



S-38.211 Signaalinkäsittely tietoliikenteessä I

Signal Processing in Communications (2 ov)

Syksy 1997

7. Luento: Adaptiiviset korjaimet II

prof. Timo Laakso

Vastaanotto torstaisin klo 10-11

Huone G210, puh. 451 2473

Sähköposti: timo.laakso@hut.fi

Stokastinen gradienttialgoritmi (LM11.2)



- ◆ Edellisessä luvussa päästiin FIR-suotimen adaptointiin MSE-gradienttialgoritmeilla (MSEG) joka korjaa askelparametrin määräämän verran kerroinvektoria kohti optimiratkaisua.
- ◆ Valitettavasti MSEG perustuu myös siihen, että kanavan ominaisuudet eli autokorrelaatiomatriisi ja ristikorrelaatiovektori tunnetaan.
- ◆ Näin ei yleensä ole, eli käytännön tuntemattomissa ja muuttuvissa kanavissa tarvitaan vielä lisää yksinkertaistuksia.



- ◆ Lopullinen algoritmi saadaan ottamalla käyttöön *stokastinen* ('kohinainen') *gradienttiestimaatti* tarkan asemesta.
- ◆ Stokastiseen gradienttiin perustuva adaptointi tunnetaan yleensä *LMS(Least-Mean Square)-algoritmina*, vaikka nimi onkin epätarkka (MSE-virhe minimoituu vain approksimatiivisesti). Näin saadaan kuitenkin käyttökelpoinen ja ehkä maailman eniten käytetty adaptiivinen suodinalgoritmi.

Gradientti ennen odotusarvoa



- ◆ MSEG-gradientti johdettiin ottamalla neliövirheestä *ensin* odotusarvo ja *sitten* gradientti:

$$\begin{aligned} E[|e_k|^2] &= E[|a_k|^2] - 2 \operatorname{Re}\{\mathbf{c}^H E[a_k \mathbf{r}_k^*]\} + \mathbf{c}^H E[\mathbf{r}_k^* \mathbf{r}_k^T] \mathbf{c} \\ &= E[|a_k|^2] - 2 \operatorname{Re}\{\mathbf{c}^H \mathbf{a}\} + \mathbf{c}^H \Phi \mathbf{c} \end{aligned} \quad (11.8)$$

- ◆ Tarkastellaan nyt ensin neliövirheen gradienttia *ilman* odotusarvoa. Koska gradientti ja odotusarvo ovat lineaarisia operaattoreita, niiden järjestystä voidaan vaihtaa:

$$|e_k|^2 = |a_k|^2 - 2 \operatorname{Re}\{a_k \mathbf{c}^H \mathbf{r}_k^*\} + \mathbf{c}^H \mathbf{r}_k^* \mathbf{r}_k^T \mathbf{c} \quad (11.49)$$



- ◆ Tälle saadaan gradientiksi (ks. **Harj. 11-5** edellä)

$$\nabla_{\mathbf{c}} |e_k|^2 = -2\mathbf{r}_k^* (a_k - \mathbf{r}_k^T \mathbf{c}) = -2e_k \mathbf{r}_k^* \quad (11.50)$$

- ◆ Tästä voitaisiin nyt ottaa odotusarvo ja saada tuttu tulos

$$\nabla_{\mathbf{c}_j} E[|e_k|^2] = 2\Phi \mathbf{c} - 2\mathbf{a} \quad (11.25)$$

- ◆ mutta (11.50) onkin itse asiassa toteutuksen kannalta parempi sellaisenaan. Voidaan osoittaa, että se antaa *harhattoman* (unbiased) estimaatin todelliselle gradientille. Estimaatti on *stokastinen* eli *kohinainen*.

Stokastinen gradienttialgoritmi (SG)



- ◆ SG-algoritmin mukainen kertoimien adaptiivinen päivitys on muotoa

$$\mathbf{c}_{k+1} = \mathbf{c}_k - \frac{\beta}{2} \nabla_{\mathbf{c}} E[|e_k|^2]_{\mathbf{c}=\mathbf{c}_k} \quad (11.52)$$

- ◆ ja sijoittamalla tähän (11.50) saadaan *SG-algoritmi* muotoon

$$\mathbf{c}_{k+1} = \mathbf{c}_k + \beta e_k \mathbf{r}_k^* = [\mathbf{I} - \beta \mathbf{r}_k^* \mathbf{r}_k^T] \mathbf{c}_k + \beta a_k \mathbf{r}_k^* \quad (11.53)$$

