



**UNIVERSITY
OF OULU**

TIETO- JA SÄHKÖTEKNIIKAN TIEDEKUNTA

**Topias Syri
Juho Valkama**

Robottiniskan tekeminen servoilla ja näytöllä

Kandidaatintyö
Tietotekniikan tutkinto-ohjelma

Syri T., Valkama J. (2019) Robottiniskan tekeminen servoilla ja näytöllä. Oulun yliopisto, tietotekniikan tutkinto-ohjelma. Kandidaatintyö. 25 s.

TIIVISTELMÄ

Robotteja käytetään auttamaan ihmisiä erilaisissa tehtävissä, ja robottien yleistyessä ihmiset ja robotit ovat yhä enemmän suorassa vuorovaikutuksessa. Robottien toiminnan voidaan ajatella koostuvan kolmesta osa-alueesta: tekoälystä, sensoreista ja motorikasta. Ihmisen ja robotin välistä vuorovaikutusta kehitettäessä täytyy jokaiseen osa-alueeseen kiinnittää huomiota. Robotin pää on erityisen tärkeässä roolissa ihmis-kone-vuorovaikutuksessa, koska ihminen kiinnittää luonnostaan huomion kasvoihin. Projektissa tehtiin pää osaksi shakkia pelaavaa robottia. Pää koostuu kaksi niveltä sisältävästä niskasta ja siihen kiinnitetystä ohjaustietokoneesta ja näytöstä. Vaatimuksena niskalle oli, että se pystyy katselemaan ympärilleen, eli etsimään mahdollisia vastustajia ja katsomaan shakkilautaa. Runko luotiin 3D-tulostimen ja laserleikkurin avulla. Nivelet olivat Dynamixel XL430-250 -servoja ja kasvoina käytettiin Raspberry Pi:tä ja siihen kytkettyä näyttöä. Nivelten ohjaamista varten toteutettiin ROS-paketti, joka pystyy kommunikoimaan eri moduulien kanssa ja muuttamaan tilaansa kommunikoinnin perusteella. Jatkossa niskan luonnollisuutta voitaisiin tutkia erilaisten runkorakenteiden, yksityiskohtaisemman ja luotettavan servon ohjauksen, ja persoonallisuutta tuovan ilmeikkäiden kasvojen avulla.

Avainsanat: robotti, ihmis-kone-vuorovaikutus, robotin hallinta, sarjaliikenne

Syri T., Valkama J. (2019) Constructing a robot neck with servos and a monitor.
University of Oulu, Degree Programme in Computer Science and Engineering.
Bachelor's Thesis. 25 p.

ABSTRACT

Robots are used to help people in various tasks, and as robotics becomes more common, humans and robots are constantly interacting more. Robotics comprises of three fields: artificial intelligence, sensors and motorics. When developing robotics, it is essential to keep these fields in mind. The head of the robot is especially important, since humans have a natural tendency to focus on faces. In this project we created a head for a chess playing robot. The head comprises of a neck with two joints and microcomputer and a monitor attached to the neck. It was necessary for the neck that it would be able to turn, as if to look around, and spot possible opponents and the chess board. The frame was 3D-modelled and laser-cut. The joints were Dynamixel XL430-250 servos. Raspberry Pi doubled as the face and the microcontroller. The control of the servos was done with Robot Operating System, or ROS, which we created a module for. The module can communicate with other compatible modules and changing its state accordingly. With further development the neck could be improved for example with different structural solutions, more detailed and robust servo control, and a more complex and expressive face.

Key words: robot, human robot interaction, robot control, serial communication

SISÄLLYSLUETTELO

TIIVISTELMÄ
ABSTRACT
SISÄLLYSLUETTELO

1.	JOHDANTO	7
2.	TAUSTAA.....	8
2.1	Historiaa.....	8
2.2	Tekoäly	9
2.3	Sensorit	10
2.4	Robottien rakenne ja sen hallinta.....	11
2.4.1	Robotin jäsenten rakenne.....	11
2.4.2	Robotin jäsenten hallinta	13
2.5	Ihmisen ja robotin vuorovaikutus	14
2.6	Robottiikan etiikka.....	15
3.	TOTEUTUS.....	17
3.1	Runko.....	17
3.2	Servot.....	21
3.3	Ohjaus	23
3.4	Testaus	25
4.	JATKOKEHITYS.....	27
4.1	Runko.....	27
4.2	Servot.....	28
4.3	Ohjaus	28
4.4	Kasvot	28
5.	PROJEKTIN KUVAUS	30
6.	YHTEENVETO.....	31
	LÄHTEET	32

ALKULAUSE

Tämä kandidaatintyö on laadittu Oulun yliopiston tieto- ja sähkötekniikan tiedekunnassa sulautettujen ohjelmistojen projekti -kurssia varten. Kurssilla osallistuttiin yhteiseen projektiin, jonka osa-alueet jaettiin ryhmien kesken.

Oulussa 6.5.2019

Juho Valkama
Topias Syri

LYHENTEIDEN JA MERKKIEN SELITYKSET

V	Voltti
Mm	Millimetri
Baudi	Signaalin muutoksien määrä per sekunti

1. JOHDANTO

Ihmiset ovat pitkään käyttäneet koneita erilaisten tehtävien helpottamiseksi. Autolla pääsee paikasta toiseen nopeasti, kutomakone tehostaa kankaiden tuotantoa merkittävästi ja laskin mahdollistaa monimutkaisten laskutoimitusten suorittamisen vaivattomasti.

Suurin osa koneista on kuitenkin vain työkaluja, jotka vaativat ihmisen valvontaa. Porakone ei poraa itsestään, ja auto vaatii sitä ohjaavan ihmisen. Itsenäisiä, ihmisen korvaavia koneita, robotteja, on vielä suhteellisen vähän. Tehtaiden tuotantolinjat olivat ensimmäisenä ottamassa vastaan tietokoneohjatut laitteet, jotka pystyivät suorittamaan alunperin ihmisille tarkoitettuja tehtäviä tarkasti ja luotettavasti. Nämä robotit kuitenkin soveltuivat lähinnä tarkkaan määritellyn tehtävän toistamiseen.

Tekoälyn, mekaanisten osien ja sensoriteknologian kehityksen myötä robotit soveltuvat myös monimutkaisempiin tehtäviin. Nykyään robotteja löytyy yllättävästikin paikoista. Hotelleissa pienet, renkailla kulkevat robotit voivat toimittaa tavaroita vastaanotolta huoneisiin, ruohoa leikkaavat robotit voivat siivota pihan ilman ihmisen ohjausta ja robotteja on jopa käytetty apuna alueiden vartioinnissa. Robotit voivat toimia yhä itsenäisemmin, ja pystyvät suorittamaan töitä tarkemmin kuin ihminen.

Robotit eivät ole kuitenkaan pelkästään työntekijöitä, vaan niillä voi olla myös viihdearvoa. Esimerkiksi koirarobotit, jotka pyrkivät matkimaan elävän koiran toimintaa, ovat lähinnä leluja ilman mitään välitöntä käytännön hyötyä. Tietokonepeleissä tekoälyä on käytetty korvaamaan ihmisvastustaja, ja myös fyysisen maailman kanssa vuorovaikutuksessa olevia pelirobotteja, kuten pöytätemnistä pelaavia käsivarsia, on kehitetty onnistuneesti. Vaikka näitä robotteja ei ole suunniteltu helpottamaan työntekoa, ovat ne kuitenkin hyödyllisiä tutkimustarkoituksessa.

Robottiikka vaatii osaajia monelta eri alalta. Matematiikan ja tietotekniikan osaamista tarvitaan algoritmien ja tekoälyn toteuttamiseen [1]. Koska robotit ovat pohjimmiltaan ympäristössä toimivia sähkölaitteita, sähkötekniikan osaaminen on ehdoton vaatimus. Mekatronikka on tieteenala, joka yhdistää sähkötekniikkaa ja konetekniikkaa yhdeksi kokonaisuudeksi, jossa tutkitaan erilaisia fyysisen ympäristön kanssa vuorovaikutuksessa olevia sovelluksia. Ympäristön havainnointi vaatii erilaisten sensorien hyväksikäyttöä. Sensoripohjaisten robottien merkittävä haaste onkin systeemin monimutkaisuus, ja vaadittavan osaamisen laajuus. Konenäön ja sensorien tuntemisen lisäksi niitä täytyy pystyä käyttämään hyväksi reaaliajassa. [2]

Tässä projektissa kehitettiin niska shakkia pelaavalle robotille, joka käyttää hyväksi robotiikan eri osa-alueita mahdollisimman laajasti. Yksinkertaisimmillaan shakkia pelaava tietokonevastustaja olisi tietokoneella ajettava shakkisimulaattori, jolle syötetään laudan tila, ja saadaan vastaukseksi vastustajan siirto. Tarkoituksena oli kuitenkin luoda kokonaisuutena laajempi robotti, joka käyttää konenäköä ja mekaanista käsivartta pelatakseen shakkia vastustajansa kanssa ihmismäisesti. Koneen oli tarkoitus pystyä kommunikoidaan ihmisvastustajan kanssa, ja tekemään itsenäisiä päätöksiä. Niskaan, eli robotin kasvoihin, kuuluu näyttö ja kamera.

2. ROBOTIIKKA

Älykkäiden robottien toiminta voidaan jakaa karkeasti kolmeen vaiheeseen: aisti, suunnittele, toimi. Sensorien avulla robotti aistii ympäristöään ja luo kuvan ympäröivästä tilasta [3]. Robotin toimintaa ohjaavan tekoälyn avulla robotti suunnittelee tulevan toimintansa sensoreista saadun palautteen avulla [3]. Kun robotti on tullut johtopäätökseen, se toteuttaa suunnitelmansa erilaisten käsivarsien ja moottoreiden avulla [3]. Toiminnan jälkeen robotti aistii ympäristöään uudelleen, tekee uusia johtopäätöksiä ja edelleen toteuttaa suunnitelmansa. Tämä vaiheiden jatkuva läpikäynti luo robottien perustan.

2.1 Historiaa

Robottiikka on kasvanut 1950-luvulta lähtien voimakkaasti. Pohjaa robotiikan konseptille on luonut 1900-luvun tieteisfiktio ja populaarikulttuuri, joka antoi inspiraation, ja siitä lähtien roboteista on tullut todellisuutta [4,5]. Robotit ovat integraalinen osa nyky-yhteiskuntaa, jossa ne ovat sisäänrakennettu muunmuassa tehtaisiin [5]. Niitä voi käyttää suorittamaan tehtäviä ihmiskehille vaarallisissa ympäristöissä. Ne siis helpottavat jokapäiväistä elämää lähes kaikkialla ympäristössämme jollain tavalla [4]. Robotiikka hyödyntää monia eri tieteenaloja, joista yksi tärkeimmistä on tekoäly [5]. On siis tärkeää tarkastella myös tekoälyn historiaa jotta voidaan ymmärtää robotiikkaa nykytilannetta sekä tulevaisuutta.

