Lisäksi WC100-tornissa on rajoituksia esimerkiksi railonseurantajärjestelmien osalta, jonka takia railonseurannan hitsauksen aikaisten korjausliikkeiden toteutus koodissa on tehty eri tavalla WC500-torniin verrattuna.

### 7.1 Uuden projektin luonti

Kun aloitetaan uusi projekti, hitsaustornille on tehty määrittely toiminnallisuuksista ennen ohjelmistosuunnittelua. Tämän määrittelyn ja sähkösuunnittelun laatimien sähkökuvien pohjalta etsitään aikaisemmista projekteista mahdollisimman samoilla toiminnallisuuksilla toteutettu hitsaustorni, jotta muokkausta tulisi mahdollisimman vähän. Kun sopiva pohjaprojekti löydetään, tarkistetaan vastaako I/O-kanavat sähkökuvissa määritellyjä kanavia. Jos projektiin tulee uusia I/O-signaaleja, niille luodaan muuttujat. Jos pohjaprojektissa on ylimääräisiä I/O-signaaleja, jotka ovat selkeästi johonkin tiettyyn projektiin sidottuja, ne yleensä poistetaan.

Jos pohjaprojektista puuttuu tarvittavia toiminnallisuuksia, ne tarvittaessa kopioidaan toisesta projektista. Hitsaustorneja on tehty niin paljon, että harvoin enää tulee uusia optioita, joille ei olisi suunniteltu ohjelmalohkoa. Useimmiten nämä ohjelmalohkot ovat kirjastossa. Tarvittava ohjelmalohko kopioidaan pohjaprojektiin, jolloin ei tehdä uusiksi jo olemassa olevaa toiminnallisuutta. Yrityksen sisäisiä uudelleen käytettäviä funktioita ja funktiolohkoja on pyritty tallentamaan yrityksen omaan kirjastoon.

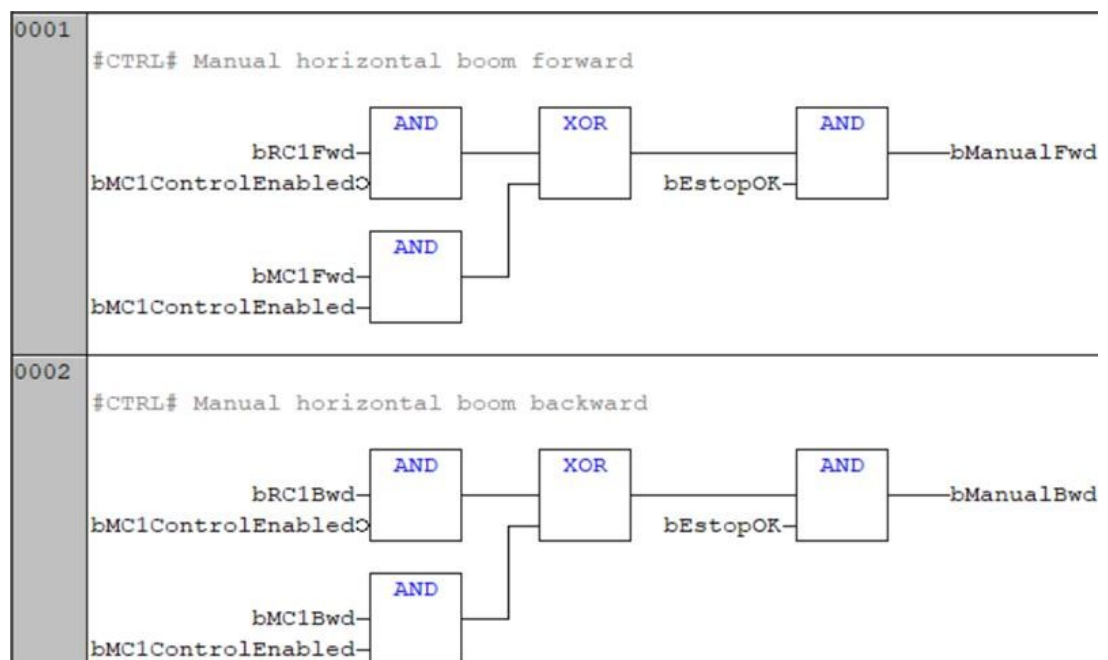
## 7.2 Nykyinen versionhallinta

Kun laite on saatu käyttöönottettua, projektin ohjelma tallennetaan yrityksen verkkolevyille. Verkkolevyille on tarkoitus tallentaa viimeisin versio projektin ohjelmasta. On tärkeää, että verkkolevyllä on varmasti viimeisin laitteessa sisällä oleva versio ohjelmasta, jotta kuka tahansa muu automaatio suunnittelija pääsee myöhemmin tarkastelemaan laitteen ohjelmaa esimerkiksi vianetsinnän yhteydessä. TwinCAT huomaa muutokset ohjelmassa ja kysyy ladataanko ohjelma logiikkaan, kun yritetään tarkastella ohjelmaa online-näkymässä. Tämä aiheuttaa tilanteen, jossa ohjelmaa tarkastelevan tulee harkita tarkkaan kannattaako versiota ladata logiikalle, jos sisään ladattavassa versiossa on jokin merkittävä muutos logiikassa olevaan ohjelmaan verrattuna. Pahimmassa tapauksessa eri versio ohjelmasta voi aiheuttaa tuotantokatkoksen, jos laite on ollut jo tuotantokäytössä. Asian selvittäminen ja varmistuminen oikeasta versiosta voi viedä runsaasti aikaa.

## 7.3 Ohjelman rakenne

Hitsaustornin ohjelmassa erilaisten toimilaitteiden ja liikkeiden ohjaukset sekä suurin osa hitsaukseen liittyvistä ohjauksista on toteutettu FBD-kielellä. FBD-kieli mahdollistaa visuaalisen ja helposti luettavan tavan esittää ohjaus. FBD-kielen visuaalisuudesta on apua etenkin online-näkymässä käyttöönoton aikana. Ohjaukset saadaan eroteltua eri osioihin, esimerkiksi kuvan 16 tapaisesti eteenpäin ajo saadaan omaan osioonsa ja taaksepäin ajo omaansa, jolloin ohjelman seuraaminen helpottuu.

Kuva 16. Esimerkki moottorin ajokäskyistä



Moottorin ohjauksen ohjelmassa eri ajokäskyt ja muut taajuusmuuttajalle tai servovahvistimelle tarvittavat muuttujat on toteutettu omissa ohjelmaosioissaan ja eri ohjelmaosioiden lähtömuuttujat linkitetään ohjelmassa olevaan taajuusmuuttajan tai servosäätimen rajapintalohkoon (kuva 17). Rajapintalohkon sisällä on ne muuttujat, jotka linkitetään taajuusmuuttajaan tai servovahvistimeen System Managerin puolella. Rajapintalohkon lähtömuuttujia voidaan käyttää hälytyksiin ja saadaan muita tietoja, esimerkiksi taajuusmuuttajan todellinen nopeus.

FBD-kielellä on myös toteutettu sekvenssejä, joissa edetään askel kerrallaan. Aikaisemman askeleen lähtömuuttujaa käytetään seuraavassa askeleessa yhtenä ehtona, eli ohjelma hyppää sekvenssin seuraavaan vaiheeseen vasta kaikkien aikaisemman vaiheen ehtojen toteuduttua.

