



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Aaron Finnilä

3-AKSELISEN JYRSIMEN CNC-OHJAUKSEN TOTEUTUS

Tekniikka ja liikenne

2010

VAASAN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma

TIIVISTELMÄ

Tekijä	Aaron Finnilä
Opinnäytetyön nimi	3-akselisen jyrsimen CNC-ohjauksen toteutus
Vuosi	2010
Kieli	Suomi
Sivumäärä	69 + 4 liitettä
Ohjaaja	Martti Mustonen

Tämän opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa erikoisvalmisteisen kolmiakselisen jyrsinkoneen CNC-ohjaus hammaspyörävalmistuksessa syntyvän jäysteen poistoa, sekä hampaan kulman viisteytystä varten. Vaatimukset jyrsimen ohjaukselle olivat nopea kappalekohtaisen asetuksen teko sekä mahdollisuus tarpeeksi monipuolisille liikkeille.

Riittävän laadukkaan lopputuloksen varmistamiseksi jyrsimen liike toteutettiin servomoottoreiden sekä lineaarijohteiden ja pyöröpöydän avulla. Jyrsin pohjautuu XZC-tekniikalle, jossa lineaariset X- ja Z-akselit siirtävät jyrsinterää, rotationaalisen C-akselin pyörittäessä työstettävää hammaspyörää oman akselinsa ympäri.

Servomoottoreiden ohjaus, akseleiden liikkeet sekä paikoitukset olivat helppokäyttöisyyden, sekä liikkeen erikoisen luonteen vuoksi toteutettava alusta asti itse, sillä mikään CAD-ohjelmisto ei tarjoa yksinkertaista ratkaisua tässä työstössä tarvittaville liikeradoille, ilman monimutkaista ja aikaa vievää työstöratasuunnittelua. Ohjaus perustuu tätä liikettä varten räätälöityyn C#-ohjelmointikieliseen sovellukseen, jota suorittaa jyrsimen yhteyteen liitetty tietokone.

Sovellus ohjaa tietokoneen rinnakkaisportin kautta kolmea Granite Devices Oy:n valmistamaa servovahvistinta, joihin kolmen akselin servomoottorit on kytketty. Käytetty tekniikka mahdollistaa hammaspyörän viisteen erittäin tarkan sekä tehokkaan koneistamisen.

Asiasanat: servomoottori, C#, CNC, jyrsin, hammaspyörä, viiste

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Tietotekniikan koulutusohjelma

ABSTRACT

Author	Aaron Finnilä
Title	Computer Control Implementation of a 3 Axis Milling Machine
Year	2010
Language	Finnish
Pages	69 + 4 Appendices
Name of Supervisor	Martti Mustonen

The aim of this thesis was to design and implement a Computerized Numerical Control of a unique milling machine tailored to remove burr from cogwheel manufacturing products. Versatile trajectories, a rapid milling stage and the least amount of product adjustments as possible were set as the main requirements of the design.

To ensure a high quality finish, the movement of the machine was created by linear guides and a rotating assembly. The machine design is based on a XZC technique which contains linear movement axes X and Z, and a rotating axis C. The milling cutter is being moved by the axes X and Z, while the machined product is being rotated around its own axle.

The control of the servos, as well as trajectories and positioning of the three axes were built from scratch, as no CAD program offers a simple and rapid solution for this trajectory of special character. The control is based on a unique application written in C# and executed by a computer attached to the machine.

This application uses the computer's parallel port to control the three different servo amplifiers manufactured by Granite Devices Ltd. These three servo amplifiers are connected to brushless direct current motors which drive the axes. The technique used in this machine enables a high precision bevel machining as well as an effective burr removal of the manufactured cogwheel.

Keywords: Servo motors, C#, CNC, Milling Machine, Cogwheel, Bevel

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1. JOHDANTO.....	7
1.1 Yleistä.....	7
1.2 Laajuus ja sisältö.....	7
1.3 Työn tausta.....	7
1.4 Kehitysympäristö.....	8
2. HAMMASPYÖRÄVALMISTUS.....	9
2.1 Hammastus.....	9
2.2 Jäysteen muodostuminen.....	12
2.3 Muu viimeistely.....	14
2.4 Hammaspyöräteollisuuden jäysteenpoistokoneet.....	15
3. G3 JÄYSTEENPOISTOKONE.....	17
3.1 Yleistä.....	17
3.2 Koneen mekaaninen rakenne.....	18
3.2.1. Ristisyöttöpöytä.....	20
3.2.2. Pyöröpöytä.....	21
3.2.3. Kappaleen kiinnitys.....	21
3.2.4. Koneen runko.....	22
3.2.5. Jyrsinpää.....	22
3.2.6. Virtalähde.....	22
3.3 Ohjausjärjestelmän rakenne.....	24
3.3.1. Servomoottorit.....	24
3.3.2. Servovahvistimet.....	25
3.3.3. Servovahvistimien konfigurointi.....	25
3.3.4. Breakout -kortti.....	27
3.3.5. Järjestelmän kytkentä.....	28
4. G3 JÄYSTEENPOISTOKONEEN OHJAUS.....	30
4.1 Järjestelmän vaatimukset.....	30
4.2 Rinnakkaisportti ja Windows XP.....	30
4.3 Moottorinohjaussäikeet sekä käyttöliittymä.....	32

4.4	Moottoreiden nopeuden määrittäminen	32
4.5	Käyttöliittymän toiminta.....	32
4.5.1.	Liikeratojen tallentaminen	34
4.5.2.	Liikeratojen avaaminen.....	36
4.5.3.	Liikeratojen poistaminen.....	37
4.5.4.	Ajonaikainen debug-ikkuna	38
4.6	Liikeradan suunnittelu	39
4.6.1.	Terän sisäänajo.....	39
4.6.2.	Monilastuajo.....	40
4.6.3.	Monikerrosajo	42
4.6.4.	Syöttö ja pikaliike	43
4.6.5.	Päähalkaisija sekä hammasluku	43
4.7	Paikkatiedon ylläpito	44
4.7.1.	Toiminta virhetilanteessa	45
4.7.2.	Päätörajahälytys	46
4.7.3.	Työkalukorjaimet ja Offset-toiminto	46
4.7.4.	Hammasjaon säilyttäminen	49
4.7.5.	C-akselin synkronointi	50
5.	KAPPALEKOHTAISEN ASETUKSEN TEKO	52
5.1	Referenssiajo.....	52
5.2	Manuaalisesti liikkuminen.....	53
5.2.1.	Liike päätörajahälytyksen aikana.....	54
5.3	Akseleiden nollakohtien määrittäminen.....	55
5.4	Liikeradan määrittely	55
5.4.1.	Esimerkki liikeradasta.....	56
5.4.2.	Inkrementaalinen liike.....	58
5.5	Käynnistys.....	58
5.6	Pysäytys	59
5.7	Aloituskohtaan palauttaminen	59
6.	KONEEN TESTAUS	60
6.1	Virhetilanteet.....	60
6.2	Koneen mekaaniset ominaisuudet.....	60

6.3	Vahvistimien konfigurointi.....	60
6.4	Sovelluksen toiminta Windows-ympäristössä.....	61
6.5	Jännitelähteen toimivuus.....	61
7.	YHTEENVETO JA JOHTOPÄÄTÖKSET	62
7.1	Yhteenveto	62
7.2	Johtopäätökset.....	62
7.3	Jatkotoimenpiteet sekä kehitysmahdollisuudet.....	63
7.3.1.	Vinohammaspyörien mittaus	64
7.3.2.	Karamoottorin automaattinen käynnistäminen	65

LÄHTEET

LIITELUETTELO

KÄYTETYT TERMIT JA LYHENTEET

LIITTEET

1. JOHDANTO

1.1 Yleistä

Hammaspyörän koneistuksessa hampaan nurkkaan muodostuva jäyste on ongelma, johon pieniin ja keskisuuriin sarjoihin keskittyneen yrityksen on hankala löytää ratkaisua. Käsien suoritettava jäysteenpoisto on hidasta, eikä lopputulos ole esteettisesti paras mahdollinen. Hammastettu tuote toimitetaan oletusarvoisesti jäysteettömänä ja siksi hammaspyörävalmistajan on tuotteen hintaa laskiessaan hankala sisällyttää jäysteenpoistoon kuluva aika valmistuskustannuksiin.

Markkinoiden mekaanisesti jäysettä työstävät koneet ovat pääasiassa manuaalisella kuormituksella ja ilman kehittyntä ohjausta olevia koneita, jotka tasaisen laadun varmistamiseksi vaativat monesti jonkinasteista käsien suoritettavaa esipuhdistusta. Jäysteenpoisto voidaan myös suorittaa CNC-työstökeskuksella, mutta tämä toteutus vaatii suuria investointeja, monimutkaisten liikeratojen suunnittelua CAM-ohjelmalla sekä aikaa vievää asetuksen tekoa. Jäysteenpoisto on viimeistelevä työvaihe jonka tulee tapahtua nopeasti ja pienin kustannuksin, ja siksi työstävän koneen ohjauksen sekä käyttöliittymän tulee olla suunniteltu juuri tätä tarkoitusta varten.

1.2 Laajuus ja sisältö

Työ keskittyy jyrsinkoneen liikettä ohjaavan PC-sovelluksen toteutukseen. Koneen mekaaninen osuus sekä moottoreiden ohjaukseen tarvittava teholähde on rakennettu opinnäytetyön ulkopuolella, testausta varten. Tulevissa luvuissa käydään kuitenkin selvyuden vuoksi kokonaisvaltaisesti läpi koneen toimintaa. Moottoreiden ohjaus sekä takaisinkytkentä on toteutettu Granite Device Oy:n valmistamilla ja toimittamilla servovahvistimilla, sekä harjattomilla tasavirtamoottoreilla.

1.3 Työn tausta

Työn pohjana on pienyrityksen tarve tehokkaalle ja helppokäyttöiselle ratkaisulle hammaspyörän valmistuksessa esiintyvässä jäysteongelmassa. Ilman automaattijyrsintä jäyste on poistettava käsien joko kulmahiomakoneella leikkaamalla, tai

nauhahiomakoneella hiomalla. Painavien sekä ison modulin hammastuksella olevien kappaleiden käsin hiominen on aikaa vievää, haastavaa sekä työturvallisuuden kannalta riskialtista. Hammaspyörävalmistuksessa tuotteiden kirjo on usein laaja, joten automaattisen jyrsimen tulee olla riittävän monipuolinen ja yleiskäyttöinen, jotta sitä voidaan soveltaa eri tuotteisiin. Näin laitteen mahdollisuudet saadaan hyödynnettyä ongelmanratkaisussa, mahdollisimman pienillä kustannuksilla.

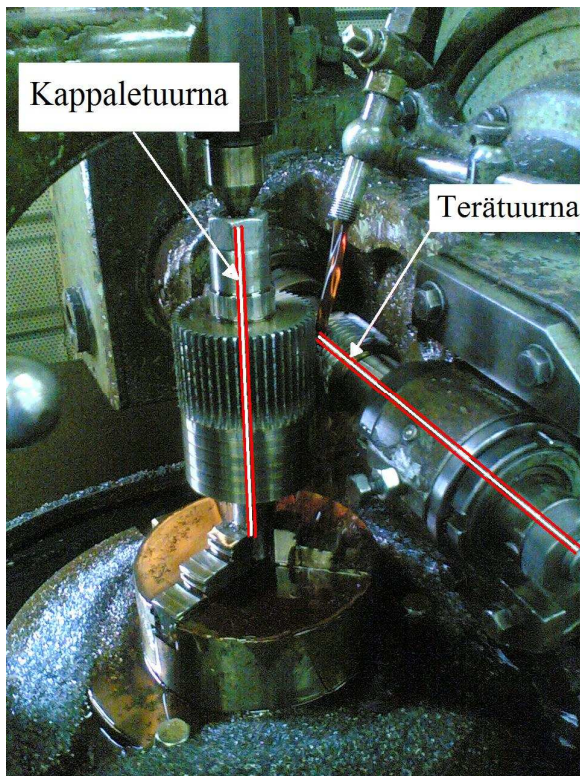
1.4 Kehitysympäristö

Sovellus on kehitetty .Net C# -kielelle, lähinnä tämän kielen käyttöliittymäsuunnittelun helppouden takia. Lähdekoodi on kirjoitettu pääasiassa Microsoft Visual C# Studio 2008 Express Edition -kehitysohjelmalla, joka on Microsoftin sivustolta ladattava ilmaisversio Visual Studio 2008:sta. Yhteensopivuuden takia sovelluksessa päätettiin käyttää MS SQL -tyyppistä tietokantaa liikeratojen tallennukseen.

2. HAMMASPYÖRÄVALMISTUS

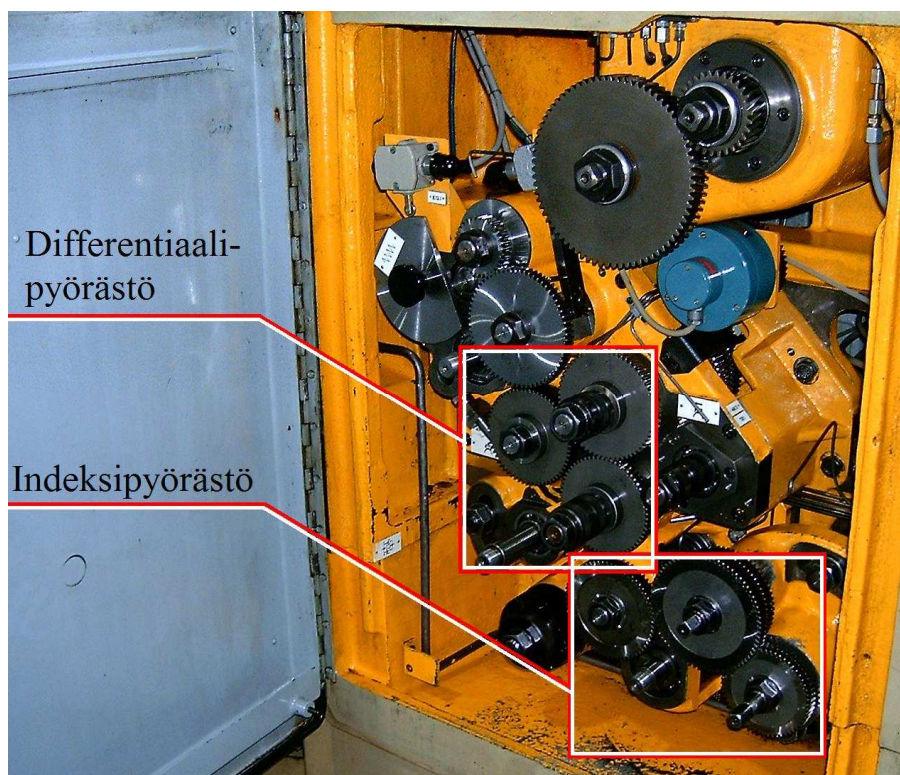
2.1 Hammastus

Hammastuspyörien massavalmistusta tapahtuu vierintäjyrsinkoneella, jossa tietyn modulin tai hammasjaon terä, pyöriessään kappaleen hammasluvusta riippuvalla nopeudella, leikkaa oman akselinsa ympäri pyörivästä kappaleesta materiaalia (kuva 1). Vierintäjyrsinkoneella sekä vierintäterällä (kuva 4) hampaan profiili muodostuu ympyräevolventtikäyrän (kuva 3) mukaisesti riippumatta kappaleen hammasluvusta tai halkaisijasta, toisin kuin esimerkiksi kiekkoterällä jyrsittäessä jolloin eri hammasluvuilla tulee käyttää eri muotoihin hiottuja teriä (Maaranen 2004: 218).



Kuva 1. Vierintäjyrsinkoneen rakenne.

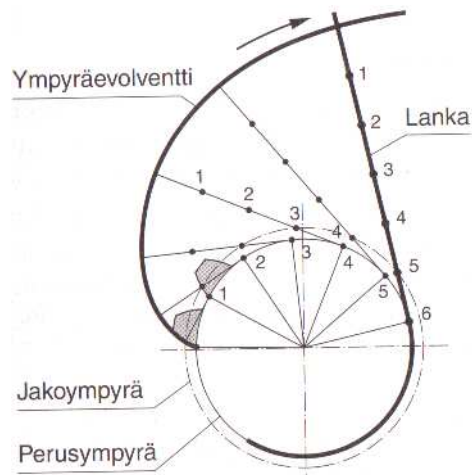
Vierintäjyrsinkoneen kierukkamainen vierintäterä mahdollistaa kappaleen ympäri-työstämisen yhdellä kiinnityksellä, jolloin kiekkoteräjyrsinnässä pakolliselle jakolaitteelle ei ole tarvetta (Maaranen 2004: 220). Kappaleen hammasluvusta johtuvan kappaleen sekä vierintäterän välisen pyörintänopeuden suhde asetetaan normaalisti mekaanisen vaihdelaatikon indeksipyörästön välityksiä muuttamalla (kuva 2). Tämä tekniikka mahdollistaa nopean sekä tarkan työstön vähemmällä työpanoksella, eikä välttämätöntä tarvetta kehittyneelle ohjausjärjestelmälle ole.



Kuva 2. Vierintäjyrsinkoneen vaihdelaatikon rakenne.

Vierintäjyrsinkoneella voidaan valmistaa myös vinohampaisia hammaspyöriä, joiden tarkka työstäminen kiekkoterällä on mahdotonta ilman CNC-ohjattua 4-akselista työstökeskusta, tai manuaalisen jyrsinkoneen syöttökoneistoon liitettävää erityistä pyörityspöytää (Maaranen 2004: 232). Tämä kuitenkin edellyttää että vierintäjyrsinkone on varustettu differentiaalipyörästöllä (kuva 2), jonka välityssuh-

teet tulee laskea ja asettaa samoin kuin kappaleen hammaslukuun tarvittava välityskin, jotta haluttu hampaan vinouskulma voidaan jyrsiä.



Kuva 3. Hampaan profiilin muodostuminen ympyräevolventtikäyrästä.

(Maaranen 2004: 215)

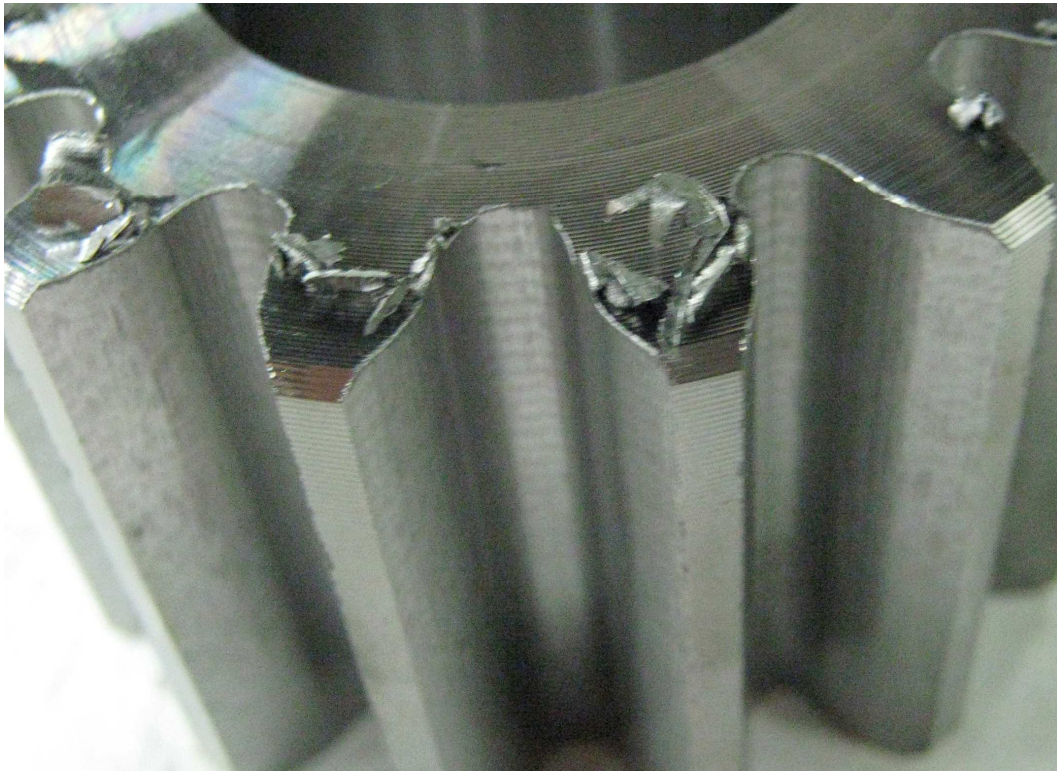
Differentiaalipyörästön välityssuhteiden laskeminen on hankalaa, koska välityspyörät tulee käytännössä selvittää iteroimalla. Konevalmistajat toimittavat monesti koneen mukana valmiin välitystaulukon eri vinouskulmista, mutta normaalisti taulukot koostuvat vain tasalukuisista asteista, joita käytännön hammaspyörävalmistuksessa harvoin käytetään. Välityssuhteiden laskentaan on olemassa tietokoneohjelmia, jotka perustuvat samaan iteroimalla parhaan lopputuloksen selvittämiseen kuin käsin laskentakin.



Kuva 4. Erilaisia vierintäteriä.

2.2 Jäysteen muodostuminen

Vierintäjyrsinkoneen tyypillinen terämateriaali on karkaistu, ja usein myös pinnoitettu pikateräs. Tästä johtuen leikkuunopeudet ovat huomattavasti pienemmät kuin kovametalliterällä työstettäessä (Aaltonen, Andersson & Kauppinen 1997: 36). Terän monimutkainen muoto aiheuttaa myös sen, ettei lastunmurtajaa tai rintakulmaa voida hioa optimaalisten leikkausominaisuuksien takaamiseksi, vaan terän leikaava pinta teroitetaan suorakulmaan (Maaranen 2004: 217). Matala leikkuunopeus, sekä epäedullinen terän muoto aiheuttavat kovametallijyrsintään verrattuna suuremman jäysteen kappaleen päätyyn, jossa terä irtoaa kappaleesta (kuva 5). Jäysteen muodostuminen johtuu ilmiöstä, jossa kappaleen päädyssä terä ei kykene työstettävän materiaalin pehmeiden vuoksi irrottamaan materiaalia lastuina, vaan materiaali kääntyy terän edellä kappaleen taakse jääden osittain kappaleeseen kiinni. Tylsällä vierintäterällä työstettäessä ilmiö korostuu entisestään (kuva 6).



Kuva 5. Hammaspyörän kulmaan muodostuva jäyste.

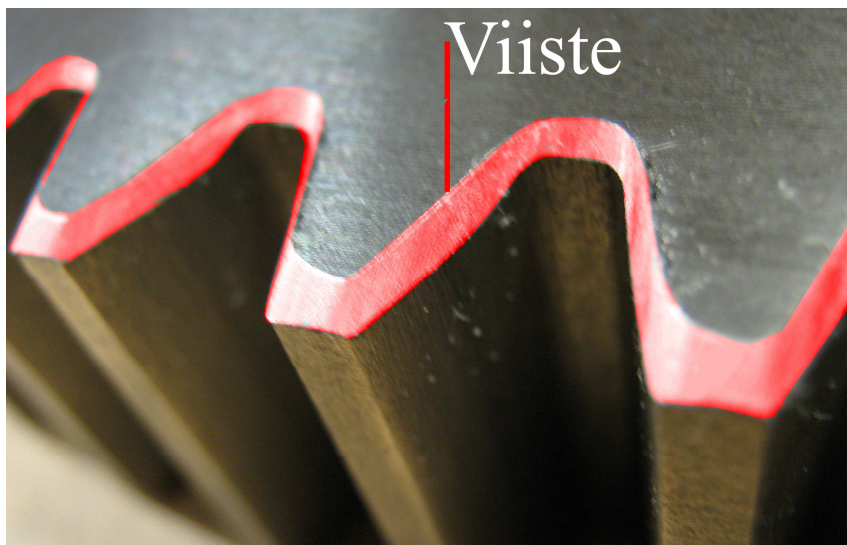
Riippuen hammaspyörän rakenteesta, kappaleita voidaan kuitenkin pinota vierintäjäyrsinkoneen koon puitteissa kappaleakselille nippuun (kuva 1), jolloin jäyste muodostuu vain alimman kappaleen nurkkaan. Jäysteen koko ja paksuus vaihtelevat merkittävästi eri kappalemateriaalien, syöttönopeuksien, terien tylsyyssasteiden, kuin leikkuunesteenä käytetyn öljyn koostumuksienkin välillä. Helposti kääntyvä jäyste saattaa myös olla orientoitunut hampaan pätyyn, väliin, tai osittain molempiin suuntiin, aiheuttaen suuria ongelmia manuaalisille jäysteenpoistokoneille, joissa viimeistelevän terän positiota ei ole servomoottorein ohjattu.



Kuva 6. Sitkeään materiaaliin muodostunut jäyste.

2.3 Muu viimeistely

Siirryttäessä suuremmille hammasjaoille vaaditaan monesti hampaan nurkan olevan jäysteettömyyden lisäksi myös viistetty (kuva 7). Tämä vaaditaan monesti esteettisistä sekä työturvallisuussyistä. Manuaalisilla jäysteenpoistokoneilla tämän viisteen työstäminen on epätarkkaa, lisäksi mahdollisen ei-toivotun viisteen välttäminen on hankalaa. Kuitenkin mikäli viiste kappaleessa sallitaan, on jäyste helpompi poistaa työstämällä kappaleeseen hienoinen viiste, kuin yrittää toteuttaa jäysteenpoistoa ilman pienintäkään viistettä. Työstötekniikan kannalta tilanne on molemmissa tapauksissa sama, vain terän kuormaa, tai positiota kappaleen keskipisteestä, muutetaan.