1960-luvulla robotiikassa oli eräänlainen jakautuminen kehityssuunnissa. Yksi suuntaus oli taas 1960-luvulla käynnissä olevasta avaruusbuumista, jossa nousi tarve itsenäisesti toimiville roboteille, kuten avaruusmönkijöille. Niiden oli pystyttävä toimimaan itsenäisesti ympäristöissä, joissa yhteyttä ei ollut välttämättä saatavilla. Ympäristössä itsenäisesti toimiminen vaatii robotilta myös tekoälyä, että ympäristön havaitsemista. Robotin täytyy osata aistia ympäristöään jollain tavalla, joten siinä täytyy olla siis sensoreita. [3,6,7]

Koska robottien yksi päätarkoitus on automatisoida eri prosesseja, on tarve tarkalle ja toistettavalle mekaniikalle, ja näin oli esimerkiksi tuotantolinjaroboteilla, jotka olivat toinen suuntaus robotiikalle 1960-luvulla. Niillä oli vaatimus mekaniikalle, mutta niiltä ei kuitenkaan vaadittu minkäänlaista virheenhavaintokykyä massatuotannosta johtuen. Tuotantolinjarobotit olivat käytännössä siis näkökyvyttömiä ja tunnotomia, sillä ne toistivat vaan samaa mekaanista liikettä. [3]

Robottiikan odotukset ja vaatimukset ovat muuttuneet viime vuosina: robotit eivät enää ole vain tehtaissa ihmisten kauko-ohjauksella toimivia koneita [10]. Robotit ovat yhä enemmän suorassa vuorovaikutuksessa ihmisten kanssa [8, 10]. Esimerkiksi sydänkohtauspotilaiden kuntoutushoidossa käytettävät robottikäsivarret, tai vanhusten seuraksi tarkoitettut palvelurobotit [9]. Robotit eivät ole vain työkaluja, vaan saattavat olla jopa ohjata autoa itsenäisesti ihmisen puolesta. Ihmisen ja robotin välinen vuorovaikutus on yhä kasvava tieteenala, jonka tarkoituksena on määritellä säännöt ihmisten ja koneen väliselle yhteistyölle [10].

2.2 Tekoäly

Historiallisesti tekoäly on ollut sen alusta lähtien 80-luvulle saakka ihmisälyn inspiroimaa ja motivoimaa. Toimintojen kuten shakin pelaamisen ja teorioiden todistamisen nähtiin toimivan esimerkkeinä ihmisälystä. Ajateltiin, että tekoäly voisi simuloida ihmisälyä spesifeissä tilanteissa, kuten shakin pelaamisessa, jolloin olisi kyseessä niin sanottu heikko tekoäly. Tai sitten tekoäly voisi tulla itsessään älykkääksi ja kykenisi mukautumaan, jolloin olisi kyseessä vahva tekoäly. Lähestymistapa oli siis yksinomaan ihmisläheinen, muita älykkäitä olentoja ei huomioitu tekoälyn kehityksessä. [11,12]

Tämä lähestymistapa osottautui kuitenkin vajaaksi, sillä vaikka shakkia pelaavat robotit pystyivät päihittämään ihmisen, nämä taidot eivät riittäneet tekemään tekoälystä ihmismäistä. Ongelmaksi osoittautuivat ihmiselle yksinkertaiset asiat, kuten huoneessa käveleminen ilman esineisiin törmäämistä, tai esineiden tunnistaminen. Siirryttiin käytännössä siis ongelmanratkaisupohjaisesta tekoälystä ”mieli kehossa” tyyppiseen tekoälyyn, jossa ympäristössä toimiminen tuli huomion kohteeksi. [11]

90-luvulla mielikuva roboteista siirtyi yksinkertaisesta pyörillä liikkuvista tietokoneista sellaiseen, jossa robotti on ympäristön kanssa vuorovaikutuksessa [11]. Se siis tutkii ja vastaa ympäristön tuomiin haasteisiin. Tällaisen robotin tekoälyn kehityksessä ei keskitytty ihmisiin, vaan inspiraatioita haettiin myös muista älyllisistä olennoista, kuten matelijoista [11]. Näiden muiden eläimien tutkiminen auttoi tutkijoita ratkaisemaan ongelmia liittyen tekoälyn kykyyn toimia ympäristössä. Nykyään robotit toimivat hyvinä testausalustoina tekoälyn sovelluksille [11, 12].

Tekoäly on viime vuosina kehittynyt kovaa vauhtia. Suurimpia tekijöitä ovat laskentatehon jatkuva kasvu ja datan laajempi saatavuus. Suurella datamäärällä ja laskentateholla voidaan luoda ratkaisuja monimutkaisille asioille koneoppimisen avulla. Kun oppivalle ohjelmalle annetaan suuri määrä kuvia ihmisten kasvoista eri tilanteissa, se erottaa niistä piirteet, jotka esiintyvät jokaisessa kuvassa, jossa on kasvot. Kun koneelle annetaan uusi kuva, se vertaa oppimiaan piirteitä annettuun kuvaan. Tätä voidaan käyttää hyväksi robotiikassa muun muassa opettamalla shakkia pelaavan robotin tunnistamaan erilaiset pelinappulat toisistaan. [1]

Koneoppimisella voidaan opettaa robotti mukautumaan vaihteleviin tilanteisiin työtehtävässä, esimerkiksi tunnistamaan shakkinaappulat eri kuvakulmista, tai löytämään tehokkaimman ratkaisun käsillä olevaan ongelmaan. Yleispätevän, ihmisen vertaisen tekoälyn luominen nykyisellä teknologialla on kuitenkin vielä kaukana tulevaisuudessa, joidenkin asiantuntijoiden ennustusten mukaan jopa mahdotonta. Nykyinen tekoäly sopii lähinnä yksittäisten tehtävien suorittamiseen, vaikkakin tehtävät saattavat olla monimutkaisia. Vaikka tekoäly pystyy suorittamaan joitakin tehtäviä paremmin kuin ihminen (kuten voittamaan shakkimestarit shakissa), ihminen on edelleen parempi mukautumaan vaihteleviin olosuhteisiin. [1]

2.3 Sensorit

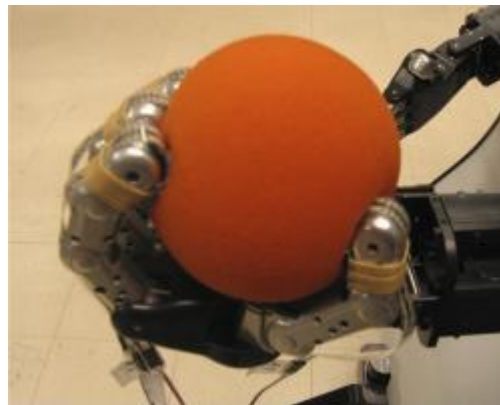
Sensoreiden, tai antureiden, kehitys on monitieteinen prosessi, jossa täytyy hyödyntää muun muassa signaalinkäsittelyä, fysiikkaa ja elektroniikan insinöritieteitä [6]. On myös otettava huomioon robotin käytön konteksti siltä kannalta, kuinka tarkkoja sensoreita siinä on järkevää käyttää [11].

Sensorit luokitellaan yleisesti kahteen ryhmään: kosketuksissa ja ei-kosketuksissa oleviin sensoreihin. Kosketuksissa olevista sensoreista yleisimmät ovat vääntömomenttianturit ja kosketussensorit. Ei-kosketuksissa olevia sensoreita ovat muun muassa kamerat ja ultraäänianturit. [2]

Suurin osa 90-luvulla esiintyneistä robotiikan sovelluksista, kuten kaupallisesti saatavilla olleet Kheperas ja Koalas olivat rajoittuneita sensorimotorisessa kyvyssään pieniin robotteja varten rakennettuihin ympäristöihin ja joskus vastaamaan tiettyihin ympäristön ärsykkeisiin. [11]

Koska näkökyky on monelle robotiikan sovellukselle olennaista, konenäön kehittyminen viimeisen 50 vuoden aikana on avustanut robotiikan kehittymistä huomattavasti [13]. 1960-luvulla konenäössä tutkittiin mahdollisuuksia saada kolmeulotteista dataa kaksiulotteisista kuvista, joissa esiintyi kuutioita [14]. Myöhemmin alettiin käsittelemään kuvia oikeasta maailmasta, jossa pyrittiin tekemään matalan tason havaintoja, kuten reunojen tunnistamista [14]. Koska konenäön sovellukset ovat laajat, sitä hyödynnetään monella eri tieteenalalla [14].

Kuten kaikki muukin teknologia, myös sensorit ovat kehittyneet vuosien aikana. Kamerat ovat entistä tarkempia ja nopeampia, joka helpottaa konenäön avulla tehtyä ympäristön havainnointia. Kosketussensorien merkitys kasvaa, kun ollaan suorassa vuorovaikutuksessa ihmisen kanssa. Ihmisen täytyy pystyä luottamaan, että kone käsittelee häntä turvallisesti, eikä käytä tarpeettoman paljon voimaa. Kuvassa 1 nähdään eräänlainen robottikäsi. [8]



Kuva 1. Robottikäsi

Vielä ei edullinen sensoritekhnologia ole kuitenkaan ihmisen aistien tasolla. Nopeat kamerat ja muunlaiset anturit pystyvät kilpailemaan ihmisten muun muassa näköaistin kanssa, mutta esimerkiksi tuntoaistin tarkka mallintaminen on osoittautunut haastavaksi. Nykyisiin ratkaisuihin kuuluu muun muassa suuri määrä hyvin pieniä paineantureita, jotka yhdessä muodostavat pintaa aistivan tuntoelimen [15]. Toisaalta

sensorien avulla voidaan havainnoida ympäristöä myös tavoilla, joihin ihminen ei kykene, muun muassa lämpökameroilla ja magneettikenttää havainnoivilla antureilla.[16] Nykyiset tarkat ratkaisut ovat kuitenkin erittäin kalliita, joka rajoittaa niiden hyötyä käytännön sovelluksissa.

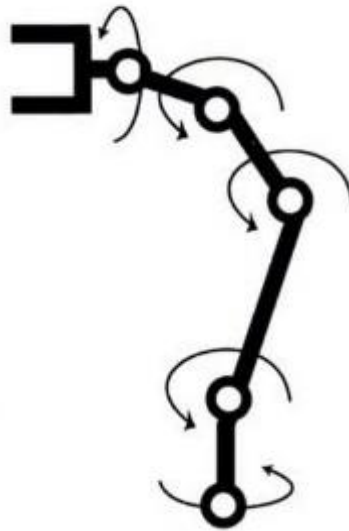
Äänianturit toimivat lähes moitteettomasti hiljaisessa huoneessa, jossa äänilähteitä on vain yksi, mutta niiden kannalta yksi keskeinen ongelma on ollut keskittyä puheeseen melun keskellä [17, 18]. Ihmisiltä ja muilta eläimiltä tällainen keskittyminen onnistuu helposti. Tätä ongelmaa kutsutaan cocktailkutsu-ongelmaksi [17, 18]. Viime vuosina on kuitenkin tähän ongelmaan saatu jonkinlaisia edistyksiä käyttämällä akustisia metamateriaaleja ja puristusantureita. [18] Tällaisista materiaaleista koostuva anturi saavutti 96,67% tarkkuuden äänen tunnistuksessa, kun ympäristössä oli kolme äänilähdettä [18].

2.4 Robotin rakenne ja sen hallinta

Robotin päärungon muodostavia linkkejä, niveliä ja muita struktuurallisia elementtejä kutsutaan manipulaattoreiksi [19]. Robotiikan kehitys on läheisesti yhteydessä teollisen automation kehitykseen. Vuodesta 1956 lähtien robottikäsiä otettiin käyttöön teollisuudessa, joista yhdenlainen tyyppi oli teollinen manipulaattori [3]. Teolliset manipulaattorit ovat ohjelmoitavia ja monikäyttöisiä mekanismeja, joiden tarkoitus on siirtää erilaisia asioita [3]. Ne on suunniteltu siten, että ne voidaan ohjelmoida tekemään erilaisia tehtäviä hyvin tarkasti ja nopeasti [3].

2.4.1 Robotin jäsenten rakenne

Jotta niitä voitaisiin käyttää moniin eri tarkoituksiin, teollisissa manipulaattoreissa täytyy olla monta eri niveltä [3, 20]. Nämä nivelet mahdollistavat liikkumisulottuvuuksia robottikädelle, ja koska tyypillisissä teollisuusroboteissa nivelet määrittelevät yhden paikkamuuttujan, niin liikkuvuusulottuvuuksien määrä on yleensä nivelten määrä [3, 20, 19]. Jokainen nivel tuottaa siis yhden vapausasteen, ja tyypillisesti manipulaattoreissa tulisi olla vähintään kuusi vapausastetta: kolme paikkaan ja kolme orientaatioon. Niveliä voidaan ajatella olevan kahta eri tyyppiä: pyöriviä tai harvemmin käytettyä liukuvia niveliä [19,20]. Kuvassa 2 nähdään MOVEMASTER-robotin nivelten liikkuvuusulottuvuudet [3].