Kuva 17. Taajuusmuuttajan rajapintalohko

STD_FB_I500_V2	
i_bEnable : BOOL	o_bRun : BOOL
i_bEnableFwd : BOOL	o_bRunFwd : BOOL
i_bEnableBwd : BOOL	o_bRunBwd : BOOL
i_bEndLimitFwd : BOOL	o_bParaError : BOOL
i_bEndLimitBwd : BOOL	o_bCommError : BOOL
i_bManMode : BOOL	o_bDriveWarning : BOOL
i_bAutoMode : BOOL	o_bDriveError : BOOL
i_bService : BOOL	o_nErrorCode : UINT
i_bManFwd : BOOL	o_bDisabled : BOOL
i_bManBwd : BOOL	o_bEndLimit : BOOL
i_bAutoFwd : BOOL	o_fAxisActSpeed : REAL
i_bAutoBwd : BOOL	o_fMotorActSpeed : REAL
i_bAck : BOOL	o_bSTOActive : BOOL
i_fManSpeed : REAL	oO_stControl : IO_ST_I500_Out_V1
i_fAutoSpeed : REAL	
i_fServSpeed : REAL	
i_fAxisMinSpeed : REAL	
i_fAxisMaxSpeed : REAL	
i_fMotorMaxSpeed : REAL	
i_tFBackTime : TIME	
iI_stStatus : IO_ST_I500_In_V1	

Hitsaustornin ohjelmassa erilaiset laskennat ja tarkastelut on toteutettu ST-kielellä. Laskennat ja tarkastelut monesti sisältävät vaativampia matemaattisia laskutoimituksia tai esimerkiksi FOR-looppeja, joiden tekeminen FBD-kielellä ei ole mahdollista. ST-kielellä saadaan ohjelmakoodi rivitettyä niin, että ohjelmoija näkee koko koodirivin sisällön. Tämä parantaa koodin luettavuutta merkittävästi.

Ohjelman Resources-välilehdellä nähdään eri globaalit muuttujalistat. Tällaisia muuttujalistoja ovat mm. IO ja Machine Configuration. Globaalin muuttujalistan muuttujia ja siellä määriteltyjen struct-tietorakenteiden muuttujia voidaan käyttää missä tahansa ohjelman sisällä. IO-muuttujalistaan määritellään kaikki ne muuttujat, jotka halutaan

linkittää fyysisiin I/O-kortteihin tai kenttäväylälaitteisiin TwinCAT System Managerin puolella.

Machine Configuration-muuttujalistasta löytyy constant-muuttujat, joiden arvo tai tila on vakio. Näillä constant-muuttujilla voidaan tehdä konfigurointia hitsaustornin toiminnallisuuksien mukaan. Muuttujalistaan voidaan esimerkiksi määritellä vaakapuomin vaakaliikkeen maksiminopeudeksi 3000 mm/min ja käyttää tätä muuttujaa kaikkialla ohjelmassa, missä tarvitaan tieto vaakaliikkeen maksiminopeudesta. Machine Configuration-muuttujalistaan määritellään mitä toiminnallisuuksia hitsaustornissa on käytössä. Esimerkiksi jos tornissa ei ole polttimien kallistusta, muuttuja TILT\_HEAD\_IN\_USE asetetaan FALSE-tilaan. Tällöin ohjelma ei ota huomioon polttimien kallistukseen liittyviä toiminnallisuuksia ohjelmassa. Muuttujalistaan lisäksi määritellään mm. mikä hitsausprosessi on käytössä ja polttimien lukumäärä tornissa, jolloin hitsauksen ohjaus toimii halutulla tavalla.

WC100-käyttöliittymä on konfiguroitu niin, että Machine Configuration-muuttujalistan constant-muuttujia muokkaamalla käyttöliittymä saadaan vastaamaan sitä, mitä hitsaustornissa todellisuudessa on. Esimerkiksi Machine Configuration-muuttujalistaan voidaan määritellä hitsauspolttimien lukumääräksi kaksi, jolloin WC100-käyttöliittymässä nähdään parametrit kahdelle virtalähteelle. Tämän avulla manuaalisen työn määrää automaattiosuunnittelussa on saatu vähennettyä.

Ohjelman eri osioille ja niihin kuuluville muuttujille on yrityksessä oma nimeämiskäytäntö, jota kaikkien tulee noudattaa. Tällä pyritään parantamaan ohjelmakoodin yhdenmukaisuutta eri kehittäjistä huolimatta. Tiettyyn ohjelman osaan viittaavat nimeämiset helpottavat vianetsintätilanteissa. Nimeämiskäytännön mukaan esimerkiksi liikeohjauksiin, väyläkommunikointiin ja laitteen hälytyksiin liittyvät muuttujat ovat kaikki omissa struct-tietorakenteissaan. Myös pääohjelmassa kutsuttavien alaohjelmien nimet ovat kuvaavia ja kertovat mitä niissä käsitellään.

## 7.4 Käytössä olevien komponenttien kartoitus

MD-, HD- ja EHD-tornien välillä on komponenttieroja laitteiden fyysisen kokoeron takia. Suurempi koko tarkoittaa, että tarvitaan raskaampiin käyttötarkoituksiin soveltuvia mekaanisia ja sähköisiä komponentteja. Nykytilannetta kartoitettiin käymällä läpi eri tuotemallien liikeohjauksiin käytettyjä mekaanisia ja sähköisiä komponentteja sekä muita varusteluja, joita eri tuotemallin torneihin on mahdollista saada. Yrityksen mekaniikkasuunnittelulla on olemassa mekaniikkamoduulit, jotka määräytyvät hitsaustornin

fyysisen koon ja määriteltyjen toiminnallisuuksien mukaan. Mekaniikkamoduulien kartoitus on tärkeä osa kehitystyötä ajatellen, sillä ensisijaisesti hitsaustornin koko määrittelee mitä komponentteja ja varusteluja on mahdollista käyttää. Eroa muodostui mm. liikeakseleiden ohjauksiin käytetyistä oikosulkumootoreista ja vaihteista sekä hitsausvarusteluista. Pienempiin MD-torneihin ei ole saatavilla yhtä kattavia varusteluja, kuin isompiin HD- ja EHD-torneihin.

#### 7.4.1 Vakioliikkeet

Kartoitusta alettiin tutkia ensiksi vakioliikkeiden osalta. Hitsaustornin vakioliikkeisiin kuuluu vaakapuomin sekä ristiluistin vaaka- ja pystyliike. Vaakapuomin pystyliikkeessä eroa tuli joka tuotemallien välillä, sillä tornin fyysisen koon kasvaessa vaakapuomista tulee raskaampi nostaa. Jokaisella tuotemallilla on oma moottori ja moottorin nimellistehon mukaan määräytyvä taajuusmuuttaja. MD- ja HD-tornien vaakapuomin nostoon käytetään yhtä moottoria. EHD-torneissa on kaksi nostomoottoria, jotka ovat samanlaisia kuin HD-tornin nostomoottori.

Vaakapuomin vaakaliikkeeseen käytössä voi olla tuotemallista riippumatta joko oikosulkumoottori tai servomoottori. Kaikkien tuotemallien kohdalla moottorin ohjaukseen riittää samankokoinen taajuusmuuttaja. Ainoa ero moottorien välillä tulee MD-tornissa, jossa riittää pienempi moottori. Jos vaakapuomin vaakaliikkeen ohjaukseen käytetään servomoottoria, on kaikissa tuotemalleissa käytössä sama servovahvistin ja servomoottori.