Kuva 7. Hampaan kulmaan työstetty viiste.

Mikäli hammaspyörä sijoitetaan kohteeseen, jossa voimansiirto tullaan kytkemään aksiaalisen liikkeen avulla, (muuttuvavälityksiset vaihdelaatikot, starttikehät), vaaditaan hammaspyörän kulmaan monesti joko yksi- tai molemminpuolinen, synkronoimatonta kytkeytymistä helpottava suurempi viiste. Tämä välityspyörien kytkeytymistä helpottava viiste saattaa olla 35...55 asteen kulmassa hammaspyörän päätypintaan nähden, sekä luokkaa $0,2...1,5$ * hammaspyörän moduli, riippuen suuresti käyttökohteesta.

Tällaisen viisteen työstämiseen tarvitaan käytännössä vähintään joko CNC-ohjattu 4-akselinen työstökeskus, tai vierintäjyrsinkone erikoishiotulla terällä. Opinnäytetyön tuloksena valmistunut kone vastaa myös tähän koneistuksen ongelmaan, sillä liikeradat voidaan suunnitella siten, että jyrsin muodostaa viisteen vain hampaan toiseen laitaan. Mikäli haluttu viiste ylittää suuruudellaan koneen fyysisten ominaisuuksien asettavat rajat, voidaan työstö suorittaa tätä konetta varten suunnitellulla monilastuajolla.

2.4 Hammaspyöräteollisuuden jäysteenpoistokoneet

Markkinoilla on tarjolla erilaisia laitteita hammastuksen jäysteongelmaan. Laitteet ovat kuitenkin monesti kompromisseja hyvän lopputuloksen ja alhaisen hinnan

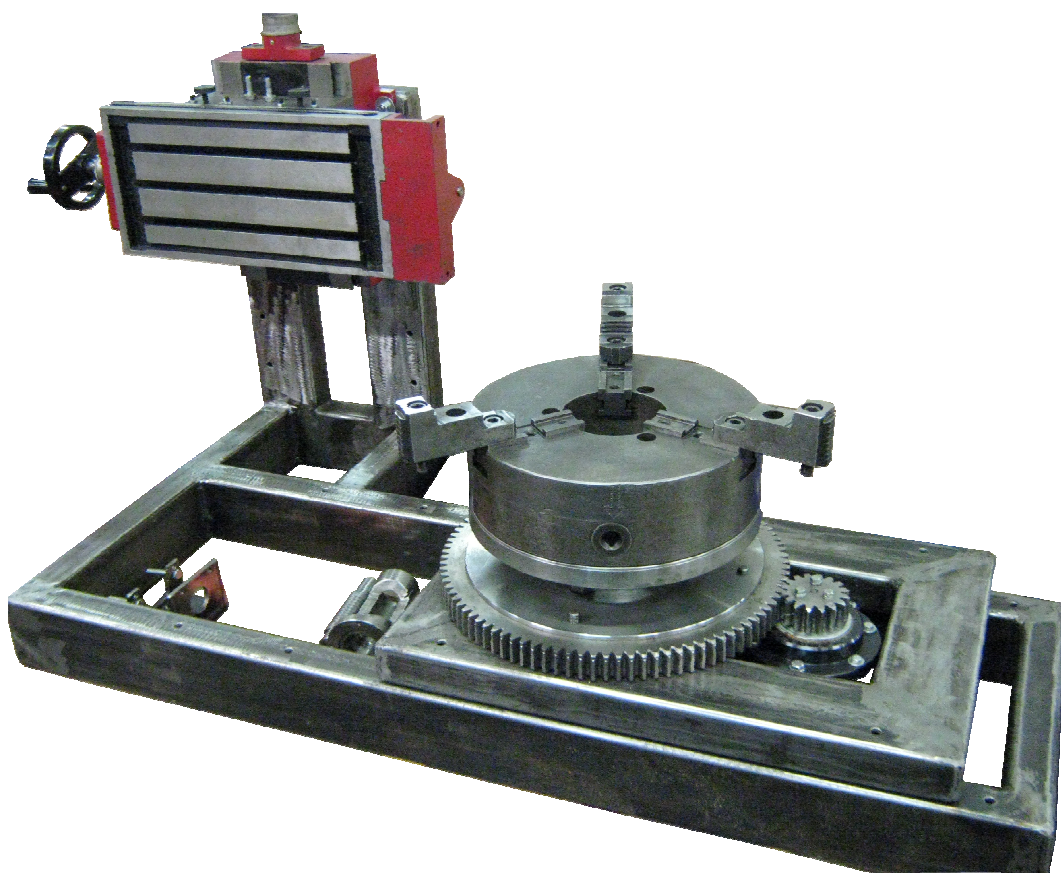
välillä, sillä ne on toteutettu mekaanisella, pneumaattisella tai hydraulisella terän kuormituksella, joka on koneen valmistuskustannusten kannalta hyvä ratkaisu, mutta tasalaatuisen lopputuloksen saaminen on hankalaa ja vaatii kappalekohtaista säätöä.

Tällaisiin tekniikkoihin perustuvilla koneilla kappaletta pyöritetään kierros ympäri, leikkaavan tai hiovan terän painaessa vakiokuormalla hammaspyörän nurkkaa vasten. Symmetrisen viisteen aikaansaamiseksi kappaletta täytyy monesti pyörittää kaksi kierrosta, yksi kierros molempiin suuntiin. Tämä mekaaninen tekniikka on myös erittäin vaikutusaltis jäysteen muodon, suunnan ja paksuuden muutoksille, ja siksi useisiin koneisiin on kehitelty erillinen harjaustoiminto, jossa ennen työstöä hampaan välissä olevat jäysteet nostetaan teräsharjalaikan avulla osoittamaan samaan suuntaan. Tällöin kappaleen tulee pyörähtää vieläkin useampia kierroksia, joihin kuluva aika tekee työstötekniikasta kannattamatonta.

Käytettäessä servomoottorein ohjattua jyrsintä voidaan terän liikettä ohjata ennalta määrätyn liikeradan perusteella sekä hammastuksen mukaisesti, eikä jäysteen laatu vaikuta lopputulokseen, edellyttäen että servomoottoreiden vääntö on riittävä jyrsinterän haluttuun positioon siirtämiseen.

3. G3 JÄYSTEENPOISTOKONE

Tämän työn tuloksena on toimiva CNC-ohjaus sekä käyttöliittymä uudelleenlaiselle G3-jäysteenpoistojyrsimelle (kuva 8). Ohjaus ei perustu millekään valmiille pohjalle, eli työhön sisältyi niin käyttöliittymän suunnittelu, servovahvistimien kanssa keskusteleminen, kuin liikeratojen toteuttaminenkin. Itse jyrsimen mekaaninen suunnittelu sekä toteutus suoritettiin Kaarlo Finnilä Oy:n konepajajäsenillä. Rungon suunnittelu aloitettiin jo ennen kuin ohjauksen toteutustapa oli varmistettu, joten ohjaus voitiin räätälöidä melkein valmiin koneen käyttöön.

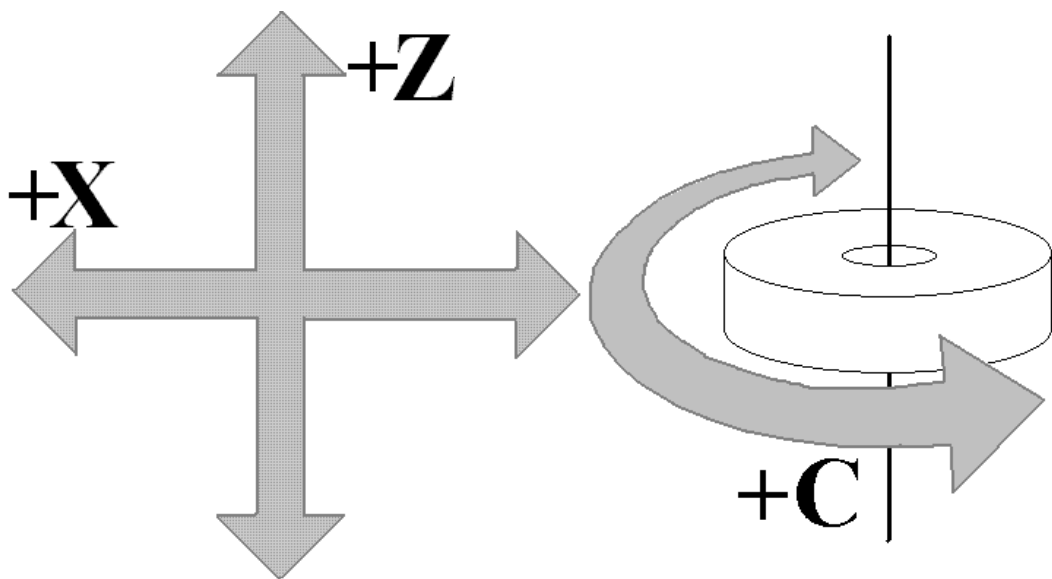


Kuva 8. Jäysteenpoistokoneen rakenne kehitysvaiheessa.

3.1 Yleistä

Koneen suunnittelun pohjana toimii C-akselisen sorvin akselirakenne, joka soveltuu parhaiten pyöreän kappaleen työstämiseen (kuva 9). Sorvista poiketen G3-

konetta käytettäessä ei ole mahdollista, eikä tarvetta, pyörittää kappaletta suurella kierrosnopeudella C-akseliin kytketyn karamoottorin avulla, vaan jysinterää pyörittävä karamoottori on sijoitettu X-akselin kelkkaan, C-akselin pyöriessä tarkkaliikkeisen servomoottorin sekä vaihteiston avulla. Koneessa hyödynnetään siis jysinkoneen sekä sorvin parhaita puolia, mutta koska leikkaava liike ei muodostu kappaleen, vaan terän pyörähdyksen seurauksena, konetta voidaan käsitellä jysimenä.



Kuva 9. G3-jäysteenpoistokoneen akselirakenne.

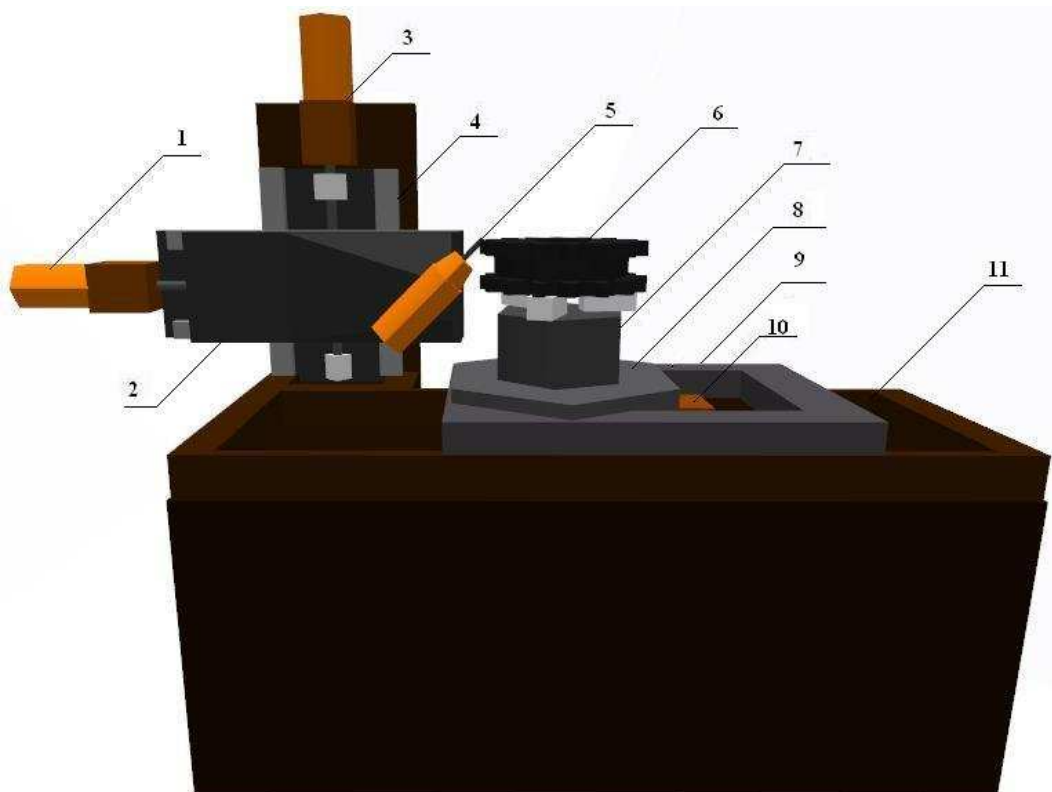
Itse työstö suoritetaan X-akseliin kiinnitetyn karamoottorin sekä jysinterän avulla, jotka asetetaan noin 45 asteen kulmaan kappaleen päätypintaan nähden (kuva 11). Kulmaa voidaan tarvittaessa muuttaa, mikäli kappaleen hampaaseen halutaan tehdä näkyvä viiste, jonka kulman arvo poikkeaa asetetusta.

3.2 Koneen mekaaninen rakenne

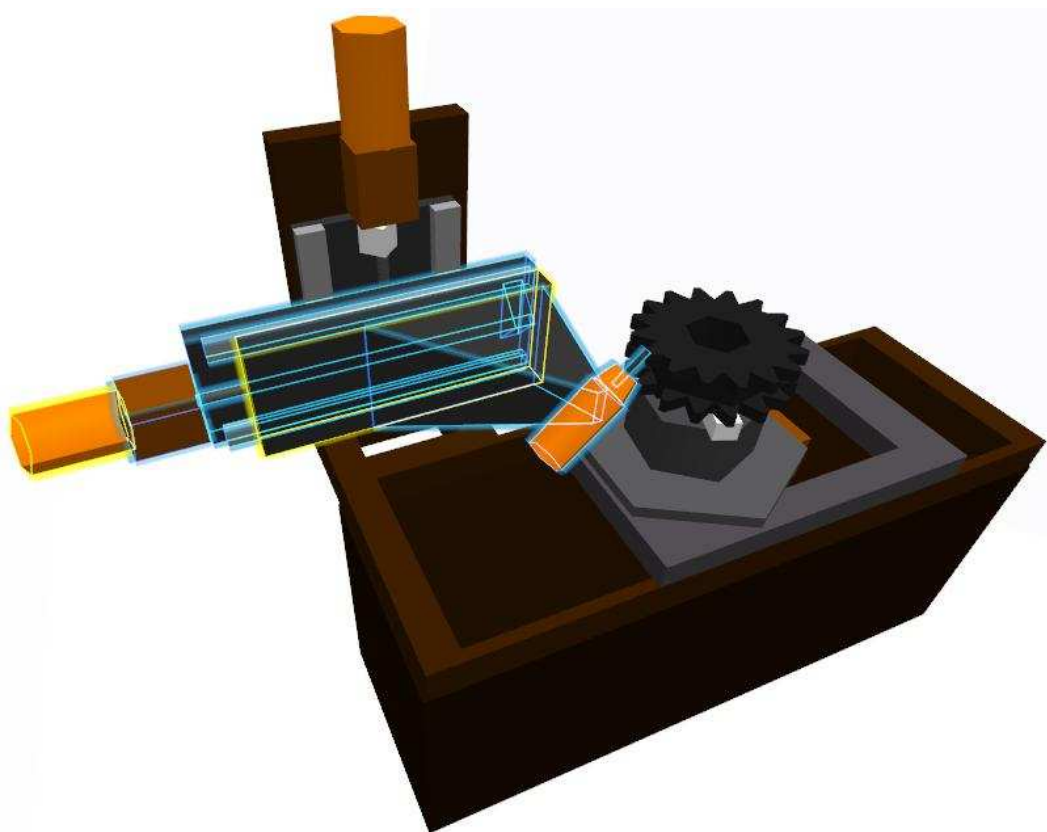
Jäysteenpoistokone (kuva 10) on jaettu yhteentoista eri osaan:

1. X-akselin servomoottori

2. X-akselin kelkka sekä lineaarijohteet
3. Z-akselin servomoottori sekä ruuvivaihde
4. Z-akselin lineaarijohteet
5. 45 asteen kulmaan asetettu jysinterä sekä karamoottori
6. Koneistettava kappale
7. Pakka kappalekiinnitystä varten
8. C-akselin pyöröpöytä
9. Pyöröpöydän runko
10. C-akselin moottori sekä siihen kuuluvat vaihteet
11. Koneen päärunko



Kuva 10. Jäysteenpoistokoneen mekaaninen rakenne.



Kuva 11. Jäysteenpoistokoneen X-akseli, karamoottori sekä jyrsinpää.

3.2.1. Ristisyöttöpöytä

Jyrsimen lineaariset X- ja Z-akselit päätettiin toteuttaa erillisen, jyrsimen pääruunkoon kiinnitettävän, ristisyöttöpöydän avulla. Pöydän X-Z -liikerata on maksimissaan 300mm x 150mm, ja se on tarkoitettu alun perin käsikäyttöiseksi. Liikeradat riittävät hyvin hammaspyörän hammaskuvion seuraamiseen, sillä jyrsinterän tarvitsema liikerata on maksimissaan vain noin 20-30mm, riippuen hammaspyörän modulista. Pöydän Z-akselin ruuvin päähän koneistettiin teline moottorille sekä pronssista jyrsityille kuivalle kierukkavaihteelle. Moottorin liike muutetaan lineaariseksi trapetsiruuvien avulla. Tämä on pöydän heikko lenkki, sillä trapetsiruuvien kitka sekä mittavirheet ovat kuularuuviin verrattuna suuria, eikä liikettä saada käytännössä välyksettömäksi. Myöskään pöydän laatu ei vastaa toivottua, sillä ohjelmiston testausvaiheessa huomattiin ruuvissa kieroutta, joka aiheuttaa mootto-

rille tarpeetonta vastusta. Jyrsintä tullaan tältä osin kehittämään, jotta mekaaninen osuus vastaisi ohjelmiston tarkkuutta.

3.2.2. Pyöröpöytä

Koneen C-akseli toteutettiin omavalmistepyöröpöydän avulla. Pöytä koostuu kolmesta erillisestä akselista: ensioakselista, toisioakselista sekä ruuviakselista. Ensiöakseli on laakeroitu välyksettömäksi kahden 80mm kartiolaakerin avulla, ja se koostuu 96-hampaisesta moduli 4:n hammaspyörästä, johon voima tuodaan toisioakselilta. Ensiöakselin yläosa koostuu itse pöydästä, johon työstettävä kappale, sekä kappaleen kiinnitykseen tarvittavat komponentit kiinnitetään. Toisioakseli on hammastettu 50mm väliakseli, joka on laakeroitu kahden 62mm lieriölaakerin avulla. Sen toiseen päähän on jyrsitty 16-hampainen hammaspyörä, toisen pään ollessa kiinnitettynä 29-hampaiseen pronssista valmistettuun kierukkapyörään. Ruuviakseli on niin ikään kahdella lieriölaakerilla laakeroitu akseli, joka koostuu yksipäisestä, moduli 2,5:n ruuvista, joka välittää voiman toisioakselin kierukkapyörään. Ruuviakseli on kiinteästi asennettuna servomoottorin ulostuloakseliin. Pyöröpöydän välitys moottorilta kappaleeseen on siis $16:96 * 1:29$ eli $1:174$. Koska moottoreiden hetkellinen maksimivääntö on 3Nm, saadaan tällä välityksellä teoreettiseksi vääntömomentiksi 522Nm. Käytännössä tämä ei kuitenkaan toteudu, sillä suorahampaisen hammaspyöräparin hyötysuhde on luokkaa 0.98 sekä ruuvivaihteen hyötysuhde huonompi, 0.2 - 0.98, riippuen suuresti hammaspyörien valmistusmenetelmistä, materiaaleista, voiteluaineista sekä rakennetun vaihteiston toleransseista (Beardmore 2010).

3.2.3. Kappaleen kiinnitys

Pyöröpöytään on koneistettu taso, johon kappaleen kiinnitykseen tarvittavat oheislaitteet voidaan asettaa. Taso sisältää ohjaukset sekä kiinnitinreiät, jotka on koneistettu DIN 55027 8-kartiokoon mukaisesti. Tämä mahdollistaa kärkisorveissakin yleisesti käytössä olevan kolmileukapakan käytön kappalekiinnityksessä. Pöytään tullaan projektin myöhemmässä vaiheessa rakentamaan erityinen mekaaninen asentokynsi, joka helpottaa uuden kappaleen asettamista oikeaan asentoon. Mikäli uutta kappaletta ei kiinnitetä tarkasti samaan asentoon kuin edellistä, työstöön

muodostuu jakovirhettä, eli kappaleeseen jyrittävä viiste tai siihen suoritettava jäysteenpoisto ei sijoitu oikeaan asentoon suhteessa hammaspyörän yksittäiseen hampaaseen.

3.2.4. Koneen runko

Koneen päärunko sekä pyöröpöydän runko on hitsattu 100x50mm rakenneteräspalkista, jonka seinämävahvuus on 3mm. Päärunkoon on koneistettu 28kpl ja pyöröpöydän runkoon 4kpl kohdistusreikiä, joiden avulla pyöröpöydän rungon sijaintia päärunkoon nähden voidaan käsikäyttöisesti muuttaa koneistettavan kappaleen mukaan. Näin ristisyöttöpöydän X-akselilta vaadittava liikerata on voitu minimoida, maksimi kappalekoon kuitenkin pysyvän n. 900 millimetrissä. Kohdistusreiät on jaettu päärungolle 150mm välein, joten ristisyöttöpöydän 200mm liikevara riittää kattamaan koko 900mm kappalekoon matkan.

3.2.5. Jyrsinpää

Jyrsinpää koostuu korkeakierroksisesta karamoottorista, karan rungosta sekä jyr-sinterästä. Karan runko ja sen kiinnitys on suunniteltu mahdollisimman tukevaksi jotta värinöitä ei esiintyisi. Karan runko kiinnitetään kuudella M12 pultilla X-akselin kelkan T-uriin. Runkoon kiinnitetään 25 000 1/min korkeakierrosmoottori, joka mahdollistaa kovametallisten koneviilojen käytön jäysteenpoistossa. Viisteytyksessä käytetään koneviilan sijasta kovametallista jyr-sinterää, jolloin moottorin nopeus lasketaan terän kuumenemisen estämiseksi n. 5000 1/min tasolle, jolloin esimerkiksi 6mm tappijyrsimellä leikkuunopeudeksi asettuu 100m/min.

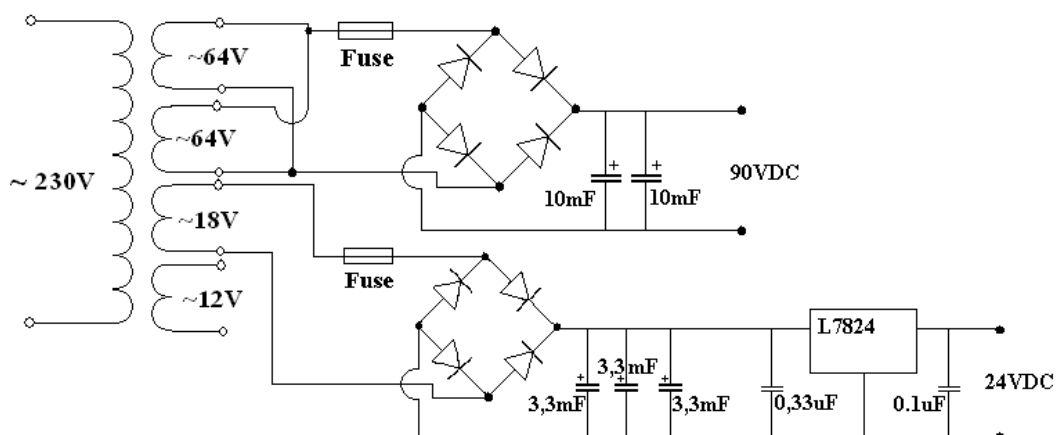
3.2.6. Virtalähde

Moottoreiden teholähteenä toimii yhdysvaltalaisen Antek Technology Corp. -yhtiön valmistama rengassydän- eli toroidimuuntaja AN-10464 (ks. liite 4), sekä tasasuuntaussilta. Muuntajan ilmoitettu maksimiteho on 1000VA. Muuntaja sisältää 6 erillistä käämiä, kaksi 115VAC ensiökäämiä, kaksi 64VAC toisiokäämiä sekä 18VAC ja 12VAC toisiokäämit. Kytkemällä ensiökäämit sarjaan saadaan Suomen sähköverkkoon soveltuva 230VAC ensiökäämi. 64VAC toisiokäämit

ovat rinnankytkennässä maksimivirran saavuttamiseksi. Ilmoitetut jännitearvot ovat tehollisarvoja. Muuntajan antaman jännitteen huippuarvo saadaan kaavasta

$$V_{pp} / (1/2^{(1/2)}) = V_{rms},$$

jossa V_{pp} on jännitteen huippuarvo sekä V_{rms} jännitteen tehollisarvo (Koskinen 2002: 9-10). Tasasuunnattu jännite on siis $64V * 1,4142 = 90,5VDC$. Rippelijännitteen minimoimiseksi tasasuunnattu jännite tasataan kahdella $10\ 000\mu F$ suotokondensaattorilla. $90VDC$ ulostuleva jännite on moottoreille ideaalinen, sillä ne on suunniteltu $70-90V$ käyttöjännitteelle (liite 3). Muuntajan $18VAC$ toisiokäämin jännite on niin ikään tasasuunnattu, sekä tasoitettu kolmella $3300\mu F$ suotokondensaattorilla. Käämistä saatava tasajännite on suuruudeltaan $25,4VDC$. Tämä soveltuu servovahvistimien käyttöjännitteiden lähteeksi, sillä Granite Devices'in VSD-A Rev-2 korttien ilmoitettu käyttöjännite on $19-26VDC$ (ks. liite 1). Mahdollisten jännitepiikkien estämiseksi piiri on reguloitu L7824-tyypin $24V$ regulaattorilla.



