

VAASAN YLIOPISTO

TEKNILLINEN TIEDEKUNTA

AUTOMAATIOTEKNIikka

Juha Tuomas Niemi

KUULAPELI

LabVIEW-toteutus

Diplomityö, joka on jätetty tarkastettavaksi diplomi-insinöörin tutkintoa varten
Vaasassa 31.12.2006.

Työn valvoja

Jarmo Alander

Työn ohjaaja

Janne Koljonen

ALKULAUSE

Olen tehnyt diplomityöni aiheesta, jonka laajuus oli niin suuri, että se oli jaettava kahden osa-alueeseen. Toisen osa-alueen tekijänä on insinööri Ismo Tupamäki. Diplomityö on tehty Seinäjoen ammattikorkeakoulun tiloissa, varastohuoneessa 244. Esitän kiitokseni diplomityön valvojana toimineelle professori Jarmo Alanderille, sekä ohjaajana toimineelle diplomi-insinööri Janne Koljoselle, jonka opastukset ja kannustukset olivat ensisijaisen tärkeitä hankkeemme onnistumisen kannalta. Lisäksi kiitän kollegaani insinööri Ismo Tupamäkeä hyvästä yhteistyöstä, Juho Seppälää pelikehikon rakentamisesta, tiedostusvastaavaa Pia-Mari Riihilahtea tiivistelmän englannin kielen tarkastuksesta, lehtori Anna-Kaisa Alataloa suomen kielen kieliopin tarkastuksesta ja KLTITE03MA-luokkaa (Honkala, Saarinen, Siirilä ja Tuuri) rakennesuunnittelusta ja muita kollegoja kannustuksesta. Lämmin kiitos Merville kannustuksesta ja tuesta.

Seinäjoella 31.12.2006

Juha Niemi

VAASAN YLIOPISTO**Teknillinen tiedekunta**

Tekijä:	Juha Niemi	
Diplomityön nimi:	Kuulapeli: LabVIEW-toteutus	
Valvojan nimi:	Jarmo Alander	
Ohjaajan nimi:	Janne Koljonen	
Tutkinto:	Diplomi-insinööri	
Laitos:	Sähkö- ja automaatiotekniikan laitos	
Koulutusohjelma:	Sähkö- ja energiatekniikan koulutusohjelma	
Suunta:	Automaatiotekniikka	
Opintojen aloitusvuosi:	2000	
Diplomityön valmistumisvuosi:	2006	Sivumäärä: 100

TIIVISTELMÄ

Työssä toteutettiin kuulapeli, jossa kuulaa liikutetaan neliön muotoisen tasopinnan kaltevuutta muuttamalla. Peli on moottoroitu ja ohjauksessa voidaan käyttää LabVIEW tietokoneohjelmaa, joka on toteutettu graafisella ohjelmointikielellä.

Pelaaja voi ohjata tasojen kaltevuutta joko manuaalisesti Joystick-ohjaimen avulla tai automaattiajolla, jolloin pelin ylle sijoitettu kamera seuraa kuulan etenemistä tasolla ja mittaa kuulan sijainnin. LabVIEW-ympäristöön rakennettu ohjelma käsittelee mittaus-tiedon ja lähettää ohjauksikäskyn PD-säätimen kautta tason kaltevuutta sääteleville askelmoottoreille kahta erillistä I2C-väylää pitkin.

Työ jaettiin laajuutensa vuoksi kahteen osaan. Tässä osassa käsitellään lähinnä pelin mekaniikkaa, turvallisuutta ja ohjausta. Ismo Tupamäen toteuttamassa osassa käsitel-lään pelin käyttöliittymää ja konenäköä.

Kuulapelillä suoritettiin useita testiajoja manuaali- ja automaattiohjauksilla. Manuaa-liohjauksella taso seurasi erinomaisesti joystick-ohjaimen liikkeitä sekä x että y-suuntien kallistuksissa. Pelaajan taidoista riippuen, kuulan liikuttelu tasolla onnistui varsin hyvin.

Automaattiohjauksella paikannus ja säätö toimivat hyvin ja kuula pysähtyi jokaisella testiajolla määriteltyn kohtaan tasoa keskimäärin 33 sekunnissa. Askelmoottorien hita-us, tasopinnan kuperuus ja kuulan pinnan epätasaisuus vaikeuttivat hiukan kuulan ohjat-tavuutta.

AVAINSANAT: LabVIEW, graafinen ohjelmointi, askelmoottori, PID-säätö, I2C, pelit

UNIVERSITY OF VAASA**Faculty of technology**

Author:	Juha Niemi
Topic of the Thesis:	Ball balancing game: LabVIEW-implementation
Supervisor:	Jarmo Alander
Instructor:	Janne Koljonen
Degree:	Master of Science in Technology
Department:	Department of Electrical Engineering and Automation
Degree Programme:	Degree Programme in Electrical and Energy Engineering
Major of Subject:	Automation
Year of Entering the University:	2000
Year of Completing the Thesis:	2006

Pages: 100

ABSTRACT

This project develops a ball balancing game, where a ball is moved through changing the inclination angle of a rectangular play arena. The game is motorized and can be controlled by a program implemented as LabVIEW graphical program modules.

The player can control the inclination angle of the play arena manually with a Joystick. An automatic drive can also be used for controlling; a camera is located above the game arena and it tracks the progress of the ball on the surface by measuring its position. A program built in the LabVIEW platform processes the measurement data and sends the command through a PD controller to the step motors that control the angle of the plane surface. The data goes through two different I2C connectors.

The project is divided into two different theses, due to the extent of the project. This thesis focuses mainly on the mechanics, safety and controls of the game. In the second thesis done by Ismo Tupamäki the focus will be on the operating system and machine vision.

Several test runs were made using both manual and automatic controls. When using the manual control, the surface followed the joystick movements in the x and y direction excellently. Depending on the player's skill, he / she was quite successful in controlling the ball movements.

With automatic control, the location and adjustment worked well and the ball stopped in the predetermined place within 33 seconds on the average. Because of the slow speed on the step motor, the platform's convex surfaces and the ball's uneven surface the ball's manoeuvrability is slightly complicated.

KEYWORDS: LabVIEW, graphical programming, stepper motor, PID-controller, I2C

SISÄLLYSLUETTELO

ALKULAUSE	2
TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYSLUETTELO	5
LYHENNE JA MUUTTUJALUETTELO	8
1. JOHDANTO	10
1.1. Työn vaiheet	11
1.2. Automaatiopelit	12
1.3. Tavoitteet	16
1.4. Kuulapelin toiminta	16
1.5. Sisältö	17
1.6. LabVIEW	18
1.6.1. LabVIEW'n taustoja	18
1.6.2. LabVIEW-ohjelmien rakenne	19
1.6.3. Graafisen ohjelmoinnin etuja	22
1.6.4. Graafisen ohjelmoinnin haittoja	23
1.7. Sääntötekniikka	23
1.7.1. Takaisinkytketty säätö	24
1.7.2. PID-säädin	25
1.7.3. Stabiilisuus ja epästabiilisuus	27
1.7.4. PID-säätimen virittäminen	28
1.7.5. PID ja sumea säätö	28
1.7.6. PID ja geneettiset algoritmit	29
1.8. I ² C – väylä	29
1.8.1. Yleistä	30
1.8.2. I ² C:n linjat	30
1.8.3. Isäntä-renki-väylä	31

2.	RAKENNE JA MEKANIikka	33
2.1.	Kuulapelin rakenne	33
2.2.	Askelmoottorit	37
2.2.1.	Yleistä	37
2.2.2.	Askelmoottorin toimintaperiaate	39
2.2.3.	Muuttuvan reluktanssin moottori	40
2.2.4.	Kestomagnetoitu askelmoottori	41
2.2.5.	Hybridimoottori	42
2.3.	Pelikuulan testaaminen	43
3.	PELIN OHJAUS	48
3.1.	Manuaaliohjaus	49
3.2.	Automaattiajo	51
3.2.1.	Sysäyskäyttö	52
3.2.2.	Perusajo	56
3.2.3.	PD-säätö	58
3.3.	Rajat	62
3.4.	Valintakytkin	64
3.5.	Askelmoottoreiden ohjaus	64
4.	TULOKSIA	67
4.1.	Nopeuden säätö	67
4.1.1.	Kuvaajan tulkitseminen	68
4.1.2.	Mittaukset ja kommentit	68
4.1.3.	PD-säädön parametriviritys	73
4.2.	Tulosten tarkastelu	77
4.3.	Kokemuksia ja ongelmia	79
5.	YHTEENVETO	82
5.1.	Tulevaisuus	84
	VIITTEET	85
	LIITTEET	90

Liite 1. Askelmoottoreiden kytkennät K8005 ohjainkorttiin.	90
Liite 2. CB-68LPR-kortti.	91
Liite 3. NI 6221 kytkentä	92
Liite 4. Askelmoottorin ohjainkortti K8005	93
Liite 5. Käsiäjon ohjauskaavio.	94
Liite 6. Automaattiajon ohjauskaavio.	95
Liitteet 7 - 11. Kuvakelohkoluettelo	96

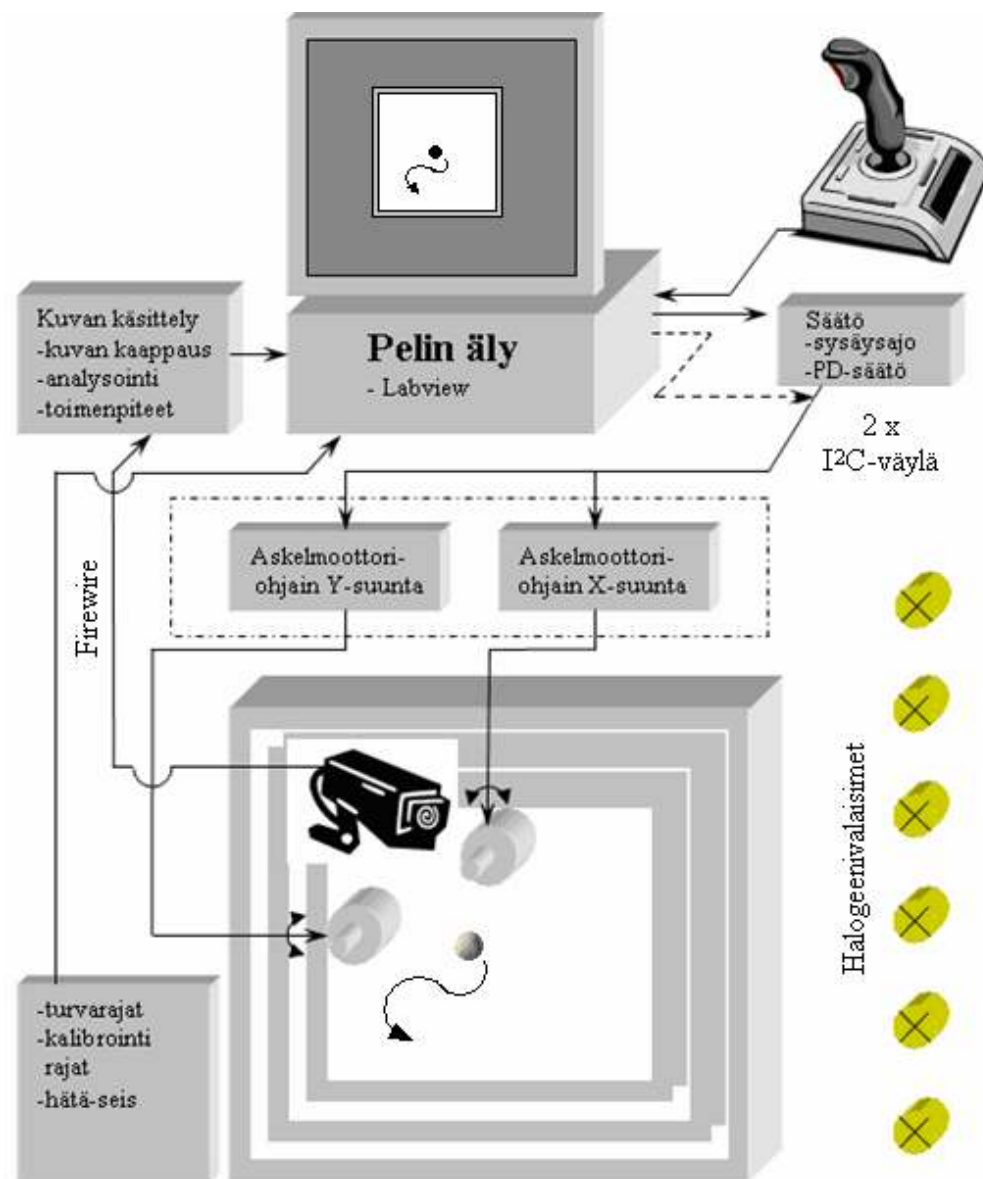
LYHENNE JA MUUTTUJALUETTELO

α	Pelitason kallistuskulma
α_s	Aseteltava sysäyskulma
α_{\max}	Pelitason kallistuksen ääriraja-arvo
ω	Kulmanopeus = v/R
$e(t)$	Paikannustiedon ja kohdekoordinaatin välinen erosuure
$e_1(t)$	Muokattu erosuurearvo PD-säädön käyttöön soveltuvaksi
E_P	Potentiaalienergia
E_L	Lineaarienergia
E_R	Rotaatioenergia
g	Maan vetovoima 9,81 m/s ²
h	Kuulan putoamismatka
J	Kuulan hitausmomentti
K_D	Derivointivahvistus
K_p	Vahvistus
L	Pelitason pituus
m	Kuulan massa
\mathbf{m}^T	Kuulan paikannustieto tasolla = (m_x, m_y)
m_x	Kuulan paikannustieto / x-akseli
m_y	Kuulan paikannustieto / y-akseli
m_{xv}	Kuulan paikannustieto viiveellä / x-akseli
m_{yv}	Kuulan paikannustieto viiveellä / y-akseli
R	Kuulan säde
\mathbf{r}^T	Kohdekoordinaatit = (r_x, r_y)
r_x	Kohdekoordinaatti / x-akseli
r_y	Kohdekoordinaatti / y-akseli

u_p	Liukupotentiometrin arvo, mikä ilmoittaa tasokaltevuuden
$u(t)$	Lähtösignaaliarvo PD-säädöltä = $u_1(t) + u_2(t)$
$u_1(t)$	Lähtösignaaliarvo P-säädöltä = $K_p e_1(t)$
$u_2(t)$	Lähtösignaaliarvo D-säädöltä = $K_D \frac{de_1(t)}{dt}$
$u_3(t)$	Lähtösignaaliarvo, joka on lisätty tasapainotila-arvoon = $2,5 + u(t)$
$u_4(t)$	Lähtösignaaliarvo, joka on vähennetty tasapainotila-arvosta = $2,5 - u(t)$
\mathbf{v}_p^T	Aseteltava perusnopeus = (v_{px}, v_{py})
v_{px}	Aseteltava perusnopeus / x-akseli
v_{py}	Aseteltava perusnopeus / y-akseli
v_s	Aseteltava sysäysnopeus / kallistus
v_{sp}	Aseteltava sysäysnopeus / sysäyksen palautus

1. JOHDANTO

Kuulapeli, jossa kuula ohjataan määrättyä reittiä pitkin erilaisten esteiden ja reikien ohitse manuaalisesti, on ollut suosikkipeli kaikenikäisille jo useiden vuosikymmenten ajan. Pelaaminen vaatii melkoista näppäryyttä ja kokemusta. Tässä diplomityössä esiteltävä peli on kehitelty alkuperäisestä mallista. Automaatio (äly) on yhdistetty peliin kuvan 1 esittämällä tavalla tuomaan lisää haasteita ja mielenkiintoa pelaajalle.



Kuva 1. Kuulapelin rakenne.

1.1. Työn vaiheet

Idea kuulapelin automatisoimiseksi syntyi suunnittelupalaverissa Tupamäen Ismon kanssa. Mietimme mahdollista yhdessä toteutettavaa diplomityöaihetta, missä molemmilla olisi omat tutkimuskohteensa. Keskustelimme erilaisista vaihtoehdoista, mutta mikään ei tuntunut tarpeeksi haasteelliselta ja mielenkiintoiselta, kunnes keksimme kuulapelin. Olimme molemmat omistaneet kyseisen pelin joskus nuoruudessamme, joten sen toiminta oli hyvinkin tuttua. Ideoimme erilaisia asioita, mitä kaikkea voisimme työhömmä liittää automaation saralta. Esille tulivat ohjaus- ja säätömenetelmät, mekaniikka, kuulan seurantajärjestelmä, käyttöliittymä yms. eli todella paljon selvitettäviä asioita. Lisäksi kuulapeli tulisi rakentaa mahdollisimman halvalla, mikä toisi oman haasteensa automatisointiin.

Ohjaajamme Janne Koljosen avustamana teimme työnjaon siitä, mitä osioita kumpikin osapuoli käsittelee opinnäytetyössään. Tässä osassa käsitellään lähinnä pelin mekaniikkaan, turvallisuuteen ja perussäätöihin liittyviä asioita. Ismo Tupamäen opinnäytteessä käsitellään pelin käyttöliittymää ja kuvankäsittelyä.

Kuulapelin suunnittelu aloitettiin kesällä 2005 pelin mekaniikan suunnittelulla. Syksyllä valittiin pelille käyttöympäristö, graafinen ohjelmointiympäristö LabVIEW. Ohjelmaan tutustuminen ja sen käytön opetteleminen työn ohessa vei lähes koko alkusyksyn. Lopputyksystä pelissä oli jo hieman ohjaukseen ja kuulan seurantaan liittyvää toimintaa. Alkukevästä 2006 työ eteni hyvin hitaasti, johtuen työkiireistä, mutta loppukevällä pelin toiminta parani koko ajan. Ongelmia oli runsaasti: yhden selvittyä tilalle ilmestyi aina uusi. Ensimmäinen takaraja itse opinnäytetyön valmistumiselle oli kesäkuu 2006. Kesällä 2006 työ eteni hyvin, säätö ja kuulan seuranta toimivat jo välttävästi ja paraniivat koko ajan. Ainoastaan kameran kuvausnopeuden ja säädön nopeuden kanssa oli ongelmia. Jos nopeutta lisäsi, prosessoritehot nousivat 100 %:iin ja järjestelmä kaatui. Ongelmat selvisivät kohta kohdalta, joten peli oli toiminta- ja testauskuntoinen 09.10.2006.

1.2. Automaatiopelit

Flipperi, ehkä kaikista tunnetuin kuulapeli, sähköistyi 1930-luvulla. Tämä mahdollisti automaattisen pistelaskusysteemin. Aluksi flipperissä kuula vain siirrettiin peliin ja toivottiin mahdollisimman hyviä pisteitä. Pelattavaksi flipperi tuli vuonna 1947, jolloin pelilaudan alareunaan asennettiin mailat, joilla kuulaa yritettiin pitää kentällä mahdollisimman kauan (Vartiainen 1992).

Automaattipelien alkutaipaleelta lähtien erilaiset testerit olivat jo erittäin suosittuja. Eri-tyisessä suosiossa olivat voimatesterit, ja shokkerit, jotka antoivat käyttäjälleen sähköiskuja. Niiden väitettiin parantavan kaiken mahdollisen reumatismista päänsärkyyn. Sittemmin tulivat reaktiotesterit, mistä hyvä esimerkki oli Speden Speleissä käytetty speedmaster-tyyppinen peli (Vartiainen 1992).

Ensimmäiset levyautomaatit olivat pelkkiä kolikoilla toimivia gramofoneja. Ne soittivat ainoastaan yhtä kappaletta. Sähköllä vahvistettua ääntäkin saatiin odottaa aina 1920-luvun lopulle. Moderni jukebox syntyi 1930-luvulla, jolloin asiakkaat pääsivät itse valitsemaan kappaleet. Suurin harppaus tapahtui 50-luvun alussa, kun vanhat savikiekkolevyt korvattiin (nopeuden 45 rpm.) vinyylilevyillä (Vartiainen 1992).

Laitteita, joihin liittyy kuulan tai pallon tasapainotus, on kehitelty monenlaisia niin malliltaan kuin ohjaukseltaankin. Muun muassa Ka Cheok ja Nan Loh (1987) julkaisussa esitellään laite, joka koostuu metallisesta kuulasta, joka vierii kahden vierekkäin olevan kaarikiskon päällä. Kaarikiskot on kiinnitetty omalla radalla kulkevaan, servomoottorilla liikuteltavaan vaunuun. Vaunun radan kaltevuutta voidaan muuttaa halutun suuriseksi (ulkoisen häiriö). Servomoottorin akseliin on kytketty takogeneraattori osoittamaan servomoottorin pyörimisnopeutta. Kuulan ja vaunun asematieto saadaan liukupotenttiometreiltä.

Laite tasapainottaa metallikuulan kaarien lakipisteeseen ja paikantaa vaunun keskelle rataprofiilia. Ohjausjärjestelmää testattiin kahdella tavalla: ilman ulkoisia häiriöitä, jolloin vaunu liikkui kiskoilla, joiden kaltevuus oli 0° ja ulkoisen häiriön kanssa, jolloin

kiskojen kaltevuus oli $> 0^\circ$. Kyseisen järjestelmän säätöön kehitelty, potentiometreiltä tulevia paikkatietoja ja moottorin takometriltä tulevaa nopeustietoa käsittelevä liikeyhtälö, on annettu matriisimuodossa. Näiden tietojen perusteella säätö laskee oikeanlaiset ohjausparametrit (nopeus, suunta) ja lähettää ne vaunua ohjaavalle servomoottorille (Ka Cheok ja Nan Loh 1987).

E. Laukonen ja S. Yurkovich (1993) suunnittelivat pallon tasapainotuslaitteen, jossa PID-säädin ohjaa DC-moottoria, joka kallistelee keskiosastaan laakeroitua palkkia molempiin suuntiin 50:1 välityssuhteen kautta. Laitteen tarkoitus on paikantaa pallo palkin keskikohtaan. Järjestelmässä on 32 valotransistoria (5V jännitesyötöllä) asennettuna palkin runkoon ja kaksi valonlähdettä, jotka valaisevat palkkia ylhäältä. Potentiometri, joka on kytketty palkin varteen, mittaa palkin kallistuskulmaa. Valotransistorit lähettävät pallon paikannustietoa ja potentiometri lähettää palkin kallistuskulmatietoa PID-säätimelle. PID-parametrit on valittu sellaiseksi, että DC-moottori toimii mahdollisimman nopeasti, eivätkä säätöarvot ylity.

Nykyään tietokoneet ja konenäkö ovat tulleet automaattipeleihin vahvasti mukaan. Esimerkiksi Kostamo, Hyötyniemi & Kuosmanen (2005) toteuttivat pelin, missä palloa pidetään kuperan levyn päällä, joko tietokoneohjatusti tai käyttäjäohjatusti. Tietokoneella ohjatussa ajossa konenäkö paikantaa pallon ja laskee levyn kääntökulman. Käyttäjäohjatussa ajossa pelaaja voi vertailla omia taitojaan tietokonetta vastaan, ohjaten itse levyä ja yrittäen saada pallon levyn keskipisteeseen nopeammin kuin tietokone ja yrittää pitää se siellä.

Levyn kallistelu on hoidettu kiinnittämällä levy ristinivelellä keskipisteestään laitteen runkoon ja kiinnittämällä levyn reunoille kaksi lineaarimoottoria 90 asteen kulmassa toisiinsa nähden. Tällöin ajamalla lineaarimoottorien karoja ylös tai alas saadaan levy kallistettua haluttuun kulmaan (Kostamo 2005).

Pallon paikantamiseen käytetään konenäköä, jossa pallo kuvataan 30 kertaa sekunnissa videokameralla. Paikkatietoa ei kuitenkaan aivan suoraan käytetä hyväksi säädössä, vaan pallon paikan ja nopeuden laskemiseksi käytetään Kalman-suodatinta. Kalman-

suodattimen käytössä on se hyvä puoli, että se ottaa huomioon myös edelliset mittaukset, jolloin saadaan tarkempi arvio pallon paikasta ja nopeudesta (Kostamo 2005).

Kameralla otettu kuva viedään tietokoneelle. Tietokoneeseen ohjelmoitu säätäjä laskee halutun alustalevyn kulman pallon paikan ja nopeuden perusteella. Toisin sanoen pallon paikkaa ja nopeutta kerrotaan eri vakioilla, jotka on laskettu säätäjää suunniteltaessa, ja kun nämä tulot lasketaan yhteen, saadaan haluttu levyn kulma. Haluttu levyn kulma syötetään servo-ohjaimelle ohjearvoksi. Servo-ohjaimen tehtävä on ohjata lineaarimootoreille annettavaa virtaa ja säätää näin moottorien karojen paikkaa (Kostamo 2005).

Koska lineaarimootorit saavat aikaan lineaariliikkeen ilman mekaanista välitystä, ovat ne käytännössä lähes kulumattomia ja välyksettömiä. Lisäksi moottoreilla on erittäin hyvät dynaamiset ominaisuudet ja niillä päästää jopa 10 g:n kiihtyvyyksiin (Kostamo 2005).

Dieter Zöpel (1998) toteutti laitteen, jonka toimintaperiaate on pitää kuula keskellä tasoa samalla tavalla, kuin esimerkiksi tarjoilija tasapainottaa tarjotinta. Laite koostuu alumiinirungosta, tasosta, videokamerasta ja kahdesta muuttuvan reluktanssin askelmoottorista. Tason kallistelua ohjataan kahdella ortogonaalisella akselilla, joita askelmoottorit ohjaavat hihnojen välityksellä. Taso kallistuu 15° molempiin suuntiin. Videokamera on sijoitettu metrin korkeudelle tason yläpuolelle, josta paikannustieto välittyy 40 ms välein ohjaukselle.

Säätö perustuu anturi- ja paikannustietojen muutoksiin reaaliajassa. Näiden muutosten perusteella säädin laskee oikeanlaiset kallistuskulmat tasolle. Ohjaus on toteutettu neljällä tietokoneella, joista ensimmäinen selvittää pallon sijainnin. Toinen tietokone laskee tason kallistuskulmat α ja β . Tietokoneet kolme ja neljä kallistavat tasot todellisiin kulmiin α ja β . Tietokoneiden välinen tiedonsiirto on toteutettu Can-väylällä (Zöpel 1998).

Manuaalijossa joystick-ohjaimelta lähtevä ohjaustieto välittyy tietokoneelle kaksi, joka laskee tason kallistuskulmat. Kallistustiedot välittyvät tietokoneille kolme ja neljä, jotka ohjaavat tasoja kallistelevia askelmoottoreita (Zöpel 1998).

Dadios, Baylon, De Guzman, Florentino, Lee & Zulueta (2000) tutkivat konenäön ja sumean logiikan käyttöä pallon tasapainotuksessa. Testauslaite koostuu alumiinisesta palkista, jonka pituus on 600 mm, leveys 25 mm ja korkeus 11 mm. Palkki on kiinnitetty keskiosastaan kuulalaakerin välityksellä DC-moottoriin, joka pystyy kallistamaan palkkia 20° molempiin suuntiin. Potentiometrillä, joka on kytketty palkin varteen, mitataan palkin kallistuskulmaa. Paineantureilla, jotka on sijoitettu palkkiin, mitataan pallon sijaintia. Palkkien päihin on asennettu puskurit, estämään pallon putoamisen. Konenäköjärjestelmän kamera on sijoitettu n. 90 cm korkeudelle palkin yläpuolelle.

Konenäköjärjestelmä määrittelee pallon paikan ja nopeuden. Konenäköjärjestelmä määrittelee tarkkaan pienimmänkin mahdollisen välimatkan, jonka pallo voi edetä palkilla. Vaikka pallo ei olisi yhteydessä palkkiin paineanturien välityksellä, niin konenäköjärjestelmä tietää silti pallon paikan, antaen johdonmukaista tietoa järjestelmän hallitsemiseksi. Tasapainotukseen voidaan käyttää sekä marmori että golf-palloja. Säädössä käytetään neljää muuttujaa: kulmanopeus, pallon paikkatieto, pallon nopeus ja palkin kallistuskulma. Kyseiset muuttujatiedot välittyvät sumealle logiikalle, joka ohjaa kulmia säätelevää DC-moottoria. Testiajoissa seurattiin kahta eri parametria, nimittäin poikkeamaa ja asetusaikaa. Sumea logiikka tasapainottaa pallon keskimäärin 4 mm toleranssilla noin 5,6 sekunnissa (Dadios 2000).

Vaasan yliopistossa Koljonen (2004) toteutti pelin nimeltään Virtual Keeper, missä animoitu maalivahti pyrkii torjumaan pelaajan heittämän pallon. Maalin kuva ja maalivahti animoidaan videotykillä valkokankaalle. Tässä pelissä konenäön avulla lasketaan maalia kohden heitetyn pallon lentorata. Kyseisen tiedon avulla animoitu maalivahti päättää, pallon ja maalin leikkauspisteen perusteella, minkälaisen torjunnan se esittää, eli mihin kohtaan maalia torjunta kohdistuu. Lisäksi simulaattori ilmoittaa heitetyn pallon lentonopeuden, sekä tilastoja parhaista nopeuksista.

1.3. Tavoitteet

Asetimme tavoitteet, jälkikäteen arvioiden, vähän turhankin korkealle eli nimesimme pelillemme viisi tasoporrasta, joista toteuttaisimme vähintään kolme ensimmäistä tasoa. Tasot olivat seuraavat:

1. Mahdollistaa pallon kuljettamisen alkupisteestä kohdepisteeseen Joystick-ohjaimen avulla.
2. Edellisen toiminnan lisäksi konenäkö valvoo suoritusta, estäen mm. seinämiin törmäilyt.
3. Tietokone ohjaa konenäön opastamana pallon kohdepisteeseen ohjelmoitua reittiä pitkin.
4. Tietokone ohjaa pallon maaliin konenäön avulla etsien itse reitin.
5. Tietokone etsii nopeimman mahdollisen (optimaalisen) reitin lähdöstä maaliin.

Työn edetessä jouduimme hiukan tinkimään tavoitteistamme, toteuttaen tasot 1 ja 3. Loput kolme tasoa jätimme mahdolliseen tulevaisuudessa tapahtuvaan jatkokehittelyyn.

1.4. Kuulapelin toiminta

Pelin ohjain on toteutettu LabVIEW-nimisellä, graafisella ohjelmointiympäristöllä, mihin kaikki pelin vaatimat toiminnat on koottu yksittäisistä ohjelmamoduuleista eli virtuaali-instrumenteista. Pelissä on kaksi toiminnallista tasoporrasta, käsi- ja automaattiajo.

Käsiäjossa pelikuula on tarkoitus kuljettaa Joystick-ohjaimen avulla haluttuun kohtaan tasoa, väistellen mahdollisia esteitä. Mekaaniset äärirajat estävät tasojen liiallisen kallistumisen. Tasojen kaltevuutta seuraavat liukupotentiometrit ohjaavat tasot takaisin vaakatasoon aina, kun Joystick-ohjaimesta päästetään irti. Joystick-ohjaimen herkkyyttä on mahdollista säätää käyttöliittymässä sijaitsevalla virtuaalipotentiometrillä.

