

BACHELOROPPGAVE

---

**Robotisert rennerensk**

---

*Skrevet av:*

KIM HANS FIMREITE  
TØRRES LITLABØ BUNES

Juni 2022

## Forord

Bacheloroppgaven er et resultat av en god del timers arbeid. Motivasjonen har kommet av snart tre hele år som studenter i Haugesund, der det er blitt dyrket samt videreutviklet interesse innen for temaene tekniske løsninger, data, PLS programmering og robotikk. Vi valgte å samarbeide for å få en oppgave med stor faglig bredde og for å kunne utfylle hverandres styrker og svakheter. Da vi fikk høre om oppgaven, ble det dannet en umiddelbar interesse for begge, da oppgaven omhandler flere tema og fag som vi tidligere i studie-sammenheng har grunnleggende kjennskap til. Dette var tema som: PLS programmering, robotikk og prosjekthåndtering.

Vi ønsker å gi en stor takk til våre veiledere, Johannes Møgster fra Goodtech og Mahdi Ghane fra HVL Haugesund for en uvurderlig hjelp. Tusen takk for god hjelp, innspill og oppmuntring gjennom hele semesteret.

## Sammendrag

Norsk industri består hovedsaklig av råvareproduksjon som oljeutvinning, gruvedrift og smelteverksindustri. Dette er næringer som i liten grad har tatt i bruk robotteknologi. Norge ligger under gjennomsnittet i verden med hensyn til antall bruk av industriroboter, dette målt mot hvor mange industriarbeidere det er i landet. Bruk av roboter for å automatisere arbeidsprosesser har vært stigende senere år.

Goodtech AS er et norsk teknologikonsern innen elektro- og automatiseringsløsninger, med økende satsing på bruk av roboter. Goodtech har en kunde som ønsker å automatisere fjerning av slaggrester ved hjelp av en industrirobot i en produksjonsrenne ved et smelteverk. Denne prosessen blir i dag utført manuelt av operatører ved bruk av spett eller slag verktøy. Bruk av en robot kan få ned HMS risiko blant operatørene.

Det er i denne oppgaven er det undersøkt om ulike industriroboter kan være med på å automatisere denne prosessen, samt bidra med å senke belastningen og sikkerhetsrisikoen for arbeideren.

I prosjektet er det utført en simulering av fjerning av slagg med Siemens PLS, simulert HMI-panel og UR5e robot sammen med whiteboard-visker som et arbeidsverktøy. Videre er maskinvaren koblet over PROFINET kommunikasjon mellom overnømte enhetene. Det er også laget en anbefaling av industrirobot som kan passe til smelteverket, samt en anbefaling av beskyttelsesutstyr til supplement ved bruk i varme omgivelser.

# Innholdsfortegnelse

<b>Forord</b>	<b>i</b>
<b>Sammendrag</b>	<b>ii</b>
<b>Figurer</b>	<b>vi</b>
<b>Tabeller</b>	<b>vii</b>
<b>Forkortelser og begrep</b>	<b>viii</b>
<b>1 Innledning</b>	<b>1</b>
1.1 Bakgrunn . . . . .	1
1.2 Formål . . . . .	2
1.3 Avgrensinger . . . . .	2
1.4 Slaggrester ved smelting av ilmenitt . . . . .	3
1.5 Goodtech AS . . . . .	4
1.6 TiZir Titanium & Iron AS . . . . .	4
1.7 Industri 4.0 . . . . .	5
<b>2 Bakgrunn</b>	<b>6</b>
2.1 Valggrunnlag til robotarmen . . . . .	6
2.2 Robotarm . . . . .	6
2.3 Industrirobot leverandører . . . . .	8
2.4 Presentering av industriroboter . . . . .	8
2.4.1 ABB . . . . .	8
2.4.2 ABB IRB . . . . .	8
2.4.3 KUKA . . . . .	9
2.4.4 KUKA QUANTEC . . . . .	10
2.4.5 FANUC . . . . .	11
2.4.6 FANUC FoundryPRO . . . . .	11
2.4.7 Sammendrag av industriroboter . . . . .	13
2.5 Behandling av varmeutfordringer . . . . .	13
2.5.1 Varmekåpe . . . . .	13
2.5.2 Aktiv luftkjøling . . . . .	14
2.5.3 Eksponeringstid . . . . .	14
2.6 HMS- Helse miljø og sikkerhet . . . . .	14
2.6.1 Risikovurdering . . . . .	15
2.7 Fysisk sikkerhetsbarriere til robot . . . . .	15
2.8 Vedlikehold . . . . .	17
<b>3 Innfallsvinkel</b>	<b>18</b>
3.1 PLS - Programmerbar logisk styring . . . . .	18
3.2 Programvarer . . . . .	19
3.2.1 Siemens TIA Portal . . . . .	19

3.2.2	SIMATIC Robot Integrator . . . . .	19
3.2.3	SIMATIC Robot Library . . . . .	20
3.2.4	Sammenkobling av SIMATIC Robot Integrator og SIMATIC Robot Library . . . . .	21
3.2.5	PROFINET . . . . .	21
3.2.6	SocketTest . . . . .	21
3.3	Deling av programkode . . . . .	22
3.3.1	GitHub i TIA Portal . . . . .	22
3.3.2	TIA Portal multiuser engineering . . . . .	22
3.4	Prototype . . . . .	23
3.4.1	Prototypens byggverk . . . . .	23
3.4.2	Universal robot . . . . .	24
3.4.3	Robotens arbeidsverktøy . . . . .	25
3.4.4	PolyScope . . . . .	26
3.4.5	Elektrisk grensesnitt . . . . .	26
3.4.6	Siemens PLS - SIMATIC S7-1500 . . . . .	27
3.4.7	HMI-panel . . . . .	28
3.4.8	Flytdiagram for utførelse av robot . . . . .	28
<b>4</b>	<b>Resultat</b>	<b>30</b>
4.1	Prototypens virkemåte . . . . .	30
4.1.1	UR5e testing . . . . .	30
4.1.2	UR5e med HMI . . . . .	31
4.2	Anbefaling av programvare . . . . .	32
4.3	Industrirobot, utstyr og komponenter for anlegget ved TTI . . . . .	32
4.3.1	ABB IRB 6700-175 . . . . .	33
4.3.2	Beskyttelsesutstyr og komponent til ABB IRB6700-175 . . . . .	33
4.3.3	Vedlikehold av ABB IRB6700-175 . . . . .	33
4.3.4	Kundekontakt med bedrifter . . . . .	33
4.4	Prisantydning . . . . .	34
<b>5</b>	<b>Diskusjon</b>	<b>35</b>
5.1	Gripper og UR5e robot . . . . .	35
5.2	Kommunikasjonsprotokoll . . . . .	35
5.3	Utfordringer . . . . .	35
5.4	Videre arbeid . . . . .	36
5.4.1	Utvikling av programkode . . . . .	36
5.4.2	Arbeidsverktøy . . . . .	36
5.4.3	Håndholdt kontroller . . . . .	36
<b>6</b>	<b>Konklusjon</b>	<b>37</b>
	<b>Kilder</b>	<b>38</b>
<b>A</b>	<b>PROFINET kommunikasjon</b>	<b>A-1</b>

---

<b>B Kode for sammenkobling</b>	<b>B-8</b>
<b>C Kode for UR5e robot</b>	<b>C-9</b>
<b>D Funksjonell designspesifikasjon</b>	<b>D-10</b>
D.1 Relaterte dokumenter . . . . .	D-10
D.2 HMI Design . . . . .	D-10
D.3 UR5e modusfunksjoner . . . . .	D-10
D.4 Robot . . . . .	D-11
D.5 På . . . . .	D-11
D.6 Av . . . . .	D-11
D.7 Valg av programmer . . . . .	D-11
D.8 Full . . . . .	D-11
D.9 Bunn . . . . .	D-11
D.10 Topp . . . . .	D-11

## Figurer

Figur 1.1	Slagg tapping [2]	1
Figur 1.2	Nedskalert prototype ved LAB HVL Haugesund	2
Figur 1.3	UR5e robot [3]	3
Figur 1.4	Eksempel på slaggest [4]	4
Figur 1.5	Illustrasjon av industri 4.0 [8]	5
Figur 2.1	2d skisse av frihetsgrader [9]	6
Figur 2.2	Illustrasjon av industriroboter i arbeid [11]	7
Figur 2.3	ABB industrirobot IRB-serie [15]	8
Figur 2.4	ABB IRB med LEAN-ID og foundry plus 2 [16]	9
Figur 2.5	KR 120 R2700-2 [20]	10
Figur 2.6	Illustrasjon av kapslingsgrad på M710iC/70	11
Figur 2.7	Fanuc M-710 Series [25]	12
Figur 2.8	Illustrasjon av varmekåpe på en industrirobot [29]	13
Figur 2.9	Illustrasjon om hvordan avsperring kan se ut [30]	15
Figur 2.10	Axelent X-Guard sikkerhetsbarriere produkter [32]	16
Figur 2.11	Forslag sikkerhetsbarrierer	16
Figur 3.1	Blokkskjema av en prosedyre til en PLS	18
Figur 3.2	Programvare i TIA Portal [34]	19
Figur 3.3	Illustrasjon av SIMATIC HMI Mobile Panel sammen med Robot Integrator [36]	20
Figur 3.4	Illustrasjon av bruken av SIMATIC Robot Intregator [36]	20
Figur 3.5	Leverandører som samarbeider med Siemens angående SIMATIC Robotic Library [36]	21
Figur 3.6	Illustrasjon av funksjonsdeling av GIT	22
Figur 3.7	Nedskalert prototype for simulering	23
Figur 3.8	Universal Robots e-serie [39]	24
Figur 3.9	Prototypens gripper	25
Figur 3.10	Prototypens verktøy	25
Figur 3.11	Universal Robots - kontrollenhet og "Teach pendant"	26
Figur 3.12	Kontrollenhetens grensesnitt for elektrisk tilkobling	27
Figur 3.13	Kontrollenhetens grensesnitt for tilkoblinger [42]	27
Figur 3.14	Flytdiagram for fjerning av slaggester	29
Figur 4.1	Rennerensk av renne uten fjøring	31
Figur 4.2	Rennerensk av renne med fjøring	31
Figur 4.3	Simulert HMI-panel	31
Figur 4.4	IRB 6700-175 [16]	32
Figur A.1	Navn på ulike datatyper [44]	A-1
Figur A.2	Tag tabell som gir informasjon om UR5e	A-2
Figur A.3	Informasjons overføring fra UR5e til tag tabell over PROFINET	A-2
Figur A.4	Beskrivelse av datastrukturen [44]	A-3
Figur A.5	Beskrivelse av datastrukturen [44]	A-3
Figur A.6	Beskrivelse av datastrukturen [44]	A-3
Figur A.7	Beskrivelse av datastrukturen [44]	A-4

Figur A.8	Beskrivelse av datastrukturen [44]	A-4
Figur A.9	Beskrivelse av datastrukturen [44]	A-4
Figur A.10	Beskrivelse av datastrukturen [44]	A-5
Figur A.11	Beskrivelse av datastrukturen [44]	A-5
Figur A.12	Beskrivelse av datastrukturen [44]	A-5
Figur A.13	Beskrivelse av datastrukturen [44]	A-6
Figur A.14	Datablokk 100	A-6
Figur A.15	Datablokk 101	A-7
Figur B.1	Kode for sammenkobling av datablokker og UR5e tags	B-8
Figur C.1	PolyScope UR5e kode	C-9
Figur D.1	HMI-panel	D-10

## Tabeller

Tabell 2.1	Datablad fra ABB sine roboter[18]	9
Tabell 2.2	Datablad fra KUKA sine roboter.[21]	10
Tabell 2.3	Industrielle roboter fra FANUC	12
Tabell 2.4	Potensielle uønskede hendelser under drift	15
Tabell 2.5	Pris til forslag av sikkerhetsbarrierer	17
Tabell 4.1	Utdrag av vedlikeholdsrutiner ABB IRB6700 [16]	33
Tabell 4.2	Prisantydning til forelått løsning	34



## Forkortelser og begrep

PLS	Programerbar logisk styring
UR	Universal robots
DOF	Frihetsgrader
HMS	Helse miljø og sikkerhet
HVL	Høgskulen på Vestland
GitHub	Deling av programkode
TTI	TiZir Titanium and Iron AS
Lean-ID	Lean integrated dressing
MTBF	Mean time between failure
UR5	Universal robots 5
KRL	KUKA robot language
PASCAL	Programmeringsspråk som støtter strukturert programmering og datastruktur
TIA	Totally Integrated Portal
Foundry robot	Robot som er designet for bruk i tøffe omgivelser
IoT	Internet of Things
IP grad	Kapslingsgraden til elektrisk materiell. Beskriver utstyret sin beskyttelse mot inntrenging av faste gjenstander og vann
GUI	Grafisk brukergrensesnitt
UDT	User defined datatypes
Teach pendant	Kontrollboks for programmering av robot
ISO	Den internasjonale standardiseringsorganisasjonen



# 1 Innledning

Goodtech AS har en kunde som ønsker å automatisere prosessen med å fjerne slaggrester fra en renne under produksjonen i et smelteverk. Denne prosessen blir per i dag utført manuelt. Kunden er TiZir Tyssedal. TiZir er en industribedrift med smelteverk som produserer blant annet jern og titandioksid. Hensikten med denne oppgaven er å undersøke hvorvidt en industrirobot kan være et mulig alternativ til denne type manuelle arbeidsoppgaver. I dette kapitlet presenteres det bakgrunn for oppgaven, formålet, avgrensinger, teori om slaggrester og informasjon om bedriftene som er oppdragsgivere.

## 1.1 Bakgrunn

Stålindustrien preges (i dag) av manuelt arbeid utført på arbeidsområder der både temperaturene er ekstremt høye og der lokalitetene er fysisk begrensede. Arbeid under disse forholdene representerer slik sett en HMS-utfordring. Hensikten med denne oppgaven er å se på alternative robotløsninger med tanke på å kunne bidra til å løse denne utfordringen. Ved bruk av robot under slike ekstreme forhold, reduseres risikoen til arbeiderne vesentlig. I tillegg kan det være med på å øke effektiviteten, redusere kostnadene med en konkurransedyktig kvalitet.

Arbeid under ekstreme temperaturer kan medføre fysisk belastning for de ansatte. I følge arbeidstilsynets modell bør ikke en arbeider jobbe i omgivelser med høy temperatur over tid, da dette kan medføre nedsatt evne til å jobbe effektivt og riktig og slik sett utgjøre en HMS-risiko [1].



Figur 1.1: Slagg tapping [2]

## 1.2 Formål

Formålet med denne oppgaven er å gjøre produksjonen av titan og jern mindre risikofylt og finne en løsning basert på dagens prosess. Løsningsforslaget har til hensikt å fremme en alternativ automatisert løsning som kan bidra til å redusere nødvendigheten av operatørens medvirkning i arbeidsoppgavene i prosessen. Hovedformålet er reduksjon av sikkerhetsrisiko ved at driften automatiseres, samtidig at kvaliteten på produktet opprettholdes.

Oppgaven tar slik sett sikte på å vurdere ulike robotiserte løsninger og lage en nedskalert prototype av problemstillingen, for deretter å konkludere med en løsning som fremstår som best egnet for denne type automatiserte oppgaver.

## 1.3 Avgrensinger

I oppgaven vil det bli konstruert en nedskalert prototype av en renne (som vist i figur 1.2). Rennen vil være mindre dimensjonert enn den tilsvarende rennen knyttet til anlegget hos TTI i Tyssedal. Tyssedalanleggets renne har en dimensjon på omtrent 7 meter i lengde og 1,3 meter i diameter.

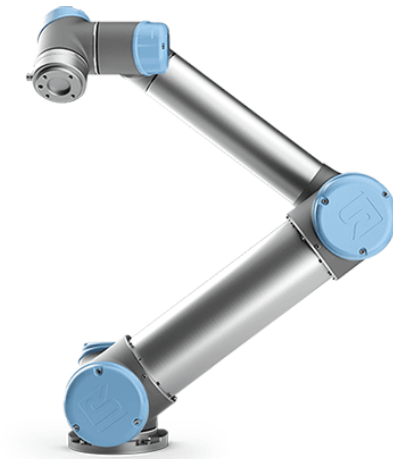


**Figur 1.2:** Nedskalert prototype ved LAB HVL Haugesund

Prototypen som blir satt sammen er en takrenne på 2 meter med en diameter på 125mm. Laboratoriet ved HVL Haugesund brukes som lokale for prototypen. Prototypen er konstruert med den hensikt at vinkelen skal være 30°. Simuleringer knyttet til prototypen, programkode og etterfølgende testing av robot vil ikke foregå i omgivelser med ekstrem varme tilsvarende som TTI i Tyssedal.

Laboratoriet ved HVL Haugesund har installert en UR5e robot. Roboten er utviklet av firmaet Universal Robots og er en av flere modeller de lager. Denne roboten er fastmontert

på en tralle, og den er slik sett mobilt flyttbar. Robotens rekkevidde er 850mm ved full utstrekning [3]. Simuleringsforsøkene i laboratoriet er ikke dimensjonert tilsvarende de reelle dimensjonene i Tyssedal, men er dimensjonert ned til omtrent 1/3 av størrelsen av rennen til TTI.



**Figur 1.3:** UR5e robot [3]

Roboten i laboratoriet ved HVL Haugesund er en kollaborativ robot. Dette betyr at roboten er en menneske-samarbeidende robot. For opprettelse av programkode for styring av UR5e roboten blir programvaren PolyScope brukt. Det ferdige programmet vil kun fungere for oppgavens simuleringer sammen med en UR5e, og HMI panelet vil kun bli simulert via dataprogrammet TIA Portal på laboratoriet.