Ristiluistin vaaka- ja pystyliikkeen osalta kaikissa tuotemalleissa voi olla käytössä DC-mootoreilla toteutettu liikeohjaus. Jos käytössä on DC-moottorit, kaikissa tuotemalleissa liikkeet on toteutettu samoilla mootoreilla. Koska moottoreiden nopeutta voidaan säätää, tarvitaan lisäksi logiikalle ohjainkortti, joka syöttää moottoreita. Sama ohjainkortti sopii kaikkiin tuotemalleihin. HD- ja EHD-torneihin on saatavilla optiona heavy load-ristiluistit, joihin on omat moottorinsa. Ohjainkortti pysyy kuitenkin samana myös tämän option kanssa. Servomootoreilla toteutettu liikeohjaus on mahdollista HD- ja EHD-torneissa. Jos servo-ohjaus on käytössä, niin servovahvistin ja moottori ovat identtisiä.

#### 7.4.2 Optioliikkeet

Hitsaustornin kiskoliike on aina toteutettu taajuusmuuttajaohjatulla oikosulkumoottorilla. MD-tornin pienemmän koon ja massan takia sen kiskoliikkeen toteutukseen riittää pienempi taajuusmuuttaja ja moottori. HD- ja EHD-torneissa kiskoliike on toteutettu samalla taajuusmuuttajalla ja moottorilla.

Hitsaustornin vaakapuomin pyörytys on toteutettu myös aina oikosulkumoottorilla. MD-tornin pyörytykseen on käytössä pienempi moottori. HD- ja EHD-torneissa pyörytysmoottori on samanlainen ja kaikkien tuotemallien kohdalla käytössä on sama taajuusmuuttaja. Lisäksi pyörytysliikkeen kanssa on paineilmajarru, joka estää puomin tahattoman pyörytyksen. Tämä on samanlainen tuotemallista riippumatta.

Hitsauspään kallistuksen liikkeet on aina toteutettu servomoottoreilla. Hitsauspään kallistus on saatavilla HD- ja EHD-torneihin. Hitsauspään kallistus koostuu kallistavasta liikeakselista ja vaakasuuntaisesta offset-liikeakselista. Nämä kummatkin ohjaukset on toteutettu aina samoilla servovahvistimilla ja servomoottoreilla riippumatta siitä, onko kyseessä HD- vai EHD-torni.

### 7.4.3 Hitsausvarustelu

Jauhekaarhitsauksessa käytetään aina Lincoln Electric Power Wave AC/DC 1000SD-hitsausvirtalähteitä. Jauhekaarhitsausprosessissa varustelu voi muuttua huomattavasti. MD-torneihin on saatavilla vain yhdellä virtalähteellä ja polttimella varusteltu hitsauspää. HD- ja EHD-torneihin on saatavilla 1–3 virtalähdettä ja poltinta. Lankakelojen, langansyöttökontrollereiden ja langansyöttölaitteiden lukumäärä määräytyy tyypillisesti polttimien mukaan. Tuotemallista riippumatta hitsausvirtalähde, langansyöttökontrolleri ja langansyöttölaitteet ovat samanlaiset. Jauhetankkeja on useampaa eri kokoista ja automaattinen jauhetankin täyttö on mahdollinen optio. Jauhetankkien anturointi on samanlainen tuotemallista riippumatta. Hitsauspäässä on imurilla varustettu lisäjauhetankki, jotta hitsauksen aikana palamaton hitsausjauhe voidaan kierrättää ja käyttää uudelleen. Tämä on kaikissa tuotemalleissa samanlainen.

Railonseurantajärjestelmänä MD-tornissa voidaan käyttää GMD-seurantasormeaa tai optista korkeuden seuranta. HD- ja EHD-torneissa näiden lisäksi on mahdollista käyttää laserrailonseuranta.

GMAW-hitsausprosessissa käytössä on aina yksi poltin. Hitsausvirtalähteenä käytetään Lincoln Electric Power Wave S500-virtalähdettä. GMAW-hitsausprosessissa eroavaisuuksia eri tuotemallien välillä ei ole. Kaikissa tuotemalleissa käytetään samaa virtalähdettä, kontrolleria ja vesijäähdytintä. GMAW-hitsausprosessin kanssa voidaan railonseurantaan käyttää tuotemallista riippumatta GMD-seurantasormeaa tai optista korkeuden seuranta.

#### 7.4.4 Yhteiset moduulit

Hitsaustornin hitsauksen ohjausjärjestelmän näyttöpaneeli voi olla vaakapuomin päässä, tornin tasolla tai erillisessä telineessä. Hitsaustornin RC1-ohjaimen ohjauspiste voi olla joko vaakapuomin päässä lähellä hitsauspäättä, tornin tasolla tai erillään tornista. Kaikissa torneissa on käytössä sama RC1-ohjain, jolla voidaan ajaa kaikkia tornin liikkeitä ja aloittaa sekä lopettaa hitsaus. Lisäksi kaikissa tuotemalleissa on mahdollisuus ajaa hitsaustornin isoja liikkeitä pääkaapin ovesta. Isoja liikkeitä ovat vaakapuomin vaaka- ja pystyliike, vaakapuomin pyöritys sekä kiskoliike.

Jauhekaarhitsausprosessin ollessa käytössä voi optiona olla apulangansyöttö, johon kuuluu langansyöttölaite, ohjausjännitteen jako hitsauspään päälangansyöttimeltä, apulangansyötön ohituskytkin ja painonappi käsiajtoa varten. Apulangansyöttölaitteet tulevat tornin tasolle lankakelojen viereen. Jos tornissa on erillinen kiskoilla tornin perässä kulkeva huoltoyksikkö, niin lankakelat ovat huoltoyksikön tasolla ja silloin apulangansyöttölaitteetkin tulevat sinne.

Jauhekaarhitsaustornissa hitsauspäähän on mahdollista saada optiona Lucid Vision Labs-prosessikamera, jonka avulla polttimien asettelu hitsausrailoon helpottuu. Tämän prosessikameran kanssa tarvitaan lisäksi näyttöpaneeli tornin vaakapuomin päähän, josta nähdään prosessikameran kuva. Hitsaustornin pääkaappiin asennetaan oma PoE-kytkin, johon näyttöpaneeli ja prosessikamera liitetään. GMAW-hitsaustorniin on mahdollista saada Cavitar-prosessikamera, jonka avulla nähdään reaaliaikaisesti ja tarkasti hitsausprosessi. Cavitar-prosessikameran kanssa tarvitaan myös lisäksi näyttöpaneeli.

Hitsaustorniin voidaan liittää kolme ulkoista laitetta. Ulkoinen laite voi olla esimerkiksi rullasto tai pyörityspöytä. Näitä kolmea ulkoista laitetta varten tornin pääkaapin kyljessä on sync-liittimet, joihin laitteet voidaan kytkeä. Tornin päästä voidaan ohjata ulkoisten laitteiden liikeakseleita. Hätäseis-piirin kytkennöissä tulee ottaa huomioon ulkoiset laitteet. Eli jos laite on liitettynä torniin, niin sekä ulkoisen laitteen omien ja tornin hätäseis-painikkeiden tulee toimia. Ulkoisten liikeakseleiden ohjaus on vakio kaikissa hitsaustorneissa.

Muita hitsaustornien yhteisiä moduuleja ovat varoitusvalo ja äänisignaali tornin pääkaapin päällä, työvalot, pistelaserit polttimien kohdistusta varten sekä hitsattavan kappaleen lämpötilan mittaukseen käytettävät infrapuna-anturit. Työvalot, pistelaserit ja lämpötila-anturit sijaitsevat hitsauspäässä. Kaikkiin torneihin on myös saatavilla ilmankuivain, jolle tuodaan tornin pääkaapista sähkösyöttö.

