Høgskulen på Vestlandet

Bacheloroppgave Elektro

ING3055

Predefinert informasjon

Startdato:	30-04-2019 16:23	**Termin:**	2019 VÅR
Sluttdato:	08-05-2019 14:00	**Vurderingsform:**	Norsk 6-trinns skala (A-F)
Eksamensform:	Bacheloroppgave Elektro med muntlig presentasjon/eksaminasjon		
SIS-kode:	203 ING3055 1 BOPPG 2019 VÅR Haugesund		
Intern sensor:	(Anonymisert)		

Deltaker

Kandidatnr.:	101

Informasjon fra deltaker

Tittel *:	Vurdere om en kommersiell robot kan brukes i Industrien		
Engelsk tittel *:	Consider whether a commercial robot can be used in the Industry		
Egenerklæring *:	Ja	**Inneholder besvarelsen konfidensiell materiale?:**	Nei
Jeg bekrefter at jeg har registrert oppgavetittelen på norsk og engelsk i StudentWeb og vet at denne vil stå på vitnemålet mitt *:	Ja		

Gruppe

Gruppenavn:	(Anonymisert)
Gruppenummer:	3
Andre medlemmer i gruppen:	102

Jeg godkjenner avtalen om publisering av bacheloroppgaven min *

Ja

2019

Vurdere om en kommersiell robot kan brukes i industrien

Svein Jarle Dyrskog Morten Håkull

Level Power & Automation

Kandidat nummer:
101 og 102

Innleveringsfrist:
08.05.2019

Høgskulen på Vestlandet

BACHELORPROSJEKT

Studenten(e)s navn: Morten Håkull og Svein Jarle Dyrskog

Linje & studieretning Bachelor i ingeniørfag, Elektro

Oppgavens tittel: *Vurdere om en kommersiell robot kan brukes i industrien*

Oppgavetekst:

Utføre en Site Assessment Test (SAT) på en kommersiell robot for å teste om den kan utføre oppgavene som en industriell robot gjør nå i dag.

Programmere et tenkt arbeidstilfelle der roboten skal åpne en dør, hente en artikkel og plassere inni en «CNC-maskin», lukke døren igjen og starte «CNC-maskinen». Måle hvor fort roboten kan gjøre det tenkte arbeidstilfelle, og vurdere om roboten kan gjør dette på et akseptabelt nivå.

Endelig oppgave gitt: *01.03.2019*

Innleveringsfrist: Onsdag 8.mai 2019 kl. 12.00

Intern veileder: Edmundo Villacorta

Ekstern veileder: *Tommy Vika*

emailadresse ekstern veileder: *tommy.vika@levelgroup.no*

Godkjent av studieansvarlig:
Dato: 25/4-19

Kandidat nummer:
101 og 102

Innleveringsfrist:
08.05.2019

Høgskulen på Vestlandet

Høgskolen på Vestlandet

Campus Haugesund

Bjørnsonsgt. 45

5528 HAUGESUND

Oppgavens tittel		Rapportnummer
Vurdere om en kommersiell robot kan brukes i industrien		*(Fylles ikke ut)*
Utført av		
Svein Jarle Dyrskog og Morten Håkull		
Linje		Studieretning
Elektro *(Stryk det som ikke passer)*		
Gradering	Innlevert dato	Veiledere
Åpen	08.05.19	Edmundo Villacora - Intern Tommy Vika - Ekstern

Ekstrakt

Level Industries har en CNC-maskin der en operatør må manuelt sette inn artikler i maskinen. Det har vært undersøkt om en vanlig kommersiell robot kan automatisere denne prosessen på en effektiv måte.

Roboten har blitt testet om den er like nøyaktig som det fabrikanten sier den er, om den klarer å bære nyttelasten som er oppgitt og om den kan repetere handlinger like presist hver gang.

Forord

Vi er en gruppe på to heltidsstudenter ved Høgskulen på Vestlandet campus Haugesund som studerer elektroingeniør Y-vei. Vi har begge fagbrev som elektriker og ser frem til å få avsluttet studiet.

Ønsker å meddele vår takknemlighet til våre veiledere Edmundo Villacorta fra høyskolen, og Tommy Vika fra Level Power & Automation.

Bjarne Magnussen fra Level Industries som viste oss rundt på verkstedhallen, vi fikk også se en CNC-maskin og virkemåten til denne.

Foreleserne på HVL som var behjelpelige med å svare på alt det vi lurte på.

Til slutt ønsker vi å takke venner, familie og de som har gått gjennom studiet med oss for all lærdom og tålmodighet.

Innholdsfortegnelse

Tabeller og figurer

Forkortelser og begrep

CNC - Computer Numerical Control. En maskin som blir styrt av en datamaskin, kan lage forskjellige deler av forskjellige materialer som for eksempel metall.

PWM - Pulse Width Modulation eller Pulsbreddemodulasjon

Shield - Utvidelseskort til Arduino mikrokontroller

SAT - Site Acceptance Test.

FAT- Factory Acceptance Test.

Degrees of freedom - Antall akser roboten har.

Chuck - Lukkemekanisme i CNC-maskin for å låse artiklene i en bestemt posisjon.

Maskinering - En prosess hvor man ved hjelp av ulike maskiner og løsninger former et råstoff som jern eller stål til en ønsket form og ofte med bestemte egenskaper.

Halvleder - Et materiale som har strømføringsevne mellom metall (som kobber og aluminium) og et isolerende material (som glass).

Elektronhull - Oppstår når et atom mangler ett elektron, og atomet får av dette en positiv ladning.

Raspberry Pi - Er en liten datamaskin, Raspberry Pi har høyde 17 mm, bredde 85 mm og dybde 56 mm.

API - Application Programming Interface, et programmeringsgrensesnitt.

Python – Er et objektorientert programmeringsspråk

Griper – Verktøyhode som monteres på robot, kan gripe rundt gjenstander.

UPS – Uninterruptible Power Supply.

Sammendrag

Level Industries har en CNC-maskin der en operatør må manuelt sette inn artikler i maskinen. Det har vært undersøkt om en vanlig kommersiell robot kan automatisere denne prosessen. Prosessen som roboten skal utføre består av:

-Åpne en skyvedør på en CNC-maskin

-Flytte en artikkel fra en lokasjon til inn i CNC-maskinen

-Lukke skyvedøren til CNC-maskin

-Gi signal til CNC-maskin om at den skal starte prosess

-Motta signal fra CNC-maskin når prosessen er ferdig

-Åpne skyvedør

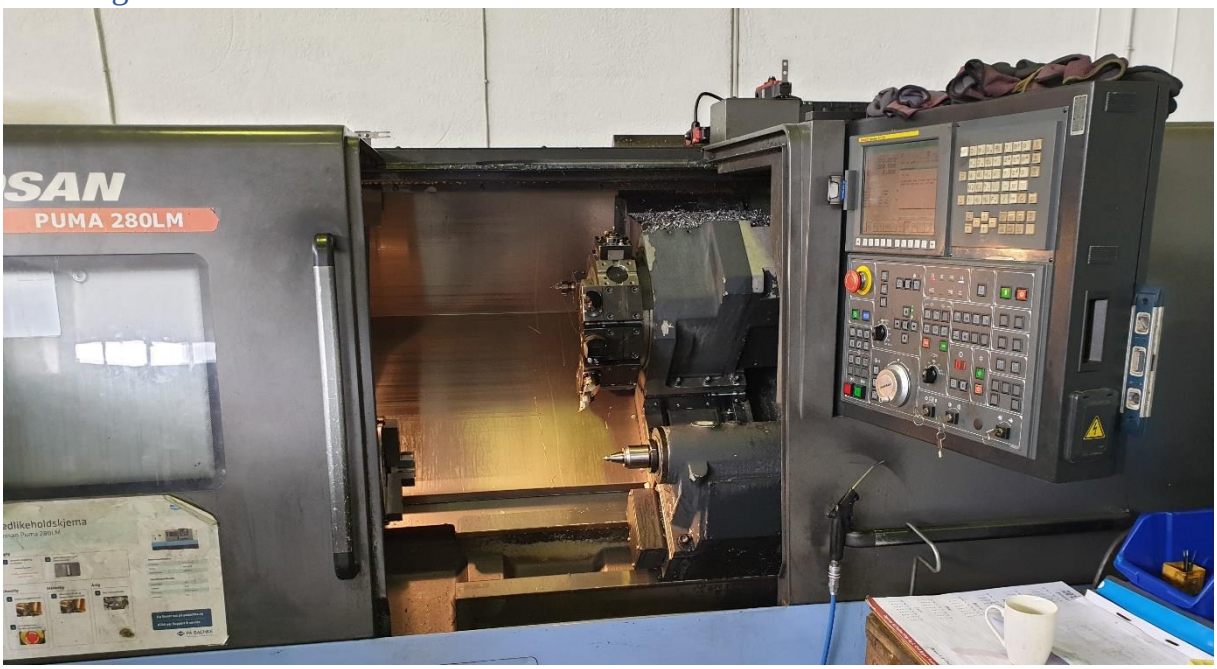
-Hente artikkel fra CNC-maskin og plassere i ferdig lokasjon

Roboten har blitt testet om den er like nøyaktig som det fabrikanten sier den er, om den klarer å bære nyttelasten som er oppgitt og om den kan repetere handlinger like presist hver gang. Det er også laget en modell for å simulere en arbeidsoppgave der roboten skal operere en CNC-maskin.

1 Innledning

Dette kapittelet vil ta for seg hva problemstillingen til oppgaven er og hva denne rapporten fokuserer på.

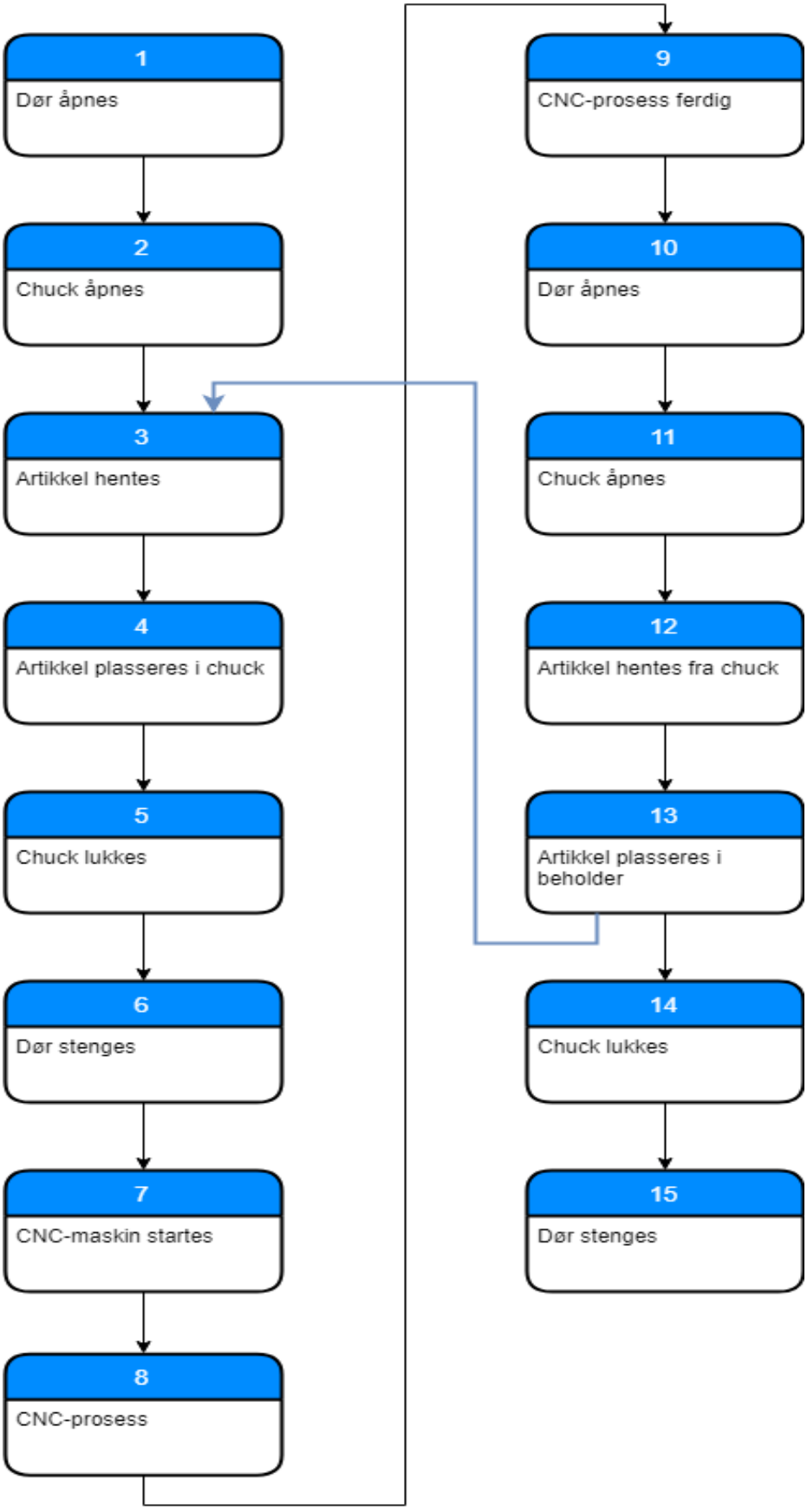
1.1 Bakgrunn



Figur 1 CNC-maskinen som artiklene skal inn i

På Level Industries står det nå i dag en CNC-maskin som det blir manuelt satt inn metall artikler som skal maskineres. Disse artiklene kan ha en maksimal størrelse på 300 mm og vekt på 60 kg. Da det står et menneske og gjør denne jobben nå i dag, er dette en prosess som kan gjøres billigere ved bruk av en robot. En CNC-operatør har en gjennomsnittslønn på 454 200 kr i året. (Utdanningsdirektoratet, 2019)

