



Sakari Heikkinen

MAASTOJUOKSUSIMULAATTORIN OHJAUSJÄRJESTELMÄN SUUNNITTELU JA TOTEUTUS

MAASTOJUOKSUSIMULAATTORIN OHJAUSJÄRJES-
TELMÄN SUUNNITTELU JA TOTEUTUS

Sakari Heikkinen
Opinnäytetyö
Kevät 2012
Automaatiotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu, tekniikan yksikkö
Automaatiotekniikan koulutusohjelma, projektointi

Tekijä: Sakari Heikkinen

Opinnäytetyön nimi: Maastajuoksusimulaattorin ohjausjärjestelmän suunnittelu ja toteutus

Työn ohjaaja: Heikki Kurki

Työn valmistumislukukausi ja -vuosi: Kevät 2012

Sivumäärä: 62 + 23 liitesivua

Maastajuoksusimulaattorin kehittäminen aloitettiin Oulun seudun ammattikorkeakoulun tekniikan yksikössä vuonna 2011. Tarkoituksena oli, että projektista tulisi usean eri yksikön ja eri osastojen välinen yhteisprojekti. Simulaattorin mekaniikkasuunnittelu aloitettiin keväällä 2011. Syksyllä 2011 lehtori Jaakko Kaski alkoi etsiä projektiin automaatiotekniikan opiskelijaa.

Automaatiotekniikan opiskelijan oli tarkoitus suunnitella kokonaisvaltainen ohjausjärjestelmä maastajuoksusimulaattoriin. Ohjausjärjestelmä kattaa käyttöliittymän videotoistoinen, kallistelurakenteen oikosulkumoottoreita ohjaavan sovelluksen, sähkösuunnittelun ja instrumentoinnin. Tämän opinnäytetyön tavoitteena oli ohjausjärjestelmän suunnittelu ja toteutus.

Työn tuloksena saatiin aikaan järjestelmä, joka sisältää Windows-käyttöliittymässä suoritettavan sovelluksen käyttöliittymineen. Sovelluksella voidaan ohjata kallistelurakenteeseen sijoitettuja, taajuusmuuttajilla ajettavia oikosulkumoottoreita. Windows-sovellus käskyyttä taajuusmuuttajia, lukee anturitietoja tietokoneeseen liitettyjen tiedonkeruulaitteiden välityksellä ja käsittelee mahdollisesti ilmaantuvat häiriötilanteet.

Työn tulokset ovat täysin hyödynnettävissä maastajuoksusimulaattorin jatkokehityksessä. Harmillisesti simulaattorin kallistelurakenteen rakennus ei edennyt suunnittelussa aikataulussa, joten ohjausjärjestelmän lopullinen virittäminen ei ollut mahdollista tämän työn puitteissa. Opinnäytetyö sisältää pohdintaa simulaattorin mahdollisista jatkokehitystoimenpiteistä.

ALKULAUSE

Tämän insinööriyön tilaajana on toiminut Oulun seudun ammattikorkeakoulun lehtori Jaakko Kaski. Työn ohjaavana opettajana on toiminut yliopettaja Heikki Kurki Oulun seudun ammattikorkeakoulusta.

Haluan kiittää suunnittelutyötäni aktiivisesti tukenutta Jaakko Kaskea, kone-tekniikan osastojohtaja Eero Korhosta ja Heikki Kurkea. Kiitokset kuuluvat myös projektissa mukana olleille vaihto-oppilaille Mateusz Frosztegalle ja Kamon Yoshimotolle. Lisäksi haluan kiittää laboratorioinsinööri Markku Timosta simulaattorin laitteiston asennukseen liittyvistä käytännön neuvoista. Kiitos kuuluu myös työn kieliasun tarkastaneelle viestinnän opettaja Pirjo Partaselle.

Erytiskiitos kuuluu luokkatoverilleni Mikko Räisäselle asennustöissä auttamisesta.

Oulussa 30.5.2012

Sakari Heikkinen

SISÄLLYS

TIIVISTELMÄ.....	3
ALKULAUSE.....	4
SISÄLLYS.....	5
SANASTO.....	7
1 JOHDANTO.....	8
2 SIMULAATTORI.....	10
2.1 Aineiston keruu simulaattoria varten.....	10
2.2 Simulaattorin päätoiminnot.....	11
2.2.1 Simulaattorin käyttöliittymä ja videon toistaminen.....	12
2.2.2 Juoksumaton kallistelu.....	13
2.3 Turvallisuus.....	14
2.3.1 Käyttäjistä riippumattomat turvallisuustoiminnot.....	14
2.3.2 Käyttäjistä riippuvat turvallisuustoiminnot.....	15
3 SÄHKÖSUUNNITTELU.....	16
3.1 I/O-listaus.....	16
3.2 Prosessia ohjaavan alustan valinta.....	16
3.3 Moottorin ja vaihteen mitoitus.....	20
3.4 Taajuusmuuttajien mitoitus ja valinta.....	21
3.5 Ohjaus- ja päävirtapiirin suunnittelu.....	23
4 INSTRUMENTOINTI.....	25
4.1 Kallistuanturivaihtoehtojen tarkastelua.....	25
4.2 Kallistusanturin valinta.....	27
4.3 Muu anturointi.....	28
5 OHJELMOINTI.....	29
5.1 Visual Studio 2010.....	30
5.2 VB.NET.....	30
5.3 Simulaattoria ohjaava Windows-sovellus.....	31
5.4 Maastajuoksureitin kallistustietojen tuonti sovellukseen.....	32
5.5 Videotoiston liittäminen osaksi sovellusta.....	34
5.6 Tiedonkeruulaitteiden luku ja kirjoitus.....	35
5.7 Digitaalinen PID-säädin.....	36
5.8 Häiriötilanteiden käsittely sovelluksessa.....	40

5.9 Sovelluksen toimintojen testaaminen	43
6 ASENNUS JA KÄYTTÖÖNOTTO.....	44
6.1 Asennus	44
6.1.1 Taajuusmuuttajien ja moottorien kaapelointi	45
6.1.2 Muiden laitteiden kaapelointi	50
6.2 Käyttöönotto	51
6.3 Toiminnan testaaminen.....	52
6.3.1 Moottorien ohjaus	52
6.3.2 Häiriötilanteiden käsittely	54
7 KEHITYSIDEAT	56
7.1 Käyttöpaneeli.....	56
7.2 Käyttöliittymän toteutus videoilla ja käyttöpaneelilla.....	57
7.3 Juoksumaton nopeuden säätö	57
7.4 PC:n sisällyttäminen osaksi kytkentäkaappia.....	58
8 YHTEENVETO	59
LÄHTEET.....	60
LIITTEET	62

SANASTO

ActiveX	Microsoftin Windows-käyttöjärjestelmässä käytetty Component Object Model (COM) -tekniikka. COM-komponentit ovat uudelleenkäytettäviä ohjelmistokomponentteja.
DAQ-laite	Tietokoneeseen liitettävä tiedonkeruulaite, johon liitettyjen anturien avulla voidaan näytteistää reaali maailman suureita tietokoneella käsiteltävään muotoon.
Hall-ilmio	Ilmiö, jossa elektronien kulkusuunta muuttuu kun johdin, jossa kulkee sähkövirta, tuodaan magneettikenttään kohtisuorassa kenttää nähden.
HDMI	Kuvan ja monikanavaäänänen siirtämiseen suunniteltu liitäntästandardi
I/O	Input/Output, signaalin siirtäminen tai signalointi laitteistojen komponenttien välillä.
N.C.-kosketin	Normaalitilassa kiinni oleva kosketin.
N.O.-kosketin	Normaalitilassa auki oleva kosketin.
PID-säädin	Yksi säätötekniikan perussäätimistä, sisältää suhde-, integrointi- ja derivointiosan.
PLC	Programmable Logic Controller, reaaliaikaisten prosessien ohjaamiseen käytetty pieni tietokone.

1 JOHDANTO

Maastajuoksusimulaattori on Oulun seudun ammattikorkeakoulun keväällä 2011 aloittama projekti. Projekti toteutetaan usean yksikön ja usean tekniikan yksikön osaston yhteisprojektina. Projektin koordinaattorina toimii OAMK:n tekniikan yksikkö.

Simulaattorin tarkoituksena on jäljitellä sitä kokemusta, joka syntyy maastajuoksussa. Kokemuksen jäljittely tapahtuu esittämällä simulaattorin käyttäjälle todelliselta maastoreitiltä kuvattua videokuvaa ja kallistelemalla käyttäjän alla pyörivää juoksumattoa maastoreitin korkeuserojen mukaisesti.

Maastajuoksusimulaattorin suunnittelu alkoi keväällä 2011 mekaanisen kallistelurakenteen suunnittelulla. Juoksumattoa kallisteleavan rakenteen suunnittelutyö valmistui syksyllä 2011. Tämän opinnäytetyön teko alkoi syyskuussa 2011, kun Oulun seudun ammattikorkeakoulun tekniikan yksikön lehtori Jaakko Kaski etsi projektiin automaatiotekniikan opiskelijaa. Simulaattorin mekaanista kallistelurakennetta jatkosuunnittelemaan valittiin puolalainen vaihto-opiskelija Mateusz Frosztega ja moottorinohjausta kehittämään japanilainen Kamon Yoshimoto. Henkilökohtaisesti toin projektiin automaatiotekniikan osaamistani.

Simulaattorin työstäminen ohjausperiaate, instrumentointi, sähköistäminen ja ohjelmointi oli liittyessäni projektiin vielä alkutekijöissään – mitään lähtötietoja näistä ei vielä ollut laadittu. Tämä tarjosi oivan mahdollisuuden tutkia eri ohjausvaihtoehtojen hyviä ja huonoja puolia ja saavuttaa ellei paras, niin ainakin hyvä lopputulos.

Tavoitteena projektissa oli saada aikaan laitteisto, joka sisältää juoksumaton kallistelurakenteineen, maastajuoksuvideota toistavan laitteiston sekä ohjausjärjestelmän, joka kykenee ohjaamaan videon toistoa ja kallistelurakennetta maastoreitin pintoja mukaillen. Tämän opinnäytetyön osuus käsitti kallistelua ja videon toistoa ohjaavan järjestelmän suunnittelun. Järjestelmä kattaa anturoinnin, toimilaitteet, sähkösuunnittelun ja Windows-sovelluksen

suunnittelun käyttöliittymineen. Suunnittelun lisäksi tehtävänä oli ohjausjärjestelmän käytännön toteutus asennuksineen.

2 SIMULAATTORI

Simulaatio on todellisuutta jäljittelevä oppimisympäristö, jossa simuloitua tapahtumia toteutetaan ennalta määritetyllä tavalla. (Laurea ammattikorkeakoulu. 2012.)

Projektissa suunnitellun simulaattorin tarkoitus on jäljitellä maastossa juoksemista. Verrattuna tavanomaiseen juoksumattoon pyrittiin suunnittelemaan laitteisto, joka välittäisi loppukäyttäjälle luonnonkauniit maisemat ja maaston pinnanvaihtelut. Tässä luvussa käydään läpi simulaattorin toiminta ja rakenteet, joista se koostuu.

2.1 Aineiston keruu simulaattoria varten

Simulaattorin monitorilla toistettavaksi videoksi on tarkoitus kuvata maastajuoksureittejä kypäräkameralla (kuva 1). Kypäräkamera mahdollistaa 3D-materiaalin kuvaamisen kahden kameransa avulla. Valitsimme kypäräkameran siksi, että se luo autenttisemman tunteen ihmisen juoksusta verrattuna kameraan, jota kuljetetaan reittiä pitkin esimerkiksi vaunussa. Monitoriksi valittiin suuri 60-tuumainen televisio, joka kykenee toistamaan 3D-materiaalia.



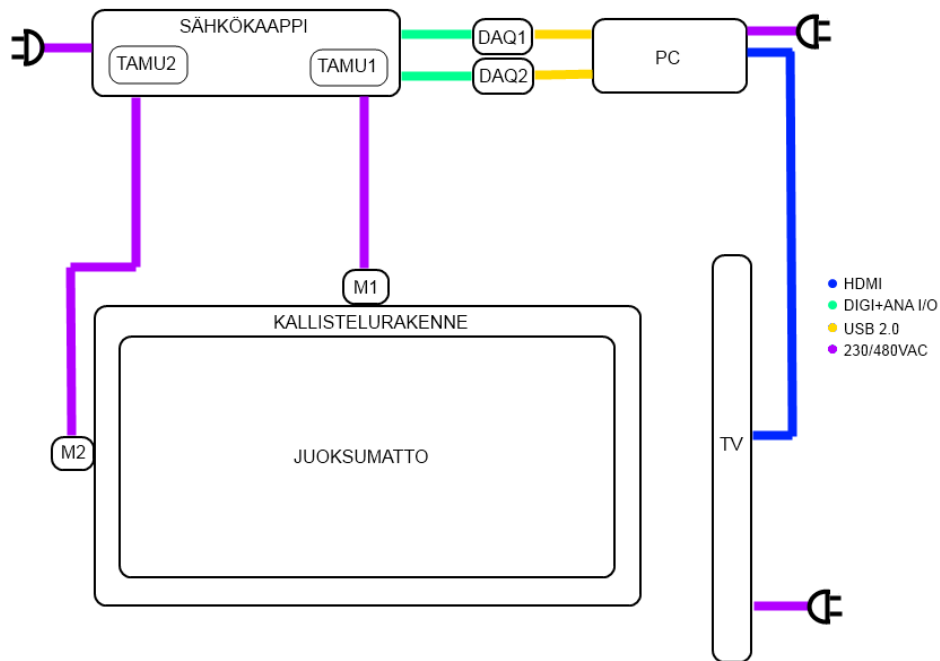
KUVA 1. Go Pro Hero 3D -kypäräkamera

Koska haluttiin mahdollisimman hyvin maaston pinnan muotoja jäljittelevä simulaattori, ei tyydytty vain jälkikäteen videolta katsoen luotuun maaston pinnanvaihtelut sisältävään kallisteluprofiiliin. Kallisteluprofiilin luontia varten rakennettiin neljällä pyörällä liikkuva vaunu, joka sisälsi kiihtyvyyssanturin ja tietokoneen tiedonkeruulaitteistoinen. Vaunu kuljetettiin maastajuoksureitin läpi ja kiihtyvyyssanturin keräämien tietojen avulla saatiin laskettua kallistelukulma reitillä.

Kallistelutiedoista koottiin taulukko, joka sisälsi reitin kallistuksen arvon sekä sivuttais- että eteen-/taakse-suunnassa. Valittu ohjausjärjestelmä käyttää hyväksi tätä kallisteluarvotaulukkoa ohjatessaan kallistelurakennetta.

2.2 Simulaattorin päätoiminnot

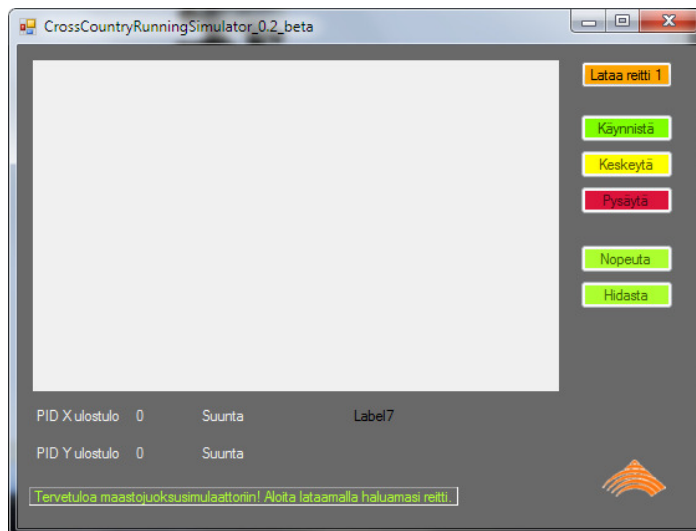
Simulaattorin päätoiminnot ovat maastajuoksuvideon toistaminen monitorissa ja juoksumaton kallistelu reittivideossa esiintyvän maaston mukaiseksi. Simulaattori kykenee kallistamaan juoksumattoa sekä sivuttais- että eteen-/taakse-suunnassa. Kuva 2 havainnollistaa maastajuoksusimulaattorin tärkeimpiä laitteita sekä kaapelointeja. Simulaattorin ydinlaitteena voidaan katsoa olevan tietokone, jossa ajetaan Windows-sovellusta. Se sisältää simulaattorin käyttöliittymän, videotoston ja USB-väyläisten tiedonkeruulaitteiden hallinnan.



KUVA 2. Simulointilaitteiston tärkeimmät laitteet ja kaapeloinnit

2.2.1 Simulaattorin käyttöliittymä ja videon toistaminen

Maastajuoksuvideota toistetaan simulaattoria varten luodun Windows-sovelluksen käyttöliittymässä (kuva 3). Projektia varten hankittu 60” televisio kytketään tietokoneen monitoriksi HDMI-liitännällä. Käynnistettyään simulaattorin Windows-sovelluksen käyttäjä voi valita haluamansa reitin. Tämän jälkeen käyttäjä saa luvan simulaattorin käynnistämiseksi. Simulointi käynnistetään painamalla ”Käynnistä”-painiketta. Simuloinnin käynnistyttyä maastajuoksureitin video alkaa toistua monitorilla täydessä ruudussa. Takaisin käyttöliittymäikkunaan käyttäjä pääsee painamalla hiirellä ”exit fullscreen” -painiketta. Käyttäjä joutuu käyttämään tietokoneen hiirtä näiden operaatioiden suorittamiseen.

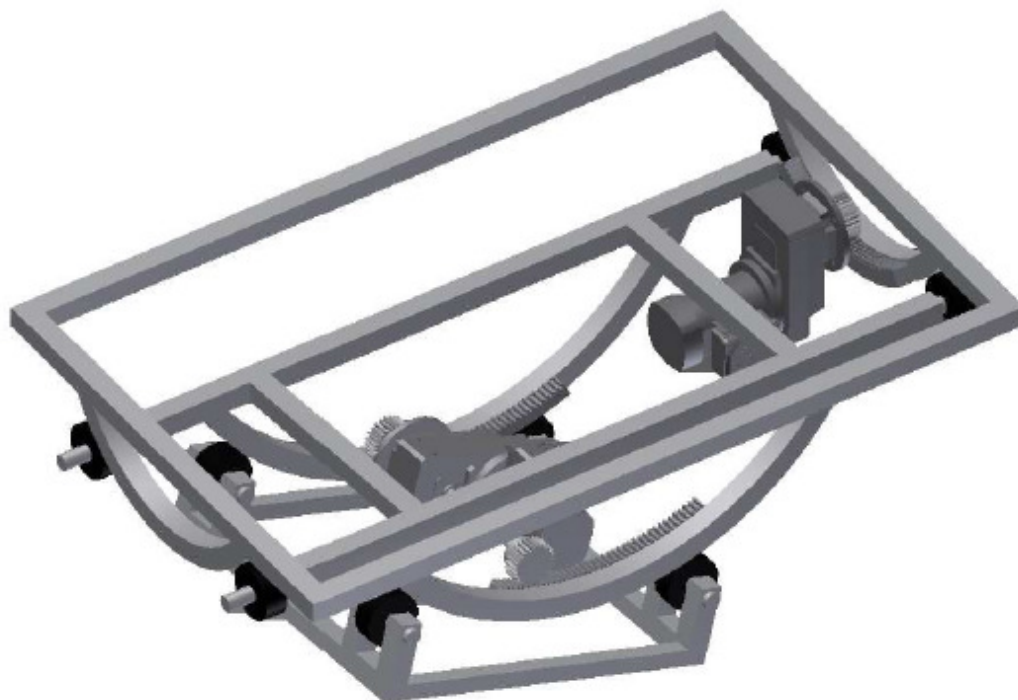


KUVA 3. Simulaattorin käyttöliittymä

2.2.2 Juoksumaton kallistelu

Juoksumaton kallistelua varten suunniteltiin mekaaninen kallistelurakenne, jota voidaan ohjata kahdella alennusvaihteella varustetulla oikosulkumoottorilla. Moottorit on kiinnitetty kallistelulaitteen runkoon. Vaihteiston pääakseli sisältää hammasrattaan jonka välityksellä se siirtää voimaa kallistelurakenteen hammaskiskoon. Suurin mahdollinen kallistuskulma eteen- ja taaksepäin on 13 astetta ja sivuttaissuunnassa 7 astetta. Oikosulkumoottorien kierrosnopeuden säätöä ja pyörimissuuntaa ohjataan kahdella taajuusmuuttajalla.

Kallistelurakenteen suhteen päädyttiin lopulta samantyyppiseen loppuratkaisuun kuin Tomi Laava opinnäytetyössään Maastojuoksusimulaattorin mekaniikkasuunnittelu (Laava 2011). Kuvassa 4 on Laavan mekaniikkasuunnittelun lopputulos.



KUVA 4. Laavan suunnittelema juoksumaton kallistelurakenne

2.3 Turvallisuus

Turvallisuusnäkökohdat olivat tärkeitä koko suunnitteluprosessin ajan. Kun otetaan huomioon, että simulaattorista olisi tulevaisuudessa mahdollista tulla kaupallinen tuote, täytyy loppukäyttäjä pitää koko ajan mielessä. Simulaattori täytyy olla turvallinen kuljettaa, varastoida, käyttöönottaa ja käyttää. Osa simulaattoriin suunnitelluista turvallisuusratkaisuista on käyttäjästä riippumattomia ja osaan käyttäjä voi itse vaikuttaa.

2.3.1 Käyttäjistä riippumattomat turvallisuustoiminnot

Kallistelurakenne, jonka päällä juoksumatto lepää, sisältää paljon sellaisia mekaanisia rakenteita, jotka voivat aiheuttaa vaaran, kun simulaattori on käynnissä ja kallistelurakennetta ohjataan moottoreilla. Tämän vuoksi päätettiin että mekaaniset rakenteet tullaan suojamaan esimerkiksi haitariverhol-

la, jotta kukaan ei pääsisi laittamaan ulokkeitaan rakenteisiin ja näin aiheuttaisi vaaratilannetta.

Maton kallistelu ohjataan kahdella taajuusmuuttajiin liitettyllä oikosulkumoottorilla. Tietokoneella suoritettava Windows-sovellus käskyttää tiedonkeruulaitteiden välityksellä taajuusmuuttajia antaen niille taajuusohjeen ja pyörimissuuntakäskyn. Jos esimerkiksi tietokoneen käyttöjärjestelmä sattuisi kaatumaan, voisi moottorien ohjaus pettää. Tällaisia tilanteita varten kallisturakenteeseen on asennettu jokaiselle sivulle mekaaniset rajakytkimet, jotka laukeavat, jos matto kallistuu liikaa. Rajakytkimen lauetessa taajuusmuuttajien tulojännitelinjaan kytketyn kontaktorin kelan jännite katkeaa, mikä saa aikaan moottorien sammumisen.

2.3.2 Käyttäjistä riippuvat turvallisuustoiminnot

Simulaattorin käyttäjän läheisyyteen on sijoitettu hätäseis-kytkin, jota painamalla käyttäjä voi katkaista taajuusmuuttajan tulojännitelinjaan kytketyn kontaktorin jännitteen, mikä saa aikaan moottorien sammumisen. Moottorit voidaan pysäyttää myös painamalla kytkentäkaapin ovesa sijaitsevaa punaista STOP-painiketta.

