

# Lineaaridemo



Ammattikorkeakoulututkinnon opinnäytetyö

Sähkö- ja automaatiotekniikka

Valkeakoski, Kevät 2019

Petri Kiuru

Sähkö- ja automaatiotekniikka  
Valkeakoski

---

<b>Tekijä</b>	Petri Kiuru	<b>Vuosi</b> 2019
<b>Työn nimi</b>	Lineaaridemo	
<b>Työn ohjaaja/t</b>	Juha Sarkula	

---

## TIIVISTELMÄ

Tämän insinööriyön aiheena on demolaite, joka on rakennettu servo- ja lineaarimoottoreiden ajamista varten lineaarisella radalla. Demolaitteen moottoreita voidaan ajaa, sekä diagnosoida ohjauspaneelin käyttöliittymältä.

Työn teoriaosuudessa käydään läpi keskeisimpiä komponentteja liikkeenohjaukseen liittyen, jossa kuvataan niiden tarkoitusta liikkeenohjauksessa, sekä mainitaan muutamia asetuksia komponentteihin liittyen. Keskeisenä osa-alueena on myös koneturvallisuuden merkitys tässä työssä, sekä yleisesti, mutta myös käyttöliittymän merkitys liikkeenohjausjärjestelmiä käyttäessä.

Käytännön työssä suoritetaan käyttöönotto Beckhoff Automation Oy Hyvinkään toimistolla olevalle demolaitteistolle. Demolaitteistoon ohjelmoidaan käyttöliittymän kautta hallinnoitavia sovelluksia, joiden avulla servomoottoreita voidaan ajaa mielivaltaisesti. Laitteiston henkilö- ja koneturvallisuus toteutetaan valoverhoilla, jonka toiminnallisuus ohjelmoidaan turvalogiikkaan.

Tulosten ja tehdyn työn perusteella voi sanoa, että liikkeenohjausjärjestelmän käyttöönotto vaatii paljon perehtymistä aiheeseen, sekä laitteiston komponentteihin. Järjestelmien tuotetuntemus on tärkeää, mutta vähintäänkin yhtä tärkeää on ymmärtää PLC-ohjelmoinnin, kuin myös laiteparametroinnin merkitys, jotta laitteistosta saadaan toiminnallinen kokonaisuus. Demolaitteiston käyttöönotto toteutui onnistuneesti ja siitä tuli osa demolaitetekonaisuutta, jota on mahdollista hyödyntää koulutuksissa, markkinoinnissa, sekä tuotteiden diagnosoinnissa.

**Avainsanat** Enkooderi, lineaarimoottori, servomoottori, servovahvistin.

**Sivut** 65 sivua, joista liitteitä 9 sivua

Electrical and Automation Engineering  
Valkeakoski

---

<b>Author</b>	Petri Kiuru	<b>Year</b> 2019
<b>Subject</b>	Linear demo	
<b>Supervisors</b>	Juha Sarkula	

---

ABSTRACT

The subject of this project was a demo device built for servo and linear motors on a linear track, where the motors can be driven, as well as diagnosed through the control panel interface.

In the theoretical part of the thesis, the most important components of motion control are described, describing also their purpose in motion control and mentioning some settings related to them. The importance of machine safety about this work is also portrayed here, as well as the significance of the user interface, when using motion control systems.

In the empirical part of the demo equipment was commissioned at Beckhoff Automation's Hyvinkää office. Demo equipment is used to program applications that can be managed through a user interface that can be used to drive the servo motors arbitrarily. Personal and machine safety of the equipment is implemented by light curtains, whose functionality is programmed into the safety logic.

Based on the results and the work done, it can be said that the commissioning of a motion control system requires a lot of familiarity with the subject and the hardware components. Product knowledge of the systems is important, but it is at least as important to understand the significance of PLC programming as well as the importance of device parameterization in order to provide a functional entity. The commissioning of the demo equipment was successfully implemented and it became a part of a demo device entity that can be utilized in training, marketing, and product diagnosis.

**Keywords** Encoder, linear motor, servo drive, servo motor.

**Pages** 65 pages including appendices 9 pages

# SISÄLLYS

1	JOHDANTO.....	1
2	YRITYS JA TARPEET.....	2
3	SERVO-OHJAUSPIIRI.....	7
3.1	Servojärjestelmän anturit .....	8
3.1.1	Optinen inkrementtianturi .....	8
3.1.2	Resolveri .....	9
3.1.3	Optinen Absoluuttianturi .....	10
3.1.4	Magneettinen enkooderijärjestelmä (MES).....	10
3.2	Servomoottori .....	11
3.2.1	Synkroninen servomoottori.....	11
3.2.2	Lineaarimoottori.....	11
3.2.3	Lineaarimoottorin ja servomoottorin ero .....	14
3.3	Moottorin kommutointi .....	15
3.3.1	Mekaaninen kommutointi.....	15
3.3.2	Sähköinen kommutointi .....	16
3.4	Servovahvistin .....	16
3.4.1	Moottorityypit ja takaisinkytkentäprotokollat.....	17
3.4.2	Virransyöttö ja vahvistimien siltakytkentä .....	17
3.5	Turvakortti AX5805 .....	18
3.6	Servojärjestelmän viritys.....	19
4	TWINCAT 3.....	21
4.1	Ohjelmointikielet.....	21
4.2	TwinCAT Drive Manager.....	23
4.3	Käyttöliittymän suunnittelu .....	23
4.3.1	TwinCAT 3 PLC HMI .....	24
4.4	Ohjaavat toimilohkot .....	24
4.4.1	PLCopen .....	25
5	KENTTÄVÄYLÄT .....	27
5.1	EtherCAT.....	27
5.1.1	Jaettu Kello (Distributed Clock) .....	28
5.2	Servo Profile over EtherCAT .....	29
5.3	CoE, CANopen over EtherCAT .....	30
6	KONE- JA HENKILÖTURVALLISUUS .....	31
6.1	TwinSAFE .....	31
6.1.1	Black Channel Protocol.....	31
6.2	Turvaprojekti .....	32
6.2.1	Signaalinvalvontapulssi.....	33
6.3	Valoverho .....	34
7	TYÖN TOTEUTUS.....	35

7.1	Käyttöliittymä.....	35
7.2	Moottoreiden käyttöönotto.....	36
7.3	Kommutointikulman hakeminen absoluuttienkooderille.....	38
7.3.1	Mekaaninen säätö .....	38
7.3.2	Elektroninen säätö.....	39
7.4	Kommutointikulman hakeminen ei-absoluuttisella enkooderilla .....	39
7.4.1	Vaihe 1.....	40
7.4.2	Vaihe 2 .....	42
7.5	Henkilöturvallisuus.....	43
7.6	Lähdekoodin luominen.....	44
7.7	Demo-/testiohjelmat.....	46
7.7.1	Käsiajo ja paikoitus .....	46
7.7.2	Sekvenssi.....	46
7.7.3	Kuormitusohjelma .....	47
8	YHTEENVETO JA POHDINTA.....	48
8.1	Tavoitteiden toteutuneisuus.....	48
8.2	Jatkokehitys.....	49
	LÄHTEET.....	51

## Liitteet

LIITE 1/1:	PLCOPEN TOIMILOHKOJA LIIKKEENOHJAUKSEEN .....	57
LIITE 1/2:	PLCOPEN TOIMILOHKOJA LIIKKEENOHJAUKSEEN .....	58
LIITE 2/1:	OHJELMALOHKO FB_MANUAL_CONTROL .....	59
LIITE 2/2:	OHJELMALOHKO FB_MANUAL_CONTROL .....	60
LIITE 2/3:	OHJELMALOHKO FB_MANUAL_CONTROL .....	61
LIITE 2/4:	OHJELMALOHKO FB_MANUAL_CONTROL .....	62
LIITE 3/1:	KUORMITUSOHJELMA .....	63
LIITE 3/2:	KUORMITUSOHJELMA .....	64
LIITE 4/1:	OHJELMALOHKO FB_SEQUENCE .....	65

## 1 JOHDANTO

Liikkeenohjaus on nykyajan tuotanto- ja tehdasautomaatiossa välttämätön osa-alue. Kappaleita siirretään, tuotetaan lineaarista voimaa ja vääntömomenttia, muokataan erilaisilla koneistustyökaluilla hallitusti mikrometriä tarkkuudella. Kilpailu massatuotannossa kiristyy jatkuvasti, jossa pienetkin edistysaskeleet tuotantokoneen suorituskyvyssä tuottavat vuositasolla huomattavia tuotannollisia kasvuja. Tuotantokoneiden kappaleiden käsittelyn nopeuksien kasvaessa kasvaa myös tuotantokonetta ohjaavien laitteiden ohjausnopeuden merkitys, jolloin reaaliaikaiset kenttäväylät, sekä nopeat säätö- ja ohjauspiirit nousevat esille.

Tämänkaltaisiin sovelluksiin servo-ohjausjärjestelmät ovat energia- ja tuotantotehokkaita ratkaisuja tuottamaan nopeita, tarkkoja ja suurta voimaa vaativia liikeratoja. Siinä missä moottorin mitoitus järjestelmään on ratkaisevaa, on myös ohjaavan laitteiston ohjelmallinen käsittely ratkaisevaa keskeisen toiminnan kannalta. Kaiken tämän tukena toimii hyvin suunniteltu käyttöliittymä operaattorin työn helpottamiseksi, sekä järkevästi toteutettu turvalaitteisto turvaohjelmineen suojaamaan henkilö-, sekä konevahingoilta.

Liikeohjausta voisi kutsua automaation alalajiksi, jonka tarkoituksena on suorittaa hallittuja liikeratoja oikealla nopeudella ja voimalla halutunlaisessa ympäristössä. Tyypillisesti tuotantotekninen liikkeenohjauslaitteisto koostuu seuraavista komponenteista: liikeohjausta suorittava kontrolleri, servomoottori, ohjauspaneeli, turvalaitteisto. (Anaheim Automation, n.d)

## 2 YRITYS JA TARPEET

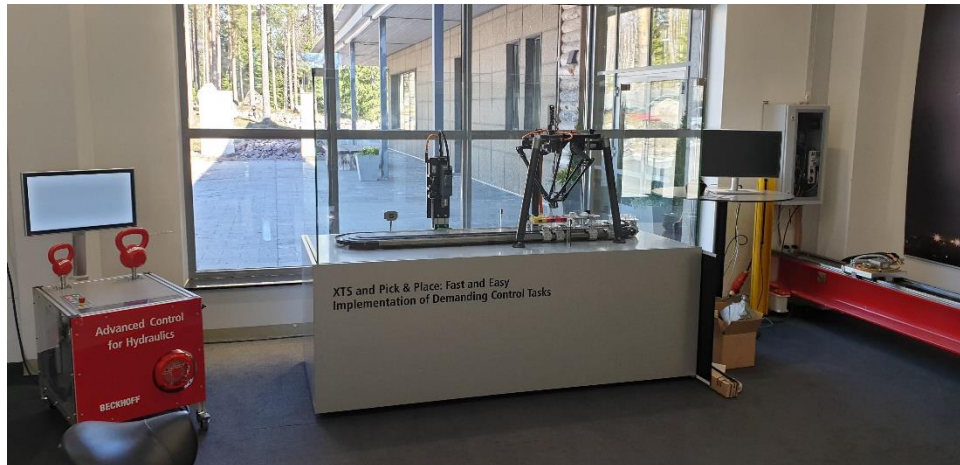
Työ toteutetaan yritykselle nimeltä Beckhoff Automation Oy, joka on kansainvälisesti automaattioratkaisuja tarjoavan yrityksen Beckhoff Automation GmbH & Co. KG Suomen yksikkö.

Yrityksen keskeisimpään tuotetarjontaan kuuluvat IO-komponentit, servomoottorit, askelmoottorit, teollisuus-PC:t, ohjauspaneelit, eri kenttäväyliä yhdistävät komponentit, sekä laitteiston toimintaa tukevia kehitysympäristön lisäosia, sekä ohjelmointikirjastoja. (Beckhoff, n.d.a)

Myynnin, sekä palveluiden kannalta keskeistä roolia näyttelevät tuotetukea tarjoava asiantuntijapalvelut, sekä jatkuva asiakkaiden kouluttaminen eri tuotekategorioiden liittyvissä asioissa, sekä laitedemonstrointi Beckhoffin tarjoamilla tuotteilla. Laitteiden toiminnan demonstroiminen on yrityksen myynnin kannalta yksi keskeisimpiä asioita havainnoillistamaan tuotteiden toiminnallisuutta. Työn tavoitteena on tehdä käyttöönotto ja luoda liikkeenohjaussovellus yrityksen demotilassa (kuva 1 ja 2) olevalle lineaaridemolle, joka on tarkoituksena saada merkittäväksi osaksi koulutuksia, sekä demokokonaisuutta.



Kuva 1. Demolaitteisto, lineaaridemo ja XTS. (Petri Kiuru, 2019)



Kuva 2. Demolaitteisto, hydraulikkademo ja XTS. (Petri Kiuru, 2019)

Koulutusten ja tuote-esittelyn kannalta on hyvin tärkeää mahdollistaa lineaaridemolla (kuva 3) moottorien ominaisuuksien esilletuonti, sekä parametroinnin vaikutus toiminnalliseen kokonaisuuteen. Keskeisenä tavoitteena on myös tuoda moottorien diagnostiikkaa luettavaksi ohjauspaneelilta, jotta saadaan selkeämpi käsitys moottoriin vaikuttavasta kuormituksesta.



Kuva 3. Lineaaridemmo. (Petri Kiuru, 2019)



Laitteiston keskeisimmät komponentit on lueteltu alla:

**CP3924** Kuva 3, nro. 1 on ohjauspaneeli, jonka kautta järjestelmää voidaan käyttää käyttöliittymän kautta. Siinä on monikosketusnäyttö, sekä liitännät DVI ja USB:lle

**SICK C2000** Kuva 3, nro. 2 on valoverhojärjestelmä, joka huolehtii käyttäjäturvallisuudesta. Valoverhot antavat turvalogiikalle signaalin, mikäli laitteisto havaitsee sen välissä olevan jotain ja turvaohjelmointi huolehtii järjestelmän alasajosta.

**AL2015** Kuva 3, nro. 3 on servomoottori, jonka tekniikka on rakennettu tuottamaan lineaarinen voima perinteisen rotaatioliikkettä tuottavan moottorin sijaan. Se tuottaa voimansa staattorikämmien tuottaman magneettikentän voiman vaikutuksesta lineaariradan magneettiliuskaan.

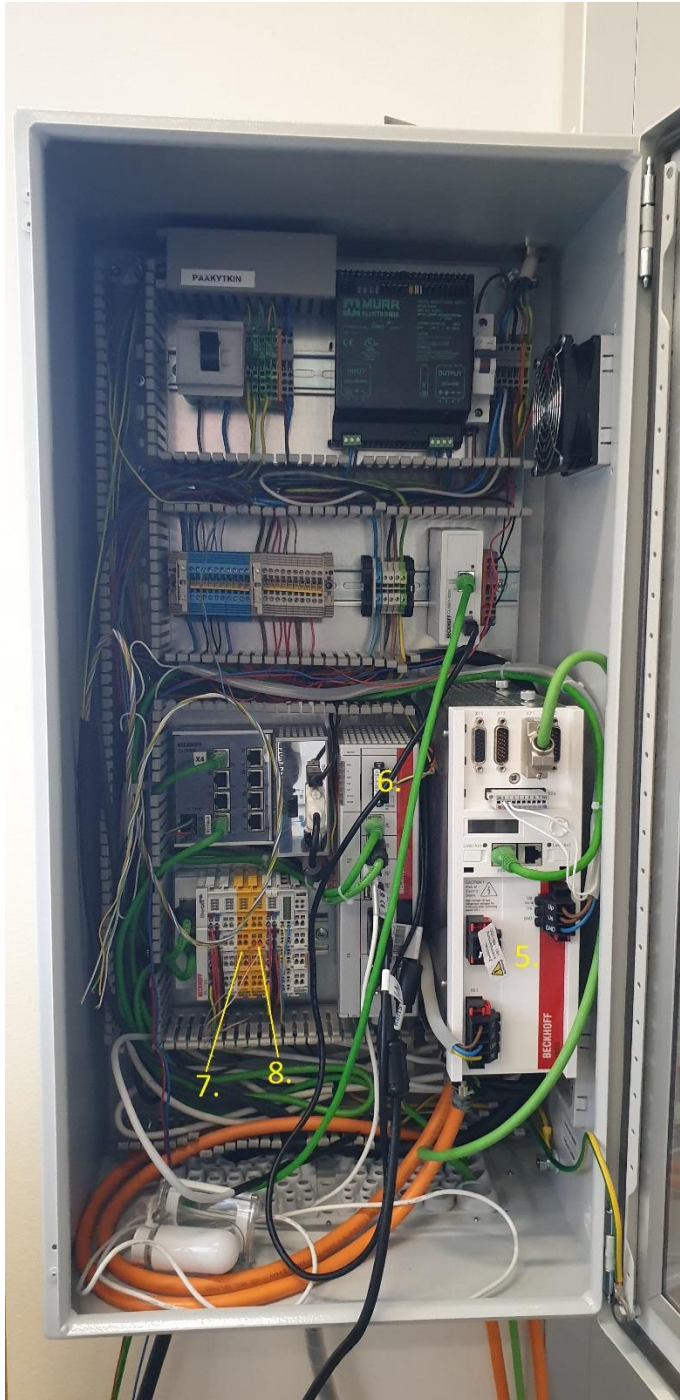
**AM8032** Kuva 3, nro. 4 on perinteinen rotaatioliikkeen tuottama servomoottori, jonka voimantuotto on kytketty vaihteen kautta lineaariradan hammaskiskoon.

**AX5203** Kuva 4, nro. 5 on servovahvistin, jossa moottorien säätöpiirin säätö tapahtuu. Se saa ohje- ja operointiohjeensa kontrollerilta C6915, jonka mukaan se syöttää moottorille virran. Turvakortti AX5805 on vahvistimen sisällä ja on kytketty turvalogiikkaan huolehtimaan vahvistimen nopeasta reaktiosta tarvittaessa moottoreiden turvafunktioiden toteuttamiseksi.

**C6915** kuva 4, nro. 6 on teollisuus-PC (IPC), jonka kautta koko laitteistoa hallinnoidaan. C6915 mahdollistaa tavallisen PC:n ja windows-ympäristön modulaarisuuden. Windows-pohjaisella ohjelmointiympäristöllä TwinCAT XAE mahdollistaa hyvin laajan skaalan eri kehitystyökaluille unohtamatta kernel-laajennuksen tuomaa etua ajaa logiikkaohjelmaa reaaliaikaytimessä.

**EL6900** Kuva 4, nro. 7 on turvalogiikkakortti, johon ohjelmoidaan ja joka toteuttaa laitteiston turvaohjelman.

**EL1904** Kuva 4, nro. 8 on turvatulokortti, joka huolehtii järjestelmän turvalaitteiston syöttämän input-signaalin vastaanottamisesta, jonka EL6900 käsittelee järjestelmän ohjelmoijan määrittämällä tavalla.



Kuva 4. Lineaaridemo, sähkökaappi. (Petri Kiuru, 2019)

Kyseinen demolaite otettiin ensi kerran käyttöön vuonna 2010 insinööriyön aiheena, jolloin laitteistolle tehtiin käyttöönotto, sekä demottava ohjelmisto.

Tämän jälkeen yrityksen tilat ovat siirtyneet ja demolaitteistoakin on päivitetty. Oleellimmat muutokset laitteiston kannalta ovat sähkökaapin uusiminen, IPC, sekä moottorikelkan muuttaminen yksilölliseksi moottoria kohden. Aikaisemmassa järjestelmässä kumpikin moottori oli kytketty samaan kelkkaan, joten muutos mahdollistaa monipuolisemman demotoiminnan suunnittelun.

Käyttöönotto tämänkaltaisessa työssä on hyvinkin laaja, jolloin eksyminen liian syväälle kuhunkin aihealueeseen on vaarana. Tämän takia keskeisimmistä tavoitteista ja aiherajauksista on syytä pitää kiinni, jotka ovat listattuna alla:

1. Ohjelmoida liikkeenohjaussovellukset, joiden avulla on mahdollista ajaa moottoreita käsiajolla, paikoittamisella, sekvenssitoiminnolla, sekä ohjelma, jonka avulla voidaan simuloida kuormaa toiselle moottorille käyttäen toista moottoria vastuksena.
2. Luoda käyttöliittymä rajapinnaksi käyttäjän ja ohjelman välille, josta sovellusta voidaan ohjata, sekä seurata diagnostiikkaa.
3. Asettaa turvarajat laitteen ympärille, sekä turvaohjelma suojaamaan henkilövahingoilta laitteiston kanssa.

### 3 SERVO-OHJAUSPIIRI

Servo-ohjauspiiri on toimilaitteen paikoitukseen tarkoitettu ohjauspiiri, joka saa ohjauskeskeisen tiedon takaisinkytkentänä sijainnin muutosta seuraavalta sensorilta. Seuraavissa kappaleissa servojärjestelmän toimilaitteesta puhuttaessa tarkoitetaan servomootoria. Servo-ohjauspiiri kuvattuna ao. kuvassa 5.



Kuva 5. Tyypillisen servo-ohjausjärjestelmän komponentit. (Beckhoff, n.d.cb)

Ohjauspiiri tarkastelee mitattavaa suuretta ja tämä tieto käsitellään halutulla tavalla ohjauspiiriä toteuttavassa laitteessa, jonka jälkeen ohjausviesti viedään servovahvistimelle, joka vahvistaa ohjauspiirin lähettämän signaalin niin korkeaksi, että sen energiamäärä riittää ohjaamaan haluttua toimilaitetta. Tämä vahvistettu signaali, joka on muutettu ohjausvirraksi syötetään edelleen servopiirin toimilaitteelle, joka suorittaa ohjauspiirin määrittelevän liikeradan.

