

LAPPEENRANNAN TEKNILLINEN YLIOPISTO

LUT School of Energy Systems

LUT Kone

BK10A0402 Kandidaatintyö

ÄLYKKÄÄN KALASTUSVIEHEEN SUUNNITTELU

DESIGN OF AN INTELLIGENT FISHING LURE

Lappeenrannassa 10.6.2018

Janne Martikainen

Tarkastaja TkT Heikki Handroos

Ohjaajat TkT Lauri Luostarinen

TkT Hamid Roozbahani

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
LUT Energiajärjestelmät
LUT Kone

Janne Martikainen

ÄLYKKÄÄN KALASTUSVIEHEEN SUUNNITTELU

Kandidaatintyö

2018

40 sivua, 17 kuvaa, 1 taulukko ja 2 liitettä

Tarkastaja: TkT Heikki Handroos

Ohjaajat: TkT Lauri Luostarinen
TkT Hamid Roozbahani

Hakusanat: kalastus, älykäs, viehe

Kalastus on yksi ihmiskunnan vanhimpia elinkeinoja metsästyksen ja keräilyn ohella. Viehekalastuksen alkaminen on ajoitettu aikaan ennen metallinkäsittelyn keksimistä. Nykyään viehemarkkinat laajenevat jatkuvasti ja markkinoilla on jo erilaisia älyvieheitä.

Työn tavoitteena on kehittää 3D-tulostettava älykäs viehe, joka pystyy värisemään satunnaisesti, mittaamaan uintisyvyytensä ja etäisyytensä pohjasta ja säätämään uintisyvyyttään itsenäisesti. Syvyyssäätötoiminnolle ideoidaan erilaisia vaihtoehtoja, joissa vaikuttavat hydrodynaamiset voimat ja vaadittavien komponenttien valintakriteerit selvitetään kirjallisuutta tutkimalla.

3D-tulostetulla manuaalisella prototyypillä tehtyjen testien perusteella valitaan lupaavimmalta vaikuttava syvyyssäätömekanismi. Mekanismin komponenttien vaatimukset mitoitetaan. Syvyyssäätöön ja lisätoimintoihin etsitään soveltuvat komponentit ja toteutetaan komponenttien kytkennät. Lopuksi kirjoitetaan toiminnot toteuttava ohjelma.

Prototyyppiä ei ehditty viimeistellä ja lopullista runkorakennetta ei suunniteltu. Prototyypin toimivuutta kalastaessa ei voitu varmistaa.

ABSTRACT

Lappeenranta University of Technology
LUT School of Energy Systems
LUT Mechanical Engineering

Janne Martikainen

DESIGN OF AN INTELLIGENT FISHING LURE

Bachelor's thesis

2018

40 pages, 17 figures, 1 table and 2 appendices

Examiner: D. Sc. (Tech.) Heikki Handroos

Supervisors: D. Sc. (Tech.) Lauri Luostarinen
D. Sc. (Tech.) Hamid Roozbahani

Keywords: fishing, intelligent, lure

Fishing is one of mankind's oldest practices for gathering food in addition to hunting and gathering. Lure fishing has been dated to have begun before metallurgy was invented. Nowadays lure markets are expanding constantly and there are already different kinds of smart lures on the market.

The goal of this thesis is to develop a 3D printed intelligent lure that can vibrate randomly, measure its swimming depth and distance from bottom and adjust its swimming depth independently. Different ideas for the depth adjustment method are developed, the hydrodynamic forces acting in them and selection criteria for the components are researched from literature.

Based on tests done with a 3D printed manual prototype, the most promising depth adjusting mechanism is selected. The requirements for the components needed for the mechanism are calculated. Fitting components for the depth adjustment and additional features are searched and the wiring is done for the components.

There wasn't enough time to finish the prototype and no finalized body was designed. Functionality of the prototype while fishing could not be verified.

SISÄLLYSLUETTELO

TIIVISTELMÄ

ABSTRACT

SISÄLLYSLUETTELO

SYMBOLI- JA LYHENNELUETTELO

1	JOHDANTO	8
2	VIEHEIDEN HISTORIA	9
	2.1 Vaaput.....	9
	2.2 Älyvieheet.....	10
3	SYVYYSSÄÄTÖMEKANISMIIEN MITOITUS JA LISÄTOIMINNOT	13
	3.1 Syvyyssäädön mahdolliset toteutustavat	13
	3.1.1 Käännettävä nokkalevy.....	13
	3.1.2 Painopisteen säätö.....	18
	3.1.3 Painon säätö	21
	3.1.4 Nokkalevyn pituuden säätö.....	23
	3.1.5 Siiman kiinnityspisteen säätö.....	24
	3.2 Lisätoiminnot	25
	3.2.1 Syvyyden mittaus.....	25
	3.2.2 Epäsäännöllinen värinä	25
	3.2.3 Mikrokontrolleri.....	26
4	MANUAALISET TESTIT	27
	4.1 Manuaalinen prototyyppi.....	27
	4.2 Testien tulokset	28
5	VALITUT KOMPONENTIT	30
	5.1 Syvyyssäätömekanismin komponentit.....	30
	5.2 Mikrokontrolleri.....	30
	5.3 Paine- ja ultraäänianturit	31
	5.4 Värinämoottori, transistori ja vastus	31
	5.5 Jännitelähde	32
6	PROTOTYYPIN RAKENTAMINEN	33
	6.1 KytKentä.....	33

6.2 Ohjelmointi	34
7 JATKOKEHITYS	37
LÄHTEET	38
LIITTEET	

Liite I: Kytkentäkaavio

Liite II: Ohjelma

SYMBOLI- JA LYHENNELUETTELO

A	Pinta-ala [m ²]
a	Nostevoiman kohtisuora etäisyys kiinnityspisteestä [m]
b	Vastusvoiman kohtisuora etäisyys kiinnityspisteestä [m]
c	Painekeskipisteen etäisyys kiinnityspisteestä [m]
C_D	Vastuskerroin
C_L	Nostekerroin
d	Halkaisija [m]
d_m	Jakohalkaisija [m]
F_A	Aksiaalivoima [N]
F_D	Vastusvoima [N]
F_H	Hydrostaattinen voima [N]
F_L	Nostevoima [N]
F_N	Normaalivoima [N]
F_R	Resultanttivoima [N]
G	Vieheen paino [N]
g	Putoamiskiihtyvyys [m/s ²]
h	Nokkalevyn korkeus [m]
\bar{h}	Nokkalevyn keskipisteen etäisyys vedenpinnasta [m]
h^*	Hydrostaattisen voiman vaikutuspisteen etäisyys vedenpinnasta [m]
h_{FE}	Virtavahvistuskerroin
I	Virta [A]
I_b	Kantavirta [A]
I_c	Kollektorivirta [A]
I_g	Nokkalevyn jäyhyysmomentti [m ⁴]
i	Hammasvaihteen välityssuhde
K	Nokkalevyn kiinnityspiste
L	Hydrostaattinen noste [N]
l	Kierteen nousu [m]
M_D	Vastusvoiman aiheuttama momentti [Nm]
M_H	Hydrostaattisen voiman aiheuttama momentti [Nm]

M_K	Kiinnityspisteessä vaikuttava momentti [Nm]
M_L	Nostevoiman aiheuttama momentti [Nm]
p	Hydrostaattinen paine [Pa]
p_k	Kierteiden harjojen välinen etäisyys [m]
R	Resistanssi [Ω]
r	Hydrostaattisen voiman vaikutuspisteen etäisyys kiinnityspisteestä
s	Nokkalevyn leveys [m]
T_m	Moottorin vääntömomentti [Nm]
U	Jännite [V]
U_m	Mikrokontrollerin antama jännite [V]
U_v	Vastuksen jännitehäviö [V]
V	Veden virtausnopeus [m/s^2]
$V_{BE(sat)}$	Kanta-emitterijännite kyllästytilassa [V]
W	Kuorma [N]
α	Nokkalevyn kulma [$^\circ$]
α_k	Kierteen kulma [$^\circ$]
β	Resultanttivoiman ja vastusvoiman välinen kulma [$^\circ$]
μ	Kitkakerroin
ρ	Veden tiheys [kg/m^3]
AR	Aspect Ratio, muotosuhde
COP	Center of Pressure, painekeskipiste

1 JOHDANTO

Perinteisillä kalastusvieheillä on yleensä jokin tietty uintisyvyys, johon voidaan kalastuksen aikana vaikuttaa vain vetonopeutta muuttamalla. Suuremman syvyyden saavuttamiseksi vetonopeutta voidaan kuitenkin joutua kasvattamaan niin paljon, että viehe liikkuu liian nopeasti eivätkä kalat kiinnostu siitä. Automaattisella syvyyssäädöllä varustetulla vieheellä pystyttäisiin muuttamaan uintisyvyyttä vedon aikana vetonopeuden pysyessä vakiona. Muita lisätoimintoja lisäämällä voidaan kasvattaa vieheen houkuttelevuutta tai antaa sille kyky tarkkailla ympäristöään, jolloin se pystyy esimerkiksi välttämään pohjaan tarttumisen.

Markkinoilla on jo lisätoiminnoilla, kuten välkkyvillä ledeillä ja pakenevaa kalaa matkivalla värinällä, varustettuja vieheitä. Automaattisella syvyyssäädöllä tai useilla eri toiminnoilla varustettuja ei kuitenkaan ole. Älykkäälle vieheelle on siis olemassa mahdollinen markkinarako.

Tämän työn tavoitteena on suunnitella älykäs kalastusviehe, jolla on automaattinen syvyyssäätö ja epäsäännöllinen värinätoiminto. Tarkoituksena on vastata siihen, onko automaattinen syvyyden säätö mahdollista toteuttaa kalastusvieheellä. Työ toteutetaan selvittämällä vieheeseen vaikuttavat hydrodynaamiset voimat alan kirjallisuutta tutkimalla ja ideoimalla erilaisia säätömekanismeja. Mekanismeista valitaan kaksi lupaavimmalta vaikuttavaa ja niille tehdään käytännön kokeiluja mekaanisesti säädettävällä prototyypillä. Kokeilujen perusteella valitaan mekanismeista se, jolla on suurin vaikutus uintisyvyyteen. Käytännön kokeilut jokaiselle eri mekanismille rajataan työstä pois ajan ja resurssien puutteen vuoksi. Lisäksi tutkitaan vieheelle mahdollisia lisätoimintoja.

Laskettujen hydrodynaamisten voimien perusteella mitoitetaan ja etsitään valitun mekanismin toteuttamiseen vaadittavat komponentit, joista rakennetaan toimiva prototyyppi.

2 VIEHEIDEN HISTORIA

Kalastus on metsästyksen ja keräilyn ohella ihmiskunnan vanhimpia elinkeinoja. Ensimmäisten ihmisten kalastusmenetelmistä ei ole varmaa tietoa, mutta vanhimmat löydetyt koukut on ajoitettu aikaan 30 000 eKr. Onkia on käytetty jo 4000 vuotta sitten Egyptissä. (Hyytinen & Hyytinen 2004, s. 5.)

Viehekalastuksessa ideana on matkia vieheellä kalojen luonnollista ravintoa. Simpukankuoret, joihin on kiinnitetty koukkuja, ovat varhaisimpia uistinmalleja. Kiinteäkoukkuisten lusikkauistimien valmistus aloitettiin metallin keksimisen jälkeen. Suomessa on löydetty Rautalammilta Konneveden saaresta kiinteäkoukkuisia kupariuistimia, jotka on ajoitettu 1200–1300 -luvulle. Pyörivät uistimet, kuten lipat, on keksitty noin 200 vuotta sitten. (Hyytinen & Hyytinen 2004, s. 5.)

Nykyään on olemassa monia erilaisia vieheitä, kuten esimerkiksi edellä mainittujen lusikkauistimien ja lippojen lisäksi jiggit, vaaput ja jerkit. Tässä työssä keskitytään vaapun kehittälemiseen.

