Jarkko Puska

Kolmeakselisen CNC-jyrsimen suunnittelu ja rakennus

Opinnäytetyö
Kevät 2019
SeAMK Tekniikka
Koneautomaatio
Työn tarkoituksena oli suunnitella ja valmistaa kolmeakselinen jyrsinkone. Työn edetessä tavoitteena oli myös saada kokemusta konesuunnittelun eri aihealueilta. Työn alussa kerrotaan NC-tekniikasta yleisesti sekä lastuavasta työstöstä. Työssä määriteltiin myös jyrsimen käyttötarkoitus, jonka tietojen pohjalta pystytettiin aloittamaan koneen mekaniikka-, sähkö- ja automaatiosuunnittelu.


Grbl-ohjauksella toteutettiin lineariliikkeet kuularuuveilla ja askelmoottoreilla, mutta työn loppuvaiheessa koneen ohjaus vaihdettiin Beckhoffin TwinCat-ohjaukseen. Ohjausen vaihdon takia koneeseen tehtiin sähkömuutoksia ja lineariliikeiden moottorit muutettiin servokäyttöiseksi.

CNC-jyrsin tehtiin kahdella erilaisella ohjauksella ja molemmilla ohjauksilla saatiin lopputuloksena toimiva CNC-jyrsin, jolla pystyttiin jyrsimään puuta, alumiinia ja terästä.
The purpose of the thesis was to design and manufacture a three axis milling machine. As the work progressed, the goal was to gain experience in various areas of machine engineering.

In the beginning of the work NC techniques were studied in general and attention was also paid to the milling process. The purpose of the milling machine was defined, so that mechanical, electrical and automation engineering could be started based on the defined values.

Mechanical engineering was performed with Autodesk Fusion 360 software. The machine risk analysis and electrical components were studied in the electrical engineering section. Electrical drawings were made with the EPLAN design software. In the automation engineering section, two different machine controls were handled. The first one was an Arduino based Grbl control and the second one was a Beckhoff TwinCat control. Various programming languages and general standards for programming were also discussed.

With the Grbl control, linear movements were made with ball screws and stepper motors. In the final stage, the Grbl control was changed to the TwinCat control, so the motors were also updated to servomotors. Because of this it was necessary to make two different electrical- and automation designs.

The CNC milling machine was made with two different controls. Both control systems produced a working CNC Mill, capable of cutting wood, aluminum and steel.

Keywords: CNC, NC, PLC, CAD, CAM, EPLAN, Sistema, TwinCat, Grbl
SISÄLTÖ

Opinnäytetyön tiivistelmä......................................................................................... 2
Thesis abstract ............................................................................................................. 3
SISÄLTÖ....................................................................................................................... 4
Kuvaluettelo .................................................................................................................. 6
Käytetyt termit ja lyhenteet ...................................................................................... 8
1 JOHDANTO............................................................................................................... 9
  1.1 Työn tausta ja tavoite ...................................................................................... 9
  1.2 Työn rakenne .................................................................................................... 9
2 CNC-TEKNIIKKA ................................................................................................... 10
  2.1 Lastuava työstö .................................................................................................. 10
  2.2 CNC-työstökoneen koordinaatisto .................................................................... 11
  2.3 NC-ohjelma ...................................................................................................... 13
3 CNC-JYRSIMEN MÄÄRITTELY ........................................................................... 16
4 MEKANIIKKASUUNNITTELU .............................................................................. 17
  4.1 Autodesk Fusion 360 .......................................................................................... 17
  4.2 Runkovaihtoehdot ............................................................................................. 17
  4.3 Lineaarijohteet ja kuularuuvit ......................................................................... 20
  4.4 Y-akseli ............................................................................................................. 21
  4.5 X-akseli ............................................................................................................. 23
  4.6 Z-akseli ............................................................................................................. 24
  4.7 Runko kokonaisuudessaan .............................................................................. 25
5 SÄHKÖSUUNNITTELU ......................................................................................... 26
  5.1 EPLAN ................................................................................................................. 26
  5.2 Karamoottori ja taajuusmuuttaja ....................................................................... 27
  5.3 Sähkösuunnittelun Grbl-ohjaukseen ................................................................. 27
    5.3.1 Virtalähteet .................................................................................................. 28
    5.3.2 Askelmoottorit ja askelmoottoriohjaimet ......................................................... 29
    5.3.3 Grbl-kytkentä ............................................................................................... 30
  5.4 Sähkösuunnittelun TwinCat 3 -ohjaukseen ....................................................... 31
    5.4.1 Virtalähteet .................................................................................................. 31
5.4.2 Boshrexroth-servomoottorit ................................................................. 31
5.4.3 IndraDrive CS -servovahvistimet ............................................................ 31
5.5 Koneen toimintakuvaus automaattiosuunnittelua varten .............................. 34
5.6 Hätäseisipiiri ja Sistema-riskianalyysi ....................................................... 35
6 AUTOMAATTIOSUUNNITTELU .................................................................. 39
  6.1 IEC 61131-3 ja PLCopen ............................................................................. 39
  6.2 Grbl v1.1 ..................................................................................................... 39
    6.2.1 Grbl-ohjauksen käyttöönotto ................................................................. 40
    6.2.2 Grbl-ohjauksen käyttöliittymä ............................................................... 42
  6.3 Beckhoff TwinCAT 3 .................................................................................. 42
    6.3.1 TC3 PLC ............................................................................................... 43
    6.3.2 TC3 I/O .................................................................................................. 44
    6.3.3 TC3 Motion control ............................................................................... 46
    6.3.4 TC3-koneen käyttöliittymä ................................................................. 48
  6.4 IndraDrive CS -vahvistimien ja -servomoottorien käyttöönotto ..................... 49
7 POHDINTA ........................................................................................................ 56
LÄHTEET ........................................................................................................... 57
LIITTEET ............................................................................................................ 60
Kuvaluettelo

Kuva 1. Koordinaatisto.................................................................11
Kuva 2. Kappalekoordinaatiston orgo. ........................................12
Kuva 3. NC-ohjelmaesimerkki......................................................13
Kuva 4. NC-ohjelmassa käytettäviä osoitekirjaimia............................14
Kuva 5. NC-ohjelmassa käytettäviä M-koodeja..................................15
Kuva 6. Portaalimallinen tasojyrsinkone.........................................18
Kuva 7. Polvimallinen tasojyrsinkone..............................................18
Kuva 8. Runkomallinen tasojyrsinkone..........................................19
Kuva 9. Lineaarijohde ja kuularuuvi. .............................................20
Kuva 10. Y-akselin runko. ..........................................................21
Kuva 11. X-akselin runko. ..........................................................23
Kuva 12. Z-akselin runko. ..........................................................24
Kuva 13. CNC-koneen runko kokonaisuudessaan...............................25
Kuva 15. WanTai Motor-paketin toimitussisältö...............................28
Kuva 16. Grbl-ohjauksen kytkentäkuva.........................................30
Kuva 17. IndraDrive Cs -vahvistin...............................................32
Kuva 18. IndraDrive Cs HCS01 -kytkentäsimerkki...........................33
Kuva 19. CNC-jyrsimen layout automaattiosuunnittelua varten............34
Kuva 20. Sistema suoritustason riskigraafi. .....................................36
Kuva 21. Turvareelen PL-taso valmistajan ohjekirjasta. ....................38
Kuva 22. Sistema riskianalysin lopputulos. ................................................................. 38
Kuva 23. Grbl-ohjauksen asetusten määrittely Grbl Panel -ohjelmiston avulla. ..... 41
Kuva 24. Grbl Panel -käyttöliittymän ulkonäkö......................................................... 42
Kuva 25. PLC-lohkon näkymä.................................................................................... 44
Kuva 26. I/O-lohkon näkymä.................................................................................... 45
Kuva 27. Motion-lohkon näkymä................................................................................ 46
Kuva 28. TwinCat 3 visualization -käyttöliittymä. .................................................... 48
Kuva 29. Yhteyden muodostaminen IndraDrive Cs -vahvistimille. ....................... 49
Kuva 30. Kommunikointi vahvistimien ja ohjauksen välillä.................................... 50
Kuva 31. Akselien mekaaniikan ja skaalauksen asetukset. ..................................... 51
Kuva 32. Akselien säätöpiiriasetukset...................................................................... 52
Kuva 33. Servovahvistimen asettaminen Easy Startup -tilaan. ............................... 52
Kuva 34. Akselin säätöpiirin automaattisen määrityksen aloittaminen. ............. 53
Kuva 35. Akselin säätöpiirin automaattisen määrityksen ensimmäinen vaihe ...... 54
Kuva 36. Akselin säätöpiirin automaattisen määrityksen toinen vaihe. ............... 55
Kuva 37. Akselin säätöpiirin automaattisen määrityksen kolmas ja viimeinen vaihe.. 55
# Käytetyt termit ja lyhenteet