#### 1.4 Slaggrester ved smelting av ilmenitt

Under forbrenningsfase i produksjon av jern og titanslagg så oppstår det et produkt som kalles slaggrester. Slaggrester dannes ved ulike metallurgiske reaksjoner. Ved høye temperaturer, slik det er under metallproduksjon, kan slaggrester i perioder feste seg i smeltebadet og bli liggende som et produkt i rennen. Slaggresterne som ligg fast i rennen må rakes bort eller fjernes manuelt med hjelp av ulike slagverktøy, til eksempel en rake eller en spett.



Figur 1.4: Eksempel på slaggest [4]

## 1.5 Goodtech AS

Goodtech AS er et norsk børsnotert teknologikonsern som har eksistert i mer enn 100-år. Selskapet har rundt 300 medarbeidere. Goodtech er en teknologileverandør som tilbyr tjenester som automatisering, robotisering, industrideknikk og digitalisering. De jobber med blant annet konsulenttjenester, og er totalentreprenører i ulike teknologi prosjekter [5]. Goodtech har hovedkontor i Oslo med flere avdelinger i Norge, Sverige og Finland.

Goodtech AS er oppdragsgiver for denne oppgaven. Selskapet har lang erfaring med arbeidsleveranser til industrien slik som metall, gruvedrift og smelteverkindustrien [6]. Selskapets hensikt med prosjektet er å fremme utvikling av teknologi knyttet til HMS innen industrirelaterte virksomheter.

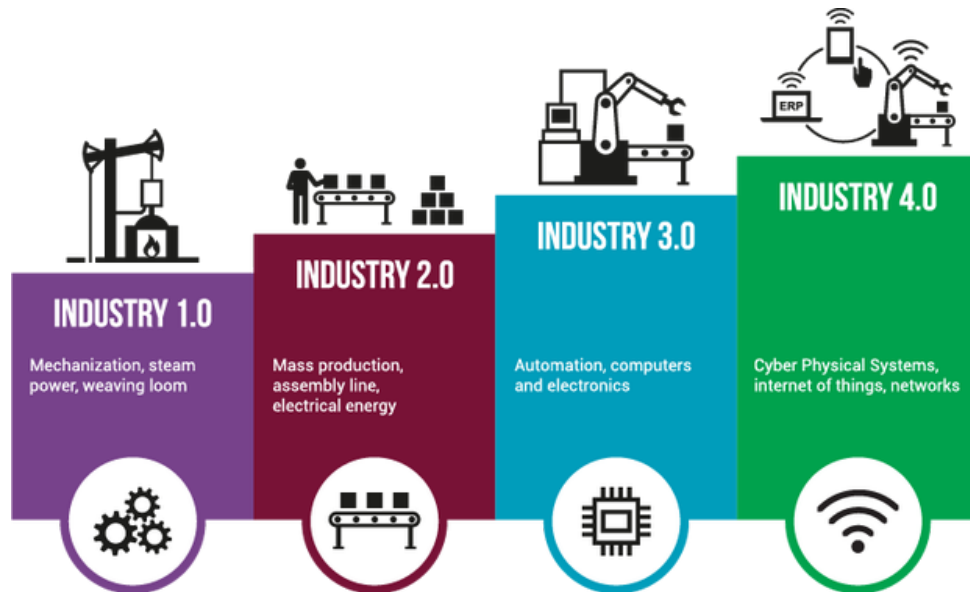
## 1.6 TiZir Titanium & Iron AS

TiZir Titanium & Iron (TTI) i Tyssedal har en relativt lang produksjonshistorie. Smelteverket var vært virksomt siden 1986. TTI er en del av Eramet Group og har omtrent 270 ansatte ved avdelingen i Tyssedal.

Anlegget i Tyssedal produserer titanslagg og høykvalitetsjern. Hovedingrediensene til produksjonen kommer fra Afrika til Tyssedal i Odde for å lage høy kvalitet på produktet. Titanslagget som blir produsert, blir for det meste brukt som tilsetning i maling. Som Europas eneste ilmenittsmelteverk, har de fått tilskudd av enova-støtte. TTI i Tyssedal har til hensikt å opprette et hydrogenprosjekt for å produsere grønn energi ut i fra sine egne produksjoner [7].

## 1.7 Industri 4.0

Oppdragsgiverene for denne oppgaven jobber tett inn i mot de industrielle revolusjonene. Den fjerde industrielle revolusjonen, kjent som Industri 4.0 er fremdeles under utvikling [8]. Industri 4.0 kan defineres som industriell produksjon med fokus på automasjonsrettet autonomi, algoritmer og kunstig intelligens.



Figur 1.5: Illustrasjon av industri 4.0 [8]

Kunstig intelligens og bruken av algoritmer er pekt ut som de sentrale elementene i denne revolusjonen, på grunn av veksten av store databehandlinger kjent som “big data”. Fra og med 2020 ble industri 5.0 revolusjonen innledet, denne baserer seg med større grad på videreutvikling av industri 4.0 og særlig samhandling mellom menneske og kunstig intelligens.

Som den fjerde revolusjonen, må det ha vært tre andre store revolusjoner før.

- Den første industrielle revolusjonen fant sted på 1700-tallet. Denne revolusjonen var starten på at manuelt arbeid kunne erstattes av dampdrevne maskiner.
- Den andre industrielle revolusjonen startet omtrent på 1800-tallet et århundre etter den første. Revolusjonens fremste representant var Henry Ford. Med denne andre revolusjonen ble masseforbruk og masseproduksjon tilgjengelig. Konseptet «samlebånd» som fortsatt er i bruk, dukket opp under denne revolusjonen.
- Den tredje industrielle revolusjonen, fant sted rundt midten av 1900-tallet. Datamaskiner og PLSer ble inkludert for å kontrollere industrielle prosesser, samt utvikling og bruk av roboter og sensorer

Hensikten med dette delkapittelet er å vise til en utvikling innen industrifeltet, fra manuelt arbeid mot en økende grad av automatisering innen ulike arbeidsoppgaver.

## 2 Bakgrunn

Det vil i dette kapitlet bli presentert ulike løsninger knyttet til oppgavens problemstilling. Kapitlet omfatter installasjon av robottyper, leverandørinformasjon, HMS og tekniske spesifikasjoner.

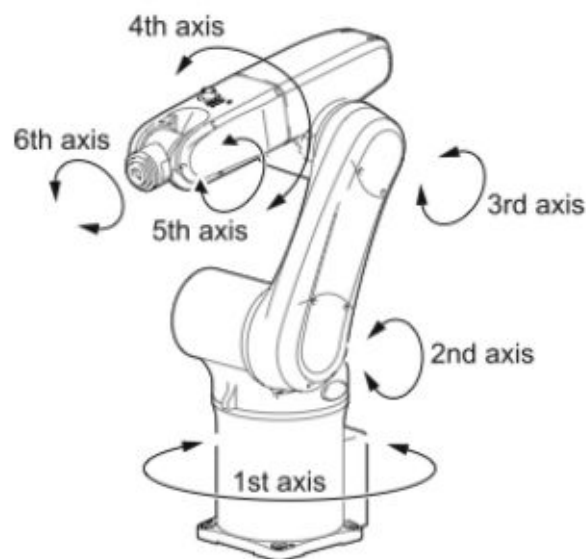
### 2.1 Valggrunnlag til robotarmen

Robotarmen har enkelte funksjonskriterier som må være oppfylte. Kriteriene som er satt for oppgaven kommer fra dagens løsning og område hos TTI og arbeidsgiverne sine innspill til oppgaven.

- Robotarmen bør ha et maksstrekke tilnærmet 3 meter for å nå mesteparten av rennen.
- Robotarmen må kunne bytte verktøy underveis når programmet kjører. Da det kan komme hendelser der en er nødt til å skille mellom fin polering av rennen eller grov polering. Det kan oppstå hendelser der slaggrester sitter ekstra godt fast i rennen.
- Det skal være mulig å kommunisere mellom valgt robot og en Siemens PLS, for å utveksle relevant data til operatør.
- Robotarmen må kunne tåle høye temperaturer i omgivelsene ved TTI.
- Robotarmen må kunne tåle å operere under støvete omgivelser.
- Det er liten plass for lokalisering av robotarm. Plassmangel må tas hensyn til.

### 2.2 Robotarm

En robotarm er en elektromekanisk maskin som vanligvis er programmert til å utføre en oppgave via flere programmerbare kommandoer. Den er designet til å ha lignende funksjoner som en menneskearm, men kan bli produsert i ulike mengder av frihetsgrader. Frihetsgrader refererer til antall bevegelige ledd på roboten (som vist i figur 2.1).



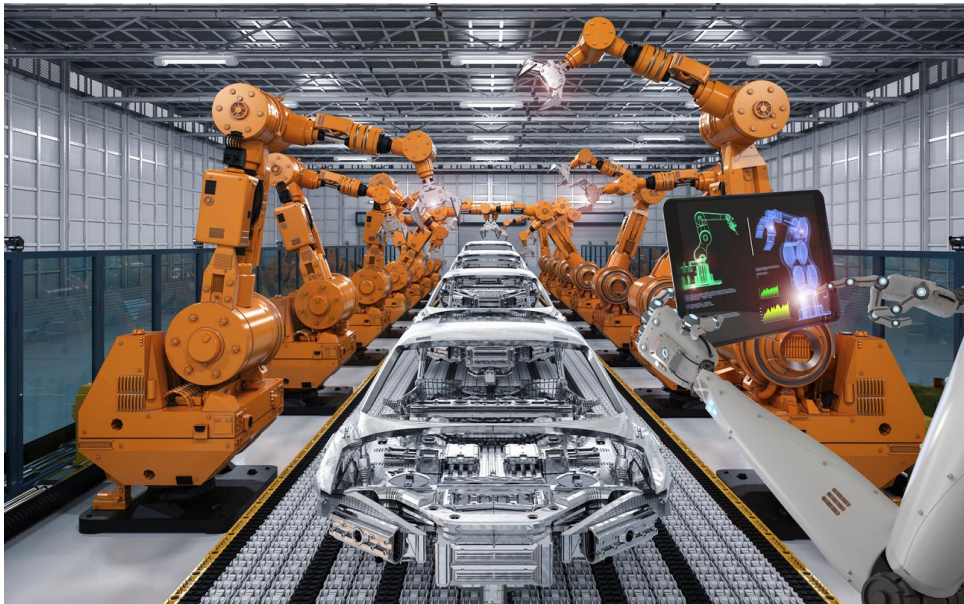
Figur 2.1: 2d skisse av frihetsgrader [9]



Robotarmer har mange bruksområder. En robotarm er programmerbar og kan enkelt tilpasses til ulike verktøy den blir satt til å operere med.

En robotarm kan deles inn i mange kategorier, men hovedfokus for denne oppgaven vil være industriroboter og kollaborative roboter, også kjent som samarbeidende robot. Forskjellen på disse er at kollaborative roboter er designet for å samarbeide med mennesker, mens industriroboter er designet for å ta over jobben til mennesker [10].

Kollaborative roboter kan gjøre sine arbeidsoppgaver i nærheten av arbeiderne, men kan ikke håndtere de tunge jobbene slik som en industrirobot kan. En fordel kollaborative roboter har, er at mangen krav til regelverk forsvinner, noe som er kostnadsbesparende i forhold til industrirobotene. Industrirobotene kan håndtere store og tunge oppgaver, men industriroboten må plasseres ved et avgrenset område for at regelverket skal ivaretas.



**Figur 2.2:** Illustrasjon av industriroboter i arbeid [11]

Industriroboter anses for å være fleksible, i og med at industriroboten er omprogrammerbare og slik sett kan finjusteres til oppgaven den er satt til å utføre. Programmering av industriroboter kan utføres via ulike typer robotapplikasjoner som hører til den spesifikke typen. Robotindustrien fremstår i større grad som det foretrukne automatiseringsverktøyet for industribedrifter, siden de er effektive og er med å øke produktiviteten, samt produserer samme høykvalitetsprodukter som manuelle arbeidstjenester. I tillegg har de en kostnadsreducerende effekt for bedriftene [12]. Denne oppgaven har til hensikt å vise at sikkerhetsrisikoen kan reduseres betydelig med hjelp av industriroboter.

## 2.3 Industrirobot leverandører

I dag er det mange ulike leverandører av industriroboter. Ledende leverandører innen industriroboter er ABB, FANUC, KUKA og YASKAWA [13]. Disse leverer alt fra små enkle utgaver til større, og mer robuste roboter. Designmessig er alle robotene tilnærmet like, og det er små forskjeller i funksjonaliteten. Hovedforskjellene mellom de ulike leverandørene er programmering i robotapplikasjonene.

Ved valg av leverandører for denne oppgaven har det blitt sett nærmere på ulike industriroboter fra bedriftene ABB, KUKA og FANUC. Bakgrunn for valg av disse leverandørene er at de er blant de fremste virksomhetene innen utvikling av industrielle roboter, samt at arbeidsgiver for denne oppgaven har kjennskap til disse leverandørene.

## 2.4 Presentering av industriroboter

### 2.4.1 ABB

ABB er et verdensledende elektro og automatiserings selskap med hovedkontor i Sveits. Sammen med industriroboter så produserer ABB flere ulike typer automatiserings løsninger. I år 2002, ble ABB første leverandør av robotarmer til å selge over 100 000 roboter [14]. Per tid er ABB robotics en av de verdensledende robotleverandørene innen installerte roboter.



**Figur 2.3:** ABB industrirobot IRB-serie [15]

### 2.4.2 ABB IRB

ABB sine industrielle robotarmer er en serie som heter IRB. I oppgaven blir det sett på ulike modeller i IRB 6700 serien til ABB. Dette på grunn av at modellene i 6700-serien har rekkevidde som er likt oppgavens behov. Disse robotarmene kommer med en versjon som heter LEAN-ID. LEAN-ID betyr integrert beskyttelse på kabelføring og er bygget for at robotene skal tåle arbeid i vanskelige omgivelser, temperaturforskjeller og blir levert med forsterket elektronikk frem til robotens gripper. ABB IRB kan bli levert med tilleggspakken foundry plus 2 (som vist i figur 2.4). Det betyr at robotene følger standarder

og regelverk som tilsier at roboten kan arbeide under tøffe miljø. Industrirobotene fra ABB er robuste til å ha en lengre levetid og kan tåle slitasje [16].



Figur 2.4: ABB IRB med LEAN-ID og foundry plus 2 [16]

ABB er kjent for sin egenskap om å ha et brukervennlig programmeringsspråk som heter RAPID. De har i tillegg til dette, utviklet verdens mest brukte offline programmeringsverktøy som heter RobotStudio [17]. I programmet blir det programmert med vekt på grafisk tilnærming både online og offline. Serien for industrirobot som kalles IRB, har et design til robotene for å tåle mye, og produktbeskrivelsene sier MTBF 400 000 timer [16]. Industrirobotarmene til ABB som er til vurdering for oppgaven kan en se i tabell, 2.1.

Tabell 2.1: Datablad fra ABB sine roboter[18]

	IRB 6700-150	IRB 6700-175	IRB 6700-205
Løftekapasitet [kg]	150	175	205
Maksimal rekkevidde [mm]	3200	3050	2800
Konstruksjon type	Gulv	Gulv	Gulv
Miljø	LEAN-ID	LEAN-ID	LEAN-ID
Beskyttelse klasse	IP67	IP67	IP67

### 2.4.3 KUKA

KUKA er et tysk automatisering selskap som ble grunnlagt i 1898. Formålet til KUKA er å kunne tilby ulike automasjons løsninger. KUKA leverer alt av løsninger fra robot, robotceller til automatiserte løsninger [19].



Figur 2.5: KR 120 R2700-2 [20]

#### 2.4.4 KUKA QUANTEC

KUKA har utviklet en industri robot som har serie navn QUANTEC. Serien har et funksjonelt design som gir muligheter for allsidig type bruk. Av KUKA seriene så er QUANTEC serien som har lengst rekkevidde og med høy nyttelast av denne typen roboter. I tillegg til at den har blitt utviklet for å tåle arbeid i varme omgivelser, og har blitt designet for arbeid i tøffe miljøer. Med en MTBF PÅ 400 000 timer [21].

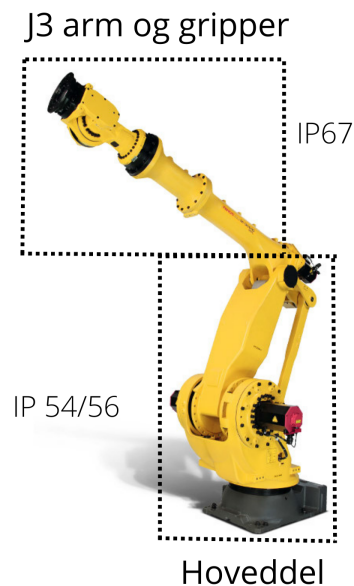
KUKA robotene blir programmert i programmeringsspråket KRL. Programmeringsspråket KRL brukes bare i KUKA roboter [22]. KUKA robotene er i tillegg kompatible med eksterne PLSer og programvarer med bruk av programvaren KUKA.PLC mxAutomation [23]. Industrirobotene som har blitt tatt i betraktning fra KUKA kan sees i tabell 2.2. Alle løsningene som har blitt tatt i betraktning er foundry versjoner som er mer robuste versjoner av industriroboter. Foundry versjonene tåler høyere eksponering av varme og har IP graden 65 eller IP graden 67. Detaljer om ulike kostnadene var det ikke muligheter å finne informasjon om.

Tabell 2.2: Datablad fra KUKA sine roboter.[21]

	KR120 R2700-2F	KR120 R3100-2F	KR120 R3500-2KF
Løftekapasitet [kg]	120	120	180
Maksimal rekkevidde [mm]	2700	3100	3505
Konstruksjon type	Gulv	Gulv	Shelf-mounted
Miljø	Foundry	Foundry	Foundry
Beskyttelse klasse	IP65 & IP67	IP65 & IP67	IP65 & IP67

### 2.4.5 FANUC

FANUC er et japansk selskap som er en av verdensledende leverandørene innen automatiserte løsninger. FANUC produserte industri robot nummer 750 000 juli 2021, som er det høyeste antall roboter som er produsert av robotleverandørene [24]. Største kundesegmentet til FANUC er bilprodusenter, og leverandørene til bilprodusentene i tillegg til at FANUC er voksende i bransjer som matindustrien, elektronikk og innenfor legemiddelin-dustrien [24].



