

Nye øvre skranker på ytelsen til adaptiv  
koding og modulasjon i OFDM-baserte  
trådløse nettverk



Trond Rognebakke Bjørstad  
Institutt for Informatikk  
Universitetet i Bergen

20. januar 2004



# Forord

Jeg vil med dette takke min veileder, Kjell Jørgen Hole, for all hjelp han har gitt meg i løpet av hovedfaget. Han har alltid vært tilgjengelig når jeg har trengt assistanse. Han har besvart faglige spørsmål og vært til stor nytte i diskusjon om oppgavens oppbygning og innhold. Han har vært en nødvendighet for min fullføring av oppgaven.

Jeg vil også takke mine medstudenter for hjelp med L<sup>A</sup>T<sub>E</sub>X og andre tekniske utfordringer. Videre vil jeg takke dem for å ha gjort min tid som hovedfagstudent til en positiv opplevelse.

Til slutt vil jeg rette en takk til mine nærmeste som har vært til støtte og lagt forhold til rette for meg gjennom hele arbeidsperioden.

Takk!

# Innhold

<b>Forord</b>	<b>2</b>
<b>Sammendrag</b>	<b>5</b>
<b>1 Introduksjon</b>	<b>6</b>
1.1 Overføring av digital informasjon . . . . .	7
1.2 Overføring av data i trådløse nettverk . . . . .	8
1.3 Definisjon av viktige begreper . . . . .	10
1.3.1 Overføringshastighet . . . . .	10
1.3.2 Bitfeilrate . . . . .	10
1.3.3 Kanalkapasitet . . . . .	10
1.3.4 Båndbredde . . . . .	11
1.3.5 Spektral effektivitet . . . . .	11
1.4 Modell for trådløse nettverk . . . . .	11
1.5 Anvendelser på modellen . . . . .	12
1.6 Egne nye resultater . . . . .	13
<b>2 Konvertering av digital informasjon til analoge signaler</b>	<b>14</b>
2.1 Amplitudemodulasjon . . . . .	15
2.2 Faseskiftnøkling . . . . .	16
2.3 Kvadraturamplitudemodulasjon . . . . .	17
<b>3 Kanalmodell for en enkel flat-fadende AWGN kanal</b>	<b>20</b>
3.1 Initielt om modellen . . . . .	21
3.2 Hva skjer med signalet underveis? . . . . .	22
3.2.1 Additiv hvit gaussisk støy (AWGN) . . . . .	22
3.2.2 Stitap (path loss) . . . . .	22
3.2.3 Flerveis-fading . . . . .	23
3.3 Mottatt signal . . . . .	24
3.4 Båndbredde . . . . .	25
3.5 Kanalkapasitet . . . . .	26

3.5.1	Kanalkapasitet ved MLD . . . . .	26
3.5.2	Kanalkapasitet ved sekvensiell dekoding . . . . .	29
<b>4</b>	<b>Adaptiv koding og modulasjon</b>	<b>32</b>
4.1	Prinsipp . . . . .	32
4.2	Bestemmelse av fadingregioner . . . . .	34
4.3	Beregning av gjennomsnittlig spektral effektivitet for en flat-fadende AWGN kanal . . . . .	35
<b>5</b>	<b>Utvidelse av kanalmodellen til et OFDM system</b>	<b>40</b>
5.1	Generelt om OFDM . . . . .	41
5.2	OFDM subkanalmodell . . . . .	44
5.3	Adaptiv koding og modulasjon i OFDM-modellen . . . . .	46
5.4	Spektral effektivitet . . . . .	48
5.5	Et praktisk OFDM-system: HIPERLAN/2 . . . . .	52
5.5.1	Litt om protokollstabelen . . . . .	52
5.5.2	Ytelsessimuleringer for HIPERLAN/2-modell . . . . .	53
5.5.3	Forbedringer av HIPERLAN/2 . . . . .	62
<b>6</b>	<b>Oppsummering</b>	<b>64</b>
6.1	Videre arbeid . . . . .	65
	<b>Notasjonsindeks</b>	<b>68</b>
	<b>Bibliografi</b>	<b>70</b>
<b>A</b>	<b>Programkode</b>	<b>74</b>
A.1	Programkode kapittel 2 . . . . .	75
A.2	Programkode kapittel 3 . . . . .	80
A.3	Programkode kapittel 4 . . . . .	94
A.4	Programkode kapittel 5 . . . . .	106

# Sammendrag

I denne hovedfagsoppgaven blir det sett på ytelsen til radiobaserte trådløse nettverk. Det blir bygget opp en modell for en kommunikasjonskanal med parametere som modellerer egenskaper fra en virkelig kanal. Kanalen blir antatt å være en *båndbreddebegrenset flat-fadende AWGN-kanal*. Kanalkapasiteten til modellen blir beregnet for dekodningsteknikkene *Maximum Likelihood Decoding* og *Sekvensiell Dekoding*. Videre blir konseptet *Adaptiv Koding og Modulasjon* (AKM) forklart og den spektrale effektiviteten til kanalmodellen med dette konseptet blir beregnet. Etter dette blir modellen utvidet til et OFDM-system. OFDM er en teknikk som blant annet brukes i trådløse høyhastighetsnettverk, og kanalmodellen blir brukt som en modell for en subkanal i denne teknikken. Det blir vist hvordan AKM kan benyttes i OFDM-systemet og deretter beregnet *øvre skranke for spektral effektivitet til OFDM/AKM/QAM-systemet*. Det blir også laget en modell av den OFDM-baserte kommunikasjonsstandarden HIPERLAN/2 og beregnet spektral effektivitet for denne. En sammenligning av resultatene viser at *ytelsen til HIPERLAN/2 kan forbedres betraktelig ved å benytte AKM slik som det er beskrevet i denne oppgaven*.

# Kapittel 1

## Introduksjon

Sending, overføring og mottaking av signaler fra et punkt til et annet går under betegnelsen telekommunikasjon. Signalene kan inneholde informasjon som tegn, skrift, bilder og lyd og kan transporteres gjennom tråd, radio, eller optiske systemer. Telekommunikasjonens historie går tilbake til begynnelsen av 1800-tallet da det ble forsket på områder innenfor elektrisitet og elektromagnetisme. Gjennombruddet var et faktum da Samuel F.B. Morse bygde verdens første telegraflinje i 1843. Året 1875 ble også en stor milepæl da Graham Bell oppfant telefonen. Siden kom det innslag av teknologi som radiotelegrafi, radiotelefoni og teleks. Det vi kjenner som Internett har sitt utgangspunkt i forskningsnettverket ARPANET som startet på slutten av 1960-tallet. Dette nettet ble gradvis utvidet ettersom nye institusjoner ønsket å koble seg til. På 1970-tallet ble det utviklet lokale nettverk og i samme tidsperiode ble det standardisert overføringsprotokoller som gjorde det mulig å koble dem sammen. Denne teknologien ble innført i ARPANET i 1983 og resultatet ble et nett av nettverk, kalt Internett. I begynnelsen av 1990-årene ble Internett gjort offentlig og åpnet for kommersielle tjenester. Det verdensomspennende mobiltelefonnettet som vi bruker i dag, GSM, ble tatt i bruk i 1993. Omtrent samtidig ble det hypertextbaserte informasjonssystemet *World Wide Web* lansert som bidro til at Internett fikk en eksplosiv vekst [1, 2].

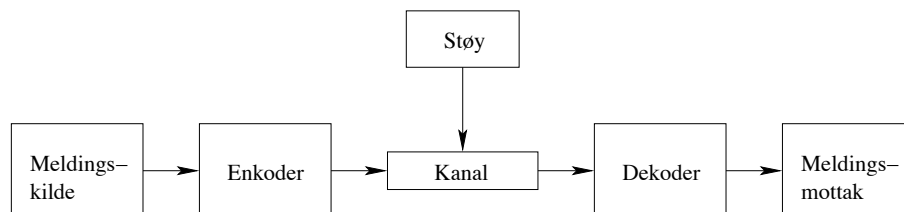
Telekommunikasjon er et spennende fagfelt. Telefon- og datanettverk tillater informasjonsutveksling uavhengig av geografisk nærhet og har revolusjonert hverdagen for mange. I et større perspektiv kan telekommunikasjon sies å være en teknologi som har påvirket internasjonal politikk, handel og kulturforståelse.

For ikke å gape over for mye vil denne hovedfagsoppgaven gå inn på et avgrenset område innenfor telekommunikasjon kalt *radiobaserte trådløse nettverkssystemer*. Det vil bli utviklet modeller for slike systemer og sett

på hvilke egenskaper disse har. Bakgrunnsteori og mer om oppbyggingen av oppgaven vil bli presentert i dette introduksjonskapittelet.

## 1.1 Overføring av digital informasjon

Læren om overføring av informasjon mellom en avsender og en mottaker kalles for informasjonsteori. Harry Nyquist og Ralph V. L. Hartley var pionerer innenfor informasjonsteorien på 1920-tallet da de presenterte de første matematiske beskrivelsene av informasjonsoverføring. Dette emnet ble senere behandlet av Claude Shannon som i 1948 publiserte *The Mathematical Theory of Communication* [3]. Arbeidet til Shannon regnes i dag for å være grunnlaget for den moderne informasjonsteorien.



Figur 1.1: Shannons kommunikasjonsmodell

Shannon presenterte en abstrakt kommunikasjonsmodell, se figur 1.1, som er anvendelig i mange kommunikasjonssystemer. Den første komponenten i modellen, meldingskilden, er entiteten som lager meldingen. Et eksempel på en meldingskilde kan være en person som snakker eller en datamaskin som genererer en informasjonsstrøm. Enkoderen er objektet som konverterer meldingen til et signal som kan sendes. Kanalen er mediet som bærer signalet og kan være f.eks. kobbertråder, fiberoptiske kabler eller luft som radiobølger går gjennom. Når signalet går over kanalen kan det bli utsatt for støy. Støy forstyrrer signalet slik at det forandrer karakter. Dekoderens oppgave er å omforme det mottatte signalet slik at det får samme form som den opprinnelige meldingen. Meldingsmottaket er entiteten som mottar meldingen, for eksempel en person eller en datamaskin [2].

For å overføre digital informasjon må enkoderen først konvertere informasjonen til analoge signaler. Denne teknikken kalles modulasjon og kapittel 2 gir en innføring i hvordan dette gjøres.

Et mye brukt konsept for å modellere signalpåvirkning er additiv hvit gaussisk støy (eng. *Additive White Gaussian Noise, AWGN*). En ser da på støy som en gaussisk prosess med uniform spektral effekttetthet (eng. *Power*



*Spectral Density*) som lineært adderes til signalet. Denne adderingen medfører at innholdet i meldingen forandres før den når mottakeren.

For at meldingen skal være forståelig selv om den påvirkes av støy kan det legges til ekstra kontrollinformasjon. Dette må gjøres av enkoderen før den sender signalene videre og kalles koding. En vanlig kodeteknikk er blokkoding. En enkoder for en blokkode deler en informasjonssekvens inn i meldingsblokker med fast størrelse. Denne størrelsen blir her betegnet ved  $u$  biter. Deretter avbildes hver meldingsblokk i en ny sekvens med  $v$  biter kalt et kodeord. For  $2^u$  mulige binære meldingsblokker finnes det altså  $2^u$  tilhørende kodeord av lengde  $v$ . Mengden av disse  $2^u$ -te kodeordene kalles for en  $(v, u)$ -blokkode. Forholdet  $R = u/v$  er koderaten og angir antall informasjonsbiter som går inn i enkoderen per antall biter som sendes ut av den. For å ha en nyttig binær kode må det tilordnes forskjellige kodeord til forskjellige meldingsblokker. Dette betyr at en må ha  $u \leq v$  slik at koderaten oppfyller  $R \leq 1$ .

Avhengig av kodesystemet en bruker vil det være mulig å detektere eller korrigere feil som er oppstått på grunn av støy under overføringen. Denne delen av prosessen utføres av dekodeheten. Generelt kan en detektere og korrigere flere feil jo flere kontrollbiter en bruker. For en grundigere innføring i kodeteknikker se [4, 5].

## 1.2 Overføring av data i trådløse nettverk

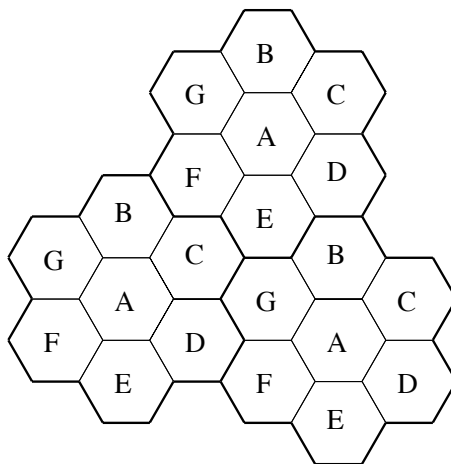
Trådløse nettverkssystemer [6, 7] har mange fordeler fremfor trådbaserte. Mobile systemer tillater kommunikasjon med omverden for eksempel når en er på reise. Trådløse nettverkssystemer kan også være mer kostnadseffektive da en slipper installasjon av kabler. Overføringshastighet på nye systemer er dessuten slik at det kan være et funksjonelt alternativ til trådbaserte lokalnettverk. Av natur er det likevel knyttet noen ekstra egenskaper til trådløse systemer som en må ta hensyn til. For eksempel er de sårbare for avlytting. I motsetning til trådbaserte systemer hvor en fysisk må koble seg til en kabel, kan en spion enkelt, uten å bli oppdaget, tappe informasjon dersom han befinner seg innenfor rekkevidden til radiosignalene. Det må derfor benyttes kryptering for å hemmeligholde informasjon som overføres i trådløse systemer.

Trådløse nettverk er også utsatt for støy. Radiosignaler fra andre systemer kan interferere med signalene som sendes. Signalene kan også påvirkes av stråling fra eksempelvis strømledninger og elektriske installasjoner. Et tiltak som kan redusere støyproblemer er frekvenshopping. Da byttes sendefrekvens med jevne mellomrom slik at tiden en sender på frekvenser med mye støy minimeres.

Et annet problem som kan oppstå oftere i trådløse nettverk enn i trådbaserte nettverk er intersymbolinterferens. Radiosignalene kan reflekteres mot gjenstander og vandre ulike veier før de ankommer mottakeren med forskjellig forsinkelse. Et forsinket signal vil kunne forstyrre mottaket av et signal sendt etter det forsinkete signalet.

Det elektromagnetiske frekvensbåndet er en begrenset ressurs. I Norge har Post- og teletilsynet [8] ansvar for å følge opp og lage nasjonale frekvensplaner. Det er ikke tillatt å ta i bruk radiosystemer som sender utenom godkjente frekvensområder. Forskjellige systemer har fått tildelt hver sine frekvensområder for at de ikke skal forstyrre hverandre.

Innenfor et nettverkssystem med flere brukere kan det også være nødvendig med frekvensplanlegging [7]. Tidlig tankegang innenfor radiobasert telekommunikasjon gikk ut på å bruke høyt monterte antenner med stor sendeeffekt. En oppnådde med dette stor rekkevidde, men det medførte at det var umulig å gjenbruke samme frekvens innenfor dekningsområdet. Løsningen på problemet var å dele geografiske områder inn i celler. I det cellulære konseptet benyttes sendere med lav sendeeffekt i hver celle. Ved å la naboceller bruke forskjellig frekvens kan det foregå kommunikasjon samtidig i cellene uten interferens. Rekkevidden til signalene sendt med lav effekt er forholdsvis kort, og dette åpner for at frekvenser kan gjenbrukes i andre celler. Figur 1.2 viser et eksempel på frekvensgjenbruk i det cellulære konseptet. Bokstavene indikerer frekvensene som benyttes og ulike bokstaver betyr forskjellige frekvenser. Den minste avstanden mellom to celler med samme frekvens kalles for co-kanal gjenbruksavstand (eng. *co-channel reuse distance*).



Figur 1.2: Cellulært radiosystem med frekvensgjenbruk

## 1.3 Definisjon av viktige begreper

Internettilgang for hjemmebrukere i Norge har tradisjonelt foregått gjennom oppringte telefonlinjer (ISDN eller analoge linjer). I de siste årene har utbredelsen av bredbånd økt og er i ferd med å bli et allemannseie. De vanligste typene bredbånd er varianter av DSL (eng. *Digital Subscriber Line*) og trådløs radiotilgang. Fordelene med det som kalles bredbånd er at man alltid er oppkoblet og ikke faktureres for brukt tid. Dette har gjort bredbånd populært i bedrifter og blant hjemmebrukere. Bredbånd har nærmest blitt et folkebegrep, men på grunn av manglende tekniske kunnskaper har det oppstått en del misforståelser knyttet til det. Blant annet snakker mange om bredden på båndet som et mål for hvor rask tilgangen er til Internett, noe som ikke er helt riktig. For å oppklare dette og andre misforståelser vil det her bli gitt definisjoner av viktige begreper knyttet til kommunikasjonskanaler. Disse definisjonene vil være en nyttig innføring for å lese resten av denne oppgaven.

### 1.3.1 Overføringshastighet

I fagterminologien brukes ordet *rate* i stedet for hastighet. *Datarate* er definert som datamengde overført gjennom en kanal per tidsenhet. Den omfatter alle paritets- og informasjonsbiter som sendes og måles vanligvis i biter/sekund (bps). En kan også snakke om *informasjonsrate*, og den er definert som antall informasjonsbiter som overføres per tidsenhet.

### 1.3.2 Bitfeilrate

Som nevnt kan meldinger forandres når de utsettes for støy. I digitale meldinger vil dette arte seg som databiter som har endret verdi. *Bitfeilrate* er definert som forholdet mellom antall bitfeil som oppstår ved en dataoverføring og totalt antall overførte biter. Bitfeilrate er et gjennomsnittsbegrep og brukes ofte som et sannsynlighetsmål for feilforekomster.

### 1.3.3 Kanalkapasitet

*Kanalkapasiteten* til en kanal med støy er den største informasjonsraten som kan oppnås for en vilkårlig liten bitfeilrate. Shannon beviste at pålitelig kommunikasjon kan oppnås gjennom koding når senderaten er mindre enn kanalkapasiteten [2, 3].

### 1.3.4 Båndbredde

Dersom en snakker om båndbredden til et signal snakker en om hvor stort frekvensområde det opptar, dvs differansen mellom den høyeste og laveste frekvensen som signalet består av. Båndbredden til en overføringskanal er definert som det største frekvensområdet kanalen kan slippe gjennom. Måleenheten for båndbredden er Hertz (Hz).

### 1.3.5 Spektral effektivitet

Spektral effektivitet er en indikator på ytelsen til et radiobasert trådløst nettverk. Den relaterer datagjennomstrømmingen til frekvensspekteret som benyttes. Spektral effektivitet er definert som informasjonsrate per båndbreddeenhet og måles i biter per sekund per Hertz. I betraktning av at båndbredde er en begrenset ressurs, er spektral effektivitet en viktig faktor for å sammenligne effektiviteten til ulike trådløse nettverkssystemer.

## 1.4 Modell for trådløse nettverk

I denne oppgaven vil det bli bygget en modell for trådløse nettverk. Som en innføring til emnet vil det i kapittel 2 bli sett på hvordan digital informasjon kan representeres når en sender det over et radiobasert system. Der vil også bli forklart hva en signalkonstellasjon er og vist en figur med signalkonstellasjonene som brukes i denne oppgaven.

I kapittel 3 blir det presentert en modell for kommunikasjon mellom to enheter. Denne modellen gjelder for en enkel trådløs kanal og vil kun omfatte enveiskommunikasjon, dvs at informasjonen går fra en sender til en mottaker. Det vil bli sett på hvordan signalene påvirkes under transport og hvordan dette kan modelleres matematisk. Videre vil båndbredden bli diskutert før det til slutt vil bli sett på kanalkapasiteten til modellen for to ulike dekodningsteknikker.

I kapittel 4 vil det bli bygget videre på kanalmodellen fra kapittel 3. Hovedkommunikasjonen foregår fortsatt fra en sender til en mottaker, men utvidelsen lar mottakeren få komme med tilbakemeldinger til senderen. Tilbakemeldingene inneholder informasjon om kvaliteten til de mottatte signalene og senderen kan bruke denne informasjonen til å tilpasse modulasjon og koderate etter forholdene på kanalen. Denne teknikken kalles adaptiv koding og modulasjon.

Kapittel 5 vil ta for seg OFDM (eng. *Orthogonal Frequency Division Multiplexing*) som er blitt en populær kommunikasjonsteknikk for høyhastighetsnettverk. Teknikken går ut på å sende flere signaler med lav båndbredde pa-

rallelt over en kanal med stor båndbredde. Det blir bygget videre på modellen som er presentert og den enkle kanalmodellen vil brukes som en modell for en subkanal i OFDM-systemet. Det vil også bli sett på hvordan adaptiv koding og modulasjon kan brukes i OFDM.

## 1.5 Anvendelser på modellen

HIPERLAN/2 [9] er den nye trådløse nettverksstandarden til ETSI (European Telecommunications Standards Institute) [10] som skal erstatte førstegenerasjons trådløse lokalnettverk (WLAN). Den første generasjonen av WLAN-produkter er basert på varianter av IEEE 802.11 [11], deriblant 802.11b, og er tilgjengelig fra mange forskjellige produsenter. Avhengig av sendeteknikk, så tilbyr disse produktene datahastigheter i området fra 1 Mbit/s til 11 Mbit/s. Selv om rekkevidden er en begrensning, er disse produktene anvendelige for en rekke applikasjoner som ikke krever for stor overføringshastighet. Dette, kombinert med et prisfall de siste årene, har gjort førstegenerasjons WLAN til et interessant alternativ til et trådbasert lokalnettverk. Likevel kan disse produktene komme til kort mot krav til større datarater for multimedia og interaktive applikasjoner.

HIPERLAN/2 opererer i 5 GHz båndet og kan tilby datahastigheter på det fysiske nivået helt opp til 54 Mbit/s. Dette er betydelig mer enn førstegenerasjons WLAN, noe som åpner for større funksjonalitet. HIPERLAN/2 støtter data-, tale- og videooverføringer med garanterte verdier for tjenestekvalitet, såkalt *Quality of Service* (QoS), i tillegg til funksjoner som strømsparekontroll, tilgangskontroll og interoperabilitet med andre nettverkstyper gjennom et såkalt *convergence layer* [12].

Kjernen i HIPERLAN/2-standarden ble publisert i år 2000 og i slutten av år 2002 ble det demonstrert fungerende prototyper med denne teknologien. IEEE som utviklet det ledende førstegenerasjonssystemet Wi-Fi [13] har et konkurrerende system til HIPERLAN/2. Dette systemet er 802.11a og er svært likt HIPERLAN/2 på det fysiske nivået. Det opererer i samme frekvensområde og skal også kunne tilby datahastigheter opp til 54 Mbit/s. Hovedforskjellen ligger i metodene for mediumstilgang, der HIPERLAN/2 bruker en sentralisert kontroll, mens 802.11a er basert på konkurranse. Med WI-FI har IEEE vist at de kan produsere gode trådløse nettverkssystemer. Det er et fortrinns som gjør at 802.11a-standardens har gode forutsetninger for å bli en suksess. Simuleringer viser likevel at HIPERLAN/2 gir bedre ytelsesverdier for faktisk datagjennomstrømning enn 802.11a og at IEEE må standardisere support for dynamisk frekvensseleksjon, strømkontroll og prioritetsmekanismer for mediumstilgang for å konkurrere med HIPERLAN/2

ytelsesmessig [14].

Den beste standarden, med tanke på ytelse, av de to nevnte andregenerasjons WLAN-systemene er altså HIPERLAN/2. Det vil derfor bli anvendt parametere fra HIPERLAN/2 på modellen som blir bygget opp i denne oppgaven. Deretter vil ytelsen til HIPERLAN/2 bli analysert ved hjelp av en egen modell. Motivet for dette er å undersøke om adaptiv koding og modulasjon vil kunne forbedre HIPERLAN/2 og å gi en indikasjon på hvor mye det eventuelt er å tjene.

## 1.6 Egne nye resultater

I kapittel 3 beskrives kanalkapasiteten til en båndbreddebegrenset minneløs AWGN-kanal med diskrete inntattverdier og kontinuerlige utputtverdier når det benyttes *Maximum Likelihood Decoding* (MLD) og *sekvensiell dekodning* (SD). Det blir gjort beregninger av kanalkapasiteten til denne kanalen for signalkonstellasjoner som inneholder  $\{4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096\}$  symboler.

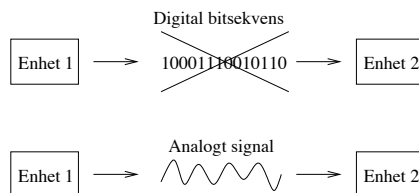
Kapittel 4 tar som nevnt for seg adaptiv koding og modulasjon. Denne teknikken deler mengden av det mulige mottatte signal/støy-forholdet inn i fadingregioner. Med utgangspunkt i kanalkapasiteten for MLD og SD til modellen fra kapittel 3 blir det bestemt terskler for signal/støy-forhold som definerer fadingregionene. Fadingregionegrensene blir bestemt slik at en oppnår forhåndsbestemte spektrale effektivitetsverdier ( $\{1.0, 1.5, 2.0, \dots, 11.5\}$  [biter/sek/Hz]). Deretter blir det beregnet øvre skranke for gjennomsnittlig spektral effektivitet når en benytter adaptiv koding og modulasjon på den presenterte kanalmodellen.

Kapittel 5 utvider modellen til et OFDM-system og viser hvordan adaptiv koding og modulasjon kan benyttes med denne utvidelsen. Kanalkapasitetene fra kapittel 3 og fadingregionegrensene bestemt i kapittel 4 benyttes for å beregne øvre skranke for gjennomsnittlig spektral effektivitet til et OFDM/AKM/QAM-system. Det blir også gjort simuleringer av den spektrale effektiviteten til en modell av HIPERLAN/2 og disse blir sammenlignet det med verdiene for OFDM/AKM/QAM-systemet.

## Kapittel 2

# Konvertering av digital informasjon til analoge signaler

En kan ikke overføre digital informasjon i form av en bitsekvens direkte mellom to forskjellige fysiske enheter. Først må informasjonen konverteres til analoge signaler.



Figur 2.1: Analoge signaler

Hvilken type analoge signaler som blir brukt, avhenger av hvilket medium som blir benyttet. For eksempel blir den digitale informasjonen modulert til lydbølger dersom en bruker et modem for telefonlinjer. Noen systemer bruker lysbølger for å utveksle informasjon. I denne oppgaven skal vi se på trådløse nettverk som benytter radiobølger for å forflytte informasjonen.

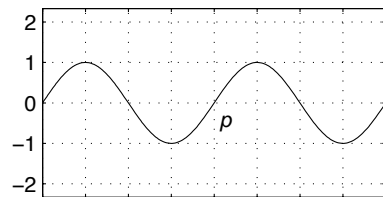
Bæresignalet karakteriseres av gjentatte svingninger, eller pulser, som representerer et digitalt signal. Hver puls kan inneholde informasjon om en eller flere digitale biter. En generell teknikk for å pakke digitale biter inn i et analogt signal kalles kvadraturamplitudemodulasjon (eng. *quadrature amplitude modulation*, ofte forkortet QAM). Under følger en beskrivelse av denne teknikken og hvordan den er bygget opp. Først blir det forklart hva amplitudemodulasjon er. Deretter kommer en seksjon om faseskiftning (eng. *phase shift keying*). Til slutt blir det forklart hvordan de to sistnevnte teknikkene kombineres i kvadraturamplitudemodulasjon.

## 2.1 Amplitudemodulasjon

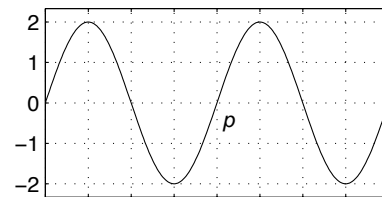
Amplitudemodulasjon er en enkel teknikk som kan brukes til å sende signaler, f.eks. over radio. Moduleringen går ut på å la amplituden til radiobølgen være proporsjonal med signalet en ønsker å sende. Anta at en ønsker å modulere signalet på en cosinusbølge. Ligningen til en slik bærebølge er i [15] oppgitt på formen  $u(t) = A \cos(2\pi f_c t)$ , hvor  $f_c$  er bærefrekvensen, og  $A$  er amplituden som kan varieres.

Dersom en ønsker å lage et system som modulerer binære signaler kan en f.eks. la et signal med amplitude 1 representere en binær 0, og tilsvarende la en binær 1 bli representert ved signal med amplitude 2 (Se (a) og (b) i figur 2.2). Legg merke til at perioden  $p = 1/f_c$  er den samme for begge signalene.

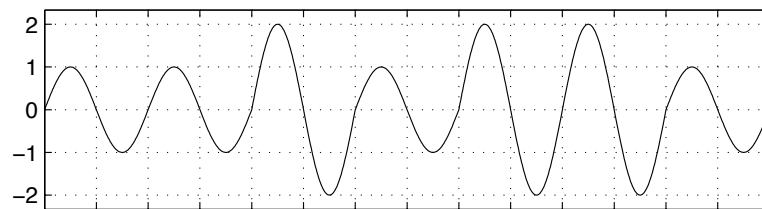
Dersom en videre kombinerer signalene i en sekvens som i figur 2.2 (c), kan man se hvordan en bærebølge i et amplitudemodulasjonssystem ser ut. I dette tilfellet representerer signalsekvensen den binære sekvensen 0010110. Signalet varierer i amplitude, mens perioden til hver puls er lik gjennom hele signalet.



(a) Signal med amplitude 1



(b) Signal med amplitude 2



(c) Kombinasjon av signaler fra (a) og (b)

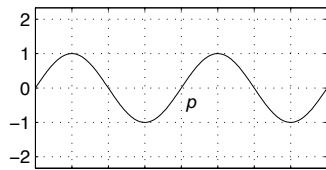
Figur 2.2: Amplitudemodulasjon



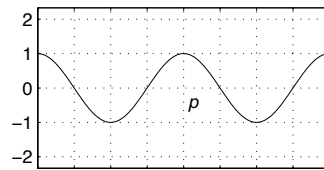
## 2.2 Faseskiftnøkling

Faseskiftnøkling er i likhet med amplitudemodulasjon også veldig enkel i oppbygningen. Teknikken går ut på å endre bærebølgens fase etter hvilket signal en ønsker å sende. Faseskiftnøkling kan modelleres med ligningen  $u(t) = \cos(2\pi f_c t + \theta)$ , hvor  $f_c$  er bærefrekvensen og  $\theta$  er fasen [15].

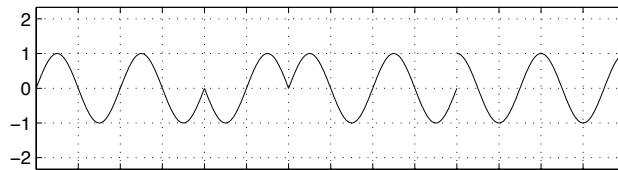
Figur 2.3 (a) og (b) viser to signaler som er forskjøvet 1/4 periode i forhold til hverandre. En kan dele en periode opp i flere deler slik at en får mange ulike signaler å sende, men valg av størrelsen på faseskiftene avhenger av hvilke signalforhold en har. Flere skift krever gode forhold for at signalet skal kunne dekodes.



(a) Vilkårlig signal med amplitude 1



(b) Signal (a) forskjøvet 1/4 periode



(c) Sekvens av signaler med forskjellig fase

Figur 2.3: Faseskiftnøkling

Signalnr.	Bitverdi	Faseskift
0	00	0
1	01	1/4
2	10	1/2
3	11	3/4

Tabell 2.1: Eksempel på faseskiftnøkling med 4 faser

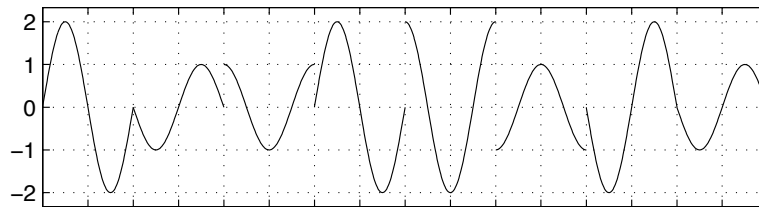
Som et eksempel kan en tenke seg et system med 4 forskjellige faseskift. Hvert signal tilordnes et nummer og en bitverdi som vist i tabell 2.1. Et eksempel på en kombinasjon av faseskiftene fra denne tabellen er vist i figur 2.3 (c). Hvis en ser på bitverdien til hvert signal, så representerer signalsekvensen i denne figuren den binære sekvensen 00 00 10 00 00 01 01.

## 2.3 Kvadraturamplitudemodulasjon

Det generelle modulasjonstilfellet hvor både amplituden og fasen kan endres mellom hvert signal kalles kvadraturamplitudemodulasjon (QAM). Denne teknikken er altså en kombinasjon av amplitudemodulasjon og faseskift-nøkling. Bølgeformen til et QAM-signal uttrykkes i [15] på formen  $u(t) = A \cos(2\pi f_c t + \theta)$ .

Signalnr.	Bitverdi	Amplitude	Faseskift
0	000	1	0
1	001	2	0
2	010	1	1/4
3	011	2	1/4
4	100	1	1/2
5	101	2	1/2
6	110	1	3/4
7	111	2	3/4

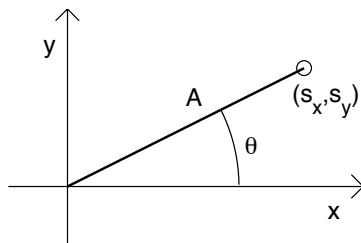
Tabell 2.2: Eksempel på et QAM-system med 8 ulike signaler



Figur 2.4: Kvadraturamplitudemodulasjon

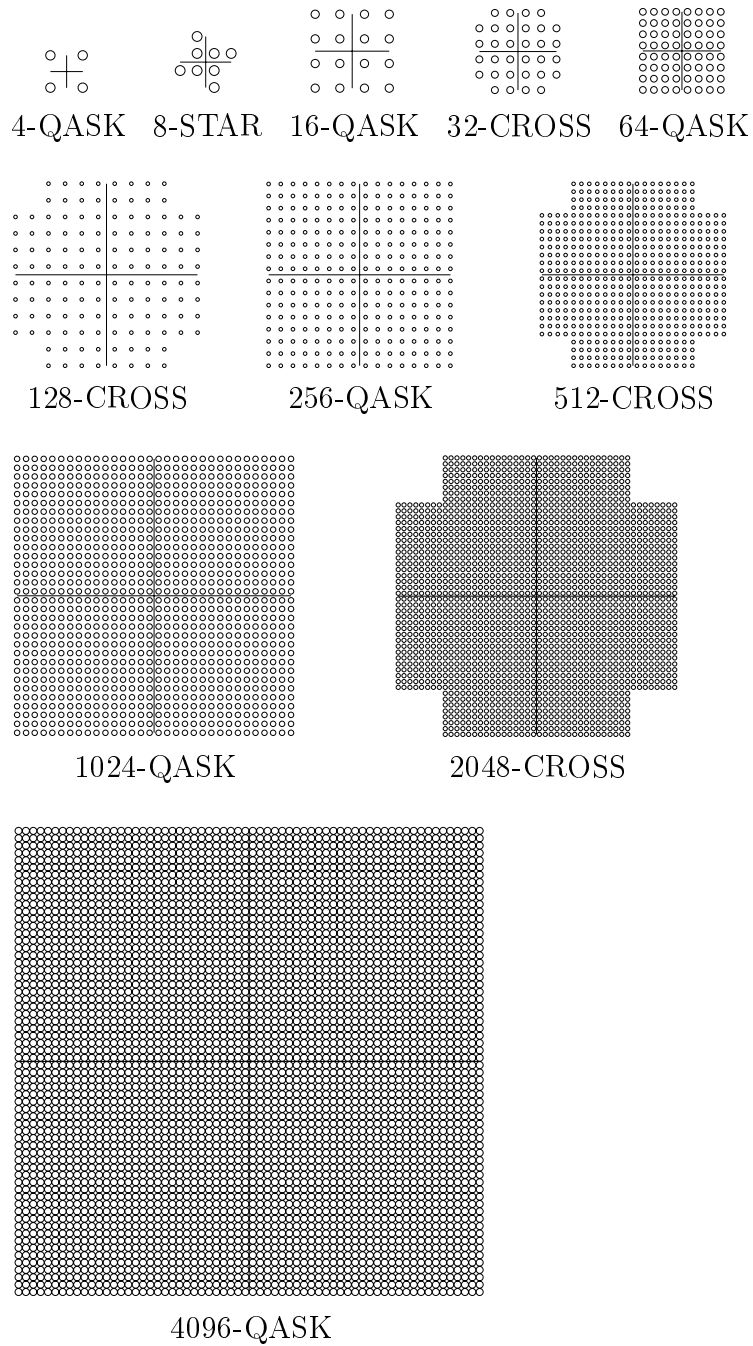
Hvert signal tilordnes en verdi etter dets karakteristika. Tabell 2.2 inneholder et eksempel på et QAM-system med 8 forskjellige signaler. Et eksempel på en sekvens av QAM-signaler finnes i figur 2.4. Den tilsvarer den binære sekvensen 001 100 010 001 011 110 101 100.

En mengde av forskjellige signaler som kan brukes, kalles en signalkonstellasjon. Siden et QAM-symbol kjennetegnes av to verdier, amplituden og fasen, kan en lage en geometrisk representasjon av signalkonstellasjonen [16]. Dette gjøres ved å plote punkter i et todimensjonelt  $xy$ -plan, hvor punktets avstand  $A$  til origo angir amplituden og vinkelen  $\theta$  mellom punktets vektor og  $x$ -aksen angir fasen. Se figur 2.5 for illustrasjon. Alternativt kan en se på et QAM-symbol som et punkt i det komplekse planet. Anta QAM-symbolet  $s = s_x + s_y i$ , eller bare  $s = (s_x, s_y)$ . Amplituden til  $s$  vil da være gitt ved modulus til  $s$  som er  $|s| = |s_x + s_y i| = \sqrt{s_x^2 + s_y^2}$ . Denne måten å representere et QAM-symbol på kalles den komplekse basisbåndrepresentasjonen.



Figur 2.5: Geometrisk representasjon av et QAM-symbol

Figur 2.6 på neste side viser signalkonstellasjonene som benyttes i denne oppgaven. Alle disse konstellasjonene har størrelse  $M = 2^k$  hvor  $k$  er et heltall. Det vil si at hvert QAM-signal representerer  $k$  digitale biter.



Figur 2.6: Signalkonstellasjoner i denne oppgaven

# Kapittel 3

## Kanalmodell for en enkel flat-fadende AWGN kanal

Dette kapittelet presenterer en modell for en enkel flat-fadende AWGN kanal. En kanalmodell er et rammeverk som beskriver egenskapene til kanalen. Det blir ofte gjort en del antagelser og forenklinger for at modellen ikke skal bli for omfattende og kompleks.

Som en introduksjon til studier av kommunikasjonskanaler er det vanlig å studere den ideelle AWGN kanalmodellen. Statistisk uavhengig gaussisk støy korrumpere dataene, og en har ikke intersymbolinterferens (eng. *inter symbol interference*). Det mottatte signalet  $r(t)$  kan med den ideelle AWGN kanalmodellen uttrykkes ved ligningen

$$r(t) = s(t) + w(t) \quad (3.1)$$

der  $s(t)$  er det sendte signalet og  $w(t)$  er støyen ved tidsindeks  $t$ .