## 8 Hitsaustornin modulaarinen kehitys

Yrityksessä käytetään ohjelmistokehitykseen TwinCAT 2-ohjelmistoa. TwinCAT 2-ohjelmistoa on tuettu tähän päivään asti ja se on ollut luotettava alusta, joten tarvetta siirtyä TwinCAT 3-ympäristöön ei ole koettu tarpeelliseksi. Microsoft on kuitenkin lopettanut tuen Windows CE-käyttöjärjestelmälle ja Beckhoff lopettaa tätä käyttöjärjestelmää käyttävien komponenttien valmistuksen. Osassa yrityksen käyttämistä logiikoista on tämä kyseinen käyttöjärjestelmä. Tämän takia yrityksessä on alettu tehdä hallitusti siirtymää kohti TwinCAT 3-ympäristöä. Samalla kun siirrytään kohti modernimpaa ja modulaarisempaa ohjelmointia tukevaa ohjelmointiympäristöä, on laitteiden ohjelmakoodien rakennetta tarkoitus uudistaa. Suurimpia muutoksia tulee liikkeiden ohjausten rajapintaan moottorilohkon kanssa ja hitsauksen moodien sekä parametrien käsittelyyn. Ohjelmien modulaarisuutta aletaan ensiksi kehittää niin kutsuttujen vakiolaitteiden kohdalla, joihin hitsaustorni kuuluu. Yritys on lisäksi kehittämässä uutta WCX-hitsauksen ohjausjärjestelmää, jonka on tarkoitus tulevaisuudessa korvata WC100- ja WC500-järjestelmä. Ensimmäisen WCX-järjestelmällä ja TwinCAT 3-ohjelmistolla toteutetun hitsaustorniprojektin on tarkoitus alkaa vuonna 2026.

### 8.1 I/O-terminaalien ja -kanavien vakiointi

Hitsaustornin I/O-terminaalit ja -kanavat haluttiin vakioida niin, että jokainen hitsaustornin vakiosignaali ja kartoitettujen optioiden signaalit olisivat aina samassa terminaalissa sekä kanavassa. Tämä helpottaa ohjelmistosuunnittelua, kun hitsaustornin tuloja ja lähtöjä ei ole tarvetta joka kerta tarkistaa mahdollisten muutosten varalta. Myös sähkösuunnittelussa työn määrä vähenee ja asennusvirheiden mahdollisuus pienenee sähkökeskuksen kytkentöjen pysyessä yhdenmukaisena. I/O-terminaalien ja -kanavien vakiointi aloitettiin hitsaustornin MC1-pääkaapista ja tämän jälkeen vakiointi tehtiin vaakapuomin päässä olevaan CB1-kenttäkoteloon, jossa on EK1100-EtherCAT Coupler. EK1100 on kytkin, jonka avulla voidaan hajauttaa I/O-terminaaleja esimerkiksi kentällä oleviin kenttäkoteloihin.

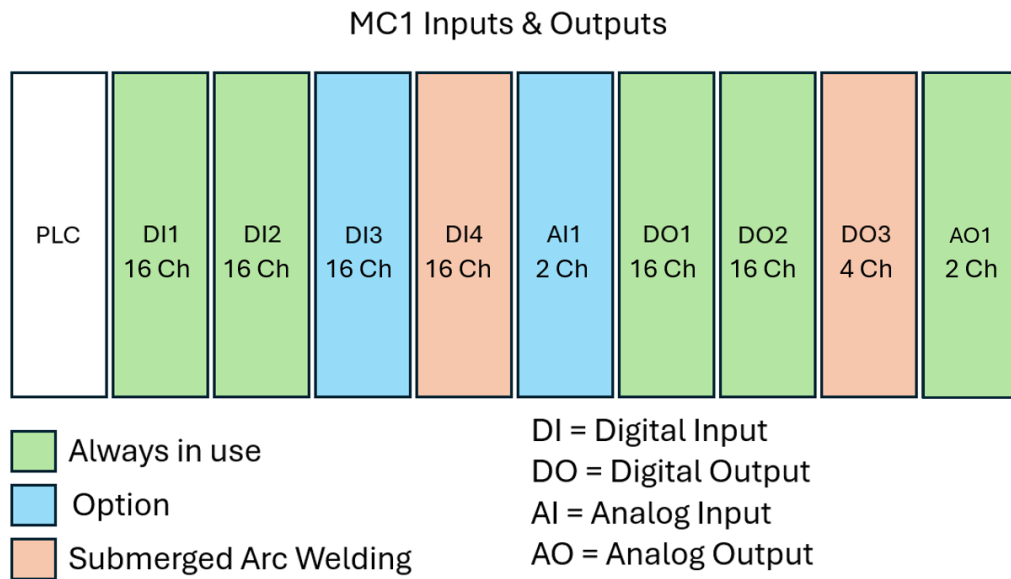
I/O-terminaalien kanavat jaettiin niin, että kaikki samantapaiset signaalit ovat aina samassa terminaalissa. Osa liikeakseleista on optioita, mutta päätyrajojen kanavapaikat ovat kuitenkin aina samat ja niiden tilalle ei tule muita signaaleja. Osa terminaaleista on varattu ainoastaan optioille tai tiettyyn hitsausprosessiin sidotuille signaaleille ja jos yksikään kyseisen terminaalin signaaleista ei ole käytössä, voidaan terminaali jättää pois. Terminaalien nimeämiskäytäntö pysyy aina vakiona, vaikka välistä otettaisiin terminaali pois. Näin yhdenmukaisuus säilyy ja terminaalin poissaolon vaikutus sähkö- ja

ohjelmistosuunnittelun osalta minimoidaan. Terminaalin lisääminen jälkikäteen ei myöskään aiheuta juurikaan lisätyötä, kun I/O-muuttujien paikat pysyvät samoina.

MC1-pääkaapin I/O-terminaalit (kuva 18) ja niiden signaalit sekä ohjaukset:

- **DI1:** Yleisiä kaikissa tornin tuotemalleissa käytössä olevia signaaleja. Näitä ovat mm. sulakkeiden kärkitieto, pystyliikkeen moottorin lämpöreleen kärkitieto, hätäseis-piirin kuittauspainikkeen signaali ja ulkoisien laitteiden signaaleja.
- **DI2:** Vaakapuomin vaaka-, pysty- ja pyörytysliikkeen sekä kiskoliikkeen päätyrajojen signaalit. Lisäksi terminaalissa on ovesa olevien liikkeenohjauskytkinten signaalit.
- **DI3:** Jos hitsaustornin ohjauspiste on tornin tasolla tai erillisessä telineessä eikä hitsauspäässä, niin tähän tulee kaikki RC1-ohjaimen signaalit. Jos ohjauspiste on vaakapuomin päässä, tämä terminaali voidaan jättää kokonaan pois.
- **DI4:** Käytössä vain silloin, jos hitsausprosessina on jauhekaarihitsaus. Terminaaliin tuodaan jauhepainetankin signaalit, apulangansyöttimien ohjaussignaali ja imurin käyntitieto ja hälytyssignaali.
- **AI1:** RC1-ohjaimen potentiometrin analoginen signaali, josta voidaan säätää portaattomasti liikkeiden nopeutta. Tämä terminaali tarvitaan ainoastaan tilanteessa, jossa hitsaustornin ohjauspiste on tornin tasolla tai erillisessä telineessä eikä vaakapuomin päässä lähellä hitsauspäättä.
- **DO1:** Yleisiä tornin tuotemallista riippumattomia ohjauksia. Näitä ovat esimerkiksi varoitusvalo ja äänisignaali, ulkoisten laitteiden ohjauksia ja RC1-ohjaimen merkkivalojen ohjauksia tilanteessa, jossa RC1-ohjain on tornin tasolla tai erillisessä telineessä.
- **DO2:** Joko jauhekaarihitsausprosessiin tai GMAW-hitsausprosessiin liittyviä ohjauksia. Jos prosessina on jauhekaarihitsaus, niin terminaaliin tulee jauhepainetankin venttiilien ohjaukset ja imurin ohjaukset. Jos puolestaan prosessina on GMAW, niin terminaaliin tulee vesijäähdyttimen ohjaus. Ylimääräiset kanavat ovat reserviä.
- **DO3:** Kun käytössä on jauhekaarihitsaus, niin tähän terminaaliin tuodaan langansyöttökontrollereiden shutdown-ohjaukset. Shutdown-ohjauksia tarvitaan tilanteessa, kun hätäseis on painettuna. Tämä terminaali voidaan jättää siinä tilanteessa pois, jos kontrollerit on sijoitettu vaakapuomin päähän.
- **AO1:** Ulkoisille laitteille tuodaan nopeuden ohjaus myös analogisena siltä varalta, jos EtherCAT-kenttäväylän kautta nopeutta ei saada. Tämä siis lähinnä varotoimenpiteenä.