Servo-ohjausta voidaan toteuttaa myös momenttiohjauksella, jolloin tarkasteltavana suurena on virta; rotaatioliikkeessä virta on verrannollinen vääntömomenttiin, lineaariohjauksessa voimaan. Kyseiselle ohjaustavalla ohjauspiiri lukee moottorin käyttämän voiman syötetyn virran perusteella. Mikäli moottorin tuottama voima jää alle määritetyn tason, ts. akselilla ei ole tarpeeksi vastusta moottorille, pyrkii järjestelmä kasvattamaan moottorille syötettävää jännitettä, jotta määritetty voima saavutettaisiin. Tämänlaisessa ohjaustavassa on huomioitava, että vastuksen puute akselilla johtaa moottorin kiihdyttämiseen niin kauan, kunnes määritely voimantuotto saavutettaisiin.

Yleensä puhuttaessa, servon säätämä suure on:

- Asema
- Nopeus
- Voima
- Momentti

(Fonselius;Rinkinen;& Vilenius, 1998, s. 7) (Beckhoff, n.d.c)

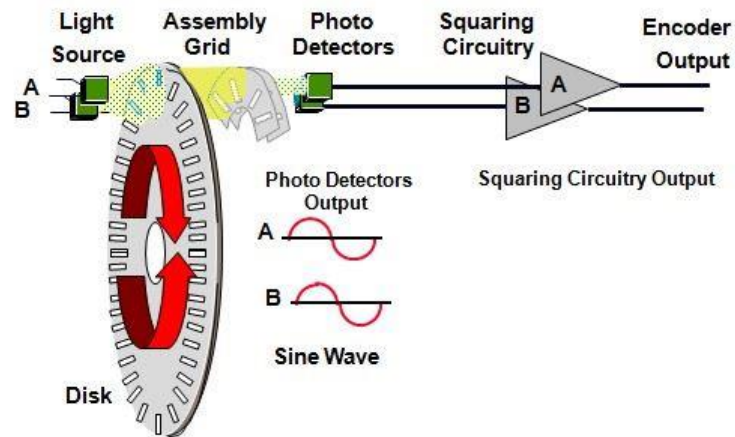
### 3.1 Servojärjestelmän anturit

Servo-ohjain tarvitsee jatkuvan takaisinkytkentätiedon operoimaltaan toimilaitteelta, jotta sen ohjaus ja muutostilojen seuranta olisi mahdollista. Tätä varten toimilaitteisiin on kiinnitetty anturi joka operointitavasta riippuen lähettää servojärjestelmälle tiedon paikasta, nopeudesta, tai kiihtyvyydestä, josta voidaan jakautua taas inkrementti-, tai absoluuttiantureihin.

Seuraavissa luvuissa käsitellään vain inkrementti- ja absoluuttiantureita niiden suuren suosionsa takia. Lopullisen päätöksen takaisinkytkentäjärjestelmän valinnalle sanelee yleensä hinta, käyttötarve ja käyttöolosuhteet. (TKK, 2008)

#### 3.1.1 Optinen inkrementtianturi

Optinen inkrementtienkooderi on sähkömekaaninen laite, joka sisältää pyörivään akseliin kiinnitettävän pulssilevyn ja tätä lukevan lukulevyn. Pulssilevy sisältää reikärivejä, joita voidaan kutsua myös sektoripareiksi ja kummankin rivin läpi suunnataan valo A ja B, joka kiekon asennosta riippuen antaa binäärisen tiedon lukulevylle. Pulssilevyn kahden sektorin väliset reiät ovat 90° vaihe-erolla, josta lukulevy voi lukea suunnan nousevan ja laskevan reunan mukaan, joka suunnasta riippuen vaihtuu. Pulssilevyn sektoriparien määrä määrittelee tarkkuuden akselin asennosta. Inkrementtienkooderi ei indikoi absoluuttista akselin asentoa, vaan se mahdollistaa akselin muutoksen seurannan. Optisen inkrementtienkooderin toimintaa havainnoillistava kuva 6. (TKK, 2008)

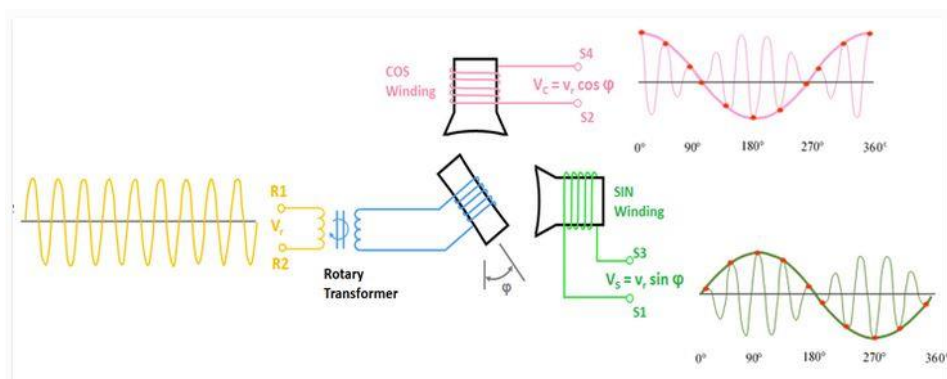


Kuva 6. Optical Incremental Encoder (Parker, 2016)

### 3.1.2 Resolveri

Resolveri on servotekniikassa käytetty anturi, joka on todettu hyväksi vaihtoehdoksi ympäristöön, jossa mittalaitteeseen kohdistuu poikkeavan kovaa tärinää. Resolverin toiminta kuvattuna ao. kuvassa 7.

Resolverin vikasietoisuus tulee sen sisältämästä tekniikasta, joka muistuttaa rakenteeltaan sähkömoottoria; ensiökäämi on kytketty roottoriin ja staattorissa on kaksi käämiä – SIN ja COS –käämitykset  $90^\circ$  vaihe-erolla. Kun resolverin ensiökäämiin (referenssikäämi) kytketään vaihtojännite niin indusoituu toisiokäämeihin, eli staattorin SIN- ja COS –käämeihin vaihtojännite, joiden amplitudi riippuu roottorin kulmasta. Resolveri tarjoaa absoluuttisen tiedon moottorin akselin kulmasta jännitearvona, joka yleensä käsitellään paikka-, nopeus-, tai etäisyystietona servovahvistimessa, tai ohjaavassa logiikassa. (MotionControlTips, 2015.a)



Kuva 7. Resolverin toimintaperiaate. (LearnChannel-TV, n.d.)

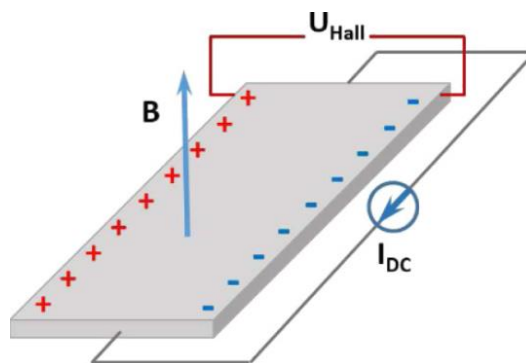
### 3.1.3 Optinen Absoluuttianturi

Optiseen absoluuttianturiin on rakennettu akselin mukana pyörivä pulssilevy, johon on kuvioitu akselin kulmaa vastaava binäärinen tieto. Kun levyä kohden syötetään valo, niin valo pääsee läpi tietyistä rei'istä lukulevyllä, josta voidaan määrittää akselin tarkka kulma.

Optinen absoluuttianturi pystyy tarjoamaan tarkan kulmatiedon virran pois kytkemisen jälkeenkin, koska tarkka kulmatieto saadaan luettua koodatulta levytä. (TKK, 2008)

### 3.1.4 Magneettinen enkooderijärjestelmä (MES)

MES koostuu lukupäästä, johon on integroitu Hall-sensori. Sensorit havaitsevat magneettikentän kestonmagneettiradasta ja lähettävät signaalin servovahvistimelle. Kuva 8 havainnoillistaa Hall sensorin Hall ilmiötä; kun sähkövirta kulkee johtavan materiaalin läpi magneettikentässä, magneettikenttä kohdistaa poikittaisen voiman liikkuviin varauksiin ja liikuttaa vastakkaismerkkiset varaukset toiselle puolelle johdinta. Varausten jakautuminen johtimen toisiin reunoihin stabiloi magneettisen vaikutuksen ja luo jännitteen johtimen vastakkaisen reunojen välille. (Hyperphysics, n.d.)



Kuva 8. Hallin ilmiö (Hyperphysics, n.d.)

Hall-jännite voidaan laskea kaavalla:

$$V_h = \frac{IB}{ned} \quad (1)$$

Missä  $V_h$  = Hall-jännite  
 $I$  = Johtimen läpi kulkeva virta  
 $B$  = Magneettivuon tiheys  
 $n$  = Varausten kuljettajien määrä  
 $e$  = Elektronivaraus  
 $d$  = Materiaalin tiheys

MES on semiabsoluuttinen takaisinkytkentäjärjestelmä ja se tarjoaa yhden sinioskillaation aina yhtä moottorin loogista kierrosta kohden. Moottorin looginen kierros on ekvivalenttinen kahden homopolaarisen magneetin etäisyyteen. Magneettisen enkooderijärjestelmän tarkkuus riippuu suurelta osin magneettiradan magneettien tarkkuudesta ja paikasta moottorin kulkureitin varrella. MES-takaisinkytkentäjärjestelmä on myös kustannustehokas lineaarimoottorille sen yksinkertaisuutensa takia, koska asteittaista sääntöä ei tarvitse asentaa. (Beckhoff, n.d.d)

## 3.2 Servomoottori

Servo-ohjausjärjestelmän moottorina voi toimia mikä tahansa sähkömoottori; vaihtovirta-, tasavirta-, harjaton-, harjallinen-, tai lineaarimoottori. Oleellista on takaisinkytkentä säätöpiirille, joka mahdollistaa moottorin ohjauksen saatujen fysikaalisten suureiden muutostietojen perusteella.

Pyörivät servomoottorit eroavat normaaleista sähkömoottoreista lähinnä siten, että niiden roottorin hitausmomentti on huomattavasti pienempi, joka johtuu moottorin suunnittelusta dynamiikasta moottorin halkaisijan ja pituuden suhteen. Pienempi hitausmomentti mahdollistaa nopean muutoksen kiihtyvyyksille ja tarkan paikoituksen, jotka ovat tärkeitä ominaisuuksia robotiikassa ja tuotantoautomaatiossa. (Fonselius;Rinkinen;& Vilenius, 1998, s. 10)

### 3.2.1 Synkroninen servomoottori

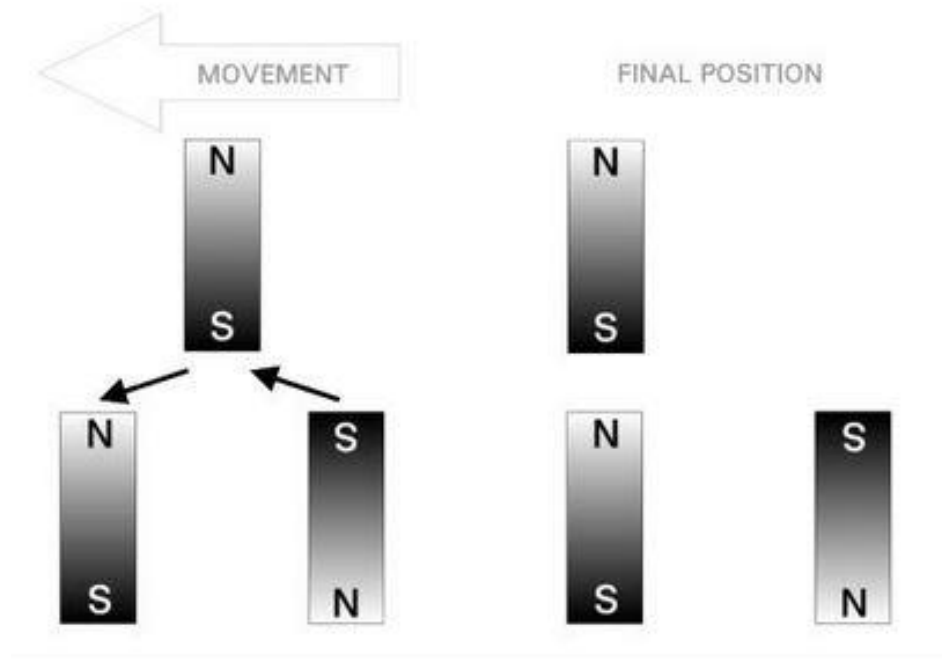
Synkroninen moottori, eli kestmagnetoitu tahtimoottori on roottoriltaan kestmagnetoitu, kun taas epätahtimoottori ei ole. Niiden staattorin rakenne on tekniikaltaan samanlainen, mutta tahtimoottorin roottori pyörii staattorin luoman magneettikentän nopeudella. Tahtimoottorin roottorin rakenne, magneettien sijainti ja materiaali vaikuttavat moottorin ominaisuuksiin. Tahtimoottoreilla on loistava suorituskyky dynaamisissa sovelluksissa ja siksi niitä käytetään paljon servokäytöissä. Tahtimoottori voi olla myös vierasmagnetoitu, jolloin roottori magnetoidaan erillisellä virtälähteellä. (Hietalahti, 2012, s. 71)

### 3.2.2 Lineaarimoottori

Lineaarimoottoriksi kutsutaan sähkömoottoria, joka tuottaa liikkeen suorassa linjassa pyörivän liikkeen sijaan. Kun perinteinen sähkömoottori tuottaa liikkeensä roottoriin pyörivän magneettikentän avulla, lineaarimoottori tuottaa liikkeensä suorassa linjassa ilman pyörivää magneettikenttää. Rakenteellisena erona lineaarimoottorilla on sen roottorikäänitys, joka on kuin avattu pitkittäiseksi matoksi perinteiseen pyörivään moottoriin verrattuna. Kuvan 9 mukaisesti lineaarimoottorin voima perustuu magnetismiin ja magnetismin polarisaatioon; kahden

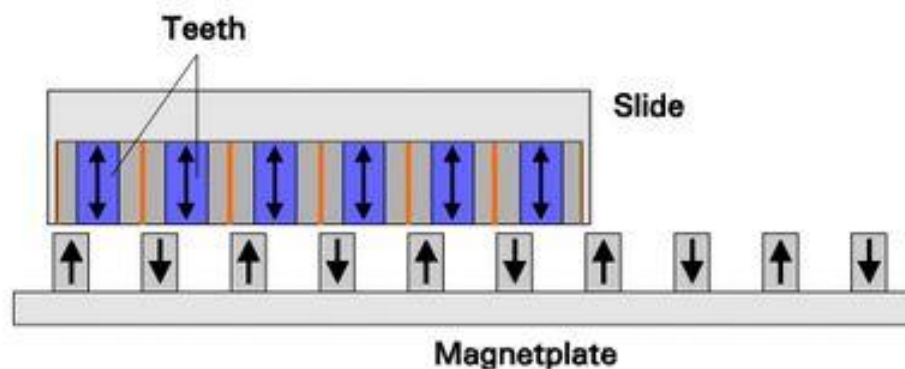


kestomagneetin välille syntyy voima, jonka suunta riippuu magneettien napaisuudesta.



Kuva 9. Magnetismi. (Beckhoff, n.d.e)

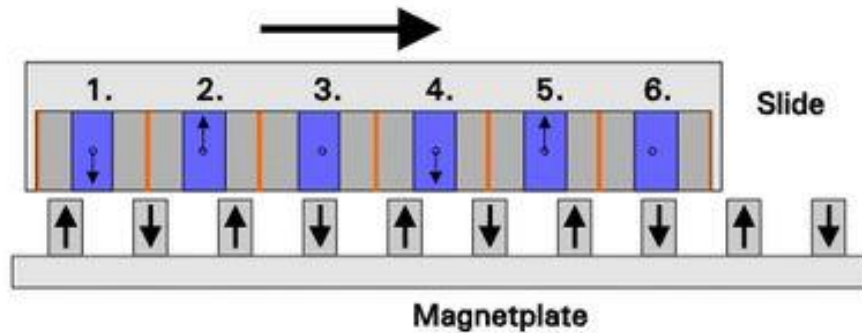
Lineaarisen liikkeen saavuttamiseksi ei kestopagneetteja voida asentaa rataan yhtä pitkin välein, kuin moottorin käämitys on kytketty. Kuvan 10 mukaisella asennustavalla moottorin käämitysten ja magneettiradan välinen voima ei veisi moottoria eteenpäin, kun vaihejärjestys kumoaisi toinen toisensa. (Beckhoff, Magnetism, n.d.e)



Kuva 10. Roottorimagneetit, sekä staattorikäätymys samassa vaiheessa. (Beckhoff, n.d.f)

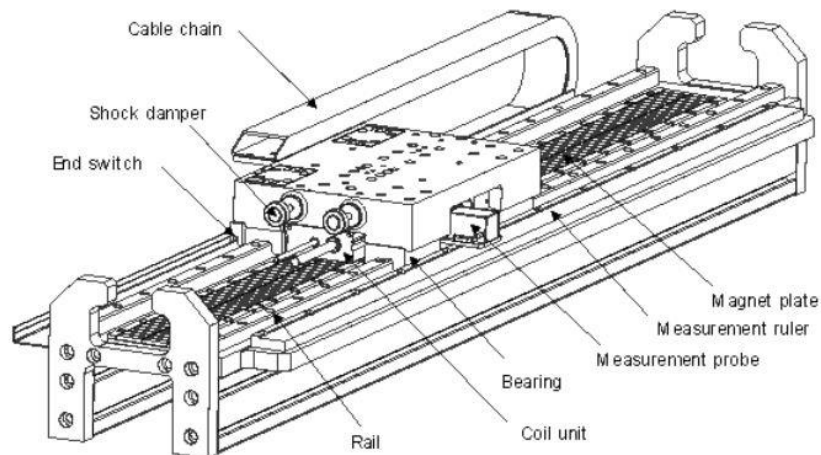
Lineaarimoottorin tarkkaan ja sulavaan liikkeeseen on erilaisia tekniikoita magneettien ja käämitysten kytkentöjen kannalta. Kuvan 11 tekniikan tavoin moottorien käämitysten paikan suhde magneetteihin on 3/4. Tällä tavoin voidaan asentaa kolmen eri vaiheen käämitykset tuottamaan sähkömagneettinen voima, jolloin vaiheiden mukainen vaihe-ero näkyy

myös käämien ja magneettien fyysisissä paikoissa. Lineaarinen liike syntyy, kun moottorille syötetään kolmivaiheinen virta. (Beckhoff, n.d.f)



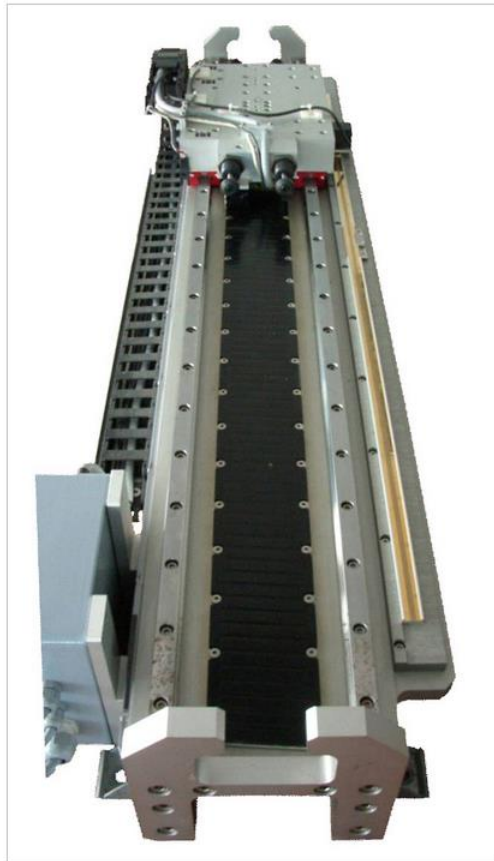
Kuva 11. Roottorimagneetit, sekä staattorikäänitys eri vaiheessa (Beckhoff, n.d.f)

Kuva 12 ja 13 havainnoillistavat yleistä lineaarimoottorin asennustapaa, jossa on myös kokonaisuudessaan servolineaarimoottorin asennukseen tarvittavat komponentit; kaapeliketju, iskunvaimentimet, päätyrajakytkin, kisko, moottorin käämiyksikkö, laakerointi, mittapää, mittarata, magneettiliuska. (Beckhoff, n.d.g)



Kuva 12. Lineaarimoottorin asennustapa.

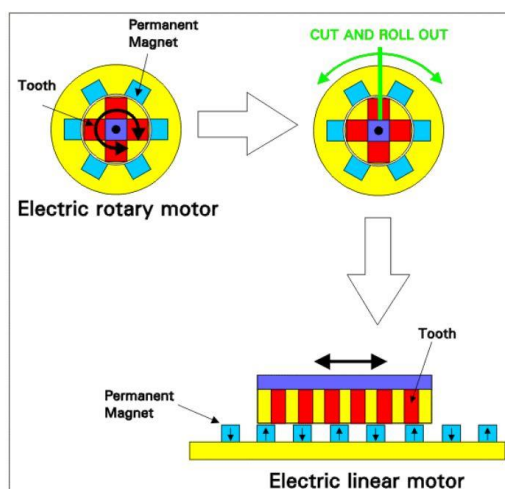
(Beckhoff, n.d.g)



Kuva 13. Lineaarimoottorijärjestelmän kokonaisuus. (Beckhoff, n.d.h)

### 3.2.3 Lineaarimoottorin ja servomoottorin ero

Lineaari- ja servomoottorin toiminta perustuu samoihin fysiikan lainalaisuuksiin, mutta rakenteellisena erona on huomattavaa, että pyörivän moottorin staattori ja roottori on kuin aukaistu ja levitetty pitkittäin. Pyörivä liike on siis muutettu lineaariseksi ja vääntömomentti voimaksi. Ao. havainnoillistava kuva 14.



