

Miska Romppainen

# OHJAUSRAJAPINNAN TOTEUTUS ITSE- VALMISTETULLE ROBOTTIKÄSIVAR- RELLE

Kandidaatintyö  
Tekniikan ja luonnontieteiden tiedekunta  
Jukka Yrjänäinen  
Joulukuu 2022

# TIIVISTELMÄ

Miska Romppainen: OHJAUSRAJAPINNAN TOTEUTUS ITSEVALMISTETULLE  
ROBOTTIKÄSIVARRELLE (CONTROL INTERFACE FOR SELF-BUILT ROBOTICARM)  
Tampereen yliopisto  
Automaatiotekniikka tutkinto-ohjelma  
Kandidaatintyö  
Joulukuu 2022

---

Työn tarkoituksena oli tutustua robotiikan perusteisiin ja hyödyntämällä robotiikan perusteita rakentaa ohjattava robottikäsi. Lisäksi työn motiivina selvitettiin, että yksinkertaisen robottikäsi valmistukseen ei tarvita suuria budjetteja tai kehitystiimejä. Yksi ihminen voi rakentaa toimivan, pienellä budjetilla valmistetun robottikäsi, joka pystyy suorittamaan yksinkertaisia tehtäviä.

Työssä valmistettiin robottikäsi. Rakennetulle robottikäsi luotiin ohjaukseen ja moottorienohjain, joiden avulla robottikäsi pystyttiin ohjaamaan. Työssä esitetään valmistetun robottikäsi suunnittelu-, ohjelmointi- ja rakennusprosessit sekä robottikäsi komponenttien ja materiaalien valinta. Robottikäsi on avoimen lähdekoodin MeArm-robotti ja sen ohjelmoinnissa hyödynnettiin Agustian et al. tekemää avoimen lähdekoodin MATLAB-ohjelmaa. MATLAB-ohjelma tarjosi MeArm-robotille rakennetun MATLAB mallin sekä pseudoversio Jacobin (käänteinen Moore–Penrose) ratkaisuun perustuvan kinematiikan ratkaisun, jota voitiin hyödyntää reitinsuunnittelussa.

Robottikäsi ohjaava ohjelmisto on jaettu kahteen osaan, jossa moottorienohjain on rakennettu Arduino-mikro-ohjaimen ja ohjaukseen ja reitinsuunnittelu on rakennettu MATLAB:illa. Moottorienohjaimen tarkoitus on säädellä robottikäsi nivelten nopeuksia ja varmistaa, että nivelet saavuttavat niille suunnitellut asennot ajallaan, ryntäämättä kohteeseen. Moottorienohjaimelle voidaan syöttää komentoja mistä tahansa ulkoisesta lähteestä, jolloin robottikäsi voidaan ohjata käyttämällä mitä tahansa ulkopuolista reitinsuunnitteluohjelmaa. MATLAB-ohjelmaan on rakennettu ohjaukseen ja reitinsuunnittelu luomalla sekvenssilistä robottikäsi asennoista eri ajanhetkillä, jonka jälkeen se lähetetään moottorienohjaimelle. Työssä tutkittiin robottikäsi ja sen ohjausta yksinkertaisella kappaleensiirtotestillä, joka suoritettiin onnistuneesti.

Työssä pohdittiin valmiin robottikäsi kehitys- ja käyttökohteita. Tulokseksi saatiin, että rakennettu robottikäsi ei sovellu jatkuvaan käyttöön tai teollisuusympäristöön, mutta valitsemalla erilaiset moottorit, materiaalit ja jatkokehittämällä ohjelmistoa, voitaisiin sitä käyttää esimerkiksi suorittamaan jatkuvia yksinkertaisia kappaleensiirtoprosesseja.

Avainsanat: Robottikäsi, Robotiikka, Ohjelmointi, Arduino, MATLAB, MeARM

# SISÄLLYSLUETTELO

1. JOHDANTO .....	1
2. ROBOTIKÄSIVARREN TEORIA .....	3
2.1. Robottikäsi .....	3
2.2. Päätyväline ja asento .....	4
2.3. Siirtyminen .....	4
2.4. Pseudoinversio Jacobin menetelmä .....	6
3. TYÖN SUORITUS, SUUNNITTELU JA RAKENNUS .....	8
3.1. Työn suunnittelu .....	8
3.2. Arduinon valinta .....	10
3.3. Rakennusmateriaali .....	12
3.4. Moottorien mitoitus .....	13
3.5. Kustannukset .....	14
3.6. Rakentaminen .....	14
3.7. Elektroniikka .....	16
3.8. Ohjelmointi .....	17
3.8.1. Moottorienohjain: Arduino .....	18
3.8.2. Ohjaukseen liittymä: MATLAB .....	20
4. ROBOTIKÄSIVARREN TOIMINNALLISUUDEN TESTAUS .....	22
5. YHTEENVETO .....	24
LÄHTEET .....	26

# KUVALUETTELO

KUVA 1. VIISI KAUPALLISESTI USEIN KÄYTETTYÄ ROBOTIRAKENNETTA [7].....	4
KUVA 2. ARDUINO MKR IOT BUNDLE, ARDUINON ALOITUSPAKKAUS .....	11
KUVA 3. KUVA LASERLEIKKAUS PROSESSISTA.....	15
KUVA 4. MOOTTORIEN SIJAINNIT .....	15
KUVA 5. KAAVIOKUVA JÄRJESTELMÄSTÄ.....	16
KUVA 6. ARDUINON KYTKENTÄKAAVIO .....	16
KUVA 7. JÄRJESTELMÄN LOHKOKAAVIO .....	17
KUVA 8. PROJEKTIN PÄIVITETTY KÄYTTÖLIITTYMÄ .....	21

# LYHENTEET JA MERKINNÄT

3D	engl. Three-Dimensional space, kolmiulotteinen avaruus
I/O	engl. Input/Output, sisään- ja ulostulo
IoT	engl. Internet of Things, esineiden internet
ISO	engl. International Organization for Standardization, kansainvälinen standardisoimisjärjestö
PWM	engl. Pulse Width Modulation, pulssileveys modulaatio
WiFi	engl. Wireless Fidelity, langaton verkko

# 1. JOHDANTO

Marvin Minskyn sanoin vuonna 1994, ”Perivätkö robotit maan? Kyllä, mutta he tulevat olemaan meidän lapsiamme” [1, s. 108] . Robotiikkaa on esiintynyt vuosikausia niin ideoina kuin toimivina rakennelmina. Robotiikan käyttökohteet ovat vain monipuolistuneet 2000-luvulla. Oxford Economicsin julkaiseman artikkelin mukaan vuoden 2010 jälkeen teollisuuden roboottikanta on yli kaksinkertaistunut [2]. Robottien kyvykkyys on parantunut huomattavasti, koska ne muun muassa kykenevät suorittamaan monimutkaisempia laskelmia [3]. Tämä mahdollistaa niiden suorittaa mitä monimutkaisempia tehtäviä, pakettien toimituksista, kuten Amazonin jakeluroboteista [4] kirurgisiin operaatioihin, kuten DaVinci Xi kirurgisiin leikkausrobotteihin [5]. Tällaisten monimutkaisten robottien luominen on suuri prosessi [6], mutta yksinkertaisempien robotien luomiseen ei vaadita suuria budjetteja tai kehitystiimejä. Yksi henkilö voi suunnitella, ohjelmoida ja rakentaa itselleen robotin, joka suorittaa yksinkertaisia toimintoja pienelläkin budjetilla.

Tässä työssä tutkitaan robottikäsivarren suunnittelu-, ohjelmointi- ja rakennusprosessia käyttäen Arduino-mikro-ohjainsarjaa sekä MATLAB-käyttöliittymää. Työn motiivina oli tehdä monimuotoinen työ, joka opettaa ja antaa kokemuksen siitä, millaisen työn takana on rakentaa niin mekaniikka, elektroniikka kuin ohjelmisto pieneen robottiin. Ennen työn tekoa ohjelmistosuunnittelun näkökulma herätti kiinnostusta. Työssä tarkastellaan pienellä budjetilla valmistetun robotin kehitystyötä ja esitellään tämän robottikäsivarren valmistusprosessia sekä vastataan kysymyksiin, millaisen työ takana on toimivan robotin rakentaminen sekä pystyykö pienellä budjetilla rakennettu robotti suorittamaan yksinkertaisia tehtäviä, kuten siirtämään jonkin kappaleen paikasta toiseen. Lisäksi työssä esitellään robottikäsivarren suunnittelu-, rakennus- ja ohjelmointiprosessit. Työn tarkoituksena on luoda robottikäsivarsi, jolle voidaan syöttää toimintalistoja ulkoisesta lähteestä ja liikuttaa käsivartta sille annettujen toimintalistan parametrien avulla. Tällöin robottikäsivarsi pystyy suorittamaan erilaisia sille suunniteltuja tehtäviä, kuten esimerkiksi siirtämään kappaletta. Työssä esitellään lisäksi robottikäsivarren toiminnallisuuksia ja pohditaan sen toimintaa eri käyttöympäristöissä.

Toisessa luvussa esitellään robotiikan taustaa, joka lukijan on hyvä ymmärtää ennen työhön perehtymistä. Kolmannessa luvussa esitellään työssä valittu, robotin toiminnalle, keskeinen komponentti, Arduino. Arduino on mikro-ohjain, joka mahdollistaa robottia lii-

kuttavien moottorien ohjaamisen. Kolmannessa luvussa esitellään työn suoritus eli robottikäsivarren suunnittelu-, ohjelmointi- ja rakennusprosessit, joissa esitellään muun muassa osien vaatimuksia, osien valintaa, robotin malli, moottoreita sekä esitellään rakentamisen työvaiheet, materiaalien työstöt ja robotin mekaniikan rakennus. Luvussa 3.8 esitetään ohjelmiston hierarkia ja erinäiset ohjelmointiratkaisut, kuten toimintalistan luonti MATLAB-käyttöliittymässä sekä sen jäsentely Arduinossa. Lisäksi luvussa vastataan seuraaviin kysymyksiin: kuinka kaksi ohjelmaa on saatu kommunikoidaan keskenään ja mitkä ovat ohjelmien vastuualueet? Neljännessä luvussa esitellään robotin testaus ja testissä ilmenneet robotin toiminnan huomioid. Viidennessä luvussa on yhteenveto projektista. Luvussa 5 kerrotaan projektissa valmistunut lopputulos ja pohditaan, miten rakennettua robottikäsivartta voitaisiin käyttää eri käyttöympäristöissä. Lisäksi pohditaan robottikäsivarren eri jatkokehityskohteita sen toiminnallisuuksien ja ominaisuuksien parantamiseksi.

## 2. ROBOTTIKÄSIVARREN TEORIA

Vuonna 2021 päivitetty ISO-standardi ISO 8373:2021 [7] määrittää robottien ja robotiikan sanaston sekä niihin liittyvät määritelmät. ISO 8373:2021:n [7] mukaan robotti on määriteltä seuraavalla tavalla: ”Robotti tarkoittaa ohjelmoitua käyttömekanismia, jolla on tietty autonomia, liikkumisen, manipuloinnin tai paikantamisen suorittamiseksi.” Robotiikka on puolestaan määriteltä robottien suunnittelun, valmistuksen ja soveltamisen tieteenä ja käytäntönä. Perinteisesti robotti rakentuu nivelistä ja jäykistä kappaleista nivelten välillä. Robotteja ohjataan liikuttamalla tai kiertämällä niveliä. Nivelten ja jäykkien kappaleiden liikkeiden tulosta voidaan määrittää robotin asento tai vaihtoehtoisesti robotin asennosta voidaan määrittää nivelten tarvitsemat liikkeet, näitä kutsutaan eteenpäin suuntautuvaksi ja käänteiseksi kinematiikaksi.