Dersom det ikke er spesifisert noe om hvordan radiosignalene forplanter seg, er det vanlig å anta at signalsvekkingen i forhold til tilbakelagt avstand foregår som om signalene er i et ideelt tomt rom. Modellen med tomt rom antar at det mellom sender- og mottakerantennen er fritt for alle objekter som kan absorbere eller reflektere radiosignaler, og at atmosfæren er et perfekt uniformt ikke-absorberende medium. Etter hvert som signalet beveger seg vil signalenergien svekkes. Denne svekkingen følger en kvadratisk faktor,  $L_s(d)$ , som kalles stitap (eng. *path loss*), og som kan uttrykkes ved ligningen [17, (1)]

$$L_s(d) = \left( \frac{4\pi d}{\lambda} \right)^2. \quad (3.2)$$

Her er  $d$  avstanden mellom sender og mottaker og  $\lambda$  er bølgelengden til signalet. I et idealisert tilfelle som dette, vil det være lett å estimere den

mottatte signaleffekten  $P$  [Watt]. Den vil være gitt ved ligningen

$$P = S \left( \frac{\lambda}{4\pi d} \right)^2 \quad (3.3)$$

der  $S$  [Watt] er sendeeffekten og de øvrige parametrene er som i uttrykket over.

Frekvensbåndet som inneholder de frekvenser som finnes i informasjons-signalet, kalles ofte for basisbåndet (eng. *baseband*). Differansen mellom den høyeste og den laveste frekvensen i kanalen kalles for båndbredden, og betegnes ved bokstaven  $B$  [Hz]. Dersom en vil finne ut noe om kvaliteten til kanalen kan det være interessant å se på signal/støy-forholdet. Det kan uttrykkes ved

$$\gamma = \frac{P}{N_0 B} \quad (3.4)$$

der  $N_0$  [W/Hz] er den ensidige spektrale tettheten til støyen [15].

Ved å bruke notasjon som er definert over, kan en uttrykke kapasiteten  $C$  til AWGN-kanalen ved [15, (9.2.17)]

$$C = B \log_2(1 + \gamma) \text{ [info.biter/sek]}. \quad (3.5)$$

Selv om den ideelle kanalmodellen med AWGN i mange tilfeller kan gi gode estimater, vil den nok for de fleste praktiske kanaler bli for enkel. Det vil derfor i resten av kapitlet bli presentert en modell som tar inn over seg flere egenskaper fra virkelige kanaler.

## 3.1 Initielt om modellen

Modellen i dette kapitlet består av en enkelt sender og en enkelt mottaker som kommuniserer over en trådløs kanal. Både senderen og mottakeren er til enhver tid i ro, dvs de er faste installasjoner, og har hver sin antenne. Videre blir det benyttet diskret tid  $t$  for å identifisere ulike signaler som sendes.

Senderen sender signaler  $s(t)$  med kompleks basisbåndsrepresentasjon for  $t \in \{0, 1, 2, \dots\}$ . Signalene går over en intersymbolinterferensfri additiv hvitstøykanal med begrenset båndbredde. Signalet sendes med en effekt som ved tiden  $t$  betegnes ved  $S(t)$  [Watt].

I modellen blir det også antatt at kun koder som gir kanalsignaler med like stor sannsynlighet blir brukt. Dvs at dersom en over tid ser på hvilke signaler som blir sendt, så skal antallet av hvert signal være noenlunde det samme.

## 3.2 Hva skjer med signalet underveis?

Signalene går som oftest gjennom atmosfæren, nær bakken, og blir reflektert underveis. Da blir fritt-rom-modellen ikke god nok til å beskrive kanalen og forhåndsestimere systemytelsen. Foruten AWGN er det i hovedsak tre mekanismer som påvirker signalene, stitap, langsom fading og hurtig fading. I [18] blir det argumentert for at i mange kanaler med fri sikt så er det i praksis umulig å skille mellom langsom og hurtig fading. Parametrene for fading forblir de samme i begge tilfeller. I denne oppgaven velger jeg derfor å se på langsom og hurtig fading representert ved flerveis-fading. I de påfølgende underseksjonene vil det bli gått nærmere inn på hvilken påvirkning AWGN, stitap, og flerveis-fading fører til og hvordan disse kan modelleres.

### 3.2.1 Additiv hvit gaussisk støy (AWGN)

I denne modellen blir det antatt at det sendte signalet  $s(t)$  går gjennom en additiv hvit gaussisk støykanal på veg til mottakeren. Dette betyr at det mottatte signalet  $r(t)$  som blir behandlet, er en sum av det sendte signalet  $s(t)$  og støysignalet  $w(t)$ . Støyen  $w$  er en uavhengig variabel, normalfordelt om 0 og med varians  $\sigma^2$  i hver dimensjon. Signalkonstellasjonene som blir brukt i denne oppgaven (se figur 2.6) er 2-dimensjonale, så derfor vil  $w$  ha varians  $2\sigma^2$ .

### 3.2.2 Stitap (path loss)

Stitap er som tidligere nevnt svekking av signalstyrken etter hvert som signalet forflytter seg. At signalstyrken svekkes kan være et problem med tanke på mottak av signalene, men det er også en svært viktig egenskap f.eks. i cellulære systemer. Hurtig svekking av signalene vil kunne tillate en liten co-kanal gjenbruksavstand. Stitap kalles også for stor-skala fading.

Stitap i et praktisk trådløst system avhenger ikke bare av den tilbakelagte avstanden og bølgelengden, men også av lokale terrengforhold som f.eks. bygninger og fjell. Den stedsavhengige naturen til radioforplanting gjør det vanskelig å lage en teoretisk beregning av stitap, men en enkel modell [19, (1.4)] uttrykker den mottatte signalstyrken  $P$  ved

$$P_{(\text{dB})}(d) = P_{(\text{dB})}(d_0) - 10\beta \log_{10}(d/d_0) \quad [\text{dBm}] \quad (3.6)$$

der  $P_{(\text{dB})}(d_0)$  er den mottatte signalstyrken i en gitt referanseavstand  $d_0$ . Typisk er  $d_0$  1 km for makroceller, 100 m for utendørs mikroceller, og 1 m for innendørs picoceller. Parameteren  $\beta$  kalles stitapsekspONENTEN (eng. *path loss exponent*) og er en nøkkelparameter som påvirker den spektrale

effektiviteten i cellulære systemer. Vanligvis bestemmes den ut fra empiriske målinger, og den ligger stort sett innenfor intervallet fra 2 til 8 [19].

### 3.2.3 Flerveis-fading

Signalet i et trådløst kommunikasjonssystem kan bli reflektert og spredd over multiple stier før det når mottakeren. Dette fenomenet kalles flerveisforplantning og kan forårsake forandringer i det mottatte signalets amplitude, fase og ankomstvinkel. Forandringene i signalet forårsaket av flerveisforplantningen kalles for flerveis-fading. Denne fadingen er små-skala fading og utgjør små hurtige forandringer i signalet, i størrelsesorden ned til en halv bølgelengde, fra det sendes til det mottas.

To viktige scenarioer for flerveis-fading beskrives i de statiske kanalmodellene til henholdsvis Rayleigh og Rice. I Rayleighs modell har en flerveisforplantning av signalene, men en har ingen direkte siktkomponent (eng. *line-of-sight component*) som mottas. Når det finnes en slik direkte komponent kan fadingen beskrives av modellen til Rice. Se figur 3.1 for illustrasjon.

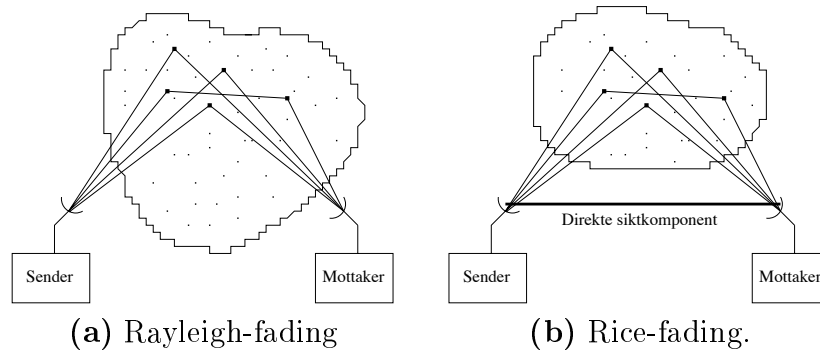
Både Rayleigh- og Rice-modellene har sannsynlighetstetthetsfunksjoner som beskriver hvordan fadingen varierer ved ulike forhold. I stedet for å benytte disse funksjonene direkte, vil det i denne oppgaven benyttes en tetthetsfordeling fra en annen modell, nemlig Nakagamis fading-modell. Nakagami-fordelingen er gitt ved sannsynlighetstetthetsfunksjonen [20, (1)]

$$p_{\alpha}(\alpha) = 2 \left(\frac{m}{\Omega}\right)^m \frac{\alpha^{2m-1}}{\Gamma(m)} \exp\left(-m\frac{\alpha^2}{\Omega}\right), \quad \alpha \geq 0 \quad (3.7)$$

der  $m$  kalles nakagamiparameteren. Parametrene  $\alpha$  og  $\Omega$  defineres i seksjon 3.3. For gammafunksjonen  $\Gamma(m)$  se seksjon 4.3.

Fordelen med å bruke modellen med Nakagami-fading er at den er lett å tilpasse ulike forhold. Ved å la parameteren  $m = 1/2$  får vi en ensidig gaussisk fordeling. For  $m = 1$  blir fordelingen den samme som Rayleighfordeling. Det er også mulig å lage en god approksimasjon til Rice-fordelingen vha Nakagami-funksjonen med  $m > 1$ . Når  $m \rightarrow \infty$  blir fordelingen som på en AWGN-kanal uten fading. En annen fordel med Nakagami-funksjonen er at den ofte resulterer i pene matematiske uttrykk.





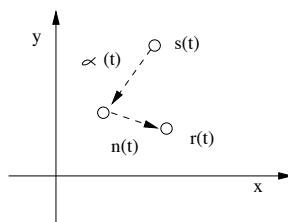
Figur 3.1: Kanalmodeller for flerveis-fading

### 3.3 Mottatt signal

Når en tar hensyn til fading, kan det mottatte basisbåndsignalet uttrykkes ved ligningen

$$r(t) = \alpha(t)s(t) + n(t) \quad (3.8)$$

der  $\alpha(t)$  er fadingfaktor,  $s(t)$  er det sendte basisbåndsignalet og  $n(t)$  er den komplekse additive gaussiske støyen. Fadingfaktoren  $\alpha(t) = \alpha$  i uttrykk (3.7) bare med en ekstra tidsindeksing  $t$ . Både  $\alpha(t)$  og  $n(t)$  vil forflytte basisbåndsignalet i det komplekse planet. Se figur 3.2 for illustrasjon. Den reelle og den imaginære delen av støyen er statistisk uavhengig, hver med varians  $\sigma^2 = (N_0B/2)$ .  $N_0$  er total ensidig spektral tetthet til støyen og  $B$  er båndbredden.



Figur 3.2: Illustrasjon av hvordan det sendte signalet påvirkes

Det instantane mottatte signal/støy-forholdet  $\gamma(t)$  er representert ved den stokastiske variabelen

$$\gamma(t) = \frac{P(t)}{\text{Var}(n)} = \frac{P(t)}{2\sigma^2} = \frac{\alpha^2(t)S(t)}{N_0B}. \quad (3.9)$$

Det er antatt at fadingen på kanalen kan beskrives av Nakagamifordelingen. Ved å gjøre transformasjoner av variabler i uttrykk (3.7) får en at det mottatte signal/støy-forholdet følger gammafordelingen gitt ved

$$p_\gamma(\gamma) = \left(\frac{m}{\bar{\gamma}}\right)^m \frac{\gamma^{m-1}}{\Gamma(m)} \exp\left(-m\frac{\gamma}{\bar{\gamma}}\right), \quad \gamma \geq 0. \quad (3.10)$$

Det forventede gjennomsnittlige mottatte signal/støy-forholdet er også interessant. La  $\bar{S}$  betegne gjennomsnittlig sendt signaleffekt. La videre  $\Omega = E[\alpha^2]$  være gjennomsnittlig mottatt fadingeffekt. Siden  $p_\gamma(\gamma)$  er en kontinuerlig funksjon på intervallet  $[0, \infty)$ , kan en uttrykke det forventete signal/støy-forholdet ved

$$\bar{\gamma} = E[\gamma] = \frac{\Omega \bar{S}}{N_0 B}. \quad (3.11)$$

### 3.4 Båndbredde

La  $T$ [s] være symbol lengden i tid, dvs tiden som er satt av til å sende et symbol. Alle komponenter som mottas innenfor signalintervallet  $T$  summeres sammen og danner det mottatte basisbåndsignalet. Båndbredden  $B$ [Hz] til kanalen sier hvor mange symboler en kan sende per tid og uttrykkes ved

$$B = \frac{1}{T}. \quad (3.12)$$

Siden det sendte signalet blir reflektert over mange stier, vil det mottas signalkomponenter med ulike forsinkelser (eng. *delay spread*) [7]. Denne forsinkelsen vil heretter bli kalt *flerveisforsinkelse*. La  $T_{maks}$ [s] være maksimal flerveisforsinkelse blant de mottatte komponentene. Et mottatt signal kan teoretisk bli påvirket av  $T_{maks}/T$  tidligere signaler.

*Koherensbåndet* er et frekvensområde der to frekvenskomponenter har stor korrelasjon i amplitude- og faseoppførsel. Koherensbåndbredden antas å være omvendt proporsjonal med flerveisforsinkelsen og la derfor  $B_k \approx \frac{1}{T_{maks}}$  være en approksimasjon på koherensbåndbredden. Siden kanalen er antatt å være uten intersymbolinterferens må  $T_{maks} < T$ . Da blir

$$B = \frac{1}{T} < \frac{1}{T_{maks}} \approx B_k \quad (3.13)$$

og kanalen sies da å ha ikke-frekvensselektiv fading, eller såkalt *flat fading*.

## 3.5 Kanalkapasitet

I denne seksjonen vil det bli sett på kapasiteten til kanalmodellen ved bruk av to forskjellige dekodningsteknikker. Funksjonen til en dekodeer er å estimere de opprinnelige informasjonsbitene fra et kodet signal. Signalene kan være påvirket av støy, og dekodeeren bruker regler eller metoder for å detektere og korrigere feil som har oppstått. Det finnes mange slike teknikker og de varierer i kompleksitet. Desto mer komplekse og tunge de er, jo lengre tid vil dekodningen ta, og det vil påvirke ratene for hvor hurtig en kan overføre data.

I den første underseksjonen vil det bli sett på kapasiteten ved *soft Maximum Likelihood Decoding (MLD)*. For å bruke denne teknikken kreves det at en har en modulasjonskanal med diskrete innputtverdier og kontinuerlige utputtverdier. Dette passer bra for denne modellen som tar inn QAM-signaler med ulike, men faste verdier. Utputt fra kanalen vil være signaler av nesten vilkårlig karakter påvirket av fading og støy. I den neste underseksjonen vil det bli sett på kanalkapasiteten i forbindelse med sekvensiell dekodning. I begge tilfeller vil det bli antatt normalisert gjennomsnittlig sendeeffekt  $\bar{S} = 1$ .

### 3.5.1 Kanalkapasitet ved MLD

*Soft Maximum Likelihood Decoding* vha Viterbi-algoritmen er en populær dekodningsteknikk som brukes til dekodning av trelliskoder [21, 22].

Ungerboeck oppgir i [23] kanalkapasiteten ved MLD for en minneløs kanal når en bruker koder med uniform fordeling av innputtsignaler til

$$C^* = \log_2(M_n) - \frac{1}{M_n} \cdot \sum_{k=0}^{M_n-1} E \left\{ \log_2 \sum_{i=0}^{M_n-1} \exp \left[ -\frac{|s^k + w - s^i|^2 - |w|^2}{2\sigma^2} \right] \right\} \quad (3.14)$$

der  $M_n$  er størrelsen på signalkonstellasjon  $n$  som benyttes. Videre er  $s^k$  og  $s^i$  henholdsvis det  $k$ -te og  $i$ -te basisbåndsignalet fra denne signalkonstellasjonen, og  $w$  er uavhengig normalfordelt støy med varians  $\sigma^2$  i hver dimensjon. Merk at i dette uttrykket vil støyvariabelen  $w$  variere for hver summeringsiterasjon  $(i, k)$ .

Uttrykket for  $C^*$  ser i første omgang ut som et litt tungt uttrykk, i hvertfall hvis en skal regne på det symbolsk, men det lar seg fint gjøre å estimere verdier for denne kanalkapasiteten på en datamaskin. En ser at uttrykket blant annet inneholder en forventningsverdi. La  $\phi_n$  være denne forventningen for den  $n$ -te konstellasjonen, dvs

$$\phi_n = E \left\{ \log_2 \sum_{i=0}^{M_n-1} \exp \left[ -\frac{|s^k + w - s^i|^2 - |w|^2}{2\sigma^2} \right] \right\}. \quad (3.15)$$

En kan estimere forventningen  $\phi_n$  ved å finne gjennomsnittet av  $l$  uavhengige estimater, dvs

$$\phi_n = \frac{\phi_n^1 + \phi_n^2 + \dots + \phi_n^l}{l} \quad (3.16)$$

der  $\phi_n^j$  er den  $j$ -te estimatet av det indre uttrykket. En kan derfor skrive om uttrykket for  $C^*$  til formen

$$C^* = \log_2(M_n) - \frac{1}{M_n} \cdot \sum_{k=0}^{M_n-1} \frac{1}{l} \cdot \sum_{j=0}^{l-1} \left\{ \log_2 \sum_{i=0}^{M_n-1} \exp \left[ -\frac{|s^k + w - s^i|^2 - |w|^2}{2\sigma^2} \right] \right\} \quad (3.17)$$

hvor  $E\{..\}$  har blitt skiftet ut med  $\frac{1}{l} \cdot \sum_{j=0}^{l-1} \{..\}$ . Dette innebærer at støyvariabelen  $w$  også vil avhenge av summeringsiterasjon  $j$ .

Et moment som er viktig å huske på når en skal beregne dette uttrykket er at både støyen  $w$  og basisbåndsignalene  $s^k$  og  $s^i$  er komplekse. Det vil si at verdiene har to koordinater og kan skrives på formen  $w = (w_x, w_y)$ ,  $s^k = (s_x^k, s_y^k)$  og  $s^i = (s_x^i, s_y^i)$ . For hver iterasjon  $(i, j, k)$  i uttrykk (3.17) må det genereres en ny støyprøve  $(w_x, w_y)$ . Denne kan simuleres med en generator for tilfeldige tall som gir tallene normalfordelt som utputt. Videre betegner  $|..|$  i uttrykk (3.17) absoluttverdi. Som vist i seksjon 2.3 kan en se på absoluttverdien til et komplekst signal som avstanden fra punktet til origo. Benytter en dette får en med litt mellomregning at deluttrykket  $|s^k + w - s^i|^2 - |w|^2$  kan skrives som

$$|s^k + w - s^i|^2 - |w|^2 = (s_x^k + w_x - s_x^i)^2 + (s_y^k + w_y - s_y^i)^2 - w_x^2 - w_y^2.$$

Med dette på plass kan en skrive uttrykket for kanalkapasiteten  $C^*$  på formen

$$C^* = \log_2(M_n) - \frac{1}{M_n} \cdot \sum_{k=0}^{M_n-1} \frac{1}{l} \cdot \sum_{j=0}^{l-1} \left\{ \log_2 \sum_{i=0}^{M_n-1} \exp \left[ -\frac{(s_x^k + w_x - s_x^i)^2 + (s_y^k + w_y - s_y^i)^2 - w_x^2 - w_y^2}{2\sigma^2} \right] \right\}. \quad (3.18)$$

Det ble utviklet en algoritme som beregner kanalkapasiteten  $C^*$  for de ulike konstellasjonene i denne oppgaven. Algoritmen beregner summene i uttrykket ved hjelp av *for*-løkker som går utenpå hverandre. Dette medfører en kompleksitet av størrelsesorden  $\mathcal{O}(M_n \times l \times M_n) = \mathcal{O}(l \times M_n^2)$ . Videre beregner algoritmen kanalkapasiteten kun for de verdiene av signal/støyforholdet som spesifiseres. En angir ytterpunktene til signal/støyforholdet

og hvor store steg det skal være mellom hvert punkt. Kompleksiteten til algoritmen avhenger altså av kvadratet av størrelsen på signalkonstellasjonen, antall ganger en må iterere for å lage en god verdi for  $\phi_n$  og antall punkter det beregnes en verdi for.

Først ble algoritmen implementert i verktøyet *MAPLE* [24], men det viste seg å fungere dårlig. Det ble benyttet en innebygd funksjon til å generere tilfeldige tall. Den var imidlertid treg og ble en altfor stor flaskehals, så løsningen ble å implementere algoritmen i verktøyet *MATLAB* [25]. Det fungerte mye bedre, men algoritmen var fortsatt tung å kjøre. Det medførte at for store  $M_n$  måtte  $l$  reduseres. Verdiene for  $l$  som ble brukt er vist i tabell 3.1. Hvorfor disse verdiene er valgt som de er, vil bli forklart mer om i kapittel 4, seksjon 4.2. For å få kjørt de største beregningene ble det benyttet distribuert prosessering, dvs at problemet ble splittet opp og flere maskiner ble tatt i bruk samtidig. Den mest krevende beregningen, dvs den for  $M_n = 4096$  hadde en kjøretid på ca. 300 timer på en 1 GHz Intel Pentium 3 prosessor. Da ble det benyttet ca 40 slike maskiner som hver kjørte beregninger i ca 8 timer. Resultat av beregningene er vist i figur 3.3 som stiplede linjer.

$n$	$M_n$	$l$	$l \times M_n^2$	# punkter	# kjøring
1	4	$10^6$	$\sim 2 \times 10^7$	41	4
2	8	$10^6$	$\sim 6 \times 10^7$	43	4
3	16	$10^5$	$\sim 3 \times 10^7$	51	4
4	32	$10^5$	$\sim 1 \times 10^8$	55	4
5	64	$10^5$	$\sim 4 \times 10^8$	61	4
6	128	$10^5$	$\sim 2 \times 10^9$	67	4
7	256	$10^4$	$\sim 7 \times 10^8$	73	4
8	512	$10^4$	$\sim 3 \times 10^9$	79	4
9	1024	$10^3$	$\sim 1 \times 10^9$	81	1
10	2048	$10^3$	$\sim 4 \times 10^9$	81	1
11	4096	$10^3$	$\sim 2 \times 10^{10}$	84	1

Tabell 3.1: Simuleringsdata i forbindelse med beregning av  $C^*$ .

### 3.5.2 Kanalkapasitet ved sekvensiell dekoding

Sekvensiell dekoding (SD) kan være å foretrekke fremfor MLD pga av at metoden har lavere kompleksitet og dekodingstid. En algoritme for sekvensiell dekoding er *Fano algoritmen* [26], som på grunn av dekodingshastighet ofte blir valgt i praktiske implementasjoner.

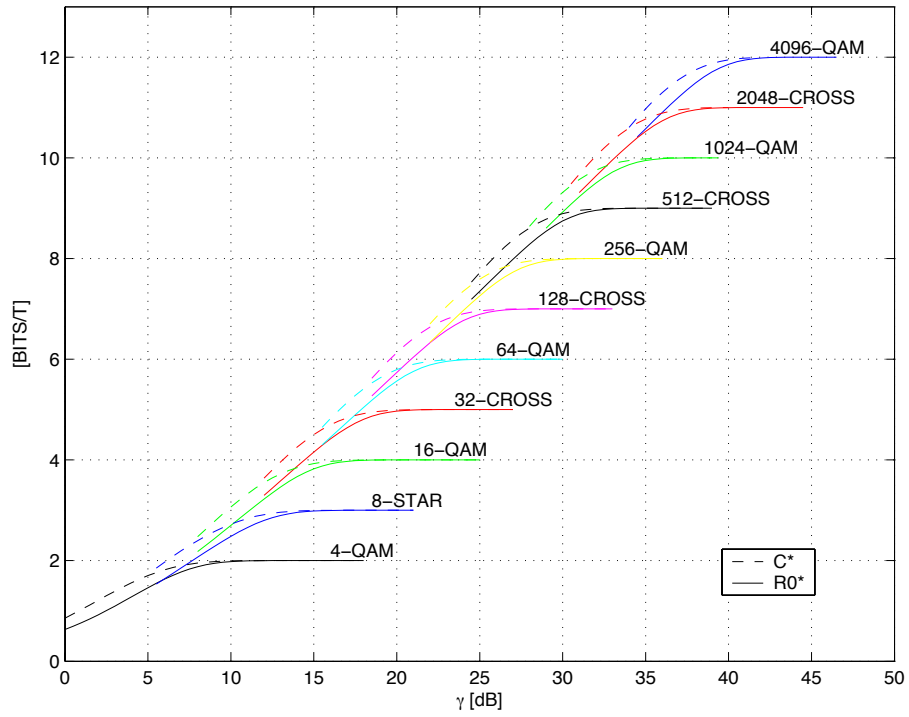
Pålitelig overføring av data kan kun gjøres når senderaten er mindre enn kanalkapasiteten  $C$ . Shannon beviste at  $C$  kun kan oppnås med uendelig kodekompleksitet. For sekvensiell dekoding sies avkuttingsraten (eng. *cutoff rate*)  $R_0$  å være den største raten der gjennomsnittlig antall beregninger er begrenset. Avkuttingsraten  $R_0^*$  for en båndbreddebegrenset AWGN-kanal med diskrete inngangsverdier og kontinuerlige utgangsverdier er i [27] oppgitt til å være

$$R_0^* = 2 \log_2(M_n) - \log_2 \left\{ \sum_{i=0}^{M_n-1} \sum_{j=0}^{M_n-1} \exp \left[ -\frac{(s_x^i - s_x^j)^2 + (s_y^i - s_y^j)^2}{8\sigma^2} \right] \right\} \quad (3.19)$$

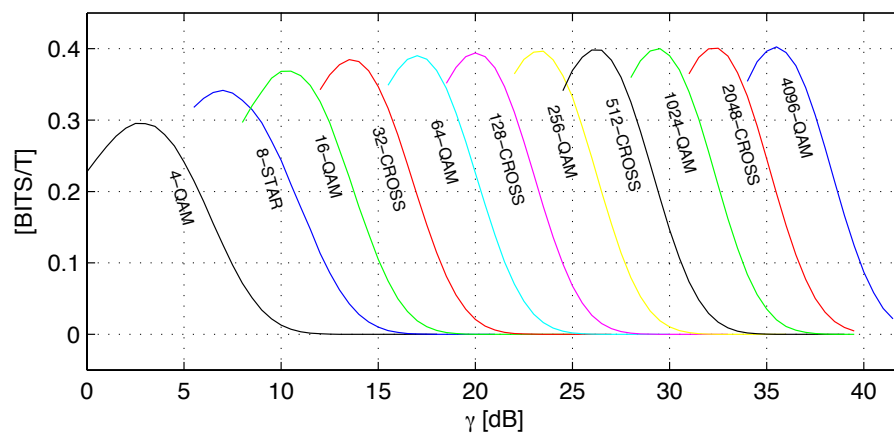
der  $(s_x^i, s_y^i)$  er koordinatene til det komplekse basisbåndsignalet  $s^i$ .

Det er lett å beregne avkuttingsraten  $R_0^*$  på en datamaskin. Det ble implementert en algoritme der summene beregnes på samme måte som ved MLD, nemlig ved å bruke to for-løkker som går utenpå hverandre. Det gir en kompleksitet av størrelsesorden  $\mathcal{O}(M_n^2)$ . Det er en faktor  $l$  mindre enn ved MLD, noe som er betydelig siden  $10^3 \leq l \leq 10^6$ . Utregningene av  $R_0^*$  gikk derfor fort og resultatet er plottet som heltrukne linjer i figur 3.3.

En ser ved sammenligning at for sekvensiell dekoding så ligger avkuttingsraten noe under den tilsvarende kanalkapasiteten for MLD. Likevel er det ikke snakk om den store forskjellen. Differansen mellom  $C^*$  og  $R_0^*$  er plottet i figur 3.4 og en ser at for de signalkonstellasjonene som er brukt er forskjellen mindre enn 0,4 [biter/T]. Det kan derfor være interessant å bruke sekvensiell dekoding i enkelte tilfeller.



Figur 3.3: Kanalkapasiteten  $C^*$  plottet med stiplede linjer og avkuttingsraten  $R_0^*$  plottet med heltrukne linjer.



Figur 3.4: Differansen mellom  $C^*$  og  $R_0^*$





# Kapittel 4

## Adaptiv koding og modulasjon

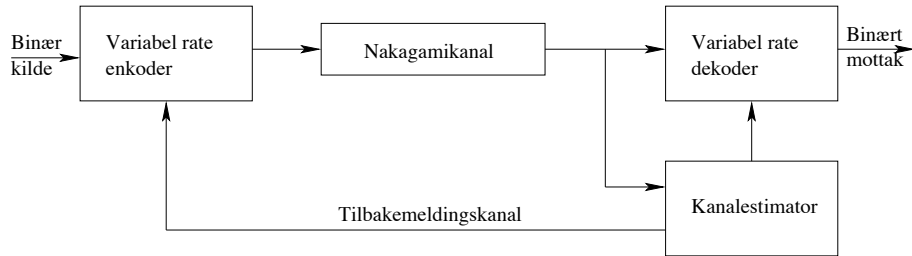
For å sende data med høy hastighet over en fadingkanal kreves en effektiv kommunikasjonsteknikk. Sendeteknikker som ikke adapterer til fadingkanalen krever robust koding som gir akseptabel ytelse selv ved mye fading. Dvs at disse systemene må være konstruert for den verste mulige kanaltilstanden, noe som gjør at de ikke utnytter hele kanalkapasiteten når signal/støy-forholdet er godt. En metode for å adaptere til fadingkanalen kan være å justere sendeeffekten etter forholdene slik at signalet mottas med en ønsket effekt [28], men det vil ikke bli sett på her. I denne oppgaven vil det kun bli sett på adaptering av koder og modulasjon, dvs at forskjellige koder og konstallasjonsstørrelser brukes ved ulike signal/støy-forhold.

Det vil bli tatt utgangspunkt i den enkle flat-fadende AWGN kanalmodellen fra kapittel 3. Ut fra simuleringene som ble gjort for kanalkapasitetene ved MLD og SD vil det bli bestemt fadingregioner for ulike koder. Det vil også utvikles nye skranker for spektral effektivitet til modellen.

### 4.1 Prinsipp

Ideen bak *adaptiv koding og modulasjon* (AKM) [20, 29, 30] er å tilpasse det sendte signalet etter hvor mye støy det er på kanalen. Det må gjøres estimater av signal/støy-forholdene. Når forholdene er gode brukes koder med høye informasjonsrater og når de er dårlige brukes koder med lavere informasjonsrater og med mer vekt på koding. La kode  $n$  basere seg på QAM-konstellasjonen med  $M_n$  symboler. For hver tidsindeks  $k = L \cdot t \in \{0, L, 2L, \dots\}$  tar enkoderen for kode  $n$  i mot  $L \cdot \log_2(M_n) - c$  informasjonsbiter der  $c$  og  $L$  er små positive heltall. Ut fra dette genererer enkoderen  $L \cdot \log_2(M_n)$  kodete biter som spesifiserer  $L$  modulasjonssymboler fra den  $n$ -te QAM-konstellasjonen.

For å estimere signal/støy-forholdet må det først sendes pilotsymboler



Figur 4.1: Systemmodell med adaptiv koding og modulasjon

over kanalen. Pilotsymboler er forhåndsbestemte symboler som er avtalt mellom sender og mottaker. Det blir antatt ideell kanalestimering og koherent deteksjon, dvs at mottakeren kan kompensere for variasjoner i det mottatte signalets amplitude og fase. Mottakeren sammenligner det mottatte signalet med det forhåndsbestemte symbolet og lager deretter et estimat av signal/støy-forholdet. Dette estimatet må til slutt sendes tilbake til senderen over en tilbakemeldingskanal som i denne modellen antas å være støyfri og uten forsinkelse. Da har senderen perfekt kjennskap til det instantane mottatte signal/støy-forholdet og kan justere sine signaler etter det. Figur 4.1 viser systemmodellen med adaptiv koding og modulasjon.

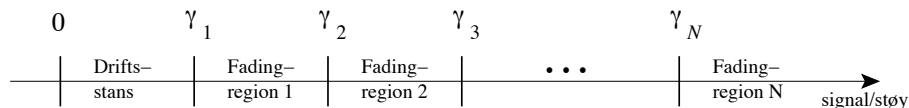
Mengden av mulige mottatte signal/støy-forhold deles opp i  $N$  deler som kalles fadingregioner. De  $N$  fadingregionene defineres ved tersklene  $0 < \gamma_1 < \gamma_2 < \dots < \gamma_{N+1} = \infty$ . Se figur 4.2 for illustrasjon. For hver fadingregion blir det brukt en egen trelliskode konstruert for å bekjempe AWGN og kode  $n$  benyttes når det instantane mottatte signal/støy-forholdet  $\gamma$  er i region  $n$ , dvs når  $\gamma_n \leq \gamma < \gamma_{n+1}$ . La  $p(\gamma)$  være sannsynlighetsfordelingsfunksjonen for signal/støy-forholdet  $\gamma$ . Antar at  $p(\gamma)$  er en kontinuerlig funksjon på intervallet  $[0, \infty)$ . Sannsynligheten for at det instantane signal/støy-forholdet  $\gamma$  havner i fadingregion  $n$  er

$$P(\gamma_n, \gamma_{n+1}) = \int_{\gamma_n}^{\gamma_{n+1}} p(\gamma) d\gamma. \quad (4.1)$$

Fadingregion  $n = 1$  representerer de minste verdiene for  $\gamma$  hvor informasjon sendes [20]. Når  $\gamma < \gamma_1$  blir det ikke sendt noe informasjon fordi kanalkvaliteten rett og slett er for dårlig for de tilgjengelige kodene, og sannsynligheten for det er

$$P(0, \gamma_1) = \int_0^{\gamma_1} p(\gamma) d\gamma = 1 - \int_{\gamma_1}^{\infty} p(\gamma) d\gamma. \quad (4.2)$$

Når dette inntreffer har en altså driftsstans (eng. *outage*).



Figur 4.2: Fadingregioner

## 4.2 Bestemmelse av fadingregioner

Fadingregionene kan settes opp på flere ulike måter, men i denne oppgaven blir fadingregionene bestemt slik at en er garantert en forhåndsbestemt spektral effektivitet. Den spektrale effektiviteten er gitt som informasjonsrate per båndbredde. Med utgangspunkt i kapasitetsratene som ble beregnet for MLD og SD kan en også beregne den spektrale effektiviteten til MLD og SD ved forskjellige  $\gamma$ . Det ble bare beregnet punktvis verdier for  $C^*$  og  $R_0^*$ , så det trengs et estimat av rateverdiene utenfor disse.

Et polynom er et godt valg for å approksimere en glatt (eng. *smooth*) funksjon lokalt. Men hvis funksjonen skal approksimeres på et større intervall kan det hende at graden på det approksimerende polynomet må velges uakseptabelt stort. Et alternativ er å dele intervallene i tilstrekkelige små delintervaller der funksjonen approksimeres med et polynom av en lavere grad på hvert delintervall. Dette kan gjøres slik at det blir myke overganger mellom polynomene og slik at den sammensatte approksimerte funksjonen er mange ganger kontinuerlig deriverbar. En slik glatt stykkesvis polynomfunksjon kalles for et *spline*.

*MATLAB* har flere innebygde splinefunksjoner. I beregningene som ble gjort ble det brukt minste kvadraters kurvetilpasning. En implementasjon av det finnes i *MATLAB*-funksjonen *spap2*, og den ble benyttet til å estimere kontinuerlige funksjoner. For hver konstellasjon ble det beregnet et spline som en tilpasning til datapunktene som ble funnet ved simuleringene av  $C^*$  og  $R_0^*$ . Etterpå ble det utført et intervallhalveringssøk på splinet etter signal/støyforholdet som gav ønsket spektral effektivitet.

Antall informasjonsbiter per QAM-symbol er  $\log_2(M_n) - c/L$ . Tiden det tar å sende et QAM-symbol er  $T$  [s], noe som betyr at informasjonsraten til kode  $n$  er  $R_n = (\log_2(M_n) - c/L)/T$ . Da kan den spektrale effektiviteten til den  $n$ -te koden uttrykkes ved [30]

$$\frac{R_n}{B} = \frac{(\log_2(M_n) - c/L)/T}{1/T} = \log_2(M_n) - c/L \quad [\text{biter/sek/Hz}]. \quad (4.3)$$

Det ble beregnet grenser for fadingregionene for  $c = 1$  og  $L = \{1, 2\}$ . Da får en fadingregiongrenser som garanterer spektral effektivitet med terskelsteg på 0.5, dvs for  $R_n/B = \{1.0, 1.5, 2.0, \dots, 11.5\}$ . Tabell 4.1 viser resultatet av disse

beregningene. Grensene for  $\gamma_n^{R_0^*}$  var rett frem å bestemme. De er vist med 3 signifikante desimaler og ble beregnet fra den ene kurveapproximasjonen til avkuttingsraten  $R_0^*$ . Verdien for  $\gamma_n^{C^*}$  var litt mer tidkrevende å få med en viss grad av nøyaktighet. For å få det til ble beregnet flere verdier for  $\gamma_n^{C^*}$  med forskjellige simuleringsserier som utgangspunkt. For en fiksert verdi av  $l$  ble først det gjort uavhengige simuleringer av kanalkapasiteten  $C^*$ . Dersom de resulterende  $\gamma_n^{C^*}$  varierte i 2. desimal ble det gjort nye simuleringer av  $C^*$  med en større fiksert  $l$ . Dette ble gjentatt inntil verdiene for  $\gamma_n^{C^*}$  var like til og med den andre desimalen. Dette ble gjort for  $n = \{1, 2, \dots, 8\}$ , og en kan derfor si at disse er signifikante med 2 desimaler. For  $n = \{9, 10, 11\}$  derimot ble det på grunn av tidskompleksitet kun gjort en enkelt beregning av  $\gamma_n^{C^*}$ . Hvilke verdier for  $l$  som måtte brukes og hvor mange uavhengige simuleringer av  $C^*$  det ble gjort for den verdien av  $l$  er vist i tabell 3.1.