Kuva 12. Virtalähteen kytkentäkaavio.

Servovahvistimien virrankulutus on teknisten tietojen mukaan $200-800mA$, riippuen enkooderiporttien kuormasta (liite 1), mutta käytännön kokemukset ovat osoittaneet, että vahvistimien yhteenlaskettu virrankulutus pysyttelee tällä moottorikokoonpanolla $1100mA$ arvossa. Muuntajasta ulostulevan virran maksimiarvo

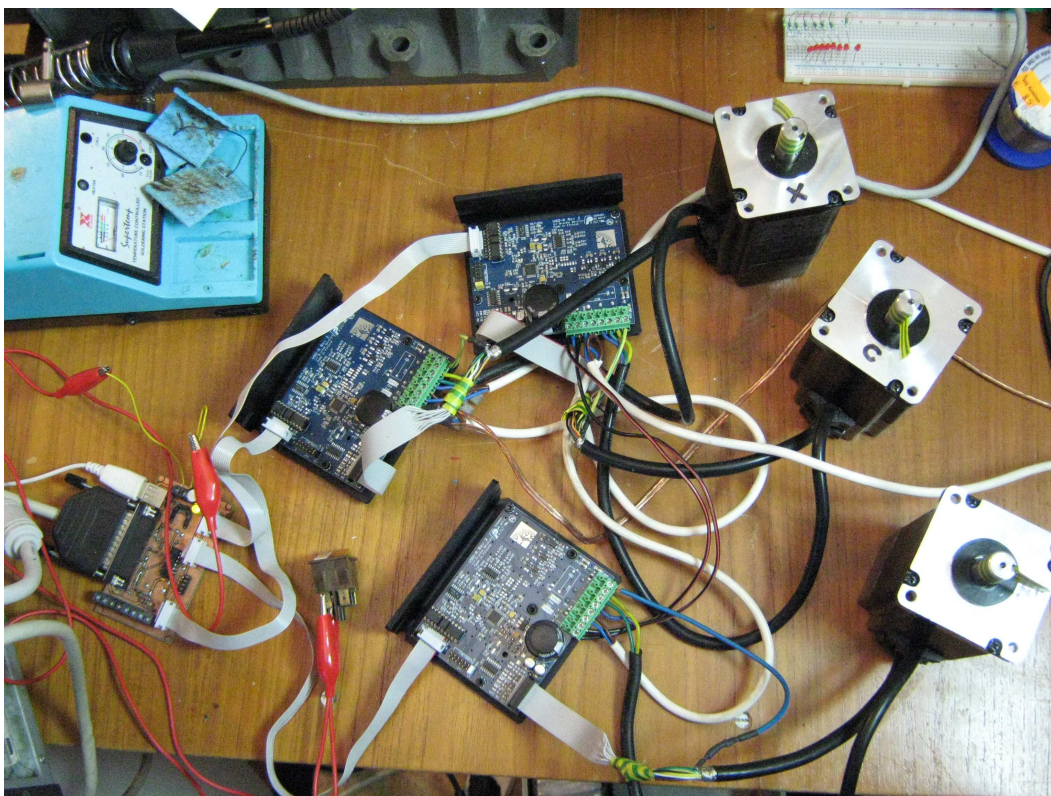
25,5VDC:llä on 2000mA, L7824 regulaattorin jatkuvan maksimivirran ollessa 1500mA, joten vahvistimien käyttöjännitteelle tarkoitetun virtalähteen teho on riittävä.

3.3 Ohjausjärjestelmän rakenne

Jyrsimen ohjaus koostuu kolmesta eri konkreettiseen osaan jaettavasta alueesta: pienoisjännitteellä toimivista servomoottoreista, pienoisjännitekäyttöisistä servo-ohjaimista eli -vahvistimista sekä loogisesta PC-ohjauksesta. Akseleiden paikoitus suoritetaan servomoottoreiden ja valmistajan servo-ohjainten välisellä takaisin-kytkennällä, liikerataa suorittava PC-sovellus ei ota kantaa liikkeen täsmällisyyteen. Position liittyvän vikatilanteen sattuessa servo-ohjaimilta välittyy kuitenkin tieto PC-ohjaukselle, jolloin ohjelma pysäyttää liikkeen ja mahdolliselta kolarilta vältytään.

3.3.1. Servomoottorit

Jyrsimessä käytetyt servomoottorit ovat 70-90V käyttöjännitteellä toimivia kolminapaisia harjattomia sekä kestopagneettisia tasavirtamoottoreita, jotka on kytketty servovahvistimiin (kuva 13). Moottorit kykenevät 300W jatkuvaan tehoon, sekä 3,0Nm hetkelliseen vääntömomenttiin. Moottorit sisältävät kiinteäasennetun 8-napaisen, 4000 CPR-resoluutioisen paikkaenkooderin, ja niiden maksimi kierrosluku on 3000rpm (liite 3). Johtuen moottoreiden suhteellisen pienestä vääntömomentista, C- sekä Z-akseleille valmistettiin vaihteistot liikkeen luotettavuuden takaamiseksi.



Kuva 13. Breakout-kortti, servovahvistimet ja servomootorit testausvaiheessa.

3.3.2. Servovahvistimet

Servovahvistimina käytetään Granite Device Oy:n valmistamia yleiskäyttöisiä VSD-A Rev-2 –ohjaimia, jotka kykenevät käskystä ohjaamaan erityyppisiä moottoreita (kuva 13). Vahvistimet toimivat 19-26VDC käyttöjännitteellä sekä kykenevät ohjaamaan maksimissaan 15A virtoja jännitealueella 24-200VDC (liite 1). Jokaiselle moottorille tulee olla oma vahvistimensa, ja jokaista vahvistinta täytyy ohjata erikseen PC-sovelluksesta käsin.

3.3.3. Servovahvistimien konfigurointi

Vahvistimet sisältävät ROM-muistin, johon käyttäjän tulee erillisellä valmistajan toimittamalla ohjelmalla sekä datasiirtokaapelilla konfiguroida moottorin, sekä siihen kytketyn enkooderin spesifiset arvot (Granite Devices Oy 2007). Vahvistimelle tallennetaan moottorin virran maksimiarvot, ja näiden arvojen perusteella

ohjain lähettää PC-sovellukselle vikasignaalin, mikäli moottoria kuormitetaan liian suurella voimalla. Myös moottorin toimintatapa tulee määrittellä, se voi olla torque-, velocity- tai position-tyyppinen. Torque-, sekä velocity-toimintatavat soveltuvat käyttöön, jossa oleellista on joko moottorin vääntö tai nopeus, mutta ei paikkatieto.

CNC-koneen akselikäytössä ainoa toimiva vaihtoehto on position-tapa, jolloin akselien liikkeitä voidaan tarkasti seurata. Vahvistimien muistiin tallennetaan myös asetus siitä, millä tavoin moottorien liiketieto välitetään tietokoneelta ohjaimille. Vaihtoehtoina on joko sarjamoottorin ja valmistajan itse määrittelemä SPI -protokolla, tai perinteinen, yleisesti käytössä oleva Step/Direction, jota tässäkin työssä käytettiin. Step/Dir -tekniikka tarvitsee minimissään kaksi loogista ulostuloa yhtä ohjain/moottori-yhdistelmää kohti. Toisesta signaalitulosta ohjain määrittelee moottorin pyörimissuunnan, toisesta nopeuden (kuva 16).

Moottorin nopeus määrittyy siitä, kuinka nopeasti ohjaimen Step-tuloa muutetaan loogisesta tilasta 0 tilaan 1. Ohjain muuttaa moottorin asentoa oletusarvoisesti yhden enkooderin askeleen verran aina ylösnousevalla signaalimuutoksella, joskin valmistajan servo-ohjaimissa on sisäänrakennettuna erinäisiä toimintoja joiden avulla sisääntuleva signaalin suhdetta enkooderin askeleisiin voidaan muuttaa.

Vaikka vahvistimien ohjaus on toteutettu rinnakkaisportin avulla, saman liittimen kautta tapahtuva konfigurointi suoritetaan USB-porttiin kytkettävällä kaapelilla. Vahvistimille tulee asettaa moottorin tiedot, joihin kuuluu mm.:

- Moottorityyppi (AC, DC, BLDC, Askelmoottori enkooderilla/ ilman enkooderia)
- Moottorin napojen lukumäärä
- Hetkellisen virrankulutuksen maksimiarvo
- Jatkuvan virrankulutuksen maksimiarvo
- Virrankulutuksen hälytysraja

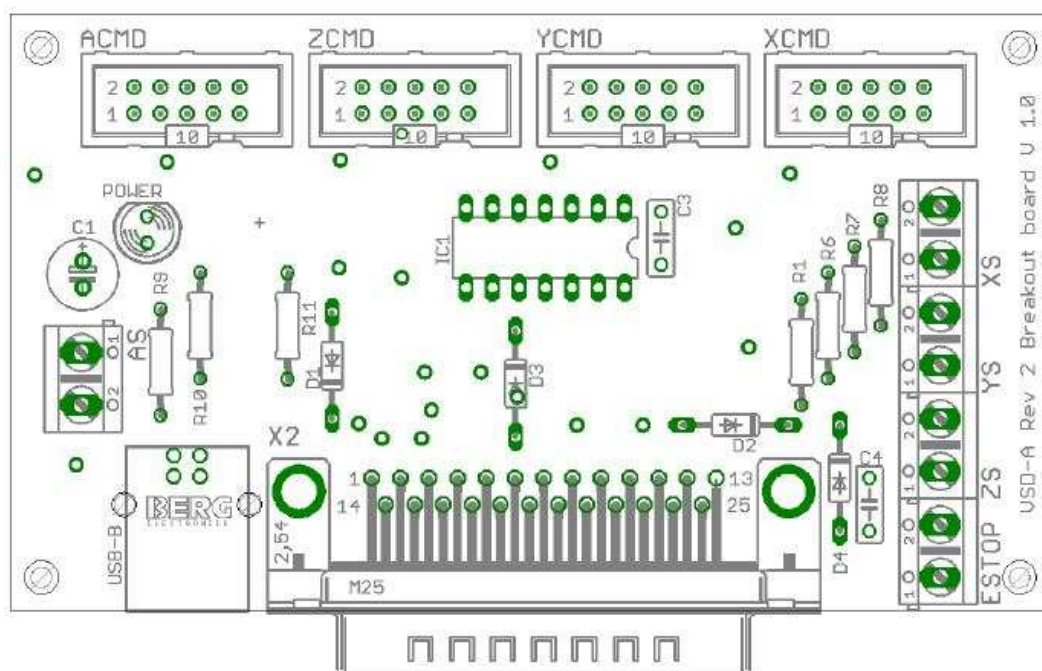
- Enkooderin resoluutio

Vahvistimien sisäänrakennetut kiihdytysrampit tulee säätää vastaamaan moottoreiden sähköisiä ominaisuuksia, moottoreiden ollessa asennettuna viimeisiin kohteisiinsa. Kiihdytysrampit on saatava erinäisten virta-arvo- sekä vahvistusmuuttujien avulla tilaan, jossa moottoreita ohjataan tarpeeksi riittävällä teholla position ylläpitämisen mahdollistamiseksi, mutta jossa moottoreiden liikkeessä ei kuitenkaan esiinny resonaatioita.

Vahvistimiin voidaan asettaa myös Anti-dither -arvot, joiden avulla vahvistin sallii moottoreiden positiolle pienen virheliikealueen ilman uudelleenpaikoitusta. Tällä saadaan eliminoidua servomoottoreille tyypillinen värinä, joka johtuu tilasta, jossa vahvistin yrittää säilyttää moottorin asentoa enkooderin kahden tai useamman askeleen välillä.

3.3.4. Breakout -kortti

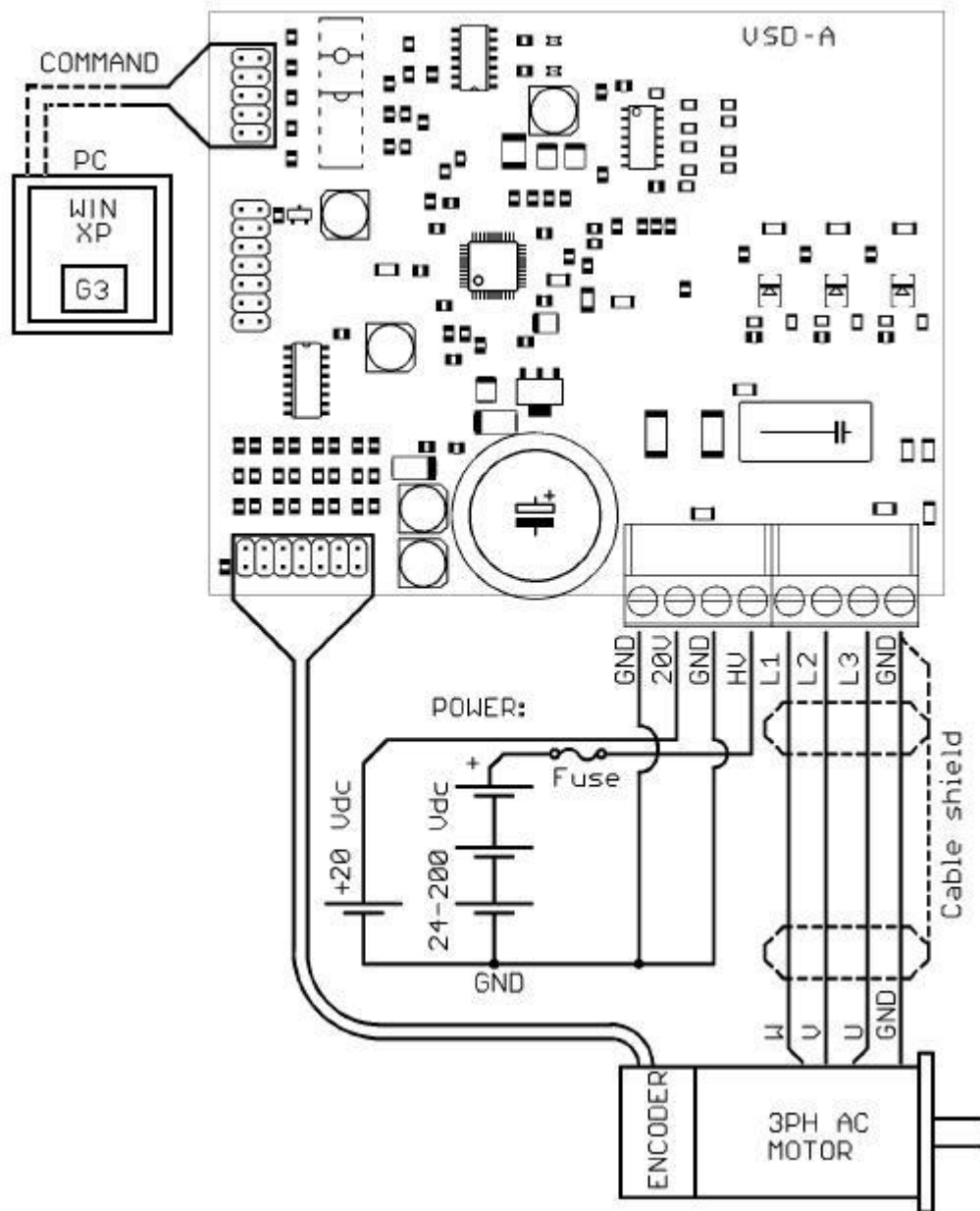
Granite Devices Oy on suunnitellut oheistuotteena erityisen breakout-piirikortin (kuva 14), jolla voidaan vaivattomasti kytkeä 1-4 ohjainta tietokoneen rinnakkaisporttiin. Tämän kortin tekniset tiedot, reitityskaaviot, valotuskalvot sekä osalistat asiakas saa veloitusetta ostaessaan yrityksen tuotteita. Tietokoneen rinnakkaisportti kytketään kaksipuoliseen piirikorttiin standardin mukaisella kaapelilla, lisäksi tarvitaan yksi vapaa USB-lähtö kortin +5VDC käyttöjännitettä varten. Korttiin voidaan liittää ohjainliitäntöjen lisäksi 1-4 rajakytkintä, esimerkiksi akseleiden ääriajakytkimiltä, kuin myös erillinen Hätä-seis -kytkin, josta tieto välittyy logiikkapiirien avulla sekä ohjaimille että ohjaavalle tietokoneelle. Valmistaja ei kuitenkaan suosittele suunnittelemaan järjestelmän turvatoimia pelkästään breakout-kortilla olevien keskeytystulojen varaan, vaan neuvoo kytkemään Hätä-seis -kytkimen siten, että kytkintä painettaessa moottoreiden käyttöjännite katkaistaan mekaanisesti. (liite 1)



Kuva 14. Breakout-kortti.

3.3.5. Järjestelmän kytkentä

VSD-A ohjainkortti on yksittäisen akselikytkennän keskus, sillä siihen kytketään niin tietokoneelta tuleva looginen ohjaus, moottorin navat, moottorin enkooderi kuin käyttöjännitteetkin (kuva 15). Tietokoneen rinnakkaisportilta tuleva Step/Dir-ohjaus kytketään ohjainkortin vasemmassa ylälaudassa sijaitsevaan CMD-liittimeen. Tätä liitintä käytetään myös ohjaimen konfiguroinnissa, jolloin liittimen alapuolella sijaitsevasta jumpperiliitimestä valitaan Step/Dir -toiminnon sijaan joko normaali konfigurointi, tai firmware -päivitys. Moottorin enkooderi kytketään ohjaimen vasemmassa alalaidassa sijaitsevaan 14-napaiseen liittimeen. Käyttöjännitteiden sisääntulot sekä moottorin ohjaus kytketään ohjaimen oikeaan alalaitaan. Sekä käyttöjännitteiden, että enkooderin signaalijohtimien tulee olla suojavaipallisia häiriöiden estämiseksi. (liite 1)



Kuva 15. Servo-ohjaimen kytkentä.

4. G3 JÄYSTEENPOISTOKONEEN OHJAUS

4.1 Järjestelmän vaatimukset

G3 jäysteenpoistokone vaatii toimiakseen ohjausohjelmistoa suorittavan tietokoneen. Lopullisessa kokoonpanossa tämä tietokone sijoitetaan kiinteästi koneeseen. Ohjelmiston suunnittelu, testaus ja toteutus on suoritettu vanhahkolla tekniikalla, tietokoneella joka sisältää 2GHz yksiydinprosessorin sekä 1024 megatavua DDR2 keskusmuistia. Käyttöjärjestelmänä toimii Microsoft Windows XP. Tämän luokan suorituskyky riittää ohjelman suorittamiseen, joskin tehokkaammalla kokoonpanolla jyrsimen liikkeiden maksiminopeudet saattavat nousta nykyisestä.

4.2 Rinnakkaisportti ja Windows XP

Rinnakkaisportin ohjaaminen on haastavaa .Net ympäristössä, sillä Windows XP perustuu NT-tekniikkaan, toisin kuin aiemmat Windowsin versiot, eikä käyttäjällä ole suoraa yhteyttä rinnakkaisporttiin. Lisäksi C# ohjelmointikielessä ei rinnakkaisporttiin ole lainkaan ohjelmointirajapintaa. Tähän ongelmaan on kuitenkin kehitetty ratkaisu Microsoftin ulkopuolelta. Logix4u.net tarjoaa ilmaisen DLL-tiedoston, joka liitetään ohjelmakehittäjän C#-kieliseen projektiin. Lisäksi projektiin tulee kirjoittaa luokkarajapinta, jonka avulla DLL-tiedostoa voidaan käyttää hyväksi. Alla kopioitu luokka G3-projektissa.

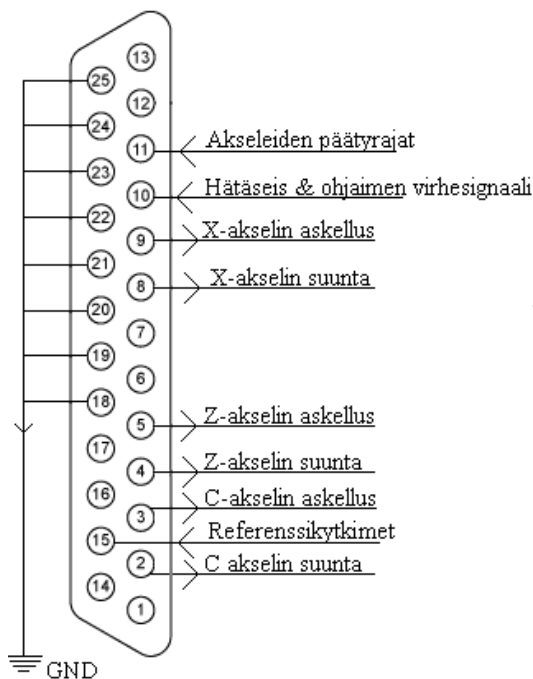
```
using System;
using System.Runtime.InteropServices;

namespace G3
{
    public class ParallelPortAccess
    {
        [DllImport("inpout32.dll", EntryPoint = "Out32")]
        public static extern void Output(int adress, int value);

        [DllImport("inpout32.dll", EntryPoint = "Inp32")]
        public static extern int Input(int adress);
    }
}
```

Liittämisen tuloksena ohjelmoija saa käyttöönsä kaksi metodia, joilla rinnakkaisporttia voidaan ohjata. `ParallelPortAccess.Input(int)`, sekä `ParallelPortAccess.Output(int, int)`. Input-metodiin viedään parametrinä rinnakkaisportin osoite ja se palauttaa int-arvona kyseisen osoitteen tilan, Output-metodiin taas viedään parametrinä osoite sekä rinnakkaisportin osoitteeseen lähetettävä arvo. Tietokoneessa olevan rinnakkaisportin osoitteen näkee Windowsin laitehallinnasta, LPT-portin ominaisuuksista.

Inpout32.DLL toimii luotettavasti Windows XP, ja sitä vanhemmissa versioissa. Kutsuttaessa kirjoitus- sekä lukufunktiota ohjelmasta käsin, DLL selvittää Windowsin version ja tekee tarvittavat toimenpiteet versiotiedon perusteella. Mikäli kyseessä on Win9X-version käyttöjärjestelmä, DLL käyttää järjestelmän valmiita toimintoja rinnakkaisportin ohjaamiseen. Mikäli kyseessä on NT-tekniikkaan pohjautuva Windows, DLL asentaa kernel-luokan ajurin, mikäli sitä ei ole jo asennettu, ja suorittaa käskyt tämän ajurin kautta. (Logix4u.net 1999)



Kuva 16. Rinnakkaisportin kytkentä.

4.3 Moottorinohjaussäikeet sekä käyttöliittymä

Moottoreiden sulavaliikkeiseen ohjaukseen tarvitaan useampi säie. Mikäli sovel-
lus rakennettaisiin yhdellä säikeellä toimivaksi, ei moottoreiden ohjausta ja käyt-
töliittymän päivitystä voitaisi suorittaa samanaikaisesti. G3-sovellus on toteutettu
eri tason säikeillä. Käyttöliittymäsäie suorittaa nimensä mukaisesti sovelluksen
käyttöliittymää. Tämä säie on ohjelmointiympäristön automaattisesti luoma. Käyt-
töliittymäsäikeellä luodaan erityinen pääsäie, joka taas luo alemman luokan työ-
säikeet. Pääsäieen tehtävänä on pitää järjestelmää koossa, toisin sanoen suorittaa
oikeat toiminnot ohjelman eri vaiheissa. Työsäikeet ovat pääsäieen alaisuudessa
toimivia työntekijöitä, jotka suorittavat vain niille käskettyjä yksittäistehtäviä. Tä-
tä säiehierarkiaa sovelletaan niin automaattiliikkeessä, manuaaliliikkeessä, refe-
renssiajossa kuin aloituspaikkaan ajossakin.

4.4 Moottoreiden nopeuden määrittäminen

Poiketen normaalista työstökoneesta, G3 ei määritä syöttö- tai pikaliikenopeutta
suhteessa aikaan tai terän pyörimisnopeuteen, vaan liikkeen nopeus on määritelty
prosentteina. Tämä prosenttiarvo on suhteellinen suure, joka ei sitoudu mihinkään
käyttäjän kannalta kiinteään arvoon. Käytännössä 100% liike on nopeus, jossa
moottoreiden katsotaan vielä kykenevän luotettavaan liikkeeseen. Johtuen käyt-
töympäristöstä sekä tietokoneen mahdollisesti hitaammasta kokoonpanosta, rin-
nakkaisporttiin lähetetyn liikesignaalin maksimi lähetysnopeus voi rajoittaa nope-
utta jo tätä aikaisemminkin. Nopeuden arvot tallennetaan kuitenkin jokaiseen tie-
tokantaan tallennetun liikeradan mukana, jolloin kappaleelle kerran sopiviksi ha-
vaitut syöttönopeudet ovat automaattisesti asetettuna.

