



# Forsuaret

## Bacheloroppgave

OPG3301

### Predefinert informasjon

<b>Startdato:</b>	26-11-2021 09:00	<b>Termin:</b>	2021 HØST
<b>Sluttdato:</b>	10-12-2021 20:00	<b>Vurderingsform:</b>	Norsk 6-trinns skala (A-F)
<b>Eksamensform:</b>	Oppgave		
<b>Flowkode:</b>	1627 OPG3301 1 O 2021 HØST		
<b>Intern sensor:</b>	Alexander Sauter		

### Deltaker

<b>Navn:</b>	Henrik Nødset Skåtun
<b>Kandidatnr.:</b>	
<b>FHS-id:</b>	hskatun@mil.no, heskatun@mil.no

### Gruppe

<b>Gruppenavn:</b>	Ulrik og Henrik
<b>Gruppenummer:</b>	13
<b>Andre medlemmer i gruppen:</b>	Ulrik Høivik Lien



# Sjøkrigsskolen

## Bacheloroppgave

Real Time Locating System ved bruk av Ultra Wideband teknologi

Personellkontroll system for marinen

av

Ulrik Høivik Lien og Henrik Nødset Skåtun

Lvert som en del av kravet til graden:

BACHELOR I MILITÆRE STUDIER MED FORDYPNING I LEDELSE – MARINE-  
INGENIØR VÅPENSYSTEMER, ELEKTRONIKK OG DATA

Antall ord:

Innlevert: Desember 2021

**Godkjent for offentlig publisering**



## Publiseringsavtale

### En avtale om elektronisk publisering av bachelor/prosjektoppgave

Kadettenene har opphavsrett til oppgaven, inkludert rettighetene til å publisere den.

Alle oppgaver som oppfyller kravene til publisering vil bli registrert og publisert i Bibsys Brage når kadettenene har godkjent publisering.

Oppgaver som er graderte eller begrenset av en inngått avtale vil ikke bli publisert.

Vi gir herved Sjøkrigsskolen rett til å gjøre denne oppgaven tilgjengelig elektronisk, gratis og uten kostnader	<input type="checkbox"/> Ja	<input type="checkbox"/> Nei
Finnes det en avtale om forsinket eller kun intern publisering? (Utfyllende opplysninger må fylles ut)	<input type="checkbox"/> Ja	<input type="checkbox"/> Nei
Hvis ja: kan oppgaven publiseres elektronisk når embargoperioden utløper?	<input type="checkbox"/> Ja	<input type="checkbox"/> Nei

## Plagiaterklæring

Vi erklærer herved at oppgaven er mitt eget arbeid og med bruk av riktig kildehenvisning.

Vi har ikke nyttet annen hjelp enn det som er beskrevet i oppgaven.

Vi er klar over at brudd på dette vil føre til avvisning av oppgaven.

**Dato: 10 – 12- 2021**

---

Ulrik Høivik Lien

---

Henrik Nødset Skåtun

## Forord

Denne oppgaven har blitt skrevet i perioden oktober til desember 2021 av Ulrik Høivik Lien og Henrik Nødset Skåtun som et krav for utdanning ved FHS Sjøkrigsskolen.

Etter en god diskusjon med Håkon Skåtun om hvilke temaer som kunne være aktuelle for en bacheloroppgave ble det diskutert om posisjoneringsteknologier og hvordan de anvendes. Det var etter denne diskusjonen at kadett Skåtun og kadett Lien fattet interessen for sanntidsposisjonering. Etter et møte hos det norske selskapet Scan Reach ble vi enige om hva vi ville skrive om.

Oppgaven tar for seg en prototype av et «real time locating system» for å kunne øke personellkontrollen om bord på marinens fartøyer. Ønsket er å øke sikkerheten om bord ved at havariansvarlig har oversikt over posisjonen til besetningsmedlemmene under kriser.

Vi ønsker å takke våre medkadetter og lærere for nyttige diskusjoner.

En ekstra stor takk rettes til førsteamanuensis Alexander Sauter for godt samarbeid, diskusjoner og ekstra øyner under bacheloroppgaven.

Bergen, Sjøkrigsskolen, 10-12-2021

Ulrik Høivik Lien

---

Henrik Nødset Skåtun

---

## Sammendrag

Da KNM Helge Ingstad kolliderte med tankskipet Sola TS gikk det hele 23 minutter før alt av personell var gjort redet for og personellkontroll ble varslet over PA-anlegget. På de 23 minuttene ble flere rom meldt tapt og luker ble stemplet igjen. Mangel på personellkontroll gjorde at flere stemplede luker ble gjenåpnet for å dobbeltsjekke at det ikke var noen igjen i rommet. Dagens modell for personellkontroll baserer seg på meldingstjeneste. Besetningsmedlemmene møter opp på sine respektive plasser som står beskrevet i havarirullen og deretter melder seg. Da starter prosessen med å telle opp besetningsmedlemmene og en kan dermed finne ut hvem som mangler. For å finne posisjonen til den savnede benytter besetningen meldingstjeneste for å finne ut sist kjente posisjon og kan deretter begynne søket. Systemet er ressurskrevende og kan ta lang tid. Med et mål om å øke sikkerheten om bord på marinens fartøyer er oppgaven å lage en modell av et real time locating system hvor vi anvender ultra wideband (UWB) teknologi for å kunne spore personellet om bord. Det ble satt fem mål til dette systemet:

1. Systemet skal være skalerbart til å håndtere opp til 130 besetningsmedlemmer.
2. Posisjonsoppløsningen skal være på  $\pm$  en meter.
3. Det skal være enkelt å lese posisjonsinformasjonen og betjene systemet.
4. Posisjonstagen skal kunne bæres av et menneske.
5. Systemet skal ha en mann over bord (MOB) funksjon som forteller om noen har falt over bord.

Systemet består hovedsakelig av en tag per person og tre ankere. Tagen sender ut radiopulser til ankerene som er plassert på bestemte plasser for å kunne regne ut distansene mellom tagen og ankerene. Deretter beregner tagen posisjonen ved å bruke disse tre distansene i en teknikk som heter trilaterasjon. Posisjonsinformasjonen hentes ut fra tagen til en Raspberry Pi som sender denne informasjonen til en database for å kunne visualisere posisjonen. Testingen av systemet har vist at posisjonsnøyaktigheten har vært svært høy. Rekkevidden er begrenset til rundt 26 meter. Brukergrensesnittet gir en enkel oversikt over posisjonene til besetningsmedlemmene. Målene står dermed til samsvar med resultatene. Denne posisjoneringsteknologien kan gi havariorganisasjonen et bedre beslutningsgrunnlag under havarisituasjoner og vil kunne øke personellkontrollen om bord. Det anbefales derfor at marinen vurderer å benytte seg av ett slikt system.

# Innholdsfortegnelse

<b>Forord</b> .....	<b>ii</b>
<b>Sammendrag</b> .....	<b>iii</b>
<b>Innholdsfortegnelse</b> .....	<b>iv</b>
<b>Figurer</b> .....	<b>1</b>
<b>Tabeller/Diagrammer</b> .....	<b>3</b>
<b>Nomenklatur / Forkortelser / Symboler</b> .....	<b>4</b>
<b>1 Innledning</b> .....	<b>5</b>
1.1 Bakgrunn.....	5
1.2 Mål .....	6
1.3 Avgrensninger.....	6
1.4 Struktur .....	7
<b>2 Teori</b> .....	<b>8</b>
2.1 Real Time Locating System.....	8
2.1.0 Metoder for å bestemme avstand .....	9
2.1.1 Trilaterasjon .....	16
2.2 Ultra-Wideband.....	18
2.3 Datakommunikasjon .....	20
2.3.0 TCP / UDP .....	20
2.3.1 UART.....	21
2.3.2 SPI.....	21
2.4 Datateknologi .....	21
2.4.1 HTML / CSS.....	21
2.4.2 PHP .....	22
2.4.3 Javascript / AJAX .....	22
2.4.4 SQL .....	22
2.4.5 Databasesystem.....	22
2.4.6 Apache Web Server .....	23
<b>3 Konseptutvikling</b> .....	<b>24</b>
3.1 Posisjoneringssteknologier .....	24
3.1.0 Bluetooth Low Energy.....	24
3.1.1 Ultra-Wideband.....	25
3.1.2 Passiv Radio Frequency Identification .....	25

3.1.3	Valg av posisjoneringsteknologi: Ultra Wideband.....	26
3.2	Systemdesign.....	26
3.2.0	Systemet i sin helhet.....	26
3.2.1	Hardware benyttet i systemet .....	27
3.2.2	Tagen .....	32
3.2.3	Mellomvare.....	33
3.2.4	Anker .....	35
3.2.5	Applikasjon.....	35
<b>4</b>	<b>Resultater og analyse.....</b>	<b>42</b>
4.1	Avstandsmålinger .....	42
4.2	Test av posisjonsnøyaktighet.....	44
4.2.0	Test 1: Tag plassert over hode.....	45
4.2.1	Test 2: Tag plassert i brystlommen på uniform.....	47
4.2.2	Test 3: Tag plassert under ermet for å simulere et armbånd .....	49
4.2.3	Test 4: Tag båret i brysthøyde med tagen orientert langs XY-planet.....	52
4.2.4	Oppdateringsfrekvens til systemet under test.....	54
<b>5</b>	<b>Drøfting .....</b>	<b>55</b>
5.1	Test og resultater .....	55
5.2	Personellkontrollsystem .....	57
5.3	Datasikkerhet .....	58
5.4	Utstråling .....	58
5.5	Strømforbruk .....	59
5.6	Personvern .....	60
5.7	Videre utvikling av personellkontrollsystemet.....	61
5.8	TDOA til fordel for TWR.....	63
<b>6</b>	<b>Avslutning .....</b>	<b>65</b>
<b>7</b>	<b>Referanser .....</b>	<b>66</b>
<b>8</b>	<b>Vedlegg .....</b>	<b>68</b>
8.1	Tag_to_DB.py .....	68
8.2	PHP.....	69
8.2.1	Nettside for visning av sanntidsposisjonering av personell .....	69
8.2.2	PHP kode for å hente ut posisjonsdata .....	72
8.2.3	PHP kode for å finne mann over bord .....	72
8.2.4	PHP kode for å vise personell om bord.....	73
8.2.5	Nettside for å vise historisk posisjonsdata av personell .....	74



8.2.6	PHP Kode for å hente ut historisk data på valgt personell og tidsrom .....	76
8.2.7	Nettside for å administrere personell – tag koblinger .....	77
8.2.8	PHP kode for å lage ny kobling mellom person og tag .....	81
8.2.9	PHP kode for å avslutte kobling mellom valgt personell og tag.....	82
8.2.10	Nettside for å legge til ny tag eller nytt personell .....	83
8.2.11	PHP kode å legge til nytt personell.....	84
8.2.12	PHP kode for å legge til ny brikke.....	85
8.3	CSS kode for brukergrensesnittet .....	86

## Figurer

Figur 2.1: Illustrasjon av et RTLS med de ulike komponentene (Malik, 2009).....	8
Figur 2.2: Illustrasjon av TOA-protokollen (Malik, 2009).....	10
Figur 2.3: Angle of Arrival med to ankere for å bestemme posisjon. (Malik, 2009) .....	11
Figur 2.4: TDOA med tre ulike situasjoner (Panoradio SDR, 2021) .....	12
Figur 2.5: TDOA med tre parvise hyperbler (Panoradio SDR, 2021).....	13
Figur 2.6: TWR-protokoll (DecaWave, 2021).....	13
Figur 2.7: Beskrivelse av TWR-protokollen (DecaWave, 2021) .....	14
Figur 2.8: Illustrasjon av trilaterasjon (Jónasson & McGuire, 2021) .....	16
Figur 2.9: Trilaterasjon .....	16
Figur 2.10: Effektivitet ved WiFi, Bluetooth, PCS og GPS (Eliko, 2020) .....	18
Figur 3.1: RTLS i sin helhet .....	27
Figur 3.2: DecaWave DWM1001 Development Board (DecaWave, 2021) .....	28
Figur 3.3: Strålmønsteret til DWM1001 (DecaWave, 2021).....	29
Figur 3.4: Skjermdump av DecaWave sin RTLS-applikasjon .....	30
Figur 3.5: Tagen tilkoblet Raspberry Pi og batteribank for strømforsyning. ....	32
Figur 3.6: Kommandoen «lec» kjørt på DWM1001 brikken.....	33
Figur 3.7: Skjermbilde av første del av pythonkoden for mellomvare .....	34
Figur 3.8: Skjermbilde av andre del av pythonkoden for mellomvare .....	34
Figur 3.9: Relasjonsdiagram for databasen.....	35
Figur 3.10: Nettside som viser real-time posisjonering av personell .....	38
Figur 3.11: Nettside som viser historisk posisjonering av valgt personell .....	39
Figur 3.12: Nettside for å opprette og avslutte kobling mellom tag og personell ...	40
Figur 3.13: Nettside for å legge til ny tag eller nytt personell i systemet.....	41
Figur 4.1: Avstandsmålinger av DecaWave DWM1001 Development Board.....	42
Figur 4.2: Oppsettet for testen. Ankerene er plassert på stoler i hvert sitt hjørne av rommet. Firkanten som testpersonen fulgte, er vist med blå markering. ....	44
Figur 4.3: Orientering av brikken under test 4 (DecaWave, 2021) .....	45
Figur 4.4: Test 1 - tag over hode, oppsett med 3 ankere og 1 tag.....	46
Figur 4.5: Test 1 - tag over hode, oppsett med 4 ankere og 1 tag.....	46
Figur 4.6: Test 1 - tag over hode, oppsett med 3 ankere og 2 tager.....	47
Figur 4.7: Test 2 - tag i brystlomme på uniform, oppsett med 4 ankere og 1 tag....	48
Figur 4.8: Test 2 – tag i brystlomme på uniform, oppsett med 4 ankere og 1 tag ...	48
Figur 4.9: Test 2 - tag i brystlomme på uniform, oppsett med 4 ankere og 1 tag....	49
Figur 4.10: Slik ble tagen båret under testen. Taggen er festet med teip til hånden. Ermet ble brettet over tagen når testen ble gjennomført.....	50
Figur 4.11: Test 3 - tag båret som armbånd, oppsett med 4 ankere og 1 tag.....	50
Figur 4.12: Test 3 - tag båret som armbånd, oppsett med 3 ankere og 1 tag.....	51
Figur 4.13: Test 3 - tag båret som armbånd. Oppsett med 3 ankere og 2 tager. Under dette oppsettet ble ikke ermet brettet over tagen som i de andre gjennomføringene. ....	51

Figur 4.14: Test 4 – tag båret i brysthøyde med tagen orientert langs XY-planet. Oppsett med 4 ankere og 1 tag. ....	52
Figur 4.15: Test 4 – tag båret i brysthøyde med tagen orientert langs XY-planet. Oppsett med 3 ankere og 2 tager. ....	53
Figur 4.16: Test 4 – tag båret i brysthøyde med tagen orientert langs XY-planet. Oppsett med 3 ankere og 1 tag. ....	53
Figur 4.17: Oppdateringsfrekvenser til systemet under testing i de forskjellige oppsettene. Test 3 med 3 ankere og 2 tager ble det feil under gjennomføring av testen. Disse verdiene er markert med lys rød farge.....	54
Figur 5.1: Resultat av test 2 og 3 – oppsett med 3 ankere og 1 tag. Tag i brystlomme er vist i blå farge. Tag som armbånd er vist i oransje farge. ....	56
Figur 5.2: RTLS med forskjellig presisjon av lokalisering.....	61
Figur 5.3: Strømforbruk til DWM1001 ved TWR.....	63

## **Tabeller/Diagrammer**

<i>Tabell 1: UWB sine ulike frekvenskanaler (Hsu, 2021) .....</i>	19
<i>Tabell 2: Tekniske spesifikasjoner for Raspberry Pi 3B+ .....</i>	27
<i>Tabell 3: Tekniske spesifikasjoner for DecaWace DWM1001 .....</i>	30
<i>Tabell 4: Gjennomsnitt og standardavvik for avstandstest. ....</i>	43

## Nomenklatur / Forkortelser / Symboler

UWB	Ultra-Wideband
TOA	Time of Arrival
TDOA	Time Difference of Arrival
TOF	Time Of Flight
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
API	Application Programming Interface
UART	Universal Asynchronous Receiver/Transmitter
SQL	Structured Query Language
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indication
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PPF	Hypertext Preprocessor
JS	JavaScript
KNM	Kongelig Norsk Marine
UHF	Ultra High Frequency
GPIO	General Purpose Input Output

# 1 Innledning

## 1.1 Bakgrunn

Klokken 04:01:15 kolliderte fregatten KNM Helge Ingstad med tankskipet Sola TS 8 november 2018 (Statens havarikommisjon, 2021). Under sammenstøtet flerret ankerklyset til Sola TS opp styrbord side på KNM Helge Ingstad som gjorde at de fikk vanninntrenging og dødt skip. Klokken 04:24 ble personellkontroll etablert og varslet over PA-anlegget. Først hele 23 minutter etter sammenstøtet ble alt personell gjort rede for. Mangel på personellkontroll gjorde at flere stemplede luker ble gjenåpnet for å dobbelt-sjekke at det ikke var noen igjen i rommet. I lys av denne kollisjonen og med et økende fokus på sikkerhet i marinen ser en hvor viktig det er med personellkontroll om bord på sjøforsvarets fartøyer, spesielt i krisesituasjoner.

Dagens modell for personellkontroll bruker intern meldingstjeneste hvor underavdelinger har ansvaret for å gjøre rede for sin detalj eller gruppe. Ved en alarm møter besetningsmedlemmene opp på sine respektive lokasjoner i henhold til havarirullen. Deretter begynner man å telle opp hvem som har møtt opp og finner ut hvem som er savnet. Gitt at en person er savnet bruker man intern meldingstjeneste for å finne ut hvor vedkommende var sist. Denne informasjonen går videre opp i hierarkiet slik at skipssjefen kan benytte informasjonen til sin beslutningstaking og vurdering. Dette er et tregt system som krever mye ressurser, som en heller kunne benyttet mer effektivt.

Etter å ha besøkt det norske selskapet Scan Reach fattet vi interessen for deres personellkontrollsystem. Selskapet leverer ett overvåkingssystem hvor besetningen bærer ett BLE-armbånd som kan telle opp hvor mange som er ved de ulike havaristasjonene. Derimot kan ikke systemet deres gi en sanntidsposisjonering av besetningsmedlemmene da nødene, for å telle opp mannskapet, kun var plassert ved havaristasjonene.

Det var etter dette besøket vi lurte på om det var noe tilgjengelig teknologi som kan anvendes for å finne sanntidsposisjonering av personell og bruke dette til et personellkontrollsystem.

## 1.2 Mål

Målsetningen er å lage en modell av et sanntidslokaliserings-system for å se hvilke muligheter og begrensinger systemet vil ha som helhet. Hensikten med systemet er å effektivisere kontrollen på personell om bord på forsvarets fartøyer for å øke sikkerheten og styrke beslutningsgrunnlaget til havariorganisasjonen. Målet deles opp i fem delmål for systemet.

1. Systemet skal være skalerbart. Sjøforsvaret sine fartøyer varierer i størrelse, antall mennesker og byggemateriale. Det er viktig at systemet kan anvendes på samtlige fartøy.
2. Systemet skal kunne gi en posisjon på besetningsmedlemmene med en oppløsning på  $\pm$  en meter.
3. Systemet skal være enkelt å betjene. Brukergrensesnittet skal være intuitivt å forstå, og være driftssikkert.
4. Posisjonstagen skal kunne bæres av et menneske. Det oppfordrer at størrelsen på tagen ikke blir større enn nødvendig og at den ikke hindrer normal drift av fartøyet.
5. Ved mann over bord skal systemet indentifisere at tagen har kommet utenfor en "sikkerhetsgrense" og slå alarm.

## 1.3 Avgrensninger

Oppgaven tar for seg en modell av et real-time locating system ved bruk av ultra wide-band-teknologi for å lage et personellkontrollsystem om bord på marinens fartøyer. Hovedfokuset er å lage et system med et intuitivt og lett brukergrensesnitt som kan vise potensialet et personellkontrollsystem kan ha på et militært fartøy.

1. Oppgaven har ikke som fokus hvordan ruting av data mellom nodene hadde blitt koordinert på et større fartøy med flere rom.
2. Personvern blir drøftet, men oppgaven vil ikke ta stilling til å oppnå en forsvarlig eller omfattende nok løsning på denne problemstillingen.
3. All slags emisjon av elektromagnetiske bølger er noe som må tas nøye i betraktning på et militært fartøy. I hvilken grad et slikt system vil sende ut UWB-pulser og om denne utstrålingen er forsvarlig vil kreve ytterligere undersøkelser som går utenfor

omfanget av oppgaven. Oppgaven vil heller ikke besvare spørsmålet om UWB-strålingen til posisjonstagen kan benyttes i sensitive rom.

4. Oppgaven tar ikke stilling til kryptering av data.
5. Budsjettet til oppgaven er 20.000 NOK
6. Grunnet budsjett og tidsramme foreholder det oppgaven til å bruke hyllevarer som er rimelige og med relativt kort leveringstid.

## **1.4 Struktur**

Oppgaven begynner med relevant teori om real time locating system (RTLS) for å gi en grunnleggende forståelse av hvordan teknologien fungerer og anvendes. Kapittel 3 vil ta for seg konseptutviklingen av ett RTLS med fokus på hvilke teknologier som er tilgjengelig og hvorfor ultra wideband (UWB) har blitt benyttet. Deretter, i kapittel 4, vil oppgaven ta for seg de ulike komponentene som blir brukt og hva slags funksjon de har i systemet. I denne delen vil det være lite vektning på programmeringen, men det vil være tilstrekkelig nok til å gi leseren forståelse av hvordan systemet fungerer. I kapittel 5 vil oppgaven drøfte systemet i sin helhet med sine fordeler og ulemper samt ulike problemstillinger som oppstår med implementasjon av et slikt system. Avslutningsvis vil oppgaven gi sine konklusjoner, etterfulgt av referanser og vedlegg som en finner i henholdsvis kapittel 6, 7 og 8.

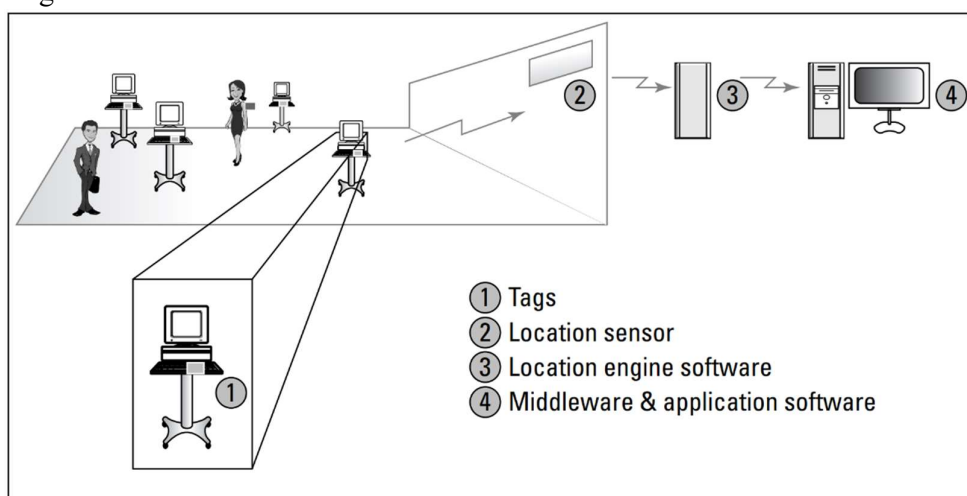


## 2 Teori

Dette kapitlet har til hensikt å plassere arbeidet i et teoretisk rammeverk. Her vil oppgaven ta for seg relevant teori som berører oppgaven.

### 2.1 Real Time Locating System

Real time locating system (RTLS) gir deg muligheten til å gjøre nytte av informasjon om hvor eiendeler eller personell er lokalisert. Grunnet den økte sikkerheten og effektiviseringen et slikt system kan gi, er bruk og utvikling av RTLS-teknologi i høysetet mange steder i verden. Militæret, helsevesenet, lagerhus, førstegangsberedskap, utdanning, nesten enhver virksomhet kan dra nytte av slike systemer. Eksempler på dette er effektiviseringen av et lagerhus ved å vite nøyaktig posisjon på enhver vare. Eller den ekstra sikkerheten et eldresenter oppnår ved å vite posisjon på pasienter med Alzheimers eller demens (Malik, 2009, s. 10). En oversikt over komponentene og funksjonsmåten for et RTLS er gitt i figur 2.1.



**Figur 2.1: Illustrasjon av et RTLS med de ulike komponentene (Malik, 2009)**

RTLS-systemer består normalt av følgende komponenter:

**Tag** - En mobil enhet som er liten nok til å kunne bli festet til en gjenstand eller en person. I figur 2.1 kan en se at tagen er tilknyttet den bærbare datamaskinen. Enheten er utstyrt med lokaliseringsteknologi som sender ut informasjon som kan bli fanget opp av andre sensorer. Tagen kan enten være passiv eller aktiv. Med en aktiv tag så menes det at tagen

selv sender ut informasjonen. Med en passiv tag, for eksempel Radio Frequency Identification (RFID), menes det at tagen ikke selv sender ut med mindre den blir bestrålt av en leser.