**Figur 2.6:** Illustrasjon av kapslingsgrad på M710iC/70

### 2.4.6 FANUC FoundryPRO

FANUC har et utvalg av industri roboter som er FoundryPro. Som vil si at industri roboten har IP grad 67, som tilsier at det kan brukes i tøffere omgivelser med mye støv, og tåler at utstyret blir spylt.



**Figur 2.7:** *Fanuc M-710 Series [25]*

FANUC sin FoundryPro robot som er M-710iC/70 serien (som vist i tabell 2.3), har maksimal løftekapasitet på 70kg som er den høyeste løftekapasitet til FANUC sin M-710iC serie [25]. Det finnes flere alternativer enn M-710iC/70 i M-710iC serien, derimot er andre modellene innen M-710iC modeller med lavere løftekapasitet. For å få en høyere løftekapasitet må valget bli en annen serie fra FANUC.

**Tabell 2.3:** *Industrielle roboter fra FANUC*

	M-710iC/70	M-900iB-280L	M-900iB/280L
Løftekapasitet [kg]	70	280	330
Maksimal rekkevidde [mm]	2050	3103	3203
Konstruksjon type	Gulv	Gulv	Gulv
Miljø	FoundryPro	...	...
Beskyttelse klasse	IP67	IP67 & IP56/IP54	IP67 & IP56/IP54

Alternativene for M-710iC serien er M-900iB (som er vist i tabell 2.3). M-900iB serien har ikke en kapslingsgrad på hele industri roboten (som vist i figur 2.6) [26]. Eneste løsningene for FANUC sin M-900iB serie er IP67 grad på J3 arm og vrist, og IP54/IP56 på robot kroppen (som illustrert i figur 2.6). Tillegg i serien M-900 er vrist motorene montert på skulderen for få en ekstra beskyttelse ved arbeid i varme omgivelser [26].

Programmering av FANUC sine industrielle roboter kan gjøres på ulike måter. Enten ved bruk av «teach pendant» som krever at roboten må bli satt ut av drift. Alternativet er å programmere i FANUC sitt programmeringspråk Karel som bare brukes i FANUC sine roboter [27]. Ved å programmere i Karel trenger ikke roboten å bli satt ut av drift, og som gir muligheten for å laste ned programmet til roboten når programmet er klart.

### 2.4.7 Sammendrag av industriroboter

Ytelsesverdiene til de prensterte industrirobotene er tilnærmet lik hverandre, det er lite eller ingen forskjell på egenskapene til de ulike industrirobotene. Alle industrirobotene har gode hastigheter og nøyaktighetene er tilnærmet perfekte. Det er heller ingen stor forskjeller i leveranse, da robotleverandørene leverer utstyr og komponenter til alle kontinenter i verden.

Der en kan trekke frem forskjeller, så er det hovedsaklig knyttet til programmering av robotene. Leverandørene som er vurdert leverer forskjellige applikasjoner som robotene programmeres i.

ABB robotens LEAN-ID og foundry plus 2 teknologi knyttet til integrert beskyttelse på kabelføring og elektronikk-skapet er et fortrinn knyttet til beskyttelsesevnen for de presenterte robotene.

## 2.5 Behandling av varmeutfordringer

For at robotarmen skal tåle de høye temperaturene fra prosessanlegget til TTI, må roboten ha et supplement for kjøling, og aktive tiltak for å tåle de høye temperaturene over tid. I dette delkapittelet blir det fremvist ulike ressurser som kan være til hjelp for at roboten skal kunne operere under forutsetningene tilknyttet ekstrem varme.

### 2.5.1 Varmekåpe

Roboter er mye brukt i industrielle miljøer med tøffe driftsforhold. Uten riktig beskyttelse, kan store pengesummer gå tapt på et øyeblikk. Varmekåper til roboter er produsert for å beskytte roboten mot varme, kjemikalier og fuktighet. Til industriroboter er kåpene godt egnet for å forhindre komponentfeil og forlenge robotens levetid, men kan være med på å begrense bevegelighetene når roboten er under drift [28].



**Figur 2.8:** Illustrasjon av varmehåpe på en industrirobot [29]

Med epostkommunikasjon fra en stor europeisk leverandør innen varmehåper, kommer det frem at materialet de bruker kan sammenlignes med en aluminiumsduk. Leverandøren opplyser om at produktet deres kan tåle opp til 1200°C. Varmehåper som leverandøren

produserer er vanskelig å prisfeste, før leverandør får innhentet informasjon om den eksakte modellen på industriroboten, hvor roboten er situert og hvordan verktøyet til roboten er designet.

Før all informasjon er tilgjengelig, kan leverandøren av varmekåper foreta et prisanslag omkringliggende 4500£. Som et tillegg til denne prisen kan det antas at denne summen vil stige når informasjon om hvilke verktøy som vil bli montert på industriroboten foreligger.

### 2.5.2 Aktiv luftkjøling

Aktiv luftkjøling er en måte å fjerne varme fra et objekt. Metoden er basert på at objektet som skal kjøles ned, bør ha mer overflate eller at en ved bruk av vifte generer luft mot objektet som skal kjøles ned. I henhold til den andre termodynamikkens lov, må luften være kjøligere enn objektet som skal avkjøles.

Aktiv luftkjøling kan kombineres med tidligere nevnt varmekåpe. Luften kan bli spylt inn med rør i underkanten av roboten. Lufttilførsel for kjøling av robot, kan være vifte ved hjelp av pumpe, installering av kjøleribbe med vifte eller ved installering av et vannkjølingsystem.

Om omgivelse temperaturen ikke kommer ned til et akseptabelt temperaturnivå under produksjonen, så kan det tilføres luft når roboten har en pause. Da får roboten nedkjølt seg periodevis og en kan opprettholde aktivitet for roboten i ettertid.

Luftkjøling kan ha sine utfordringer om det ikke blir opprettholdt et viss vedlikehold eller luftsystemet slutter å virke. Det vil være nødvendig å ha rutinemessig kontroll om at systemet fungerer. For kommunikasjon med luftsystem kan det kobles til en elektrisk styring, så personell har kontroll på at system fungerer under produksjon.

### 2.5.3 Eksponeringstid

For å unngå over opphetning av industrirobot så må det utføres tiltak for å redusere tiden roboten er eksponert direkte mot varmen under tapping.

Tiltak som kan være med på å forhindre eksponeringstid ved høye temperaturer kan være:

- Snu industrirobot 180°, slik at den er lengst unna produksjonsrennen under tapping.
- Vente lengst mulig etter tapping i rennen før industrirobot starter med å fjerne slaggrester.
- Montere temperatursensor til industrirobot slik at operatør vet eksakt temperatur på robotarm.
- Opprette rutiner for industrirobot og tapping ved produksjonsrenne.
- Industrirobot opererer i intervaller.

## 2.6 HMS- Helse miljø og sikkerhet

HMS er en betegnelse for å ivareta liv, helse, miljø og materialer. Godt arbeidsmiljø blir ivaretatt om en har tatt høyde til risiko for farer og ulykker. Dette med å gå aktivt inn for å sette lys mot helsefremmende faktorer og positive holdninger i arbeidsmiljøet. En av de viktigste arbeidsoppgavene som må gjøres før arbeidstart er å starte en risikovurdering.



### 2.6.1 Risikovurdering

Risikovurderingen skal utføres og en skal gå gjennom alle arbeidsuhell ved arbeidsoppgaver som eventuelt kan oppstå. Det blir viktig at det blir utført en risikovurdering som ser for seg alle arbeidsoppgaver og uhell som kan oppstå i industrirobotens levetid.

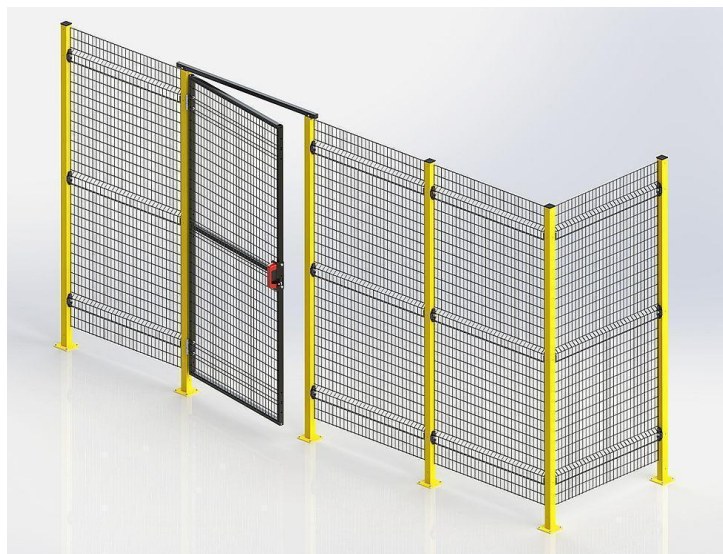
En ønsker å opprettholde kravene som er satt i ISO 13857:2019 for å hindre personskader i forbindelse med en applikasjon. Opplæring av industriroboten må skje under oppsyn av personell som har tatt del i utvikling av robotinstallasjonen. Feilsøking og vedlikeholds-rutiner må skje ved normale driftsforhold av industriroboten.

**Tabell 2.4:** *Potensielle uønskede hendelser under drift*

Hendelse	Mulige farer
Element i bevegelse	- Kuttskade - Klemskade - Støtskade - Øyeskade
Uforutsett	- Robot setter seg fast - Forskyvning av renne
Sensorikk	- Feil data lest fra sensorikk

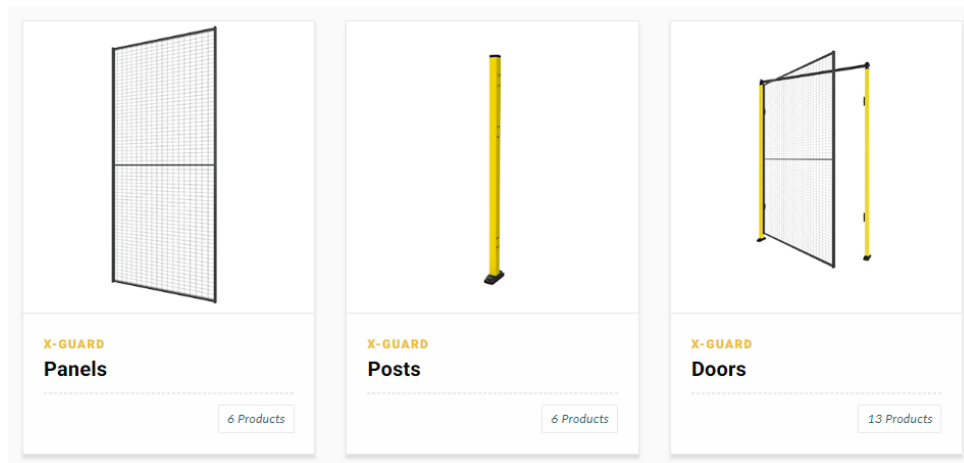
### 2.7 Fysisk sikkerhetsbarriere til robot

I tilfeller der industriroboter er aktive, må fysiske sikkerhetsbarrierer monteres for å forhindre skade på mennesker og kroppsdeler som oppholder seg nær utstyret. Regelverket ble vedtatt først i 2006 som en standard fra EU maskindirektivet. Maskindirektivet inneholder mange lovverk og regler for bruk av ulike typer elektromekaniske maskiner deriblant industriroboter.



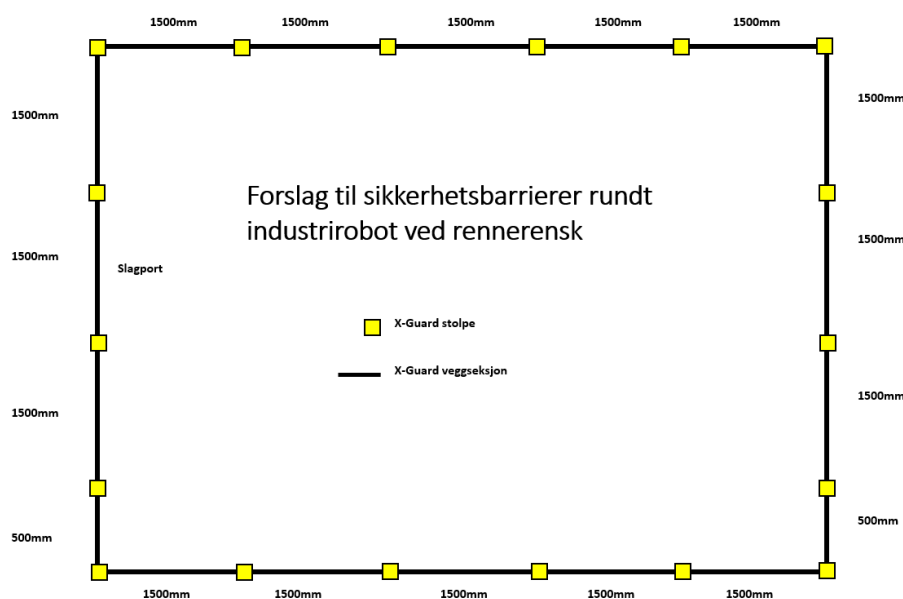
**Figur 2.9:** *Illustrasjon om hvordan avsperring kan se ut [30]*

På bakgrunn av regelverket ISO 13857:2019 [31], er sikkerhetsbarrierene utviklet (Illustrert i figur 2.9) og er til hensikt å forhindre at kroppsdeler kommer inn i arbeidsområdet når en industrirobot er i arbeid. Regelverket tilsier også at alle industriroboter skal være utstyrt med en “kill-switch”, eller en sensor for å stoppe roboten eller hindre oppstart frem til sikkerhetsdør er lukket og låst. Dette for å forhindre at menneske skal kunne oppholde seg sammen med en aktiv industrirobot.



Figur 2.10: Axelent X-Guard sikkerhetsbarriere produkter [32]

Som et forslag til sikkerhetsbarrierer til den aktuelle oppgaven, blir det presentert flere veggseksjoner samt en inngangsport med integrert “kill-switch” og stolper for å holde veggseksjonene sammen. Det er valgt ut leverandøren Axelent med modellene X-Guard til illustrasjon og forslag for sikkerhetsbarrierer til oppgaven.



Figur 2.11: Forslag sikkerhetsbarrierer

Tabell 2.5: Pris til forslag av sikkerhetsbarrierer

Produkt	Serienummer	Antall	Per enhet pris
X-guard Slagport, Høyde 2000mm, Bredde 1500mm.	D30-XX150A	1	2095,-
X-guard Nettingseksjon, Høyde 1900mm, Bredde 1500mm.	W322-190150	15	1155,-
X-guard Nettingseksjon, Høyde 1900mm, Bredde 500mm.	W322-19050	2	995,-
X-guard Stolpe, Høyde 2000mm, Bredde 50mm.	P11-200	18	565,-
		Total sum:	42820,- Kr

## 2.8 Vedlikehold

For å unngå nedgang eller stopp i produksjonen til industriroboten er det viktig å ha en god vedlikeholdsplan. Ved regelmessig vedlikehold kan industrirobotens levetid forlenges og antall feil på industrirobot reduseres. Robotprodusenter har ulike krav eller anbefalinger i tilknytning til vedlikehold.

Om vedlikeholdsrutiner ikke blir opprettholdt, så kan industriroboten oppleve problemer som: [33].

- Slitasje eller skade på kabler
- Robot kan arbeide utenfor tiltenkte plasseringer
- Oppføre seg uforutsigbart
- Deler av industriroboten eller komponenter kan løsne
- Robot kan føre til skade på mennesker eller utstyr

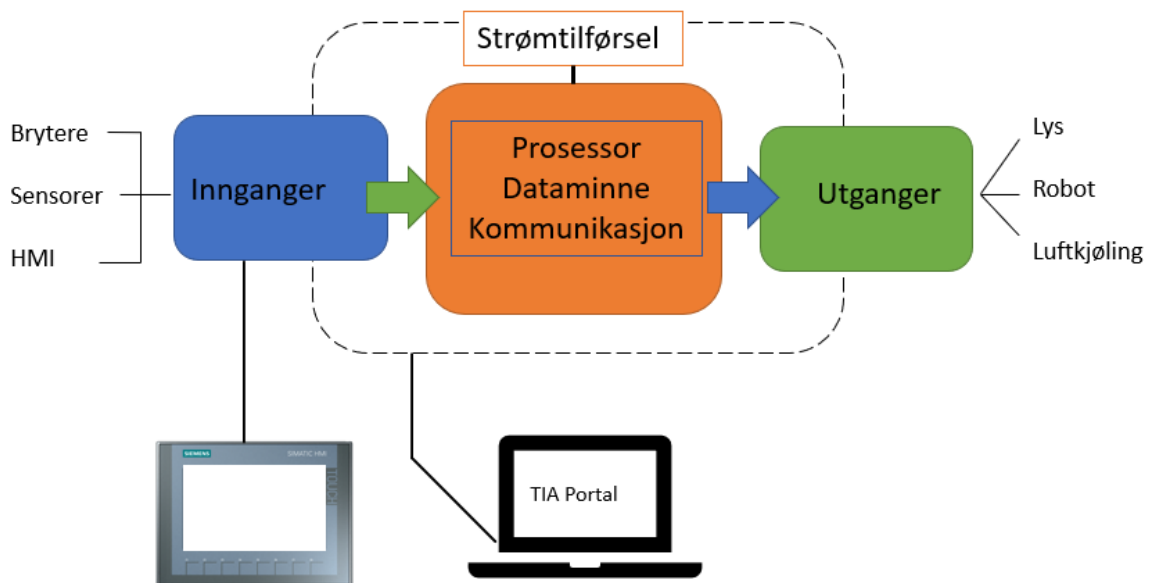
### 3 Innfallsvinkel

Dette kapitlet har til hensikt å klargjøre sammensetningen og den videre sammenkoblingen av hvilke dataprogram og hvilke funksjoner som oppgaven vil nytte seg av. I tillegg vil det bli sett nærmere på funksjoner, sikkerhet og datastrukturer som blir brukt i oppgaven. Avslutningsvis vil oppgavens sammensatte prototype bli presentert.