Om det skal produseres et stort antall artikler med samme mål, kan dette være psykisk krevende prosess for en ansatt som gjerne skal stå med maskinen og gjør samme oppgave i flere timer, hvor selve maskineringsprosessen kan stå og gå noen minutter før artikkelen kan hentes ut. Før en igjen må gjenta prosessen med å sette inn en ny artikkel (se figur 2 side 2). Dette kan påvirke den ansatte med å være psykisk utslitende og fører til dødtid.



Figur 2 Flytskjema for den menneskelige operasjonen

Robotens oppgave vil være å hente artiklene en etter en og plassere de i chucken inni CNC-maskinen, for så å starte CNC-maskinen. Etter CNC-maskinen har vært igjennom maskinering prosessen, skal artiklene igjen hentes ut av maskinen.

Roboten skal kun betjene CNC-maskinen fysisk ved å åpne skyvedøren. Resten skal skje via signaler mellom styringsenhetene til robot og CNC-maskin. Signalene som blir sendt mellom disse styringsenhetene vil sørge for at CNC-maskinens chuck lukker seg rundt artikkel, at CNC maskinen starter og at CNC-maskinen gir beskjed om at prosessen er ferdig.

1.2 Oppgaven og dens formål

Hovedfokuset i denne oppgaven har vært å sjekke om en vanlig kommersiell robot fra Dorna er god nok til å gjøre prosessen med CNC-maskinen automatisk. Da robotarmer har blitt billigere og enklere å få tak i. Rapportens formål er å vurdere om prosessen kan automatiseres uten alt for store kostnader. Det har blitt utført en SAT, og det er kommet frem noen forslag til hvilke eventuelle forbedringer som kan gjøres for å optimalisere driften av roboten best mulig.

Systemgrensene vi har satt er:

- Det skal programmeres i Dorna programvaren som kan lastes ned på hjemmesiden til Dorna.
- Styringsenheten og roboten skal testes i fabrikkstandard. Ingen konfigurasjoner vil bli gjort fysisk med styringsenheten eller roboten.

2 Robotens historie og bruksområder

I dette kapittelet vil det bli tatt for seg litt om historien til robot, og i hvilke tilfeller en robot kan brukes.

2.1 Historie

Joseph Engelberger blir sett på som «robotens far». Han var inspirert av science fiction litteraturen til Isaac Asimov. Han overtalte firmaet han jobbet for med å lage en prototype som het Unimate #001. Dette var den første roboten som ble montert langs et samlebånd i 1959. Dette var hos General Motors i Trenton, New Jersey som drev med formstøping.

Den første industrielle roboten som ble masseprodusert het Unimate 1900 og ble brukt til formstøping. Dette skjedde i 1969. (Robotic Industries Association, 2019)

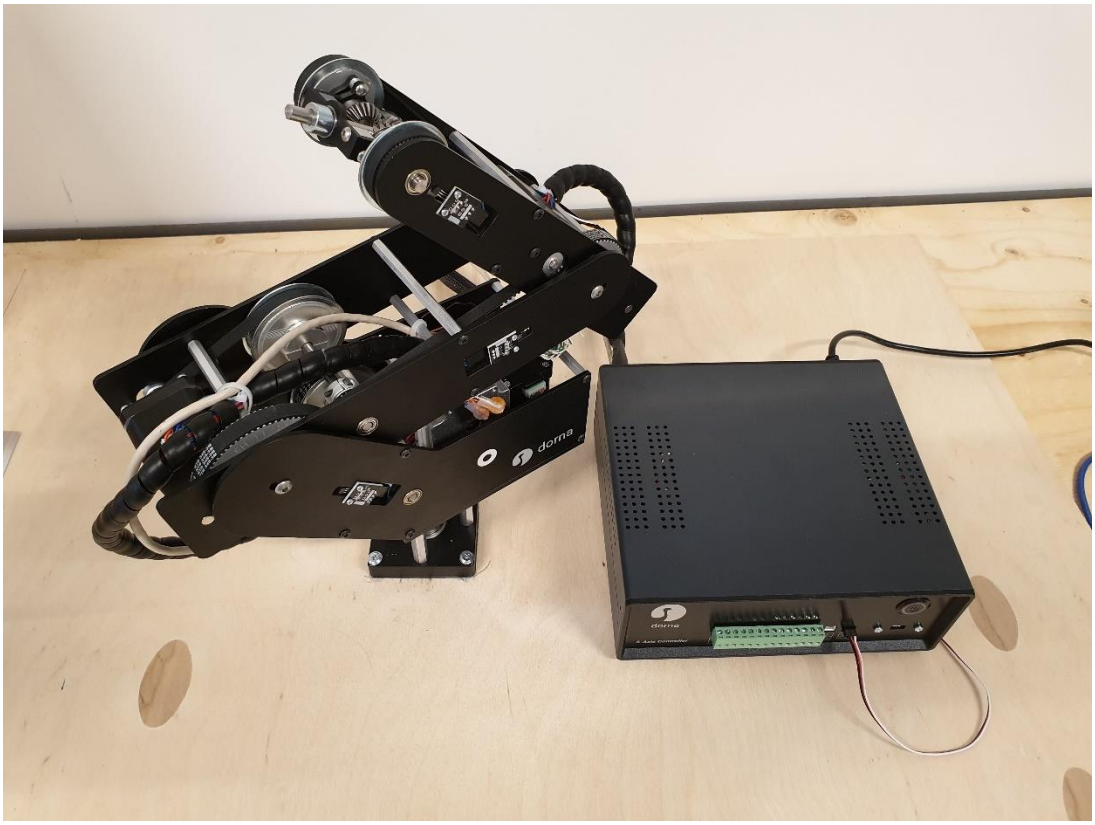
2.2 Bruksområder

Robot er veldig godt egnet til å utføre arbeidsoppgaver som blir sett på som for farlig, dyrt og ineffektivt for mennesker å gjøre selv. En robot kan programmeres til å utføre utfordrende presist arbeid gjentatte ganger som sveising, lakkering, gravering og kan også brukes til å flytte gjenstander, pakker etc. En robot kan gjerne arbeide hele døgnet rundt og dette vil gjøre produksjonen billigere. Det kan argumenteres med at bedrifter her i Norge kan konkurrere mot billig produksjon i fra utlandet, som kan hindre utflagging av bedrifter.

3 Presentasjon av robot

For at en robotarm skal kunne utføre varierte operasjoner, er den bygget opp av flere komponenter. Følgende kapittel vil ta for seg disse komponentene og forklarer litt om disse.

3.1 Spesifikasjoner



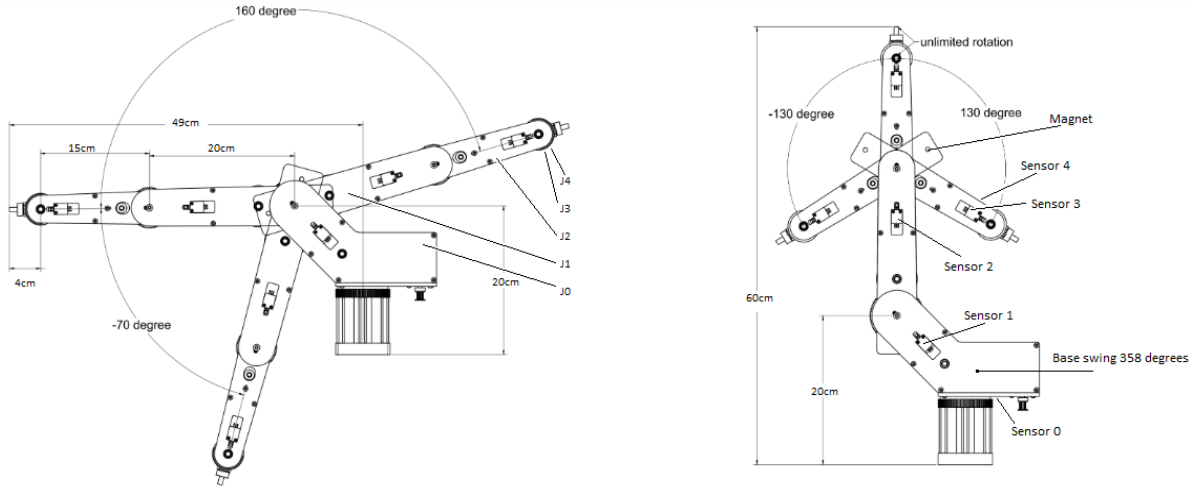
Figur 3 Dorna robot og styringsenhet

Robot	
Number of axis	5
Payload	1,1 Kg
Horizontal reach	497,1 mm
Vertical reach	607,9 mm
Resolution	0,1 mm
Repeatability	0,025 mm
Base swing	358 deg
Lower arm swing	230 deg
Upper arm swing	260 deg
Wrist swing	Unlimited
Max base speed	300 deg/s
Max lower arm speed	150 deg/s
Max upper arm speed	150 deg/s
Max wrist speed	300 deg/s
Max body width	109,6 mm
Body weight	5,4 Kg

Tabell 1 Databladet til robotarmen. Tabell hentet fra Dorna

3.2 Degrees of Freedom

Degrees of Freedom er et begrep som ofte blir brukt til å beskrive bevegelsesfunksjonene til roboter. En kan tenke seg en arm på et menneske, fra skulder til håndledd. Der kan det sies at armen har 5 Degrees of Freedom, Skulder kan bevege seg opp og ned (1), mot høyre og venstre (2). Albuen kan bevege seg opp og ned (3), og håndledd kan beveges opp og ned (4) og mot høyre og venstre (5). Dette gir et totalt antall Degrees of Freedom på 5. (TechTarget, 2009)

