

# Simulering av mobile ad hoc nettverk ved hjelp av topologiemulator

**Morten Gjellerud**

Master i kommunikasjonsteknologi

Oppgaven levert: Juni 2007

Hovedveileder: Tor A Fjeldly, IET

Biveileder(e): Vivianne Jodalen, Forsvarets forskningsinstitutt



### Oppgavetekst

Implementere nødvendig dynamikk i topologi emulatoren. Sette opp aktuelle scenarier for simuleringer av ad hoc nettverk.

Oppgaven gitt: 15. januar 2007  
Hovedveileder: Tor A Fjeldly, IET



# Sammendrag

En topologiemulator er en innretning som muliggjør transmisjon mellom et antall radiostasjoner via ledninger og komponenter, og som søker å modellere reell trådløs transmisjon. Fordelen med en slik emulator er at nettverkstopologien kan manipuleres og konfigureres slik at maskin- og programvare kan testes og verifiseres i et kontrollerbart miljø, under forhold som er praktisk sett enkle. Knyttet til begrepet om Mobile Ad Hoc Networks (MANET) er en topologiemulator spesielt gunstig. Med emulatorens kan man da nytte et vidt spekter av aktuelle radiosystemer kombinert med ulike protokoller, og på en relativt enkel måte verifisere virkemåte og ytelse. Den primære fordelen er at med en topologiemulator kan dette foregå på laboratoriet, noe som holder både kostnader, kompleksitet og praktiske forhold på et akseptabelt nivå når det gjelder innledende tester og forsøk.

Arbeidet med hovedoppgaven har dreiet seg om denne topologiemulatorens emulering av trådløse kanaler, i Very High Frequency (VHF) og Ultra High Frequency (UHF) båndene mellom 200 Megahertz (MHz) og 3 Gigahertz (GHz). Oppgaven startet der prosjektoppgaven fra faget TFE 4605, Radioteknikk og Fjernmåling fordypning sluttet. Prosjektoppgaven tok for seg en studie og en kalibrering av systemet med en i hovedsak todelt intensjon til grunn. Primært var målsetningen å kalibrere emulatorens, sekundært og implisitt i dette å kartlegge karakteristikk, muligheter og begrensninger ved systemet. Arbeidet endte med en oversikt over hvordan emulatorens kanaler påvirker hverandre, samt en dokumentasjon for hele systemet. Kanalenes påvirkning på hverandre er kvantisert ved hjelp av et sett med matriser som gjengir det dynamiske området hver kanal kan opereres i for å gjenskape de ulike situasjoner som normalt assosieres med overføring av radiosignaler. I hovedoppgaven har arbeidet fortsatt fra dette, med å utvikle og implementere en dynamikk på to områder som anses vesentlige for utvikling av emulatorens som et nyttig verktøy for instituttet. Ulike mobilitetsmodeller har blitt studert og vurdert for å kunne utføre en kvalifisert simulering av radionoders posisjoner og bevegelse. Dette har resultert i en implementering av to ulike modeller for mobilitet. Som en logisk fortsettelse av dette har jeg studert modeller for propagasjon i det aktuelle frekvensområdet, både stokastiske og deterministiske, og så implementert en modell som passer systemet og dets bruksområde. Disse mekanismene er så sydd sammen, knyttet til et brukervennlig grensesnitt, og utgjør totalt sett et velfungerende system for prøver og forsøk i forbindelse med MANET. Systemets styrke er at en stor del av utfordringene knyttet til simulering av fysisk lag elimineres da det faktiske radiomateriellet kan nyttes.



# Forord

Denne rapporten kommer som en konklusjon etter mine aktiviteter i faget TFE 4900, Signalbehandling og kommunikasjon, masteroppgave. Faget er gjennomført i tiende semester av sivilingeniørstudiet i kommunikasjonsteknologi, studieretning signalbehandling og kommunikasjon. I mitt tilfelle er faget administrativt sett gjennomført ved Universitetsstudiene på Kjeller (UniK), mens arbeidet er utført i sin helhet ved Forsvarets forskningsinstitutt (FFI), avdeling for Ledelsessystemer.

Gjennom arbeidet med oppgaven har jeg kunnet fortsette der prosjektoppgaven fra niende semester sluttet, og således fått anledning til å sette meg godt inn i topologi-emulatorens karakteristikker og egenskaper. Jeg har kunnet fordype meg i ulike mobilitetsmodeller, hva gjelder karakteristikker, matematiske egenskaper og sammenhenger, samtidig som mye av tiden har gått med til studier av propagasjonsmodeller for overføring av radiosignaler. Jeg føler derfor at jeg har fått god kjennskap til de utfordringene som er en del av fagfeltet knyttet til teorien rundt overføring av radiosignaler, og jeg har fått god kjennskap til hvilke muligheter og begrensninger man har når det kommer til forsøk på modellering av virkeligheten på dette området.

Utover dette har arbeidet gjennom både niende og tiende semester ført til at jeg har opparbeidet meg god kompetanse innen National Instruments programmeringspakke LabVIEW, og kommunisering med deres DAQPad-system<sup>1</sup> for kontrollsignaler. Gjennom hele arbeidet har signalgenerering og analyse blitt foretatt med nettverksanalysatorer av typen Rohde og Schwarz ZVC og ZVRE, og jeg har således opparbeidet meg god kompetanse på bruken av denne instrumentfamilien til analyse av et nettverks egenskaper.

*Morten Gjellerud*  
*Kjeller, 12. juni 2007*

---

<sup>1</sup>Data Acquisition (DAQ)





# Innhold

Sammendrag . . . . .	i
Forord . . . . .	iii
<b>1 Innledning</b>	<b>1</b>
<b>2 Bakgrunn</b>	<b>5</b>
2.1 Definerings av oppgave . . . . .	5
2.2 Begrensninger . . . . .	5
2.3 Fremgangsmåte . . . . .	6
<b>3 Logisk oppbygging av emulatorsystemet</b>	<b>7</b>
3.1 Arven . . . . .	7
3.1.1 Maskinvare . . . . .	7
3.1.2 Programvare . . . . .	7
3.2 Modulinndeling . . . . .	10
3.2.1 Programløkke 1 . . . . .	10
3.2.2 Programløkke 2 . . . . .	10
3.2.3 Posisjonsmodul . . . . .	10
3.2.4 Avstandsmodul . . . . .	11
3.2.5 Tegnemodul . . . . .	11
3.2.6 Kanalmodul . . . . .	11
3.2.7 Emulatorkontrollmodul . . . . .	11
3.2.8 Grensesnitt . . . . .	11
<b>4 Posisjonsmodul</b>	<b>13</b>
4.1 Behovet for en posisjonsmodul . . . . .	13
4.2 Virtuelt kart . . . . .	13
4.3 Bevegelse og posisjon . . . . .	14
4.4 Mobilitetsmodeller . . . . .	14
4.4.1 Individuelle modeller . . . . .	15
4.4.2 Gruppemodeller . . . . .	18
4.5 Valg av modeller . . . . .	19
4.5.1 Gauss-Markov-modellen . . . . .	19
4.5.2 Stjerneformet bevegelse . . . . .	20

4.6	Avstandsmodul . . . . .	20
4.7	Tegnemodul . . . . .	22
<b>5</b>	<b>Kanalmodul</b>	<b>23</b>
5.1	Innledning . . . . .	23
5.2	Frittroms modellen . . . . .	23
5.3	Hensyn til hindringer . . . . .	24
5.4	Flat-jord modellen . . . . .	24
5.5	Egli's modell . . . . .	26
5.6	Kategorisering av effekter . . . . .	26
5.6.1	Stor-skala . . . . .	27
5.6.2	Mellom-skala . . . . .	27
5.6.3	Liten-skala . . . . .	27
5.6.4	Resulterende svekkelse . . . . .	28
5.6.5	Tidsspredning . . . . .	28
5.7	Kompleksitet og modeller . . . . .	29
5.8	Faste forbindelser . . . . .	29
5.9	Makroceller . . . . .	30
5.9.1	Empiriske modeller . . . . .	31
5.10	Valg av kanalmodell . . . . .	32
5.10.1	Banetap . . . . .	33
5.10.2	Skyggevirkning . . . . .	33
5.10.3	Flerbanetap . . . . .	33
5.10.4	Oppsummering . . . . .	34
<b>6</b>	<b>Emulatorkontroll</b>	<b>35</b>
6.1	Arv . . . . .	35
6.2	Utfordring og detaljer . . . . .	35
6.3	Løsning . . . . .	36
<b>7</b>	<b>Grensesnitt</b>	<b>37</b>
7.1	Overordnet kontroll . . . . .	37
7.2	Verdier og parametre . . . . .	37
7.2.1	Korreksjonsmatriser . . . . .	39
7.3	Programkontroll . . . . .	40
7.4	Presenterte data . . . . .	40
7.5	Lagring av data . . . . .	40
<b>8</b>	<b>Konklusjon</b>	<b>43</b>
<b>A</b>	<b>Forkortelser</b>	<b>47</b>
<b>B</b>	<b>Kalibrering</b>	<b>49</b>
B.1	Korreksjoner . . . . .	49

<b>C</b> Eksempler fra mobilitetsmodell	<b>53</b>
<b>D</b> Eksempel fra kanalmodul	<b>57</b>
<b>E</b> Instrumenter for signalgenerering og analyse	<b>59</b>



# Figurer

3.1	Systemet montert i rack . . . . .	8
3.2	Topologiemulatorens forside . . . . .	9
3.3	Logisk oppbygging av systemet . . . . .	12
4.1	Avstand . . . . .	14
4.2	Refleksjon . . . . .	16
4.3	Rundhopp, med oppdatering av hastighet og retning i punkt 4. . . . .	17
4.4	Retningsendring ved grensetilfeller . . . . .	21
5.1	Skisse av refleksjon og to-stråle prinsippet . . . . .	25
5.2	Prinsippskisse for modellering av avbøying . . . . .	25
5.3	Prinsippskisse for modellering av spredning . . . . .	25
7.1	Grensesnittet mot brukeren . . . . .	39
B.1	Eksempel på kalibreringsmatrise . . . . .	52
C.1	Bevegelse med Gauss-Markov modell, én node, 100 iterasjoner . . . . .	53
C.2	Bevegelse med Gauss-Markov modell, én node, 10000 iterasjoner . . . . .	54
C.3	Bevegelse med Gauss-Markov modell, seks noder, 100 iterasjoner . . . . .	54
C.4	Bevegelse med Gauss-Markov modell, seks noder, 5000 iterasjoner . . . . .	55
C.5	Bevegelse med stjernemodell, tre noder, 100 iterasjoner, 20 meter ut maks . . . . .	55
C.6	Bevegelse med stjernemodell, seks noder, 100 iterasjoner, 20 meter ut maks . . . . .	56
D.1	Banetap over 500 meter . . . . .	57
D.2	Banetap over 500 meter, med skyggeeffekter . . . . .	58
D.3	Sammensetning av Rayleigh og Rice verdier . . . . .	58
E.1	Nettverksanalysator Rohde & Schwarz ZVC . . . . .	59
E.2	Nettverksanalysator Rohde & Schwarz ZVRE . . . . .	60



# Tabeller

5.1	Tabell over frisiktkoeffisienten $k$ som følge av avstand . . . . .	34
B.1	Tabell over innsettingsdempningene i emulatorene . . . . .	49
B.2	Tabell over maksimal forbindelsesdempning i emulatorene . . . . .	50
B.3	Dempeverdier for kanalkvaliteter ved direkteforbindelser . . . . .	50
B.4	Dempeverdier for kanalkvaliteter ved viaforbindelser . . . . .	51





# Kapittel 1

## Innledning

I nettverksterminologien generelt, og i aspektene knyttet til fremtidig mobil datakommunikasjon spesielt, kommer man raskt i befatning med prinsippene om såkalte MANET. Disse anses mer eller mindre som premissleverandører for det som omtales som fjerde generasjons mobiltelefonsystemer (4G) [1]. Sammenlignet med dagens celleinndelte mobiltelefonsystemer, og også Wireless Local Area Network (WLAN)-nettene, skiller prinsippene om MANET seg vesentlig ut. Det mest iøyenfallende er nok det sterkt reduserte behovet for infrastruktur som følge av en sterk desentralisering av intelligensen i nettverket og det dette innebærer for radionodene i nettverket. Her kan nevnes utviklingen av programvaredefinerte radioer, Software Defined Radio (SDR). Kort sagt kan man si at der dagens kjente systemer primært er totalt avhengige av direkte kontakt med en sentral tjener som relativt sett betjener mange klienter, bygges et MANET opp rundt prinsippet om direkte kontakt mellom klientene. Ved å inneha kjennskap til nettverkstopologien i et definert område rundt seg, kan en node kommunisere denne informasjonen ut til sine naboer som oppdaterer med sin informasjon. Informasjonen viderefremmes til denne nodens naboer igjen og slik vil nettverkets totale topologi til enhver tid være kjent for alle nodene. På denne måten kan en node viderefremme data langs den ruten som er mest egnet for å nå destinasjonen. Nodens avgjørelse om hvor data skal sendes når den er i en reléfunksjon, ivaretas av en protokoll tilpasset disse prinsippene. Den best kjente av disse er Optimized Link State Routing Protocol (OLSR) [2].

Prinsippene om MANET og anvendelsen av denne teknologien til videreutvikling av kommersielle tjenester som mobiltelefonsystemer og datautvekslingssystemer for bedrifter og forbrukere er naturlig nok en sterk drivkraft for forskning og utvikling på denne arenaen. Prinsippene har dog andre anvendelsesmuligheter, og tanken om en videreutvikling av Internett<sup>1</sup> slik vi kjenner det i dag, i retning av en frigjøring fra infrastruktur og på sikt en sammensmelting av egenskapene i mobiltelefonsystemene og trådløse datanettverk er også svært interessante fra et militært synspunkt. Det er derfor ingen overraskelse at denne utviklingen på mange måter drives av de militære forskningsmil-

---

<sup>1</sup>Det strides om hvorvidt Internett skal skrives med liten eller stor forbokstav, men ettersom Defense Advanced Research Project Agency (DARPA) benytter stor forbokstav er denne konvensjonen valgt i rapporten

jøene, på samme måten som var tilfelle for det tidligere DARPA-nettet, utviklet i USA av DARPA, som ga grunnlaget for det senere Internett. Teorien rundt MANET kan meget enkelt sees å være særdeles attraktiv i rammen av militære operasjonelle og taktiske kommunikasjonsnettverk. Militære operasjoner kjennetegnes ofte av å finne sted i miljøer med mer eller mindre frafall av fungerende infrastruktur og av særdeles høy grad av dynamisk nettverkstopologi. Nodene i nettverket har ofte enkelte andre noder innen rekkevidde for seg, som igjen når flere noder og så videre. Nodene vil ofte være i konstant relativ bevegelse og svært ofte vil kommunikasjonen hindres av terreng eller bygninger. I et miljø uten en kommunikasjonsinfrastruktur er det dermed en sterk fordel hvis nodene kan kommunisere direkte med hverandre i en slags kjedetopologi, og således få dataene frem dit de skal via et for operatøren på forhånd ukjent antall noder, og via en tilsvarende ukjent rute.

Et MANET vil kjennetegnes av at noder kommer innenfor nettverkets dekningsområde, opererer i nettet i en viss tid, for så å forsvinne utenfor dekningsområdet igjen. For å bidra til nettets fulle funksjon kreves det av en node som tilslutter seg til nettet at den, relativt raskt, utveksler informasjon med nodene den når direkte. Denne informasjonen inneholder normalt hvilke andre noder den har direkte kontakt med, og hvor god denne forbindelsen er. Det er i hovedsak to ulike tilnærminger til dette. Vi har såkalte reaktive protokoller som får kjennskap til noder i nettverket og oppretter ruter ved behov, og vi har proaktive protokoller hvor nodene i nettverket kontinuerlig opprettholder en kjennskap til andre noder i nettverket og informasjon om beste rute til den enkelte, uavhengig av behov. I en proaktiv protokoll er informasjonsutvekslingen mellom naboer kritisk og må foregå regelmessig. Ettersom en node i det den faller utenfor dekningsområdet av natur ikke evner å melde fra om dette, er det vesentlig at nabonodene detekterer frafallet og oppdaterer sine rutingtabeller. Disse mekanismene ivaretas eksempelvis av den tidligere nevnte OLSR-protokollen, og denne sørger for oppdatering av informasjon om den distribuerte nettverkstopologien, samt tar avgjørelser rundt når kanalen mellom to noder ansees å være for dårlig. Fra dette kan vi så utlede at et nettverk med en stor mengde trafikk mellom relativt mange noder vil tjene på å nytte en proaktiv protokoll, mens et nettverk med relativt mindre trafikk kan tjene på en reaktiv protokoll. Dette har sin begrunnelse i at dersom trafikkbelastningen er stor og hyppig vil mengden av kontrolltrafikk i nettet være unødig stor. Det er da gunstigere at rutingmulighetene er kjent i nettverket til enhver tid. Ved sjeldnere og mindre mengde trafikk blir det unødvendig mye kontrolltrafikk å holde oversikt over hele nettet til enhver tid.

Fra et praktisk og kostnadmessig ståsted vil det i tidlige stadier av utviklingen av et MANET være svært attraktivt å kunne nytte andre miljøer enn det faktiske systemet er tenkt nyttet i. Forskning og utvikling innen kommunikasjonssystemer i det militære domenet innebærer gjerne bruk av plattformer som både er svært omfattende og kostnadskrevede i drift, men som også er bundet opp i militære og politiske gjøremål på en slik måte at det ikke nødvendigvis er kompatibelt med prøver og forsøk slik de tekniske miljøene kan tenke seg. Simulering er derfor en svært naturlig retning å gå. Radiobølgeutbredelse er noe som i en enkel form kan gjenskapes i et laboratorium, men all den tid man ønsker en modellering av virkeligheten, og den innvirkningen hindringer i form

---

av terreng og bygninger har på bølgeutbredelsen, begynner utfordringene å melde seg. Det reelle miljøet systemet skal fungere i kan bestå av alt fra fri sikt kontakt mellom to noder, tilnærmet fritt for refleksjon, avbøying og spredning, til en signalvei uten direkte kontakt, og alt som når frem til mottakeren er et konglomerat av komponenter som er reflektert, avbøyd og spredd. Endringen i signalet vil således kunne finne sted i tids-, frekvens-, og fasedomenet. Gjenskapingen av dette miljøet kan være en utfordring på laboratoriet, spesielt dersom man ønsker å holde kostnadene nede.