#### 3.1 PLS - Programmerbar logisk styring

PLS er en industriell datamaskin. En industriell datamaskin blir brukt til styring og kontroll av industrielle prosesser. En PLS kan sammenlignes med en konvensjonell datamaskin, den inneholder blant annet en prosessor. Prosessoren er primærelementet i en datamaskin som utfører instruksjoner, beregninger og arbeider videre med databehandlingen.

En PLS med flere inn- og utganger kan ta imot og sende ut data, samt motta instruksjoner og foreta flere mengdeanalyser samtidig. PLS kan behandle digitale og analoge signaler. Fra en kommunikasjonsmodul så kan PLS hente eller sende data mot andre PLSer.



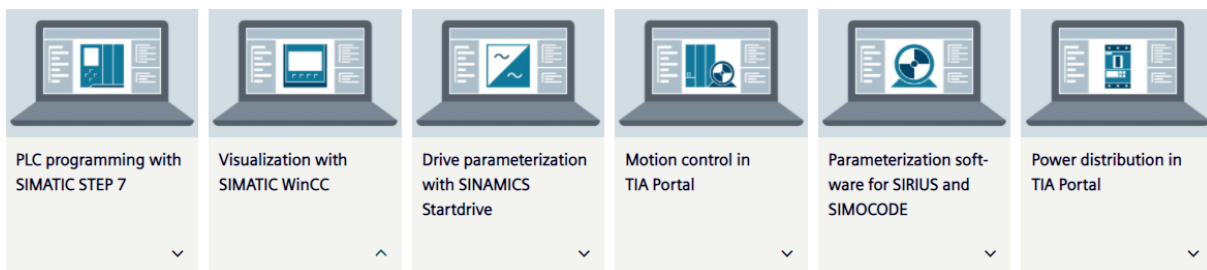
Figur 3.1: Blokkskjema av en prosedyre til en PLS

## 3.2 Programvarer

I prosjektet blir det tatt i bruk programvarer hovedsakelig fra Siemens og Universal Robots. Anvendte programvarer har både åpen og lukket kildekode. Den åpne kildekoden er tilgjengelig for allmenbruk, mens den lukkede er lisensbasert.

### 3.2.1 Siemens TIA Portal

TIA Portal er en portal utviklet av Siemens som gjør det mulig å konfigurere og programmere en PLS. I tillegg innehar TIA Portal ulike programvarer. I figur 3.2 framvises de ulike programvarene til Siemens samt bruksområdene:



Figur 3.2: Programvare i TIA Portal [34]

I TIA Portal er det muligheter for å programmere i programspråkene: Ladder-diagram, Funksjonsblokkdiagram og strukturert tekst blant flere. Siemens forbedrer TIA Portal og PLSene kontinuerlig ved å gjøre disse mer funksjonelle gjennom bruk av ulike maskinvare. Siemens er lukket kildekode, hvilket betyr at det behøves ulike lisenser for å få tilgang til deres programprodukt.

### 3.2.2 SIMATIC Robot Integrator

Simatic Robot Integrator defineres som en applikasjon integrert i TIA Portal. Denne applikasjonen har til hensikt å redusere robotleverandørenes behov for egne spesifikke applikasjoner. Grunnleggende sett er hensikten med appen å gi robotleverandørene et universalt brukergrensesnitt i TIA Portal [35].

Det betyr at Goodtech kan anvende brukergrensesnittet i TIA Portal, dette innebærer besparelser for bedriften. I tillegg behøves ikke spesifikke kunnskaper eller programmeringsverktøy laget av de ulike robotleverandørene.

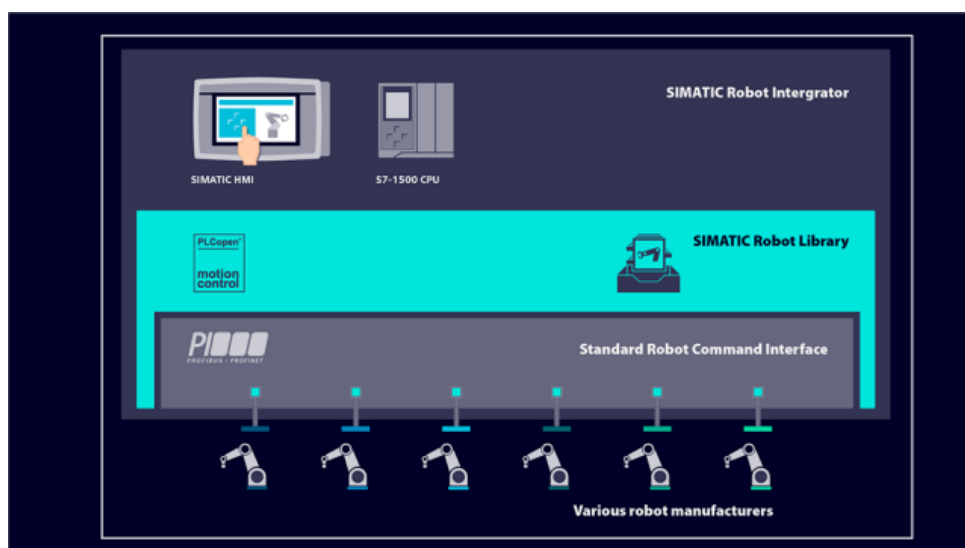
Et eksempel er å bruke et SIMATIC HMI Mobile Panel fra Siemens (som vist i figur 3.3), i stedet for å tilegne seg kunnskap om de ulike robotleverandørenes produkter.



Figur 3.3: Illustrasjon av SIMATIC HMI Mobile Panel sammen med Robot Integrator [36]

### 3.2.3 SIMATIC Robot Library

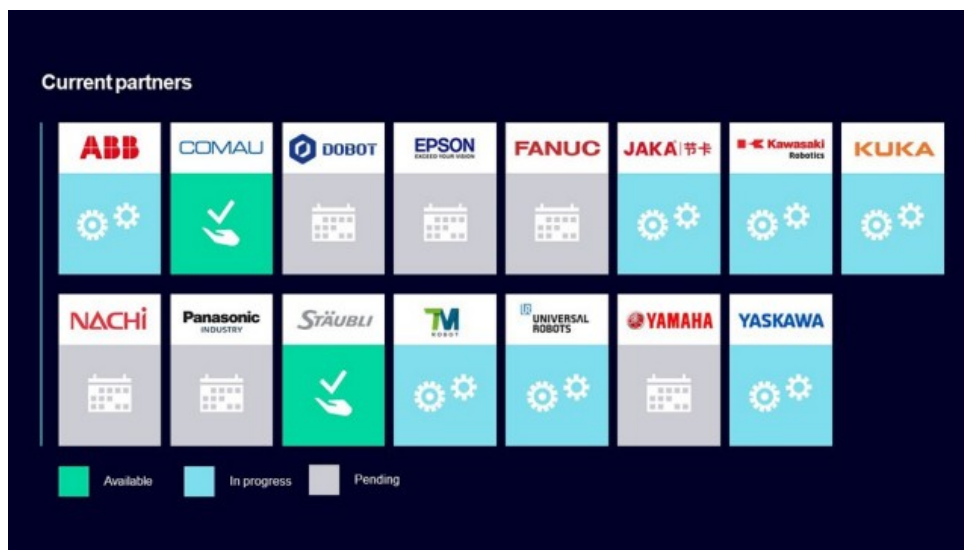
Siemens har utviklet SIMATIC Robot Library for TIA Portal, som har til hensikt å erstatte leverandørens egne robotbibliotek [35]. SIMATIC Robot Library sammen med Simatic Robot Integrator fremmer muligheten for at disse programvarene kan integreres i TIA Portal. I figur 3.4 vises tanken bak løsningen til Siemens, med det tiltenkte universale produkt.



Figur 3.4: Illustrasjon av bruken av SIMATIC Robot Integrator [36]

### 3.2.4 Sammenkobling av SIMATIC Robot Integrator og SIMATIC Robot Library

Ulempen ved produktet til Siemens er at et stort antall av robotleverandørene ikke har implementert sine tjenester mot det sammenkoblede systemet. Det blir samarbeidet med ulike leverandører med å få iverksatt dette i TIA Portal programvaren. (Som vist i Figur 3.5), er leverandørene som er merket “Available” tilgjengelig i SIMATIC Robot Library, og leverandørene som er merket “In Progress” er under utvikling av programvare til å knytte seg til det sammenkoblede SIMATIC-systemet.



Figur 3.5: Leverandører som samarbeider med Siemens angående SIMATIC Robotic Library [36]

### 3.2.5 PROFINET

For å integrere robot sammen med PLS trengs det en form for datakommunikasjon. PROFIBUS & PROFINET International er definert om standardisering av brukergrensesnitt på tvers av ulike leverandører. Målet med PROFINET standarden er sømløs kommunikasjon over Ethernet. Arbeidet med en ny standard for grensesnittet til robot sammen med PLSer ble startet i 2021 av PROFIBUS & PROFINET International med profilen som heter “Profile for Robot System” [37].

Bruk av en standard for PLS-leverandører er essensielt for et sammenkoblet robotbibliotek. PLS leverandørens sin standard-programvare, innebærer at det blir enklere å knytte sammen med robotleverandørens programvare. Figur 3.4 viser hvordan en universal løsning kan nyttes ved bruk av PROFINET-standarden.

### 3.2.6 SocketTest

SocketTest er en programvare som brukes til kommunikasjonstesting mellom enheter i et nettverk. SocketTest er en programvare for å opprette TCP/UDP servere. Ved testing brukes det enheter som tar i bruk TCP/UDP protokollen. SocketTest kan opprette TCP/UDP servere eller klienter for testing av tilkobling over et nettverket.

I denne oppgaven anvendes denne programvaren til å teste ut tilkoblingen mellom UR5e robot og Siemens PLS. Dette ved å gi ulike kommandoer fra SocketTest samt bekreftelse av returnerende signaler fra robot.

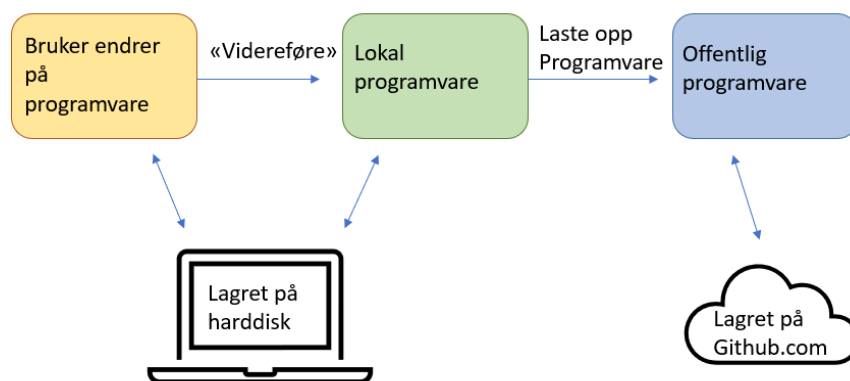
### 3.3 Deling av programkode

Under ulike teknologiske prosjekt jobbes det ofte i grupper. Ved programmeringsprosessen kan det være nyttig å dele programkode innad i bedrifter eller samarbeidspartner. I denne oppgaven har det vært et hovedfokus på å finne fruktbare løsninger for hvordan Goodtech kan dele programkode innad i bedriften.

#### 3.3.1 GitHub i TIA Portal

Tilgjengeligheten til GitHub er universell i og med at programmet har en åpen kildekode. Fordelen ved bruk av GitHub sammen med TIA Portal er funksjonen “Version Control”. “Version Control” er en funksjon konstruert for å kunne håndtere programkodeendringer, og muliggjør kodeendringer i allerede opplastede filer.

Figur 3.6 har til hensikt å vise hvordan opplasting til GitHub fungerer. Om eventuell feilprogrammering oppstår finnes det metoder for å både etterspore feilkode og tilbake stille til korrekt programkode.



Figur 3.6: Illustrasjon av funksjonsdeling av GIT

GitHub er et verktøy som brukes i lag med TIA Portal. For å dele innholdet som er programmert i TIA Portal som “PLC Tags”, organisasjonsblokker, funksjonsblokker, og i tillegg for å se endringene som har blitt gjort i de ulike filene.

#### 3.3.2 TIA Portal multiuser engineering

TIA Portal multiuser engineering er en applikasjon i TIA Portal for deling og synkronisering av programkode. Ved samarbeidsprosjekter med applikasjonen “multiuser”, må



det opprettes en flerbruker-server. For å kunne jobbe samtidig med flere brukere må hver bruker lage sin egen lokale server.

I den lokale serveren kan redaktører legge inn endringene sine i et prosjekt for deretter å kunne se endringene i den globale serveren, for så å kunne publisere koden i ettertid. Etter publisering av programkode er endringer fra den lokale serveren tilgjengelig for alle redaktørene i serverprosjektet.

Multiuser er fremdeles under utvikling, og det jobbes fremdeles med å utvikle denne applikasjonen.

### 3.4 Prototype

Det vil i dette delkapittelet bli presentert oppgavens sammensatte prototype. Delkapittelet vil inneholde hvordan prototypen er bygget opp, og satt sammen av både fysisk arbeid og utstyr som er brukt.

#### 3.4.1 Prototypens byggverk

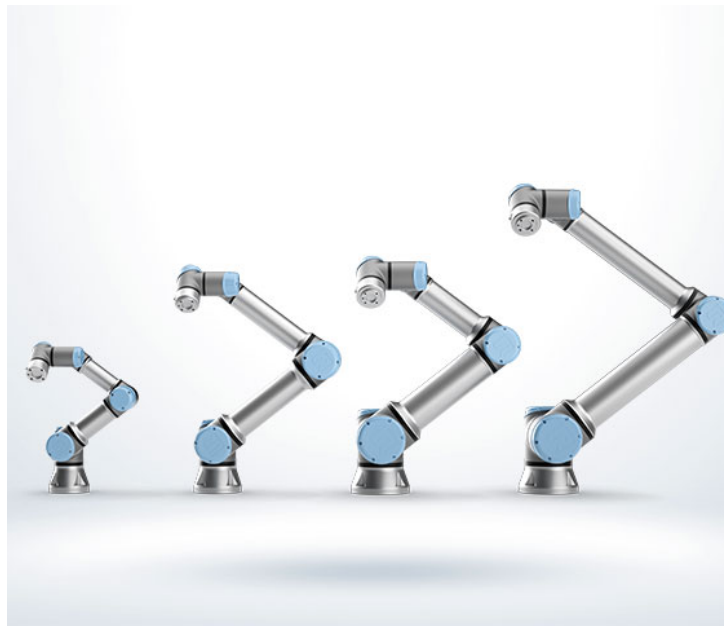
I denne oppgaven er det konstruert en prototype av en renne (som vist i figur 3.7). Byggverket til rennen er satt sammen på laboratoriet ved HVL Haugesund. Rennen har til hensikt å gjenspeile TTI sin renne ved produksjonsanlegget i Tyssedal. Den konstruerte prototypen har en hellingsgrad på  $30^\circ$ . Dette er i samsvar med produksjonsrennen i Tyssedal.



**Figur 3.7:** Nedskalert prototype for simulering

### 3.4.2 Universal robot

Universal Robots var det første selskapet som utviklet lettvektede robotarmer. Tre forskere fra Syddansk Universitet i Odense kom frem til en konklusjon at markedets industriroboter er høyt priset, tunge og komplekse ved bruk [38]. Universal Robots sin konklusjon var at daværende robotmodeller eller robotutviklingsidéer ikke fremstod som en tjenelig og funksjonell investering for små og mellomstore bedrifter.



**Figur 3.8:** *Universal Robots e-serie [39]*

Universal Robots ble offisielt grunnlagt i 2005 [38]. De har økt sine automatiseringsprodukter til fordel for lavere kostnader og mindre behov for ekstern arbeidskraft. Robotens lave kostnader fremkommer på bakgrunn av den innovative strukturen på robotens “Teach pendant” og robotens kontrollenheter. Fordeler ved robotmodeller fra Universal Robots er robotenes fremtoning og enkle grafiske brukergrensesnitt. Roboten kan tillæres grunnleggende oppgaver på kort tid, uten at det er nødvendig å skrive programkoder.

Robotmodellene til Universal Robots er utviklet som samarbeidsrobot og krever dermed færre sikkerhetsregler mellom robot og operatør, enn hva en industrirobot krever. Standardregelverket for samarbeidsroboter er ISO/TS 15066 [40]. Standarden tilsier at robotens programvare må kunne overvåke robotmotorens dynamiske belastning. Hensikten er å forhindre at forekomsten av energien i kollisjoner blir for høy knyttet til vekt, hastighet og kontaktflate i kollisjon.

Per i dag finnes det en rekke ulike modeller fra Universal Robots. Sist i rekken er serien “e-series” (som vist i figur 3.8). Denne er den siste serien av Universal Robots og ble lansert i 2018 [41]. Modellen ‘e-series’ blir nyttet i denne oppgaven, og blir kallet UR5e.

### 3.4.3 Robotens arbeidsverktøy

UR5e-roboten som er brukt i oppgaven har montert en gripper med en flat kontaktflate (Som vist i figur 3.9).