Automaattiajossa annetaan käyttöliittymässä sijaitsevilla säätimillä x- ja y-koordinaattiarvot eli paikka, mihin halutaan pelikuulan tasolla kulkeutuvan. Kamera

seuraa pallon kulkua, lähettäen tietyin väliajoin x-y-koordinaattia pallon sijainnista ns. takaisinkytkentätietona säätimeen. Eroelin vertaa takaisinkytkentätietoa asetusarvoon ja lähettää erosuureen kallistuskulmia ohjaavalle säädölle. Kyseinen säätö vaikuttaa tasojen kallistuskulmia ohjaavien askelmoottojen suuntaan ja kallistuskulmiin. Askelmoottorien suuntatietoon vaikuttavat myös erosuureen napaisuus, sekä x- ja y-akseleille sijoitettujen liukupotentiomietrien ääriaraja-asetukset.

Automaattiajooon on myös mahdollista määritellä jokin erillinen rataprofiili, jota pitkin pelikuulan halutaan kulkevan. Kamera seuraa tässäkin tilanteessa teräskuulaa ja ohjaa sen edellä mainittujen menetelmien avulla määriteltyä rataprofiilia pitkin.

1.5. Sisältö

Ensimmäinen luku sisältää edellisten lisäksi perusasioita LabVIEW-ohjelmointiympäristöstä, mm. taustoista ja ohjelmien rakenteesta. LabVIEW-ohjelmointiympäristö on esitelty tarkemmin Ismo Tupamäen diplomityössä. Lisäksi ensimmäisessä luvussa käsitellään, mitä säädöllä tarkoitetaan ja tutustutaan lähemmin PID-säätimeen ja sen viritämiseen, sekä esitellään I2C-sarjaväylä. Luvussa kaksi esitetään kuulapelin prototyyppi ja muutama rakenneidea mahdolliseen jatkokehittelyyn. Lisäksi tutustutaan erilaisiin askelmoottorityyppeihin ja niiden toimintaan, sekä etsitään kuulapelille paras mahdollinen pelikuula erilaisten testien avulla. Luvussa kolme esitetään LabVIEW-sovelluksella toteutetun ohjelman rakenne, koskien kuulapelin manuaali- ja automaattiajooa.

Kuulapeli vaatii toimiakseen muutakin kuin hyvin suunnitellun mekaniikan ja toimivan säätöjärjestelmän, nimittäin tarkat säätöparametrit, oikeanlaisen pelivälineen ja tarkan paikannustiedon. Tämän vuoksi kuulapelillä on suoritettava erilaisia testiajooja, jotta nämä edellä luetellut asiat saataisiin kohdalleen. Luvussa neljä haetaan säätimelle parhaat mahdolliset säätöparametrit kokeellisesti, suoritetaan tuloksien tarkastelu ja kerrotaan kuulapelin suunnittelun ja rakentamisen tuomista kokemuksista. Kappaleessa viisi tehdään yhteenveto toteutuksesta ja suunnitelma, kuinka kuulapeliä voisi jatkokehittää.

1.6. LabVIEW

LabVIEW on monipuolinen teollisuusautomaation, elektroniikan testauksen ja suunnittelun ohjelmisto. Sen erinomainen graafisen suunnittelun kehitysympäristö on uudistanut automaation ja mittausten suunnittelun. LabVIEW tarjoaa tehokkaan tavan mitata, analysoida ja esittää mittaustietoja tuotekehityksen kaikissa vaiheissa prototyypin suunnittelusta tuotantotestaukseen. Se on avoin alusta, joka on yhteensopiva satojen kameroiden, moottorikäyttöjen, TEDS-yhteensopivien anturien (älykkäät anturit) ja toimilaitteiden, OPC-käyttöisten teollisuuslaitteiden sekä yli 4 000 GPIB-, VXI-, PXI-, PCI-, sarja-, Ethernet- ja USB-liitäntäisten mittaustaitteiden kanssa. LabVIEW-ohjelmistoa on saatavana Windows-, Macintosh- ja Linux-käyttöjärjestelmiin (National Instruments 2006).

Kuulapelin älynä toimii LabVIEW, graafinen ohjelmointiympäristö. Toisin sanoen kuulapelin kaikki ohjaus- ja säätötoiminnot on ohjelmoitu LabVIEW-ohjelmiston erilaisista ohjelmamoduuleista ja yksittäisistä komponenteista. Tässä luvussa tutustutaan edellä mainitun ohjelmointiympäristön perusasioihin.

1.6.1. LabVIEW'n taustoja

LabVIEW on graafinen ohjelmointiympäristö, jonka National Instruments kehitti noin kaksikymmentä vuotta sitten Macintosh-ympäristöön hyödyntääkseen tuon ajan tietokoneita mittaustietojen käsittelyssä. National Instruments on yksi maailman suurimmista PC-mittauskorttien valmistajista. LabVIEW-ohjelmasta on vuosien myötä kehittynyt valmistajakohtainen graafinen ohjelmointikieli, joka toimii nykyään myös muissa käyttöjärjestelmissä. LabVIEW-ohjelman alkuperäinen käyttötarkoitus on lähinnä mittaustietojen keruu ja analysointi. Tämän vuoksi välineen käyttöliittymätoiminnoissa ovat hyvin edustettuina erilaiset painonapit, vierityspalkit ja piirturit. ASCII-merkkien syöttöönkin on omat välineensä.

Ohjelma voi tallettaa tietoa merkkimuodossa, jolloin se on helposti siirrettävissä Windows-välineillä esimerkiksi Exceliin jatkokäsittelyä varten. Muut talletusmuodot ovat numeerinen sellaisenaan ja numeerinen pakattuna. Nopeissa ja laajoissa mittaussovel-

luksissa tietoa voi nopeasti kerääntyä hyvinkin paljon, jopa useita kymmeniä megatavuja. Tämän vuoksi tiedon pakkaus on olennaisen tärkeää. Hyvät tietokantalinkit välineestä vielä puuttuvat (National Instruments 2006).

LabVIEW-ohjelmalla tehdään esimerkiksi kuva-analyysejä. Graftek on yksi tällainen, jolla on sovelluksia mm. liekin analysointiin kolmiulotteisesti. Samoin on Helsingin teknillisessä korkeakoulussa tehty keinohorisontilla toteutettu lentosimulaattori. Ohjelman sponsorina on ollut Ilmavoimat, jolla on puolestaan ollut antaa malliksi oikea USA:sta ostettu lentosimulaattori (Narikka 1996: 5).

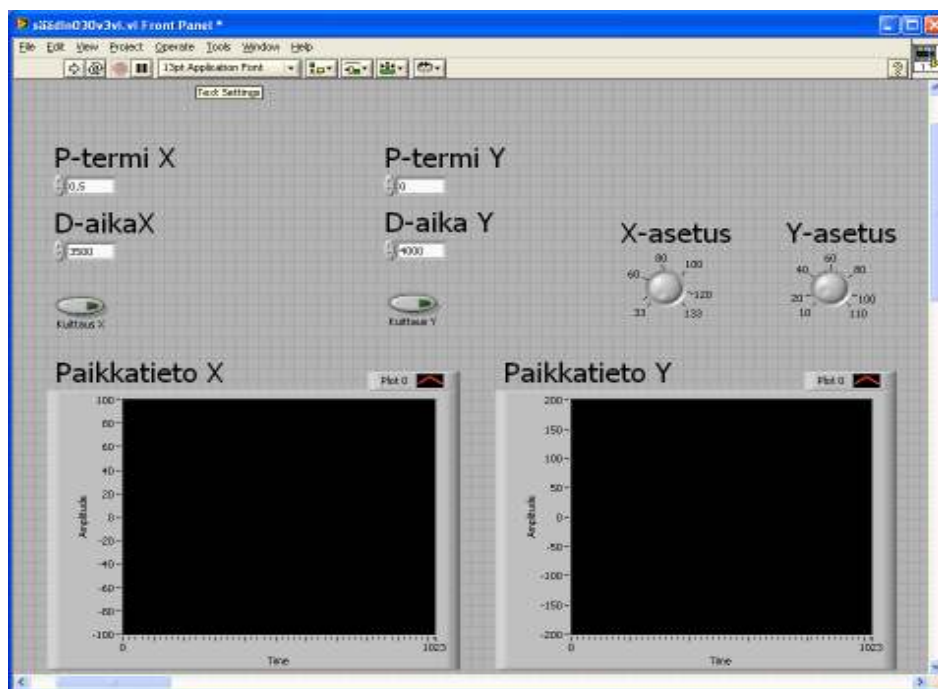
1.6.2. LabVIEW-ohjelmien rakenne

LabVIEW-ohjelma perustuu niin sanottujen ohjelmamoduulien eli virtuaali-instrumenttien (VI) käyttöön. Virtuaali-instrumenttien ulkonäkö ja toiminta on rakennettu mukailemaan mahdollisimman tarkasti fyysisiä eli olemassa olevia laitteita ja komponentteja. Yksittäisellä instrumentilla on vain rajallinen määrä ominaisuuksia, kuten esimerkiksi ulkoisen analogiasignaalin mittaus tai mitatun signaalin graafinen esittäminen. Tästä johtuen varsinaiset LabVIEW-sovellukset ovat useista eri toimintoja simuloivista virtuaali-instrumenteista koottuja kokonaisuuksia, joista jokaisella on oma tarkoituksensa (National Instruments 2006).

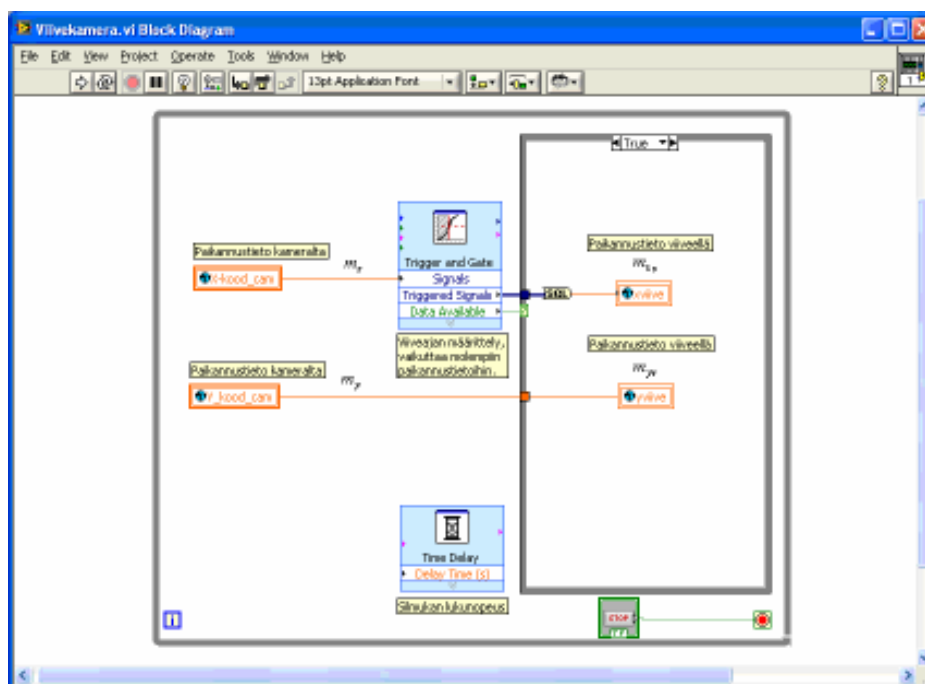
Riippuen valituista instrumenteista käyttäjä pystyy myös itse muuttamaan mitattavan prosessin kulkua muuttamalla prosessin parametreja. Muutosten vaikutus on helppo nähdä välittömästi PC:n näytöltä. Juuri tämä ominaisuus tekee LabVIEW-ohjelmasta ihanteellisen tutkimus-, testaus- ja opetuskäyttöön. Olkoon ohjelma sitten liitetty johonkin fyysisesti olemassa olevaan prosessiin, kuten laboratoriotyökokonaisuuteen, tai tehtäessä pelkkä ohjelmapohjainen simulaatio, LabVIEW on turvallinen vaihtoehto prosessin säädön tutkimiseen tai harjoitteluun. Ohjelman tietokanta sisältää useita valmiita VIKokonaisuuksia, mutta käyttäjä voi myös muokata olemassa olevia instrumentteja itse tai luoda kokonaan uusia (National Instruments 2006).

LabVIEW-sovelluksessa yksittäinen ohjelmamoduuli on siis nimeltään virtuaali-instrumentti eli VI. Käytännössä ohjelmamoduulien nimen laajenteena on `vi` eli ohjelma voi olla nimeltään vaikka `PID.vi`. On huomattava, että nimi ei välttämättä ole tässä tapauksessa DOS-tyyppinen tiedostonimi, vaan DOS-tiedostona on kirjasto (lib), jossa on VI:ja. Tämä johtuu siitä, että yhteensopivuuksien (Mac, Unix, DOS) vuoksi LabVIEW-ohjelmassa on sallittu jo alusta lähtien yli kahdeksanmerkkiset nimet (National Instruments 2006).

Jos tarkastellaan VI:n rakennetta, niin huomaamme sen muodostuvan kahdesta eri osasta: Front Panel, missä sijaitsee etupaneeli eli käyttöliittymä (kuva 2), ja Block Diagram, missä sijaitsee varsinainen ohjelma eli ohjelman looginen toiminta (kuva 3) (Narikka 1996: 4).



Kuva 2. Front Panel-ikkunassa sijaitsee etupaneeli eli käyttöliittymä. Ohjelmasta näkyvä etupaneeli muistuttaa fyysistä laitetta, koska se muodostuu napeista, mittareista, näyttöistä, säätimistä, sekä muista hallintaan liittyvistä toiminnoista. Näyttöön voidaan generoida data hiirellä tai näppäimistöllä käsin ja muutokset ovat heti nähtävillä näytöllä. Tämä osa ohjelmasta on käyttäjän nähtävissä ja käytettävissä (Narikka 1996: 4).



Kuva 3. Block Diagram-ikkunassa, sijaitsee varsinainen ohjelma eli ohjelman looginen toiminta. Ohjelma on toteutettu graafisesti ja se muodostuu kuvakeloista. LabVIEW ohjelma voi muodostua itse pääohjelmasta sekä sen sisällä olevista pienemmistä aliohjelmissä (Narikka 1996: 4). Kuvan kytkennässä kameralta tulevista paikannustiedoista (m_x, m_y) otetaan näytteitä (m_{x_v}, m_{y_v}) yhden sekunnin välein.

VI:n pääohjelma sijaitsee lohkokaaviossa. Ohjelma on toteutettu graafisesti ja se muodostuu kuvakeloista. LabVIEW ohjelma voi muodostua itse pääohjelmasta sekä sen sisällä olevista pienemmistä aliohjelmissä. Alemman tason ohjelmia eli aliohjelmia kutsutaan nimellä (SubVI) ja niitä voidaan käyttää itsenäisesti, kuten myös ylemmän tason pääohjelmia. Itse pääohjelma kannattaa jakaa näihin pienempiin aliohjelmiin, koska silloin kokonaisuus pysyy selkeämpänä ja toimintaa on helpompi ymmärtää (Kauhanen 1995: 9).

LabVIEW'n hyvä puoli on juuri se, että toimivia tuloksia saadaan aikaan hyvinkin yksinkertaisella kuvakelohkokytkennällä ja kytkentöjen toimivuus on suhteellisen helppo testata. Yleisesti ohjelman suunnittelu kannattaa aloittaa etupaneelin (Front panel) puolelta siten, että haetaan tarvittavat ohjaukseen ja valvontaan liittyvät kuvakelohkot Cont-

rol-valikosta ja asetetaan ne etupaneeliin. Etupaneelissa ei haettujen kuvakelohkojen yhdistäminen onnistu, joten siirrytään seuraavaksi lohkokaaavion (Block Diagram) puolelle. Etupaneeliin haetut kuvakelohkot näkyvät lohkokaaavion puolella kytkennällisessä muodossa. Valitsemalla lohkokaaavion Functions-valikosta ohjauksen apuvälineitä ja kytkemällä (langoittamalla) ne halutulla tavalla etupaneeliin haettujen kuvakelohkojen kanssa, saadaan aikaiseksi halutunlaisia toimintoja. (Lohilahti 2002).

Liitteessä 7 on esitelty kuvakelohkoluettelo, josta näkyy kaikki tässä opinnäytetyössä esitetyt kuvakelohkot ja niiden selitykset.

1.6.3. Graafisen ohjelmoinnin etuja

LabVIEW-ratkaisun periaatteiden selittäminen jollekin asiasta kiinnostuneelle, on syntaksista johtuen yleensä varsin vaivatonta (Narikka 1996).

Jos LabVIEW-sovelluksen lohkokaaviossa esitetty kytkentä kirjoitettaisiin auki sanalliseen muotoon, esimerkiksi Java-kielellä, niin ohjelmaa tulisi todella paljon (Nadin 2004).

Kysymyksessä on tietovuo-ohjelmointi, joten ohjelman sisäisiä apumuuttujia ei tarvita, jolloin niiden kanssa ei voi tulla ongelmiakaan. Lisäksi tyyppitarkastukset tulevat automaattisesti tehdyksi siinä vaiheessa, kun tietopisteet kytketään toisiinsa (Nadin 2004).

Vektori- ja klusterimuotoiselle tiedolla on graafisessa ohjelmoinnissa kehittyneempiä käsittelytapoja kuin monissa ns. peruskielissä. Kielestä löytyvät perusstruktuurit peräkkäisyydelle ja iteraatiosilmukoille, joten ohjelman voi halutessaan tehdä myös perinteistä ohjelmaa vastaavaksi. Oletusarvoisesti ohjelmointi on kuitenkin tapahtumaohjattua (Nadin 2004).

Voi asioida I/O-korttien kanssa, edellyttäen sopivaa ajuria. Siihen puoleen National Instruments on eräänä alan (=kortti-alan) suurista valmistajista paneutunut todella huolel-

la, sillä ilmaisia ajureita löytyy sekä omiin että kilpailijoiden kortteihin runsaasti (Narikka 1996).

Ohjelmaan voi tulla tietoa ja tapahtumia satunnaisesti monesta eri kohteesta, ja ohjelma selviytyy siitä kuten ohjelmoitava logiikka tai automaatiojärjestelmä eli tapahtumaohjatusti (Narikka 1996).

Analogiasignaaleja voidaan käsitellä jatkuvina signaaleina, joten abstraktiotaso on selvästi korkeampi kuin perinteisillä kielillä (Narikka 1996).

1.6.4. Graafisen ohjelmoinnin haittoja

Ohjelma on nopeuden ja muistitilan suhteen tehottomampaa kuin esimerkiksi Java. Toisaalta, jos tarvitaan todella nopea mittaus, niin se ei ole kiinni sovelluksen nopeudesta. Silloin pitää olla kortti, jossa on prosessorit itsessään ja joka osaa tallettaa tietoja sekä toimittaa niitä eteenpäin silloin kun (LabVIEW:lle) sopii (Nadin 2004).

LabVIEW saa asiat näyttämään siltä, että toiminta on reaaliajassa, johtuen käyttöjärjestelmäympäristöistä ja mahdollisesta tietojen tallennuksesta levyille. Jos sovellus on pieni eikä talleta levyille, niin silloin lähes ajantasainen toiminta on mahdollisuuksien rajoissa. Yleensä se riittää useille teollisuussovelluksille. Varsinaiseen ajantasaiseen toimintaan tarvitaan oikea automaatiojärjestelmä, ohjelmoitava logiikka yms. tai täytyy hankkia erillisellä prosessorilla varustettu kortti, jolle LabVIEW toimii vain isäntänä (Narikka 1996).

1.7. Sääteknikka

Käytännön säätehtävissä takaisinkytketyllä säädöllä pyritään pitämään jokin kemiallisen, fysikaalisen tai muun vastaavan järjestelmän suure tai suureet (esimerkiksi pitoisuus, pyörimisnopeus, teho, lämpötila, pinnankorkeus, paine yms.) vakiona tai muuttamaan niiden arvoa halutun suuruiseksi erilaisista järjestelmässä vaikuttavista häiriöistä riippumatta. Säädön onnistuminen riippuu säädettävien suureiden ja toimitusnopeiden lukumäärästä, niiden välisistä vuorovaikutuksista säädettävässä järjestelmässä, häiriöi-

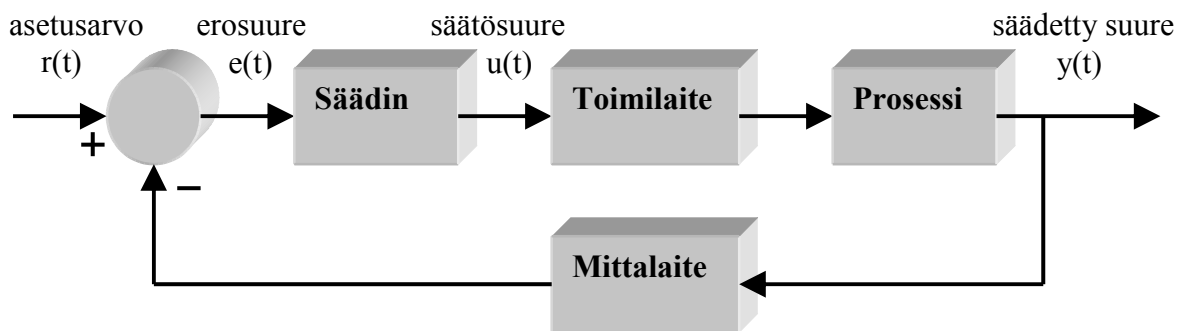
den laadusta ja ajallisesta käyttäytymisestä sekä tietysti valitusta säädintyyppistä ja parametreista (Nihtilä, Oksanen & Eriksson 2002: 2).

Perus-säätimessä eroelin vertaa takaisinkytkentätietoa ja asetusrvoa keskenään, lähettäen erosuureen suuruisen ohjaukskäskyn toimilaitteelle. Näin toimii myös kuulapeliin rakennettu säädin, se vertaa firewall-kameralta tulevaa kuulanpaikannustietoa asetusrvoon ja lähettää erosuureen suuruisen ohjaukskäskyn tasojen kaltevuutta ohjaaville askelmootoreille.

Tässä luvussa kerrotaan, mitä säädöllä tarkoitetaan, ja tutustutaan tarkemmin PID-säätimeen ja sen virittämiseen.

1.7.1. Takaisinkytketty säätö

Kuvassa 4 on esitetty takaisinkytketyn säätöjärjestelmän periaatteellinen rakenne. Asetusrvo $r(t)$ on säätöjärjestelmän ulkopuolelta tuleva ohje, jota säädetyn suureen $y(t)$ halutaan seuraavan. Erosuure $e(t)$ on asetusrvon ja säädetyn suureen erotus, jonka perusteella säädin muodostaa toimilaitetta ohjaavan säätösuureen $u(t)$. Toimilaite voi olla vaikkapa toimimootorilla ohjattava säätöventtiili (Eriksson 2002:2).



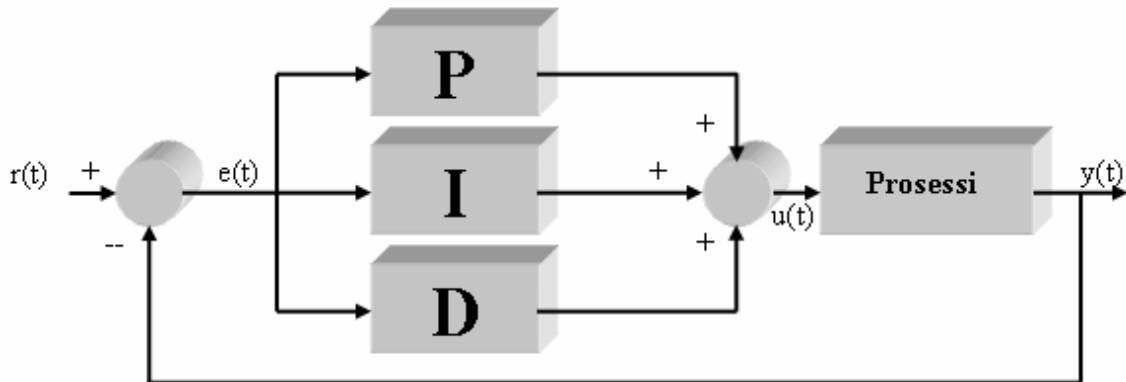
Kuva 4. Takaisinkytketyn säädön periaatekuva.

Takaisinkytketty säätöjärjestelmä toimii siten, että jos prosessin säädetty suure alkaa kasvaa esimerkiksi häiriön vaikutuksesta, kasvaa myös mittaussuure, minkä seurauksena erosuure puolestaan pienenee. Myös säätimen lähtösuure pienenee pienentäen prosessin lähtösuuretta kohti asetusrvoa (Eriksson 2002:2).

Säätimen tehtävä on siis huolehtia siitä, että säädetty suure seuraa mahdollisimman hyvin asetusrvoa torjuen mahdolliset säätöjärjestelmän ulkopuolelta tulevat häiriötekijät.

1.7.2. PID-säädin

Yksi säätötekniikan perussäätimistä on PID-säädin (Proportional – Integral – Derivate).



Kuva 5. PID-säätimen yleinen rakenne.

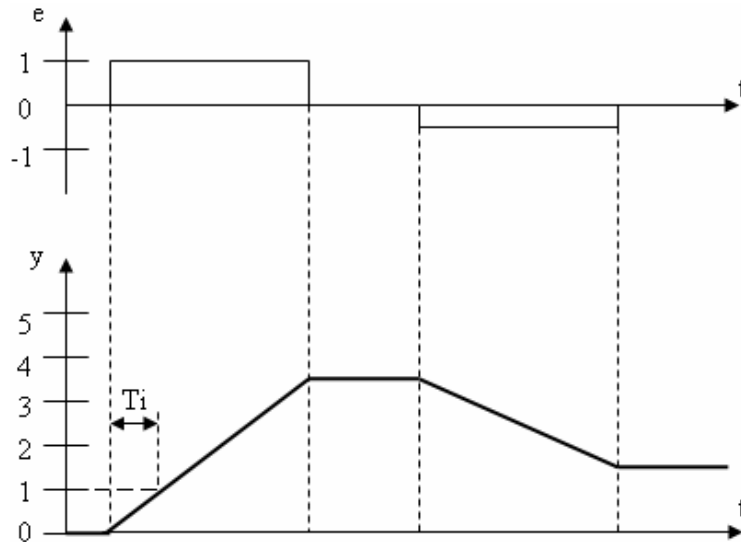
Säätimen ulostulo koostuu kolmesta eri termistä:

$$u = K_p e + K_I \int e dt + K_D \frac{de}{dt} \quad (1)$$

Suhdeosalla (P) tarkoitetaan toimintoa, jossa säätimen ulostulo (ohjaussuure) on suoraan verrannollinen säätimen sisäänmenoon. Tavallisesti sisäänmenona käytetään erosuureta (e), joka on asetusrvon ja mittausarvon erotus ($r-m$). Tällöin P-osan ulostulon itseisarvo on sitä suurempi, mitä kauempana säädettävän suureen arvo on toivotusta arvosta. Vahvistus K_p kuvaa säätötoimenpiteen voimakkuutta. P-termille on tyypillistä se, että se ei pysty kompensoimaan erosuureta kokonaan (Knuutila, Lindqvist & Rätty 2000).

Integroivalla osalla (I) tarkoitetaan seuraavanlaista toimintoa: Kun syntyy ykkösaskeleen suuruinen erosuure, alkaa ohjaussuure integroida erosuureen pinta-alaa. Ohjaussuure muuttuu suoraviivaisesti erosuureen ollessa vakio niin kauan kuin erosuure on olemassa. Erosuureen mentyä nolnaan jää ohjaussuure vakioarvoon ja pysyy siinä, kunnes

syntyy negatiivinen erosuure, jolloin ohjaussuure pienenee vakionopeudella niin kauan kuin erosuure on olemassa (Kuva 6). Integroivan termin vahvistuksena käytetään integrointivahvistusta K_I (Knuutila 2000).



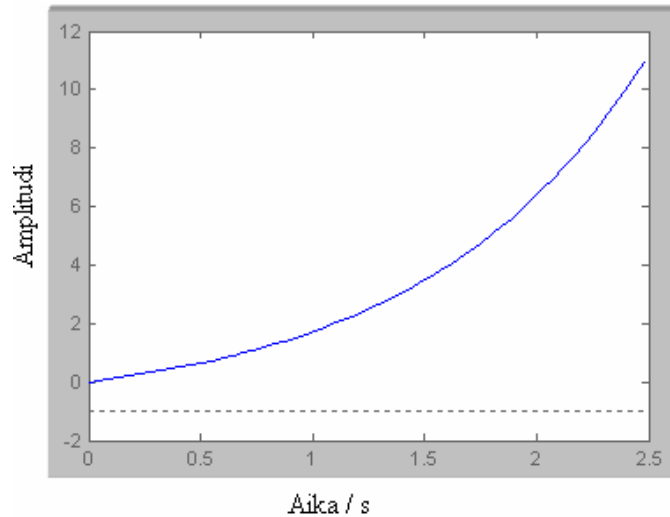
Kuva 6. I-säätimeen syötetään erosuurena yksikköaskelfunktio. Kun ohjaussuure on muuttunut ykkösen verran (=erosuure), niin integrointiaika T_i on kulunut. Integrointiaika tarkoittaa aikaa, missä integraattorin lähtö saavuttaa sisääntuloarvon ja se määritetään kuvassa 6 esitetyllä tavalla (Halme 1992).

Derivoiva osa (D) tarkastelee erosuureen muutosnopeutta, ja sen kohdalla puhutaan derivointivahvistuksesta K_D . D-säätö reagoi toimintatapansa mukaisesti hyvin nopeasti ja aggressiivisesti syntyvään erosuureeseen. Derivoivaa osaa kutsutaan myös ennakkoivaksi säädöksi, koska se pyrkii kompensoimaan poikkeaman jo siinä vaiheessa, kun se on vasta muodostumassa (Knuutila 2000).

Käytännössä PID-säätimestä käytetään usein yksinkertaistettuja versioita. Useissa käytännön tilanteissa saavutetaan riittävän hyvä säätötarkkuus, vaikka kaikkea PID-säätimen potentiaalia ei hyödynnettäisikään. Tällöin voidaan tarpeettomaksi katsottu tai joissain tilanteissa jopa ongelmia tuottava termi jättää säädöstä kokonaan pois. Tyypillisiä ovat esimerkiksi PI- ja PD-säätimet (Wikipedia 2006).

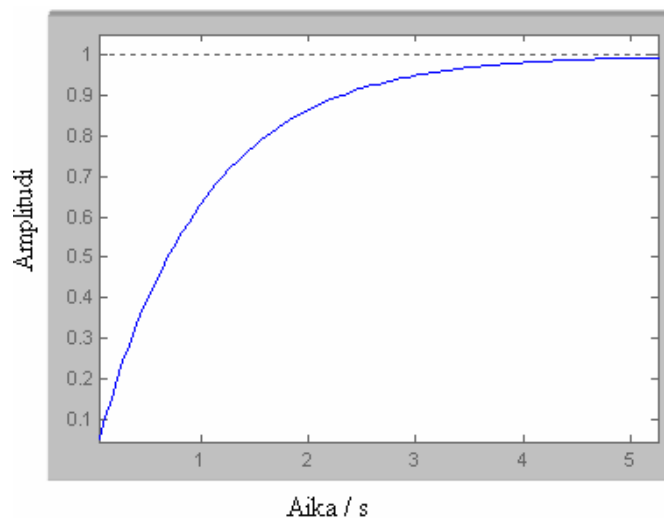
1.7.3. Stabiilisuus ja epästabiilisuus

Järjestelmä on epästabiili (kuva 7), jos muutos tulosuureessa saa aikaan sen, että vaste kasvaa rajattomasti.