Det som dog kan tenkes å være en alternativ løsning er å forsøke å etterligne kun transmisjonsmiljøet selv, slik at mest mulig av kjeden er så nært opptil det reelle som mulig. En mulighet å gjøre dette på er å konstruere en emulator som tar sikte på å gjenskape transmisjonsmediet, og som da kan påvirkes i en eller annen grad for å forsøke å gjenskape deler av virkeligheten. Ved å få til dette kan man oppnå å komme mye lenger i utviklingen av systemet uten å forlate laboratoriet, og dermed redusere kostnadene. Et virkelighetsnært transmisjonsmiljø vil være svært komplisert, hvis i det hele tatt mulig, å gjenskape, så det vil være snakk om en kraftig forenkling. På den annen side får man mulighetene til å fremprovosere elementer som kan tenkes å være svært sjeldne i virkeligheten, samt at det er mulig å monitorere systemets oppførsel bedre, nettopp fordi man vet hvordan transmisjonsmiljøet oppfører seg. Situasjoner kan også gjenskapes eksakt slik at scenarier og hendelser kan gjentas. Ved å nytte reell maskin- og programvare så langt som mulig i kjeden får man også færre feilkilder knyttet til simuleringsmodeller.

Arbeidet med denne hovedoppgaven startet der prosjektoppgaven fra faget TFE 4605, Radioteknikk og Fjernmåling fordypning sluttet. Arbeidet har dreiet seg om topologi-emulatoren for emulering av trådløse kanaler i VHF og UHF båndene, mellom 200 MHz og 3 GHz. Prosjektoppgaven tok for seg en studie og en kalibrering av systemet med en i hovedsak todelt intensjon til grunn. Primært var målsetningen å kalibrere emulatoren, sekundært og implisitt i dette å kartlegge karakteristikk, muligheter og begrensninger ved systemet. Arbeidet endte med en fyldig oversikt over hvordan emulatorens kanaler påvirker hverandre, samt en dokumentasjon for hele systemet. Kanalenes påvirkning på hverandre er kvantisert ved hjelp av et sett med matriser som gjengir det dynamiske området hver kanal kan opereres i for å gjenskape en situasjon der kanalen kan være helt åpen, delvis åpen eller helt lukket. Sistnevnte definerer vi til å være der vi ikke har et Signal-to-Noise ratio (SNR) som er høyt nok til at signalet entydig kan skilles fra støygulvet. I dette området er kanalen uforutsigbar, og vi kan derfor ikke på forhånd vite noe sikkert om utfallet av transmisjonen.

Kapittel 2 i denne rapporten omhandler bakgrunnen for oppgaven, med en definering av denne, begrensninger og en overordnet fremgangsmåte. Den logiske oppbyggingen av emulatorsystemet går gjennom i kapittel 3, og i etterfølgende kapitler en gjennomgang av de enkelte logiske modulene. Kapittel 4 omhandler posisjonsmodulen og kapittel 5 kanalmodulen. Emulatorkontrollmodulen går gjennom i kapittel 6, mens det noe overordnede grensesnittet omhandles i kapittel 7. Avslutningsvis i kapittel 8 følger konklusjonen. En detaljert gjennomgang av maskinvaren som utgjør systemet finnes i rapporten fra den tidligere prosjektoppgaven[8].



# Kapittel 2

## Bakgrunn

### 2.1 Definerings av oppgave

Fra den foregående prosjektoppgaven kommer kjennskap til de dynamiske områdene det er aktuelt å operere i med topologiemulatorene. Det som i det videre er ønskelig er å modellere en bevegelse for de involverte nodene. Kjennskapet til nodenes virtuelle bevegelser bør så nyttes som deler av grunnlaget for å beregne en tilhørende dynamisk demping mellom nodene. Det er også behov for en overordnet styringsmekanisme for emulatorene, slik at systemet totalt sett kan modellere et MANET bestående av inntil seks radionoder med tilstrekkelig grad av realisme.

Dette innebærer at en modell for beregning av tidskontinuerlig kanaldemping må implementeres som grunnlag for beregning av verdiene som skal settes på dempeleddene. Fra arbeidet med prosjektoppgaven tegnet det seg også et behov for å implementere en modell for simulering av nodenes posisjon. Dette vil muliggjøre en enkel visualisering, samt at en overordnet styringsrutine vil gjøre systemets oppbygning av ulike modeller blir transparent for brukeren.

### 2.2 Begrensninger

Ettersom radiokanalen i emulatorene består av en kablet forbindelse mellom nodenes antenneutgang, og eneste mulighet for påvirkning av kanalen er en diskret endring av dennes effektive demping, vil det ikke være aktuelt å ta hensyn til såkalt flerbandedemping (multipath fading). Konklusjonene fra prosjektoppgaven hva gjelder begrensninger knyttet til hvor raskt kanaldempningen kan endres vil bli utforsket noe mer, da videre arbeid har avdekket muligheter for å presse hastigheten ytterligere opp.

Tanken bak utviklingen av emulatorene er at den skal nyttes til evaluering av protokoller og radiosystemer under paraplyen av såkalte MANET's. Det har vært en begrensning fra designen av systemet at det maksimale antallet radionoder er seks, og at disse på grunn av toveis forbindelser i et alle-til-alle mønster utgjør 15 tidsvarierende kanaler. Dette valget ble opprinnelig gjort på bakgrunn av en vurdering av kompleksitet og nytteverdi av systemet. Det viste seg i løpet av designprosessen at når nodeantallet øker forbi seks

blir antallet komponenter i emulatoren høyt, og kompleksitetsgraden kraftig økende ved sammenkobling/konfigurering av disse. På utsiden av selve topologiemulatoren vil det i hver node så være naturlig å koble på en sender/mottaker for radiosignaler, samt utenfor denne igjen en eller annen form for trafikkgenerator. Sistnevnte kan komme fra et vidt spekter av enheter, men det kan tenkes for eksempel en datamaskin som nytter en gitt protokoll, fortrinnsvis en for ruting i et ad hoc nettverk, til sitt kommunikasjonsbehov. Denne datamaskinen kan være koblet direkte til den radiosender/-mottakeren. Sammen med flere noder med tilsvarende prinsipiell oppbygging, også koblet til topologiemulatoren, kan disse utgjøre en laboratorium-testbenk for en gitt systemkonfigurasjon. Som en direkte konsekvens av dette fokuseres det i det videre mot fysisk lag, og det som finner sted i lagene over dette<sup>1</sup> faller utenfor målsetningen for arbeidet oppgaven.

## 2.3 Fremgangsmåte

Innledningsvis vil det bli sett på ulike mobilitetsmodeller for simulering av nodeposisjon, samt for å ha kvalifiserte inngangsverdier til kanalmodellen. Deretter vil det bli sett på muligheter for implementering av kanalmodeller, og hvordan dette kan gjøres på en mest mulig hensiktsmessig måte. Dette defineres som det primære arbeidet, og er det som vil dimensjonere resterende innsats. Parallelt med dette vil det utvikles en grafisk styringsrutine for det hele, som skal gjøre systemet brukervennlig. All programmering foretas i LabVIEW, og maskinvaren som nyttes til styring av systemet må ha LabVIEW installert. Det har blitt utviklet LabVIEW rutiner som muliggjør evaluering av kanalene over en Rohde og Schwarz ZVx nettverksanalysator. For bruk av disse må LabVIEW-installasjonen komplementeres med drivere for disse analysatorene; siste versjon fåes i `rszv_1v61_4_3_3.zip`. Denne pakken er fritt tilgjengelig via Rohde & Schwarz's nettsider <sup>2</sup>

---

<sup>1</sup>Referert til eksempelvis OSI-modellen

<sup>2</sup><http://www.rohde-schwarz.com>

## Kapittel 3

# Logisk oppbygging av emulatorsystemet

### 3.1 Arven

#### 3.1.1 Maskinvare

Maskinvaren som utgjør hoveddelen av topologiemulatorsystemet; selve emulatoren, har ikke vært gjenstand for noe arbeid i forbindelse med denne oppgaven. Fysisk består systemet[8] av selve emulatoren og en PC for kontroll av systemet. I tillegg til disse komponentene kommer kabling for å knytte det hele sammen. Signalgenerering og måling utføres med samme apparat, en nettverksanalysator av typen ZVC fra Rohde & Schwarz[5].

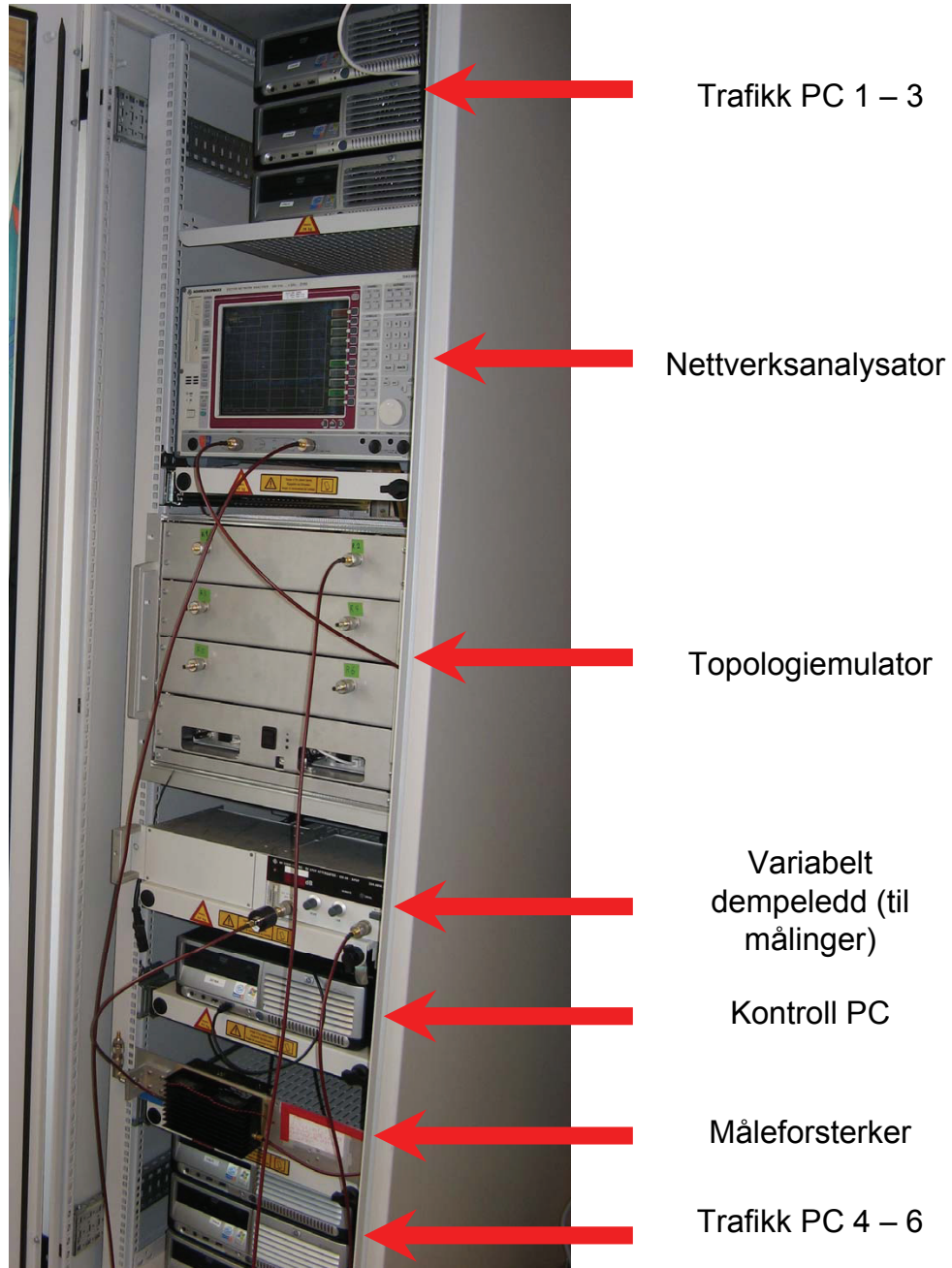
#### 3.1.2 Programvare

Dette er området hvor arbeidet har blitt utført. Kontroll-PC'en kjører National Instrument's LabVIEW programpakke, og denne programvaren nyttes til både styring av selve systemet og til målinger.

Det eksisterende systemet besto ved starten av arbeidet av en enkel styringslogikk som muliggjorde setting av statiske, ukalibrerte verdier for kanaldemping. Dette fungerer slik at ønsket verdi på hver av systemets 15 kanaler legges inn i en  $[15 \times 1]$ -matrise, hvoretter verdiene settes på kanalene. For at systemet skal være egnet til den bruken som er ønskelig, må en større grad av dynamikk inn i styringslogikken. Samtidig er det ønskelig at verdiene som settes, og endringen av disse, har en viss relasjon til virkeligheten slik at systemet til en viss grad kan nyttes til semirealistisk<sup>1</sup> testing av maskin- og programvare.

---

<sup>1</sup>Som beskrevet i [8] er det begrensninger av en slik art i systemet at fullstendig realisme ikke er mulig å skape i dette miljøet



Figur 3.1: Systemet montert i rack





Figur 3.2: Topologiemulatorens forside

## 3.2 Modulinndeling

Det er en klar målsetning at systemet bygges opp mest mulig modulært slik at senere modifisering og vedlikehold blir mest mulig intuitiv og forståelig. Samtidig vil det være slik at systemets bruksområde spenner relativt vidt, og det er derfor avgjørende å kunne gå inn å gjøre applikasjonsspesifikke endringer, kun kjøre deler av systemet og så videre. Derfor velges en meget modulær tilnærming hvor det søkes å sette sammen styringslogikk av hensiktsmessig sammensatte moduler.

Systemet er bygget opp rundt en iterativ tankegang, bestående av to primære programløkker. Innenfor disse finnes et antall logiske moduler, hver bestående av en eller flere filer. De primære programløkkene kan fremstå som noe skjevt satt sammen når det gjelder mengden innhold, men dette kommer av kjøretidsmessige betraktninger.

Programløkke 1 ivaretar alle beregninger og simuleringer, mens programløkke 2 foretar settingen av kanalverdiene på kanalene. Inndelingen i moduler i hver av løkkene har vært gjenstand for mange endringer, men tilslutt har en inndeling som den i punkt 3.2.1 og 3.2.2 blitt valgt. I tillegg til disse kommer modulen *Grensesnitt* som ikke faller inn under noen av løkkene men ligger overordnet disse som en hovedrutine og står for kontrollen av hele systemet.

### 3.2.1 Programløkke 1

Består av følgende moduler:

**Posisjonsmodul**

**Avstandsmodul**

**Tegnemodul**

**Kanalmodul**

### 3.2.2 Programløkke 2

Består av:

**Emulatorkontrollmodul**

### 3.2.3 Posisjonsmodul

Denne modulen har som oppgave å ta vare på nodenes posisjoner samt deres hastighet. I tillegg forestår modulen den kontinuerlige oppdateringen av posisjonene etter den mobilitetsmodellen som er valgt. Systemet støtter to ulike mobilitetsmodeller. Posisjonsmodulen gjennomgås nærmere i kapittel 4.

### 3.2.4 Avstandsmodul

Avstandsmodulen tar utgangspunkt i en graf den får tilsendt i matriseform fra posisjonsmodulen. Grafen representerer nodenes posisjon i et koordinatsystem<sup>2</sup>. Basert på nodenes posisjoner beregner avstandsmodulen innbyrdes avstand mellom samtlige noder, og presenterer dette i en [15x1]-matrise. Dette innebærer at vi har 15 aktuelle avstander. Modulen er integrert i posisjonsmodulen og den behandles derfor nærmere i kapittel 4.

### 3.2.5 Tegnemodul

For at brukeren skal få et lett forståelig innblikk i hva som foregår og hvordan nodene beveger seg, utfører denne modulen en plotting av den grafen posisjonsmodulen produserer i et gitt øyeblikk. Plottet presenteres i systemets kontrollgrensesnitt. Tegnemodulen er så liten i omfang at det ikke er hensiktsmessig å ha den som en selvstendig modul i systemet. Den inngår derfor som en del av posisjonsmodulen og omtales således nærmere i kapittel 4.

### 3.2.6 Kanalmodul

Her ligger den langt tyngste logikken i systemet, og det er med denne modulen hovedtyngden av arbeidet har foregått. Kanalmodulen skal, basert på innbyrdes avstand mellom nodene, beregne en serie verdier for en gitt tidsperiode for hver kanal. I denne sammenhengen forstås kanal som forbindelse mellom nodene. Modulen tar utgangspunkt i avstanden mellom to noder, utfører beregninger og operasjoner slik at resultatet blir en tidsdomene serie av varierende dempeleddsverdier for kanalen. Denne serien har et forhåndsdefinert antall verdier, og gir grunnlaget for en direkte manipulasjon av kanaldempningen basert på disse verdiene direkte. Verdiene er i desibel (dB)<sup>3</sup>-format. Mer om denne modulen i kapittel 5.