2.1 Vaaput

Vaappu on pikkukalaa muistuttava viehe, jonka nokassa on uintilevy joka määrää vaapun uintisyvyyden ja -liikkeen. Nykyään yleisin vaappujen valmistukseen käytettävä materiaali on kevyt balsapuu. Läpinäkyvä muovi on vakiintunut uintilevyn materiaaliksi, mutta suuremmille kaloille tarkoitettujen vaappujen uintilevy valmistetaan pellistä. (Hyytinen & Hyytinen 2004, s. 77.)

Yleisen teknisen kehityksen myötä tapahtui myös uistinten kehittelyä Euroopassa ja Yhdysvalloissa 1800-luvun puolivälissä, jolloin uistimia alettiin valmistaa metallin lisäksi myös kumista ja keinomuoveista. Kelluvista materiaaleista, kuten puusta ja kaarnasta, valmistetut vaaput keksittiin vasta 1800-luvun lopulla. Yhdysvaltalainen James Heddon aloitti kokeilut puu-uistinten valmistuksesta vuonna 1894. Vuonna 1902 Heddon patentoi keksintönsä ja perusti vieheitä valmistavan Heddon-yhtiön. Suomessa amerikkalaisten puuvaappujen laajempi markkinointi aloitettiin 1920-luvulla. (Hyytinen 2014, s. 4.)

Suomalainen innovaatio vaappujen valmistuksessa oli rungon sisälle upotettava runkolanka, josta taivutettuihin lenkkeihin koukut voitiin kiinnittää. Runkolangan ansiosta vaapuista pystyttiin tekemään kevyempiä ja paremmin uivia kuin ennen. Sen lisäksi myös kuka tahansa pystyi valmistamaan vaappuja. Esimerkiksi vanhoissa amerikkalaisissa vaapuissa koukut kiinnitettiin prässätyillä heloilla ja ruuveilla, mikä vaati erikoiskoneita. Ohut teräslanka sen sijaan on helposti saatavissa ja työstettävissä. Runkolangan keksijäksi voidaan ajoittaa Karl Östring, joka aloitti 1920–30 -luvulla kokeilut korkkivaapuilla. Vaappuihin laitettiin pituussunnassa rungon läpi keskeltä lanka, jonka päihin tehtiin lenkit siimalle ja koukulle. Vaapun mahan alle tulevalle koukulle tehtiin lenkki erillisestä langasta. Yhtenäisen runkolangan kehittivät Kustaa ja Ville Tuulos sekä Martti Palomäki. (Hyytinen 2014, s. 32.)

Suomalaiset keksivät myös laittaa uintilevyn vaapun leuan alle, kun amerikkalaisissa vaapuissa se oli sijoitettu vieheen nokkaan. Tämä mahdollisti uintiliikkeen monipuolisemman säädön ja vaappuun saatiin sekä pituus- että sivuttaissuuntaista liikettä. Suomalaisen vaappukehityksen voidaan katsoa huipentuneen Rapalan Original-vaappuun, joka saavutti suuren suosion Amerikassa ja loi pohjan suomalaisten vaappujen menestykselle maailmalla. (Hyytinen 2014, s. 5.)

2.2 Älyvieheet

Markkinoilla on jo olemassa erilaisilla toiminnoilla varustettuja älyvieheitä. Toiminnot vaihtelevat yksinkertaisesta LED-valaistuksesta vieheen värin vaihtamiseen veden lämpötilan mukaan.

Balista on vuonna 2008 perustettu australialainen yritys, joka valmistaa LED-vieheitä. Yrityksen tekemien testien mukaan punaisen valon on havaittu olevan kaikista tehokkain kalojen houkuttelemisessa. Tästä syystä kaikissa yrityksen vieheissä on punainen LED. Myös valon välkkyminen tehostaa vaikutusta. Vieheiden pyrstössä oleva LED syttyy kun viehe lasketaan veteen ja sammuu kun se nostetaan vedestä. Vieheet ovat paristokäyttöisiä, mutta paristoa ei voi vaihtaa. (Balista 2018.)



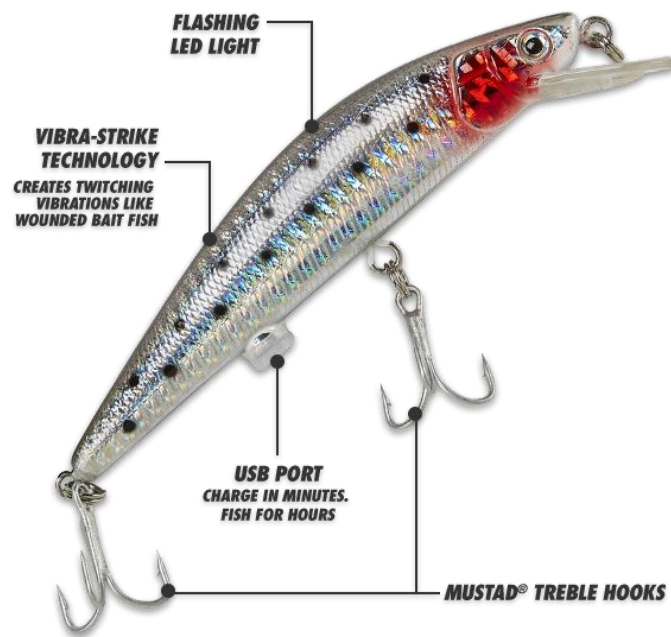
Kuva 1. Balista Trigger. (Balista 2018)

eMinnow valmistaa paristokäyttöisiä motorisoituja vieheitä. Ne ovat suunniteltu elävien täykalojen korvaamiseen. Vieheissä on mikrokontrollerin ohjaama pienikokoinen sähkömoottori, joka väräyttää vieheen pyrstöä. Vieheelle saadaan näin aikaiseksi oikeaa kalaa muistuttava uintiliike. Moottori on ohjelmoitu värisemään epäsäännöllisesti oikean kalan simuloimiseksi. Moottori saa myös aikaan naksuttavan äänen, joka valmistajan mukaan houkuttelee petokaloja. Viehe käynnistyy kun se lasketaan veteen ja sammuu kun se nostetaan vedestä. (eMinnow 2016.)



Kuva 2. eMinnow FatBobba. (eMinnow 2016)

Rechargeable Twitching Lure on Isossa-Britanniassa kehitetty viehe, jossa on yhdistetty LED- ja värinätoiminnot. Vieheessä on välkkyvä LED-valo, joka kiinnittää petokalojen huomion myös sameassa vedessä tai huonoissa valaistusolosuhteissa ja sen värinä simuloi haavoittunutta saaliskalaa. Viehe käynnistyy vedessä. Viehe voidaan ladata USB-liitännän avulla. (Twitching Lure 2018.)



Kuva 3. Rechargeable Twitching Lure. (Twitching Lure 2018)

Smartbaits on kehittänyt vieheisiinsä veden lämpötilan mukaan vaihtuvan värityksen. Vieheet soveltuvat kalastukseen vaihtelevissa olosuhteissa. Tekniikkaa sovelletaan sekä vaappuihin että pehmovieheisiin, kuten jigeihin. Valmistajan mukaan värin vaihtelu imitoi oikeiden saaliskalojen käyttäytymistä. (Smartbaits.)



Kuva 4. Clever Crank 55. (Smartbaits)

3 SYVYYSSÄÄTÖMEKANISMIIEN MITOITUS JA LISÄTOIMINNOT

Sopivien komponenttien etsintää varten täytyi ensin selvittää vieheeseen eri toteutustavoilla kohdistuvat hydrodynaamiset voimat. Voimat laskettiin kirjallisuudesta etsittyjen kaavojen ja perusteiden mukaisesti. Laskettujen voimien kestämisen lisäksi on tärkeää pitää komponenttien fyysinen koko mahdollisimman pienenä. Myös käyttöjännite ja virrankulutus tulisi pyrkiä minimoimaan.

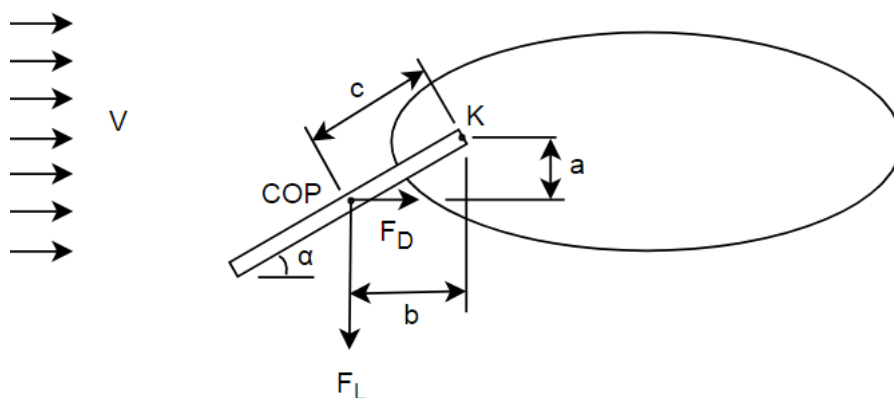
3.1 Syvyyssäädön mahdolliset toteutustavat

Syvyyden säätämistä varten ideoitiin viisi erilaista toteutustapaa: nokkalevyn kulmaa, vieheen painopistettä, painoa, nokkalevyn pituutta tai siiman kiinnityspisteen sijaintia säätämällä. Mitoitusperusteet tutkittiin jokaiselle tapaukselle.

3.1.1 Käännettävä nokkalevy

Nokkalevyn kääntämiseen vaadittavan mekanismin määrittämiseksi on tiedettävä nokkalevyn vaikuttava maksimivoima ja sen aiheuttama momentti. Kun momentti on tiedossa, voidaan etsiä sopiva hammasvaihe ja riittävällä väännöllä varustettu askelmoottori.

Kuvassa 5 esitellään virtauksen nokkalevyn aiheuttamat nostevoima F_L ja vastusvoima F_D . Molemmat voimat vaikuttavat nokkalevyn painokeskipisteessä COP (Center of Pressure). Nokkalevy on kiinnitetty pisteeseen K .



Kuva 5. Virtauksen nokkalevyn kohdistamat voimat F_L ja F_D .

Nostevoima on virtauksen aiheuttaman voiman virtauksen suuntaan kohtisuora voimakomponentti, vastusvoima puolestaan virtauksen suunnan mukainen. (Newman 1977, s. 20–21). Nokkalevyn asennon vuoksi nostevoima kohdistuu alaspäin ja täten nostevoimaa kasvattamalla viehe saadaan uimaan syvemmällä. Nokkalevyn kulmaa muuttamalla pyritään siis joko kasvattamaan tai pienentämään nostevoimaa, riippuen halutusta syvyydestä. Nostevoima F_L ja vastusvoima F_D voidaan laskea seuraavilla kaavoilla (Bansal 2005, s. 412):

$$F_L = \frac{1}{2} C_L A \rho V^2 \quad (1)$$

$$F_D = \frac{1}{2} C_D A \rho V^2 \quad (2)$$

Yhtälöissä C_L on nostekerroin, A referenssipinta-ala, ρ veden tiheys, V veden virtausnopeus ja C_D vastuskerroin. Veden virtausnopeutena toimii tässä tapauksessa vieheen vetonopeus. Kertoimet C_L ja C_D riippuvat kappaleen kulmasta ja geometriasta.