### 4.3 Beregning av gjennomsnittlig spektral effektivitet for en flat-fadende AWGN kanal

Spektral effektivitet er som tidligere nevnt definert som informasjonsrate per båndbredde. Dvs at den er et mål på hvor stor datagjennomstrømmingen er i forhold til tilgjengelig båndbredde. Det gjør den til en velegnet parameter for å sammenligne ulike trådløse systemer med hverandre. Ved å gjøre en vektet sum over den spektrale effektiviteten til hver kode i forhold til sannsynligheten for at koden benyttes, får man et mål på den gjennomsnittlige ytelsen til AKM på den enkle kanalmodellen [20]. Med andre ord kan en uttrykke den gjennomsnittlige spektrale effektiviteten (eng. *Average Spectral Efficiency* (ASE)) til den enkle flat-fadende AWGN-kanalmodellen med AKM ved

$$\text{ASE}(\gamma_n) = \sum_{n=1}^N \frac{R_n}{B} \cdot P(\gamma_n, \gamma_{n+1}). \quad (4.4)$$

Sannsynligheten for at  $\gamma$  havner i den  $n$ -te fadingregionen er [20, (10)]

$$P(\gamma_n, \gamma_{n+1}) = \frac{\Gamma(m, \frac{m\gamma_n}{\bar{\gamma}}) - \Gamma(m, \frac{m\gamma_{n+1}}{\bar{\gamma}})}{\Gamma(m)} \quad (4.5)$$

der  $\bar{\gamma}$  er gjennomsnittlig mottatt signal/støy-forhold,  $\Gamma(\cdot, \cdot)$  er den komplementære ukomplette gammafunksjonen [31, (8.350.2)] og  $\Gamma(m)$  er gammafunksjonen [31, (8.310)]. Gammafunksjonen er definert som

$$\Gamma(m) = \int_0^\infty \exp^{-t} t^{m-1} dt \quad (4.6)$$

$n$	$M_n$	$L$	$R_n/B$ [bits/s/Hz]	$\gamma_n^{C^*}$ [dB]	$\gamma_n^{R_0^*}$ [dB]
1	4	1	1.0	0.83	2.478
		2	1.5	3.64	5.205
2	8	1	2.0	6.14	7.680
		2	2.5	8.55	10.083
3	16	1	3.0	9.77	11.096
		2	3.5	11.76	13.155
4	32	1	4.0	13.12	14.405
		2	4.5	14.99	16.361
5	64	1	5.0	16.55	17.811
		2	5.5	18.37	19.741
6	128	1	6.0	19.59	20.817
		2	6.5	21.36	22.714
7	256	1	7.0	22.86	24.087
		2	7.5	24.60	25.934
8	512	1	8.0	25.79	27.030
		2	8.5	27.53	28.856
9	1024	1	9.0	29.0*	30.215
		2	9.5	30.7*	32.066
10	2048	1	10.0	31.9*	33.135
		2	10.5	33.6*	34.909
11	4096	1	11.0	35.1*	36.272
		2	11.5	36.8*	38.111

Tabell 4.1: Terskler  $\gamma_n^{C^*}$  og  $\gamma_n^{R_0^*}$  for fadingregioner for MLD og SD med  $c = 1$ . Felter som er markert med \* er av usikker nøyaktighet.

og er lik  $(m - 1)!$  når  $m$  er et positivt heltall. Videre er den komplementære ukomplette gammafunksjonen definert som

$$\Gamma(m, x) = \int_x^\infty \exp^{-t} t^{m-1} dt. \quad (4.7)$$

For å beregne gjennomsnittlig spektral effektivitet for MLD og SD er det da bare å bruke formlene over og sette inn de respektive fadingregion-grensene. For å korte ned programkoden kan det være lurt å bruke ferdige implementasjoner av gammafunksjonene. I *MAPLE* finnes funksjonene implementert som  $GAMMA(m) = \Gamma(m)$  og  $GAMMA(m, x) = \Gamma(m, x)$ . Det er også mulig å bruke *MATLAB* som tilbyr funksjonene  $gamma(m) = \Gamma(m)$  og  $gammainc(x, m) = \frac{1}{\Gamma(m)} \cdot \int_0^x \exp^{-t} t^{m-1} dt$ . Den komplementære ukomplet-

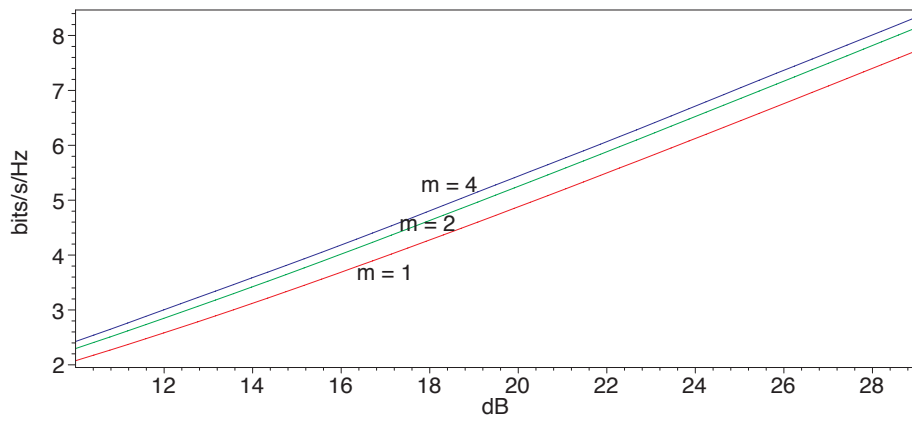
te gammafunksjonen kan uttrykkes ved disse funksjonene, og en får ved litt regning at

$$\Gamma(m, x) = \text{gamma}(m) \cdot (1 - \text{gammainc}(x, m)). \quad (4.8)$$

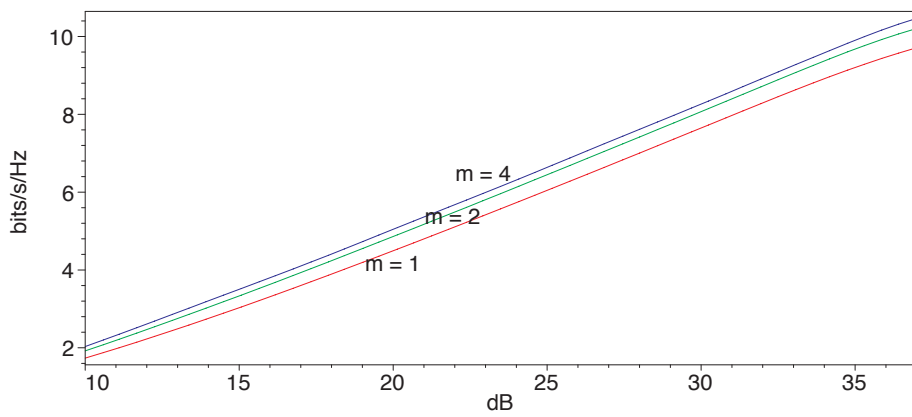
Den gjennomsnittlige spektrale effektiviteten for den flat-fadende AWGN-kanalen med AKM, med parametrene  $c = L = 1$ , er plottet for MLD og SD henholdsvis i figur 4.3 og figur 4.4. Differansen mellom disse plottene er vist i figur 4.5. For MLD er det bare plottet opp til og med 29 [dB] fordi fadingregioner med høyere signal/støy-forhold ikke er beregnet med tilstrekkelig nøyaktighet. Fra differanseplottet ser en at forskjellen mellom MLD og SD for  $\bar{\gamma} \in [10, 29]$  [dB] og  $c = L = 1$  er mindre enn 0.41 [biter/s/Hz].

Tilsvarende er den gjennomsnittlige spektrale effektiviteten for AKM med den flat-fadende AWGN-kanalen ved MLD og SD plottet for  $c = 1$  og  $L = 2$  i henholdsvis figur 4.6 og figur 4.7. Differansen mellom disse plottene er vist i figur 4.8. MLD- og differanseplottet er med samme begrunnelse som over bare plottet for gjennomsnittlig mottatt signal/støy-forhold opp til og med 30.7 [dB]. Fra differanseplottet ser en at forskjellen mellom MLD og SD for  $c = 1$  og  $L = 2$  er mindre enn 0.45 [biter/s/Hz].

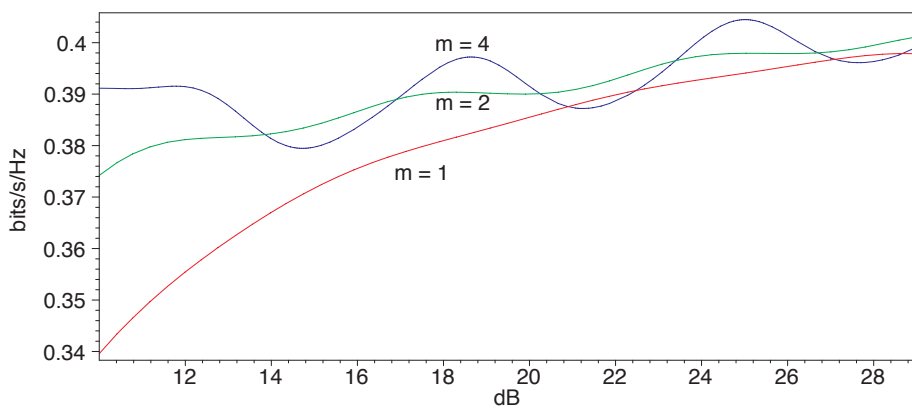
Merk at beregningene er basert på at AKM-systemet sender med en informasjonsrate lik kanalkapasiteten til en AWGN-kanal med et signal/støy-forhold lik den laveste terskelen til fadingregionen som det mottatte signal/støy-forholdet faller i. Det vil si at informasjonsraten hele tiden er tilnærmet optimal. En kan derfor se på plottene i figur 4.3, 4.4, 4.6 og 4.7 som øvre skranke på gjennomsnittlig spektral effektivitet for AKM på en flat-fadende AWGN-kanal.



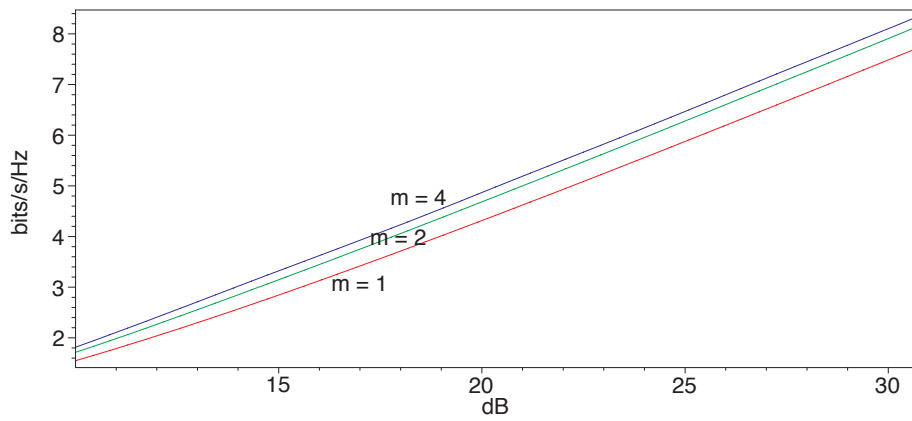
Figur 4.3: ASE for  $C^*$  der  $c = L = 1$



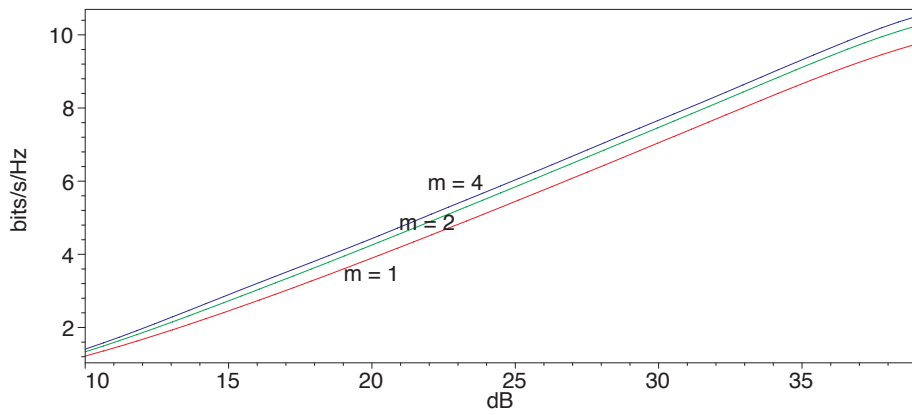
Figur 4.4: ASE for  $R_0^*$  der  $c = L = 1$



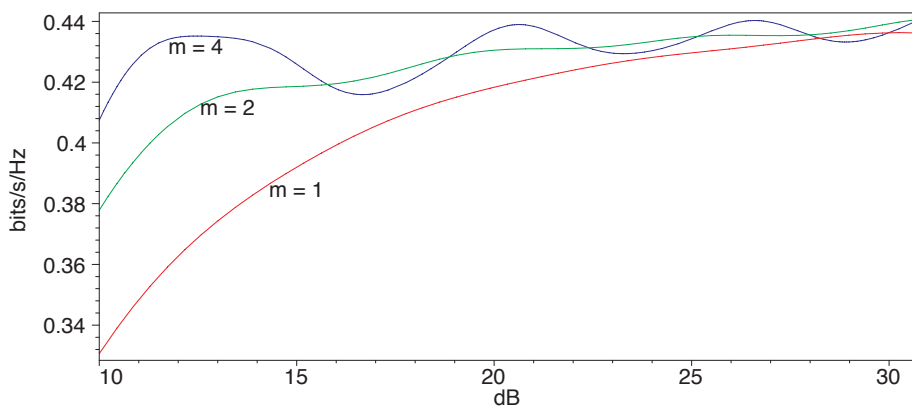
Figur 4.5: Differanse i ASE mellom  $C^*$  og  $R_0^*$  der  $c = L = 1$



Figur 4.6: ASE for  $C^*$  der  $c = 1, L = 2$



Figur 4.7: ASE for  $R_0^*$  der  $c = 1, L = 2$



Figur 4.8: Differanse i ASE mellom  $C^*$  og  $R_0^*$  der  $c = 1, L = 2$



# Kapittel 5

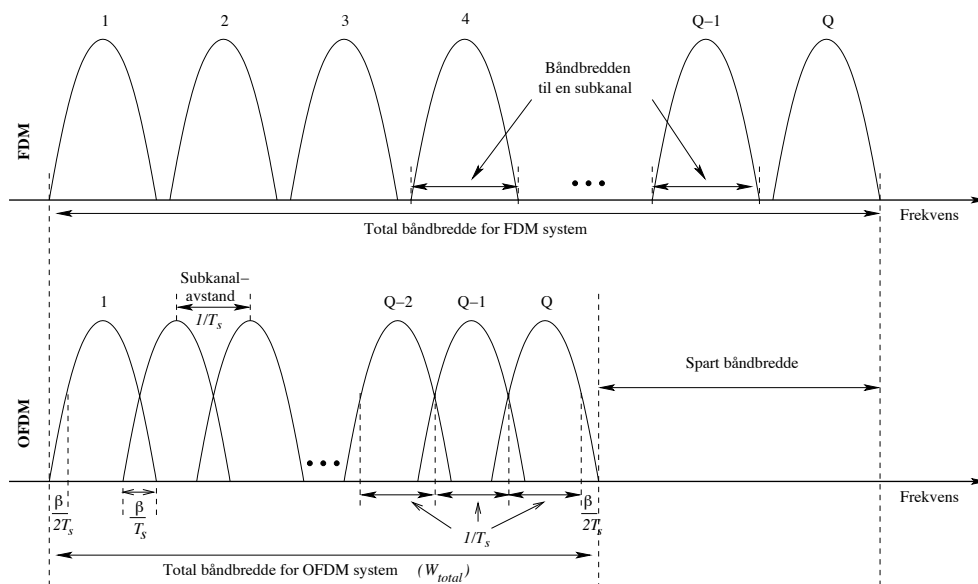
## Utvidelse av kanalmodellen til et OFDM system

Ønsket om høyhastighetsoverføring i trådløse kommunikasjonssystemer vokser hurtig, og OFDM (*Orthogonal Frequency Division Multiplexing*) [32, 33] er en interessant teknologi som kan brukes til å imøtekomme dette ønsket. Ved å bruke denne teknikken kan en overføre data med høy hastighet over begrenset båndbredde og dermed oppnå en høy spektral effektivitet.

OFDM er en variant av FDM (*Frequency Division Multiplexing*). FDM er en teknikk som går ut på sende flere signaler parallelt over samme kanal. I denne teknikken splittes det totale frekvensbåndet inn i flere subfrekvensbånd og hvert signal som skal sendes moduleres med en egen bærefrekvens. Denne frekvensen ligger et stykke fra frekvensområdet til de andre signalene som sendes samtidig, slik at det kun eksisterer ett signal innenfor hvert subfrekvensbånd til enhver tid. For at det ikke skal bli overlapp med andre signaler, blir signalene også separert av et beskyttelsesintervall, som består av en ubrukt del av frekvensspekteret. De ulike modulerte signalene blir deretter kombinert (eng. *multiplexed*) til et sammensatt signal. Dette signalet sendes over kanalen til mottakeren som deretter skiller de ulike signalkomponentene fra hverandre. En har da oppnådd å sende flere ulike signaler med liten båndbredde parallelt over kanalen i form av et signal med stor båndbredde.

Dersom en unngår spektral overlapping mellom parallelle signaler vil en kunne forhindre interkanalinterferens. Selv om dette kan virke smart, fører det til en lite effektiv utnyttelse av den totale tilgjengelige båndbredden. OFDM, eller ortogonal FDM, går også ut på å sende flere samtidige signaler over en fysisk kanal. Det blir gjort på en mer effektiv måte hvor det er frekvensoverlapp mellom signalene. Mer om hvordan dette gjøres, og hvordan et OFDM-signal kan sees på som signaler som går over forskjellige subkanaler, vil bli forklart i dette kapittelet.

Den enkle flat-fadende AWGN kanalmodellen vil bli benyttet som modell for en subkanal i OFDM-systemet. Det vil bli sett på adaptiv koding og modulasjon på subkanalene og bli beregnet øvre skranker for gjennomsnittlig spektral effektivitet for hele systemet. Videre vil det bli sett på den aktuelle nettverksstandarden HIPERLAN/2 (High PERFORMANCE Local Area Network type 2) [12, 34, 35] som benytter seg av OFDM. HIPERLAN/2 har mulighet for adaptivering av koder og konstellasjoner og det vil redegjøres for hva som blir gjort i standarden. Til slutt vil det bli gjort en sammenligning av verdier for spektral effektivitet mellom en modell av HIPERLAN/2-standard og modellen i denne oppgaven.



Figur 5.1: Sammenligning av båndbreddebruk mellom OFDM og FDM

## 5.1 Generelt om OFDM

Hovedideen bak OFDM er å dele det tilgjengelige frekvensbåndet inn i flere subkanaler. Antallet slike subkanaler vil i denne modellen betegnes ved  $Q$ . Figur 5.1 illustrerer forskjellen mellom FDM, som er uten frekvensoverlapping, og OFDM som har frekvensoverlappende subkanaler. En ser av figuren at OFDM er en båndbreddeeffektiv teknikk og at en kan spare båndbredde ved å bruke OFDM fremfor FDM.  $T_s$  er tidslengden til et QAM-symbol som skal sendes. Avstanden mellom frekvenssenteret til to nabosubkanaler i OFDM-systemet er gitt ved  $B_s = 1/T_s$  [Hz]. Denne avstanden er lik

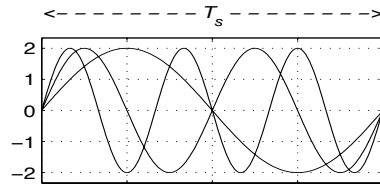
Nyquist-båndbredden til subkanalene. Utover dette har subkanalene en overskytende båndbredde (eng. *excess bandwidth*)  $B_o = \beta/T_s$  [Hz] for en *roll-off*-faktor  $0 \leq \beta \leq 1$ . På grunn av frekvensoverlapping vil bare en halvdel av den overskytende båndbredden til de to ytre subkanalene bidra til den totale båndbredden. Den totale båndbredden  $W_{total}$  til OFDM-systemet kan med notasjonen over uttrykkes ved

$$W_{total} = Q \cdot B_s + 2 \cdot B_o/2 = \frac{(Q + \beta)}{T_s}. \quad (5.1)$$

Ved parallelliseringen i OFDM blir en datastrøm med høy rate delt inn i mange parallelle datastrømmer med lavere rate. Hver parallelle datastrøm blir så modulert og tildelt en bærefrekvens for en av de  $Q$  subkanalene. Disse signalene blir så slått sammen til et OFDM-signal som sendes over den fysiske kanalen. En oppnår ved dette å øke symboltiden. Dermed blir flerveisforsinkelsen relativt til symboltiden redusert og færre symboler kan bli påvirket av tidligere sendte, forsinkete symboler.

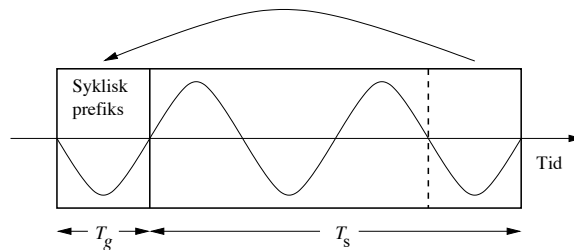
Når mottakeren mottar OFDM-signalet kan den skille de ulike signalkomponentene fra hverandre. I motsetning til FDM, hvor signalene kan skilles fra hverandre vha frekvensfiltre, må en i et OFDM-system bruke en annen teknikk pga frekvensoverlapping mellom subkanalene. Det er de ortogonale egenskapene til OFDM-signalet som gjør det mulig å gjenopprette signalene fra subkanalene. Subsignalene er ortogonale når bærebølgene kan beskrives av matematisk ortogonale funksjoner. En annen måte å se på ortogonalitet mellom subsignaler er å se på antall sykler til bærebølgene på et intervall. Når bærebølgene består av et heltall antall sykler for hvert QAM-symbol og når forskjellen i antall sykler mellom hver nabobærebølge er akkurat en, så er det ortogonalitet mellom subsignalene [32]. Figur 5.2 viser et eksempel på bærebølger i et OFDM-symbol med et heltallig antall sykler for hver bærebølge. Merk at i eksempelet har alle bærebølgene samme amplitude og fase, mens i et virkelig system kan amplituden og fasen moduleres forskjellig for hvert subsignal. Moduleringen slik at subbærebølgene blir ortogonale kan enklest gjøres ved en *invers diskret Fourier transformasjon* (IDFT), så de parallelle subdatastrømmene i OFDM-systemet blir derfor sendt gjennom en slik transformasjon før de blir kombinert til et OFDM-symbol.

Før OFDM-symbolet sendes blir det utvidet med en syklisk forlengelse av seg selv. Denne utvidelsen plasseres i forkant av OFDM-symbolet og fører til et større tidsvindu der en kan finne brukbare QAM-symboler. Se figur 5.3 for illustrasjon. Hensikten med det sykliske prefikset er å eliminere interkanalinterferens som kan oppstå f.eks. ved at ortogonaliteten mellom bærebølgene opphører pga forsinkelse i enkelte av bærebølgene. Det sykliske prefikset vil også fungere som et ekstra tidsintervall som beskytter mot intersymbolin-



Figur 5.2: Eksempel på tre subbærebølger i et OFDM symbol

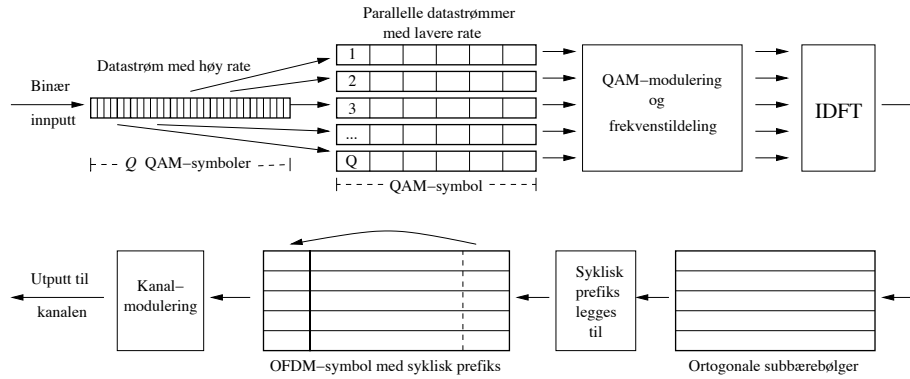
terferens. Prefikset velges derfor lenger enn den forventede forsinkelsen som følger av flerveisforplantningen på kanalen.



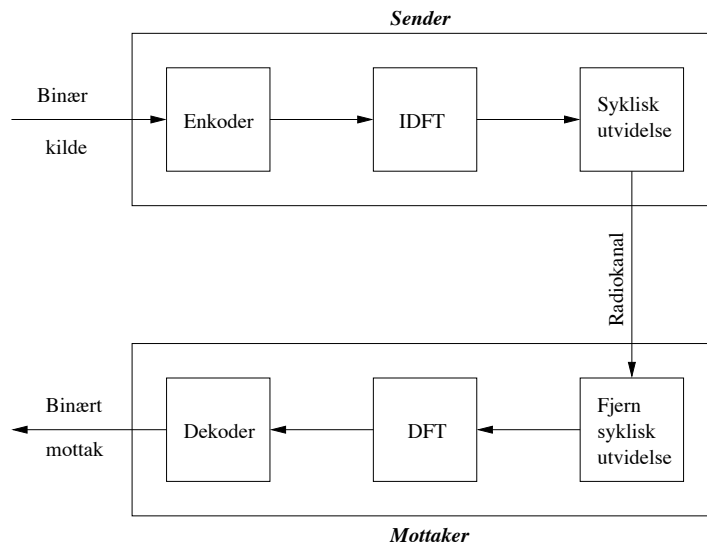
Figur 5.3: OFDM-symbol utvidet med syklisk prefiks

Dersom en tenker seg et parallelliseringsystem der en datastrøm av høy rate deles inn i flere parallelle datastrømmer med lavere rate vil den totale dataraten i utgangspunktet være den samme. Men OFDM bruker som nevnt et beskyttelsesintervall som gjør den totale symbollengden større, og da blir samtidig den totale dataraten redusert. La lengden av beskyttelsesintervallet betegnes ved  $T_g$  og la fremdeles  $T_s$  være lengden av et QAM-symbol på en subkanal. Den totale lengden til et OFDM-symbol blir da  $T_{OFDM} = T_g + T_s$ . Merk at siden den totale dataraten reduseres ved introduksjon av det sykliske prefikset velges vanligvis  $T_g$  ikke større enn  $T_s/4$ .

Figur 5.4 viser en skisse av senderen i OFDM-modellen, og den tar for seg de ulike stegene som gjennomgås for å generere et OFDM-symbol. Etter at OFDM-symbolet er generert blir det sendt over kanalen til mottakeren. Den tar i mot signalet og bestemmer de opprinnelige subsignalene. For å gjøre det må mottakeren først reversere behandlingen signalene fikk etter QAM-moduleringen. Den fjerner derfor først den sykliske forlengelsen av OFDM-symbolet. Deretter går signalene gjennom en *diskret Fourier transformasjon* (DFT) før de til slutt går gjennom en ordinær dekodning. En oversikt over OFDM-systemmodellen er vist i figur 5.5. Merk at stegene før den inverse diskrete Fourier-transformasjonen er slått sammen til fellesbetegnelsen enkoder. Tilsvarende er stegene etter den diskrete Fourier-transformasjonen i mottakeren sammenfattet til betegnelsen dekode.



Figur 5.4: Generering av et OFDM-symbol



Figur 5.5: Systemmodell for OFDM

## 5.2 OFDM subkanalmodell

Den enkle flat-fadende AWGN-kanalmodellen beskrevet i kapittel 3 vil her bli brukt som en modell for en subkanal i OFDM-systemet. Det henvises derfor til kapittel 3 med den originale modellen for utdypende forklaringer av symboler som blir brukt.

Det vil bli gjort en utvidelse av modellen som går på den fysiske oppstillingen. Senderen i OFDM-systemet antas nå å være en fastmontert basestasjon som har en enkel sendeantenne. Mottakeren blir antatt å være en mobil enhet, f.eks. en bærbar pc, og har en antenne med  $H$  antennegrener som hver fanger opp en versjon av OFDM-signalet. Det antas videre at de mottatte

signalene på antennegrenene er statistisk uavhengige, og at det er opp til mottakeren å bestemme en enkelt versjon av det OFDM-signalet. Det kan gjøres ved vha en *Maximum Ratio Combiner* (MRC), som vekter signalene fra hver antennegren etter signal/støy-forholdet og deretter summerer de sammen. Det totale mottatte signal/støy-forholdet etter at signalene fra hver antennegren er behandlet av MRC'en er lik summen av signal/støy-forholdet i hver antennegren [7].

I den opprinnelige flat-fadende kanalmodellen ble den gjennomsnittlige sendeeffekten betegnet ved  $\bar{S}$ . Det vil den også bli i denne modellen, men det antas nå at hver subkanal har gjennomsnittlig sendeeffekt  $\bar{S}$  på hver av antennegrenene. Siden antennegrenene hos mottakeren mottar hver sin kopi OFDM-signalet og disse signalene kan være av ulik kvalitet, vil det være rimelig å anta det skyldes forskjellig fading under overføringen. Vi betegner derfor fadingamplituden på den  $h$ -te antennegrenen og den  $q$ -te OFDM-subkanalen ved  $\alpha_{h,q}$ . Det instantane mottatte signal/støy-forholdet på den  $h$ -te antennegrenen for den  $q$ -te subkanalen kan da uttrykkes ved

$$\gamma_{h,q} = \frac{\alpha_{h,q}^2 \bar{S}}{N_0 B}. \quad (5.2)$$

Det totale instantane mottatte signal/støy-forholdet for den  $q$ -te subkanalen får en ved å summere over alle antennegrener, og det er gitt ved

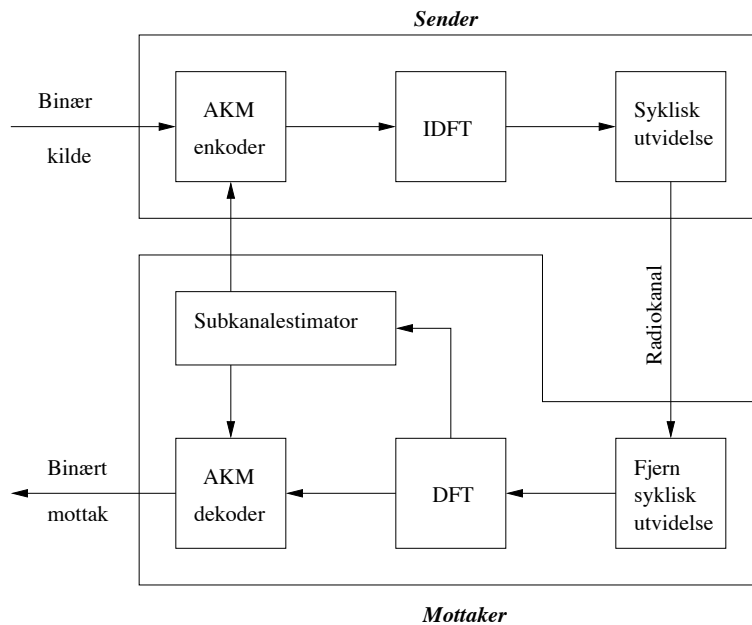
$$\gamma_q = \sum_{h=1}^H \gamma_{h,q} = \sum_{h=1}^H \frac{\alpha_{h,q}^2 \bar{S}}{N_0 B}. \quad (5.3)$$

Hvis en antar at gjennomsnittlig mottatt fadingeffekt  $\Omega_q = E[\alpha_{h,q}^2]$  er lik for alle  $H$  antennegrener på den  $q$ -te subkanalen, kan det gjennomsnittlige mottatte signal/støy-forholdet uttrykkes ved

$$\bar{\gamma}_q = E[\gamma_q] = \sum_{h=1}^H E[\gamma_{h,q}] = \frac{H \Omega_q \bar{S}}{N_0 B}. \quad (5.4)$$

## 5.3 Adaptiv koding og modulasjon i OFDM-modellen

I denne seksjonen vil det bli sett på hvordan adaptiv koding og modulasjon kan brukes i OFDM-systemet. Som det ble nevnt i kapittel 4 må mottakeren gjøre estimater av det mottatte signal/støy-forholdet på hver subkanal. Disse estimatene må formidles tilbake til senderen slik at den kan tilpasse enkodingen etter forholdene. Dette blir antatt å kunne gjøres uten forsinkelse slik at senderen har perfekt kjennskap til signal/støy-forholdene til enhver tid. Mottakeren må også kompensere for det mottatte signal/støyforholdet på hver subkanal etter samme retningslinjer som senderen for at det skal foregå forståelig kommunikasjon dem i mellom. Figur 5.6 viser OFDM-systemmodellen der AKM er implementert. Sammenlignet med modellen i figur 5.5 har senderen fått byttet ut enkoderen med en AKM-enkoder, og tilsvarende har mottakeren fått byttet ut dekodere med en AKM-dekoder. I tillegg har mottakeren en estimator for det mottatte signal/støy-forholdet på hver subkanal. Den er tegnet inn som en boks med merkelappen subkanalestimator. Grunnen er at det må gjøres estimater for de mottatte subsignalene slik at en kan adaptere koding og modulasjon på hver enkelt av de  $Q$  subkanalene.

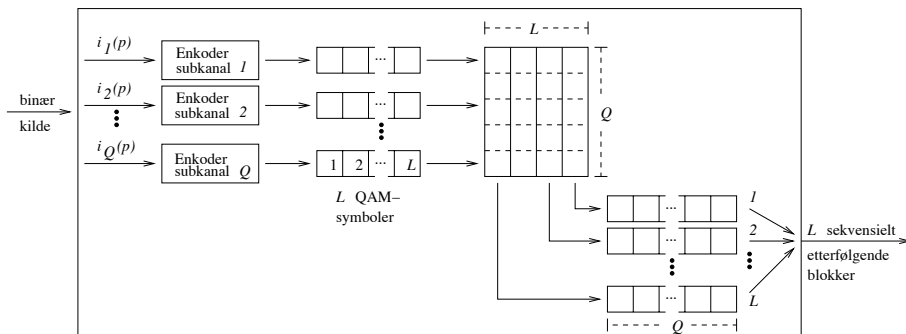


Figur 5.6: Systemmodell for OFDM med AKM

Figur 5.7 viser detaljene i AKM-enkoderen som benyttes i OFDM-systemet. Den tar først i mot  $i_q(p)$  informasjonsbiter for hver subkanal  $q \in \{1, 2, \dots, Q\}$ .

Mengden biter  $i_q(p)$  kan være forskjellig på de ulike subkanalene og antallet bestemmes av det instantane mottatte signal/støy-forholdet på subkanalene.  $i_q(p)$  vil variere over tid hvor gjeldende tid er indikert ved indekssymbolet  $p \in \{0, 1, 2, \dots\}$ . Hver subkanal encoder genererer  $L$  QAM-symboler slik at det totale antallet QAM-symboler ved tidsindeks  $p$  er  $Q \cdot L$ . Disse symbolene blir deretter delt inn  $L$  blokker med  $Q$  symboler som vist i figuren. Hver blokk inneholder da ett QAM-symbol fra hver subkanal, noe som gjør at signalene i en blokk er spredd jevnt ut over frekvensspekteret. De  $L$  blokkene går så sekvensielt ut av AKM-encoderen og videre gjennom IDFT-enheten hvor de blir transformert til  $L$  etterfølgende OFDM-symboler. Til slutt blir det som vanlig lagt til et syklisk prefiks til hvert OFDM-symbol før det er klart til å sendes over den fysiske kanalen.

Som nevnt er det antatt at mottakeren har en antenne bestående av  $H$  uavhengige antennegrener. Det betyr at det blir mottatt  $H$  ukorrelerte versjoner av hvert OFDM-symbol. For hver av disse versjonene blir først den sykliske utvidelsen fjernet før de blir prosessert av DFT-enheten som bestemmer subsignalene fra OFDM-symbolet. En tenker seg at det er en DFT-enhet for hver antennegren slik at en har totalt  $H \cdot Q$  QAM-signaler med støy etter DFT-prosesseringen. En MRC beregner deretter en vektet sum av signalene slik at mottakeren står igjen med  $Q$  QAM-symboler for hvert OFDM-symbol. For hver blokk av  $L$  OFDM-symboler tar AKM-dekoderen i mot  $L \cdot Q$  støybefengte QAM-symboler og produserer estimater av de sendte informasjonsbitene. Den interne strukturen til AKM-dekoderen er invers av AKM-encoderen. Med det menes at den prøver å reversere informasjonsbehandlingen som foregikk i AKM-encoderen.



Figur 5.7: AKM enkoder i OFDM-systemet



## 5.4 Spektral effektivitet

I denne seksjonen vil det bli sett på uttrykk for den spektrale effektiviteten til OFDM-systemet der AKM er implementert. I forrige seksjon ble det introdusert en AKM-enkoder som for hver tidsindeks  $p$  genererer  $L$  OFDM symboler. Innputt til AKM-enkoderen per tidsindeks  $p$  var  $i_q(p)$  informasjonsbiter per subkanal  $q \in \{1, 2, \dots, Q\}$ . Det totale antallet informasjonsbiter per  $L$  OFDM-symboler blir da  $\sum_{q=1}^Q i_q(p)$ . Informasjonsraten ved tidsindeks  $p$  blir  $\frac{1}{L \cdot T_{OFDM}} \cdot \sum_{q=1}^Q i_q(p)$ , og en kan da uttrykke den spektrale effektiviteten ved tidsindeks  $p$  ved

$$\frac{\sum_{q=1}^Q i_q(p)}{L \cdot T_{OFDM}} \cdot \frac{1}{W_{total}}. \quad (5.5)$$

Antallet informasjonsbiter  $i_q(p)$  på subkanal  $q$  vil variere fordi det avhenger av det instantane signal/støy-forholdet på subkanalen, noe som betyr at den totale informasjonsraten også vil kunne variere ved ulike tidsindekser. Den variasjonen er det ikke mulig å forhåndsbestemme eksakt. For å uttale seg om den gjennomsnittlige spektrale effektiviteten vil det derfor være nødvendig å ta i bruk en statistisk tilnærming til problemet. I den forbindelse vil det bli sett vekk fra tidsindekseringen som ble markert ved symbolet  $p$  fordi den utgjør en grov inndeling som omfatter  $L$  OFDM-symboler i en gang. Dersom en summerer gjennomsnittlig antall informasjonsbiter i QAM-symbolene på hver subkanal får en ut gjennomsnittlig antall informasjonsbiter per OFDM-symbol. Dvs at gjennomsnittlig antall informasjonsbiter per intervall  $T_{OFDM}$  er

$$\langle \#biter \rangle = \sum_{q=1}^Q \sum_{n=1}^N (\log_2(M_n) - c/L) P_q(\gamma_n, \gamma_{n+1}) \quad (5.6)$$

der  $P_q(\gamma_n, \gamma_{n+1}) = \int_{\gamma_n}^{\gamma_{n+1}} p(\gamma_q) d\gamma$  er sannsynligheten for at det instantane signal/støy-forholdet  $\gamma_q$  faller i den  $n$ -te fadingregionen.

