



Norges miljø- og
biovitenskapelige
universitet

Masteroppgave V2020 30 stp.
Fakultetet for realfag og teknologi

Mobil robot til inspeksjon og deteksjon av fuktighet i bygninger

Mobile robot for inspection and detection of
moisture in buildings

Christian R. Hoffstad
Maskin, Prosess og Produktutvikling

Mobil robot til inspeksjon og deteksjon av fuktighet i bygninger

En mulighetsstudie for bruk av mobil inspeksjonsrobot i bygde miljøer

Av

Christian R. Hoffstad



Masteroppgave
Fakultetet for realfag og teknologi
Faculty of Science and Technology
Våren 2020

FORORD

Masterprosjektet markerer slutten på mitt femårige sivilingeniørstudie innen maskin og produktutvikling, ved Fakultetet for realfag og teknologi ved Norges miljø- og biovitenskapelige universitet (NMBU). Gradsarbeidet er gjennomført våren 2020 og er på 30 studiepoeng.

Helt siden barndommen har jeg interessert meg for teknologi og hvordan enkle komponenter kan settes sammen til noe større, mer komplekst og tilslutt velfungerende maskiner. Da jeg var elev ved Sandefjord videregående skole, fikk vi anledning til å dra på et bedriftsbesøk hos Komplett Group AS. Der ble jeg introdusert for fremtidens teknologi. Lagersystemet på Komplett Group AS var automatisert ved bruk av roboter som beveget seg rundt, hentet artikler og fraktet dem til bestemte arbeidsstasjoner. Dette var fascinerende. På NMBU har jeg fått kunnskap til å videreføre interessene med å finne løsninger på utfordringer i hverdagen. Jeg ønsket også å tilegne meg kompetanse innen robotikk. Dette danner grunnlag for mitt masterprosjekt.

Gradsarbeidet har vært lærerikt, tidkrevende og utfordrende. Det har vært mange opp og nedturer, men læringsutbyttet har vært stort. Jeg har tilegnet meg verdifull kunnskap innenfor mine interesseområder. Dette er kunnskap jeg vil ta med meg inn i mitt yrkesliv. Jeg håper også at mitt gradsarbeid er starten på et robotkonsept som kan videreutvikles til å bli et hjelpemiddel for inspektører og fagfolk innen inspeksjon og deteksjon av fuktighet i bygninger.

Jeg vil takke mine veiledere, førsteamanuensis Lars Grimstad og førsteamanuensis Jan Kåre Bøe for alle diskusjoner og veiledningstimer. En spesiell takk til Lars som introduserte meg for robotikk og som gav meg muligheten til å arbeide med dette konseptet. En spesiell takk også til Jan Kåre for lærerike diskusjoner og som utfordret meg til å se nye og andre muligheter med dagens teknologi.

Jeg vil også takke robotikkgruppen ved Fakultetet for realfag og teknologi ved NMBU som inviterte mastergruppen med til workshop i Brasil, for å lære mer om robotikk. En takk også til Tormod Aurlien, Ingunn Burud, Andreas Flø ved NMBU, Per Asbjørn Trevland ved Mortens Rørinspeksjonsfirma AS og Kolbjørn M. Jenssen ved Mycoteam for at dere tok dere tid til å besvare mine spørsmål, diskutere mine idéer og konseptet. Deres konstruktive og informative tilbakemeldinger har bidratt til dette resultatet.

Takk til min familie, mor, far og lillebror, for deres støtte gjennom sivilingeniørutdanningen ved NMBU som også inneholdt et spennende utvekslingssemester ved Technische Universität München. Takk til Ingrid for hjelpen med å ferdigstille masterprosjektet. Takk også til mine medstudenter på «Hemsen», studiet ville ikke vært det samme uten dere.

Ås, 29.05 2020



Christian R. Hoffstad

SAMMENDRAG

Forskning og studier viser at dagens og fremtidens samfunn vil bli påvirket og utsatt for store klimaendringer. Ofte og kraftigere ekstremvær vil forårsake økte og hyppigere forekomster av fuktrelaterte problemer i bygde miljøer som følge av lekkasjer og innsig av vann. Derfor er intensjonen med gradsprosjektet å utrede, utvikle og foreslå et konsept og system for en mobil robot som skal kunne gjennomføre inspeksjoner og deteksjoner av fuktighet på vanskelig tilgjengelige områder i bygningsmassen. Formålet med roboten er å bistå og være et verktøy til fagfolk. Gradsarbeidet ble delt inn i flere faser for få til en strukturert og oversiktlig fremdrift i prosjektet.

I planleggingsfasen ble mål, rammer og metoder presentert. Det ble formulert problemstillinger, fokuspunkter og teknologiske utfordringer. Det ble satt opp en arbeidsplan, definert hoved- og delmål, og begrensninger slik at man hele tiden hadde fokus mot å nå prosjektets intensjon. Det ble benyttet anvendelige metodeverktøy som waterfall, integrert produktutvikling (IPD), quality function deployment og pughs metode, SCAMPER og modularisering. I tillegg ble relevante løsningsverktøy som Python og SolidWorks benyttet.

Utredning- og læringsfasen handlet om å tilegne seg en overordnet kunnskapsbase om roboter og aktuelle applikasjoner. Her ble man kjent med et bredt spekter av roboter og deres bruksområder. Basert på denne kunnskapsinnhenting ble det kartlagt hvilke utfordringer den kommersielle roboten vil møte under arbeid. Det vil si at god mobilitet, fremkommelighet og manøvrering er essensielle egenskaper. Muligheten til å inspisere og detektere i ulike høyder og vinkler er andre viktige egenskaper. Videre i fasen ble det utredet teknologi og underliggende teori som roboten må utstyres med for å løse arbeidsoppgaven. I tillegg ble fagområder som sensortechnologi, mikrokontroller, programvare, navigering og HSV-fargemodell gjennomgått.

Konsept- og systemspesifiseringer av den kommersielle roboten ble utarbeidet i utviklingsfasen. Det ble etablert konkrete målsettinger for både konseptet og systemet, samt systemkrav og viktige sensoregenskaper som roboten må besitte for å kunne bli en funksjonell inspeksjonsrobot. Manøvrering, inspisering og detektering, og karakterisering av objekt er viktige systemdelmål. Forslag til viktige delmål for konseptet til den kommersielle roboten er design-, bruker- og økonomiske mål. Det ble også anslått metriske grensespesifikasjoner og variasjonsbredder for de kritiske dimensjonene i robotens arbeidsområder. I tillegg ble det presentert en grovspefisering av dimensjonene på den kommersielle roboten.

Resultatet av den mobile roboten er presentert i utviklingsfasen der konsept og systemforslag til den kommersielle inspeksjonsroboten er gjennomgått. Robotkonseptet gir leseren en visuell tilnærming til selve roboten. Stikkord som beskriver robotens design og komponenter er at den skal være modulær, standardisert, enkel og symmetrisk. Hjul versus belter ble diskutert, hvor hjul ble valgt på dette utviklingsstadiet. Roboten utstyres også med en arm for å kunne inspisere og detektere i ulike høyder og vinkler. Med utgangspunkt i de konkrete systemdelmålene var det flere temaer som ble diskutert i utviklingsfasen. Det ble blant annet argumentert for hvorfor ROS skal være rammeverket for robotens programvare. Roboten skal fjernstyres manuelt av operatør gjennom et frontkamera og håndkontrolleren Logitech Gamepad F710. Dette gjøres ved hjelp av et ROS-program som konverterer kontrollkommandoene til hastighetskommandoer. Ved autonom navigering skal roboten generere et kart og lokalisere seg

selv ved hjelp av SLAM og ROS gmapping. Gmapping benytter en laserskanner og encodere til innhenting av data om omgivelsene. For å kunne navigere i det aktuelle arbeidsområdet vil ROS navigation stack benyttes. For å gjennomføre effektive inspeksjoner ble roboten utstyrt med et infrarødt kamera som observerer og inspiserer omgivelsene. Det ble utviklet et pythonprogram som analyserer og segmenterer ut de potensielle fuktige områdene fra kameraets termiske bilder. Det er disse områdene som trenger en ytterligere inspeksjon. Da vil en pinless fuktighetsmåler føres bort til det segmenterte området fra bildet, for å bekrefte eller avkrefte fuktighet. For å få en bedre forståelse og tilnærming av sensoroppsettet for applikasjonen samt undersøke om den utviklede programvaren egnet seg til å analysere de termiske bildene, skulle det blitt gjennomført en tidlig komponenttest. Testen med kameraet og fuktighetsmåleren utgikk som en følge av Covid-19, men programmet ble testet ved hjelp av et virtuelt oppsett.

Samtidig som gradsarbeidet representerer en start på konseptet, er målet om å utvikle, utrede og foreslå et konsept og system for en mobil inspeksjonsrobot nådd. For å få en kommersiell prototype bør det videre arbeidet begynne med en markedsundersøkelse med fokus på optimalisering av roboten. Deretter vil de neste stegene være å designe roboten og programmere det foreslåtte systemet.

ABSTRACT

Research and studies show that today's and future societies will be affected and exposed to major climate changes. More frequent extreme weather will cause increased and more often occurrences of moisture-related problems in buildings, due to leaks and water seeping. Therefore, this thesis intends to investigate, develop, and propose a concept and system for a mobile robot that will be able to perform inspection and detection of moisture in difficult accessible areas in buildings. The purpose of the robot is to assist professionals. The thesis was then divided into several phases to achieve a structured and orderly progress throughout the project.

Goals, frames, and methodologies were formulated during the planning phase. The research question was identified, focus points and related technological challenges were formulated. Furthermore, a project plan was established, main and sub-goals defined, as well as limitations. Applicable methods such as, Waterfall, Integrated product development (IPD), Quality function deployment (QFD), Pugh's method, SCAMPER, and modularization were used. Relevant tools as Python and SolidWorks were also used.

The purpose of the study and learning phase was to acquire an overall knowledge based on robots and relevant applications. Robots and their area of expertise were presented. Based on this knowledge, challenges the commercial robot will face during the work assignment were mapped out. Good mobility, accessibility and manoeuvring are essential qualities that the robot should possess. The ability to inspect and detect at different heights and angles are other important features. If the robot should be able to work as defined, it has to be equipped with relevant technology and theory. This is discussed later in this phase. Furthermore, subject areas such as sensor technology, microcontroller, software, navigation, and HSV colour model were reviewed.

During the development phase, concept- and system specifications of the commercial inspection robot were established. Specific objectives were formulated for both the concept and the system. Important concept properties, system requirements, and important sensor properties that the robot must possess to become a functional robot, were also highlighted. Navigation, inspection and detection, and object characterization are important sub-goals for the system. Important sub-goals for the commercial robot concept are design-, user- and economical goals. Metric boundary specification and range of variation were also estimated for critical dimensions in the robots' work areas. Additionally, a rough specification of the commercial robots' dimensions was presented.

The result of the mobile robot is presented in the development phase were concept and system proposal of the commercial inspection robot are outlined. First, the robot concept is presented to give the reader a visual approach to the robot. Keywords that describe the robots' design and components are modular, standardized, simple and symmetrical. Wheels versus belts were also discussed. However, since this is an early stage of development, we concluded that the robot should be equipped with wheels. The robot should also be equipped with an arm to be able to inspect and detect at different heights and angles. Based on the specific system sub-goals, several topics are discussed during the development phase. Among other things, it is argued why ROS should be the framework for the robots' software. An operator should be able to control the robot manually. This will be done through a front camera and the Logitech Gamepad

F710 hand controllers and a ROS program that converts the control signals to speed commands. In autonomous navigation, the robot will generate a map and locate itself using SLAM and ROS gmapping. Gmapping uses a laser scanner and encoders to gather data about the surroundings. To be able to navigate in the workspace, the ROS navigation stack will be used. To carry out effective inspections, the robot is equipped with an infrared camera that observes and inspects the surroundings. A python program was developed to analyze and segment out the potential moisture areas from the cameras' thermal images. The result is areas that need further inspection. Then, a pinless moisture meter will be routed to the segmented area from the image to confirm or disprove moisture. To gain a better understanding and approach of the sensor setup for the application, and to investigate whether the developed software is suitable for analyzing the thermal images, an early component test should have been performed. However, the test with the camera and moisture meter was not performed due to the Covid-19 situation. On the other hand, the program for analysing the thermal images were tested using a virtual setup.

Even though this thesis represents the start of the concept, the goal of developing, investigating and proposing a concept and system for a mobile inspection robot has been achieved. In order to get a commercial prototype, further work should begin with a market research. The focus should be on how to optimize the robot. Then, the next steps should be to design the robot and program the proposed system.

FORKORTELSER

Forkortelser	Beskrivelse
IPD	Integrated Product Development / Integreert produktutvikling
ROS	Robot Operating System
R&D	Research & Development / Forskning & Utvikling
CPU	Central Processing Unit
RAM	Random Access Memory
SI-enhet	Internasjonalt system for måling av fysiske størrelser
AMR	Autonomous mobile robot / Autonom mobil robot
AGV	Autonomous guided vehicle / Autonomt fjernstyrt kjøretøy
NASA	National Aeronautics and Space Administration
IR	Infrarødt
DC	Likestrøm
USB	Universal Serial Bus
LED	Lysdiode
I/O-port	Inngang/Utgangsport
KI	Kunstig intelligens
CAD	Computer Aided Design / Data assistert tegning
FoV	Field-of-View / Synsfelt
hFoV	Horizontal Field-of-View / Horisontalt synsfelt
vFoV	Vertical Field-of-View / Vertikalt synsfelt

INNHALDSFORTEGNELSE

FORORD.....	i
SAMMENDRAG	ii
ABSTRACT	iv
FORKORTELSER	vi
INNHALDSFORTEGNELSE	vii
1 INTRODUKSJON	1
1.1 Motivasjon og bakgrunn	1
1.2 Problemstilling.....	1
1.3 Oppdragsbeskrivelse.....	4
1.4 Fokuspunkter og teknologiske utfordringer	5
2 PROSJEKTRAMME	6
2.1 Prosjekt målsettinger	6
2.1.1 Hovedmål	6
2.1.2 Delmål	6
2.2 Tids og arbeidsplan med milepæler.....	6
2.3 Prosjektets begrensninger	8
3 METODEBESKRIVELSE	9
3.1 Terminologi	9
3.1.1 Koordinatsystem.....	9
3.1.2 Terminologi og begreper	9
3.1.3 Symboler	10
3.1.4 Formler og likninger.....	11
3.2 Metodebruk.....	11
3.2.1 Waterfall Development Method.....	11
3.2.2 Integrert produktutvikling	12
3.2.3 Quality Function Deployment og Pughs metode	14
3.2.4 SCAMPER	15
3.2.5 Modularisering	15
3.3 Løsningsverktøy og ressurser	15
3.4 Prosesstrinn.....	16
4 ROBOTER OG POTENSIELLE APPLIKASJONER	18
4.1 Klassifisering av roboter.....	18
4.2 Kartlegging av sammenlignbare inspeksjonsroboter.....	19

4.3	Betraktninger rundt prosjektets inspeksjonsrobot	23
5	TEKNOLOGI- OG TEORIUTREDELSE.....	25
5.1	IMRTBot	25
5.2	Sensorteknologi	26
5.2.1	Sensorklassifisering.....	26
5.2.2	Sensorer for navigasjon	27
5.2.3	Sensorer for deteksjon av fuktighet.....	30
5.2.4	Sensorytelse og pålitelighet.....	33
5.3	Mikrokontroller	33
5.4	Programvare og datasystemer.....	35
5.4.1	Robot Operating System	35
5.4.2	Fordeler ved bruk av ROS.....	38
5.5	Navigering av mobile roboter.....	38
5.5.1	Navigering med ROS	39
5.5.2	Occupancy Grid Map	39
5.5.3	Costmap.....	40
5.5.4	Global planlegger	41
5.5.5	Lokal planlegger.....	42
5.5.6	Odometri.....	43
5.5.7	Simultaneous Localization and Mapping.....	44
5.6	HSV fargemodell.....	45
6	KONSEPTSPESIFISERING	46
6.1	Konseptmålsettinger	46
6.2	Rangering av viktige konseptegenskaper	47
6.3	Metriske grensespesifikasjoner og variasjonsbredde	47
6.4	Metrisk grovspefisering for konseptet	48
7	SYSTEMSPESIFISERING	49
7.1	Systemmålsettinger.....	49
7.2	Systemkrav	49
7.3	Viktige sensoregenskaper for inspeksjons- og deteksjonsarbeidet.....	51
8	KONSEPT- OG SYSTEMBESKRIVELSE	54
8.1	Robotkonseptet	54
8.2	Robotarkitektur.....	57
8.3	Manøvrering av roboten	57
8.3.1	Manuell fjernstyring	57

8.3.2	Autonom navigering.....	59
8.4	Inspisering, detektering og verifisering	63
8.4.1	Inspisering og detektering med IR-kamera	63
8.4.2	Verifisering av observasjon ved hjelp av fuktighetsmåler	64
8.5	Analysering av termiske inspeksjonsbilder	65
8.6	Sammendrag av systembeskrivelsen	65
9	TIDLIG KOMPONENTTESTING	67
9.1	Målsetting	67
9.2	Beskrivelse av teknologi og materialer	67
9.2.1	Utstyr.....	67
9.2.2	Modellering av fuktighet.....	68
9.2.3	Analysering av termiske bilder	69
9.3	Metodisk fremgangsmåte	70
9.4	Resultat og diskusjon.....	71
10	PROSESSEVALUERING OG DISKUSJON.....	73
10.1	Konseptutviklingsarbeidet og forbedringspotensialer	73
10.2	Design, produksjon og kostnad	74
11	KONKLUSJON	76
11.1	Resultater og anbefalinger	76
11.2	Videre arbeid	77
11.3	Fremtidig utvikling av konseptet.....	78
12	REFERANSER	79
12.1	Skriftlige kilder.....	79
12.2	Nettbaserte kilder.....	79
13	VEDLEGG	84

1 INTRODUKSJON

Dette kapittelet setter gradsarbeidet i en kontekst, med å presentere motivasjonen og bakgrunnen for prosjektoppgaven, og oppdragsbeskrivelse.

1.1 Motivasjon og bakgrunn

Dagens samfunn påvirkes av den teknologiske utviklingen. Teknologien utvikler seg så raskt at det som er nytt i dag, er utdatert i morgen. En stor bidragsyter til denne utviklingen er bilindustrien [1]. I dagens samfunn er smarte, autonome biler ikke lenger fremtidens idéer, de er dagens teknologi. Det samme gjelder roboter.

Forfatterens første møte med roboter var hos Komplet Group AS. I 2013 hadde de Europas mest effektive lagersystem [2, 3]. Små roboter «føyk» rundt i et høyt tempo på toppen av en stor matrise av plastbokser. Hver robot hadde i utgangspunktet sin egen plukkordre, men ved større bestillinger samarbeidet flere roboter om oppgaven. Robotene hentet varene, fraktet dem direkte til bestemte arbeidsstasjoner hvor ansatte pakket varene til utsendelse. Å se de selvkjørende robotene flytte seg rundt i et høyt tempo uten å kolliderer, samtidig som de samarbeidet om plukkordrene, var fascinerende. Bedriftsbesøket gav også innblikk i hvordan arbeidet kunne effektiviseres, kostnader reduseres og lette på arbeidsoppgavene til de ansatte. Omvisningen vekket interessen rundt roboter og hvilke muligheter denne teknologien kunne gi.

Etter videre fordypning, viste det seg at roboter er komplekse systemer. Roboter er satt sammen av flere teknologiske fagfelt som krever en bred kunnskapsbase. I dag er robotteknologien en del av vår hverdag. Den benyttes innenfor mange fagfelt, som for eksempel industriell produksjon, kirurgisk virksomhet og til inspeksjons- og deteksjonsarbeid. Roboter er utstyrt med ulike funksjoner og applikasjoner som tilpasses de ønskede arbeidsoppgavene. Det vil bare være kreativiteten som setter grenser for robotenes utvikling.

Som maskin- og produktutviklingsstudent ved et «grønt» universitet, ønsker man å være med på å løse utfordringer som dagen samfunn står ovenfor. Gradsarbeidet vil være innovativt med hensyn til bruk av roboter og tilgjengelig teknologi for å løse utfordringer knyttet til klimaendringer.

1.2 Problemstilling

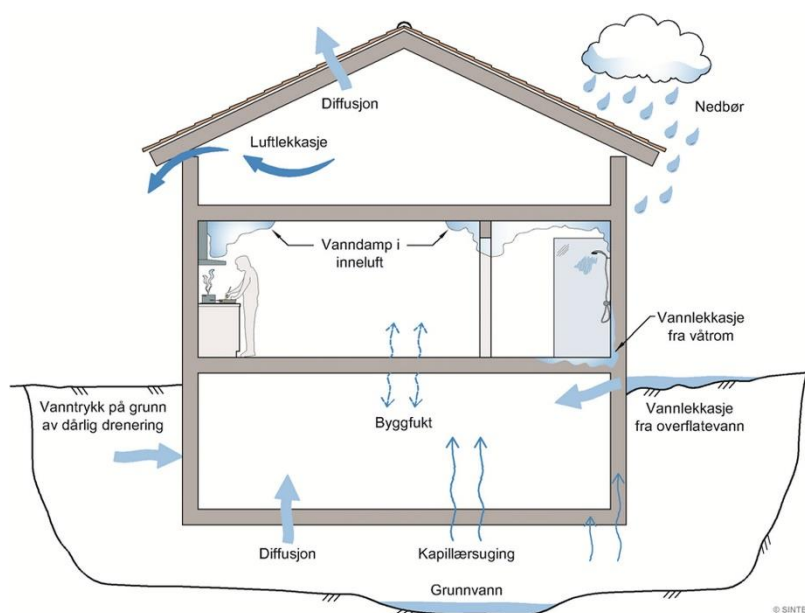
Forskning og studier gjort av FNs klimapanel viser entydig hvordan klimaet har endret seg de siste 100 årene. Ved regelmessige målinger av temperatur, ismengde, nedbør og pH-verdier i havet danner forskerne seg et bilde av klimaendringene. Basert på disse resultatene kan vi med sikkerhet si at den økte gjennomsnittstemperaturen på jorden medfører mer ekstremvær, også i fremtiden [4].

Med bakgrunn i den globale oppvarmingen har Direktoratet for samfunnssikkerhet og beredskap (DSB), i samarbeid med Meteorologisk Institutt (MET) og Norges vassdrag- og energidirektorat (NVE), samt andre statlige, regionale og lokale aktører utarbeidet rapporten *Krisescenarioer 2016* [5]. Det ene scenarioet er en risikoanalyse over en regnflom i Drammen

by. Dette krisescenariet kan fort være reelt i morgen. Situasjonen bygger på *Klima i Norge 2100*-rapporten [6] som er utviklet på oppdrag for Miljødirektoratet. I rapporten kommer det frem at Norge blir stadig oftere utsatt for ekstremvær. Med ekstremvær menes kraftig nedbør, overvann- og flomproblematikk, stigning i havnivået og økte stormflonivåer.

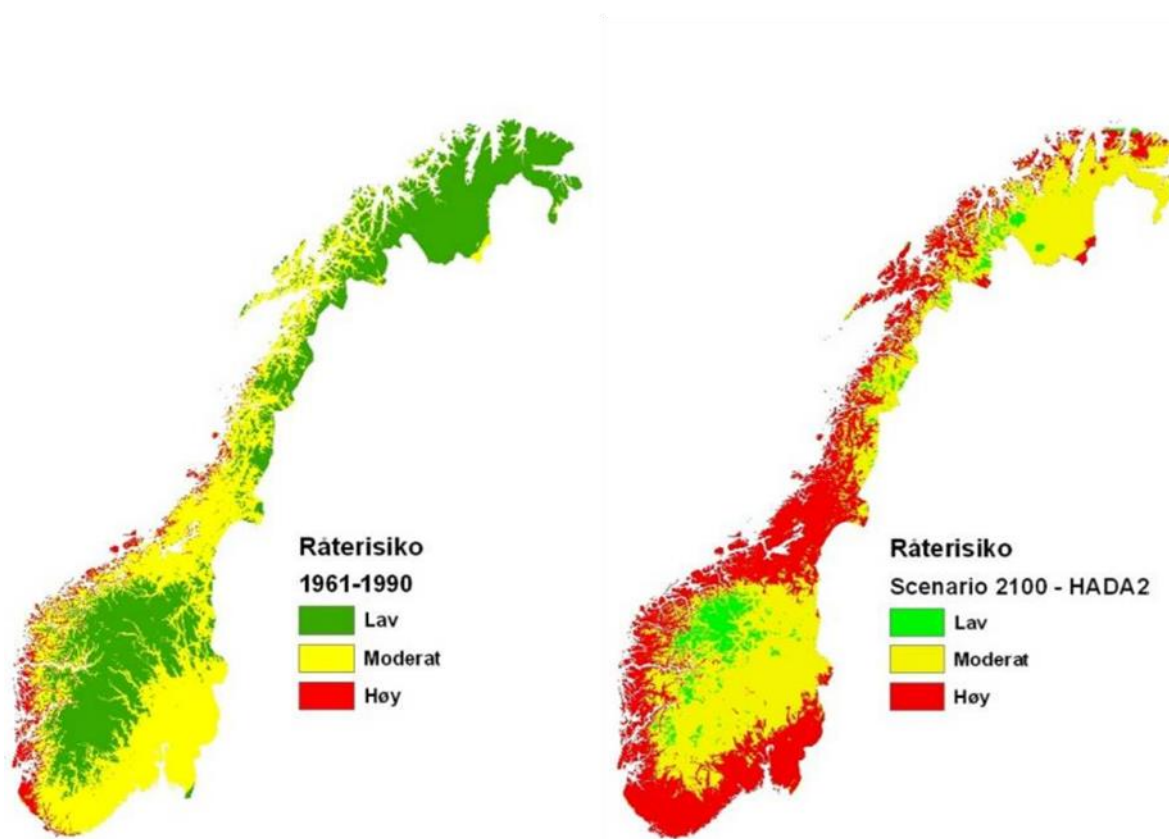
Ekstremværet skaper utfordringer og ødeleggelser for samfunnet [6]. All infrastruktur blir påvirket og utsatt i direkte møte og kontakt med vær, vind og nedbørsmengder. Alvorlighetsgraden vil variere, men behovet for inspeksjon og vedlikehold vil være til stede. Klimaendringene forsterker også behovet for preventive tiltak som økt krav til planlegging, materialvalg og utforming av samfunnets infrastruktur. Dette vil også være tilfellet for det bygde miljø, hvor lekkasjer og innsig av vann ofte oppstår som følge av denne utviklingen.

Flere bygg er i dag plassert på områder som er utsatt for kraftige nedbørsmengder, flomproblematikk og havnivåstigninger. Ifølge SINTEF Byggforsk utgjør ca. 75% av alle skader på bygg uoppdaget fuktighet. Som illustrert i figur 1.1 finnes det mange fuktilder. Konsekvensene av uoppdaget fuktighet over tid er mange. Denne fuktigheten har en negativ påvirkning på både materialer og konstruksjoner, samt et innemiljø som kan medføre helseproblemer [7]. Dette kan også føre til store økonomiske konsekvenser.



Figur 1.1: Mulige kilder til fuktighet i bygninger [7].

Ifølge SINTEF vil fuktighetsproblematikken være en stor utfordring i store deler av det bygde miljøet i dag og i fremtiden, i Norge [8]. I rapporten *Tilpassing til eit klima i endring* publisert av Norges offentlige utredninger, vises det til et råteomgang for perioden 1961 – 1990, samt en estimert utvikling frem mot 2100. Dette er vist i figur 1.2 [8, 9].



Figur 1.2: Råteutvikling i Norge fra 1961-1990, og forventet utvikling frem mot 2100 [9].

En annen direkte årsak av klimaendringene er biologisk vekst i bygningsmassen. Dette er en fellesbetegnelse på levende organismer som sopp, insekter og andre mikroorganismer som lever av næringsstoffene i disse miljøene. Det finnes ca. 144 000 kjente og klassifiserte sopparter [10]. Ifølge professor Katherine J. Willis ved Royal Botanic Gardens [10] finnes det mellom 2,2 og 3,8 millioner forskjellige sopparter på jorden, og av disse er ca. 7% kjente.

I forbindelse med bygninger klassifiseres sopp i to hovedgrupper, fargeskadesopp og råtesopp. Begge soppartene indikerer fuktighet, men i tillegg angriper råtesoppen treverket og bryter det ned slik at bæreevnen til konstruksjonen svekkes. *Serpula Lacrymans* som er illustrert i figur 1.3, er bedre kjent som «ekte hussopp». Denne har en stor utbredelsesevne, en aggressiv råtedannelse og bryter derfor trematerialene hurtig ned [11, 12].

De endrede klimaforholdene fører til en rekke utfordringer for det bygde miljø. Lekkasje og innsig av vann forekommer ofte i vanskelige tilgjengelige områder som for eksempel i kryperom, hulrom og kanal ganger. Dette er områder hvor det er vanskelig for inspektører å kontrollere for fukt og skader på bygningsmassen i form av råte og sopp. Dette gradsarbeidet skal derfor utvikle et konsept med bruk av en mobil robot. Roboten skal være verktøyet som fagfolk kan benytte til inspeksjoner i vanskelige tilgjengelige områder.



Figur 1.3: Uoppdaget fuktighet har resultert i at «ekte hussopp» har angrepet tak og pilarer [13].

1.3 Oppdragsbeskrivelse

Det skal gjennomføres et konseptutviklingsprosjekt og et mulighetsstudium relatert til utfordringer som er knyttet til klimaendringene. Det skal benyttes relevante utviklingsmetoder og prosjektet skal dokumenteres gjennom en skriftlig rapport og en muntlig presentasjon.

Gradsarbeidet skal ta utgangspunkt i de utfordringene klimaendringene påfører det bygde miljø. Uoppdagede lekkasjer og innsig av vann i bygninger er ofte en konsekvens av ekstremværet. Det skal derfor utvikles et konsept- og systemforslag til en mobil robot som skal inspisere og detektere den uønskede fuktigheten i bygninger. Roboten skal bistå fagfolk i områder hvor det er vanskelig å komme til. Det vil være alt fra kryperom, hulrom og andre trange rom som tekniske mellometasjer og kanal ganger i bygninger.

Det skal gjennomføres en kartlegging av mobile roboter som i dag anvendes til inspeksjons- og deteksjonsarbeid. Dette arbeidet vil danne grunnlaget for de teknologiske egenskapene denne roboten må utstyres med for å løse de tiltenkte arbeidsoppgavene.

Hovedfokuset til roboten vil være autonom navigering og manuell fjernstyring samt inspisere, observere og detektere fuktighet i bygningsmassen. Det vil også bli gjennomført en tidlig komponenttest for å kontrollere at sensoroppsettet for inspeksjon og deteksjon er velegnet til denne arbeidsoppgaven. Konseptet begrunnes gjennom nødvendig teori.

For å få en visuell fremstilling av den kommersielle prototypen, vil robotkonsept bli 3D modellert.

1.4 Fokuspunkter og teknologiske utfordringer

I utviklingen av den mobile roboten vil det være flere uforutsette utfordringer. For å nå hovedmålet er det viktig å kjenne til relevant teknologi og teori som kan anvendes i prosjektet.

I tabell 1.1 er det satt opp korresponderende fokuspunkter og teknologiske utfordringer. Dette er gjort for at prosjektet skal kunne utvikle et fullverdig produkt. På grunn av tidsrammen vil noen av fokuspunktene ikke bli utredet i dette prosjektet. Det kan derimot senere danne grunnlag for videre arbeid med inspeksjonsroboten.

Tabell 1.1: Korresponderende fokuspunkter og teknologiske utfordringer i prosjektet.

Tidlige fokuspunkter	Teknologiske utfordringer
a) Kartlegge markedet for sammenlignbare og relaterte inspeksjonsroboter.	a) Det finnes ingen direkte sammenliknbare produkter.
b) Finne passe teknologi og teori for å løse arbeidsoppgaven.	b) Finne optimale sensorer og programvare som detekterer fuktighet og som kommuniserer mellom sensorer, roboten og operatøren.
c) Konsept- og systemspesifisering.	c) Finne relevant teknologi slik at roboten får de egnede egenskapene.
d) Utvikle relevant programvare og verifisere funksjonalitet gjennom en tidlig komponenttesting.	d) Komme frem til de best egnede sensorene og analysemetode.

2 PROSJEKTRAMME

Prosjektet har en tidsbegrensning på 900 timer. For å oppnå hovedmålet innen tidsfristen er det hensiktsmessig å utarbeide en arbeidsplan med tydelige delmål.

2.1 Prosjekt målsettinger

Prosjektets hovedmål defineres i et tidlig stadium for å få en grunnleggende forståelse av prosjektets rammer. Hovedmålet deles deretter inn i delmål som sørger for at fremdriften opprettholdes og prosjektet gjennomføres, med en sluttrapport innen tidsfristen.

2.1.1 Hovedmål

Gradsarbeidets hovedmål er å utrede, utvikle og foreslå et konsept for en mobil inspeksjonsrobot. Roboten skal være et redskap for å observere, detektere og lokalisere fuktighet i bygningsmassen som er forårsaket av lekkasjer og innsig av vann. Arbeidet vil bli dokumentert gjennom en skriftlig rapport som beskriver konseptet.

2.1.2 Delmål

Delmålene defineres for å kunne nå hovedmålet innen tidsrammen. Følgende delmål inngår i prosjektet:

1. Utføre litteratursøk og research relatert til mobile roboter, utarbeide en prosjektplan, målsettinger og metodisk fremgangsmåte.
2. Redegjøre for teknologi og underliggende teori for denne inspeksjonsroboten.
3. Etablere konsept- og systemspesifikasjoner og egenskaper roboten skal besitte. Konseptet og systemet skal beskrives.
4. Det skal utarbeides et program som analyserer inspeksjonsbildene. I tillegg skal det ferdigstilles en 3D modell av konseptet, til en kommersiell prototype.
5. Tidlig komponenttest av sensoroppsettet for inspeksjon og deteksjon. Test av programmet i delmål 4 gjennomføres.
6. Evaluering og konkludering av gradsarbeidet og ferdigstilling av rapporten.
7. Innlevering og forsvaring.

