

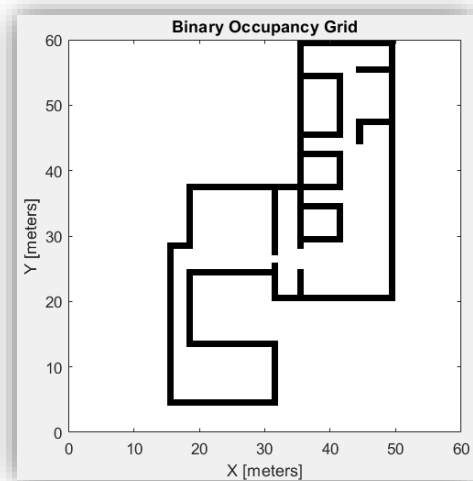


AALBORG UNIVERSITET

ROBUST TIDSSYNKRONISERING MELLEM RADIOLINKMÅLINGER OG POSITIONERING AF MOBILE ROBOTTER I KANALMÅLING.

René Dam Marcker – Kommunikationsteknologi

Kandidat speciale





AALBORG UNIVERSITET

Institut for Elektroniske Systemer
Aalborg Universitet
<http://es.aau.dk>

Titel: Robust tidssynkronisering mellem radiolinkmålinger og positionering af mobile robotter i kanalmåling.

Tema: NTP, protokoller, distribueret system, 5G, I4.0, kanalmåling, ROS, simulering, realtid.

Projekt Periode:
Sommer Semester 2021

Gruppe Medlem(er):
René Dam Marcker

Vejleder(er):
Troels B. Sørensen (AAU)
Poul Olesen (Nokia Bell Labs – Aalborg)

Antal Sider:
48 sider + bilag

Afleveringsdato:
2. Juli 2021

Abstrakt:

Denne rapport er en analyse om nøjagtighederne i en kanalmålemetode, hvor der anvendes mobile enheder for antennerne.

Analysen kigger på hvor i delsystemerne der opstår afvigelser i målemetoden og hvor meget metoden bliver påvirket i forhold anvendelsen til modellering af kanalen.

Der tages udgangspunkt i en RF-målemetode til at vurdere om kanalmålemetoden opfylder de krav for anvendelse.

Den største udfordring med kanalmålemetoden er tidssynkronisering mellem positioner og RF-målingerne, hvilket har den største unøjagtighed i form af asymmetrisk netværk kommunikation.

99% af tiden er synkroniseringen lavere end 15 ms mellem TX og RX, hvilket er lavere end den givet grænse på 50 ms.

Indholdet af denne rapport er frit tilgængeligt, men publikationer (med reference) må kun ske med samtykke med forfatteren.

ABSTRACT

This project is about a channel-sounding method, which utilizes industrial robots to mobilize the RX and the TX antennas. The scope of the project is to analyze the dominant outliers, where they occur and how their effect the channel-sounding method.

The method use the brand MiR for the mobile robots, which are running the framework ROS. The framework ROS make it possible to get two kind of positions data, the global position and the relative position. From the relative position data, the linear and angular velocity is used to interpolation between the global positions data and add velocity to the global positions data. To extract the data, each MiR robot get a Raspberry Pi connected.

Measurement and test of the method is conducted in Smart Production Lab on AAU [Fibigerstræde 14, , 9220 Aalborg, Denmark]. MiR have its own web based interface, where missions can defined and added to the queue.

NTP is used for synchronising the RF part with the position system. Looking at the position data after outliers, it seems impossible to find as the results is the output of filters like a Kalman filter.

To get a better understanding of the NTP, the analyse went deep down the algorithms inside the protocol. To get an understanding of the protocol, a simple simulation were made to investigate the different parameters, like wandering of the clients clock, different between adding the combined jitter and not to. The biggest reveal were how most a asymmetric network effect the behaviour of the synchronization.

The project explore also the idea of a one UI to united every processes, as starting the system require open three terminal, one remote control and two browser interfaces.

The project also designed a specific structure to controlling the data before processing. A measurement campaign were conducted, which concluded in results regarding NTP, position, interpolation method and the applicating of a definition to categories measurement to be LOS.

NTP were measured in real-time, by clients constant sending timestamp to the central pc while the campaign were active. The mean offset between the two clients were at 2.73 ms and had a maximum offset of 14.89 ms, but 99% of the time the offset were 10 ms or lower.

The results from the measurement campaign seems similar to the results from the simulation. in the results of the distance between RX and TX, some outliers were found. But, it were only 8 outliers, which had a mean offset about 3.57 cm.

FORORD

Jeg, René Dam Marcker har udarbejdet denne rapport, som dokumentation for mit kandidatspeciale i kandidatprogrammet, "*Kommunikationsteknologi (specialisering i Netværk og Distribueret Systemer)*" på AAU, i sommer semesteret 2021. Min baggrund for dette speciale er en bachelor i "*Internetteknologier- og Computersystemer*" fra AAU i 2019.

Projektet tager udgangspunkt i et tidligere semester projekt, "**5G Channel Sounding Between Mobile Robots in an Industrial Setting @3.57 GHz and 26.43 GHz**" [1]. dette projekt fokuserer på selve kanalmetoden som platform og ser bort fra RF delen, da det ses som en tilkobling til platformen.

Der skal rettes en speciel tak til min vejleder, **Troels Bundgaard Sørensen** (lektor i WCN-afdelingen på elektroniske systemer under AAU), for hans vejledning og støtte under hele projektføreløbet. Troels Sørensen har været en kæmpe hjælp gennem gode diskussioner til vejledermøder og hurtige besvarelser på mails.

Udover hovedvejleder, skal der også en kæmpe tak til RF-afdelingen i Nokia Bell Labs Aalborg for deres kæmpe støtte og vejledning. Specielt tak til **Bent Ryesgaard** (afdelingsleder), **Paul Olsen** og **Stig Blucher Brink** for deres vejledning og feedback, hvilket har gjort en kæmpe forskel på formningen og strømliningen af projektet.

INDHOLD

ABSTRACT	3
FORORD	4
1 MOTIVATION	8
2 PROBLEMANALYSE	10
2.1 Eksisterende målemetoder og kanalmåling.....	10
2.2 Kanalundersøgelse, overblik og kanalmålinger i produktionsfaciliteter	12
2.3 Positionering system.....	14
2.4 Tidssynkronisering	15
2.5 Tidssynkroniseringsprotokoller	16
2.6 Brug af kanalmåling metode sammen med multipel tone metode	17
3 PROBLEMFORMULERING	19
4 PROBLEMAFGRÆNSNING	20
5 KANALMÅLING MED MOBILE ENHEDER	21
5.1 De mobile enheder	21
5.2 Data fra MiR – ROS	22
5.3 MiR Fleet – ruteplanlægning.....	22
5.4 NTP	25
6 ANALYSE AF USIKKERHEDERNE	26
6.1 Problemstilling med tidssynkronisering forhold til NTP	26
6.2 Analyse af protokollen i NTP	28
6.2.1 Peer procedure.....	28
6.2.2 System procedure	29
6.3 Simuleringstest af NTP værdierne	30

7	DESIGN AF FULDLIG AUTOMATISERET SYSTEM	31
7.1	Back-end - Scripts	32
8	DATA ANALYSE	34
8.1	Struktur over mapper og filer	34
8.2	Rå data til RF måling med positionsdata	35
9	RESULTATER.....	38
9.1	Tidssynkronisering	38
9.2	Position	39
9.3	Metode til at finde LOS->NLOS.....	41
10	DISKUSSION.....	43
10.1	Traditionel kanalmåling	43
10.2	Synkronisering af robotpositioner og RF målinger	44
10.3	NTP topologien	45
10.4	MiR som en del af platformen	45
11	KONKLUSION	46
12	REFERENCER	48
13	APPENDIKS	50
13.1	Resultater fra simulering af NTP	50
13.2	1.forsøg.....	50
13.3	2. forsøg.....	50
13.4	3. forsøg.....	51
13.5	4. forsøg.....	51
13.6	RF måleudstyr og opsætning	52
13.7	SSH key gen	53
13.8	Resultater	53

Robust synkronisering mellem radiolinkmålinger og positionering af mobile robotter i kanalmåling.

1 MOTIVATION

Alt trådløs kommunikation bygger på modeller over den anvendte kanal, hvilket beskriver egenskaberne af de medvirkende systemer. Kanalmodeller varierer mellem simplificeret modeller, hvor nogle systemer er ekskluderet til mere sofistikeret. For eksempel model over path loss; der er svinger fra udelukkende at tage afstand som eneste faktor, til udvide med refleksioner, højde af antennerne og type af omgivelser [2].

Specielt i denne tid er der brug for nye kanalmålinger, da trådløs kommunikation er i gang med et epokeskift i form af femte generation mobile trådløse netværk (5G) og alle kanalmodeller er baseret på kanalmåling. I takt med udviklingen af teknologien, øges kompleksiteten af kommunikationssystemerne og krævet kanalmodel, hvilket øger antal elementer til en kanalmåling metode.

Tilbage i 1960'erne skulle kanalmåling metoden kun tage højde for at måle feltstyrken. Overgangen til bredbånd systemer krævede en udvikling af kanalmåling metoderne, sådan de også kunne måle forsinkelsesspredningen. I 1990'erne var der fokus på egenskaber ved retningsbestemt radioudbredelse og første til interesse for multiple antenne systemer. Kanalmåle metoden blev derfor i stand til at kunne måle flere impulsrespons på samme tid [2].

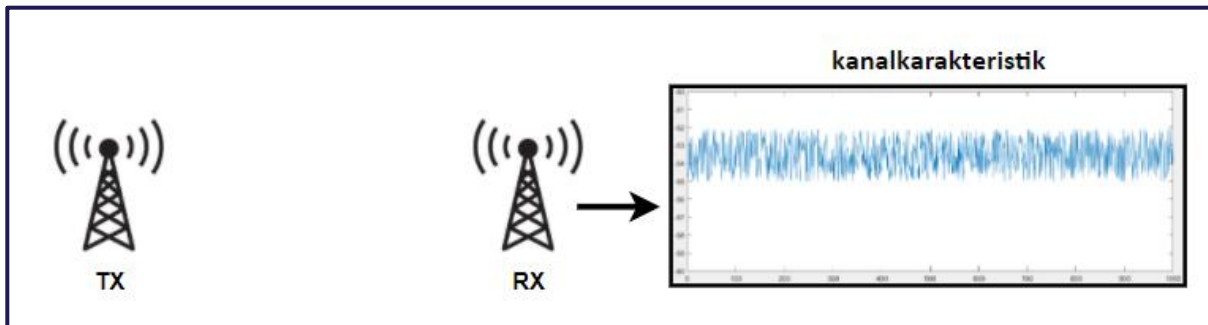
For at opnå kravspecifikationerne for 5G, skal der integreres en masse nye teknologier til at fungere i 5G. Device-to-device (D2D) kommunikation er en af nøgleteknologierne givet til at forbedre 5G ydelse præsentation og forme fremtidens trådløse kommunikation. D2D er direkte kommunikation mellem to enheder uden brug af nogen centrale point (for eksempel base station). Det er en foretrukket teknik til at forbedre energieffektiviteten, data kapaciteten, latens og udnyttelsen af spektrum i det mobile netværk [3].

D2D teknologien var allerede en del af tidligere mobilnetværk generationer, men ikke rigtig blevet anvendt udover nogle meget specifikke scenarier. D2D kommunikationen indebærer, at kommunikationen sker uden en base station. Der vil være scenarier hvor D2D par er begge mobile, hvilket stiller krav til peer Discovery, handover, styring og optimering for allokation, samt flere krav til kanalmåling metoden [4].

I kanalmåling indgår to systemer; RF-måleudstyr og platformen. RF-måleudstyret er afhængig af valgt antenne, hvilket giver forskellige muligheder for RF-måling og dermed kanal karakteristik. Det er selve platformen, som skal udvikles til at få mest indsigt i kanal karakteristik, da RF-måleudstyret lægges oven på platformen. Metoden til kanalmåling vurderes på, dens nøjagtig til at afspejle virkeligheden, i forhold til fejlkilder i systemet.

Den mest basale kanalmåling består af et afsendermodul (TX), hvor RF-målestyret bygger på en signalgenerator og en antenne, og et modtagermodul (RX) til at modtage signalet. Figur 1 viser opsætningen af den mest basale kanalmåling. Signalet ved RX

kan bruges til at analysere kanalkarakteristikken for eksempel via signaleffekten til path loss model.



Figur 1: Den mest basale opstilling for kanalmåling, en stationær afsender TX og en stationær modtager RX, samt en skitse af noget kanalkarakteristik, som bruges til kanalmåling.

Kanalmåling er lokation specifikt, hvilket betyder at kanalmodel ikke kan bygges alene på RF signaler. Det kræver data fra omgivelserne, hvilket giver information om; afstanden mellem antennerne, og afstande til inventar samt vægge. Disse målinger af omgivelserne kan fortælle om mulige signalveje via refleksioner eller om det er LOS signal. Løsning kunne være at data fra omgivelserne er baseret på et kort over miljøet, som bliver delt ind i et koordinatsystem og måler antennernes positioner i koordinatsystemet, hvor afstande kan udregnes.

En anden løsning til en målemetode, består i at gøre både TX og RX til mobile enheder, som kører en præ-defineret rute samt RF systemet kørende. Denne metode vil også emulere D2D kommunikation.

Den mobile platform kræver flere delsystemer, hvilket introducerer nye udfordringer og usikkerheder, specielt at det hele inddrager et tidselement. TX kører en rute, som er uafhængige forhold til ruten som RX kører, samt kører RF-systemet ligeså uafhængigt.

Målet ved målemetoden er at hvert signal modtaget i RF-systemet ved RX, skal positionerne for TX og RX kendes. Dette kræver at de uafhængige systemer har et fælles referenceforløb.

2 PROBLEMANALYSE

Dette kapitel er en analyse af hvordan den generelle metode for kanalmåling er opbygget, designet, og processen i den praktiske udførelse. Analysen af den mest hyppige målemetode for kanalmåling er baseret på arbejdet fra "National Institute of Standards and Technology", hvor der også ses på hvordan målingerne bliver håndteret forhold til kanalmodellering. Herefter analyseres, hvordan den hyppige metode bliver udført under selve målekampagnen.

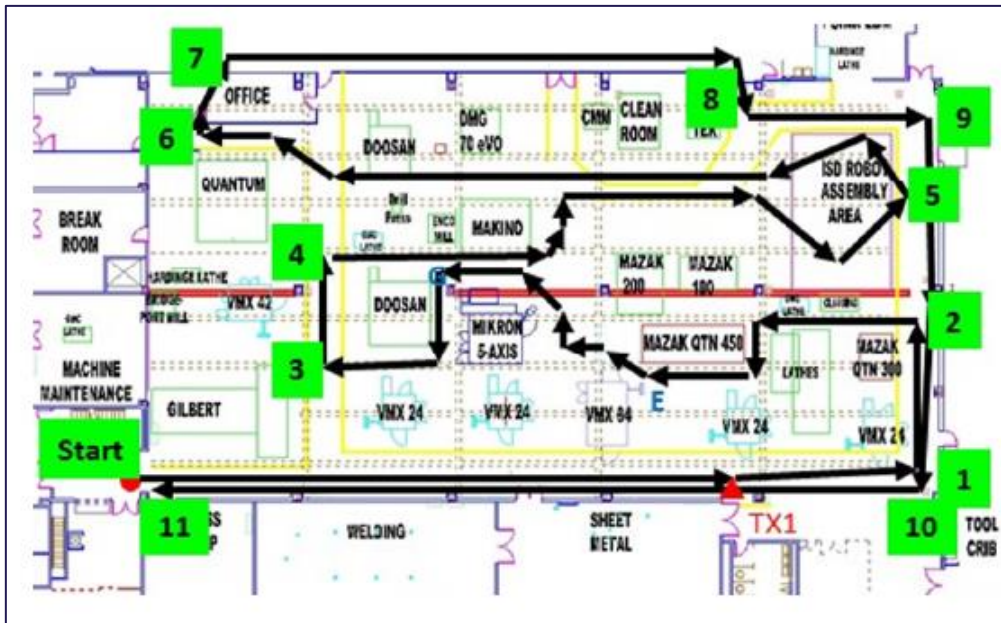
Til sidst i analysen er et afsnit, som handler om en case for kanalmåling, hvor det omhandler en unik metode for at lave radiolink og omfatter D2D kommunikation og 5G. denne case af kanalmåling er udarbejdet af Nokia Bell Labs.

2.1 EKSISTERENDE MÅLEMETODER OG KANALMÅLING

National Institute of Standards and Technology (NIST) har udarbejdet en teknisk rapport [5] (nr. 1951), som takler de udfordringer der følger, ved implementering af trådløs kommunikation i industrielle omgivelser med fokus på at anvende eksisterende teknologier.

Metoden for denne teknisk rapport er målekampagnerne blandt tre forskellige industrielle omgivelser. Rapporten giver en bedre forståelse for implementering af trådløse løsninger samt kommunikation for fremtidens produktionssystemer, der kræver ultra lav latens og høj pålidelighed. Disse effekter undersøges og forstås bedre gennem fysiskbaseret modellering korreleret med virkelige målinger.

