# Gruppe 2 – ELE306 – Tilleggsnotat

Dette dokumentet tar for seg tilleggsnotat for vår semesteroppgave som ikke ble tatt med i sluttrapport på grunn av begrensninger i rapport lengde.

## Til kapittel 1 - Beskrivelse av problemstilling

Robotarmen skal ha følgende egenskaper:

- Minimum fire frihetsgrader for løfting

- Gripper skal ha egen frihetsgrad.

- Rekkevidde 600 mm i horisontal plan.

- Robot armen skal kunne løfte gripper (1 kg) og last (4 kg).

Mobil base krav:

- Båten er 1000 mm lang, 400 mm bred med egenvekt 30kg, den skal kunne ta 30 kg last.

- Den skal kunne søke autonomt i et område på 10 kvadratmeter.

- Svingradius skal være mindre enn 1 meter.

- To thrustere bak i båten skal brukes for styring, disse er plassert 300 mm fra hverandre.

- Det skal ikke hensyntas bølger, men det er en strøm på 0,1 m/s

Krav til sensor:

- 3D kamera for å detektere objekter i havet. (Eksempel RealSense435)

Med utgangspunktet i kravene angitt skal vi designe en robot som utfører den definerte oppgaven. For den mobile basen har vi utarbeidet kinematiske ligninger, navigeringsstrategi, lokalisering strategi og hvilke sensorer vi mener bør brukes for å utføre disse strategiene. Oppgaven er åpent definert, derfor er det gjort det en del antagelser i gjennomføringen av oppgaven.

Underveis i arbeidet, spesielt under felles presentasjoner ble vi oppmerksom på at parameterne i kravspesifikasjonen er også noe vi kan redefinere gitt vi er innenfor oppgavens kjerne, å plukke opp plastikk fra havet. Basert på dette har vi også utfordret designet angitt i kravspesifikasjonen for å være sikre på at disse er optimale i forhold til oppgaven som skal utføres.

Ettersom oppgaven er utarbeidet for robotikkfaget, vil rapporten fokusere på designaspektene som kan relateres til pensumet i faget. Noen parameter som er relevant for robotdesignet, men utenfor pensum kan bli nevnt og diskutert på et overordnet nivå, uten at designvurderinger og beregninger vil bli utredet.

## Til kapittel 1.2 – Tilnærming til problemstilling

Havet utgjør 70,8% av jordas overflate (REFERANSE1 [hav – Store norske leksikon](https://snl.no/hav) ). Før vi kan utarbeide design, strategier for navigering/lokalisering/kontroll, valg av sensor og design av robot arm er vi nødt til å definere rammene for hvordan og hvor roboten skal operere.

En prinsipiell beslutning er arbeidsområde for roboten. Skal roboten kunne settes i havet hvor som helst og begynne å arbeide, eller skal arbeide i område som er definert før den settes i drift? Når vi gjorde forstudie for oppgaven fant vi at det estimeres at mellom 70 – 80 % av plastikk (i kilo) kommer fra elver. (REFERANSE2 [Where does the plastic in our oceans come from? - Our World in Data](https://ourworldindata.org/ocean-plastics))

Basert på dette konkluderte vi med at roboten skal arbeide i et elveutløp hvor en elv møter havet. Med denne rammen for oppgaven oppnår vi en effektiv løsning av oppgaven. Hvis det søkes i en elv med høy plastforurensing kan vi forvente å finne mye plast i elveutløpet. Roboten vil ha et predefinert område å arbeide i som gir en grunnleggende forutsetning for valg av strategier for kontroll, navigering og lokalisering.

For å ytterligere konkretisere oppgaven roboten skal løse ønsket vi å finne et passende elveutløp som roboten som kan arbeide i. Et hovedpunkt i denne vurderingen er at vi ønsker å finne et lokalt område, og samtidig et område med en grad av plastforurensning som er håndterbar for en robot arm å samle opp. I følge NORCE er Arnaelva med utløp ved Øyrane Torg i Arna, Bergen den tredje mest forurensede elven i Norge (REFERANSE3 [Plast fra industrien forsøpler norske elver - Norce](https://www.norceresearch.no/aktuelt/plast-fra-industrien-forsopler-norske-elver)). Basert på dette har vi valgt å bruke dette elveutløpet som arbeidsområde.

