Helsinki University of Technology
Laboratory of Telecommunications Technology

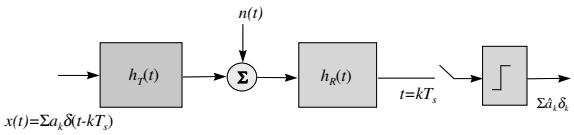
S-38.211 Signaalinkäsittely tietoliikenteessä I
Signal Processing in Communications (2 ov)

Syksy 1998

3. Luento: Optimaalinen ML-vastaanotto ja Viterbi-algoritmi

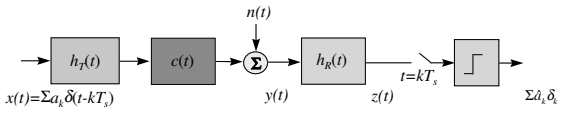
prof. Timo Laakso
Vastaanotto torstaisin klo 10-11
Huone G210, puh. 451 2473
Sähköposti: timo.laakso@hut.fi

Kertausta: AWGN-kanava



- Edellisellä luennolla tarkasteltiin siirtojärjestelmää AWGN-kanavassa
- *Yksittäiselle pulssille* johdettiin SNR:n maksimoiva vastaanottosuodin. Tämä on *optimaalinen vastaanotin* siinä mielessä että se minimoi bittivirhetodennäköisyyden (kun ainoa häiriö kanavassa on AWGN-kohina!)
- Jos lähetys- ja vastaanottosuodatin yhdessä täyttävät Nyquistin kriteerin, järjestelmä toimii myös jatkuvassa siirrossa (ISI = 0)

Kertausta: Lineaarinen kanava

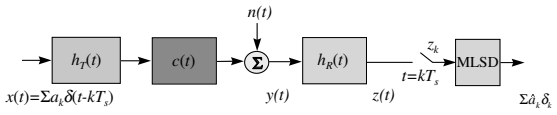


- MUTTA: jos kanavassa on *lineaarista vääristymää* (esim. taajuusriippuvaa vaimennusta), *lineaarisella* suodatuksella ei pystytä poistamaan ISIä ja vaimentamaan kohinaa samanaikaisesti niin että BER minimoituu (ISI on *epälineaarinen* datariippuva häiriö!)
- Tarvitaan *epälineaarisia* vastaanottomenetelmiä!

Optimaalinen ML-vastaanotto

- Kun ISIä ei ole, riittää kun tarkastellaan yhtä symbolia kerrallaan
- ISI:n tapauksessa yksi lähetetty pulssi häiritsee viereisten pulssien vastaanottoa => kannattaa tarkastella usempaa kuin yhtä symbolia kerrallaan (*sekvenssin* vastaanotto)
- Voidaan johtaa *maximum likelihood* (ML) -tyyppinen optimaalinen sekvenssin vastaanottoalgoritmi joka minimoi sekvenssin *a priori* (etukäteis-) virhetodennäköisyyden
- *Viterbi-algoritmi* on ML-sekvenssi-ilmaisimen tehokas approksimatiivinen toteutus

Optimaalinen PAM-signaalin ML-vastaanotto



- Tarkastellaan kuvan (kantataajuista) PAM-järjestelmää
- Lähetys- ja vastaanottosuodin voivat olla sovitettu suodinpari jotka yhdessä täyttävät Nyquistin kriteerin (EI välttämätöntä!)
- Kanavassa on lineaarista vääristymää ja siten aina ISIä!
- Tarkastellaan *K*:n symbolin mittaisen sekvenssin ML-ilmaisua (Maximum Likelihood Sequence Detection, MLSD)

PAM-signaalin ML-vastaanotto

◆Vastaanotettu signaali on muotoa:

$$y(t) = \sum_{k=1}^{K} a_k h_{TC}(t - kT) + n(t)$$

jossa symboli a_k valitaan *M*-jäsenisestä aakkostosta, $h_{TC}(t)$ on lähetyspulssimuoto kanavan jälkeen (= $h_T(t)*c(t)$) ja $n(t)$ on *valkoista* Gaussin kohinaa.