3 SÄHKÖSUUNNITTELU

3.1 I/O-listaus

I/O-listauksen suunnittelu aloitettiin simulaattorin toiminnan kuvauksen pohjalta. Toiminnan kuvauksen ansiosta oli helppo arvoida prosessia ohjaavalta alustalta vaadittava I/O:n määrä ja niiden laatu. Kuvassa 5 on simulaattorin ohjausjärjestelmän I/O-listaus.

<u>Digitaaliset sisään</u>	<u>Jännitetaso</u>	<u>I/O purkki / liittimet</u>	<u>Huomioitavaa</u>
1. Releen apukosketin	5 VDC	USB-1208FS / Port A0 / Pin 21	
2. Releen apukosketin	5 VDC	USB-1208FS / Port A1 / Pin 22	
Kallistuksen eturaja	5 VDC	USB-1208FS / Port A2 / Pin 23	
Kallistuksen takaraja	5 VDC	USB-1208FS / Port A3 / Pin 24	
Kallistuksen vasen raja	5 VDC	USB-1208FS / Port A4 / Pin 25	
Kallistuksen oikea raja	5 VDC	USB-1208FS / Port A5 / Pin 26	
<u>Digitaaliset ulos</u>	<u>Jännitetaso</u>	<u>I/O purkki / liittimet</u>	<u>Huomioitavaa</u>
Taajuusmuuttaja 1:n moottorin ajo eteenpäin	24 V	USB-3101 / DIO0	Vaatii jännitteen vahvistuksen 5 VDC -> 24 VDC
Taajuusmuuttaja 1:n moottorin ajo taaksepäin	24 V	USB-3101 / DIO1	Vaatii jännitteen vahvistuksen 5 VDC -> 24 VDC
Taajuusmuuttaja 2:n moottorin ajo eteenpäin	24 V	USB-3101 / DIO2	Vaatii jännitteen vahvistuksen 5 VDC -> 24 VDC
Taajuusmuuttaja 2:n moottorin ajo taaksepäin	24 V	USB-3101 / DIO3	Vaatii jännitteen vahvistuksen 5 VDC -> 24 VDC
<u>Analogiset sisään</u>	<u>Jännitetaso</u>	<u>I/O purkki / liittimet</u>	<u>Huomioitavaa</u>
Kallistusanturi eteen/taakse	0...10 V	USB-1208FS / CH0 / Pin 1	
Kallistusanturi sivuille	0...10 V	USB-1208FS / CH1 / Pin 4	
<u>Analogiset ulos</u>	<u>Jännitetaso</u>	<u>I/O purkki / liittimet</u>	<u>Huomioitavaa</u>
Taajuusmuuttaja 1:n taajuusohje	0...10 V	USB-3101 / VOUT0 / Pin 1	
Taajuusmuuttaja 2:n taajuusohje	0...10 V	USB-3101 / VOUT1 / Pin 29	

KUVA 5. Simulaattorin ohjausjärjestelmän I/O-listaus

Kuvassa 5 näkyvää I/O:n määrää pidettiin ehdottomana miniminä alustaa valittaessa. Oli parempi siis valita alusta, jossa on joustovaraa I/O:n suhteen, koska hyvin todennäköistä on, että simulaattoria tullaan kehittämään tulevaisuudessa.

3.2 Prosessia ohjaavan alustan valinta

Koska olen koulutukseni myötä tottunut käyttämään ohjelmoitavia logiikoita, oli luontevaa alkaa tutkia, millaisia vaihtoehtoja ohjelmoitavista logiikoista löytyisi tähän projektiin. Seuraavat asiat tuli pitää koko ajan mielessä, kun eri ohjausalustoja vertailtiin keskenään:

- analogisen ja digitaalisen I/O:n riittävyys
- videon käskyttämisen liittäminen osaksi ohjausjärjestelmää
- maastajuoksuvideon aikaleiman tuonti ohjausjärjestelmään
- reitiltä kerätyn kallisteludatan tuominen ohjausjärjestelmän käyttöön
- käytössä oleva budjetti.

Huomattiin pian, että tarjolla olevien ohjelmoitavien logiikoiden tarjoamat ominaisuudet eivät vastaa tämän projektin vaatimuksia. Suurimman ongelman olisi tuottanut videotistorajapinnan puuttuminen. Ohjelmoitavien logiikoiden käyttö olisi ollut hyvä vaihtoehto, jos reittivideon toistoa ei olisi tarvinnut hallita ohjausjärjestelmästä. Tällaisessa tapauksessa hyvä vaihtoehto olisi ollut esimerkiksi Siemens LOGO! -pienoislogiikka (kuva 6), koska se sisältää riittävän määrän I/O:ta ja on suhteellisen edullinen. Ajallisesti tämä olisi ollut säästeliäs vaihtoehto, koska ohjelmoitavien logiikoiden valmistajat tarjoavat valmiit työkalut käyttöliittymien ja logiikkakaavioiden tekoon. Lisäksi suunnitteluohjelmistot tarjoavat valmiita automaatiotekniikassa käytettyjä perussäätimiä ja häiriötilanteiden hallitsijoita, jotka voidaan lisätä helposti osaksi omaa logiikkaohjelmaa.



KUVA 6. Siemens LOGO! -ohjelmoitava pienoislogiikka (LOGO!-tuoteluettelo. 2012. Siemens.)

Toisena vaihtoehtona ohjausjärjestelmän alustaksi olivat mikrokontrollerit. Mikrokontrollerit ovat yksinkertaisia tietokoneita, jotka sisältävät prosessorin, muisteja ja I/O-liitäntöjä (Ball & Stuart. 2002, 29.) Mikrokontrollerien etu on

siinä että ne ovat pieniä, edullisia ja niiden ohjelmointitapa – yleisesti mahdollisuus käyttää C-kieltä – tarjoaa joustavan tavan sovelluksen luontiin. Mikrokontrollereiden ongelmaksi muodostui se, että ne on luotu toimimiaan käytännössä itsenäisesti, kuten ohjelmoitavat logiikatkin. Sovellus suunnitellaan tietokoneilla ja syötetään sitten laitteen muistiin. Sovellus kykenee toimimaan ilman erillisen tietokoneen antamia käskyjä. Tässäkin tapauksessa ongelmaksi muodostui se, että ei löydetty rajapintaa moottorinohjauksen ja tietokoneella toistettavan videon välillä. Laitteiston yksi parhaista ominaisuuksista – kyky toimia itsenäisesti – onkin tässä tapauksessa vain haitta, koska se vie ohjausjärjestelmän kauemmaksi Windows-ympäristöstä, jossa video tullaan toistamaan. Eräs tutkituista mikrokontrollerialustoista oli Arduino Uno (kuva 7.)



KUVA 7. Arduino Uno -mikrokontrolleri (Arduino Uno. 2012.)

Kolmantena vaihtoehtona ohjausalustan valinnassa olivat tiedonkeruulaitteet, joita kutsutaan nimellä DAQ. Itse fyysinen laite eroaa ohjelmoitavista logiikoista ja mikrokontrollereista siinä, että se ei sisällä yhtä kyvykästä prosessoria ja muistia. Tiedonkeruulaitteet on suunniteltu ensisijaisesti anturi-tiedon keruuta varten. Tiedonkeruulaite toimii yhteistyössä siihen liitetyn tietokoneen kanssa. Tietokone antaa tiedonkeruulaitteelle käskyjä, ja laite toimii näiden käskyjen mukaisesti (National Instruments. 2012.) Tässä projektissa tiedonkeruulaitteiden käytössä on seuraavanlaisia hyviä ja huonoja puolia:

Hyvät puolet:

- Mallista riippuen riittävä määrä analogista ja digitaalista I/O:ta
- Mahdollisuus käyttää useampaa laitetta rinnakkain
- I/O:n käsittely tapahtuu tietokoneella
- Mahdollisuus valita useamman korkean tason ohjelmointikielen välillä

Huonot puolet:

- Ei suunnitteluohjelmistoa sovelluksen tekoon
- Ei valmiita automaatiotekniikan perussäätimiä
- Ei häiriötilanteiden hallintaa
- Ei käyttöliittymäeditoria
- Monimutkainen konfiguroida
- Vaatii ohjelmointikielten tuntemusta

Listatut tiedonkeruulaitteiden huonot puolet eivät missään tapauksessa kerro siitä, että laitteet olisi huonosti suunniteltu. Ne on ensisijaisesti suunniteltu anturitiedon tietokoneelle keräämistä varten ja mahdollisesti antamaan tilatietoja merkkivalolla. Listatut huonot puolet kertovat ensisijaisesti siitä, ettei niitä ole tarkoitettu käytettäväksi ohjausjärjestelmän alustana.

Kun lopullinen projektissa tarvittava I/O:n määrä ja laatu oli tiedossa, päätettiin tilata kaksi erillistä tiedonkeruulaitetta. Laitteet (kuva 8) ovat Measurement Computingin valmistamia ja mallimerkinnöiltään USB-1208FS ja USB-3101. Kummatkin laitteet voidaan suoraan kytkeä tietokoneen USB-väylään. I/O:n määrän puolesta projektissa olisi selvitty pelkän USB-1208FS:n käytöllä, mutta se ei sisällä 0–10 V:n analogista ulostuloa, jolla taajuusohje viettää taajuusmuuttajille. Tämän vuoksi rinnalle hankittiin myös USB-3101.



KUVA 8. Measurement Computing USB-1208FS ja USB-3101 tiedonkeruulaitteet (Measurement Computing. 2012.)

3.3 Moottorin ja vaihteen mitoitus

Opinnäytetyössään Maastajuoksusimulaattorin mekaniikkasuunnittelu Tomi Laava laski kallistelurakenteen ohjauksessa käytettäviltä moottoreilta ja niihin liitetyiltä alennusvaihteilta vaaditun pyörimisnopeuden ja vääntömomentin (Laava 2011). Jotta kallistelurakennetta saadaan ohjattua riittävällä nopeudella, täytyy kallistelurakenteen hammastankoon vastaavan hammasrattaan kyetä pyörimään noin 39 kierrosta minuutissa. Vaadittu vääntömomentti on 43 Nm. Näiden tietojen pohjalta projektiin hankittiin Watin valmistama moottorin ja alennusvaihteen yhdistelmä. Tähän yhdistelmään hankittiin myös lisätuuletin. Taulukossa 1 on esitelty tekniset tiedot moottorista (Watt 7WAR 72 K4 TH TF), lisätuulettimesta (Watt FLAI Bg71) ja alennusvaihteesta (Watt ASA 46A 72K4 FL).

TAULUKKO 1. Moottorin, lisätuulettimen ja alennusvaihteen tekniset tiedot

Laite	Jännite (AC) / V	Teho / W	Pyörintänopeus / rpm	Virta / A	Taajuus / Hz
Moottori	220-240(Δ) 380-420(Y)	250	1360	1.45(Δ) 0.84(Y)	50
Lisätuuletin	230-277	27	?	0.1	50

	Vääntömomentti / Nm	Teho / W	Pyörintänopeus / rpm	Välityssuhde
Alennusvaihde	66	250	36	37.7

Oikosulkumoottoreiden jäähdytys on toteutettu yleisesti moottorin akselille kiinnitettävällä jäähdytystuulettimella. Tämä tarkoittaa sitä, että mitä nopeammin moottori pyörii, sitä suurempi ilmavirta sitä jäähdyttää. Simulaattorissa moottoreita ajetaan hyvin usein pienillä pyörintänopeuksilla, mikä saa aikaan sen, että moottorit saattavat ylikuumentua, koska jäähdyttävä ilmavirta on niin heikko. Tämän vuoksi moottoreihin hankittiin jäähdytystä tehostavat lisätuulettimet.

3.4 Taajuusmuuttajien mitoitus ja valinta

Kun kolmivaiheiset oikosulkumoottorit oli mitoitettu, tarvittiin niille sopivat taajuusmuuttajat. Taajuusmuuttajien tehtävänä on oikosulkumoottorien kierrosnopeuden ja pyörimissuunnan ohjaus. Taajuusmuuttaja saa taajuusohjeen ja pyörimissuuntatiedon tiedonkeruulaitteelta, jota puolestaan ohjataan tietokoneen Windows-sovelluksesta. Pyörimissuunnan valinta annetaan taajuusmuuttajalle digitaaliviestinä ja taajuusohje 0–10 V analogiaviestinä. Nämä arvot lasketaan Windows-sovellukseen ohjelmoiduissa PID-säätimissä.

Koska johdotus päädyttiin toteuttamaan niin sanotusti perinteisellä tavalla, antoi tämä laajan valinnanvaran taajuusmuuttajan suhteen. Perinteisellä tavalla johdotettu ratkaisu on myös edullinen verrattuna kenttäväyläteknikalla toteutettuun ratkaisuun. Jokaiseen moderniin taajuusmuuttajaan on mahdollista liittää moduuli, jolla tuki halutulle kenttäväyläteknikalle saadaan toteutettua. Esimerkiksi ABB:n taajuusmuuttajiin tarkoitettu Profibus DP moduuli maksaa arvonlisäverottomana noin 200 euroa (ABB ProfibusDP-sovitin ACS550/800. 2012.)

Tärkeän vaatimuksen taajuusmuuttajan valinnalle aiheutti simulaattorin sijoituspaikka. Jotta simulaattori olisi mahdollista sijoittaa mahdollisimman vapaasti, ei olisi ollut järkevää vaatia sijoituspaikan läheisyydestä 3-vaiheista pistokeliitäntää. Tästä syystä vaadimme taajuusmuuttajalta mahdollisuutta kytkeä se syöttöverkkoon 1-vaiheisena. Tämä mahdollistaa simulaattorin si-

joittamisen paikkaan jossa on tarjolla tavallinen 1-vaiheinen pistorasia. Taajuusmuuttajan ulostulo täytyi moottorien takia olla 3-vaiheinen.

Näiden vaatimusten pohjalta päädyttiin valitsemaan Deltan mikrotaajuusmuuttaja. Moottorien vaatiman taajuusmuuttajan koko selvitettiin käyttöohjeesta (Delta Electronics. 2009, 318). Päädyttiin 0,4 kW VFD-E malliin, joka vastasi vaatimuksia (kuva 9). Taajuusmuuttajan tekniset tiedot ovat kuvassa 10 (malli 004).



KUVA 9. Delta Electronics 0,4kW VFD-E mikrotaajuusmuuttaja

Voltage Class	115V		
Model Number VFD-___ E	002	004	007
Max. Applicable Motor Output (kW)	0.2	0.4	0.75
Max. Applicable Motor Output (hp)	0.25	0.5	1.0
Output Rating	Rated Output Capacity (kVA)	0.6	1.0
	Rated Output Current (A)	1.6	2.5
	Maximum Output Voltage (V)	3-phase proportional to twice the input voltage	
	Output Frequency (Hz)	0.1-600Hz	
Carrier Frequency (kHz)	1-15		
Input Rating	Rated Input Current (A)	Single-phase	
	Rated Voltage/Frequency	6	9
	Voltage Tolerance	Single phase 100-120V, 50/60Hz	
	Frequency Tolerance	± 10%(90-132V)	
Cooling Method	Natural Cooling		Fan Cooling
Weight (kg)	1.2	1.2	1.2

KUVA 10. Delta Electronic VFD-E-mikrotaajuusmuuttajien tekniset tiedot

Arvoitiin että moottoriin kytketty kallistelulaitteisto ei kykene saamaan niin paljon liike-energiaa, että taajuusmuuttajaan kytkettävälle jarrutusvastukselle olisi ollut käyttöä. Valittiin kuitenkin taajuusmuuttaja, johon jarruvastuksen voi

jälkikäteen asentaa. Näin varmistettiin, että mahdollinen virhearviointi ei tulisi kalliiksi.

3.5 Ohjaus- ja päävirtapiirin suunnittelu

Ennen kuin sähkösuunnitelmaa tai Windows-sovellusta alettiin tehdä, täytyi olla laadittuna simulaattorin toiminnankuvaus (liite 1). Toiminnankuvaus on yksityiskohtainen selostus laitteiston toiminnasta kaikissa mahdollisissa tilanteissa. Aiemmin laaditut turvallisuusvaatimukset täytyi ottaa huomioon kuvausta laadittaessa. Toiminnankuvaus jaettiin seuraaviin osakokonaisuuksiin:

- normaali käynnistys
- käynnistys häiriön jälkeen
- ajo
- ajon pysäyttäminen kun reitti loppuu
- ajon pysäyttäminen kun tulee häiriö.

Jokainen osakokonaisuus jaettiin rauta- ja sovelluspuolen toimintoihin. Näin saatiin kattava kuva siitä, mitä vaaditaan Windows-sovellukselta ja mitä rautapuolen osilta simulaattorin toiminnan eri tilanteissa.

Päädyttiin liitteen 2 mukaiseen ratkaisuun, joka sisältää kolme ohjausvirtapiiriä ja yhden päävirtapiirin. Ensimmäinen ohjausvirtapiiri sisältää kallistelurakenteeseen sijoitetut mekaaniset N.C. -rajakytkimet ja releen R1 kelan. Tässä piirissä kulkee 24 VAC jännite, joka saadaan sähkökaappiin asennetulta muuntajalta, jonka muuntosuhde on 230 V / 24 V. Piirin ollessa jännitteinen rele R1 vetää.

Ohjausvirtapiiri 2 sisältää lukittuvan hätäseis-painikkeen, releen R2 kelan sekä start/stop-painikkeet. Start/stop-painikkeet on kytketty releen R2 koskettimiin siten, että saadaan muodostettua pitopiiri start-painiketta painettaessa. Tässä ohjausvirtapiirissä vaikuttaa 230 VAC jännite. Kun piiri on jännitteinen, rele R2 vetää.

Ohjausvirtapiirin 3:n kaapelointi kulkee releiden R1 ja R2 apukoskettimien kautta. Kun releet R1 ja R2 vetävät, kytkee se 230 VAC jännitteen kolmannen ohjausvirtapiiriin. Tämä kytkee ohjausvirtapiiriin 3 kytketyn kontaktorin K3 kelalle jännitteen. Sen seurauksena rele R3 vetää. Releen R3 koskettimien kautta kytkeytyy jännite päävirtapiiriin moottoreita ohjaaville taajuusmuuttajille.

Koska haluttiin lisätä laitteiston turvallisuutta, päätettiin käyttää ensimmäisessä ohjausvirtapiirissä 24 VAC suojajännitettä. Tämä jännite vaikuttaa todella lähellä simulaattorin loppukäyttäjää kallistelurakennelman mekaanisissa rajakytkimissä. Vaikka kytkennät onkin suojattu, haluttiin näin varmistaa laitteiston turvallisuus.

Päävirtapiiri ja ohjausvirtapiirit sisältävät niiden kaapeloinnin mukaan valitut johdonsuojakatkaisijat. Valinnassa käytettiin hyväksi kuparijohtimille tarkoitettua johdonsuojavalintataulukkoa(liite 4). Ohjausvirtapiireissä käytettiin pienestä virrasta johtuen 1,5 neliömillimetrin kaapelointia ja 10 A:n johdonsuojakatkaisijaa. Käytännössä ainoat kuormitukset ohjausvirtapiireissä ovat releiden kelat, joiden aiheuttamat kuormat ovat pieniä, luokkaa 3-4 VA. Päävirtapiirissä käytettiin 2,5 neliömillimetrin kaapelointia ja 16 A:n johdonsuojakatkaisijaa. Päävirtapiiriin suositellun johtimen koko löytyy taajuusmuuttajan käyttöoppaasta (Delta Electronics. 2009.)

4 INSTRUMENTOINTI

Simulaattori on anturoitu useilla erityyppisillä antureilla. Tässä käydään läpi näiden anturien valintaan vaikuttaneita asioita ja esitellään käytetyt anturit.

4.1 Kallistuanturivaihtoehtojen tarkastelua

Maastajuoksusimulaattorin tärkein anturointi liittyy kallisteluun. Taajuusmuuttajille ohjeita antavien Windows-sovelluksen PID-säädinten laskemat ohjauksuureet riippuvat oleellisesti siitä, kuinka suuri on kallistelurakenteen kallistelukulma eteen, taakse ja sivuille tietyllä ajanhetkellä.

Tarvittiin anturi, joka kykenee ilmaisemaan kallistusrakenteen kallistuskulman kahdella eri akselilla. Eri anturivaihtoehtoja ja anturien sijoituspaikkoja kallistelurakenteessa on lukuisia. Anturoinnin suunnittelu aloitettiin asettamalla sille tiettyjä ehtoja:

- 0–5 V tai 0–10 V ulostulo
- hyvä resoluutio ulostuloarvossa
- mahdollisimman lineaarinen ulostulo käytetyillä toiminta-alueilla (30 ja 15 astetta)
- hyvä häiriönsietokyky (simulaattorin käyttäjän askellukset juoksumatolla eivät saa tuottaa liiallista häiriötä mittaustulokseen)
- muuttuvien ympäristöolosuhteiden sietokyky (ilman laatu, lämpötila ja valaistus)
- alhainen hankintahinta
- koteloinnissa ruuvikiinnitys.

Listatut ehdot rajasivat pois seuraavantyyppiset anturit: laskurittomat inkrementtianturit, kiihtyvyyssanturit, kiihtyvyyssantureihin perustuvat kallistusanturit ja ultraääneen ja infrapunatekniikkaan perustuvat etäisyysanturit.

Inkrementiaalianturi tuottaa jännitepulsseja, kun akselia pyöritetään. Riippuen mallista, anturi kykenee tuottamaan pulsseja muutamasta kappaleesta tuhansiin yhtä kierrosta kohden. Inkrementtiantureita käytetään yleensä pyörimisnopeuksien määrittämiseen. Inkrementtianturi olisi ollut mahdollista kytkeä tiedonkeruulaitteiden laskurikanavaan, joka kykenee havaitsemaan hyvin suuritaajuisia signaalia (jopa 1 MHz). Kyseisessä tilanteessa ongelmaksi ei olisi muodostunut inkrementtianturien tarkkuus, vaan se, että ne eivät kykene määrittämään pyörimissuuntaansa itse. Tämän vuoksi pyörimissuunnan määrittämiseksi olisi tarvittu lisäksi anturi jolla tunnistetaan kallistelua ohjaavan moottorin pyörimissuunta. Lisää ongelmia olisi tuottanut se, että anturi ei anna absoluuttista kallistusarvoa. Tämän vuoksi simulaattorin kallistelu olisi täytynyt kalibroida aina kun virrat kytketään. Kalibrointi olisi voitu sisällyttää osaksi Windows-sovellusta, jolloin se olisi hoitunut lähes automaattisesti. Ideasta päätettiin kuitenkin luopua, koska se olisi ollut muita vaihtoehtoja työläämpi.

Kiihtyvyyssanturit ja kiihtyvyyssantureihin perustuvat kallistusanturit olivat vahvana ehdokkaana kallistuksen anturointiin. Niistä päätettiin kuitenkin luopua, koska arveltiin juoksijan aiheuttavan askelluksellaan liiallista häiriötä mittaus tuloksiin. Kiihtyvyyssantureihin perustuvat kallistusanturin olisivat olleet ilman tätä ratkaisevaa asiaa vahvana ehdokkaana muunmuassa syystä, että ne tarjoavat mahdollisuuden suorittaa kallistuksen mittausta kahdella akselilla yhtäaikaan. Näin ollen ei olisi tarvittu kuin yksi anturilaitte kallistuksien mittaukseen.