Kuva 7. Epästabiili järjestelmä, missä vaste kasvaa rajatta, eikä lähesty mitään raja-arvoa (Tervo 2006).

Vastaavasti systeemi on stabiili (kuva 8), jos vaste saavuttaa jonkin vakioarvon äärellisessä ajassa. Järjestelmä on myös stabiili, jos se oskilloi (värähtelee) vakioamplitudilla (stabiili, mutta ei asympotoottisesti stabiili) (Tervo 2006).



Kuva 8. Järjestelmä on stabiili, kun rajattu tulo aiheuttaa rajatun lähdön (Tervo 2006).

1.7.4. PID-säätimen virittäminen

PID-säädin voidaan virittää useilla erilaisilla tavoilla. Yksiselitteisesti oikeita arvoja säätimen parametreille ei voida nimetä. Valittavat parametriarvot riippuvat sekä säädettävän suureen käyttäytymisestä että säädölle asetetuista tavoitteista. Käytännön tilanteissa jonkin tietyn ominaisuuden parantaminen voi aiheuttaa jonkin toisen ominaisuuden huononemisen. Esimerkiksi vahvistusta kasvattamalla saadaan säätöpiiri reagoimaan asetusarvon muutoksiin nopeasti, mikä on säädön kannalta positiivinen asia. Toisaalta vahvistuksen lisäys vaikuttaa järjestelmään myös negatiivisesti, heikentäen stabiilisuutta (Eriksson 2002:7-11).

Tavallisesti viritys suoritetaan etsimällä P-säätimen kriittisen vahvistuksen arvo. Tällöin säätöpiiri toimii stabiilisuusrajalla ja alkaa värähdellä vakioamplitudilla. Kriittisen vahvistuksen ja kriittisen värähtelyn jaksonajan avulla voidaan arvioida sopivia säätimen parametreja (Eriksson 2002:7-11).

Käytännössä säätimen hienovirityksessä käytetään myös yrityksen ja erehdyksen menettelmää. Tällöin on tunnettava, miten kukin termi vaikuttaa säätöön. Vahvistuksen (P) kasvattaminen lisää säädön nopeutta, vähentäen pysyvän tilan poikkeamaa. Lisäksi se heikentää stabiilisuutta eli kasvattaa säädön asettumisaikaa ja maksimipoikkeamia. Paimottamalla integroivaa termiä (I) pysyvän tilan virhe poistuu. Tällöin säädön nopeus kasvaa ja stabiilius jälleen heikkenee. Käyttämällä derivoivaa termiä (D) voidaan stabiilisuutta jossain tapauksissa parantaa. Derivoivan termin käyttö ei kuitenkaan ole kaikissa tilanteissa mahdollista (Eriksson 2002:7-11).

1.7.5. PID ja sumea säätö

Sumea säätö on yksi sumean logiikan laajimmista sovellutusalueista. Sen etuna voidaan pitää yksinkertaisuutta ja luonnollista ajattelutapaa, joka tekee säädöstä helpommin ymmärrettävää. Sumeaa säädintä käytetään normaalisti prosessin säätimenä. Säädintä voidaan käyttää myös PID-säädön rinnalla esimerkiksi muutostilanteissa. Lisäksi sitä voidaan käyttää laskemaan PID-säätimen parametrit eri toimintapisteissä. Sumean säädön tärkein etu on järjestelmän ymmärrettävyys. Säätimen muodostaminen on mahdol-

lista ilman hankalia matemaattisia malleja, ja se on mahdollista virittää ilman matemaatiikan tai säätötekniikan tuntemusta. Haittapuolena voidaan pitää vaikeata simulointia ja testattavuutta (Laurikkala 2006).

1.7.6. PID ja geneettiset algoritmit

PID-säätimen vahvistusparametrien optimoinnissa eri toimintapisteissä on käytetty myös geneettisiä algoritmeja, jotka ovat uudehko tulokas oppivien ja älykkäiden menetelmien joukossa. Geneettisillä algoritmeilla pyritään löytämään jonkun annetun kohdefunktion minimi tai maksimi. Geneettinen algoritmi käyttää erilaisia rinnakkaisia ratkaisuyrityksiä (joukkoa, sukupolvea), joista genetiikasta tutuilla operaatioilla, risteytyksellä ja mutaatiolla, pyritään muodostamaan parempia ratkaisuyrityksiä. Paremmuus mitataan laskemalla kohdefunktion arvot näille yrityksille. Menetelmän voisi tiivistää tapana kehittää uusia muunnelmia nykyisestä ratkaisusta erehdyksen testinä, joka hylkää syntyneistä ratkaisuista huonoiten toimivat (Törmänen 1997).

Kuulapelin automaattiohjaus on toteutettu kahdella PD-säätimellä (x- ja y-akselit). Kyseiset säätimet ajavat tasot haluttuihin kulmiin kuulun paikan ja nopeuden perusteella.

1.8. I²C – väylä

Philipsin kehittämää yksinkertaista ja yleiskäyttöistä kaksi-johtimista IIC-sarjaväylää (Inter-Integrated Circuit bus, IIC, I2C tai I²C) käytetään lähinnä kulutus-, auto- ja teollisuuselektroniikan sovelluksissa. Näissä sovelluksissa tarvitaan yleensä jotakin älykästä ohjainta, joka on tavallisesti mikro-ohjain. Ohjaimen lisäksi järjestelmään tarvitaan joidakin yleiskäyttöisiä oheispiirejä, kuten näyttö, näppäimistö, ohjaimen tulo- ja lähtöliitäntöjä, ulkoisia muisteja tms. Lisäksi tarvitaan joitain sovellukseen liittyviä erikoispiirejä (Koskinen 2002: 286).

Kuulapelin tasojen kallistuksia ohjaavien askelmoottoreiden ohjainkortit on kytketty tietokoneeseen I2C-sarjaväylien välityksellä. Molemmilla korteilla on omat väylänsä. Tässä luvussa on esitetty I2C-sarjaväylän periaate.

1.8.1. Yleistä

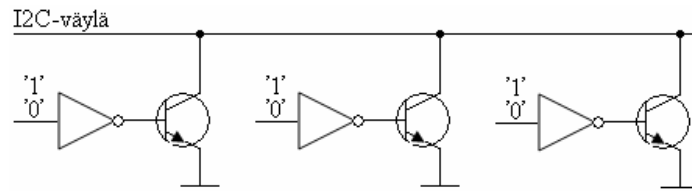
Väylään kiinnitettävien solmujen maksimikuormituskapasitanssiksi on sovittu 400 pF, joka vastaa noin kymmenen metrin pituista väylää ja muutamaa solmua. Eri osoitteita solmuille on käytettävissä kaikkiaan 127 kpl. I²C on todellinen väyläprotokolla, jossa on kehykseen upotettu vastaanottajan osoite ja jossa viestin luotettavuutta parannetaan kuittausmenetelmällä. I²C-protokolla ei tunne tarkistussummaa eikä virhekehyksiä. Se on siis suhteellisen yksinkertainen tiedonsiirtotapa, joka sopii luotettavuudeltaan keskinkertaisiin ratkaisuihin (Philips Semiconductors 2000).

I²C on sarjaväylä, johon voidaan liittää useita erilaisia muisti- ja I/O-piirejä, LCD-ohjaimia sekä erilaisia dataliityntöjä. Sitä käytetään mm. kulutuselektronikkatuotteissa, erityisesti TV:n sisäisenä väylänä se on suosittu. Useissa mikrokontrolleissa, mm. monissa Philipsin tuotteissa I²C-väyläohjain on sisäänrakennettuna. I²C-väylän simuloiminen ohjelmallisesti tavallisia I/O-linjoja käyttäen on mahdollista, mutta edellyttää kaksisuuntaista porttia ja open-collector-tyyppistä lähtöä (Koskinen 2002: 287).

1.8.2. I²C:n linjat

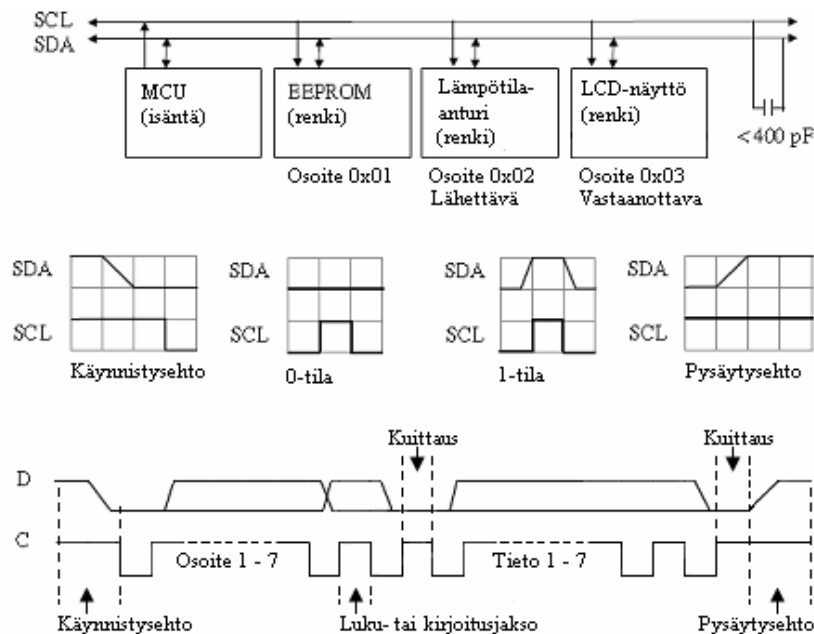
I²C sarjaväylässä on kaksi linjaa, nimittäin tietolinja (SDA) ja kellolinja (SCL). Väylä on synkroninen, joten tiedon tulee olla tahdissa kellolinjan kanssa. I²C-protokollassa datalla on kaksi tilaa: 1 ja 0 (kuva 9). Väylän open-collector-tyyppisyys vastaa tilannetta, jossa linja kelluu 1:llä ja 0:lla linja on vedetty alas aktiivisesti. Siten 1 on resessiivinen bitti ja 0 on dominoiva bitti (Philips Semiconductors 2000).

Tätä ominaisuutta käytetään hyväksi datan kuittauksessa odotustilan generoinnissa. Tieto siirretään kahdeksan bitin pituisina tavuina. Tavujen määrä yhdessä kehyksessä ei ole rajoitettu. Jokaista tavua seuraa kuittausbitti. Kuittauksen tekee vastaanottava solmu. Kuittauksessa vastaanottaja kirjoittaa dominoivan 0 lähettävän resessiivisen 1-bitin päälle. Mikäli vastaanottaja ei kykene tilapäisesti ottamaan vastaan seuraavaa tavua jonkin muun kiireellisen tehtävän vuoksi, se voi vetää kellolinjan alas, jolloin lähettäjä (isäntä) jää odottamaan (Philips Semiconductors 2000).



Kuva 9. I2C-väylä on toteutettu open-collector tyyppisillä ohjainpiireillä. Siten looginen 0 (dominoiva bitti) yliajaa loogisen 1 (ressessiivinen bitti) (Philips Semiconductors 2000).

1.8.3. Isäntä-renki-väylä



Kuva 10. Liikennöinti I2C-väylässä tapahtuu seuraavasti: Mikrokontrolleri (isäntä) lähettää käynnistysehdon laittamalla SDA:n alas ja kohta perään SCL:n alas. Tällöin kaikki muut laitteet alkavat kuunnella väylää. Seuraavaksi isäntä lähettää 7 tai 10 bittisen osoite tiedon, joka kertoo mihin laitteeseen isäntä haluaa olla yhteydessä ja heti perään tieto siitä mihin suuntaan (kirjoittaa/lukea) haluaa liikennöidä. Toiset laitteet ottavat bitit vastaan ja vertaavat omiin tunnuksiinsa. Jos tunnus ei ole sama, laite jää odotamaan lopetus ilmoitusta ja jos tunnus täsmää, laite kuittaa signaalin. Kun käsittely on hoidettu, isäntä aloittaa tiedon siirron. Kun tiedonsiirto saadaan loppuun, isäntä lähettää pysäytysehdon laittamalla SDA:n ja SCL:n ylös (Koskinen 2002: 286).

I²C on isäntä-renki-väylä. Kaikki lähetys on synkronoitu isännän kellosignaaliin, joten masteri aloittaa ja lopettaa viestin. Viestiin kuuluu tyhjiä kenttiä, joita renki-solmu voi täydentää kesken viestin lähetyksen. Renki tunnistaa itselleen kuuluvan lähetyksen viestin alussa olevasta 7-bittisestä osoitteesta. Näin I²C-verkossa voi olla 127 eri I²C-solmua. I²C on moni-isäntäinen väylä. Tämä tarkoittaa, että väylällä voi olla useita solmuja, jotka voivat aloittaa viestin. Tilannetta, jossa useat isännät pyrkivät lähettämään tietoa yhtä aikaa, hallitaan väyläarbitroinnilla. Arbitroinnilla se solmu, joka lähettää 1:n (resessiivinen) samaan aikaan jonkun toisen solmun lähettämän 0:n kanssa, häviää väyläkilvan ja vetäytyy lähetyksestä (Koskinen 2002: 286).

2. RAKENNE JA MEKANIikka

Tässä luvussa käsitellään lyhyesti kuulapelin rakennetta ja mekaniikkaa, sekä valitaan kuulapelille sopivin pelikuula erilaisten testien avulla. Mitään tarkkoja mitoituksia tai teknisiä arvoja ei rakenteesta esitetä, vaan ainoastaan prototyypin erilaisia rakenneratkaisuja, mahdollista myöhempää jatkokehittelyä varten.

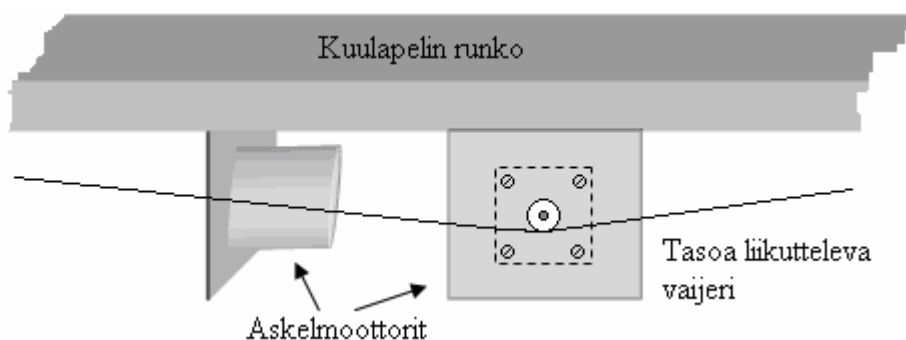
2.1. Kuulapelin rakenne

Kuulapeli koostuu kolmesta sisäkkäisestä puukehikosta (kuva 11), jotka on yhdistetty toisiinsa laakerien ja holkkien välityksellä. Pelitaso on sijoitettu sisimpään kehikkoon, ja se on vaihdettavissa.



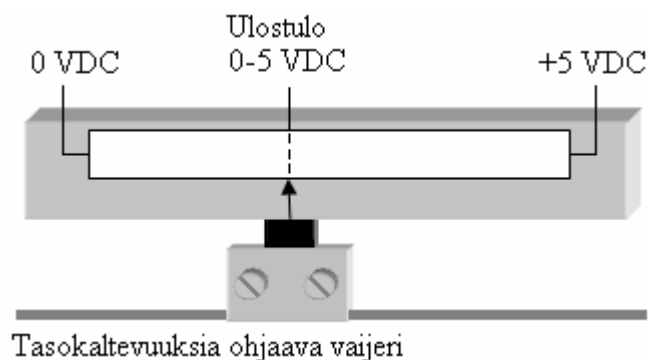
Kuva 11. Kuulapeli muodostuu kolmesta sisäkkäisestä puukehikosta, jotka on laakeroitu yhteen.

Tasojen kallistuskulmia ohjaavat askelmoottorit on sijoitettu pelitason alla olevaan runkoon kuvan 12 mukaisesti. Ne liikuttavat pelitasoja vaijerien välityksellä, sekä x- että y-suunnissa. Askelmoottorit ovat tyypiltään muuttuvan reluktanssin moottoreita (kuva 22) ja ohjaustavaltaan bipolaarisia (kuva 18).



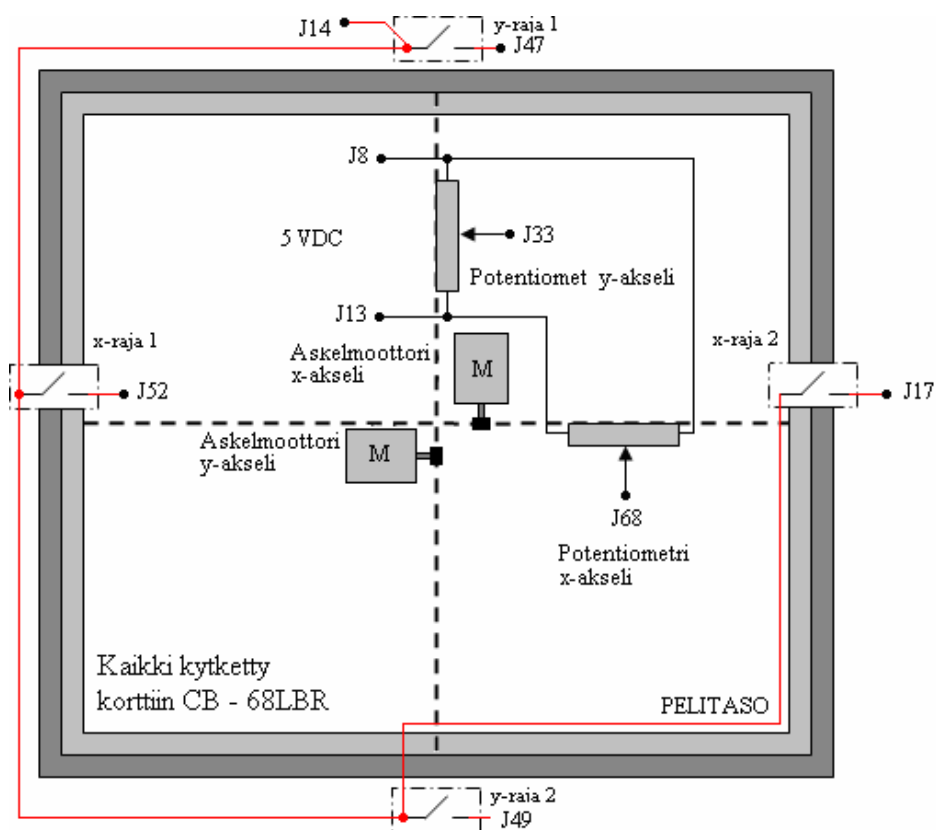
Kuva 12. Askelmoottorit on kiinnitetty pelin runkoon kulmarauodoilla.

Vaijereihin on kiinnitetty tasojen kaltevuuskulmia seuraavat liukupotentiometrit (kuva 13) ja laatikon kylkiin on sijoitettu kaltevuuden ääri rajoja valvovat mekaaniset anturit (kuva 14).

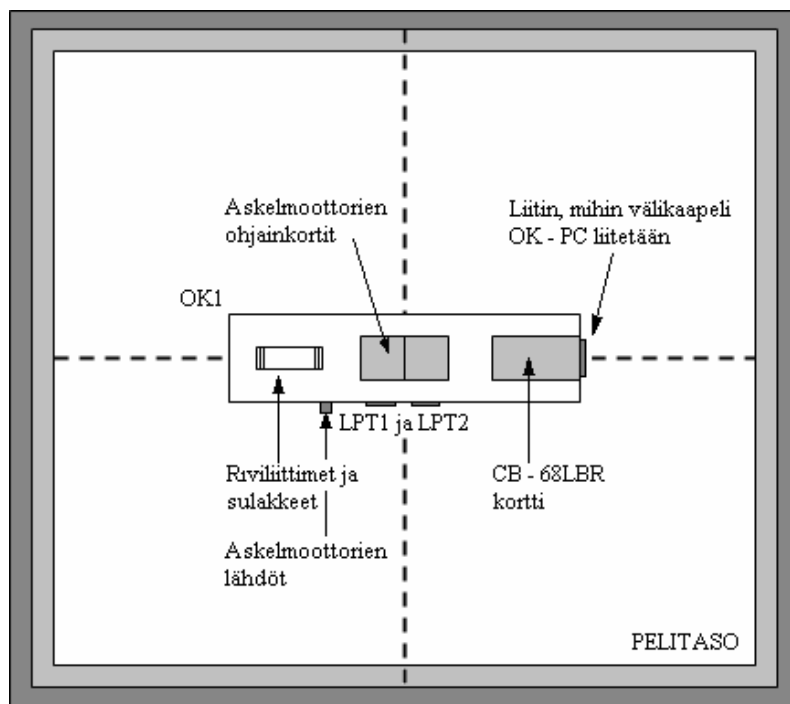


Kuva 13. Tason kaltevuutta seuraavan liukupotentiometrin kiinnitys.

Askelmoottorit ja rajat on kytketty ohjauskeskukseen (kuva 15). Kyseinen keskus sijaitsee pelitason alla olevassa rungossa.



Kuva 14. Rajojen, potentiometriä ja askelmoottorien sijainti ja kytkentä pelikehikolla.



Kuva 15. Ohjauskotelo ja sen sisältämät komponentit.

Pelin kamera ja halogeenivalaisimet on kiinnitetty (prototyypissä) sitä varten suunniteltuun telineeseen (kuva 16).

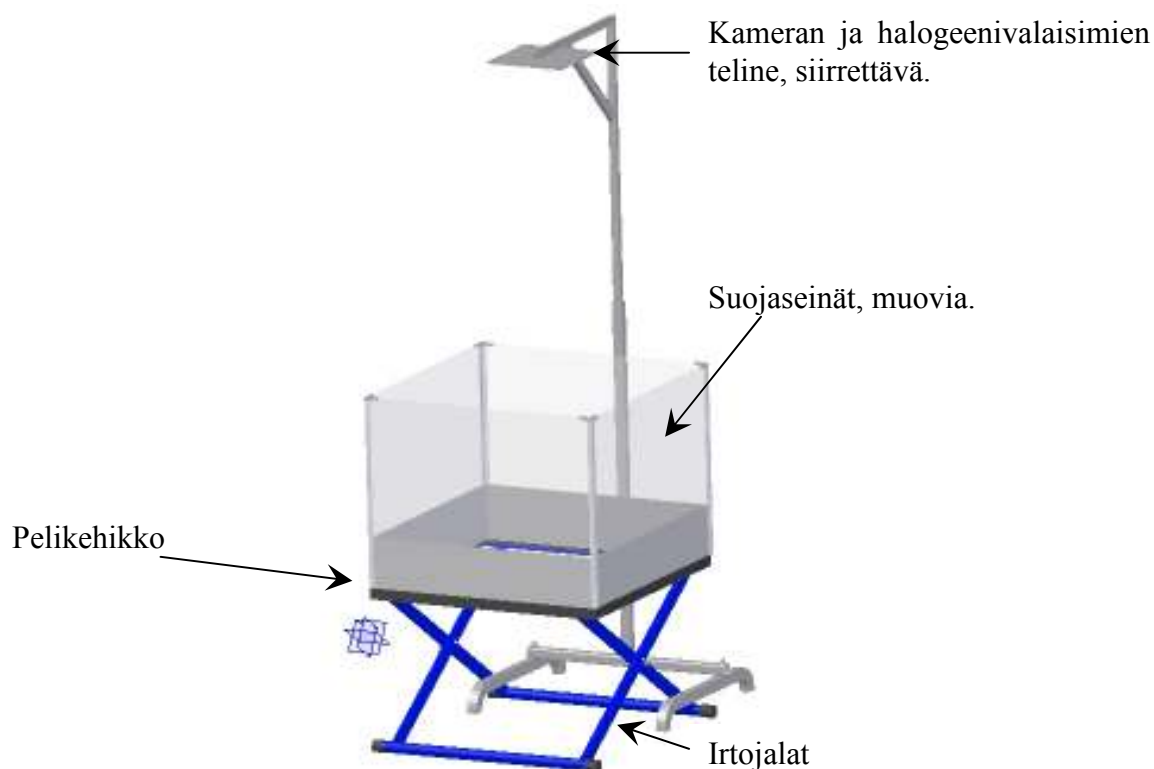


Kuva 16. Kuulapelin halogeenivalaisimet ja Firewire-kamera.

Edellä esitetyt kuvat ovat valmiista prototyypistä, mutta ne eivät selvitä, minkälainen pelin rakenne voisi olla jatkokehittelyn jälkeen.

Lähtökohta kuulapelin rakennesuunnittelulle oli turvallisuus, toimivuus ja kestävyys. Lisäksi kuulapelin siirtäminen paikasta toiseen tulisi olla mahdollisimman vaivatonta. Näiden alkuehtojen perusteella syntyi kuvan 17 mukainen rakenne-ehdotus.

Teline, johon kamera ja halogeenivalaisimet kiinnitetään, on siirrettävää mallia ja lisäksi sen saa kolmeen osaan. Pelin jalat ovat myös irrotettavissa ilman työkaluja, ja ne voidaan taittaa yhteen. Suojaseinät saa poistettua nostamalla ne irti pidikkeistään, ja pelitason saa pelikehikosta irti suhteellisen vaivattomasti.



Kuva 17. Kuulapelin rakenne-ehdotus jatkokehitystä varten.

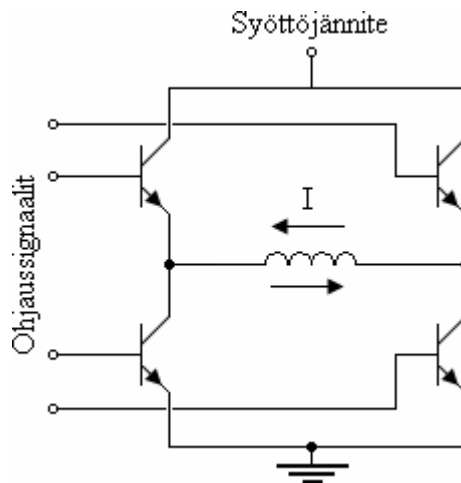
2.2. Askelmoottorit

Askelmoottoreita on monenlaisia, niin rakenteeltaan kuin ohjaustavoiltaan. Rakenteensa puolesta toisistaan eroavat esimerkiksi muuttuvan reluktanssin moottori, kestmagnetoitu askelmoottori ja hybridimoottori. Ohjaustavoista tyypillisiä ovat bipolaarinen- ja unipolaarinen ohjaus (Keinänen, Kärkkäinen, Metso & Putkonen 2002:138–139).

Kuulapelin motoriikka on hoidettu askelmoottoreilla. Tässä luvussa kerrotaan yleisesti, minkälaisia askelmoottoreita on olemassa ja kuinka ne toimivat.

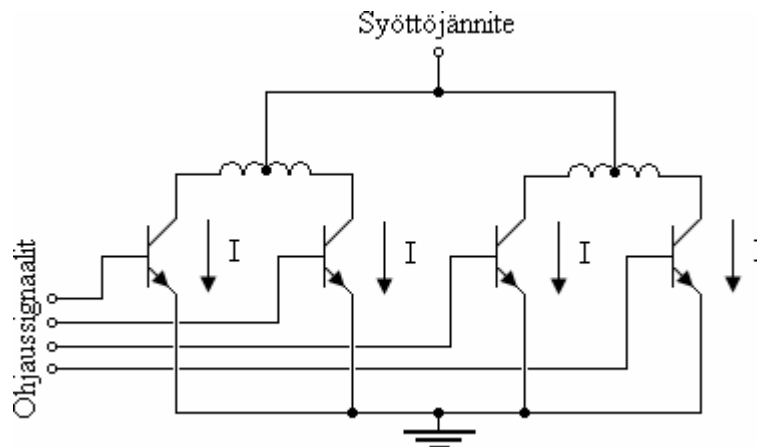
2.2.1. Yleistä

Askelmoottori sopii erityisen hyvin tietokoneella tai ohjelmoitavalla logiikalla tapahtuvaan ohjaukseen, koska ohjauspulssit ovat kaksitilaisia. Ohjaustapoina käytetään joko bipolaarista- tai unipolaarista ohjausta. Nämä liittyvät ohjaustransistorien kahteen erilaiseen kytkentätapaan (Keinänen, Kärkkäinen, Metso & Putkonen 2002:138–139).



Kuva 18. Bipolaarinen ohjaus mahdollistaa virran kulkemisen molempiin suuntiin moottorin käämissä (Airila 1999: 35–36).

Kuvan 18 bipolaarisen ohjauksen etuna on hyvä hyötysuhde ja moottorista saatavat maksimaaliset ominaisuudet. Ohjausta käytetään suurta suorituskykyä vaativiin toimintoihin. Bipolaarinen ohjauspiiri on monimutkaisempi kuin vastaava unipolarinen ohjauspiiri (Airila 1999: 35–36).

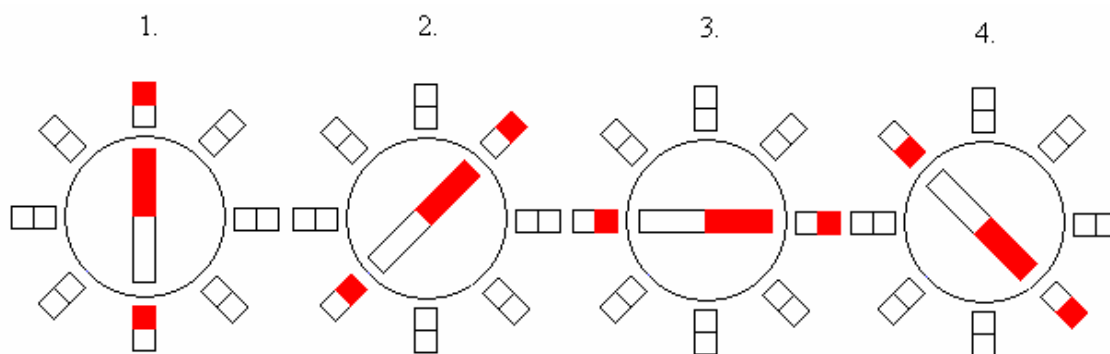


Kuva 19. Unipolarisella ohjauksella virta kulkee yhteen suuntaan moottorin käämissä (Airila 1999: 35–36).

Kuvan 19 unipolarinen ohjaus mahdollistaa yksinkertaisemman ohjauspiirin käytön ja vähäisemmät kustannukset, mutta moottorista ei saada maksimaalisia ominaisuuksia (Airila 1999: 35–36).