◆Vastaanottimen esiaste koostuu sovitetusta suotimesta ja näytteenotosta:

$$z_k = y(t)*h_R(t-kT)\big|_{t=kT} = \int_{-\infty}^{\infty} y(t)h_{TC}(t-kT)dt$$

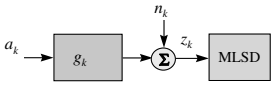
◆Sovitettu suodatin voidaan tulkita *korrelaattoriksi*!

PAM-signaalin ML-vastaanotto

- Voidaan osoittaa, että sekvenssi z_k, k=1,2,…,K sisältää *riittävän statistiikan* (sufficient statistics) optimaaliseen ilmaisuun, eli *saman informaation* kuin vastaanotettu jatkuva-aikainen signaali
- Vastaanotettua sekvenssiä voidaan siis tarkastella *K*-ulotteisena *vektorina* johon on summautunut *K*-ulotteinen kohinavektori

PAM-signaalin ML-vastaanotto

- Diskreettiaikainen malli siirtojärjestelmälle symbolitaajuudella:



$$z_k = a_k * g_k + n_k = \sum_{m=-\infty}^{\infty} a_k g_{m-k} + n_k$$

- Tässä g_k on *diskreettiaikainen vastine koko suodatusketjulle* eli lähetyssuotimelle, kanavalle ja vastaanottimelle!
- Voidaan myös osoittaa, että sekvenssiin summautunut *diskreettiaikainen* kohina on normaalijakautunutta

PAM-signaalin ML-vastaanotto

- Optimaalisen ML-sekvenssi-ilmaisun toteutus:
  1) Generoi kaikki mahdolliset lähetyssekvenssit a_k
  2) Generoi vastaavat ulostulosekvenssit $s_k = a_k * g_k$ diskreetin mallin avulla
  3) Vertaa s_k:ta vastaanoton päätösmuuttujasekvensseihin z_k ja valitse se lähetyssekvenssi a_k joka on 'lähin'
- Vertailukriteeri: *euklidinen (neliöllinen) etäisyys:*

$$\min_{\{a_k, k=1,2,\ldots,K\}} \sum_{k=1}^{K} |z_k - s_k|^2 = \min_{\{a_k, k=1,2,\ldots,K\}} \sum_{k=1}^{K} |z_k - a_k * g_k|^2$$

PAM-signaalin ML-vastaanotto

Toteutuksessa useita käytännön ongelmia:

1) Kanava tunnettava

Ratkaisu: yleensä kanava estimoidaan erikseen

2) Kohina voi olla alunperin *värillistä*

Ratkaisu: lisätään valkaisusuodatus (=> valkaistu sovitettu suodatin, Whitened Matched Filter, WMF)

3) Kun käytössä M-tasoinen aakkosto, mahdollisia sekvenssejä on M^K kappaletta - yleensä tolkuton määrä

(esim. M=2, K=10 => M^K= 1024)

Ratkaisu: **Viterbi-algoritmi**!

Markov-ketjuista

- Markov-ketjuja käytetään mallintamaan *tilakoneita* (finite state machines) joiden tulosignaali on satunnainen.
- Seuraavassa niitä käytetään mallintamaan symbolien *keskinäisvaikutusta*.
- Markov-ketju $\{\psi_k\}$ on diskreetti*aikainen* ja *-arvoinen* satunnaisprosessi jonka tila $\{\psi_k\}$ toteuttaa ehdon

$$p\left(\Psi_{k+1} \middle| \Psi_k, \Psi_{k-1}, \ldots\right) = p\left(\Psi_{k+1} \middle| \Psi_k\right)$$

- *Tilan todennäköisyys* riippuu siis vain *edellisestä* tilasta (ja mahd. satunnaistekijästä, mutta ei aiemmista tiloista).