Kuva 2. MOVEMASTER-robotin nivelet.

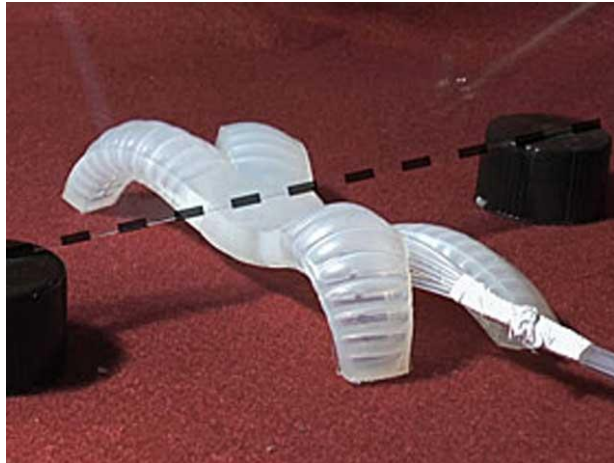
Yksittäisiä kiinteitä runkoja, jotka muodostavat robotin kutsutaan linkeiksi [19]. Manipulaattorin toimintaa voidaan ajatella linkeistä koostuvana ketjuna, jossa nivelet toimivat linkejä yhdistävinä tekijöinä [20]. Liikeketjussa olevista nivelistä käytetään joskus nimitystä ranne [19].

Komponenttien pienentyttyä, voimanlähteenä käytettävien akkujen tehokkuuden kasvaessa ja valmistusmenetelmien parantuessa robottien komponentit on saatu pakattua edellistä pienempään tilaan. Materiaalit ovat yhä kevyempiä ja kestävämpiä, jolloin niiden liikkuttamiseen vaadittavan voima on pienentynyt.

Robottiikan suurimpia fyysisiä haasteita on sen toimiminen monimutkaisissa ympäristöissä. Kehityksestä huolimatta painavan robotin tasapainottaminen, jotta se ei kaatuisi epätasaisella maastolla vaan pystyisi kulkemaan vaivattomasti on ongelma, jota on yritetty ratkaista monella eri tavalla. Sen sijaan, että pyrittäisiin matkimaan ihmisen kaksijalkaista liikkumistapaa, robotit käyttävät usein renkaita tai telaketjuja, joissa paino saadaan jaettua laajemmalle alalle ja liikeradat saadaan pienemmiksi [21]. Viime aikoina on onnistuttu tekemään myös luotettavia kaksijalkaisia robotteja. Esimerkkinä Boston Dynamicsin Atlas-robotti, jota esitellään mm. niiden nettisivulla. Kaksijalkaisuus ei kuitenkaan vielä korvaa renkaita ja telaketjuja halpana ja luotettavana liikkumistapana roboteille.

Perinteisten metalli- ja muoviosien sijaan on kehitetty myös muita malleja robotin rakenteelle [21]. Mallia on otettu muun muassa selkärangattomista eläimistä, kuten mustekaloilta ja madoilta. Joustava ja pehmeä materiaali mahdollistaa monipuolisemmat liikeradat ja lisää bioyhteensopivuutta [22]. Kuvassa 3 on nähtävissä yhdenlainen pehmeä robotti [21].

Pehmeät järjestelmät luovat kuitenkin omia haasteitaan: täysin pehmeät järjestelmät pitää yleensä sitoa ulkoisesti kiinni, jotta pneumaattiset ja elektroniset komponentit toimivat [22]. Vaihtoehtoisesti sitomattomaan pehmeään järjestelmään pitää integroida kömpelöitä osia, kuten pattereita tai moottoreita [22].



Kuva 3. Selkärangattomien mukaan mallinnettu robotti¹

2.4.2 Robotin jäsenten hallinta

Alkukantaiset robotit ohjelmoitiin metodilla, jossa liikutettiin robotti haluttuun kohdesijaintiin ja tallennettiin muistiin toiminnot, jotka se toistaisi. Kuitenkaan tällainen opetustapa ei ole laajemmissa sovelluksissa käytännöllistä erilaisten mahdollisten tilanteiden suuren määrän takia. Tietokoneiden hintojen laskettua robottien hallinta ja ohjelmointi on siirtynyt ohjelmointikielipohjaiseksi. [20]

Jos ihminen on vuorovaikutuksessa manipulaattorin kanssa, esimerkiksi ihmistä avustamassa oleva robotti, täytyy ottaa huomioon manipulaattorin rajoitukset muun muassa voimantuoton kanssa. Rajoituksia täytyy asettaa esimerkiksi siihen, ettei ihminen voi saattaa manipulaattoria tilanteeseen, josta se ei voi palata takaisin. Myös koordinaatiston singulariteetit on huomioitava. [23]

Teollisen manipulaattorin käytössä ollaan tekemisissä objektien kanssa. Karkealla tasolla objekti voidaan pelkistää kahdeksi ominaisuudeksi: sijainniksi ja orientaatioksi. Manipuloinnin kannalta on tärkeää näiden ominaisuuksien käsitteleminen, joten on tarpeellista ottaa käyttöön koordinaattijärjestelmä, jossa objektilla on joku kehys tässä järjestelmässä. Kehyksellä tarkoitetaan neljän vektorin sarjaa, jotka sisältävät informaation paikasta ja orientaatiosta. [20]

Manipulaattoreiden liike voidaan luokitella avoimen ja suljetun silmukan hallintaan [3, 19]. Avoimen silmukan hallinnassa liikkeen suunta ja sen voimakkuus lasketaan vain kerran, jonka jälkeen se suoritetaan, liikkeen aikana ei siis tehdä mahdollisia korjausliikkeitä [3]. Tällaista hallintaa hyödyntäviä robotteja kutsutaan ei-servo-roboteiksi ja niitä käytetään pääasiassa materiaalien siirtämiseen. [19]. Vastaavasti suljetun silmukan hallinnassa mahdollinen este tunnistetaan sensoreilla, jonka jälkeen korjausliikkeen parametrit lasketaan, ja se suoritetaan [3]. Suljetun silmukan hallintaa hyödyntäviä robotteja kutsutaan joskus servo-roboteiksi [19].

¹ Robert F. Shepherd, Filip Ilievski, Wonjae Choi, Stephen A. Morin, Adam A. Stokes, Aaron D. Mazzeo, Xin Chen, Michael Wang, and George M. Whitesides "Multigait soft robot", Proceedings of the National Academy of Sciences Dec 2011. Kuvan käyttöoikeus sallittu PNAS-käyttöehtojen alaisena.

Monijänteisissä systeemeissä jokaiselle nivelelle on laskettava omat arvonsa, jotka siirtyvät järjestelmän seuraavalle nivelelle. Tätä dynaamisuutta voidaan pienentää tekemällä systeemistä jäykkä tietyllä nopeudella. Tämä jäykkyys kuitenkin johtaa suurempaan massaan. [3]

Robotin lihaksina toimivat servot [19, 22]. Käytännössä ne tuottavat manipulaattorille voimaa toimia ulkoisia voimia, kuten paino- sekä liikevoima, vastaan ja siten saavat aikaan liikkeen [19].

Servoja voi olla erilaisia, kuten sähköisiä, hydraulisia tai pneumaattisia, ja niiden kaikkien pitää olla jollain tavalla hallittavissa [19]. Sähköiset servot ovat tarkempia kuin muut, hydrauliset servot saavuttavat suuren nopeuden sekä voima/massa suhteen ja pneumaattiset servot ovat halpoja ja yksinkertaisia, mutta eivät ole yhtä tarkkoja kuin hydrauliset servot [19].

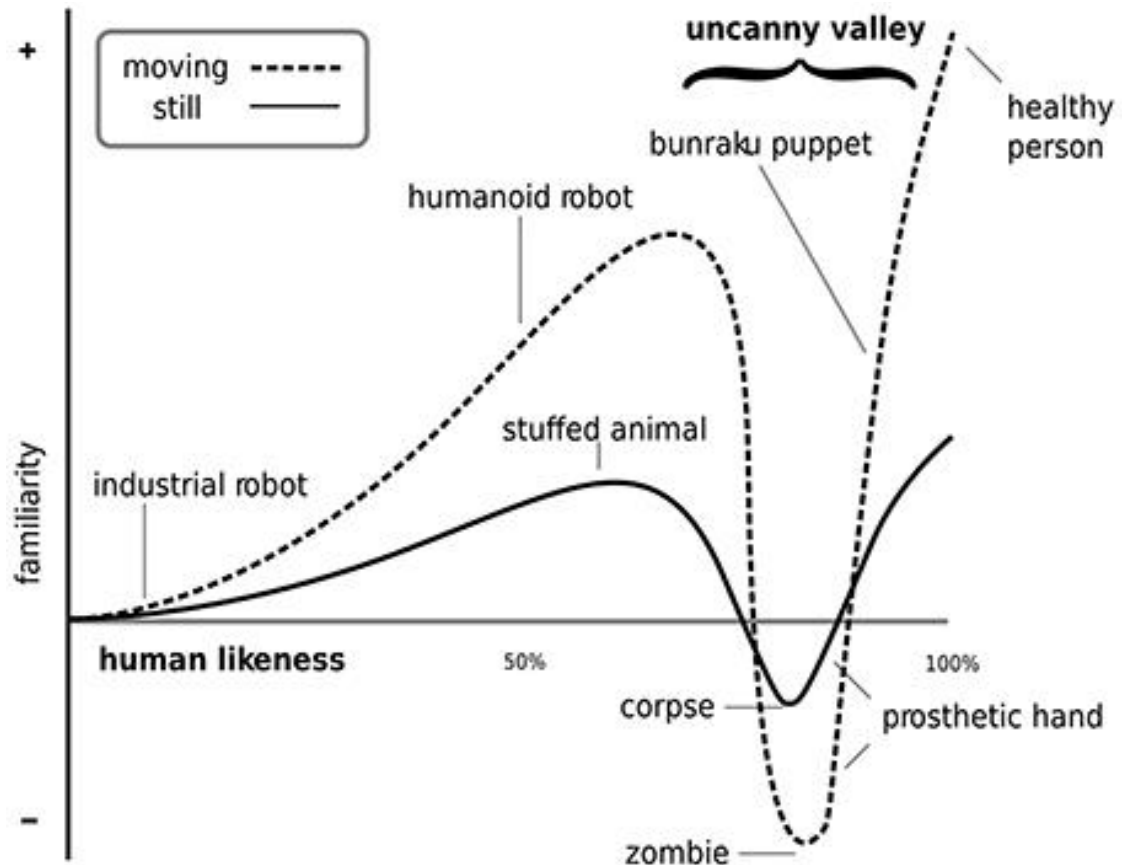
2.5 Ihmisen ja robotin vuorovaikutus

Kun suunnitellaan ihmisen kanssa vuorovaikutuksessa olevaa robottia, täytyy kiinnittää erityistä huomiota käyttötilanteeseen, ja siihen, miten ihminen kokee vuorovaikutuksen. Jos ihminen kokee robotin uhkaavaksi on käyttötilanne epämiellyttävä. Vuorovaikutuksen miellyttävyys riippuu siitä, kuinka helposti käyttäjä hyväksyy robotin, ja pystyy siten luontevasti toimimaan sen kanssa.

