

# Sekvensiell dekoding av trelliskoder

Jostein Lund  
14. mai 2004





# Oversikt

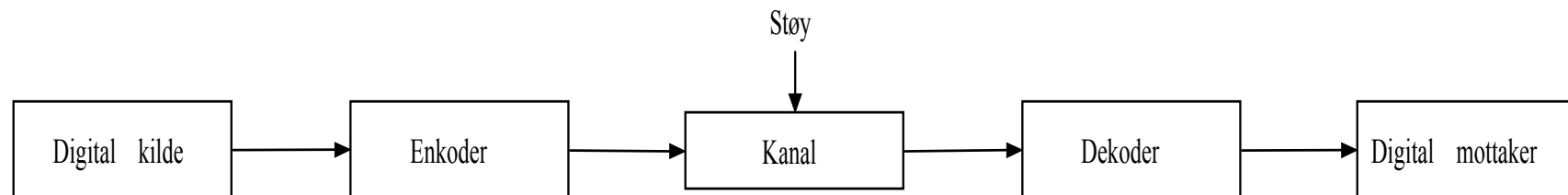
- Introduksjon
- Digital kommunikasjonsmodell
- Signalkonstellasjoner
- Kanalkapasitet og "cutoffrate"
- Konvolusjonskoder
- Sekvensiell dekoding
- Resultater og konklusjon



# Introduksjon

- Digital kommunikasjonsmodell
  - Støy
  - Koding
  - Dekoding

# Digital kommunikasjonsmodell



- Kilde med digital informasjon
- Enkoderen koder informasjonen ved å legge til redundante biter. Enkoderen modulerer også signalet. Moduleringen endrer den digitale informasjonen til analoge signaler
- Signalene overføres over en kanal der de utsettes for støy
- Dekoderen demodulerer signalet samt detekterer og korrigerer eventuelle feil i overføringen
- Digital mottaker mottar den samme informasjonen som ble sendt



# Støy

- Additiv hvit Gaussisk støy (AWGN) benyttes for å modellere støyen
- Sendt signal  $a(t)$ , mottatt signal  $r(t)$ :

$$r(t) = a(t) + n(t).$$

Der  $n(t)$  er normalfordelt støy



# Koding

- Legger til redundante biter
- Dette gjør at feil kan detekteres og korrigeres
- Informasjonsraten blir mindre



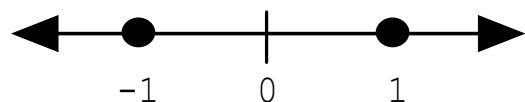
# Dekoding

- Bruker redundante biter til å detektere og korrigere evt. feil i overføringen
- Målet er å få tilbake samme informasjon som ble kodet i enkoderen