Kosketuksettomat anturit hylättiin, sillä niille ei löydetty hyvää sijoituspaikkaa kallistelurakenteesta eikä kosketuksettomuudesta koettu saatavan etua. Ainoa sijoituskohta näille antureille olisi ollut simulaattorin tasaiset sivut tai simulaattorin alusta. Kummatkin sijoituspaikat olisivat olleet huonoja, sillä etäisyyttä mitattaessa olisi kohteen, josta anturi anturi heijastaa joko valon tai äänen takaisin, hyvä olla mahdollisimman tasainen. Mittausepä-tarkkuutta olisi aiheuttanut kallisturakenteen muuttuva kulma. Vaikka muuttuvan kulman ongelma olisi kyetty ratkaisemaan, olisi ongelmaksi noussut riittävään mittatarkkuuteen kykenevien anturien korkea hankintahinta. Lisäksi mahdol-

linen tilanne, jossa anturin mittausreitille olisi tuotu jokin objekti – esimerkiksi ihmisen käsi – olisi aiheuttanut suhteettoman ison häiriön ohjaukseen.

4.2 Kallistusanturin valinta

Lopulta päädyttiin valitsemaan hall-ilmioon perustuva Vishay 981HE0B4WA1F16 kulma-anturi (kuva 11), joka kykenee absoluuttiseen mittaustulokseen. Anturin sisältämä absoluuttinen ulostulo tekee tarpeettomaksi kallistuksen kalibroinnin, koska Windows-sovellus tietää heti käynnistyessään, missä kulmissa kallistelurakenne on. Kallistusanturin tekniset tiedot ovat taulukossa 2.



KUVA 11. Vishay kallistusanturi

TAULUKKO 2. Kallistusanturin tekniset tiedot

Elektroninen kulma	360°
Lineaarisuus	± 1 %
Käyttöjännite	5 VDC ± 10 %
Virta (maksimi)	16 mA
Analoginen ulostulosignaali	10 - 90 % käyttöjänniteestä

Kulma-anturit kiinnitetään juoksumaton alapuolella olevan tason sivuille. Anturin akseleihin kiinnitetään varret. Pitkittäiskallistelu mittaava anturi kiinnitetään lattiarajassa olevaan alustaan ja sivuttaiskallistelu mittaavan anturi juoksumaton alla olevaan rakenteeseen. Varret ovat teleskooppimalliset, ja niiden kiinnikkeet ovat joustavat, koska niiden täytyy kyetä myötäilemään kallistelurakenteen liikettä.

4.3 Muu anturointi

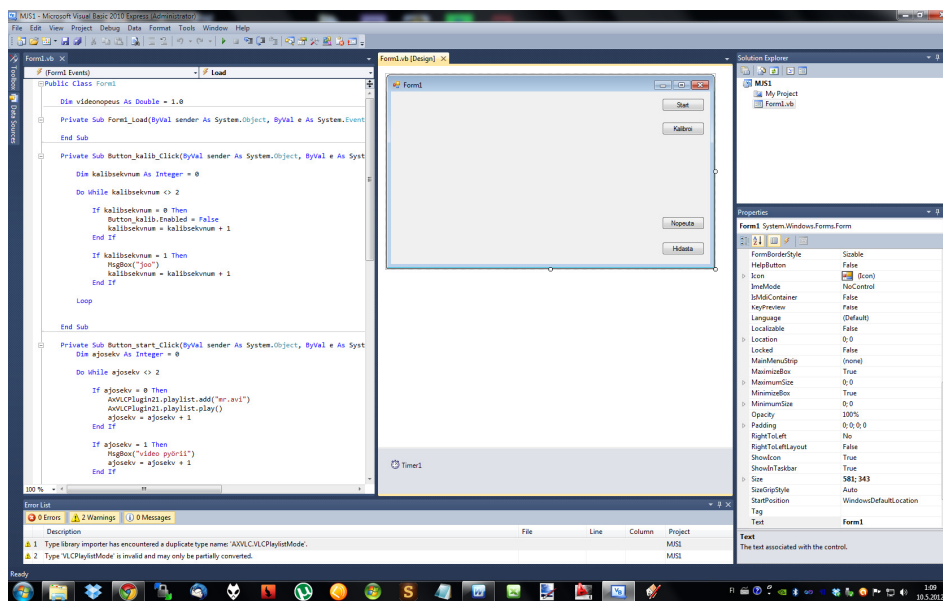
Kallistelurakenteen ääriasennot anturoitiin reed-anturein. Kallistelurakenteen tason jokaiselle laidalle asennettiin magneetit, jotka ilmasevat reed-anturille jos kallistelurakenne on ajanut rajalle. Tässä tapauksessa rajalle ajo tarkoittaa häiriötilannetta, jossa moottorien ohjauksessa on tapahtunut virhe. Virheen seurauksena kallistelurakennetta on kallistettu liikaa. Liika kallistaminen on johtanut siihen, että kallistelurakenteeseen asennettu mekaaninen raja-anturi on lauennut. Laukeaminen johtaa siihen, että ohjausvirtapiirin ke-lajännite katkeaa, minkä seurauksena moottoreita ohjaavien taajuusmuuttajien käyttöjännite on katkeaa.

Reed-anturointia vaaditaan, että Windows-sovellus saisi tiedon siitä mille rajalle on ajettu. Tieto vaikuttaa siihen, miten sovellus käsittelee ilmenneen häiriötilanteen. Jos sovellus saa tiedon, että rajalle ajo on tapahtunut, se jää odottamaan kunnes käyttäjä painaa kytkentäkaapista mekaaniset rajakytkimet ohittavaa painiketta. Kun mekaaniset rajakytkimet on ohitettu, ohjausvirtapiirien releet ja kontaktori sallivat taajuusmuuttajille käyttöjännitteen. Tässä vaiheessa Windows-sovellus ryhtyy ajamaan kallistelurakennetta pois rajalta pienellä nopeudella. Jos sovellus saa siis esimerkiksi tiedon, että rajakytkin on lauennut kallistelurakenteen etupäässä, ajaa se moottoria taaksepäin, kunnes on päästy pois alueelta, jossa mekaaninen rajakytkin painuu pohjaan.

5.1 Visual Studio 2010

Visual Studio on monipuolinen ympäristö sovelluksien kehittämiseen. Sillä voidaan ohjelmoida millä tahansa ohjelmointikielellä. (Franklin 2002.)

Windows-sovelluksen päädyimme suunnittelemaan Microsoftin Visual Studio 2010 -ohjelmointityökaluperheeseen kuuluvalla Visual Basic 2010 Express -ohjelmointityökalulla (kuva 13). Visual Basic 2010 Express sisältää työkalut ohjelman käyttöölyttymän ja itse sovelluksen luontia varten.



KUVA 13. Visual Basic 2010 Express

5.2 VB.NET

Visual Basic on Microsoftin kehittämä BASIC-sukuinen yleiskäyttöinen ohjelmointikieli. Visual Basic pohjautuu 1980-luvulla julkaistuun QuickBASIC-kieleen. Visual Basic -kielen ensimmäinen versio esiteltiin vuonna 1991. Versio 1.0 oli saatavissa sekä Windowsille että MS-DOS-käyttöjärjestelmään. DOS-version kehitys kuitenkin loppui ja seuraavat versiot (2.0-6.0) toimivat enää vain Windowsissa. Vuonna 2002 tapahtui kielessä merkittävä uudistus. Tuolloin julkaistiin Visual Basic .Net (VB.NET), joka on myös kielen nykyinen versio. Se kuuluu Microsoftin .NET-perheeseen. (Wikipedia 2012.)

Nykyisellä Visual Basic .Net -kielellä ohjelmoidaan yleensä käyttäen Microsoftin Visual Studiota, joka on graafinen ohjelmointiympäristö, mutta joka nykyisin on osa Microsoft .NET -perhettä (Wikipedia 2012.)

Visual Basic .Net -kielellä ohjelmoidut ohjelmat toimivat .Net-ympäristössä. Ohjelmien ajamista varten tulee asentaa .NET framework, jonka saa ladata Microsoftin sivuilta. (Wikipedia 2012.)

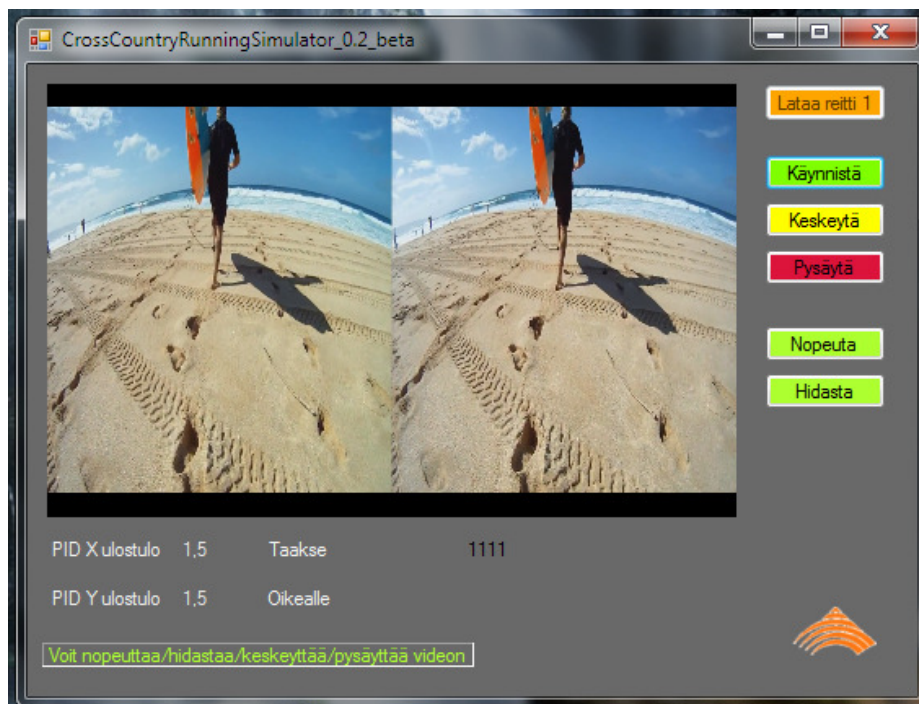
5.3 Simulaattoria ohjaava Windows-sovellus

Simulaattorin sovellus käynnistetään Windows-käyttöjärjestelmässä kuten mikä tahansa muukin tavanomainen graafinen Windows-sovellus. Kuvassa 14 on sovelluksen käyttöliittymä. Tämä näkymä käyttäjälle tulee näkyviin, kun ohjelma käynnistyy. Sovellus sisältää 6 painiketta, joilla käyttäjä voi ohjata simulaattorin toimintaa. Sovellusikkunan alalaidassa oleva teksti ohjeistaa käyttäjää ohjelman käytön eri vaiheissa. Tämä helpottaa erityisesti ensimmäistä kertaa simulaattoria käyttäviä henkilöitä. Video toistaminen on upotettu osaksi sovellusta, kuten kuvasta 14 nähdään.

Simulaattorin käyttäminen aloitetaan painamalla "Lataa reitti 1" -painiketta. Kun ohjelma on ladannut reitin onnistuneesti, käyttäjää informoidaan esiin hyppäävällä infolaatikolla joka kertoo reitin onnistuneesta lataamisesta. Ohjelman kehittämisen tässä vaiheessa reittejä on valittavissa vain yksi kappaletta. Tämän jälkeen informoidaan sovelluksen alalaidassa sijaitsevassa tekstilaatikossa, että käyttäjällä on lupa painaa "Käynnistä"-painiketta. "Käynnistä"-painikkeen painaminen tuo käyttäjän eteen jälleen infolaatikon, jossa kehoitetaan käyttäjää nousemaan simulaattorin juoksumaton päälle, laittamaan 3D-lasit päähänsä ja painamaan infolaatikon "OK"-nappia, kun on valmis aloittamaan. Kun "OK"-nappia on painettu, simulaattori käynnistyy. Video menee automaattisesti koko ruudun täyttävään tilaan, ja kallisteluaitteisto alkaa kallistella käyttäjän alla olevaa juoksumattoa videolla näkyvän maaston pintojen mukaiseksi.

Simulaattorin nykyisessä kehitysvaiheessa käyttäjän tulee klikata video pois kokoruututilasta päästäkseen keskeyttämään/pysäyttämään/muuttamaan videon nopeutta.

Kun video on loppumaisillaan, kallistelurakenne ohjataan takaisin lähtötilanteeseen. Kun video loppuu, Windows-sovellus antaa pysäytyskäskyn tiedonkeruulaitteiden kautta taajuusmuuttajille. Videon loputtua palataan koko ruudun täyttävästä videosta takaisin ikkunoituun käyttöliittymään. Tämän jälkeen ohjelma voidaan joko sulkea tai simulointi voidaan aloittaa alusta ”Käynnistä”-painikkeella.



KUVA 14. Simulaattorin Windows-sovellus

5.4 Maastajuoksureitin kallistustietojen tuonti sovellukseen

Jotta tietokoneella suoritettava Windows-sovellus kykenisi antamaan tiedonkeruulaitteiden välityksellä taajuusmuuttajille oikeanlaisia pyörimissuuntakäskyjä ja pyörimisnopeusohjeita, täytyy ohjelman tietoon tuoda maastajuoksureitiltä kerätyt kallistustiedot. Kallistustiedot on kerätty ulkoiseen

Excel-taulukkotiedostoon, jossa on kaksi saraketta. Ensimmäinen sarake sisältää eteen/taakse-akselin kallistustiedot. Jälkimmäinen sarake sisältää sivuttaissuunnassa tapahtuvat kallistustiedot. Kallistustiedot on kerätty maastoreitiltä tietyllä näytteenottovälillä. Tässä tapauksessa käytettiin 100 ms:n näytteenottoväliä. Tämä tarkoittaa sitä, että taulukossa ensimmäisellä rivillä sijaitsevat kallistustiedot on otettu ajanhetkellä 0 ja toisella rivillä sijaitsevat kallistustiedot on otettu 100 ms myöhemmin ja niin edelleen.

Ulkoisen taulukkotiedoston käytön vaihtoehtona olisi ollut luoda ohjelman sisään taulukkotiedosto ja suoraan syöttää ohjelmamme koodiin käytettävät kallistustiedot. Tämä olisi vaatinut jokaisen kallisteluarvon käsin määrittelyä, joka olisi ollut todella aikaa vievää. Tosin, jos simulaattorissa haluttaisiin käyttää vain yhtä maastajuoksureittiä joka olisi ollut lyhyt ja kallisteludatan näytteenottoväli olisi ollut pitkä, olisi ulkoisen taulukkotiedoston käyttö ollut perusteetonta. Kuitenkin tässä tapauksessa maastajuoksusimulaattoriin on tulevaisuudessa tarkoitus tuoda lisää maastajuoksureittejä ja näytteenottoväli on lyhyt, joten päädyttiin ulkoisen taulukkotiedoston käyttöön.

Työn edetessä havaittiin, että jatkuva ulkoisen taulukkotiedoston lukeminen Office Automation -apuohjelman avulla aiheuttaa epätoivottua viivettä simulaattorin Windows-sovelluksen suorittamiseen. Jotta ohjelma pääsisi simuloinnin aikana mahdollisimman nopeasti käsiksi haluttuun kallistelutietoon, päätettiin, että aina kun ohjelmassa valitaan haluttu reitti painamalla käyttöliittymän ”Lataa reitti” -painiketta, ulkoisesta taulukkotiedostosta tehdään kopio ohjelman käyttöön.

Ulkoisen taulukkotiedoston solujen arvojen luku suoritettiin käyttämällä hyväksi Visual Studio-ohjelmiston Office Automation -apuohjelmaa. Ohjelman toimintaperiaate on, että kun ulkoisen taulukkotiedoston arvojen luku aloitetaan käyttäjän painettua ”Lataa reitti” -painiketta käyttöliittymästä, sovellus käynnistää Windowsin taustaprosessiksi Excel-taulukkolaskenta-ohjelman. Se toimittaa sovellukseen ulkoisen taulukon solujen arvot solu kerrallaan. Aina kun solun arvo on toimitettu sovellukselle, kirjoitetaan tämä arvo sovelluksen omaan, sisäiseen taulukkoon. Kun sovellus on lukenut kaikki ulkoisen taulukkotiedoston arvot, käyttäjälle kerrotaan esiin hyppäävällä infolaatikolla,

että "Reitti ladattu onnistuneesti". Käytännössä tämä yllä kuvattu prosessi kestää noin yhden sekunnin.

Epätoivotun viiveen poistamisen lisäksi edellä läpi käyty prosessi estää sen, että jos ulkoiseen taulukkotiedostoon tehdään syystä tai toisesta muutoksia simulaattorin toiminnan aikana, ei sillä ole vaikutusta simulaattorin suoritukseen.

5.5 Videotoiston liittäminen osaksi sovellusta

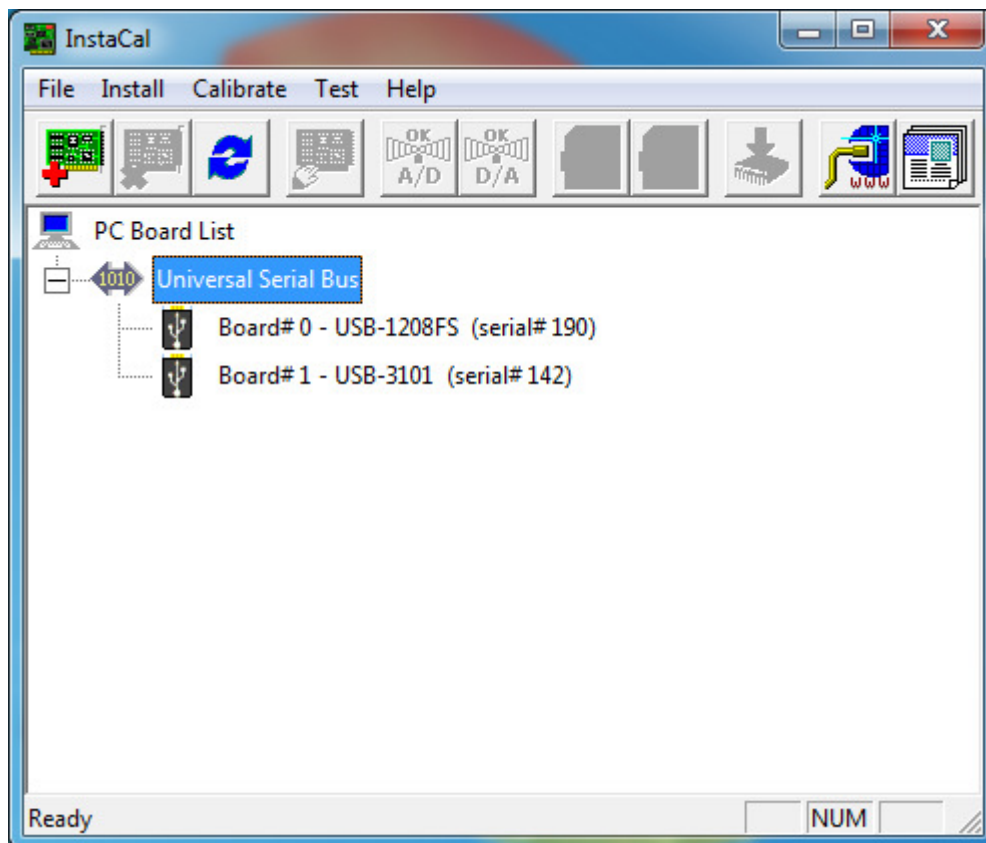
Visual Studion työkaluvalikoimaan kuuluu oletuksena Windows Media Player ActiveX -komponentti. Tämän komponentin avulla voi liittää videotoistominaisuuden VB.NET-sovellukseen. Tämän komponentin ominaisuudet olisivat periaatteessa riittäneet simulaattorille. Sovellukseen upotetulta komponentilta voidaan pyytää videon aikaleimaa ja siihen ladattua videota voidaan ohjata ohjelmasta. Syy sille, että päädyttiin käyttämään VLC-mediatoistinta löytyy videoformaattivalikoimasta: VLC-mediatoistin tukee luonnostaan suurempaa määrää videoformaatteja kuin Windows Media Player. Koska simulaattorissa käytettävän videon tyyppiä ei ollut vielä päätetty, päädyttiin varmempana vaihtoehtona VLC-mediatoistimeen.

Tietokoneelle asennettiin VLC-mediatoistimen versio 1.1.13 ActiveX -komponentteineen. Jotta ActiveX -komponenttia voidaan käyttää ohjelmassa, täytyy se tuoda Visual Studion työkalukirjastoon. Kun ActiveX -komponentti on tuotu työkalukirjastoon, se voidaan liittää raahaamalla sovellukseen.

Tämän jälkeen VB.NET-sovelluksen koodiin on mahdollista liittää ActiveX -komponenttia ohjaavia komentoja.

5.6 Tiedonkeruulaitteiden luku ja kirjoitus

Ennen kuin tiedonkeruulaitteita voidaan käyttää tietokoneella, täytyy ne kalibroida niiden ostopakkauksen mukana toimitetulla InstaCal-ohjelmistolla (kuva 15).



KUVA 15. Instacal-ohjelmisto

Instacal tunnistaa tietokoneen USB-väylään liitetyt Measurement Computingin valmistamat tiedonkeruulaitteet. Ensin tunnistettu tiedonkeruulaite saa järjestysnumeron 0 ja jälkimmäinen järjestysnumeron 1. Kun tunnistaminen on tapahtunut, voidaan tiedonkeruulaitteet kalibroida painamalla "calibrate"-painiketta. USB-1208FS:n tapauksessa I/O:n kalibroiminen vaatii Instacalin opastamien väliaikaisten johdotuksien tekemistä. USB-3101:n kalibrointi ei tätä vaatinut.

Kalibroinnin valmistuttua Instacal luo tiedonkeruulaitteista kalibrointitiedoston. Tiedoston luonti on pakollista, koska Visual Studiossa luotava Windows-

sovellus käyttää kalibroitiedostoa hyväksi, kun tiedonkeruulaitteita luetaan ja kirjoitetaan.

Tiedonkeruulaitteiden lukemista ja kirjoittamista varten täytyy ohjelmassa ensin luoda niistä uudet objektit käyttämällä hyväksi Universal Libraryn luokkia. Määrittelyssä tärkein parametri on tiedonkeruulaitteet erottava Instacalin määräämä järjestysnumero.

Kumpikin käytetty tiedonkeruulaite sisältää digitaalista I/O:ta jotka ovat ohjelmoitavissa. Tämä tarkoittaa sitä, että jokainen käytetty digitaalinen I/O täytyy erikseen asettaa joko tuloksi tai lähdöksi. Alla on esimerkki, jossa USB-1208FS:n digitaalisen I/O:n A-ryhmän ensimmäinen portti konfiguroidaan sisään tuloksi ja koodin viimeisellä rivillä luetaan portin arvo muuttujaan Ekarelearvo. Koska muuttujan tyypiksi on asetettu looginen tila voi se saada vain joko arvon 0 tai 1.

```
Private USB1208 As MccDaq.MccBoard = New MccDaq.MccBoard(1)
```

```
DigisisaanDirection = MccDaq.DigitalPortDirection.DigitalIn
```

```
A_PortNum = MccDaq.DigitalPortType.FirstPortA
```

```
Dim Ekarelearvo As MccDaq.DigitalLogicState
```

```
USB1208.DConfigPort(A_PortNum, DigisisaanDirection)
```

```
USB1208.DBitIn(A_PortNum, 0, Ekarelearvo)
```

Tiedonkeruulaitteiden lukemiset ja kirjoitukset tapahtuvat yllä kuvatun koodin kaltaisesti.