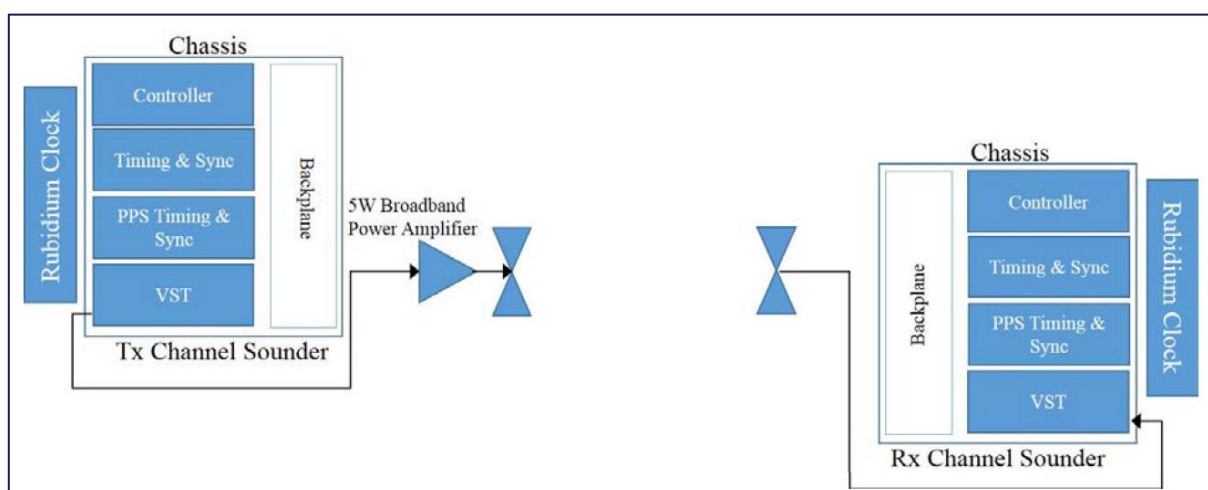
I rapporten bliver de forskellige omgivelser beskrevet i forhold til størrelse og indhold, hvor nogle ruter blev valgt baseret på mulighed for kommunikation mellem enheder og hvor de befinder sig, for eksempel AGV. Nogle ruter repræsenterer scenarier, hvor der er mange refleksioner eller få line-of-sight (LOS) path til antennemodtageren. Figur 2 illustrerer en af omgivelserne for en målekampagne.



Figur 2 Grundplan af en fabrik for måling kampagne, hvor TX1, TX2 og TX3 markerer steder for afsender antenner. [5]

NIST har udviklet en kanalmåling metode, som er en Pseudo Noise (PN) sekvens korrelation system. Systemet samler synkroniseret radioudbredelses målinger, som understøtter frekvenser fra 65 MHz til 6 GHz med op til 250 MHz båndbredde. Systemet består af én afsender (TX) med en power amplifier og én modtager (RX), hvor de er koordineret med synkronisering via rubidium clocks.

Figur 3 er afsender systemet og modtager systemet. Den afsendte PN sekvens af digitale symboler moduleres med en Binary Phased-Shift Keying (BPSK) signal, som opkonverteres til RF bærefrekvensen. Ved modtageren, bliver signalet nedkonverteret til aflæsning og gemt til efterbehandling. Hardwarets påvirkning er fjernet, ved brug af back-to-back målinger, før selve målekampagnen.



Figur 3: Blokdiagram over kanalmåle systemet, afsender og modtager [5]

Målingerne blev lavet ved placeringen for afsenderen var statisk, mens modtageren flyttes manuelt af mennesker rundt på en prædefineret rute, med en konstant hastighed.

Radioudbredelsesværdierne noteres hver gang modtageren passerer et checkpoint, hvor målinger sluttet samme sted som begyndte.

2.2 KANALUNDERSØGELSE, OVERBLIK OG KANALMÅLING I PRODUKTIONSFACILITETER

Den teknisk rapport af NIST [6] (nr. 1979), tager udgangspunkt i deres kanalmåling system, som beskrevet i **afsnittet Eksisterende målemetoder og kanalmåling**. Denne rapport har lidt andet fokus og beskriver andre aspekter end den tidligere rapport.

Målet for rapporten er at hjælpe med at udvikle retningslinjer, for bedste praksis til at integrere ydeevneevaluering af sikre trådløse platforme i dynamiske produktionsmiljøer. Metoden er baseret på de målinger og udbredelsesmodeller, som er udviklet til deres korrelationsbaserede kanalmåling system. Disse retningslinjer er ment til at blive brugt af producenter, teknologiudbydere og løsningsudbydere til at designe, implementere og vurdere robuste, sikre integrerede trådløse løsninger til brug i smart fremstilling.

En af de ting, som denne rapport fokuserer mere på, er vektorsignal generatorens indstillinger og systemet for tidsynkroniseringen. **Vektorsignal generatoren** er indstillet til at have en maksimal udgangseffekt på +10 dBm med en faktor på fire, hvilket giver 8188 målinger ved en 5 nanosekund/symbolsamlingshastighed. Dette giver en registrering, af et enkelt kodeord (400 PN-sekvenser), en varighed på 16,37 millisekunder.

Den anden del er **tidsynkroniseringen**, hvilket er et meget kritisk system til at kunne lave virkelighedstro udbredelsesmodeller baseret på målinger. Dette system bruger rubidium-clocks, som kan ses i Figur 3, til at synkronisere mellem afsenderen og modtageren, samt alle moduler i transmissionssystemet. Afsenderen og modtageren anvender et 10 MHz rubidium-ur (NI PXI 6674T), som deler tiden og udløser mellem modulerne i kabinettet.

Før en målekampagne begynder, skal tidsynkroniseringen initialiseres. Dette sker ved at forbinde en timing-synkroniseringsenhed (NI PXI 6683H) til de to rubidium-ure. Denne genererer udløser fra PPS-signalet til rubidium-urene og disciplinerer deres temperaturkompenserede krystaloscillator (TCXO). De vil være forbundet i 24 timer før målekampagnen, for at minimere usikkerheder.

Når målekampagnen skal begynde, frakobles synkroniseringsenheden. Tilbage er det rubidium-urets PPS på afsenderen, som bruges til referencetrigger til at starte signalgenerering og rubidium-urets PPS til trigger af optagelse hos modtageren. Denne tilgang skulle minimere fordelingen af jitter og minimerer fase- og tidsdrift.

Til selve målekampagnerne, blev der udført en mobil opsætning og en stationær opsætning.

Ved den **stationær opsætning**, var det afsenderen som var stationær og modtageren stod stationær ved et checkpoint, hvor målinger er middelværdi ved forskellige positioner af antennen på toppen af vognen.

I den **mobile opsætning**, var modtageren den mobile del, som blev flyttet via en prædefineret rute med afmærket checkpoints. Hele miljøet har et globalt kort, hvor de afmærket checkpoints har kendte koordinater (x, y).

Til udførelsen af denne opsætning, kræves 2-3 personer. En person til at skubbe vognen med modtageren rundt på ruten, hvilket sker ved gå-tempo. En anden til at anrabe når vognens ene fronthjul rammer afmærkningen samt nummer, og sidste person nedskriver værdier i en testjournal. Figur 4 viser et eksempel på en testjournal.

TU

2 Mobile Receiver Test omni-Tx antenna for 2-9-16

Test Run Name	TX Location Room	Frequency	Date	Time	Attenuator Factor (dB)	PN Chip Rate	Measurement Sample Rate
TX2 - 2G - Butler - Vpolar - 12.5	TX2 - 1.5	2.358.5	2-9-2016	12:59.7	50.0	11.5	200.0
Complex Reference	Waveform	-ref0.142					

Check Point	Acquisition Number	Check Point	Acquisition Number
1		21	2.22
2	56	22	2.38
3	67	23	2.46
4	89	24	2.55
5	89	25	2.60
6	720	26	2.66
7	1.6	27	2.71
8	1.86	28	2.73
9	1.86	29	2.82
10	1.79	30	2.82
11	1.74	31	2.82
12	1.55	32	2.82
13	1.62	33	2.80
14	1.72	34	2.80
15	1.79	35	2.81
16	1.87	36	2.80
17	1.99	37	2.85
18	2.03	38	
19		39	
20	2.02	40	

Notes:

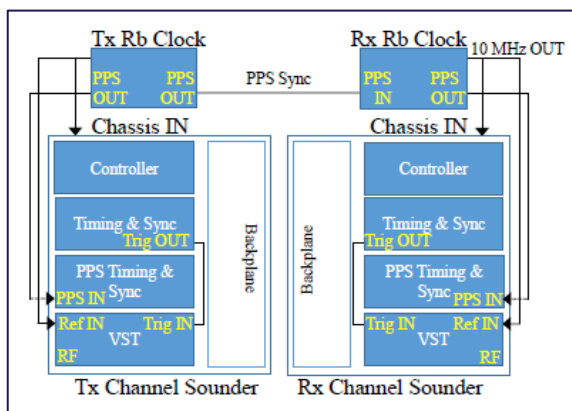
Temp: 35.69 Tx Power: 2.3W Capable met
 Pga: 40 Ant level: on
 Final par: 2.3W 37 dBm Final ref: 1.0 dBm/100m
 Set gain: 70% Stop gain: 375
 End: 1:10 Tx antenna: omni-antenna

Figur 4: Eksempel på testjournal med data. [6]

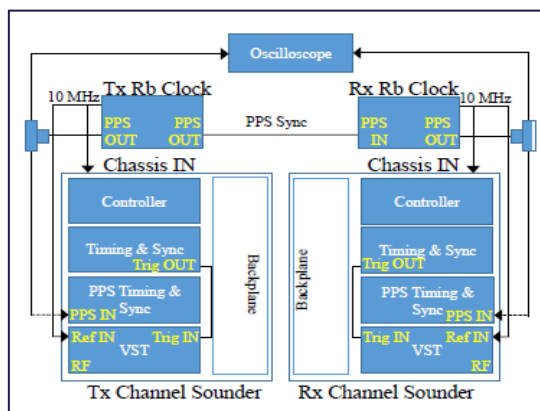
Før målekampagne kan påbegyndes skal rubidiumurene klargøres. Denne proces indebærer at stabilisere urene forhold til hinanden og komponenter på tværs af systemer, samt måle driften af urene. Stabilisering af rubidiumurene skal ske 72 timer for målekampagnen. Opsætningen er vist i Figur 5.

Ydermere skal driften mellem TX-rubidium uret og RX-rubidium uret måles, illustreret i Figur 6. Denne set-up er tilnærmelsesvis ens for stabilisering, den store forskel ligger i brugen af oscilloskopet til at måle driften.

Der bruges et software kaldt Lab VIEW til at disciplinere uret på RX-systemet og TX, samt sætte tid referencen. Softwaret har en funktion kaldt "udregnet fra tid referencen" vs. Tiden.



Figur 5: Stabilisering af rubidiumurene. [6]

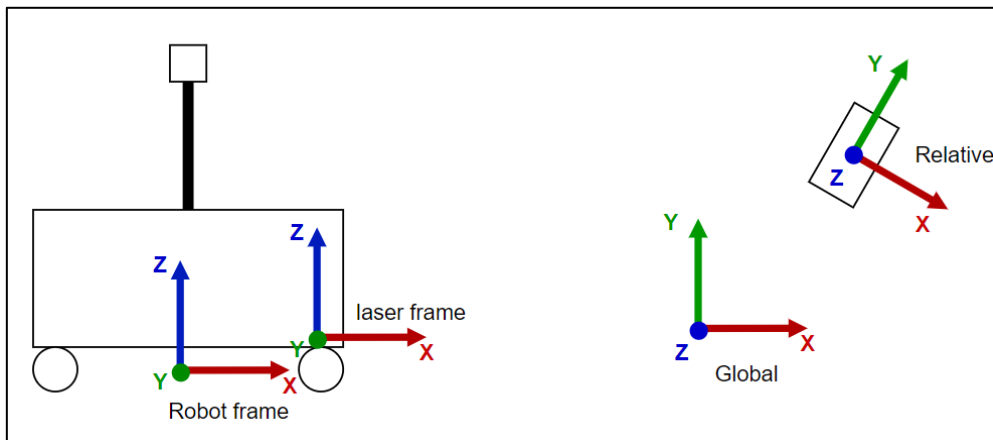


Figur 6: Setup for måling af drift. [6]

2.3 POSITIONERING SYSTEM

Positionssystemet bygger på ROS i MiR, hvor der bliver brugt to positions niveauer, de globale og de relative. Data begynder ved sensor, såsom LiDAR, hvilket defineres som laser frame, hvor der en masse algoritmer anvendes til at bestemme en position, for eksempel kunne være et Kalman filter. Figur 7 viser laser frame forhold til robotten og de andre frames.

Positionen udregnet ved laser frame er for sensors placering, men det er robotens position som er relevant, hvilket robot frame er det næste niveau. Der sker en transformation mellem laser frame og robot frame.

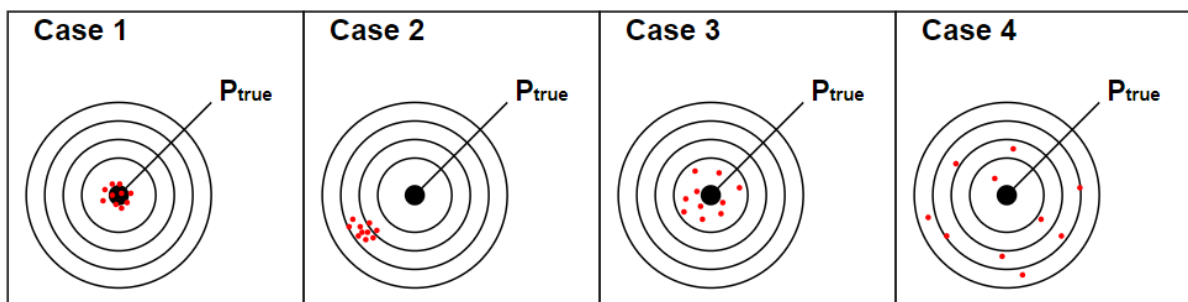


Figur 7: Frames forhold til position af robotten.

Næste niveau er den relative position, hvilket tager udgangspunkt i begyndelsespunktet. Fra relative frame laves der en transformation til den globale frame og til kort i slutningen.

Softwaret i MiR bruger ROS til at lave alle disse transformationer, hvilket ikke er til gængelige for undersøgelse. Der vil være støj på sensor og kontrollers, men MiR har som nævnt indbygget filter til at tage højde for det.

Nøjagtigheden kan ses på via de fire hyppige tilfælde af præcisionen af et positionssystem, som vist i Figur 8.



Figur 8: De fire cases vedrørende nøjagtigheden af positionssystemet. En normalisering af alle målinger lagt op imod den rigtige position.

I alle cases kan offsetten defineres til at være afvigelsen til tyngdepunktet af målingerne eller den maksimale offset. Case 1 viser det mest optimale resultat, da det viser både at være præcis og nøjagtig. Case 2 er case 1 med en konstant afvigelse, hvilket er lige med afstanden mellem P_{true} og tyngdepunktet for punkterne, som kan lægges til målingerne til at komme til case 1.

Case 3 og case 4 er en smule værre, da de har en meget højere spredning, specielt i case 4 hvor måske en sandsynlighedstæthedsfunktion skal laves til at bestemme σ_p . Disse cases er dog dem, hvor et Kalman filter vil komme ind og justere eller fjerne de afvigelser, som ikke stemmer overens med historikken, nuværende hastighed og position.

Når platformen skal bruge til kanalmåling, skal der også tages højde for antennens placering forhold til positionspunktet på robotten. Hvis antennen er placeret lige over positionspunktet på robotten, vil der minimum være en offset i z-aksel.

Tilføjelse, når der er bruges en retningsbestemt antenne, er orientering en del af positionssystemet, hvilket en unøjagtighed ved placeringen af antennen forhold til robotens orientering. Robotens orientering har også unøjagtigheder i dens værdier fra sensorer og kontroller.

Når man udtrækker positionsdata fra MiR via ROS, kommer x- og y-værdierne i meter med 14 decimaler, hvilket rent logisk er urealistisk uden at være nogle uden decimaler fra støj. Brugen af positionsdata til denne kanalmåling er det kun relevant med 3 decimaler, sådan det er nede på millimeter præcision.

Det er oplagt at lave målinger, hvor ved se om der er nogle afvigelser, som kommer gennem diverse filter og hvor meget afvigelse er det.

2.4 TIDSSYNKRONISERING

Ideelt er synkronisering af to ure (interne computer ure) en meget simpel opgave, hvor to ure skal kun synkroniseres en gang og derefter vil forblive synkroniseret. Problemet med synkroniseringen er at de to ure har forskellige ur faser, da de har forskellige fase afvigelse.

De to mest nøjagtige løsninger er hardware løsninger, såsom GPS-enheder og ure med rubidium-frekvensstandarder (almindeligvis kendt som atomure), hvilket har henholdsvis synkroniseringsfejl på under 10 ns og endnu lavere. Men disse løsninger har begge en meget høj energibehov og koster meget, hvilket ikke kan betale sig for de fleste applikationer, specielt uegnet for indlejrede og distribuerede systemer, hvor alle enheder skal have en. GPS-enheder er også begrænset til specielle miljøer, hvor indendørs ikke er blandt.

Langt de fleste systemer bruger en form for krystaloscillatorer som kilde til det interne ur, hvilket har en anstændig præcision, lille størrelse og lav omkostninger.

Krystaloscillatorer er typisk sammensat af en lille præcisionsfremstillet kvarts (SiO_2), som resonerer ved en bestemt frekvens på grund af den piezoelektrisk effekt af kvarts.

Ved en krystaloscillator kan denne frekvens afvige fra den tilsigtede frekvens på grund af små produktionsfejl og forskellige miljøforhold, herunder temperatur, lufttryk og fugtighed. Denne afvigelse er defineret som en drift over tid, da de miljømæssige virkninger kan ændres og sammensættes, jo mere tid der går. Denne drift er skadeligt for systemer, som kræver nøjagtig datatidsstempling.

Tidssynkroniseringsprotokoller tillader systemer at bevare fordelene ved deres krystaloscillatorer og samtidig opretholde synkroniserede ure gennem netværksprotokol, hvormed to eller flere noder trådløst kan synkronisere deres interne ure. Målet med tidssynkronisering er at afhjælpe driften af uret via netværksprotokoller, der justerer et netværks interne ure for at skabe konsistens. Det er værd at bemærke, at selve det interne ur ikke ændres via en tidssynkroniseringsprotokol, der bliver beregnet et lokalt tidsoffset, som bliver tilføjet til den interne ur værdi for at beregne det aktuelle tid.

Tidssynkroniseringsprotokoller er lette, skalerbare og i stand til at synkronisere et netværks ure i tilstrækkelig grad med få transmissioner af netværksbeskeder og uden større forstyrrelses af systems primære applikationer.