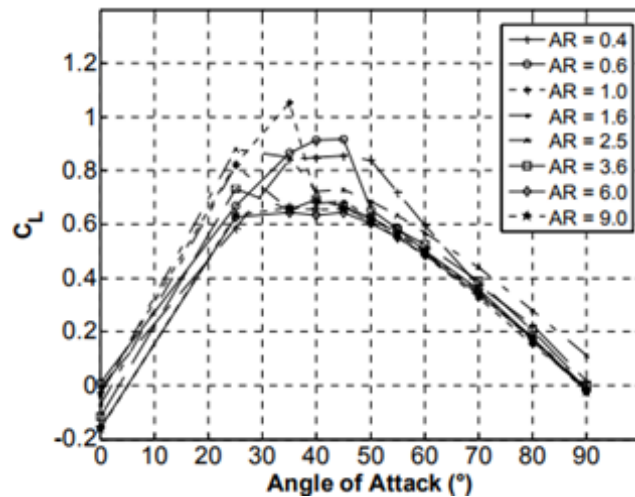
Vieheen rungon vaikutus kertoimiin jätetään tässä huomioimatta ja nokkalevyä ajatellaan litteänä levynä. Nostekerroin käyttäytyy lineaarisesti pienillä kulmilla, mutta suurilla kulmilla tapahtuu sakkausta ja käytös muuttuu epälineaariseksi (Newman 1977, s. 172). Nostekertoimen arvosta riippuen suuretkin kulmat ovat haluttuja, joten kertoimelle on etsittävä kokeellisesti selvitettyjä arvoja.

Oikean nostekertoimen löytämiseksi on ensin laskettava nokkalevyn muotosuhde AR (Aspect Ratio), joka saadaan kaavalla

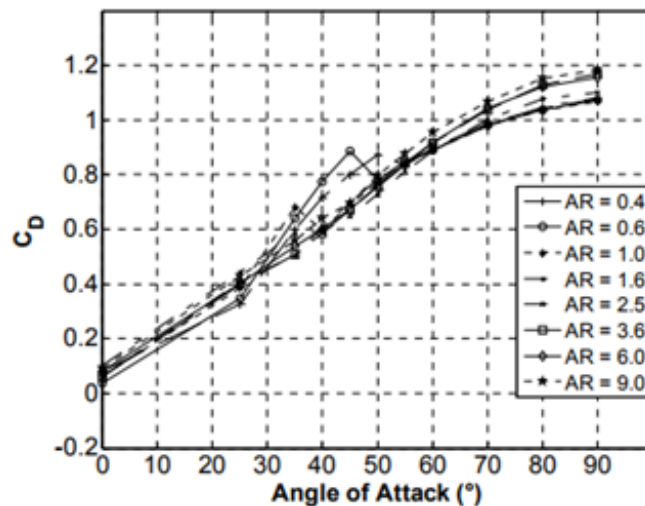
$$AR = \frac{s^2}{A} \quad (4)$$

jossa s on levyn leveys ja A pinta-ala (Faltinsen 2006, s. 32). Kuvissa 6 ja 7 esitetään litteiden levyjen nostekertoimelle ja vastuskertoimelle kokeellisesti laskettuja arvoja korkeilla kulmilla. Tarkasteltava käyrä valitaan halutun mittaisen nokkalevyn muotosuhteen mukaan. Kun muotosuhde on tiedossa, valitaan kuvaajasta suurin nostekerroin kyseisellä muotosuhteella. Koska kuvaajan mukaan nostekerroin pienenee

saatuaan ääriarvonsa, nokkalevyn maksimikulmaksi valitaan kulma, jolla ääriarvo saavutetaan. Vastuskerroin valitaan samalla kulmalla kuin nostekerroinkin. Kertoimien laskemisessa referenssipinta-alana on käytetty levyn pinta-alaa (Ortiz et al. 2015, s. 2443–2444). Noste- ja vastusvoiman laskemisessa käytetään siis nokkalevyn pinta-alaa.



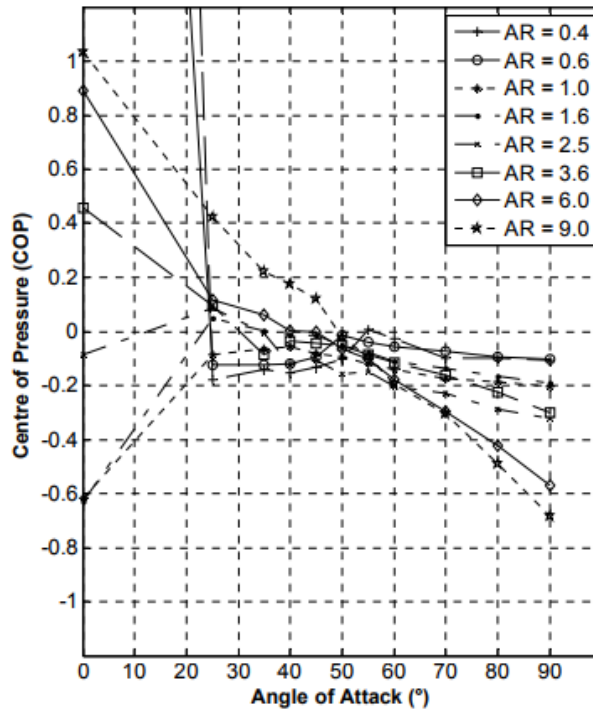
Kuva 6. Nostekerroin C_L levyllä muotosuhteen AR ja kulman α funktiona (Ortiz et al. 2015, s. 2444).



Kuva 7. Vastuskerroin C_D levyllä muotosuhteen AR ja kulman α funktiona (Ortiz et al. 2015, s. 2444).

Kuvassa 8 esitetään levyn paineakeskipisteelle kokeellisesti laskettuja arvoja. Kuvaajan pystyakseli kuvaa paineakeskipisteen sijaintia siten, että luku -1 merkitsee levyn etureunaa, 0 levyn keskipistettä ja 1 levyn takareunaa (Ortiz et al. 2015, s. 2446). Paineakeskipisteen

sijainti valitaan nostekertoimen mukaisesti saadun maksimikulman ja muotosuhteen mukaan.



Kuva 8. Levyn painekeskkipiste muotosuhteen AR ja kulman α funktiona (Ortiz et al. 2015, s. 2447).

Nostevoiman ja vastusvoiman kiinnityspisteeseen K aiheuttamat momentit M_L ja M_D saadaan kaavoilla

$$M_L = F_L a \quad (5)$$

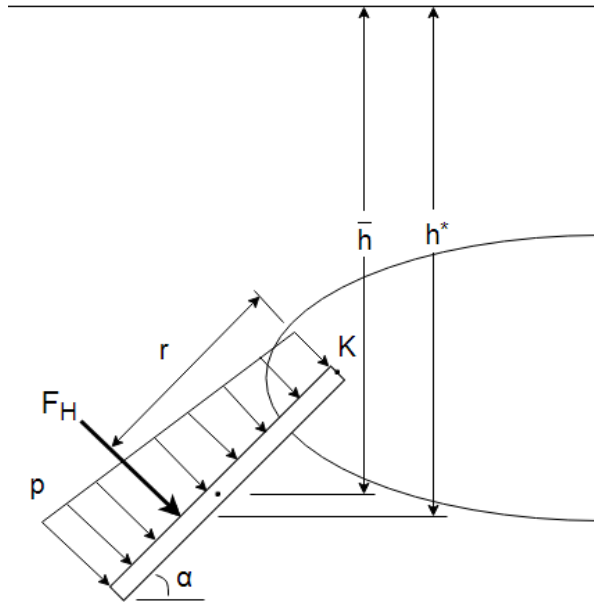
$$M_D = F_D b \quad (6)$$

jossa a on painekeskkipisteen vaakasuuntainen ja b pystysuuntainen etäisyys kiinnityspisteestä. Trigonometrian perusteella

$$a = c \sin(\alpha) \quad (7)$$

$$b = c \cos(\alpha) \quad (8)$$

Seuraavaksi selvitetään maksimikulmassa nokkalevyyn vaikuttava hydrostaattisen paineen aiheuttava voima F_H . Kuvassa 9 esitetään hydrostaattisen paineen jakauma nokkalevyn pinnalla ja hydrostaattinen voima F_H .



Kuva 9. Hydrostaattinen paine p ja voima F_H nokkalevyn pinnalla.

Hydrostaattinen voima voidaan laskea kaavalla

$$F_H = \rho g A \bar{h} \quad (9)$$

jossa ρ on veden tiheys, g putoamiskiihtyvyys, A levyn pinta-ala ja \bar{h} levyn keskipisteen etäisyys vedenpinnasta (Bansal 2005, s. 87). Painopisteen etäisyytenä käytetään päätettyä maksimiuintisyvyyttä. Hydrostaattisen voiman suunta on kohtisuoraan levyn pinnan kanssa. Voiman vaikutuspisteen syvyys h^* puolestaan saadaan kaavalla

$$h^* = \frac{I_g \sin^2(\alpha)}{A \bar{h}} + \bar{h} \quad (10)$$

jossa I_g on levyn jäyhyysmomentti (Bansal 2005, s. 88). Suorakulmion jäyhyysmomentti lasketaan kaavalla

$$I_g = \frac{sh^3}{12} \quad (11)$$

jossa s on levyn leveys ja h levyn korkeus. Kun vaikutuspisteen syvyys tiedetään, voidaan voiman etäisyys r kiinnityspisteestä ratkaista trigonometrian perusteella kaavalla

$$r = \frac{\frac{h}{2} \sin(\alpha) + \frac{(h^* - \bar{h})}{\sin(\alpha)}}{\sin(\alpha)} \quad (12)$$

Hydrostaattisen voiman aiheuttama momentti M_H pisteessä K on tällöin

$$M_H = F_H r \quad (13)$$

Pisteessä K vaikuttava kokonaismomentti M_K saadaan laskemalla yhteen kaavoilla (5), (6) ja (13) ratkaistut momentit

$$M_K = M_L + M_D + M_H \quad (14)$$

Moottorille välittyvää momenttia on saatava vähennettyä, jotta valittava moottori olisi fyysiseltä kooltaan mahdollisimman pieni. Tähän tarkoitukseen käytetään hammasvaihdetta. Moottorilta vaadittava momentti saadaan kaavalla

$$T_m = \frac{M_K}{i} \quad (15)$$

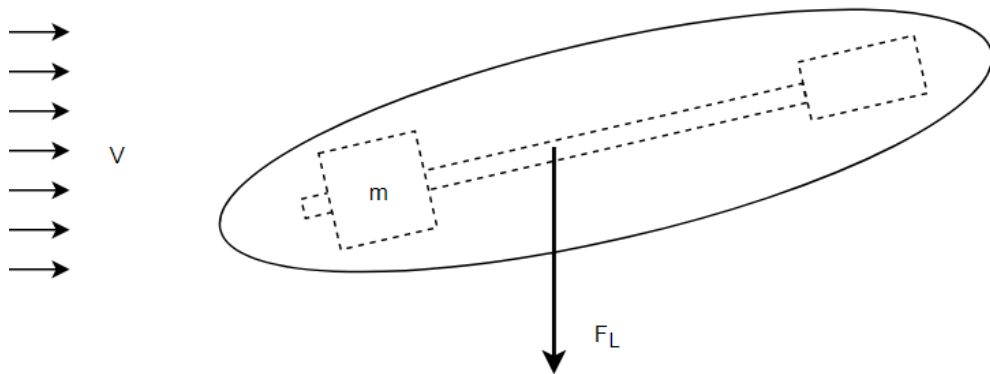
jossa T_m on askelmoottorin vääntö ja i hammasvaihteen välityssuhde. Hammasvaihteen olisi hyvä olla myös itselukittuva, jolloin nokkalevy kääntyy vain askelmoottorin voimasta eikä veden aiheuttamista voimista. Moottorin voi tällöin sammuttaa, kun nokkalevyn asentoa ei tarvitse muuttaa. Virrankulutus saadaan näin pidettyä pienenä.

Edellä olevia kaavoja käyttämällä ja iteroimalla saadaan valittua vieheelle sopiva askelmoottori ja hammasvaihde.