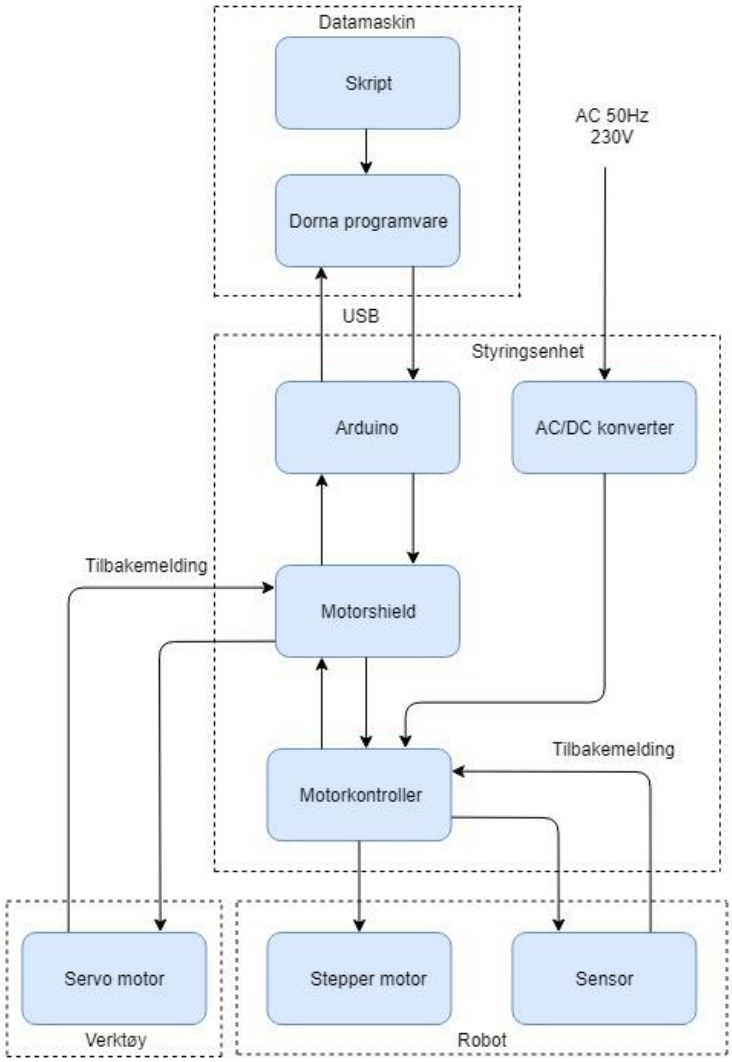


Figur 4 Dimensjoner, illusterer hvor sensorene er plassert og hvor mye hver akse kan rotere. Bilde hentet fra Dorna

Robotarmen som blir brukt i forbindelse med denne rapporten, har 5 Degrees of Freedom. Leddene er navngitt J0, J1, J2, J3 og J4. J0 kan rotere 358 grader horisontalt, den blir begrenset i software på grunn av ledningene til roboten. J1, J2 og J3 roterer vertikalt, der J1 kan rotere 230 grader, der den blir begrenset i software for å ikke kollidere med aksen J0. Det kan være aktuelt å begrense J1 mer, slik at den ikke kolliderer med gulvet. J2 kan rotere 260 grader, og er også begrenset for å ikke kollidere med leddet J1.

J3 kan rotere uten begrensninger, men en bør sette inn noen begrensninger i forhold til dimensjonene på utstyret som festes på J3. J4 kan også rotere uendelig og bør også bli begrenset av samme grunn som J3, da J3 og J4 kontrollerer samme ledd.

3.3 Blokkskjema



Figur 5 Blokkskjema for hvordan datamaskin, styringsenhet og robotarm er bygd opp.

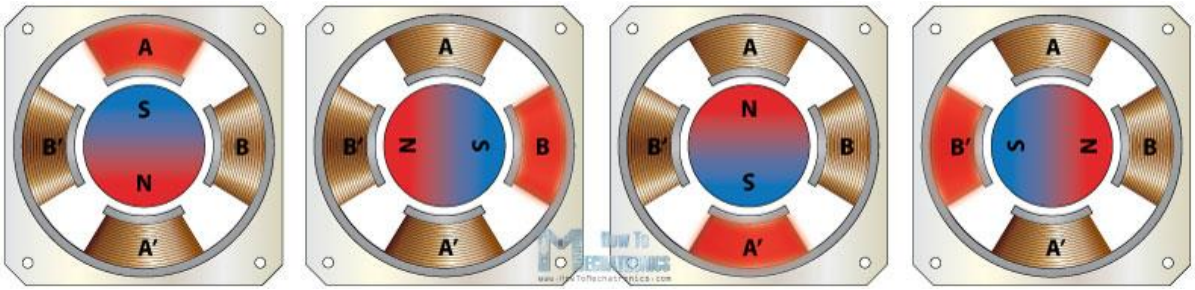
Blokkskjemaet viser hvordan datamaskin og styringsenhet kommuniserer. Og hvordan styringsenhet kommuniserer med robotarmen. Scriptet blir skrevet i Dorna programvaren og overføres via en USB-kabel til mikrokontrolleren. På mikrokontrolleren er det montert et motorshield (se kapittel 3.9 side 13). Servomotoren i griperen er koblet til motorshieldet. Motorshieldet kommuniserer også med motorkontrollene.

AC/DC konverteren konverterer vekselstrøm om til likestrøm. Den forsyner motorkontrollene med strøm for å kunne drive steppermotorene. Motorkontrollene mottar også signal fra sensorene om posisjon.

3.4 Stepper motor

Det er montert 5 steppermotorer på roboten, 1 for hvert ledd. Steppermotorene er drivkraften for at leddene beveges.

En generell steppermotor er en børsteløs DC motor som roterer trinn eller også kalt «steps». Dette gjør at steppermotoren kan beveges nøyaktig uten noen tilbakemelding. Steppermotoren består av en rotor som er en permanent magnet, som er omgitt av statorviklinger. For at rotoren skal bevege seg, settes det spenning på statorviklingene trinnvis og rotoren vil rotere ved hjelp av magnetisme.

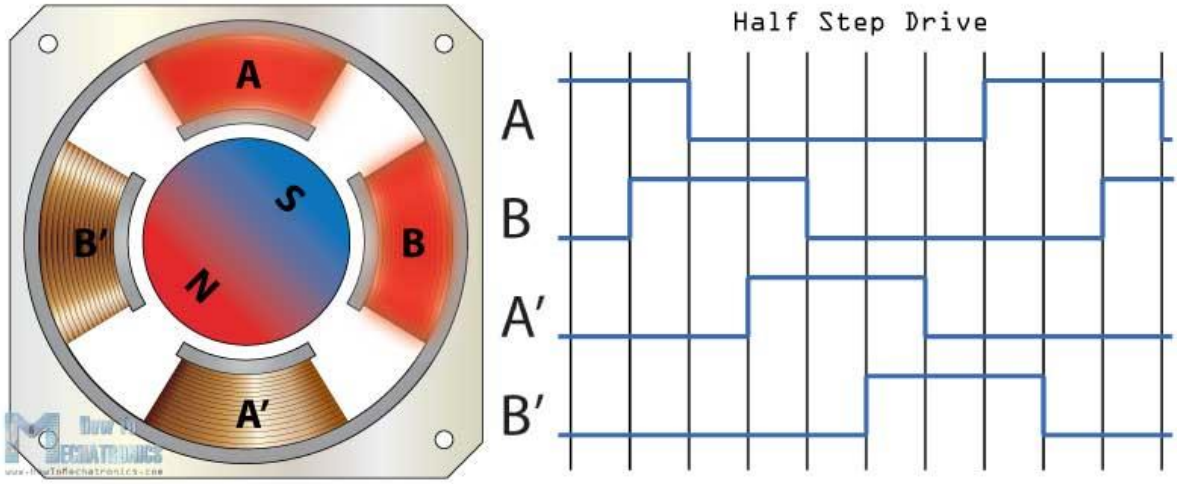


Figur 6 Prinsippet bak en steppermotor. Bilde hentet fra HowToMechathronics.com

Oppløsningen til en steppermotor blir bestemt av hvor mange steps steppermotoren må ha for å fullføre en hel rotasjon. Dette kan bli justert ved å bruke tannhjul, som gjør at selv om steppermotoren har tatt en hel rotasjon på 360 grader, kan akslingen på motoren bare ha rotert 45 grader. I figur 5 er oppløsningen som er illustrert 90 grader. Dette blir regnet ut slik:

$$\frac{360\ grader}{Antall\ steps} = \frac{360\ grader}{4\ steps} = 90\ grader$$

Det finnes en metode for å øke denne oppløsningen, som kalles for «Half stepping». Half-stepping blir utført med at man aktiverer 2 av viklingene ved siden av hverandre samtidig.



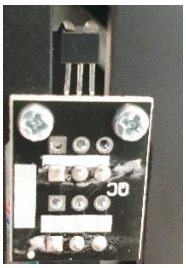
Figur 7 Illustrasjon bruk av Half step. Bilde hentet fra HowToMechathronics.com

I figur 7 er det samme konstruksjon som i figur 6, men her blir det brukt half step som gjør at det blir en oppløsning på 45 grader. Her får man 8 steps for en full rotasjon. (HowToMechatronics.com, 2019)

$$\frac{360\ grader}{8\ steps} = 45\ grader$$

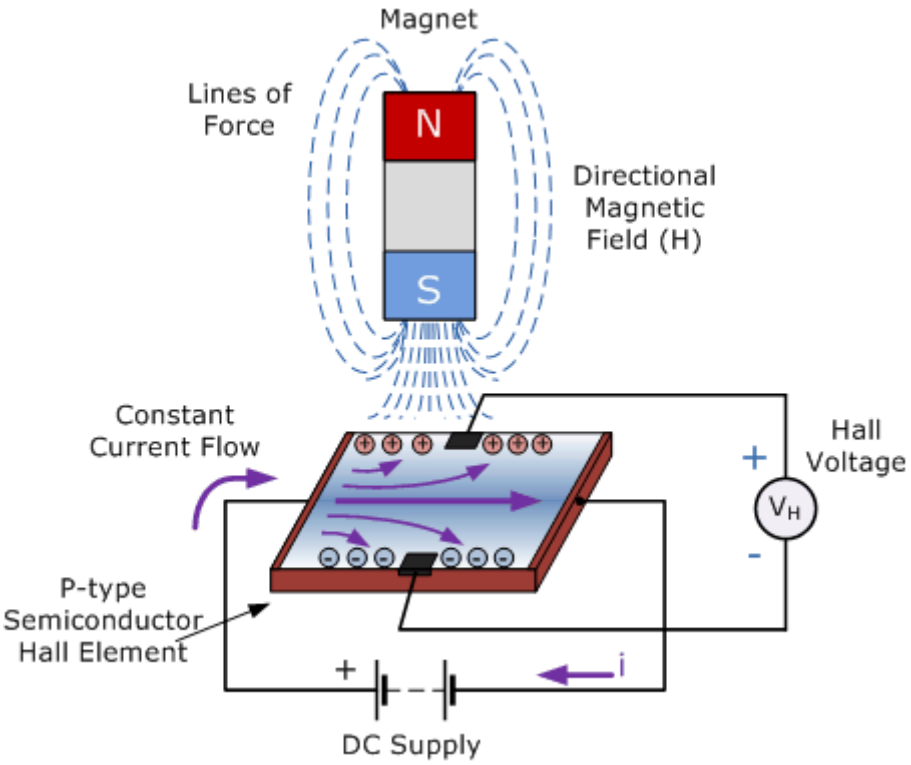
3.5 Hall-effektsensor

På roboten er det plassert 5 stykk Hall-effektsensorer. Sensorene er plassert på et fast punkt på hvert ledd. Det er plassert en magnet på leddet etter slik at når magneten passerer sensoren, har roboten fått en indikasjon på hvilken posisjon leddet er i. Dette er den eneste kjente posisjonen på leddet.



Figur 8 Hall-effektsensor montert på et av robotleddene

Hall-effektsensorer består av et tynt lag halvledermateriale som Galliumasenid (GaAs), Indiumantimonid (InSb) eller Indimarsenid (InAs). Dette gjør at sensoren har en kontinuerlig strøm gjennom seg selv. Når sensoren er i et magnetfelt, påvirker magnetfeltet halvledermaterialet, som fører til at elektronene (-) og elektronhullene (+) trekker seg til hver sin side av halvlederplaten. Det vil da oppstå en potensialforskjell, og en målbar spenning vil bli generert. (Electronics Tutorials, u.d.)