4.5 Käyttöliittymän toiminta

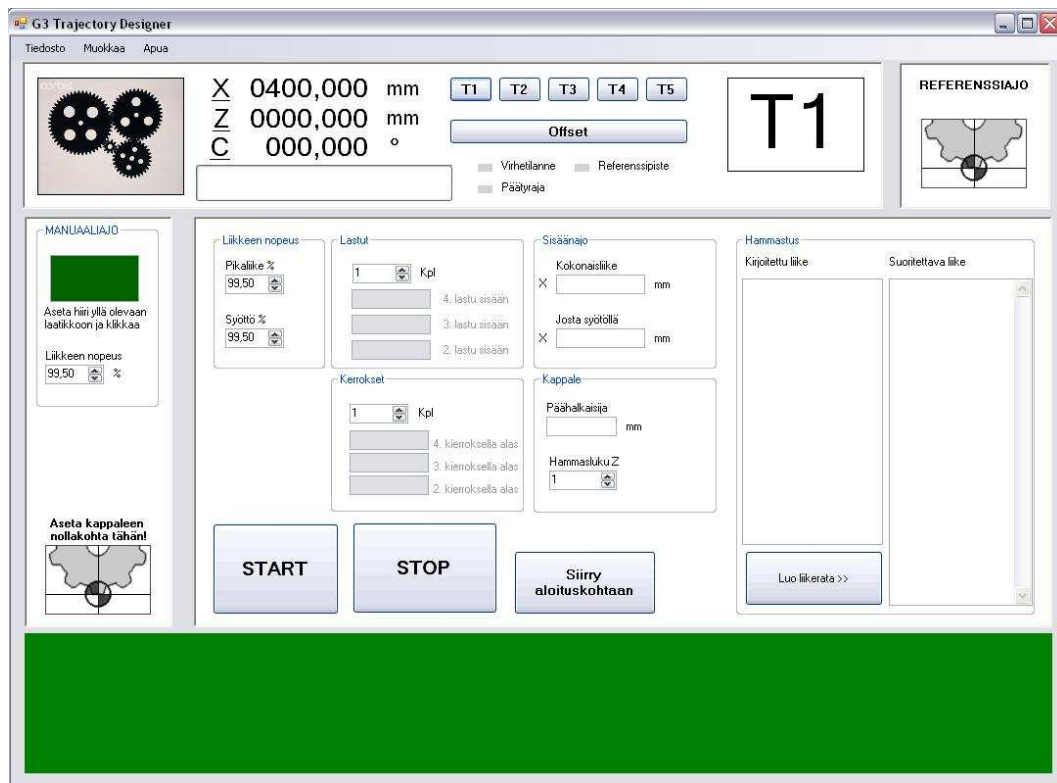
G3-sovelluksen käyttöliittymäikkuna (kuva 17) on jaettu neljään eri toimintaosi-
oon. Ikkunan yläosa koostuu referenssiajon kuvakkeesta sekä paikoitusosion
Paikoitusosion keskeisin toiminto on paikkakoordinaatisto, jonka avulla käyttäjä
näkee jyrsimen terän position suorittaessaan manuaaliajtoa tai kappaleen asetusta.
Paikoitusosio sisältää myös viisi teräkorjainta, offset-toiminnon sekä teräkorjain-

indikaattorin. Offset-toimintonapin alle on sijoitettu merkkivalot sovelluksen eri tiloille. Päätyrajatila sekä yleinen virhetila ilmoitetaan punaisella merkkivalolla. Näistä käyttäjä voi liikkeen pysähtymisen tapahduttua selvittää, johtuuko virhe ohjaimiin rakennetusta automatiikasta, vai akseleiden liikeratojen loppumisesta. Referenssipisteeseen saapuminen ilmoitetaan vihreällä merkkivalolla, josta käyttäjä voi seurata referenssiajon edistymistä. Onnistuneessa ajossa merkkivalon on syyttävä ja sammuttava kolme kertaa. Sovelluksen pääikkunan alalaitaan on myös sijoitettu suuri, koko ikkunan levyinen merkkivalo, josta käyttäjä voi pidemmänkin matkan päästä nähdä koneen automaattiajon tilan. Koneen ollessa valmiustilassa, merkkivalo palaa vihreänä. Mikäli työkierto on vielä kesken, merkkivalo palaa punaisena.

Ikkunan vasen alaosa on manuaaliliikkeelle sekä kappaleen asetukseen tarkoitettu osio. Se sisältää manuaalitoiminnon tarkoituksettoman liikkeen estämiseen kehitetyn lukon, manuaaliliikkeen nopeusvalitsimen sekä kappaleen nollakohdan asettamiseen tarkoitettua kuvakkeen.

Ikkunan suurinta osiota käytetään automaattiajon määrittämiseen. Se sisältää automaattiajossa suoritettavan pikaliikkeen ja syötön nopeusvalitsimet, monilastu- sekä monikerrosajon toimintaan liittyvät tiedonsyöttökentät, sisäänajon määrittämiseen liittyvät tiedonsyöttökentät, kappaleen fyysisten ominaisuuksien tiedonsyöttökentät, sekä liikeradan syöttökentän ja syötetyn liikeradan ajonaikaisen debug-toiminnon ikkunan.

Ikkunan ylälaidassa sijaitsevista alavetovalikoista käyttäjä voi aloittaa uuden kappaleasetuksen teon, hakea tallennetun liikeradan tietokannasta, tallentaa muodostetun liikerata tietokantaan, poistaa liikeratoja tai sulkea ohjelman.



Kuva 17. G3-sovelluksen pääikkuna.

Sovelluksen paikkakoordinaatio ilmoittaa C-akselin position asteina. C-position arvot ovat siis aina välillä 0-360 astetta, myös jos C-akselia liikutetaan manuaalisesti yli kierroksen. Z- sekä X-akselin positio ilmoitetaan millimetreinä. X-akselin tieto on selvyuden vuoksi halkaisijatieto sädetiedon sijaan. Toisin sanoen mikäli terällä otetaan kosketus 100mm halkaisijalla olevan kappaleesta, näytetään paikkakoordinaatistossa 100mm. Mikäli terää viedään ulospäin kappaleesta mitattuna 20mm, näyttää sovellus paikaksi 140mm eli terän sijainti kappaleen keskipisteestä on $100\text{mm} / 2 + 20\text{mm} = 70\text{mm}$.


4.5.1. Liikeratojen tallentaminen

Käyttäjä voi tallentaa liikeradat tietokoneen kovalevyille luotuun tietokantaan (kuva 19). Tallennusikkuna (kuva 18) avautuu valitsemalla Tiedosto-alasvetovalikosta kohta Tallenna.



Kuva 18. Liikeradan tallennusikkuna.

Tallennusikkunan auetessa käyttäjän tulee kirjoittaa tallennettavan liikeradan tunnistustiedot. Nämä tiedot auttavat liikeratoja tietokannasta haettaessa löytämään nopeasti halutun kappaleen liikeradan, mikäli kappaleen ohjelmanumeroa ei tiedetä. Käytännössä tiedoiksi riittää pelkkä kappaleen jako sekä hammasluku, mutta sekaannusten välttämiseksi on tietoihin suotavaa kirjoittaa esimerkiksi käytettävä materiaali ja/tai asiakkaan nimi. Liikerata voidaan tallentaa kahdella eri tavalla: tallentamalla liikerata uudella ohjelmanumerolla, tai tallentamalla muutettu liikerata vanhan, aikaisemmin avatun liikeradan päälle. Mikäli käyttäjä ei ole aikaisemmin avannut yhtään liikerataa ja yrittää tallentaa muutokset vanhan olemattoman liikeradan päälle, G3 huomauttaa asiasta virheilmoituksella.

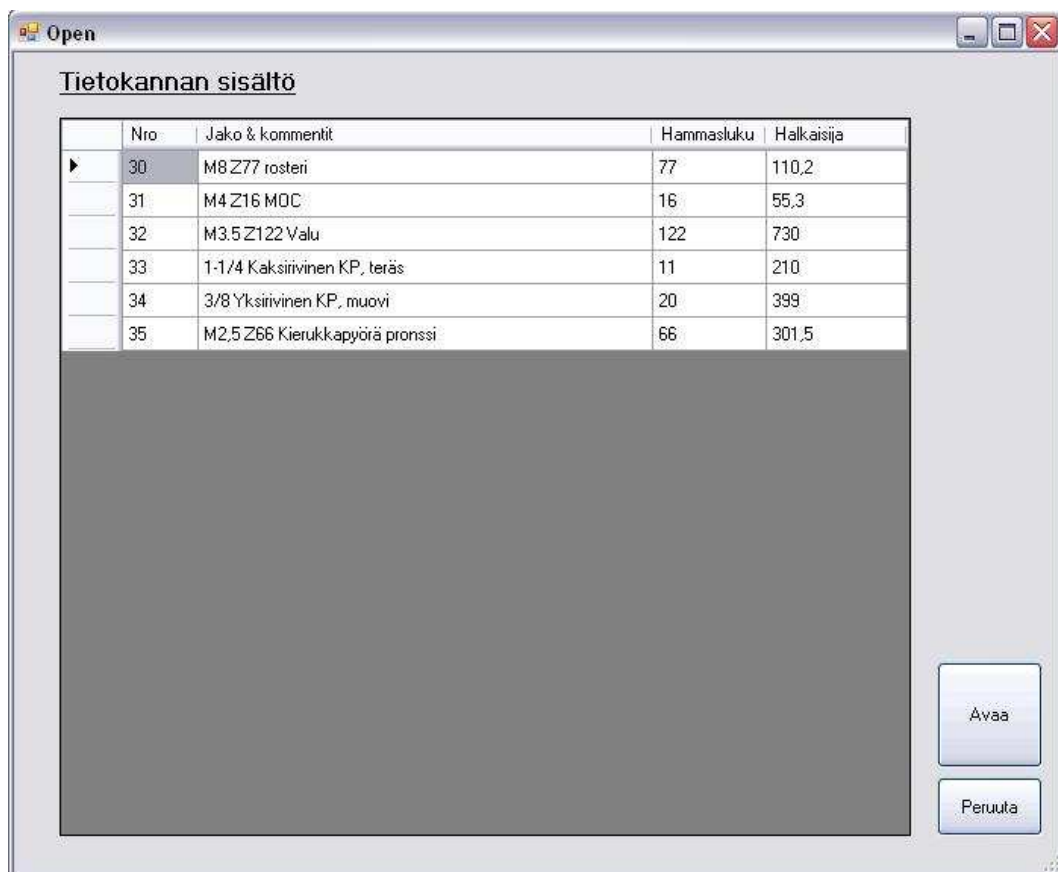
	Column Name	Data Type	Allow Nulls
	ID	int	<input type="checkbox"/>
	Jako	nchar(30)	<input type="checkbox"/>
	Hammasluku	int	<input type="checkbox"/>
	Halkaisija	float	<input type="checkbox"/>
	Pikaliike	float	<input type="checkbox"/>
	Syotto	float	<input type="checkbox"/>
	Sisaanajo	float	<input type="checkbox"/>
	Sisaanajo_syotto	float	<input type="checkbox"/>
	Kerros1	float	<input type="checkbox"/>
	Kerros2	float	<input type="checkbox"/>
	Kerros3	float	<input type="checkbox"/>
	Lastu1	float	<input type="checkbox"/>
	Lastu2	float	<input type="checkbox"/>
	Lastu3	float	<input type="checkbox"/>
	Liikerata	nchar(1000)	<input type="checkbox"/>

Kuva 19. Liikeratatiekannan rakenne.

4.5.2. Liikeratojen avaaminen

Tallennettuja liikeratoja voidaan hakea tietokannasta valitsemalla Tiedosto-
 alavetovalikosta kohta ”Avaa”. Tällöin näytetään avausikkuna (kuva 20). Avaus-
 ikkunan auetessa G3 hakee kovalevylle sijoitetusta tietokannasta tallennetut liike-
 radat. Liikeradat listataan ikkunan taulukkoon, josta käyttäjä voi valita työstettä-
 vän kappaleen liikeradan.

Käyttäjystävällisyyden vuoksi taulukkoon tulostetaan vain liikeratojen oleelli-
 simmat tiedot. Liikeradat on järjestetty oletusarvoisesti ohjelmanumeron perus-
 teella, joka toimii myös liikeratatiekannan pääavaimena. Liikeradoista tuloste-
 taan myös tallennusvaiheessa syötetyt kommentit sekä kappaletiedot, kuin myös
 automaattiliikettä määriteltäessä syötetyt kappaleen hammasluku- sekä halkaisija-
 tiedot. Mikäli käyttäjällä ei ole tiedossa tarvittavaa ohjelmanumeroa, voidaan tau-
 lukko järjestää myös muun sarakkeen mukaan aakkos- tai suuruusjärjestykseen.



Kuva 20. Avausikkuna.

4.5.3. Liikeratojen poistaminen

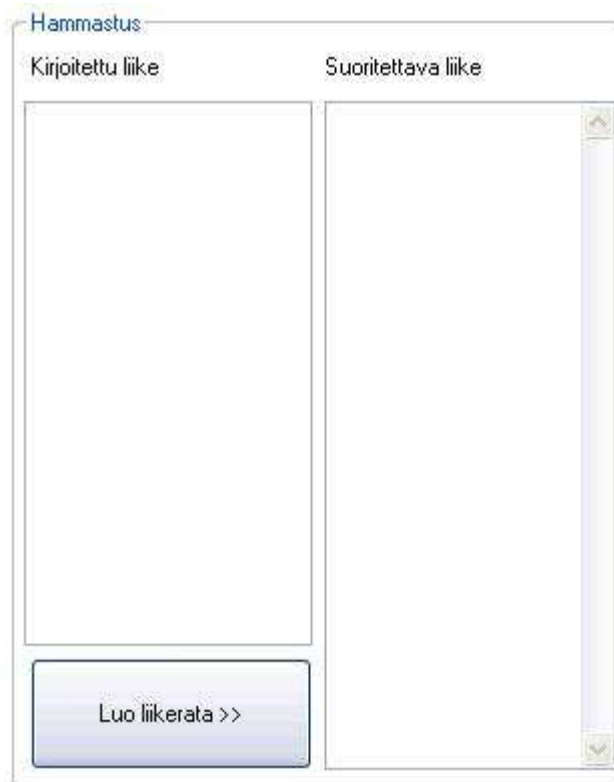
Liikeratoja voidaan poistaa valitsemalla Muokkaa-alasvetovalikosta kohta ”Poista liikeratoja”, jolloin näytetään poistoikkuna (kuva 21). Poistamiseen tarvitaan automaattisesti generoitu ohjelmanumero, jotka listataan liikeratoja avatessa. Ohjelmanumeron syötettyä voidaan liikerata poistaa ”Poista Ohjelma”-nappia painamalla. Sovellus tulostaa ilmoituksen, jossa ilmoitetaan joko onnistuneesta tai epäonnistuneesta poistosta.



Kuva 21. Poistoikkuna.

4.5.4. Ajonaikainen debug-ikkuna

Koska G3 on niin kutsuttu opetettava CNC-ohjaus, ei käyttäjän tarvitse kirjoittaa jokaista liikettä erikseen, vaan ohjelmaan syötetään tarvittavat tiedot kappaleesta ja ohjelma muodostaa liikeradat näiden tietojen perusteella. Tämä tuo työstökoneeseen uudenlaisen ongelman: Käyttäjällä ei ole mahdollisuutta seurata liikeratojen kulkua ja estää virheellisesti syötettyjen tietojen aiheuttamia onnettomuuksia. G3-sovellukseen on kehitetty tätä ongelmaa silmälläpitäen ns. ajonaikainen debug-ikkuna (kuva 22). Debug-ikkuna nimensä mukaisesti tulostaa kappaleelle suoritettavan työstöradan, jota käyttäjä voi tarkkailla ja havaita mahdolliset virheet. Tulostus tapahtuu joko käyttäjän painaessa ”Luo liikerata”-nappia, tai aloittamalla automaattisen työstön ”Start”-napista, jolloin liikerata tulostetaan automaattisesti.



Kuva 22. Liikeradan syöttökenttä sekä debug-ikkuna.

4.6 Liikeradan suunnittelu

Automaattiajaja hyödynnettäessä täytyy kappaleelle asettaa tarvittavat arvot liikeradan suunnittelua varten. G3 sovellukseen on sisäänrakennettuna toimintoja, jotka estävät käyttäjää syöttämästä virheellisiä tietoja. Mikäli numeerisiin kenttiin on syötetty ei-numeerista dataa, sovellus huomauttaa asiasta eikä liikerataa yritetä muodostaa tai suorittaa. Käyttäjälle jää kuitenkin vastuu numeeristen tietojen virheellisydestä, sillä sovellus ei voi tietää minkälainen kappale on kyseessä.

4.6.1. Terän sisäänajo

Vaihdettaessa koneeseen uutta kappaletta terän väistö on käytännössä välttämätön. Mikäli terä ei työstöradan päätteeksi liiku kappaleesta pois päin X-akselin suunnassa, ei uutta kappaletta voida vaihtaa ilman joko terän rikkoutumista tai sen position muuttumista. Kuitenkin riippuen kappaleen koosta, terän väistöliikkeen pi-

tuuden on oltava vaihdettavissa. Työstöradan aloituspiste määräytyy kappaleen halkaisijan sekä sisäänajomatkan summasta. Kiinteän väistöliikkeen tapauksessa terän väistö tulisi mitoittaa suurimman mahdollisen kappaleen mukaan, joka taas hidastaisi pienien kappaleiden työstämistä.

Kuva 23. Sisäänajon osio.

Sovelluksen väistöliike syötetään automaattiajoikkunan Sisäänajo-osioon (kuva 23). Sisäänajo koostuu kahdesta eri liikkeestä: pikaliikkeestä sekä syötöstä. Näin käyttäjä voi pikaliikkeen arvoksi asettaa tarvittavan pikaliikkeenopeudella ajettavan väistöliikkeen, sekä syötön arvoksi mitan, joka ajetaan pikaliikkeen jälkeen syötön nopeudella kappaleen pinnan läheisyydessä. Mikäli väistöliike toteutettaisiin pelkällä pikaliikkeellä, terä törmäisi kappaleeseen suurella nopeudella, joka aiheuttaisi kovametallisten terien rikkoutumista sekä värinäjälkeä kappaleeseen, työstöradan aloituskohdassa. Terän väistöliikettä tarvitaan myös monikerrosajossa, jolloin terän tulee siirtyä Z-akselin suunnassa toiselle kerrokselle.

4.6.2. Monilastuajo

G3 on suunniteltu yleiskäyttöiseksi ohjelmaksi joka mahdollistaa erilaisten kappaleiden työstämisen. Suuria kappaleita viistettäessä koneen rungon kevytrakenteisuus saattaa aiheuttaa karkeaa pinnanlaatua, tai lastuamisvoimien kasvaessa akselien servomoottoreiden tai karamoottorin tehot eivät riitä liikkeen ylläpitämi-

seen. Tämän takia sovellukseen on suunniteltu monilastuajo, jonka avulla käyttäjä voi valita 1-5 työstettävää lastua.



Kuva 24. Lastuajon osio.

Useamman lastun työstössä käyttäjä valitsee automaattiajon ikkunan Lastut-osiosta (kuva 24) lastujen lukumäärän sekä paksuuden. Mikäli käyttäjä valitsee 1-lastuajon, sovellus suorittaa työstöradan annettujen tietojen perusteella yhden keran. Kahden lastun tapauksessa sovellus jättää terän ensimmäistä työstörataa suorittaessaan 2. lastun lastunpaksuuden verran ulos X-akselin suunnassa, jonka jälkeen terä ajetaan syötön nopeudella 2. lastun lastunpaksuuden verran sisään, ja suoritetaan toinen työstörata. Useamman lastun tapauksessa työstöradat suoritetaan kuten kahden lastun tapauksessa, mutta lastunpaksuudet summataan yhteen ja lastujen työstöratojen suorittaminen aloitetaan kauempaa.

Tällä tekniikalla viimeinen työstörata suoritetaan aina samassa X-akselin positiossa, jolloin käyttäjän ei tarvitse ottaa lastunpaksuuksia huomioon liikerataa suunnitellessaan. Monilastuajolla voidaan myös suorittaa niin kutsuttu viimeistelylastu, jolloin ensimmäisellä työstöradalla suoritetaan karkea viisteen rouhinta, kun taas toisella, viimeisellä, lastulla terää siirretään X-akselilla vain n. 0.05mm, jolloin saavutetaan sileä ja värinätön pinta.

4.6.3. Monikerrosajo

G3:n sovellusta suunnitellessa on otettu huomioon myös ketjupyörien viisteytys ja jäysteenpoisto. Tämän vuoksi on suotavaa, että sovelluksella voidaan suorittaa myös monirivisen ketjupyörän työstö ilman liikeradan uudelleensuunnittelua. Monirivinen ketjupyörä koostuu kahdesta tai useammasta hammasrivistä, jotka on jyrskitty samalla kiinnityksellä eli hampaat ovat aksiaalisesti samassa linjassa. Luonnollisesti myös rivien hammasluku on sama. Tällaista kappaletta varten sovellukseen on suunniteltu monikerrosajo.



Kuva 25. Kerrosajon osio.

Käyttäjän tulee valita automaattiajoikkunan Kerrokset-osiossa (kuva 25) 1-5 työstettävää kerrosta. Käyttäjästävällisyyden vuoksi sovellus muuttaa syöttötietojen tekstikentät automaattisesti aktiivi- sekä passiivitilaan sen mukaan, montako kerrosta työstörataan on valittu. Jokaisen kerroksen välimatka Z-akselilla on muutettavissa, jolloin kappale voi koostua esimerkiksi kahdesta kaksirivisestä, yhdestä kaksirivisestä ja yhdestä kolmirivisestä, tai vaikkapa viidestä yksirivisestä ketjupyörästä. Hammasrivien välimatka riippuu myös ketjupyörän jaosta, joten välimatkan monipuolinen määrittely on välttämätöntä. Kerroksesta toiseen siirryttäessä sovellus käyttää hyväkseen kappaleelle ilmoitettua sisäänajomatkaa. Sovellus tekee terän X-akselin suuntaisen väistöliikkeen sisäänajomatkan mukaisesti, jotta toiselle kierrokselle voidaan siirtyä turvallisesti. Käyttäjän vastuulle jää määrittää

sisäänajomatkan suuruus siten, ettei terän alas siirtäminen aiheuta törmäystä kappaleeseen.

4.6.4. Syöttö ja pikaliike

Jokaisella tallennetulla kappaleella on oma pikaliike- sekä syöttöarvonsa (kuva 26). Käyttäjän tulee käytännön kokemusten perusteella valita tietylle työstölle sopivat nopeusarvot. Myös manuaaliliikkeen nopeus määritellään automaattiajon tavoin, mutta tätä arvoa ei tallenneta kappaleen tietojen mukana tietokantaan. Manuaaliliikettä voi laajan nopeusalueensa vuoksi käyttää joko pikaliikkumiseen tai syöttöön.

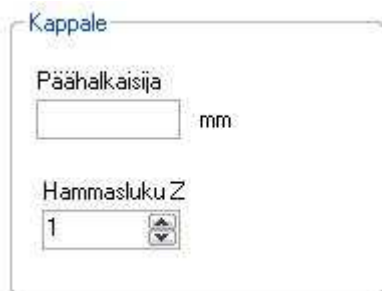


Kuva 26. Syötön ja pikaliikkeen osio.

4.6.5. Päähalkaisija sekä hammasluku

Kappaleen päähalkaisijalla tarkoitetaan hammaspyörän suurinta halkaisijaa, eli hampaan kärkeä. Päähalkaisija syötetään sovelluksen automaattiajoikkunan Kappale-osioon (kuva 27). Päähalkaisijaa tarvitaan sovelluksessa eri tarkoituksiin. Käytännössä halkaisijatietoa ei pelkistetyssä käytössä tarvittaisi, sillä terän liike ilmoitetaan inkrementaalisenä positiotietoina absoluuttisen sijaan, eikä kappaleen keskipisteellä tai halkaisijalla täten ole merkitystä. Kuitenkin kappaleen halkaisija on käyttäjän kannalta välttämätön, osin siksi, että haettaessa liikeratoja tietokannasta ohjelmat pystytään erottamaan toisistaan nopeammin, mutta suurimmaksi

osaksi siksi, että sovelluksessa tulostettavan ajonaikaisen akselipositiotiedon säilyttäminen helpottuisi.



Kappale

Päähalkaisija
 mm

Hammasluku Z

Kuva 27. Kappaletietojen osio.

Halkaisijaa tarvitaan myös tarkentamaan sovelluksen suorittamaa C-akselin synkronointia. Ilman halkaisijatieta, suunniteltuja liikeratoja on mahdotonta toistaa tarkasti, joka aikaansaa sen, että käyttäjän on tehtävä pieniä korjauksia liikkeisiin, jotta ne vastaisivat kappaleen muotoja.