**Location sensor** - Enheter som er plassert med en kjent posisjon og fungerer som referansepunkter til tagene. Sensorene tar imot signal fra tagen og kan da utregne avstanden mellom seg og tagen. Videre i oppgaven vil location sensor bli kalt for anker.

**Location engine** – Programvare som kommuniserer med tager og ankere for å regne ut posisjonen til tagene. Her kan mange forskjellige algoritmer benyttes for å kalkulere seg fram til posisjonen. Normalt brukes vei, fart og tid-formlene for å finne avstanden, typisk basert på en gitt bølgefart til signalet. Deretter bruker man sfæriske likninger for å beregne posisjon basert på avstandsinformasjon til tagene relativ til posisjonen til ankere.

**Middleware** – Middleware er et generisk begrep som brukes for å beskrive programvare som kobler sammen to forskjellige applikasjoner, slik at de kan kommunisere med hverandre og utveksle data. I sammenheng med RTLS er middleware den programvaren som bruker informasjonen som er hentet ut fra RTLS-komponentene og prosesserer den slik at det blir tilgjengelig og nyttig for bruk av andre applikasjoner (Malik, 2009, s. 50).

**Application** – Applikasjonen er det som fremstiller informasjonen om posisjon av tagen med relevant informasjon. Dette kan være et brukergrensesnitt på en datamaskin som viser posisjonen til personell om bord på et fartøy.

Hvor nøyaktig sporing et RTLS har vil være avhengig av teknologien som velges og måten avstanden bestemmes. Behovene må så dekkes ved å velge rett teknologi, om det finnes noe som er godt nok, gitt andre krav som begrenser bruken. Noen ganger vil en vite nøyaktig posisjonering av en gjenstand i tre dimensjoner i centimeter-precisjon. I andre situasjoner er det nok å vite hvilket rom gjenstanden befinner seg i.

### 2.1.0 Metoder for å bestemme avstand

For å kunne fastslå en posisjon til en tag finnes det flere teknikker for å beregne avstanden fra tagen til et anker og motsatt. Avstandsmåleteknikkene bruker normalt vei, fart, tid-formlene. Da tagen sender ut elektromagnetiske bølger som forplanter seg med tilnærmet lysets hastighet, vet en dermed farten. Med ulike teknikker kan en finne ut tidsdifferansen mellom utsendelse og mottakelse og en kan dermed regne ut distansen. Her vil de ulike

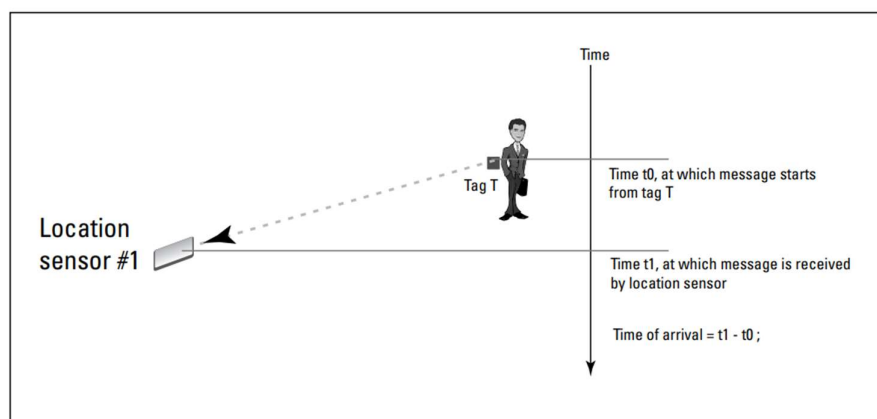
teknikkene bli gjennomgått. Oppgaven vil benytte Two Way Ranging og vil derfor gå dypere i teorien på dette punktet.

### 2.1.0.1 Nærhetsmåling

Nærhetsmåling er en teknikk hvor en måler signalstyrken til tagene med ankere. Signalstyrken blir benyttet for å finne ut hvilken sensor som fanger opp det sterkeste signalet til en tag og en kan da vite hvilket anker tagen er nærmest. Det er viktig å presisere at avstanden ikke blir beregnet, men heller hvilket anker som fanger opp det sterkeste signalet. Posisjonsoppløsningen blir derfor begrenset til antall ankere en har til å fange opp signalet til tagene.

### 2.1.0.2 Time of Arrival

Time of Arrival (TOA) bruker tiden det tar fra signalet blir sendt ut fra tagen til sensoren mottar signalet eller motsatt. For å beregne denne tiden sender tagen en puls med tidsinformasjonen  $t_0$  som vist på figur 2.2. Når signalet ankommer ankeret i tiden  $t_1$  kan en

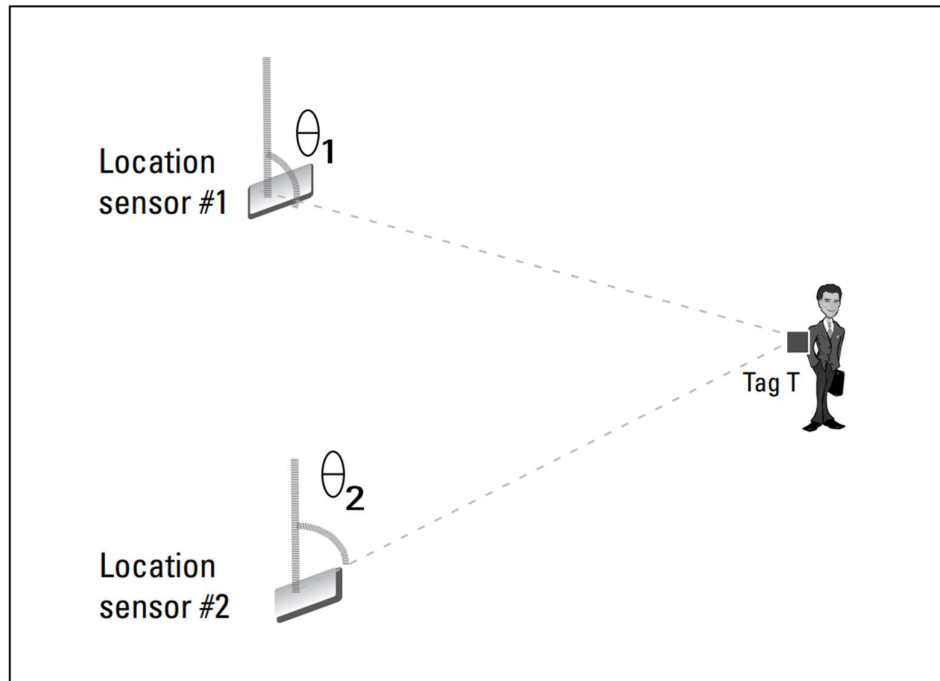


**Figur 2.2: Illustrasjon av TOA-protokollen (Malik, 2009)**

deretter regne ut tidsdifferansen mellom disse to tidene. Dette krever synkroniserte og meget presise klokker både hos tag og anker. Flyvetiden for signalet, også kjent som utbredelsesforsinkelsen, kan bli omgjort til avstand ved å multiplisere tiden med signalets hastighet, ved elektromagnetiske bølger, lysets hastighet. Hvis signalet bruker 0.1 mikrosekunder fra tag til sensor kan en multiplisere dette med lysets hastighet og beregne at tagen er 30 meter fra sensoren.

### 2.1.0.3 Angle of Arrival

Angle of Arrival (AOA) bruker retningssensitive antenner for å kunne beregne retning, med andre ord innfallsvinkel til signalet. Posisjonen blir kalkulert ved å finne skjæringspunktet mellom forplantningsveiene til signalet som vist i figur 2.3



**Figur 2.3: Angle of Arrival med to anker for å bestemme posisjon.  
(Malik, 2009)**

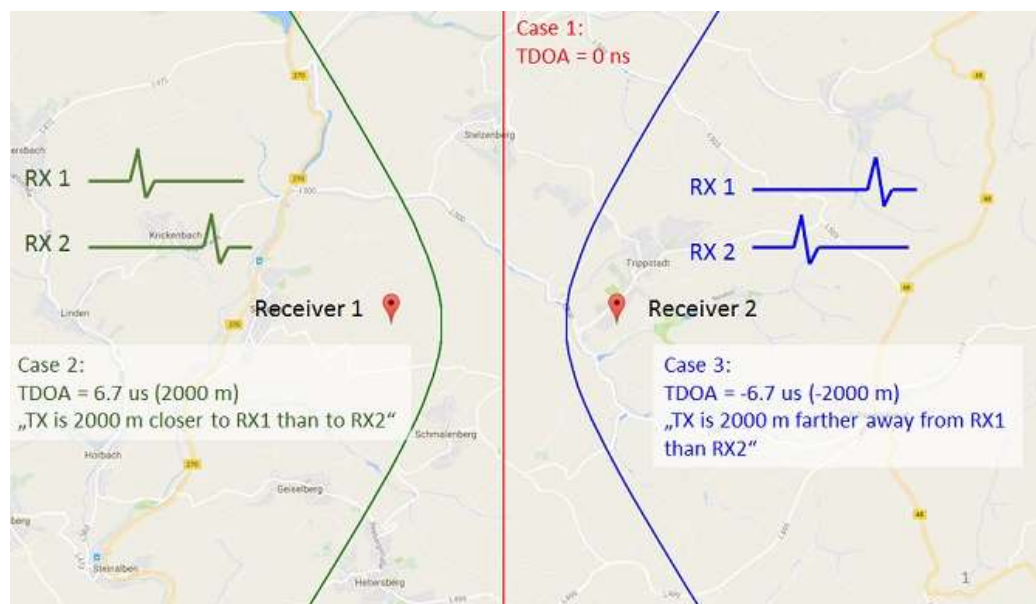
En liten feil i vinkelberegning kan resultere i en stor feil når man skal beregne posisjonen. Fri siktelinje fra tag til sensor er derfor ekstremt viktig for AOA, da flerveis forplantning av signalet kan gi feil vinkel til sensoren og dermed feil posisjon.

### 2.1.0.4 Time Difference of Arrival

Time Difference of Arrival (TDOA) ligner veldig på TOA, men de er ulike på noen områder. Hvor TOA bruker synkroniserte klokker både på tag og anker for å kunne beregne forplantningstiden bruker TDOA tidsdifferansen ved mottakelse av signalet mellom de

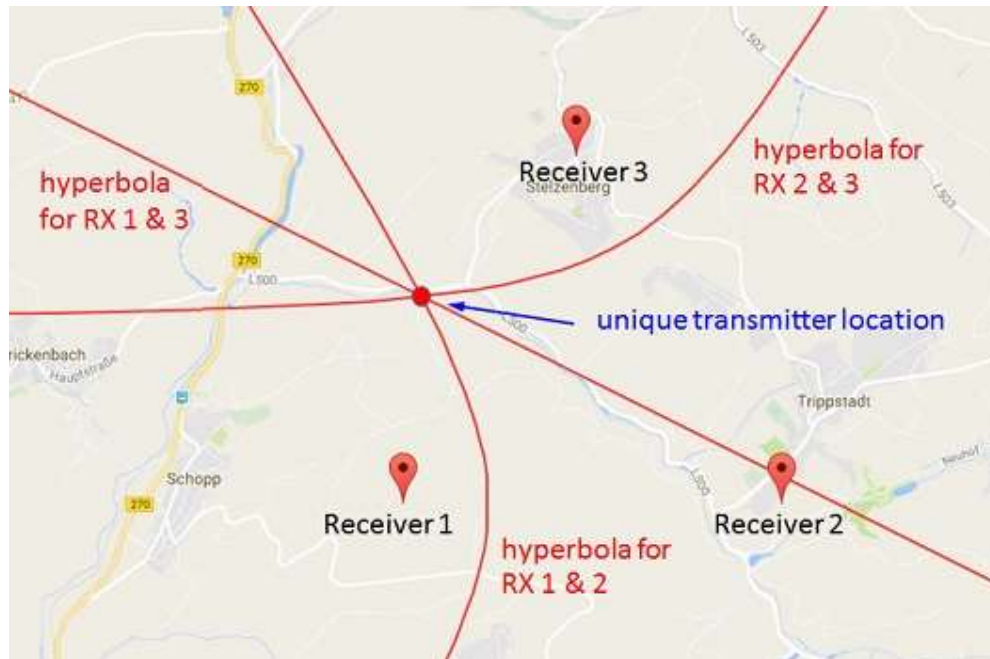
ulike ankerene for å beregne posisjonen. Med andre ord behøver ikke tagen å sende tidsinformasjon, men TDOA krever at ankerene er tidssynkroniserte og svært nøyaktige. Et synkroniseringsavvik på ett nanosekund vil gi et avstandsavvik på ca. 30 centimeter.

Figur 2.4 illustrerer «case 1» hvor tidsdifferansen til de to mottakerne er lik null. Da må signalet komme fra et punkt hvor avstanden mellom de to mottakerne er lik, illustrert som den røde streken. I «case 2», med en positiv tidsdifferanse, hvor «Receiver 1» mottar signalet 6,7 mikrosekunder før «Receiver 2», må signalet komme fra et punkt på hyperbelen illustrert i grønt. Det samme gjelder en negativ tidsdifferanse illustrert i blått.



**Figur 2.4: TDOA med tre ulike situasjoner (Panoradio SDR, 2021)**

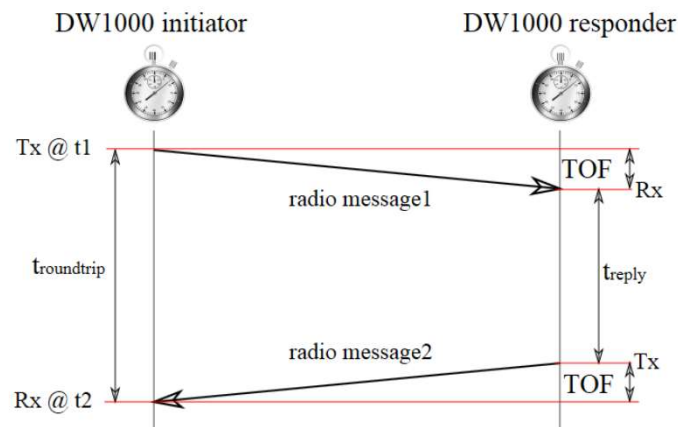
For å kunne fastslå posisjonen på et plan trenger man tre eller flere ankere. Da lager man parvise hyperbler for de tre mottakerne illustrert i figur 2.5. Ideelt vil en da få ett skjæringspunkt mellom hyperblene som vil være posisjonen til tagen.



Figur 2.5: TDOA med tre parvise hyperbler (Panoradio SDR, 2021)

#### 2.1.0.5 Two Way Ranging (TWR)

Two way ranging (TWR) er en metode å bruke TOA til å finne avstanden mellom tag og anker uten å måtte ha tidsynkroniserte klokker. Metoden starter med at initiator sender ut et signal og merker tiden  $t_1$  ved utsendelse. Mottakeren mottar dette signalet og sender en respons tilbake til initiatoren etter en tid  $t_{reply}$ . Når initiator får denne responsen, markerer den tiden med  $t_2$ .



Figur 2.6: TWR-protokoll (DecaWave, 2021).

Tidsdifferansen mellom  $t_2$  og  $t_1$  er tur-retur tiden for prosessen. Ved å vite tiden det tok for mottaker å motta signalet for så å svare  $t_{reply}$  kan en regne ut Time of Flight (TOF) til signalet med formel (2.1).

$$TOF = \frac{t_2 - t_1 - t_{reply}}{2} \quad (2.1)$$

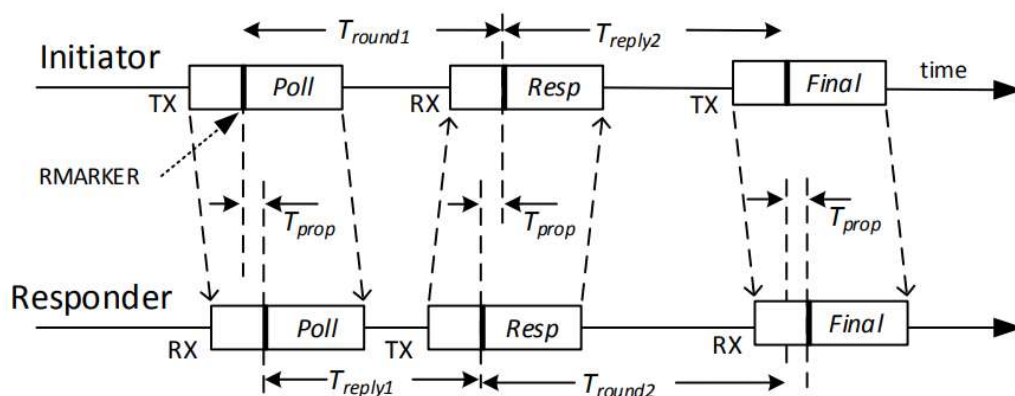
Videre kan distansen bli kalkulert med formelen (2.2).

$$Distance = c * TOF \quad (2.2)$$

Når det gjelder tag-til-anker TWR, er det en rekke feilkilder grunnet klokke- og frekvensdrift. Produsenten til modulene som vil bli brukt videre i oppgaven bruker derfor asymmetrisk dobbeltsidig TWR. Denne metoden reduserer feil grunnet klokke- og frekvensdrift. Figur 2.7 viser denne metoden.

Initiator sender en forespørsel. Mottaker mottar forespørselen og bruker tiden  $T_{reply1}$  på å sende respons.  $T_{round1}$  er tiden det tok for første tur-retur runde. Initiator vil så sende en siste avsluttende melding. Mottaker bruker tiden mellom denne og siden den sendte forrige respons til å finne tur-retur tiden  $T_{round2}$ .  $T_{prop}$  er tiden det tar fra et signal blir sendt til det blir mottatt, altså TOF. For å finne  $T_{prop}$  bruker man formelen (2.3), denne kan så brukes til å finne distansen som vist tidligere.

$$T_{prop} = \frac{T_{rou} * T_{rou} - T_{reply} * T_{reply}}{T_{round1} + T_{round2} + T_{reply1} + T_{re}} \quad (2.3)$$



Figur 2.7: Beskrivelse av TWR-protokollen (DecaWave, 2021)

### 2.1.0.6 Received Signal Strength Indicator

Received Signal Strength Indicator (RSSI) er en teknikk som bruker signalstyrken til å måle avstanden. Elektromagnetiske bølger vil miste energi når de propagerer. Friis' transmisjonsformel (2.1) under forklarer forholdet mellom utsendt effekt, mottatt effekt, bølgelengde og avstand.

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi R}\right)^2 \quad (2.4)$$

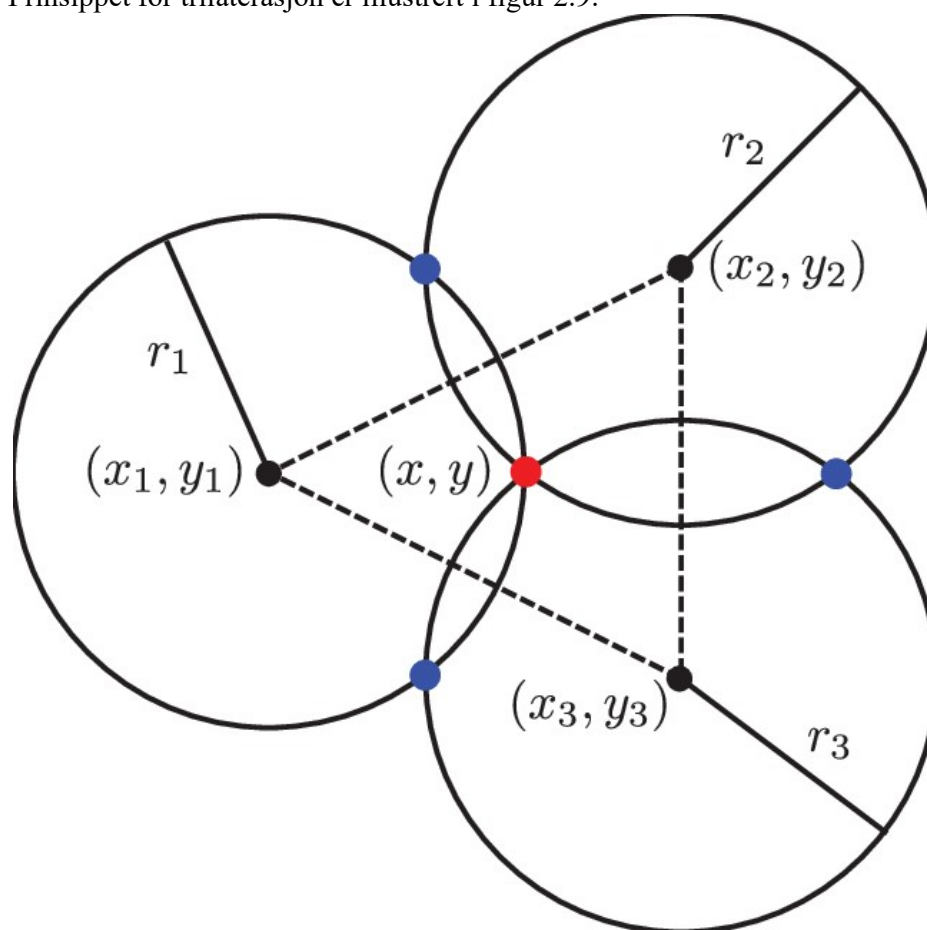
Den målte effekten som mottas er  $P_r$  mens  $P_t$  betegner den utstrålte effekten.  $\lambda$  er bølgelengden og  $R$  er avstanden mellom sender og mottaker.  $G_t$  er gainen fra antennen som sender og  $G_r$  er gainen til den mottakende antennen. Med disse parameterne kan en beregne avstanden fra tag til anker. Målt effekt kan bli påvirket av fysiske hindringer og propagasjonsvei og dermed skape avvik. Det er derfor viktig med siktelinje mellom avsender og mottaker. RSSI blir ofte brukt fordi den er så anvendbar da teknikken kan anvendes på alle typer antenner. Så lenge en har målt utstrålt effekt og muligheten til å måle mottatt effekt kan en beregne avstanden mellom sender og mottaker.



### 2.1.1 Trilaterasjon

Teknikkene over har blitt brukt til å beskrive hvordan en kan finne avstanden mellom en tag og ankere, med unntak av TDOA og AOA. Det er med disse avstandsteknikkene at vi kan beregne posisjonen i neste delkapittel.

Trilaterasjon er en teknikk hvor en kan beregne posisjonen til en enhet basert på tre avstandsmålinger mellom enheten og kjente posisjoner som referansepunkter (Jónasson & McGuire, 2021). Hvis det er flere enn tre målepunkter kaller vi teknikken for multiiterasjon. Prinsippet for trilaterasjon er illustrert i figur 2.9.



Figur 2.8: Illustrasjon av trilaterasjon (Jónasson & McGuire, 2021)

I punktene  $(x_1, y_1)$ ,  $(x_2, y_2)$  og  $(x_3, y_3)$  er det plassert ankere. Hvert anker måler avstanden mellom seg og en tag. Disse avstandene inneholder ingen retningsinformasjon, så avstanden tilsvarer radiusen som beskriver en sirkel der tagen må oppholde seg, illustrert i figur 2.8. Den røde prikken blir skjæringspunktet for alle sirklene, og det er denne posisjonen som skal regnes ut. Når en kun har to avstandsmålinger fra kjente posisjoner til tagen vil en da få en skjæring mellom to sirkler og kan derfor ikke fastslå posisjonen

entydig. Skjæringen ville generelt gi to punkter. En trenger minst tre avstander til kjente posisjoner for å regne ut posisjonen i ett plan. For å beregne ut posisjonen i et rom trenger en minst fire avstandsmålinger.