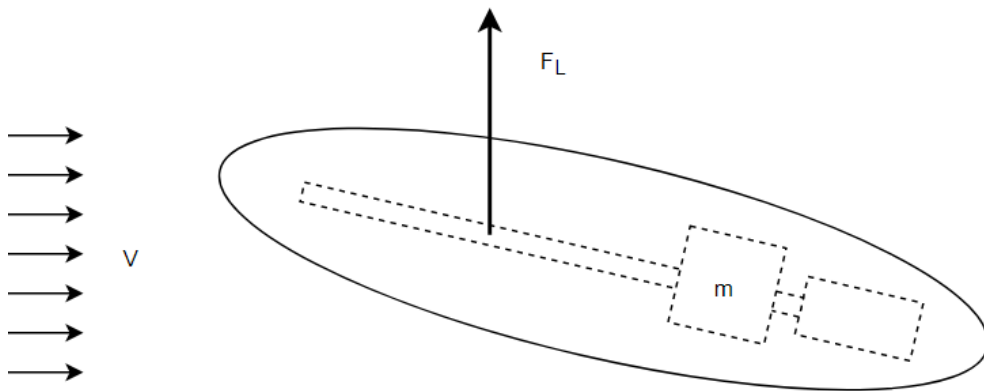
3.1.2 Painopisteen säätö

Vieheen painopisteen säädöllä on samanlainen toimintaperiaate kuin nokkalevyn käännöllä, mutta nyt nostevoima kohdistuu vieheen runkoon. Siirtämällä vieheen painopiste eteen kuvan 10 mukaisesti saadaan sen nokka kääntymään alaspäin, jolloin

nostevoimankin suunta on alaspäin. Vieheen muoto ja kulma määräävät nostekertoimen. Takapainoisena nostevoiman suunta on ylöspäin kuten kuvassa 11. Painopisteen säätö tapahtuu täysin vieheen sisäisesti eivätkä hydrodynaamiset voimat vaikuta komponenttien mitoittamiseen. Mekanismin etuna on myös se, että siinä ei ole antureita lukuun ottamatta veden kanssa kosketuksissa olevia osia, joille pitäisi miettiä tiivistys.



Kuva 10. Etupainoiseen vieheeseen kohdistuva nostevoima.



Kuva 11. Takapainoiseen vieheeseen kohdistuva nostevoima.

Painopistettä säätäisi lineaarijohteella liikkuva kuorma. Lineaarijohteeksi valittiin askelmoottorin liikuttama ruuvijohde. Kuorman liikuttamiseen vaadittava vääntö saadaan kaavalla

$$T_m = \frac{Wd_m}{2} \tan(\phi + \alpha_k) \quad (16)$$

jossa W on kuorma, d_m ruuvijohteen jakohalkaisija, ϕ kitkakulma ja α_k kierteen kulma (Bhandari 2010, s. 189). Jakohalkaisija saadaan kaavalla

$$d_m = d - 0,5p_k \quad (17)$$

jossa d on ruuvijohteen halkaisija ja p_k kierteen harjojen välinen etäisyys (Bhandari 2010, s. 188).

Kuormasta aiheutuva voima lasketaan kaavalla

$$W = F + mg \quad (18)$$

jossa F on ulkoinen voima ja m on kuorman massa. Tässä tapauksessa ulkoista voimaa ei ole ja ruuvijohteessa vaikuttaa vain massan paino.

Kierteen kulma voidaan myös ilmaista muodossa

$$\tan(\alpha_k) = \frac{l}{\pi d_m} \quad (19)$$

jossa l on kierteen nousu (Bhandari 2010, s. 188). Kitkakulman avulla voidaan laskea kitkakerroin kaavalla

$$\mu = \tan(\phi) \quad (20)$$

jossa μ on kitkakerroin (Bhandari 2010, s. 189).

Taulukossa 1 esitellään erinäisten materiaalien keskinäisiä kitkakertoimia. Kun haluttu ruuvijohde on valittu, otetaan laskennassa käytettäväksi kitkakertoimeksi taulukosta ruuvijohteen ruuvin sekä mutterin materiaalin mukainen arvo.

Taulukko 1. Materiaalien keskinäisiä kitkakertoimia (Gope 2012, s. 502).

Ruuvin materiaali	Mutterin materiaali			
	Teräs	Pronssi	Messinki	Valurauta
Teräs, kuiva	0,15–0,25	0,15–0,23	0,15–0,19	0,15–0,25
Teräs, öljytty	0,11–0,17	0,10–0,16	0,10–0,15	0,11–0,17
Pronssi	0,08–0,12	0,04–0,06	–	0,06–0,09

Takaperin liikuttamiseen tarvittava vääntö saadaan kaavalla (Bhandari 2010, s. 190)

$$T_m = \frac{Wd_m}{2} \tan(\phi - \alpha_k) \quad (21)$$

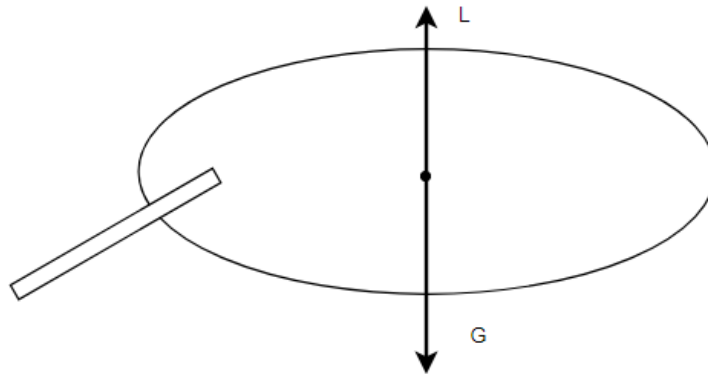
Tarvittava askelmoottori valitaan kaavan (16) mukaan lasketun väännön mukaisesti. Kuorman paino valitaan niin, että se on moottorin painoa suurempi. Tällöin varmistetaan painopisteen siirtyminen vieheen etu- tai takapuolelle kuorman ja moottorin ollessa eri puolilla viehettä. Jos kuorma ja moottori ovat samanpainoisia, on riskinä että painopiste ei lainkaan siirry vieheen toiselle puolelle. Moottorille etsitään käännettävän nokkalevyn mekanismin tapaisesti sopiva ohjain.

Kuorman ääripisteisiin tarvitaan rajakytkimet tarkkailemaan sen sijaintia.

3.1.3 Painon säätö

Sukellusveneen tavoin vieheen painoa muuttamalla on mahdollista säätää sen uintisyvyyttä. Veteen upotettuun vieheeseen vaikuttaa kuvan 12 mukainen hydrostaattinen noste. Arkhimedeiden lain mukaan veteen upotettuun kappaleeseen kohdistuva hydrostaattinen noste on yhtä suuri kuin sen syrjäyttämän veden paino (Giordano 2012, s. 328). Täten tiheyttä kasvattamalla pyritään saamaan vieheen paino suuremmaksi kuin vieheeseen kohdistuva noste, jolloin viehe alkaa vajota syvemmälle. Tiheyttä pienentämällä painosta saadaan nostetta pienempi, jolloin viehe alkaa kohota. Tiheyden kasvattaminen toteutettaisiin pumppaamalla vieheen sisälle vettä.

Mekanismi toteutettaisiin kahdella vesipumpulla, joista toinen pumpkaa ympäröivää vettä vieheen sisällä olevaan vesisäiliöön ja toinen pumpkaa sitä ulos. Vieheen runkoa suunniteltaessa on otettava huomioon komponenttien ja rungon paino, jotta sen sisälle tulevan vesisäiliön tilavuus saadaan mitoitettua oikein. Vieheen painon tulisi olla tyhjällä säiliöllä pienempi kuin siihen kohdistuva noste.



Kuva 12. Vieheeseen kohdistuva hydrostaattinen noste.

Vesipumppuja ohjattaisiin kytkiminä toimivilla transistorilla. Transistorien valitsemiseksi on ensin tiedettävä pumppujen käyttämä jännite ja virta, sekä kuinka suuren virran mikrokontrolleri pystyy antamaan. Kun ne ovat tiedossa, etsitään transistori joka kestää vähintään nämä arvot. Sopivan transistorin löydyttyä lasketaan kantavirta, jolla se muuttuu täysin johtavaksi. Se saadaan kaavalla

$$I_b = \frac{I_c}{h_{FE}} \quad (22)$$

jossa I_b on transistorin kantavirta, I_c kollektorivirta ja h_{FE} virtavahvistuskerroin. Kollektorivirtana toimii ohjattavan pumpun virta. Virtavahvistuskertoimen arvo löytyy valitun transistorin teknisistä tiedoista.

Kyllästystilan, eli täyden johtavuuden, varmistamiseksi kantavirta halutaan kuitenkin laskettua suuremmaksi ja tähän tarkoitukseen transistorin kanssa sarjaan kytketään vastus. Kantavirta ei kuitenkaan saa olla mikrokontrollerin I/O-pinnien antamaa maksimivirtaa suurempi. Halutuksi kantavirraksi valitaan sopiva arvo lasketun kantavirran ja I/O-pinnin

maksimivirran väliltä, jonka jälkeen lasketaan tarvittun vastuksen resistanssi Ohmin lain mukaan kaavalla

$$R = \frac{U_v}{I_b} \quad (23)$$

jossa R on vastuksen resistanssi ja U_v vastuksen jännitehäviö. Jännitehäviö saadaan kaavalla

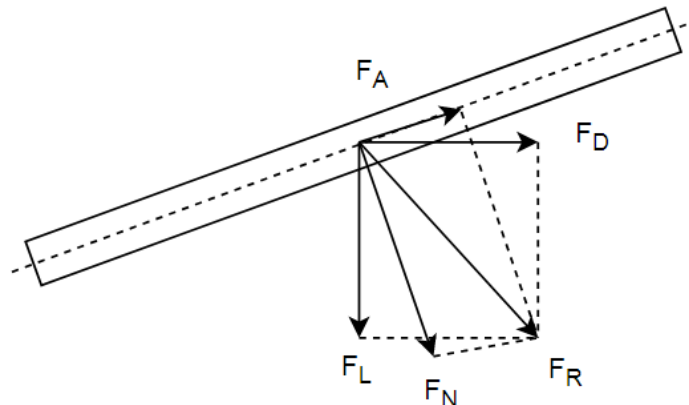
$$U_v = U_m - V_{BE(sat)} \quad (24)$$

jossa U_m on mikrokontrollerin antama jännite ja $V_{BE(sat)}$ transistorin kanta-emitterijännite kyllästystilassa, joka annetaan transistorin teknisissä tiedoissa.

3.1.4 Nokkalevyn pituuden säätö

Kaavasta (1) huomataan, että myös nokkalevyn pinta-alaa kasvattamalla saadaan nostevoima suuremmaksi. Käytännössä pinta-alan kasvattaminen toteutettaisiin vieheen ulkopuolisen levyosuuden pituutta kasvattamalla.

Nokkalevyn pituutta säädettäisiin askelmoottorin ja ruuvijohteen avulla. Kuvassa 13 esitellään nokkalevyyn vaikuttavien noste- ja vastusvoimien resultantti F_R sekä sen komponentit normaalivoima F_N ja aksiaalivoima F_A . Levy kulki vieheen runkoon tehtyä uraa pitkin, joka ottaisi vastaan levyyn kohdistuvista voimista aiheutuvat momentit. Tällöin askelmoottorin mitoittavana voimana toimii vain aksiaalivoima F_A



Kuva 13. Noste- ja vastusvoiman resultantti ja sen komponentit.