### 3.2.7 Emulatorkontrollmodul

Dette er hvor selve manipulasjonen av kanalene foregår. Modulen får tidsdomeneserier for kanaldempning fra kanalmodulen. Den delen av arven som går på statisk styring av emulatoren gjenfinnes i denne modulen. Ved hjelp av en løkkestruktur settes på en iterativ måte første element fra tidsdomeneserien på de respektive kanalene, deretter andre element og så videre. Forsinkelsen mellom hvert element kan endres, slik at endringshastigheten på kanalene kan manipuleres. Modulen beskrives nærmere i kapittel 6

### 3.2.8 Grensesnitt

Grensesnittet er ikke like klart definert som en modul i programvaren som de ovennevnte modulene er. Den er vinduet brukeren får inn i systemet ved normal bruk, og kommu-

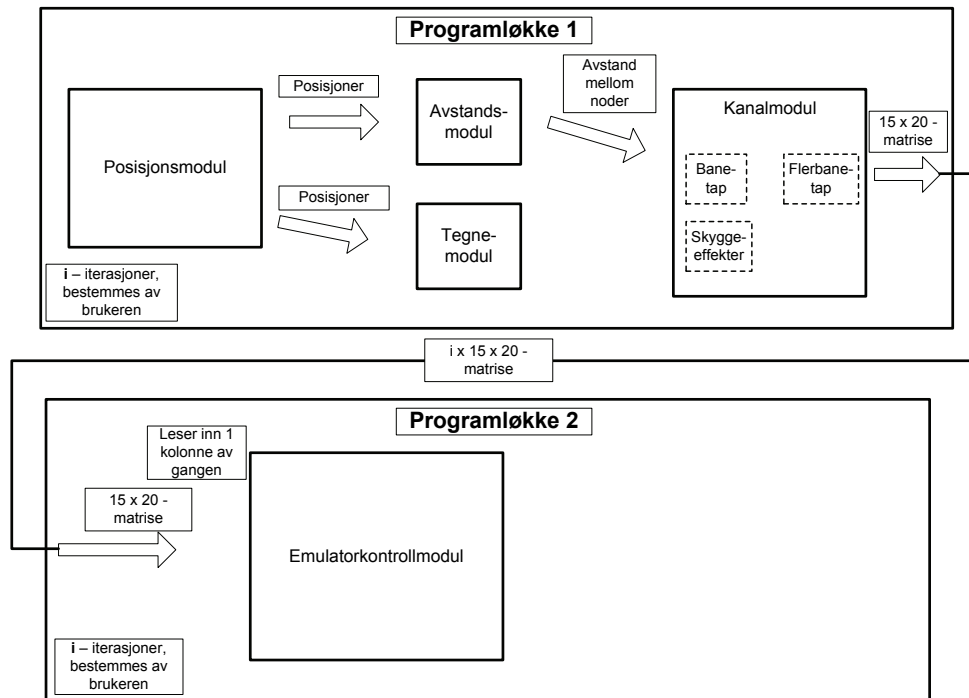
---

<sup>2</sup>Koordinatsystemet er kartesisk for den ene mobilitetsmodellen, og polart for den andre

<sup>3</sup>dB er definert som  $X_{dB} = 10 \log \frac{X}{X_0}$

niserer med de ulike modulene etter behov. Samtidig som den gir brukeren mulighet til å sette parametre og lese ut tilstander og situasjon forestår grensesnittmodulen den overordnede styringen av systemet ved å benytte seg av de andre modulene. Dermed kan man si at denne modulen er mer en logisk modul basert på en programrutine enn den er en faktisk modul. Grensesnittets oppbygging beskrives nærmere i kapittel 7

Sammenhengen mellom de ulike modulene slik de opererer i emulatorssystemet fremgår av figur 3.3



Figur 3.3: Logisk oppbygging av systemet

## Kapittel 4

# Posisjonsmodul

### 4.1 Behovet for en posisjonsmodul

Topologiemulatorene har mulighet for tilkobling av seks radioenheter, og systemet er utviklet for å være en testplattform for en dataoverføringskjede på et laboratorium. Radiomateriellet som er tenkt brukt er primært militære radiostasjoner som opererer i UHF-båndet. Dette innebærer at radioenhetenes faktiske posisjon inne på laboratoriet ikke har noen praktisk betydning når scenarier skal simuleres. Det vi dermed må gjøre er å posisjonere nodeobjekter på et virtuelt kart med et sentralt referansesystem. Som jeg vil komme tilbake til i kapittel 5 definerer vi detaljbehovet slik at vi ikke har behov for å vite noe om terrengets beskaffenhet. Det interessante er i første omgang noderenes posisjon til enhver tid.

For at systemet i sin helhet skal få en bruksmessig verdi, må nodene kunne bevege seg på det virtuelle kartet vi opererer på. Dette medfører at vi må ha en algoritme for endring av posisjonene per gitt tidsenhet. Her kan vi samtidig ta høyde for en hastighet som den enkelte noden til enhver tid holder.

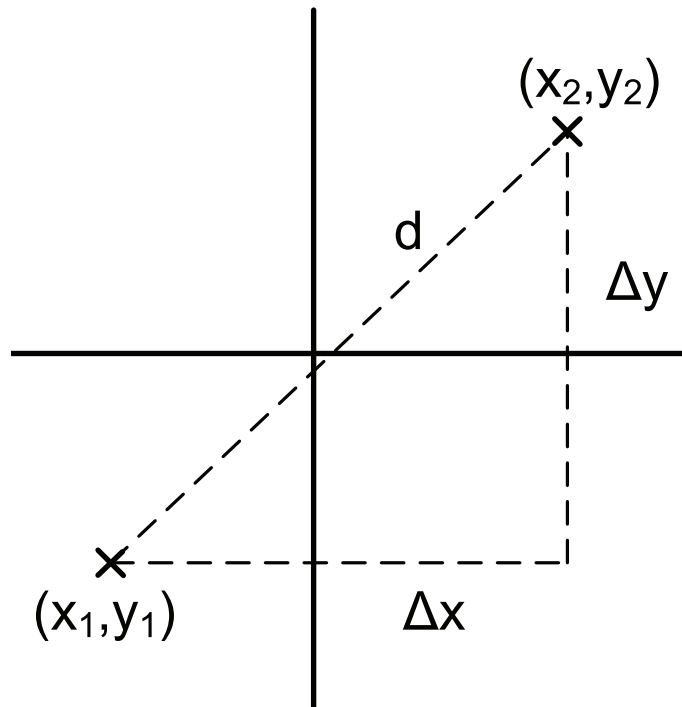
Oppgaven med å opprette og manipulere oversikten over noderenes posisjoner og oppførsel ligger fysisk hos kontrollmaskinen som styrer hele systemet.

### 4.2 Virtuelt kart

For å enklere få et intuitivt bilde av hvordan posisjonering og bevegelse gjøres i emulatorsystemet kan man tenke seg et sterkt forenklet kart, der det eneste karttemaet er referansesystemet. Det virtuelle kartet som implementeres har således form som et to-dimensjonalt kartesisk koordinatsystem, og nodene vil dermed ha entydige posisjoner på formatet  $(x, y)$ . Det er i tillegg til dette ønskelig å vite noe om den enkelte nodes momentane hastighet, og dette gjøres ved å lagre en hastighetsvariabel sammen med posisjonsvariablene. Etersom vi hverken opererer med noe terrengbilde eller terrengdetaljer i det hele tatt i denne sammenhengen, forenkles det hele til å være en datastruktur i form av en  $[6 \times 3]$ -matrise. Vi har seks noder, og for hver av disse har vi en  $x$ -posisjon, en  $y$ -posisjon og en hastighet i meter per sekund.

### 4.3 Bevegelse og posisjon

Så langt har vi altså en datastruktur som inneholder alt vi trenger å vite om den enkelte nodes oppførsel posisjonsmessig. Men, så langt er systemet helt statisk. For å få et element av dynamikk inn i det hele må vi ha muligheten til å kontinuerlig endre nodenes posisjon, samt beregne avstanden nodene mellom. Det siste gjøres ved en enkel geometrisk beregning. Ettersom alle nodene er i vårt todimensjonale kartesiske system kan vi simpelthen basere oss på Pythagoras' prinsipp som vist i figur 4.1



Figur 4.1: Avstand

Avstanden  $d$  i figur 4.1 kan beregnes på følgende måte:

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2} \quad (4.1)$$

Når det gjelder endring av nodenes posisjoner finnes det et stort antall mobilitetsmodeller for simulering av bevegelse. Utfordringen er at nodene bør oppføre seg tilnærmet virkelighetsnært, noe som ekskluderer de aller enkleste mobilitetsmodellene. Samtidig er det ønskelig å holde kompleksiteten på et rimelig nivå. I det følgende kommer eksempler på ulike mobilitetsmodeller.

### 4.4 Mobilitetsmodeller

Tradisjonelt sett kan mobilitetsmodeller deles inn i følgende hovedgrupper[2],[14]:

**Individuelle modeller** Dette er modeller hvor hver enkelt node beveger seg individuelt

**Gruppemodeller** I gruppemodellene beveger nodene seg i forhold til sin gruppe

#### 4.4.1 Individuelle modeller

I denne kategorien finner vi de mobilitetsmodellene hvor den enkelte node beveger seg individuelt, ut fra ulike typer teorier og mekanismer.

##### Tilfeldig gange (Random Walk)

Denne modellen omtales gjerne også som Brownske bevegelser. Her er nodens endring i hastighet og retning ved hver iterasjon tilfeldig, gjerne basert på en variabel med en uniform fordeling.

##### Tilfeldig rutepunkt (Random Waypoint)

Her har vi et antall rutepunkt, uniformt fordelt over et gitt areal. En node beveger seg i en rett linje mot et rutepunkt, med en gitt, konstant hastighet. Ved ankomst til rutepunktet velges et nytt tilfeldig rutepunkt, og noden beveger seg i retning dette nye punktet.

**Tilfeldig rutepunkt på randen (Random Waypoint on the border)** Dette er en avart av Tilfeldig rutepunkt-modellen, og forskjellen er at i stedet for at rutepunktene er uniformt fordelt over det gitte området er de nå fordelt på kanten av området. Dette innebærer at senter for sannsynlighetsfordelingen av alle punktene ikke lenger er i sentrum av området, men langs randen. Dermed kan bevegelser i enkelte tilfeller kun følge randen.

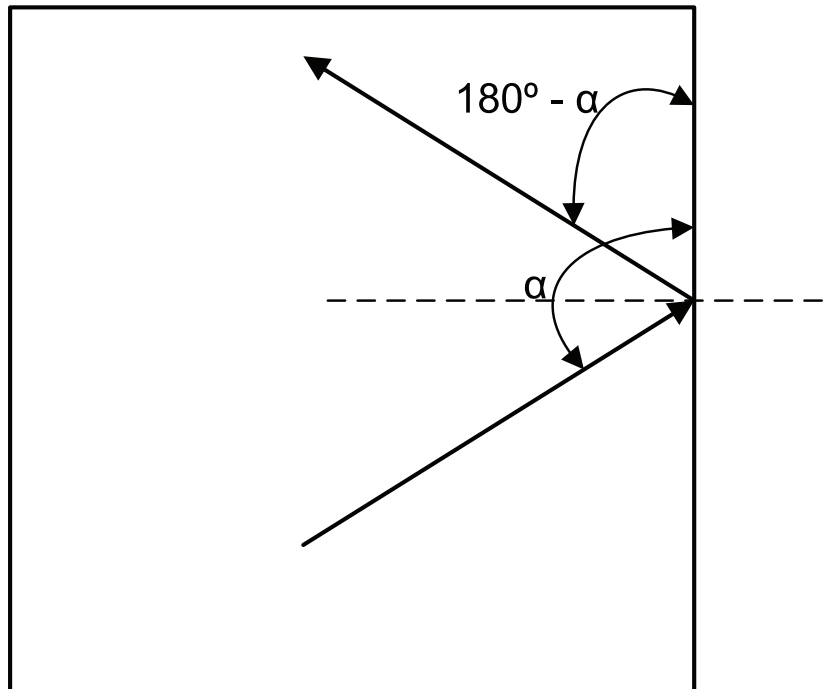
**Markovisk rutepunkt (Markovian Waypoint Model)** Her introduserer man en grad av minne i systemet slik at valget av neste rutepunkt til en gitt grad avhenger av forrige rutepunkt.

**Andre variasjoner** Andre alternativer over denne modellen kan være å la hastigheten på neste strekning avhenge av forrige streknings hastighet. Dette vil på samme måte som introduksjonen av et minne i forbindelse med destinasjonsvalg ”demppe” tilfeldigheten i systemet slik at de momentane endringene ikke blir så store.

##### Tilfeldig retning (Random Direction)

Hver node velger en tilfeldig retning og beveger seg i denne retningen med en gitt konstant hastighet[12]. Når noden støter på randen av området tar den en pause av en gitt lengde før den fortsetter basert på én av følgende to mulige hendelser;

**Refleksjon** Det ene som kan skje er at noden reflekteres i randen, med utgangsvinkel relativ til randen lik  $180^\circ - \alpha$  (*innfallsvinkel*) relativ til samme referansesystem (figur 4.2).



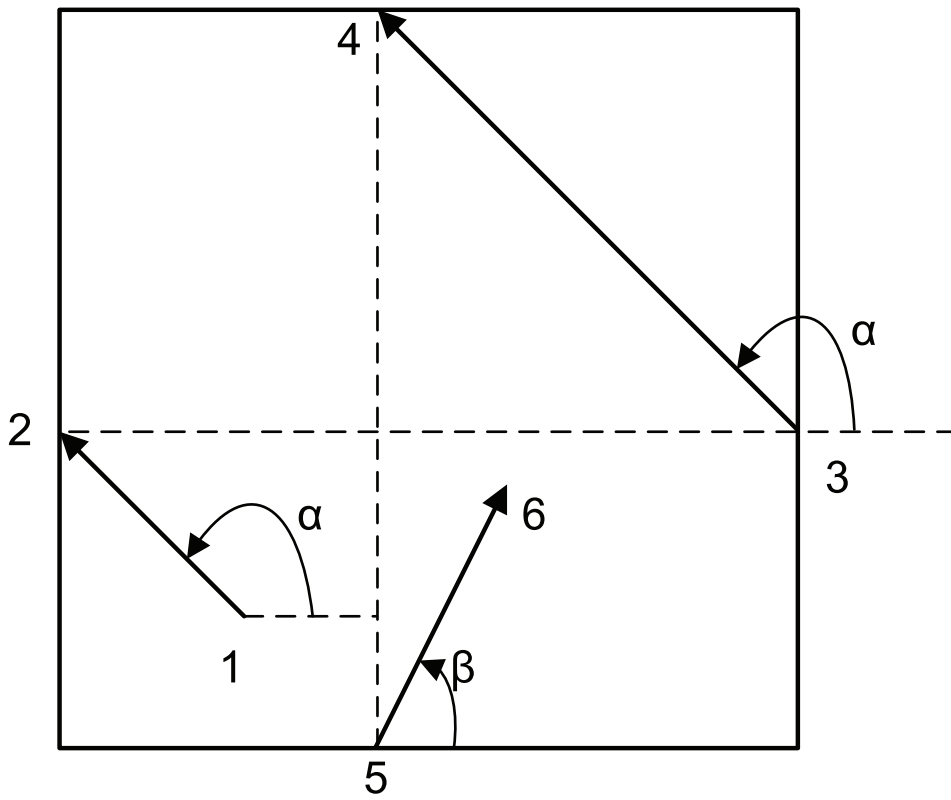
Figur 4.2: Refleksjon

**Rundhopp** Alternativt utfører noden et rundhopp hvor den momentant beveger seg til tilsvarende posisjon på motstående rand, og fortsetter sin bevegelse for resterende tid av inneværende periode (figur 4.3). I denne situasjonen tar noden ofte en pause på randen, samtidig som retning- og hastighetsverdiene oppdateres. I figur 4.3 skjer dette i posisjon 4.

#### Randløst område (Boundless Simulation Area)

Her finnes ingen rand i simuleringsarealet, som har form som en torus. Nodenes retning og hastighet oppdateres i tilfeldige intervaller og hentes fra utvalg med gitte maksimalverdier for retningsendring og akselerasjon. Nodene tar ingen pauser og er i konstant bevegelse.





Figur 4.3: Rundhopp, med oppdatering av hastighet og retning i punkt 4.

### Gauss-Markov

Denne modellen[3] har en minnefunksjon for både retning og hastighet. Nodene har en initiell retning og hastighet, som oppdateres ved hvert tidssteg. Ved beregningen av nye verdier tas det hensyn til nåværende verdier, middelverdier for retning og hastighet, samt en tilfeldig variabel trukket fra en Gaussisk fordeling. Parametrene vektet i henhold til en tilfeldighetsparameter  $\alpha$ . Oppdatert hastighet  $s_n$  og retning  $d_n$  beregnes på bakgrunn av tidligere verdier,  $\alpha$ , middelverdier for hastighet og retning  $\bar{s}$  og  $\bar{d}$ , og en Gaussisk fordelt variabel  $s_{x_{n-1}}$  og  $d_{x_{n-1}}$ :

$$s_n = \alpha s_{n-1} + (1 - \alpha)\bar{s} + \sqrt{1 - \alpha^2} s_{x_{n-1}} \quad (4.2)$$

$$d_n = \alpha d_{n-1} + (1 - \alpha)\bar{d} + \sqrt{1 - \alpha^2} d_{x_{n-1}} \quad (4.3)$$

### Bymiljø (City Section)

Nodene beveger seg langs forhåndsdefinerte stier, etter en korteste vei algoritme. Det kan legges begrensninger på hastighet og minimum distanse mellom nodene. Nodene kan også instrueres til å ta pauser i en destinasjon før fortsettelse.

#### 4.4.2 Gruppemodeller

Felles for alle gruppemodeller er at en deltagende node beveger seg relativt til en referanse i en gruppe den tilhører. Fra et militært synspunkt er dette generelt sett de mest interessante, da en node i militær sammenheng som oftest beveger seg i henhold til enkelte overordnede prinsipper, og innen visse rammer[1].

### Referansepunkt (Reference Point)

Dette er den mest generiske av gruppemodellene. Her opererer nodene i grupper, og den enkelte node i en gruppe følger gruppens definerte referansepunkt. Hvor strengt enkelt-nodene følger referansepunktet bestemmes ved hjelp av parametre. Referansepunktet kan for eksempel bevege seg etter en av de individuelle modellene omtalt tidligere.

