

BACHELOROPPGAVE

Robotarm baneplanlegging med blandet virkelighet

Robot arm trajectory planning with mixed reality

Sivert Hagelin Benjaminsen og Håvar Dankel

Fakultet for ingeniør- og naturvitenskap/Institutt for datateknologi,
elektroteknologi og realfag

Informasjonsteknologi & Automatisering med robotikk

Veileder: Martin Fostad Stølen

Innleveringsdato: 30. mai 2022

TITTELSIDE FOR HOVEDPROSJEKT

<i>Rapportens tittel:</i> Robotarm baneplanlegging med blandet virkelighet	<i>Dato:</i> 24.02.2022
<i>Forfatter(e):</i> Sivert Hagelin Benjaminsen, Håvar Dankel	<i>Antall sider u/vedlegg:</i> 38
	<i>Antall sider vedlegg:</i> 4
<i>Studieretning:</i> Informasjonsteknologi, Automatisering med Robotikk	<i>Antall disketter/CD-er:</i> Ingen
<i>Kontaktperson ved studieretning:</i> Martin Fodstad Stølen	<i>Gradering:</i> Ingen
<i>Merknader:</i> Ingen	

<i>Oppdragsgiver:</i> Trolltunga Robotics AS	<i>Oppdragsgivers referanse:</i>
<i>Oppdragsgivers kontaktperson:</i> Atle Rettedal	<i>Telefon:</i>

Sammendrag: Bachelorprosjektet går ut på å lage en programvare som lar brukeren visualisere og styre en robotarm kun ved bruk av blandet virkelighet. Formålet med oppgaven er å undersøke potensialet utvidet virkelighet har innen industriell robotikk.

Stikkord:

Blandet virkelighet	Hololens 2	Robotarm
---------------------	------------	----------

FORORD

Rapporten dokumenterer utførelsen av prosjektet. Målet med oppgaven var å lage et system som demonstrerer fordelene av blandet virkelighet som et verktøy i industriell robotikk.

Dette prosjektet er i samarbeid med Trolltunga Robotics AS. Vi har vært i kontakt med Atle Rettedal, leder av Trolltunga Robotics. Oppgaven fikk vi av han, og de har fulgt opplegget nøye gjennom hele prosessen. En stor takk til dem.

Vi vil også takke Martin Fostad Stølen for all veiledning og tilbakemelding, og stipendiatene på Campus Førde for god hjelp og tålmodighet.



INNHOLDSFORTEGNELSE

FORORD.....	3
1 INNLEDNING	1
1.1 MOTIVASJON OG PROBLEMBESKRIVELSE	1
1.2 KONTEKST.....	1
1.2.1 Oppdragsgiver.....	1
1.2.2 Utvidet Virkelighet og virtuell virkelighet.....	2
1.3 MÅL OG PROBLEMSTILLING	2
1.4 RESSURSER	3
1.5 OPPBYGGING AV RAPPORTEN	3
2 PROSJEKTBESKRIVELSE.....	4
2.1 PROSJEKTEIER	4
2.2 TIDLIGERE ARBEID	4
2.2.1 Tidligere arbeid Trolltunga Robotics AS	4
2.2.2 Tidligere arbeid i forskningsartikler	4
2.3 INITIELLE KRAV	5
2.4 AVGRENSNINGER.....	5
2.5 INITIELL LØSNING	5
3 DESIGN AV PROSJEKTET.....	6
3.1: ALTERNATIVE LØSNINGER	6
3.1.1 Lakkering av et objekt.....	6
3.1.2 Banemanipulering.....	6
3.1.3 Visualisering av krefter og arbeidsområde for robotarmen.....	6
3.1.4 Invers Kinematikk for baneoptak eller sanntidsstyring.....	6
3.2 VALGT LØSNING	7
3.3 VALG AV VERKTØY	7
3.3.1 Valg av Robot.....	7
3.2.1.1 Yumi.....	8
3.2.1.2 UR5e.....	8
3.3.2 Valg av AR headset	8
3.2.2.1 HTC Vive	8
3.2.2.2 HoloLens 2	9
3.3.3 Operativsystem.....	9
3.3.4 Valg av plattform.....	9
3.3.4.1 RobotStudio.....	9
3.3.4.2 ROS	10
3.3.4.3 Unity.....	10
3.3.5 Hjelpemidler (programmer og scripts)	10
3.3.5.1 Mixed Reality Toolkit (MRTK).....	10
3.3.5.2 NuGet	10



3.3.5.3	Robotics Hub	10
3.3.5.4	MoveIt	11
3.3.5.5	Microsoft.MixedReality.QR	11
3.4	PROSJEKTMETODIKK	11
3.4.1	Utviklingsmetodikk	11
3.4.2	Prosjektplan	12
3.4.3	Risikoanalyse	13
3.5	EVALUERINGSPLAN	14
4	DETALJERT LØSNING	14
4.1	SYSTEMETS ARKITEKTUR	14
4.2	VISUALISERING	15
4.3	POSISJONERING AV HOLOGRAM	16
4.4	KONTROLLPANEL	18
4.5	PLUKK OG PASSER	18
4.6	BANEPLANLEGGING	19
4.7	UTFØRELSE	20
4.8	UTFORMINGSMODELLER	21
4.8.1	Brukstilfeller	21
4.8.2	Sekvensdiagram	22
5	RESULTATER	23
5.1	EVALUERINGSMETODE	23
5.1.1	SCRUM	23
5.1.2	Evaluering av løsningen	23
5.2	EVALUERINGSRESULTAT	26
5.2.1	Effektivitet	26
6	DISKUSJON	28
6.1	FREMGANGSMÅTE OG VALG	28
6.2	REFLEKSJON RUNDT VALGENE	29
7	KONKLUSJON OG VIDERE ARBEID	30
7.1	KONKLUSJON	30
7.2	VIDERE ARBEID	30
8	REFERANSER	32
9	VEDLEGG	33
9.1	GANTT	33
9.2	WIREFRAME	33
9.3	SAMMENHENG MELLOM UNITY OG ROS MED FOKUS PÅ UNITY	33
9.4	GITHUB LINK TIL KILDEKODE	33
9.5	YOUTUBEVIDEO AV BRUKSTILFELLET	33
9.6	SPØRRESKJEMA FOR GRUPPETESTING	35

Ordliste:

Augmentet Reality (AR)/ Utvidet Virkelighet	Teknologi som kombinerer data fra virkeligheten med virtuell data.
Mixed Reality (MR)/ Blandet Virkelighet	Teknologi som kombinerer data fra virkeligheten med virtuell data hvor fysiske og digitale objekter sameksisterer og samhandler i sanntid.
CoBot	En sammenslåing av Collaboration og Robot. Menes en robotarm som er laget for tett samarbeid med mennesker.
HoloLens	Et produkt fra Microsoft. Dette er en AR-enhet/briller som legger digital informasjon oppå det du allerede ser i den virkelige verden
Unity	Spillplattform som blant annet muliggjør utvikling av applikasjoner til HoloLens 2
ROS	Open-Source rammeverk til utvikling av robot applikasjoner
MRTK	Microsofts tilleggspakke til Unity programmering av AR og VR
Hologram	Motiver blir vist tredimensjonalt
End-effektor /slutteffektor	Enden på en robotarm som samhandler med miljøet
Aktuator	Et objekt som forvandler energi til bevegelse
HMS	Forkortelse for Helse, Miljø og Sikkerhet.
ABB	Produkt og teknologiselskap med programvarer og roboter.
RobotStudio	ABB sin programvare for programmering av roboter
Movelt	ROS-pakke for robotbaneplanlegging.
Arbeidsområdet	Rekkevidden til tobotarmen
SCRUM	Rammeverk med mål om å utvikle, levere og opprettholde produkter i et komplekst miljø.
Degrees of freedom(DoF)	Frihetsgrader til en robotarm, beskriver hvor mange ledd robotarmen har
Invers kinematikk	Utrekningen av leddverdier når TCP er kjent
Tool center placement (TCP)	Bestemmer det eksakte arbeidspunktet for en robot
UR5e	Samarbeidende robot fra Universal Robots
Yumi	Samarbeidende robot fra ABB
NuGet	Pakkemanager
URDF	Unified Robot Description Format, XML fil som representerer robot
Backlog	Prioritert liste over funksjonaliteter løsningen skal tilby
Pose	Spesifikk posisjonering og orientering i rommet
Prefab	Forhåndskonfigurerte gjenbrukbare spillobjekter

1 INNLEDNING

Rapporten er skrevet for oppdragsgiver Trolltunga Robotics (heretter Trolltunga) i den hensikt å utforske mulighetene av blandet virkelighet (heretter MR) som en tjeneste innen industriell robotikk. Mulighetene for MR innen robotikk er store. På grunn av avgrensningene i dette prosjektet er det valgt å fokusere på noen få brukstilfeller som skal demonstrere anvendbarheten av teknologien. Prosessen bak arbeidet vil bli beskrevet i flere detaljer gjennom rapporten. Hensikten med kapittel 1 er å gi leseren en forståelse for målet og problembeskrivelsen.

1.1 Motivasjon og Problembeskrivelse

Med dagens utvikling innen industriell robotikk blir arbeidsoppgavene og arbeidsmiljøene industrielle roboter opererer i mer og mer komplekse. Det å lære seg å programmere, visualisere og forstå roboter er en tidkrevende og komplisert prosess. Den vanligste fremgangsmåten for bedrifter i dag er at en ansatt eller et team eller tredjepart via innleid firma manuelt programmerer og utvikler løsninger til industrielle roboter. For å manuelt programmere robotene som skal operere i disse miljøene krever det mye tid og ressurser. (H. Zhang, 2018) Hvis det fantes en lettere og mer intuitiv måte å programmere og visualisere robotløsninger kunne det vært mer tiltrekkende for bedrifter med hyppig endrende miljøer å ta i bruk industrielle roboter som en del av deres løsning.

1.2 Kontekst

1.2.1 Oppdragsgiver

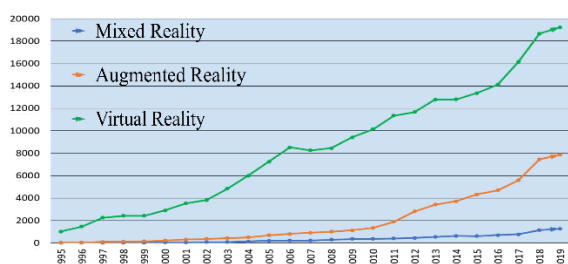
Trolltunga er et morselskap til en nordisk gruppe av selskaper innen robotikk. Deres visjon er å «utvikle nye innovative løsninger muliggjort av roboter og AI for å løse farlige eller repetitive oppgaver.» (Trolltunga Robotics AS, n.d.) RobNor er ett av datterselskapene til Trolltunga som spesialisere seg på automatisering av lakkeringsprosesser. Dette kan være både komplisert og tidkrevende. Ettersom det som skal lakkeres er av ulik form og størrelse krever det at RobNor sine ansatte stadig må programmere nye løsninger. For å håndtere dette tok de i bruk virtuell virkelighet (VR) som et verktøy for å ta ned tiden det tok å programmere robotene. Som resultat gikk tiden det tok til en ferdig løsning var implementert ned med 80 prosent. (RobNor, n.d.) Trolltunga er derfor interessert i om også MR kan være et virkemiddel innen industriell robotikk.



Figur 1 RoboPall - Industriell ABB robot for plassering av bokser. Hentet fra: <https://robotmorge.no/applikasjoner/>

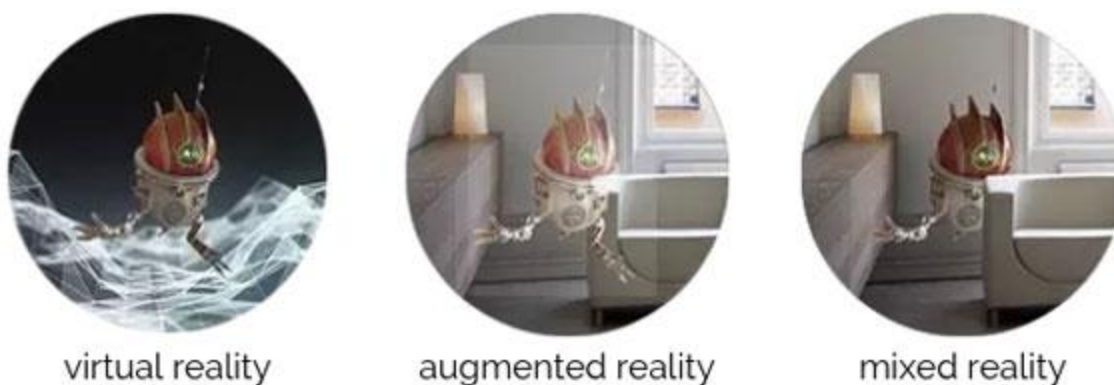
1.2.2 Utvidet Virkelighet og virtuell virkelighet

Utvidet virkelighet (heretter AR) er en teknologi som kombinerer data fra virkeligheten med virtuell data. Denne teknologien finnes på de fleste telefoner i dag og brukes for eksempel i sosiale media applikasjoner som Snapchat filter eller det populære mobilspillet Pokemon Go. Her blir virtuell data lagt over data fra virkeligheten. Det er ikke bare mobiler som kan prosjektere virtuell data over virkeligheten. Microsoft kom i 2016 ut med en AR-enhet kalt HoloLens. Denne tar i bruk en teknologi kalt blandet virkelighet (heretter MR). Forskjellen mellom AR og MR er at sistnevnte ikke bare som kombinerer data fra den virkelige verden med virtuell data, men lar også fysiske og digitale objekter sameksistere og samhandle i sanntid.