Kuva 18. MC1-pääkaapin I/O-terminaalit



Tulevaisuudessa halutaan toteuttaa hitsaustornin hätäseis-piiri turvalogiikan avulla. Tähän asti tornin hätäseis-piiri on toteutettu pelkillä turvareleillä. Turvalogiikan tuloille ja lähdöille piti siis myös määrittää vakioidut kanavapaikat. Turvalogiikkaan tarvittiin kaksi 8-kanavaista tuloterminaalialia, joista toisessa on neljä kanavaa optioita varten. Näihin tuodaan hätäseispainikkeiden, ulkoisten laitteiden, ketjurikkoantureiden ja pääkaapin huoltokytkimen signaalit. 4-kanavaisia lähtöterminaaleja tulee turvalogiikkaan kaksi kappaletta. Niistä saadaan ohjaus turvareleille ja taajuusmuuttajille. Turvalogiikka tulee sijaitsemaan hitsaustornin MC1-pääkaapissa ja se tulee olemaan vakio tuotemallista riippumatta.

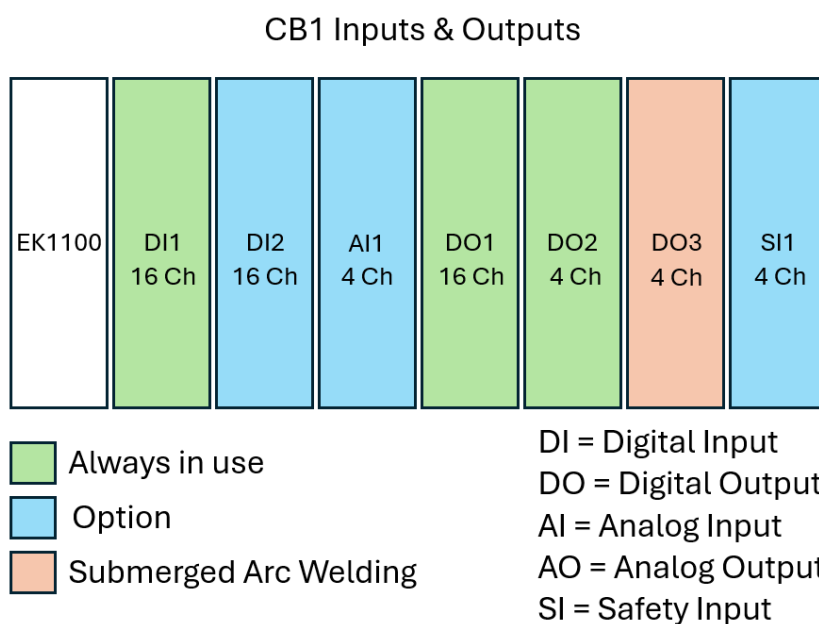
Vaakapuomin päässä sijaitsevan CB1-kenttäkotelon I/O-terminaalit määriteltiin samalla periaatteella kuin MC1-pääkaapissa. Normaalien I/O-terminaalien lisäksi CB1-kenttäkoteloon tarvitaan yksi turvaterminaalialia, jos hitsaustornin ohjauspiste on vaakapuomin päässä lähellä hitsauspäättä.

CB1-kenttäkotelon I/O-terminaalit (kuva 19) ja niiden signaalit sekä ohjaukset:

- **DI1:** Terminaaliin tuodaan kaikille hitsaustorneille yhteiset ristiluistin vaaka- ja pystyliikkeen päätyrajat. Näiden lisäksi terminaaliin tuodaan optiona polttimien kallistuksen liikeakseleiden päätyrajat. Jos hitsausprosessina on GMAW, niin terminaaliin tuodaan vaaputusliikkeen päätyrajat, langansyötön kylmäajon painikkeen signaali ja hitsauskaasun sekä jäähdytinnesteen virtauskytkimien signaalit. Ylimääräiset kanavat ovat reserviä.

- **DI2:** Tämä terminaali tulee tilanteessa, jossa hitsaustornin ohjauspiste on vaakapuomin päässä lähellä hitsauspäättä. RC1-ohjaimen painikkeiden signaalit tuodaan tähän terminaaliin.
- **AI1:** Tähän terminaaliin tuodaan RC1-ohjaimen potentiometrin signaali, jos ohjauspiste on vaakapuomin päässä. Terminaaliin tuodaan railon seurannan signaalit seurantatavan ollessa joko optinen railon seuranta tai GMD-seurantasormi. Mahdollinen lämpötilamittaus tuodaan myös tähän terminaaliin. Jos ohjauspiste ei ole vaakapuomin päässä tai käytössä ei ole joko optista korkeuden seurantaa tai GMD-seurantasormeaa, voidaan terminaali jättää pois.
- **DO1:** Tähän terminaaliin tuodaan hitsauspäässä olevien merkkivalojen ohjaus. Terminaalissa on lisäksi pistelaserin ohjaus, jauheentiputusventtiilien ja jauheenimuventtiilin ohjaus sekä prosessikameran / laserrailon seurannan jäähditysventtiilin ohjaus.
- **DO2:** Terminaaliin tuodaan työvalojen ja railon seurannan merkkivalon ohjaus sekä ensimmäisen hitsauspolttimen langansyöttökontrollerin shutdown-ohjaus. Shutdown-ohjausta ei tule tilanteessa, jossa kontrollerit on asennettu hitsaustornin pystyvuomiin.
- **DO3:** Tämä terminaali tarvitaan vain jauhekaarihitsausprosessissa. Terminaaliin tuodaan toisen ja kolmannen hitsauspolttimen langansyöttökontrollereiden shutdown-ohjaukset. Jos tornissa on vain yksi hitsauspoltin tai kontrollerit on asennettu pystyvuomiin, tätä terminaalia ei tarvita.
- **SI1:** Turvaterminaaliin tuodaan ohjaimen hätäseispainikkeen ja käyttölupakytkimen signaalit. Tätä terminaalia ei tarvita, jos hitsaustornin ohjauspiste on tornin tasolla tai erillisessä telineessä.