- ◆ Tätä voidaan nyt verrata *MSEG-algoritmiin*

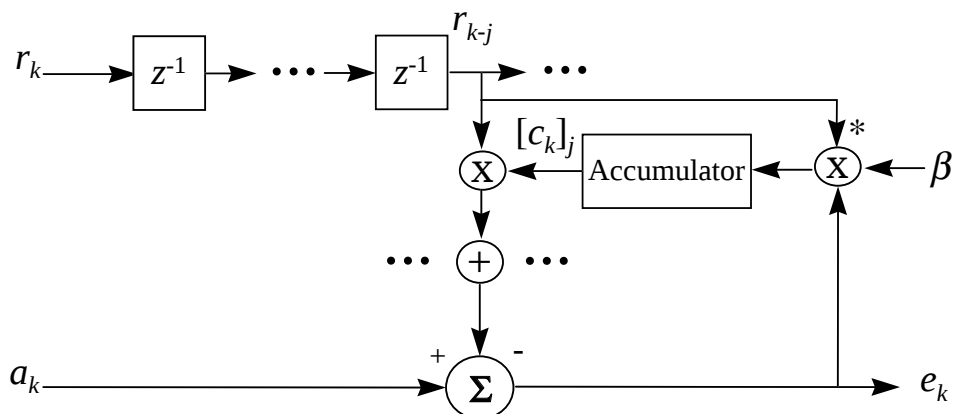
$$\mathbf{c}_{j+1} = \mathbf{c}_j + \beta (\mathbf{a} - \Phi \mathbf{c}_j) = (\mathbf{I} - \beta \Phi) \mathbf{c}_j + \beta \mathbf{a} \quad (11.28)$$



SG-algoritmista tehdään seuraavat havainnot:

- ◆ 1) SG-algoritmi sisältää (valittavaa askelparametria β lukuunottamatta) *vain havaintoarvoja ja muita tunnettuja suureita*, eli on *suoraan laskettavissa* ilman muuta estimointia
- ◆ 2) Kun MSEG-algoritmissa iteraatioindeksi j oli aikaindeksistä k riippumaton, SG-algoritmissa $j = k$. Gradientin estimointi toteutetaan siis *aikakeskiarvoistamalla* tulevaa dataa.

SG-algoritmin laskentatoteutus



- ◆ SG-algoritmin laskenta yhtä päivitettävää kerrointa kohden on esitetty **Kuvassa 11-7**.



- ◆ Adaptaatiokaava (11.53) voidaan kirjoittaa komponentti-muodossa

$$[\mathbf{c}_{k+1}]_j = [\mathbf{c}_k]_j + \beta e_k r_{k-j}^*, \quad -L \leq j \leq L \quad (11.54)$$

- ◆ eli laskenta etenee modulaarisesti komponentti kerrallaan. Uusi virhetermi lasketaan kierroksen päätteeksi seuraavaa päivitystä varten.

SG-algoritmin suppeneminen (LM11.2.1)



- ◆ Perusero MSEG- ja SG-algoritmissa on että *MSEG on deterministinen*, eli (kunhan auto- ja ristikorrelaatiodata tunnetaan) suppeneminen etenee ennaltamäärättyä polkua pitkin. Sen sijaan SG-algoritmin suppenemisessä on aina stokastista variaatiota, eli joillain iteraatiokerralla tulos voi huonotakin. Suppeneminen on *keskimääräistä*.
- ◆ SG-algoritmin suppenemistä voidaan tarkastella ottamalla odotusarvo (11.53):sta:

$$\mathbf{c}_{k+1} = E \left[\left[\mathbf{I} - \beta \mathbf{r}_k^* \mathbf{r}_k^T \right] \mathbf{c}_k + \beta a_k \mathbf{r}_k^* \right] = (\mathbf{I} - \beta \Phi) \mathbf{c}_k + \beta \mathbf{a} \quad (11.57)$$



- ◆ Ottamalla odotusarvo vielä *kerroinprosessin* suhteen saadaan

$$E[\mathbf{c}_{k+1}] \approx (\mathbf{I} - \beta\Phi)E[\mathbf{c}_k] + \beta\mathbf{a} \quad (11.58)$$

- ◆ SG-algoritmin suppenemispolku on siis *keskimäärin* sama kuin MSEG-algoritmilla. Tästä seuraa, että myös *suppenemisehdot* (askelparametrin valinta) ovat pääpiirteissään samat molemmille algoritmeille.