Figur 2 Graf som viser antall ganger MR,AR og VR er brukt som nøkkelord i forskningsartikler
Hentet fra: <https://www.mdpi.com/2218-6581/9/2/21/htm#>

Selv om både utvidet virkelighet, blandet virkelighet og virtuell virkelighet baserer seg på å vise virtuell data til brukeren, er det svært forskjellig brukeropplevelse mellom teknologiene. I figur 3 er det eksempler på hvordan data blir presentert for brukeren med de forskjellige teknologiene. I VR blir hele synsfeltet til brukeren dekket av virtuell informasjon, i motsetning til AR. Data brukeren får se er pre-programmert, som gir utvikler større kontroll over alt som skjer i miljøet brukeren operer i. Dette gir både fordeler og ulemper innen begge teknologiene. VR gir mulighetene til å skape nøyaktig det miljøene som kreves gitt utfordringene, mens AR og MR gir muligheten til å bruke teknologien i skiftende miljøer. Data visualisert i MR tar i motsetning til AR hensyn til omgivelsene brukeren opererer i.



Figur 3 Forskjellen på det brukeren ser med de forskjellige teknologiene. Hentet fra:

1.3 Mål og Problemstilling

Bruken av utvidet/blandet virkelighet er nytt (se figur 2), men de konseptuelle anvendelsene teknologien kan ha innen industriell robotikk er store Som nevnt i punkt 1.2 er det vanskelig og ressurskrevende å programmere dagens industrielle roboter. RobNor har bevist at det er mulig å ta denne komplekse og ressurskrevende prosessen og effektivisere den ved bruk av VR. På grunn av dette er Trolltunga interessert i om en liknende løsning kan utvikles med utvidet/blandet virkelighet. Disse teknologiene kan på

overflaten virke identiske ut, men de opererer på forskjellige måter. Problemstillingen til gruppen blir å undersøke om også MR kan anvendes som et verktøy innen industriell robotikk for å effektivisere en prosess.

For å svare på problemstillingen er målet for oppgaven å lage et system som demonstrerer fordelene av MR som et verktøy i industriell robotikk. Programmeringen av en industriell robotarm er et eksempel på noe prosjektgruppen (heretter gruppen) kan se på, hvor spesifikke bruksområder kan være:

- Programmering av enkle oppgaver som å flytte end-effektor fra punkt A til B.
- Fjernstyring av robotarm ved manipulering av end-effektor.
- Legge inn out-of-bounds områder end-effektor eller robotarm ikke har lov å bevege seg i.
- Legge over informasjon om kreftene/hastighetene/vinklene i leddene, både simulert og fysisk.
- Lagre og endre armbevegelser.
- Plukk opp/plasser funksjon, operatør legger inn bokser i rommet der armen skal fra og til.

1.4 Ressurser

For å gjennomføre prosjektet trengte gruppen tilgang til diverse ressurser og verktøy. For å kunne bruke AR teknologi var det behov for en enhet som kunne prosjektere dette. En robotarm var også nødvendig til den fysiske delen av oppgaven. Høgskolen på Vestlandet Førde hadde et kontorlokale ved sentrum hvor gruppen fikk tilgang til både roboter og AR enheter. Det ble også stilt et kontor til disposisjon i perioden januar til april før robotene ble flyttet til Campus Førde. Tilgangen til kontoret på Verftet gjorde at gruppen til enhver tid kunne bruke utstyret oppgaven krevde med mindre det ble brukt av andre.

Doktorgradsstipendiatene satt også på disse kontorene. Disse var en verdifull ressurs da de har erfaring med utstyret som gruppen måtte bruke og var åpne for spørsmål. Det var også veileder Martin Stølen og oppdragsgiver Atle Rettedal som har mye erfaring innen robotikk og prosjektledelse.

1.5 Oppbygging av rapporten

I de neste kapitlene vil det komme detaljer om prosjektbeskrivelse, design av prosjektet, valgt løsning, resultater, diskusjon og videre arbeid. Rapporten er delt inn i ni kapitler som har følgende oppsett:

Kapittel 1: Introduksjon til rapporten med mål, beskrivelse, og ressurser

Kapittel 2: Her er det beskrevet i mer detaljer om bakgrunnen til oppgaven, avgrensinger og den initielle løsningen gruppen startet med.

Kapittel 3: Beskrivelse av brukstilfellet som representerer den initielle løsningen og valgene rundt dette.

Kapittel 4: Detaljert løsning av brukstilfellene hvor arkitekturen av programmet og koden blir beskrevet.

Kapittel 5: Evaluering av den endelige løsningen.

Kapittel 6: Diskusjon rundt valgene som ble tatt under prosjektet.

Kapittel 7: Konklusjonen av arbeidet og hvordan det kan utvikles videre.

Kapittel 8: Kilder og referanser.

Kapittel 9: Vedleggs kapittel med blant annet Gantt-diagram.

2 PROSJEKBESKRIVELSE

2.1 Prosjekteier

Prosjektet er et samarbeid mellom HVL og Trolltunga Robotics AS.

2.2 Tidligere arbeid

2.2.1 Tidligere arbeid Trolltunga Robotics AS

Før gruppen kunne starte utviklingen måtte det evalueres om oppgaven var gjennomførbar, og om det var tilstrekkelig med informasjon tilgjengelig. Et av hoveddrivkreftene til prosjektet er oppdragsgivers tidligere arbeid med VR som et verktøy innen robotikk. Selv om RobNor fra tidligere er kjent med VR har de derimot liten erfaring innen AR teknologi.

2.2.2 Tidligere arbeid i forskningsartikler

De siste årene har det blitt publisert mange forskningsartikler som omhandler AR i sammenheng med robotikk (Makhataeva, 2020). I 2019 ble det publisert en artikkel som benytter seg av HoloLens, en SIA5D robotarm, og en kobling mellom ROS og Unity (Manring, 2019). Målet her var å kunne kontrollere den fysiske armen realtime og å kunne spille inn bevegelser ved å manipulere robotarmen. Noen av problemene var at forhåndsvisningen av robotarmbevegelse ikke var den samme bevegelsen som den fysiske robotarm hadde. Dette vil si at den banen som de ble vist simulert, ikke var den samme som den fysiske robotarmen valgte. På videre arbeid nevner de at en mer nøyaktig kalibreringsfase kunne være hensiktsmessig. Et annet ønske var å bevege hvert ledd individuelt. Denne kan det være hensiktsmessig å se på under utførelsen av dette prosjektet siden disse har mange av de samme prinsippene som det er planlagt å bruke her.

En annen forskningsartikkel ble publisert i 2018 (C. P. Quintero, 2018). Her styres en robotarm med syv frihetsgrader, herav DoF. Styringen skjer ved bruk av håndbevegelser gjennom Microsoft HoloLens første generasjon. Det brukes egen programvare til å koble sammen robot og HoloLens. Løsningen har med plukk opp og plasser funksjon med

mulighet for å modifisere banen robotarmen planlegger å ta. Dette er relevant for dette prosjektet når det er et ønske om å implementere liknende funksjonalitet.

2.3 Initielle krav

Med opparbeidet kunnskap var det nå grunnlag for å utforske noen av mulighetene MR kan tilby. Målet med oppgaven var å utvikle et system for å undersøke fordelene og ulempene ved bruk av MR innen industriell robotikk. Det ble konkludert med at dette skulle undersøkes ved styring av en robotarm med MR. Etter å ha diskutert funnene gjort i forskningsartiklene under tidligere arbeid i 2.2, ble det tenkt ut noen initielle krav oppgaven skulle inneha for å bevise dette.

- Systemet skal kunne brukes kun gjennom MR. Dette innebærer samhandling med hologrammene
- Systemet skal være brukervennlig og intuitivt.
- Det skal kunne bli hentet opp et hologram av aktuell robotarm som skal kunne modifiseres gjennom håndbevegelser.
- Hologrammet skal kunne synkroniseres med den ekte robotarmen og ha de samme leddverdiene.
- Hologrammet skal kunne vise banen til fysisk robotarm før aksjonen blir utført.
- Kontrollpanelet skal være brukervennlig og ha et intuitivt oppsett.

2.4 Avgrensninger

Prosjektet har en del avgrensninger som må bli tatt hensyn til. Disse kommer i form av tid, omfang tidligere erfaring og teknologien.

- Tiden er begrenset, og ferdig rapport skal være innlevert innen 30. Mai.
- Omfanget er begrenset grunnet konstant antall medlemmer i prosjektarbeidet kombinert med tidsbegrensningen.
- Tidligere erfaring innen de forskjellige emnene som inngår i prosjektet vil gjenspeiles i løsningen.
- Hvor langt teknologien er kommet i utvikling har en del å si for hvor godt sluttproduktet vil fungere.

2.5 Initiell løsning

Den initielle løsningen var å lage et system som muliggjorde alle punktene satt under initielle krav. Prosjektet skulle ha et eller flere spesifikke brukstilfeller som oppfylte disse kravene. Dette skulle bli utført ved å styre og simulere en robotarm ved bruk av en MR-enhet. En simulert robotarm i form av et hologram skulle legge seg over den ekte roboten med et 1:1 forhold. Etter dette skulle den simulerte robotarmen kunne manipuleres på forskjellige måter. Den skulle gjøre opptak av hologrambevegelse og spille dem tilbake for å se om den ekte robotarmen ville kolliderer med noe. I tillegg skulle det være mulig å endre på denne banen for å unngå kollisjoner. Det var også et ønske om å kunne legge inn hindringer som robotarmen skulle unngå i planleggingen og utførelsen. I neste

kapittel vil det gått dypere i detaljer om de ulike brukstilfellene og begrunnelsene for valgt løsning.

3 DESIGN AV PROSJEKTET

3.1: Alternative løsninger

I dette delkapittelet vil det bli beskrevet fire alternative løsninger som ble vurdert som svar på problemstillingen i oppgaven.

3.1.1 Lakkering av et objekt

Her ble det sett på muligheten til å gjenskape arbeidet RobNor hadde gjort med lakkering av forskjellige objekter, bare at det her ble benyttet MR. Utførelsen ville gå ut på å lage en applikasjon som gjorde det mulig å overføre håndbevegelsene til brukeren over til en robotarm utstyrt med en malepistol. Banene skulle utføres så presist som mulig. Brukeren ville ha styrt en virtuell slutteffektor over området som skulle bli spraymalt. Når prosessen var ferdig, ville det være mulighet for å se bevegelsen bli utført i virtuelle omgivelser. Om bevegelsen ble godkjent ville brukeren kunne kjøre den på den fysiske roboten.

3.1.2 Banemanipulering

Bevegelsen robotarmen velger fra start til slutt kalles robotens bane. Visualisering av robotarmens bane kan utføres ved at hologrammet av robotarmen beveger seg slik reell robotarm planlegger å bevege seg. Det kan tegnes opp en linje som følger posisjonen av tuppen til robotarmen (herav TCP). Denne linjen kunne det tenkes at en kunne manipulere. Med dette kunne man endret stien bort ifra et tenkt hinder, og at en ny bane som følger den nye linjen ville da blitt kalkulert.

3.1.3 Visualisering av krefter og arbeidsområde for robotarmen

Det kunne være aktuelt å lage en applikasjon som lar brukeren se informasjon om kreftene utført, og arbeidsområdet av robotarmen. Å visualisere kreftene på en robotarm kan være hensiktsmessig eksempelvis om roboten håndterer store objekter som kan være til skade for mennesker. Dette, sammen med visualisering av arbeidsområde til robotarmen er relevant med tanke på HMS.

3.1.4 Invers Kinematikk for baneopptak eller sanntidsstyring

Kinematikk er betegnelsen på hvordan leddene beveger seg i forhold til hverandre, og utvekslingen av energi mellom dem. Robotarm kinematikk handler da om hvordan koblingene mellom leddene i robotarmen henger sammen, i tillegg til utvekslingen av krefter mellom disse koblingene. Ofte i robotikken, er en ute etter å bevege tuppen av robotarmen til et punkt i rommet. For å gjøre dette må en vite hvilke leddverdier hvert

ledd skal ha. Invers kinematikk er navnet på metodikken som brukes for å finne ut hvilke leddverdier dette skal være.

Logger vi posisjonen til TCP på robotarmen, vil det kunne regnes ut invers kinematikk for hvert punkt i denne loggen. Dette kunne til slutt brukes til å få robotarmen til å enten følge et punkt i sanntid, eller lagre en bane fra A til B i rommet. Den lagrede banen skulle også være mulig å spille av slik at en kunne se om det var en kollisjon mellom robotarm og ekte omgivelser.

3.2 Valgt løsning

Etter diskusjon rundt avgrensingene av oppgaven ble “plukk & plassér” valgt. Dette ville kunne vise anvendbarheten til MR-teknologi ved visualisering av tenkt bevegelse for robotarm og samhandling med hologrammer. Dette er også en oppgave som det ville være mulig å gjennomføre, gitt avgrensningene. De andre alternative løsningene hadde et for stort omfang.