# Signalkonstellasjoner

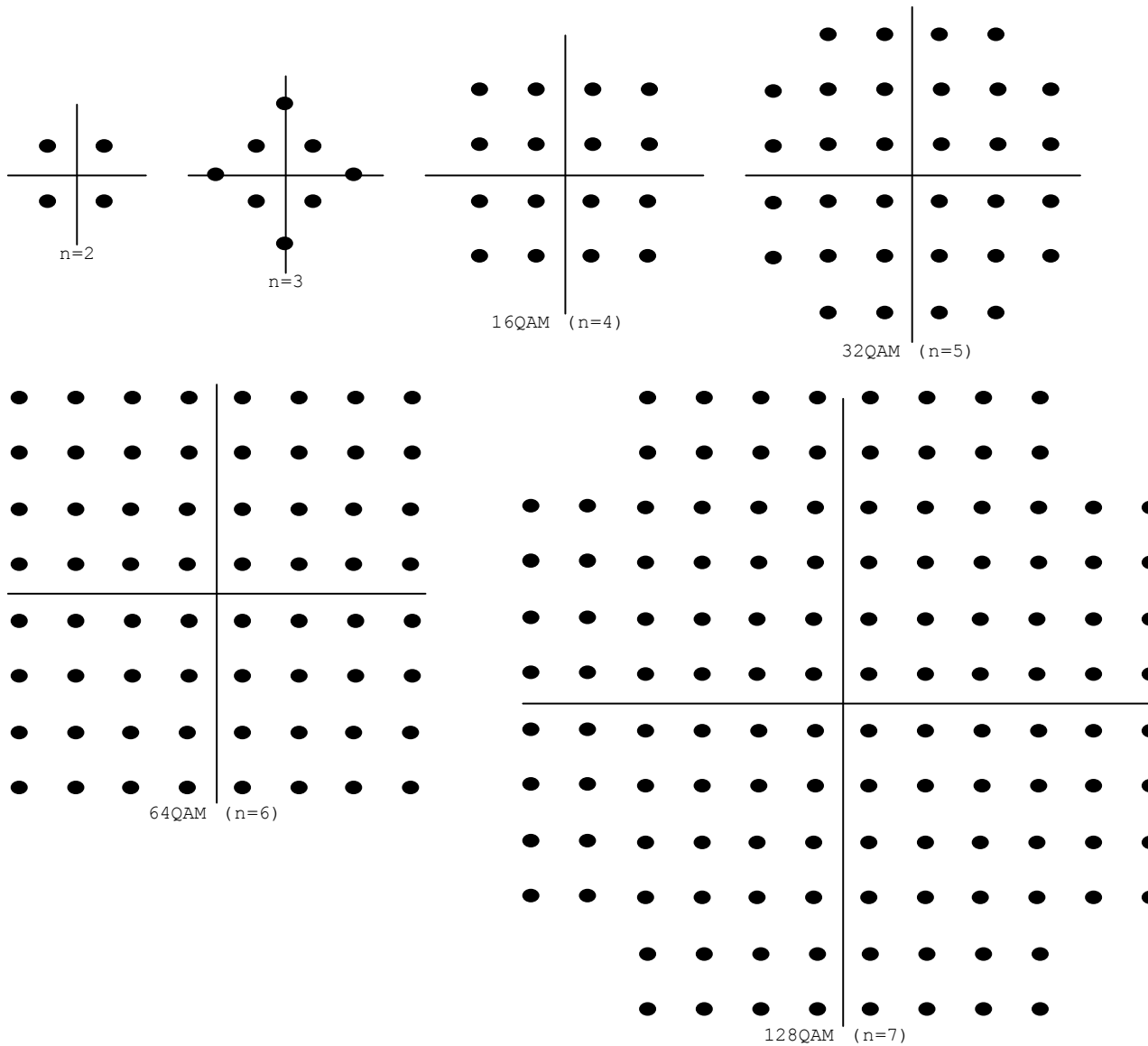
- Digital informasjon gjøres om til signalpunkter i en signalkonstellasjon før de overføres over kanalen

- BPSK-konstellasjon med to signalpunkter:



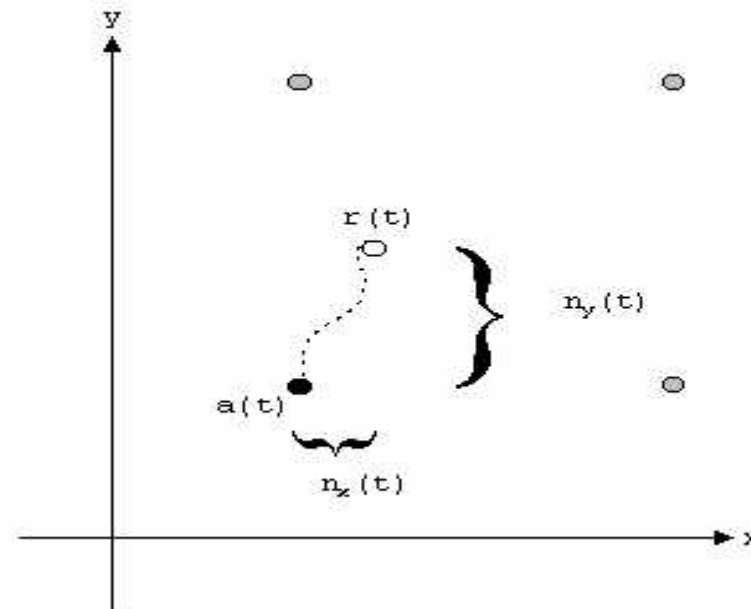
- Signalpunkter egner seg godt til overføring med analoge signaler
- En vanlig type signalkonstellasjon er "Quadrature Amplitude Modulation" (QAM). QAM er 2-dimensjonale signalkonstellasjon

# QAM-konstellasjoner



# Støy i QAM-konstellasjon

- Når et QAM-symbol påvirkes av støy vil punktet vandre i koordinatsystemet
- Hvis punktet vandrer langt nok vil vi få en feil i dekodingen ved bruk av Maximum Likelihood Dekoding (MLD)
- MLD dekker den mottatte sekvensen til den mest sannsynlige kodesekvensen





# Spektral effektivitet

- Antall overførte *informasjonsbiter* pr. sekund pr. Hz båndbredde
- For BPSK-konstellasjonen med kun to signalpunkter trengs kun 1 bit for å representere et symbol. Den spektrale effektiviteten blir derfor 1 bit/sek/Hz
- Hvis dataraten overstiger den tilgjengelige båndbredden kan man øke den spektrale effektiviteten ved å velge en større signalkonstellasjon
- For eksempel for 16-QAM er den spektrale effektiviteten 4 bit/sek/Hz.
- Antall symboler pr. sekund er fortsatt den samme