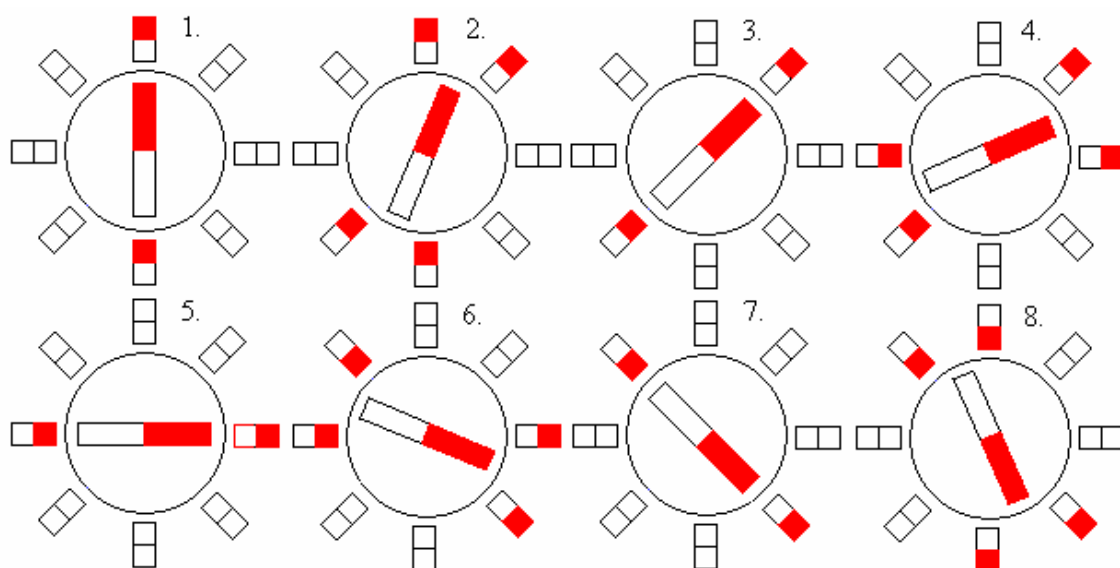
2.2.2. Askelmoottorin toimintaperiaate

Askelmoottorin staattori muodostuu sähkömagneetista ja roottori kestopagneetista. Roottorin kestopagneetin muodostavat N- ja S-navat. Staattorikämmityksien muodostama magneettikenttä vetää puoleensa näitä magneettinapoja saaden aikaan roottorin kääntymisen askelittain. Askelluskulma riippuu moottorin magneettinapojen lukumäärästä. Tarkoilla askelmoottoreilla askelluskulma on alle yhden asteen (Kankaanpää 2003).



Kuva 20. Kokoaskelluksen toimintakaavio. Staattorikämmityksien muodostama magneettikenttä vetää puoleensa roottorin magneettinapoja saaden aikaan roottorin kääntymisen askelittain (Harries 1997).

Staattorikämmien askellusjärjestystä muuttamalla voidaan muuttaa moottorin pyörimissuuntaa ja askellustaajuutta muuttamalla moottorin pyörimisnopeutta (kuva 20) (Kankaanpää 2003).



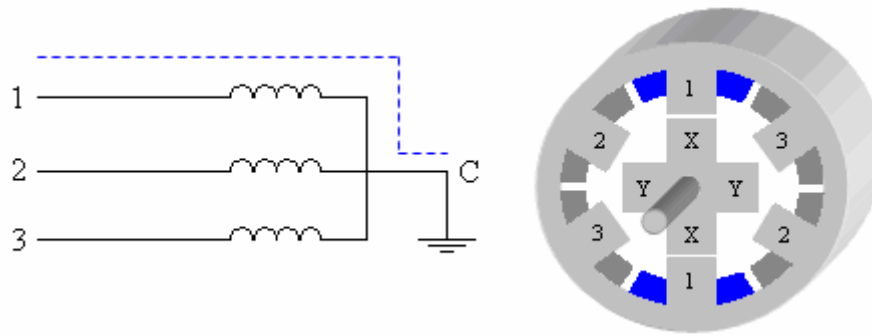
Kuva 21. Puoliaskelluksen toimintakaavio. Ohjaus tuodaan välillä aina vierekkäisille käämeille yhtä aikaa, jolloin roottori kääntyy käämien muodostamien magneettikenttien resultantin suuntaisesti (Harries 1997).

Tarkkuutta voidaan kasvattaa myös käyttämällä puoliaskelohjausta. Tällöin ohjaus tuodaan välillä aina vierekkäisille käämeille yhtä aikaa, jolloin roottori kääntyy käämien muodostamien magneettikenttien resultantin suuntaisesti. Tällöin askellustarkkuus kaksinkertaistuu, mutta nopeus vastaavasti puolittuu (kuva 21). Askellustarkkuutta voidaan lisätä vielä jakamalla askeleet useampiin osiin (mikroaskellus) (Kankaanpää 2003).

Askelmoottoreita on pääasiassa kolmea eri tyyppiä. Ne ovat muuttuvan reluktanssin moottori (kuva 22), kestmagnetoitu askelmoottori (kuva 23) ja hybridimoottori (kuva 24) (Airila, 1999: 28–31).

2.2.3. Muuttuvan reluktanssin moottori

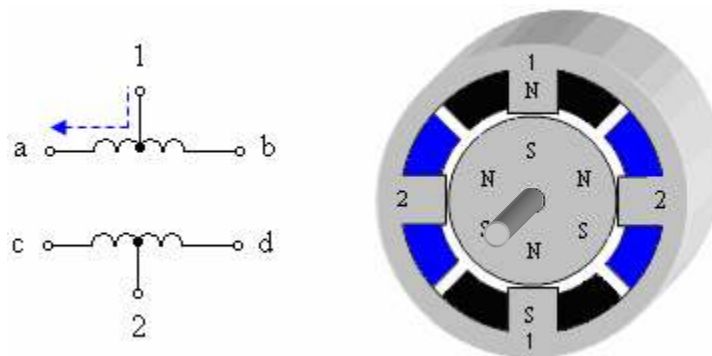
Muuttuvan reluktanssin moottori on rakenteeltaan selkein. Se koostuu monihampaisesta metallisesta roottorista ja metallisesta käämitetystä staattorista, jossa käämit kiertävät roottorin säteen ympäri. Kun osaan staattorin käämeistä johdetaan jännite, pyrkii roottorin hammastukset asettumaan muodostuvan magneettikentän suuntaisesti, aiheuttaen vääntömomentin roottorin akselille. (Jones 2004:2).



Kuva 22. Muuttuvan reluktanssin askelmoottori koostuu monihampaisesta metallisesta roottorista ja metallisesta käämitetystä staattorista, jossa käämit kiertävät roottorin säteen ympäri. Kun jännitettä ohjataan eri käämeihin 1, 2 ja 3 tietyssä järjestyksessä (C on yhteydessä maahan), saadaan aikaan jatkuva kiertoliike (Jones 2004:2).

2.2.4. Kestomagnetoitu askelmoottori

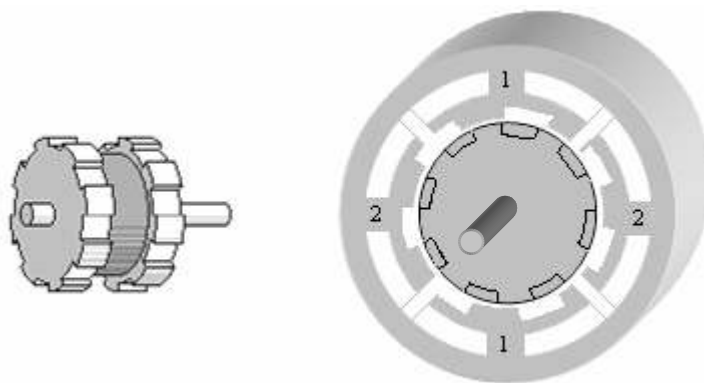
Kestomagnetoitu askelmoottori on suhteellisen epätarkka, mutta edellistä voimakkaampi askelmoottorityyppi. Sen roottorin metallinen hammastus on korvattu kestmagneetilla, jossa napaisuus vaihtelee roottorin pyörimissuunnassa. Tästä johtuen momentin aiheuttava magneettikenttä on voimakkaampi, kehittäen suuremman vääntömomentin (Jones 2004:2).



Kuva 23. Kestomagnetoitun askelmoottorin (unipolar) roottorissa on metallinen hammastus korvattu kestmagneetilla, jossa napaisuus vaihtelee roottorin pyörimissuunnassa. Käämit ovat kaksiosaisia. Kun jännitettä ohjataan eri käämeihin 1-a, 2-b, 2-c ja 2-d tietyssä järjestyksessä, saadaan aikaan jatkuva kiertoliike (Jones 2004:2).

2.2.5. Hybridimoottori

Hybridimoottori yhdistää edellisten moottorityyppien parhaat puolet. Se on voimakkaampi ja tarkempi kuin edelliset, mikä näkyy myös sen hinnassa. Hybridimoottori koostuu kestmagneettisydämisestä hammastetusta metalliroottorista, jossa magneettikenttä on akselin suuntainen, ja metallisesta staattorista, jossa samaan napapariin on sijoitettu kahden vaiheen käämitys. Tällä tavalla saadaan voimakkaampi magneettikenttä ja suurempi vääntömomentti, sekä suuremmalla määrällä roottorin hammastusta suurempi määrä askeleita eli pienempi askellus (Jones 2004:2).



Kuva 24. Hybridi askelmoottori koostuu kestmagneettisydämisestä hammastetusta metalliroottorista, jossa magneettikenttä on akselin suuntainen, ja metallisesta staattorista, jossa samaan napapariin on sijoitettu kahden vaiheen käämitys (Jones 2004:3).

Askelmoottoreita on myös muita tyyppejä, mutta tällä hetkellä enemmänkin erikoissovellutuksissa, joissa edellisten askelmoottoreiden ominaisuudet eivät riitä. Näistä mainittakoon levymagneettimoottori, jossa roottori on valmistettu magnetoidusta levystä, jonka napaisuus vaihtelee pyörimisliikkeen suunnassa. Tällä saavutetaan muun muassa pieni inertia (liikemäärä) ja optimaalisempi magneettivuo (Nevalainen 2004: 3).

Kuulapeliin valittiin muuttuvan reluktanssin askelmoottorit bipolaarisella ohjauksella. Paras ja helpoin vaihtoehto ohjaukseen olisi ollut servomoottori tai lineaarimoottori nopeutensa ja tarkkuutensa vuoksi, mutta niiden kallis hinta esti kyseisten moottorien käytön. KytKentä on esitetty liitteessä 1.

2.3. Pelikuulan testaaminen

Kitka eli liikevastus on kahden kappaleen pintamolekyylien keskinäisestä vetovoimasta johtuva liikettä vastustava voima, joka ilmenee kahden toisiaan koskettavan pinnan välissä. Kitkan voittamiseksi kappaleen on tehtävä työtä yhtälön $W = Fs$ mukaisesti, missä vakiovoima F vetää kappaletta omaan suuntansa matkan s . Tällöin kappaleeseen varastoitunut energia muuttuu muun muassa lämmöksi (Mäkelä 1997).

Kitkan eri lajeja ovat vierimiskitka, pyörimiskitka sekä liukumiskitka. Lepokitka pyrkii vastustamaan liikkeelle lähtöä, ja liikekitka pyrkii pysäyttämään jo alkaneen liikkeen. Lepokitka on aina suurempi kuin kappaleen liikekitka, minkä seurauksena kappaleen lähdettyä liikkeelle se on helpompi pitää liikkeessä. Kitkakertoimet eri aineiden välille on määritelty kokeellisesti, ja ne ovat suuruudeltaan aina pienempiä kuin 1. Kitkavoima on vektorisuure, ja sillä on siis suuruuden lisäksi aina myös suunta. Kitkaa voidaan usein pienentää muun muassa pintoja tasoittamalla tai käyttämällä voiteluainetta, sekä muuttamalla liukumiskitka vierimiskitkaksi (Wikipedia 2006).

Ensimmäisenä kriteerinä pelikuulalle oli pitää lepokitka mahdollisimman pienenä. Liian suuri lepokitka aiheuttaa sen, että tasoja täytyy kallistaa melkoisesti kuulan liikkeelle saattamiseksi. Kallistuksen seurauksena kuulan lähtönopeus saattaa nousta niin suureksi, että kuulaa on vaikea hallita.

Tutkittavana oli 11 erilaista kuulatyyppeä taulukon 1 mukaisesti.

1,34 V:n muutos tasapainotilasta kumpaan tahansa suuntaan aiheutti $0,7 \pm 0,1$ cm kallistuksen pelitason keskiviivaan nähden. Mittaus suoritettiin työntömitalla. Tason leveys keskiviivasta kallistuksen mittauskohtaan on $37,2 \pm 0,3$ cm. Mittaus suoritettiin rullamitalla.

Kallistuskulma on tällöin

$$\sin \alpha = 0,7 \pm 0,1 \text{ cm} / 37,2 \pm 0,3 \text{ cm}, \text{ joten } \alpha = 1,1 \pm 0,1^\circ, \quad (2)$$

toisin sanoen $1,3 \pm 0,05$ V vastaa $1,1 \pm 0,1^\circ$, joten $1 \text{ V} \pm 0,05 \text{ V}$ vastaa $0,8 \pm 0,1^\circ$ kallistusta.

Testaus suoritettiin y-akselilla, jonka ääriarajat ovat 0 V ja 5 V ja tasapainotila 2,5 V.

Taulukko 1 Testattavat kuulat.

Muoto	Materiaali	Pinta	Halkaisija / mm	Paino / g
1. kuula	teräs, kiiltävä	sileä	$20,0 \pm 0,1$	$28,2 \pm 0,1$
2. kuula	teräs, kiiltävä	sileä	$16,0 \pm 0,1$	$16,7 \pm 0,1$
3. kuula	teräs, kiiltävä	sileä	$8,0 \pm 0,1$	$2,1 \pm 0,1$
4. kuula	teräs, maalattu	sileä	$20,0 \pm 0,1$	$28,2 \pm 0,1$
5. kuula	marmori, maalattu	epätasainen	$24,5 \pm 0,1$	$18,5 \pm 0,1$
6. golf-pallo	kumi	epätasainen	$39,5 \pm 0,1$	$25,8 \pm 0,1$
7. kuula	muovi	sileä	$35,5 \pm 0,1$	$5,1 \pm 0,1$
8. kuula	muovi	sileä	$14,0 \pm 0,1$	$1,4 \pm 0,1$
9. kuula	lasi	sileä	$16,0 \pm 0,1$	$5,4 \pm 0,1$
10. kuula	kumitettu teräs	sileä	$22,0 \pm 0,1$	$26,5 \pm 0,1$
11. kuula	kumitettu muovi	sileä	$22,0 \pm 0,1$	$8,4 \pm 0,1$

Itse mittauksessa pelitaso asetettiin tasapainotilaan ja kuula asetettiin pelitasolle. Tasoa kallistettiin hitaasti, kunnes kuula lähti vierimään. Jokaisella kuulalla tehtiin viisi samanlaista testausta, ja näistä laskettiin keskiarvo. Kuulien ja tason epätasaisuudet aiheuttivat hiukan ongelmia mittauksissa. Kuula saattoi jäädä joissakin mittauksissa kiinni alustaansa, mikä johtui alustan epätasaisuuksista. Tasoa piti tällöin kallistaa melkoisesti, ennen kuin kuula lähti liikkeelle. Jos kyseinen kulma-arvo poikkesi huomattavasti neljästä muusta mittauksesta, niin sitä ei otettu huomioon keskiarvossa. Saadut tulokset on esitetty taulukossa 2.

Seuraavaksi selvitettiin laskennallisesti, minkälainen nopeus kullakin kuulalla oli 10 cm etenemisen jälkeen. Nopeuden laskemiseen käytettiin seuraavanlaista kaavaa (M. Mäkelä, R. Mäkelä & Siltanen 1997):

Kuula on pudonnut maan vetovoiman vaikutuksessa matkan

$$h = L \sin \theta. \quad (3)$$

Kuulalle on tällöin tullut energiaa

$$E_p = mgh = mgL \sin \theta. \quad (4)$$

Kyseinen energia on mennyt kuulan rotaatioenergiaksi

$$E_R = J\omega^2 / 2 \quad (5)$$

ja lineaarienergiaksi

$$E_L = mv^2 / 2. \quad (6)$$

Energia säilyy, joten

$$E_p = E_R + E_L. \quad (7)$$

Tällöin

$$mgL \sin \theta = J\omega^2 / 2 + mv^2 / 2 \quad (8)$$

$$\Leftrightarrow mgL \sin \theta = J\omega^2 / 2 + mv^2 / 2.$$

Kun kuula vierii,

$$\omega = v / R \quad (9)$$

$$mgL \sin \theta = (2mR^2 / 5)v^2 / 2R^2 + mv^2 / 2 \quad (10)$$

$$\Leftrightarrow 2gL \sin \theta = 2v^2 / 5 + v^2$$

$$\Leftrightarrow 2gL \sin \theta = 7v^2 / 5$$

$$\Leftrightarrow v^2 = 10gL \sin \theta / 7.$$

Kun esimerkiksi kiiltävän teräskuulan mittausarvot sijoitetaan kaavaan, saadaan kyseisen kuulan nopeudeksi $4,9 \pm 0,5$ cm/s $10 \pm 0,3$ cm vierimisen jälkeen. Virheen määrittely muodostui suhteellisen hankalaksi, koska kuulat saattoivat jumittua tasopintaan kuulan tai tasopinnan epätasaisuuksien vuoksi useammassakin mittauksessa. Tällöin ne vaativat huomattavasti suuremman, edellisistä mittauksista poikkeavan kallistuksen irrotukseen. Tämän vuoksi virhemarginaalit ovat epätarkkoja, mutta suuntaa antavia. Tulokset on esitetty taulukossa 2.

Seuraavaksi selvitettiin kuulan kimmoisuus. Tasoa kallistettiin $1,13^\circ$ ja päästettiin kuula vapaasti vierimään ylälaidasta alalaitaan, ja mitattiin kyseiseen matkaan käytetty aika ja se, kuinka kauas kuula kimposi takaisin alalaitaan osuessaan. Jokaisella kuulalla tehtiin jälleen viisi samanlaista testausta, ja näistä laskettiin keskiarvo. Lisäksi mittauksesta laskettiin kuulan keskinopeus. Tulokset on esitetty taulukossa 2.

Taulukko 2 Testattujen kuulien valintakriteerit pelikuulaksi.

Muoto	Keskiarvo / $\alpha_L / ^\circ$	Nopeus 10cm:n jälkeen cm / s	Jousto seinästä / cm	Keskinopeus cm / s
1. kuula	$0,7 \pm 0,1$	$13,1 \pm 0,5$	$10,0 \pm 0,6$	$31,3 \pm 0,5$
2. kuula	$0,7 \pm 0,1$	$13,1 \pm 0,5$	$12,2 \pm 0,7$	$30,8 \pm 0,5$
3. kuula	$0,8 \pm 0,1$	$14,0 \pm 0,5$	$0,6 \pm 0,1$	$30,0 \pm 0,5$
4. kuula	$0,8 \pm 0,1$	$14,0 \pm 0,5$	$11,0 \pm 0,6$	$29,9 \pm 0,5$
5. kuula	$0,9 \pm 0,3$	$14,8 \pm 0,8$	$3,8 \pm 0,6$	$31,3 \pm 0,8$
6. golf-pallo	$1,1 \pm 0,3$	$16,4 \pm 0,8$	$5,5 \pm 1,0$	$24,6 \pm 0,8$
7. kuula	$0,8 \pm 0,2$	$14,0 \pm 0,6$	yli reunan	$28,5 \pm 0,6$
8. kuula	$1,1 \pm 0,2$	$16,4 \pm 0,6$	$1,3 \pm 0,1$	$28,2 \pm 0,6$
9. kuula	$0,8 \pm 0,1$	$14,0 \pm 0,5$	$8,2 \pm 1,2$	$31,8 \pm 0,5$
10. kuula	$0,8 \pm 0,1$	$14,0 \pm 0,5$	$2,9 \pm 0,2$	$29,7 \pm 0,5$
11. kuula	$1,0 \pm 0,1$	$15,6 \pm 0,5$	$1,4 \pm 0,1$	$28,9 \pm 0,5$

Kuulien liikkeellelähtökulmat α_L vaihtelivat välillä $0,7^\circ$ - $1,1^\circ$ (taulukko 2.2). Mitä suurempi liikkeellelähtökulma α_L on, sitä suuremmaksi kuulan lähtönopeus v_L nousee, mikä vaikeuttaa kuulan hallittavuutta. Kuulien 5 ja 6 epätasainen pinta aiheutti aina välillä sen, että kuula lähti liikkeelle hyvinkin suurella kulma-arvolla α_L . Tästä johtuen kuulat 5 ja 6 hylättiin välittömästi niiden pintojen epätasaisuuksien vuoksi. Kuulat 3 ja 7

hylättiin kokonsa ja keveytensä vuoksi, koska esimerkiksi kameraseuranta kyseisillä kuulilla oli erittäin vaikeaa. Kuulien 1, 2, 4 ja 9 hallittavuus oli erittäin hankalaa. Esimerkiksi törmätessään seinään pienelläkin nopeudella kimmoisuus oli suhteellisen suuri ja liikkeen hallinta vaikeaa. Lisäksi heijastavan tai läpinäkyvän pintamateriaalin vuoksi kameraseuranta oli kyseisillä kuulilla melko vaikeaa. Kuula 8 oli hiukan epäsymmetrinen, joten sen meno oli tasolla hiukan nykivää. Kuulat 10 ja 11 olivat hyvinkin soveltuvia pelin käyttökuuliksi. Kuulan 10 kimmoisuus oli hiukan suurempi kuin kuulan 11 mikä vaikutti kyseisen kuulan hallittavuuteen. Kuula 11 oli pintaväriiltään musta, joten se erottui kameraseurannassa erittäin hyvin. Näiden seikkojen vuoksi pelikuulaksi valittiin kuula numero 11.

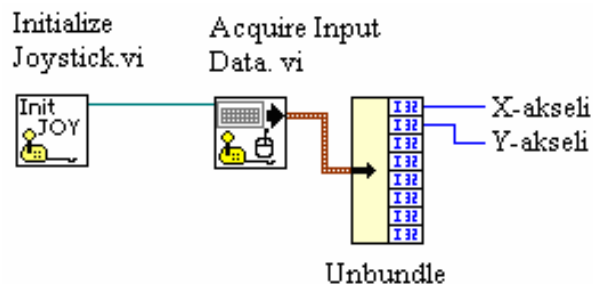
3.1. Manuaaliohjaus

Kuten jo aikaisemmin mainittiin, LabVIEW-ohjelma perustuu niin sanottujen ohjelma-moduulien eli virtuaali-instrumenttien (VI) käyttöön. Nimi virtuaali-instrumentti on tullut siitä, että instrumenttien ulkonäkö ja toiminta on rakennettu mukailemaan mahdollisimman tarkasti fyysisiä eli olemassa olevia laitteita ja komponentteja (National Instruments 2006).

Manuaaliohjaukseen käytettävä Joystick-ohjain on liitetty tietokoneeseen USB-portin kautta. LabVIEW-sovelluksessa on Joystick-ohjaimelle oma virtuaali-instrumenttinsa (Initialize Joystick.vi), joka sisältää kyseiselle ohjaimelle tarvittavat liityntäasetukset.

Tämä ei kuitenkaan vielä yksistään riitä Joystick-ohjaimelta lähetetyn tiedon hyödyntämiseen. Kyseinen tieto täytyy vielä muuntaa ohjaukseen soveltuvaksi. Tätä toimenpidettä varten on oma virtuaali-instrumenttinsa (Acquire Input Data.vi).

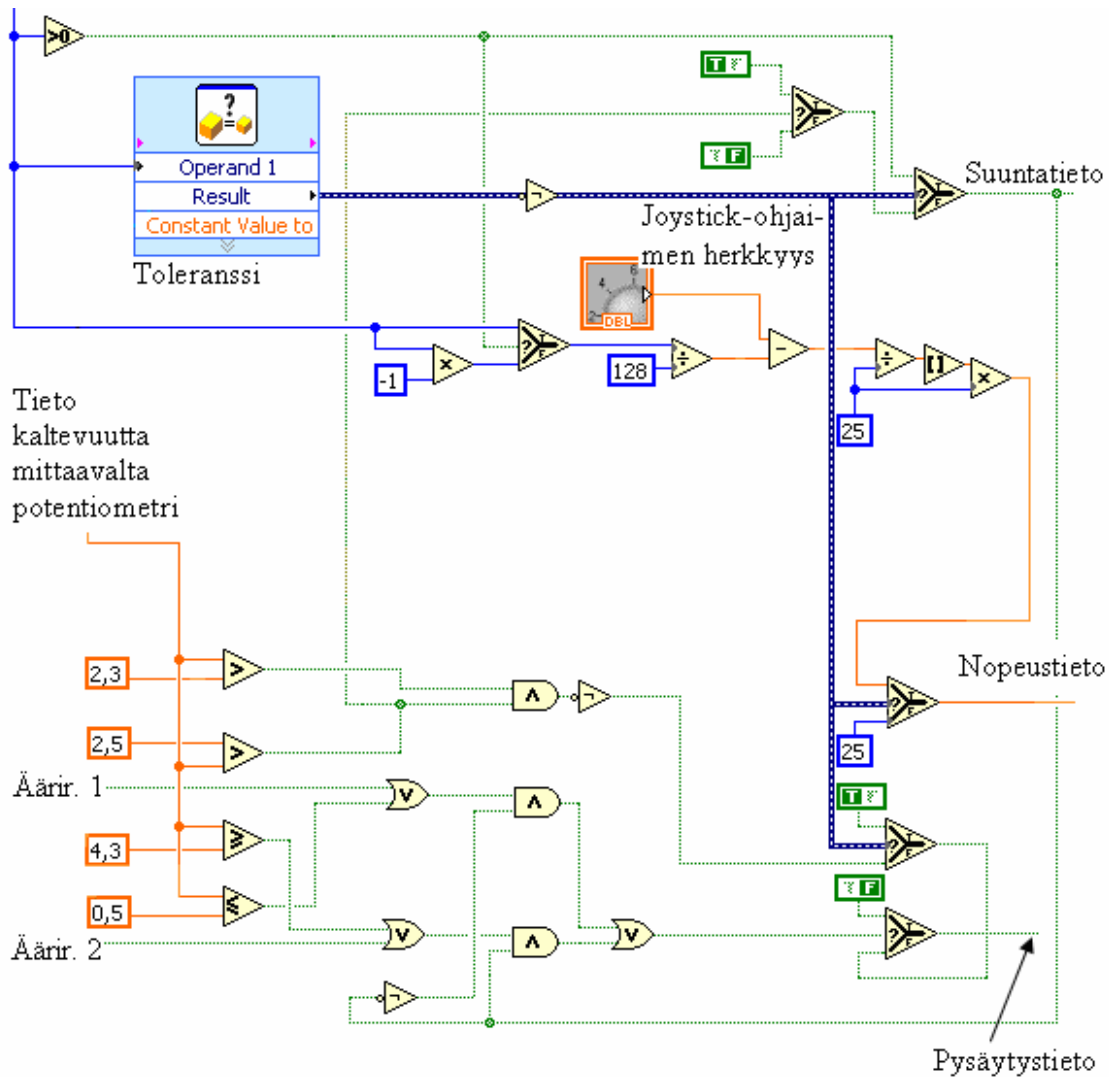
Joystick-ohjaimessa on yleensä sauvaohjain, jolla voidaan säätää potentiometriä arvoja x- ja y-suunnissa, sekä useita painonappeja erilaisia käyttäjiä varten. Nämä tiedot tulevat jakamattomana ryhmätietona aina Unbundle.vi-moduulille saakka. Kyseisellä elementillä ryhmätieto jaetaan yksilötietoihin kuvan 26 esittämällä tavalla.



Kuva 26. Joystick-ohjain tarvitsee omat virtuaali-instrumenttinsa, jotta sen ohjauskäsky saadaan hyödynnettyä ohjelmassa.

Sekä x- että y-suunnille on rakennettu kuvan 27 mukainen ohjaustoiminta. Aluksi ohjauspiiri tutkii, onko Joystick-ohjaimelta tuleva tieto positiivinen vai negatiivinen, eli se selvittää ohjaussauvan kääntösuunnan. Tämä tieto välittyy suuntatietona askelmoottorille.

Joystick tieto



Kuva 27. Joystick-ohjaimen nopeustieto muutetaan positiiviseksi ja kalibroidaan askelmoottorin ohjaukseen soveltuvaksi. Liukupotentiometrin maksimikallistusasetus huolehtii, ettei tasoa kallisteta liikaa. Jos kyseinen kallistusasetus ei toimi, niin lopullisena turvarajana ovat mekaaniset äärirajat, jotka vikatilanteessa pysäyttävät askelmoottorin.

Kun Joystick-ohjaimen kahvasta päästetään irti, niin taso palautuu vaaka-asentoon automaattisesti määrättyllä (säädetävällä) nopeudella. Joystick-ohjain ei kuitenkaan välttämättä mene kyseisessä palautuksessa täysin nollatilaan, vaan se saattaa lähettää edelleen pientä nopeusarvoa prosessiin. Tämä aiheuttaisi korjaamattomassa tilassa, ettei taso pysähtyisikään vaaka-asentoon vaan jatkaisi kallistusta toiseen suuntaa. Kyseisen ongelman torjumiseksi Joystick-ohjaimen syöttämälle arvolle on laitettu pieni toleranssi.

Joystick-ohjaimen herkkyyttä voidaan säätää ns. virtuaalipotentiometrillä. Askelmoottorin ohjaus tapahtuu skaalalla 0 – 255. Nolla vastaa askelmoottorin maksiminopeutta ja 255 hitainta mahdollista nopeutta. Nopeustiedot, suuntatiedot ja pysäytyskäskyt välittyvät askelmoottorin ohjaimelle I2C-väylää pitkin 8-bittisinä jonoina. Ohjaimen prosessorihin ohjelmoitu C-kielinen ohjelma muuntaa bittijonot askelmoottorin ohjaamiseen sopiviksi käskyiksi. Kyseisen potentiometrin avulla voidaan Joystick-ohjaimen alkuarvoa nolla, joka on askelmoottorin suurin mahdollinen nopeus, nostaa alkamaan jostakin suuremmasta luvusta, jolloin nopeusskaalaus olisikin vastaavasti välillä *säätöarvo* - 255.

Käsiohjauksen ohjauskaavio on esitetty kokonaisuudessaan liitteessä 5.

3.2. Automaattiajo

Automaattiajossa tietokoneelle määritellään kohdekoordinaatit $\mathbf{r}^T = (r_x, r_y)$ eli kohdepiste, johon kuulan halutaan tasolla siirtyvän. Kuula etenee kohti tavoitepistettä tietokoneohjatusti. Firewire-kamera seuraa kuulan kulkua tasolla, ja se lähettää kuulan paikkatietoa takaisinkytkentätietona säätimelle 133 ms väliajoin. Säädin määrittelee mihin suuntaan ja kuinka nopeasti tasoja pitää kulloinkin kallistaa, ja ohjaa kuulan tavoitepisteeseen.

Automaattiajossa käytetään kahta eri ajotyyppiä, nimittäin sysäys- ja perusajo. Sysäysajo saa kuulan liikkeelle aloituksessa ja silloin, kun kuula pysähtyy tavoitepisteen ulkopuolelle. Perusajossa PD-säädin ohjaa liikkuvaa kuulaa kohti tavoitepistettä. Se

kuinka edellä mainitut ajot on toteutettu ja mitä niillä tarkoitetaan, käsitellään seuraavissa luvuissa.