Trolltunga har mange oppdrag som innebærer bruk av robotarmer, som palletering, meat mover og etikettapplikasjoner (RobotNorge, u.d.). Fremgangsmåten er da å se på om dette kunne vært lettere og gjennomføre ved bruk av teknologien blandet virkelighet. Løsningen vil gå ut på at man bruker et blandet-virkelighet-brillepar for styring av robotarmen. Denne skal gjøre en plukk- og plasser bevegelse på to ulike steder i rommet. Dette simulerer eksempelvis stabling av paller eller sortering av produkter. Forskjellen fra vanlig VR eller tradisjonell koding er at brukeren vil ha bedre romforståelse. Det gir også muligheten til å bevege seg rundt i rommet for å se at arbeidet blir utført på riktig måte. Plukk og Plasser punktene ville bli bestemt av to objekter som brukeren kunne plassere fritt innenfor arbeidsområdet til robotarmen. Disse objektene blir flyttet på gjennom håndbevegelser.

3.3 Valg av verktøy

Opgavene som var ønsket å utføre gikk ut på styring av robotarm ved bruk av MR teknologi. Det var flere roboter som var aktuelle basert på tilgjengelighet og bruk. Det var også to MR-brillepar tilgjengelige som kunne prosjektere data brukeren trengte. Videre måtte det velges plattformer som var compatible med sammenkobling av robotarmen og visualiseringsverktøyet.

3.3.1 Valg av Robot

Valget sto mellom to roboter. Disse var UR5e fra robotikklabben på campus Førde (UNIVERSAL ROBOTS, u.d.), og Yumi fra Trolltunga (ABB, u.d.).

3.2.1.1 Yumi



Figur 4 ABB YuMi, en samarbeidende robot med to armer
Hentet fra: <https://new.abb.com/es/tecnologia>

Yumi fra ABB er i likhet med UR5e en samarbeidende robot. Denne brukes ofte til montering av små deler da den kun har en maks bærekapasitet på 0.5 kilogram. Roboten er som vist i figur 4 utstyrt med to robotarmer som gjør at den lettere etterligner et menneske på en produktlinje. At roboten har to robotarmer åpner for spennende muligheter, men øker også kompleksiteten ved programmering. Det er her flere ting å tenke på, som kollisjoner og valg av arm som utfører en gitt oppgave. (ABB, n.d.)

3.2.1.2 UR5e

Til oppgaven ble Universal Robots UR5e valgt. Dette er en samarbeidende fleksibel robot arm designet for å simulere repetitive manuelle oppgaver med bærekapasitet på 5 kilogram. (Universal Robots, n.d.) Den har også et generisk robotarmdesign som går mye igjen i andre industrielle robotarmer. At den har lignende oppbygning som mange andre industrielle robotarmer var et viktig bidrag til beslutningen. UR5e ble også valgt grunnet gruppe medlemmenes tidligere erfaring med roboten, og at den var kompatibel med plattformen gruppen valgte for oppgaven.



Figur 5 UR5e, en samarbeidende robotarm.
Hentet fra: <https://www.universal-robots.com/no/nedlastingscenter/#/e-series>

3.3.2 Valg av AR headset

3.2.2.1 HTC Vive



Figur 6 HTC VIVE Pro
Hentet fra: <https://www.vive.com/eu/product/vive-pro/>

VIVE Pro kommer i et sett med to kontrollere som representerer hendene til brukeren og to basestasjoner. Basestasjonene plasseres på to forskjellige hjørner i rommet og sender ut signaler som brillene bruker til å posisjonere seg. Grunnen til at HTC Vive Pro ble vurdert er de to kontrollere. Disse kan føre til en bedre nøyaktighet enn uten. Kameraene plassert foran på brillene kan sende informasjon direkte inn til brukeren som integreres med VR omgivelsene. (Vive, n.d.) Dette betyr at i praksis fungerer disse da som et MR-brillepar.

3.2.2.2 HoloLens 2

HoloLens 2 er Microsoft sin versjon av utvidet virkelighet kalt «Mixed Reality» eller MR som kjører på Windows Holographic Operating System. Brillene tar i bruk flere sensorer, dybdekamera og holografisk prosessering som lar hologrammene gli sømløst inn i omgivelsene. (Microsoft, n.d.) Disse kan samhandles med, eller bli brukt til å vise informasjon til brukeren. En av fordelene med blandet virkelighet er muligheten til å kunne bruke det i endrende miljøer hvor brukeren alltid har ett overblikk over omgivelsene. Denne oversikten kan gjøre det tryggere for en operatør som samarbeider med roboter.

Brillene har en lang liste med funksjoner.

Blant disse finner man:

- Hode sporing
- Øye sporing
- Dybdesensor
- Kamera
- Mikrofon
- Hånd sporing
- Forståelse for omgivelsene
 - 6DoF tracking
 - Spatial Mapping
 - Mixed Reality Capture



Figur 7: <https://www.microsoft.com/nb-no/hololens/buy>

3.3.3 Operativsystem

For programmering av HoloLens2 kreves Windows 10 operativsystem og Mac OS eller Linux er ikke kompatible. Tilleggstjenestene mange av plattformene krever for å kunne programmere AR i Microsoft HoloLens2 kjøres ikke på de to sistnevnte. Windows 11 er kompatibel, men er per skrivende stund for ustabil.

3.3.4 Valg av plattform

3.3.4.1 RobotStudio

RobotStudio markedsføres som den største offline programmerings- og simuleringsplattformen. (ABB, n.d.) Denne er laget for å primært kjøre og programmere ABB-roboter. Tidlig ble det konkludert med at UR5e var den robotarmen som skulle brukes i prosjektet. Det ble funnet veldig lite informasjon på Universal Robots sammen med RobotStudio. Grunnen til at RobotStudio var et viktig alternativ å vurdere, var at de har allerede en innebygd integrering av VR og AR. I tillegg er det denne plattformen Trolltunga benyttet under VR utviklingen.

3.3.4.2 ROS

Robot Operating System(ROS) er et open-source robotikk rammeverk som baserer seg på noder og informasjonsflyten mellom disse. (ROS, n.d.) Siden ROS er open-source er det enkelt å studere informasjonsflyten i programmet. Å kjøre et system på flere datamaskiner er heller ikke noen problem da ROS har tilrettelagt for dette. ROS har i tillegg vært med i pensumet til begge gruppemedlemmene.

3.3.4.3 Unity

Unity er et integrert utviklingsmiljø designet for 2D og 3D utvikling ved bruk av .Net og programmeringsspråket C#. (Unity, n.d.)Unity lar brukeren bygge «scener» som inneholder alle objektene i systemet. Plattformen har flere tilleggstjenester og funksjoner som Microsoft Mixed Reality Toolkit (MRTK) og NuGet som er nødvendig for løsningen og beskrives i punkt 3.3.5. OpenXR er også brukt da det er den eneste backend programvaren som er anbefalt av Unity for HoloLens 2 programmering.

3.3.5 Hjelpemidler (programmer og scripts)

For at kommunikasjonen og programmets gang skal gå sømløst, må det tas i bruk en del hjelpemidler. Disse er i form av bibliotek, scripts og programmer.

3.3.5.1 Mixed Reality Toolkit (MRTK)

MRTK, er et prosjekt drevet av Microsoft som tilbyr en rekke komponenter og egenskaper brukt til å akselerere Mixed Reality applikasjonsutvikling i Unity. Komponentene fra MRTK som er valgt å bruke i oppgaven er

- MRTK Foundation
- MRTK Tools
- OpenXR

Av disse er MRTK Foundation den eneste pakken som må være inkludert, da den muliggjør Mixed Reality plattform funksjonalitet på applikasjonen.

3.3.5.2 NuGet

NuGet er en programvare for pakkebehandling som gjør det mulig for utviklere å dele gjenbrukbar kode. Det er utviklet en versjon av NuGet kalt NuGetForUnity som brukes i prosjektet. Dette ble brukt til å laste ned en SDK kalt Microsoft.MixedReality.QR fra NuGet brukt til å aktivere QR lokalisering på HoloLens 2.



Figur 8: <https://github.com/microsoft/MixedRealityToolkit-Unity>

3.3.5.3 Robotics Hub

Figur 7

Robotics Hub er designet kun for Unity. Dette gjør at de kan benytte seg av en direkte TCP-basert kommunikasjon. Via et TCP-endpoint på ROS siden, og en TCP-connector på

Unity siden, blir kommunikasjonen mye raskere mellom plattformene. Robotics Hub sin URDF-importer inkluderer i tillegg denne komponenten.

3.3.5.4 MoveIt

MoveIt! er en 3D-visualiseringsprogrammaren implementert i ROS som brukes i oppgaven til å visualisere og planlegge banen robotarmen skal ta. Planlegging er en viktig del av prosjektet, så en form for planleggingsalgoritme er nødvendig. Å skrive kode som planlegger banen mellom to punkt til en UR5e, som har seks ledd, er komplekst og tidkrevende. Om start- og sluttkoordinat for UR5e-en gis til MoveIt!, blir en bane automatisk kalkulert.

3.3.5.5 Microsoft.MixedReality.QR

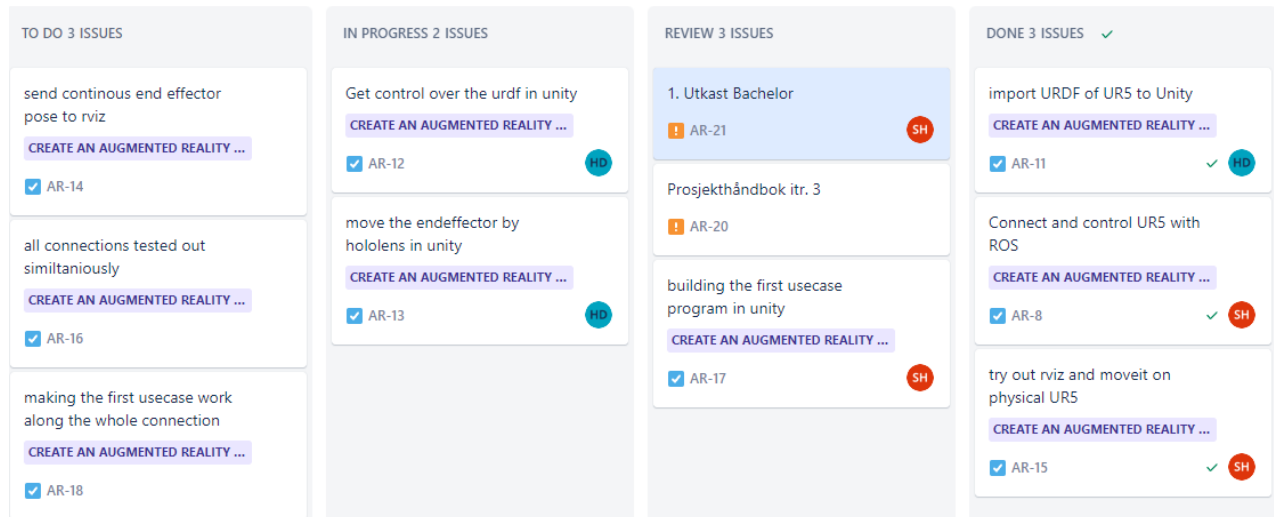
HoloLens 2 kommer standard med en ferdig QR skanner, denne skanner vanligvis automatisk etter QR koder men er ikke automatisk aktivert ved utvikling av nye HoloLens applikasjoner. For utvikling av applikasjoner for HoloLens 2 som krever tilgang til QR skanneren må Microsoft.MixedReality.QR være lagt til i programmet. Dette er et programvareutviklingssett (SDK) utviklet av Microsoft for deteksjon av QR koder på Windows Mixed Reality enheter. SDK'en som lastes ned gjennom NugetForUnity gir utvikleren mulighet til å lese av lokasjon, størrelse og innhold av detekterte QR koder som oppdateres i ekte tid. Denne muligheten til å lese av lokasjon og innhold gjør QR koder til en base å plassere roboten etter.

3.4 Prosjektmetodikk

3.4.1 Utviklingsmetodikk

For å arbeide systematisk og effektivt har gruppen valgt å anvende SCRUM, et rammeverk rettet mot systemutvikling. Med SCRUM fokuserer gruppen på å dele oppgaven inn i flere delmål, eller oppgaver, som må, eller burde, bli utført før produktet er klart. Delmålene blir satt inn i en backlog, som er en liste over fremtidige gjøremål. Når backlogen er utarbeidet velges ett visst antall oppgaver som skal bli gjennomført, ofte i en periode som vanligvis varer i 2-4 uker kalt en Sprint. Dette gjør det lettere å stykke opp arbeidet og jobbe med små, konkrete oppgaver.

Med en oppgave som denne blir det mye endringer i fremgangsmåten etter hvert som ny kunnskap opparbeides. SCRUM passer utmerket til dette da en bare kan legge inn nye oppgaver i backlogen og gi den en prioritet for når den må utføres. Den samme grunnen gjør at det også er lettere å få ny input av arbeidsgiver på arbeidet som er utført og hva som skal gjøres videre. En av filosofiene i SCRUM er daglige møter, dette gjør det lettere for alle i gruppen å ha oversikt over arbeidet.