A screen shot of a computer

Description automatically generated

## Til kapittel 2.1 – Notat om kamera

For å kunne plukke opp plastikk fra havet er det behov for en sensor som oppfyller følgende:

- Kan detektere et objekt og gi tilbake en posisjon relativ til sensorens koordin<atsystem. Dette for at roboten skal detektere og navigere mot et plastikk objekt.

- Kan estimere/detektere form og størrelse til et objekt. Gripperen til roboten har begrenset størrelse. Hvis roboten detekterer et objekt som er for stort for gripperen, skal roboten anse objektet som en hindring.

For dette er et Intel RealsENSE D435i kamera passende. Det har en oppgitt rekkevidde på opptil 10 meter. (REFERANSE4[Depth Camera D435 – Intel® RealSense™ Depth and Tracking Cameras](https://www.intelrealsense.com/depth-camera-d435/)) Kombinerer man dette med en god navigasjon strategi for søk vil dette kameraet kunne detektere objekter i arbeidsområdet. Vi har valgt å implementere tre kamera for å gi best mulig dekning og geometri input av plastikk som skal plukkes.

Realsense D435i kamera kommer i tillegg med en integrert Bosch BMI055 IMU som kan brukes som en del av lokaliseringsstrategien for roboten.

Roboten skal detektere et objekt og plukke det opp. Under plukking vil det være viktig at roboten vet både avstand og geometri for delen som skal plukkes opp. Vi har derfor valgt at det skal implementeres tre kamera. Det vil være et kamera som er plasser direkte forut i båten og et kamera på hver side som er rotert med XX antall grader. Med denne vinkelen vil sidekameraene sitt synsfelt dekke frontkamera sitt synsfelt noe som gir dobbel dekning, sidekameraene vil hjelpe med å data på geometrien til plastikken i dybderetningen i forhold til båten. Dette vil gjøre at man ha bedre målinger for å posisjonere gripperen over objektet som skal plukkes.

## Til kapittel 2.2.1 – Kinematikk for mobil base

Retningen til en differensialstyrt mobil robot styres ved hjelp av forskjellen mellom thrustkraften fra thrusterne bak i båten. I et tilfelle hvor roboten går framover (positiv X retning) i forhold til sitt lokale koordinatsystem og thrustkraften fra venstre thruster er høyere enn høyre thruster, vil roboten svinge til venstre. Videre i denne oppgaven vil vi se vekk ifra thrustkraften og bruke den resulterende hastigheten som hver thruster bidrar med.

En differensialstyrt” robot følger en kurve rundt et momentant rotasjonssenter (ICR – Instantaneous Center of Rotation). Høyre thruster har en hastighet langs en kurve , venstre thruster har en hastighet langs en kurve .

## Til kapittel 2.2.3 notat om Lattice navigasjon

En svakhet med Lattice planlegger er at den ikke kan bevege seg diagonalt, som gjør at roboten ikke vil ta den mest effektive banen med tanke på distanse. Denne svakheten er ikke et problem for vårt tilfelle, da vi ønsker at roboten skal dekke mest mulig av søkeområdet, dvs at å kjøre kortest avstand under søk etter plast ikke er noe vi ønsker.

## Til kapittel 2.2.4 - Notat om lokalisering

Ettersom vi har et kart for området som roboten skal arbeides i er en SLAM algoritme som lager et kart ikke nødvendig. Det er sannsynlig at roboten vil arbeide over lang tid mellom lading og avlasting av plastikk. Dead reckoning ved kun bruk av IMU kan gi for høy usikkerhet i lokasjon over tid.

Det er derfor valgt å bruke et Kalman filter for lokalisering av robot som både kombinerer interne og eksterne sensorer. Intern sensor har vi allerede med de integrerte IMU’en i d435i kameraene, det er i tillegg 3 slike kamera om bord.

For ekstern sensor trenger vi høy grad av nøyaktighet ettersom roboten kan potensielt arbeide i trange områder. Det er derfor valgt å bruke RTK GPS (Real-Time Kinematic Global Positioning System) som ekstern sensor.