# Energi

- Et punkt som ligger i origo er definert til å ha 0 energi. Signalkonstellasjoner inneholder derfor ikke punkter som ligger i origo
- Punktene gis odde koordinater med avstand 2 mellom hvert punkt
- Ved å øke den spektrale effektiviteten trengs det også større energi for å opprettholde den samme minimumsdistansen i konstellasjonen
- Man vil spare energi ved å øke antall dimensjoner

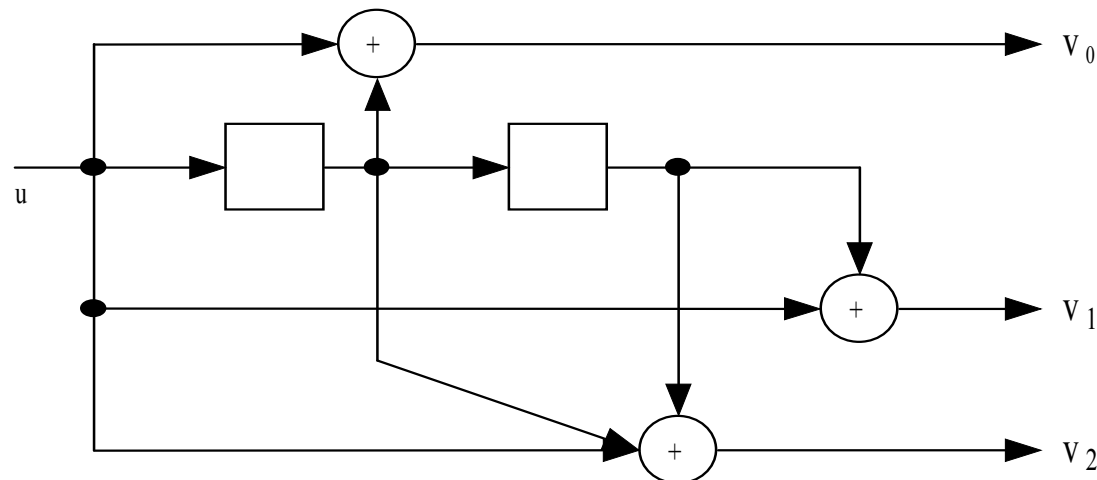


# Kanalkapasitet og ”cutoffrate”

- Shannon har vist at pålitelig kommunikasjon kan oppnås ved koding når overføringsraten (biter/sekund) er mindre enn kanalkapasiteten  $C$ .  $C$  kan bare nås med meget stor kompleksitet i dekodningen
- ”cutoffraten”  $R_0$  er den maksimale overføringsraten der gjennomsnittlig antall operasjoner pr. dekodet bit er bundet for sekvensiell dekodning.  $R_0 \leq C$
- Mens  $R_0$  kun er en funksjon av SNR, finnes  $R_0^*$  som også avhenger av en spesifikk signalkonstellasjon

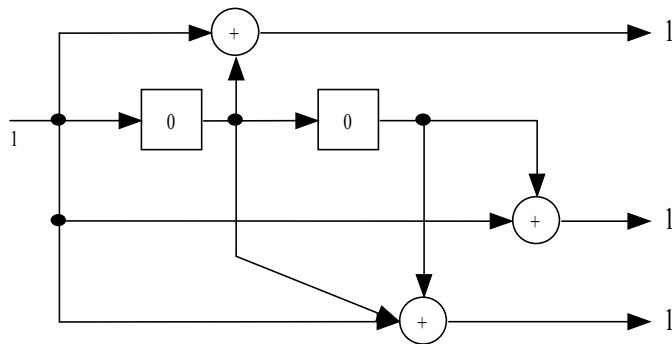
# Konvolusjonskoder

- For en  $(n,k,m)$ -konvolusjonskode gjelder:
  - $n$ -Antall utdata biter ved gitt tidspunkt
  - $k$ -Antall inndata biter ved gitt tidspunkt
  - $m$ -Antall minneblokker i enkoderen. Dette kalles også registerlengden til koden
- Eksempel: Enkoder til  $(3,1,2)$  kode med inndata  $u$  og utdata  $v$

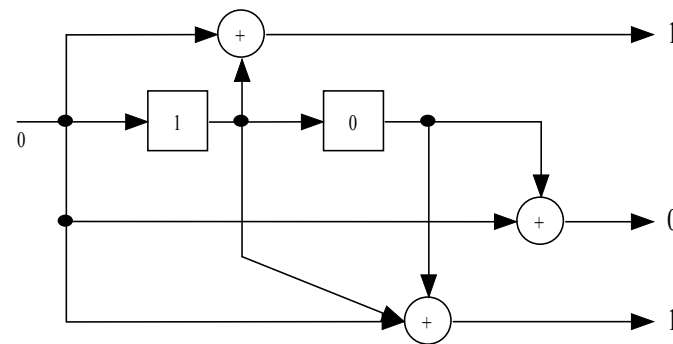


Her kodes bitsekvensen  $u=1001$  og vi får kodesekvensen  $v=111\ 101\ 011\ 111$ .