2.5 TIDSSYNKRONISERINGSPROTOKOLLER

Der er tre tidssynkroniseringsprotokoller, som er mest anvendt; PTP (Precision Time Protocol), NTP (Network Time Protocol) og SNTP (Simple Time Protocol). Princippet og metoden er den samme ved alle tre, men den lille forskel i anvendelsesområde gør det store forskel.

PTP er den mest nøjagtig metode, hvor den er nede på sub-mikrosekunder, men den er ikke anvendelig i trådløs opsætning da protokollen antager der er en stabile forbindelse mellem klient og server. PTP kræver også specifikt hardware til enhederne og specifikke router, som understøtter PTP. I fremtiden vil det være muligt med PTP i takt med 5G giver den samme stabile forbindelse som kabel.

Tilbage er NTP og SNTP, hvor forskellen er lille men signifikant. Mens NTP har øvre grænse på offset, hvor den vil køre synkronisering på og den har en grænse på 500 PPM, hvor meget justering den vil lave, vil SNTP lave en hård justering ved hver synkroniseringshændelse.

Dette betyder at SNTP kan komme ud med nogle markante ændringer på tiden, mens NTP vil lave bløde justeringer. Fordelen med SNTP vil være ved store offset, hvor den bliver næsten nul med det samme, men det er ikke brugbart ved systemer med logging, da det kan give en "tidsmaskine" effekt, hvor tidsstemplerne går bagud.

2.6 BRUG AF KANALMÅLING METODE SAMMEN MED MULTIPEL TONE METODE

Denne metode er i skrivende stund ikke mere et koncept for en alternativ RF-metode for kanalmåling. Derfor, vil dette afsnit kun holde sig til hvad, som er relevant til en målemetode med mobile enheder og ikke konkret hardware og tilhørende opsætning/arkitektur.

Formålet med "Multipel tone metoden" er at optimere kanalmåling, sådan ud fra målingerne kan sætte kravene til design af RF-fronten, hardwarearkitekturen og software sådan det opfylder alle krav til QoS.

Impulsrespons over RF-kanalen er sammensætningen af signaler fra direkte vej og signaler, som reflekterer på overflader i miljøet. De reflekterende signaler skaber flere signalveje, hvilket resulterer i at signal styrken varierer afhængigt hvordan de forskellige signalveje krydser hinanden.

Ideen er at kunne switche mellem at sende to korreleret signaler lige efter hinanden, hurtig nok til at begge signaler er i luften samtidig. RX vil have en software kontroller som switcher mellem to LO til at fange hver deres signal.

"Multipel tone metoden" adskiller sig fra andre kanalmålinger, da den kan skelne mellem om RF forstærkningen kommer fra refleksioner eller ved shadowing fra enhederne bevæger sig i miljøet. Ydermere kan den skelne mellem ændringer i miljøet eller overgang mellem LOS til NLOS.

Denne metodes fokus er at anvende til scenarier med D2D kommunikation og 5G beamforming, hvor hovedspørgsmålet omhandler hvor hurtig skal softwaren kontrolleren være i antennen være forhold til opdagelse og handover.

Ved begge enheder i kanalmålingen er mobile, vil der opstå fading periodisk med deres relative bevægelse til hinanden og der vil være mange af overgange mellem LOS og NLOS, som sætter nogle krav til målemetodens robusthed og nøjagtighed.

Der vil gerne bruge for eksempel 5G frekvens på 28 GHz, vil der den hurtige fading vil ske hver halve bølgelængde, hvilket er cirka 5,4 mm ved 28 GHz. Med andre ord, der vil ske en dyb fading hver gang enhederne eller miljøet har ændret sig 5,4 mm.

Med en hastighed på 1 km/t vil dette ske cirka hver 19,44 ms og med en hastighed på 5 km/t vil det ske hver 3,9 ms.

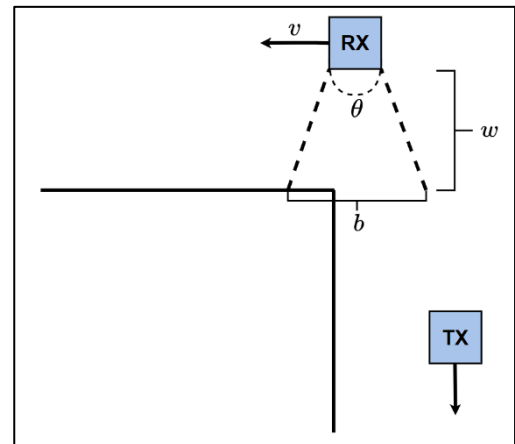
Ser man på scenariet med LOS->NLOS, vil det fleste tilfælde være at en af enhederne kører bag om et hjørne. Det kan man opstille matematisk med basal trigonometri, som illustreret i Figur 9.

.

"Multipel tone metoden" ser på hvor lang tid det tager at komme rundt om et hjørne og der ikke længere er LOS. Der gælder at en antenne modtager indenfor en givet vinkel spredning, θ hvilket er hardware bestemt.

Når enheden bevæger sig mod et hjørne med en hastighed, v følger den parallelt væggen, hvilket sker med en afstand w til væggen.

Tiden der går fra RX opfanger signaler fra væggen til NLOS er afstanden på antenne område ved væggen, b og over hastigheden.



Figur 9: Trigonometri aspekt af LOS->NLOS.

Tiden indenfor antennes opfattelse er:

$$T_b = \frac{b}{v} \quad (1)$$

Afstanden b kan udformes fra antennens spredning former en ligebenet trekant, hvor b er grundlinjen, hvilket giver en række af forskellige formler og den mest oplagte:

$$b = 2 * w * \tan\left(\frac{\theta}{2}\right) \quad (2)$$

Ud fra ligningen ses det at en lavere afstand w giver lavere T_b , samme effekt kommer med en højere hastighed eller lavere antenne vinkel. Worst-case, hvor den kører med 1 km/t og afstanden til væggen, $w = 0,5 \text{ m}$ med en antenne vinkel, $\theta = 10^\circ \rightarrow$

$$T_b = \frac{\tan(5)}{0,28} = 0,3125 \text{ s} = 312,5 \text{ ms}$$

Ud fra analysen om robotter kører rundt om hjørner, vil den ideelle hastighed på 1 km/t, hvor 5 km/t eller højere vil være uacceptabelt og hastigheden skal være en kendt variabel under hele målekampagnen.

De mobile enheders position skal være kendt til RF-målingerne, både relativt til hinanden, men også globalt i miljøet. En del af positionen skal orienteringen også være kendt og det skal være med en nøjagtighed på 10-20 cm baseret på rundt om hjørnet.

Tidssynkroniseringen skal koble positionssystemet sammen med RF-data, hvilket ikke må maksimum have en unøjagtighed på 50 ms.

3 PROBLEMFORMULERING

Til implementering af D2D teknologien i industrielle omgivelser, kræver det specielt designet antenner for at minimere de forskellige interferenser og de udfordringer der følger D2D kommunikation. Til at designe og implementere en antenne for D2D kommunikation, mangler der nogle vigtige kanal karakteristik (D2D handovers, allokering af DU'er og LOS til NLOS-transaktion), hvilket gøres ved at have en model over radioubredelsen over kanalen [4].

En model af radioubredelsen over kanalen kan også anvendes til at teste, simulere og typegodkende trådløse systemer, da modellen reflekterer de vigtige egenskaber ved kanalen.

En radioubredelsesmodel kan udformes på forskellige måder, blandt andet kanalmåling, hvilket giver en mere realistisk impulsrespons og god måde til at teste hardware i de rigtige omgivelser, da det vil være baseret på målte data.

Data fra kanalmåling kan blive brugt til at danne værdierne for parametrene i en stokastisk model, eller være sanitet check for kvaliteten i deterministiske modeller.

For at lave en rig model via kanalmåling, skal der laves målinger med forskellige egenskaber af kanalen; afstand mellem antennerne, retning af antennerne og omgivelserne. Normalt er kanalmåling en lang proces, hvor man rykker en eller begge antenner fra position til position, hvor ændring i position er nogle gange 1 meter, men mest over 5-10 meter.

Derfor, vil det være interessant at gøre både afsender og modtager mobile, sådan ændring af position bliver automatiseret. Andre fordele med mobile enheder, er at det bliver en emulering af D2D kommunikation, samt giver mulighed for at undersøge de små ændringer i position, måske ned i mmW.

Usikkerhederne fra målingerne skal ligge indenfor en acceptable værdi, hvilket rent videnskabelig vil være lig med nul. Fra en ingeniørs synsvinkel, giver det ikke mening at gøre modellen mere nøjagtig end til et vist punkt. Til at definere dette punkt for acceptabel værdi angående usikkerhederne, bliver det taget ud fra den ønsket måle case.

Hvis man tager udgangspunkt i Nokia Bell Labs case med en "multipel tone RF-model" til kanalmåling, så kan man opsætte nogle rammer til metoden, sådan kanalmålemetoden kan bruge til at lave en kanalmodel. Case omhandler meget om LOS->NLOS, hvilket kræver i det opstår flere gang i en målekampagne, i flere forskellige omstændigheder forhold til omgivelser og afstande.

Hvilke usikkerheder er der i en kanalmåling med mobile enheder, og hvor meget påvirker det brugbar af metoden til at lave en kanalmodel?

Case: Multiple tone RF-metode.

- *Hvordan kan sådan en kanalmetode opbygges?*
- *Hvordan sikres konsistente sæt af radio link data (inkl. Retning), position, tid og interpolation passer sammen?*
- *Hvad forårsager at måle afvigelser i de forskellige delsystemer?*
- *Hvilke parametre påvirker usikkerhederne?*
- *Hvor hurtig ændrer RF signal niveau i forbindelse ved LOS->NLOS?*

4 PROBLEMAFGRÆNSNING

Det overordnede mål er at sikre sammenhængende sæt af radiolinkmålinger, position og orientering, samt relativ tid (mellem målinger). Målingerne kan derved benyttes til analyse og karakterisering af radiolink med henblik på modellering og algoritme udvikling.

Dette projekt, som problemformuleringen lægger op til, fokuserer kun på at analysere metoden for kanalmåling robusthed, og ikke på anvendelsen af kanalmåling til kanalkarakterisering, men vil stadig være relevant at kører målekampagner til validere robustheden.

Projektet vil foregå i samarbejde med Nokia Bell Labs, hvor de vil fungere som interesseaktører og vejlede ud fra deres ekspertise. Kravene er baseret på brugen af metoden "Multipel tone RF kanalmåling", men selve softwarekontrollen og integrationen af antennen til målemetoden er udenfor dette projekts fokus.

Fysiske tests og eventuelt målekampagner vil foregå i Smart Produktion Lab på AAU [Fibigerstræde 14, 9220 Aalborg, Danmark] og lab i Nokia Bell Labs [Alfred Nobels Vej 27, NOVI 8, 9220 Aalborg, Danmark].

5 KANALMÅLING MED MOBILE ENHEDER

Målemetoden beskrevet i projektrapporten "5G Channel Sounding Between Mobile Robots in an Industrial Setting @3.57 GHz and 26.43 GHz" [1] lægger grundlag for dette projekt, hvor dette afsnit beskriver design og arkitekturen for målemetoden.

5.1 DE MOBILE ENHEDER

De to mobile enheder i systemet, er industrielle robotter MiR™ fra det danske firma med samme navn MiR© [7]. Der findes mange forskellige slags MiR robotter til forskellige opgaver, hvor dette system bruger MiR200 som er mest til at flytte noget fra A til B.

Til MiR robotter findes der moduler, som kan sættes ovenpå. Selve robotten sørger for transporten, mens topmodul udfører en opgave ved endepunkterne eller undervejs. Til brug i kanalmåling har de ikke modul på toppen, da i stedet for er alt RF-måleudstyret. En MiR200 kan ses i henholdsvis Figur 10 og Figur 11.



Figur 10: MiR200 fra fronten og en smule skråt. [7]



Figur 11: MiR200 fra bag og en smule skråt. [7]

Der skal et samarbejde mellem elementer og systemer til at få en MiR robot til at virke og udfører dens opgave. De mest relevante hardware er dens ultralyd sensorer i front og bag, samt LiDAR til at måle dens opgivelser, som bruges til SLAM algoritme og rute planlægning.

Robotten har et drejelig svingbart gaffelhjul i hver af dens hjørner, som kun giver stabilitet og har ikke motorkraft, hvilket ligger i de to hjul i midten af langsiden. De to hjul er fikset og kan ikke dreje, derfor har de hver deres motor som regulerer forhold til hinanden når den skal dreje eller kører lige ud.

På toppen af robotten er der nogle interface, som er dmx port til nødstop knappen og et CAT port til dens interne router, som bruges til at forbinde med maskinen og udtrække data.

5.2 DATA FRA MiR – ROS

MiR bruger frameworket Robot Operating System (ROS) [8] til at styre ud fra input fra sensorerne og MiR software. I ROS opbygges data indenfor forskellige dataframes lige fra sensor til diagnostik [9].

Til at udtrække de data fra ROS, laves der en 'subscriber' enhed, hvilket er skrevet i Python kode og bliver kørt på en Raspberry Pi 4™. Raspberry Pi enheden er forbundet til CAT porten i toppen af MiR robotten og kører med fuld funktionel Ubuntu, da ROS framework kræver det for at kunne virke.

Raspberry Pi enheden subscriber til to slags positionsdata, "geometry_msgs" og "nav_msgs", hvilket er henholdsvis den global position (G) og relative position (R). De data som hører med henholdsvis G og R kan ses i Tabel 1.

	Pose			Quaternion				Linear			Angular		
R	x	y	z	x	y	z	w	x	y	z	x	y	z
G	x	y	z	x	y	z	w						

Tabel 1: Positionsdata fra ROS, R: relative position data og G: globale position data.

Forskellen på de relative og de globale positionsdata er niveauet i informationstræet, mellem hardware og software interface. De relative er tættere på hardware niveau end de globale, som er tæt på software interface. For hvert niveau mellem hardware og software interface, bliver der lagt header på data som resulterer i at de bliver opdateret med forskellige frekvenser. De relative data bliver opdateret med cirka 41 Hz, mens de globale bliver opdateret med cirka 10 Hz.

Det højeste niveau er softwareinterface, hvilket består af det globale kort og MiR Fleet'en.

5.3 MiR FLEET – RUTEPLANLÆGNING

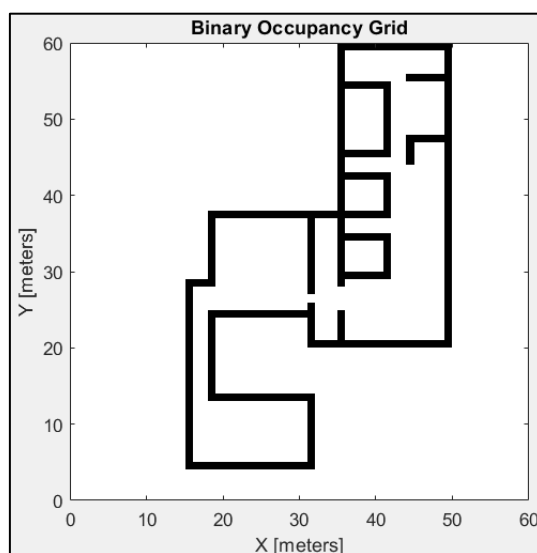
Hver MiR robot har dens egen webbaseret interface, som kører fra dens interne netværk. Via interface kan der ændre på robotens indstillinger såsom hastigheder og hvilket kort som den kører i. Kortet defineres i det webbaseret interface, hvor der kan blive defineret waypoints og forskellige slags zoner. Systemer med flere MiR robotter, bliver det meget relevant at integrere en speciel enhed, MiR Fleet™. Figur 12 viser hvordan det webbaseret interface af MiR Fleet kan se ud på forskellige enheder.



Figur 12: Billedet af MiR Fleet på forskellige enheder. [7]

MiR Fleet centraliserer hele operationen for gruppen af robotter, som kan få adgang fra via dens egen webbaseret interface.

Kortet over testområdet er tegnet i Figur 13, som består af to haller. Den store hal kaldes Hal A, som er til højre mellem 35-50 i x-aksen og mellem 20-60 i y-aksen, mens den anden del kaldes Hal B. Mellem Hal A og Hal B er en gang, hvor både RX og TX begynder målekampagnen.

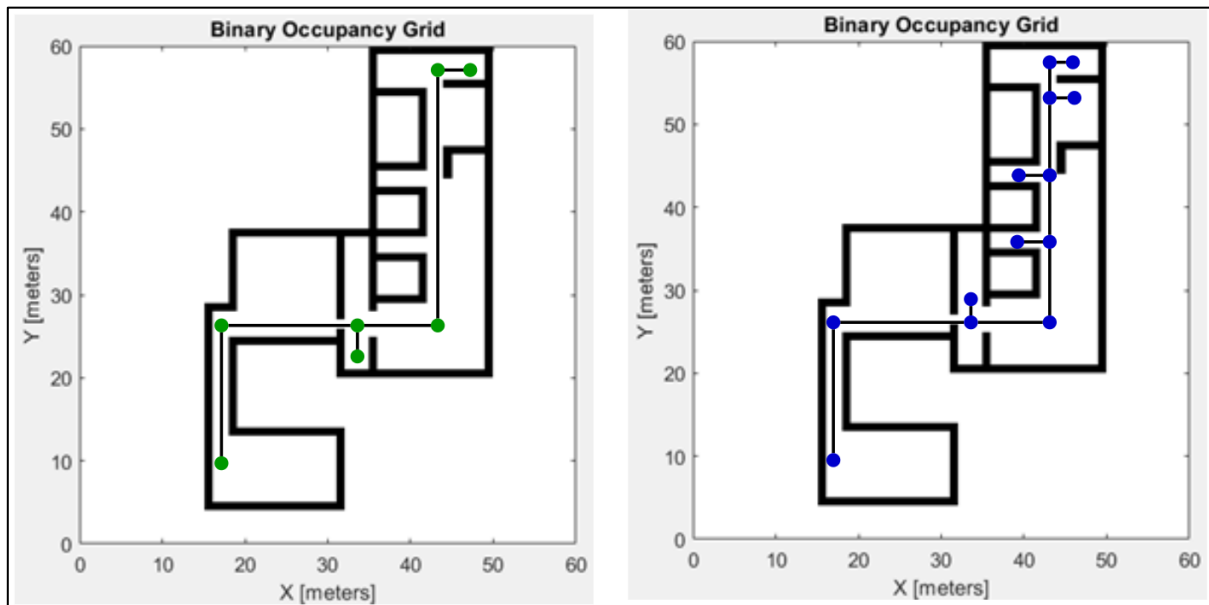


Figur 13: Det aktive kort over Smart Production Lab.