Resultanttivoima lasketaan kaavalla

$$F_R = \sqrt{F_L^2 + F_D^2} \quad (25)$$

Resultanttivoiman ja vastusvoiman välinen kulma saadaan kaavalla

$$\beta = \cos^{-1}\left(\frac{F_D}{F_R}\right) \quad (26)$$

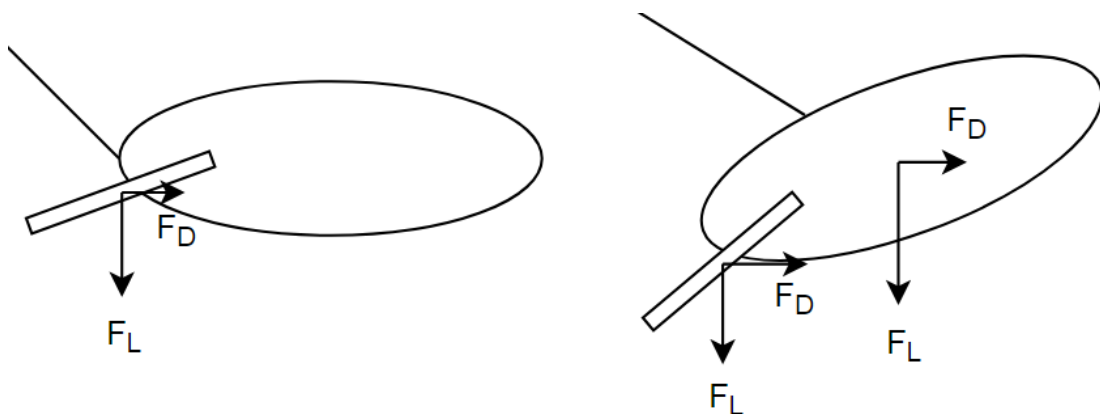
jossa β on resultanttivoiman ja vastusvoiman välinen kulma. Kun resultanttivoima ja kulma β ovat tiedossa, voidaan aksiaalivoima ratkaista kaavalla

$$F_A = F_R \cos(\alpha + \beta) \quad (27)$$

Moottorilta vaadittava vääntö lasketaan kaavoilla (16) ja (21). Nyt kuorman kaavassa (18) ulkoinen voima F on F_A ja paino mg on 0.

3.1.5 Siiman kiinnityspisteen säätö

Siiman kiinnityspisteen ollessa lähellä nokkalevyn paineakeskipistettä voidaan olettaa, että nokkalevyyn kohdistuvista voimista aiheutuva momentti ei saa viehettä kallistumaan. Siiman kiinnityspisteen sijaintia säätämällä muutetaan tätä momenttitasapainoa, jolloin viehe kallistuu kuvan 14 mukaisesti.



Kuva 14. Siimapisteen sijainnin vaikutus vieheen asentoon.

Toimintaperiaate on sama kuin säädettävän painopisteen tapauksessakin. Viehettä kallistamalla muutetaan nokkalevyn kulmaa ja myös runkoon saadaan kohdistumaan nostevoima, joka saa vieheen uppoamaan syvemmälle.

Mekanismi toteutettaisiin askelmoottorilla ja ruuvijohteella. Askelmoottorilta vaadittava vääntö lasketaan kaavoilla (16) ja (21). Kaavassa (18) käytettävä ulkoinen voima riippuu nokkalevyyn ja runkoon kohdistuvien noste- ja vastusvoimien ruuvijohteen suuntaisista voimakomponenteista. Nokkalevyn ja rungon noste- ja vastusvoimat lasketaan kaavoilla (1) ja (2). Rungon noste- ja vastuskertoimet riippuvat sen muodosta, joten kertoimet voidaan selvittää vasta kun vieheen lopullinen muoto on tiedossa.

3.2 Lisätoiminnot

Vieheelle tarvittaviksi lisätoiminnoiksi päätettiin syvyyden mittaus ja epäsäännöllinen värinä. Vieheen kaikkia toimintoja ohjaamaan tarvitaan myös mikrokontrolleri.

3.2.1 Syvyyden mittaus

Syvyyden mittaaminen voidaan hoitaa tarkkailemalla etäisyyttä joko veden pinnasta tai pohjasta. Etäisyys pinnasta voidaan laskea paineanturin avulla. Anturi mittaa vieheen uintisyvyydellä vaikuttavan paineen ja mikrokontrolleri laskee tiedon avulla vieheen syvyyden.

Etäisyys pohjasta toteutettaisiin kaikuluotaimen tavoin ultraäänianturilla. Viehe tunnistaisi tällöin muuttuvat pohjanmuodot ja pystyisi välttämään pohjaan tarttumisen. Ultraäänianturin avulla vieheelle voidaan myös ohjelmoida toiminto, joka saa vieheen liikkumaan aina tietyllä etäisyydellä pohjasta, tavoittaen pohjan tuntumassa pysyttelevät kalat. Tällä menetelmällä toteutettava viehe olisi ainutlaatuinen.

Molemmat anturit joutuvat olemaan kosketuksissa veden kanssa, jolloin vedenkestävyys on ehdoton vaatimus.

3.2.2 Epäsäännöllinen värinä

Värinän toteuttamiseen käytetään värinämoottoria. Tärkeimmät vaatimukset moottorille ovat pieni koko ja pieni käyttöjännite. Epäsäännöllisyyden toteuttamiseksi värinämoottorin

ohjaukseen käytetään kytkimenä toimivaa transistoria. Mikrokontrolleri antaa transistorille satunnaisesti virtasignaalin, joka saa sen johtamaan virtaa ja moottorin värisemään. Transistori mitoitetaan kaavoilla (22) ja (23). Kyllästytilan ja mikrokontrollerille turvallisen virran varmistamiseksi värinämoottoria ohjaavan transistorin kanssa kytketään sarjaan vastus, joka mitoitetaan kaavan (24) mukaisesti.

3.2.3 Mikrokontrolleri

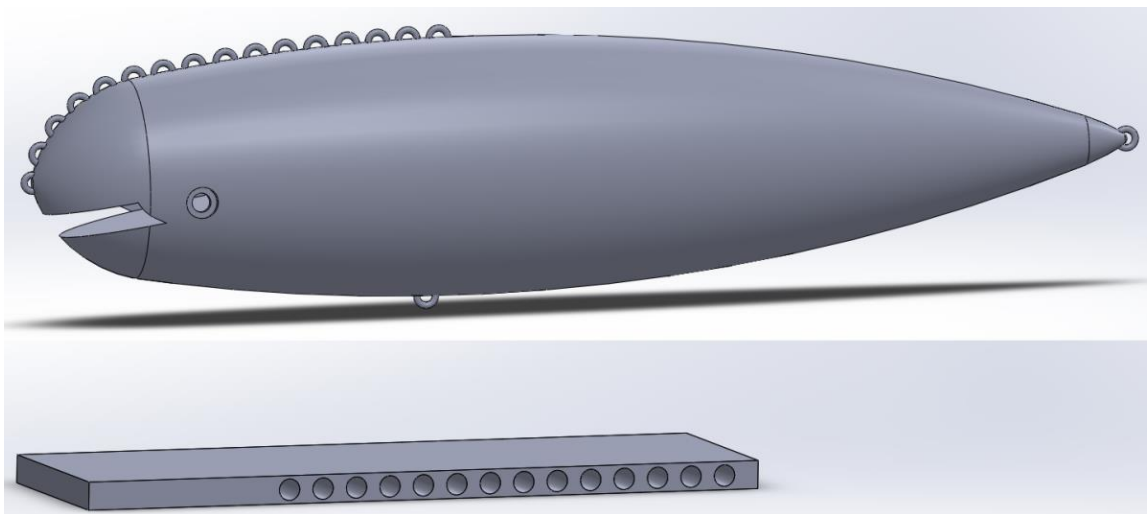
Kaikkien toteutustapojen tärkein yhteinen komponentti on mikrokontrolleri. Vieheen toimintoja ohjaava koodi ladataan mikrokontrollerille. Kuten muissa sähköisesti toimivissa osissa, mikrokontrollerissakin pieni käyttöjännite on tärkeää. Sen on kuitenkin kyettävä antamaan antureille ja muille komponenteille vaadittava jännite ja virta. Mikrokontrollerissa on myös oltava tarpeeksi I/O-pinnejä kaikkien tarvittavien komponenttien käyttämiseen. Helppo uudelleenohjelmointi on suotavaa. Tämä voidaan toteuttaa esimerkiksi mikrokontrolleriin integroidun USB-portin avulla.

4 MANUAALISET TESTIT

Manuaalisiin testeihin valittiin kaksi lupaavimmalta vaikuttavaa säätömekanismia. Kaavojen (1) ja (2) sekä noste- ja vastuskertoimien perusteella nokkalevyn pinta-alan muutoksella saadaan aikaan suurempi noste- ja vastusvoimien muutos kuin kulmaa kääntämällä. Vieheen painon merkitys suhteessa nokkalevyyn vaikuttaviin voimiin arvioitiin pieneksi, joten painon säätelyä ei otettu jatkotarkasteluun. Painopisteen säädön toimivuus ei tuntunut vakuuttavalta, joten sen jatkokehityksestä luovuttiin. Siimapisteen säätäminen sen sijaan vaikutti teoriassa lupaavammalta tavalta kallistaa viehettä ja saada näin säädettyä myös runkoon vaikuttavia voimia. Jatkotarkasteluun valittiin täten nokkalevyn pituuden ja siiman kiinnityspisteen sijainnin säätö. Testejä varten 3D-tulostettiin manuaalinen prototyyppi ja sille nokkalevy. Testit toteutettiin vetämällä prototyyppiä virvelillä järvessä.

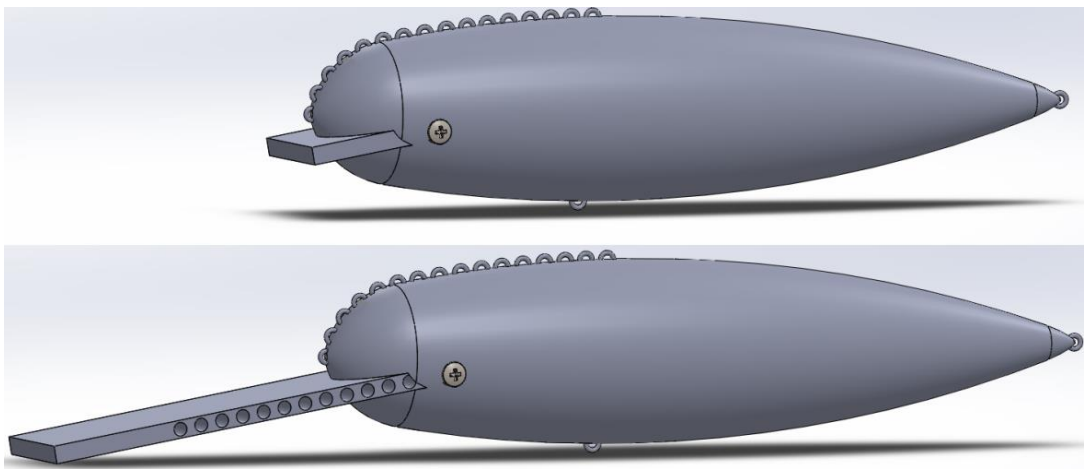
4.1 Manuaalinen prototyyppi

Prototyypin 3D-malli tehtiin SolidWorks-ohjelmistolla. Prototyypin vaatimuksena oli, että sekä siimapisteen sijainnin että nokkalevyn pituuden muutokset voitaisiin testata samalla prototyypillä. Tällöin rungon muoto ja massa pysyvät vakiona, jolloin muuttuvina parametreina ovat vain siimapiste ja nokkalevy. Lopputuloksena valmistuivat kuvan 15 mukaiset runko ja nokkalevy.



Kuva 15. 3D-mallit manuaalisesta prototyypistä ja nokkalevystä.