### Eksponentielt korrelert (Exponential Correlated)

Innsikt i denne modellen er ikke trivielt, og den baserer seg på å beregne en bevegelsesvektor  $\vec{b}(t)$ . Gitt en posisjon i tidspunkt  $t$  enten for en node eller en gruppe, nyttes  $\vec{b}(t)$  til å beregne nodens eller gruppens posisjon i  $t + 1$ . Denne posisjonen er  $b(\vec{t} + 1)$ . Vektoren er i det komplekse plan og uttrykkes ved:

$$b(\vec{t} + 1) = b(\vec{t})e^{-\frac{1}{\tau}} + (\sigma\sqrt{1 - e^{-\frac{2}{\tau}}})r \quad (4.4)$$

Her er  $r$  en tilfeldig variabel fra en Gaussisk fordeling med varians  $\sigma$ .  $\tau$  definerer endringshastigheten på en slik måte at desto mindre verdi for  $\tau$ , desto hurtigere endring.

En stor utfordring med denne modellen er bestemmelse av parametrene for å oppnå ønskede effekter.

### **Nomadisk samfunn (Nomadic Community)**

Dette fremstår som en kombinasjon av modellene Tilfeldig gange og Tilfeldig rutepunkt. Nodene opererer i grupper og gruppen i sin helhet beveger seg fra punkt til punkt etter modellen om Tilfeldig rutepunkt. Inne i hvert enkelt slikt rutepunkt beveger de enkelte nodene seg etter modellen om Tilfeldig gange, men skalaen for disse bevegelsene er betydelig mindre slik at bevegelsene skjer inne i rutepunktet.

### **Forfølgelse (Pursue)**

Her følger et antall enkeltnoder simpelthen etter en hovednode som opererer i henhold til sin egen mobilitetsmodell. Denne modellen er gjerne en av de tidligere nevnte. Enkeltnodene har gjerne en akselererende bevegelse i retning av denne hovednoden, med et bidrag fra en tilfeldig vektor.

### **Kolonne (Column)**

Kolonnemodellen går ut på å modellere en rekke noder som beveger seg i en gitt retning. Eksempel på bevegelse kan være en søkestruktur slik man praktiserer ved søk i sneskred eller lignende. Bevegelsesretningen må ikke nødvendigvis være normalt på linjen nodene ligger på, alle retninger er mulig slik at nodene kan bevege seg på en rekke, en kile og så videre.

## **4.5 Valg av modeller**

Valget av modell for implementering ble basert på et ønske om en overkommelig grad av kompleksitet samtidig som bevegelsene må være relativt realistiske. Studier av de ulike modellene og implementasjoner av disse viser at i denne sammenhengen er Gauss-Markov-modellen vel egnet. Den primære fordel med denne modellen er muligheten til å relativt enkelt endre få parametre og derved endre hele bevegelsesmønsteret.

Det kan dog tenkes at man ved å nytte en semirealistisk mobilitetsmodell ikke på en enkel måte kan gjenta en spesifikk transisjon fra åpen kanal via delvis åpen til lukket kanal. På bakgrunn av dette er det valgt å implementere muligheten for en stjerneformet bevegelse i tillegg til Gauss-Markov-modellen. Stjernemodellen muliggjør repetitive bevegelser slik at bestemte effekter kan studeres.

### **4.5.1 Gauss-Markov-modellen**

Modellen i seg selv er redegjort for i punkt 4.4.1. De vesentlige parametrene i modellen er middelverdiene for hastighet og retning  $\bar{s}$  og  $\bar{d}$  og tilfeldighetsparameteren  $\alpha$ .  $\bar{s}$  er

konstant gjennom hele scenariet, men den til enhver tid gjeldende  $\bar{d}$  velges fra forhåndsdefinert mengde. Ved start av scenariet er middelveidien for retning  $0^\circ(360^\circ)$ . Dette innebærer at nodenes hovedbevegelse går i denne retningen, med variasjoner. Når noden nærmer seg kanten av simuleringsområdet, endres  $\bar{d}$  slik at noden holder seg innenfor området. Dette gjøres ved at  $\bar{d}$  endres avhengig av hvilket grensetilfelle som inntreffer. Dette er gjort ved å dele inn grenseområdet i åtte grensebokser i henhold til figur 4.4 med tilhørende retningsvektorer. Uavhengig av hvilken verdi  $\bar{d}$  har, vil den i det noden kommer inn i én av disse grenseboksene få en ny verdi slik hver grenseboks' vektor indikerer. Dette ivaretas ved at systemet ved start av et scenario regner ut verdier for størrelsen på grenseboksene basert på en innparameter for dette. Størrelsen på simuleringsområdet beregnes ut fra største nodehastighet og en erfaringsmessig god parameter. Denne sistnevnte parameteren er et antall prosent, og systemet tolker verdien som at retningsendring skal påbegynnes ved nådd prosent ut mot verdens ende. Systemet beregner en absoluttverdi i x- og y-retning, og i det en eller begge absoluttverdier av posisjonen i x- og y-retning overskrides oppdateres  $\bar{d}$  med dertilhørende verdi som vist i figur 4.4.

#### 4.5.2 Stjerneformet bevegelse

Tanken bak denne modellen springer ut fra målsetningen om å blant annet kunne se nærmere på rutingprotokollers egenskaper og ytelser under gjentakende forhold. Det anses i denne sammenhengen fordelaktig å kunne bevege nodene slik at en kanal går fra være helt åpen, via delvis åpen og til lukket repetitivt. Dette vil muliggjøre sammenligninger, samt å se hvordan protokollene behandler samme situasjon over tid.

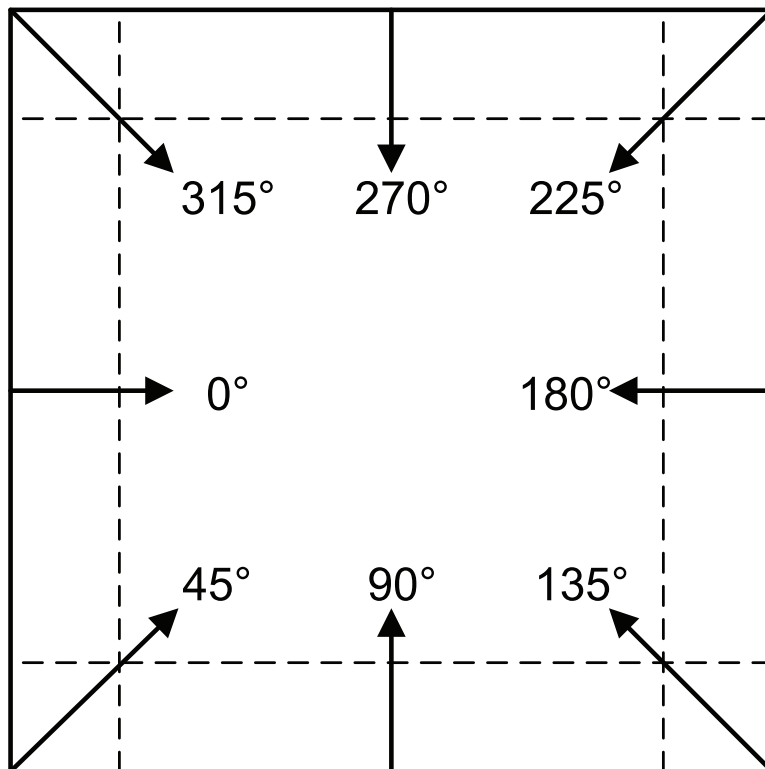
En stjerneformet mobilitetsmodell er implementert ved å konstruere et polart koordinatsystem liggende på det eksisterende kartesiske. Antall deltagende noder betraktes, og den enkelte node gis en retningsvektor ut fra følgende sammenheng der  $i$  er aktuell node,  $\vec{i}_r$  er node  $i$ 's retningsvektor i radianer og  $n$  er antall deltagende noder:

$$\vec{i}_r = \frac{2\pi}{n}i \quad (4.5)$$

Dette innebærer at nodene fordeles symmetrisk over sirkelen, og de beveger seg i en rett linje ut fra origo. Hastighet defineres for hver enkelt node, slik at ulike effekter kan simuleres. Brukeren definerer i denne modellen også en størrelse på simuleringsområdet, og i denne modellen behandles grensetilfellene slik at nodene ved ankomst til grensen snur  $180^\circ$  og beveger seg mot origo. Ved ankomst til origo nyttes igjen den utregnede retningsvektoren, og nodene beveger seg ut igjen. Dette bevegelsesmønsteret gjentas så lenge scenariet kjører. Her vil man kunne konstruere et scenarie for å se på spesifikke effekter, og scenariet kan gjentas repetitivt.

### 4.6 Avstandsmodul

Avstandsmodulen får inn en posisjonsmatrise fra mobilitetsmodellen. Denne har form som en  $[6 \times 2]$ -matrise, der hver node har en x- og y-posisjon. På bakgrunn av dette



Figur 4.4: Retningsendring ved grensetilfeller

beregnes en  $[15 \times 1]$ -matrise, med innbyrdes avstander mellom alle nodene. Utrekning av avstandene gjøres ved hjelp av metoden skissert i punkt 4.3.

Avstandsutregningen er nøkkelen til videre beregninger i systemet. Kanalmodulen omtales nærmere i kapittel 5, og som vi skal se der starter simuleringen av en forbindelse mellom to noder med kjennskap til deres innbyrdes avstand.

## 4.7 Tegnemodul

Dette er en modul som ikke er kritisk for systemet, men som sørger for at brukeren kan tolke et pågående scenarie basert på mer enn rene tall. Modulen får inn posisjonsmatrisen før den sendes til avstandsmodulen, og parallelt med beregningen av sistnevnte plotter tegnemodulen nodedes posisjon i et xy-koordinatsystem. Plottingen skjer for hver iterasjon og med historikk for den enkelte node, slik at man for alle noder til enhver tid har en grafisk presentasjon av alle tidligere, og gjeldene, posisjoner.

# Kapittel 5

## Kanalmodul

### 5.1 Innledning

Topologiemulatorene er tenkt brukt som en emulator for fysisk lag for transmisjon mellom mobile enheter, og frekvensområdet som er valgt for emulatorene er UHF-båndet ( $\sim 200$  MHz - 3 GHz). Målsetningen med å studere og implementere en form for kanalmodell i styringslogikken til emulatorene er å oppnå en tilfredsstillende grad av realisme i emulatorens oppførsel. Det primære virkemiddelet i denne sammenhengen er å variere dempingen på hver kanal over tid. Det søkes derfor å finne frem til en modell hvor et antall punkter i en todimensjonal<sup>1</sup> mengde kan genereres på bakgrunn av definerte parametre. Disse punktene skal så skaleres i begge dimensjoner til de representerer variasjonene i kanaldempning for en gitt kanal i en gitt tidsperiode. Når dette er utført på en hensiktsmessig måte skal de mates videre gjennom systemet for å nyttes som inngangsverdier til styringen av emulatorene som omtales nærmere i kapittel 6.

Hensikten med kanalmodellimplementasjonen i sin helhet er dermed å kunne generere en serie dempeverdier for hver kanal for en gitt tidsperiode, og det er en forutsetning at disse verdiene har en så høy grad av realisme som mulig ut fra det simulerte scenariet. Det vi har av inngangsverdier til kanalmodellen er det som kommer fra posisjonsmodulen, omtalt i kapittel 4. Dette begrenser seg til nodenes posisjoner samt deres momentane hastighet og retning. Det virtuelle terrenget nodene opererer i er tomt for gjenstander og legemer av noen art, slik at vi forholder oss til et fullstendig generisk system, i et sterkt forenklet miljø. Derfor må det etterstrebtes å kunne produsere egne verdier for kanaldempningene, uavhengig av terrengbildet, og det er nettopp denne prosedyren kanalmodulen står for.

### 5.2 Frittroms modellen

I et idealisert miljø totalt fritt for fremmedlegemer, og med en sender og en mottaker med null utstrekning, ville det mottatte signalet på ethvert punkt være lik det utsendte

---

<sup>1</sup>Tid og kanaldempning

signalet korrigert med en multiplikativ faktor,  $L_s(d)$ [15].

$$L_s(d) = \left(\frac{4\pi d}{\lambda}\right)^2 \quad (5.1)$$

Eller uttrykt som forholdet mellom utsendt og mottatt effekt;

$$\frac{P_r}{P_t} = \left(\frac{\lambda}{4\pi d}\right)^2 \quad (5.2)$$

Her er antennevinning hos både sender og mottaker forutsatt lik 1.

Vi merker oss at signalstyrken avtar med kvadratet av avstanden mellom sender og mottaker, eller 20 dB per dekad. Denne modellen for propagasjon kalles *frittroms modellen* (free space loss). Dette er et tildels svært urealistisk scenario, og signalene vil svært sjelden oppføre seg slik. Det vil ofte være noen legemer eller hindringer involvert, for eksempel jordoverflaten. Topologiemulatoren er tenkt nyttet først og fremst knyttet til bakkebaserte systemer, og dermed er neste trinn på stigen mot en mer realistisk modell nettopp det å ta inn jordoverflaten og hindringer i systemet.

### 5.3 Hensyn til hindringer

Det er tradisjonelt sett tre effekter det tas hensyn til når det er interessant å gjøre propagasjonsmiljøet mer komplekst og dermed mer realistisk. Disse bidrar hver på sin måte til ulike effekter på signalet, som vi skal komme tilbake til. Effektene omtales gjerne som refleksjon, avbøying og spredning.

**Refleksjon (Reflection)** Ved refleksjon[4] treffer bølgefronten en overflate som har mye større utstrekning enn signalets bølgelengde  $\lambda$ , og en retningsendring finner sted, som illustrert i figur 5.1

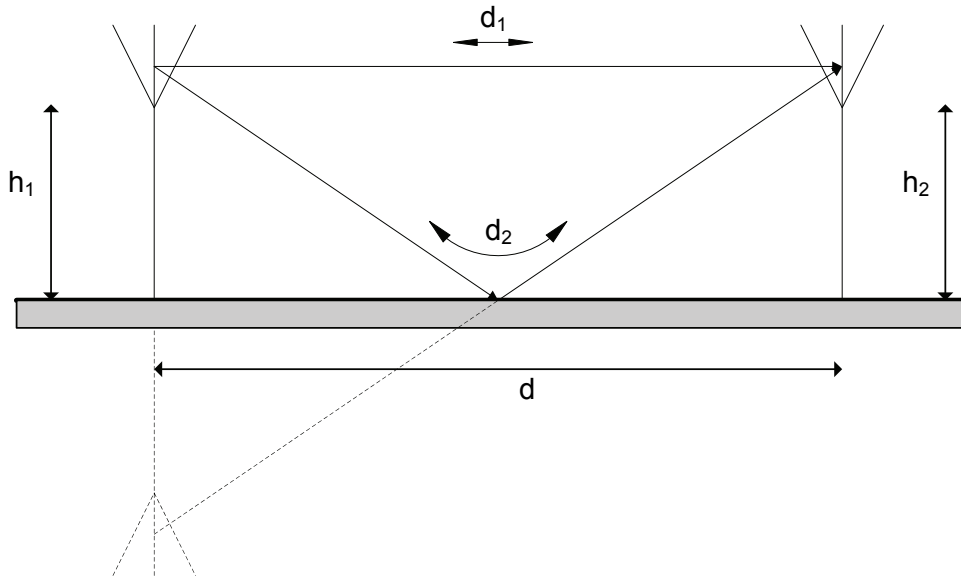
**Avbøying (Diffraction)** Avbøying innebærer at bølgefronten treffer en hindring av større dimensjoner enn bølgelengden  $\lambda$ , men med endelig utstrekning slik at sekundære bølger dannes bak hindringen. Dette kalles også skyggevirking (shadowing) ettersom de sekundære bølgene fører til at en mottaker bak hindringen, uten direkte sikt til senderen, kan motta signalet. Dette vises i figur 5.2

**Spredning (Scattering)** Ved spredning støter hele eller deler av bølgefronten på en hindring med dimensjoner like som, eller mindre enn, sin bølgelengde  $\lambda$ . Her blir signalenergien spredt ut eller reflektert i alle retninger. Se figur 5.3

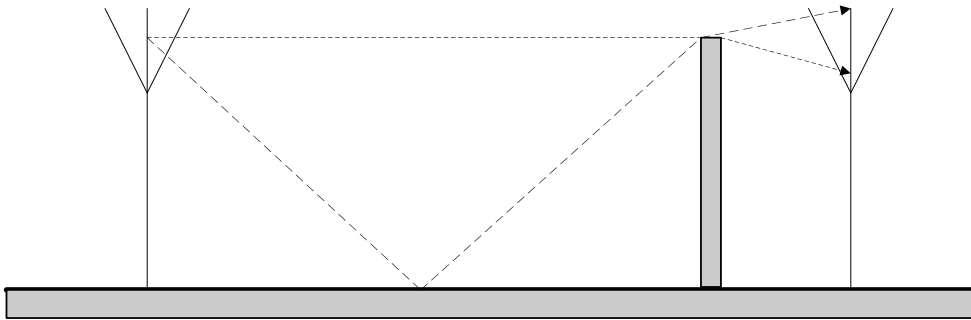
### 5.4 Flat-jord modellen

Dersom vi så tar utgangspunkt i frittromsmodellen, og så diskretiserer det utsendte signalet slik at vi forholder oss til tynne stråler som sendes ut, kan vi tenke oss at en stråle belyser mottakeren i form av en rett linje mellom de to enhetene. Denne strålen

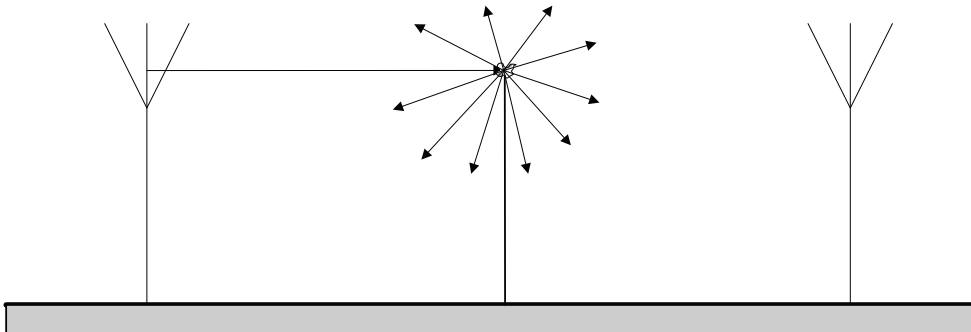




Figur 5.1: Skisse av refleksjon og to-stråle prinsippet



Figur 5.2: Prinsippskisse for modellering av avbøying



Figur 5.3: Prinsippskisse for modellering av spredning

lar vi representere alle stråler som går direkte fra sender til mottaker, i en tilnærmet rett linje. I tillegg til disse vil vi ha et antall stråler som treffer bakken og reflekteres slik at de treffer mottakeren etter denne refleksjonen. Disse kan vi representere ved en stråle som vi antar utsendt fra et speilbilde av senderen, speilet ned under bakkenivå. Trekker vi en direktelinje fra denne speilede senderen og direkte til mottakeren krysser denne bakkenivået på et gitt punkt. Strålen fra den fysiske senderantennen, via dette punktet på bakken og frem til mottakerantennen er strålen som representerer alle reflekterte komponenter som vist i figur 5.1.