Tässä työssä robottien nivelten kulmien ratkaisemiseen on käytetty pseudoinversio Jacobin menetelmää (engl. *pseudoinverse Jacobian method*) eli Moore–Penrose-menetelmää. Tässä luvussa esitetään työlle keskeisiä robotiikan teorioita, joiden avulla voidaan esitellä teoriaa MATLAB-koodissa käytetystä pseudoinversio Jacobin menetelmästä.

### 2.1. Robottikäsivarsi

Robotin niveliä on kahdenlaisia:

1. translaatio (merkitään T) eli nivel, joka pystyy liukumaan ulos tai sisäänpäin, ja
2. rotaatio (merkitään R) eli nivel, joka pystyy pyörimään.

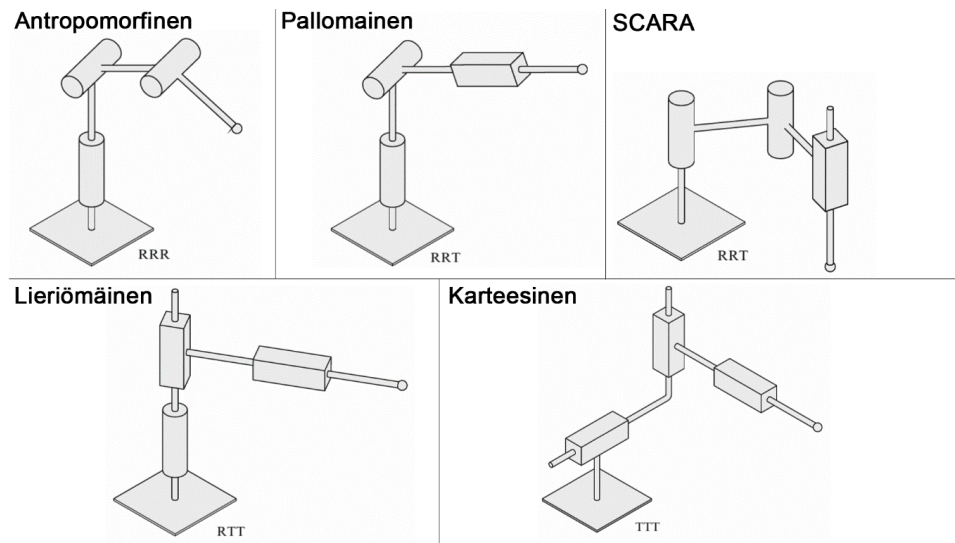
Nivelyhdistelmien avulla voidaan ilmoittaa robottikäsivarren rakenne, esimerkiksi yhdistäen kolme rotaatio (pyörivää) niveltä saadaan RRR-nivelyhdistein robotti.

Robottikäsivarsien yhteydessä käytetään usein termiä vapausaste (engl. *DOF, Degree of Freedom*). Vapausasteiden lukumäärä määritellään riippumattomien koordinaattien lukumääränä, jotka ovat tarpeen tarkasteltavan partikkelin sijainnin kuvauksessa. [8, s. 2] Esimerkiksi viivaa pitkin etenevä partikkeli tai heiluri jäykällä segmentillä on yhden vapausasteen järjestelmä. Robotiikassa vapausasteilla tarkoitetaan robotin mahdollisia liikesuuntia. Yhdellä nivelellä voidaan siis sanoa olevan yksi vapausaste, ja niveliä yhdistämällä voidaan luoda järjestelmiä, joissa on useita vapausasteita.

Bajd esittää kirjassaan *Robotics* [8], että robottikäsivarrella, jolla on kolme vapausastetta, on olemassa rajoitettu määrä nivelyhdistelmiä, jotka yhdessä johtavat 36 eri robottikäden rakenteeseen. Niistä vain 12 on toiminnallisesti erilaisia ja niistä 5 eri rakennetta



on usein kaupallisesti saatavilla: antropomorfisia, pallomaisia, SCARA, lieriömäisiä ja karteesisia. Nämä voidaan nähdä kuvassa 1.



**Kuva 1.** Viisi kaupallisesti usein käytettyä robottirakennetta [8, s. 5–6]

Tässä työssä on rakennettu antropomorfisen rakenteen omaava, kuvan 1 mukainen robotti, jolla on kolme vapausastetta. Antropomorfinen robotti rakentuu kolmesta rotaatiotavasta. Antropomorfisen robotin rakenne voidaan kuvata siinä olevan tyyppiä RRR.

## 2.2. Päätyöväline ja asento

Robotin niin kutsuttu päätyöväline (engl. *end-effector*) tarkoittaa robotin liikuttamaa toimilaitetta tai työkalua, jonka avulla robotti suorittaa tehtävänsä. Päätyövälineen sijainnin ja suunnan summa on robotin asento. Robotin asentoon vaikuttavat robotin nivelien saavuttamat kulmat ja tilat. Robotin nivelien kulmien perusteella voidaan laskea päätyövälineen sijainti, mitä kutsutaan eteenpäin suuntautuvaksi kinematiikaksi (engl. *forward kinematics*), tai päätyövälineen sijainnin perusteella voidaan laskea nivelien kulmat tätä kutsutaan käänteiskinematiikaksi (engl. *inverse kinematics*).

Robotin asento saattaa vaikuttaa sen nivelten vapausasteisiin. Robotin asento voi aiheuttaa tilanteen, jolloin vapausasteita saattaa puuttua tai nivelten toiminta saattaa olla rajattua. Tällaista tilannetta kutsutaan singulariteetiksi [9].

## 2.3. Siirtyminen

Päätyövälineen liike koostuu kahdesta erinäisestä liikkeestä, translaatiosta eli päätyövälineen liikkeestä eteen- tai taaksepäin tai rotaatiosta eli kiertymästä, joka voi tapahtua koordinaatistossa kolmen eri akselin ympäri. Homogeenisen matriisin avulla näiden

liikkeiden summa voidaan esittää kertolaskuna, jolloin saadaan lopullinen sijainti päätetyövälineelle eli siirtomatriisien tulo. [10 s. 13] Robotin liikuttaessa päätetyövälinettä koordinaatistonsa läpi täytyy valita koordinaatiston vertailukehys, eli niin sanottu karteesinen kehys, johon muita liikkeitä verrataan. Tällöin homogeenisten siirtomatriisien liike tapahtuu suhteessa karteesiseen kehykseen, jolloin voidaan ketjutetusti siirtää vertailukehystä koordinaatistossa. Tämä voidaan tehdä laskemalla translaatioiden ja rotaatioiden siirtomatriisien tulo ja alkuperäisen asennon homogeenisen matriisin tulo

$$\mathbf{H}_i = \mathbf{H} \cdot \mathbf{D}, \quad i \in N, \quad (1)$$

jossa  $\mathbf{H}_i$  on seuraavan askeleen homogeeninen matriisi,  $i$  askeleiden lukumäärä,  $\mathbf{H}$  alkuperäisen asennon homogeeninen matriisi ja  $\mathbf{D}$  on siirtomatriisien tulo. [10, s. 13] Tämän avulla vertailu eri koordinaatistojen välillä on mahdollista, koska robotin pohjan koordinaatiston suhde ensimmäiseen niveleen, ensimmäisen nivelen suhde seuraavan ja niin edelleen, kunnes voidaan verrata lopulta pohjan suhdetta nivelten kautta päätetyövälineeseen

$${}^0\mathbf{H}_i = \prod_{i=1}^i ({}^{i-1}\mathbf{H}_i \cdot \mathbf{D}_i), \quad i \in N, \quad (2)$$

jossa  ${}^{i-1}\mathbf{H}_i$  matriisi on edellisen kehyksen origon kehyksen suhde seuraavan kehyksen origoon ja  $\mathbf{D}_i$  matriisi on  $i$  askeleen siirtomatriisien tulo [10 s.18].

Käänteiskinematikan analyttisessä menetelmässä päätetyövälineen sijainnin perusteella voidaan ratkaista nivelten kulmat käyttäen algebrallisia ja geometrisiä ratkaisutapoja. Trigonometrinen funktioiden ratkaisut saattavat palauttaa useita eri ratkaisuja nivelten kulmille. Tätä kutsutaan redundanssiksi [11]. Kaksiulotteisessa tasossa näitä ratkaisuja voidaan kutsua ”kynärpää ylös”- ja ”kynärpää alas” -ratkaisuiksi [8, s. 37]. Tässä työssä rakennettu robotti käyttää vain ”kynärpää ylös” -ratkaisuja sen mekaanisen rakenteen vuoksi.

Denavit–Hartenberg-matriisi (lyh. DH) on kinemaattinen analysointi-, parametrisointi- ja merkintätyökalu. Sen avulla voidaan selvittää eteenpäin suuntautuvan kinematikan ratkaisun, jota voidaan hyödyntää käänteiskinematikan ratkaisussa. DH-parametrisoinnissa koordinaattikehys sijoitetaan näennäisesti jokaiselle nivelelle, mutta ei välttämättä nivelten kohdalle. Koordinaattikehysten muutosta voidaan verrata edellisen nivelen asetettuun koordinaattikehykseen. [12] DH-menetelmällä voidaan esittää jokainen nivel neljällä parametrillä  $(\theta, \alpha, a, d)$ , jossa  $\theta$  sekä  $\alpha$  ovat kulmia valitun z- ja x-akselin ympäri ja  $d$  sekä  $a$  etäisyyksiä z- ja x-akselin suuntaisesti. Taulukossa 1 on esitetty esimerkki DH-matriisista.

**Taulukko 1.** Esimerkki DH-matriisista

$i$	$\theta$	$\alpha$	$d$	$a$
<b>1</b>	$\theta_1$	$\alpha_1$	$d_1$	$a_1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
<b><math>i</math></b>	$\theta_i$	$\alpha_i$	$d_i$	$a_i$

DH-matriisista saadut arvot voidaan esittää muodossa, jolloin saadaan ratkaistua asennon homogeeninen matriisi

$$A_i = R_z(\theta_i)R_x(\alpha_i)T_x(a_i)T_z(d_i), \quad i \in N \quad (3)$$

jossa  $R_z(\theta_i)$  sisältää nivelten kiertymän Z-akselin ympäri,  $R_x(\alpha_i)$  sisältää kiertymän X-akselin ympäri,  $T_x(a_i)$  sisältää X-akselin suuntaisen translaation ja  $T_z(d_i)$  Z-akselin suuntaisen translaation. Valittua koordinaattikehystä merkitään  $i$ . Parametrisointi toimii vain, mikäli DH merkintätavan koordinaattikehykset asetetaan siten, että koordinaattikehyksen  $i$  X-akselin leikkaa koordinaattikehyksen  $i - 1$  Z-akselia sekä koordinaattikehyksen  $i$  X-akseli on kohtisuorassa koordinaattikehyksen  $i - 1$  Z-akseliin nähden. [12] Esittämällä jokainen asento DH-matriisilla voidaan sen eteenpäin suuntautuvan kinematiikan ratkaisu esittää koordinaattikehysten DH-matriisien tulona, jolloin voidaan esittää sen muodossa

$$A_i^{i-1} = \prod_{j=1}^i (R_z(\theta_j)T_z(d_j)T_x(a_j)R_x(\alpha_j)), \quad i \in N, \quad (4)$$

Mikäli jokainen robotin nivel on rotaatio nivel, kuten työssä valitussa antropomorfisessa robotissa. Jossa  $R_z(\theta)$  esittää nivelten kulmia ja pysyvät  $T_z(d)$ ,  $T_x(a)$  ja  $R_x(\alpha)$  vakioina [12].