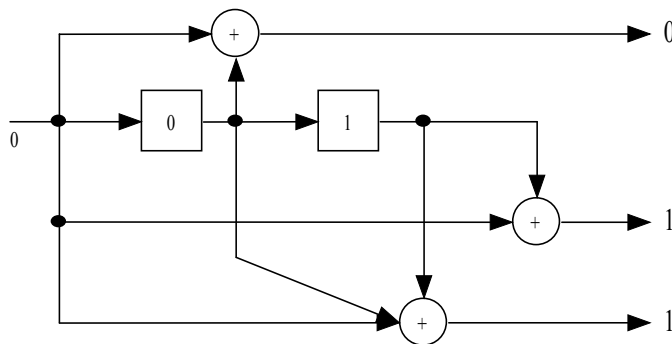
Steg 1:



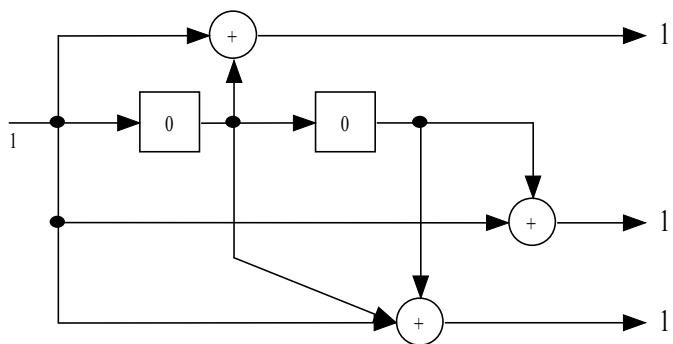
Steg 2:



Steg 3:

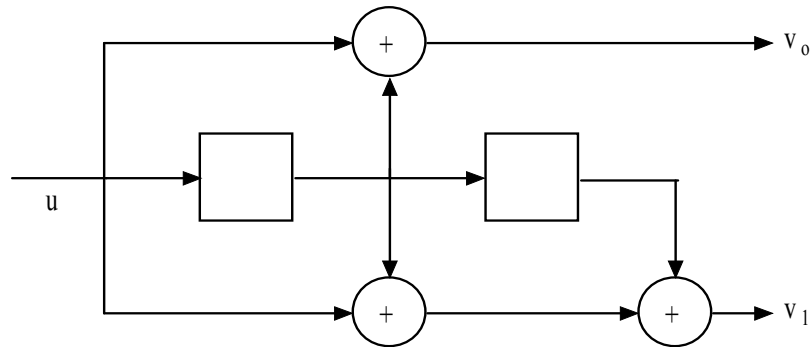


Steg 4:



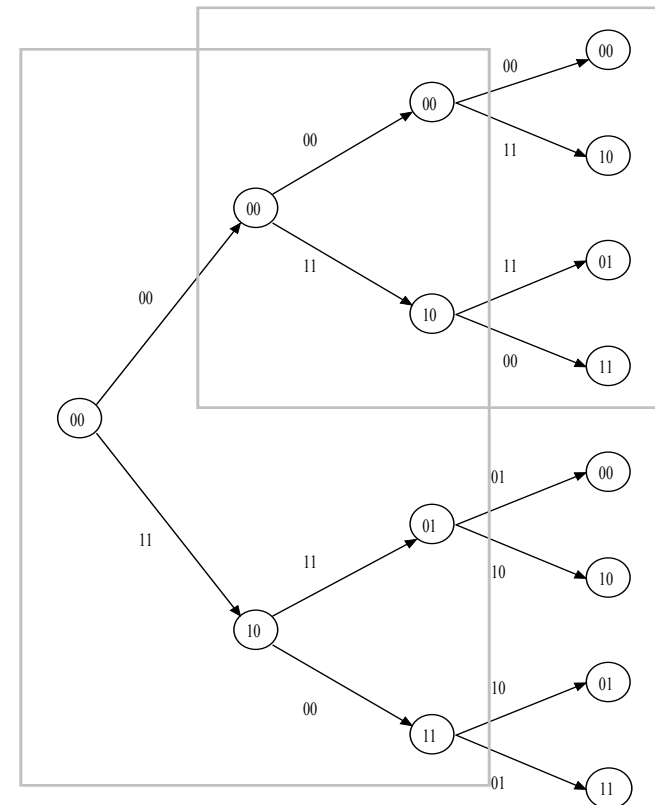
*Tilstanden* til enkoderen tilsvareer innholdet i minneblokkene.