Siimapisteen vaikutuksen testaamiseksi prototyypissä oli tasaisin välimatkoin kiinteitä siimalenkkejä, joihin siima voitiin kiinnittää. Nokkalevyn pituuden vaikutusten testaamiseksi prototyypissä oli 8 cm:n syvyinen kolo, johon 3D-tulostettu nokkalevy laitettiin. Nokkalevyssä oli reikiä, joiden avulla pituutta säädeltiin. Levyn pituus oli 10 cm. Levy kiinnitettiin runkoon ruuvilla ja mutterilla. Sekä siimalenkkien että levyn reikien välinen etäisyys oli 5 mm.



Kuva 16. Manuaalisen prototyypin nokkalevyn pituuden säätäminen.

4.2 Testien tulokset

Siimapisteen sijainnin merkitystä testattaessa nokkalevy pidettiin lyhyimmässä asennossa. Siiman ollessa kiinnitettynä vieheen kärjessä olevaan ensimmäiseen lenkkiin uintisyvyys oli hitaasti vedettäessä noin 20 cm. Kun siima kiinnitettiin noin 4 cm:n päähän vieheen kärjestä, vieheen uinti muuttui liian epävakaaksi ja se alkoi pyöriä. Välillä 0–4 cm vieheen uintisyvyudessa ei tapahtunut huomattavaa muutosta.

Nokkalevyn pituuden merkitystä testattaessa siima pidettiin kiinni vieheen kärjessä olevassa ensimmäisessä siimalenkissä. Nokkalevyn lyhyimmässä asennossa uintisyvyys oli edellä mainittu noin 20 cm. Pituutta kasvattaessa myös uintisyvyudessa tapahtui huomattavaa kasvua. Nokkalevyn pituuden ollessa 5 cm vieheen syvyys oli noin 0,5 m. Järven sameuden vuoksi vieheen mennessä tätä syvemmälle se hävisi näkyvistä. Jos vetonopeutta kasvatettiin tai nokkalevyn pituus oli lähellä sen ääriasentoa, viehe alkoi pyöriä vedettäessä.

Koska testit suoritettiin sameassa järvessä, mitään tarkkoja syvyystietoja ei ollut mahdollista saada. Testien perusteella kuitenkin nokkalevyn pituuden säätäminen on kahdesta testatusta menetelmästä vieheen uinnin kannalta varmempi ja sillä saadaan huomattavasti kasvatettua uintisyvyyttä. Täten vieheelle valitaan nokkalevyn pituutta säätävä mekanismi.

Molemmilla menetelmillä ongelmaksi muodostui vieheen epävakautuminen ja siitä aiheutuva vieheen pyöriminen. Ongelmat johtuivat todennäköisesti rungon muotoilusta ja painopisteen sijainnista. Kun vieheen annettiin upota vapaasti, se upposi perä edellä. Tämä viittaa siihen, että prototyypin painopiste oli liian takana. Lopullista prototyyppiä suunniteltaessa on siis kiinnitettävä huomiota komponenttien sijoitteluun ja pyrittävä saamaan painopiste mahdollisimman eteen. Mikäli komponentteja ei saada sijoiteltua riittävän hyvin, painopistettä voidaan saada lähemmäksi vieheen kärkeä esimerkiksi lisäpainoilla, jotka laitetaan rungon sisälle. Myös rungon muotoilua on mahdollisesti optimoitava.

5 VALITUT KOMPONENTIT

Manuaalisten testien perusteella vieheen syvyys säädetään nokkalevyn pituutta muuttamalla, joten vaadittavat komponentit etsittiin tälle mekanismille. Askelmoottorilta vaadittiin riittävän väännön lisäksi mahdollisimman pientä kokoa ja käyttöjännitettä. Myös lisätoimintoihin vaadittavien osien etsinnässä pyrittiin löytämään mahdollisimman pieniä ja käyttöjännitteeltään samantasoisia komponentteja.

5.1 Syvyysäätomekanismin komponentit

Ruuvijohteeksi valittiin 100 mm pitkä, ruostumattomasta teräksestä valmistettu M8-ruuvi messinkisellä mutterilla. Ruuvijohteelle etsittiin myös muovinen liukulaakeri kitkan pienentämiseksi.

Moottoriksi valittiin lasketun suurimman aksiaalivoiman perusteella unipolaarinen Nema 11 -runkoinen askelmoottori. Moottorin maksimivääntö on 4,3 Ncm, käyttöjännite 2,7 V ja vaihevirta 0,95 A. (OMC-StepperOnline.)

Moottorin ohjauksen helpottamiseksi valittiin moottoriohjaimeksi Pololun DRV8834-ohjain. Ohjain on tarkoitettu bipolaarisille askelmoottoreille, mutta valmistajan mukaan sillä voi ohjata myös unipolaarisia moottoreita. Valmistaja suosittelee jännitepiikkien tasaamiseksi ohjaimelle kondensaattoria, jonka kapasitanssi on vähintään 47 μ F. (Pololu.)

5.2 Mikrokontrolleri

Mikrokontrolleriksi valittiin Adafruit Pro Trinket 3V. Siinä on 18 GPIO-pinniä, joten ne riittävät kaikkiin tarvittaviin toimintoihin. Mikrokontrollerin GPIO-pinnien lähtöjännite on 3,3 V ja maksimivirta 20 mA, mutta siinä on myös 3V-pinni, jonka maksimivirta on 150 mA. Ulkoisen jännitelähteen on oltava vähintään 3,5 V ja enintään 16 V. Kommunikointiin oheislaitteiden kanssa voidaan käyttää I²C- ja SPI-väylää. Ohjelmien lataamista varten mikrokontrollerissa on Micro-B -tyypin USB-portti. (Adafruit 2018, s. 4—11.)

5.3 Paine- ja ultraäänianturit

Paineanturiksi valittiin MS5803-14BA. Se on veden syvyyden mittaamiseen optimoitu ja pienikokoinen absoluuttinen paineanturi. Anturin kestävä maksimipaine on 14 bar ja sen resoluutio on noin 0,2 mbar. Sen maksimivirta on 1 μ A ja käyttöjännite 1,8–3,6 V. Tiedonsiirtoon mikrokontrollerin kanssa voidaan käyttää joko I²C- tai SPI-väylää. (TE Connectivity 2017, s.1.)

Ultraäänianturiksi valittiin MB7062 XL-MaxSonar-WR1. Anturin keskimääräinen virrankulutus on 3,4 mA ja käyttöjännite 3–5,5 V. Havaintoetäisyys ilmassa on 20–765 cm ja resoluutio 1 cm. Mitattu etäisyys voidaan lähettää mikrokontrollerille analogisena jännitesignaalina, sarjamoitaisena datana tai pulssinleveytenä. Valmistaja suosittelee anturille kohinan vähentämiseksi kondensaattoria, jonka kapasitanssi on 100 μ F. (Maxbotix, s. 2.)

Anturille on valittavissa erilaisia koteloiteja ja tähän projektiin valittiin Compact-mali. Anturia ei ole suunniteltu vedessä käytettäväksi, mutta valmistajan mukaan sitä on käytetty onnistuneesti vedenalaisissa projekteissa (Carlson 2015). Otettaessa äänen noin nelinkertainen nopeus vedessä huomioon havaintoetäisyydeksi saadaan noin 80–3000 cm.

5.4 Värinämoottori, transistori ja vastus

Värinämoottoriksi valittiin VPM2, jonka käyttöjännite on 3 V ja maksimivirta 80 mA (Solarbotics, s. 1). Suuren virtavaatimuksen vuoksi moottori on yhdistettävä mikrokontrollerin 3V-pinniin. Moottoria ohjaavaksi transistoriksi valittiin moottorille mainittujen tietojen perustella NPN-tyyppinen PN2222A, jonka maksimijännite on 40 V ja maksimivirta 1 A (Fairchild 2014, s. 1). Virtavahvistuskertoimeksi ilmoitetaan 300 (Fairchild 2014, s. 3).

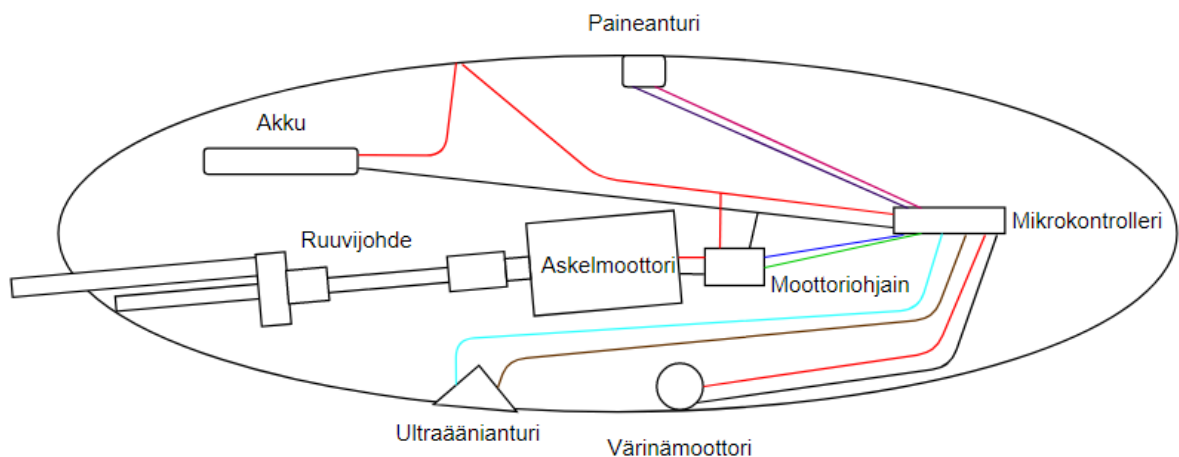
Käyttäen kaavaa (22) kantavirraksi saatiin 0,27 mA. Kylläystilan varmistamiseksi halutuksi kantavirraksi valittiin sen ja mikrokontrollerin GPIO-pinnin maksimivirran väliltä 3–7 mA. $V_{BE(sat)}$ katsotaan kollektorivirran 150 mA mukaan. Maksimiarvoksi on tällöin annettu 1,2 V (Fairchild 2014, s. 3). Sijoittamalla se ja mikrokontrollerin lähtöjännite 3,3 V kaavaan (23) saadaan jännitehäviöksi 2,1 V. Kaavaa (24) käyttämällä vastuksen resistanssiksi saatiin tällöin noin 300–760 Ω .

5.5 Jännitelähde

Anturit kytketään mikrokontrolleriin ja askelmoottori moottoriohjaimeen, jolloin suoraan jännitelähteeseen kytketään vain mikrokontrolleri ja moottoriohjain. Näistä kahdesta suurin jännitevaatimus on mikrokontrollerilla, joten jännitelähde valitaan sen mukaan. Koska mikrokontrollerille tarvitaan vähintään 3,5 V, valittiin jännitelähteeksi suojapiirillinen 18650-tyypin Li-ion -akku, jonka nimellisjännite on 3,7 V, kapasiteetti 2600 mAh ja maksimivirta 5A (Akkula). Akulle etsittiin myös sopiva pidin.

6 PROTOTYYPIN RAKENTAMINEN

Kun komponentit saatiin valittua, aloitettiin prototyypin rakentaminen. Rakennettavassa prototyypissä paine- ja ultraäänianturit antavat mikrokontrollerille tietoja vieheen uintisyvyydestä ja ympäristöstä, joiden perusteella säädetään nokkalevyn pituutta. Virran säästämiseksi vieheen tulisi käynnistyä vasta vedessä. Tämä toteutuu yksinkertaisimmillaan siten, että akusta lähtevä ja mikrokontrolleriin sekä moottoriohjaimen liittyvä johdin sijoitetaan lähellekin rungon ulkopinnalle, jolloin vesi sulkee virtapiirin. Prototyypin konsepti esitellään kuvassa 17.



Kuva 17. Prototyypin konsepti.