Figur 8 Eksempel på en sprint

3.4.2 Prosjektplan

For at prosjektet skal bli ferdig er det en del oppgaver som må ferdigstilles og for å ha oversikt over hva som er gjennomført og mangler er arbeidet lagt inn i et Gantt-diagram. De ulike oppgavene er delt inn i sprinter og tildelt antatt start- og sluttidspunkt. Fordelen med dette er at gruppen har oversikt over hvor i prosjektet en er og hvor en eventuelt burde ligge. Gantt-diagrammet ble fulgt og tidene ble oppdatert gjennom prosjektet og etter hvert som gruppen kom lenger i arbeidet ble nye oppgaver lagt til i diagrammet.

3.4.3 Risikoanalyse

Ved oppstart av nye prosjekter er det viktig å ta høyde for eventuelle risikoer. Dette gjøres tidlig i prosjektet slik at gruppen er kjent med risikoene som kan oppstå og ha en plan på hvordan de skal håndtere problemet.

S: Sannsynlighet

K: Konsekvens

RP: Risikoprodukt (S * K)

1-5: 1 er lavest og 5 er høyest

Tabell 1 Risikoanalyse

	Hendelse /Risiko	Årsak	S	K	RP	Tiltak
1	Utstyr blir ødelagt	Uforsiktig under testing	Høy (3)	Høy (4)	12	Bruke utstyr på en forsvarlig måte
2	Produktet er teknisk ustabil	Ny teknologi med lite forkunnskap om testing	Høy (4)	Høy (5)	20	Legge av god tid til testing av produktet
3	Vanskelig å videreutvikle produktet	Dårlig dokumentert arbeid og kode	Høy (3)	Høy (3)	9	Skrive «Clean Code»
4	Produktet tilfredsstillende ikke et behov	Manglende forståelse og kunnskap fra utvikler teamet	Lav (2)	Høy (4)	8	Tilegne kunnskap fra Trolltunga ansatte
5	Prosjektet blir ikke ferdigstilt	Oppgavens mål er uoppnåelig innenfor gitt tidsramme	Middels (3)	Middels (3)	9	Ta i bruk et gjennomtenkt GANT-diagram med ukentlig oppfølging
6	Manglende kompetanse	Ny teknologi med lite forkunnskap om testing	Middels (3)	Middels (3)	9	Bruke kilder kritisk.

3.5 Evalueringsplan

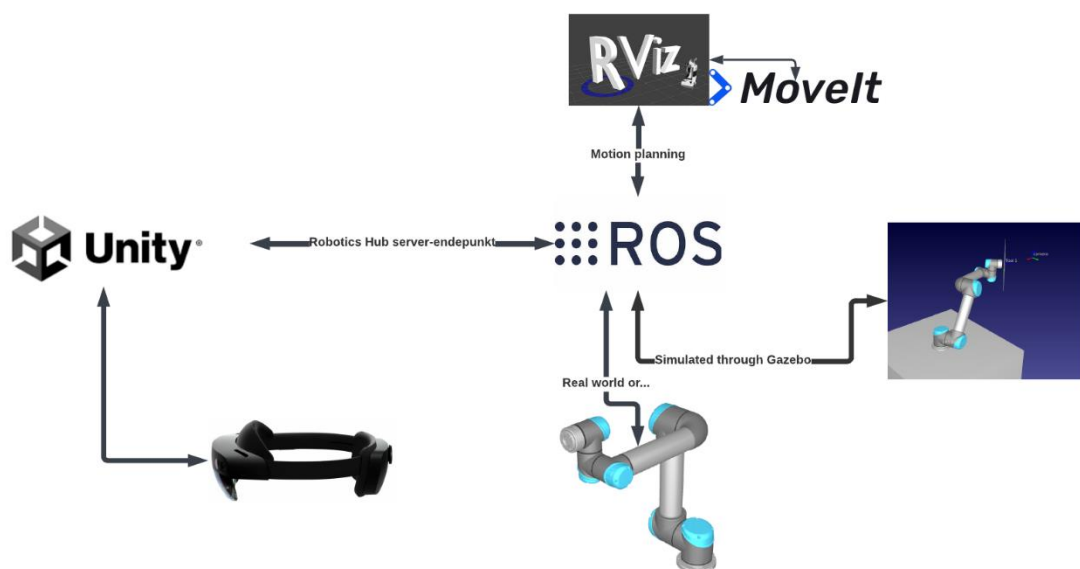
Planen for testing under utviklingen foregår i hovedsak av medlemmene i gruppen. Arbeidet blir utført i sprints, hvor hver sprint avsluttes ved testing av implementasjonen. Det arbeides til tider på ulike implementasjoner parallelt som gjør at disse også må testes opp mot hverandre for å se at løsningene fungerer korrekt sammen. Når gruppen nærmer seg ferdig, blir løsningen på prosjektet evaluert av medlemmene i gruppen og av tredjeparter. Tredjepartene vil bestå av veileder og arbeidsgiver, doktorgradsstudentene ved HVL Førde, og personer uten tidligere kunnskap om robotikk og HoloLens2. Et spørreskjema er formatert og gitt ut til hver person hvor det skal måle effektiviteten, brukervennligheten og kvaliteten til den endelige løsningen.

4 DETALJERT LØSNING

Kapittel 4 vil omhandle den konkrete løsningen som er utviklet. Problembeskrivelsen som denne skal svare på er om MR-teknologien er klar for implementering innen industriell robotikk.

4.1 Systemets Arkitektur

For å kunne styre, visualisere og regne ut robotarmbevegelser er løsningen basert på en tilkobling mellom to hovedplattformer. Disse er Unity og ROS, hvor Unity styrer logikken knyttet til HL2, og ROS styrer robotarmlogikken. Plattformene er knyttet sammen ved en serverkobling som vist i figur 11. På ROS-siden er det to hovedgreiner. Den ene er for planlegging av robotarmbevegelse, og den andre er for styring av ekte eller simulert robotarm.



Figur 9 Viser sammenkoblingen mellom programvare og maskinvare

4.2 Visualisering

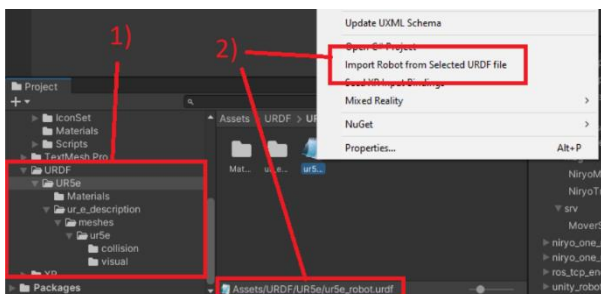


Figur 10 Visualisering av hologram

For visualisering av egenimplementert kode trengs Mixed Reality Toolkit (herav MRTK) sin Holographic remoting. Dette er et viktig verktøy som gjør at en kan kjøre prosjektet via en datamaskin. Holographic remoting aktiveres når Unity går i "Play mode". Når Unity spiller av prosjektet skjer all prosessering på datamaskinen som kjører unity. Det etableres her en kobling mellom datamaskinen og HoloLens 2. Alle pådrag fra HoloLens 2 blir her sendt til Unity for prosessering, og alt av hologrammer og funksjoner blir sendt tilbake. Dette er et viktig verktøy da HoloLens 2 har begrenset med minne

og prosessorkraft. Dette gjør at det under testing gikk en del tid på å laste opp prosjektet. Sletting av gamle prosjekter som hadde samlet seg opp hadde også tatt tid. Den andre grunnen er behovet for kommunikasjon med ROS. Her fungerer Unity som bindeleddet mellom HoloLens 2 og ROS. ROS er koblet til Unity, og Unity er koblet til HoloLens 2.

Når en skal visualisere robotarmen i Unity er det viktig at denne oppfører seg som forventet. URDF-importer befinner seg i Robotics Hub. URDF står for Unified Robot Description Format (Wiki Ros, u.d.). Dette er et format som gjør det enkelt å beskrive

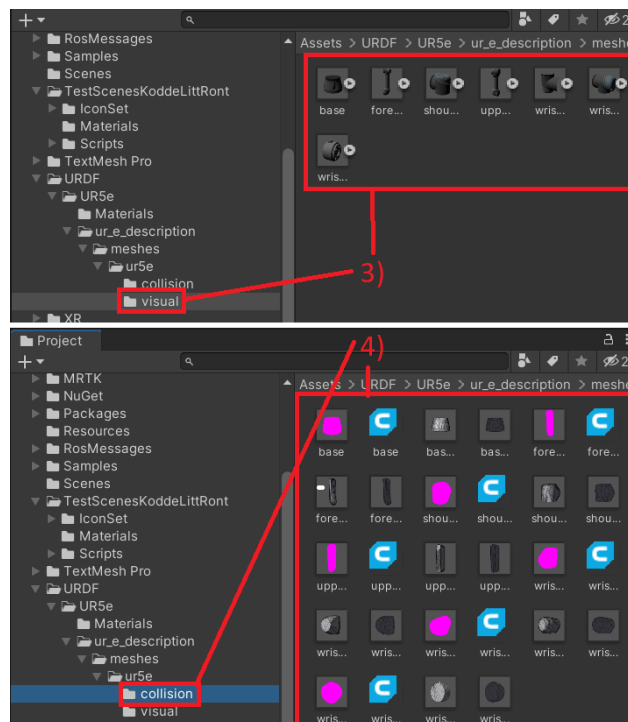


Figur 15:11 Hierarkiet i Unity, viser filstruktur for opplasting av URDF

utseende og funksjonalitet av alle deler i en robot. En URDF er da en filtype som brukes i hovedsakelig i ROS til å beskrive roboter. For å få en slik beskrivelse av en robot inn i Unity har Robotics hub laget URDF-importeren.

Bruken av denne baserer seg på å ta inn hele filsystemet som beskriver roboten, som vi ser i 1). Dette innebærer URDF-filen som beskriver alle delene som sammen blir hele robotarmen, og leddene mellom disse delene og beskrivelsen av hvilken retning disse virker, som vist i 2). I tillegg skal det være med mesh- og kollisjonsbeskrivelser av alle deler som beskriver hvordan delene skal se ut og virke sammen, som vist i 3) og 4). Når filsystemet er som forventes, vil vi kunne generere modellen av robotarmen i Unity-prosjektet.

Når modellen er på plass i Unity-prosjektet, blir det lagt til et standard kontroll-script på robotarmen. Dette har en hovedoppgave, å kontrollere roboten.



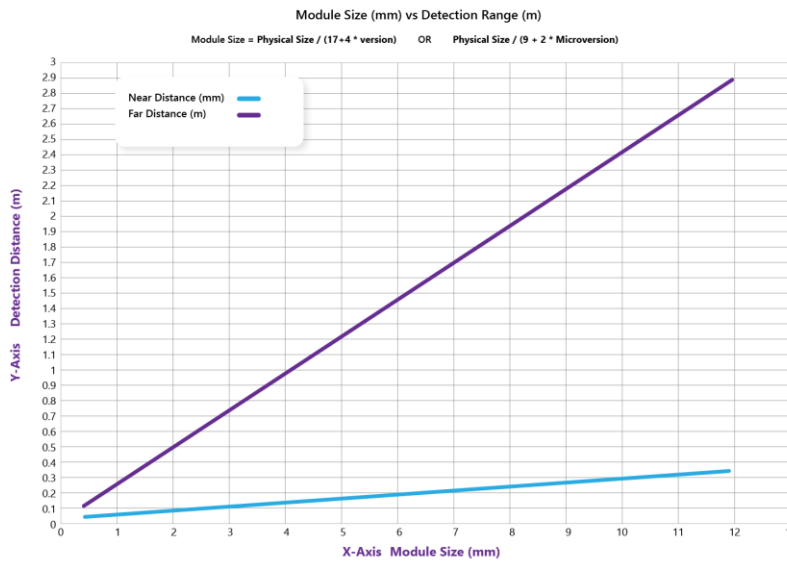
Figur 16:12 Viser innhold i URDF

4.3 Posisjonering av hologram

For at hologrammet skal være en representativ visualisering av robotarmen må den være plassert over den ekte roboten. Dette krever at basen til hologrammet blir posisjonert over basen til den fysiske roboten. Unity har en innebygget transform funksjon hvor utvikleren manuelt kan plassere objekter i rommet ved å endre på verdiene i Position og Rotation. Denne funksjonen er god i et statisk miljø med lite bevegelse og endringer. I denne oppgaven kan plasseringen til hologrammet endre seg med omgivelsene den fysiske roboten befinner seg i. For å håndtere dette er det implementert en QR skanner hvor hologrammet plasseres basert på en QR kode som plasseres ved basen til den fysiske roboten.

Før QR skanneren kan bli tatt i bruk i HoloLens applikasjoner må noen krav være oppfylt. SDK'en, Microsoft.MixedReality.QR og Microsoft.VCRTForwarders.140 være lagt til, webcam capabilities aktivert i Unity's Player Settings og Universal Windows Platform valgt som plattform. For optimal QR deteksjon er det anbefalt å ha en margin rundt sidene til QR koden som er uten innhold. Størrelsen på QR koden må ikke være mindre enn fem cm

Figur 17:13 Graf som illustrerer deteksjonsavstand(y-akse) i forhold til QR størrelse(x-akse)
Hentet fra: <https://docs.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/qr-code-tracking-overview#how-do-i-make-the-qr-code-tracking-feature-work-on-hololens-2-devices>



og for QR koder med sider mellom fem og ti cm kreves det at enheten er nærme koden. QR koder kommer i flere versjoner fra 1-40, versjon 1-10 er anbefalt til bruk for lettere deteksjon. I prosjektet er QR versjon 1 brukt med kantlengde på 20 cm

I løsningen er det brukt en modifisert kode med utgangspunkt i Microsoft. En UI-panel er også implementert med en start/stop knapp som aktiverer QR skanneren på HoloLens enheten.