Det kan vises[13] at forholdet mellom mottatt og utsendt effekt i dette tilfellet er

$$\frac{P_r}{P_t} \sim \frac{h_r^2 h_t^2}{r^4} \quad (5.3)$$

Samme forutsetning om antennevinning som i ligning (5.2).

I dette tilfelle kan vi merke oss at signalstyrkens svekkelse ikke lenger er kvadratisk, men i fjerde potens, eller 40 dB per dekada. Dette kalles *flat jord modellen* (plane earth loss). Denne modellen er heller ikke eksakt nok til å modellere realistiske situasjoner og scenarier, men er mye brukt som en referansemodel.

Denne enkle fremstillingen har vist to meget sentrale komponenter ved modellering av propagasjon; de signalkomponentene som beveger seg i en direkte linje mellom sender og mottaker, samt de komponentene som opplever en eller annen form for hindring underveis. Hindringer underveis vil så bidra med ulike grader av refleksjon, avbøying og spredning, og dette danner grunnlaget for et behov for matematiske modeller i et forsøk på å modellere virkeligheten.

## 5.5 Egli's modell

Egli[6] fant ut at over varierende terreng, for økende frekvenser og avstander ble avvirket fra flat-jord-modellen store, og det måtte dermed korrigeres for dette. En empirisk korreksjonsfaktor inngår derfor i Egli's semiempiriske<sup>2</sup> modell, for avstander mellom 1 kilometer (km) og 50 km, samt for frekvenser mellom 30 MHz og 1 GHz.

$$\frac{P_r}{P_t} = \left(\frac{40}{f}\right)^2 \left(\frac{h_t h_r}{d}\right)^2 G_t G_r \quad (5.4)$$

I ligning (5.4) er  $G_t$  antennevinningen i senderantennen, mens  $G_r$  er vinningen i mottakerantennen. Som vi ser har Egli også en tapsekspont på 4.

## 5.6 Kategorisering av effekter

For topologiemulatorsystemet er det kun aktuelt å betrakte signalsvekkelse i tidsdomenet [8]. Vi betrakter dermed svekkelsen som flat, eller frekvensuavhengig. Dette skyldes

---

<sup>2</sup>Modellen baserer seg på flat-jordsmodellen, men kontrollmålinger avdekket behovet for, og kvantiserte tilleggene

karakteristika ved systemets konstruksjon og oppbygging. Svekkelseseffekter deles normalt inn i tre hovedkategorier, stor-skala, mellom-skala og liten-skala. Disse skal vi se nærmere på.

### 5.6.1 Stor-skala

Dette brukes som en fellesbetegnelse på effekter som påvirker den mottatte signalstyrken relativt sakte, som for eksempel tidligere omtalte svekkelse som følge av avstand mellom sender og mottaker. Den er dermed gjerne karakterisert ved to komponenter. Hovedelementet er en middelvei for svekkelsen som følge av avstand, denne følger en eksponentiell kurve, med en eksponent avhengig av miljøet. Generelt har vi at middelveien av tapet som følge av avstand  $\overline{L}_p(d)$  er proporsjonalt med avstanden mellom sender og mottaker  $d$  med en eksponent  $n$ , relativ til en referanseavstand  $d_0$ . Dette vil si:

$$\overline{L}_p(d) \propto \left(\frac{d}{d_0}\right)^n \quad (5.5)$$

Utrykt i dB, og med uttrykket fra ligning (5.1) får vi:

$$\overline{L}_p(d)(dB) = L_s(d_0)(dB) + 10n \log\left(\frac{d}{d_0}\right) \quad (5.6)$$

### 5.6.2 Mellom-skala

Mellom-skala effekter er et begrep som ofte nyttes om skyggevirkingene som gir en variasjon i banetapet. Svekkelsen varierer hurtigere enn selve banetapet, men ikke fullt så hurtig som liten-skala effektene. Man ser gjerne utslaget av skyggevirkingene dersom antennens posisjon endres i størrelsesorden meter og oppover. Effektene blir variasjoner i banetapet som funksjon av posisjonen, gjerne med en log-normal fordeling<sup>3</sup>. Et slikt varierende bidrag er gjerne i størrelsesorden 6-10 dB. Eksempelvis kan det nevnes at Egli kom frem til et logaritmisk standardavvik på ca 8,3 dB for VHF-frekvenser, og ca 12 dB for UHF-frekvenser[11].

Variasjonene om middelveien som følge av en log-normal fordelt variabel,  $X_\sigma$ , tas inn i systemet fra ligning (5.6) slik at vi får følgende uttrykk for svekkelsen som følge av avstand:

$$L_p(d)(dB) = L_s(d_0)(dB) + 10n \log\left(\frac{d}{d_0}\right) + X_\sigma(dB) \quad (5.7)$$

### 5.6.3 Liten-skala

Disse effektene kommer av små endringer i posisjon, gjerne i størrelsesorden en halv bølgelengde. Basert på dualitetsprinsippet ved Fouriertransformen[15], kan disse effektene observeres både i tids- og frekvensdomenet. I det følgende vil betraktninger i tids-

<sup>3</sup>Fordeling fra hvilken den tilfeldige variabelens logaritmiske verdier er Gaussisk fordelt

domenet legges til grunn. Effektene blir da normalt synlige i form av tidsspredning av signalet selv, samt en tidsvariant kanal. Tidsspredningen av signalet skyldes at i et virkelig propagasjonsmiljø vil ulike deler av det utsendte signalet treffe ulike hindringer, reflekteres i ulike nye retninger, og som et resultat av dette nå mottakeren til ulike tider. Dette kan videre føre til ulike grader av interferens, både konstruktiv og destruktiv. Det faktum at deler av signalet følger sine helt egne veier frem mot mottakeren kalles gjerne ved et samlebegrep for flerbanepropagasjon.

Vi skiller prinsipielt mellom to situasjoner når vi har en flerbanepropagasjon; om det eksisterer en komponent som kan gå direkte mellom sender og mottaker, såkalt Line-of-sight (LOS), eller om alle signalkomponentene som kommer frem er reflektert på en eller flere måter, i et Non line-of-sight (NLOS)-tilfelle. Dersom vi har en direktekomponent betrakter vi liten-skala svekkelsen som en Rice-fordelt tilfeldig variabel, og dersom vi ikke har en slik direktekomponent henter vi bidraget fra en Rayleigh-fordelt variabel. I et typisk scenarie for modellering av propagasjon i et miljø med mobile, og gjerne relativt små mottakere sammenlignet med senderen, er en situasjon *uten* en direktekomponent den fremtredende, og av denne grunn omtales gjerne liten-skala svekkelsen som en Rayleigh-svekkelse.

Generelt i et propagasjonsmiljø med flerbanepropagasjon vil vi også oppleve en Doppler-spredning av spekteret ettersom hver enkelt flerbanekomponent får sitt eget Doppler skift. Dette klarer vi ikke å gjenskape i topologiemulatorene.

#### 5.6.4 Resulterende svekkelse

Resultatet av de ulike svekkelseeffektene er et signal som vi kan si er sammensatt som følger:

$$\alpha(t) = m(t) \times r_0(t) \tag{5.8}$$

Her er  $\alpha(t)$  den totale svekkelsen som virker inn på signalet.  $m(t)$  er en sammensetning av stor- og mellom-skala effektene og  $r_0(t)$  er liten-skala effektene. Systemets logikk gjør beregningene med logaritmiske verdier som adderes, og vi har da fra dualitetsprinsippet at beregningen i ligning (5.8) utføres som en multiplikasjon.

#### 5.6.5 Tidsspredning

Emulatorsystemet er ikke i stand til å simulere en tidsspredning av det mottatte signalet som følge av flerveis propagasjon. Som redegjort for i det innledende arbeidet[8], har vi en flerbanepropagasjon i form av ufullstendig isolasjon mellom kanalene, men den maksimale forskjellen i tid på disse er ubetydelig slik at den kan ignoreres. I tillegg er det slik at det er et svært begrenset antall av disse ekstra veiene, maksimalt to, slik at vi totalt sett har tre signalkomponenter når vi vurderer tidsdifferanse hos mottakeren. Avstandsmessig differanse mellom disse signalveiene er på ca to meter, og med en forsinkelse i de aktuelle kablene på 4,7 nanosekund (ns)/meter (m)[9], har vi en maksimal tidsforskjell på  $\sim 10$  ns. Dette innebærer at vi ikke opplever noen frekvensselektiv oppførsel i kanalen, kun et

lite tap i mottatt SNR, såkalt flat svekkelse. Dette tapet blir tatt hensyn til i forbindelse med kalibrering av emulatoren.

## 5.7 Kompleksitet og modeller

Propagasjonsmiljøet er svært sjelden så enkelt og fritt for fremmedlegemer som tidligere omtalt. Det vil i mange tilfeller eksistere hindringer mellom sender og mottaker, og som eksempler på dette kan vi tenke oss skiftende topografi, vegetasjon og menneskeskapt elementer som bygninger og strukturer. Det er åpenbart at et system for modellering av dette må basere seg på forenklinger og generisk tilnærming. Kompleksiteten forbundet med å kartlegge alle parametre i et hvert scenarie slik at det stemmer med virkeligheten blir så massiv at det nok i så fall er enklere å faktisk utføre målingene i det aktuelle miljøet.

Topologiemulatorsystemet krever ikke en ekstremt høy grad av realisme da systemet i høyeste grad har en generisk karakter. Bruksområdet er primært testing og verifisering av rutingprotokoller i sammenheng med ulike former for radiomateriell, og således er ikke den eksakte likheten med et stykke terreng påkrevet. Hensikten er å oppnå et dynamisk miljø av skiftende karakter, noenlunde tilsvarende det man vil se i virkeligheten.

Det er utviklet et stort antall modeller for simulering av egenskapene ved radiokanaler. Disse kan deles inn i fysiske og empiriske modeller. Fysiske modeller tar sikte på å modellere kanalens egenskaper ved hjelp av matematiske sammenhenger, mens empiriske modeller tar utgangspunkt i utførte målinger og erfaringer i bestemte situasjoner, og tilpasser modellen til disse. Der erfaringene ikke fullt ut dekker et aktuelt scenario utføres gjerne en ekstrapolasjon så langt det lar seg gjøre. Generelt kan det sies at fysiske modeller er relativt nøyaktige og gir en god tilnærming til virkeligheten, men med en dertilhørende kompleksitet i implementeringen. Empiriske modeller har den fordel at de generelt sett er relativt enkle å implementere, men jo mer rammebetingelsene avviker fra den eksakte situasjonen modellen er laget ut fra, jo større feil og avvik får man fra virkeligheten.

Når vi skal begynne å se på en modell å bruke i denne sammenhengen har vi to prinsipielt ulike måter å betrakte forbindelsen på. På den ene siden kan vi se på forbindelsen mellom to noder som fast, det vil si at nodene er stasjonære. En annen mulighet er å betrakte én node som hovednode og de resterende som mobile enheter innen rekkevidde av hovednoden, en cellulær inndeling slik vi kjenner det fra blant annet mobiltelefonsystemene. Vi skal se nærmere på de to variantene, samt eksempler på tilhørende eksisterende modeller.

## 5.8 Faste forbindelser

Posisjonering i emulatorsystemet vil, i likhet med posisjonering i virkeligheten, kunne diskretiseres. Over en gitt tidsperiode kan vi derfor nesten uansett definere sender og mottaker som stasjonære. Dersom forbindelsen mellom to radionoder således tenkes å være mellom to stasjonære enheter, kan fastsetting av banetapet over forbindelsen da

gjøres etter følgende kokebokmodell [13]. I denne beregningen tas det høyde for effektene utover frittromstapet, og vi kan betrakte den som en flat-jord modell som tar høyde for tapseffekter forårsaket av hindringer i forbindelsen. Det primære er dog at den tar høyde for atmosfærens effekt på signalet.

1. Bestemme posisjon og høyde for antennene
2. Bestemme linjen kjent som "den store sirkelen"<sup>4</sup> mellom antennene
3. Fastslå terrengprofilen mellom antennene fra kartdata
4. Korrigere terrenghøyden for eventuelle bygninger og andre hindringer langs linjen
5. Velge en korreksjonsverdi for effektiv jordradius tilpasset prosentvis tid som systemet skal være tilgjengelig, og så korrigere signalveiprofilen i henhold til denne verdien. Dette skal korrigere for endringer i forbindelse med refleksjon i troposfæren.
6. Beregne frittromstapet for den aktuelle veien
7. Dersom hindringer finnes innenfor 0,6 x den første Fresnel-sonen må avbøyningen over disse hindringene beregnes og tilleggstapet dette medfører legges inn i linkbudsjettet
8. Beregne den delen av veilengden som går gjennom trær og ta inn dette tilleggstapet.

Dersom systemet krever stor grad av tilgjengelighet, noe et system som det som skal simuleres med emulatoren gjør, må det i tillegg tas hensyn til flerbaneløstapet som kommer som følge av kanalisering og refleksjoner.

Dersom emulatorsystemet skulle nytte denne beregningsmetoden ville det innebære en stor mengde beregninger som må gjøres kontinuerlig ettersom noder i bevegelse kun kan anses stasjonære over meget kort tid. Denne tiden kan dog anses relativt sett lang i forhold til variasjoner i kanalen samt aktuell symbolvarighet. For utregning og fastsetting av nødvendige parametre vil den relative bevegelsen mellom nodene innebære at det å tilnærme seg problemet som stasjonære noder og beregne etter ovennevnte modell er lite attraktivt. Resultatet kan bli tapt ytelse i form av en kanal som ikke er i stand til å endre egenskaper tilfredsstillende raskt, og vi vil dermed ikke oppnå den grad av realisme som er ønsket. Samtidig kan den nøyaktigheten som her oppnås betraktes som "for god" og unødvendig for systemet som skal modelleres.

## 5.9 Makroceller

To radionoder kan også betraktes innen rammen av et makrocellesystem. Her ser vi prinsipielt på den ene noden som en sender, og den andre som en mottaker. Senderen sender

---

<sup>4</sup>Rett linje mellom antenne som er korrigert for jordoverflatens krumning

et rundtstrålende signal, og vi betrakter dekningsområdet som sirkulært. En vanlig forenkling i dette scenarioet er å betrakte området som belyses av senderen for uniformt, noe som forenkler beregningen av propagasjonen betraktelig. I denne sammenhengen nyttes gjerne empiriske modeller for beregning av transmisjonstapet. Eksempler på slike modeller er gitt i [13]

### 5.9.1 Empiriske modeller

**Clutter Factor Models** Disse modellene nyttiggjør seg det faktum at målinger i by- og forstads miljø gjerne viser en tapsekspont på fire som i flat-jord modellen, men med et større tap i absoluttverdi. Dette har ført til modeller som bruker samme flat-jord modell, men også tar med et ekstra tap som følge av forstyrrende elementer langs signalveien.

**Okumura-Hata** Fullstendig empirisk modell, basert på målinger utført i Tokyo, Japan. Målingene er utført i frekvensområdet 200 MHz - 1,5 GHz. Modellen opererer med tre ulike terrengetyper:

- Åpent område - Helt uten hindringer i ca 300-400 m avstand fra senderen
- Forstadsområde - Spredt bebyggelse og vegetasjon, noen hindringer ved den mobile noden, men ikke veldig tett
- Byområde - Tett bebyggelse, høye bygninger, trange passasjer, tett vegetasjon

**COST 231-Hata** En utvidelse av Hata-Okumura modellen til å gjelde for frekvenser i området 1,5 GHz - 2 GHz. Denne kan nyttes for små byer, forsteder og større bymiljøer.

**Lee's modell** Basert på målinger utført ved 900 MHz. Modellen har mulighet for korreksjon for avvikende frekvenser. Målinger er tatt i en rekke forskjellige miljøer, og parametre eksisterer for tilpassing til ulike terrengetyper fra helt åpent område til New York City.

**Ibrahim og Parsons modell** Denne modellen er basert på målinger utført rundt London, England. Modellen er ikke en komplett propagasjonsmodell, men et verktøy for kvantisering av tapet i bymiljø. Modellen deler opp det aktuelle området i kvadrater med sider av 500 m lengde. Målingene er utført ved 900 MHz.

Empiriske modeller tilbyr gjerne en slags 'blind' måte å tilnærme seg problemet på, og bidrar derfor i svært liten grad til innsikt i hvilke sammenhenger som råder. De fysiske modellene har ofte et større eller mindre empirisk element i seg. I det følgende er noen eksempler på slike kombinerte modeller.

**Allsebrook og Parsons modell** Dette er en tidlig modell basert på målinger i ulike storbymiljøer i Storbritannia. Målingene ble utført på frekvenser under 500 MHz, og modellen er derfor ikke aktuell for emulatorsystemets mye bredere frekvensområde. Modellen er likevel interessant i det at den kombinerer empiriske data med

en korreksjonsfaktor for avviket som oppstår ved ekstrapolering opp i frekvens. Denne korreksjonen tilnærmes ved å anta en 10 m høy vegg 30 m unna den mobile enhetens to meter høye antenne. Deretter beregnes veggens innvirkning på propagasjonen. Modellen var den første som tok høyde for spesifikke forhold ved en signalvei.