Robotin fyysinen olemus, kuten eleet ja rakenne, asettavat käyttäjälle erilaisia oletuksia robotista. Antamalla robotille ihmismäisiä piirteitä, kuten kasvot, on todettu tekevän ihmisen ja robotin välisestä vuorovaikutuksesta luontevampaa. Toisaalta ihmisen liian tarkka kopioiminen saattaa johtaa niin sanottuun "Uncanny Valley", eli Outo Laakso -ongelmaan, jossa robotti on tarpeeksi ihmismäinen muistuttaakseen elävää ihmistä, mutta siltä puuttuu joitain ihmismäisiä piirteitä, jolloin se muistuttaa lähinnä liikkuvaa ruumista. Kun robotin ihmismäisyys saavuttaa Oudon Laakson, käyttäjän mukavuustaso laskee jyrkästi, kuten kuvassa 4 on esitetty. [24]

Outo laakso voidaan välttää käyttämällä sarjakuvamaisia kasvonpiirteitä ja liioiteltuja ilmeitä. Sarjakuvamaisesti liioitellut ilmeet on helppo tulkita, ja niillä voidaan välttää häiritsevän ihmismäinen olemus. Outo Laakso ei kuitenkaan rajoitu vain ilmeisiin, vaan myös liikkeet voivat siirtää kokemuksen epämukavalle alueelle. [24]

Sujuvan vuorovaikutuksen edellytyksenä on käyttäjän ja robotin välinen kommunikointi. Robotit kykenevät tunnistamaan puhetta, ja tuottamaan synteettistä puhetta vastineeksi. Ihmisten välisestä kommunikaatiosta kuitenkin vain murto-osa on suullista. Ihminen kiinnittää erityisesti huomiota katsekontaktiin ja eleisiin [25]. Tämä tuottaa lisähaasteita ihmisen ja robotin väliselle vuorovaikutukselle.



Kuva 4. Ihmisen mukavuustaso kuvattuna ihmisenkaltaisuuden suhteen. “Uncanny Valley” eli Outo laakso näkyy tuttuuden jyrkkänä laskuna¹

2.6 Robotiikan etiikka

Robottien yleistyminen on herättänyt huolta robotiikan eettisyydestä. Työmarkkinoiden näkökulmasta robotit vievät työpaikkoja ammatti-ihmisiltä, joka saattaisi johtaa työttömyyden nousuun. Toisaalta tarkat ja väsymättömät robotit lisäävät tuottavuutta ja tehokkuutta yrityksessä, joka saattaa johtaa yrityksen kasvuun. [26]

Kun on kyse palveluroboteista, nousee esille kysymys yksityisyydestä. Ihmistyöntekijät ymmärtävät yksityisyyden merkityksen, ja ihmisten välinen luottamus syntyy helposti esimerkiksi vanhusten tukipalveluissa. Vanhusten hoidossa vanhuksen terveystiedot ovat välttämättömiä, jotta heidän hyvinvointia voidaan ylläpitää. [27] Robotin tapauksessa robotin suunnittelijan täytyy ymmärtää yksityisyyden merkitys käyttäjäkunnalle ja suunnitella robotin sensoreiden toiminta sekä informaation keräyksen mukaisesti, jotta asiakas pystyy luottamaan siihen, että yksityisyys säilyy. Tutkimuksissa on todettu että lapset kiintyvät liikkuvaan robottiin helpommin kuin pehmonalleen, mutta syvempiä tutkimuksia siitä, miten lapsenvahdin korvaaminen robotilla vaikuttaa lapsen kasvuun. [28]

¹ MacDorman et al., 2005, kuvan käyttöoikeus saatu Creative Commons Attribution 4.0 International –lisenssin alaisena

Moni nykyinen tekoälyn sovellus vaatii valtavat määrät dataa toimiakseen. Dataa kerätään erilaisilta käyttäjiltä, pääosin internetin ja muiden palveluiden käytön perusteella. Suuret teknologiayhtiöt, joilla on suuri käyttäjäkunta ja laaja palveluvalikoima ovat kärkijoukossa tekoälyn jokapäiväiseen käyttöön tuomisessa, koska heille on kertynyt suuri määrä dataa vuosien varrella. Viime aikoina henkilökohtaisen käyttäjätiedon kerääminen suurien ohjelmistoyritysten toimesta on ollut yleinen puheenaihe ja se on johtanut muun muassa EU:n luomiin uusiin tietosuojasääntöihin, joita jokaisen EU maissa toimivan yrityksen täytyy noudattaa.

Robottiikan kehittyessä sovelluksia käytetään yhä enemmän sodankäynnissä. Väsymättömät ja tarkat koneet ovat täydellisiä sodankäynnin välineitä. Ihmishenkien tekoälyn varaan laittamisen mahdollisuus on eettinen ongelma joka nousee esille aika-ajoin. Kun kone tekee päätöksen, kuka on vastuussa lopputuloksesta? [29] Yhdysvalloissa tutkitaan itsenäisesti toimivia robotteja, jotka pystyvät tunnistamaan ja tuhoamaan vihollisen ilman ihmisen tekemää päätöstä. Ongelmana kuitenkin on se, että robotin on vaikeaa erottaa vihollista viattomasta sivullisesta. Toisaalta se johtaisi ihmishenkien laskuun, olettaen, että molemmat osapuolet käyttävät robotteja [28, 29]. Lisäksi robotteja käytetään muun muassa pommien purkuun, jolloin ihmishenkien riski pienenee.

3. TOTEUTUS

Projektissa rakennettiin robotille niska ja kasvot, jotka olisi helppo yhdistää osaksi shakkia pelaavaa robottia. Tarvittiin siis kasvot, jota pystyy käyttämään erilaisen informaation välittämiseen. Lisäksi kasvojen täytyi pystyä liikkumaan, jotta käyttötilanne olisi miellyttävämpi.

Kasvoina päätettiin käyttää Raspberry Pi 3:een kiinnitettävää kosketusnäyttöä. Näyttö oli sopivan kokoinen tarkoitusta varten, ja taustalla oleva Raspberry Pi-minitietokone tarjosi laajan valikoiman erilaisia lähestymistapoja, verrattuna esimerkiksi Arduino UNO:n rajattuun muistiin ja ohjelmistoon. Lisäksi näyttö tarjosi tukevat kiinnityspisteet, joihin oli helppo tehdä vaatimukset täyttävä kannatin.

Liikkeitä varten valittiin Dynamixel XL430-W250-T -servot. Raspberry-näyttö yhdistelmän liikuttamiseen vaadittiin tehokkaita servoja, ja XL430 täytti vaatimuksen. Lisäksi servot tarjoavat hyvät työkalut niiden tarkkaan ohjaamiseen. Servojen asennon pystyy määrittämään kirjoittamalla halutun kulman rekisteriin, ja servon senhetkisen kulman pystyy lukemaan suoraan sen muistista. Lisäksi servon kääntymisnopeutta pystyy muuttamaan suoraan muistiin kirjoittamalla, joka helpotti liikkeen luonnollisuuden suunnittelussa. Muistiin kirjoittaminen tapahtuu lähettämällä servojen protokollaa noudattava viesti sarjaporttimuodossa tietokoneelta servon dataväylään.

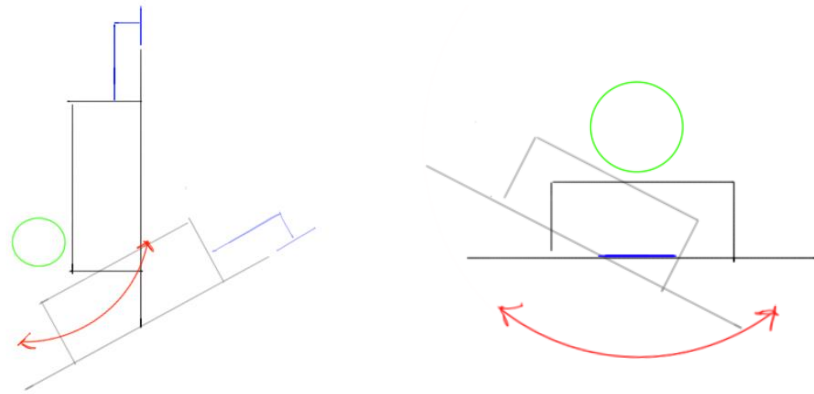
Jotta servon ja Raspberryn välinen kommunikointi toimisi, tarvittiin välikappaleeksi piiri, joka ohjaa kommunikointia servon vuoroasuuntaisen dataväylän ja Raspberryn kaksisuuntaisen sarjaportin välillä. Tässä toteutuksessa kommunikoinnin ohjaukseen käytettiin Arduino UNO:a sen helpon saatavuuden ja käytettävyyden vuoksi. Arduino toimii datan välittäjänä Raspberryn ja servojen välillä Arduinon sarjaporttien kautta.

3.1 Runko

Runkoa suunnitellessa täytyi ottaa huomioon useampia vaatimuksia ja rajoituksia. Robotin tarkoituksena oli toimia ihmisen ja koneen välisenä vuorovaikutuspisteenä shakkia pelaavalle robotille. Nivelten liikkeet ja rakenne täytyi suunnitella siten, että käyttäjän kokemus olisi mahdollisimman miellyttävä ja luonnollinen. Toisaalta täytyi myös ottaa huomioon projektin muut osat ja mahdolliset laajennukset. Koska päätettiin käyttää Raspberry Pin näyttöä robotin kasvoina, liikkeet täytyi suunnitella se huomioiden.

Kasvojen täytyy pystyä kohdistumaan sekä pelaajaa että lautaa kohti. Kamera, jota voidaan käyttää muun muassa laudan tilan havainnointiin ja ympäristön seuraamiseen ja vastustajien tunnistamiseen, sijoittuu kasvojen yläpuolelle. Näin robotin kasvot luonnollisesti suuntaavat senhetkiseen havainnointisuuntaan. Kun robotti lukee laudan tilaa, se on kumartunut laudan ylle kasvot lautaa kohti. Kun robotti kommunikoi pelaajan kanssa, on kasvot kohdistettuna pelaajaa kohti. Robotin aikeet välittyvät käyttäjälle luonnollisella tavalla, joka tekee kokemuksesta miellyttävämmän.

Jotta käyttäjän ei tarvitse istua kohtisuoraan kasvojen edessä, olisi hyvä että kasvot pystyisivät kääntymään myös sivusuunnassa. Näin kasvot pystyvät seuraamaan käyttäjän sijaintia vaikka tämä vaihtaa asentoa. Jos kasvot eivät pysty seuraamaan käyttäjää vaan jää tuijottamaan eteenpäin, robotti jää liian konemaiseksi. Näillä vaatimuksilla päädyttiin kuvan 5 mukaiseen ratkaisuun. Kahdella nivelellä voidaan saavuttaa vaatimuksen mukaiset liikeradat.



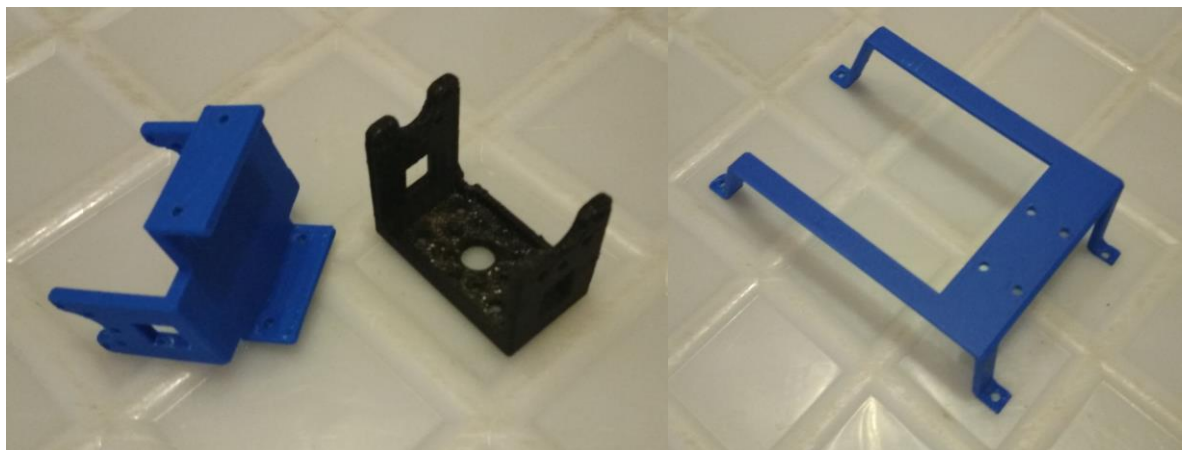
(a) Näyttö kuvattuna sivulta

(b) Näyttö kuvattuna ylhäältä

Kuva 5. Luonnos näytön liikeradoista. Vihreällä esitetty nivelen sijainti, punainen nuoli kuvaa liikkeiden suuntaa ja sinisellä esitetty kameran sijainti.