5.7 Digitaalinen PID-säädin

Digitaalinen säädin voidaan toteuttaa esimerkiksi tietokoneella tai ohjelmoitavalla logiikalla. Käytännössä digitaalinen säädin on pala ohjelmointikoodia. (Savolainen – Vaitinen 2001, 70.)

Kuva 16 esittää signaalien kulun digitaalisessa säätimessä. Jos kuvaa peilaan maastajuoksusimulaattorin järjestelmään ovat laitteiden ja signaalien vastaavuudet seuraavat:

$r(t)$ = Säätimen asetusarvo. Sovellus tuo tämän arvon säätimelle taulukosta, joka sisältää maastajuoksureitin kallistusarvot.

$c(t)$ = Kallistuksen mittausarvo. Tämä arvo (0-5v) tuodaan kallistusanturin ulostulosta.

$e(k)$ = Erosuure. Taulukosta haetun kallistuksen asetusarvon ja kallistuksen mittausarvon erotus.

$u(k)$ = Pitopiirille syötettävä analoginen ohjaussignaali.

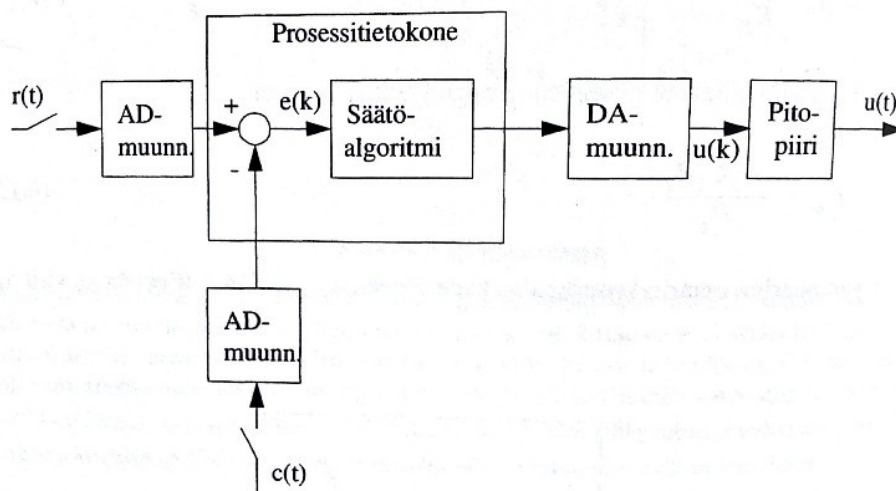
Pitopiiri = Tiedonkeruulaitteessa oleva laite, joka pitää sille syötetyn analogiasignaalin arvon kunnes uusi ohjaussignaali tulee sisään.

Säätöalgoritmi = Windows-sovellukseen ohjelmoitu digitaalinen PID-säädin.

AD-muunnin = Signaalin muunto analogisesta digitaaliseen muotoon.

DA-muunnin = Signaalin muunto digitaalisesta analogiseen muotoon.

$u(t)$ = Säätimen ulostulo. 0-10 V jännite, jolla annetaan taajuusmuuttajalle taajuusohjearvo.



KUVA 16. Signaalin kulku digitaalisessa säätimessä (Savolainen – Vaitinen 2001, 70)

Digitaalisessa säädössä otetaan näytteet asetusarvosta ja mittausarvosta. Näistä arvoista lasketaan erosuure, joka viedään säätimelle. Tietokoneeseen ohjelmoitu PID-säättöalgoritmi laskee ohjauksen arvon. Säädin on tyyppillinen automatiotekniikan säädin. PID-säättöalgoritmi koostuu suhde-, integrointi- ja derivointiosasta sekä näiden osien vakiokertoimista. Suhdeosan ja sen kertoimen avulla voidaan suoraan vaikuttaa säätimen lähtösignaaliin. Käytännössä erosuure vain kerrotaan vakiokertoimella. Integrointiosalla ja sen kertoimella voidaan vaikuttaa siihen kuinka säädin käsittelee pidempään jatkunutta asetus- ja mittausarvon eroa. Pelkkä suhdeosa kertoimineen kykenee huonosti reagoimaan pitkään jatkuneeseen asetus- ja mittausarvoneroon, koska pienillä erosuureilla ei enää ole vaikutusta itse ohjattavaan laitteeseen. Yksinkertaistettuna integrointiosassa summataan aiemmin laskettuja erosuureita yhteen. Derivointiosalla voidaan vaikuttaa nopeisiin erosuureen muutoksiin. Yksinkertaistettuna derivointiosassa verrataan edellisen laskukerran erosuuretta nykyiseen. Suhde-, integrointi- ja derivointiosien painotuksiin voidaan vaikuttaa vakiokertoimien avulla. (Savolainen – Vaitinen 2001, 70.)

Simulaattorin Windows-sovellus sisältää kaksi yksinkertaistettua PID-säädintä. Toisella PID-säätimellä lasketaan taajuusohjearvoja ja pyörimis-

suuntia taajuusmuuttaja TAMU1:lle ja toisella TAMU2:lle. Tiedonkeruulaitteiden valmistaja Measurement Computing ei tarjoa valmista PID-säätimen koodia käytettäväksi VB.NET ympäristössä. Tämän vuoksi täytyi etsiä ratkaisuja muualta.

Ratkaisu PID-säätimen ohjelmointiin löytyi Arduino-ohjelmointikirjastoista. Arduino on ohjelmoitava mikrokontrolleri. Arduinolle on harrastelijavoimin toteutettu useita valmiita ohjelmointiratkaisuja. Myös PID-säädin on osa Arduinon ohjelmakirjastoja. Arduinon PID-kirjastojen suunnittelija Brett Beauregard on laatinut yksityiskohtaisen esittelyn kirjastojen sisällöstä. (Beauregard 2012.) Vaikka ohjelmakoodi on tarkoitettu Arduinon omaan, Wiring-kieleen pohjautuvaan ohjelmointiympäristöön, on se sovellettavissa myös VB.NET-ohjelmointikieleen. Muutoksia Arduinon PID-säätimen vakiokoodiin täytyy tehdä muuttujien määrittelyssä, järjestelmän ajan noutamisessa sekä I/O:n kirjoittamisessa ja lukemisessa. Säätimen perustoiminta on seuraavanlainen:

- Kun videon toisto aloitetaan, käynnistetään samalla PID-säätimet.
- Pyydetään VLC-activex-komponentilta tieto montako millisekuntia videota on toistettu.
- Luetaan kallisteluarvotaulukosta videon aikaleimaa vastaava arvo.
- Asetetaan luettu kallisteluarvo PID-säätimen asetusarvoksi.
- Luetaan kallistelulaitteiston kulma-anturin arvo.
- Lasketaan mitatusta kulma-arvosta ja asetusarvosta erosuure.
- Lasketaan kaunko aikaa on kulunut viime säädöstä.
- Lasketaan säätimen integrintiosa.
- Lasketaan säätimen derivointiosa.
- Kerrotaan P, I ja D-osat ja summataan ne yhteen.
- Tarkastetaan ohjauksen suunta ohjausarvon etumerkin perusteella.
- Asetetaan ohjaussuunta USB-1208FS:n välityksellä taajuusmuuttajalle.
- Asetetaan PID-säätimen laskema moottorin taajuusohjeen arvo USB-3101 välityksellä taajuusmuuttajalle.

- Otetaan talteen tämän laskentakierroksen erosuureen ja ajan arvo.

Yllä kuvattu toiminta suoritetaan ohjelmaan liitetyn ajastimen avulla 100 ms:n välein. Kummankin säätimen yhteydessä lasketaan myös moottorin pyörimissuunnat, jotka syötetään tiedonkeruulaitteen avulla taajuusmuuttajille. Säätimien ohjelmakoodi löytyy sivuilta 5–8 liitteestä 5.

Ennen kuin voidaan syöttää kulma-anturin tuottama arvo säätimelle, täytyy arvoa muokata. Kulma-anturin tuottama 0–5 V jännitearvo täytyy skaalata esimerkiksi välille -15–15, koska säädin tulkitsee arvon 0 tarkoittavan tasapainoasemaa kallistelurakenteessa. Tällöin anturin tuottama jännitearvo 2.5 V kertoo säätimelle kallistelurakenteen olevan tasapainoasemassa. Jännitearvon ollessa yli 2.5 V säädin tulkitsee kallistelurakenteen olevan eteenpäin kallistunut. Jännitearvon ollessa alle 2.5 V säädin tulkitsee kallistelurakenteen olevan taaksepäin kallistunut.

5.8 Häiriötilanteiden käsittely sovelluksessa

Kun simulaattoria ohjaava Windows-sovellus käynnistetään, käynnistyy sovellukseen sisälle taustaprosessi, joka tarkastaa kytkentäkaapissa sijaitsevien releiden R1 ja R2 tilan 10 ms:n välein. Releiden tilatieto saadaan logiikalle viemällä USB-1208FS:n 5 VDC jännitenavasta signaalit releiden R1 ja R2 vaihtokoskettimien kautta takaisin USB-1208FS:n digitaalisiin sisääntuloihin. Releiden apukoskettimet ovat johtavassa tilassa, kun kallistelurakenne on rajojen sisäpuolella eikä stop- tai hätäseis-painikkeita ole painettu.

Jos taustaprosessi havaitsee että 5 VDC signaali USB-1208FS:n digitaalisääntuloissa vaihtuu maatasoon (0 V), menee simulaattorisovellus häiriötilaan. Häiriötilassa moottoreita ohjaavien PID-säätimien suoritus keskeytetään ja taajuusmuuttajille annetaan tiedonkeruulaitteiden välityksellä käsky pysäyttää moottorit. Taajuusohje asetetaan 0 V:ksi (0 Hz).

Seuraavaksi käynnistetään käyttäjälle häiriöstä ilmoittavan videon toisto. Häiriöistä ilmoittavia videoita on yhteensä 3 kpl. Toistettavan ohjevideon va-

linta riippuu siitä, kumman releen vaihtokosketin on vaihtanut tilansa maata vasten. Jos kumpikin releistä on mennyt maata vasten, esitetään kolmas video. Ohjevideoissa kerrotaan käyttäjälle, miten tämän tulee toimia, jotta simulaattori saadaan pois häiriötilasta (kuva 17).



KUVA 17. Kuvakaappaukset Windows Movie Makerilla luoduista häiriötilanvideoista

Mahdollisia häiriötilanteita on kolme. Ensimmäisessä häiriötilanteessa häiriö on aiheutunut siitä, että kallistelurakenne on ajautunut rajalle. Rajalle ajo on aiheuttanut mekaanisen rajakytkimen laukeamisen, joka on saanut aikaan sen, että releen R1 kelajännite on katkennut. Tämä puolestaan johtaa siihen, että kontaktorin K1 kelajännite katkeaa, jolloin taajuusmuuttajien käyttöjännitteen syöttö katkeaa. Niin kauan kun kallistelurakenne on rajalla painaen samalla mekaanista rajakytkintä ei taajuusmuuttaja voi saada käyttöjännitettä. Päästäkseen pois häiriötilasta käyttäjän tulee painaa mekaaniset rajakytkimet ohittava palautuva painonappi S1 pohjaan kytkentäkaapin ovesta.

Samaan aikaan kun käyttäjä painaa mekaaniset rajakytkimet ohittavaa painonappia, Windows-sovelluksen häiriötilanteita käsittelevä taustaohjelma on alkanut tiedonkeruulaitteiden kautta syöttämään taajuusmuuttajille matalaa taajuusohjetta ja sellaista pyörimissuuntaa, että kallistelurakenne saadaan pois painamasta mekaanista raja-anturia. Häiriötilanteita käsittelevä taustaohjelma saa oikean ajosuunnan selville kallistelurakenteeseen asennettujen

reed-anturien ja kestopagneettien avulla. Reed-anturin ollessa kestopagneetin vaikutuspiirissä, se vaihtaa sisäisen vaihtokoskettimensa tilaa ja kytkee 5 V jännitteen USB-1208FS:n digitaaliseen sisääntuloon. Reed-anturit on asennettu kallistelurakenteeseen siten, että kun niihin kytketyt digitaaliset sisääntulot menevät maata vasten, on kallistelurakenne lakannut painamasta mekaanista rajakytkintä. Tämä mahdollistaa sen, että häiriötilanteita käsittelevän taustaprosessin suorittaminen voidaan lopettaa, kun reed-anturit kytkettyvät takaisin maata vasten. Ennen häiriötä käsittelevästä taustaohjelmasta poistumista muutetaan taajuusmuuttajien taajuusohje 0 Hz ja moottorin pyöriminen lopetetaan.

Toisen häiriötilanteen ilmetessä on simulaattorin käyttäjä joko painanut hätäseis-painiketta juoksumaton lähellä tai stop-painiketta kytkentäkaapin ovesta. Näiden painikkeiden painaminen aiheuttaa ohjausvirtapiirissä 2:n releen R2 kelajännitteen katkeamisen. Koska kontaktori K1:n kelajännite on riippuvainen releiden R1 ja R2 tilasta, katkeaa taajuusmuuttajien käyttöjännite. Windows-sovelluksen häiriötilanteita käsittelevän taustaohjelman ei tällä kertaa täydy kuin syöttää taajuusmuuttajille 0 Hz taajuusohjetta ja pysäyttää moottorien ajo. Häiriötilanteita käsittelevä taustaohjelma jää odottamaan että rele R2 menee johtavaan tilaan. R2 saadaan johtavaan tilaan takaisin palauttamalla mahdollisesti pohjaan painettu hätäseis-painike ja painamalla vihreää start-painiketta kytkentäkaapin ovesta. Kun rele R2:n kela saa jälleen jännitteen, saavat taajuusmuuttajatkin käyttöjänniteensä kontaktorin K1-koskettimien kautta.

Kolmas häiriötilanne on kahden yllä esitellyn häiriötilanteen yhdistelmä. Kyseinen häiriötilanne voi ilmetä esimerkiksi sellaisessa tilanteessa että simulaattorin käyttäjä painaa stop- tai hätäseis-painiketta sen jälkeen kun kallistelurakenne on ajanut rajalle. Tässä tilanteessa hätäseis-painike täytyy palauttaa ja painaa vihreää start-painiketta ennen kuin voidaan alkaa suorittaa mekaaniset rajakytkimet ohittavan napin painamista ja rajoilta pois ajamista.

5.9 Sovelluksen toimintojen testaaminen

Windows-sovelluksen toiminnot testattiin, kun sovelluksen rakenne oli saatu lähelle lopullista muotoaan. Liitteessä 3 on nähtävissä testauspöytäkirja. Testaaminen aloitettiin kytkemällä tiedonkeruulaitteet USB-kaapelilla tietokoneeseen. Windows-sovellus käynnistettiin ja sen toiminta käytiin vaihe vaiheelta läpi. Testauspöytäkirja on rakenteeltaan samankaltainen kuin projektin alussa luotu simulaattorin toiminnankuvaus. Testauspöytäkirja on jaettu seuraavien vaiheiden tutkimiseen:

- sovellus käynnistetty työpöydältä
- reitin ajo käynnistetty uudelleen häiriön jälkeen (hätäseis-painike)
- reitin ajo käynnistetty uudelleen häiriön jälkeen (rajalle ajo)
- reitin ajo käynnistetty ensimmäistä kertaa
- reitin ajo loppunut
- häiriö ilmaantunut (rajalle ajo)
- häiriö ilmaantunut (hätäseis-painike)
- häiriötilasta poistuttu (rajalle ajo)
- häiriötilasta poistuttu (hätäseis-painike).

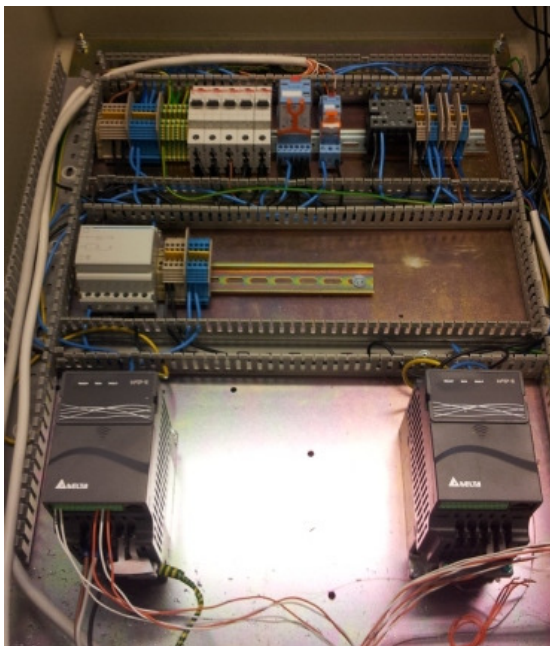
Jokaisessa yllä listatussa vaiheessa tarkastettiin jännitemittarin avulla sen hetkinen I/O:n tila ja kirjattiin nämä arvot ylös testauspöytäkirjaan.

6 ASENNUS JA KÄYTTÖÖNOTTO

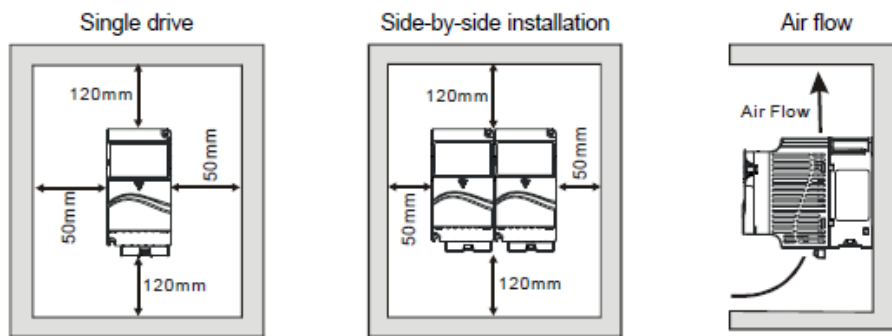
Opinnäytetyöhön sisältyi myös laitteiston asentaminen ja käyttöönotto. Kyt-kentäkaappi pyrittiin rakentamaan siistiksi, mutta ennen kaikkea tavoitteena oli toimiva ja turvallinen järjestelmä. Asennuksen kytkennät tarkastettiin lopuksi yhdessä sähköasennusluvut omaavan opinnäytetyön ohjaajan, Heikki Kurjen kanssa.

6.1 Asennus

Mahdollisimman suuri osa simulaattoriin liittyvistä sähkölaitteista pyrittiin sijoittamaan kytkentäkaappiin. Kyt-kentäkaapissa sijaitsevat ohjausvirtapiirin laitteet, päävirtapiirin laitteet (poislukien moottorit), tiedonkeruulaitteet ja optoerottimet (kuva 18). Kyt-kentäkaappina käytettiin Rittal AE1037.500:a. Kaappi on teräksinen ja se on varustettu yhdellä ovella. Kaapin koko on 400x800x300 mm. Näin suureen malliin päädyttiin taajuusmuuttajien vaatiman jäähdytystilan takia (kuva 19).



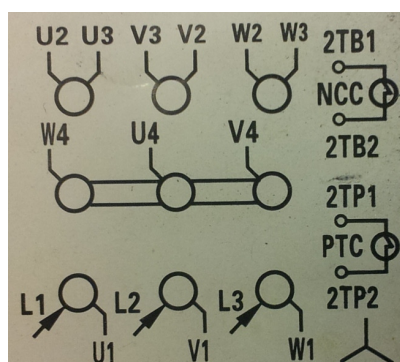
KUVA 18. Yleiskuva kytkentäkaapin sisältä



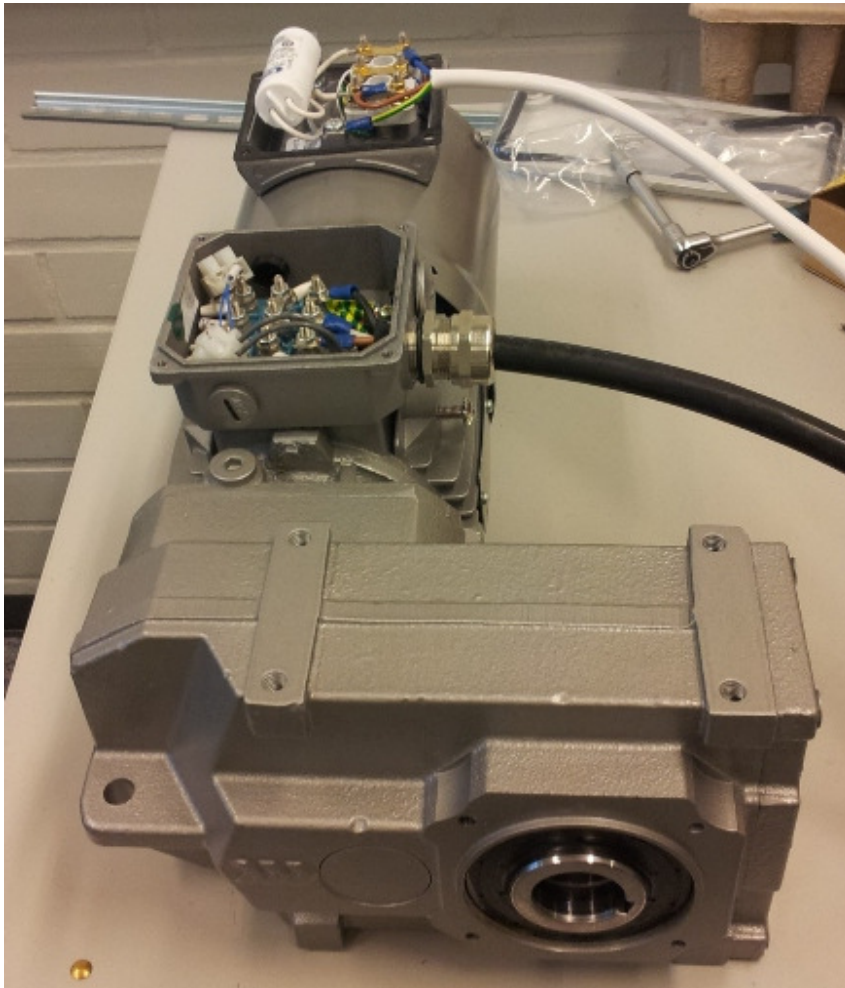
KUVA 19. Taajuusmuuttajan vaatima jäähdystystila (Delta Electronics. 2006, 20.)

6.1.1 Taajuusmuuttajien ja moottorien kaapelointi

Taajuusmuuttajat ovat tyypiltään sellaisia, että niiden käyttöjännite on 1-vaiheinen, mutta moottoriliitäntä 3-vaiheinen. Käyttöjännite tuodaan taajuusmuuttajille 2,5 neliömillimetrin MMJ 2+1 kaapelilla ja moottori kytketään 2,5 neliömillimetrin MCCMK 3+1 kaapelilla. Moottori (kuva 21) kytkettiin tähtikytkentänä kuvan 20 mukaisesti. Taajuusmuuttajan ja moottorin välinen kaapeli on EMC-suojattu ja läpivienneissä käytetään EMC-standardin mukaisia läpivientiholkkeja. EMC-standardin täyttävän kaapeloinnin ja läpivientiholkkien käyttö taajuusmuuttajakäytöissä estää taajuusmuuttajan tuottamien sähkömagneettisten häiriöiden leviämisen ympäristöön (Sähkömagneettinen yhteensopivuus (EMC). 2012.)