# Kodetre



Enkoder for en (2,1,2)-kode

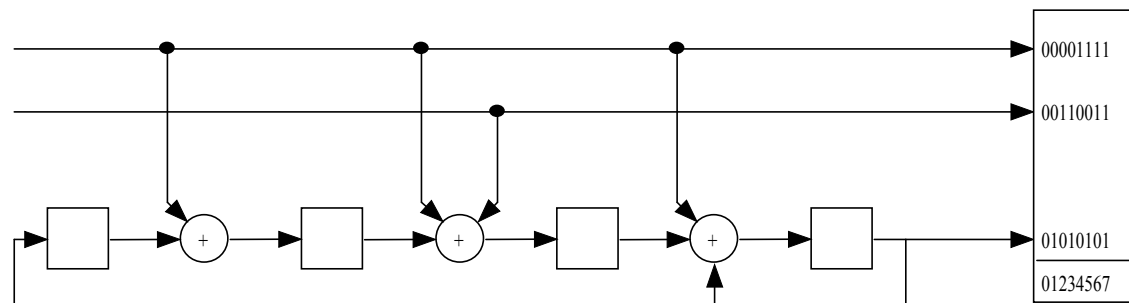
- Enkoderen til (2,1,2)-koden danner kodetreet til høyre
- Antall noder øker eksponentielt med lengden informasjonssekvensen
- Treet inneholder en del redundant informasjon
- Innholdet i de to boksene i kodetreet er det samme



Kodetre generert av enkoderen

# ”Feedback” enkodere

- Enkoderene presentert til nå kalles ”feedforward” enkodere
- Finnes også ”feedback” enkodere som har en tilbakeføringskanal mellom minneblokkene



- I ”feedforward” enkodere legges det ofte til  $m$  antall 0’ere i informasjonssekvensen
  - Dette fører enkoderen tilbake i nulltilstanden. Disse bitene kalles halen i treet
  - Halen sikrer feilkorreksjon i de siste bitene
- I ”feedback” enkodere vil tilbakeføringskanalen gjøre at biter sirkulerer i enkoderen
  - Kan ikke føres tilbake i nulltilstand på en enkel måte
- Det er vist at man ved å legge til en hale med lengde lik registerlengden sikrer feilkorreksjon i de siste bitene for ”feedback” enkodere



# MLD og sekvensiell dekodning

- Koder med store registerlengder gir generelt bedre feilkorreksjon enn koder med små registerlengder
- MLD krever mange beregninger og det er derfor begrenset hvor store registerlengder som er mulig å benytte
- Antall beregninger ved MLD er de samme uansett hvor mange feil som oppstår
- Sekvensiell dekodning er uavhengig av registerlengden. Denne typen dekodning tilpasser seg støynivået
  - Lite støy gir få beregninger
  - Mye støy gir mange beregninger