**Figur 3.9:** Prototypens gripper

Ved simulering av rennerenssk ble det brukt en whiteboard-visker som verktøy for å utføre program for rennerenssk. Ved testing ble det satt inn en trekloss for ytterligere støtte på whiteboard-viskeren opp mot robotens gripper, (Som vist i figur 3.10). Det ble også lagt ved en fjøring mellom treklossen og whiteboard-viskeren for bedre kontaktpunkt under simulering av rennerenssk.

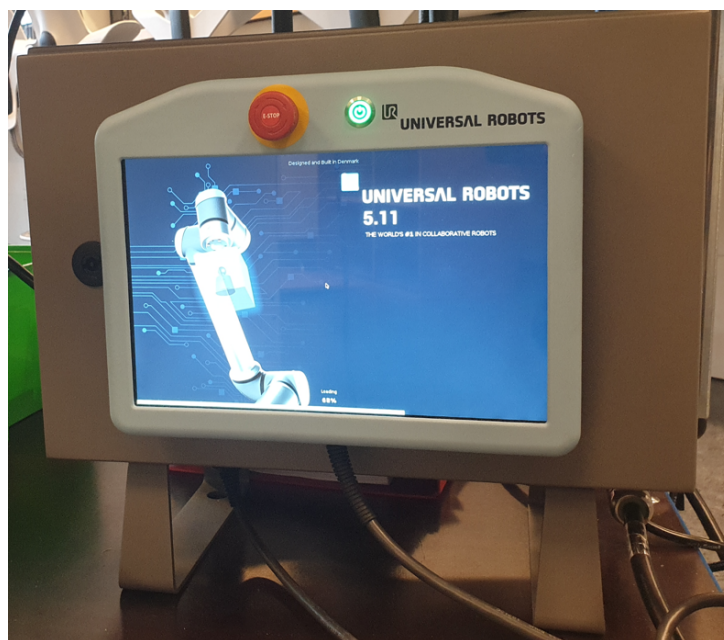


**Figur 3.10:** Prototypens verktøy

### 3.4.4 PolyScope

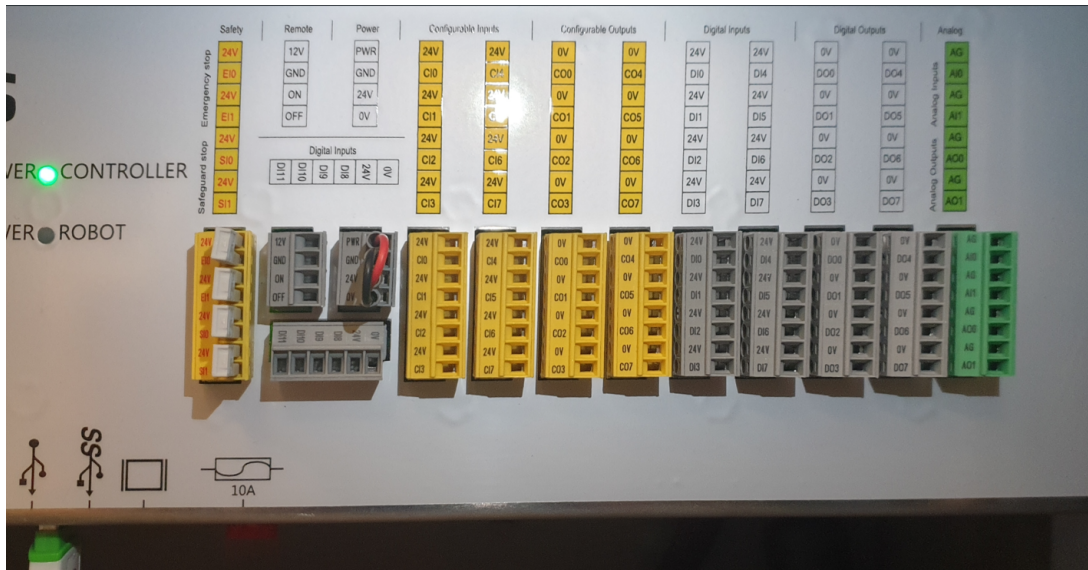
PolyScope er et programmeringsprogram knyttet til programmering av kommandoer til Universal Robots. PolyScope er en programvare som er implementert på en “Teach pendant” som følger med UR5e roboter, (som vist i figur 3.11). “Teach pendanten” er en berøringsskjerm som brukes både til å styre roboten, lage robotprogram og styre kontrollenheten. I denne oppgaven blir det brukt PolyScope for å programmere i et grafisk brukergrensesnitt til UR5e roboten.

### 3.4.5 Elektrisk grensesnitt



**Figur 3.11:** Universal Robots - kontrollenhet og “Teach pendant”

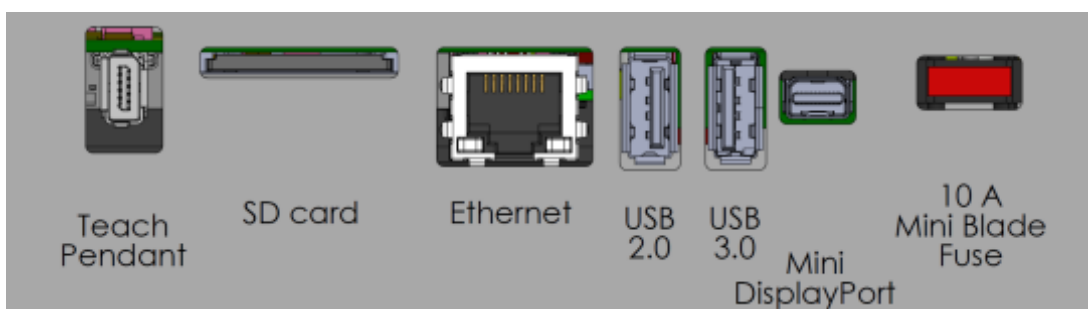
Kontrollenheten (skap) (som vist i figur 3.11), inneholder en liten datamaskin som bruker et linux-basert operativsystem. Kontrollenheten inneholder ulike inn- og utganger som kan brukes av ulike komponenter t.eks.: releer, sensorer, PLSer og nødstopper. Komponentene kan enkelt tilkobles kontrollenheten.



Figur 3.12: Kontrollenhetens grensesnitt for elektrisk tilkobling

De gule tilkoblingsmulighetene i kontrollenheten med (rød tekst til venstre i figur 3.12), indikerer at tilkoblingen er dedikert til sikkerhetssignal. Her blir t.eks. nødstopp tilkoblet. De gule tilkoblingsmulighetene med svart tekst kan tilpasses for sikkerhet. For generelle tilkoblinger knyttet til digitale- og analoge signaler kan de grå og grønne tilkoblingsmulighetene brukes.

Kontrollenheten har også tilkoblingsmuligheter for USB og ethernet-kommunikasjon (som vist i figur 3.13). Hensikten er å foreta opp- og nedlasting av robotprogrammer, samt muligheter for oppdateringer og simulering av robot via en PLS.



Figur 3.13: Kontrollenhetens grensesnitt for tilkoblinger [42]

### 3.4.6 Siemens PLS - SIMATIC S7-1500

En rekke PLSer på markedet kunne ha blitt brukt for løsning av denne oppgaven. Dagens PLSer kan håndtere både omfattende oppgaver og flere oppgaver samtidig.

I oppgavens løsning er det blitt tatt i bruk en PLS-modell som HVL Haugesund har tilgjengelig på laboratoriet, modelltypen er SIMATIC s7-1500. Oppdragsgiver har god kjennskap til programmet TIA Portal som blir brukt for konfigurering av Siemens PLSer.

PLSene fra Siemens støtter PROFINET kommunikasjon. PLS fra Siemens var slik sett det mest tjenlige og funksjonelle valg i prosessen med å finne en PLS til oppgaven.

### 3.4.7 HMI-panel

HMI-panel er et grafisk brukergrensesnitt som lar brukere styre en robot, en maskin, et system eller komponenter. Ved å gi kommando til en PLS, for så å motta en bekreftelse av at handlingen er utført. Brukergrensesnittet for et HMI-panel er designet for brukeroversiktighet. For enklere å kunne navigere i HMI-panelet.

I denne oppgaven blir det opprettet et HMI-panel som har til hensikt til å fungere sammen med UR5e-kontrollenheten via en PLS tilkoblet TIA Portal. HMI-panelet er designet for å ivareta alle deler av helhetsbildet i den tiltenkte prosessen, hvor roboten har til hensikt å utføre rennerensk i tilknytning til TTIs anlegg. HMI-panelet er laget ut fra flytskjema (som vist i figur 3.14). Innholdet til ulike funksjonene i HMI-panelet blir beskrevet tydelig i vedlegg D, kommunikasjonen mellom simulert HMI-panel og UR5e-kontrollenheten er knyttet sammen med PROFINET-kommunikasjon.

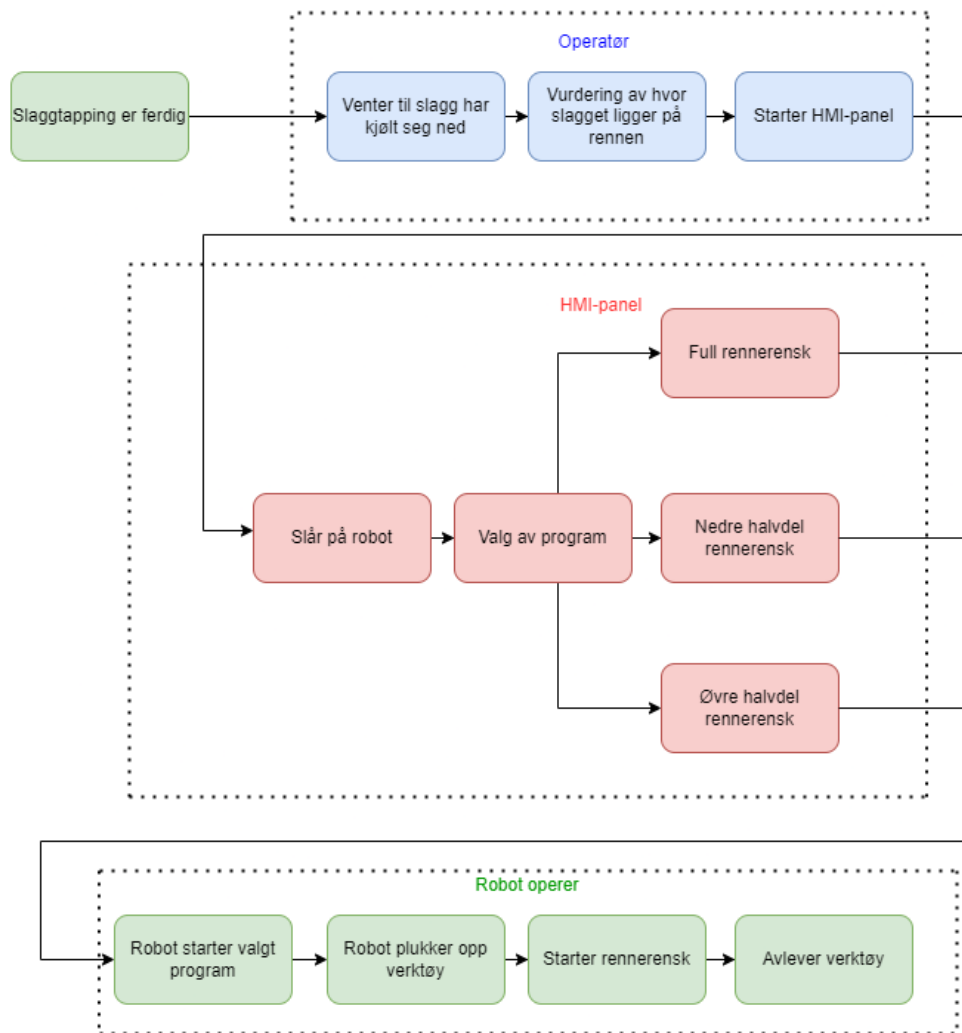
### 3.4.8 Flytdiagram for utførelse av robot

Flytdiagram (som vist i figur 3.14), er laget med hensikt for å belyse hvordan oppgavens robotiserte rennerensk brukes.

Flytdiagrammet er delt inn i tre ulike seksjoner: operatør, HMI-panel og robot.

Når slaggtapping er utført må operatør fysisk vurdere produksjonsrennen, deretter ta en vurdering på hvor i rennen det trengs fjernes slaggrester. Ved oppstart av HMI-panel og robot, må det velges hvilke rennerensk-program som roboten skal utføre. Operatør må navigere frem valgt verktøy som skal brukes, deretter starter roboten valgt program.

Flytdiagrammet er opprettet med hensikt på prototypens oppgaver, og kan fungere som anbefaling eller til inspirasjon for videre utvikling innen robotisert rennerensk.



Figur 3.14: Flytdiagram for fjerning av slaggrester

## 4 Resultat

Oppgavens formål er å drøfte en løsning av en industrirobot som kunne passe til anlegget hos TTI, samt å lage en nedskalert prototype som skal gjenspeile anlegget ved TTI. I dette kapittelet vil det bli presentert et resultat av de overnevnte tema.

### 4.1 Prototypens virkemåte

Rennen er montert på prototypens byggverk med helling på  $30^\circ$ , prototypen er plassert på en kontorpult på laboratoriet ved HVL Haugesund. UR5e-roboten som er stasjonert på laboratoriet i Haugesund er mobilt flyttbar, og under testing blitt plassert ved midten av rennen for at robot skal kunne nå både bunn- og topp av rennen.

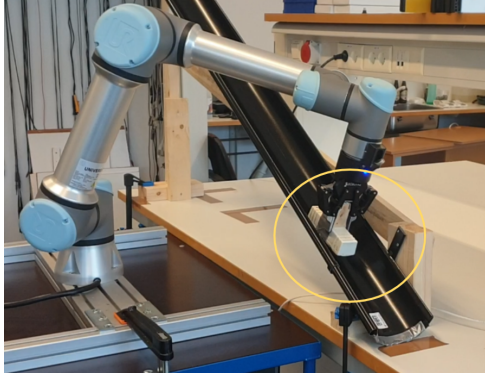
Datakommunikasjon som er blitt brukt i oppgaven er PROFINET via en datasvitsj i den hensikt å koble sammen komponentene: PC, PLS og UR5e. I vedlegg A, blir det presentert ytterligere detaljer om hvordan PROFINET-kommunikasjon mellom data, PLS og robot er opprettet.

#### 4.1.1 UR5e testing

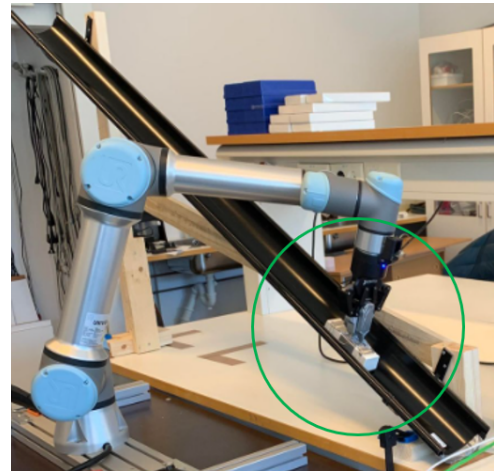
Ved bruk av Universal Robots sin programvare PolyScope, er programmering av UR5e roboten en enkel prosess sammenlignet med programmering av en industriell robot. PolyScope har et grafisk brukergrensesnitt som tilrettelegger for betjening av robotarm, kontrollboks og lar brukeren kjøre roboten simultant mens det blir laget programkode. Roboten er i denne oppgaven programmert til å utføre rennerensking av takrenne montert med helling på  $30^\circ$ . For oppgavens løsning av programmering av roboten, (se vedlegg C). I dette vedlegget blir det forklart hvordan programstruktur er satt opp samt tankegang rundt den opprettede programkode.

Ved simulering av rennerensking er det brukt en whiteboard-visker til robotens verktøy. Verktøyet har til hensikt å utføre poleringen av rennen. Ved testing ble det satt inn en trekloss for ytterligere støtte på whiteboard-viskeren opp mot robotens gripper. Under simuleringen ble det oppdaget at whiteboard-visker ikke stryker med hele flaten mot rennen, (som vist i figur 4.1). Løsningen ble å sette inn en fjøring mellom trekloss og whiteboard-viskeren, for å få hele whiteboard-viskerens kontaktflate inn mot rennen (som vist i figur 4.2). Med denne løsningen er det også tatt hensyn til "force kontroll" av verktøyet når det opereres i programmet rennerensking.





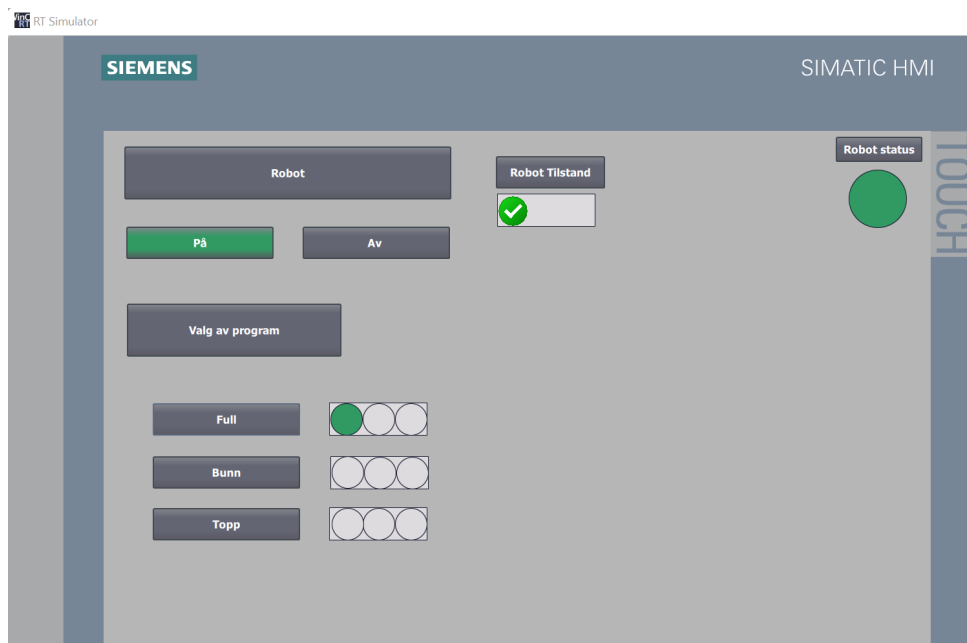
**Figur 4.1:** Rennerensk av renne uten fjøring



**Figur 4.2:** Rennerensk av renne med fjøring

#### 4.1.2 UR5e med HMI

Under simuleringen av prototypen over PROFINET ble det utført kommandoer til UR5e robot fra HMI-panel (som vist i figur 4.3). I kommunikasjon mellom UR5e og HMI-panel er det laget et vedlegg for å forklare PolyScope kode, (se vedlegg C). Vedlegget har til hensikt å få frem funksjonaliteten til ulike brytere på HMI-panelet. Det er også laget et vedlegg D, som forklarer HMI-panelets bryterfunksjonalitet. Robot er programmert til å gå i parkeringsmodus etter hver rennerensk. Om rennen ikke blir rengjort godt nok etter brukerens standard, har brukeren mulighet til å starte ny rennerensk via HMI-panel, enten lav, høy eller fullvask.