**Ikegami** Modellen utvider det fysiske elementet ytterligere og modellerer en mobil enhet midt mellom to vegger av lik høyde. Kun én refleksjon fra veggene antas. De største feilkildene ved modellen er at høyden på basestasjonens antenne antas uvesentlig og at den baserer seg på en frittroms tapsekspont. Dermed underestimerer den tapet ved store avstander, og variasjonene ved frekvens er i virkeligheten større enn det som beregnes. Kontrollmålinger er utført mellom 200 MHz og 600 MHz. Dette frekvensområdet dekker kun i meget liten grad det dynamiske området til topologiemulatoren, og således vurderes modellen som lite egnet.

**Ytterligere komplisering av det fysiske miljøet** Det finnes også et antall modeller som fortsetter fra de tidligere nevnte modellene. Med utgangspunkt i et miljø tilsvarende det i Ikegami-modellen, økes antallet vegger suksessivt i modellene slik de er nevnt her:

- Avbøying som følge av hustak-modellen
- Flatkant-modellen
- Walfisch-Bertoni-modellen
- COST 231 / Walfisch-Ikegami-modellen

COST 231 / Walfisch-Ikegami er gyldig for store deler av det aktuelle frekvensområdet, opp til 2 GHz, men kompleksiteten anses for stor i forhold til målsetningen med topologiemulatorsystemet

## 5.10 Valg av kanalmodell

Basert på ovennevnte argumenter vil kanalmodellen som implementeres i styringslogikken til topologiemulatoren baseres på en sammensetning av følgende elementer:

- Et banetap som direkte følge av avstanden. Dette tapet modelleres med en implementasjon av Egli's modell[6].
- Et log-normalt tillegg til dette tapet, som simulerer skyggevirksomhet.
- En tidsdomenebasert flerbanesvekkelse av signalet som baserer seg på en sammensetning av en Rice-fordelt komponent og en Rayleigh-fordelt komponent. Graden av friskt mellom nodene (LOS eller NLOS) settes som en direkte følge av avstanden mellom nodene. Dette er naturligvis en grov forenkling av virkeligheten, men godt nok for systemets bruksområde.



### 5.10.1 Banetap

Denne delen av kanalmodulen har som primæroppgave å beregne tapet i SNR som følge av avstanden mellom nodene. Inngangsparametrene er avstanden mellom to noder, som betraktes som avstand i meter, frekvens i MHz, antennehøyder i meter, og avstander tapet skal beregnes for. Banetapet beregnes etter Egli's modell, og utgjør stor-skala bidraget slik det er skissert i punkt 5.6.1.

### 5.10.2 Skyggevirksomhet

Etter beregningen av banetapet sitter vi med et monotont økende tap som følge av avstand. Dette er ikke nødvendigvis en realistisk modellering for det avstandsbestemte tapet i SNR. For å modellere skyggeeffektene skissert i punkt 5.6.2 genererer vi en normalfordelt variabel med standardavvik 12. For at bidraget skal være av mellom-skala karakter utføres en glatting av den normalfordelte variabelen umiddelbart etter generering. Glattingen gjøres med et rektangulært glidende gjennomsnitt vindu, med bredde 21. Det gjøres så en skalering i amplitude. Etter glatting legges variabelen til banetapsverdiene, og i sum gir dette oss tapet i SNR som følge av avstand.

### 5.10.3 Flerbanetap

Denne delen av modulen skal simulere de effektene som oppstår som følge av flerbaneeffekter. Dette gjøres ved å innledningsvis generere to fordelinger; én Rayleigh-fordeling, og én Rice-fordeling. Som tidligere redegjort for representerer den Rice-fordelte variabelen en situasjon med en sterk signalkomponent tilstede, en situasjon vi benytter som et friskttilfelle. Den Rayleigh-fordelte variabelen modellerer situasjonen hvor det ikke eksisterer noen distinkt sterk komponent. Dette tilsvarer et propagasjonsmiljø hvor samtlige mottatte signalkomponenter i mottakeren er reflekterte, og det er ingen fri sikt.

I emulatorsystemet har vi ingen terrengdata å forholde oss til, kun relativ avstand nodene mellom. Dermed velges det å la avstanden mellom to noder si noe om graden av fri sikt. Systemet er laget slik at en parameter,  $k$  gir denne graden av fri sikt. Brukeren må ved oppstart bestemme en verdi i antall meter hvor fri sikt tapes. For eksempelets skyld kan vi her anta at den er 1000 m. Systemet er deretter strengt lineært i sin fastsetting av graden av fri sikt, på følgende måte:

- Avstanden mellom null m og parameteren brukeren gir inn (i dette tilfellet 1000 m) deles inn i 11 like intervaller.
- Fra null tildeles hvert intervall  $k$ -verdi startende på 1,0
- Det siste intervallet blir dermed fra brukerens inngangsverdi og til uendelig, og dette intervallet tildeles  $k=0$  (fri sikt tapt)

Et eksempel på denne inndelingen er gitt i tabell 5.1, også her med antatt inngangsverdi for tapt fri sikt lik 1000 m.

<i>Avstand</i>	<i>Koeffisient</i>
0 - 100 m	1,0
100 - 200 m	0,9
200 - 300 m	0,8
300 - 400 m	0,7
400 - 500 m	0,6
500 - 600 m	0,5
600 - 700 m	0,4
700 - 800 m	0,3
800 - 900 m	0,2
900 - 1000 m	0,1
1000 - $\infty$	0,0

Tabell 5.1: Tabell over frisiktkoeffisienten  $k$  som følge av avstand

Det totale beregnede flerbanetapet settes sammen av de Rice- og Rayleigh-fordelte variablene på følgende måte.

$$X_{total} = kX_{Rice} + (1 - k)X_{Rayleigh} \quad (5.9)$$

$X_{total}$  er den resulterende tilfeldige variabelen som nyttes til å modellere flerbanetapet basert på en gitt avstand.  $X_{Rice}$  representerer den Rice-fordelte variabelen,  $X_{Rayleigh}$  den Rayleigh-fordelte variabelen, mens  $k$  er koeffisienten fra tabell 5.1.  $X_{total}$  blir så multiplisert med dataene fra ligning (5.9), og det gjøres så en konvertering til logaritmiske verdier i dB<sup>5</sup>.

#### 5.10.4 Oppsummering

Denne kanalmodellen har den åpenbare svakheten at den er meget generisk, og tar ingen hensyn til terrenget som omgir nodene. Til bruken systemet er tenkt, er den imidlertid god nok ettersom det er de relative endringene over tid som er interessante i en testsituasjon. Det vesentlige er at de relative endringene over tid og avstand har en likhet med slik man vil oppleve det på en aktuell trådløs kanal. En klar fordel med oppbyggingen av systemet slik det er nå, er at det er relativt enkelt for en bruker å gå inn å endre på logikken, det være seg enkelte parametre, eller operasjoner nyttet i beregningene.

---

<sup>5</sup>I emulatorens logikk utføres operasjonene med logaritmiske verdier, og således summeres de.

## Kapittel 6

# Emulatorkontroll

### 6.1 Arv

Styringslogikken som ligger til grunn ble utviklet ved FFI/Ledelsessystemer<sup>1</sup> for enkel bruk av emulatoren. Rutinen består av én VI<sup>2</sup>. Ved oppstart setter programmet verdier for maksimal demping (127 dB) på alle 15 kanaler, og avventer så operatørens parametre. I en [15x1]-matrise kan så operatøren manuelt legge inn verdier for dempeverdi på den enkelte kanalen. 1 - 15 kanaler kan påvirkes per operasjon. Når ønskede dempeverdier er lagt inn i matrisen og operasjonen er bekreftet, setter programmet de ønskede verdiene i emulatoren. Dette skjer ved at verdiene sendes fra LabVIEW via et Universal Serial Bus (USB)-grensesnitt til to National Instruments DAQPad'er<sup>3</sup>[8][10], som videre setter det ønskede bitmønsteret på de ulike linjene. Dette fører igjen til at den aktuelle dempningsverdien settes på kanalen av de programmerbare dempeleddene. Systemet vender så tilbake til en hviletilstand hvor det venter på nye inngangsverdier fra operatøren.

### 6.2 Utfordring og detaljer

En utfordring som er avdekket gjennom arbeidet med oppgaven er knyttet til hvordan bitmønsteret settes på DAQPad'ens utgangslinjer. Styringen av DAQPad'en fra LabVIEW er relativt triviell å implementere da det aller meste er standardfunksjoner i LabVIEW. Manipuleringen av en verdi i et dempeledd krever fire innparametre; DAQPad enhet, linje nummer, tilstand og selve verdien.

**Enhetsnummer** Hver DAQPad, som vi har to av, kan betjene åtte underenheter. I vårt tilfelle innebærer dette at de programmerbare dempeleddene er delt inn i to grupper og DAQPad'ene er tildelt enhetsnummer 1 og 2.

---

<sup>1</sup>ved forsker Vegard Arneson

<sup>2</sup>VI=Virtual Instrument; National Instruments betegnelse på et LabVIEW program

<sup>3</sup>DAQPad'ens oppgave er å oversette en numerisk verdi til et bitmønster som den setter på et antall utgangslinjer.

**Linjenummer** Linjene identifiseres fra LabVIEW med et ”digital kanal nummer” i området 0-11. I DAQPad mappes dette til en 3-bokstavsforkortelse med betegnelse [A-D]P[A-C] på utgangen

**Tilstand** Denne parameteren er nøkkelen til å forbedre systemets ytelse. Dersom verdien er null, kjøres en konfigureringsrutine som setter opp kommunikasjonen mellom LabVIEW og DAQPad riktig med tanke på adresseområder, datastørrelser også videre. Som del av denne rutinen skrives alle verdiene på utgangen midlertidig til null. I tillegg til at konfigureringen innfører et tidstillegg, vil denne ”nullverdi”-skrivningen innebære et urealistisk kanalforløp.

**Verdi** Dette er simpelthen den desimale dempeverdien som ønskes satt på det aktuelle dempeleddet

### 6.3 Løsning

Den eksisterende rutinen er i seg selv nyttig og effektiv, men det er ønskelig å automatisere prosessen noe. Slik det er redegjort for tidligere i denne rapporten bør prosessen med verdisetting på de enkelte kanalene skje automatisk, og på bakgrunn av beregnede verdier for kanaldemping. Den eksisterende rutinen kan dog ikke kalles gjentatte ganger<sup>4</sup> helt uten videre, ettersom vi da for hver utførelse får ”nullverdi”-problematikken. En enkel justering av rutinen er dermed utført slik at den ved gjentagende anledninger kan kalles eksternt, og nyttes til fortløpende manipulasjon av kanalene. Tilstandsparameteren er dermed ”0” kun ved oppstart av hele scenariet. I tillegg til dette er det foretatt endringer for å muliggjøre innlesing av dempeverdier fra en matrise lagret i en annen del av programmet. Rutinen er også endret slik at antallet programmerbare dempeledd som endres er avhengig av antallet noder som deltar i scenariet. Dette bidrar til en vesentlig reduksjon av tiden det tar å få satt samtlige verdier.

Dermed kan tiden mellom kanalverdisetting reduseres. Fra tidligere er det konkludert med 50 millisekund (ms) som en nedre grense, men med denne endringen kan lavere verdier også nyttes. Ved å ha færre enn seks noder deltagende i scenariet kan tider ned til 20 ms håndteres av systemet. Ved en slik endring må antallet genererte verdier per sekund også endres slik at totalt antall verdier ”fyller” et sekund.

---

<sup>4</sup>Det er ikke ønskelig at rutinen står i ventestilling til enhver tid, da dette binder opp systemressurser. Gjennom hele systemet er det valgt å kjøre rutiner kun ved behov så lenge dette er det mest effektive, for å minimere ressursbruken.

# Kapittel 7

## Grensesnitt

### 7.1 Overordnet kontroll

Grensesnittet har en tredelt funksjon. Det gir brukeren mulighet til å sette startverdier samt å endre parametre. Dernest forestår det den overordnede programkontrollen, delvis transparent for brukeren. Den tredje hovedfunksjonen er å presentere data, status og fremdrift overfor brukeren. Grensesnittet slik brukeren møter det er gjengitt i figur 7.1. Systemet er bygget opp slik at brukeren innledningsvis må definere et scenario basert på et antall parametre. Scenariet startes så, og kjøres automatisk. Brukeren vil i løpet av scenariet kunne monitorere et situasjonsbilde som grafisk viser nodenes bevegelser, samt avstandene nodene mellom i form av en matrise. Avslutningsvis sørger grensesnittsrutinen for at data fra det kjørte scenariet skrives til tre loggfiler.

### 7.2 Verdier og parametre

Ved definering av et scenarie må altså brukeren bestemme en rekke parametre. Parametrene legges så inn i et grafisk kontrollmiljø i LabVIEW, og scenariet startes. Parametrene er som følger:

**Antall iterasjoner** Hvor mange iterasjoner skal gjennomføres. Dette sammenfaller med antallet posisjonsendringer for nodene, og antall sekunder scenariet kjøres.

**Antall noder** Hvor mange noder som skal delta i scenariet. På grunn av topologi-emulatorens koblinger anses én deltagende node som uaktuell. Det minste antall dempeledd som kan manipuleres i et scenarie er dermed ett<sup>1</sup>. På grunn av en arbeidsmengdevurdering i forbindelse med denne oppgaven er systemet konstruert slik at nodene i et scenarie må kobles til emulatoren sekvensielt fra tilkoblingspunkt én<sup>2</sup>. Lovlige verdier er to til seks.

---

<sup>1</sup>Se punkt 6.3

<sup>2</sup>Første punkt: R1 - R2, andre punkt: R1 - R3 osv

**Frekvens** Hvilken frekvens det tilkoblede radiomateriellet opererer på. Lovlige verdier er i området 200 MHz - 3 GHz

**Frisikt tapt avstand** Dette er avstanden i meter hvor fri sikt mellom nodene anses tapt (se punkt 5.10.3). For nodeavstander over denne verdien har vi dermed et rent NLOS-tilfelle, og dette innvirker på utregningen av flerveistapet

**Path loss random factor** Denne bestemmer graden av variasjon rundt banetapet, og skal simulere skyggevirksomheter. Det er en rent multiplikativ faktor med lovlige verdier fra 0 til 1

**Høyde antenne 1** Dette er høyden antenne 1 har over bakken

**Høyde antenne 2** Dette er høyden antenne 2 har over bakken

**Standardavvik banetap** Standardavviket på variasjonene i banetap som funksjon av tiden. Denne har standardverdi 12 dB slik Egli konkluderte med for UHF-frekvenser. Dette er tidligere omtalt i punkt 5.6.2

**Flerbanetapsberegning av eller på** Denne parameteren bestemmer om det skal gjøres beregninger for flerbanetap, eller om det kun skal beregnes tap som følge av avstander (banetap)

**Kalibreringsmatrise** Her velger brukeren om systemet skal nytte sin egen kalibreringsmatrise, om brukeren vil legge inn sin egen matrise, eller om systemet skal nytte matrisen fra forrige kjøring

**Korreksjon for innsetningsdemping** Hvorvidt det skal gjøres en korreksjon for innsetningsdempingen i emulatoren. Dette innebærer i så fall et fratrekk på 49 dB på verdiene rett før de skrives til systemet

**Startposisjoner** Settes som enheter basert på et kartesisk koordinatsystem, og relativt til origo. Oppløsningen i systemet er 1 m, og verdiene på koordinatsystemaksene er også ekvivalent med meter

**Nodehastigheter** Nodenes hastigheter. Systemet oppdateres én gang per sekund, slik at en nodes hastighet er definert som antall enheter bevegelse pr iterasjon. Dette blir dermed analogt med antall meter per sekund

**Mobilitetsmodell** Hvilken mobilitetsmodell som skal ligge til grunn for nodenes virtuelle bevegelser. Brukeren kan velge mellom en stjerneformet bevegelse og en implementasjon av en Gauss-Markov mobilitetsmodell<sup>3</sup>

**Maks meter ut (Stjernemodell)** Bestemmer hvor langt ut fra origo nodene skal bevege seg før de snur, og begynner å bevege seg innover igjen.

---

<sup>3</sup>Se kapittel 4

I tillegg til dette blir brukeren ved oppstart av scenariet bedt om å spesifisere navnet på mappen som skal inneholde scenariets loggfiler. Dette omtales nærmere i punkt 7.5. Brukeren har også muligheten til å avslutte kjøringen ved å benytte ”Stopp”-knappen i statusfeltet. Dette avbryter kjøringen i den iterasjonen systemet er i når knappen trykkes, og verdiene så langt skrives til fil.



Figur 7.1: Grensesnittet mot brukeren

### 7.2.1 Korreksjonsmatriser

Som tidligere omtalt ligger det inne standardverdier for korreksjoner basert på kalibreringsarbeidet. Ved oppstart kan brukeren velge å nytte disse verdiene, eller å legge inn egne korreksjonsmatriser. Dette valget må tas i det grafiske kontrollgrensesnittet. Velges det å legge inn egne verdier legges disse inn i en LabVIEW-dialog, og deretter lagres matrisen i mappen `C:\Topologiemulator\Scenarier` med filnavn `Kalibreringsmatrise.txt`. Filen bygges opp som en  $[15 \times 1]$ -matrise hvor radene er forbindelsene og kolonnen kanaldempningen.

### 7.3 Programkontroll

Grensesnittet er i realiteten programmet som styrer hele systemet. Det er her hovedløkken kjøres, og det er fra denne alle kall til underliggende funksjoner gjøres, samt kontroll over underliggende løkker forestås. Den prinsipielle virkemåten er som følger:

- Basert på startposisjonene beregnes avstanden mellom alle noder.
- Nodenes neste posisjon bestemmes, basert på aktuell mobilitetsmodell.
- Avstanden mellom nodene gis som innparametre til kanalmodulen som beregner kanalverdier for 1 sekund for samtlige forbindelser deltagende i scenariet.
- Verdier for kanalene for 1 sekund (20 stk) gis som verdier til kontrollmodulen som løper gjennom disse verdiene i løpet av sekundet.
- Løkken starter forfra og beregner gjeldene avstand mellom nodene.