Den gjennomsnittlige informasjonsraten blir  $\bar{R} = \langle \#biter \rangle / T_{OFDM}$ . Med dette på plass kan en uttrykke den total gjennomsnittlige spektrale effektiviteten i et OFDM/AKM/QAM system ved

$$ASE_{OFDM}(\{\gamma_n\}) = \frac{\bar{R}}{W_{total}} = \frac{\langle \#biter \rangle}{T_{OFDM} \cdot W_{total}}. \quad (5.7)$$

I utgangspunktet kan fadingen være forskjellig på de ulike subkanalene. Siden den totale båndbredden til OFDM-systemet (16.26 MHz) er liten i

forhold til sendefrekvensen ( $\approx 5$  GHz), er det rimelig å anta at alle  $H \cdot Q$  antennegrener har et instantant mottatt signal/støy-forholdet  $\gamma_{h,q}$  med lik fordeling, og samme gjennomsnittlige mottatte fadingeffekt  $\Omega_q$  [30]. Med denne antakelsen kan fadingamplituden på alle antennegrenene karakteriseres av en Nakagamifordeling med parameter  $m$ . Det instantane mottatte signal/støy-forholdet  $\gamma_q$  på den  $q$ -te OFDM-subkanalen har da en gammafordelt sannsynlighetstetthet og er gitt ved [36]

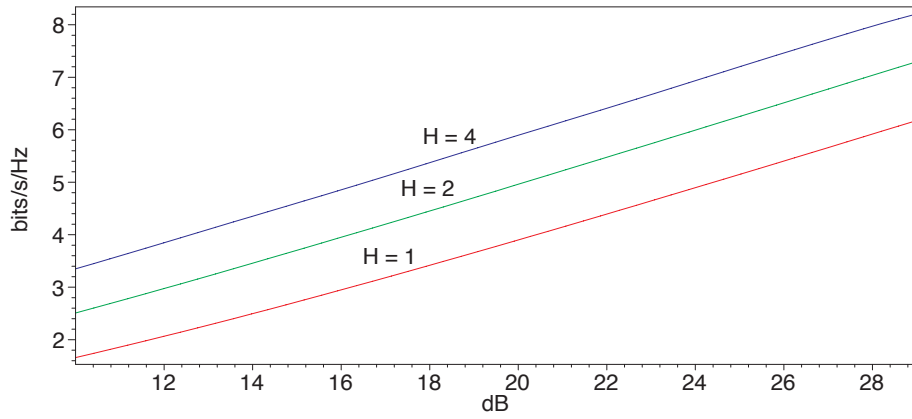
$$p(\gamma_q) = \left(\frac{m}{\bar{\gamma}}\right)^{Hm} \frac{\gamma_q^{Hm-1}}{\Gamma(Hm)} \exp\left(-\frac{m\gamma_q}{\bar{\gamma}}\right). \quad (5.8)$$

For alle  $q$  subkanaler vil sannsynligheten for at  $\gamma_q$  havner i  $n$ -te fadingregion være lik og er gitt ved [30]

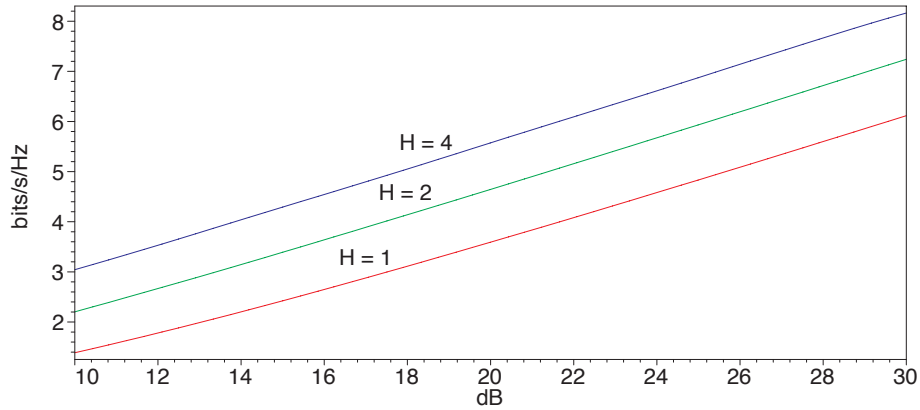
$$P_q(\gamma_n, \gamma_{n+1}) = \frac{\Gamma(Hm, \frac{m\gamma_n}{\bar{\gamma}}) - \Gamma(Hm, \frac{m\gamma_{n+1}}{\bar{\gamma}})}{\Gamma(Hm)} \quad (5.9)$$

der  $\Gamma(\cdot, \cdot)$  og  $\Gamma(\cdot)$  er som i seksjon 4.3, henholdsvis den komplementære ukomplette gammafunksjonen og gammafunksjonen.

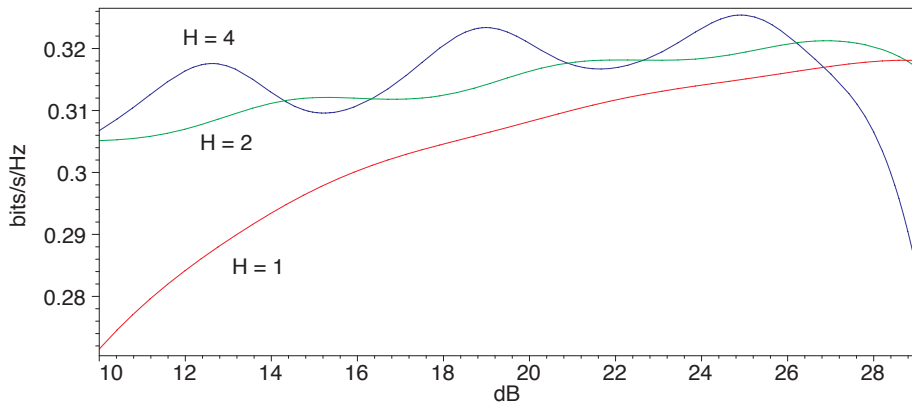
Den totale gjennomsnittlige spektrale effektiviteten til OFDM/AKM/QAM-systemet for MLD og SD ble beregnet ved å benytte henholdsvis fadingregionersklene  $\gamma_n^{C^*}$  og  $\gamma_n^{R^*}$  fra tabell 4.1. Figur 5.8-5.10 viser resultatet av beregningene for  $c = L = m = 1$  og  $H \in \{1, 2, 4\}$ . Differanseplottet viser at for  $\bar{\gamma} \in [10, 29]$  [dB] så er forskjellen mellom MLD og SD med disse parameterverdiene i intervallet (0.27, 0.33) [biter/s/Hz]. Merk at her er  $\bar{\gamma}$  gjennomsnittlig mottatt signal/støy-forhold per antennegren. Tilsvarende viser figur 5.11- 5.13 resultatet av beregningene av den totale gjennomsnittlige spektrale effektiviteten til OFDM/AKM/QAM-systemet for  $c = L = 1$ ,  $m = 2$  og  $H \in \{1, 2, 4\}$ . Differanseplottet i figur 5.13 viser at for  $\bar{\gamma} \in [10, 29]$  [dB] så er forskjellen mellom MLD og SD i intervallet (0.29, 0.34) [biter/s/Hz].



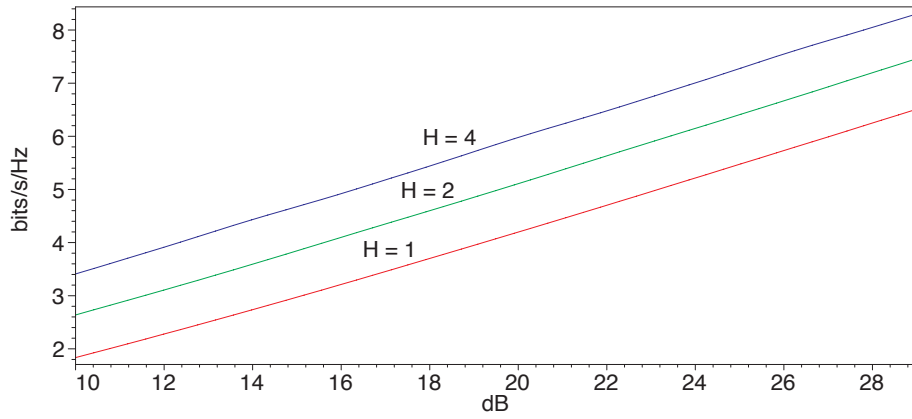
Figur 5.8: Total ASE for  $C^*$  der  $c = 1$ ,  $L = 1$ , og  $m = 1$



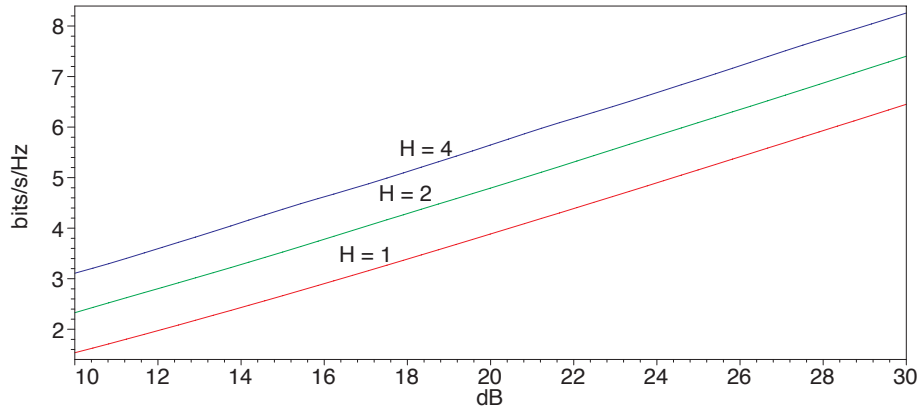
Figur 5.9: Total ASE for  $R_0^*$  der  $c = 1$ ,  $L = 1$ , og  $m = 1$



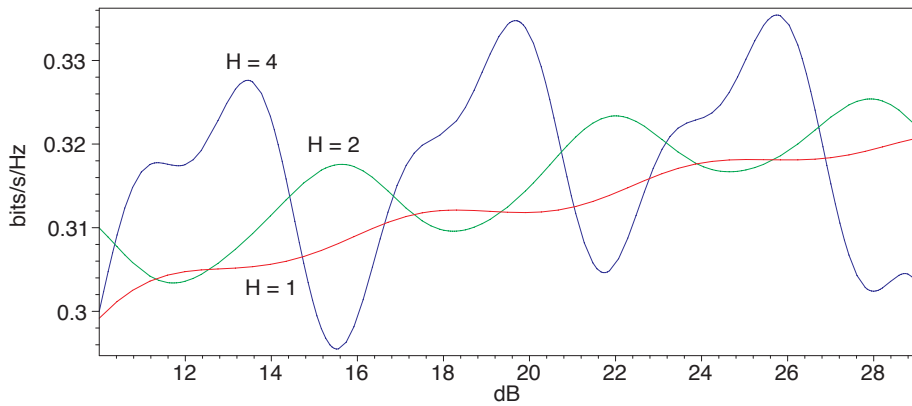
Figur 5.10: Differanse i total ASE mellem  $C^*$  og  $R_0^*$ , der  $c = 1$ ,  $L = 1$ , og  $m = 1$



Figur 5.11: Total ASE for  $C^*$  der  $c = 1$ ,  $L = 1$ , og  $m = 2$



Figur 5.12: Total ASE for  $R_0^*$  der  $c = 1$ ,  $L = 1$ , og  $m = 2$



Figur 5.13: Differanse i total ASE mellom  $C^*$  og  $R_0^*$ , der  $c = 1$ ,  $L = 1$ , og  $m = 2$

## 5.5 Et praktisk OFDM-system: HIPERLAN/2

Som nevnt i introduksjonen i kapittel 1, så er HIPERLAN/2 en ny aktuell standard for trådløse høyhastighetsnettverk. Standarden tillater to kommunikasjonsmodi, enten direkte kommunikasjon mellom mobile enheter, eller kommunikasjon mellom en mobil enhet og en fastmontert basestasjon. Med en rekkevidde på opp mot 200m egner systemet seg for større kontorer, hjemmenettverk eller for vrimeområder på kafeer, flyplasser, togstasjoner o.l. Felles for slike områder er ofte at topologien er slik at radiosignalene reflekteres mot forskjellige objekter før de når mottakeren med forsinkelse. Dette kan føre til problemer med intersymbol- og interkanal-interferens og HIPERLAN/2 benytter derfor OFDM som er effektiv teknikk i slike miljøer.

### 5.5.1 Litt om protokollstabelen

Tabell 5.1 viser en grov oversikt over protokollagene for HIPERLAN/2 [34, 12]. I de høyere lag (eng. *Higher Layers* (HL)) på toppen av protokollen befinner det seg nettverkstjenester på applikasjonsnivå. Eksempler på dette kan være autentiserings og krypteringsfunksjonalitet som krever korrespondanse mellom to enheter. Under de høyere lag finner en konvergenslaget (eng. *Convergence Layer* (CL)). Konvergenslaget har som hovedoppgave å tilby tjenester fra det lavere datalinkkontrollaget (eng. *Data Link Control Layer* (DLC)) til det høyere lag. Det må derfor oversette tjenestene slik at de blir forståelige for applikasjonene som skal bruke dem. I tillegg må konvergenslaget konvertere datapakker som skal sendes mellom de høyere lag og datalinkkontrollaget.

Datalinkkontrollaget er den delen av protokollen hvor en finner data-transport- og linkkontrollfunksjoner. En viktig del av datatransportfunksjonene er mediumtilgangskontroll (eng. *Medium Access Control* (MAC)). Gjennom MAC-protokollen bestemmer basestasjonene tidsrammer for når mobile terminaler kan kommunisere. Ved en tidsdeling av sendekapasiteten kan en effektivt supportere flere mobile terminaler i samme nett. En annen del av datatransportfunksjonene er feilkontroll. Datapakker får tildelt et sekvensnummer og en CRC-kontrollsum (*Cyclic Redundancy Check*). Ved hjelp av dette blir det foretatt en feilsjekk hos mottakeren og retransmisjon av pakker kan foretas dersom feil oppdages. Linkkontrollfunksjonene finnes i RLC-sublaget (eng. *Radio Link Control sublayer*), og der transporteres det kontrollinformasjon på DLC-nivået. Et eksempel på slik kontrollinformasjon kan være syntakskontroll på utvekslete meldinger. Applikasjoner fra høyere lag trenger ikke vite hva som skjer radionivået og de kommuniserer derfor gjennom RLC som skaper uavhengighet fra det fysiske laget.

Det nederste nivået i protokollstabelen er det fysiske laget (eng. *Physical Layer* (PHY) [37]) hvor de radiorelaterte funksjonene finnes. Her bestemmes det hvordan databitene skal enkodes, moduleres og sendes. Sendeteknikken er OFDM med 48 subkanaler for datatransmisjon og 4 subkanaler for pilotsignaler. Det er definert 7 sendemodi i PHY-laget med forskjellig modulasjon og koderater, se tabell 5.2 for spesifikasjon. HIPERLAN/2 tilbyr med dette en mulighet for å adaptere linken etter kanaltilstanden, og med utgangspunkt i estimater av signal/støy-forholdet må det velges en senderate som samsvarer med et av sendemodiene.

Higher layers (HL)
Convergence Layer (CL)
Data Link Control Layer (DLC)
Physical layer (PHY)

Tabell 5.1: Protokollagene for HIPERLAN/2

Modus	Modulasjon	Koderate	PHY bit rate	bytes/OFDM symb.
1	BPSK	1/2	6 Mbps	3.0
2	BPSK	3/4	9 Mbps	4.5
3	QPSK	1/2	12 Mbps	6.0
4	QPSK	3/4	18 Mbps	9.0
5	16-QAM	9/16	27 Mbps	13.5
6	16-QAM	3/4	36 Mbps	18.0
7	64-QAM	3/4	54 Mbps	27.0

Tabell 5.2: PHY-modi definert for HIPERLAN/2

### 5.5.2 Ytelsessimuleringer for HIPERLAN/2-modell

I denne subseksjonen vil det bli gjort ytelsessimuleringer for en modell av HIPERLAN/2. Modellen tar utgangspunkt i OFDM-systemet som er presentert i dette kapitlet og benytter parametere hentet fra HIPERLAN/2, se tabell 5.3. En viktig forenkling som blir gjort er at modellen benytter alle 52 subkanaler for å sende data. Det vil altså bli sett bort fra at noen kanaler er reservert for pilotsignaler.

For å vurdere ytelsen til HIPERLAN/2-systemet vil det være interessant å se på hvilke verdier av spektral effektivitet det kan oppnå. For å finne disse

symbol	navn	verdi
$\sigma_d$	delay spread	250 ns
$B$	Båndbredde subkanal	312.5 kHz
$T_s$	symboltid	3.2 $\mu$ s
$\beta$	rolloff faktor	0.25
$Q$	subkanaler	52
$T_g$	guard intervall	800 ns
$T_{OFDM}$	OFDM-intervall	4 $\mu$ s
$W_{total}$	system bandwith	$\approx$ 16.26 MHz

Tabell 5.3: Parametere i HIPERLAN/2

verdiene kan en benytte uttrykk (5.5). Det blir altså antatt at det genereres  $L$  OFDM-symboler for hver tidsindeks  $p$ . Antallet informasjonsbiter  $i_q(p)$  på den  $q$ -te subkanalen vil avhenge av hvilken QAM-konstellasjon og koderate som benyttes der. Basert på estimater av signal/støy-forholdene i systemet velges det fadingregioner og dermed konstellasjoner og koderater. Det vil være nødvendig å simulere verdier av signal/støy-forholdet på subkanalene for å kunne bestemme verdier for den instantane spektrale effektiviteten. Det vil dessuten bli benyttet samme koding og modulasjon på alle subkanaler slik som det gjøres i HIPERLAN/2. Dette er en begrensning av OFDM som tillater adaptivering på hver subkanal.

Som nevnt i seksjon 5.4 følger det instantane signal/støy-forholdet på subkanalene i OFDM-systemet en gammafordeling. Det ble derfor benyttet en algoritme, [38, s. 277], som tar et eksponentielt fordelt tall som innputt og gir ut en variabel  $x$  som følger fordelingen

$$f_x(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} \exp^{-x/\beta}, \quad 0 < x < \infty, \quad (5.10)$$

hvor  $x$ ,  $\alpha$  og  $\beta$  er hentet fra [38] og ikke følger notasjonen i denne oppgaven. Ved å sette  $\alpha = Hm$ ,  $x = \gamma$  og  $\beta = \bar{\gamma}/m$  får en

$$\begin{aligned} f_\gamma(\gamma) &= \frac{1}{\Gamma(Hm) \left(\frac{\bar{\gamma}}{m}\right)^{Hm}} \gamma^{Hm-1} \exp\left(-\frac{\gamma}{(\bar{\gamma}/m)}\right) \\ &= \left(\frac{m}{\bar{\gamma}}\right)^{Hm} \frac{\gamma^{Hm-1}}{\Gamma(Hm)} \exp\left(-\frac{m\gamma}{\bar{\gamma}}\right) = p(\gamma) \end{aligned}$$

som er det samme som uttrykk (5.8). For å simulere signal/støy-forholdet på subkanalene ble algoritmen implementert i *MATLAB*. De eksponentielt fordelte tallene ble bestemt fra uniformfordelte tilfeldige tall generert av *MAT-*

*LAB*-funksjonen *rand*. Dette ble også gjort ved hjelp av en algoritme hentet fra [38, s. 277].

I HIPERLAN/2 velges det et PHY-sendemodus basert på estimater av signal/støy-forholdet. Når et PHY-modus er valgt innebærer det at det brukes samme koding og modulasjon på alle subkanalene samtidig. For at alle subkanaler skal være operative, må det velges konstellasjoner og koderater tilpasset fadingregionen hvor det laveste signal/støy-forholdet blant subkanalene faller. La derfor  $\gamma_{min}$  betegne det minste signal/støy-forholdet blant de  $Q$  subkanalene, dvs

$$\gamma_{min} = \min\{\gamma_1, \gamma_2, \dots, \gamma_Q\}. \quad (5.11)$$

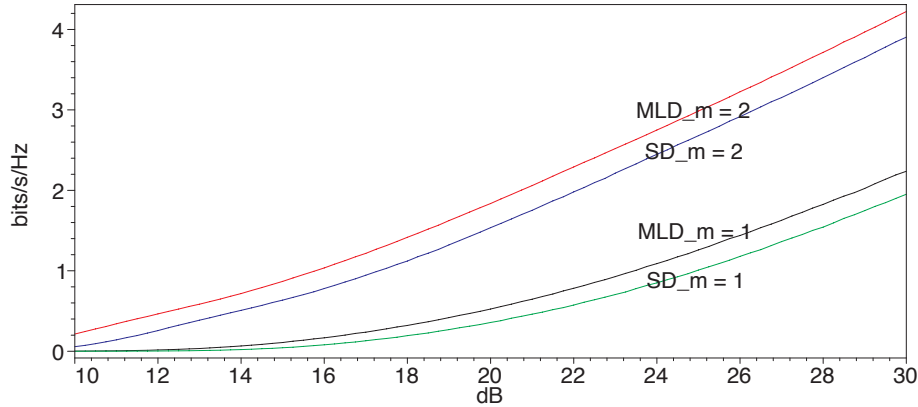
La videre  $n_{min} \in \{1, 2, \dots, N\}$  være fadingregionen hvor  $\gamma_{n_{min}} \leq \gamma_{min} < \gamma_{n_{min}+1}$ . Antall informasjonsbiter  $i_q = \log_2(M_{n_{min}}) - c/L$  på en subkanal vil være det samme for alle subkanaler. Den totale instantane spektrale effektiviteten kan da uttrykkes ved

$$\frac{Q \cdot (\log_2(M_{n_{min}}) - c/L)}{T_{OFDM} \cdot W_{total}}. \quad (5.12)$$

Simuleringene som ble gjort for modellen av HIPERLAN/2 bruker ikke direkte de 7 PHY-modiene som er beskrevet i standarden. I stedet brukes fadingregionene med de tilhørende signalkonstellasjonene som er gitt i tabell 4.1. Det ble implementert en algoritme i *MATLAB* som for et gitt gjennomsnittlig mottatt signal/støy-forhold per antennegren beregner den spektrale effektiviteten. For å finne verdier for den gjennomsnittlige spektrale effektiviteten ble det iterert 100000 ganger over hvert punkt. Resultatet av simuleringene er vist i figur 5.14. En ser ved å sammenligne med figur 4.3 og 4.4 at denne modellen kommer dårligere ut enn den enkle subkanalmodellen med AKM. Det kommer av at signal/støy-forholdet ble simulert uavhengig for hver subkanal. Når en da minimerer over alle subkanalene så er det stor sannsynlighet for at minst en av kanalene har et lavt signal/støy-forhold. Dette fører til at kun konstellasjoner og koderater fra de laveste fadingregionene blir brukt.

Resultatet av simuleringene over viser at det er for pessimistisk å modellere signal/støy-forholdet uavhengig for hver subkanal. Det vil derfor være nødvendig å se om en kan si noe om avhengigheten mellom subkanalene. Koherensbåndbredden er størrelsen på et frekvensspekter hvor signalkomponenter har et sterkt potensial for amplitude- og fasekorrelasjon. En kan også si at koherensbåndbredden indikerer frekvensforskjellen hvor to frekvenskomponenter blir så ukorrelerte at frekvenskorrelasjonsfunksjonen  $\rho$  når en forhåndsbestemt verdi. Denne verdien har ofte blitt satt til 0.5, 0.9 og  $1/e$  [21].





Figur 5.14: ASE for HIPERLAN/2-modell der  $c = 1$ ,  $L = 1$ ,  $H = 1$

For  $\rho = 0.5$  er koherensbåndbredden i [21, (2.100)] gitt ved  $B_k = \frac{1}{2\pi \cdot T_{rms}}$ , hvor  $T_{rms}$  er rms (eng. *root mean square*) flerveisforsinkelse. Et annet resultat, [7, (5.39)], sier at dersom frekvenskorrelasjonsfunksjonen er over 0.5 så er koherensbåndbredden gitt ved  $B_k = \frac{1}{5 \cdot T_{rms}}$ .

Tabell 5.4 viser en oppsummering av kanalmodeller for HIPERLAN/2 definert av ETSI BRAN [39]. Disse modellene er basert på måleprøver for typiske miljøer hvor HIPERLAN/2 kan benyttes. Den andre kolonnen inneholder verdier for rms-flerveisforsinkelse  $T_{rms}$ . Modellene er brukt i flere simuleringsarbeider og det er aksept for at verdiene er gode for HIPERLAN/2. La antallet korrelerte subkanaler være betegnet ved  $b$ . For å estimere dette antallet kan en dividere koherensbåndbredden med subkanalbåndbredden, dvs  $b = B_k/B$ . Den fjerde kolonnen i tabell 5.4 viser avrundet antall korrelerte subkanaler for  $B_k = \frac{1}{5 \cdot T_{rms}}$ .

Kanalmodell	$T_{rms}$	Miljø	$\approx b$
A	50 ns	Typisk kontor, ikke direktesikt	13
B	100 ns	Åpent rom/kontor, ikke direktesikt	6
C	150 ns	Stort åpent rom, ikke direktesikt	4
D	200 ns	Stort åpent rom, direktesikt	3
E	250 ns	Stort åpent rom, ikke direktesikt	3

Tabell 5.4: ETSI BRAN-kanalmodeller for HIPERLAN/2

En kan endre modellen ved å dele subkanalene inn i  $n_B$  blokker med  $b$  subkanaler i hver blokk. Ved å generere en felles prøve av signal/støyforholdet for hver blokk kan en få med subkanalkorrelasjon i modellen. Det vil fortsatt benyttes samme konstellasjon og koderate på alle subkanalene

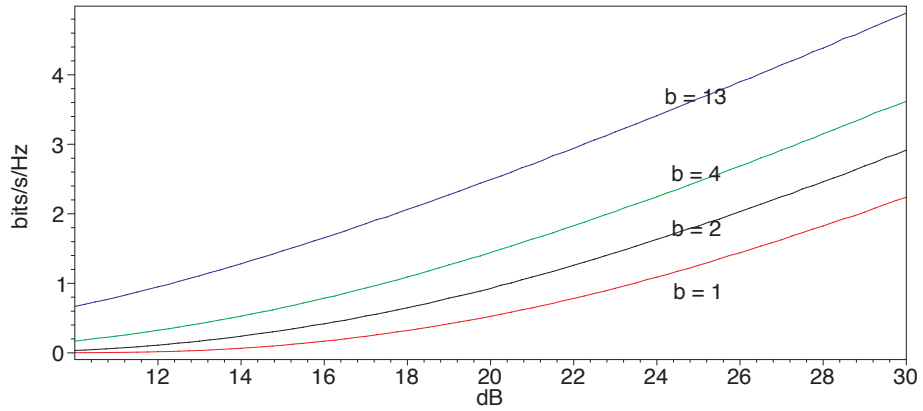
slik som i HIPERLAN/2, men dette må være tilpasset subkanalblokken med lavest signal/støy-forhold. La derfor  $\gamma_{min_B} = \min\{\gamma_1, \gamma_2, \dots, \gamma_{n_B}\}$  være det minste signal/støy-forholdet blant de  $n_B$  blokkene og la  $n_{min_B} \in \{1, 2, \dots, N\}$  være fadingregionen hvor  $\gamma_{n_{min_B}} \leq \gamma_{min_B} < \gamma_{n_{min_B}+1}$ . For et gjennomsnittlig mottatt signal/støy-forhold  $\bar{\gamma}$  per antennegren og en blokkstørrelse  $b$  kan en estimere den gjennomsnittlige spektrale effektiviteten ved

$$ASE(\bar{\gamma}, b) \approx \frac{1}{K} \sum_{k=1}^K \frac{Q \cdot (\log_2(M_{n_{min_B}}^k) - c/L)}{T_{OFDM} \cdot W_{total}}, \quad (5.13)$$

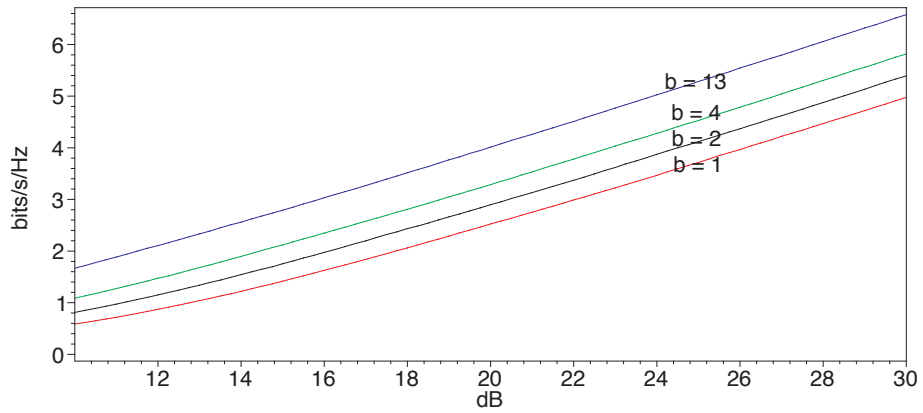
hvor  $K$  er antall uavhengige beregninger av instantan spektral effektivitet. Merk at antall informasjonsbiter per subkanal,  $\log_2(M_{n_{min_B}}^k) - c/L$ , vil kunne variere for hver av de  $k$  beregningene. Konstellasjonstørrelsen  $M_{n_{min_B}}^k$  er derfor markert med superskript  $k$  for å indikere dette.

For å få like mange subkanaler i hver blokk, må blokkstørrelsen være heltallsdivisor av  $Q$ . Tabell 5.4 antyder at det største antallet korrelerte subkanaler er 13. Simuleringene av gjennomsnittlig spektral effektivitet ble derfor gjort for  $b \in \{1, 2, 4, 13\}$ . Resultatet av simuleringene for  $K = 100000$  er vist i figur 5.15-5.26. En kan se at den gjennomsnittlige spektrale effektiviteten øker når antall korrelerte subkanaler øker.

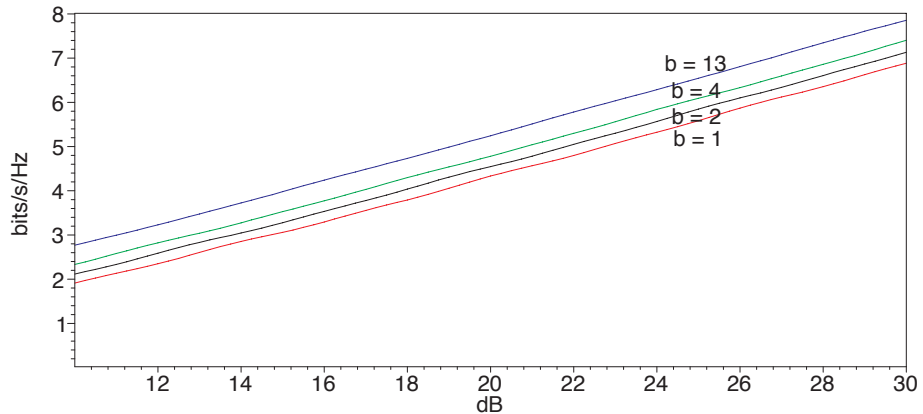
Koherensbåndbreddeformelen  $B_k = \frac{1}{5 \cdot T_{rms}}$  ble valgt fordi den for ETSI-kanalmodell A ga antall korrelerte subkanaler  $b \approx 13$  som er en heltallsdivisor av  $Q = 52$ . Dersom en hadde brukt uttrykket  $B_k = \frac{1}{2\pi \cdot T_{rms}}$  ville en fått at antallet korrelerte subkanaler ville være  $b \approx 10$  for ETSI-kanalmodell A. Dette indikerer at det faktiske maksimale antallet korrelerte subkanaler kan være mindre enn 13. Sett i betraktning av dette kan verdiene som ble beregnet for  $b = 13$  sees på som et estimat av en øvre grense for gjennomsnittlig spektral effektivitet til HIPERLAN/2-modellen.



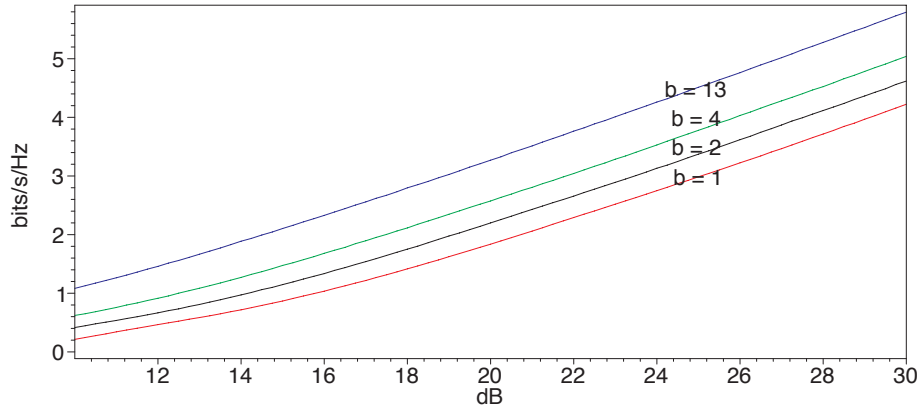
Figur 5.15: Total ASE for HIPERLAN/2-modell for  $C^*$  der  $c, L, m = 1$  og  $H = 1$



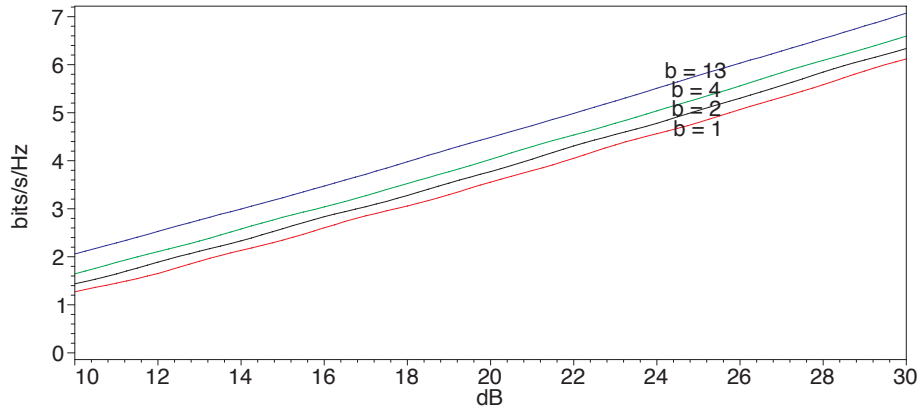
Figur 5.16: Total ASE for HIPERLAN/2-modell for  $C^*$  der  $c, L, m = 1$  og  $H = 2$



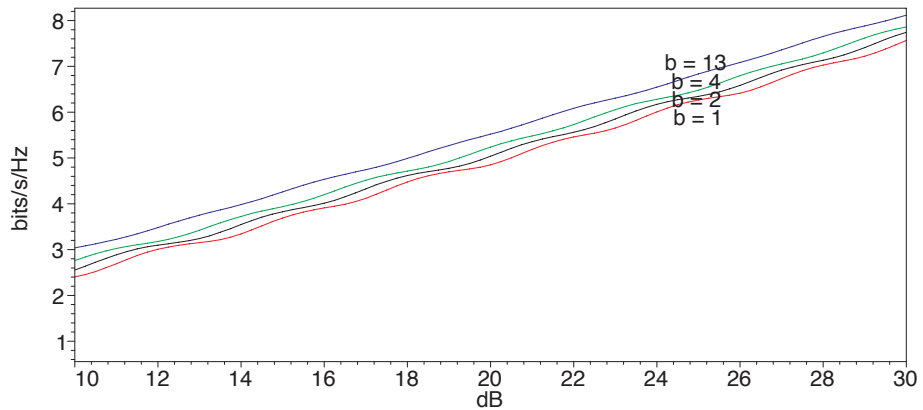
Figur 5.17: Total ASE for HIPERLAN/2-modell for  $C^*$  der  $c, L, m = 1$  og  $H = 4$



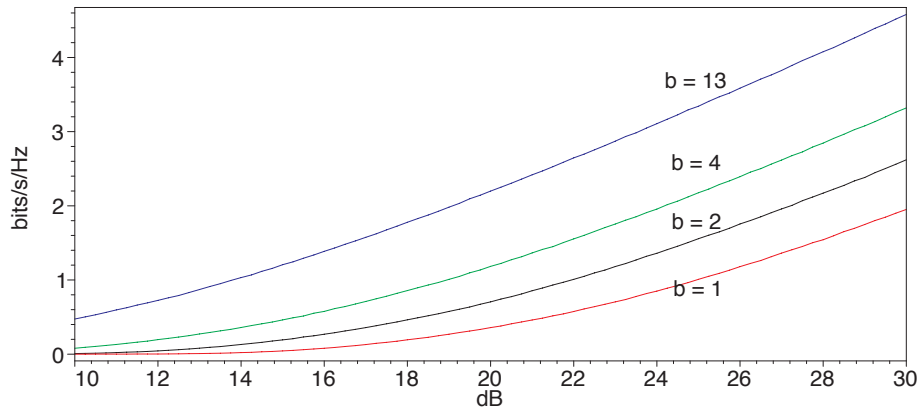
Figur 5.18: Total ASE for HIPERLAN/2-modell for  $C^* c, L = 1, m = 2$  og  $H = 1$



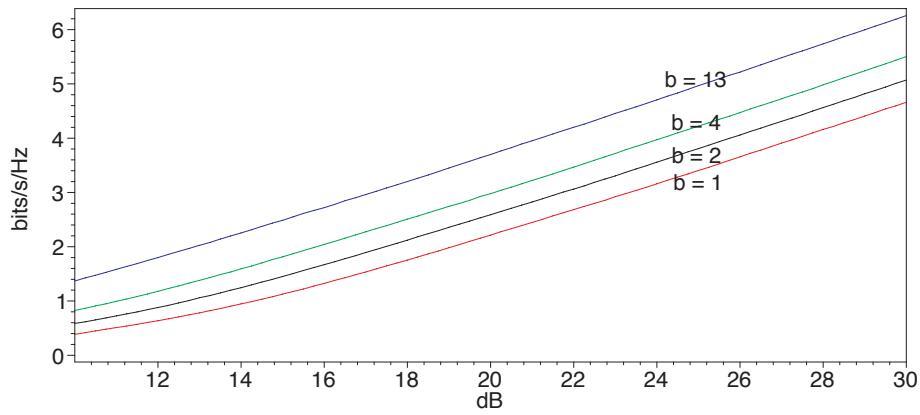
Figur 5.19: Total ASE for HIPERLAN/2-modell for  $C^* c = 1, L = 1, m = 2$  og  $H = 2$



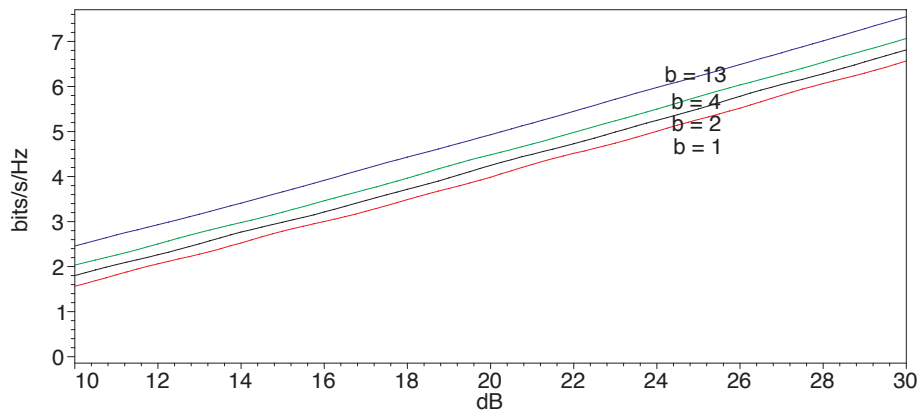
Figur 5.20: Total ASE for HIPERLAN/2-modell for  $C^* c = 1, L = 1, m = 2$  og  $H = 4$



Figur 5.21: Total ASE for HIPERLAN/2-modell for  $R_0^* c, L, m = 1$  og  $H = 1$



Figur 5.22: Total ASE for HIPERLAN/2-modell for  $R_0^* c, L, m = 1$  og  $H = 2$



Figur 5.23: Total ASE for HIPERLAN/2-modell for  $R_0^* c, L, m = 1$  og  $H = 4$