**Figur 4.3:** Simulert HMI-panel

Det simulerte HMI-panelet ble laget som et eksempel. Funksjonen på HMI-panelet kan ikke påføre skade på renne eller på UR5e. Kommunikasjonen mellom HMI-panelet og UR5e fungerer tilfredstillende for testingforsøk av rennerensk.

## 4.2 Anbefaling av programvare

For å effektivisere programmering av ulike industrirobot fra ulike leverandører anbefales det et universalt programmeringsverktøy. Det å programmere en robotarm i dag krever programmeringskompetanse til hver enkelt robotleverandør (Slik som beskrevet tidligere i kapittel 2).

Bruk av TIA Portal med Siemens SIMATIC Robot Integrator biblioteket kan fremstå som en forenklet fellesløsning for robotbrukere. Fordelen med SIMATIC-biblioteket er at bruk av andre typer robotleverandører kan bli mer omfattende grunnet de fellesløsningene som SIMATIC Robot Integrator representerer. Personell som skal programmere en ny type robot kan programmere i et standardisert programmeringsspråk. I tillegg kan personell som programmerer få en økt produktivitet, høyere inntjening til bedriften og større spekter av roboter å velge mellom.

## 4.3 Industrirobot, utstyr og komponenter for anlegget ved TTI

I dette delkapittelet vil det ble presentert en anbefaling av indutrirobot, utstyr, komponenter som kan passe til anlegget ved TTI. Fra kapittel 2, ble det presentert ulike roboter fra ABB, KUKA og FANUC. Videre er det konkludert at roboter fra ABB, spesifikt modell ABB IRB6700-175 som den mest funksjonell tjenlige i tilknytning til denne oppgaven.



Figur 4.4: IRB 6700-175 [16]

### 4.3.1 ABB IRB 6700-175

Industriroboten ABB IRB6700-175 har en maksimal rekkevidde på 3050mm fra senter av roboten, som passer valggrunnlaget (som vist i kapittel 2), om at rennen i anlegget ved TTI har en lengde på om lag 7m. Dersom industriroboten blir plassert i midten av rennen, kan den klare å polere og rense bort slaggrester etter tapping på hele rennen. Industriroboten har en løftekapasitet på 175kg, dette gjør at industriroboten er robust og kan håndtere et bredt utvalg av forskjellige verktøy knyttet til robot-griperen. Industriroboten er gulvplassert, har 6-akser og har en beskyttelsesklasse IP67. Ut fra presentasjonen som ble gjort i kapittel 2, ble det tatt hensyn til TTI-anleggets utfordringer som varme, støv og plassmangel. Ved drøfting ble det konkluderte med at ABB IRB6700-175 LEAN-ID sammen med supplement er i stand til å tåle slike utfordringer. Industriroboten bør suppleres med ekstra utstyret “foundry plus 2” og tilleggskomponenter for beskyttelse (se kapittel 2).

### 4.3.2 Beskyttelsesutstyr og komponent til ABB IRB6700-175

Industriroboten ABB IRB6700-175 bør suppleres med varmekåpe og få lufttilskudd under periodene roboten ikke er virksom (Se kapittel 2).

### 4.3.3 Vedlikehold av ABB IRB6700-175

Ved produktmanualen til ABB IRB6700-175 [43], fremvises det en tabell knyttet til industrirobotens vedlikeholdsrutiner. Det er laget en tabell hvor det vises et utdrag av de mest nødvendige vedlikeholdsrutinene fra produktmanual (Se tabell 4.1).

**Tabell 4.1:** *Utdrag av vedlikeholdsrutiner ABB IRB6700 [16]*

Beskrivelse	Tidsintervall
Holde robot ren	Alltid
Sjekke motortetning	Hver 12mnd
Sjekke oljenivå i girboks	Hver 12mnd
Sjekke roboten for skade	Hver 12mnd
Sjekke fjøringer	Hver 12mnd
Stram utvendige bolter	Hver 12mnd
Sjekke gjennomføringer	Hver 12mnd
Sjekke mekaniskstopp	Hver 12mnd
Skifte olje i alle girboksene	Hver 18mnd, eller etter hver 20 000t brukt
Skifte SMB batteripakke	Etter hver 12 000t brukt
Sjekke kabler	Hver 12mnd

### 4.3.4 Kundekontakt med bedrifter

I sammenheng med valg av industrirobot ble det opprettet kontakt med RobotNorge, siden RobotNorge er leverandør av ABB roboter i Norge. RobotNorge er det nærmeste

leverandør for kontakt og leveranse av ABB roboter. Gjennom samtalen med RobotNorge sin salgsleder ble flere spørsmål besvart angående ABB sine industriroboter. Dette medførte at det ble enklere å ta et standpunkt i valget av oppgavens anbefaling med tanke på ABB som robotlevandør. I samtale med salgsleder ble det mottatt prisestimer og avklaringer på hvordan produktet vil håndtere varmen på anlegget. Det ble også gitt forklaring på hvilke produkter som en kunne anbefale videre i oppgaven. Under samtale ble det også opplyst hvilke produkter og tjenester som er inkludert i prisen fra RobotNorge (Se tabell 4.2).

Ved valg av varmekåpe og sikkerhetsbarrierer ble det opprettet kontakt med firmaene Robot Protection Systems og Axellent. Dette ble gjort på grunnlag av samtalen med RobotNorge, der de henviste til produkter som ble brukt i varme industriområder. I samtalene med Robot Protection Systems og Axellent ble det delt nyttig informasjon om priser på produktene og erfaringsdeling i henhold til regelverk som industriroboten må følge.

#### 4.4 Prisantydning

Det vil i dette delkapittelet bli presentert en prisantydning til den forslåtte løsningen i oppgaven. Store deler av prisantydningen er fra samtaler med bedrifter som leverer komponenter, utstyr og materialer. Priser kan variere og tabellen 4.2 som blir fremvist er kun tenkt som en antydning og en cirka pris.

**Tabell 4.2:** *Prisantydning til forelått løsning*

Produkt	Prisantydning (i NOK)
ABB IRB6700-175	400 000,-
Sikkerhetsbarriere	42 820,-
Varmekåpe	45 000,-
<b>Totalt</b>	<b>487 820,-</b>

Det er nødvendig utstyr som ikke er tatt med i prisantydningen for det totale regnskapet. Dette er robot-gripper, luftkjølingsystem, sensorikk og dørlåsesystem for sikkerhetsbarrieren.

## 5 Diskusjon

I dette kapittelet vil det bli redegjort for utfordringer som oppstod underveis samt løsninger på disse utfordringene. Det vil til slutt bli presentert videre arbeid, der det blir lagt frem forbedringer for at robotløsningen skal kunne optimaliseres.

### 5.1 Gripper og UR5e robot

Problemstillingen knyttet til om en whiteboard-visker som arbeidsverktøy ble diskutert i forkant av oppgaven. En whiteboard-visker vil ikke simulere arbeidsverktøyet som blir brukt hos TTI. Dette fordi operatører hos TTI etter dagens standard bruker spett til fjerning av slaggrester i produksjonsrennen. For å simulere bruk av en spett må arbeidsverktøyet whiteboard-visker eventuelt byttes ut, og programvaren må programmeres på nytt. En whiteboard-visker har ikke har det samme funksjonene som en spett, men har fungert for bruk i simuleringsforsøk av rennerensk.

### 5.2 Kommunikasjonsprotokoll

Oppkoblingen av UR5e og Siemens har foregått over kommunikasjonsprotokollen PROFINET (som nevnt i kapittel 4.1). Alternative kommunikasjonsprotokoller som kan brukes for oppkobling av UR5e og Siemens PLS:

- PROFINET IO
- MODBUS
- EtherNet IP

Kommunikasjonsprotokollene som er nevnt over kan alle brukes ved sammenkobling av UR5e og PLS. Mest brukte kommunikasjonsprotokollene er PROFINET og MODBUS, forskjellen er at PROFINET er full dupleks og MODBUS er halv dupleks. Samt at PROFINET er raskere enn MODBUS, har flere funksjoner som å overføre informasjon i sanntid. Anbefaling av kommunikasjonsprotokoll for videre bruk er PROFINET.

### 5.3 Utfordringer

Under arbeid med oppgaven har det vært lite informasjon å finne om oppsett for oppkobling av PROFINET mellom UR5e robot og Siemens PLS. Det ble brukt en god del tid på å gjøre forarbeid på hvordan koble seg opp mot PROFINET, og koble sammen Siemens PLS og UR5e i TIA Portal. Universal Robots har i ettertid (9.mai.2022) laget en manual for sammenkoblingen av UR5e og Siemens PLS i TIA Portal. Som gjør at prosessen med sammenkoblingen og oppkobling mot PROFINET blir en enklere prosess. For i manualen for PROFINET guide på Universal Robots står alle prosessene steg for steg.

Under arbeid med oppgaven har det vært lite informasjon å finne i søk av roboter som opererer under farlige, varme og skitne områder. Informasjonen som er funnet har vært fra nettsider og artikler uten en troverdig kilde eller fra rapporter som har lite grunnlag og ikke har kunnet blitt videreført til denne oppgaven.

## 5.4 Videre arbeid

I dette delkapittelet vil det bli presentert videre arbeid. Basert på erfaringer underveis gjort i oppgaven, kan det trekkes frem tre teknologiske kategorier som med fordel bør kunne utvikles i tilknytning til oppgavens problemstilling: Utvikling av programkode, arbeidsverktøy og håndholdt kontroller.

### 5.4.1 Utvikling av programkode

Programkode til prototype kan videre utvikles for å optimalisere programmet rennerensk. Programkode for robot er i dag simplifisert med programkode for kun enkel rennerensk, men kan utvikles til å identifisere en slaggest eller et objekt i rennen ved hjelp av kamera eller sensorer. Det kan også utvikles programkode for robot til å følge et koordinatsystem under rennerensk, slik at det muliggjør en enkel kalibrering av robot dersom rennen blir flyttet på.

Prototypen kan optimaliseres med flere sikkerhetsfunksjoner på anlegget. Dette kan være å montere en ekstern nødstopp eller sensor med sikringstopp. Samt kan det vurderes å legge til programkode for redusering av fart om robot er for nærme personer under rennerensk.

### 5.4.2 Arbeidsverktøy

Vurdere alternative løsninger for arbeidsverktøy som kan være gunstig for rensing av slaggrester i stålindustrien. Roterende arbeidsverktøy t.eks. slagbor, roterende meisel eller avrundet fil.

Se på løsninger med robot som bruker slagfunksjoner med en sprett samt videre undersøke “force control” av arbeidsverktøy for å ikke kunne skade produksjonsrennen når det blir utført rennerensk.

Vurdere alternative løsninger for arbeidsverktøyet til prototypen ved rennerensk. Prototypens verktøy er under denne oppgaven en whiteboard-visker, men kan se på alternative løsninger et annet arbeidsverktøy.

### 5.4.3 Håndholdt kontroller

En håndholdt kontroller kan brukes for fjernstyring av robot. Håndholdt kontroller sammen med en robot gir muligheter for mer fleksibilitet på hvilke oppgaver som kan utføres. Bruk av håndholdt kontroller gjør at brukeren kan utføre ulike handlinger på en sikker avstand. Om robot ikke får fjernet slag på et spesifikt område etter programvaren har kjørt, så kan en programmere inn styring på en håndholdt kontroller. En ferdigprogrammert kontroller vil gjøre robot mer fleksibel og kan gi muligheter som:

- HMS sikker avstand i farlige omgivelser
- Fjerne slag på spesifikke punkt i rennen
- Fleksibel styring av robot

## 6 Konklusjon

Formålet med oppgaven var å gjøre produksjon av titan og jern mindre risikofylt samt å finne en løsning basert på dagens prosess. Oppgaven var å se på muligheter for å bruke robot til å utføre rennerensk i en produksjonsrenne ved TTI, foreslå robot som kan brukes og å lage en nedskalert prototype. I oppgaven har det blitt vurdert ulike typer robotiserte løsninger basert på reduksjon av sikkerhetsrisiko og opprettholdelse av produktets kvalitet.

Simulering av prototypen ble laget med PROFINET kommunikasjon og satt sammen ved hjelp av en datasvitsj. Samt at programkode er laget ved bruk av Siemens TIA Portal og programvaren PolyScope. Industriroboten som ble anbefalt er ABB IRB6700-175. Roboten har 6 akser, lang nok rekkevidde for anlegget til TTI samt en robusthet som kan passe utfordrende arbeidsområder.

Under løsning av oppgaven er det satt opp en PLS og et HMI-panel som kan være til inspirasjon til fremtidig løsning. HMI-panelet kan brukes til å styre robotens utførelse av rennerensk i fremtiden.

Det har blitt foreslått en anbefaling av robot samt det er blitt laget en prototype for simulering på laboratoriet ved HVL Haugesund. Den foreslåtte løsningen vil kunne føre til rensing og polering av slaggrester i produksjonsrennen hos TTI og senke belastningen for en operatør som jobber etter dagens løsning med manuelt arbeid. Løsningsforslaget innebærer slik sett at sikkerhetsrisikoen blir redusert, samt at kvalitet på produktet opprettholdes.

Løsningsforslaget har ikke hatt til hensikt å opprette en fullstendig kostnadsanalyse ut over prisantydning, men det er mulig å anta at bruk av automatisert robotløsning kan ha en kostnadsreduserende effekt satt opp mot dagens løsning.

## Kilder

- [1] Arbeidstilsynet. *Arbeid i varme områder*. 2000. (Internett). Tilgjengelig: <https://www.arbeidstilsynet.no/tema/temperatur/>. (Sjekket 24.01.2022).
- [2] Aftenposten. *Slaggtapping*. Aug. 2016. (Internett). Tilgjengelig: <https://www.af tenposten.no/viten/i/xVq3Q/smelteverk-dropper-kull-helt-frivillig>. (Sjekket 02.02.2022).
- [3] Wiredworkers. *UR5e robot*. (u.å). (Internett). Tilgjengelig: <https://wiredworker s.io/universal-robots-ur5/>. (22.03.2022).
- [4] Digitalmuseumet. *Slagg eksempel*. (u.å). (Internett). Tilgjengelig: <https://digita ltmuseum.no/021028688758/slagg/media?slide=0>. (Sjekket 20.01.2022).
- [5] Goodtech. *Historikk goodtech*. (u.å). (Internett). Tilgjengelig: <https://www.goodt ech.no/om-goodtech/historikk/>. (Sjekket 23.01.2022).
- [6] Goodtech. *Metall og gruvedrift*. (u.å). (Internett). Tilgjengelig: <https://www.good tech.no/industri/metall-og-gruvedrift/>. (Sjekket 11.04.2022).
- [7] Thomas Førde. *TiZir avd Tyssedal*. Des. 2021. (Internett). Tilgjengelig: <https://w ww.tu.no/artikler/kutter-utslippene-med-85-prosent-gar-fra-kull-til-hydrogen/515944?key=HQWdjIcQ>.
- [8] Morteza Ghobakhloo. «Industry 4.0, digitization, and opportunities for sustainabi- lity». I: *Journal of cleaner production* 252 (2020), s. 2.
- [9] Mark Fairchild. *Types of Industrial Robots and Their Different Uses*. Aug. 2021. (Internett). Tilgjengelig: <https://www.howtorobot.com/expert-insight/indus trial-robot-types-and-their-different-uses>.
- [10] Univeral Robots. *ABB becomes first company in world to sell 100,000 robots*. Okt. 2017. (Internett). Tilgjengelig: <https://www.universal-robots.com/blog/know -your-machine-industrial-robots-vs-cobots/>.
- [11] TheRobotReport. *Top 5 countries using industrial robots in 2018*. Sep. 2019. (In- ternett). Tilgjengelig: <https://www.therobotreport.com/top-5-countries-usi ng-industrial-robots-2018/>.
- [12] Wei Ji og Lihui Wang. «Industrial robotic machining: a review». I: *The International Journal of Advanced Manufacturing Technology* 103.1 (2019), s. 1.
- [13] Automate. *Topp 10 industri robot leverandører i verden*. Feb. 2021. (Internett). Tilgjengelig: <https://www.automate.org/news/10-industrial-robot-compani es-that-lead-the-industry>.
- [14] ABB. *ABB becomes first company in world to sell 100,000 robots*. Feb. 2002. (In- ternett). Tilgjengelig: <https://new.abb.com/news/detail/13638/abb-becomes- first-company-in-world-to-sell-100000-robots>.
- [15] Automate. *ABB IRB 6700 Arbeidsområde*. (u.å). (Internett). Tilgjengelig: <https://www.automate.org/news/abb-introduces-the-irb-1660id-its-latest-co mpact-robot-for-arc-welding-and-machine-tending-applications>. (Sjekket 26.01.2022).
- [16] ABB. *IRB 6700*. pdf. Jan. 2019. (Internett). Tilgjengelig: <https://library.e.a bb.com/public/161cc19a854b47959a9097f9086d1fce/IRB6700-presentation- 9AKK105713A9749-RevF.pdf>.