<table>
<thead>
<tr>
<th>Lyh.</th>
<th>Plk.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNC</td>
<td>Computer numerical control, tietokoneohjattu numeerinen ohjaus.</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable logic control, ohjelmoitava logiikka.</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-aided drafting, tietokoneavusteen piirtäminen.</td>
</tr>
<tr>
<td>CAM</td>
<td>Computer-aided manufacturing, tietokoneavusteen NC-koodin tekeminen.</td>
</tr>
<tr>
<td>I/O</td>
<td>Inputs and outputs, tulot ja lähdöt.</td>
</tr>
<tr>
<td>Interpolointi</td>
<td>Rataohjaus, jossa useampaa akselia voidaan liikuttaa hallitusti yhdenaikaisesti.</td>
</tr>
<tr>
<td>Interpolaattori</td>
<td>NC-koodin kääntäjä ohjaukselle.</td>
</tr>
<tr>
<td>PL</td>
<td>Turvalaitteiden suoritustaso.</td>
</tr>
<tr>
<td>PLr</td>
<td>Turvalaitteilta vaadittava suoritustaso.</td>
</tr>
</tbody>
</table>
1 JOHDANTO

1.1 Työn tausta ja tavoite


1.2 Työn rakenne

Työn alussa käydään läpi NC-tekniikkaa yleisesti, sekä käsitellään lastuavaa työstöä ja NC-ohjelmointia. Ennen suunnittelun aloittamista määritellään rakennettavan CNC-jyrssimen käyttötarkoitus sekä liikealueet.

Mekaniikasuunnittelussa käydään läpi Autodesk Fusion 360 -ohjelmistoa, erilaisia runkovaihtoehtoja, sekä itse koneen rungon valmistusta.

Sähkösuunnittelussa käsitellään EPLAN-ohjelmistoa, sekä kahden erilaisen ohjauksen kanssa käytettäviä komponentteja ja niiden kytkentää. Koneesta tehdään myös turvalaitteiden riskianalyysi Sistema-ohjelmistolla.

Automatiotosuunnittelussa käsitellään kahtia erilaista ohjausta. Ensimmäisenä tarkastellaan Arduino pohjaista Grbl-ohjausta ja tämän jälkeen TwinCat 3 -ohjausta.
2 CNC-TEKNIKKA

CNC on lyhennettävä sanasta computer numerical control, joka tarkoittaa tietokoneohjattua numeerista ohjausta. CNC-koneen tarkoituksena on suorittaa hallittuja liikkeenohjauksia työkalun ja työkappaleen välillä. CNC-teknologiaa käytetään koneissa, jotka yleensä perustuvat materiaalin poistamiseen, lisäämiseen tai muodon muokkaukseen. Materiaalin poistaminen suoritetaan työkalulla jyrsimällä, sorvamalla, hiomalla tai lävistämällä. (Kandray 2010, 15.) 3D-tulostin on hyvä esimerkki materiaalin lisäämisestä. Materiaalin muokkaus keinoja ovat esimerkiksi putkentaivutuskoneet ja levyntaivutusautomaatit.


2.1 Lastuava työstö


Manuaalityöstökone ja CNC-työstökone noudattavat samoja lastuavan työstön periaatteita. Suurin ero näiden koneiden välillä on ohjaustavassa. CNC-koneelle tehdään NC-ohjelma, jonka koneen ohjaus suorittaa. Manuaalikoneessa koneen käyttäjä huolehtii akselien liikkeistä sekä karamoottorin nopeudesta. (Vesamäki 2014, 10.)
Materiaalia koneistaessa pitää osata määrittää oikea lastuamisnopeus, joka tarkoittaa pyörivän työkalun ja koneistettavan kappaleen liikettä toisiinsa nähden. Sopivan lastuamisnopeuden pystyy tarkistamaan taulukoista, jos koneistettava materiaali ja työkalun materiaali on tiedossa. (Maaranen 2012, 16.)

Lastuamisnesteiden käyttö koneistuksessa pienentää työkalun ja kappaleen välistä kitkaa ja samalla jäähdytää kuljettaen koneistuksessa irronneet lastut pois. Lastuamisnesteiden valinta tulisi tehdä nesteen valmistajan ohjeiden mukaan. Koneen alla on yleensä metallinen kuljetin, joka kuljettaa lastut pois työstökoneen alueelta. (Maaranen 2012, 23.)

2.2 CNC-työstökoneen koordinaatisto


![Diagram of CNC coordinate system](image)

Kuva 1. Koordinaatisto (Siemens [Viitattu 25.5.2019]).

CNC-työstökoneissa voi olla useampia koordinaatiston nollapisteitä. Osa on kiinteitä pisteitä, joita ei voida muuttaa, mutta esimerkiksi kappalekoordinaatistoa voi-
daan muuttaa nollaamalla koneen akselien arvot NC-ohjelmassa määritettynä pisteeseen. Kiinteitä koordinaatiston pisteitä ovat esimerkiksi koneen nollapiste, joka toimii koneen koordinaatiston perustana. Kaikki muut nollapisteet käyttävät konekoordinaatiston arvoja, myös koneen maksimiliikealueet lasketaan konekoordinaatistosta. (Maaranen 2012, 378.)

Kun työstöohjelmaa tehdään, ohjelman teon aikana määritellään kappalekoordinaatiston origo. Origo tarkoittaa pistettä, jossa kaikkien akselien koordinaatit ovat nollassa. NC-ohjelmassa olevat liikkeet suoritetaan tämän pisteen mukaan, jos ohjelman alussa aktivoidaan kappalekoordinaatiston käyttö. (Maaranen 2012, 379.)

Kuva 2. Kappalekoordinaatiston origo.

NC-työstökoneita käytetään pääasiassa automaattikäytöllä, mutta manuaalikäytöllä voidaan esimerkiksi ajaa kone kappalekoordinaatiston nollauspaikkaan.
2.3 NC-ohjelma

NC-ohjelma voidaan tehdä joko käsin ohjelmoimalla tai tietokoneavusteisesti (Vesamäki 2014, 61).

Käsinojelmoiminta on helppoa sitä osaavalle, mutta siitäkin muodostuu työläs prosessi, jos työstettävä kappale on monimutkainen. Ohjelmoijan tulee määrittää ja laskea akselin liikkeet, lastuamisnopeudet, työstöjärjestykset jne. sekä kirjoittaa NC-ohjelma. (Vesamäki 2014, 61.)

Tietokoneavusteisissa ohjelmissa (CAD/CAM) käytetään työstettävän kappaleen CAD-kuvaa pohjana työstöratoja tehdessä. CAD-ohjelmissa pystytään piirtämään 3D-mallinnus työstettävästä kappaleesta ja CAM-ohjelmasta tehdään työstökoneelle NC-ohjelma. CAM-ohjelmistossa pystytään yleensä simuloimaan työstöohjelmaa ennen kuin sitä testataan oikealla työstökoneella. CAD- ja CAM-toiminnot voivat löytyä samasta ohjelmistosta tai voidaan käyttää kahta erillistä suunnittelutoimintaa. (Pikkarainen & Mustonen 2010, 110.)

NC-ohjelma sisältää erilaisia sanoja ja näistä sanoista muodostuu NC-lause. Yksi NC-lause sisältää yhtä työvaihetta koskevaa tietoa, näiden lauseiden muodostamaa listaa kutsutaan NC-ohjelmaksi. (Vesamäki 2014, 64.)

Kuva 3. NC-ohjelmaesimerkki.
Sanassa esitellään aluksi osoite, jonka jälkeen syötetään osoitteen lukuarvo. Osoite on kirjain. Tämä kirjain ilmaiseen, mihin tarkoituksseen osoitteen perässä esitettyä lukua käytetään. (Vesamäki 2014, 64.) Kuvassa 4 on esitelty yleisempiä osoitekirjaimia ja niiden merkitys.

