



Høgskulen på Vestlandet

Bacheloroppgave

Elektro

ING3055

Predefinert informasjon

Startdato:	31-03-2020 09:00	Termin:	2020 VÅR
Sluttdato:	07-05-2020 14:00	Vurderingsform:	Norsk 6-trinns skala (A-F)
Eksamensform:	Bacheloroppgave Elektro med muntlig presentasjon/eksaminasjon		
SIS-kode:	203 ING3055 1 BOPPG 2020 VÅR HAUGESUND		
Intern sensor:	Andrés Franklin Olivares Lopez		

Deltaker

Kandidatnr.: 301

Informasjon fra deltaker

Tittel *: Robot styrt sveising med videre automatisering

Engelsk tittel *: Robot controlled welding with further automation

Navn på veileder *: Andrés Franklin Olivares Lopez

Sett hake dersom ja **Egenerklæring *:** Ja
besvarelsen kan brukes **Inneholder besvarelsen** Nei
som eksempel i **konfedensielt**
undervisning?: **materiale?:**

Jeg bekrefter at jeg har ja
registrert
oppgavetittelen på
norsk og engelsk i
StudentWeb og vet at
denne vil stå på
vitnemålet mitt *:

Gruppe

Gruppenavn: (Anonymisert)

Gruppenummer: 1

Andre medlemmer i Deltakeren har innlevert i en enkeltmannsgruppe
gruppen:

Jeg godkjenner avtalen om publisering av bacheloroppgaven min *

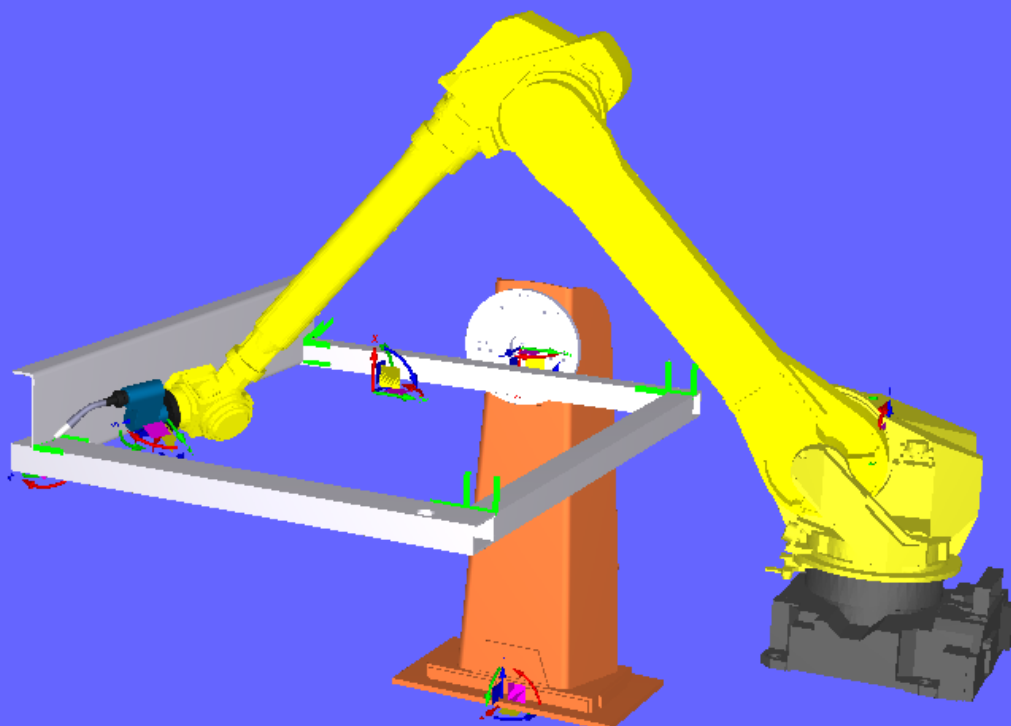
Ja

Er bacheloroppgaven skrevet som del av et større forskningsprosjekt ved HVL? *

Nei

Er bacheloroppgaven skrevet ved bedrift/virksomhet i næringsliv eller offentlig sektor? *

Ja, LEVEL Power & Automation



Robot styrt sveising med videre automatisering

KARI NES

BACHELORPROSJEKT UTFØRT VED HØGSKULEN PÅ VESTLANDET – AVD. HAUGESUND – STUDIE FOR
INGENIØRFAG – ING3055 - ELEKTRO

Oppgavetekst

BACHELORPROSJEKT

Studenten(e)s navn: Kari Nes

Linje & studieretning Bachelor i ingeniørfag, Elektro

Oppgavens tittel: *Robotsveiser med videre automatisering*

Oppgavetekst:

LEVEL Power & Automation AS har en kunde som ønsker å automatisere sveisingen av containere slik at produksjonen blir mer effektiv. Kunden A. Olufsen Ship & Offshore AS produserer Push-in containerne i Aluminium. De ønsker i utgangspunktet å bruke robot til å sveise panelene til containeren (gulv, vegger og tak) samt dørene. Containeren vil da ligge «flatpakket» på lager og monteres etter ordre. Dette reduserer arealbehovet stort i forhold til å ha ferdige containere på lager. Selve monteringen av containeren vil gjøres for hånd.

Oppgaven skal i hovedtrekk se om det er mulig å få en robot til å sveise panelene, samt kommunisere med en mekanisk ramme som roterer panelene slik at roboten kommer til alle steder.

Videre skal oppgaven: Simulere sveiseoperasjonen eller operasjoner, programkode for operasjoner som blir simulert (Inkludert kode for kommunikasjon med mekanisk ramme), forklare hvordan programvare koden kan testes på en UR5 robot for å kvalitet sikre koden, identifisere krav og eventuelle problemstillinger som må løses, komme med forslag til en robot som kan utføre denne jobben.

Endelig oppgave gitt: 17.03.20

Innleveringsfrist: Torsdag 7.mai 2020 kl. 12.00

Intern veileder: Andrés Franklin Olivares Lopez

Tommy Vika

Ekstern veileder:
emailadresse ekstern
veileder:

Tommy.vika@levelgroup.no

Godkjent av
studieansvarlig:
Dato:

H. Spanghus
14/04-20

Høgskulen på Vestlandet

Studie for ingeniørfag

Bjørnsonsgt. 45

5528 HAUGESUND

Oppgavens tittel Robot styrt sveising med videre automatisering		Rapportnummer (Fylles ikke ut)
Utført av Kari Nes		
Linje Elektroingeniør y-vei, industriell automatisering		Studieretning Studie for ingeniørfag
Gradering: Åpen	Innlevert dato: 07.05.2020	Veiledere: Intern: Andrés Franklin Olivares Lopez Ekstern Tommy Vika

Ekstrakt

LEVEL Power & Automation har en kunde som ønsker å automatisere sveisingen av containere slik at produksjonen blir mer effektiv. De ønsker i utgangspunktet å bruke robot til å sveise panelene til containeren. Det har blitt undersøkt mulighet for å automatisere sveisingen av container panel ved bruk av robot og automatisert jigg.

I dette prosjektet ble det laget simulering og programkode for sveise-operasjoner som skal utføres av robot, foreslått robot som kan brukes, forklart hvordan programvaren kan testes på for eksempel en UR5 robot, laget PLS program for rotering av jigg og identifisert problemstillinger. Simulering og programkode ble laget ved bruk av *off-line* programmering RoboDK v.4.2.4.

Forord

Denne bacheloroppgaven markerer avslutningen på et 3 årig studie på Høgskulen på Vestlandet, campus Haugesund ved linjen elektroingeniør y-vei industriell automatisering. Jeg ser frem til å starte yrkeslivet og lære mer. Oppgaven har vært spennende og lærerik. Mottaker av produkt, som er RoboDK stasjonene med koder og simuleringer av operasjonene som ble laget, er veileder Tommy Vika.

Jeg ønsker å gi en stor takk til mine veiledere, Andrés Franklin Olivares Lopez fra høyskolen, og Tommy Vika fra LEVEL Power & Automation. Tusen takk for gode innspill og god veiledning. Tusen takk til LEVEL Power & Automation for at dere ordnet PLS programvare slik at jeg kunne lage PLS programmet. Tusen takk til Arne Olufsen fra A. Olufsen Ship & Offshore AS for omvisning på verkstedene og god veiledning til hvordan containerne ønskes sveises. Tusen takk til RoboDK for 4 måneders gratis utdanningslisens til bruk i oppgaven for å lage simulering og programkode. Til slutt vil jeg takke familie og venner for god støtte gjennom hele studiet.

Definisjoner og forkortelser

OLP	Off-line programming
CAD	Computer-aided design
RoboDK	Programvare for OLP og simulering
Autodesk Inventor	Programvare for 3D modellering, CAD program
MIG	Metal inert gas
Inert	Edelgass
TIG	Tungsten inert gas
Tungsten	Wolfram
PLS	Programmerbar logisk styring
FT-sensor	Force Transmitter, kraft sensor
OPC UA	Open Platform Communications Unified Architecture
CPU	Central Processing Unit
RS485	Seriell kommunikasjon metode for datamaskiner og enheter
FTP	File Transfer Protocol, filoverføringsprotokoll
USB	universell seriebuss, kobler enheter til datamaskin
UR-robot	Universal robot
EN AW-6082	Type Legering
T6	Type Hardhetsgrad
EN AW 5754	Type Legering
H111	Type Hardhetsgrad
TAST	Påstandstidspunktet, representerer kontrasten mellom to målinger langs en tidslinje
IP-grad	Kapslingsgrad til elektrisk utstyr
IP-adresse	Internet Protocol adress, adresse som tildeles en enhet
URP-fil	Et URP-program kan ha en eller flere SCRIPT-filer som underprogrammer. URP-programmer transformeres til SCRIPT-programmer bak kulissene før utførelse.
SCRIPT-fil	En type skriptfil som utfører instruksjonene den inneholder når den åpnes
DOF	Degrees Of Freedom, grader av frihet
ISO	International Organization for Standardization. Den internasjonale standardiseringsorganisasjonen
LOTO	Lock Out, Tag Out
SFC	Sekvensielle funksjonskart, programmeringsspråk PLS

UPS	Uninterruptible power supply, avbruddsfri strøm
V	Spenning målt i Volt
I	Strøm målt i Ampere
AC-lysbue	Vekselstrøm lysbue ved sveising
Jigg	Verktøy for å sette deler sammen på en bestemt måte
PL	Performance Level
Postprosessor	En robotpostprosessor definerer hvordan robotprogrammer må genereres for en spesifikk robotkontroller. Konverteringen fra RoboDK-simuleringen til et spesifikt robotprogram gjøres av en postprosessor.
I/O	Input/Output, innganger og utganger
u	Spesifikk energi
e	Effektiviteten i prosessen
v	Hastighet

Innholdsfortegnelse

Oppgavetekst	i
Ekstrakt.....	iii
Forord.....	iv
Definisjoner og forkortelser	v
Figurer	ix
Tabeller.....	x
Formler	x
Sammendrag	xi
Summary	xii
1. Innledning.....	1
1.1 Bakgrunn	1
1.2 Formål og problemstilling.....	1
1.3 Avgrensninger.....	2
1.4 LEVEL Power & Automation	2
1.5 A. Olufsen Ship & Offshore AS.....	3
2. Metode sveising	5
2.1 Teori og metode for aluminium	5
2.1.1 TIG-metode.....	6
2.1.2 MIG-metode	6
2.2 Defekt Sveis	7
2.3 Kvalitet sveis	7
2.4 Gass	8
2.5 Sveise parametere.....	8
2.5.1 Avstand og hastighet.....	8
2.5.2 Strømstyrke og spenning.....	9
2.5.3 Parameter for container panel	10
2.6 Sveise operasjoner container panel	11
2.7 Sveising med robot.....	12
3. Metode robot og PLS.....	13
3.1 Introduksjon Robot.....	13
3.2 Introduksjon PLS.....	14
3.3 Programmering.....	15
3.4 Programvare RoboDK	16
3.4.1 RoboDK Autodesk Inventor tilleggsprogram.....	17

3.4.2 Simuleringer og programkode robot strategi.....	17
3.5 Test programvare	22
3.5.1 Risikovurdering	22
3.5.2 Starte opp roboten	23
3.5.3 Forslag testing	24
3.5.4 Overføre program.....	25
3.6 PLS styring av automatisert jigg	26
3.6.1 Flytskjema Program	27
3.6.2 Rotering i simuleringen	28
3.7 Robot forslag	29
3.8 Vedlikehold.....	32
4. Sikkerhet.....	33
4.1 Risikoanalyse	33
4.2 Sikkerhet ved sveising	35
4.3 Programvare sikkerhet	35
4.4 Forslag sikkerhet.....	36
5. Diskusjon	39
5.1 Andre OLP programvarer	39
5.2 Sveiseutstyr	39
5.3 Hypotetisk investeringsanalyse	40
5.4 Utfordringer.....	41
5.5 Strømbrudd, UPS.....	42
5.6 Sensorer.....	43
5.7 OPC UA	44
5.8 Kalibrering og nøyaktighetstest.....	45
6. Konklusjon	47
7. Kilder.....	48
8. Vedlegg.....	A
Vedlegg 1	B
Vedlegg 2	E
Vedlegg 3	G

Figurer

Figur 1. PUSH-IN container fra A. Olufsen Ship & Offshore AS [7].....	1
Figur 2. Containeren som er uten tak. CAD-tegning fra A. Olufsen Ship & Offshore AS.....	2
Figur 3. Tidslinje som viser viktige hendelser for LEVEL Group AS [1].	3
Figur 4. Verkstedet til LEVEL Power & Automation, Haugesund. Bilde tatt 28.04.2020.....	3
Figur 5. Verkstedsområdet til A. Olufsen Ship & Offshore AS [7].	4
Figur 6. PUSH-IN container fra A. Olufsen Ship & Offshore AS [7].....	4
Figur 7. Illustrasjon av TIG-metode [9].....	6
Figur 8. Illustrasjon av MIG-metode [9].	6
Figur 9. Resultater av for lang, OK og kort avstand ved sveising [14].	8
Figur 10. Resultater ved OK, rask og treig hastighet ved sveising [14].	9
Figur 11. Resultater av for lav, høy og OK strømstyrke ved sveising [14].	9
Figur 12. Illustrasjon av delene til rammen, bilde tatt i Autodesk Inventor 2020, 15.02.2020.	11
Figur 13. Illustrasjon nærbilde av hvor roboten må sveise rundt 2 ganger. Bilde tatt i Autodesk Inventor 2020, 15.02.2020.	11
Figur 14. Mål av rammen som det er blitt simulert program for. CAD tegning tilsendt av A. Olufsen Ship & Offshore AS.	12
Figur 15. Unimate, den første industriroboten [74].....	13
Figur 16. UR robot fra Universal Robots [25].	14
Figur 17. Omron PLS [27].....	14
Figur 18. Kuka robot som sveise en del av rammen. Bilde tatt i RoboDK programvare, 15.03.2020. ..	16
Figur 19. Fra Autodesk Inventor 2020, RoboDK tillegget. Bilde tatt i Autodesk Inventor 2020, 25.03.2020.....	17
Figur 20. Simulering av sveising ramme til venstre og simulering av sveising ramme med korrugerte plater til høyre. Bilde tatt av simulering- film som ble lagd fra RoboDK programvare, 30.04.2020.....	17
Figur 21. Bilde av simulering sveiseoperasjon dør ramme, bilde tatt 04.05.2020.....	18
Figur 22. plassering av robot i forhold til komponent. Bilde tatt i RoboDK programvare, 30.04.2020. 18	18
Figur 23. Viser skjermbilde fra programmet som ble laget. Bilde tatt i RoboDK programvare, 30.04.2020.....	19
Figur 24. Curve follow project. Bilde tatt i RoboDK programvare, 30.04.2020.....	20
Figur 25. Kode fra simulering. Bilde tatt i RoboDK programvare, 30.04.2020.	21
Figur 26. Litt av hovedprogrammet. Bilde tatt i RoboDK programvare, 30.04.2020.	21
Figur 27. Håndholdt kontroll [76], fra Universal Robots.	23
Figur 28. UR5 robot som beveger seg fra ulike posisjoner. Bilde tatt i RoboDK programvare, 05.03.2020.....	24
Figur 29. UR5 robot som beveger seg over CAD tegning. Bilde tatt i RoboDK programvare, 05.03.2020.	24
Figur 30. Bilde av skjermen på en UR5 kontroll boks [36].	25
Figur 31. Prosessflyt av overføring programkode, laget 28.02.2020.	25
Figur 32. Programkode SFC for program til jigg til venstre, og rotering av panel til høyre ved bruk av KUKA KP1-H enkeltakseposisjoner. Bilde til venstre tatt i Omron programvare CX-programmer, 20.03.2020. Bilde til høyre i RoboDK, 02.05.2020.....	26
Figur 33. Viser flytskjema over program for PLS og program for robot, laget 01.04.2020.	27
Figur 34. KUKA KP1-H enkeltakseposisjoner sammen med robot og panel i programvare RoboDK. Bilde tatt i RoboDK programvare, 20.03.2020.	28
Figur 35. Viser hvordan robot må komme til nærmeste ende uten å krasje i seg selv. Bilde tatt i RoboDK programvare, 30.03.2020.	29

Figur 36. Viser robot som sveiser ende lengst vekke. Bilde tatt i RoboDK programvare, 30.03.2020..	30
Figur 37. Robot utstrakt til venstre og ovenfra til høyre, sammen med illustrasjon av største panel. Bilde tatt i RoboDK programvare, 26.03.2020.	30
Figur 38. Fanuc M-710iC/45M robot [42].	31
Figur 39. Eksempel på å sette en vegg inn i RoboDK programvare. Bilde tatt i RoboDK programvare, 26.04.2020.	35
Figur 40. Robotcelle sikkerhet eksempel [51].	37
Figur 41. Kemppi A7 MIG welder [62].	40
Figur 42. Tilbakebetalings diagram, laget i Excel, 27.03.2020.	41
Figur 43. Bilde av dørlås hvor det må sveises manuelt på PUSH-IN containeren [7].	42
Figur 44. On-Line UPS hvordan koblet [56].	43
Figur 45. Baumer SmartReflect [77].	43
Figur 46. OPC UA fra RoboDK programvare. Bilde tatt i RoboDK programvare, 30.03.2020.	44
Figur 47. ABB-robot som utfører ballbar test [66].	45
Figur 48. PDF-rapport for robot nøyaktighet resultater fra RoboDK [66].	45

Tabeller

Tabell 1. Risikovurdering laget 25.03.2020, [35].	22
Tabell 2. Spesifikasjoner Fanuc M-710iC/45M [43].	31
Tabell 3. Eksempler på andre roboter som kan vurderes. Laget i Excel, 27.04.2020, [67].	32
Tabell 4. Risikovurdering laget 25.03.2020 [35].	34
Tabell 5. Performance Level. [53] og [https://www.iso.org/].	36
Tabell 6. Eksempel nedbetaling hvert år, laget i Excel, 27.03.2020.	41

Formler

Formel 1. Sveisehastighet	10
Formel 2. Sveisehastighet med verdier	10

Sammendrag

I dette prosjektet var oppgaven å lage simulering og programkode for sveise-operasjonene som skal utføres av robot, foreslå robot som kan brukes, forklare hvordan programvaren kan testes på for eksempel en UR5 robot, lage PLS program for rotering av automatisk jigg og identifisere problemstillinger. Containeren består av gulv, vegger og dør som er laget av aluminium. Da det er mye som skal sveises på containeren så ønsket LEVEL Power & Automation å se på muligheten til å effektivisere dette. Ved bruk av robot og rotering av panel med PLS styring så kan sveiseprosessen for alle panelene bli automatisert. Montering av den komplette containeren gjøres manuelt.

Anbefalt løsning for programmering av roboten vil være å bruke RoboDK programvare som er *off-line* programmering med simulering. Roboten som ble anbefalt er Fanuc M-710iC/45M. Den har 6 akser, lang nok rekkevidde og er testet i simuleringen. Utstyret som blir valgt til sveiseoperasjonene må tegnes i CAD for å implementeres i RoboDK for en nøyaktig simulering. Simuleringen brukte sveiseutstyr som allerede var i RoboDK sitt bibliotek.

PLS kan brukes til å styre automatisert jigg. I løsningen har det blitt brukt *timer* for å få robot til å vente til roteringen er fullført. Ved bruk av sensorer kan man få PLS og robot til å samarbeide.

Prosessen ble:

- Starte robotprogram for sveising.
- Robot gir signal til PLS når den er ferdig med å sveise øvre del av panelet (sensor signal)
- PLS mottar signalet og snur panelet slik at robot kan sveise den andre siden
- Roboten venter til jigg har snudd (setter *timer*)
- *Timeren* er over og robot sveiseprogram for andre siden starter

Man kan erstatte denne løsningen med OPC UA kommunikasjon for da slipper man å bruke for eksempel *timer*. OPC UA kan brukes for å motta informasjon om sensorenes tilstand og sende til Robot eller PLS. Da bruker man I/O utganger som er mulig å programmere i RoboDK.

Det ble totalt 8 programmer for operasjonene. Program for valgt panel må lastes opp til robot. Når prosessen er ferdig for valgt panel stoppes programmet og rutine for LOTO iverksettes slik at neste panel kan klargjøres. Program lastes opp til robot for valgt panel og startes. Brukeren velger selv hvilket panel som skal sveises og laster opp ønsket program til robot.

Sikkerhet er en viktig del ved robot som utfører sveising og det ble konkludert med PL nivå d som det anbefales i Norsk Standard ISO 10218.

Videre arbeid kan være å lage program for den andre containeren, som er større, eller lage sveise program for komplett container i stedet for at det gjøres manuelt. Man kan også endre programmet til å bruke OPC UA kommunikasjon for å unngå bruk av *timer*. Den automatiske jigg ble ikke laget så derfor kan denne også arbeides med. Kan også teste sveiseparametre, programvare og kode laget for simuleringene. Det kan også ses på mulighet til andre løsninger ved for eksempel bruk av 2 UR roboter i stedet for 1 industrirobot.

Summary

In this project, the task was to create a simulation and program code for the welding operations performed by a robot, suggest which robot can do the job, explain how the software can be tested on for example a UR5 robot, create a PLC program for rotating an automatic jig and identify problems. The container consists of a floor, walls and a door made of aluminum. Since there is a lot to be welded to the container, LEVEL Power & Automation wanted to look at the possibility of making this more efficient. By using a robot and rotating the panel with PLC control, the welding processes for all panels can be automated. The assembly of the complete container will be done manually.

The recommended solution for programming the robot is to use RoboDK software, which is software offering off-line programming with simulation. The recommended robot is the Fanuc M-710iC/45M which has 6 axes, sufficient range and have been tested in the simulation. The equipment selected for the welding operations must be drawn in a CAD-software to be implemented in RoboDK for accurate simulation. The simulation existing welding equipment from RoboDK's library.

A PLC can be used to control the automated jig. In the solution, timers have been used to make the robot wait for the rotation to be complete. The robot and the PLC can work together by using sensors.

The process:

- Start the robot welding program.
- Robot gives a signal to the PLC when it has finished welding the upper part of the panel (sensor signal)
- The PLC receives the signal and turns the panel so that the robot can weld the other side
- The robot waits until the panel is rotated (sets timer)
- When the timer is over, the robotic welding program for the other side starts

It is possible to replace this solution with OPC UA communication. OPC UA can be used to receive information about the state of the sensors and send it to the Robot or the PLC. This can be done by using I/O (inputs and outputs) that are possible to program in RoboDK.

A total of 8 programs were programmed for this operation. The program for the selected panel must be uploaded to the robot. When the process is completed for the selected panel, the program will stop and routine for LOTO is implemented so that the next panel can be prepared. Program is uploaded to the robot for the selected panel. The user picks which panel to weld and uploads the program for that panel to the robot.

Safety is an important part of a robot that performs welding and it was concluded with PL level d as recommended in Norwegian Standard ISO 10218.

Further improvements may be to create a program for the other container, which is bigger, or create a welding program for the complete container instead of doing it manually. You can also alter the program to use OPC UA communication to avoid using timers. The automatic jig was not created; therefore, you can create that. It is also possible to test welding parameters, software and code created for the simulations. It is also possible to look at the possibility of other solutions, for example using 2 UR robots instead of 1 industrial robot.

1. Innledning

I dette kapittelet presenteres bakgrunn, formålet med oppgaven, problemstillinger, avgrensninger og informasjon om bedriftene som den blir skrevet for.

1.1 Bakgrunn

LEVEL Power & Automation har en kunde som ønsker å automatisere sveise prosessen for containere som er laget i aluminium. Denne kunden er A. Olufsen Ship & Offshore AS og de produserer 2 typer PUSH-IN containere.



Figur 1. PUSH-IN containere fra A. Olufsen Ship & Offshore AS [7].