…Markov-ketjuista

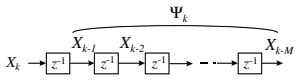
- Markov-ketju on *homogeeninen* jos siirtymätodennäköisyys $p(\psi_k|\psi_{k+1})$ ei riipu k:sta (vastaa stationäärisyyttä tai aikainvarianssia).
- Homogeenisen Markov-ketjun ominaisuudet määräytyvät *tilanmuutostodennäköisyyksistä*:

$$p\left(j|i\right) = p_{\Psi_{k+1}|\Psi_k}\left(j|i\right)$$

(Vasen puoli *lyhennetty* merkintä.)

Esimerkki Markov-ketjusta

- *Siirtorekisteriprosessi* (LM **Kuva 3-7**):



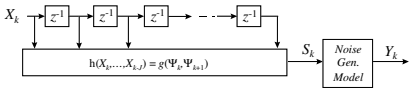
- Tässä X_k on *diskreettiarvoinen* satunnaismuuttuja joka on riippumaton edellisistä arvoista $X_{k-1},\ldots,X_{k-M-1}$. Tilaksi määritellään

$$\Psi_k = \{X_{k-1},\ldots, X_{k-M}\}$$

- Järjestelmä toteuttaa Markov-ehdon. Kyseessä on *vektoriarvoinen* Markov-ketju.

Symbolien keskinäisvaikutus

- Tarkastellaan symbolien ilmaisua kanavassa jossa on keskinäisvaikutusta. Tämä määräytyy kanavan *impulssivasteesta* (olettaen Nyquist-kriteerin toteutuvan ilman kanavaa). Oletetaan että näytetaajuus on sama kuin symbolitaajuus.
- Signaalia, jossa on symbolien keskinäisvaikutusta, voidaan mallintaa homogeenisellä Markov-ketjulla. Käytetään *siirtorekisteriprosessia* mallina (LM **Kuva 9-14**):



...Symbolien keskinäisvaikutus

- Tila on $\psi_k=\{X_{k-1},\ldots,X_{k-M}\}$ ja signaalinäyte saadaan tilanmuutosten funktiona:

$$S_k = g(\Psi_k, \Psi_{k+1})$$

missä $g(.,.)$ on *muistiton* funktio.

- Satunnaismuuttujat X_k ovat riippumattomia ja niiden jakaumat ovat identtisiä.
- Impulssivaste h_k on yleensä äärellinen kestoltaan ja niin on keskinäisvaikutuskin, eli

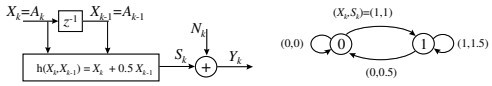
$$S_k = \sum_{i=0}^{M} h_i X_{k-i}$$

LM Esimerkki 9-25:

- Tarkastellaan kanavaa jonka impulssivaste on

$$h_k = \delta_k + 0.5\delta_{k-1}$$

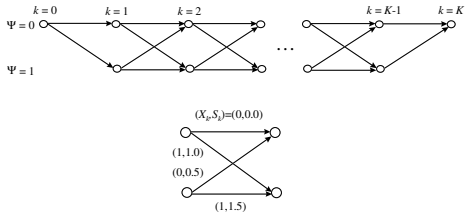
- Tätä voidaan mallintaa Markov-ketjulla (**LM Kuva 9-15**):



- Tilakaavion haaroihin on merkitty tulon A_k=X_k ja lähdön S_k arvot (ennen kohinaa N_k). Huomaa että Markov-ketjun tila(vektori) on aina *diskreettiarvoinen*, vaikka järjestelmän lähtö olisikin jatkuva-arvoinen.