Scriptene som er brukt for QR søkingen er QRCodeManager.cs, SpatialGraphNodeTracker.cs, QRCodeVisualizer.cs, QRCode.cs og SpatialGraphCoordinateSystem.cs. En prefab kalt QRCode inneholder en akse og en gjennomskiktig Quad (vist på figur 15) som blir fremstilt over QR koden for å vise koordinatene og området som blir skannet. For at disse funksjonene skal fungere er det to scripts i prefaben QRCode.cs og SpatialGraphCoordinateSystem.cs. Programvareutviklingssettet Microsoft.MixedReality.QR lar scriptene få tilgang til lokasjonen og innholdet av de detekterte QR kodene.

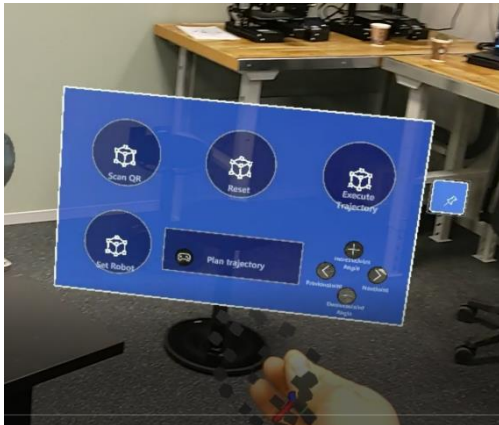


Figur 14 Viser digital informasjon (prefaben) lagt over QR-koden

QRCodeSetup initierer QR søkefunksjonen i QRCodeManager.cs og QRCodeVisualizer.cs. QRCodeManager.cs fungerer som hovedklassen som håndterer QR programvareutviklingssettet. Når programmet starter, sender den en forespørsel til QRCodeWatcher som er en del av programvareutviklingssettet om å få tilgang til webkameraet på HoloLens2 enheten. Dette sjekker at QR søking er støttet i systemet. Den håndterer også en del andre funksjoner, men de som er anvendelige nå er metodene som starter og stopper QR søkingen. QRCodeVisualizer.cs tar seg av QR kodene som blir visualisert i

Unity scenen når bruker ser på en fysisk QR kode og lagrer disse lokalt i QRCodeManager.cs. Den gjør det ved å instansiere et nytt qrCodeObject og legger på id fra SpatialGraphNodeTracker.cs og informasjon om QR koden fra QRCode.cs. Dette skjer automatisk når brukeren trykker på «Start Scan» og en QR kode er detektert. QRCode.cs er scriptet festet til QR kode objektet og gir QR informasjonen i tekst vinduet over QR koden som vist på bildet. Scriptet SpatialGraphNodeTracker.cs tar seg av posisjoneringen av den fysiske QR koden i rommet. Den har også en metode setRobot() som aktiverer og deaktiverer URDF-gameobjektet og tar Pose fra fysiske QR koden og setter den inn i en transform metode som setter posisjonen relativt av QR koden. .

4.4 Kontrollpanel



Figur 15 Kontrollpanel

For å oppfylle kravene om et brukergrensesnitt kun gjennom HL2, var det et behov om å kunne kontrollere robotarmen via knappene i scenen med håndbevegelser. Det ble da tatt i bruk MRTK, som har et lite bibliotek av knapper og tavler som kan implementeres, som vist i 1). Knappene kobles videre til en metode implementert i kontroller-scriptet. Det er fire metoder, en til hver av de fire knappene. Høyre/venstre er valg av ledd som skal justeres, og pluss/minus justerer ønsket leddverdi, target, som det da videre reguleres inn mot. I 2) er det koblet til en metode som heter PublishTrajectory().

4.5 Plukk og plasser

Før plukk og plasser funksjonen kan bli iverksatt, må spillobjektene som indikerer plukk og plasser bli plassert. For at dette skal kunne gjøres er det lagt til et «NearInteractionGrabable»-script på begge objektene. Dette gjør at brukeren med HoloLens 2 enheten kan plukke og plassere hvert objekt til ønsket posisjon.

Når en skal planlegge en bane trykker man på Plan Trajectory-knappen. Denne sender pose-koordinatene av spillobjektene og nåværende leddverdier av roboten til ROS for utregning av robot banen gjennom en server. Banen roboten har planlagt å ta blir da visualisert for brukeren. Bruker kan her velge å iverksette ekte robot for gjennomføring av samme bevegelse.

Kommunikasjonen til ROS skjer gjennom Robotics Hub sin Ros-connector. Denne henter og sender meldinger fra serveren som ROS kommuniserer med ved å koble seg til denne via IP-adressen til det operativsystemet ROS blir kjørt på.

ROS og Unity har forskjellige koordinatsystem. ROS har høyrehånds, og Unity har venstrehånds-koordinatsystem. I ROS-connector velges det at "Z Axis Direction" settes nord i forhold til ROS. Dette løser da alle konverteringsproblemer mellom plattformene.

Fordelen med denne typen kommunikasjon er at en kan kjøre ROS og Unity på to forskjellige datamaskiner om en vil. Med innsyn i lignende prosjekter, som nevnt i kapittel 2.1.3, ble det konkludert med at dette var veldig hensiktsmessig. Dette gjør at prosessorkraften som trengs for å kjøre hele prosjektet kan bli fordelt utover to datamaskiner, som igjen betyr at helhetlig produkt oppfører seg raskere og oppleves bedre.

4.6 Baneplanlegging

ROS blir kjørt på en virtuell datamaskin. Kommunikasjonen fra Unity til ROS skjer gjennom et serverendepunkt. Dette opprettes når vi kjører launch-filen sammen med en del andre elementer.

```
roslaunch ur5_e_moveit_config PlukkOgPlasser.launch tcp_ip:=192.168.81.16 tcp_port:=30003
<launch>
  <arg name="tcp_ip" default="172.31.1.161"/>
  <arg name="tcp_port" default="30003"/>

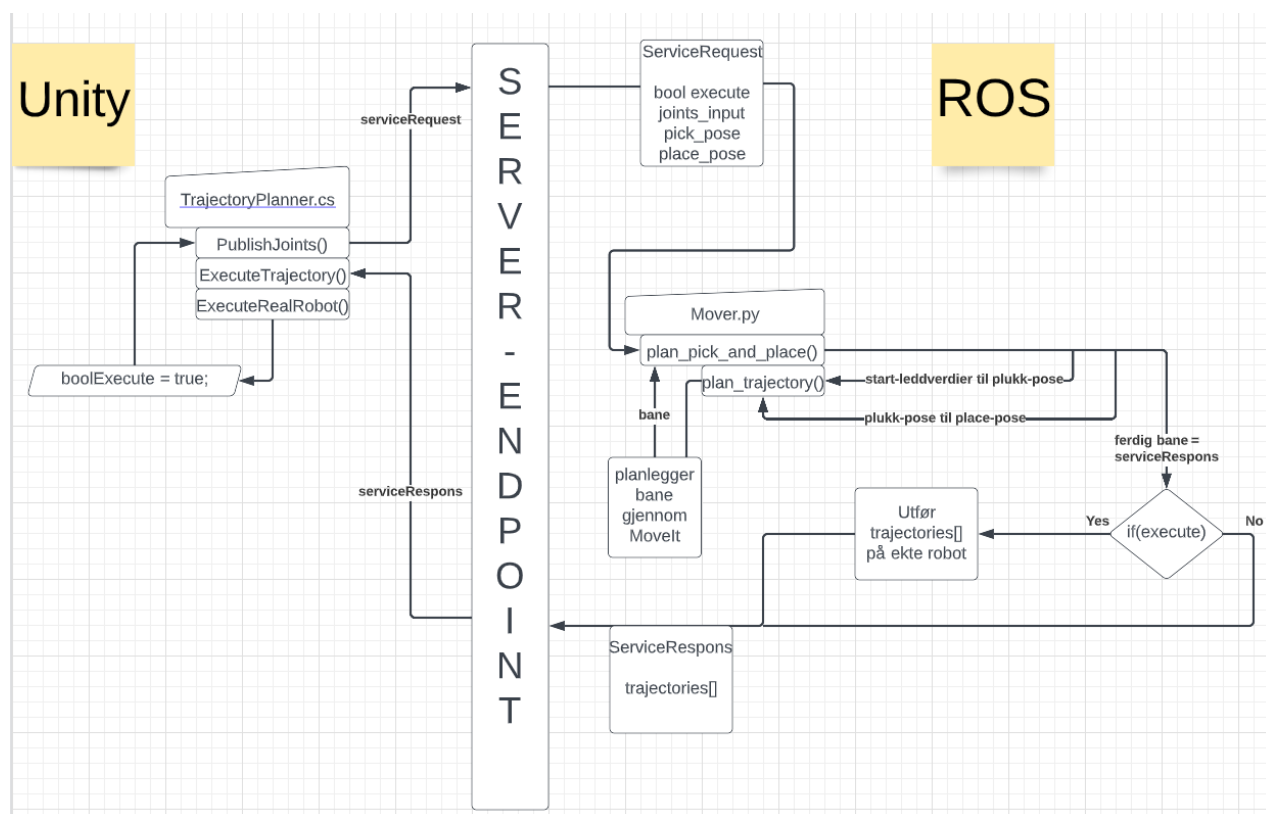
  <include file="$(find ur_e_gazebo)/launch/ur5e.launch" />
  <node name="server_endpoint" pkg="ros_tcp_endpoint" type="default_server_endpoint.py" args="--wait" output="screen"
    respawn="true">
    <param name="tcp_ip" type="string" value="$(arg tcp_ip)"/>
    <param name="tcp_port" type="int" value="$(arg tcp_port)"/>
  </node>
  <node name="mover" pkg="ur5_e_moveit_config" type="mover.py" args="--wait" output="screen"/>
  <include file="$(find ur5_e_moveit_config)/launch/ur5_e_selfmade_planning_execution.launch" />
</launch>
```

Figur 20: 16 Viser først kommandoen for launch-filen etterfulgt av hvordan launch-filen ser ut

Dette er den launch-filen som blir kjørt når programmet skal starte. Det opprettes først en node som fungerer som serverendepunkt. Videre kjøres mover.py scriptet som står for logikken i ROS-delen av systemet. Dette etterfølges av en initialisering av en ny launch-fil som står for oppsett av den ekte roboten og de komponentene denne trenger. Det tildeles her en kontroller til robotarmen etterfulgt av etablering av kommunikasjon mellom mover.py og denne kontrolleren.

Robot Operating System er den plattformen som brukes for planlegging av robotarm-bane og styring av fysisk robot. Det er tatt i bruk en del filer og filoppsett fra Universal Robots og Robotics Hub. Universal Robots har et bibliotek med kontrollere og robotbeskrivelser for alle robotene sine. Dette er samme robotbeskrivelse som er brukt for å få roboten inn i Unity. Robotics Hub har noen eksempeloppgaver som går på ROS kommunikasjon og styring av en robotarm gjennom MoveIt. En del av denne koden er også tatt i bruk i dette prosjektet, men modifisert en del for å få den til å virke på UR5e.

4.7 Utførelse



Figur 17: Beskrivelse av sammenkoblingen Unity - ROS

Når plukk, plasser og leddverdier kommer inn til ROS fra Unity, vil disse bli tatt opp i mover.py scriptet. Her vil disse bli tatt inn som argument i `plan_pick_and_place()` metoden. Hovedmålet til denne er å bygge opp et svar, kalt `response`, som skal sendes tilbake til Unity. Dette gjøres ved å sende koordinatene stegvis over til MoveIt for planlegging. Banene som til slutt konstrueres er først fra nåværende posisjon til plukkobjektet, deretter fra plukk- til plasser-objektet.

Når disse fire banene er konstruert, kommer vi til en `if`-setning. Denne velger om den fulle banen skal sendes til ekte robot med hensyn på den boolske variabelen `execute`. Uavhengig av denne blir banen sendt tilbake til Unity i etterkant. Her vil den bli vist i sin helhet av den simulerte robotarmen.

Det er i tillegg to ekstra baner som konstrueres. Metoden er en modifisert utgave av Robotics Hub sin `PickAndPlace` eksempelkode. I denne var det tenkt at det skulle være en griper på tuppen, og derfor er det her to ekstra baner som blir konstruert. Disse to er for å gripe det bevegelige objektet i plukkposisjonen, og for å slippe det i plasser-posisjonen. Det ble konkludert med at disse kunne stå grunnet at de viser tydelig hvordan det ville sett ut hvis roboten hadde hatt en griper montert.

For å løse problemet med hvordan en skal velge riktige robotarm-konfigurasjoner som gir det mest effektive banen, gir vi start-, plukk- og plasser-koordinatene til MoveIt. Den kalkulerer hvordan robotens bane blir slik at slutteffektoren blir stående i en gitt posisjon

etter endt bevegelse. Mover.py initialiserer moveit_commander, og med denne blir det gitt tilgang til å sende og motta informasjon fra MoveIt.