1. For tre avstandsmålinger kan en sette opp tre likninger til sirklene som representerer avstanden mellom tag og anker. Man ønsker å finne den ukjente posisjonen i  $(x, y)$

som er lik i alle tre likninger.

$$(x - x_1)^2 + (y - y_1)^2 = r_1^2 \quad (2.5)$$

$$(x - x_2)^2 + (y - y_2)^2 = r_2^2 \quad (2.6)$$

$$(x - x_3)^2 + (y - y_3)^2 = r_3^2 \quad (2.7)$$

2. Uttrykkene i parentesene utvides i alle tre likningene.

$$x^2 - 2x_1x + x_1^2 + y^2 - 2y_1y + y_1^2 = r_1^2 \quad (2.8)$$

$$x^2 - 2x_2x + x_2^2 + y^2 - 2y_2y + y_2^2 = r_2^2 \quad (2.9)$$

$$x^2 - 2x_3x + x_3^2 + y^2 - 2y_3y + y_3^2 = r_3^2 \quad (2.10)$$

3. Man subtraherer likning (2.9) fra likning (2.8) som gir likning (2.11). Man subtraherer også likning (2.10) fra likning (2.9) som gir likning (2.12).

$$(-2x_1 + 2x_2)x + (-2y_1 + 2y_2)y = r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 \quad (2.11)$$

$$(-2x_2 + 2x_3)x + (-2y_2 + 2y_3)y = r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \quad (2.12)$$

4. Likningene (2.11) og (2.12) kan så skrives om med verdiene: A, B, C, D, E og F. Vi får da følgende formler.

$$Ax + By = C \quad (2.13)$$

$$Dx + Ey = F \quad (2.14)$$

5. Løsning for koordinatene blir da følgende uttrykk.

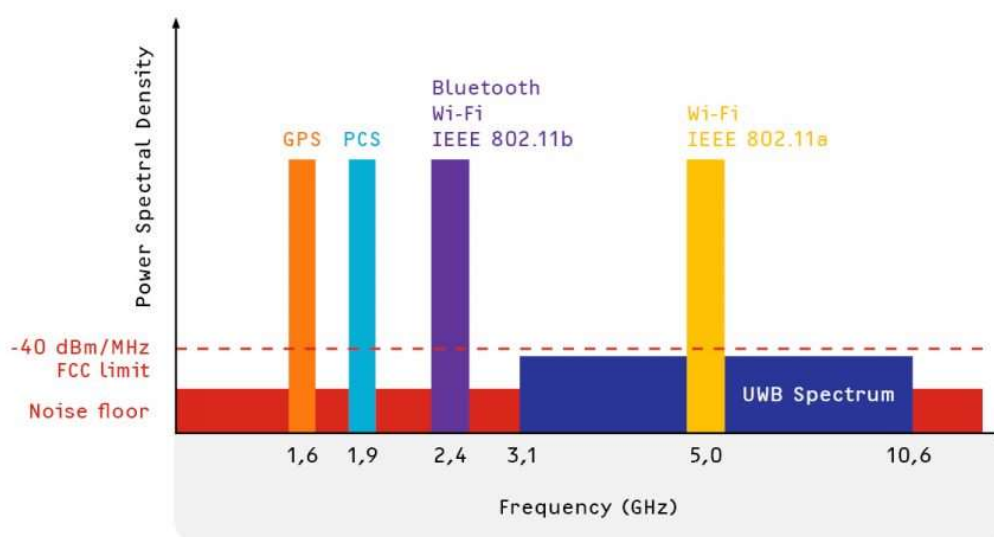
$$x = \frac{CE - FB}{EA - BD} \quad (2.15)$$

$$y = \frac{CD - AF}{BD - A} \quad (2.16)$$

En kan dermed fastslå posisjonen til tagen ved bruk av tre avstandsmålinger til kjente referansepunkter. Det er viktig at ankerene har en god geometrisk figur slik at avviket ikke blir for stort. Med andre ord er det lurt å plassere de langt nok fra hverandre og ikke svært nærme hverandre.

## 2.2 Ultra-Wideband

Definisjonen på UWB er radiobølger med båndbredde som er over 500 Mhz eller mer enn 20% av den aritmetiske senterfrekvensen (Angel, Antonio, David, & Ramon, 2016). Ultrabredbånd har ingen senterfrekvens som «carrier» i forhold til for eksempel FM-signal som brukes til radionettet. UWB er ifølge Institute of Electrical and Electronics Engineers (IEEE) standardisert under 802.15.4-standarden for trådløs kommunikasjon med høy bitrate. Normalt har UWB mellom 11 og 55 Mbit/s for wireless personal area network (WPANs). Frekvensspekteret til UWB ligger normalt mellom 3,1 GHz og 10,6



**Figur 2.10: Effekttetthet ved WiFi, Bluetooth, PCS og GPS (Eliko, 2020)**

GHz som vist i figur 2.10.

Grunnet den store båndbredden og det høye frekvensspekteret har UWB mulighet til å sende korte pulser ned i nanosekunder (Angel, Antonio, David, & Ramon, 2016). Normalt vil UWB sende korte pulser med en lav arbeidssyklus. Med andre ord vil forholdet mellom pulsbredden og tiden mellom to pulser være lite (Zafer, Gezici, & Guvenc, 2008). En annen egenskap med ultrabredbånd er at signalet er mindre påvirket av flerveis forplantning, som er et fenomen der radiosignalet som treffer antennen kommer fra to eller flere forplantningsveier. Den store båndbredden gjør at refleksjonene vil bli forskjellige for de ulike andelene til signalet. Dette vil gi lite overlaging av de elektromagnetiske bølgene ved mottakeren grunnet at refleksjonene er ute av fase, men siktlinje-signalet vil være ett godt signal. Grunnen til at flerveis forplantning oppstår kan være ionosfæriske refleksjoner

eller brytninger, refleksjoner fra vannforekomster eller atmosfærisk dukting. Da UWB begrenser seg i rekkevidde vil det normalt være refleksjoner, da spesielt fra metall, som er den vanligste årsaken til flesveis forplantning. Ved at signalet tar to eller flere veier til antennen gjør det at radiosignalet kan bli faseforskyvet eller at det kan oppstå interferens når signalet ankommer antennen. UWB vil heller ikke være like påvirket av interferens fra andre smalbandssignaler da disse signalene kun påvirker en liten andel av det fulle signalet til UWB som vist i figur 2.10. Effekten til UWB fordeles over et stort frekvensområde og vil ikke interferere, og dermed forstyrre, de bestående frekvensene som brukes til annet.

Figuren 2.10 viser power spectral density til ulike teknologier som GPS, PCS, Bluetooth, Wifi og UWB. En kan se fra grafen at Federal Communications Commission (FCC) har satt en øvre grense på  $-40$  dBm/Mhz uten frekvenstillatelse. Den europeiske ekvivalenten til FCC, European Telecommunications Standard Institute (ETSI) har satt en lavere grense for effektettheten for UWB, men dette gjelder kun ved innendørsbruk. Videre ser vi den røde linjen som er støygrensen. Siden UWB sender med lav effektetthet gjør det at andre frekvenslyttere tolker signalet som støy og vil dermed ikke forstyrre.

*Tabell 1: UWB sine ulike frekvenskanaler (Hsu, 2021)*

Båndgruppe	Kanalnummer	Senterfrekvens (MHz)	Båndbredde (MHz)
1. (Low Band)	0	499.2	499.2
	1	3494.4	499.2
	2	3993.6	499.2
	3	4492.8	499.2
	4	3993.6	1331.2
2. (High Band)	5	6489.6	499.2
	6	6988.8	499.2
	7	6489.6	1081.6
	8	7488.0	499.2
	9	7987.2	499.2
	10	8486.4	499.2

	11	7987.2	1331.2
	12	8985.6	499.2
	13	9484.8	499.2
	14	9484.0	499.2
	15	9484.8	1354.97

## 2.3 Datakommunikasjon

For at kommunikasjonen mellom de ulike komponentene i ett RTLS skal fungere finnes det ulike kommunikasjonsprotokoller for dette.

### 2.3.0 TCP / UDP

For å sende data over nettverk opprettes det en forbindelse mellom avsender og mottaker. Protokollene TCP og UDP oppretter denne forbindelsen og kontrollerer sendingen og mottakelsen av pakker (Øverby, 2021). TCP protokollen deler informasjonen som skal bli sendt opp i pakker med en bestemt størrelse. Pakkene får også et sekvensnummer slik at de kan bli satt sammen i riktig rekkefølge når de kommer frem. TCP danner en pålitelig ende- til ende- forbindelse mellom avsender og mottaker. Etter at avsender har sendt pakke- kene vil mottaker kvittere tilbake hvilke pakker den har mottatt. Dermed vet avsender hvilke pakker den er nødt til å sende på nytt hvis ikke alle pakkene ble mottatt. Slik sikrer TCP at all informasjon man ønsker å sende blir mottatt og satt sammen i riktig rekkefølge, men denne prosessen kan være tidkrevende.

I andre tilfeller er det ikke like viktig at all informasjon kommer frem, men heller at prosessen skjer så fort som mulig. UDP protokollen deler også informasjonen i mindre pakker og sender fortløpende (Rosencrance, 2020). Mottaker kvitterer ikke tilbake hvilke pakker den har mottatt, så om en pakke går tapt vil ikke denne bli sendt på nytt. En slik forbindelse er lurt om en heller ønsker hurtig oppdatering av informasjon enn å sikre at all informasjonen kommer frem. Et godt eksempel er strømming av video, her er det mer

ønskelig at en går glipp av et bilde enn at videoen stopper opp for å vente på informasjonen som må bli sendt på nytt.

Oppgaven skal vise posisjon i ekte tid med hurtige oppdateringer. UDP vil nok være nyttig her ettersom det er viktigere å få inn ny posisjon enn at systemet pauser til den har fått forrige posisjon før den kan vise nyeste igjen.

### **2.3.1 UART**

UART står for universal asynchronous receiver-transmitter og er ikke en protokoll slik som UDP og TCP. UART er en fysisk kobling eller en enkel krets. Hensikten til UART er å sende og motta data hvor hastighet og format er konfigurerbart. Transmisjonen skjer bit for bit, fra den minst signifikante biten, med start frame, til den mest signifikante biten, med stop frame.

### **2.3.2 SPI**

SPI (Serial Peripheral Interface) er en seriell kommunikasjonsprotokoll som bruker master-slave metoden for kommunikasjon. Dette gjør det mulig å enkelt kommunisere mellom forskjellige elektroniske enheter som støtter klokkelede seriestrømmer. Normalt bruker SPI minst fire ledninger for kommunikasjon.

## **2.4 Datateknologi**

For å kunne framstille posisjonsinformasjonen til brukeren av ett RTLS har oppgaven benyttet ett webbasert brukergrensesnitt. Posisjonsinformasjonen vil bli lagret i en database og vil bli visuelt framstilt på en nettside.

### **2.4.1 HTML / CSS**

HyperText Markup Language, eller HTML er standard markeringspråk for formatering av nettsider. (MDN Web Docs, 2021) HTML tar seg av struktureringen og presentering av innholdet på nettsiden. Derimot vil ikke en nettside se presentabel og fullverdig ut med bare HTML. Språket kan derimot bli assistert av teknologier som Cascading Style Sheets (CSS) og skriptspråk som Javascript. CSS blir brukt til å redigere og gi stil til innholdet som HTML presenterer på nettsiden. En kan sammenligne HTML med skjelettet til menneskekroppen. CSS gir utseende og form til skjelettet slik at det til slutt blir et menneske.

## 2.4.2 PHP

PHP er et skriptspråk som brukes som en koblingsløsning for å lage en kobling mellom nettleseren og databasen. PHP tilbyr funksjoner som gjør at en kan aksessere og skrive til databaser samt kjøre SQL-spørringer.

## 2.4.3 Javascript / AJAX

Skriptspråket Javascript er en av kjerneteknologiene i verdensveven i tillegg til HTML og CSS. Javascript gjør det mulig for nettsiden å være dynamisk da den kan kjøre funksjoner som endrer på innhold i både HTML og CSS (Keycdn, 2018). AJAX gjør det mulig å laste inn nytt innhold på nettsiden uten å måtte oppdatere hele siden. Dette gjør det for eksempel mulig å hente ut og vise den nyeste dataen fra en database som kontinuerlig oppdateres. AJAX er ikke en ny teknologi i seg selv, det er heller ikke et nytt programmeringsspråk. I stedet er det eksisterende teknologier som brukes på en ny måte.

## 2.4.4 SQL

Structured Query Language (SQL) er som navnet tilsier et spørrespråk for å kunne hente ut relevant informasjon fra en database. De aller fleste databasesystemer i dag gir støtte for SQL noe som gjør det lett å aksessere og endre på data fra en database på en hensiktsmessig måte. I motsetning til “vanlige” programmeringsspråk har det seg slik at SQL er standarden for spørringer av databaser slik at man har kun ett “språk” å forholde seg til.

## 2.4.5 Databasesystem

Databasesystem har blitt en essensiell del av hverdagen i et moderne samfunn. I løpet av en dag vil de fleste av oss utføre aktiviteter som samhandler med et databasesystem. Det kan for eksempel være å ta ut penger i en minibank eller logge deg inn på et sosialt nettverk. En database er logisk organisering av data som hører sammen og en beskrivelse av disse dataene, designet for å møte informasjonskravene i en organisasjon. Et databasesystem er samlingen av programmer og verktøy som gjør det mulig å opprette, bruke og kontrollere hvem som har tilgang til dataen i en database.

Det finnes en rekke programmer for å opprette og drifte en database, denne oppgaven vil bruke databaseprogrammet MariaDB. MariaDB er en relasjonsdatabase som er fri og åpen under GNU General Public License.

### **2.4.6 Apache Web Server**

Apache Web Server er et gratis "open source" programvare som betyr at kildekoden er allment åpen i den hensikt at "alle" kan ha innsikt i koden og gjøre endringer for utbedringer.



### **3 Konseptutvikling**

I konseptutviklingsfasen vil oppgaven analysere styrkene og svakhetene ved de ulike posisjoneringsteknologiene som er tilgjengelig for å lage ett lokaliseringssystem. Først utredes hvilke posisjonsteknologier som kan være aktuelle for et lokaliseringssystem om bord på marinen fartøyer. Deretter vil det bli utviklet et rammesystem for å teste et slikt system for å kunne møte målene som er satt i kapittel 1.2.

#### **3.1 Posisjoneringsteknologier**

Det finnes en rekke ulike teknologier som gjør det mulig å spore en person eller et objekt. Felles til alle disse teknologiene er at det kreves en form for tag og ankere for å kunne bestemme posisjonen. Tre teknologier står frem som mest aktuelle for posisjonering i kontekst av personsporing: Bluetooth Low Energy, Ultra Wideband, og Radio Frequency Identification. I det følgende vil disse teknologiene bli sammenlignet og én vil bli valgt for bruk i testsystemet i denne oppgaven.

##### **3.1.0 Bluetooth Low Energy**

Bluetooth Low Energy (BLE) er en trådløs nettverksstandard som er designet for Personal Area Network (PAN). Bluetooth var ikke designet for RTLS, men passer bra da Bluetooth har funksjoner for å gjenkjenne nære Bluetooth-naboer og identifisere disse. Som en del av dette kan protokollen grovt bestemme avstanden til sendere i nærheten i form av signalstyrken. Noen av styrkene ved BLE er at tagene krever svært lite energi for å drives. Det betyr at posisjonstagen kan plasseres i ett armbånd og vil ha batteritid på flere år. BLE har også en nøyaktighet på posisjonering i området ca. 1-5 meter. Noen av svakhetene ved BLE er at de bruker RSSI for å beregne avstand og vil dermed være negativt påvirket av et operasjonsområde med mye stål som vil føre til refleksjoner. Da kan det oppstå problemer med flerveis forplantning som kan forårsake feilberegninger av posisjonen. Grunnet Bluetooth sin forespørselsprosedyre vil det være en forsinkelse på ca. 15 – 30 sekunder før en får oppgitt posisjonen til tagen.

### 3.1.1 Ultra-Wideband

Ultra-Wideband defineres som radiobølger med båndbredde høyere enn 500 MHz eller 20 prosent av den aritmetiske senterfrekvensen (Angel, Antonio, David, & Ramon, 2016). Grunnet den store båndbredden vil UWB-pulser være ekstremt korte, noe som gir de en posisjonsnøyaktighet på opp til noen få centimeter. Den store båndbredden, og dermed korte pulsen, gjør også UWB-teknologi seg egnet i høyreflekterende områder hvor det er mye metall, som for eksempel en fregatt. UWB gir svært nøyaktig posisjonering, men teknologien kommer med noen ulemper. Selv om UWB-pulser er korte og brede, og dermed mindre påvirket av flerveis forplantning, kan det oppstå problemer hvis en ikke har siktelinje mellom tag og anker. Grunnet ikke-siktelinje-propagasjon vil mottakeren kun belage seg på refleksjoner hvor signalet være forstyrret. UWB kan også få problemer hvis det er mange tager som sender samtidig grunnet interferens mellom sendingene fra tagene.

### 3.1.2 Passiv Radio Frequency Identification

Ett passivt RFID system består av radiotransponder, radiomottaker og radiosender. Systemet fungerer ved at RFID-tagen mottar en *interrogating wave* fra RFID-leseren, som gir energien som tagen kan svare med og kan deretter sende informasjon tilbake igjen til RFID-leseren. Informasjonen som blir sendt er som oftest identifikasjonsnummeret til RFID-tagen. Passive RFID-tags trenger ikke batteri da de bruker energien fra radiobølgene fra RFID-leseren til å drive seg selv. Det betyr at tagene kan være små og krever ingen vedlikehold. Noen av styrkene ved passiv RFID er at RFID-kort allerede er i bruk om bord på marinens fartøy for adgangskontroll. Nøyaktigheten til ett RFID-system vil være på rundt en meter som gjør det attraktivt om bord på marinens fartøyer. Derimot krever det veldig mange lesere hvis en ønsker høy oppløsning på hele fartøyet. RFID sin oppløsning burde derfor begrenses til seksjoner eller rom i fartøyet ved at RFID-lesere plasseres i knutepunkter hvor besetningen ferdes. Da kan en få kontroll på hvor besetningen befinner seg begrenset til hvilket rom, men ikke hvor i rommet. En annen nedside ved passiv RFID er at RFID sender med Ultra High Frequency (UHF) radiobølger. Dette passer dårlig i ett miljø hvor det er mye stål, som for eksempel en fregatt, grunnet flerveis forplantning.

### 3.1.3 Valg av posisjoneringsteknologi: Ultra Wideband

Da BLE har en oppløsning i området 1-5 meter møter ikke dette oppløsningskravet som ble satt tidligere i oppgaven. Med tidskrevende forespørselsprosedyrer er det uhensiktsmessig at posisjonen fra de forskjellige tagene ikke vises i sanntid. Selv om energibruken til BLE gjør teknologien svært attraktiv vil BLE fortsatt være uegnet til dette systemet. Når det gjelder passiv RFID egner ikke dette seg i operasjonsmiljø med mye stål grunnet refleksjoner. Da RTLS-systemet skal anvendes på marinens fartøyer hvor flesteparten av fartøyene er laget av stål vil dette være negativt. RFID krever også større endringer i infrastrukturen om bord på fartøyene for å kunne implementere systemet. For å kunne få en presis posisjon kreves det langt flere ankere enn ved for eksempel ved UWB. UWB heller derfor som det mest egnede teknologien til å overvåke personell om bord på marinens fartøyer da denne teknologien lover meget høy oppløsning og mindre påvirkning av flerveis forplantning. Dessuten er UWB en spennende og ny teknologi som er verdt å undersøke.

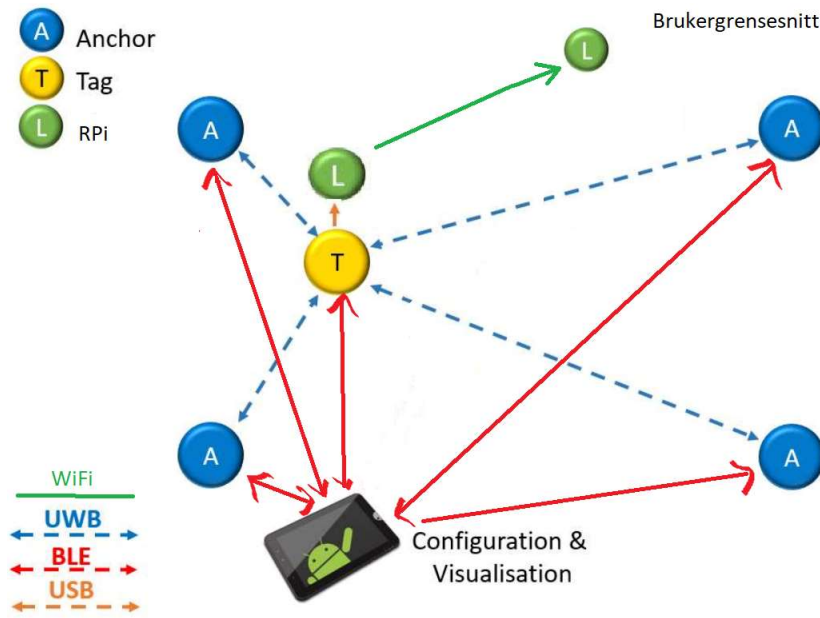
## 3.2 Systemdesign

Systemdesign omhandler om hvordan en bruker elementer i et system som moduler, grensesnitt og data for å oppnå bestemte krav gjennom utforming av et veldrevet system (Hva er systemdesign?, 2021). Da oppgaven tar for seg UWB-teknologi i ett RTLS-system ble det utforsket hvordan UWB-teknologi anvendes i ett slikt system og hvilke komponenter en trenger. Det ble deretter gjort mye arbeid med å finne hylleware-komponenter som passer for bruksområdet og som kan møte målene som er satt i oppgaven.

### 3.2.0 Systemet i sin helhet

Dette avsnittet er ment til å gi leseren en enkel oversikt over hvordan hele systemet er satt opp og kommuniserer med hverandre. Oppgaven vil deretter gå i dybden og forklare hver enkelt komponent og deres funksjoner.

Tagen, markert med T i figur 3.1, har fire ankere som den bruker som referansepunkter for å kunne utregne posisjonen. Da tagen også blir brukt som “location engine” benyttes en Raspberry Pi 3B+ for å overføre posisjonsinformasjonen fra tagen til en server. Denne posisjonsinformasjonen blir sendt over et lokalt nettverk videre til en ny Raspberry Pi.



**Figur 3.1: RTLS i sin helhet**

Denne Raspberry Pien fungerer som server og lagrer posisjonsinformasjonen slik at dette kan fremstilles på en nettside for brukeren.

### 3.2.1 Hardware benyttet i systemet

#### 3.2.1.1 Raspberry Pi model 3 B +

Til både kommunikasjon med tag og til server for databasen samt webserveren benyttes Raspberry Pi. Raspberry Pi er en liten og billig datamaskin som kan kjøre enkle Python- eller Scratch-koder. Den benytter UNIX-versjonen Debian som operativsystem og krever svært lite energi for å operere. Med sine 40 GPIO-pins er den ideell for personer som ønsker å utforske verden innenfor datakommunikasjon og programmering.