Trellis-kaavio

- Toinen esitysmuoto Markov-ketjulle on *trellis-kaavio* (trellis = ristikko, säleikkö). Edellisellä esimerkille saadaan tällainen trellis-esitys (LM **Kuva 9-17**):



...Trellis-kaavio

- Kukin kaavion jakso edustaa yhtä symbolijaksoa. Kaaviossa nähdään *kaikki mahdolliset tilat* ja *tilatransitiot* eri hetkinä kaikille tulosekvensseille.
- Kaaviolla on *alkutila* $\psi_0 = 0$ ja *lopputila* $\psi_K = 0$. Alkutilan ja lopputilan välillä on joukko *polkuja*. Kukin polku edustaa yhtä *K*:n pituista symbolisekvenssiä.
- Polun haaroista voidaan lukea vastaava signaali(sekvenssi) S_k. Tämä olisi vastaanotettu, *keskinäisvaikutusta sisältävä signaali kohinattomassa* tapauksessa.

Sekvenssi-ilmaisin

Esim. jatkuva-arvoisen kanavan tapauksessa ilmaisimelle tulee signaali $Y_k = S_k + N_k$.

- *Sekvenssi-ilmaisin* pyrkii tähän havaintoon perustuen *päättämään mikä oli lähetetty symbolisekvenssi*.
- HUOM! *Kanavan malli tunnettava* (miten kohinaton signaali riippuu tilamuuttujista)
- Trelliskaaviossa sekvenssin ilmaisu vastaa *oikean polun etsimistä* alku- ja lopputilan välillä.

Metriikka

- ML-periaatteen mukaan valitaan se polku, joka on *lähinnä* havaittua sekvenssiä *tietyn metriikan mukaisesti.*
- Kun on kyse summautuvasta jatkuva-arvoisesta kohinasta, on etsittävä polku jolla on *pienin euklidinen etäisyys* (=neliövirhemitta) havaittuun sekvenssiin.
- Kullekin polulle määritellään *polun mitta* (path metric), joka on ko. *polkua vastaavien signaaliarvojen ja havaitun sekvenssin välinen euklidinen etäisyys.*

…Metriikka

- Kukin trellis-kaavion jakso vastaa yhtä symbolijaksoa. Tämän jakson havaintonäyte on Y_k. Kaavion eri haaroista nähdään kaikki mahdolliset signaali näytearvot symbolijaksolla k.
- Kullekin trellis-kaavion haaralle määritellään *haaran mitta* (branch metric):
  - Jatkuva-arvoinen Gaussin kanava: euklidinen eli neliöllinen etäisyys $|Y_k\text{-}S_k|^2$
  - Binäärinen symmetrinen kanava: Hammingin etäisyys (bittivirheiden lukumäärä) $d_H(Y_k,S_k)$
- Polun mitta saadaan summaamalla haarojen mitat

Sekvenssi-ilmaisimen toteuttaminen

- ML-sekvenssi-ilmaisin voitaisiin toteuttaa seuraavasti:
  1) Lasketaan kaikkien *haarojen* mitat
  2) Lasketaan kaikkien mahd. *polkujen* (= *haarakombinaatioiden*) mitat ja etsitään pienin
  3) Pienintä mittaa vastaavan *polun tulosymbolit* muodostavat *ilmaistun sekvenssin*

Ongelma: Laskenta kasvaa *eksponentiaalisesti* sekvenssin pituuden mukana!

Viterbi-algoritmi

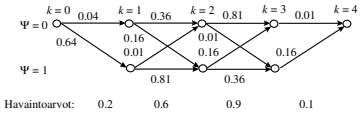
- Viterbi-algoritmi on *dynaamisen ohjelmoinnin* periaatteen soveltamista polun mitan minimointiongelman tehokkaaseen ratkaisemiseen.
- Periaate:
  1) Trelliskaavio käydään läpi *jakso kerrallaan* alkusolmusta loppusolmuun
  2) Kussakin vaiheessa tiedetään *lyhimmät* (= pienimmän mitan) polut alkusolmusta ko. jakson kaikkiin *tulosolmuihin*.
  3) Seuraavaksi etsitään lyhimmät polut ko. jakson *lähtösolmuihin*. Nämä löydetään käymällä läpi kaikki haarautumisvaihtoehdot.