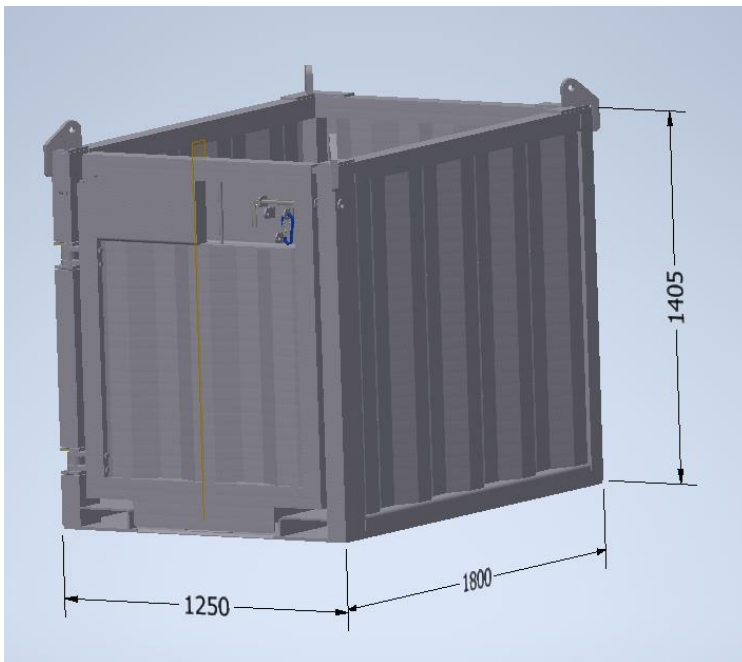
Den ene er med tak som har høyde 2200 mm og den andre er uten tak og med høyde 1430 mm [6]. I figur 1 kan man se den containeren som er på høyde 1430 mm. De ønsker å se på muligheten til å bruke robot til å sveise panelene til containeren slik at kun monteringen av containeren gjøres manuelt. Panelene til containeren vil si gulv, vegger og dør.

1.2 Formål og problemstilling

Formålet med oppgaven er å gjøre produksjon av containere mer effektivt. Det vil bli sett på muligheter for å automatisere denne prosessen ved hjelp av robot til å utføre sveiseoperasjonene og PLS til å styre den automatiske jigg. Løsningen vil bli presentert i form av simulering av sveiseoperasjonene for panelene, programkode for simuleringen og PLS kode for automatisert jigg. Det vil også bli forklart utføring av programvare test. Krav og problemstillinger vil bli identifisert. Forslag til robot som kan utføre jobben skal foreslås.

1.3 Avgrensninger

På grunn av tid, mengde arbeid og covid-19 (Korona viruset), har det blitt tatt avgrensninger av oppgaven. Automatisert jigg blir ikke lagd. Løsning blir ikke satt i produksjon denne gang da dette bare er et forslag til løsning av problemstillingen. Simulering og programkode blir laget for panelene til den ene containeren i aluminium. Programvare og sveiseparametrene ble ikke testet pga. covid-19.

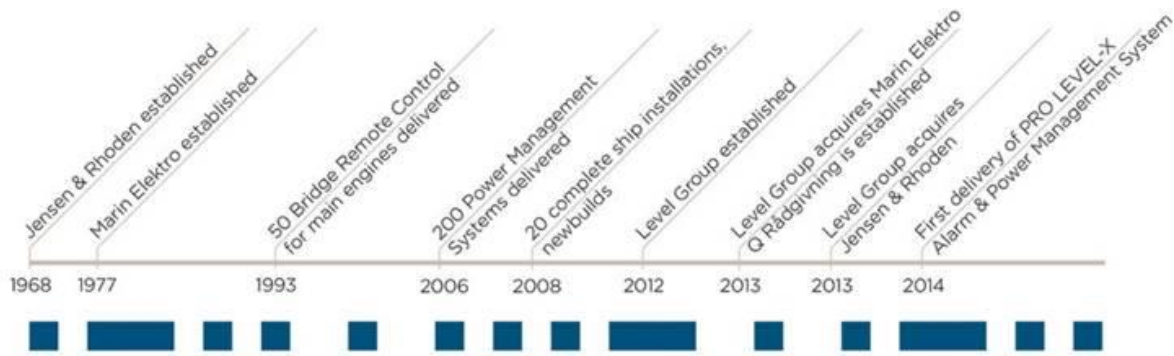


Figur 2. Containeren som er uten tak. CAD-tegning fra A. Olufsen Ship & Offshore AS.

Simuleringene er laget for den ene containeren som vises i figur 2. Det blir tatt hensyn til å velge robot som har rekkevidde stor nok til å kunne brukes på begge containerne. Containeren blir satt sammen manuelt og ikke med robot.

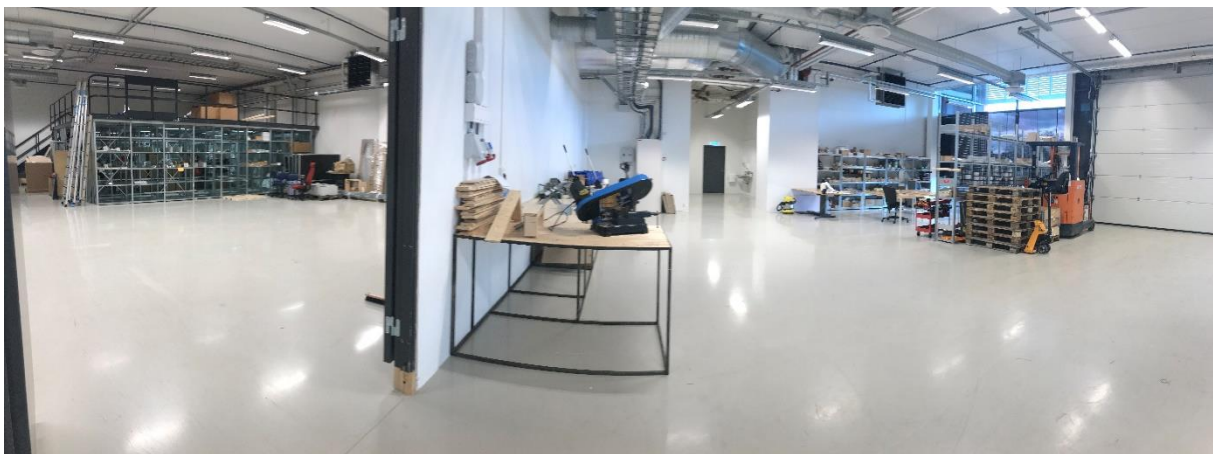
1.4 LEVEL Power & Automation

LEVEL Power & Automation er en del av LEVEL Group AS og startet firmaet for elektriske installasjoner og tjenester i 1977. De tilbyr verdiskapende ingeniørtjenester, teknisk rådgivning, prosjektstyring og fullt integrerte løsninger innen automatisering og elektriske installasjoner [1]. LEVEL Power & Automation spesialiserer seg på høyteknologiske automatiske systemer og leverer komplette løsninger for mekaniske, hydrauliske, pneumatiske og elektriske systemer [2].



Figur 3. Tidslinje som viser viktige hendelser for LEVEL Group AS [1].

I figur 3 vises en tidslinje som viser diverse viktige hendelser for LEVEL Group AS.



Figur 4. Verkstedet til LEVEL Power & Automation, Haugesund. Bilde tatt 28.04.2020.

I figur 4 ser man litt av verkstedet til LEVEL Power & Automation.

1.5 A. Olufsen Ship & Offshore AS

Firmaet startet opp i 1977 som et privat foretak. I dag er de lokalisert med egen kai i Haugesund hvor de produserer skreddersydde konstruksjoner for subsea- og offshoremarkedene. De er offshore-relatert og har også operasjoner i utlandet for stålkonstruksjoner, mekanisk arbeid og rørarbeid [3]. De har ulike tjenester som produktutvikling, stålarbeid, sveising, maskin, motorarbeid, rørarbeid, mobilisering - demobilisering, skipsreparasjoner, kontroll og sertifisering av mekanisk løfteutstyr [4].



Figur 5. Verkstedsområdet til A. Olufsen Ship & Offshore AS [7].

I figur 5 kan man se verkstedsområdet deres. Verkstedsområdet er på 2000 kvadratmeter, utendørsområde på 2000 kvadratmeter og 5 dypvannskaier med 12 meters dybde [5].



Figur 6. PUSH-IN container fra A. Olufsen Ship & Offshore AS [7].

De produserer PUSH-IN containere som, i denne oppgaven, skal se på muligheten til å automatisere sveiseprosessen. I figur 6 ser man den største containeren. Disse containerne er designet slik at man kan kjøre jekketralen direkte inn i containeren og gjør derfor innlasting og utlasting enklere. Disse lages i solid aluminium, tåler opp til 2000 kg last og er produsert i Haugesund [6].

2. Metode sveising

I dette kapittelet vil det bli presentert teori og metode for å sveise aluminium, kvalitet av sveis, defekt sveis, gass ved sveising, hvordan sveisingen skal utføres av robot og fordeler ved bruk av robot til å utføre sveising.

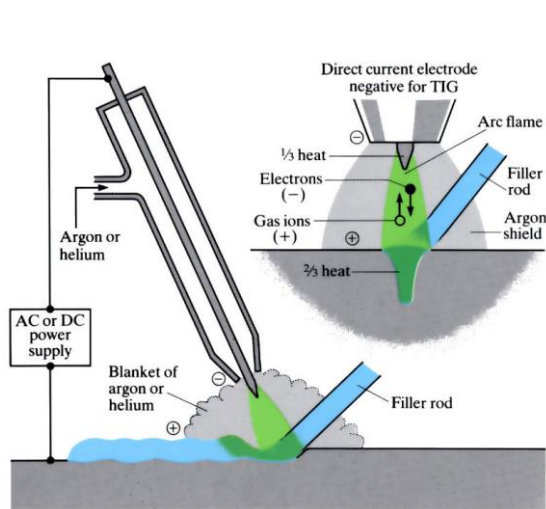
2.1 Teori og metode for aluminium

Containerne er laget av aluminium. Viktige faktorer ved sveising av aluminium er bredden av varmpåvirket sone og tiden temperaturen blir holdt i denne sonen. Det er ønskelig å holde disse faktorene så små som mulig ved sveising av aluminium. Man bør tilføre minst mulig varme på minst mulig areal for å sveise så kaldt som mulig [8].

Siden Aluminium har høy varmelednings-evne og varmelednings-koeffisient så kan det føre til deformasjon ved disse faktorene: godstykkelse, sveisens tverrsnitts areal, sveisens lengde, antall sveisestrenger, sveisehastighet, innspenningsforhold og sveisestilling. Det er derfor viktig å bruke minst mulig fuge, raskest mulig sveising, jevn varmetilførsel, bruk av jigg, planlagt sveiserekkefølge, ikke for høy temperatur og ved benyttelse av forvarming på tykke materialer må temperaturen være jevnt fordelt og ikke bli for høy [8].

Man bør velge en sveisemetode med konsentrert varmetilførsel og sveise så hurtig og kaldt som mulig for å unngå store krympekrefter som kan forårsake deformasjoner og sprekker. TIG- eller MIG-sveising er gode metoder fordi de gir en mer konstant varmetilførsel. MIG og TIG metodene gir konsentrert effekt og god oppbrytning av oksid belegget. Varmepåvirkningen blir da minst mulig [8]. I dag bruker sveiserne hos A. Olufsen Ship & Offshore AS MIG rundt rammen og TIG rundt de korrugerte platene (refererer til befaring hos A. Olufsen Ship & Offshore AS, 27.01.2020).

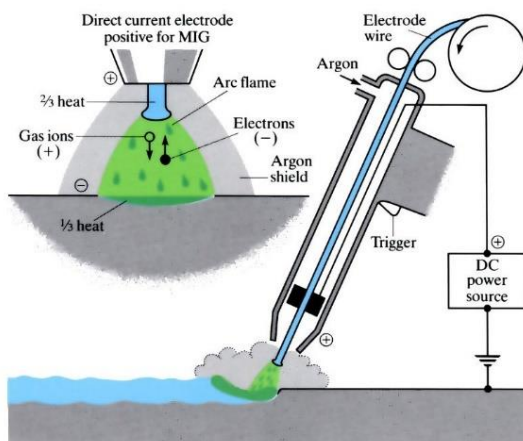
2.1.1 TIG-metode



Figur 7. Illustrasjon av TIG-metode [9].

I figur 7 viser en illustrasjon av hvordan TIG-metode foregår. TIG brukes ofte på områder hvor det er viktig med en pen sveis. Sveisepistolen føres mot komponent og tilsettmaterialet i andre hånden føres inn i smeltebadet. Sveisingen utføres med en skyvende bevegelse av brenneren samtidig som at tråden mates inn i sveisen [10]. Man må ha tilsettmateriale her fordi elektroden er av wolfram og har smeltepunkt på 3422 grader [11]. Metoden er mer krevende fordi man må holde flere elementer under kontroll og trenger 2 hender [10].

2.1.2 MIG-metode



Figur 8. Illustrasjon av MIG-metode [9].

Ved MIG-metode skal elektroden smelte og bli til tilsettmateriale [8]. Buen brenner mellom komponent og en kontinuerlig matet smeltende elektrode, som vist i figur 8, og derfor trenger man ikke 2 hender her. Velger denne metoden da det er lettere å få en robot til å håndtere denne metoden enn ved TIG-metode hvor man trenger "2 hender".

2.2 Defekt Sveis

Det stilles store krav til renslighet ved sveising av aluminium pga. hydrogenporer som kommer av hydrogen som løses i smeltebadet som ikke får tid til å slippe ut igjen ved avkjøling og størkning av sveiseforbindelsen. Disse lager derfor gassblærer og porer i sveisen. Det er viktig å rengjøre og eventuelt forvarme eller tørke for å hindre hydrogen. Feil ved sveising skyldes ofte de fysiske egenskapene og varmepåvirkning, porer, slagg, bindefeil og varmesprekker. Bindefeil kan komme ved dannelse av oksid på overflaten. Slagg kan komme av oksid eller forurensninger. Om det er feil med utstyret kan også bindefeil forekomme. Det er viktig med rengjøring og fjerning av oksid belegget. Porer og slagg kan oppdages ved bruk av røntgen- eller ultralydkontroll. ved ultralydkontroll kan man også oppdage bindefeil. Ved sveising med TIG- eller MIG-metode benyttes edelgasser som holder luftens oksygen borte [8]. Sveisedefekter skyldes ofte feil teknikk, parametere, dårlig beskyttelsesgassdekning eller feil kjørehastighet [68].

Panelene bør ligge horisontalt pga. aluminium renner og kan derfor bli problematisk eller vanskeligere å sveise ovenfra eller vertikalt. Ved å la panelene ligge «flatpakket» vil mest mulig av arbeidet være horisontalt (refererer til befaring hos A. Olufsen Ship & Offshore AS, 27.01.2020).

2.3 Kvalitet sveis

Sveisestyrke og tilhørende ytelse kan variere på størrelse til sveisen. Om den er underdimensjonert kan det hende at sveisen ikke tåler påkjenningene som den kan bli utsatt for. Plassering av avbrudd på sveisen kan også føre til redusert sveisestyrke. Sveisekoder og sveisestandarder gir nivåer på hva som er akseptabelt av antall sveiseavbrudd. Inspeksjon av sveisingen gjøres for å avgjøre om sveisen har god nok kvalitet. Inspeksjonen krever at man har kunnskap og forståelse av sveisetegninger, sveisesymboler, design av sveiseskjøter, sveiseprosedyrer, krav til koder og standarder, samt inspeksjons og testeteknikker. Man kan ta kurs i dette [22].

Undersøke områder metoder (fra kilde [22]):

- Inspeksjon og testing av kvalifisert sveisemetode
- Visuell inspeksjon
- Avsløring av overflatesprekker
- Radiografisk inspeksjon og ultralydinspeksjon av sveis
- Ødeleggende testing av sveis

De tre vanligste metodene er visuell sveisinspeksjon, testing av væskefargestoff penetrant, radiografisk testing eller røntgen [79]. Visuell inspeksjonsmetode baserer seg på vurdering av sveis kvalitet basert på det inspektøren kan se. Penetrant overflatekontroll detekterer sprekker, kantsår og porer. Radiograf detekterer innvendige, volumetriske feil som slagg, porer, bindefeil og sprekker. Ultralyd detekterer innvendige feil, som slagg, bindefeil, porer, rotfeil og sprekker [80]. Origo Test & Inspection AS er de som tester sveisingen fra A. Olufsen Ship & Offshore AS. De bruker stort sett penetrant og visuell sveiseinspeksjon. De bruker sjeldent røntgen på containerne og ultralyd bruker de på godstykkelse over 10 mm. (refererer til telefonsamtale Origo Test & Inspection AS, 30.04.2020).

2.4 Gass

Ved MIG-metode bruker man en edelgass som for eksempel Argon. Oksygen kan ikke brukes da det er en reaktiv gass og ikke edelgass (*inert gas*) [12]. Gassene som brukes ved lysbuesveising av aluminium er argon og helium og kan brukes både rent, og som blandinger. Argon er den mest populære, og nest vanlige er å bruke blandinger av argon og helium. Effekten av disse har noe å si for sveiseoperasjonen og kan påvirke formen på sveisemetallets tverrsnitt og sveisehastighet. Ved økt sveisehastighet kan man oppnå besparelser. For en gitt bue lengde vil tilsetning av helium til rent argon øke bue spenningen med 2-3 volt. Rent argon vil gi en sveis med et lysere og blankere overflateutseende. En sveis med blanding av helium og argon vil kreve børstning etter sveisetråd for å oppnå et lignende overflateutseende. Rent argon vil gi god lysbue-stabilitet og forbedret rengjøring. Blandingene pleier å være 25% helium og 75% argon. Større mengde helium enn 25% har en tendens til å produsere ustabilitet under AC-lysbuen. Heliumgassene er dyrere, har lavere tetthet og høyere strømningshastigheter. Det er mulig å øke sveisehastigheten i noen tilfeller ved å bruke helium og argon blandinger som derfor utligner kostnadene fordi produksjonen blir mer produktiv [13]. A. Olufsen Ship & Offshore AS bruker argon (refererer til telefonsamtale A. Olufsen Ship & Offshore AS, 30.04.2020).

2.5 Sveise parametere

Her blir det presentert metode for valg av strøm, spenning og hastighet ved sveising. Mye av sveisekunnskapen er basert på sveiserens individuelle erfaringer og basert på lyd og utseendet til buen og sveisebassenget for å lage en god sveis. Forskjellig individer har forskjellige buehastigheter, høyder og matehastigheter. Derfor er det ingen rett frem måte å gå fra manuell sveising til robotisert sveising. Siden de forskjellige parametrene gitt i lærebøker og andre kilder vanligvis er retningslinjer, må et sett med eksperimenter utføres. En god sveis er et resultat av riktig forhold mellom strøm, lysbuehastighet og matehastighet [70].

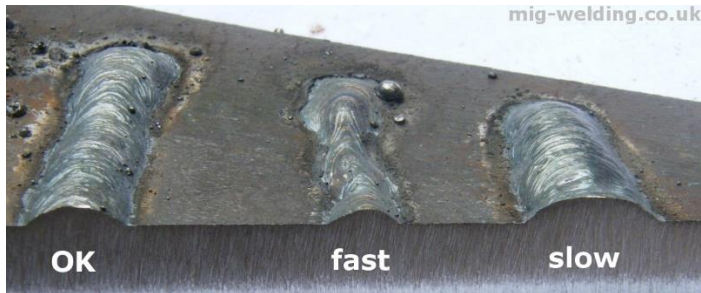
2.5.1 Avstand og hastighet



Figur 9. Resultater av for lang, OK og kort avstand ved sveising [14].

Avstand fra komponent til sveiseelektroden påvirker mengden strøm og varme som kommer inn i skjøten. Med for stor avstand kan det forekomme lite varme og mye sprut, og med for liten avstand kan varmen og strømmen bli for høy. Resultatene kan man se i figur 9. Ved MIG sveising holdes elektroden lenger borte fra skjøten enn ved pinnesveising pga. lysbuen er mer

konsentrert. Vinkelen holdes 90 grader på skjøten bortsett fra ved T-ledd hvor vinkelen varierer mellom 30 til 50 graders vinkel [15].



Figur 10. Resultater ved OK, rask og treig hastighet ved sveising [14].

Hastigheten er også en viktig faktor som man kan se i figur 10. For stor hastighet vil gi en sveisestørrelse som er for liten og gir utilstrekkelig gjennomtrenging. For liten hastighet vil gi feit sveis og sannsynligvis for mye varme som går inn i arbeidsplatene [15].

Anbefaling av sveiseparametre vil være å ta utgangspunkt i kapittel 2.5.3, med en hastighet på ca. 2,4 mm/s, og dermed teste dette. Med dette utgangspunktet kan man justere strøm, spenning og avstand til man får ønsket resultat. Eventuelt justere hastigheten i programmet.

2.5.2 Strømstyrke og spenning



Figur 11. Resultater av for lav, høy og OK strømstyrke ved sveising [14].

I figur 11 viser resultat av for lav, høy og OK strøm styrke. Sveisestrøm er den elektriske strømstyrken i kraftsystemet når sveisen blir laget. I MIG-metoden er sveisestrømmen direkte relatert til trådmatehastighet. En økning i trådmatehastigheten vil føre til en økning av strømmen [16]. Materialtykkelsen bestemmer strømstyrken. For eksempel krever 0.001 tomme av materialtykkelse 1 amp utgang. Ledningsstørrelse velges etter strømstyrke og bør velge en for de mest brukte tykkelsene. For stor strømstyrke kan føre til dårlig bue oppstart, for bred sveis, gjennombrenning og mye sprut. For liten strømstyrke gir en liten sveis som gir dårlig gjennomtrenging [17].

Spenningen bestemmer høyden og bredden. For mye spenning gir dårlig lysbue-kontroll og inkonsekvent gjennomtrenging. Hvis spenningen er for lav, gir det dårlig lysbueoppstart og for mye sprut [17].

2.5.3 Parameter for container panel

Containerene er i utgangspunktet EN AW 6082-T6 og de korrugerte/knekkede platene er EN AW 5754 H111. EN AW-6082 (Legering) T6 (Hardhetsgrad) har smeltepunkt på mellom 575-650 grader [18]. EN AW 5754 (legering) H111 (hardhetsgrad) har smeltepunkt på 595-645 grader [19].

Eksempel tykkelse på det som skal sveises:

- De korrugerte platene har en tykkelse på 3 mm.
- Rammene har en tykkelse på ca. 7 mm rundt det som sveises da den er hul innvendig. (Ikke alle stedene har denne tykkelsen)

Når vi vet tykkelse og smeltepunkt til komponentene er det lettere å finne avstand, hastighet og strømstyrke.

For å finne sveisehastighet kan formel på side 883 i *Manufacturing Engineering and Technology, Seventh Edition* av Serope Kalpakjian og Steven R. Schmid [20], brukes som vist i formel 1:

Formel 1. Sveisehastighet

$$v = \frac{eVI}{uA}$$

Da trenger man å velge spenning og strøm. For tykkelse på 7 mm kan man bruke 25 V og 210 A [21]. Verdiene ble sammenlignet med hva de bruker hos A. Olufsen Ship & Offshore AS og Marine Aluminium AS (refererer til telefonsamtale A. Olufsen Ship & Offshore AS og Marine Aluminium AS, 27.04.2020) med kilde [21]. For aluminium er spesifikk energi, u , 2.9 J/mm^3 ifølge tabell 30.2 side 883 i *Manufacturing Engineering and Technology, Seventh Edition* av Serope Kalpakjian og Steven R. Schmid [20]. e er effektiviteten av prosessen og velger her 0.75 som brukt i eksempel i side 883 i *Manufacturing Engineering and Technology, Seventh Edition* av Serope Kalpakjian og Steven R. Schmid. Denne varierer fra rundt 75% til 90% [20]. Regner man dette ut med disse verdiene og har en sveiselengde på 80 mm kan man se resultatet i formel 2:

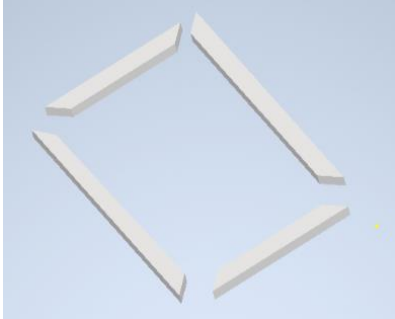
Formel 2. Sveisehastighet med verdier

$$v = \frac{0.75(25 \text{ V})(210 \text{ A})}{(2.9 \text{ J/mm}^3)(7 \text{ mm})(80 \text{ mm})} = 2.4 \frac{\text{mm}}{\text{s}}$$

Sveise parameterne bør testes for å finne rette verdier til en god sveise kvalitet.

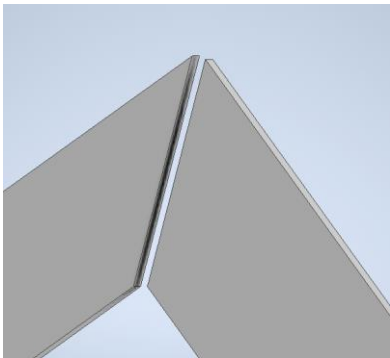
2.6 Sveise operasjoner container panel

Sveise operasjoner som skal bli utført av roboten vil bli forklart her. Først settes delene til rammen sammen.