- [17] ABB. *The world's most used offline programming tool for robotics*. (u.å). (Internett). Tilgjengelig: <https://new.abb.com/products/robotics/robotstudio#>. (Sjekket 12.04.2022).
- [18] ABB. *ABB IRB 6700 Datasheet*. (u.å). (Internett). Tilgjengelig: <https://new.abb.com/products/robotics/cs/prumyslove-roboty/irb-6700/irb-6700>. (Sjekket 27.01.2022).
- [19] KUKA. *Generell informasjon om KUKA*. (u.å). (Internett). Tilgjengelig: <https://www.kuka.com/en-ch/company/about-kuka>. (Sjekket 03.02.2022).
- [20] KUKA. *KR120 R2700-2*. pdf. Okt. 2021. (Internett). Tilgjengelig: [https://www.kuka.com/-/media/kuka-downloads/imported/6b77eecacfe542d3b736af377562e0000325899\\_en.pdf?rev=2eeba4ae839a4d5e916fe15f02d42d4e&hash=D846037A1480FC646C8F8399064D5799](https://www.kuka.com/-/media/kuka-downloads/imported/6b77eecacfe542d3b736af377562e0000325899_en.pdf?rev=2eeba4ae839a4d5e916fe15f02d42d4e&hash=D846037A1480FC646C8F8399064D5799).
- [21] KUKA. *Kuka Quantec Datasheet*. (u.å). (Internett). Tilgjengelig: <https://www.kuka.com/en-ch/products/robotics-systems/industrial-robots/kr-quantec>. (Sjekket 27.01.2022).
- [22] KUKA. *Programming KUKA robot*. (u.å). (Internett). Tilgjengelig: <https://www.kuka.com/en-de/services/engineering/application-and-robot-programming>. (Sjekket 03.02.2022).
- [23] KUKA. *KUKA.PLC mxAutomation*. (u.å). (Internett). Tilgjengelig: <https://www.kuka.com/en-de/products/robot-systems/software/hub-technologies/kuka,-d-,plc-mxautomation>. (Sjekket 26.05.2022).
- [24] FANUC. *FANUC produces 750,000th robot*. Jul. 2021. (Internett). Tilgjengelig: <https://www.fanuc.eu/es/en/who-we-are/news/eu-750k-robots-07-2021>.
- [25] FANUC. *Fanuc M-710 Series*. (u.å). (Internett). Tilgjengelig: <https://www.fanuc.eu/es/en/robots/robot-filter-page/m-710-series>. (Sjekket 21.04.2022).
- [26] FANUC. *Fanuc M-900 Series*. (u.å). (Internett). Tilgjengelig: <https://www.fanuc.eu/es/en/robots/robot-filter-page/m-900-series>. (Sjekket 21.04.2022).
- [27] FANUC. *ROBOGUIDE Intelligent 3D simulation*. pdf. (u.å). (Internett). Tilgjengelig: [https://www.fanuc.eu/~/\\_media/files/pdf/products/robots/brochures/mbr-02857-ro%20roboguide%20brochure/roboguide%20brochure%20en.pdf?la=en](https://www.fanuc.eu/~/_media/files/pdf/products/robots/brochures/mbr-02857-ro%20roboguide%20brochure/roboguide%20brochure%20en.pdf?la=en). (Sjekket 22.04.2022).
- [28] Mid-mountain. *Protecting robotic equipment in extreme environments*. (u.å). (Internett). Tilgjengelig: <https://mid-mountain.com/protecting-robotic-equipment-in-extreme-environments/>. (Sjekket 04.02.2022).
- [29] Robotprotection. *Heat protection - Foundry - forging*. (u.å). (Internett). Tilgjengelig: <https://www.robotprotection.com/protection-systems/heat-protection/>. (Sjekket 11.02.2022).
- [30] Schmachtl. *A frameless modular protective fencing system*. (Internett). Tilgjengelig: <https://www.schmachtl.at/en/news/news/safety-adaptaguard-frameless-fence/>. (Sjekket 10.04.2022).
- [31] ISO Central Secretary. *Safety of machinery*. en. Standard ISO/TS 13857:2019. Geneva, CH: International Organization for Standardization, 2019. (Internett). Tilgjengelig: <https://www.iso.org/standard/69569.html>.
- [32] Axellent.com. *X-Guard*. (u.å). (Internett). Tilgjengelig: <https://www.axellent.com/products/machine-guarding/x-guard/>. (Sjekket 14.04.2022).

- [33] Roboticsimulationservices. *Maintenance and Repair of Industrial Robots: What You Should Know*. (u.å). (Internett). Tilgjengelig: <https://roboticsimulationservices.com/maintenance-and-repair-of-industrial-robots-what-you-should-know/>. (Sjekket 04.03.2022).
- [34] Siemens. *Software in TIA Portal*. (u.å). (Internett). Tilgjengelig: <https://new.siemens.com/us/en/products/automation/industry-software/automation-software/tia-portal/software.html>. (Sjekket 24.03.2022).
- [35] Siemens. *Cooperation for the multi-vendor integration of industrial robots into the TIA Portal*. pdf. Nov. 2021. (Internett). Tilgjengelig: <https://assets.new.siemens.com/siemens/assets/api/uuid:64c5e885-1a25-4b8c-862b-d7d305884d55/HQDIPR202011166062EN.pdf>.
- [36] Siemens. *Software in TIA Portal*. (u.å). (Internett). Tilgjengelig: <https://new.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal/highlights/robot-integration.html>. (Sjekket 08.04.2022).
- [37] Profibus. *A new profile for robot systems*. Feb. 2021. (Internett). Tilgjengelig: <https://www.profibus.com/newsroom/press-news/a-new-profile-for-robot-systems>.
- [38] Universal robots. *MAKE ROBOT TECHNOLOGY ACCESSIBLE TO ALL*. (u.å). (Internett). Tilgjengelig: <https://www.universal-robots.com/about-universal-robots/our-history/>. (Sjekket 23.03.2022).
- [39] Universal robots. *e-Serien*. (u.å). (Internett). Tilgjengelig: <https://www.universal-robots.com/no/nedlastingssenter/#/e-series>. (Sjekket 23.03.2022).
- [40] ISO Central Secretary. *Robots and robotic devices — Collaborative robots*. en. Standard ISO/TS 15066:2016. Geneva, CH: International Organization for Standardization, 2016. (Internett). Tilgjengelig: <https://www.iso.org/standard/62996.html>.
- [41] Universal robots. *Universal robots launches e-series*. (u.å). (Internett). Tilgjengelig: <https://www.universal-robots.com/about-universal-robots/news-centre/universal-robots-launches-e-series-setting-a-new-standard-for-collaborative-automation-platforms/>. (Sjekket 24.03.2022).
- [42] Universal Robots. *UR5e User Manual*. pdf. (u.å). (Internett). Tilgjengelig: [https://s3-eu-west-1.amazonaws.com/ur-support-site/115744/99422\\_UR5e\\_User\\_Manual\\_no\\_Global.pdf](https://s3-eu-west-1.amazonaws.com/ur-support-site/115744/99422_UR5e_User_Manual_no_Global.pdf). (Sjekket 29.04.2022).
- [43] «Product manual IRB 6700». I: (mar. 2022). (Internett). Tilgjengelig: <https://search.abb.com/library/Download.aspx?DocumentID=3HAC044266-001&LanguageCode=en&DocumentPartId=&Action=Launch>. (Sjekket 30.04.2022).
- [44] Universal Robots. *pn-iomessage*. pdf. Nov. 2021. (Internett). Tilgjengelig: <https://s3-eu-west-1.amazonaws.com/ur-support-site/20596/pn-iomessage.pdf>. (Sjekket 13.02.2022).
- [45] Universal Robots. *PROFINET GUIDE*. Nov. 2021. (Internett). Tilgjengelig: <https://www.universal-robots.com/articles/ur/interface-communication/profinet-guide/>.

## A PROFINET kommunikasjon

Vedlegget vil gå ut på forklaring av datatypene fra UR5 til PLS over PROFINET forbindelsen. Kjennskap til ulike datatyper i programmet er nødvendig for å programmere og sammensette PLS programmet med UR5 over PROFINET. For å forstå betydningene av noen av de ulike variabelklassene som står i figur A.3 kan figur A.1 brukes for å forstå de ulike forkortelsene som blir brukt i figur A.3

AOM	Analog output mask
AOT	Analog output types
ES	Is emergency stopped
FT	Is fault
NO	Is normal mode
PB	Is power button pressed
PR	Is program running
PS	Is protective stopped
PW	Is power on
RC	Is recovery mode
RD	Is reduced mode
RES	Is robot emergency stopped
SES	Is system emergency stopped
SS	Is safeguard stopped
SSM	Speed slider fraction mask
ST	Is stopped due to safety
TAIT	Tool analog input types
TB	Is teach button pressed
TDI	Tool digital inputs
TDO	Tool digital outputs
TDOM	Tool digital output mask
VL	Is violation

**Figur A.1:** Navn på ulike datatyper [44]

Datatypene i tabellen over beskriver de ulike tilstandene for programmeringen i TIA Portal, det er for å framvise tilstandene oversiktlig i TIA Portal. PLSen skal kontrollere UR5e ut i fra informasjonen som er innhentet til TIA Portal. Synkroniseringen av informasjon mellom de ulike enhetene foregår over bruk av digitale og bineære dataene.

I TIA portal er det predefinerte datatyper som er standard ved programmering i TIA Portal. For å programmere UR5e robotene så må det brukes datatyper som ikke allerede finnes i datatypekatalogen, dette er datatyper som: int, byte, bool og date og osv. For å få inn denne datatypen må datastruktur-filen lastes ned fra Univeral robots nettside for å få tilgang til de unike datatypene i TIA Portal. (nettsiden finnes her: [45]). For å få tilgang til datatypene fra datastruktur filen må AWL filen lastes ned fra kilden [45]. AWL filen inneholder UDT som TIA Portal trenger for å ha muligheten til å programmere ulike kommandoer til UR5e.

Innholdet i datastrukturfilen (er vist i figur A.2).

Datatypes ved navn "UR T20" representerer dataoverføringer fra roboten,

Datatypes ved navn "UR O2T" representerer dataoverføringer fra PLS.

Forklaring av ulike datatypes vises i figur A.4 - A.13 som beskriver strukturfilen som er implementert i figur A.3. Hvor kommunikasjonen med bruk av binære og digitale signaler mellom UR5e og PLS er det mulig å verifisere om robot er slått på (som vist på linje 13 på figur 3.2).

UR5e_tags							
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...
1	UR_Input_State	*UR_1_T20_State*	%I0.0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	UR_Input	*UR_2_T20_IO*	%I32.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	UR_Input_Joint	*UR_3_T20_Joints*	%I100.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	UR_Input_TCP	*UR_4_T20_TCP*	%I204.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	UR_Input_General_Purpose_Bits	*UR_5_T20_BitRegisters*	%I280.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	UR_Input_General_Purpose_Ints	*UR_6_T20_IntRegisters*	%I288.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	UR_Input_General_Purpose_Floats	*UR_7_T20_FloatRegisters*	%I384.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	UR_Output	*UR_8_O2T_Robot_IO*	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	UR_Output_General_Purpose_Register_1	*UR_9_O2T_Registers*	%Q24.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	UR_Output_General_Purpose_Register_2	*UR_9_O2T_Registers*	%Q124.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figur A.2: Tag tabell som gir informasjon om UR5e

UR5e_tags										
Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Monitor value	Supervision	Comment	
UR_Input_State	*UR_1_T20_State*	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Robot	UR_T20_Robot	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Controller major version	USInt	%I80	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5			
Controller minor version	USInt	%I81	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	11			
Reserved	ULint	%I82	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Robot mode	USInt	%I84	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	7			
Real time machine seconds	USInt	%I85	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	33			
Real time machine milliseconds	ULint	%I86	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	431			
Real time machine minutes	USInt	%I88	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	50			
Real time machine hours	USInt	%I89	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Real time machine days	ULint	%I90	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Robot current [A]	Real	%I12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.7553206			
PW: Is power on	Bool	%I16.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	TRUE			
PR: Is program running	Bool	%I16.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE			
TB: Is teach button pressed	Bool	%I16.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE			
PB: Is power button pressed	Bool	%I16.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE			
Reserved_4	Bool	%I16.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE			
Reserved_5	Bool	%I16.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE			
Reserved_6	Bool	%I16.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE			
Reserved_7	Bool	%I16.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE			
Reserved_8_15	USInt	%I17	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Reserved_16_31	ULint	%I18	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Speed slider fraction	Real	%I20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1.0			
Safety	UR_T20_Safety	%I24.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
UR_Input	*UR_2_T20_IO*	%I32.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
UR_Input_Joint	*UR_3_T20_Joints*	%I100.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Joint position [rad]	Array[0..5] of Real	%I100.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Joint position [rad][0]	Real	%I100.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-3.263365			
Joint position [rad][1]	Real	%I101.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-1.447734			
Joint position [rad][2]	Real	%I102.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-1.426083			
Joint position [rad][3]	Real	%I103.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-1.956239			
Joint position [rad][4]	Real	%I104.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-4.224273			
Joint position [rad][5]	Real	%I105.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-4.737816			
Joint velocity [rad/s]	Array[0..5] of Real	%I124.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Joint current [A]	Array[0..5] of Real	%I148.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Joint temperature [°C]	Array[0..5] of Real	%I172.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Joint mode	Array[0..5] of USInt	%I196.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
Reserved_9	ULint	%I202.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
UR_Input_TCP	*UR_4_T20_TCP*	%I204.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
UR_Input_General_Purpose_Bits	*UR_5_T20_BitRegisters*	%I280.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
UR_Input_General_Purpose_Ints	*UR_6_T20_IntRegisters*	%I288.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
UR_Input_General_Purpose_Floats	*UR_7_T20_FloatRegisters*	%I384.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			
UR_Output	*UR_8_O2T_Robot_IO*	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0			

Figur A.3: Informasjons overføring fra UR5e til tag tabell over PROFINET

UR 1 T2O State : T (Robot) -> O (PLC) - Robot state and safety mode (32 bytes)

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Group
0	Controller major version (uint)				Controller minor version (uint)				Reserved																Data type (.udt file)								
32	Robot mode (uint)								Real time machine seconds (uint)								Real time machine milliseconds (uint)								Robot UR_T2O_Robot								
64	Real time machine minutes (uint)								Real time machine hours (uint)								Real time machine days (uint)																
96	Robot current (float) [A]																Reserved																
128	PW	PR	TB	PB	Reserved																												
160	Speed slider fraction (float)																Reserved																
192	Safety mode (uint)																Reserved																Safety UR_T2O_Safety
224	NO	RD	PS	RC	SS	SES	RES	ES	VL	FT	ST	Reserved																					

Figur A.4: Beskrivelse av datastrukturen [44]

UR 2 T2O IO : T (Robot) -> O (PLC) - I/Os of control box and tool (68 bytes)

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Group
256	Standard digital inputs				Configurable digital inputs				Standard digital outputs				Configurable digital outputs				Data type (.udt file)																
288	Analog I/O Types								Reserved								I/Os UR_T2O_IOs																
320	Standard analog input 0 (float) [A or V]																																
352	Standard analog input 1 (float) [A or V]																																
384	Standard analog output 0 (float) [A or V]																																
416	Standard analog output 1 (float) [A or V]																																
448	I/O current (float) [A]																																
480	Euromap67 input bits																																
512	Euromap67 output bits																																
544	Euromap67 24V voltage (float) [V]																																
576	Euromap67 24V current (float) [A]																																
608	Tool mode (uint)								Reserved								Tool UR_T2O_Tool																
640	TDI	Reserved				TDO	Reserved				TAIT	Reserved																					
672	Tool analog input 0 (float) [A or V]																																
704	Tool analog input 1 (float) [A or V]																																
736	Tool output voltage (float) [V]																																
768	Tool current (float) [A]																																

Figur A.5: Beskrivelse av datastrukturen [44]

UR 3 T2O Joints: T (Robot) -> O (PLC) - Joint states (104 bytes)

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Group
800	Joint 0 position (float) [rad]																Joints																
832	Joint 1 position (float) [rad]																																
864	Joint 2 position (float) [rad]																																
896	Joint 3 position (float) [rad]																																
928	Joint 4 position (float) [rad]																																
960	Joint 5 position (float) [rad]																																
992	Joint 0 velocity (float) [rad/s]																																
1024	Joint 1 velocity (float) [rad/s]																																
1056	Joint 2 velocity (float) [rad/s]																																
1088	Joint 3 velocity (float) [rad/s]																																
1120	Joint 4 velocity (float) [rad/s]																																
1152	Joint 5 velocity (float) [rad/s]																																
1184	Joint 0 current (float) [A]																																
1216	Joint 1 current (float) [A]																																
1248	Joint 2 current (float) [A]																																
1280	Joint 3 current (float) [A]																																
1312	Joint 4 current (float) [A]																																
1344	Joint 5 current (float) [A]																																
1376	Joint 0 temperature (float) [°C]																																
1408	Joint 1 temperature (float) [°C]																																
1440	Joint 2 temperature (float) [°C]																																
1472	Joint 3 temperature (float) [°C]																																
1504	Joint 4 temperature (float) [°C]																																
1536	Joint 5 temperature (float) [°C]																																
1568	Joint 0 mode (uint)				Joint 1 mode (uint)				Joint 2 mode (uint)				Joint 3 mode (uint)																				
1600	Joint 4 mode (uint)				Joint 5 mode (uint)				Reserved																								

Figur A.6: Beskrivelse av datastrukturen [44]