<table>
<thead>
<tr>
<th>Kirjain</th>
<th>Osoitekirjaimen merkitys</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>- ohjelman numero</td>
</tr>
<tr>
<td>N</td>
<td>- lausenumero</td>
</tr>
<tr>
<td>G</td>
<td>- valmistelevat toiminnot, liiketila</td>
</tr>
<tr>
<td>X, Y, Z</td>
<td>- liikekoordinaatit koordinaattiaukseleiden suuntaan</td>
</tr>
<tr>
<td>A, B, C</td>
<td>- lisääkseleiden liikekoordinaatit</td>
</tr>
<tr>
<td>U, V, W</td>
<td>- lisääkseleiden liikekoordinaatit</td>
</tr>
<tr>
<td>R</td>
<td>- ympyräkaaren sädä, kulman pyöristy</td>
</tr>
<tr>
<td>I, J, K</td>
<td>- kaaren keskipisteen koordinaatit, viisteet</td>
</tr>
<tr>
<td>F</td>
<td>- syöttönopeus</td>
</tr>
<tr>
<td>S</td>
<td>- karan pyöränopeus</td>
</tr>
<tr>
<td>T</td>
<td>- työkalun numero</td>
</tr>
<tr>
<td>M</td>
<td>- kytkentätoiminto</td>
</tr>
<tr>
<td>B</td>
<td>- pyöräpöydän asema</td>
</tr>
<tr>
<td>H</td>
<td>- pituuden kompensointinumero</td>
</tr>
<tr>
<td>D</td>
<td>- sateen kompensointinumero</td>
</tr>
<tr>
<td>P, X</td>
<td>- viive</td>
</tr>
<tr>
<td>P</td>
<td>- alichjelman numero, toistettavan lauseen lausenumero</td>
</tr>
<tr>
<td>L</td>
<td>- toistokertojen määrä</td>
</tr>
<tr>
<td>P, Q, R</td>
<td>- kuntean työkierron parametrit</td>
</tr>
</tbody>
</table>

Kuva 4. NC-ohjelmassa käytettäviä osoitekirjaimia (Opetushallitus [Viitattu 25.4.2019]).

G-koodit voivat tarkoittaa esimerkiksi liikekäskyn tyyppiä, joka määrittelee, onko kyse pikaliikkeestä, suoraviivaisesta liikkeestä tai akselien yhdenaikaisesta interpo-loinnista. G-koodit voivat olla modaalisia, eli käskey pysyy voimassa niin kauan kun-
nes toinen käsky kumoaa edellisen. G-koodit on luokiteltu eri ryhmiin, kuten liiketyypeihin, koordinaatiston ja mittayksikköjen hallintaan. Eri ryhmään kuuluvia G-koodeja pystytään esittelemään samassa NC-lauseessa. (Vesamäki 2014, 66.)


M00 - ohjelman loppu
M01 - valinnainen pysäytys
M02 - ohjelman loppu
M03 - karan käynnistys myötäpäivään
M04 - karan käynnistys vastapäivään
M05 - karan pysäytys
M06 - työkalun vaihto
M08 - jäädytys päälle
M09 - jäädytys pois päältä
M10 - oheislaitteen lukinta
M11 - oheislaitteen lukinnan avaus
M19 - karan pysäytys vakiokulmaan
M30 - ohjelman loppu

Kuva 5. NC-ohjelmassa käytettäviä M-koodeja (Opetushallitus [Viitattu 25.4.2019]).
3 CNC-JYRSIMEN MÄÄRITTELY

Ennen suunnittelutyön aloittamista määriteltiin koneen käyttötarkoitus, liikealueet ja niihin käytettävät komponentit, jotta saatiin koneen suunnittelun aloittamiseen tarvittavat alku tiedot.

Koneessa on kolme akselia ja sillä on tarkoitus työstää puuta ja pehmeitä metallia, esimerkiksi alumiinia. Teräksen koneistusta on tarkoitus kokeilla testimielessä. Koneen liikealue tulee olla vähintään X-suunnassa 1000 mm, Y-suunnassa 500 mm ja Z-suunnassa 400 mm. Jokainen lineaariliike suoritetaan lineaariohjeksi ja kuularuuveilla. Kuularuuveja pyöritetään moottoreilla.


4 MEKANIIKASUUNNITTELU

4.1 Autodesk Fusion 360


Saatailla on erilaisia 3D-suunnittelutoimintoja, mutta kyseinen ohjelmisto osoittautui sopivaksi ohjelmaksi jyrssimen työstöohjelman tekoon. Ohjelmisto on helppo käyttää, ja verkosta löytyy hyvin materiaalia ohjelman käytön aloittamiseksi. Koneen mallinnus on suoritettu myös kyseisellä ohjelmistolla.

4.2 Runkovaihtoehdot

Runkovaihtoehtoina CNC-jyrssimeen harkittiin portaali-, polvi- tai runkomalliista ratkaisua.


Polvimallisessa tasojyrskinkoneessa karamoottorin työkalu pysyy paikallaan ja pöytää liikuttamalla toteutetaan työstön syöttöliike X-, Y- ja Z-suunnassa. Tämä on yleisin malli, jota käytetään pienissä ja keskisuurissa manuaalityösköneissä. Näillä koneistetaan yleisemmin pienehköjä työkappaleita (Maaranen 2012, 244.)

Kuva 7. Polvimallinen tasojyrskinkone (Kandray 2010, 97).


CNC-jyrsimen malliksi valittiin runkomallinen tasojyrssinkone, koska tämä vaikuttaa sopivan koneen käyttötarkoituksen parhaiten. Jyrsimellä on tarkoituksena kokeilla myös teräksen koneistamista. CNC-jyrsin joudutaan valmistamaan ilman kappaleiden koneistusmahdollisuutta, joten tämä tuo haasteita koneen mekaanisiin säätötoimenpiteisiin.
4.3 Lineaarijohteet ja kuularuuvit

Lineaarijohde on profiloitu liukukisko, jota pitkin asennettu laakerikelkka pääsee liukumaan. Laakerikelkka sisältää pieniä kuulia, jotka mahdollistavat melkein kitkattoman liikkeen. Lineaarijohdepaketteja on saatavilla kaiken kokoisina ja mallisina. (Hiwin [Viittattu 2.5.2019].)


4.4 Y-akseli

Y-akselin rungon valmistukseen on käytetty 50 mm x 100 mm teräsprofilia, jonka vahvuus on 5 mm. Tämä runko toimii koko koneen alustana. Runkoon on käytetty laserleikattuja osia, joita on kuularuuvin ja moottorin kiinnikkeissä sekä takana olevassa tasossa, johon Z-akselin pylväs kiinnitetään. Rungon teräsprofilit on kiinnitetty toisiinsa hitsaamalla, samalla on pidetty rungon suoruus mahdollisimman hyvänä. Tulevat johdepinnat on kevyesti hiottu tasahiomakoneella.


Aluksi toinen johdeista asennettiin rungon päälle ja johteen kyljen- ja pinnansuoruseen tarkastettiin käytettävällä tarkkuusvatupasseja. Työvaiheessa käytetyn tarkkuusvatupassin yhden viivan välinen tarkkuus on 0,02 mm metriä kohti. Johteen alapinnalla
jouduttiin käyttämään korotuspaloja, jotta johde olisi mahdollisimman suorassa kor- keussuunnassa. Tämän jälkeen johde pultattiin runkoon kiinni. Tätä johdetta voi-
daan käyttää toisen johteen asentamisessa. Asennettuun johteesseen asetettiin yksi laakerikelkka, johon on kiinnitetty magneettijalalla varustettu mittakello. Tämän jäl-
keen toinen johde asetellaan paikalleen ja mittakellon mittakärki laitetaan asennet-
tavan johteen kylkeen. Mittakelloa siirtämällä saadaan uusi asennettava johde käännettyä samansuuntaiseksi jo asennetun johteen kanssa, se voidaan tämän jäl-
keen pultata kiinni. Uuden johteen korkeus voidaan tarkastaa asettamalla mittakel-
lon kärki johteen yläpintaan ja siirtämällä mittakelloa. Tällä saadaan selville johteen
alle asetettavien korotuspalojen määrä. Mittausten jälkeen molempiin johteen tu-
lisi olla suorassa toisiinsa nähdä.