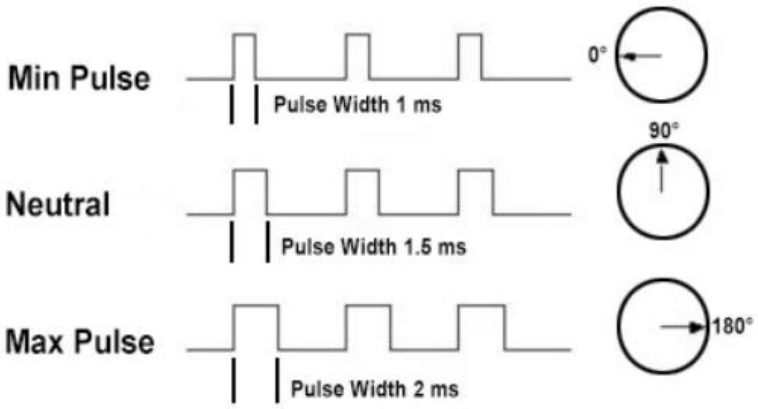


Figur 9 Illustrasjon om hvordan en Hall-effektsensor virker (Illustrasjon hentet fra Electronics Tutorials)

Sensoren vil gi et signal på 3,3 Volt til styringsenheten når den detekterer magneten. En rød lysdiode vil også lyse på kretskortet sensoren er montert på som indikasjon på at sensoren detekterer magneten. Når sensoren ikke detekterer magneten så vil spenningen ut være 0 Volt.

3.6 Servo motor

En Servo motor er en motor som har tilbakemelding på posisjon. Den ønskede posisjonen til Servo motoren blir sendt via en signal-ledning i form av elektriske pulser. Disse elektriske pulsene (PWM) er pulser med forskjellig bredde. For eksempel kan motoren forvente en puls hvert 20. millisekund, og lengden på pulsen bestemmer hvor langt motoren skal rotere. Så om pulsen er nøytral (1.5 ms) vil motoren stå i ro. Er pulsen lengre enn 1.5 ms vil motoren gå > 90 grader. Er pulsen lavere enn 1.5 ms vil motoren gå < 90 grader.



Figur 10 Illustrasjon av PWM for servomotor bilde hentet fra Jameco Electronics

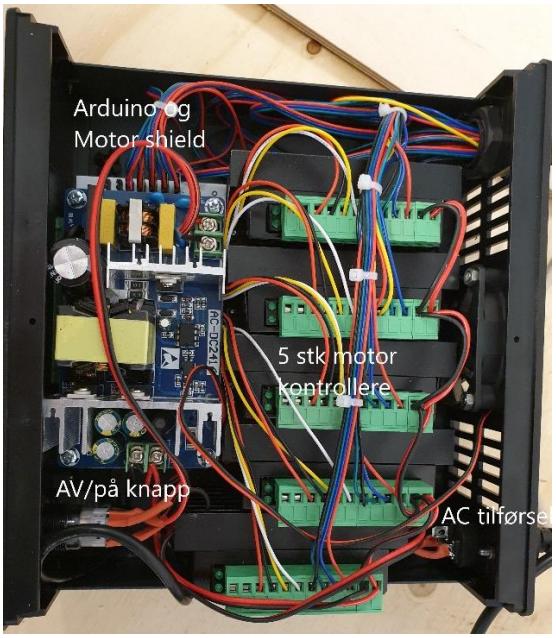
Figur 11 Servo motor. Blir brukt til å drive griperen

En Servo motor kan vanligvis bare snu/rotere 90 grader i hver retning, som gir en total vinkel på 180 grader. Det er også et potensiometer inni «motorhuset». Når rotoren i motoren roterer, endrer potensiometeret motstand og på den måten kan man vite hvilken posisjon motoren er i.

Servo motorer finnes både som AC-motorer og DC-motorer. AC servo motorer kan håndtere større strømmer og brukes mer i industrien. DC servo motorer er mest brukt i små applikasjoner. (Jameco Electronics, 2019)

3.7 Styringsenheten

Inneholder en Arduino Mega, Arduino SHIELD, 5 motorkontroller. Har utganger til en servo motor, laser, ekstra stepper motor og 8 I/O porter, som er tilkoblingspunkter som kan bli brukt som både inngang og utgang. Disse blir definert i programmet «Dorna».



Figur 12 Styringsenhet. har egen strømforsyning på baksiden, inneholder 8 digitale I/O tilkoblinger, en utgang for servomotor, en ekstra stepper motor og en laser. USB for tilkobling av en datamaskin

Controller and I/Os	
Input Voltage	110 – 240 V
Number of I/O pins	8
Number of supported Axes	6
Communication	USB
Power switch output	12 V Mosfet
Servo connector	6 V PWM control
External stepper	4 wire bipolar support
Dimensions	220 x 202 x 87 mm
Weight	2,5 Kg
Cable length	1 m

Tabell 2 Datablad for styringsenheten. Tabell hentet fra Dorna.

Figur 13 Dorna styringsenhet. Bilde hentet fra Dorna.

3.8 Arduino

Arduino er en Open-Source mikrokontroller. Mikrokontrolleren blir brukt til å introdusere folk til programmering, og profesjonelle aktører bruker den til mer avanserte styringer da den er veldig enkel å lære seg. Arduino er hjernen bak mange hjemmelagde og profesjonelle prosjekter. (Arduino , 2019)

I styringsenheten fra Dorna er det en Arduino Mega mikrokontroller. Denne har flere utganger enn en Arduino Uno.

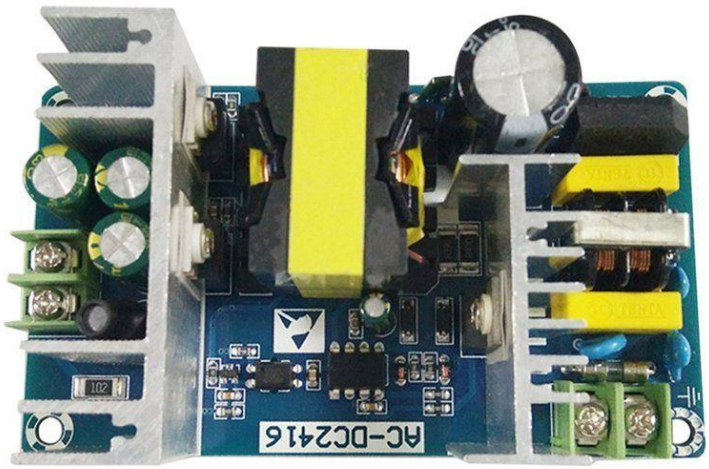
3.9 Shield

På en Arduino mikrokontroller kan man plassere et kort som kalles et «Shield». Et Arduino Shield blir brukt for å utvide mulighetene til mikrokontrolleren. Da Arduino-mikrokontrollere er Open Source kan hvem som helst utvikle et Shield til et eller flere formål. I vår kontroller er det et installert et motor-shield som gjør at mikrokontrolleren kan kontrollere steppermotorer og lese av sensorer.

3.10 Motorkontroller

Da det er steppermotorer montert på robotarmen, brukes det motorkontroller for steppermotorer. En motorkontroller fungerer som et mellomledd mellom mikrokontrolleren og en elektrisk motor. En mikrokrontoller har ikke kapasitet til å forsyne steppermotorene med strømmen de trenger for å operere. Motorkontrolleren sin oppgave er å forsyne steppermotoren med spenningen og strømmen den skal ha. Motorkontrolleren henter spenning fra en ekstern strømkilde (RobotShop Inc, 2019).

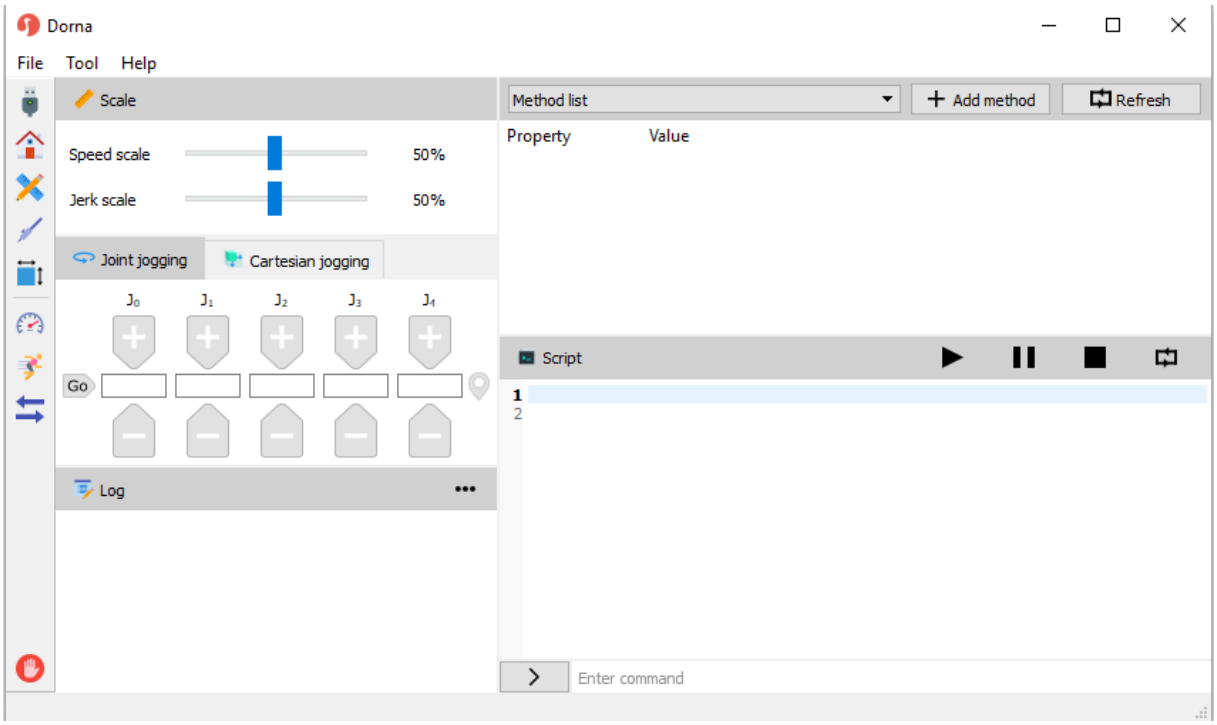
Motorkontrollerene får strøm fra en AC-DC konverter. AC-DC konverteren heter: «AC-DC4216». Denne konverterer 230V AC til 24V DC 6A. (eBay, u.d.)



Figur 14 AC-DC2416, konverterer 230V AC til 24V DC 6A. Bilde hentet fra eBay

3.11 Programvare

Det blir skrevet et script i programvaren fra Dorna. Denne programvaren heter «Dorna», og her blir programmet skrevet og roboten kontrollert. Programmet blir overført fra en datamaskin over micro-USB til styringsenheten. Programvaren kan bare fortelle hvor roboten er i teorien. Her blir også I/O tilkoblingene og definert.



Figur 15 Interface til Dorna programvaren. Programvaren er lokalt installert på en datamaskin. Mens prosjektet pågikk ble programvaren oppdatert, og ble i senere tid programmert via en serverbasert utgave.

3.11.1 Programmering

Mens programmering pågikk, gikk Dorna sine hjemmesider og programvare under en stor oppdatering. Som gjorde at scriptene vi hadde lagd begynte å oppføre seg rart. Etter litt testing oppdaget vi at i den gamle programvaren var koordinatene til roboten i «resting position» (se kapittel 3.11.2) [(J0), (J1), (J2), (J3), (J4)] byttet verdier fra [(0), (0), (0), (0)] til [(0), (-145), (90), (0), (0)]

Den nye programvaren var en serverbasert versjon. Som inkluderte et animert bilde av robotens posisjon.

Kommandoene som blir brukt for å flytte robotarmen er av stilen:

{"command":"move","prm":{"path":"joint","movement":0, “J0”: 90, “J1”: 20, “J2”; 4}}

Her blir J0 flyttet til 90 grader, J1 til 20 grader og J2 til 4 grader.