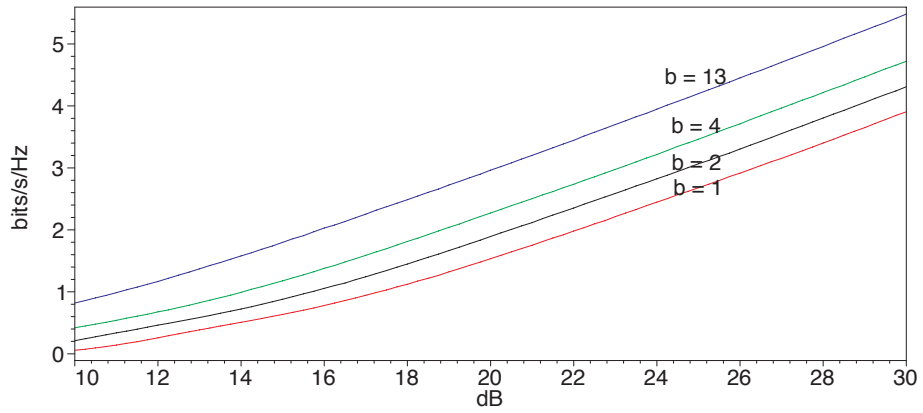


Figure 5.24: Total ASE for HIPERLAN/2-modell for  $R_0^* c, L = 1, m = 2$  og  $H = 1$

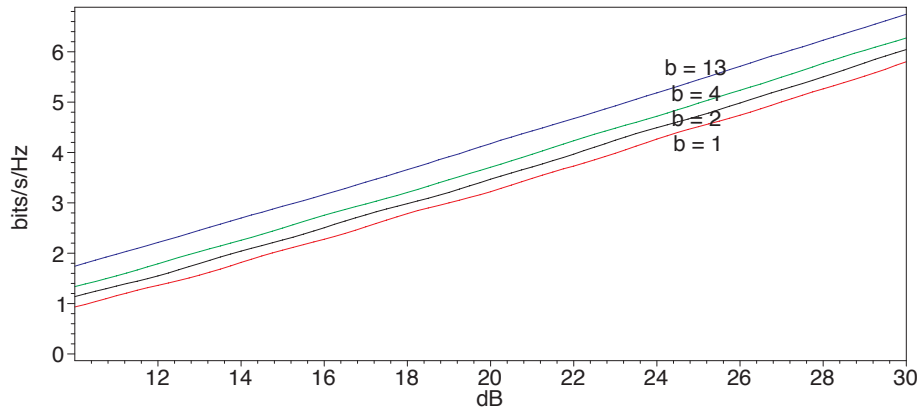


Figure 5.25: Total ASE for HIPERLAN/2-modell for  $R_0^* c, L = 1, m = 2$  og  $H = 2$

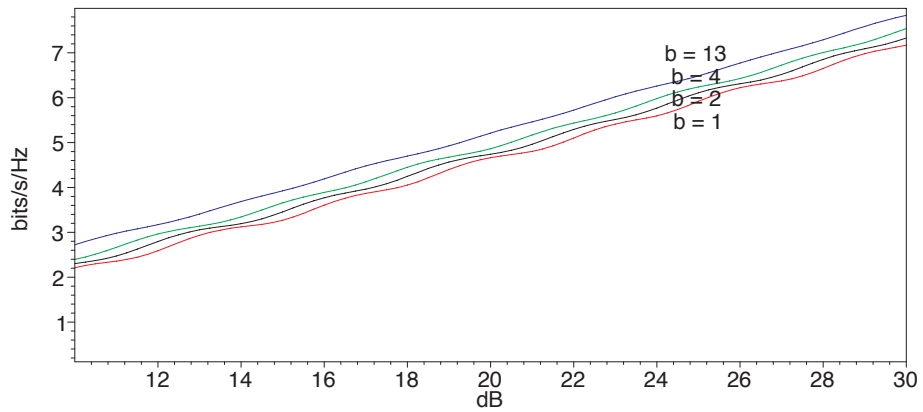
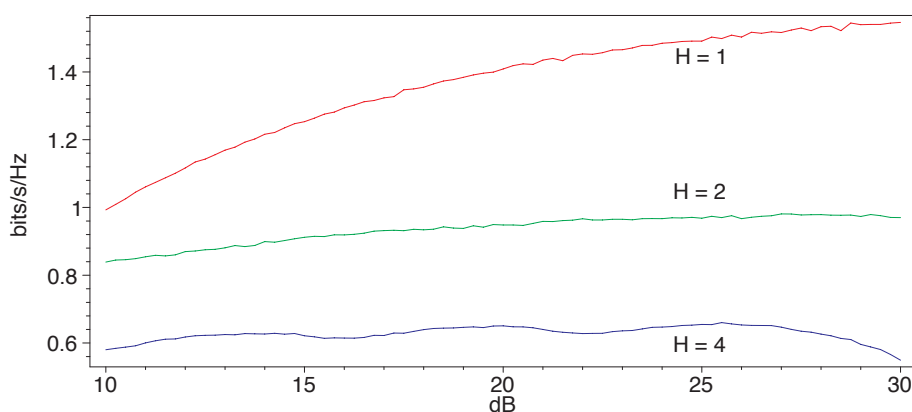


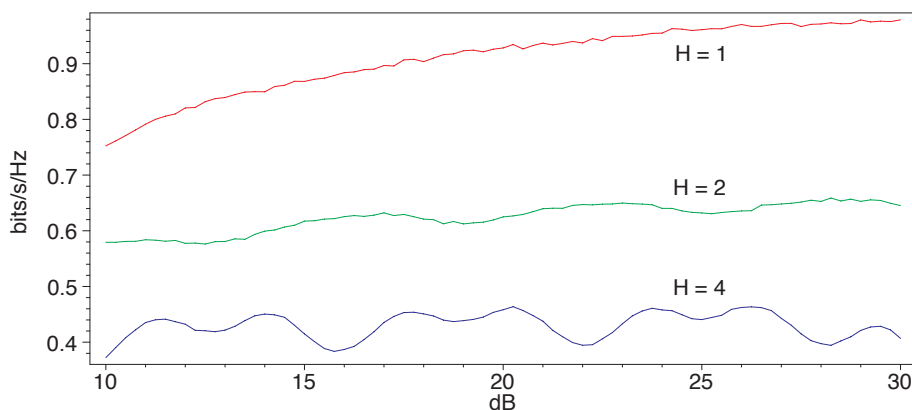
Figure 5.26: Total ASE for HIPERLAN/2-modell for  $R_0^* c, L = 1, m = 2$  og  $H = 4$

### 5.5.3 Forbedringer av HIPERLAN/2

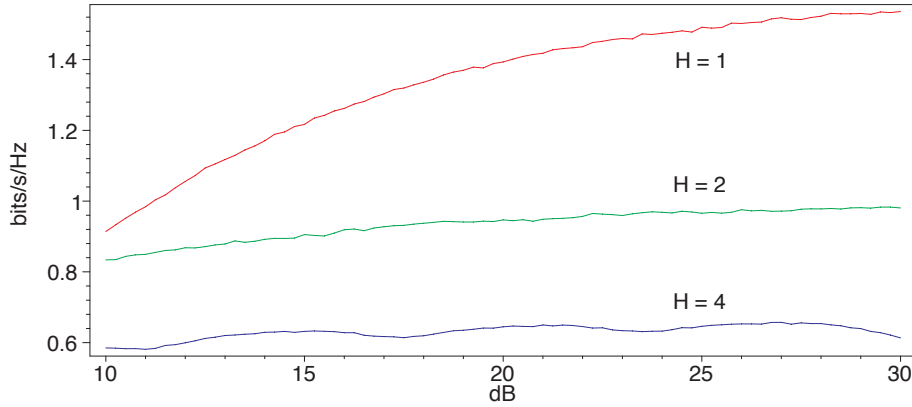
Når et PHY-modus er satt i HIPERLAN/2 benyttes samme modulasjon og koding på alle subkanaler. Dette er en begrensning av OFDM som tillater forskjellige senderater og konstellasjonsstørrelser på ulike subkanaler. Det er sannsynlig at noen av subkanalene i HIPERLAN/2 har et signal/støy-forhold som er høyere enn andre subkanaler. Disse kanalene har da også en høyere kanalkapasitet og tillater derfor overføring av større datamengder enn subkanaler med lavere signal/støy-forhold. Ved å bruke systemet med adaptiv koding og modulasjon for OFDM som er presentert i dette kapittelet vil en kunne fordele databiter mer optimalt i forhold kapasiteten til den enkelte subkanalen.



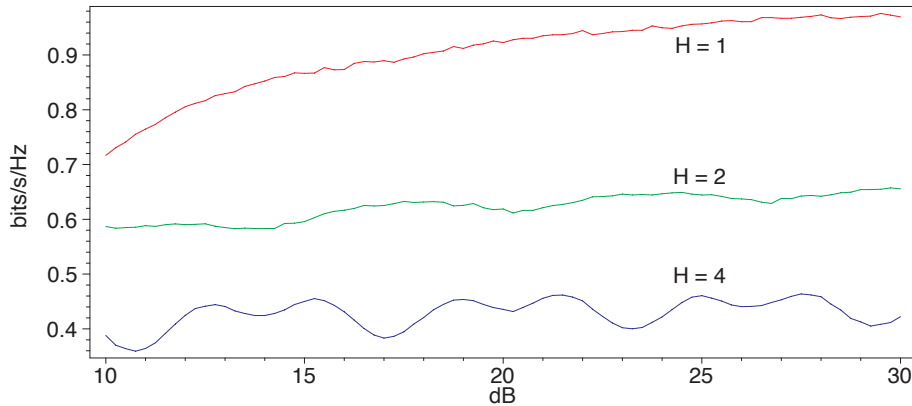
Figur 5.27: Differanse i ASE mellom OFDM/AKM/QAM-system og HIPERLAN/2-modell ved MLD med parametrene  $c, L = 1, m = 1$ .



Figur 5.28: Differanse i ASE mellom OFDM/AKM/QAM-system og HIPERLAN/2-modell ved MLD med parametrene  $c, L = 1, m = 2$ .



Figur 5.29: Differanse i ASE mellom OFDM/AKM/QAM-system og HIPERLAN/2-modell ved SD med parametrene  $c, L = 1, m = 1$ .



Figur 5.30: Differanse i ASE mellom OFDM/AKM/QAM-system og HIPERLAN/2-modell ved SD med parametrene  $c, L = 1, m = 2$ .

En kan beregne differansen mellom den gjennomsnittlige spektrale effektiviteten til OFDM/AKM/QAM-systemet og HIPERLAN/2-modellen og få et estimat på hvor mye det er å tjene på å adaptere på hver subkanal. Figur 5.27 og 5.28 viser differansen mellom systemene for henholdsvis Rayleigh- og tilnærmet Rice-fading ( $m = 1$  og  $m = 2$ ) når en bruker MLD. Fra HIPERLAN/2-modellen ble de høyeste verdiene for gjennomsnittlig spektral effektivitet brukt, dvs verdiene som ble beregnet for  $b = 13$  korrelerte subkanaler. Beregningene ble gjort for  $H \in \{1, 2, 4\}$  antennegrener og figurene viser at differansen i gjennomsnittlig spektral effektivitet er minst for  $H = 4$ . Tilsvarende viser figur 5.29 og 5.30 differansen i gjennomsnittlig spektral effektivitet for SD. Det kan være interessant å merke seg at ASE-differansene for samme nakagamiparameter  $m$  er tilnærmet like for MLD og SD.



# Kapittel 6

## Oppsummering

I denne hovedfagsoppgaven ble det presentert nye øvre skranke for ytelsen til trådløse radiobaserte nettverk. For å gi en introduksjon til emnet ble det først gitt en beskrivelse av digital informasjonsoverføring. Det ble gitt definisjoner av viktige begreper knyttet til dette og fortalt om egenskaper som er spesielle for trådløse nettverk. Det ble også gitt et innføringskapittel om hvordan digital informasjon kan representeres som analoge signaler. Blant annet ble den utbredte teknikken *kvadraturamplitudemodulasjon* (QAM) gjennomgått.

Deretter ble det bygget opp en modell av en båndbreddebegrenset flat-fadende AWGN-kanal og beregnet kanalkapasiteten for denne når en benytter *Maximum Likelihood Decoding* (MLD) og *sekvensiell dekoding* (SD). Beregningene viste en forskjell på mindre enn 0.4 biter per QAM-symbol mellom kanalkapasiteten for MLD og SD. SD gir altså ikke mye dårligere verdier enn for MLD, og det kan derfor være interessant å bruke SD fremfor MLD på grunn av lavere dekodingskompleksitet.

Videre ble det sett på hvordan *adaptiv koding og modulasjon* (AKM) kan benyttes på den flat-fadende AWGN-kanalen. Basert på kanalkapasiteten ved MLD og SD ble det bestemt fadingregiongrenser som garanterer forhåndsbestemte spektrale effektivitetsverdier. Det ble også beregnet gjennomsnittlig spektral effektivitet for den presenterte modellen med AKM implementert. Siden disse beregningene tok utgangspunkt i senderater lik de tidligere beregnede kanalkapasitetene, kan de sees på som øvre ytelsesskranke for en flat-fadende AWGN-kanal med AKM.

Til slutt ble kanalmodellen utvidet til et system med *Orthogonal Frequency Division Multiplexing* (OFDM). Prinsippet bak OFDM ble forklart og det ble vist hvordan AKM kan benyttes sammen med OFDM. Den enkle flat-fadende AWGN-kanalmodellen ble brukt som modell for en subkanal i OFDM-systemet. Det ble estimert total gjennomsnittlig spektral effektivitet for OFDM/AKM/QAM-systemet. Disse estimatene kan også sees på som øvre skranke for et OFDM-

/QAM/AKM-system fordi det blir antatt at senderatene er lik kanalkapasiteten. Det ble også gjort simuleringer av ytelsen til en modell av kommunikasjonsstandarden HIPERLAN/2. Ved å sammenligne resultatene fra disse simuleringene med estimatene for OFDM/AKM/QAM-systemet, kan en konkludere med at HIPERLAN/2 har betydelig å tjene på benytte AKM slik som det er beskrevet i denne oppgaven.

Ved Rayleigh-fading er de gjennomsnittlige spektrale effektivitetsforskjellene i intervallet (0.6, 1.4) [biter/sek/Hz] for både MLD og SD. Videre er de gjennomsnittlige spektrale effektivitetsforskjellene i intervallet (0.4, 0.9) for MLD og SE ved tilnærmet Rice-fading. Både Rayleigh- og Rice-fading har fordelinger som passer godt for typiske miljøer (se tabell 5.4) hvor HIPERLAN/2 kan benyttes. Ved å multiplisere disse verdiene med den totale systembåndbredden får en informasjonsrateforskjellene mellom systemene.

*For Rayleigh-fading ligger informasjonsratens forbedringspotensiale for HIPERLAN/2 i intervallet (9.8, 22.8) [Mbit/s], mens den for tilnærmet Rice-fading ligger i intervallet (6.5, 14.6) [Mbit/s].*

## 6.1 Videre arbeid

Nøyaktigheten på de estimerte verdiene til kanalkapasiteten for den enkle flat-fadende AWGN-kanalmodellen ved MLD er usikker for konstellasjonsstørrelsene  $M_n \in \{1024, 2048, 4096\}$  siden det bare ble gjort en kjøring av hver simuleringsserie. For å kunne bestemme disse mer nøyaktig vil det være nødvendig å kjøre flere simuleringsserier og sannsynligvis øke parameteren  $l$  (se seksjon 3.5.1). Dette vil kreve lang kjøretid. En løsning kan være å implementere simuleringene i et raskere programmeringsspråk som f.eks. C/C++. Da vil det også være mulig å gjøre punktregninger parallelt på en flerprosessormaskin, noe som ikke støttes av *MATLAB*. Alternativt kan det utvikles et fleksibelt skript som kjører distribuerte *MATLAB*-prosesser på mange arbeidsstasjoner.

Det er viktig å huske på at det ikke ble simulert korrelasjon mellom subkanalblokker i modellen av HIPERLAN/2. Dette ble gjort som en forenkling og kan ha påvirket inntrykket av hvor mye det er å tjene på å benytte AKM i HIPERLAN/2. Forbedringer av HIPERLAN/2-modellen vil kunne gi bedre estimater på ytelsesforskjeller mellom HIPERLAN/2 og OFDM/AKM/QAM-systemet. Forslag til tillegg som kan gi forbedringer er:

- Simulering av korrelasjon mellom subkanalblokker.
- Simulering av korrelasjon over tid.

- Benytte alle parameterene fra HIPERLAN/2. Dvs å bruke 48 subkanaler for å sende data, benytte de 7 PHY-modiene som er beskrevet i standarden, og lage en sammenlignbar OFDM/AKM/QAM-systemmodell.

Det vil også være interessant å gjøre noen utvidelser av AKM-systemmodellen. Blant annet kan følgende gjøres:

- Se på effekten av forsinkelse i tilbakemeldingskanalen.
- Adaptere sendeeffekten etter forholdene

Når det kommer HIPERLAN/2-produkter på sluttbrukermarkedet vil det også være mulig med praktiske tester for å vurdere ytelsesforskjeller. AKM medfører økt kompleksitet. Uansett mulige ytelsesforbedringer vil det være nødvendig å vurdere ekstra kostnader ved å implementere AKM i HIPERLAN/2. Om et produkt blir en suksess er avhengig av flere faktorer, deriblant pris.



# Notasjonsindeks

Symbolene er ordnet alfabetisk med greske symboler først.

Tallene i den høyre kolonnen angir sidetall hvor symbolet er definert.

$\alpha^2$	Fading envelope .....	24
$\alpha_{h,q}$	Fadingamplitude på $h$ -te antennenegren og $q$ -te subkanal .....	45
$\beta$	<i>Rolloff</i> -faktor .....	42
$\gamma$	Instantant mottatt signal/støy-forhold .....	21
$\bar{\gamma}$	Gjennomsnittlig mottatt signal/støy-forhold .....	25
$\gamma_{h,q}$	Signal/støy-forhold på $h$ -te antennenegren og $q$ -te subkanal .....	45
$\gamma_q$	Totalt instantant signal/støy-forhold på $q$ -te subkanal .....	45
$\bar{\gamma}_q$	Gjennomsnittlig signal/støy-forhold på $q$ -te subkanal .....	45
$\gamma_n^{C^*}$	Fadingregiongrenser for MLD .....	35
$\gamma_n^{R_0^*}$	Fadingregiongrenser for sekvensiell dekoding .....	35
$\Gamma(\cdot)$	Gammafunksjonen .....	35
$\Gamma(\cdot, \cdot)$	Den komplementære ukomplette gammafunksjonen .....	35
$\Omega$	Gjennomsnittlig mottatt fadingeffekt .....	25
$\Omega_q$	Gjennomsnittlig mottatt fadingeffekt på $q$ -te subkanal .....	45
$\rho$	Frevenskorrelasjonsfunksjon .....	55
$\theta$	Signalfase .....	16
$\langle \#biter \rangle$	Gjennomsnittlig antall informasjonsbiter per OFDM-symbol .....	48
$A$	Signalamplitude .....	15
$ASE(\bar{\gamma}, b)$	Gjennomsnittlig spektral effektivitet for HIPERLAN2/-modellen	57
$ASE_{\text{OFDM}}(\{\gamma_n\})$	Total gjennomsnittlig spektral effektivitet for OFDM/AKM/QAM-system	48
$B$	Båndbredde .....	25
$b$	Størrelse til subkanalblokker i modell av OFDM for HIPERLAN/2	56
$B_k$	Koherensbåndbredde .....	25
$B_o$	Overskytende båndbredde .....	42
$B_s$	Nyquist båndbredde til OFDM-subkanalene .....	41
$c$	Antall paritetsbiter per $L$ QAM-symboler .....	32
$C^*$	Kanalkapasitet ved MLD .....	26

$f_c$	Bærefrekvens .....	15
$H$	Antall antennegrener .....	44
$i_q(p)$	Antall informasjonsbiter på den $q$ -te subkanalen for tidsindeks $p$ ...	46
$k$	Tidsindeks .....	32
$l$	Antall beregninger for å bestemme en gjennomsnittsverdi .....	27
$L$	Antall QAM-symboler generert av (sub)kanalenkoder .....	32
$m$	Nakagamiparameteren .....	23
$M$	Størrelsen på en signalkonstellasjon .....	18
$M_n$	Størrelsen på en signalkonstellasjon $n$ .....	32
$N_0$	Ensidig spektral tetthet til støy .....	21
$n_B$	Antall subkanalblokker i modell av OFDM for HIPERLAN/2 .....	56
$r$	Mottatt basisbåndsignal (komplekst) .....	20
$p$	Tidsindeks .....	47
$P$	Instantant mottatt signaleffekt .....	21
$P(\gamma_n, \gamma_{n+1})$	Sannsynlighet for at $\gamma$ havner i fadingregion $n$ .....	33
$P_q(\gamma_n, \gamma_{n+1})$	Sannsynlighet for at $\gamma_q$ havner i fadingregion $n$ .....	49
$q$	Subkanalidentifikator .....	45
$Q$	Antall subkanaler i OFDM-systemet .....	41
$R$	Koderaten til en blokkode .....	8
$\bar{R}$	Gjennomsnittlig informasjonsrate .....	48
$R_0^*$	Cutoff raten ved sekvensiell dekoding .....	29
$R_n$	Informasjonsrate til kode $n$ .....	34
$s$	Sendt basisbåndsignal (komplekst) .....	20
$S$	Instantant sendt signaleffekt .....	21
$\bar{S}$	Gjennomsnittlig sendt signaleffekt .....	25
$t$	Diskret tid .....	21
$T$	Signalintervall .....	25
$T_g$	Beskyttelsesintervall for OFDM-symbol .....	43
$T_s$	Lengden av et QAM-symbol .....	43
$T_{maks}$	Maksimal forsinkelse blant komponentene til et signal .....	25
$T_{\text{OFDM}}$	Lengden av et OFDM-symbol .....	43
$T_{rms}$	RMS forsinkelse blant komponentene til et signal .....	56
$u$	Antall informasjonsbiter i et kodeord fra en blokkode .....	8
$v$	Lengden av et kodeord i en blokkode .....	8
$w$	Additiv hvit gaussisk støy .....	20
$W_{total}$	Total båndbredde til OFDM-systemet .....	42

# Bibliografi

- [1] Store Norske Leksikon, URL: <http://www.storenorskeleksikon.no>, Kunnskapsforlaget Aschehoug & Gyldendal.
- [2] Encyclopædia Britannica Online, URL: <http://search.eb.com>.
- [3] C. E. Shannon, *A Mathematical Theory of Communication*, Reprinted with corrections from The Bell System Technical Journal, Vol. 27, pp. 379-423, 623-656, 1948. URL: <http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf>.
- [4] R. Hill, *A First Course in Coding Theory*, Oxford University Press, 2000.
- [5] S Lin, D. J. Costello Jr., *Error Control Coding: Fundamentals and applications*, Prentice Hall, 1983.
- [6] Wireless Communications Association International (WCA), URL: <http://www.wcai.com/>.
- [7] T. S. Rappaport, *Wireless Communication - Principles and Practice*. Prentice Hall, second edition 2002.
- [8] Post- og teletilsynet, URL: <http://www.npt.no>.
- [9] HiperLAN2 Global Forum. URL: <http://www.hiperlan2.com>.
- [10] ETSI, European Telecommunications Standards Institute, URL: <http://www.etsi.org>.
- [11] Institute of Electrical and Electronics Engineers Inc.(IEEE), URL: <http://www.ieee.org/>.
- [12] M. Johnsson, *HiperLAN/2 - the Broadband Radio Transmission Technology Operating in the 5 GHz Frequency Band*, HiperLAN/2 Global Forum, 1999. Version 1.0.

- [13] The Wi-Fi Alliance, URL: <http://www.wi-fi.org/>.
- [14] F. Kristensen and M. Sandgren, *A system simulation of wireless local area networks operating in the 5 GHz band*, Master of Science Thesis for Telia Research AB, August 2001. URL: <http://www.hiperlan2.com/presdocs/site/rapporten039.doc>
- [15] J. G. Proakis and M. Salehi, *Communication Systems Engineering*. Prentice-Hall, second edition 2002.
- [16] E. Ström, *Notes on Quadrature Amplitude Modulation*,, october 22 2002, URL: <http://www.s2.chalmers.se/undergraduate/courses/ess140/doc/Handouts/qam-notes.pdf>.
- [17] B. Sklar, *Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization*, IEEE Communications Magazine, July 1997.
- [18] S. A. Abbas and A. U. Sheikh, *On understanding the nature of slow fading in LOS microcellular channels* Proc. IEEE Veh. Technol. Conf (VTC'97)(Phoenix, AZ May 1997), pp. 662-666.
- [19] Gordon L. Stüber, *Principles of Mobile Communication*, second printing 1997.
- [20] M.-S. Alouini and A. Goldsmith, *Adaptive M-QAM Modulation Over Nakagami Fading Channels*, Department of Electrical Engineering, California Institute of Technology.
- [21] R. Steele and L. Hanzo, *Mobile Radio Communications* second edition 1999.
- [22] G. D. Forney, Jr., *The Viterbi algorithm*, Proc. IEEE, vol. 61, pp. 268-278, 1973.
- [23] G. Ungerboeck, *Channel Coding with Multilevel/Phase Signals*, IEEE Trans. Inform. Theory, Vol IT-28, pp. 55-67.
- [24] Maplesoft, *MAPLE*,  
URL: <http://www.maplesoft.com/products/maple/>.
- [25] The MathWorks, *MATLAB*,  
URL: <http://www.mathworks.com/products/matlab/>.



- [26] R. M. Fano, *A heuristic discussion of probabilistic decoding*, IEEE Transactions on Information Theory, col. IT-9, April 1963.
- [27] F.-Q. Wang and D. J. Costello, *Erasure-Free Sequential Decoding of Trellis Codes*, IEEE Trans. Inform. Theory Vol 40. NO 6. November 1994.
- [28] G. Caire, G. Taricco and E. Biglieri, *Optimum Power Control Over Fading Channels*, IEEE Transactions on Information Theory, Vol.41, July 1999.
- [29] A. J. Goldsmith and S.-G. Chua, *Adaptive coded modulation for fading Channels*, IEEE Trans. Commun., vol. 46, pp. 595-602, May 1998.
- [30] K. J. Hole, *Average Spectral Efficiency of Adaptive Coded Modulation in Wireless Communication Systems: Bounds and Interpretations*, DRAFT October 31, 2001.
- [31] I.S. Gradshteyn and I.M. Ryzhi, *Table of integrals, series and products* 5th edition, Academic Press 1994.
- [32] R. van Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*, 2000.
- [33] O. Edfors, M. Sandell, J. J. van de Beek, D. Landström and Frank Sjöberg, *An introduction to orthogonal frequency-division multiplexing*, September 1996.
- [34] ETSI HIPERLAN/2 standard. URL: <http://www.etsi.org/frameset/home.htm?technicalactiv/Hiperlan/hiperlan2.htm>.
- [35] J. Khun-Jush, G. Malmgren, P. Schramm and J. Torsner, *HIPERLAN type 2 for broadband wireless communication*, Ericsson Review No.2, 2000.
- [36] M.-S. Alouini and A. Goldsmith, *Capacity of Nakagami Multipath Fading Channels*, Proc. 47th IEEE Vehicular Technology Conference (VTC'97) pp. 358-362.
- [37] ETSI, *Broadband Radio Access Networks; HIPERLAN Type 2; Physical (PHY) layer*, ETSI TS 101 475 v1.1.1 Technial Specification.
- [38] M. C. Jeruchim, P. Balaban and K. Sam Shanmugan, *Simulation of Communication Systems* Plenum Press 1992.

- [39] J. Medbo and P. Schramm, Ericsson Radio Systems AB, *Channel Models for HIPERLAN/2 in Different Indoor Scenarios*. ETSI EP BRAN 3ERI085B 30 March 1998.

# Tillegg A

## Programkode

Dette tillegget inneholder programkode som finnes vedlagt på CD-ROM til denne hovedfagsoppgaven. Kun koden som er benyttet for å produsere resultatene i oppgaven er vist i dette tillegget. Programkoden er skrevet i verktøyene *MATLAB* og *MAPLE*, og under følger katalogstrukturen som viser hvor filene er lagret.

```
/kode
|-- AKM
|   |-- beregn_ASE
|   |   '-- maple
|   '-- bestemme_fadingregioner
|       '-- matlab
|-- AWGN_kanalmodell
|   |-- cutoff_rate_SD
|   |   '-- matlab
|   |       '-- resultater
|   |-- felles
|   |   '-- matlab
|   '-- kanalkapasitet_MLD
|       '-- matlab
|           '-- resultater
|-- Modulasjon
|   '-- matlab
'-- OFDM
    |-- beregn_totalASE_Hiperlan2_modell
    |   |-- beregn_ASE_punkter
    |   |   '-- resultater
    |   '-- plott_ASE_punkter
    |-- beregn_totalASE_OFDM-AKM-QAM
    |   '-- maple
    '-- sammenligning_totalASE_Hiperlan2_OFDM-AKM-QAM
```

## A.1 Programkode kapittel 2

Følgende filer er inkludert:

```
/kode
  '-- Modulasjon
    '-- matlab
      |-- CROSS.m
      |-- QASK.m
      |-- STAR8.m
      |-- normaliser.m
      |-- plottKonstellasjon.m
    '-- plottMineKonstellasjoner.m

/kode/Modulasjon/matlab/CROSS.m

function [x,y] = CROSS(N,delta)
%   [x,y] = CROSS(N,delta);
%
%Genererer punktene i en N-CROSS konstellasjon.
%Minste avstand mellom punktene er (2*delta).
%Dataene lagres i to vektorer med x-koordinater i den
%første vektoren og y-koordinater i den andre vektoren.
%
%NB! N tar verdiene i rekken 32, 128, 512, 2048, ... ,
%   Dvs N = 2^(2k+1), k=2,3,4,...

x=zeros(N,1);
y=zeros(N,1);
h=1;
n = sqrt(N/2)*3/2;

for i=0:1:n-1
  for j=0:1:n-1
    xx = (2*i+1-n)*delta;
    yy = (2*j+1-n)*delta;
    if ~(xx<-(n*2/3-1)*delta&yy>(n*2/3-1)*delta) &
        ~(xx>(n*2/3-1)*delta&yy>(n*2/3-1)*delta) &
        ~(xx>(n*2/3-1)*delta&yy<-(n*2/3-1)*delta) &
        ~(xx<-(n*2/3-1)*delta&yy<-(n*2/3-1)*delta)
      x(h) = xx;
      y(h) = yy;
      h=h+1;
    end;
  end;
end;
end;
```

### /kode/Modulasjon/matlab/QASK.m

```
function [x,y] = QASK(N,delta)
% [x,y] = QASK(N,delta);
%
%Genererer punktene i en kvadratisk N-QAM konstellasjon.
%Minste avstand mellom punktene er (2*delta).
%Dataene lagres i to vektorer med x-koordinater i den
%første vektoren og y-koordinater i den andre vektoren.
x=zeros(N,1);
y=zeros(N,1);
h=1;
sqrtN = sqrt(N);
for i=0:1:sqrtN-1
    for j=0:1:sqrtN-1
        x(h) = (2*i+1-sqrtN)*delta;
        y(h) = (2*j+1-sqrtN)*delta;
        h=h+1;
    end;
end;
```

### /kode/Modulasjon/matlab/STAR8.m

```
function [x,y] = STAR8(delta)
% [x,y] = 8-STAR(delta);
%
%Genererer punktene i en 8-STAR konstellasjon.
%Minste avstand mellom punktene er (2*delta).
%Dataene lagres i to vektorer med x-koordinater i den første
%og y-koordinater i den andre vektoren.

x=zeros(8,1);
y=zeros(8,1);

[x1,y1]=QASK(4,delta);
for i=1:1:4
    x(i)=x1(i);
    y(i)=y1(i);
end;

x(5)=-delta;
y(5)=3*delta;
x(6)=3*delta;
y(6)=delta;
x(7)=delta;
y(7)=-3*delta;
x(8)=-3*delta;
y(8)=-delta;
```

### /kode/Modulasjon/matlab/normaliser.m

```
function [x,y] = normaliser(x,y,norm)
% [x,y] = normaliser(x,y,norm);
%
%Normaliserer modulasjonspunktenes avstand fra origo slik at
%den gjennomsnittlige sendeeffekten blir lik 'norm'.

S=0;
N=length(x);
for i=1:1:N
    S=S+x(i)^2+y(i)^2;
end;
S=S/N; %gj.snitt energi

n=sqrt(S/norm);
x=x./n;
y=y./n;
```

### /kode/Modulasjon/matlab/plottKonstellasjon.m

```
function f = plottKonstellasjon(x,y)
% plottKonstellasjon(x,y)
%
%Plotter en signalkonstellasjon fra vektorene x og y.

max=0;
for i=1:length(x)
    if abs(x(i))>max
        max = abs(x(i));
    end;
end;

plot(x,y,'o');
hold on,
%fplot('0',[-max-.2, max+.2])
plot([-max-.0 max+.0], [0 0], 'k');
plot([0 0], [-max-.0 max+.0], 'k');
axis equal;
set(gca,'Xlim',[-max-.2, max+.2]);
set(gca,'Ylim',[-max-.2, max+.2]);
set(gca,'Visible','off');
```

## /kode/Modulasjon/matlab/plottMineKonstellasjoner.m

```
function plottMineKonstellasjoner
%Lager plott av de ulike konstellasjonene som jeg bruker i hovedfagsoppgaven

% 4-QASK
[x,y]=QASK(4,.2);
normaliser(x,y,1);
figure
plottKonstellasjon(x,y)
set(gca,'Units','pixels')
set(gca,'Position',[50,50,52,52])

% 8-STAR
[x,y]=STAR8(.2);
normaliser(x,y,1);
figure
plottKonstellasjon(x,y)
set(gca,'Units','pixels')
set(gca,'Position',[50,50,52,52])

% 16-QASK
[x,y]=QASK(16 ,.2);
normaliser(x,y,1);
figure
plottKonstellasjon(x,y)
set(gca,'Units','pixels')
set(gca,'Position',[50,50,90,90])

% 32-CROSS
[x,y]=CROSS(32 ,.2);
normaliser(x,y,1);
figure
plottKonstellasjon(x,y)
set(gca,'Units','pixels')
set(gca,'Position',[50,50,100,100])

% 64-QASK
[x,y]=QASK(64 ,.2);
normaliser(x,y,1);
figure
plottKonstellasjon(x,y)
set(gca,'Units','pixels')
set(gca,'Position',[50,50,100,100])

% 128-CROSS
[x,y]=CROSS(128 ,.2);
normaliser(x,y,1);
figure
```

```

plottKonstellasjon(x,y)
set(gca,'Units','pixels')
set(gca,'Position',[30,30,200,200])

% 256-QASK
[x,y]=QASK(256 ,.2);
normaliser(x,y,1);
figure
plottKonstellasjon(x,y)
set(gca,'Units','pixels')
set(gca,'Position',[30,30,250,250])

% 512-CROSS
[x,y]=CROSS(512 ,.2);
normaliser(x,y,1);
figure
plottKonstellasjon(x,y)
set(gca,'Units','pixels')
set(gca,'Position',[30,30,300,300])

% 1024-QASK
[x,y]=QASK(1024 ,.2);
normaliser(x,y,1);
figure
plottKonstellasjon(x,y)
set(gca,'Units','pixels')
set(gca,'Position',[30,30,400,400])

% 2048-CROSS
[x,y]=CROSS(2048 ,.2);
normaliser(x,y,1);
figure
plottKonstellasjon(x,y)
set(gca,'Units','pixels')
set(gca,'Position',[30,30,500,500])

% 4096-QASK
[x,y]=QASK(4096 ,.2);
normaliser(x,y,1);
figure
plottKonstellasjon(x,y)
set(gca,'Units','pixels')
set(gca,'Position',[30,30,500,500])

```



## A.2 Programkode kapittel 3

Følgende filer er inkludert:

```
/kode
  '-- AWGN_kanalmodell
    |-- cutoff_rate_SD
    |   '-- matlab
    |   |-- beregnAlle.m
    |   |-- beregnCutoffRate.m
    |   |-- cutoffRatePlott.m
    |   |-- genererPlottPunkt.m
    |-- felles
    |   '-- matlab
    |   |-- lesPunkterFraFil.m
    |   |-- skrivPunkterTilFil.m
    |-- kanalkapasitet_MLD
    |   '-- matlab
    |   |-- beregnAlle.m
    |   |-- beregnKanalkapasitet.m
    |   |-- genererPlottPunkt.m
    |-- C_stjerne_og_R0_stjerne_diffplt.m
    '-- C_stjerne_og_R0_stjerne_plott.m
```

`/kode/AWGN_kanalmodell/cutoff_rate_SD/matlab/beregnAlle.m`

```
function f = beregnAlle

beregnCutoffRate(4, 'QAM', [0,0.5,40]);
beregnCutoffRate(8, 'STAR', [0,0.5,40]);
beregnCutoffRate(16, 'QAM', [0,0.5,40]);
beregnCutoffRate(32, 'CROSS', [0,0.5,40]);
beregnCutoffRate(64, 'QAM', [0,0.5,40]);
beregnCutoffRate(128, 'CROSS', [0,0.5,40]);
beregnCutoffRate(256, 'QAM', [0,0.5,40]);
beregnCutoffRate(512, 'CROSS', [0,0.5,40]);
beregnCutoffRate(1024, 'QAM', [0,0.5,45]);
beregnCutoffRate(2048, 'CROSS', [0,0.5,45]);
beregnCutoffRate(4096, 'QAM', [0,0.5,50]);
```

/kode/AWGN\_kanalmodell/cutoff\_rate\_SD/matlab/beregnCutoffRate.m

```
function f = beregnCutoffRate(N,type,dB_Verdier)
%   beregnCutoffRate(N,type,dB_Verdier)
%
%Beregner kanalkapasiteten til en båndbreddebegrenset flat-fadende
%AWGN kanal når det benyttes Sekvensiell Dekoding.
%
%Innvariable:
%   N = konstallasjonstørrelsen
%   type = konstallasjonstype, f.eks. 'QAM', 'CROSS', 'STAR'
%   dB_Verdier = vektor med SNR-verdier en vil beregne kapasiteten for
%
%Eksempel:   beregnKanalkapasitet(16,'QAM',[0,0.5,25]);
%   Beregner kanalkapasiteten når det benyttes 16-QAM
%   for SNR_dB i (0.0, 0.5, 1.0,..., 25.0) og skriver dataene til
%   filen   'resultater/16-QAM_SNR=0-25.dat'

path(path, '..../felles/matlab');
path(path, '..../Modulasjon/matlab');

dBmin=dB_Verdier(1);
dBmax=dB_Verdier(length(dB_Verdier));
dBstep=0;
if length(dB_Verdier) > 1
    dBstep=dB_Verdier(2)-dB_Verdier(1);
end;

if strcmp(type,'CROSS')==1
    [x,y]=CROSS(N,0.2);
elseif strcmp(type,'QAM')==1
    [x,y]=QASK(N,0.2);
elseif strcmp(type,'QASK')==1
    [x,y]=QASK(N,0.2);
elseif strcmp(type,'STAR')==1
    [x,y]=STAR8(0.2);
    N=8;
elseif strcmp(type,'STAR8')==1
    [x,y]=STAR8(0.2);
else
    disp('Ukjent konstallasjon');
end

%Koden under vil ikke eksekvere hvis ikke x eller y er definert
[x,y]=normaliser(x,y,1);
[s,t]=genererPlottPunkt(x,y,dBmin,dBmax,dBstep);
skrivPunkterTilFil(s,t, strcat('resultater/',num2str(N),'-',type,
    '_SNR=',num2str(dBmin),'-',num2str(dBmax),'.dat'));
```