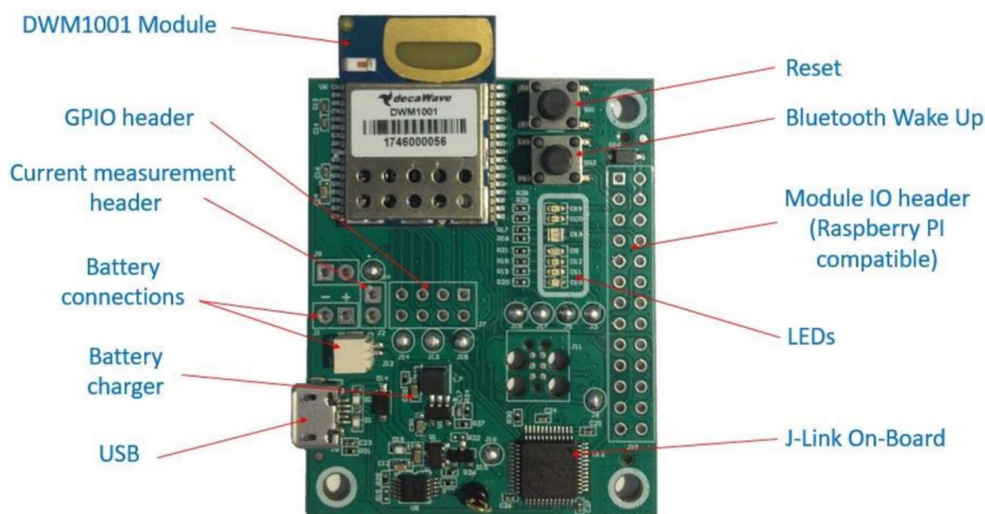
*Tabell 2: Tekniske spesifikasjoner for Raspberry Pi 3B+*

Prossessor	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
RAM	1GB LPDDR2 SDRAM

Nettverkstilknytning	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
Grensesnitt	LAN - RJ-45 CSI - ZIF 4 x USB 2.0 - Type A GPIO - 40-pins IDC DSI - ZIF Sammensatt video- / lydutlinje - 4-pols minijakk HDMI-utgang - Type A PoE HAT-inngang - 4-pins IDC

### 3.2.1.2 DevaWave DWM 1001 Development Board

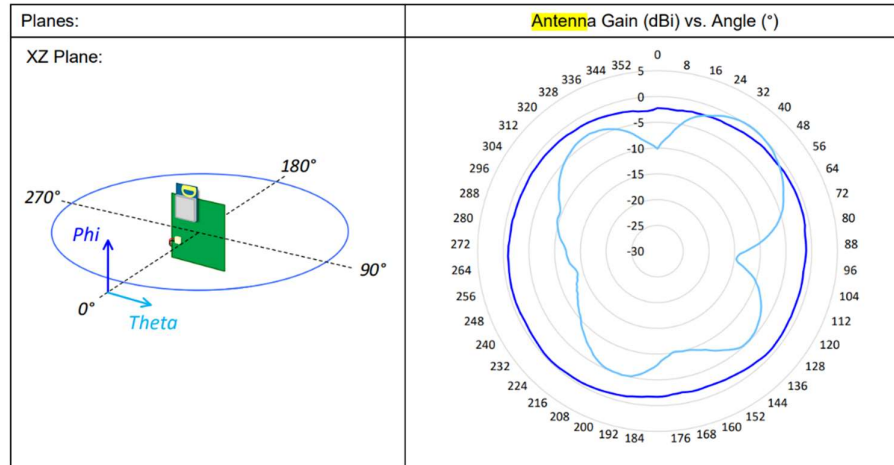
DecaWave sitt utviklerbrett baserer seg på DWM 1001 modulen. Denne modulen er igjen basert på den integrerte kretsen DWM 1000 som er en UWB transceiver. Brettet kan fungere både som tag og anker, men kan også brukes som en nettverksgateway for overføring av posisjonsdata til en server. For å beregne avstander bruker brikken Two Way Ranging som er en avstandsteknikk hvor tag og anker kommuniserer med hverandre for å beregne avstanden. Oppdateringsfrekvens per tag er 10 Hz, det betyr at posisjonsinfor-



**Figur 3.2: DecaWave DWM1001 Development Board (DecaWave, 2021)**

masjonen kan bli oppdatert ti ganger i sekundet. Systemkapasiteten er satt til 150 Hz. Det

betyr at en kan ha 15 tager med en oppdateringsfrekvens på 10 Hz eller 150 tager med en oppdateringsfrekvens på 1 Hz. Utviklerbrettet har også støtte for eksternt batteri, ekstern API via SPI, UART og BLE for konfigurasjon og kontroll. Brettet har også 26 pins for lett tilkobling med Raspberry Pi. Dette gjør det mulig å sende ulike kommandoer til brett-



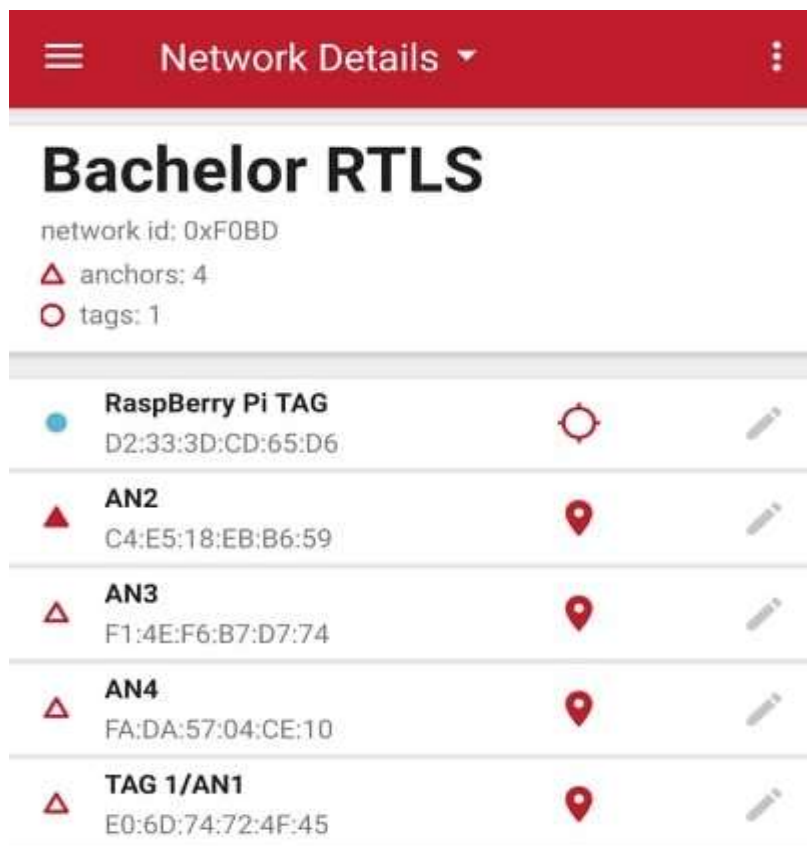
**Figur 3.3: Strålingsmønsteret til DWM1001 (DecaWave, 2021)**

tet via Python-kode.

Figur 3.3 illustrerer strålingsmønsteret til antennen. Antennen er vertikalt polarisert, noe som betyr at den er utviklet for å posisjoner oppreist. Ett rundstrålende strålingsmønster er illustrert i mørkeblått når en annen antenne, som også er vertikalt polarisert, mottar signalet. Hvis antennene er orientert normalt på hverandre vil polariseringen forandre seg. Strålingsmønsteret for denne orienteringen er illustrert i lyseblå farge. I dette tilfellet vil det være visse vinkler som kan begrense rekkevidden og skape posisjonsavvik. For eksempel vil Theta-verdien ved 95 grader være -15 dBi som er tilnærmet 3% av referansesignalet.

DecaWave har utviklet en android-applikasjon for å kunne enklere initiere systemet. Ved første initiering skal utviklerbrettene være i nærheten av telefonen når en åpner opp applikasjonen. Dette gjøres for å tilegne brettene ett eget nettverk. Deretter kan man deklare de ulike brikkene til både tag og anker. Videre har man muligheten til å konfigurere de ulike parameterne som for eksempel oppdateringsfrekvens til tagen og posisjonen til de ulike ankerene. Ett skjermbilde av applikasjonen er illustrert i figur 3.4. Her kan en se de ulike brettene som er tilegnet nettverket hvor trekant betegner anker og sirkler betegner tager. Det skal nevnes at alle disse konfigureringsmulighetene også er gjennomførbare

gjennom ekstern API via USB. Det kan argumenteres at initiell oppsett av systemet er enklere med DecaWave sin android-applikasjon og ble derfor benyttet i denne oppgaven.



Figur 3.4: Skjermdump av DecaWave sin RTLS-applikasjon

DecaWave DWM 1001 sitt utviklerbrett egner seg godt til denne oppgaven da den kommer som et utviklerbrett. Brettet kommer med en rekke støttefunksjoner som gjør det enkelt å komme i gang. Det følger med kildekode samt en android-applikasjon for å lettere konfigurere systemet. I og med at brettet har mulighet for ekstern API via USB kan en tilkoble en Raspberry Pi for å kunne kjøre kommandoer for å hente ut posisjonsinformasjonen til tagen da lokalisering algoritmen kan kjøres direkte på tagen.

Tabell 3: Tekniske spesifikasjoner for DecaWave DWM1001

Avstandspresisjon	Inntil 10 centimeter
-------------------	----------------------

UWB	Kanal 5 (6.5 GHz) IEEE 802.15.4 UWB kompatibel
Datahastighet	6.8 Mbps
Strømforsyning	2.8 V til 3.6 V
Båndbredde	500 MHz
Oppdateringsfrekvens	150 Hz 750 tags @ 0.2 Hz 150 tags @ 1 Hz 15 tags @ 10 Hz etc.
Max oppdateringsfrekvens / tag	10 Hz
Pulsrepetisjonsstid	15.62 nanosekunder
Pulsrepetisjonsfrekvens	64 MHz

DecaWave sitt utviklerbrett kan brukes som både anker og tag og kommer med en rekke støttefunksjoner. Derfor vil denne brikken benyttes i oppgaven. Videre vil oppgaven gjennomgå de ulike komponentene som utgjør et RTLS.



### 3.2.2 Tagen

Tagen er som nevnt tidligere små mobile enheter som festes til objektet eller personen en ønsker å spore. I dette tilfellet vil tagen være tilkoblet en Raspberry Pi som brukes som mellomvare. Et problem med dette er at formfaktoren kan bli for stor til praktiske formål om bord på marinen fartøyer. Ved senere bruk bør man gå over til en mer kompakt løsning, men for utvikling av systemet er formfaktoren håndterlig. En mulighet hadde vært å oppgradere til DecaWave sine DWM 1004C moduler da denne modulen er mye mindre og kunne for eksempel passet inn på et armbånd.



**Figur 3.5: Tagen tilkoblet Raspberry Pi og batteribank for strømforsyning.**

DWM1001 tagen blir brukt til å kommunisere og beregne avstanden mellom seg selv og minst tre ankere. Dette gjøres gjennom TWR-protokollen som forklart i avsnitt 2.1.0.5. Protokollen benytter seg av spørringer og svar til tre eller flere ankere for å kunne beregne avstanden mellom seg og ankerene. Etter at minst tre distanser er beregnet kalkulerer så tagen ut posisjonen ved hjelp av «lec»-kommandoen i «shell» som forklares i kapitlet under.

Tagen er som tidligere nevnt tilkoblet en Raspberry Pi 3B+. Fuksjonen til Raspberry Pi'en er tredelt. For det første sender den Shell-kommandoer til DecaWave-brikken slik at den kan kjøre ulike kommandoer. Dette gjøres med et enkelt python-script. Deretter henter den ut informasjonen om posisjonen til tagen. Til slutt sender den posisjonsinformasjonen videre til en database over et lokalt nettverk slik at posisjonsinformasjonen kan hentes ut

fra et egnet brukergrensesnitt. All kommunikasjon mellom Raspberryen og tagen blir gjort via USB.

### 3.2.3 Mellomvare

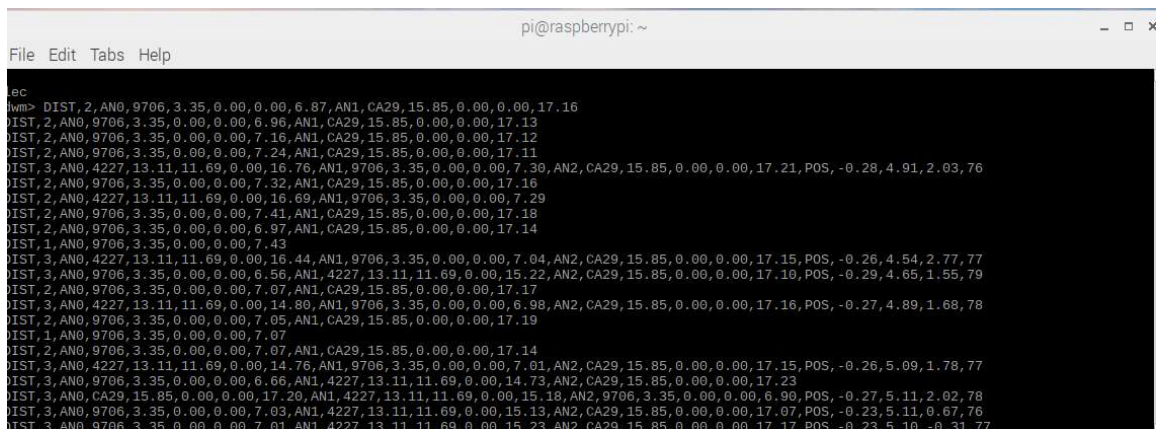
Mellomvare er som nevnt tidligere et generisk begrep som brukes for å beskrive programvare som kobler sammen to forskjellige applikasjoner. I dette tilfellet ønsker vi en kobling mellom tagen og serveren. Dette gjøres med en Raspberry Pi for å kunne kjøre shell-kommandoer til tagen via USB.

Kommandoen som benyttes for å hente ut posisjonsinformasjonen er «lec» og leverer følgende informasjon:

*Lec: "Show distances to ranging anchors and the position if location engine is enabled in CSV format. Sending this command multiple times will turn on/off this functionality." (DecaWave, 2021)*

Følgende format kommer fra utskriften:

*"DIST,Number of Anchors,[AN+Anchor number,Anchor ID,X-Position,Y-Position,Z-Position,Distance to the tag],POS,X-Position of Tag,Y-Position of Tag,Z-Position of Tag "* (DecaWave, 2021)



```
pi@raspberrypi: ~  
File Edit Tabs Help  
lec  
lvm> DIST,2,AN0,9706,3.35,0.00,0.00,6.87,AN1,CA29,15.85,0.00,0.00,17.16  
IST,2,AN0,9706,3.35,0.00,0.00,6.96,AN1,CA29,15.85,0.00,0.00,17.13  
IST,2,AN0,9706,3.35,0.00,0.00,7.16,AN1,CA29,15.85,0.00,0.00,17.12  
IST,2,AN0,9706,3.35,0.00,0.00,7.24,AN1,CA29,15.85,0.00,0.00,17.11  
IST,3,AN0,4227,13.11,11.69,0.00,16.76,AN1,9706,3.35,0.00,0.00,7.30,AN2,CA29,15.85,0.00,0.00,17.21,POS,-0.28,4.91,2.03,76  
IST,2,AN0,9706,3.35,0.00,0.00,7.32,AN1,CA29,15.85,0.00,0.00,17.16  
IST,2,AN0,4227,13.11,11.69,0.00,16.69,AN1,9706,3.35,0.00,0.00,7.29  
IST,2,AN0,9706,3.35,0.00,0.00,7.41,AN1,CA29,15.85,0.00,0.00,17.18  
IST,2,AN0,9706,3.35,0.00,0.00,6.97,AN1,CA29,15.85,0.00,0.00,17.14  
IST,1,AN0,9706,3.35,0.00,0.00,7.43  
IST,3,AN0,4227,13.11,11.69,0.00,16.44,AN1,9706,3.35,0.00,0.00,7.04,AN2,CA29,15.85,0.00,0.00,17.15,POS,-0.26,4.54,2.77,77  
IST,3,AN0,9706,3.35,0.00,0.00,6.56,AN1,4227,13.11,11.69,0.00,15.22,AN2,CA29,15.85,0.00,0.00,17.10,POS,-0.29,4.65,1.55,79  
IST,2,AN0,9706,3.35,0.00,0.00,7.07,AN1,CA29,15.85,0.00,0.00,17.17  
IST,3,AN0,4227,13.11,11.69,0.00,14.80,AN1,9706,3.35,0.00,0.00,6.98,AN2,CA29,15.85,0.00,0.00,17.16,POS,-0.27,4.89,1.68,78  
IST,2,AN0,9706,3.35,0.00,0.00,7.05,AN1,CA29,15.85,0.00,0.00,17.19  
IST,1,AN0,9706,3.35,0.00,0.00,7.07  
IST,2,AN0,9706,3.35,0.00,0.00,7.07,AN1,CA29,15.85,0.00,0.00,17.14  
IST,3,AN0,4227,13.11,11.69,0.00,14.76,AN1,9706,3.35,0.00,0.00,7.01,AN2,CA29,15.85,0.00,0.00,17.15,POS,-0.26,5.09,1.78,77  
IST,3,AN0,9706,3.35,0.00,0.00,6.66,AN1,4227,13.11,11.69,0.00,14.73,AN2,CA29,15.85,0.00,0.00,17.23  
IST,3,AN0,CA29,15.85,0.00,0.00,17.20,AN1,4227,13.11,11.69,0.00,15.18,AN2,9706,3.35,0.00,0.00,6.90,POS,-0.27,5.11,2.02,78  
IST,3,AN0,9706,3.35,0.00,0.00,7.03,AN1,4227,13.11,11.69,0.00,15.13,AN2,CA29,15.85,0.00,0.00,17.07,POS,-0.23,5.11,0.67,76  
IST,3,AN0,9706,3.35,0.00,0.00,7.01,AN1,4227,13.11,11.69,0.00,15.23,AN2,CA29,15.85,0.00,0.00,17.17,POS,-0.23,5.10,-0.31,77
```

Figur 3.6: Kommandoen «lec» kjørt på DWM1001 brikken.

Med denne kommandoen kan vi hente ut posisjonen som er beregnet av tagen. I og med at utskriften er i CSV-format kan en lett parse utkriften for å hente ut informasjonen om posisjonen til tagen.

Python3 blir benyttet for å sende kommandoer til tagen samt uthenting av posisjonsinformasjon. I figur 3.8 kan en se et skjermbilde av første del av koden. Hensikten med denne delen av pythonkoden er å koble seg til databasen og koble seg til tagen via USB. For å kunne koble seg til tagen sin shell må en sende to bits i løpet av ett sekund (DecaWave, 2021). Linje 28 sender to “enter” kommandoer slik at en får tilgang på shell.

```

1  ##Kode for å kjøre kommandoer på DecaWave DWM1001 for å skrive til en database
2
3  import serial
4  import time
5  import datetime
6  import mysql.connector as mariadb
7  import sys
8
9
10 #informasjon om databasen
11 conn = mariadb.connect(
12     user="personellkontroll",
13     password="UlrikHenrik",
14     host="192.168.1.30",
15     port=3306,
16     database = "exampledb")
17 cur = conn.cursor()
18
19 #Hva skal oppdateres i databasen
20 tid_naa=datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
21 sql_hist = "INSERT INTO posisjon_historisk (ID_brikke, X_Koordinat, Y_Koordinat, tid) values ('DECAE9E29814162D',%s,%s,'" + tid_naa + "'"
22 sql_now = "UPDATE posisjon_naa SET X_Koordinat = %s, Y_Koordinat= %s WHERE id_brikke = 'DECAE9E29814162D'"
23
24
25 #tilkobling til DWM1001 via USB
26 DWM=serial.Serial(port="/dev/ttyACM0", baudrate=115200)
27 print("Connected to " +DWM.name)
28 DWM.write("\r\n".encode())
29 time.sleep(1)
30

```

**Figur 3.7: Skjermbilde av første del av pythonkoden for mellomvare**

Deretter skriver kommandoen “lec” for å kunne hente ut posisjonsinformasjonen som nevnt tidligere. Deretter blir x- og y-koordinatene parset slik at databasen kan bli oppdatert med sine respektive SQL-spørringer som står i linje 21 og 22 i figur 3.8.

```

32 #Skriver "lec" til brikken for å hente ut posisjonen ved å parse CVS-formatet som blir printet ut
33 DWM.write("lec\r\n".encode())
34 time.sleep(1)
35 while True:
36     try:
37         line=DWM.readline()
38         if(line):
39             if len(line)>=117:
40                 #Parser informasjonen om posisjonen til tagen
41                 parse=line.decode().split(",")
42                 x_pos=parse[parse.index("POS")+1]
43                 y_pos=parse[parse.index("POS")+2]
44                 quality=parse[parse.index("POS")+4]
45                 val = (x_pos,y_pos)
46
47                 #Legger til historisk data i databasen
48                 cur.execute(sql_hist, val)
49                 conn.commit()
50                 #oppdaterer posisjonen nå
51                 cur.execute(sql_now, val)
52                 conn.commit()
53
54                 #skriver ut i prompt
55                 print(datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S"),(" ",x_pos," ", "y_pos,") qt: ", quality)
56             else:
57                 print("Position not calculated: ",line.decode())
58
59     except mariadb.Error as e:
60         print("Error: ",e,"")
61         break
62 DWM.write("\r\n".encode())
63 DWM.close()

```

**Figur 3.8: Skjermbilde av andre del av pythonkoden for mellomvare**

### 3.2.4 Anker

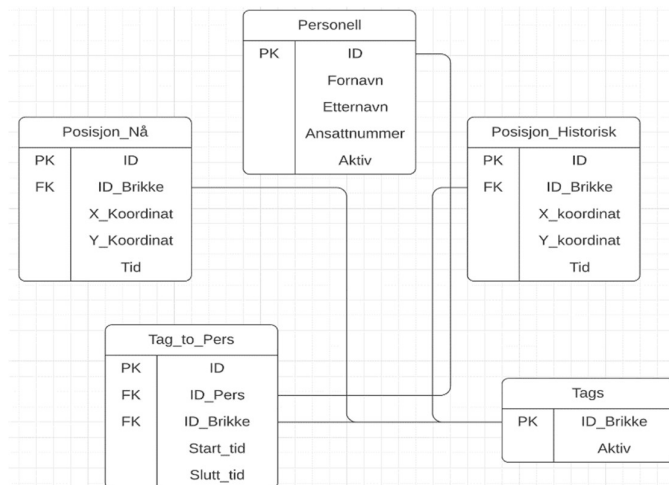
Ankerene i systemet fungerer som referansepunkter en kan bruke til å finne posisjonen til et objekt i rommet. Ønsker en å finne nøyaktig posisjon i tre dimensjoner er en nødt til å ha fire eller flere referansepunkter. Systemet til oppgaven begrenser seg til å finne posisjon i ett plan da det ikke er særlig nødvendig å vite hvor høyt eller lavt en person i rommet befinner seg. Ankerene er tilkoblet strømforsyning, men har også mulighet for batteri for redundans. Dette er svært gunstig da det som oftest er i krisesituasjoner en ønsker å benytte seg av overvåkingssystemet. Ved å koble ankerene til et batteri som kan brukes ved dødt skip vil systemet fortsatt fungere ved strømbrudd.

### 3.2.5 Applikasjon

Serveren for applikasjonen bruker en Raspberry Pi 3B+ som er tilkoblet det lokale nettverket. Den har i oppgave å drifte databasen samt nettsiden. Rpi-en har relativt lav regnekraft, men er tilstrekkelig for denne oppgaven. Oppsettet kan gjøres likt senere, bare på en mer kraftig datamaskin for ekstra regnekraft. Som nevnt tidligere vil serveren benytte MariaDB som databaseprogram. Da det er åpen kildekode, robust, skalerbart og enkelt å sette opp passer det ypperlig for denne oppgaven.

#### 3.2.5.1 Databasesystemet

Databasen i oppgaven inneholder totalt fem tabeller med ulike funksjoner. Figur 3.10 viser databasen med de ulike tabellene.



**Figur 3.9: Relasjonsdiagram for databasen**

Databaser er normalt oppbygget med ulike tabeller med ulike relasjoner. For at de ulike radene i tabellene skal kunne identifiseres, har man en attributt som er unik for hver rad. En praksis på dette er et heltall som øker for hver rad. Denne attributten definerer vi som primærnøkkel. Dette er illustrert som “PK” i figur 3.10. Relasjonene gjør at en kan knytte rader i én tabell til en annen for å sammensette relaterte rader. Relasjonene blir illustrert i figur 3.10 som “FK” som er en forkortelse for “foreign key”. For eksempel benyttes primærnøkkel i tabellen *Tags* som fremmednøkkel i tabellen *Posisjon\_Nå*.

Tabellen *Personell* har som oppgave å holde oversikt over besetningsmedlemmene. Her ble det valgt å sette et økende heltall “ID” som primærnøkkel da ansattnumre kan endre seg etter tid. I brukergrensesnittet har en mulighet til å legge til nytt personell. Når en lager en kobling mellom en person og en tag vil *aktiv* attributtet bli satt til TRUE. Når en person ikke har noen kobling, vil *aktiv* være satt til NULL. Funksjonen til *aktiv* er for å enklere kunne skille mellom personell som har en aktiv kobling med tag og personell som ikke har det.

Tabellen *Tags* inneholder identifikasjonen til de ulike tagene. Attributtet *Aktiv* forteller systemet om tagen er i bruk og tilknyttet et besetningsmedlem. En kan også bruke det som en sikkerhetsfunksjon ved å se på hvilke tags som er aktive, men som ikke sender eller blir fanget opp. Tabellen har blitt begrenset til de nødvendige attributtene for et RTLS, men her kan informasjon som for eksempel mac-adresse, serienummer og produsent legges inn.

Tabellen *Posisjon\_Nå* er en tabell hvor sist kjente posisjon til tagene oppdateres kontinuerlig. Oppgaven skiller mellom historisk data og sanntidsdata grunnet treghet under spørringer. Ved å oppdatere posisjonen ved bruk av UPDATE-funksjonen i stedet for INSERT vil tabellen bli mindre og vil dermed gjøre at spørringene går fortere. Med andre vil sist kjente posisjon oppdateres i *Posisjon\_Nå*.

Tabellen *Posisjon\_Historisk* er en tabell for å kunne se historisk data fra tagene. Her lagres posisjonen til de ulike tagene med tidsmerking slik at en kan gå tilbake i tid for å se hvilken tag som har vært hvor. Det er verdt å nevne at denne tabellen kan bli for stor og dermed uhensiktsmessig. Ved å ha 130 tager med en oppdateringsfrekvens på 1 Hz per tag vil en i løpet av ett døgn ha over 11 millioner rader med posisjonsdata. En løsning på dette kan være å slette den eldste informasjonen etter en gitt tid.

Tabellen *Tag\_to\_Pers* er ansvarlig for å knytte en tag opp mot ett besetningsmedlem. Med denne tabellen kan man da finne hvilken tag som har vært tilknyttet hvem. Tabellen blir også brukt som en sikkerhetsfunksjon da identifikasjonen til tagen kan anses som anonymt, men navn kan medføre personvernsproblemer, noe oppgaven vil drøfte senere i kapittelet *drøfting*.