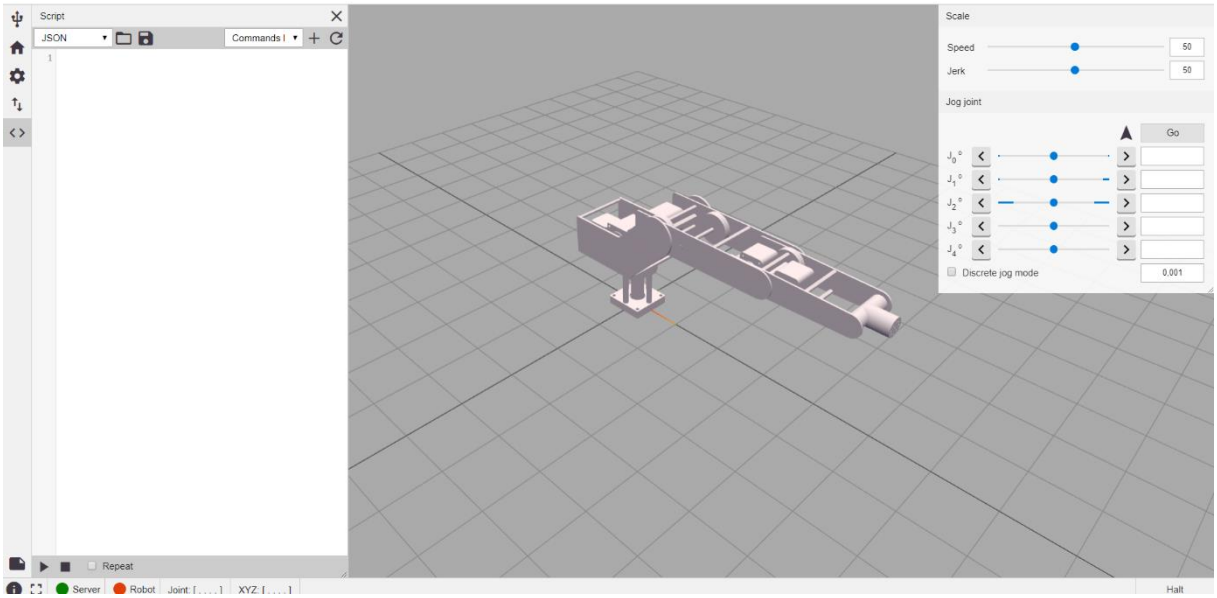
«Movement» som er nevnt i kommandolinjen kan endres fra 0 til 1. Hvor 0 betyr absolutt, og 1 betyr relativ. Det vil si at om en bruker absolutt, vil alle vinklene som blir gitt gå ut ifra vinklene den har i «resting position». Brukes relativ, vil alle vinklene flytte seg i forhold til posisjonen den hadde sist.

«Path» som er nevnt i kommandolinjen kan endres mellom «joint» og «line».

«Joint» vil si at den beveger alle leddene på likt for å komme frem til koordinatene som er oppgitt.

«Line» betyr at enden på robotarmen beveger seg lineært til koordinatene som er oppgitt. Det er ikke alltid robotarmen klarer denne lineære bevegelsen på grunn av leddenes begrensninger.

Forklaringen på hvordan koordinatsystemet fungerer, blir tungvint forklart på hjemmesidene til Dorna og krever noe forståelse for vektorberegning.



Figur 16 Interface til serverbasert Dorna programvare. Denne versjonen blir programmert via nettleser. Inneholder også en 3D-animasjon av roboten som viser posisjonen til de forskjellige leddene.

3.11.2 Homing

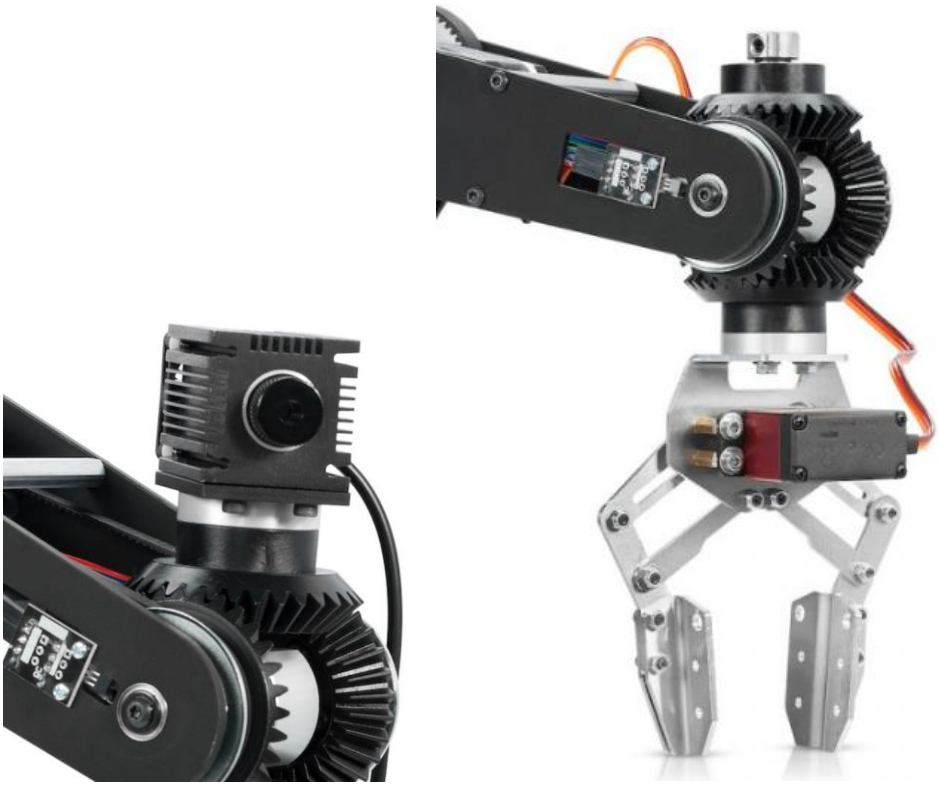
Hver gang styringsenheten starter opp, må robotarmen igjennom en prosess som blir kalt for Homing. Dette er for å sette robotarmen i en «start posisjon», denne posisjonen er den eneste kjente posisjonen styringsenheten har tilbakemelding om. Alle posisjoner blir regnet ut fra denne start posisjonen.

Før styringsenheten blir slått på, må robotarmen bli satt i en posisjon som kalles «resting position». Om dette ikke blir gjort, kan robotarmen få problemer under Homing prosessen. Leddene kan kollidere med hverandre, armen kan svinge ned i gulvet eller hele robotarmen kan rotere utenfor begrensningene som er satt i programvaren.

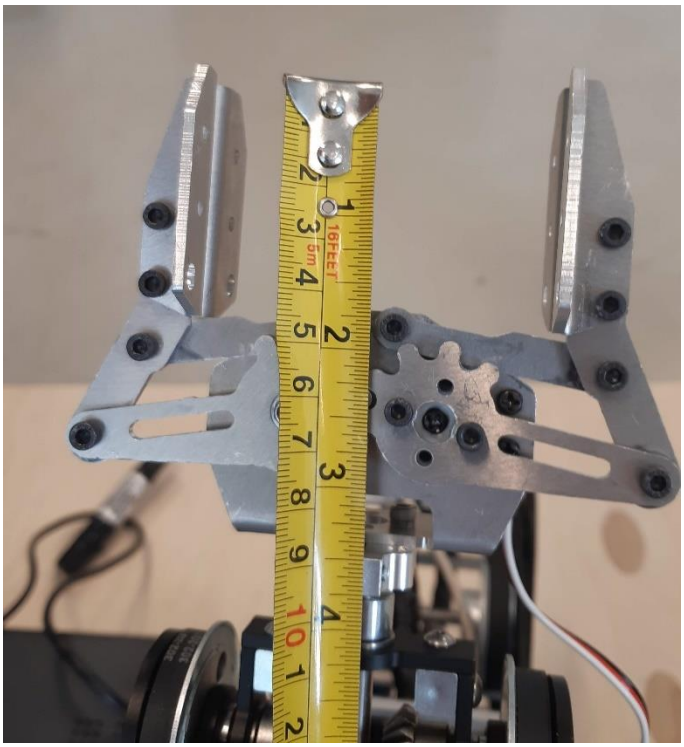
Figur 17 Roboten satt i "resting position"

3.12 Verktøyhoder

På roboten er det mulig å koble til ulike verktøyhoder. Verktøyhodene kan være alt ifra laser, sveiseapparat, kamera, gripeklo etc. etter hva som trengs for å kunne utføre den bestemte arbeidsoppgaven.
Verktøyhodene blir ikke sett på som en del av roboten, men er en egen del som blir montert på robotarmen. Styringsenheten har tilkoblinger for servo og laser, som gjør at noe av utstyret som kan monteres på robotarmen vil kunne bli styrt fra styringsenheten.

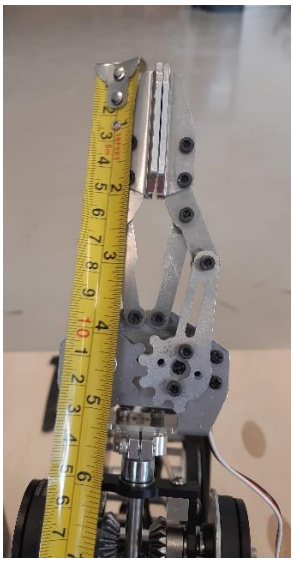


Figur 18 Laser gravør for gravering av tre, plastikk, papir og griper som festes på roboten. Bildene er hentet fra Dorna



Figur 19 Når griperen er åpen når den 95mm fra roboten og det er 65mm mellom kontaktflatene

Griper kan gripe rundt fra 65mm tykke objekter. Ett problem med konstruksjonen av denne griperen er at lengden vil variere på hvor tynt objektet den skal gripe rundt er, på figur 19 kan en se at når griperen er helt åpen så er den 95mm lang, mens i figur 20 er den 145mm lang.



Figur 20 Når griperen er helt lukket så er den 145mm lang

Dette gjør at den er har lett for å være upresis når den skal gripe tak i en artikkel da den blir lengre og vil skyve artikkelen bort fra roboten når den lukkes.

4 Praktisk modell

For å simulere prosessen med å åpne skyvedøren til CNC-maskinen og plassere en artikkel i chucken for å så lukke skyvedøren ble det bygd en liten modell. Størrelsen på modellen ble begrenset på grunn av størrelsen på roboten og materialene som var tilgjengelig. Da CNC-maskinen har en skyvedør som må åpnes manuelt, ble det enighet å ha en skyvedør som roboten kan åpne eller lukke i modellen også.

Figur 21 Skyvedøren som det skal lages en nedskalert modell av.

4.1 Modell

Hjulene som er montert på skyvedøren er noen hjul som vanligvis brukes på garderobeskap, kan kjøpes på Montèr. Skapdøren som blir brukt som skyvedør var tidligere brukt som en demo hos Montèr, den har målene: 395mm, 348mm og 15mm. Rammen er laget av tre og er 640mm høy, 430mm lang og har en bredde på 95mm. Åpningen er 360mm høy og 405mm lang.

På bunnplaten er det montert 2 halve overgangslister for gulv på 500mm som er montert slik at de danner et spor som hjulene til skyvedøren kunne bevege seg mellom. Overgangslisten ble handlet hos Montèr.

Håndtaket brukt i modellen er handlet hos Montèr.