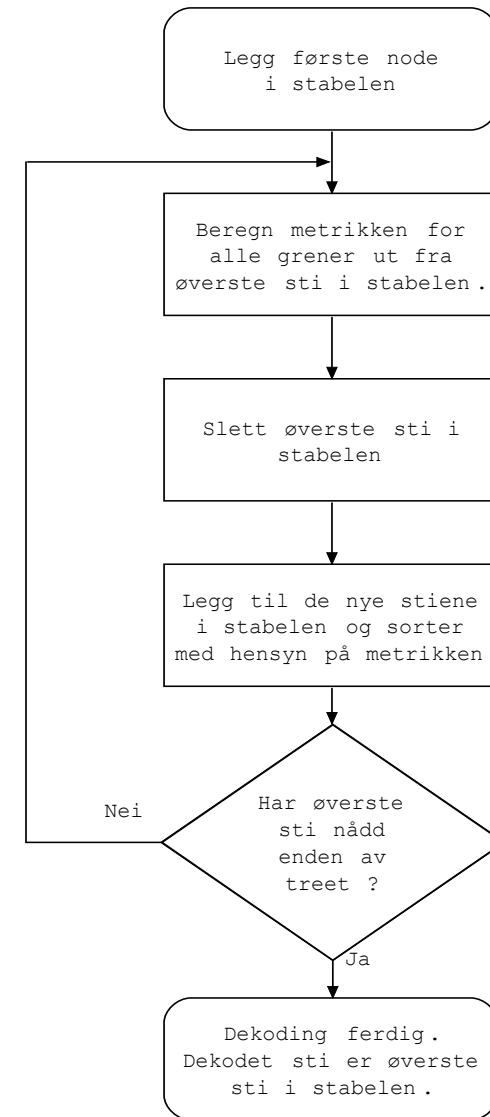


# Sekvensielle dekodingsalgoritmer

- To mest kjente algoritmer for sekvensiell dekoding:
  - Stabelalgoritmen
  - Fanoalgoritmen
- Jeg har konsentrert meg om stabelalgoritmen da denne er enklest å implementere
- Algoritmene benytter en *metrikk* til å sammenligne ulike stier. Metrikken gir et tall på hvor nær en sti ligger en annen sti
- Fanometrikken er en metrikkberegning som tar hensyn til lengden på stien og derfor egner seg til bruk i sekvensiell dekoding

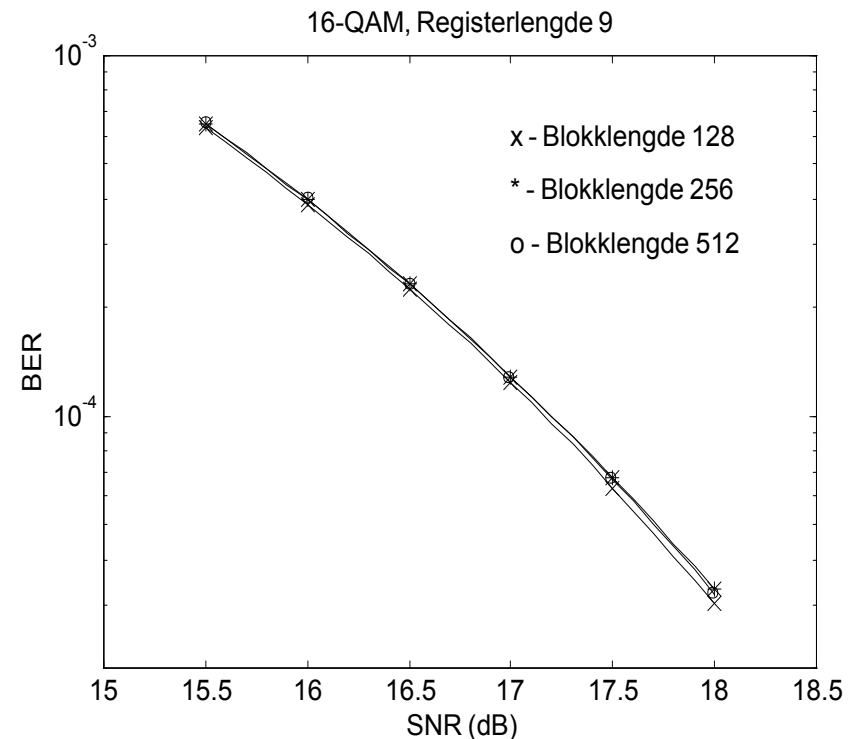
# Stabelalgoritmen

- Benytter en stabel til å organisere de ulike grenene i et kodetre
- Hver node i stabelen inneholder selve stien og metrikken til denne
- Stabelen er hele tiden sortert etter beste metrikk



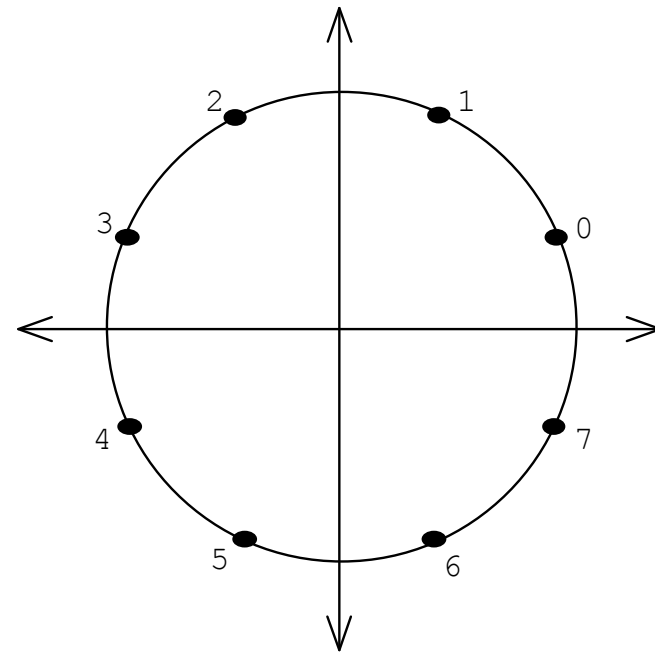
# Resultater

- Tidsbruk
  - Simuleringer er en tidkrevende prosess
  - Størrelsen på signalkonstellasjonen har størst innvirkning
  - Blokk lengden har også betydning for kjøretiden
    - Simuleringer med tre ulike blokk lengder
    - Se om dette hadde innvirkning på bitfeilraten (BER)
  - Resultatet viser at dette spiller liten rolle å variere blokk lengden
    - Benyttet en hale med lengde lik registerlengden