...SG-algoritmin suppeneminen



- ◆ Suppenemisen tarkemman analyysin helpottamiseksi määritellään vanhaan tapaan virhevektori

$$\mathbf{q}_k = \mathbf{c}_k - \mathbf{c}_{opt} \quad (11.59)$$

- ◆ **(Harj. 11-10.)** Johda SG-algoritmin päivityskaava muotoon

$$\mathbf{q}_{k+1} = \Gamma_k \mathbf{q}_k + \beta d_k \mathbf{r}_k^* \quad (11.60)$$

missä

$$\Gamma_k = \mathbf{I} - \beta \mathbf{r}_k^* \mathbf{r}_k^T \quad (11.61)$$

ja d_k on virhesignaali optimisuotimelle

$$d_k = a_k - \mathbf{r}_k^T \mathbf{c}_{opt} \quad (11.62)$$



- ◆ Koska virhesignaali d_k on nolla vain keskimäärin, SG-algoritmi (11.60) ei koskaan suppene tarkasti kohti optimiratkaisua vaan jää keikkumaan sen ympärille. Ratkaisu on vain 'keskimäärin' optimaalinen!
- ◆ SG-algoritmin hyvyttä voidaankin arvioida *kerroinvektorin* Euklidisen (neliöllisen) *virhemitan* avulla

$$\|\mathbf{q}_k\|^2 = \mathbf{q}_k^H \mathbf{q}_k \quad (11.63)$$

joka kertoo kuinka lähellä kertoimet ovat optimia.

SG-algoritmin jäännösvirhe



- ◆ Viime kädessä kiinnostaa kuitenkin mikä on suodatetun ja halutun signaalin ero (virhesignaali e_k). Tämän neliövirhe ja kerroinvirhe (11.63) ovat yhteydessä seuraavasti kuten edellä (11.15) johdettiin:

$$E[|e_k|^2] = \xi_{\min} + \mathbf{q}_k^H \Phi \mathbf{q}_k \quad (11.65)$$

- ◆ Keskiarvoistamalla tämä myös kerroinvektorien suhteen saadaan

$$E[|e_k|^2] = \xi_{\min} + E[\mathbf{q}_k^H \Phi \mathbf{q}_k] \quad (11.66)$$

- ◆ Jälkimmäistä termiä sanotaan *jäännösvirheeksi* (excess MSE) ja se kertoo kuinka paljon kertoimien stokastisesta vaihtelusta aiheutuu lisävirhettä.



- ◆ Tarkastellaan jäännösvirheen lauseketta siinä erikoistapauksessa että input-prosessi r_k on korreloimatonta (valkoista kohinaa), eli

$$\Phi = \Phi_0 \mathbf{I}, \quad \Phi_0 = E[|r_k|^2] \quad (11.67)$$

- ◆ Tällöin keskimääräinen MSE saadaan muotoon

$$E[|e_k|^2] = \xi_{\min} + \Phi_0 \|\mathbf{q}_k\|^2 \quad (11.68)$$

- ◆ Voidaan johtaa (ks. kirjan **Appendix 11-A**) tulos:

$$E[\|\mathbf{q}_{k+1}\|^2] = \gamma E[\|\mathbf{q}_k\|^2] + \beta^2 N \Phi_0 \xi_{\min} \quad (11.69)$$

missä $\gamma = 1 - 2\beta\Phi_0 + \beta^2 N \Phi_0^2$

...SG-algoritmin suppenemisehto



- ◆ Valkoisen kohinan tapauksessa on vain yksi adaptaatiomoodi (ominaisarvot ovat samat). Kaavasta (11.69) näkyy, kuinka jäännösvirhe riippuu askelparametrasta β .
- ◆ Tuloksesta voidaan myös johtaa ehto algoritmin suppenemiselle, eli