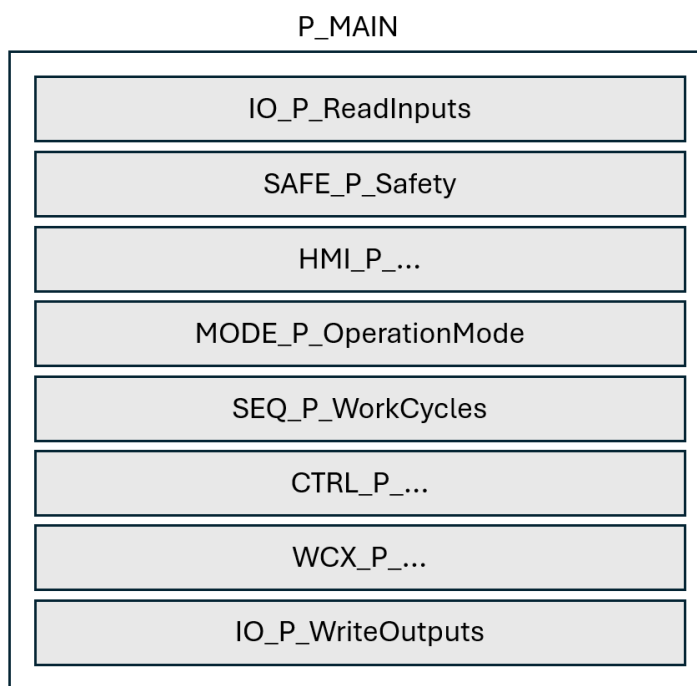
Kuva 19. CB1-kenttäkotelon I/O-terminaalit



## 8.2 Ohjelman modulaarisuuden kasvattaminen

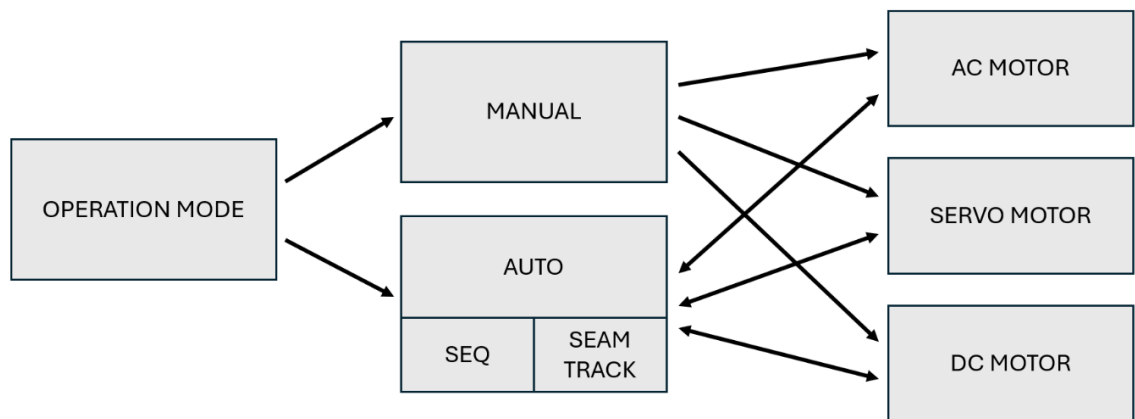
Ohjelman rakenne pääohjelmassa tulee pysymään pitkälti samanlaisena. Koska PLC lukee koodia ylhäältä alaspäin, on ohjelmien kutsumisjärjestyksellä merkitystä. Ensiksi pääohjelmassa tulee kutsua tulosignaalien lukeminen, sitten koneen turvapiirin tarkastelu ja muut toiminnallisuudet, kuten hitsauksen sekä liikeakseleiden ohjaukset. Lopuksi pääohjelmassa kutsutaan lähtöjen kirjoittaminen. Alla olevassa kuvassa 20 on määrittely pääohjelman rakenteelle. Kuvasta myös nähdään millainen nimeämiskäytäntö alaohjelmille on määritelty.

Kuva 20. Pääohjelman rakenteen määrittely



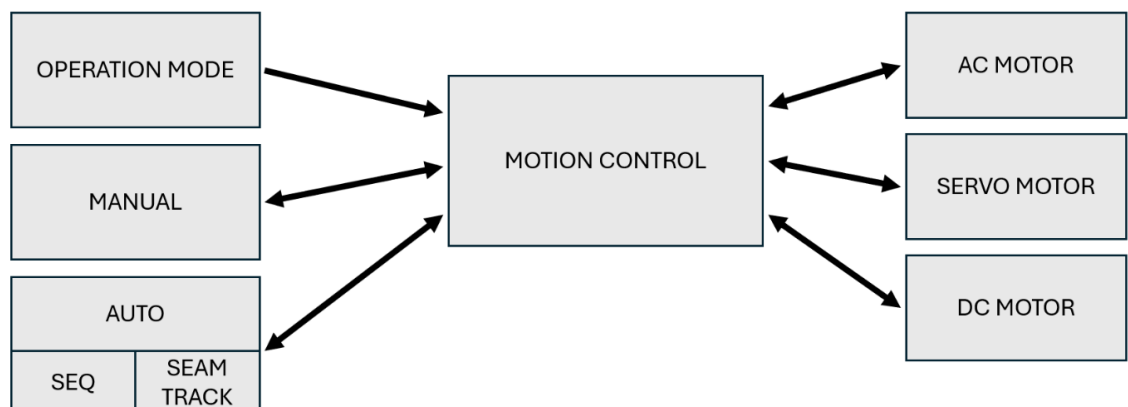
Alaohjelmiin tehdään enemmän muutoksia tulevaisuudessa. Ohjelmoinnissa halutaan panostaa siihen, että ohjelmistokehitystä tehdään olio-ohjelmoinnin periaatteita hyödyntäen. Liikkeiden ohjauksesta pyritään tekemään modulaarisempi niin, että samalla liikeohjauslohkolla voidaan ohjata moottoria riippumatta siitä, millaista moottoria todellisuudessa ohjataan. Tällä hetkellä liikkeiden ohjauksissa on paljon riippuvuuksia moottorilohkojen eroavien rajapintojen takia kuvan 21 tapaisesti. Kun jokaisella moottorilla on oma lohkonsa ja rajapintansa muuhun ohjelmaan, se tekee moottorin vaihtamisesta työlästä.

Kuva 21. Moottorilohkojen rajapinnat muuhun ohjelmaan



Moottorin ohjaukselle kehitellään oma funktiolohko, jossa käsitellään esimerkiksi missä moodissa ja millä nopeudella moottoria halutaan ohjata. Funktiolohkoon tuodaan muualta ohjelmasta vaadittavat muuttujat, mutta rajapintaan itse moottorilohkon kanssa ei tarvitsisi tehdä muutoksia kuvan 22 määrittelyn tapaan. Oli sitten kyseessä taajuusmuuttajaohjattu oikosulkumoottori, servomoottori tai DC-moottori, tätä samaa funktiolohkoa voidaan käyttää ohjaukseen. Moottorilohkon itsessään ei tarvitse tietää ohjataanko sitä manuaalisesti, automaattimoodissa tai sekvenssin aikana. Moottorilohkoon tarvitaan tieto milloin liikkuu, mihin suuntaan ja millä nopeudella. Kerran hyvin toteutettu moottorin ohjauslohko vähentäisi merkittävästi automaatio suunnittelun työkuormaa.

Kuva 22. Määrittely moottorin ohjauksen toteutukselle tulevaisuudessa



Kirjastoon tullaan kehittämään myös ohjausloikkoja esimerkiksi venttiilien ja hydraulikkakoneikon ohjaukselle. Hydraulikka- ja paineilma venttiilien ohjaus on voitu tällä hetkellä toteuttaa useammalla eri tavalla. Venttiilien ohjaukseen voidaan käyttää yrityksen omasta kirjastosta löytyvää päälle/pois-ohjausloikkaa, mutta osassa ohjelmista ohjaus on toteutettu itsetehdyllä versiolla. Itsetehty versio ei välttämättä toimi suoraan toisessa

käyttötarkoituksessa. Ohjelmoinnissa pyritään jatkossa enemmän hyödyntämään yrityksen omia kirjastolohkoja, sillä ne tulee olemaan sellaisia, jotka on toteutettu modulaarisuutta ajatellen. Jatkossa yritetään vähentää tiettyyn koneeseen sidottuja niin sanotusti polven päällä tehtyjä ohjelmia.