I dette projekt har ruteplanlægningen for TX og RX mål, at opnå målinger med forskellige afstande, mange krydsninger og mange steder af LOS->NLOS ind i ruteplanlægningen. Til at opfylde disse mål, er der lavet en ruteplanlægning ud fra en meget simpelt koncept.

Konceptet tager udgangspunkt i de to haller, sådan at TX kører nogle omgange i Hal A, nogle omgange i Hal B og nogle omgange mellem Hal A og Hal B. Det vil sige ruten for TX er trefoldig, hvor RX har den samme trefoldig, men kortere omgange sådan RX kører alle tre for hver TX-del og gentager sådan den sker for alle TX-dele.

TX skal kun køre på den store gang i hallerne, mens RX kører ind på alle sidegrene, som er tilgængelige for robotten. Figur 14 viser hvor både TX og RX kommer rundt i hallerne i kortet fra Figur 13.



Figur 14: Rute oversigt; venstre: waypoints over TX-rute, højre: waypoints over RX-rute.

Robotterne optimerer løbende dens rute til næste givet punkt, hvor hvis den rammer forhindringer (den anden robot) på sin vej, laver den beregninger indtil den finder en vej. Robotten har en øvre grænse af forsøg til at finde en alternativ, hvilket vil sætte den i stop hvis den kommer over den grænse.

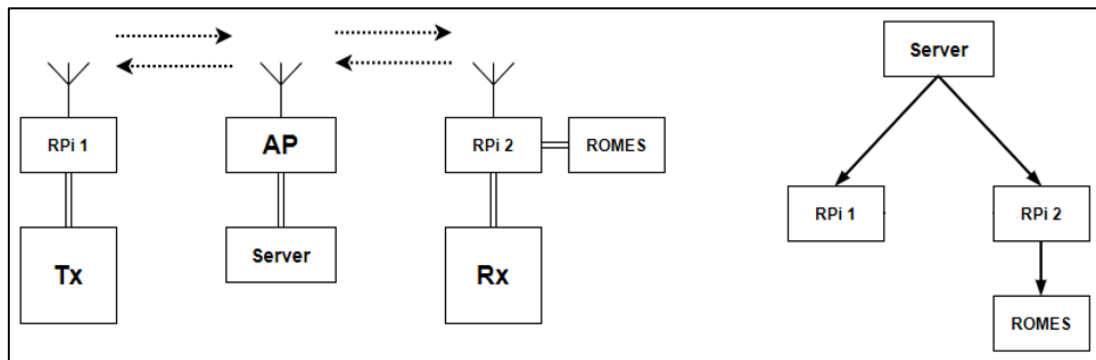
Dette miljø er generelt ikke optimalt for flere MiR opererende samtidig, da gangene er smalle, der er ingen løkker, kun en lang hovedgang som går gennem begge haller og nogle få sidegange, som ikke går langt ind. Det giver en del problemer, når to robotter mødes og skal krydses til at fortsætte modsat vej, som dermed skaber Deadlocks.

Løsningen er blevet, at styrer RX via dens webbaseret interface med den manuele kontroller, sådan at når deres veje krydses, medfører det løsning på deadlock før de går stop mode.

5.4 NTP

En MiR robot er et lukket og uafhængigt system til hinanden, når der snakkes om deres interne ure, hvilket ikke er funktionelt ved kanalmåling, hvor en synkroniseret tid er hvad forbinder positionering systemet med RF-systemet.

I denne opsætning kan Raspberry Pi enhederne udnyttes til også at sørge for tidssynkronisering, da de kan have NTP daemon kørende i baggrunden. Figur 15 illustrer hvordan topologien og stratum opbygningen er i systemet.



Figur 15: NTP-opsætning; Venstre: topologien hvor dobbelt streg er kabel forbindelse og de stiplede pile er trådløs forbindelse. Højre: Hierarkiet af NTP-stratum i systemet. [1]

Opsætningen har serveren forbundet kabel til routeren og NUC'en med ROMES er kabel forbundet til den Raspberry Pi på RX-systemet, dermed har Raspberry Pi'ene har kun et enkelt trådløs hop til serveren. Til at optimere den trådløs forbindelse, anvendes der en WiFi6 router [10].

Serveren er en Raspberry Pi, som er konfigureret til at synkronisere til de danske ntp pools, 0.dk.pool.ntp.org, sådan den ikke drifter væk og beholder den rigtige dato og tidszone. Serveren er konfigureret til kun at lave IP'erne fra klienterne til at synkronisere op imod den, hvor klienterne er konfigureret til burst synkronisering hvilket er 8 synkroniseringspakker pr. poll, som sker med det laveste mulige pollinterval på 16 sekunder.

Denne opsætning har en lineær drift efter NTP-forbindelsen er væk.

$$\text{drift} \quad \theta_t [ms] = 0,0001 * \frac{t [ms]}{250} + \theta_0 \quad (3)$$

Hvor θ_0 er udgangspunktets offset og θ_t er offsetten efter der er gået t millisekunder. Ud fra denne forskrift kan der ses, at θ_0 er den dominerende faktor og hældningen svarer til driften på 100 PPM.

6 ANALYSE AF USIKKERHEDERNE

Selve platformen for kanalmålemetoden defineret ud fra systemets arkitektur, topologi og design. Platformen består hvilke data bliver udtrækket fra MiR robotterne og hvordan, hvilken rute de kører til en givet hastighed. NTP-delen testet for optimal ydeevne, samt monitoreret. RF-måleudstyret ses som en tilføjelse til platformen, hvor det giver nogle ekstra klienter til NTP, men bliver forbundet til den lokale Raspberry Pi, som fungerer som server for klienter indenfor robotten.

Dette afsnit vil gennemgå ruten af data for de to delsystemer, hvor opstår der usikkerheder, derefter se hvor meget variation kan der forekomme i de enkelte delsystemer, men også sættes sammen til kanalmålemetoden.

6.1 PROBLEMSTILLING MED TIDSSYNKRONISERING FORHOLD TIL NTP

Forhold til positionssystemet i ROS, er NTP meget grundig dokumenteret i en RFC under RFC5905 [11] for den seneste version. Et system som bruger NTP til synkronisering bliver delt op i en hierarkisk forbindelse, hvor en enhed på et givet niveau synkroniserer til en enhed eller enheder på niveauet over den.

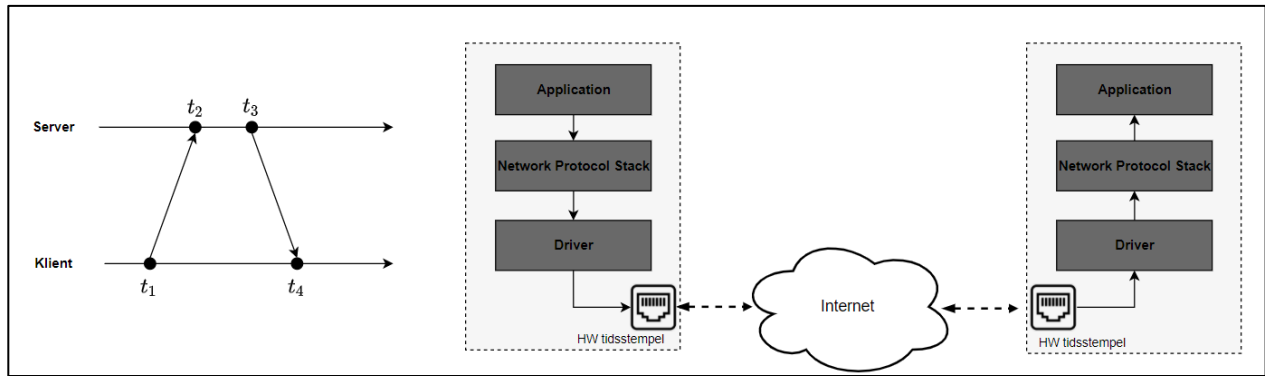
I et NTP-hierarki hedder et niveau stratum, hvor stratum 0 er det øverste stratum som består meste af et atomur, som er hovedkilden og er referencen for synkroniseringen. Efter enhederne synkroniserer til enheden i stratum 0 med antal step lig med deres stratum nummer, så vil der hvert stratum vil offsetten øges, hvor ved stratum 16 er kæden for langt og dermed blevet definitionen af ikke muligt for synkronisering.

Lad enhed A være på stratum n, som synkroniserer til en enhed på stratum n-1, hvor de bliver kaldt henholdsvis klient og server. Ved enhed A bliver tiden justeret med et interval på 1 sekund, som er baseret på udregnet offset. Denne event kaldes **synkroniseringsevent**. Baseret på synkroniseringsevent er diskret periodisk og der er en kontinuert drift, så vil offsetten være en savtakket funktion.

Offsetten bliver udregnet fra fire tidsstempler, to fra klientens tid og to fra servers tid. Klienten får de fire tidsstempler ud fra UDP request-reply kommunikation til serveren. Denne request-reply kaldes en **poll event**, som sker med et defineret interval.

Et pollevent er en UDP-pakke fra klienten med tidsstempel for afsendelsen, serveren modtager pakken, hvor tidsstemplet for modtagelsen bliver tilføjet og samt tidsstempel ved afsendelsen tilbage, hvor klienten også taget tidsstemplet for modtagelsen.

Figur 16 viser disse fire tidsstempler og hvorhenne i processen tidsstemplerne bliver tilføjeret.



Figur 16: Poll event mellem klient og server, hvor de fire tidsstempler er defineret. Til venstre er hvor tidsstemplerne i hardwaret bliver angivet [12].

Intervallet for poll event kan sætte fra 16 sekunder til 64 timer, som bestemmer hvor meget der skal justeres ved synkroniseringsevent, hvilket betyder om det er en hård eller blød justering.

Ud fra de værdier bliver der udregnet round-trip, som er ligning (4), samt offset i ligningen (5). Klienten modtager også statistiske data omkring sin server til stratum 0.

**Round-trip
delay**

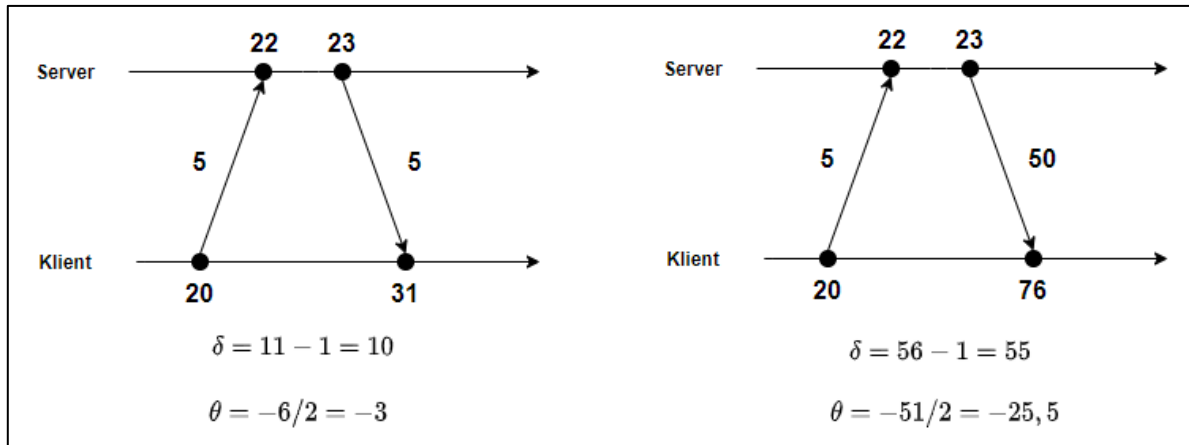
$$\delta = (t_4 - t_1) - (t_3 - t_2) \quad (4)$$

Offset

$$\begin{aligned} t_2 &= t_1 + \theta + \frac{\delta}{2}, \quad t_3 = t_4 + \theta - \frac{\delta}{2} \\ \Leftrightarrow \frac{\delta}{2} &= t_2 - t_1 - \theta \rightarrow t_3 = t_4 + \theta - (t_2 - t_1 - \theta) \\ \Leftrightarrow t_3 - t_4 + t_2 - t_1 &= 2\theta \\ \theta &= \frac{(t_2 - t_1) + (t_3 - t_4)}{2} \end{aligned} \quad (5)$$

Disse er hvad NTP bygger dens tidsjustering på, men det er ikke bare at anvende den udregnet offset, da der er flere elementer, blandt andet fra tidsstemplerne kommer fra en kommunikation mellem klient og server. Forsinkelsen OTA giver meget af usikkerheden i nøjagtigheden af synkroniseringen, især med asymmetrisk da der er antagelse om symmetri i kommunikationen.

Figur 17 viser et eksempel på forskellen mellem der er symmetri og asymmetri.



Figur 17: Illustration af forskellen mellem symmetri og asymmetri.

Eksemplet viser en stor forskel mellem symmetrisk og asymmetrisk kommunikation, samt hvor stor påvirkning det kan have. Alene vil asymmetrisk ikke være et problem, men sammen med trådløs kommunikation, som betyder stokastiske forsinkelser.

Den sidste påvirkning er den omtalte drift i den lokale ur hos klienten, men også ved serveren. NTP-protokollen har derfor en del mere udregninger end basal kigge på udregnet offset.

I synkroniseringspakken fra server på stratum n-1, er der en masse ekstra statistisk data som omhandler dens synkronisering til enhed op stratum n-2 og videre til stratum 0. En klient kan også lave poll event med flere enheder på stratum n-1 og måske også stratum n. Begge aspekter bliver dækket i protokollen, hvor det samlede statistik bliver udregnet og sendt videre, samt filetere gennem de peers, som er blevet lavet poll på, hvor de bliver vurderet for at finde de bedste forhold til forrige jitter og gennemsnit.

Aspektet med en total synkroniserings statistisk og filtreringen af flere peers bliver ikke gennemgået. Dette projekt består af en server, derfor fokuserer analysen på at se hvordan protokollen udregner offset, men også den forventet offset over tid baseret på de nævnte usikkerheder.

6.2 ANALYSE AF PROTOKOLLEN I NTP

Følgende afsnit bygger på dokumentationen over NTP [13] og RFC5905 [11]. Hele protokollen dækker over en masse scenarier og finder den bedste antagelse af offset baseret på en masse data.

6.2.1 Peer procedure

Denne procedure dækker over at modtage reply fra serveren, hvor data bliver filteret og de forskellige statistiske variabler bliver udregnet. Det er offset, round-trip og spredningen:

$$\varepsilon = \rho + \rho + \phi(t_4 - t_1) \quad (6)$$

Næste procedure er en filetering af disse værdier, hvor der bliver set på de sidste otte reply data, hvor der vælges den offset med den korteste tilhørende round-trip delay.

6.2.2 System procedure

Hvis den valgte offset er tre gange højere end den tidligere peer jitter, bliver den smidt væk og vil jitteren blive opdateret, da det vil være en unaturlig høj peak i offset hvis dette sker.

$$x_i = \theta_i, \quad y = \delta_i, \quad z = \varepsilon_i$$

Dette bliver gemt i en matrice hvor det er (x, y, z, w) , som bliver sorteret forhold til y i stigende rækkefølge. hvorved;

$$\theta = x_0, \quad \delta = y_0, \quad \varepsilon = \sum_i \frac{z_i}{2^{i+1}}, \quad \varphi = \sqrt{\frac{1}{n} \sum_i (x_0 - x_i)^2}$$

Herefter sker en masse filetering af data fra multiple server ("Peer"), men for simplificering bliver der antaget kun er en peer, som er valid og godkendt.

Der sker en kombination af værdierne fra de godkendte peers, hvor der bruger midlertidige variabler: $y = z = w = 0$.

$$\begin{aligned} x &= \frac{\delta}{2} + \varepsilon + \phi\mu + \vartheta \\ y &= \frac{1}{x}, \quad z = \frac{\theta_i}{x}, \quad w = (\theta_i - \theta_0)^2 \\ \Theta &= \frac{z}{y}, \quad \vartheta = \sqrt{\frac{w}{y}} \end{aligned}$$

Dette er hvad udregninger af forventet fejl og maksimale fejl, hvor den eneste nye variable er opdateringsintervallet, μ . Der er nogle af udtrykkene, som har notationen i'te element, hvilket er ikke implicit i protokollen hvad der hentydes til.

6.3 SIMULERINGSTEST AF NTP VÆRDIERNE

Først antages at læsningsfejlen ved klienten og server er så lille eller de går ud med hinanden, sådan at spredningen i ligning (6) bliver

$$\varepsilon = \phi(t_4 - t_1)$$

Konfigurationen af simuleringen, omfatter en pollintervallet på 16 sekunder samt poll event sker med en burst, hvilket betyder, at for hver poll event sendes der otte synkroniseringspakker med et interval på 2 sekunder.

Dette eksempel har kun en klient, som har en offset= 17 ms til serveren og en drift på 100 PPM. Simulere symmetriske round-trip delay til alle af synkroniseringspakkerne, som er tager en tilfældig værdi mellem 10 og 30 med enheden i ms.