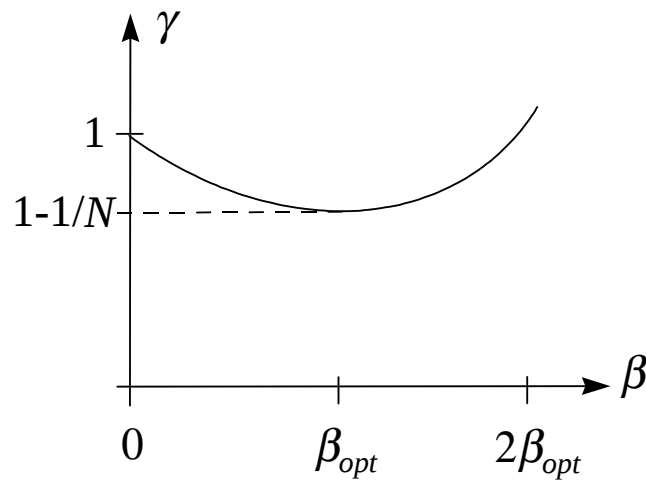
$$|\gamma| = |1 - 2\beta\Phi_0 + \beta^2 N \Phi_0^2| < 1 \quad (11.70)$$

- ◆ Nopeimpaan suppenemiseen johtaa pienin γ :n arvo, josta saadan optimi- β :ksi

$$\beta_{opt} = \frac{1}{N \Phi_0} \quad (11.71)$$



- ◆ Suppenemisehtoa (11.71) havainnollistaa **Kuva 11-8**:



- ◆ SG-algoritmin stabiilisuusehdoksi saadaan ($-1 < \gamma < 1$):

$$0 < \beta < \frac{2}{N\Phi_0} = 2\beta_{opt} \quad (11.72)$$

- ◆ Tätä voidaan verrata MSEG-algoritmin stabiilisuusehtoon

$$0 < \beta < \frac{2}{\lambda_{max}} \quad (11.39)$$

- ◆ SG-algoritmin stabiilisuusehto on usein paljon tiukempi - huomaa että se on kääntäen verrannollinen suotimen pituuteen N . Algoritmin stokastinen luonne vaatii siis ylimääräistä pelivaraa.



Askelparametrin normalisointi

- ◆ Epästationäärisessä ympäristössä signaalin teho voi vaihdella paljonkin, mikä vaikuttaa suoraan algoritmin suppenemisnopeuteen jos askelparametri β pidetään vakiona. Tämä voidaan helposti eliminoida *normalisoimalla* β seuraavasti

$$\beta_k = \frac{a}{\sigma_k^2 + b} \quad (11.77)$$

missä a ja b ovat sopivia parametreja ja σ_k^2 on input-signaalin tehon estimaatti aikaindeksillä k . Tehon estimointiin voidaan käyttää *eksponentiaalisesti painotettua aikakeskiarvoa*

$$\sigma_k^2 = (1 - \alpha) \sum_{j=0}^{\infty} \alpha^j |r_{k-j}|^2 \quad (11.78)$$

...SG-algoritmin modifikaatioita



- ◆ Kaavassa (11.78) tietysti $0 < \alpha < 1$. Tällöin tehoestimaatille saadaan näppärä rekursiokaava

$$\sigma_k^2 = \alpha \sigma_{k-1}^2 + (1 - \alpha) |r_k|^2 \quad (11.79)$$

Kaksimoodialgoritmit (Gear-Shift Algorithms)

- ◆ Johtotransmissio (esim. äänitaajuusmodeemit) kanava muuttuu vain vähän yhteyden aikana, jolloin alkuestimoinnin jälkeen askelparametria voidaan pienentää.