Kuva 14. Pyörivä moottori, sekä lineaarimoottori. (Beckhoff, n.d.i)

Lineaarimoottorin suoraohjaus (direct drive) mahdollistaa hyvin tarkan paikkasäädön ja vasteen muutosnopeuksille. Verrattuna pyörivään moottoriin, lineaarimoottorin korkea hetkellinen voimantuotto suhteessa moottorin hitausmomenttiin mahdollistaa suuren kiihtyvyyden ja nopeuden. (Beckhoff, n.d.i)

### 3.3 Moottorin kommutointi

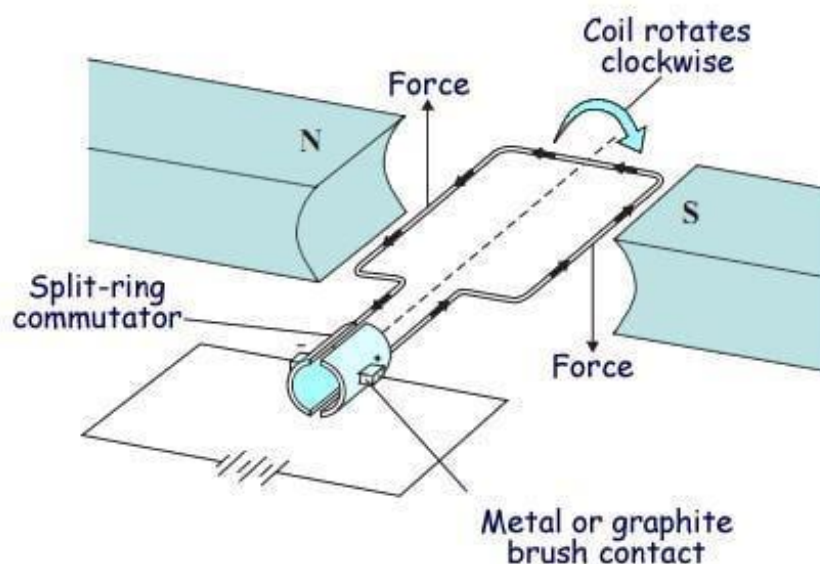
Servomoottorien korkea suorituskyky, tarkkuus, nopeus ja lämmönhallinta ovat hyvin suurissa määrin yhteydessä kommutointiin.

Kommutoinnilla tarkoitetaan moottorin staattorivirtojen ajamista käämistä toiseen. Jotta servomoottorin käyttö olisi tehokasta, on kommutointi harmonisoitava roottorin magneettikentän kanssa.

Harjallisissa moottoreissa kommutointi on helppo ymmärtää, kun harjat koskettavat kommutaattoriin ja vaihtavat käämivirtoja moottorin inertian avustaessa. Harjattomien moottoreiden teknologia on luotettavampaa, koska laakeroinnin lisäksi kuluvia osia ei ole, mutta kommutoinnin suorittama teknologia on hieman monimutkaisempaa.

#### 3.3.1 Mekaaninen kommutointi

Moottorit, jotka käyttävät liukurengasteknologiaa muuttuvan magneettikentän luomiseen luovat vaihtuvan virran käämitykseen liukuvan kontaktin avulla, joka on geometrisesti suunniteltu muuttamaan syötön napaisuutta tietyssä akselikulmassa.



Kuva 15. Kommutointivirran muutos. (MouserElectronics, n.d.)

Kuvan 15 mukaisesti, magneettikentän kenttäviivat kulkevat pohjoisnavasta etelänapaan kohden. Kun magneettikentässä olevaan

silmukkaan kytketään virta, kohdistuu siihen magneettinen voima, joka on suurimmillaan johdinsilmukan ollessa magneettikentän suuntainen.

Kun silmukan asento lähestyy SIN 0 astetta, on siihen kohdistuva magneettinen voima pienimmillään ja tällöin inertia jatkaa sen liikettä kuolleen kohdan yli, jolloin kommutaattori kääntää silmukassa kulkevan virran suunnan ja silmukkaan kohdistuu taas kasvava voima. (HSC Physics, n.d)

Silmukkaan kohdistuva voima saadaan laskettua kaavasta:

$$F = nBIl \quad (2)$$

Missä

- $F$  = Silmukkaan kohdistuva magneettinen voima
- $n$  = Johdinsilmukoiden määrä
- $B$  = Magneettivuon tiheys
- $I$  = Silmukassa kulkeva virta
- $l$  = Johtimen pituus

### 3.3.2 Sähköinen kommutointi

Nykyiseltään moottorit ovat lähes poikkeuksetta ohjattu tuottamaan vaihteleva magneettikenttä elektronisen ohjauksen avulla, jossa tarvetta geometrisesti oikein rakennetulle kommutaattorille ei ole. Moottorin tyyppi ja takaisinkytkentäteknikka päättävät kommutointitavan. Jotta kommutointi olisi mahdollista elektronisella ohjauksella, on oikea kommutointikulma saavutettava. (Beckhoff, n.d.j)

Kommutointikulman asettaminen on riippuvainen takaisinkytkentäjärjestelmästä, johon käytettäviä menetelmiä käydään läpi luvussa 7.3.

### 3.4 Servovahvistin

Servovahvistin on laite, jonka tarkoituksena on mitata signaalia takaisinkytkentäjärjestelmästä, verrata sitä ohjearvoon ja säätää moottorille syötettävää virtaa niissä määrin, jotta haluttu ohjearvo toteutuu. Servovahvistin ottaa moottoriin syötettävän energiansa pääsääntöisesti vaihtovirtana ja muuntaa sen tasajännitteeksi, joka syötetään määriteltynä pulssina moottorille. Logiikkakontrollerin tehtävänä on laskea ja määrittää paikoitus ja ehdot, joilla se ohjeistaa vahvistimen ohjaamaan moottoria.

Servovahvistimen ensisijaista suuretta säätäväksi operointitavaksi voidaan määrittää:

Virtasäätö, nopeussäätö ja paikkasäätö. (Beckhoff, n.d.k)

Servovahvistimen suorittama toimenpide kuulostaa yksinkertaiselta, mutta vahvistimeen kätkeytyy paljon ominaisuuksia ja modulaarisuutta, mitkä eivät kaikille ole itsestäänselvyksiä. Seuraavissa kappaleissa on listattu AX5000 servovahvistimen ominaisuuksia.

### 3.4.1 Moottorityypit ja takaisinkytkentäprotokollat

AX5000 mahdollistaa useiden eri moottorityyppien ohjauksen normaaleista epätahtimoottoreista raudattomiin lineaarimoottoreihin:

- Harjattomat tahtimoottorit – Servoajo takaisinkytkennällä
- Vääntömoottorit
- Lineaarimoottorit (ilman rautasydäntä) - Servoajo takaisinkytkennällä
- Lineaarimoottorit (rautasydämellä) - Servoajo takaisinkytkennällä
- Epätahtimoottorit – Taajuusmuuttajakäyttö takaisinkytkennällä, Korkeataajuinen akseliajo 60000 rpm asti, servoajo takaisinkytkennällä.

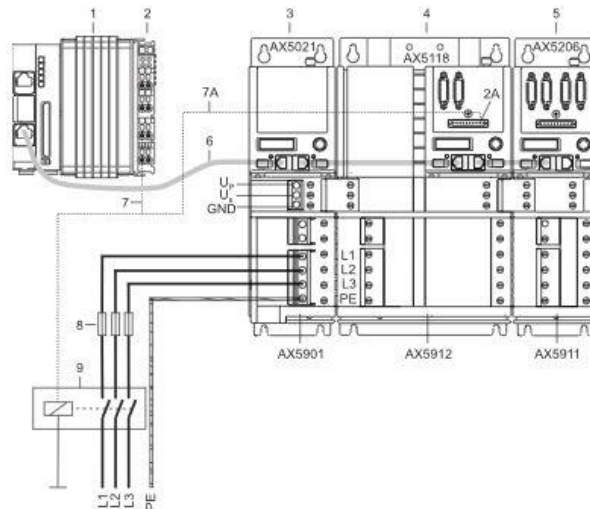
AX5000 tarjoaa rajapinnan kaikille yleisimmille takaisinkytkentäprotokollille:

- Sine / cosine 1 V<sub>pp</sub>
- EnDAT, Singleturn, Multiturn
- Hiperface, Singleturn, Multiturn
- Biss, Singleturn, Multiturn
- Resolver, 2-napainen, 8-napainen
- TTL
- Elektronisen tuotekilven tuki

### 3.4.2 Virransyöttö ja vahvistimien siltakytkentä

Kaksikanavaiset AX5000-sarjalaiset tukevat muuttuva-arvoista virransyöttöä, joten on mahdollista käyttää kahta nimellisvirraltaan eriarvoista moottoria, tarkoittaen: kaksikanavaisella vahvistimella, joka on nimellisvirroiltaan 2 X 6A (6A kanavavirta), voidaan ajaa esim. epätahtimoottoria (3A), sekä servomoottoria (9A). Vain kokonaisvirralla on merkitystä. (Beckhoff, n.d.l)

Useita AX5000 sarjan vahvistimia voidaan liittää yhtään siltausyksikön avulla. Sen avulla saadaan jaettua tehonsyöttö, DC link, ohjaus- ja jarrujännite jokaiselle moduulille. Kytkemissekvenssin on kuitenkin oltava laskeva; nimellisvirraltaan isoin vahvistin on oltava ensimmäisenä, josta laskevasti jatketaan viimeiseen. Kuva 16 havainnoillistaa ohjeistuksen mukaista siltakytkentää.



Kuva 16. Vahvistimien siltaus esimerkki (Beckhoff, System Manual Servo Drives AX5000, 2018.a)

Kaapeloinnin on täytettävä EN 60204-1 mukaiset standardit.

### 3.5 Turvakortti AX5805

AX5000 –sarjan servovahvistimille on mahdollista asentaa laajenuksena TwinSAFE turvakortti AX58xx. Kuten TwinSAFE-turvalogiikan ohjelmointi ja parametointi, on myös AX5805 parametointi tehtävissä suoraan TwinCAT-järjestelmässä. Järjestelmäkohtaiset turva-asetukset tallentuvat AX5805 aloitusparametreihin, tai TwinSAFE-turvaterminaaliin EL6900, jonka takia turvakortti voidaan vaihtaa muuttamatta ohjelmaa ja parametrit latautuvat kortille käynnistyksen yhteydessä. Systemistä ja liikeohjauksen luonteesta riippuen turvakortti mahdollistaa useiden erilaisten turvareaktioiden määrittämisen ohjausjärjestelmään, joiden avulla voidaan monitoroida tietynlaista mekaniikkaa. Näitä ovat esimerkiksi paikan, nopeuden, kiihtyvyyden ja suunnan monitorointi, joista kaksi toimintoa on kuvattu alla:

Safe Torque Off Function (STO) saattaa moottorin vääntövapaaseen tilaan STO-toiminnon aktivoituttua. Toiminto voidaan määrittää toteutuvaksi esimerkiksi valoverhojen aktivoituttua tilanteessa, jossa henkilö kävelee valoverhon väliin. (Beckhoff, AX5805 Manual, 2017.ö, s. 43)

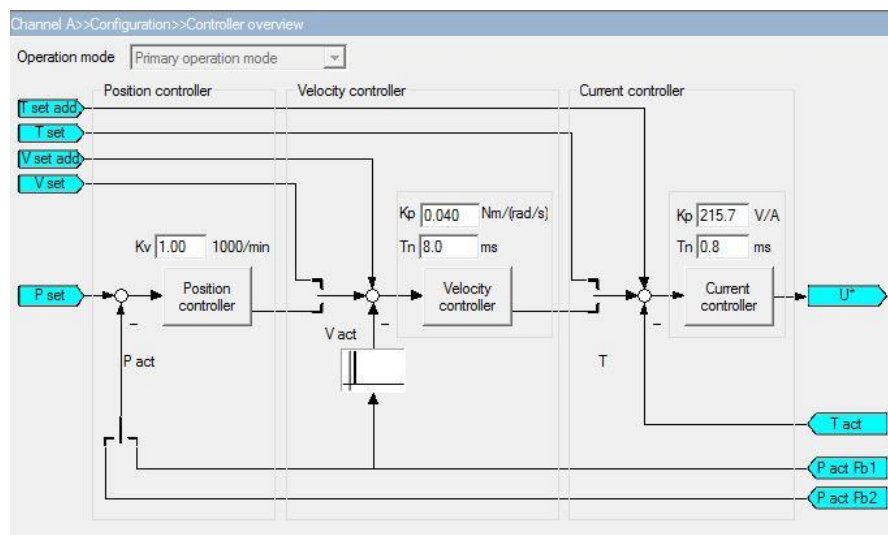
Safe Maximun Acceleration Function (SMA) monitoroi moottorin maksimikiihtyvyyttä suuntiin positiiviseen ja negatiiviseen suuntaan; mikäli jokin määritetyistä rajoista ylitetään, toteutuu määritetty virhereaktio ErrorReaction\_SMA. Toiminnon tapahtuma voi tapahtua tilanteessa, jossa moottorin on tarkoitus hidastaa kuormaa hallitusti, mutta kuorma pääseekin hallitsemattoman nopeaan kiihtyvyyteen, jolloin SMA-toiminto pysäyttää kuorman. (Beckhoff, AX5805 Manual, 2017.ö, s. 87)

### 3.6 Servojärjestelmän viritys

Servomoottorin virityksellä tarkoitetaan ohjaussäätimen optimoimista takaisinkytkentäjärjestelmältä saatujen tietojen käsittelyyn. Liikkeestä on tarkoitus saada nopea, tasainen, mahdollisimman äänetön, sekä värähtelemätön. Virityksen avulla pyritään saamaan moottori ajamaan yhdestä pisteestä toiseen mahdollisimman nopeasti ja tarkasti, sekä estää moottorin ylikuumentumista.

Servojärjestelmän virittämisessä on tärkeää huomioida moottorin kuorma. Virittämisen pitäisi tapahtua massaltaan ja ominaisuuksiltaan sellaisella kuormalla, joka vastaa sen tulevaa käyttötarkoitusta. Iso massa akselilla saattaa korjata värähtelyvirheitä piirissä, kun taas moottoria pyörittävä hihna saattaa venyä ja pysähtymisvaiheessa inertia vie massan ja vedon pienellä viiveellä toiseen suuntaan, joka voi saada ohjausjärjestelmän värähtelemään jatkuvan virhekorjauksen takia. Täten myös nopeuden, sekä kiihtyvyyden tulisi vastata käyttöympäristöä. (MotionControlTips, 2016.b)

Servomoottorin säätäminen TwinCAT 3 ympäristössä tapahtuu hyödyntäen TwinCAT Scope ohjelmaa, jonka toiminto muistuttaa PC-oskilloskooppia. Moottorin säätöpiirien järjestys on, kuten ao. kuvassa: virtasäätö, nopeussäätö, paikkasäätö. Ensimmäisenä virtasäädön on oltava kunnossa, jotta voidaan tehdä nopeussäätö, jonka jälkeen nopeussäädön on oltava kunnossa tehdäkseen paikkasäätö. Säätöpiiri kuvattuna kuvassa 17.

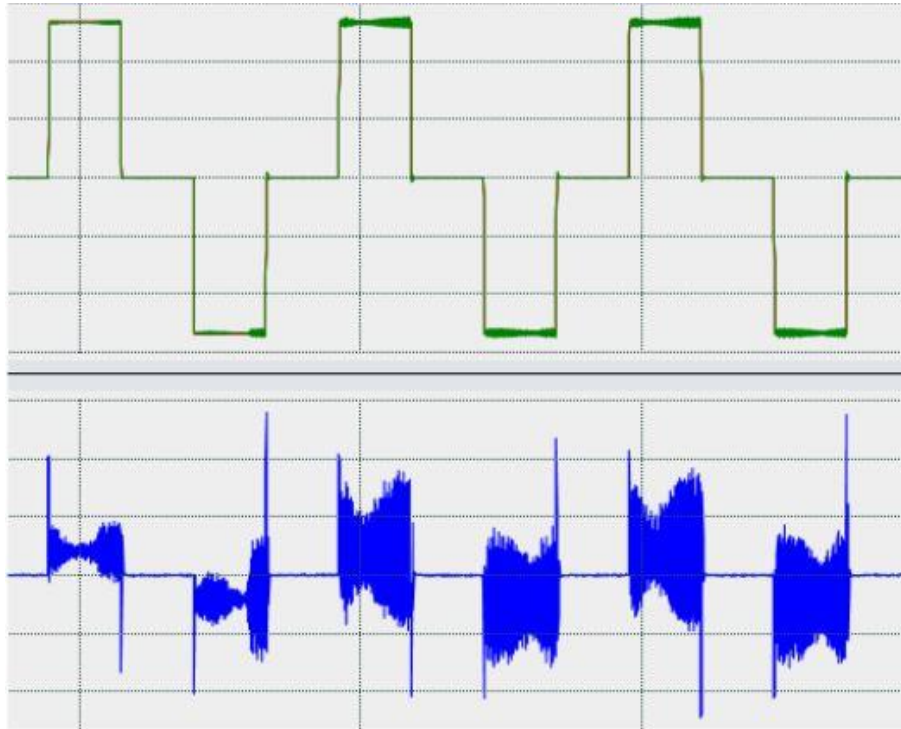


Kuva 17. Moottorin säätöpiiri. (Beckhoff, AX5000 Tuning Guide, n.d.bc)

Tämä esimerkki koskee nopeussäätöä. TwinCAT 3 Scopen avulla otetaan monitoroitavaksi nopeussäätimen voima-arvoa, nopeuden pyyntiarvoa, sekä nopeuden todellista arvoa, hyödyntäen Drive Managerin sekvenssijoa Velo Step. Paikkasäätimen vahvistus  $K_v$ , sekä nopeussäätimen integrointi-aika  $T_n$  asetetaan arvoon 0, jonka jälkeen



pyritään asettamaan moottori oskilloivaan tilaan. Ao. kuva 18 havainnoillistaa oskilloivaa nopeus-(ylempi) ja voimasignaalia(alempi).



Kuva 18. Nopeus- ja voimasignaali. (Beckhoff, AX5000 Tuning Guide, n.d.bc)

Kun moottori on säädetty tasaisesti oskilloivaksi, niin voidaan päätellä Zieglerin ja Nicholsin sääntöjen mukaan, oikean nopeusvahvistuksen  $K_p$  olevan:

$$K_p = 0,45 \times K_{pcrit} \quad (3)$$

Missä:

$K_p$  = Vahvistuksen oikea arvo

$K_{pcrit}$  = Vahvistuksen oskilloiva arvo

Kun nopeusvahvistus on saatu sopivaksi, niin asetetaan integrointiaika. Integrointiaika  $T_n$  vaikuttaa oskillointiin, joten sitä säätäessä on syytä seurata nopeussäätimen stabiiliutta. Integrointiajan määrittäminen riippuu moottorin ja sen kuorman yhdistämisestä; mitä elastisempi liitos, sen isompi integrointiajan tulisi olla. Automaatiovalmistaja Beckhoffin mukaan yleisesti normaalit järjestelmät toimivat 8ms integrointiajalla, mutta jos on syytä pudottaa integrointiaikaa vieläkin pienemmäksi, on syytä pitää silmällä virta- ja nopeussäätimen oskillointia. (Beckhoff, AX5000 Tuning Guide, n.d.bc)

## 4 TWINCAT 3

TwinCAT3 on Visual Studioon integroitu automaatio-sovelluksien kehitysympäristö. Se kykenee hyödyntämään moniydinprosessorien sekä 64-bittisten käyttöjärjestelmien tarjoamat edut, sekä kernel-laajennus mahdollistaa sovellusten ajamisen reaaliaikaytimessä. TwinCAT tarjoaa IEC 61131-3 –mukaiset ohjelmointikieliset, sekä rajapinnat kaikkiin yleisimpiin kenttäväyläprotokollisiin TwinCAT kykenee mm. 50µs kiertoaikaan ja nanosekuntiluokan signaalimittauksiin. Samalla järjestelmällä onnistuvat kaikki koneautomaation avaintoiminnot: logiikka, liikkeenohjaus, käyttöliittymät, turvatoiminnot ja MATLAB/Simulink ja C/C++ -integraatio. (Beckhoff, n.d.m)

### 4.1 Ohjelmointikieliset

TwinCAT 3 ohjelmointiympäristö tukee kaikkia IEC 61131-3 standardin ohjelmointikieliä.

- Instruction List
- Structured Text
- Function Block Diagram
- Ladder Diagram
- Continuous Function Chart
- Sequential Function Chart

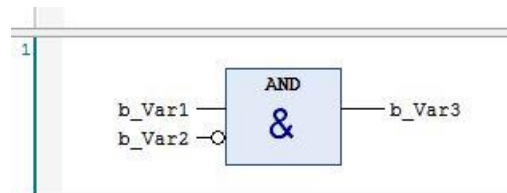
**Instruction List** on matalan tason ohjelmointikieli ja sen suoritus logiikkakontrollerissa on nopeampaa, kuin graafisen ohjelmointikielen suorittaminen, sekä se vaatii myös vähemmän muistia käyttöönsä. IL muistuttaa tikapuukaaviota tekstisyötteellä; siinä annetaan ohjelmalle toimintaohje rivi kerrallaan. Kukin ohje(expression) koostuu operaattorista ja halutusta operaatiosta riippuen yhdestä, tai useammasta operandista.

**Structured Text** on korkean tason PLC-ohjelmointikieli, joka on tekstisyötteistä ja verrattavissa ohjelmointikieliin PASCAL ja C. ST:n avulla on hyvin paljon yksinkertaisempaa toteuttaa monimutkaisia funktioita, kuin esim Ladder Diagrammilla, tai Instruction Listillä. Ohjelmointi koodi koostuu lauseiden ja ohjeiden yhdistelmästä, jotka voivat olla ehdollisia, tai silmukoita, jotka tekevät ohjelman toteutuksesta kompaktimpaa. Structured Text Editor, johon ohjelmakoodi syötetään mahdollistaa myös koodin helpon dokumentoinnin mahdollistamalla kommentoinnin mihin kohtaan tahansa. Structured Text esimerkki kuvassa 19.

```
IF b_Var1 AND NOT b_Var2 THEN
    b_Var3 := TRUE; ELSE
    b_Var3 := FALSE;
END_IF
```

Kuva 19. Structured Text (TwinCAT 3, 2019).