4.8 Utformingsmodeller

4.8.1 Brukstilfeller

Skanne QR kode

En flytende meny inneholder en knapp «start QR» brukeren trykker på denne som vil aktivere QR skanneren på enheten.

Posisjonere robot

Når en QR kode er funnet kan brukeren trykke på en knapp «SetRobot» som vil plassere roboten relativt til QR koden og stoppe skanner funksjonen til HoloLens2.

Stopper QR kode skanner

Brukeren trykker på «StopScan» som vil avslutte QR skanneren.

Bestemmer plukk og plasser posisjoner

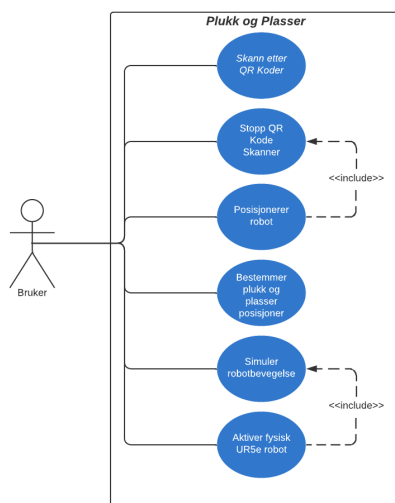
Bruker kan bevege på to objekter som bestemmer plukk og plasser posisjonene.

Simuler robotbevegelse

Bruker trykker på «Plan Trajectory». Hologrammet av UR5e roboten simulerer da plukk og plasser bevegelsen mellom de to virtuelle objektene.

Aktiver fysisk robot

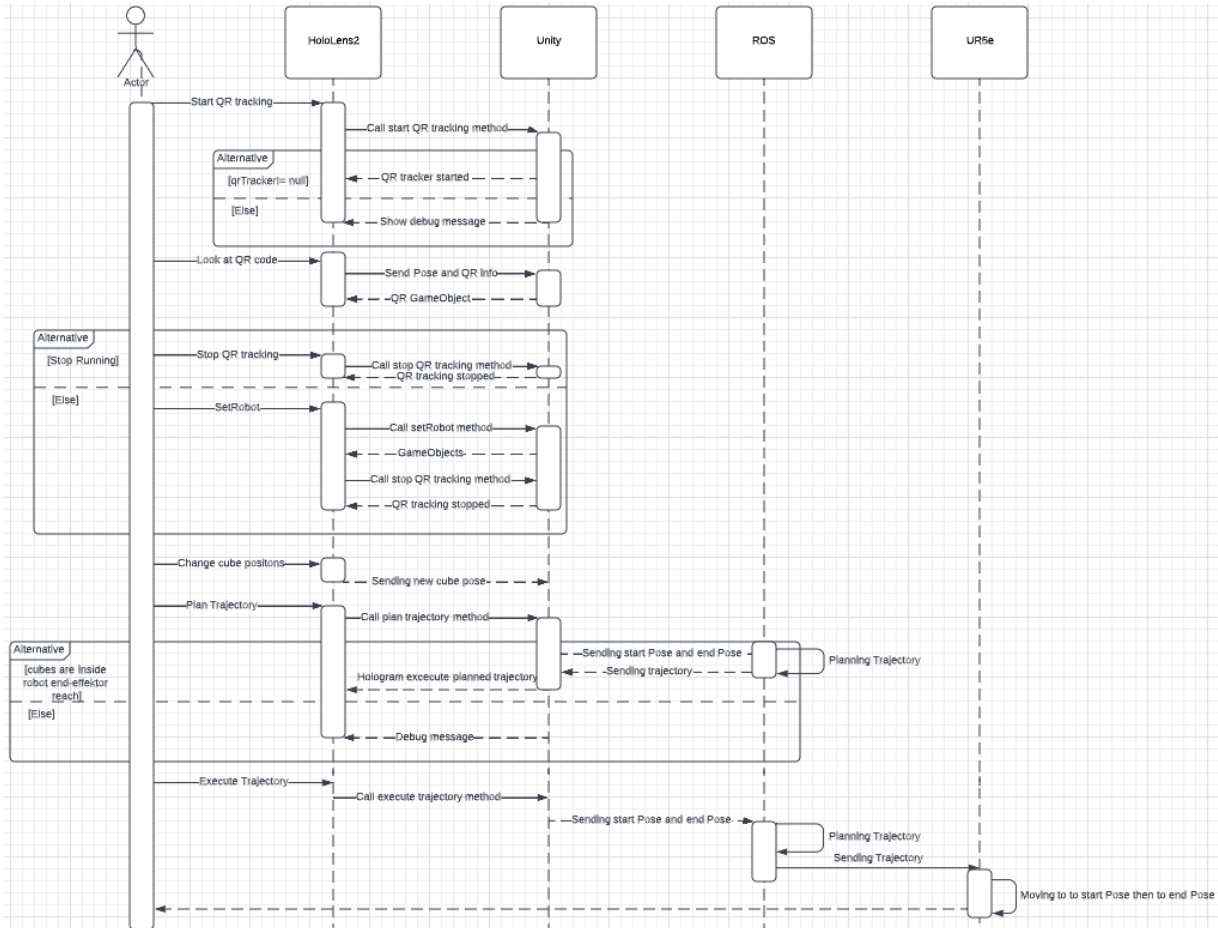
Bruker trykker på en «Execute Trajectory» som sender plukk og plasser posisjonen til fysisk UR5e robot. Hologrammet av UR5e roboten vil simulere bevegelsene den fysiske roboten utførte for å ende opp med samme leddverdier.



Figur 18 Brukstilfellediagram

4.8.2 Sekvensdiagram

Sekvensdiagram beskriver prosessen fra bruker starter søk etter QR koden, til den fysiske roboten har utført plukk og plasser funksjonen.



Figur 19 Sekvensdiagram av de viktigste brukertilfellene: Sekvensdiagram. Forklarer sekvensen fra start QR til robotarm har utført bevegelse.

5 RESULTATER

5.1 Evalueringsmetode

5.1.1 SCRUM

Gruppemedlemmene jobbet ofte på forskjellige arbeidsoppgaver under prosjektet. Med lite forkunnskaper og tidligere erfaring med mange av komponentene var det viktig å teste ofte. For å være sikre på at løsningene var integrerbare ble SCRUM et viktig verktøy. Ved å jobbe i korte iterasjoner lot det gruppen evaluere hverandres kode og sørge for at arbeidet hvert gruppemedlem hadde utført var kompatibelt. Det lot også gruppen evaluere om arbeidet som var gjort var i tråd med problemstillingen og målet gruppen jobbet mot.

5.1.2 Evaluering av løsningen

Effektivitet

Målet med oppgaven var å lage et system som demonstrerer fordelene av MR som et verktøy på industriell robotikk. Gruppen har ikke tilgang på en industriell robot til testingen av løsningen så effektiviteten blir derfor testet opp mot UR5e sin egen integrerte løsning. Testen gikk ut på at brukeren skulle simulere en plukk og plasser funksjon liknende den under pallestering. Tiden det tok for brukeren å legge inn bevegelsene ble målt fem ganger ved å bruke HoloLens 2 og fem ganger ved bruk av den integrerte løsningen. Brukeren var kjent med begge systemene og roboten var ferdig satt opp og klar til bruk. Det eneste brukeren trengte å gjøre var å sette plukk og plasser punktene og utføre bevegelsen med roboten. Tiden ble stoppet da robotarmen var kommet til sluttpunktet(plasser). Testen ble utført uten gripper på robotarmen montert da fokuset på oppgaven var at robotarmen skulle utføre banen fra start- til sluttpunkt.

Testing av effektivitet

Testen som var planlagt med doktorgradsstudentene er beskrevet i bildeseriene under. De skulle først få en demonstrasjon om hvordan systemet fungerte før de etterpå skulle utføre samme steg som er beskrevet i bildeserien. Først ved UR5e sin integrerte løsning, så gruppens løsning.

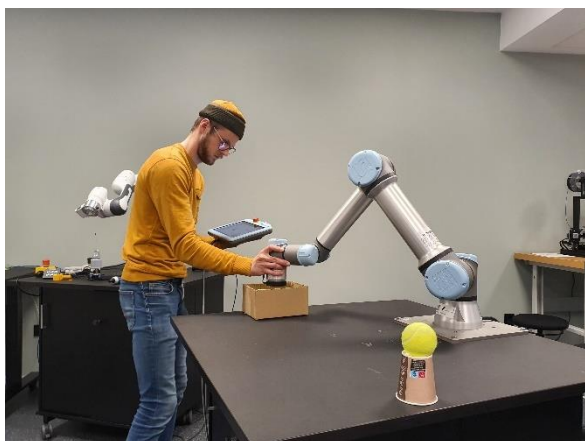
UR5e:



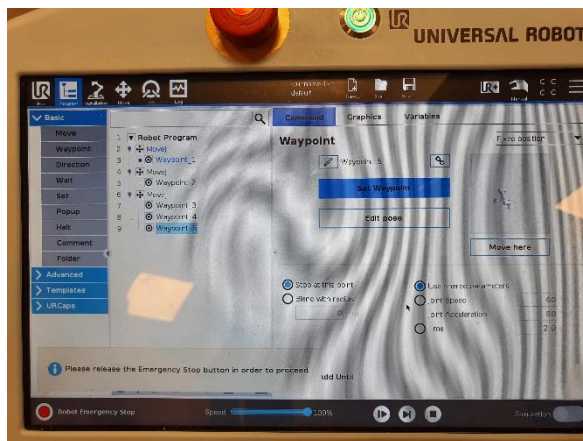
Figur 20: Startoppsett for testing



Figur 21: Første posisjon er gitt til robot



Figur 23: Sluttposisjon er gitt til robot



Figur 24: Når start og sluttposisjon er lagt inn kan bruker utføre bevegelsen

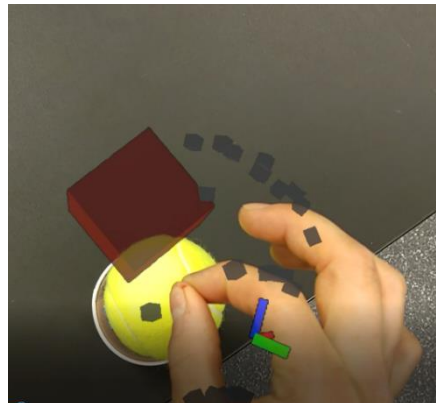


Figur 22: Testen avsluttes da robot har beveget seg til sluttposisjon

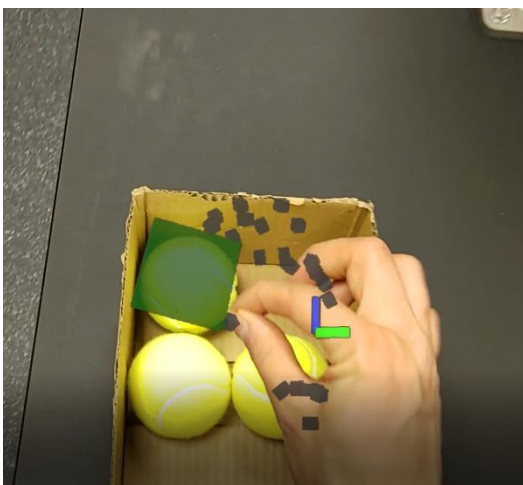
Testing ved bruk av MR



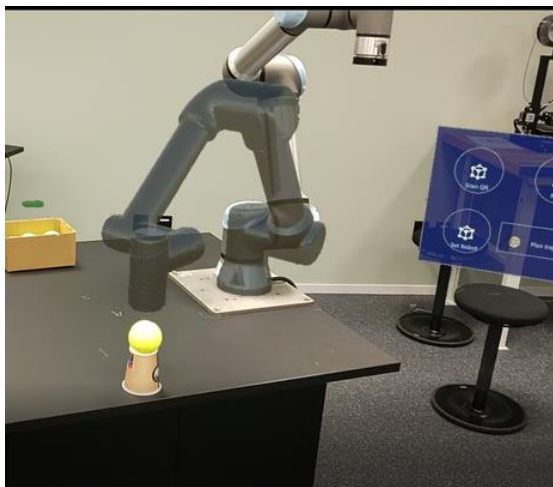
Figur 27: Startoppsett for testing



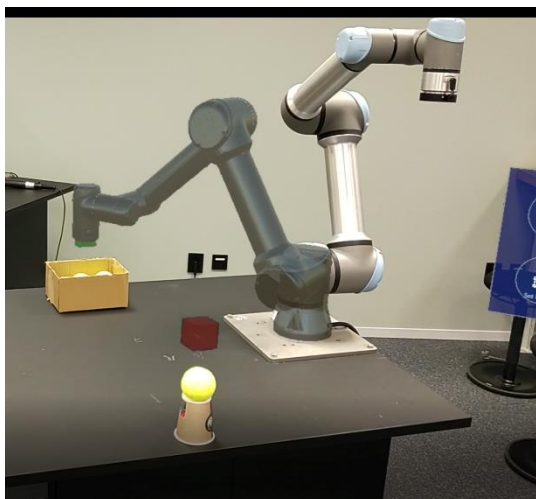
Figur 26: Plukk-objektet blir posisjonert