...Viterbi-algoritmi

- Viterbi-algoritmin olennainen idea on *polkujen karsiminen* seuraavaan sääntöön perustuen: jos lyhin polku alkusolmusta loppusolmuun kulkee solmun x kautta, niin tämän polun mitta on kahden lyhimmmän osapolun (alkusolmu-x ja x-loppusolmu) mittojen summa.
- **=> N tilan (solmun) järjestelmässä riittää N polun ja niiden mittojen tallennus globaalin minimin löytämiseen!**

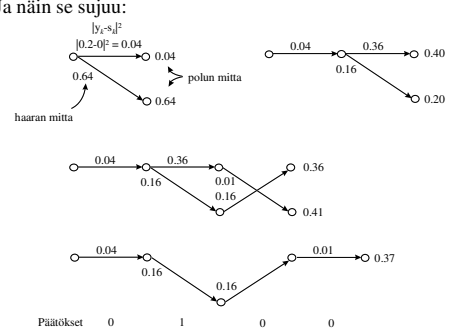
Viterbi-esimerkki

- Tarkastellaan **Esimerkin 9-25** tilannetta jossa kanava on $h_k = \delta_k + 0.5\delta_{k-1}$ ja mukana on summautuvaa Gaussin kohinaa. Vastaanotettu sekvenssi olkoon {0.2, 0.6, 0.9, 0.1}. Kaaviosta nähdään haarojen mitat ja alkupolkujen mitat kussakin vaiheessa.



Viterbi-esimerkki

- Ja näin se sujuu:



Viterbi-algoritmin ominaisuuksia

- Viterbi-algoritmin kullakin jaksolla (alkua ja loppua lukuunottamatta) tehdään tietty laskentatyö. Siten *laskennan määrä on verrannollinen sekvenssin pituuteen K.*
- Esitetyssä muodossa Viterbi-algoritmi antaa tuloksena ilmaistun sekvenssin vastakun koko sekvenssi on käyty läpi. Tästä aiheutuu *haitallista viivettä.*
- Osittaisratkaisu voi löytyä jo aiemmin. Esim. ensimmäinen symboli selvisi jo toisen jakson aikana, koska tällöin kaikki alkupolut kulkivat saman haaran kautta.
- Viterbi-algoritmissa käy yleensä näin. Pitkää sekvenssiä ilmaistaessa *alkupolut yleensä yhtyvät*. Tuntuisi järkevältä päättää alkupään symboleista tietyn varoajan jälkeen.

Katkaisusyvyys

- Käytännössä tehdään niin, että jaksoa *k* käsiteltäessä jakson *k-d* kohdalla karsitaan pidemmät polut ja ilmaistaan vastaava symboli. Kun *katkaisusyvyys d* valitaan riittävän suureksi, tämä ei juuri huononna lopputulosta.
- Tyypillinen valinta katkaisusyvyydelle on n. 5 kertaa kanavan impulssivasteen pituus

Viterbi-algoritmin käyttö

- Viterbi-algoritmi sopii kaikkiin ML-ilmaisuongelmiin joissa
  - signaalinmuokkausta voidaan mallintaa homogeenisena Markov-ketjuna
  - kohinakomponentit ovat riippumattomia (=valkoista)
- Tärkeitä erikoistapauksia ovat
  - ilmaisu Gaussin kanavassa jossa on lisäksi keskinäisvaikutusta
  - eräiden koodausmenetelmien dekoodaus (mm. konvoluutio- ja trelliskoodaus) binäärisessä kanavassa
  - purskeinen siirto häipyvässä radiokanavassa (esim. GSM)