2.2 Tids og arbeidsplan med milepæler

For å opprettholde en jevn og målrettet fremdrift, er det utarbeidet en arbeidsplan. Arbeidsplanen er presentert i tabell 2.1, og er delt inn etter delmålene slik at disse følges og fullføres. De gjennomførte delmålene, presentert i kapittel 2.1.2, blir markert med «✓» og representerer en fullført milepæl som presentert i tabell 2.2.

Tabell 2.1: Arbeidsplan og milepæler markert med «✓» [14].

Delmål	Aktivitet	Januar				Februar				Mars				April				Mai			
		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	Valg av setting og problemstilling																				
	Litteratursøk																				
	Research																				
	Metodikk					✓															
2	Redegjøre for relevant teknologi																				
	Redegjøre for nødvendig teori								✓												
	Workshop - Brasil																				
3	Spesifiseringer og mål																				
	Konsept- og systembeskrivelse												✓								
4	Programvare																				
	3D CAD modellering														✓						
5	Målsettinger																				
	Planlegging																				
	Setup																				
	Testing og revisjon																				
	Oppsummering																				
6	Evaluering og konklusjon																				
7	Skrive og rapport levert *)																				

Tabell 2.2: Delmålene inndelt i milepæler, med tidsfrist.

Delmål	Milepæl	Dato
1	Utarbeidet en prosjektplan, definert målsettinger, begrenset prosjektet og kartlagt metodikken.	09.02.2020
2	Redegjort for teknologi og underliggende teori for den mobile inspeksjonsroboten.	01.03.2020
3	Målsettinger og spesifikasjoner for konseptet og systemet er definert og beskrevet.	22.03.2020
4	Utviklet et program for analysering av inspeksjonsbilder og 3D CAD modellert et konseptforslag til en kommersiell prototype.	05.04.2020
5	Tidlig komponenttest er målsatt, planlagt, satt opp og gjennomført.	19.04.2020
6	Egenevaluering, diskutering og konkludering av prosjektet er gjennomført. Det er satt opp anbefalinger for videre arbeid av konseptet.	01.05.2020
7	Prosjektet er ferdigstilt, trykket og levert. *)	15.05.2020

*) På grunn av Covid-19 er det blitt forskyvninger i arbeidsplanen, og innleveringsdatoen er utsatt fra 15.05.2020 til 02.06.2020 og deretter forsvaring.

2.3 Prosjektets begrensninger

På grunn av masterprosjektets tidsramme på 900 timer brutto er det utarbeidet konkrete begrensninger. Disse er:

- Prosjektet er et mulighetsstudium (proof of concept). Det vil fokuseres på relevant teknologi og underliggende teori, samt nødvendige egenskaper som inspeksjonsroboten må besitte.
- Det vil ikke være en faglig fordypning på mykologi, læren om sopp.
- Det gjøres ingen matematiske fordypninger innenfor systemteorien.
- Konseptet vil bli 3D CAD modellert, men det er ikke et endelig design og utforming av roboten. 3D modellen vil bare være visuell presentasjon.
- Det vil ikke bli gjennomført grunnlagsberegninger på komponenter eller utredning av materialvalg.
- Det vil ikke bli redegjort for motorer og elektronikk roboten må utstyres med. Det er heller ikke valgt leverandører og modeller til sensorene.
- Roboten vil ikke bli programmert.
- Tidlig komponenttesting vil bli utført med tilgjengelig utstyr på universitetet.
- Det vil ikke gjennomføres en kostnads- eller livsløpsanalyse av konseptet.
- Det vil ikke utredes eller utvikles festemekanismer til sensor og utstyr som skal festes på roboten.

3 METODEBESKRIVELSE

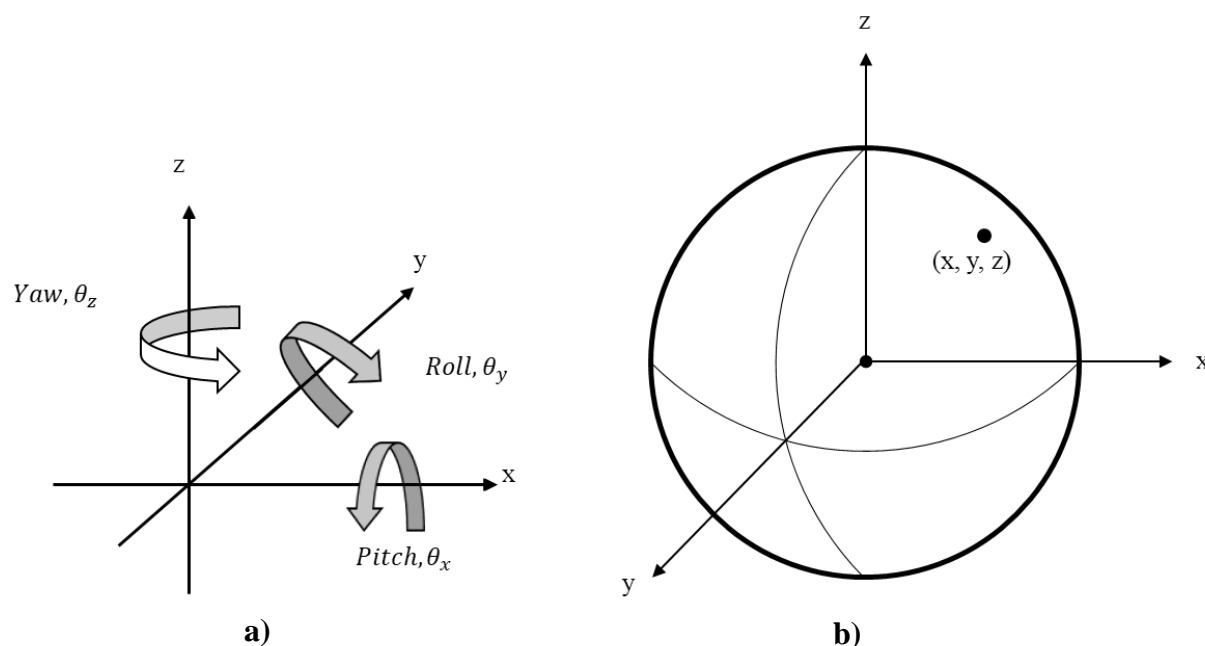
Det vil bli benyttet ulike metoder for å besvare problemstillingen. I dette kapittelet vil metoder for utviklingsprosessen og terminologi som blir benyttet gjennom rapporten presentert.

3.1 Terminologi

I dette underkapittelet presenteres relevant terminologi. Det blir definert et koordinatsystem, begreper og uttrykk blir presentert sammen med symboler og matematiske formler.

3.1.1 Koordinatsystem

Figur 3.1 introduserer aktuelle koordinat- og referansesystemer slik at robotens posisjon, orientering og retning kan estimeres i forhold til andre referansepunkter, objekter eller observasjoner. Figur 3.1a) introduserer et generelt koordinatsystem for roboten. Figur 3.1b) presenter et globalt referansesystem som gjør at roboten kan orientere seg i kartet.



Figur 3.1: a) Robotens koordinatsystem, med seks frihetsgrader. Positiv rotasjonsretning defineres av høyrehåndsregelen sett fra origo. b) Globalt referansesystem for orientering i kartet.

3.1.2 Terminologi og begreper

I tabell 3.1 presenteres begreper og uttrykk som blir benyttet i rapporten. De teoretiske begrepene er definert av eksterne kilder.

Tabell 3.1: Begreper med beskrivelse som benyttes i rapporten.

Teoretiske begreper	Beskrivelse
Kryperom	Luftrom mellom husets nederste etasje og bakkenivå.
Mykologi	Biologisk fagområde, læren om sopp.
Elektromekanisk system	Systemer satt sammen av elektriske og mekaniske komponenter.
Global ruteplanlegger	Navigasjonsalgoritmer som planlegger en optimal rute fra A til B for roboten.
Lokal ruteplanlegger	Ruteplanleggingsalgoritmer som sørger for at roboten unngår objekter som er til hinder for ønsket bevegelsesrute.
Translasjon	Bevegelse av et objekt uten rotasjon.
Ikke-destruktiv testing	Påvise feil uten å redusere objektets nåværende fysiske egenskaper.
IR-kamera	Infrarødt kamera.
Elektromagnetisk stråling	Energi i form av fotoner som beveger seg med lysets hastighet.
Elektromagnetisk spekter	Inneholder all elektromagnetisk stråling.
Mikrobølger	Elektromagnetiske bølger, som er en del av bølgespekteret.
Fixed-base manipulator	En stasjonær robotarm.
Open-source	Programvare som er frigitt av opprettshaver slik at alle brukere kan benytte seg av programmet til å studere, endre og distribuere etter eget ønske.
Mapping	Kartlegging av omgivelsene, basert på sensordata.
Pose	Robotens posisjon og orientering i henhold til et koordinatsystem
Airborne	Objekter som er og beveger seg i luften.
Terrestrial	Objekter som er og beveger seg på land.
Aquatic	Objekter som er og beveger seg i vann.

3.1.3 Symboler

Tabell 3.2 inneholder hovedsymboler som brukes i rapporten.

Tabell 3.2: Symboler med beskrivelse og benevninger som blir brukt i rapporten.

Symboler	Beskrivelse	Benevning (SI-enhet)
x	Koordinat i lengderetning	m
y	Koordinat i horisontalretning	m
z	Koordinat i vertikalretning	m
θ_x	Rotasjon om x-aksen, pitch	$^\circ$
θ_y	Rotasjon om y-aksen, roll	$^\circ$
θ_z	Rotasjon om z-aksen, yaw	$^\circ$
s	Strekning	m
v	Hastighet	m/s

Tabell 3.2: Forts.

Symboler	Beskrivelse	Benevning (SI-enhet)
t	Tid	t
d	Distanse	m
δ_{rot}	Vinkelrotasjon	$^{\circ}$
δ_{trans}	Translatorisk rotasjon	$^{\circ}$

3.1.4 Formler og likninger

Tabell 3.3 presenterer hovedformler og -likninger som blir brukt i rapporten for å beskrive teknologi og underliggende teori.

Tabell 3.3: Hovedformler og likninger som blir brukt i rapporten.

Beskrivelse	Formel	Indeks
Strekning	$s = vt$	3.1
Bevegelse x-retning	$x = v t \cos(\theta)$	3.2
Bevegelse y-retning	$y = v t \sin(\theta)$	3.3
Avstandsmåling for laserskanner og ultrasonisk sensor	$d_{ToF} = \frac{c \cdot t_{ToF}}{2}$	5.4
Avstand for encoder	$d_{encoder} = 2\pi r n$	5.5
Output - bevegelsesmodell	$u_t = \begin{pmatrix} \bar{x}_{t-1} \\ \bar{x}_t \end{pmatrix}$	5.6

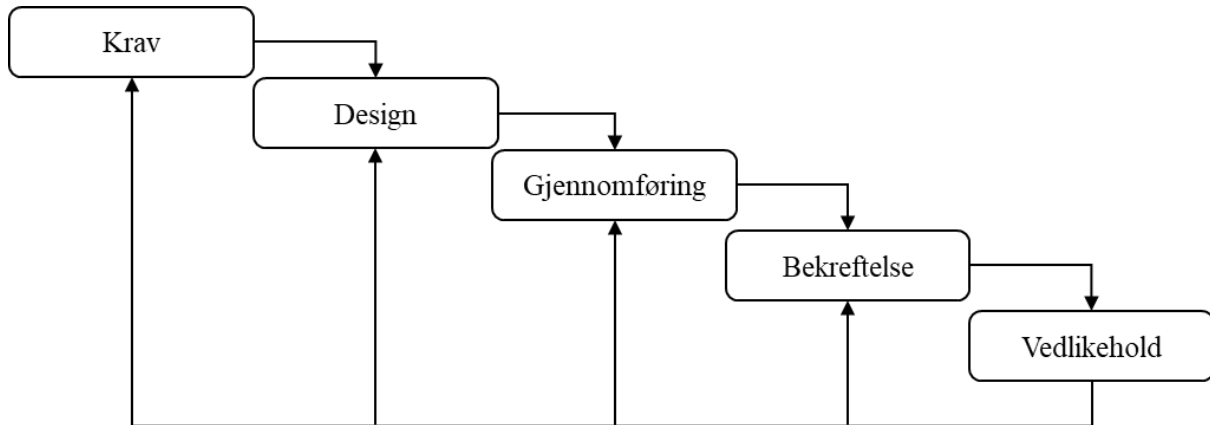
3.2 Metodebruk

I dette underkapittelet presenteres metodene som blir benyttet til utviklingsarbeidet. Metodene bidrar med struktur, effektivitet og sørger for at prosessen har en jevn fremgang også i perioder hvor uforutsette problemer og utfordringer oppstår.

3.2.1 Waterfall Development Method

Waterfall [15] er en oversiktlig og brukervennlig metode som er bygd opp som en trappetrinnsmodell. Modellen er vist i figur 3.2. Den sørger for at den teknologiske utviklingsprosessen begynner på toppen og jobber seg strukturert og gradvis nedover. Metoden er en lineær modell. Modellen er også satt opp med iterasjoner slik at man kan gå tilbake til et tidligere stadium og gjøre endringer, om uventede problemer eller utfordringer oppstår.

I gradsarbeidet benyttes waterfallmetoden som en del av den teknologiske utviklingen i prosesstrinnene. Metoden sørger for en jevn fremdrift gjennom alle prosjektets faser.



Figur 3.2: Waterfallmetoden.

3.2.2 Integreert produktutvikling

Integreert produktutvikling (IPD) [16] er en metodisk fremgangsmåte som tilrettelegger for en vellykket utviklingsprosess og egner seg for små og store prosjekter. Metoden fokuserer på samspill og tverrfaglighet mellom de involverte arbeidsgruppene og sørger for kommunikasjon ut til kunden gjennom hele prosessen. Dersom dette samarbeidet fungerer, vil resultatet være økt læringsutbytte, økt effektivitet innad i teamet og en kortere gjennomføringstid. Gode resultater oppnås gjennom god data- og informasjonsflyt og det vil nødvendig med en felles digital plattform hvor man tar i bruk datahjelpemidler og -kommunikasjon.

IPD benyttes ofte som en huskeliste, fordi den hjelper prosjektet til å prioritere oppgavene i riktig rekkefølge og at viktige elementer ikke blir glemt. Konsekvenser ved å ikke ta i bruk moderne IPD kan bli dårlig kommunikasjon, misforståelser, forsinkelser og kostnader som overskrider budsjett. Årsaken er ofte at hver fagavdeling arbeider alene frem til produktet settes sammen.

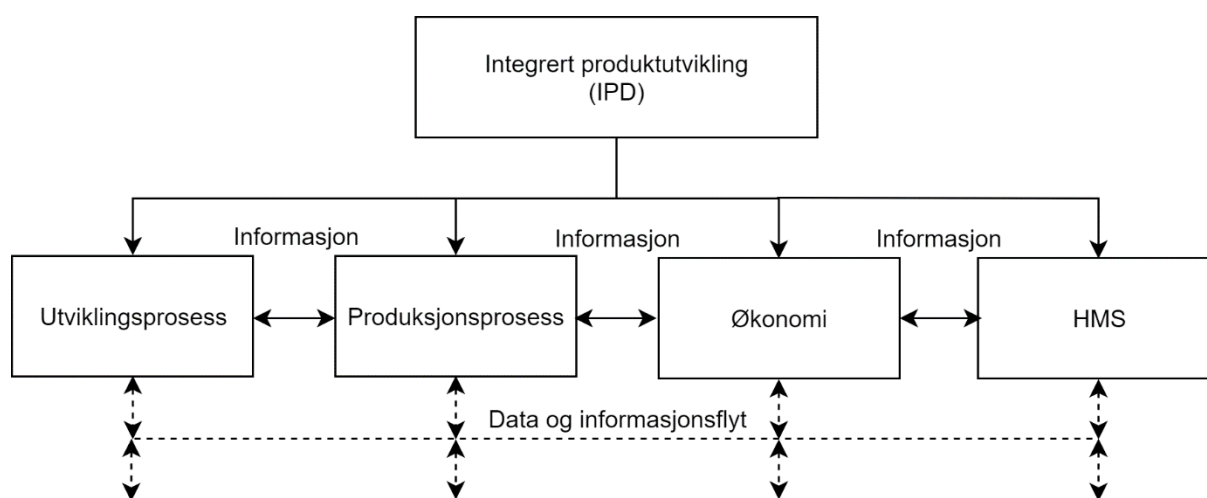
Hovedelementene i modellen er utvikling, produksjon, økonomi og helse, miljø og sikkerhet (HMS) som vist i figur 3.3. I gradsarbeidet fokuseres det på prosessen med denne produktutviklingen, samtidig som de økonomiske aspektene ved produktet og hvordan produktet kan bidra til bedre HMS. Tabell 3.4 presenterer hvordan IPD-prinsippene benyttes i prosjektet.

Tabell 3.4: Presentasjon av IPD-prinsippene og anvendelsen av dem i rapporten. Prinsippene følger DRM Associates [17].

Hovedprinsipp	Anvendelse av prinsipp
Kartlegge og forstå kundens behov	Behov og bruksområdet må kartlegges. Basert på dette skal det utarbeides en robot med hensiktsmessige egenskaper. Dette vil danne grunnlag for valg av teknologi og spesifikasjoner i konseptet.

Tabell 3.4: Forts.

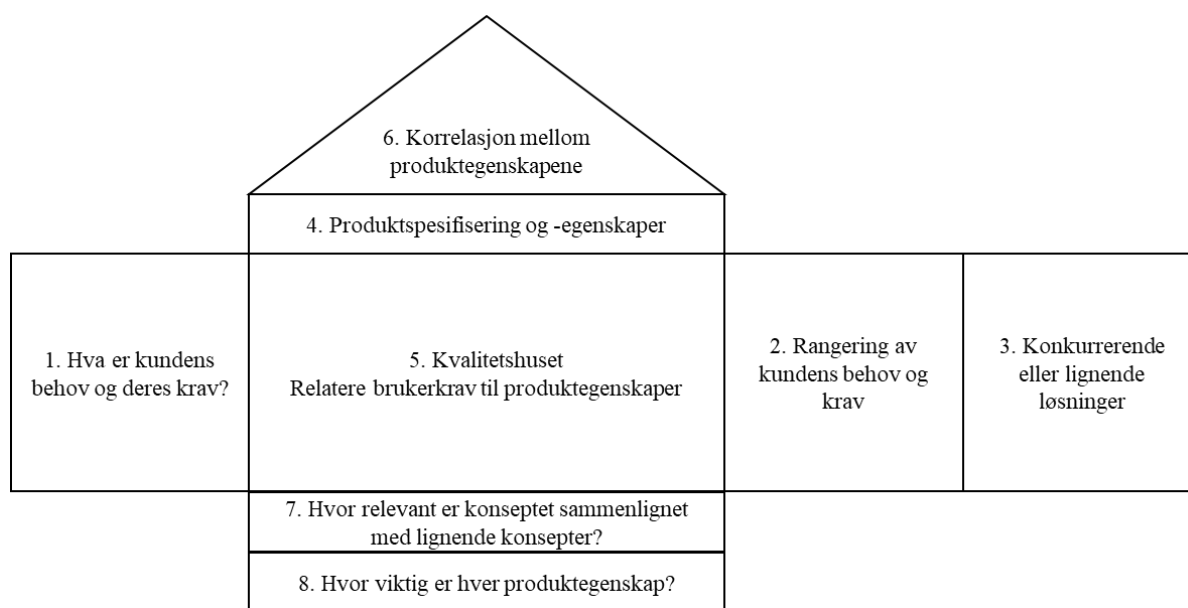
Hovedprinsipp	Anvendelse av prinsipp
Planlegging av utviklingsprosessen for produktet	I tidlig fase skal det utarbeides målsettinger hvor hovedmålet deles inn i flere delmål. Det skal også utarbeides en fremdriftsplan.
Benytte ulike produktutviklingsteam	Gradsarbeidet gjennomføres som et selvstendig prosjekt. Forfatteren vil benytte seg av intern veiledning fra flere fagmiljøer ved NMBU og ekstern fagpersonell for tilbakemeldinger og inspirasjon.
Integrert prosessdesign	Gradsarbeidet er et tidlig produktkonsept som gjør at det vil være muligheten for videre utvikling av konseptet gjennom endringer i valg av teknologi og funksjonelt design.
Kostnader til utviklingen av konseptet	I utviklingen av den kommersielle protypen skal det benyttes tilgjengelige og standardiserte komponenter og geometrier. Dette medfører modularitet og reduserte produksjonskostnader.
Utvikling av robust design	Før konseptet kan kommersialiseres må alle komponenter og programvare testes og kvalitetssikres med hensyn til robotens arbeidsoppgaver.
Integrering av relevant dataverktøy til utviklingsprosessen	SolidWorks blir benyttet til å modellere en 3D modell og Python til å utvikle et bildeanalyseprogram.
Kontinuerlig forbedring av utviklingsprosessene	Under utviklingsprosessen skal prosjektet kontinuerlig forbedres gjennom egnevaluering og diskusjoner med veiledere og fagpersoner. Det er derfor satt opp prosessstrinn med iterasjoner, slik at man kan gå tilbake å gjøre nødvendige endringer.



Figur 3.3: IPD-modellen med de fire hovedelementene. Modellen vektlegger data-, kommunikasjon- og informasjonsflyt mellom alle involverte parter. Dette er illustrert med stiplede linjer.

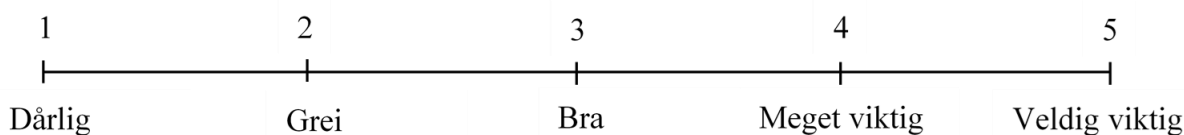
3.2.3 Quality Function Deployment og Pughs metode

Formålet med masterprosjektet er å utvikle et produkt som kunden vil benytte. Gjennom utviklingsprosessen er det derfor viktig å ha kunden i fokus. Dette gjøres ved å bruke Quality Function Deployment (QFD)[18]. Områder som bør vektlegges er kundens ønsker, behov og krav til produktet. For at prosjektet skal kunne ta i bruk QFD vil det være hensiktsmessig å benytte kvalitetshuset (House of Quality, HOQ). Prinsippet i HOQ er at prosjektet skal svare på noen spørsmål i kronologisk rekkefølge, disse er presentert i figur 3.4.



Figur 3.4: Grunnleggende prinsipper i House of Quality. Husets horisontale akse retter seg mot kunden og deres ønsker. Den vertikale delen representerer det tekniske fokuset.

Basert på kartleggingen av kunden og markedets behov kan man benytte Pughs metode som også inngår i QFD. Pughs metode [19] benyttes i prosjektet til å formulere og utarbeide produktspesifiseringer, hvor produktets målsettinger og egenskaper presenteres. Konseptegenskapene vektet med en skala på 1-5, med hensyn til hvor viktig egenskapene er for sluttproduktet. Skalaen er vist i figur 3.5. På denne måten konvergerer egenskapene mot de som egner seg best til sluttproduktet. Videre benyttes Pughs metode til å formulere robotens metriske grensespesifikasjoner og metriske grovspefisikasjoner. Den metriske grensespesifiseringen estimerer produktets yttergrenser, mens den metriske grovspefisifisering anslår et grovt dimensjonsestimat.



Figur 3.5: Vekting av egenskaper, hvor 1 er dårlig og 5 er veldig viktig i forhold til produktet.

3.2.4 SCAMPER

SCAMPER [20] er et brainstormingsverktøy som ofte benyttes i utviklingsprosesser for å utfordre eksisterende løsninger og konsepter. SCAMPER er et akronym for Substitute (S), Combine (C), Adapt (A), Magnify (M), Purpose (P), Eliminate (E), Reverse (R). Verktøyet utformer en rekke spørsmål som utvikleren kan benytte inn i prosjektet hvor man står ovenfor utfordringer og problemer med produktet. Her vil gamle og nye idéer vurderes, forenkles eller vinkles på andre måter, i håp om å finne nye løsninger. Metoden benyttes ofte ubevist.

3.2.5 Modularisering

Modularisering [20] er en teknikk der formålet er å dele produktet inn i mindre selvstendige moduler og hver modul har sin egen hovedfunksjon. Ved å bygge produktet opp på denne måten får man en fleksibel produktplattform som enkelt kan tilpasses ulike formål og arbeidsoppgaver. Hver modul kan enkelt modifieres, erstattes og flyttes mellom ulike systemer. Siden hver modul er uavhengig av hverandre kan produktet optimaliseres, ved å endre på modulenes sammensetning. Standardiserte enheter og dimensjoner, samt symmetri er derfor viktige egenskaper, fordi de gir økt fleksibilitet. Hvis produktet er modulbasert, kan det også settes sammen på fire ulike måter, lineær-, vinkelrett-, delvis integrert- eller i en integrert sammensetning.

I prosjektet benyttes modularisering gjennom hele utviklingsprosessen for å komme frem til et funksjonelt og anvendelig konseptforslag. Modulariseringen sørger også for at vedlikeholdsarbeidet blir enklere, fordi defekte komponenter enkelt kan skiftes ut og erstattes. Ved modularisering vil også utviklingskostnadene reduseres, fordi enkle, symmetriske, standardiserte geometrier og komponenter benyttes.

3.3 Løsningsverktøy og ressurser

Det er benyttet flere løsningsverktøy og ressurser for å tilegne seg den nødvendige kunnskapen innen emnet robotikk. Tabell 3.5 presenterer hjelpemidler og programvare som er benyttet i prosjektet.

Tabell 3.5: Programvare og hjelpemidler som er benyttet gjennom prosjektet.

Verktøy	Anvendelse
Microsoft PowerPoint 2016	Generering av figurer.
Microsoft Word 2016	Rapportskriving og tabeller.
Microsoft Excel 2016	Generering av tabeller.
Draw.io	Generering av figurer.
Python 3	Programmering av bildeanalyseprogram.
SOLIDWORKS 2019	3D CAD modellering.

Litteraturstudie

Det er gjennomført en litteraturstudie av relatert arbeid, relevant teknologi og underliggende teori for å bygge opp en grunnleggende kunnskapsbase. Litteraturstudiet baseres hovedsakelig

på kvalitetssikrede bøker utgitt av forlag, vitenskapelige artikler, research papers og relevante nettsider. Nettsidene er benyttet til påfyll av ytterligere informasjon som det henvises til i litteraturlisten.

Fagkompetanse

Gjennom prosjektet er det tatt kontakt med ekstern fagkompetanse for ytterligere innspill og påfyll av kunnskap til konseptet. Det er gjennomført et bedriftsbesøk til Mortens Rørinspeksjon AS som benytter mobile roboter til inspeksjon av rørledninger. Forfatteren opplevde besøket som nyttig og lærerikt. Det var også verdifullt å høre om selskapets erfaringer og hvilke egenskaper ved robotene de mente var viktige for deres bruksområder.

For å kartlegge konseptets potensiale ble det tatt kontakt med Mycoteam AS. De er et ledende rådgivningsfirma innen fuktproblematikk og biologiske bygningsskader. Tilbakemeldingene fra Kolbjørn M. Jenssen ved Mycoteam AS var både konstruktive og informative. Det var også motiverende at de viste interesse for konseptet og ønsket nærmere kontakt. Dessverre responderte Mycoteam AS på et sent stadium i gradsarbeidet som har medført at noen tilbakemeldinger er hensyntatt, mens andre innspill er nevnes i videre arbeid med konseptet.

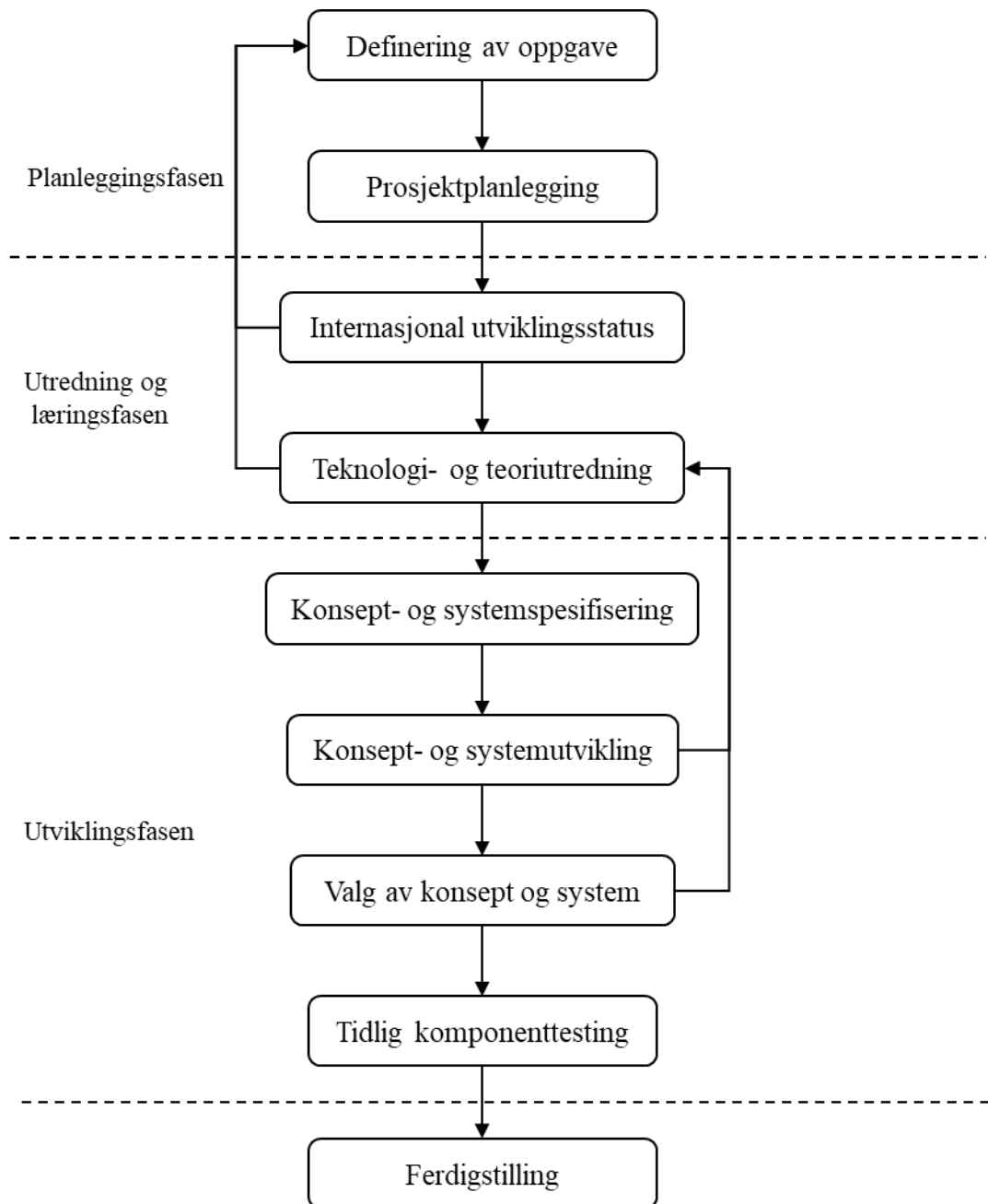
Professor Tormod Aurlien, professor Ingunn Burud og senioringeniør Andreas S. Flø ved NMBU har delt sine kunnskaper innen fuktproblematikk i bygninger, i bruken av IR-kameraer, samt arbeidet med analysering av de termiske inspeksjonsbildene.

I forbindelse med at prosjektet gjøres i samarbeid med robotikkgruppen ved Fakultetet for realfag og teknologi ved NMBU, er det gjennomført en workshop i Brasil. Forfatteren reiste hjem fra workshopen med en bredere og dypere innsikt i utviklingen av roboter til kommersielle arbeidsoppgaver. Workshopen ga også et innblikk i arbeidet som kreves for å utvikle en velfungerende robot. Studieturen gjorde det mulig å få diskutert teknologi som kunne benyttes til dette konseptet med andre studenter og ansatte ved vertskaps universitetet. Deres erfaringer og tanker rundt teknologien som er relevant var en viktig bidragsyter i utviklingsfasen av robotkonseptet.

Hovedveileder førsteamanuensis Lars Grimstad har bidratt til økt forståelse av hvordan roboter fungerer. Biveileder førsteamanuensis Jan Kåre Bøe har stilt mange gode spørsmål rundt konsept.

3.4 Prosesstrinn

Prosjektets utviklingsprosess er presentert i figur 3.6 på neste side. Formålet med prosesstrinnene er å få til en strukturert og oversiktlig disposisjon for prosjektet. Siden uforutsette aspekter, utfordringer eller nye vinklinger kan oppstå, er utviklingsprosessen satt opp med itereringer. Itereringene gjør det mulig å gå tilbake til tidligere stadier for å gjøre tilpasninger eller endringer dersom det er nødvendig. Den teknologiske utviklingsprosessen, bestående av utredning og læringsfasen, og utviklingsfasen bygger på waterfallmetoden, mens prosessdiagrammet presenterer hele prosjektet.



Figur 3.6: Skjematisk gjennomgang av utviklingsarbeidets prosesstrinn. Pilene som går tilbake viser itereringer.