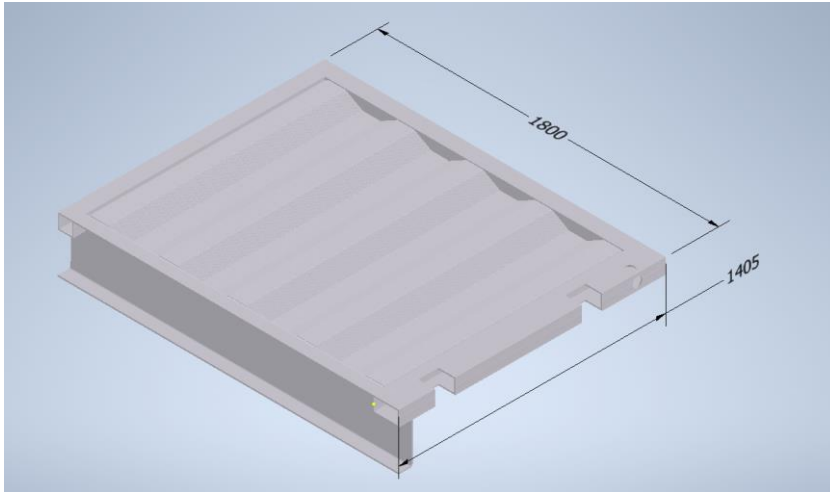
Figur 12. Illustrasjon av delene til rammen, bilde tatt i Autodesk Inventor 2020, 15.02.2020.

Figur 12 viser illustrasjon av delene som settes sammen til rammen som først skal sveises sammen. De plasseres i jiggen og festes godt slik at ikke deformasjoner skal oppstå.



Figur 13. Illustrasjon nærbilde av hvor roboten må sveise rundt 2 ganger. Bilde tatt i Autodesk Inventor 2020, 15.02.2020.

Rammen sveises 2 ganger rundt glipen som man kan se i figur 13. Grunnen til at man må sveise rundt 2 ganger er for å få nok sveise tykkelse inni da den er hul og krever større styrke. Rammen er tykkere og krever mer strøm enn de tynnere korrugerte platene. Man kan bruke MIG-metode over alt ved å justere strømstyrken (refererer til befaring hos A. Olufsen Ship & Offshore AS, 27.01.2020). I kapittel 3 vil metodene for robot sveiseprogram bli forklart.



Figur 14. Mål av rammen som det er blitt simulert program for. CAD tegning tilsendt av A. Olufsen Ship & Offshore AS.

Figur 14 viser hvordan panelet vil bli seende ut til slutt når de korrugerte platene sveises sammen med rammen.

2.7 Sveising med robot

Robot kan gi betydelig økning i produktivitet ved at den beveger seg raskere og behandler komponentene raskere enn mennesker. Når man skal sveise manuelt kreves det at man har høyt ferdighetsnivå med god konsentrasjon for å oppnå gode resultater. Det er ikke alltid man klarer å ha god nok konsentrasjon og dermed kan det føre til omarbeiding. Robot kan utføre nøyaktig samme sveiseoperasjon kontinuerlig hele tiden. Dermed oppnår man god kvalitet hver gang ved å programmere riktig sveisevinkel, hastighet og avstand. Dette vil redusere behovet for kostbar omarbeiding. Risiko ved sveiseoperasjoner er elektrisk støt, eksponering for gasser, blitsforbrenning og sveiseblink. Ved bruk av robot kan man redusere antall ulykker for arbeidere mot skader og risikoer som kan skje ved sveiseoperasjoner [23].

3. Metode robot og PLS

I dette kapittelet vil det bli presentert metode for programmering av sveiseoperasjonene, programvare RoboDK, simulering og programkode strategi, forklaring test av programvare, metode for PLS styring av rotering panel, robot forslag og vedlikehold av robot.

3.1 Introduksjon Robot

En robot er en maskin som kan ved hjelp av programvare og sensorer iverksette handlinger og kan ta over arbeid for mennesker som kan være farlig og helseskadelig [26]. På 1950-tallet ble den første industriroboten opprettet av George C. Devol. Denne roboten ble kalt "Unimate". Han lyktes ikke å selge produktet sitt i bransjen. Joseph Engleberger kjøpte robot patenten fra Devol på 1960-tallet og endret den til en industrirobot og dermed dannet selskapet "Unimation" for å produsere robotene. Han er kjent som "Faren til robotikk" [24].



Figur 15. Unimate, den første industriroboten [74].

Unimate, som vises i figur 15, var en hydraulisk manipulatorarm som kunne utføre repeterende oppgaver. Det ble brukt av bilprodusenter for å automatisere metallbearbeiding og sveiseprosesser [73]. 1960 til 2000 var en populær tid for industriroboter i bilindustrien og andre produksjonssektorer. International Robotics Federation estimerte at det var 742 500 industriroboter i drift i år 2000 [78].



Figur 16. UR robot fra Universal Robots [25].

I figur 16 kan man se en UR-robot fra Universal Robots som er en samarbeidsrobot. En samarbeidsrobot er en robot som er ment å fysisk samhandle eller kommunisere med mennesker i et delt arbeidsområde. Disse har forbedringer innen visuell teknologi, sensorer, programmering av kunstig intelligens og relativt mindre plassbehov [78]. Programvaren som ble brukt i prosjektet skulle bli testet på en UR5 robot på skolen, men på grunn av covid-19 så ble ikke dette mulig å gjøre. UR roboter har heller ikke lang nok rekkevidde til å utføre sveiseoperasjonene på så store paneler som i denne oppgaven, da det krever en lengde på over 2 meter og den lengste rekkevidden UR roboter har er 1300 mm [67].

3.2 Introduksjon PLS

PLS ble utviklet på 1960-tallet på grunn av forespørsel fra en amerikansk bilprodusent for å erstatte releer som tar mye plass. På 80-tallet ble størrelsen redusert. På 1990-tallet ble standarden "IEC 1131-3" forsøkt slått sammen programmeringsspråkene til PLS gjennom én internasjonal standard [28].



Figur 17. Omron PLS [27].

En PLS er en industriell datamaskin som brukes til å overvåke innganger. I figur 17 viser en PLS fra produsenten Omron. Den tar beslutninger basert på deres tilstander og kontrollerer utgangene. Vanlige bruk for PLS er ved å kontrollere prosesser og utstyr i anlegg [29].

3.3 Programmering

De 3 mest brukte metodene for å programmere industri roboter er kinetiq, håndholdt kontroll og *off-line* programmering (fra kilde [30]):

- Kinetiq undervisnings metode er en metode for å programmere robot hvor en flytter robotleddene manuelt og lagrer posisjonene med en håndholdt kontroll.
- Håndholdt kontroll er en kontroller som brukes til å styre roboten og lagre posisjonene til roboten med. Man kan styre roboten med denne kontrolleren og lagre posisjonene. Ved denne metoden flytter man da ikke leddene manuelt, men styrer med kontrolleren.
- *Off-line* programmering (OLP) er en metode hvor man lager program ved bruk av simulering og tekstkode.

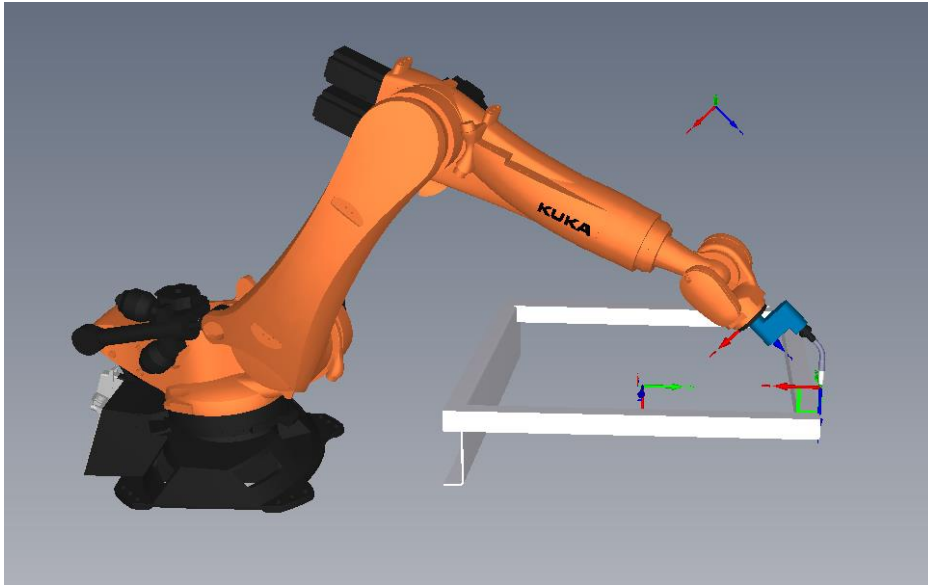
OLP er en programmeringsmetode for roboter der man programmerer først programmet og deretter laster opp programkoden til den virkelige roboten. I OLP er robotcellen representert gjennom en grafisk 3D-modell i en simulator [72].

OLP er best egnet for sveising, trimming, laserskjæring og maling. Simulering av robot operasjoner er en 3D-presentasjon av en robotcelle eller produksjonslinje som viser visuelt hvordan roboten beveger seg langs en bane med X-, Y-, og Z-koordinater. Ved å bruke offline programmering kan man forutse hvordan programmeringen vil bevege roboten og dermed unngå feil som kollisjon. Man kan også sette leddgrenser for å unngå at roboten beveger seg i områder den ikke skal jobbe. Ulike robotprodusenter har sine egne programmeringsspråk og med postprosessorer blir programmeringskoden omformet til et språk roboten kan forstå. Fordelen med *off-line* programmering er at man trenger kun stoppe produksjonen når programmet skal lastes opp til roboten. Når man da er klar for neste operasjon kan man bare bytte over programmet og så starte produksjonen igjen. Produksjonsavbrudd og driftsstans blir betydelig redusert. Det er viktig ved OLP at den virtuelle verden samsvarer med den virkelige for å få mest mulig nøyaktig simulering og programmering. Dette ved hjelp av kalibrerte data for robotkinematikk, eksterne akser og arbeidsstykkeposisjon. Det er viktig at CAD-modellen er så nøyaktig som mulig. Millimeter til centimeter kan gjøre at det ikke passer med programmeringen/simuleringen. Man kan bruke gjennomlysbue-sømsporing (TAST) for sanntidssporing av sveiseskjøten når CAD modellen og den faktiske delen ikke samsvarer. Lasersømsporer eller berøringsfølelse sensor kan også brukes [31].

RoboDK programvare for simulering og programmering ble valgt i denne oppgaven fordi det var mye informasjon om det på *Youtube* og nettsiden deres. På nettsiden deres har de dokumenter som forklarer hvordan man bruker programvaren. På *Youtube* har de over 100 videoer om RoboDK hvor de blant annet viser hvordan man kan bruke programvaren. De har også et forum på nettstedet deres som de aktivt hjelper folk som har spørsmål.

3.4 Programvare RoboDK

RoboDK v4.2.4 er en kraftfull simulator og programmering for industrielle roboter. Det ble grunnlagt i 2015 i Canada. Biblioteket til RoboDK har per i dag rundt 500 robotarmer å velge mellom [32].



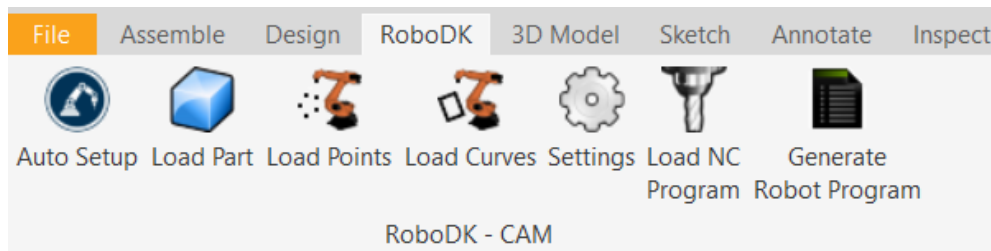
Figur 18. Kuka robot som sveise en del av rammen. Bilde tatt i RoboDK programvare, 15.03.2020.

I figur 18 viser et eksempel ved sveising av ramme. Man kan bruke RoboDK til mange forskjellige operasjoner (fra kilde [33]):

- Maskinering
- Sveising
- Skjæring
- Maling
- Inspeksjon
- Fresing
- Plukking og plassering
- Pakking og merking
- Robotkalibrering

Det er enklere å finne ut hvilken robot som passer best til oppgaven siden man kan teste operasjonen i programmet. RoboDK har også funksjoner for å unngå kollisjoner, men dette er bare en programvare sikkerhet og man bør også ha flere sikkerhets tiltak for å unngå at robot kolliderer med mennesker og gjenstander [33]. Mer om sikkerhet finnes i kapittel 4.

3.4.1 RoboDK Autodesk Inventor tilleggsprogram

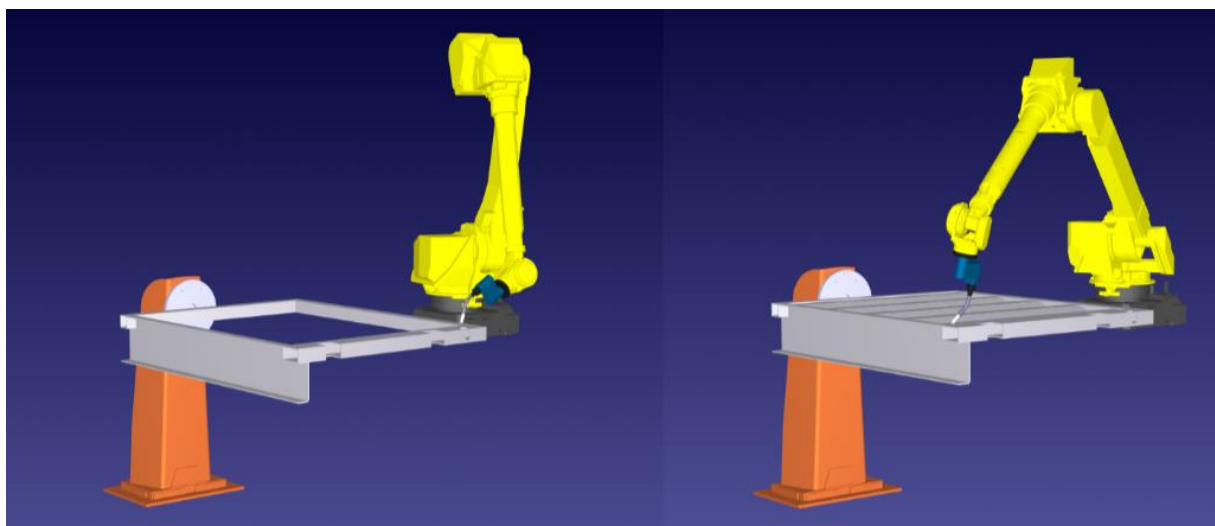


Figur 19. Fra Autodesk Inventor 2020, RoboDK tilleggset. Bilde tatt i Autodesk Inventor 2020, 25.03.2020.

RoboDK-tillegget for Autodesk Inventor brukes for å kombinere Autodesk Inventors 3D CAD modelleringsfunksjoner med RoboDK slik at det blir lettere å simulere og programmere nøyaktig. Dette tillegget finnes som vist i figur 19. For å aktivere denne funksjonen kan man trykke på *Tools* og så *Add-ins* i Autodesk Inventor. Hvis den ikke ligger der så må man laste ned tillegget manuelt [34]. Dette ble brukt for å overføre komponentene som skulle sveises fra Autodesk Inventor til RoboDK for å lage simulering og program.

3.4.2 Simuleringer og programkode robot strategi

Etter at rammen er sveiset sammen, så settes de korrugerte platene inni og da må man skifte til program for korrugerte plater. Dermed ble det 2 programmer på denne veggen.

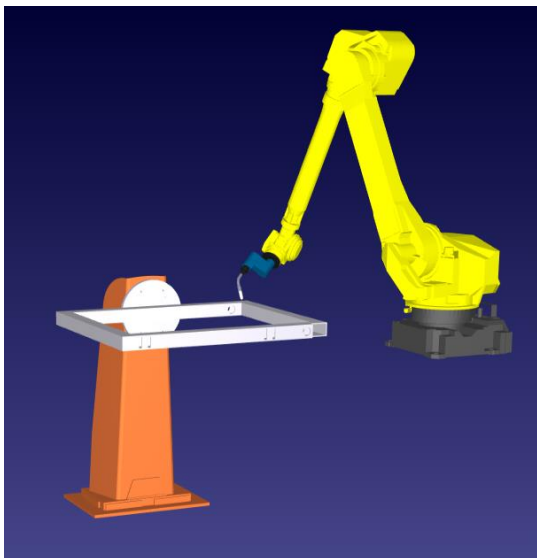


Figur 20. Simulering av sveising ramme til venstre og simulering av sveising ramme med korrugerte plater til høyre. Bilde tatt av simulering-film som ble lagd fra RoboDK programvare, 30.04.2020.

I figur 20 kan man se rammen som blir sveiset til venstre og ramme med korrugerte plater som blir sveiset til høyre.

Liste over alle simuleringene for sveiseoperasjoner som ble laget i RoboDK:

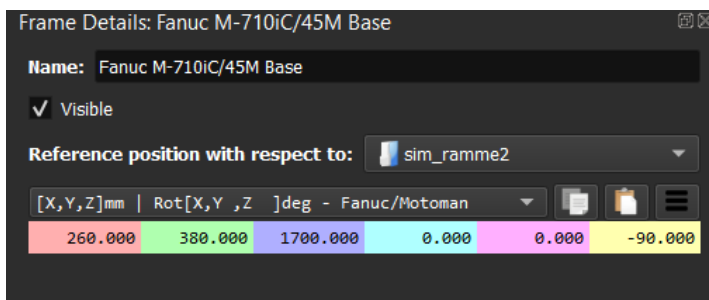
- Simulering av sveiseoperasjon gulv
- Simulering av sveiseoperasjon høyre vegg, ramme
- Simulering av sveiseoperasjon høyre vegg, med korrugerte plater
- Simulering av sveiseoperasjon venstre vegg, ramme
- Simulering av sveiseoperasjon venstre vegg, med korrugerte plater
- Simulering av sveiseoperasjon bakside vegg
- Simulering av sveiseoperasjon dør, ramme
- Simulering av sveiseoperasjon dør, med innhold



Figur 21. Bilde av simulering sveiseoperasjon dør ramme, bilde tatt 04.05.2020.

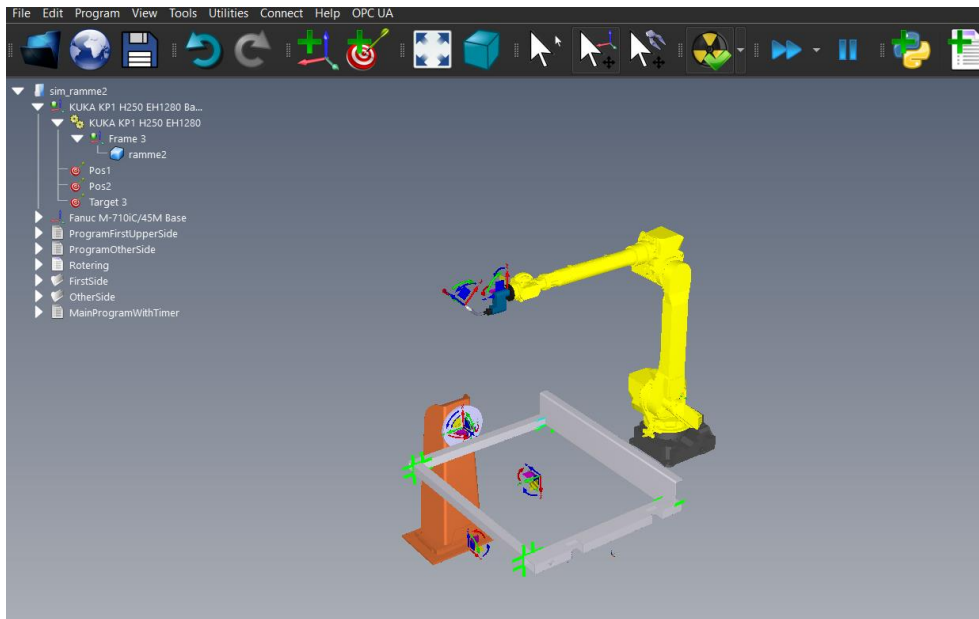
Kodene for alle simuleringene totalt ble på ca. 480 sider og derfor ble kun testkodene og kode for 1 simulering lagt til som vedlegg slik at rapporten ikke blir veldig lang. Simulering av sveiseoperasjon dør ramme ser man bilde av i figur 21 og kode finnes i vedlegg 3.

For å bruke RoboDK ble dokumentene deres på robotk.com/doc brukt for å lære bruken av programvaren. For å lage et slikt program i RoboDK så kan man laste opp en robot fra biblioteket, laste opp CAD tegning av komponent til RoboDK som ønskes å sveises, laste opp sveisepistol og plassere denne på roboten. Her ble det også brukt en KUKA KP1-H enkeltakseposisjoner til å rotere panelene. Disse må da plasseres i RoboDK til ønsket posisjon.



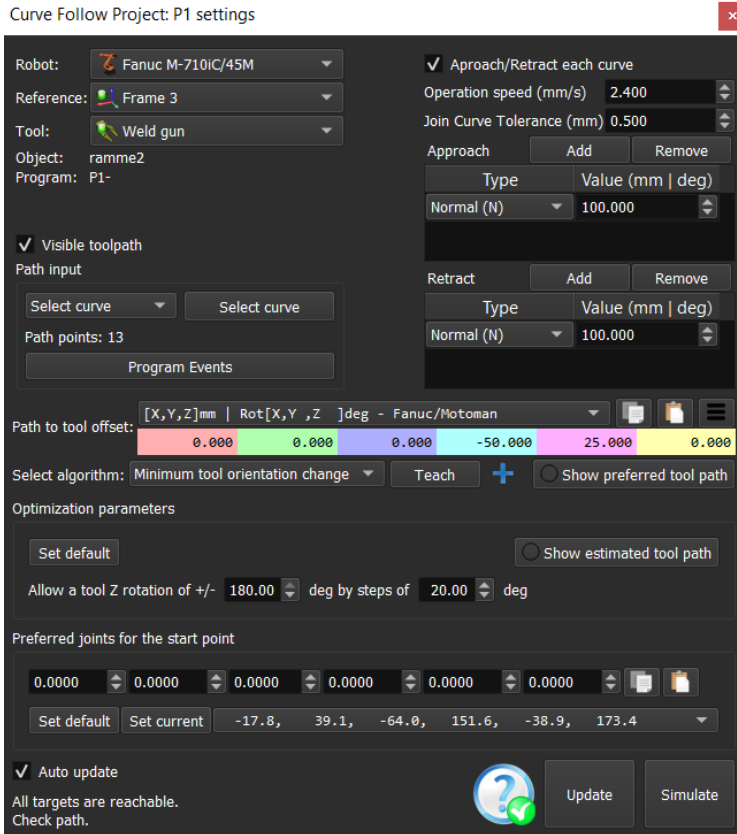
Figur 22. plassering av robot i forhold til komponent. Bilde tatt i RoboDK programvare, 30.04.2020.

I figur 22 kan man se verdier på hvor roboten er plassert i forhold til komponent, altså ramme 2 (venstre vegg).



Figur 23. Viser skjermbilde fra programmet som ble laget. Bilde tatt i RoboDK programvare, 30.04.2020.

I figur 23 kan man se programtreet til venstre i RoboDK. For å lagre posisjoner til roboten kan man bevege roboten til ønsket posisjon og trykke på *Add new target*. Alle disse kan finnes i program treet. Man kan markere alle bevegelsene som ble laget og trykke *Create new program* og dermed opprettes et program med disse bevegelsene for roboten. For å få roboten til å bevege sveisepistolen langs en linje/sti som skal sveises så kan man bruke *curve follow project* [81].



Figur 24. Curve follow project. Bilde tatt i RoboDK programvare, 30.04.2020.

I figur 24 kan man se *Curve follow project*. Her kan man trykke på *Select curve* for å velge linjen som skal sveises. Deretter kan man justere roboten slik at sveisepistolen beveger seg slik man ønsker ved å endre verdier i *Path to tool offset*. Deretter trykkes *Update* og *Simulate* for å se bevegelsen som ble laget. Dette blir da til et program. Alle disse linjene som skal sveises blir da mange småprogrammer som kan samles i et hoved program. I hovedprogrammet kan man legge til andre funksjoner som *pause*, *targets*, *call program*, osv. Når programmet er ferdig laget kan man høyre klikke på hovedprogrammet og velge *Generate robot program*. Da dukker det opp et vindu med kodene i som kan sendes til roboten. Eventuelt kan man trykke *Send program to robot* og programmet blir sendt til roboten [81].

```

1  |/PROG P1
2  |/ATTR
3  |OWNER   = MNEDITOR;
4  |COMMENT = "RoboDK sequence";
5  |PROG_SIZE = 0;
6  |CREATE   = DATE 31-12-14 TIME 12:00:00;
7  |MODIFIED = DATE 31-12-14 TIME 12:00:00;
8  |FILE_NAME = P1;
9  |VERSION  = 0;
10 |LINE_COUNT = 10;
11 |MEMORY_SIZE = 0;
12 |PROTECT  = READ_WRITE;
13 |TCD: STACK_SIZE = 0,
14 |    TASK_PRIORITY = 50,
15 |    TIME_SLICE = 0,
16 |    BUSY_LAMP_OFF = 0,
17 |    ABORT_REQUEST = 0,
18 |    PAUSE_REQUEST = 0;
19 |DEFAULT_GROUP = 1,*,*,*,*;
20 |CONTROL_CODE = 00000000 00000000;
21 |/MN
22 | 1: UFRAME_NUM=3 ;
23 | 2: ! UF3:1370.0,-570.0,-4 ;
24 | 3: ! 20.0,180.0,-0.0,90.0 ;
25 | 4: ! Show Weld gun ;
26 | 5:J P[1] 20% CNT1 ;
27 | 6:L P[2] 1000mm/sec CNT1 ;
28 | 7:L P[3] 2mm/sec CNT1 ;
29 | 8:L P[4] 2mm/sec CNT1 ;
30 | 9:L P[5] 2mm/sec CNT1 ;
31 |10:L P[6] 1000mm/sec CNT1 ;
32 |/POS
33 |P[1]{
34 |  GP1:
35 |  | UF : 3, UT : 9,
36 |  | J1=  -77.692 deg, J2=  60.639 deg, J3=  -78.932 deg,
37 |  | J4=  -65.102 deg, J5= -115.987 deg, J6= -123.235 deg
38 | };
39 |P[2]{
40 |  GP1:
41 |  | UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',
42 |  | X = -526.025 mm, Y = -652.060 mm, Z = 60.000 mm,
43 |  | W = -120.353 deg, P = -0.004 deg, R = -90.706 deg

```

Figur 25. Kode fra simulering. Bilde tatt i RoboDK programvare, 30.04.2020.