## 2.4. Pseudoinversio Jacobin menetelmä

Analyttisestä ratkaisutavasta tulee vaikeampi ja haastavampi ratkaista, mikäli kinematiikan ongelma on monimutkainen. Numeraalisen ratkaisun iteratiivinen lähestymistapa saattaa olla parempi ratkaisu [8, s. 33]. Projektissa käytetty MATLAB-koodi käyttää pseudoinversio Jacobin menetelmää servomoottorien kulmien ratkaisuun, joka on numeraalinen menetelmä. Jacobin matriisin pseudoinversio menetelmä tunnetaan myös nimellä käänteinen Moore–Penrose-menetelmä [14].

Jacobin matriisi on matriisi, joka kuvaa muunnosta kahden avaruuden välillä. Robotiikassa Jacobin matriisilla voidaan esittää nivelten nopeuksien ja päätetyövälineen nopeuden yhteys. [8, s. 37][13] Voidaan siis ratkaista päätetyövälineen nopeuden ja suunnan eri nivelten nopeuksien perusteella. Jacobin matriisi voidaan esittää lyhyesti muodossa

$$\dot{x} = J(q)\dot{q}, \quad (5)$$

missä  $\dot{x}$  on sarakematriisi, joka edustaa päätetyövälineen nopeuksia,  $J(q)$  on mielivaltaisen kokoinen Jacobin matriisi, joka on nykyisen asennon funktio,  $\dot{q}$  on sarakematriisi, joka sisältää nivelten nopeudet [13]. Esimerkiksi xy-tasossa toimivan kaksi nivelisen robotin Jacobin matriisi voidaan esittää muodossa

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (6)$$

missä  $\dot{x}$  ja  $\dot{y}$  luovat päätetyövälineen sarakematriisin ja sisältää tason x ja y suuntaiset päätetyövälineen nopeudet,  $\dot{\theta}_1$  ja  $\dot{\theta}_2$  esittävät nivelten nopeudet ja Jacobin matriisi sisältää päätetyövälineen nopeuden ja suunnan nivelten nopeuksien perusteella [8, s. 37].

Tämän perusteella voidaan ratkaista nivelten nopeudet, mikäli tiedetään päätetyövälineen sarakematriisin  $\dot{x}$  kertomalla yhtälö (5) käänteisellä Jacobin matriisilla  $J(q)^{-1}$ . Mikäli Jacobin matriisi  $J(q)$  ei ole neliömatriisi. Voidaan sille laskea pseudokäänteinen matriisi. Tätä metodia kutsutaan pseudoinversioksi.

Pseudoinversiolla tarkoitetaan matriisin käänteistä objektia, joka voidaan määrittää kompleksiselle matriisille, vaikka se ei välttämättä olisi neliömatriisi. Pseudoinversio ratkaisun laskeminen laskee ratkaisuille, joille ei muuten saada ratkaisua, parhaan mahdollisen arvion. Käänteinen Jacobin matriisi voidaan esittää muodossa

$$\dot{q} = J(q)^{-1}\dot{x}, \quad (7)$$

jolloin käänteiskinematikan nojalla voidaan ratkaista nivelten kulmat loppuelimen koordinaattien perusteella.

Esitetään päätetyövälineen alkupisteen olevan  $x_a$  ja sen päätepisteen olevan  $x_p$ . Käänteiskinematikan prosessia suoritetaan, kunnes alkua ja päätepisteen välinen virhe on mahdollisimman pieni [14]

$$(x_p - x_a)^2 \approx 0. \quad (8)$$

Käyttäen päätekselihajotelmaa (engl. *Singular Value Decomposition, SVD*) saadaan laskettua parhaan mahdollisen arvion kaikille ratkaisuille [14].

## 3. TYÖN SUORITUS, SUUNNITTELU JA RAKEN- NUS

Projekti valmistukseen sisältyy työvaiheita, kuten työn suunnittelu sekä työn rakentaminen. Tässä luvussa esitetään työvaiheet. Työn suunnitteluun kuuluu sopivan mallin valinta, käytettävien materiaalien valinta, sopivien moottorien mitoitus ja moottorien valinta. Tämän jälkeen voidaan laskea työhön kuluneet kustannukset.

Työn rakentaminen alkaa osien hankinnalla ja robotin rakentamisella. Kun robotin runko on saatu rakennettua, voidaan kytkeä moottorit ja rakentaa robotin elektroniikka. Elektroniikan valmistuttua on rakennettava ohjelmisto, joka ohjaa robottia.

Tässä luvussa esitellään myös ohjelmisto sekä ohjelmointi prosessi. Ohjelmisto koostuu käyttöliittymän sisältävästä MATLAB koodista ja ohjausrajapinnan sisältävästä Arduino koodista. Ohjelmiston keskisenä osana on näiden välinen kommunikaatio.

### 3.1. Työn suunnittelu

Suunnittelutyö alkaa rakenteiden, materiaalien ja kustannusten määrittämisellä. Suunnittelun lähtökohtana oli luoda robotti, joka ei olisi liian haastava rakentaa, mutta sen rakenne mahdollistaisi yksinkertaisten tehtävien suorittamisen. Robottikäsivarren tulee pystyä toimimaan ja moottorien jaksaa liikuttaa robotin niveliä.

Projektin aloitusvaiheessa tärkeää on löytää haastavuudeltaan sopiva malli. Työn laajuuden vuoksi malli on valittu jo tehdyistä avoimen lähdekoodin (engl. *Open-Source*) malleista. Avoin lähdekoodi tarkoittaa lähdekoodia, jonka tekijänoikeuden haltija antaa käyttäjille oikeuden käyttää, tutkia, muuttaa ja jakaa lähdekoodia, suunnitteluasiakirjoja tai sisältöä kenelle tahansa ja mihin tahansa tarkoitukseen [15].

Työhön valittavia mahdollisia avoimenlähdekoodin rakennusmalleja vertailtiin. Vertailukohteina malleissa käytettiin rakentamisen haastavuutta, niveltenmäärää sekä designin tasoa. Työhön tuli valita malli, joka ei olisi liian haastava rakentaa, se sisältäisi tarpeeksi niveliä eri funktioiden toteuttamiseksi sekä siinä olisi ”levymäinen” design, jotta se voitaisiin laser leikata eikä osia tarvitsisi 3D-tulostaa. Haastavuusarvio perustuu arvioon työn rakentamisen haastavuudesta perustuen robotin nivelten määrään, käytettyyn materiaaliin sekä verkkosivuilla esitettyyn rakennusmateriaaliin. Vertailut mallit on esitetty taulukossa 2.

**Taulukko 2.** avoimenlähdekoodin mallien vertailutaulukko

Nimi	Saatavilla	Haastavuus-arvio	Niveliä	Design taso
<b>LittleArm</b>	<a href="https://www.instructables.com/LittleArm-V3-Arduino-Robot-Arm/">https://www.instructables.com/LittleArm-V3-Arduino-Robot-Arm/</a>	Keskitaso	2–3	Yksinkertainen
<b>EEZYbotARM MK3</b>	<a href="https://www.thingiverse.com/thing:2838859">https://www.thingiverse.com/thing:2838859</a>	Matala	3	Yksinkertainen
<b>Instructables 3D Printed Robot Arm</b>	<a href="https://www.instructables.com/3D-Printed-Robot-Arm/">https://www.instructables.com/3D-Printed-Robot-Arm/</a>	Keskitaso	4	Yksinkertaista korkeampi
<b>MeARM</b>	<a href="https://mearm.com/">https://mearm.com/</a>	Matala	3	Yksinkertainen
<b>Kauda</b>	<a href="https://www.diy-technology.com/010-kauda">https://www.diy-technology.com/010-kauda</a>	Korkea	5	Huolittelu
<b>DIY Arduino Robot arm</b>	<a href="https://thangs.com/designer/m/3d-model/38899">https://thangs.com/designer/m/3d-model/38899</a>	Keskitaso	3	Yksinkertaista korkeampi
<b>Zortrax</b>	<a href="https://library.zortrax.com/project/zortrax-robotic-arm/">https://library.zortrax.com/project/zortrax-robotic-arm/</a>	Korkea	4	Huolittelu
<b>Roboteurs</b>	<a href="https://3dprint.com/164591/roboteurs-3dp-six-axis-robot/">https://3dprint.com/164591/roboteurs-3dp-six-axis-robot/</a>	Erittäin korkea	6	Erittäin huolittelu
<b>Haddington Dynamics</b>	<a href="https://www.hdrobotic.com/">https://www.hdrobotic.com/</a>	Keskitaso	4	Huolittelu
<b>BCN3D Moveo</b>	<a href="https://www.bcn3d.com/bcn3d-moveo-the-future-of-learning-robotic-arm">https://www.bcn3d.com/bcn3d-moveo-the-future-of-learning-robotic-arm</a>	Korkea	3	Keskivertoa

Työssä päädyttiin valitsemaan MeARM 0.4 v –niminen avoimen lähdekoodin –malli, jonka on tehnyt *MeArm Robotics Labs* [16]. MeARM-robottikäsivarsi täyttää valintakriteerit. MeARM-robottikäsivarsi valittiin sen yksinkertaisen ja levymäisen designin takia, lisäksi se sisältää 3 liikkuvaa niveltä; alustan nivelen, ensimmäisen liitoksen nivelen ja toisen liitoksen nivelen. Nämä nivelet mahdollistavat yksinkertaisten funktioiden suorittamisen.

MeARM-malli tarvitsee neljä servomootoria toimiakseen: alustan servomootorin, 1. nivelen servomootorin, 2. nivelen servomootorin sekä päätetyövälineen servomootorin. Mallin levymäinen design mahdollistaa osien leikkaamisen 3D-tulostamisen sijasta. Osien leikkaus mahdollistaa myös eri rakennusmateriaalien valinnan, sillä työ voidaan rakentaa muustakin materiaalista kuin 3D-tulostus muovista. Parhaat mahdolliset moottorit ja materiaali saadaan tarkastelemalla eri vaihtoehtojen ominaisuuksia ja valitsemalla niistä parhaat.

### 3.2. Arduinin valinta

Arduinot ovat avoimen laitteistoon perustuva elektroniikka-alusta. Arduinin alkuperäiseen tutkimusryhmään kuuluivat Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino ja David Mellis, *Ivean Interaction design* -instituutissa Italiassa 2000-luvun alussa. [17] Ensimmäinen Arduino kehitettiin vuonna 2005 auttamaan suunnittelijaopiskelijoita luomaan toimivia prototyypppejä. Suunnittelijaopiskelijoilla ei ollut aikaisempaa kokemusta elektroniikasta tai mikro-ohjaimien ohjelmoinnista, joten Arduino kehitettiin luomaan matala kynnys digitaalisen maailman ja fyysisen maailman välillä [18]. Työssä Arduinoa käytetään ohjaamaan valittuja servomootoreita, se ohjelmoidaan toimimaan moottorienohjaimena.

Arduinoja on useita erilaisia. Niitä on luotu eri käyttökohteisiin ja eri Arduinoissa on näin eri komponentteja. Taulukossa 3 on vertailtu eri Arduino levyjen ominaisuuksia: prosessoreita, prosessorien kellotaajuutta, välimuistien kokoa (SRAM), Flash-muistien kokoa, IO-pinnien määrää, levyjen fyysistä kokoa sekä hintaa. Taulukossa 4 on listattu Arduino levyjen verkkosivut, mistä taulukon 3 tiedot ovat peräisin.