4.7 Paikkatiedon ylläpito

Sovellus käyttää paikkatiedon määrittelemiseen kaupallisten työstökoneiden tavoin referenssipistettä. Tämä piste voi käytännössä sijaita missä tahansa akseleiden liikealueiden sisäpuolella. Koneen akselit ajetaan referenssipisteeseen, jossa sovellus nolaa jokaisen akselin positioarvot. Terän positiota verrataan aina referenssipisteen arvoon, joten kappaleen identtistä työstöä voidaan toistaa sovelluksen uudelleenkäynnistämisen jälkeen, mikäli kone on ajettu referenssiin.

Terän etäisyys referenssipisteestä on arvo, jota sovellus käyttää liikkeiden muodostamiseen. Käyttäjälle näkyvä paikkatieto on viitteellinen luku, joka muodostuu kappaleen halkaisijan sekä työkalukorjaimen arvoista. Suljettaessa sovellusta senhetkinen paikkatieto tallennetaan XML-dokumenttiin, jotta käyttäjä voi halutesaan jatkaa työstöä sovelluksen uudelleenkäynnistämisen jälkeen, ilman referenssiajoa. Näin voidaan säästää aikaa asetuksen teossa, mutta tämä edellyttää, että

akseleita ei ole mekaanisesti siirretty, tai etteivät ne ole esimerkiksi painovoiman vaikutuksesta liikkuneet sovelluksen ollessa suljettu. Vastuu tästä jätetään käyttäjälle, ja referenssiajo suositellaan suoritettavaksi aina uudelleenkäynnistämisen yhteydessä, riippumatta siitä onko akseleita siirretty.

4.7.1. Toiminta virhetilanteessa

Järjestelmän takaisinkytkentä suoritetaan moottoreiden ja Granite Devices Oy:n valmistamien servo-ohjainten välillä. Tästä syystä koneen toiminta ei ole kaupallisten CNC-koneiden luotettavuuden tasolla. Servo-ohjaimet eivät lähetä G3-sovellukselle tietoa moottoreiden positioista, tai siitä montako askelta kukin moottori on jäljessä ohjatusta liikkeestä. Vasta ohjaimiin asetetun virhearvon ylittyessä tieto virheestä lähetetään sovellukselle, ja tämän tiedon saapuessa sovellus pysäyttää liikeradan suorittamisen. Tämän tapahtuessa moottoreiden positio on saattanut olla virheellinen jo hetken aikaa, riippuen ohjaimiin asetetun virherajan arvosta.

Virhearvoa asettaessa tulee tehdä kompromissi tarkkuuden sekä nopeuden välillä. Mitä pienempi arvo asetetaan, sitä herkemmin vikasignaali lähetetään, mutta tämä asettaa tiettyjä rajoituksia liikkeen käynnistämisessä sekä pysäytyksessä. Moottoreiden kiihtyvyys ohjattuun arvoon riippuu moottorin pyörivien massojen inertiasista, sekä liikutettavan mekaanisen laitteiston massasta, eikä liikkeen nopeutta voida tehokkaasti muuttaa ilman tarpeeksi suurta virhearvoa. G3 jäysteenpoistokone asettaa vieläkin suurempia vaatimuksia laitteiston kiihtyvyydelle, sillä X-akselin liike on lyhyttä, nykivää edestakaista liikettä, joka käytännössä vaatii moottorin suunnan vaihtamista vastakkaiseen suuntaan erittäin pienessä ajassa, nopeuden kuitenkin pysyessä vakiona.

Virhesignaalin saapuessa sovellukseen, ohjelman suoritus pysäytetään, mutta varmaa tietoa moottoreiden positioista ei ole. Tämä johtuu Granite Devices Oy:n valmistamasta servo-ohjainohjelmistosta, johon on rakennettu automaattinen virhetilannetoipumistoiminto, mutta joka ei ole osoittautunut luotettavaksi. Ohjaimen tulisi pitää muistissa ohjatun liikkeen suunta ja määrä, ja virhetilanteesta toipueensa siirtää moottoreiden positio oikeaan arvoon, mutta virhetilanteita testattaessa toipumista ei aina tapahtunut. Tästä syystä ainoa luotettava tapa käsitellä virheti-

lanteet on pysäyttää ohjelman suoritus, ja ajaa moottorit takaisin referenssipisteeseen.

4.7.2. Päätyrajahälytys

Sovelluksessa saadaan päätyrajahälytys, mikäli mikä tahansa akseleihin asennettu päätyrajakytkin kytkeytyy. Rajojen tarkoitus on estää akseleiden liikkumista pisteeseen, jossa aiheutuisi mekaanisia vaurioita ristisyöttöpöydän liikeradan loppumisen seurauksena. Akseleiden päätyrajojen kytkeytyminen käsitellään sovelluksessa omana virheilmoituksenaan. Virheen aikana ei voida suorittaa mitään toimintoja, automaattiajoo tai perinteistä manuaaliajoo. Manuaaliajooon on kuitenkin rakennettu toiminto, jolla akseleita saadaan siirrettyä, mikäli virhetilanne johtuu pelkästään päätyrajahälytyksestä. Tästä kerrotaan lisää manuaaliliikkeen luvussa. Muun virheilmoituksen tapauksessa servo-ohjaimet on uudelleenkäynnistettävä hälytyksen poistamiseksi.

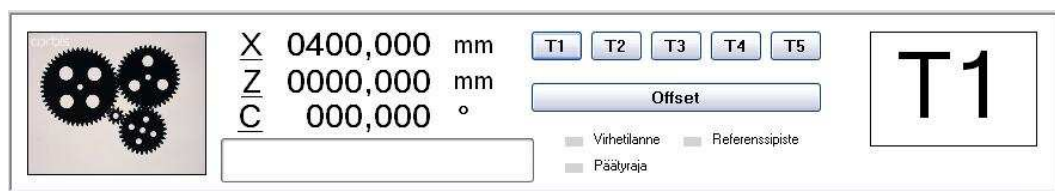
4.7.3. Työkalukorjaimet ja Offset-toiminto

G3 tarjoaa käytettäväksi viisi erillistä työkalukorjainta (kuva 29), joiden tarkoitus on pitää tallessa eri teriin tai kappaleisiin liittyviä paikkatietoja. Jokainen työkalukorjain sisältää tietyn arvon jokaiselle akselille. Korjainten arvot määritellään kappaleen nollakohdan määrittämisen yhteydessä.



Kuva 28. Offset-ikkuna.

Työkalukorjainten arvoja voidaan muuttaa myös Offset-toiminnolla (kuva 28). Offset-ikkuna aukaistaan, mikäli käyttäjä painaa paikoitusosiosta ”Offset”-nappia. Tällöin valittuna olevan työkalukorjaimen akseleiden arvoihin voidaan lisätä tai vähentää haluttu matka. Offset-toiminto on suunniteltu kappaleen asetuksenteon helpottamiseksi. Koska terän position arviointi on käytännössä hankalaa, voi käyttäjä tehdä karkean paikkamäärityksen, sekä ajaa ensimmäisen kappaleen työstöradan näillä arvoilla. Mikäli työstöjälki ei ensimmäisessä kappaleessa ole halutun mukainen, voi käyttäjä Offset-toiminnolla tehdä tarvittavat säädöt akseleiden positiotietoihin.



Kuva 29. Paikkatiedot, työkalukorjaimet sekä Offset-toiminto.