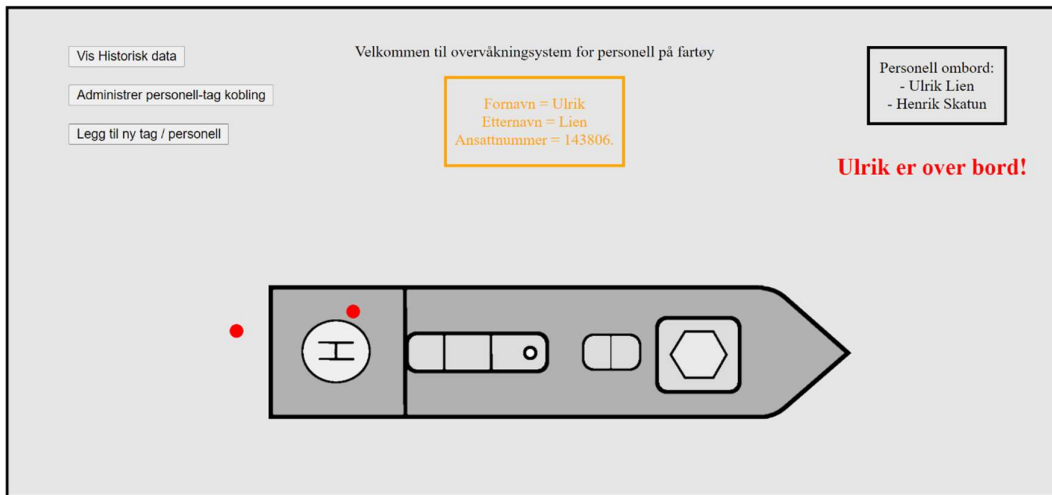
### 3.2.5.2 Brukergrensesnittet

Brukergrensesnittet består av 11 PHP-filer og én CSS-fil som danner tre nettsider. Disse nettsidene utfører en rekke funksjoner, blant annet sanntidsfremstilling av posisjonsdata på registrerte tager, mulighet for å se på historisk posisjonsdata av personellet, opprettelse av nye tag-personell-koblinger. Nettsidene vil også kunne slette koblinger, registrere nytt personell eller tag, se hvem som er på fartøyet og gi varslings ved mann over bord.

At brukergrensesnittet til systemet baserer seg på PHP handler i stor grad om tilgjengelighet. Enhver datamaskin med tilgang på det lokale nettverket og en internettleser vil kunne bruke systemet. PHP er relativt enkelt og godt dokumentert, noe som gjør at utviklingen av produktet tar mye kortere tid. Derimot er PHP av samme grunn mer utsatt for angrep da det er mange som har kunnskap til programmet (Roznovsky, 2021).

Hovedsiden for nettsiden viser sanntidsposisjon til personellet om bord (se figur 3.11). Hver røde prikk som blir vist på plantegningen av fartøyet representerer en tag som er tilknyttet en person i systemet. Ønsker man mer informasjon om en av de røde prikkene kan en dobbeltrykke på sirkelen og det vil da bli presentert en oransje firkant med ytterligere informasjon om personen. En får også generell informasjon om hvem som er om bord til høyre på siden. Ved en MOB-situasjon vil brukergrensesnittet varsle om dette i rød skrift som vist til høyre i figur 3.11.

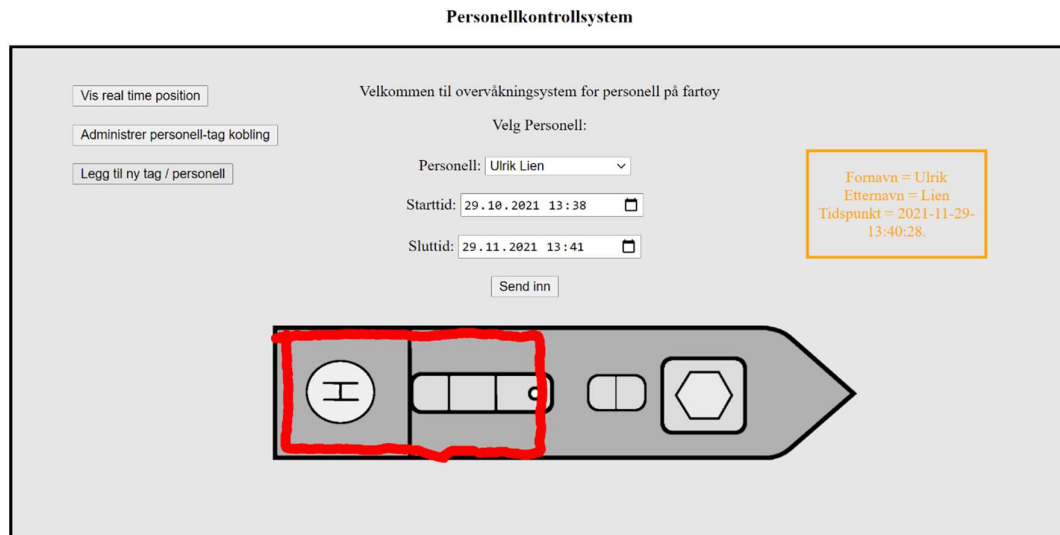
### Personellkontrollsystem



**Figur 3.10: Nettside som viser real-time posisjonering av personell**

Figur 3.12 viser en skjermdump av hvordan nettsiden viser historisk data. Brukeren velger besetningsmedlem og tidsrommet en ønsker å hente ut data på og trykker deretter "Send inn". Alle lagrede posisjoneringer på personen i det gitte tidsrommet vil så bli vist på plantegningen (røde linjer i figur 3.12). En har også mulighet til å hente ut ytterligere informasjon om en posisjonering ved å trykke på de røde sirklene. Da vil informasjon om tidspunktet for posisjoneringen bli vist i en oransje informasjonsboks til høyre på siden. Historisk posisjonsdata kan være nyttig til å understøtte i debriefing etter en øvelse eller

etter hendelser for å gjennomgå beslutninger som er tatt og utvikle og forbedre prosedyrer.



**Figur 3.11: Nettside som viser historisk posisjonering av valgt personell**

Figur 3.13 viser en skjermdump av nettsiden for å administrere koblinger mellom personell og tags. Her har en mulighet til å avslutte aktive koblinger samt å danne nye koblinger. Hvis en ønsker å lage ny kobling vil bare personell og tager som ikke har aktive koblinger bli presentert. Slik sikrer systemet at det ikke blir lagt til flere enn én person på én tag eller flere tager på en person. Hvis en ønsker å avslutte en kobling kan en gå gjennom alle aktive koblinger i nedtrekksmenyen og velge koblingen man vil avslutte. Når en



kobling blir avsluttet blir tidspunktet for dette lagt til i *slutt\_tid* i *tag\_to\_pers*-tabellen slik at systemet vet at koblingen er avsluttet.

**Personellkontrollsystem**

Velg personell og tag for å sette opp en ny kobling til personellkontrollsystemet

Personell:

Tag:

Velg kobling du vil avslutte:

Kobling:

**Figur 3.12: Nettside for å opprette og avslutte kobling mellom tag og personell**

Figur 3.14 viser en skjermdump av nettsiden hvor brukeren kan legge til nytt personell eller tag i systemet. Ønsker brukeren å legge til en ny tag, fyller en inn identifikasjonsnummeret til brikken i innskrivningsfeltet og trykker «send inn». Ny tag vil så bli lagt til

i *tags* tabellen med attributtet *aktiv* satt til NULL. Brukeren kan også legge til nytt personell ved å fylle inn fornavn, etternavn og ansattnummer for så å trykke «send inn». Personellet blir lagt til i tabellen *personell* i databasen.

**Personellkontrollsystem**

Legg til ny Tag

Id til brikke:

Legg til nytt Personell

Fornavn:

Etternavn:

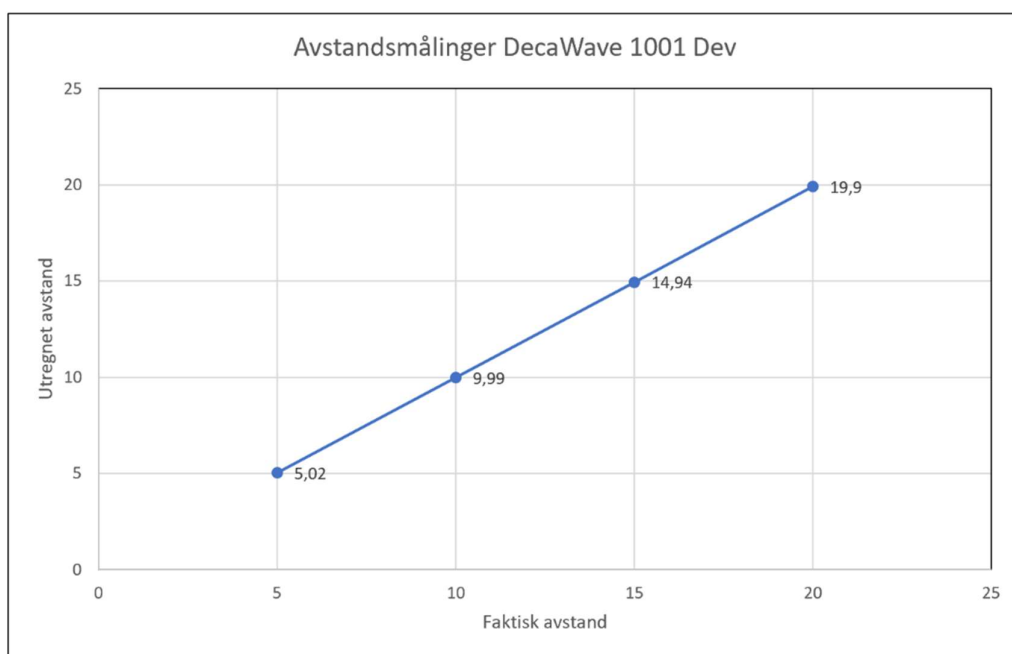
Ansattnummer:

**Figur 3.13: Nettside for å legge til ny tag eller nytt personell i systemet**

## 4 Resultater og analyse

For å undersøke om det utviklede systemet oppnår kravene, ble det gjennomført noen tester. Disse skal først kartlegge hvor nøyaktig avstandsmålingene mellom anker og tag er ved økende avstander, for så å se på stedsnøyaktigheten til systemet ved ulike konfigurasjoner.

### 4.1 Avstandsmålinger



**Figur 4.1: Avstandsmålinger av DecaWave DWM1001 Development Board.**

For å kartlegge hvor nøyaktig avstandsberegningene til systemet er, ble det gjennomført målinger for å estimere avviket til systemet. Eksperimentet tok ikke for seg posisjonsnøyaktigheten til systemet, men avstandsnøyaktigheten mellom tag og ett anker. Målingene ble gjennomført i ett stort rom på ca. 25 \* 15 meter illustrert i figur 4.2. Rommets vegger består av betong, slik at målingene skulle bli minst mulig påvirket av multiveis propagering og andre forstyrrelser. Det ble gjort ti avstandsmålinger på hver femte meter opp til 20 meter. Deretter ble gjennomsnittet utregnet som er presentert i figur 4.1.

Tabell 4: Gjennomsnitt og standardavvik for avstandstest.

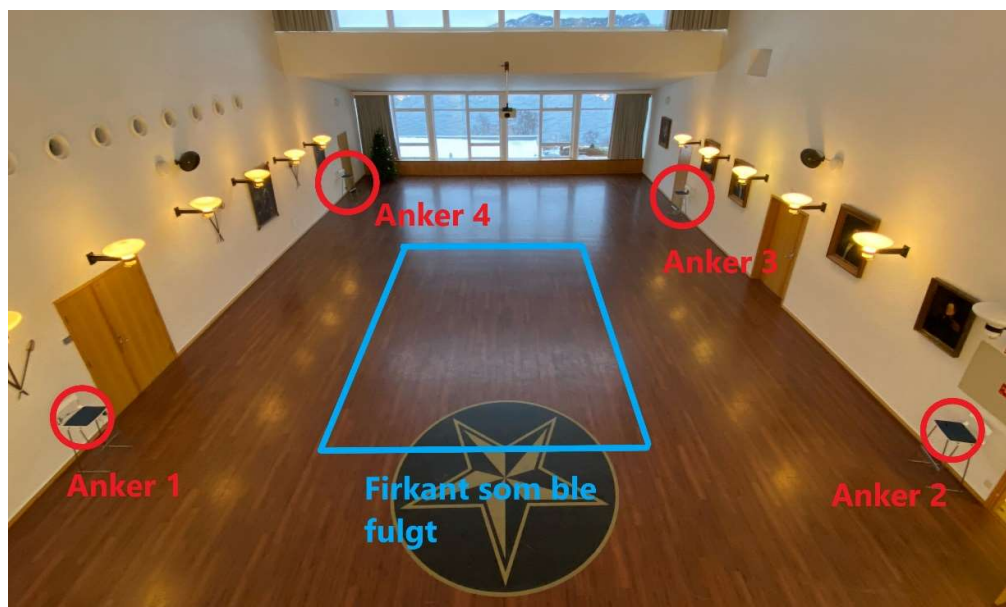
Avstand	Gjennomsnitt	Standardavvik
5 meter	5,02	0,02590581
10 meter	9,99	0,02716207
15 meter	14,94	0,02
20 meter	19,90	0,03521363

Som et krav til systemet satt vi at oppløsningen skulle være på  $\pm$  en meter. Det skal sies at oppløsningskravet ikke kan sammenlignes med distansemålingene her, da en trenger minst tre referansepunkter for å kunne utregne posisjonen til tagen. Hvis en ser på figur 4.1 kan en se at det største avviket i distanse var på 15 meter hvor tagen målte 14,94 meter. Hvorvidt dette avviket skyldes feilaktig målinger til den faktiske posisjonen eller andre faktorer er vanskelig å si. Da avstandene fra ankeret til tagen ble målt opp med målebånd kan det være at dette avviket skyldes menneskelige feil. Uansett er ett avvik på seks centimeter mer enn nøyaktig nok for dette systemet.

Det ble gjennomført en test for å finne ut den maksimale distansen mellom tag og anker. Den maksimale distansen hvor tag og anker klarte å kommunisere med hverandre var målt til å være 26 meter.

## 4.2 Test av posisjonsnøyaktighet

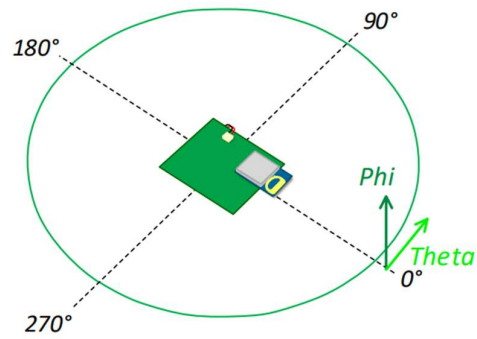
For å teste posisjonsnøyaktigheten til systemet ble det gjennomført en rekke tester med forskjellig oppsett av ankere og tager. Testene ble utført i samme rom som avstandsmålingene i forrige delkapittel. Oppsettet for testen er vist i figur 4.2. Det ble lagt opp en merket firkant som deltager med tag gikk én runde rundt langs kanten per test. Firkanten ble brukt i alle testene som et felles referansepunkt. I diagrammene som blir vist senere i delkapittelet har hjørnene i firkanten koordinatene (1.95, 2.7), (1.95, 8.35), (13.975, 8.325) og (13.975, 2.7). Det ble gjennomført fire forskjellige tester med tre forskjellige oppsett.



**Figur 4.2: Oppsettet for testen. Ankerene er plassert på stoler i hvert sitt hjørne av rommet. Firkanten som testpersonen fulgte, er vist med blå markering.**

De forskjellige testene var.

1. Tag plassert over hode.
2. Tag plassert i brystlommen på uniform.
3. Tag plassert på hånden under ermet til uniformen for å simulere et armbånd.
4. Tag båret i brysthøyde med tagen orientert langs XY-planet som vist i figur 4.3 med antennen pekende ut fra kropp.



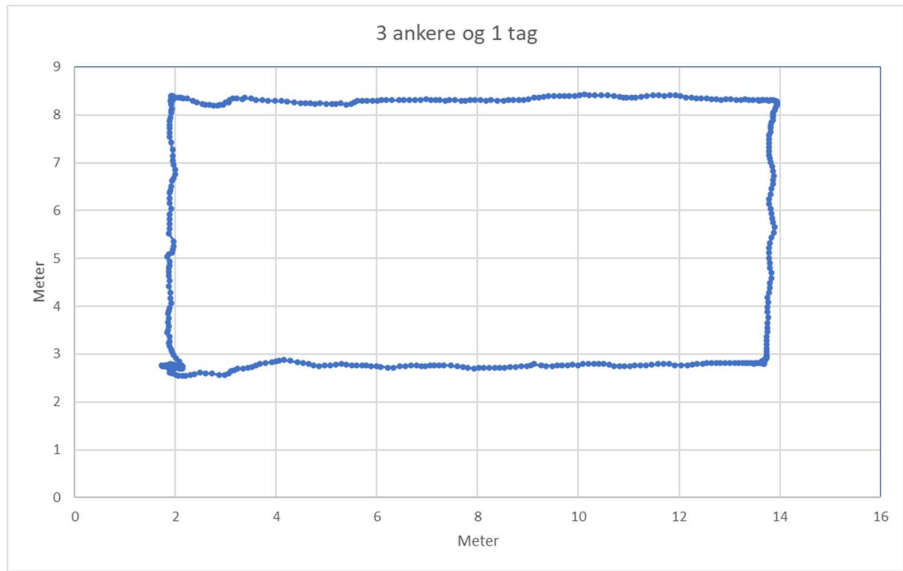
**Figur 4.3: Orientering av brikken under test 4  
(DecaWave, 2021)**

De tre forskjellige oppsettene for testene var.

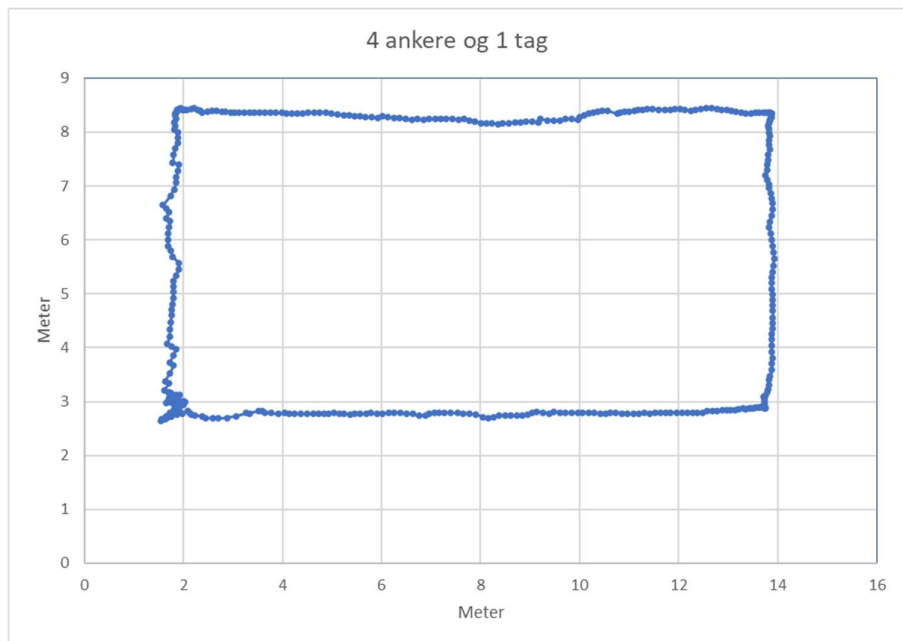
1. 4 ankere og 1 tag
2. 3 ankere og 1 tag
3. 3 ankere og 2 tager

#### **4.2.0 Test 1: Tag plassert over hode**

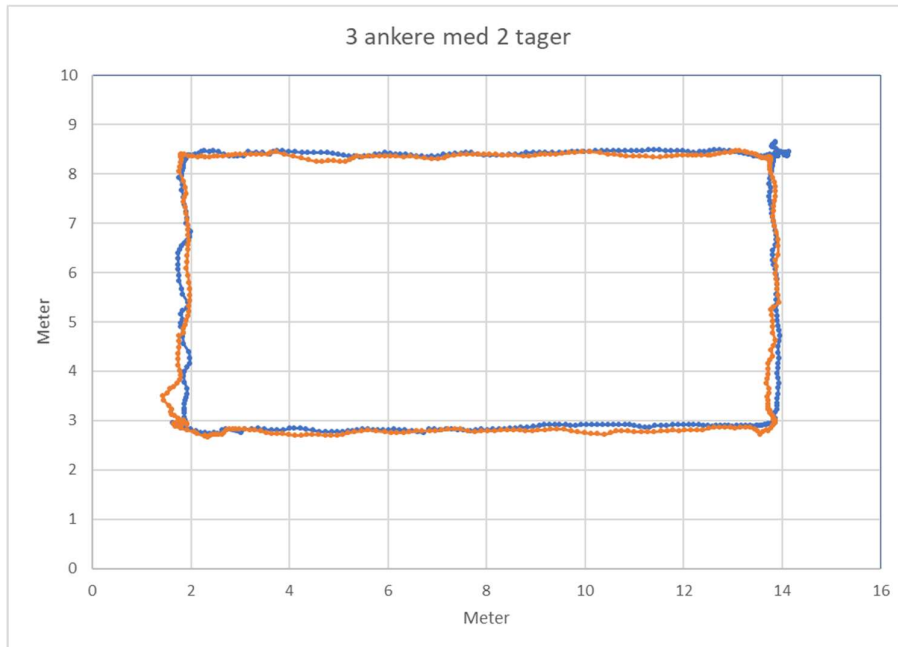
Figur 4.4, 4.5 og 4.6 viser målingene fra de forskjellige oppsettene da testen ble gjennomført. Testen ble gjennomført for å kartlegge posisjonsnøyaktigheten til systemet med fri siktlinje mellom anker og tag. Det er lite variasjon i posisjonsnøyaktigheten til de tre oppsettene.



**Figur 4.4: Test 1 - tag over hode, oppsett med 3 ankere og 1 tag**



**Figur 4.5: Test 1 - tag over hode, oppsett med 4 ankere og 1 tag**

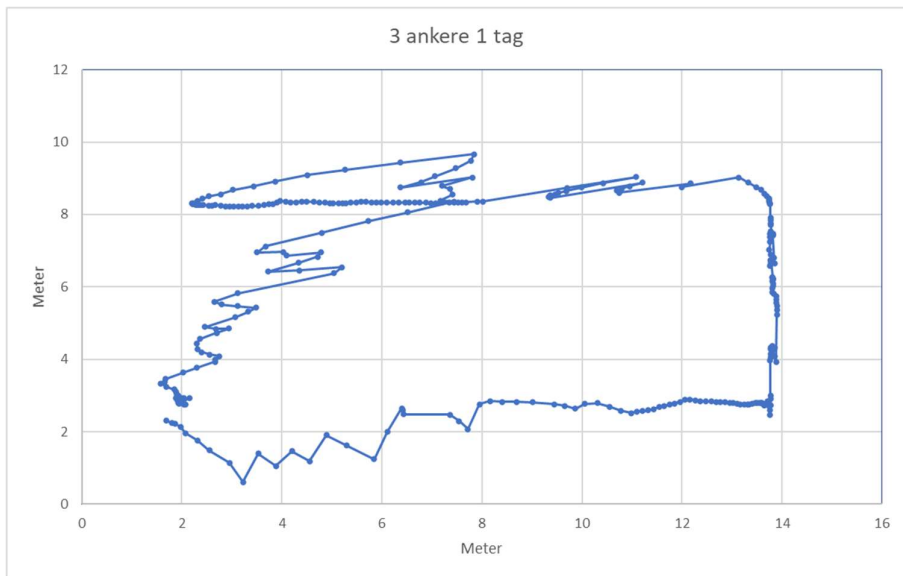


**Figur 4.6: Test 1 - tag over hode, oppsett med 3 ankere og 2 tager**

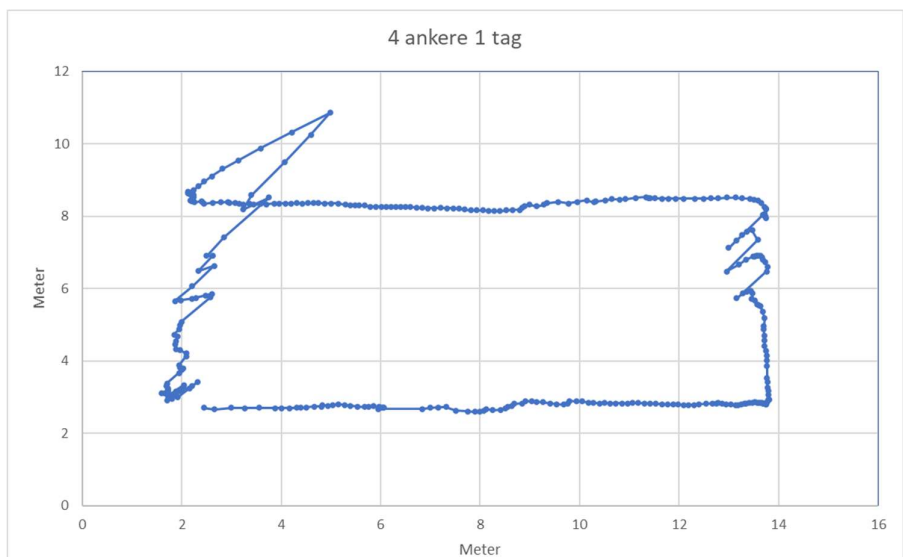
#### **4.2.1 Test 2: Tag plassert i brystlommen på uniform**

Testen ble gjennomført for å teste posisjonsnøyaktigheten til systemet hvis tagen blir plassert under tøystoff. Målingene fra de forskjellige oppsettene da testen ble gjennomført er vist i figur 4.7, 4.8 og 4.9.