Rungon osat tehtiin tehdä muovista 3D-tulostimen ja laserleikkurin avulla. Muoviset osat ovat halpoja mutta kestäviä, ja 3D-tulostuksella ja laserleikkurilla voi muovista valmistaa osia tarkoilla mitoilla. Vaadittavat osat olivat kannatin kasvoille, pidikkeet nivelten servoille sekä kameralle ja kaulan tukivarsi.

Kuvassa 6 näkyy servon ja näytön pidikkeet. Koska osat vaativat tarkkoja mittoja, mm. ruuvien sijaintien ja koon takia, ja monimutkaisempia muotoja 3D-tulostus soveltui hyvin osien valmistamiseen. Kasvojen pidike ja kameran kannattimen mallit tehtiin AutoCAD-ohjelmiston avulla. Servon pidikkeen malli on saatu shakkirobotiprojektin käyttöä varten tehdystä mallista, jota muokattiin vastaamaan kaulan vaatimuksia. Jotta osat olisivat tarpeeksi kestäviä, minimipaksuudeksi määriteltiin 3 mm.

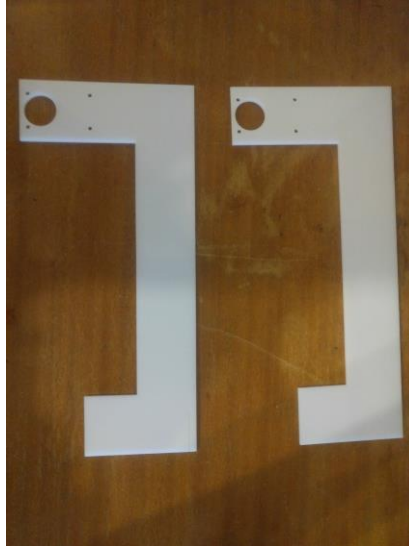


(a) Servojen pidikkeet

(b) Näytön pidike

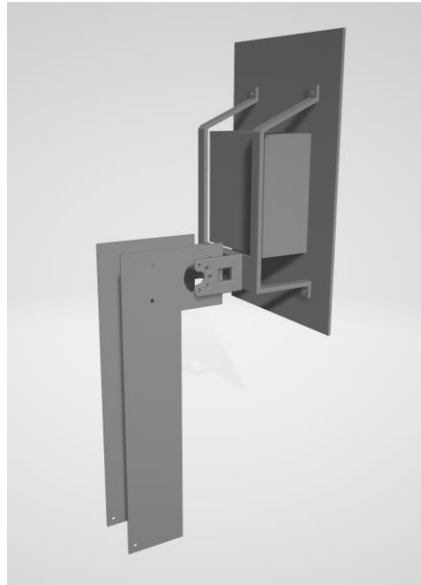
Kuva 6. 3D-tulostetut pidikkeet

Kuvassa 7 esitetyn kaulan varsi oli mahdollista toteuttaa kahdella laserleikatulla paneelilla. 3D-tulostaminen on monen tunnin prosessi, kun taas laserleikkurilla leikkaamiseen menee vain muutama minuutti. Käytettävissä olevien materiaalien vuoksi ainoa vaihtoehto paneelien paksuudelle oli 3 mm.



Kuva 7. Kaulan varren paneelit

Lopullista lopputulosta oli mahdollista simuloida yhdistetyllä 3D-mallilla, esitettynä kuvassa 8. Mallin eri osia pystyi liikkuttamaan itsenäisesti. Näin pystyttiin varmistamaan se, että mikään osa ei tule liikkumisen tielle, ja että mitat toimivat suhteessa toisiinsa. Varren lopullinen pituus oli noin 250 mm, jolloin pöydällä seistessä kasvot jäivät hieman käyttäjän kasvojen alapuolelle. Näin robotti näyttää vähemmän uhkaavalta. Korkeutta on myös mahdollista lisätä jos kasvot jäivät liian matalalle.



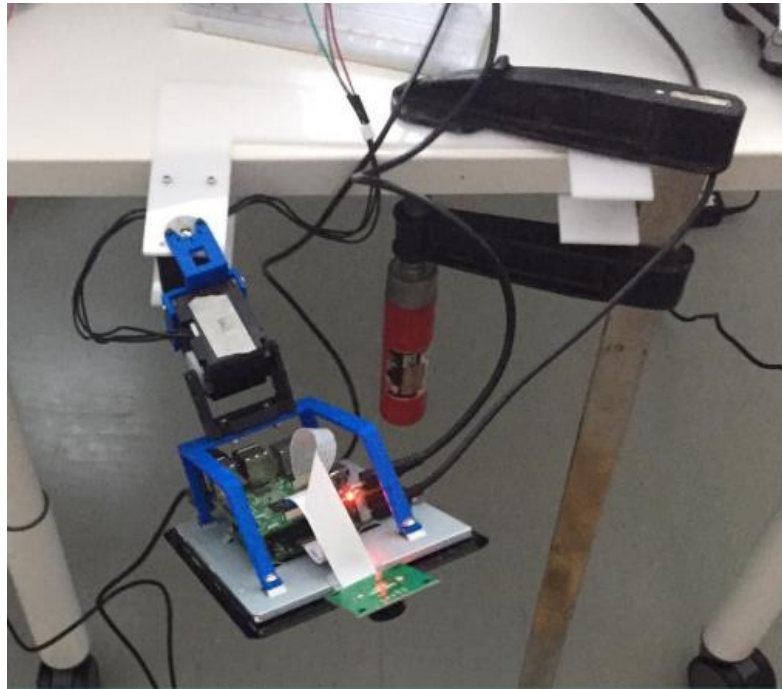
Kuva 8. Yhdistetty 3D-malli

Osien kiinnittäminen tapahtui ruuveilla. Servojen akselien kiinnittämiseen tarvittiin M2-mallin ruuveja, ja servot kiitettiin runkoon M2,5-ruuveilla, jotka molemmat toimitettiin servojen mukana. Näytön kiinnitys sekä tulostettujen osien väliset liitokset tehtiin M3-ruuveilla.

Alkuperäisenä tarkoituksena oli asettaa sivuttain liikkuva servo varren juureen, jolla voitaisiin vähentää itse pään painoa. Kun ensimmäiset osat olivat valmiita ja yhden servon sisältävä prototyyppi kasattu, todettiin että varren juuressa oleva nivel saisi kasvot näyttämään liian konemaiselta. Juuressa oleva nivel kääntää koko vartta, joka on ristiriidassa käyttäjän oletusten kanssa. Kun ihminen katselee ympärilleen, koko ruumis ei liiku mukana.

Suunnitelmat siis muuttuvat, ja sivuttaisnivel siirrettiin varren yläosaan, toisen servon viereen. Ongelmaksi kuitenkin tuli näytön asento, joka ylimääräisen nivelen vuoksi kääntyi sivuttain. Lisäksi uuden servon tuoman painon ja lisävarren takia servon vääntövoima ei riittänyt nostamaan kasvoja silloin, kun kaikki komponentit olivat kiinnitettyinä.

Molemmat ongelmat ratkesivat kääntämällä vartta 90° ja kiinnittämällä se esimerkiksi pöytään kiinni, kuvassa 9 esitetyllä tavalla. Näin servojen liikeradat saatiin vaihdettua keskenään, joka vähensi ylimääräistä painoa ylöspäin suuntautuneelle liikkeelle. Kyseinen kiinnitystapa ei ole ideaali, sillä se rajoittaa kasvojen sijaintia merkittävästi. Se kuitenkin mahdollisti nivelten liikkeiden testaamisen kokonaisuuden ollessa koottuna.



Kuva 9. Niska kiinnitettynä pöydänreunaan.

3.2 Servot

Liikkeen aikaansaamiseksi käytettiin Dynamixel XL430-W250-T -servoja, jollainen näkyy kuvassa 10. Servoihin tutustuminen aloitettiin sillä, että pyrittiin saamaan niihin jonkinlaista liikettä.

XL430-servoa ohjataan muuttamalla sen rekisteriarvoja. Ohjaimelta lähetetään kahdeksanbittinen paketti, joka sisältää muunmuassa servon tunnisteen, rekisteriosoitteen ja -arvon. Linjan nopeutta servoissa voidaan muuttaa, mutta vakioarvo sille on 57600 baudia. Kommunikaatio tapahtuu UART:n avulla, ja viesti noudattaa Dynamixelin 2.0 -protokollaa.



Kuva 10. Dynamixel XL430-W250-T -servot

Koska servon lähdössä/tulossa on vain yksi datalinja, ohjeet servolle annetaan samaa kaapelia pitkin kuin sieltä saatava informaatio. Servon datalinja on siis epäsynkroninen vuoroasuuntainen linja. Koska halusimme myös vastaanottaa tietoa servolta, täytyi Arduinon liikenne muuntaa yksilinjaisesta vuoroasuuntaisesta kaksisuuntaiseksi käyttämällä logiikkapiiriä SN74LS241.

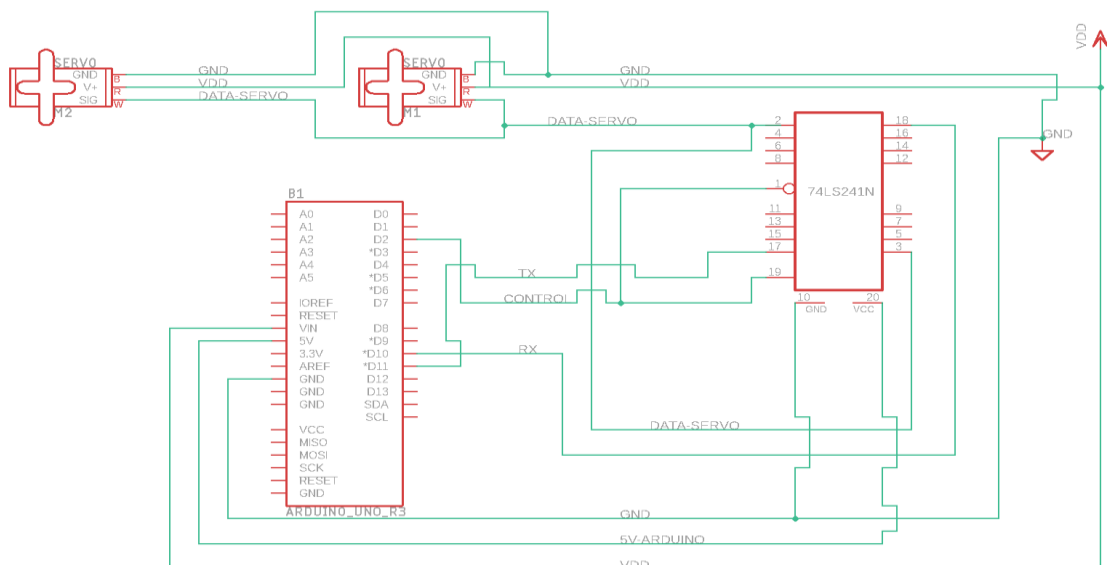
Koska ratkaisumme sisältää kaksi niveltä, eli kaksi servoa, kumpaakin servoa oli pystyttävä ohjaamaan erikseen. XL430-servot on mahdollista kytkeä sarjaan, joka mahdollistaa itsenäisen ohjauksen. Informaatio kulkee servojen läpi, jolloin se ottaa huomioon vain komennot joissa on oikea tunniste kyseiseen servoon.

Servon lähettämä data käsittää muun muassa kommunikaation lopputuloksen (onnistuiko servo tulkitsemaan vastaanotetun paketin) sekä kysytyjen rekisterien arvot. Servon palauttamien statuspakettien ja erilaisten rekisterien arvoilla on mahdollista seurata servon senhetkistä tilaa, ja tunnistaa mahdolliset virheet.