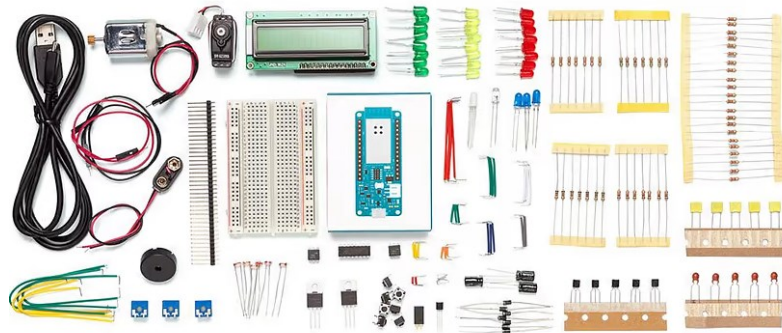
**Taulukko 3.** Arduinoiden vertailu

Malli	Proessori	Kello- taajuus	SRAM	Flash muisti	I/O-määrä	Fyysinen koko	Hinta
<b>Arduino Uno Rev3</b>	ATmega328P	16 MHz	2 KB	32 KB	20 (6 Analogista)	53,4 mm x 68,6 mm	24,00 €
<b>Arduino Nano 33 IoT</b>	SAMD21 Cortex®-M0+ 32bit low power ARM MCU	48MHz	32KB	256 KB	22 (8 Analogista)	18 mm x 45 mm	20,80 €
<b>Arduino Nano</b>	ATmega328	16 MHz	2 KB	32 KB	22 (8 Analogista)	18 mm x 45 mm	21,60 €
<b>Arduino MKR1000 WiFi</b>	SAMD21 Cortex-M0+ 32bit low power ARM MCU	48 MHz	32 KB	256 KB	20 (7 Analogista)	25 mm x 61,5 mm	41,20 €
<b>Arduino Mega 2560 Rev3</b>	ATmega2560	16 MHz	8 KB	256 KB	70 (16 Analogista)	53,3 mm x 101,52 mm	42,00 €

**Taulukko 4. Arduino levyjen verkkosivut**

Malli	URL
<b>Arduino Uno Rev3</b>	<a href="https://store.arduino.cc/products/arduino-uno-rev3">https://store.arduino.cc/products/arduino-uno-rev3</a>
<b>Arduino Nano 33 IoT</b>	<a href="https://store.arduino.cc/products/arduino-nano-33-iot">https://store.arduino.cc/products/arduino-nano-33-iot</a>
<b>Arduino Nano</b>	<a href="https://store.arduino.cc/products/arduino-nano">https://store.arduino.cc/products/arduino-nano</a>
<b>Arduino MKR1000 WiFi</b>	<a href="https://docs.arduino.cc/hardware/mkr-1000-wifi">https://docs.arduino.cc/hardware/mkr-1000-wifi</a>
<b>Arduino Mega 2560 Rev3</b>	<a href="https://store.arduino.cc/products/arduino-mega-2560-rev3">https://store.arduino.cc/products/arduino-mega-2560-rev3</a>

Tässä työssä valittiin Arduino MKR1000 malli sen prosessorin kellotaajuuden, muistin määrän sekä levyn WiFi ominaisuuksien perusteella. MKR1000 mallin hinta on hieman korkeampi kuin Nano 33 IoT mallilla, joka sisältää samoja ominaisuuksia ja tarjoaa WiFi ominaisuuden. WiFi lisäominaisuutta ei tässä työssä käytetty. Tarjoaa se kuitenkin jatkok kehitys mahdollisuuden luoda robotille esimerkiksi rajapinnan langattomalle ohjaukselle. Kuvassa 2 on hankitun Arduino MKR1000 aloituspakkauksen sisältö.



**Kuva 2. Arduino MKR IoT Bundle, Arduinon aloituspakkaus**

Arduino IDE on Arduinon oma kehitysympäristö. Kehitysympäristöä käytetään Arduinon ohjelmiston kirjoittamiseen. Ohjelmistojen kirjoittamiseen Arduino IDE käyttää C/C++ kieliä. Kehitysympäristö sisältää tuen eri kirjastojen käyttämiseksi ja oletuksena se sisältää kielten peruskirjastot sekä Arduinon omia kirjastoja. Eri ohjelmointilaudat sisältävät eri kirjastot perustuen niiden käyttötarkoitukseen, mutta niillä on useita yhteisiä metodeja, jotka löytyvät jokaiselta ohjelmointilaudalta, kuten servomootorien ohjaukseen tarkoitettu metodi "Servo". Esimerkki ohjelmointilautojen omista kirjastoista on Arduino MKR-sarjan "MKRWAN", "MKRGSM" sekä "MKRNB", jotka hyödyntävät MKR-sarjan yh-



teysominaisuuksia. Arduino IDE:ä käytetään ohjelmistojen kirjoittamisen lisäksi ohjelmiston lataamiseen Arduinon muistiin. Lukuisista kirjastoista ja niiden toiminnallisuuksista huolimatta Arduino on erittäin aloittelijaystävällinen. Yksinkertaisen ohjelman luominen ei vaadi käyttäjän ymmärrystä lisäkirjastoista. Ainoana lisämetodina tässä työssä käytettiin servojen ohjaukseen tarkoitettua metodia.

### 3.3. Rakennusmateriaali

Rakennusmateriaalia valittaessa, tulee ottaa huomioon materiaalin taipuvuus, kestävyys, paino sekä materiaalin hinta. Taulukossa 5 on esitetty eri materiaalien hintoja. Hinnat ovat peräisin FabLabin hinnastosta [19].

Fablab Tampere on Fablab-verkoston kuuluva valmistuslaboratorio, josta löytyy mm. materiaaleja, 3D-tulostimia, laserleikkureita ja työkaluja, joita voi käyttää omien projektien rakentamiseen. Fablab Tampere sijaitsee Tampereen Yliopiston Hervannan kampuksella [20].

**Taulukko 5.** *Materiaalien hintoja*

Materiaali	Paksuus (mm)	Koko (X x Y) (mm)	Hinta (€)
Akryyli, Kirkas	3,00	600 x 300	9,00
Akryyli, Opaali	3,00	600 x 300	10,00
Vaneri, Laservaneri	4,00	600 x 300	4,00
Vaneri, Ohutvaneri	2,00	610 x 610	12,00
Korkki	3,00	925 x 625	8,00
MDF	3,00	600 x 300	1,00

Projektin 3D-mallista on mahdollista laskea osien kokonaispinta-ala käyttämällä CAD-ohjelmistoa. Taulukossa 6 on esitetty osien kokonaispinta-ala, kun kaikki osat on asetettu yhdelle tasolle.

**Taulukko 6.** *Osien kokonaispinta-ala*

X (mm)	Y (mm)	Z (mm)	Tilavuus (cm <sup>3</sup> )	Pinta-ala (m <sup>2</sup> )
461,214	203,593	3,000	281,702	0,094

Taulukosta 6 saadaan tarvittavan materiaalin määrä. Mikä tahansa levy, joka on suurempi kuin noin 462 mm x 204 mm riittää. Osien mitoituksessa tulee myös huomioida paksuus 3 mm, jotta rakennusvaiheessa osien liitokset eivät jää liian löysiksi.

Materiaaliksi valittiin kirkas akryyli. Akryyli on muovia, kevyttä, taipuvaa, kestäväää painoonsa nähden, halpaa ja sen työstäminen on helppoa. Akryylin tiheydeksi on ilmoitettu

1,14 g/cm<sup>3</sup> [21]. Osien kokonaistilavuus voidaan lukea taulukosta 6. Tästä saadaan osien kokonaismassaksi 321,14 grammaa. Materiaaliksi olisi voitu valita myös esimerkiksi vaneri, mutta sopivan paksuista vaneria ei ollut saatavissa FabLabista, jossa materiaalihankinnat ja työstöt suoritettiin. Materiaalin työstöt tehtiin Epilog Laser Fusion 120W laserleikkurilla.

### 3.4. Moottorien mitoitus

Ennen kustannuksien laskemista, määritettiin käyttötarkoitukseen sopivimmat servomoottorit sekä materiaalit.

Servomoottorit ovat PWM-ohjattuja moottoreita. Niiden toiminta perustuu niiden vastaanottamaan sähkösignaalin pituuteen. Sähkösignaalin pituus määrittää servomoottorin kulman. Moottorien tarvitsema virta on riippuvainen moottorin koosta ja sen väntömomentista.

Työssä tarvittavien servomoottorien vaatimukset perustuvat servomoottorien lopulliseen sijaintiin, sille aiheutuvaan pysäytysmomenttiin sekä sen liikuttamaan kuormaan. Taulukossa 7 on luetteloitu mahdollisia eri servomoottoreita. Suurimmat huomioitavat kohdat ovat moottorien koko, paino, suurin mahdollinen pysähtymismomentti ("Max Stall Torque") sekä hinta.

**Taulukko 7.** Servomoottorien vertailutaulukko

Malli	Paino	Koko (mm)	Toiminta- vauhti	Toiminta- jännite	Stall Torque	Max Stall Torque	Hinta (€)
<b>SG-90</b>	9 g	32 x 30 x 12	0.12 s/60°	4.8V-6V (5V)	1.2 kg/cm	1.6 kg/cm (6V)	5,45 €
<b>MG-90S</b>	13.4 g	35.5 x 32.5 x 12	0.1 s/60°	4.8V-6V (5V)	1.8 kg/cm	2.2 kg/cm (6V)	4,90 €
<b>VTS-08A</b>	43 g	43 x 23 x 38.2	0.33 s/60°	4.8V-6V (5V)	5 kg/cm	6 kg/cm	6,03 €
<b>MG995</b>	55 g	40.7 x 19.7 x 42.9	0.20 s/60°	4.8V-6.6V (5V)	9.4 kg/cm	11 kg/cm	10,45 €
<b>MG996R</b>	55 g	40.5 x 20 x 39.5	0.17 s/60°	4.8V-7V (5V)	13.5 kg/cm	15.2 kg/cm	7,90 €

Malli	Saatavilla
<b>SG-90</b>	<a href="http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf">http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf</a>
<b>MG-90S</b>	<a href="https://www.electronicoscaldas.com/datasheet/MG90S_Tower-Pro.pdf">https://www.electronicoscaldas.com/datasheet/MG90S_Tower-Pro.pdf</a>
<b>VTS-08A</b>	<a href="http://www.vigorprecision.com.hk/uploadfile/20120530/20120530173638250.pdf">http://www.vigorprecision.com.hk/uploadfile/20120530/20120530173638250.pdf</a>
<b>MG995</b>	<a href="https://html.alldatasheet.com/html-pdf/1132435/ETC2/MG995/109/1/MG995.html">https://html.alldatasheet.com/html-pdf/1132435/ETC2/MG995/109/1/MG995.html</a>
<b>MG996R</b>	<a href="https://html.alldatasheet.com/html-pdf/1131873/ETC2/MG996R/110/1/MG996R.html">https://html.alldatasheet.com/html-pdf/1131873/ETC2/MG996R/110/1/MG996R.html</a>

Moottorien valinta perustuu arvioon moottorin liikuttelemapainosta. Materiaalien painoksi saatiin 321,14 grammaa, ja voidaan siten arvioida pienien MG-90S moottorien, toimimaan muina paitsi alustaa kääntävänä moottorina, sillä robotin rakenteen takia suurin osa robotin painosta jakautuu sen alustalle. MG-90S moottorien paino voidaan lukea taulukosta 7. Ensimmäisen nivelen tulee pystyä kääntämään sen päälle kasautuvaa kuormaa, joka voidaan laskea olevan materiaalien paino sekä muiden moottorien paino eli 362,20 grammaa. Voidaan arvioida VTS-08A moottorin riittävän liikuttamaan robottia sekä pientä kuormaa.