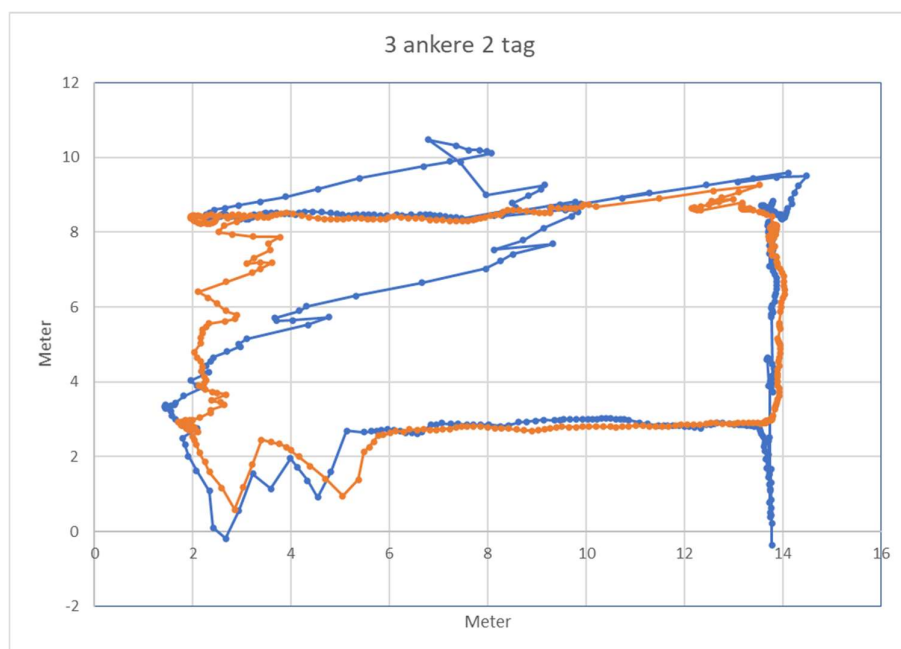




**Figur 4.8: Test 2 – tag i brystlomme på uniform, oppsett med 4 ankere og 1 tag**



**Figur 4.7: Test 2 - tag i brystlomme på uniform, oppsett med 4 ankere og 1 tag**



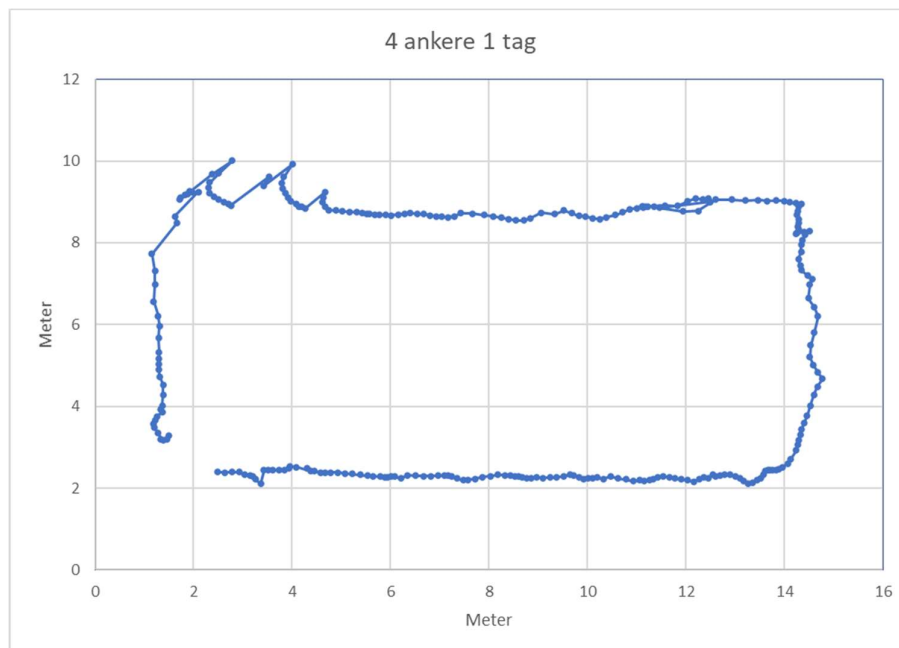
**Figur 4.9: Test 2 - tag i brystlomme på uniform, oppsett med 4 ankere og 1 tag**

#### **4.2.2 Test 3: Tag plassert under ermet for å simulere et armbånd**

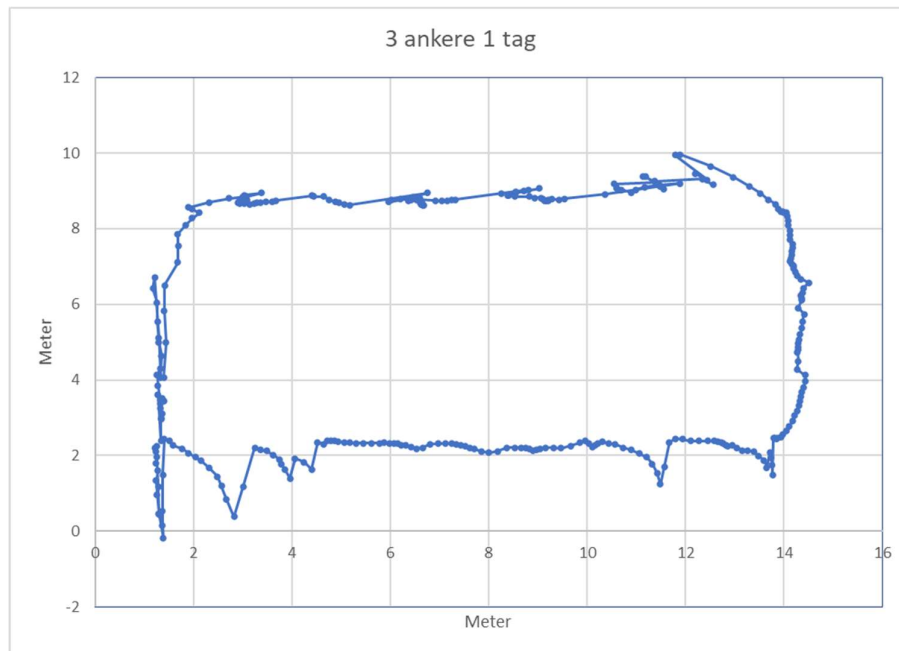
Figur 4.11, 4.12 og 4.13 viser målingene som var tatt under de forskjellige oppsettene da testen ble gjennomført. Testen ble gjennomført for å kartlegge posisjonsnøyaktigheten til systemet hvis tagen hadde blitt brukt som et armbånd. Figur 4.10 viser hvordan tagen ble plassert på hånden hvor ermet ble brettet over tagen. Ettersom hånden svinger ved naturlig gange medførte dette at orienteringen til antennen på tagen var under kontinuerlig endring i testen. Svingningen vil også bety at det er periodisk høyere og lavere hastighet på tagen enn når tagen for eksempel er plassert over hode. Under oppsettet med 3 ankere og 2 tagger ble ikke ermet brettet over tagen. Dette er en feil under gjennomføring av denne testen som medfører at resultatet i figur 4.13 ikke kan sammenlignes i samme grad med de to andre oppsettene. En annen presisering under samme gjennomføring er at tagen ble båret på motsatt side av hverandre. Den ene tagen ble holdt med høyre hånd, men andre tag ble holdt med venstre hånd. I resulterer derfor med at den ene firkanten ligger inne i den andre i figur 4.13.



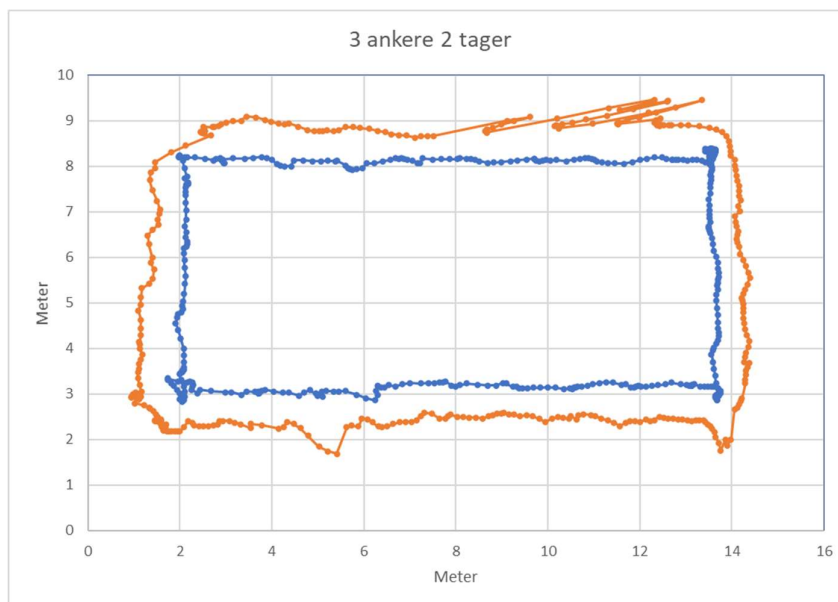
**Figur 4.10:** Slik ble tagen båret under testen. Tagen er festet med teip til hånden. Ermet ble brettet over tagen når testen ble gjennomført



**Figur 4.11:** Test 3 - tag båret som armbånd, oppsett med 4 ankere og 1 tag



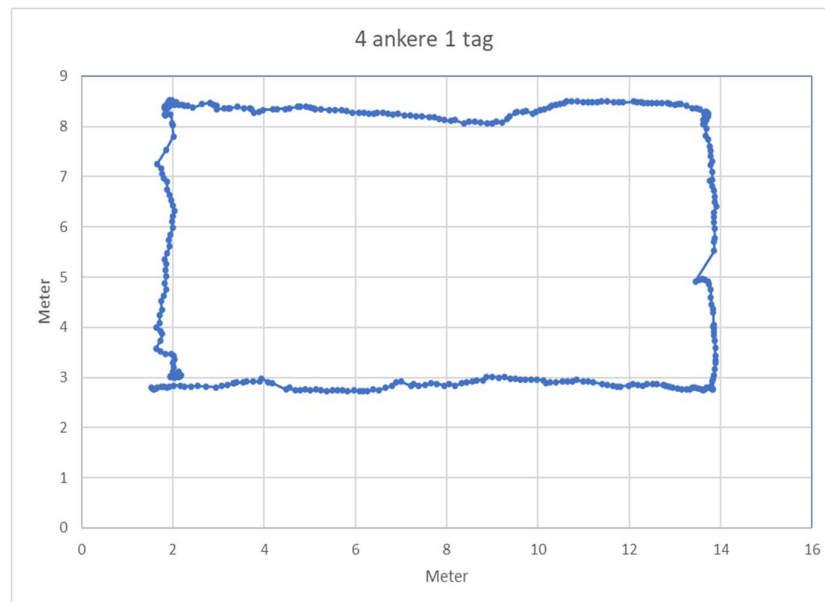
**Figur 4.12: Test 3 - tag båret som armbånd, oppsett med 3 ankere og 1 tag**



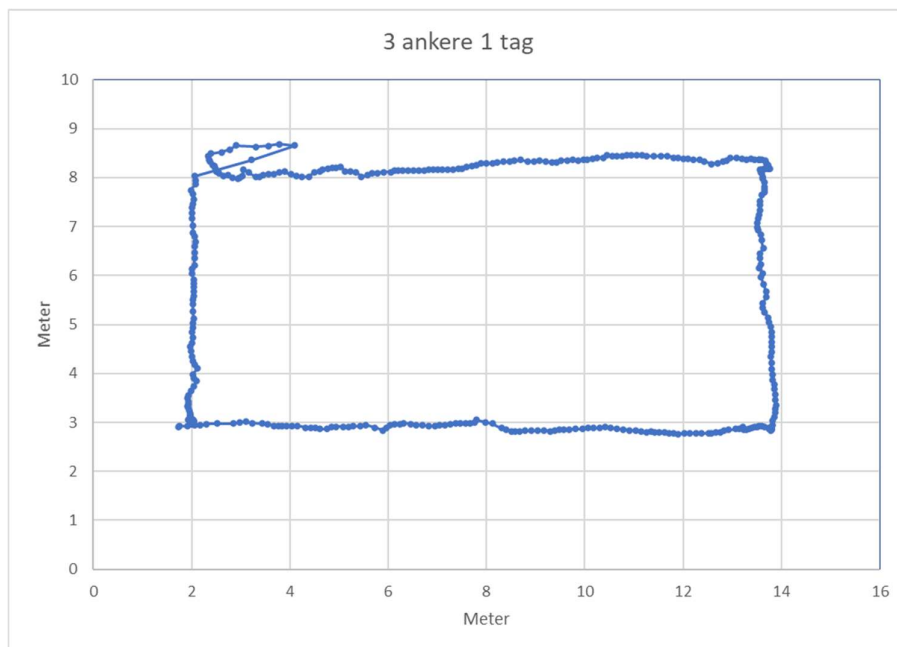
**Figur 4.13: Test 3 - tag båret som armbånd. Oppsett med 3 ankere og 2 tager. Under dette oppsettet ble ikke ermet brettet over tagen som i de andre gjennomføringene.**

### 4.2.3 Test 4: Tag båret i brysthøyde med tagen orientert langs XY-planet

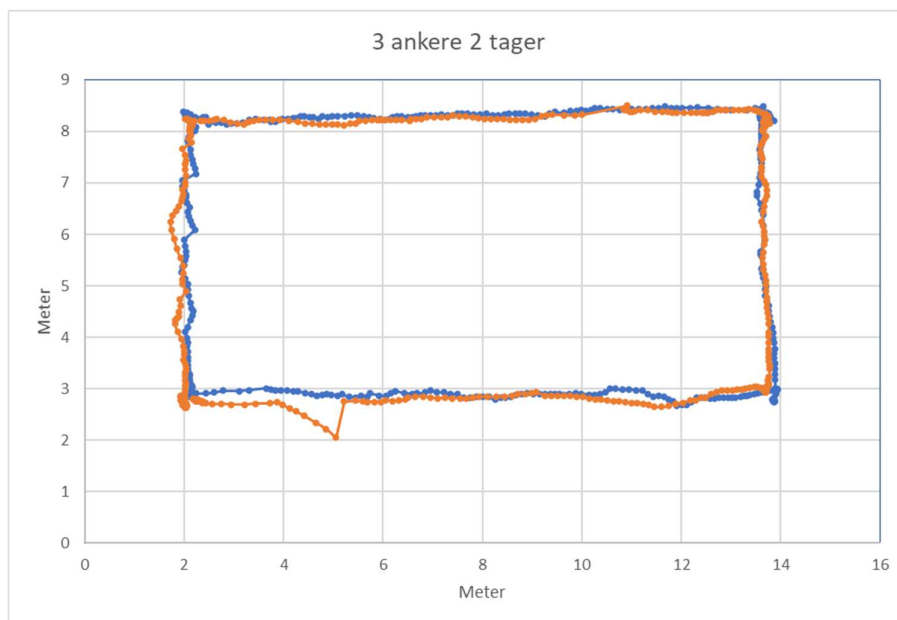
Figur 4.14, 4.15 og 4.16 viser målingene som var tatt under de forskjellige oppsettene da testen ble gjennomført. Testen ble gjennomført for å kartlegge hvordan orienteringen av antennen på tagen påvirket posisjonsnøyaktigheten til systemet. Antennen var orientert langs XY-planet som vist i figur 4.3 med antenedelen av modulen pekende ut fra kroppen.



**Figur 4.14: Test 4 – tag båret i brysthøyde med tagen orientert langs XY-planet. Oppsett med 4 ankere og 1 tag.**



**Figur 4.16: Test 4 – tag båret i brysthøyde med tagen orientert langs XY-planet. Oppsett med 3 ankere og 1 tag.**



**Figur 4.15: Test 4 – tag båret i brysthøyde med tagen orientert langs XY-planet. Oppsett med 3 ankere og 2 tager.**

#### 4.2.4 Oppdateringsfrekvens til systemet under test

Figur 4.17 viser oppdateringsfrekvensen til systemet under testene ved de forskjellige oppsettene. Tiden det tok for å gjennomføre de ulike testene ble notert. Antall oppdateringer som var sendt inn til databasen i løpet av denne tiden er brukt til å regne ut oppdateringsfrekvensen. Oppdateringsfrekvensen som er konfigurert i android-applikasjonen ble satt til å være 10 Hz. En feil under gjennomføringen av test 3 var at ermet på uniformen ikke ble brettet over tagene under oppsettet med 3 ankere og 2 tager. Disse oppdateringsfrekvensene er markert i lys rød farge.

Systemet laster opp tidspunktet for oppdatering med nøyaktighet i sekunder. Derfor ble det ikke mulig å regne ut standardavviket til oppdateringsfrekvensene da en ville trenge bedre tidsnøyaktighet.

Test 1 : Tag over hode	Hz (tag 1)	Hz (tag 2)	Test 3 : Tag som armbånd	Hz (tag 1)	Hz (tag 2)
4 ankere og 1 tag	9.90		4 ankere og 1 tag	7.28	
3 ankere og 1 tag	9.50		3 ankere og 1 tag	6.55	
3 ankere og 2 tager	9.27	9.08	3 ankere og 2 tager	9.45	8.13

Test 2 : Tag i brystlomme	Hz (tag 1)	Hz (tag 2)	Test 4 : Tag orientert horisontalt	Hz (tag 1)	Hz (tag 2)
4 ankere og 1 tag	8.66		4 ankere og 1 tag	9.53	
3 ankere og 1 tag	7.40		3 ankere og 1 tag	9.46	
3 ankere og 2 tager	7.90	9.21	3 ankere og 2 tager	9.61	9.12

**Figur 4.17: Oppdateringsfrekvenser til systemet under testing i de forskjellige oppsettene. Test 3 med 3 ankere og 2 tager ble det feil under gjennomføring av testen.**

**Disse verdiene er markert med lys rød farge.**

## 5 Drøfting

I dette kapittelet skal det presenterte systemet drøftes opp mot kravene som har blitt satt i oppgaven. Først vil de viktigste funnene fra resultatene bli drøftet. Videre vil oppgaven se på hvordan systemet presterer som et personellkontrollsystem. Deretter vil oppgaven belyse flere problemstillinger som systemet møter på. Til slutt vil oppgaven se på videre utvikling av systemet som et personellkontrollsystem.

### 5.1 Test og resultater

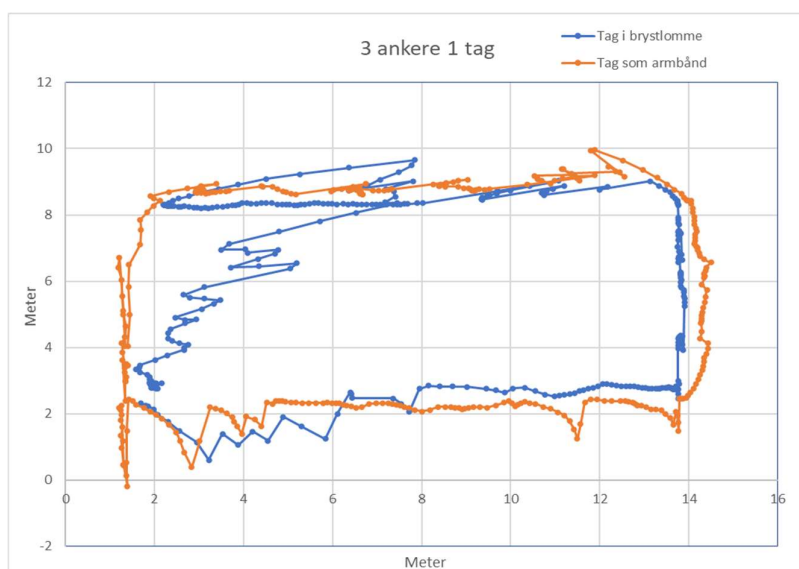
Testen med best posisjonsnøyaktighet og oppdateringsfrekvens var testen hvor tagen ble plassert over hode. Det kan gi en indikasjon på at fri siktlinje mellom anker og tag har en innvirkning på hvor nøyaktige og hyppige målinger systemet kan produsere. Testen hvor tagen ble båret orientert horisontalt hadde også god oppdateringsfrekvens og posisjonsnøyaktighet. Denne testen hadde i likhet med test 1 ingen klesplagg som dekket siktlinjen mellom taggen og ankerene. Både test 1 og 4 viste bedre resultater enn test 2 og 4 hvor tagen var dekket av uniformen og hadde dermed ikke fri siktlinje til ankerene.

For å kartlegge påvirkningen antall ankere i et rom har på systemet kan en sammenligne resultatet på testene til oppsettene med 4 ankere og 1 tag, med resultatet til oppsettet med 3 ankere og 1 tag. Oppsettet med 4 ankere og 1 tag hadde høyere oppdateringsfrekvens i alle testene. I test 1 og 4 var posisjonsnøyaktigheten tilnærmet lik. I test 2 og 3 var posisjonsnøyaktigheten noe bedre i oppsettet med 4 ankere, men denne forskjellen var ikke stor. Dette kan gi en indikasjon på at et høyere antall ankere i rommet har en innvirkning på oppdateringsfrekvensen, men ikke så stor innvirkning på posisjonsnøyaktigheten. Det trengs distansemålinger mellom tag og tre ankere for å beregne en posisjon. Hvis det er fire ankere i rommet er en ikke avhengig av at tagen får kommunikasjon med alle fire. Så lenge tagen får kommunisert med tre er det tilstrekkelig. I oppsettet med tre ankere og en tag er derimot tagen avhengig av å kommunisere med alle ankerene for at systemet kan utregne posisjonen. Hvis en ønsker et system som har så høy oppdateringsfrekvens som mulig vil det bety at en burde ha mange ankere. Derimot er det ingen selvfølge at flere ankere vil gi systemet en bedre posisjonsnøyaktighet. Den største forskjellen i avvik mellom 3 og 4 ankere i gjennomsnittlig oppdateringsfrekvens var på 1,26 Hz. Her hadde systemet likevel en oppdateringsfrekvens på 7,40 Hz med tre ankere, mer enn godt nok til praktisk bruk.



En sammenligning av oppsettet med 3 ankere og 1 tag og oppsettet med 3 ankere og 2 tager kan gi et innblikk i hvordan systemet fungerer med et økende antall tager i samme rom. Oppdateringsfrekvensen på de to oppsettene gir ingen tydelig uttrykk for at systemet blir særlig påvirket av å gi sanntidsposisjonering av to tager samtidig. Posisjonsnøyaktigheten ser heller ikke ut til å ha blitt påvirket av at systemet sporet to tager istedenfor en. Resultatet fra testene med disse to oppsettene viser at systemet ikke blir påvirket av å måtte håndtere to tager istedenfor en. Derimot er det en begrensning i testene at oppgaven bare har tilgang på fem moduler av DecaWave sine DWM1001 moduler. Det hadde vært nyttig med videre testing for å se hvordan systemet hadde taklet å spore et enda høyere antall tager, gjerne 10-20 tager i et rom samtidig. For at systemet skal være skalerbart til å implementeres på marinens fartøyer er det nødt til å takle å spore et slikt antall tager i samme rom.

En fellesnevner for test 2 og 4 er at tagen var dekket av uniformen og hadde dermed ikke fri siktlinje til ankerene, likevel var resultatet av testene veldig forskjellige. Oppdateringsfrekvensen var høyere når tagen var plassert i brystlommen enn når den var plassert under ermet til uniformen. Derimot var posisjonsnøyaktigheten noe bedre i test 4 enn i test 2, resultatene fra de to testene er vist sammen i figur 5.1, her under oppsettet med 3 ankere og 1 tag.



**Figur 5.1: Resultat av test 2 og 3 – oppsett med 3 ankere og 1 tag. Tag i brystlomme er vist i blå farge. Tag som armbånd er vist i oransje farge.**

En grunn til dette kan være at tagen kommuniserer bedre når den holdes i ro enn når den beveger seg fort. I delkapittel 4.2.2 ble det nevnt at hånden svinger frem og tilbake ved naturlig gange. Det betyr at tagen beveger seg fort når hånden svinger fremover, men

sakte når den svinger bakover. Figur 4.11 som viser målingene i test 3 for oppsettet med 3 ankere og 1 tag kan en se denne svingningen relativt godt. Målingene mens tagen beveger seg lite er da også ganske gode, men når den beveger seg fort er det nesten ingen oppdateringer. Dette er en observasjon som trenger videre tester som sammenligner målinger etter å ha gått samme ruten i forskjellige hastigheter.

## 5.2 Personellkontrollsystem

For å ha et effektivt personellkontrollsystem innebærer det å minske tiden det tar å oppnå personellkontroll på. Oppgaven har presentert et RTLS system slik at man i krisesituasjoner har mulighet til å raskt få oversikt over hvor personell om bord befinner seg. Dette gjør det mulig for den som er ansvarlig for å oppnå personellkontroll til å finne savnet personell. Det burde samtidig være funksjoner i systemet som kan monitorere helsetilstanden på personellet slik at ansvarlig vet at de som blir telt opp faktisk er friske. Med et slikt system kan en dermed raskt få et overblikk på hvordan tilstanden til besetningen på fartøyet er og effektivisere personellkontroll-prosessen.