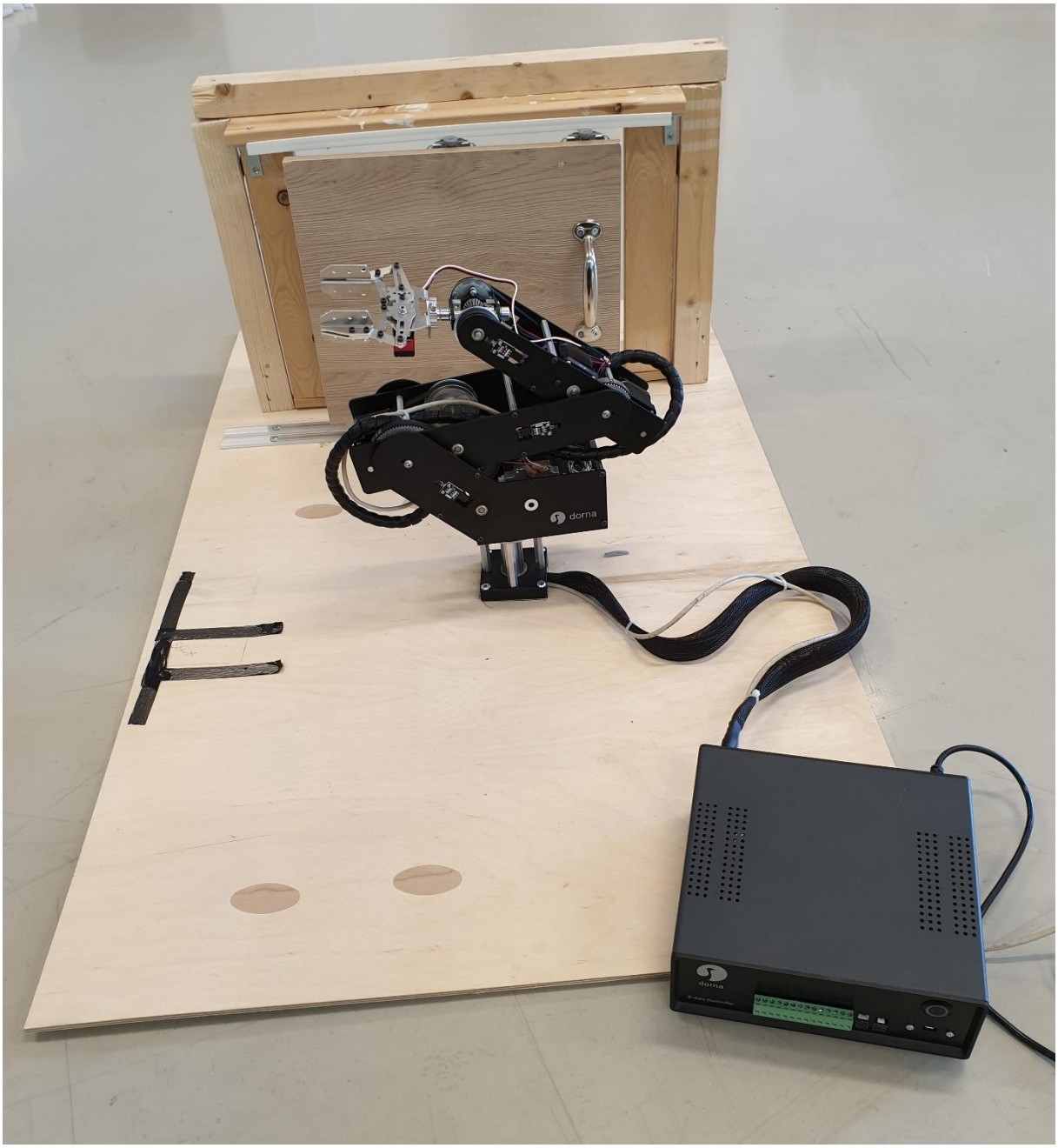
Figur 22 Nedskalert modell av skyvedøren. På bildet vises: Skyvedør, Håndtak, overgangslist for gulv, skillevegg for kanaler, hjul til garderobeskap, festebraketter, diverse trematerialе og bunnplate

Den store platen i bunn er handlet hos Coop OBS bygg. Denne er stor nok til at både skyvedøren og roboten kan monteres på den. Målene på platen er 1200mm, 748mm, 7mm. Det å ha roboten og modellen på samme plate vil gjøre programmeringen av roboten mer lettvint da modellen alltid har samme posisjon og avstand fra roboten.

Det er montert noe hvit PVC i toppen av rammen som egentlig blir brukt som skillevegg i elektrokanaler. Skilleveggen er 500mm lang. Bredden ble kortet ned til 45mm bred og har en T-profil i enden som er 15mm høy. Skilleveggen blir brukt i modellen som skinne til hjulene i toppen av skyvedøren slik at skyvedøren kan stå rett og ikke faller ned. Skilleveggen ble handlet hos Onninen.

Det er brukt skruer for å hindre at skyvedøren glir ut av rammen den er montert i. Skulle døren bli åpnet for mye vil det bakerste hjulet stoppe i skruen. Det er også satt en skrue for å hindre at døren går for langt når den blir lukket. Disse skruene sørger også for at døren kan bli plassert i en kjent posisjon på en enkel måte. Skruene er montert i bunnplate og igjennom den hvite skilleveggen, skruene kan ikke sees på figur 22.

For å stive av rammen ble det brukt metall vinkler med hull til skruer, disse ble handlet hos Clas Ohlson.



Figur 23 Modell for CNC-maskin og roboten fra Dorna med styringsenhet. Det svarte feltet indikerer hvor artiklene skal plasseres ved start.

4.2 Griper

På griperen er det montert en servomotor. Servomotoren hadde en kort kabel på bare 15 cm og det fulgte med en skjøtekabel på 100 cm. Denne var ikke lang nok, og da ble den delt opp og skjøtet med en CAT5-kabel på midten slik at kontakten ikke ble ødelagt siden det ikke var flere av dem. For å skjøte ledningene ble det brukt skjøteklemmer av typen 3M Scotchlok UY2 som kan fås tak i hos Clas Ohlson.

4.3 Oppsett for programmeringen

Før programmeringen kunne begynne, måtte det lages et flytskjema for hva roboten skulle gjøre, dette er for å få en enklere oversikt over programmeringen av styringsenheten til robotarmen. Flytskjemaet er utarbeidet ut ifra Figur 2 side 2.

Prosessen roboten skal utføre er:

1. Programmet startes.
2. Robotarmen åpner døren til CNC-maskin.
3. Hente artikkel som er lagt klar for maskinering.
4. Gi signal til CNC-maskin om at Chuck skal åpnes.
5. Plassere artikkel inn i chuck i CNC-maskin.
6. Gi signal til CNC-maskin om at Chuck skal stenges.
7. Robotarmen lukker døren.
8. Gi signal til CNC-maskin om den skal starte.
9. Motta signal fra CNC-maskin om at den er ferdig.
10. Robotarm åpner døren til CNC-maskin.
11. Robotarm tar rundt artikkel.
12. Gi signal til CNC-maskin om at Chuck skal åpnes.
13. Plassere artikkel i beholder.
14. Robotarm lukker døren.
15. Robotarm plasseres i klar stilling.

Avhengig om hvor mange artikler som skal prosesseres, gjentas steg 3-13.

Figur 24 Flytskjema for robotens prosess. Bokser i grønn indikerer enkel implementering til styringsenheten. Blå bokser indikerer ekstra komponenter som enkelt kan legges til. Røde bokser indikerer at styringsenheten ikke klarer det fra fabrikkstandard.

5 Site Acceptance Test

En SAT er en test som blir utført etter fullført installasjon og konfigurasjon. Før produktet har blitt sendt ut, har det vært utført en FAT (Factory Acceptance Test). FAT er en test som blir gjennomført av produsenten for å teste om produksjonen har blitt utført korrekt og alt virker som det skal før produktet blir sendt ut til forbruker (AutomationForum, 2015).

Ved å utføre en SAT kan det vurderes om en skal satse på å oppskalere roboten. Det er laget noen script for å teste om roboten er like presis og robust som produsenten har oppgitt.

Da roboten er tenkt til å sette artikler inn i en CNC-maskin, er det viktig at robotarmen er nøyaktig, at den kan utføre samme oppgaver flere ganger etter hverandre og at den kan bære vekten av artiklene. Testene er konstruert spesielt for å sjekke om robotarmen kan bære vekten, er like presis og repeterbar som er oppgitt i tabell 1.

5.1 Repeterbarhet

Repeterbarheten ble testet i 2 ulike situasjoner. Den første prosessen hvor det simulerte tilfellet hvor roboten skal sette artikler inn i CNC-maskin. Den andre prosessen ble det markert opp 4 punkter som robotarmen skulle treffe flere ganger etter hverandre over lengre tid.

5.1.1 Simulert arbeidstilfelle

Her ble modellen som ble laget i kapittel 4 brukt. Og prosessen som er beskrevet i kapittel 4.2 ble programmert. Roboten gjorde denne prosessen rett over 2 timer i 2 forskjellige hastigheter hvor den ble overvåket for å se at prosessen gikk som den skulle. Steppermotorene ble også sjekket for varme ved å kjenne på dem hvert 10 min.

Roboten klarer denne prosessen uten problemer. Patentet til gripekloen og materialet som ble brukt (tre) gjorde at artikkelen flyttet seg noen millimeter for hver gang griperen slapp artikkelen. Dette problemet vil ikke ha noe å si i den ferdige installasjonen, da artikkelen skal settes inn i en Chuck, som «låser» artikkelen på plass før gripeklo slipper. Å øke eller senke hastigheten hadde ingen utslag på resultatet.

5.1.2 Punkt test

Det ble i denne testen laget et script for å treffe 4 punkter. Ruten bort til hvert punkt ble med intensjon programmert til å bevege alle leddene. På denne måten ble alle leddene testet samtidig siden alle ledd må vær i samme posisjon for å treffe punktet.

Testen ble utført i 2 forskjellige hastigheter i 1 time hver, med 1 dag mellom hver gang. Steppermotorene ble sjekket for varme med å kjenne på de fysisk med fingre mens roboten var i gang.

Denne prosessen klarte roboten også uten problemer. Roboten treffer punktene hver gang uten problemer. Og hastighet har ingenting å si på resultatet.

5.2 Testing av belastning

I databladet (Tabell 1) er det beskrevet at robotarmen skal klare 1,1 kg i nyttelast. Da det ble festet 976 gram 14,5 cm fra rotasjonsaksen (160 gram egenvekt på gripeklo, 816 gram på artikkel), sviktet Leddene J3 og J4 når disse ble satt i bevegelse. Dette kan forklares med momentet som blir påført leddene. Og man kan beregne dette med formelen for dreiemoment. Det er sett bort fra massen til leddene, og massesenter.

Formelen for dreiemoment er:

$$\tau = G * r$$

Hvor «G» er kraften i Newton, og «r» er vektoren fra rotasjonssenteret til kraftens angrepspunkt.

$$G = m * g$$

Hvor «m» er masse i kg, og «g» er tyngdeakselerasjonen (9,81m/s^2).

Avstanden fra endepunktet til rotasjonsaksen uten verktøy er (4,4 cm), og forteller oss at det maksimale momentet den garantert skal klare er:

$$1{,}1kg * 9{,}81\,{}^{m}/_{s^2} * 0{,}044m = 0{,}474Nm$$

Fester man på en griper blir armen lenger (18 cm til ytterpunkt) som gjør at regnestykket blir:

$$1{,}1kg * 9{,}81\,{}^{m}/_{s^2} * 0{,}18m = 1{,}942Nm$$

Som er en differanse på 1,468Nm. Dette forteller at det er ingen garanti for at roboten skal klare å holde 1,1kg på enden etter gripekloen er montert.

I testtilfellet hvor artikkel var festet 14,5cm fra rotasjonsaksen:

$$0{,}976kg * 9{,}81\,{}^{m}/_{s^2} * 0{,}145m = 1{,}388Nm$$

Dette er 0,914Nm mer enn det som er beregnet mulig.