Kuvassa 4 on numeroitu moottorien sijainnit valmiissa robotissa. Moottoriksi 1 valittiin VTS-08A ja moottoreiksi 2, 3 ja 4 valittiin MG-90S [22]. Toimitusvaikeuksien takia VTS-08A moottoreita ei ollut saatavilla, joten VTS-08A korvattiin kilpailevalla King Maxin KM4806A [23] moottorilla, joka on ominaisuuksiltaan identtinen VTS-08A moottoriin.

### **3.5. Kustannukset**

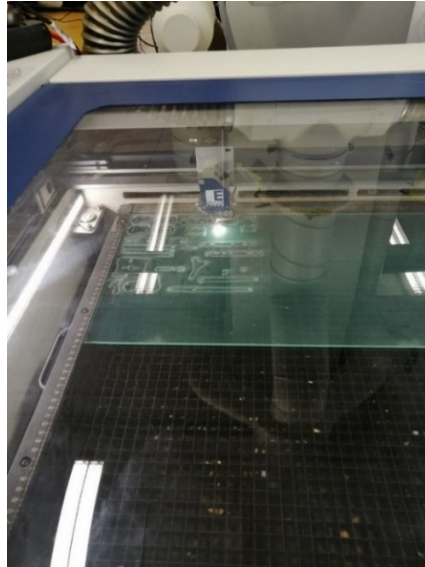
Kokonaiskustannukset työssä voidaan määrittellä syntyvän täysin osien ja materiaalien kustannuksista.

Kaksi akryylilevyä tarvittiin, sillä alkuperäiseen robotin malliin piti tehdä pieniä muutoksia (pohjalevyn suurennus tasapainon ylläpitämiseksi), jotka huomattiin vasta ensitesteissä ja rakennusprosessissa. Lisäksi joillekin osille tarvittiin varaosia hajonneiden tilalle.

Moottorien yhteiskustannukset olivat 27,50 €. Materiaalien kustannuksiksi saatiin yhteensä 45,30 €, joka syntyi kahdesta erinäisestä kirkkaasta akryylilevystä yhteensä 18,00 €, kiinnikkeiden (ruuvit ja mutterit) yhteensä 27,30 €. Kustannuksiksi hankinnoille saatiin 72,80 €, lisäksi toimituskulut 9,80 €, jolloin valmistuksen kokonaiskustannukseksi saadaan 82,60 € (sis. verot).

### **3.6. Rakentaminen**

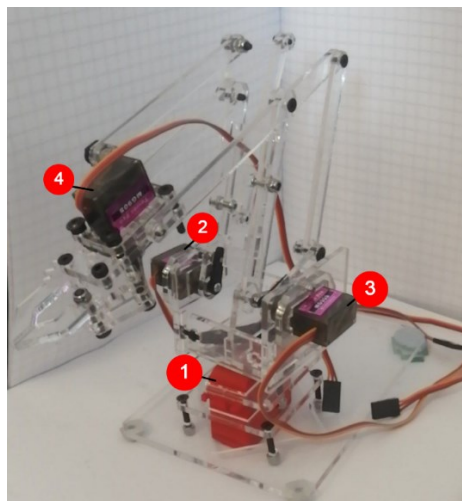
Rakennusprosessi aloitetaan osien leikkaamisella. Valitun MeArm-mallin levymäinen design mahdollisti laserleikkauksen, joten työssä valittiin materiaalintyöstöön laserleikkaus. Valittu materiaali työstettiin käyttämällä Epilog Laser Fusion 120 W laserleikkuria Tampereen Yliopiston FabLab tiloissa. Kuvassa 3 nähdään laser leikkausprosessi käynnissä.



**Kuva 3.** Kuva laserleikkaus prosessista

Joitain osia leikattiin uudelleen ja muokattiin rakennusprosessin aikana, kuten robotin alusta sekä rakennuksessa vaurioituneet osat. Robotin alkuperäiseen malliin, alustan moottoriksi oli mitoitettu pieni MG-90S-kokoinen moottori. Alustaa täytyi muokata, jotta suurempi KM4806A moottori mahtuisi kiinnittymään alustaan.

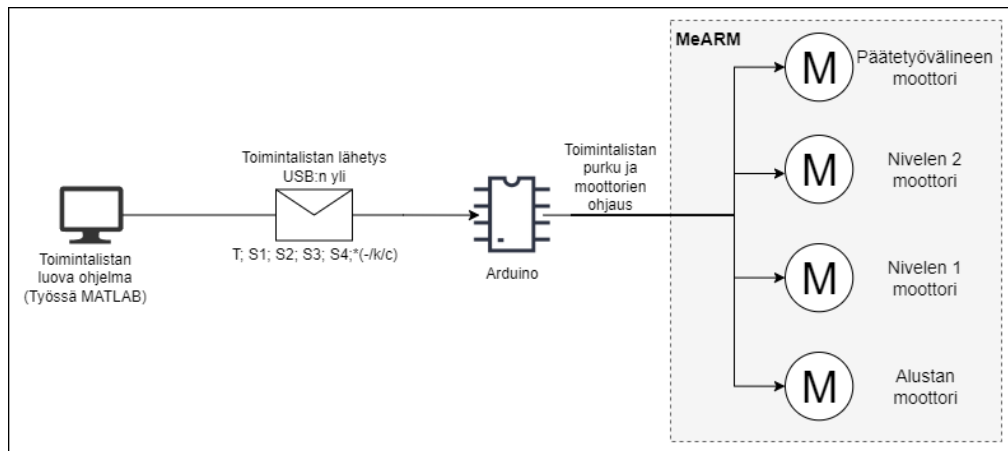
Osat kasattiin ja moottorit yhdistettiin rakennetun robotin runkoon ja kytkettiin lopulta Arduinoon. Kuvassa 4 on esitetty kasattu MeArm robotti sekä moottorien sijainnit. Mekaniikkaa voitiin testata käyttämällä yksinkertaista ohjelmaa.



**Kuva 4.** Moottorien sijainnit

Testiohjelmalla voitiin varmistaa moottorien liikuttavan niveliä oikein ja mekaniikan toimivan, kuten suunniteltu. Työssä rakennettua robottikäsiä ohjaa tietokoneelle suunnit-

teltu MATLAB-ohjelma, joka lähettää toimintalistan Arduinolle. Arduinon vastuualueeseen kuuluu servomoottorien ohjaus. Kuvassa 5 on kaaviokuva järjestelmän toiminnasta.

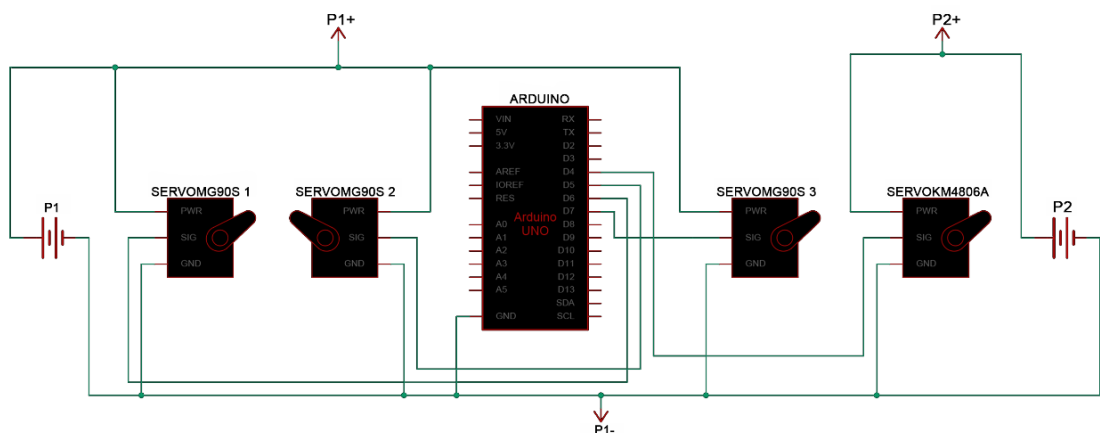


**Kuva 5.** Kaaviokuva järjestelmästä

Käyttöliittymän tarkoituksena on helpottaa ohjelman käyttämistä sekä tarjota korkeamman tason algoritmitoiminnasta esimerkki. Arduino ei ole MATLAB riippuvainen, jolloin sille voidaan lähettää minkä tahansa muun ohjelman kautta, USB-kommunikaatio väylän yli ohjauslista. Ohjauslistan tulee olla oikeassa viestimuodossa. Lähdekoodit voidaan löytää projektin GitHubista [24].

### 3.7. Elektroniikka

Järjestelmän kytkentä on nähtävissä kuvassa 6, jossa moottorit on liitetty sarjaan siten, että ne saavat virtansa ulkoisesta virtalähteestä, mutta ohjaussignaali tulee Arduinolta.

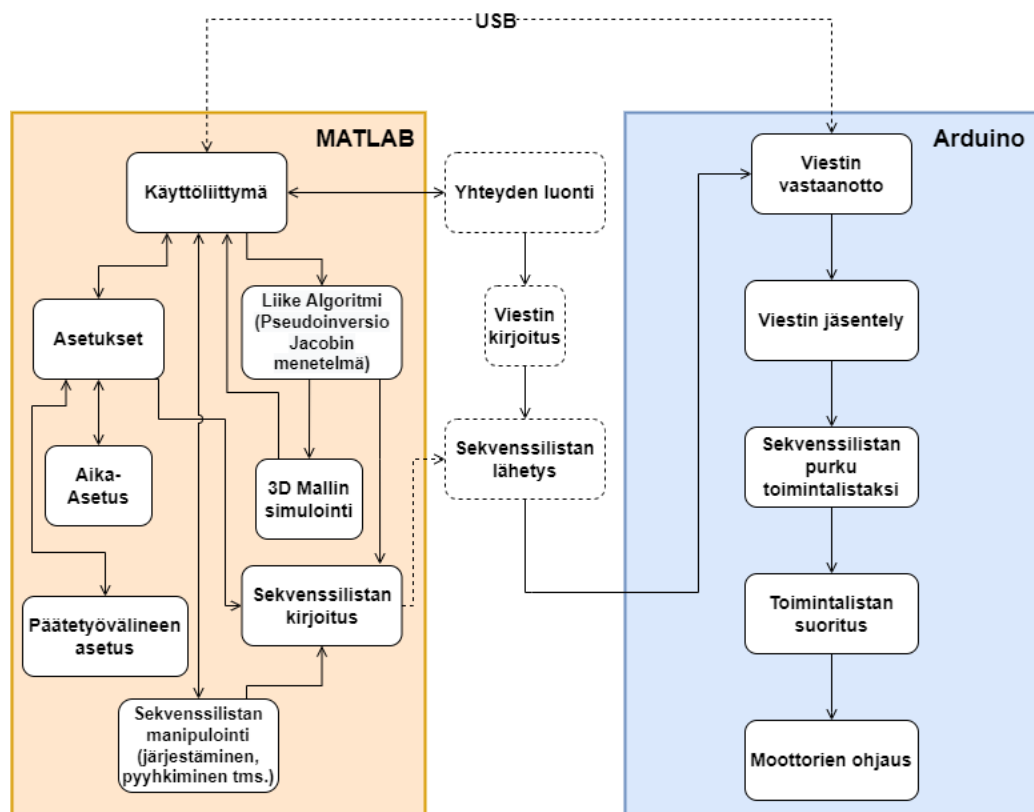


**Kuva 6.** Arduinon kytkentäkaavio

KM4806A on suurempi ja näin ollen raskaampi moottori verrattuna järjestelmässä käytettyyn toiseen pienempään MG90S moottoriin. KM4806A moottorille on jouduttu tekemään oma tehölähteellinen virtapiiri. Projektissa käytettiin yhtä modulaarista virtalähdettä, jolla on kaksi säädeltävää lähtöä. Kuvassa 6 nämä lähdöt voidaan nähdä P1 ja P2.