Servon muisti koostuu EEPROM-osiosta ja RAM-osiosta. EEPROM-muisti määrittelee servon ominaisuudet, kuten tunniste (ID), käyttömoodi (Operating Mode) ja kommunikointiin käytetty baudinopeus ja protokolla. RAM-osiosta löytyy ajon aikana muuttuvat arvot, kuten servon senhetkinen asento, seuraava kohdekulma, sekä erilaisia kääntymisnopeuteen ja kiihtyvyyteen liittyviä rekistereitä.

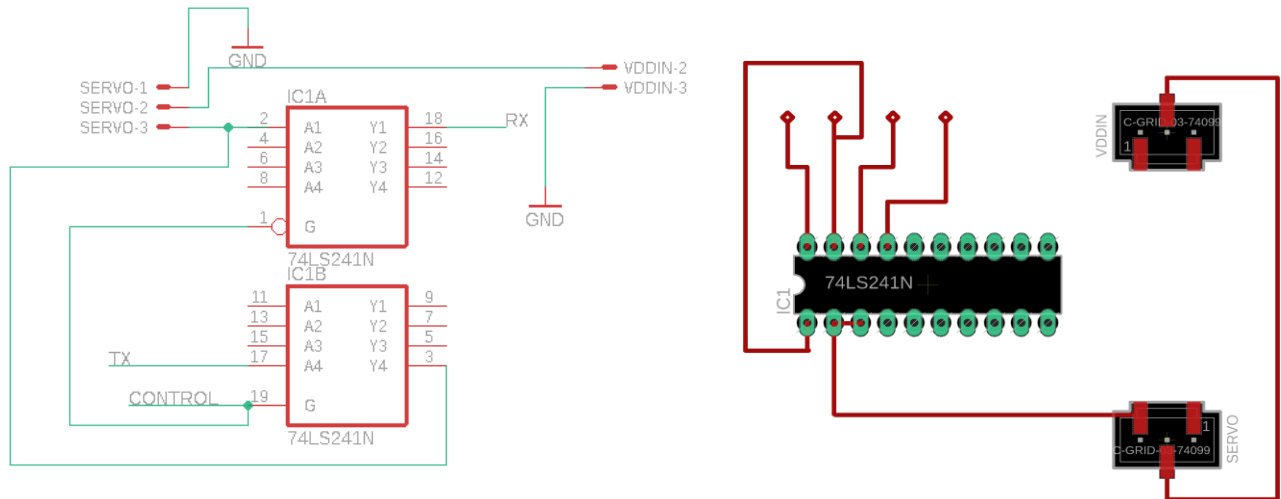
Servo toimii 6V-12V jännitteellä, joten Arduinon tarjoama 5V ei riittänyt servon pyörittämiseen. Servo siis vaatii toimiakseen ulkoisen virtalähteen. Testauksen aikana virta saatiin verkkovirtaan kytketystä muuntajasta.

Arduino toimii välikappaleena kytkennässä, jolloin siihen ladattu koodi välitti USB:stä saamansa informaation sarjaporttikytkentään. Arduinosta lähti kolme piuhaa aiemmin mainitulle logiikkapiirille, RX, TX ja ohjaus. RX siis vastaanottaa informaation servolta, TX lähettää ohjauskomennot servolle ja ohjauslinja vastaa logiikkapiirin asennosta, eli ohjaako se servon datalinjan RX- vai TX -piuhaan. Kommunikoinnissa käytettiin Arduinon ohjelmistotason sarjaporttia, jonka suurin mahdollinen nopeus on 115200 baudia. Jännite Arduinolle saatiin ulkoisesta virtalähteestä. Koko kytkennän suunnitelma näkyy kuvassa 11.



Kuva 11. Koko kytkennän suunnitelma

Testausvaiheessa kytkennät oli tehty koekytkentälevylle. Lopullista kytkentää varten suunniteltiin piirilevy, joka näkyy kuvassa 12, mutta käytettävissä oleva laitteisto oli liian epätarkkaa; käytössä olevan jyrsimen puinen alusta oli epätasainen, ja korjausyrityksistä huolimatta tarkkuutta ei saatu parannettua. Piirilevyn suunnitelma on kuitenkin valmis ja se voitaisiin tarvittaessa toteuttaa suhteellisen helposti



(a) Piirilevyn suunnitelma

(b) Lopullinen piirilevy

Kuva 12. Eagle-ohjelmalla tehdyt suunnitelmat

3.3 Ohjaus

Rungon ohjaukseen käytettiin Raspberry Pi 3 Model B:tä. Servon ohjaus tapahtui Raspberrissä toteutetun koodin avulla, joka lähetettiin Arduino UNO:n kautta servoihin. Arduinon ja Raspberrin välinen yhteys toimi USB:llä.

Ohjauksessa käytettävä kirjasto, Dynamixel SDK, tarjosi komennot servon rekisteriarvojen muuttamiseen ja lukemiseen. Kirjasto loi viestin valmiiksi Dynamixel Protocol 2.0:n mukaisesti, jolloin pystyttiin keskittymään itse liikkeiden testaamiseen. Suurin osa ohjelmiston testauksesta tapahtui ilman Raspberrystä. Arduino oli kytketty tietokoneeseen, jossa oli Linux Mint 19.1- käyttöjärjestelmä. Testikoodi kirjoitettiin Pythonilla.

Kun Dynamixel SDK -kirjastoa käyttää Python-kielessä, täytyy sarjaportin aukaisuun ja ensimmäisen viestin lähetyksen välillä olla noin 1 sekunnin viive. Tämä johtuu mahdollisesti siitä, että Dynamixel SDK on suunniteltu käytettäväksi erillisen sarjaporttimuuntimen kanssa, kun taas Arduino lisää hieman suoritusviivettä portin aukaisun ja datan välittämisen välille. Kun sama ohjelma ajetaan C-kielessä ilman erillistä viivettä portin aukaisemisen jälkeen, ei vastaavaa ongelmaa esiinny.

Ensimmäisenä servojen liikkeitä testattiin irtonaisena niin, että niitä ei oltu kiinnitetty kaulan osiin. Tavoite oli varmistaa, että molempia servoja pystyttiin liikuttamaan itsenäisesti ja luotettavasti halutulla tavalla. Jos servo yrittäisi liikkua pidemmälle kuin rungon osat sallivat servojen ollessa kiinnitettynä, se saattaisi rikkoa joitain osia. Python-koodilla, joka otti parametreikseen halutun servon kulman ja

ohjattavan servon tunnisteeseen, pystyttiin ajamaan molemmat servot kulmasta toiseen ja seuraamaan niiden käyttäytymistä erilaisissa tilanteissa.

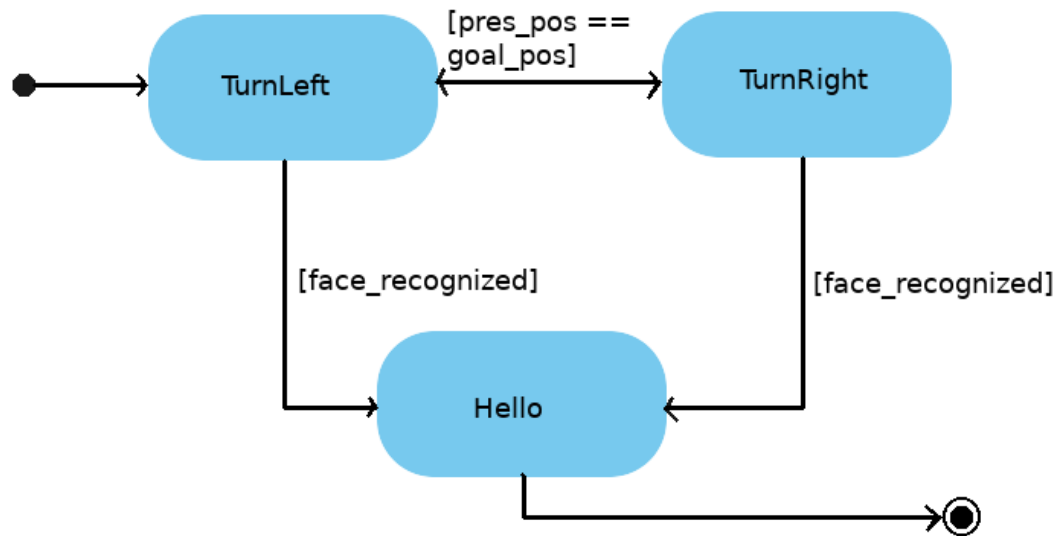
Servojen liikuttamiseen käytetyt kulmat noudattavat servojen sisäistä logiikkaa, jossa täysi kierros akselin ympäri on 4095 yksikköä. Yksi aste on siis 11,375 yksikköä. Toteutuksessa päätettiin käyttää servojen omia yksiköitä, sillä ne olivat numeerisesti riittävän helppo ymmärtää, ja ylimääräisten yksikkömuunnosten tekeminen monimutkaistaisi ohjelmakoodia tarpeettomasti.

Ajon alussa ohjelma lukee servojen senhetkisen kulman niiden rekistereistä. Kaikki liikkeet tehdään luettujen arvojen suhteen, sen sijaan että käytettäisiin absoluuttisia kulmia servon nollakulman suhteen liikkeiden toteuttamiseen. Pääsyyinä tähän on servojen "Torque Enable" ominaisuus. Kun vääntömomenttia ei ole kytkettyä päälle, servojen akselit liikkuvat vapaasti. Vääntömomentin ollessa päällä, servo pyrkii pitämään senhetkisen asennon. Servon EEPROM-muistin arvoja ei pysty muuttamaan jos vääntömomentti on päällä, ja aina kun servoon kytketään virta, vääntömomentti on automaattisesti pois päältä.

Tästä johtuen käynnistyksen yhteydessä olevaa servon asentoa on vaikea ennakoita. Servon akseli on saattanut kääntyä esimerkiksi osien kiinnityksen yhteydessä, joka muuttaa sen kulmaa nollapistettä varten. Riskinä on osien kiinnitys väärässä kulmassa, jolloin liikkeet eivät vastaa odotuksia. Vaihtoehtoinen tapa korjata kyseinen ongelma on ajaa servot niiden sisäiseen nollapisteeseen ajon alussa, mutta riippuen kulmasta, se saattaa joutua kiertämään akselin sellaista reittiä, jossa rakenne ottaa vastaan.

Kun liikkeet tehdään senhetkisen asennon suhteen, liikkeet on mahdollista ennakoita riippumatta sisäisistä arvoista. Ohjelma olettaa aloitusasennon olevan se, johon painovoima luonnollisesti asettaa nivelet vääntömomentin ollessa poissa päältä.

Hyödyllinen funktio lopputuotteen toiminnallisuutta ja muita robotin parissa työskenteleviä ryhmiä ajatellen on skannausfunktio, jossa kamera liikkuu sivulta sivulle. Esimerkiksi kasvojentunnistuksen tai laudantunnistuksen parissa työskenteleville tämä mahdollistaa kohteen paikantamisen suuremmalta alueelta, kuin että kamera osoittaisi vain yhteen suuntaan. Funktiolle annetaan haluttu kääntymiskulma ja servon tunniste, jolloin kasvot kääntyvät puolelta toiselle. Kaula lopettaa skannauksen kun se tunnistaa kasvot tai laudan näkökentässä. Ohjelman toimintaa kasvojentunnistuksessa kuvaava tilakaavio on esitetty kuvassa 13. Kun konenäkö tunnistaa kasvot, robotti lopettaa skannaamisen ja nyökkää kohteelle tervehtiäkseen.