Man kan med formelen og data som er oppgitt regne ut hvor mye vekt som skal kunne monteres på endepunktet:

$$\frac{0{,}474Nm}{9{,}81\,{}^{m}/_{s^2} * 0{,}18m} = 0{,}268kg$$

Og trekker man fra egenvekten på gripekloen:

$$0{,}268kg - 0{,}160kg = \ 0{,}108kg$$

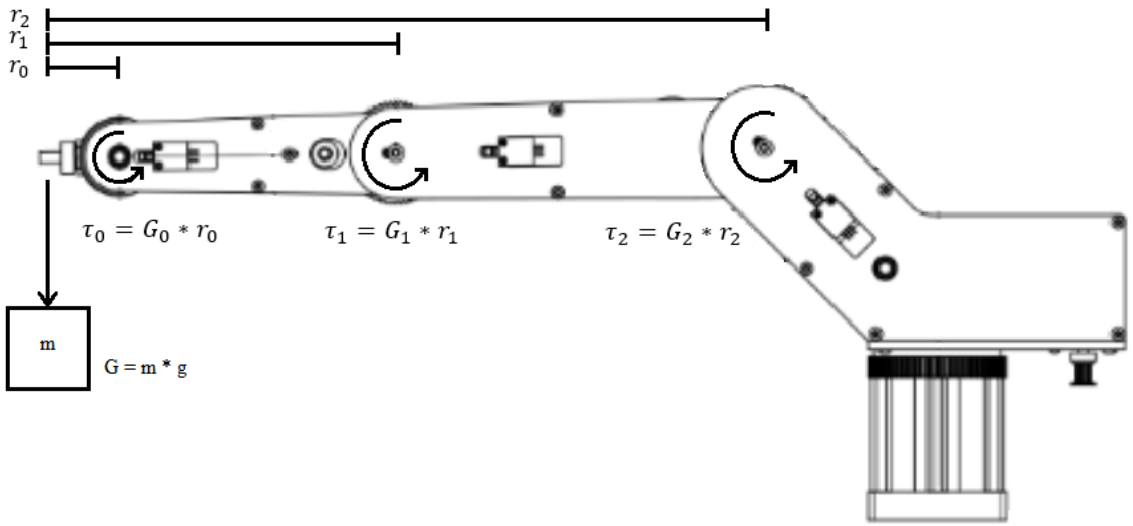
På enden av gripekloen skal roboten klare å ha en nyttelast på minst 108 gram.

Under testing ble det funnet ut av at den klarer å løfte 879 gram 16,5 cm fra rotasjonsaksen før leddene J3 og J4 gir etter.

Da blir det totale momentet roboten klarer:

$$(0{,}879kg + 0{,}160kg) * 9{,}81\,{}^{m}/_{s^2} * 0{,}165m = 1{,}681Nm$$

Det ble utført en test for å sjekke hvor mye vekt roboten klarer før et av leddene svikter. Denne vekten ble målt til 2190g. Det var leddet J1 som sviktet. Med disse resultatene kan det sies at robotarmen klarer å løfte 1,1 kg. Men om massesenter blir plassert for langt ut ifra rotasjonsaksene J3 og J4, vil ikke robotarmen klare dette.



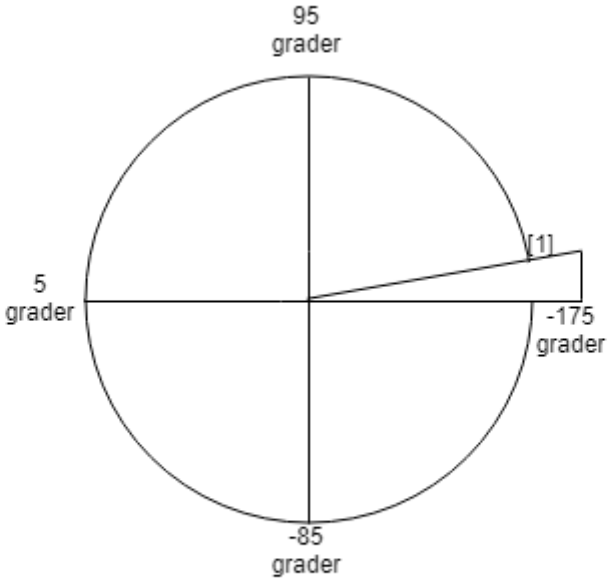
Figur 26 Illustrasjon som viser hvordan dreiemoment påvirker de ulike leddene.

5.3 Rotasjon nøyaktighet

I databladet er det oppgitt at «base swing» (J0) kan rotere 358 grader. Den stopper før en hel rotasjon slik at det blir unngått at kablene som forsyner robotarmen blir tvunnet om hverandre og tar skade. Dette er en software begrensning som blir satt i programmet, det ble oppdaget at roboten kan ta en hel runde om den under oppstart er på en posisjon som ikke er i «resting position» (se kapittel 3.11.2 side 17).

Testen ble utført med å plassere en sprittusj i griperen som er montert på roboten, og deretter lage et script som fikk roboten til å rotere fra -175 grader til 175 grader mens sprittusjen var i kontakt med bunnplaten. Dette tilsvarer totalt 350 grader og siden 360 grader er en hel omdreining, ble det forventet at sirkelen ikke ble sluttet og at 10 grader gjenstod. Og at ved hjelp av trigonometriske funksjoner kan det regnes ut hvor mye som gjenstår for å få sirkelen hel.

$$\cos^{-1}\left(\frac{Hosliggende\ katet}{Hypotenus}\right) = Vinkelen$$



Figur 27 Den runde sirkelen illustrerer sirkelen roboten lagde. Roboten står i senter. Start er på -175 grader og stopp på punkt [1]. Satt opp en rettvinklet trekant ved hjelp av punktene hver 90 grad som også ble markert med roboten, for å enklere kunne regne ut resterende vinkel.

$$cos^{-1}\left(\frac{28{,}8cm}{29{,}3cm}\right) = 10{,}60\ grader$$

Her kan det ha oppstått noe unøyaktighet da det ble brukt en tykk tusj. Og måleverktøyet som ble brukt til å måle lengden på hosliggende katet og hypotenus var en meterstokk, som bare viser mål ned til 1 mm. Ved bruk av flere desimaler i regnestykket blir svaret mer nøyaktig og tilnærmet 10 grader som er svaret vi ville ha.

6 Helse, Miljø og Sikkerhet

Helse, Miljø og Sikkerhet (HMS) handler om å ivareta liv, helse og materiell.

Dette er vesentlig viktig for bedrifter for å sikre ett godt arbeidsmiljø og redusere risikoen for å forhindre ulykker og uønskede hendelser som skader på personell, materiell og miljø. Utfra HMS får man gode retningslinjer og bidrar til å oppnå bedre effektivitet og kvalitet av produkter.

For å hindre skader på personell, bør det vurderes om roboten skal arbeide innenfor oppmerket eller avgrenset område. På denne måten hindrer det at personer kommer i veien for roboten eller i klem mellom de forskjellige leddene.
Med å bruke robot til å ta de tunge løftene så reduserer man risikoen for belastningsskader på personell.

7 Økonomi

I mai 2019 er prisen på en robot fra Dorna 1500 USD og griper 100 USD, dette tilsvarer en pris på 14 444,16 NOK (DNB, 2019)

Til sammenlikning koster en robot fra «Universal Robots» uten verktøy 27 000 euro (Edstroms, 2019), dette tilsvarer 275 022 NOK (DNB, 2019).

Et estimat av tiden som trengs for å skrive programmet for en som kjenner til programmet er omtrent 2 timer, i tillegg så må roboten testes. En må derfor beregne rundt 1 time i testing og justering av programmet.

Roboten bruker rundt 0,069 kWh i strøm, med en strømpris på 0,5239 kr per kWh (Fjordkraft, 2019). Det er beregnet at det er i gjennomsnitt 150 arbeidstimer i måneden. Da blir det et strømforbruk på 5,42 kroner i måneden. Dette tilsvarer 65 kr i året. Det er sett bort fra prisendringer på strøm, som varierer daglig. Det er også sett bort fra at noen måneder har mindre arbeidsdager enn andre som jul, påske og andre røde dager.

Da en operatør har i gjennomsnitt en årslønn på 454 200 kroner. Vil robotarmen ha en stor økonomisk fordel om den skulle bli brukt i prosessen som er beskrevet i kapittel 4.3, figur 24 side 24.

Driftskostnadene som er beregnet, er gjort ut ifra produktet som ble levert. Denne må oppskaleres om den skal bli brukt til den automatiske prosessen og vil av den grunn få et høyere strømforbruk.

8 Diskusjon

Dette kapitelet vil ta for seg hvilke problemer som oppstod, løsninger på disse problemene og noen forbedringer for at systemet skal bli mest mulig optimalt.

8.1 Robotens posisjon

Om robotarmen skulle kollidere med et hinder, gjenstand, person etc. slik at den ikke klarer å flytte seg til ønsket posisjon, vil programvaren tro at roboten har flyttet seg til programmert posisjon, men roboten vil ikke ha gjort dette fysisk. Da må roboten kalibreres på nytt for å få riktig posisjon.

Om det skulle skje at roboten skulle treffe på en hindring eller at den ikke klarer å utføre en handling, har ikke programmet noen tilbakemelding på dette og robotarmen vil fortsette med neste kommando som ligger i skriptet. Dette kan være skadelig for roboten, og andre gjenstander eller personer. Dette gjør at det er en risiko å la roboten jobbe alene uten oppsyn. Blir derimot programmet laget på en ordentlig måte, vil ikke dette skje uten eksterne påvirkninger. Legges det til endebrytere på skyvedøren og lignende vil programmet automatisk stoppe om signal ikke blir sendt til styringsenheten, fordi den ikke kommer videre i scriptet. Det bør plasseres en nødstopp bryter koblet inn på strømtilførselen som er lett tilgjengelig for å stanse roboten om en situasjon som gjør at roboten mister sin egen posisjon skulle oppstå.

8.2 Lastesystemet

Når roboten skal plukke opp flere artikler så trengs det et system for å sørge for at alle artiklene blir plukket opp av robotarmen. Slik programvaren er laget, er det mest gunstige å bruke en rampe. Da artiklene som skal inn i CNC-maskinen er runde kan disse stables oppå hverandre. Og hver gang robotarmen plukker opp en artikkel, ruller neste artikkel ned og havner i samme posisjon som den forrige.

8.3 Strømbrudd

Ved strømbrudd vil steppermotorene som er montert på roboten miste sin spenning og derav sitt magnetfelt. Dette gjør at robotarmen «kollapser».

Så lenge mikrokontrolleren er koblet til en datamaskin vil programmet fortsette å gå så lenge datamaskinen har strøm. Dette er et problem om det er en bærbar datamaskin som er tilkoblet, da disse har batteri i seg. Og vil av den grunn ikke bli avslått ved strømbrudd. Om strømbruddet blir ordnet vil styringsenheten gå direkte på. Og programmet fortsetter ikke fra

hvor det slapp, men hvor den har kommet mens strømbruddet har vært. Og problemstillingen som er beskrevet i kapittel 8.1 vil oppstå.

En kan også ha en UPS som sørger for at roboten ikke mister spenningen ved strømbrudd. Ved strømbrudd vil ikke CNC-maskinen kunne gi signal til robotens styringsenhet om å fortsette prosessen. Og robotarmen vil holde sin posisjon helt til UPS er utladet. Dette vil ikke hindre roboten fra å kollapse når en trykker inn nødstopp eller skrur av styringsenheten.

8.4 Programvaren

Som nevnt i kapittel 3.11.1 så blir programvaren fortsatt oppdatert med store oppdateringer. I perioden der denne rapporten ble skrevet så fikk programvaren en oppdatering som gjorde at våre script som ble laget i den gamle versjonen ikke fungerte lenger, og scriptet måtte endres. Oppdateringen var at en kunne bruke en serverbasert løsning i nettleser som gjør det lettere å kontrollere roboten over nettverk fra en annen datamaskin.

8.5 Python

Det er mulig å programmere styringsenheten via Python. Dorna har lagt ut en API til Python som gjør at programmeringen skjer på samme metode som i Dorna programvaren. Raspbetty Pi kan bli brukt som en løsning for å kunne operere roboten trådløst slik at en sparer plass med å ha en liten datamaskin ved siden av roboten.

Å bruke Python til å programmere, vil gjøre mulighetene litt mer fleksible. Da kan man bruke teller funksjoner, løkker etc. som ikke kan brukes i Dorna programvaren.

8.6 Sensorer

Hall-effektsensorene som er montert på hvert ledd på robotarmen, er lett utsatt for ytre påvirkninger. Om robotarmen blir flyttet manuelt, kan man med et uhell være borti en av sensorene og bøye de. Dette vil da gjøre at robotens startposisjon etter «homing» som er beskrevet i kapittel 3.11.2 blir noe forskjøvet.