Gennemgang af undersøgelse for uvisheder:

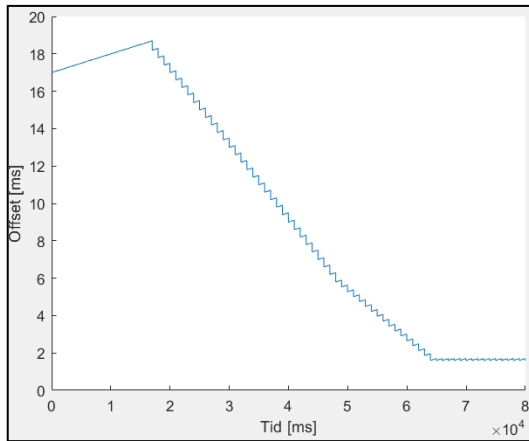
1. indikerer notationen 'i' ved ε den i'te element af de gemte værdier.
2. ved ligningen for ϕ , er n antal elementer og det i'te element af de gemte værdier.
3. opdateringsintervalvariablen, μ er det for poll event på 16000 m's eller synkroniseringsevent på 1000 ms.
4. indikerer notationen 'i' ved z og w det sidste element af de gemte værdier.

Værdierne undersøges ved simulering, hvor der ændres en ting ad gangen eller få tilføjeret til, se nedenstående oversigt.

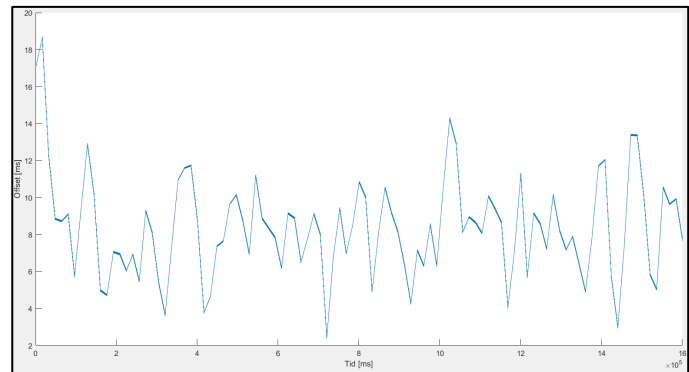
	oversigt
1. forsøg	Her sættes $\phi = 15$ PPM og $\mu = 1000$ og resten af punkterne, antages det som de er formuleret.
2. forsøg	Samme opsætning, men der vil køres to omgange poll events, hvilket svarer til cirka 32000 ms. Nu skal der se hvor meget den kombineret jitter påvirker udregningerne.
3.forsøg	Nu aktiverer synkroniseringsevents, hvor det bliver baseret på θ_0 og θ , med samme opsætning, men nu kører den det antal poll event, som det kræver til at komme i steady-state.
4. forsøg	Ændrer til at netværket kan være asymmetrisk, men begge veje er indenfor 10-30 ms

Uden drift bliver synkroniseringen ikke en savtakket funktion, men i stedet er det en funktion.

Under 3. forsøg er der en konstant symmetri i netværk kommunikationen, hvilket ender synkroniseringen i steady-state som vist i Figur 18. I 4. forsøg, som har et asymmetrisk netværk, bliver adfærden af synkroniseringen mere kaotisk som illustreres i Figur 19.



Figur 18: Synkronisering ved symmetrisk netværk kommunikation.



Figur 19: Synkronisering ved asymmetrisk netværk kommunikation.

7 DESIGN AF FULDIG AUTOMATISERET SYSTEM

Kanalmålemetoden indeholder en masse delsystemer, som alle skal sættes i gang før selve kampagnen sker. Det blev valgt at lave ét User Interface (UI) på disse processere, som samler de forskellige delsystemer og gør det mere automatiseret, sådan det er overskueligt at sætte en målekampagne i gang.

Nogle processere og delsystemer som er mere fysiske anlagt kan ikke integreres indtil et UI, såsom processen at klargøre RF måleudstyret, fra at koble til batteriet, lad dem varme op og tænde systemet. Der er også forberedelse af missionerne, som kræver at definere ruten ud fra prædefineret waypoints.

Når man ser bort fra de systemer som kræver fysisk tilgang, vil der stadig være en del systemet tilbage. Der kræves tre terminaler til at SSH forbinde til Raspberry Pi'ene (RX, TX og server), hvor der skal tjekkes om NTP er kørende på dem alle, samt der skal starte scriptet på RX og TX til at hente positioner. Raspberry Pi ved RX skal der også tjekkes om interface Eth1 har de rigtige IP-konfigurationer forhold til at kunne forbinde direkte med NUC'en, sådan den kan tidssynkronisere.

Der skal åbnes en browser fane til webbaseret interface af både RX og TX, hvor der skal tjekkes om de virker kalibreret forhold til kortet, de har den ønsket maksimum hastighed og der skal vælges mission til dem, som igangsættes når alt er klar.

Der skal forbindes til NUC'en via Remote Desktop, hvor der skal tjekkes om den kan synkronisere til den lokale Raspberry Pi og der skal startes ROMES op, hvor der skal tjekkes at den opfanger de rigtige signaler.

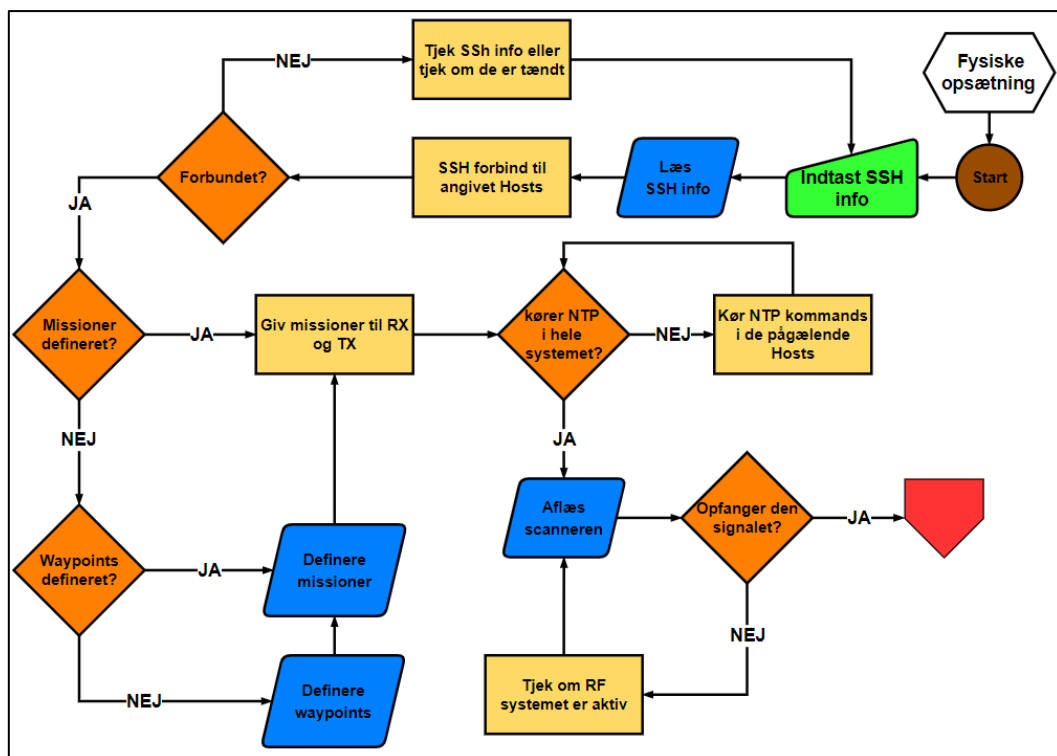
Plus der skal et nyt script til at lave monitorering af synkronisering under målekampagnen, sådan det giver konkret oversigt af synkroniserings usikkerhed til den givet målekampagne. Der vil dermed efter en målekampagne, være måledata på fire

forskellige maskiner; to Raspberry Pi har positionsdata, NUC'en har RF målinger og den centrale computer har data for synkroniseringen.

7.1 BACK-END - SCRIPTS

Til en hver UI er der en back-end, alt den koden relateret til udførelsen af selve opgaven med en UI. UI deles op i to dele. Første del er opsætningen og tjek af systemer før selve målekampagnen kan begynde.

Figur 20 er et flowdiagram over den første del, hvor del 2 begynder ved den røde form. Tager udgangspunkt fra en computer, som skal virke som en kontroller central med UI.



Figur 20: Flowdiagram over del 1 i automatiseret system.

I den første del er der fire vigtige aspekter i et back-end sammenhæng; der er SSH forbindelse, ruteplanlægning, NTP og RF-systemet.

Forbinde med SSH sker til de tre Raspberry Pi i systemet; TX, RX og NTP-serveren, hvor det er den basale terminalkommando, men for at gøre det nemmere kan der genereres en SSH-nøgle til de tre enheder.

Når den centrale computer har fået lavet SSH nøglerne, skal der ikke længere angives password i forbindelsen. I Appendix 13.7 står der terminal kommandoerne til at lave en SSH-nøgle mellem host A til host B.

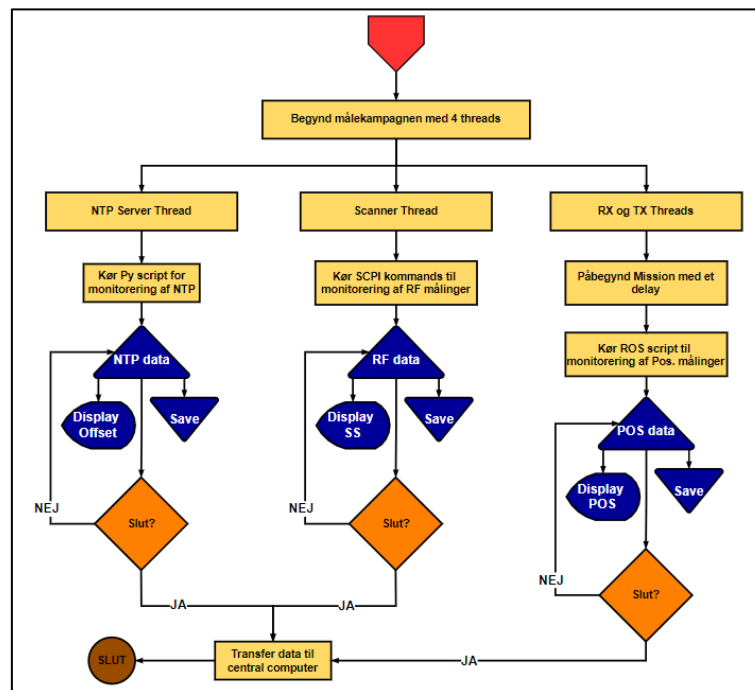
Ruteplanlægningen sker via MiR's webbaseret interface eller gennem MiR Fleet, men alt via de interface bruger ROS til at kommunikere til robotterne og gør det til kommandoer.

Derfor, kan man lave sit eget interface til ruteplanlægning, da ROS er et open source-framework. Der findes endda ROS API til de fleste programmeringssprog.

NTP-delen har nogle få terminale kommandoer, som er relevante i den her sammenhæng. Der er en kommando til at tjekke om NTP servicen kører, som kan også se om det er den rigtig IP og rigtige tidszone. I forbindelse med SSH, vil programmet kende de forskellige IP'er, hvor der skal tjekkes om det er dem i de forskellige NTP konfigurationsfiler. Ekstra skal der bruges terminale kommandoer til at stoppe NTP service, når der skal rettes i konfigurationsfilen og terminalkommandoen til at genstarte en service efter rettelsen.

Moderne RF-måleudstyr, såsom Rohde & Schwartz, er det muligt at fjernstyre og monitorere via SCPI kommandoer.

Til den anden del, selve udførelsen af målekampagnen, hvor UI viser en anden fordel med at kunne illustrere alle delsystemerne i samme interface. Figur 21 er flowdiagrammet for del 2.



Figur 21: Flowdiagram over del 2 i automatiseret system.

Der bliver nødt til at blive lavet fire threads, da de dækker om en SSH-forbindelse eller en SCPI-forbindelse. Scanner threads bruger SCPI-kommandoer, som beskrevet i del 1, hvor tilføjeret display i UI og save data. RX og TX threads kører ROS script, hvor de "subscriber" til positionsdata. NTP-serveren kører et monitoreringsscript over NTP på dens klienter og display offsetten mellem klienterne.

Efter en målekampagne er afsluttet eller stoppet, bliver alle de forskellige data overført til den centrale computer, hvor UI kører på.

8 DATA ANALYSE

Efter en målekampagne er der otte filer, som indeholder en masse rå data. Dette afsnit afspejler hvordan data flowet og metoder, som der er blevet brugt i dette projekt i forhold at resultaterne giver mening. Der er fokus på metoder til at afslører usikkerheder i systemet, tidssynkroniseringen og positioneringen. Der vil også være et tiltag på den givet case med henblik en indsigt i LOS->NLOS ud fra en geometrisk tilgang end den normale RF-tilgang.

8.1 STRUKTUR OVER MAPPER OG FILER

Alt data processeringen forgår i MATLAB™, hvor til en struktur er designet til at opnå nemmere tilgang til data og muligheden til at tilføje procedure midt vejs eller i slut.

Strukturen i MATLAB kan ses i Figur 22**Fejl!**

Henvisningskilde ikke fundet., hvor der er nogle mapper og MATLAB funktioner.

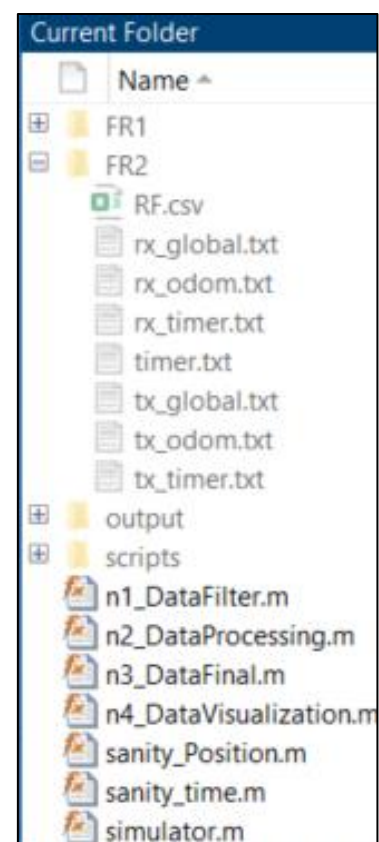
Mapperne "FR1" og "FR2" er mapperne fra to målekampagner henholdsvis med Frekvens range 1 og Frekvens range 2, som indeholder de otte filer fra en målekampagne. Mappen "FR2" er åben i Figur 22, hvor de nævnte filer skal have disse navne for at MATLAB funktionerne fungere.

Filen "RF.csv" indeholder alt data fra scanneren, fire af filerne er for positionerne fra RX og TX, hvor der er global og odom (relative), og de resterende er over monitoreringen af NTP.

Mappen "Output" er hvor alt data processeringen ender, efter de forskellige MATLAB funktioner er blevet kørt, og mappen "scripts" indeholder de scripts som er beskrevet i **7.1**.

De MATLAB funktionerne, der har "nX_" i prefix af deres navne, er en opdeling af data processeringen. Opdeling gør det muligt at tage udgangspunkt i et output ved et givet punkt i processen. Designet har til fordel under arbejdet af funktionerne, at når et step er færdigt udviklet, skal det ikke køres igennem ved test af det næste step.

Ved udførelsen af "n3_DataFinal.m", står der tilbage positionsdata for alle tidspunkterne med RF-målinger, henholdsvis i en RX-fil og TX-fil.



Figur 22: Skærbillede af strukturen i MATLAB.

8.2 RÅ DATA TIL RF MÅLING MED POSITIONSDATA

De første tre funktioner er `n1_DataFilter.m`, `n2_DataProcessing.m` og `n3_DataFinal`, hvor det munder ud i hele tre steps af data processering før de er klar til analyse;

- step 1.** Filtrering og justering af de rå data
 - Output = `RX.mat` og `TX.mat`
- step 2.** Udregn hastigheder til de globale positioner
 - Output = `RX_process.mat` og `TX_process.mat`
- step 3.** Interpolation af RF-målinger mellem positionsdata
 - Output = `RX_RFdata.mat` og `TX_RFdata.mat`

De forskellige funktioner tager navnet på målekampagnens mappe, for eksempel "FR2" som input, hvor første step også tager hvilken frekvens range og den 1 meter reference måling af signalstyrken.

Der bliver loadet `tx_global.txt`, `rx_global.txt`, `tx_odom.txt`, `rx_global.txt` og `RF.csv`.

Justering af RF-data

Blandt de rå RF-målinger bliver kun tidsstempel og `RePower` udtrykket. Afhængig af frekvens range, sendes der henholdsvis to eller fire signaler ad gangen, hvor gennemsnittet af `RePower` værdierne tages. Yderligere bliver pathloss udregnet, baseret på den 1 meter referencemålingen. Tilbage er tidsstempel, gennemsnit `RePower` pr. burst og den tilhørende pathloss.

1. Justering af tid

Tidsstemplerne bliver lavet om til millisekunder, sådan det er nemmere og hurtigere at arbejde med. Eftersom de rå tidsstempler, er i format "HH:mm:ss.SSSS", bliver det nogle store værdier i millisekunder, derfor findes den laveste tidværdi i alle filerne, som bliver reference tid og alle de andre målinger er tid efter det.

Ændring af rotations repræsentation

Laver en afbildning af orienteringen fra Quaternion til Euler notation.

Sammensætter data til en fælles matrix

Der bliver lavet `RX.mat` og `TX.mat`, som indeholder deres globale målinger og relative målinger, samt begge får en kopi af RF-målingerne. Disse matricer bliver sorteret til at være i kronologisk rækkefølge. Laver en ekstra kolonne med navnet "global", denne kan indeholde 0,1 og 'inf', som repræsenterer respektive relative, globale og RF-data.

1	2	3	4	5	6	7	8	9	10	11
time	x	y	z_euler	y_euler	x_euler	linear	angular	global	RePower	PathLoss

Figur 23: Data strukturen i `RX.mat`.

2. Justering af tid

Opstart processen er meget lang, hvilket giver en masse RF-målinger før positionssystemet kører og når positionssystemet kører, vil der også være noget tid før en ad robotterne begynder på deres mission. Derfor fundet tidspunktet hvor den første robot stopper med at stå stille, alt data før det bliver filteret ud. Der tages også hensyn, sådan det er en global positionsdata, som er det første data i begge og den ene er den sidste tidspunkt hvor den står stille og den anden har den første globale position efter det fundet tidspunkt. Det modsatte sker i enden af matricerne.