`/kode/AWGN_kanalmodell/cutoff_rate_SD/matlab/cutoffRatePlott.m`

```
function f = cutoffRatePlott
```

```
path(path, '../..//felles/matlab')
```

```
[x1,y1] = lesPunkterFraFil('data/4-QAM.dat');  
[x2,y2] = lesPunkterFraFil('data/8-STAR.dat');  
[x3,y3] = lesPunkterFraFil('data/16-QAM.dat');  
[x4,y4] = lesPunkterFraFil('data/32-CROSS.dat');  
[x5,y5] = lesPunkterFraFil('data/64-QAM.dat');  
[x6,y6] = lesPunkterFraFil('data/128-CROSS.dat');  
[x7,y7] = lesPunkterFraFil('data/256-QAM.dat');  
[x8,y8] = lesPunkterFraFil('data/512-CROSS.dat');  
[x9,y9] = lesPunkterFraFil('data/1024-QAM.dat');  
[x10,y10] = lesPunkterFraFil('data/2048-CROSS.dat');  
[x11,y11] = lesPunkterFraFil('data/4096-QAM.dat');
```

```
x1 = x1(1:37);  
y1 = y1(1:37);  
x2 = x2(12:43);  
y2 = y2(12:43);  
x3 = x3(17:51);  
y3 = y3(17:51);  
x4 = x4(25:55);  
y4 = y4(25:55);  
x5 = x5(32:61);  
y5 = y5(32:61);  
x6 = x6(38:67);  
y6 = y6(38:67);  
x7 = x7(45:73);  
y7 = y7(45:73);  
x8 = x8(50:79);  
y8 = y8(50:79);  
x9 = x9(59:79);  
y9 = y9(59:79);  
x10 = x10(63:90);  
y10 = y10(63:90);  
x11 = x11(70:100);  
y11 = y11(70:100);
```

```
plot(x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6,x7,y7,x8,y8,x9,y9,x10,y10,x11,y11);  
ylim([0 13]);  
xlim([0 50]);  
xlabel('\gamma (SNR) [dB]');  
ylabel('[BITS/T]');  
grid;
```

```
text(16,2.2,'4-QAM');
```

```

text(18.5,3.2,'8-STAR');
text(21,4.2,'16-QAM');
text(23.5,5.2,'32-CROSS');
text(26,6.2,'64-QAM');
text(28.5,7.2,'128-CROSS');
text(32.5,8.2,'256-QAM');
text(36,9.2,'512-CROSS');
text(38.5,10.2,'1024-QAM');
text(40.5,11.2,'2048-CROSS');
text(42.5,12.2,'4096-QAM');

```

**/kode/AWGN\_kanalmodell/cutoff\_rate\_SD/matlab/genererPlottPunkt.m**

```

function [x,y] = genererPlottPunkt(a1,a2,dBmin,dBmax,dBstep)
%Beregner uttrykk (3.19) for oppgitte SNR-verdier

N = length(a1);
if dBstep==0
    n=1;
else
    n = (dBmax-dBmin)/dBstep + 1;
end;
x=zeros(n,1);
y=zeros(n,1);

dim=1;
for i=2:1:N
    if a2(i)~=a2(1)
        dim=2;
    end;
end;
dim

for b=1:1:n
    disp(strcat(num2str(b),'/',num2str(n)));
    dB = dBmin + (b-1)*dBstep;
    sigma = sqrt(1/(2*10^(dB/10)));
    sigma2 = sigma^2;
    sigma2dim = sigma^2*dim;
    S=0;
    for i=1:1:N
        s = exp(- ((a1(i) - a1(:)).^2 + (a2(i)-a2(:)).^2)/(8*sigma2));
        S = S+sum(s);
    end;
    x(b) = dB;
    y(b) = 2*log2(N) - log2(S);
end;

```

### **/kode/AWGN\_kanalmodell/felles/matlab/skrivPunkterTilFil.m**

```
function f = skrivPunkterTilFil(s,t,filNavn)
%   skrivPunkterTilFil(s,t,filNavn);
%
%Skriver (x,y)-koordinater til fil.
%Funksjonen vil lage en fil, 'filNavn.dat',
%som inneholder en tekstformatering av dataene.
%
%Innvariable:
%   s = KOLONNEvektor med x-koordinater
%   t = KOLONNEvektor med y-koordinater
%   filNavn = foerste del av filens navn

fid = fopen(strcat(filNavn, '.dat'), 'w');
fprintf(fid, '%2.1f\t%2.14f\n', [s';t']);
fclose(fid);
```

### **/kode/AWGN\_kanalmodell/felles/matlab/lesPunkterFraFil.m**

```
function [s,t] = lesPunkterFraFil(filnavn)
%   [s,t] = lesPunkterFraFil(filnavn);
%
%Leser punkter fra filer generert av funksjonen 'skrivPunkterTilFil'

if length(findstr('.dat',filnavn)) ~= 0
    fid = fopen(filnavn);
    a=fscanf(fid, '%g %g', [2 inf]);
    a=a';
    s=a(:,1);
    t=a(:,2);
else
    error('Innputtfil maa vaere av typen .dat !');
end;
```

## /kode/AWGN\_kanalmodell/kanalkapasitet\_MLD/matlab/beregnAlle.m

```
function f = beregnAlle
% Denne funksjonen ble laget i ettertid for å illustrere
% beregningene som ble gjort av kanalkapasiteten ved MLD.
% NB! Det er ikke anbefalt å kjøre denne direkte pga
% veldig lang kjøretid!!!
```

```
beregnKanalkapasitet(4,'QAM',[0,0.5,20],1000000,1);
beregnKanalkapasitet(8,'STAR',[0,0.5,21],1000000,1);
beregnKanalkapasitet(16,'QAM',[0,0.5,25],100000,1);
beregnKanalkapasitet(32,'CROSS',[0,0.5,27],100000,1);
beregnKanalkapasitet(64,'QAM',[0,0.5,64],100000,1);
beregnKanalkapasitet(128,'CROSS',[0,0.5,33],100000,1);
beregnKanalkapasitet(256,'QAM',[0,0.5,36],10000,1);
beregnKanalkapasitet(512,'CROSS',[0,0.5,39],10000,1);
beregnKanalkapasitet(1024,'QAM',[0,0.5,40],1000,1);
beregnKanalkapasitet(2048,'CROSS',[0,0.5,40],1000,1);
beregnKanalkapasitet(4096,'QAM',[0,0.5,41.5],1000,1);
```

## /kode/AWGN\_kanalmodell/kanalkapasitet\_MLD/matlab/beregnKanalkapasitet.m

```
function f = beregnKanalkapasitet(N,type,dB_Verdier,l,nr)
% beregnKanalkapasitet(N,type,dB_Verdier,l,nr)
%
% Beregner kanalkapasiteten til en båndbreddebegrenset flat-fadende
% AWGN kanal når det benyttes Maximum Likelihood Decoding.
%
% Innvariable:
% N = konstellasjonstørrelsen
% type = konstellasjonstype, f.eks. 'QAM', 'CROSS', 'STAR'
% dB_Verdier = vektor med SNR-verdier en vil beregne kapasiteten for
% l = antall iterasjoner for å bestemme forventning
% nr = en identifikator for en kjøring
%
% Eksempel: beregnKanalkapasitet(16,'QAM',[0,0.5,25],100000,1);
% Beregner kanalkapasiteten når det benyttes 16-QAM
% for SNR_dB i (0.0, 0.5, 1.0,..., 25.0) og skriver dataene til
% filen 'resultater/16-QAM_SNR=0-25_l=100000_nr=1.dat'

path(path, '..../felles/matlab');
path(path, '..../Modulasjon/matlab');

dBmin=dB_Verdier(1);
dBmax=dB_Verdier(length(dB_Verdier));
dBstep=0;
if length(dB_Verdier) > 1
    dBstep=dB_Verdier(2)-dB_Verdier(1);
```

```

end;

if strcmp(type,'CROSS')==1
    [x,y]=CROSS(N,0.2);
elseif strcmp(type,'QAM')==1
    [x,y]=QASK(N,0.2);
elseif strcmp(type,'QASK')==1
    [x,y]=QASK(N,0.2);
elseif strcmp(type,'STAR')==1
    [x,y]=STAR8(0.2);
    N=8;
elseif strcmp(type,'STAR8')==1
    [x,y]=STAR8(0.2);
else
    disp('Ukjent konstellasjon');
end

%Koden under vil ikke eksekvere hvis ikke x eller y er definert
[x,y]=normaliser(x,y,1);
[s,t]=genererPlottPunkt(x,y,dBmin,dBmax,dBstep,1);

skrivPunkterTilFil(s,t, strcat('resultater/',num2str(N),'-',type,'_SNR=',
    num2str(dBmin),'-',num2str(dBmax),'_l=',num2str(1),'_nr=',num2str(nr)) );

```

```

/kode/AWGN_kanalmodell/kanalkapasitet_MLD/
matlab/genererPlottPunkt.m

```

```

function [x,y] = genererPlottPunkt(a1,a2,dBmin,dBmax,dBstep,1)
%Beregner uttrykk (3.18) i oppgaven for oppgitte SNR-verdier

%Setter støy-generatoren til forskjellig tilstand hver gang programmet kjøres
randn('state',sum(100*clock));

N = length(a1);
if dBstep==0
    n=1;
else
    n = (dBmax-dBmin)/dBstep + 1;
end;

x=zeros(n,1);
y=zeros(n,1);

dim=1;
for i=2:1:N
    if a2(i)~=a2(1)
        dim=2;
    end;
end;
dim

for b=1:1:n
    disp(strcat(num2str(b),',',num2str(n)));
    dB = dBmin + (b-1)*dBstep;
    sigma = sqrt(1/(2*10^(dB/10)));
    sigma2 = sigma^2;
    sigma2dim = sigma2*dim;
    F=0;
    for k=1:1:N
        E=0;
        w1 = sigma * randn(N,1);
        w2 = sigma * randn(N,1);
        for j=1:1:l
            e = exp(-( (a1(k)+w1(:,j))-a1(:)).^2 + (a2(k)+w2(:,j))-a2(:)).^2
                    - w1(:,j).^2 - w2(:,j).^2 )/sigma2dim);
            E = E + log2(sum(e));
        end;
        E = E / l;
        F = F + E;
    end;
    x(b) = dB;
    y(b) = log2(N) - F/N;
end;

```



## /kode/AWGN\_kanalmodell/C\_stjerne\_og\_R0\_stjerne\_diffplt.m

```
function f = C_stjerne_og_R0_stjerne_diffplt
path(path, 'felles/matlab');

[x1,y1] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/4-QAM.dat');
[x2,y2] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/8-STAR.dat');
[x3,y3] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/16-QAM.dat');
[x4,y4] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/32-CROSS.dat');
[x5,y5] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/64-QAM.dat');
[x6,y6] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/128-CROSS.dat');
[x7,y7] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/256-QAM.dat');
[x8,y8] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/512-CROSS.dat');
[x9,y9] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/1024-QAM.dat');
[x10,y10] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/2048-CROSS.dat');
[x11,y11] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/4096-QAM.dat');

[u1,v1] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/4-QAM.dat');
[u2,v2] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/8-STAR.dat');
[u3,v3] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/16-QAM.dat');
[u4,v4] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/32-CROSS.dat');
[u5,v5] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/64-QAM.dat');
[u6,v6] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/128-CROSS.dat');
[u7,v7] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/256-QAM.dat');
[u8,v8] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/512-CROSS.dat');
[u9,v9] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/1024-QAM.dat');
[u10,v10] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/2048-CROSS.dat');
[u11,v11] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/4096-QAM.dat');

if length(v1)>length(y1)
    y1=[y1;zeros([length(v1)-length(y1) 1])];
elseif length(v1)<length(y1)
    v1=[v1;zeros([length(y1)-length(v1) 1])];
    u1=x1;
end;

if length(v2)>length(y2)
    y1=[y2;zeros([length(v2)-length(y2) 1])];
elseif length(v2)<length(y2)
    v2=[v2;zeros([length(y2)-length(v2) 1])];
    u2=x2;
end;

if length(v3)>length(y3)
    y3=[y3;zeros([length(v3)-length(y3) 1])];
elseif length(v3)<length(y3)
    v3=[v3;zeros([length(y3)-length(v3) 1])];
    u3=x3;
end;
```

```

if length(v4)>length(y4)
    y4=[y4;zeros([length(v4)-length(y4) 1])];
elseif length(v4)<length(y4)
    v4=[v4;zeros([length(y4)-length(v4) 1])];
    u4=x4;
end;

if length(v5)>length(y5)
    y5=[y5;zeros([ length(v5)-length(y) 1])];
elseif length(v5)<length(y5)
    v5=[v5;zeros([length(y5)-length(v5) 1])];
    u5=x5;
end;

if length(v6)>length(y6)
    y6=[y6;zeros([length(v6)-length(y6) 1])];
elseif length(v6)<length(y6)
    v6=[v6;zeros([length(y6)-length(v6) 1])];
    u6=x6;
end;

if length(v7)>length(y7)
    y7=[y7;zeros([length(v7)-length(y7) 1])];
elseif length(v7)<length(y7)
    v7=[v7;zeros([length(y7)-length(v7) 1])];
    u7=x7;
end;

if length(v8)>length(y8)
    y8=[y8;zeros([length(v8)-length(y8) 1])];
elseif length(v8)<length(y8)
    v8=[v8;zeros([length(y8)-length(v8) 1])];
    u8=x8;
end;

if length(v9)>length(y9)
    y9=[y9;zeros([length(v9)-length(y9) 1])];
elseif length(v9)<length(y9)
    v9=[v9;zeros([length(y9)-length(v9) 1])];
    u9=x9;
end;

if length(v10)>length(y10)
    y10=[y10;zeros([length(v10)-length(y10) 1])];
elseif length(v10)<length(y10)
    v10=[v10;zeros([length(y10)-length(v10) 1])];
    u10=x10;
end;

```

```

if length(v11)>length(y11)
    y11=[y11;zeros([length(v11)-length(y11) 1])];
elseif length(v11)<length(y11)
    v11=[v11;zeros([length(y11)-length(v11) 1])];
    u11=x11;
end;

z1=v1-y1;
z2=v2-y2;
z3=v3-y3;
z4=v4-y4;
z5=v5-y5;
z6=v6-y6;
z7=v7-y7;
z8=v8-y8;
z9=v9-y9;
z10=v10-y10;
z11=v11-y11;

u1 = u1(1:37);
z1 = z1(1:37);
u2 = u2(12:43);
z2 = z2(12:43);
u3 = u3(17:51);
z3 = z3(17:51);
u4 = u4(25:55);
z4 = z4(25:55);
u5 = u5(32:61);
z5 = z5(32:61);
u6 = u6(38:67);
z6 = z6(38:67);
u7 = u7(45:73);
z7 = z7(45:73);
u8 = u8(50:79);
z8 = z8(50:79);
u9 = u9(57:80);
z9 = z9(57:80);
u10 = u10(63:80);
z10 = z10(63:80);
u11 = u11(69:84);
z11 = z11(69:84);

plot(u1,z1,'k',u2,z2,'b',u3,z3,'g',u4,z4,'r',u5,z5,'c',u6,z6,'m',
     u7,z7,'y',u8,z8,'k',u9,z9,'g',u10,z10,'r',u11,z11,'b');
ylim([-0.05 0.45]);
xlim([0 42]);
xlabel('\gamma [dB]');
ylabel('[BITS/T]');

```

```

grid;

rot=-75;
text(4.5,0.23,'4-QAM','rotation',rot,'fontsize',7);
text(9,0.26,'8-STAR','rotation',rot,'fontsize',7);
text(12,0.28,'16-QAM','rotation',rot,'fontsize',7);
text(15,0.29,'32-CROSS','rotation',rot,'fontsize',7);
text(18,0.3,'64-QAM','rotation',rot,'fontsize',7);
text(21,0.31,'128-CROSS','rotation',rot,'fontsize',7);
text(24,0.32,'256-QAM','rotation',rot,'fontsize',7);
text(27,0.33,'512-CROSS','rotation',rot,'fontsize',7);
text(30,0.34,'1024-QAM','rotation',rot,'fontsize',7);
text(33,0.35,'2048-CROSS','rotation',rot,'fontsize',7);
text(36,0.36,'4096-QAM','rotation',rot,'fontsize',7);

set(gca,'Units','pixels')
set(gca,'Position',[74,46,434,190])

```

/kode/AWGN\_kanalmodell/C\_stjerne\_og\_R0\_stjerne\_plott.m

```

function f = C_stjerne_og_R0_stjerne_plott

path(path, 'felles/matlab');

[x1,y1] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/4-QAM.dat');
[x2,y2] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/8-STAR.dat');
[x3,y3] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/16-QAM.dat');
[x4,y4] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/32-CROSS.dat');
[x5,y5] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/64-QAM.dat');
[x6,y6] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/128-CROSS.dat');
[x7,y7] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/256-QAM.dat');
[x8,y8] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/512-CROSS.dat');
[x9,y9] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/1024-QAM.dat');
[x10,y10] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/2048-CROSS.dat');
[x11,y11] = lesPunkterFraFil('cutoff_rate_SD/matlab/resultater/4096-QAM.dat');

x1 = x1(1:37);
y1 = y1(1:37);
x2 = x2(12:43);
y2 = y2(12:43);
x3 = x3(17:51);
y3 = y3(17:51);
x4 = x4(25:55);
y4 = y4(25:55);
x5 = x5(32:61);
y5 = y5(32:61);
x6 = x6(38:67);
y6 = y6(38:67);
x7 = x7(45:73);

```

```

y7 = y7(45:73);
x8 = x8(50:79);
y8 = y8(50:79);
x9 = x9(59:79);
y9 = y9(59:79);
x10 = x10(63:90);
y10 = y10(63:90);
x11 = x11(70:94);
y11 = y11(70:94);

[u1,v1] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/4-QAM.dat');
[u2,v2] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/8-STAR.dat');
[u3,v3] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/16-QAM.dat');
[u4,v4] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/32-CROSS.dat');
[u5,v5] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/64-QAM.dat');
[u6,v6] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/128-CROSS.dat');
[u7,v7] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/256-QAM.dat');
[u8,v8] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/512-CROSS.dat');
[u9,v9] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/1024-QAM.dat');
[u10,v10] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/2048-CROSS.dat');
[u11,v11] = lesPunkterFraFil('kanalkapasitet_MLD/matlab/resultater/4096-QAM.dat');

u1 = u1(1:37);
v1 = v1(1:37);
u2 = u2(12:43);
v2 = v2(12:43);
u3 = u3(17:51);
v3 = v3(17:51);
u4 = u4(25:55);
v4 = v4(25:55);
u5 = u5(32:61);
v5 = v5(32:61);
u6 = u6(38:67);
v6 = v6(38:67);
u7 = u7(45:73);
v7 = v7(45:73);
u8 = u8(50:79);
v8 = v8(50:79);
u9 = u9(57:81);
v9 = v9(57:81);
u10 = u10(62:81);
v10 = v10(62:81);
u11 = u11(69:84);
v11 = v11(69:84);

plot(u1,v1,'k--',x1,y1,'k',x2,y2,'b',x3,y3,'g',x4,y4,'r',x5,y5,'c',
      x6,y6,'m',x7,y7,'y',x8,y8,'k',x9,y9,'g',u2,v2,'b--',u3,v3,'g--');%
ylim([0 13]);
xlim([0 50]);

```

```

xlabel('\gamma [dB]');
ylabel('[BITS/T]');
grid;

text(16,2.2,'4-QAM');
text(18.5,3.2,'8-STAR');
text(21,4.2,'16-QAM');
text(23.5,5.2,'32-CROSS');
text(26,6.2,'64-QAM');
text(28.5,7.2,'128-CROSS');
text(32.5,8.2,'256-QAM');
text(36,9.2,'512-CROSS');
text(38.5,10.2,'1024-QAM');
text(40.5,11.2,'2048-CROSS');
text(42.5,12.2,'4096-QAM');

hold on;
plot(x11,y11,'b',x10,y10,'r',u10,v10,'r--',u11,v11,'b--',u4,v4,'r--',
      u5,v5,'c--',u6,v6,'m--',u7,v7,'y--',u8,v8,'k--',u9,v9,'g--');
hold off;

```

## A.3 Programkode kapittel 4

Følgende filer er inkludert:

```
/kode
  '-- AKM
    |-- bestemme_fadingregioner
    |   '-- matlab
    |       |-- bestemSNR.m
    |       '-- bestemFadingRegioner.m
  '-- beregn_ASE
    '-- maple
        |-- beregn_ASE_C_stjerne_c1_L1.mws
        |-- beregn_ASE_C_stjerne_c1_L2.mws
        |-- beregn_ASE_R0_stjerne_c1_L1.mws
        |-- beregn_ASE_R0_stjerne_c1_L2.mws
        |-- beregn_diff_ASE_c1_L1.mws
        '-- beregn_diff_ASE_c1_L2.mws
```

## /kode/AKM/bestemme\_fadingregioner/matlab/bestemSNR.m

```
function SNR = bestemSNR(spline1,SEO,dBmin,dBmax,tol)
%   SNR = bestemSNR(spline1,SEO,dBmin,dBmax)
%
%Estimerer SNR for target spektral effektivitet SEO vha intervallhalvering
%
%Innvariable:
%   spline1      = et spline som estimerer funksjonen
%   SEO          = target spektral effektivitet for hvor SNR skal bestemmes
%   dBmin,dBmax  = grenser for hvor soeket etter SNR skal begynne.
%                 Tips: dBmin og dBmax kan finnes ved aa se paa plottet
%                 av den estimerende funksjonen. (dBmin<SNR<dBmax).
%                 Funksjon kan plottes ved flg kommando: fnplt(spline1);
%   tol          = toleranseverdi for treff i halveringssoeket
%
%   NB! Det er antatt at spline1 er ikke-avtagende i [dBmin,dBmax]

if nargin ~= 5
    error('Ugyldig antall innvariable. Skriv <help bestemSNR> for hjelp!')
end;

if fnval(spline1, dBmin)> SEO | fnval(spline1,dBmax) < SEO
    error(strcat('Oensket SNR finnes ikke i intervallet [', num2str(dBmin),
        ', ', num2str(dBmax),']'))
end;

SNR = (dBmin+dBmax)/2;
SE = fnval(spline1,SNR);
while abs(SE-SEO)>tol | (dBmax-dBmin)>tol
    SNR = (dBmin+dBmax)/2;
    SE = fnval(spline1,SNR);

    if SE<SEO
        dBmin=SNR;
    else
        if SE>SEO
            dBmax=SNR;
        end;
    end;
end;
end;
```



**/kode/AKM/bestemme\_fadingregioner/  
matlab/bestemFadingRegioner.m**

```
function f = bestemFadingRegioner
% Denne funksjonen ble skrevet i ettertid for å vise
% hvordan fadingregionegrensene i tabell 4.1 ble beregnet.
% Se seksjon 4.2 i oppgaven for detaljert fremgangsmåte.

path(path, '../.../AWGN_kanalmodell/felles/matlab');

disp('Maximum Likelihood Decoding:');

[u1,v1] = lesPunkterFraFil('../.../AWGN_kanalmodell/
                           kanalkapasitet_MLD/matlab/resultater/4-QAM.dat');
sp = spap2(1,14,u1,v1);
disp(strcat('4-QAM: SE=1.0 SNR=',num2str(bestemSNR(sp,1.0,0,2,1000*eps))));
disp(strcat('4-QAM: SE=1.5 SNR=',num2str(bestemSNR(sp,1.5,2,4,1000*eps))));

[u2,v2] = lesPunkterFraFil('../.../AWGN_kanalmodell/
                           kanalkapasitet_MLD/matlab/resultater/8-STAR.dat');
sp = spap2(1,14,u2,v2);
disp(strcat('8-STAR: SE=2.0 SNR=',num2str(bestemSNR(sp,2.0,5,7,1000*eps))));
disp(strcat('8-STAR: SE=2.5 SNR=',num2str(bestemSNR(sp,2.5,7,9,1000*eps))));

[u3,v3] = lesPunkterFraFil('../.../AWGN_kanalmodell/
                           kanalkapasitet_MLD/matlab/resultater/16-QAM.dat');
sp = spap2(1,13,u3,v3);
disp(strcat('16-QAM: SE=3.0 SNR=',num2str(bestemSNR(sp,3.0,9,11,1000*eps))));
disp(strcat('16-QAM: SE=3.5 SNR=',num2str(bestemSNR(sp,3.5,11,13,1000*eps))));

[u4,v4] = lesPunkterFraFil('../.../AWGN_kanalmodell/
                           kanalkapasitet_MLD/matlab/resultater/32-CROSS.dat');
sp = spap2(1,14,u4,v4);
disp(strcat('32-CROSS: SE=4.0 SNR=',num2str(bestemSNR(sp,4.0,12,14,1000*eps))));
disp(strcat('32-CROSS: SE=4.5 SNR=',num2str(bestemSNR(sp,4.5,14,16,1000*eps))));

[u5,v5] = lesPunkterFraFil('../.../AWGN_kanalmodell/
                           kanalkapasitet_MLD/matlab/resultater/64-QAM.dat');
sp = spap2(1,14,u5,v5);
disp(strcat('64-QAM: SE=5.0 SNR=',num2str(bestemSNR(sp,5.0,16,17,1000*eps))));
disp(strcat('64-QAM: SE=5.5 SNR=',num2str(bestemSNR(sp,5.5,17,19,1000*eps))));

[u6,v6] = lesPunkterFraFil('../.../AWGN_kanalmodell/
                           kanalkapasitet_MLD/matlab/resultater/128-CROSS.dat');
sp = spap2(1,14,u6,v6);
disp(strcat('128-CROSS: SE=6.0 SNR=',num2str(bestemSNR(sp,6.0,19,20,1000*eps))));
disp(strcat('128-CROSS: SE=6.5 SNR=',num2str(bestemSNR(sp,6.5,20,22,1000*eps))));
```

```

[u7,v7] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        kanalkapasitet_MLD/matlab/resultater/256-QAM.dat');
sp = spap2(1,14,u7,v7);
disp(strcat('256-QAM: SE=7.0 SNR=',num2str(bestemSNR(sp,7.0,22,24,1000*eps))));
disp(strcat('256-QAM: SE=7.5 SNR=',num2str(bestemSNR(sp,7.5,24,25,1000*eps))));

[u8,v8] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        kanalkapasitet_MLD/matlab/resultater/512-CROSS.dat');
sp = spap2(1,14,u8,v8);
disp(strcat('512-CROSS: SE=8.0 SNR=',num2str(bestemSNR(sp,8.0,25,27,1000*eps))));
disp(strcat('512-CROSS: SE=8.5 SNR=',num2str(bestemSNR(sp,8.5,27,29,1000*eps))));

[u9,v9] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        kanalkapasitet_MLD/matlab/resultater/1024-QAM.dat');
sp = spap2(1,14,u9,v9);
disp(strcat('1024-QAM: SE=9.0 SNR=',num2str(bestemSNR(sp,9.0,28,30,1000*eps))));
disp(strcat('1024-QAM: SE=9.5 SNR=',num2str(bestemSNR(sp,9.5,30,32,1000*eps))));

[u10,v10] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        kanalkapasitet_MLD/matlab/resultater/2048-CROSS.dat');
sp = spap2(1,14,u10,v10);
disp(strcat('2048-CROSS: SE=10.0 SNR=',num2str(bestemSNR(sp,10.0,31,33,1000*eps))));
disp(strcat('2048-CROSS: SE=10.5 SNR=',num2str(bestemSNR(sp,10.5,32,34,1000*eps))));

[u11,v11] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        kanalkapasitet_MLD/matlab/resultater/4096-QAM.dat');
sp = spap2(1,14,u11,v11);
disp(strcat('4096-QAM: SE=11.0 SNR=',num2str(bestemSNR(sp,11.0,34,36,1000*eps))));
disp(strcat('4096-QAM: SE=11.5 SNR=',num2str(bestemSNR(sp,11.5,36,38,1000*eps))));

disp(' ');
disp('Sekvensiell dekoding:');
[x1,y1] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        cutoff_rate_SD/matlab/resultater/4-QAM.dat');
sp = spap2(1,14,x1,y1);
disp(strcat('4-QAM: SE=1.0 SNR=',num2str(bestemSNR(sp,1.0,2,3,1000*eps))));
disp(strcat('4-QAM: SE=1.5 SNR=',num2str(bestemSNR(sp,1.5,4,6,1000*eps))));

[x2,y2] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        cutoff_rate_SD/matlab/resultater/8-STAR.dat');
sp = spap2(1,14,x2,y2);
disp(strcat('8-STAR: SE=2.0 SNR=',num2str(bestemSNR(sp,2.0,7,9,1000*eps))));
disp(strcat('8-STAR: SE=2.5 SNR=',num2str(bestemSNR(sp,2.5,9,11,1000*eps))));

[x3,y3] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        cutoff_rate_SD/matlab/resultater/16-QAM.dat');
sp = spap2(1,14,x3,y3);
disp(strcat('16-QAM: SE=3.0 SNR=',num2str(bestemSNR(sp,3.0,10,12,1000*eps))));
disp(strcat('16-QAM: SE=3.5 SNR=',num2str(bestemSNR(sp,3.5,12,14,1000*eps))));

```

```

[x4,y4] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        cutoff_rate_SD/matlab/resultater/32-CROSS.dat');
sp = spap2(1,14,x4,y4);
disp(strcat('32-CROSS: SE=4.0 SNR=',num2str(bestemSNR(sp,4.0,13,15,1000*eps))));
disp(strcat('32-CROSS: SE=4.5 SNR=',num2str(bestemSNR(sp,4.5,15,17,1000*eps))));

[x5,y5] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        cutoff_rate_SD/matlab/resultater/64-QAM.dat');
sp = spap2(1,14,x5,y5);
disp(strcat('64-QAM: SE=5.0 SNR=',num2str(bestemSNR(sp,5.0,17,19,1000*eps))));
disp(strcat('64-QAM: SE=5.5 SNR=',num2str(bestemSNR(sp,5.5,19,21,1000*eps))));

[x6,y6] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        cutoff_rate_SD/matlab/resultater/128-CROSS.dat');
sp = spap2(1,14,x6,y6);
disp(strcat('128-CROSS: SE=6.0 SNR=',num2str(bestemSNR(sp,6.0,20,22,1024*eps))));
disp(strcat('128-CROSS: SE=6.5 SNR=',num2str(bestemSNR(sp,6.5,22,24,1024*eps))));

[x7,y7] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        cutoff_rate_SD/matlab/resultater/256-QAM.dat');
sp = spap2(1,14,x7,y7);
disp(strcat('256-QAM: SE=7.0 SNR=',num2str(bestemSNR(sp,7.0,23,25,1000*eps))));
disp(strcat('256-QAM: SE=7.5 SNR=',num2str(bestemSNR(sp,7.5,25,27,1000*eps))));

[x8,y8] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        cutoff_rate_SD/matlab/resultater/512-CROSS.dat');
sp = spap2(1,14,x8,y8);
disp(strcat('512-CROSS: SE=8.0 SNR=',num2str(bestemSNR(sp,8.0,26,28,1000*eps))));
disp(strcat('512-CROSS: SE=8.5 SNR=',num2str(bestemSNR(sp,8.5,28,30,1000*eps))));

[x9,y9] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        cutoff_rate_SD/matlab/resultater/1024-QAM.dat');
sp = spap2(1,14,x9,y9);
disp(strcat('1024-QAM: SE=9.0 SNR=',num2str(bestemSNR(sp,9.0,29,31,1000*eps))));
disp(strcat('1024-QAM: SE=9.5 SNR=',num2str(bestemSNR(sp,9.5,31,33,1000*eps))));

[x10,y10] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        cutoff_rate_SD/matlab/resultater/2048-CROSS.dat');
sp = spap2(1,19,x10,y10);
disp(strcat('2048-CROSS: SE=10.0 SNR=',num2str(bestemSNR(sp,10.0,32,34,1024*eps))));
disp(strcat('2048-CROSS: SE=10.5 SNR=',num2str(bestemSNR(sp,10.5,34,36,1024*eps))));

[x11,y11] = lesPunkterFraFil('.../.../AWGN_kanalmodell/
                        cutoff_rate_SD/matlab/resultater/4096-QAM.dat');
sp = spap2(1,14,x11,y11);
disp(strcat('4096-QAM: SE=11.0 SNR=',num2str(bestemSNR(sp,11.0,35,37,1000*eps))));
disp(strcat('4096-QAM: SE=11.5 SNR=',num2str(bestemSNR(sp,11.5,37,39,1000*eps))));

```

```
/kode/AKM/beregn_ASE/maple/beregn_ASE_C_stjerne_c1_L1.mws
```

```
>restart;
```

```
Sannsynlighet for at gamma havner i n-te fading region.
```

```
>probNakagami := proc(g,g1,g2,m)
> (GAMMA(m,m*g1/g)-GAMMA(m,m*g2/g))/GAMMA(m);
>end proc;
```

```
Prosedyre for konvertere vekk dB
```

```
>remove_dB := proc(x_dB)
> 10 ^ (x_dB/10);
>end;
```

```
Inndataene:
```

```
>SE := [1,2,3,4,5,6,7,8,9,10,11]:
>gMLD_dB := [0.83,6.14,9.77,13.12,16.55,19.59,22.86,25.79,29.0,31.9,35.1,1000]:
```

```
Konverter vekk dB-en...
```

```
>for i from 1 to 12 do
> gMLD[i]:=remove_dB(gMLD_dB[i]);
>end do;
```

```
Prosedyre som beregner ASE:
```

```
>ASE := proc(g_dB,m)
> local g,s,i;
> g := remove_dB(g_dB);
> s := 0;
> for i from 1 to 11 do
> s:= s + SE[i] * probNakagami(g,gMLD[i],gMLD[i+1],m);;
> end do;
> s;
>end proc;
```

```
Plottet skrives til filen "ASE_C_stjerne_c1_L1.eps" dersom denne aktiveres.
```

```
>plotsetup(cps,plotoutput="ASE_C_stjerne_c1_L1.eps", plotoptions="portrait,
height=170pt,width=400pt,noborder,leftmargin=0,bottommargin=0");
```

```
Genererer ASE-plottet for c=L=1 og m={1,2,4} for MLD.
```

```
>with(plots):
>F:=plot({ASE(x,1),ASE(x,2),ASE(x,4)},x=10..29, color=[red,green,blue], axes=
BOXED, labels=["dB","bits/s/Hz"], labeldirections=[horizontal,vertical]):
>G1:=textplot([17,3.7,'m=1']):
>G2:=textplot([18,4.6,'m=2']):
>G4:=textplot([18.5,5.3,'m=4']):
>display(F,G1,G2,G4);
>plotsetup(default);
```

```
/kode/AKM/beregn_ASE/maple/beregn_ASE_C_stjerne_c1_L2.mws
```

```
>restart;
```

```
Sannsynlighet for at gamma havner i n-te fading region.
```

```
>probNakagami := proc(g,g1,g2,m)
> (GAMMA(m,m*g1/g)-GAMMA(m,m*g2/g))/GAMMA(m);
>end proc;
```

```
Prosedyre for konvertere vekk dB
```

```
>remove_dB := proc(x_dB)
> 10 ^ (x_dB/10);
>end;
```

```
Inndataene:
```

```
>SE := [1,2,3,4,5,6,7,8,9,10,11]:
>gMLD_dB := [3.64,8.55,11.76,14.99,18.37,21.36,24.60,27.53,30.7,33.6,36.8,1000]:
```

```
Konverter vekk dB-en...
```

```
>for i from 1 to 12 do
> gMLD[i]:=remove_dB(gMLD_dB[i]);
>end do;
```

```
Prosedyre som beregner ASE:
```

```
>ASE := proc(g_dB,m)
> local g,s,i;
> g := remove_dB(g_dB);
> s := 0;
> for i from 1 to 11 do
> s:= s + SE[i] * probNakagami(g,gMLD[i],gMLD[i+1],m);;
> end do;
> s;
>end proc;
```

```
Plottet skrives til filen "ASE_C_stjerne_c1_L2.eps" dersom denne aktiveres.
```

```
>plotsetup(cps,plotoutput="ASE_C_stjerne_c1_L2.eps", plotoptions="portrait,
height=170pt,width=400pt,noborder,leftmargin=0,bottommargin=0");
```

```
Genererer ASE-plottet for c=1, L=2 og m={1,2,4} for MLD.
```

```
>with(plots):
>F:=plot({ASE(x,1),ASE(x,2),ASE(x,4)},x=10..30.7, color=[red,green,blue], axes
=BOXED, labels=["dB","bits/s/Hz"], labeldirections=[horizontal,vertical]):
>G1:=textplot([17,3.1,'m=1']):
>G2:=textplot([18,4.0,'m=2']):
>G4:=textplot([18.5,4.8,'m=4']):
>display(F,G1,G2,G4);
>plotsetup(default);
```

/kode/AKM/beregn\_ASE/maple/beregn\_ASE\_R0\_stjerne\_c1\_L1.mws

```
>restart;
```

Sannsynlighet for at gamma havner i n-te fading region.

```
>probNakagami := proc(g,g1,g2,m)
> (GAMMA(m,m*g1/g)-GAMMA(m,m*g2/g))/GAMMA(m);
>end proc;
```

Prosedyre for konvertere vekk dB

```
>remove_dB := proc(x_dB)
> 10 ^ (x_dB/10);
>end;
```

Inndataene:

```
>SE := [1,2,3,4,5,6,7,8,9,10,11]:
>gMLD_dB := [2.478,7.680,11.096,14.405,17.811,20.817,24.087,27.030,
30.215,33.135,36.272,1000]:
```

Konverter vekk dB-en...

```
>for i from 1 to 12 do
> gMLD[i]:=remove_dB(gMLD_dB[i]);
>end do;
```

Prosedyre som beregner ASE:

```
>ASE := proc(g_dB,m)
> local g,s,i;
> g := remove_dB(g_dB);
> s := 0;
> for i from 1 to 11 do
> s:= s + SE[i] * probNakagami(g,gMLD[i],gMLD[i+1],m);;
> end do;
> s;
>end proc;
```

Plottet skrives til filen "ASE\_R0\_stjerne\_c1\_L1.eps" dersom denne aktiveres.

```
>plotsetup(cps,plotoutput="ASE_R0_stjerne_c1_L1.eps", plotoptions="portrait,
height=170pt,width=400pt,noborder,leftmargin=0,bottommargin=0");
```

Genererer ASE-plottet for  $c=L=1$  og  $m=\{1,2,4\}$  for SD.

```
>with(plots):
>F:=plot({ASE(x,1),ASE(x,2),ASE(x,4)},x=10..37, color=[red,green,blue], axes
=BOXED, labels=["dB","bits/s/Hz"], labeldirections=[horizontal,vertical]):
>G1:=textplot([20,4.2,'m=1']):
>G2:=textplot([22,5.4,'m=2']):
>G4:=textplot([23,6.5,'m=4']):
>display(F,G1,G2,G4);
>plotsetup(default);
```

/kode/AKM/beregn\_ASE/maple/beregn\_ASE\_R0\_stjerne\_c1\_L2.mws

```
>restart;
```

Sannsynlighet for at gamma havner i n-te fading region.

```
>probNakagami := proc(g,g1,g2,m)
> (GAMMA(m,m*g1/g)-GAMMA(m,m*g2/g))/GAMMA(m);
>end proc;
```

Prosedyre for konvertere vekk dB

```
>remove_dB := proc(x_dB)
> 10 ^ (x_dB/10);
>end;
```

Inndataene:

```
>SE := [1,2,3,4,5,6,7,8,9,10,11]:
>gSD_dB := [5.205,10.083,13.155,16.361,19.741,22.714,25.934,28.856,
32.066,34.909,38.111,1000]:
```