### 3.8. Ohjelmointi

Ohjelmallisesti järjestelmä on jaettu kahteen osaan: ohjaukseyttöliittymään (MATLAB) ja moottorien ohjaimen (Arduino). Järjestelmän ohjelmallinen jako voidaan nähdä kuvassa 7. MATLAB toimii ohjaukseyttöliittymänä, joka vastaa käyttöliittymän lisäksi kinematiikan laskemisesta ja oikeiden moottorien kulmien tallentamisesta kronologiseen järjestykseen ennen lähettämistä. Arduino on vastuussa USB-väylää pitkin saapuvien viestien prosessoinnista ja moottorien liikuttamisesta.



Kuva 7. Järjestelmän lohkokaavio

Arduinon ja MATLAB:n välinen kommunikaatio tapahtuu 9600 baudinopeudella, tämä on määritetty Arduino -ohjelmassa.

### 3.8.1. Moottorienohjain: Arduino

Arduinon ohjelmaa voidaan kutsua nimellä moottorienohjain, sillä Arduinon ohjelmisto vastaa moottorien ohjaamisesta toimintalistan perusteella. Moottorienohjain rajapinnasta on tehty mahdollisimman yleinen ja helppokäyttöinen, jotta ohjelmistoa voidaan käyttää monissa eri ympäristöissä mahdollisimman helposti.

Arduino ohjaa moottoreita saamansa sekvenssilistan perusteella. Sekvenssilista saapuu ulkoisesta lähteestä ja se sisältää tiedot aikaleimoista ja nivelten kulmista. Sen sisältämät aikaleimat ovat aikoja, jolloin moottorien tulee saavuttaa viestissä annetut kulmat. Arduino ei ota kantaa sekvenssilistan järjestykseen, vaan olettaa ulkoiselta algoritmilta, tässä työssä MATLAB:lta, saapuvan listan olevan jo kronologisessa järjestyksessä. Moottorienohjain käy sekvenssilistan läpi ja jäsenetelee sen omaksi toimintalistaksi. Tämän jälkeen se käy toimintalistan läpi riveittäin ja lukee rivin jokaisen elementin ja liikuttaa nivelet oikeisiin kulmiin. Kun robotti on käynyt toimintalistan läpi, palaa se alkuasentoon ja jää odottamaan seuraavaa sekvenssilistaa tai komentoa ulkoisesta lähteestä.

Arduinon ja MATLAB:n käyttämä kommunikaatio protokolla on luotu tätä työtä varten. Moottorienohjain ottaa vastaan USB-väylää pitkin komentoja ja ohjaa servomoottoreita sen perusteella. Kommunikaatio protokollan komentojen tulee olla muodossa:

aikaleima; Servo 1 kulma; Servo 2 kulma; Servo 3 kulma; Servo 4 kulma;\*(-/k/c)

Viestin aikaleima tulee olla muodossa "hh:mm:ss" ja kulmat tulee olla esitetty asteina. Viestin lopun määrittää "\*" -merkki. Viestin lopussa oleva "k" tai "c" on vaihtoehtoinen ulkoisesti lähetetty komento. Komento "k" on merkki moottorienohjaimelle viimeisestä sekvenssilistan rivistä, jonka jälkeen moottorienohjain aloittaa toimintansa. Komento "c" on käsky pyyhkiä ja alustaa sisäinen tapahtumalista. Jos viestin lopussa ei ole mitään merkkiä, ei moottorienohjain tee mitään, vaan jää odottamaan seuraavaa riviä.

Saapunut sekvenssilista jäsenellään tietorakenteeseen niiden saavuttua moottorienohjaimelle. Käytetyksi tietorakenteeksi valittiin taulukko (engl. *array*), sillä Arduino ei pysty käsittelemään dynaamisia tietorakenteita. Taulukon ennalta määritetty koko on 100, tämä tarkoittaa sitä, että tapahtumien maksimimäärä on taulukon koko. Tapahtuman aikaleima määrittää milloin moottoreiden (alusta, nivel 1, nivel 2 ja koura) tulee olla viestin määrittämässä kulmissa. USB-väylän yli tulleiden viestien perusteella Arduino luo itselleen tapahtuman ("Event") jäsentämällä viestistä olennaiset tiedot ja tallentamalla ne toimintalistaan ("EventArray").

Robotin nivelien liikuttamiselle rakennettiin moottoreita tasaisesti liikuttava funktio, jolla voitiin estää nopeat ryntäävät liikkeet ja tasainen tahti toimintalistan alkioden välillä. Arduino ei tarjoa rajapintaa ohjata moottorien nopeuksia suoraan, eikä yhteisön luoma "ServoVarSpeed"-kirjasto ole yhteensopiva MKR1000-sarjan "SAMD21 Cortex-M0+ 32bit low power ARM MCU"-mikroprosessorin kanssa. Moottorien nopeuksien säätelyyn on rakennettu oma ratkaisu.

Toimintalistan aikaleimojen perusteella voidaan määrittää moottoreiden siirtymäaika. Siirtymäajalla tarkoitetaan aikaa, joka moottoreilla on siirtyä kulmasta seuraavaan. Seuraavan ja nykyisen tapahtuman erotuksena oleva aika (mikäli kyseessä ei ole listan viimeinen alkio) määrittää servojen tälle liikkeelle varatun ajan  $t_{väli}$ .

$$t_{seuraava} - t_{nykyinen} = t_{väli}, \quad (9)$$

jotta  $t_{väli}$  on positiivinen, tulee nykyisen tapahtuman olla kronologisesti ennen seuraavaa tapahtumaa. Ajan perusteella voidaan laskea moottorien nopeuden, jolla moottoreita tulee liikuttaa, jotta moottorit ehtivät niille asetettuun kulmaan. Moottorien nopeuden säätelyyn käytetään "millis"-ajastusfunktiota ja "map"-funktiota. "Millis"-ajastusfunktio kertoo kutsuhetkellä Arduinon toiminta-ajan. "Map"-funktio kääntää luvun alueelta toiselle. "Map"-funktion avulla skaalataan arvoja valituilla alueilla samassa suhteessa, mutta omien alueidensa mittakaavassa [25]. Tämän käyttöä esitetään ohjelmassa 1.

```

    unsigned long MOVING_TIME = t_next - t;
2   unsigned long progress = millis() - startMillis;

4   long SBaseangle_now = ServoBase.read();
    long SBaseangle_next = EventArray[next].SBaseAngle;
6
8   if (progress <= MOVING_TIME) {
        long SBase_angle = map(progress, 0, MOVING_TIME,
                               SBaseangle_now, SBaseangle_next);
10  } move_Servos(SBase_angle);

```

### **Ohjelma 1. Servo moottorien nopeuden säätely Arduino ohjelmassa**

"MOVING\_TIME" muuttuja sisältää tiedon siirtymäajasta. Kun ajallisesti lähestytään listan seuraavaa tapahtuma-aikaa, eli siirtymäaika on loppumassa, "map"-funktion edistys tallennetaan muuttujaan "progress". "Progress"-muuttujan perusteella Arduino laskee tarvittavan sen hetken kulman ja ohjaa moottorin oikeaan sen hetkiseen asentoon. Tällä periaatteella robotti pystyy liikuttamaan niveliä ajan kanssa samassa suhteessa tasaisesti, eivätkä robotin nivelet ryntää uuteen asentoonsa.



### 3.8.2. Ohjauskäyttöliittymä: MATLAB

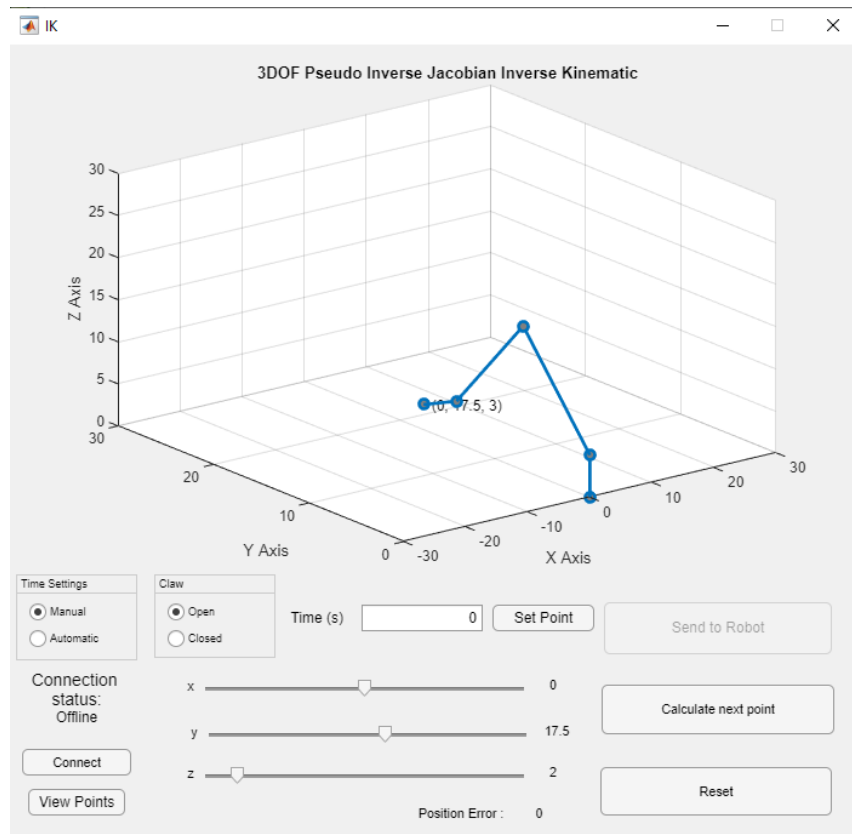
MATLAB-koodin pohjalla on Agustian, Indra et al. [14] tekemä MATLAB-koodi, johon on kirjoitettu pseudoinversio Jacobin (ts. käänteiseen Moore–Penrose) menetelmään perustuva ratkaisu MeArm robotille. MATLAB-koodissa pseudoinversio Jacobin menetelmää on suojattu singulariteeteilta rajoittamalla nivelten kulmia. Indra et al. [14] tekemä MATLAB-koodi on vastuussa kulmien laskemisesta annettujen X-, Y- ja Z-koordinaattien perusteella. Ennalta tehdyn MATLAB-koodin päälle on laajennettu projektin käyttötarkoitukseen omia funktioita. MATLAB koodissa käytetty Denavit–Hartenberg matriisin parametrit tulee asettaa oikein, tällä voidaan varmistaa, että robotin parametrit vastaavat MATLAB koodin parametrejä. Denavit–Hartenberg matriisista asetettiin vastaamaan kaavan 6 matriisia.

$$DH = \begin{bmatrix} j1 & j2 & j3 & j4 \\ 5 & 0 & 0 & 0 \\ 0 & 14 & 14 & 4 \\ 90 & 0 & 0 & 0 \end{bmatrix}, \quad (10)$$

jossa sarakkeet vastaavat asetettujen koordinaattikehyksien arvoja ja jokainen rivi vastaa nivelten DH-parametrejä  $\theta$ ,  $d$ ,  $a$ ,  $\alpha$ . Ensimmäinen rivi vastaa nivelten selvitettäviä kulmia  $R_z(\theta)$ , toinen rivi nivelten siirtymää z-suunnassa (engl. *Offset*)  $T_z(d)$ , kolmas rivi liitosten etäisyyksiä  $T_x(a)$  ja neljäs rivi nivelten kiertymäkulmia  $R_x(\alpha)$ . Työssä koordinaattikehykset on asetettu jokaisen nivelen kohdalle, jolloin liitosten etäisyydet sekä siirtymät voitiin mitata robotista.