Kuva 13. Kasvojenskannauksen tilakaavio

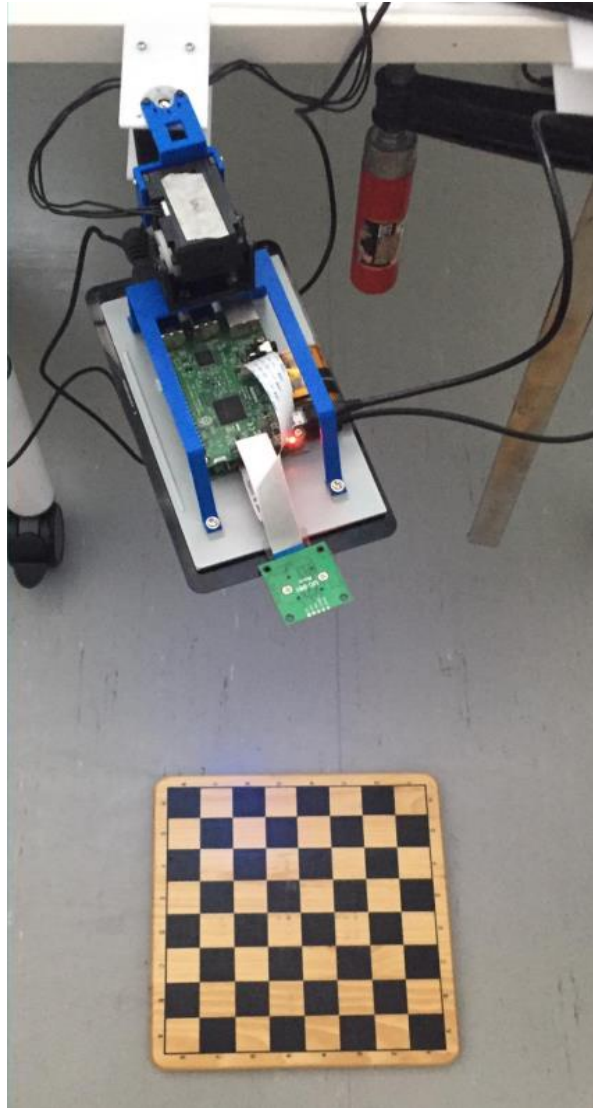
3.4 Testaus

Käyttötilannetta testattiin yhdessä laudantunnistusta työstävän ryhmän kanssa. Laudantunnistuksessa käytettiin näytön yläpuolelle sijoitettua kameraa. Niska liikkui kameraa edestakaisin, kunnes se tunnisti näkökentässä shakkilaudan, jonka jälkeen se lopetti skannaamisen.

Eri komponenttien, laudantunnistuksen ja niskan ohjaamisen, kommunikointiin käytettiin Robot Operating System (ROS)-ympäristöä. ROS tarjoaa valmiin ympäristön erilaisten robotiikkaan liittyvien sovellusten luomiseen ja niiden väliseen kommunikointiin. Testauksen ajaksi Raspberryn asennettiin Ubiquity Robotics:n ylläpitämä, raspberrystä varten luotu Ubuntu 16.04-käyttöjärjestelmän versio, jonka mukana toimitetaan ROS-ympäristö valmiiksi asennettuna. Testaus tehtiin ROS:n “Kinetic Kame” versiolla.

Skannausfunktioista luotiin ROS-moduuli, joka ajettaessa suorittaa skannaussilmukkaa. Moduuli kommunikoi laudantunnistuksen tarjoaman ROS-moduulin kanssa, joka palauttaa “True” arvon kun kameran ottamassa kuvassa näkyy shakkilauta. Moduuli olisi myös palauttanut nappuloiden tilanteen, mutta sitä ei oltu integroitu nykyiseen versioon. Ohjausmoduulin toiminta siis vastaa kuvassa TK esitettyä tilakaaviota, jossa [face_recognized]-ehdot on korvattu laudantunnistusehdolla, ja “Hello” tila ohitettiin kokonaan.

Osat onnistuivat kommunikoimaan keskenään, ja voitiin todeta että niskaa pystyy käyttämään ympäristön skannaamiseen konenäön avulla. Niska liikkui edestakaisin samalla kun kamera otti kuvia ympäristöstä, ja lopetti liikkumisen kun laudantunnistusmoduuli ilmoitti, että lauta on tunnistettu. Niskan liikkuminen oli testissä rajoitettu sivuttaissuuntaan, jolloin ylösalaissuunnan määrittävän nivelen kulma täytyi alustaa sopivaan kulmaan katsomaan alaspäin suorituksen alussa, kuten esitetty kuvassa 14.



Kuva 14. Kamera pysähtyneenä laudan yläpuolella

4. JATKOKEHITYS

Kehityksen aikana nousi paljon ideoita, joita ei ajan tai laitteiden puutteen takia pystytty toteuttamaan. Parannusten avulla päästä tulisi muun muassa ihmismäisempi, sen ohjaus monipuolistuisi ja päästä tulisi kestävämpi.

Rungon suunnittelussa ei jätetty tarpeeksi varaa muutoksille myöhemmin. Joitain ongelmia ei onnistuttu ottamaan huomioon suunnitteluvaiheessa, ja ongelmien korjaaminen osoittautui vaikeaksi. Servojen kytkentöjen yhtenäistäminen jäi nykyisen projektin puitteissa vajaaksi. Piirilevyn valmistaminen esimerkiksi syövyttämällä helpottaisi systeemin kasaamista. Isojen servojen lisäksi kaulaan olisi voitu lisätä pienempiä servoja hienovaraisempia liikkeitä varten.

Servojen ohjauksessa voitaisiin jatkossa käyttää paremmin hyväksi servojen tarjoamia ominaisuuksia ja toteuttaa yhtäaikaista rinnakkain toimintaa. Servojen tilaa lukemalla voidaan tehdä toiminnasta luotettavampaa ja tarkempaa. Nivelten ohelle voitaisiin tehdä kasvot, jotka kommunikoivat ihmisen kanssa erilaisin ilmein. Näin voitaisiin tutkia kaulan luonnollisuutta itse käyttötilanteessa.

4.1 Runko

Tulevia iteraatioita suunnitellessa täytyy servojen vääntövoima ottaa huomioon tarkemmin. Ylösalaisessa liikkeessä täytyy minimoida vaadittavan voiman määrä. Liikeradan tarjoava nivel kannattaa siis sijoittaa nivelsarjan viimeiseksi osaksi, lähimmäksi kasvoja.

Näytön asennon korjaamiseksi kiinnikkeet voisi suunnitella symmetrisiksi, jolloin näytön asentoa voisi kääntää tarvittaessa. Erilaiset kulmat lisäävät eri sovellusvaihtoehtoja. Toisaalta nivelsarjaan voisi lisätä myös kolmannen nivelen, joka sallisi näytön pyörittämisen. Uusia osia tehtäessä olisi siis hyvä tehdä kiinnikkeistä jossain määrin keskenään yhteensopivia. Näin servojen paikkaa voidaan vaihtaa keskenään tarvittaessa, tai nivelsarjaan voitaisiin helposti lisätä uusia niveliä.

Nykyinen varsi ei sovellu kovin hyvin itse laudan tarkasteluun. Kasvojen suuntautuessa alaspäin kamera on liian lähellä laudaa. Varren pituutta ei ole tarpeeksi helppoa kasvattaa, ja sen kiinnittäminen tukevaan alustaan on tällä hetkellä epävarmaa.

Nykyisessä ratkaisussa yksi servoista on kiinnitetty suoraan varteen, josta johtuen pidemmän varren tekeminen vaatisi kokonaan uuden servokannakkeen tekemisen. Lisäksi varren vaihtaminen vaatii monen osan purkamisen. Jos servokannake olisi kiinnitettynä varteen jonkinlaisella yksinkertaisella kiinnitysmekanismilla, olisi erilaisten varsien kehittäminen suoraviivaisempaa.

4.2 Servot

Koska piirilevy oli suunniteltu, mutta sitä ei saatu toteutettua, jatkokehityksen kannalta piirilevy voitaisiin valmistaa esimerkiksi syövyttämällä. Robotin kannalta piirilevyn integrointi helpottaisi testausta ja tekisi niskasta kestävämmän. Lisäksi testattaessa kytkentöjä ei tarvitsisi tehdä aina uudelleen, jolloin kytkentävirheet vähenisivät ja itse testaamiseen voidaan käyttää enemmän aikaa.

Projektin alussa harkittiin myös pienempien servojen, Dynamixelin XL-320 -servojen, käyttämistä, mutta päädyimme nykyisiin servoihin yksinkertaisuuden vuoksi. Jatkokehityksen kannalta pienemmät servot voisivat olla parempia niveliä paikkoihin, joissa voimaa ei tarvita niin paljoa mitä nykyiset XL430-W250-T -servot antavat. Tällöin ne muun muassa vähentäisivät tehonkulutusta ja keventäisivät rakennetta.

4.3 Ohjaus

Projektissa servojen yhtäaikainen liikuttaminen ei ole mahdollista johtuen koodin suoritustavasta. Jatkokehitysideana kummankin servon liikutus voitaisiin jakaa eri prosesseihin, jotka suorituisivat rinnakkain. Tällöin ne voisivat toimia toisistaan riippumatta. Tämä voitaisiin toteuttaa erillisillä ROS-palveluilla eri servoille, jotka kommunikoivat ohjaavan prosessin kautta mutta toimisivat muuten itsenäisesti.

Niskan liikkumisen inhimillistäminen lisäisi käyttökokemuksen mukavuutta. Jatkossa voitaisiin testata, kuinka koehenkilöt arvostelevat liikeprofiilien inhimillisyyden erilaisissa tilanteissa, kuten esimerkiksi jos robotti on yllätynyt vastustajan liikkeestä. Tuloksista riippuen paras liikeprofiili integroitaisiin lopulliseen robottiin, jolloin lopputulos olisi mahdollisimman käyttäjäystävällinen.

Ohjauksen ennakoitavuutta varten voitaisiin toteuttaa jonkinlainen kalibrointifunktion. Servoilta on mahdollista lukea niiden senhetkinen kuorma, jonka perusteella voidaan päätellä milloin liikerata kohtaa periksiantamattoman esteen. Kalibrointivaiheessa servoja ajetaan eri suuntiin kunnes ne törmäävät esteeseen, ja tämä kulma tallennetaan muistiin. Tallennettujen kulmien avulla voidaan luoda lailliset liikeradat niskalle, joita se ei missään suoritusvaiheessa pyri ylittämään. Kuorman mittaamista voidaan myös käyttää suorituksen aikaisena turvamekanisminä. Kun kuorma kasvaa liian suureksi esteen takia, voidaan servojen toiminta keskeyttää välittömästi ennen kuin ne ehtivät aiheuttaa vahinkoa runkoon tai itse servoihin.

4.4 Kasvot

Projektin lopulliset kasvot olivat vain yksinkertainen kuva näytöllä, joten tulevaisuudessa kasvoihin panostamalla saataisiin robotista huomattavasti ihmisläheisempi vaihtamalla varsinaisia kasvoja ja lisäämällä esimerkiksi mahdollisia ilmeitä, joita kasvot voisivat saada.

Näyttöön voisi myös lisätä mekaanisia komponentteja, kuten kulmakarvoja,

joita voitaisiin liikuttaa pienemmillä servoilla. Ne toisivat ulottuvuuden tunnetta kasvoille, kun vuorovaikutuspinta ihmisen ja robotin välillä olisi enemmän kuin pelkkä ruutu.