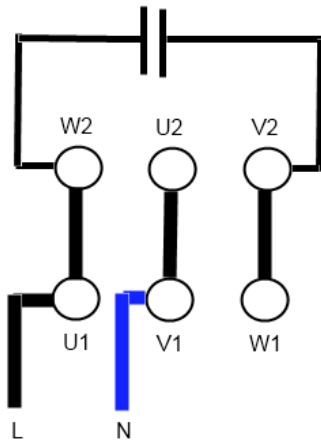


KUVA 20. Moottorin tähtikytkentä



KUVA 21. Kallistelurakennetta sivuttaissuunnassa ohjaava 3-vaiheinen moottori alennusvaihteineen ja lisätuulettimineen

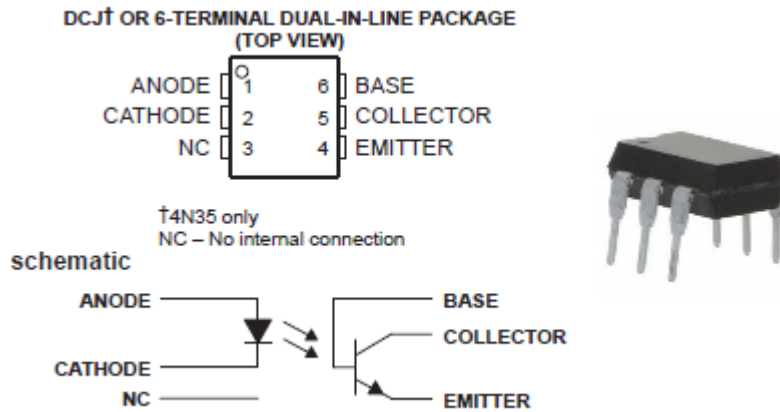
Moottoreihin jälkiasennetut lisätuulettimet kytketään 1,5 neliömillimetrin MMJ 2+1 kaapelilla yksivaiheisena. Tuulettimen moottori on tyypiltään 3-vaiheinen oikosulkumoottori. Valmistaja on asentanut tuulettimen kytkentärasiaan kondensaattorin, jonka avulla moottoria voidaan käyttää yksivaiheisena. Kuvassa 22 on tuulettimen moottorissa käytetty johtimien kytkentä.



KUVA 22. Kallistusrakennetta ohjaavien moottorien lisätuulettimien kytkentä

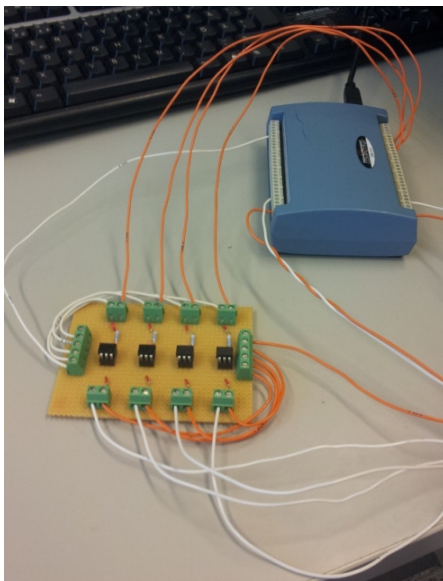
Tuulettimien käyttöjännite syötetään kytkentäkaapin ovesa sijaitsevan kiertokytkimen kautta. Tuulettimet eivät siis automaattisesti kytkeydy päälle, kun simulaattorin käyttöjännitejohdin kytketään pistorasiaan. Kun simulaattorin käyttö aloitetaan, täytyy tuulettimille kytkeä käyttöjännite käsin kiertokytkimestä kääntämällä.

Erään haasteen perinteisen johdotuksen käyttö aiheutti kytkettäessä tiedonkeruulaitteilta tulevia digitaalisia ohjaussignaaleja taajuusmuuttajiin. Ohjausalustaksi valittujen tiedonkeruulaitteiden digitaaliset ulostulot kykenevät antamaan suurimmillaan 5 V jännitteen. Valitun taajuusmuuttajan digitaalisia sisääntuloja voidaan aktivoida vain 24 V jännitteellä. Tämä aiheuttaa sen, että jännitteen nosto täytyy suorittaa tiedonkeruulaitteen digitaalisen ulostulon ja taajuusmuuttajien digitaalisen sisääntulon välillä. Jännitteen nosto suoritettiin käyttämällä optoerottimia. Optoerottimiksi valittiin Fairchildin valmistama 4N36 (kuva 23). Käytännössä optoerotin koostuu koteloinnin sisällä olevasta infrapunaledistä ja fototransistorista. Tiedonkeruulaitteen digitaalinen ulostulosignaali antaa ledille käyttöjännitteen ja ledin loistaessa fototransistorin kollektorin ja emitterin väli muuttuu johtavaksi.



KUVA 23. Fairchild 4N36-optoerotin

Optoerottimia käyttämällä saadaan myös aikaan galvaaninen erotus tiedonkeruulaitteille. Näin tiedonkeruulaitteet eivät ole enää samassa piirissä johtimien kanssa, jotka kulkevat taajuusmuuttajalle. Optoerottimen anodiin kytkettiin 5 V digitaalinen ulostulo tiedonkeruulaitteesta. Sarjaan lisättiin tämän kytkennän kanssa vielä vastus, jolla tiedonkeruuyksikön syöttämää virtaa saatiin rajoitettua. Vastukset ja optoerottimet kantoineen kolvattiin verolevylle (kuva 24). Verolevylle kolvattiin myös riviliittimet, jotka helpottavat johtojen liittämistä.



KUVA 24. Verolevylle kolvatut optoerottimet, vastukset ja riviliittimet. Yllä USB-3101-tiedonkeruulaite

Optoerottimen sisällä olevan infrapunaledin etuvastuksen resistanssiarvo laskettiin kaavalla 1.

$$R = \left(\frac{U_{DAQ} - U_{LED}}{0,01A} \right) \Omega \quad \text{KAAVA 1.}$$

$$R = \left(\frac{5V - 1,5V}{0,01A} \right) \Omega$$

$R = 350\Omega$, jossa

R = vastuksen resistanssi

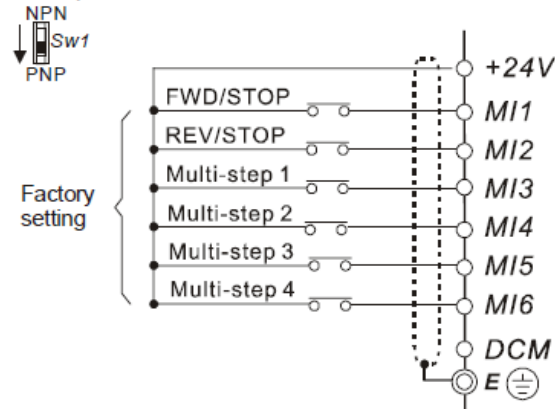
U_{DAQ} = tiedonkeruulaitteen digitaalisen ulostulon jännite

U_{LED} = optoeristimen sisältämän infrapunaledin kynnysjännite

Virran rajoitus on tärkeää koska tiedonkeruulaite kykenee syöttämään enimmillään 100 mA:a virtaa I/O:nsa kautta (Measurement Computing, 2006, 21). Taajuusmuuttajan 24 VDC jännitelähtö kytkettiin optoerottimen sisältämän fototransistorin kollektorin ja emitterin kautta taajuusmuuttajan digitaalisiin sisääntuloihin MI1 ja MI2 (kuva 25).

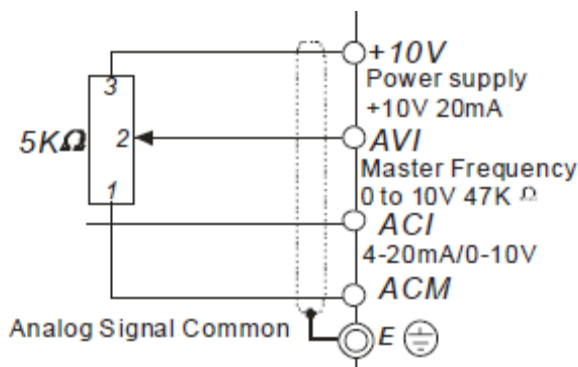
Taajuusmuuttajan MI1 porttiin kytketty 24 V saa aikaan moottorin pyörimisen myötäpäivään ja MI2 porttiin kytketty 24 V ja moottorin pyörimään vastapäivään.

C. PNP mode without external power



KUVA 25. Taajuusmuuttajan digitaalisten signaalien sisääntulo

Taajuusmuuttajien 0–10 V taajuusohje saadaan USB-3101:ltä. Eteen- ja taaksepäin kallistelua hoitavan taajuusmuuttajan taajuusohje saadaan USB-3101:n analogisesta ulostulosta VOUT1 ja sivuttaissuunnassa hoitavan taajuusmuuttajan taajuusohje analogisesta ulostulosta VOUT2. Analogiasignaalin nollassa kytketään USB-3101:n AGND:n ja taajuusmuuttajan ACM:n välille (kuva 26).



KUVA 26. Taajuusmuuttajan analogisen signaalin sisääntulo

6.1.2 Muiden laitteiden kaapelointi

Ohjausvirtapiirien kaapeloinnissa käytetään halkaisijoiltaan 1,5 neliömillimetriä olevia kuparijohtimia ja 10 A johdonsuojakatkaisijoita. Ohjausvirtapiirien

laitteisiin kuuluvat releet R1 ja R2, kontaktori K1, hätäseis-, start/stop- ja rajojen_ohitus –painikkeet sekä mekaaniset rajakytkimet.

Reed-antureiden kaapelointiin käytettiin 0,5 neliömillimetrin NESMAK 8x2 -kaapelia.

6.2 Käyttöönotto

Laitteiden kaapeloinnin jälkeen suoritettiin järjestelmän käyttöönotto, joka aloitettiin kytkemällä käyttöjännite 1-vaiheisesta pistorasiasta kytkentäkaappiin. Johdonsuojakatkaisijat kytkettiin päälle, lisätuulettimille käyttöjännitteen kytkävä kiertokytkin asetettiin johtavaan tilaan ja kytkentäkaapin etuovesta painettiin vihreää start-painiketta. Ohjausvirtapiirit kytkivät päävirtapiirin taajuusmuuttajille käyttöjännitteen.

Ennen johdonsuojakatkaisijoiden päälle kytkentää asennettiin taajuusmuuttajiin ohjauspaneelit (kuva 27). Ohjauspaneeli sisältää painikkeita ja pienen näytön. Sen avulla voidaan muuttaa taajuusmuuttajan parametreja ja ohjata moottoreita ilman ulkoisia ohjaussignaaleja.



KUVA 27. Delta Electronic VFD-E -taajuusmuuttajan ohjauspaneeli

Taajuusmuuttajiin asetetut lopulliset parametrit olivat seuraavat:

- Pr.02.00 = Taajuusohjeen lähteeksi asetettiin analoginen jännitesääntulo.

- Pr.02.01 = Moottorien pyörimissuuntaa käskyttäväksi lähteeksi asetettiin ulkoiset johdinterminaanlit.
- Pr.02.04 = Moottorin sallituiksi pyörimissuunniksi asetettiin eteen ja taakse.
- Pr.01.09 = Moottorin kiihdytysaika asetettiin 1 sekuntiin.
- Pr.01.10 = Moottorin hidastusaika asetettiin 1 sekuntiin.

Ennen lopullisten parametrien asetusta moottoria koekäytettiin ajamalla moottoria eteen- ja taaksepäin. Koekäyttöä varten taajuusohjeen lähde ja moottorien pyörimissuunnan lähde muutettiin hetkellisesti käytettäväksi ohjauspaneelist.

6.3 Toiminnan testaaminen

Asennusten ja käyttöönoton jälkeisessä toiminnan testaamisessa ei enää juurikaan tarvinnut keskittyä Windows-sovellukseen. Videotoiston tiedettiin toimivan ja sovelluksen muu toiminta oli testattu aiemmin yksityiskohtaisesti (liite 3). Tässä luvussa käsitellään moottorien ohjauksen ja häiriötilanteiden käsittelyn testaaminen.

6.3.1 Moottorien ohjaus

Moottoreiden koekäytön jälkeen oli mahdollista testata taajuusmuuttajille ohjeita antavaa Windows-sovellusta. KytKentäkaappiin syöttöjännitteen antava pistoke kytkettiin yksivaiheiseen pistorasiaan ja kytkentäkaapista painettiin start-painiketta. Tämä sai aikaan taajuusmuuttajien käynnistymisen. Tietokoneelta käynnistettiin simulaattoria ohjaava sovellus.

Kallistelurakenteen puutteesta johtuen ei simulaattorin toimintaa pystynyt testaamaan käytännössä. Tämä teki myös simulaattorin Windows-sovelluksen digitaalisten säätimien virityksen hankalaksi.

Moottorien ohjauksen toiminnan testaaminen aloitettiin kokeilemalla moottorien ajoa myötä- ja vastapäivään simulaattorin Windows-sovelluksella. Käytännössä tämä tapahtui asettamalla sovellus lukemaan taulukon 3 mukaiset kallistustiedot todellisen maastojuoksureitin kallisteluarvojen sijasta.

TAULUKKO 3. Moottorien testausta varten laadittu kallisteluarvotaulukko

Näytteenottohetki	Kallistus eteen/taakse	Kallistus sivusuunnassa
1000 ms	5	5
2000 ms	5	5
3000 ms	5	5
4000 ms	5	5
5000 ms	5	5
6000 ms	-10	-10
7000 ms	-10	-10
8000 ms	-10	-10
9000 ms	-10	-10
10000 ms	-10	-10
11000 ms	15	15
12000 ms	15	15
13000 ms	15	15
14000 ms	15	15
15000 ms	15	15

Windows-sovelluksen ohjelmointikoodiin on mahdollista asettaa ulkoisessa kallisteluarvotaulukossa käytetty näytteenottoväli. Tässä tapauksessa näytteenottoväliksi asetettiin 1000 ms. Koska säätimen lukema kallisteluarvo (asetusarvo) pysyy aina 5 sekuntia samana, antaa se ohjaukselle aikaa kasvattaa ulostuloarvoa. Digitaalisille PID-säätimille asetettiin seuraavat viritysparametrit.

$$K_p = 1.0$$

$$K_i = 0.3$$

$$K_d = 0.3$$

Testauksen tässä vaiheessa ei vielä käytetty kallistusanturia. Sovelluksen lukema kallisteluanturitieto pysyi siis jatkuvasti jännitearvossa 0 V. Käytännössä tämä tarkoitti sitä, että digitaalisen säätimen erosuure pysyi jatkuvasti

nollasta poikkeavana. Tämä puolestaan aiheutti sen, että digitaalisen säätimen ulostulo antoi taajuusmuuttajalle jatkuvasti 0 Hz suurempaa taajuusohjetta. Toisin sanoen moottori pysyi jatkuvasti liikkeessä. Tarkasteltaessa taulukon 3 kallisteluarvoja huomataan niiden muuttuvan 5 sekunnin välein. Kun taulukon kallisteluarvo on nollaa suurempi, pyrkii säädin ohjaamaan moottoria myötäpäivään. Kun taulukon kallisteluarvo on nollaa pienempi, pyrkii säädin ohjaamaan moottoria vastapäivään.

Kun taulukon 3 mukaiset kallisteluarvot oli ladattu sovellukseen painamalla ”Lataa reitti 1” -painiketta, käynnistettiin simulointi. Moottorien ohjaaminen tietokoneelta toimi kuten oli suunniteltu. Moottorin vaihtaessa suuntaa kiihtyminen muuttui kerta kerralta nopeammaksi, kun digitaalinen säädin sai tiedon muuttuneesta asetusarvosta. Säätimen luonteeseen kuuluu se, että mitä kauempana mittausarvo on asetusarvosta, sitä suuremman ohjausarvon se tuottaa.

Moottoriohjauksen testaaminen sujui onnistuneesti. Lopullinen digitaalisten säätimien viritys voidaan suorittaa kallistelukentän valmistuttua.

6.3.2 Häiriötilanteiden käsittely

Tämän työn puitteissa ehdittiin testaamaan Windows-sovelluksen häirönkäsittelyä ainoastaan hätäseis- ja stop-painikkeen painamisen osalta. Häiriökäsittelyn toimintaa testattiin käynnistämällä simulointi ja painamalla hätäseis- tai stop-painiketta. Kummassakin tapauksessa painikkeen painaminen aiheutti taajuusmuuttajien syöttäjännitteen katkeamisen ja moottorien pysähtymisen. Samaan aikaan maastajuoksuvideon toisto monitorilla keskeytyi ja ohjevideon toistaminen alkoi. Ohjevideossa kehoitetaan käyttäjää palauttamaan mahdollisesti pohjaan painettu hätäseis-painike, kun hätätilanne on ohi ja painamaan tämän jälkeen vihreää painiketta kytkentäkaapin ovesta. Vihreän painikkeen painaminen saa aikaan ohjausvirtapiiri 2:een liitetyn rele R2:n vetämisen. Tämä saa aikaan taajuusmuuttajien syöttäjännitteen palautumisen. Releen R2 vetäminen saa myös aikaan Windows-sovelluksen ohjevideon toiston pysäyttämisen ja näkymän palaamisen takai-

sin ikkunoituun käyttöliittymään. Käyttäjä voi nyt halutessaan aloittaa simuloinnin alusta painamalla "Käynnistä"-painiketta.

7 KEHITYSIDEAT

7.1 Käyttöpaneeli

Simulaattorin kehitystyön tämänhetkisessä vaiheessa käyttäjä joutuu ope-
roimaan simulaattoria tietokoneen hiirellä. Simulointivaiheessa juoksumatto
pyörii ja sitä kallistellaan reitin mukaisesti. Luonnollisesti tällaisessa tilan-
teessa tietokoneen hiiren käyttö voi olla haastavaa. Esimerkiksi videon tois-
toa nopeuttaakseen käyttäjän tulee poistua tietokoneen hiiren avulla maasto-
juoksuvideon täyden ruudun toistosta takaisin ikkunoituun
käyttöliittymänäkymään ja painaa hiirellä ”Nopeuta”-painiketta. Yhtä hanka-
laa on myös keskeyttää tai pysäyttää videon toisto.

Jotta simulaattorin käyttö saataisiin käyttäjän osalta sujuvaksi, täytyy hiiren
käytöstä luopua simulaattorin ohjauksessa. Nopeasti toteutettava ja edulli-
nen vaihtoehto olisi rakentaa käyttöpaneeli, joka sisältää painikkeet simu-
laattorin ohjaukseen. Paneeli voitaisiin kiinnittää juoksumaton alkuperäisen
hallintaelektroniikan tilalle tai rinnalle.

Paneeli voisi koostua esimerkiksi vahvasta muovisesta tai alumiinisesta
taustalevystä jota on helppo työstää. Taustalevyyn kiinnitettäisiin esimerkiksi
kuvan 28 kaltaiset palautuvat painikkeet reitin lataamiseen ja simuloinnin
käynnistämiseen, pysäyttämiseen ja keskeyttämiseen. Maastajuoksuvideon
nopeuden muutos voitaisiin toteuttaa käyttämällä potentiometriä.



KUVA 28. Palautuva painike sulkeutuvalla ja avautuvalla koskettimella

Tämänhetkisessä simulaattorin kehitysvaiheessa esimerkiksi USB-1208FS-tiedonkeruulaitteessa on ohjelmoitavia digitaalisia I/O-liitäntöjä vielä 10 kappaletta ja analogisia sisääntuloja 7 kappaletta. Nämä riittäisivät mainiosti yllä kuvatun kaltaisen käyttöpaneelin toteuttamista varten.

7.2 Käyttöliittymän toteutus videoilla ja käyttöpaneelilla

Jos luvussa 8.1 kuvattu käyttöpaneeli toteutettaisiin, tällä hetkellä käytössä oleva ikkunoitu Windows-sovelluksen käyttöliittymä kävisi tarpeettomaksi. Simulaattorin käyttöliittymä voitaisiin toteuttaa siten, että kun Windows-sovellus käynnistetään videokomponentti alkaa välittömästi toistaa videota, jossa simulaattorin käyttäjää ohjeistetaan käyttöpaneelin käytöstä ja simulaattorin toiminnoista. Nykyisessä käyttöliittymässä käyttäjälle esitettävät infolaatikot voitaisiin korvata videoilla, joissa käyttäjää opastetaan kunkin tilanteen mukaan. Osittain tämä onkin jo käytössä häiriötilanteiden ilmetessä simulaattorissa.

7.3 Juoksumaton nopeuden säätö

Juoksumaton mattoa pyörittää tavanomainen tasavirtamoottori. Juoksumaton valmistaja on sisällyttänyt laitteeseen elektroniikkaa, jolla käyttäjä voi ohjata juoksumaton pyörintänopeutta. Jotta juoksumaton pyörintänopeutta voitaisiin ohjata tiedonkeruulaitteiden avulla Windows-sovelluksesta, täytyy tiedonkeruulaitteet joko johdottaa osaksi juoksumaton alkuperäistä ohjaus-elektroniikkaa tai kytkeä ohjauselektroniikka pois käytöstä ja tehdä tasavirtamoottorille täysin uusi ohjaus. Koska valmistaja ei tarjoa minkäänlaista tietoa juoksumattoon sisällytetystä elektroniikasta, on helpompaa kytkeä se pois käytöstä ja suunnitella uusi ohjaus moottorille.

Tasavirtamoottorin kierrosnopeutta voidaan säätää sille syötettävää jännitettä muuttamalla. Mitä pienempi on jännite, sitä pienempi pyörintänopeus. Jännitteen säätö voidaan toteuttaa esimerkiksi tyristorilla, jota ohjataan tiedonkeruulaitteella.

7.4 PC:n sisällyttäminen osaksi kytkentäkaappia

Tällä hetkellä simulaattorin Windows-sovellusta suoritetaan todella suorituskykyisellä työasematietokoneella. Simulaattoria ohjaava Windows-sovellus ei kuitenkaan vaadi kovinkaan suorituskykyistä tietokonetta. Suurin rasitus Windows-sovelluksessa aiheutuu tietokoneen näytönohjaimelle, kun sillä toistetaan teräväpiirtovideota.

Nykyisellä tietokoneella on suuri ATX-kotelo, joka vie paljon tilaa. Ajatellen simulaattorin kuljetusta ja käyttöönottoa, olisi kätevintä, jos tietokone sijaitsi simulaattorin kytkentäkaapissa. Valitsemamme kytkentäkaappi on suurikokoinen ja sisältää riittävästi tilaa pienikokoiselle tietokoneelle. Markkinoilla on laaja valikoima mini-itx-emolevyjä, jotka ovat todella pienikokoisia (17 cm x 17 cm). Tällaisia emolevyjä käytetään yleisesti HTPC-tietokonekokoonpanojen pohjana. Esimerkkinä tällaisesta on Asus AT5IONT-I nVidia ION Mini ITX -emolevy (kuva 29), joka sisältää Intel Atom D525 kaksiydinprosessorin esiasennettuna.