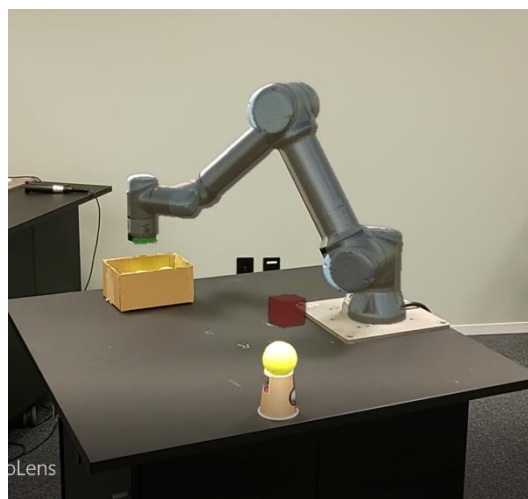
Figur 28: Plasser-objektet blir posisjonert



Figur 25: Hologram visualiserer planlagt bane



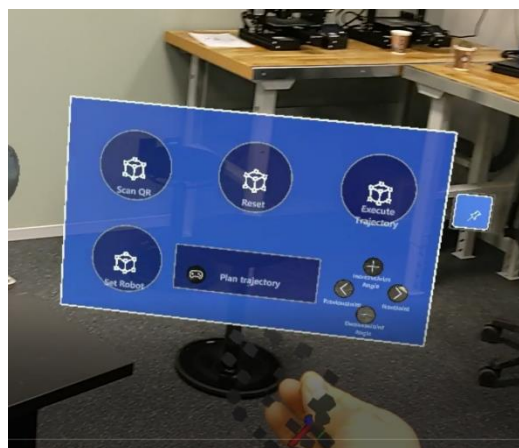
Figur 30: Hologram når sluttpunktet



Figur 29: UR5e har utført bevegelse og er i sluttposisjon

Brukergrensesnitt

For at løsningen skulle være intuitiv å bruke og kreve lite forkunnskaper var det viktig å evaluere brukergrensesnittet. Et spørreskjema ble derfor konstruert som gruppemedlemmene fylte ut etter systemet var testet. Testen ble utført ved at hver av gruppemedlemmene først prøvde ut UR5e sin integrerte løsning, etterfulgt av å bruke løsningen med HoloLens2. Spørreskjemaet var tiltenkt å bruke på doktorgradsstudentene, men det ble ikke gjennomført. Spørreskjemaet ligger under vedlegg.



Figur 31: Kontrollpanelet til systemet

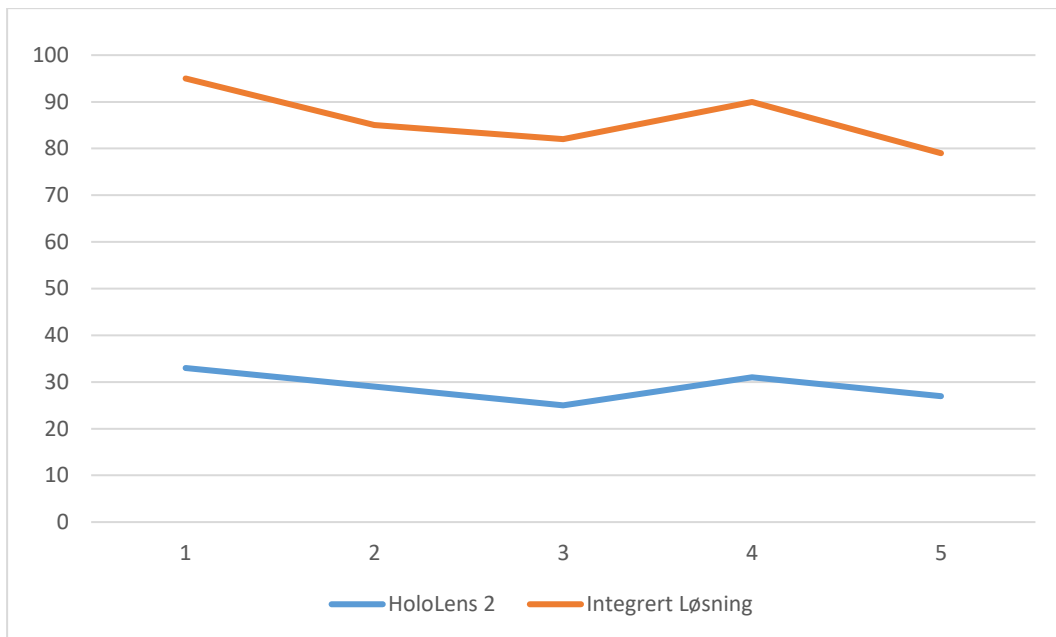
5.2 Evalueringresultat

5.2.1 Effektivitet

Målet for oppgaven var å lage et system som demonstrerer fordelene av MR som et verktøy på industriell robotikk. Målet tok inspirasjon fra RobNor sin VR løsning hvor programmeringstiden gikk ned med 80 prosent etter teknologien ble implementert. Det var derfor naturlig at gruppen målte effektiviteten av systemet med tid som faktor.

Etter testen var utført fem ganger på hver løsning kom gruppen frem til at løsningen med MR var raskest med en gjennomsnittsutførelse på 29 sekunder. Utførelse med UR5e sin integrerte løsning brukte i gjennomsnitt 86 sekunder. Tiden det tok å utføre bevegelsen med MR gikk dermed ned 66% sammenlignet med UR5e sin integrerte løsning.

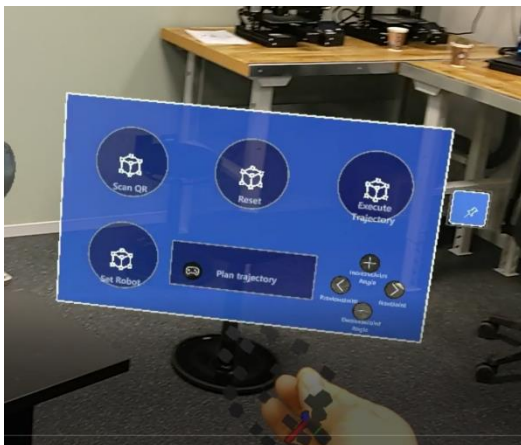
Denne testen ble utført av gruppemedlemmene og har dermed begrenset verdi. Resultatet er med som en indikasjon på at systemet har potensial. Dette skal testes i senere tid med en nøytral gruppe.



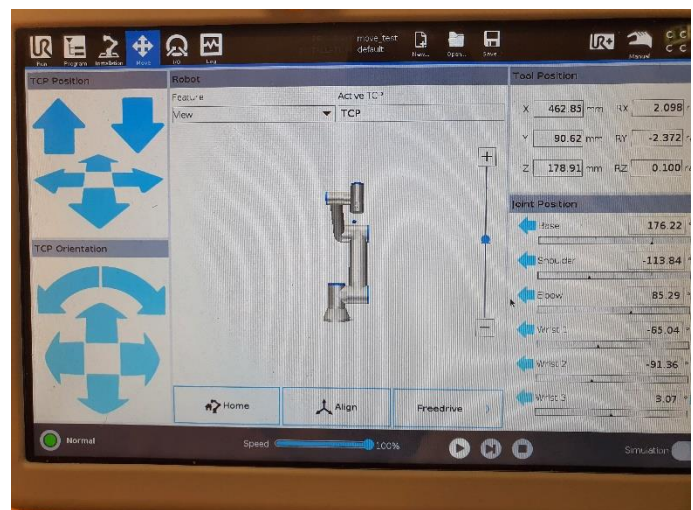
Figur 32: y-akse: tiden hvert forsøk tok. x-akse: representerer hvert forsøk

Brukergrensesnitt

En viktig del av løsningen var å ta ned kompleksiteten rundt det å programmere en robot. Kontrollpanelet som er designet til oppgaven var, basert på gruppemedlemmenes opplevelser intuitiv å bruke sammenlignet med den som kommer med UR5e. Her skal det også testes i senere tid opp mot en nøytral gruppe da gruppemedlemmenes meninger kan oppfattes som subjektive.



Figur 33: Kontrollpanel MR løsning



Figur 34: Kontrollpanel UR5e

6 DISKUSJON

Når gruppen begynte arbeidet var det ingen av medlemmene som har hatt tidligere erfaring med HoloLens eller Unity. Dette kapittelet handler om hvordan gruppen ser tilbake på valgene som ble tatt under arbeidet og hvordan det kan ha påvirket resultatet. Det vil også være diskusjon på hva som kunne bli gjort annerledes, og hvordan disse valgene kan ha endret utfallet.

6.1 Fremgangsmåte og valg

Da gruppen først kontaktet Trolltunga var det usikkert hva oppgaven skulle dreie seg om. Prosjektet startet med møter hvor veileder Martin Stølen og oppdragsgiver Atle Rettedal kom med ulike forslag til løsninger sammen med gruppen. Trolltunga var interessert i å anskaffe mer informasjon rundt MR teknologi, så valget landet på dette.

Det første gruppen måtte se på var om det var mulig å kommunisere mellom HoloLens2 og robotarmer. Det lå kildekode på nettet om sammenkobling mellom Unity og ROS som på den tid virket enkel å gjennomføre. Alternativet var å bruke ABB's plattform, RobotStudio[xxx], til koblingen. Gruppen ønsket også å kunne bevege på roboten og for å spare tid ble Movelt valgt for å planlegge robotens bevegelse da en på gruppen hadde tidligere erfaring med dette. Siden Movelt's ikke var kompatibel med Yumi, roboten Trolltunga tilbydde å sende opp, ble en UR5e valgt som robotarmen til prosjektet sammen med ROS da denne støttet Movelt.

Gruppen trengte et brukstilfelle for å demonstrere anvendeligheten til AR innen robotikk, som skulle dekke noen behov gruppen hadde satt til prosjektet. Roboten skulle styres ved bruk av AR. Brukeren skulle kunne utføre hele prosessen kun med HoloLens2-enheten. Omfanget av skulle ikke overstige det som ble sett på som overkommelig av gruppen, innenfor fristene gitt av Høgskolen. Plukk og plasser brukstilfellet vi endte opp med fylte alle kravene.

Tidlig i arbeidet ble gruppen tipset om å bruke en arbeidsmetodikk. Siden det ville bli mye koding og mange små implementeringer falt valget på Scrum.

Den endelige løsningen måtte være brukervennlig og lett å forstå. For at det skulle være enkelt for brukeren ble plukk og plasser posisjonene bestemt av to kuber som var intuitive å flytte på. Ønsket var også at brukeren skulle kunne se og få en forståelse av bevegelsen den fysiske robotarmen skulle ta. Dette krevde at basen til hologrammet ble plassert på basen til den fysiske UR5e roboten. HoloLens 2 har en innebygget QR søking som gir enheten tilgang til QR kodens Pose. Dette gjorde at gruppen kunne bruke en fysisk QR kode som referanse for hvor hologrammet skulle bli plassert.

Den endelige løsningen måtte evalueres for effektivitet og brukervennlighet. Gruppen valgte å evaluere i to grupper. En gruppe med kjennskap til utstyret og AR teknologi, og en gruppe uten tidligere kjennskap til prosjektet. En kort demonstrasjon ble gitt til gruppene før evalueringen startet. Å dele gruppene inn på denne måten gav gruppen ett bedre overblikk over brukervennligheten.

6.2 Refleksjon rundt valgene

Valget av programmeringsplattformer og robot henger sammen. Enten skulle det brukes RobotStudio med Yumi, eller ROS og Unity med UR5e. Om RobotStudio og Yumi hadde blitt valgt, hadde alle komponentene i systemet vært ukjente, i stedet for bare ett som det var i dette tilfellet. I gjengjeld hadde tilkoblingsmetode vært et mye mindre emne. Omfanget av oppgaven hadde her vært litt annerledes, og det er mulig at dette hadde vært den beste løsningen. Selv om gruppemedlemmene var kjent med UR5e og ROS fra før, er det mulig at flere funksjonaliteter i systemet kunne blitt programmert inn med RobotStudio og Yumi. Kanskje, i dette tilfellet, at gruppemedlemmene kunne ha brukt mindre tid på tilkobling, og mer tid på implementering av funksjoner og i gjengjeld få et bedre svar på problemstillingen.

Brukstilfellet valgt for oppgaven skulle demonstrere hvorvidt MR var brukbart innen robotikk. Om vi hadde gått for maling av deler med MR i stedet for plukk og plasser, ville mest sannsynlig løsningen vært mindre fullstendig. Dette hadde gjort at en klar konklusjon ikke kunne trekkes, og prosjektet i sin helhet hadde tapt kvalitet. Det ble i tillegg valgt plukk og plasser med tanke på at kanskje gruppen kom lengre enn planlagt. Dette hadde gjort at om gruppen var effektive nok, ville det kunne bli implementert flere funksjoner for systemet enn først planlagt.

Scrum var en viktig del for fremgangsmåten. Scrum lot gruppen jobbe målrettet mot samme mål, i tillegg til å bidra til å opprettholde god kommunikasjon i gruppen. Om gruppen ikke hadde brukt dette verktøyet kan det tenkes at resultatet hadde blitt veldig annerledes. Fordelene med scrum er at det er laget for koding. I tillegg har scrum link til github som også var aktivt i bruk.