Prosedyren gjentas til antall posisjonsendringer tilsvarer antall steg gitt som innparameter av brukeren.

Ved avslutning av et scenarie får brukeren tre valg; scenariet kan kjøres én gang til med samme parametre, gjeldende kjøring kan avsluttes, og hele seansen kan avsluttes. Ved sistnevnte valg vil LabVIEW avsluttes.

### 7.4 Presenterte data

Mens et scenarie er under utførelse presenteres noe data fortløpende for brukeren. Disse er:

**Nodeposisjon og bevegelse** Nodenes posisjoner med en historikk, presenteres i sann tid for brukeren. Dette innebærer at et situasjonsbilde oppdateres hvert sekund, og at nodenes bevegelsehistorikk beholdes slik at deres tilbakelagte vei er synlig

**Nodeavstander** Den til enhver tid oppdaterte avstanden mellom nodene

### 7.5 Lagring av data

Systemet muliggjør gjennomgang av data etter at scenariet har blitt kjørt. For hver oppdatering av systemet gjennom et scenarie lagres en datamengde som beskriver situasjonen til flere filer. Ved endt scenarie skrives så disse dataene til 3 loggfiler. Loggfilenes dimensjoner kan bli relativt store, og for å gi en indikasjon på disse kan vi se for oss et scenarie der nodenes posisjon oppdateres  $i$ -ganger.

Følgende loggfiler produseres av systemet ved kjøring:

**Posisjoner.txt** Filen logger nodenes posisjoner. Det vil være to kolonner i filen, for henholdsvis nodenes kartesiske x-posisjon og y-posisjon. Ettersom det er 6 noder i systemet vil filen ha et antall rader tilsvarende  $Antallrader = i * 6$



**Avstander.txt** Denne loggfilen tar vare på avstanden mellom de ulike nodene for hver posisjonsoppdatering. Det er 1 kolonne i filen, og ettersom vi har 15 stk nodeforbindelser vil denne filen ha et antall rader tilsvarende  $Antallrader = i * 15$

**Kanalverdier.txt** For hver posisjonsoppdatering beregnes en serie verdier for kanalenes dempning. Systemet oppdateres hvert sekund og er definert slik at en kanalverdi er gyldig i 50 ms. Dermed vil et sekund bestå av 20 verdier for kanaldempningen, og antall kolonner i filen vil dermed være 20. Ettersom det legges inn en blank rad mellom hver posisjons kanalverdier bestemmes antall rader av sammenhengen  $Antallrader = (i * 16) - 1$

Loggfilene for kjøring av et scenarie lagres i en mappe med et navn som velges av brukeren ved oppstart. Denne mappen opprettes automatisk i `C:\Topologiemulator\Scenarier`.



## Kapittel 8

# Konklusjon

Arbeidet med dette prosjektet har i all hovedsak konsentrert seg om utvikling av en styringsdynamikk for topologiemulatoren ved Forsvarets forskningsinstitutt (FFI), avdeling for Ledelsessystemer. Topologiemulatoren er en fysisk sammenkobling av inntil seks radionoder i en alle-til-alle konfigurasjon, med brukerstyrt mulighet til å endre kanaldempningen mellom alle nodene. Alle forbindelser er toveis, og således gir dette 15 ulike forbindelser som kan påvirkes individuelt. Det primære bruksområdet for emulatoren vil være i forbindelse med såkalte Mobile Ad Hoc Networks (MANET). I denne forbindelsen vil systemet kunne være et godt verktøy for studier og verifikasjon av aktuelle rutingprotokoller i sammenheng med ulike typer radiomateriell.

Prosjektet tok utgangspunkt i en statisk kontrollogikk, som så har dannet basisen for en dynamisk styring av hele systemet. Den opprinnelige kontrollogikken baserte seg på at brukeren ga inn en verdi for ønsket demping over en forbindelse. Verdien ble valgt fra mengden 0 - 127 desibel (dB), men på grunn av manglende kalibrering hadde brukeren ingen kvalifisert oppfatning av hvilken verdi som fysisk var tilfelle over forbindelsen. Fraværet av dynamikk gjorde systemet lite egnet for scenarier og situasjoner hvor parametrene er i endring, og det var derfor ønskelig å forbedre dette.

Innledningsvis ble systemets karakteristikk og egenskaper studert, slik at en kalibrering kunne utføres. Dette arbeidet ble utført som del av prosjektoppgaven i foregående semester[8]. Med utgangspunkt i dette ble det i hovedoppgaven utviklet en konseptuell skisse over hvordan kontrollogikken kunne bygges opp. Det bærende prinsippet i dette arbeidet var å tilstrebe en så modulær oppbygging som mulig. Dette for å gjøre det enklere for brukere å tilpasse dynamikken til den aktuelle situasjonen, samt å gjøre det mulig å nytte kun deler av systemet dersom det synes formålstjenlig. Skissen som ble lagt til grunn for arbeidet har gjennom arbeidet med prosjektet vist seg å være gunstig, og av den grunn er det også slik systemet er bygget opp ved termineringen av dette prosjektet. Skissen er tidligere vist i figur 3.3.

Med bakgrunn i ovennevnte har hovedtyngden av arbeidet funnet sted med de såkalte posisjons- og kanalmodulene. Posisjonsmodulen har som sin oppgave å simulere en kontinuerlig endring av noderes virtuelle posisjon. Dette gjøres ved å generere det ønskede antall posisjonskoordinater for ønsket antall noder. Genereringen skjer før selve

scenariet startes, slik at belastningen på systemet, som kjører i sann tid, gjøres mindre. Dette gir den gunstige effekten at hastigheten på endringene av en kanals dempeverdier kan økes dersom ønskelig. Systemet tar så for seg alle nodenes posisjoner i et gitt tidspunkt, og nytter disse som basis for flere beregninger: avstanden mellom alle nodene regnes ut, og basert på prinsippet om at forbindelsene ikke har noen retning gir dette 15 avstander. Samtidig med dette utføres en plotting av alle noders posisjoner. Denne plottingen nytter alle posisjoner opp til og med den nåværende, slik at plottet brukeren ser inneholder alle nodenes utførte bevegelser. Etter disse beregningene sendes avstandene til kanalmodulen. Kanalmodulen ser på avstanden mellom nodene, og gjør så en rekke beregninger basert på disse. Først fastsettes et tap basert på avstanden (path loss). Tapsberegningen gjøres med en implementasjon av Egli's propagasjonsmodell[6]. Denne gis så et log-normalt tillegg for å simulere skyggevirksomheter. For å ta hensyn til flerbaneeffektene (multipath fading) beregnes så et tap basert på en Rice-fordelt variabel og en Rayleigh-fordelt variabel. Disse simulerer i isolerte former henholdsvis Line-of-sight (LOS)-forhold og Non line-of-sight (NLOS)-forhold. Delresultatene kombineres slik at kanalmodulen returnerer en tidsserie med dempeverdier for hver forbindelse. Systemet er konfigurert slik at nodeposisjoner oppdateres én gang per sekund, og kanalmodulen beregner 20 verdier for hver forbindelse (15) per posisjon. Avslutningsvis gis kanalverdiene inn til emulatorekontrollmodulen som sekvensielt setter de 20 verdiene gjeldende for hver forbindelse i den aktuelle posisjonen. Dette gjøres med 50 ms forsinkelse mellom hver verdi, slik at kontrolløkken dermed kjører én gang per sekund.

Algoritmene som nyttes til posisjonsbestemmelser og utregning av dempeverdier til kanalene er relativt enkle. På begge områder er det tildels meget vanskelig å kvantisere graden av realisme. I et system som dette blir det dermed en avveining hvor mye realisme det er kosteffektivt å etterstrebe, all den tid emulatoren kanskje bare er tenkt brukt til verifisering av systemers virkemåter og ytelser i enkelte situasjoner. Et videre arbeid med emulatorsystemet burde primært konsentrere seg om en videreutvikling av posisjons- og kanalmodulen. Det er ved videre utvikling av disse at bruksområdene kan endres og økes ved å øke graden av realisme og å øke antallet modeller implementert. Posisjonsmodulen kan tenkes å utvides med ulike mobilitetsmodeller[7] til å omfatte flere alternativer for enkeltnoders bevegelser, samt modeller for gruppevis mobilitet. Kanalmodulen kan, fra dagens Egli-baserte banetapsberegning og generiske flerbanetapsberegning, videreutvikles til å nytte detaljerte propagasjonsmodeller som gir en mer realistisk tids- og posisjonsavhengig variasjon.

# Bibliografi

- [1] Basile Agba, Francois Gagnon, and Ammar Kouki. Simulating Tactical Ad Hoc Networks: Mobility and Channel Modeling. Rapport.
- [2] Vincent Borrel, Marcelo Dias de Amorim, and Serge Fdida. On Natural Mobility Models. *Internet*.
- [3] Tracy Camp, Jeff Boleng, and Vanessa Davies. A Survey of Mobility Models for Ad Hoc Network Research. *Internet*, April 2002.
- [4] David K. Cheng. *Field and Wave Electromagnetics*. Addison-Wesley Publishing Company, 75 Arlington Street, Suite 300, Boston, MA 02116A, USA, 2nd edition, 1989.
- [5] Rohde & Schwarz GmbH & Co. Vector Network Analyzer Family ZVR. Internet, February 2005.
- [6] John J. Egli. Radio Propagation Above 40 MC Over Irregular Terrain. *PROCEEDINGS OF THE IRE*, June 1957.
- [7] Anders Fongen and Morten Gjellerud. A Military Mobility Model for MANET Research. June 2007.
- [8] Morten Gjellerud. Topologiemulator for Trådløse Nettverk. Prosjektrapport TFE 4605, Universitetsstudiene på Kjeller, Forsvarets forskningsinstitutt, Norges Teknisk-Naturvitenskapelige Universitet, 2006.
- [9] HUBER+SUHNER. Data Sheet Coaxial Cable SUCOFORM 141 CU PE. Internet, April 2005.
- [10] National Instruments. DAQPad 6507 & 6508 data sheet. Internet.
- [11] Victor Jones. Radio Propagation Models. Internet, 2003.
- [12] Philippe Nain, Don Towsley, Benyuan Liu, and Zhen Liu. Properties of Random Direction Models. Forskningsrapport, Institut national de recherche en informatique et en automatique, Domaine de Voluceau, Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France, July 2004.

## *BIBLIOGRAFI*

---

- [13] Simon R. Saunders. *Antennas and Propagation for Wireless Communication Systems*. John Wiley & Sons, LTD, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 5th edition, 2004.
- [14] Deepanshu Shukla. Mobility Models in Ad Hoc Networks. *Internet*, November 2001.
- [15] Bernard Sklar. *Digital Communications Fundamentals and Applications*. Prentice Hall P T R, 1 Lake Street, Upper Saddle River, NJ 07458, USA, 2nd edition, 2001.

# Tillegg A

## Forkortelser

**4G** fjerde generasjons mobiltelefonsystemer

**DAQ** Data Acquisition

**DARPA** Defense Advanced Research Project Agency

**dB** desibel

**FFI** Forsvarets forskningsinstitutt

**GHz** Gigahertz,  $10^9$  hertz

**km** kilometer,  $10^3$  meter

**LOS** Line-of-sight

**m** meter

**MANET** Mobile Ad Hoc Networks

**MHz** Megahertz,  $10^6$  hertz

**ms** millisekund,  $10^{-3}$  sekund

**NLOS** Non line-of-sight

**ns** nanosekund,  $10^{-9}$  sekund

**OLSR** Optimized Link State Routing Protocol

**SDR** Software Defined Radio

**SNR** Signal-to-Noise ratio

**UHF** Ultra High Frequency, 300 MHz - 3 GHz

**UniK** Universitetsstudiene på Kjeller

**USB** Universal Serial Bus

**VHF** Very High Frequency, 30 MHz - 300 MHz

**WLAN** Wireless Local Area Network



# Tillegg B

## Kalibrering

### B.1 Korreksjoner

I forbindelse med prosjektoppgaven ble det utført en kalibrering av systemet. Denne gikk i prinsippet ut på å kartlegge isolasjonen mellom de ulike fysiske forbindelsene i emulatoren, for å se i hvor stor grad signalene smitter over fra en forbindelse til en annen. I emulatorsystemet opplever vi ikke en "ekte" flerbaneeffekt, men vi får signalkomponenter som tilbakelegger ulik banelengde gjennom systemet grunnet ufullstendig isolasjon. Som omtalt tidligere i dette kapitlet er tidsdifferansen ved mottak her så liten at vi kan se bort fra den. Det vi derimot må ta høyde for er bidraget i Signal-to-Noise ratio (SNR) fra de ulike forbindelsene.

I tillegg til dette er det slik at de ulike komponentene i emulatoren introduserer innsettsdempning<sup>1</sup>, slik at hvis vi setter en dempeverdi "0" på et dempeledd så vil ikke det være den faktiske dempningen i systemet. Den totale dempningen over en forbindelse kaller vi forbindelsesdempning. Som omtalt i [8], kan vi klassifisere forbindelsene våre som direkteforbindelser, ettstegs viaforbindelser og tostegs viaforbindelser. For helt åpne forbindelser<sup>2</sup> får vi dermed:

<i>Forbindelse</i>	<i>Teoretisk minimum kanaldempning</i>
Direktekanal	49,5 dB
Ettstegs viakanal	98 dB
Tostegs viakanal	155,5 dB

Tabell B.1: Tabell over innsettsdempningene i emulatoren

Teoretisk maksimal kanaldempning i de 3 ulike forbindelsestypene vil være innsettsdempningen fra tabell B.1, pluss det som dempeleddet selv kan introdusere, 127 dB. For lukket kanal får vi da:

Fra disse tabellene kan vi se at vi har behov for å utføre en kalibrering av verdiene for

---

<sup>1</sup>Minste mulige dempning en komponent introduserer i forbindelsen

<sup>2</sup>Programmerbart dempeledd satt til "0"

<i>Forbindelse</i>	<i>Teoretisk maksimal kanaldempning</i>
Direktekanal	176,5 dB
Ettstegs viakanal	225 dB
Tostegs viakanal	282,5 dB

Tabell B.2: Tabell over maksimal forbindelsesdempning i emulatoren

å nivåere de enkelte forbindelsene. I tillegg til dette er det slik at når utregningene våre sier at vi skal ha en kanaldempning på for eksempel 30 dB, så blir det ikke nødvendigvis riktig å sette 30 dB på dempeleddet direkte. Systemet må ta høyde for at vi uansett har ca 49 dB dempning innebygd. Sistnevnte er gjort på en relativt enkel måte ved at brukeren har mulighet til å velge en korrigering av dette slik at det effektivt sett trekkes fra 49 dB på verdien som settes på kanalene. Svakheten med denne løsningen er at ved små innbyrdes avstander mellom nodene vil dempningen ligge på 0 dB. Denne korreksjonen gjøres ikke før etter utregning av kanalverdiene slik at selve utregningen er den samme.

I de direkte forbindelsene mellom nodene har vi kommet frem til verdier for når vi akkurat har en åpen kanal, når den er delvis åpen, og når den er lukket. Åpen innebærer høy SNR og et signal som står entydig over støygulvet. Delvis betyr at signalet tidvis er over, og tidvis drukner i støyen. Lukket er verdier hvor SNR er lav og det er umulig å skille signalet fra støygulvet. I sistnevnte tilfelle går signalet tapt i sin helhet. Verdiene for de ulike forbindelsenes direktekanaler er:

<i>Forbindelse</i>	<i>Lukket</i>	<i>Delvis</i>	<i>Åpen</i>
R1-R2	112	100	95
R1-R3	112	100	95
R1-R4	112	105	95
R1-R5	115	107	97
R1-R6	115	107	97
R2-R3	115	107	97
R2-R4	115	107	97
R2-R5	115	107	97
R2-R6	115	107	97
R3-R4	115	107	97
R3-R5	115	107	97
R3-R6	115	107	97
R4-R5	115	107	97
R4-R6	115	107	97
R5-R6	111	105	95

Tabell B.3: Dempeverdier for kanalkvaliteter ved direkteforbindelser

Når det gjelder isolasjonen mellom de ulike forbindelsene har vi fra [8] en tabell med

trygge verdier over alle forbindelser. Disse er funnet ved å koble en sender på R1 og en mottaker på R2. Den direkte forbindelsen mellom R1 og R2 er satt til maksimal dempning, og de trygge verdiene er valgt slik at de er gyldige for viaforbindelser over hver og én av de andre forbindelsene. Vi får da verdier som gjengitt i tabell B.4.

<i>Forbindelse</i>	<i>Lukket</i>	<i>Delvis</i>	<i>Åpen</i>
R1-R2	26	20	14
R1-R3	28	22	14
R1-R4	32	28	16
R1-R5	34	26	22
R1-R6	34	26	22
R2-R3	34	28	16
R2-R4	32	26	16
R2-R5	32	26	16
R2-R6	34	26	16
R3-R4	32	26	14
R3-R5	30	22	14
R3-R6	32	24	16
R4-R5	34	26	16
R4-R6	34	26	16
R5-R6	36	26	18

Tabell B.4: Dempeverdier for kanalkvaliteter ved viaforbindelser

Korreksjonen for å nivellere de enkelte forbindelsene baserer seg på datagrunnlaget over, og formaliserer seg i form av en  $[15 \times 1]$ -matrise med en korreksjonsverdi (i desibel) for den enkelte forbindelse. Denne matrisen forventer systemet å finne i mappen `C:\Topologiemulator\Scenarier`, og med navnet `Kalibreringsmatrise.txt`. Brukeren får et valg mellom å nytte systemets kalibreringsverdier, legge inn egne verdier, eller nytte sist brukte matrise. Sistnevnte alternativ er et hensiktsmessig valg dersom egne verdier nyttes og gjentatte scenarier kjøres.