KUVA 29. Asus AT5IONT-I nVidia ION Mini ITX –emolevy

Kyseinen emolevy sisältää HDMI-liitännän simulaattorin monitorin kytkentää varten ja siinä on nVidia ION -grafiikkapiiri, joka kykenee teräväpiirtovideon purkamiseen.

8 YHTEENVETO

Tämän insinööriyön tarkoituksena oli suunnitella ja toteuttaa maastajuoksusimulaattoriin ohjausjärjestelmä. Ohjausjärjestelmän oli tarkoitus kattaa maastajuoksuvideon toistaminen, simulaattorin käyttöliittymä ja kallistusrakennetta ohjaavien moottoreiden ohjaus. Projektin alkuvaiheessa ei mekaanisen kallistelurakenteen toteuttamisesta ollut olemassa tarkkaa aikataulua, joten insinööriyön päätavoitteeksi muodostui saada aikaan yllä mainitut ominaisuudet sisältävä järjestelmä, joka toimii testiympäristössä ilman kallistelurakennetta.

Maastajuoksusimulaattorin ohjausjärjestelmän Windows-sovellus sisältää kaksi digitaalista säädintä. Digitaaliset säätimet käyttävät hyväkseen mekaaniseen kallistusrakenteeseen liitettyä kallistusanturia. Koska kallistusrakenne ei valmistunut ajallaan, oli digitaalisten säätimien virittäminen hankalaa. Muilta osin projektin voidaan kuitenkin katsoa olleen onnistunut, koska asetetut tavoitteet saavutettiin.

Haastavin asia suunnittelussa oli ehdottomasti simulaattorin ohjausalustan valinta. Haasteellisen tästä allekirjoittaneelle teki erityisesti kokemuksen vähäisyys muista ohjausalustoista kuin ohjelmoitavista logiikoista. Ohjausalustan valintaan vaikutti ratkaisevasti se, että moottorien ohjauksen ja videotoston tuli kyetä vuorovaikutukseen.

Opinnäytetyön tekemisen aikana syntyi paljon dokumentaatiota, jota voidaan käyttää hyödyksi simulaattorin jatkokehityksessä. Kommentoitu Windows-sovelluksen koodi on helppo omaksua, mikä tekee siitä helpon jälleenkehittää.

Olen erittäin tyytyväinen päätöksestäni liittyä tähän projektiin. Erityisen palkitsevaa oli, kun sai suunnitelmiansa pohjalta tilata tarvittavat laitteet ja asennustarvikkeet ja lopulta itse rakentaa laitteiston. Projektiin käytetty työmäärä maksoi itsensä takaisin ansaittuna kokemuksena. Erityisesti kokemusta kertyi sähkösuunnittelusta ja anturityyppien tutkimisesta.

LÄHTEET

ABB ProfibusDP-sovitin ACS550/800. 2012. SLO. Saatavissa: <http://www.slo.fi/www/fi/Tuotteet/Tuoteluettelo/Sivut/tuotetietosivu.aspx?partno=3804096>. Hakupäivä 25.5.2012

Arduino Uno. 2012. Saatavissa: <http://arduino.cc/en/Main/ArduinoBoardUno>. Hakupäivä 30.5.2012

Ball, P.E. – Stuart R. 2002. Embedded Microprocessor Systems: Real World Design. Amsterdam: Newnes.

Beauregard, B. 2012. Improving the Beginner`s PID. Saatavissa: <http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/>. Hakupäivä 27.5.2012

Franklin, K. 2002. VB.NET. Docendo Finland Oy.

Laava, T. 2011. Maastajuoksusimulaattorin mekaniikkasuunnittelu. Oulun seudun ammattikorkeakoulu. Kone- ja tuotantotekniikan koulutusohjelma. Opinnäytetyö.

LOGO!-tuoteluettelo. 2012. Siemens. Saatavissa: http://www.siemens.fi/pool/products/industry/iadt_is/palvelut_ja_koulutus/tek_ninen_tuki/tuoteluettelot/st70.pdf. Hakupäivä 30.5.2012

Savolainen, Jari – Vaittinen, Risto. 2001. Sääätötekniikan perusteita. Opetushallitus.

Simulointi. 2012. Laurea ammatikorkeakoulu. Saatavissa: <http://ppp.laurea.fi/Sivut/simulaatio.htm>. Hakupäivä 25.5.2012.

Sähkömagneettinen yhteensopivuus (EMC). 2012. Tukes. Saatavissa: <http://www.tukes.fi/fi/Toimialat/Sahko-ja-hissit/EMC/>. Hakupäivä 25.5.2012

Tiedonkeruutuotteet. 2012. National Instruments. Saatavissa:
<http://finland.ni.com/tiedonkeruu>. Hakupäivä 25.5.2012

USB-1208FS User Manual. 2006. Measurement Computing. Käyttöohje.

USB-3101 User Manual. 2006. Measurement Computing. Käyttöohje.

USB DAQ Modules. 2012. Measurement Computing. Saatavissa:
<http://www.mccdaq.com/solutions/USB-Data-Acquisition.aspx>. Hakupäivä
30.5.2012

VFD-E User Manual. 2009. Delta Electronics. Käyttöohje.

Visual Basic. 2012. Wikipedia. Saatavissa:
http://fi.wikipedia.org/wiki/Visual_Basic. Hakupäivä 30.5.2012

LIITTEET

- Liite 1. Toimintakuvaus maastajuoksusimulaattorista
- Liite 2. Ohjaus- ja päävirtapiirin sähkösuunnitelma
- Liite 3. Simulaattorin toiminnantestauspöytäkirja
- Liite 4. Kuparijohtimien koot ja niiden kuormitettavuus
- Liite 5. Windows-sovelluksen ohjelmointikoodi

Toiminnankuvaus maastajuoksusimulaattorista

Normaali käynnistys:

Softa

- Tarkastetaan releiden apukoskettimien avulla, tuleeko taajuusmuuttajille käyttöjännite
- Aloitusruudusta painetaan 'Lataa reitti'-painiketta, jolloin ohjelma luo ulkoisesta kallisteludatan sisältävästä taulukkotiedostosta oman, ohjelman sisäisen taulukon
- Aloitusruudusta painetaan käynnistä painiketta, jolloin käyttäjää kehoitetaan nousemaan matolle ja laittamaan 3d-lasit päähän. Kun käyttäjä painaa pop-up-ikkunasta 'Ok'-painiketta niin videon pyörittäminen kokoruututilassa alkaa ja taajuusmuuttajia ohjaavat PID-säätimet alkavat toimia
- Ohjausvirtapiirin releiden tilaa(häiriötilanteita) tutkiva taustatarkkailija-aliohjelma käynnistyy

Rauta

- Mekaaniset rajakytkimet normaalitilassa eli suljettu (kallistelurakenne ei ole rajalla)
- Hätäseis-painike ylhäällä
- Päälle/seis-nappipaketista painettu vihreää nappia
- Ohjausvirtapiirien releiden kelat saavat käyttöjännitteen

Käynnistys häiriön jälkeen (rajalle ajo / hätäpainikkeen paino):

Softa

- Jos saadaan aikaiseksi häiriö, ohjelma tunnistaa onko kyseessä rajan ylitys vai hätkäkatkaisimen/pois päältä –painikkeiden painaminen (tunnistus releiden vaihtokoskettimista)
- Kun häiriö on havaittu niin käyttäjälle näytetään opastusvideo
- Häiriöille on omat aliohjelmansa joiden suorittamisesta päästään takaisin normaalitilaan vain suorittamalla seuraavat asiat
 - o Jos raja ylitettiin, vaaditaan käyttäjää painamaan palautuva painokyt-kin pohjaan ja pitämään se pohjassa. Tällä napilla saadaan väliaikaisesti ohitettua ensimmäisessä ohjauspiirissä syntynyt katko lauenneen mekaanisen rajakytkimen kohdalla. Kun rele kytkeytyy takaisin johtavaan tilaan, ohjelma alkaa suorittaa rajalta poisajoa. Kallistelurakenteeseen on asennettu myös kosketuksettomat raja-anturit, joiden avulla tiedetään millä rajalla mekaaninen raja-anturi on lauennut. Ajetaan siis niin kauan että kosketuksettomalta raja-anturilta saadaan tieto ettei olla enää rajalla. Tämän jälkeen tämä häiriöitä käsittelevä aliohjelma keskeytetään.

- Jos on painettu hätä-seis painiketta tai punaista nappia päälle/pois nappiboksista, käyttäjää pyydetään hoitamaan mahdollinen hätätilanne ja tämän jälkeen asettamaan hätäseis-painike normaalitilaan ja painamaan vihreää nappia päälle/pois-nappiboksista. Ohjelma tunnistaa, että rele on jälleen johtavassa tilassa ja häiriötä käsittelevä aliohjelma lopetetaan.

Rauta

- Häiriötilanteen tapahduttua rele r3 ei johda käyttöjännitettä taajuusmuuttajille
- Releiden r1 ja r2 kelojen jännitteen palaututtua aktivoituu rele r3 joka kytkee käyttöjännitteen taajuusmuuttajille

Ajo:

Softa

- Ohjelma lukee ulkoisesta taulukkotiedostosta arvoja, jotka toimivat asetusarvoina säätimelle, joka muuttaa taajuusmuuttajan taajuusohjetta ja moottorien ajosuuntaa
- Ajoitus hoidetaan siten että kun tulee aika kallistaa uuteen asentoon, luetaan videon aikaleima ja aikaleiman perusteella luetaan oikea arvo taulukon sarakkeesta. Tämä onnistuu koska taulukon arvot on otettu tietyllä näytteenottotaajuudella.
- Samaan aikaan taustatarkkailija aliohjelma tarkkailee releiden tilaa vaihtokoskettimien ja IO-boksin digitaalisen sisääntulon avulla
- Jos releiden vaihtokoskettimet muuttavat tilaansa ajon aikana eli tulee häiriö niin suoritetaan kohdan *Ajon pysäyttäminen kun tulee häiriö mukaiset toimet*

Rauta

- Mekaaniset rajakytkimet normaalitilassa eli suljettu (kallistelurakenne ei ole rajalla)
- Hätäseis-painike ylhäällä
- Päälle/seis-nappipaketista painettu vihreää nappia
- Ohjausvirtapiirien releiden kelat saavat käyttöjännitteen

Ajon pysäyttäminen kun reitti loppuu:

Softa

- Kun ajon aikainen ohjelma saa aikaleiman, joka kertoo että video on loppumaisillaan, moottoreilla ohjataan kallistelurakenne taikaisin tasapainoasemaan
- Videon pyöritys päättyy, ja ohjelma palaa takaisin aloitusnäkyeseen

Rauta

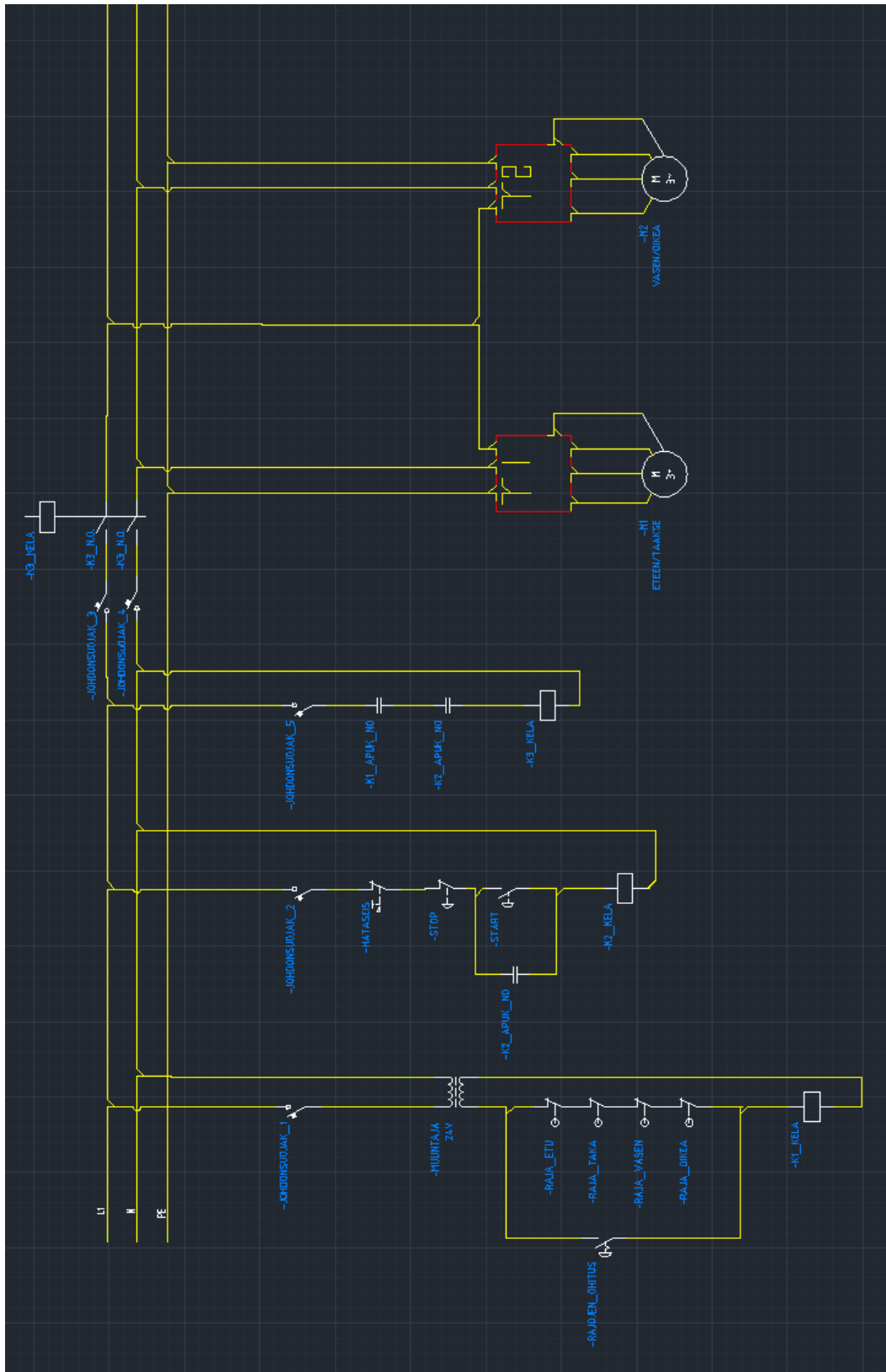
Ajon pysäyttäminen kun tulee häiriö (rajalle ajo / hätäpainike)

Softa

- Ohjelma tunnistaa releiden vaihtokoskettimien avulla kumpi häiriöistä on kyseessä
- Reittivideon pyöritys lopetetaan, säätimet pysäytetään, käynnistetään tilanteen mukainen video jossa neuvotaan käyttäjää häiriötilanteesta pääsemiseksi
- Taajuusmuuttajille aletaan syöttämään taajuusohjeeksi 0 Hz ja käskyä taajuusmuuttajille kumpaankaan suuntaan pyörimiselle ei anneta

Rauta

- Jompikumpi releiden kelajännitteiden lenkeistä katkeaa ja aiheutuu kummankin taajuusmuuttajan käyttöjännitteen katkaisu
- Rajakatkaisimet on asennettu niin, että kun jännite on rajalle tulon takia katkennut, ohjauspiirin jännite ei voi palata muuten kuin pitämällä rajakatkaisimet ohittavaa painonappia pohjassa.



Maastajuoksusimulaattorin toiminnan testaus

Ohjelman klikattu käyntiin työpöydältä eli alkutila

Digitaaliset sisään

	<u>Tila</u>	<u>I/O purkki / liittimet</u>
1. Releen apukosketin	1	USB-1208FS / Port A0 / Pin 21
2. Releen apukosketin	1	USB-1208FS / Port A1 / Pin 22
Kallistuksen eturaja	0	USB-1208FS / Port A2 / Pin 23
Kallistuksen takaraja	0	USB-1208FS / Port A3 / Pin 24
Kallistuksen vasen raja	0	USB-1208FS / Port A4 / Pin 25
Kallistuksen oikea raja	0	USB-1208FS / Port A5 / Pin 26

Digitaaliset ulos

	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Taajuusmuuttaja 1:n moottorin ajo eteenpäin	0	USB-3101 / DIO0
Taajuusmuuttaja 1:n moottorin ajo taaksepäin	0	USB-3101 / DIO1
Taajuusmuuttaja 2:n moottorin ajo eteenpäin	0	USB-3101 / DIO2
Taajuusmuuttaja 2:n moottorin ajo taaksepäin	0	USB-3101 / DIO3

Analogiset sisään

	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Kallistusanturi eteen/taakse	x	USB-1208FS / CH0 / Pin 1
Kallistusanturi sivuille	x	USB-1208FS / CH1 / Pin 4

Analogiset ulos

	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Taajuusmuuttaja 1:n taajuusohje	0	USB-3101 / VOUT0 / Pin 1
Taajuusmuuttaja 2:n taajuusohje	0	USB-3101 / VOUT1 / Pin 29

Reitin ajo käynnistetty uudelleen häiriön jälkeen (häätäpainikkeen paino):

Digitaaliset sisään

	<u>Tila</u>	<u>I/O purkki / liittimet</u>
1. Releen apukosketin	1	USB-1208FS / Port A0 / Pin 21
2. Releen apukosketin	1	USB-1208FS / Port A1 / Pin 22
Kallistuksen eturaja	0	USB-1208FS / Port A2 / Pin 23
Kallistuksen takaraja	0	USB-1208FS / Port A3 / Pin 24
Kallistuksen vasen raja	0	USB-1208FS / Port A4 / Pin 25
Kallistuksen oikea raja	0	USB-1208FS / Port A5 / Pin 26

Digitaaliset ulos

	<u>Tila</u>	<u>I/O purkki / liittimet</u>
	1/0	
Taajuusmuuttaja 1:n moottorin ajo eteenpäin	OK	USB-3101 / DIO0
	1/0	
Taajuusmuuttaja 1:n moottorin ajo taaksepäin	OK	USB-3101 / DIO1
	1/0	
Taajuusmuuttaja 2:n moottorin ajo eteenpäin	OK	USB-3101 / DIO2
	1/0	
Taajuusmuuttaja 2:n moottorin ajo taaksepäin	OK	USB-3101 / DIO3

<u>Analogiset sisään</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Kallistusanturi eteen/taakse	x	USB-1208FS / CH0 / Pin 1
Kallistusanturi sivuille	x	USB-1208FS / CH1 / Pin 4

<u>Analogiset ulos</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Taajuusmuuttaja 1:n taajuusohje	0..10v OK	USB-3101 / VOUT0 / Pin 1
Taajuusmuuttaja 2:n taajuusohje	0..10v OK	USB-3101 / VOUT1 / Pin 29

Reitin ajo käynnistetty ensimmäistä kertaa:

<u>Digitaaliset sisään</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
1. Releen apukosketin	1	USB-1208FS / Port A0 / Pin 21
2. Releen apukosketin	1	USB-1208FS / Port A1 / Pin 22
Kallistuksen eturaja	0	USB-1208FS / Port A2 / Pin 23
Kallistuksen takaraja	0	USB-1208FS / Port A3 / Pin 24
Kallistuksen vasen raja	0	USB-1208FS / Port A4 / Pin 25
Kallistuksen oikea raja	0	USB-1208FS / Port A5 / Pin 26

<u>Digitaaliset ulos</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Taajuusmuuttaja 1:n moottorin ajo eteenpäin	vaihtelee 1/0 OK	USB-3101 / DIO0
Taajuusmuuttaja 1:n moottorin ajo taakse-päin	vaihtelee 1/0 OK	USB-3101 / DIO1
Taajuusmuuttaja 2:n moottorin ajo eteenpäin	vaihtelee 1/0 OK	USB-3101 / DIO2
Taajuusmuuttaja 2:n moottorin ajo taakse-päin	vaihtelee 1/0 OK	USB-3101 / DIO3

<u>Analogiset sisään</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Kallistusanturi eteen/taakse	x	USB-1208FS / CH0 / Pin 1
Kallistusanturi sivuille	x	USB-1208FS / CH1 / Pin 4

<u>Analogiset ulos</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Taajuusmuuttaja 1:n taajuusohje	vaihtelee 0..10v OK	USB-3101 / VOUT0 / Pin 1
Taajuusmuuttaja 2:n taajuusohje	vaihtelee 0..10v OK	USB-3101 / VOUT1 / Pin 29

Reitti loppunut:

<u>Digitaaliset sisään</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
1. Releen apukosketin	1	USB-1208FS / Port A0 / Pin 21
2. Releen apukosketin	1	USB-1208FS / Port A1 / Pin 22
Kallistuksen eturaja	0	USB-1208FS / Port A2 / Pin 23
Kallistuksen takaraja	0	USB-1208FS / Port A3 / Pin 24
Kallistuksen vasen raja	0	USB-1208FS / Port A4 / Pin 25
Kallistuksen oikea raja	0	USB-1208FS / Port A5 / Pin 26

<u>Digitaaliset ulos</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Taajuusmuuttaja 1:n moottorin ajo eteenpäin		USB-3101 / DIO0
Taajuusmuuttaja 1:n moottorin ajo taaksepäin		USB-3101 / DIO1
Taajuusmuuttaja 2:n moottorin ajo eteenpäin		USB-3101 / DIO2
Taajuusmuuttaja 2:n moottorin ajo taaksepäin		USB-3101 / DIO3

<u>Analogiset sisään</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Kallistusanturi eteen/taakse		USB-1208FS / CH0 / Pin 1
Kallistusanturi sivuille		USB-1208FS / CH1 / Pin 4

<u>Analogiset ulos</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Taajuusmuuttaja 1:n taajuusohje		USB-3101 / VOUT0 / Pin 1
Taajuusmuuttaja 2:n taajuusohje		USB-3101 / VOUT1 / Pin 29

Häiriö ilmaantunut (rajalle ajo)

<u>Digitaaliset sisään</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
1. Releen apukosketin	0	USB-1208FS / Port A0 / Pin 21
2. Releen apukosketin	1	USB-1208FS / Port A1 / Pin 22
Kallistuksen eturaja	1 !	USB-1208FS / Port A2 / Pin 23
Kallistuksen takaraja	0	USB-1208FS / Port A3 / Pin 24
Kallistuksen vasen raja	1 !	USB-1208FS / Port A4 / Pin 25
Kallistuksen oikea raja	0	USB-1208FS / Port A5 / Pin 26

<u>Digitaaliset ulos</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Taajuusmuuttaja 1:n moottorin ajo eteenpäin	0 OK	USB-3101 / DIO0
Taajuusmuuttaja 1:n moottorin ajo taaksepäin	1 OK	USB-3101 / DIO1
Taajuusmuuttaja 2:n moottorin ajo eteenpäin	1 OK	USB-3101 / DIO2
Taajuusmuuttaja 2:n moottorin ajo taaksepäin	0 OK	USB-3101 / DIO3

<u>Analogiset sisään</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Kallistusanturi eteen/taakse	x	USB-1208FS / CH0 / Pin 1
Kallistusanturi sivuille	x	USB-1208FS / CH1 / Pin 4

<u>Analogiset ulos</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Taajuusmuuttaja 1:n taajuusohje	0,75v = 3,75Hz	USB-3101 / VOUT0 / Pin 1
Taajuusmuuttaja 2:n taajuusohje	0,75v = 3,75Hz	USB-3101 / VOUT1 / Pin 29