4 ROBOTER OG POTENSIELLE APPLIKASJONER

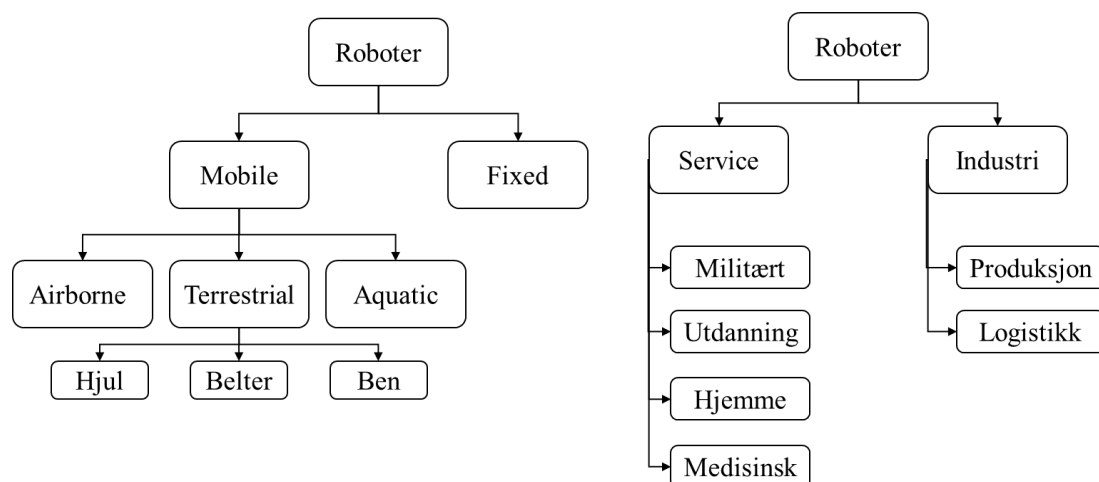
Mobile roboter har i flere år vært i bruk til ulike formål innenfor industrien, men i dag ser vi den samme utviklingen i de private hjem. I dette kapittelet vil det bli gitt en introduksjon til mobile roboter og deres bruksområder, i tillegg vil relaterte inspeksjonsroboter presenteres. Basert på dette, kartlegges det for hvilke utfordringer denne inspeksjonsroboten kan støte på og må løse.

4.1 Klassifisering av roboter

Roboter er dagens og fremtidens arbeidere. De er elektromagnetiske systemer som har evnen til å tilegne seg informasjon om omgivelsene de opererer i. Roboten kan ta beslutninger som er basert på de innsamlede dataene og deretter utføre ønskede handlinger [21]. For å utføre handlingene, er systemet satt sammen av sensorer, regulatorer og aktuatorer som kommuniserer med hverandre gjennom en tilpassede programvare.

Allerede i 1921 ble ordet «robot» benyttet av den tsjekkiske forfatteren Karel Capek i forbindelse med et skuespill. Capek mente ordet «robot» betød slave eller tvangsarbeider [22]. I dagens samfunn brukes roboter til å utføre arbeidsoppgaver som tidligere ble utført av arbeidere, på ett vis hadde Capek rett i sin tolkning. Roboter og deres applikasjoner har utviklet seg i takt med den teknologiske utviklingen. For at industrien skal kunne tilby produkter til en konkurransedyktig pris og god kvalitet, er de avhengig av å ta i bruk denne teknologien.

Roboter klassifiseres inn i to hovedgrupper, stasjonære og mobile roboter som vist i figur 4.1. Deretter organiseres mobile roboter i henhold til bruksområder. De første industrielle robotene som ble utviklet og tatt i bruk var fastmonterte gulvmodeller. Disse ble ofte kalt fixed-base manipulators. Dette var stasjonære roboter med dynamiske armer som ble brukt til repetitive produksjon- og monteringsoppgaver. Eksempler er sveising, montering av biler, kretskort til pakking av utstyr og pick-and-placeoppgaver i fabrikker [23]. Stasjonære roboter har et begrenset bruksområde.



Figur 4.1: Klassifisering av roboter i henhold til miljø og bruksområder.

De tradisjonelle robotarmene revolusjonerte industrien i det 20. århundret. Tilsvarende vil de mobile robotene være viktige for den industrielle utviklingen i det 21. århundret [23]. Der de stasjonære robotene overtok jobben til arbeiderne ved produksjonsbåndet, vil de mobile robotene overta arbeidsplassene som handler om bevegelse fra A til B. I motsetning til de stasjonære robotene, har de mobile robotene evnen til å bevege seg fritt rundt. Dette åpner for nye bruksområder. Mobile roboter kan legge bak seg store landområder, i dårlig definerte, usikre og dynamiske miljøer som ikke egner seg for mennesker [24]. Det samme vil gjelde helsefarlige arbeidsplasser hvor arbeidsmiljøene kan være utsatt for eksplosjonsfare, giftige gasser med mer [23]. Dersom en robotarm monteres på en mobil robot som vist i figur 4.2, vil bruksområdene ytterligere ekspandere.



Figur 4.2: Den mobile roboten Husky UGV er utstyrt med en robotarm. Armen gir roboten utvidede bruksområder [25].

Mobile roboter kan deles inn i to ulike kategorier, Autonomous mobile robots og Autonomous guided vehicle, henholdsvis kalt AMR og AGV. AMR er autonome mobile roboter som benytter seg av komplekse programvarer. Gjennom sensorer tilegner de seg nødvendig informasjon om omgivelsene og dette brukes til å lokalisere, mappe og navigere i området ved hjelp av aktuatorer. Det vil si at de beveger seg fritt rundt, med forhåndsprogrammert programvare eller at de bruker kunstig intelligens til å ta beslutninger. Derimot er AGV fjernstyrt med enkle guiding-systemer i forhåndsdefinerte omgivelser [26].

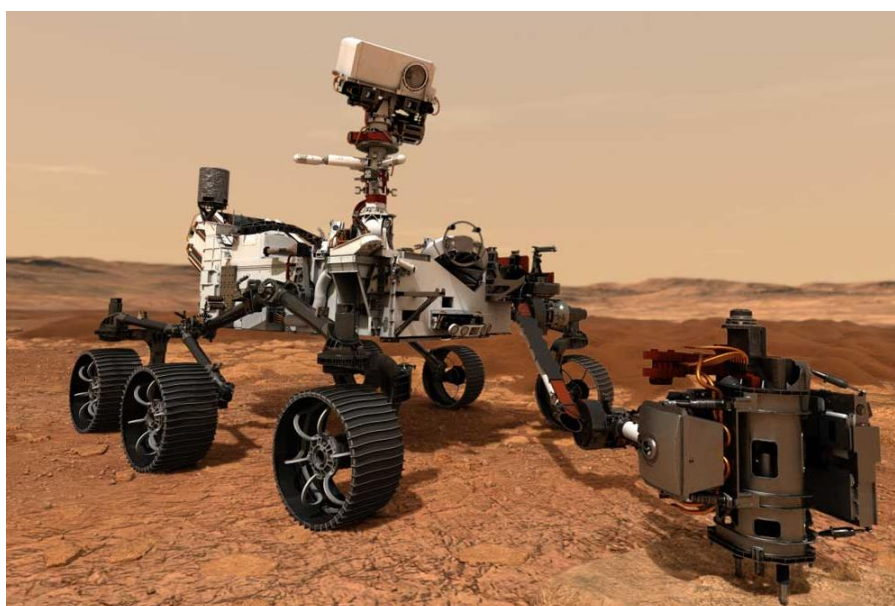
4.2 Kartlegging av sammenlignbare inspeksjonsroboter

I dette underkapittelet vil det presenteres ulike typer roboter som er utviklet for inspeksjons- og deteksjonsarbeid.

Mobile roboter, både autonome og fjernstyrte, har i flere år vært i bruk til ulike arbeidsoppgaver innen inspeksjon, deteksjon og vedlikehold. Det finnes roboter som er generelle og roboter som er utviklet for mer spesifikke formål. Generelt er de fleste mobile robotene utstyrt med enkle kamera- og visualiseringssystemer som gir operatøren muligheten til å kontrollere og styre robotens fremdrift [27]. Slike systemer kan brukes til å inspisere og visualisere områder og utstyr. I senere tid er det utviklet flere komplekse mobile roboter. Disse robotene er utstyrt med en dynamisk robotarm som gjør det mulig å utføre enkle operasjoner [23]. Det kan være alt fra

å flytte objekter, til å skru på kraner eller åpne dører for å nevne noen. Dette viser at teknologien er i stadig utvikling.

Langt fremme i den teknologiske utviklingen av mobile roboter er NASA og deres Mars Exploration Program. Mars Science Laboratory har utviklet Mars Curiosity Rover [28] som er vist i figur 4.3. Dette er en mobil robot utstyrt med en rekke sensorer og instrumenter som benyttes til å undersøke miljøet på planeten Mars. Utstyret er kameraer, spektrometre, strålingsdetektorer og andre sensorer som gjør målinger av miljøet og atmosfæren. Hovedmålet med Mars ekspedisjonen er delt inn i fire vitenskapelige områder: undersøke om det har vært liv, karakterisere klimaet og geologien, utrede om Mars er tilgjengelig for mennesker.



Figur 4.3: NASAs fjernstyrte robot, Mars Curiosity Rover [28].

Sensabot [29, 30] er illustrert i figur 4.4 og var en av de første sertifiserte mobile inspeksjonsroboten til bruk i eksplosive miljøer. Den er utviklet ved Carnegie Mellon University (CMU) på oppdrag fra Shell. Roboten er utviklet for å inspisere og overvåke industrielle fasiliteter innen olje- og gassindustrien. Dette er farlige arbeidsmiljøer for ansatte.

For å kunne gjennomføre inspeksjons- og overvåkningsarbeid må roboten kunne bevege og forflytte seg i og mellom, de samme områdene som de ansatte skulle ha gjort.

Utover de grunnleggende sensorene som laserskannere, ulike sensorer til å måle temperaturer, gasskonsentrasjon og vibrasjoner er Sensabot utstyrt med et 3D kamera som sørger for at operatøren kan styre roboten ved bruk av joystick. Sensabot er også utstyrt med en bevegelig robotarm som utvider inspeksjonsområdet.



Figur 4.4: Sensabot, inspeksjonsrobot til overvåking av industrianlegg [29].

I de senere årene er det utviklet alternative mobile roboter til Sensabot. Taurob Tracker [31], vist i figur 4.5, er hovedsakelig utviklet for å bistå brannmenn i farlige oppdrag. Sammenliknet med Sensabot er Taurob Tracker utstyrt med belter istedenfor hjul. Beltene gir en bedre fremkommelighet og manøvrering. Beltene sørger også for at roboten kan klatre i trapper og stigninger opp mot 40° helning. Utover dette er roboten utstyrt med antenner, gassensor og ulike kameraer. Robotarmen er utstyrt med termografisk kamera, men kan også utstyres med andre nødvendige verktøy som klo eller vindusknuser.



Figur 4.5: Taurob Tracker, utviklet for brannvesenet[31].

SMP Robotics S3 [32], illustrert i figur 4.6, er en mobil robot som brukes til å utføre regelmessige temperaturinspeksjoner på strømmettet og i transformatorstasjoner. Disse er med på å forhindre ulykker fordi roboten har overtatt den manuelle oppgaven til serviceingeniøren. S3 er utstyrt med et autonomt styresystem slik at den kan navigere rundt på kraftstasjonen og utføre inspeksjoner ved hjelp av et termografisk kamera. Oppdages unormale temperaturer vil det sendes det et alarmsignal og bilde til operatøren.



Figur 4.6: SMP Robotics' S3, termisk inspeksjon av transformatorstasjoner og strømnett [32].

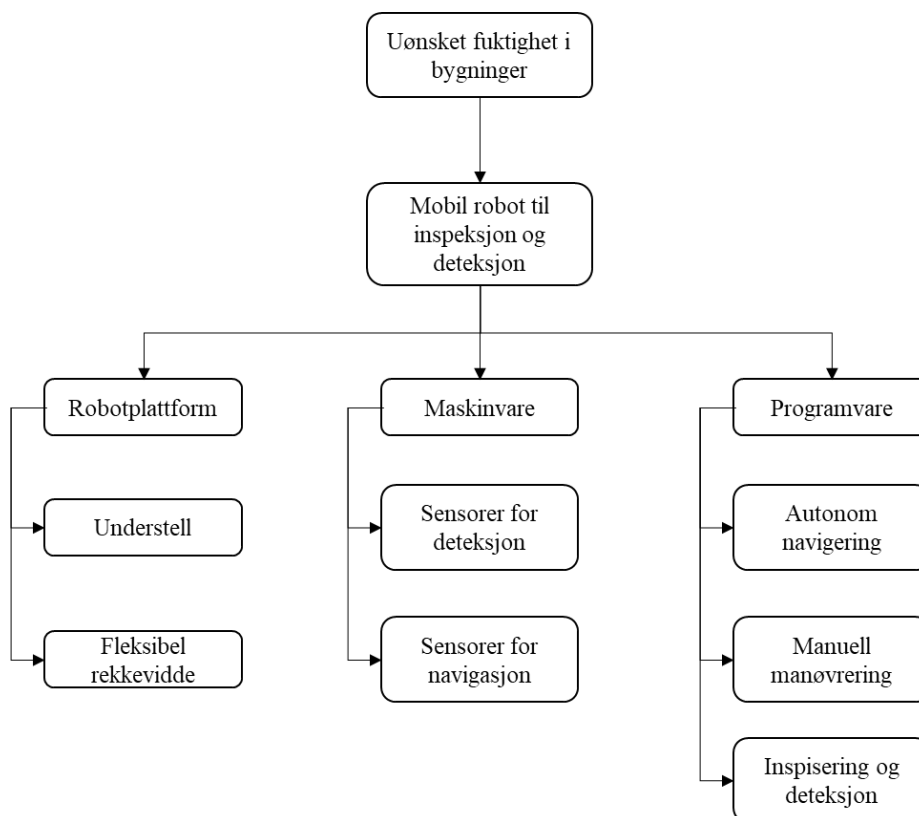
Boston Dynamics er et av de ledende selskapene innen utvikling av mobile roboter. Robothunden Spot [33], presentert i figur 4.7, er Bostons første sivile kommersielle produkt og ble nylig introdusert som Aker BPs nye inspeksjonsmedarbeider på Skarv-plattformen [34]. Fordelen med Spot er at de fire bene gjør at den kan bevege seg i trapper. Videre er Spot utstyrt med et 3D stereokamera som sørger for at den kan bevege seg rundt hindringer. Utover de grunnleggende funksjonene som andre robotene også har, kan Spot påmonteres tilleggsutstyr med en vekt opptil 14 kg. Et av tilleggsutstyrene er en robotarm, med 6 frihetsgrader. Robotarmen gjør Spot ytterligere fleksibel fordi den da kan åpne dører, plukke opp og flytte på objekter med mer [35].



Figur 4.7: Aker BPs nye medarbeider, roboten Spot [33].

4.3 Betraktninger rundt prosjektets inspeksjonsrobot

Formålet til roboten er inspeksjon og deteksjon av fuktighet i bygningsmassen. Roboten skal supplere inspektører og fagfolk i områder hvor det er vanskelig å komme til. Figur 4.8 presenterer en skjematisk fremstilling av funksjoner roboten må utstyres med for å løse arbeidsoppgavene.



Figur 4.8: Tidlig vurdering av egenskaper roboten må utstyres med for å utføre arbeidsoppgavene.

Roboten må bestå av en robotplattform som utgjør robotens kropp. Det er viktig at roboten er forberedt på ujevne underlag og har muligheter til å forser hindringer og objekter som kommer i veien. Det vil si at god mobilitet, fremkommelighet og manøvrering er viktige egenskaper for roboten. Begynner roboten å spinne vil det for eksempel kunne påvirke estimeringen av robotens posisjon over tid. Dette vil ha påvirkning på navigasjonen slik at roboten ikke når den ønskede posisjonen eller målet. Det betyr at valg av understell vil være av betydning for presisjonen til roboten.

Det aktuelle inspeksjonsområdet er ikke alltid i nærheten av robotkroppen. Området kan for eksempel være opp under taket eller i hjørner hvor det er vanskelig for roboten å komme til. Derfor må det vurderes hensiktsmessig utstyr slik at roboten får en optimal rekkevidde både i horisontal og vertikal retning. Eksempler kan være robotarm, rotasjonsplater og liknende.

Med en autonom robot vil man kunne øke fleksibiliteten og effektiviseringen av arbeidet som skal utføres. En autonom robot utfører arbeidsoppgaver uten eller med minimal involvering av operatøren, men til gjengjeld krever dette at roboten innehar grunnleggende egenskaper. De

grunnleggende egenskapene er for eksempel lokalisering av seg selv, planlegging av rutealternativer og navigere til målet uten å kolliderer med objekter og hindringer [36]. Til enkelte arbeidsoppgaver kan forhåndsprogrammerte rutealternativer benyttes, mens i andre tilfeller må roboten selv kartlegge omgivelsene.

Dersom roboten skal være autonom er det flere aspekter som må hensyntas. En av utfordringene er tilgjengelig areal roboten skal bevege seg i. Arbeidsrommet kan være trange passasjer som hulrom, kryperom, kanal ganger og tekniske mellometasjer. Som vi ser av figur 4.9 kan slike rom ha ujevne underlag med groper og helninger, samt forekomster av stein, sand og materialrester. Dette ujevne underlaget kan skape problemer med autonomien. Arbeidsområdene kan også ha ulike typer kabler og rør hengende fra taket som kan skape problemer for roboten. Dette er noen utfordringene som kan gjøre det krevende for roboten å navigere autonomt. Dette medfører at roboten er avhengig av å ha moderne sensorer for å utføre oppdraget.



Figur 4.9: Et typisk kryperom. Takhøyden er lav, ujevnt underlag og forekomster av stein og andre gjenstander [37].

Andre problemer som roboten kan møte på under arbeidet er at autonomien svikter, at roboten setter seg fast eller at omgivelsene ikke egner seg for autonom navigering. Det er også vanskelig å programmere roboten for alle situasjoner den kan møte på underveis i arbeidet. Direkte operatørkontroll av roboten kan gjøre roboten mer fleksibel, robust og brukervennlig. For at operatøren skal kunne fjernstyre roboten er det behov for en visuell kommunikasjon mellom operatøren og roboten. Operatøren må kunne se det samme som roboten. Roboten må også kunne løse den visuelle utfordringen i mørklagte rom.

Dersom noe skjer under arbeidsprosessen, er det en fordel å ha implementert en løsning som kan få roboten raskt og sikkert ut. En slik løsning bør også kunne benyttes når roboten må heises ned til arbeidsområdet.

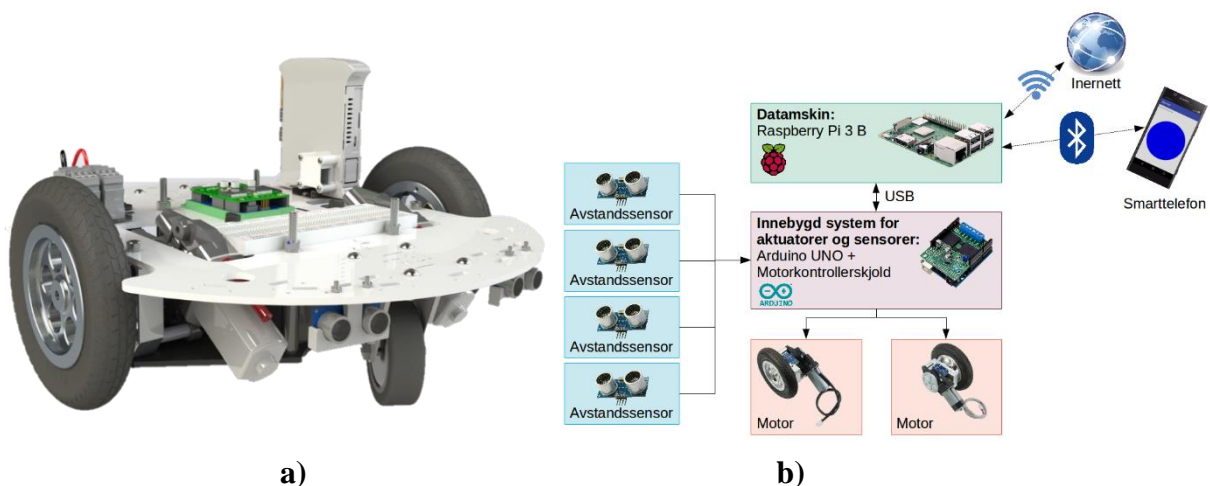
For at roboten skal kunne detektere og verifisere fuktighet må den utstyres med teknologi som kan utføre ønskede arbeidsoppgaver. Parallelt med inspiseringen må dataene tolkes og sendes til operatøren. Analysen må kunne gi roboten signaler eller operatøren beskjed om den skal gå videre med en applikasjon som kan verifisere observasjonen.

5 TEKNOLOGI- OG TEORIUTREDELSE

For å opparbeide en grunnleggende kunnskapsbase vil dette kapittelet redegjøre for teknologi og underliggende teori som er relevant for utviklingen av inspeksjonsroboten.

5.1 IMRTBot

Den mobile roboten som skal benyttes til å utvikle den kommersielle robotens programvare er en IMRTBot, vist i figur 5.1a. Den benyttes daglig i forbindelse med robotikkstudiet ved fakultetet for realfag og teknologi ved NMBU. Motorene og girene er hentet fra en Arlo Robotic Platform, mens de resterende komponentene er utviklet og valgt av Lars Grimstad. Figur 5.1b viser en skjematisk fremstilling av robotens maskinvare ved prosjektstart. IMRTBot er en allsidig, robust og solid robotplattform som egner seg til R&D, eksperimentering og utdanning. Den er satt sammen av enkle, tilgjengelige og prisgunstige komponenter på en slik måte at operatøren enkelt kan utstyre den med ønsket tilleggsutstyr.



Figur 5.1: a) IMRTBot, b) Skjematisk fremstilling av robotens maskinvare, sensorer-, aktuatorer- og styringsystem ved prosjektstart [38].

Figur 5.1b illustrerer oppsettet til IMRTBot. Den er utstyrt med avstandssensorer som er koblet til et Arduino UNO-brett og med et påsatt motorkontrollskjold. Arduinoen behandler sensordataene ved å hente ut den viktigste informasjonen, for deretter å sende kontrollkommandoer til drivsystemet. Drivsystemet består av to 12V DC motorer. Videre er arduinoen koblet til datamaskinen, Raspberry Pi 3 B, gjennom en USB. Raspberry Pi fungerer som styringsenhet, den kobler roboten mot internett og andre tilkoblingsmuligheter som for eksempel bluetooth [39].

IMRTBot er et naturlig valg som utviklingsplattform til robotens og applikasjonenes programvare fordi den benyttes og er tilgjengelig på NMBU. I markedet finnes det også andre alternativer, som for eksempel:

- TurtleBot3
- LeoRover
- ROSBot 2.0 fra Husarion
- Roboter fra Clearpath Robotics
- AmigoBot

Ytterligere spesifikasjoner av IMRTBot finnes på NMBU Robotics GitHub [38].

5.2 Sensorteknologi

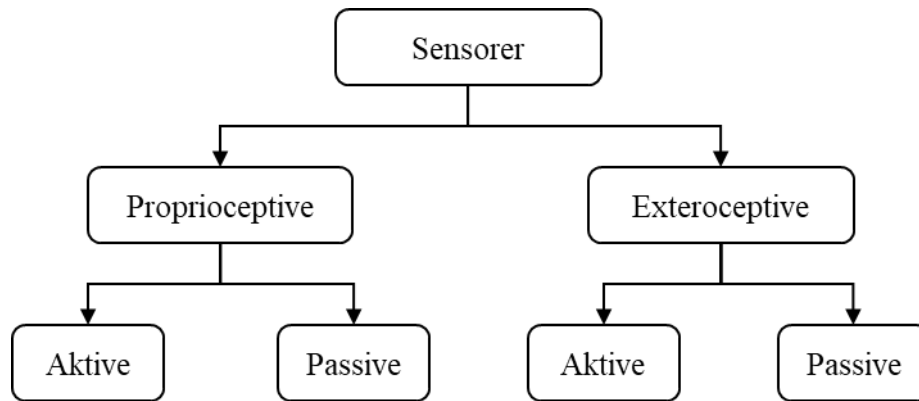
For at roboten skal kunne utføre arbeidsoppgavene, er det avgjørende å tilegne seg informasjon og kunnskap om arbeidsmiljøet. Ved å utstyre roboten med ulike sensorer og kameraer, kan man samle inn den nødvendige data om omgivelsene og inspeksjonsområdene. Sensorene er robotens øyne og øre, uten dem vil ikke roboten kunne gjennomføre sine arbeidsoppgaver [40, 41].

De neste underkapitlene tar for seg sensorer som den mobile roboten må utstyres med for å kunne observere og detektere fuktighet, men også navigere autonomt.

5.2.1 Sensorklassifisering

Basert på funksjonalitet blir sensorene ofte klassifisert i to grupper, ut ifra om de måler interne eller eksterne variabler i systemet. Dette er illustrert i figur 5.2. Videre klassifiseres de interne og eksterne sensorene inn etter om de bare mottar eller også sender signaler [40].

- *Proprioceptive sensorer* måler interne parametere i et system. Det kan være målinger av motorhastighet som gjøres ved å telle antall rotasjoner på akslingen, måling av batterispenningen eller vinkelen på en robotarm. Aktuelle sensorer kan da være akselerometer, gyroskop, kompass, encodere eller temperaturmålere.
- *Exteroseptive sensorer* måler de eksterne parameterne. For å gi viktig informasjon om omgivelsene kan man ta i bruk avstandsmålinger eller lysintensiteter. Disse dataene må roboten analysere for deretter å trekke ut den nødvendige informasjonen. Til dette arbeidet kan man bruke kamera, laserskanner, radar eller sonar.
- *Passive sensorer* måler absorbert energi som omgivelsene sender ut. Et kamera vil fange opp lyset som reflekteres fra objektene. Fuktighetssensor, temperatursensor og mikrofon er andre eksempler.
- *Aktive sensorer* sender energi ut til omgivelsene og baserer sine målinger på energien som returnerer tilbake til sensoren. Disse sensorene har derfor ofte en bedre ytelse siden de har en kontrollert interaksjon med miljøet og omgivelsene rundt. Eksempler på aktive sensorer er ultrasoniske sensorer og laserskannere.



Figur 5.2: Klassifisering av sensorer [40].

5.2.2 Sensorer for navigasjon

For at roboten skal kunne kartlegge og navigere i arbeidsområdet må den utstyres med egnede sensorer. Det vil her bli presentert noen relevante sensorer som roboten kan utstyres med.

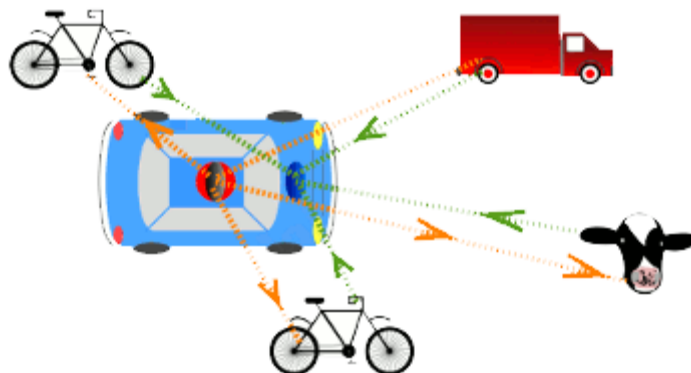
Laserskanner

Laserskannere, bedre kjent som LiDAR (Light Detection and Ranging) er aktive exteroceptive sensorer som måler avstanden til et objekt, relativt til robotens posisjon. Det er opp mot 150 000 laserstrålepulser som emitteres hvert sekund fra skanneren og det er disse laserstrålene som treffer overflatene i omgivelsene rundt roboten. Disse strålene reflekteres tilbake til sensorens mottaker [42]. Basert på Time-of-Flight (ToF) metoden vil avstanden beregnes fra roboten og frem til objektet [40]. Avstanden beregnes ved hjelp av tiden det tar fra laserpulsen sendes, til sensoren registrerer refleksjonen igjen. Følgende formel benyttes,

$$d_{ToF} = \frac{c \cdot t_{ToF}}{2} \quad 5.4$$

hvor d er distansen frem til objektet, c er lysets hastighet ($c \approx 343 \text{ m/s}$), og t_{ToF} er tiden strålen benytter tur-retur.

Laserskannere sender kontinuerlige pulser med laserstråler ut mot omgivelsene som illustrert i figur 5.3. Dette medfører at avstandsmålingene kan settes sammen i et felles koordinatsystem og en punktsky kan genereres. Dette vil da danne et «bilde» av robotens omgivelser. Har roboten behov for et kart over området, kan punktskyene settes sammen til en kartmodell. Avhengig av laserskanneren kan det genereres en 2D eller 3D modell. Når det gjelder synsfeltet til lasterskannerne, kan dette variere fra punktmåling til 360 graders. Man må derfor finne laserskanneren som passer til formålet. Den største fordelen med laserskannere er at de genererer robuste målinger både i sollys og i mørket. Årsaken til dette er den aktive lyssensoren. Laserskannere blir derimot mindre effektive i regnvær og områder med mye røyk eller tåke, fordi laserstrålepulsene blir brutt på veien [43].



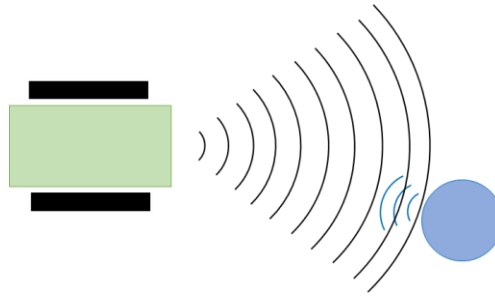
Figur 5.3: Grunnleggende prinsipp for laserskannere. Basert på den reflekterte laserstrålen kan avstanden frem til objektet estimeres [44].

Ultrasonisk sensor

Ultrasonisk sensor [40] benyttes for å estimere avstanden fra roboten og frem til et objekt. Avstanden beregnes med samme metode og formel som for laserskannerne, men lysets hastighet (c) i formel 5.4 er byttet ut med lydens hastighet i luft. I stedet for laserstråler, emitter ultrasoniske sensorer lyd med en frekvens på over 20 000 Hz [24].

Materialets akustiske egenskaper har stor innvirkning på sensorens registreringsevne [40]. Siden ultrasoniske sensorer sender ut lydbølger er sensordataen robust mot både lysrefleksjoner fra objekter, ulike lysintensiteter i omgivelsene og endringer i farger, men også faste materialer som tre og metall. Lydbølger er derimot sensitive mot stoff og tekstiler siden disse materialene absorberer lyd. Dette betyr at sensoren vil motta redusert mengde lydbølger. En annen negativ side med ultrasoniske sensorer er at lyden har en lavere hastighet enn lys. Dette medfører at avstandsberegningene blir tregere enn ved laserskannere. Når lydbølgene blir emittert fra sensorens høyttaler, vil de ved hjelp av tiden bre seg utover i omgivelsene. Dette kan sammenliknes med vannets oppførsel når gjenstander treffer vannoverflaten. Dette er illustrert i figur 5.4.

Fordelen med lydbølgenes vandring er at de vil dekke et større område, enn det laserstrålene vil gjøre. Men, til gjengjeld vil det bli vanskeligere å detektere den spesifikke posisjonen objektet befinner seg i fordi distansen kan bli noe unøyaktig [24]. Basert på påstandene over så vil nok ikke ultrasoniske sensorer alene være optimale for en sikker navigering av roboten. Benyttes flere ultrasoniske sensorer sammen med en laserskanner vil dette kunne gi et bedre resultat, siden de kan utfylle hverandres styrker og svakheter. Fordelen til laserskanneren er at den kan generere et kart over området, og de ultrasoniske sensorene observerer hindringer på nært hold, slik at kollisjoner kan unngås.



Figur 5.4: Prinsippet ved ultrasonisk sensor. Lydbølger emitteres fra sensoren og reflekteres når de treffer et objekt. Vandring av lydbølger over tid benyttes til å estimere avstanden fra sensoren og frem til objektet.

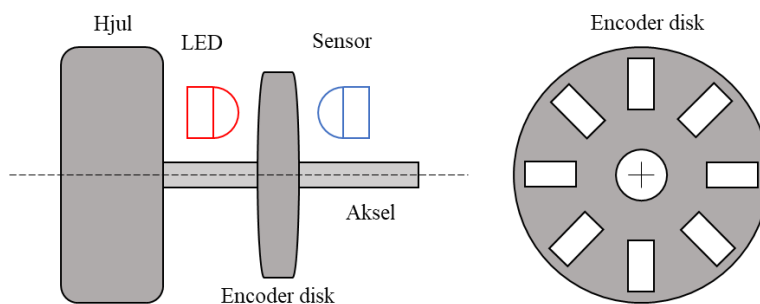
Encodere

Encodere er elektromekaniske enheter som måler antall omdreininger et roterende objekt roterer om robotens aksler eller motorer. Dataen sendes til mikrokontrolleren i form av elektroniske pulser. Kontrollsystemet kan da beregne robotens tilbakelagte strekning, rotasjonshastighet og vinkelen roboten beveger seg i forhold til et gitt koordinatsystem. Dette er viktig informasjon for den autonome navigeringen og lokaliseringen av roboten. Det finnes flere ulike encodere, optiske-, mekaniske- og magnetiske for å nevne noen [45].

Med encodere kan man enkelt estimere robotens tilbakelagte strekning og retning, fordi encoderne måler hvor langt hjulene har rullet. Robotens retning beregnes ut ifra forskjellen på hvor langt de forskjellige hjulene har rullet i forhold til hverandre. Derfor er disse enhetene en viktig del av robotens odometri. Drivhjulets radius, r , er gitt av produsenten og omkretsen er gitt av $2\pi r$. Antall rotasjoner, n , måles av encoderne. Den tilbakelagte distansen til hvert hjul over et tidsintervaller er gitt ved følgende formel [24],

$$d_{encoder} = 2\pi r n \quad 5.5$$

Prinsippene er enkle. En optiske encoders hovedkomponenter er et lys, gjerne en LED, en lyssensor og en encoderdisk med hull som illustrert i figur 5.5. Når hullene i disken er på samme nivå som LEDen, detekterer lyssensoren lys og genererer et signal. Dette signalet sendes til mikrokontrolleren og videre til robotens styringskontroll. For hvert signal som genereres har hjulet rotert en gitt vinkel.



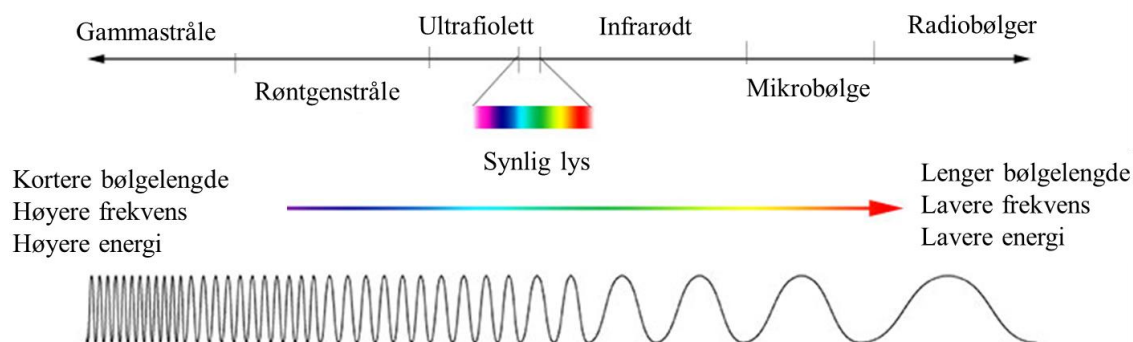
Figur 5.5: En optisk hjulencoder benyttes til å konvertere roterende bevegelser til digitale signaler. Lyset fra LEDen fanges opp av sensoren i de transparente områdene på disken. Illustrasjon hentet fra [24].