Konverter vekk dB-en...

```
>for i from 1 to 12 do
> gSD[i]:=remove_dB(gSD_dB[i]);
>end do;
```

Prosedyre som beregner ASE:

```
>ASE := proc(g_dB,m)
> local g,s,i;
> g := remove_dB(g_dB);
> s := 0;
> for i from 1 to 11 do
> s:= s + SE[i] * probNakagami(g,gSD[i],gSD[i+1],m);
> end do;
> s;
>end proc;
```

Plottet skrives til filen "ASE\_R0\_stjerne\_c1\_L2.eps" dersom denne aktiveres.

```
>plotsetup(cps,plotoutput="ASE_R0_stjerne_c1_L2.eps", plotoptions="portrait,
height=170pt,width=400pt,noborder,leftmargin=0,bottommargin=0");
```

Genererer ASE-plottet for c=1, L=2 for SD.

```
>with(plots):
>F:=plot({ASE(x,1),ASE(x,2),ASE(x,4)},x=10..39, color=[red,green,blue], axes
=BOXED, labels=["dB","bits/s/Hz"], labeldirections=[horizontal,vertical]):
>G1:=textplot([20,3.5,'m=1']):
>G2:=textplot([22,4.9,'m=2']):
>G4:=textplot([23,6.0,'m=4']):
>display(F,G1,G2,G4);
>plotsetup(default);
```

/kode/AKM/beregn\_ASE/maple/beregn\_diff\_ASE\_c1\_L1.mws

>restart;

Sannsynlighet for at gamma havner i n-te fading region.

```
>probNakagami := proc(g,g1,g2,m)
> (GAMMA(m,m*g1/g)-GAMMA(m,m*g2/g))/GAMMA(m);
>end proc;
```

Prosedyre for konvertere vekk dB

```
>remove_dB := proc(x_dB)
> 10 ^ (x_dB/10);
>end;
```

Inndataene:

```
>SE := [1,2,3,4,5,6,7,8,9,10,11]:
>gMLD_dB_C0star := [0.83,6.14,9.77,13.12,16.55,19.59,22.86,25.79,
                    29.0,31.9,35.1,1000]:
>gSD_dB_R0star := [2.478,7.680,11.096,14.405,17.811,20.817,24.087,
                  27.030,30.215,33.135,36.272,1000]:
```

Konverter vekk dB-en...

```
>for i from 1 to 12 do
> gMLD_C0star[i]:=remove_dB(gMLD_dB_C0star[i]);
> gSD_R0star[i]:=remove_dB(gSD_dB_R0star[i]);
>end do;
```

Prosedyre som beregner differansen i ASE mellom C\* og R0\*:

```
>diff_ASE := proc(g_dB,m)
> local g,ASE_C0star,ASE_R0star,diffASE,i;
> g := remove_dB(g_dB);
> ASE_C0star := 0;
> ASE_R0star := 0;
> for i from 1 to 11 do
> ASE_C0star:= ASE_C0star
>             + SE[i] * probNakagami(g,gMLD_C0star[i],gMLD_C0star[i+1],m);
> ASE_R0star:= ASE_R0star
>             + SE[i] * probNakagami(g,gSD_R0star[i],gSD_R0star[i+1],m);
> end do;
>
>diffASE:= ASE_C0star - ASE_R0star;
>end proc;
```

Plottet skrives til filen "diff\_ASE\_c1\_L1.eps" dersom denne aktiveres.

```
>plotsetup(cps,plotoutput="diff_ASE_c1_L1.eps", plotoptions="portrait,
            height=170pt,width=400pt,noborder,leftmargin=0,bottommargin=0");
```

Genererer ASE-differanseplottet for c=1, L=2.

```
>with(plots):
```



```

>F:=plot({diff_ASE(x,1),diff_ASE(x,2),diff_ASE(x,4)},x=10..29,
         color=[red,green,blue], axes=BOXED, labels=["dB","bits/s/Hz"],
         labeldirections=[horizontal,vertical]):
>G1:=textplot([17.5,0.375,'m=1']):
>G2:=textplot([18.5,0.3875,'m=2']):
>G4:=textplot([18.5,0.401,'m=4']):
>display(F,G1,G2,G4);
>plotsetup(default);

```

/kode/AKM/beregn\_ASE/maple/beregn\_diff\_ASE\_c1\_L2.mws

```
>restart;
```

Sannsynlighet for at gamma havner i n-te fading region.

```

>probNakagami := proc(g,g1,g2,m)
> (GAMMA(m,m*g1/g)-GAMMA(m,m*g2/g))/GAMMA(m);
>end proc:

```

Prosedyre for konvertere vekk dB

```

>remove_dB := proc(x_dB)
> 10 ^ (x_dB/10);
>end:

```

Inndataene:

```

>SE := [1,2,3,4,5,6,7,8,9,10,11]:
>gMLD_dB_C0star := [3.64,8.55,11.76,14.99,18.37,21.36,24.60,27.53,
                   30.7,33.6,36.8,1000]:
>gSD_dB_R0star := [5.205,10.083,13.155,16.361,19.741,22.714,25.934,
                  28.856,32.066,34.909,38.111,1000]:

```

Konverter vekk dB-en...

```

>for i from 1 to 12 do
>  gMLD_C0star[i]:=remove_dB(gMLD_dB_C0star[i]);
>  gSD_R0star[i]:=remove_dB(gSD_dB_R0star[i]);
>end do:

```

Prosedyre som beregner differansen i ASE mellom C\* og R0\*:

```

>diff_ASE := proc(g_dB,m)
> local g,ASE_C0star,ASE_R0star,diffASE,i;
> g := remove_dB(g_dB);
> ASE_C0star := 0;
> ASE_R0star := 0;
> for i from 1 to 11 do
>   ASE_C0star:= ASE_C0star
>     + SE[i] * probNakagami(g,gMLD_C0star[i],gMLD_C0star[i+1],m);
>   ASE_R0star:= ASE_R0star
>     + SE[i] * probNakagami(g,gSD_R0star[i],gSD_R0star[i+1],m);
> end do;
>diffASE:= ASE_C0star - ASE_R0star;
>end proc:

```

```
Plottet skrives til filen "diff_ASE_c1_L2.eps" dersom denne aktiveres.  
>plotsetup(cps,plotoutput="diff_ASE_c1_L2.eps", plotoptions="portrait,  
           height=170pt,width=400pt,noborder,leftmargin=0,bottommargin=0");
```

Genererer ASE-differanseplottet for  $c=1$ ,  $L=2$ .

```
>with(plots):  
>F:=plot({diff_ASE(x,1),diff_ASE(x,2),diff_ASE(x,4)},x=10..30.7,  
         color=[red,green,blue], axes=BOXED, labels=["dB","bits/s/Hz"],  
         labeldirections=[horizontal,vertical]):  
>G1:=textplot([15,0.385,'m=1']):  
>G2:=textplot([13.3,0.412,'m=2']):  
>G4:=textplot([12.5,0.432,'m=4']):  
>display(F,G1,G2,G4);  
>plotsetup(default);
```

## A.4 Programkode kapittel 5

Følgende filer er inkludert:

```
/kode
  '-- OFDM
    |-- beregn_totalASE_OFDM-AKM-QAM
    |   '-- maple
    |       |-- beregn_totalASE_C_stjerne.mws
    |       |-- beregn_totalASE_RO_stjerne.mws
    |       '-- beregn_totalASE_differanse.mws
    |-- beregn_totalASE_Hiperlan2_modell
    |   |-- beregn_ASE_punkter
    |   |   |-- MLDgenASE.m
    |   |   |-- SDgenASE.m
    |   |   |-- bestemFadingRegion.m
    |   |   |-- randGamma.m
    |   |   '-- remove_dB.m
    |   '-- plott_ASE_punkter
    |       |-- H2_ASE_ingen_subkanalkorrelasjon.mws
    |       '-- plott_H2_ASE.mws
    '-- sammenligning_totalASE_Hiperlan2_OFDM-AKM-QAM
        |-- MLD_sammenligning.mws
        '-- SD_sammenligning.mws
```

```
/kode/OFDM/beregn_totalASE_OFDM-AKM-QAM/
maple/beregn_totalASE_C_stjerne.mws
```

```
>restart;
```

Prosedyre som beregner total ASE for OFDM:

```
>totalASE := proc(g_dB,H)
> local g,s,i;
> g := remove_dB(g_dB);
> s := 0;
> for i from 1 to 11 do
>   s:= s + SE[i] * probNakagami(g,gMLD[i],gMLD[i+1],m,H);
> end do;
> s := s*Q/(Ttotal*Wtotal);
>end proc;
```

Skriver neste plott til filen totalASE\_C\_stjerne\_c1\_L1\_m1.eps  
hvis denne linjen aktiveres:

```
>plotsetup(cps,plotoutput="totalASE_C_stjerne_c1_L1_m1.eps",
           plotoptions="portrait,height=170pt,width=400pt,
           noborder,leftmargin=0,bottommargin=0");
```

Plotter total ASE for  $c=L=1$ ,  $H=\{1,2,4\}$  og  $m=1$

```
>m:=1:
```

```

>with(plots):
>F:=plot({totalASE(x,1),totalASE(x,2),totalASE(x,4)},x=10..29,
        color=[red,green,blue], axes=BOXED, labels=["dB","bits/s/Hz"],
        labeldirections=[horizontal,vertical]):
>G1:=textplot([17.1,3.6,'H=1']):
>G2:=textplot([18,4.9,'H=2']):
>G4:=textplot([18.5,5.9,'H=4']):
>display(F,G1,G2,G4);
>plotsetup(default);

```

Skriver neste plott til filen totalASE\_C\_stjerne\_c1\_L1\_m2.eps  
hvis denne linjen aktiveres:

```

>plotsetup(cps,plotoutput="totalASE_C_stjerne_c1_L1_m2.eps",
          plotoptions="portrait,height=170pt,width=400pt,
          noborder,leftmargin=0,bottommargin=0");

```

Plotter total ASE for  $c=L=1$ ,  $H=\{1,2,4\}$  og  $m=2$

```

>gMLD[12]:=remove_dB(438):
>m:=2:
>with(plots):
>F:=plot({totalASE(x,1),totalASE(x,2),totalASE(x,4)},x=10..29,
        color=[red,green,blue], axes=BOXED, labels=["dB","bits/s/Hz"],
        labeldirections=[horizontal,vertical]):
>G1:=textplot([17.1,3.9,'H=1']):
>G2:=textplot([18,5,'H=2']):
>G4:=textplot([18.5,6,'H=4']):
>display(F,G1,G2,G4);
>plotsetup(default);

```

Skriver neste plott til filen totalASE\_C\_stjerne\_c1\_L1\_m3.eps  
hvis denne linjen aktiveres:

```

>plotsetup(cps,plotoutput="totalASE_C_stjerne_c1_L1_m3.eps",
          plotoptions="portrait,height=170pt,width=400pt,noborder,
          leftmargin=0,bottommargin=0");

```

Plotter total ASE for  $c=L=1$ ,  $H=\{1,2,4\}$  og  $m=3$

```

>gMLD[12]:=remove_dB(279):
>m:=3:
>with(plots):
>F:=plot({totalASE(x,1),totalASE(x,2),totalASE(x,4)},x=10..29,
        color=[red,green,blue], axes=BOXED, labels=["dB","bits/s/Hz"],
        labeldirections=[horizontal,vertical]):
>G1:=textplot([17.1,3.9,'H=1']):
>G2:=textplot([18,5,'H=2']):
>G4:=textplot([18.5,6,'H=4']):
>display(F,G1,G2,G4);
>plotsetup(default);

```

```
/kode/OFDM/beregn_totalASE_OFDM-AKM-QAM/  
maple/beregn_totalASE_RO_stjerne.mws
```

```
>restart;
```

```
Sannsynlighet for at gamma havner i n-te fading region.
```

```
>probNakagami := proc(g,g1,g2,m,H)  
> (GAMMA(H*m,m*g1/g)-GAMMA(H*m,m*g2/g))/GAMMA(H*m);  
>end proc:
```

```
Prosedyre for konvertere vekk dB
```

```
>remove_dB := proc(x_dB)  
> 10 ^ (x_dB/10);  
>end:
```

```
Inndataene:
```

```
>SE := [1,2,3,4,5,6,7,8,9,10,11]:  
>gSD_dB := [2.478,7.680,11.096,14.405,17.811,20.817,  
24.087,27.030,30.215,33.135,36.272,1000]:  
>Q:=52:  
>Wtotal:=16.26:  
>Ttotal:=4:
```

```
Konverter vekk dB-en...
```

```
>for i from 1 to 12 do  
> gSD[i]:=remove_dB(gSD_dB[i]);  
>end do:
```

```
Prosedyre som beregner total ASE for OFDM:
```

```
>totalASE := proc(g_dB,H)  
> local g,s,i;  
> g := remove_dB(g_dB);  
> s := 0;  
> for i from 1 to 11 do  
> s:= s + SE[i] * probNakagami(g,gSD[i],gSD[i+1],m,H);  
> end do;  
> s := s*Q/(Ttotal*Wtotal);  
>end proc:
```

```
Skriver neste plott til filen totalASE_RO_stjerne_c1_L1_m1.eps
```

```
hvis denne linjen aktiveres:
```

```
>plotsetup(cps,plotoutput="totalASE_RO_stjerne_c1_L1_m1.eps", noborder,  
plotoptions="portrait,height=170pt, width=400pt,  
leftmargin=0,bottommargin=0");
```

```
Plotter total ASE for c=L=1, H={1,2,4} og m=1
```

```
>m:=1:  
>with(plots):  
>F:=plot({totalASE(x,1),totalASE(x,2),totalASE(x,4)},x=10..30,
```

```

        color=[red,green,blue], axes=BOXED, labels=["dB","bits/s/Hz"],
        labeldirections=[horizontal,vertical]):
>G1:=textplot([17,3.3,'H=1']):
>G2:=textplot([18,4.6,'H=2']):
>G4:=textplot([18.5,5.6,'H=4']):
>display(F,G1,G2,G4);
>plotsetup(default);

```

Skriver neste plott til filen totalASE\_R0\_stjerne\_c1\_L1\_m2.eps  
hvis denne linjen aktiveres:

```

>plotsetup(cps,plotoutput="totalASE_R0_stjerne_c1_L1_m2.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");

```

Plotter total ASE for  $c=L=1$ ,  $H=\{1,2,4\}$  og  $m=2$

```

>gSD[12]:=remove_dB(438):
>m:=2:
>with(plots):
>F:=plot({totalASE(x,1),totalASE(x,2),totalASE(x,4)},x=10..30,
         color=[red,green,blue], axes=BOXED, labels=["dB","bits/s/Hz"],
         labeldirections=[horizontal,vertical]):
>G1:=textplot([17,3.5,'H=1']):
>G2:=textplot([18,4.7,'H=2']):
>G4:=textplot([18.5,5.6,'H=4']):
>display(F,G1,G2,G4);
>plotsetup(default);

```

Skriver neste plott til filen totalASE\_R0\_stjerne\_c1\_L1\_m3.eps  
hvis denne linjen aktiveres:

```

>plotsetup(cps,plotoutput="totalASE_R0_stjerne_c1_L1_m3.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");

```

Plotter total ASE for  $c=L=1$ ,  $H=\{1,2,4\}$  og  $m=3$

```

>gSD[12]:=remove_dB(279):
>m:=3:
>with(plots):
>F:=plot({totalASE(x,1),totalASE(x,2),totalASE(x,4)},x=10..30,
         color=[red,green,blue], axes=BOXED, labels=["dB","bits/s/Hz"],
         labeldirections=[horizontal,vertical]):
>G1:=textplot([17,3.6,'H=1']):
>G2:=textplot([18,4.7,'H=2']):
>G4:=textplot([18.5,5.7,'H=4']):
>display(F,G1,G2,G4);
>plotsetup(default);

```

```
/kode/OFDM/beregn_totalASE_OFDM-AKM-QAM/  
maple/beregn_totalASE_differanse.mws
```

```
>restart;
```

```
Sannsynlighet for at gamma havner i n-te fading region.
```

```
>probNakagami := proc(g,g1,g2,m,H)  
> (GAMMA(H*m,m*g1/g)-GAMMA(H*m,m*g2/g))/GAMMA(H*m);  
>end proc;
```

```
Prosedyre for konvertere vekk dB
```

```
>remove_dB := proc(x_dB)  
> 10 ^ (x_dB/10);  
>end;
```

```
Inndataene:
```

```
>SE := [1,2,3,4,5,6,7,8,9,10,11]:  
>gMLD_dB := [0.83,6.14,9.77,13.12,16.55,19.59,  
22.86,25.79,29.0,31.9,35.1,1000]:  
>gSD_dB := [2.478,7.680,11.096,14.405,17.811,20.817,  
24.087,27.030,30.215,33.135,36.272,1000]:  
>Q:=52:  
>Wtotal:=16.26:  
>Ttotal:=4:
```

```
Konverter vekk dB-en...
```

```
>for i from 1 to 12 do  
> gMLD[i]:=remove_dB(gMLD_dB[i]);  
> gSD[i]:=remove_dB(gSD_dB[i]);  
>end do;
```

```
Prosedyre som beregner total ASE for OFDM:
```

```
>diff_totalASE := proc(g_dB,H)  
> local g,i,totASE_Cstar, totASE_ROstar;  
> g:=remove_dB(g_dB);  
> totASE_Cstar := 0;  
> totASE_ROstar := 0;  
>for i from 1 to 11 do  
> totASE_Cstar:= totASE_Cstar  
+ SE[i] * probNakagami(g,gMLD[i],gMLD[i+1],m,H);  
> totASE_ROstar:= totASE_ROstar  
+ SE[i] * probNakagami(g,gSD[i],gSD[i+1],m,H);  
>end do;  
> totASE_Cstar := totASE_Cstar*Q/(Ttotal*Wtotal);  
> totASE_ROstar := totASE_ROstar*Q/(Ttotal*Wtotal);  
>diffTotASE:=totASE_Cstar - totASE_ROstar  
>end proc;
```

```
Skriver neste plott til filen diff_totalASE-Cs_R0s_c1_L1_m1.eps
```

hvis denne linjen aktiveres:

```
>plotsetup(cps,plotoutput="diff_totalASE_Cs_R0s_c1_L1_m1.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");
```

Plotter total ASE for  $c=L=1$ ,  $H=\{1,2,4\}$  og  $m=1$

```
>m:=1:
>with(plots):
>F:=plot({diff_totalASE(x,1),diff_totalASE(x,2),diff_totalASE(x,4)},x=10..29,
         color=[red,green,blue], axes=BOXED, labels=["dB","bits/s/Hz"],
         labeldirections=[horizontal,vertical]):
>G1:=textplot([13.7,1,0.285,'H=1']):
>G2:=textplot([13,0.305,'H=2']):
>G4:=textplot([12.6,0.322,'H=4']):
>display(F,G1,G2,G4);
>plotsetup(default);
```

Skriver neste plott til filen diff\_totalASE\_Cs\_R0s\_c1\_L1\_m2.eps

hvis denne linjen aktiveres:

```
>plotsetup(cps,plotoutput="diff_totalASE_Cs_R0s_c1_L1_m2.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");
```

Plotter total ASE for  $c=L=1$ ,  $H=\{1,2,4\}$  og  $m=2$

```
>gSD[12]:=remove_dB(1000):
>gMLD[12]:=remove_dB(1000):
>m:=2:
>with(plots):
>F:=plot({diff_totalASE(x,1),diff_totalASE(x,2),diff_totalASE(x,4)},x=10..29,
         color=[red,green,blue], axes=BOXED, labels=["dB","bits/s/Hz"],
         labeldirections=[horizontal,vertical]):
>G1:=textplot([14,0.303,'H=1']):
>G2:=textplot([15.5,0.32,'H=2']):
>G4:=textplot([13.6,0.33,'H=4']):
>display(F,G1,G2,G4);
>plotsetup(default);
```

Skriver neste plott til filen diff\_totalASE\_Cs\_R0s\_c1\_L1\_m3.eps

hvis denne linjen aktiveres:

```
>plotsetup(cps,plotoutput="diff_totalASE_Cs_R0s_c1_L1_m3.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");
```

Plotter total ASE for  $c=L=1$ ,  $H=\{1,2,4\}$  og  $m=3$

```
>gSD[12]:=remove_dB(279):
>gMLD[12]:=remove_dB(279):
>m:=3:
>with(plots):
>F:=plot({diff_totalASE(x,1),diff_totalASE(x,2),diff_totalASE(x,4)},x=10..29,
```



```

        color=[red,green,blue], axes=BOXED, labels=["dB","bits/s/Hz"],
        labeldirections=[horizontal,vertical]):
>G1:=textplot([15.7,1,0.31,'H=1']):
>G2:=textplot([12,0.295,'H=2']):
>G4:=textplot([12.4,0.333,'H=4']):
>display(F,G1,G2,G4);
>plotsetup(default);

/kode/OFDM/beregn_totalASE_Hiperlan2_modell/
beregn_ASE_punkter/MLDgenASE.m

function f = MLDgenASE(K,m,H,b,min_dB,step_dB,max_dB)
%   MLDgenASE(K,m,H,b,min_dB,step_dB,max_dB)
%
%Generer ASE for fadingregiongrensene ved Maximum Likelihood Dekoding (c=L=1).
%Innparametre:
% K = # ganger det skal snittes over hver punkt
% m = nakagamiparameteren
% H = # antenner
% b = blokkstørrelse for antall subkanaler som er korrelerte
% (min_dB,step_dB,max_dB) = grenser og punkt for hvor det skal genereres ASE.

path(path, '../..../AWGN_kanalmodell/felles/matlab/');

%Fadingregiongrenser, c=L=1:
gMLD_dB = [0.83,6.14,9.77,13.12,16.55,19.59,22.86,25.79,29.0,31.9,35.1,1000];
gMLD=remove_dB(gMLD_dB);

Q=52;
W_total=16.26;
T_total=4;

avgSNR_dB=[min_dB:step_dB:max_dB]';
ASE=zeros(size(avgSNR_dB));

for i=1:length(avgSNR_dB);
    avgSNR = remove_dB(avgSNR_dB(i));
    % Q*K matrise med gammafordelte tall følger under
    randg = randGamma(m,H,avgSNR,Q/b,K,sum(100*clock));
    % SE=0;
    % for k=1:1:K
    %     minSNR = min(randg(:,k)); %minste SNR blant de Q subkanalene
    %     n=bestemFadingRegion(minSNR,gMLD); %Alle subkanaler skal bruke
    %     %kode for denne fadingregionen
    %     SE(k)=Q * n/(T_total*W_total); %Spektral effektivitet for alle Q
    %                                     subkanalene.
    %
    %         %Kode n har rate R_n=(log2(M_n)-c/L)/T_total.
    %         % ==> R_n=n/T_total for c=L=1;

```

```

% end;
if Q/b == 1
    %Vektorisering av for-løkken over.
    SE=Q*bestemFadingRegion(randg,gMLD)/(T_total*W_total);
else
    %Vektorisering av for-løkken over.
    SE=Q*bestemFadingRegion(min(randg),gMLD)/(T_total*W_total);
end;
ASE(i)=sum(SE)/K;
disp(['ASE(avgsNR=' num2str(avgsNR_dB(i)) ') = ' num2str(ASE(i))])
end;

skrivPunkterTilFil(avgsNR_dB,ASE,['resultater/ASE_MLD_K=' num2str(K) '_m='
    num2str(m) '_H=' num2str(H) '_b=' num2str(b) '_avgsNR='
    num2str(min_dB) '- ' num2str(max_dB)]);

/kode/OFDM/beregn_totalASE_Hiperlan2_modell/
beregn_ASE_punkter/SDgenASE.m

function f = SDgenASE(K,m,H,b,min_dB,step_dB,max_dB)
% SDgenASE(K,m,H,b,min_dB,step_dB,max_dB)
%
%Generer ASE for fadingregiongrensene ved Sekvensiell Dekoding (c=L=1).
%Innparametre:
% K = # ganger det skal snittes over hver punkt
% m = nakagamiparameteren
% H = # antenner
% b = blokkstørrelse for antall subkanaler som er korrelerte
% (min_dB,step_dB,max_dB) = grenser og punkt for hvor det skal genereres ASE.

path(path, '../.../AWGN_kanalmodell/felles/matlab/');

%Fadingregiongrenser, c=L=1:
gSD_dB = [2.478,7.680,11.096,14.405,17.811,20.817,
    24.087,27.030,30.215,33.135,36.272,1000];

gSD=remove_dB(gSD_dB);

Q=52;
W_total=16.26;
T_total=4;

avgsNR_dB=[min_dB:step_dB:max_dB]';
ASE=zeros(size(avgsNR_dB));

for i=1:length(avgsNR_dB);
    avgsNR = remove_dB(avgsNR_dB(i));
    % Q*K matrise med gammafordelte tall:
    randg = randGamma(m,H,avgsNR,Q/b,K,sum(100*clock));

```

```

% SE=0;
% for k=1:1:K
%     minSNR = min(randg(:,k));   minste SNR blant de Q subkanalene
%     n=bestemFadingRegion(minSNR,gSD); %Alle subkanaler skal bruke
%                                     %kode for denne fadingregionen.
%     SE(k)=Q * n/(T_total*W_total); %Spektral effektivitet for alle
%                                     %Q subkanalene.
%     %Kode n har rate R_n=(log2(M_n)-c/L)/T_total.
%                                     %=> R_n=n/T_total for c=L=1;
% end;
if Q/b == 1
    %Vektorisering av for-løkken over.
    SE=Q*bestemFadingRegion(randg,gSD)/(T_total*W_total);
else
    %Vektorisering av for-løkken over.
    SE=Q*bestemFadingRegion(min(randg),gSD)/(T_total*W_total);
end;
ASE(i)=sum(SE)/K;
disp(['ASE(avgsNR=' num2str(avgsNR_dB(i)) ') = ' num2str(ASE(i))])
end;

```

```

skrivPunkterTilFil(avgsNR_dB,ASE,['resultater/ASE_SD_K=' num2str(K) '_m='
    num2str(m) '_H=' num2str(H) '_b=' num2str(b) '_avgsNR='
    num2str(min_dB) '- ' num2str(max_dB)]);

```

### /kode/OFDM/beregn\_totalASE\_Hiperlan2\_modell/ beregn\_ASE\_punkter/bestemFadingRegion.m

```

function i = bestemFadingRegion(SNR,regionGrenser)
i=zeros(size(SNR));
for j=1:1:length(i)
    while SNR(j)>=regionGrenser(i(j)+1)
        i(j)=i(j)+1;
    end;
end;

```

### /kode/OFDM/beregn\_totalASE\_Hiperlan2\_modell/ beregn\_ASE\_punkter/randGamma.m

```

function X = randGamma(m,H,avgsNR,Q,K,randstate)
%     randg = randGamma(m,avgsNR,Q,K,randstate)
%
% Returner en matrise med Q*K tilfeldige Gammafordelte tall.
% NB! Gjelder for heltallig nakagamiparameter m og H.

rand('state',randstate);

%Ekspontielt fordelte tilfeldige tall mellom [0,inf (ca 18 i praksis?):
Z=-log(rand(Q,K,m*H));

```

```
X=zeros(Q,K);
```

```
for i=1:m*H  
    X=X+Z(:, :, i);  
end;
```

```
X=X*avgSNR/m;
```

```
/kode/OFDM/beregn_totalASE_Hiperlan2_modell/  
beregn_ASE_punkter/remove_dB.m
```

```
function x = remove_dB(x_dB)  
    x= 10.^(x_dB/10);
```

```
/kode/OFDM/beregn_totalASE_Hiperlan2_modell/  
plott_ASE_punkter/H2_ASE_ingen_subkanalkorrelasjon.mws
```

```
>restart;  
Prosedyre som sorterer to punkter etter x-koordinatene.  
>sorter:=proc(x,y)  
>    if x[1] < y[1] then true  
>    elif x[1] > y[1] then false  
>    else evalb(x[2] < y[2])  
>    fi  
>end:
```

```
Leser inn og sorterer simuleringsdataene fra HiperLAN/2-modellen:  
>data_MLD_m1_H1_b1:=sort(readdata(ASE_MLD_K100000_m1_H1_b1,2),sorter):  
>data_MLD_m2_H1_b1:=sort(readdata(ASE_MLD_K100000_m2_H1_b1,2),sorter):  
>data_SD_m1_H1_b1:=sort(readdata(ASE_SD_K100000_m1_H1_b1,2),sorter):  
>data_SD_m2_H1_b1:=sort(readdata(ASE_SD_K100000_m2_H1_b1,2),sorter):
```

```
Skriver neste plott til filen H2_ASE_b1.eps hvis denne linjen aktiveres:  
>plotsetup(cps,plotoutput="H2_ASE_b1.eps", plotoptions="portrait,  
    height=170pt,width=400pt,noborder,leftmargin=0,bottommargin=0");
```

```
Plotter ASE for m={1,2} H=1 og b=1 for MLD og SD:  
>with(plots):  
>F:=plot({data_MLD_m1_H1_b1,data_MLD_m2_H1_b1,data_SD_m1_H1_b1,  
    data_SD_m2_H1_b1},x=10..30, color=[green,red,blue,black],  
    axes=BOXED, labels=["dB","bits/s/Hz"],  
    labeldirections=[horizontal,vertical]):  
>b1:=textplot([25,0.9,'SD_m=1']):  
>b2:=textplot([25,1.5,'MLD_m=1']):  
>b4:=textplot([25,2.5,'SD_m=2']):  
>b13:=textplot([25,3,'MLD_m=2']):  
>display(F,b1,b2,b4,b13);  
>plotsetup(default):
```

/kode/OFDM/beregn\_totalASE\_Hiperlan2\_modell/  
plott\_ASE\_punkter/plott\_H2\_ASE.mws

```
>restart;
```

Prosedyre som sorterer to punkter etter x-koordinatene.

```
>sorter:=proc(x,y)
>  if x[1] < y[1] then true
>  elif x[1] > y[1] then false
>  else evalb(x[2] < y[2])
>  fi
>end:
```

Leser inn og sorterer simuleringsdataene fra HiperLAN/2-modellen:

```
>data_MLD_m1_H1_b1:=sort(readdata(ASE_MLD_K100000_m1_H1_b1,2),sorter):
>data_MLD_m1_H1_b2:=sort(readdata(ASE_MLD_K100000_m1_H1_b2,2),sorter):
>data_MLD_m1_H1_b4:=sort(readdata(ASE_MLD_K100000_m1_H1_b4,2),sorter):
>data_MLD_m1_H1_b13:=sort(readdata(ASE_MLD_K100000_m1_H1_b13,2),sorter):
>data_MLD_m1_H1_b26:=sort(readdata(ASE_MLD_K100000_m1_H1_b26,2),sorter):
>data_MLD_m1_H1_b52:=sort(readdata(ASE_MLD_K100000_m1_H1_b52,2),sorter):
>data_MLD_m2_H1_b1:=sort(readdata(ASE_MLD_K100000_m2_H1_b1,2),sorter):
>data_MLD_m2_H1_b2:=sort(readdata(ASE_MLD_K100000_m2_H1_b2,2),sorter):
>data_MLD_m2_H1_b4:=sort(readdata(ASE_MLD_K100000_m2_H1_b4,2),sorter):
>data_MLD_m2_H1_b13:=sort(readdata(ASE_MLD_K100000_m2_H1_b13,2),sorter):
>data_MLD_m2_H1_b26:=sort(readdata(ASE_MLD_K100000_m2_H1_b26,2),sorter):
>data_MLD_m2_H1_b52:=sort(readdata(ASE_MLD_K100000_m2_H1_b52,2),sorter):
>data_MLD_m1_H2_b1:=sort(readdata(ASE_MLD_K100000_m1_H2_b1,2),sorter):
>data_MLD_m1_H2_b2:=sort(readdata(ASE_MLD_K100000_m1_H2_b2,2),sorter):
>data_MLD_m1_H2_b4:=sort(readdata(ASE_MLD_K100000_m1_H2_b4,2),sorter):
>data_MLD_m1_H2_b13:=sort(readdata(ASE_MLD_K100000_m1_H2_b13,2),sorter):
>data_MLD_m1_H2_b26:=sort(readdata(ASE_MLD_K100000_m1_H2_b26,2),sorter):
>data_MLD_m1_H2_b52:=sort(readdata(ASE_MLD_K100000_m1_H2_b52,2),sorter):
>data_MLD_m2_H2_b1:=sort(readdata(ASE_MLD_K100000_m2_H2_b1,2),sorter):
>data_MLD_m2_H2_b2:=sort(readdata(ASE_MLD_K100000_m2_H2_b2,2),sorter):
>data_MLD_m2_H2_b4:=sort(readdata(ASE_MLD_K100000_m2_H2_b4,2),sorter):
>data_MLD_m2_H2_b13:=sort(readdata(ASE_MLD_K100000_m2_H2_b13,2),sorter):
>data_MLD_m2_H2_b26:=sort(readdata(ASE_MLD_K100000_m2_H2_b26,2),sorter):
>data_MLD_m2_H2_b52:=sort(readdata(ASE_MLD_K100000_m2_H2_b52,2),sorter):
>data_MLD_m1_H4_b1:=sort(readdata(ASE_MLD_K100000_m1_H4_b1,2),sorter):
>data_MLD_m1_H4_b2:=sort(readdata(ASE_MLD_K100000_m1_H4_b2,2),sorter):
>data_MLD_m1_H4_b4:=sort(readdata(ASE_MLD_K100000_m1_H4_b4,2),sorter):
>data_MLD_m1_H4_b13:=sort(readdata(ASE_MLD_K100000_m1_H4_b13,2),sorter):
>data_MLD_m1_H4_b26:=sort(readdata(ASE_MLD_K100000_m1_H4_b26,2),sorter):
>data_MLD_m1_H4_b52:=sort(readdata(ASE_MLD_K100000_m1_H4_b52,2),sorter):
>data_MLD_m2_H4_b1:=sort(readdata(ASE_MLD_K100000_m2_H4_b1,2),sorter):
>data_MLD_m2_H4_b2:=sort(readdata(ASE_MLD_K100000_m2_H4_b2,2),sorter):
>data_MLD_m2_H4_b4:=sort(readdata(ASE_MLD_K100000_m2_H4_b4,2),sorter):
>data_MLD_m2_H4_b13:=sort(readdata(ASE_MLD_K100000_m2_H4_b13,2),sorter):
>data_MLD_m2_H4_b26:=sort(readdata(ASE_MLD_K100000_m2_H4_b26,2),sorter):
```

```

>data_MLD_m2_H4_b52:=sort(readdata(ASE_MLD_K100000_m2_H4_b52,2),sorter):

>data_SD_m1_H1_b1:=sort(readdata(ASE_SD_K100000_m1_H1_b1,2),sorter):
>data_SD_m1_H1_b2:=sort(readdata(ASE_SD_K100000_m1_H1_b2,2),sorter):
>data_SD_m1_H1_b4:=sort(readdata(ASE_SD_K100000_m1_H1_b4,2),sorter):
>data_SD_m1_H1_b13:=sort(readdata(ASE_SD_K100000_m1_H1_b13,2),sorter):
>data_SD_m1_H1_b26:=sort(readdata(ASE_SD_K100000_m1_H1_b26,2),sorter):
>data_SD_m1_H1_b52:=sort(readdata(ASE_SD_K100000_m1_H1_b52,2),sorter):
>data_SD_m2_H1_b1:=sort(readdata(ASE_SD_K100000_m2_H1_b1,2),sorter):
>data_SD_m2_H1_b2:=sort(readdata(ASE_SD_K100000_m2_H1_b2,2),sorter):
>data_SD_m2_H1_b4:=sort(readdata(ASE_SD_K100000_m2_H1_b4,2),sorter):
>data_SD_m2_H1_b13:=sort(readdata(ASE_SD_K100000_m2_H1_b13,2),sorter):
>data_SD_m2_H1_b26:=sort(readdata(ASE_SD_K100000_m2_H1_b26,2),sorter):
>data_SD_m2_H1_b52:=sort(readdata(ASE_SD_K100000_m2_H1_b52,2),sorter):
>data_SD_m1_H2_b1:=sort(readdata(ASE_SD_K100000_m1_H2_b1,2),sorter):
>data_SD_m1_H2_b2:=sort(readdata(ASE_SD_K100000_m1_H2_b2,2),sorter):
>data_SD_m1_H2_b4:=sort(readdata(ASE_SD_K100000_m1_H2_b4,2),sorter):
>data_SD_m1_H2_b13:=sort(readdata(ASE_SD_K100000_m1_H2_b13,2),sorter):
>data_SD_m1_H2_b26:=sort(readdata(ASE_SD_K100000_m1_H2_b26,2),sorter):
>data_SD_m1_H2_b52:=sort(readdata(ASE_SD_K100000_m1_H2_b52,2),sorter):
>data_SD_m2_H2_b1:=sort(readdata(ASE_SD_K100000_m2_H2_b1,2),sorter):
>data_SD_m2_H2_b2:=sort(readdata(ASE_SD_K100000_m2_H2_b2,2),sorter):
>data_SD_m2_H2_b4:=sort(readdata(ASE_SD_K100000_m2_H2_b4,2),sorter):
>data_SD_m2_H2_b13:=sort(readdata(ASE_SD_K100000_m2_H2_b13,2),sorter):
>data_SD_m2_H2_b26:=sort(readdata(ASE_SD_K100000_m2_H2_b26,2),sorter):
>data_SD_m2_H2_b52:=sort(readdata(ASE_SD_K100000_m2_H2_b52,2),sorter):
>data_SD_m1_H4_b1:=sort(readdata(ASE_SD_K100000_m1_H4_b1,2),sorter):
>data_SD_m1_H4_b2:=sort(readdata(ASE_SD_K100000_m1_H4_b2,2),sorter):
>data_SD_m1_H4_b4:=sort(readdata(ASE_SD_K100000_m1_H4_b4,2),sorter):
>data_SD_m1_H4_b13:=sort(readdata(ASE_SD_K100000_m1_H4_b13,2),sorter):
>data_SD_m1_H4_b26:=sort(readdata(ASE_SD_K100000_m1_H4_b26,2),sorter):
>data_SD_m1_H4_b52:=sort(readdata(ASE_SD_K100000_m1_H4_b52,2),sorter):
>data_SD_m2_H4_b1:=sort(readdata(ASE_SD_K100000_m2_H4_b1,2),sorter):
>data_SD_m2_H4_b2:=sort(readdata(ASE_SD_K100000_m2_H4_b2,2),sorter):
>data_SD_m2_H4_b4:=sort(readdata(ASE_SD_K100000_m2_H4_b4,2),sorter):
>data_SD_m2_H4_b13:=sort(readdata(ASE_SD_K100000_m2_H4_b13,2),sorter):
>data_SD_m2_H4_b26:=sort(readdata(ASE_SD_K100000_m2_H4_b26,2),sorter):
>data_SD_m2_H4_b52:=sort(readdata(ASE_SD_K100000_m2_H4_b52,2),sorter):

```

Skriver neste plott til filen H2\_ASE\_C\_stjerne\_c1\_L1\_m1\_H1.eps

hvis denne linjen aktiveres:

```

>plotsetup(cps,plotoutput="H2_ASE_C_stjerne_c1_L1_m1_H1.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");

```

Plotter ASE for  $m=1$ ,  $H=1$  og  $b=\{1,2,4,13,26,52\}$  for MLD:

```

>with(plots):
>F:=plot({data_MLD_m1_H1_b1,data_MLD_m1_H1_b2,data_MLD_m1_H1_b4,
          data_MLD_m1_H1_b13},x=10..30, color=[green,red,blue,black],