**Function Block Diagram** on graafinen ohjelmointikieli, jossa lohkot liitetään toisiinsa yhteen sekvenssiin, jota on helppo seurata kulkemalla lohkoa jatkuvaa linjaa seuraavaan lohkoon. Sen toiminta muistuttaa Ladder Diagramia, mutta visuaalinen olemus on erilainen. FBD sopii hyvin yksinkertaisten ohjausten tekemiseen, sekä on loppukäyttäjää ajatellen yksinkertainen ymmärtää. Esimerkki Function Block Diagramista kuvassa 20.



Kuva 20. Function Block Diagram. (TwinCAT 3, 2019).

**Ladder Diagram** on graafinen ohjelmointikieli, joka muistuttaa graafista esitystä relekaaviosta. FBD:n tapaan se on hyvin yksinkertaista ymmärtää, jopa ilman aiempaa kokemusta PLC-ohjelmoimisesta. Ladder diagrammin kulku on vasemmalta oikealle, joka voidaan kuvitella virtaliittiminä, jonka väliin releet, lamput ja muut komponentit asetetaan. Esimerkki Ladder Diagramista kuvassa 21.



Kuva 21. Ladder Diagram. (TwinCAT 3, 2019).

**Continuous Function Chart** on graafinen ohjelmointikieli, jonka ohjelmasyötteessä on vapaasti kelluvia lohkoja. Lohkot on liitetty toisiinsa riveillä lähdöistä tuloihin, joka määrittelee ohjelman logiikan. Takaisinkytkennän määrittäminen vapaan lohkojen sijoittelun takia on helppoa ja lohkojen suorituserjestys on määriteltävissä.

**Sequential Function Chart** ohjelmointikieli muistuttaa Flowchartia, jossa ohjelman suoritus lähtee ylhäältä (init) ja jatkuu alaspäin, jonka jälkeen suoritetaan toimilohkoja järjestyksessä ja ehtojen täytyessä siirrytään aina seuraavaan lohkoon alaspäin, tai suoritetaan ohjelmaan määritetty siirtyminen haluttuun askeleeseen. (Beckhoff, n.d.n)

## 4.2 TwinCAT Drive Manager

TwinCAT ohjelmointiympäristö tarjoaa servovahvistimien parametroiden käyttöliittymän nimeltään Drive Manager.

Drive Manageriin on tallennettu jo valmiita parametreja tietyn mallisille moottoreille, sekä takaisinkytkentäantureille. Uusimpien moottorien parametrit tulevat automaattisesti valituksi IO-väylän skannauksen yhteydessä. Vanhemmilla moottoreilla on määriteltävä itse moottorin tyyppinumeron mukaisesti takaisinkytkentä, sekä moottori. Järjestelmä mahdollistaa myös epätahtimoottoreiden ajamisen ilman takaisinkytkentää. AX5000-sarjan vahvistimella on mahdollista ajaa samalla vahvistimella servomoottoria suljetussa piirissä, sekä epätahtimoottori avoimessa piirissä yhtäaikaaisesti. (Beckhoff, n.d.o)

## 4.3 Käyttöliittymän suunnittelu

Lineaaridemon käytettävyyden, sekä demonstroinnin kannalta käyttöliittymä näyttelee merkittävää roolia. Sitä kautta on pystyttävä tekemään kaikki laitteen toiminnallisuuden kannalta merkittävät muutokset, sekä mahdollistaa tiedonsaanti. Liittymän on oltava käyttäjäväläinen siten, että demolaitetta voi operoida kuka tahansa henkilö aiheuttamatta vaaratilanteita laitteistolle, tai läsnäolijoille.

HMI(Human Machine Interface) on käyttöliittymä, joka mahdollistaa ohjausrajapinnan käyttäjän ja koneen, laitteiston, tai järjestelmän välillä. Vaikka termiä voidaan teknisesti soveltaa mihin tahansa näyttöön, jonka avulla käyttäjä voi lähettää ja vastaanottaa tietoja laitteen kanssa, HMI: tä käytetään yleisimmin teollisen prosessin yhteydessä. (InductiveAutomation, n.d.)

Käyttöliittymän tarjoamat edut loppukäyttäjän kannalta ovat merkittäviä. Järjestelmän käytettävyys olisi hyvin tehontota, mikäli esim. tuotantolinjalla toimiva operaattori joutuisi syöttämään, sekä keräämään kaiken tarvitsemansa tiedon PLC:n ohjelmointiympäristöstä. Käyttöliittymällä tavoiteltaviin etuihin kuuluu virheen vähentäminen, järjestelmän ja käyttäjän tehokkuuden parantaminen, luotettavuuden ja ylläpidon parantaminen, käyttäjien käyttömukavuuden lisääminen, koulutuksen ja taitovaatimusten vähentäminen, sekä käyttäjien fyysisen ja henkisen stressin vähentäminen ja tuotannon tehostaminen. (InductiveAutomation, n.d.)

Visualization mahdollistaa PLC-ohjelman liittämisen visuaaliseen ympäristöön. TwinCAT 3 kehitysympäristö sisältää integroidun visualisointieditorin, jonka avulla käyttäjä voi luoda visualisointikohteita samaan käyttöliittymään sovelluksen kehittämisen rinnalle. Visualisointi on siis graafinen toteutus projektin toiminnasta, jonka avulla projektin diagnostiikan seuranta, sekä muuttujien arvojen muuttaminen on toteutettavissa kosketusnäytön, hiiren ja näppäimistön avulla.

Riippuen ohjattavasta automaatiojärjestelmästä, visualisointi mahdollistaa käytettävillä laitteilla omintakeiset elementit käyttöön; kytkimet, lamput, mittalaitteet, potentiometrit. Visualisointi mahdollistaa useiden sivujen luomisen, joten tilan loppuessa on mahdollista siirtyä aina seuraavalle sivulle. (Beckhoff, n.d.p)

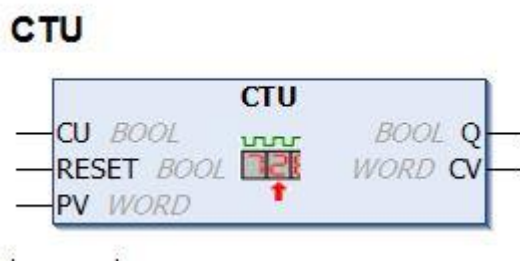
#### 4.3.1 TwinCAT 3 PLC HMI

TwinCAT 3 PLC HMI on runtime-järjestelmän laajennus, joka mahdollistaa visualisoinnin suorittamisen kehitystietokoneella, tai kohdekoneella (PLC), jossa ei ole kehitysympäristöä. Visualisointikoodi luodaan olemassa olevien visualisointiobjektien perusteella ja ladataan ohjaavaan tietokoneeseen. Käyttämällä PLC HMI:tä visualisoinnin suorittamiseen kohdekoneella, säästyy muistia huomattavasti, koska kehitysympäristöä ei tarvita. Tämä kannattaa huomioida kevyemmissä tietokoneissa. (Beckhoff, n.d.q)

Visualisointi voidaan myös suorittaa web-selaimessa. PLC HMI Web toteuttaa visualisoinnin javascriptinä, joka pyytää näytettävän tiedon web-serveriltä. Kaikki selainpohjaiseen visualisointiin tarvittavat tiedostot tallentuvat TwinCAT-hakemistoon; java-komentosarjan, HTML-sivun, näytettävät kuvakkeet. (Beckhoff, n.d.r)

#### 4.4 Ohjaavat toimilohkot

Toimilohkot ovat PLC-ohjelmointiin suunniteltuja työkaluja nopeuttamaan ja selkeyttämään ohjelman rakennetta. toimintolohkot perustuvat ennalta määritellyn koodin segmentteihin, jotka on suunniteltu suorittamaan tietty toistuva tehtävä. Niiden avulla on helppo ja nopea tapa määrittää ohjaimen käyttäytymisrutiineja, luoda erilaisia toimintoja, asettaa vertailuja, tai luoda laskureita. Ao. kuvassa 22 on esitelty yksinkertainen IEC61131-3 standardoitu toimilohko CTU, joka laskee pulssien määrää inputista CU ja antaa arvon outputista CV. (UaAutomation, n.d.)



Kuva 22. Toimilohko CTU (Beckhoff, n.d.s)

Toimilohkon toimintasuunta kuvataan vasemmalta oikealle. Vasemmalle puolelle on sijoitettu tulot ja oikealle lähdöt.

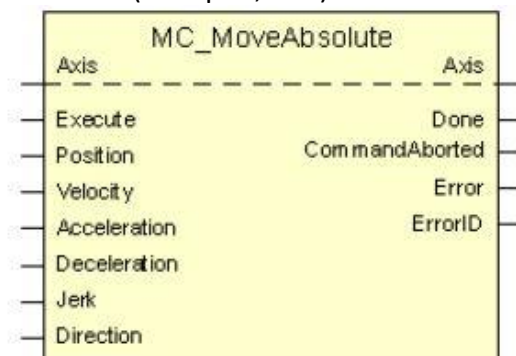
Toimilohkot voivat yksinkertaisimmillaan seurata pulssin nousevaa, tai laskevaa reuna yhdessä tuloportissa ja ilmoittaa pulssien summan lähdössä. Toimilohkon hyödyllisyys on käytännössä universaali ja sen toiminta riippuu sen sisään rakennetusta ohjelmasta. Logiikkaohjaimien kehitysympäristöt tarjoavat mahdollisuuden rakentaa omanlaisia toimilohkoja, mutta mahdollistavat myös valmistajakohtaisten toimilohkojen käytön, jotka ovat erityisesti suunniteltu toteuttamaan valmistajan laitteiston toiminnallisuuteen vaikuttavia toimintoja.

#### 4.4.1 PLCopen

Liikeohjauksen kannalta valmiiden ja hyvin suunniteltujen toimilohkojen käyttö näyttää merkittävää roolia sovellusten rakentamiseen käytetyssä ajassa ja varmuudessa. Toimilohkojen rakentaminen jokaista eri sovellusta varten veisi hyvin paljon aikaa, sekä vaatisi järjestelmän luojalta huomattavasti suurempaa pätevyyttä. Monet laitevalmistajat ovat luoneet omille järjestelmilleen omia kirjastoja omia sovelluksiaan varten, jotka eivät kuitenkaan ole käytettävissä toisen osapuolen järjestelmissä.

PLCopenin liikkeenohjaus-standardi tarjoaa tavan käyttää vakio-sovelluskirjastoja, jotka ovat uudelleenkäytettäviä useille laitteistoalustoille. Tämä alentaa kehitystyötä, ylläpitoa ja tukikustannuksia samalla järjestelmän pysyessä selkeämpänä. Lisäksi suunnittelu helpottuu, koulutuskustannukset laskevat ja ohjelmisto on uudelleenkäytettävissä eri alustoilla. Tämä standardointi tehdään tehokkaasti määrittelemällä uudelleenkäytettävien komponenttien kirjastot ja tällä tavoin ohjelmointi on vähemmän laitteistosta riippuvaista, sekä uudelleenkäytettävyys kasvaa. Tietojen piilottamisen ja kapseloinnin ansiosta se on käyttökelpoinen erilaisissa arkkitehtuureissa, esimerkiksi keskitetysti hajautettuun tai integroituun verkko-ohjaukseen. Sitä ei ole erityisesti suunniteltu yhdelle sovellukselle, vaan se toimii peruserroksena eri alojen toiminnassa oleville määritelmille. Siten se on avoin olemassa oleville ja tuleville teknologioille. Myös automaatiovalmistaja Beckhoff on ottanut käyttöönsä PLCopen toimilohkot käyttöön liikkeenohjauksessa, jotka ovat valmiina käytettävissä TwinCAT 3 kehitysympäristössä Tc2\_MC2-kirjastossa. (Beckhoff, n.d.t)

Ao. kuvassa 23 on esitelty PLCopen-mukainen toimilohko MC\_MoveAbsolute, jonka tarkoitus on ohjata moottori määriteltyyn position. (PLCopen, n.d.)



Kuva 23. Toimilohko MC\_MoveAbolute (Beckhoff, n.d.u)

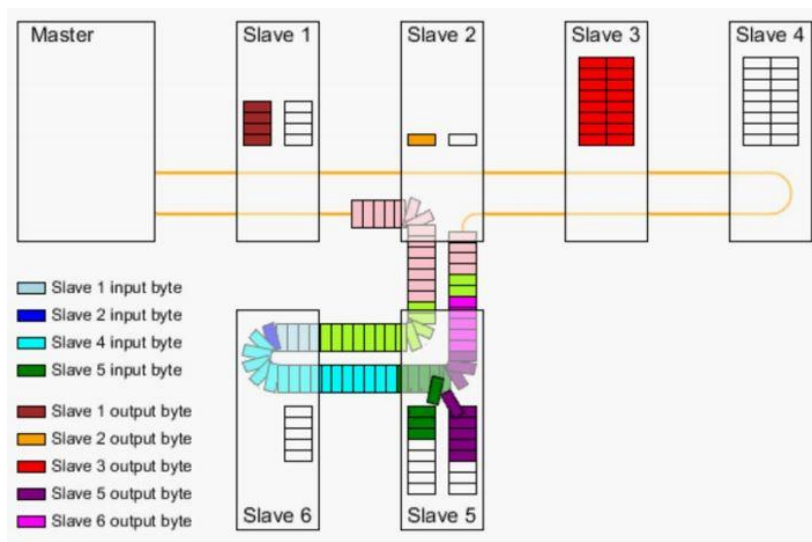
**Liite 1/1, 1/2** esittelee PLCopen toimilohkoja yksittäisen akselin ohjelmointiin.

## 5 KENTTÄVÄYLÄT

### 5.1 EtherCAT

EtherCAT on Beckhoffin kehittämä reaaliaikainen kenttäväyläteknikka, joka on esitelty standardissa IEC-61158. EtherCAT-protokolla soveltuu reaaliaikaisuutta, lyhyitä sykli-aikoja ja lyhyitä viiveitä vaativille automaattioratkaisuille. Syklisen toiminnan periaate on, että EtherCAT-isäntä lähettää sanoman, joka kulkee kunkin haaran läpi ja jokainen EtherCAT-orja lukee sille osoitetun sanoman lennosta ja lisää I/O-laitteen kenttätiedon sen kehykseen kehyksen liikuessa orjalaitteelta toiselle. Kehys viivästyy vain laitteistokeskeisen käsittelyn verran, eikä syklien välistä hajontaa (jitter) tapahdu. Segmentin viimeinen solmu havaitsee avoimen portin ja lähettää viestin takaisin isäntälaitteelle käyttäen Ethernet-teknikan kokodupleksi-toimintoa. EtherCAT-teknikalla on päästy yli Ethernetin rajoituksista reaaliaikaisuuteen liittyen. Prosessi on orjalaitteistoon implementoitu ohjaus ja on siten riippumaton ohjelmistojen ajoajoista tai prosessorin tehosta. (EtherCAT Technology Group, n.d.ca)

Ao. kuvassa 24 on esitelty EtherCAT-kehys, joka siirtyy isännältä orjille, jotka lennosta lukevat kehyksestä niille tarkoitetun tiedon, sekä jakavat tietonsa.

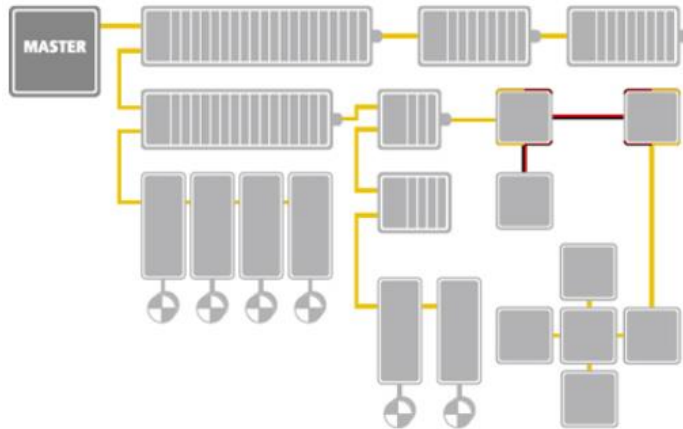


Kuva 24. EtherCAT-frame. (EtherCAT Technology Group, n.d.ca)

EtherCATin suorituskyky nopeutta ja tarkkuutta vaativissa ohjauksissa on kiistaton: 1000 hajotetun I/O:n datan päivitysaika on vain 30  $\mu$ s - mukaan lukien päätelaitteiden sykli-aika. Jopa 1 486 tavua prosessidataa voidaan vaihtaa yhdellä Ethernet-kehyksellä - tämä vastaa lähes 12 000 digitaalista tuloa ja lähtöä. Tämän datamäärän siirto kestää vain 300  $\mu$ s. Viestintä sadalle servoakselilla tapahtuu joka 100  $\mu$ s:n välein.



Tällä jaksolla kaikki akselit on varustettu asetetuilla arvoilla ja ohjaustiedoilla, ja ne ilmoittavat niiden todellisen sijainnin ja tilan. Hajautetun kellotekniikan avulla akselit voidaan synkronoida alle 1 mikrosekunnin viiveen vaihteluun. EtherCAT-topologia voidaan toteuttaa linja-, tähti-, tai puu –tekniikalla, sekä yksi segmentti mahdollistaa 65535 laitteen kytkennän. Ao. kuvassa 25 havainnoillistava kuva topologiasta. (EtherCAT Technology Group, n.d.ca)

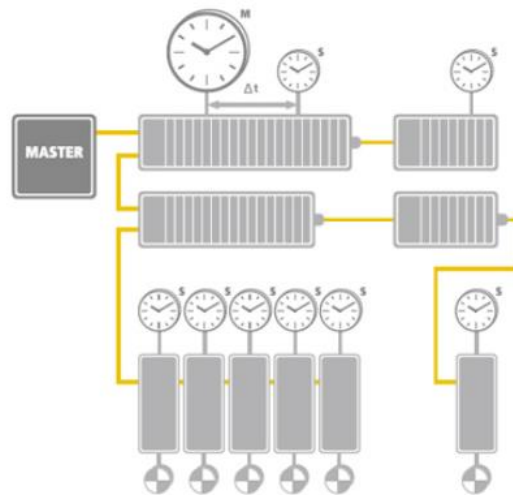


Kuva 25. EtherCAT Topologia. (EtherCAT Technology Group, n.d.ca)

#### 5.1.1 Jaettu Kello (Distributed Clock)

Referenssikellosta lähetetty aika saapuu orjalaitteisiin pienellä viiveellä, tämä viive on mitattava ja kompensoitava jokaiselle orjalaitteelle synkronisuuden ja samanaikaisuuden varmistamiseksi.

Aika ensimmäisestä DC-orjalaitteesta jaetaan syklisesti kaikkiin muihin laitteisiin järjestelmässä. Tällä mekanismilla orjalaitteiden kellot voidaan säätää tarkasti referenssikellon aikaan, jolloin viestien vastaanottamisen ajoitus osuu juuri viestin saapumishetkeen. Ao. kuva 26 havainnoillistaa Distributed Clock-tekniikkaa. (EtherCAT Technology Group, n.d.ca)



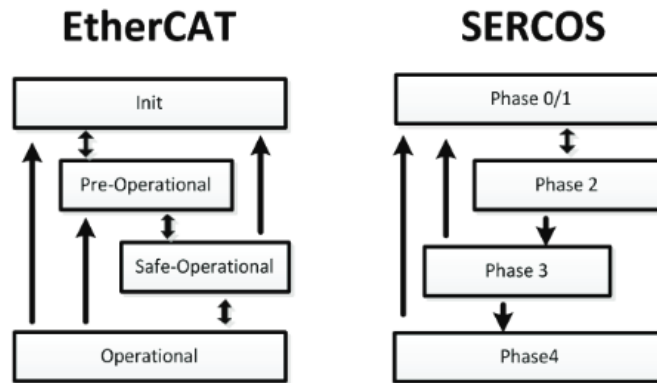
Kuva 26. EtherCAT Distributed Clock. (EtherCAT Technology Group, n.d.ca)

## 5.2 Servo Profile over EtherCAT

Servo Profile over EtherCAT (SoE) integroi SERCOS-teknologiaan perustuvat mallit EtherCAT-ympäristöön. Se sisältää IEC 61491 -tilakoneen, synkronoinnin, prosessidatayhteyden ja pääsyn palvelukanavan kautta yksilölliseen tunnusnumeroon (IDN), joka tarjoaa elementtien tiedot; datatila, atribuutti, nimi, yksikkö, min, max ja arvo. (Motion Engineering, n.d)

SERCOS-käyttöliittymä on IEC 61491 standardisoitu sarja- ja reaaliaikainen tietoliikenneprotokolla teollisuuden laitteistolle ja digitaalisten servolaitteiden välillä. Se tarjoaa kattavan ja tarkan määritelmän fyysisestä kerroksesta, tiedonsiirtokerroksesta, tiedonvaihtorakenteesta ja sisällöstä. Se tarjoaa myös kattavasti profiileja servokäyttöön, tietorakenteita ja prosesseja koskevia komentoja niihin liittyvien koneiden ja laitteiden käyttämiseksi. SERCOS-rajapinta käyttää rengas topologiaa, yksi ohjausyksikkö voi kuljettaa yhden tai useamman SERCOS-säätörengas ja ohjata enintään 254 servolaitetta. (ScienceDirect, n.d)

SERCOS ja EtherCATin yhdistämiseksi SERCOS-yhteysvaiheet on sijoitettu EtherCAT-tilakoneeseen: vaiheet 0 ja 1 kuuluvat EtherCATin tilakoneen Init-tilaan, vaihe 2 vastaa PreOp -tilaa ja sallii pääsyn SERCOS IDN:n EtherCAT-postilaatikon kautta, joka on mekanismi, joka toteuttaa kättelymekanismin tiedonvaihtoa varten ei-syklisesti. Vaihe 3 vastaa Safe-Op -tilaa, jolloin sykliset tiedot lähetetään ja ajurilla on aikaa synkronointiin. EtherCAT SafeOp-tila määrittää, että ESC:n (EtherCAT Slave Controller) täytyy toimittaa voimassa olevat input-tiedot ja sivuuttaa isännän outputit. Ao. kuva 27. havainnoillistaa EtherCAT-tilakonetta ja SERCOS-vaiheita. (Beckhoff, n.d.v)



Kuva 27. EtherCAT State Machine, Sercos Phases. (ScienceDirect, n.d, s. 752)

### 5.3 CoE, CANopen over EtherCAT

CANopen over EtherCAT (CoE) -protokolla mahdollistaa CANopen-profiilin hyödyntämisen EtherCATin kautta. SDO-protokollaa käytetään suoraan, jotta olemassa olevia CANopen-pinoja voidaan käyttää lähes muuttamattomina. Vaihtoehtoisia laajennuksia määritellään, jotka nostavat CAN-väylän 8-tavun rajaa ja mahdollistavat objektiluettelon täydellisen luettavuuden. EtherCATin ja CANopenin verkonhallinta ovat vertailukelpoisia keskenään, mutta myös viestintämekanismit: Object Dictionary (objektin sanakirja), PDO (prosessidatan objektit) ja SDO (palveludatan objektit). EtherCAT voidaan siis sulauttaa CANopenin kanssa varustettuihin laitteisiin. CANopen over EtherCAT hyödyntää siis jo olemassaolevia CANopen-profiileja ja on näin käytettävissä olemassaolevissa CANopen-laitteissa. (Beckhoff, n.d.w)

## 6 KONE- JA HENKILÖTURVALLISUUS

Tässä luvussa käsitellään laitteiston turvaratkaisun keskeisiä komponentteja. Demolaitteen automaattisen toiminnallisuuden vuoksi on kone- sekä henkilövahinkojen turvaamiseksi tehty riskianalyysi standardin SFS-EN ISO 12100 mukaan hyödyntäen teknisen raportin SFS-ISO/TR 14121-2 ohjeita tarvittavan suojan toteuttamiseksi. Standardin mukaisen turvaetäisyyksien mitoittamiseen on käytetty standardin SFS-EN ISO 13857 määrittämiä.