UR 4 T2O TCP : T (Robot) -> O (PLC) - TCP (76)

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Group
1632	TCP position X (float) [m]																TCP																
1664	TCP position Y (float) [m]																																
1696	TCP position Z (float) [m]																																
1728	TCP position RX (float)																																
1760	TCP position RY (float)																																
1792	TCP position RZ (float)																																
1824	TCP velocity X (float) [m/s]																																
1856	TCP velocity Y (float) [m/s]																																
1888	TCP velocity Z (float) [m/s]																																
1920	TCP velocity RX (float)																																
1952	TCP velocity RY (float)																																
1984	TCP velocity RZ (float)																																
2016	TCP force X (float) [N]																																
2048	TCP force Y (float) [N]																																
2080	TCP force Z (float) [N]																																
2112	TCP torque X (float) [N.m]																																
2144	TCP torque Y (float) [N.m]																																
2176	TCP torque Z (float) [N.m]																																
2208	TCP force scalar (float) [N]																																

Figur A.7: Beskrivelse av datastrukturen [44]

UR 5 T2O BitRegisters : T (Robot) -> O (PLC) - General purpose bit registers (8 bytes)

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Group
2240	Bit output registers 0 - 31																															Bit registers	
2272	Bit output registers 32 - 63																																

Figur A.8: Beskrivelse av datastrukturen [44]

UR 6 T2O IntRegisters : T (Robot) -> O (PLC) - General purpose int registers (96 bytes)

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Group
2304	Int output register 0 (int)																															Int registers	
2336	Int output register 1 (int)																																
2368	Int output register 2 (int)																																
2400	Int output register 3 (int)																																
2432	Int output register 4 (int)																																
2464	Int output register 5 (int)																																
2496	Int output register 6 (int)																																
2528	Int output register 7 (int)																																
2560	Int output register 8 (int)																																
2592	Int output register 9 (int)																																
2624	Int output register 10 (int)																																
2656	Int output register 11 (int)																																
2688	Int output register 12 (int)																																
2720	Int output register 13 (int)																																
2752	Int output register 14 (int)																																
2784	Int output register 15 (int)																																
2816	Int output register 16 (int)																																
2848	Int output register 17 (int)																																
2880	Int output register 18 (int)																																
2912	Int output register 19 (int)																																
2944	Int output register 20 (int)																																
2976	Int output register 21 (int)																																
3008	Int output register 22 (int)																																
3040	Int output register 23 (int)																																

Figur A.9: Beskrivelse av datastrukturen [44]

UR 7 T2O FloatRegisters : T (Robot) -> O (PLC) – General purpose float registers (96 bytes)

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Group
3072														Float output register 0 (float)													Float registers						
3104														Float output register 1 (float)																			
3136														Float output register 2 (float)																			
3168														Float output register 3 (float)																			
3200														Float output register 4 (float)																			
3232														Float output register 5 (float)																			
3264														Float output register 6 (float)																			
3296														Float output register 7 (float)																			
3328														Float output register 8 (float)																			
3360														Float output register 9 (float)																			
3392														Float output register 10 (float)																			
3424														Float output register 11 (float)																			
3456														Float output register 12 (float)																			
3488														Float output register 13 (float)																			
3520														Float output register 14 (float)																			
3552														Float output register 15 (float)																			
3584														Float output register 16 (float)																			
3616														Float output register 17 (float)																			
3648														Float output register 18 (float)																			
3680														Float output register 19 (float)																			
3712														Float output register 20 (float)																			
3744														Float output register 21 (float)																			
3776														Float output register 22 (float)																			
3808														Float output register 23 (float)																			

Figur A.10: Beskrivelse av datastrukturen [44]

UR 8 O2T Robot IO : O (PLC) -> T (Robot) – Speed slider and electrical outputs (24 bytes)

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Group Data type (.udt file)
0	Reserved																															Robot	
32	Speed slider fraction (float)																															UR_O2T_Robot	
64	Standard digital output mask					Configurable digital output mask					Standard digital outputs					Configurable digital outputs					Outputs UR_O2T_IOs												
96	AOM			Reserved		AOT			Reserved		TDOM			Reserved		TDO			Reserved														
128	Standard analog output 0 (float) [0..1]																																
160	Standard analog output 1 (float) [0..1]																																

Figur A.11: Beskrivelse av datastrukturen [44]

UR 9 O2T Req1 : O (PLC) -> T (Robot) – General Purpose registers (100 bytes)

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Group Data type (.udt file)
192	Bit input registers 0 - 31																															Bit registers	
224	Int input register 0 (int)																															Int registers UR_O2T_ints	
256	Int input register 1 (int)																																
288	Int input register 2 (int)																																
320	Int input register 3 (int)																																
352	Int input register 4 (int)																																
384	Int input register 5 (int)																																
416	Int input register 6 (int)																																
448	Int input register 7 (int)																																
480	Int input register 8 (int)																																
512	Int input register 9 (int)																																
544	Int input register 10 (int)																																
576	Int input register 11 (int)																																
608	Float input register 0 (float)																															Float registers UR_O2T_floats	
640	Float input register 1 (float)																																
672	Float input register 2 (float)																																
704	Float input register 3 (float)																																
736	Float input register 4 (float)																																
768	Float input register 5 (float)																																
800	Float input register 6 (float)																																
832	Float input register 7 (float)																																
864	Float input register 8 (float)																																
896	Float input register 9 (float)																																
928	Float input register 10 (float)																																
960	Float input register 11 (float)																																

Figur A.12: Beskrivelse av datastrukturen [44]

UR 10 O2T Req2 : O (PLC) -> T (Robot) - General Purpose registers (100 bytes)

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Group	Data type (.udt file)
992	Bit input registers 32 - 63																															Bit registers		
1024	Int input register 12 (int)																															Int registers UR_O2T_ints		
1056	Int input register 13 (int)																																	
1088	Int input register 14 (int)																																	
1120	Int input register 15 (int)																																	
1152	Int input register 16 (int)																																	
1184	Int input register 17 (int)																																	
1216	Int input register 18 (int)																																	
1248	Int input register 19 (int)																																	
1280	Int input register 20 (int)																																	
1312	Int input register 21 (int)																																	
1344	Int input register 22 (int)																																	
1376	Int input register 23 (int)																																	
1408	Float input register 12 (float)																															Float registers UR_O2T_floats		
1440	Float input register 13 (float)																																	
1472	Float input register 14 (float)																																	
1504	Float input register 15 (float)																																	
1536	Float input register 16 (float)																																	
1568	Float input register 17 (float)																																	
1600	Float input register 18 (float)																																	
1632	Float input register 19 (float)																																	
1664	Float input register 20 (float)																																	
1696	Float input register 21 (float)																																	
1728	Float input register 22 (float)																																	
1760	Float input register 23 (float)																																	

Figur A.13: Beskrivelse av datastrukturen [44]

For å kunne programmere kommandoer fra PLS til UR5 så har det blitt lastet ned 2 AWL filer som beskrevet tidligere i vedlegget. AWL filene inneholdt udt datatyper som måtte brukes for å programmere i functions eller functions blokk. Datablokkene DB100 og DB101 har ingen funksjon om det ikke blir sammenkoblet med “tagsene” i figur A.2.

DB101 brukes til å sende data til UR5e. For å sende ulike kommandoer til UR5e fra PLS. Ved bruken av bitregisteret til UR5e brukes det i TIA Portal i oppgaven datatypen “UDT 164”. Som gir muligheten for eksempelvis å starte et program via TIA Portal med bruk av et HMI-panel.

DB100									
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	...	Comment
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	UR_T2O_Robot	*UDT150*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Robot
3	UR_T2O_Safety	*UDT151*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Safety
4	UR_T2O_IO	*UDT152*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IOs
5	UR_T2O_Tool	*UDT153*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tool
6	UR_T2O_Joints	*UDT154*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Joints
7	UR_T2O_TCP	*UDT155*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	TCP
8	UR_BitRegisters	*UDT156*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Bit output
9	UR_IntRegisters	*UDT157*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Int output
10	UR_FloatRegisters	*UDT158*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Float output

Figur A.14: Datablokk 100

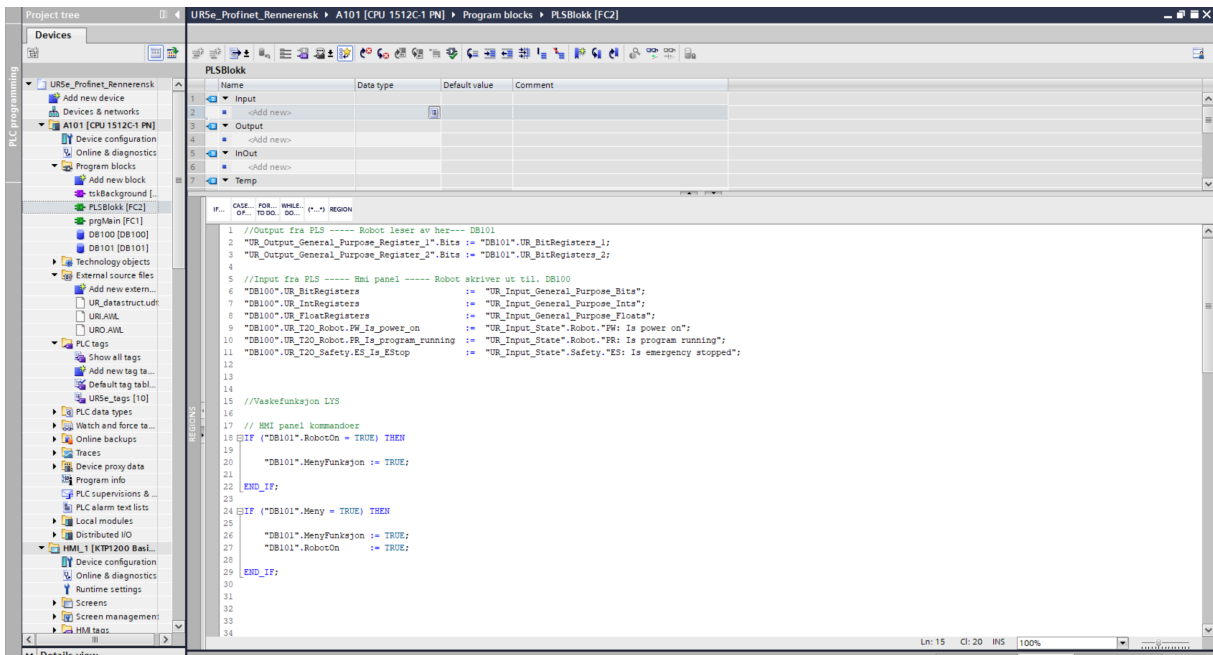


DB101										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision	Comment
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	UR_O2T_Robot	*UDT160*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Robot
3	UR_O2T_Outputs	*UDT161*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Outputs
4	UR_BitRegisters_1	*UDT162*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Bit input
5	UR_IntRegisters_1	*UDT163*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Int input
6	UR_FloatRegisters_1	*UDT164*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Float input
7	UR_BitRegisters_2	*UDT165*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Bit input
8	UR_IntRegisters_2	*UDT166*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Int input
9	UR_FloatRegisters_2	*UDT167*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Float input

Figur A.15: Datablokk 101

## B Kode for sammenkobling

Koden som er brukt ved sammenkobling av prototypen inne på TIA Portal ved bruk av datablokkene og tags som beskrevet i vedlegg A.



Figur B.1: Kode for sammenkobling av datablokker og UR5e tags

## C Kode for UR5e robot

### Relevante dokument

Relevant dokument: vedlegg A.

Koden som er brukt ved sammenkoblingen mellom prototype og TIA Portal kan en finne i vedlegg A. Vedlegget har til hensyn å gi en oppklaring med bruk av datablokkene og tagger i TIA Portal.

Dette vedlegget har til hensyn i å vise frem programkode for roboten brukt i oppgaven. Programkoden er programmert i programmet PolyScope.

Kode som vises i figur C.1, starter fra venstre og kan leses mot høyre. Programkoden starter med å lese boolsk inngangs register - vente til inngangs registeret er mottatt før den aktiverer gripper.

Rennerenssk programmene som er programmert inn er bunn, full og topp- polering er skilt inn i tre programkoder og hver enkelte program har fått tildelt en egen "if-løkke". Programmet mottar en boolsk variabel fra PLS for en av de valgte rennerenssk programmene, og deretter starter det valgte programmet.

Når programmet starter, så er det programmert inn at robot plukker opp verktøy før den starter rennerenssk. Dette gjelder også når programmet er fullført, da leverer roboten verktøyet før den avslutter.

<pre> Program Robot Program robotPaa:=read_input_boolean_register(5) Wait read_input_boolean_register(5) Gripper Activate Gripper Open (1) bunn:=read_input_boolean_register(0) topp:=read_input_boolean_register(1) full:=read_input_boolean_register(2) MoveL   startpos If bunn≠ True   MoveL     plukkeVerktoy     Gripper Close (1)     klar     Waypoint_1     Vask1_1     Loop 10 times       Vask1_1       Vask1_2     mellomlog2     Loop 10 times       Vask2_1       Vask2_2     Waypoint_2     plukkeVerktoy     Gripper Open (1)     startpos     Halt </pre>	<pre> If full≠ True MoveL   plukkeVerktoy   Gripper Close (1)   klar   Waypoint_1   Vask1_1   Loop 10 times     Vask1_1     Vask1_2   mellomlog2   Loop 10 times     Vask2_1     Vask2_2   mellom2og3   Loop 10 times     Vask3_1     Vask3_2   mellom3og4   Loop 10 times     Vask4_1     Vask4_2   ned   MellompunktVeiNed   plukkeVerktoy   Gripper Open (1)   startpos   Halt </pre>	<pre> If topp≠ True MoveL   plukkeVerktoy   Gripper Close (1)   klar   mellom2og3   Loop 10 times     Vask3_1     Vask3_2   mellom3og4   Loop 10 times     Vask4_1     Vask4_2   ned   MellompunktVeiNed   plukkeVerktoy   Gripper Open (1)   startpos   Halt </pre>
--	--	--

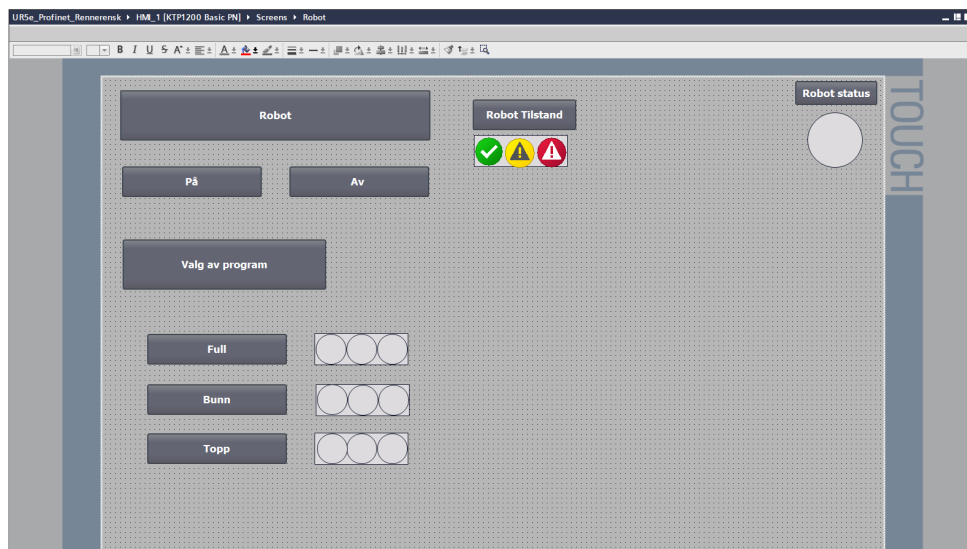
Figur C.1: PolyScope UR5e kode

## D Funksjonell designspesifikasjon

### D.1 Relaterte dokumenter

- Vedlegg A
- Vedlegg B
- Vedlegg C

### D.2 HMI Design



Figur D.1: HMI-panel

Følgende retningslinjer skal bli fulgt for HMI-skjermens farger:

- Grey (RGB 192,192,192) skal bli brukt som standard bakgrunn.
- Rød farge skal bare brukes til å indikere unormale hendelser som feil.
- Grønn farge brukes ved å indikere at robot er slått på og eventuelle program kjører.
- Gul farge brukes ved å indikere at robot program pågår.

### D.3 UR5e modusfunksjoner

UR5e robot styres med følgende programmer:

- Robot
- På
- Av
- Valg programmer

- Full
- Bunn
- Topp

Programmet som er aktivert vil få indikasjon at programmet er aktivt.

## D.4 Robot

“Robot” åpner valgmenyen om å aktivere UR5e robot eller deaktivere.

## D.5 På

På funksjon vil aktivere UR5e-robot gjennom PLS.

## D.6 Av

Av funksjon vil deaktivere UR5e-robot gjennom PLS.

## D.7 Valg av programmer

“Valg av programmer” åpner menyen til valgprogrammene Full, Bunn og Topp.

## D.8 Full

«Full» gir signal til PLS om at UR5e-robot skal starte rennerensk. Funksjonen «Full» vil starte med å plukke opp verktøy, gå til takrennens posisjon for å starte rennerensk på rennen fra topp til bunn, ved programslutt vil UR5e-robot parkere.

## D.9 Bunn

«Bunn» gir signal til PLS om at UR5e-robot skal starte rennerensk. Funksjonen “Bunn” vil starte med å plukke opp verktøy, gå til takrennens posisjon for å starte rennerensk på bunnen av rennen, ved programslutt vil UR5e-robot parkere.

## D.10 Topp

“Topp” gir signal til PLS om at UR5e-robot skal starte rennerensk. Funksjonen “Topp” vil starte med å plukke opp verktøy på verktøyets plassering, gå til takrennens posisjon for å starte rennerensk på toppen av rennen, ved programslutt vil UR5e-robot parkere.