En kraftsensor installert på robotarmen vil kunne registrere om den kolliderer med noe og vil da kunne stanse prosessen slik at den ikke ødelegger gjenstander og seg selv. Om en tar og logger kraften på en kollisjon og når den plukker opp en gjenstand eller beveger seg, vil man kunne programmere roboten til å kjenne igjen en kollisjon. Med en kraftsensor vil det også være mulig å få en indikasjon på om roboten har fått tak i artikkelen da kraften blir annerledes.

8.7 Maksimal nyttelast

Fabrikanten lover at roboten skal kunne løfte 1,1 kg i nyttelast. Dette ble bekreftet i testen som er nevnt i kapitel 5.2. Det er ikke anbefalt å overstige denne vekten.

Skulle det bli montert på en lengre arm på leddene J3 og J4 blir vekten roboten klarer lavere.

8.8 Programmering

Dorna programvaren er enkel å sette seg inn i, men det tar tid og programmere. Om man i programmeringen skulle være borti en hindring, slik at roboten havner ut av sin kjente posisjon, må en kalibrere og kjøre scriptet slik at robotarmen ender tilbake til den siste kjente posisjonen for å prøve å styre den på nytt.

Programvaren har ingen opptaksfunksjon. Med opptaksfunksjon menes det at softwaren tar og lagrer bevegelser som blir gjort fysisk med robotarmen. Programvaren klarer da å gjengi de samme bevegelsene da de har blitt lagret.

Programmeringen av prosessen beskrevet i kapittel 4.3 figur 24 side 24 ble også programmert i Python. Se Vedlegg A.

8.9 Ekstra akse

Med en ekstra akse vil roboten kunne være litt mer praktisk. Og kunne gjøre linære bevegelser mer presist. Det er utganger for en ekstra steppermotor på styringsenheten. Styringsenheten er laget for å få montert på en ekstra akse. Om ikke steppermotorene blir oppgradert, vil nyttelasten den klarer å løfte bli noe redusert.

8.10 Griper

Griperen som ble brukt har en flate kontaktflate noe som gjør at det er vanskeligere å plukke opp runde/sylinder formede artikler. En forbedring vil være å ha runde kontaktflater på griperen eller med et adaptivt grep som roboten kan justere etter hva arbeidsoppgaven krever.

Figur 28 Adaptiv griper fra Robotiq, laget for Universal Robots colabrative roboter. De sølvfargede «fingrene» kan byttes ut. Hentet fra Universal Robots hjemeside.

8.11 Problem med servomotor

Under testing opplevdes det at servo motoren til griperen brant opp. Dette var på grunn av overbelastning, da griperen aldri kom frem til ønsket posisjon når den skulle gripe tak i og holde fast en artikkel. Da gikk den varm over tid og brant seg. Det bør være en tilbakemelding fra servoen om hvilken posisjon den har.

9 Konklusjon

Målet med rapporten var å teste om en kommersiell robot kan brukes i industrien. Etter å ha testet og brukt robotarmen og programvaren fra 28 februar 2019 til 7 mai 2019, er det konkludert med at robotarmen er god nok til små oppgaver. Selve prosessen om å sette et metall artikkel inn i en CNC-maskin vil være en oppgave robotarmen og styringsenheten klarer.

Nøyaktigheten til robotarmen ble oppgitt til å være 0.1mm. Robotarmen treffer samme posisjon hver gang og nøyaktigheten er god nok til oppgaven den er tenkt til.

Repeterbarheten til robotarmen ble testet i to ulike situasjoner der den ble testet i et tenkt arbeidstilfelle og en test der den skulle treffe fire forskjellige punkter flere gjentatte ganger over lengre tid. Resultatet av testene viser at roboten klarer å utføre disse oppgavene flere gjentatte ganger.

Nyttelasten til robotarmen kan være et problem om vekten blir plassert for langt vekk fra rotasjonsaksene J3 og J4. Momentet kan bli for stort. Dette bør en tenke på om gjenstander rundt 1 kg blir montert for langt utpå leddet som blir kontrollert av J3 og J4.

Robotarmen i seg selv er god nok til å utføre mer avanserte oppgaver, den største begrensningen er styringsenheten.

Det kan kobles til en Raspberry Pi som kan legges inn i styringsenheten for å spare plass og for å kunne styre robotarmen trådløst.

Bibliografi

Arbeidstilsynet. (2019, mai). *Arbeidstilsynet*. Hentet fra arbeidstilsynet.no: https://www.arbeidstilsynet.no/tema/ergonomi/manuelt-arbeid/tungt-arbeid/

Arduino . (2019). *Arduino*. Hentet fra https://www.arduino.cc/en/guide/introduction

AutomationForum. (2015, Januar). Hentet fra https://automationforum.in/t/what-is-the-meaning-of-sat-and-fat-site-acceptance-test-and-factory-acceptance-test/227

DNB. (2019, 05 2). *DNB hjemmeside*. Hentet fra www.dnb.no

Dorna. (2019, Mai 5). *Hjemmeside for Dorna*. Hentet fra www.dorna.ai

eBay. (u.d.). Hentet fra https://www.ebay.com.au/itm/150W-AC-110V-220V-To-6A-DC-24V-Switching-Power-Supply-Board-Power-Module-LOT-IB-/163132239625

Edstroms. (2019, 04 30). Hentet fra https://automation.edstroms.com/

Electronics Tutorials. (u.d.). Hentet fra https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html

Fjordkraft. (2019, 05 08). *fjordkraft.no*. Hentet fra https://www.fjordkraft.no/privat/strompriser/

HowToMechatronics.com. (2019). Hentet fra https://howtomechatronics.com/how-it-works/electrical-engineering/stepper-motor/

Jameco Electronics. (2019). *Jameco Electronics*. Hentet fra https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html

Kodegenet. (u.d.). Hentet fra https://kodegenet.no/track/raspberrypi/courses/raspi_java/chapter/raspi_java_ch3

Olsen, O. A. (2005). *Instrumenteringsteknikk.* Tapir akademisk forlag.

Robotic Industries Association. (2019). *A Tribute to Joseph Engelberger*. Hentet fra https://www.robotics.org/joseph-engelberger/unimate.cfm

Robotiq. (2019). Hentet fra Robotiq.com: https://robotiq.com

RobotShop Inc. (2019, Mai 5). *RobotShop*. Hentet fra https://www.robotshop.com/community/tutorials/show/basics-what-is-a-motor-controller

TechTarget. (2009, Juli). *TechTarget*. Hentet fra https://whatis.techtarget.com/definition/degrees-of-freedom

Utdanningsdirektoratet. (2019, Mars 19). *utdanning.no*. Hentet fra https://utdanning.no/yrker/beskrivelse/cnc-operator

Vedlegg

Python program

```
from dorna import Dorna

robot = Dorna()

x=0

#Tilkoble robot
robot.port_list()
robot.update_firmware()
robot.connect()

#Kalibrering
robot.homed()
robot.home("j0")
robot.home("j1")
robot.home("j2")
robot.home("j3")

robot.set_joint({"j1":0})
robot.set_joint({"j2":0})

while x < 10: #Antall ganger programmet skal gjentas

#Hovedprogram
#Bevege seg til dorhandtak
    robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -98.073, "j1": -29.937, "j2": 0, "j3": 3.6715, "j4": -32.4265}})
```

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -98.073, "j1": -29.937, "j2": 0, "j3": 3.6715, "j4": -32.4265}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -98.073, "j1": -29.937, "j2": -49.5530, "j3": -5.8985, "j4": -48.8615}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -98.073, "j1": -53.923, "j2": -43.349, "j3": -16.9655, "j4": -48.8615}})

#Apne dor

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 10000, "j0": -80.165, "j1": -53.923, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 10000, "j0": -58.349, "j1": -53.923, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

#Hente Artikkel

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -60, "j1": -53.923, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -60, "j1": -25.22, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": 30.36, "j1": -25.22, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": 30.36, "j1": -90.506, "j2": -31.749, "j3": -17.521, "j4": -3.9530}})

{"command": "sleep", "prm": 0.5}

{"command": "set_io", "prm": {"servo": 420}}

{"command": "sleep", "prm": 0.5}

#Flytte artikkel inn i CNC maskin

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -40.689, "j1": 0, "j2": -29.072, "j3": -17.521, "j4": -0.264}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": 0, "j2": -29.072, "j3": -62.521, "j4": -22.454}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": -52.43, "j2": -29.072, "j3": -62.521, "j4": -22.454}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": -81.629, "j2": 2.797, "j3": -62.521, "j4": -22.454}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": -114.325, "j2": 32.955, "j3": -53.9785, "j4": -32.1255}})

{"command": "sleep", "prm": 0.5}

{"command": "set_io", "prm": {"servo": 100}}

{"command": "sleep", "prm": 0.5}

#Bevege seg ut av CNC maskin

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": -81.629, "j2": 2.797, "j3": -62.521, "j4": -22.454}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": -52.43, "j2": -29.072, "j3": -62.521, "j4": -22.454}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": 0, "j2": -29.072, "j3": -62.521, "j4": -22.454}})

#Lukke dor

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": 0, "j2": 0, "j3":0, "j4": -22.454}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -60, "j1": 0, "j2": 0, "j3":0, "j4": -22.454}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -60, "j1": -53.923, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 10000, "j0": -80.165, "j1": -53.923, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 10000, "j0": -98.073, "j1": -53.923, "j2": -43.349, "j3": -16.9655, "j4": -48.8615}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 10000, "j0": -97.165, "j1": -53.923, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -97.165, "j1": 0, "j2": 0, "j3": 0, "j4": 0}})

#Resting position

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": 0, "j1": 0, "j2": 0, "j3": 0, "j4": 0}})

#Bevege seg til dorhandtak

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": 0, "j1": 0, "j2": 0, "j3": 0, "j4": 0}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -98.073, "j1": -29.937, "j2": 0, "j3": 3.6715, "j4": -32.4265}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -98.073, "j1": -29.937, "j2": -49.5530, "j3": -5.8985, "j4": -48.8615}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -98.073, "j1": -53.923, "j2": -43.349, "j3": -16.9655, "j4": -48.8615}})

#Apne dor

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 10000, "j0": -80.165, "j1": -53.923, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 10000, "j0": -58.349, "j1": -53.923, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

#Hente artikkel

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -59, "j1": 0, "j2": -29, "j3": -17.5, "j4": -0.2}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": 0, "j2": -39.5, "j3": -62.5, "j4": -22.454}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": -64.5, "j2": -39.5, "j3": -62.521, "j4": -22.454}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": -52.43, "j2": -28.092, "j3": -62.521, "j4": -22.454}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": -81.629, "j2": 2.797, "j3": -62.521, "j4": -22.454}})

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": -114.325, "j2": 32.955, "j3": -53.9785, "j4": -32.1255}})

{"command": "sleep", "prm": 0.5}

{"command": "set_io", "prm": {"servo": 420}}

{"command": "sleep", "prm": 0.5}

#Bevege seg ut av CNC maskin

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": -81.629, "j2": 2.797, "j3": -62.521, "j4": -22.454}})

```
    robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": -52.43, "j2": -28.092, "j3": -62.521, "j4": -22.454}})

    robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": -64.599, "j2": -39.465, "j3": -62.521, "j4": -22.454}})

    robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -83.718, "j1": 0, "j2": 0, "j3": 0, "j4": 0}})


#Lukke dor

    robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": 28, "j1": 0, "j2": 0, "j3":0, "j4": -22.454}})

    robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -60, "j1": 0, "j2": 0, "j3":0, "j4": -22.454}})

    robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": -60, "j1": -53.923, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

    robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 10000,"j0": -80.165, "j1": -53.923, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

    robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 10000,"j0": -79.165, "j1": -53.923, "j2": -43.349, "j3": -24.3435, "j4": -38.9035}})

    robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000,"j0": -79.165, "j1": 0, "j2": 0, "j3": 0, "j4": 0}})


#Tilbake til start posisjon

    robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": 0, "j1": 0, "j2": 0, "j3": 0, "j4": 0,}})

    x=x+1


#Tilbake til start og avslutt

robot.play({"command": "move", "prm": {"movement": 0, "path": "joint", "speed": 20000, "j0": 0, "j1": 0, "j2": 0, "j3": 0, "j4": 0,}})
```