5.2.3 Sensorer for deteksjon av fuktighet

Fuktighet forårsaket av lekkasjer og innsig av vann kan ofte være vanskelig å observere og detektere, og over tid kan denne uoppdagede fuktigheten føre til store skader. Ved å ta i bruk moderne utstyr og teknologi vil det være mulig å oppdage og detektere innsig av vann og fuktighet på et tidligere stadium slik at man kan forhindre råte- og soppskader på bygningsmassen. I dette underkapittelet vil det presenteres sensorer som kan utføre inspeksjons- og deteksjonsarbeid.

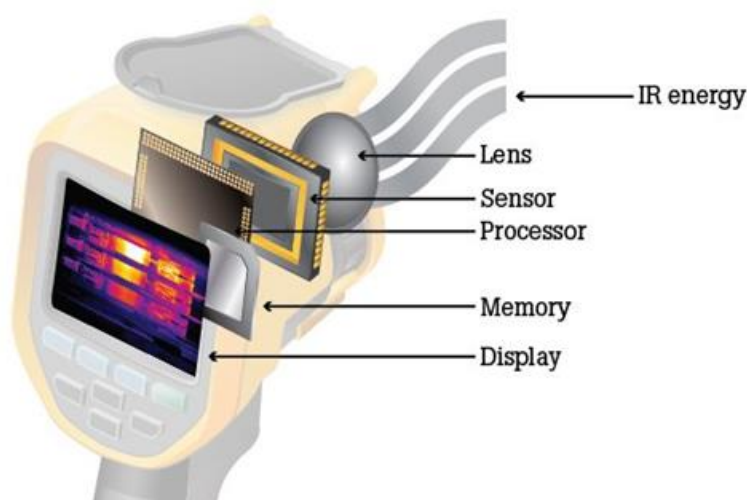
Infrarødt kamera

Alle objekter utstråler energi i form av infrarøde stråler. Dette er elektromagnetisk stråling som emitterer fra objekter når atomer absorberer og frigjør energi. Styrken på denne strålingen er avhengig av objektets temperatur. Jo høyere temperatur, jo sterkere stråling. De elektromagnetiske strålene befinner seg mellom synlig lys og mikrobølger på det elektromagnetiske spekteret i figur 5.6. Infrarøde stråler er en av tre måter å transportere varme på. Mennesker eksponeres daglig for infrarøde stråler og kjenner dem på kroppen gjennom varme [46].



Figur 5.6: Her presenteres det elektromagnetiske spekteret som beskriver sammenhengen mellom bølgelengde, frekvens og energi [47].

Infrarøde kameraer (IR-kamera) er utviklet for å detektere og visualisere infrarøde stråler. Kameralinsen fanger opp strålene og konverterer strålene til elektroniske signaler i sensoren og prosessoren som illustrert i figur 5.7. De elektriske signalene brukes til å generere termografiske bilder der temperaturkombinasjonene avhenger av mengden energi objektet utstråler [48]. Temperaturforskjellene beskrives gjennom fargekoder. De røde, oransje og gule fargene indikerer varmere områder, mens de mørke fargene som svart, blått og lilla indikerer kjøligere temperaturområder.

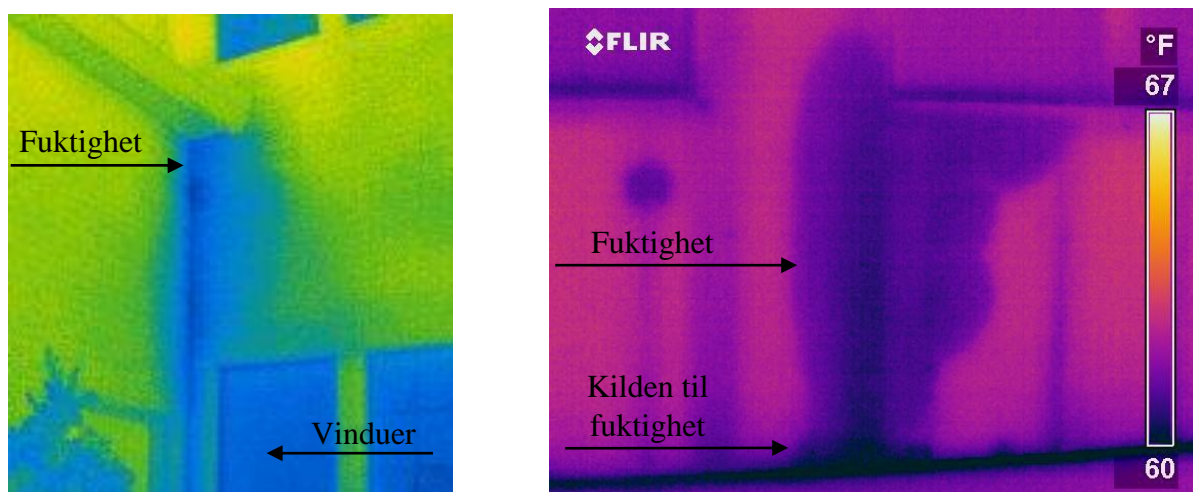


Figur 5.7: Grunnleggende prinsipper ved et IR-kamera. Linsen tar imot de termiske strålene som blir analysert og prosessert i sensoren og prosessoren. På dette kameraet lagres de termiske bildene på et minnekort før de presenteres på en skjerm [49].

Ved å se på temperaturforskjellene i bygningsmassen vil det være mulig for inspeksjonsroboten å benytte denne teknologien til å detektere potensielle fuktige områder. Ifølge boken *Infrared Thermography for Building Moisture* [50] er fuktigheten i materialet direkte relatert til endringene i materialets overflatetemperatur, grunnet tre fysiske prinsipper:

- Fordampningen av materialets overflate er en endotermisk reaksjon som vil si at fuktigheten absorberer energi fra omgivelsene. Denne reaksjonen forårsaker en reduksjon av overflatetemperaturen.
- Fuktige materiale har en lavere termisk motstand som vil si at varmen beveger seg lettere gjennom det fuktige materialet enn det tørre materialer.
- Fuktig materiale har en høyere varmekapasitet enn tørre materialer. Det vil si at overflatetemperaturen i et fuktig materiale reagerer saktere med lufttemperaturen enn ett tørt materiale. Dette betyr at et fuktig materiale lagrer mer varme enn tørre materialer og de vil avkjøles saktere.

Fuktigheten vil i de fleste tilfeller ha en lavere temperatur enn omgivelsene. Derfor vil de fuktige områdene skille seg ut på det termografiske bildet som vist i figur 5.8. IR-kameraer er ikke avhengig av lys for å generere termografiske bilder. Det betyr at kamerat kan benyttes i alle lysforhold. Dette er spesielt gunstig for roboten som skal inspisere i trange og mørke områder. På denne måten vil det være mulig å utføre ikke-destruktive inspeksjoner [51].



Figur 5.8: Termiske bilder som viser hvordan fuktigheten brer seg utover veggen. I bildet til venstre representerer det blå området fuktighet. Men bildet inneholder også vinduer som i første omgang kan tolkes som fuktighet. I bilde til høyre ser man tydelig hvordan fuktigheten trekker opp fra kilden i bunn av veggen. Det fuktige området er presentert som svart og mørkelilla områder [52].

Ifølge kameraprodusenten FLIR er det krevende å detektere fuktighet ved bruk av IR-kamera, men under de rette omstendighetene og med det riktige oppsettet vil det la seg gjøre. Kameraet bør ha en høy temperaturfølsomhet. Det vil si at kameraet må være sensitiv til temperaturforskjeller. Den termiske sensitiviteten bør være lav og ikke overstige anbefalte 30mK [52]. Det er viktig å poengtere at kameraet ikke verifiserer fuktighet, men observerer et potensielt fuktige område som må undersøkes videre. Årsaken er unormale temperaturforskjeller som danner unaturlige mønstre. Som vist i figur 5.8, er det de mørke områdene som indikerer et fuktig områder.

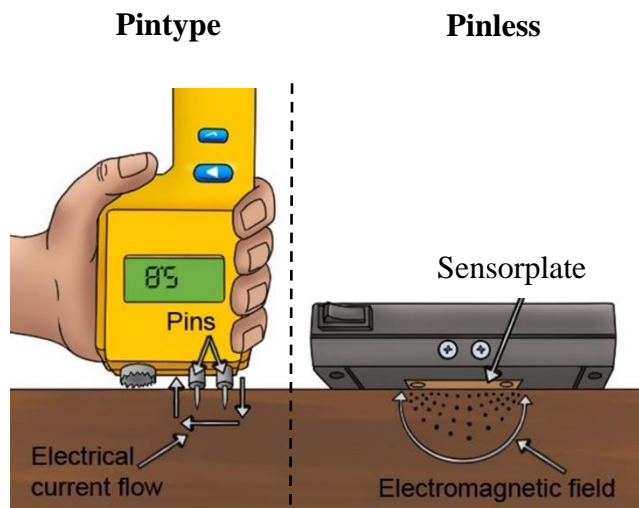
Fuktighetssensorer

En fuktighetssensor er et apparat som måler prosentvis mengde vann i bygningsmaterialet. Disse benyttes til inspeksjoner for å bekrefte eller avkrefte fukt i et aktuelt område. Det finnes to typer fuktighetsmålere, pintype og pinless som vist i figur 5.9.

Pintype fuktighetsmåler benyttes oftest når det er materialets overflate eller området rett under overflaten som skal undersøkes [53]. To metallpinner med en viss avstand, stikkes ned i materialets overflate. Deretter sendes det elektrisk strøm mellom pinnene. Fuktigheten måles basert på materialets elektriske ledningsevne. Fuktige materialer har en lavere resistans enn tørre og leder dermed elektrisk strøm bedre. Den målte resistansen mellom de to metallpinnene gjøres om til en prosent i en skala på måleapparatet [54]. Optimale fuktighetsverdier for treverk ligger mellom 5% – 12%, moderate og akseptable verdier ligger i intervallet 12% – 17%. Materialet anses som fuktig dersom måleapparatet viser verdier over 17% [55].

Pinless fuktighetsmåler benyttes oftest når fuktighet skal detekteres under materialets overflate [53]. Målerens sensorplate plasseres over det aktuelle området og emitterer elektromagnetiske

bølger ned i materialet med en bestemt frekvens. Da dannes det et elektromagnetisk felt under sensorplaten. På lik linje med pintype, påvirker også fuktigheten både frekvensen og rytmen til de elektromagnetiske bølgene. Pinless fuktighetsmåler estimerer fuktigheten ved å se på endringene i de returnerte bølgene [54].



Figur 5.9: To ulike fuktighetsmålere, pintype til venstre og pinless til høyre. Pintype sender strøm gjennom to pinner for å måle fuktigheten, mens pinless benytter endringer i frekvensen til de returnerende elektromagnetiske bølgene til å estimere fuktigheten [56].

5.2.4 Sensorytelse og pålitelighet

Når systemet baserer seg på sensorer er det viktig at man er kritisk til dataene de samler inn. Det kan fremkomme ulike måleresultater, til tross for at det er like modeller og fra samme leverandør. Det vil også være forskjeller fra modell til modell og mellom ulike leverandører, når det kommer til ytelse og pålitelighet. Som oftest er de dyrere sensormodellene mer pålitelige enn de billigere på markedet, fordi bedre komponenter benyttes. I utviklingsfasen av konseptet må det derfor prøves ut ulike modeller og leverandører, for å komme frem til et optimalt sensoroppsett.

5.3 Mikrokontroller

I robotikken benyttes mikrokontrollere til å samle inn data fra ulike sensorer, brytere og lignende. Dataen mikrokontrolleren mottar blir deretter prosessert og analysert gjennom et tilpasset dataprogram. Dataprogrammet omgjør så informasjonen til kommandosignaler som blir sendt til for eksempel motorer, servoer og lys. I dagens samfunn anvendes mikrokontrollere ikke bare i roboter, men også i mange forskjellige forbruksartikler. Eksempler på dette er biler, mikrobølgeovner og vaskemaskiner.

PCer og mikrokontrollere deler mange av de samme funksjonalitetene, men det er noen vesentlige forskjeller. Mikrokontrolleren er på størrelse med en hånd eller mindre, og fokuserer kun på en oppgave om gangen. I tillegg er mikrokontrollerenes arbeidsoppgaver små og enkle, og handler ofte om å overvåke, kontrollere eller styre bestemte prosesser i et større system [57].

Siden det er single arbeidsoppgaver som skal kontrolleres, er det ikke behov for stort minne og høy prosessorytelse slik som det er i de tradisjonelle datamaskinene som skal utføre flere arbeidsoppgaver samtidig.

Sammenlikner man innholdet i en mikrokontroller og en datamaskin ser man tydelig hvorfor mikrokontrollere ofte blir definert som små datamaskiner. Mikrokontrollere er satt sammen av:

- En CPU (Central Processing Unit) som kjører programmer og sender kommandosignaler slik at arbeidsoppgaven blir utført.
- RAM (Random Access Memory) er hovedminnet til kontrolleren som gjør at programmet kan kjøres av CPU-en.
- I likhet med en PC, kan også et mikrokontrollerbasert system utstyres med et brukergrensesnitt. I motsetning til PC-ens tastatur, mus og skjerm, er mikrokontrollerens grensesnitt ofte enkelt, eller svært applikasjonsrettet. For eksempel termostathjul eller et LCD-display på en panelovn.
- I stedet for å benytte tastatur og mus, kommuniserer mikrokontrolleren gjennom sensorer, LED og andre enheter.

Mikrokontrollere blir ofte benyttet istedenfor datamaskiner på grunn av sine mange fordeler. Årsaken til dette er:

- De er små og enklere å integrere i større systemer hvor spesifikke arbeidsoppgaver skal styres og kontrolleres.
- De er rimeligere i innkjøp på grunn av mindre minne og lavere prosessorytelse.
- De er enkle å koble til sensorer, brytere og lys, men også til skjermer.
- De har lavt energiforbruk og benyttes ofte i systemer som drives av et batteri.
- De er enkle å programmere og kan tilpasses etter behov.

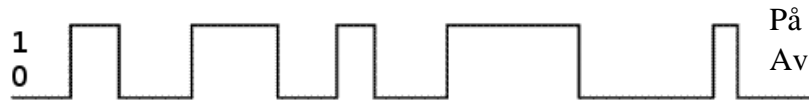
En mikrokontroller er utstyrt med input- og outputporter (I/O porter). Årsaken til dette er at mikrokontrolleren skal lese sensordata og kommunisere med andre maskiner, eller kontrollere andre elektriske enheter som motorer, skjermer og lys. Inngangsportene mottar data fra sensorer, brytere eller andre tilsvarende komponenter, mens utgangsportene brukes til å sende handlingskommandoer til robotens eksterne maskinvarer.

Det finnes to typer I/O-porter på et kretskort, analoge og digitale. De digitale portene benyttes ofte i forbindelse med brytere med av/på-funksjoner. Sensorer kobles derimot ofte til de analoge portene fordi sensordataen konverteres til diskrete verdier i et måleintervall. Måleintervallet varierer med typen mikrokontroller. Figur 5.10 illustrerer forskjellen på analoge og digitale signaler [58].

Analoge signal



Digitalt signal



Figur 5.10: *Analoge signaler, de er kontinuerlige og endrer seg både i amplitude og frekvens over tid. De tar også inn ulike verdier innenfor et måleintervall. Digitale signaler kan bare måle to verdier, enten 0 eller 1 [59].*

5.4 Programvare og datasystemer

Det er krevende å utvikle en robust og velfungerende programvare til en robot fra bunnen av. Roboten krever alt fra kinematiske modeller og kontrollere, til navigasjon og objektgjenkjenning. Et vanlig problem var at utviklere brukte mye tid og energi på å implementere grunnleggende egenskaper på roboten og dette var med på å bremse innovasjonen innenfor fagfeltet.

Å ta i bruk et egnet programmeringsrammeverk er en løsning på innovasjonsproblemet. Dette er en plattform bestående av verktøy, biblioteker og konvensjoner som gjør det enklere å utvikle komplekse robotsystemer. Disse gjør at utvikleren kan fokusere på videreutvikling av systemer fra et stadig høyere nivå. Flere av rammeverkene er open-source som gjør at utvikleren kan utvikle og dele sin programvare med andre. Dette medfører en mer effektiv utvikling av robotene. I dag finnes det flere utviklingsplattformer, Mobile Robot Programming Toolkit, Robot Operating System, Orocos (Open Robot Control System) og Microsoft Robotics Developer Studios for å nevne noen.

5.4.1 Robot Operating System

Robot Operating System (ROS) er et open-source rammeverk for programmering og utvikling av programvare til roboter og tilhørende applikasjoner, der hovedvisjonen er gjenbruk og samarbeid av kode [60]. Rammeverket er mye brukt i forbindelse med utdanning og R&D, men også til utvikling av kommersielle roboter. ROS består også av verktøy, biblioteker og konvensjoner som gjør det enklere å utvikle programvare til roboter. Programvaren er kompatibel med Linux og støtter flere programmeringsspråk, hvor Python og C++ er mest brukt [61] [62].

Researchprosjektene Stanford Artificial Intelligence (STAIR) og Personal Robotics (PR) ved Stanford University var på midten av 2000-tallet starten på ROS. Målet var å utvikle et dynamisk og fleksibelt robotrammeverk som kunne benyttes på alle typer roboter, alt fra mobile roboter, til droner og båter [63]. I 2006 ble Willow Garage grunnlagt, en researchlab og

teknologiinkubator som videreutviklet ROS-rammeverket til Personal Robotics Project [64]. I dag er ROS et økosystem av utviklere som sammen bidrar til utvikling av programvare til roboter.

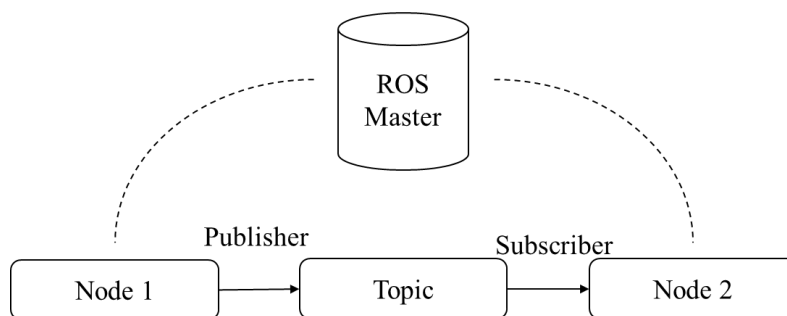


Figur 5.11: ROS logo [61].

I utgangspunktet fungerer ROS som en bro mellom to prosesser eller programmer, slik at de kan kommunisere med hverandre. En robot er satt sammen av mange prosesser og hver arbeidsoppgave kalles *node*. Hver node har et individuelt navn, slik at det er mulig å skille dem fra hverandre. En node kan kontrollere en motor, mens en annen node kan kontrollere robotens navigasjon eller ha kontroll over en sensor.

Et kontrollsystem er satt sammen av flere noder som sørger for at roboten utfører de arbeidsoppgavene den er programmert til å gjøre. Det er flere fordeler med å ordne programvaren i noder. Dersom en node ikke fungerer, vil roboten fortsatt kunne utføre oppgaver, men det avhenger naturligvis av hvor defekt den ene noden er. Samtidig vil det være enklere å finne feil når kontrollsystemet er satt opp med noder. Nodesystemet reduserer programvarens kompleksitet, siden hver node håndterer en arbeidsoppgave.

Noder som sender data kalles *publisher nodes*, og nodene som mottar data kalles *subscriber nodes*. Dataen sendes og mottas som *meldinger* (messages) via ulike *topics* som illustrert i figur 5.12. Topics er systemets kommunikasjonskanaler siden de mottar og sender meldinger med bestemte datatyper. En node publiserer sensordata til eller gjennom en topic. Har andre noder behov for denne datatypen i sin prosess, må noden abonnere på denne bestemte topicen.

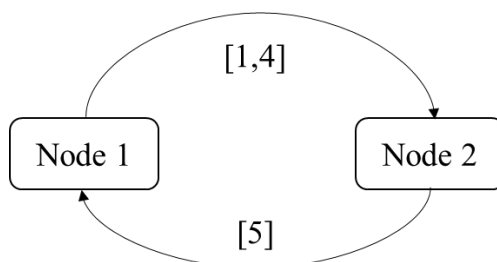


Figur 5.12: Publisher og subscriber i ROS. Node 1 publiserer en melding til topicen, hvor node 2 abonnerer på denne informasjonen.

I et system kan derfor flere noder abonnere og publisere til en topic, og en node kan abonnere og publisere til flere topics. Man kan derfor se på det som om nodene ikke kjenner hverandres eksistens, da de kun ser på navnet og datastrukturen til topicen. Ved å kommunisere på denne

måte kreves det mindre av systemet, siden det bare er den nødvendige informasjonen som sendes og mottas. Figur 5.12 illustrerer kommunikasjonsprosessen.

ROS topic fungerer på mange måter som en enveis- eller mange-til-mange kommunikasjon. Noen applikasjoner har behov for en toveiskommunikasjon. Toveiskommunikasjonen kalles ROS *service*. Nodene tar da direkte kontakt med hverandre, for å løse problemet. Problemet kan være så enkelt at en node sender en rekke tall og ber den andre noden returnere summen. Dette er vist i figur 5.13. Mer informasjon om programvare og maskinvare i ROS finnes på ROS Wiki [62].



Figur 5.13: ROS Service, en toveiskommunikasjon mellom noder. Node 1 etterspør summen av 1+4 og node 2 responderer.

ROS inneholder også nødvendige og verdifulle verktøy for simulering og visualisering av roboten. Ved å teste programvaren i en simulator gjennom utviklingsfasen, er dette tids- og kostnadsbesparende. Dette gir også en økt produktsikkerhet. Gazebo er en open-source robotsimulator som enkelt integreres med ROS. Simulatoren emulerer på reelle fysiske miljøer som kan modelleres i 3D. På denne måten kan utvikleren få testet robotalgoritmer på en effektivt og nøyaktig måte. I praksis innebærer dette å etablere hindringer som roboten kan møte i det virkelige arbeidsmiljøet. Gazebo bygger på at utvikleren skal kunne kartlegge hvordan de fysiske hindringene påvirker robotens funksjon.

Andre fordeler med Gazebo er at den kan brukes til å designe roboter samt gjennomføre trening av kunstige intelligente systemer. Robotalgoritmen som blir simulert i Gazebo kan også utstyres med de tilgjengelige sensorene fra Gazebos bibliotek. Dette gir utvikleren muligheten til å teste hele systemet. Sensordataen som blir generert i Gazebo, er de samme sensordataen som blir publisert fra robotens datamaskin. Når roboten skal simuleres, vil dette gjennomføres uten en direkte tilgang til den reelle roboten. Siden ROS har et grensesnitt til Gazebo medfører dette at roboten man ønsker simulert, kan styres ved å sende meldinger fra andre ROS noder, til Gazebo noden. Det vil si at når man simulerer en robot i Gazebo, vil resultatet være som om vi har en «digital tvilling». Gazebo har også utviklet et bibliotek med flere ulike miljøer og områder, hvor man også har mulighet til å importere en egen CAD-modell av det ønskede miljøet [65, 66].

Et annet visualiseringsprogram som er mye brukt er RViz. RViz er et 3D visualiseringsverktøy for ROS hvor utvikleren kan visualisere robotens funksjoner, sensordata fra virkelige sensorer eller simulerte sensordata fra Gazebo. Data fra 3D-sensorer presenteres som punktskyer eller dybdebilder, mens 2D sensordata som kommer fra webkameraer eller RGB kameraer vil visualiseres som billedata.

Fordelen med et simulerings- og visualiseringsprogram som Gazebo og RViz er at utvikleren får muligheten til å undersøke hvordan roboten registrer og reagerer på de tilbakemeldingene

den mottar fra sensordataene. Utvikleren får også muligheten til å feilsøke på roboten, basert på de gjeldende input dataene [67].

5.4.2 Fordeler ved bruk av ROS

Hovedmålet til ROS er å være den foretrukne plattformen for utvikling av roboter, hvor hensikten er at man benytter seg av den samme koden som mange andre har gjort tidligere. Det er flere gode grunner til å benytte ROS i utviklingen av roboter. Utviklingstiden til produktet vil reduseres fordi man kan ta i bruk den eksisterende programvaren og fordi ROS støtter gjenbruk av kode. Dette medfører at utvikleren kan laste ned den nødvendige programvaren i pakker som trengs til prosjektet, og bruke den resterende tiden på videreutvikling av programvaren det er behov for. Å laste ned kode og programvare i pakker gjør ROS modulær.

En annen grunn til å benytte seg av ROS er fellesskapet det gir for brukerne som sammen bidrar til at ROS hele tiden er up-to-date. Dette er også grunnen til at produsenter av roboter og sensorer selger utstyr tilpasset ROS [68]. ROS kan uttrykkes gjennom en matematisk likning som vist i figur 5.14.



Figur 5.14: Likningen beskriver ROS. ROS er en plattform for struktur, verktøy, evne til gjennomføring og et økosystem [61].

5.5 Navigering av mobile roboter

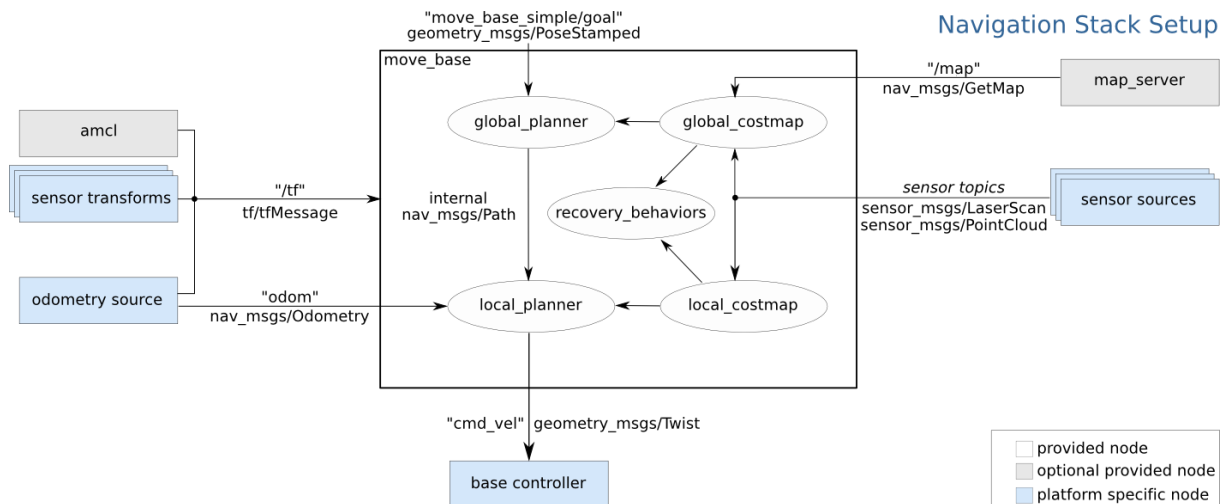
Navigasjon og manøvrering er grunnleggende egenskaper hos mobile roboter. For å kunne manøvrere mellom to posisjoner er roboten avhenger av flere faktorer. Roboten må kjenne sin egen posisjon, men også destinasjonen. En brukbar rute planlegges ved å gi roboten et kart over omgivelsene den kan manøvrere ut ifra.

Som nevnt tidligere har ROS mange tilgjengelige programvarepakker som kan lastes ned. En av disse er ROS Navigation Stack, også kalt *nav stack*. Dette er et komplekst system som gjør det mulig for roboten å navigere på egenhånd. Dette underkapittelet vil gi et innblikk over ROS nav stack, med underliggende teori.

ROS har også tilgjengelige pakker for å kunne manøvrere roboten manuelt. Et eksempel på dette er programvarepakken *joy* [69] som inneholder alle nødvendige noder og drivere slik at operatøren kan kontrollere roboten med en håndkontroller.

5.5.1 Navigering med ROS

ROS nav stack [70] er en programvarepakke i ROS som er mye brukt. Det er et komplekst navigasjonssystem som lar roboten manøvrere i ett miljø. Ved å integrere informasjonen fra robotens omgivelser, lokalisering, odometri og sensorer, vil nav stack planlegge en brukbar rute fra robotens nåværende posisjon til ønsket destinasjon. Dersom roboten møter ukjente hindringer, vil nav stack planlegge en ny rute og omdirigere roboten. Figur 5.15 illustrerer en skjematisk fremstilling av ROS nav stack [71].



Figur 5.15: Skjematisk fremstilling av ROS Navigation Stack [71].

På et overordnet nivå fungerer ROS nav stack på følgende måte [72]:

- Navigasjonmålets posisjon og orientering i koordinatsystemet blir sendt til nav stack.
- Nav stacks *global planner* (fig. 5.15) benytter et kart og informasjonen fra punktet over, til å beregne den brukbare ruten fra nåværende posisjon og til destinasjonen. Denne ruten planlegges før roboten begynner å bevege seg. Den langsiktige ruten planlegges ut ifra et *global costmap*.
- *Local planner* (fig. 5.15) manøvrerer etter den brukbare ruten som er generert i den globale planleggeren. Ved hjelp av sensordata vil den lokale ruteplanleggeren redigere ruten hvis det uoppdagede hindringer underveis.
- Når roboten nærmer seg målet, avsluttes operasjonen.

5.5.2 Occupancy Grid Map

Occupancy Grid Map [73] er et rutenett som representerer robotens arbeidsområde. Ulike sensorer samler informasjon om omgivelsene. Rutenettet blir brukt av ruteplanleggerne for å finne hensiktsmessig rute fra A til B, men også for å lokalisere roboten i et miljø. Basert på sensordataene viser rutenettet hindringer og andre objekter som kan skape utfordringer for robotens navigering. Hver celle i rutenettet har en verdi som representerer sannsynligheten for at den er opptatt. Verdier nære null indikerer at cellen mest sannsynlig er tom, og verdier opp mot 1 indikerer at cellen er opptatt.

Roboten er utstyrt med en laserskanner som fungerer som en avstandsmåler. Den gir også muligheten for å generere et occupancy grid map bestående, av en array med okkupasjonsvariablene 0'ere og 1'ere. Dette er presentert på matematisk form,

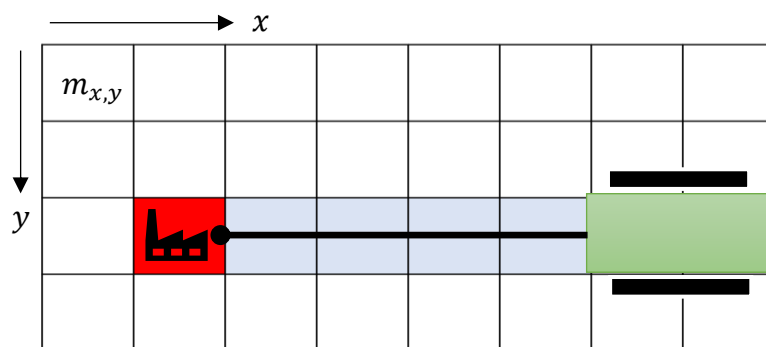
$$m_{x,y} : (\text{ledig, opptatt}) \rightarrow (0,1)$$

hvor m er kartet.

Når laseren skanner omgivelsene sender den ut laserstråler som gir tilbakemeldinger på om cellene i rutenettet, er tomme eller okkupert av et objekt. Dersom cellen er tom passerer laserstrålen rett gjennom arealet. Dersom cellen er okkupert av et objekt, vil laserstrålen treffe objektet. Figur 5.16 illustrerer et 2D occupancy grid map hvor de blå rutene indikerer tomme celler. Den røde cellen indikerer et objekt som vil si at den er okkupert. Occupancy grid mapping benytter sannsynlighetsteori sammen med sensordata, for å konkludere om roboten bør følge den planlagte ruten. Flere celler laserstrålen passerer gjennom, jo større er sannsynligheten for at den siste cellen laserstrålen treffer, er okkupert av et objekt. Sannsynligheten for at cellen er okkupert er gitt av,

$$p(m | z_{1:t}, x_{1:t})$$

der m er kartet, $z_{1:t}$ er alle sensormålingene frem til tiden t . $x_{1:t}$ er robotens posisjoner frem til nåværende posisjon og tid.



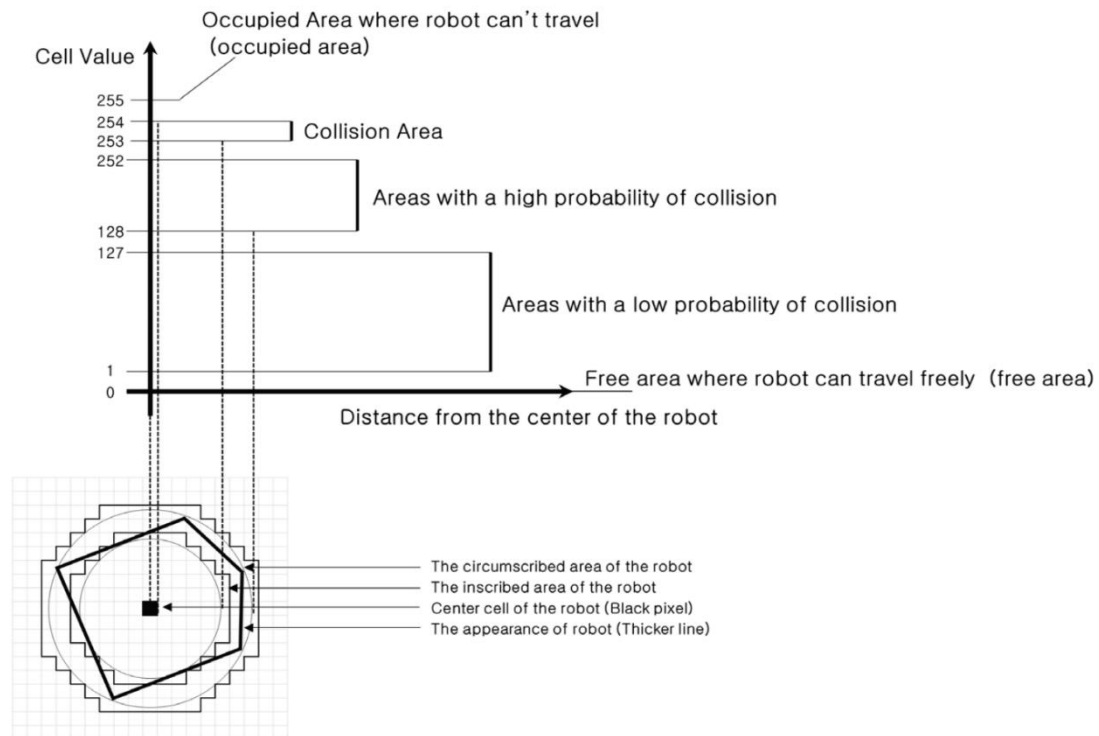
Figur 5.16: Et 2D occupancy grid map. Hvor m er kartet, og x og y er henholdsvis horisontal og vertikal retning.