Häiriö ilmaantunut (häätäpainike)

<u>Digitaaliset sisään</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
1. Releen apukosketin	1	USB-1208FS / Port A0 / Pin 21
2. Releen apukosketin	0 !	USB-1208FS / Port A1 / Pin 22
Kallistuksen eturaja	0	USB-1208FS / Port A2 / Pin 23
Kallistuksen takaraja	0	USB-1208FS / Port A3 / Pin 24
Kallistuksen vasen raja	0	USB-1208FS / Port A4 / Pin 25
Kallistuksen oikea raja	0	USB-1208FS / Port A5 / Pin 26

<u>Digitaaliset ulos</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Taajuusmuuttaja 1:n moottorin ajo eteenpäin	0 OK	USB-3101 / DIO0
Taajuusmuuttaja 1:n moottorin ajo taakse-päin	0 OK	USB-3101 / DIO1
Taajuusmuuttaja 2:n moottorin ajo eteenpäin	0 OK	USB-3101 / DIO2
Taajuusmuuttaja 2:n moottorin ajo taakse-päin	0 OK	USB-3101 / DIO3

<u>Analogiset sisään</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Kallistusanturi eteen/taakse	x	USB-1208FS / CH0 / Pin 1
Kallistusanturi sivuille	x	USB-1208FS / CH1 / Pin 4

<u>Analogiset ulos</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Taajuusmuuttaja 1:n taajuusohje	0,75v OK	USB-3101 / VOUT0 / Pin 1
Taajuusmuuttaja 2:n taajuusohje	0,75v OK	USB-3101 / VOUT1 / Pin 29

Häiriötilasta poistuttu (rajalle ajo)

<u>Digitaaliset sisään</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
1. Releen apukosketin	1	USB-1208FS / Port A0 / Pin 21
2. Releen apukosketin	1	USB-1208FS / Port A1 / Pin 22
Kallistuksen eturaja	0	USB-1208FS / Port A2 / Pin 23
Kallistuksen takaraja	0	USB-1208FS / Port A3 / Pin 24
Kallistuksen vasen raja	0	USB-1208FS / Port A4 / Pin 25
Kallistuksen oikea raja	0	USB-1208FS / Port A5 / Pin 26

<u>Digitaaliset ulos</u>	<u>Tila</u>	<u>I/O purkki / liittimet</u>
Taajuusmuuttaja 1:n moottorin ajo eteenpäin	0	USB-3101 / DIO0
Taajuusmuuttaja 1:n moottorin ajo taakse-päin	0	USB-3101 / DIO1
Taajuusmuuttaja 2:n moottorin ajo eteenpäin	0	USB-3101 / DIO2
Taajuusmuuttaja 2:n moottorin ajo taakse-päin	0	USB-3101 / DIO3

Analogiset sisään

Kallistusanturi eteen/taakse
Kallistusanturi sivuille

Tila

x
x

I/O purkki / liittimet

USB-1208FS / CH0 / Pin 1
USB-1208FS / CH1 / Pin 4

Analogiset ulos

Taajuusmuuttaja 1:n taajuusohje
Taajuusmuuttaja 2:n taajuusohje

Tila

0,75v OK
0,75v OK

I/O purkki / liittimet

USB-3101 / VOUT0 / Pin 1
USB-3101 / VOUT1 / Pin 29

Häiriötilasta poistuttu (häätäpainike)**Digitaaliset sisään**

1. Releen apukosketin
2. Releen apukosketin
Kallistuksen eturaja
Kallistuksen takaraja
Kallistuksen vasen raja
Kallistuksen oikea raja

Tila

1
1
0
0
0
0

I/O purkki / liittimet

USB-1208FS / Port A0 / Pin 21
USB-1208FS / Port A1 / Pin 22
USB-1208FS / Port A2 / Pin 23
USB-1208FS / Port A3 / Pin 24
USB-1208FS / Port A4 / Pin 25
USB-1208FS / Port A5 / Pin 26

Digitaaliset ulos

Taajuusmuuttaja 1:n moottorin ajo eteenpäin
Taajuusmuuttaja 1:n moottorin ajo taakse-
päin
Taajuusmuuttaja 2:n moottorin ajo eteenpäin
Taajuusmuuttaja 2:n moottorin ajo taakse-
päin

Tila

0 OK
0 OK
0 OK
0 OK

I/O purkki / liittimet

USB-3101 / DIO0
USB-3101 / DIO1
USB-3101 / DIO2
USB-3101 / DIO3

Analogiset sisään

Kallistusanturi eteen/taakse
Kallistusanturi sivuille

Tila

x
x

I/O purkki / liittimet

USB-1208FS / CH0 / Pin 1
USB-1208FS / CH1 / Pin 4

Analogiset ulos

Taajuusmuuttaja 1:n taajuusohje
Taajuusmuuttaja 2:n taajuusohje

Tila

0,75v OK
0,75v OK

I/O purkki / liittimet

USB-3101 / VOUT0 / Pin 1
USB-3101 / VOUT1 / Pin 29

KUPARIJOHTIMIEN KOOT JA NIIDEN KUORMITETTAVUUS

LIITE 4

Johdin	Asennustapa A	Asennustapa C	Asennustapa D	Asennustapa E	Yleisin sulakekoko (automaatti/tulppa/kahva)
1,5 mm ²	14 A	18,5 A	26 A	19 A	10 A
2,5 mm ²	19 A	25 A	35 A	26 A	16 A
4 mm ²	24 A	34 A	46 A	36 A	20 A
6 mm ²	31 A	43 A	57 A	45 A	25-32 A
10 mm ²	41 A	60 A	77 A	63 A	32-35 A
16 mm ²	55 A	80 A	100 A	85 A	50 A
25 mm ²	72 A	102 A	130 A	107 A	63 A
35 mm ²	88 A	126 A	160 A	134 A	80 A
50 mm ²	105 A	153 A	190 A	162 A	100 A
70 mm ²	133 A	195 A	240 A	208 A	125 A
95 mm ²	159 A	236 A	285 A	252 A	125-160 A
120 mm ²	182 A	274 A	325 A	292 A	160 A
150 mm ²	208 A	317 A	370 A	338 A	160-200 A
185 mm ²	236 A	361 A	420 A	386 A	200 A
240 mm ²	278 A	427 A	480 A	456 A	250 A
300 mm ²	316 A	492 A	550 A	527 A	315 A


```
Imports Microsoft.Office.Interop
Imports System.Math
```

```
'Sovellus sisältää tämän ohjelmointikoodin lisäksi Design-osion puolella olevan käyttöliit-
tymän. Sovelluksessa on käytetty
'paljon backgroundworker-komponentteja, joilla mahdollistetaan koodin suorittaminen tausta-
prosessina. Background-workerit trigataan
'käytiin Design-osion puolella määritettyjen ajastimien avulla. Tiedonkeruulaitteiden USB-
1208 ja 3101 manuaaleista löytyy
'ohjeistus ja näyteohjelmat kuinka I/O:ta luetaan ja kirjoitetaan. Tämä sovellus saattaa si-
sältää paljonkin useaan kertaan määriteltyjä
'I/O-portteja. Voi olla että on mahdollista määritellä kaikkien porttien tiedot kerralla
esim. Public Class Form1:ssä.
```

```
Public Class Form1
```

```
    Public moneskokerta1 As Integer
    Public moneskokerta2 As Integer
    Public moneskokerta3 As Integer
    Public moneskokerta4 As Integer

    'IO-boksien alustaminen
    Private USB1208 As MccDaq.MccBoard = New MccDaq.MccBoard(0) 'Määritä suluissa olevat nu-
meroinnit InstaCalin antamien tunnistusnumeroiden perusteella
    Private USB3101 As MccDaq.MccBoard = New MccDaq.MccBoard(1)

    'I0 boksien muuttujat
    'Digitaaliset sisään
    Private DigisisaanDirection As MccDaq.DigitalPortDirection
    Private A_PortNum As MccDaq.DigitalPortType

    'Digitaaliset ulos
    Private DigiulosDirection As MccDaq.DigitalPortDirection
    Private Tamu1eteenPortNum As MccDaq.DigitalPortType
    Private Tamu1taaksePortNum As MccDaq.DigitalPortType
    Private Tamu2eteenPortNum As MccDaq.DigitalPortType
    Private Tamu2taaksePortNum As MccDaq.DigitalPortType
    'Analogiset sisään
    Private ADResolution, NumAICHans, HighChan As Integer
    Private AnasisaanRange As MccDaq.Range
    'Analogiset ulos
    Private AnaulosRange As MccDaq.Range
    Private Tamu1taajuusohjeChannelNum As Integer = 0
    Private Tamu2taajuusohjeChannelNum As Integer = 1

    'Yleisiä muuttujia

    Dim videonopeus As Double = 1.0 '1.0 = 100%
    Dim reittitaulukko(9000, 1) As Integer 'Taulukko johon tiedot ulkoisesta kallistuarvo-
taulukosta tuodaan. Muista asettaa riviarvo
    Dim saatimenlupa As Integer = 1 'vastaamaan ulkoisen taulukkotiedoston rivimää-
rää, jotta tämä taulukko ei käy pieneksi
    Dim aikaleimaX As Integer
    Dim aikaleimaY As Integer
    Dim taulukkonaytevali As Integer = 1000 'Kallisteluarvotaulukossa käytetty näyteväli
millisekunneissa
    Dim paikkataulukossadoubleX As Double
    Dim paikkataulukossadoubleY As Double
    Dim paikkataulukossaintX As Integer
    Dim paikkataulukossaintY As Integer
    Dim taulukonarvosaatimelleX As Integer
    Dim taulukonarvosaatimelleY As Integer

    'Saatimen X eli eteen ja taakse liikehdintään liittyvät muuttujat alla

    Public lastTimeX As Integer
    Dim InputX As Double
    Dim OutputX As Double
```

```

Dim SetpointX As Double
Dim errSumX As Double
Public lastErrX As Double
Dim kpX As Double = 1.5 'SaadinX viritysparametrit
Dim kiX As Double = 0.5
Dim kdX As Double = 0.5
Dim nowX As Integer
Dim timeChangeX As Double
Dim errorX As Double
Dim dErrX As Double

'Saatimen Y eli sivuliikehdintään liittyvät muuttujat alla

Public lastTimeY As Integer
Dim InputY As Double
Dim OutputY As Double
Dim SetpointY As Double
Dim errSumY As Double
Public lastErrY As Double
Dim kpY As Double = 0.1 'SaadinY viritysparametrit
Dim kiY As Double = 0.1
Dim kdY As Double = 0.1
Dim nowY As Integer
Dim timeChangeY As Double
Dim errorY As Double
Dim dErrY As Double

Public stopnappi As Integer

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Button_start.Enabled = False 'Evätään käyttäjältä lupa painaa muuta kuin "Lataa reitti 1"-painiketta
    'ennen kuin reitin kallistelum tiedot on ladattu ohjelmaan
    'ulkoisesta kallistusarvot sisältävästä taulukosta
    Button_pause.Enabled = False
    Button_stop.Enabled = False
    Button_nopea.Enabled = False
    Button_hidas.Enabled = False

    'Alustetaan analogia sisan kanavat
    Dim LowChan As Integer
    Dim ChannelType As Integer
    Dim DefaultTrig As MccDaq.TriggerType

    ChannelType = 1

    NumAIChans = FindAnalogChansOfType(USB1208, ChannelType, _
    ADResolution, AnasisaanRange, LowChan, DefaultTrig)

End Sub

Private Sub Form1_FormClosing(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Me.FormClosing

    DigiulosDirection = MccDaq.DigitalPortDirection.DigitalOut 'Varmistetaan moottorien pysähtyminen kun ohjelmasta poistutaan painamalla X-painiketta
    Dim Digiulospaalle As MccDaq.DigitalLogicState
    Digiulospaalle = MccDaq.DigitalLogicState.High
    Dim Digiulospois As MccDaq.DigitalLogicState
    Digiulospois = MccDaq.DigitalLogicState.Low
    Tamu1eteenPortNum = MccDaq.DigitalPortType.AuxPort
    Tamu1taaksePortNum = MccDaq.DigitalPortType.AuxPort
    Tamu2eteenPortNum = MccDaq.DigitalPortType.AuxPort
    Tamu2taaksePortNum = MccDaq.DigitalPortType.AuxPort

    Dim Tamu1taajuusohjearvo As UInt16

```

```

Tamultaajuusohjearvo = 0
Dim Tamu2taajuusohjearvo As UInt16
Tamu2taajuusohjearvo = 0

AxVLCPlugin21.playlist.stop()
Saatimet.Stop()

USB3101.DConfigBit(Tamu1eteenPortNum, 0, DigiulosDirection)
USB3101.DBitOut(Tamu1eteenPortNum, 0, Digiulospois)

USB3101.DConfigBit(Tamu1taaksePortNum, 1, DigiulosDirection)
USB3101.DBitOut(Tamu1taaksePortNum, 1, Digiulospois)

USB3101.DConfigBit(Tamu2eteenPortNum, 2, DigiulosDirection)
USB3101.DBitOut(Tamu2eteenPortNum, 2, Digiulospois)

USB3101.DConfigBit(Tamu2taaksePortNum, 3, DigiulosDirection)
USB3101.DBitOut(Tamu2taaksePortNum, 3, Digiulospois)

USB3101.AOut(Tamu1taajuusohjeChannelNum, AnaulosRange, Tamu1taajuusohjearvo)
USB3101.AOut(Tamu2taajuusohjeChannelNum, AnaulosRange, Tamu2taajuusohjearvo)

End Sub

Private Sub Button_start_Click(ByVal sender As System.Object, ByVal e As Sys-
tem.EventArgs) Handles Button_start.Click
    'Alla oleva koodi suoritetaan aina kun Käynnistä-nappia painetaan käyttöliittymästä

    Dim ajosekv As Integer = 0
    Do While ajosekv <> 3

        If ajosekv = 0 Then
            MsgBox("Hyppäähän ratsaille ja laita rillit päähän. Paina ok kun olet val-
            mis")
            ajosekv = ajosekv + 1
        End If

        If ajosekv = 1 Then
            AxVLCPlugin21.playlist.add("GoPro.mp4") 'Maastajuoksureitin video (videon
            nimi voidaan kirjoittaa ilman polkua jos sijaitsee Debug kansiossa)
            AxVLCPlugin21.playlist.next()
            AxVLCPlugin21.playlist.play()
            AxVLCPlugin21.video.toggleFullscreen()
            ajosekv = ajosekv + 1
        End If

        If ajosekv = 2 Then
            Button_reitti.Enabled = False 'Annetaan käyttäjälle lupa painaa vain tiet-
            tyjä painikkeita käyttöliittymästä
            Button_pause.Enabled = True
            Button_stop.Enabled = True
            Button_nopea.Enabled = True
            Button_hidas.Enabled = True
            Label8.Text = "Voit nopeuttaa/hidastaa/keskeyttää/pysäyttää videon"
            Saatimet.Start() 'Käynnistetään säätimiä triggaava kello
            Taustatarkkailu.Start()
            ajosekv = ajosekv + 1
        End If

    Loop

End Sub

Private Sub Button_nopea_Click(ByVal sender As System.Object, ByVal e As Sys-
tem.EventArgs) Handles Button_nopea.Click
    videonopeus = videonopeus + 0.2 'Nopeutus-napin painallus lisää 20% nopeutta
    AxVLCPlugin21.input.rate = videonopeus
End Sub

```

```
Private Sub Button_hidas_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button_hidas.Click
```

```
videonopeus = videonopeus - 0.2 'Hidastus-napin painallus vähentää 20% nopeutta
AxVLCPlugin21.input.rate = videonopeus
End Sub
```

```
Private Sub Button_reitti_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button_reitti.Click
```

```
'Tämän aliohjelman suoritus käynnistyy klikkaamalla "lataa reitti" -nappia ohjelmas-
ta
```

```
Dim xlApp As Microsoft.Office.Interop.Excel.Application
Dim xlWorkBook As Microsoft.Office.Interop.Excel.Workbook
Dim xlWorkSheet As Microsoft.Office.Interop.Excel.Worksheet
Dim looppi As Integer
Dim laskuri As Integer
laskuri = 0
Dim arvoX As Integer
Dim arvoY As Integer
```

```
xlApp = New Microsoft.Office.Interop.Excel.Application
xlWorkBook = xlApp.Workbooks.Open("C:\MJSP\test3.xls") 'Maastajuoksureitin kulma-
arvot HOX. muuttuja 'taulukkonaytevali' näytteenottotaajuuden asetusta varten (ms)
xlWorkSheet = xlWorkBook.Worksheets("sheet1")
```

```
For looppi = 0 To 200 Step 1 'HOX! Muista asettaa arvon "200" tilalle ulkoisen
taulukkotiedoston sisältämien rivien määrä
```

```
laskuri = laskuri + 1
arvoX = xlWorkSheet.Cells(laskuri, 1).value
arvoY = xlWorkSheet.Cells(laskuri, 2).value
reittitaulukko(laskuri, 0) = arvoX
reittitaulukko(laskuri, 1) = arvoY
```

```
Next looppi
xlWorkBook.Close()
xlApp.Quit()
```

```
releaseObject(xlApp)
releaseObject(xlWorkBook)
releaseObject(xlWorkSheet)
MsgBox("Reitti ladattu onnistuneesti")
Label18.Text = "Aloita simulointi painamalla Käynnistä-painiketta"
Button_start.Enabled = True
```

```
End Sub
```

```
Private Sub releaseObject(ByVal obj As Object)
```

```
Try
System.Runtime.InteropServices.Marshal.ReleaseComObject(obj)
obj = Nothing
Catch ex As Exception
obj = Nothing
Finally
GC.Collect()
End Try
```

```
End Sub
```

```
Private Sub Saatimet_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Saatimet.Tick
```

```
'Aina kun Saatimet-ajastin lyö, suoritetaan SaadinX:n ja SaadinY:n backgroundworke-
rit(SaadinX_DoWork & SaadinY_DoWork)
```

```
'Saatimet-ajastimen lyöntihetkeä voit säätää Design-osion puolelta
```

```

If SaadinX.IsBusy = False Then
    SaadinX.RunWorkerAsync()
End If

If SaadinY.IsBusy = False Then
    SaadinY.RunWorkerAsync()
End If

End Sub

Private Sub Taustatarkkailu_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Taustatarkkailu.Tick
    If Taustatarkkailija.IsBusy = False Then
        Taustatarkkailija.RunWorkerAsync()
    End If
End Sub

Private Sub SaadinX_DoWork(ByVal sender As System.Object, ByVal e As System.ComponentModel.DoWorkEventArgs) Handles SaadinX.DoWork

    Dim KallistusarvoXakseli As System.UInt16
    Dim KallistusarvoXakseliChan As Integer
    Dim KallistusarvoXakseliEngUnits As Single
    KallistusarvoXakseliChan = 0

    Dim Tamultaajuusohjearvo As UInt16

    DigiulosDirection = MccDaq.DigitalPortDirection.DigitalOut
    Dim Digiulospaalle As MccDaq.DigitalLogicState
    Digiulospaalle = MccDaq.DigitalLogicState.High
    Dim Digiulospois As MccDaq.DigitalLogicState
    Digiulospois = MccDaq.DigitalLogicState.Low
    TamuleteenPortNum = MccDaq.DigitalPortType.AuxPort
    TamultaaksePortNum = MccDaq.DigitalPortType.AuxPort

    'luetaan videon aikaleima ja haetaan leimaa vastaava arvo taulukosta. Muista asettaa
    ulkoisessa kallisteluvarvotaulukossa käytetty näyttөөntoväliä määrittävä muuttuja "tauluk-
    konäytevali" oikeaksi
    aikaleimaX = AxVLCPlugin21.input.Time
    paikkataulukossadoubleX = aikaleimaX / taulukkonaytevali
    paikkataulukossaintX = paikkataulukossadoubleX 'Pyöristetään kokonaisluvuksi poista-
    malla desimaalit
    taulukonarvosaatimelleX = reittitaulukko(paikkataulukossaintX, 0)

    'haetaan kallistusanturin arvo
    USB1208.AIn(KallistusarvoXakseliChan, AnasisaanRange, KallistusarvoXakseli)
    USB1208.ToEngUnits(AnasisaanRange, KallistusarvoXakseli, KallistusarvoXakseliEn-
    gUnits)

    InputX = KallistusarvoXakseliEngUnits

    If InputX >= 0 And InputX <= 1.5 Then 'Kallisteluanturi tuottaa ulostuloonsa 0-3v.
Tässä skaalataan 0-1.5v = 0-(-15) &
        InputX = (1.5 - InputX) * (-10) '1.5-3.0 = 0-15
    End If

    If InputX > 1.5 And InputX < 3.1 Then
        InputX = (InputX - 1.5) * 10
    End If

    'PID säädin X

    'Kauanko aikaa kulunut viime ohjauksesta
    nowX = Date.Now.TimeOfDay.TotalMilliseconds

```

```

    If lastTimeX = 0 Then 'Tämä suoritetaan ensimmäisellä laskukerralla. Ilman tätä säädin
    tuottaa alkuun mielettömän suuren timechangeX:n joka jumittaa säätimen
        lastTimeX = nowX - 1
    End If
    timeChangeX = nowX - lastTimeX

    'Lasketaan muuttujille arvot säätimeen
    errorX = taulukonarvosaaatimelleX - InputX 'asetusarvo-anturiarvo
    errSumX = errSumX + (errorX * timeChangeX) 'integrointilaskenta
    dErrX = (errorX - lastErrX) / timeChangeX 'derivointilaskenta

    'PID säätimen ulostulo
    OutputX = kpX * errorX + kiX * errSumX + kdX * dErrX 'Säätimen P, I ja D osien
    painotuskertoimet(kpX, kiX ja kdX) määritellään ohjelman alussa

    e.Result = OutputX 'Tämä arvo viedään backgroundworkerista ulos Saa-
    dinX_RunWorkerCompleted aliohjelmaan

    If OutputX > 65000 Then 'DAQ ei hyväksy yli 65000 olevia analogisen ulostulon ase-
    tusarvoja
        OutputX = 65000
    End If

    If OutputX < -65000 Then
        OutputX = -65000
    End If

    If OutputX < 0 Then 'Säätimen ulostulon asettaminen tamun taajuusohjeksi. Tässä
    myös estetään (-)merkkisen ulostulon käyttö. Negatiivinen taajuusohjearvo otetaan huomioon
    alla olevassa moottorin pyörimssuunnan ohjauksessa
        Tamu1taajuusohjearvo = OutputX * (-1)
    Else
        Tamu1taajuusohjearvo = OutputX
    End If

    USB3101.AOut(Tamu1taajuusohjeChannelNum, AnulosRange, Tamu1taajuusohjearvo)

    'Ajetaanko moottoriaeteen vai taakse
    If OutputX < 0 Then
        USB3101.DConfigBit(Tamu1eteenPortNum, 0, DigiulosDirection)
        USB3101.DBitOut(Tamu1eteenPortNum, 0, Digiulospaalle)
        USB3101.DConfigBit(Tamu1taaksePortNum, 1, DigiulosDirection)
        USB3101.DBitOut(Tamu1taaksePortNum, 1, Digiulospois)
    Else
        USB3101.DConfigBit(Tamu1eteenPortNum, 0, DigiulosDirection)
        USB3101.DBitOut(Tamu1eteenPortNum, 0, Digiulospois)
        USB3101.DConfigBit(Tamu1taaksePortNum, 1, DigiulosDirection)
        USB3101.DBitOut(Tamu1taaksePortNum, 1, Digiulospaalle)
    End If

    'Arvot talteen seuraavaa tämän backgroundworkerin suorituskertaa varten
    lastErrX = errorX
    lastTimeX = nowX

End Sub

Private Sub SaadinX_RunWorkerCompleted(ByVal sender As System.Object, _
ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs) _
Handles SaadinX.RunWorkerCompleted
    'Kutsutaan kun SaadinX valmistuu

    If e.Result < 0 Then 'Säätimen ulostuloarvon esitys käyttöliittymässä
        Label1.Text = e.Result * (-1)
    Else
        Label1.Text = e.Result
    End If