```

```

        axes=BOXED, labels=["dB","bits/s/Hz"],
        labeldirections=[horizontal,vertical]):
>b1:=textplot([25,0.9,'b=1']):
>b2:=textplot([25,1.8,'b=2']):
>b4:=textplot([25,2.6,'b=4']):
>b13:=textplot([25,3.7,'b=13']):
>b26:=textplot([25,4.5,'b=26']):
>b52:=textplot([25,5.3,'b=52']):
>display(F,b1,b2,b4,b13);
>plotsetup(default):

Skriver neste plott til filen H2_ASE_C_stjerne_c1_L1_m1_H2.eps
hvis denne linjen aktiveres:
>plotsetup(cps,plotoutput="H2_ASE_C_stjerne_c1_L1_m1_H2.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");
>Plotter ASE for m=1, H=1 og b={1,2,4,13,26,52} for MLD:
>with(plots):
>F:=plot({data_MLD_m1_H2_b1,data_MLD_m1_H2_b2,data_MLD_m1_H2_b4,
          data_MLD_m1_H2_b13},x=10..30, color=[blue,black,red,green],
          axes=BOXED, labels=["dB","bits/s/Hz"],
          labeldirections=[horizontal,vertical]):
>b1:=textplot([25,3.7,'b=1']):
>b2:=textplot([25,4.2,'b=2']):
>b4:=textplot([25,4.7,'b=4']):
>b13:=textplot([25,5.3,'b=13']):
>b26:=textplot([25,5.9,'b=26']):
>b52:=textplot([25,6.5,'b=52']):
>display(F,b1,b2,b4,b13);
>plotsetup(default):

Skriver neste plott til filen H2_ASE_C_stjerne_c1_L1_m1_H4.eps
hvis denne linjen aktiveres:
>plotsetup(cps,plotoutput="H2_ASE_C_stjerne_c1_L1_m1_H4.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");

Plotter ASE for m=1, H=1 og b={1,2,4,13,26,52} for MLD:
>with(plots):
>F:=plot({data_MLD_m1_H4_b1,data_MLD_m1_H4_b2,data_MLD_m1_H4_b4,
          data_MLD_m1_H4_b13},x=10..30, color=[blue,green,red,black],
          axes=BOXED, labels=["dB","bits/s/Hz"],
          labeldirections=[horizontal,vertical]):
>b1:=textplot([25,5.2,'b=1']):
>b2:=textplot([25,5.7,'b=2']):
>b4:=textplot([25,6.3,'b=4']):
>b13:=textplot([25,6.9,'b=13']):
>b26:=textplot([25,6.9,'b=26']):
>b52:=textplot([25,7.5,'b=52']):

```

```
>display(F,b1,b2,b4,b13);
>plotsetup(default):
```

Skriver neste plott til filen H2\_ASE\_C\_stjerne\_c1\_L1\_m2\_H1.eps  
hvis denne linjen aktiveres:

```
>plotsetup(cps,plotoutput="H2_ASE_C_stjerne_c1_L1_m2_H1.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");
Plotter ASE for m=1, H=1 og b={1,2,4,13,26,52} for MLD:
>with(plots):
>F:=plot({data_MLD_m2_H1_b1,data_MLD_m2_H1_b2,data_MLD_m2_H1_b4,
          data_MLD_m2_H1_b13},x=10..30, color=[green,red,black,blue],
          axes=BOXED, labels=["dB","bits/s/Hz"],
          labeldirections=[horizontal,vertical]):
>b1:=textplot([25,3,'b=1']):
>b2:=textplot([25,3.5,'b=2']):
>b4:=textplot([25,4,'b=4']):
>b13:=textplot([25,4.5,'b=13']):
>b26:=textplot([25,5,'b=26']):
>b52:=textplot([25,5.5,'b=52']):
>display(F,b1,b2,b4,b13);
>plotsetup(default):
```

Skriver neste plott til filen H2\_ASE\_C\_stjerne\_c1\_L1\_m2\_H2.eps  
hvis denne linjen aktiveres:

```
>plotsetup(cps,plotoutput="H2_ASE_C_stjerne_c1_L1_m2_H2.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");
Plotter ASE for m=1, H=1 og b={1,2,4,13,26,52} for MLD:
>with(plots):
>F:=plot({data_MLD_m2_H2_b1,data_MLD_m2_H2_b2,data_MLD_m2_H2_b4,
          data_MLD_m2_H2_b13},x=10..30, color=[black,red,green,blue],
          axes=BOXED, labels=["dB","bits/s/Hz"],
          labeldirections=[horizontal,vertical]):
>b1:=textplot([25,4.7,'b=1']):
>b2:=textplot([25,5.1,'b=2']):
>b4:=textplot([25,5.5,'b=4']):
>b13:=textplot([25,5.9,'b=13']):
>b26:=textplot([25,5.9,'b=26']):
>b52:=textplot([25,6.5,'b=52']):
>display(F,b1,b2,b4,b13);
>plotsetup(default):
```

Skriver neste plott til filen H2\_ASE\_C\_stjerne\_c1\_L1\_m2\_H4.eps  
hvis denne linjen aktiveres:

```
>plotsetup(cps,plotoutput="H2_ASE_C_stjerne_c1_L1_m2_H4.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");
```



```

Plotter ASE for m=1, H=1 og b={1,2,4,13,26,52} for MLD:
>with(plots):
>F:=plot({data_MLD_m2_H4_b1,data_MLD_m2_H4_b2,data_MLD_m2_H4_b4,
          data_MLD_m2_H4_b13},x=10..30, color=[red,green,blue,black],
          axes=BOXED, labels=["dB","bits/s/Hz"],
          labeldirections=[horizontal,vertical]):
>b1:=textplot([25,5.9,'b=1']):
>b2:=textplot([25,6.3,'b=2']):
>b4:=textplot([25,6.7,'b=4']):
>b13:=textplot([25,7.1,'b=13']):
>b26:=textplot([25,6.9,'b=26']):
>b52:=textplot([25,7.5,'b=52']):
>display(F,b1,b2,b4,b13);
>plotsetup(default):

```

Skriver neste plott til filen H2\_ASE\_C\_stjerne\_c1\_L1\_m1\_H1.eps  
hvis denne linjen aktiveres:

```

>plotsetup(cps,plotoutput="H2_ASE_RO_stjerne_c1_L1_m1_H1.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");

```

```

Plotter ASE for m=1, H=1 og b={1,2,4,13,26,52} for SD:
>with(plots):
>F:=plot({data_SD_m1_H1_b1,data_SD_m1_H1_b2,data_SD_m1_H1_b4,
          data_SD_m1_H1_b13},x=10..30, color=[red,green,blue,black],
          axes=BOXED, labels=["dB","bits/s/Hz"],
          labeldirections=[horizontal,vertical]):
>b1:=textplot([25,0.9,'b=1']):
>b2:=textplot([25,1.8,'b=2']):
>b4:=textplot([25,2.6,'b=4']):
>b13:=textplot([25,3.7,'b=13']):
>b26:=textplot([25,4.5,'b=26']):
>b52:=textplot([25,5.3,'b=52']):
>display(F,b1,b2,b4,b13);
>plotsetup(default):

```

Skriver neste plott til filen H2\_ASE\_RO\_stjerne\_c1\_L1\_m1\_H2.eps  
hvis denne linjen aktiveres:

```

>plotsetup(cps,plotoutput="H2_ASE_RO_stjerne_c1_L1_m1_H2.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");

```

```

Plotter ASE for m=1, H=1 og b={1,2,4,13,26,52} for SD:
>with(plots):
>F:=plot({data_SD_m1_H2_b1,data_SD_m1_H2_b2,data_SD_m1_H2_b4,
          data_SD_m1_H2_b13},x=10..30, color=[green,black,red,blue],
          axes=BOXED, labels=["dB","bits/s/Hz"],
          labeldirections=[horizontal,vertical]):

```

```

>b1:=textplot([25,3.2,'b=1']):
>b2:=textplot([25,3.7,'b=2']):
>b4:=textplot([25,4.3,'b=4']):
>b13:=textplot([25,5.1,'b=13']):
>b26:=textplot([25,5.9,'b=26']):
>b52:=textplot([25,6.5,'b=52']):
>display(F,b1,b2,b4,b13);
>plotsetup(default):

```

Skriver neste plott til filen H2\_ASE\_RO\_stjerne\_c1\_L1\_m1\_H4.eps  
hvis denne linjen aktiveres:

```

>plotsetup(cps,plotoutput="H2_ASE_RO_stjerne_c1_L1_m1_H4.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");

```

Plotter ASE for  $m=1$ ,  $H=1$  og  $b=\{1,2,4,13,26,52\}$  for SD:

```

>with(plots):
>F:=plot({data_SD_m1_H4_b1,data_SD_m1_H4_b2,data_SD_m1_H4_b4,
          data_SD_m1_H4_b13},x=10..30, color=[blue,red,black,green],
          axes=BOXED, labels=["dB","bits/s/Hz"],
          labeldirections=[horizontal,vertical]):
>b1:=textplot([25,4.7,'b=1']):
>b2:=textplot([25,5.2,'b=2']):
>b4:=textplot([25,5.7,'b=4']):
>b13:=textplot([25,6.3,'b=13']):
>b26:=textplot([25,6.9,'b=26']):
>b52:=textplot([25,7.5,'b=52']):
>display(F,b1,b2,b4,b13);
>plotsetup(default):

```

Skriver neste plott til filen H2\_ASE\_RO\_stjerne\_c1\_L1\_m2\_H1.eps  
hvis denne linjen aktiveres:

```

>plotsetup(cps,plotoutput="H2_ASE_RO_stjerne_c1_L1_m2_H1.eps",
           plotoptions="portrait,height=170pt,width=400pt,noborder,
           leftmargin=0,bottommargin=0");

```

Plotter ASE for  $m=1$ ,  $H=1$  og  $b=\{1,2,4,13,26,52\}$  for SD:

```

>with(plots):
>F:=plot({data_SD_m2_H1_b1,data_SD_m2_H1_b2,data_SD_m2_H1_b4,
          data_SD_m2_H1_b13},x=10..30, color=[blue,red,black,green],
          axes=BOXED, labels=["dB","bits/s/Hz"],
          labeldirections=[horizontal,vertical]):
>b1:=textplot([25,2.7,'b=1']):
>b2:=textplot([25,3.1,'b=2']):
>b4:=textplot([25,3.7,'b=4']):
>b13:=textplot([25,4.5,'b=13']):
>b26:=textplot([25,5,'b=26']):
>b52:=textplot([25,5.5,'b=52']):
>display(F,b1,b2,b4,b13);

```

```
>plotsetup(default):
```

Skriver neste plott til filen H2\_ASE\_RO\_stjerne\_c1\_L1\_m2\_H2.eps  
hvis denne linjen aktiveres:

```
>plotsetup(cps,plotoutput="H2_ASE_RO_stjerne_c1_L1_m2_H2.eps",  
           plotoptions="portrait,height=170pt,width=400pt,noborder,  
           leftmargin=0,bottommargin=0");
```

Plotter ASE for  $m=1$ ,  $H=1$  og  $b=\{1,2,4,13,26,52\}$  for SD:

```
>with(plots):  
>F:=plot({data_SD_m2_H2_b1,data_SD_m2_H2_b2,data_SD_m2_H2_b4,  
          data_SD_m2_H2_b13},x=10..30, color=[black,red,green,blue],  
          axes=BOXED, labels=["dB","bits/s/Hz"],  
          labeldirections=[horizontal,vertical]):  
>b1:=textplot([25,4.2,'b=1']):  
>b2:=textplot([25,4.7,'b=2']):  
>b4:=textplot([25,5.2,'b=4']):  
>b13:=textplot([25,5.7,'b=13']):  
>b26:=textplot([25,5.9,'b=26']):  
>b52:=textplot([25,6.5,'b=52']):  
>display(F,b1,b2,b4,b13);  
>plotsetup(default):
```

Skriver neste plott til filen H2\_ASE\_RO\_stjerne\_c1\_L1\_m2\_H4.eps  
hvis denne linjen aktiveres:

```
>plotsetup(cps,plotoutput="H2_ASE_RO_stjerne_c1_L1_m2_H4.eps",  
           plotoptions="portrait,height=170pt,width=400pt,noborder,  
           leftmargin=0,bottommargin=0");
```

Plotter ASE for  $m=1$ ,  $H=1$  og  $b=\{1,2,4,13,26,52\}$  for SD:

```
>with(plots):  
>F:=plot({data_SD_m2_H4_b1,data_SD_m2_H4_b2,data_SD_m2_H4_b4,  
          data_SD_m2_H4_b13},x=10..30, color=[blue,green,red,black],  
          axes=BOXED, labels=["dB","bits/s/Hz"],  
          labeldirections=[horizontal,vertical]):  
>b1:=textplot([25,5.5,'b=1']):  
>b2:=textplot([25,6.0,'b=2']):  
>b4:=textplot([25,6.5,'b=4']):  
>b13:=textplot([25,7.0,'b=13']):  
>b26:=textplot([25,6.9,'b=26']):  
>b52:=textplot([25,7.5,'b=52']):  
>display(F,b1,b2,b4,b13);  
>plotsetup(default):
```

## /kode/OFDM/sammenligning\_totalASE\_Hiperlan2\_OFDM-AKM-QAM/MLD\_sammenligning.mws

```
>restart;
```

Sannsynlighet for at gamma havner i n-te fading region.

```
>probNakagami := proc(g,g1,g2,m,H)
> (GAMMA(H*m,m*g1/g)-GAMMA(H*m,m*g2/g))/GAMMA(H*m);
>end proc;
```

Prosedyre for konvertere vekk dB

```
>remove_dB := proc(x_dB)
> 10 ^ (x_dB/10);
>end;
```

Inndataene:

```
>SE := [1,2,3,4,5,6,7,8,9,10,11]:
>gMLD_dB := [0.83,6.14,9.77,13.12,16.55,19.59,22.86,25.79,29.0,31.9,35.1,1000]:
>Q:=52:
>Wtotal:=16.26:
>Ttotal:=4:
```

Konverter vekk dB-en...

```
>for i from 1 to 12 do
> gMLD[i]:=remove_dB(gMLD_dB[i]);
>end do;
```

Prosedyre som beregner total ASE for OFDM:

```
>totalASE := proc(g_dB,H)
> local g,s,i;
> g := remove_dB(g_dB);
> s := 0;
> for i from 1 to 11 do
> s:= s + SE[i] * probNakagami(g,gMLD[i],gMLD[i+1],m,H);;
> end do;
> s := s*Q/(Ttotal*Wtotal);
>end proc;
```

Prosedyre for å sortere lister med (x,y)-punkter. Sortering etter x-verdi.

```
>sorter:=proc(x,y)
> if x[1] < y[1] then true
> elif x[1] > y[1] then false
> else evalb(x[2] < y[2])
> fi
>end;
```

Prosedyre som genererer en liste med (x,y)-punkter fra en liste med x-verdier og en tilhørende liste med y-verdier

```
>createListOfPoints := proc(listX,listY)
```

```
> [seq([op(i,listX),op(i,listY)],i=1..nops(listX))];
>end proc:
```

Prosedyre som henter ut y-verdiene fra en liste med punkter

```
>getY:=proc(pointList)
> [seq(op([i,2],pointList),i=1..nops(pointList))];
>end proc:
```

Prosedyre som henter ut x-verdiene fra en liste med punkter

```
>getX:=proc(pointList)
> [seq(op([i,1],pointList),i=1..nops(pointList))];
>end proc:
```

Prosedyre som sjekker om to lister med en kolonne er like.

```
>isEqual:=proc(listA,listB)
> local i;
> if (nops(listA)<>nops(listB)) then
> return false;
> else
> for i from 1 to nops(list(A)) do
> if (not verify(listA[i],listB[i],'equal')) then return i; end if;
> end do;
> end if;
> return true;
>end proc:
```

Prosedyre som finner elementvis differanse mellom to sorterte lister med punkter. Returnerer en ny liste med punkter.

```
>pointDifference := proc(pointsA,pointsB)
> local i;
> if (not isEqual(getX(pointsA),getX(pointsB))) then
> error ". Cannot compare lists, x-values don't match!"
> else
> [seq([pointsA[i,1],evalf(pointsA[i,2]-pointsB[i,2])],i=1..nops(pointsA))];
> end if;
>end proc:
```

Prosedyre som velger ut punkter fra ønsket område fra en sortert liste med punkter.

```
>pointsInRange:=proc(points,xMin,xMax)
> local i,minPos,maxPos;
> for i from 1 to nops(points) do
> if (points[i,1]=xMin) then minPos:=i;
> elif (points[i,1]=xMax) then maxPos:=i;
> end if;
> end do;
> [seq([points[i,1],points[i,2]],i=minPos..maxPos)];
>end proc:
```

Leser inn HIPERLAN/2 datapunkter fra fil og legger dem i sorterte lister

```

>ASE_HIPERLAN2_MLD_m1_H1_b1:=sort(readdata(ASE_MLD_K100000_m1_H1_b1,2),sorter):
>ASE_HIPERLAN2_MLD_m1_H1_b2:=sort(readdata(ASE_MLD_K100000_m1_H1_b2,2),sorter):
>ASE_HIPERLAN2_MLD_m1_H1_b4:=sort(readdata(ASE_MLD_K100000_m1_H1_b4,2),sorter):
>ASE_HIPERLAN2_MLD_m1_H1_b13:=sort(readdata(ASE_MLD_K100000_m1_H1_b13,2),sorter):
>ASE_HIPERLAN2_MLD_m2_H1_b1:=sort(readdata(ASE_MLD_K100000_m2_H1_b1,2),sorter):
>ASE_HIPERLAN2_MLD_m2_H1_b2:=sort(readdata(ASE_MLD_K100000_m2_H1_b2,2),sorter):
>ASE_HIPERLAN2_MLD_m2_H1_b4:=sort(readdata(ASE_MLD_K100000_m2_H1_b4,2),sorter):
>ASE_HIPERLAN2_MLD_m2_H1_b13:=sort(readdata(ASE_MLD_K100000_m2_H1_b13,2),sorter):
>ASE_HIPERLAN2_MLD_m1_H2_b1:=sort(readdata(ASE_MLD_K100000_m1_H2_b1,2),sorter):
>ASE_HIPERLAN2_MLD_m1_H2_b2:=sort(readdata(ASE_MLD_K100000_m1_H2_b2,2),sorter):
>ASE_HIPERLAN2_MLD_m1_H2_b4:=sort(readdata(ASE_MLD_K100000_m1_H2_b4,2),sorter):
>ASE_HIPERLAN2_MLD_m1_H2_b13:=sort(readdata(ASE_MLD_K100000_m1_H2_b13,2),sorter):
>ASE_HIPERLAN2_MLD_m2_H2_b1:=sort(readdata(ASE_MLD_K100000_m2_H2_b1,2),sorter):
>ASE_HIPERLAN2_MLD_m2_H2_b2:=sort(readdata(ASE_MLD_K100000_m2_H2_b2,2),sorter):
>ASE_HIPERLAN2_MLD_m2_H2_b4:=sort(readdata(ASE_MLD_K100000_m2_H2_b4,2),sorter):
>ASE_HIPERLAN2_MLD_m2_H2_b13:=sort(readdata(ASE_MLD_K100000_m2_H2_b13,2),sorter):
>ASE_HIPERLAN2_MLD_m1_H4_b1:=sort(readdata(ASE_MLD_K100000_m1_H4_b1,2),sorter):
>ASE_HIPERLAN2_MLD_m1_H4_b2:=sort(readdata(ASE_MLD_K100000_m1_H4_b2,2),sorter):
>ASE_HIPERLAN2_MLD_m1_H4_b4:=sort(readdata(ASE_MLD_K100000_m1_H4_b4,2),sorter):
>ASE_HIPERLAN2_MLD_m1_H4_b13:=sort(readdata(ASE_MLD_K100000_m1_H4_b13,2),sorter):
>ASE_HIPERLAN2_MLD_m2_H4_b1:=sort(readdata(ASE_MLD_K100000_m2_H4_b1,2),sorter):
<ASE_HIPERLAN2_MLD_m2_H4_b2:=sort(readdata(ASE_MLD_K100000_m2_H4_b2,2),sorter):
>ASE_HIPERLAN2_MLD_m2_H4_b4:=sort(readdata(ASE_MLD_K100000_m2_H4_b4,2),sorter):
>ASE_HIPERLAN2_MLD_m2_H4_b13:=sort(readdata(ASE_MLD_K100000_m2_H4_b13,2),sorter):

```

Beregner plottepunkter for m=1 og H=1

```

>m:=1:
>h:=1:
>dBfrom:=10:
>dBto:=30:
>dBstep:=1/4:
>x:=seq(evalf(x*dBstep),x=dBfrom/dBstep..dBto/dBstep):
>y:=seq(evalf(totalASE(x*dBstep,h)),x=dBfrom/dBstep..dBto/dBstep):
>diff_AKM_Hiperlan2_m1_H1:=pointDifference(pointsInRange(createListOfPoints(
[x],[y]),dBfrom,dBto),pointsInRange(ASE_HIPERLAN2_MLD_m1_H1_b13,dBfrom,dBto)):

```

Beregner plottepunkter for m=1 og H=2

```

>m:=1:
>h:=2:
>dBfrom:=10:
>dBto:=30:
>dBstep:=1/4:
>x:=seq(evalf(x*dBstep),x=dBfrom/dBstep..dBto/dBstep):
>y:=seq(evalf(totalASE(x*dBstep,h)),x=dBfrom/dBstep..dBto/dBstep):
>diff_AKM_Hiperlan2_m1_H2:=pointDifference(pointsInRange(createListOfPoints(
[x],[y]),dBfrom,dBto),pointsInRange(ASE_HIPERLAN2_MLD_m1_H2_b13,dBfrom,dBto)):

```

Beregner plottepunkter for m=1 og H=4

```

>m:=1:

```

```

>h:=4:
>dBfrom:=10:
>dBto:=30:
>dBstep:=1/4:
>x:=seq(evalf(x*dBstep),x=dBfrom/dBstep..dBto/dBstep):
>y:=seq(evalf(totalASE(x*dBstep,h)),x=dBfrom/dBstep..dBto/dBstep):
>diff_AKM_Hiperlan2_m1_H4:=pointDifference(pointsInRange(createListOfPoints(
[x],[y]),dBfrom,dBto),pointsInRange(ASE_HIPERLAN2_MLD_m1_H4_b13,dBfrom,dBto)):

```

Beregner plottepunkter for m=2 og H=1

```

>m:=2:
>h:=1:
>dBfrom:=10:
>dBto:=30:
>dBstep:=1/4:
>x:=seq(evalf(x*dBstep),x=dBfrom/dBstep..dBto/dBstep):
>y:=seq(evalf(totalASE(x*dBstep,h)),x=dBfrom/dBstep..dBto/dBstep):
>diff_AKM_Hiperlan2_m2_H1:=pointDifference(pointsInRange(createListOfPoints(
[x],[y]),dBfrom,dBto),pointsInRange(ASE_HIPERLAN2_MLD_m2_H1_b13,dBfrom,dBto)):

```

Beregner plottepunkter for m=2 og H=2

```

>m:=2:
>h:=2:
>dBfrom:=10:
>dBto:=30:
>dBstep:=1/4:
>x:=seq(evalf(x*dBstep),x=dBfrom/dBstep..dBto/dBstep):
>y:=seq(evalf(totalASE(x*dBstep,h)),x=dBfrom/dBstep..dBto/dBstep):
>diff_AKM_Hiperlan2_m2_H2:=pointDifference(pointsInRange(createListOfPoints(
[x],[y]),dBfrom,dBto),pointsInRange(ASE_HIPERLAN2_MLD_m2_H2_b13,dBfrom,dBto)):

```

Beregner plottepunkter for m=2 og H=4

```

>m:=2:
>h:=4:
>dBfrom:=10:
>dBto:=30:
>dBstep:=1/4:
>x:=seq(evalf(x*dBstep),x=dBfrom/dBstep..dBto/dBstep):
>y:=seq(evalf(totalASE(x*dBstep,h)),x=dBfrom/dBstep..dBto/dBstep):
>diff_AKM_Hiperlan2_m2_H4:=pointDifference(pointsInRange(createListOfPoints(
[x],[y]),dBfrom,dBto),pointsInRange(ASE_HIPERLAN2_MLD_m2_H4_b13,dBfrom,dBto)):

```

Skriver neste plott til filen diff\_ASE\_AKM\_Hiperlan2\_MLD\_m1.eps

hvis denne linjen aktiveres:

```

>plotsetup(cps,plotoutput="diff_ASE_AKM_Hiperlan2_MLD_m1.eps",
plotoptions="portrait,height=170pt,width=400pt,noborder,
leftmargin=0,bottommargin=0");

```

Plotter differansen i ASE mellom OFDM/AKM og Hiperlan2 for MLD og m=1,h={1,2,4}

```

>with(plots):
>F:=plot([diff_AKM_Hiperlan2_m1_H1,diff_AKM_Hiperlan2_m1_H2,
        diff_AKM_Hiperlan2_m1_H4], color=[red,green,blue],
        axes=BOXED,labels=["dB","bits/s/Hz"],
        labeldirections=[horizontal,vertical]):
>H1:=textplot([25,1.45,'H=1']):
>H2:=textplot([25,1.05,'H=2']):
>H4:=textplot([25,0.6,'H=4']):
>display(F,H1,H2,H4);
>plotsetup(default):

```

Skriver neste plott til filen diff\_ASE\_AKM\_Hiperlan2\_MLD\_m2.eps

hvis denne linjen aktiveres:

```

>plotsetup(cps,plotoutput="diff_ASE_AKM_Hiperlan2_MLD_m2.eps",
        plotoptions="portrait,height=170ps,width=400pt,noborder,
        leftmargin=0,bottommargin=0");

```

Plotter differansen i ASE mellom OFDM/AKM og Hiperlan2 for MLD og  $m=2, h=\{1,2,4\}$

```

>with(plots):
>F:=plot([diff_AKM_Hiperlan2_m2_H1,diff_AKM_Hiperlan2_m2_H2,
        diff_AKM_Hiperlan2_m2_H4], color=[red,green,blue],
        axes=BOXED,labels=["dB","bits/s/Hz"],
        labeldirections=[horizontal,vertical]):
>H1:=textplot([25,0.92,'H=1']):
>H2:=textplot([25,0.68,'H=2']):
>H4:=textplot([25,0.5,'H=4']):
>display(F,H1,H2,H4);
>plotsetup(default):

```



## /kode/OFDM/sammenligning\_totalASE\_Hiperlan2\_OFDM-AKM-QAM/SD\_sammenligning.mws

```
>restart;
```

Sannsynlighet for at gamma havner i n-te fading region.

```
>probNakagami := proc(g,g1,g2,m,H)
> (GAMMA(H*m,m*g1/g)-GAMMA(H*m,m*g2/g))/GAMMA(H*m);
>end proc:
```

Prosedyre for konvertere vekk dB

```
>remove_dB := proc(x_dB)
> 10 ^ (x_dB/10);
>end:
```

Inndataene:

```
>SE := [1,2,3,4,5,6,7,8,9,10,11]:
>gSD_dB := [2.478,7.680,11.096,14.405,17.811,20.817,
24.087,27.030,30.215,33.135,36.272,1000]:
>Q:=52:
>Wtotal:=16.26:
>Ttotal:=4:
```

Konverter vekk dB-en...

```
>for i from 1 to 12 do
> gSD[i]:=remove_dB(gSD_dB[i]);
>end do:
```

Prosedyre som beregner total ASE for OFDM:

```
>totalASE := proc(g_dB,H)
> local g,s,i;
> g := remove_dB(g_dB);
> s := 0;
> for i from 1 to 11 do
> s:= s + SE[i] * probNakagami(g,gSD[i],gSD[i+1],m,H);;
> end do;
> s := s*Q/(Ttotal*Wtotal);
>end proc:
```

Prosedyre for sortere lister med (x,y)-punkter. Sortering etter x-verdi.

```
>sorter:=proc(x,y)
> if x[1] < y[1] then true
> elif x[1] > y[1] then false
> else evalb(x[2] < y[2])
> fi
>end:
```

Prosedyre som genererer en liste med (x,y)-punkter fra en liste med x-verdier og en tilhørende liste med y-verdier

```

>createListOfPoints := proc(listX,listY)
>  [seq([op(i,listX),op(i,listY)],i=1..nops(listX))];
>end proc:

```

Prosedyre som henter ut y-verdiene fra en liste med punkter

```

>getY:=proc(pointList)
>  [seq(op([i,2],pointList),i=1..nops(pointList))];
>end proc:

```

Prosedyre som henter ut x-verdiene fra en liste med punkter

```

>getX:=proc(pointList)
>  [seq(op([i,1],pointList),i=1..nops(pointList))];
>end proc:

```

Prosedyre som sjekker om to lister med en kolonne er like.

```

>isEqual:=proc(listA,listB)
>  local i;
>  if (nops(listA)<>nops(listB)) then
>    return false;
>  else
>    for i from 1 to nops(list(A)) do
>      if (not verify(listA[i],listB[i],'equal')) then return i; end if;
>    end do;
>  end if;
>  return true;
>end proc:

```

Prosedyre som finner elementvis differanse mellom to sorterte lister med punkter. Returnerer en ny liste med punkter.

```

>pointDifference := proc(pointsA,pointsB)
>  local i;
>  if (not isEqual(getX(pointsA),getX(pointsB))) then
>    error ". Cannot compare lists, x-values don't match!"
>  else
>    [seq([pointsA[i,1],evalf(pointsA[i,2]-pointsB[i,2])],i=1..nops(pointsA))];
>  end if;
>end proc:

```

Prosedyre som velger ut punkter fra nsket omrde fra en sortert liste med punkter.

```

>pointsInRange:=proc(points,xMin,xMax)
>  local i,minPos,maxPos;
>  for i from 1 to nops(points) do
>    if (points[i,1]=xMin) then minPos:=i;
>    elif (points[i,1]=xMax) then maxPos:=i;
>    end if;
>  end do;
>  [seq([points[i,1],points[i,2]],i=minPos..maxPos)];
>end proc:

```

```

Leser inn HIPERLAN/2 datapunkter fra fil og legger dem i sorterte lister
>ASE_HIPERLAN2_SD_m1_H1_b1:=sort(readdata(ASE_SD_K100000_m1_H1_b1,2),sorter):
>ASE_HIPERLAN2_SD_m1_H1_b2:=sort(readdata(ASE_SD_K100000_m1_H1_b2,2),sorter):
>ASE_HIPERLAN2_SD_m1_H1_b4:=sort(readdata(ASE_SD_K100000_m1_H1_b4,2),sorter):
>ASE_HIPERLAN2_SD_m1_H1_b13:=sort(readdata(ASE_SD_K100000_m1_H1_b13,2),sorter):
>ASE_HIPERLAN2_SD_m2_H1_b1:=sort(readdata(ASE_SD_K100000_m2_H1_b1,2),sorter):
>ASE_HIPERLAN2_SD_m2_H1_b2:=sort(readdata(ASE_SD_K100000_m2_H1_b2,2),sorter):
>ASE_HIPERLAN2_SD_m2_H1_b4:=sort(readdata(ASE_SD_K100000_m2_H1_b4,2),sorter):
>ASE_HIPERLAN2_SD_m2_H1_b13:=sort(readdata(ASE_SD_K100000_m2_H1_b13,2),sorter):
>ASE_HIPERLAN2_SD_m1_H2_b1:=sort(readdata(ASE_SD_K100000_m1_H2_b1,2),sorter):
>ASE_HIPERLAN2_SD_m1_H2_b2:=sort(readdata(ASE_SD_K100000_m1_H2_b2,2),sorter):
>ASE_HIPERLAN2_SD_m1_H2_b4:=sort(readdata(ASE_SD_K100000_m1_H2_b4,2),sorter):
>ASE_HIPERLAN2_SD_m1_H2_b13:=sort(readdata(ASE_SD_K100000_m1_H2_b13,2),sorter):
>ASE_HIPERLAN2_SD_m2_H2_b1:=sort(readdata(ASE_SD_K100000_m2_H2_b1,2),sorter):
>ASE_HIPERLAN2_SD_m2_H2_b2:=sort(readdata(ASE_SD_K100000_m2_H2_b2,2),sorter):
>ASE_HIPERLAN2_SD_m2_H2_b4:=sort(readdata(ASE_SD_K100000_m2_H2_b4,2),sorter):
>ASE_HIPERLAN2_SD_m2_H2_b13:=sort(readdata(ASE_SD_K100000_m2_H2_b13,2),sorter):
>ASE_HIPERLAN2_SD_m1_H4_b1:=sort(readdata(ASE_SD_K100000_m1_H4_b1,2),sorter):
>ASE_HIPERLAN2_SD_m1_H4_b2:=sort(readdata(ASE_SD_K100000_m1_H4_b2,2),sorter):
>ASE_HIPERLAN2_SD_m1_H4_b4:=sort(readdata(ASE_SD_K100000_m1_H4_b4,2),sorter):
>ASE_HIPERLAN2_SD_m1_H4_b13:=sort(readdata(ASE_SD_K100000_m1_H4_b13,2),sorter):
>ASE_HIPERLAN2_SD_m2_H4_b1:=sort(readdata(ASE_SD_K100000_m2_H4_b1,2),sorter):
>ASE_HIPERLAN2_SD_m2_H4_b2:=sort(readdata(ASE_SD_K100000_m2_H4_b2,2),sorter):
>ASE_HIPERLAN2_SD_m2_H4_b4:=sort(readdata(ASE_SD_K100000_m2_H4_b4,2),sorter):
>ASE_HIPERLAN2_SD_m2_H4_b13:=sort(readdata(ASE_SD_K100000_m2_H4_b13,2),sorter):

```

Beregner plottepunkter for m=1 og H=1

```

>m:=1:
>h:=1:
>dBfrom:=10:
>dBto:=30:
>dBstep:=1/4:
>x:=seq(evalf(x*dBstep),x=dBfrom/dBstep..dBto/dBstep):
>y:=seq(evalf(totalASE(x*dBstep,h)),x=dBfrom/dBstep..dBto/dBstep):
>diff_AKM_Hiperlan2_m1_H1:=pointDifference(pointsInRange(createListOfPoints(
    [x],[y]),dBfrom,dBto),pointsInRange(ASE_HIPERLAN2_SD_m1_H1_b13,dBfrom,dBto)):

```

Beregner plottepunkter for m=1 og H=2

```

>m:=1:
>h:=2:
>dBfrom:=10:
>dBto:=30:
>dBstep:=1/4:
>x:=seq(evalf(x*dBstep),x=dBfrom/dBstep..dBto/dBstep):
>y:=seq(evalf(totalASE(x*dBstep,h)),x=dBfrom/dBstep..dBto/dBstep):
>diff_AKM_Hiperlan2_m1_H2:=pointDifference(pointsInRange(createListOfPoints(
    [x],[y]),dBfrom,dBto),pointsInRange(ASE_HIPERLAN2_SD_m1_H2_b13,dBfrom,dBto)):

```

Beregner plottepunkter for m=1 og H=4

```

>m:=1:
>h:=4:
>dBfrom:=10:
>dBto:=30:
>dBstep:=1/4:
>x:=seq(evalf(x*dBstep),x=dBfrom/dBstep..dBto/dBstep):
>y:=seq(evalf(totalASE(x*dBstep,h)),x=dBfrom/dBstep..dBto/dBstep):
>diff_AKM_Hiperlan2_m1_H4:=pointDifference(pointsInRange(createListOfPoints(
    [x],[y]),dBfrom,dBto),pointsInRange(ASE_HIPERLAN2_SD_m1_H4_b13,dBfrom,dBto)):

```

Beregner plottepunkter for m=2 og H=1

```

>m:=2:
>h:=1:
>dBfrom:=10:
>dBto:=30:
>dBstep:=1/4:
>x:=seq(evalf(x*dBstep),x=dBfrom/dBstep..dBto/dBstep):
>y:=seq(evalf(totalASE(x*dBstep,h)),x=dBfrom/dBstep..dBto/dBstep):
>diff_AKM_Hiperlan2_m2_H1:=pointDifference(pointsInRange(createListOfPoints(
    [x],[y]),dBfrom,dBto),pointsInRange(ASE_HIPERLAN2_SD_m2_H1_b13,dBfrom,dBto)):

```

Beregner plottepunkter for m=2 og H=2

```

>m:=2:
>h:=2:
>dBfrom:=10:
>dBto:=30:
>dBstep:=1/4:
>x:=seq(evalf(x*dBstep),x=dBfrom/dBstep..dBto/dBstep):
>y:=seq(evalf(totalASE(x*dBstep,h)),x=dBfrom/dBstep..dBto/dBstep):
>diff_AKM_Hiperlan2_m2_H2:=pointDifference(pointsInRange(createListOfPoints(
    [x],[y]),dBfrom,dBto),pointsInRange(ASE_HIPERLAN2_SD_m2_H2_b13,dBfrom,dBto)):

```

Beregner plottepunkter for m=2 og H=4

```

>m:=2:
>h:=4:
>dBfrom:=10:
>dBto:=30:
>dBstep:=1/4:
>x:=seq(evalf(x*dBstep),x=dBfrom/dBstep..dBto/dBstep):
>y:=seq(evalf(totalASE(x*dBstep,h)),x=dBfrom/dBstep..dBto/dBstep):
>diff_AKM_Hiperlan2_m2_H4:=pointDifference(pointsInRange(createListOfPoints(
    [x],[y]),dBfrom,dBto),pointsInRange(ASE_HIPERLAN2_SD_m2_H4_b13,dBfrom,dBto)):

```

Skriver neste plott til filen diff\_ASE\_AKM\_Hiperlan2\_SD\_m1.eps

hvis denne linjen aktiveres:

```

>plotsetup(cps,plotoutput="diff_ASE_AKM_Hiperlan2_SD_m1.eps",
    plotoptions="portrait,height=170pt,width=400pt,noborder,
    leftmargin=0,bottommargin=0");

```

```

Plotter differansen i ASE mellom OFDM/AKM og Hiperlan2 for SD og m=1,h={1,2,4}
>with(plots):
>F:=plot([diff_AKM_Hiperlan2_m1_H1,diff_AKM_Hiperlan2_m1_H2,
         diff_AKM_Hiperlan2_m1_H4], color=[red,green,blue],
         axes=BOXED,labels=["dB","bits/s/Hz"],
         labeldirections=[horizontal,vertical]):
>H1:=textplot([25,1.4,'H=1']):
>H2:=textplot([25,1.05,'H=2']):
>H4:=textplot([25,0.72,'H=4']):
>display(F,H1,H2,H4);
>plotsetup(default):

```

```

Skriver neste plott til filen diff_ASE_AKM_Hiperlan2_SD_m2.eps
hvis denne linjen aktiveres:
>plotsetup(cps,plotoutput="diff_ASE_AKM_Hiperlan2_SD_m2.eps",
           plotoptions="portrait,height=170ps,width=400pt,noborder,
           leftmargin=0,bottommargin=0");

```

```

Plotter differansen i ASE mellom OFDM/AKM og Hiperlan2 for SD og m=2,h={1,2,4}
>with(plots):
>F:=plot([diff_AKM_Hiperlan2_m2_H1,diff_AKM_Hiperlan2_m2_H2,
         diff_AKM_Hiperlan2_m2_H4], color=[red,green,blue],
         axes=BOXED,labels=["dB","bits/s/Hz"],
         labeldirections=[horizontal,vertical]):
>H1:=textplot([25,0.92,'H=1']):
>H2:=textplot([25,0.68,'H=2']):
>H4:=textplot([25,0.5,'H=4']):
>display(F,H1,H2,H4);
>plotsetup(default):

```