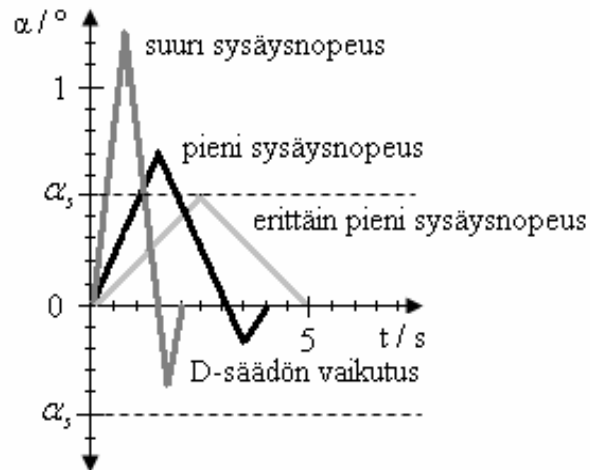
3.2.1. Sysäyskäyttö

Kun kohdekoordinaatit $\mathbf{r}^T = (r_x, r_y)$ on määritelty ja asetetaan ohjaus ajotilaan tai kun kuula pysähtyy ajotilanteessa tavoitepisteen ulkopuolelle, lähettäen saman koordinaattiarvon kahteen kertaan $m_x = m_{xv}$ (m_{xv} = paikannustieto viiveellä), niin säädin aiheuttaa sysäyskäytön saadakseen kuulan liikkeelle. Tällöin taso kallistuu jyrkästi tavoitepisteen $\mathbf{r}^T = (r_x, r_y)$ suuntaan saavuttaen asetellun sysäyskulman α_s (kuva 28). Liukupotentiometriltä tulevan arvon u_p käsittely säätimessä ja edelleen toimintaohjeen siirto askelmoottorille vie kuitenkin tietyn ajan. Tämän vuoksi askelmoottori ei ehdi reagoida heti säätöön, vaan ajaa tason sysäyskulman α_s yli. Se kuinka paljon sysäyskulman α_s yli mennään, riippuu pelkästään aseteltavasta askelmoottorin sysäysnopeudesta v_s kuvan 28 mukaisesti.

Kuula lähtee liikkeelle suhteellisen suurella nopeudella, joten lähtösysäyksen nopea vastakallistus (palautus) tasapainotilaan hidastaa kuulan nopeutta. Tasapainotilan kohdalla ($\alpha = 0^\circ$) ohjaus siirtyy perusajoon, jolloin PD-säätö tulee mukaan ohjaukseen ja joko vahvistaa tai heikentää tasapainotilan yliajtoa, riippuen kuulan nopeudesta ja PD-säätöarvoista (K_P, K_D). Tasapainotilalla tarkoitetaan tilannetta, jossa taso on täysin vaakatasossa ($\alpha = 0^\circ$).

Esimerkiksi kuvassa 28 on suurella sysäysnopeudella ($v_s = 40$) ylitetty sysäyskulma α_s 0,75 asteella ja tasapainotila 0,4 asteella, sysäysajan t ollessa kaksi sekuntia. Kun sysäyskulma α_s on suuri, kuulan nopeus nousee liikkeellelähtötilanteessa suureksi, jolloin D-säätö $K_D \frac{de_1(t)}{dt}$ reagoi voimallisemmin, ja aiheuttaa suhteellisen suuren vastakallistuskulman tasolle, hidastaen kuulan nopeutta. Pienellä nopeudella ($v_s = 200$) sy-

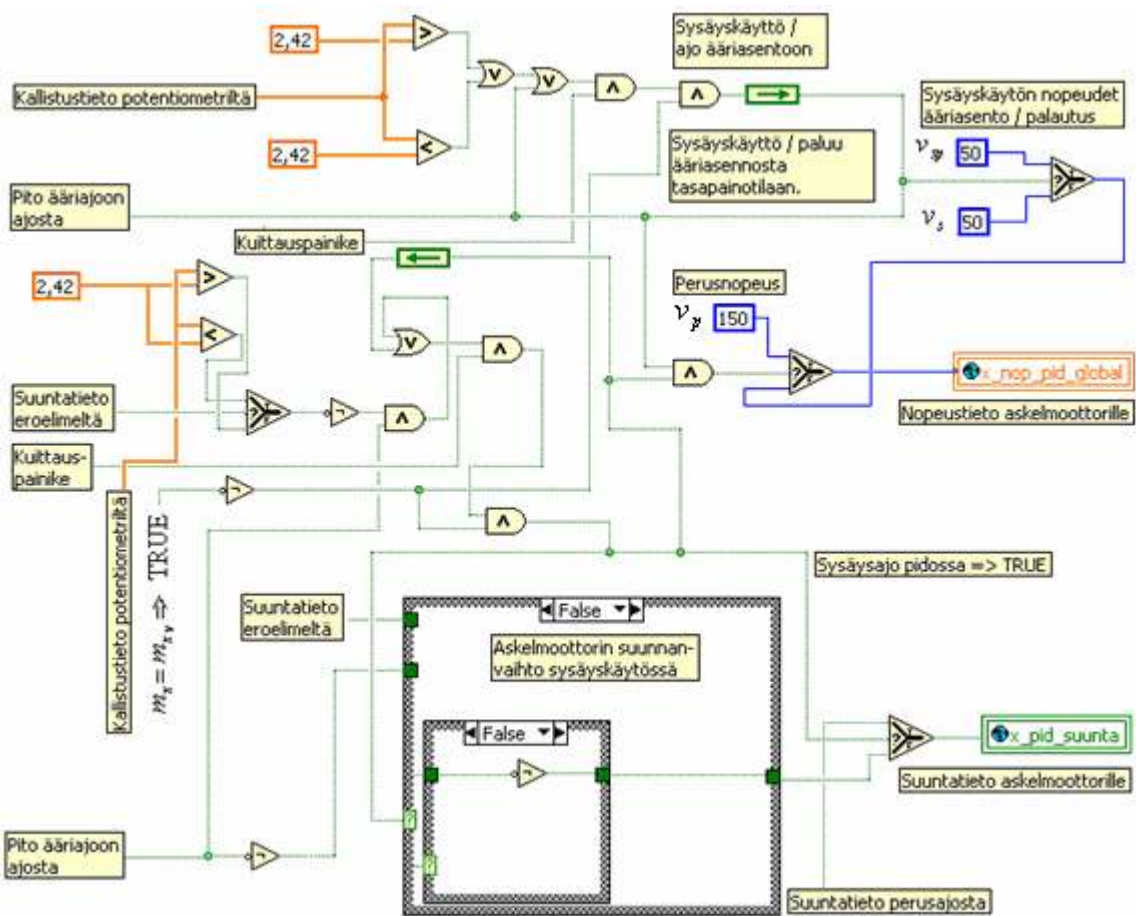
säyskulma α_s on ylitetty 0,2 asteella ja tasapainotila 0,2 asteella, sysäysajan t ollessa neljä sekuntia.



Kuva 28. Sysäysnopeuden v_s vaikutus säädetyn sysäyskulman α_s ylitykseen sysäyskäytön myötäkallistuksessa ja PD-säädön vaikutus tasapainotilan ($\alpha_s = 0^\circ$) ylitykseen sysäyskäytön vastakallistuksessa.

Erittäin pienellä sysäysnopeudella ($v_s = 230$) sysäyskulma α_s ja tasapainotila eivät enää ylity. Kyseiset mittaukset ovat tehty y-akselilla ja ovat suuntaa antavia. Sysäysaika t pitää kutakuinkin paikkansa, mutta virhemarginaali sysäyskulmasta α_s vaihtelee hiukan joka sysäyskerralla. Tämä edellä mainittu sysäyskäyttö suoritetaan sen vuoksi, että kuulan lepokitka ylittyisi ja kuula lähtisi suhteellisen tasaisesti liikkeelle.

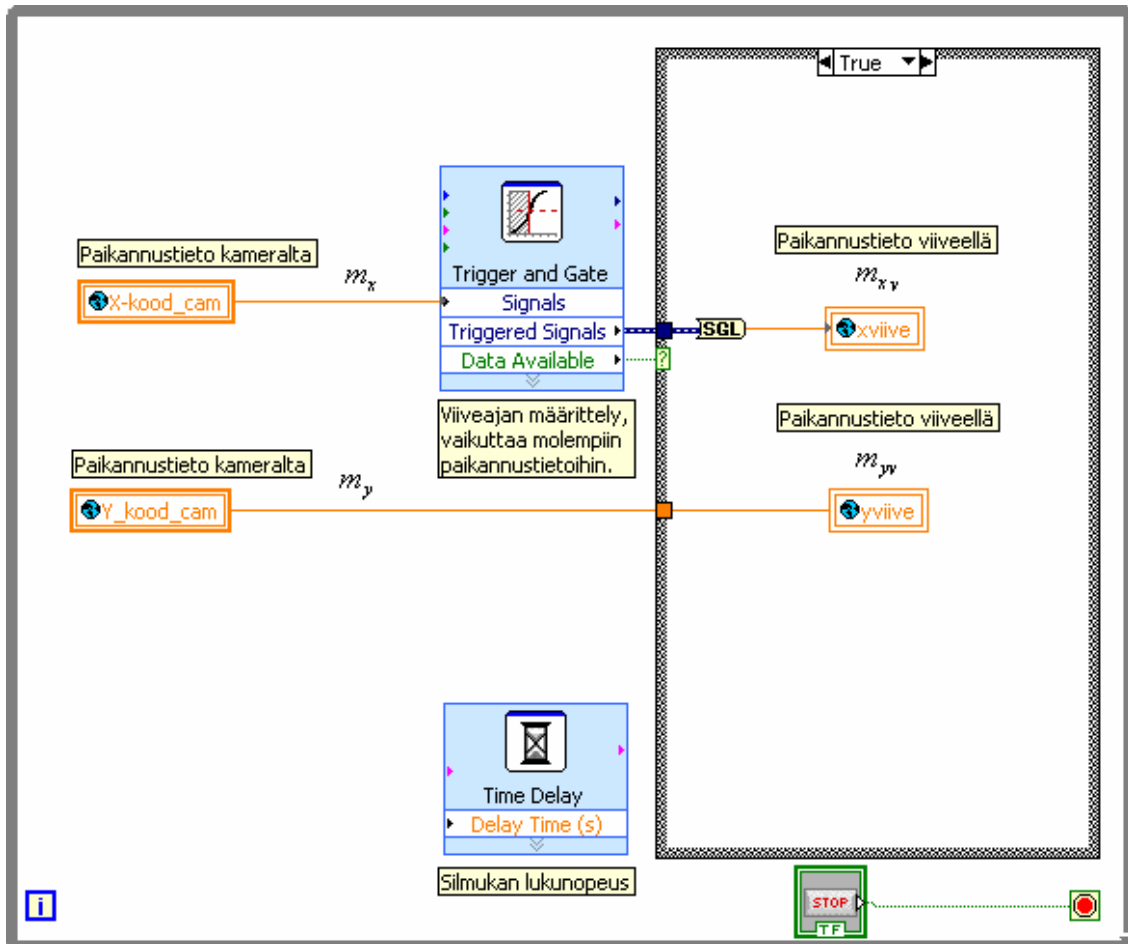
Sysäyskäyttö on toteutettu kuvan 29 mukaisella kytkennällä. Kamera seuraa koko ajan kuulan liikkumista, lähettäen x- ja y-koordinaatteja $\mathbf{m}^T = (m_x, m_y)$ takaisinkytkentätietona säätöjärjestelmään tietyin väliajoin.



Kuva 29. Lähtötilanteessa tai jos kuula pysähtyy ajotilanteessa tavoitepisteen $\mathbf{r}^T = (r_x, r_y)$ ulkopuolelle, sysäyskäyttö saa sen liikkeelle. Sysäyskäyttö on toteutettu kuvan mukaisella kytkennällä.

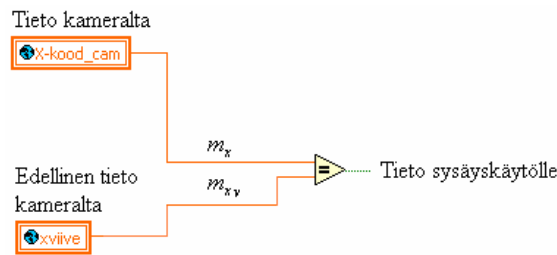
Kun ohjaus asetetaan ajotilaan, sysäyskäyttö tapahtuu automaattisesti, ja se saa kuulan liikkeelle. Kun kuula liikkuu, sysäystoiminta pysyy lukkiutuneena pitoon eli se ei vaikuta ohjaukseen. Jos kuula pysähtyy ja lähettää samat koordinaatit kaksi kertaa peräkkäin ($m_x = m_{xv}$), pito purkaantuu ja edellä mainittu sysäystoiminta toistuu ja saa kuulan taas liikkeelle. Jos kuula saavuttaa tavoitepisteen \mathbf{r} kesken sysäyskäytön, niin askelmoottorit ajavat tasot ensin tasapainotilaan ja vasta sitten säädin lähettää askelmoottoreille pysäytyskäskyn (kuva 34) ja toiminta pysähtyy.

Säädin säilyttää edellisen koordinaatin ja vertaa sitä uuteen koordinaattiin kuvan 30 mukaisella ohjauksella. Pidon saa purettua myös käyttöliittymässä sijaitsevasta kuitauspainikkeesta kesken ajon.



Kuva 30. Kameralta tulevista paikannustiedoista (m_x, m_y) otetaan näytteitä (m_{xv}, m_{yv}) yhden sekunnin välein.

Kuvassa 30 olevalla kytkennällä otetaan kameralta tulevista paikannustiedoista (m_x, m_y) näytteitä (m_{xv}, m_{yv}) yhden sekunnin välein ja verrataan niitä kuvan 31 mukaisella kytkennällä toisiinsa. Jos kyseiset arvot ovat samat ($m_x = m_{xv}$), niin kuula on silloin pysähtynyt. Jos tavoitepistettä ei ole kyseisessä tilanteessa vielä saavutettu, niin säädin toteuttaa säsäysajon tavoitepisteen suuntaan, saadakseen kuulan taas liikkeelle.

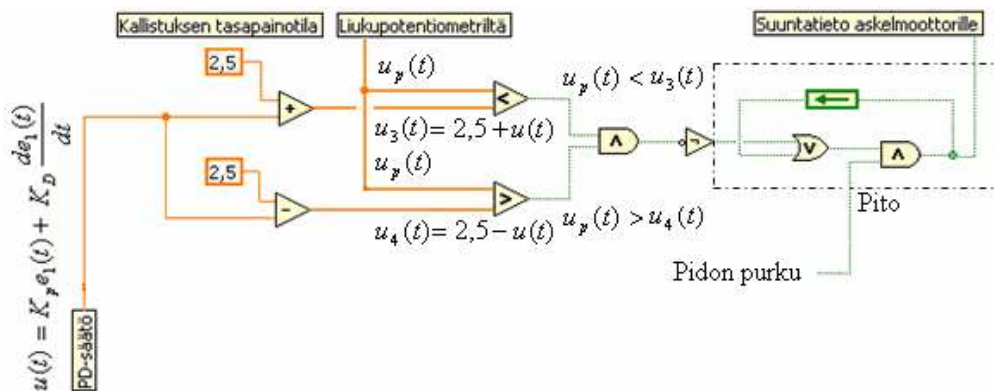


Kuva 31. Jos kuula pysähtyy tavoitepisteen ulkopuolelle ja lähettää samat paikannustiedot ($m_x = m_{xv}$) kaksi kertaa peräkkäin, lähtee tieto sysäyskäytölle.

3.2.2. Perusajo

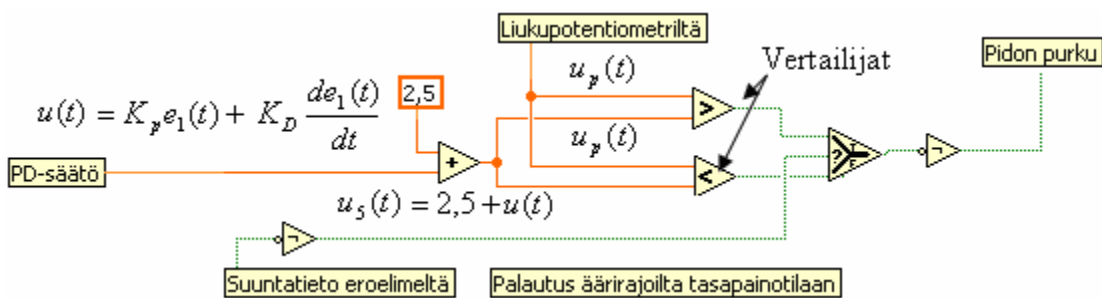
Kun sysäysajo on jäänyt pitoon, järjestelmä ohjautuu perusajolle. Perusajossa tasot kallistuvat kolmannella säädettävällä nopeudella $\mathbf{v}_p^T = (v_{px}, v_{py})$ (kuva 29) ja tasojen kallistuskulmat (α_x, α_y) ohjautuvat PD-säädöllä.

Perusajossa erosuure e kerrottuna vahvistuksella K_p määrittelee ensisijaisesti tasojen kallistuskulmien suuruudet, joita D-termi redusoi suuntaan tai toiseen tilanteesta riippuen (kuva 32). PD-säädön toiminta on esitetty tarkemmin luvussa 3.2.3.



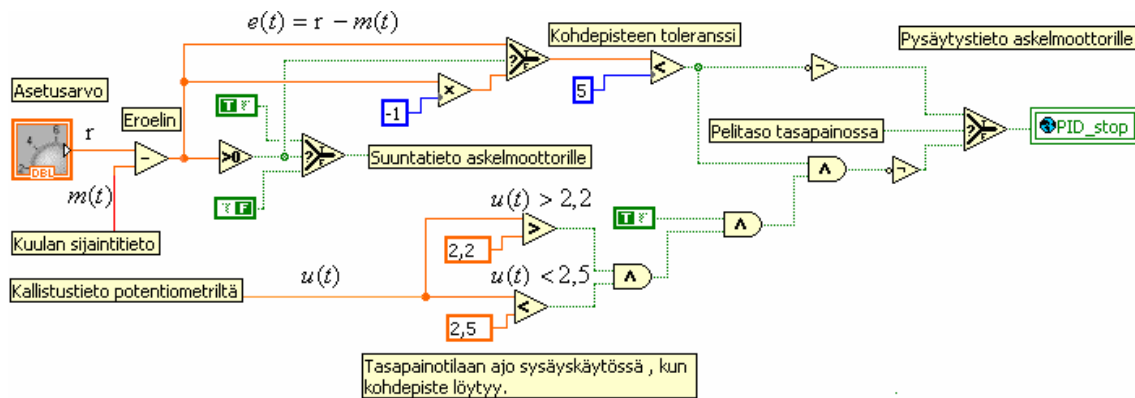
Kuva 32. Perusajon ohjausperiaate. PD-säädöltä tuleva positiivinen tai negatiivinen lähtösignaaliarvo $u(t)$ lisätään tasapainotila-arvoon $u_3(t) = 2,5 + u(t)$ ja vähennetään tasapainotila-arvosta $u_4(t) = 2,5 - u(t)$. Saatuja arvoja verrataan liukupotentiometriltä tulevaan jännitearvoon $u_p < u_3$ ja $u_p > u_4$.

Jos jompikumpi sallituista sen hetkisistä ääriarjoista ylittyy, vertailijalta lähtee FALSE-käskey, tällöin TAI-portti saa invertterin kautta TRUE-käskyn ja asettaa kyseisen ohjauksen pitoon, jolloin tasapainotilaohjaus (kuva 33) palauttaa tason tasapainotilaan tai sen lähelle (yli tai ali), riippuen kuulan nopeudesta ja D-parametristä. Kuulan nopeus nousee joskus niin suureksi, että säätö ei D-säädöstä huolimatta ehdi ohjata askelmootoria, vaan ääriarja ylittyy. Tällöin tasapainotilaan ajo rauhoittaa tilannetta.



Kuva 33. PD-säädöltä tuleva positiivinen tai negatiivinen lähtösignaaliarvo $u(t)$ lisätään tasapainotila-arvoon. Mitä lähempänä tasapainotilaa se on, sitä pienemmäksi lähtösignaaliarvo $u(t)$ muuttuu, riippuen myös D-säädön reagoinnista (ks. PD-säätö). Se kumpi vertailijoista on käytössä, riippuu siitä, kumpi ääriarjoista on ylitetty. Kun haluttu raja on saavutettu, pito purkautuu ja ajo poistuu tasapainotila-ajosta.

Kun kuula saavuttaa tavoitepisteen toleranssin rajoissa, säädin lähettää pysäytyskäskyn askelmootorille kuvan 34 mukaisella kytkennällä.



Kuva 34. Kun kuula saavuttaa tavoitepisteen $\mathbf{r}^T = (r_x, r_y)$ toleranssin rajoissa, säädin lähettää askelmoottorille pysäytyskäskyn.

Kuvassa 34 asetusarvoa r verrataan mitta-arvoon $m(t)$. Saatua eroa $e(t)$ verrataan toleranssiarvoon ($5 = 3,7$ cm). Jos ero $e(t)$ on pienempi kuin toleranssiarvo ja taso on tasapainotilassa, niin askelmoottorille lähtee pysäytyskäsky.

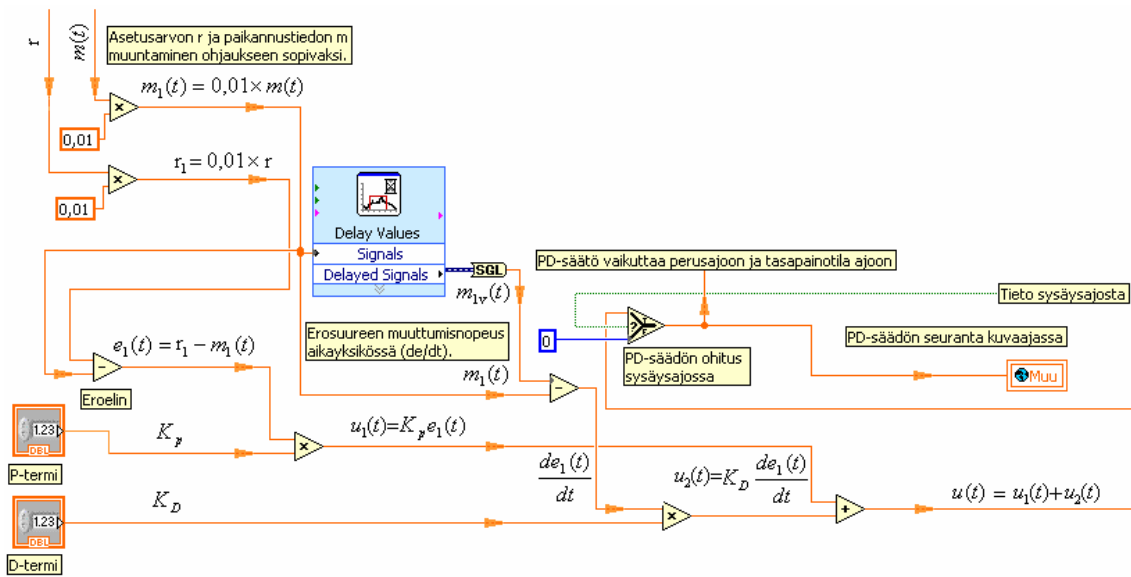
3.2.3. PD-säätö

Asetusarvo \mathbf{r} säädetään virtuaalipotentiometrillä, ja mitta-arvo $\mathbf{m}(t)$ saadaan takaisin-kytkennän kautta firewire-kameralta ja konenäkömoduulista. Kameran tarkkuus on 160×120 pistettä, ja se näyttää myös pelin reunat. Itse pelitason kooksi muodostuu 100×100 pistettä, joten ero $e(t)$ vaihtelee välillä ± 100 pistettä (ääriarvot). PD-säädön ohjaussuureen täytyy olla välillä $0 - 1$, minkä vuoksi mitta-arvo $\mathbf{m}(t)$ ja asetusarvo \mathbf{r} kerrotaan $0,01$:llä seuraavasti:

$$m_1(t) = 0,01 \times m(t) \quad (11)$$

$$r_1 = 0,01 \times r \quad (12)$$

Tämän jälkeen kyseiset arvot soveltuvat PD-säädön käyttöön. PD-säädön toiminta on esitetty kuvassa 35.



Kuva 35. PD-säädön toteutus kuulapelissä. PD-säätimen derivoiva termi toimii ennustajana, sillä se ottaa huomioon erosuuren muuttumisnopeuden. Jos positiivinen erosuure on kasvamassa, se vahvistaa P-termin vaikutusta, jos taas pienenemässä, se heikentää vaikutusta.

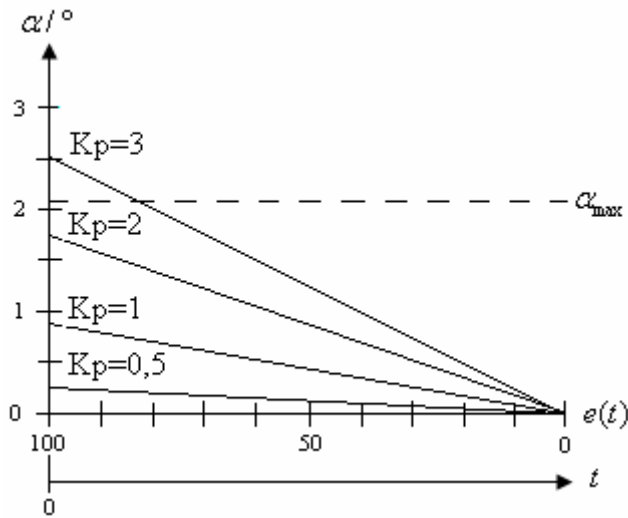
Mittausarvo $m_1(t)$ ja asetusrvo r_1 käsitellään eroelimessä seuraavasti:

$$e_1(t) = r_1 - m(t) \quad (13)$$

Eli mitä kauempana kuula on tavoitepisteestä, niin sitä suurempi on erosuure $e_1(t)$ ($e_{\max} = 1$). Tämän jälkeen erosuure $e_1(t)$ kerrotaan aseteltavalla vahvistusarvolla K_p seuraavasti:

$$u_1(t) = K_p e_1(t) \quad (14)$$

Mitä suurempi vahvistusarvo valitaan, sitä jyrkemmin taso kallistuu, silloin kun kuula on maksimietäisyydellä tavoitepisteestä (kuva 36). Vahvistuksen arvolla $K_p = 3$ kallisutus menee ääritilanteessa yli sallitun kulma-arvon α_{\max} .



Kuva 36. Vahvistuksen vaikutus kallistuskulmiin erisuuruisilla $e(t)$:n arvoilla, D-termin vaikutusta ei kyseisessä kuvassa ole huomioitu.

Erosuureen muuttumisnopeutta aikayksikössä seurataan viivästämällä paikannustietoa $m_1(t)$ ja vähentämällä viivästytyksestä paikannustiedosta $m_{1v}(t)$ alkuperäinen arvo seuraavasti:

$$\frac{de_1(t)}{dt} = m_{1v}(t) - m_1(t) \quad (15)$$

Näin saatu arvo kerrotaan vielä aseteltavalla derivointikerroimella K_D seuraavasti:

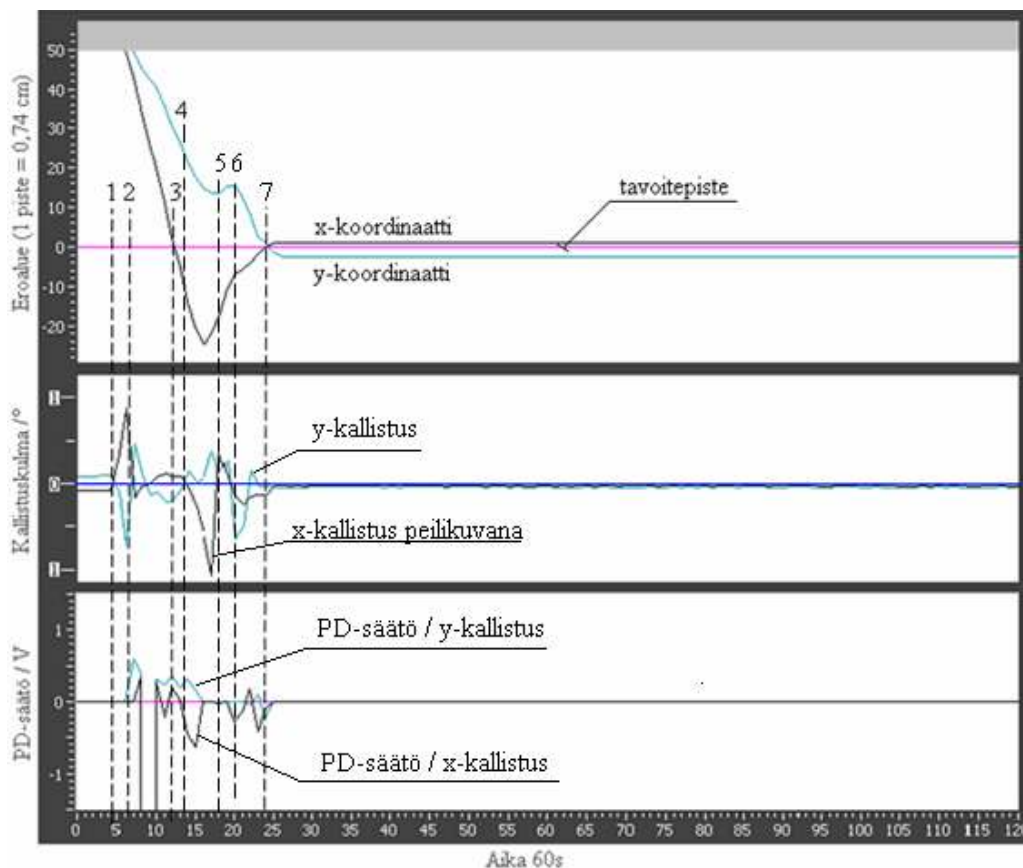
$$u_2(t) = K_D \frac{de_1(t)}{dt} \quad (16)$$

Jos derivointikerroin K_D on nolla, ainoastaan vahvistus on mukana säädössä. Jos derivointikerroin K_D valitaan liian suureksi, niin kuulan liikkeessa hiukankin nopeammin säätö ajaa tasot ääriarjoille ja toiminta pysähtyy.

Lopullinen kaltevuusohjauksen lähtösignaali $u(t)$ muodostuu näistä kahdesta edellä lasketusta P- ja D-kertoimesta seuraavasti:

$$u(t) = u_1(t) + u_2(t) \quad (17)$$

Jos kuulan nopeus $v_k(t)$ on suuri eli positiivinen erosuure on kasvamassa, D-termi vahvistaa P-termin vaikutusta, jos taas pienenemässä, se heikentää vaikutusta.



Kuva 37. Kallistuskulman (keskimmäinen kuvaaja) ja PD-säädön (alin kuvaaja) vaikutus kuulan liikkeeseen (ylin kuvaaja) kullakin ajan hetkellä t .

Kuvassa 37 on esitetty kuulapelin säädön idea kuvaajan avulla. Aluksi seurataan kuulan etenemistä y-akselilla. Kun asetusarvo on valittu ja järjestelmä asetetaan ajotilaan, taso kallistuu tavoitepisteen suuntaan ja suorittaa sysäyskäytön (1). Kuula lähtee vierimään suhteellisen suurella nopeudella, mutta lähtösysäyksen nopea palautus tasapainotilan yli hidastaa kuulan nopeutta (2). Säädin reagoi heti tasapainotilaan ja ohjaa perusajon pääl-

le, mutta askelmoottori ajaa tason jonkin verran tasapainotilan yli. Tasapainotilan ylityksen suuruus riippuu kuulan nopeudesta D-säädön mukaisesti.