I figur 25 kan man se litt av programmet som ble vist i *curve follow project* P1. P1 vil si program 1 som ble laget med *curve follow project*.

```

46  | 25: CALL P1 ;
47  | 26: CALL P2 ;
48  | 27: CALL P3 ;
49  | 28: CALL P4 ;

```

Figur 26. Litt av hovedprogrammet. Bilde tatt i RoboDK programvare, 30.04.2020.

I figur 26 kan man se at programmene (P1, P2, osv) blir kalt opp i hovedprogrammet. Navn på programmer er valgfritt og kan endres. Til slutt for å lage simuleringen trykker man på programmet som er blitt laget og velger *export*.

3.5 Test programvare

For å teste programvaren RoboDK kan man lage enkle programmer og overføre dem til for eksempel en UR5 robot og sjekke at simulering og virkelighet samsvarer.

3.5.1 Risikovurdering

Før testing så må man ta en risikovurdering som vist i tabell 1:

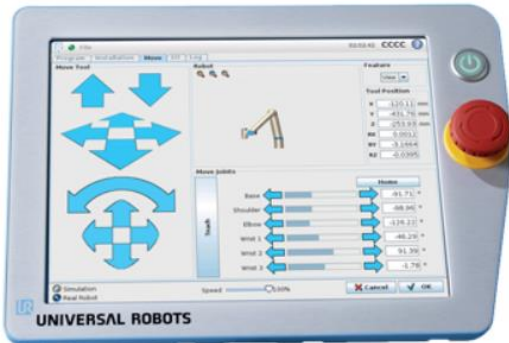
Tabell 1. Risikovurdering laget 25.03.2020, [35].

Nr.	Uønsket hendelse	Årsak	Konsekvens	S	K	R	Risikoreduserende tiltak
1	Fingre i klem	Person innenfor robotens arbeidsområde	Personskade	Middels	Høy	Høy	Rutiner, lese brukermanual og bruk av nødstop
2	Skade på hud	Skarpe kanter	Personskade	Lav	Middels	Middels	Rutiner og nødstop
3	Slag fra robot	Fartsgrense for høy eller ikke vært utenfor robotens arbeidsområde	Skade på person, eller utstyr	Middels	Høy	Høy	Sette fartsgrense, være utenfor robotens arbeidsområde og bruk av nødstop
4	Robot eller verktøy faller	Dårlig festet	Skade på person eller utstyr	Lav	Høy	Høy	Feste med rett moment
5	Feil i innstillinger	Uautoriserte endringer	Skade på person eller utstyr	Middels	Middels	Middels	Passord på kontroll

Merk! Grad av sannsynlighet (S), konsekvens (K) og risikonivå (R).

3.5.2 Starte opp roboten

For å starte opp roboten utføres disse trinnene (Fra kilde [35]):

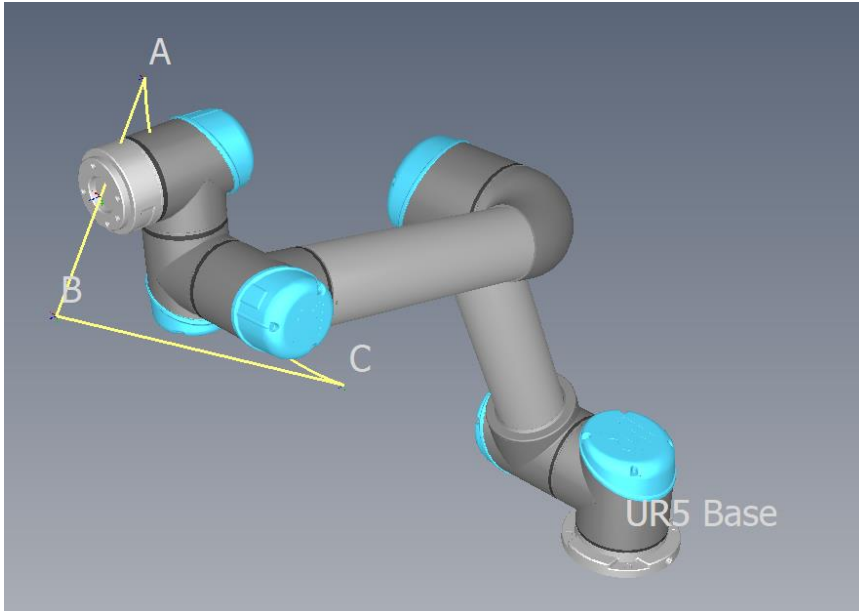


Figur 27. Håndholdt kontroll [76], fra Universal Robots.

- Trykk på nødstopknappen på fremsiden av den håndholdte kontrollen, se bilde 27.
- Trykk på av / på-knappen på kontrollen.
- Vent mens systemet starter, og til tekst vises på berøringsskjermen.
- Det vises en melding på berøringsskjermen når systemet er klart med angivelse om at roboten må initialiseres.
- Trykk på knappen i popup-dialogboksen for å bli ført til initialiseringen.
- Vent til *Confirmation of applied safety configuration* dialogboksen for sikkerhetskonfigurasjon og trykk *Confirm safety configuration*-knappen. Dette gjelder et sett med sikkerhetsparametere som må justeres.
- Lås opp nødstopknappen og dermed blir robottilstanden endret fra *Emergency stopped* til *power off*.
- Gå utenfor rekkevidden til roboten.
- Trykk på *On*-knappen på berøringsskjermen og vent noen sekunder til robottilstanden endres til inaktiv.
- Sørg for at nyttelastmassen og den valgte monteringen er riktig, ellers vil man bli varslet at oppdaget montering basert på sensor data ikke samsvarer med den valgte monteringen.
- Trykk på *Start*-knappen på berøringsskjermen og roboten vil lage en lyd og bevege seg litt mens man slipper bremsene.
- Trykk på *OK*-knappen for å komme til velkomstskjermen.

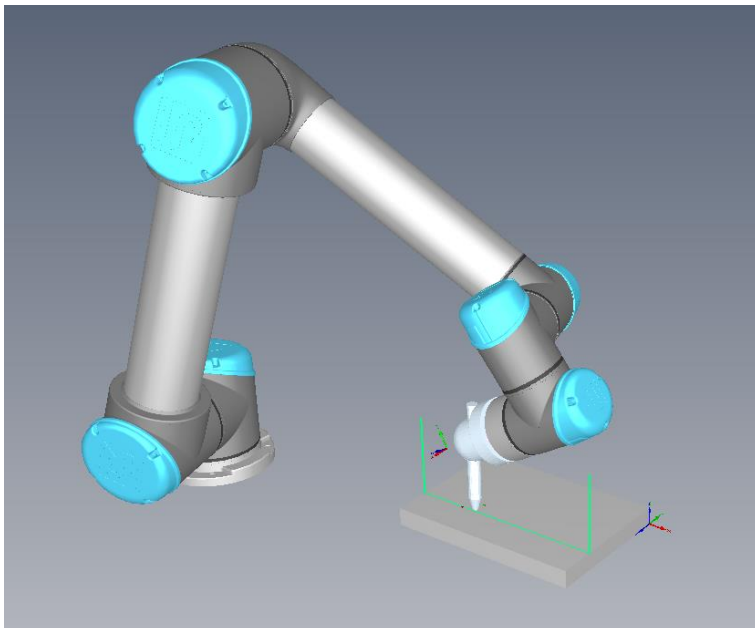
3.5.3 Forslag testing

Tester som kan bli utført kan for eksempel være bevegelse av robot for å sjekke at simulering og virkelighet beveger seg likt. Man kan også få robot til å utføre en oppgave for å sjekke hvor nøyaktig virkelighet og simulering er.



Figur 28. UR5 robot som beveger seg fra ulike posisjoner. Bilde tatt i RoboDK programvare, 05.03.2020.

Man kan lage et enkelt program hvor roboten beveger seg fra A til B til C, som vist i figur 28. Kode for denne operasjonen kan finnes i vedlegg 1.

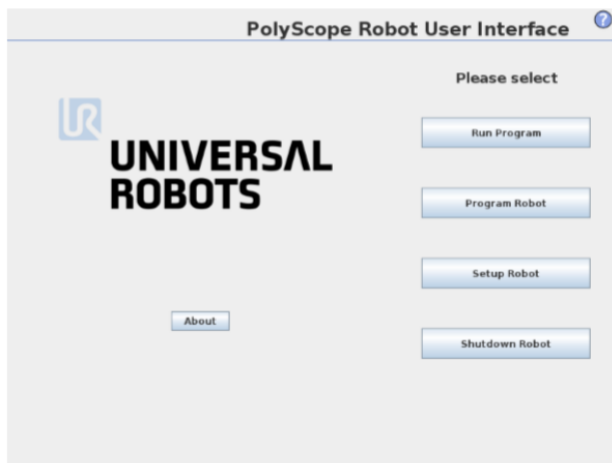


Figur 29. UR5 robot som beveger seg over CAD tegning. Bilde tatt i RoboDK programvare, 05.03.2020.

Man kan også feste en tusj eller penn til robot og få den til å tegne på for eksempel et ark/bok som vist i figur 29. Kode for denne operasjonen kan finnes i vedlegg 2.

3.5.4 Overføre program

Med RoboDK kan man generere SCRIPT- og URP-filer som kan kjøres av for eksempel en UR-robot. Man lagrer URP-filene på en minnepenn og kobler denne til robotens håndholdte kontroll. Man kan også utføre programmene på roboten fra RoboDK om roboten er koblet til datamaskinen. Man kan også importere SCRIPT-filene til simulatoren. For å overføre program kan man enten bruke USB eller FTP hvis roboten er koblet til datamaskinen. Man trenger IP-adressen til roboten for å sende over program. SCRIPT-programmet lagres i robotkontrolleren. PolyScope er det grafiske brukergrensesnittet til UR-roboten [36].

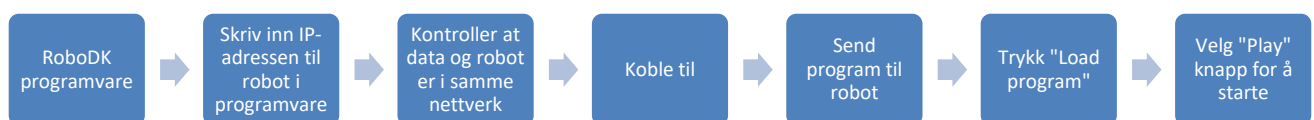


Figur 30. Bilde av skjermen på en UR5 kontroll boks [36].

For å overføre program til robot så må man vite IP-adressen. Trykk på *About* og der vil man finne IP-adressen. I figur 30 kan man se hvor man finner *About*. Forsikre at datamaskinens IP er i samme nettverk som roboten [36].

Følg disse trinnene for å koble til roboten fra RoboDK (fra kilde [36]):

- Høyreklikk på roboten i RoboDK
- Velg *Robots Universal Robots - Connect to robot*
- Angi robotens IP-adresse
- Velg *Connect*.
- Det skal være en grønn melding som viser *Ready* hvis tilkoblingen lykkes.



Figur 31. Prosessflyt av overføring programkode, laget 28.02.2020.

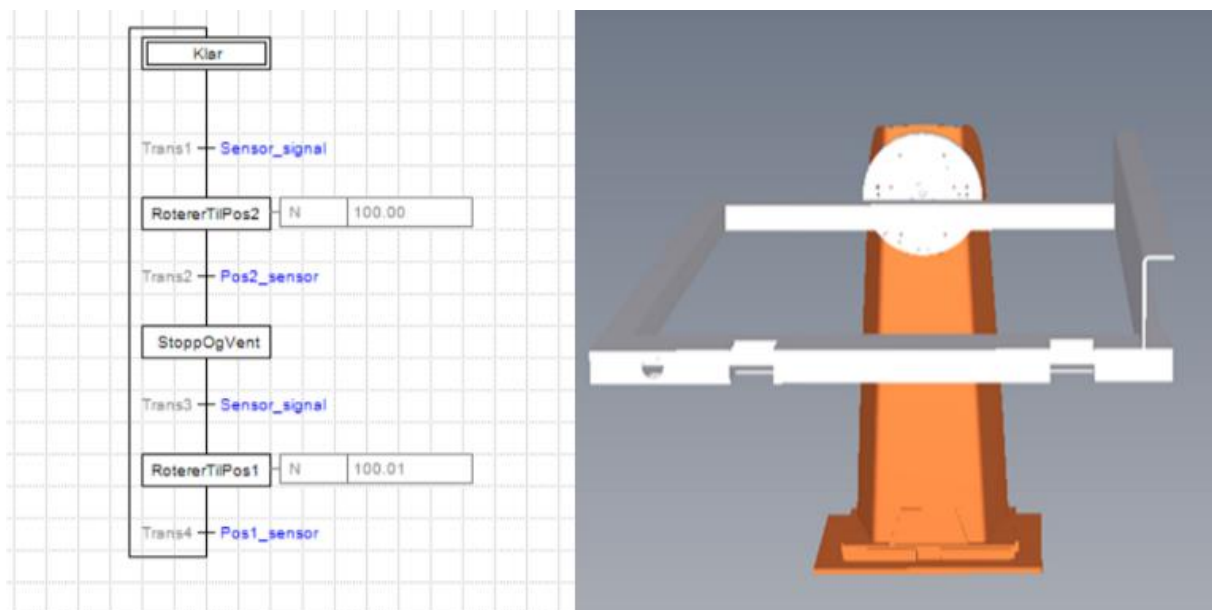
I figur 31 kan man se de ulike stegene for å overføre programmet fra programvare på datamaskinen til robot og starte programmet.

3.6 PLS styring av automatisert jigg

En PLS brukes mye i industriell produksjon hvor den tar informasjon fra ulike prosesser, analyserer dataene og deretter utfører oppgaver. For å programmere en PLS brukes det en datamaskin med et program som blir overført til PLS gjennom for eksempel en RS485 seriell tilkobling. Analoge signaler blir omdannet til digitale som blir behandlet av PLS. CPU i PLS er hjernen som behandler all informasjonen PLS mottar. Den lagrer viktig informasjon som trengs for at PLS skal kunne utføre sine oppgaver [37].

Tanken med PLS i denne oppgaven er å bruke den til å kommunisere med den automatiske jiggen slik at den roterer panelene. Med tanke på kapittel 4 bør man velge en PLS med sikkerhetsnivå PL=d.

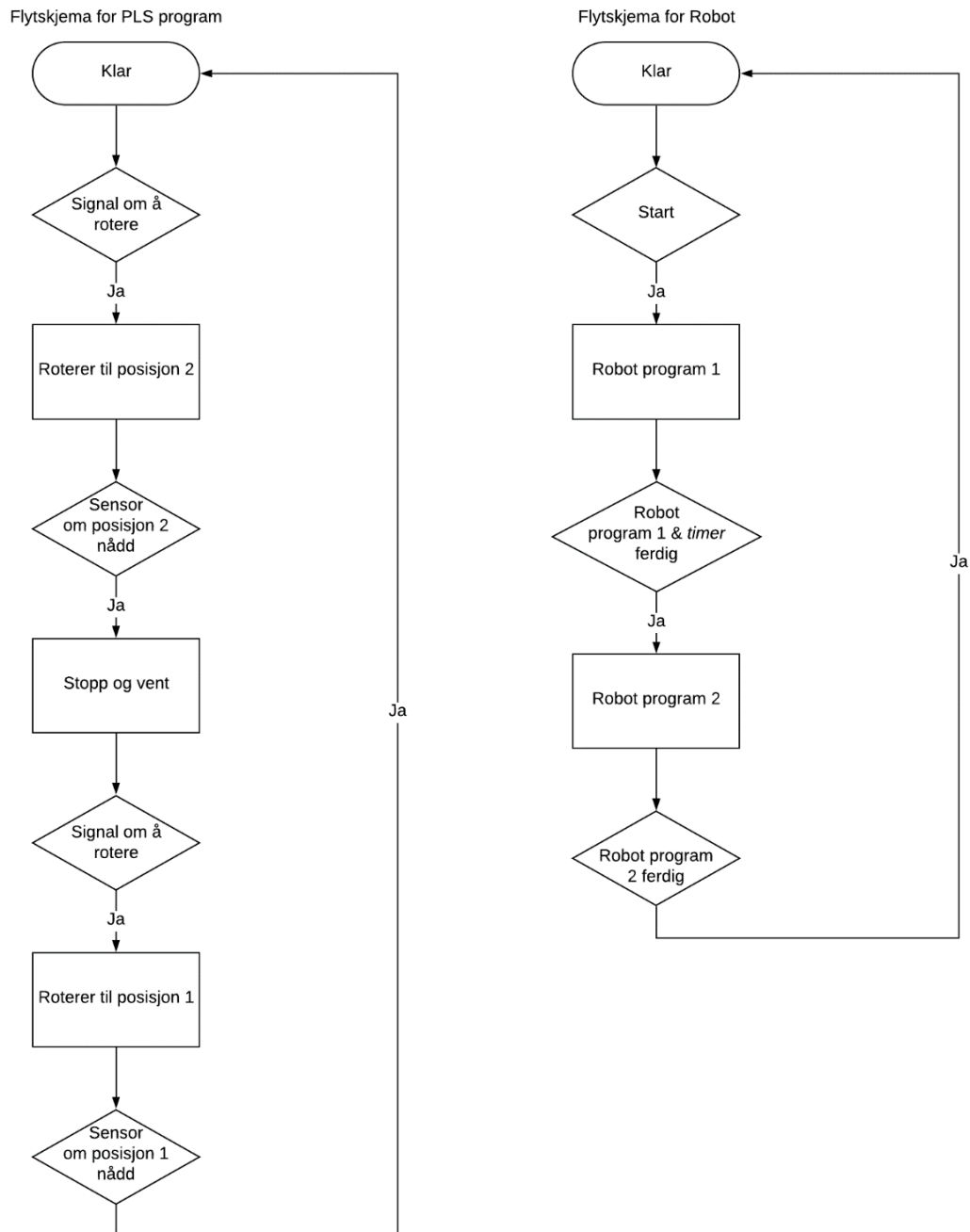
Sekvensielle funksjonskart er et programmeringsspråk for PLS som ble brukt i denne oppgaven. Det er et oversiktlig språk hvor man har trinn, transisjoner og aksjoner. Trinn er tilstandene i prosessen, transisjonene er de betingelsene som må være oppfylt for å gå over til neste trinn, aksjonene er knyttet til de enkelte trinnene og definerer instruksjonene som skal skje i det trinnet [38]. I programmet ser man blant annet bokstaven N. Det er en aksjon som utføres så lenge det tilhørende trinn er aktivt [39]. Her vil da adresse 100.00 være at motor går den ene veien mot posisjon 2 og 100.01 går den andre veien mot posisjon 1.



Figur 32. Programkode SFC for program til jigg til venstre, og rotering av panel til høyre ved bruk av KUKA KP1-H enkeltakseposisjoner. Bilde til venstre tatt i Omron programvare CX-programmer, 20.03.2020. Bilde til høyre i RoboDK, 02.05.2020.

Figur 32 viser programmet for PLS styringen. I første tilstand vil ikke motoren gå og den vil vente på signal om å rotere. Når den mottar signal, vil den gå til tilstand 2. I tilstand 2 vil da panelet bli rotert til den treffer posisjon 2 (den var i posisjon 1) og når en sensor detekterer det vil den gå over til tilstand 3 hvor den da skal stoppe for den er kommet til rett posisjon og venter på neste signal. Da signalet mottas igjen roterer den tilbake til posisjon 1. Her vil også en sensor detektere dette og stoppe motoren når posisjon 1 er nådd. Dermed er den tilbake i tilstand 1 hvor den da venter på signal.

3.6.1 Flytskjema Program



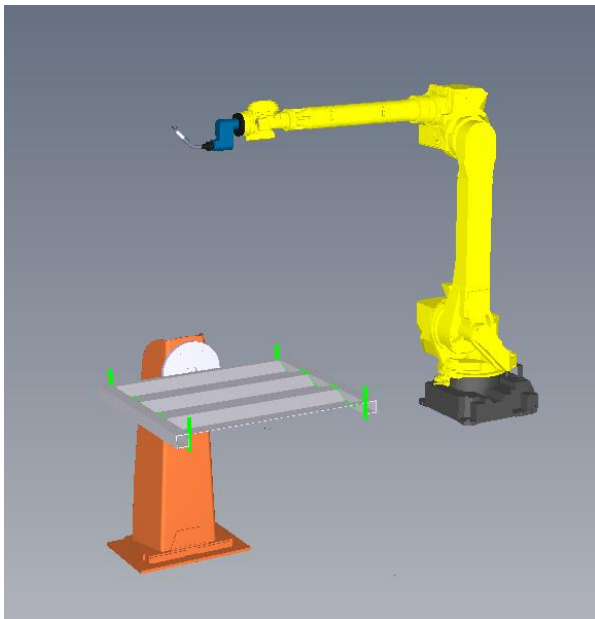
Figur 33. Viser flytskjema over program for PLS og program for robot, laget 01.04.2020.

Figur 33 viser program flyten for både PLS styring av rotasjon og Robot sveise program. Ved bruk av sensorer kan man få robot og PLS til å samarbeide. Robot utfører sveiseprogrammet for øvre del av panel. Når roboten er ferdig med sveiseprogrammet for den ene siden av panelet, går den til en sensor for å indikere at den er ferdig. Dette kan for eksempel være en sensor som blir høy når noe kommer i veien for den (se kapittel 5.6 for forslag sensor her). PLS mottar dette signalet og begynner å rotere panelet samtidig som robotens *timer* teller ned.

Timeren må stilles inn til minst den tiden det tar for rotering av panel. PLS roterer til den treffer en endestopper/sensor som signaliserer at den er kommet frem til rett posisjon. Da skal den vente helt til den får signal igjen om å rotere. Når robotens timer er ferdig så begynner den å sveise neste program for den siden som er øverst.

3.6.2 Rotering i simuleringen

For å rotere panelene i simuleringen ble det brukt en KUKA KP1-H enkeltakseposisjoner. Med denne kan man programmere likt som roboter blir programmert, altså med RoboDK.



Figur 34. KUKA KP1-H enkeltakseposisjoner sammen med robot og panel i programvare RoboDK. Bilde tatt i RoboDK programvare, 20.03.2020.

Figur 34 viser KUKA KP1-H enkeltakseposisjoner sammen med robot og panel som skal sveises.

Spesifikasjoner til KUKA KP1-H enkeltakseposisjoner (fra kilde [40]):

- Nyttelast: 250 kg
- Verktøy radius: 950 til 1250 mm
- Max høyde: 980 til 1280
- Repeterbarhet: ± 0.08 mm
- Rotasjon med energi tilførsel: ± 185 grader
- Rotasjon uten energi tilførsel: uendelig
- Rotasjonstid: 2.3 s / 3.7 s

Med denne rotasjonstiden så bør Roboten ha en *timer* på minst 4 sekund. Eventuelt mer for å være sikker på at det ikke skal skje en feil som for eksempel at robot begynner å sveise før jigg er skikkelig rotert (se kapittel 5.7 for annet forslag til løsning, i stedet for bruk av *timer*).