### 6.1 TwinSAFE

TwinSAFE on TwinCAT 3:n rakennettu ohjelmointimoduuli, jonka avulla voidaan huolehtia turva-automaatiosta ohjelmoimalla turvaprojekti TwinCAT 3 kehitysympäristössä System Managerissa. Kirjoitettu turvaohjelma toteutetaan erillään tavallisesta PLC-ohjelmasta. Turvaohjelman käsittely tapahtuu turvalogiikassa EL6900. Turvatulot kytketään EL1904 tulokortteihin ja turvalähdöt EL2904-lähtökortteihin, jotka ovat sertifioituja turvalaitteita. TwinSAFE:n kaltainen turvajärjestelmän perussääntö on, että järjestelmän osan, komponentin, tai koko järjestelmän pettäminen ei saa johtaa vaaralliseen tilanteeseen. Turvallinen tila on aina poiskytketty tila, tai energiaton tila. (Beckhoff, 2017.x, s. 13)

Safety-over-EtherCAT (FSoE, Fail Safe over EtherCAT) -protokolla määrittelee avoimen, suojatun tiedonsiirron. TÜV:n sertifioima teknologia on kansainvälisesti standardoitu ja täyttää IEC 61508 -standardin mukaiset turvallisuusvaatimukset SIL 3 (Safety Integrity Level) turvatasoon asti.

Tiedonkulku toteutetaan ns. "mustassa kanavassa" FSoE-kommunikoinnissa, tarkoittaen, että FSoE-protokollaa voidaan lähettää käyttäen mitä tahansa kommunikointitekniikkaa, kuten EtherCAT, Ethernet ja langaton tiedonsiirto, mutta myös tukien vakio kenttäväyläjärjestelmiä, kuten CAN. (Beckhoff, n.d.y, s. 7)

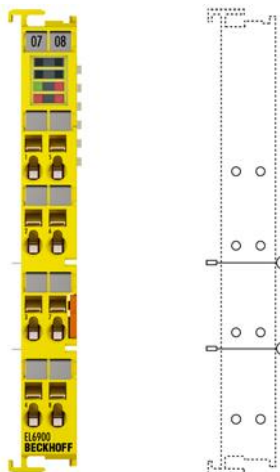
#### 6.1.1 Black Channel Protocol

SoE-kommunikointijärjestelmä on osa mustan kanavan protokollaa (black channel protocol). Riippumatta tavallisesta tietoliikenneväylästä, turvakomponentit voivat lähettää dataa käyttämällä erillistä turvallista protokollaa, joka tunneloi taustalla olevan verkon kanavan. EtherCAT-protokolla käyttää yhtä kanavaa siirtämään sekä vakio- että turvallisuuskriittisiä tietoja. Turvakehykset, jotka tunnetaan nimellä Turvakontit, sisältävät turvallisuuden kannalta tärkeitä prosessitietoja ja lisätietoja näiden tietojen turvaamiseksi. Turvakontit kuljetetaan osana viestinnän prosessitietoja. Riippumatta siitä, onko tiedonsiirto

turvallinen, ei riipu taustalla olevasta viestintätekniikasta eikä ole rajoitettu EtherCATiin. Koska safety-kenttäväylät ovat puhtaasti sovellusprotokollia, joilla ei ole omia fyysisiä ominaisuuksia, käytettävissä olevat kaistanleveydet ja kierrosajat riippuvat käytetystä datansiirtoprotokollasta. Mahdolliset lähetysvirheet on lueteltu tekniikkaan liittyvissä standardeissa IEC 61784-3 ja IEC 61508. Lähetysvirheiden ehkäiseminen on toteutettava olennaisena osana turvallisuustietoliikenneprotokollasta. (EtherCAT Technology Group, n.d.b)

## 6.2 Turvaprojekti

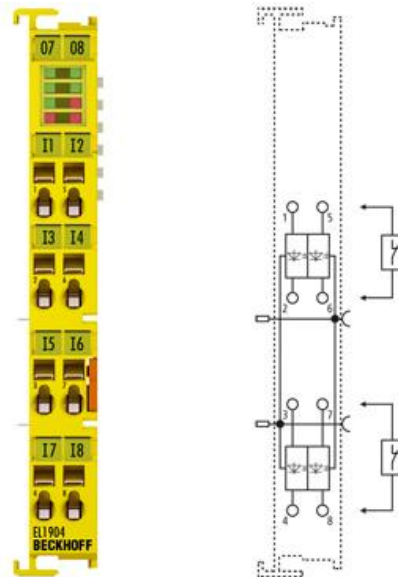
Laitteen turvallisesta käytettävyydestä huolehtii turvaprojekti, johon ohjelmoidaan tarvittavat turvalaitteistoon kohdistuvat ominaisuudet. Havainnoillistava kuva 28 EL6900-turvalogiikkakortista.



Kuva 28. Turvalogiikka EL6900.

EL6900 on turvalogiikkaterminaali, PLC, joka toimii väylänä TwinSAFE IO-korteille. Terminaalilla on mahdollista ohjata 128 turvalaitetta ja se tarjoaa 256 integroitua turvatoimilohkoa erilaisiin turvaohjelmien tarpeisiin, joita voidaan yhdistää toisiinsa, sekä rakentaa turvallisuuskeskeinen logiikka. (Beckhoff, n.d.z)

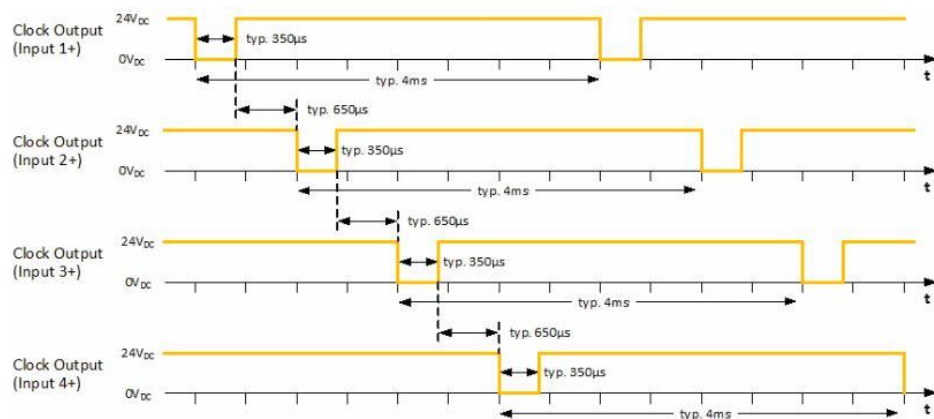
EL1904 on digitaalinen turvatulo-terminaali sensoreille, joissa on potentiaalivapaa kontakti 24VDC:lle. EL1904 sisältää neljä Fail-Safe –tuloa, joka tarkoittaa, että kanava valvoo jatkuvaa toiminnantarkastuspulssia (OSSD) varmentaa laitteen oikean tilan. Tämä on otettava huomioon parametreissa korttia EL1904 esitettynä kuvassa 29.



Kuva 29. EL 1904 Turvaterminaali. (Beckhoff, EL1904, 2017.x)

### 6.2.1 Signaalinvalvontapulssi

SICK C2000-valoverhojärjestelmä käyttää toiminnan tarkastamiseen OSSD-pulssia (Output Signal Switching Device). Toimivassa valvovassa tilassa valoverhojen kaksi lähtöä tuovat turvakortille EL1904 24V jännitteen, mutta samaan aikaan valoverhot lähettävät testipulssia, joka vie jännitteen hetkeksi tasoon 0V. Pulssin tarkoituksena on varmistaa, että turvatason jännite 0V kulkeutuu turvakortille vaaratilanteen ilmetessä. Kuva 30 havainnoillistaa tyypillistä turvalaitteen testipulssia kortille EL1904:



Kuva 30. OSSD Pulssi. (Beckhoff, 2017.x)

Kytkevävaihtoehdot vaihtelevat toiminnon soveltuvuuden mukaan, sovellettavia laitteita ovat esimerkiksi: . hätäseis-painikkeet kytkimet, turvamatot, valoverhot, laserskannerit. (Beckhoff, EL1904 Manual, 2017.x, s. 38)

### 6.3 Valoverho

ISO 12100 Standardi 6.3.2.4 ja 6.3.2.5 mukaisesti koneenkäyttäjän suojaamiseksi laitteiston käytön aikana suojaavaksi toimilaitteeksi on valittu valoverhojärjestelmä Sick C2000. (12100, ISO, n.d, s. 78)

Valoverhot ovat optoelektronisia laitteita, joita käytetään suojaamaan henkilöstöä automaattisesti liikkuvien koneiden ja laitteistojen läheisyydessä, jossa henkilö- tai laitteistovahingon riski on läsnä fyysisen kappaleen joutuessa laitteiston välittömään läheisyyteen. Valoverhoja voidaan käyttää vaihtoehtoisena suojausmenetelmänä mekaanisten esteiden ja muiden perinteisten koneiden suojausmuotojen sijaan. Vähentämällä fyysisten suojusten ja esteiden tarvetta, turva-valoverhot voivat lisätä suojattavien laitteiden ylläpitoa. Koneiden käyttökelpoisuutta ja tehokkuutta voidaan parantaa myös käyttämällä turvavaloverhoja esimerkiksi helpottamalla puoliautomaattisia menettelyjä. Sick C2000 täyttää koneturvallisuuteen perustuvien automaattisten turvajärjestelmien vaatimukseen soveltuvat standardit (IEC 61496), SIL1 (IEC 61508), PL c (EN ISO 13849). (SICK, n.d.da)

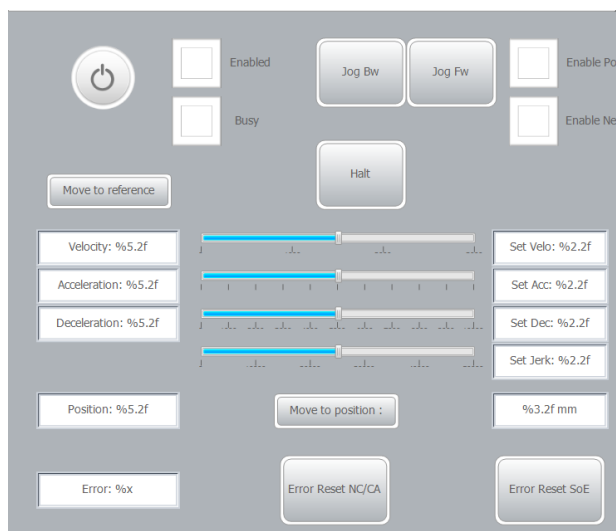
## 7 TYÖN TOTEUTUS

Seuraavissa osissa kuvataan työn etenemistä ja sitä, miten käytännössä työtä on lähdetty tekemään, mitä dokumentaatiota on hyödynnetty, sekä muuta huomion arvoista.

Työtä aloittaessa ei ollut muuta suunnitelmaa, kuin tehdä olemassa olevalle laitteistolle käyttöönotto, sekä laitteiston ominaisuuksia esittelevä ohjelmisto. Tarkkaa ja lopullista suunnitelmaa ei ollut siitä, mitä ohjauksia järjestelmällä tulaisiin tekemään, mutta selvää oli, että ideoita alkaisi syntyä samalla, kun tutustuu laitteistoon.

### 7.1 Käyttöliittymä

Totesin, että helpoimmaksi tavaksi lähteä rakentamaan, sekä hahmottamaan ohjelmaa ja sen toimintaa, oli luoda käyttöliittymän visuaalinen kokonaisuus visualisoinnin elementeillä aina siihen asti, mihin suunnitelman oli saanut valmiiksi. Käyttöliittymä on kuitenkin se rajapinta käyttäjän ja koneen välillä, joka määrittelee laitteen käytettävyyden ja soveltuvuuden. Mikäli on oletettavissa, että samanlaista käyttöliittymää samoilla ominaisuuksilla halutaan käyttää useissa eri toimilaitteissa, on syytä luoda referenssi-käyttöliittymä. Tällä tavoin käyttöliittymä saadaan toimimaan toimilohkon tavoin, johon loppujen lopuksi tarvitsee linkittää vain ohjattava toimilaitte, joka tässä tapauksessa on moottori. Ao. kuvan 31 mukainen käyttöliittymä on määritelty referenssiksi, joka on otettu käyttöön kummallekin moottorille liittämällä se frameen käyttöliittymässä Manual Control.



Kuva 31. Referenssi Manual Control (TwinCAT 3, 2019).

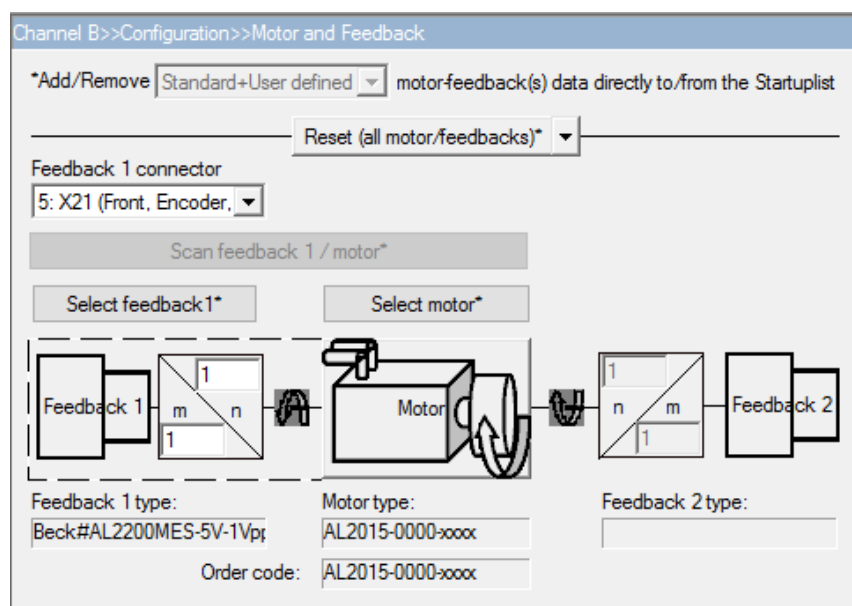
Oheisen lähteen mukaisesti luodaan rajapinta PLC-ohjelmassa olevien muuttujien ja visualisoinnin välille. Visualisointi linkittyy struktuureihin, jotka sisältävät projektissa tarvittavat muuttujat, jotka taas linkitetään niille määritettyihin visualisointiobjekteihin.



Kun referenssikäyttöliittymä on valmis, voidaan se määrittää omana "toimilohkonaan" visualisoinnin eri sivuille objektin Frame avulla, johon linkitetään referenssimuuttuja (moottori), joka sisältää käytettävänä kaikki sille struktuuriin määritetyt muuttujat. (Beckhoff, n.d.y)

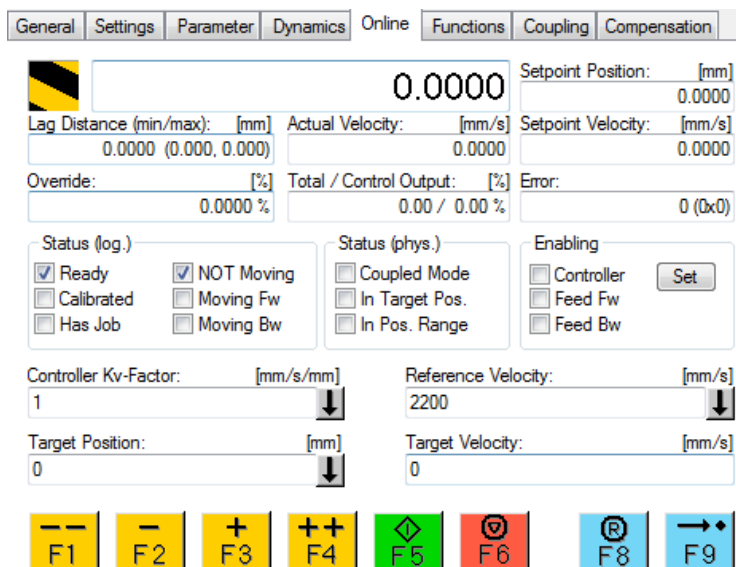
## 7.2 Moottoreiden käyttöönotto

Beckhoff-laitteistolla moottoreiden käyttöönotto helpoimmillaan käy muutamalla hiiren klikkauksella, tai sitten hieman useammalla. Tähän vaikuttaa tyypillisesti moottorin elektroninen tyyppikilpi, sekä käytettävä enkooderi. Kuvassa 32 on asetettu moottorin skannaus, sekä enkooderin valinta.



Kuva 32. Moottorin ja enkooderin valinta/skannaus (TwinCAT 3, 2019).

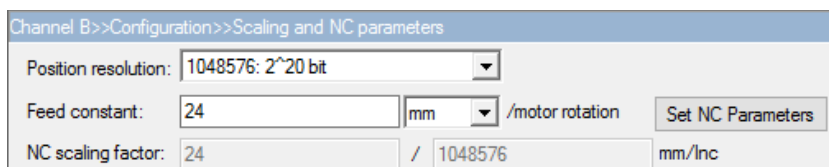
Normaalin servomoottorin AM8032, sekä sen kanssa toimivan absoluuttienkooderin EKM36 Hiperface protokollalla käyttöönotto onnistui hyvin helposti, eikä ns. ylimääräisiä parametreinteja tarvitse tehdä. Drive Manager skannailee moottorin, sekä enkooderin tiedot ja asettaa alustavat parametrit, jonka jälkeen moottoria voidaan ajaa kyseisen akselin online-näkymässä, kuva 33 havainnoillistaa online-näkymää.



Kuva 33. Online näkymä (TwinCAT 3, 2019).

Ennen moottorin ajamista on kuitenkin olosuhteet huomioiden syytä varmentaa moottorin Feed Constant –arvo, joka määrittää moottorin kuljettaman matkan yhtä rotaatiota kohden. (Beckhoff, System Manual Servo Drives AX5000, 2018.a, s. 118)

Lineariservomoottorin käyttöönotto poikkesi hieman yksityiskohtaisuudellaan normaalin servomoottorin käyttöönotosta; käyttöönotto aloitetaan, kuten tavallisella servomoottorilla, mutta moottorityyppi AL2015-0000, sekä takaisinkytkentänä toimiva semiabsoluuttinen enkooderi AL2200MES-5V-1Vpp on valittava manuaalista moottori- ja takaisinkytkentälistalta, kuten kuvassa 28. Lineaarimoottorin napaparietäisyys on 24mm, joka vastaa yhtä loogista kierrosta ja tämä on otettava huomioon skaalauskerroimessa. Tämä määrittelee Feed Constant –arvon, joka on oltava 24mm tällä moottorityypillä. Asetettu kuvassa 34. (Beckhoff, System Manual Servo Drives AX5000, 2018.a, s. 154), (Beckhoff, AL2xxx Linear Motor Manual, 2018, s. 19)



Kuva 34. Feed constant (TwinCAT 3, 2019).

Lineaarimoottori ei myöskään sisältänyt lämpötilasensoria, joka on ilmoitettava moottoriparametrissa P-0-061. (Beckhoff, AX5000 IDN-Description, 2017, s. 227)

Kommutointikulma oli haettava laitteelle manuaalisesti MES-takaisinkytkentäjärjestelmän takia. (Beckhoff, System Manual Servo Drives AX5000, 2018.a, s. 159)

Kun kumpikin moottori on otettu käyttöön onnistuneesti, on huomioitava parametri S-0-0112, joka määrittelee nimellisvirran ylärajan vahvistimelle, kun sillä syötetään virtaa kahteen kanavaan, eli ajetaan kahta moottoria. Tämä parametri on A ja B kanavan nimellisvirtojen summan yläraja, joka määritetään kanavakohtaisesti parametrilla P-0-0093. (Beckhoff, AX5000 IDN-Description, 2017, s. 100)

Tämän työn kohdalla toinen moottori on reilusti ylimitoitettu vahvistimen virransyöttöön nähden, mutta AX5000 vahvistimella on mahdollista "lainata" virtaa toisen moottorin kanavasta ja antaa se toisen moottorin käyttöön, kuten kappaleessa 3.4.2 on esitelty.