Nu findes den laveste tidsstempel i begge filer, som bliver det nye reference tid og alle tidspunkter er ud fra dette.

Interpolation mellem relative positionsdata

Alle relative positionsdata har fremdriftshastighed og rotationshastighed, hvilket ikke er givet ved de globale positionsdata eller RF-data. Der bliver gået gennem alle indgange i den samlet matrix, hvor mellem to efterfølgende relative data, bliver der lavet interpolation mellem dem til alle indgange mellem dem (global og RF) for fremdriftshastighed og rotationshastighed. Ved interpolationen antages der en konstant acceleration mellem de relative positionsdata. Dette kan udtrykkes matematisk:

Lad TX.mat eller RX.mat $\rightarrow \bar{A}$, hvor

$$\bar{A} = \{\bar{d}_1, \bar{d}_2, \dots, \bar{d}_n\}, \bar{d}_x \in \{\bar{R}, \bar{G}, \bar{M}\}, x = 1, \dots, n, \text{ hvor } \bar{d}_1 < \bar{d}_2 < \dots < \bar{d}_j \text{ i tidstempel}$$

$$\bar{R} = \{\bar{r}_1, \bar{r}_2, \dots, \bar{r}_j\} = \text{relative data, hvor } \bar{r}_1 < \bar{r}_2 < \dots < \bar{r}_j \text{ i tidstempel}$$

$$\bar{G} = \{\bar{g}_1, \bar{g}_2, \dots, \bar{g}_k\} = \text{globale data, hvor } \bar{g}_1 < \bar{g}_2 < \dots < \bar{g}_j \text{ i tidstempel}$$

$$\bar{M} = \{\bar{m}_1, \bar{m}_2, \dots, \bar{m}_l\} = \text{RF data, hvor } \bar{m}_1 < \bar{m}_2 < \dots < \bar{m}_j \text{ i tidstempel}$$

$$\rightarrow |\bar{A}| = n = j + k + l$$

Mellem alle relative data i $\bar{A}$, skal der se på de data som er imellem forhold den kronologiske rækkefølge.

$$[\bar{d}_i, \bar{d}_t] \in \bar{A} \equiv [\bar{r}_{j'}, \bar{r}_{j'+1}] \in \bar{R} \text{ for } \forall j' = 0, 1, \dots, j-1,$$

$[\bar{d}_i, \bar{d}_t]$ vil være relative data, som vil være lige efter hinanden i $\bar{R}$, men i $\bar{A}$ vil der måske være data imellem, $\bar{p} \in \{\bar{G}, \bar{M}\} \rightarrow [\bar{d}_i, \bar{p}, \bar{d}_t]$ Til at udregne fremdriftshastigheden og rotationshastigheden til $\bar{p}$:

$$T = \bar{d}_t.time - \bar{d}_i.time, \quad T' = \bar{p}.time - \bar{d}_i.time$$

$$acc_v = \frac{\bar{d}_t.linear - \bar{d}_i.linear}{T}, \quad acc_{rot} = \frac{\bar{d}_t.angular - \bar{d}_i.angular}{T}$$

$$\bar{p}.linear = \bar{d}_i.linear + T' * acc_v, \quad \bar{p}.angular = \bar{d}_i.angular + T' * acc_{rot}$$

Interpolation mellem de globale positionsdata

Nu har de relative positionsdata tjent sit formål, derfor fjernes de fra matricerne RX og TX. Tilbage er globale positionsdata og RF-data, hvor RF har fået angivet fremdriftshastigheder og rotationshastigheder, men mangler positionerne.

Her vil der være en del RF-data mellem to globale positionsdata når det ses i kronologisk rækkefølge, hvor der skal laves interpolation mellem de globale positioner til at finde positionerne til tiden hvor der er RF data.

Stedet for at bruge samme fremgang i forrige interpolation, bliver der her brug af forskydningsformlen [14]:

$$\bar{s} = ut + \frac{1}{2}at^2 \quad (7)$$

Der udarbejdes interpolation mellem position 1 ($\bar{P}_1$) og position 2 ($\bar{P}_2$), hvor punkt 3 ($\bar{P}_3$) er i mellem tiden og den positionen der skal findes ($\bar{P}_3$).

Variablen u i ligning (7) er hastigheden ved ($\bar{P}_1$), t er tiden mellem ($\bar{P}_1$) og ($\bar{P}_3$) og a er accelerationen mellem ($\bar{P}_1$) og ($\bar{P}_2$), altså ændringen af hastighed ratioændringen af tid.

Dermed positionen

$$(P_3)_x = (P_1)_x + \frac{s_x}{D} * ((P_2)_x - (P_1)_x), \quad (P_3)_y = (P_1)_y + \frac{s_y}{D} * ((P_2)_y - (P_1)_y)$$

Hvor $D = \text{afstand mellem } (\bar{P}_1) \text{ og } (\bar{P}_3)$.

Data bliver gemt til henholdsvis RX_process.mat og TX_process.mat.

1	2	3	4	5	6	7	8	9	10	11
time	x	y	z_euler	y_euler	x_euler	linear	angular	global	RePower	PathLoss

Figur 24: Data strukturen i RX_process.mat.

Padding af data og filetering

En data padding i begyndelsen og slutningen udarbejdes som det sidste. De samme RF data i forhold til tidsstempler og globale positionsdata bliver fjernet. RX_RFdata.mat og TX_RFdata.mat, som kan ses i Figur 25.

1	2	3	4	5	6	7	8	9	10
time	x	y	z_euler	y_euler	x_euler	linear	angular	RePower	PathLoss

Figur 25: Data strukturen i RX_RFdata.mat.

9 RESULTATER

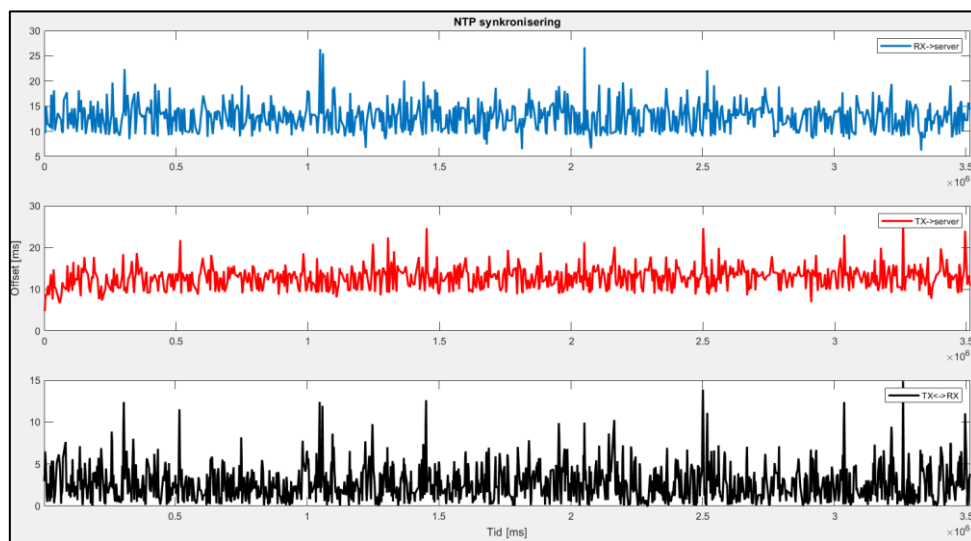
Dette afsnit vil præsentere resultater baseret på en målekampagne kørt med FR2. De rå data er blevet processeret som beskrevet i forrige afsnit, hvor dette afsnit ikke fokuserer på anvendelsen af RF-modeller, men på om der kan validere systemets metoder. Der er tidssynkronisering, position metode evaluering og LOS->NLOS-metode. I **Appendiks Resultater** findes ruterne plottet i miljøet og histogram over RF-målinger til afstand mellem RX og TX. Fra målekampagnen er der cirka 57 min RF-målinger.

9.1 TIDSSYNKRONISERING

Et NTP-monitor script kører under målekampagnen, der samler tidsstempler af klienterne og tidsstempler ved modtagelsen af pakkerne med monitor computerens egen tid. Computeren med NTP-monitor scriptet, sammenligner de modtaget tidsstempler med de tidsstempler som den har lavet for modtagelsen.

Til at sammenligne tiderne mellem klienterne, bliver tidsstemplerne fra monitor computeren brugt til interpolation mellem to tider, sådan at tidsstemplerne hos klienterne kan sammenlignes.

Figur 26 viser de tre offset under målekampagnen og tabel med basal statistik.

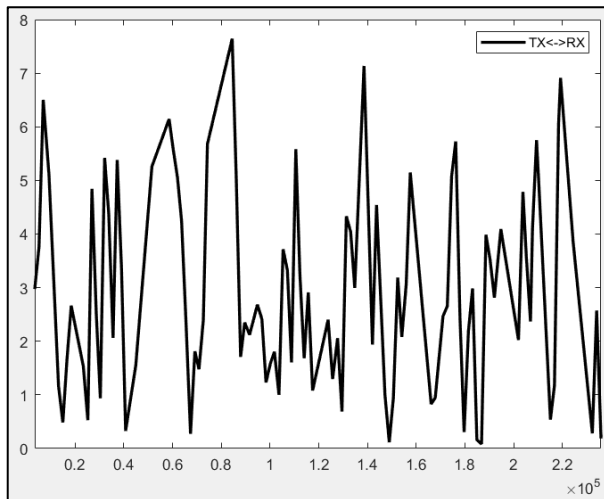


Figur 26: Præstation af NTP-synkronisering, hvor det er server til klient og klient til klient.

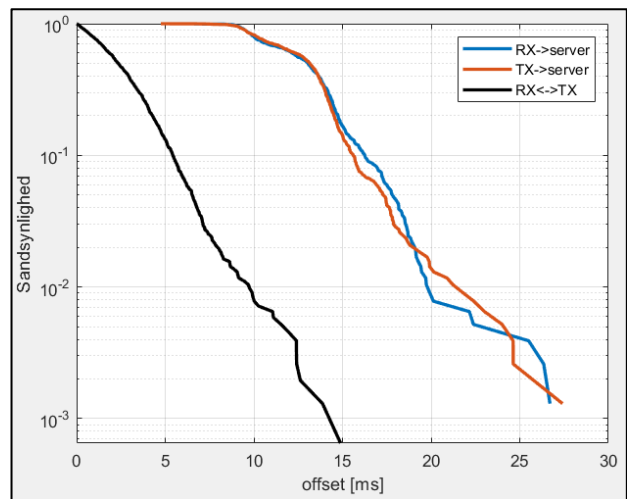
	Min [ms]	Mean [ms]	Max [ms]	Std [ms]
RX-> server	6,177	12,8379	26,706	2,7858
TX->server	4,752	12,7954	27,413	2,6701
RX<->TX	0	2,7344	14,8911	2,0755

Der samles en del data under en kampagne, hvilket derfor Figur 27 viser et begrænset interval, som gør det nemmere for at se adfærden af synkroniseringen.

Figur 28 viser CCDF af synkroniseringen i målekampagnen, hvor klient til server, både RX og TX har 99% af tiden har offset på 20 ms eller lavere. Mellem klienterne er offsetten med 99% af tiden ved 10 ms eller lavere.



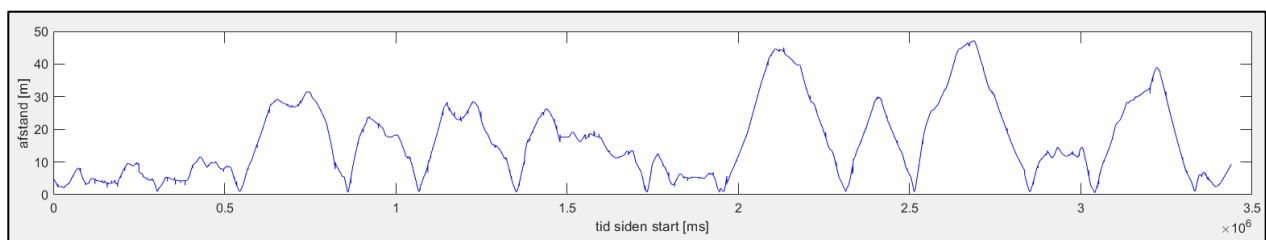
Figur 27: Forkortet interval af Figur 26.



Figur 28: CCDF af synkroniseringsmålinger.

9.2 POSITION

Det første relevante aspekt for positionerne er historikken gennem målekampagnen for afstanden mellem RX og TX, som vist i Figur 29. Der er cirka 12 gange dybe dyk nede ved 1 meter afstanden, hvilket betyder RX og TX har krydset hinandens veje.



Figur 29: Afstand mellem RX og TX under hele målekampagnen.

Hertil er nogle statistiske værdier forhold til målekampagnen:

Minimum	Mean	Maximum	Std
0,6699 m	14,7472 m	47,116 m	9,8142 m

En anden ting med Figur 29, som er værd at lægge mærke til, er der nogle små spikes i kurven som virker en smule unaturlige. Disse spikes kan være et tegn på en fejlkilde med positionssystemet.

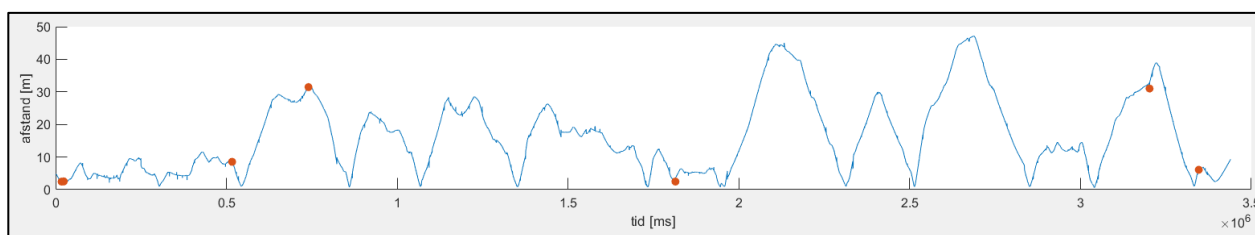
Til at filtrere de spike, som er med sikkerhed en afvigelse og ikke fjerner usikkerhederne, laves der en sanity check. Sanity checket går på om ændringerne i afstand er indenfor det største afstand, baseret på den samlet acceleration.

Det vil sige, hvis de begge kører med 1m/s, så vil den største mulige afstand være 2 meter efter et sekund.

Der bliver udregnet den maksimale forskydning baseret på den sammenlagte acceleration.

$$s = ut + \frac{1}{2} * at^2$$

Hvis afstanden mellem RX og TX overstiger forskydningen, har de rykket sig længere end muligt. Figur 30 viser afmærkninger ved de steder, som blevet fanget til at være afvigelser.



Figur 30: Markeringer af afstand mellem RX og TX, som ikke er mulige baseret på deres hastigheder.

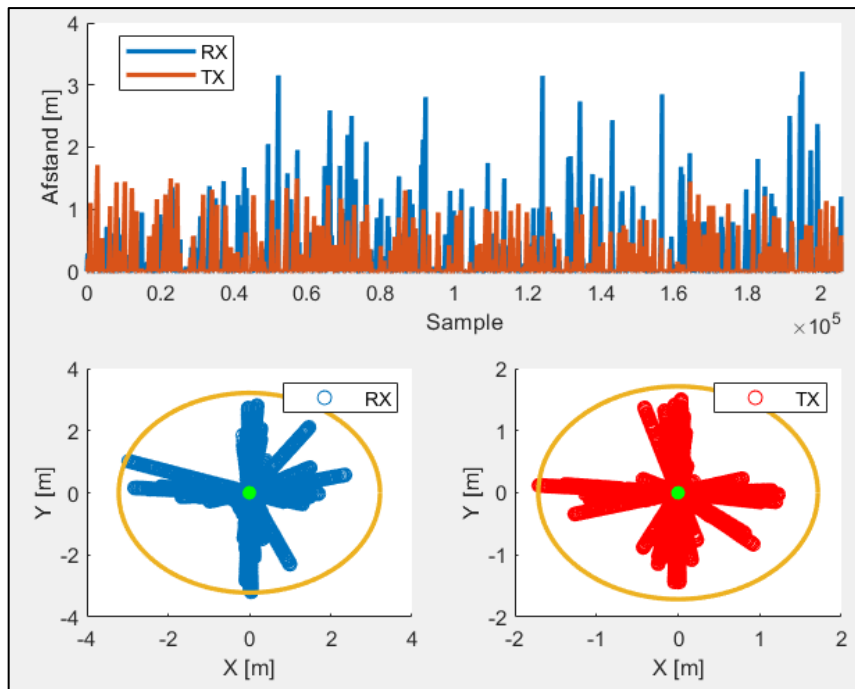
Her er der tale om 8 ud af 21375 målinger, som slår ud som en afvigelse. Blandt de 8 afvigelser gælder følgende hvor meget de afviger for det mulige:

Minimum	Mean	Maximum	Std
0,015 mm	3,57 cm	27,18 cm	9,854 cm

En del af positionerne er interpolation mellem punkter, sådanne RF-målinger kan få positionsdata. Der er to fremgange ved dette, første er at kun bruge de globale positionsdata, hvor der ikke er noget data forhold til hastigheder og derfor laves interpolation på antagelsen af en konstant acceleration.

Den anden metode er at bruge de målte hastigheder i de relative positionsdata.

Figur 31 viser forskellen for RX og TX separat, da der er forskel på hvordan de har kørt. TX har kørt med en betydeligt mere konstant hastighed end RX, men som også blev kontrolleret manuelt.



Figur 31: Forskellen mellem interpolation på en konstant acceleration eller hastighedsmålinger. Cirklerne er en normalisering af målingerne til at illustrerer forskellen.