Sysäyskäyttö jää pitoon, ja taso ohjautuu perusohjauksella. Alkusysäyksen (1) jälkeen kuula ei ole vielä saavuttanut tavoitepistettä, joten vastakallistus vaihtuu pieneksi P-säädöstä johtuvaksi myötäkallistukseksi, jota D-säätö rajoittaa. Kuula lähestyy tavoitepistettä suhteellisen suurella nopeudella, jolloin D-säätö vaimentaa hiukan kuulan liikettä vaihtamalla tason kallistussuuntaa (3). Kuulan nopeus hidastuu edelleen D-säädön vaikutuksesta (4). D-säädön aseteltava K_D -vahvistusarvo ei ole aivan kohdallaan, joten kuula liikkuu hetken tavoitepisteestä pois päin. Säättöpiiri reagoi välittömästi muutokseen, mikä aiheuttaa myötäkallistuksen (6). Se saa kuulan taas etenemään kohti tavoitepistettä, saavuttamaan tavoitepisteen toleranssin rajoissa (7) ja pysähtymään siihen.

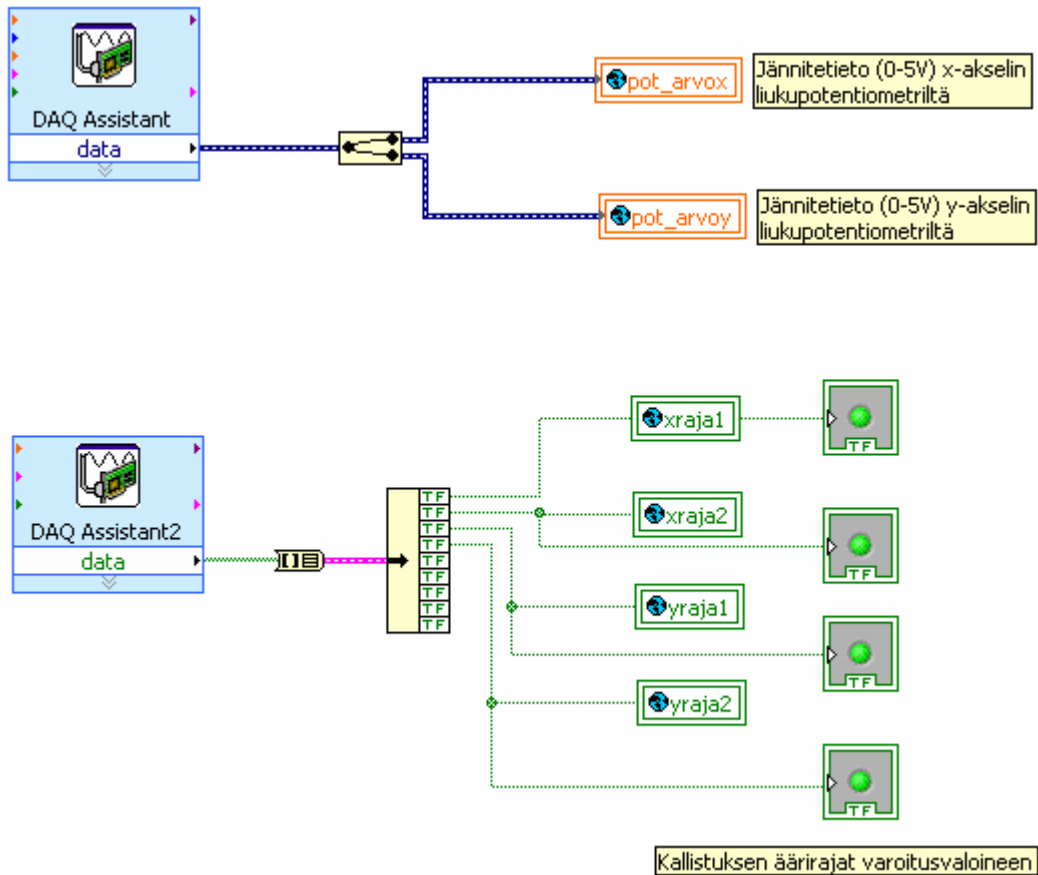
X-akselilla tapahtuu vastaavanlainen sysäyskäyttö, kuten y-akselillakin (1). Sysäyskäytön jälkeinen vastakallistus jää suhteellisen pieneksi D-termin reagoiessa myötäkallistuksen puolesta (2). D-säädön aseteltava K_D -vahvistusarvo ei ole x-kallistuksen säädössä aivan kohdallaan. Kuulan vauhti nousee suureksi, joten D-termi reagoi liiankin nopeasti, mikä aiheuttaa suuren vastakallistuspyynnön. Askelmoottori ei ehdi täysin reagoida kyseiseen pyyntöön, vaan aloittaa vastakallistuksen vasta kuulan ylittäessä tasapainotilan. Tällöin D-säätö reagoi kuulan vauhtiin vahvistamalla P-säätöä, ja se aiheuttaa suuren vastakallistuksen (4). Kuulan suunta vaihtuu ja se etenee jälleen kohti tavoitepistettä. Taso kallistuu kohti tasapainotilaa ja antaa kallistussysäyksen vastakkaiselle puolelle (5). D-säätö reagoi vaimentamalla P-säädön vaikutusta, mikä hidastaa kuulan kulkua tavoitepisteen tuntumassa (6), joten kuula saavuttaa tavoitepisteen (7) ja pysähtyy.

3.3. Rajat

Rajat on asennettu kuvan 14 mukaisesti kohtiin pelikehikossa. Rajojen kytkennät on suoritettu liitteen 2 mukaisesti kytkentäalustalle CB-68LPR. Sieltä rajatiedot kulkeutuvat

SCH-68-EPM SHIELDED-välikaapelia pitkin tietokoneessa sijaitsevaan tiedonkeruukorttiin NIPCI-6221.

Tiedonkeruukortilta tiedot saadaan ohjelman käyttöön tiedonkeruusovellusten toteuttamiseen tarkoitetun DAQ-assistant-apuohjelman avulla.



Kuva 38. Pelin tasojen ohjauksissa käytetyt rajatiedot ovat mekaanisilta rajoilta tulevat digitaaliset 0/1-tiedot ja liukupotentiometriltä tulevat analogiset 0-5 V jännitetiedot. Nämä tiedot saadaan DAQ-assistant-apuohjelman avulla pelin ohjauksen käyttöön, valitsemalla Block Diagram -ikkunaan tätä toimintaa varten tarkoitetun virtuaali-instrumentin (DAQ assistant express vi).

Rajatiedot siirtyvät ohjelman käyttöön globaalien muuttujien välityksellä (kuva 38). Globaali muuttuja on funktion ulkopuolella esitelty muuttuja, jonka arvo on käytettävissä ja muunneltavissa ilman rajoituksia.

Kallistuskulmia seurataan x- ja y-akseleihin yhdistetyillä potentiometreillä. Tiedot potentiometreiltä tulevat jännitetietoina 0 – 5 V. Eli tasojen tasapainopiste sijaitsee jännitearvona ilmaistuna noin 2,5 V kohdalla ja ääriarvot (max. kallistukset) noin 0 V ja 5 V kohdilla.

Turvarajat ovat mekaanisia kytkimiä, jotka toimivat ns. ääriarjoina potentiometriä petäessä. Tiedot turvarajoilta tulevat 0-1-tietoina.

3.4. Valintakytkin

Askelmoottorien tiedot vastaanotetaan kolmiportaisessa valintaikkunassa (Case Structure). Se, mikä toiminta (ikkuna) on milloinkin käytössä, valitaan käyttöliittymän valintakytkimellä.

Valintakytkimen demo-asennon ohjaamaksi voidaan rakentaa joku oma sovellus, esimerkiksi rataprofiili, jota pitkin kuulan halutaan kulkevan. Automaatioasennossa toimii automaattiajo (kappale 3.1) ja käsiasennossa manuaaliajo (kappale 3.2). Valittaessa valintakytkimen asennoksi 2 askelmoottorien ohjaukset tapahtuvat säätimen kautta. Tällöin kolmiportaisen valintaikkunan ikkuna numero kaksi välittää tiedot säätimeltä askelmoottorien ohjausparametreille globaalien muuttujien välityksellä. Sisempi silmukka (for-loop) määrittelee, kuinka usein kyseinen ohjaus käydään lukemassa. For-silmukan yläosassa on kello, joka määrittelee lukuajan, esim. 10 tarkoittaa sitä, että toiminta käydään läpi kerran 10 ms:ssa.

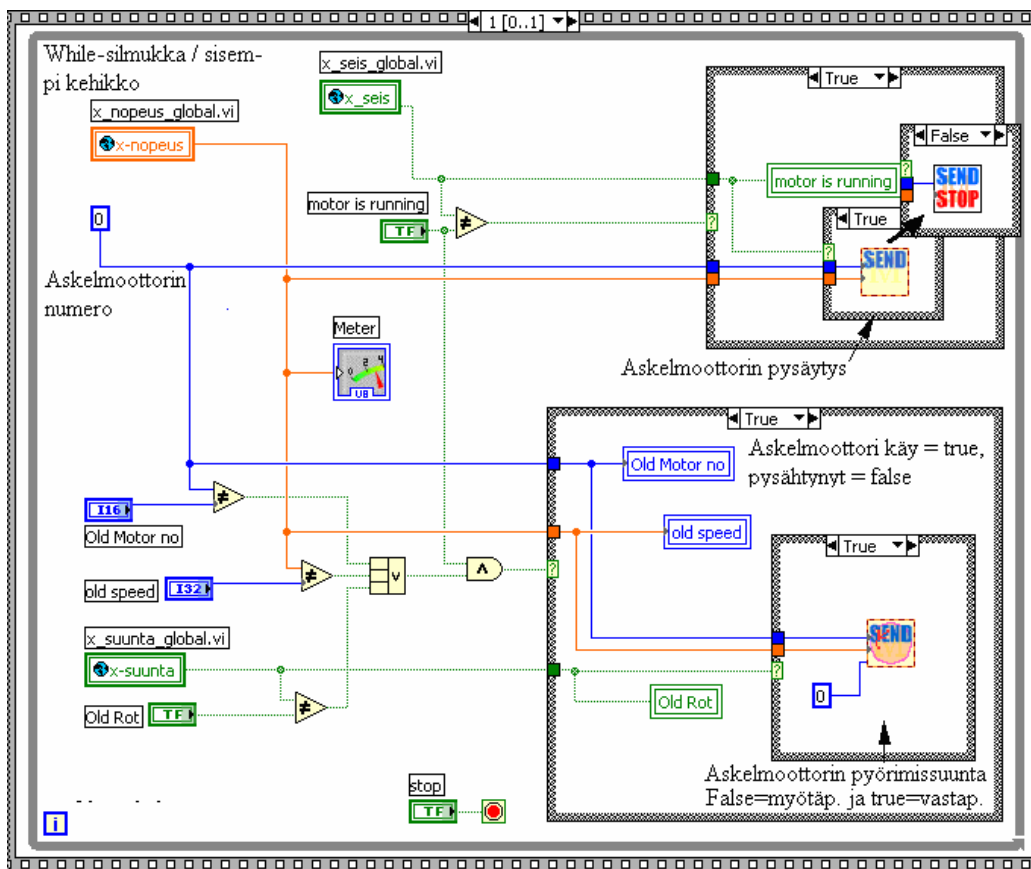
Manuaaliajossa valintakytkin on asennossa 3. Tällöin Joystick-ohjaimelta lähetetyt ohjauspyynnöt välittyvät valintaikkunan kolme kautta askelmoottoreille globaalien muuttujien välityksellä. Automaattiohjauksen ohjauskaavio on esitetty liitteessä 6.

3.5. Askelmoottoreiden ohjaus

Edellä mainitut toimenpiteet eivät saa vielä askelmoottoreita liikkumaan, vaan niille tarvitaan vielä erilliset ohjaimet. Kun askelmoottorille tulee ensimmäisen kerran tietoja,

ohjelma käy lukemassa askelmoottorin alustustiedot, kuten ohjainkortin numeron, rinnakkaisportin numeron, eli paikan mihin I2C-väylä on liitetty, sekä I2C-väylän määritelyt. Alustustiedot luetaan yhden ainoan kerran käynnistyksen yhteydessä.

Alustuksen jälkeen ohjelma siirtyy lukemaan ikkunaa 1 (kuva 39), missä sijaitsevat askelmoottorin käyttöön tarvittavat ohjaustiedot.

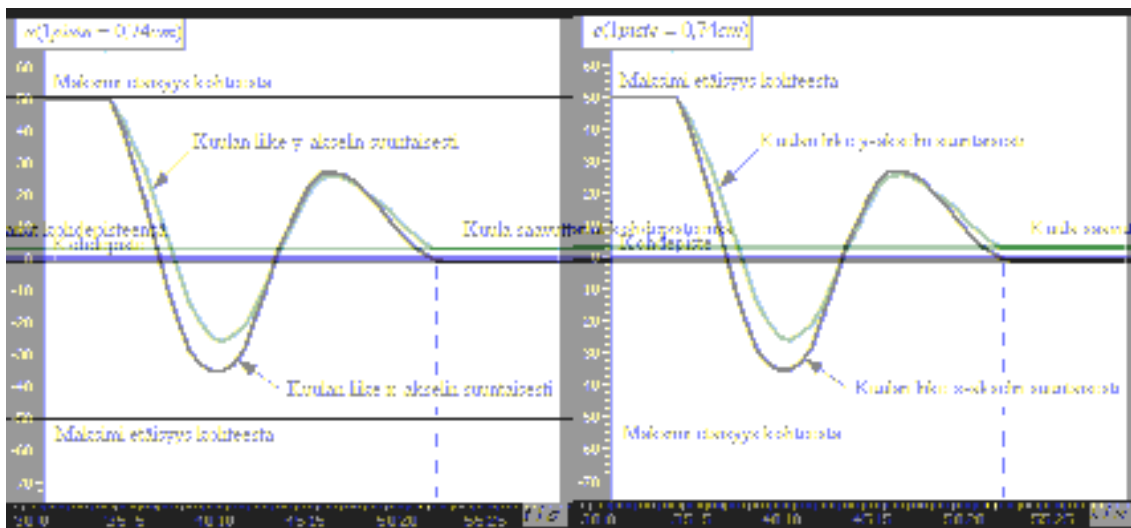


Kuva 39. Askelmoottorien ohjauskäskyt kulkeutuvat kyseiseen tarkoitukseen rakennettuihin ohjelmamoduuleihin eli virtuaali-instrumentteihin. Kyseinen moduuli on rakennettu päällekkäin oleviin ikkunoihin (Stacked Sequence Structure), joilla on peräkkäinen suoritusjärjestys.

Kyseiset ohjaukset on sijoitettu while-silmukkaan, joten ohjausviestit kulkeutuvat sinne niin kauan kun silmukan asetusarvo on true.

Niin kauan kun säätimeltä tai Joystick-ohjaimelta ei tule stop-käskyä (false), askelmoottori pysyy käynnissä. Askelmoottorin pyörimissuuntaa ohjataan true- ja false-käskyillä. Rakennettu ohjelma vertaa nykyistä nopeus-, suunta- ja moottorinumerotietoa edellisiin tietoihin. Jos joku näistä tiedoista on muuttunut, se välittyy I2C-väylää pitkin askelmoottorin ohjainkortille ja siitä edelleen askelmoottorille.

Kuulan liikettä tasolla voidaan seurata mm. kuvan 40 mukaisesta kuvaajasta. Kyseisestä kuvasta selviää kohdepiste ja kuulan eteneminen kohti kohdepistettä sekä x- että y-suunnissa.



Kuva 40. Kuulan liikkeen esittäminen kuvaajalla. Maksimietäisyys kertoo kuulan etäisyyden kohdepisteestä lähtötilanteessa sekä x- että y-akselien suhteen (kuvassa $50 \times 0,74 \text{ cm} = 37 \text{ cm}$ ja aikajana kertoo muun muassa, koska kuula tavoittaa kohdepisteen (kuvassa 23 s).

4. TULOKSIA

Kuulapeli vaatii toimiakseen muutakin kuin hyvin suunnitellun mekaniikan, toimivan säätöjärjestelmän ja oikeanlaisen pelivälineen, nimittäin tarkat säätöparametrit, ja tarkan paikannustiedon. Tässä luvussa haetaan säätimelle parhaat mahdolliset säätöparametrit kokeellisesti. Ismo Tupamäen diplomityössä keskitytään lähinnä kuulan paikannuksen konenäkö toteutukseen.

4.1. Nopeuden säätö

Tasoa kallistavien askelmoottoreiden nopeutta voidaan säätää sekä perusajossa v_p että sysäysajossa. Lisäksi sysäysajossa voidaan säätää erikseen sekä sysäysnopeutta v_s että sysäyksen palautusnopeutta v_{sp} . Nopeutta säädetään numeroarvoin, välillä 0 – 255, missä 0 tarkoittaa maksiminopeutta v_{\max} ja 255 tarkoittaa miniminopeutta v_{\min} . (Nopeus-tiedot, suuntatiedot ja pysäytyskäskyt välittyvät askelmoottorien ohjaimille I2C-väylää pitkin 8-bittisinä jonoina). Kyseisillä nopeuksilla on suuri merkitys säädön onnistumiselle eli sille, kuinka nopeasti ja tasaisesti kuula saavuttaa kohdepisteensä $\mathbf{r}^T = (r_x, r_y)$ vai saavuttaako se sitä ollenkaan.

Nopeussäädön lisäksi kohdepisteiden löytymistä edesauttaa kallistuskulman säätöön perusajossa vaikuttava PD-säätö.

Aluksi tutkitaan nopeuden vaikutusta ohjaukseen. Säädetään K_p - ja K_D -parametrit suhteellisen hyvin toimiviin arvoihin: x-suunnan säätö, $K_p = 0,7$ ja $K_D = 20.000.000$, ja y-suunnan säätö, $K_p = 0,4$ ja $K_D = 25.000.000$. Nämä kyseiset parametrit määriteltiin kokeellisesti tietyllä askelmoottorinopeudella (v_p, v_s, v_{sp}) ja todettiin suhteellisen hyvin toimiviksi. K_p ja K_D -parametrien tarkasteluun palataan nopeusarvojen tutkimisen jälkeen. Tällöin nopeuden ja PD-säädön yhteisvaikutus saadaan viritettyä tarkemmaksi.

Kuulan sijainti lähtötilanteessa on pelitason vasen alareuna, koordinaatteinaan (33, 10). Kohdearvoksi määriteltiin (84, 60) eli suurin piirtein laudan keskikohta. Nopeusarvojen asettamisen jälkeen kuulan kulkua kohti kohdekoordinaattia seurattiin yhden minuutin ajan. Kuulan sijaintipisteet $\mathbf{m}^T = (m_x, m_y)$ päivittyivät x-y-kuvaajaan 0,5 sekunnin välein.

4.1.1. Kuvaajan tulkitseminen

Kuvaajat on esitetty allekkain samalla aikajanalla (kuva 8.1). Ylimmäisessä kuvaajassa on esitettyä kohdepiste kohdassa $e = 0$ (harmaa viiva). Kuulan etenemistä kohti tavoitustepistettä $\mathbf{r}^T = (r_x, r_y)$ x-akselilla kuvaa musta viiva $e_x(t)$ ja y-akselilla harmaa viiva $e_y(t)$. Keskimmäisessä kuvaajassa on esitetty tasojen tasapainopiste ($\alpha = 0^\circ$) harmaalla viivalla. X-tason kallistuksia tasapainotilan molemmin puolin kuvaa musta viiva $\alpha_x(t)$ ja y-tason kallistuksia harmaa viiva $\alpha_y(t)$. Alimmaisessa kuvaajassa on esitetty PD-säädön vaikutus säätöön. Silloin kun säätö ei vaikuta, $u = 0\text{ V}$ (harmaa viiva). PD-säädön vaikutusta x-tason kallistuksessa $\alpha_x(t)$ kuvataan mustalla viivalla $u_x(t)$ ja y-tason kallistuksessa $\alpha_y(t)$ harmaalla viivalla $u_y(t)$.

4.1.2. Mittaukset ja kommentit

Mittauksia suoritettiin useilla nopeusarvoilla hakien säädön kannalta parasta nopeutta. Osa saaduista tuloksista, lähinnä ääriarvot ja parhaat arvot, on esitetty kuvissa 41 – 45. Ensimmäinen mittaus suoritettiin suurilla nopeuksilla, ja etsittiin maksimaalisia äärirajoja nopeuden suhteen. Sekä x, että y-akseleille asetettiin seuraavanlaiset nopeudet: $v_p = 30$, $v_s = 30$ ja $v_{sp} = 30$.

Perusnopeudella v_p tarkoitetaan sitä nopeutta, millä säädin ohjaa askelmoottoreita (tasokaltevuuksia α_x ja α_y), kun kuula jo liikkuu. Sysäysajo sisältää syvän kallistuksen (liikkeellelähtösysäys v_s) ja palautuksen tasapainotilaan (sysäyksen palautus v_{sp}). Näille molemmille voidaan asettaa omat nopeutensa.

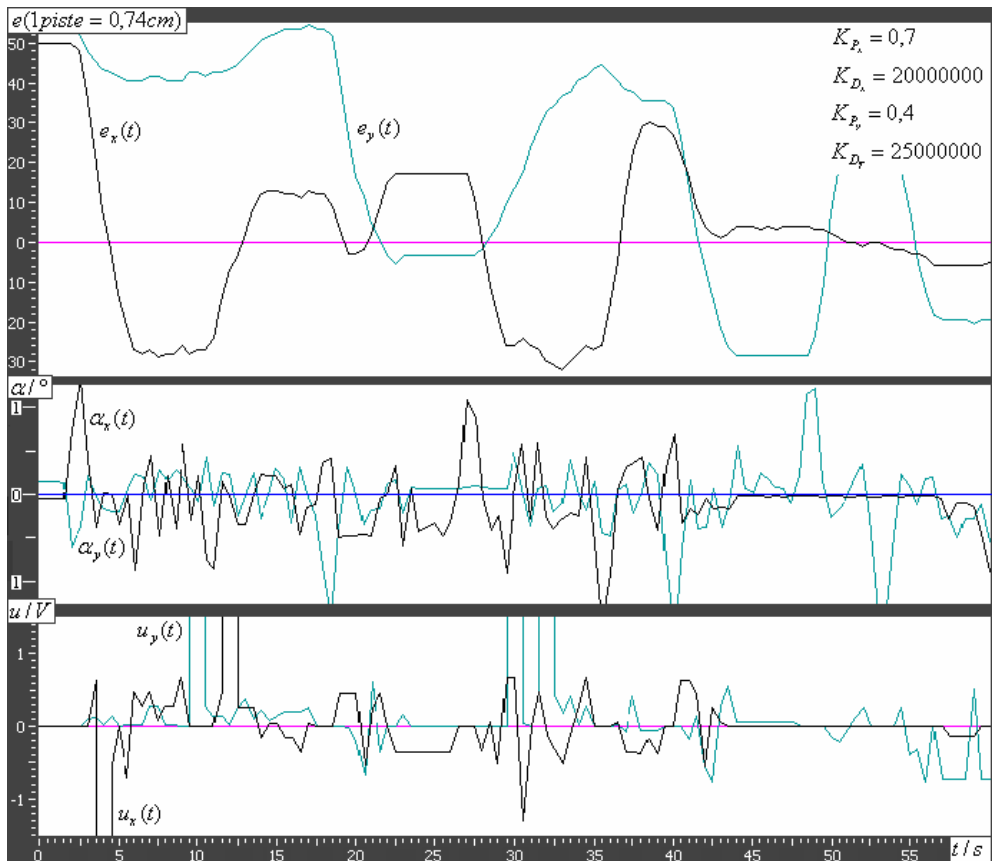
Suoritettiin kaksi ajoa, joista molemmat kokeilut kaatoivat koko säätöjärjestelmän siinä vaiheessa, kun ohjaus siirtyi perusajolle, joten lisäkokeilut todettiin turhaksi.

Seuraava mittaus suoritettiin edelleen suurilla nopeuksilla. Sekä x- että y-akseleille aseteltiin nopeudet: $v_p = 40$, $v_s = 40$ ja $v_{sp} = 40$.

Suoritettiin kolme ajoa, joista yksikään ei löytänyt tietään perille asti, ja yksi ajo kaatoi koko säätöjärjestelmän. Ilmaus ”perille asti” tarkoittaa sitä, että kuula saavuttaa asetellun kohdekoordinaatin $\mathbf{r}^T = (r_x, r_y)$ toleranssin rajoissa, jolloin toiminta pysähtyy ja kuula pysyy määritellyssä kohteessa liikkumattomana.

Jo ensimmäinen ajo näytti, että tason ja kuulan epätasaisuudet, sekä askelmoottorien hitaus, tuovat ongelmia säätöön. Kuulan pysähtyessä se saattaa kiinnittyä tasopintaan niin tiukasti, että edes sysäyskäyttö ei saa sitä heti liikkeelle. Lisäksi tason kuperuus vaikeuttaa kuulan pysähtymistä tietyissä kohdin tasoa.

Testiajoesimerkeissä seurataan tapahtumia aika-janan suhteen. Esimerkiksi jos sysäyskäyttö on tapahtunut 5s päästä siitä, kun automaattiajo on alkanut, sitä merkitään ($t = 5$ s).

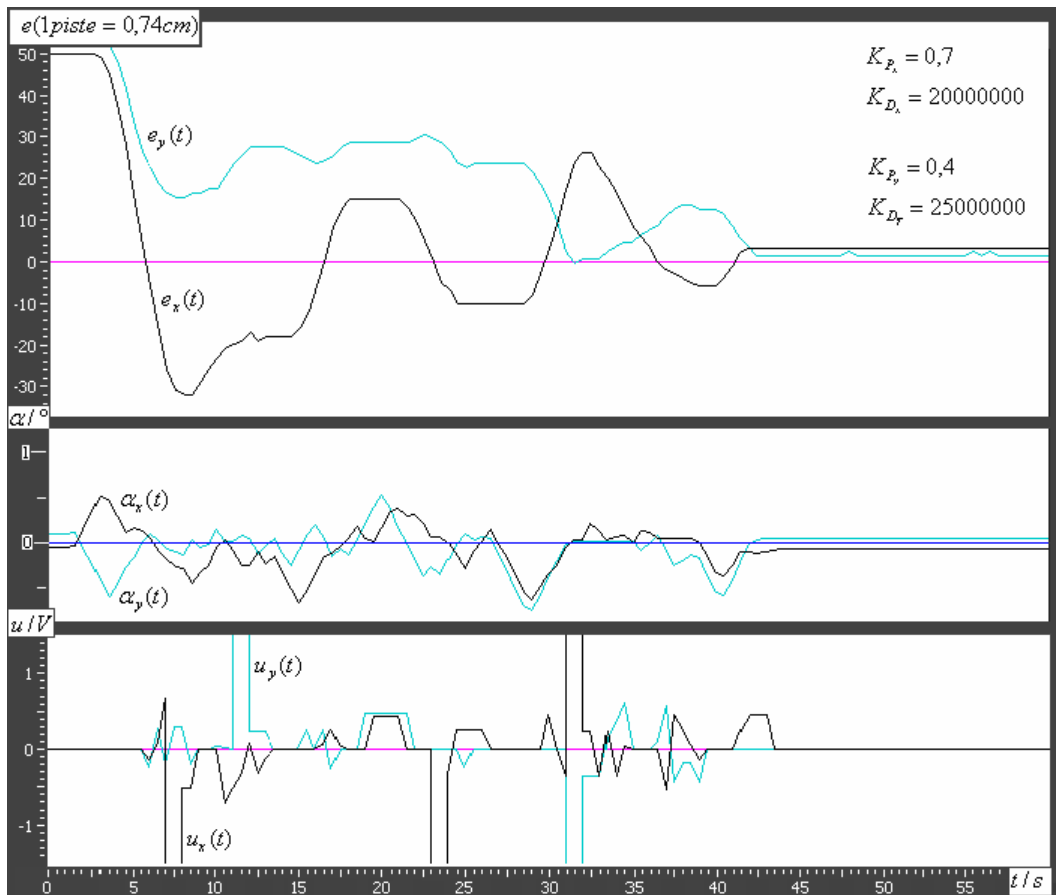


Kuva 41. Testiajo suurilla nopeuksilla $v_p = 40$, $v_s = 40$ ja $v_{sp} = 40$, jossa liikkeellelähtösäys ($t = 1,5$ s) ajaa x-akselin kaltevuuden α_x lähes ääriajaan asti, mikä saa kuulan vierimään suurella nopeudella kohti tavoitepistettä, y-akselilla sysäyskäyttö jää jostain syystä vajaaksi. Pikainen vastakallistus ($t = 2,5$ s) hidastaa kuulan liikettä jonkin verran.

Säätö siirtyy perusajoon tason saavutettua tasapainopisteen ($\alpha = 0^\circ$). PD-säätö tulee mukaan ohjaukseen, mikä aiheuttaa tason kallistuksen vastakkaiseen suuntaan ja hidastaa hiukan kuulan etenemistä. Kuula ylittää tavoitepisteen suhteellisen suurella nopeudella ($t = 4$ s). D-säätö reagoi tasapainopisteen ($\alpha = 0^\circ$) ylitykseen nopeasti, mikä tehostaa P-säädön vaikutusta ja pysäyttää kuulan etenemisen tavoitepisteestä pois päin ($t = 6$ s). Kuula pysyy hetken aikaa lähes paikoillaan (jumittuu alustaan), mutta liikkuu kuitenkin sen verran, että sysäysohjaus ei mene päälle. Taso tekee edestakaisia liikkeitä tasapainotilan molemmin puolin irrottaen kuulan ($t = 10$ s). Taso kallistuu jälleen jyrkästi tavoitepisteen suuntaan, ja ylittää sen hetkisen ääriajan suuren perusnopeuden v_p

vuoksi. Vaikka D-säätö reagoi jatkuvasti nopeisiin kallistuksiin, niin askelmoottori ei ehdi reagoida jokaiseen käskyyn. Tämän vuoksi kuula etenee rauhattomasti sekä x- että y-akselilla saavuttamatta koskaan tavoitepistettä $\mathbf{r}^T = (r_x, r_y)$. Tätä nopeutta voidaan pitää perusnopeuden maksimi äärirajana.

Koska suurien nopeuksien äärirajat löytyivät, niin seuraavaksi aseteltiin nopeudet: $v_p = 230$, $v_s = 230$ ja $v_{sp} = 230$ eli erittäin hitaat nopeudet. Suoritettiin kolme ajoa, joista kuula löysi tiensä tavoitepisteeseen toleranssin rajoissa kaksi kertaa.



Kuva 42. Testiajo pienillä nopeuksilla $v_p = 230$, $v_s = 230$ ja $v_{sp} = 230$, jossa liikkeelähtösysäys ($t = 2$ s) kallistaa tasoja vain hyvin vähän, joten kuula irtoaa x-akselilla vasta kallistuksen α_x ollessa suurimmillaan ja y-akselilla vasta sysäyksen palautuksensa. Lisäksi sysäysaika muodostuu suureksi (noin 2,5 s) mikä vaikeuttaa kuulan ohjattavuutta tasolla.

Ohjauksen siirryttyä perusajolle ($t = 3,5$ s) PD-säätö tulee mukaan ohjaukseen. PD-säätö reagoi heti kuulun nopeuteen ja etäisyyteen tavoitepisteestä, mutta hitaasta perusnopeudesta v_p johtuen askelmoottorin hitaat liikkeet eivät ehdi ohjata tasoja säädön edellyttämällä tavalla. Kuula pysähtyy jatkuvasti aiheuttaen aina hitaan sysäyskäytön, mutta löytää kuitenkin useimmiten tavoitepisteen, epävarmasta ohjauksesta huolimatta.