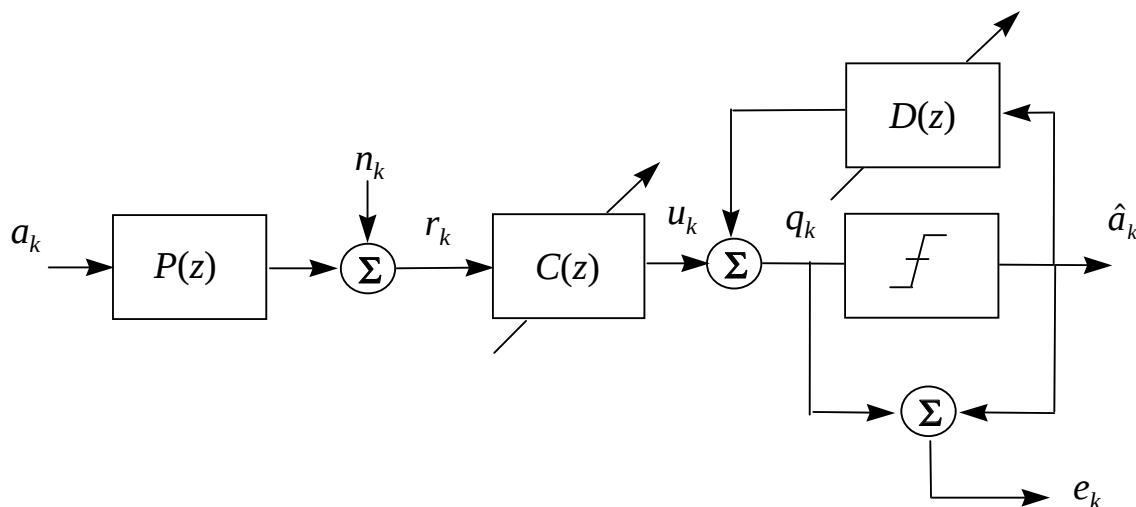


- ◆ Edellä saatiin lineaarisen FIR-korjaimen adaptointiin kehitettyä lopultakin käyttökelpoinen menetelmä: *stokastinen gradientti(SG- eli LMS-)algoritmi*.
- ◆ Seuraavaksi kehitellään vastaava *algoritmi päätöstakaisin-kytketylle eli DFE-korjaimelle*, jossa on lineaarisen korjaimen lisäksi osuus joka vähentää ISIä edellisiin päätöksiin perustuen. Osoittautuu, että lineaarisen FIR-korjaimen SG-algoritmia voidaan käyttää pienin muutoksin lähes sellaisenaan DFE-korjaimen adaptointiin.

Adaptiivinen DFE (LM11.3)



- ◆ DFE-korjaimen rakenne oli jo esillä aiemmin **Kuvassa 11-2**:





- ◆ DFE-korjaimessa on siis lineaarinen FIR-osa (*prekursori*) joka suodattaa tulevia näytteitä kuten pelkkä LE.
- ◆ Lisäksi rekursiivinen osa (*postkursori*) poistaa ISIä vähentämällä vanhoja symboliarvoja (estimaatteja) sopivilla painokertoimilla. Postkursori vähentää siis *vain* ISIä eikä kohinaa. Se vaikuttaa kuitenkin koko korjaimen kohinavahvistukseen helpottamalla prekursorin toimintaa.
- ◆ Käytettäessä MSE-virheen minimointikriteeriä postkursorin lisääminen vaikuttaa siis myös prekursorin optimaalisiin kertoimiin. Adaptointi on siis tehtävä yhteisesti.