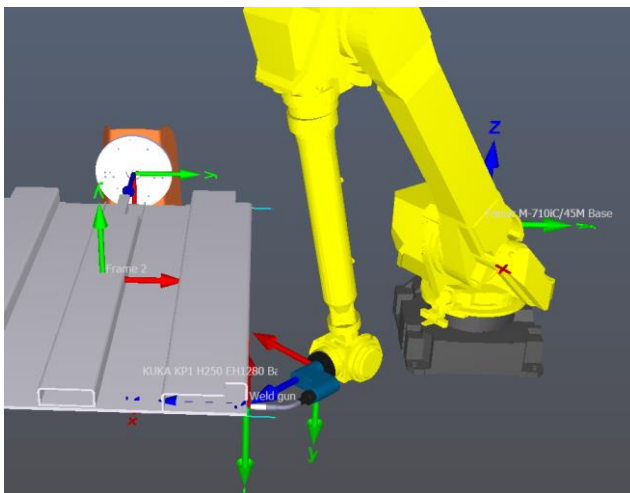
3.7 Robot forslag

Viktige faktorer ved valg av robot:

- Nyttelast
- Antall akser
- Rekkevidde
- Repeterbarhet
- Robotens vekt
- Hastighet
- IP-rangering

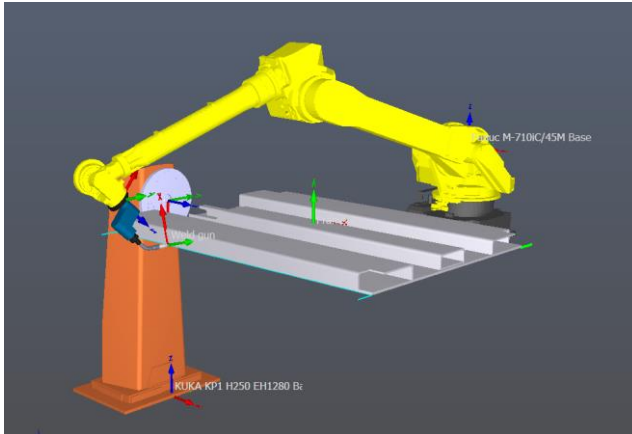
Industrielle roboter har forskjellig antall akser avhengig av oppgaven. En akse representerer en grad av frihet, forkortet DOF (*Degrees of freedom*). En 6-akset-robot tilbyr mest mulig fleksibilitet. De fleste industri roboter bruker 6 akser. Det anbefales å ha en 6-akset-robot på grunn av at man ønsker å oppnå alle posisjoner som er nødvendige for tredimensjonal sveising og da har man robotbevegelse i X-, Y- og Z-planet. 6-akse roboter brukes også i bil-industrien [41]. En akse i robotterminologi representerer en grad av frihet. Hvis en robot har tre frihetsgrader, kan den operere i planene X, Y og Z. Den kan ikke vippe eller snu. Ved å øke antall akser kan roboten få tilgang til større plass. [69]

Ved bruk av programvaren RoboDK kan man raskt finne ut om roboten man ønsker å bruke har lang nok rekkevidde. Slik kan man også finne ut hvor man bør plassere roboten for at den skal ha rekkevidde over alt som skal sveises. Rekkevidden man ønsker er fra nærmeste posisjon roboten kan nå til den lengste posisjonen.



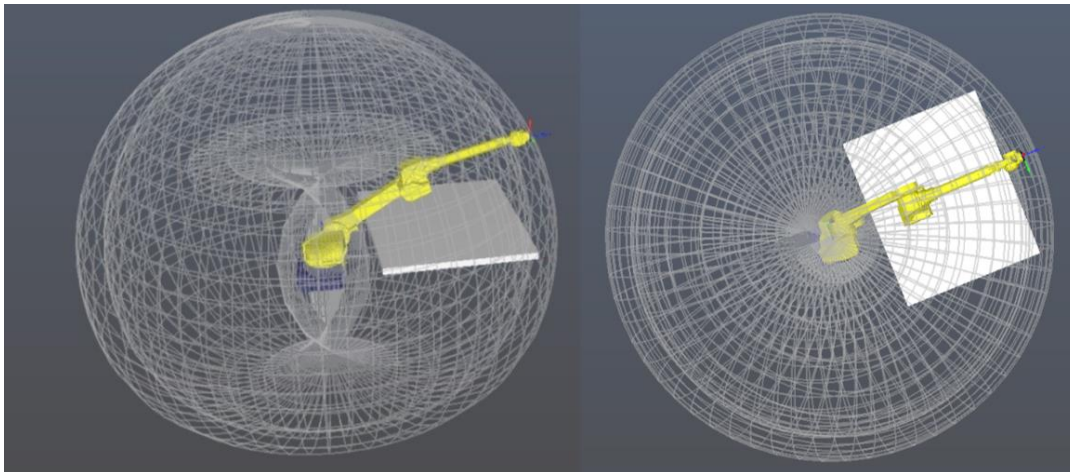
Figur 35. Viser hvordan robot må komme til nærmeste ende uten å krasje i seg selv. Bilde tatt i RoboDK programvare, 30.03.2020.

For å finne en robot som kan sveise alle panelene må man ta utgangspunktet i det største panelet til den største containeren og finne ut den lengste distansen roboten må komme til. Hvis roboten står ved midtre ende så må den ha en rekkevidde på minst 2110 mm. Men den kan ikke stå helt inntil panelet for den må kunne sveise nærmeste ende også uten å krasje i seg selv, som vist i figur 35. Fanuc M-710iC/45M har en rekkevidde som er en halv meter ekstra og virker som et godt valg til å utføre jobben.



Figur 36. Viser robot som sveiser ende lengst vekte. Bilde tatt i RoboDK programvare, 30.03.2020.

Den må også rekke til lengste ende som vist i figur 36. Denne roboten ble brukt i simuleringene.



Figur 37. Robot utstrakt til venstre og ovenfra til høyre, sammen med illustrasjon av største panel. Bilde tatt i RoboDK programvare, 26.03.2020.

Det ble tegnet en like stor plate i Autodesk Inventor for å sjekke om roboten rekker over alt som vist i figur 37. Arbeidsområdet til roboten er hele denne «kulen». Rekkevidden til roboten er 2606 mm.



Figur 38. Fanuc M-710iC/45M robot [42].

I figur 38 ser man hvordan roboten ser ut.

Tabell 2. Spesifikasjoner Fanuc M-710iC/45M [43].

Controlled axes	Repeatability (mm)	Mechanical weight (kg)	Motion range [°]						Maximum speed [°/s]						J4 Moment/ Inertia (Nm/kgm ²)	J5 Moment/ Inertia (Nm/kgm ²)	J6 Moment/ Inertia (Nm/kgm ²)
			J1	J2	J3	J4	J5	J6	J1	J2	J3	J4	J5	J6			
6	± 0.06*	570	360	225	440	800	250	800	180	180	180	250	250	360	206/28	206/28	127/20

Tabell 2 viser tabell tatt fra databladet til roboten som gir informasjon om diverse spesifikasjoner til roboten (fra kilde [43]):

- Nyttelast: 45kg
- Antall akser: 6
- Rekkevidde: 2606 mm
- Repeterbarhet: ± 0.06
- Robotens vekt: 570 kg
- Hastighet: se tabell
- IP-rangering: IP67

IP67 er en tethetsrangering som forteller om hvor tett produktet er. Første tall definerer støvtetthet, og går fra 0 til 6. Det andre tallet definerer vanntettheten, og går fra 0 til 8. IP67 vil da si: Ingen inntrenging av støv og inntrenging av vann skadelig kvantitet skal ikke skje når produktet nedsenkes under definert trykk og tid, inntil 1 meters dypde [44].

Roboten ble valgt på grunnlag av at den har 6 akser og lang nok rekkevidde for å utføre operasjonene. Men andre roboter kan også vurderes.

Tabell 3. Eksempler på andre roboter som kan vurderes. Laget i Excel, 27.04.2020, [67].

Robot	Akser	Rekkevidde (mm)	Nyttelast (kg)	Repeterbarhet (mm)	Vekt (kg)
KUKA KR 60 L16 KS	6	2952	16	0,06	650
KUKA KR 210 R2700 prime	6	2696	210	0,06	1111
Fanuc M-710iC/20M	6	2582	20	0,06	530
Daihen OTC FD-V166	6	2654	166	0,1	1010
Comau Smart NJ 40-2.5	6	2503	40	0,06	655
Comau Smart5 NJ 165-3.0	6	3000	165	0,1	1240
ABB IRB 4600-20/2.50	6	2500	20	0,05	412
ABB IRB 4600-40/2.55	6	2550	40	0,06	435

Tabell 3 viser eksempler på flere roboter som kan vurderes, men har ikke blitt testet i simuleringene. Disse ble skrevet ned basert på at de har riktige spesifikasjoner som trengs til jobben. Noen av rekkeviddene er også mindre enn Fanuc M-710iC/45M og derfor bør de testes for å sjekke at de rekker over alt på største panel som skal sveises. Flere roboter finnes i RoboDK sitt bibliotek, men det kan være unødvendig med lengre rekkevidde enn nødvendig. UR roboter har ikke lang nok rekkevidde så derfor ble de ikke vurdert her.

3.8 Vedlikehold

Det er viktig å ha en forebyggende vedlikeholdsplan for å unngå nedgang eller avslutning av produksjon. Med godt vedlikehold kan levetiden til roboten vare lenge. Roboter fra ulike produsenter har forskjellige krav på vedlikehold. For eksempel skal FANUC sine roboter ha vedlikehold hver 3850 time eller hver 12 måned. Et annet eksempel er KUKA roboter som krever vedlikehold hver 10 000 timer [45].

Forslag vedlikehold (fra kilde [45]):

- Kontrollerens minne sikkerhetskopies
- Sjekk kabler
- Kontroller bremsedriften
- Sjekk repeterbarheten med for eksempel nøyaktighetstest (se kapittel 5.8)
- Sjekk ulyder og hørbar vibrasjon
- Fettfuger
- Sjekk strømforsyning og sikkerhetsutstyr
- Test og erstatt batterier om nødvendig
- Rengjør ventilasjonsåpninger og filtre med trykkluft
- Smør gjennomføringer
- Stram utvendige bolter

Hvis man ikke sørger for forebyggende vedlikehold så kan man oppleve disse problemene (fra kilde [46]):

- Roboten kan begynne å jobbe utenfor de tiltenkte områdene
- Repeterbarheten kan bli dårligere
- Slitasje kan oppstå på kabler og ledninger
- Forstyrrelse av program og produksjonsprosesser
- Robot kan oppføre seg uforutsigbart og kan føre til skade.

4. Sikkerhet

I dette kapittelet vil det bli presentert risikoanalyse, sikkerhet ved sveising, programvaresikkerhet, sikkerhet rundt industri roboter og forslag til sikkerhet.

4.1 Risikoanalyse

Kartlegging av farer er en viktig del å vurdere ved robot styrt sveising. Det er blitt tatt en risikoanalyse av uønskede hendelser som kan oppstå, hvilken konsekvens det kan ha og hvilke tiltak som er relevante [49]. Dette vises i tabell 4 på neste side.

Grad av sannsynlighet (S), konsekvens (K) og risikonivå (R) er blitt gradert med bokstavene H, M og L. Det er også blitt markert med farger etter de ulike bokstavene.

H står for Høy, og betyr at man **må** iverksette tiltak. M står for Middels, og betyr at man **bør** iverksette tiltak. L står for lav, og man kan vurdere tiltak.

Tabell 4. Risikovurdering laget 25.03.2020 [35].

Nr.	Uønsket hendelse	Årsak	Konsekvens	S	K	R	Tiltak
1	Brann	Sveising	Personskade og skade op utstyr	M	H	H	Brannalarmanlegg og slukkeapparat tilgjengelig
2	Klem, slag, kollisjon, uforventet bevegelse	Programfeil, EMC forstyrrelse	Personskade eller skade på utstyr	M	H	H	Gjerder, lysgardin, skjermet kabler og kraft sensor
3	Strømbrydd	Kortslutning, overspenning, overbelastning, feil på strømmettet	Program stopp	M	M	M	UPS
4	Strømgjennomgang	Ikke fulgt rutiner	Personskade	L	H	H	Rutiner vedlikehold og LOTO
5	Sveisefeil	Feil sveiseparameter	Tid og kostnader på å rette opp i feil	M	M	M	Testing
6	Uautorisert tilgang	Person ikke opplært	Feil innstillinger	L	M	M	Passord på kontrollpanel
7	Sveiseblink	Ikke brukt verneustyr	Smerter øyne	H	M	H	Verneustyr
8	Ødelagte elektroniske utstyr	Slitasje, magnetiske felt	Kostnader	L	M	M	Vedlikehold og utstyr som tåler forholdene/miljøet
9	Sveiseprogram starter før jeg er skikkelig rotert	Jig henger seg opp, ikke roterer når den skal, signal ikke motatt	Skade på utstyr	L	H	H	Kraftsensor, lang nok tid på timer i program eller OPC UA kommunikasjon
10	Feil installasjon	Feil konfigurert/plassert	Feil sveising	M	M	M	Testing

Merk! Grad av sannsynlighet (S), konsekvens (K) og risikonivå (R).

4.2 Sikkerhet ved sveising

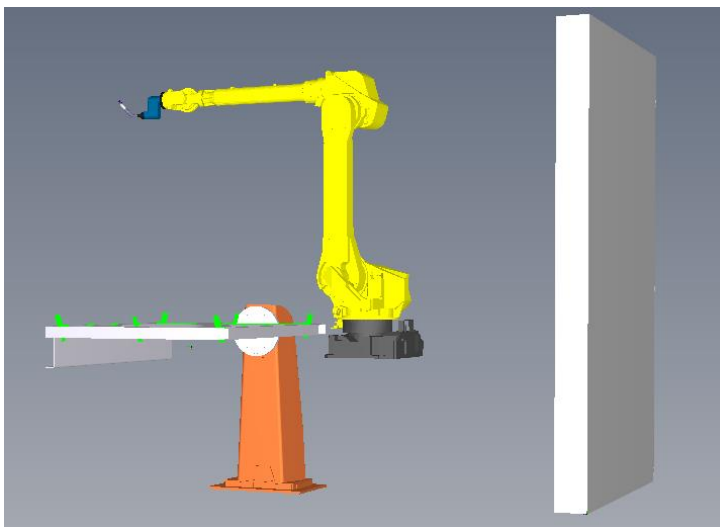
Helseskader som kan oppstå ved sveising (fra kilde [47]):

- Akutte skader ved gassutvikling.
- Kroniske skader ved finfordelt støv under sveising.
- Sveiseblink ved ultrafiolett stråling.

Verneutstyr tilpasset operasjonen som utføres er viktig og kan være sveisemaske med riktig glass, hansker for å unngå forbrenning og god bekledning slik at man ikke får skade på hud fra lyset som oppstår når det sveises. De ultrafiolette strålene kan gi hudkreft [48]. Ved bruk av robot kan arbeidsmiljøet for sveiseren bli bedre, og gi økt produktivitet [47].

4.3 Programvare sikkerhet

Man kan lage kollisjonsfrie programmer ved kollisjonsdeteksjon i RoboDK. Det bør da plasseres en robot, verktøy og minst ett objekt innenfor arbeidsområdet, ellers er det ikke nødvendig å unngå kollisjon. Hvis roboten plasseres i nærheten av en vegg kan det være en fordel å sjekke om den kolliderer og dermed endre programmet slik at dette ikke skjer [50].



Figur 39. Eksempel på å sette en vegg inn i RoboDK programvare. Bilde tatt i RoboDK programvare, 26.04.2020.

Man kan tegne veggen inn i Autodesk Inventor og deretter plassere den i RoboDK som vist i figur 39. Det anbefales å gjøre 3D tegningene større enn de faktisk er for å trygt forutse kollisjoner. Slik kan man da plassere veggen fra der roboten skal stå for å sjekke at roboten ikke kolliderer i den. Roboten kan fremdeles kollidere med objekter i den virkelige verden hvis den kommer for nær dem og dette skjer vanligvis på grunn av små forskjeller mellom den simulerte roboten og den fysiske roboten. Det er uvanlig at en oppgave krever alt av robotens arbeidsområde. Man kan derfor sette inn leddgrenser i RoboDK slik at roboten ikke beveger seg mer enn de leddgrensene som er satt. Dette er kun programvare sikkerhet og det anbefales å ha en ekstra fysisk sikkerhet for å hindre kollisjoner. Ved å sette inn leddgrenser så vil også hastigheten på prosessen med å generere kollisjonskart bli hurtigere [50].

4.4 Forslag sikkerhet

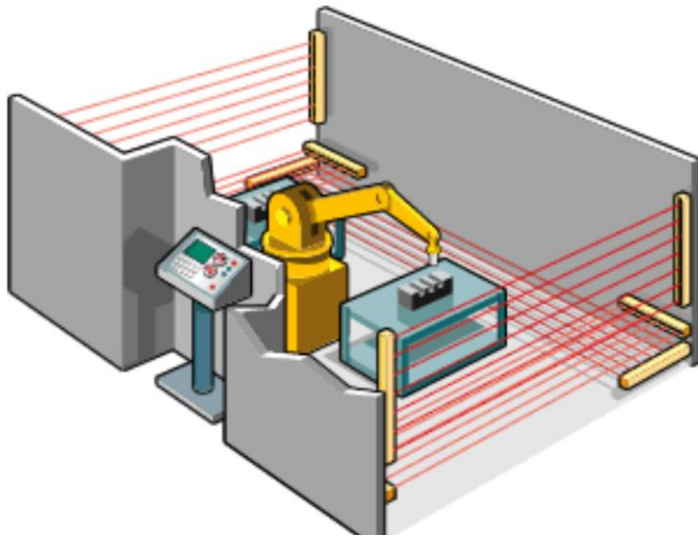
De viktigste standardene for robotiske løsninger kan klassifiseres i tre kategorier: A-, B- og C - standarder. Klasse A er ISO 12100 og IEC 61508 og de definerer grunnleggende terminologi og metodikk som brukes for å oppnå sikkerhet for maskiner. Klasse B er delt inn i underkategori B1 og B2. B1 er for eksempel ISO 13849-1 og IEC 62061. Disse henviser til utformingen av henholdsvis sikkerhetssystem med lav kompleksitet og "Safety PLCs". B2 er ISO 13850 og ISO 13851. Disse beskriver de spesifikke funksjonelle aspektene ved henholdsvis nødstoppenheter og tohånds kontrollenheter. Klasse C er sikkerhetsstandarder som spesifiserer sikkerhetsmessige tiltak for spesifikke maskiner. Hvis Type C-standarder er gitt, har disse prioritet fremfor Type A- og Type B-standarder. Type C-standarder som regulerer sikkerheten til industri roboter er ISO 10218 [71].

Norsk standard ISO 10218 er delt inn i 2 deler hvor del 1 handler om roboter og sikkerhetskrav til disse, samt hvordan foreta en risikovurdering. Del 2 handler om sikkerhetskrav til robotcellen. Denne standarden krever PL=d sikkerhet for farlige robot applikasjoner [52].

Tabell 5. Performance Level. [53] og [<https://www.iso.org/>].

Performance Level (PL)	Probability of Dangerous Failure per Hour (PFHd) 1/h
a	$\geq 10^{-5}$ and $< 10^{-4}$ (0.001% to 0.01%)
b	$\geq 3 \times 10^{-6}$ and $< 10^{-5}$ (0.0003% to 0.001%)
c	$\geq 10^{-6}$ and $< 3 \times 10^{-6}$ (0.0001% to 0.0003%)
d	$\geq 10^{-7}$ and $< 10^{-6}$ (0.00001% to 0.0001%)
e	$\geq 10^{-8}$ and $< 10^{-7}$ (0.000001% to 0.00001%)

PL = d er en verdi av ytelsesnivå som beskriver evnen til sikkerhetsrelaterte deler av et system. PL = d er svært pålitelig. PL står for Performance Level og er beskrevet av sannsynlighet for farlig feil per time som vist i tabell 5 [53].



Figur 40. Robotcelle sikkerhet eksempel [51].

Varme partikler, røyk, støv, intense lysglimt, elektriske overspenninger og elektromagnetisk interferens er utfordringer som må tas hensyn til og kan være skadelig for utstyr og maskiner i nærheten. Maskinsikringsutstyr må velges ved bruk av streng og formell risikovurdering av potensielle farer. Roboter tåler røyk, gnister og lys fra intense sveise buer. Disse farene blir da mindre eksponert for menneskene ved bruk av robot. Men man må allikevel ta hensyn til at menneskene også må beskyttes fra roboter og maskiner slik som vist i figur 40 [51].

Målet er å stoppe roboten fra å kollideres/skade mennesker og utstyr. Dette kan gjøres ved å for eksempel gjerde inn robot som vist i figur 40. Man kan bruke sikkerhets-lysgardin som bryter tilførsel til robot når person eller gjenstand bryter de infrarøde lysstrålene. Disse tåler røyk, støv, svevestøv og lyskilder som sveiseblits som oppstår ved sveising. Eventuelt kan man bruke sikkerhetsmatter som sender et signal til kontrolleren når noen har tråkket på matten, som stopper eller forhindrer oppstart av roboten. Strømforsyninger for sikkerhetsutstyr bør skjermes mot elektriske lysbuer slik at elektriske ledninger og sveisebuer ikke ødelegger elektronisk utstyr [51]. Etter risikoanalysen bør det tas hensyn til en rekke farer og tiltak som bør eller må tas i bruk:

Oppsummert de tiltak som **må** iverksettes:

- Slokkeapparat tilgjengelig
- Brannalarmanlegg
- Gjerder rundt robotcelle
- Lysgardin for å bryte strømtilførsel eller stoppe robot om person går gjennom lysgardinen og inn i robotcelle.
- Skjermet kabler for å beskytte mot EMC forstyrrelse
- Kraftsensor for å stoppe kollisjon
- Rutiner for vedlikehold
- LOTO
- Verneutstyr
- Lang nok tid på *timer* i program for robot eller OPC UA kommunikasjon

Oppsummert de tiltak som **bør** iverksettes:

- UPS
- Teste sveiseparametre
- Passord på kontrollpanel
- Rett utstyr for miljøet
- Teste om plassering av robot og komponenter er riktig plassert

Sikkerhetsbrudd kan være kostbart og føre til personskade. Det er viktig å følge manualer og sikkerhetsstandarder for systemet. Det er viktig å følge prosedyren for LOTO hvor en fjerner strømtilførselen, låser av og merker med skilt. Nødstoppe må også være tilgjengelig for å kunne bryte strømtilførsel. Med en god vedlikeholdsplan vil også levetiden til roboten være lengre. Robot celle må være ren og fri for komponenter som ikke skal være der og ting som kan forårsake fare for brann under sveising. Med rett type utstyr kan man også tenke at ved PL=d sikkerhetsnivå bør en ha for eksempel ekstra kontaktorer slik at om en av dem brenner seg fast på grunn av feil, så er det en ekstra som kan bryte strømtilførselen. PLS må også ha rett sikkerhetsnivå. Det blir konkludert med sikkerhetsnivå PL nivå d som anbefalt i ISO 10218.

5. Diskusjon

I dette kapittelet vil det bli diskutert andre muligheter til programmering, sveiseutstyr som kan brukes, hypotetisk investeringsanalyse, utfordringer, forslag ved strømbrydd, forslag sensorer, alternativ løsning ved bruk av OPC UA, kalibrere og nøyaktighets teste robot ved bruk av RoboDK.

5.1 Andre OLP programvarer

Det finnes også andre OLP programvarer som kan brukes:

- Artiminds RPS [61]
- Octopuz offline programming software [61]
- Robotmaster [61]
- RobotStudio (ABB) [82]

De kan alle brukes til sveiseoperasjoner og også flere operasjoner som (fra kilde [61]):

- Montering
- Materiell tilsetning
- Materialhåndtering / pakking
- Fjerning av materiale

5.2 Sveiseutstyr

Valg av sveisepistol bør testes for kollisjon ved å tegne det i CAD og overføre til RoboDK for å justere programmet nøyaktig. Sveiseapparat brukt i simuleringen er et som allerede eksisterte i RoboDK sitt bibliotek.

Siden man ønsker å kunne bruke MIG over alt så kan man kjøpe sveiseutstyr som kan automatiseres med roboten. Da kan roboten styre strømstyrken etter hvor den skal sveise. Man må ha større strømstyrke rundt rammen og mindre strømstyrke rundt de korrugerte platene. Eventuelt kan strømstyrken manuelt styres under produksjonen. Man må uansett ha en person til stede for å sette komponentene på plass i jiggen og overvåke roboten.



Figur 41. Kemppi A7 MIG welder [62].

Kemppi A7 MIG welder er en robotdrevet løsning for lysbuesveising og kan brukes med alle robotmerker. Man kan se hvordan den ser ut i figur 41. Denne kan brukes til en lang rekke ulike sveiseoppgaver og har en rask sveisehastighet. Man får tilgang til strømkilden via en datamaskin og en hvilken som helst nettleser og krever ingen tilleggsprogrammer. Den har en integrert gasstrømsensor i trådmateren og overvåker dekkgasstrømmen nærmere på lysbuen slik at det ikke er behov for tilleggssensorer [62].