5.5.3 Costmap

I figur 5.15 fremkommer det at ROS nav stack benytter et globalt og lokalt costmap for å generere rutealternativer. Et costmap inneholder informasjon om mulige kollisjonsområder roboten må ta høyde for, når den følger den optimale bevegelsesbanen i det aktuelle området. Det er rutenettverket som bestemmer om bestemte posisjoner i kartet er gode eller dårlige posisjoner for roboten å bevege seg i. Basert på sensordata og et statisk costmap, genererer ROS gjennom sin programvarepakke costmap_2d [74] et occupancy grid med kostverdier. Det betyr at et åpent område uten hindringer er en god posisjon og vil gi en lav kostverdi i occupancy grid mappet. En kollisjon med en hindring vil gi en høy kostverdi og en dårlig posisjon. Det vil si at jo nærmere roboten kommer en hindring, jo høyere blir kostverdien.

Inflation er prosessen i costmap_2d som evaluerer kostverdiene. Dette gjøres ved å lage en radius rundt roboten med ulike kostverdier som illustrert i figur 5.17. Basert på disse

kostverdiene kan ruteplanleggerne optimalisere ruten for roboten fra A til B. Basert på hva robotens sensorer oppdager, kan objekter legges til eller fjernes fra costmappet. Costmappet vil være under kontinuerlig oppdatering.



Figur 5.17: Forholdet mellom avstand til objekter og kostverdiene i costmappet [68]. Et område med lite hindringer gir lav kostverdi, og en kollisjon vil gi høy kostverdi.

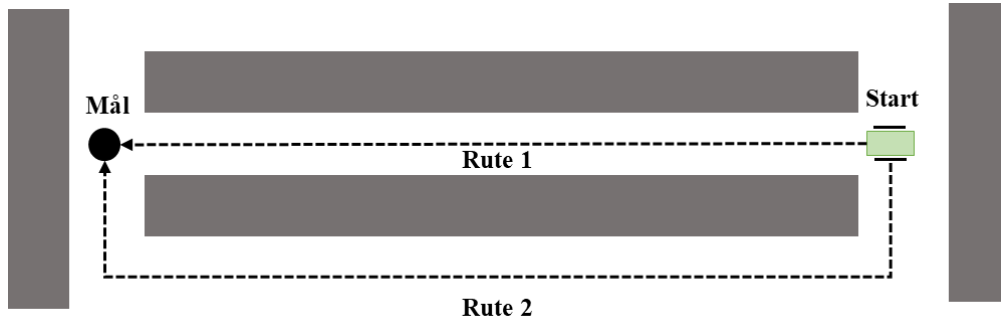
ROS nav stack benytter to typer costmaps, et globalt og et lokalt. Det globale costmappet bruker kostverdiene til de ulike pikslene i det globale kartet og rutenettet, for å finne en optimal rute til roboten. Det lokale costmappet benytter derimot sensordata til å registrere hindringer i nærhet til roboten. I dette arealet gjøres det en ruteplanlegging som har til mål å styre unna uoppdagede hindringer som ikke er registrert i det globale costmappet.

Verdiene i costmappet defineres i området fra 0 til 255. Tallverdien sier noe om risikoen roboten har for å støte på en hindring eller ikke, i det aktuelle området. Figur 5.17 beskriver betydningen av de ulike verdiområdene [68].

5.5.4 Global planlegger

ROS nav stack er også utrustet med en programvarepakke for en global ruteplanlegger. Denne baserer seg på to ruteplanleggingsalgoritmer, Dijkstra's og A*. Disse er ikke dekket i dette prosjektet. Hovedmålet er å planlegge et brukbart rutealternativ før roboten begynner å bevege seg. Ruten tar utgangspunkt i et gitt startpunkt eller robotens nåværende posisjon. Den korteste og sikreste ruten planlegges og genereres i det globale costmappet som benytter seg av et 2D occupancy rutenettverk [75].

I figur 5.18 illustreres et typisk problem som den globale ruteplanleggeren må løse. Rute 1 er kortest og smalest. Rute 2 er lengre, men mer åpen og lettere å navigere i. Man tenker som oftest at kortest er raskest, men for en robot kan det være mer hensiktsmessig å velge den lengre og enklere ruten, versus den korte ruten med større sjanser for kollisjoner.

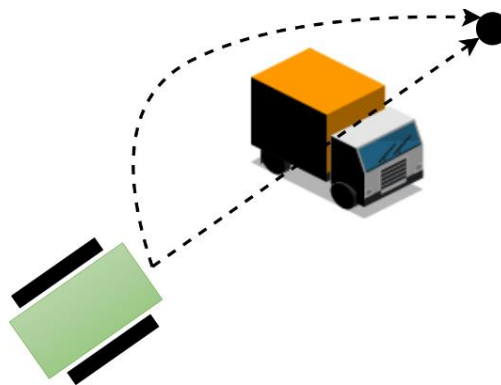


Figur 5.18: Et mulig scenario som den globale planleggeren må løse. Valg mellom to ulike ruter med forskjellig vanskelighetsgrad.

5.5.5 Lokal planlegger

I tillegg til den globale planleggeren, består ROS nav stack av en lokal planlegger. Den lokale ruteplanleggeren benytter et lokalt costmap. Dette oppdateres kontinuerlig ved hjelp av sensorer på roboten som gir informasjon om omgivelsene og hindringer. Roboten vil prøve å følge ruten fra den globale ruteplanleggeren så godt det lar seg gjøre, basert på robotens kinematikk og dynamikk. Det lokale costmappet sørger for at roboten ikke kolliderer med uventede hindringer ved å sende hastighetskommandoer til robotens aktuatorer [75]. ROS benytter to typer lokale ruteplanleggere, Trajectory Rollout og Dynamic Window Approach. Disse er utenfor prosjektets rammer.

Figur 5.19 illustrerer et problem den lokale ruteplanleggeren kan møte. Roboten følger den globale ruteplanleggerens rutealternativ og plutselig dukker det opp et dynamisk hinder som ikke var lokalisert i det globale kartet. Data fra robotens sensorer benyttes til å oppdatere det lokale costmappet som igjen sørger for at den lokale ruteplanleggeren rekker å om dirigere roboten.



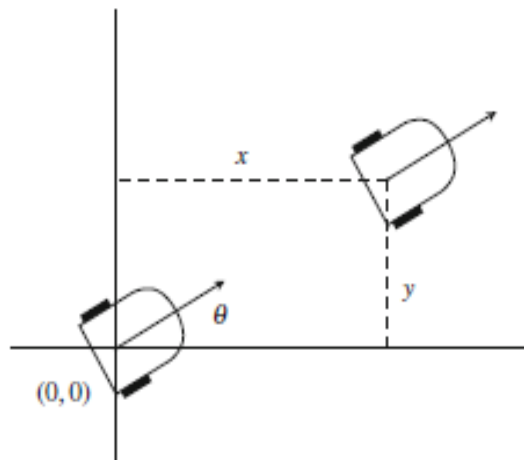
Figur 5.19: Den mobile roboten følger det globale rutealternativet, men må revurdere den planlagte ruten og må finne et alternativt rutevalg.

5.5.6 Odometri

Robotens odometri benyttes ofte i forbindelse med navigasjon. I figur 5.15 ser vi at både lokaliseringssystemet og den lokale planleggeren benytter odometri, hver for seg. Odometrien er et mål på endring i posisjon over tid, og brukes ofte som et såkalt prediksjonssteg for å estimere det beste forslaget på hvor roboten befinner seg. Det estimerte posisjonsforslaget sammenliknes opp mot et korreksjonssteg som kan være en scan matching (laserskanner og kart). Dette gjøres av lokaliseringssystemet. [24, 73].

Den tilbakelagte distansen kan estimeres enten fra hastighetskommandoene til robotens motorer eller med odometrien. Robotens hastighet måles oftest, men ikke nødvendigvis med encoderne. Tiden måles med en intern klokke. Odometrien beregner den tilbakelagte strekningen ved bruk av formel 3.1 i tabell 3.3. Odometrien kan bare beregnes etter at roboten har forflyttet seg og beregnes ofte i periodiske intervaller som for eksempel 20 ganger i sekundet. Encoderne benyttes av odometrien til å måle avstanden over tid.

Dersom roboten beveger seg i en rett linje langs x-aksen med startposisjon $(0,0)$, vil roboten etter t -sekunder befinne seg i posisjon $(vt, 0)$. Dersom roboten skal bevege seg i to dimensjoner (2D) må man vite robotens posisjon (x, y) i forhold til startposisjonen og retningen (θ) den skal bevege seg i. Som vist i figur 5.20. Robotens bevegelse i planet beskrives som et tredimensjonalt system på formen (x, y, θ) og kalles på fagspråket *pose*. Dersom roboten beveger seg i en rett linje, er robotens nye posisjon (x, y) gitt av formel 3.2 og 3.3 i tabell 3.3.



Figur 5.20: Odometri: posisjon og retning [24].

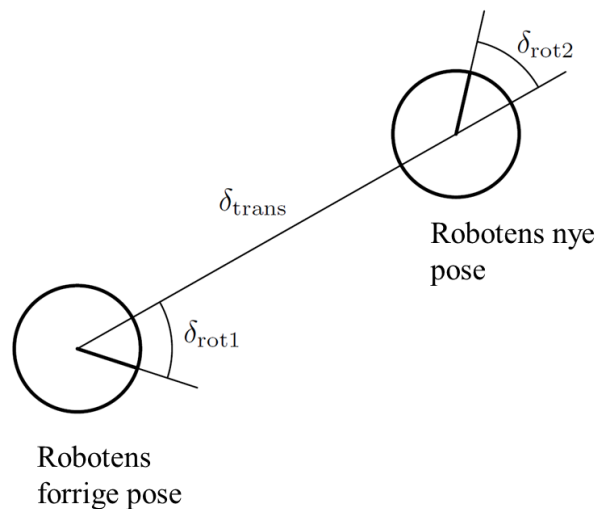
Som vist i figur 5.20 brukes encoderne til å beregne avstanden roboten har forflyttet seg i et gitt tidsintervall. For å kunne beregne robotens bevegelse trenger bevegelsesmodellen informasjon om robotens pose ved forrige tidssteg og hastigheten i nåværende tidssteg som er beregnet ved hjelp av encoderdata og robotens kinematiske modell. Output fra bevegelsesmodellen er robotens pose ved nåværende tidssteg, som igjen vil være input ved neste tidssteg. Dette uttrykkes gjennom følgende formel:

$$u_t = \begin{pmatrix} \bar{x}_{t-1} \\ \bar{x}_t \end{pmatrix} \quad 5.6$$

hvor u_t er robotens bevegelsesmodell, \bar{x}_{t-1} er odometrimåling ved forrige tidssteg og \bar{x}_t er odometrimåling ved nåværende tidssteg.

Denne bevegelsesmodellen brukes for å estimere forskjellen mellom to posisjoner. Modellen baserer seg på tre enkle bevegelser slik at roboten kan komme inn på riktig bevegelsesretning. De overnevnte bevegelsene er rotasjon (δ_{rot1}), etterfulgt av en rettlinjet bevegelse av roboten (translasjon) (δ_{trans}) og til slutt en ny rotasjon (δ_{rot2}). Dette er illustrert i figur 5.21. En mer detaljert fremgangsmåte kan ses i *Probabilistic Robotics* [73].

Bruken av odometri for å estimere robotens posisjon er nesten aldri nøyaktig nok. Årsaken til dette kan være unøyaktige målinger som følge av ujevnheter i overflaten som roboten beveger seg på eller hjulene kan spinne som følge av glatt underlag, feil dekk eller belter. Resultatet av de unøyaktige målingene akkumuleres over tid, som vil si at usikkerheten forbundet med posisjonsestimatet alltid vil øke.



Figur 5.21: Odometrimodellen presenterer robotens bevegelse fra forrige pose $[t-1]$ til ny pose $[t]$. Estimering av ny pose er først en rotasjon δ_{rot1} , etterfulgt av en translasjon δ_{trans} og tilslutt en ny rotasjon δ_{rot2} [73].

5.5.7 Simultaneous Localization and Mapping

Simultaneous Localization and Mapping (SLAM) [73] benyttes når roboten befinner seg i et ukjent miljø, hvor den verken har tilgang på kart eller kjennskap til sin egen lokalisering.

Siden roboten ikke har kjennskap til omgivelsene eller sin egen pose, må kartet genereres ved hjelp av målinger når roboten er i bevegelse. Roboten må derfor benytte de påmonterte sensorene til å estimere avstander og pose samtidig som den kartlegger området og genererer et kart. Det finnes to typer SLAM som bygger på sannsynlighetsteori, online SLAM og full SLAM. Online SLAM gjør estimering av pose ut ifra kartet som er generert ved et bestemt tidspunkt t . Sannsynlighetsestimaten er gitt ved,

$$P(x_t, m \mid z_{1:t}, u_{1:t})$$

hvor x_t er posisjonen ved den bestemte tiden t , m er kartet, og $z_{1:t}$ og $u_{1:t}$ er respektivt sensormålingene og handlingene roboten har gjort frem til tiden t .

Full SLAM derimot benytter seg av tidligere beregninger av pose i forhold til kart i tillegg til nåværende pose, når en ny pose skal estimeres. Dette er gitt ved,

$$P(x_{1:t}, m | z_{1:t}, u_{1:t})$$

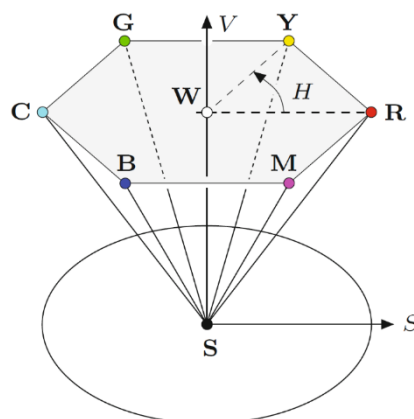
Dersom vi sammenlikner estimatene over ser vi tydelig at online SLAM lokaliserer seg basert på nåværende tid. Full SLAM estimerer lokasjonen basert på nåværende og tidligere lokasjoner.

SLAM kan generere to typer kart, enten grid map eller feature-based map. Grid map er beskrevet tidligere og egner seg for roboter utstyrt med laserskannere. Feature-based map baserer seg på deteksjon av landemerker som skiller seg ut, enten i farge eller form, er oftest benyttet på roboter som beveger seg i åpne landskap [76]. I ROS finnes SLAM i pakken, gmapping [77]. ROS gmapping benytter et occupancy grid map, basert på data fra en laserskanner og robotens pose.

5.6 HSV fargemodell

Farger er et komplisert fenomen som består av elektromagnetiske stråler med bølgelengder mellom 380nm og 700nm. Strålene i dette området kalles synlig lys. Fargene som er en del av det synlige lyset er de grunnleggende fargene i det elektromagnetiske spekteret som danner regnbuens farger. Hver av regnbuens farger er satt sammen av en enkel grunnfarge, med sin egen bølgelengde. Andre farger som rosa og brun er satt sammen av flere farger og dermed flere bølgelengder [78].

En fargemodell er en måte å presentere og beskrive fargenyanser på. I RGB-fargemodellen beskrives fargebilder basert på mengden rød, grønn og blå farge. I HSV-fargemodell beskrives fargene ut ifra tre komponenter, Hue (fargetone), Saturation (metning) og Value (verdi). Modellen presenteres ofte ved en sylinder eller kjegle som vist i figur 5.22. De nøytrale fargetonene (H) ligger i 360° rundt den vertikale aksen. Metningen (S) bestemmer fargens gråtone, og går fra grå i origo til primærfargene i ytterkanten av kjeglen. Verdien (V) sier noe om fargetonens lysstyrke og går fra svart i origo, i en vertikal retning, til primærfargene rundt kjeglens ytterkant [79].



Figur 5.22: HSV-fargemodell presentert som en kjegle [79].

6 KONSEPTSPESIFISERING

I dette kapittelet skal konseptets målsettinger og de viktigste egenskapene som er av betydning for den funksjonelle inspeksjonsroboten beskrives. I tillegg vil robotens dimensjoner kartlegges slik at den blir hensiktsmessig til de områder den skal benyttes. Designet av den kommersielle roboten er ikke en direkte del av prosjektets rammer, men beskrives for å sette roboten inn i riktig perspektiv. Dimensjonene er tenkt som en veileder til andre som eventuelt ønsker å ta konseptet videre. Konseptet for den mobile inspeksjonsroboten er utarbeidet med hensyn til kommersialisering av produktet.

6.1 Konseptmålsettinger

Konseptets **hovedmål** er en funksjonell robot som fysisk skal passe inn og ta seg godt frem i det aktuelle inspeksjonsområdet. Roboten skal være modulær slik at den enkelt kan tilpasses forskjellige inspeksjonsområder og type inspeksjoner. Inspeksjonsroboten skal fremstilles til en konkurransedyktig pris.

Det konseptuelle hovedmålet deles inn i tre **delmål** som er viktige for å nå den ønskede funksjonelle inspeksjonsroboten.

1. Funksjonelle mål

Robotens design skal være modulært og robust mot ulike belastninger og fuktighet. Modulariteten gjør at det vil være enkelt å tilpasse roboten i henholdt til ulike inspeksjonsforhold. Festeordningene til sensorene bør være standardiserte slik at sensorer enkelt kan byttes ut ved behov. Sensorene som monteres utenpå robotplattformen må ha en tilstrekkelig beskyttelse, slik at de ikke ødelegges under inspeksjon. I denne sammenheng må også roboten være beskyttet mot fuktighet.

2. Brukermål

Roboten skal være «den forlengede armen» til fagfolk ved inspeksjonsoppdrag. Det er derfor viktig at roboten er brukervennlig, enkel å håndtere og at den tar seg godt frem i arbeidsområdet. Siden roboten er modulær vil den også være enkel å vedlikeholde. I tillegg er det viktig for operatøren med lang batteritid og kort ladetid, men hensyn til brukstid.

3. Økonomiske mål

Utviklings- og produksjonskonstandene holdes nede ved å benytte enkle og symmetriske geometrier, samt at komponentene er lett tilgjengelig. På denne måten vil man oppnå stordriftsfordeler og en konkurransedyktig pris.

6.2 Rangering av viktige konseptegenskaper

For å få en bedre tilnærming til konseptet ble det definert egne målsettinger i delkapittelet over. For den kommersielle prototypen vil hvert delmål ha egenskaper som er av betydning for konseptet. Ved bruk av Pughs metode som ble presentert i kapittel 3.2.3, vil egenskapene bli vektet. Delmålene med tilhørende konseptegenskaper, begrunnelse og vektning er presentert i tabell 6.1.

Tabell 6.1: Egenskaper som er viktig for inspeksjonsroboten

Delmål	Konseptegenskap	Begrunnelse	Vekting
Funksjonelt mål	Robust	Roboten må være robust mot ulike belastninger og mot vann.	4
	Modularitet	Konseptet må være så modulært at det enkelt kan tilpasses de ulike arbeidsforholdene og skiftes ut ved defekte komponenter.	5
	Estetikk	Et estetisk konseptdesign vil ikke være nødvendig for å få en funksjonell robot, men med hensyn til kommersialisering kan den ha en betydning.	2
	Størrelse	Robotens dimensjoner må tilpasses de arbeidsforhold den skal operere i.	5
	Vekt	For gode manøvreringsegenskaper er det viktig med god bakkekontakt. Det betyr at vekt er en viktig egenskap for å få en god fremkommelighet.	3
Brukermål	Brukervennlighet	Inspeksjonsroboten må være enkel å betjene og ha et intuitivt og oversiktlig kontrollpanel.	5
	Vedlikehold	Roboten må enkelt kunne holdes ved like slik at kvaliteten opprettholdes.	4
	Montering	Det må være enkelt å skifte defekte komponenter eller bytte til andre komponenter ved behov.	4
Økonomiske mål	Symmetri og tilgjengelige komponenter	For å holde kostnadene nede er det nødvendig å benytte standardiserte geometrier og komponenter.	4
	Kompleksitet	Dersom konseptet er for avansert vil det øke både produksjonskostnad.	3

6.3 Metriske grensespesifikasjoner og variasjonsbredde

I utviklingsfasen er det mange aspekter å ta hensyn til når det gjelder bruksområder. Det viktigste er at robotens dimensjoner ikke overskrider arbeidsområdets tilgjengelige areal. Det er vanskelig å estimere nøyaktige dimensjoner på de minste, trangeste og smaleste passasjene som denne roboten er tiltenkt å arbeide i. De metriske grensespesifikasjonene vil dermed være grove anslag.

SINTEFs dimensjonskrav til kryperom vil derfor være utgangspunktet for grensespesifikasjonene til denne kommersielle roboten. Minstekrav til høyde i kryperom er 500mm og inngangsluken skal være stor nok til at personer og materialer kan komme gjennom [80]. Gitt disse antakelsene er variasjonsbredden på dimensjonene presentert i tabell 6.2.

Tabell 6.2: *Metriske grensespesifikasjoner presenteres med grove anslag på kryperoms høyde og ankomstlukens lengde og bredde.*

Spesifikasjon	Minimum	Maksimum	Enhet
Høyde – kryperom	100	500	mm
Lengde – ankomstluke	100	500	mm
Bredde - ankomstluke	300	500	mm

6.4 Metrisk grovspesifisering for konseptet

De metriske grovspesifikasjonene baserer seg på de metriske grensespesifikasjonene i delkapittelet over. Spesifiseringen gjøres for å danne et inntrykk av robotens fysiske dimensjoner. Dimensjonene gir også utvikleren og fremtidige utviklere, arbeidsbetingelser og retningslinjer for 3D modellering av roboten. Grovspesifiseringen i tabell 6.3 baserer seg på variasjonsbredden i tabell 6.2 over. Kolbjørn M. Jensen ved Mycoteam AS poengterte at robotens høyde har betydning for robotens anvendelighet.

Tabell 6.3: *Metriske grovspesifiseringer for robotkonseptet.*

Spesifisering	Dimensjon	Enhet
Robotens høyde	100 – 200	mm
Robotens lengde	300	mm
Robotens bredde	250	mm

7 SYSTEMSPESIFISERING

Sammen med konseptspesifiseringen er systemspesifiseringen viktig i utviklingen av den kommersielle inspeksjonsroboten. I dette kapitlet vil systemets målsettinger og de viktigste systemegenskapene defineres. Det vil også bli kartlagt viktige sensoregenskaper slik at fuktighet kan detekteres.

7.1 Systemmålsettinger

For at en robot skal kunne benyttes til inspeksjon og deteksjon av fuktighet i bygninger er det avgjørende at den oppfyller målsettingene.

Hovedmålet til systemet er en funksjonell robot som manøvrerer og navigerer godt i inspeksjonsområdet. Den skal benytte sensorer for å inspisere og samle informasjon om forholdene i bygningsmassen. Basert på sensordata skal roboten vurdere og gjøre en beslutning på bakgrunn av funnet. Deretter skal resultatene av analysen sendes til operatøren.

Det er satt opp tre **delmål** som er essensielle for konseptet:

- **Manøvrering**
For at roboten skal være fleksibel og funksjonell må den kunne navigere og manøvrere i arbeidsområdet, både ved hjelp av en operatør og autonomt. Navigeringen sørger for at roboten kan manøvrere trygt i omgivelsene. Det er også viktig at roboten kan lokalisere seg selv i arbeidsområdet, slik at operatøren vet hvor den befinner seg dersom eventuelle fuktige områder detekteres.
- **Inspisering og detektering**
Alle sensorene roboten utstyres med skal samarbeide om deteksjon og verifisering av fuktighet. Inspeksjonen foregår kontinuerlig i den tiden roboten manøvrerer og navigerer i aktuelt arbeidsområde.
- **Karakterisering**
Dataene som sensorene samler inn må analyseres fordi både normale- og unormale verdier vil bli registrert av sensorene. Det er nødvendig at roboten fortløpende analyserer all informasjon og kan definere uregelmessige mønstre som bør undersøkes ytterligere. Viktig og relevant informasjon må trekkes ut av rådataen og tydeliggjøres for operatøren.

7.2 Systemkrav

For å utvikle et konsept som fagfolk kan bruke under inspeksjon, er det satt opp flere systemkrav som baserer seg på delmålene over. Kravene er listet opp i tabell 7.1, og vil fungere som retningslinjer i den videre utviklingsprosessen.

Tabell 7.1: Beskrivelse og forklaring av viktige systemkrav for robotkonseptet.

Delmål	Kriterium	Begrunnelse
Manøvrering	Navigering	Å kunne navigere i omgivelsene er en viktig faktor for en mobil robot. Det innebærer å manøvrere i omgivelsene og frem mot et ønsket mål. Navigering handler også om å unngå kollisjoner med objekter i aktuelt arbeidsområde.
	Lokalisering	For at roboten skal kunne navigere i omgivelsene må den til enhver tid kjenne sin egen posisjon og orientere seg i henhold til en referanseramme. Lokalisering er også nødvendig for operatøren når roboten gjør en deteksjon.
	Mapping	Autonome roboter trenger et kart av omgivelsene for å kunne manøvrere og navigere. Der det ikke er tilgjengelige kartmodeller av området må roboten kunne ta i bruk et selvgenerert kart av omgivelsene, for selvlokalisering.
	Sensorer for manøvrering	Roboten må utstyres med sensorer som fungerer som øyne. Slik kan den operere autonomt. Sensorene samler inn informasjon om omgivelsene og i samarbeid med programvare vil roboten forstå og navigere rundt i området. Den vil unngå kollisjoner og kommunisere sin posisjon.
	Fjernstyring	Roboten kan kjøre seg fast eller møte på hindringer den ikke klarer å løse. Da må operatøren ha en mulighet til å overta styringen av roboten. Fjernstyring er også egnet hvis eller når inspeksjonsområdet er for vanskelig å navigere i autonomt.
	Visuell kommunikasjon	Dersom roboten skal fjernstyres må den ha øyne og stilstrekkelig lysforhold slik at operatøren ser det samme som roboten.
Inspisering og detektering	Sensorer for deteksjon	For å kunne detektere og verifisere fuktighet i bygningens konstruksjoner må roboten utstyres med de riktige sensorene. Sensorene må kunne detektere og verifisere områder med fuktighet.
	Bevegelighet	Det må påmonteres bevegelige festeanordninger til sensorene, på roboten. Formålet er å gjøre sensorene mer fleksible for operatøren, slik at sensorene kan detektere i ulike høyder, vinkler samt rotere om sin egen akse.
	Modularitet	Applikasjonen skal være mest mulig modulær slik at den blir fleksibel og enkelt kan implementeres på den kommersielle roboten ved behov.
Karakterisering av objekt	Analysering av data	Dataen som sensorene samler inn må analyseres fortløpende slik at den nødvendige informasjonen trekkes ut. Det vil si i real-time.

7.3 Viktige sensoregenskaper for inspeksjons- og deteksjonsarbeidet

Det finnes mange ulike leverandører og modeller av ønskede og aktuelle sensorer, på markedet. Tabell 7.2 presenterer derfor de viktigste generelle egenskapene som må være på plass i de sensorene som skal utføre deteksjonsarbeidet for denne kommersielle roboten.

Tabell 7.2: Viktige sensoregenskaper med begrunnelse.

Generelle egenskaper	Begrunnelse
Presisjon og nøyaktighet	Det er viktig at sensordataene er presis og nøyaktig. Det vil si at sensormålingene må stemme overens med kjente referanseverdier gjort i laboratorier.
Målefeil og sensitivitet	Det er viktig at forskjellen mellom den målte og den virkelige verdien i materialet er marginal, slik at man unngår følgefeil i systemet.
Oppløsning	For å oppnå pålitelige målinger er valg av oppløsning på sensoren viktig. Oppløsningen sier noe om den minste og største målbare verdien sensoren kan detektere, for å gi en sikker registrering av den fysiske størrelsen. Har en sensor en oppløsning på 3 grader celsius, vil det være umulig å differensiere temperaturer i områder rundt 20-22 grader celsius.
Repetisjonsbarhet	Dersom sensoren møter på identiske forhold er det viktig at resultatene blir like.
Stabile målinger	Sensorene må kunne utføre stabile målinger over tid og ved gjentakende repetisjoner.
Pris	Økt kompleksitet på sensorer medfører økte kostnader. Jo bedre, jo dyrere. Det er viktig å vurdere sensorenes kompleksitet opp imot behov, slik at valget gir best resultat i henhold til pris.
Dimensjon	Sensorene er begrenset til robotens designmessige størrelse.
Field of View (FoV)	Sensorens synsfelt forteller hvilket område sensoren kan registrere og gjøre deteksjoner fra. Det defineres ofte i horisontale (hFoV) og vertikale komponenter (vFoV). Et bredt synsfelt samler mer informasjon i samme bilde, enn en sensor med smalt synsfelt.

I likhet med sensorer finnes det også mange ulike leverandører og modeller av IR-kameraer på markedet. Tabell 7.3 på neste side presenterer en liste med egenskaper som bør vurderes før et spesifikt IR-kamera velges [81].

Tabell 7.3: Viktige egenskaper ved IR-kamera med begrunnelse [81].

Egenskaper	Begrunnelse
Høy detektoropløsning og god bildekvalitet	Ved høy detektoropløsning vil det genereres mer nøyaktige og pålitelige resultater. Det vil være mulig å detektere mindre objekter og lekkasjer, samt generere et skarpere og mer troverdig termografisk bilde. God bildekvalitet gjør det også enklere å visualisere og argumentere for resultatene.
Innebygd RGB-kamera for visuelle bilder, lommelykt og laserpeker	Et termografisk bilde illustrerer bare temperaturforskjellene i undersøkelsesområdet. Det kan derfor i noen sammenhenger være gunstig å illustrere funnområdet med et vanlig RGB-bilde. Operatøren og kunden vil da enklere danne seg et oversiktsbilde, se sammenhenger i funn og lokasjon og komme med en løsning på problemet. RGB-bilde kan også være nyttig som dokumentasjon i en rapport. En innebygd laserpeker gjør det også enklere å markere det aktuelle området.
Et kamera som genererer nøyaktige, repeterbare resultater og observasjoner	Det er viktig at IR-kameraet måler tilnærmet presise temperaturforskjeller slik at objekter og observasjoner kan klassifiseres. FLIR anbefaler kameraer som har en nøyaktighet på $\pm 2\%$. Kameraer av den kvaliteten genererer de beste resultatene. Dersom det er mulig å kontrollere verdier som emissivitet og reflektert temperatur på kameraet, kan målingene bli ytterligere nøyaktige. Fuktighet er vanskelig å detektere. Det er derfor nødvendig å kunne måle de minste temperaturforskjellene slik at små fuktighetsmønstre blir registrert.
De genererte bildene lagres i et standard filformat.	Observasjonsbilder bør kunne lagres i et standard filformat som for eksempel JPEG. Bildet vil da kunne åpnes og analyseres uten å måtte ta i bruk annen spesifikk programvare. Standard JPEG med innebygd temperaturanalyse kan for eksempel distribueres over mail uten tap av kvalitet. Kameraet bør også støtte MPEG 4 video slik at inspeksjonsvideoer kan spilles av og analyseres. Dette vil være en stor fordel hvis det skal utføres en dynamisk termisk inspeksjon.
Koble IR-kameraet til en trådløs monitor	For operatøren vil det være optimalt å kunne analysere og diskutere observasjoner direkte på nettbrett eller smarttelefon.
Picture-in-Picture (P-i-P) og image fusion	I enkelte situasjoner kan det være hensiktsmessig å implementere et termografisk bilde over et korresponderende vanlig bilde. Det vil være enklere å forstå sammenhenger og situasjoner. P-i-P setter inn en boks med det termisk bilde, over det korresponderende RGB-bildet. Image fusion implementerer RGB- og IR-bildet, til et sammensatt bilde.
Kamera med høy termografisk sensitivitet	Sensitiviteten sier noe om den minste temperaturforskjellen kameraet kan detektere i. Avhengig av lokasjonen på fuktigheten, kan temperaturen på det fuktige området og omgivelsene være minimale. Det er derfor viktig å velge et IR-kamera som kan skille de minste temperaturforskjellene.

Det finnes også mange leverandører og modeller av fuktighetsmålere. Tabell 7.4 presenterer de viktigste egenskapene som må være til stede for å definere en pålitelig fuktighetsmåler [82].

Tabell 7.4: Viktige egenskaper ved fuktighetsmålere med begrunnelse [82].

Egenskaper	Begrunnelse
Målenøyaktighet	Presisjonen til måleapparatet er spesielt viktig når man skal opererer i de lavere fuktighetsområdene, hvor de viktigste og fleste avgjørelsene tas. Målere med stor feilmargin bør unngås, da de ikke detekterer fuktighet i materialer med lavt fuktighetsinnhold.
Fuktighetsområde	Det viktigste er at apparatet kan detektere fuktighet i overgangen mellom tørt og fuktig materiale. Ifølge Wagner Meters er treverket fuktig når måleapparatet viser 5-6% på en fuktighetsskala for treverk. Viser apparatet over 30% er treverket mettet av fuktighet.
Avlesningsdybde	Fuktighet kan fremkomme på materialets overflate, men det kan også ha trukket inn i materialet. Det er derfor viktig å velge et måleapparat ut ifra behov, enten for overflate eller dybdemåling.
Rask og enkel i bruk	Fuktighetsmåleren må raskt og enkelt kunne gjøre målinger i det aktuelle området.
Innstillinger for ulike materialer	Ulike materialer har ulike fuktighetsgrenser og måleren må kunne programmeres i henhold til aktuell fuktighetsskala.