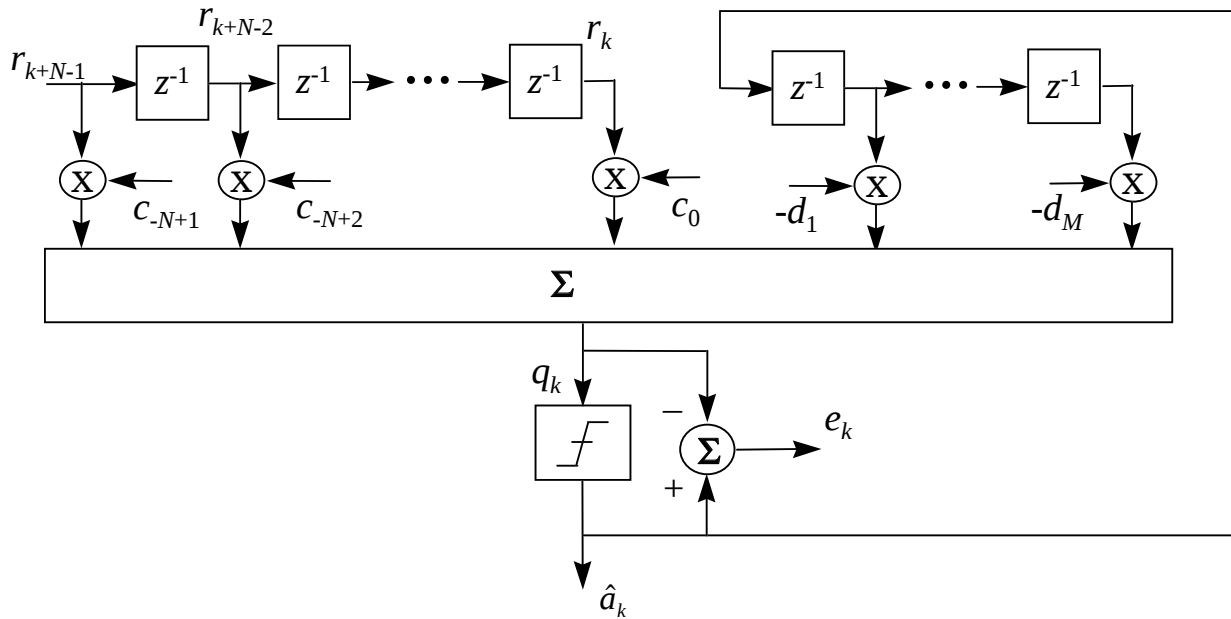
Adaptiivisen DFE:n FIR-rakenne



- ◆ Oletamme nyt, kuten edellä, että prekursori on antikausaalinen N -kertoiminen FIR-suodin ja postkursori on M -kertoiminen ja kausaalinen (ei-kausalisuus poistuu käytännössä viivettä lisäämällä). Laskenta voidaan esittää differenssiyhtälöllä

$$q_k = \sum_{i=-(N-1)}^0 c_i r_{k-i} - \sum_{i=1}^M d_i \hat{a}_{k-i} \quad (11.80)$$

- ◆ eli se on painotettu keskiarvo uusista input-näytteistä ja vanhoista päätöksistä. Vaikka järjestelmä sisältää takaisinkytkentää, se voidaan esittää formaalisti **Kuvan 11-10** FIR-rakenteella.



Kuva 11-10. DFE-korjaimen rakenne.

Adaptiivisen DFE:n MSE-ratkaisu



- ◆ Input-näytteet koostuvat paitsi halutusta symboleista myös keskinäisvaikutuksesta ja kohinasta, eli ne ovat muotoa

$$r_k = p_0 a_k + \sum_{n \neq 0} p_n a_{k-n} + n_k = \sum_n p_n a_{k-n} + n_k \quad (11.80b)$$

jossa summausindeksit ja kertoimet p_n eivät yleensä ole tarkkaan tiedossa. Sijoittamalla kaavaan (11.80) saadaan

$$\begin{aligned} q_k &= \sum_{i=-(N-1)}^0 c_i \sum_n p_n a_{(k-i)-n} - \sum_{i=1}^M d_i \hat{a}_{k-i} + \sum_{i=-(N-1)}^0 c_i n_{k-i} \\ &= \sum_m v_m a_{k-m} + \sum_{i=-(N-1)}^0 c_i n_{k-i} \end{aligned} \quad (11.80c)$$

jossa päätökset on oletettu oikeiksi.



- ◆ Tämä voidaan jakaa kolmeen termiin:

$$q_k = v_0 a_k + i_k + n_k$$

- ◆ Ensimmäinen on haluttu symboli. Toinen on ISI-termi joka on muotoa

$$i_k = \sum_{m \neq 0} v_m a_{k-m} \quad (11.81)$$

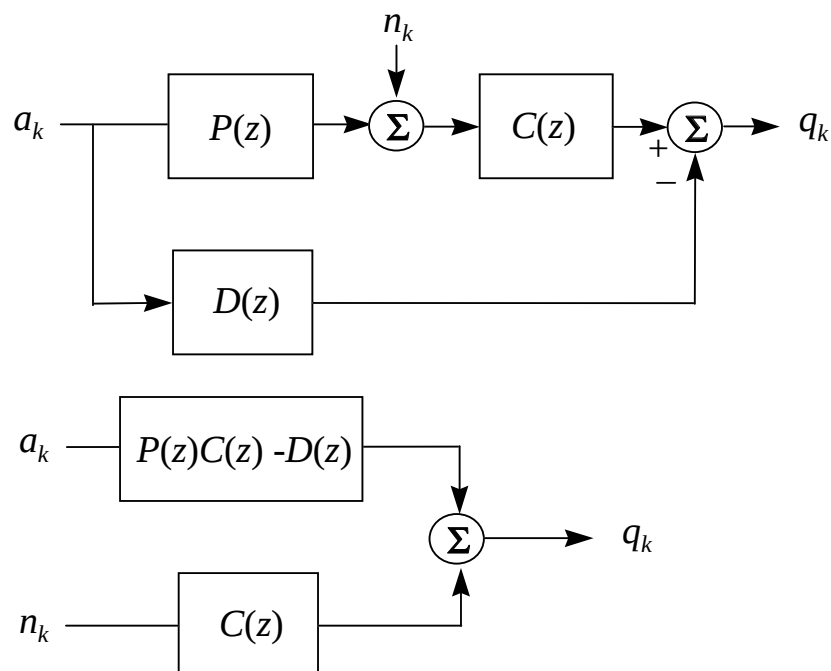
jossa

$$v_k = \begin{cases} \sum_{i=(-N-1)}^0 c_i p_{k-i} - d_k, & 1 \leq k \leq M \\ \sum_{i=(-N-1)}^0 c_i p_{k-i}, & \text{muulloin} \end{cases} \quad (11.82)$$