Kuularuuvi on 800 mm pitkä ja 32 mm halkaisijaltaan. Kuularuuvin nousu on 40 mm.
Nousu tarkoittaa, kuinka paljon kuularuuvissa oleva mutteri liikkuu yhden pyörä-
hyksen aikana. Moottorin ja kuularuuvin välinen liitos tapahtuu hammashihnapyö-
rillä ja hihnalla. Välityssuhde on 5:1 eli moottorin pitää pyöräähtää viisi kertaa ympäri, että kuularuuvi saadaan pyörähtämään kerran. Samaa välityssuhdentta käytetään
myös X- ja Z-akseleilla. Kuularuuvi asennetaan päädytä paikalleen ja kuulamutteri
kiinnitetään seuraavassa vaiheessa X-liikkeen pöytään.
4.5 X-akseli


Kuva 11. X-akselin runko.
4.6 Z-akseli


4.7 Runko kokonaisuudessaan

Kuva 13. CNC-koneen runko kokonaisuudessaan.


CNC-jyrsimen liikealueeksi muodostui Y-suunnassa 520 mm, X-suunnassa 1010 mm ja Z-suunnassa 410 mm.
5 SÄHKÖSUUNNITTELU

Tässä osiossa käsitellään kahden erilaisen koneenhauksen sähkösuunnittelua ja niihin käytettäviä komponentteja. Grbl-hauksen kanssa on käytetty valmistajan sivustolta löytyviä kytkentäesimerkkejä, mutta TwinCat 3-haukseen on piirretty sähkösuunnitelmat EPLAN-sähkösuunnitteluhjelmistolla.

TwinCat 3-hauksen sähkösuunnittelueloon on sisällytetty myös toimintakuvauksen tekeminen automaatirosuunnittelun avuksi.

5.1 EPLAN

EPLAN on laaja-alainen suunnitteluhjelmisto. Ohjelmisto voi sisältää toiminnallisen suunnittelun, tekniilisen esi-suunnittelun, hydraulikasuunnittelun, sähkösuunnittelun, keskusten suunnittelun ja johdinsarjojen suunnittelun. EPLAN Platform on alusta, joka yhdistää nämä kaikki suunnitteluhjelmistot toisiinsa ja mahdollistaa tiedonkulun eri ohjelmistojen välillä. (Eplan [Viitattu 12.5.2019].)


5.2 Karamoottori ja taajuusmuuttaja

Karamoottoriksi on valittu teholtaan 1.5 kW:n kolmevaihemoottori, jota ohjataan taajuusmuuttajalla. Kyseisessä moottorissa on vesijäähdytys ja ER16-työkalunpidike. Karamoottorin maksimi kierrosnopeus minuutissa on 16 000, mikä soveltuu hyvin puunjyrsintään.


Moottorin jäähdytys on tehty erillisellä jäähdytinyksiköllä, jota ohjaaa Arduino nano-logiikka. Jäähdyttimesä on vesisäiliö ja lämmönvaihdin, joiden sisällä kiertää jäähdytinneste. Logiikkaan on kytetty karamoottorin lämpötila-anturi, jonka mukaan jäähdytinyksikön pumppu käynnistyy, pitääen lämpötilan halutulla alueella.

5.3 Sähkösuunnitelu Grbl-ohjaukseen


Työhön on valittu JiangSu WanTai Motor co., Ltd:n toimittamat askelmoottorit ja askelmoottorihjaimet. Toimitus sisälsi valmiin kolmen akselin paketin, jossa oli kolme askelmoottoria, kolme askelmoottorihjainta sekä näille virtalähteet.

Toimitussisältö oli:

230 VAC → 12 VDC:n virtalähde

230 VAC → 36 VDC:n virtalähde

2 kpl DQ542MA-askelmoottoriorhainta

2 kpl 57BYGH115.003B -askelmoottoria

230 VAC → 60 VDC:n virtalähde

1 kpl DQ860MA-askelmoottoriorhain

1 kpl WT86STH118-6004A-askelmoottori

5.3.1 Virtalähteet


5.3.2 Askelmoottorit ja askelmoottoriohjaimet


Työssä käytettävissä askelmootoreissa on kaikissa 1,8 asteen askelkulma.


WT86STH118-6004A-askelmoottori tulee Z-liikkeen moottoriksi. Tämä on Nema 34 -kokoluokan moottori, jossa pitomomentti on 9,4 N/m, ja moottorin ottama maksimivirta on 5,6 A. Kyseistä askelmoottorien ohjataan DQ860MA-askelmoottoriohjaimella, joka saa käyttöjännitteen 60 VDC:n virtalähteestä.
5.3.3 Grbl-kytkentä


Askelmanottoriohjaimet tarvitsevat suuntatiedon (Direction), askelpulssin (Step Pulse) ja luvan liikkua (Stepper Enable/Disable). Grbl-ohjauksen sivustolta löytyy muutamia kytkentäesimerkkejä erilaisille askelmanottoriohjaimille.
5.4 Sähkösuunnittelu TwinCat 3 -ohjaukseen

TwinCat 3 -ohjauksen kanssa X-, Y- ja Z-moottoreina on käytetty Bosch Rexrothin servomoottoreita. Moottoreita ohjataan saman valmistajan IndraDrive Cs -servovahvistimilla. Sähkösuunnitelmissa on käytetty valmistajan sivuilta löytyviä kytkentäesimerkkejä.

5.4.1 Virtalähteet

TwinCat 3 -ohjauksen kanssa käytettävät sähköiset komponentit toimivat joko 3 x 400 V:n vaihtojännitteellä tai 24 V:n tasajännitteellä, joten työssä ei tarvita kuin 24 V:n tasajännitevirtalähde.

5.4.2 Boshrexroth-servomoottorit


5.4.3 IndraDrive CS -servovahvistimet

IndraDrive Cs HCS01 -sarjan servovahvistimet ovat kompakteja vahvistimia, joissa on yhdistettynä tehonsyöttö- ja ohjausmoduuli. Vahvistimissa on tuki Ethernet-pohjaisille tiedonsiirtomenetelmille, erilaisille pulssiantureille, näissä on myös sisäänrakennettu turvatoimintoteknologia. (Boshrexroth [Viitattu 14.5.2019].)

Kuva 16. IndraDrive Cs -vahvistin.

HCS01-sarjan servovahvistimen syötön edessä suositellaan käytettäväksi kuristinta HNL01. Tämä varmistaa korkeamman ja tasaisemman tehon vahvistimille ja kuristimet vähentävät harmonisia aaltoja syötössä, minkä takia häiriöt järjestelmässä vähentyvät. (Bosch Rexroth [Viitattu 15.5.2019].)
Wiring Diagram

Kuva 17. IndraDrive Cs HCS01 -kytkentäesimerkki (Boschrexroth 11.3.2013).

Optional
24V Control voltage supply
HLC DC bus capacitor unit (for devices with DC bus connection)
COM Communication
DST Autotransformer
F Fuses
HCS01 Converter
NF Mains filter
HNL Mains choke
K1 External mains contactor
M Motor
RB Braking resistor (at the back of the drive controller)

Fig.5-1: Drive System Rexroth IndraDrive Cs
5.5 Koneen toimintakuvaus automaatiosuunnittelua varten

Kuva 18. CNC-jyrsimen layout automaatiosuunnittelua varten.


Karamoottorina toimii kolmevaiheinen moottori 20M1, jota ohjataan taajuusmuuttajalla 20U1. Taajuusmuuttajan kierrospaineutetta ohjataan analogisignaalilalla kortin 2A9_AO kautta. Muut ohjaussignaalit ovat lähtökortilla 2A5_DO. o_SpindleForward, o_SpindleReverse, o_SpindleReset.

Koneen alla on jyrsinlastuille tarkoitettu jätekuljetin, joka siirtää lastut koneen suojauksien ulkopuolelle. Kuljetinta ohjataan lähdöllä o_ScrapConveyorON.
Jyrsintapin ympärillä on paineilmasylinteriillä laskeva lastusuoja, josta on tarkoitus myös imeä lastua/pölyä imurilla. Sylintereiden rajat 3SQ7 ja 3SQ8 on kytetty 2A3_D1-tulokorttiin. Imurille on varattuna pistorasia sähkökaapissa, jonka toimintaa voidaan ohjata lähtökortin 2A5_DO: o_AspiratorON lähdöllä.