Työkalukorjainten arvot tallennetaan paikkatiedon tavoin automaattisesti kiintolevyllä sijaitsevaan XML-tiedostoon (kuva 30), josta ne sovellusta käynnistäessä haetaan niin ikään automaattisesti takaisin sovelluksen käyttöön.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
- <root>
- <tool>
  <index>0</index>
  <offset type="X">39999</offset>
  <offset type="Z">-70000</offset>
  <offset type="C">347663</offset>
</tool>
- <tool>
  <index>1</index>
  <offset type="X">239999</offset>
  <offset type="Z">-70000</offset>
  <offset type="C">347663</offset>
</tool>
- <tool>
  <index>2</index>
  <offset type="X">0</offset>
  <offset type="Z">0</offset>
  <offset type="C">0</offset>
</tool>
- <tool>
  <index>3</index>
  <offset type="X">0</offset>
  <offset type="Z">0</offset>
  <offset type="C">0</offset>
</tool>
- <tool>
  <index>4</index>
  <offset type="X">0</offset>
  <offset type="Z">0</offset>
  <offset type="C">0</offset>
</tool>
- <tool>
  <index>5</index>
  <offset type="X">0</offset>
  <offset type="Z">0</offset>
  <offset type="C">0</offset>
</tool>
</root>

```

Kuva 30. Työkalukorjainten sekä teräposition XML-taulun rakenne.

4.7.4. Hammasjaon säilyttäminen

C-akselin paikan määrittäminen on käyttäjälle hankalaa, sillä liikkeitä muodostaessa oikeaoppinen tapa olisi käsitellä C-akselin, eli kappaleen asennon arvoa asteina. Käyttäjälle on kuitenkin erittäin haastavaa, tai jopa mahdotonta arvioida hammaspyörän hampaan C-akselin suuntaisia ulottuvuuksia asteina, joten G3 sovellukseen on kehitetty erikoinen järjestelmä, jonka avulla käyttäjä voi tarvittaessa mitata kappaleesta esimerkiksi hampaan leveyden, sekä ilmoittaa tämän perusteella tiedot millimetreinä. Tämän järjestelmän avulla liikeradan muodostaminen hampaan arvioinnin avulla on huomattavasti tarkempaa kuin asteita käsiteltäessä.

Käyttäjän syöttämässä yhden hampaan liikeradassa tulisi siis C-akselin inkrementaalisten arvojen summa olla myös osamäärä, joka saadaan jakamalla kappaleen päähalkaisijalla sijaitsevan kehän pituus kappaleen hammasluvulla. Koska käyttäjän mittaama arvo kappaleesta ei ole hampaiden välisen kehän pituus päähalkaisijalla, vaan hampaiden välille muodostuvan jänteen pituus, on mittaustapa jossain määrin epätarkka. Lisäksi käyttäjä voi antaa jänteen pituusmittoja samaan liikerataan sekä kappaleen hampaan kärjestä että hampaan pohjasta, joka osaltaan aiheuttaa myös epätarkkuutta sillä halkaisija, jolla mittaus suoritetaan, eriiä tapausten välillä. Ongelma korostuu suurimodulisen sekä pienihammaslukuisen kappaleen ollessa kyseessä, sillä hampaan pää- sekä pohjahalkaisijalla olevien kehien pituudet voivat olla hyvinkin kaukana toisistaan.

Tästä syystä G3 sovelluksessa tapahtuva liikkeenmuodostus ei käsittele käyttäjän syöttämiä C-akseliliiketietoja sellaisenaan, vaan liike muodostetaan suhteuttamalla syötetyt arvot toisiinsa.

Pelkistettynä erimerkkinä voidaan ajatella ohjelmaa, jossa terä pysyy vakioetäisyydellä kappaleen keskipisteestä, eli työstettävä halkaisija sekä kehäpituus ovat vakioarvoisia: Käyttäjä syöttää 36-hampaiselle, sekä päähalkaisijaltaan 150mm kappaleelle, yhden hampaan liikerataan C-akselin inkrementaaliseksi arvoiksi $3 + 3 + 3 + 1.5 + 1.5$ mm, eli yhteensä 12mm matkan. Koska kappaleen kehäpituus 150mm halkaisijalla on $150\text{mm} * \pi = 471.24$, jaettuna 36:lle hampaalle on 13.09mm, käyttäjän antamien arvojen mukaan suoritettaisiin vain 432mm matka

kehällä, joten kappale ei pyörähtäisi täyttä kierrosta. Tämä taas aiheuttaisi sen, että lyhyemmälle matkalle mahdutettaisiin sama määrä hampaita kuin kappaleetoihin on syötetty, jonka vuoksi hammasjako muuttuisi, eikä terä seuraisi kappaleen muotoja ja työstettävä kappale olisi pilalla. Sama ongelma saattaisi toistua myös astearvoja käytettäessä, jolloin käyttäjän tulisi pitää huoli, että syötettyjen C-akselin arvojen summan ja kappaleen hammasluvun tulo olisi aina tarkalleen 360 astetta.

Hammasjaon määrittelyä ei siis voida luotettavasti muodostaa käyttäjän syöttämien tietojen pohjalle. Tästä syystä G3:n liikemuodostuksessa yksittäisen hampaan liikeradan toistojen määrä mahdutetaan aina täydelle kierrokselle kappaleen hammasluvun mukaan. C-akselin arvot suhteutetaan toisiinsa ja C-akselia siirretään vain tarvittava määrä.

Aiemmassa esimerkissä C-akselin liike hammasjaon säilyttäviin asteisiin muutettuna on $2.5 + 2.5 + 2.5 + 1.25 + 1.25$ astetta = 10 astetta, sillä hammasluvulla 36 kerrottuna tulokseksi saadaan täydet 360 astetta. Mikäli käyttäjä syöttäisi C-akselin arvoiksi $30 + 30 + 30 + 15 + 15$ mm, pysyisi työstettävä rata silti muuttumattomana, sillä syötettyjen arvojen suhde toisiinsa ei muutu, eikä sovellus käytä arvoja muulla tavoin.

4.7.5. C-akselin synkronointi

Sovelluksen suorittama liike perustuu vakionopeuksiseen C-akseliin. Tämä tarkoittaa sitä, että C-akselin työstöradan aikainen kierrosnopeus on aina vakio, riippumatta muiden liikkeiden arvoista. Kuitenkin muiden, käytännössä X-akselin, liikkeet saattavat rajoittaa C-akselin maksiminopeutta. C-akselin nopeuden on pysyttävä vakiona, sillä akselille kiinnitettävän kappaleen massa voi olla useita satoja kiloja, eikä servomoottori kykene nopeasti kiihdyttämään tällaista kappaletta työstöradan vaatimaan nopeuteen. Tällä myös estetään pyöröpöytämekaniikan enenaikainen kuluminen.

Sovellukseen syötettävä syöttönopeus ei ole kiinteästi sidoksissa C-akselin nopeuteen, vaan sillä määritellään nopeimman akselin liike, joka useimmiten on X-

akselilla. Riippuen muodostetusta liikeradasta, syöttönopeus voi kuitenkin tarkoittaa C-akselin nopeutta. Koska C-akseli toimii vakionopeudella, ja mikäli syöttönopeus määräytyy X-akselin mukaan, C-akselin nopeus määritellään siitä syötetyn liikeradan yksittäisestä point-to-point -liikkeestä, jonka aikana C-akselin nopeus on pienimmillään. Tämä tarkoittaa sitä, että mikäli yhden liikkeen aikana C-akseli siirtyy huomattavan vähän, X-akselin samanaikaisesti siirtyessä huomattavan paljon, on C-akselin nopeuden oltava tätä liikettä suorittaessa pieni, sillä asetettu syöttönopeus rajoittaa X-akselin liikkeen nopeutta. Seuraavaa liikettä suorittaessa C-akseli saattaa liikkua matkan, jonka aikana X-akseli ei liiku lainkaan, mutta myös tällä siirtymällä on käytettävä hitaimman liikkeen nopeutta, mikäli C-akselin halutaan olevan vakionopeuksinen.

Tällä tekniikalla on haittapuolena tiettyjä liikkeitä suorittaessa liikkeen tarpeeton hitaus, mutta syöttönopeutta ei myöskään voida kiinteästi sijoittaa C-akseliin, sillä riippuen suunnitellussa liikeradassa olevien akselien liikematkojen suhteesta toisiinsa, X-akselin työstönopeus saattaisi kohota joko arvoon, jolla servo-ohjain lähettää vikasignaalin moottoreiden tehojen loppuessa kesken, tai arvoon, jolla työstön jälki on karkea ja värinällinen.

5. KAPPALEKOHTAISEN ASETUKSEN TEKO

Jokaista kappaletta kohden on vähintään kerran suoritettava kappalekohtainen asetuksenteko. Mikäli kappaleen liikerata on aikaisemmin tallennettu tietokantaan, kappaleen työstöön käytetyn työkalukorjaimen arvoja ei ole muutettu, eikä pyöröpöytärunkoa ole siirretty päärungolla, voidaan asetuksenteko ohittaa ja työstö aloittaa pelkällä referenssiajolla.

5.1 Referenssiajo

Ennen asetuksentekoa tai työstön aloittamista tulee kone ajaa referenssiin. Tämä tapahtuu siten, että jokainen akseli siirretään manuaaliliikkeen avulla referenssipistettä lähellä oleviin positioihin. Nämä positiot on merkattu akseleiden sivuihin kohdistusviivoilla. Kohdistusviivojen tarkka kohdistaminen on tarpeetonta, sillä tämän toiminnon tarkoituksena on lähinnä nopeuttaa referenssiajtoa, mikäli akselit sijaitsevat kaukana referenssipisteestä. Kun akselit on sijoitettu lähelle referenssipistettä, referenssiajo voidaan aloittaa.



Kuva 31. Referenssiajon ikkuna.

Painamalla sovelluksen pääikkunan oikeasta ylä laidasta referenssiajon kuvaketta, käyttäjä voi avata erillisen ajoa varten suunnitellun ikkunan (kuva 31). Ikkunan auetessa käyttäjälle tarjotaan mahdollisuus valita akselit, joille referenssiajo suoritetaan. Tämä siksi, että monikerrosajon vähäisen tarpeen vuoksi Z-akseli on vertikaalisen asentonsa takia saatettu lukita mekaanisesti, jotta se ei painovoiman vaikutuksesta menetä positiotaan laitteen ollessa sammutettu. Lukitun akselin siirto-yritys ja referenssiin ajo tuottaisi välittömän virhesignaalin.

Sovellus ajaa valitut akselit vuorotellen referenssipisteeseen. Paikoitus perustuu liikkeeseen, jossa akselin referenssiin syöttö suoritetaan aina vakionopeudella, kunnes referenssikytin painautuu. Tämän jälkeen sovellus pysäyttää akselin liikkeen 5ms aikarajan sisällä, sekä ajaa akselin ulos, jolloin kytkin vapautuu. Ulosajo suoritetaan, koska eri akseleilla olevat kytkimet on kytketty rinnakkaisportin rajallisten liitäntöjen takia rinnankytkentään, jonka vuoksi edellisen akselin kytkimen täytyy vapautua, jotta seuraava akseli voidaan ajaa referenssiin.

Referenssikytimet ovat breakout-kortin sekä rinnakkaisportin kautta suoraan yhteydessä sovellukseen. Paikoituksen toimivuuden kannalta referenssikytimien tulee olla kytkeytymispisteidensä toistotarkkuuksien osalta tarkkoja. Elektroniikkaliikkeissä myynnissä olevien, paikoitukseen tarkoitettujen, mikrokytkimien toistotarkkuus on käytännön kokemusten perusteella osoittautunut kyllin riittäväksi, kytkimien hajonta on ollut $n. \pm 30\mu\text{m}$, joka riittää varsin hyvin jäysteenpoistokoneen käyttöön.

Kun valitut akselin on vuorotellen ajettu referenssiin, sovellus ilmoittaa onnistuneesta ajosta sekä sulkee referenssiajon ikkunan.

5.2 Manuaalisesti liikkuminen

Manuaaliliikettä voidaan käyttää aina, kun automaattiajo tai nollakohtaan palautus ei ole kesken. Liikkeen käyttämiseksi on painettava hiirellä pääikkunan vasemmassa laidassa sijaitsevasta manuaaliosiosta (kuva 32) vihreää nappia. Nappia painettaessa manuaaliliike aktivoituu ja nappi muuttuu punaiseksi. Mikäli hiiren kursori siirretään pois napin yltä, manuaaliliike deaktivoidaan välittömästi, ja na-

pin väri muuttuu jälleen vihreäksi. Tämän toiminnon tarkoitus on estää manuaaliliikkeen tahaton käyttö, sillä tietokoneen näppäimistö sekä hiiri eivät ole ideaalisia hallintalaitteita työstökoneen käyttöön. Käyttäjän tulee asettaa manuaaliliikkeen nopeus joko hiirellä ennen manuaaliajtoa, tai manuaaliajon aikana näppäimistön + sekä - merkein. Mikäli hiiren kursori säilytetään napin päällä, voidaan koneen akseleita siirtää näppäimistöllä. X-akselia voidaan liikuttaa vasemmasta sekä oikeasta nuolinäppäimistä, Z-akselia taas ylä- sekä alanuolinäppäimistä. C-akselin positiota taas muutetaan S- sekä X-näppäimistä.



Kuva 32. Manuaaliosio.

5.2.1. Liike päätyrajahälytyksen aikana

Mikäli sovellus on havainnut päätyrajahälytyksen, eli tilan, jossa liikkeen jatkuaminen aiheuttaa mekaanisia vaurioita akseleiden voimansiirrolle, tulee tästä tilasta poistua manuaaliajtoon rakennetulla erikoistoiminnolla. Tällöin käyttäjän on aktivoitava manuaaliliiketoiminto, sekä siirrettävä hälytyksen aiheuttama akseli pois päätyrajasta pitämällä näppäimistön Insert-nappia yhtäaikaisesti pohjassa. Tällä

toiminnolla koneen hälytystila on saatettu käyttäjän tietoon, eikä vauriota aiheuttavaa liikettä voida vahingossa jatkaa käyttäjän ymmärtämättä tilannetta.

5.3 Akseleiden nollakohtien määrittely

Mikäli koneelle tehdään uuden kappaleen asetus, täytyy myös nollakohdat määrittellä uudestaan. Tämä tapahtuu manuaaliliikkeen avulla. Käyttäjän tulee syöttää kappaleen päähalkaisijatieto automaattiajon osioon, siirtää terä manuaalisesti X- sekä Z-suunnissa mahdollisimman lähelle kappaleen hampaan kulmaa, sekä asettaa C-akseli siten, että terä kohdistuu keskelle hammasta. C-akseli voidaan käytännössä kohdistaa mihin kohtaan hammasta tahansa, mutta liikeradan suunnittelun selkeyden vuoksi kohdistus suositellaan sijoitettavan hampaan keskilinjaan. X-akselin positiota asetettaessa voidaan apuvälineenä käyttää esimerkiksi paperipalaa, joka asetetaan terän ja kappaleen väliin: Terää tuodaan pienellä syötöllä lähemmäs kappaletta, liikuttaen paperia samalla edestakaisin, kun paperia puristuu terän ja kappaleen väliin siten, ettei se enää liiku vapaasti, on terän etäisyys kappaleesta paperin paksuuden verran eli n. 0,10mm.

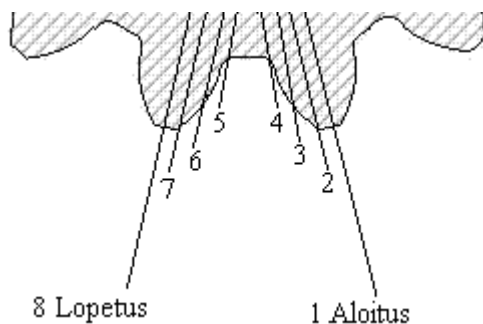
Tämän jälkeen painetaan manuaaliosiossa sijaitsevaa ”Kappaleen nollakohta”-kuvaketta (kuva 32). Tällöin käytössä olevan työkalukorjaimen X-akselin arvoksi asetetaan kappaleen päähalkaisijaksi ilmoitettu mitta, Z- ja C-akseleihin tallentuksessa arvo 0. Sovellus käyttää näitä arvoja työstöradan pohjana, joten esimerkiksi monikerrosajossa uudelle kerrokselle siirryttäessä sovellus siirtää terää Z-akselin nollakohdan alapuolelle.

5.4 Liikeradan määrittely

Liikerata muodostetaan automaattiajoikkunan hammastusosiossa olevien käyttäjän syöttämien tietojen mukaan. ”Kirjoitettu liike” -tekstikenttään tulee syöttää yhden hampaan työstöön tarvittava liikerata. Liikerata suositellaan kirjoitettavaksi siten, että liikkeen aloituspaikka sijaitsee hampaan kärjen keskipisteessä, sillä tässä pisteessä silmämääräinen C-akselin kohdistus on helpointa. Liikkeen aloittaminen esimerkiksi hampaan pohjasta on myös turvallisuusriski.

5.4.1. Esimerkki liikeradasta

Sovellus muodostaa koko kappaleen liikeradan yksittäisen hampaan liikeradan pohjalle, jolloin käyttäjän tulee työstöradan laskemiseksi asettaa lisäksi vain kappaleen hammasluku sekä päähalkaisija. Ohjelmankirjoituksessa tulee kuitenkin ottaa huomioon terän halkaisija, koska sovellus ei pisteestä pisteeseen liikkuesaan ota kantaa siihen, kuinka paksu terä on. Käytännössä tämä tarkoittaa, että terän siirtymää matkaa hampaan pohjassa tulee pienentää sekä hampaan kärjessä kasvattaa. Käyttäjä voi myös manuaalisesti liikuttaa terää pisteestä pisteeseen läpi halutun liikeradan, ottaen paikkakoordinaatistosta samalla pisteiden positioita ylös tulevaa ohjelmankirjoittamista varten.



Kuva 33. Ohjelmaesimerkki.

Esimerkkiohjelmassa (kuva 33) liike muodostetaan hampaalle, jonka korkeus on 10mm, sekä hampaan kärjestä kärkeen muodostetun janan pituus noin 8mm. Ohjelmakoodi tulee kirjoittaa muotoon:

- 1)
- 2) X0C1;
- 3) X-5C1.2;
- 4) X-5C0.8;
- 5) X0C2;
- 6) X5C0.8;

7) X5C1.2;

8) X0C1;

(Rivinumeroitua ei kirjoiteta koodiin)

Kirjoitusmuoto vastaa siltä osin standardia CNC-koodia, että samalle riville kirjoitetut akselien siirtymät suoritetaan yhtäaikaaisesti. Kirjoitusasu on rajattu yksinkertaisemman liikkeen vuoksi muotoon, jossa jokainen rivi tulee sisältää X- sekä C-akselin siirtymän, huolimatta siitä onko siirtymä 0mm. Rivi tulee myös aina aloittaa X-siirtymällä. Muilta osin ohjelmakoodin kirjoitus on vapaata, käyttäjä voi syöttää käytännössä rajattomasti akselisiirtymiä, joiden arvo voi olla mitä vain, kunhan ne pysyvät akselien päätyrajojen sisäpuolella.

Liikeradassa ensimmäinen siirto on asemasta 1 asemaan 2, jolloin terä siirretään hampaan keskipisteestä hampaan nurkkaan. Tämä tapahtuu pitämällä X-akselia liikkumattomana, sekä siirtämällä C-akselia noin millin päähalkaisijan kehältä mitattuna. Liikkeissä 2-3 sekä 3-4 terä siirretään hampaan sivua pitkin hampaan pohjaan, mutta johtuen kyseisen pienihammaslukuisen hammaspyörän pyöreähköstä hammasprofiilista, on siirtymä muodostettu kahdesta osasta, joissa akselien liikkeiden suhde eroaa toisistaan. Tällöin muodostettavan viisteen suuruus ei silminnähtävästi vaihtele hampaan kylkien, pohjan sekä kärjen välillä. Esimerkissä pyöreyttä mukaileva kulma on sijoitettu X-akselin suunnassa hampaan kärjen sekä pohjan väliseen keskipisteeseen, sekä C-akselin arvoa muutettu siten, että liikkeessä 2-3 on akselien siirtymien välinen kulmakerroin 0,24, liikkeen 3-4 kulmakertoimen ollessa 0,16.

Mikäli terän halutaan seuraavan kappaleen pyöreää muotoa tarkemmin, voidaan liike jakaa käytännössä rajattomaan lukumäärään osia. Liikkeessä 4-5 siirrytään hampaan pohjaa pitkin hampaan toiselle kyljelle, jolloin terä pysyy jälleen X-akselin suuntaisesti liikkumattomana, C-akselin liikkeessä noin 2mm. Liikkeet 5-6-7 ovat identtisiä 2-3-4-liikkeiden kanssa sillä erotuksella, että X-akseli liikkuu päinvastaiseen suuntaan, kappaleesta ulospäin. Viimeisessä liikkeessä 7-8 terä siirretään hampaan kulmasta takaisin hampaan keskipisteeseen liikkeen 1-2 tavoin.

5.4.2. Inkrementaalinen liike

Vaikka liikeradan kirjoituksessa käytetään akselikomentoja X sekä C, liikeradat muodostetaan käyttämällä absoluuttisen käskymuodon sijaan inkrementaalista käskymuotoa, jota käytettäessä tulisi akselit merkata tunnuksin U sekä W. Koska akseleista käytetään kansankielessä aina absoluuttisen liikkeen tunnuksia X, Z sekä C, on G3-sovelluksessakin selvyuden vuoksi pysyttäytytty absoluuttisissa tunnuksissa, sillä käyttäjällä ei ole mahdollisuutta vaihtaa käskymuotoa inkrementaalista absoluuttiseen. Inkrementaalissa liikkeessä siirtymä suoritetaan suhteessa edelliseen positioon, absoluuttisessa liikkeessä taas suhteessa kappaleen keskipisteeseen (Maaranen 2004: 256). Toisin sanoen, mikäli sovellukseen syötetään komento X10, inkrementaalissa liikkeessä siirrettäisiin terä 10mm ulospäin, riippumatta sen hetkisestä positioista. Absoluuttisessa liikkeessä terä siirrettäisiin paikkaan, joka sijaitsee 10mm kappaleen keskipisteestä. Koska G3-sovelluksessa liikerata määritellään yhden hampaan profiilin fyysisten mittojen mukaan, aiheuttaisi absoluuttisesti liike käyttäjälle vain aikaa vievää päässälaskua. Tämä laskenta voidaan suorittaa tehokkaasti sovelluksen toimesta, käyttäjän ilmoittaessa vain tarvittavat inkrementaaliset siirtymät sekä päähalkaisija.

Liikeratojen X-siirtymät kirjoitetaan lisäksi säteen muodossa. Mikäli sovellukseen syötetään käsky X10, terää siirretään fyysisesti 10mm kappaleesta ulospäin, ja liikkeen suorituksen jälkeen X-akselin positio on paikkakoordinaatistossa 20mm aiempaa suurempi, sillä paikkakoordinaatiston X-arvo ilmoittaa halkaisijan, jolla terä sijaitsee. Myös säteen muodossa ilmoitettavien komentojen tarkoitus on säätää asetuksen teossa kuluva aikaa, sillä halkaisijan muodossa ilmoitettavia mittoja varten käyttäjän olisi kerrottava kahdella jokainen hampaan X-akselin suunnassa mitattu mitta.

5.5 Käynnistys

Mikäli kone on ajettu aloituskohtaan, voidaan automaattiajon suoritus käynnistää ”Start”-napista. Tämän jälkeen sovellus suorittaa tarvittavat toiminnot ja palaa takaisin nollakohtaan. Sovellukseen kirjoitetaan myös automaattinen aloituskohtaan palautus, joka suoritetaan aina automaattiajon käynnistyttyä yhteydessä.

Tämä toiminto on välttämätön turvallisuuseikka, sillä käyttäjä on saattanut siirtää akselien positioita manuaalisesti eri kappaleiden automaattiajojen välillä.

5.6 Pysäytys

Käyttäjä voi missä ohjelman vaiheessa tahansa painaa pysäytysnappia. Tämä aiheuttaa liikkeen välittömän pysähtymisen, riippumatta siitä mitä liikeradan vaihetta sovellus on suorittamassa. Turvallisuussyistä terää ei ajeta pysähdysten tapahtuessa automaattisesti nollakohtaan, vaan käyttäjän tulee suorittaa tämä toiminto erikseen.

5.7 Aloituskohtaan palauttaminen

Automaattiajon liikeradan suorittaminen aloitetaan aina aloituskohdasta. Mikäli aloituskohtaan ajoa ei ole suoritettu, ei työstöä voida aloittaa turvallisesti. Aloituskohtaan ajo tapahtuu painamalla automaattiajoikkunan alalaidassa olevaa ”Siirry aloituskohtaan”-nappia. Tällöin sovellus ajaa, mahdollisen törmäyksen estämiseksi, ensin X-akselin kappaleen aloituspisteeseen, joka sijaitsee syötetyn sisäänajomatkan verran kappaleen päähalkaisijan ulkopuolella. Tämän jälkeen nollakohtaan ajo suoritetaan Z- sekä C-akseleille. Akseleiden nopeus nollakohtaan ajossa määritellään automaattiajon pikaliikenopeuden mukaan. C-akselin aloituskohtaan palautus suoritetaan aina vastapäivään. Tämä on asetuksenteon kannalta hidaste, sillä terä saattaa sijaita vain pienen matkan päässä aloituspisteestä, mutta aloituskohtaan ajo suoritetaan päinvastaiseen suuntaan pyörittämällä kappaletta kokonaisen kierroksen verran. Sovellusta tullaankin kehittämään siten, että sovellus laskee aina lyhyemmän tien aloituskohtaan ja siirtää akselia tähän suuntaan.

6. KONEEN TESTAUS

6.1 Virhetilanteet

Virhetilanteita muodostaessa (ylikuorma moottorille, virherajan ylitys) huomattiin, että Granite Devices Oy:n valmistamien servovahvistimien toipumistoiminto ei kaikissa tapauksissa toiminut odotettavalla tavalla. Käytännön merkitystä tällä ei ole, sillä järjestelmän rakenteesta johtuen liikettä ei ole turvallista jatkaa virhetilanteen jälkeen. Käyttäjälle tästä koituu silti tarpeetonta työtä, sillä vahvistimet on uudelleenkäynnistettävä katkaisemalla niiltä logiikan käyttöjännite.

6.2 Koneen mekaaniset ominaisuudet

Jyrsimen mekaaninen vakaus todettiin käsikäyttötestauksella riittäväksi myös automaattiseen työstöön. Käytetty rungon teräspalkkirakenne on osoittautunut kyllin tukevaksi, eikä värinöitä tai huomattavia mittavirheitä ole esiintynyt. Sekä riskisyöttöpöytä että pyöröpöytä osoittautuivat tukeviksi ratkaisuiksi, joita käytettäessä terän viimeistelyjälki oli sileää sekä värinätöntä. X- sekä Z-akseleiden välykset saatiin pienennettyä n. 50-60um tasolle, trapetsiruuvien kitkan silti säilyessä kohtuullisena. C-akselin välykset saatiin käytännössä poistettua kokonaan, mutta johtuen hammaspyörävälityksien marginaalisesta epäkeskeisyydestä, sekä kierukavaihteissa käytössä mahdollisesti esiintyvistä lämpölaajenemisista, välystä tulee lopulliseen kokoonpanoon jättää n. 10 servomoottorin enkooderin askelta. Tämä välykset näkyvät työstettävän kappaleen akselilla 0 asteen, 0 minuutin ja 18,6 sekunnin välyksenä.

6.3 Vahvistimien konfigurointi

Servovahvistimia konfiguroidessa kävi ilmi, että ohjaimet ovat hyvin herkkiä häiriöille. Myös valmistaja oli tietoinen asiasta. Vaikka moottoreiden käyttöjännitelähde oli sammutettuna, sekä johdot vaihdettu paremmin häiriösuojattuihin, konfiguroinnin suorittaminen valmistajan kaapelilla ei aina onnistunut. Ongelma korostui, mikäli ohjainten käyttöjännitelähteenä käytetyn laboratorioteholähteen jännitettä laski lähelle käyttöjännitteen alarajaa. Käyttöjännitteen pysyessä maksii-

miarvon tuntumassa, konfigurointi onnistui useimmissa tapauksissa. Häiriöiden aiheuttamia ongelmia ei kuitenkaan ole esiintynyt käytettäessä G3-sovellusta.

6.4 Sovelluksen toiminta Windows-ympäristössä