8 KONSEPT- OG SYSTEMBESKRIVELSE

I dette kapitlet presenteres robotkonseptet og -systemet, basert på spesifiseringene i de to foregående kapitlene. Konsept- og systembeskrivelsen bygger på hva som er viktig for det arbeidet roboten skal utføre. Beskrivelsen vil være kortfattet, men samtidig gi tilstrekkelig begrunnelse for hvorfor aktuell løsning er valgt.

Elementene som danner grunnlaget for konseptet og systemet blir beskrevet som følgende:

- Robotkonseptet
- Robotarkitektur
- Manøvrering av roboten
- Inspisering, detektering og verifisering
- Analysering av data

8.1 Robotkonseptet

Den kommersielle inspeksjonsroboten skal være modulær og som nevnt i kapittel 3.2.5 vil dette gi konseptet flere fordeler. Roboten blir da satt sammen av enkle og symmetriske geometrier hvor komponentene er tilgjengelig fra underleverandører. Det vil redusere utviklingskostnadene, gi enklere vedlikehold og det vil være mulig å tilpasse roboten etter de arbeidsforhold og -oppgaver den skal utføre.

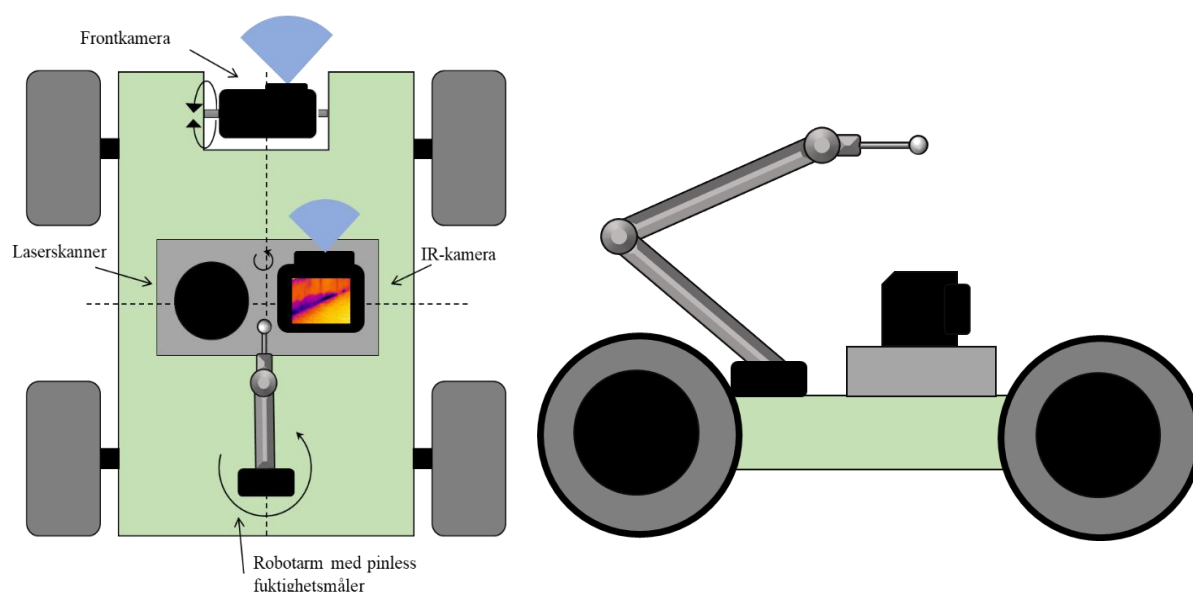
Den tiltenkte systemutviklingsroboten, IMRTBot, benytter et differensielt drivsystem hvor hvert drivhjul er uavhengig av hverandre. Dette oppsettet passer ikke på den kommersielle inspeksjonsroboten fordi roboten da vil bli ustabil, vanskelig å manøvrere og til å forsere hindringer i arbeidsområdet. Av den grunn vil den kommersielle inspeksjonsroboten bli utstyrt med et firehjulsdrevet skid-steeringssystem fordi firehjulsdrift vil gi betydelig bedre fremkommelighetsegenskaper og manøvreringsevner.

I tillegg til firehjulsdrift må man også ta stilling til om roboten skal utstyres med belter eller hjul. Belter har flere fordeler. En av dem er at de gir god fremkommelighet i ulent terreng fordi vekten fordeles over en stor overflate som gir god bakkekontakt. Belter vil også gjøre det enklere å forsere hindringer, trapper og groper som gjør at roboten blir mer anvendelig. Den negative siden med belter er at de krever mer motorkraft, komponentene er dyrere, manøvreringen blir mindre presis og beltene krever mer vedlikehold. Fordelene med hjul sammenliknet med belter er at de er rimeligere, mindre komplekse og krever mindre vedlikehold. Hjul er også enklere å kontrollere og trenger mindre dreiemoment ved start av bevegelser, samtidig som de gir god fremkommelighet. Den negative siden med hjul versus belter, er at hjulenes sentring må være høyere for å kunne håndtere de ujevne underlagene og eventuelle hindringene. Hjulene må også være større enn beltene dersom roboten skal kunne forsere de samme hindringene [83].

For at roboten skal kunne navigere autonomt i arbeidsområdet, er det som nevnt viktig at roboten har god fremkommelighet, bakkekontakt og er enkel å manøvrere. Dersom roboten mister veigrepet på grunn av glatt underlag, kan dette medføre feil i odometrien som igjen fører til at navigasjonen blir feil og at roboten ikke når den ønskede posisjonen. Derfor bør den kommersielle roboten utstyres med belter. Siden prosjektet er på et tidlig utviklingsstadium vil roboten utstyres med hjul da dette er et rimeligere og enklere alternativ.

Roboten skal også utstyres med en navlestreng slik at operatøren kan dra roboten tilbake hvis den setter seg fast, eller heise den ned til et arbeidsområde om nødvendig.

I kravspesifikasjonen i tabell 7.1 ble det nevnt at verifisering av fuktighet må kunne gjennomføres i ulike høyder og vinkler som ligger utenfor robotkroppens rekkevidde. Som beskrevet i kapittel 4.3 kan det være nødvendig å gjøre slike inspeksjoner oppunder taket i arbeidsområdet, i hjørner eller i områder det er vanskelig å komme til. Roboten må derfor utstyres med en robotarm som nødvendig utstyr kan påmonteres. Armen sørger for en mobil og fleksibel rekkevidde, både i horisontal og vertikal retning. Figur 8.1 presenteres et skjematisk konseptforslag til den kommersielle inspeksjonsroboten og figur 8.2 og 8.3 presenterer en 3D CAD modell av det foreslåtte konseptet.



Figur 8.1: Skjematisk fremstilling av et konseptforslag til den kommersielle inspeksjonsroboten, utstyrt med sensorer, kameraer og robotarm. Figuren til venstre er roboten sett ovenfra og til høyre sett fra siden.



Figur 8.2: Konseptuell 3D CAD modell av inspeksjonsroboten, utstyrt med et frontkamera, et IR-kamera og en robotarm med pinless fuktighetsmåler. GoPro Hero4 [84], Robotarm [85], IR-kamera [86]



Figur 8.3: Inspeksjonsroboten på oppdrag i et potensielt arbeidsområde.

8.2 Robotarkitektur

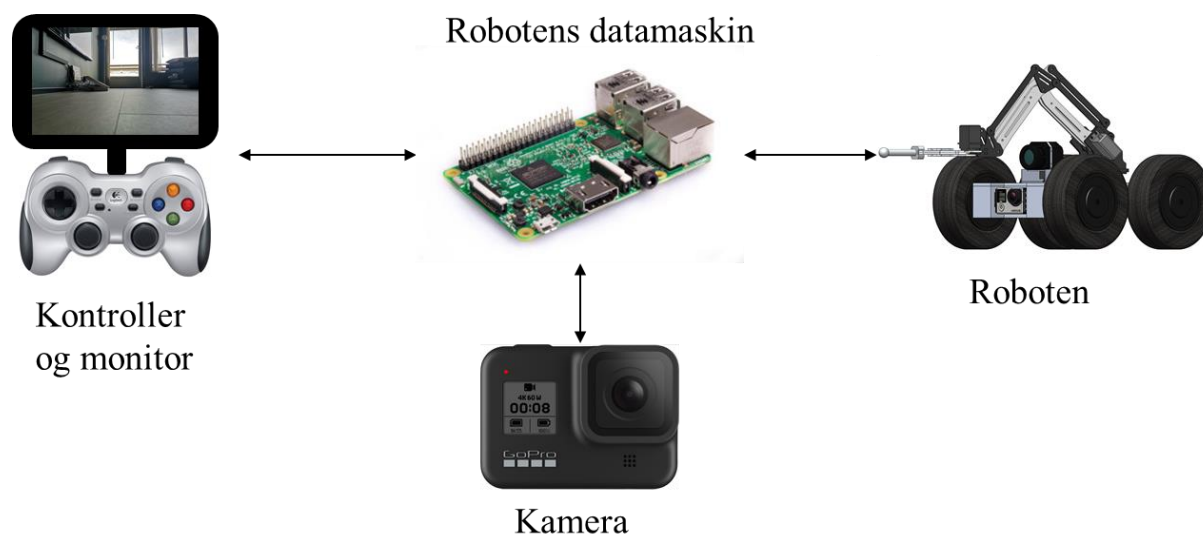
Til å utvikle robotens programvare skal man benytte rammeverket ROS. Årsakene er mange brukere, allsidigheten, modulariteten og friheten til å benytte seg av allerede eksisterende programvare. Siden det er mange brukere av dette rammeverket er det lett tilgjengelige og up-to-date verktøy, biblioteker og programvarepakker som kan implementeres på ulike typer roboter. Programvarepakkene dekker for eksempel fjernstyring av roboten, navigering og sensorsystemer. Denne modulariteten gjør at utvikleren kan laste ned de nødvendige pakkene til sin robot. En annen stor fordel med ROS er at det finnes flere detaljerte brukermanualer for nybegynnere. Brukermanualene er et nyttig hjelpemiddel for dem som ønsker å videreutvikle denne inspeksjonsroboten, men appellerer som oftest ikke til sluttbrukeren av roboten.

8.3 Manøvrering av roboten

For å kunne gjennomføre nøyaktige inspeksjoner er det avgjørende at roboten kan bevege seg i de gitte omgivelsene. Manøvrering er derfor en fundamental og viktig egenskap. Som beskrevet i systemspesifiseringen, tabell 7.1, skal roboten kunne manøvreres på to måter: enten manuelt av en operatør eller autonomt.

8.3.1 Manuell fjernstyring

Manuell kontroll av roboten er en egenskap roboten må inneha av flere årsaker. Det er en fordel at operatøren kan fjernstyre roboten dersom arbeidsområdet ikke egner seg for autonom navigering. Det kan være tilfellet i kryperom dersom underlaget er for ujevnt med groper og forekomster av steiner, sand og jord, men også i tekniske mellometasjer med ulike typer kabler og rør som skaper problemer. Andre tilfeller der manuell fjernstyring er nødvendig er hvis autonomien svikter, eller at roboten kjører seg fast og autonomien ikke klarer å få roboten tilbake på plass.



Figur 8.4: Robotens manuelle manøvreringsoppsett. Kontroller[87], datamaskin som kjører ROS [88], kamera [89].

Manuell manøvrering av roboten kan ses på som et master-slave prinsipp [90], hvor henholdsvis operatøren er masteren og roboten slaven. Ved hjelp av en kompatibel håndkontroller kan operatøren styre roboten manuelt enten trådløst fra annen lokasjon eller gjennom en USB-kabel som vist i figur 8.4. En USB-kabel vil fort bli uhensiktsmessig, så en trådløs kontroll vil være en fordel. I ROS benyttes programvarepakken joy fordi den inneholder alle de nødvendige nodene og driverne som gjør at operatøren kan kontrollere roboten med en håndkontroller. En ROS-node i joy leser dataen og konverterer operasjonene fra håndkontrolleren til ROS-meldinger. Det må så utvikles et program som omsetter ROS-meldingene (kontrollsignalene) fra operatøren til hastighetskommandoer. Da Logitech Gamepad F710, figur 8.5, er en standardkontroll for denne programvarepakken og tilgjengelig på universitetet, er dette et naturlig valg for prosjektet på et tidlig utviklingsstadium. Dersom konseptet skal kommersialiseres kan håndkontrolleren byttes ut med noe mer industrielt.



Figur 8.5: Logitech Gamepad F710 [87].

I de fleste arbeidssituasjonene vil ikke operatøren ha direkte oversikt over robotens bevegelser, i forhold til omgivelsene. Det må derfor implementeres et visuelt hjelpemiddel. Roboten utstyres derfor med et kamera med direktesendt video, slik at operatøren ser det samme som roboten. I 3D modellen, figur 8.1 og 8.2, er roboten utstyrt med et GoProkamera som vist i figur 8.6. Dette kameraet har flere fordeler, det er et kompakt og robust actionkamera med lav vekt, vannrett, bredt synsfelt og med en bildekvalitet på opptil 4K. Den nyeste modellen er også kompatibel med WiFi og bluetooth som vil si at det er mulig å få direktesendte oppdateringer inn på mobil og nettbrett gjennom GoPro appen [89]. Dette er et godt egnet kamera som kan bli montert i fronten på roboten.



Figur 8.6: GoPro HERO8 [89].

For at operatøren skal ha god oversikt over robotens omgivelser må GoProkameraet monteres på en servomotor. Servomotoren sørger for at kameravinkelen og synsfeltet kan justeres etter behov i vertikal retning. En annen utfordring som ble beskrevet i kapittel 4.3 er mangel på lys i arbeidsområdet. Det gjør at roboten må utstyres med kraftige LED-lys med høy effekt som følger kameraets bevegelser. Dette gjør at frontkameraet kan overføre video med god kvalitet til operatøren. Operatøren kan nå manøvrere roboten på en optimal måte. Frontkameraet kan også benyttes utenom manuell manøvrering dersom operatøren ønsker å følge med på hva roboten observerer.

8.3.2 Autonom navigering

Autonom navigering er når roboten gjennomfører inspeksjoner på egenhånd, uten manuell styring fra operatøren. Som nevnt i kapittel 4.3 kan autonom navigering medføre flere fordeler som økt effektivitet og produktivitet, men det er også et sikkerhetsaspekt for inspektøren i likhet med manuell fjernstyring.

For at roboten skal kunne navigere autonomt behøver den et kart over omgivelsene den skal navigere i. Det er ofte tilgjengelig tekniske tegninger over arbeidsområdene, men dersom disse er utilgjengelige må roboten lage dem selv. Det er derfor det autonome navigeringssystemet er satt sammen av to hovedkomponenter, kartlegging og lokalisering med SLAM, og navigasjon med ROS navigation stack.

SLAM – Simultaneous Localization and Mapping

Det første steget for roboten mot autonom navigering i ukjente omgivelser er å kartlegge og tilegne seg den nødvendige kunnskapen om omgivelsene den befinner seg i, samtidig som den estimerer sin egen pose i kartet. For å løse denne kartleggings- og lokalisering utfordringen vil programvarepakken ROS gmapping som beskrevet i kapittel 5.5.7 benyttes, sammen med laserskanner- og odometridata.

Som beskrevet i kapittel 5.2.2 vil den påmonterte laserskanneren estimere avstanden fra roboten til omgivelsene. Denne avstandsdataen benytter gmapping til å opprette et 2D occupancy grid map (okkupasjonskart) som gir informasjon om okkuperte og ikke-okkuperte soner i omgivelsene rundt roboten. Okkuperte soner kan være vegger, materialerester, søyler og lignende.

Odometrien beregnes ved hjelp av sensordata fra robotens encodere og benyttes av gmapping sammen med laserskannerdataen, til å lokalisere roboten i det genererte kartet. Det er derfor viktig at sensordataen er nøyaktig og troverdig. Små feil i odometri-estimatet som friksjon, at robotens hjul spinner eller sklir, kan alle føre til feil i lokaliseringen fordi encoderdataene ikke representerer robotens faktiske bevegelse. På grunn av dette kan det være hensiktsmessig å implementere en inertial measurement unit (IMU) som kan kompensere for disse feilene.

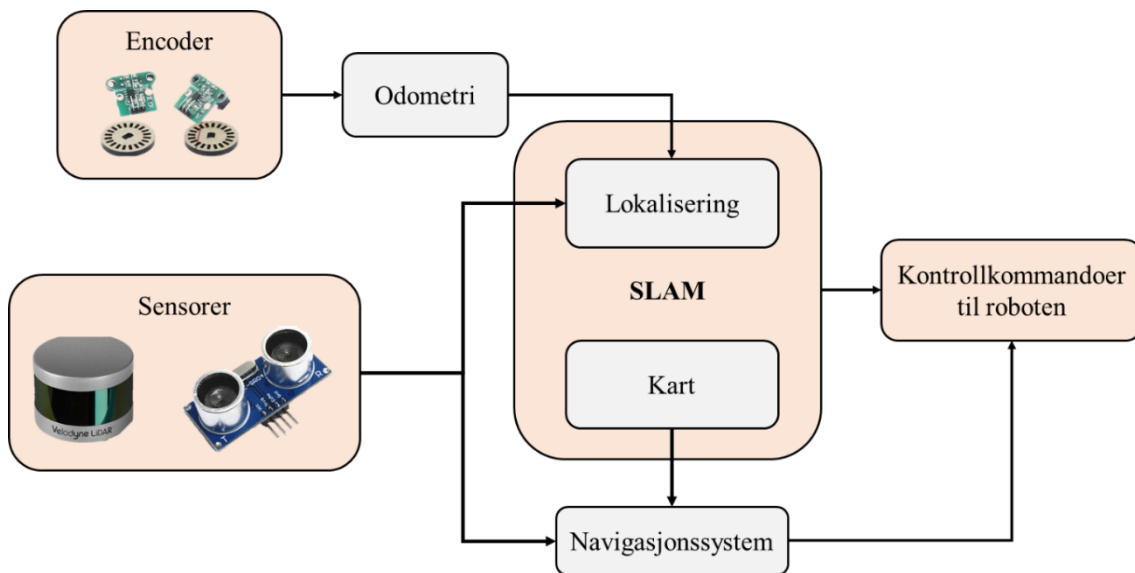
Roboten skal utstyres med en 3D laserskanner med et bredt synsfelt, til å kartlegge arbeidsomgivelsene. Det er valgt en 3D laserskanner fordi det er ønskelig å bygge en 3D representasjon av arbeidsområdet, for det bygningsvisuelle. Det er også en fordel å få registrert hindringer og objekter som fremkommer i ulike høyder og ikke bare i det horisontale planet hvor 2D skannere opererer.

Andre årsaker til at laserskannere er et godt alternativ, er at de fleste er kompatible med ROS. Dette vil forenkle oppkoblingen. Laserskannere fungerer like godt i mørket som i opplyste arbeidsrom og den samler raskt inn data med stor nøyaktighet.

Sensordataene fra laserskanneren som genereres under inspeksjonen, registreres i en punktsky. Punktskyen vil være en 3D presentasjon av robotens omgivelser i den bestemte posisjonen og orienteringen roboten befinner seg i. Dersom det benyttes en 3D SLAM-algoritme, kan sensordataene postprosesseres og et 3D kart av omgivelsene genereres. Siden både ROS gmapping og nav stack opererer i 2D-planet, vil 3D-hindringene projiseres ned til robotens 2D costmap. Dersom laserskanneren oppdager objekter som henger ned fra taket i et kryperom, vil disse registreres som hindringer og roboten vil navigere utenom. Dette er første steg i kartleggingen av robotens arbeidsomgivelsene og lokaliseringen.

Det neste steget er at roboten beveger seg videre til en ny vilkårlig posisjon og orientering. Her vil roboten gjenta og repetere skanneprosessen og generere en ny punktsky. Fra denne punktskyen vil roboten generere et nytt lokalt 3D-kart. Gmapping oppdaterer robotens posisjon fortløpende.

Dersom man ønsker et globalt 3D-kart av arbeidsområdet, kan man enten kjøre en 3D-SLAM underveis eller postprosessere ved hjelp av SLAM i etterkant. Siden robotens prosessorkraft kan være begrenset vil det lønne seg å postprosessere dataen i etterkant av inspeksjonen. Dette fordi 3D-SLAM er prosessorkrevende. Det globale 3D-kartet vil da beskrive hele arbeidsområdet til roboten. Forhåpentligvis kan dette kartet også benyttes til å markere hvor det fuktige området befinner seg slik at det er mulig å finne tilbake til det skadde området. Dette må undersøkes ytterligere etter at konseptet er ferdigstilt. En skjematisk fremstilling av robotens autonome navigering er presentert i figur 8.7.



Figur 8.7: Skjematisk fremstilling av robotens autonome navigering. Odometrien og sensorene benyttes i SLAM som genererer et kart over arbeidsområdet, og lokaliserer roboten i dette. Kontrollkommandoene sørger for at roboten forflytter seg mellom hver skanneprosess i SLAM. Kartet benyttes av navigasjonssystemet for å manøvrere til ønsket område. Inspirasjon hentet fra [91]. Encodeere [92], LiDAR [93], Ultrasonisk sensor [94].

Andre alternativer til kartleggingen av arbeidsområdet enn det som er nevnt (laserskanner) vil være å benytte kameraer og visuell SLAM. Det vil medføre behov for ekstra programvare til bildeanalysingen, samt behov for nok lys i arbeidsområdet. Dersom underlaget genererer mye støy, vil kameraets bildekvalitet bli redusert.

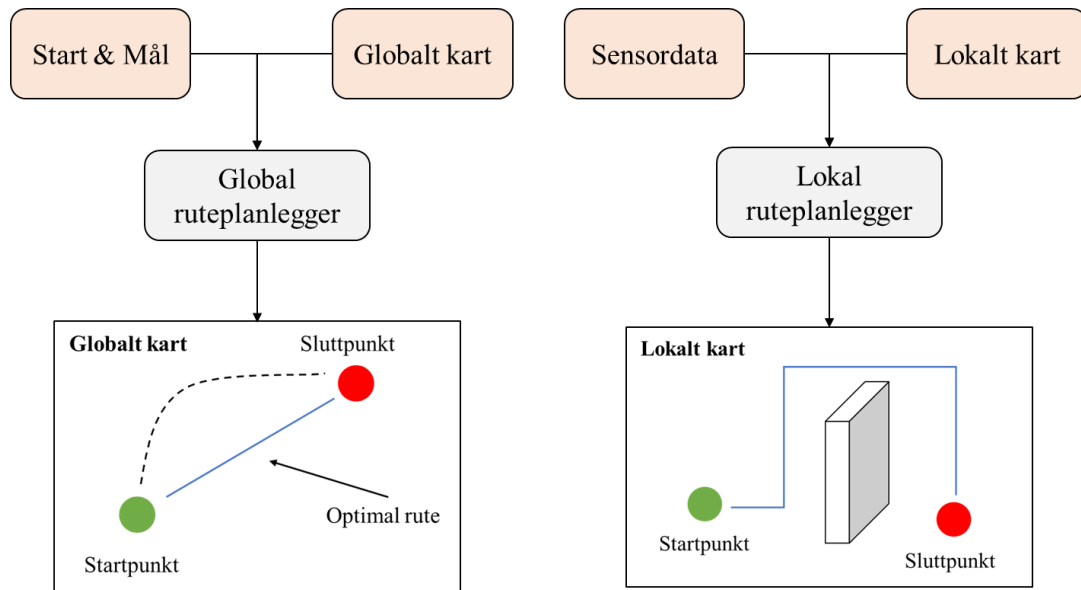
En annen mulighet vil vært å benytte ultrasoniske sensorer til avstandsmålinger. Problemet med disse sensorene er at de genererer upålitelige måleresultater i større områder og med stor avstand mellom objekt og robot. Laserskanneren velges fordi de henter raskt inn data med stor nøyaktighet, slik at dataen enkelt kan hentes ut ved bruk. Pålitelige og nøyaktige 3D laserskannere er kostbare, men til gjengjeld genererer de gode resultater. Dette er en viktig faktor på et utviklingsstadium. Dersom roboten skal kommersialiseres på et senere tidspunkt, kan rimeligere skannere vurderes.

Ruteplanleggingen er nødvendig når roboten har observert potensielle fuktighetsområder og skal manøvrere mot dette for ytterligere verifisering med fuktighetsmåleren. Ruteplanleggeren beregner en brukbar rute fra nåværende posisjon eller et gitt startpunkt, til det aktuelle området basert på kartet som ble generert med SLAM. I arbeidsområdet kan det komme hindringer som ikke er registrert og da vil roboten ha behov for en lokal ruteplanlegger i tillegg til den globale. Figur 8.8 visualiserer begge ruteplanleggerne.

Global ruteplanlegger

Etter at det er observert et potensielt fuktig område i bygningsmassen, må observasjonen verifiseres med fuktighetsmåleren. Roboten må da navigere frem til dette området. Siden ROS er valgt som robotens rammeverk, er det naturlig å benytte den globale ruteplanleggeren i programvarepakken ROS nav stack. Nav stack benytter det globale costmappet som baserer seg på et statisk kart av arbeidsområdet, generert av SLAM. Det statiske kartet inneholder informasjon om hvor de kjente, statiske objektene befinner seg. Som beskrevet i kapittel 5.5.3, vil costmappet benytte informasjonen i kartet til å kartlegge hindringene roboten må ta høyde for, før et egnet rutealternativ blir generert. Den egnede ruten planlegges før roboten starter sin bevegelse mot målet. Hensikten med den langsiktige planen er at roboten skal navigere trygt fra sin nåværende posisjon og frem til målet uten å kolliderer eller sette seg fast.

Under navigeringen vil den påmonterte laserskanneren kontinuerlig sende data om omgivelsene til ROS nav stack, det medfører at det globale costmappet kontinuerlig oppdateres og at robotens posisjon bestemmes. Den kontinuerlige oppdateringen vil også gi roboten fortløpende informasjon om det befinner seg ytterligere hindringer i nærheten som den må ta hensyn til. Dette gjøres ved hjelp av en lokal ruteplanlegger.



Figur 8.8: Skjematisk fremstilling av den globale og lokale ruteplanleggeren. Fremstillingen er inspirert av [91].

Lokal ruteplanlegger

Den lokale ruteplanleggeren konsentrerer seg om det lokale området rundt roboten. Ruteplanleggeren planlegger fortløpende en ny rute dersom uventede hindringer fremkommer på den forhåndsplanlagte ruten. Roboten vil navigere utenom hinderet for å unngå en kollisjon. Dersom navigeringsarbeidet blir for vanskelig, vil også den globale planen oppdateres fortløpende.

Som beskrevet i kapittel 5.5.5 har ROS nav stack også en programvarepakke for en lokal ruteplanlegger. Den lokale ruteplanleggeren benytter et lokalt costmap til å lagre sensordata fra avstandsmålerne (laserskanner og ultrasoniske sensorer). Det lokale costmappet oppdateres kontinuerlig og inneholder informasjon om uventede hindringer i de nære omgivelsene. Den lokale ruteplanleggeren tar utgangspunkt i det globale rutevalget, og prøver å følge dette så langt som mulig. I denne vurderingen tar roboten hensyn til sin kinematikk, dynamikk og hindringenes størrelser.

Det er flere mulige avstandsmålere som kan benyttes til å hente inn lokal informasjon om robotens nære omgivelser. I forbindelse med kartleggingen av arbeidsområdet i SLAM, ble det benyttet en 3D laserskanner. Denne kan også benyttes av den lokale ruteplanleggeren, indirekte gjennom costmappet. I tillegg vil ultrasoniske sensorer også kunne benyttes som en ekstra avstandsmåler for å sikre seg at alle objektene i nærhet av roboten blir registrert. De ultrasoniske sensorene behøver bare å monteres foran og bak på roboten.

Oppsummering av autonom navigering

Roboten vil utstyres med en 3D laserskanner og ultrasoniske sensorer. Dette medfører at roboten vil klare å kartlegge omgivelsene, generere et globalt kart samtidig som den lokaliserer seg selv i kartet ved bruk av SLAM-pakken gmapping. Når den egnede ruten er planlagt ved hjelp av den globale ruteplanleggeren, vil justeringer på rutevalget fremkomme ved hjelp av

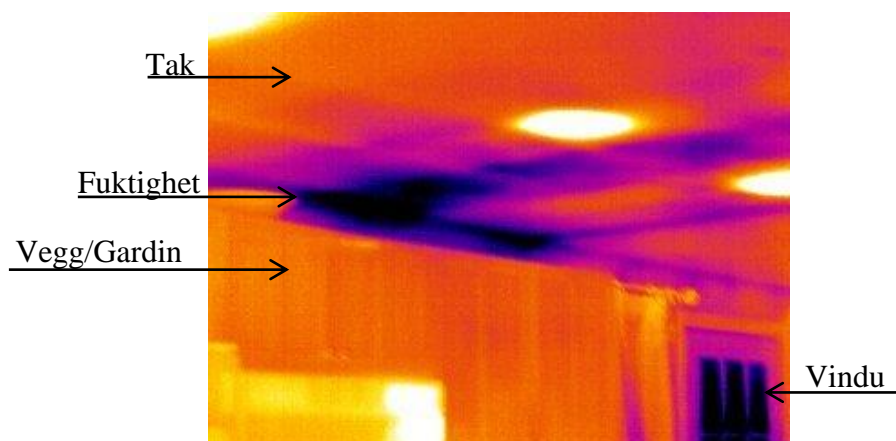
den lokale ruteplanleggeren slik at roboten ikke kolliderer med hindringer som ikke er registrert i det globale costmappet.

8.4 Inspisering, detektering og verifisering

I tillegg til å navigere, må roboten også kunne observere og detektere fuktigheten som befinner seg i arbeidsområdet. Som beskrevet tidligere vil inspeksjonsområdene primært være kryperom og andre mindre hulrom i bygninger hvor det er vanskelig for inspektøren å komme til. Robotens applikasjon består av to deler: inspeksjon og observasjon, og verifisering av observasjonen.

8.4.1 Inspisering og detektering med IR-kamera

For at roboten skal kunne inspisere og observere potensielle fuktige områder på en effektiv og enkel måte, benyttes et IR-kamera. Det er flere årsaker til dette valget. Kameraet krever verken kontakt eller en destruktiv fremgangsmåte for å gjennomføre inspeksjonen. Den andre fordelen er kameraets egenskaper til å detektere temperaturforskjeller, fordi endringer i materialets overflatetemperatur kan være relatert til fuktighet i materialet som beskrevet i kapittel 5.2.3. I figur 8.9 ser man også hvordan det termiske bildet gir nyttig informasjon om hvor potensiell fuktighet befinner seg, brer seg utover i bygningsmassen og hvor den kan ha sin opprinnelse.



Figur 8.9: Termiske bildet fra et IR-kamera som viser tydelig hvor kilden (svart område) er lokalisert, samt hvordan fuktigheten brer seg utover taket (svart, blått og lilla område) [52].

Det finnes mange modeller og leverandører av IR-kameraer. Prosjektet har ikke bestemt et spesifikt kamera som bør benyttes. Tabell 7.2 og 7.3 presenterer derimot egenskaper det er viktig å ta hensyn til før man investerer i et kamera til dette formålet. For dette konseptet er det viktig at kameraet kan detektere et potensielt fuktig område. Det må derfor velges et kamera med høy oppløsning og termografisk sensitivitet, slik at de minste temperaturforskjellene detekteres. Videre må kameraet gjennomføre pålitelige, nøyaktige og repeterbare resultater. Til slutt må også de termografiske bildene eller videoen lagres i et standard filformat slik at de enkelt kan behandles og distribueres videre. Med disse kameraegenskapene vil det være mulig å observere potensielle fuktige områder i bygningsmassen.

Det er viktig å presisere at IR-kameraet bare observerer de termiske forskjellene i bygningsmassen og ikke verifiserer fuktigheten. Av figur 8.9 ser man at det er tydelige temperaturforskjeller, men det kan være flere forskjellige årsaker til dette, for eksempel dårlig isolering eller luftlekkasjer. Dette ser man tydelig rundt vinduet, nede i bildets høyre hjørne.

8.4.2 Verifisering av observasjon ved hjelp av fuktighetsmåler

For å kunne verifisere observasjonene som er gjort med IR-kameraet må roboten utstyres med en fuktighetsmåler. Denne må raskt og enkelt kunne skanne det interessante området som fremkommer av det termiske bildet. Som beskrevet i kapittel 5.2.3 finnes det to aktuelle fuktighetsmålere, pintype og pinless. Pintype fuktighetsmåler er uaktuell for dette konseptet, fordi den har behov for tilleggsutstyr som kan banke målepinnene inn i materialet.

Derfor vil denne roboten utstyres med en pinless fuktighetsmåler som vist i figur 8.10 samt beskrevet i kapittel 5.2.3. Fordelen med denne fuktighetsmåleren versus en pintype er at den oppfyller kravet om rask og enkel skanning over det aktuelle området.

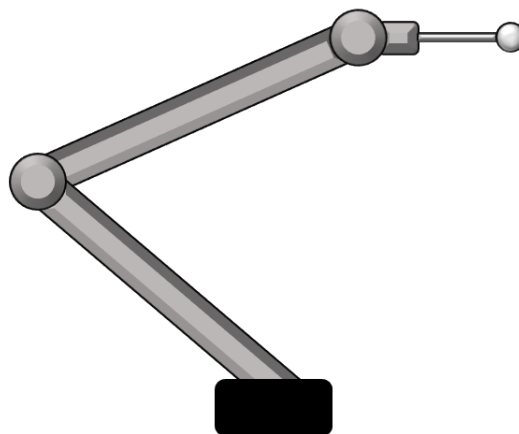


Figur 8.10: FLIR MR59, pinless fuktighetsmåler [95].

Dersom den valgte fuktighetsmåleren har en kuleformet sensortupp som fuktighetsmåleren fra FLIR, figur 8.10, vil sensortuppen enkelt kunne monteres på en robotarm som vist i figur 8.11. Dette gjør det enkelt å verifisere aktuelle inspeksjonsområder som befinner seg utenfor robotkroppens rekkevidde i kryperommene og hulrommene hvor det er vanskelig for roboten å komme til.