- ◆ Kolmas termi edustaa kohinaa.



- ◆ Tätä havainnollistaa **Kuva 11-11**:





- ◆ Kun oletetaan että kohina ja lähetetyt symbolit (ja siten myös ISI) ovat riippumattomia, niitä voidaan MSE:n minimoinnin kannalta tarkastella erikseen. Kun lisäksi symbolit keskenään ovat riippumattomia, on ilmeistä että kokonaisvirheen odotusarvo voidaan minimoida *valitsemalla postkursorin kertoimet d_k siten, että M ISI-termiä nollautuu.* Näin saadaan MMSE-ratkaisu postkursorikertoimille muodossa

$$d_m = \sum_{i=(-N-1)}^0 c_i p_{m-i} \quad 1 \leq m \leq M \quad (11.83)$$

- ◆ Olettaen että ISI-termit tunnetaan, postkursorikertoimet riippuvat suoraan prekursorista.



- ◆ Toteutuksessa voitaisiin optimoida suoraan prekursorikertoimet sopivalla algoritmilla ja laskea postkursorikertoimet kaavasta (11.83).
- ◆ Käytännössä on kuitenkin helpompaa käyttää **Kuvan 11-11** rakennetta ja signaaleja suoraan adaptointiin, kuten seuraavaksi johdetaan.



- ◆ Formaalisti SG-algoritmi voidaan johtaa DFE:n tapaukselle suoraviivaisesti. Määritellään *kerroinvektori*

$$\mathbf{v} = [c_{-(N-1)} \quad \cdots \quad c_0 \quad -d_1 \quad \cdots \quad -d_M]^T \quad (11.86)$$

- ◆ joka sisältää siis prekursorin ja postkursorin kertoimet peräkkäin, $N+M$ elementtiä. Vastaava pidennetty *datavektori* on muotoa

$$\mathbf{w}_k = [r_{k+(N-1)} \quad \cdots \quad r_k \quad a_{k-1} \quad \cdots \quad a_{k-M}]^T \quad (11.87)$$

...SG-algoritmi DFE:lle



- ◆ Virhesignaali voidaan nyt esittää muodossa

$$e_k = a_k - \mathbf{v}^T \mathbf{w}_k \quad (11.88)$$

joka on formaalisti samanlainen kuin LE-korjaimen virheen lauseke. Analogiaan vedoten voidaan katsoa mallia SG-algoritmile kaavasta (11.53):

$$\mathbf{c}_{k+1} = \mathbf{c}_k + \beta e_k \mathbf{r}_k^* \quad (11.53)$$

- ◆ Ottamalla kerroinvektorillekin päivitysindeksi k saadaan DFE-korjaimen SG-algoritmi muotoon

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \beta e_k \mathbf{w}_k^* \quad (11.89)$$



- ◆ Aiemmin analysoitiin *ideaalisen* DFE-korjaimen ominaisuuksia stationäärisessä tilanteessa. Olennaista oli *kohina-vahvistuksen pieneneminen* koska ISIn poisto helpottuu.
- ◆ Lähteessä *Bingham: The Theory and Practice of Modem Design* (Wiley 1988, s. 289) todetaan, että SG-DFE *suppenee nopeammin* kuin SG-LE, koska kanavaa ei tarvitse ‘kääntää’ kokonaan. Näin SG-DFE on vähemmän herkkä input-signaalin autokorrelaatiomatriisin ominaisarvohajalle.
- ◆ Koska pre- ja postkursorien kerrointen laskenta perustuu samaan virhesignaaliin, kompleksisuus ja *suppeneminen* ovat suunnilleen samanlaista kuin $(N+M)$ -tappisella SG-LE-FIR-korjaimella. *Toteutus* on siis yksinkertainen mutta tehokas.