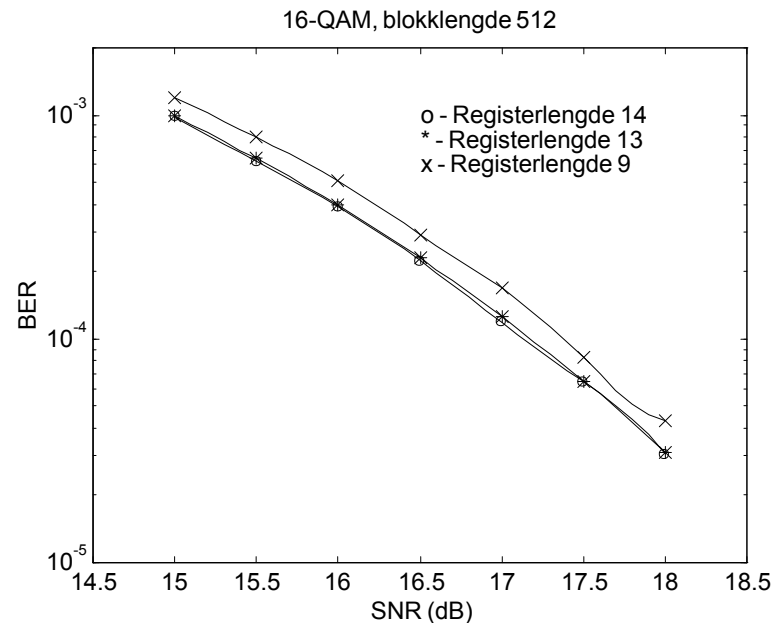
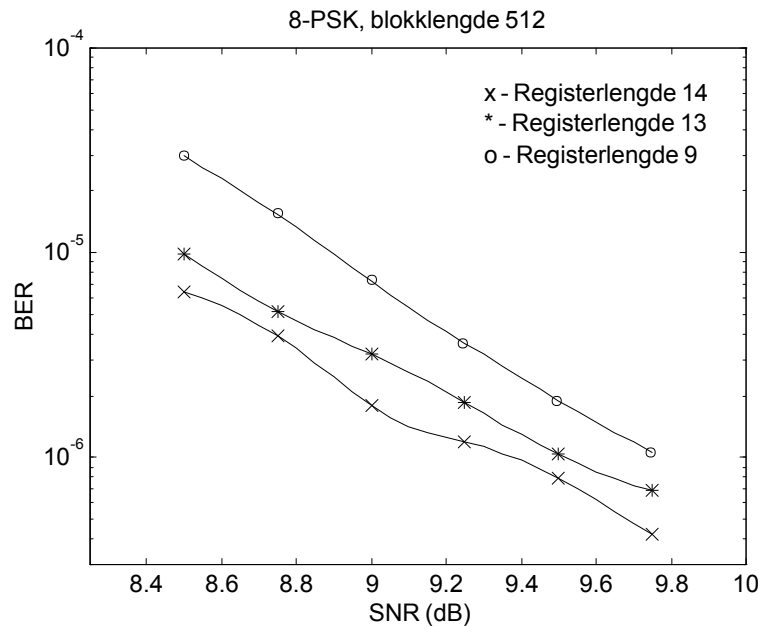


# Simuleringer for 8-PSK og 16-QAM

- Figuren til høyre viser en 8-PSK konstellasjon
- Kodene jeg har brukt i mine simuleringer er funnet av Wang og Costello
  - god feilkorleksjon
- Wang og Costello har publisert simuleringresultater for 8-PSK der Fanoalgoritmen er benyttet
- For 8-PSK trengs et SNR på 7,6 dB for å nå  $R_0^*$
- For 16-QAM treng et SNR på 11,1 dB for å nå  $R_0^*$