Ohjelmoinnissa tullaan tulevaisuudessakin käyttämään nykyisen Machine Configuration-globaalin muuttujalistan tapaisia muuttujia, joilla voidaan määritellä erilaisia toimintoja. Machine Configuration-muuttujalistaan tullaan määrittelemään esimerkiksi moottorien maksiminopeudet, enkooderien resoluutiot ja tiettyjen lisätoiminnallisuuksien käyttö, kuten polttimien kallistus tai kiskoliike.

### 8.3 WCX-hitsauksen ohjausjärjestelmä

Uutta hitsauksen ohjausjärjestelmää alettiin kehittää, koska osassa käytetyistä logiikoista on Windows CE-käyttöjärjestelmä, jonka tukemisen Microsoft on lopettanut. Windows CE-käyttöjärjestelmällä olevia logiikoita on käytetty mm. WC100-hitsaustorneissa. Tuen loppumisen myötä WC100-torneja ei tulla enää toteuttamaan. WC100-järjestelmään on sidottu tiettyjä ominaisuuksia ja WC500-järjestelmä on kehitelty laserrailonseurantaa ajatellen. WC500-järjestelmän muokkaaminen sellaiseksi, että se tukee kaikkia hitsauslaitteita olisi sen verran työlästä, että ennemmin lähdettiin luomaan uutta kaikkiiin hitsauslaitteisiin sopivaa järjestelmää. Tarkoituksena on luoda hitsauksen ohjausjärjestelmä, joka yhdistäisi WC100- ja WC500-järjestelmän ominaisuudet.

WCX-hitsauksen ohjausjärjestelmän käyttöliittymä tullaan toteuttamaan TwinCAT 3 HMI:n avulla. WCX-järjestelmän myötä hitsauksen ohjaukseen tulee merkittäviä muutoksia. Tällä hetkellä hitsausvirtalähteen kanssa kommunikoidaan DeviceNet-kenttäväylän yli. Hitsauksessa käytetään virtalähteen omaa sisäistä sekvensseriä ja virtalähteelle kerrotaan etukäteen hitsausmoodi sekä parametrit, joilla halutaan hitsata. Tämän jälkeen virtalähteelle lähetetään hitsauspyyntö ja virtalähde toteuttaa hitsaussekvenssin. Kaikki hitsausmoodit ja parametrit on tällä hetkellä jouduttu kirjoittamaan PLC:n persistent-muuttujiin.

Tulevaisuudessa kommunikointi virtalähteen kanssa toimii ArcLink-kommunikaatiolla. ArcLink-kommunikaatio toimii UDP/IP-protokollalla Ethernetin yli. Tämän uudistuksen myötä vanhaa DeviceNet-kommunikaatiota ei tarvita ja hitsaussekvenssiä hallitsee PLC. ArcLink-kommunikaation myötä hitsausvirtalähteestä saadaan enemmän diagnostiikkaa kuin tällä hetkellä, esimerkiksi hälytyksiä varten.

Jatkossa kaikki hitsausmoodit ja parametrit tallennetaan tietokantaan. PLC kysyy parametrit seuraavalle hitsauspalolle tietokannasta kyselyrajapinnan yli. Näin kaikkia hitsausparametreja ei tarvitse kirjoittaa PLC:n persistent-muuttujiin, vaan PLC saa tietokannasta oleelliset parametrit seuraavan hitsauspalon suorittamiseksi.

## 8.4 Versionhallintajärjestelmän hyödyntäminen

TwinCAT 3-ympäristöön siirtymisen myötä versionhallintajärjestelmien hyödyntäminen olisi järkevää. Ohjelmistoja halutaan yhdenmukaistaa, jotta tulevien projektien eteneminen sujuu hallitummin. Yrityksessä aiotaan entistä enemmän panostaa funktioiden ja funktiolohkojen kirjastointiin, jotta niitä voidaan hyödyntää myös tulevaisuudessa. Tässä kohtaa versionhallintajärjestelmästä olisi suuri hyöty. Versionhallintaan voitaisiin käyttää esimerkiksi Git-versionhallintaa ja sen kanssa Git Extensions-graafista käyttöliittymää.

Kaikki kirjastolohkot tullaan tallentamaan yrityksen omaan kirjastoon, jota tullaan ylläpitämään aktiivisesti. Kun uusia toiminnallisuuksia halutaan lisätä kirjastoon, se viedään prosessin läpi, jossa vähintään kaksi muuta suunnittelijaa tarkistaa sen. Tarkistuksella pyritään siihen, että koodi noudattaa määriteltyjä nimeämiskäytäntöjä ja on helposti muiden ymmärrettävissä. Samalla myös ohjelmalohkon toiminta tulee tarkistettua ulkopuolisen toimesta. Jos jo kirjastossa olevia lohkoja muokataan, tulee tämäkin viedä saman prosessin läpi uusiksi. Tällä tavalla kirjastoa voidaan ylläpitää hallitusti.

Hitsaustornille voitaisiin versionhallinnan myötä rakentaa pohjaprojekti, jonka päälle aina uusi projekti luodaan määrittelyjen mukaan. Pohjaprojektia voidaan ylläpitää ja päivittää aina tarpeen mukaan, jolloin ei enää tule sitä tilannetta, että aikaisemmasta projektista otettu pohja ei sovi täydellisesti ilman kehittäjän omia korjauksia. Näistä korjauksista tai muutoksista harvoin jää mitään muutoshistoriaa, joten niiden tarkastelu jälkikäteen on haastavaa.

## 9 Yhteenveto ja pohdinta

Tämän opinnäytetyön tarkoituksena oli tehdä modulaarisen kehityksen määritelmiä hitsaustornin I/O-komponenttien ja liikkeiden ohjauksen osalta. Työssä myös huomioitiin miten tämänhetkinen versionhallinta on toteutettu ja miten sitä voitaisiin kehittää tulevaisuutta ajatellen. Työn ansiosta saatiin selkeä dokumentaatio eri hitsaustornin tuotemallien mekaanisista ja sähköisistä komponenteista. Dokumentaation pohjalta voitiin lähteä toteuttamaan I/O-terminaalien ja -kanavien vakiointia sekä selvittämään millaisia

eroavaisuuksia tuotemallien välillä on toiminnallisuuksien osalta. Nämä eroavaisuudet vaikuttavat olennaisesti sähkö- ja automaatio suunnitteluun.

I/O-terminaalien ja -kanavien vakiointi saatiin tehtyä kokonaisuudessaan. Vakioinnilla saavutetaan se hyöty, että sähkösuunnittelussa tarkistetaan mitä toiminnallisuuksia on käytössä ja mitkä I/O-terminaalit tarvitaan, mutta kanavien osalta ei tarvitse tehdä muutoksia. Lisäksi automaatio suunnittelussa I/O-muuttujia ei tarvitse käydä yksitellen läpi. Vakiointi siis toisin sanoen vähentää sähkö- ja automaatio suunnitteluun kuluva aikaa ja näin ollen hitsaustornin valmistuskustannuksia saadaan maltillisemmiksi.