Eksempel på en slik kalibreringsmatrise er i figur B.1

-2
-2
-2
0
0
0
0
0
0
0
0
0
0
0
0
0
-2

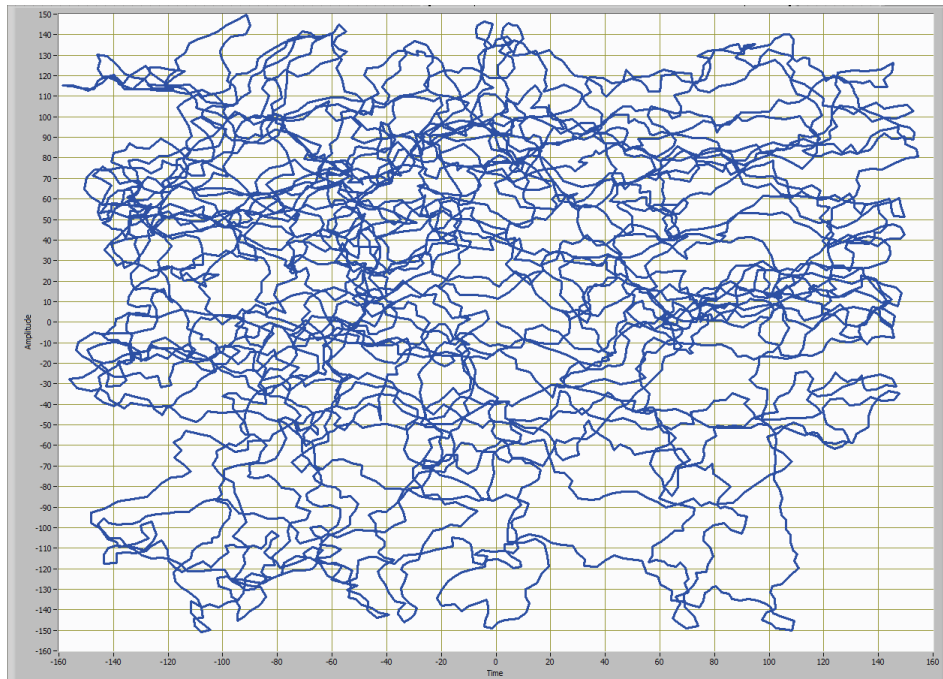
Figur B.1: Eksempel på kalibreringsmatrise

## Tillegg C

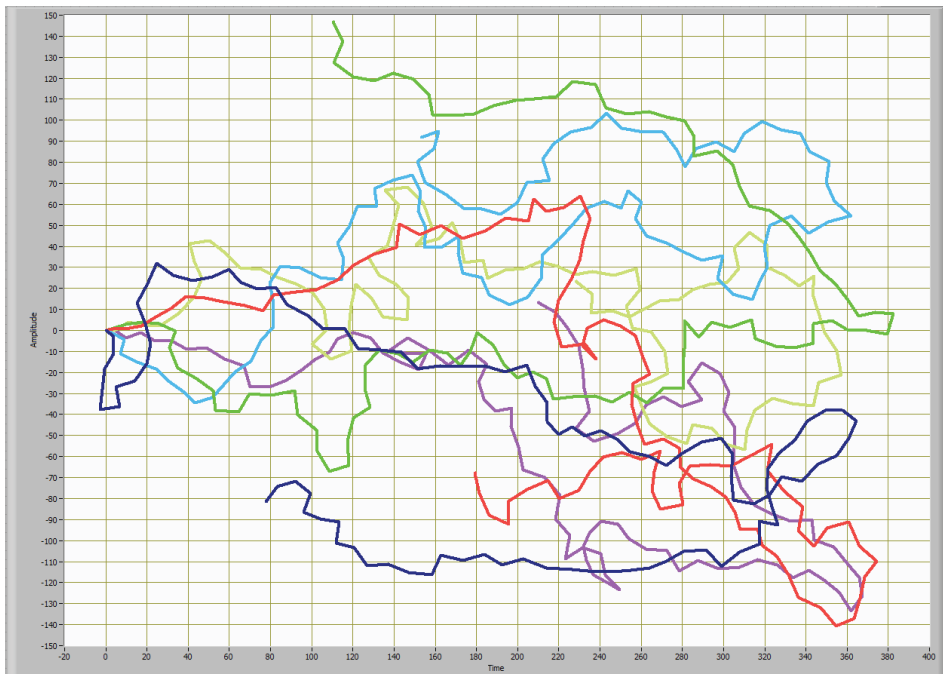
# Eksempler fra mobilitetsmodell



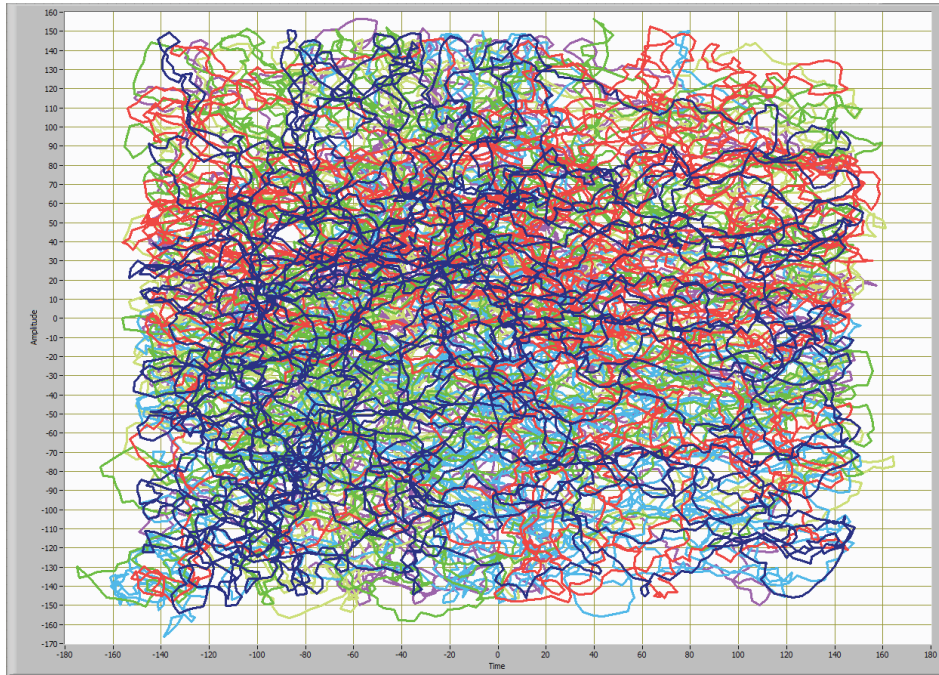
Figur C.1: Bevegelse med Gauss-Markov modell, én node, 100 iterasjoner



Figur C.2: Bevegelse med Gauss-Markov modell, én node, 10000 iterasjoner



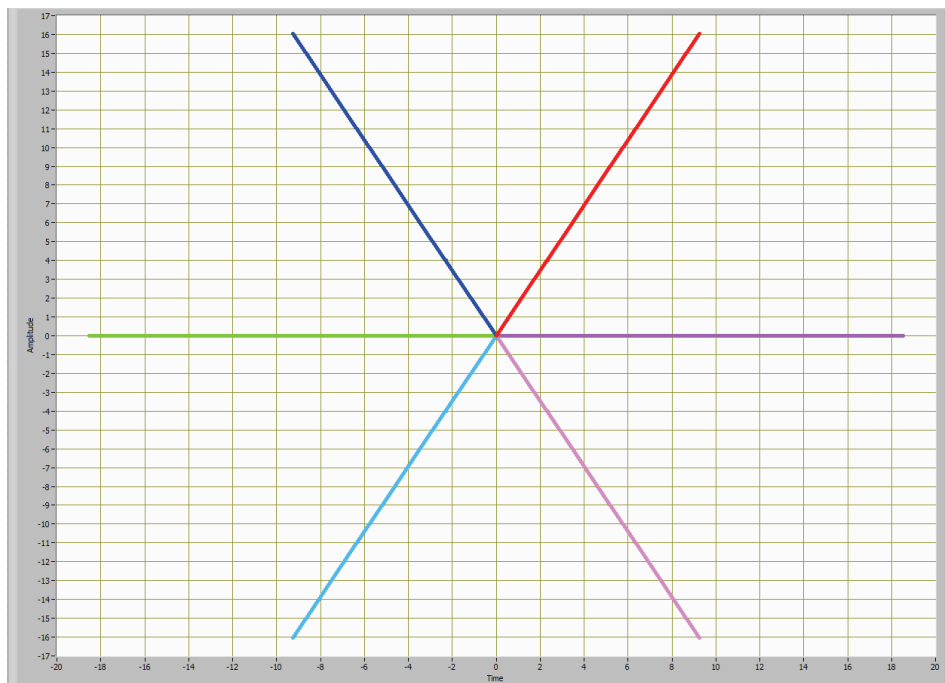
Figur C.3: Bevegelse med Gauss-Markov modell, seks noder, 100 iterasjoner



Figur C.4: Bevegelse med Gauss-Markov modell, seks noder, 5000 iterasjoner



Figur C.5: Bevegelse med stjernemodell, tre noder, 100 iterasjoner, 20 meter ut maks

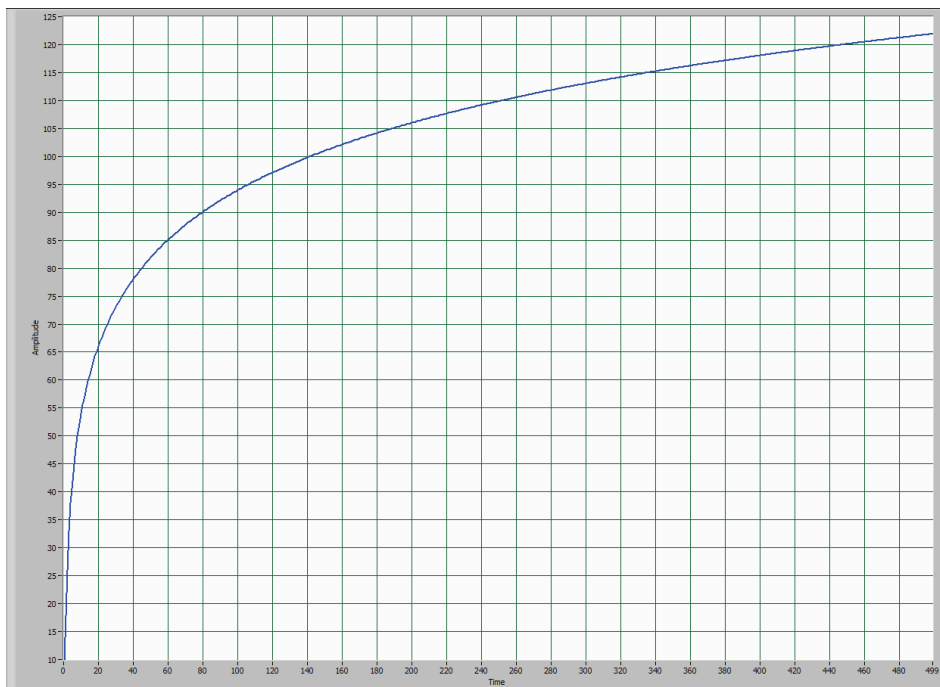


Figur C.6: Bevegelse med stjernemodell, seks noder, 100 iterasjoner, 20 meter ut maks



## Tillegg D

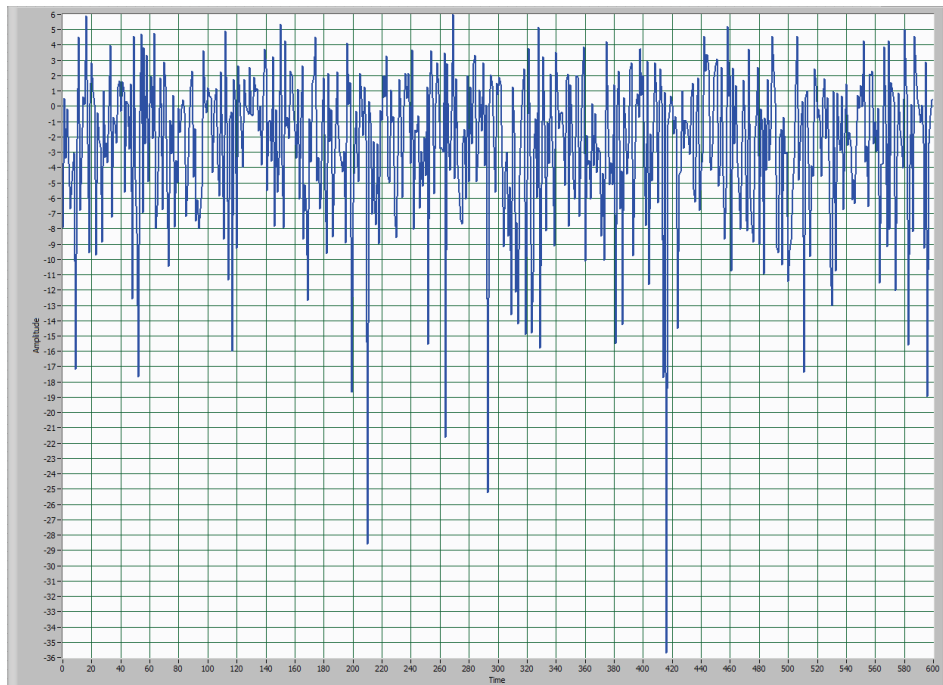
# Eksempel fra kanalmodul



Figur D.1: Banetap over 500 meter



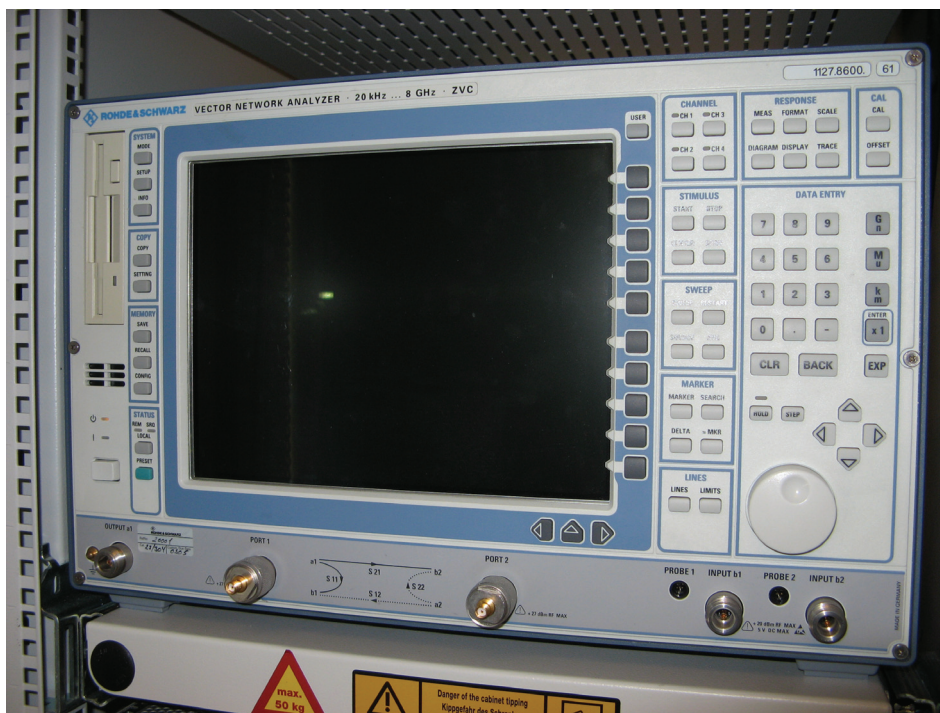
Figur D.2: Banetap over 500 meter, med skyggevirksomheter



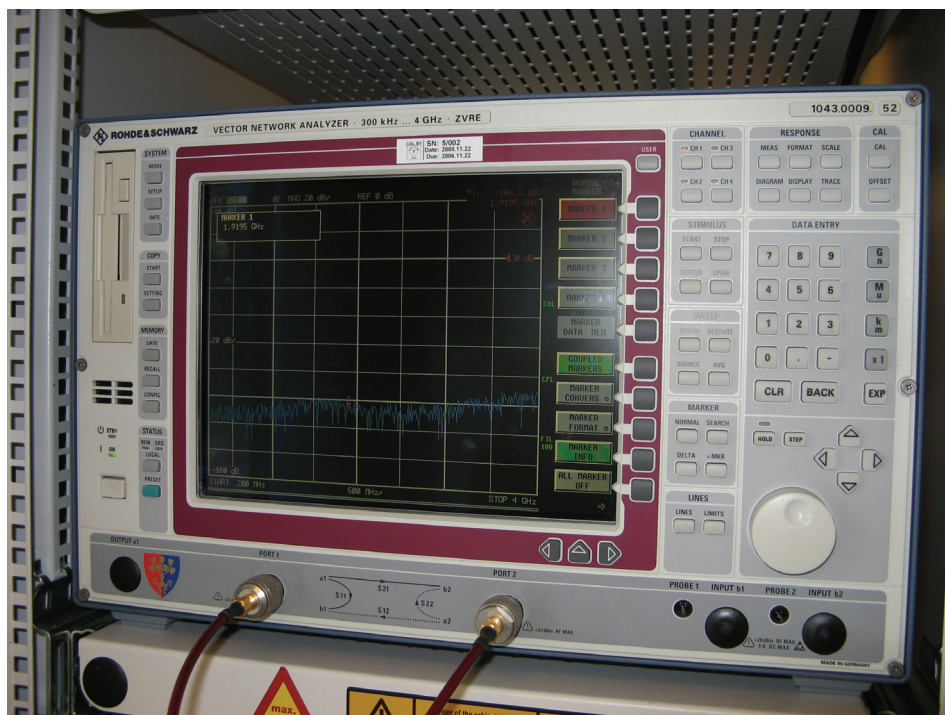
Figur D.3: Sammensetning av Rayleigh og Rice verdier

## Tillegg E

# Instrumenter for signalgenerering og analyse



Figur E.1: Nettverksanalysator Rohde & Schwarz ZVC



Figur E.2: Nettverksanalysator Rohde & Schwarz ZVRE