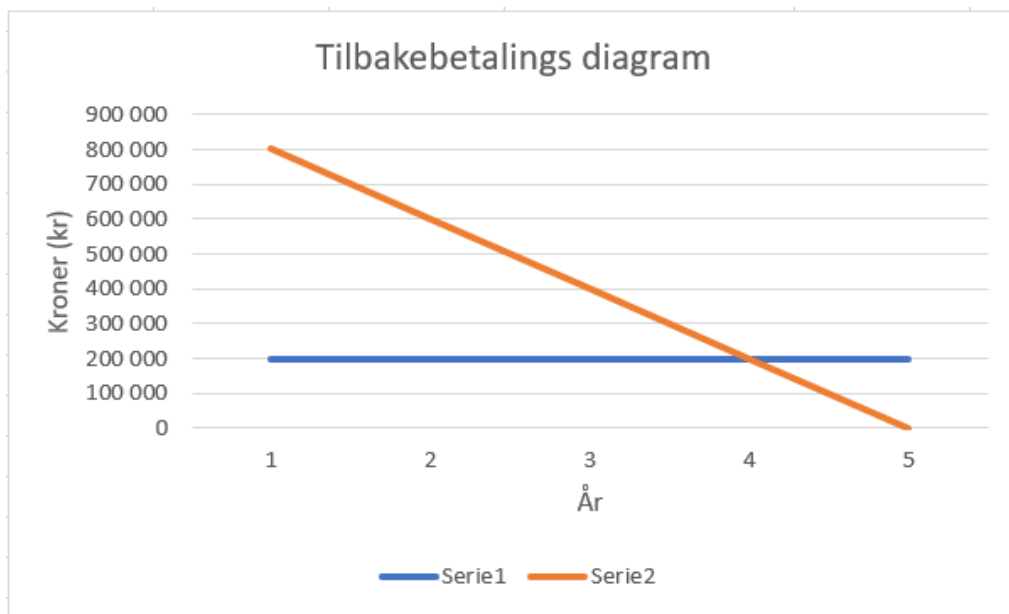
Kemppi A7 MIG welder har en *Touch Sense Ignition*-funksjon som gir minimalt med sprut og stabiliserer lysbuen under tenning. Kemppi WiseFusion er en programvare som sikrer jevn sveisekvalitet ved automatisk justering av buelengden. Det finnes også produktalternativer når det gjelder programvare, strømkilder og trådmaterer [63].

5.3 Hypotetisk investeringsanalyse

Man kan for eksempel ta utgangspunktet i prisene fra rapporten «Fresing med industrirobot» av Marås, Christian; Sundsøy, Morten; Thomsen, Svanhild N.; Gjeraker, Lars Martin [54] fra 2015. Prisen ble da rundt kr 750 000 for dette systemet, men det kan komme flere kostnader ved implementering av andre sikkerhetsfunksjoner og utstyr. Merk at prisene er ikke nøyaktige. Man kan ta utgangspunkt i at start investeringen kan være rundt kr 1 000 000. For eksempel kan man ha en årlig kostnad med 1 operatør som jobber 37,5 timer per uke til en kostnad på ca. kr 350 000 per år og 150 000 i jigg, prosessforbedring og vedlikehold. Hvis roboten da erstatter 1 mann så blir besparelser $350\,000 - 150\,000 = \text{kr } 200\,000$. Tiden det tar før systemet er tilbakebetalt vil da være investering delt på besparelser: $1\,000\,000 / 200\,000 = 5$ år, som vist i tabell 6. Dette er en grov beregning og bør betraktes som en estimering [75].

Tabell 6. Eksempel nedbetaling hvert år, laget i Excel, 27.03.2020.

År	Besparelser (kr)	Robot system kostnad (kr)
1	200 000	800 000
2	200 000	600 000
3	200 000	400 000
4	200 000	200 000
5	200 000	0



Figur 42. Tilbakebetalings diagram, laget i Excel, 27.03.2020

Den blå linjen i figur 42 vil være besparelsene hvert år og den oransje er investeringen. Mange av FANUC sine kunder har opplevd 10+ års pålitelig service og dermed vil dette være en lønnsom investering. En slik robot vil kunne kjøre 80 000 til 100 000 timer uten feil [64].

Man vil også kunne spare inn penger ved at robot gir økning i produktivitet, god kvalitet ved rette sveiseparametre, og mindre omarbeiding [23]. Da vil tilbakebetalingen av investeringen ta kortere tid enn 5 år. Robotenes kostnader varierer fra produsenter og spesifikasjoner på roboten.

5.4 utfordringer

Hva man bør tenke på før man investerer i en robotstyrt sveise prosess er om man har plass til det. Roboten kan være veldig stor og kreve mye utstyr og beskyttelse. Man må også ha nok elektrisitet for å gi strøm til alt utstyret. Det er også viktig å ha en erfaren sveiser som kan bedømme og kalibrere robot sveisingen. Programmeringsmetodene er såpass enkle at en kan fort (uker) lære seg å programmere en robot og dermed omprogrammere roboten om nødvendig [55].



Figur 43. Bilde av dørlås hvor det må sveises manuelt på PUSH-IN containeren [7].

Sveise operasjoner som er komplekse bør utføres av en manuell operatør. For eksempel kan de stedene som vist i figur 43 være litt vanskeligere for en robot og da kan de sveises manuelt. Skal man bestemme om en sveise prosess bør automatiseres kan en bruke CAD-tegning og overføre til OLP programvare for å simulere operasjonen for å finne en pålitelig løsning. Dette vil gjøre det enklere å planlegge sveise operasjonen [55].

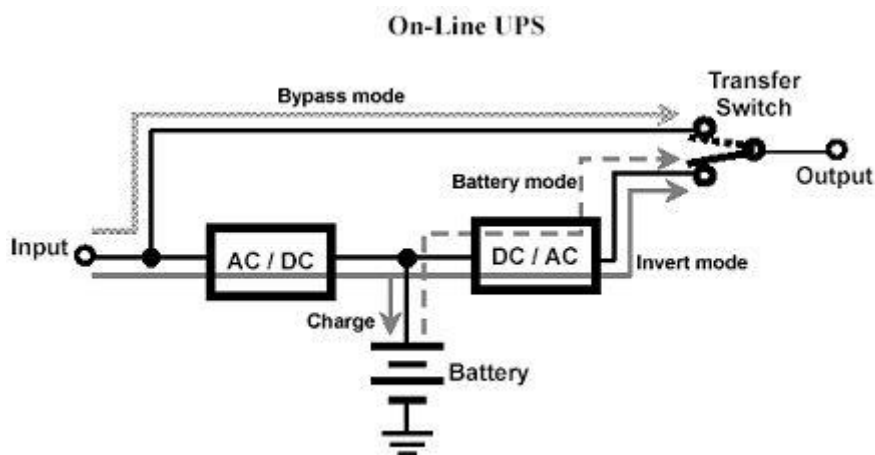
Reduksjon i arbeidskraft er ofte den største økonomiske besparelsen som gjøres ved automatisert produksjon. Hvis en sveiser erstattes av en robot, kan han sendes til en annen arbeidsstasjon eller oppgraderes til en tilsynsstilling [55]. I denne oppgaven bør det uansett være person til stede for å overvåke roboten, sette delene på plass i jiggen, overføre program etter hvilket panel som skal sveises og styre strøm. Eventuelt kan strømmen også bli automatisert som foreslått i kapittel 5.2. Strømmen må stilles etter om roboten sveiser ramme eller korrugerte plater siden rammen er tykkere og krever derfor mer strøm enn de korrugerte platene.

5.5 Strømbrudd, UPS

UPS står for *Uninterruptible Power Supply* som betyr avbruddsfri strømforsyning. Den består av vekselretter, likeretter og batterier. Den hindrer at man mister strømmen ved strømbrudd og beskytter også mot ustabil nettspenning. Det finnes 3 typer (fra kilde [56]):

- Off-line
- Line interactive
- On-line

Off-line og *Line interactive* fungerer slik at ved strømbrudd vil UPS koble inn batteriene ved bruk av rele, noe som tar mellom 4-10 ms. En *On-line* UPS vil hele tiden være tilkoblet og derfor vil det ikke være noe omkoblingstid mellom normal og batteridrift. *On-line* UPS vil da være et godt valg for total beskyttelse. Her blir spenningen først likerettet før den blir på nytt vekselrettet. Denne typen slår seg ikke av ved overlast fordi den har innebygget *bypass*. Den gir brukeren en generell beskyttelse som spenningsregulering, overspenningsvern og filtrerer andre kritiske variasjoner i nettspenningen [56].



Figur 44. On-Line UPS hvordan koblet [56].

I figur 44 kan man se et skjema over hvordan en *On-Line* UPS fungerer. Det kan være en fordel å ha en UPS for å forsikre seg om at sikkerhetsfunksjoner alltid har tilførsel av strøm. Kan også brukes til andre apparater om det trengs, som for eksempel sveiseutstyr.

5.6 Sensorer

Kraftsensor (også kalt FT-sensor) er designet for å oppdage krefter som brukes mellom dens base og sensorplate. De plasseres vanligvis på en robotarm. Ved bruk av kraftsensor gir det roboten mulighet til å samarbeide med miljøet sitt. Dette er også et godt sikkerhetstiltak for å hindre at roboten skader noe med for mye krefter. Ved bruk av en 6-aksed FT-sensor kan roboten oppdage krefter i alle tre kartesiske aksene. De kartesiske aksene er X, Y og Z. Man kan også feste dem til en ekstern struktur [57]. Man kan bruke gjennomlys-bue-sømsporing, lasersømsporer eller berøringsfølelse for å sanntidsspore sveiseskjøten om CAD modell ikke samsvarer med den faktiske delen [35].

Andre sensorer som bør implementeres:

- Detektore beskjed fra robot om å snu panel
- Detektore posisjon for roteringen



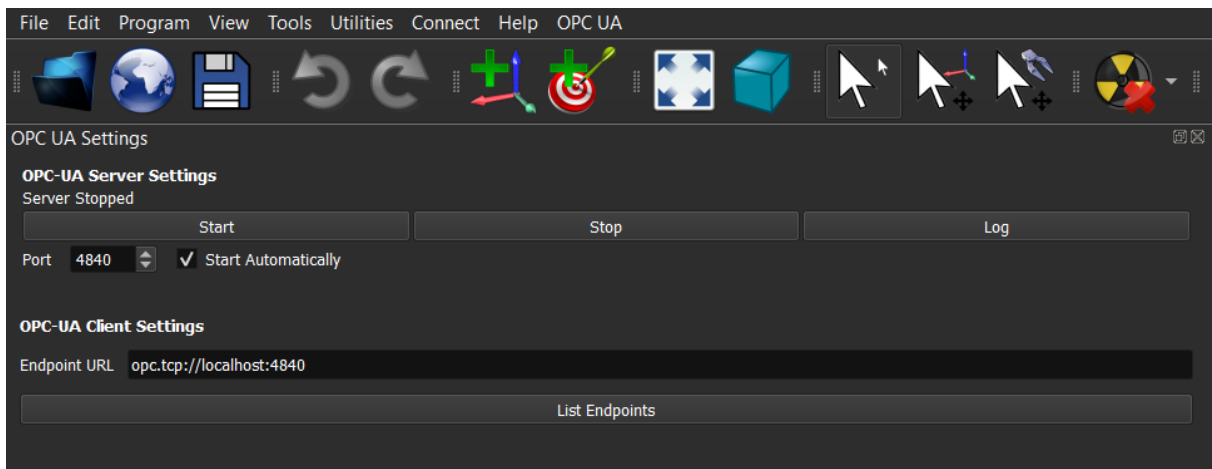
Figur 45. Baumer SmartReflect [77].

Etter at roboten har sveiset øvre del av panel så skal den plassere seg ved en sensor som registrer dette og sende signal til PLS om at den skal rotere til neste posisjon. Denne sensoren som detekterer dette, kan for eksempel være en *SmartReflect*, som vist i figur 45, som er en sensor som detekterer at et objekt bryter lyset og dermed endres vinkelen slik at sensoren registrerer denne endringen [65].

5.7 OPC UA

Man kan laste ned et tilleggsprogram som gjør at man kan bruke PLS program sammen med RoboDK. Da kan man sende og lese signaler i stedet for å bruke *timer* i programmet. Ved å bruke dette får man kontroll på det industrielle utstyret og får tilgang til verdier over nett [58].

OPC står for *Open Platform Communications* og er et tilleggsprogram som gir tilgang til maskiner, enheter og andre systemer. Produsenten av maskinvaren gir en OPC-server for systemet sitt. OPC-klienten er det logiske motstykket til serveren og brukes til å lese opp dataene som leveres av serveren. OPC-klienter kan være applikasjoner som er avhengig av datautveksling med industrisystemer. Man kan ha flere servere som venter på å bli lest fra klienter slik at serveren kan sende data over til klienten. Når ny data kommer inn i serveren kan klienten instrueres til å få tilsendt denne dataen. Serveren kan hente denne dataen fra for eksempel PLS eller robot. Det finnes forskjellige typer OPC men OPC UA er den nyeste [59].



Figur 46. OPC UA fra RoboDK programvare. Bilde tatt i RoboDK programvare, 30.03.2020.

OPC UA kan brukes i RoboDK som vist i figur 46. UA står for *Unified Architecture*. Forskjellen mellom de klassiske OPC standardene og OPC UA er at med OPC UA er det ikke en utfordring med kommunikasjon over nettverk og den er plattformuavhengig og kan derfor brukes på både Apple, Linux og Windows [59].

Det er en risiko om det skjer en feil ved rotering av panelet når robot har innstilt *timer*. Og ved bruk av OPC UA kan roboten få signal fra PLS til å begynne sveiseprogram i stedet for at roboten venter en tid. Da brukes det I/O i RoboDK i stedet for *timer* funksjonen.

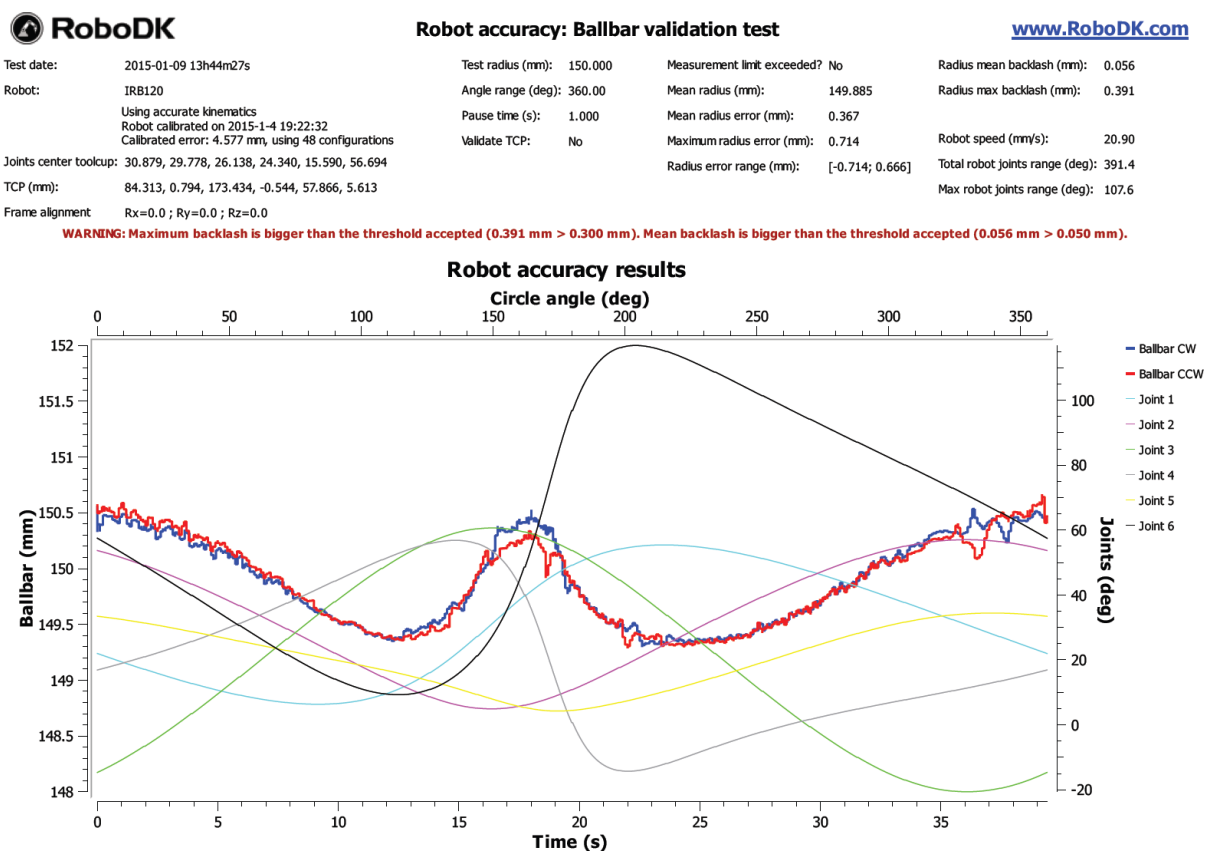
5.8 Kalibrering og nøyaktighetstest

Industri roboter er svært repeterbare men ikke nøyaktige. Ved robotkalibrering kan nøyaktigheten til industriroboten forbedres med en faktor 2 til 10. Eventuelt kan man utføre en «ballbar test» for å validere nøyaktigheten til en robot [60].



Figur 47. ABB-robot som utfører ballbar test [66].

Figur 47 viser ballbar test utført ved bruk av en ABB-robot. RoboDK programvare guider deg gjennom hvert trinn for å fullføre en ballbar test og genererer en ytelsesrapport for valgt robot [66].



Figur 48. PDF-rapport for robot nøyaktighet resultater fra RoboDK [66].

Når *ballbar* test er fullført, oppnås en PDF-rapport som viser måling av kulestangen i mm (Y-aksen til venstre) for med urviser og mot urviseren (henholdsvis blå og røde linjer) med hensyn til tiden i sekunder (nederste X-akse) som vist i figur 48. Robotleddene vises i grader (høyre Y-akse) med hensyn til sirkelvinkelen. 360 grader betyr en full sving. En defekt motor vil vise betydelige måleendringer når den korresponderende leddbevegelsen endrer retningen og merker betydelig tilbakeslag [66].

6. Konklusjon

Målet med oppgaven var å se på mulighet for å bruke robot til å utføre sveiseoperasjoner på container panel laget i aluminium. Oppgaven var å lage simulering og programkode for sveiseoperasjoner som skal utføres av robot, foreslå robot som kan brukes, forklare hvordan programvaren kan testes på for eksempel en UR5 robot, lage PLS program for rotering av jigg og identifisere problemstillinger.

Simulering og programkode ble laget ved bruk av RoboDK v4.2.4 programvare som er off-line programmering med simulering. Roboten som ble anbefalt er Fanuc M-710iC/45M. Den har 6 akser, lang nok rekkevidde og er testet i simuleringen. Utstyret som blir valgt til sveiseoperasjoner må tegnes i CAD for å implementeres i RoboDK for en nøyaktig simulering. Simuleringen brukte sveiseutstyr som allerede var i RoboDK sitt bibliotek.

PLS kan brukes til å styre automatisert jigg. I løsningen har det blitt brukt *timer* for å få robot til å vente til roteringen er fullført. Ved bruk av sensorer kan man få PLS og robot til å samarbeide. Man kan erstatte denne løsningen med OPC UA kommunikasjon for da slipper man å bruke for eksempel *timer*. OPC UA kan brukes for å motta informasjon om sensorenes tilstand og sende til Robot eller PLS.

Det ble totalt 8 programmer for operasjonene. Program for valgt panel må lastes opp til robot. Når prosessen er ferdig for valgt panel stoppes programmet og rutine for LOTO iverksettes slik at neste panel kan klargjøres. Program lastes opp til robot for valgt panel og startes. Brukeren velger selv hvilket panel som skal sveises og laster opp ønsket program til robot.

Sikkerhet er en viktig del ved robot som utfører sveising og det ble konkludert med PL nivå d som det anbefales i Norsk Standard ISO 10218.

Videre arbeid kan være å lage program for den andre containeren, som er større, eller lage sveise program for komplett container i stedet for at det gjøres manuelt. Man kan også endre programmet til å bruke OPC UA kommunikasjon for å unngå bruk av *timer*. Den automatiske jiggen ble ikke laget så derfor kan denne også arbeides med. Kan også teste sveiseparametre, programvare og kode laget for simuleringene. Det kan også ses på mulighet til andre løsninger ved for eksempel bruk av 2 UR roboter i stedet for 1 industrirobot.

7. Kilder

- [1] LEVEL Power & Automation. About us. Dato hentet: 11.02.2020. [Internett]. Tilgjengelig: <https://www.levelpower.no/about/>
- [2] LEVEL Power & Automation. Automation products. Dato hentet: 11.02.2020. [Internett]. Tilgjengelig: <https://www.levelpower.no/automation-products/>
- [3] A. Olufsen Ship & Offshore. Om oss. Dato hentet: 11.02.2020. [Internett]. Tilgjengelig: <https://www.aoso.no/nb/om-oss/>
- [4] A. Olufsen Ship & Offshore. Tjenester. Dato hentet: 03.03.2020. [Internett]. Tilgjengelig: <https://www.aoso.no/nb/tjenester/>
- [5] A. Olufsen Ship & Offshore. Fasiliteter. Dato hentet: 03.03.2020. [Internett]. Tilgjengelig: <https://www.aoso.no/nb/fasiliteter/>
- [6] A. Olufsen Ship & Offshore. Push-in container. Dato hentet: 03.03.2020. [Internett]. Tilgjengelig: <https://www.aoso.no/nb/tjenester/produkter/push-in-container/>
- [7] Video A. Olufsen Ship & Offshore. A. Olufsen – Push in Container (English). Tuesday, March 26, 2019 at 10:42 AM. Dato hentet: 03.03.2020 [Online Video]. Tilgjengelig: <https://www.aoso.no/nb/tjenester/produkter/push-in-container/>
- [8] Henning Johansen. Sveiste forbindelser. Dato hentet: 08.02.2020. [Internett]. Tilgjengelig: <http://www.ansatt.hig.no/henningj/materialteknologi/joining-al-sveising.htm>
- [9] OpenLearn. Gas shielded arc welding processes (TIG/MIG/MAG). Dato hentet: 03.03.2020. [Internett]. Tilgjengelig: <https://www.open.edu/openlearn/science-maths-technology/engineering-technology/manupedia/gas-shielded-arc-welding-processes-tig/mig/mag>
- [10] Kemppi. TIG-sveising. Dato hentet: 07.03.2020. [Internett]. Tilgjengelig: <https://www.kemppi.com/no/stotte/sveise-abc/tig-sveising/>
- [11] Per K. Kofstad (Universitetet i Oslo) og Bjørn Pedersen (Universitetet i Oslo). Wolfram. Dato hentet: 07.03.2020. [Internett]. Tilgjengelig: <https://snl.no/wolfram>
- [12] Norvald Nesse. Inertgass. Dato hentet: 03.03.2020. [Internett]. Tilgjengelig: <https://snl.no/inertgass>
- [13] ESAB knowledge center. What shielding gas should I use when welding aluminium. Dato hentet: 02.03.2020. [Internett]. Tilgjengelig: <https://www.esabna.com/us/en/education/blog/what-shielding-gas-should-i-use-when-welding-aluminum.cfm>
- [14] MIG welding. Arc welding faults. Dato hentet: 08.03.2020. [Internett]. Tilgjengelig: <https://www.mig-welding.co.uk/arc-welding-faults.htm>
- [15] Welders Universe. Welding parameters. Dato hentet: 01.03.2020. [Internett]. Tilgjengelig: https://www.weldersuniverse.com/welding_parameters.html
- [16] esab. Welding Parameters and Techniques. Dato hentet: 01.03.2020. [Internett]. Tilgjengelig: https://www.esabna.com/euweb/mig_handbook/592mig7_1.htm

- [17] Miller. MIG WELDING: Setting the Correct Parameters. Dato hentet: 01.03.2020. [Internett]. Tilgjengelig: <https://www.millerwelds.com/resources/article-library/miggmaw-101-setting-the-correct-parameters>
- [18] Alumeco. EN AW-6082. Dato hentet: 22.02.2020. [Internett]. Tilgjengelig: <https://www.alumeco.no/aluminium/profiler/bolter/skinner/skinner/en-aw-6082/15-x-5-mm--anodisert-15-my/p/576/18177>
- [19] Alumeco. EN AW-5754. Dato hentet: 22.02.2020. [Internett]. Tilgjengelig: <https://www.alumeco.no/aluminium/plater/ubehandlede-plater/en-aw-5754/8-x-1020-x-2020-mm/p/391/3792>
- [20] Manufacturing Engineering and Technology, Seventh Edition av Serope Kalpakjian og Steven R. Schmid. Side 883.
- [21] Alfred Byron. MIG Welding Settings Explained. Dato hentet: 03.03.2020. [Internett]. Tilgjengelig: <https://www.ratemywelder.com/best-mig-reviews/mig-welding-settings-explained/>
- [22] Esab. En introduksjon til sveiseinspeksjon. Dato hentet: 22.02.2020. [Internett]. Tilgjengelig: <https://www.esab.no/no/no/education/blog/an-ntroduction-to-welding-inspection.cfm>
- [23] SCOTT. Top 10 Advantages of Robot Welding. Dato hentet: 02.03.2020. [Internett]. Tilgjengelig: <https://www.scottautomation.com/news/articles/top-10-advantages-of-robotic-welding/>
- [24] cs.stanford.edu. Robotics: A Brief History. Dato hentet: 04.03.2020. [Internett]. Tilgjengelig: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/1998-99/robotics/history.html>
- [25] Universal Robots. Collaborative robotic automation | Cobots from Universal Robots. Dato hentet: 08.03.2020. [Internett]. Tilgjengelig: <https://www.universal-robots.com/>
- [26] Ivar M. Liseter. Robot. Dato hentet: 20.02.2020. [Internett]. Tilgjengelig: <https://snl.no/robot>
- [27] Farnell. Omron. Dato hentet: 08.03.2020. [Internett]. Tilgjengelig: <https://no.farnell.com/omron-industrial-automation/cpl1e-n40dr-a/plc-cpu-unit-relay-o-p-40-i-o/dp/2730076>
- [28] PLC Manual. PLC history. Dato hentet: 31.03.2020. [Internett]. Tilgjengelig: <http://www.plcmanual.com/plc-history>
- [29] PLC Manual. Introduction to PLCs. Dato hentet: 31.03.2020. [Internett]. Tilgjengelig: <http://www.plcmanual.com/>
- [30] Alex Owen-Hill. What are the different programming methods for Robots? Dato hentet: 03.03.2020. [Internett]. Tilgjengelig: <https://blog.robotiq.com/what-are-the-different-programming-methods-for-robots>
- [31] Tanya M. Ananden. Demystifying Robot Offline Programming. Dato hentet: 06.02.2020. [Internett]. Tilgjengelig: https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/Demystifying-Robot-Offline-Programming/content_id/7467