# Simuleringer for 8-PSK og 16-QAM



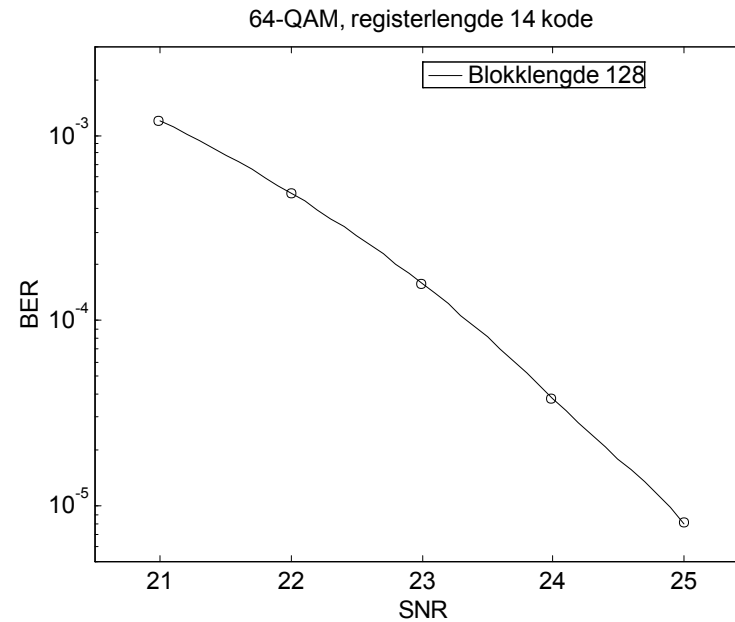
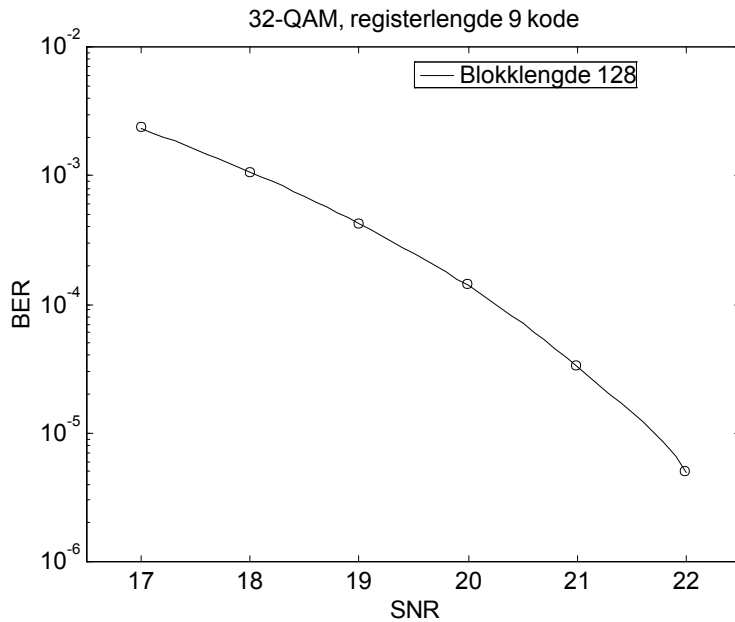
- Resultatene ligger ca. 1 dB dårligere enn resultatene til Wang og Costello for 8-PSK
- Ser at man ved å øke registerlengden oppnår bedre resultater selv om forskjellen er større for 8-PSK enn for 16-QAM



## Tilfeldig genererte koder for 32-QAM og 64-QAM

- Teorien sier at man ved å velge en tilfeldig kode med stor sannsynlighet vil finne en god kode
- Jeg har valgt ut 10 tilfeldig genererte koder for både 32-QAM og 64-QAM. Ved hjelp av simuleringer har jeg funnet den beste av disse kodene og laget et plott for dette
- Kodene jeg testet ut ga relativt like resultater noe som kan tyde på at alle kodene jeg fant var gode koder
- Legg merke til at man ved å øke størrelsen på konstellasjonen trenger et høyere SNR for å nå den samme BER

# Simuleringsresultater for 32-QAM og 64-QAM



- For en 32-QAM konstellasjon trengs det et SNR=14,5 dB for å nå cutoffraten 4 info. Biter pr. symbol
- For en 64-QAM konstellasjon trengs et SNR=17,8 for å nå cutoffraten 5 info. Biter pr. symbol



# Oppsummering

- Har gitt et lite innblikk i digital kommunikasjon. Jeg har forklart hvordan støy påvirker signalpunkter i en signalkonstellasjon og hvordan koding kan sikre pålitelig kommunikasjon
- Jeg har gitt en forklaring på sekvensiell dekoding med vekt på stabelalgoritmen
- Videre har jeg presentert simuleringsresultater fra mitt simuleringsverktøy
- Fått verifisert flere teorier om sekvensiell dekoding