Når det kom til å sette opp programmene for koding i Unity til HoloLens viste det seg å være mer komplisert enn antatt. Flere av programmene er versjonsavhengige da teknologien er ny og nye oppdateringer kommer jevnlig. En ting som raskt ble oppdaget var at Unity og Visual Studio komponentene som kreves for programmering i Microsoft HoloLens prosjekter kun var compatible med Windows 10. Sett i ettertid kunne gruppen spart mye tid på å undersøke kompatibiliteten til de ulike komponentene i forkant. Dette er god lærdom å ta med i videre prosjekter.

7 KONKLUSJON OG VIDERE ARBEID

7.1 Konklusjon

Formålet med oppgaven var å finne ut om MR er klar for industriell robotikk. Fra dette ble det konkludert med noen krav til oppgaven. Først måtte gruppen sette seg inn i emnet, og kom frem til at det var visualisering og kontrollering av en robotarm som skulle være løsningen. Ut dette begynte prosjektet å ta form. Det ble videre diskutert litt rundt nøyaktig hvordan løsningen skulle utformes. Videre gikk vi inn på en detaljert løsningsbeskrivelse som forklarer hvordan de viktigste komponentene i systemet virker. Valideringen av prosjektet skjedde gjennom en iterativ prosess med veileder, oppdragsgiver, og gruppemedlemmene internt. I tillegg har doktorgradsstipendiatene vært hjelpsomme både under utvikling og under testing.

Når det kom til testing, var det et ønske om å måle effektiviteten av løsningen. Tiden ble her redusert med 66%. Dette kan tyde på at det er noe potensiale i teknologien. Nøyaktigheten til HL2 begrenser nøyaktigheten til hele systemet. Det ble observert slingring på rundt 3cm på det meste. Dette innebærer at kalibrering av hologram over fysisk UR5e er gjort korrekt. Resultatet kan tyde på at i noen tilfeller, vil det være hensiktsmessig å implementere et slikt system innen industriell robotikk. En mulig løsning på dette problemet er å ta i bruk andre former for ankring til den ekte verden, eksempelvis HTC Vive Pro sine kontrollere.

Kravene vi satte til oppgaven er til dels dekket. Systemet er intuitivt og enkelt å forstå, men bruken av kontrolleren i prosjektet virket ikke på den måten den skulle. Det var vanskelig å treffe knappene med fingeren. Dette betydde at under testing, måtte bruker stå lengre borte fra kontrolleren enn ønskelig.

Med hensyn på dette, blir det konkludert med at målet for oppgaven er nådd. Svaret er at i enkelte tilfeller kan teknologien implementeres. Eksempler kan være plukk og plasser av bokser hvor nøyaktighet ikke er like viktig. Et annet eksempel er demonstrering av robotsystemer i sin helhet. Med dette menes det at det kan lastes opp ferdige robotsystemer til HoloLens 2. Denne kan sendes til kunder for visualisering av mulige fremtidige robotikk integrasjoner.

7.2 Videre Arbeid

Etter ferdigstilt prosjekt har gruppen lært en del om nåværende MR teknologi, og hvordan denne kan anvendes videre. Det er mulig å se for seg mange forskjellige brukstilfeller for teknologien. Videre reflekteres det rundt videre arbeid konkret av oppgaven, etterfulgt av mulige andre anvendelser av teknologien innen robotikk.

I 3.1 ble det diskutert ulike implementeringer av funksjoner for systemet. Om en skulle ha fortsatt prosjektet, ville implementering av disse vært hensiktsmessig å starte med. Dette vil gi sluttbruker flere alternativer for utføring av mange forskjellige oppgaver. Det anbefales spesifikt banemanipulering, baneopptak ved hjelp av invers kinematikk i Unity og visualisering av krefter og arbeidsområde for robotarmen.

En enkel metode for implementering av en invers kinematikk (IK) på simulert robotarm i Unity er allerede påbegynt. Denne baserer seg på den enkleste formen for IK, gradient descent. Denne tar som inngang distansen fra hvert ledd til målet, regner ut hvilken retning leddene må beveges for at distansen skal bli mindre, beveger leddene i denne retningen, og gjentar prosessen. Ved prosjektslutt er ikke denne fungerende, men bør enkelt kunne modifiseres til formålet. Om dette lykkes vil opptak av baner i rommet være mulig å gjennomføre, etterfulgt av sanntidskontroll av robotarmen.

Om en finner en koblingsmetode mellom Unity og Matlab, vil dette kunne hjelpet sluttbruker med forståelsen av aktuell robotarmbevegelse. For rekkevidde er det også mulig å opprette et script som viser rekkevidden av robotarmen, eller stopper plukk- eller plasser-objektet når det er på grensen av rekkevidden.

8 Referanser

ABB, u.d. *ABB YuMi*. [Internett]

Available at: <https://new.abb.com/products/robotics/collaborative-robots/irb-14000-yumi>

[Funnet 16 Mars 2022].

C. P. Quintero, S. L. M. K. P. W. P. C. H. F. M. V. d. L. a. E. C., 2018. *IEEE Xplore*. [Internett]

Available at: <https://ieeexplore.ieee.org/document/8593700>

[Funnet 18 Februar 2022].

H. Zhang, Y. W. J. Z. a. J. Y., 2018. *IEEE Xplore*. [Internett]

Available at: <https://ieeexplore.ieee.org/abstract/document/8468190>

[Funnet 28 April 2022].

Makhataeva, Z. a. V. H. A., 2020. *MDPI*. [Internett]

Available at: <https://www.mdpi.com/2218-6581/9/2/21/htm#B79-robotics-09-00021>

[Funnet 18 Februar 2022].

Manring, L. a. P. J. a. P. D. a. B. B. a. M. D. a. H. T. a. C. A., 2019. *Springer Link*. [Internett]

Available at: https://link.springer.com/chapter/10.1007/978-3-030-12243-0_2

[Funnet 18 Februar 2022].

RobNor, u.d. [Internett]

Available at: <https://www.robno.se/virtual-reality>

[Funnet 22 March 2022].

RobotNorge, u.d. *ROBOTNORGE*. [Internett]

Available at: <https://robotnorge.no/>

[Funnet 28 Januar 2022].

Trolltunga Robotics AS, u.d. *Trolltunga Robotics*. [Internett]

Available at: <https://www.trolltunga-robotics.no/hjem/>

[Funnet 29 April 2022].

UNIVERSAL ROBOTS, u.d. *Universal Robots ur5e*. [Internett]

Available at: <https://www.universal-robots.com/no/produkter/ur5-robot/>

[Funnet 17 Mars 2022].

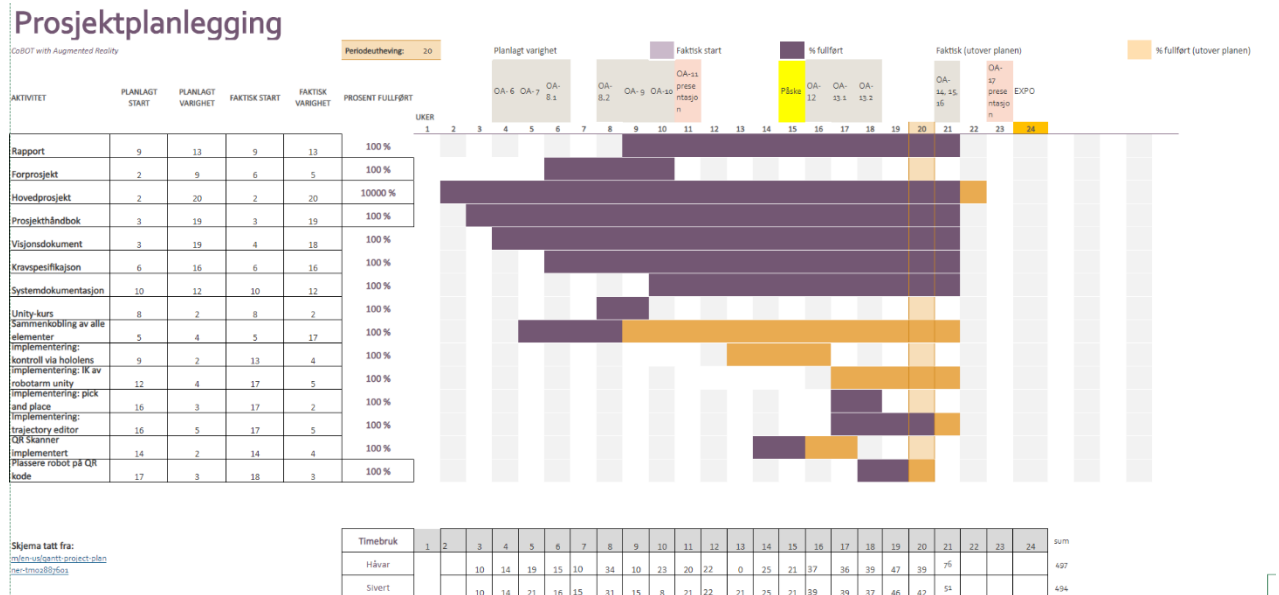
Wiki Ros, u.d. *Wiki Ros*. [Internett]

Available at: <http://wiki.ros.org/urdf>

[Funnet 09 april 2022].

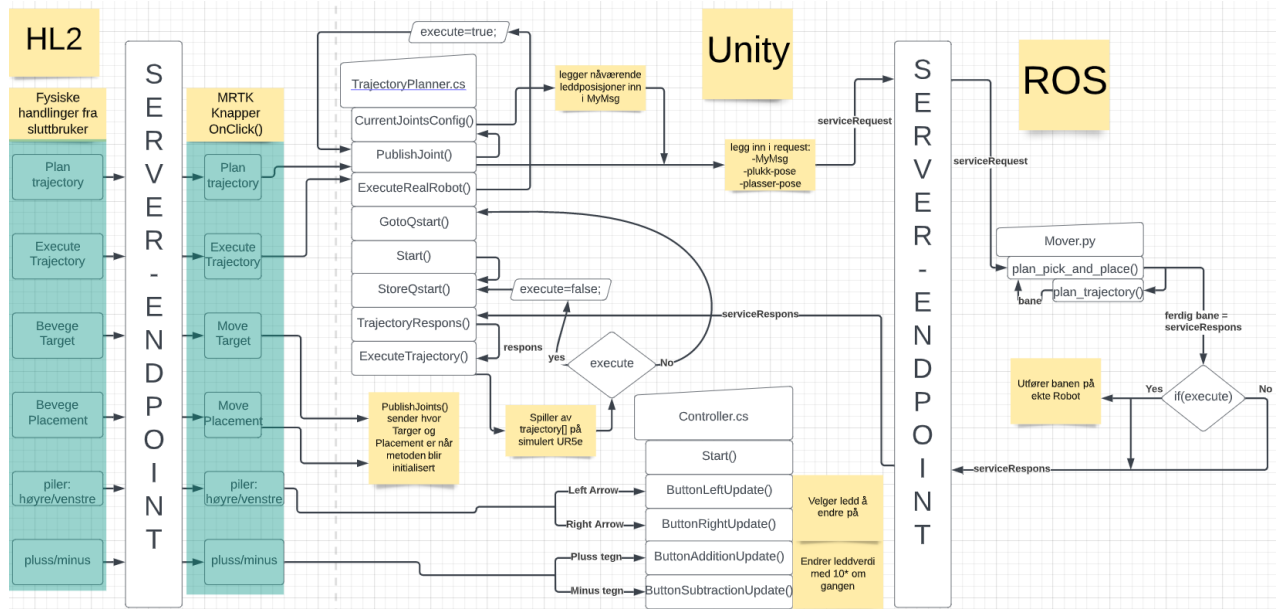
9 VEDLEGG

9.1 Gannt



9.2 Wireframe

9.3 Sammenheng mellom Unity og ROS med fokus på Unity



9.4 GitHub link til kildekode

<https://github.com/havadank/MyBachelor.git>

9.5 youtubevideo av brukstilfellet

<https://www.youtube.com/watch?v=QJSgIUxqN80>

Denne er med kalibrering av posisjonen til hologrammet. Vi brukte en del tid på dette her. Om en bare ønsker brukertilfellet, kan man spole til 07:30

9.6 Spørreskjema for gruppetesting

Pre-Test Questionnaire

I'm not a robot



1. Age: __

2. Gender: __ male __ female

3. What is your affiliation with the university?

__ undergraduate student

__ graduate student __ staff __ other

4. Do you have any physical impairments? (select all that apply)

__ reduced vision, if so is it corrected with glasses or lenses?

__ yes __ no

__ reduced arm mobility, if so is it likely to affect your speed and accuracy in movements?

__ yes __ no

6. Do you have previous experience with 3D input devices (for computer games, CAD software etc.)?

__ yes __ no

if yes, what kind? _____

,and have you used hand gestures? __ yes __ no

7. Do you have previous experience with robotics (robot arms, mobile robots etc.)?

__ yes __ no

if yes, what kind? _____

(Do not fill out) Subject ID: _____

Post-Test Questionnaire

I'm not a robot



How satisfied are you with.... (1: not satisfied, 5: satisfied)	1	2	3	4	5
Features of the product					
User friendliness					
System speed					
Time it took to program					
The movement of the simulated robot arm					
The stability of the holograms					
The movement of the real robot arm with respect to the planned path you saw beforehand					
The usage of AR to control the real robot arm					
The usage of AR to visualize information of the real robot					
The usage of cubes to mark the real time position of pick and place					
The control panel					

Do you have any comments?

(do not fill out) Subject ID: _____