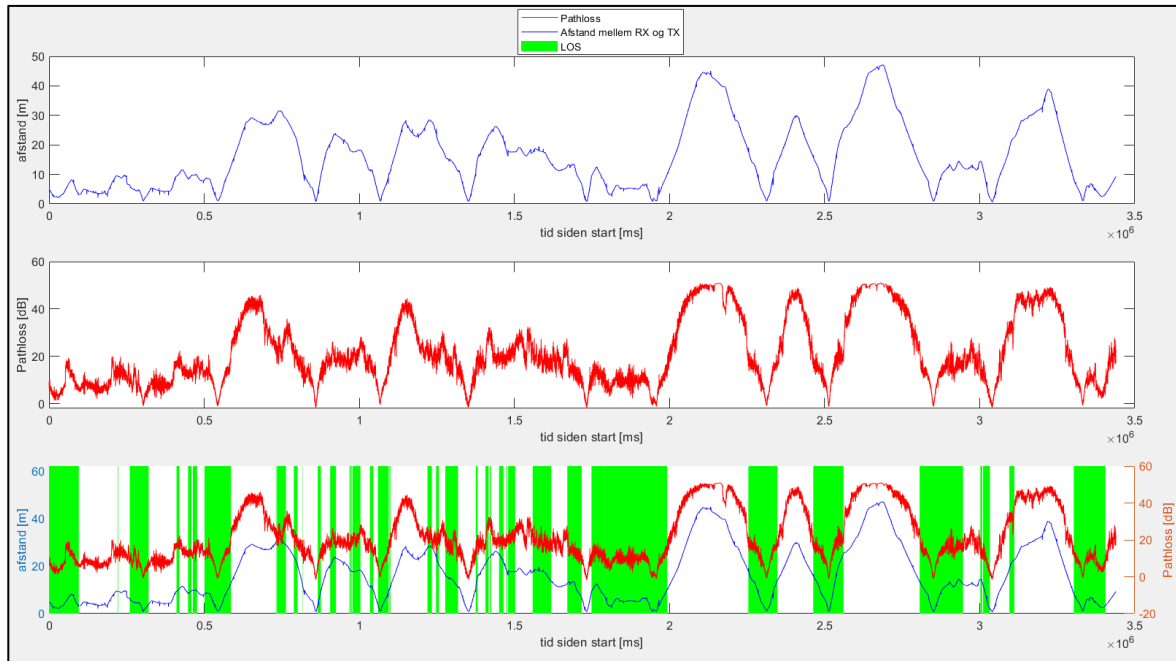
hvor til er de statistiske repræsentation:

	Mean	Maximum	Std
RX	0,1204 m	3,2137 m	0,3545 m
TX	0,0809 m	1,7122 m	0,21829 m

9.3 METODE TIL AT FINDE LOS->NLOS

Figur 32 viser sammenhængen mellem Pathloss og afstand mellem RX og TX, samtidig at vise de steder hvor RX og TX er defineret til at være i LOS. Definitionen bygger på om den lineær linje mellem RX og TX krydser nogle af de defineret vægge i kortet, Figur 42.

Ud fra denne definition befinder RX og TX i LOS i 23 min i målekampagnen, hvilket er cirka 40% af den samlet tid.



Figur 32: Sammenlign mellem afstand og pathloss over tid i kampagnen. Til sidst et filter med markeringer af LOS perioder.

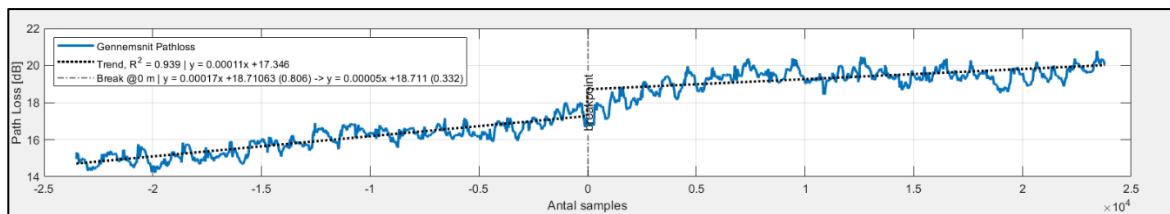
Derefter bliver der set på overgangen fra LOS til NLOS, hvor der bliver valgt de LOS områder, som opfylder to kriterier.

For at få en forståelse af overgangen skal ses på noget tid før den hårde skilning (ud fra definitionen) og noget tid efter. Den hårde overgang vil være når området går fra grøn til hvid.

Lad den hårde overgang være ved t_0 , hvor analysen vil have tid før og efter t_0 . I det interval kan det ikke bruges at der er flere overgange, altså tiden før t_0 skal være i det samme felt af LOS og tiden efter t_0 skal være i det samme felt af NLOS.

Figur 33 viser resultatet efter at tage normalisering af de udvalgte overgange og taget gennemsnittet. Det giver en forskrift på overgangen

$$y = 0,00011x + 17.346$$



Figur 33: Normalisering af de valgte LOS->NLOS-overgange, hvor der er forskrift på gennemsnit.

10 DISKUSSION

I dette projekt er der blevet set på en kanalmålemetode, som anvender robotter til at gøre RX og TX til mobile enheder mens foretage en målekampagne, til at samle data for en kanalmåling, input i en stokastisk model, eller være sanitet check for kvaliteten i deterministiske modeller. Til denne kanalmålemetode har fokusset været robustheden i metoden som en platform, hvor der kan tilføjes et RF-målemetode system ovenpå og lave en form for kanalmåling. Til at vurdere robustheden, har projektet taget udgangspunkt i en case, hvor en RF-målemetode er blevet defineret af Nokia Bell Labs.

Casen for Nokia Bell Labs omhandler D2D kommunikation, hvor der anvendes retningsbestemte antenner med 5G mmWave signaler til fokus om LOS->NLOS.

Denne diskussion vil omhandle de unøjagtigheder, som kommer med designet af kanalmåle platformen, hvorfor de valg forhold til andre og hvordan opfylder platformen krav for en rigtig applikationscase.

10.1 TRADITIONEL KANALMÅLING

De mere traditionelle metoder for kanalmåling, såsom [6] bruger for det meste statiske positioner for RX og TX, hvor en bliver rykket og mens der bliver taget værdier ved de forskellige markeret steder. Nogle gange bliver den ene antenner flyttet manuelt rundt, hvor der bliver signaleret at vognen er ved en markeret position, som bliver registreret af en anden som noterer værdierne.

Sådan en fremgang kræver mange hænder og tid for udførelsen af en målekampagne, da antennen eller antennerne skal flyttes mange gang før kampagnen, er rig på data. Konsekvensen af den høje tidsforbrug under en målekampagne, vil være at ændringen af positionerne vil typiske ikke være lavere end en meter, hvilket misser en masse vigtige data om karakteristikkene over radioudbredelsen, som er vigtig i D2D kommunikation og mmWave signaler.

Kanalmålemetoden med mobile enheder bringer forbedringer til, hvor de traditionelle metoder hænger efter med. Med mobile enheder vil selve målekampagnen tage noget kortere tid, hvor vil få flere gang mere data ind. Det hele vil kunne blive opereret af en enkel person. Metoden med mobile enheder kan blive set som en platform, da RF-systemet kan blive udskiftet med et andet og bibeholde anvendelsen.

En anden vigtig forskel er, at kanalmålemetoden med mobile enheder er mere eller mindre 100% objektivt forhold til den traditionelle metode, hvor de fleste dele er subjektive; den ukendte hastighed af vognen, reaktionstiden af personerne ved notering af data og præcisionen af positionerne.

Kanalmålemetoden med mobile enheder introducerer dog en masse kompleksitet i form af alle dens delsystemer, hvilket har usikkerheder som definerer brugbarheden af

metoden. Fordelen med alt bliver styret af computer kode, så kan processere udføres hurtigere end menneskeligt muligt, hvilket åbner op for en masse applikationer.

10.2 SYNKRONISERING AF ROBOTPOSITIONER OG RF MÅLINGER

Synkronisering af robotpositioner og RF-målinger er kernen i denne slags kanalmålemetode, hvor dette projekt fokuseret ekstra meget på dette i forhold til påvirkninger af platformens ydeevne. Dette projekt har valgt NTP vejen som løsning, men der alternative løsninger, hvor ikke alle ligger indenfor tidssynkronisering.

På skrivende stund, vil alternativet indenfor tidssynkronisering kun være mellem brug af GPS-rubidium ure hardware, som beskrevet i [6] og brug af PTP-hardware. Både rubidium og PTP-løsning vil have den samme fremgang, hvilket vil være at synkronisere enhederne lige før en måler kampagne, hvor grunden til denne tilgang er ingen af dem kan blive anvendt i trådløse systemer.

Der også bliver lavet en back-to-back måling af offsetten lige før og lige efter målekampagnen, da under målekampagnen vil enhederne drifte i deres interne ure. Resultatet fra back-to-back målingen vil give start offset og slut offset, hvor brugbarheden afhænger af driften er lineær, som antaget af de fleste ure er og dermed en model over offsetten over tid i målekampagnen.

Konsekvensen ved denne fremgang, at der er uvist om den har driftet lineært lige i den pågældende målekampagne, hvilket betyder det kan kun bruges til at sætte nogle estimeret nedre og øvre grænser for offsetten.

De største ulemper og udfordringer med rubidium og PTP-løsninger er, at det er meget dyre løsninger forhold til to Raspberry Pi's. De kræver også en masse tid med opvarmning af udstyr, hvor rubidium hardwaret er op til 72 timer før selve målekampagnen, hvilket modarbejder ideen om den givet fleksibilitet i målemetoden.

NTP er kun i millisekunder niveauet, mens PTP er nede i nanosekunder og picosekunder, og rubidium er helt nede på 10^{-9} sekunder om dagen nøjagtighed.

hvilket ikke betyder særlig meget, når der ses på en drift model på Raspberry Pi, som har en drift på 100 PPM, hvilket er meget normalt. Ved at anvende aktiv NTP under målekampagnen, vil offsetten ligger gennemsnitlig omkring de 3 ms og under de 30 ms med den realtids monitoreringsmetode, hvor forsinkelsen af pakkerne er med regnet. I [1] blev der lavet en back-to-back måling på en gennemsnit på 1 ms og 4,5 ms i std.

I forbindelse med at forstå algoritmen i NTP, en simplificeret simulering blev lavet, hvor resultaterne viser en adfærd i synkroniseringen, som ligner meget den adfærd de empiriske målinger har.

I projektet bliver NTP monitoreret realtid under målekampagnen, hvilket sker med trådløs kommunikation hvilket forøger afbildning af offsetten, da der er noget tilføjeret forsinkelse i sendingen af pakken med tidsstemplerne. Dette gør ikke noget, da det giver

en øvre grænse ved at den reelle offset er streng lavere end angivet, derfor når disse målinger viser bedre nøjagtighed end kravet, er der ingen problemer.

Rubidium og PTP vil praktisk begynde med nul i offset, men et ur med 100 PPM drift vil drifte med 360 ms i timen.

Alternativt til tidssynkronisering vil være en integration med RF-systemet, sådan der brugs bestemte signaler såsom SSB-signal eller sende data over bæresignalet. Dette er dog langt væk fra ekspertise.

10.3 NTP TOPOLOGIEN

I projektet er der også blevet taget nogle valg angående topologien af NTP, hvor man kunne godt have godt andet. Der er en Raspberry Pi konfigureret til at være NTP server koblet direkte til routeren, hvor der mindsker antal af trådløst hop mellem server og klienter. Ideen med denne topologi er at Raspberry Pi ved RX og TX er den lokale server for deres AGV-robot, da der kan de være forbundet stabilt med kabel, hvor det er muligt at tilføje flere enheder, hvilke kunne være dele af et RF-system.

Alternativt kunne Raspberry Pi på RX være serveren, hvor med Raspberry Pi synkronisere direkte til den anden enhed, men det giver double så mange trådløse hops til den ene Raspberry Pi, hvilket er den største usikkerhed til en stabil synkronisering.

For at øge stabiliteten bliver der brug WiFi 6, hvilket er den bedste trådløse løsning i skrivende stund.

10.4 MiR SOM EN DEL AF PLATFORMEN

Den anden store del er positionssystemet, hvor i dette projekt anvender den gennemførte og gennemtestet robot MiR, hvilket er et produkt som er optimeret til kører i et indendørs miljø og softwaret kører på ROS. Brugen af ROS gør det nemmere at få data for positionen, samt åbner op for muligheder til at styre det meste af robot.

Men de er ikke den eneste AGV såsom OMRON® [15], hvilket har de samme funktioner såsom en Fleet og styring af robot. OMRON lover præcision på ± 10 mm i position ved kører hen til stationer og $\pm 1^\circ$ i rotations nøjagtighed.

Af praktiske årsager blev det MiR som valgt AGV, da universitet har valgt dem til forskningsprojekterne.

Alternativet til en professionel færdigbygget AGV løsning ville være at lave sin egen robot med positionssystem. Det er dog ude af rækkevidde af projektet, da det vil kræve lave alle controllernes fra bunden og få det til at passe med forskellige sensorer.

Hvis det var løsningen, ville den hjemmelavet robot højst sandsynlig også anvende ROS, da det er open source og indeholder alt nødvendigt i et framework med en masse dokumentation.

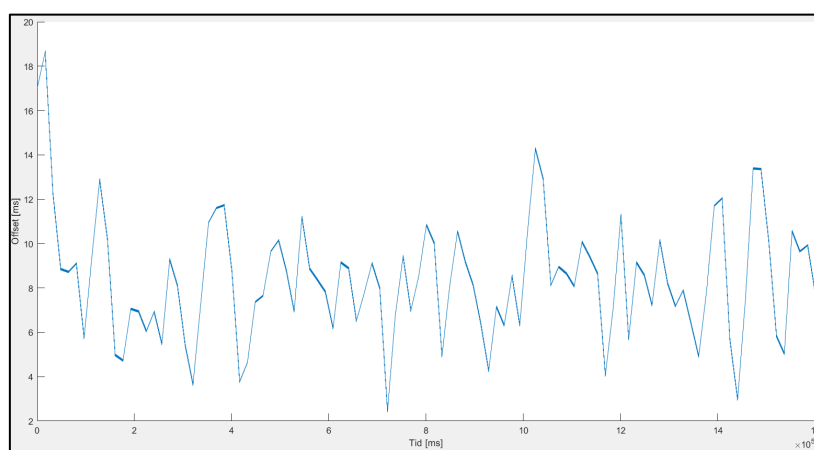
11 KONKLUSION

En metode for kanalmåling, som bruger robotter til at gøre antennerne mobile har uden tvivl en masse applikationer at lave kanalmåling for. Sådan en metode kan blandt andet være anvendelig i casen fra Nokia Multiple tone RF-metode, da det har vist sig at den opfylder casen krav om stabilitet.

I dette system og alle elektroniske systemer, opstår der afvigelser i alle målinger da elektroniske systemer er sårbar over for støj. Samt hver gang der er en hændelse af en operation, så vil der være afvigelse i udførelsen og opfattelsen af hvornår det skete.

Det er applikationen som bestemmer præcisionen af systemet, hvilket i dette projekt har sat krav om minimum 50 m's nøjagtighed i tidssynkroniseringen og maksimum 20 cm afvigelse på positionen.

Til at forstå mulige afvigelser i NTP, blev der lavet en simulering af NTP baseret på strukturen og algoritmerne. Figur 34 viser adfærden af synkroniseringen, hvor den største synder er fundet, hvilket er den asymmetriske netværkskommunikation.

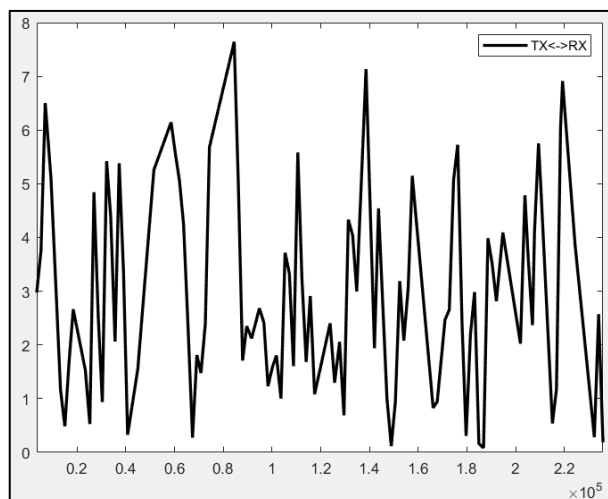


Figur 34: Synkronisering ved asymmetrisk netværk kommunikation.

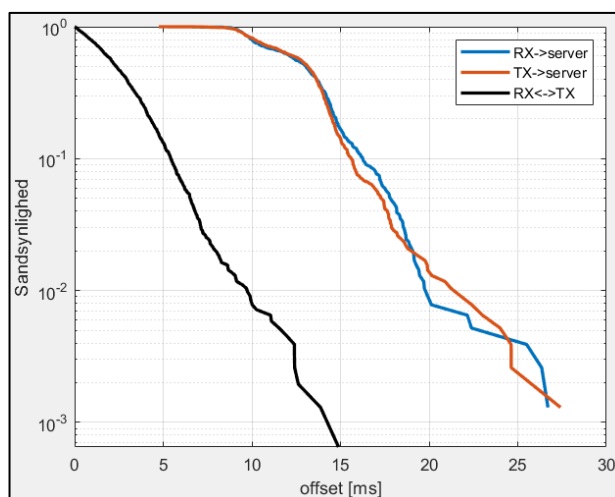
Yderligere end simulering, blev der lavet fysiske målinger i en målekampagne i realtid, hvilket gav temmelig gode resultater med gennemsnit på 2,7 ms mellem klienterne.

	Min [ms]	Mean [ms]	Max [ms]	Std [ms]
RX-> server	6,177	12,8379	26,706	2,7858
TX->server	4,752	12,7954	27,413	2,6701
RX<->TX	0	2,7344	14,8911	2,0755

Der blev set på et mindre interval, Figur 35 som bekræfter adfærden i den simuleret målinger. Måske der er et maksimum af 14,8 m's offset mellem klienterne, men ifølge CCDF i Figur 36 er 99% af tiden under 10 ms.