Sovellusta suoritettaessa havaittiin tiettyjä käyttöä haittaavia ongelmia. Moottoreiden liikkuaessa raskaiden, visuaalisten toimintojen suorittaminen Windowsissa aiheutti katkonaista liikettä moottoreissa. Tämän vuoksi käyttöliittymän muokkausta ei suositella käytön aikana. Ongelma pystytään välttämään asettamalla sovelluksen käyttöliittymäsäikeen prioriteetti moottoreita liikuttavien säikeiden prioriteettia pienemmäksi, mutta sivuvaikutuksena paikkatiedon päivitys hidastuu, sekä käyttöliittymäkomponenttien, kuten pysäytysnapin, toiminta muuttuu epäluotettavaksi. Manuaaliliikkeen ohjaus jouduttiin vaihtamaan sille alun perin tarkoitettuista näppäimistä, sillä testausvaiheessa ilmeni ongelmia kolmen akselin yhtäaikaisessa siirrossa. Tämä johtui todennäköisesti näppäimistön, tai näppäimistön lukemisen rajoituksista.

6.5 Jännitelähteen toimivuus

Jännitelähteen 1000 VA teho on osoittautunut riittäväksi moottoreiden käyttöön. Sovelluksen suorittaessa automaattiajooa on käytössä kerrallaan vain kaksi moottoria, joten jatkuva tehontarve muodostuu näiden moottoreiden tehontarpeen summasta, joka on kuormituksesta riippuen maksimissaan 600W jatkuvaa, sekä 900W hetkellistä. Käytännössä tehontarve jää reilusti näiden maksimiarvojen alapuolelle. Kuormitettaessa kolmea moottoria yhtäaikaisesti maksimiarvoihinsa käyttöjännitteessä ei myöskään havaittu merkittävää notkahdusta, eikä jännitteen rippe-liarvo muuttunut juuri lainkaan, johtuen hiukan ylimitoitetuista suotokondensaatoreista. Muuntaja tai tasasuuntauskomponentit eivät testin aikana kuumentuneet, vaan niiden lämpötila pysyi n. 34 asteessa, ympäristön lämpötilan ollessa 26 astetta.

7. YHTEENVETO JA JOHTOPÄÄTÖKSET

7.1 Yhteenveto

Ottaen huomioon koneeseen tarvittavien komponenttien hintaluokan, koneen toiminta osoittautui erittäin hyvin toimivaksi. Kustannusarvio ohjauksen osalta on n. 1350e, johon sisältyy:

- Moottorit ja servovahvistimet
- Moottoreiden sekä ohjainkorttien teholähteitä varten tilattu 1000VA muuntaja
- Teholähteisiin tarvittavat komponentit
- Breakout-kortin komponentit
- Sovellusta suorittava tietokone sekä tietokoneeseen liitettävät oheislaitteet
- IP65-luokan sähkökaapit ohjaimia varten
- Järjestelmän lopulliseen kokoonpanoon tarvittavat kaapelit

Koneen runko valmistui n. 8 metristä rakenneterästä, jonka metrihinta oli 9,50e. Lisäksi tarvittiin mm teräs- sekä pronssivaluaihioita vaihteiston valmistamista varten, kuin myös erinäisiä laakereita akseleiden sekä vaihteistojen laakerointiin. Ristisyöttöpöydän, pää- sekä pyöröpöytärunгон, vaihteistojen, sekä karamoottorin yhteiseksi hinnaksi muodostui n. 900e.

7.2 Johtopäätökset

Sovellusta suorittavan tietokoneen ohjaus hiiren sekä näppäimistön avulla ei täysin sovellu työstökonemaailmaan. Ympäristö sisältää kosteutta veden ja öljyn muodossam sekä metallihiukkasia sisältävää pölyä, jotka siirtyvät käyttäjän käsien kautta hallintalaitteisiin ja aiheuttavat toimimattomuutta. Tietokoneelle sekä servo-ohjaimille tulee rakentaa tehokas ilmanvaihto sekä suodatinjärjestelmä, jotta metallipöly ei pääse aiheuttamaan oikosulkuja kalliissa laitteissa.

Ristisyöttöpöydän trapetsiruuvien pronssimutterit saattavat kulua käytössä, jolloin akseleiden välykset kasvavat. Ruuvien pinnankarheus todettiin kuitenkin erittäin sileäksi, joka osaltaan hidastaa muttereiden kulumista. Käytännössä mutterit saattavat kulua koneen käytön alkuvaiheessa, mutta kuluessaan vastaamaan ruuvien muotoja täydellisesti kulumisen hidastuu, tai pysähtyy kokonaan, jonka jälkeen välyksenpoisto voidaan suorittaa entistä tarkemmin, kitkan pienennettyä kulumisen johdosta.

Rinnakkaisportin käyttö moottorinohjauksessa ei tuottanut hankaluuksia, askelten lähetyksen maksiminopeutta rajoitti lähinnä tietokoneen suorituskyky. Lähetetty bittikuvio oli oskilloskoopilla tarkasteltuna virheetöntä maksiminopeudessaakin. Huono puoli rinnakkaisportin käytössä on I/O pinnien rajallisuus. Pelkästään moottoreiden ohjaukseen tarvitaan 6 erillistä output-pinniä, jonka jälkeen data-porttiin jää kaksi vapaata pinniä. Rinnakkaisportin Status-portista voidaan hyödyntää viittä pinniä input-käytössä, control-portista neljää joko input- tai output-käytössä. Nämä rajalliset liitännät tuli ottaa sovelluksessa toiminnassa huomioon, kuten esimerkiksi referenssiajossa, jossa jokaisen akselin referenssikytkin toteutettiin yhden input-pinnin avulla. Liitäntöjen määrä ei rajoita sovellusta sen nykyisessä muodossaan, mutta mahdollisten jatkokehittelyjen, kuten mm. karamoottorin automaattisen käynnistystoiminnon, takia pinnien käytön määrä minimoitiin.

Sovelluksen kehittäminen Windows-ympäristössä sekä C#-kielellä onnistui kaikkiaan vaivattomasti. Tiettyjä ongelmia esiintyi lähinnä moottorinohjaukseen liittyen, mutta oliopohjaisen sovelluksen nopea rakentaminen, yhdessä käyttöliittymäsuunnittelun helppouden kanssa kompensoivat kohdattuja ongelmia. Tietokoneen sijoituttua viimeiseen sijaintiinsa, Windowsissa taustalla suoritettavien ”turhien” sovellusten määrää täytyy karsia, jotta prosessorin aikaa käytettäisiin mahdollisimman tehokkaasti G3-sovelluksen suoritukseen.

7.3 Jatkotoimenpiteet sekä kehitysmahdollisuudet

Ristisyöttöpöydän akseleiden voimansiirrot on tarkoitus muuttaa trapetsiruuveista kuularuuveihin. Tämän muutoksen myötä moottoreiden kuormitusta voidaan vähentää huomattavasti, liikkeiden tarkkuuden myös parantuessa.

Sovelluksen kehitysmahdollisuudet ovat rajattomat. Toiminnan kehittämistä ja toteuttamista olisi voitu jatkaa loputtomiin, mutta ylipitkän työn estämiseksi raja vedettiin tilaan, jossa sovellus täyttää vaaditut kriteerit sekä sisältää tärkeimmät turvallisuuden vaikuttavat suoja-toiminnot. Lähitulevaisuuden kohteina voidaan pitää joitain käyttäjäystävällisyyttä sekä toimintavarmuutta edistäviä muutoksia, jotka eivät merkittävästi vaikuta sovelluksen toimintaan.

Sovelluksen suorittamalle liikkeelle voidaan kehittää servovahvistimien kiihdytys-rampin lisäksi tehokkaammat rampit. Tällä saataisiin suurimodulisten hammaspyörien syöttönopeutta poikittaisliikkeessä nostettua, ilman nopeassa suunnanvaihdossa tapahtuvaa virtapiikkiä ja sitä seuraavaa mahdollista virheilmoitusta.

Työstöratasuunnittelussa voidaan ottaa käyttöön standardipohjainen, tai enemmän standardipohjaisempi, G-koodijärjestelmä, jonka avulla käyttäjä voi muodostaa monipuolisempia liikeratoja. Tällöin yksittäisen hampaan liikkeeseen voitaisiin lisätä esimerkiksi pikaliikesiirtymiä, joita hyödyntämällä saataisiin minimoitua yksipuolisen viisteen työstössä kuluva aikaa, sillä terä voitaisiin siirtää nopeammin hampaan työstettävälle laidalle.

7.3.1. Vinohammaspyörien mittaus

Koska kone sisältää valmiit toteutukset kolmen akselin liikkeille, sovellukseen voidaan jatkossa rakentaa toiminto, joka helpottaa varaosina toimitettujen vinohammaspyörien valmistusta. Koska varaosavalmistuksessa on äärimmäisen harvinaista, että valmistettavasta tuotteesta on saatavilla tarkkoja tietoja piirustusten muodossa, täytyy kappaleiden fyysiset mitat useimmiten määrittellä mittaamalla, joissain tapauksessa jopa arvioimalla.

Yksi näistä on vinohammaspyörien vinouskulma, jonka määrittäminen on paitsi hankalaa, myös aikaa vievää. Vinouden määrittämiseen on kappaleesta ensin arvioitava tai mitattava vinouskulma mahdollisimman tarkasti, jonka jälkeen vierintäjyrsinkoneelle on tehtävä arvioidun vinouskulman mukainen asetus. Kappale on sitten asetettava koneeseen, ja mitattava heittokellolla kappaleen sekä koneen asetuksen välisen vinouskulmien ero. Tämän jälkeen vinouskulmien ero tulee lisä-

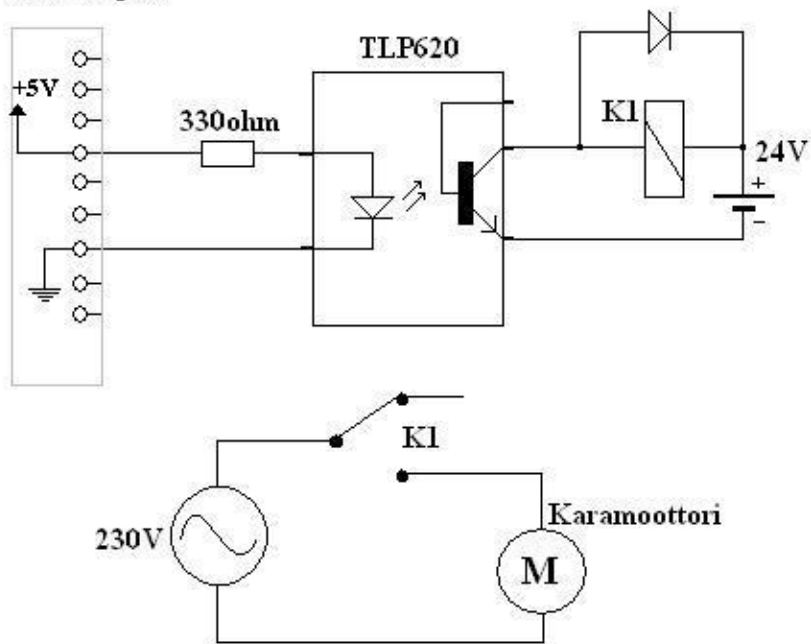
tä tai vähentää koneen asetettuun vinouteen, ja mitata kulmien välinen ero uudestaan. Prosessin toistaminen voi kappaleesta riippuen tapahtua useamman kerran, ja jokaista asetusta varten tulee vierintäjyrsinkoneen differentiaalipyörästön välityspyörät laskea ja asettaa uudestaan.

G3-sovellukseen voidaan hyvinkin vähäisellä vaivalla toteuttaa vinousmittari, jolla vierintäjyrsinkoneella tehtävä kulman iteroiminen voidaan korvata. G3-jyrsinkoneen X-akseliin kiinnitettäisiin heittokello samoin kuin perinteisessäkin tapauksessa, mutta differentiaalipyörästön sijaan vinouskulma sekä hammaspyörän jakohalkaisija syötetään digitaalisessa muodossa sovellukseen, jonka perusteella ohjelma liikuttaa heittokelloa Z-C-akseleiden suunnassa samalla tavoin kuin vierintäjyrsinkone. Tämän jälkeen virhe voidaan lukea heittokellosta, ja tarvittaessa asettaa sovellukseen syötetty kulma eri arvoon. Näin tarkka kulma saadaan selville nopeammin, ja vierintäjyrsinkoneen differentiaalipyörästön välitykset asetetaan kerralla oikeaan vinouskulmaan.

7.3.2. Karamoottorin automaattinen käynnistäminen

Koneen nykyisessä kokoonpanossa käyttäjän tulee huolehtia karamoottorin käynnistyksestä. Toimintaa tullaan kehittämään tältä osin siten, että rinnakkaisportin vapaaseen output-pinniin lähetetään sovelluksesta signaali automaattiajon käynnistyessä. Tällä signaalilla voidaan optoerotin avulla ohjata 24VDC releitä K1, jolla puolestaan ohjataan 230VAC karamoottoria (kuva 34). Karamoottorin lisäksi releellä voidaan ohjata esimerkiksi lastunpoistotarkoitukseen asennettua imuria. Moottorit voidaan tarvittaessa myös varustaa kytkimellä, joka ohittaa automaattisen releohjauksen.

Parallel port



Kuva 34. Karamoottorin automaattiohjaus.

LÄHTEET

Aaltonen, K., Andersson, P., Kauppinen, V. (1997). Koneistustekniikat. Porvoo. WSOY

Airila, M. (1993). Mekatroniikka. Espoo. TKK Offset.

Archer, Tom (2001), Inside C#, Helsinki. Edita Oyj.

Beadmore, Roy (2010). Gear_Efficiency. Päivitetty 30.04.2010 [viitattu 03.10.2010] Saatavilla www-muodossa:
http://www.roymech.co.uk/Useful_Tables/Drive/Gear_Efficiency.html

Fonselius, J., Rinkinen J. & Vilenius M. (1998). Koneautomaatio. Servotekniikka. Helsinki. Oy Edita Ab.

Granite Devices Oy (2007). GDtool 2.0 position mode tuning tutorial for VSD-A. Päivitetty 27.04.2007 [viitattu 03.09.2010] Saatavilla www-muodossa:
http://www.granitedevices.fi/pub_files/docs/GDtool_pos_tutorial.pdf

Keinänen T., Kärkkäinen P. (1998). Konetekniikan perusteet. Porvoo. WSOY

Koskinen, M. (2002), Analogiasuunnittelu. WS Bookwell, Juva.

Logix4u.Net (1999). How Inpout32.dll Works? Päivitetty: 1999.
[viitattu 04.09.2010] Saatavilla www-muodossa:
http://logix4u.net/Legacy_Ports/Parallel_Port/How_Inpout32.dll_works_.html

Maaranen, K. (2004). Koneistustekniikat. Porvoo. WS-Bookwell.

SFS-käsikirja 24 Hammaspyörät (1983). Suomen Standardisoimisliitto ry. Helsinki 1983. Kyriiri Oy.

LIITELUETTELO

LIITE 1 Fluxeon VSD-A, documentation

LIITE 2 VSD-A Breakout board Revision 2

LIITE 3 3 Phase Brushless Servomotors, specification

LIITE 4 AN-10464 1000VA Toroidal Transformers

KÄYTETYT TERMIT JA LYHENTEET

C#	Oliopohjainen C:stä ja C++:sta periytetty ohjelmointikieli. (Archer 2001, xiii)
CAD	Computer Aided Design, Tietokoneavusteinen suunnittelu (Keinänen & Kärkkäinen 1998, 50)
CNC	Computerized Numerical Control, Tietokoneistettu numeerinen ohjaus. (Keinänen & Kärkkäinen 1998, 113)
CPR	Counts Per Revolution, enkooderin askelluksen erottelukyky eli tarkkuus. (Fonselius, Rinkinen & Vilenius 1998, 126)
Enkooderi	Digitaalinen siirtymäanturi, jolla mitataan moottorin suorittamaa kierto liikettä. (Fonselius ym. 1998, 126)
Evolventti	”Käyrä, jonka suoralla oleva piste piirtää, kun suora vierii liukumatta ympyrän (perusympyrän) kehällä.” (SFS-käsikirja 24 1983, 10)
Hammasjako	Hampaiden välinen etäisyys eli jakoympyrän kaaren pituus hammaspyörän jakohalkaisijalla. (Maaranen 2004, 216)
Jäyste	Koneistuksessa muodostunut ei-toivottu terävä materiaalikynsi osittain kiinnittyneenä kappaleeseen.
Moduli	Hammaspyöräsuure joka saadaan jakamalla hammasjako piillä. (Maaranen 2004, 215)
Servo	Mekaanista suuretta kuten positiota tai nopeutta säätävä järjestelmä (Airila 1993, 5.1)
Servovahvistin	Ohjattava teholähde joka säätää moottorin positiota tai nopeutta takaisinkytkennän sekä ohjearvon mukaisesti. (Airila 1993, 5.2.4)
Viisteytys	Kappaleen terävän nurkan koneistaminen tylpäksi.

Fluxeon VSD-A preliminary documentation Ver. 0.60



VSD-A is a precision servo motor drive/controller for permanent magnet AC and DC servomotors with quadrature encoder feedback. Possible control methods include **position**, **velocity** and **torque** modes. Built-in trajectory planner enables simple point-to-point moves as well as continuous synchronous motion tracking.

VSD-A features industry standard optoisolated **pulse/direction**, **quadrature** and **PWM** inputs in addition to a device specific **SPI** (Serial Peripheral Interface) protocol.

Highlights

- High bandwidth field oriented flux vector control
- Position, velocity and torque modes
- AC servo motor phasing without hall sensors
- Quadrature encoder feedback
- Fully optoisolated command I/O
- User upgradeable firmware
- PC software based tuning
- Arbitrary ratio input command scaling and gearing
- Trajectory planner with tunable velocity/acceleration limits
- Acceleration and velocity feedforwards
- Soft recovery from following error
- Configurable motor homing sequence
- Settable absolute position limits eliminates limit switches
- Separate supply terminals for logic and high voltage
- No loss of position on high voltage outage
- Typical efficiency of 95% at full power



Three VSD-A Rev 2 drives

Applications

- CNC machine tools
- Spindle control
- Automation & robotics
- Pick and place machines
- General positioning & speed control applications

Input voltage range	24-200 Vdc
Cont. output current	10.0 Adc
Peak output current	15.0 Adc

AC servomotor	Yes
Brush DC servomotor	Yes
BLDC servomotor	Yes
Linear servomotor	Yes
Stepping motor	Preliminary support

Note: specifications are subject to change without notice

Fluxeon VSD-A preliminary documentation Ver. 0.60



Important notices

This drive has been designed to be operated on **isolated DC power supply** only. Optoisolator isolation distance (creepage) on circuit board is less than 4 millimeters.

A recommended way for **emergency stopping** is to cut HV bus voltage and activating motor brake if possible. Using optoisolated *disable* input may not be enough for emergency stopping.

Drive should be installed in **ventilated** enclosure. Dust filters are recommended when fans are used. The worst case operating temperature should not exceed 70 Celsius degrees (measured from aluminum plate).

Drive **should not be used** in applications where failure or malfunction could lead to danger, large financial loss, health hazard, injury, death, or other unbearable loss.

This document may contain **errors**. When operating with drive, take every precaution you can. Granite Devices do not take any responsibility of damages that may be caused by following or not following this document.

Granite Devices reserves rights to make changes to this document and specifications without notice.





Electrical specifications

Note: for VSD-A rev 1 specifications, download Rev 1 data sheet from www.granitedevices.fi.

	Description	Min.	Typ.	Max.	Units	Notes
Device operating conditions	Logic supply voltage	19		26	Vdc	VSD-A Rev 1 max 22V
	HV supply voltage	24		200	Vdc	Recommended max. 180V
	Logic supply current consumption	200		800	mA	Depends on encoder port load
	HV supply current consumption	0.05		13	A	Depends on motor load
	Operating temperature for currents <12A Operating temperature for currents 12-15A	10 10		70 55	°C	Plate surface temperature
	Humidity	0		95	%	Non-condensing
	Power dissipation	4		115	W	
Encoder	Encoder count rate	0		3	MHz	4X decoding, digitally filtered
	Encoder port input impedance		100		Ohms	
	Encoder supply current	0		500	mA	
	Encoder supply voltage	4.8	5	5.2	V	
I/O (CMD) characteristics	SPI bus baud rate	0		80	kbit/s	
	Output optoisolator source current (Fault & SPI out) source current	0.5			mA	Internal 2200 Ohm pull-up resistor
	Output optoisolator (Fault & SPI out) sink current	1.0	2.0		mA	
	Output optoisolator supply voltage	3		6	Vdc	Voltage against Pin 1
	Optoisolator input current for logic 1	2	3	10	mA	All inputs
	Optoisolator input voltage for logic 1	2		5	V	All inputs
	Optoisolator minimum logic 1 time	1			µs	Step & dir inputs
	Optoisolator minimum logic 0 time	3.5			µs	Step & dir inputs
	PWM mode input frequency	3	5..10	30	kHz	
Motor control characteristics	Motor stall output current			10.0	A	Peak of sine wave
	Motor peak output current			15.0	A	Peak of sine wave
	Peak current duration		0.5		sec	
	Motor output switching frequency		17.5		kHz	
	Effective motor output voltage swing			88	%	Percentage of HV voltage
	Torque control bandwidth		1-3		kHz	Motor dependent
	Feedback (PID) loop sampling frequency		2.5		kHz	
	Efficiency		95		%	At full power
	Motor inductance (per HV supply voltage)	0.01			mH/V	e.g. 200V supply: 0.01mH*200=2mH

Note: socketed step/dir optoisolator (HCPL2531) can be replaced by a faster HCPL2631 if necessary. Default optoisolator provides **step rate** up to 200 kHz and HCPL2631 up to 2 MHz.

Fluxeon VSD-A preliminary documentation Ver. 0.60



Features

Position control

- Relative and absolute position commands
- Infinite motion range
- Configurable 32 bit absolute position limits after homing
- Trajectory planner with settable acceleration & velocity limits
- Soft recovery from error, configurable recovery speed

Velocity control

- Configurable velocity and acceleration limits
- Soft recovery from error
- Configurable input command sensitivity

Command inputs

- Optoisolated step/direction input (step on rising edge)
- Optoisolated quadrature input
- Optoisolated PWM input
- Optoisolated SPI (Serial Peripheral Interface) bus
- Configurable moving average input command filter, averaging time from 1 to 32 PID cycles
- Configurable scaling or gearing ratio
 - Input multiplier from 1 to 32767
 - Input divider from 1 to 32767

PID controller

- Anti-windup design
- 32 bit PID filter
- Acceleration and velocity feed-forwards
- Adjustable anti-dither region with separate PID gains
- Adjustable moving average filter for derivative gain
- 2.5 kHz update rate

Torque controller

- Field oriented sinusoidal flux vector control
- Tunable anti-windup PI torque controller
- Full PWM frequency update rate
- Precise 12 bit current sensing
- HV bus voltage variation compensation
- Power stage deadtime crossover distortion compensation algorithm

Fluxeon VSD-A preliminary documentation Ver. 0.60



Homing controller

- Fully configurable homing sequence forming from the following parts
 - Home switch or hard stop search, settable direction ¹
 - Encoder index pulse search, settable direction
 - Adjustable 32 bit position offset move after homing
- Automatic homing after power-up or manual homing via SPI command
- Configurable torque, acceleration and velocity limits for homing sequence

Fault detection

- Configurable following error limits from 1 to 16383 units (position encoder counts, velocity error or torque error)
- Configurable motion fault detection with 0.2 second response time
 - Sensing of DC motor runaway
 - Sensing of stepping motor stall when encoder feedback is present
 - Sensing of blocked motion
- Overvoltage detection and power stage shutdown to prevent failures caused by regenerative braking current
- Undervoltage detection
- Configurable overcurrent detection and shutdown (single PWM cycle response time, not short circuit safe)
- Internal flash memory error detection
- Input command range error detection
- SPI communication error detection

Other features

- Adjustable AC motor phasing without hall-effect sensors
- Optoisolated fault-output
- Optoisolated disable-input
- User upgradeable firmware

It should be noted that some of these features are new and current release of GDtool configuration software may not support them until the next release.

1 Home switch input available in drive Rev 2 and later

Connectors

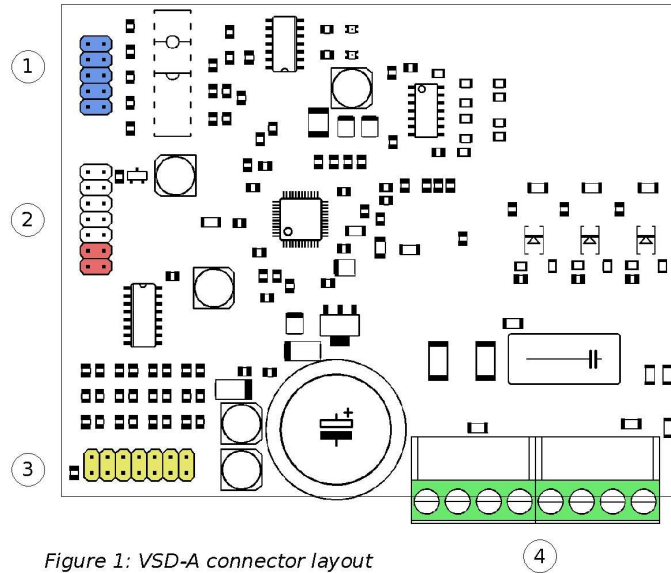


Figure 1: VSD-A connector layout

Name	Location in Figure 1	Description
CMD	● 1	Optoisolated SPI and Step/dir port
JP1	● 2	SPI and CLR jumpers
ENCODER	● 3	Quadrature encoder connector
POWER	● 4	Supply & motor terminals

Fluxeon VSD-A preliminary documentation Ver. 0.60



Power connector

This is a modular (removable) high current connector for **logic supply**, **high voltage supply** and **motor leads**. All ground terminals are electrically connected to same ground.

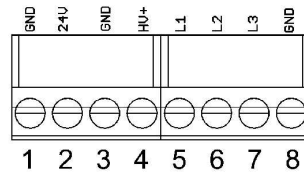


Figure 2: Power connector pin out

Pin (text on board)	Function (split for different motor types)		
1 (GND)	GND (logic)		
2 (24V)	Positive logic supply voltage (recommended 20 VDC)		
3 (GND)	GND (High voltage)		
4 (HV+)	Positive high voltage supply		
5 (L1)	AC motor winding W	DC motor winding 1 or -	Stepper A+ & B+ windings common
6 (L2)	AC motor winding V	DC motor winding 2 or +	Stepper winding A-
7 (L3)	AC motor winding U		Stepper winding B-
8 (GND)	GND (motor)		

Note: motor rotation direction can be reversed by flipping any of two wires in motor windings. For example by connecting V to pin 5 and W to pin 6. Correct rotation direction **must** be set for stable servo operation. If unsure about correct setting, it can be found experimentally by using a freely running motor.



Encoder connector

This is a connector for quadrature encoder with differential line driver (26LS31 or equivalent). Single-ended encoders without inverted (-) outputs are not supported directly. 26LS31 conversion chip can be used to interface to single-ended encoders.

Optional *home switch* should be connected directly between pins 9 and 10. Contact to other conductors and machine ground must be avoided.

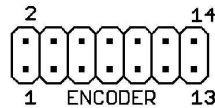


Figure 3: Encoder connector (2x7 0.1" pin header)

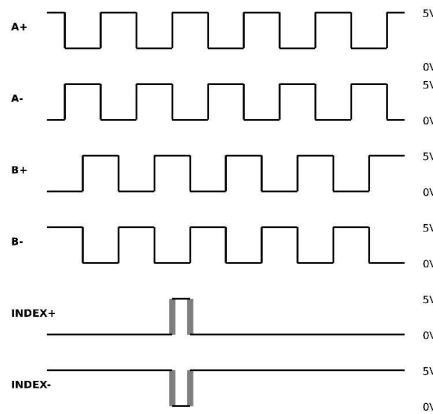


Figure 4: Differential encoder signals

Pin number	Function
1	+5V
2	GND
3	Channel A-
4	Channel A+
5	Channel B-
6	Channel B+
7	GND
8	GND
9	Home switch input
10	Home switch ground (connected to GND)
11	Index+ or C+ or Z+
12	Index- or C- or Z-
13	GND
14	No connection



CMD connector

This is a connector for optoisolated **SPI**, **step/dir**, **quadrature** or **PWM** interface.

Positive and negative terminals of optoisolators are brought to connector. In most cases negative (-) pins are connected to control source ground.

In step/dir mode, step is being taken at rising edge of step signal. Fault output pin FLT+ is logically inverted, logic 0 in FLT+ indicates fault state and logic 1 normal operation. Disable/clear faults input is active on logic 1. For more information, see chapter **Operation in pulse input mode**.