Oppgaven har testet hvordan posisjonsnøyaktighet og oppdateringsfrekvens blir påvirket av måten en bærer tagen på. Under testene så en at systemet fikk best resultater når tagen ble båret over hode. Men hvis en skal ferdigstille et personellkontrollsystem må tagen kunne bli båret slik at den ikke er til bry. Mulige implementeringer av tagen kan for eksempel være som et armbånd på hånden eller halskjede rundt nakken. Testene i oppgaven viste at slike implementeringer fikk noe dårligere posisjonsnøyaktighet og oppdateringsfrekvenser. Derimot var det godt nok til å se at en lignende implementering kunne vært en bra løsning. Testene med tag festet som et armbånd hadde sitt største avvik på rett over 1,5 meter. Dette avviket varte bare i et kort øyeblikk. Som regel holdt målingene på posisjonering av tag seg innenfor kravet for oppgaven på  $\pm$  en meter. Ved å bære tagen som et armbånd istedenfor å ha det i brystlommen på uniformen unngår en problemet som oppstår hvis personen tar av seg overdelen.

Systemet som oppgaven har presentert har totalt bestått av fem DecaWave DWM1001-moduler, men et fullt skalert system på marinens fartøyer ville trenge mange flere. Under resultater ble det vist at systemet taklet å spore to tager like greit som én tag. Det ble senere drøftet at systemet burde testes med mange flere, men det ga allikevel en indikasjon på at systemet kunne skaleres med flere tager. En viste også at systemet taklet å bli

skalert med flere ankere da det ble testet oppsett med både tre og fire ankere. I brukergrensesnittet har en mulighet til å registrere nytt personell samt nye tager. En har også mulighet til å opprette nye koblinger mellom personell og ledige tager. Dermed har brukergrensesnittet mulighet til å skaleres opp til å spore antall personell en har tager til.

### **5.3 Datasikkerhet**

Forsvaret er under daglige cyberangrep. Hensikten er som oftest å spionere på forsvarets bemanning, lokalisering, operasjoner og planlegging (Sundlisæter, 2013). Med ett RTLS-system installert på fartøyene kan dette utgjøre en stor sikkerhetstrussel. Informasjon om hvem som seiler og når de seiler er informasjon som en potensiell fiende kan utnytte for å svekke forsvarets operasjoner. Det er derfor svært viktig at denne informasjonen ikke er tilgjengelig for andre enn de som skal benytte systemet. Selv om oppgaven ikke tar for seg datasikkerheten rundt systemet er det fortsatt en viktig faktor en må tenke på før det implementeres. I oppgaven blir posisjonsinformasjonen sendt over ett lokalt nettverk. Det kan derfor argumenteres at cyberangrep på et lokalt nettverk om bord på marinens fartøyer er å anse som ekstremt vanskelig, men en kan ikke se bort i fra dette. Det må gjøres tiltak for å sikre at denne informasjonen ikke kan aksesseres av andre enn forsvaret og at nødvendig kryptering av data blir gjennomført.

### **5.4 Utstråling**

Under daglig drift av marinens fartøyer har fartøyene ulike grader av emisjonskontroll av hensyn til operasjonssikkerhet. For eksempel vil overflateradarer sende ut elektromagnetiske bølger hvor energien avtar med avstanden. For at radaren skal fungere trenger de elektromagnetiske bølgene å bli reflektert tilbake igjen for å kunne fange opp signalet. Dermed må signalet bevege seg to ganger avstanden fra utsendelse til refleksjon og miste masse energi. Dette betyr at fartøy A, som lytter etter radarpulser, kan detektere fartøy B, som har aktiv radar, på langt hold uten å at fartøy A blir detektert. Derfor foretrekker man å lytte etter andre utstrålinger, da en kan detektere signalene på lengere avstand uten å selv bli sett. UWB-brikkene vil sende ut elektromagnetiske bølger som kan detekteres, de kan også forstyrre andre elektroniske komponenter om bord på fartøyet. Da UWB sender på såpass lav effekt, sammenlignet med WiFi, kan det argumenteres for at UWB ikke vil påvirke interne komponenter om bord da WiFi allerede er utstyr på fartøyene.

På overflatefartøyer vil det være tidspunkter hvor besetningsmedlemmer beveger seg over le. Da vil UWB-pulsene være mer eksponert for andre lyttere. For minefartøy som søker etter miner er emisjonskontroll ekstremt viktig da noen miner kan utløses av elektromagnetiske signaturer. Det blir derfor viktig å kartlegge om UWB-pulsene påvirker sjøminer. En ønsker heller ikke at en potensiell fiende kan utvikle sjøminer som utløses av den elektromagnetiske signaturen til UWB-pulsen. Hvorvidt signalstyrken er høy nok til at systemet ikke kan benyttes grunnet operasjonssikkerhet, tar ikke oppgaven for seg, men det er viktig at dette vurderes nøye før ett slikt system kan implementeres, spesielt på minefartøyer.

## 5.5 Strømforbruk

I utviklingen av ett sanntidslokaliseringssystem vil det være beslutninger om designet en må ta som påvirker flere faktorer. Batteristørrelsen påvirker levetiden til tagen, men vil også påvirke formfaktoren. Det er ønskelig å ha en tag med lang levetid så en slipper å lade batteriene hyppig. Samtidig er det heller ikke hensiktsmessig å ha for stort batteri da formfaktoren til tagen og batteriet kan bli for stort for daglig bruk. I denne oppgaven ble tagen tilkoblet en Raspberry Pi for å hente ut posisjonsinformasjonen. Dette er på ingen måte en ideell formfaktor da størrelsen på tagen blir for stor og strømforbruket blir for høyt. Det er derfor viktig å presisere at dette oppsettet ble benyttet for å gjøre oppgaven mindre omfattende da dette er en modell for et RTLS. Videre utvikling av formfaktor blir derfor viktig før en kan implementere systemet.

Noe som kjennetegner alle radiosendere, er at strømforbruket vil øke signifikant når en begynner med utsendelse. DecaWave sin DWM1001 modul har en systemkapasitet på 150 Hz. Med andre ord vil det si at man kan ha 15 tags med en oppdateringsfrekvens på 10 Hz eller 150 tags med en oppdateringsfrekvens på 1 Hz. Hvilken oppdateringsfrekvens som er mest fordelaktig vil ikke oppgaven ta for seg, men det er noe en må vurdere før ett slikt system implementeres. Ved høyere oppdateringsfrekvenser vil en få ett mer nøyaktig system og kan følge bevegelsene til personellet mer presist. Nedsiden med en høy oppdateringsfrekvens er at antall blink som utsendes vil øke, og dermed vil strømforbruket øke.

DecaWave har lansert sin DWM1004 brikke som bruker TDOA-måleteknikk hvor de hevder en batteritid på over åtte år med 1000 blink per dag fra ett CR2450 batteri med

620mAh (DecaWave, 2021). Det betyr at hvert blink krever ca. 0.2  $\mu$ Ah. Hvis en ønsker en oppdateringsfrekvens på 1 Hz per tag, vil batterilevetiden bli i underkant av 34 dager. Dette betyr at formfaktoren til brikke og batteri bli så liten at den enkelt kan bæres av ett menneske uten forstyrrelser og krever lite vedlikehold.

## 5.6 Personvern

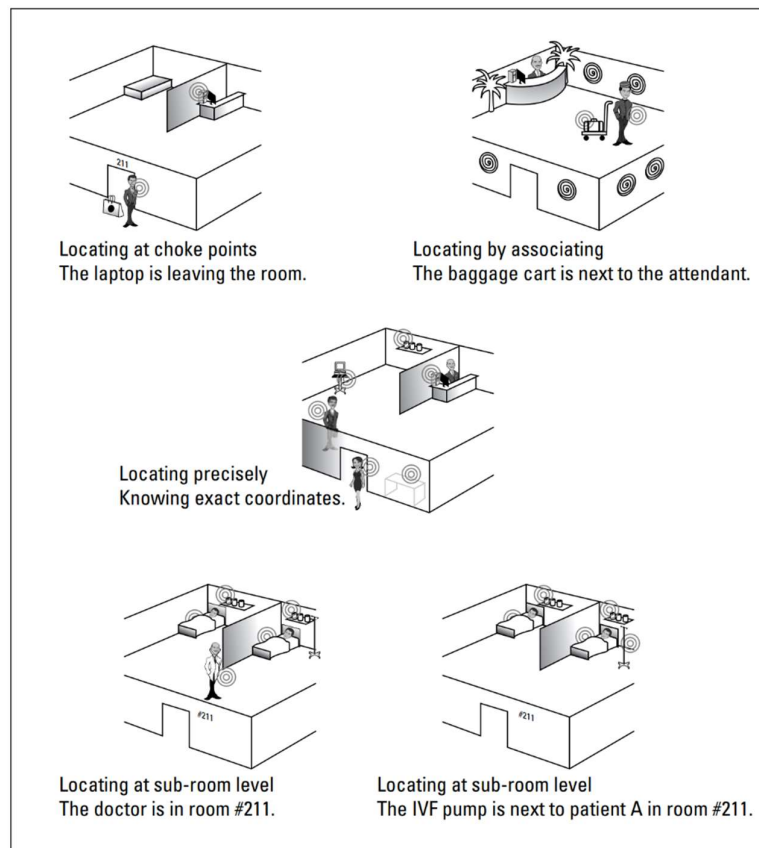
Personvern handler om retten til et uforstyrret privatliv og retten til å regulere sine egne personvernsopplysninger. Dette idealet er et grunnleggende prinsipp for en rettsstat (Datatilsynet, 2021). Prinsippet om retten til eget privatliv har blitt forankret flere steder. I den europeiske menneskerettighetskonvensjonen står det i artikkel 8 at “enhver har rett til respekt for sitt privatliv, sitt hjem og sin korrespondanse.” I Norge valgte Stortinget å styrke personvernet ved å vedta bestemmelsen om personvern i Grunnloven. I grunnlovsparagraf 102 står det at “Enhver har rett til respekt for sitt privatliv og familieliv, sitt hjem og sin kommunikasjon. Husransakelse må ikke finne sted, unntatt i kriminelle tilfeller. Statens myndigheter skal sikre et vern om den personlige integritet.” (Datatilsynet, 2021). Det er ingen tvil om at personvern er i søkelyset om dagen. Hvorvidt et militært fartøy kan anses som hjem er diskuterbart. Flere av besetningsmedlemmene oppholder seg månedsvis om bord på tokt. Det kan argumenteres at fartøyet derfor blir ett midlertidig hjem og en har dermed rett til respekt for sitt privatliv. På den andre siden er besetningsmedlemmene om bord på jobb og at fartøyet dermed ikke blir definert som hjem. Uansett må kontinuerlig overvåkning anses som svært inngripende for privatlivet.

Et grunnleggende personvernprinsipp er at innsamling av personvernsopplysninger ikke skal foregå i det skjulte. For virksomheter som ønsker å overvåke sine ansattes bevegelser og posisjoner må det bli gitt samtykke til dette. Som bruker av et RTLS-system har man rett på å vite hva slags informasjon som lagres, hvem som har tilgang på denne informasjonen og hva informasjonen skal brukes til. Det blir derfor viktig med en skriftlig kontrakt som opplyser om hva slags informasjon som lagres om den enkelte, hva den brukes til og hvem som har tilgang. Da hovedintensjonen med overvåkningssystemet er å redde liv i krisesituasjoner, vil det ikke ha noe hensikt å overvåke personellet under daglig drift. Dette vil oppleves som mindre krenkende for privatlivet, men vil fortsatt opprettholde intensjonen til systemet. Det kan derfor være hensiktsmessig at ett fåtall personer får tilgang på systemet slik at det ikke misbrukes.

## 5.7 Videre utvikling av personellkontrollsystemet

Det er en rekke funksjoner som kan bli lagt til i tagen som vil gjøre systemet som en helhet bedre. I delkapittel 5.2 ble det nevnt at det burde være en funksjon som kan monitorere helsetilstanden til personen som bruker tagen. En måte å gjøre dette på er ved å ha en pulsmåler implementert i tagen. Pulsmålere har blitt en svært vanlig funksjon på smartklokker i dagens samfunn (Paxton, 2020). En pulsmåler vil ikke bare si om personen er i live eller ikke, men den kan gi et innblikk i hvilken helsetilstand personen befinner seg i. Om en har mulighet til å monitorere hastighet, styrke og rytme på pulsen med tagen kan personen som er ansvarlig for å oppnå personellkontroll få et bedre innblikk i tilstanden til personellet. Dette vil gjøre det enklere å fatte beslutning om hvor og hvem som trenger hjelp i krisesituasjoner.

DWM1001 modulen har et innebygd akselerometer som ikke ble benyttet i denne oppgaven, derimot er det mulig å forbedre systemet ved å ta nytte av denne kapasiteten. Akselerometeret kan fortelle når tagen står i ro over lengre tid, og kan dermed fortelle modulen



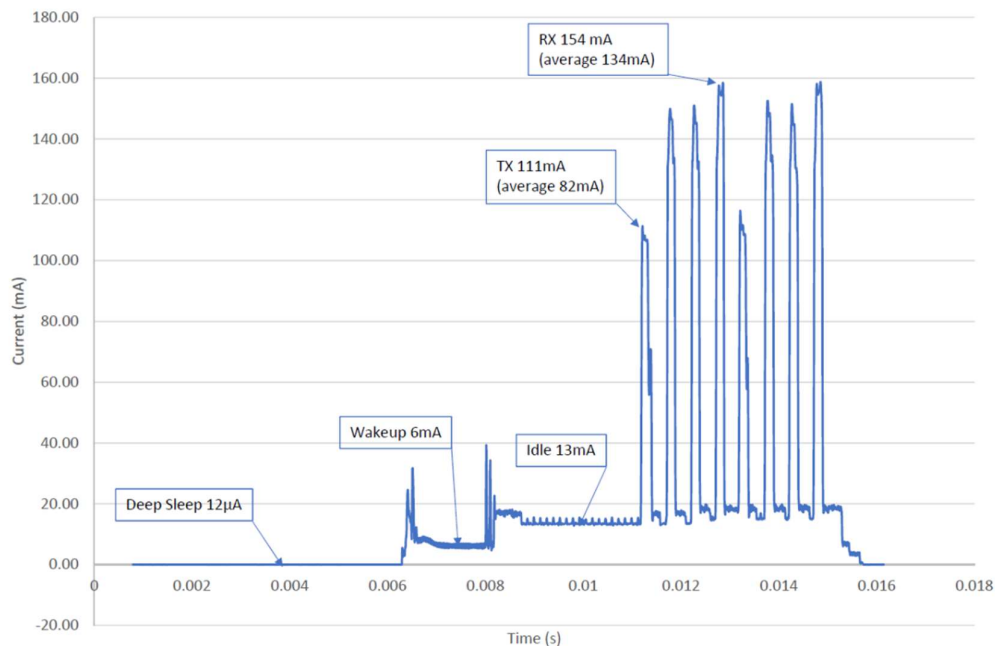
Figur 5.2: RTLS med forskjellig presisjon av lokalisering

til å ikke sende ut nye forespørsler om oppdatering (TWR). Så fort akselerometeret merker at den er i bevegelse igjen kan den be modulen starte posisjonsoppdateringer. På denne måten kan en minske strømforbruket til modulen. Derimot er fartøy bevegelige plattformer som vil bli påvirket av bølger, vær og vind. Dette vil ha innvirkning på potensialet av en slik funksjon. Et annet bruksområde for akselerometeret kan være å bedre detektere mann over bord. Utvikling av algoritmer som gjenkjenner akselerasjonsmønsteret til en person i et mann-over-bord-scenario kan gjøre at en hurtig varsles om en slik ulykke inntreffer. Dette kan spare viktig tid og potensielt redde liv.

Oppgaven tar for seg å lage et personellkontrollsystem ved bruk av UWB til å finne presis posisjonering. Derimot kan en argumentere for at en posisjonsoppløsning på  $\pm$  en meter blir unødvendig nøyaktig i flere tilfeller. Det kan være tilstrekkelig nok å vite hvilket rom personen befinner seg i. I andre tilfeller holder det å vite strupepunkter som personen sist befant seg i. Figur 5.2 viser forskjellige teknikker på hvordan posisjonsnøyaktigheten kan begrense seg. Det kan være flere fordeler ved å senke kravet for posisjonsoppløsningen. For å få presis posisjonering i et plan trenger du tre eller flere referansepunkter (ankere). Hvis du bare vil vite hvilket rom en person befinner seg i trenger du ikke flere enn ett referansepunkt. Dermed er det potensielt to færre ankere en er nødt til å installere per rom, noe som vil gjøre det enklere å skalere samt senke kostnaden for systemet. Siden tagen trenger å kommunisere med færre ankere om gangen betyr det at strømforbruket også reduseres. Slik kan det også bli mer brukervennlig og mindre krevende å vedlikeholde systemet.

## 5.8 TDOA til fordel for TWR

Systemet som har blitt presentert i oppgaven bruker avstandsteknikken TWR og regner ut posisjonen direkte på tagen. Dette krever mye mer strøm enn nødvendig for ett slikt system. Ved å bruke TDOA-protokollen for å måle avstander slipper en langt flere utsendelser enn nødvendig. TWR-protokollen til DecaWave, gitt en tag og tre ankere, vil tagen normalt ha to utsendelser og mottar seks pulser som illustrert i figur 5.3. TDOA derimot



**Figur 5.3: Strømforbruk til DWM1001 ved TWR**

trenger kun én utsendelse og er ikke avhengig av svar fra ankere. Dette vil gjøre at tagene krever vesentlig mindre strøm som gjør at levetiden blir langt høyere enn ved TWR. Noe en må tenke på ved bruk av TDOA er at systemet da krever en sentralisert «location engine». Det betyr at alle ankere må være tilkoblet en server som kalkulerer posisjonen, da man ikke kan utregne posisjonen direkte fra tagen. Dette gjør at systemet kan bli komplekst med tanke på all datatrafikken, men håndterbart. En annen utfordring med TDOA er at ankerene må være tidssynkroniserte, noe som øker kompleksiteten til systemet vesentlig. Da UWB-pulsene er på noen få nanosekunder betyr det at et synkroniseringsavvik på ett nanosekund vil gi et avstandsavvik på ca. 30 centimeter. Selv om kompleksiteten øker ved å benytte seg av TDOA, vil fordelene være mange. Færre utsendelser gir mindre strømforbruk og formfaktor, samt minsker sannsynligheten for interferens og øker skalerbarheten.





## 6 Avslutning

Denne oppgaven har tatt for seg en modell av et RTLS system ved bruk av UWB-teknologi for å øke personellkontroll om bord på marinens fartøyer.

Skalerbarheten til systemet er tilstrekkelig for å kunne benyttes om bord på marinenes fartøyer. Oppløsningen til systemet, som er fremstilt i resultatene, bekrefter at posisjonsnøyaktigheten tilfredsstiller kravet på  $\pm$  en meter. Brukergrensesnittet er enkelt å betjene og gir en ryddig oversikt over posisjonen på personellet om bord. Hvis et besetningsmedlem beveger seg utenfor en gitt sikkerhetssone, vil systemet slå alarm for å varsle mann om bord. Derimot står utviklingen av en mer kompakt tag igjen samt flere undersøkelser under mer krevende forhold. Det kreves også å undersøke hvordan ruting av data kan bli gjort på en hensiktsmessig måte.

Fordelene med ett slikt system er mange. Ved å ha ett lokaliseringssystem på personellet kan en oppnå personellkontroll på langt kortere tid enn uten. Søk av savnet mannskap under krisesituasjoner vil ta langt kortere tid, da en vet posisjonen og slipper å søke gjennom tomme rom. Systemet vil gjøre at havariansvarlig får et bedre og raskere beslutningsgrunnlag. Havariansvarlig får dermed større overskudd slik at hele havariorganisasjonen effektiviseres.

Vi anbefaler forsvaret å vurdere ett slikt system om bord på marinens fartøyer.

## 7 Referanser

- Angel, R., Antonio, L., David, G., & Ramon, V. (2016). *RFID and Wireless Sensors Using Ultra-Wideband Technology*. ISTE Press - Elsevier.
- Datatilsynet. (2021, Desember 4). *Rettigheter og Plikter*. Hentet fra Hva er personvern?: <https://www.datatilsynet.no/rettigheter-og-plikter/hva-er-personvern/>
- DecaWave. (2021, November 27). <https://www.decawave.com>. Hentet fra decawave-announces-dwm1004module: <https://www.decawave.com/decawave-announces-dwm1004module/>
- Eliko. (2020, Desember 8). *The Story Behind Ultra-Wideband Technology and Indoor Positioning*. Hentet fra Eliko: <https://eliko.ee/history-of-ultra-wideband-technology/>
- Hallek, P. (2020, November 20). *New Forest*. Hentet fra Precise Realtime Indoor Localization With Raspberry Pi And Ultra-Wideband Technology (Decawave DWM1001 Development Boards): <https://medium.com/@newforestberlin/precise-realtime-indoor-localization-with-raspberry-pi-and-ultra-wideband-technology-decawave-191e4e2daa8c>
- Hsu, E. (2021, Desember 4). *Keysight Technologies*. Hentet fra Keysight Blogs: [https://blogs.keysight.com/blogs/tech/rfmw.entry.html/2021/07/28/an\\_overview\\_of\\_ieee-J7ac.html?fbclid=IwAR12NPOUYMdxNAMS5qr3nc2X4UBmBo9J2sSidO5KcTQjKyBK3Eicx\\_CmJ4g](https://blogs.keysight.com/blogs/tech/rfmw.entry.html/2021/07/28/an_overview_of_ieee-J7ac.html?fbclid=IwAR12NPOUYMdxNAMS5qr3nc2X4UBmBo9J2sSidO5KcTQjKyBK3Eicx_CmJ4g)
- Hva er systemdesign?* (2021, Desember 1). Hentet fra education wiki: <https://no.education-wiki.com/3283438-what-is-system-design>
- Jónasson, I. B., & McGuire, K. H. (2021, November 11). *www.uio.no*. Hentet fra Webområde for Universitetet i Oslo: <https://www.uio.no/studier/emner/matnat/astro/AST2000/h20/blogger/bloggen-til-karl-og-ingi/Del-4-Om-bord-orienterings-software/trilaterasjon.html>
- Keycdn. (2018, oktober 4). *What is AJAX programming - Explained*. Hentet fra Keycdn: <https://www.keycdn.com/support/ajax-programming>
- Lagerkvist, A. (2019). *Construction of an indoor positioning*. Luleå University of Technology .