- [32] RobodK. About RoboDK. Dato hentet: 07.03.2020. [Internett]. Tilgjengelig: <https://robodk.com/about>
- [33] Universal Robots. RoboDK simulation and offline programming. Dato hentet: 22.02.2020. [Internett]. Tilgjengelig: <https://www.universal-robots.com/plus/software/robodk-simulation-and-offline-programming/>
- [34] RoboDK. RoboDK Add-In for Inventor. Dato hentet: 03.03.2020. [Internett]. Tilgjengelig: <https://robodk.com/doc/en/Plugin-Inventor.html#PluginInventor>
- [35] Universal Robots. User Manual UR5/CB3, Version 3.4.5. Dato hentet: 04.03.2020. [Internett]. Tilgjengelig: https://courses.ideate.cmu.edu/60-428/s2018/wp-content/uploads/2017/12/UR5_User_Manual_en_US-3.4.5.pdf
- [36] RoboDK. Universal Robots. Dato hentet: 28.02.2020. [Internett]. Tilgjengelig: <https://robodk.com/doc/en/Robots-Universal-Robots.html#UR>
- [37] Datamaskin. Grunnleggende om PLS. Dato hentet: 27.02.2020. [Internett]. Tilgjengelig: <http://www.datamaskin.biz/Hardware/computer-drives-storage/53186.html#.XlfSwmhKhPY>
- [38] Programmerbare logiske styringer av Dag Håkon Hanssen, 4. utgave, år 2015. kapittel 12 Sekvensielle Funksjonskart (SFC) s.357
- [39] Programmerbare logiske styringer av Dag Håkon Hanssen, 4. utgave, år 2015. kapittel 12 Sekvensielle Funksjonskart (SFC) s.369
- [40] Kuka. KP1-H single-axis positioner. Dato hentet: 03.03.2020. [Internett]. Tilgjengelig: <https://www.kuka.com/en-us/products/robotics-systems/robot-periphery/positionierer/kp1-h>
- [41] Mark Crawford. How many axes does your robot need? Dato hentet: 03.03.2020. [Internett]. Tilgjengelig: <https://www.asme.org/topics-resources/content/many-axes-does-robot-need>
- [42] Fanuc. M-710iC/45M. Dato hentet: 03.03.2020. [Internett]. Tilgjengelig: <https://www.fanuc.eu/hr/en/robots/robot-filter-page/m-710-series/m-710ic-45m>
- [43] Fanuc. M-710iC/45M. Datasheet M-70iC/45M. Hentet: 15.03.2020. [Internett] Tilgjengelig: <https://www.fanuc.eu/hr/en/robots/robot-filter-page/m-710-series/m-710ic-45m>
- [44] Arbeidslys. IP-tetthetsgrader. Dato hentet: 04.03.2020. [Internett]. Tilgjengelig: <https://www.arbeidslys.no/tetthetsrating>
- [45] RobotWorx. Preventative Maintenance for Industrial Robots. Dato hentet: 04.03.2020. [Internett]. Tilgjengelig: <https://www.robots.com/articles/preventative-maintenance-for-industrial-robots>
- [46] Robot world. How often do robots need preventative maintenance? Dato hentet: 04.03.2020. [Internett]. Tilgjengelig: https://www.northlinerobotworld.com/about/blog/How-Often-Do-Robots-Need-Preventative-Maintenance_AE15.html
- [47] Einar Halmøy, Almar Almar-Næss og Tor John Rødsås (Universitetet i Agder). Sveising. Dato hentet: 05.02.2020. [Internett]. Tilgjengelig: <https://snl.no/sveising>
- [48] ndla. HMS ved sveising. Dato hentet: 04.03.2020. [Internett]. Tilgjengelig: <https://ndla.no/nb/subjects/subject:28/topic:1:22898/topic:1:58242/resource:1:45141>

- [49] HMS.NO. Risikovurdering. Dato hentet: 31.03.2020. [Internett]. Tilgjengelig: <https://www.hms.no/risikovurdering/risikovurdering/>
- [50] RoboDK. Collision Detection. Dato hentet: 28.01.2020. [Internett]. Tilgjengelig: <https://robodk.com/doc/en/Collision-Avoidance.html#GeneralCollisions>
- [51] Sergio Aguilar. Playing it safe with robotic welding. Dato hentet: 20.01.2020. [Internett]. Tilgjengelig: <https://www.thefabricator.com/thewelder/article/automationrobotics/playing-it-safe-with-robotic-welding>
- [52] Norsk Standard NS-EN ISO 10218-1:2011, Roboter og robottekniske innretninger. Sikkerhetskrav for roboter for industrielt miljø Del 1: Roboter.
- [53] Keyence. PLr and PL (from ISO 13849-1:2006). Dato hentet: 04.03.2020. [Internett]. Tilgjengelig: <https://www.keyence.com/ss/products/safetyknowledge/performance/level/>
- [54] Marås, Christian; Sundsøy, Morten; Thomsen, Svanhild N.; Gjeraker, Lars Martin (Bachelor thesis, 2015). «Fresing med industrirobot». Tilgjengelig: <https://hvlopen.brage.unit.no/hvlopen-xmlui/handle/11250/286124>
- [55] Robotiq. 5 factors to consider before moving to a robotic welding process. Dato hentet: 02.03.2020. [Internett]. Tilgjengelig: <https://blog.robotiq.com/bid/70461/5-Factors-to-Consider-Before-Moving-to-a-Robotic-Welding-Process>
- [56] Power source. Hva er egentlig UPS? Dato hentet: 09.03.2020. [Internett]. Tilgjengelig: <https://www.powersource.no/hva-er-egentlig-ups/>
- [57] Mariane Davids. How force sensors are used in robotics research. Dato hentet: 08.02.2020. [Internett]. Tilgjengelig: <https://blog.robotiq.com/robotiq-ebook-summary-force-sensors-in-robotics-research>
- [58] RoboDK. RoboDK Plug-In Interface Introduction. Dato hentet: 20.03.2020. [Internett]. Tilgjengelig: <https://robodk.com/doc/en/PlugIns/index.html>
- [59] Novotek. OPC og OPC UA. Dato hentet: 27.02.2020. [Internett]. Tilgjengelig: <https://www.novotek.com/no/l-sninger/kepware-opc-kommunikasjonsplattform/opc-og-opc-ua/>
- [60] RoboDK. Robot Calibration (Laser Tracker). Dato hentet: 04.03.2020. [Internett]. Tilgjengelig: <https://robodk.com/doc/en/Robot-Calibration-LaserTracker.html>
- [61] Universal robots. Programvare. Dato hentet: 22.02.2020. [Internett]. Tilgjengelig: <https://www.universal-robots.com/no/plus/programvare/>
- [62] Kemppi. A7 MIG Welder. Datablad. Dato hentet: 11.02.2020. [Internett]. Tilgjengelig: <https://www.kemppi.com/no/offering/family/a7-mig-welder/pdf/>
- [63] Kemppi. A7 MIG Welder. Dato hentet: 11.02.2020. [Internett]. Tilgjengelig: <https://www.kemppi.com/no/offering/family/a7-mig-welder/>
- [64] Motion controls. Robot life cycle. Dato hentet: 02.03.2020. [Internett]. Tilgjengelig: <https://motioncontrolsrobotics.com/robot-life-cycle-faqs/>
- [65] Joachim Seehusen. Ny sensor for objekt-deteksjon. Dato hentet: 01.04.2020. [Internett]. Tilgjengelig: <https://www.tu.no/artikler/ny-sensor-for-objekt-deteksjon/277099>

- [66] RoboDK. Robot ballbar testing. Dato hentet: 26.04.2020. [Internett] Tilgjengelig: <https://robodk.com/doc/en/Robot-Validation-Ballbar-Test.html>
- [67] RoboDK. Library. Dato hentet: 26.04.2020. [Internett]. Tilgjengelig: <https://robodk.com/library>
- [68] Miller Electric Mfg. LLC. Tips for Troubleshooting Common MIG Weld Defects. Dato hentet: 25.04.2020. [Internett]. Tilgjengelig: <https://www.millerwelds.com/resources/article-library/troubleshooting-weld-defects>
- [69] Mark Crawford. How Many Axes Does Your Robot Need? Dato hentet: 28.04.2020. [Internett]. Tilgjengelig: <https://www.asme.org/topics-resources/content/many-axes-does-robot-need>
- [70] Maske, Tomas (Master thesis). "Heavy multi-pass TIG welding". Tilgjengelig: <https://munin.uit.no/bitstream/handle/10037/11066/thesis.pdf?sequence=2&isAllowed=y>
- [71] Kilde: Valeria Villani; Fabio Pini; Francesco Leali; Cristian Secchi (Article 2018). "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications". Tilgjengelig: <https://www.sciencedirect.com/science/article/abs/pii/S0957415818300321>
- [72] Bennett Brumson. Robotic Simulation and Off-line Programming: From Academia to Industry. Dato hentet: 29.04.2020. [Internett]. Tilgjengelig: https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Featured-Articles/Robotic-Simulation-and-Off-line-Programming-From-Academia-to-Industry/content_id/1825
- [73] IEEE. Unimate. Dato hentet: 28.04.2020. [Internett]. Tilgjengelig: <https://robots.ieee.org/robots/unimate/>
- [74] Robotic Industries Association. UNIMATE. Dato hentet: 28.04.2020. [Internett]. Tilgjengelig: <https://www.robotics.org/joseph-engelberger/unimate.cfm>
- [75] Mathieu Bélanger-Barrette. How Long to Payback my Robot Investment? Dato hentet: 29.04.2020. [Internett]. Tilgjengelig: <https://blog.robotiq.com/how-long-before-i-can-payback-my-robot-investment>
- [76] HTE Technologies. Universal Robots. Dato hentet: 28.04.2020. [Internett]. Tilgjengelig: <https://www.htetechnologies.com/item/index/Universal%20Robots>
- [77] Baumer. Light barriers without reflector – SmartReflect. Dato hentet: 26.04.2020. [Internett]. Tilgjengelig: <https://www.baumer.com/ch/en/product-overview/object-detection/photoelectric-sensors/smartreflect/c/13514>
- [78] Universal Robots A/S.FROM ROBOTS TO COLLABORATIVE ROBOTS - A BRIEF HISTORY. Dato hentet: 29.04.2020. [Internett]. Tilgjengelig: <https://blog.universal-robots.com/in/history-of-cobots>
- [79] GoWelding. What is a Welding Certification? Dato hentet: 29.04.2020. [Internett]. Tilgjengelig: <http://gowelding.org/welding-certification/>
- [80] Vitec AS. KONTROLLMETODER AV SVEIS OG MATERIALER. Dato hentet: 29.04.2020. [Internett]. Tilgjengelig: <https://vitec.as/tjeneste/ndt/kontrollmetoder-av-sveis-og-materialer/>

[81] RoboDK. Robot Machining. Dato hentet: 15.04.2020. [Internett]. Tilgjengelig:
<https://robodk.com/doc/en/Robot-Machining.html#RobotCNC>

[82] ABB. The world's most used offline programming tool for robotics. Dato Hentet:
06.05.2020. [Internett]. Tilgjengelig: <https://new.abb.com/products/robotics/robotstudio>

8. Vedlegg

Vedlegg 1 – Kode for test program bevegelse A til B til C	B
Vedlegg 2 – Kode for test program tegne langs ark/bok/komponent	E
Vedlegg 3 – Kode for simulering av sveiseoperasjon dør ramme	G

Vedlegg 1

```
def TestProgram():  
    # Global parameters:  
    global speed_ms = 0.300  
    global speed_rads = 0.750  
    global accel_mss = 3.000  
    global accel_radss = 1.200  
    global blend_radius_m = 0.001  
  
    #-----  
    # Add any default subprograms here  
    # For example, to drive a gripper as a program call:  
    # def Gripper_Open():  
    # ...  
    # end  
    #  
    # Example to drive a spray gun:  
    def SprayOn(value):  
        # use the value as an output:  
        DO_SPRAY = 5  
        if value == 0:  
            set_standard_digital_out(DO_SPRAY, False)  
        else:  
            set_standard_digital_out(DO_SPRAY, True)  
        end  
    end  
  
    # Example to drive an extruder:  
    def Extruder(value):  
        # use the value as an output:
```

```
if value < 0:
```

```
    # stop extruder
```

```
else:
```

```
    # start extruder
```

```
end
```

```
end
```

```
# Example to move an external axis
```

```
def MoveAxis(value):
```

```
    # use the value as an output:
```

```
    DO_AXIS_1 = 1
```

```
    DI_AXIS_1 = 1
```

```
    if value <= 0:
```

```
        set_standard_digital_out(DO_AXIS_1, False)
```

```
        # Wait for digital input to change state
```

```
        #while (get_standard_digital_in(DI_AXIS_1) != False):
```

```
            # sync()
```

```
        #end
```

```
    else:
```

```
        set_standard_digital_out(DO_AXIS_1, True)
```

```
        # Wait for digital input to change state
```

```
        #while (get_standard_digital_in(DI_AXIS_1) != True):
```

```
            # sync()
```

```
        #end
```

```
    end
```

```
end
```

```
#-----
```

```
# Main program:

# Program generated by RoboDK v4.2.4 for UR5 on 02/05/2020 11:29:37

# Using nominal kinematics.

# set_reference(p[0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000])

movej([0.000000, -1.875382, -0.855133, -0.411078, 1.570796, -
0.000000],accel_radss,speed_rads,0,0)

movej([-0.000000, -2.082466, -1.019019, -0.040108, 1.570796, -
0.000000],accel_radss,speed_rads,0,blend_radius_m)

movej([0.000000, -1.529943, -2.323676, 0.712026, 1.570796, -
0.000000],accel_radss,speed_rads,0,blend_radius_m)

# End of main program

end

TestProgram()
```

Vedlegg 2

```
def Path1():
  # Global parameters:
  global speed_ms = 0.300
  global speed_rads = 0.750
  global accel_mss = 3.000
  global accel_radss = 1.200
  global blend_radius_m = 0.001

  #-----
  # Add any default subprograms here
  # For example, to drive a gripper as a program call:
  # def Gripper_Open():
  # ...
  # end
  #
  # Example to drive a spray gun:
  def SprayOn(value):
    # use the value as an output:
    DO_SPRAY = 5
    if value == 0:
      set_standard_digital_out(DO_SPRAY, False)
    else:
      set_standard_digital_out(DO_SPRAY, True)
    end
  end

  # Example to drive an extruder:
  def Extruder(value):
    # use the value as an output:
    if value < 0:
      # stop extruder
    else:
      # start extruder
    end
  end

  # Example to move an external axis
  def MoveAxis(value):
    # use the value as an output:
    DO_AXIS_1 = 1
    DI_AXIS_1 = 1
    if value <= 0:
      set_standard_digital_out(DO_AXIS_1, False)

      # Wait for digital input to change state
      #while (get_standard_digital_in(DI_AXIS_1) != False):
      # sync()
      #end
    else:
```

```
set_standard_digital_out(DO_AXIS_1, True)

# Wait for digital input to change state
#while (get_standard_digital_in(DI_AXIS_1) != True):
# sync()
#end
end
end
#-----

# Main program:
# Program generated by RoboDK v4.2.2 for UR5 on 25/03/2020 09:32:28
# Using nominal kinematics.
blend_radius_m = 0.001
speed_ms = 1.000
# set_reference(p[0.546579, 0.180519, 0.000000, 1.570796, 0.000000, 0.000000])
set_tcp(p[0.000000, 0.000000, 0.100000, 0.000000, 0.000000, 0.000000])
# Show Tool 1
movej([0.536212, -1.213126, -2.414445, -1.084818, 1.570796,
2.107008],accel_radss,speed_rads,0,0)
movel(p[0.306579, 0.055402, 0.022010, 0.000000, 3.141593, -0.000000],accel_mss,speed_ms,0,0)
speed_ms = 0.010
movel(p[0.546579, 0.055402, 0.022010, 0.000000, 3.141593, -
0.000000],accel_mss,speed_ms,0,blend_radius_m)
speed_ms = 1.000
movel(p[0.546579, 0.055402, 0.122010, -0.000000, 3.141593, -
0.000000],accel_mss,speed_ms,0,blend_radius_m)
# End of main program
end

Path1()
```

Vedlegg 3

```
/PROG MainProgra
/ATTR
OWNER = MNEDITOR;
COMMENT = "RoboDK sequence";
PROG_SIZE = 0;
CREATE = DATE 31-12-14 TIME 12:00:00;
MODIFIED = DATE 31-12-14 TIME 12:00:00;
FILE_NAME = MainProgra;
VERSION = 0;
LINE_COUNT = 49;
MEMORY_SIZE = 0;
PROTECT = READ_WRITE;
TCD: STACK_SIZE = 0,
    TASK_PRIORITY = 50,
    TIME_SLICE = 0,
    BUSY_LAMP_OFF = 0,
    ABORT_REQUEST = 0,
    PAUSE_REQUEST = 0;
DEFAULT_GROUP = 1,*,*,*,*;
CONTROL_CODE = 00000000 00000000;
/MN
1: ! Program generated by ;
2: ! RoboDK v4.2.4 for F ;
3: ! anuc M-710iC/45M on ;
4: ! 03/05/2020 13:29:53 ;
5: ! Using nominal kinema ;
6: ! tics. ;
7: PR[9,1]=1370.000 ;
8: PR[9,2]=-1110.000 ;
```


9: PR[9,3]=-1700.000 ;
10: PR[9,4]=0.000 ;
11: PR[9,5]=0.000 ;
12: PR[9,6]=90.000 ;
13: UFRAME[9]=PR[9] ;
14: UFRAME_NUM=9 ;
15: PR[9,1]=-4.000 ;
16: PR[9,2]=0.000 ;
17: PR[9,3]=371.300 ;
18: PR[9,4]=0.000 ;
19: PR[9,5]=45.000 ;
20: PR[9,6]=0.000 ;
21: UTOOL[9]=PR[9] ;
22: UTOOL_NUM=9 ;
23:J P[1] 20% FINE ;
24:J P[2] 20% FINE ;
25: CALL P1 ;
26: CALL P2 ;
27: CALL P3 ;
28: CALL P4 ;
29:J P[3] 20% FINE ;
30: CALL P5 ;
31:J P[4] 20% FINE ;
32: CALL P6 ;
33:J P[5] 20% FINE ;
34:J P[6] 20% FINE ;
35: WAIT 10.00(sec) ;
36:J P[7] 20% FINE ;
37: CALL P7 ;
38: CALL P8 ;
39:J P[8] 20% FINE ;

40: CALL P9 ;
41: CALL P10 ;
42: CALL P11 ;
43: CALL P12 ;
44:J P[9] 20% FINE ;
45:J P[10] 20% FINE ;
46: CALL P13 ;
47:J P[11] 20% FINE ;
48: CALL P14 ;
49:J P[12] 20% FINE ;

/POS

P[1]{

GP1:

UF : 9, UT : 9,

J1= 0.000 deg, J2= 0.000 deg, J3= 0.000 deg,

J4= 0.000 deg, J5= 0.000 deg, J6= 0.000 deg

};

P[2]{

GP1:

UF : 9, UT : 9,

J1= -83.216 deg, J2= 36.528 deg, J3= -62.463 deg,

J4= -68.242 deg, J5= -124.249 deg, J6= -139.861 deg

};

P[3]{

GP1:

UF : 9, UT : 9,

J1= 7.238 deg, J2= 26.168 deg, J3= 3.079 deg,

J4= 135.688 deg, J5= 14.702 deg, J6= 333.443 deg

};

P[4]{

GP1:

```
UF : 9, UT : 9,  
J1= -26.561 deg, J2= 8.203 deg, J3= -42.799 deg,  
J4= -61.385 deg, J5= -49.777 deg, J6= 115.079 deg  
};  
P[5]{  
  GP1:  
    UF : 9, UT : 9,  
    J1= -9.746 deg, J2= 31.396 deg, J3= -54.612 deg,  
    J4= -111.603 deg, J5= -33.422 deg, J6= -52.596 deg  
};  
P[6]{  
  GP1:  
    UF : 9, UT : 9,  
    J1= -0.000 deg, J2= 0.000 deg, J3= 0.000 deg,  
    J4= 0.000 deg, J5= 0.000 deg, J6= 0.000 deg  
};  
P[7]{  
  GP1:  
    UF : 9, UT : 9,  
    J1= -0.000 deg, J2= 0.000 deg, J3= 0.000 deg,  
    J4= 0.000 deg, J5= 0.000 deg, J6= 0.000 deg  
};  
P[8]{  
  GP1:  
    UF : 9, UT : 9,  
    J1= -78.523 deg, J2= 2.963 deg, J3= -58.169 deg,  
    J4= -88.661 deg, J5= -67.095 deg, J6= -34.133 deg  
};  
P[9]{  
  GP1:  
    UF : 9, UT : 9,
```

```
J1= -17.275 deg, J2= 21.051 deg, J3= -45.404 deg,  
J4= 54.265 deg, J5= -51.534 deg, J6= 80.403 deg  
};  
P[10]{  
  GP1:  
    UF : 9, UT : 9,  
    J1= -22.157 deg, J2= 21.015 deg, J3= -49.561 deg,  
    J4= -56.713 deg, J5= -48.814 deg, J6= 124.142 deg  
};  
P[11]{  
  GP1:  
    UF : 9, UT : 9,  
    J1= -16.157 deg, J2= 22.708 deg, J3= -49.376 deg,  
    J4= -81.176 deg, J5= -40.183 deg, J6= 120.262 deg  
};  
P[12]{  
  GP1:  
    UF : 9, UT : 9,  
    J1= -0.000 deg, J2= 0.000 deg, J3= 0.000 deg,  
    J4= 0.000 deg, J5= 0.000 deg, J6= 0.000 deg  
};  
/END  
/PROG P1  
/ATTR  
OWNER = MNEDITOR;  
COMMENT = "RoboDK sequence";  
PROG_SIZE = 0;  
CREATE = DATE 31-12-14 TIME 12:00:00;  
MODIFIED = DATE 31-12-14 TIME 12:00:00;  
FILE_NAME = P1;  
VERSION = 0;
```

```
LINE_COUNT = 10;
MEMORY_SIZE = 0;
PROTECT = READ_WRITE;
TCD: STACK_SIZE = 0,
    TASK_PRIORITY = 50,
    TIME_SLICE = 0,
    BUSY_LAMP_OFF = 0,
    ABORT_REQUEST = 0,
    PAUSE_REQUEST = 0;
DEFAULT_GROUP = 1,*,*,*,*;
CONTROL_CODE = 00000000 00000000;
/MN
1: UFRAME_NUM=3 ;
2: ! UF3:1370.0,-570.0,-4 ;
3: ! 20.0,180.0,-0.0,90.0 ;
4: ! Show Weld gun ;
5:J P[1] 20% CNT1 ;
6:L P[2] 1000mm/sec CNT1 ;
7:L P[3] 2mm/sec CNT1 ;
8:L P[4] 2mm/sec CNT1 ;
9:L P[5] 2mm/sec CNT1 ;
10:L P[6] 1000mm/sec CNT1 ;
/POS
P[1]{
  GP1:
  UF : 3, UT : 9,
  J1= -77.692 deg, J2= 60.639 deg, J3= -78.932 deg,
  J4= -65.102 deg, J5= -115.987 deg, J6= -123.235 deg
};
P[2]{
  GP1:
```

```
UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
X = -526.025 mm, Y = -652.060 mm, Z = 60.000 mm,
W = -120.353 deg, P = -0.004 deg, R = -90.706 deg
};
P[3]{
  GP1:
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
    X = -526.025 mm, Y = -652.060 mm, Z = -20.000 mm,
    W = -120.353 deg, P = -0.004 deg, R = -90.706 deg
};
P[4]{
  GP1:
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
    X = -546.148 mm, Y = -652.060 mm, Z = -40.000 mm,
    W = -30.000 deg, P = 0.000 deg, R = -90.000 deg
};
P[5]{
  GP1:
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
    X = -466.148 mm, Y = -572.060 mm, Z = -40.000 mm,
    W = -30.000 deg, P = -0.000 deg, R = -90.000 deg
};
P[6]{
  GP1:
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
    X = -466.148 mm, Y = -572.060 mm, Z = -140.000 mm,
    W = -30.000 deg, P = -0.000 deg, R = -90.000 deg
};
/END
/PROG P2
/ATTR
```

```
OWNER = MNEDITOR;
COMMENT = "RoboDK sequence";
PROG_SIZE = 0;
CREATE = DATE 31-12-14 TIME 12:00:00;
MODIFIED = DATE 31-12-14 TIME 12:00:00;
FILE_NAME = P2;
VERSION = 0;
LINE_COUNT = 7;
MEMORY_SIZE = 0;
PROTECT = READ_WRITE;
TCD: STACK_SIZE = 0,
    TASK_PRIORITY = 50,
    TIME_SLICE = 0,
    BUSY_LAMP_OFF = 0,
    ABORT_REQUEST = 0,
    PAUSE_REQUEST = 0;
DEFAULT_GROUP = 1,*,*,*,*;
CONTROL_CODE = 00000000 00000000;
/MN
1: ! Show Weld gun ;
2:J P[1] 20% CNT1 ;
3:L P[2] 1000mm/sec CNT1 ;
4:L P[3] 2mm/sec CNT1 ;
5:L P[4] 2mm/sec CNT1 ;
6:L P[5] 2mm/sec CNT1 ;
7:L P[6] 1000mm/sec CNT1 ;
/POS
P[1]{
GP1:
UF : 3, UT : 9,
J1= -11.683 deg, J2= 41.209 deg, J3= -73.622 deg,
```