Koneen ympärille tulee suojaseinät, joissa on käyntiovi koneen sisälle. Häätäiseispappeja on kaksi kappaletta SF1 ja SF2, toinen on suojauksien ulkopuolella ja toinen sisäpuolella. SF1-napissa on myös hätäseisipiirin kuittausnappi. HS-piirissä käytetään Pilz-turvareleitä, joissa on 0,5 sekunnin päästöhidas. Koneelle annetaan 0,5 sekuntaa aikaa pysäyttää liikkeet hallitusti, ennen kuin syöttö katkaistaan.


5.6 Häätäiseispäiri ja Sistema-riskianalyysi

Sistema on saksalaisen työturvallisuus- ja työterveyslaitoksen IFA:n kehittämä ilmainen ohjelmisto koneturvallisuuden tason arviointiin ja testaamiseen. Sistema tarkoittaa ”Safety Integrity Software Tool for the Evaluation of Machine Applications“.

Ohjelmisto noudattaa standardia ISO 13849-1, joka on kansainvälinen koneturvallisuuden standardi. (IFA [Viitattu 15.5.2019].)

Sistema-ohjelmistoon on tarkoitus lisätä kaikki koneessa käytettävät turvalaittekomponentit ja määritellä jokaisen komponentin suoritustaso PL. PL-luokkia on yhteensä viisi a, b, c, d, e, joista a tarkoittaa matalimman turvallisuustason vaatimusta ja e tarkoittaa suurinta turvallisuustason vaatimusta. Koneessa olevat turvalaitteet voidaan lisätä ohjelmistoon joko käsineen komponentti- tai komponentit voidaan lisätä valmistajan tekemän kirjaston mukaan. IFA:n sivuilta löytyy lista valmistajista, jotka tukevat suoraan komponenttitikirjaston lisäämistä.

Ohjelmistossa on työkalu, jolla voidaan määritellä koneen turvalaitteiden vaatima suoritustaso PLr. Ohjelmiston laskema turvalaitteiden suoritustason PL-luokan tulee siis olla suurempi tai yhtä suuri kuin koneeseen käsin määritelty suoritustason PLr-taso.

Seuraavaksi selvitetään CNC-jyrsityöhön tehty Sistema-riskianalyysin PLr-tason määrittely ja määrittelyyn liittyviä muuttujia.


![Sistema riskigraafi](image)

PLr-tason ensimmäinen määritelmä on vamman vakavuusluokka, joka voi olla S1 tai S2. S1-luokka tulee valita, jos kyseessä on pienet ruhjeet tai lievät vammat. S2-luokka valitaan, jos kyseessä on vakavat vammat, irtileikkautuminen tai kuolema. (SFS-EN ISO 13849-1 2015, 53.) Tähän työhön on valittu S2-luokka, koska koneessa käytetään erilaisia jyrsintyökaluja, joilla tulevat vammat voivat olla vakavia. Myös koneen liikkuvien osien väliliin joutuminen on vaarallista.


Näiden sytetyjen tietojen perusteella ohjelma näyttää turvalaitteilta vaadittavan suoritustason PLr. Tässä työssä määritellyjen luokkien perusteella vaadittava suoritustaso PLr on d.


Kun kaikki komponentit on lisätty listalle ja näille on määritelty suoritustasot, ohjelma laskee syötettyjen arvojen perusteella kaikkien turvalaitteiden suoritustason eli PL- arvon.


Työssä käytetyillä turvalaitteilla PL-arvoksi muodostui d. Sisteman riskigraafin mukaan, vaadittava suoritustaso PLr tulisi olla vähintään d, joten koneen turvalaitteet ovat tarpeeksi turvalliset kyseiseen laitteeseen.
6 AUTOMAATIOSUUNNITTELU

Tässä osiossa käsitellään kahden erilaisen koneenohjauksen käyttöönottoa sekä automaatiosuunnittelun yleisiä ohjelmointikieliä.

6.1 IEC 61131-3 ja PLCopen

IEC 61131-3 on kansainvälinen standardi, joka käsittelee ohjelmointikieltä. Standardin tarkoituksena on muodostaa alalla oleville valmistajille yhtenäisen ohjelmointikieltä, mutta standardissa hyväksytään myös valmistajien omien ohjelmointisääntöjen käyttöä. Standardissa käsitellään viitta erilaista ohjelmointikieltä. (Sands & Verhappen 2018, Chapter 16.)

IEC 61131-3-standardin mukaiset ohjelmointikiellet ovat:

- Structured Text (ST)
- Instruction List (IL)
- Function Block Diagram (FBD)
- Ladder Diagram (LD)
- Sequential Function Chart (SFC). (Sands & Verhappen 2018, Chapter 16.)

PLCopen on kansainvälinen organisaatio, jonka tarkoituksena on kehittää ohjelmointikielette liikkeenohjaukseen ja liikkeenohjauksiin ja turvatoimintoihin. (PLCopen [Viitattu 25.5.2019].)

6.2 Grbl v1.1

Grbl on Arduino-pohjainen avoimen lähdekoodin koneenhajausjärjestelmä. Ohjausta voidaan käyttää kolmeakselisen koneen ohjauksessa ja ohjelmisto toimii kai-
kissa Arduinoissa, joissa on a 328p mikroprosessori (Uno, nano). Ohjaus on suunniteltu askelmoottorikäyttöön, Grbl pystyy tuottamaan 30 kHz ohjaussignaalia. Tämä tarkoittaa, että askelmoottoreille voidaan syöttää 30 000 pulssia sekunnissa. (Github [Viitattu 28.4.2019].)


Ohjauksessa on kiihtyvyydenhallintajärjestelmä, joka lukee NC-ohjelmasta 16:sta liikekomentoa ennen niiden suorittamista ja suunnittelee nopeudet näiden mukaan. Tämä mahdollistaa sulavat liikkeenohjaukset. (Github [Viitattu 28.4.2019].)

Tällä hetkellä version pääkehittäjänä toimii Sungeun Jeon, nykyinen ohjaus on kehitetty Simen Svale Skogsrudin vuonna 2009 aloittamasta Grbl-projektista. Grbl on ilmainen ohjelmisto ja se on julkaistu GPLv3-lisenssin alla. (Github [Viitattu 28.4.2019].)

6.2.1 Grbl-ohjauksen käyttöönotto


Kun config.h-tiedosto on muokattu, voidaan Grbl-ohjauksen koodi ladattaa Arduino IDE -ohjelmistoa. On suositeltavaa käyttää aina uusinta versiota lataukseen.


![Grbl Panel](image)

6.2.2 Grbl-ohjauksen käyttöliittymä


Kuva 23. Grbl Panel -käyttöliittymän ulkonäkö (Github [Viitattu 1.5.2019]).

6.3 Beckhoff TwinCAT 3

Beckhoffin valmistama TwinCat 3 -ohjelmisto on tietokonepohjainen reaaliajassa toimiva koneautomaatio-ohjausjärjestelmä. Reaaliajassa toimiva ohjaus tarkoittaa tässä tapauksessa, että ohjelmoinnissa suoritettavia tehtäviä tehtäviä käsitellään järjestelmällisesti ohjelmointilohko kerrallaan. Ohjelmakoodin kiertoaika voi olla jopa 50 us, eli ohjelmakoodi luetaan ja käsitellään uudestaan 50 us:n välein. Kiertoajan pituus
vaikuttaa tietokoneen prosessorin kuormaan, aikaa suurentamalla prosessorikuorma pienenee. Tämä asia on hyvä muistaa, jos tietokoneessa on vain rajallisesti tehoa. TwinCat 3 tukee myös useamman ytimen käyttöä, joten eri tehtäviä voidaan jakaa ytimien kesken. TwinCat 3 noudattaa edellisen versionsa lailla IEC 61131-3 -standardia ja ohjelmistolla on PLC Open Motion Control hyväksyntä. (Beckhoff 2012.)


Vanhojen TwinCat 2 -projektien kääntö uudeksi TwinCat 3 -projektiksi onnistuu suoraan ohjelmistolla (Beckhoff 2012).