Määrittely liikkeiden ohjauslohkolle saatiin valmiiksi ja sen pohjalta aloitetaan kehittämään modulaarisempaa ohjauslohkoa. Työssä määriteltyjen selkeämpien rajapintojen ja yhteisen liikkeiden ohjauslohkon avulla hitsaustornin määrittelyistä johtuvien komponenttierojen aiheuttama lisätyö automaatio suunnittelussa vähenee. Yhteinen liikkeiden ohjauslohko ei ainoastaan vähennä työn määrää hitsaustornien osalta, vaan tätä samaa lohkoa voidaan käyttää myös muiden laitteiden liikeakseleiden ohjaukseen. Modulaarisempi ohjelmarakenne vähentää automaatio suunnittelun työkuormaa ja projektin läpimeno on suoraviivaisempi. Todellinen hyöty käytännön tasolla kuitenkin tullaan näkemään vasta tulevaisuudessa, kun enemmän projekteja on toteutettu modulaarisempia ohjelmaloikoja hyödyntäen. Ohjelman modulaarisen kehityksen osalta jää vielä epäselväksi, kuinka ohjelmamuutokset todellisuudessa tullaan toteuttamaan.

Versionhallintaan tullaan panostamaan enemmän ja sitä modulaarisuus vaatii. Yrityksen oman kirjaston aktiivinen ylläpito ja kehitys takaavat sen, että tulevaisuudessa hitsaustornien ja muiden laitteiden ohjelmat tulevat yhdenmukaistumaan ja modulaaristen ohjauslohkojen avulla koodia ei tarvitse yhtä paljoa rakentaa tyhjästä. Jos aktiivista kirjaston ylläpitoa ja muutosten hallintaa ei ole, käy vuosien mittaan niin, että samasta ohjauslohkosta on useampi hieman toisistaan poikkeava versio, jotka eivät toimi täysin toisessa käyttötarkoituksessa. Versionhallinta myös varmistaa sen, että aina voidaan palauttaa varmasti toimiva versio ohjelmasta.

Kun aikaisemmista projekteista valitaan sopivin pohjaprojektiksi, siihen liittyy omat riskinsä. Pohjaprojektin mukana saattaa tulla bugeja, joita ei olla huomattu aikaisemmassa projektissa. Toinen mahdollinen riskitekijä on useamman vuoden takainen pohjaprojekti. Tässä ajassa jokin toiminnallisuus on saattanut muuttua, eikä toimi enää toivotulla tavalla uudessa ohjelmassa. Tämä saattaa aiheuttaa ongelmia silloin, kun ohjelmaan joudutaan kopioimaan toiminnallisuksia useammasta eri hitsaustornin ohjelmasta. Nämä bugit havaitaan viimeistään käyttöönottovaiheessa, mutta tuovat lisää työkuormaa.

Hitsaustornille pyritään luomaan pohjaprojekti, jota tullaan käyttämään uuden projektin luomiseen. Pohjaprojektia voidaan versionhallinnan avulla ylläpitää ja tarvittaessa päivittää ja varmistetaan siitä, että pohjaprojekti on aina ajan tasalla ja toimii. On vielä kuitenkin epäselvää, halutaanko sama pohjaprojekti jokaiselle hitsaustornin tuotemallille vai luodaanko hitsaustornille useampi pohjaprojekti, kuten tällä hetkellä olemassa on WC100- ja WC500-hitsaustorni. Yksi mahdollinen ratkaisu olisi tehdä Basic- ja Advanced-pohjaprojekti. Basic-pohjaprojekti voisi olla enemmän suunnattu MD-torneille, joihin ei ole saatavilla niin edistyneitä toiminnallisuuksia. Advanced-pohjaprojekti olisi taas enemmän suunnattu HD- ja EHD-torneille, joissa mahdolliset lisätoiminnallisuudet ovat lähes samanlaiset.

## Lähteet

AutomationDirect. (n.d.). *Structured Text (ST)*.

[https://cdn.automationdirect.com/static/helpfiles/ls\\_plc/Content/A\\_IntroductionTopics/LP001-2.htm](https://cdn.automationdirect.com/static/helpfiles/ls_plc/Content/A_IntroductionTopics/LP001-2.htm)

Beckhoff. (n.d.-a). *Beckhoff Automation*. Haettu 11.3.2025 osoitteesta

<https://www.beckhoff.com/fi-fi/company/>

Beckhoff. (n.d.-b). *EtherCAT – the Ethernet Fieldbus*. Haettu 19.3.2025 osoitteesta

<https://www.beckhoff.com/en-en/products/i-o/ethercat/>

Beckhoff Information System. (n.d.-a). *Structured Text (ST)*.

<https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/925248523.html&id=>

Beckhoff Information System. (n.d.-b). IF instruction [kuva].

<https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/925258763.html&id=8919064708929057569>

Beckhoff Information System. (n.d.-c). *FBD/LD/IL*.

[https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_plc\\_intro/2526631051.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/2526631051.html&id=)

Beckhoff Information System. (n.d.-d). *TwinCAT 2*.

<https://infosys.beckhoff.com/english.php?content=../content/1033/ethercatsystem/2584717451.html&id=>

Beckhoff Information System. (n.d.-e). *TwinCAT 2 System Manager Overview*.

<https://infosys.beckhoff.com/english.php?content=../content/1033/tcssystemmanager/1084905739.html&id=>

Beckhoff Information System. (n.d.-f). Configuration with System Manager [kuva].

<https://infosys.beckhoff.com/english.php?content=../content/1033/tcel60xx/11400311819.html&id=7755405821417450735>

Beckhoff Information System. (n.d.-g). *Overview*.

<https://infosys.beckhoff.com/english.php?content=../content/1033/tcssystemover/12695808651.html&id=4621739027787748838>

Beckhoff Information System. (n.d.-h). *TwinCAT 3 Philosophy*.

[https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_overview/4275768971.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_overview/4275768971.html&id=)

Beckhoff Information System. (n.d.-i). TwinCAT 3 [kuva].

<https://infosys.beckhoff.com/english.php?content=../content/1033/el30xx/2584719371.html&id=>

Beckhoff Information System. (n.d.-j). *Object-oriented programming*.

[https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_plc\\_intro/2527303819.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/2527303819.html&id=)

Bråtegren, K. (n.d.). Software Modularity. *Modular Management*.  
<https://www.modularmanagement.com/blog/software-modularity>

Devasia, A. (25.3.2021). PLC Programming: Overview of IEC 61131-3 in Industrial Automation Systems. *Control*. <https://control.com/technical-articles/an-overview-of-iec-61131-3-Industrial-Automation-Systems/>

Dietrich, S. (29.8.2023). PLC Programming With Function Block Diagrams. *Control*.  
<https://control.com/technical-articles/plc-programming-with-function-block-diagrams/>

Donnell, A. (8.1.2024). The Power of Modular PLC Code. *Automation World*.  
<https://www.automationworld.com/control/article/33017035/the-power-of-modular-plc-code>

EtherCAT Technology Group. (n.d.). Technology Overview [kuva].  
<https://www.ethercat.org/en/technology.html>

Gillis, A. (n.d.). What is Object-Oriented Programming (OOP)? *TechTarget*.  
<https://www.techtarget.com/searcharchitecture/definition/object-oriented-programming-OOP>

GitLab. (n.d.). *What is version control?* <https://about.gitlab.com/topics/version-control/>

Kithara Software. (n.d.). *How does EtherCAT work?* <http://kithara.com/en/solutions/ethercat>

Polycase. (10.9.2021). *What Is a Programmable Logic Controller (PLC)?*  
<https://www.polycase.com/techtalk/electronics-tips/what-is-a-programmable-logic-controller.html>

Real Time Automation. (n.d.). *Control IEC 61131-3*.  
<https://www.rtautomation.com/technologies/control-iec-61131-3/>

Sharma, S. (29.8.2024). What is a PLC (Programmable Logic Controllers): A Comprehensive Guide. *Wevolver*. <https://www.wevolver.com/article/what-is-a-plc-programmable-logic-controllers-a-comprehensive-guide>