### 7.3 Kommutointikulman hakeminen absoluuttienkooderille

#### 7.3.1 Mekaaninen säätö

Kommutointikulman hakeminen Single Turn -tyyppisille absoluuttienkoodereille, jotka tukevat seuraavia takisyntykentäprotokollia:

- EnDat
- Biss
- Resolver
- HIPERFACE

Moottorin roottori, sekä enkooderi on valmistusvaiheessa säädetty tehtaalla mekaanisesti yhteensopiviksi, jolloin kyseistä takisyntykentäprotokollaa käyttävä järjestelmä voi valmistajan määrittämällä tekniikalla etsiä oikean kommutointikulman.

Kommutointikulma asetetaan Drive Managerista seuraavin parametrein:

1. Asetetaan kommutointikulman kalibrointi parametrilla P-0-0160 Set & enable cmd. Tämän asetuksen käynnistäminen voi aiheuttaa moottorissa liikettä.
2. Asetustavaksi määritetään parametri P-0-0165 Static current vector.
3. Kommutointi offset-arvo moottorin pyörimissuunnan havaitsemiseksi P-0-057 270 astetta. Tämä arvo riippuu moottorin mekaanisesta ja elektronisesta rakenteesta. Beckhoff-moottoreille se on 270 astetta.
4. P-0-058 lukee moottorin digitaalisesta tuotekilvestä vaadittavan kommutoinnin offset-arvon ja tallentaa sen parametriin P-0-150 Adjustable commutation offset (moottorin tietokanta).
5. Jotta arvoa P-0-150 Adjustable commutation offset voidaan käyttää, on P-0-150 Commutation mode-arvoksi määriteltävä 3: Adjustable mechanical offset.

### 7.3.2 Elektroninen säätö

Elektroninen kommutointikulman säätäminen voidaan toteuttaa kahdella tavalla. Ensimmäisellä tavalla, jossa moottorin roottori, sekä enkooderi on valmistusvaiheessa säädetty yhteneväiseen asentoon, jolloin kyseistä takaisinkytkentäprokollaa käyttävä järjestelmä voi valmistajan määrittämällä tekniikalla etsiä oikean kommutointikulman samalla tavalla, kuin kohdassa 7.3.1. Ainoa eroavaisuus on, että parametrin P-0-058 arvo tallennetaan myös enkooderin tietojärjestelmään ja että tämä tapa vaatii enkooderijärjestelmältä tiedonsiirto- ja tallennustoiminnon. Toinen tapa koskee laitteistoa, jossa roottorin ja enkooderijärjestelmän välinen kulma on päätetty valmistajan toimesta käyttäen komentoa, joka on spesifinen enkooderille ja joka kommunikoi enkooderijärjestelmän kanssa. Enkooderijärjestelmä tallentaa tämän arvon ja käyttää sitä sisäisessä laskennassa. Kommutointikulma haetaan yllämainitulla tavalla, mutta ainoa eroavaisuus on, että parametrin P-0-0150 Adjustable offset –arvo käytetään aina sisäisessä laskentaprosessissa ja tämä tapa vaatii älykkään enkooderijärjestelmän.

### 7.4 Kommutointikulman hakeminen ei-absoluuttisella enkooderilla

Kommutointikulman hakeminen Single Turn –tyyppisille ei-absoluuttienkoodereille, jotka tukevat seuraavia takisinkytkentäprotokollia:

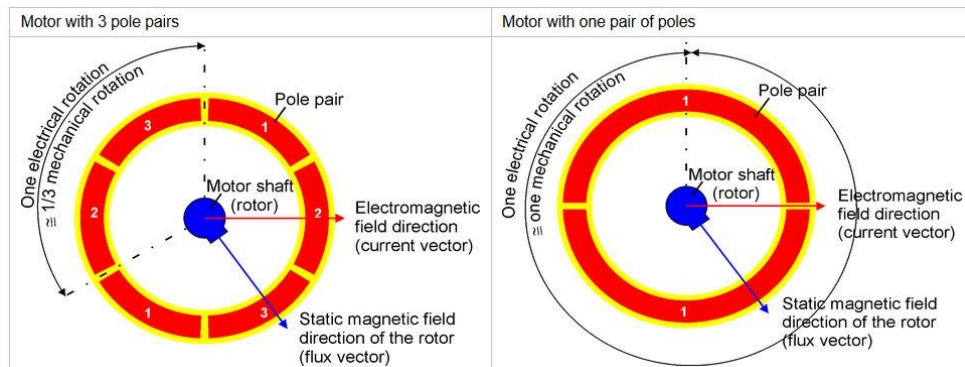
- SIN / COS 1V<sub>ss</sub>, 4.2
- TTL, 4.1

Tässä tapauksessa kommutointikulman hakeminen toteutetaan toteuttamaan Wake&Shake –toiminnolla. Tieto tallentuu sisäiseen järjestelmään. Mikäli servovahvistin AX5000 kytketään pois päältä, tai Ethercat-väylän tila muutetaan tilaan Pre-op, tai alemmaksi, kommutointikulma kadotetaan, koska enkooderijärjestelmä ei ole absoluuttinen. Jotta Wake&Shake –toiminto voidaan toteuttaa virheettömästi vain moottorijärjestelmän ollessa hallitusti paikoillaan niin, ettei siihen kohdistu ulkopuolista resonanssia.

Wake&Shake –toiminto koostuu kahdesta vaiheesta, jotka kuvaillaan periaatteellisella tasolla tässä luvussa. Tarkempi ohjeistus ja parametrisointi on löydettävissä valmistajan sivuilta.

Vaihe 1 koostuu likiarvoisesta roottorin asennon määrittämisestä, kun taas vaihe 2 määrittelee asennon tarkemmin. Toiminnon tarkoituksena on löytää oikea kommutointikulma minimaalisimmalla liikkeellä. Napaparien mukaan, servomoottorien sähköisellä ja mekaanisella kierroksella on suhde; yksi sähköinen kierros vastaa yhtä mekaanista kierrosta jaettuna napaparien määrällä.

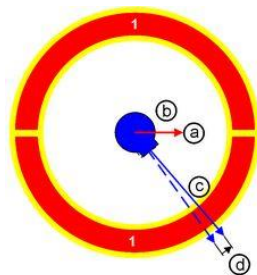
Kuva 35 havainnoillistaa sähköisen kierroksen suhdetta mekaaniseen kierrokseen, sekä virta- ja vuovektorin suuntaa sähkömoottorissa.



Kuva 35. Sähköinen ja mekaaninen kierros (Beckhoff, n.d.j)

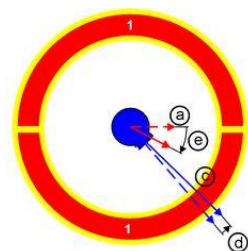
#### 7.4.1 Vaihe 1

Kuva 36 esittää ensimmäisen askeleen, jossa muodostetaan virtavektori (a) ajanhetkellä (b), jonka vaikutuksesta muodostuva magneettinen voima kääntää roottoria (c) virtavektorin suuntaan (a). Pyörimissuunta (d) lähetetään takaisinkytkentäjärjestelmälle ja servovahvistimelle.



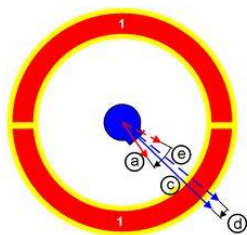
Kuva 36. Step 1 (Beckhoff, n.d.j)

Kuva 37 esittää toisen askeleen, jossa muodostetaan virtavektori (a) ajanhetkellä (b), jonka vaikutuksesta muodostuva magneettinen voima kääntää roottoria (c) virtavektorin suuntaan (a). Pyörimissuunta (d) lähetetään takaisinkytkentäjärjestelmälle ja servovahvistimelle, missä se analysoidaan. Jos analyysi näyttää, että roottorin (c) pyörimissuunta (d) ei ole muuttunut edelliseen vaikuttuneeseen virtaan verrattuna, prosessi jatkuu.



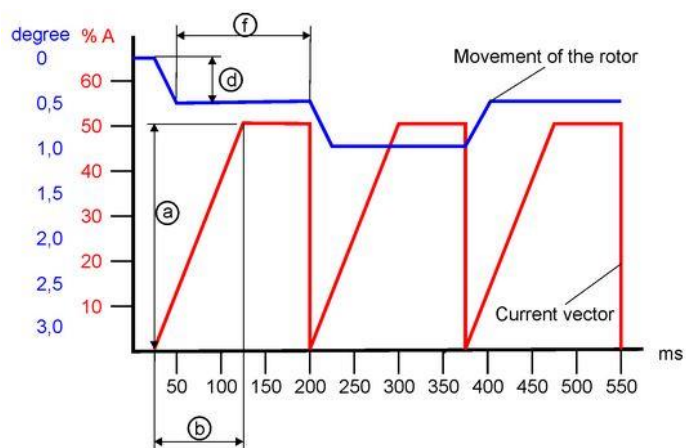
Kuva 37. Step 2 (Beckhoff, n.d.j)

Kuva 38 esittää kolmannen askeleen, jossa muodostetaan virtavektori (a) magnitudilla (e) ajanhetkellä (b), jonka vaikutuksesta muodostuva magneettinen voima kääntää roottoria (c) virtavektorin suuntaan (a). Pyörimissuunta (d) lähetetään takaisinkytkentäjärjestelmälle ja servovahvistimelle, missä se analysoidaan. Tässä tapauksessa analyysi näyttää, että roottorin (c) suunta (d) on muuttunut suhteessa edelliseen vaikuttuneeseen virtaan. Tulosten perusteella sektori, jossa roottori (c) sijaitsee on löydetty ja vaihe 1 on suoritettu.



Kuva 38. Step 3 (Beckhoff, n.d.)

Kuva 39 havainnoillistaa oskilloskooppinäkymän kommutointikulman hakemisesta vaiheessa 1. Virtavektorin arvo (a) kasvaa ajassa (b), jolloin roottori tekee käännöksen. Virtavektorin (a) arvo pidetään samassa ja annetaan oskilloimisen tasottua ajan (f), jolloin roottorin kulma luetaan. Roottorin kulman muutoksen havaitsemiseksi, on roottorin käännettävä minimissään arvon (d) verran.

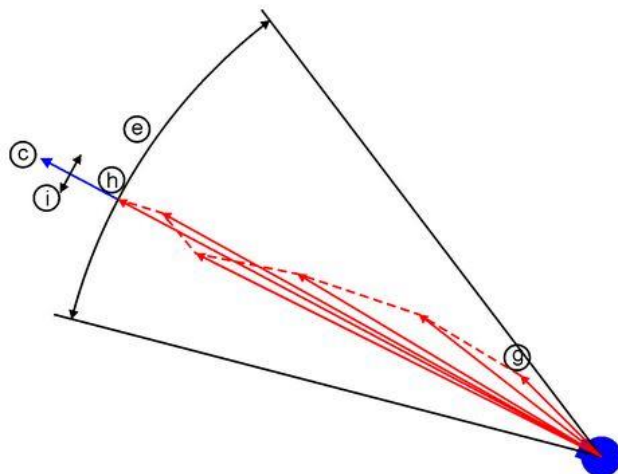


Kuva 39. Commutation Phase 1 Oscilloscope (Beckhoff, n.d.)

## 7.4.2 Vaihe 2

Tämä luku käsittelee kahta tapaa, joista toinen voidaan suorittaa vaiheessa 2 vaiheen 1 jälkeen. Tapa 1 vaikuttaa roottoriin vain minimalistisen liikkeen verran, joka vaatii järjestelmältä korkeaa stabiilisuutta oskilloimisen välttämiseksi. Tapa 2 vaikuttaa roottoriin siten, että se voi liikkua maksimissaan puolen sektorin (e) verran. Tapa 2 on sietävämpi oskilloimisen kannalta. Säättöä havainnoillistava kuva 40.

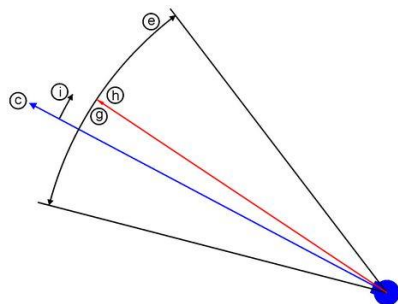
1.



Kuva 40. Kommutointikulma:  $K_p$  (Paikkavahvistus)  $> 0$  (Beckhoff, n.d.j)

Virtavektori (g) luodaan alkavaksi virtavektorin (a) viimeisestä paikasta vaiheessa 1. Magneettinen voima kääntää roottoria virtavektorin (g) suuntaan. Liikkeen muutos lähetetään takaisinkytkentäjärjestelmään servovahvistimelle ja siten suljettuun ohjauspiiriin. Ohjauspiiri korjaa välittömästi virtavektorin suunnan. Tämä ohjauspiirin algoritmi suoritetaan niin monta kertaa, kunnes virran intensiteetti saavutetaan ja virtavektori lähestulkoon vastaa vuovektoria. Virta säilytetään ajan (h), jotta optimaalinen kommutointi on suoritettavissa. Tämä algoritmi liikuttaa roottoria vain minimalisesti verran (i). Säättöä havainnoillistava kuva 41.

2.



Kuva 41. Kommutointikulma:  $K_p$  (Paikkavahvistus)  $= 0$  (Beckhoff, n.d.j)

Kun sektori (e) on määritetty vaiheessa 1, virtavektori (g) määritetään tarkalleen sektorin (e) keskelle. Magneettinen voima kääntää roottoria (c) virtavektorin suuntaan (g), kunnes ne kohtaavat. Tässä staattisessa järjestelyssä, roottori ei voi liikkua, kuin puolet sektorista (e). (Beckhoff, System Manual Servo Drives AX5000, 2018.a, s. 204)

## 7.5 Henkilöturvallisuus

Henkilöturvallisuuden kannalta oli välttämätöntä ottaa käyttöön valoverhot demolaitteen ympärille. Ympäristön yksinkertaisuuden vuoksi valoverhoille ei ollut tarvetta tehdä sädekoodausta, joka yksilöi lähettimen ja vastaanottimen toimimaan yksilöllisellä signaalilla, jotta voidaan erottaa ympäristöstä muiden valoverholähtettimien signaali. (SICK, SICK C/M 2000 Manual, n.d.db, s. 201)

Valoverhojen kytkentä, kohdistus ja laitteiston toimivuus suoritettiin manuaalin kappale 6 mukaisesti. (SICK, SICK C/M 2000 Manual, n.d.db, s. 223)

Valoverhojen asennuksen jälkeen oli vuorossa turvaohjelman luominen laitteilla SICK2000, EI1904, EL6900 ja AX5805. Hardware-taso haetaan väyläkonfiguraatioon samalla tavalla, kuten muutkin EtherCAT-termiinaalit. (Beckhoff, EL20xx Manual, n.d.x, s. 123)

Turvatermiinaalit ovat EtherCAT-väylässä, kuten mitkä tahansa kenttäväyläkomponentit, ja tätä kautta niiden CoE-online -tiedot ovat käytettävissä. (Beckhoff, Safety Editor - Hardware Level, n.d.å)

Jotta rauta- ja ohjelmistotasot saadaan yhdistettyä toiminnalliseksi kokonaisuudeksi, täytyy ensiksi luoda turvaprojekti ja ns. aliastraso, jossa yhdistäminen tapahtuu. Aliastrasossa määritellään TwinSAFE-ryhmän aliaslaitteet, joihin linkitetään hardware-tasolta vastaava kenttäväylätermiinaali. (Beckhoff, TwinSAFE PLC, 2017.ä)

Kun käytetään AX5000 vahvistimen omaa turvakorttia, on myös Drive Managerista määritettävä kyseinen turvakortti käyttöön parametrilla P-0-2000, mutta myös aliaslaitteiden kyseisen turvakortin turvaparametreihin on määritettävä vahvistimen käytettävien kanavien määrä, sekä turvareaktio. (Beckhoff, AX5805 Manual, 2017.ö, s. 90)

Kun TwinSAFE rauta- ja aliastraso on yhdistetty, on nämä yhdistettävä ohjelmistotasoon. Ohjelmistotasossa ohjelmoidaan turvaprojektin toiminnallinen logiikka graafisella editorilla, johon syötetään aliastrasoon määritellyt laitteet, sekä toiminnot. Aliastrasoon määritetyt laitteet toimivat turvatoimilohkojen tuloina ja lähtöinä. Kun turvaohjelma on rakennettu, suoritetaan variable mapping, jolla liitetään haluttuja turvatoiminnallisuuksia toimilohkoon halutun toiminnon toteuttamiseksi,



joka tässä tapauksessa valoverhojen käynnistämä turvareakto on STO, joka poistaa vääntömomentin akselilta. (Beckhoff, n.d.ba)

Turvaprojektin valmistuttua, sekä valoverhojen kytkettyä EL1904 tuloihin, on huomioitava valoverhon lähettämä kontaktorivalvonnan signaali. Mikä tulokortti E1904 ei tiedä, että kyseinen signaalin ”pätkiminen” on kontaktorivalvonnan testausignaali, katkaisee turvaohjelma laitteiston toiminnan ja asettaa sen turvalliseen tilaan. Jotta kyseinen kontaktorivalvonta ei aiheuta turvapiirin laukeamista, on aliaslaitteen EL1904 turvaparametreista määriteltävä valoverhojen kanavien operointitavaksi Any pulse repetition OSSD, sensor test deactivated(2). (Beckhoff, EL1904 Manual, 2017.x, s. 36)

## 7.6 Lähdekoodin luominen

Ennen ohjelmoimisen aloittamista, oli syytä selvittää, millä tavalla ohjelma luodaan visuaalisesti selkeäksi ymmärtää, jotta ohjelman lukeminen olisi helppoa ja että kolmas osapuoli saisi selvää siitä, mikäli on tarpeen tutkia, tai muokata koodia. Toisekseen ohjelman rakenteellinen puoli on oltava myös järkevästi tehty, jotta se olisi tulevaisuuden kestävä ja että mahdollisten lisäosien luominen valmiiseen ohjelmaan onnistuu vaivattomasti. Kuva 42 havainnoillistaa muuttujien kommentointia ja jäsentämistä.

```

1  {attribute 'qualified_only'}
2  VAR_GLOBAL
3
4      //Commands to TwinSAFE controllers safety group
5
6      bGroupRun          AT %Q*      : BOOL := TRUE;
7      bErrorAck          AT %Q*      : BOOL;
8
9      // Commands to AX5805 card
10
11     bSTO1              AT %Q*      : BOOL;
12     bAX5805_Restart    AT %Q*      : BOOL;
13
14     //Feedback from TwinCAT controllers safety group
15
16     bFBError           AT %I*      : BOOL;
17     bComError          AT %I*      : BOOL;
18     bEStopOut          AT %I*      : BOOL;
19
20 END_VAR

```

Kuva 42. Global Variable, kommentointi (TwinCAT 3, 2019)

Lähdekoodin tekstin asettelu ja sisentäminen on ohjelmoinnin kannalta peruseikka, mutta aloitteleva koodari ei välttämättä ole kuullutkaan sellaisesta, vaan koodia kirjoitetaan yhdelle riville tarpeettoman pitkiä pätkiä koodin määritelmän vaihtuessa. Ohjelmointikieleksi valitsin struktuuritekstin (Structured Text), koska se on mielestäni helpointa luettavaa hyvän kommentoinnin kanssa, sekä nopeaa kirjoittaa. Ladder Diagram, tai Function Block Diagram lähdekoodina voi vaikuttaa ensisilmäyksellä ohjelmaa luettaessa helpoimmalta ja selkeimmältä

vaihtoehdolta, mutta koodin paisuessa siitä tulee hyvin vaikeaselkoista. Struktuuriteksti on kuin kirjoittaisi englanniksi lauseita, sekä sen asettelu tuntuu moninverroin vapaammalta graafiseen ohjelmointiin verraten.

Ohjelmointitapoja on niin monia, kuin on tekijöitäkin, mutta yritin toimia yleisten mallien mukaisesti ja noudattaa ohjeistuksia PLC-koodin luomiseen. Lähdekoodia luodessa kiinnitin huomiota tekstin sistentämiseen, sekä kommentoimiseen, jotta ohjelmaa muuttaessa, tai lukiessa tietää, mitä koodissa tapahtuu. Havainnoillistava kuva 43. Tähän on onneksi paljon asetteluohjeita netissä. (PLCAcademy, 2015)

```

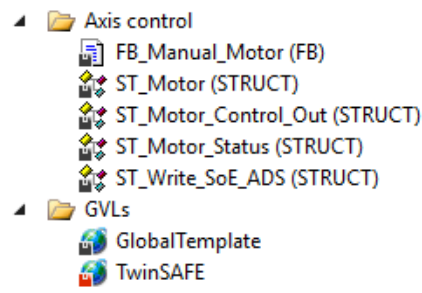
1 //Setting torque mode ON
2
3 CASE nTorque OP
4 //Bit to set torque mode
5 0:
6 IF i_bTorqueMode THEN
7 nTorque := 10;
8 END_IF
9 //Driving to reference, to ensure absolute position
10 10:
11 GlobalTemplate.MotorControl_Linear.stControl.bDriveToReference := TRUE;
12 GlobalTemplate.MotorControl_Servo.stControl.bDriveToReference := TRUE;
13 nTorque := 20;
14
15 20:
16 GlobalTemplate.MotorControl_Linear.stControl.bDriveToReference := FALSE;
17 GlobalTemplate.MotorControl_Servo.stControl.bDriveToReference := FALSE;
18 //If reference run succeeded, then set reference bits ON
19 IF GlobalTemplate.MotorControl_Linear.stStatus.i_bReferenced THEN
20 o_bReferencedLinear := TRUE;
21 END_IF
22 IF GlobalTemplate.MotorControl_Servo.stStatus.i_bReferenced THEN
23 o_bReferencedServo := TRUE;
24 END_IF
25 IF o_bReferencedLinear AND o_bReferencedServo THEN
26 nTorque := 30;
27 END_IF
28

```

Kuva 43. Sisennys ja kommentointi. (TwinCAT 3, 2019).