6.1 Kytkeä

Suoraan jännitelähteeseen kytkettiin mikrokontrolleri ja moottoriohjain. Jännitelähteen plusnapa kytkettiin mikrokontrollerin BAT+-pinniin ja moottoriohjaimen VMOT-pinniin, miinusnapa mikrokontrollerin G-pinniin ja moottoriohjaimen VMOT-pinnin vieressä olevaan GND-pinniin. Moottoriohjaimen VMOT- ja GND-pinnien välille kytkettiin kondensaattori, jonka kapasitanssi on 47 μF .

Koska väriämoottorin käyttövirta on 80 mA, se kytkettiin mikrokontrollerin 3V-pinniin. Väriämoottoria ohjaava transistori voitiin kytkeä mihin tahansa mikrokontrollerin digitaaliseen pinniin ja pinnin numeroksi valittiin 8. Transistorin kanssa sarjaan kytkettiin vastus, jonka resistanssi on 300 Ω .

Paineanturin tiedonsiirtoprotokollaksi valittiin I²C. Tällöin anturin SDA-pinni kytketään mikrokontrollerin SDA-pinniin, joka on analoginen pinni A4. Vastaavasti anturin SCLK-pinni kytketään mikrokontrollerin SCL-pinniin, joka on analoginen pinni A5. Käytettävä protokolla valitaan anturin PS-pinnillä. I²C-protokolla aktivoituu, kun pinnin tilaksi asetetaan HIGH (TE Connectivity 2017, s. 7). Pinni kytkettiin mikrokontrollerin digitaaliseen pinniin 11. Anturin CSB-pinni on yhdistettävä joko lähdejännitteeseen tai maahan (TE Connectivity 2017, s. 8). Pinni yhdistettiin maahan. Anturin VDD-pinni yhdistettiin mikrokontrollerin 3V-pinniin ja GND-pinni G-pinniin.

Ultraäänianturin datan lähetystavaksi valittiin analoginen signaali, joten anturin AN-pinni on kytkettävä johonkin mikrokontrollerin analogiseen pinniin. Pinniksi valittiin A0. Anturin V+-pinni yhdistettiin mikrokontrollerin 3V-pinniin ja GND-pinni G-pinniin. Kohinan vaimentamiseksi V+- ja GND-pinnien välille kytkettiin kondensaattori, jonka kapasitanssi on 100 µF.

Koska valittu askelmoottori on unipolaarinen ja ohjain on tarkoitettu bipolaarisille moottoreille, moottorin O+ - ja O- -johtimet jätettiin kytkemättä. A+ -johdin kytkettiin ohjaimen A2-pinniin, A- -johdin A1-pinniin, B+ -johdin B2-pinniin ja B- -johdin B1-pinniin.

Moottoriohjaimen M0-pinni kytkettiin mikrokontrollerin pinniin 4. STEP-pinni kytkettiin pinniin 12 ja DIR-pinni kytkettiin pinniin 10. Mikrokontrollerin 3V-pinni kytkettiin ohjaimen SLEEP-pinniin ja G-pinni kytkettiin ohjaimen FAULT-pinnin vieressä olevaan GND-pinniin.

Kytchentäkaavio esitellään liitteessä I.

6.2 Ohjelmointi

Vieheen ohjelmointi toteutettiin Arduino IDE -ohjelmiston versiolla 1.8.5. Paineanturia ei saatu ollenkaan toimimaan kummallakaan protokollalla, joten anturi oli todennäköisesti hajallinen. Koska uutta ei olisi ehtinyt saada ajoissa, vieheelle ohjelmoitiin kaksi syvyysäättötoimintoa, jotka eivät vaadi tietoa uintisyvyydestä.

Ensimmäisellä toiminnolla viehe ui tietyllä etäisyydellä pohjasta. Ohjelmassa etäisyydeksi asetettiin 40 cm. Jos vieheen etäisyys pohjasta on enemmän kuin 40 cm, askelmoottori pidentää nokkalevyä, jolloin vieheen syvyys kasvaa. Jos etäisyys taas on alle 40 cm, askelmoottori lyhentää nokkalevyä, jolloin vieheen syvyys pienenee. Kun etäisyys on 40 cm, moottori ei yritä säätää levyn pituutta.

Toisella toiminnolla viehe vaihtelee uintisyvyyttään satunnaisesti. Moottorin askelten määrä ja suunta sekä saadulla syvyydellä vietetty aika valitaan käyttäen satunnaislukufunktiota `random()`. Askelten määrä valitaan väliltä 600—1600. Suunta valitaan luvuista 1 ja 2, joista luku 1 madaltaa syvyyttä ja luku 2 kasvattaa syvyyttä. Syvyydellä vietetty aika valitaan väliltä 2—5 s. Kuluneen ajan tarkkailuun käytetään `millis()`-funktiota ajastimena. Laskuri käynnistyy, kun moottori on ottanut valitun askelmäärän. Ultraäänianturi tarkkailee vieheen etäisyyttä pohjasta ja estää levyn pidentämisen, jos havaittu etäisyys on ≤ 40 cm.

Koska prototyypissä ei ole rajakytkimiä, ääriasentojen yli liikkuminen estetään askellaskurilla. Mutterin siirtymisen ruuvin päästä päähän todettiin kokeellisesti vaativan 36 kierrosta. Koska moottorin askelkulma on $1,8^\circ$, yhteen kierrokseen vaaditaan 200 askelta ja 36:n kierrokseen 7200 askelta. Ohjelmassa ääriasentojen määrättiin olevan yhden 7000 ja levyn lyhentäminen estetään kun askelmäärä on ≤ 200 . Askelmääräksi ohjelman alussa määrättiin 3600, joka vastaa mutterin sijaintia ruuvin keskellä. Tämä ratkaisu on kuitenkin ongelmallinen, sillä sen toiminta vaatii sitä, että mutteri on ruuvin keskellä aina ohjelman alkaessa. Prototyypin ohjelmaa testatessa mutteri siirrettiin aina kokeilujen välissä manuaalisesti takaisin ruuvin keskiosaan.

Ultraäänianturin mittaama etäisyys luetaan käyttäen funktiota `analogRead()`. Mitatun etäisyyden resoluutio on 1 cm. Pienen resoluution vuoksi etäisyys voi vaihdella jatkuvasti, jolloin askelmoottori pyrkii säätämään syvyyttä pyörimällä edestakaisin. Syvyyden säätäminen senttimetrin tarkkuudella ei ole tarpeellista ja moottorin jatkuva säätäminen kuluttaa virtaa turhaan. Tästä syystä resoluutioksi muutettiin 10 cm. Koska anturin mittaustulos tallennetaan long-tyyppiseen kokonaislukumuuttujaan, resoluutio muutettiin jakamalla mittaustulos kymmenellä, jolloin tulos pyöristyy lähimpään kokonaislukuun. Kertomalla tulos uudelleen kymmenellä muutetaan yksiköt takaisin senttimetreiksi.

Satunnaisvärinätoiminnon toteutukseen käytettiin `random()`-funktiota. Funktio generoi satunnaisen ajan väliltä 0—2 s, jonka jälkeen muutetaan transistoripinnin tilaa. Jos pinnin tila on LOW, muutetaan tilaksi HIGH, jolloin moottori käynnistyy. Vastaavasti pinnin tilan ollessa HIGH muutetaan tilaksi LOW, jolloin moottori sammuu. Funktiota `milis()` käyttäen tarkkaillaan kulunutta aikaa. Kun oikea aika on kulunut, generoidaan jälleen uusi aika ja muutetaan pinnin tilaa.

Askelmoottorille haluttiin täysaskellus, joten ohjelman alussa M0-pinnin tilaksi asetetaan LOW. Kokeellisesti havaittiin askelmoottorin pyörivän myötäpäivään, kun DIR-pinnin tila on LOW. Moottorin ohjaus toteutettiin `micros()`-funktiolla. Funktiota käytetään ajastimena, kuten `millis()`-funktioitakin. Askellus toteutetaan siten, että moottorin suunta valitaan asettamalla DIR-pinnin tilaksi joko HIGH tai LOW, jonka jälkeen STEP-pinnin tilaksi asetetaan HIGH. Ajastimen avulla tila pidetään samana halutun pulssin pituuden verran. Kun aika on kulunut, asetetaan pinnin tilaksi LOW, jolloin moottori on ottanut yhden askeleen. Ajastimen avulla annetaan tilan jälleen pysyä samana yhtä kauan kuin tilassa HIGH, jonka jälkeen voidaan aloittaa uusi askel. Kokeellisesti todettiin moottorin pyörivän sulavimmin, kun ohjauspulssin pituus on 1600 μ s.

Värinälle, etäisyyden mittaamiselle, satunnaiselle syvyydelle ja tietyllä etäisyydellä pysymiselle luotiin omat funktiot. Näitä funktioita kutsutaan `loop()`-funktiossa, joka toistuu niin kauan kuin mikrokontrollerissa on virta. Syvyysäättötoiminto valitaan prototyypin ohjelmassa siten, että se toiminto jota ei haluta toteuttaa joko poistetaan `loop()`-funktioista tai kommentoidaan pois. Ohjelma esitellään liitteessä II.

7 JATKOKEHITYS

Ajanpuutteen vuoksi prototyyppiä ei saatu viimeisteltyä. Koska saatu paineanturi oli viallinen, prototyyppiin ei voitu sisällyttää vieheen syvyyttä tarkkailevaa toimintoa. Jatkossa tulisi kehittää prototyypille kyky tarkkailla syvyyttään ja toimia saadun tiedon perusteella.

Prototyypille ei ehditty suunnitella runkoa. Rungon suunnittelua varten tulisi pohtia komponenttien optimaalisinta sijoittelua rungon sisälle, jolloin painopiste saadaan mahdollisimman eteen. Rungolle pitäisi myös kehitellä sellainen muoto, että vieheen uinti on vakaata eikä se ala pyörimään.

Koska runkoa ei ehditty suunnitella, ei rungon tiivistystäkään saatu ratkaistua. Suurin ongelma tiivistyksen kannalta on nelikulmainen nokkalevy, joka liikkuu edestakaisin. Komponenteille pitäisi myös kehittää suojaus siltä varalta, että rungon sisälle pääsee vettä tiivistyksestä huolimatta.

Vieheeseen olisi hyvä kehitellä jokin keino muuttaa käytettävää syvyysäättötoimintoa. Tällä hetkellä toiminnon vaihtamiseksi on muutettava ohjelmaa ja ladattava se uudestaan mikrokontrollerille.

Ilman runkoa ei myöskään voitu testata syvyysäädön lopullista toimivuutta. Vaikka manuaalisten testien mukaan sen pitäisi olla mahdollista, ei ole varmaa miten vaihteleva nokkalevyn pituus vaikuttaa vieheen uintiin vedon aikana. Vieheen toimivuus kalastuksessa ei ole tiedossa ilman käytännön kokeiluja toimivalla prototyypillä.

LÄHTEET

Adafruit. 2018. Introducing Pro Trinket. [verkkodokumentti]. Päivitetty 11.1.2018. [Viitattu: 29.5.2018]. 38 s. Saatavissa PDF-tiedostona: <https://cdn-learn.adafruit.com/downloads/pdf/introducing-pro-trinket.pdf>

Akkula. 18650 Keppower Samsung 2600 mAh 5A Li-Ion akku 3,7 V – suojapiirillä – Button-top. [verkkodokumentti]. [Viitattu: 29.5.2018]. Saatavissa: https://www.akkula.fi/epages/akkula.sf/fi_FI/?ObjectPath=/Shops/20110228-11092-27846-1/Products/13292

Balista. [verkkodokumentti]. [Viitattu: 29.5.2018]. Saatavissa: <https://www.balista.com.au/>

Bansal, R.K. 2005. A Textbook of Fluid Mechanics. New Delhi: Laxmi Publications. 501 s.