Tätä pienempiä nopeuksia ei tarvinnut enää testata, eli tätä nopeutta voitiin pitää hitaan nopeuden äärirajana. Seuraavaksi testattiin nopeuksia välillä 230–150, ja perusajon ohjattavuus parani koko ajan.

Seuraavaksi testattiin säätöä nopeuksilla $v_p = 150$, $v_s = 150$ ja $v_{sp} = 150$. Suoritettiin kolme ajoa, joista kaksi löysi tiensä perille määritellyssä ajassa. Perusnopeus v_s näytti hyvältä, koska se ohjasi kuulaa kohti tavoitepistettä suhteellisen hallitusti. Mutta kun kuula pysähtyi, aiheuttaen sysäyskäytön (v_s, v_{sp}), niin se karkasi hyvinkin kauas tavoitepisteestä, koska sysäyskäyttö oli liian hidaskäyttö. Koska perusnopeus näytti suhteellisen hyvältä, niin seuraavaksi testattiin sysäysnopeuksia välillä 150 – 20. Toimivuus parani koko ajan, kunnes päästiin seuraaviin sysäysnopeuksiin: x-akseli, $v_s = 70$ ja $v_{sp} = 20$, y-akseli, $v_s = 45$ ja $v_{sp} = 35$.

Kolmesta testiajosta kaikki löysivät tiensä perille asti määritellyssä ajassa. Sysäyskäyttö sai kuulun tasaisesti liikkeelle ja askelmoottori ehti reagoida PD-säädöltä tuleviin ohjauskäskyihin $u_x(t)$ ja $u_y(t)$ ohjauksen siirryttyä perusajoon. Kuula eteni x-akselin suuntaisesti ($e_x(t)$) tavoitepisteen tuntumassa suhteellisen pienin poikkeamin, y-akselilla ($e_y(t)$) tuli heittoa hieman enemmän, mikä johtui sysäysajosta. Kyseiset arvot valittiin kuulapelin nopeusarvoiksi, mutta PD-parametrien (K_p, K_D) virittämisen jälkeen palataan vielä y-akselin sysäysajoon.

4.1.3. PD-säädön parametriviritys

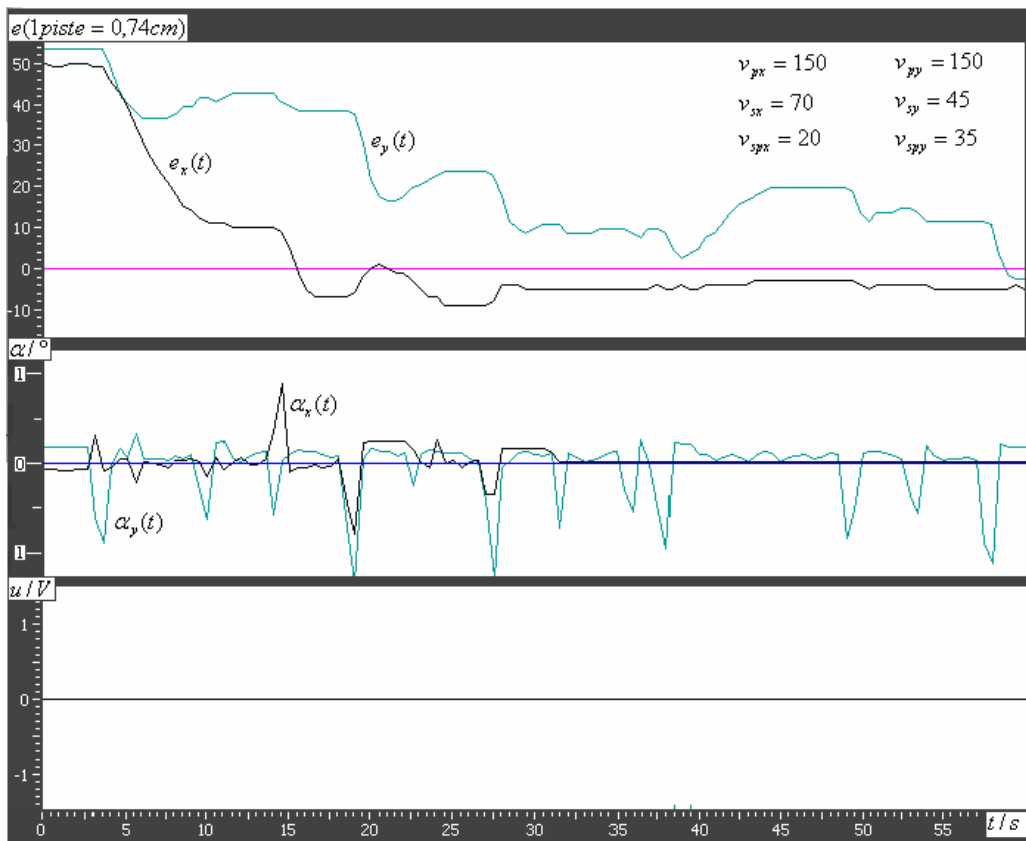
Säätimelle asetetut vaatimukset (stabiilius, nopeus ja tarkkuus) ovat osittain ristiriitaisia, koska esimerkiksi nopeuden lisääminen huonontaa järjestelmän suhteellista stabiiliutta. Tämän vuoksi joudutaan aina tekemään kompromissi eri vaatimusten välillä. On pyrittävä suureen nopeuteen ja tarkkuuteen ilman että joudutaan niin lähelle värähtelyrajaa, että häiriöt voisivat tehdä järjestelmän epästabiiliksi.

Kuulapelin säätimen parametointi on aika hankalaa millään tietyllä menetelmällä, joten pelin PD-säädin viritetään kokeellisesti.

Nopeusparametrit aseteltiin seuraavasti: $v_{px} = 150$, $v_{sx} = 70$ ja $v_{spx} = 20$, $v_{py} = 150$, $v_{sy} = 45$ ja $v_{syy} = 35$.

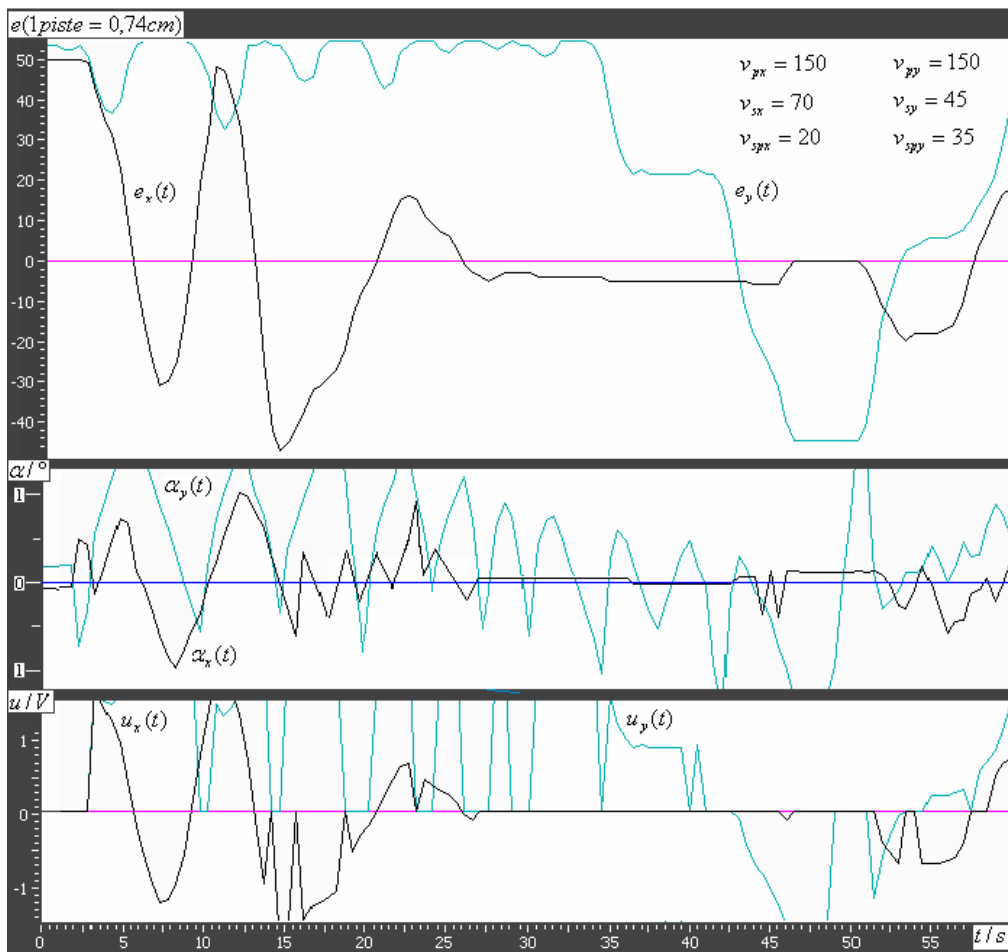
Kuulan sijainti lähtötilanteessa on sama kuin edellisessäkin mittauksessa, eli pelitason vasen alareuna, missä koordinaattipisteet ovat (33, 10). Kohdearvoksi määriteltiin jälleen (84, 60) eli suurin piirtein laudan keskikohta. Aseteltiin K_P ja K_D -parametrit säätimeen ja seurattiin kuulan kulkua kohti kohdekoordinaattia yhden minuutin ajan. Kuulan sijaintipisteet päivittyivät kuvaajaan 0,5 sekunnin välein.

Ensimmäisessä mittauksessa molempien askelmoottoreiden ohjauksien K_P - ja K_D -parametrit asetettiin nolaksi, jolloin PD-säätö ei vaikuttanut prosessiin. Tällä mittauksella haettiin vertailukohde tuleville mittauksille.



Kuva 43. Testausajo ilman PD-säätöä ($K_p = 0$ ja $K_D = 0$). Kuula etenee kohti tavoittepistettä $\mathbf{r}^T = (r_x, r_y)$ pelkästään sysäyskäyttöjen ohjaamana. Eli aina, kun kuula tasolla pysähtyy, sysäyskäyttö laittaa sen jälleen liikkeelle. Pieni liike, joka näkyy tasojen kallistuksissa, johtuu siitä, että taso hakee jatkuvasti tasapainotilaa ($\alpha = 0^\circ$), jota se ei kuitenkaan koskaan pysty saavuttamaan. Vertailuarvona tasapainotilassa, ilman PD-säätöä, on toleranssiton luku (2,5), joka ylittyy tai alittuu koko ajan, mikä aiheuttaa edestakaista ajoa.

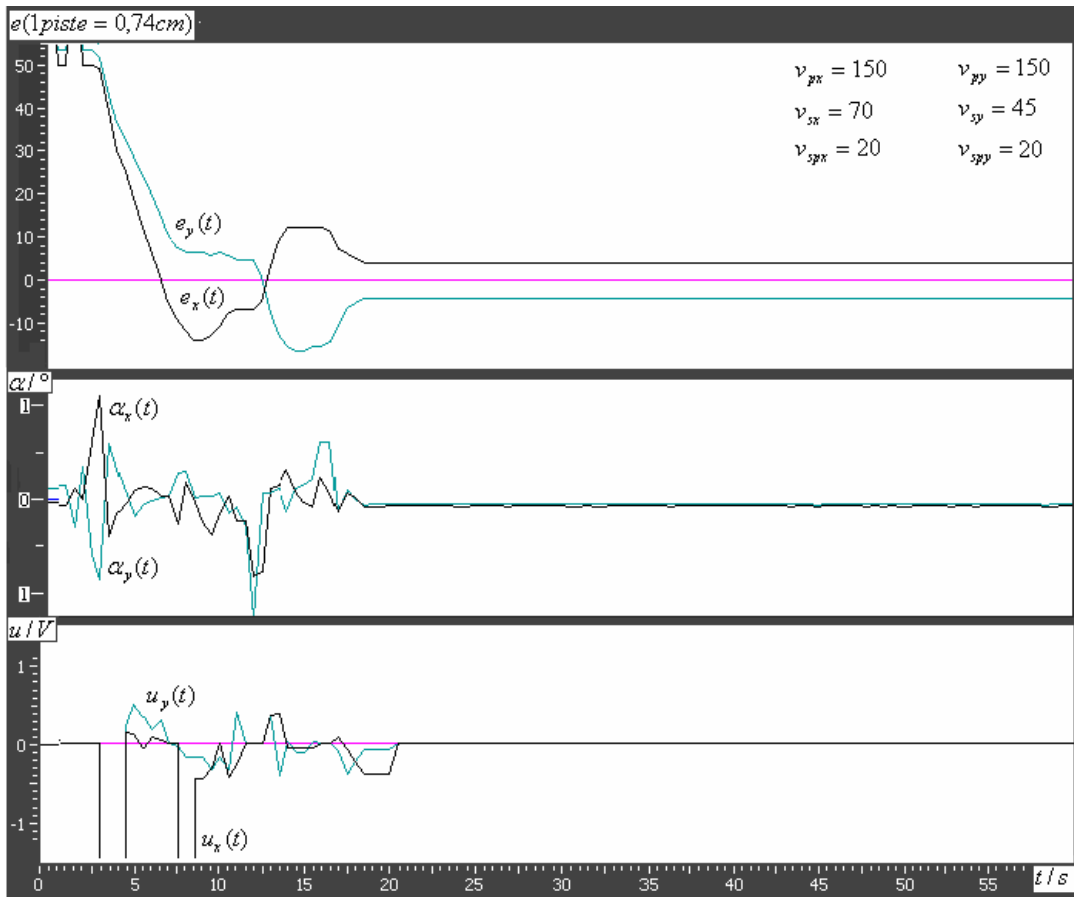
Seuraavaksi etsittiin vahvistuksen K_p arvo, missä säätö lähtee värähtelemään. Vahvistuksen arvo $K_p = 4$ oli aivan liian suuri, mikä aiheuttaa säätöjärjestelmän kaatumisen. Vahvistuksen arvolla $K_p = 3$ säätöjärjestelmä ei enää kaatunut, mutta lähti värähtelemään ja ylitti kallistuksen ääriraja-arvot useaan otteeseen. Kyseisen testiajon kuvaaja on esitetty kuvassa 44.



Kuva 44. Testausajo PD-parametriarvoilla $K_p = 3$ ja $K_D = 0$. Tasot kallistelevat tasa-painotilan molemmin puolin epämääräisesti, mikä aiheutti täysin hallitsemattoman ohjauksen kuulalle.

Seuraavaksi kokeiltiin vahvistuksen arvoja $K_p = 1 - 2$ ja ohjaus parani koko ajan. Vahvistuksen arvolla $K_p = 1$ otettiin D-vahvistus K_D mukaan säätöön. K_D -arvon suuruusluokka määriteltiin virtuaalisen mittapään avulla testiajotilanteessa. Erosuure $e(t)$ vaihteli välillä $0 - 1$, joten D-termi muodostui hyvin pieneksi. K_D -vahvistusarvoja testattiin välillä $1 \times 10^7 - 3 \times 10^7$, kunnes löydettiin sopivin vaihtoehto, mikä oli molemmilla, sekä x- että y-akselilla $1,5 \times 10^7$. Tämän jälkeen säädettiin vielä vahvistuksen K_p -parametria ja todettiin, että $K_p = 1$ on paras vaihtoehto molempien suuntien (x, y) säädöissä. Tämän lisäksi säädettiin vielä y-akselin sysäysajon palautusta hiukan nopeammaksi $v_{syy} = 20$. Kyseiset parametriarvot valittiin kuulapelin säätöarvoiksi. Testiajoja valitui-

la säätöarvoilla suoritettiin kaiken kaikkiaan 10 kpl, joista yksi on esitetty kuvassa 45 ja tulokset on esitetty taulukossa 3.



Kuva 45. Testausajo 1 parhailla parametriarvoilla. Sysäyskäyttö saa kuulan tasaisesti liikkeelle. Perusohjaukseen siirryttäessä PD-säätö saa kuulan liikkeit hyvin hallintaansa ja ohjaa sen suhteellisen joustavasti tavoitepisteeseen.

Taulukko 3. Testiajot valituilla nopeus- ja PD-parametriarvoilla.

Testiajo	t / s
1	10
2	32
3	32
4	60
5	12
6	80
7	35
8	8
9	30
10	32

Pisin aika t tavoitepisteen löytymiseen oli 80 s ja lyhyin 8 s. Tavoitepisteen löytymiseen vaikutti tasopinnan epätasaisuus (keskeltä hiukan kupera) ja kuulun pinnan epätasaisuus. Tavoitepiste löytyi suhteellisen nopeasti jokaisella mittauksella, mutta kuula saattoi vieriä hitaasti kohteen yli kuperuuden vuoksi tai pysähtyä tavoitealueen ulkopuolelle, jolloin sysäyskäyttö aiheutti pienen karkaamisen.

4.2. Tulosten tarkastelu

Nopeutta säädetään numeroarvoin välillä 0 – 255, missä 0 tarkoittaa maksiminopeutta v_{\max} ja 255 tarkoittaa miniminopeutta v_{\min} . (Nopeustiedot, suuntatiedot ja pysäytyskäskyt välittyvät askelmoottorien ohjaimille I2C-väylää pitkin 8-bittisinä jonoina). Nopeusparametreja valittaessa, testattiin perusnopeutta (v_p) ja sysäysnopeuksia (v_s, v_{sp}) useilla erisuuruksilla nopeusarvoilla. PD-säätimien parametrit (K_p, K_D) olivat tällöin aseteltu suhteellisen hyvin toimiviin arvoihin. Liian nopea perusnopeus $v_p < 80$ aiheutti sen, että vaikka D-säätö reagoi jatkuvasti nopeisiin kallistuksiin, niin askelmoottori ei ehtinyt reagoida jokaiseen käskyyn. Kuula eteni rauhottomasti pelitasolla saavuttamatta koskaan tavoitepistettään. Liian hidaskäyttö perusnopeus $v_p > 200$ taas aiheutti sen, että vaikka PD-säätö reagoi heti kuulun nopeuteen ja etäisyyteen tavoitepisteestään, niin askelmoottorien hitaat liikkeet eivät ehtineet ohjata tasoa säädön edellyttämällä tavalla.

Testien perusteella perusnopeus $v_s = 150$ näytti molempien akselien suhteen parhaalta, koska se ohjasi kuulaa kohti tavoitepistettä suhteellisen hallitusti.

Tason edestakaisen kallistelun (sysäyksen) tarkoitus on saada paikallaan oleva kuula mahdollisimman tasaisesti liikkeelle, viemättä kuulaa kuitenkaan kovin kauas irtoamispisteestään. Hitaat (>80) sysäysnopeudet (v_s, v_{sp}) veivät kuulan hyvinkin kauas irtoamispisteestään, nostaen samalla kuulan nopeuden vaikeasti hallittavaksi. Tämän vuoksi sysäysnopeudet (v_s, v_{sp}) valittiin mahdollisimman suuriksi, eli sellaisiksi, että askelmoottorien vääntömomentit ja kuulapelin mekaniikka kestivät nopeat sysäysajot häiriöttä. Kuulapelille valittiin seuraavanlaiset sysäysnopeudet: $v_{sx} = 70$ ja $v_{spx} = 20$, $v_{sy} = 45$ ja $v_{syy} = 35$.

Kuulapelin säätimen parametointi oli aika hankalaa millään tietyllä menetelmällä, joten pelin PD-säädin viritettiin kokeellisesti. PD-säätö tulee mukaan ohjaukseen heti sysäysajon jälkeen, kun säätö siirtyy perusajoon. Aluksi etsittiin vahvistuksen K_p ääriraja-arvo, jossa säätö lähti värähtelemään. Kun tämä löytyi, testattiin pienempiä vahvistuksen K_p arvoja, kunnes löydettiin suhteellisen toimiva arvo. Derivointivahvistuksen K_D suuruusluokka määriteltiin virtuaalisen mittapään avulla testiajotilanteessa. Erosuure vaihteli välillä 0-1, joten D-termi muodostui hyvin pieneksi. Kun K_p ja K_D suuruusluokat olivat selvillä, suoritimme kuulapelillä useita testiajoja erilaisilla K_p ja K_D arvoilla, kunnes sopivin vaihtoehto löytyi. Sopivin K_D parametri oli molemmilla, sekä x-että y-akselilla $1,5 \times 10^7$ ja vahvistuksen K_p -parametri 1.

Valituilla parametreilla suoritettiin 10 testausajoa. Kuula pysähtyi tavoitepisteeseen joka testausajolla, mutta ajat vaihtelivat 8s ja 80s välillä. Tavoitepisteen löytymiseen vaikutti tasopinnan epätasaisuus (keskeltä hiukan kupera) ja kuulan pinnan epätasaisuus. Tavoitepiste löytyi suhteellisen nopeasti jokaisella mittauksella. Kuula saattoi kuitenkin vierä hitaasti kohteen yli tason kuperuuden vuoksi tai pysähtyi tavoitealueen ulkopuolelle

hiljaisesta nopeudesta ja kuulan pinnan epätasaisuudesta johtuen. Sysäyskäyttö sai kuitenkin kuulan taas liikkeelle, mutta aiheutti pienen karkaamisen.

4.3. Kokemuksia ja ongelmia

LabVIEW oli toteutusympäristönä aluksi melko hankala, johtuen tekijöiden kokemattomuudesta. Lähes kaikki toiminnot toteutuivat yrityksen ja erehdyksen kautta. Esimerkiksi Joystick-ohjauksen x- ja y-suuntien ohjaukset (askelmoottorit) tuottivat aluksi ylipääsemättömiä vaikeuksia ohjelmankiertoa säätelevien kellopulssien määrittelyjen suhteen: Kaksi samassa kytkennässä olevaa kelloa aiheuttivat aikapulssien vääristymistä, mikä vaikeutti askelmoottorien tahdistamista. Toiminta korjaantui siirtämällä molemmat dimensiot omiin silmukoihinsa ja käyttämällä tietojen siirrossa globaaleja muuttujia.

Askelmoottoriohjauksessa ongelmaksi muodostui Ismo Tupamäen valmistama ohjainkortti, joka yhdistettiin suoraan rinnakkaisporttiin. Kortin komponentit paloivat useaan otteeseen, joten työhön hankittiin valmiit kortit Vaasan elektroniikkakeskuksesta. Hankittujen ohjainkorttien askelmoottorit olivat aivan liian pienitehoiset ohjaamaan pelin tasoja, joten kortteihin hankittiin suurempitehoiset (ja -kokoiset) askelmoottorit. Vaikka askelmoottorien ottama virta ei noussutkaan yli askelmoottorien ohjainkorttien kestoraajan, niin silti toisesta ohjainkortista paloi komponentteja yksi kerrallaan. Spesifikaatioiden mukaan kaiken piti olla kunnossa. Kun tämä tapahtuma toistui useaan otteeseen, niin kortin kaikki komponentit vaihdettiin virtakestoltaan suurempiin. Epäilimme moottorilta porttiin päin tulevan virtapiikin aiheuttavan komponenttien tuhoutumisen, jolloin diodisuojaus olisi auttanut, mutta emme lähteneet tutkimaan asiaa sen syvällisemmin. Komponenttien vaihdon jälkeen ongelmat poistuivat.

Työ eteni ja syntyneet ongelmat selvisivät suhteellisen vaivattomasti, kunnes ilmeni aivan uudenlainen ongelma. Tietokoneen prosessoritehot nousivat 100 % automaattijolla, ja kone jumittui kesken ohjelman suorituksen. Saimme prosessoritehot pienemmiksi pudottamalla ohjelman kiertonopeutta. Tämä auttoi prosessoritehoon, mutta

ohjelman toiminta huononi entisestään. Lukunopeuden pienentäminen vaikutti muun muassa potentiometreiltä ja kameralta tulevien tietojen käsittelynopeuteen. Pitkän selvitksen jälkeen vika löytyi laaditun ohjelman rakenteesta, eli kun tietyt ohjelman osat yhdistettiin järjestelmään globaalien muuttujien välityksellä, prosessoritehojen kanssa ei ollut enää ongelmia.

X- ja y-ajoissa ilmeni edelleen ristiriitoja. Yritimme ratkaista ristiriidat rakentamalla ohjelmallisesti vuorottelun, eli x- ja y-ajot toimivat vuorotellen tietyn ajan. Käsiäjo-ominaisuudet paranivat huomattavasti, mutta automaattiajossa ilmeni edelleen jumittumista. Automaattiajoon liittyvässä säädössä tasot eivät ehtineet alkuunkaan mukaan haluttuun toimintaan. Vika ilmeni I2C-väylän viestien törmäilyillä. Tämän vuoksi asensimme järjestelmään toisen I2C-väylän y-akselin ohjaukseen. Asentamisessa oli omat hankaluutensa, mutta saimme sen lopulta toimimaan. Käsi- ja automaattikäyttö toimi tältä osin erinomaisesti, eikä enää ilmennyt jumittumista.

Pitkällisten kokeilujen jälkeen toiminta saatiin kuitenkin suhteellisen varmaksi ja säätö alkoi totella asetusarvoja ja kuula löysi tiensä kohteeseen.

Kun kuulapelin mekaniikka ja ohjaus saatiin toimintavarmaksi, suoritettiin useita pelikuulaan ja säätöön liittyviä testejä. Pelikuulan valintakriteereinä olivat näkyvyys kamerassa ja hallittavuus. Testattavana oli yksitoista erikokoista ja painoista kuulaa, joilla tehtiin lähtökitka-, nopeus- ja kimmoisuustestit. Kaksi kuulaa oli edellä mainittujen testien valossa yli muiden, joten näkyvyys kamerassa ratkaisi valinnan. Pelikuulaksi valittiin musta, hiiriohjaimesta otettu kuula.

Tämän jälkeen haettiin säätöön oikeanlaiset parametrit kokeilumenetelmällä. Aluksi testattiin erisuuruisia nopeusparametreja perusajoon sekä sysäysajoon. Sysäysajossa oli omat nopeussäädöt, kallistukseen ja palautukseen. Ennen nopeusparametrien testausta valittiin perusajoon vaikuttavaan PD-säätöön suhteellisen hyvin toimivat parametriarvot. Nopeuksia testattiin useilla eri arvoilla ja tultiin siihen tulokseen, että perusajon täytyi olla suhteellisen hidas ja sysäysajon täytyi olla erittäin nopea kuulan hallinnan

tähden. Kun nopeusarvot olivat kohdallaan, testattiin PD-säätöä erilaisilla K_P ja K_D -parametrien arvoilla.

Ongelmista huolimatta, kuulapelin prototyyppi saatiin toimintakuntoiseksi (kuva 46).



Kuva 46. Tekijä testaamassa omia taitojaan.

5. YHTEENVETO

Kuulapeli on taitopeli, jossa kuula ohjataan tasoja kallistamalla määrättyä reittiä pitkin haluttuun kohteeseen, joko manuaalisesti tai automaattiajolla. Kuulapeli koostuu kolmesta hieman erikokoisista neliönmuotoisista puukehikoista, jotka on aseteltu sisäkkäin ja yhdistetty laakerien välityksellä toisiinsa. Askelmoottorit on kiinnitetty pelin runkoon ja ne ohjaavat vaijerien välityksellä tasojen kaltevuuksia. Askelmoottorien ohjainkortit sijaitsevat pelin runkoon sijoitetussa ohjauskotelossa ja ne ovat yhdistetty tietokoneeseen kahdella erillisellä I2C-väylällä (molemmilla askelmoottoreilla omansa). Tasokulmia seuraavat potentiometrit ja äärirajat on yhdistetty ohjauskeskuksessa sijaitsevan CB-68LPR-kortin kautta tietokoneeseen SCH-68-EPM SHIELDED välikaapelilla. Pelin ohjaimena tietokoneessa toimii LabVIEW, graafinen ohjelmointiympäristö, mihin kaikki pelin vaatimat toiminnot on koottu yksittäisistä ohjelmamoduuleista eli virtuaali-instrumenteista.

Kuulapelissä on kolme erilaista toimintoa:

- Tietokoneelle määritellään kohdekoordinaatit eli tavoitepiste, johon kuulan halutaan tasolla siirtyvän. Kuula etenee tavoitepisteen tietokoneohjatusti.
- Tietokoneelle määritellään kohdekoordinaatit eli rataprofiili, jota pitkin kuulan halutaan tasolla kulkevan. Kuula etenee määritelyä rataprofiilia tietokoneohjatusti.
- Käyttäjä ohjaa itse kuulan tavoitepisteeseen Joystick-ohjaimen avulla.

Manuaaliajossa käyttäjä ohjaa kuulan tavoitepisteeseen Joystick-ohjaimen avulla. Joystick-ohjain on liitetty tietokoneeseen USB-portin kautta.

Automaattiajossa Firewire-kamera seuraa kuulan kulkua tasolla, ja se lähettää kuulan paikkatietoa takaisinkytkentätietona säätimelle 133 ms väliajoin (kuvankäsittely on esi-

tetty tarkemmin Ismo Tupamäen diplomityössä). Tietokone paikantaa kuulan ja laskee, mihin kulmaan tasot kannattaa kääntää, jolloin paikannus onnistuu nopeasti.

Säädön toimintaperiaate on seuraavanlainen: Automaattiajossa käytetään kahta eri ajo-tyyppiä, sysäys- ja perusajo. Sysäysajo saa kuulan liikkeelle aloituksessa ja silloin kun kuula pysähtyy tavoitepisteen ulkopuolelle ja lähettää kaksi samaa paikannustietoa peräkkäin. Sysäysajolla tarkoitetaan sellaista tason nopeaa kallistusta ja sen palautusta, mikä saa kuulan liikkeelle. Sysäysajon jälkeen kuulapeli siirtyy automaattisesti perusajoon. Perusajossa PD-säätö ohjaa kuulaa erosuureen ja kuulan sen hetkisen nopeuden perusteella.

P-säätö määrittelee tason äärikallistuspisteen, mikä on suoraan verrannollinen erosuureeseen eli mitä pidempi matka tavoitepisteeseen on, sitä jyrkemmin taso kallistuu. D-säätö joko vahvistaa tai heikentää P-säätöä seuraavasti:

Kun kuula lähestyy tavoitepistettä D-säätö alkaa heikentää P-säädön vaikutusta, se kuinka suuri D-säädön vaikutus on, riippuu kuulan senhetkisestä nopeudesta ja D-vahvistuksesta. Jos tavoitepiste ylittyy ja kallistus-suunta muuttuu vastakkaiseksi, D-säätö vahvistaa P-säätöä niin kauan, kunnes kuulan suunta jälleen muuttuu kohti tavoitepistettä. Jos kallistuksen senhetkinen ääriraja ylittyy esimerkiksi kuulan nopeuden vuoksi, askelmoottorit ajavat tasot automaattisesti tasapainotilaan, hilliten kuulan etenemistä. Kun kuula saavuttaa tavoitepisteen toleranssin rajoissa, askelmoottoreille välit-tyy I2C-väylää pitkin pysäytyskäsky ja toiminta pysähtyy.

Kuulapeliä voidaan käyttää automaatioalan-opetuksessa, viihdykkeenä erilaisissa tilai-suuksissa ja mahdollisessa graafiseen ohjelmointiin kohdistuvassa opetus- ja tutkimus-työssä.