Ohjelmointiympäristössä on syytä kiinnittää huomiota myös sen rakenteeseen, kun käytössä on ohjelmia, toimilohkoja, funktioita, globaaleja muuttujalistoja ja objekteja. Ohjelmarakenne tulee pitää selkeänä, joten eri toimintojen jakaminen omiin osiinsa selkeyttää ohjelman käsittelyä. Muuttujalistat määritetään tietynlaisille muuttujille erikseen, esim. statuksen, tai ohjauksen määrittävät muuttujat. Mikäli suoritetaan useita funktioita, tai määritetään monen toimilohkon tekeviä sekvenssejä yhtenä kokonaisuutena, on nämä syytä määrittää omaan ohjelmaan, tai tarpeen mukaan hajauttaa toisistaan, mikäli halutaan suorittaa funktiota tietyssä osissa.

Tässä työssä muuttujien hakemisen helpottamiseksi ja selkeyttämiseksi, loin lähes kaikki muuttujat struktien alle jakaen ne toimintojensa mukaan, jolloin vain globaalissa muuttujalistassa määritellään moottorin tyyppi haluttu strukti, jonka alta voidaan noutaa jokainen moottorikohtainen muuttuja, kuten kuvassa 40 on määritetty. Tällä tyyllillä muuttujat luodaan vain kerran, jolloin niitä voidaan käyttää kullekin moottorille niiden yksilöllisinä muuttujinaan. (Beckhoff, n.d.bb)



Kuva 44. Jakaminen toiminnallisuuden mukaan (TwinCAT 3, 2019).

## 7.7 Demo-/testiohjelmat

Työn keskeisiin osa-alueisiin kuului demottavat ohjelmat, joita voi ajaa käyttöliittymän kautta. Ohjelmien tarkoituksena on havainnollistaa moottorien toimintaa lineaariradalla, sekä mahdollistaa niiden suorituskyvyn testaamista ajaen vapaasti moottoria yksilönä, tai toinen moottori toimiessa kuormana toiselle.

### 7.7.1 Käsiäjo ja paikoitus

Käsiäjo (MC\_Jog) ja paikka-ajo (MC\_MoveAbsolute) määriteltiin ohjelmista ensimmäisinä ohjattavaksi käyttöliittymän kautta. Näiden avulla on mahdollista testata liikeohjausjärjestelmän dynamiikkaa ja nähdä, miten moottorit radalla toimivat. Käsiäjojen olennaisena kokeilumenetelmänä, myös moottorin dynamiikan ( $V, a, jerk$ ) muuttaminen paneelista näyttelee merkittävää roolia kokeilijalle, jotta ymmärtää arvojen muutoksien vaikutuksen käytännössä.

**Liite 2/1, 2/2, 2/3, 2/4** havainnollistavat käsiäjon lähdekoodia.

### 7.7.2 Sekvenssi

Sekvenssiohjelma luotiin laitteistolle, jotta voidaan mahdollistaa kummankin moottorin itsenäinen ajaminen omilla nopeuksillaan, sekä omilla paikkatiedoillaan.

Sekvenssi on myös ohjelmapohjanaan osa kuormitusohjelman rakennetta, jossa se näyttelee toisen moottorin työskentelydynamiikkaa, sekä sen toimintaa voidaan käyttää hyödyksi tulevaisuuden muutoksia varten lineaaridemossa. Jatkokehityksenä sen tarkoituksena on olla osa törmäystarkastelukirjaston (TF5410) demonstroinnissa, jolloin sen avulla voidaan luoda kiinteä sekvenssi toiselle moottorille, joka on kuitenkin väistämivelvollinen toista moottoria kohden.

**Liite4/1** Havainnollistaa Sekvenssiohjelman lähdekoodia.

### 7.7.3 Kuormitusohjelma

Kuormitusohjelman tarkoituksena asettaa moottorit tilaan, jossa normaali servomoori on kuormitettavana, ja lineaariservomoottori toimii kuormittavana moottorina.

Kyseisen ohjelman tavoitteena on mahdollistaa moottorin kuormituksen seurantaa; virta- ja voima-arvot, lämpötila, stabiilisuus. Sen avulla voidaan myös seurata moottorin toiminnan stabiilisuutta, muuttamalla kuorman määrää, tai säätämällä moottorin ohjauspiiriä. Ohjelman toiminnallinen järjestys kuvattu alla:

1. Kumpikin moottori suorittaa referenssiajon, jotta voidaan varmistaa moottoreiden oikea sijainti.
2. Moottorien kelkkojen kyljet ajetaan hitaasti samaan pisteeseen koordinaatistossa niin, että ne lähes koskevat toisiaan.
3. Vastustava moottori aloittaa paikka-ajon jonka määritelty sijainti on kuormitettavan moottorin takana, jolloin kuormitettava moottori aloittaa sekvenssiajon ja liikkuu toistuvasti pisteiden A ja B välillä.

Kuormittavan moottorin vastustava voima  $F$ , kuormitettavan moottorin nopeus  $v$ , kiihtyvyys ja hidastuvuus  $m/s^2$ , sekä kiihtyvyyden muutosnopeus  $m/s^3$  ovat käyttäjän säädettävissä.

Ohjelman parametroinnissa on huomioitava paikkamonitorointi (IDN 159), sekä TwinCAT NC:n position lag monitoring. Nämä asetukset mahdollistavat moottorin toiminnan silloinkin, kun moottori ei ajokäskystä huolimatta liiku eteenpäin. Huomioitavaa on myös se, kun paikkaviivettä kertyy moottorille, pyrkii moottori jatkuvasti saavuttamaan paikan koordinaatistossa, jossa paikkaviive on 0.

**Liite 3/1, 3/2** havainnoillistavat kuormitusohjelman lähdekoodia.

## 8 YHTEENVETO JA POHDINTA

Työtä aloittaessa demolaitteisto oli tekniikan puolesta päivitetty uudelle tasolla, mutta kaikki järjestelmään liittyvät asetukset ja ohjelmat puuttuivat. Tavoitteiden kannalta merkittävintä on, että laitteiston käyttöönotto onnistui kaiken kaikkiaan hyvin ja demolaitteisto on kokonaisuudessaan käyttökelpoinen ja sitä voi turvallisesti käyttää ketä tahansa. Demolaitteistosta tuli helpommin lähestyttävä järjestelmä, jota voidaan jatkossa käyttää laitteistoesittelyyn, koulutukseen, sekä testaukseen erilaisten tuotevalikoimien kanssa.

Työn laajuus alkoi pieniltä osin osoittautua ongelmaksi aikataulullisista syistä, kun jokaisesta aihealueesta olisi mielellään raapaissut pintaa syvemmältä, jotta aiheita ensikertaa käsittelevä henkilö saisi asiasta helpommin ymmärryksen. Toisaalta voisi ajatella, että mielenkiintoinen aihe nappaa helposti mukaansa. Työn monipuolisuus ja mielenkiintoisuus toi kokemusta ja uusia asioita hyvin paljon.

Lähtiessäni työtä tekemään, kuvittelin järjestelmän toiminnan saattamisen toiminnalliseksi kokonaisuudeksi tapahtuvan hieman yksinkertaisemmin. Demolaitteen eri osa-alueet yksittäisinään ovatkin melko selkeitä, mutta yhteen sulautettuna toimivana kokonaisuutena käyttöönotto ilmenikin monimutkaisemmaksi. Laitteistokokonaisuuden ohjelmallinen saattaminen toimintakuntoiseksi ja turvallisiksi on yksityiskohtaisempaa, kuin voisi ensiksi kuvitella. Yksittäisenä osanaan turvalaitteisto, moottorin liikuttaminen, tai käyttöliittymän luominen ei ole isokaan asia, mutta näiden toiminnan yhteen liittäminen järkevällä toiminnalla vaatii tuotetuntemusta, sekä kehitysympäristön työkalujen käyttöön liittyvää osaamista puhumattakaan PLC-ohjelmoinnin osaamisesta.

Näin jälkikäteen ajateltuna työn etenemistä ja päätökseen saattamista olisi helpottanut, jos olisin hyväksynyt tavoitteelliset toiminnot saavutetuiksi jo aikaisemmassa vaiheessa. Ohjelmoinnin kannalta ohjelma voisi aina olla hieman älykkäämpi, tai diagnostiikka laajempaa. Käyttöliittymän toiminnallisuus, sekä ulkoasu ovat hyvin paljon mielipiteiden varassa, joihin ei olisi kannattanut lähteä liiaksi etsimään ns. "oikeaa" tapaa. Hyvin paljon tuli käytettyä aikaa monien keskeisten asioiden selvittämiseen, jotka ovat välttämättömiä ymmärtää liikkeenohjausjärjestelmää rakentaessa, mutta jotka eivät kuitenkaan näy raportissa rajauksien takia.

### 8.1 Tavoitteiden toteutuneisuus

Ensimmäinen tavoite, jonka päämääränä oli ohjelmoida liikkeenohjaussovellukset, joiden avulla on mahdollista ajaa moottoreita käsiajolla, paikoittamisella, sekvenssitoiminnolla, sekä ohjelma, jonka avulla voidaan simuloida kuormaa toiselle moottorille käyttäen toista moottoria vastuksena saavutettiin, mutta ongelmiakin ilmeni.

Kuten kappaleessa 7.6 on mainittu, että ohjelman suunnittelu olisi syytä tehdä huolella niin, että sen rakenne on kunnossa ja sitä on helppo ymmärtää. Tässä kohtaan menin alussa metsään vaikka olin koittanut pitää selvästi ohjelmistoarkkitehtuurista kiinni. Seurauksena tästä aiheutui ohjelmavirheitä (bugeja), joita voi olla hyvinkin vaikea selvittää, mistä väärä laitteiden toiminnallisuus johtuu. Oma ohjelmani on hyvin pieni verrattuna isoihin projekteihin, jossa selkeä arkkitehtuuri näyttelee merkittävää osaa. Pienikin bugi jossain, voi odottamattomassa tilanteessa tulla esille ja virheen aiheuttajaa ei helposti löydä. Loppujen lopuksi ohjelmat käsiajo, sekvenssi ja kuormitusohjelma saatiin onnistuneesti kasaan ilman toimintavirheitä.

Toisena tavoitteenani oli luoda käyttöliittymä rajapinnaksi käyttäjän ja ohjelman välille, josta sovellusta voidaan ohjata, sekä seurata diagnostiikkaa. Käyttöliittymän luominen oli mielestäni helpoin osuus työstä, jonka luominen onnistuikin vaivattomasti. Jos jotain olisin voinut käyttöliittymää luodessa tehdä järkevämmiin, niin jälkikäteen ajateltuna, olisi pitänyt vain rohkeasti luoda halutunlainen käyttöliittymä ajattelematta sen enempää muun ohjelman etenemistä. Käyttöliittymä on kuitenkin visuaalinen kuva laitteiston toiminnasta, joten hyvän käyttöliittymän rakentamisen jälkeen, jäljelle jää enää siihen liitettävä ohjelmisto ja kuitenkin valmis käyttöliittymä toimii hyvänä muistiona ohjelman toiminnallisuudesta.

Kolmantena tavoitteena oli asettaa turvarajat laitteen ympärille, sekä turvaohjelma suojaamaan henkilövahingoilta laitteiston kanssa. Turvarajojen asentaminen ja kytkeminen saatiin onnistuneesti myös päätökseen. Ainoat ongelmat turvarajojen asetusten kanssa syntyivät valoverhojen OSSD-signaalin kanssa, kuvattu kappaleessa 6.2.1. Signaalia vastaanottava laite EL1904:n asetuksiin oli määriteltävä Any Pulse Repitition OSSD, jotta kyseisen valoverhon OSSD-signaali hyväksytään turva-IO:ssa.

Henkilö- ja koneturvallisuuden kannalta olisi turvallisinta luoda turvarajat laitteen ympärille henkilösuojaksi, kuten myös moottoreiden suojaksi asettamalla moottoreille maksimidynamiikka siten, että moottorit eivät pysty ajamaan itseään yli päätyrajoista stoppareita päin. Kuten kappaleessa 3.5 on esitetty turvakorttia AX5805 koskevia turvafunktioita, voi tämän kortin turvafunktioilla määrittää maksimidynamiikan monitorointeja suojaamaan kyseisiltä vaaroilta.

## 8.2 Jatkokehitys

Laitteistoiesittelyn ohjelmallinen toiminnallisuus on käytännössä universaali, jonka takia demolaitteen toiminnan muuttaminen tulevaisuudessa käy yksinkertaisesti. Käytännössä laitteiston demoamiseen asettuvat rajat kohdistuvat laitteiston mekaniikkaan, mutta PLC-koodia

päivittämällä, sekä parametroimalla moottorijärjestelmää on mahdollista luoda monipuolisia ohjelmia ja simuloiteja.

Ensimmäinen jatkokehitystoimenpide laitteistolle tulee olemaan, TF5410 TC3 Motion Collision Avoidance –kirjaston käyttöönotto, jonka ominaisuuksia simuloidaan demolaitteistolla. TF5410:n ominaisuuksien esittely on kannattava jatkohanke sen myyntiesittelyn kannalta. Työhön luotu PLC-ohjelma käyttöliittymineen on mahdollista ottaa käyttöön myös muissa TwinCAT-kehitysympäristön omaavissa kontrollereissa, joissa on käytössä servojärjestelmän komponentit liitettävänä TwinCAT ohjelmointiympäristöön.

## LÄHTEET

12100, ISO. (n.d). ISO 12100 Standardi. Haettu 28. 4 2019

Anaheim Automation. (n.d). *Servo Motor Guide*. Haettu 28. 4 2019 osoitteesta AnaheimAutomation:  
<https://www.anaheimautomation.com/manuals/forms/servo-motor-guide--old.php>

Beckhoff. (2017). AX5000 IDN-Description. Teoksessa Beckhoff, *AX5000 IDN-Description* (s. 227). Haettu 28. 4 2019

Beckhoff. (2017.x). *EL1904*. Haettu 28. 4 2019 osoitteesta Beckhoff:  
<https://www.beckhoff.fi/el1904/>

Beckhoff. (2017.x). *EL1904 Manual*. Haettu 28. 4 2019 osoitteesta  
<https://download.beckhoff.com/download/document/automation/twinsafe/el1904en.pdf>

Beckhoff. (2017.x). *EL1904 Manual*. Haettu 28. 4 2019 osoitteesta  
<https://download.beckhoff.com/download/document/automation/twinsafe/el1904en.pdf>

Beckhoff. (2017.ä). TwinSAFE PLC. Teoksessa Beckhoff, *TwinSAFE PLC*. Haettu 28. 4 2019

Beckhoff. (2017.ö). *AX5805 Manual*. Haettu 26. 4 2019 osoitteesta  
[https://download.beckhoff.com/download/document/automation/twinsafe/ax5805\\_ax5806en.pdf](https://download.beckhoff.com/download/document/automation/twinsafe/ax5805_ax5806en.pdf)

Beckhoff. (2018). *AL2xxx Linear Motor Manual*. Haettu 28. 4 2019

Beckhoff. (2018.a). *System Manual Servo Drives AX5000*. Beckhoff. Haettu 26. 4 2019 osoitteesta  
[https://download.beckhoff.com/download/document/motion/ax5000\\_system\\_manual\\_hw2\\_en.pdf](https://download.beckhoff.com/download/document/motion/ax5000_system_manual_hw2_en.pdf)

Beckhoff. (n.d.a). *Beckhoff*. Haettu 26. 4 2019 osoitteesta Beckhoff: <https://beckhoff.fi/>

Beckhoff. (n.d.b). *Open CNC architecture with TwinCAT 3*. Haettu 26. 4 2019 osoitteesta Beckhoff:  
[https://beckhoff.es/english.asp?beckhoff/trade\\_show\\_emomilano.htm](https://beckhoff.es/english.asp?beckhoff/trade_show_emomilano.htm)

Beckhoff. (n.d.ba). *Safety Editor - Application Level*. Haettu 28. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_safety\\_editor/html/tc3\\_safetyeditor\\_040\\_aliaslayer.htm&id=16933](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_safety_editor/html/tc3_safetyeditor_040_aliaslayer.htm&id=16933)



- Beckhoff. (n.d.bb). *TwinCAT PLC Control, Objects*. Haettu 28. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/tcplctrl\\_componentsobjects.htm&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/tcplctrl_componentsobjects.htm&id=)
- Beckhoff. (n.d.bc). *AX5000 Tuning Guide*. Haettu 28. 4 2019
- Beckhoff. (n.d.c). *Current Controller*. Haettu 2019. 4 2019 osoitteesta Beckhoff Infosys:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/ax5000\\_function\\_doku\\_hw2/9007202500532363.html&id=6977360479130135846](https://infosys.beckhoff.com/english.php?content=../content/1033/ax5000_function_doku_hw2/9007202500532363.html&id=6977360479130135846)
- Beckhoff. (n.d.d). *Magnetic Encoder System*. Haettu 26. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/content/1033/al2200\\_mes\\_ba/9007200867122827.html?id=39619506597336762](https://infosys.beckhoff.com/content/1033/al2200_mes_ba/9007200867122827.html?id=39619506597336762)
- Beckhoff. (n.d.e). *Magnetism*. Haettu 26. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/content/1033/al\\_grundlagen/2036345995.html?id=715220788344263710](https://infosys.beckhoff.com/content/1033/al_grundlagen/2036345995.html?id=715220788344263710)
- Beckhoff. (n.d.f). *Phase Linear Motor*. Haettu 26. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/content/1033/al\\_grundlagen/2036402699.html?id=4853161992891955706](https://infosys.beckhoff.com/content/1033/al_grundlagen/2036402699.html?id=4853161992891955706)
- Beckhoff. (n.d.g). *Primear of linear motors*. Haettu 26. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/al\\_grundlagen/2036465803.html&id=2361486327634417703](https://infosys.beckhoff.com/english.php?content=../content/1033/al_grundlagen/2036465803.html&id=2361486327634417703)
- Beckhoff. (n.d.h). *Primear of linear motors*. Haettu 26. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/al\\_grundlagen/2036471947.html&id=8303018993228353612](https://infosys.beckhoff.com/english.php?content=../content/1033/al_grundlagen/2036471947.html&id=8303018993228353612)
- Beckhoff. (n.d.i). *Linear and Rotary Compared*. Haettu 26. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/content/1033/al\\_grundlagen/2036411275.html?id=8968105296317619326](https://infosys.beckhoff.com/content/1033/al_grundlagen/2036411275.html?id=8968105296317619326)
- Beckhoff. (n.d.j). *Commutation Methods*. Haettu 4. 26 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/content/1033/ax5000\\_usermanual/html/ax5000\\_kommutierungsverfahren.htm?id=8350892642470419167](https://infosys.beckhoff.com/content/1033/ax5000_usermanual/html/ax5000_kommutierungsverfahren.htm?id=8350892642470419167)
- Beckhoff. (n.d.k). *AX5000 Operating Modes*. Haettu 26. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/content/1033/ax5000\\_usermanual/html/ax5000\\_operationmodes.htm?id=8491271635880908994](https://infosys.beckhoff.com/content/1033/ax5000_usermanual/html/ax5000_operationmodes.htm?id=8491271635880908994)

- Beckhoff. (n.d.l). *AX5000 Properties*. Haettu 26. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/content/1033/ax5000\\_system\\_doku\\_hw2/18014399085884811.html?id=8102731246561979472](https://infosys.beckhoff.com/content/1033/ax5000_system_doku_hw2/18014399085884811.html?id=8102731246561979472)
- Beckhoff. (n.d.m). *TwinCAT 3 eXtended Automation*. Haettu 28. 4 2019 osoitteesta Beckhoff:  
[https://download.beckhoff.com/download/document/catalog/TwinCAT\\_3\\_Booklet.pdf](https://download.beckhoff.com/download/document/catalog/TwinCAT_3_Booklet.pdf)
- Beckhoff. (n.d.n). *Programming Languages*. Haettu 28. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/content/1033/tcplccontrol/html/tcplctrl\\_languages.htm?id=529798189732973665](https://infosys.beckhoff.com/content/1033/tcplccontrol/html/tcplctrl_languages.htm?id=529798189732973665)
- Beckhoff. (n.d.o). *TwinCAT Drive Manager*. Haettu 26. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/content/1033/ax5000\\_system\\_doku\\_hw2/9007199899982475.html?id=2077890023184209214](https://infosys.beckhoff.com/content/1033/ax5000_system_doku_hw2/9007199899982475.html?id=2077890023184209214)
- Beckhoff. (n.d.p). *Creating Visualization*. Haettu 26. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_plc\\_intro/3523377803.html&id=7251892297822772979](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/3523377803.html&id=7251892297822772979)
- Beckhoff. (n.d.q). *PLC HMI*. Haettu 26. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/content/1033/tc3\\_plc\\_intro/36028797155077259.html?id=1858649937909089386](https://infosys.beckhoff.com/content/1033/tc3_plc_intro/36028797155077259.html?id=1858649937909089386)
- Beckhoff. (n.d.r). *PLC HMI Web*. Haettu 26. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_plc\\_intro/36028797155077259.html&id=1858649937909089386](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/36028797155077259.html&id=1858649937909089386)
- Beckhoff. (n.d.s). *CTU*. Haettu 27. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/tcplctrl\\_setup.htm&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/tcplctrl_setup.htm&id=)
- Beckhoff. (n.d.t). *Overview Of PLCopen FB's*. Haettu 4. 27 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/tccncmcpatform\\_fb/html/overviewoftheplcopenfbs.htm&id=1805245752876068656](https://infosys.beckhoff.com/english.php?content=../content/1033/tccncmcpatform_fb/html/overviewoftheplcopenfbs.htm&id=1805245752876068656)
- Beckhoff. (n.d.u). *Command MC\_MoveAbsolute*. Haettu 27. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/TcPlcCtrl\\_Languages%20FBD.htm&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/TcPlcCtrl_Languages%20FBD.htm&id=)