- Malik, A. (2009). *RTLS for Dummies*. Indianapolis: Wiley Publishing Inc.
- MDN Web Docs. (2021, Oktober 2). *HTML: HyperText Markup Language*. Hentet fra MDN Web Docs: <https://developer.mozilla.org/en-US/docs/Web/HTML>
- Panoradio SDR. (2021, Desember 7). *Panoradio SDR*. Hentet fra TDOA Transmitter Localization with RTL-SDRs: <http://www.panoradio-sdr.de/tdoa-transmitter-localization-with-rtl-sdrs/>
- Paxton, M. (2020, August 17). *S&P Global*. Hentet fra Wearable Tech Fitness trackers on the rebound: <https://www.spglobal.com/marketintelligence/en/news-insights/blog/wearable-tech-fitness-trackers-on-the-rebound>
- Raspberry Pi. (2021, November 2). <https://www.raspberrypi.org>. Hentet fra help/what-%20is-a-raspberry-pi/: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>
- Rosencrance, L. (2020, April 17). *User Datagram Protocol (UDP)*. Hentet fra TechTarget: <https://www.techtarget.com/searchnetworking/definition/UDP-User-Datagram-Protocol>
- Roznovsky, A. (2021, Desember 2). *Why use PHP? Main advantages and disadvantages*. Hentet fra Blog: <https://light-it.net/blog/why-use-php-main-advantages-and-disadvantages/>
- Statens havarikommisjon. (2021). *Delrapport 2 om kollisjonen mellom fregatten KNM Helge Ingstad og tankbåten Sola TS utenfor Stureterminalen i Hjeltefjorden, Hordaland, 8. november 2018*. Oslo: Statens havarikommisjon.
- Sundlisæter, T. (2013, April 23). *Forsvaret forbereder Norge på cyberangrep*. Hentet fra E24: <https://e24.no/teknologi/i/5086qb/forsvaret-forbereder-norge-paa-cyberangrep>
- WENGER, M. (2019). *Indoor positioning using Raspberry Pi*. Bern: UNIVERSITY OF BERN.
- Zafer, S., Gezici, ., & Guvenc, I. (2008). *Ultra-wideband Positioning*. Cambridge: Cambridge University Press.
- Øverby, H. (2021, september 30). *TCP/IP*. Hentet fra Store Norske Leksikon: <https://snl.no/TCP/IP>

## 8 Vedlegg

### 8.1 Tag\_to\_DB.py

```
##Kode for å kjøre kommandoer på DecaWave DWM1001 for å skrive til en da-
tabase

import serial
import time
import datetime
import mysql.connector as mariadb
import sys

#informasjon om databasen
conn = mariadb.connect(
    user="personellkontroll",
    password="UlrikHenrik",
    host="192.168.1.30",
    port=3306,
    database = "exampledb")
cur = conn.cursor()

#Hva skal oppdateres i databasen
tid_naa=datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
sql_hist = "INSERT INTO posisjon_historisk (ID_brikke, X_Koordinat, Y_Ko-
ordinat, tid) values ('DECAE9E29814162D',%s,%s,'" + tid_naa + "'" ) "
sql_now = "UPDATE posisjon_naa SET X_Koordinat = %s, Y_Koordinat= %s
WHERE id_brikke = 'DECAE9E29814162D'"

#tilkobling til DWM1001 via USB
DWM=serial.Serial(port="/dev/ttyACM0", baudrate=115200)
print("Connected to " +DWM.name)
DWM.write("\r\r".encode())
time.sleep(1)

#Skriver "lec" til brikken for å hente ut posisjonen ved å parse CVS-for-
matet som blir printet ut
DWM.write("lec\r".encode())
time.sleep(1)
while True:
    try:
        line=DWM.readline()
        if(line):
            if len(line)>=117:
                #Parser informasjonen om posisjonen til tagen
                parse=line.decode().split(",")
```

```

x_pos=parse[parse.index("POS")+1]
y_pos=parse[parse.index("POS")+2]
quality=parse[parse.index("POS")+4]
val = (x_pos,y_pos)

#Legger til historisk data i databasen
cur.execute(sql_hist, val)
conn.commit()
#oppdaterer posisjonen nå
cur.execute(sql_now, val)
conn.commit()

#skriver ut i prompt
print(datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S"), "(" ,x_pos," , ",y_pos," ) qt: ", quality)
else:
    print("Position not calculated: ",line.decode())

except mariadb.Error as e:
    print("Error: ",e,"")
    break
DWM.write("\r".encode())
DWM.close()

```

## 8.2 PHP

### 8.2.1 Nettside for visning av sanntidsposisjonering av personell

```

<!DOCTYPE html>
<html>
  <head>
    <link href="stylesheet.css" rel="stylesheet" type="text/css">
    <script>
      function VisInfo(id, fornavn, etternavn, ansattnummer){}
      function VisPosisjon(){}
      function MannOverBord(fornavn, etternavn, ycoord, xcoord){}
    </script>
  </head>
  <body>
    <h3 class="overskrift">Personellkontrollsystem</h3>

    <div class="outer" >
      <p id="txtHint"><br>Velkommen til Personellkontrollsystem for perso-
nell på fartøy</p>

      <div id="VisPersonellOmbord" class="PersonellOmbord">

```

```

<script>
function VisPersonellOmbord() {
    var xhttp;

    xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("VisPersonellOmbord").innerHTML =
this.responseText;
        }
    }
    xhttp.open("GET", "VisPersonellOmbord.php", true);
    xhttp.send();
}
setInterval(VisPersonellOmbord, 300);

</script>
</div>

<div id="info2">
    <p id="infoskrift"></p>
</div>
<div class = "historiskKnapp" onclick="VisHistoriskData()">
<a href="HistoriskData.php"><button>Vis Historisk data</button></a>
</div>
<div class = "LeggTilPersonellKnapp">
<a href="AdministrerPersonellTags.php"><button onclick = >Administrer
personell-tag kobling</button></a>
</div>
<div class = "AddPersonellTagsKnapp">
<a href="AddPersonellTags.php"><button>Legg til ny tag / perso-
nell</button></a>
</div>

<div class="inner">
    <div class="log">
        
        <div id="posisjon">

<script>
function VisPosisjon() {
    var xhttp;

    xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("posisjon").innerHTML = this.re-
sponseText;
        }
    }
}

```

```

    });
    xmlhttp.open("GET", "VisPosisjon.php", true);
    xmlhttp.send();
}
setInterval(VisPosisjon, 200);

function VisInfo(id, fornavn, etternavn, ansattnummer){
    document.getElementById("info2").style.display = "block";
    document.getElementById("infoskrift").innerHTML = "Fornavn = "+
fornavn + "<br>Etternavn = " + etternavn + "<br>Ansattnummer = " + ansatt-
nummer + ".";
}

</script>
</div>
</div>
</div>
<div class="MannOverbord" id="Mannoverbord"></div>

<script>

function MannOverBord() {
    var xmlhttp;

    xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("Mannoverbord").innerHTML = this.re-
sponseText;
        }
    };
    xmlhttp.open("GET", "VisMannOverBord.php", true);
    xmlhttp.send();
    //document.getElementById("Mannoverbord").style.display="block";
}
setInterval(MannOverBord, 300);

</script>

</div>
</body>
</html>

```



## 8.2.2 PHP kode for å hente ut posisjonsdata

```
<?php

    $servername = "localhost";
    $username = "exampleuser";
    $password = "UlrikHenrik";
    $dbname = "exampledb";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "SELECT p.id, fornavn, etternavn, ansattnummer, X_Koordinat, Y_Koordinat FROM personell p, tags t, tag_to_pers ttp, posisjon_naa pa WHERE p.id = ttp.id_pers AND pa.id_brikke = t.id_brikke AND t.id_brikke = ttp.id_brikke AND ttp.slutt_tid IS NULL ORDER BY etternavn, fornavn";

    if ($result=mysqli_query($conn,$sql)) {
        $rowcount=mysqli_num_rows($result);
    }
    while ($row = $result->fetch_assoc()){
        $x = $row['X_Koordinat']*20;
        $y =$row['Y_Koordinat']*20;
        echo "<div class='circle2' id='person'" . $row['id'] . "' style='left:" . $x . "px;top:" . $y . "px' onclick = VisInfo(" . $row['id'] . "," . $row['fornavn'] . "','" . $row['etternavn'] . "','" . $row['ansattnummer'] . ")></div>";
    }?>
```

## 8.2.3 PHP kode for å finne mann over bord

```
<?php

    $servername = "localhost";
    $username = "exampleuser";
    $password = "UlrikHenrik";
    $dbname = "exampledb";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
}
```

```

        $sql = "SELECT p.id, fornavn, etternavn, ansattnummer, X_Koordinat, Y_Koordinat FROM personell p, tags t, tag_to_pers ttp, posisjon_naa pa WHERE p.id = ttp.id_pers AND pa.id_brikke = t.id_brikke AND t.id_brikke = ttp.id_brikke ORDER BY etternavn, fornavn AND ttp.slutt_tid IS NULL;";
        if ($result=mysqli_query($conn,$sql)) {
            $rowcount=mysqli_num_rows($result);
        }
        while ($row = $result->fetch_assoc()){

            if($row['X_Koordinat']*20<540 && $row['X_Koordinat']>0 && $row['Y_Koordinat']*20<132 && $row['Y_Koordinat']>0){

                return;

            }else{

                echo "<div id='mann'>". $row['fornavn'] . " er over bord!</div>";

            }

        }?>

```

## 8.2.4 PHP kode for å vise personell om bord

```

<?php

    $servername = "localhost";
    $username = "exampleuser";
    $password = "UlrikHenrik";
    $dbname = "exampledb";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    echo "Personell ombord:";
    $sql = "SELECT fornavn, etternavn FROM personell p INNER JOIN tag_to_pers ttp ON p.id = ttp.id_pers WHERE ttp.slutt_tid IS NULL";

    if ($result=mysqli_query($conn,$sql)) {
        $rowcount=mysqli_num_rows($result);
    }
    while ($row = $result->fetch_assoc()){

        echo "<div> - ". $row['fornavn'] . " ". $row['etternavn'] . "
    </div>";

```

```
    }  
?>
```

## 8.2.5 Nettside for å vise historisk posisjonsdata av personell

```
<!DOCTYPE html>  
<html>  
  <head>  
    <link href="stylesheet.css" rel="stylesheet" type="text/css">  
    <script>  
      function VisInfo(id, fornavn, etternavn, tidspunkt){}  
      function VisPosisjon(){}  
      function visHistoriskData(){}  
      function MannOverBord(fornavn, etternavn, ycoord, xcoord){}  
      function clear(){}  
    </script>  
  </head>  
  <body>  
    <h3 class="overskrift">Personellkontrollsystem</h3>  
  
    <div class="outer" >  
      <p id="txtHint"><br>Velkommen til personellkontrollsystem for perso-  
nell på fartøy</p>  
  
      <div class = "LeggTilPersonellKnapp">  
        <a href="AdministrerPersonellTags.php"><button onclick = >Administrer  
personell-tag kobling</button></a>  
      </div>  
      <div class = "historiskKnapp">  
        <a href="Posisjon.php"><button onclick = >Vis real time posi-  
tion</button></a>  
      </div>  
      <div class = "AddPersonellTagsKnapp">  
        <a href="AddPersonellTags.php"><button>Legg til ny tag / perso-  
nell</button></a>  
      </div>  
  
      <div class="HistoriskData" id="HistoriskD">  
        <p>Velg Personell:</p>  
        <div id="info">  
          <p id="infoskrift"></p>  
        </div>  
  
    <div class="form">  
  
      <?php  
  
      $servername = "localhost";  
      $username = "exampleuser";
```

```

$password = "UlrikHenrik";
$dbname = "exampledb";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "<br>Personell: ";
$sql = "SELECT * FROM personell";

echo "<select name='Personell' id='SelectPersonell'>";
echo "<option value=''>Velg Personell</option><br>";

if ($result=mysqli_query($conn,$sql)) {
    $rowcount=mysqli_num_rows($result);
}
while ($row = $result->fetch_assoc()){
    echo "<option value='". $row['id'] .' '>". $row['fornavn'] ."
". $row['etternavn'] . "</option>";

}
echo "</select><br><br>";
echo "Starttid: ";
echo "<input type='datetime-local' id='starttid' name='start-
tid'>";
echo "<br><br>Sluttid: ";
echo "<input type='datetime-local' id='sluttid' name='slut-
tid'>";
echo "<br><br>";
?>
<div>
<button onclick = "visHistoriskData();">Send inn</button>
</div>

</div>
<br>
<br>
<div class="inner">
<div class="log">

<div id="posisjon">

<script>

function visHistoriskData(){

```

```

clear();
const elements = document.getElementsByClassName("circle");
while(elements.length > 0){
elements[0].parentNode.removeChild(elements[0]);
}
var personellID = document.getElementById("SelectPersonell").value;
var starttid = document.getElementById("starttid").value;
var sluttid = document.getElementById("sluttid").value;
var xhttp;

xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        document.getElementById("posisjon").innerHTML = this.responseText;
    }
};
xhttp.open("GET", "VisHistoriskData.php?personellID="+personellID+"&starttid="+starttid+"&sluttid="+sluttid+"", true);
xhttp.send();

}

function VisInfo(id, fornavn, etternavn, tidspunkt){
document.getElementById("info").style.display = "block";
document.getElementById("infoskrift").innerHTML = "Fornavn = " + fornavn + "<br>Etternavn = " + etternavn + "<br>Tidspunkt = " + tidspunkt + ".";

}

function clear(){
document.getElementById("info").style.display = "none";
document.getElementById("infoskrift").innerHTML = " ";
}

</script>
</div>
</div>
</div>
</div>
</body>
</html>

```

## 8.2.6 PHP Kode for å hente ut historisk data på valgt personell og tidsrom

```
<?php
```

```

$personellID = $_GET['personellID'];
$starttid = $_GET['starttid'];
$starttid = str_replace("T", " ", $starttid);
$starttid = str_replace("%3A", ":", $starttid);
$starttid = $starttid . ":00";
$sluttid = $_GET['sluttid'];
$sluttid = str_replace("T", " ", $sluttid);
$sluttid = str_replace("%3A", ":", $sluttid);
$sluttid = $sluttid . ":00";
$servername = "localhost";
$username = "exampleuser";
$password = "UlrikHenrik";
$dbname = "exampledb";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql2 = "SELECT p.id, fornavn, etternavn, X_Koordinat, Y_Koordinat, tid FROM personell p, tags t, tag_to_pers ttp, posisjon_historisk ph WHERE p.id = ttp.id_pers AND ph.id_brikke = t.id_brikke AND t.id_brikke = ttp.id_brikke AND p.id=" . $personellID . " AND tid > '" . $starttid . "' AND tid < '" . $sluttid . "'";

if ($result=mysqli_query($conn,$sql2)) {
    $rowcount=mysqli_num_rows($result);
}

while ($row = $result->fetch_assoc()){
    $x = $row['X_Koordinat']*18;
    $y =$row['Y_Koordinat']*18;
    $tidspunkt = str_replace(" ", "-", $row['tid']);
    echo "<div class='circle' style=left:" . $x . "px;top:" . $y . "px onclick = VisInfo(" . $row['id'] . "," . $row['fornavn'] . "','" . $row['etternavn'] . "','" . $tidspunkt . "')></div>";
}
?>

```

## 8.2.7 Nettside for å administrere personell – tag koblinger

```

<!DOCTYPE html>
<html>
    <head>
        <link href="stylesheet.css" rel="stylesheet" type="text/css">

```

```

<script>function leggstil(){}</script>
<script>function avsluttkobling(){}</script>
</head>
<body>
    <h3 class="overskrift">Personellkontrollsystem</h3>

<div class="outer" >
    <p id="txtHint"><br>Velg personell og tag for å sette opp en ny kob-
ling til personellkontrollsystemet</p>

    <div id="info">
        <p id="infoskrift"></p>
    </div>
    <div class = "historiskKnapp">
    <a href="Posisjon.php"><button onclick = >Vis real time posi-
tion</button></a>
    </div>
    <div class = "LeggTilPersonellKnapp">
    <a href="HistoriskData.php"><button>Vis Historisk data</button></a>
    </div>
    <div class = "AddPersonellTagsKnapp">
    <a href="AddPersonellTags.php"><button>Legg til ny tag / perso-
nell</button></a>
    </div>

    <div class="form">
        <?php

            $servername = "localhost";
            $username = "exampleuser";
            $password = "UlrikHenrik";
            $dbname = "exampledb";

            // Create connection
            $conn = new mysqli($servername, $username, $password, $dbname);
            // Check connection
            if ($conn->connect_error) {
                die("Connection failed: " . $conn->connect_error);
            }
            echo "<br>Personell: ";
            $sql = "SELECT fornavn, etternavn, id FROM personell WHERE aktiv
IS NULL";

            echo "<select name='Personell' id='SelectPersonell'>";
            echo "<option value=''>Velg Personell</option><br>";

            if ($result=mysqli_query($conn,$sql)) {
                $rowcount=mysqli_num_rows($result);
            }

```

```

        while ($row = $result->fetch_assoc()){
            echo "<option value='". $row['id'] ."'>". $row['fornavn'] .".
". $row['etternavn'] . "</option>";

        }
        echo "</select><br><br>";
        echo "Tag: ";
        $sql = "SELECT id_brikke FROM tags WHERE aktiv IS NULL";

        echo "<select name='Tag' id='SelectTag'>";
        echo "<option value=''>Velg Tag</option>";

        if ($result=mysqli_query($conn,$sql)) {
            $rowcount=mysqli_num_rows($result);
        }
        while ($row = $result->fetch_assoc()){
            echo "<option value='". $row['id_brikke'] ."'>".
$row['id_brikke'] . "</option>";

        }
        echo "</select><br><br>";
    ?>
<div>
<button onclick = "leggtil();">Send inn</button>
</div>
<script>
function leggtil(){
var xhttp;

var personellid = document.getElementById("SelectPersonell").va-
lue;
var tagid = document.getElementById("SelectTag").value;

xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        document.getElementById("posisjon").innerHTML = this.respon-
seText;
    }
};
xhttp.open("GET", "AdministrerPersonellTagsphp.php?Perso-
nell="+personellid+"&Tag="+tagid+", true);
xhttp.send();
location.reload();
}
</script>

```



```

<p><br><br>Velg kobling du vil avslutte:</p>
<?php

    echo "Kobling: ";
    $sql = "SELECT fornavn, etternavn, id_brikke, start_tid,
id_tag_to_pers FROM tag_to_pers ttp, personell p WHERE p.id = ttp.id_pers
AND ttp.slutt_tid IS NULL";

    echo "<select name='Tag' id='SelectKobling'>";
    echo "<option value='start'>Velg Tag-Personell kobling</option>";

    if ($result=mysqli_query($conn,$sql)) {
        $rowcount=mysqli_num_rows($result);
    }
    while ($row = $result->fetch_assoc()){
        echo "<option value='". $row['id_tag_to_pers'] .' ' - " .
$row['fornavn'] ." ". $row['etternavn'] ." - ". $row['id_brikke'] ." -
</option>";

    }
    echo "</select><br><br>";
?>

<div>
<button onclick = "avsluttkobling(document.getElementById('SelectKo-
bling').value);">Send inn</button>
</div>
<script>
function avsluttkobling(kobling){
    var xhttp;

    xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("koblingavsluttet").innerHTML =
this.responseText;
        }
    };
    xhttp.open("GET", "Avsluttkobling.php?Kobling="+kobling+",
true);
    xhttp.send();
    location.reload();
}

</script>
<p id="koblingavsluttet"></p>
<div>
</div>

```

```

    </div>
  </div>
</body>
</html>

```

## 8.2.8 PHP kode for å lage ny kobling mellom person og tag

```

<?php

    $servername = "localhost";
    $username = "exampleuser";
    $password = "UlrikHenrik";
    $dbname = "exampledb";
    $idbrikke = $_GET['Tag'];
    $personellid = $_GET['Personell'];

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "INSERT INTO tag_to_pers (start_tid, id_brikke, id_pers)
VALUES (now(),'" . $idbrikke . "', " . $personellid . ")";

    if ($conn->query($sql) === TRUE) {
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
    $sql = "UPDATE tags SET aktiv = 1 WHERE id_brikke = '" . $idbrikke
. "'";

    if ($conn->query($sql) === TRUE) {
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
    $sql = "UPDATE personell SET aktiv = 1 WHERE id = '" . $perso-
nellid . "'";

    if ($conn->query($sql) === TRUE) {
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
?>

```

## 8.2.9 PHP kode for å avslutte kobling mellom valgt personell og tag

```
<?php
```

```
    $servername = "localhost";
    $username = "exampleuser";
    $password = "UlrikHenrik";
    $dbname = "exampledb";
    $kobling = $_GET['Kobling'];

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "UPDATE tag_to_pers SET slutt_tid = now() WHERE
id_tag_to_pers = " . $kobling . """;

    if ($conn->query($sql) === TRUE) {
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
    $sql = "UPDATE tags t INNER JOIN tag_to_pers ttp ON t.id_brikke =
ttp.id_brikke SET aktiv = NULL WHERE ttp.id_tag_to_pers = " . $kobling
."";
    if ($conn->query($sql) === TRUE) {
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
    $sql = "UPDATE personell p INNER JOIN tag_to_pers ttp ON
ttp.id_pers = p.id SET aktiv = NULL WHERE ttp.id_tag_to_pers = " . $ko-
bling . """;

    if ($conn->query($sql) === TRUE) {
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}

?>
```

## 8.2.10 Nettside for å legge til ny tag eller nytt personell

```
<!DOCTYPE html>
<html>
  <head>
    <link href="stylesheet.css" rel="stylesheet" type="text/css">
    <script>function leggtil(){}</script>
    <script>function avsluttkobling(){}</script>
  </head>
  <body>
    <h3 class="overskrift">Personellkontrollsystem</h3>

    <div class="outer" >
      <p id="txtHint"><br>Legg til ny Tag</p>

      <div id="info">
        <p id="infoskrift"></p>
      </div>
      <div class = "historiskKnapp">
        <a href="Posisjon.php"><button onclick = >Vis real time posi-
tion</button></a>
      </div>
      <div class = "LeggTilPersonellKnapp">
        <a href="HistoriskData.php"><button>Vis Historisk data</button></a>
      </div>
      <div class = "AddPersonellTagsKnapp">
        <a href="AdministrerPersonellTags.php"><button onclick = >Administrer
personell-tag kobling</button></a>
      </div>

      <label for="idbrikke">Id til brikke: </label>
      <input type="text" id="idbrikke" name="idbrikke"><br><br>

      <div>
        <button onclick = "leggtilidbrikke();">Send inn</button>
      </div>

      <script>
function leggtilidbrikke(){
var xhttp;

var idbrikke = document.getElementById("idbrikke").value;

xhttp = new XMLHttpRequest();
xhttp.open("GET", "AddIdbrikke.php?idbrikke="+idbrikke+", true);
xhttp.send();
location.reload();
}
</script>
<p><br><br>Legg til nytt Personell</p>
```

```

<label for="fornavn">Fornavn: </label>
<input type="text" id="fornavn" name="fornavn"><br><br>
<label for="etternavn">Etternavn: </label>
<input type="text" id="etternavn" name="etternavn"><br><br>
<label for="ansattnummer">Ansattnummer: </label>
<input type="text" id="ansattnummer" name="ansattnummer"><br><br>
<div>
<button onclick = "leggtilpersonell();">Send inn</button>
</div>
<script>
function leggtilpersonell(){
    var xhttp;
    var fornavn = document.getElementById("fornavn").value;
    var etternavn = document.getElementById("etternavn").value;
    var ansattnummer = document.getElementById("ansattnummer").value;
    xhttp = new XMLHttpRequest();

    xhttp.open("GET", "AddPersonell.php?fornavn="+fornavn+"&etter-
navn="+etternavn+"&ansattnummer="+ansattnummer+", true);
    xhttp.send();
    location.reload();
}

</script>
<p id="koblingavsluttet"></p>
<div>
</div>
</div>
</body>
</html>

```

## 8.2.11 PHP kode å legge til nytt personell

```

<?php

$servername = "localhost";
$username = "exampleuser";
$password = "UlrikHenrik";
$dbname = "exampledb";
$fornavn = $_GET['fornavn'];
$etternavn = $_GET['etternavn'];
$ansattnummer = $_GET['ansattnummer'];

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

```

```

    }

    $sql = "INSERT INTO personell (fornavn, etternavn, ansattnummer)
VALUES ('" . $fornavn . "','" . $etternavn . "','" . $ansattnummer . "')";

    if ($conn->query($sql) === TRUE) {
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
?>

```

## 8.2.12 PHP kode for å legge til ny brikke

```

<?php

$servername = "localhost";
$username = "exampleuser";
$password = "UlrikHenrik";
$dbname = "exampledb";
$idbrikke = $_GET['idbrikke'];

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO tags (id_brikke) VALUES ('" . $idbrikke . "')";

if ($conn->query($sql) === TRUE) {
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
?>

```

## 8.3 CSS kode for brukergrensesnittet

```
.overskrift{
  text-align: center;
}
.outer {
  width: 80%;
  height: 500px;
  border: 3px solid black;
  margin: auto;
  background-color: #e6e6e6;
  text-align: center;
}
.form {
  position: absolute;
  margin: auto;
  left: 40%;
  top: 20%;
}
#info{

  width:180px;
  left:70%;
  top:13%;
  margin: 0 auto;
  position: sticky;
  border: 3px solid orange;
  display:none;
}
#info2{

  width:180px;
  top:13%;
  margin: 0 auto;
  position: sticky;
  border: 3px solid orange;
  display:none;
}
#infoskrift{
  color : orange;
}
.boat{
  position:inherit;
  width:600px;
  display: block;
}
.circle {
  position:absolute;
  top: 8px;
  left: 8px;
```

```

        width: 10px;
        height: 10px;
        -webkit-border-radius: 5px;
        -moz-border-radius: 5px;
        border-radius: 5px;
        background: red;
    }
    .circle2 {
        position: absolute;
        top: 12px;
        left: 12px;
        width: 14px;
        height: 14px;
        -webkit-border-radius: 8px;
        -moz-border-radius: 8px;
        border-radius: 8px;
        background: red;
    }
    .posisjon{
        position: absolute;
        height: inherit;
        width: inherit;
        display: block;
        margin-left: auto;
        margin-right: auto;
    }

    .inner {
        width: auto;
        position: absolute;
        height: auto;
        margin: 0 auto;
        width: 50%;
        top: 45%;
        left: 30%;
        background-color: #e6e6e6;
        cursor: crosshair;
        user-select: none;
        -moz-user-select: none;
        -ms-user-select: none;
        -webkit-user-select: none;
    }

    .log {
        position: relative;
        width: 100%;
        text-align: center;
    }
    .PersonellOmbord{

```



```
    position:absolute;
    left:75%;
    top:100px;
    padding: 10px;
    border: 3px solid black;
}
#Mannoverbord{
    position:absolute;
    left:72%;
    padding: 10px;
    top:200px;
    color: red;
    font-size:X-large;
    font-weight:bold;
}

.historiskKnapp{
    position:absolute;
    left:15%;
    top:100px;
}
.LeggTilPersonellKnapp{
    position:absolute;
    left:15%;
    top:140px;
}
.AddPersonellTagsKnapp{
    position:absolute;
    left:15%;
    top:180px;
}
```