Mukautettu käyttöliittymä on laajennettu alkuperäisestä, mutta laskennallisesti taustalla toimiva pseudoinversio Jacobin menetelmään perustuva käänteiskinematikan ratkaisu tapa on muokkaamaton. Lisätyt ominaisuudet sisältävät mm. toimintalistan kaikki toiminnot (luominen, avaaminen, lukeminen, järjestely, lähetys yms.), ajallisen määrittämisen joko manuaalisesti tai automaattisesti sekä toiminnot Arduinon väliseen kommunikointiin. Päivitetty käyttöliittymä voidaan nähdä kuvassa 8.

Ajanasetusperuste voidaan valita joko automaattiseksi tai manuaaliseksi. Ajanasetuksen avulla voidaan määrittää joko manuaalisesti tietty aika, jolloin robotin täytyy olla tietyssä asennossa tai automaattisesti, jolloin aika asettuu MATLAB:n laskennallisen ajan mukaan. Käyttöliittymän avulla voidaan antaa koordinaatiston perusteella robotin ympäriltä X-, Y- ja Z-koordinaatit johon päätetyväline voidaan ohjata painamalla "Calculate" tai "Start Timing" (riippuen aika-asetuksesta). "Calculate" -toiminto on käytössä, mikäli ajanasetusperusteeksi on asetettu "manuaalinen". "Start Timing" -toiminto on käytössä, mikäli ajanasetusperuste on "automaattinen".



**Kuva 8.** Projektin päivitetty käyttöliittymä

Käyttöliittymästä voidaan valita missä asennossa päätetyöväline on auki ja missä asennoissa päätetyöväline on kiinni. Kun asento on saavutettu, päätetyövälineen tila ja aika on asetettu halutuiksi, täytyy piste lisätä toimintalistaan painamalla "Set Point"-nappia. Manuaalisella pisteen lisäyksellä voidaan varmistaa, että vain halutut asennot saadaan toimintalistalle. Toimintalista voidaan avata painamalla "View Point"-nappia, jonka jälkeen toimintalista voidaan päivittää ("Refresh"), pyyhkiä ("Clear") tai järjestellä ("Sort"). Järjestely toimintalistalla tapahtuu aikaperusteisesti nousevassa järjestyksessä, siten että ensimmäisenä Arduinolle lähetetään kronologisesti ensimmäinen toiminta.

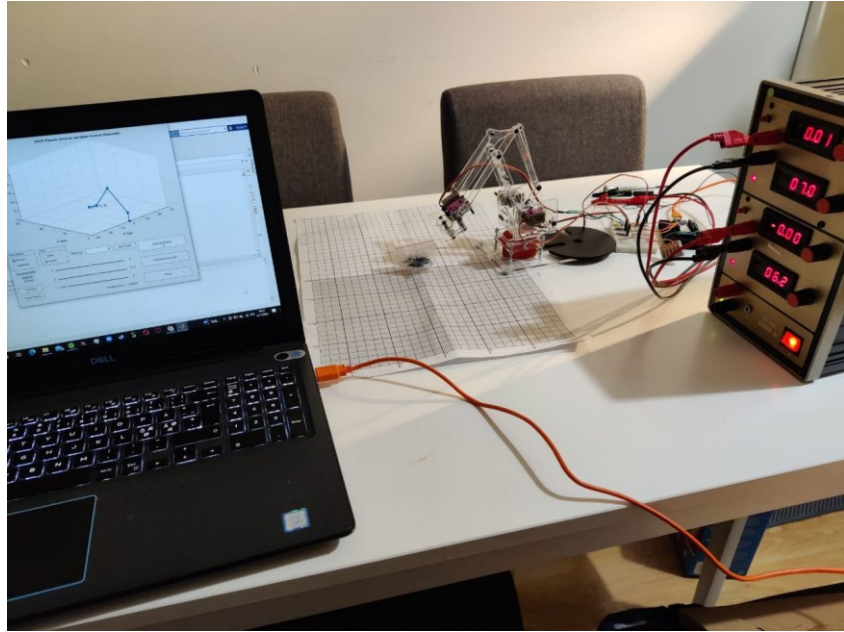
Pyyhkimis-komento ("Clear") lähettää sarjaväylää pitkin Arduinolle käskyn "c" eli käskyn pyyhkiä ja alustaa toimintalistan. Kun lista on oikein, voidaan se lähettää moottorien ohjaimelle (mikäli yhteys sovelluksen ja Arduinon välillä on päällä ("Connection status: Online")) painamalla "Send to Robot"-nappia. "Send to Robot"-napin funktio kirjoittaa automaattisesti toimintalistan loppuun käskyn "k", jolloin Arduino aloittaa moottorien liikuttamisen. Kun toimintalista on lähetetty onnistuneesti, ilmoittaa käyttöliittymä siitä kirjoittamalla "Connection status" -kohtaan "Data sent successfully" eli tiedot lähetetty onnistuneesti.

## 4. ROBOTTIKÄSIVARREN TOIMINNALLISUUDEN TESTAUS

Robottikäsi­varren toiminnallisuutta testattiin yksinkertaisella prosessilla, jossa robotti siirtää kappaleen sijaintia oman koordinaatistonsa sisällä, ensin nostaen kappaleen maasta, siirtäen sen toiseen paikkaan, laskemalla sen alas ja päästämällä irti kappaleesta. Tässä luvussa esitetään valmiin robottikäsi­varren toiminnallisuuden testausta yksinkertaisella kappaleensiirto testillä, jossa robotti siirtää kappaleen alueensa sisällä, esitetään testin tuloksia sekä pohditaan robottikäsi­varren toimintaa eri käyttöympäristöissä.

Testiä varten tulostettiin A3-kokoinen testialusta, johon on tehty 1 cm x 1 cm ruudukko. Ruudukko vastaa MATLAB-käyttöliittymän XY-koordinaatistoa. Koordinaatistojen koordinaattien suhde on 1:1 eli yksi MATLAB yksikkö vastaa 1 cm siirtymää todellisessa maailmassa. Siirrettävänä kappaleena toimi salpapus­si, joka sisälsi Arduino koontisarjan käyttämättömiä johtoja. Siirrettäväksi kappaleeksi voitaisiin valita mikä tahansa pehmeä ja kevyt kappale. Robottikäsi­varren leukojen on helppo tarttua salpapus­si­ista sekä pussin sisältö on kevyt. Kuvassa 9 voidaan nähdä testi­ympäristö toiminnassa.

Järjestelmän virtalähteenä on käytetty Sahitec Oy:n TL-230D laboratorio virtalähdettä, jonka avulla voidaan kahta ulostulojännitettä säädellä. Kuvassa 9 mustat pyöreät levyt ovat pohjan tasapainoittavia painoja. Kuvassa 9 näkyvän elektroniikka laudan oikea puoli ottaa vastaan 7 V ja vasen puoli 6 V. Koko kytkentäkaavio voidaan nähdä kuvassa 6. Arduino ja MATLAB ovat yhteydessä ("Connection Status: Online"). Arduinon oranssi johto on yhdistettynä tietokoneeseen, jonka kautta taustalla toimiva MATLAB kommunikoi Arduinon kanssa. MATLAB:n kautta on mahdollista ohjata robottia. Kaavio järjestelmästä voidaan nähdä kuvassa 5.



**Kuva 9. Testiympäristö**

Robottikäsivarsi pystyy siirtämään pussia koordinaatiston sisällä onnistuneesti. Alkutilanteessa salpapussin tulee olla pystyssä, jotta kappaleesta saadaan leuoilla varmasti kiinni. Testien alkutilanteessa robottikäsivarsi on sen alustetussa tilassa, johon se myös palaa testien lopussa automaattisesti. Robottikäsivarren tulee siirtyä siirrettävää salpapussia kohti, salpapussin yläpuolelta, jotta käsivarsi ei kaada pussia ennen kuin siitä saadaan kiinni. Testissä leuoilla siirrytään salpapussin sijaintia kohti sen yläpuolelta, otetaan pussista kiinni, nostetaan se ylös, siirretään haluttuun paikkaan, päästetään irti ja palataan alkuperäiseen asentoon. Testeissä huomattiin, että välillä on tilanteita, jolloin pussi on asetettu epätarkasti sen aloitustilaan, joka johti siihen, etteivät leuat saa pussista kiinni. Lopputilanteessa pussia ei saada myöskään aina asetettua takaisin pystyyn, kun se asetetaan takaisin alas ja päästetään irti. Lisäksi testeissä huomattiin, että robottikäsivarren liikkeen tasaisuutta voidaan parantaa asettamalla toimintalistaan mahdollisimman monta ”väliasentoa” eri asentojen välille. Väliasennolla tarkoitetaan asentoja toimintalistassa, jotka ovat haluttujen asentojen välillä. Robottikäsivarrella pystytään siirtämään kappaleita koordinaatistonsa sisällä.

Rakennettu robottikäsivarsi on suhteellisen pieni, kevyt sekä se pystyy toteuttamaan pihdeillään esimerkiksi kappaleen siirtoprosesseja ja näitä robottikäsivarren ominaisuuksia voitaisiin käyttää esimerkiksi siirtämään pieniä pehmeitä kappaleita tai pusseja linjasta toiseen tai esimerkiksi laatikkoon. Robottikäsivarren päätetyövälinettä muuttamalla voitaisiin robottikäsivartta hyödyntää muissakin käyttöympäristöissä, esimerkiksi mikäli päätetyöväline vaihdettaisiin kauhamalliseen päätetyövälineeseen, voitaisiin sillä siirtää esimerkiksi suuri määrä pieniä kappaleita (esim. muttereita) paikasta toiseen.

## 5. YHTEENVETO

Materiaali työstämällä saatiin robottikäsivarren osat, osat kiinnitettiin, robotti rakennettiin, moottorit kiinnitettiin, MATLAB käyttöliittymä ja Arduino ohjelma saatiin kommunikoidaan sekä ohjaamaan robottikäsivarren moottoreita. Lopputuloksena saatiin valmis ja ohjattava robottikäsi. Tässä luvussa esitetään yhteenveto projektista, valmiin robottikäsivarsi projektin lopputulos sekä esitetään mahdollisia jatkokehityskohteita.

Lopputuloksena on n. 7x20x10 cm kokoinen kirkkaasta akryylistä leikattu ja rakennettu robotti, jota voidaan ohjata esim. MATLAB käyttöliittymän kautta. Ohjaus voidaan myös toteuttaa käyttämällä mitä tahansa muuta ulkoista sovellusta. Sovelluksen täytyy pystyä tekemään reittisuunnittelu ja lähettämään toimintalista moottorienohjaimelle sen tarvitsemassa muodossa. Robottikäsivartta voidaan ohjata esimerkiksi suoraan Arduinon oman sarjamonitorin kautta, lähettämällä sille manuaalisesti yksittäisiä komentoja.