Det finnes mange leverandører og modeller av pinless fuktighetsmålere. Til dette konseptet er det ikke bestemt et spesifikt måleapparat som skal benyttes. Tabell 7.2 og 7.4 presenterer egenskaper det er viktig å ta hensyn til før det investeres i en fuktighetsmåler. På et tidlig utviklingsstadium av konseptet er man ute etter gode resultater som kan være med på å bygge opp under konseptets mål. Derfor er det viktig at fuktighetsmåleren har en høy målenøyaktighet,

spesielt i de lave fuktighetsområdene. For å få pålitelige resultater er det også nødvendig at måleren kan bekrefte og avkrefte fuktighet både på overflaten og lenger ned i materialet. Siden materialene i arbeidsområdet kan variere, må fuktighetsmåleren kunne innstilles i henhold til materialtypen som skal undersøkes. Det vil også være en fordel om fuktighetsmåleren kan kommunisere trådløst med IR-kameraet, slik at fuktighetsnivået kan registreres inn på de termiske bildene fortløpende.



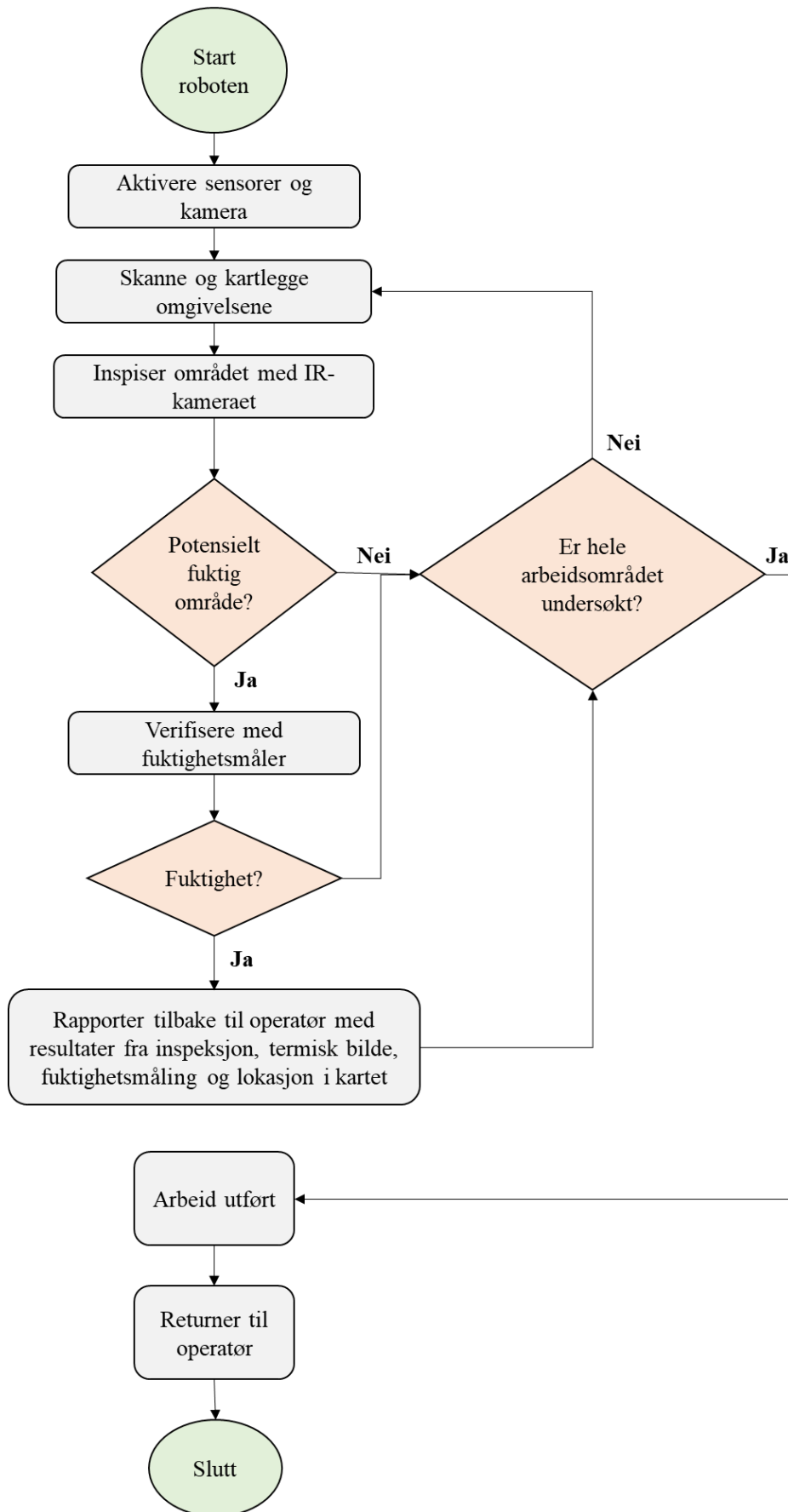
Figur 8.11: Robotarm utstyrt med en kuleformet pinless sensortupp.

8.5 Analysering av termiske inspeksjonsbilder

For at roboten skal vite at IR-kameraet har oppdaget et potensielt fuktig område som fuktighetsmåleren må undersøke ytterligere, må de termiske inspeksjonsbildene analyseres. Det er derfor utviklet en programvare som leser inn termiske inspeksjonsbilder fra IR-kameraet og benytter HSV-fargemodell til å segmentere ut potensielle fuktige områder. Programvaren presenteres ytterligere i kapittel 9.2.3. Denne analyseringen må gjennomføres kontinuerlig og fortløpende under inspeksjonen, slik at fuktighetsmåleren kan aktiveres. Dersom fuktighetsmåleren bekrefter fuktighet må roboten melde observasjonsdataen til operatøren.

8.6 Sammendrag av systembeskrivelsen

Roboten skal enten manuelt eller autonomt navigere rundt i et definert arbeidsområdet for å inspisere og detektere fuktproblematikk i bygde konstruksjoner. For å inspisere området benyttes et IR-kamera for å detektere temperaturforskjeller i materialet. Dersom real-time analyseringen av den termografiske direktesendingen registrerer et potensielt fuktig område, vil roboten motta en melding om å navigere til dette stedet. Der vil roboten aktivere fuktighetsmåleren for å verifisere observasjonen. Dersom det bekreftes fuktighet i det aktuelle område, vil all observasjonsdata overføres til operatøren. Operatøren har nå korrekt informasjon om fuktighetsgraden, omfang og posisjon, for å igangsette tiltak. I figur 8.12 vises en skjematisk arbeidssyklus.



Figur 8.12: Skjematisk presentasjon av robotens arbeidssyklus, steg for steg.

9 TIDLIG KOMPONENTTESTING

Etter å ha kommet frem til et konsept og system for en funksjonell inspeksjonsrobot, er neste utviklingssteg å gjennomføre en tidlig komponenttest av oppsettet for inspeksjon og deteksjon av fuktighet.

På grunn av Covid-19 ble lokalene på NMBU stengt for studenter og ansatte. Dette har påvirket denne tidlige komponenttestingen. Komponenttestingen blir derfor bare beskrevet slik den var tenkt utført. For å teste programvaren som er blitt utviklet for å segmentere fuktigheten fra de termiske bildene er det tatt i bruk relevante bilder funnet på internett. Disse bildene vil også danne grunnlag for resultatene. Det bes om forståelse for denne måten å løse komponenttestingen på. Det er heller ikke mottatt teknisk data på FLIR-kameraet og fuktighetsmåler, disse oppbevares på avstengte arealer.

9.1 Målsetting

Siden konseptet er på et tidlig utviklingsstadium, er det hensiktsmessig å teste applikasjonen som skal kunne inspisere og detektere fuktighet i bygningsmasser. Hensikten er å få en bedre forståelse av applikasjonens komponenter som er IR-kamera og en pinless fuktighetsmåler.

Hovedmålet med denne tidlige komponenttestingen er å undersøke om teorien stemmer overens med IR-kameraets og fuktighetssensorens egenskaper. Dette vil da danne grunnlaget for om denne sensorkombinasjonen egner seg til å detektere fuktighet i bygningsmassen. Det andre målet med komponenttestingen er å utvikle en programvare som segmenterer ut de kalde områdene i de termiske bildene. Programvaren skal analysere den termiske videofeeden i real-time, og melde ifra til roboten dersom den finner et interessant område som trenger ytterligere inspeksjon.

9.2 Beskrivelse av teknologi og materialer

For å få en bedre forståelse av sensoroppsettets egenskaper skulle komponenttestingen blitt gjennomført med utstyr og materialer som var tilgjengelig på universitetet.

9.2.1 Utstyr

Det ville blitt benyttet et IR-kamera og en pinless fuktighetsmåler. I dialog med professor og bildeanalyseekspert Ingunn Burud ble det bekreftet at IR-kameraet kunne egne seg til å detektere fuktighet. Komponenttesten vil benytte et håndholdt FLIR-kamera og et Optris PI 640 (figur 9.1) kamera. Spesifikasjoner på Optris PI 640 kameraet presenteres i tabell 9.1. De termiske målingene på FLIR-kameraet lagres på en minnebrikke inne i kameraet, mens optriskameraet må kobles direkte til en datamaskin. Hensikten med å benytte begge kameraene er for å teste de oppimot hverandre, basert på nøyaktighet og pålitelighet. Fuktighetsmåleren benyttes for å få bekreftet eller avkreftet fuktighet i det området hvor IR-kameraet har registrert temperaturer som kan være relatert til fuktighet.

Tabell 9.1: Optris PI 640, spesifikasjoner [96].

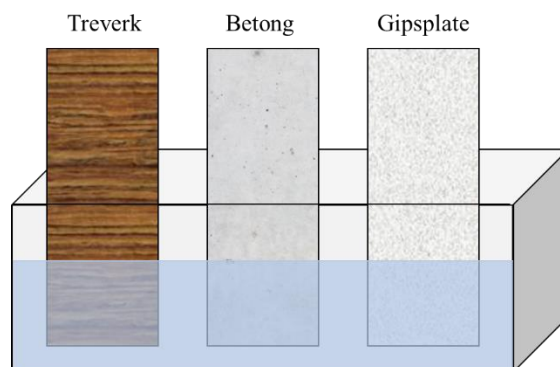
Egenskaper	Spesifikasjoner
Oppløsning	640 x 480 piksler
Bildefrekvens	32Hz
Temperaturområde	-(20) – 900 °C
Termisk sensitivitet	75 mK
Nøyaktighet	± 2 °C eller ± 2%
PC-grensesnitt	USB 2.0
Pris	Fra 75 000 NOK



Figur 9.1: Optris PI 640 IR-kamera [96].

9.2.2 Modellering av fuktighet

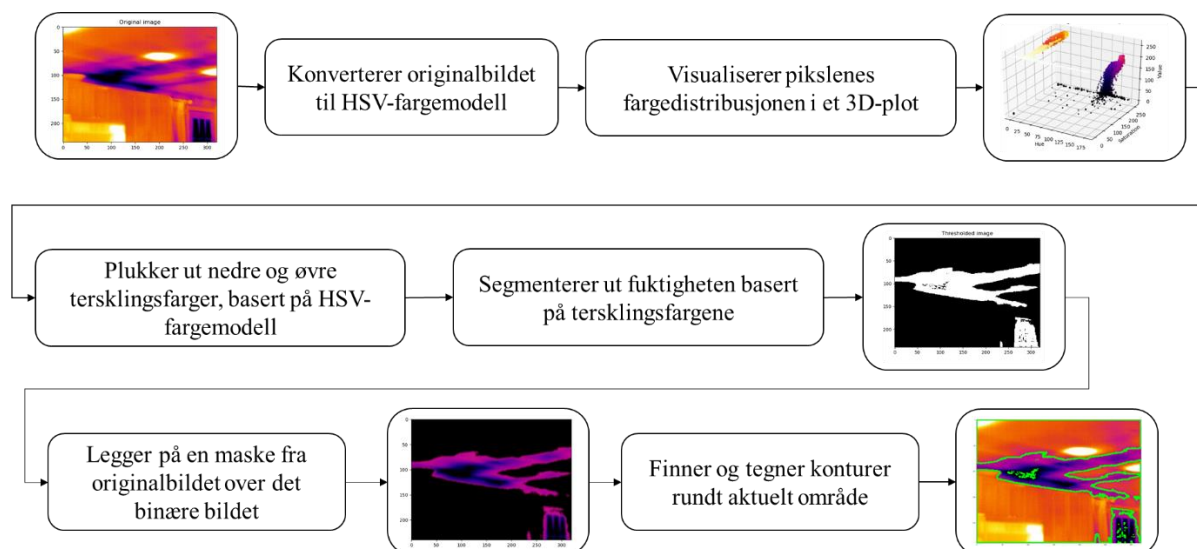
Det ideelle hadde vært å gjennomføre komponenttestingen i et eksisterende bygg som var utsatt for fuktighet. For å få til en mest mulig optimal test velges materialer som ofte benyttes i inspeksjonsområder. Ifølge SINTEF Byggforsk er det treverk og betong i kryperom og trebaserte plater og gipsplater i vegger vendt ned mot disse [97]. I komponenttesten benyttes derfor treverk, betong og gipsplater. Materialene plasseres delvis ned i vannbad som vist i figur 9.2. Dette skal simulere en lekkasje eller innsig av vann.



Figur 9.2: Tre ulike materialer som er delvis senket ned i et vannbad for å simulere en lekkasje og innsig av vann.

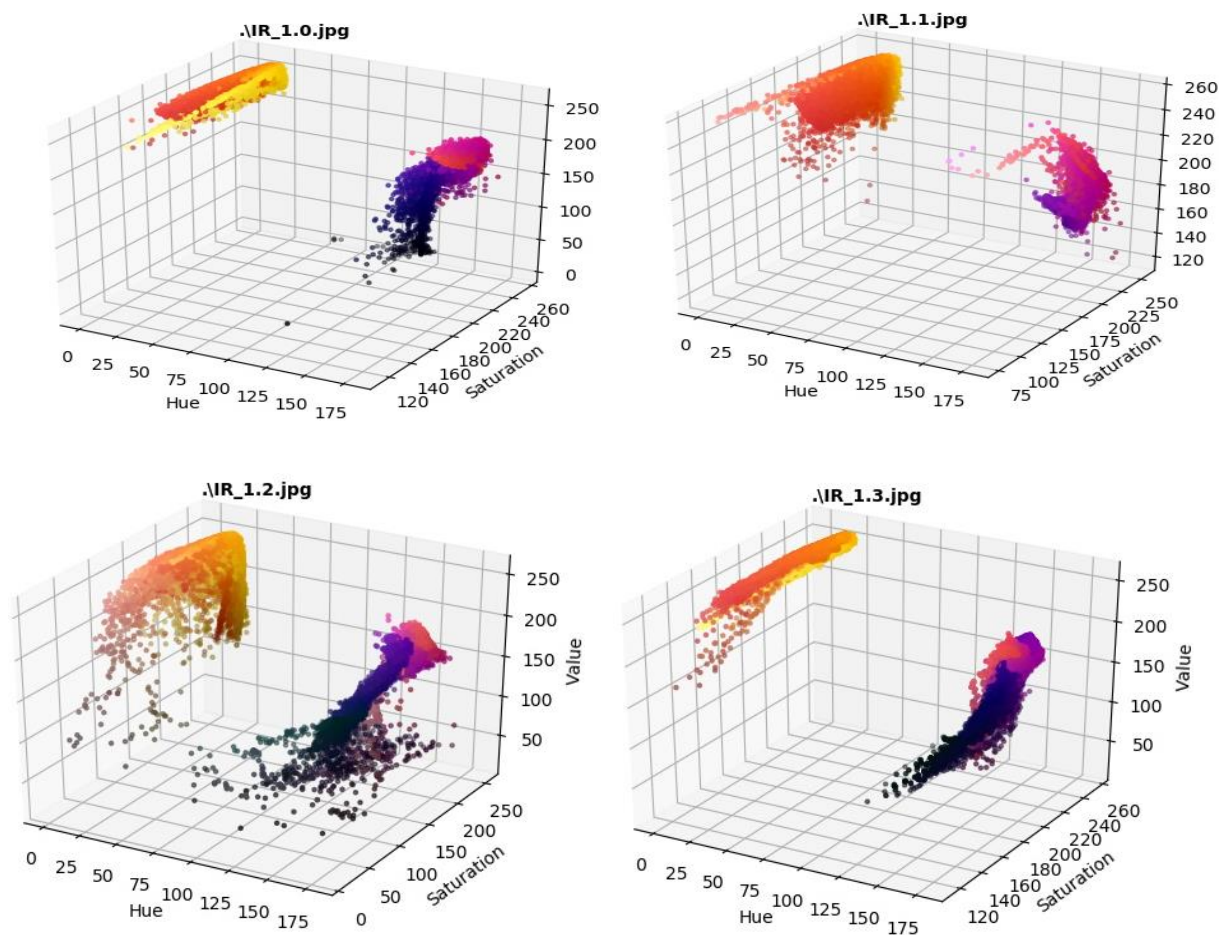
9.2.3 Analysering av termiske bilder

Python, open-source biblioteket OpenCV og HSV-fargemodell blir benyttet for å segmentere ut de fuktige områdene fra de termiske bildene. Det ble også tatt utgangspunkt i eksempelkode fra artikkelen *Image Segmentation Using Color Space in OpenCV + Python* [98] som grunnlag for programvaren. Programvaren er dokumentert i vedlegg A. En skjematisk fremstilling av programmet er vist i figur 9.3.



Figur 9.3: Skjematisk fremstilling av pythonprogrammet. Programmet segmenterer ut et mulig fuktig område som ytterligere skal undersøkes av den pinlesse fuktighetsmåleren.

De fuktige områdene i bygningsmassen segmenteres ut fra de termiske bildene. Som nevnt tidligere og fremstilt i figur 8.9, vil de fuktige områdene illustreres med mørke farger. I figur 9.4 presenteres distribusjonen pikslene i fire termiske fuktighetsbilder, med HSV-fargemodell. Disse fire bildene viser tydelig hvorfor vi ønsker å benytte denne fargemodellen, fordi de aktuelle fargene som man skal ta utgangspunkt i er lette å visualisere og lokalisere. Dette gjøres ved hjelp av fargeområdene $H \in [100, 190]$, $S \in [200, 255]$ og $V \in [0, 200]$. Disse representerer fargetoner i området svart til lilla. Det er dette fargeområdet som benyttes til å segmentere de termiske fuktighetsbildene. Det medfører at pikslene som befinner seg innenfor fargeområdet blir tildelt fargen hvit og de resterende pikslene blir svarte. Resultatet er et binært bilde hvor det fuktige området er segmentert ut. Området som er segmentert ut vil deretter legges oppå det binære bildet. Rundt det segmenterte området blir det lagt konturlinjer på originalbildet og arealet innenfor konturen representerer det området som må bli bekreftet eller avkreftet med hensyn til fuktighet.



Figur 9.4: Figuren presenterer distribusjonen av piksler i HSV-fargemodell, for de fire første termiske bildene (a - d) i figur 9.6.

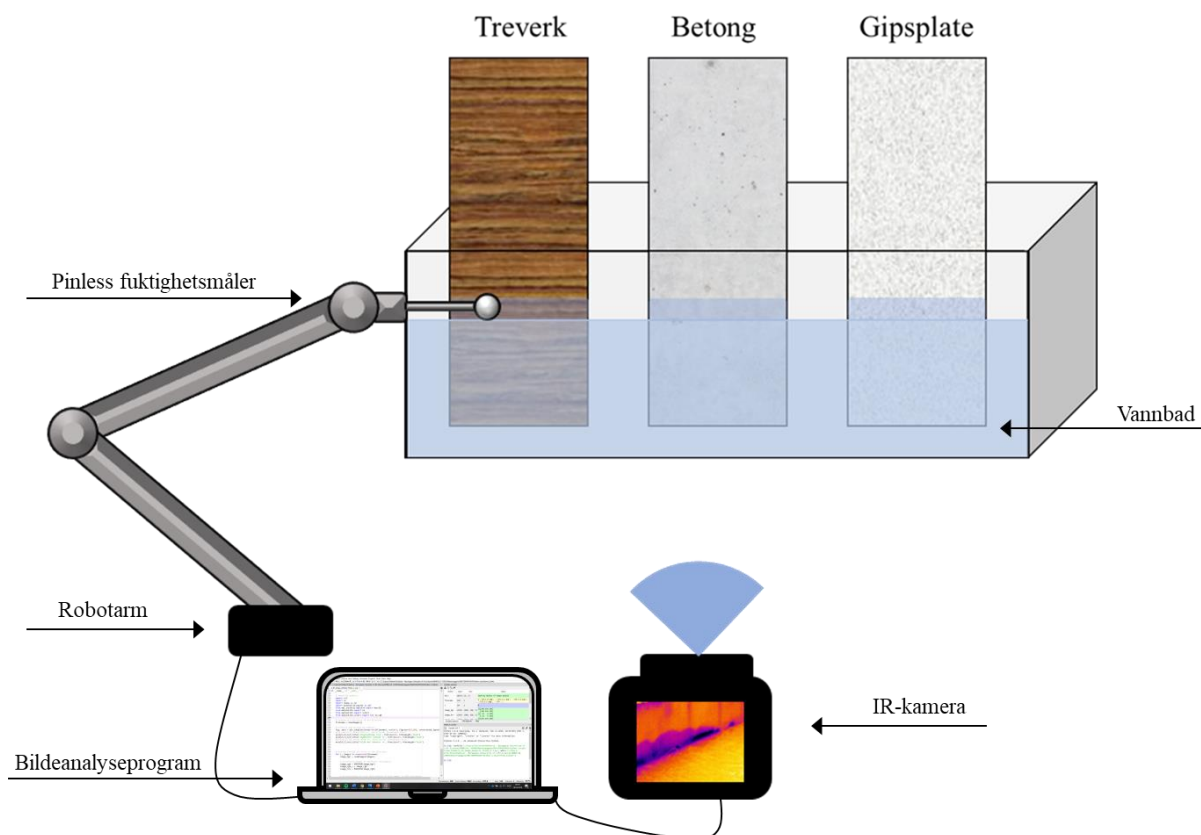
9.3 Metodisk fremgangsmåte

Etter å ha valgt materialer, settes komponenttesten opp som vist i figur 9.5. IR-kameraene settes opp slik at de dekker materialene over vannlinjen. For å simulere en så reell fuktighetssituasjon som mulig, gjennomføres det to testsituasjoner: 1) når materialene står i vannbad og fuktigheten er økende og 2) når materialene er tatt ut av vannbadet og fuktigheten tørker ut.

I første testsituasjon vil det bli tatt jevnlig målinger med IR-kamera. For hver måling vil de termiske bildene bli analysert i det utviklede bildeanalyseprogrammet. Resultatet fra denne analysen avgjør om fuktighetsmåleren skal benyttes til å gjøre målinger i det aktuelle området på materialet. Her er formålet å undersøke om IR-kameraet og programmet kan observere og segmentere ut det fuktige området. Deretter skal fuktighetsmåleren bekrefte resultatet av analysen. Første testperiode er satt til tre uker, jamfør et liknende forsøk som er presentert i boken *Infrared Thermography in Building Moisture Inspection* [50]. I den andre testsituasjonen skal prosessen reverseres og de samme metodiske fremgangsmåten benyttes.

Dette er to reelle scenarier inspeksjonsroboten kan møte. Den første testsituasjonen illustrerer lekkasje og innsig av vann frem til fuktigheten i materialet når metningspunktet. Den andre testsituasjonen undersøker prosessen etter at skaden har skjedd, for å se om fuktigheten er på

retur. Det har betydning for eier, med hensyn på eventuelle tiltak som må igangsettes. I begge situasjonene vil det bli gjort flere målinger. Nullpunktsmåling gjøres for å dokumentere at materialene er tørre før start av komponenttesten, deretter vil det gjøres målinger hver tredje dag i begge testsituasjonene.



Figur 9.5: Skjematiske fremstilling av testoppsettet, med vannbad, materialer, IR-kamera, analyseringsverktøy og pinless fuktighetsmåler. Det termiske bildet er hentet fra [52].

9.4 Resultat og diskusjon

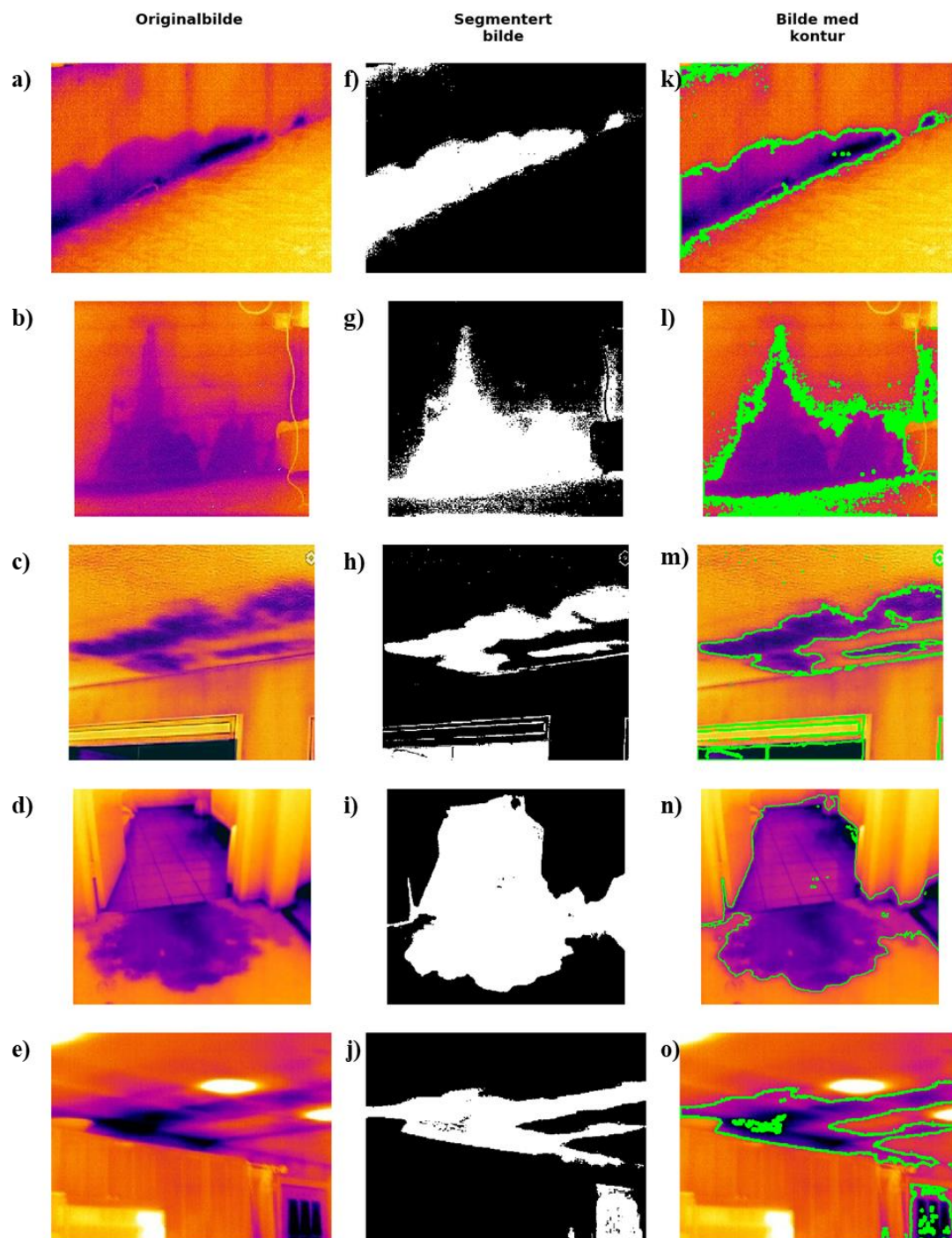
På grunn av Covid-19 måtte den utviklede programvaren testes mot fem ulike termiske fuktighetsbilder som ble hentet fra internett. Disse bildene er vist i figur 9.6, a-e, og representerer reelle fuktighetssituasjoner i bygningsmassen.

Originalbildene viser at IR-kameraet observerer tydelige temperaturforskjeller i materialet. Resultatene fra det utviklede bildeanalyseprogrammet vises i figur 9.6, f – o. Dette bekrefter at det er mulig å analysere de termiske bildene, ved hjelp av HSV-fargemodell. Hensikten med å segmentere ut de fuktige områdene er å definere det området fuktighetsmåleren skal verifisere i henhold til originalbildet med kontur. De binære bildene i kolonnen *Segmentert* viser at det valgte HSV-fargeområdet som ble presentert i kapittel 9.2.3 stemmer overens med områdene i originalbildet.

Resultatene i kolonnen *Segmentert* viser også at noen områder i det fuktige området ikke ble detektert. Mest sannsynlig er årsaken er at disse pikslene lå utenfor det valgte fargeområdet eller at det kan være støy i bildet. Bildene med kontur representerer det området

fuktighetsmåleren skal sjekke ut. Disse målingene vil da bekrefte eller avkrefte om det er fuktighet tilstede i bygningsmassen.

Konklusjonen er at man kan benytte OpenCV og HSV-fargemodell til å segmentere ut aktuelle områder fra de termiske bildene. Hadde den tidlige komponenttesten blitt gjennomført i henhold til plan, ville testen også gitt svar på om denne sensorkombinasjonen, IR-kamera og fuktighetsmåler, hadde passet til arbeidsoppgaven.



Figur 9.6: Fem ulike termiske bilder av reelle fuktighets situasjoner presentert i første kolonne, med ulik mengde fuktighet og område i bygningsmassen. I andre kolonne er det aktuelle området segmentert ut og i tredje kolonne vises det området fuktighetsmåleren skal undersøke. Bilde a), b) og e) er hentet fra [52], c) er hentet fra [99] og d) er hentet fra [100].

10 PROSESSEVALUERING OG DISKUSJON

I dette kapitlet evalueres og drøftes utviklingsprosessen. Hensikten er å få frem forbedringspotensialer og få kartlagt usikkerhetsmomenter i prosjektet.

10.1 Konseptutviklingsarbeidet og forbedringspotensialer

I starten av konseptutviklingen hadde jeg høye forventninger og ambisjoner til konseptet, og målet var å utvikle en fungerende kommersiell prototype av en inspeksjonsrobot. Etter å ha tilegnet meg mer fagligkunnskap ble det satt opp konkrete målsettinger og begrensninger for prosjektet. Dette tydeliggjorde at ambisjonene i utgangspunktet var for høye i henhold til tidsrammen. Masterprosjektet fokuserte derfor på å utvikle et konsept- og systemforslag til en mobil inspeksjons- og deteksjonsrobot.

Siden dette er et nytt bruksområde for mobile roboter var det utfordrende å utvikle et optimalt konsept på nåværende utviklingsstadium. Det ble derfor tatt kontakt med Mycoteam AS som er en ledende aktør innen undersøkelse og vurdering av fukt- og biologiske bygningsskader i Norge. Dette for å få en ekstern vurdering og tilbakemelding på konseptet som var utviklet gjennom prosjektperioden. Tilbakemeldingene var både konstruktive og motiverende. På grunn av tidsrammen og at masterprosjektet var i sluttfasen når Mycoteam AS responderte, var det vanskelig å hensynta deres innspill på nåværende tidspunkt. Innspillene vil bli presentert i det videre arbeidet og vil bli vurdert på neste utviklingsstadium av inspeksjonsroboten. Det var motiverende å få en bekreftelse på at konseptet var interessant og at de ønsker en nærmere kontakt og oppfølging på utviklingen av konseptet. Tilbakemeldingene fra Kolbjørn M. Jenssen hos Mycoteam AS tyder derfor på at konseptet er originalt og kan være et fremtidig verktøy i bransjen.

Basert på responsen fra Mycoteam AS, håper jeg at gradsarbeidet kan være en forløper til en videre utviklingen av robotkonseptet. Dette for at inspektører og fagfolk kan få et nyttig verktøy til inspeksjoner i områder som er mindre tilgjengelige for mennesker.

Når det nå skal reflekteres og diskuteres over konseptutviklingsarbeidet, refereres det til fasene i prosesstrinnene (kapittel 3.4):

Planleggingsfasen

- Planen var å lære mer om roboter, men jeg brukte forholdsvis lang tid på å konkretisere en problemstilling. Det var utfordrende å få til et konsept som kunne videreutvikles inn i noe konkret basert på mine ønsker. Det var først når bruksområdet ble tydeliggjort at arbeidet mitt kunne starte. I denne prosessen mistet jeg en del tid.
- Når robotens bruksområde var besluttet, burde det vært gjennomført en behovsundersøkelse til bedrifter som driver med inspeksjon av fukt- og biologiske bygningsskader. Dette ville ført til et ytterligere spesifisert konsept mot kundebehovet.
- Ambisjonene om å sitte igjen med en prototype på en inspeksjonsrobot, virket overkommelig på et tidlig stadium i prosjektet.

Utredning og læringsfasen

- For å opparbeide en grunnleggende kunnskapsbase startet jeg med å kartlegge kommersielle mobile inspeksjonsroboter. Dette for å få en oversikt over bruksområder, hvor langt utviklingen var kommet og hvilken teknologi som ble benyttet. På et senere stadium i prosessen kom jeg over mer konseptuelle inspeksjonsroboter som jeg burde ha presentert i rapporten. De er mer sammenliknbare med det konseptet jeg ønsker å utvikle.
- Det ble brukt for lang tid på teknologi- og teoriutredelsen fordi robotikk var et nytt fagfelt for meg. Jeg brukte tid på å bygge opp en grunnleggende kunnskapsbase som jeg kunne dra nytte av. Det medførte også at ambisjonen om å sitte igjen med en prototype av inspeksjonsroboten ved gradsarbeidets slutt måtte revurderes. Prosjektets målsetting ble derfor endret til å fokusere på konseptet, og systemet denne inspeksjonsroboten måtte utstyres med.
- Det burde blitt kartlagt og utredet om andre rammeverk enn ROS kunne blitt benyttet til utvikling av robotens programvare.
- Det burde blitt undersøkt om det fantes andre versjoner av SLAM-algoritmer som kunne blitt benyttet.