5.1. Tulevaisuus

Kuulapelin kehitystyö on vasta alussa. Ainoastaan edellä mainittu pohjatyö kyseisen laitteen automatisoimiseksi ja mekanisoimiseksi on tehty. Tulevaisuudessa kuulapeliin olisi mahdollisuus kehitellä runsaasti uusia ominaisuuksia, sillä esimerkiksi viidestä pelin toiminnallisesta tavoiteportaasta (kappale 1.3) käytettiin ainoastaan kaksi, ja lisätavoitteitakin varmasti löytyy, esimerkiksi peliin voisi lisätä vaikkapa verkkopelattavuusmahdollisuuden.

Kuulapeli sopisi mm. koulujen vapaa-ajantiloihin, kerhoihin ja huoltoasemille flipperin tyylliseksi harrastepeliksi. Lisäksi kuulapeli sopisi monipuolisuutensa vuoksi oppilaitoksiin esimerkiksi automaatio- ja tietotekniikka-alan opetusvälineeksi. Toisin sanoen, kuulapeli on mahdollista jatkokehittää kaupalliseksi tuotteeksi, jolloin sen on täytettävä vaaditut turvallisuusvaatimukset, joista seuraavassa lyhyt selvitys:

Koneita koskevat tekniset vaatimukset sekä koneen vaatimustenmukaisuuden osoittaminen on yhdenmukaistettu Euroopan talousalueella. Kun tuote on valmistettu yhdenmukaistetun standardin mukaan ja sen vaatimustenmukaisuuden arviointimenettely on ao. direktiivin tai direktiivien mukainen, myös viranomaiset katsovat tuotteen täyttävän turvallisuusvaatimukset. Tällöin tuotteeseen voidaan kiinnittää CE-merkintä ja saattaa se markkinoille (Pahkala, Siirilä 1999).

Kuulapelin jatkokehittely edellyttää, että joku innostuu jatkokehittämään suunniteltua prototyyppiä. Työ sopisi päättötyöksi jollekin markkinointihenkiselle automaatioalan diplomi-insinööriopiskelijalle tai automaatioalan insinööriopiskelijalle, jolla on jonkin verran pohjatietoa LabVIEW-ohjelmoinnista.

VIITTEET

Airila, Mauri (1999). *Mekatroniikka*. 5. painos. Helsinki: Otatieto Oy.

Dadios, E.P, R. Baylon, R. De Guzman, A. Florentino, R.M. Lee & Z. Zulueta (2000). *Vision guided ball-beam balancing system using fuzzy logic*. Industrial Electronics Society. IECON 2000, 26th Annual Conference of the IEEE, vol. 3, sivut 1973 – 1978.

Jones, Douglas (2004). *Stepping motors fundamentals*. University of Iowa. Saatavana World Wide Webistä: <http://ww1.microchip.com/downloads/en/AppNotes/00907a.pdf>.

Halme, Terho (1992). *Säätö- ja ohjaustekniikka*. Oppikirja. Opetushallitus. Helsinki: Valtion painatuskeskus.

Harries, Ian (1997). *Sequences for a Four-Phase Unipolar Permanent Magnet Stepper Motor*. Saatavana World Wide Webistä: <http://www.doc.ic.ac.uk/~ih/doc/stepper/control2/sequence.html>, tilanne 20.06.2006.

Johansson, Erik (2000). *Säätötekniikka 2000*. Iisalmi: IS-PRINT.

Kankaanpää, Matti (2003). *Käyttötekniikan laboraatio*. Askelmoottori. Seinäjoen ammattikorkeakoulu.

Keinänen, Toimi, Pentti Kärkkäinen, Tommi Metso & Kari Putkonen (2002). *Logiikat ja ohjausjärjestelmät*. Porvoo: WS Bookwell Oy.

Knuuttila, Aarne, Atte Lindqvist & Harri Rätty (2004). *Säätöpiirit*. Teknillinen Korkeakoulu. Tuotantoautomaatio-kurssin seminaariesitelmä. Saatavana World Wide Webistä: http://www.tkk.fi/Yksikot/Konepaja/Opinnot/Kurssiesitteet/Tuotantoautomaatio/v2004/H_1_Saatopiirit1.pdf.

Koljonen, Janne (2004). *Virtual Keeper – Java-pohjainen konenäkösovellus*. Vaasan yliopisto. Diplomityö. Saatavana World Wide Webistä: <http://lipas.uwasa.fi/~jako/publications/VirtualKeeper.doc>.

Koskinen, Jari (2002). *Mikrotietokonetekniikka, sulautetut järjestelmät*. Keuruu: Otavan Kirjapaino Oy.

Kostamo, Jari, Heikki Hyötyniemi & Petri Kuosmanen (2005). *Ball balancing system: An educational device for demonstrating optimal control*. Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on 27-30 June 2005, sivut 379 – 384.

Laukonen, Eric & Stephen Yurkovitch (1993). *A ball and beam testbed for fuzzy Identification and control design*. American Control Conference, San Francisco, CA, sivut 665 – 669.

Laurikkala, Mikko (2006). *Sumea säätö, ACI-41050 Sumea laskenta*. Luentomateriaali. Saatavana World Wide Webistä: <http://www.ac.tut.fi/aci/courses/ACI-41050/Luennot/Luento6-2006-6s.pdf>.

Loh, Nan & Ka Cheok (1987). *A ball balancing demonstration of optimal and disturbance-accomodating control*. Control Systems Magazine, IEEE, vol 7, Issue 1, sivut 54 – 57.

Lohilahti, Jarmo (2002). *Ilmavirtauksen mittaus käyttäen ristikorrelaatiota*. The Manchester Metropolitan University. Opinnäytetyö.

Mäkelä, Mikko, Riitta Mäkelä & Olavi Siltanen (1997). *Insinöörikoulutuksen fysiikka*. Jyväskylä: Gummerus Kirjapaino Oy.

Nadin, Mihai (2004). *Visual Programming Languages – Efficiency of the Visual Driving Technology*. Saatavana World Wide Webistä:
<http://www.utdallas.edu/~pds017300/nadin/VPL.pdf>.

Narikka, Jorma (1996). *LabVIEW-ohjelmointi*. Ohjelmistotekniikan seminaarityö. Jyväskylän Yliopisto. Saatavana World Wide Webistä:
<http://www.mit.jyu.fi/opiskelu/seminaarit/ohjelmistotekniikkaLabIEW/#1.%20Esi-puhe>.

National Instruments. Kotisivut. <http://www.ni.com> , tilanne 10.10.2006.

National Instruments (2005). *Getting Started with LabVIEW version 8.0*. Irland.

National Instruments (2003). *Labview 7 express user manual*. Irland.

National Instruments (2006). *Controlling an X-Y stage with a Joystick*.
 Saatavana World Wide Webistä: <http://zone.ni.com/devzone/cda/tut/p/id/3161>.

National Instruments (2006). *Motion Control with USB Joystick*.
 Saatavana World Wide Webistä: <http://zone.ni.com/devzone/cda/epd/p/id/4726>.

Nevalainen, Ari (2004). *Askelmoottorit ja mikroaskellus*, 183116 Kone, 24501 Mekatroniikan perusteet, 7.12.2004.

Nihtilä, Markku (1976), Timo Oksanen (2001) & Lasse Eriksson (2002). P-, PI- ja PID – Sääto. Teknillinen korkeakoulu. Automaatio- ja säätötekniikan laboratoriotyöt.

Philips Semiconductors (2000). The I2C-bus specification. Saatavana World Wide Webistä: http://www.nxp.com/acrobat_download/literature/9398/39340011.pdf

Siirilä, Tapio (1999). EU-Määräysten mukainen koneiden turvallisuus. Keuruu, Otavan Kirjapaino Oy.

Säätötekniikan matematiikan verkkokurssi. *Dynaamisen järjestelmän stabiilisuus*.

Saatavana World Wide Webistä:

<http://www.control.hut.fi/Kurssit/AS-74.1102/verkkokurssi/Teoria/dynaaminen/dyn-stabiilisuus.html>.

Tervo, Kalevi (2006). *Säätötekniikan matemaattinen verkkokurssi*. Teknillinen korkeakoulu, systeemitekniikan laboratorio. Saatavana World Wide Webistä:

<http://www.control.hut.fi/Kurssit/AS-74.2111/materiaali.html>. Tilanne 9.10.2006.

Törmänen, Pasi (1997). *Geneettisten algoritmien käyttö tutkittaessa sumean logiikan hyötyä PID-säädössä - esimerkkinä taajuusmuuttajan sisäiset säätäjät*. Vaasan yliopisto. Pro gradu-tutkielma.

Vaittinen, Reijo & Jari Savolainen (2001). *Säätötekniikan perusteita*. Saarijärvi: Gummerus kirjapaino Oy.

Vartiainen, Lauri (1992). *Peliautomaattisivu*. Saatavana World Wide Webistä:

<http://www.huvipuisto.net/games/games>.

Velleman. Kotisivut. <http://www.velleman-kit.com/nl/en/product/view/?id=337828>, tilanne 03.12.2006.

Wexon, kotisivu. <http://www.wexon.fi/perus.php?id=119>, tilanne 15.10.2006.

Wikipedia. Vapaa tietosanakirja. *PID-säädin*. Saatavana World Wide Webistä:

<http://fi.Wikipedia.org/wiki/PID-s%C3%A4%C3%A4din>).

Wikipedia. Vapaa tietosanakirja. Kitka. Saatavana World Wide Webistä:

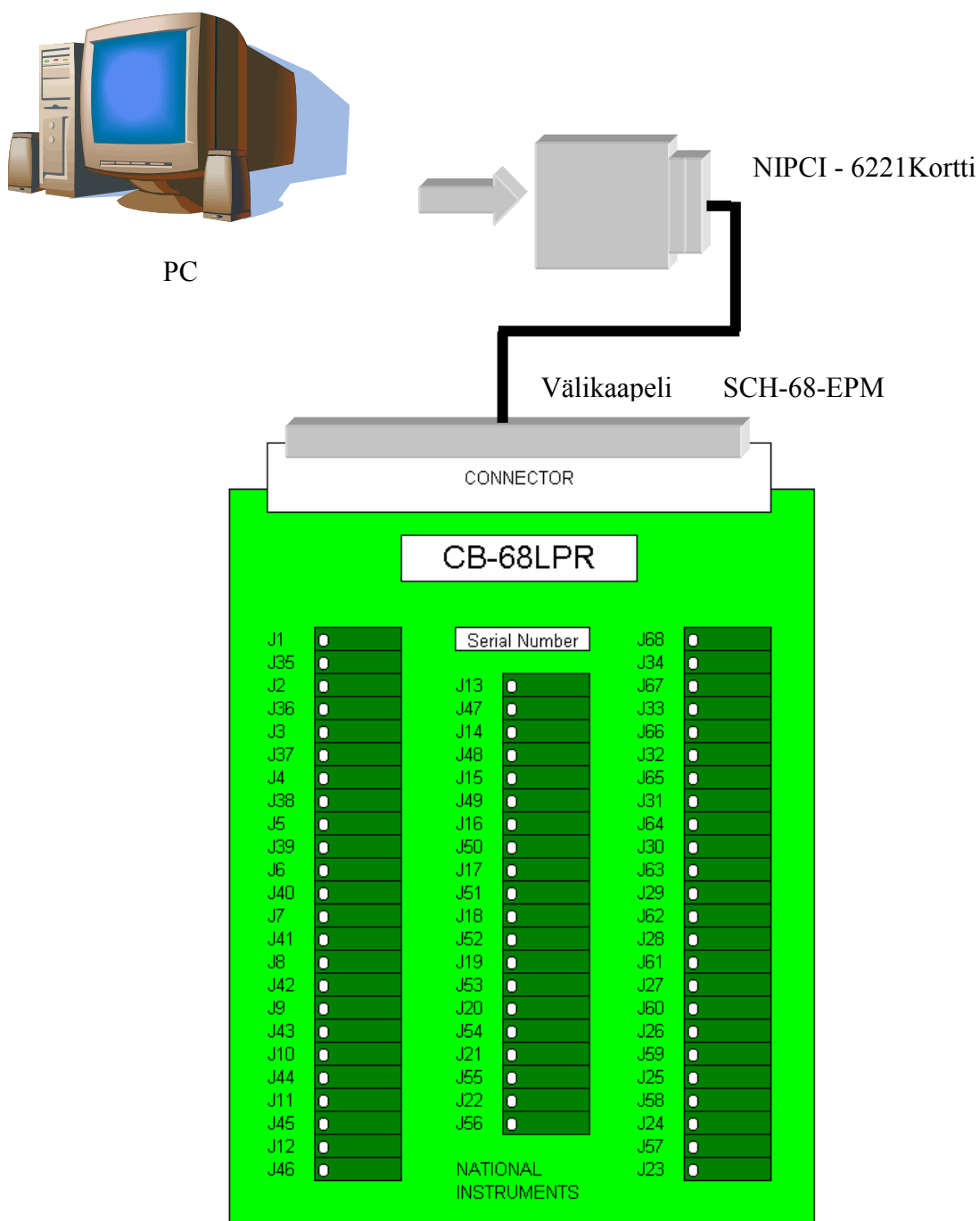
<http://fi.Wikipedia.org/wiki/Kitka>.

Zopel, D (1998). A Versatile Real-Time Experiment: Balancing a Ball on a Flat Board.

sivut. Real-Time Systems Education III, sivut 98 – 104.

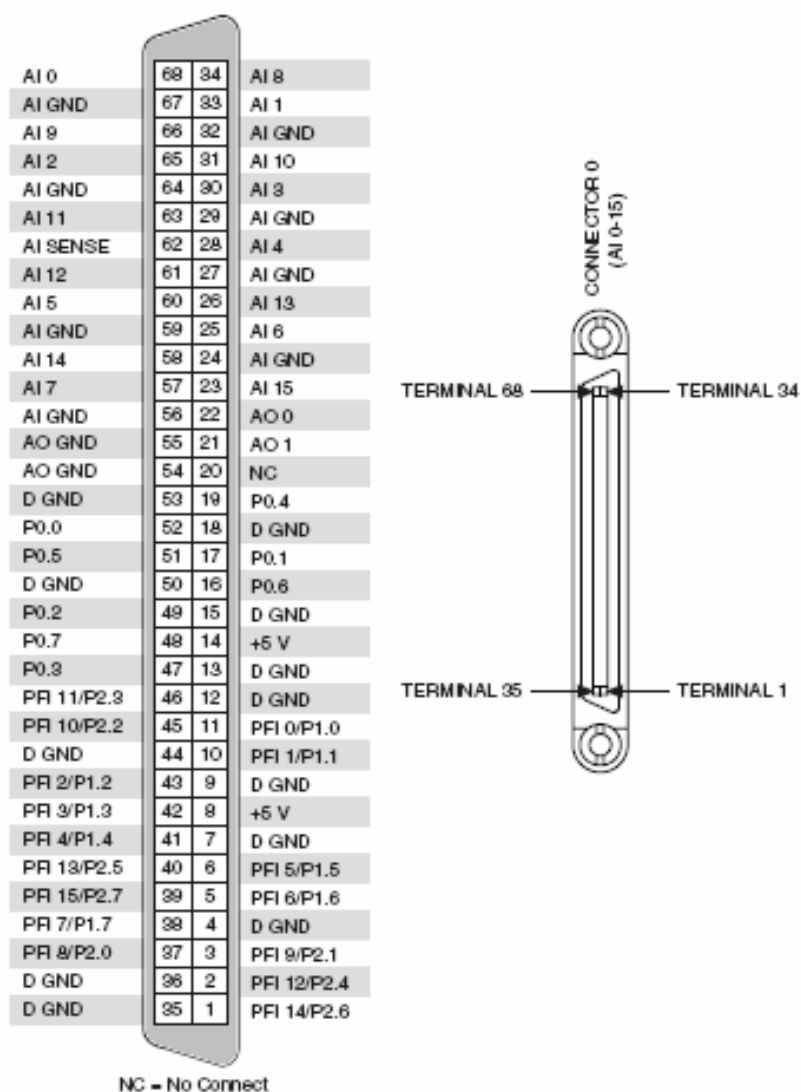
Liite 2. CB-68LPR-kortti.

Digitaalisten ja analogisten tietojen keruukortti (CB-68LPR-kortti), sijaitsee kuulapelin ohjauskeskuksessa ja siihen on kytketty muun muassa kuulapelin äärirajat ja liukupotentiometroit.



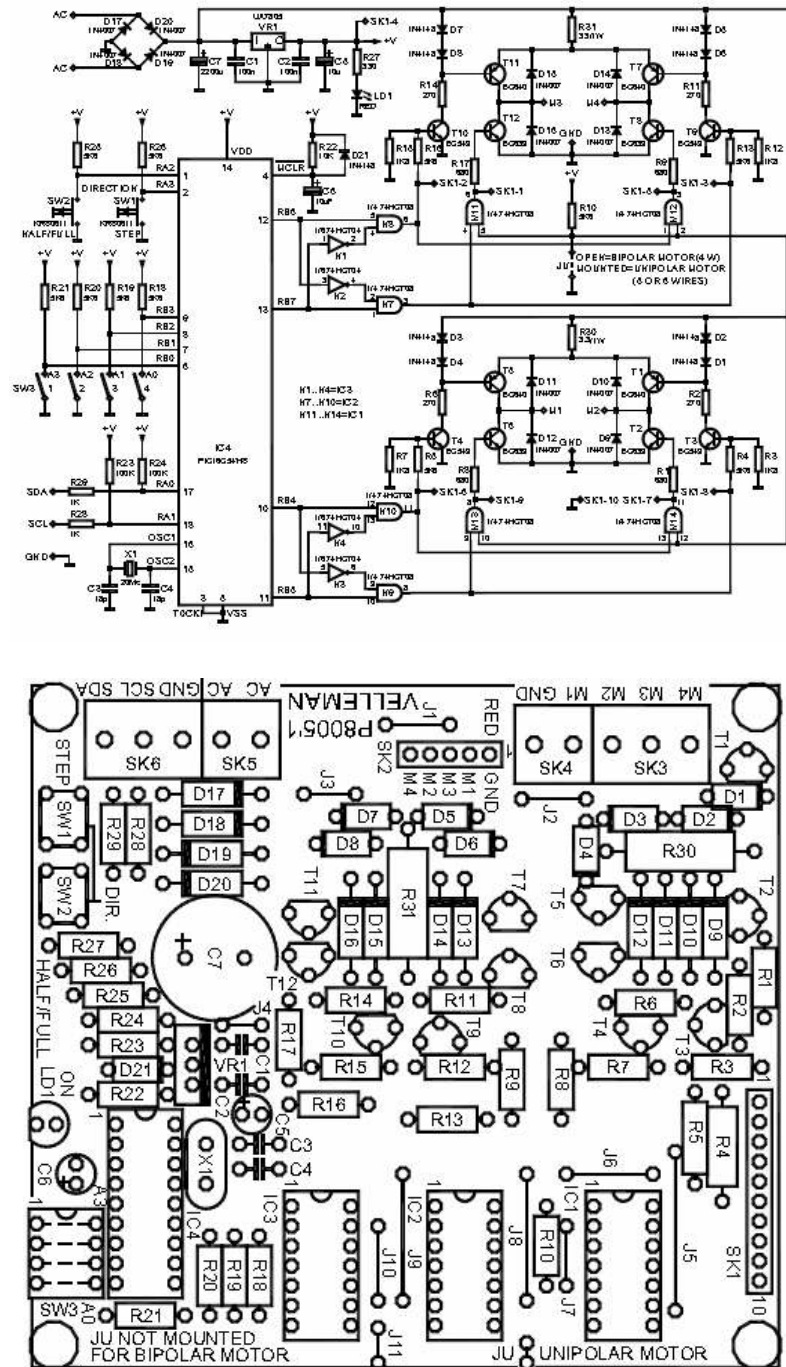
Liite 3. NI 6221 kytkentä (National Instruments 2006).

CB-68LPR-kortti yhdistetään tietokoneessa sijaitsevaan NIPCI – 6221 korttiin SCH-68-EPM-välikaapelilla, jonka kytkentä on esitetty alla olevassa kuvassa.



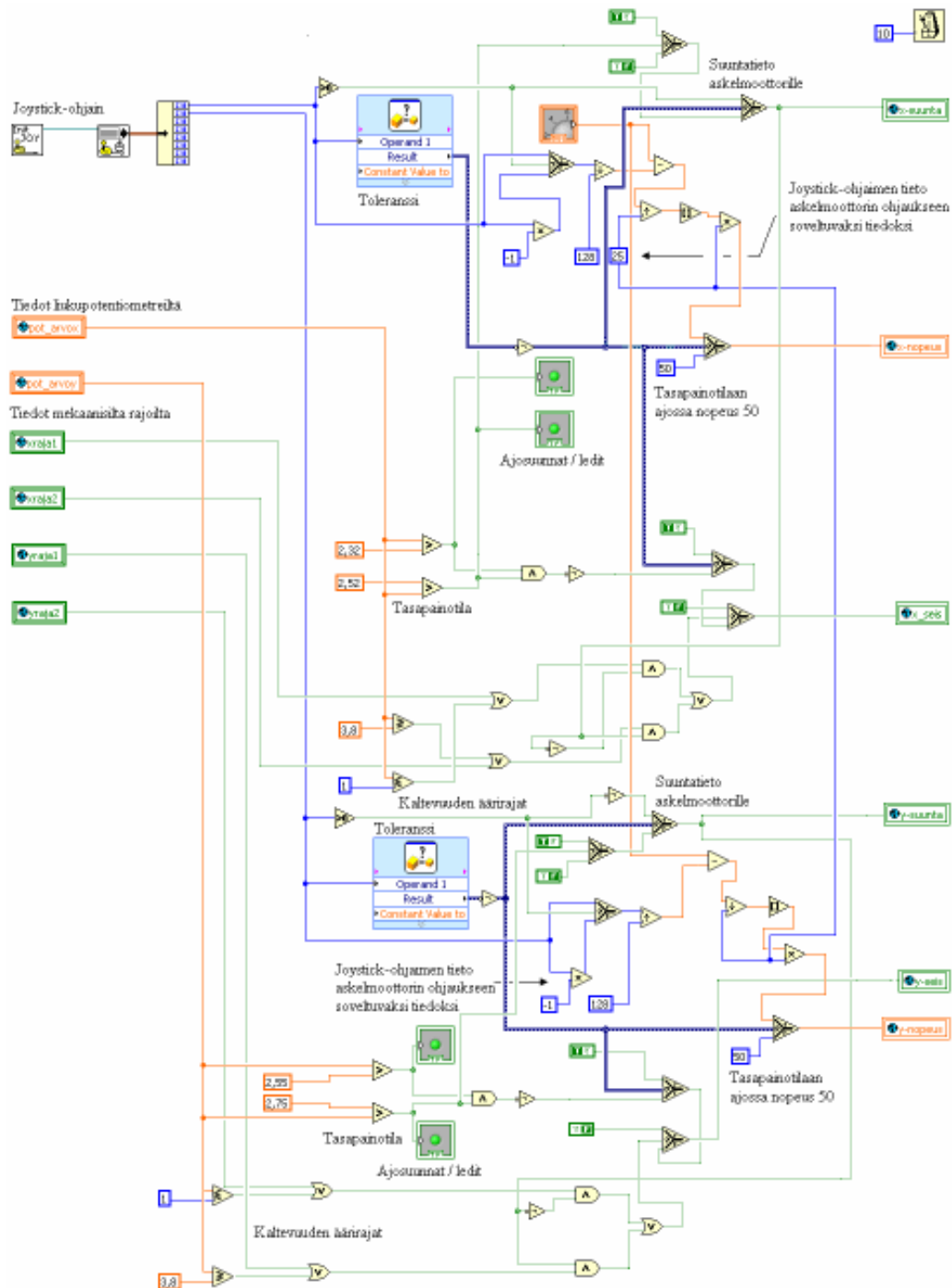
Liite 4. Askelmoottorin ohjainkortti K8005 (Velleman 2006).

Soveltuu bi- ja unipolarmoottoreille (maksimi 1A). Mahdollisuus suureen tarkkuuteen (jopa 0,9 astetta tai vieläkin parempi käytettäessä alennusvaihdetta). Samalla kun moottoria ohjataan, on mahdollista tarkistaa kytkimien tila tai ajaa toista moottoria. Häätäpysäytys on aina mahdollista suorittaa erillisen signaalin avulla (Velleman 2006).



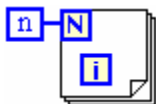
Liite 5. Käsiajon ohjauskaavio.

Käsiajossa pelikuula kuljetetaan Joystick-ohjaimen avulla haluttuun kohtaan tasoa, väistellen mahdollisia esteitä. Mekaaniset äärirajat estävät tasojen liiallisen kallistumisen. Tasojen kaltevuutta seuraavat liukupotentimetriä ohjaavat tasot takaisin vaakatasoon aina, kun Joystick-ohjaimesta päästetään irti.



Liitteet 7 - 11. Kuvakelohkoluettelo

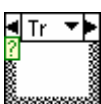
Kaikki tässä opinnäytetyössä esitetyt kuvakelohkot ja niiden selitykset.



For-silmukka toistaa silmukkaa n-kertaa ja i ilmoittaa silmukan toistot välillä 0 – (n-1). Erillisellä ajastimella määritellään silmukan toistonopeus.



While-silmukka toistaa silmukkaa aina, kun silmukkaehto on false ja i ilmoittaa silmukan toistot. Erillisellä ajastimella määritellään silmukan toistonopeus.



Case-valinta, voidaan kytkeä useita ohjauskaavioita rinnakkain. Tarvittava valinta suoritetaan joko numeroarvoilla (useampi, kuin kaksi ohjausta) tai true / false lauseilla.



Kytkin, lähettää true/false-tyyppistä ohjaustietoa (boolean).



Suorakaiteen muotoinen merkkivalo, toimii true/false-tyyppisellä ohjauksella (boolean).



Pyöreä merkkivalo, toimii true/false-tyyppisellä ohjauksella (boolean).



Kytkin, lähettää true/false-tyyppistä ohjaustietoa (boolean).



Lineaarinen säätö, esim. joystick-ohjaimen herkkyys.



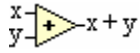
Asetusarvojen syöttö ohjelmaan, esim. PID-säätimen parametrit.



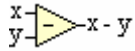
Mittari, esim. askelmoottorin nopeuden esittäminen.



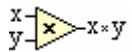
Kuvaaja, esim. säädön, tasokallistuksien ja kuulan etenemisen esitys aikayksikössä.



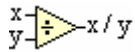
Yhteenlasku



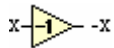
Vähennyslasku



Kertolasku



Jakolasku



Muuttaa luvun etumerkin vastakkaiseksi.



Pyöristää luvun x lähimpään kokonaislukuun y.



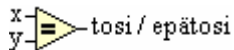
Syöttää arvoa 20 ohjaukseen.



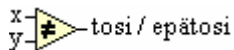
Lähetää true-tyyppistä tietoa ohjaukseen (boolean).



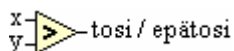
Lähetää false-tyyppistä tietoa ohjaukseen (boolean).



Jos tulo x on samansuuruinen kuin tulo y, niin lähtö on tosi. Muissa tapauksissa lähtö on epätosi.

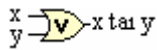


Jos tulo x on erisuuri kuin tulo y, niin lähtö on tosi. Muissa tapauksissa lähtö on epätosi.



Jos tulo x on suurempi kuin tulo y, niin lähtö on tosi. Muissa tapauksissa lähtö on epätosi.

 tosi / epätosi	Jos tulo x on pienempi kuin tulo y , niin lähtö on tosi. Muissa tapauksissa lähtö on epätosi.
 tosi / epätosi	Jos tulo x on suurempi tai yhtä suuri kuin tulo y , niin lähtö on tosi. Muissa tapauksissa lähtö on epätosi.
 tosi / epätosi	Jos tulo x on pienempi tai yhtä suuri kuin tulo y , niin lähtö on tosi. Muissa tapauksissa lähtö on epätosi.
 tosi / epätosi	Jos tulo x on 0, niin lähtö on tosi. Muissa tapauksissa lähtö on epätosi.
 tosi / epätosi	Jos tulo x on erisuuri kuin 0, niin lähtö on tosi. Muissa tapauksissa lähtö on epätosi.
 tosi / epätosi	Jos tulo x on suurempi kuin 0, niin lähtö on tosi. Muissa tapauksissa lähtö on epätosi.
 tosi / epätosi	Jos tulo x on pienempi kuin 0, niin lähtö on tosi. Muissa tapauksissa lähtö on epätosi.
tosi  epätosi	Invertteri, muuttaa tiedon tosi epätodeksi, ja päinvastoin (boolean).
	Vaihtelee tulotietoja x ja y , riippuen siitä, onko ohjaustieto s , tosi vai epätosi. Jos s on tosi, lähtö on x ja jos s on epätosi, lähtö on y .
	Lukunopeus, esim. silmukan sisällä olevat ohjelmat luetaan 100 ms välein.
	JA-veräjä, jos x ja y ovat samoja, lähtö on tosi (ja-portti). Voidaan käyttää joko boolean (tosi, epätosi) tai numeroarvoja (0, 1).



TAI-veräjä, jos jompikumpi tuloista (x, y) toteutuu, niin lähtö on tosi. Voidaan käyttää joko boolean (tosi, epätosi) tai numeroarvoja (0, 1).



Lokaali muuttuja on funktion sisäpuolella esitelty muuttuja, jonka arvo on käytettävissä ja muunneltavissa ilman rajoituksia.



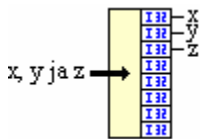
Globaali muuttuja on funktion ulkopuolella esitelty muuttuja, jonka arvo on käytettävissä ja muunneltavissa ilman rajoituksia.



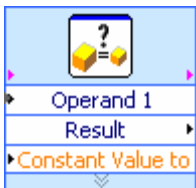
Joystick-ohjaimen liityntäasetukset (Initialize Joystick.vi).



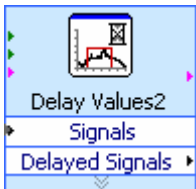
Muuntaa joystick-ohjaimelta tulevan tiedon ohjaukseen soveltuvaksi tiedoksi (Acquire Input Data.vi).



Jakaa ryhmässä tulevan tiedon yksittäisiin tietoihin (Unbundle.vi). Esimerkiksi, joystick-ohjaimen x, y ja z-akselien ryhmäohjaustieto eritellään yksittäisiksi tiedoiksi.



Toleranssi



Viivästetty signaali



Silmukan lukunopeus, silmukan sisällä olevat ohjelmat luetaan tietyin aseteltavin välein.



Näytteen otto. Esimerkiksi kameralta tulevista paikannustiedoista (m_x, m_y) otetaan näytteitä (m_{xv}, m_{yv}) 1s välein.



Takaisinkytkentä



Pelin tasojen ohjauksissa käytetyt rajatiedot (digitaaliset ja analogiset) saadaan DAQ-assistant-apuohjelman avulla pelin ohjauksen käyttöön.



Signaalin jakaminen.



Tiedonsiirto johto, ryhmätieto (9 kpl true-false tietoa), vaalean punainen.



Viivytetty signaali, sininen.



Tiedonsiirto johto, boolean-tieto (true / false), vihreä.



Tiedonsiirto johto, 64-bittinen liukuluku, oranssi.



Tiedonsiirto johto, 32-bittinen kokonaisluku, sininen.