J4= -5.019 deg, J5= -92.758 deg, J6= 144.030 deg

};

P[2]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = 428.852 mm, Y = -572.060 mm, Z = 40.000 mm,

W = 126.029 deg, P = 47.067 deg, R = 70.808 deg

};

P[3]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = 428.852 mm, Y = -572.060 mm, Z = -40.000 mm,

W = 126.029 deg, P = 47.067 deg, R = 70.808 deg

};

P[4]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 1',

X = 428.852 mm, Y = -572.060 mm, Z = -40.000 mm,

W = -30.000 deg, P = 40.000 deg, R = -75.000 deg

};

P[5]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 1',

X = 478.852 mm, Y = -622.060 mm, Z = -40.000 mm,

W = -30.000 deg, P = 40.000 deg, R = -75.000 deg

};

P[6]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 1',

X = 478.852 mm, Y = -622.060 mm, Z = -140.000 mm,

W = -30.000 deg, P = 40.000 deg, R = -75.000 deg

O

```
};  
  
/END  
  
/PROG P3  
  
/ATTR  
  
OWNER = MNEDITOR;  
  
COMMENT = "RoboDK sequence";  
  
PROG_SIZE = 0;  
  
CREATE = DATE 31-12-14 TIME 12:00:00;  
  
MODIFIED = DATE 31-12-14 TIME 12:00:00;  
  
FILE_NAME = P3;  
  
VERSION = 0;  
  
LINE_COUNT = 5;  
  
MEMORY_SIZE = 0;  
  
PROTECT = READ_WRITE;  
  
TCD: STACK_SIZE = 0,  
      TASK_PRIORITY = 50,  
      TIME_SLICE = 0,  
      BUSY_LAMP_OFF = 0,  
      ABORT_REQUEST = 0,  
      PAUSE_REQUEST = 0;  
  
DEFAULT_GROUP = 1,*,*,*,*;  
  
CONTROL_CODE = 00000000 00000000;  
  
/MN  
  
1: ! Show Weld gun ;  
  
2:J P[1] 20% CNT1 ;  
  
3:L P[2] 1000mm/sec CNT1 ;  
  
4:L P[3] 2mm/sec CNT1 ;  
  
5:L P[4] 1000mm/sec CNT1 ;  
  
/POS  
  
P[1]{  
  
GP1:
```

```
UF : 3, UT : 9,  
J1= -21.153 deg, J2= 41.029 deg, J3= -56.466 deg,  
J4= 104.871 deg, J5= -33.033 deg, J6= -124.632 deg  
};  
P[2]{  
  GP1:  
  UF : 3, UT : 9,   CONFIG : 'N U T, 0, 0, 0',  
  X = -446.148 mm, Y = 562.940 mm, Z = 40.000 mm,  
  W = -121.105 deg, P = 78.667 deg, R = 24.027 deg  
};  
P[3]{  
  GP1:  
  UF : 3, UT : 9,   CONFIG : 'N U T, 0, 0, 0',  
  X = -446.148 mm, Y = 562.940 mm, Z = -40.000 mm,  
  W = -121.105 deg, P = 78.667 deg, R = 24.027 deg  
};  
P[4]{  
  GP1:  
  UF : 3, UT : 9,   CONFIG : 'N U T, 0, 0, 0',  
  X = -446.764 mm, Y = 462.944 mm, Z = -39.384 mm,  
  W = -121.105 deg, P = 78.667 deg, R = 24.027 deg  
};  
/END  
/PROG P4  
/ATTR  
OWNER = MNEDITOR;  
COMMENT = "RoboDK sequence";  
PROG_SIZE = 0;  
CREATE = DATE 31-12-14 TIME 12:00:00;  
MODIFIED = DATE 31-12-14 TIME 12:00:00;  
FILE_NAME = P4;
```

```
VERSION = 0;

LINE_COUNT = 5;

MEMORY_SIZE = 0;

PROTECT = READ_WRITE;

TCD: STACK_SIZE = 0,
    TASK_PRIORITY = 50,
    TIME_SLICE = 0,
    BUSY_LAMP_OFF = 0,
    ABORT_REQUEST = 0,
    PAUSE_REQUEST = 0;

DEFAULT_GROUP = 1,*,*,*,*;

CONTROL_CODE = 00000000 00000000;

/MN
1: ! Show Weld gun ;
2:J P[1] 20% CNT1 ;
3:L P[2] 1000mm/sec CNT1 ;
4:L P[3] 2mm/sec CNT1 ;
5:L P[4] 1000mm/sec CNT1 ;

/POS
P[1]{
    GP1:
        UF : 3, UT : 9,
        J1= -25.161 deg, J2= 35.066 deg, J3= -52.510 deg,
        J4= 150.628 deg, J5= -21.062 deg, J6= 3.173 deg
};

P[2]{
    GP1:
        UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
        X = -446.148 mm, Y = 562.940 mm, Z = -40.000 mm,
        W = -15.000 deg, P = -10.000 deg, R = -40.000 deg
};
```

```
P[3]{
  GP1:
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
    X = -526.148 mm, Y = 642.940 mm, Z = -40.000 mm,
    W = -15.000 deg, P = -10.000 deg, R = -40.000 deg
};

P[4]{
  GP1:
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
    X = -526.148 mm, Y = 642.940 mm, Z = -140.000 mm,
    W = -15.000 deg, P = -10.000 deg, R = -40.000 deg
};

/END

/PROG P5

/ATTR

OWNER   = MNEDITOR;

COMMENT = "RoboDK sequence";

PROG_SIZE = 0;

CREATE   = DATE 31-12-14 TIME 12:00:00;

MODIFIED = DATE 31-12-14 TIME 12:00:00;

FILE_NAME = P5;

VERSION   = 0;

LINE_COUNT = 5;

MEMORY_SIZE = 0;

PROTECT   = READ_WRITE;

TCD: STACK_SIZE = 0,
    TASK_PRIORITY = 50,
    TIME_SLICE = 0,
    BUSY_LAMP_OFF = 0,
    ABORT_REQUEST = 0,
    PAUSE_REQUEST = 0;
```

DEFAULT_GROUP = 1,*,*,*,*;

CONTROL_CODE = 00000000 00000000;

/MN

1: ! Show Weld gun ;

2:J P[1] 20% CNT1 ;

3:L P[2] 1000mm/sec CNT1 ;

4:L P[3] 2mm/sec CNT1 ;

5:L P[4] 1000mm/sec CNT1 ;

/POS

P[1]{

GP1:

UF : 3, UT : 9,

J1= -14.792 deg, J2= 34.385 deg, J3= -58.866 deg,

J4= -84.412 deg, J5= -37.816 deg, J6= 134.010 deg

};

P[2]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = 428.852 mm, Y = 562.940 mm, Z = 40.000 mm,

W = -90.176 deg, P = 89.502 deg, R = -45.528 deg

};

P[3]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = 428.852 mm, Y = 562.940 mm, Z = -40.000 mm,

W = -90.176 deg, P = 89.502 deg, R = -45.528 deg

};

P[4]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = 428.236 mm, Y = 462.944 mm, Z = -39.384 mm,

T

```
W = -90.176 deg, P = 89.502 deg, R = -45.528 deg
};
/END
/PROG P6
/ATTR
OWNER = MNEDITOR;
COMMENT = "RoboDK sequence";
PROG_SIZE = 0;
CREATE = DATE 31-12-14 TIME 12:00:00;
MODIFIED = DATE 31-12-14 TIME 12:00:00;
FILE_NAME = P6;
VERSION = 0;
LINE_COUNT = 5;
MEMORY_SIZE = 0;
PROTECT = READ_WRITE;
TCD: STACK_SIZE = 0,
    TASK_PRIORITY = 50,
    TIME_SLICE = 0,
    BUSY_LAMP_OFF = 0,
    ABORT_REQUEST = 0,
    PAUSE_REQUEST = 0;
DEFAULT_GROUP = 1,*,*,*,*;
CONTROL_CODE = 00000000 00000000;
/MN
1: ! Show Weld gun ;
2:J P[1] 20% CNT1 ;
3:L P[2] 1000mm/sec CNT1 ;
4:L P[3] 2mm/sec CNT1 ;
5:L P[4] 1000mm/sec CNT1 ;
/POS
P[1]{
```


GP1:

UF : 3, UT : 9,

J1= -13.109 deg, J2= 29.201 deg, J3= -56.065 deg,

J4= -109.827 deg, J5= -35.724 deg, J6= -51.394 deg

};

P[2]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = 428.852 mm, Y = 562.940 mm, Z = -40.000 mm,

W = -10.000 deg, P = -10.000 deg, R = -110.000 deg

};

P[3]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = 508.852 mm, Y = 642.940 mm, Z = -40.000 mm,

W = -10.000 deg, P = -10.000 deg, R = -110.000 deg

};

P[4]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = 508.852 mm, Y = 642.940 mm, Z = -140.000 mm,

W = -10.000 deg, P = -10.000 deg, R = -110.000 deg

};

/END

/PROG P7

/ATTR

OWNER = MNEDITOR;

COMMENT = "RoboDK sequence";

PROG_SIZE = 0;

CREATE = DATE 31-12-14 TIME 12:00:00;

V

MODIFIED = DATE 31-12-14 TIME 12:00:00;

FILE_NAME = P7;

VERSION = 0;

LINE_COUNT = 8;

MEMORY_SIZE = 0;

PROTECT = READ_WRITE;

TCD: STACK_SIZE = 0,

TASK_PRIORITY = 50,

TIME_SLICE = 0,

BUSY_LAMP_OFF = 0,

ABORT_REQUEST = 0,

PAUSE_REQUEST = 0;

DEFAULT_GROUP = 1,*,*,*,*;

CONTROL_CODE = 00000000 00000000;

/MN

1: UFRAME_NUM=3 ;

2: ! UF3:1370.0,-570.0,-4 ;

3: ! 20.0,0.0,0.0,90.0 ;

4: ! Show Weld gun ;

5:J P[1] 20% CNT1 ;

6:L P[2] 1000mm/sec CNT1 ;

7:L P[3] 2mm/sec CNT1 ;

8:L P[4] 1000mm/sec CNT1 ;

/POS

P[1]{

GP1:

UF : 3, UT : 9,

J1= -82.029 deg, J2= 26.778 deg, J3= -74.940 deg,

J4= -81.549 deg, J5= -60.714 deg, J6= 164.369 deg

};

P[2]{

W

```
GP1:
  UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
  X = -526.148 mm, Y = 642.940 mm, Z = -40.000 mm,
  W = 5.795 deg, P = -75.315 deg, R = 93.003 deg
};

P[3]{
  GP1:
  UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
  X = -526.148 mm, Y = 642.940 mm, Z = 40.000 mm,
  W = 5.795 deg, P = -75.315 deg, R = 93.003 deg
};

P[4]{
  GP1:
  UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
  X = -524.300 mm, Y = 742.921 mm, Z = 40.616 mm,
  W = 5.795 deg, P = -75.315 deg, R = 93.003 deg
};

/END
/PROG P8
/ATTR
OWNER = MNEDITOR;
COMMENT = "RoboDK sequence";
PROG_SIZE = 0;
CREATE = DATE 31-12-14 TIME 12:00:00;
MODIFIED = DATE 31-12-14 TIME 12:00:00;
FILE_NAME = P8;
VERSION = 0;
LINE_COUNT = 5;
MEMORY_SIZE = 0;
PROTECT = READ_WRITE;
TCD: STACK_SIZE = 0,
```

```
TASK_PRIORITY = 50,
TIME_SLICE = 0,
BUSY_LAMP_OFF = 0,
ABORT_REQUEST = 0,
PAUSE_REQUEST = 0;
DEFAULT_GROUP = 1,*,*,*,*;
CONTROL_CODE = 00000000 00000000;
/MN
1: ! Show Weld gun ;
2:J P[1] 20% CNT1 ;
3:L P[2] 1000mm/sec CNT1 ;
4:L P[3] 2mm/sec CNT1 ;
5:L P[4] 1000mm/sec CNT1 ;
/POS
P[1]{
GP1:
UF : 3, UT : 9,
J1= -79.037 deg, J2= 20.246 deg, J3= -71.673 deg,
J4= -93.851 deg, J5= -67.613 deg, J6= -19.503 deg
};
P[2]{
GP1:
UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',
X = -526.148 mm, Y = 642.940 mm, Z = 40.000 mm,
W = 180.000 deg, P = 25.000 deg, R = 90.000 deg
};
P[3]{
GP1:
UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',
X = -446.148 mm, Y = 562.940 mm, Z = 40.000 mm,
W = -180.000 deg, P = 25.000 deg, R = 90.000 deg
```

```
};  
P[4]{  
  GP1:  
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',  
    X = -446.148 mm, Y = 562.940 mm, Z = 140.000 mm,  
    W = 180.000 deg, P = 25.000 deg, R = 90.000 deg  
};  
/END  
/PROG P9  
/ATTR  
OWNER   = MNEDITOR;  
COMMENT = "RoboDK sequence";  
PROG_SIZE = 0;  
CREATE   = DATE 31-12-14 TIME 12:00:00;  
MODIFIED = DATE 31-12-14 TIME 12:00:00;  
FILE_NAME = P9;  
VERSION   = 0;  
LINE_COUNT = 5;  
MEMORY_SIZE = 0;  
PROTECT   = READ_WRITE;  
TCD: STACK_SIZE = 0,  
    TASK_PRIORITY = 50,  
    TIME_SLICE = 0,  
    BUSY_LAMP_OFF = 0,  
    ABORT_REQUEST = 0,  
    PAUSE_REQUEST = 0;  
DEFAULT_GROUP = 1,*,*,*;  
CONTROL_CODE = 00000000 00000000;  
/MN  
1: ! Show Weld gun ;  
2: J P[1] 20% CNT1 ;
```

3:L P[2] 1000mm/sec CNT1 ;

4:L P[3] 2mm/sec CNT1 ;

5:L P[4] 1000mm/sec CNT1 ;

/POS

P[1]{

GP1:

UF : 3, UT : 9,

J1= 38.104 deg, J2= 21.686 deg, J3= -85.635 deg,

J4= 58.016 deg, J5= -65.471 deg, J6= -164.809 deg

};

P[2]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = 508.852 mm, Y = 642.940 mm, Z = -40.000 mm,

W = 29.610 deg, P = -69.750 deg, R = -15.124 deg

};

P[3]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = 508.852 mm, Y = 642.940 mm, Z = 40.000 mm,

W = 29.610 deg, P = -69.750 deg, R = -15.124 deg

};

P[4]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = 608.820 mm, Y = 640.477 mm, Z = 40.616 mm,

W = 29.610 deg, P = -69.750 deg, R = -15.124 deg

};

/END

/PROG P10

/ATTR

Æ

```
OWNER = MNEDITOR;
COMMENT = "RoboDK sequence";
PROG_SIZE = 0;
CREATE = DATE 31-12-14 TIME 12:00:00;
MODIFIED = DATE 31-12-14 TIME 12:00:00;
FILE_NAME = P10;
VERSION = 0;
LINE_COUNT = 5;
MEMORY_SIZE = 0;
PROTECT = READ_WRITE;
TCD: STACK_SIZE = 0,
    TASK_PRIORITY = 50,
    TIME_SLICE = 0,
    BUSY_LAMP_OFF = 0,
    ABORT_REQUEST = 0,
    PAUSE_REQUEST = 0;
DEFAULT_GROUP = 1,*,*,*,*;
CONTROL_CODE = 00000000 00000000;
/MN
1: ! Show Weld gun ;
2:J P[1] 20% CNT1 ;
3:L P[2] 1000mm/sec CNT1 ;
4:L P[3] 2mm/sec CNT1 ;
5:L P[4] 1000mm/sec CNT1 ;
/POS
P[1]{
GP1:
UF : 3, UT : 9,
J1= -13.120 deg, J2= -15.158 deg, J3= -48.929 deg,
J4= -13.755 deg, J5= -52.937 deg, J6= -19.354 deg
};
```



```
P[2]{
  GP1:
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
    X = 508.852 mm, Y = 642.940 mm, Z = 40.000 mm,
    W = -175.000 deg, P = -30.000 deg, R = -135.000 deg
};

P[3]{
  GP1:
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
    X = 428.852 mm, Y = 562.940 mm, Z = 40.000 mm,
    W = -175.000 deg, P = -30.000 deg, R = -135.000 deg
};

P[4]{
  GP1:
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
    X = 428.852 mm, Y = 562.940 mm, Z = 140.000 mm,
    W = -175.000 deg, P = -30.000 deg, R = -135.000 deg
};

/END

/PROG P11

/ATTR

OWNER   = MNEDITOR;
COMMENT = "RoboDK sequence";
PROG_SIZE = 0;
CREATE   = DATE 31-12-14 TIME 12:00:00;
MODIFIED = DATE 31-12-14 TIME 12:00:00;
FILE_NAME = P11;
VERSION   = 0;
LINE_COUNT = 5;
MEMORY_SIZE = 0;
PROTECT   = READ_WRITE;
```

```
TCD: STACK_SIZE = 0,
      TASK_PRIORITY = 50,
      TIME_SLICE = 0,
      BUSY_LAMP_OFF = 0,
      ABORT_REQUEST = 0,
      PAUSE_REQUEST = 0;
DEFAULT_GROUP = 1,*,*,*,*;
CONTROL_CODE = 00000000 00000000;

/MN
1: ! Show Weld gun ;
2:J P[1] 20% CNT1 ;
3:L P[2] 1000mm/sec CNT1 ;
4:L P[3] 2mm/sec CNT1 ;
5:L P[4] 1000mm/sec CNT1 ;

/POS
P[1]{
  GP1:
    UF : 3, UT : 9,
    J1= -27.225 deg, J2= 38.390 deg, J3= -58.955 deg,
    J4= 162.640 deg, J5= -21.653 deg, J6= -177.644 deg
};
P[2]{
  GP1:
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
    X = -446.148 mm, Y = -572.060 mm, Z = -40.000 mm,
    W = -46.331 deg, P = -80.295 deg, R = 95.176 deg
};
P[3]{
  GP1:
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
    X = -446.148 mm, Y = -572.060 mm, Z = 40.000 mm,
```

```
W = -46.331 deg, P = -80.295 deg, R = 95.176 deg
};
P[4]{
  GP1:
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
    X = -444.300 mm, Y = -472.079 mm, Z = 40.616 mm,
    W = -46.331 deg, P = -80.295 deg, R = 95.176 deg
};
/END
/PROG P12
/ATTR
OWNER   = MNEDITOR;
COMMENT = "RoboDK sequence";
PROG_SIZE = 0;
CREATE   = DATE 31-12-14 TIME 12:00:00;
MODIFIED = DATE 31-12-14 TIME 12:00:00;
FILE_NAME = P12;
VERSION   = 0;
LINE_COUNT = 5;
MEMORY_SIZE = 0;
PROTECT   = READ_WRITE;
TCD: STACK_SIZE = 0,
    TASK_PRIORITY = 50,
    TIME_SLICE = 0,
    BUSY_LAMP_OFF = 0,
    ABORT_REQUEST = 0,
    PAUSE_REQUEST = 0;
DEFAULT_GROUP = 1,*,*,*,*;
CONTROL_CODE = 00000000 00000000;
/MN
1: ! Show Weld gun ;
```

2:J P[1] 20% CNT1 ;

3:L P[2] 1000mm/sec CNT1 ;

4:L P[3] 2mm/sec CNT1 ;

5:L P[4] 1000mm/sec CNT1 ;

/POS

P[1]{

GP1:

UF : 3, UT : 9,

J1= -21.109 deg, J2= 39.617 deg, J3= -57.698 deg,

J4= 140.081 deg, J5= -45.290 deg, J6= 40.076 deg

};

P[2]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = -446.148 mm, Y = -572.060 mm, Z = 40.000 mm,

W = -160.000 deg, P = 25.000 deg, R = 55.000 deg

};

P[3]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = -526.148 mm, Y = -652.060 mm, Z = 40.000 mm,

W = -160.000 deg, P = 25.000 deg, R = 55.000 deg

};

P[4]{

GP1:

UF : 3, UT : 9, CONFIG : 'N U T, 0, 0, 0',

X = -526.148 mm, Y = -652.060 mm, Z = 140.000 mm,

W = -160.000 deg, P = 25.000 deg, R = 55.000 deg

};

/END

/PROG P13

/ATTR

```
OWNER = MNEDITOR;
COMMENT = "RoboDK sequence";
PROG_SIZE = 0;
CREATE = DATE 31-12-14 TIME 12:00:00;
MODIFIED = DATE 31-12-14 TIME 12:00:00;
FILE_NAME = P13;
VERSION = 0;
LINE_COUNT = 5;
MEMORY_SIZE = 0;
PROTECT = READ_WRITE;
TCD: STACK_SIZE = 0,
    TASK_PRIORITY = 50,
    TIME_SLICE = 0,
    BUSY_LAMP_OFF = 0,
    ABORT_REQUEST = 0,
    PAUSE_REQUEST = 0;
DEFAULT_GROUP = 1,*,*,*,*;
CONTROL_CODE = 00000000 00000000;
```

/MN

```
1: ! Show Weld gun ;
2:J P[1] 20% CNT1 ;
3:L P[2] 1000mm/sec CNT1 ;
4:L P[3] 2mm/sec CNT1 ;
5:L P[4] 1000mm/sec CNT1 ;
```

/POS

P[1]{

GP1:

UF : 3, UT : 9,

J1= -15.007 deg, J2= 34.699 deg, J3= -60.550 deg,

J4= -93.641 deg, J5= -39.221 deg, J6= 142.863 deg

DD

```
};  
P[2]{  
  GP1:  
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',  
    X = 428.852 mm, Y = -572.060 mm, Z = -40.000 mm,  
    W = 32.074 deg, P = -85.233 deg, R = 101.817 deg  
};  
P[3]{  
  GP1:  
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',  
    X = 428.852 mm, Y = -572.060 mm, Z = 40.000 mm,  
    W = 32.074 deg, P = -85.233 deg, R = 101.817 deg  
};  
P[4]{  
  GP1:  
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',  
    X = 430.700 mm, Y = -472.079 mm, Z = 40.616 mm,  
    W = 32.074 deg, P = -85.233 deg, R = 101.817 deg  
};  
/END  
/PROG P14  
/ATTR  
OWNER = MNEDITOR;  
COMMENT = "RoboDK sequence";  
PROG_SIZE = 0;  
CREATE = DATE 31-12-14 TIME 12:00:00;  
MODIFIED = DATE 31-12-14 TIME 12:00:00;  
FILE_NAME = P14;  
VERSION = 0;  
LINE_COUNT = 5;  
MEMORY_SIZE = 0;
```

```
PROTECT = READ_WRITE;

TCD: STACK_SIZE = 0,
    TASK_PRIORITY = 50,
    TIME_SLICE = 0,
    BUSY_LAMP_OFF = 0,
    ABORT_REQUEST = 0,
    PAUSE_REQUEST = 0;

DEFAULT_GROUP = 1,*,*,*,*;

CONTROL_CODE = 00000000 00000000;

/MN

1: ! Show Weld gun ;
2:J P[1] 20% CNT1 ;
3:L P[2] 1000mm/sec CNT1 ;
4:L P[3] 2mm/sec CNT1 ;
5:L P[4] 1000mm/sec CNT1 ;

/POS

P[1]{
    GP1:
        UF : 3, UT : 9,
        J1= -6.785 deg, J2= 27.360 deg, J3= -59.754 deg,
        J4= -161.395 deg, J5= -30.813 deg, J6= -16.244 deg
};

P[2]{
    GP1:
        UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
        X = 428.852 mm, Y = -572.060 mm, Z = 40.000 mm,
        W = 175.000 deg, P = 15.000 deg, R = 90.000 deg
};

P[3]{
    GP1:
        UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',
```

```
X = 478.852 mm, Y = -622.060 mm, Z = 40.000 mm,  
W = 175.000 deg, P = 15.000 deg, R = 90.000 deg  
};  
P[4]{  
  GP1:  
    UF : 3, UT : 9,    CONFIG : 'N U T, 0, 0, 0',  
    X = 478.852 mm, Y = -622.060 mm, Z = 140.000 mm,  
    W = 175.000 deg, P = 15.000 deg, R = 90.000 deg  
};  
/END
```