Bhandari, V.B. 2010. Design of Machine Elements, Third Edition. New Delhi: McGraw-Hill Education. 960 s.

Carlson, C. 2015. Notes on Underwater Ranging. [verkkodokumentti]. [Viitattu: 29.5.2018]. Saatavissa: <https://www.maxbotix.com/Tutorials/underwater-ranging.htm>

eMinnow. 2016. [verkkodokumentti]. [Viitattu: 29.5.2018]. Saatavissa: <http://eminnow.com/>

Fairchild. 2014. PN2222A – NPN General-Purpose Amplifier. [verkkodokumentti]. Julkaistu 2004, päivitetty 7.2014. [Viitattu: 29.5.2018]. 8 s. Saatavissa PDF-tiedostona: <https://www.fairchildsemi.com/datasheets/PN/PN2222A.pdf>

Faltinsen, O.M. 2006. Hydrodynamics of High-Speed Marine Vehicles. Cambridge: Cambridge University Press. 454 s.

Giordano, N. 2012. College Physics: Reasoning and Relationships. Second Edition. Boston: Cengage Learning. 1208 s.

Gope, P.C. 2012. Machine Design: Fundamentals and Applications. New Delhi: PHI Learning. 1312 s.

Hyytinen, T. 2014. Suomalaisen vaapun historia. Jyväskylä: Arma Fennica. 144 s.

Hyytinen, H., Hyytinen, T. 2004. Uistinkalastus ja sata ottuistinta. Jyväskylä: Arma Fennica. 223 s.

MaxBotix. XL-MaxSonar-WR/WRC Series. High Performance, IP67 Weather Resistant, Ultrasonic Range Finder. [verkkodokumentti]. [Viitattu: 29.5.2018]. 25 s. Saatavissa PDF-tiedostona: https://www.maxbotix.com/documents/XL-MaxSonar-WR_Datasheet.pdf

Newman, J.N. 1977. Marine Hydrodynamics. Cambridge, MA: The MIT Press. 402 s.

OMC-StepperOnline. Stepper Motor Model 11HS12-0956D. [verkkodokumentti]. [Viitattu: 29.5.2018]. Saatavissa PDF-tiedostona: <https://www.omc-stepperonline.com/download/11HS12-0956D.pdf>

Ortiz, X., Rival, D. & Wood, D. 2015. Forces and Moments on Flat Plates of Small Aspect Ratio with Application to PV Wind Loads and Small Wind Turbine Blades. Energies. Vol 8:4. S. 2438–2453.

Pololu. DRV8834 Low-Voltage Stepper Motor Driver Carrier. [verkkodokumentti]. [Viitattu: 29.5.2018]. Saatavissa: <https://www.pololu.com/product/2134>

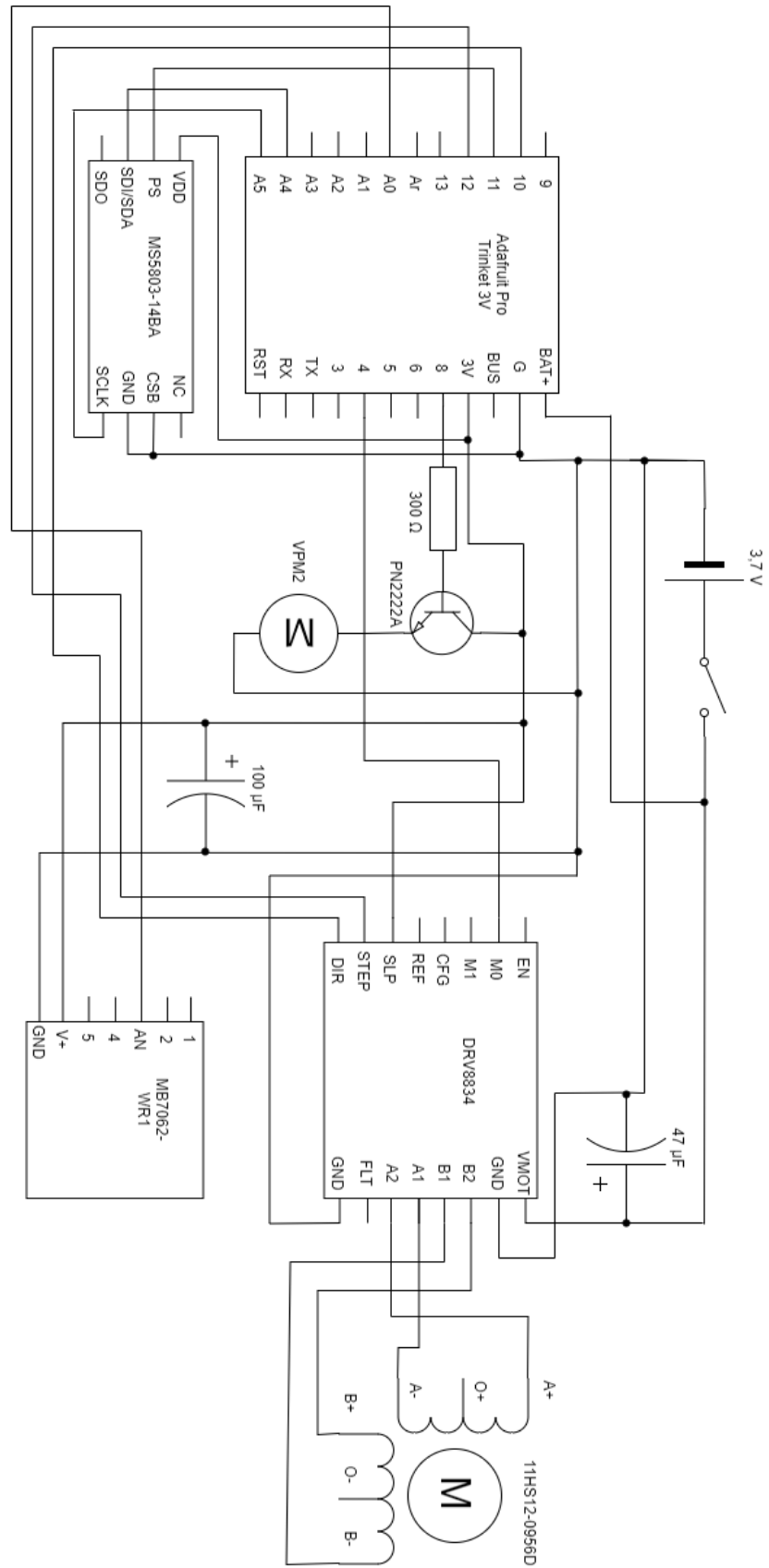
Smartbaits. [verkkodokumentti]. [Viitattu: 29.5.2018]. Saatavissa: <https://www.smartbaits.com/>

Solarbotics. Motor Specification. Flat Type Vibration Motor VPM2. [verkkodokumentti]. [Viitattu: 29.5.2018]. 3 s. Saatavissa PDF-tiedostona: <https://www.robotshop.com/media/files/pdf/datasheet-vpm2.pdf>

TE Connectivity. 2017. MS5803-14BA. Miniature 14 bar Module. [verkkodokumentti].
[Viitattu: 29.5.2018]. 19 s. Saatavissa PDF-tiedostona:
http://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS5803-14BA%7FB3%7Fpdf%7FEnglish%7FENG_DS_MS5803-14BA_B3.pdf%7FCAT-BLPS0013

Twitching Lure. 2018. [verkkodokumentti]. [Viitattu: 29.5.2018]. Saatavissa:
<https://www.twitchinglure.co.uk/>

KYTKENTÄKAAVIO



OHJELMA

```

const int m0 = 4; // Connects pin 4 to driver's M0 pin
const int transistor = 8; // Transistor pin
const int dir = 10; // Connects pin 10 to driver's DIR pin
const int motor_step = 12; // Connects pin 12 to driver's STEP pin
const int an = 14; // Connects A0 pin to ultrasound sensor's AN pin
int vibr_motor = LOW; // Vibration motor's state, LOW=off, HIGH=on
int step_counter = 3600; // Stepper motor steps taken, 3600=middle
                           position
int random_step_counter = 0; // Random steps taken
int random_dir; // Random direction
unsigned long interval = 0; // Time the vibrating motor is on or off
unsigned long vibration_delay; // Vibration motor timer
unsigned long pulse_delay; // Stepper motor control pulse length
unsigned long maintain_depth; // Time spent at current depth when using
                               random_depth() mode
unsigned long depth_interval = 0; // Depth timer
long range; // Range detected by ultrasound sensor
long desired_range = 40; // Insert desired distance from bottom in cm
bool pulse_sent = false; // STEP pin state, true=HIGH, false=LOW
bool forward = false; // Stepper motor direction, true=CCW, false=CW

void setup() {
  pinMode(transistor, OUTPUT);
  pinMode(m0, OUTPUT);
  pinMode(motor_step, OUTPUT);
  pinMode(an, INPUT);
  pinMode(dir, OUTPUT); // LOW=CW, HIGH=CCW
  randomSeed(analogRead (15)); // Generates random seed for random()
                               function from unused analog pin A1
  digitalWrite(m0, LOW); // Setting M0 LOW enables full steps
  vibration_delay = millis(); // Starts vibration motor timer
  pulse_delay = micros(); // Starts stepper motor timer
}

void loop() {
  vibration();
  ultrasound();
  near_bottom(); // Comment out the mode you don't want executed
  //random_depth(); // random_depth() not executed
}

void vibration() {
  if (millis() - vibration_delay >= interval) { // Waits for current
                                                delay for vibration to
                                                pass
    vibration_delay = millis(); // Resets timer
  }
}

```

```

interval=random(0,20)*100; // Random time ranging from 0 to 2000 ms

// Turns vibrating motor on if it was off and vice versa
if (vibr_motor == LOW) {
  vibr_motor = HIGH; // Turns vibrating motor on
}
else {
  vibr_motor = LOW; // Turns vibrating motor off
}
digitalWrite(transistor, vibr_motor);
}
}

void ultrasound(){
  range = analogRead(an); // Outputs range in cm
  range = range/10*10; // Converts range to multiples of 10 cm
}

void near_bottom() {
  if (micros() - pulse_delay >= 1600) { // Control pulse length 1600 us
    pulse_delay = micros(); // Resets timer
    if (pulse_sent == false) { // Executes if STEP pin was set LOW
      if (range < desired_range && step_counter > 200) { // If too close
                                                                    to bottom
                                                                    and 1 rev
                                                                    away from
                                                                    max position

          digitalWrite(dir, LOW); // Turns motor CW
          digitalWrite(motor_step, HIGH);
          pulse_sent = true; // STEP pin was set HIGH
          forward = false; // Lure lip moves backward
        }
      if (range > desired_range && step_counter < 7000){ // If too far
                                                                    from bottom
                                                                    and 1 rev
                                                                    away from
                                                                    min
                                                                    position

          digitalWrite(dir, HIGH); // Turns motor CCW
          digitalWrite(motor_step, HIGH);
          pulse_sent = true; // STEP pin was set HIGH
          forward = true; // Lure lip moves forward
        }
    }
  }
  if (pulse_sent == true) { // Executes if STEP pin was set HIGH
    digitalWrite(motor_step, LOW);
    if (forward == true) {
      step_counter = step_counter + 1;
      pulse_sent = false; // STEP pin was set LOW
    }
  }
}

```



```
        }
    }
    if (pulse_sent == true) { // Executes if STEP pin was set HIGH
        digitalWrite(motor_step, LOW);
        if (forward == true) {
            step_counter = step_counter + 1;
            random_step_counter = random_step_counter - 1;
            pulse_sent = false; // STEP pin was set LOW
        }
        if (forward == false){
            step_counter = step_counter - 1;
            random_step_counter = random_step_counter - 1;
            pulse_sent = false; // STEP pin was set LOW
        }
    }
}
}
```

random_step_ counter if at max or min position