6.3.1 TC3 PLC


Akselien liikkeenohjaus suoritetaan TwinCat Motion Control -lohkossa, jota käsitellään myöhemin, mutta PLC-ohjelmakoodin puolella kuitenkin kutsutaan ja hallinnoidaan toimintalohkoja, jotka ohjaavat Motion Control -lohkossa olevia akseleita.

Opinnäytetyössä oleva PLC-lohko näyttää kuvan 24 mukaiselta.
Kuva 24. PLC-lohkon näkymä.

6.3.2 TC3 I/O

TwinCat 3 I/O-lohkossa muodostetaan yhteys koneessa oleviin sähköisiin komponentteihin. Yhteyden muodostukseen käytetään väylää. TwinCatissä on tuki seuraaville väylille.

- EtherCat
- Lightbus
- Profibus Dp
- Interbus
I/O toimii rajapintana sähköisten komponenttien ja PLC-ohjelmakoodin välillä.


Kuva 25. I/O-lohkon näkymä.
TwinCat I/O-lohkossa on mahdollista seurata sähköisten komponenttien online-tilaa ja tätä kautta on mahdollista pakottaa tuloja/lähtöjä päälle. Tämä toiminto voi olla tarpeellinen, kun laitteistoa halutaan testata.

6.3.3 TC3 Motion control

Liikkeenohjaus voidaan toteuttaa eri tavalla, mutta tässä työssä on käsitelty vain työssä tarvitsemia toimintoja, joita ovat NC PTP ja NC I.

![Kuva 26. Motion-lohkon näkymä.](image)

**NC PTP** (point to point) -toiminto on yksinkertaisin liikkeen ohjaus. Akselien liike voidaan suorittaa PLC Open Motion Control -lohkojen avulla PLC-ohjelmakoodissa. Liike tapahtuu alkipisteestä loppupisteeseen, mutta liikerataa näiden pisteiden välillä ei valvota. (Beckhoff [Viitattu 13.4.2019].)

**NC I** (interpolated motion) -toiminnolla voidaan ohjata useampaa akselia yhdennäkaisesti ja synkroniassa toisiinsa nähden. Tämä liikkeenohjaus on välttämätöntä, jos tarkoituksena on muodostaa koneella esimerkiksi pyöreitä tai lineaarisia liikeratoja. Akselien interpolointi ohjelmoidaan päätason avulla PLC-ohjelmakoodissa NC-koodilla, jossa interpolaatorti kääntää liikkeen ohjauksen komennoksi. Ohjelmointia tehdään PLC-ohjelmakoodin kautta, haluttuja toimintoja kutsutaan silloin, kun niitä tarvitaan. (Beckhoff [Viitattu 14.4.2019].)

NC I -interpolaattoria käytetään työssä automaattiseen ajoon. Interpolaattoriin syötetään NC-koodia, jonka komentoja koneen akselit seuraavat. PLC-koodissa olevassa FB_AutoModeSequence-aliohjelmassa suoritetaan akselien interpoloinnin vaatimat toimenpiteet. Tässä aliohjelmassa yhdistetään ajettavat akselit, luetaan mahdollinen kappalekoordinaatisto, ladataan NC-ohjelma interpolaattoriin ja suoritetaan NC-ohjelma.

G54-kappalekoordinaatistoa ja M-koodejä joudutaan käsittelemään erikseen PLC-koodin puolella. TwinCat 3 -interpolaattori ymmärtää suoraan sinne lähettetyt G-koodit, mutta M-koodit joudutaan käsittelemään itse.

Työssä kappalekoordinaatiston hallinta toimii niin, että akselit ajetaan työstettävän kappaleen origo-pisteeseen käsinaajolla ja käytöliittymästä painetaan G54-kappalekoordinaatiston asetuspainiketta. Tämä toiminta kirjoittaa konekoordinaattitoimikkean muistiin ja tämä luetaan FB_AutoModeSequence-aliohjelmassa, ennen kuin automaattiohjelmakoodi aloitetaan ajamaan. G54-kappalekoordinaatiston hallinta tapahtuu FB_SetWorkCoordinates-aliohjelmassa.

M-koodien käsitteily tapahtuu myös FB_AutoModeSequence-aliohjelmassa. M-koodien käsitteily suoritetaan valmiilla ItpGetHskMFunc-toimilohkolla, joka poimii NC-koodista M-koodin tunnuksen ja tätä vertailemalla pystytään tekemään haluttuja tiemotoja esimerkiksi M3-komennon tullessa käynnistetään jyräimen karamoottori myötäpäivään, M4-komennolla vastapäivään ja M5-komennolla pysäytetään karamoottori.
6.3.4 TC3-koneen käyttöliittymä

Kuva 27. TwinCat 3 visualization -käyttöliittymä.


Kone voidaan vaihtaa käskikäyttötilaan JogMode-painikkeella ja tämän jälkeen voidaan valita liikuteltava akseli kokea koordinaatiston vasemalta puoleltta. NC-ohjelmien ajo toimii, kun automaattitila on valittu AutoMode-painikkeella. Ajettava NC-ohjelma tulee siirtää C\TwinCat\Mc\Nci-kansioon. Tämä ohjelman nimi pitää kirjoittaa Program Name -kenttään. RefMode-painikkeella saadaan kone referenssitilaan,
49

jossa koneen akselit voidaan ajaa hallitusti opetusrajoille ja konekoordinaatti kirjoitetaan uudestaan nollaksi.


6.4 IndraDrive CS -vahvistimien ja -servomoottorien käyttöönotto


Kun yhteys on muodostettu vahvistimeen, pitää sieltä tarkastaa muutamia asioita. Vahvistimen Signal Control Word -asetuksissa määritellään, minkälaisia komentoja

Kuva 29. Kommunikointi vahvistimien ja ohjauksen välillä.

Esimerkkinä PLC-ohjelmakoodissa oleva Signal Control Word: Bit 0 käynnistää vahvistimen puolella C0500-komennon, joka nollaa vahvistimessa olevat hälytykset. Tätä nollauskomentoa voidaan nyt ohjata PLC-ohjelmakoodin puolelta lähetämällä Bit 0 -signaali vahvistimelle.

Vahvistimen Signal Status Word -asetuksia ei ole muokattu, mutta tätä kautta vahvistimilta lähtee TwinCat-ohjauksen puolelle esimerkiksi akselien paikatieto.

Akselien mekaniikan ja skaalauksen säädöt pitää myös määritellä. Axis mechanics / Scaling -asetuksissa on määritelty, että akseli on lineaarisesti liikkuva ja paikkatieto
ilmoitetaan absoluuttisesti. Akselin maksimiliikealue on 1100 mm. Mootorin ja kuu- 
laruuvin välityssuhde on 5:1, kuularuuvin nousu on 40 mm/kierros. Nämä arvot vaa-
ditaan, että moottorissa olevan pulssianturin tieto voidaan käsittellä oikein ja akselin 
liike vastaa todellista liikettä. Työn asetukset näkyvät kuvassa 30.

Kuva 30. Akselien mekaniikan ja skaalauksen asetukset.

Axis control setting -asetuksissa säädetään moottorin säätöpiiriä. Säätöpiiri-asetu-
set vaikuttavat moottorin käyttäytymiseen ja reagointiaikaan. IndraWorks-ohjelmis-
tossa on automaattinen toiminto näiden arvojen saamiseksi.
Kuva 31. Akselien säätöpiiriasetukset.

Säätöpiirin automaattisen toiminnon aloittaminen vaatii, että vahvistin laitetaan Easy Startup-tilaan. Tämän toiminnon saa aktivoitua Optimization / commissionin -asetusten alta. Easy Startup mode asettaa vahvistimen ajokuntoon testausta varten.

Kuva 32. Servovahvistimen asettaminen Easy Startup-tilaan.

Kuva 33. Akselin säätiöpiirin automaattisen määrityksen aloittaminen.

Seuraavalla välilehdellä voidaan valita, mitä arvoja halutaan testata. Kuvassa 34 kaikki testit on valittu ja akselin rakenteeseen on valittu hihnaväliys.
Kuva 34. Akselin säätöpiirin automaattisen määrityksen ensimmäinen vaihe.