Figure 5: CMD connector (2x5 0.1" pin header)

Pin number (signal name)	Function in SPI mode	Function in Step/Dir and Quadrature mode	Function in PWM mode
1 (OUT-)	Data out -	Fault indicator output - (FLT-)	Fault indicator output - (FLT-)
2 (OUT+)	Data out +	Fault indicator output + (FLT+)	Fault indicator output + (FLT+)
3 (DISABLE-)	Disable/clear faults input -	Disable/clear faults input -	Disable/clear faults input -
4 (DISABLE+)	Disable/clear faults input +	Disable/clear faults input +	Disable/clear faults input +
5 (CMD_VCC)	Output VCC supply	Output VCC supply	Output VCC supply
6 (STEP+)	Clock in +	Step+ or quadrature A+	PWM+
7 (STEP-)	Clock in-	Step- or quadrature A-	PWM-
8	No connection	No connection	No connection
9 (DIR+)	Data in +	Dir+ or quadrature B+	No function
10 (DIR-)	Data in -	Dir- or quadrature B-	No function

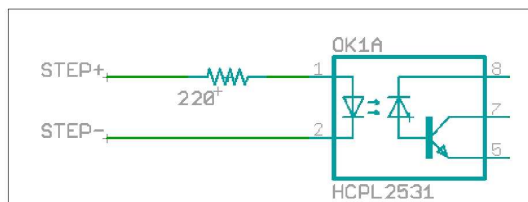


Figure 6: Step & Dir input equivalent circuit

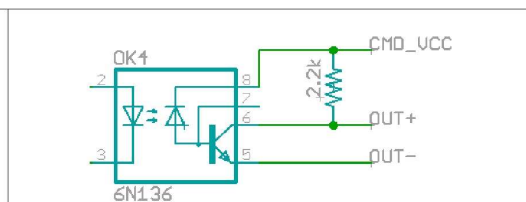


Figure 7: Output optoisolator equivalent circuit

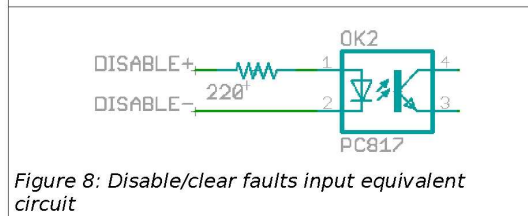


Figure 8: Disable/clear faults input equivalent circuit

Jumpers

This is a combined jumper connector and factory programming connector.

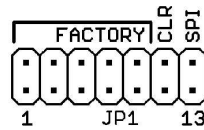


Figure 9: JP1 connector

Jumper settings take effect on power-up.

<i>Jumper</i>	<i>Function</i>
SPI	<ul style="list-style-type: none"> ● Jumper on: step/dir, quadrature or PWM mode is selected ● Jumper off: SPI mode is selected
CLR	<ul style="list-style-type: none"> ● Jumper on: Bootloader (firmware upgrade) mode is selected ● Jumper off: Normal motor control mode is selected
Factory	Factory programming connector, do not use

Lite 1.

Installation notes

Wiring

Typical wiring of driving is presented in figure 10. For detailed information about encoder connector pin out, see the chapter of **Connectors** and compare pin order with the specifications of motor or encoder. If you experience runaway or motor jamming, try reversing any of two phase wires to reverse motor rotation. *Microstepping* feature of tuning software can be used to test proper motor configuration.

Grounding

Encoder, motor and power supply should be grounded to their respectable connector ground pins on drive. *Avoid* grounding these on other ground points like enclosure. For HV bus and logic voltage use separate grounding leads from power supply to each drive.

In typical cases control source ground should be connected to all of the negative terminals (-) of the optoisolated **CMD** connector.

Shielding

Shielded cables are strongly recommended to minimize high frequency electrical interference that can cause errors in operation. Connect cable shields to **ground** pins of the drive.

A shielded metal enclosure is recommended for complete drive electronics system.

If a compliance for CE or UL is required, the system should be characterized as whole in appropriate test arrangements.

Protection

Separate **fuses** should be used for HV bus for every drive. Fuse sizing should be done by estimating peak load power of the motor axis. Using larger fuse than configured peak current is not recommended.

For additional motor protection, fuses can be added in series to motor phase wires. In three phase motor fusing two leads should be enough in most cases and for DC motor one lead will be sufficient protection.

It is recommended to do initial testings with reduced HV bus voltage and with lower current fuse rating.

Cooling

Additional cooling should be used if aluminum plate temperature rises over 70° Celsius during hard use. Improved cooling can be achieved generally by two ways:

- Adding forced air flow by using a dust filtered fan
- Mounting a heat sink on back side of drive's aluminum plate

The most efficient cooling can be achieved by combining the both methods.

See mechanical drawings for heat sink mounting hole pattern.

Warning: when mounting heat sink, take care screws are not too long to reach circuit board through aluminum plate. Leave at least 3 mm isolation distance between the circuit board and screw ends.

Fluxeon VSD-A preliminary documentation Ver. 0.60

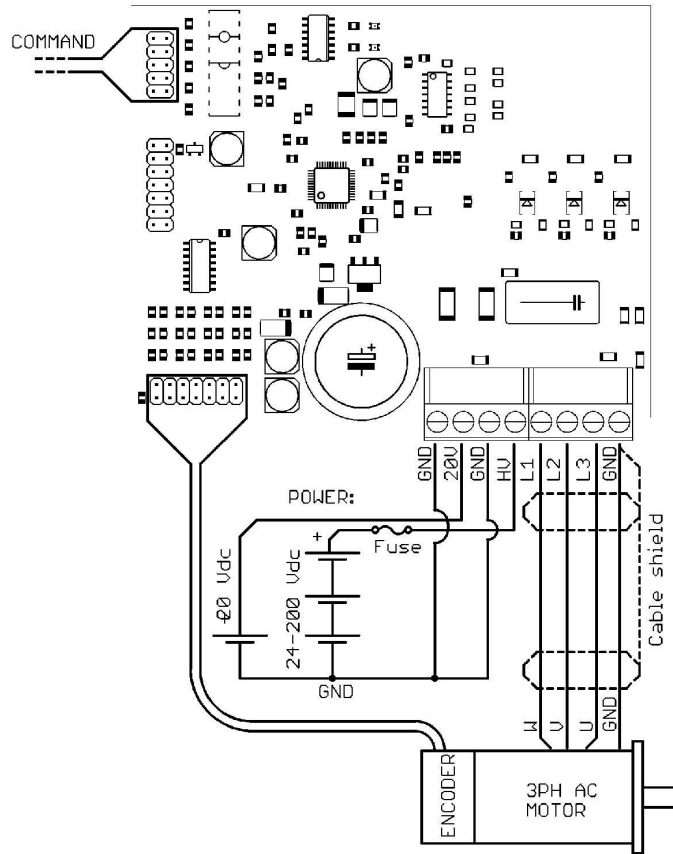


Figure 10: Typical wiring of 3 phase servo motor



Power supply

VSD-A runs on unregulated **isolated** power supply which means that there is no galvanic or conductive connection between the AC line and DC bus. A linear transformer based PSU is preferred over switching mode power supplies (SMPS) for servo systems since transformers are capable of delivering high peak output power just like motors are.

For logic power, a separate regulated power supply is required.

Warning: This is only a very brief guide of powering the system in basic cases. An experienced electrician should always be consulted when designing or building power system.

Pay attention to *RMS* and *peak* terms in this text. Mixing these can lead to unpredictable results.

Basic guide for sizing linear PSU components

In short, PSU should be sized so that it does not overload or overheat during any condition in machine use. Since servo systems typically have greatly varying load, it might be necessary to find effective power consumption by measuring *RMS power* consumption of the system during at least 10 second period of heavy use.

If measurements can't be done, then maximum RMS power load can be estimated roughly by summing rated power values of motors in the system. However, in typical setups power consumption is significantly less than summed motor power since required power is directly proportional to actual produced Speed \diamond Torque.

Transformer

Transformer can be selected after RMS power need is determined. One should choose transformer with some safety margin since VA rating of transformers do not equal to RMS watts in linear PSU. For example, if RMS power consumption is 200 Watts, then using of at least 300 VA transformer is recommended.

Transformer primary voltage should match with the voltage of AC mains network of your area. Secondary voltage should be about 1.41 times *smaller* than desired DC bus voltage. To convert DC bus voltage to transformer secondary voltage, use equation

$$U_{secondary} = \frac{U_{DC}}{1.41}$$

Bridge rectifier

Bridge rectifier should be able to handle *peak current* of rectification. Typically a very high peak currents can be present during power-up and during motor peak loads. Use at least safety factor of 4 when choosing rectifier current rating compared to RMS current. Rectifier may need cooling to prevent overheating damage.

Capacitors

VSD-A HV bus accepts unregulated power up to 20% ripple voltage. To solve required power supply capacitance need, use equation

$$C = \frac{I_{load} * T}{U_{ripple}}$$

where C is required capacitance in Farads, I_{load} is the *peak load current*, T cycle time of rectified voltage and U_{ripple} is the desired maximum ripple voltage.

Fluxeon VSD-A preliminary documentation Ver. 0.60



Calculation example

1. If you have 70 Vdc DC bus voltage, then $70 * 20\% = 14$ V ripple (U_{ripple}) voltage is allowed.
2. If you are using full-wave rectifier for 50 Hz mains voltage, then cycle time T is 0.01 seconds
3. If your peak power load is 500 Watts, then I_{load} becomes $500W/70V = 7.2$ A
4. By substituting these values in equation above, the minimum required capacitor value becomes

$$C = \frac{I_{load} * T}{U_{ripple}} = \frac{7.2A * 0.01s}{14V} = 0.00514 Farads \approx 5000 \mu F$$

Notice that capacitor voltage rating should be at least 20% more than rectified DC bus voltage.

Warning: capacitor voltage can reach drive's maximum DC bus voltage (200 Vdc) during motor braking because of regenerative current flowing back to PSU. A braking resistor circuit in PSU is recommended when significant loads are driven by motor.

Fuses

Use slow-blowing fuses that can withstand the peak currents required by the circuitry after fuse.

Example circuits

Following figures show simplified PSU cases. Line filters may be required before AC input to comply with local EMI regulations.

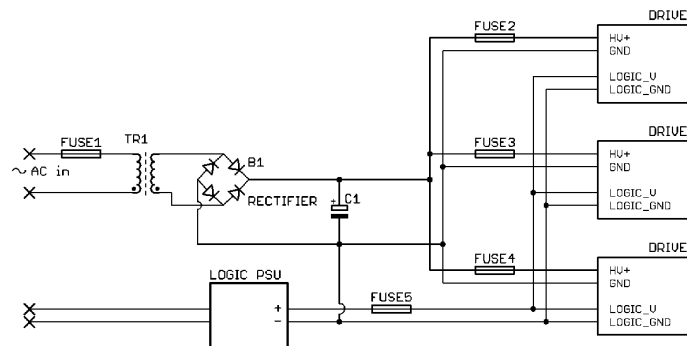


Figure 11: Simple transformer based linear PSU up to about 500 Watts

Drive can be powered also by a switching mode power supply (SMPS). A diode (D1) and capacitor (C1) are required to prevent regenerative current from flowing back to SMPS.

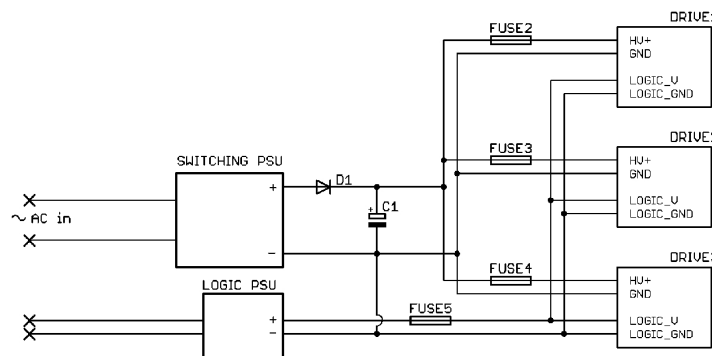


Figure 12: Switching mode power supply (SMPS) circuit



Status indicators

The VSD-A has two LEDs (**green** and **red**) which have various sequences described in table below. Read sequences from left to right and note that all patterns are repeating.

No	LED sequence (time →). Repeating approx. once a second.	Sequence description	Status	Motor output on
1		Green blinking	AC motor phase search or drive init	Yes
2		Green on	Enabled and running	Yes
3		Green flashing fast	Recovering from follow error	Yes
4		Red, green, green repeating sequence.	Following error	No
5		Two short red and two green flashing sequence	Motion error. Motion stalled or encoder (wiring) failure	No
6		Short green flashes	Disabled by user	No
7		Green on and short red flashes	Input motion command range error	Yes
8		Green blinking and red on	Init or AC motor phase search failed. Possibly too low voltage.	No
9		Green and red blinking simultaneously	Other fault, get details via SPI bus. Possibly over current fault.	No
10		Two red and green flashes simultaneously	Over voltage or under voltage fault	No
11		Red and green alternating constantly	Communication error, invalid command or invalid command parameter. Check cabling & jumper settings.	No
12		Red and green alternating very fast and pausing	Internal error, contact us for support. See also #11.	No
13		Red on and green blinking in short bursts	Drive stopped until next power cycle. Occurs after motor type changed in configuration or motor not configured.	No
14		Green flashing off	Drive ready for firmware update	No
15		Green on and red flashing twice per second	Firmware upgrade failed, cycle power and try again	No
16		Red blinking slowly (0.5 Hz)	Memory checksum error, install upgrade firmware again or contact us if problem stays	No

Controller diagram

VSD-A implements a position mode controller by utilizing PID error amplifier with encoder feedback. A greater performance has been achieved by utilizing separate high-bandwidth vector torque controller and feedforward paths.

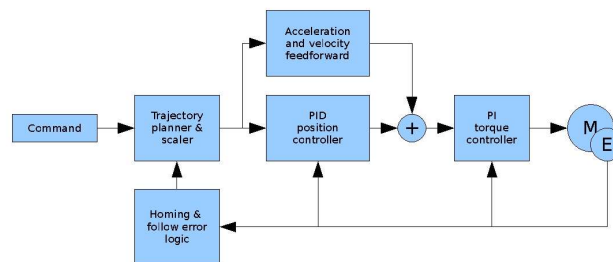


Figure 13: Simplified controller diagram

Operation in pulse input mode (Step/Dir, Quadrature, PWM)

In many cases various pulse input modes are preferred methods of control because of their wide availability and simplicity. In this chapter operation of pulse mode is discussed in detail. Please refer to chapter **Connectors** about pin-out of optoisolated pulse interface.

Setup

After drive configuration & tuning, a jumper **SPI** must be placed before logic supply power-up to enter in pulse input mode. After successful power-up and initialization drive starts following pulsed input from the optoisolated interface.

Powering up

Powering up logic and HV bus can be done in any order or simultaneously. If logic power is being connected first, drive will wait for HV bus rise before initiating motor control.

Delaying power-up

Drive start-up can be delayed by driving logic 1 value to **disable input** while power supplies are switched on. Drive begins initialization after input value is released to logic 0. Logic 0 should be active during whole initialization process (i.e. while green led *not* constantly on). If drive gets interrupted by disable signal during initialization process, an **init fault** condition will occur. Init fault can be cleared only by power cycling.

Faults

In fault condition, optoisolated **fault out** gets activated (CMD connector **FLT+** pin is logic 0). This signal can be used to inform control source about fault condition or, for example, to activate an external alarm. Fault output will return to its non-active state after faults have been cleared.

Clearing faults can be done by cycling logic value 1 in **disable input**. In no-fault condition OUT+ pin is in logic high state.



Notes about PWM mode

In the PWM mode, drive's target value is controlled by input signal duty cycle (percentage of logic 1 time against whole cycle time period). PWM range **0%...100% equals absolute command value range of -16384...16384** in all operation modes (position, velocity or torque). 50% duty cycle equals command 0.

User may adjust PWM **sensitivity** by adjusting input scaling (multiplier/divider) just like in other modes. User may also adjust PWM **offset** to match control source PWM to drive PWM.

0% and 100% duty cycles are theoretical references only and in practice cannot be used. Drive sets output command value to zero if PWM input is not detected (i.e. no logic transitions pass through optoisolator). To stay in **valid duty cycle range**, limiting input to 10%...90% range is recommended.

PWM mode is a digital equivalent of standard **analog input** but has benefit of optical isolation and lower sensitivity to noise. It is possible to convert standard +/- 10V analog signal to PWM with the help of a simple external circuit.

Differences between drive revisions

Determining drive revision

In Rev 2 and later the revision number is marked on circuit board edge with clear white text. In Rev 2 the identifier text reads "**VSD-A Rev 2**". Rev 2 and later also has a black anodized back plate while Rev 1 has bright aluminum plate.

Changes from Rev 1 to Rev 2

New features in Rev 2

- Faster output optoisolator, CMD_VCC input added in CMD connector
- Electrically isolated back plate
- Added home switch input
- 24V logic supply compatibility
- Black anodized back plate instead of uncoated aluminum



Known limitations

VSD-A does not include mechanical *brake control* so it may be unsuitable for applications where axis may fall by gravity if not held at position by a brake or by the motor. However, in brushed DC servo mode this application may be possible since motor position control starts almost instantly after power-up.

For AC motors, VSD-A does phase alignment by driving user specified amount of DC current into windings for several seconds. DC injection is safe when current has been configured to be less or equal than motor rated current. Alignment process may cause a short position jump at power up (typically less than 90 mechanical degrees). The AC servo start-up process takes typically few seconds before final control mode (position etc.) is being activated.

Motor compatibility

This is a short guide for determining whether or not a motor is suitable for VSD-A.

Motor type must be one of the following:

- Permanent magnet brushed DC
- Permanent magnet brushless DC (BLDC)
- Permanent magnet AC

Motor must be equipped with an quadrature/incremental **encoder** with **differential line outputs** (see chapter *Encoder*). However, a *single ended* or *open collector* encoder can be fitted to VSD-A by using a conversion circuit (26LS31 chip) between the drive and encoder.

Encoder notes:

- 2 channel encoder is enough for all motor types
- *Index* channel is not required but is supported for homing
- *Hall* or *commutation* sensors are not needed or used
- Suitable encoder resolutions are from about 100 PPR to 20000 PPR (pulses or lines per revolution)

Motor **voltages** and **currents** can introduce some limitations to motor output speed and torque but *does not cause unsuitability*.

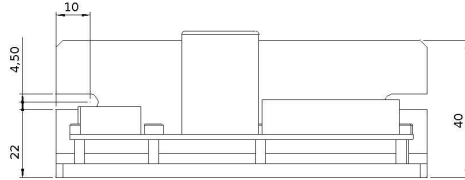
- **Motor voltage** is the limiting factor for maximum **speed**. For example, if you have a 300 VDC brushed DC servo motor and run it at 170 VDC using VSD-A, then you can expect to get a speed of $88\% \cdot 170/300 = 50\%$ of motor's rated speed. 88% comes from VSD-A effective voltage swing at power outputs (see *Electrical specifications*).
- **Motor current** is the limiting factor for maximum **torque**. For example, if you have motor rated for 20A DC and drive's maximum output is 10A DC, then you get 50% of the rated torque.

VSD-A may be unoptimal or even unsuitable for very small motors that have rated current below 1 Amp.

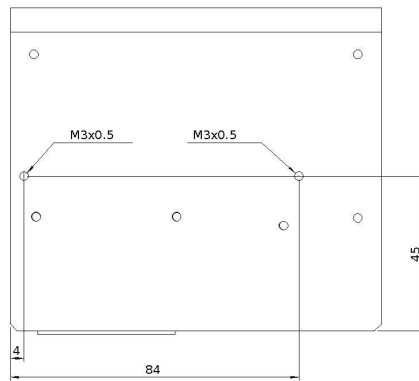
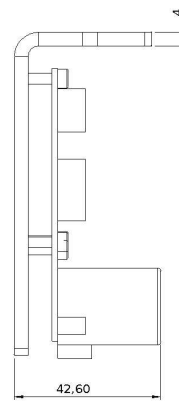
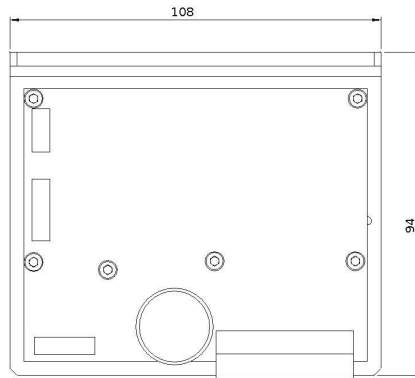
Fluxeon VSD-A preliminary documentation Ver. 0.60



Mechanical drawings



Unit: mm



Heatsink mounting hole pattern (bottom view)

Granite Devices Oy
Opiskelijankatu 4 D 644
FI-33720 Tampere
Finland

Phone: +358 500 555420
<http://www.granitedevices.fi>
VAT code FI20944279

VSD-A breakout board Rev2 for parallel port CNC

documentation Ver. 0.20



Introduction

This is a brief description of a CNC interface board for parallel port CNC step/dir operation of VSD-A. Breakout board serves as easy to use interface between the PC and up to 4 drives. This board is suitable for VSD-A Rev1 and Rev2 drives.

Obtaining boards

This board is **available as bare PCB** from Granite Devices. You can also use provided Cadsoft Eagle or Gerber files to produce breakout board. Several PCB houses accept these files directly for fabrication. See attached *vsd-a-rev2_bom.txt* for bill of materials and needed components.

Note: IC1 must be 7408 or 74LS08 chip. Any other logic family will not work (74HCT, 74AC etc).

Note 2: Use at least VSD-A firmware version 1.14 for optimum operation

Features

- 1-4 axis VSD-A outputs
- Direct ribbon cable connections to VSD-A drives
- 4 limit switch inputs + E-stop input
- USB connector for board power
- Drive enable pin from PC
- Drive fault to PC signal

Pinout

Configure your CNC control software pinout accordingly

<i>Parallel port pin</i>	<i>Function</i>
1	<i>Not connected</i>
2	A axis direction
3	A axis step (rising edge)
4	Z axis direction
5	Z axis step (rising edge)
6	Y axis direction
7	Y axis step (rising edge)
8	X axis direction
9	X axis step (rising edge)
10	Drive fault & E-stop input to PC (active low)
11	X switch to PC
12	Y switch to PC
13	Z switch to PC
14	Drive enable from PC (drives active when high, disabled when low)
15	A switch to PC
16	<i>Not connected</i>
17	<i>Not connected</i>
18-25	Ground

VSD-A breakout board Rev2 for parallel port CNC

documentation Ver. 0.20



Schematics & layout

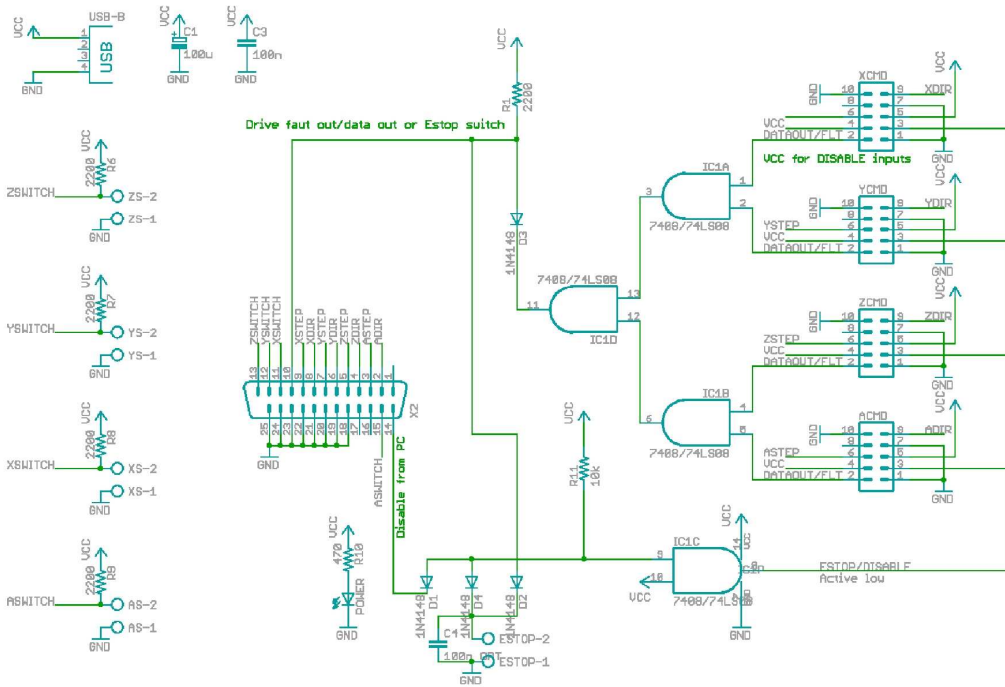


Figure 1: Breakout board schematics

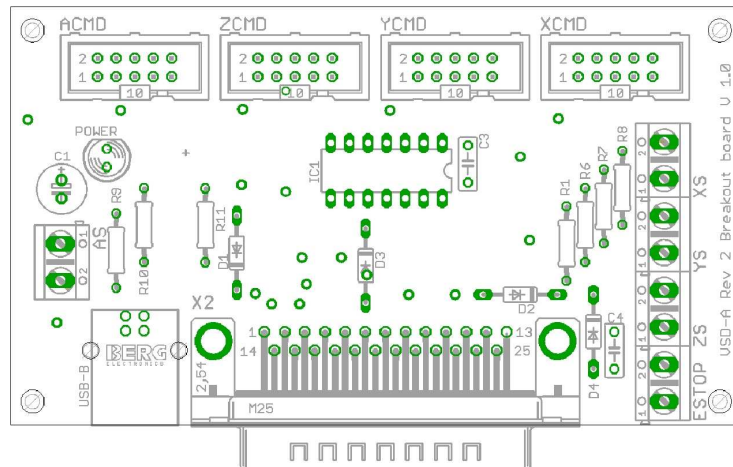


Figure 2: Board layout & component placement

VSD-A breakout board Rev2 for parallel port CNC

documentation Ver. 0.20



Usage

Sample pin configuration for Mach 3 (www.artofcnc.ca). Note: with Mach it is recommended to use at least version 114 of VSD-A firmware.

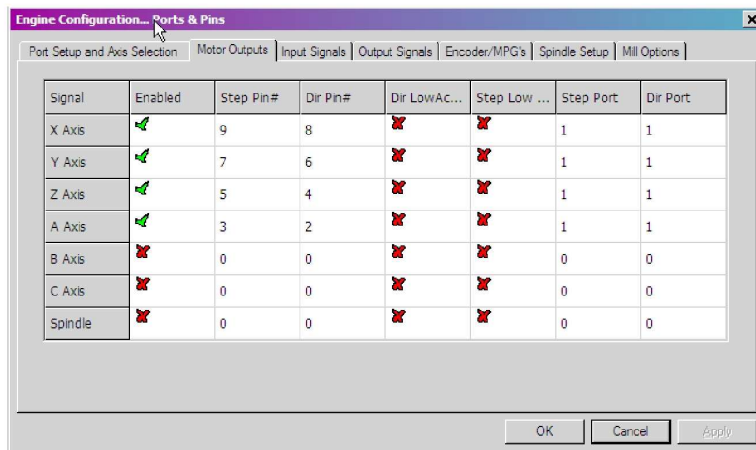


Figure 3: Step/dir outputs

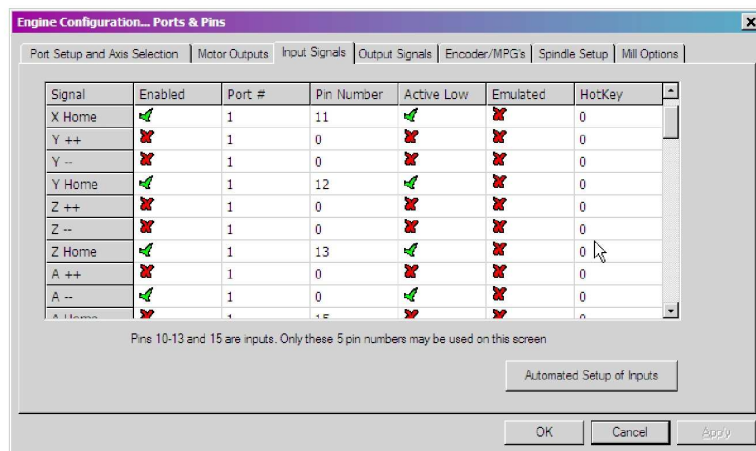


Figure 4: Home switch inputs. Note that home switch inputs can also be configured differently for different kind of setups (for example use as limit switches).

VSD-A breakout board Rev2 for parallel port CNC

documentation Ver. 0.20

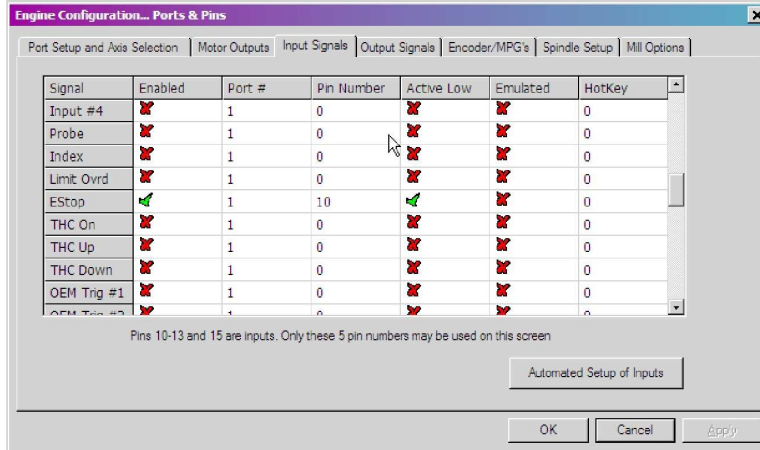


Figure 5: EStop input

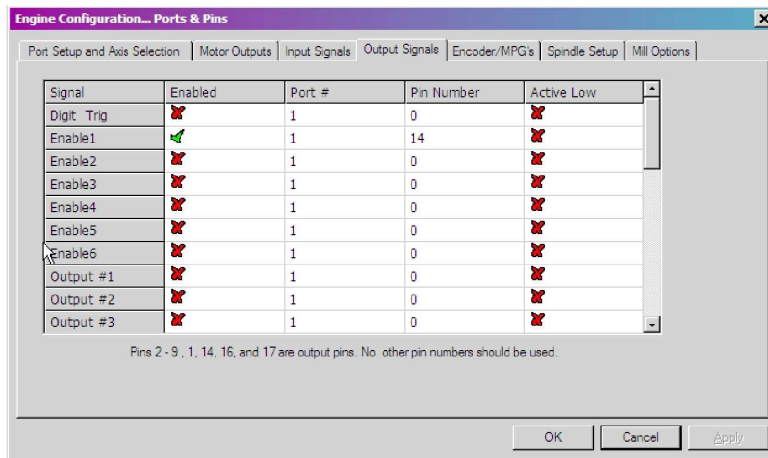


Figure 6: Drive enable output

Important notices

Granite Devices is not responsible for possible damages or losses caused by this document or supplied files. This information or files should not be used in applications where malfunction may result to injury, death or large financial losses.

3 Phase Brushless Servomotors preliminary specification Ver. 0.10

Specifications

Typical values

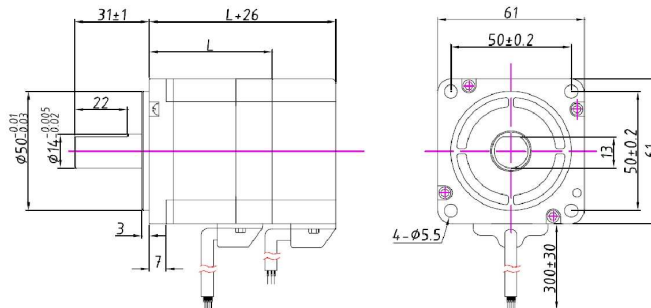
Model	Rated power	Peak power	Rated speed	Cont. torque	Peak torque	Optimal supply voltage	Cont. current	Peak current
BL60-150	150W	450W	3000 RPM	0.5 Nm	1.5 Nm	60-80 Vdc	4.33 Adc	13.0 Adc
BL70-300	300W	900W	3000 RPM	1.0 Nm	3.0 Nm	70-90 Vdc	6.6 Adc	20.0 Adc

Model	Base length (L)	Frame size	Weight	Encoder resolutuon	Phase resistance	Phase inductance	Poles
BL60-150	73 mm	61 x 61 mm	1.25 kg	4000 CPR ¹	0.8 Ohm	1.15 mH	8
BL70-300	90 mm	70 x 70 mm	2.08 kg	4000 CPR ¹	0.74 Ohm	1.5 mH	8

¹) CPR equals effective counts per revolution with 4X decoding

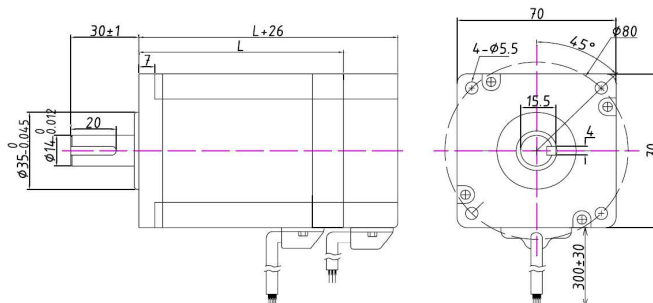
BL60

Dimensions in mm



BL70

Dimensions in mm



3 Phase Brushless Servomotors preliminary specification Ver. 0.10

Encoder & hall sensor wiring

Color	Signal
Black	GND
Red	+5 VDC
Yellow	Hall U+
Yellow/Black	Hall U-
Green	Hall V+
Green/Black	Hall V-
White	Hall W+
White/Black	Hall W-
Brown	A+
Brown/Black	A-
Gray	B+
Gray/Black	B-
Orange	Index+
Orange/Black	Index-

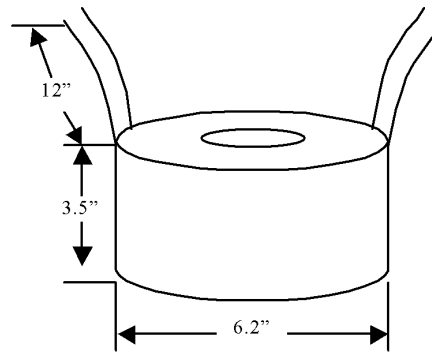
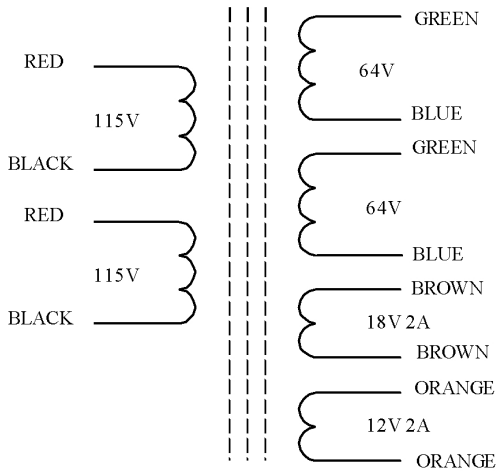
Motor phase wiring

Color	Phase
Yellow	U
Green	V
Blue	W



The 1000VA toroidal transformers are commonly used in the noise sensitive equipment, high-end audio products, stepper motor supply, or servo motor supply. They are specially designed to work on all standard 115V or 230V at 50Hz or 60Hz. These transformers have heavier gauge wires than the normal requirement to avoid the copper lost during the full power output. The dielectric test is more than 3500V in between primary and secondary coils. Please see the test data for short circuit and open circuit.

In most of the cases, this transformer can be output 20% more power from its rating at 60Hz power source without any problem. This transformer comes with 2 rubber pads and all mounting hardware.



Weight = 18 LB

Open Circuit Test (core loss test): TEST CONDITION: Apply variable voltage to primary coils (in parallel). Set voltages 120 and 140VAC at 60Hz. No load on secondary coils. Measure the primary current and input power.	Voltage input	Current input	Power lost
	120V 140V	.07A .14A	6W 13W
Short Circuit Test (copper loss test): TEST CONDITION: Short all secondary coils, and apply variable voltage to (parallel) primary coils. Varies the voltage from 0-20VAC at 60Hz and freeze the voltage at rated primary current.	Voltage input	Current rated	Power lost
	4.4V	8.45A	37W
Load Test (operation test): TEST CONDITION: Input 120VAC 60Hz to the primary coils (in parallel). Output 1, 2, 3, and 4 in parallel to load, and measure voltage and current at different load levels.	Voltage output	Current output	Power output
	67.0V	0.0A	0W
	64.2V 61.0V	7.8A 16.0A	499W 978W