Robottikäsivarren rakennus prosessi itsessään ei ole suuri prosessi, mikäli projektin suunnitelmat ovat selvät ja projektin toteuttajalla on jonkinlaista kokemusta robottien toiminnasta tai niiden rakentamisesta. Työn alkuvaiheessa kokemuksen puute johti siihen, että työn suorittamisen aikana huomattiin erinäisiä pieniä ongelmia, jotka veivät enemmän aikaa kuin oli suunniteltu. Nämä ongelmat syntyivät asioista, joita ei osattu ottaa huomioon suunnittelu vaiheessa, kuten esimerkiksi robottikäsivarren riittävien kiinnityskohtien varmistaminen, niiden vahvistaminen alkuperäistä 3D-mallia muuttamalla, laserleikkauksen skaalausasetusten muutokset, komponenttien mahdolliset toimitusvaikeudet, moottorien johtojen läpivientien suunnittelu, tarvittavien työkalujen hankinta (rautasaha ja oikean kokoiset jakoavaimet) ja huomattavimpana robotin pohjan tasapainottaminen. Materiaalien työstö veivät jonkin verran ylimääräistä aikaa, sillä laserleikkuri ei itsesään ollut tuttu ennen työn aloittamista. Suurin osa robottikäsivarren ongelmista ja puutteista selvitettiin ja korjattiin työn valmistuksen aikana, mutta yhden ratkaisun jälkeen esiin nousi lisää tai ratkaisu aiheutti muita ongelmia, joiden korjaus vei lisää suunnittelematonta aikaa. Työ pyrittiin pitämään kohtuullisissa rajoissa, jonka takia esimerkiksi pohjan tasapainotusongelma, jätettiin tällä kertaa ratkaisematta, ja siihen tehtiin tilapäinenratkaisu.

Robottikäsivarsi pystyy suorittamaan onnistuneesti kappaleensiirtoprosesseja. Mikäli robottia kuitenkin haluttaisiin käyttää esimerkiksi teollisissa ympäristöissä tai jatkuvassa toiminnassa, tulisi sen rakennusmateriaaliksi valita jokin muu kuin akryyli, esimerkiksi

metalli. Moottorit on mitoitettu liikuttelemaan akryylistä valmistettua robottia ja tulisi materiaalinvaihdon yhteydessä valita ja mitoitaa myös eri moottorit. Tässä työssä rakennettu robotti on prototyyppi eivätkä sen osat ole suunniteltu kestävään jatkuvaa toimintaa. Robotin ominaisuuksia voitaisiin jatkokehittää vieläkin entisestään. Robotin moottorien tarkalla kalibroinnilla voitaisiin parantaa robotin tarkkuutta. Osien 3D-mallia kehittämällä, esimerkiksi MeArm-robotin avoimenlähdekoodin 3D-mallia muuttamalla robotin rakenteesta olisi mahdollista saada paljon tukevampi esimerkiksi lisäämällä sen alustan kokoa siten, ettei se tarvitsisi vastapainoja tai vaihtoehtoisesti kiinnittämällä se pohjasta paikoilleen. Lisäksi itse johdotuksille, virtalähteelle ja Arduinolle olisi mahdollista rakentaa sijainti itse robottiin. Pseudoinversio Jacobin ratkaisutapa on nopea ja tehokas tapa laskea robotille kulmat, mutta tätäkin olisi mahdollista vielä parantaa, sillä tämänhetkinen ratkaisumalli tekee vain arvion asennon tarvitsemista kulmista, mutta ei esimerkiksi ota kantaa siihen ylettääkö robotti tiettyyn pisteeseen vai ei. Asentamalla robottiin eri sensoreita, kuten pienen painesensorin päätetyövälineeseen, voitaisiin esimerkiksi varmistua siitä, että kappaleesta saadaan varmasti kiinni. Harrastusbudjetilla rakennettu robotti on silti ohjattavissa sekä se kykenee suorittamaan pieniä kappaleen siirtoprosesseja.

# LÄHTEET

- [1] M. Minsky, “*Will Robots Inherit the Earth?*”, Scientific American, vol. 271, no. 4. Päivitetty 1994. Saatavissa (viitattu 26.9.2021):  
<http://www.jstor.org/stable/24942877>
- [2] Oxford Economics, “*How robots change the world*”, 2020. Saatavilla (Viitattu 21.9.2022): [https://resources.oxfordeconomics.com/how-robots-change-the-world#:~:text=New%20industrial%20robot%20installations%20\(thousands,as%20over%20the%20eight%20previous](https://resources.oxfordeconomics.com/how-robots-change-the-world#:~:text=New%20industrial%20robot%20installations%20(thousands,as%20over%20the%20eight%20previous)
- [3] D. Rus. “*A Decade of Transformation in Robotics*”, OpenMind BBVA, Technology, Robotics, 2018. Saatavilla (Viitattu 21.9.2022):  
<https://www.bbvaopenmind.com/en/articles/a-decade-of-transformation-in-robotics/>
- [4] O. Vainio. “*Verkkokauppajätti Amazon kehittää jakelurobottinsa Helsingissä*”, Iltalehti, Digi & Tekniikka, Digi uutiset, 1.7.2021. Saatavilla (viitattu 21.9.2022): <https://www.iltalehti.fi/digi uutiset/a/0bcc7089-0aac-4f7a-89b4-f47938cf994c>
- [5] HUS. “*Uusi Da Vinci Xi -leikkausrobotti vahvistaa HUSin roolia robottikirurgian kehittäjänä Euroopassa*”, mediatiedote, 10.11.2021. Saatavilla (viitattu 21.9.2022): <https://www.hus.fi/ajankohtaista/uusi-da-vinci-xi-leikkausrobotti-vahvistaa-husin-roolia-robottikirurgian-kehittajana>
- [6] G. McComb. “*Robot Builder's Bonanza*”, Fourth Edition (McGraw-Hill Education: New York, Chicago, San Francisco, Athens, London, Madrid, Mexico City, Milan, New Delhi, Singapore, Sydney, Toronto, 2011, 2001, 1987). <https://www-accessengineeringlibrary-com.libproxy.tuni.fi/content/book/9780071750363>
- [7] ISO/TC 299, the International Organization of Standardization, ISO 8373:2021, Robotics – Vocabulary, 2021. Saatavilla (viitattu 8.6.2022): <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-3:v1:en>
- [8] T. Bajd, M. Munih, M. Mihelj, J. Lenari, A. Stanovnik “*Robotics*”, Springer Netherlands, 2010. *ProQuest Ebook Central*, Saatavilla (viitattu 8.6.2022): <http://ebookcentral.proquest.com/lib/tampere/detail.action?docID=510462>
- [9] I. Bonev, “*What are Singularities in a six-axis robot arm*”, Mecademic. Päivitetty: 27.7.2019. Saatavissa (viitattu: 10.6.2022): <https://www.mecademic.com/en/what-are-singularities-in-a-six-axis-robot-arm>
- [10] R. P. Paul, “*Robot Manipulators — Mathematics, Programming and Control*”. The MIT Press Series in Artificial Intelligence. The MIT Press, 1981. Saatavissa (viitattu 26.9.2022):  
<https://books.google.fi/books?hl=en&lr=&id=UzZ3LAY-qvRkC&oi=fnd&pg=PP9&dq=Robot+Manipulators:+Mathematics,+Prog->

[ramming,+and+Control&ots=zXtmTEu6aP&sig=EBrAhL-bRS\\_m2q8LYjNz9D3W6THg&redir\\_esc=y#v=onepage&q=homogen&f=false](https://www.researchgate.net/publication/280204921_Robot_Modeling_and_Control20061Mark_W_Spong_Seth_Hutchinson_M_Vidyasagar_Robot_Modeling_and_Control_Wiley_www.wiley.com_December_2005_Hardcover_496_pages_ISBN_0-471-64990-2_32954950_http://www.wiley.com/WileyC)

- [11] M. W. Spong "Robot Modeling and Control", John Wiley & Sons, INC, 2005. *ResearchGate*, Saatavilla (viitattu 22.9.2022): [https://www.researchgate.net/publication/280204921 Robot Modeling and Control20061Mark W Spong Seth Hutchinson M Vidyasagar Robot Modeling and Control Wiley www.wiley.com December 2005 Hardcover 496 pages ISBN 0-471-64990-2 32954950 http://www.wiley.com/WileyC](https://www.researchgate.net/publication/280204921_Robot_Modeling_and_Control20061Mark_W_Spong_Seth_Hutchinson_M_Vidyasagar_Robot_Modeling_and_Control_Wiley_www.wiley.com_December_2005_Hardcover_496_pages_ISBN_0-471-64990-2_32954950_http://www.wiley.com/WileyC)
- [12] P. Corke, "Denavit–Hartenberg notation", QUT Robot Academy. Päivitetty: 10.4.2017. Saatavissa (viitattu 12.10.2022): <https://robotacademy.net.au/lesson/denavit-hartenberg-notation/>
- [13] P. Corke. "Jacobian and Number of Robot Joints", QUT Robot Academy. Päivitetty: 11.4.2017. Saatavissa (viitattu 29.9.2022): <https://robotacademy.net.au/lesson/jacobian-and-number-of-robot-joints/>
- [14] I. Agustian, N. Daratha, R. Faurina, A. Suandi, "Robot Manipulator Control with Inverse Kinematics PD-Pseudoinverse Jacobian and Forward Kinematics Denavit–Hartenberg", *Jurnal Elektronika dan Telekomunikasi*, 2021. *ResearchGate*, Saatavilla (Viitattu 10.6.2021): [https://www.researchgate.net/publication/354251825 Robot Manipulator Control with Inverse Kinematics PD-Pseudoinverse Jacobian and Forward Kinematics Denavit Hartenberg](https://www.researchgate.net/publication/354251825_Robot_Manipulator_Control_with_Inverse_Kinematics_PD-Pseudoinverse_Jacobian_and_Forward_Kinematics_Denavit_Hartenberg)
- [15] Wikipedia, Open-source. Saatavissa (viitattu 26.9.2022): [https://en.wikipedia.org/wiki/Open\\_source](https://en.wikipedia.org/wiki/Open_source)
- [16] B. Gary, MeArm Robotics Labs, "Introducing the MeArm". Päivitetty 14.6.2019. Saatavissa (viitattu 3.6.2022): <https://mearm.com/2019/06/14/welcome-to-my-blog/>
- [17] Arduino, "What is Arduino?" Päivitetty: 5.2.2018. Saatavissa (viitattu 24.10.2021): <https://www.arduino.cc/en/Guide/Introduction>
- [18] John, Circuits Today, "Story and History of Development of Arduino", 26.3.2014. Saatavilla (viitattu 24.10.2021): <https://www.circuitstoday.com/story-and-history-of-development-of-arduino>
- [19] FabLab Tampere, Hinnasto. Saatavissa (viitattu 13.6.2022): <https://sites.tuni.fi/fablabtampere/hinnasto/>
- [20] Tampereen yliopisto, Tutustu meihin, FabLab Tampere. Saatavissa (viitattu 28.9.2022): <https://www.tuni.fi/fi/tutustu-meihin/fablab-tampere>
- [21] Etola Yhtiö, Etra, Tekniset muovit PDF. Saatavissa (viitattu 19.7.2022): [https://www.etra.fi/media/mageworx/downloads/attachment/file/t/e/tekniset\\_muovit\\_luettelo.pdf](https://www.etra.fi/media/mageworx/downloads/attachment/file/t/e/tekniset_muovit_luettelo.pdf)
- [22] Electronicoscaldas, Datasheet, Micro Servo Motor MG90S – Tower Pro. Saatavissa (viitattu 14.6.2022): [https://www.electronicoscaldas.com/datasheet/MG90S Tower-Pro.pdf](https://www.electronicoscaldas.com/datasheet/MG90S_Tower-Pro.pdf)



- [23] Emtech, KM4806A Standard 6Kg Analog Servo. Saatavissa (viitattu 14.6.2022): [http://www.emtechhobby.com/index.php?main\\_page=product\\_info&cPath=11&products\\_id=122](http://www.emtechhobby.com/index.php?main_page=product_info&cPath=11&products_id=122)
- [24] Github, Miska Romppainen Robotarm-projekti, repo. Saatavissa (viitattu 19.7.2022): <https://github.com/Fazsu/Robotarm>
- [25] Arduino, Documentation, Functions, Map. Päivitetty 19.7.2022. Saatavissa (viitattu 19.7.2022): <https://www.arduino.cc/reference/en/language/functions/math/map/>