Liikkeen maksimiarvoksi asetettiin 200 mm. Koneen liikkeen nopeudeksi määriteltiin 5000 mm/min ja kiihtyvyydeksi 1000 mm/s². Tämän jälkeen Start otimization -painikkeella kone aloittaa 200 mm:n suuruisen liikkeen edestakaisin ja lisäksi automaattisesti määrittelee akselin säätöpiiriarvot, jotka tulevat heti voimaan Finish-painikkeella.
Kuva 35. Akselin säätöpiirin automaattisen määrityksen toinen vaihe.

Kuva 36. Akselin säätöpiirin automaattisen määrityksen kolmas ja viimeinen vaihe.
7 POHDINTA

Opinnäytetyön tuloksena syntyi toimiva CNC-tyysin, jonka tarkkuus ylitti omat vaatimukset. Koneella on mahdollista myös koneistaa terää, mutta rungossa oli havaittavissa värinää. Rungon täyttäminen esimerkiksi hiekalla voisi vaimentaa teräksen koneistamisesta johtuvia värinöitä, koska rungon massaa saisi näin kasvatettua.

Grbl-ohjaus osoittautui erittäin toimintakykyiseksi ja tämän käyttöönotto oli helppoa. Ohjauksesta oli kytkenesimerkkejä ja ohjeistuksia hyvin saatavilla. Tämä oli ensimmäinen kerta, kun tämän työn tekijä käytti askelmoottoreita liikseenohjaukseen ja ainoana huonona puolena tuli askelmoottoreista tulevaa värinää. Värinää on kuitenkin mahdollista pienentää muuttamalla askelmoottoriohjaimen mikroaske


Projekti oli kokonaisuudessaan erittäin opettavainen ja sain haluamaani kokemusta koneensuunnittelusta. Autodesk Fusion 360 -ohjelmiston käytön suoritin itseopiskeluna ja tämän ohjelmiston CAM-ominaisuus tulee tarpeeseen tulevia koneistusprojekteja ajatellen.
LÄHTEET


LIITTEET

Liite 1. Keskeisimmät sähkökuvat TwinCat 3 ohjaukseen

Liite 2. Piirikaaviokuva askelmoottoreista ja askelmoottoriohjaimista

Liite 3. CNC-jyrsimen TwinCat 3 MAIN-ohjelmakoodi

Liite 4. CNC-jyrsimen TwinCat 3 FB_SetWorkCoordinates-ohjelmakoodi

Liite 5. CNC-jyrsimen TwinCat 3 FB_AutoModeSequence-ohjelmakoodi
Liite 1. Keskeisimmät sähkökuvat TwinCat 3 ohjaukseen
Liite 2. Piirikaaviokuva askelmoottoreista ja askelmoottoriohjaimista
PROGRAM MAIN

VAR

   fbEnableAxes :FB_AxisEnable;
   fbModeSelection :FB_ModeSelection;
   fbRefModeSequence :FB_RefModeSequence;
   fbJogModeSequence :FB_JogModeSequence;
   fbAutoModeSequence :FB_AutoModeSequence;
   fbSetWorkCoordinates :FB_SetWorkCoordinates;
   fbReset :FB_Reset;
   fbSpindleControl :FB_SpindleControl;

   bRefModeBusy :BOOL;
   bJogModeBusy :BOOL;
   bAutoModeBusy :BOOL;

END_VAR

// CNCmill V1.1
//==========================================

// Axis reset
fbReset();

// Enable all axis if Enable axis switch is ON and safety is OK
fbEnableAxes(bEnableAxes := i_bEnableAxesSW AND i_bSafetyOK);

// Auto, Jog or Ref Mode selection
fbModeSelection();

// Reference mode sequence, this machine has absolute encoders so referencing is not used
// but FB is still checking that absolut encoders are in reference
fbRefModeSequence(
   bExecute := i_bCycleStartPB AND bRefMode AND i_bSafetyOK AND NOT bMachineStateError,
   bBusy=> bRefModeBusy,
   bError=> bRefModeError,
   bDone=> );

// Jog Mode sequence
fbJogModeSequence( 
   bExecute := bJogMode AND i_bSafetyOK, //AND NOT bMachineStateError,
   bBusy=> bJogModeBusy,
   bError=> bJogModeError,
   bDone=> );

// Sets and calculate work coordinates
fbSetWorkCoordinates();

// Setting interpolation feedrate
ItpSetOverridePercent(fOverridePercent := fUserOverride, sPlcToNci:=out_stPlcToItp);

// Auto Mode sequence. 3 axis interpolation.
fbAutoModeSequence(
   bExecute := i_bCycleStartPB AND bAutoMode AND i_bSafetyOK AND bAllAxesReady AND NOT bMachineStateError,
   bBusy=> bAutoModeBusy,
   bError=> bAutoModeError,
   bDone=> bNcProgramDone);

// Spindle motor control
fbSpindleControl();

// Machine State info
bMachineStateBusy := bRefModeBusy OR bJogModeBusy OR bAutoModeBusy;
bMachineStateError := bRefModeError OR bJogModeError OR bAutoModeError;
FUNCTION_BLOCK FB_SetWorkCoordinates

VAR_INPUT
END_VAR

VAR_OUTPUT
END_VAR

VAR

fbReadG54offset : ItpReadZeroShiftEx;
fbResetG54offset : ItpSetZeroShiftNullEx;
END_VAR

// Write G54 X offset to ItpZeroShiftEx function block
IF bWriteXorigin THEN
  fShiftX := Xaxis.NcToPlc.ActPos;
  out_G54ZeroShift.fShiftX := fShiftX;
END_IF

// Write G54 Y offset to ItpZeroShiftEx function block
IF bWriteYorigin THEN
  fShiftY := Yaxis.NcToPlc.ActPos;
  out_G54ZeroShift.fShiftY := fShiftY;
END_IF

// Write G54 Z offset to ItpZeroShiftEx function block
IF bWriteZorigin THEN
  fShiftZ := Zaxis.NcToPlc.ActPos;
  out_G54ZeroShift.fShiftZ := fShiftZ;
END_IF

// Resets G54 offset values from interpreter
fbResetG54offset(
  bExecute := bResetG54offset,
  tTimeOut := ,
  sNciToPlc := in_stItpToPlc,
  bBusy := ,
  nErrId := );

// Reads G54 Offset values
fbReadG54offset(
  bExecute := bReadG54offset,
  nZsNo := 54,
  tTimeOut := ,
  sNciToPlc := in_stItpToPlc,
  sZeroShiftDesc := out_G54ZeroShift,
  bBusy := ,
  bErr := ,
  nErrId := );

// Calculate Work coordinates
XWorkCoordinate := Xaxis.NcToPlc.ActPos - fShiftX;
YWorkCoordinate := Yaxis.NcToPlc.ActPos - fShiftY;
ZWorkCoordinate := Zaxis.NcToPlc.ActPos - fShiftZ;
FUNCTION_BLOCK FB_AutoModeSequence

VAR_INPUT
  bExecute : BOOL;
END_VAR

VAR_OUTPUT
  bBusy : BOOL;
  bError : BOOL;
  bDone : BOOL;
END_VAR

VAR_IN_OUT
  END_VAR

VAR
  nState : UDINT := 0; // Auto cycle current state
  nErrorState : UDINT; // Error state number
  fbWriteG54offset : ItpWriteZeroShiftEx; // G54 offsets to interpreter
  fbBuildGroup : CfgBuildExt3DGroup; // Build 3D interpolation group
  fbLoadGCode : ItpLoadProgEx; // Load G-code to interpreter
  fbStartGCode : ItpStartStopEx; // Execute G-code program
  fbClearGroup : CfgReconfigGroup; // Clear 3D interpolation group
  fbConfirmHsk : ItpConfirmHsk; // Execute M-function
  fbStop : RS;
  fUserOverrideOld : LREAL;
  nInterpreterState : UDINT := 0; // Interpreter state
  bConfirmHsk : BOOL := FALSE; // Confirm current M-function
  sPrgName : STRING(255); // Program file name
END_VAR

// Reset State machine
IF i_bResetPB THEN
  bBusy := FALSE;
  bDone := FALSE;
  bError := FALSE;
  nState := 0;
  nErrorState := nState;
END_IF