```

```

If e.Result < 0 Then 'Moottorin ajosuunnan esitys ohjelmassa
    Label5.Text = "Eteen"
Else
    Label5.Text = "Taakse"
End If

End Sub

Private Sub SaadinY_DoWork(ByVal sender As System.Object, ByVal e As Sys-
tem.ComponentModel.DoWorkEventArgs) Handles SaadinY.DoWork

    Dim KallistusarvoYakseli As System.UInt16
    Dim KallistusarvoYakseliChan As Integer
    Dim KallistusarvoYakseliEngUnits As Single
    KallistusarvoYakseliChan = 1

    Dim Tamu2taajuusohjearvo As UInt16

    DigiulosDirection = MccDaq.DigitalPortDirection.DigitalOut
    Dim Digiulospaalle As MccDaq.DigitalLogicState
    Digiulospaalle = MccDaq.DigitalLogicState.High
    Dim Digiulospois As MccDaq.DigitalLogicState
    Digiulospois = MccDaq.DigitalLogicState.Low
    Tamu2eteenPortNum = MccDaq.DigitalPortType.AuxPort
    Tamu2taaksePortNum = MccDaq.DigitalPortType.AuxPort

    'luetaan videon aikaleima ja haetaan leimaa vastaava arvo taulukosta
    aikaleimaY = AxVLCPlugin21.input.Time
    paikkataulukossadoubleY = aikaleimaY / taulukkonaytevali
    paikkataulukossaintY = paikkataulukossadoubleY
    taulukonarvosaatimelleY = reittitaulukko(paikkataulukossaintY, 1)

    'haetaan anturin arvo
    USB1208.AIn(KallistusarvoYakseliChan, AnasisaanRange, KallistusarvoYakseli)
    USB1208.ToEngUnits(AnasisaanRange, KallistusarvoYakseli, KallistusarvoYakseliEn-
gUnits)
    InputY = KallistusarvoYakseliEngUnits

    If InputY >= 0 And InputY <= 1.5 Then 'Kallisteluanturi tuottaa ulostuloonsa 0-3v.
Tässä skaalataan 0v-1.5v = 0-(-15) &
        InputY = (1.5 - InputY) * (-10) '1.5v-3.0v = 0-15. Kallistuksen tasapai-
noasema on siis silloin kun anturi antaa ulostuloon 1.5V
    End If

    If InputY > 1.5 And InputY < 3.1 Then
        InputY = (InputY - 1.5) * 10
    End If

    'PID säädin Y

    'Kauanko viime kerrasta
    nowY = Date.Now.TimeOfDay.TotalMilliseconds
    If lastTimeY = 0 Then 'Ilman tätä säädin tuottaa alkuun mielettömän suuren ti-
mechangeY:n joka jumittaa säätimen. Ts. Tamu2taajuusohjearvo menee yli 65000
        lastTimeY = nowY - 1
    End If
    timeChangeY = nowY - lastTimeY

    'Säätimen muuttujat
    errorY = taulukonarvosaatimelleY - InputY
    errSumY = errSumY + (errorY * timeChangeY)
    dErrY = (errorY - lastErrY) / timeChangeY

    'Ulostulo
    OutputY = kpY * errorY + kiY * errSumY + kdY * dErrY
    e.Result = OutputY

```

```
If OutputY > 65000 Then 'Arithmetic overflown estämiseksi
    OutputY = 65000
End If

If OutputY < -65000 Then 'Arithmetic overflown estämiseksi
    OutputY = -65000
End If

If OutputY < 0 Then
    Tamu2taajuusohjearvo = OutputY * (-1)
Else
    Tamu2taajuusohjearvo = OutputY
End If

USB3101.AOut(Tamu2taajuusohjeChannelNum, AnulosRange, Tamu2taajuusohjearvo)

'Ajetaanko eteen vai taakse. Eteen oikealle, taakse vasemmalle
If OutputY < 0 Then
    USB3101.DConfigBit(Tamu2eteenPortNum, 2, DigiulosDirection)
    USB3101.DBitOut(Tamu2eteenPortNum, 2, Digiulospaalle)
    USB3101.DConfigBit(Tamu2taaksePortNum, 3, DigiulosDirection)
    USB3101.DBitOut(Tamu2taaksePortNum, 3, Digiulospois)
Else
    USB3101.DConfigBit(Tamu2eteenPortNum, 2, DigiulosDirection)
    USB3101.DBitOut(Tamu2eteenPortNum, 2, Digiulospois)
    USB3101.DConfigBit(Tamu2taaksePortNum, 3, DigiulosDirection)
    USB3101.DBitOut(Tamu2taaksePortNum, 3, Digiulospaalle)
End If

'Arvot talteen seuraavaa kertaa varten
lastErrY = errorY
lastTimeY = nowY

End Sub

Private Sub SaadinY_RunWorkerCompleted(ByVal sender As System.Object, _
ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs) _
Handles SaadinY.RunWorkerCompleted
    'Tämä suoritetaan kun SaadinY valmistuu
    If e.Result < 0 Then
        Label4.Text = e.Result * (-1)
    Else
        Label4.Text = e.Result
    End If
    If e.Result < 0 Then
        Label6.Text = "Vasemmalle"
    Else
        Label6.Text = "Oikealle"
    End If
End Sub

End Sub

Private Sub Taustatarkkailija_DoWork(ByVal sender As System.Object, ByVal e As Sys-
tem.ComponentModel.DoWorkEventArgs) Handles Taustatarkkailija.DoWork
    'luetaan releiden apuskoskettimien tilat
    'jos jompikumpi lauennut niin Saatimet.stop() ja asetetaan taajareille 0 taajuutta
    ja ajolupa
    'evätään + video stopataan ja käynnistetään Hairiotilanne aliohjelma

    DigisisaanDirection = MccDaq.DigitalPortDirection.DigitalIn
    A_PortNum = MccDaq.DigitalPortType.FirstPortA
    Dim Ekarelearvo As MccDaq.DigitalLogicState
    'TokarelePortNum = MccDaq.DigitalPortType.SecondPortA
    Dim Tokarelearvo As MccDaq.DigitalLogicState

    DigiulosDirection = MccDaq.DigitalPortDirection.DigitalOut
    Dim Digiulospaalle As MccDaq.DigitalLogicState
    Digiulospaalle = MccDaq.DigitalLogicState.High
```



```

Dim Digiulospois As MccDaq.DigitalLogicState
Digiulospois = MccDaq.DigitalLogicState.Low
Tamu1eteenPortNum = MccDaq.DigitalPortType.AuxPort
Tamu1taaksePortNum = MccDaq.DigitalPortType.AuxPort
Tamu2eteenPortNum = MccDaq.DigitalPortType.AuxPort
Tamu2taaksePortNum = MccDaq.DigitalPortType.AuxPort

Dim Tamu1taajuusohjearvo As UInt16
Tamu1taajuusohjearvo = 0
Dim Tamu2taajuusohjearvo As UInt16
Tamu2taajuusohjearvo = 0

USB1208.DConfigPort(A_PortNum, DigisisaanDirection)
USB1208.DBitIn(A_PortNum, 0, Ekarelearvo)

USB1208.DConfigPort(A_PortNum, DigisisaanDirection)
USB1208.DBitIn(A_PortNum, 1, Tokarelearvo)

e.Result = Tokarelearvo + Ekarelearvo

If (Ekarelearvo + Tokarelearvo) < 2 Then 'Tämä suoritetaan jos jompikumpi relei-
den luvuista mennyt nolllaksi

    e.Result = 1111
    Saatimet.Stop()

    USB3101.DConfigBit(Tamu1eteenPortNum, 0, DigiulosDirection)
    USB3101.DBitOut(Tamu1eteenPortNum, 0, Digiulospois)

    USB3101.DConfigBit(Tamu1taaksePortNum, 1, DigiulosDirection)
    USB3101.DBitOut(Tamu1taaksePortNum, 1, Digiulospois)

    USB3101.DConfigBit(Tamu2eteenPortNum, 2, DigiulosDirection)
    USB3101.DBitOut(Tamu2eteenPortNum, 2, Digiulospois)

    USB3101.DConfigBit(Tamu2taaksePortNum, 3, DigiulosDirection)
    USB3101.DBitOut(Tamu2taaksePortNum, 3, Digiulospois)

    USB3101.AOut(Tamu1taajuusohjeChannelNum, AnaulosRange, Tamu1taajuusohjearvo)
    USB3101.AOut(Tamu2taajuusohjeChannelNum, AnaulosRange, Tamu2taajuusohjearvo)

    lastErrX = 0
    lastErrY = 0
    lastTimeX = 0
    lastTimeY = 0
    errSumX = 0
    errSumY = 0

End If
End Sub

Private Sub Taustatarkkailija_RunWorkerCompleted(ByVal sender As System.Object, _
ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs) _
Handles Taustatarkkailija.RunWorkerCompleted
    'Tämä suoritetaan kun Taustatarkkailija_DoWork valmistuu
    Label7.Text = e.Result

    If e.Result = 1111 Then 'Jos jompikumpi rele lauennut, suoritetaan tämä. e.Resultin
arvo saadaan Taustatarkkailija_DoWorkin puolelta
        AxVLCPlugin21.playlist.togglePause()
        Taustatarkkailu.Stop()
        Hairiotilannetimer.Start()
    End If
End Sub

Private Sub Hairiotilannetimer_Tick(ByVal sender As System.Object, ByVal e As Sys-
tem.EventArgs) Handles Hairiotilannetimer.Tick

```

```
'Tämän kellon lyönti käynnistyy jos Taustatarkkailija-Dowork on havainnut releen lau-
keamisen
    If Hairio.IsBusy = False Then
        Hairio.RunWorkerAsync()
    End If

End Sub

Private Sub Hairio_DoWork(ByVal sender As System.Object, ByVal e As Sys-
tem.ComponentModel.DoworkEventArgs) Handles Hairio.DoWork

    DigisisaanDirection = MccDaq.DigitalPortDirection.DigitalIn

    Dim Ekarelearvo As MccDaq.DigitalLogicState

    Dim Tokarelearvo As MccDaq.DigitalLogicState

    A_PortNum = MccDaq.DigitalPortType.FirstPortA

    Dim Kallistuseturajaarvo As MccDaq.DigitalLogicState
    Dim Kallistustakarajaarvo As MccDaq.DigitalLogicState
    Dim Kallistusvasenrajaarvo As MccDaq.DigitalLogicState
    Dim Kallistusoikearajaarvo As MccDaq.DigitalLogicState

    DigiulosDirection = MccDaq.DigitalPortDirection.DigitalOut
    Dim Digiulospaalle As MccDaq.DigitalLogicState
    Digiulospaalle = MccDaq.DigitalLogicState.High
    Dim Digiulospois As MccDaq.DigitalLogicState
    Digiulospois = MccDaq.DigitalLogicState.Low
    Tamu1eteenPortNum = MccDaq.DigitalPortType.AuxPort
    Tamu1taaksePortNum = MccDaq.DigitalPortType.AuxPort
    Tamu2eteenPortNum = MccDaq.DigitalPortType.AuxPort
    Tamu2taaksePortNum = MccDaq.DigitalPortType.AuxPort

    Dim Tamu1taajuusohjearvo As UInt16
    Tamu1taajuusohjearvo = 5000
    Dim Tamu2taajuusohjearvo As UInt16
    Tamu2taajuusohjearvo = 5000
    'Jos häiriön aiheutti rajalle ajo niin suoritetaan tilanne 1. Jos hätäpainike niin
tilanne 2

    USB1208.DConfigPort(A_PortNum, DigisisaanDirection)
    USB1208.DBitIn(A_PortNum, 0, Ekarelearvo)

    USB1208.DConfigPort(A_PortNum, DigisisaanDirection)
    USB1208.DBitIn(A_PortNum, 1, Tokarelearvo)

    USB1208.DConfigPort(A_PortNum, DigisisaanDirection)
    USB1208.DBitIn(A_PortNum, 2, Kallistuseturajaarvo)

    USB1208.DConfigPort(A_PortNum, DigisisaanDirection)
    USB1208.DBitIn(A_PortNum, 3, Kallistustakarajaarvo)

    USB1208.DConfigPort(A_PortNum, DigisisaanDirection)
    USB1208.DBitIn(A_PortNum, 4, Kallistusvasenrajaarvo)

    USB1208.DConfigPort(A_PortNum, DigisisaanDirection)
    USB1208.DBitIn(A_PortNum, 5, Kallistusoikearajaarvo)

    'Jos rele 1:n tila 0 niin tarkastetaan reed-antureilla kummalle rajalle ajettu.
    'Tarkastetaan jatkuvasti onko rele 1 kytkeytynyt johtavaksi. Kun rele 1:n tila muut-
tuu 1 niin
    'annetaan pieni taajuus taajarille joka ohjaa sitä suuntaa jossa hairio tapahtui
    'annetaan ajolupa oikeaan suuntaan (reed-anturit kertovat) kunnes reed-anturi antaa
arvoa 0. Tämän jälkeen annetaan taajuusarvoksi 0 ja
    'pysäytetään ajo. Lopetaan Hairiotilanne() aliohjelma

    e.Result = Ekarelearvo
```

```
    If Ekarelearvo = 0 Then      'Kolmella seuraavalla if-lauseella määritellään mikä
häiriötilanteista sattunut. e.Result arvosta riippuu mikä häiriötilannevideo käynnistetään
Hairio_RunWorkerCompleted osiossa.
        e.Result = 1
    End If

    If Tokarelearvo = 0 Then
        e.Result = 2
    End If

    If Tokarelearvo + Ekarelearvo = 0 Then
        e.Result = 3
    End If

    If Kallistuseturajaarvo = 1 Then
        USB3101.AOut(Tamu1taajuusohjeChannelNum, AnaulosRange, Tamu1taajuusohjearvo)
        USB3101.DConfigBit(Tamu1taaksePortNum, 1, DigiulosDirection)
        USB3101.DBitOut(Tamu1taaksePortNum, 1, Digiulospaalle)
    End If

    If Kallistustakarajaarvo = 1 Then
        USB3101.AOut(Tamu1taajuusohjeChannelNum, AnaulosRange, Tamu1taajuusohjearvo)
        USB3101.DConfigBit(Tamu1eteenPortNum, 0, DigiulosDirection)
        USB3101.DBitOut(Tamu1eteenPortNum, 0, Digiulospaalle)
    End If

    If Kallistusvasenrajarvo = 1 Then
        USB3101.AOut(Tamu2taajuusohjeChannelNum, AnaulosRange, Tamu2taajuusohjearvo)
        USB3101.DConfigBit(Tamu2eteenPortNum, 2, DigiulosDirection)
        USB3101.DBitOut(Tamu2eteenPortNum, 2, Digiulospaalle)
    End If

    If Kallistusoikearajaarvo = 1 Then
        USB3101.AOut(Tamu2taajuusohjeChannelNum, AnaulosRange, Tamu2taajuusohjearvo)
        USB3101.DConfigBit(Tamu2taaksePortNum, 3, DigiulosDirection)
        USB3101.DBitOut(Tamu2taaksePortNum, 3, Digiulospaalle)
    End If

    'Moottorien pysäytys kun rajalta on päästy

    If Kallistuseturajaarvo = 0 Then
        USB3101.AOut(Tamu1taajuusohjeChannelNum, AnaulosRange, Tamu1taajuusohjearvo)
        USB3101.DConfigBit(Tamu1taaksePortNum, 1, DigiulosDirection)
        USB3101.DBitOut(Tamu1taaksePortNum, 1, Digiulospois)
    End If

    If Kallistustakarajaarvo = 0 Then
        USB3101.AOut(Tamu1taajuusohjeChannelNum, AnaulosRange, Tamu1taajuusohjearvo)
        USB3101.DConfigBit(Tamu1eteenPortNum, 0, DigiulosDirection)
        USB3101.DBitOut(Tamu1eteenPortNum, 0, Digiulospois)
    End If

    If Kallistusvasenrajarvo = 0 Then
        USB3101.AOut(Tamu2taajuusohjeChannelNum, AnaulosRange, Tamu2taajuusohjearvo)
        USB3101.DConfigBit(Tamu2eteenPortNum, 2, DigiulosDirection)
        USB3101.DBitOut(Tamu2eteenPortNum, 2, Digiulospois)
    End If

    If Kallistusoikearajaarvo = 0 Then
        USB3101.AOut(Tamu2taajuusohjeChannelNum, AnaulosRange, Tamu2taajuusohjearvo)
        USB3101.DConfigBit(Tamu2taaksePortNum, 3, DigiulosDirection)
        USB3101.DBitOut(Tamu2taaksePortNum, 3, Digiulospois)
    End If

    If Kallistuseturajaarvo + Kallistustakarajaarvo + Kallistusvasenrajarvo + Kallis-
tusoikearajaarvo = 0 And Ekarelearvo = 1 And Tokarelearvo = 1 Then
        'Kun ollaan päästy rajalta pois tämä if lause toteutuu ja Hai-
rio_RunWorkerCompletedissa voidaan lopettaa Hairio ohjelman suoritus pysäyttämällä ajastin
```

```
e.Result = 4

End If

End Sub

Private Sub Hairio_RunWorkerCompleted(ByVal sender As System.Object, _
ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs) _
Handles Hairio.RunWorkerCompleted
'Tämä suoritetaan aina kun Hairio bgworker valmistuu. moenskokerta-muuttujat estävät
jo käynnistetyn ohjevideon uudelleenkäynnistykseen

If e.Result = 1 Then
Label8.Text = "Paina rajakytkimien ohitusnappi pohjaan sahkokaapista"
If moneskokerta1 = 0 Then
AxVLCPlugin21.playlist.add("Rajat.wmv")
AxVLCPlugin21.playlist.next()
AxVLCPlugin21.playlist.play()
moneskokerta1 = moneskokerta1 + 1
End If

End If

If e.Result = 2 Then
Label8.Text = "Tarkasta että suorituspaikka on ok. palauta hätäpainike + paina
nappipaketista vihreää painiketta"
If moneskokerta2 = 0 Then
AxVLCPlugin21.playlist.add("Hatapainike.wmv")
AxVLCPlugin21.playlist.next()
AxVLCPlugin21.playlist.play()
moneskokerta2 = moneskokerta2 + 1
End If
End If

If e.Result = 3 Then
Label8.Text = "Tarkasta että suorituspaikka on ok. palauta hätäpainike + paina
nappipaketista vihreää painiketta. paina rajakytkimien ohitusnappi pohjaan sahkokaapista"
If moneskokerta3 = 0 Then
AxVLCPlugin21.playlist.add("RajatHatapainike.wmv")
AxVLCPlugin21.playlist.next()
AxVLCPlugin21.playlist.play()
moneskokerta3 = moneskokerta3 + 1
End If

End If

If e.Result = 4 Then
AxVLCPlugin21.video.toggleFullscreen()
Label8.Text = "Voit jatkaa simulointia painamalla Käynnistä-painiketta"
moneskokerta1 = 0
moneskokerta2 = 0
moneskokerta3 = 0
Hairiotilannetimer.Stop() 'Lopetetaan Hairio_DoWorkin suoritus pysäyttämällä
sitä triggava ajastin ja käynnistetään simulaattori-sovelluksen taustatarkkailu
Taustatarkkailu.Start()
AxVLCPlugin21.playlist.stop() 'Pysäytetään käynnissä oleva häiriötilanteessa
opastava video

End If

End Sub

Private Sub Button_pause_Click(ByVal sender As System.Object, ByVal e As Sys-
tem.EventArgs) Handles Button_pause.Click
AxVLCPlugin21.playlist.togglePause() 'Keskeytä-painikkeen painaminen keskeyttää
videon toiston. Digitaaliset säätimet hakevat kulloisenkin asetusarvon(kallisteluarvon) vi-
deon aikaleiman perusteella.
End Sub
```

```
Private Sub Button_stop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button_stop.Click
    'Stop-painikkeen painaminen sammuttaa videon, k skee taajarit 0 Hz ja pys ytt  
    moottorien ajon
    DigiulosDirection = MccDaq.DigitalPortDirection.DigitalOut
    Dim Digiulospaalle As MccDaq.DigitalLogicState
    Digiulospaalle = MccDaq.DigitalLogicState.High
    Dim Digiulospois As MccDaq.DigitalLogicState
    Digiulospois = MccDaq.DigitalLogicState.Low
    Tamu1eteenPortNum = MccDaq.DigitalPortType.AuxPort
    Tamu1taaksePortNum = MccDaq.DigitalPortType.AuxPort
    Tamu2eteenPortNum = MccDaq.DigitalPortType.AuxPort
    Tamu2taaksePortNum = MccDaq.DigitalPortType.AuxPort

    Dim Tamu1taajuusohjearvo As UInt16
    Tamu1taajuusohjearvo = 0
    Dim Tamu2taajuusohjearvo As UInt16
    Tamu2taajuusohjearvo = 0

    AxVLCPlugin21.playlist.stop()
    Saatimet.Stop()

    USB3101.DConfigBit(Tamu1eteenPortNum, 0, DigiulosDirection)
    USB3101.DBitOut(Tamu1eteenPortNum, 0, Digiulospois)

    USB3101.DConfigBit(Tamu1taaksePortNum, 1, DigiulosDirection)
    USB3101.DBitOut(Tamu1taaksePortNum, 1, Digiulospois)

    USB3101.DConfigBit(Tamu2eteenPortNum, 2, DigiulosDirection)
    USB3101.DBitOut(Tamu2eteenPortNum, 2, Digiulospois)

    USB3101.DConfigBit(Tamu2taaksePortNum, 3, DigiulosDirection)
    USB3101.DBitOut(Tamu2taaksePortNum, 3, Digiulospois)

    USB3101.AOut(Tamu1taajuusohjeChannelNum, AnaulosRange, Tamu1taajuusohjearvo)
    USB3101.AOut(Tamu2taajuusohjeChannelNum, AnaulosRange, Tamu2taajuusohjearvo)

    lastErrX = 0    'Ilman n it  s  timien arvojen nollauksia s  timill  olisi sis ss 
    pys ytyksen j lkeen ennen pys ytyst  olleet s  t arvot. Simuloinnin uudelleenk ynnistyksen
    j lkeen s  timet jatkaisivat s  t   siit  mihin ne viimeksi j iv t
    lastErrY = 0    'erityisesti ongelmaa aiheuttaisi mahdollinen suuri integrointiar-
    vo(varsinkin kun s  timess  ei ole anti-windup-toimintoa)
    lastTimeX = 0
    lastTimeY = 0
    errSumX = 0
    errSumY = 0
End Sub
End Class
```