Den grunnleggende funksjonen av et Kalman filter er at det er først et estimeringssteg hvor roboten estimerer sin posisjon ved neste tidssteg basert på sin nåværende posisjon og input fra kontroller. Andre steget er oppdateringssteget som korrigerer estimatet med ny informasjon, som vil være posisjon fra RKT GPS.

## Til kapittel 2.2.5 - Notat om kontrollstrategi

Følgende kontrollstrategier for mobil base er vurdert:

Vi har følgende kontrollstrategier for den mobile basen, med en kort forklaring av hva de gjør:

**Beveg mot et punkt(Moving to a point)** – Roboten navigerer mot et mål som er et punkt i omgivelsene. Hastigheten er proporsjonal med avstanden til målet og retningen til roboten vil bli styrt slik at den styres mot målet.

**Følg en linje(Following a line)** – Her vil roboten følge en linje. Den vil ha to kontrollere som justerer styring, en som svinger roboten inn mot linjen for å minimere avstanden fra linjen, og en kontroller som justerer kursvinkelen til roboten slik at den er parallell med linjen.

**Følg en bane(Following a trajectory)** - Her er et målet til robot et bevegelig punkt som som følger banen generert av Lattice planlegger Denne kontrollstrategien ligner på «Moving to a point», med den forskjellen at her flyttes punktet.

**Beveg til en positur (Moving to a pose)** – Med tanke på output er denne lik som «Beveg mot et punkt», men her vil roboten ankomme målet i en gitt positur.

Under søk skal roboten følge banen som er generert av Lattice planlegger algoritmen. Denne består av rette strek og kurver. Vi har derfor valgt at roboten skal følge en bane slik at den på best mulig måte følger svingene i banen som Lattice planleggeren genererer.

For plukking av plast skal roboten navigere seg direkte mot målet(plastikk) ved hjelp av bug2. Normalt kan det tenkes at det ikke vil være hindringer mellom båt og objekt som skal plukkes opp, men dette kan ikke utelukkes. Derfor velges også her «Følg en bane» for å kunne dra nytte av at bug2 hensyntar hindringer når den navigerer.

## Notat om 3.1.1 3.1.1 Navigasjon – søk

Under søk skal roboten navigere i en bane som dekker aktuelt søkeområde ved hjelp av Lattice planlegger. I oppgavespesifikasjon er det definert en svingradius på 1 meter som gir oss grid avstand på 1 meter(10 celler).

Når vi skal definere Lattice grid som roboten skal navigere i har vi hensyntatt følgende:

* Hvor fin skal oppløsningen av griddet være ved å justere «grid».
* Definere kostnader for å kjøre rett frem og svinge ved å justere «cost»
* Hvor mye skal hindringer i occupancy griddet «blåses opp» ved å justere ‘inflate’.

Oppløsningen til griddet vil påvirke hvor presist vi kan navigere, hvilken svingradius roboten vil ha når den svinger. Finere oppløsning fører til flere noder som gir tyngre prosessering både når man skal generere griddet og utføre spørringer.

Fra oppgavespesifikasjon er det angitt at båten skal ha en svingradius på under 1 meter. For at roboten skal klare dette må avstanden mellom hver node i Lattice griddet tilsvare en avstand på under 1 meter(10 celler). Ved å sette «grid» og «inflation» i Lattice objekt til 10 har vi 1 meter mellom hver node og 1 meter klaring til hindringer.

Dette gir et grid med 4109 noder:

A red and blue grid with lines and dots

Description automatically generated

<figur>

Under søking mener vi at gridoppløsningen ikke trenger å være så fin, ettersom kameraene dekker et stort areal kan roboten navigere ganske grovt i området og allikevel dekke søkeområdet. Vi har derfor valgt å øke «grid» til 20(2 meter), dette gir et plot med 984 noder som senker behov for prosesseringskraft:

A screen shot of a computer screen

Description automatically generated

Svingradiusen har da økt til for baneplanleggingen under søk. Til gjengjeld har vi er antallet noder 24% av original. Følgende plot viser 6 veipunkt som vil slås sammen og danne inputen til kontroller:

A screenshot of a computer

Description automatically generated

<figur>