// If CycleStop is ON set feedrate to 0
fbStop(
  SET:= i_bCycleStopPB,
  RESET:= i_bCycleStartPB OR i_bResetPB,
  Q1=> bCycleStop);
IF bCycleStop THEN
  fUserOverride := fUserOverrideOld;
  fUserOverride := 0;
  MsgMachineState := 'Machine Stopped';
  ELSE
  fUserOverrideOld := fUserOverride;
END_IF

// Automatic program cycle
// First step is to set machine state to busy
CASE nState OF
  0:
    IF bExecute THEN
      bBusy := TRUE;
      bDone := FALSE;
      bError := FALSE;
      nErrorState := nState;
      MsgMachineState := 'Start the program';
      nState := 10;
    END_IF
END_VAR
// Build 3D interpolation group
fbBuildGroup(
  bExecute:=TRUE,
  nGroupId:=ItpGetGroupId(sNciToPlc:=in_stItpToPlc),
  nXAxisId:=Xaxis.NcToPlc.AxisId,
  nYAxisId:=Yaxis.NcToPlc.AxisId,
  nZAxisId:=Zaxis.NcToPlc.AxisId,
  tTimeOut:= ,
  bBusy=> ,
  bErr=> ,
  nErrId=> );
IF NOT fbBuildGroup.bBusy THEN
  IF NOT fbBuildGroup.bErr THEN
    nState := 15;
  ELSE
    (* error handling *)
    nErrorState := nState;
    bBusy := FALSE;
    bError := TRUE;
    MsgAlarm := 'Error when building interpolation group';
    nState := 9999;
  END_IF
  fbBuildGroup(bExecute:=FALSE);
END_IF

15:

// Write G54 offsets to interpreter
fbWriteG54offset(
  bExecute:= TRUE,
  nZsNo:= 54,
  tTimeOut:= ,
  sNciToPlc:= in_stItpToPlc,
  sZeroShiftDesc:= out_G54ZeroShift,
  bBusy => ,
  bErr => ,
  nErrId => );
IF NOT fbWriteG54offset.bBusy THEN
  IF NOT fbWriteG54offset.bErr THEN
    nState := 20;
  ELSE
    (* error handling *)
    nErrorState := nState;
    bBusy := FALSE;
    bError := TRUE;
    MsgAlarm := 'Error when writing G54 offsets';
    nState := 9999;
  END_IF
  fbWriteG54offset(bExecute:=FALSE,
    sNciToPlc:=in_stItpToPlc,
    sZeroShiftDesc:= out_G54ZeroShift,);
END_IF

20:

// Load g-code file
fbLoadGCode(
  sNciToPlc:=in_stItpToPlc,
  bExecute:=TRUE,
  sPrg:= sPrgName,
  nLength:= INT_TO_UDINT(LEN(sPrgName)),
  tTimeOut:= ,
  bBusy=> ,
  bErr=> ,
  nErrId=> );
IF NOT fbLoadGCode.bBusy THEN
  IF NOT fbLoadGCode.bErr THEN
    nState := 30;
  ELSE
    (* error handling *)
    nErrorState := nState;
    bBusy := FALSE;
    bError := TRUE;
    MsgAlarm := 'Error when loading g-code file';
    nState := 9999;
  END_IF
  fbLoadGCode(sNciToPlc:=in_stItpToPlc, bExecute:=FALSE);
END_IF
// Check the status of the interpreter. It has to be in ready state, in case of loading the program successfully.

nInterpreterState := ItpGetStateInterpreter(sNciToPlc:=in_stItpToPlc);

IF nInterpreterState = Tc2_NCI.NCI_INTERPRETER_READY THEN
  nState := 40;
ELSE
  (* error handling *)
  nErrorState := nState;
  bBusy := FALSE;
  bError := TRUE;
  MsgAlarm := 'Interpreter is not ready';
  nState := 9999;
END_IF

40:

// Start g-code file

fbStartGCode(
bStart:=TRUE,
bStop:=FALSE,
tTimeOut:= ,
sNciToPlc:= in_stItpToPlc,
bBusy=> ,
bErr=> ,
nErrId=> );

IF NOT fbStartGCode.bBusy THEN
  IF NOT fbStartGCode.bErr THEN
    nState := 50;
  ELSE
    (* error handling *)
    nErrorState := nState;
    bBusy := FALSE;
    bError := TRUE;
    MsgAlarm := 'Error when starting program';
    nState := 9999;
  END_IF
  fbStartGCode( bStart:=FALSE, sNciToPlc:= in_stItpToPlc );
ENDIF

50:

// check state, again - we are at least not in ready state for several ticks
// this is to ensure that we don't indicate program has finished, before we have started

nInterpreterState := ItpGetStateInterpreter(sNciToPlc:=in_stItpToPlc);

IF nInterpreterState <> Tc2_NCI.NCI_INTERPRETER_READY THEN
  // Program is running
  MsgMachineState := 'Program Running';
  nState := 60;
ELSE
  nErrorState := nState;
  bBusy := FALSE;
  bError := TRUE;
  MsgAlarm := 'Program interrupted';
  nState := 9999;
END_IF

60:

nInterpreterState := ItpGetStateInterpreter(sNciToPlc:=in_stItpToPlc);

IF nInterpreterState = Tc2_NCI.NCI_INTERPRETER_READY THEN
  // program has finished
  MsgMachineState := 'Program Finished';
  nState := 70;
ELSIF nInterpreterState = Tc2_NCI.NCI_INTERPRETER_ABORTED THEN
  // a run-time error occoured - this could be a lag error or something else...
  // error handling
  nErrorState := nState;
  bBusy := FALSE;
  bError := TRUE;
  MsgAlarm := 'Program interrupted';
  nState := 9999;
ENDIF

70:

// program has finished
// clear 3D interpolation group

fbClearGroup(
bExecute:=TRUE,
 nGroupId:=ItpGetGroupId(sNciToPlc:=in_stItpToPlc) ,
 tTimeOut:= ,
 bBusy=> ,
 bErr=> ,
 nErrId=> );

IF NOT fbClearGroup.bBusy THEN
  IF NOT fbClearGroup.bErr THEN
    nState := 80;
  ELSE
    (* error handling *)
  END_IF
nErrorState := nState;
bBusy := FALSE;
bError := TRUE;
MsgAlarm := 'Error when clearing interpolation group';
nState := 9999;
END_IF
fbClearGroup(bExecute:=FALSE);
END_IF

80:
bDone := TRUE;
bBusy := FALSE;
IF NOT bExecute THEN
  nState := 0;
END_IF
9999:
  // error state
  IF NOT bExecute THEN
    nState := 0;
    bError := FALSE;
  END_IF
END_CASE

(** M-Function handling ***)
// M-functions are handled outside of state machine
// Reads M-function from G-code
IF ItpIsHskMFunc(in_stItpToPlc) AND NOT fbConfirmHsk.bBusy THEN
  ItpGetHskMFunc(in_stItpToPlc);
  // IF M3-function detected start Spindle CW
  IF in_stItpToPlc.HskMFuncNo = 3 THEN
    o_SpindleCW := TRUE;
    MsgSpindleState := 'Spindle CW';
    bConfirmHsk := TRUE;
  ELSE
    o_SpindleCW := FALSE;
  END_IF
  // If M4-function detected starts Spindle CCW
  IF in_stItpToPlc.HskMFuncNo = 4 THEN
    o_SpindleCCW := TRUE;
    MsgSpindleState := 'Spindle CCW';
    bConfirmHsk := TRUE;
  ELSE
    o_SpindleCCW := FALSE;
  END_IF
  // If M5-function detected Stops Spindle
  IF in_stItpToPlc.HskMFuncNo = 5 THEN
    o_SpindleCW := FALSE;
    o_SpindleCCW := FALSE;
    MsgSpindleState := 'Spindle Stopped';
    bConfirmHsk := TRUE;
  END_IF
ELSE
  bConfirmHsk := FALSE;
END_IF

// Execute M-function
fbConfirmHsk{
  bExecute:=bConfirmHsk ,
  sNCIToPlc:=in_stItpToPlc ,
  sPlcToNCI:=out_stPlcToItp ,
  bBusy=> ,
  bErr=> ,
  nErrId=> );

// Reading current block number
IF nInterpreterState = Tc2_NCI.NCI_INTERPRETER_RUNNING THEN
  bCurrentBlockNumber := ItpGetBlockNumber(sNCIToPlc:= in_stItpToPlc);
END_IF