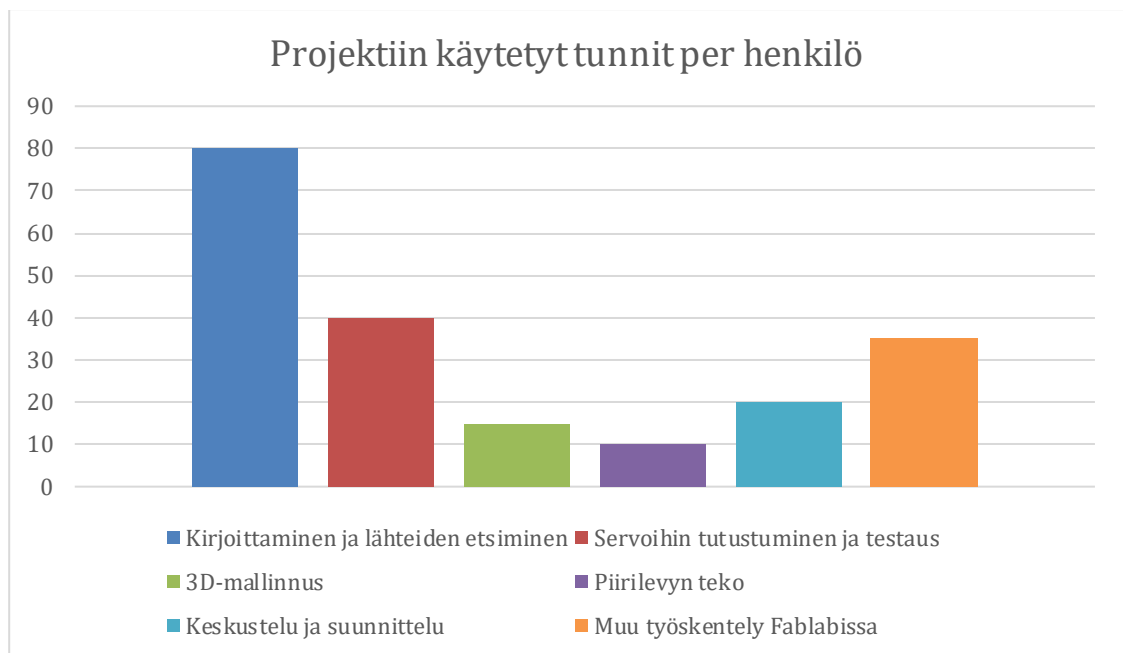
Kasvojen avulla niskan luonnollisuutta voitaisiin testata tarkemmin. Pelkästään nivelten liikkeillä niska jää persoonattomaksi koneeksi. Kasvot lisäämällä niskaan tulee ihmismäisyyttä, joka luo käyttäjälle tunteen siitä että ollaan tekemisissä älykkään kohteen kanssa.

Kun näyttöön piirtää kasvot jotka muuttuvat liikkeiden kanssa yhdessä, voidaan koko systeemin luonnollisuutta testata koehenkilöiden avulla. Testauksen voisi tehdä esimerkiksi Turingin testin tyyppisellä ratkaisulla. Koehenkilöiden annetaan seurata niskan toimintaa ensin puhtaasti tekoälyn ohjaamana, jonka jälkeen hän seuraa niskaa ihmisen ohjaamana. Ohjaamiselle voitaisiin toteuttaa helppo käyttöliittymä. Koehenkilön tehtävänä olisi erottaa ihmisen ohjaama niska tekoälystä. Lisäksi koehenkilö täyttäisi kyselyn niskan liikkeiden yleisestä luonnollisuudesta.

5. PROJEKTIN KUVAUS

Projektin työskentely koostui eri työskentelytavoista. Viikon alussa sovittiin yhdessä työnjaosta, jossa molemmat ottivat vastuulleen jonkin osa-alueen työstä. Osa-alueiden parissa työskentely tapahtui pääosin itsenäisesti. Kerran viikossa kokoonnuttiin Oulun Yliopiston Fablabiin testaamaan molempien töiden tuloksia. Fablab-tapaamisissa sovittiin mahdollisesti uudesta työnjaosta ja tarkennettiin työn vaatimuksia ja tavoitteita. Lisäksi Fablabissa toteutettiin kaikki rakenteeseen liittyvä työskentely, kuten 3D-tulostaminen ja laserleikkaus, Fablabissa olevien laitteiden avulla. Raportin kirjoittaminen tapahtui pääosin yhdessä internetpuheluiden välityksellä.

Keskimäärin työskentelyyn käytettiin keskimäärin noin 12 tuntia viikossa. Työskentelyn alkuvaiheessa ajankäyttö painottui taustan kirjoittamiseen. Työn keskivaiheilla Fablabissa vietettiin pidempiä aikoja. Loppuvaiheessa itsenäinen työskentely lisääntyi ja toteutuksen kirjoittamista painotettiin enemmän. Henkilöä kohden työtunteja kertyi projektin aikana noin 200 tuntia. Kuvassa 15 näkyy tuntien jakautuminen projektin aikana.



Kuva 15. Projektiin käytetyt tunnit ja niiden jakautuminen

6. YHTEENVETO

Projektin aikana tehtiin onnistuneesti niska shakkia pelaavalle robotille. Niska koostui 3D-tulostetuista ja laserleikatuista osista, nivelinä toimivista servoista ja kokonaisuutta ohjaavasta Raspberry Pi-tietokoneesta sekä siihen kytketystä näytöstä. Näyttö toimi eräänlaisina kasvoina robotille ja näytön yläpuolelle sijoitettua kameraa oli mahdollista käyttää ympäristön havainnointiin.

Niska pystyi toimimaan osana kokonaisuutta tarjoamalla laudantunnistukselle skannausfunktion, jonka avulla pystyttiin etsimään shakkilautaa laajalta alueelta. Skannausfunktio toteutettiin ROS-moduulin avulla, ja sen yksittäisiä parametreja, kuten skannausaluetta, oli helppo muokata tilanteeseen sopivaksi. Ohjaus voidaan toteuttaa joko erillisen tietokoneen tai robotin osana olevan Raspberryn kautta, jolloin robotti toimii täysin itsenäisesti.

Projektin perusteella voidaan todeta, että 3D-tulostuksella ja laserleikkauksella sekä kahdella nivelellä voidaan toteuttaa monipuolinen robottiniska. Niskalla on laajat liikeradat, ja liikkeen nopeutta pystytään muuttamaan tilanteeseen sopivaksi. Nivelten avulla kasvoina toimiva näyttö ja näytön yläpuolelle sijoitettu kamera pystytään kohdistamaan haluttuun suuntaan. Tämä mahdollistaa niskan soveltamisen konenäköön liittyvissä projekteissa.

Projektin puitteissa tehtyä niskaa on mahdollista käyttää esimerkiksi liikkeiden luontevuuden ja kasvojen merkityksen tutkimiseen. Käytetyt servot tarjoavat laajat vaihtoehdot liikkeiden tarkkaan ohjaamiseen, ja näytölle on mahdollista piirtää erilaisia kasvoja. Jatkokehitystä voisi tehdä osien kiinnityksen helpottamisen, yleisen luotettavuuden ja käyttömukavuuden osalta.

LÄHTEET

- [1] Ethem Alpaydin, "Introduction to machine learning, third edition", MIT Press 2014 3rd ed
- [2] Dimitrios I. Kosmopoulos, "A Design Framework for Sensor Integration in Robotic Applications", National Centre for Scientific Research "Demokritos
- [3] Murphy, R., Murphy, R. R., & Arkin, R. C. (2000). Introduction to AI robotics. MIT press.
- [4] Hockstein, N.G., Gourin, C.G., Faust, R.A. et al. J Robotic Surg (2007) 1: 113. <https://doi.org/10.1007/s11701-007-0021-2>
- [5] Yang G., McNutt M. "Robotics takes off". Science. 10 Jun 2016: 1255. 10.1126/science.aag2724
- [6] Dargahi, J., & Najarian, S. (2005). Advances in tactile sensors design/manufacturing and its impact on robotics applications - a review. The Industrial Robot, 32(3), 268-281. Retrieved from <https://search.proquest.com/docview/216993325?accountid=13031>
- [7] Aiguo Song, Guangming Song, Daniela Constantinescu, Lei Wang, and Qunjun Song, "Sensors for Robotics," Journal of Sensors, vol. 2013, Article ID 293656, 2 pages, 2013. <https://doi.org/10.1155/2013/293656>.
- [8] Ravinder S. Dahiya, Maurizio Valle, "Robotic Tactile Sensing: Technologies and System", p.3-4
- [9] Mehrholz J, Pohl M, Platz T, Kugler J, Elsner B. Electromechanical and robot-assisted arm training for improving activities of daily living, arm function, and arm muscle strength after stroke. Cochrane Database of Systematic Reviews 2018, Issue 9
- [10] Thomas B. Sheridan, "Human–Robot Interaction: Status and Challenges", Massachusetts Institute of Technology, Cambridge
- [11] Kerstin Dautenhahn, "Socially intelligent robots: dimensions of human–robot interaction". 362. Philosophical Transactions of the Royal Society B: Biological Sciences. <http://doi.org/10.1098/rstb.2006.2004>
- [12] Bruce G. Buchanan, "A (Very) Brief History of Artificial Intelligence". AI Magazine Volume 26 Number 4 (20056) <https://doi.org/10.1609/aimag.v26i4.1848>
- [13] Oleg Sergiyenko, Vera Tyrsa, Wendy Flores-Fuentes, Julio Rodriguez-Quiñonez, and Paolo Mercorelli, "Machine Vision Sensors," Journal of Sensors, vol. 2018, Article ID 3202761, 2 pages, 2018. <https://doi.org/10.1155/2018/3202761>.
- [14] S Huang, T. (2019). Computer Vision: Evolution and Promise.

- [15] Chan-Maestas, H., & Sofge, D. A. (2011, December). Tactile Sensor System Processing Based On K-means Clustering. In *Machine Learning and Applications and Workshops (ICMLA)*, 2011 10th International Conference on (Vol. 1, pp. 287-292). IEEE.
- [16] *Magnetic Sensors and Their Applications*: James Lenz and Alan S. Edelstein
- [17] Bronkhorst, A.W. *Atten Percept Psychophys* (2015) 77: 1465. [https://doi-org.pc124152.oulu.fi:9443/10.3758/s13414-015-0882-9](https://doi.org/pc124152.oulu.fi:9443/10.3758/s13414-015-0882-9)
- [18] *Proceedings of the National Academy of Sciences* Aug 2015, 112 (34) 10595-10598; DOI: 10.1073/pnas.1502276112
- [19] Jazar, R. N. (2010). *Theory of applied robotics: kinematics, dynamics, and control*. Springer Science & Business Media.
- [20] Craig, John J. *Introduction to robotics: mechanics and control*. Vol. 3. Upper Saddle River, NJ, USA.: Pearson/Prentice Hall, 2005.
- [21] Robert F. Shepherd, Filip Ilievski, Wonjae Choi, Stephen A. Morin, Adam A. Stokes, Aaron D. Mazzeo, Xin Chen, Michael Wang, and George M. Whitesides “Multiga it soft robot”, *Proceedings of the National Academy of Sciences* Dec 2011
- [22] Rich, S. I., Wood, R. J., & Majidi, C. (2018). Untethered soft robotics. *Nature Electronics*, 1(2), 102.
- [23] Dimeas, F. (2018). Manipulator performance constraints in human-robot cooperation. *Robotics and Computer Integrated Manufacturing*, 50, pp. 222-233.
- [24] Dautenhahn, K., Nehaniv, C. L., Walters, M. L., Robins, B., Kose-Bagci, H., Mirza, N. A., & Blow, M. (2009). KASPAR—a minimally expressive humanoid robot for human–robot interaction research. *Applied Bionics and Biomechanics*, 6(3-4), 369-397.
- [25] Kanda, T., Ishiguro, H., Imai, M., & Ono, T. (2003, August). Body movement analysis of human-robot interaction. In *IJCAI* (Vol. 3, pp. 177-182).
- [26] Daron Acemoglu, Pascual Restrepo “Robots and Jobs: Evidence from Us Labor Markets”, NBER Working Paper No. w23285
- [27] Roy, N., Baltus, G., Fox, D., Gemperle, F., Goetz, J., Hirsch, T., ... & Thrun, S. (2000, May). Towards personal service robots for the elderly. In *Workshop on Interactive Robots and Entertainment (WIRE 2000)* (Vol. 25, p. 184).
- [28] Sharkey, Noel. "The ethical frontiers of robotics." *Science* 322.5909 (2008): 1800-1801.
- [29] Patrick Lin, Geroge Bekey, Keith Abney, “Robots In War: Issues Of Risk And Ethics”