- Beckhoff. (n.d.w). *CANopen over EtherCAT*. Haettu 27. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/ax2000-b110/html/ax2000-b110\\_canopen.htm&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/ax2000-b110/html/ax2000-b110_canopen.htm&id=)
- Beckhoff. (n.d.v). *EtherCAT as Drive Bus*. Haettu 27. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/content/1033/ax5000\\_usermanual/html/ax5000\\_ethercat.htm?id=6241818689511109923](https://infosys.beckhoff.com/content/1033/ax5000_usermanual/html/ax5000_ethercat.htm?id=6241818689511109923)
- Beckhoff. (n.d.x). EL20xx Manual. Teoksessa *EL20xx Manual*. Haettu 28. 4 2019
- Beckhoff. (n.d.y). Catalogue TwinSAFE. Haettu 28. 4 2019 osoitteesta  
[http://download.beckhoff.com/download/Document/catalog/Beckhoff\\_TwinSAFE\\_e.pdf](http://download.beckhoff.com/download/Document/catalog/Beckhoff_TwinSAFE_e.pdf)
- Beckhoff. (n.d.y). *Handling of Visualization Pages*. Haettu 28. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_plc\\_intro/3523971851.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/3523971851.html&id=)
- Beckhoff. (n.d.z). *EL6900*. Haettu 28. 4 2019 osoitteesta Beckhoff:  
<https://www.beckhoff.fi/el6900/>
- Beckhoff. (n.d.å). *Safety Editor - Hardware Level*. Haettu 28. 4 2019 osoitteesta Beckhoff Information System:  
[https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_safety\\_editor/html/tc3\\_safetyeditor\\_040\\_aliaslayer.htm&id=16933](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_safety_editor/html/tc3_safetyeditor_040_aliaslayer.htm&id=16933)
- EtherCAT Technology Group. (n.d.ca). *EtherCAT The Ethernet Fieldbus*. Haettu 27. 4 2019 osoitteesta EtherCAT Technology Group:  
<https://www.ethercat.org/en/technology.html#4>
- EtherCAT Technology Group. (n.d.cb). *Safety Over EtherCAT*. Haettu 28. 4 2019 osoitteesta EtherCAT Technology Group:  
<https://www.ethercat.org/en/safety.html>
- Fonselius;Rinkinen;& Vilenius. (1998). *Servotekniikka*.
- Hietalahti, L. (2012). Synkroninen Servomoottori. Teoksessa L. Hietalahti, *Säädetyt Sähkömoottorikäytöt*.
- HSC Physics. (n.d). *Motors And Generators*. Haettu 26. 4 2019 osoitteesta HSC Physics:  
[https://dc.edu.au/dot-point-summary-motors-and-generators/#Motors\\_use\\_the\\_Effect\\_of\\_Forces\\_on\\_Current-carrying\\_Conductors\\_in\\_Magnetic\\_Fields](https://dc.edu.au/dot-point-summary-motors-and-generators/#Motors_use_the_Effect_of_Forces_on_Current-carrying_Conductors_in_Magnetic_Fields)
- Hyperphysics. (n.d.). *Hall Effect*. Haettu 26. 4 2019 osoitteesta Hyperphysics:  
<http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/Hall.html>

- InductiveAutomation. (n.d.). *InductiveAutomation*. Haettu 26. 4 2019 osoitteesta What is HMI?:  
<https://inductiveautomation.com/resources/article/what-is-hmi>
- LearnChannel-TV. (n.d.). *Resolver*. Haettu 28. 4 2019 osoitteesta Learn Channel-TV:  
<http://learnchannel-tv.com/drives/servomotor/resolver/>
- Motion Engineering. (n.d.). *Data types in Sercos*. Haettu 27. 4 2019 osoitteesta Motion Engineering: [http://support1.motioneng.com/soft/zSercos/dta\\_typts.htm](http://support1.motioneng.com/soft/zSercos/dta_typts.htm)
- MotionControlTips. (2015.a). *Resolver vs encoder*. Haettu 26. 4 2019 osoitteesta MotionControlTips: <https://www.motioncontroltips.com/faq-why-are-so-many-designers-replacing-resolvers-with-encoders/>
- MotionControlTips. (24. 2 2016.b). *How are the controls for servo motors tuned?* Haettu 26. 4 2019 osoitteesta MotionControlTips: <https://www.motioncontroltips.com/faq-how-are-the-controls-for-servo-motors-tuned/>
- MouserElectronics. (n.d.). *Motor Control*. Haettu 26. 4 2019 osoitteesta Mouser Electronics: <https://www.mouser.fi/applications/dont-ignore-the-brushed-dc-motor/>
- Parker. (2016). *Struggling to Select the Right Encoder Feedback*. Haettu 28. 4 2019 osoitteesta Parker Blog: <http://blog.parker.com/struggling-to-select-the-right-encoder-feedback-read-this>
- PLCAcademy. (2015). *Structured Text Tutorial*. Haettu 28. 4 2019 osoitteesta PLCAcademy: <https://www.plcademy.com/structured-text-tutorial/>
- PLCopen. (n.d.). *PLCopen Motion Control*. Haettu 28. 4 2019 osoitteesta PLCopen: [http://www.plcopen.org/pages/tc2\\_motion\\_control/](http://www.plcopen.org/pages/tc2_motion_control/)
- ScienceDirect. (n.d.). *Motion Control System using SERCOS over EtherCAT*. Haettu 27. 4 2019 osoitteesta ScienceDirect: [https://ac.els-cdn.com/S1877705811055822/1-s2.0-S1877705811055822-main.pdf?\\_tid=f07d1754-f313-465f-947f-fe97e38c50bc&acdnat=1551376395\\_04c74d29142923561d04045157c89337](https://ac.els-cdn.com/S1877705811055822/1-s2.0-S1877705811055822-main.pdf?_tid=f07d1754-f313-465f-947f-fe97e38c50bc&acdnat=1551376395_04c74d29142923561d04045157c89337)
- SICK. (n.d.da). *Safety Light Curtains C2000*. Haettu 28. 4 2019 osoitteesta SICK Sensor Intelligence: [https://www.sick.com/ag/en/opto-electronic-protective-devices/safety-light-curtains/c2000-standard/c/g187267?q=:Def\\_Type:Product](https://www.sick.com/ag/en/opto-electronic-protective-devices/safety-light-curtains/c2000-standard/c/g187267?q=:Def_Type:Product)
- SICK. (n.d.db). *SICK C/M 2000 Manual*. Teoksessa *SICK C/M 2000 Manual*. Haettu 28. 4 2019
- TKK. (2008). *Servojärjestelmän viritys*. Haettu 26. 4 2019 osoitteesta TKK Automation Technology Laboratory: [http://automation.tkk.fi/attach/AS-0-2230/Labratyo4\\_2008.pdf](http://automation.tkk.fi/attach/AS-0-2230/Labratyo4_2008.pdf)

UaAutomation. (n.d.). *IEC61131-3 Functions and Function Blocks*. Haettu 26. 4 2019 osoitteesta UaAutomation: <http://ua.automation.com/resources-tools/application-stories/programmable-logic-controller-plc/iec-61131-3-functions-and-function-blocks-what-is-the-difference>

## LIITE 1/1: PLCOPEN TOIMILOHKOJA LIIKKEENOHJAUKSEEN

- MC\_MoveRelative - moves the axis a specified distance relative to the actual position at the time of the execution.
- MC\_MoveAdditive - for a specified relative distance additional to the original commanded position in the discrete motion state. In the Continuous Motion the specified relative distance is added to the actual position at the time of the execution.
- MC\_MoveSuperimposed - for a specified relative distance additional to an existing motion. The existing Motion is not interrupted, but is superimposed by the additional motion.
- MC\_MoveVelocity - for a never ending controlled motion at a specified velocity.
- MC\_MoveContinuous - commands a controlled motion of a specified relative distance ending with the specified velocity.
- MC\_TorqueControl - continuously exerts a torque or force of the specified magnitude, approached using a defined ramp, and sets the InTorque output if the torque level is reached.
- MC\_SetPosition - shifts the coordinate system of an axis by manipulating both the set-point position as well as the actual position of an axis with the same value without any movement caused.
- MC\_SetOverride - sets the values of override for the whole axis and allw functions that are working on that axis.
- MC\_TouchProbe - is used to record an axis position at a trigger event.
- MC\_AbortTrigger - is used to abort function blocks which are connected to trigger events (e.g. MC\_TouchProbe).
- MC\_DigitalCamSwitch - provides the analogy to switches on a motor shaft: it commands a group of discrete output bits to switch in analogy to a set of mechanical cam controlled switches connected to an axis. Forward and backward movements are allowed.
- MC\_Home - commands the axis to perform the «search home» sequence. The details of this sequence are manufacturer dependent and can be set by axis' parameters, as well as the function blocks as defined in Part 5 – Homing sequences.

## LIITE 1/2: PLCOPEN TOIMILOHKOJA LIIKKEENOHJAUKSEEN

- MC\_Stop - commands a controlled motion stop and transfers the axis to the state "Stopping". It aborts any ongoing function block execution. With the Done output set, the state is transferred to the StandStill. While the axis is in state Stopping, no other FB can perform any motion on the same axis.
- MC\_Halt - commands a controlled motion stop. It aborts any ongoing function block execution. The axis is moved to the state "DiscreteMotion", until the velocity is zero. With the Done output set, the state is transferred to StandStill.
- MC\_Power - switches the power stage on or off.
- MC\_ReadStatus - returns in detail the status of the axis with respect to the motion currently in progress.
- MC\_ReadAxisError - Indicates errors not relating to the function blocks.
- MC\_Reset - makes the transition from the state ErrorStop to StandStill by resetting all internal axis-related errors and clearing pending commands.
- MC\_ReadParameter & MC\_ReadBoolParameter - Returns the value of a vendor specific parameter.
- MC\_WriteParameter & MC\_WriteBoolParameter - Modifies the value of a vendor specific parameter.
- MC\_ReadActualPosition - returns the actual position.
- MC\_ReadDigitalInput - provides the value of the digital input as referenced by INPUT\_REF.
- MC\_ReadDigitalOutput - - provides the value of the digital output as referenced by OUTPUT\_REF.
- MC\_WriteDigitalOutput - writes a value to the output referenced by the argument "Output" once.
- MC\_ReadActualVelocity - returns the value of the actual velocity as long as enabled.
- MC\_ReadActualTorque - returns the value of the actual torque as long as enabled.
- MC\_PositionProfile - commands a time-position locked motion profile.
- MC\_VelocityProfile - commands a time-velocity locked motion profile.
- MC\_AccelerationProfile - commands a time-acceleration locked motion profile.

## LIITE 2/1: OHJELMALOHKO FB\_MANUAL\_CONTROL

```

//Axis enable

fb_Enable(
  Axis:= AXIS,
  Enable:= stMotor.stControl.bEnable,
  Enable_Positive:= stMotor.stControl.bEnablePositive,
  Enable_Negative:= stMotor.stControl.bEnableNegative,
  Override:= ,
  BufferMode:= ,
  Options:= ,
  Status=> stmotor.stStatus.b_MC_PowerStatus,
  Busy=> ,
  Active=> ,
  Error=> ,
  ErrorID=> );

//FB JOG For Manual moving

mc_Jog(
  Axis:= AXIS,
  JogForward:= stMotor.stControl.bJogForward,
  JogBackwards:= stMotor.stControl.bJogBackward,
  Mode:= 2,
  Position:= stMotor.stControl.fMoveAbsolutePosition,
  Velocity:= stMotor.stControl.fMoveAbsoluteVelocity,
  Acceleration:= stMotor.stControl.fMoveAbsoluteAcceleration,
  Deceleration:= stMotor.stControl.fMoveAbsoluteDeceleration,
  Jerk:= stMotor.stControl.fMoveAbsoluteJerk,
  Done=> ,
  Busy=> ,
  Active=> ,
  CommandAborted=> ,
  Error=> ,
  ErrorID=> );

//Moving to set absolute position

mc_MoveAbsolute(
  Axis:= AXIS,
  Execute:= stMotor.stControl.bMoveAbsolutePosition,
  Position:= stMotor.stControl.fMoveAbsolutePosition,
  Velocity:= stMotor.stControl.fMoveAbsoluteVelocity,
  Acceleration:= stMotor.stControl.fMoveAbsoluteAcceleration,
  Deceleration:= stMotor.stControl.fMoveAbsoluteDeceleration,
  Jerk:= stMotor.stControl.fMoveAbsoluteJerk,
  BufferMode:= ,
  Options:= ,
  Done=> stMotor.stStatus.i_bMoveAbsoluteDone,
  Busy=> ,
  Active=> ,
  CommandAborted=> ,
  Error=> ,
  ErrorID=> );

```

Liitekuva 1: Ohjelmaloheko FB\_Manual\_Motor 1/5.



## LIITE 2/2: OHJELMALOHKO FB\_MANUAL\_CONTROL

```

//FB for NC reset
mc_Reset(
  Axis:= AXIS,
  Execute:= stMotor.stControl.bErrorResetNC,
  Done=> ,
  Busy=> ,
  Error=> ,
  ErrorID=> );

//MC_ReadStatus determines the current operating state of an axis and signals it at the block outputs
mc_ReadStatus(
  Axis:= AXIS,
  Enable:= bReadStatus,
  Valid=> ,
  Busy=> ,
  Error=> ,
  ErrorID=> ,
  ErrorStop=> ,
  Disabled=> ,
  Stopping=> ,
  StandStill=> ,
  DiscreteMotion=> ,
  ContinuousMotion=> ,
  SynchronizedMotion=> ,
  Homing=> ,
  ConstantVelocity=> ,
  Accelerating=> ,
  Decelerating=> ,
  Status=> );

//Status to HMI
stMotor.stStatus.stNcToPlc := AXIS.NcToPlc;
stMotor.stControl.stPlcToNc := AXIS.PlcToNc;
stMotor.stStatus.stReadStatus := mc_ReadStatus;

//FB for reading SoE error

```

Liitekuva 2: Ohjelmaloikko FB\_Manual\_Motor 2/5.

## LIITE 2/3: OHJELMALOHKO FB\_MANUAL\_CONTROL

```

//FB for reading SoE error

fb_SoERead(
  NetId:= ,
  Idn:= ,
  Element:= ,
  pDstBuf:= ,
  BufLen:= ,
  Execute:= bReadSoe,
  Timeout:= ,
  Axis:= AXIS,
  Busy=> ,
  Error=> stmotor.stStatus.bSoeError,
  AdsErrId=> ,
  SercosErrId=> ,
  Attribute=> );

//Drive reset

fb_SoEReset(
  NetId:= ,
  Execute:= stmotor.stControl.bErrorDriveReset,
  Timeout:= ,
  Axis:= AXIS,
  Busy=> ,
  Error=> ,
  AdsErrId=> ,
  SercosErrId=> );

//Axis Halt

mc_Halt(
  Axis:= Axis,
  Execute:= stmotor.stControl.bHalt,
  Deceleration:= ,
  Jerk:= ,
  BufferMode:= ,
  Options:= ,
  Done=> ,
  Busy=> ,
  Active=> ,
  CommandAborted=> ,
  Error=> ,
  ErrorID=> );

```

Liitekuva 3: Ohjelmaloikka FB\_Manual\_Motor 3/5.

## LIITE 2/4: OHJELMALOHKO FB\_MANUAL\_CONTROL

```
//Calibration of the axis (referencing) is carried out with the function block MC_Home.
```

```
mc_Reference(  
  Axis:= AXIS,  
  Execute:= stMotor.stControl.bDriveToReference,  
  Position:= ,  
  HomingMode:= ,  
  BufferMode:= ,  
  Options:= ,  
  bCalibrationCam:= stMotor.stStatus.bReferenceSensor,  
  Done=> stmotor.stStatus.i_bReferenced,  
  Busy=> ,  
  Active=> ,  
  CommandAborted=> ,  
  Error=> ,  
  ErrorID=>  
);
```

Liitekuva 4: Ohjelmaloikko FB\_Manual\_Motor 4/5.

## LIITE 3/1: KUORMITUSOHJELMA

```

CASE nTorque OF

0:
  IF o_bTorqueMode THEN
    // See_IDN_Value := 1048576;
    nTorque := 10;
  END_IF

10:
  GlobalTemplate.MotorControl_Linear.stControl.bDriveToReference := TRUE;
  GlobalTemplate.MotorControl_Servo.stControl.bDriveToReference := TRUE;
  nTorque := 20;

20:
  GlobalTemplate.MotorControl_Linear.stControl.bDriveToReference := FALSE;
  GlobalTemplate.MotorControl_Servo.stControl.bDriveToReference := FALSE;

  IF GlobalTemplate.MotorControl_Linear.stStatus.i_bReferenced THEN
    o_bReferencedLinear := TRUE;
  END_IF
  IF GlobalTemplate.MotorControl_Servo.stStatus.i_bReferenced THEN
    o_bReferencedServo := TRUE;
  END_IF
  IF o_bReferencedLinear AND o_bReferencedServo THEN
    nTorque := 30;
  END_IF

30:
  o_bReferencedLinear := FALSE;
  o_bReferencedServo := FALSE;

// Linear motor going towards servo motor
GlobalTemplate.MotorControl_Linear.stControl.bMoveAbsolutePosition := TRUE;
GlobalTemplate.MotorControl_Linear.stControl.fMoveAbsolutePosition := 1500;
GlobalTemplate.MotorControl_Linear.stControl.fMoveAbsoluteVelocity := 100;
GlobalTemplate.MotorControl_Linear.stControl.fMoveAbsoluteAcceleration := 100;
GlobalTemplate.MotorControl_Linear.stControl.fMoveAbsoluteDeceleration := 100;
GlobalTemplate.MotorControl_Linear.stControl.fMoveAbsoluteJerk := 10000;

// Servo motor going towards linear motor
GlobalTemplate.MotorControl_Servo.stControl.bMoveAbsolutePosition := TRUE;
GlobalTemplate.MotorControl_Servo.stControl.fMoveAbsolutePosition := 1500;
GlobalTemplate.MotorControl_Servo.stControl.fMoveAbsoluteVelocity := 100;
GlobalTemplate.MotorControl_Servo.stControl.fMoveAbsoluteAcceleration := 100;
GlobalTemplate.MotorControl_Servo.stControl.fMoveAbsoluteDeceleration := 100;
GlobalTemplate.MotorControl_Servo.stControl.fMoveAbsoluteJerk := 10000;

```

Liitekuva 5: Kuormitusohjelma 1/2.

## LIITE 3/2: KUORMITUSOHJELMA

```

IF GlobalTemplate.MotorControl_Linear.stStatus.i_bMoveAbsoluteDone THEN
  o_bLinearDone := TRUE;
END_IF
IF GlobalTemplate.MotorControl_Servo.stStatus.i_bMoveAbsoluteDone THEN
  o_bServoDone := TRUE;
END_IF
IF o_bLinearDone AND o_bServoDone THEN
  nTorque := 80;
END_IF

80:
GlobalTemplate.MotorControl_Linear.stControl.bMoveAbsolutePosition := FALSE;
GlobalTemplate.MotorControl_Servo.stControl.bMoveAbsolutePosition := FALSE;

o_bLinearDone := FALSE;
o_bServoDone := FALSE;

nTorque := 80;

90:

//Driving against servo motor
o_nLinearTorquePositive := o_nLinearVisuTorque;
o_nLinearTorqueNegative := o_nLinearVisuTorque;

GlobalTemplate.MotorControl_Linear.stControl.bMoveAbsolutePosition := TRUE;
GlobalTemplate.MotorControl_Linear.stControl.fMoveAbsolutePosition := 10000;
GlobalTemplate.MotorControl_Linear.stControl.fMoveAbsoluteVelocity := 100;
GlobalTemplate.MotorControl_Linear.stControl.fMoveAbsoluteAcceleration := 100;
GlobalTemplate.MotorControl_Linear.stControl.fMoveAbsoluteDeceleration := 100;
GlobalTemplate.MotorControl_Linear.stControl.fMoveAbsoluteJerk := 10000;

SoI_IDM := 158;
Soe_IDM_Value := 100000000;
o_bWriteSoeLinear := TRUE;

100:
o_bWriteSoeLinear := FALSE;
GlobalTemplate.MotorControl_Linear.stControl.bMoveAbsolutePosition := FALSE;
nTorque := 0;
o_bReferencedLinear := FALSE;
o_bReferencedServo := FALSE;
// Tässä kirjoitetaan lag distance koliaksi, jos mahdollista. Nyörs IDW 158 tulisi asettaa defaultiksi,
// sekä non puolesta akselille lajistuneuma

END_CASE

IF GlobalTemplate.MotorControl_Linear.stControl.bHalt OR GlobalTemplate.MotorControl_Servo.stControl.bHalt OR TwinSAFE.bAKS505_Reset OR sMotor.stControl.bErrorResetNC OR sMotor.stControl.bErrorDriveReset THEN
  nTorque := 100;
END_IF

i_fServoCurrent := GlobalTemplate.MotorControl_Servo.stStatus.i_nCurrentValuehis * 0.4/100;
i_fServoTemperature := GlobalTemplate.MotorControl_Servo.stStatus.i_nTemperatureValuehis / 10;

```

Liitekuva 6: Kuormitusohjelma 2/2.

## LIITE 4/1: OHJELMALOHKO FB\_SEQUENCE

```

// Case nSinus carries out endless sequence, moving back and forth position 1 and 2.

CASE nSinus OF
  0:
    IF stmotor.stSequence.bMoveSinusoidal THEN
      nSinus := 10;
    END_IF

  10:
    stMotor.stControl.fMoveAbsolutePosition := stmotor.stSequence.fMoveSinPos1;
    stMotor.stControl.bMoveAbsolutePosition := TRUE;

    IF stMotor.stStatus.i_bMoveAbsoluteDone THEN
      stMotor.stControl.bMoveAbsolutePosition := FALSE;
      nSinus := 20;
    END_IF

  20:
    stMotor.stControl.fMoveAbsolutePosition := stmotor.stSequence.fMoveSinPos2;
    stMotor.stControl.bMoveAbsolutePosition := TRUE;

    IF stMotor.stStatus.i_bMoveAbsoluteDone THEN
      stMotor.stControl.bMoveAbsolutePosition := FALSE;
      nSinus := 10;
    END_IF;

  100:
    stMotor.stControl.bMoveAbsolutePosition := FALSE;
    nSinus := 0;

END_CASE

IF stmotor.stControl.bHalt OR TwinSAFE.bAX5805_Restart OR stmotor.stControl.bErrorResetNC OR stMotor.stControl.bErrorDriveReset THEN
  nSinus := 100;
END_IF

```

Liitekuva 7: Ohjelmaloikko FB\_Sequence.