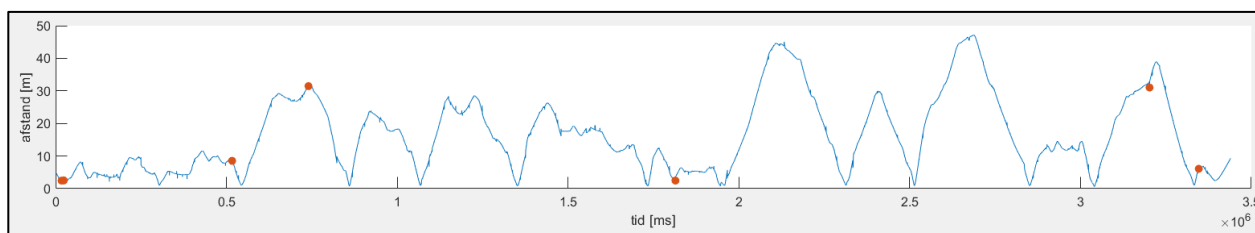


Figur 35: Forkortet interval af Figur 34.



Figur 36: CCDF af synkroniseringsmålinger.

Målinger af positionerne, hvor Figur 37 repræsenterer afstanden mellem RX og TX, hvor der bliver fundet nogle afvigelser, at de stikker ud fra mulige ændring af afstand baseret på den maksimale afstand ændring.



Figur 37: Markeringer af afstand mellem RX og TX, som ikke er mulige baseret på deres hastigheder.

Der er otte afvigelser, hvor en af dem er over det tilladte forhold til RF-casen. De otte har en gennemsnit på 3,57 cm.

Minimum	Mean	Maximum	Std
0,015 mm	3,57 cm	27,18 cm	9,854 cm

Ud fra en definition af LOS, resulterer det i en forskrift for pathloss ved LOS->NLOS, hvor x er tid og $x=0$ er den defineret LOS->NLOS.

$$y = 0,00011x + 17.346$$

12 REFERENCER

- [1] R. D. Marcker, »5G Channel Sounding Between Mobile Robots in an Industrial Setting @3.57 GHz and 26.43 GHz,« Aalborg University, Aalborg, 2020.
- [2] A. F. Molisch, Wireless Communications, West Sussex, England: John Wiley & Sons, LTD - IEEE Press, 2005.
- [3] M. Kottkamp, A. Pandey, D. Raddino, A. Roessler og R. Stuhlfauth, 5G Newo Radio - Fundamentals, procedures, testing aspects, München: Rohde&Schwarz GmbH & Co. KG, 2019.
- [4] H. Hussein, H. Elsayed og S. Abd El-Kader, »Intensive Benchmarking of D2D communication over 5G cellular networks: prototype, integrated features, challenges, and main applications.,« *Springer - Wireless Networks*, årg. 26, p. 6, 2020.
- [5] R. Candell, K. A. Remley, J. T. Quimby, D. Novitny, A. Curtin, P. B. Papazian, K. Mohamed og J. Diener, »Industrial Wireless Systems Radio Propagation Measurements,« National Institute of Standards and Technology, 2017.
- [6] J. T. Quimby, R. Candell, K. A. Remley, d. Novotny, J. Diener, P. B. Papazian, A. Curtin og G. Koepke, »NIST Channel Sounder Overview and Channel Measurements in Manufacturing Facilities,« National Institute of Standards and Technology, New York, 2017.
- [7] Mobile Industrial Robots A/S, »MiR hjemmeside,« Mobile Industrial Robots A/S, 2019-2020. [Online]. Available: <https://www.mobile-industrial-robots.com/en/solutions/robots/>. [Senest hentet eller vist den 21 marts 2021].
- [8] Open Robotics, »ROS,« Open Robotics, [Online]. Available: <https://www.ros.org/>. [Senest hentet eller vist den 4 Marts 2021].
- [9] Open Robotics, »Rosmsg,« [Online]. Available: http://wiki.ros.org/geometry_msgs. [Senest hentet eller vist den 23 Februar 2021].
- [10] »Cisco Meraki,« Cisco Systems, Inc., 2020. [Online]. Available: https://documentation.meraki.com/MR/MR_Overview_and_Specifications/MR56_D atasheet. [Senest hentet eller vist den 2 april 2021].
- [11] D. D. L. Mills, J. Martin, J. Burbank og W. Kasch, »Network Time Protocol Version 4: Protocol and Algorithms Specification,« Juli 2010. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc5905>.

- [12 M. Burnicki, »Meinberg,« 4 april 2019. [Online]. Available:
] https://kb.meinbergglobal.com/kb/time_sync/time_synchronization_accuracy_with_ntp. [Senest hentet eller vist den 24 maj 2021].
- [13 D. Mills, »Network Time Synchronization Research Project,« 13 Maj 2012. [Online].
] Available: <https://www.eecis.udel.edu/~mills/ntp.html>. [Senest hentet eller vist den 10 juni 2021].
- [14 E. Furey, »Calculator Soup - onlube calculators,« 2016-2021. [Online]. Available:
] https://www.calculatorsoup.com/calculators/physics/displacement_v_a_t.php.
[Senest hentet eller vist den 13 maj 2021].
- [15 Omron, 2021. [Online]. Available: <https://industrial.omron.dk/da/products/hd-1500>.
] [Senest hentet eller vist den 25 juni 2021].
- [16 »A-info,« Antenne producent, 20 april 2020. [Online]. Available:
] <http://www.ainfoinc.com.cn/NL/200420/200420.html>. [Senest hentet eller vist den 26 april 2021].

13 APPENDIKS

13.1 RESULTATER FRA SIMULERING AF NTP

13.21. FORSØG

Pakke	1	2	3	4	5	6	7	8
$\delta/2$	19	13	21	15	25	14	27	11
intern	1	3	1	2	2	3	2	2
t_1	2002	4004	6006	8008	10010	12012	14014	16016
t_2	2002	3996	6004	7998	10008	11997	14010	15994
t_3	2003	3999	6005	8000	10010	12000	14012	15996
t_4	2041	4033	6049	8040	10062	12043	14070	16040
Delay	38	26	42	30	50	28	54	22
Offset	-18,983	-20,983	-22,983	-24,983	-26,983	-28,983	-54,983	-32,983
ε	5,8e-5	4,35e-5	6,45e-5	4,8e-5	7,8e-5	4,65e-5	8,4e-5	3,6e-5

θ_0	δ_0	ε	φ	x	y	z	w	Θ	ϑ
-32,983	22	4,14e-5	11,4018	11,0015	0,0909	-1,7255	196	-18,983	46,436

Den reelle offset er

$$\theta' = \theta_{start} + \phi_c * t = 17 + 100 \text{ PPM} * 16000 = 33 \text{ ms}$$

13.32. FORSØG

Første poll event:

θ_0	δ_0	ε	φ	x	y	z	w	Θ	ϑ
-32,983	20	4,17e-5	8,366	10,0015	0,1	-1,898	196	-18,983	44,2753

Andet poll event:

θ_0	δ_0	ε	φ	x	y	z	w	Θ	ϑ
-48,983	22	4,11e-5	8,366	11,0015	0,0909	-3,1798	196	-34,983	46,436

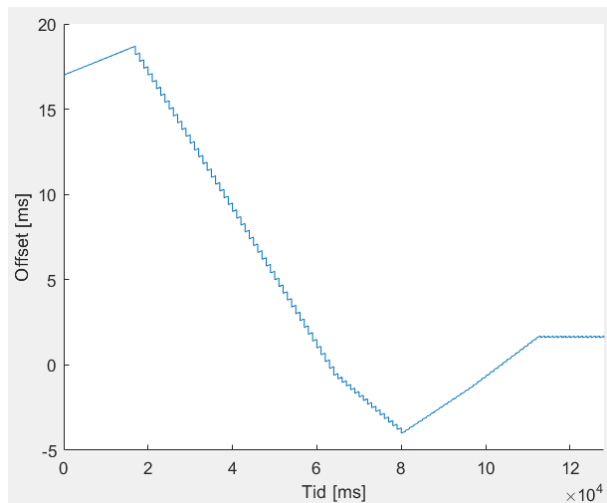
57,4375	0,0174	-0,6091	-34,983	106,4
---------	--------	---------	---------	-------

Den reelle offset er

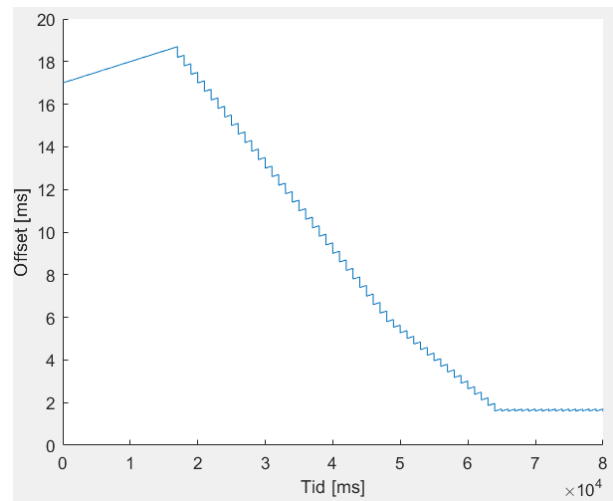
$$\theta' = \theta_{start} + \phi_c * t = 17 + 100 \text{ PPM} * 32000 = 48 \text{ ms}$$

13.43. FORSØG

Resultatet er i **Fejl! Henvisningskilde ikke fundet.** med run time lig med otte poll events.



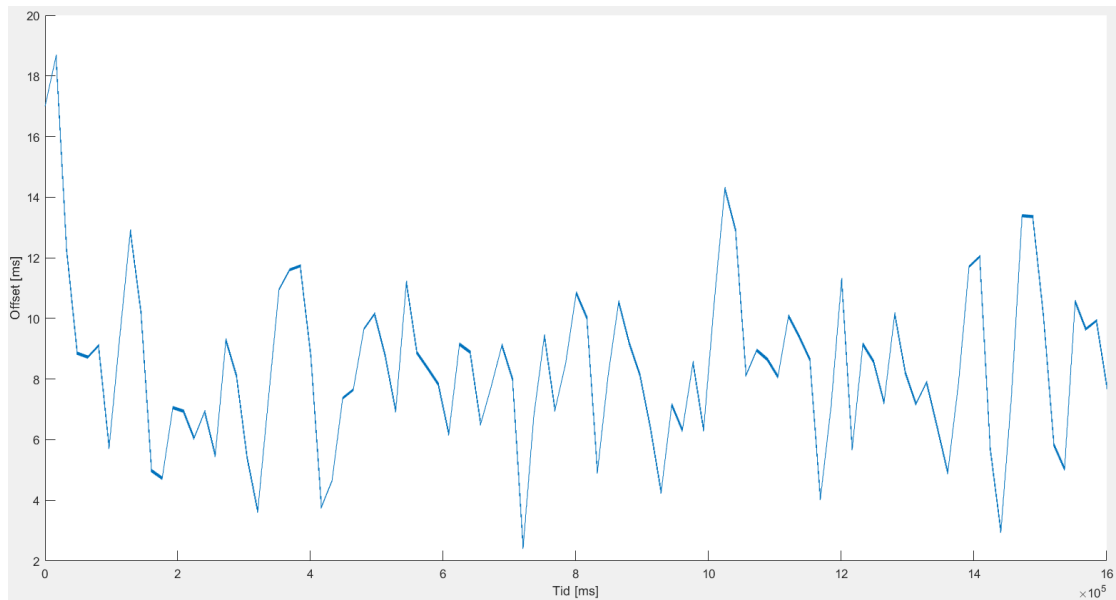
Figur 38: 8 poll events til steady-state med θ_0 .



Figur 39: 5 poll events til steady-state med Θ .

Begge test går i steady-state ved en offset på cirka 1,6-1,7 ms, hvilket betyder at driften svarer til justeringen.

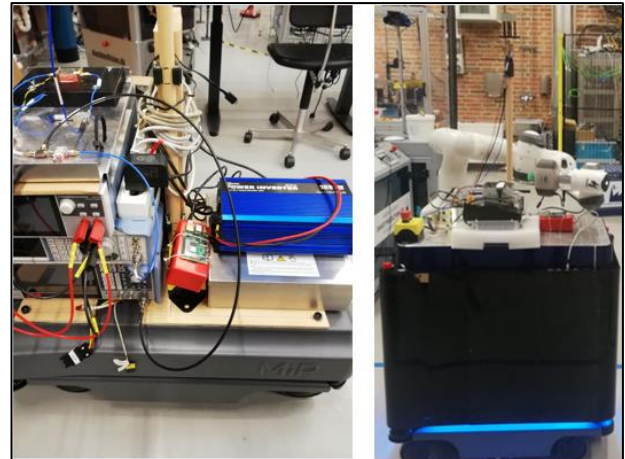
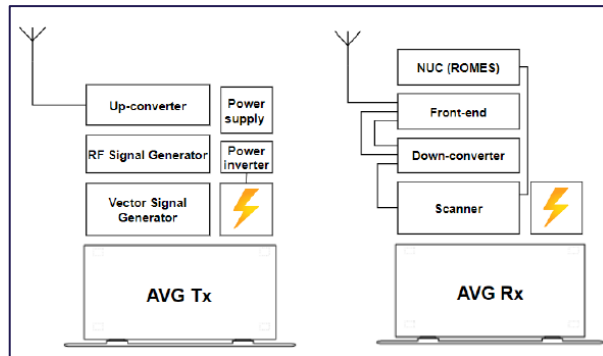
13.54. FORSØG



Figur 40: offset resultater ved forsøg 4.

13.6 RF MÅLEUDSTYR OG OPSÆTNING

Denne del er kun en smule anderledes for "Proof-of-Concept" projektet, det er samme styr og opsætning, der er nok nogle komponenter ved TX sådan der nogle andre frekvensværdier. Opsætningen kan ses i Figur 41.



Figur 41: RF-måleudstyr på TX og RX. [4]

Det meste af udstyret er fra firmaet Rohde & Schwartz®, hvor blandt TX er en signalgenerator og en LO-generator, samt en scanner i RX siden og en NUC med softwaret ROMES™.

Ligesom i "Proof-of-Concept" projektet, bliver der anvendt to frekvens range, hvor signalgeneratoren er indstillet henholdsvis:

FR1 = 3570,06 MHz med power level = -5 dBm

FR2 = 5,4324 GHz med power level = -15 dBm

Ved FR2 bliver LO generatoren forbundet til en mixer sammen med signalgeneratoren, hvor LO generatoren er sat til 21 GHz, +5.2 dBm i power level.

Signalet som bliver sendt, er et SSB-signal, som er en gruppering af 2 og 4 signaler henholdsvis FR1 og FR2, hvor scanneren operere ved 50 Hz. Før målingerne er brugbar i forhold til kanalmodellering, skal signal styrke målinger sættes op mod en reference af udstyret. I dette projekt laves der også reference målinger OTA (over the air), hvor der måles med RX ved flere steder rundet om TX med afstand på en meter.

Referencemålingen ved FR1 og FR2 og gennemsnit af målinger ved en meter afstand:

Reference ved FR1 = -46,2 dBm

Reference ved FR2 = -34,9 dBm

Hvilket er en forbedring henholdsvis på cirka 10 dBm og 3 dBm. Der bruges den samme omni antenne, model SZ-3004000/P [16].

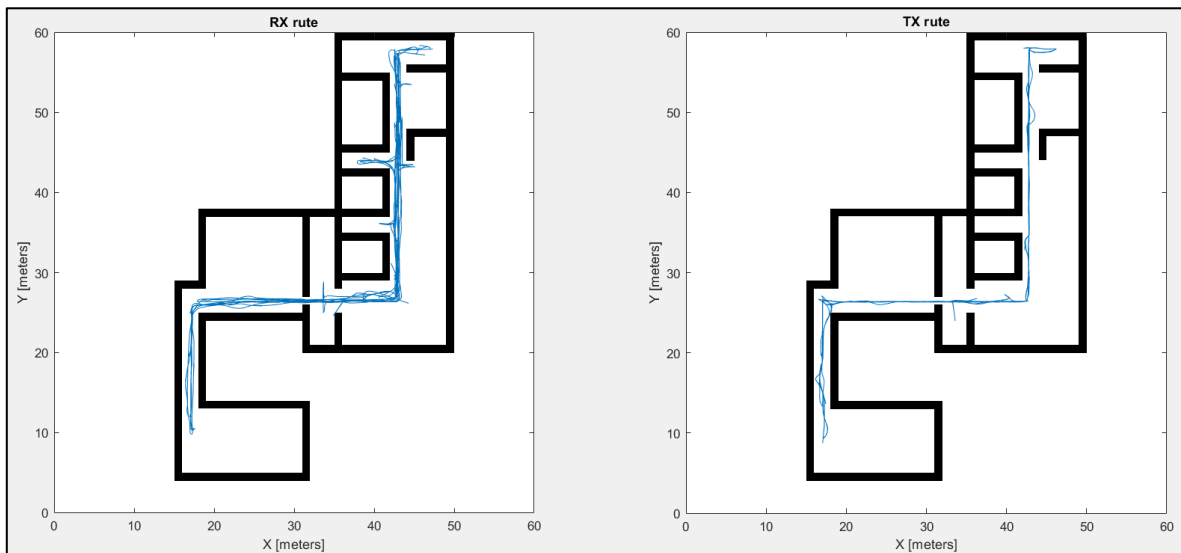
13.7 SSH KEY GEN

Dette er fremgangen til at generere en ssh nøgle, som gør det muligt at SSH fra host A til host B uden at angive password hver gang.

Noterer at user 'a' på host A og user 'b' på host B:

- 1) a@A:~> ssh-keygen -t rsa
- 2) a@A:~> cd /. / # **flyt nuværende mappe til /... /.ssh/id_rsa**
- 3) a@A:~> ssh b@B mkdir -p .ssh
- 4) **Indtast b@B's kodeord**
- 5) a@A:~> cat .ssh/id_rsa.pub | ssh b@B 'cat >> .ssh/authorized_keys2'

13.8 RESULTATER



Figur 42: Plottet rute af RX og TX i målekampagnen,