Utviklingsfasen

- Det burde blitt gjennomført en ekstern undersøkelse av konseptet på et tidligere stadium i prosjektarbeidet. Tilbakemeldingene kunne ha bidratt med flere innspill til egenskaper, utstyr, dimensjoner og programvare roboten burde utstyres med. En slik tilbakemelding kunne også ha medført at utviklingen av konseptet hadde komme lenger på den tilgjengelige tiden.
- Jeg burde ha hatt ytterligere kontakt med fagfolk innen bygg for å kartlegge utfordringer og potensielle sensorer til dette deteksjonsarbeidet.
- Jeg kunne tatt kontakt med robotikkmiljøer for å øke egen kunnskapsbase og diskutere egenskaper og programvare som kunne være aktuelle for roboten.
- Når det kommer til analysering av de termiske bildene burde det også blitt vurdert andre løsninger. Det burde blitt undersøkt om kunstig intelligens kunne blitt benyttet til å segmentere ut aktuelle områder i bildene.

10.2 Design, produksjon og kostnad

Design, produksjon og utviklingskostnader er ikke innenfor prosjektets ramme, men de er en del av det overordnede konseptet. Det vil derfor være hensiktsmessig å diskuteres muligheter og idéer for et videre arbeid. Evalueringen baserer seg på 2D og 3D modellene som er presentert i kapittel 8, men også de designmessige tilbakemeldingene fra Mycoteam AS.

Designrevisjon

- Det er vektlagt et enkelt og modulært design slikt at roboten enkelt kan tilpasses etter behov. Det er ikke tatt hensyn til robotens estetikk og utforming.
- I prosjektet er det vektlagt et enkelt og modulært design, med henblikk på at roboten skal være enkel og brukervennlig i drift. Det kunne ha vært vurdert estetikk, funksjonalitet og optimalisering som salgsfremmende momenter.
- Det er ikke bestemt hvordan sensorplattformen og robotarmen skal monteres på robotkroppens topplokk. Det finnes mange løsninger, men det er viktig at sensorplattformen er bevegelig, slik at IR-kameraet kan rotere uavhengig av robotens kjøreretning. Dersom roboten skal kartlegge arbeidsområdet med en 3D laserskanner, må denne monteres slik at robotarmen og IR-kameraet forstyrrer målingene.
- Et av rådene fra Mycoteam var at robotens høyde er viktig. De benytter i dag sensorer med en høyde på 250 mm, denne setter seg ofte fast. Robotens totale høyde i burde ikke overstige disse 250 mm, men alt under 150 mm er optimalt.
- Basert på tilbakemeldingen fra Mycoteam, bør det finnes en robotarm som kan bli så kompakt som mulig. Det optimale hadde vært å senke robotarmen ned i robotkroppen.
- Det må utvikles modulære festeanordninger til alle komponenter som skal monteres på roboten, slik at roboten blir enkel å tilpasse arbeidsområde og hvilken type inspeksjon som skal gjennomføres.

Produksjon og kostnadsreduksjon

- Robotkroppen består av enkle og standardisert geometrier. De kan derfor enkelt skjæres ut av råmaterialeplater, enten med laser-, vann- eller plasmakutter. For å redusere overskuddsmateriale bør flest mulig komponenter skjæres ut av samme plate.
- Alle komponenter bør vær hyllevarer og kjøpes av eksterne leverandører.
- Hyllevarer kan være billig å kjøpe separat, men det burde undersøkes om det er mulig å oppnå stordriftsfordeler ved å kjøpe større kvantum. Dette kan føre til et billigere konsept.
- Estetikk og utforming, den kan være kostnadsøkende i forhold til produksjonen

11 KONKLUSJON

Gjennom dette masterprosjektet er det utredet, utviklet og foreslått et mobilt robotkonsept for inspeksjon og deteksjon av fuktighet i bygninger.

For at roboten skal kunne brukes til inspeksjonsarbeid, må den bevege seg i det aktuelle arbeidsområdet. Dette kan gjøres på to forskjellige måter. Med en Logitech håndkontroller og programvarepakken ROS joy, kan roboten fjernstyres manuelt av operatøren. Det andre alternativet er en autonom robot. SLAM (Simultaneous Localization and Mapping) fra programvarepakken ROS gmapping vil bli benyttet til å kartlegge arbeidsområdet og lokalisere roboten i det genererte kartet, og ROS navigation stack vil bli benyttet til å navigere roboten rundt i det området som skal inspiseres.

Samtidig som at roboten manøvrerer og navigerer, skal roboten også kartlegge områder som kan være fuktige ved hjelp av et IR-kamera. De termiske bildene fra kameraet analyseres med et bildeanalyseprogram som segmenterer ut et potensielt fuktig område. Dette området må sjekkes ut med en pinless fuktighetsmåler som er montert på en robotarm. Fuktighetsmåleren vil da raskt og effektivt bekrefte eller avkrefte om det er fuktighet i det aktuelle området.

For å få en visuell presentasjon av konseptet, er det ved hjelp av et 3D CAD-verktøy modellert et forslag til en kommersiell inspeksjonsrobot. Roboten er modulær, noe som gjør det enkelt å utstyre roboten med nødvendig utstyr samt enkelt å demontere, skifte ut og vedlikeholde komponentene. Det er ikke lagt vekt på design og festemekanismer til sensorer og robotarm. Her er det muligheter for videre utviklingsarbeid.

På grunn av Covid-19 ble ikke komponenttesten gjennomført i henhold til plan. Oppsettet er beskrevet, men bilder ble hentet fra internett for å teste om det utviklede bildeanalyseprogrammet egnert seg til å segmentere ut det ønskede området basert på HSV-fargemodell. Programmet bekreftet at det er mulig å segmentere ut og markere det ønskede området basert på HSV-fargemodell, slik at roboten og robotarmen vet hvor fuktighetsmåleren skal bekrefte og avkrefte fuktighet.

11.1 Resultater og anbefalinger

Gjennom gradsarbeidet har man fått bekreftet at det er mulig å ta i bruk eksisterende teknologi for å utvikle en inspeksjonsrobot. Utviklingsprosjektet har kommet frem til følgende resultater og anbefalinger.

Resultater:

- Det er kartlagt og beskrevet et konsept- og systemforslag til en kommersielle inspeksjonsroboten.
- ROS kan brukes til å utvikle en mobil inspeksjonsrobot. Ved å benytte dette rammeverket, sparer prosjektet mye utviklingsressurser.
- IR-kameraet som er montert på en bevegelig plattform, kan benyttes til å observere potensielle fuktige områder i bygningsmassen.

- En pinless fuktighetsmåler som er montert på en robotarm, sørger for dynamiske inspeksjoner. Robotarmen sørger for en rask og enkel skanning og at den pinlesse fuktighetsmåleren kommer til i vanskelig tilgjengelige områder.
- HSV-fargemodell og den utviklede programvaren kan benyttes til å segmentere ut og markere et potensielt fuktig område i det termiske bildet.
- Ved bruk av frontkamera og en LED-bar, oppnås visuell kommunikasjon mellom roboten og operatøren, også i mørke omgivelser.
- En modulær robot kan tilpasses arbeidsoppgavene, samt at den er enkel å vedlikeholde.
- Det er 3D CAD-modellert et konseptforslag til den kommersielle prototypen.
- HSV-fargemodell og den utviklede programvaren kan benyttes til å segmentere ut og markere et potensielt fuktig område i det termiske bildet.

Anbefalinger:

- Den tidlige komponenttesten bør gjennomføres med IR-kamera og fuktighetsmåleren, slik at man er sikker på at oppsettet fungerer i praksis.
- Programvaren for segmentering av potensielle fuktige områder i de termiske bildene må gjøres om, slik at termiske videoer kan analyseres. Deretter må det undersøkes om programmet fortsatt kan segmentere ut og markere de potensielle fuktige områdene.
- Det anbefales at den autonome navigeringen med kartlegging og lokalisering av roboten med gmapping, og navigation stack testes i et relevant bruksområde. Det er nødvendig å undersøke om systemet fungerer optimalt.
- Innspill fra Mycoteam AS var at robotarmen også burde utstyres med en gripeklo, som kan ta med råtten ved og soppvekst tilbake til laboratoriet for analysering.
- Innspill fra Mycoteam AS var at robotarmen må kunne komprimeres til et minimum når den ikke er i bruk, slik at totalhøyden til roboten ikke overstiger 250 mm.
- Mycoteam AS har signalisert ønske om kontakt for videre optimalisering av roboten.

11.2 Videre arbeid

På grunn av omfanget, utfordringene og den begrensede tiden, er det fortsatt arbeid som bør gjøres for en kommersiell prototype kan realiseres. Det er derfor satt opp en grov punktvis prioriteringsliste som må gjennomføres før en kommersiell prototype kan settes sammen. Flere av disse arbeidsoppgavene kan være mulige fremtidige masteroppgaver.

- Gjennomføre en omfattende markedsundersøkelse for å optimalisere roboten, basert på de eksterne bedrifter, fagfolks erfaringer, behov og ønsker.
- Designe en modulær, funksjonell og optimal inspeksjonsrobot, samt universelle festeanordninger.
- Kartlegge, velge og kjøpe inn motorer, hjul, kretskort og andre nødvendige komponenter til roboten.
- Kartlegge, velge og kjøpe inn de nødvendige sensorene og kameraene til navigering og inspeksjon.
- Utvikle og designe en optimal robotarm.

- Utvikle, programmere og implementere systemet som er presentert i gradsarbeidet. Det er tenkt å benytte IMRTBot for utvikling av programvaren.
- Videreutvikle pythonprogrammet for real-time bildeanalyse.
- Sette sammen en kommersiell prototype og testing.

11.3 Fremtidig utvikling av konseptet

Etter at den kommersielle prototypen er testet, kan det være aktuelt å se på utvidelser eller andre løsninger til konseptet. Her er noen innspill til en fremtidig utvidelse:

- Roboten bør videreutvikles til å observere, detektere og bekjempe mugg- og råtesopp.
- Benytte kunstig intelligens til å avgjøre om det er et fuktig område, eller om området er angrepet av biologiske organismer. Det optimale hadde vært hvilken type organisme som har angrepet materialet.

12 REFERANSER

12.1 Skriftlige kilder

- [14] E. S. Andersen, K. V. Grude, and T. Haug, *Goal directed project management: effective techniques and strategies*. London, UK: Kogan Page Publishers, 2009.
- [16] W. Eversheim, *Innovation Management for Technical Products: Systematic and Integrated Product Development and Production Planning (RWTH-edition)*, 1. ed. Springer Verlag, Berlin, Heidelberg, Tyskland, 2009, p. 444.
- [18] G. E. Dieter and L. C. Schmidt, *Engineering design*, 4 ed. McGraw-Hill Higher International Edition, New York, USA, 2013, p. 827.
- [19] S. Pugh, *Total Design: Integrated Methods for Successful Product Engineering*. Pearson Education, Essex, United Kingdom, 1991, p. 296.
- [20] M. Baxter, *Product design (Design Toolkits)*, 2nd Reprint ed. CRC Press, (1995)/Nelson Thorne Ltd, Cheltenham, United Kingdom, 2001, 1995, p. 308.
- [22] S. G. Tzafestas, *Introduction to Mobile Robot Control*. Elsevier Science, 2013.
- [23] P. J. From, J. T. Gravdahl, and K. Y. Pettersen, *Vehicle - Manipulator Systems*. Springer, 2014.
- [24] M. Ben-Ari and F. Mondada, *Elements of robotics*. Springer International Publishing Cham, Switzerland, 2018.
- [40] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. The MIT Press, 2004, p. 335.
- [41] H. R. Everett, *Sensors for Mobile Robots*. Taylor & Francis, 1995.
- [50] E. Barreira and R. M. S. F. Almeida, *Infrared Thermography for Building Moisture Inspection*. Springer, 2019.
- [60] K. Bipin, *Robot Operating System Cookbook: Over 70 recipes to help you master advanced ROS concepts*. Packt Publishing, 2018.
- [67] C. Fairchild and T. L. Harman, *ROS Robotics By Example: Learning to control wheeled, limbed, and flying robots using ROS Kinetic Kame*. Packt Publishing, 2017.
- [68] Y. Pyo, H. Cho, R. Jung, and T. Lim, *ROS Robot Programming (EN)*, 1. ed ed. ROBOTIS Co, 2017, p. 487.
- [72] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*. O'Reilly Media, 2015.
- [73] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [75] A. Koubaa, *Robot Operating System (ROS)*. Springer, 2016.
- [79] W. Burger and M. J. Burge, *Digital image processing: an algorithmic introduction using Java*, 2. ed. Springer, 2016.

12.2 Nettbaserte kilder

- [1] K. Utermohlen. "The Future of Technology in the Automotive Industry." Towards data science. <https://towardsdatascience.com/the-future-of-technology-in-the-automotive-industry-11081c8a1999> (accessed 26.03, 2020).
- [2] E24, "Slik fungerer Europas mest effektive lagersystem " in *E24*, ed. E24: E24.
- [3] P. Qvale. "Her plukkes et produkt hvert 6. sekund " tu.no. <https://www.tu.no/artikler/industri-her-plukkes-et-produkt-hvert-6-sekund/235032> (accessed 20.01, 2020).
- [4] FN-Sambandet. "Klimaendringer." FN-Sambandet. <https://www.fn.no/Tema/Klima-og-miljoe/Klimaendringer> (accessed 21.01, 2020).
- [5] Direktoratet for samfunnssikkerhet og beredskap (DSB), "Risikoanalyse av regnflom i by," dsb.no, 978-82-7768-411-6 (PDF) 2016. Accessed: 21.01.2020. [Online]. Available: https://www.dsb.no/globalassets/dokumenter/rapporter/delrapport_-regnflom_2016.pdf

- [6] I. Hanssen-Bauer *et al.*, "Klima i Norge 2100," Miljødirektoratet, 2015. Accessed: 21.01.2020. [Online]. Available: <https://www.miljodirektoratet.no/globalassets/publikasjoner/m406/m406.pdf>
- [7] SINTEF. "Byggforskserien med nyttig anvisning om fuktteori." SINTEF. <https://www.sintef.no/siste-nytt/byggforskserien-med-nyttig-anvisning-om-fuktteori/> (accessed 11.02, 2020).
- [8] *Unngå byggskader: Reduser risikoen for mugg- og råtesopp*, SINTEF Byggforsk, 03-10. [Online]. Available: <https://www.sintef.no/globalassets/upload/artikkel-03-10-byggaktuelt.pdf>.
- [9] Norges offentlige utredninger, "Tilpassing til eit klima i endring," Regjeringen, 978-82-583-1065-2, 2010. Accessed: 21.01.2020. [Online]. Available: <https://www.regjeringen.no/contentassets/01c4638b3f3e4573929f3b375f4731e0/nno/pdfs/nou201020100010000dddpdfs.pdf>
- [10] K. J. Willis, "State of the world's fungi 2018. Report," Kew Publishing, Richmond, UK, 9781842466780, 2018. Accessed: 26.03.2020. [Online]. Available: <https://stateoftheworldsfungi.org/>
- [11] *FOKUS på tre - Soppskader på tre*, Treteknisk, p. 6. [Online]. Available: <http://www.treteknisk.no/resources/filer/publikasjoner/fokus-pa-tre/Fokus-nr-51-1.pdf>. Accessed: 26.03.2020.
- [12] Mycoteam as. "Ekte hussopp - *Serpula lacrymans*." Bygg og bevar. <https://www.byggogbevar.no/pusse-opp/brann/artikler/ekte-hussopp> (accessed 26.03, 2020).
- [13] Norsk Hussopp Forsikring. "Ekte Hussopp." Norsk Hussopp Forsikring. <https://hussoppen.no/skadevoldere/rate/ekte-hussopp/> (accessed 10.05, 2020).
- [15] MaRS. "Product development: The Waterfall methodology (model) in software development." MaRS. <https://learn.marsdd.com/article/product-development-the-waterfall-methodology-model-in-software-development/> (accessed 04.05, 2020).
- [17] DRM Associates. "The Principles of Integrated Product Development." <http://www.npd-solutions.com/principles.html> (accessed 10.05, 2020).
- [21] IEEE Spectrum: Robots. "What is a Robot?" IEEE Spectrum: Robots. <https://robots.ieee.org/learn/> (accessed 20.02, 2020).
- [25] Clear Path Robotics. "Husky - Unmanned Ground Vehicle." Clear Path Robotics. <https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/> (accessed 26.04, 2020).
- [26] M. Köseoğlu, O. M. Çelik, and Ö. Pektaş, "Design of an autonomous mobile robot based on ROS," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 2017: IEEE, 2017, pp. 1-5. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8090199>. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8090199>
- [27] Petrobot. "Robotic Inspection Technology." petrobot. <http://petrobotproject.eu/robotic-inspection-technology/> (accessed 21.02, 2020).
- [28] NASA. "Mars Curiosity Rover." NASA. <https://mars.nasa.gov/msl/home/> (accessed 28.02, 2020).
- [29] National Robotics Engineering Center (NREC). "Sensabot Inspection Robot." National Robotics Engineering Center. <https://www.nrec.ri.cmu.edu/nrec/solutions/energy/sensabot-inspection-robot.html> (accessed 23.01, 2020).
- [30] M. Sahi. "Sensabot is the first inspection robot approved for use by oil and gas companies." Tractica <https://www.tractica.com/robotics/sensabot-is-the-first-inspection-robot-approved-for-use-by-oil-and-gas-companies/> (accessed 23.01, 2020).
- [31] Taurob. "UGV - Taurob Tracker." Taurob. <http://taurob.com/produkte/ugv-taurob-tracker/> (accessed 23.01, 2020).
- [32] S. Robotics. "S3: Electrical Substation Insepction Robot." SMP Robotics. https://smprobotics.com/products_autonomous_ugv/electrical-substation-inspection-robot/ (accessed 21.02, 2020).
- [33] Boston Dynamics. "Spot." Boston Dynamics. <https://www.bostondynamics.com/spot> (accessed 23.01, 2020).

- [34] AkerBP. "Exploring the potential of robotics in the oil and gas industry." AkerBP. <https://www.akerbp.com/exploring-the-potential-of-robotics-in-the-oil-and-gas-industry/> (accessed 21.02, 2020).
- [35] E. Guizzo. "Boston Dynamics' Spot Robot Dog Goes on Sale." IEEE Spectrum <https://spectrum.ieee.org/automaton/robotics/industrial-robots/boston-dynamics-spot-robot-dog-goes-on-sale> (accessed 21.02, 2020).
- [36] T. Melanson. "Augmented Autonomy – The Future of Autonomous Machines." Robotic Industries Association. <https://www.robotics.org/blog-article.cfm/Augmented-Autonomy-The-Future-of-Autonomous-Machines/86> (accessed 24.02, 2020).
- [37] P. Heating&Air. "This is the right way to encapsulate your crawlspace." PV Heating&Air. <https://www.pvhvac.com/blog/this-is-the-right-way-to-encapsulate-your-crawl-space> (accessed 20.04, 2020).
- [38] L. Grimstad. "NMBURobotics/IMRT100/wiki." Lars Grimstad. <https://github.com/NMBURobotics/IMRT100/wiki> (accessed 15.01, 2020).
- [39] L. Grimstad. "Systemoversikt." <https://github.com/NMBURobotics/IMRT100/wiki/electronics> (accessed 15.01, 2020).
- [42] LiDAR-UK. "How does LiDAR work?" <http://www.lidar-uk.com/how-lidar-works/> (accessed 05.02, 2020).
- [43] V. Mazzari. "What is LiDAR technology?" Generation ROBOTS. <https://www.generationrobots.com/blog/en/what-is-lidar-technology/> (accessed 24.03, 2020).
- [44] C. Woodford. "LIDAR." Explain That Stuff. <https://www.explainthatstuff.com/lidar.html> (accessed 20.04, 2020).
- [45] Society of Robots. "Sensors - Robot Encoder." https://www.societyofrobots.com/sensors_encoder.shtml (accessed 05.02, 2020).
- [46] J. Lucas. "What is Infrared?" Live Science. <https://www.livescience.com/50260-infrared-radiation.html> (accessed 18.02, 2020).
- [47] NASA's Imagine the Universe. "The Electromagnetic Spectrum." NASA. <https://imagine.gsfc.nasa.gov/science/toolbox/emspectrum1.html> (accessed 20.04, 2020).
- [48] FLIR. "How does an IR Camera work?" FLIR. <https://www.flir.com/discover/how-does-an-ir-camera-work/> (accessed 18.02, 2020).
- [49] Fluke. "How infrared cameras work." Fluke. <https://www.fluke.com/en-us/learn/best-practices/measurement-basics/thermography/how-infrared-cameras-work> (accessed 20.04, 2020).
- [51] FLIR. "Seeing through fog and rain with a thermal imaging camera." FLIR. https://www.flirmedia.com/MMC/CVS/Tech_Notes/TN_0001_EN.pdf (accessed 18.02, 2020).
- [52] FLIR. "How to detect a water leak with thermal imaging " FLIR. <https://www.flir.com/discover/professional-tools/how-to-detect-a-water-leak-with-thermal-imaging/> (accessed 14.02, 2020).
- [53] N. Gromicko and K. Shepard. "Moisture Meters for Inspectors." International Association of Certified Home Inspectors. <https://www.nachi.org/moisture-meters.htm> (accessed 25.03, 2020).
- [54] R. Smith. "Pin vs. Pinless Moisture Meters Guide." Wagner Meters. <https://www.wagnermeters.com/moisture-meters/wood-info/pin-moisture-meter/> (accessed 18.02, 2020).
- [55] Grainger. "How do moisture meters work?" Grainger. <https://www.grainger.com/know-how/equipment-information/kh-types-of-moisture-meters-qt-346> (accessed 18.02, 2020).
- [56] P. Yost. "Pinless Moisture Meter." Green Building Advisor. <https://www.greenbuildingadvisor.com/article/pinless-moisture-meters> (accessed 20.04, 2020).
- [57] S. Sattel. "How Do Microcontrollers Work?" Autodesk. <https://www.autodesk.com/products/eagle/blog/how-microcontrollers-work/> (accessed 15.01, 2020).
- [58] Society of Robots. "Introduction to Microcontrollers." https://www.societyofrobots.com/microcontroller_tutorial.shtml (accessed 05.02, 2020).

- [59] Electronics Lovers. "An Introduction to Sensors and its Types-Analog and Digital." Electronics. <https://www.electronicshobby.com/sensors> (accessed 20.02, 2020).
- [61] ROS. "About ROS." ROS. <https://www.ros.org/about-ros/> (accessed 16.01, 2020).
- [62] ROS. "Documentation." ROS wiki. <http://wiki.ros.org/> (accessed 16.01, 2020).
- [63] ROS. "ROS History." ROS. <https://www.ros.org/history/> (accessed 16.01, 2020).
- [64] Willow Garage. "Willow Garage History." Willow Garage. <http://www.willowgarage.com/pages/about-us/history> (accessed 29.01, 2020).
- [65] Gazebo. "Why Gazebo?" Gazebo. <http://gazebo.org/> (accessed 29.01, 2020).
- [66] V. Mazzari. "Robotic simulation scenarios with Gazebo and ROS." Generation Robots. <https://www.generationrobots.com/blog/en/robotic-simulation-scenarios-with-gazebo-and-ros/> (accessed 29.01, 2020).
- [69] ROS. "joy." ROS wiki. <http://wiki.ros.org/joy> (accessed 17.04, 2020).
- [70] ROS. "ROS Navigation." ROS. <http://wiki.ros.org/navigation?distro=indigo> (accessed 20.02, 2020).
- [71] ROS. "Setup and Configuration of the Navigation Stack on a Robot." ROS.org. <http://wiki.ros.org/navigation/Tutorials/RobotSetup> (accessed 09.02, 2020).
- [74] ROS. "Costmap_2d." ROS wiki. http://wiki.ros.org/costmap_2d?distro=melodic (accessed 22.03, 2020).
- [76] K. M. Wurm, C. Stachniss, and G. Grisetti, "Bridging the gap between feature- and grid-based SLAM," *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 140-148, 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889009001481>.
- [77] ROS. "gmapping." ROS.org. <http://wiki.ros.org/gmapping> (accessed 01.02, 2020).
- [78] N. Science. "Tour of the Electromagnetic Spectrum - Visible Light." NASA Science. https://science.nasa.gov/ems/09_visiblelight (accessed 31.03, 2020).
- [80] J. E. Thommesen. "Moduler på ringmur. Nye kryperomsanvisninger." SINTEF. <https://www.sintef.no/projectweb/nasjonalt-seminar-om-fuktskader/nasjonalt-fuktseminar-2015/> (accessed 06.04, 2020).
- [81] *12 Things to Consider Before Buying an Infrared Camera*, FLIR. [Online]. Available: https://assets.omega.com/manuals/FLIR12_BOOKLET.pdf. Accessed: 26.02.2020.
- [82] E. Wagner. "5 Things to Look for in a Quality Moisture Meter." wagnermeters. <https://www.wagnermeters.com/moisture-meters/wood-info/5-quality-moisture-meter-traits/> (accessed 26.02, 2020).
- [83] SuperDroidRobots. "How to Build a Robot." SuperDroidRobots. <https://www.superdroidrobots.com/shop/custom.aspx/how-to-build-a-robot/70/> (accessed 09.04, 2020).
- [84] J. A. G. Canales. "GoPro Hero 4." Grabcad. <https://grabcad.com/library/gopro-hero-4-2> (accessed 07.04, 2020).
- [85] B. P. N. "[Component] Robot Arm 3DOF | Lengan Robot 3DOF Dengan Akrilik." Grabcad. <https://grabcad.com/library/component-robot-arm-3dof-lengan-robot-3dof-dengan-akrilik-1> (accessed 18.04, 2020).
- [86] AutominableSnowbot. "Flir Vue." Grabcad. <https://grabcad.com/library/flir-vue-1> (accessed 07.04, 2020).
- [87] Logitech. "Logitech Gamepad F710." Logitech. <https://www.logitechg.com/nb-no/products/gamepads/f710-wireless-gamepad.940-000142.html> (accessed 17.04, 2020).
- [88] Raspberry Pi. "Raspberry Pi 3 Model B+." Raspberry pi. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> (accessed 28.04, 2020).
- [89] GoPro. "GoPro HERO8 " GoPro. <https://gopro.com/en/no/shop/cameras/hero8-black/CHDHX-801-master.html/> (accessed 17.04, 2020).
- [90] I. Farkhatdinov, J.-H. Ryu, and J. Poduraev, "A user study of command strategies for mobile robot teleoperation," *Intelligent Service Robotics*, vol. 2, no. 2, pp. 95-104, 2009. [Online]. Available: <https://link.springer.com/article/10.1007/s11370-009-0036-9>.
- [91] Y. Lu, Z. Xue, G.-S. Xia, and L. Zhang, "A survey on vision-based UAV navigation," *Geospatial information science*, vol. 21, no. 1, pp. 21-32, 2018. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/10095020.2017.1420509>.

- [92] Nerokas Engineering Solutions. "Wheel Encoder Kit." Nerokas. https://store.nerokas.co.ke/index.php?route=product/product&product_id=1227 (accessed 28.04, 2020).
- [93] E. Abent. "Velodyne LiDAR launches Puck Hi-Res 3D sensor for autonomous cars." Slash Gear. <https://www.slashgear.com/velodyne-lidar-launches-puck-hi-res-3d-sensor-for-autonomous-cars-14456055/> (accessed 28.04, 2020).
- [94] Amazon. "HC-SR04 Ultrasonic Distance Rangefinder/Obstacle Detection Module Blue." Amazon. https://www.amazon.co.uk/HC-SR04-Ultrasonic-Distance-Rangefinder-Detection/dp/B0066X9V5K/ref=sr_1_13?dchild=1&keywords=HC-SR04+Ultrasonic+Distance+Rangefinder&qid=1588099277&sr=8-13 (accessed 28.04, 2020).
- [95] FLIR. "FLIR MR59 - ball probe moisture meter with bluetooth." FLIR. <https://www.flir.com/products/mr59/> (accessed 18.02, 2020).
- [96] Optris. "Infrared camera Optris PI 640." Optris. <https://www.optris.global/thermal-imager-optris-pi-640> (accessed 23.04, 2020).
- [97] SINTEF Byggeforsk. "Unngå fuktskader i kryperom." Byggeforsk. https://www.sintef.no/globalassets/upload/byggeforsk/fagartikler/byggfakta_04-2013.pdf (accessed 22.04, 2020).
- [98] R. Stone. "Image Segmentation Using Color Spaces in OpenCV + Python." Real Python. <https://realpython.com/python-opencv-color-spaces/#reader-comments> (accessed 01.03, 2020).
- [99] Thermal Cameras UK. "Thermography for Building Diagnostics." Thermal Cameras UK. <http://www.thermal-camera.co.uk/info/thermography-for-building-diagnostics-19> (accessed 20.03, 2020).
- [100] Thermal Imaging Reviews. "The top 10 best thermal imaging cameras you can buy in 2020 | Reviews and buying guide." Thermal Imaging Reviws. <https://thermalimagingcamerareviews.com/> (accessed 14.02, 2020).

13 VEDLEGG

Vedlegg A Pythonkode for segmentering av fuktige områder i termiske bilder.

```
# -*- coding: utf-8 -*-

"""
This code is based on OpenCV library and the article "Image Segmentation Using
Color Spaces in OpenCV + Python" provided by Rebecca Stone.
"""

def loadImages(path="."):
    """
    Loads files from directory.

    Input: Directory path, by default loads from the directory where its stored
    Output: Files in the folder ending with .jpg
    """
    return [os.path.join(path,file) for file in os.listdir(path) if file.endswith(".jpg")]

def readImages(image):
    """
    Reads the image

    Input: Loaded images file
    Output: Image in BGR-colorspace
    """
    return cv2.imread(image)

def BGR2RGB(BGR_image):
    """
    Converting the image from BGR- to RGB-colorspace.

    Input: BGR-image
    Output: RGB-image
    """
    return cv2.cvtColor(BGR_image, cv2.COLOR_BGR2RGB)

def RGB2HSV(RGB_image):
    """
    Converting the image fra RGB- to HSV-colorspace.

    Input: RGB-image
    Output: HSV-image
    """
    return cv2.cvtColor(RGB_image, cv2.COLOR_RGB2HSV)

def ColorDistribution(RGB_image, HSV_image, image_file):
    """
    Visualization of the images colordistribution in a 3D scatterplot.
    Generating a 3D scatterplot, represent the color distribution within the
    HSV-colorspace.

    Input: RGB-image, HSV-image and the filename
    Output: 3D scatterplot
    """

    ## Setting up the plot
    # Splitting the image into its color component channels.
    h, s, v = cv2.split(HSV_image)
    fig = plt.figure()
    axis = fig.add_subplot(1, 1, 1, projection="3d")
```

```
# Setting up the pixel colors, reshaping from (x, y, z) to (x*y, z)
# where z is 3. color channels, rgb
pixel_colors = RGB_image.reshape((np.shape(RGB_image)[0]*np.shape(RGB_image)[1], 3))
norm = colors.Normalize(vmin=-1., vmax=1.)
norm.autoscale(pixel_colors)
# Normalizing and converting from array to List
pixel_colors = norm(pixel_colors).tolist()

# Building the scatterplot and viewing it.
axis.scatter(h.flatten(), s.flatten(), v.flatten(), facecolors=pixel_colors, marker=".")
axis.set_xlabel("Hue")
axis.set_ylabel("Saturation")
axis.set_zlabel("Value")
plt.title(image_file, fontsize=10, fontweight='bold')

def segment_IR_image(RGB_image, HSV_image, image_file):
    """
    Segmenting the area withing the specified color range, and plotting an
    subplot containing the original image, binary image of the segmented area
    and the binary image with a mask on top.

    Input: RGB-image, HSV-image and the filename
    Output: Result, mask
    """

    # Setting the color range
    black = (0, 200, 0)
    purple = (190, 255, 200)

    # Apply the mask, defined by the color range
    mask = cv2.inRange(HSV_image, black, purple)
    # Impose the mask on top of the binary image
    result = cv2.bitwise_and(RGB_image, RGB_image, mask=mask)

    # Clean up the segmentation using a blur
    result = cv2.GaussianBlur(result, (7, 7), 0)

    return result, mask

def draw_contours(RGB_image_copy, mask):
    """
    Creating contours around the thresholded area.

    Input: A copy of the RGB image, thresholded image.
    Output: Contours around the thresholded area, in the original RGB image
    """

    # Find the contours from the threshold image
    contours, hierarchy = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    # Draw all contours in the image
    image_contour = cv2.drawContours(RGB_image_copy, contours, -1, (0, 255, 0), 2)

    return image_contour
```

```
if __name__ == "__main__" :

    # Importing modules
    import cv2
    import os
    import numpy as np
    import matplotlib.pyplot as plt
    from mpl_toolkits.mplot3d import Axes3D
    from matplotlib import cm
    from matplotlib import colors
    from matplotlib.colors import hsv_to_rgb

    # Loading all the files in the directory
    filename = loadImages()

    # Creating and setting up subplots
    fig, axis = plt.subplots(nrows=len(filename), ncols=4, figsize=(10,10), constrained_layout=
    fig.suptitle('RESULTATER \n', fontsize=30, fontweight='bold')
    axis[0,0].set_title('Originalbilde \n', fontsize=20, fontweight='bold')
    axis[0,1].set_title('Segmentert \n', fontsize=20, fontweight='bold')
    axis[0,2].set_title('Mask \n', fontsize=20, fontweight='bold')
    axis[0,3].set_title('Med kontur \n', fontsize=20, fontweight='bold')

    # Running through all files in the directory
    for i, images in enumerate(filename):
        image_bgr = readImages(images)

        # Converting images to different colorspaces
        image_rgb = BGR2RGB(image_bgr)
        image_rgb_c = image_rgb
        image_hsv = RGB2HSV(image_rgb)

        # Visualization of color distribution of each image, in HSV-colorspace.
        ColorDistribution(image_rgb, image_hsv, images)

        # Segmentation
        results, masks = segment_IR_image(image_rgb, image_hsv, images)
        masks_c = masks
        axis[i,0].imshow(image_rgb)
        axis[i,1].imshow(masks, cmap='gray')
        axis[i,2].imshow(results)
        axis[i,3].imshow(draw_contours(image_rgb_c, masks_c))

    plt.savefig('RESULTATER.png')
```




Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitenskapelige universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway