Tehnici de procesare a semnalelor audio
Filtrări adaptive

CORNELIU RUSU

Corneliu.Rusu@bel.utcluj.ro

Filtrări adaptive

- Filtrarea Wiener
- Filtre LMS
- Filtre RLS

Filtrarea Wiener

- Filtrul Wiener = filtru liniar optim

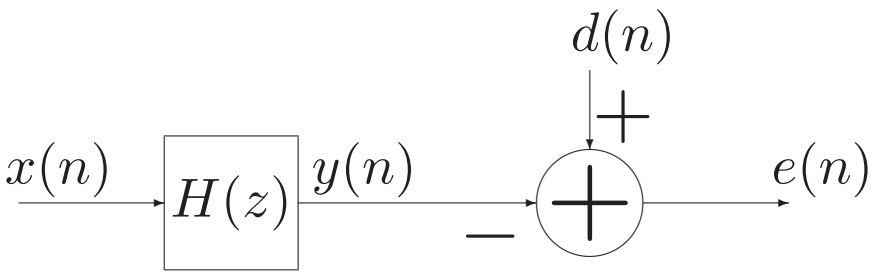


Figura 1: Filtrul Wiener

- $x(n)$ semnal de intrare
  $h(n)$ secvența pondere a filtrului liniar
  $y(n)$ semnal de ieșire
  $d(n)$ semnal dorit
  $e(n)$ eroare de estimare

- Obiectivul filtrării Wiener constă în a determina secvența pondere $h(n)$ care minimizează eroarea $e(n)$ într-un anumit sens statistic.

- Există câteva modalități de a alege această eroare minimă

- Funcție de cost (funcție obiectiv, index de performanță)

- Dpdv matematic cel mai des se minimizează eroarea pătratică a lui $e(n)$

- Aceasta funcție obiectiv utilizată la filtrarea Wiener

- Sunt câteva structuri pentru filtrul Wiener $\Rightarrow$ soluții diferite

  1. Filtru necauzal și de durată infinită
  2. Filtru cauzal IIR
  3. Filtru Wiener FIR de ordinul M

- Vom căuta soluția optimă din fiecare clasă.

- Dintre toate soluțiile, FIR s-a dovedit cea mai populară.

- Cel mai important avantaj este legat de abilitatea structurilor FIR de a implementa soluții adaptive la problema filtrării Wiener.

- Să considerăm sistemul de comunicații următor (fig. 2):

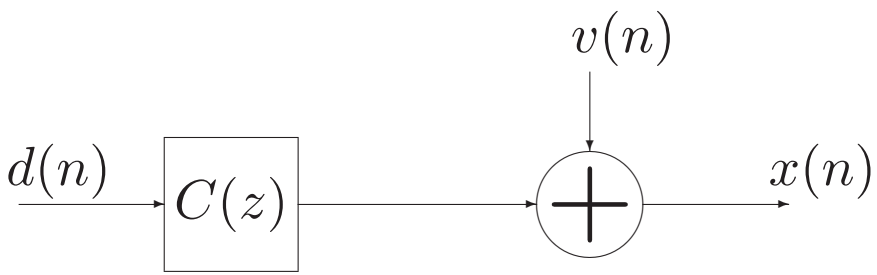


Figura 2: Sistem de comunicații

- Un semnal $d(n)$ este transmis printr-un canal de comunicație care este descris printr-o funcție de transfer $C(z)$ posibil necunoscută. $C(z)$ poate incorpora în modelul său și efectele distorsiunii canalului.

- Avem zgomot pe canal $v(n)$ care corupe semnalul recepționat $x(n)$.

- Dorim să proiectăm un egalizator $E(z)$ care să încerce să înlăture efectele nedorite ale lui $C(z)$ și $v(n)$ (fig.3).

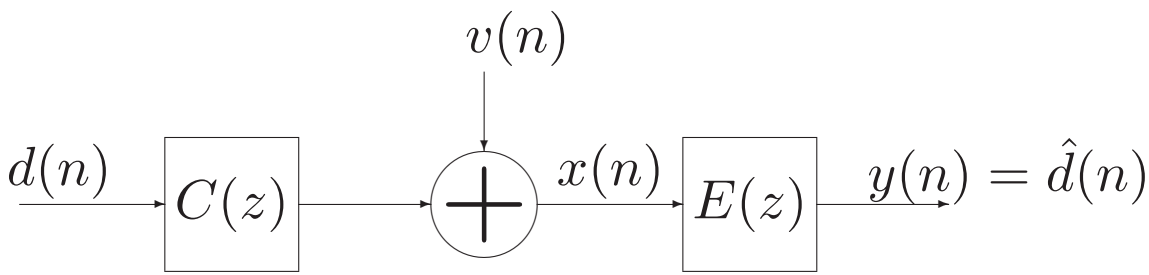


Figura 3: Sistem de comunicații completat

- Egalizatorul are intrarea $x(n)$ și produce o ieșire $y(n)$. Semnalul dorit este $d(n)$, adică semnalul transmis.

- Vom utiliza un filtru Wiener $E(z)$ la egalizator, care minimizează eroarea medie pătratică dintre $y(n)$ și $d(n)$.

- Soluția acestei probleme necesită o descriere statistică a semnalului recepționat.

- Egalizator cauzal sau necauzal, soluții diferite

- Se preferă soluții cauzale și dintre diferitele tipuri de filtre liniare se impune pentru egalizator o soluție FIR sau IIR.

- În cele mai multe cazuri un FIR cu suficiente derivații este de ajuns în implementare.

Principiul de ortogonalitate

- Presupunem că $x(n)$ și $d(n)$ sunt mixt stationare în sens larg.

- Pentru discuția prezentă nu trebuie să mai facem nici o altă presupunere relativ la natura filtrului, în afara liniarității.

- Ieșirea filtrului este dată de:

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k).$$

  Eroarea dintre răspunsul dorit și ieșire este $e(n) = d(n) - y(n)$.

- Obiectivul minimizării Wiener este să determinăm coeficienții $h(n)$ care minimizează indexul de performanță J dat de $J = E[e^2(n)]$.

- Filtrul Wiener va minimiza eroarea medie pătratică dintre semnalul dorit și ieșirea filtrului

- Indexul de performanță J este o funcție de secvența pondere h

- Minimizarea se face setând zero gradientul lui J în raport cu componentele lui h

- Componenta a n-a a lui ∇J este dată de:

$$[\nabla J]_n = \frac{\partial J}{\partial [h(n)]}; \quad \frac{\partial J}{\partial [h(n)]} = E\left\{2\frac{\partial [e(n)]}{\partial [h(k)]} \cdot e(n)\right\}.$$

Pe de altă parte:

$$\frac{\partial [e(n)]}{\partial [h(k)]} = -\frac{\partial [y(n)]}{\partial [h(k)]} = -x(n-k)$$

$$\frac{\partial J}{\partial [h(n)]} = -2E\left\{x(n-k)e(n)\right\}.$$

- Deoarece fiecare componentă a gradientului trebuie să fie nulă pentru optimalitate, obținem că:

$$E\left\{x(n-k)e_0(n)\right\} = 0, \ \forall n.$$

unde $e_0(n)$ se referă la eroarea filtrului când filtrul operează optimal.

- Ecuația anterioară se mai numește și principiul de ortogonalitate.

- Astfel se afirmă că în caz optimal eroarea dintre ieșire și răspunsul dorit este necorelat cu orice eșantion de intrare folosit în calculul ieșirii.

- Principiul de ortogonalitate ne oferă astfel condiții necesare pentru optim.

- Este ușor de arătat că pentru un filtru care operează în condiții optimale eroarea este de asemenea ortogonală pe ieșirea filtrului:

$$E\left\{y_0(n-k)e_0(n)\right\} = 0, \ \forall n.$$

- Valoarea lui J pentru care filtrul este optimal este $J_{\min} = E[e_0^2(n)]$.

$$e_0(n) = d(n) - y_0(n) \quad d(n) = y_0(n) + e_0(n)$$

$$E[d^2(n)] = \sigma_d^2 = E[y_0^2(n) - e_0^2(n) + 2e_0(n)y_0(n)].$$

Dar deoarece $e_0(n)$ și $y_0(n)$ sunt ortogonale, rezultă:

$$\sigma_d^2 = \sigma_{y_0}^2 + J_{\min},$$

deci:

$$J_{\min} = \sigma_d^2 - \sigma_{y_0}^2.$$

- Vom arăta acum cum se determină parametrii filtrului optimal.

- Principiul de ortogonalitate afirmă că $E\left\{x(n-k)e_0(n)\right\} = 0$.

- Eroarea optimală poate fi exprimată ($h_0(i)$ indică parametrii filtrului optimal):

$$e_0(n) = d(n) - \sum_{i=-\infty}^{\infty} h_0(i)x(n-i)$$

- Substituind această relație în ecuația anterioară, avem că:

$$E\left\{x(n-k)\left[d(n) - \sum_{i=-\infty}^{\infty} h_0(i)x(n-i)\right]\right\} = 0$$

$$\sum_{i=-\infty}^{\infty} h_0(i)E[x(n-k)x(n-i)] = E[x(n-k)d(n)].$$

Dar:

$$E[x(n-k)x(n-i)] = r_{xx}(k-i);$$
$$E[x(n-k)d(n)] = r_{xd}(k).$$

Rezultă că obținem:

$$\sum_{i=-\infty}^{\infty} h_0(i)r_{xx}(k-i) = r_{xd}(k), \ \forall k. \tag{1}$$

- Ecuația anterioară se numește ecuația Wiener-Hopf discretă în timp și rezolvarea ei ne dă coeficienții $h(i)$ optimali.

- Se observă că este o ecuație implicită.

Filtrul Wiener IIR necauzal

- Vom caracteriza în domeniul z filtrul Wiener IIR și necauzal, adică coeficienții $h(i)$ pot lua orice valori în ecuațiile Wiener-Hopf (1).

- Vom calcula soluția acestei ecuații în domeniul z: $H(z)P_{xx}(z) = P_{xd}(z)$.

- Filtrul Wiener este caracterizat de:

$$H(z) = \frac{P_{xd}(z)}{P_{xx}(z)}.$$

- Dacă densitățile spectrale $P_{xd}(z)$ și $P_{xx}(z)$ nu sunt cunoscute, ele pot fi estimate din datele măsurate.

- Deoarece deocamdată nu există nici o constrângere temporală asupra coeficienților filtrului, soluția poate fi și necauzală.

- Când $H(z)$ este soluția Wiener, eroarea medie pătratică este minimă:

$$J_{\min} = E[(d(n) - y(n))^2] = E[d^2(n)] - 2E[d(n)y(n)] + E[y^2(n)]$$
$$J_{\min} = \sigma_d^2 + r_{yy}(0) - 2r_{dy}(0).$$

- Această expresie poate fi evaluată cu inversa transformatei în z:

$$J_{\min} = \sigma_d^2 + \frac{1}{2\pi j} \oint_C [P_{yy}(z) - 2P_{dy}(z)] z^{-1} dz. \tag{2}$$

- Deoarece $y(n)$ este generată prin filtrarea lui $x(n)$ prin $H(z)$, rezultă că avem:

$$P_{dy}(z) = H(z)P_{dx}(z); \quad P_{yy}(z) = H(z)H(z^{-1})P_{xx}(z).$$

Astfel ecuația (2) devine:

$$J_{\min} = \sigma_d^2 + \frac{1}{2\pi j} \oint_C [H(z^{-1})P_{xx}(z) - P_{dx}(z)]H(z)z^{-1}dz \qquad (3)$$

și poate fi utilizată ca să calculăm eroarea medie pătratică produsă de filtrarea Wiener.

Interpretarea filtrării Wiener

- Vom considera același sistem de comunicație (fig. 2), unde vom presupune că $v(n)$ este un zgomot alb de medie nulă cu putere σ_v^2 și necorelat cu semnalul transmis $d(n)$.

$$P_{xd}(z) = C(z^{-1})P_{dd}(z) + P_{dv}(z)$$

- Deoarece $d(n)$ și $v(n)$ sunt necorelate, avem:

$$P_{xd}(z) = C(z^{-1})P_{dd}(z); \quad P_{xx}(z) = C(z)C(z^{-1})P_{dd}(z) + \sigma_v^2.$$

- Astfel filtrul Wiener este caracterizat de:

$$H(z) = \frac{P_{xd}(z)}{P_{xx}(z)} = \frac{C(z^{-1})P_{dd}(z)}{C(z)C(z^{-1})P_{dd}(z) + \sigma_v^2}.$$

- În cazul limită $\sigma_v^2 = 0$, avem că:

$$H(z) = \frac{1}{C(z)},$$

adică filtrul Wiener acționează ca și filtrul invers al canalului de comunicație $C(z)$.

- Filtrul Wiener acționează ca și un eliminator corelat.

- Principiul de ortogonalitate: ieșirea $y(n)$ a filtrului Wiener este produsă de semnalul $x(n)$ rezultând o eroare $e(n)$ care este necorelată cu $x(n)$.

- Din $e(n) = d(n) - y(n)$ și $e(n)$ este necorelat cu $x(n) \Leftrightarrow$ intercorelația dintre $e(n)$ și $x(n)$ a fost eliminată prin extragerea lui $y(n)$.

- $y(n)$ este un estimator optimal al părții din $d(n)$ care este corelată cu $x(n)$.

- În consecință, dacă $x(n) = x_1(n) + x_2(n)$, unde $x_1(n)$ și $d(n)$ sunt necorelate, filtrul Wiener va acționa la fel ca și când $x(n) = x_1(n)$.

- Această proprietate a filtrului Wiener este importantă în aplicații precum eliminarea zgomotului sau a ecoului.

Filtre Wiener IIR cauzale (cu prealbire)

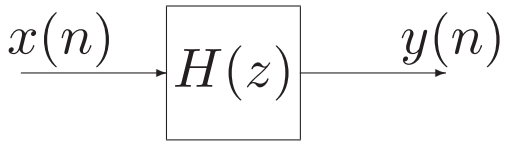
- Să considerăm acum soluția cauzală a problemei Wiener de filtrare

- Putem utiliza teorema de factorizare spectrală pentru a simplifica problema de filtrare Wiener cauzală.

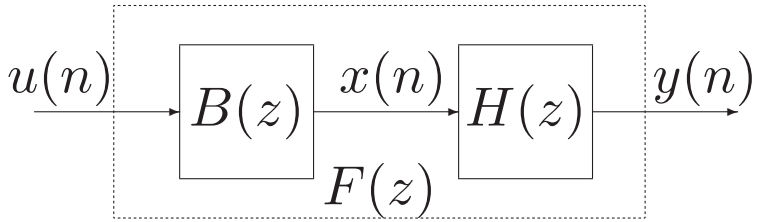
- Acest procedeu se numește prealbire.
  Factorizăm $P_{xx}(z) = \sigma_u^2 B(z) B(z^{-1})$
  $B(z)$ este un sistem cauzal și de fază minimă

- În acest scop, modelăm procesul stochastic $x(n)$ ca și ieșirea filtrului $B(z)$ excitat de un proces zgomot alb $u(n)$.

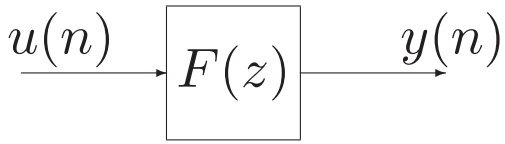
- Problemă de estimare a lui $d(n)$ în funcție de secvența zgomot alb $u(n)$



a) Problema inițială: aflarea lui $H(z)$ astfel încât $E\{[d(n) - y(n)]^2\}$ să fie minimă.



b) Factorizăm $P_{xx}(z) = \sigma_u^2 B(z)B(z^{-1})$.



c) $F(z) = H(z)B(z)$

Figura 4: Filtrarea Wiener cu prealbire

- Funcția de transfer peste întreg sistemul, de la $u(n)$ la $y(n)$ este $F(z) = B(z)H(z)$.

- Dacă găsim o funcție cauzală $F(z)$, atunci putem considera filtrul Wiener ca fiind:
$$H(z) = \frac{F(z)}{B(z)}.$$
care va fi la rândul său cauzal deoarece $B(z)$ este cauzal, de fază minimă.

- Rescriem ecuațiile Wiener-Hopf pentru secvența pondere $f(n)$:
$$\sum_{i=0}^{\infty} f(i) r_{uu}(k-i) = r_{ud}(k), \forall k \geq 0,$$
unde suma este extinsă doar pentru indicii pozitivi, $F(z)$ fiind presupus cauzal.

- Dacă $u(n)$ este alb, atunci $r_{uu}(k-i) = \delta(k-i)$ și ecuația Wiener-Hopf se reduce la:

$$r_{ud}(k) = f(k),$$

care ne dă secvența pondere a lui $F(z)$:

$$f(k) = r_{ud}(k), \forall k \geq 0.$$

- Vom calcula acum funcția de transfer $F(z)$ ca și o transformată în z:

$$F(z) = \sum_{n=0}^{\infty} f(n) z^{-n} = \sum_{n=0}^{\infty} r_{ud}(n) z^{-n},$$

adică suma peste partea cauzală a lui $P_{ud}(z)$, notată $[P_{ud}(z)]_+$. Rezultă că:

$$F(z) = [P_{ud}(z)]_+.$$

- Relația dintre $P_{xd}(z)$ și $P_{ud}(z)$? Dar $P_{xd}(z) = X(z^{-1})D(z)$

$$P_{ud}(z) = \frac{P_{xd}(z)}{B(z^{-1})}.$$

- Din $F(z) = [P_{ud}(z)]_+$, filtrul $F(z)$ este dat de:

$$F(z) = [P_{ud}(z)]_+ = \left[\frac{P_{xd}(z)}{B(z^{-1})}\right]_+.$$

- Astfel soluția problemei de filtrare Wiener cauzală este dată de:

$$H(z) = \frac{1}{B(z)}\left[\frac{P_{xd}(z)}{B(z^{-1})}\right]_+.$$

- Deoarece $B(z)$ este de fază minimă, soluția $H(z)$ va fi stabilă dacă partea cauzală a lui $P_{xd}(z)$ este stabilă.

- Acest lucru este valabil în orice problemă de interes.

- În acest fel am rezolvat problema cauzală a filtrării Wiener.

- Singura dificultate reală constă în determinarea părții cauzale a funcției de transfer.

- În mod obișnuit putem utiliza teorema reziduurilor pe un contur de integrare.

Filtre FIR Wiener

- Filtrul Wiener este constrâns să aibă o secvență pondere de lungime N.
- Cea mai populară structură
- Fie deci $H(z)$ funcția de sistem a unui filtru transversal cu N derivații, caracterizat de secvența pondere:

$$\mathbf{h} = [h_0, h_1, \ldots, h_{N-1}]^T.$$

Ecuația Wiener-Hopf afirmă că:

$$\sum_{i=0}^{N-1} h_{oi} r_{xx}(n-i) = r_{xd}(n), \; n = 0, 1, \ldots, N-1$$

- Sunt doar N derivații și numai cele mai recente N eșantioane sunt utilizate în optimizare.

1. Vectorul de intrare (cele mai recente N eșantioane):

$$\mathbf{x}(n) = [x(n), x(n-1), ..., x(n-N+1)]^T;$$

2. Matricea de autocorelație a vectorului $\mathbf{x}(k)$:

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & \cdots & r(N-1) \\ r(1) & r(0) & \cdots & r(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(N-1) & r(N-2) & \cdots & r(0) \end{bmatrix};$$

3. Vectorul intercorelație dintre intrare și semnalul dorit:

$$\mathbf{p} = E[\mathbf{xd}] = \begin{bmatrix} r_{xd}(0) \\ r_{xd}(1) \\ \vdots \\ r_{xd}(N-1) \end{bmatrix}.$$

Astfel ecuația Wiener-Hopf poate fi scrisă sub formă matricială $\mathbf{Rh} = \mathbf{p}$, iar soluția optimală devine $\mathbf{h}_0 = \mathbf{R}^{-1}\mathbf{p}$.

- Ieșirea filtrului transversal se poate scrie sub forma unei relații matriciale:

$$\mathbf{y}(n) = \mathbf{h}^T\mathbf{x}(n),$$

iar eroarea va fi:

$$e(n) = d(n) - \mathbf{h}^T\mathbf{x}(n).$$

- S-a folosit următorul index de performanță pentru filtrul Wiener:

$$J(\mathbf{h}) = E[e^2(n)]$$

care se poate scrie sub forma:

$$J(\mathbf{h}) = E[e^2(n)] = \sigma_d^2 - \mathbf{h}^T E[\mathbf{x}(n)d(n)] - E[\mathbf{x}^T(n)d(n)]\mathbf{h} + \mathbf{h}^T E[\mathbf{x}(n)\mathbf{x}^T(n)]\mathbf{h}.$$

- Utilizând descrierile matriciale anterioare avem:

$$J = \sigma_d^2 - 2\mathbf{h}^T\mathbf{p} + \mathbf{h}^T\mathbf{R}\mathbf{h}. \tag{4}$$

- Valoarea minimă se obține pentru $\mathbf{h} = \mathbf{h}_o$, unde $\mathbf{h}_o = \mathbf{R}^{-1}\mathbf{p}$.

- Substituind această soluție, se obține $J_{\min} = \sigma_d^2 - \mathbf{h}_o^T\mathbf{p}$.

- Ecuația anterioară pătratică (4) descrie un paraboloid.

- Este important că această funcție de performanță are un minim global la $\mathbf{h} = \mathbf{h}_o$ și nu mai are nici un alt minim local.

- Rezultă că avem o singură soluție la problema de filtrare Wiener, proprietate care nu este valabilă în cazul filtrării Wiener IIR.

- Uneori este mai convenabil din punct de vedere geometric să scriem funcția de performanță sub o formă cuadratică în $\mathbf{h}$, adică:

$$J(\mathbf{h}) = J_{\min} + (\mathbf{h} - \mathbf{h}_o)^T \mathbf{R} (\mathbf{h} - \mathbf{h}_o).$$

- $\mathbf{R}$ este matrice de autocorelație $\Rightarrow$ descompunerea[1] $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H$

[1] Prin $\mathbf{Q}^H$ notăm conjugata hermitică $\mathbf{Q}^H = (\mathbf{Q}^T)^*$, adică conjugata transpusei matricei $\mathbf{Q}$.

- $\mathbf{Q}$ este matrice ortogonală, având coloanele egale cu vectorii proprii ai lui $\mathbf{R}$ și $\mathbf{\Lambda}$ este matricea diagonală corespunzătoare valorilor proprii λ_i:

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$$

Avem deci că:

$$J = J_{\min} + (\mathbf{h} - \mathbf{h}_o)^T \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H (\mathbf{h} - \mathbf{h}_o)$$

- Această ecuație ne permite o reprezentare simplă a suprafeței de eroare descrisă de o formă cuadratică.

- Să executăm o transformare în spațiul ponderilor astfel:

1. Prima dată translatăm originea la vectorul optimal $\mathbf{h}_o$;

2. Facem o schimbare de bază unde noua bază sunt vectorii proprii ai matricei funcției de autocorelație $\mathbf{R}$. Avem transformarea afină:

$$\mathbf{h} = \mathbf{Q}\mathbf{z} + \mathbf{h}_o.$$

- Deoarece matricea $\mathbf{Q}$ este ortogonală, rezultă că:

$$\mathbf{z} = \mathbf{Q}^H(\mathbf{h} - \mathbf{h}_o), \quad J = J_{\min} + \mathbf{z}^T\mathbf{\Lambda}\mathbf{z} = J_{\min} + \sum_{i=1}^{N} \lambda_i |z_i|^2.$$

- Minimul este chiar în origine la $\mathbf{z} = 0$ în spațiul transformat.

- Secțiunile paraboloidului prin plane vor fi elipse, eventual cercuri când toate λ_i sunt egale.

Metoda celei mai rapide descreșteri a gradientului

- Este o metodă iterativă de căutare a soluției optime Wiener în care ne mișcăm pe suprafața de performanță în direcția negativă a gradientului.

- Metoda presupune cunoașterea completă a matricelor $\mathbf{R}$ și $\mathbf{p}$ și de aceea nu poate fi utilizată atunci când nu se cunosc statisticile semnalelor.

- Cunoașterea acestei metode ne furnizează un model de bază pentru câțiva algoritmi adaptivi des utilizați.

- În general algoritmii rezultați se bazează pe aproximații ale statisticilor semnalului care pot fi calculate din datele disponibile.

- În funcție de efortul computațional, în general se dorește să îmbunătățim aproximația, și astfel calitatea rezultatelor va varia de la caz la caz.

- Revenim la configurațiile anterioare
  Singura diferență: vectorul $\mathbf{h}(n)$ este variant în timp

$$\mathbf{h}(n) = [h_0(n), h_1(n), \ldots, h_{N-1}(n)]^T$$

- În continuare ieșirea filtrului adaptiv este $y(n) = \mathbf{h}^T\mathbf{x}(n)$. Eroarea va fi:

$$e(n) = d(n) - \mathbf{h}(n)^T\mathbf{x}(n).$$

- Obiectivul nostru este să determinăm un algoritm care să actualizeze ponderile variabile ale vectorului $\mathbf{h}(n)$ astfel încât să se apropie de soluția Wiener $\mathbf{h}_o$.

- Reamintim că $\mathbf{h}_o$ satisface ecuația Wiener-Hopf $\mathbf{R}\mathbf{h}_o = \mathbf{p}$ unde

$$\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)], \ \mathbf{p} = E[d(n)\mathbf{x}(n)].$$

- Ponderea optimă minimizează eroarea medie pătratică:

$$J = E[e^2(k)] = \sigma_d^2 - 2\mathbf{h}^T\mathbf{p} + \mathbf{h}^T\mathbf{R}\mathbf{h}.$$

și avem $J_{\min} = \sigma_d^2 - \mathbf{h}_o^T\mathbf{p}$.

- În acest fel, eroarea medie pătratică este acum o funcție de n:

$$J(n) = \sigma_d^2 - 2\mathbf{h}^T(n)\mathbf{p} + \mathbf{h}^T(n)\mathbf{R}\mathbf{h}(n)$$

și dorim ca:

$$\lim_{n\to\infty} \mathbf{h}(n) = \mathbf{h}_o, \quad \lim_{n\to\infty} J(n) = J_{\min}.$$

- Principiul care stă la baza metodei celei mai rapide descreșteri este să începem de la o pondere inițială $\mathbf{h}(0)$ la care este asociată eroarea medie pătratică corespunzătoare $J[\mathbf{h}(0)]$.

- Apoi ponderile sunt actualizate într-o astfel de manieră care face ca eroarea medie pătratică să se miște în direcția celei mai mici descreșteri a suprafeței de performanță.

- Direcția este determinată prin gradientul negativ al suprafeței de performanță $J(\mathbf{h})$.

- Ponderile sunt actualizate prin (factorul $\frac{1}{2}$ din motive matematice):

$$\mathbf{h}(n+1) = \mathbf{h}(n) - \frac{1}{2}\mu\nabla_n, \text{ unde } \nabla_n = \frac{\partial J(n)}{\partial \mathbf{h}(n)}.$$

Gradientul este:

$$\nabla_n = -2\mathbf{p} + 2\mathbf{R}\mathbf{h}(n)$$

astfel încât ecuația de iterație a gradientului devine:

$$\mathbf{h}(n+1) = (\mathbf{I} - \mu\mathbf{R})\mathbf{h}(n) + \mu\mathbf{p}$$

- Ecuație matricială cu diferențe de ordinul I, a cărei comportare este controlată de parametrul μ, numit și pasul algoritmului.

- Dacă ținem seama că $\mathbf{R}\mathbf{h}_o = \mathbf{p}$, atunci avem:

$$\mathbf{h}(n+1) = (\mathbf{I} - \mu\mathbf{R})\mathbf{h}(n) + \mu\mathbf{R}\mathbf{h}_o.$$

- Extrăgând $\mathbf{h}_o$ din ambii termeni, vom avea:

$$\mathbf{h}(n+1) - \mathbf{h}_o = (\mathbf{I} - \mu\mathbf{R})(\mathbf{h}(n) - \mathbf{h}_o). \tag{5}$$

- Să reamintim că matricea de autocorelație are descompunerea $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H$, unde $\mathbf{\Lambda}$ este o matrice reală, diagonală și $\mathbf{Q}$ satisface egalitatea $\mathbf{Q}^H\mathbf{Q} = \mathbf{Q}\mathbf{Q}^H = \mathbf{I}$.

- Cu descompunerea lui $\mathbf{R}$, ecuația (5) devine:

$$\mathbf{h}(n+1) - \mathbf{h}_o = (\mathbf{I} - \mu\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H)[\mathbf{h}(n) - \mathbf{h}_o]$$

sau

$$\mathbf{Q}^H[\mathbf{h}(n+1) - \mathbf{h}_o] = (\mathbf{I} - \mu\mathbf{\Lambda})\mathbf{Q}^H[\mathbf{h}(n) - \mathbf{h}_o].$$

- Expresiile din ambii termeni seamănă cu $\mathbf{z} = \mathbf{Q}^H(\mathbf{h} - \mathbf{h}_o)$. Fie:

$$\mathbf{z}(n) = \mathbf{Q}^H[\mathbf{h}(n) - \mathbf{h}_o] \text{ sau } \mathbf{z}(n+1) = [\mathbf{I} - \mu\mathbf{\Lambda}]\mathbf{z}(n),$$

care ne dă punctul optim $\mathbf{z} = 0$.

- Convergența algoritmului este garantată dacă $\mathbf{z}(n)$ converge spre zero.

- Rezultă că avem: $\mathbf{h}(n) = [\mathbf{I} - \mu\mathbf{R}]^n[\mathbf{h}(0) - \mathbf{h}_o] + \mathbf{h}_o$.

- Revenim la ecuația anterioară în $\mathbf{z}$, care pe fiecare axă devine:

$$z_m(n+1) = (1 - \mu\lambda_m) z_m(n), \; m = 1, 2, \ldots, N,$$

sau

$$z_m(n) = (1 - \mu\lambda_m)^n z_m(0),$$

unde $z_m(0)$ sunt coordonatele inițiale.

- Este clar că $z_m(n)$ vor converge către zero dacă $|1 - \mu\lambda_m| < 1$ sau

$$0 < \mu < \frac{2}{\lambda_m}$$

- Marginea este dată de cea mai mare valoare proprie:

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

- Deci dacă μ satisface condiția anterioară, atunci algoritmul converge indiferent de condiția inițială.

- Problema care rămâne: viteza cu care se apropie de origine.

- Dacă toate valorile proprii ale lui $\mathbf{R}$ sunt aproximativ egale, atunci este posibil să găsim o valoare μ care să forțeze cantitatea $1 - \mu\lambda_m$ suficient de mică pentru orice m.

- Astfel algoritmul poate să conveargă tot așa de repede ca și cantitatea $(1 - \mu\lambda_m)^n$, adică foarte rapid.

- Dacă valorile proprii ale lui $\mathbf{R}$ sunt împrăștiate situația este mai dificilă.

Algoritmul LMS

- Algoritmul LMS (Least Mean Square) este unul dintre cei mai simpli algoritmi adaptivi stochastici de gradient.

- Există multe variante de algoritmi derivați din LMS.

- Structura de filtru pentru algoritmul LMS de bază este filtrul transversal cu ponderi $\mathbf{h}(n)$ variabile în timp.

- Procesul stochastic de intrare este $\mathbf{x}(n)$ și răspunsul dorit este $d(n)$.

- Ieșirea filtrului transversal este dată de $y(n) = \mathbf{h}^T(n)\mathbf{x}(n)$, unde $\mathbf{x}(n)$ este vectorul de intrare conținând cele mai recente N eșantioane.

- Eroarea dintre ieșire și răspunsul dorit este:

$$e(n) = d(n) - y(n) = d(n) - \mathbf{h}^T(n)\mathbf{x}(n).$$

- Scopul algoritmului LMS este de a actualiza iterativ ponderile $\mathbf{h}(n)$ astfel încât să minimizeze criteriul de performanță $J = E[e^2(n)]$.

- Reamintim că algoritmul de gradient utilizează recursia:

$$h(n+1) = h(n) - \frac{1}{2}\mu\nabla_n.$$

- Algoritmul LMS va fi descris de recursia ($\hat{\nabla}_n$ este numit gradient stochastic, adică o aproximație a gradientului algoritmului adevărat ∇_n):

$$h(n+1) = h(n) - \frac{1}{2}\mu\hat{\nabla}_k \tag{6}$$

- Reamintim că gradientul algoritmul adevărat este dat de:

$$\nabla_n = \frac{\partial J}{\partial \mathbf{h}(n)} = -2\mathbf{R}\mathbf{h} - 2\mathbf{p}.$$

- Dacă facem aproximațiile grosiere: $\mathbf{R} \approx \mathbf{x}(n)\mathbf{x}^T(n)$ și $\mathbf{p} \approx d(n)\mathbf{x}(n)$, adică de fapt ignorăm operatorii statistici de expectație, atunci gradientul stocastic este dat de:

$$\hat{\nabla}_n = 2[\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{h}(n) - 2d(n)\mathbf{x}(n)].$$

- Atunci ecuația (6) devine recursia LMS:

$$\mathbf{h}(n+1) = \mathbf{h}(n) - \mu[\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{h}(n) - d(n)\mathbf{x}(n)].$$

- Această ultimă ecuație poate avea și o altă formulare.

- Pentru aceasta reamintim că $e(n) = d(n) - \mathbf{h}^T(n)\mathbf{x}(n)$, caz în care recursia anterioară poate fi scrisă sub forma consacrată:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu e(n)\mathbf{x}(n).$$

Convergența algoritmului LMS

- Algoritm adaptiv: este necesar să examinăm proprietățile de convergență.
- De dorit ca LMS să conveargă către soluția Wiener = vectorul pondere optim care minimizează valoarea medie pătratică a semnalului eroare.
- Este relativ ușor de arătat convergența în medie a algoritmului LMS.
- Aplicând expectația în ambele părți ale ecuației anterioare obținem:

$$E[\mathbf{h}(n+1)] = E[\mathbf{h}(n)] - \mu E[\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{h}(n)] + E[d(n)\mathbf{x}(n)].$$

- O presupunere des utilizată și care o vom aplica în acest moment este că ponderile $\mathbf{h}(n)$ și vectorul de intrare $\mathbf{x}(n)$ sunt independente.

- Această presupunere este justificată oarecum prin faptul că $\mathbf{h}(n)$ este determinat numai de $\mathbf{x}(n-1)$, $\mathbf{x}(n-2)$, ... Ecuația anterioară devine:

$$E[\mathbf{h}(n+1)] = E[\mathbf{h}(n)] - \mu E[\mathbf{x}(n)\mathbf{x}^T(n)]E[\mathbf{h}(n)] + E[d(n)\mathbf{x}(n)].$$

Astfel obținem:

$$E[\mathbf{h}(n+1)] = [\mathbf{I} - \mu\mathbf{R}]E[\mathbf{h}(n)] + \mu\mathbf{p}.$$

- Relație recursivă de aceeași formă ca și cea pentru algoritmul celei mai rapide descreșteri; acolo s-a arătat că va converge către $\mathbf{h}_o$ dacă:

$$0 < \mu < \frac{2}{\lambda_{\max}},$$

unde λ_{max} este cea mai mare valoare proprie a matricii de corelație $\mathbf{R}$.

- Concluzionăm că și LMS va converge în medie către $\mathbf{h}_o$, dacă aceleași condiții sunt satisfăcute, adică:

$$\lim_{n\to\infty} E[\mathbf{h}(n)] = \mathbf{h}_o,$$

dacă:

$$0 < \mu < \frac{2}{\lambda_{\max}}.$$

- Pentru o aproximare a lui μ, reamintim că suma valorilor proprii a unei matrici este egală cu urma sa.

- Deoarece toate valorile proprii ale lui $\mathbf{R}$ sunt pozitive, rezultă că avem:

$$\lambda_{\max} < \mathrm{tr}(\mathbf{R}) = Nr(0) = N\sigma_x^2.$$

- Practic o margine pentru μ care garantează convergența în medie este:

$$0 < \mu < \frac{2}{N\sigma_x^2},$$

- Convergența în medie este doar un prim pas în examinarea comportării algoritmului LMS.

- Din $\mathbf{h}(n)$ converge în medie la $\mathbf{h}_o$ nu garantează că algoritmul LMS converge într-adevăr (varianța sa poate diverge).

- Eroarea medie pătratică este o funcție cuadratică de $\mathbf{h}(n)$

- Din faptul că $\mathbf{h}(n)$ converge în medie către $\mathbf{h}_o$, nu se garantează că J converge către $J_{\min}$.

- Pentru a rezolva această chestiune se scrie gradientul stochastic ca o sumă dintre adevăratul gradient și un termen zgomotos $\mathbf{N}(n)$, adică:

$$\hat{\nabla}_n = \nabla_n + \mathbf{N}(n) = 2\mathbf{R}\mathbf{h}(n) - 2\mathbf{p} + \mathbf{N}(n).$$

- Algoritmul LMS devine astfel:

$$\mathbf{h}(n+1) = \mathbf{h}(n) - \mu\mathbf{R}\mathbf{h}(n) + \mu\mathbf{p} - \frac{1}{2}\mu\mathbf{N}(n).$$

- Vom folosi din nou descompunerea $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H$ ca să definim schimbarea de coordonate $\mathbf{z}(n) = \mathbf{Q}^H[\mathbf{h}(n) - \mathbf{h}_o]$, care este echivalentă cu $\mathbf{h}(n) = \mathbf{Q}\mathbf{z}(n) + \mathbf{h}_o$, și recursia LMS devine:

$$\mathbf{Q}\mathbf{z}(n+1) + \mathbf{h}_o = \mathbf{Q}\mathbf{z}(n) + \mathbf{h}_o - \mu\mathbf{R}[\mathbf{Q}\mathbf{z}(n) + \mathbf{h}_o] + \mu\mathbf{p} - \frac{1}{2}\mu\mathbf{N}(n).$$

- Dacă colectăm termenii după $\mathbf{z}$ și ținem cont că $\mathbf{R}\mathbf{h}_o = \mathbf{p}$, obținem:

$$\mathbf{Q}\mathbf{z}(n+1) = [\mathbf{I} - \mu\mathbf{R}\mathbf{Q}]\mathbf{z}(n) - \frac{1}{2}\mu\mathbf{N}(n)$$

sau dacă înmulțim ambii termeni cu $\mathbf{Q}^H$ rezultă:

$$\mathbf{z}(n+1) = [\mathbf{I} - \mu\mathbf{\Lambda}]\mathbf{z}(n) - \frac{\mu}{2}\mathbf{Q}^H\mathbf{N}(n). \tag{7}$$

- Ecuația anterioară descrie evoluția vectorului ponderilor în spațiul vectorial al ponderilor, unde vectorul optim $\mathbf{h}_o$ corespunde originii.

- Precizăm că am arătat deja convergența în medie a lui $\mathbf{h}_o$, ceea ce însemnă că $\mathbf{z}(n)$ converge în medie către zero.

- Pentru a demonstra că și covarianța lui $\mathbf{h}(n)$ este mărginită, vom arăta că $\lim_{n\to\infty} E[\mathbf{z}(n)\mathbf{z}^T(n)] < M$, unde majorantul M trebuie determinat.

- Ecuația (7) în $\mathbf{z}(n)$ reprezintă o ecuație de stare într-un sistem N-dimensional, unde $\mathbf{N}(n)$ este zgomotul de intrare N-dimensional.

- Să amintim un rezultat din staționaritatea asimptotică a sistemului cu variabile de stare:

Teorema 1. *Fie sistemul:* $\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{B}\mathbf{u}(n)$ *și presupunând că matricea* $\mathbf{A}$ *este stabilă și că* $\mathbf{u}(n)$ *este un zgomot alb de intrare cu medie zero și matrice de corelație* $\sum$*, atunci* $\mathbf{x}(n)$ *este staționar asimptotică cu medie zero și matrice de corelație* $\mathcal{K}$*, unde* $\mathcal{K}$ *este soluția unei ecuații matriciale Lyapunov:*

$$\mathcal{K} = \mathbf{A}\mathcal{K}\mathbf{A}^T + \mathbf{B}\sum\mathbf{B}^T.$$

- În cazul nostru $\mathbf{A} = \mathbf{I} - \mu\mathbf{\Lambda}$ și $\mathbf{B} = -\frac{\mu}{2}\mathbf{Q}^H$.

- μ este deja ales $0 < \mu < \frac{2}{\lambda_{\max}}$ astfel încât să se garanteze convergența în medie, deci matricea $\mathbf{I} - \mu\mathbf{\Lambda}$ este stabilă.

- În plus presupunem că vectorul zgomot $\mathbf{N}(n)$ este alb, o presupunere standard pentru o astfel de analiză.

- Astfel $\mathbf{z}(n)$ va fi asimptotic staționar cu medie zero și matrice de corelație:

$$E[\mathbf{z}(n)\mathbf{z}^H(n)] = \mathcal{K},$$

unde $\mathcal{K}$ este soluția ecuației:

$$\mathcal{K} = (\mathbf{I} - \mu\mathbf{\Lambda})\mathcal{K}(\mathbf{I} - \mu\mathbf{\Lambda}) + \frac{\mu^2}{4}\mathbf{Q}^H E[\mathbf{N}(n)\mathbf{N}^T(n)]\mathbf{Q}. \tag{8}$$

- Trebuie să aflăm o expresie pentru matricea de corelație a lui $\mathbf{N}(n)$.

- Presupunem că algoritmul este foarte aproape de convergență :

$$\hat{\nabla}_n = \nabla_n + \mathbf{N}(n).$$

- Deoarece suntem aproape de soluția optimală, atunci $\nabla_n \approx 0$ și astfel avem că $\mathbf{N}(n) \approx \hat{\nabla}_n$. Dar:

$$\hat{\nabla}_n = 2[\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{h}(n) - 2d(n)\mathbf{x}(n)],$$

$$\hat{\nabla}_n = 2\mathbf{x}(n)[\mathbf{x}^T(n)\mathbf{h}(n) - d(n)] = -2e(n)\mathbf{x}(\mathbf{n}),$$

$$\mathbf{N}(n) \approx -2e(n)\mathbf{x}(n)$$

- Matricea de autocorelație a lui $\mathbf{N}(n)$ este dată de

$$E[\mathbf{N}(n)\mathbf{N}(n)] \approx 4E[e^2(n)\mathbf{x}(n)\mathbf{x}^T(n)].$$

- Principiul de ortogonalitate (la convergență): $E[e(n)\mathbf{x}(\mathbf{n})] = 0$.

- Avem nevoie de încă o aproximație dată de teorema de factorizare a momentelor gaussiene, care spune că dacă x_1, x_2, x_3, x_4 sunt eșantioane ale unor procese gaussiene de medie zero, atunci:

$$E[x_1x_2x_3x_4] = E[x_1x_2]E[x_3x_4] + E[x_1x_3]E[x_2x_4] + E[x_1x_4]E[x_2x_3].$$

- Deci să presupunem că $e(n)$ și $\mathbf{x}(n)$ sunt gaussiene și aplicând teorema anterioară plus principiul de ortogonalitate, avem:

$$E[\mathbf{N}(n)\mathbf{N}^T(n)] \approx 4E[e^2(n)]E[\mathbf{x}(n)\mathbf{x}^T(n)].$$

- Din nou, fiindcă ne aflăm aproape de convergență avem $E[e^2(n)] \approx J_{\min}$ și astfel ecuația anterioară devine $E[\mathbf{N}(n)\mathbf{N}^T(n)] \approx 4J_{\min}\mathbf{R}$.

- Astfel am obținut o expresie pentru matricea de corelație a zgomotului.

- Ecuația matricială (8) devine $\mathcal{K} = (\mathbf{I} - \mu\mathbf{\Lambda})\mathcal{K}(\mathbf{I} - \mu\mathbf{\Lambda}) + \mu^2 J_{\min}\mathbf{Q}^H\mathbf{R}\mathbf{Q}$.

- Deoarece $\mathbf{Q}^H\mathbf{R}\mathbf{Q} = \mathbf{\Lambda}$, rămâne $\mathcal{K} = (\mathbf{I} - \mu\mathbf{\Lambda})\mathcal{K}(\mathbf{I} - \mu\mathbf{\Lambda}) + \mu^2 J_{\min}\mathbf{\Lambda}$.

- Soluția acestei ecuații nu este evidentă.

- Reamintim că $\mathcal{K} = \lim_{n\to\infty} E[\mathbf{z}(n)\mathbf{z}^T(n)]$

- Rezultă că elementele diagonale ale lui $\mathcal{K}$ ne dau varianța elementelor obținute din ponderile transformate z_i, adică $k_{ii} = \lim_{n\to\infty} E[z_i^2(n)]$.

- Deoarece toate matricile din ecuația Lyapunov sunt diagonale, scriem direct ecuațiile pentru elementele k_{ii}:

$$k_{ii} = (1 - \mu\lambda_i)^2 k_{ii} + \mu^2 J_{\min}\lambda_i.$$

$$\Rightarrow \quad k_{ii} = \frac{\mu J_{\min}}{2 - \mu\lambda_i}.$$

- Pentru μ mic, varianța lui z_i poate fi făcută oricât de mică: $k_{ii} \approx \frac{\mu}{2} J_{\min}$.

- Examinăm acum comportarea erorii medii pătratice.
  Eroarea medie pătratică în exces este definită ca și distanța medie de la eroarea medie pătratică la $J_{\min}$.

$$J = J_{\min} + \mathbf{z}^T \mathbf{\Lambda} \mathbf{z} = J_{\min} + \sum_{i=1}^{N} \lambda_i z_i^2$$

$$E[J - J_{\min}] = \sum_{i=1}^{N} E[z_i^2] = \sum_{i=1}^{N} \lambda_i k_{ii}$$

$$E[J - J_{\min}] = \mu J_{\min} \sum_{i=1}^{N} \frac{\lambda_i}{2 - \mu\lambda_i}$$

- Pentru μ mic devine:

$$E[J - J_{\min}] = \frac{\mu J_{\min}}{2} \sum_{i=1}^{N} \lambda_i = \mu \frac{J_{\min} N}{2} \sigma_x^2.$$

- Prin urmare eroarea medie pătratică în exces este proporțională cu μ
- Acest fapt este în contradicție cu viteza de convergență în medie.

- Pentru valori mari ale pasului vom avea o viteză ridicată de convergență în comparație cu un pas mic. Dezavantajul în acest caz este o eroare medie în exces mare.

- Reciproca e valabilă. Astfel obținem problema clasică a LMS-ului.

- Dezadaptarea (misadjustment) M este definită ca și raportul dintre eroarea în exces și $J_{\min}$.

$$M = \mu \sum_{i=1}^{N} \frac{\lambda_i}{2 - \mu\lambda_i}$$

și care pentru μ mic poate fi aproximat prin:

$$M = \frac{\mu}{2} \sum_{i=1}^{N} \lambda_i = \frac{\mu N}{2} \sigma_x^2.$$

Modificări ale algoritmului LMS

- Au fost propuse îmbunătățiri ale algoritmului LMS

1. Normalised LMS (NLMS) - alegerea unui pas universal
2. Leaky LMS - evitarea singularității matricei de autocorelație
3. Block LMS - eficientiza LMS
4. Sign LMS - simplifica LMS
5. Variable Step-Size LMS - compromis viteza - dezadaptare

Algoritmul NLMS

- Dificultățile în proiectarea și implementarea LMS: selectarea pasului

- Procese staționare: algoritmul converge în medie dacă $0 < \mu < \frac{2}{\lambda_{max}}$

- Converge în medie pătratică dacă $0 < \mu < \frac{2}{\mathrm{tr}(\mathbf{R})}$

- $\mathbf{R}$ este în general necunoscută
  λ_{max} sau $\mathbf{R}$ trebuie estimate pentru a folosi majoranții anteriori

- Pentru procese staționare: $\mathrm{tr}(\mathbf{R}) = NE\{|x(n)|^2\}$

$$0 < \mu < \frac{2}{NE\{|x(n)|^2\}}$$

- $E\{|x(n)|^2\}$ este puterea procesului $x(n)$; estimată prin

$$\hat{E}\{|x(n)|^2\} = \frac{1}{N}\sum_{k=0}^{N}|x(n-k)|^2$$

- Acest estimator utilizează doar acele valori disponibile în memoria filtrului transversal, deci nu necesită memorie în plus

- Estimatorul anterior sugerează următoarea margine a pasului care să asigure convergența în medie pătratică

$$0 < \mu < \frac{2}{\mathbf{x}^T(n)\mathbf{x}(n)}$$

- O cale convenabilă de a incorpora această margine în filtrul adaptiv LMS este de a utiliza un pas variabil în domeniul timp

$$\mu(n) = \frac{\beta}{\mathbf{x}^T(n)\mathbf{x}(n)} = \frac{\beta}{||\mathbf{x}(n)||^2}$$

- $\beta =$ pas normalizat

- Rezultă algoritmul Normalised LMS (NLMS)

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\beta}{||\mathbf{x}(n)||^2} e(n)\mathbf{x}(n)$$

- Efectul normalizării prin $||\mathbf{x}(n)||^2$ este să altereze modulul, dar nu schimbă direcția estimată a vectorului gradient
- Algoritmul LMS converge în medie pătratică dacă $0 < \beta < 2$
- Corecția aplicată vectorului $\mathbf{h}(n)$ este proporțională cu vectorul $\mathbf{x}(n)$.
- Atunci când $\mathbf{x}(n)$ este mare, algoritmul LMS întâmpină probleme cu amplificarea gradientului zgomotos
- Normalizând pasul prin $||\mathbf{x}(n)||^2$, amplificarea zgomotului este diminuată

- Apare o problemă similară atunci când $||\mathbf{x}(n)||^2$ este prea mic
- Se modifică NLMS astfel ($\epsilon > 0$)

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\beta}{\epsilon + ||\mathbf{x}(n)||^2} e(n)\mathbf{x}(n)$$

- NLMS necesită calcule adiționale pentru evaluarea termenului $||\mathbf{x}(n)||^2$
- Calcul recursiv $||\mathbf{x}(n+1)||^2 = ||\mathbf{x}(n)||^2 + |x(n+1)|^2 - |x(n-N)|^2$
- Necesită în plus 2 operații de ridicare la pătrat, o adunare și o scădere

Algoritmul Leaky LMS

- Dacă procesul de intrare are o matrice de autocorelație cu valori proprii zero, filtrul adaptiv LMS are unul sau mai multe moduri de oscilație care nu pot fi controlate și atenuate.

- Dacă $\lambda_m = 0$, atunci $(1 - \lambda_m)^n$ nu poate descrește spre zero

- Deoarece aceste moduri pot fi instabile, este important să stabilizăm filtrul adaptiv LMS forțând aceste moduri la zero

- O soluție constă în introducerea unui coeficient de pierdere sau de dispersie γ (**Leakage**)

- Rezultă algoritmul Leakage LMS ($0 < \gamma \leq 1$)

$$\mathbf{h}(n+1) = (1 - \mu\gamma)\mathbf{h}(n) + \mu e(n)\mathbf{x}(n)$$

- Efectul coeficientului **leakage** este

1. de a forța coeficienții filtrului adaptiv către zero, dacă eroarea sau intrarea devin zero
2. de a forța orice mod de oscilație neatenuat către zero

- Rescriem recurența Leakage LMS pentru studiul convergenței

$$\mathbf{h}(n+1) = [\mathbf{I} - \mu\{\mathbf{x}^T\mathbf{x}(n) + \gamma\mathbf{I}\}]\mathbf{h}(n) + \mu d(n)\mathbf{x}(n)$$

$$E[\mathbf{h}(n+1)] = [\mathbf{I} - \mu(\mathbf{R} + \gamma\mathbf{I})]\mathbf{h}(n) + \mu\mathbf{r}_{dx}$$

- Ecuația recursivă de la Leakage LMS este identică cu cea de la la LMS, dacă $\mathbf{R}$ se înlocuiește cu $\mathbf{R} + \gamma\mathbf{I}$

- Echivalent, coeficientul **leakage** înseamnă un zgomot alb adăugat la $x(n)$

- Apare un coeficient γ pe diagonala principală a matricei de autocorelație

- Valorile proprii ale lui $\mathbf{R} + \gamma\mathbf{I}$ sunt $\lambda_m + \gamma$

- Cum $\lambda_m \geq 0$. nici unul dintre nodurile algoritmului Leaky LMS nu vor rămâne neatenuate

- Condițiile pentru pasul algoritmului Leaky LMS

$$0 < \mu < \frac{2}{\lambda_{\max} + \gamma}$$

- (-) Pentru procese staționare, regimul staționar va fi polarizat (deplasat)

- Dacă $\mathbf{h}(n)$ converge în medie, atunci

$$\lim_{n\to\infty} E[\mathbf{h}(n)] = [\mathbf{R} + \gamma\mathbf{I}]^{-1}\mathbf{r}_{dx}$$

- Coeficientul **leakage** introduce o polarizare a soluției de regim staționar

- Algoritmul Leaky LMS poate fi obținut minimizând funcția de cost

$$J_{\mathrm{LeakyLMS}}(n) = |e(n)|^2 + \gamma||\mathbf{h}(\mathbf{n})||^2$$

$$\nabla J_{\mathrm{LeakyLMS}}(n) = -e(n)\mathbf{x}(n) + \gamma\mathbf{h}(n)$$

$$\mathbf{h}(n+1) = \mathbf{h}(n) - \mu\nabla J_{\mathrm{LeakyLMS}}(n) = \mathbf{h}(n) - \mu\gamma\mathbf{h}(n) + \mu e(n)\mathbf{x}(n)$$

Algoritmul Block LMS

- Reducerea efortului de calcul (comunicații de viteză)
- Block LMS este identic cu LMS, exceptând faptul că actualizarea coeficienților se face la fiecare bloc cu L eșantioane
- Intre actualizări, coeficienții filtrului adaptiv sunt menținuți constanți
- Pentru fiecare valoare n care aparține unui bloc, ieșirea filtrului $y(n)$ și eroarea $e(n)$ se calculează utilizând coeficienții filtrului din acel bloc
- La sfârșitul blocului, coeficienții sunt actualizați utilizând o medie a celor L estimări de gradient din blocul respectiv

- Ecuațiile de adaptare ale coeficienților din blocul k sunt

$$\mathbf{h}_{(k+1))} = \mathbf{h}_{kL} + \mu \frac{1}{L} \sum_{l=0}^{L-1} e(kL+l)\mathbf{x}(kL+l)$$

$$y(kL+l) = \mathbf{h}_{kL}^T \mathbf{x}(kL+l),\ l = 0, 1, \ldots, L-1$$

$$e(kL+l) = d(kL+l) - y(kL+l),\ l = 0, 1, \ldots, L-1$$

- $y(kL+l)$ fiind convoluție, se preferă utilizarea FFT

Algoritmii Sign LMS

- Se aplică operatorul signum ori la eroarea $e(n)$, ori la datele de intrare $x(n)$, ori la amindouă

- Algoritmul Sign-Error LMS $\mathbf{h}(n+1) = \mathbf{h}(n) + \mu \mathrm{sgn}[e(n)]\mathbf{x}(n)$

- Sign-Error LMS = cuantizor cu două nivel aplicat asupra erorii
- Simplificare $\mu = 2^{-l}$: translațiile înlocuiesc multiplicările
- Utilizează un estimator zgomotos al gradientului
  Se modifică modulul corecției, nu alterează direcția
- Echivalent cu un LMS cu pas invers proporțional cu modulul erorii
- Sign-Error LMS minimizează funcția de cost $J_{\mathrm{S-E\ LMS}} = |e(n)|$

$$\frac{\partial |e(n)|}{\partial \mathbf{x}_m(n)} = \mathrm{sgn}[e(n)] \frac{\partial e(n)}{\partial \mathbf{x}_m(n)} = \mathrm{sgn}[e(n)] x(n-m)$$

- $\nabla J_{\mathrm{S-E\ LMS}} = \mathrm{sgn}[e(n)]\mathbf{x}(n)$
  Algoritmul Sign-Error LMS = LMAS (Least Mean Absolute Value)

- Algoritmul Sign-Data LMS $\mathbf{h}(n+1) = \mathbf{h}(n) + \mu[e(n)]\mathrm{sgn}[\mathbf{x}(n)]$

$$h_m(n+1) = h_m(n) + \mu e(n)\mathrm{sgn}[x_m(n)]$$

- Algoritmul Sign-Data LMS alterează direcția vectorului gradient
  Sign-Data LMS mai puțin robust, spre deosebire de Sign-Error LMS
  Sunt seturi de date care converg cu LMS, dar diverg cu Sign-Data LMS

- $\mathrm{sgn}x = x/|x| \Rightarrow$ Sign-Data LMS normalizează individual fiecare coeficient din vectorul gradient

$$h_m(n+1) = h_m(n) + \frac{\mu}{|x(n-m)|}e(n)x(n-m)$$

- Algoritmul Sign-Data LMS = LMS cu pas variabil în timp, diferit pentru fiecare coeficient din vectorul gradient

- Algoritmul Sign-Sign: se cuantizează atât eroarea, cât și data

$$h_m(n+1) = h_m(n) + \mu \mathrm{sgn}[e(n)]\mathrm{sgn}[x(n)]$$

- Coeficienții $h_m(n)$ sunt adaptați adăugând sau scăzând o constantă μ

- Pentru stabilitate, se introduce un termen *leakage* în ecuația de actualizare a ponderilor

$$\mathbf{h}(n+1) = (1-\mu\gamma)\mathbf{h}(n) + \mu \mathrm{sgn}[e(n)]\mathrm{sgn}\mathbf{x}(n)$$

- Sign-Sign Data este mai lent convergent decât LMS
  Sign-Sign Data are eroare in exces mai mare decât LMS

- Sign-Sign Data este simplu de implementat, popular, adoptat de CCITT ca standard ADPCM

Variable Step-Size LMS

- Selectarea pasului μ în algoritmul LMS este un compromis dintre viteza de convergență, eroarea medie pătratică în exces și abilitatea filtrului de a urmări în timp semnalele ce-și modifică statistica

- Ideal ar fi (**Gear-shifting**):

  1. Când adaptarea începe și $\mathbf{h}(n)$ este departe de soluția optimă, pasul μ ar trebui să fie mare pentru a mișca vectorul pondere spre soluția optimă
  2. Odată ce filtrul converge în medie către soluția staționară, pasul trebuie să scadă pentru a reduce eroarea medie în exces

- Utilizăm un pas variabil; trebuie respectat un set de reguli
  eroare minimă în exces + tracking

- **Algoritmul Variable Step-Size** (Harris, Chabries and Bishop)

$$h_m(n+1) = h_m(n) + \mu_m(n)e(n)x(n-m)$$

- Pasii $\mu_m(n)$ sunt ajustați independent pentru fiecare coeficient

- Se actualizează în funcție de viteza cu care gradientul estimat $\nabla e^2(n) = -2e(n)x(n-m)$ își schimbă semnul

- Semnul lui $\nabla e^2(n)$ se schimbă des când ne aflăm aproape de optim
  Dacă semnul lui $\nabla e^2(n)$ nu se schimbă des, suntem departe de optim

- În primul caz descreștem $\mu_m(n)$ cu o constantă
  În al doilea caz creștem $\mu_m(n)$ cu o constantă

- Pentru a asigura convergența în medie a algoritmului Variable-Step Size

$$\mu_{\min} \leq \mu_m(n) \leq \mu_{\max}$$

- **Algoritmul Variable-Step Size** (Kwong-Johnston)
  Același pas pentru toate ponderile ($\alpha = 0.97$, $\gamma = 48 \cdot 10^{-5}$)

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu(n)e(n)\mathbf{x}(n)$$
$$\mu(n+1) = \alpha\mu(n) + \gamma e^2(n); \quad \mu_{\min} \leq \mu(n) \leq \mu_{\max}$$

- **Algoritmul Variable-Step Size** (Aboulnasr-Mayyas) ($\beta = 0.99$)

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu(n)e(n)\mathbf{x}(n)$$
$$p(n) = \beta p(n-1) + (1-\beta)e(n)e(n-1)$$
$$\mu(n+1) = \alpha\mu(n) + \gamma p^2(n); \quad \mu_{\min} \leq \mu(n) \leq \mu_{\max}$$

Introducere RLS

- Metoda de filtrare RLS (Recursive Least Squares) este o metodă distinctă
- Filtrările adaptive bazate pe filtrul Wiener erau interesate în minimizarea după criteriul erorii mediei pătratice, care era dependent de statisticile semnalelor relevante în aplicatie.
- Metoda RLS de filtrare adaptivă este interesată mai degrabă în aflarea unei soluții exacte la problema de minimizare, care este independentă de statisticile semnalelor.
- RLS este o generalizare a metodei de estimare a celor mai mici pătrate.
- Problema standard a celor mai mici pătrate constă în aflarea vectorului de regresie multiplă care minimizează suma pătratelor erorilor dintre ieșirea modelului de regresie multiplă și răspunsul dorit.

- Obiectivul celor mai mici pătrate recursive este să includem în mod secvențial mai multe observații și să actualizăm modelul regresiilor multiple într-un mod recursiv, adică fără a rezolva explicit ecuațiile normale de fiecare dată când o nouă observație este adăugată.

- RLS posedă avantaje distincte în comparație cu LMS cu privire la viteza de convergență și dezadaptare (misadjustment).

- Revers: creșterea complexității calculelor. LMS are o complexitate de aproximativ N multiplicări pe ponderea actualizată (N=ordinul filtrului). Tehnicile directe RLS au o complexitate proporțională cu N^2.

- Un fapt pozitiv este că sunt disponibili algoritmi rapizi RLS care pot fi folosiți la implementarea RLS cu o complexitate de ordin N. Acești algoritmii sunt senzitivi la erorile aritmetice și alegerea condițiilor inițiale.

Metoda celor mai mici pătrate

- Problema liniară LS este:

1. Avem un număr de L observații $\{x(n)\}$ și o altă colecție de L răspunsuri dorite $\{d(n)\}$. Observațiile $\{x(n)\}$ sunt intrările, iar răspunsurile dorite $\{d(n)\}$ se consideră ieșiri.
2. Presupunem, de asemenea că există o relație liniară între intrări și ieșiri, adică $d(n)$ poate fi scris:

$$d(n) = \sum_{k=0}^{N-1} h_o(k)x(n-k) + e_o(n)$$

   unde N este ordinul modelului și $e_o(n)$ este un termen eroare.
3. Scopul metodei LS constă în estimarea parametrilor $h_o(n)$ ai modelului, date fiind observațiile $\{x(n)\}$ și răspunsul dorit $\{d(n)\}$.

- Metoda celor mai mici pătrate (Least Squares - LS) este un echivalent deterministic al filtrării Wiener.

- În timp ce filtrarea Wiener era bazată pe medieri pe ansamblu a semnalelor stochastice staționare în sens larg, metoda LS utilizează medieri în timp și vor rezulta diferite filtre optime pentru fiecare realizare a procesului stochastic.

- Pentru problema LS observațiile $\{x(n)\}$ sunt privite ca și date cunoscute, iar erorile $\{e_o(n)\}$ pot fi considerate ca erori de măsurare care măsoară discrepanța dintre ieșirea modelului și răspunsul dorit.

- O presupunere standard a metodei LS este că eroarea este un zgomot alb, de medie nulă $E[e_o(n)] = 0$, $E[e_o(n)e_o(m)] = \sigma^2\delta(n-m)$.

- Deoarece valorile $x(n)$ sunt cunoscute $\Rightarrow E[d(n)] = \sum_{k=0}^{N-1} h_o(k)x(n-k)$.

- Să considerăm acum că avem filtrul transversal de ordinul N și având ponderile $h(k)$, atunci ieșirea și eroarea vor fi

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k), \quad e(n) = d(n) - y(n) = d(n) - \sum_{k=0}^{N-1} h(k)x(n-k)$$

- Reamintim că încercăm să estimăm niște parametrii de model ipotetici $h_o(n)$ dintre intrare și răspunsul dorit.

- În alte cuvinte noi am dori să determinăm un set fix de parametri de model care să explice răspunsurile dorite pentru toți indicii de timp peste care observațiile sunt făcute.

- Obiectivul metodei LS va fi deci determinarea $h_o(n)$ care minimizează suma pătratelor erorilor.

- În consecință parametrii $h_o(n)$ sunt ficși și noi vrem să minimizăm indexul de performanță LS:

$$J_{\mathrm{LS}}(\mathbf{h}) = \sum_{n=N_a}^{N_b} e^2(n)$$

- Să observăm că indicii N_a și N_b nu au fost încă specificați.

- Alegerea diferită indicilor N_a și $N_b \Rightarrow$ criterii de performanță diferite.

- Dacă măsurătorile disponibile sunt $x(1), x(2), \ldots, x(L)$, atunci ponderile filtrului transversal sunt acoperite pentru prima dată cand $n = N$.

- În consecință, minimizând $J_{\mathrm{LS}}(\mathbf{h})$ cu $N_a < N$, am avea nevoie de presupuneri suplimentare (în mod obișnuit considerând nule datele respective) despre secvența $x(n)$ înainte ca măsurătorile să fie disponibile.

- Similar pentru $n = L$: ponderile filtrelor vor fi pentru ultima dată acoperite cu date măsurate. Minimizând $J_{\mathrm{LS}}(\mathbf{h})$ cu $N_b > L$ va necesita de asemenea presupuneri care trebuie făcute asupra valorilor $\{x(n)\}$.

- Prin urmare cel mai convenabil este să considerăm $N_a = N$ și $N_b = L$, astfel că vom avea următorul criteriu de performanță

$$J_{\mathrm{LS}}(\mathbf{h}) = \sum_{n=N}^{L} e^2(n)$$

care nu ne solicită sa facem nici un fel de presupunere asupra datelor înainte de momentul $n = 1$ și după $n = L$.

Soluția problemel LS

- Vectorul de intrare în filtru $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^T$
- Vectorul ponderilor este $\mathbf{h} = [h(0), h(1), \ldots, h(N-1)]^T$
- Ieșirea filtrului transversal este $y(n) = \mathbf{h}^T\mathbf{x}(n) = \mathbf{x}^T(n)\mathbf{h}$
- Eroarea va fi $e(n) = d(n) - y(n) = d(n) - \mathbf{h}^T\mathbf{x}(n)$
- Indexul de performanță este

$$J_{\mathrm{LS}}(\mathbf{h}) = \sum_{n=N}^{L} [d(n) - \mathbf{h}^T\mathbf{x}(n)]^2$$

unde $J_{\mathrm{LS}}(\mathbf{h})$ indică acum dependența explicită a indexului de performanță de ponderile $\mathbf{h}$.

- Reamintim că $\mathbf{h}$ este constant pe intervalul de minimizare.

$$J_{\mathrm{LS}}(\mathbf{h}) = \sum_{n=N}^{L} d^2(n) - 2\mathbf{h}^T \sum_{n=N}^{L} d(n)\mathbf{x}(n) + \mathbf{h}^T \left[\sum_{n=N}^{L} \mathbf{x}(n)\mathbf{x}^T(n) \right] \mathbf{h}$$

$$\hat{\mathbf{p}} \doteq \sum_{n=N}^{L} d(n)\mathbf{x}(n); \quad \hat{\mathbf{R}} \doteq \sum_{n=N}^{L} \mathbf{x}(n)\mathbf{x}^T(n)$$

$$J_{\mathrm{LS}}(\mathbf{h}) = \sum_{n=N}^{L} d^2(n) - 2\mathbf{h}^T\hat{\mathbf{p}} + \mathbf{h}^T\hat{\mathbf{R}}\mathbf{h}$$

- Trecem la minimizarea lui $J_{\mathrm{LS}}(\mathbf{h})$, adică să rezolvăm ecuația $\nabla(J_{\mathrm{LS}}(\mathbf{h})) = 0$, unde

$$\nabla(J_{\mathrm{LS}}(\mathbf{h})) = \frac{\partial J_{\mathrm{LS}}(\mathbf{h})}{\partial \mathbf{h}} = -2\hat{\mathbf{p}} + 2\hat{\mathbf{R}}\mathbf{h}$$

- Valoarea optimală a lui $\mathbf{h}$ se obține rezolvând sistemul de ecuații $\hat{\mathbf{R}}\mathbf{h} = \hat{\mathbf{p}}$

$$\Rightarrow J_{\min} = \sum_{n=N}^{L} d^2(n) - \mathbf{h}^T \hat{\mathbf{p}}$$

- Să notăm similaritatea acestei soluții cu problema de filtrare Wiener

- Cantitățile corespund acum mediilor în timp, spre deosebire de filtrarea Wiener când au corespuns mediilor în ansamblu.

Principiul ortogonalității pentru filtre LS

- Soluția problemei liniare LS are câteva proprietăți care sunt similare proprietăților de ortogonalitate pentru filtre Wiener.

- Să începem prin ortogonalitatea ieșirii și a vectorilor de eroare.

- Să reamintim că pentru filtrul optimizat în LS sens, avem $\mathbf{y}^T\mathbf{e} = 0$, adică

$$\sum_{n=N}^{L} y(n)e(n) = 0$$

- Această sumă poate fi considerată ca și o mediere în timp a intercorelației dintre seriile de timp $\{e(n)\}$ și $\{y(n)\}$ și ne spune că atunci când filtrul LS este optimizat, ieșirea și eroarea sunt ortogonale.

- Vom considera produsul matrice-vector $\mathbf{X}^T\mathbf{e} = \mathbf{X}^T\mathbf{Q}\mathbf{d}$.

$$\mathbf{Q} = \mathbf{I} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \Rightarrow \mathbf{X}^T\mathbf{Q} = \mathbf{X}^T - \mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T = 0$$

- $\mathbf{X}^T\mathbf{e} = 0$; Versiunea scalară a acestei ecuații este

$$\sum_{n=N}^{L} e(n)x(n-k), \;\; k = 0, 1, \ldots, N-1$$

- Pentru k fix, ecuația reprezintă o medie în timp a intercorelației dintre secvența de eroare și derivația a k-a a filtrului.

- Astfel, atunci când filtrul este optimizat în LS sens, seria de timp a erorilor $\{e(n)\}$ și cea de a k-a derivație a filtrului $\{x(n-k)\}$ sunt ortogonale pentru $k = 0, 1, \ldots, N-1$.

Proprietățile soluției LS

- Ne întoarcem la ecuațiile normale pentru problema LS.
- Să reamintim că avem sistemul de ecuații $\mathbf{X}^T\mathbf{X}\mathbf{h} = \mathbf{X}^T\mathbf{d}$.
- Dacă $\mathbf{X}^T\mathbf{X}$ este inversabilă, atunci soluția este dată de $\mathbf{h} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{d}$.
- Matricea $\mathbf{X}^T\mathbf{X}$ poate fi exprimată ca și

$$\mathbf{X}^T\mathbf{X} = \sum_{n=N}^{L} \mathbf{x}(n)\mathbf{x}^T(n)$$

- Prin urmare $\mathbf{X}^T\mathbf{X}$ este o medie temporală a matricei de autocorelație pentru seria de timp $\{x(n)\}$.

- $\mathbf{X}^T\mathbf{X}$ este simetrică și pozitiv semidefinită.

- Dacă $\{x(n)\}$ este staționară în sens larg, atunci pentru valori mari L avem

$$\mathbf{X}^T\mathbf{X} = \sum_{n=N}^{L} \mathbf{x}(n)\mathbf{x}^T(n) \approx (L - N + 1)\mathbf{R}$$

- În mod similar, dacă considerăm vectorul $\mathbf{X}^T\mathbf{d}$ avem

$$\mathbf{X}^T\mathbf{d} = \sum_{n=N}^{L} d(n)x(n)$$

care este o medie temporală a intercorelației dintre $d(n)$ și $x(n)$.

- Astfel dacă $\{x(n)\}$ și $\{d(n)\}$ sunt mixt staționare în sens larg

$$\mathbf{X}^T\mathbf{d} \approx (l - N + 1)\mathbf{p}$$

- Prin urmare, pentru serii temporale staționare în sens larg și L mare avem $\mathbf{h} \approx \mathbf{R}^{-1}\mathbf{p}$, care corespunde soluției Wiener.

- Să examinăm proprietățile statistice ale soluției LS $\mathbf{h}$

- Inițial s-a presupus că modelul pentru răspunsul dorit este

$$d(n) = \sum_{k=0}^{N-1} h_o(k)x(n-k) + e_o(n)$$

- Utilizând notații vectoriale, aceasta poate fi exprimat $\mathbf{d} = \mathbf{y}_o + \mathbf{e} = \mathbf{X}\mathbf{h}_o + \mathbf{e}_o$

- Am arătat că soluția LS este $\mathbf{h} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{d}$.

- Urmează că $\mathbf{h} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\mathbf{h}_o + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{e}_o$ și deci $\mathbf{h} = \mathbf{h}_o + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{e}_o$.

- Presupunerea făcută era ca seria de timp $\{e_o(n)\}$ este de medie nulă. Astfel $E[\mathbf{h}] = \mathbf{h}_o$ ceea ce înseamnă că estimarea LS este o estimare nedeplasată a lui $\mathbf{h}_o$.

- Să calculăm covarianța estimatorului LS. Avem:

$$\mathrm{cov}[\mathbf{h}] = E[(\mathbf{h} - \mathbf{h}_o)(\mathbf{h} - \mathbf{h}_o)^T] = E[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{e}_o\mathbf{e}_o^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}]$$

$$\mathrm{cov}[\mathbf{h}] = \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T E[\mathbf{e}_o\mathbf{e}_o^T]\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}$$

- Deoarece presupunem că seria de timp a erorilor $\{e_o(n)\}$ este albă cu varianță σ^2, avem $E[\mathbf{e}_o\mathbf{e}_o^T] = \sigma^2\mathbf{I}$, ceea ce ne dă $\mathrm{cov}[\mathbf{h}] = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$.

- Astfel am arătat că estimarea LS este nedeplasată și are o matrice de covarianță dată de $\sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$, unde σ^2 este puterea procesului de eroare.

- O proprietate importantă a estimării LS este aceea că ne dă cel mai bun estimator liniar și nedeplasat a vectorului $\mathbf{h}_o$.

- Dacă producem un alt estimator de forma $\hat{\mathbf{h}} = \mathbf{E}\mathbf{d}$ care este nedeplasat, acest estimator va avea o matrice de covarianță care satisface $\mathrm{cov}[\hat{\mathbf{h}}] \geq \mathrm{cov}[\mathbf{h}]$.

Problema recursivă LS. Soluția recursivă a ecuației normale LS

- Problema recursivă LS este o extensie a problemei LS studiată anterior

- Problema recursivă LS:
  La momentul de timp n, avem observațiile $x(1), x(2), \ldots x(n)$ și răspunsurile dorite $d(1), d(2), \ldots d(n)$
  Presupunem că vectorul pondere care rezolvă problema LS pentru aceste observații a fost calculat
  Obținem o măsurătoare nouă $x(n+1)$ și un răspuns nou dorit $d(n+1)$
  Vrem să actualizăm soluția anterioară LS utilizând noile date, fără a recalcula soluția LS cu toate datele, de la început

- Filtrul transversal are N derivații $\mathbf{h}(k) = [h_0(k), h_1(k), \ldots, h_{N-1}(k)]^T$

- Intrarea filtrului transversal $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^T$

- Răspunsul dorit la momentul n este $d(n)$, iar eroarea dintre ieșirea filtrului și răspunsul dorit este

$$e(n) = d(n) - \mathbf{h}^T(k)\mathbf{x}(n) = d(n) - \mathbf{x}^T(n)\mathbf{h}(k)$$

- Folosim doi indecși: k - pentru indexul ponderilor; n - pentru intrare și răspuns dorit

- Obiectivul RLS: la fiecare moment k de timp, dorim să realizăm un fel de regresii multiple ale intrării și ale răspunsului dorit, până la momentul de timp k

- $\mathbf{h}(k)$ este dependent de $\mathbf{x}(n)$ și $d(n)$, unde $n = 1, 2, \ldots, k$

- Determinăm vectorii $\mathbf{h}(k)$ care minimizează indexul de performanță

$$E(k) = \sum_{n=1}^{k} w^{k-n} e^2(n)$$

- $w \in \mathbb{R},\ 0 < W < 1$ = forgetting factor = factor de uitare
  w ne permite să ponderăm cele mai recente erori (aproape de timpul k) mult mai puternic decât erorile mai depărtate Avantajos când filtrul lucrează într-un mediu unde statisticile semnalelor variază în timp

$$E(k) = \sum_{n=1}^{k} w^{k-n} [d(n) - \mathbf{h}^T(k)\mathbf{x}(n)]^2 = \sum_{n=1}^{k} w^{k-n} d^2(n)$$
$$-2\mathbf{h}^T(k) \sum_{n=1}^{k} w^{k-n} d(n)\mathbf{x}(n) + \mathbf{h}^T(k) \left[\sum_{n=1}^{k} w^{k-n} \mathbf{x}(n)\mathbf{x}^T(n) \right] \mathbf{h}(k)$$

$$E_d(k) \doteq \sum_{n=1}^{k} w^{k-n} d^2(n); \quad \mathbf{p}(k) \doteq \sum_{n=1}^{k} w^{k-n} d(n)\mathbf{x}(n);$$

$$\mathbf{R}(k) \doteq \sum_{n=1}^{k} w^{k-n} \mathbf{x}(n)\mathbf{x}^T(n)$$

- $E_d(k)$ - energia ponderată a răspunsului dorit
  $\mathbf{p}(k)$ - estimator de intercorelație; $\mathbf{R}(k)$ - estimator de autocorelație

$$E(k) = E_d(k) - 2\mathbf{h}^T(k)\mathbf{p}(k) - \mathbf{h}^T(k)\mathbf{R}(k)\mathbf{h}(k)$$

- Minimizăm $E(k)$ setând gradientul zero

$$\nabla E(k) = \frac{\partial E(k)}{\partial \mathbf{h}(k)} = -2\mathbf{p}(k) + 2\mathbf{R}(k)\mathbf{h}(k)$$

- $\nabla E(k) = 0 \Rightarrow \mathbf{R}(k)\mathbf{h}(k) = \mathbf{p}(k) \Rightarrow \mathbf{h}(k) = \mathbf{R}^{-1}(k)\mathbf{p}(k) =$ soluția ecuației normale LS

- Când vectorul pondere $\mathbf{h}(k) = \mathbf{R}^{-1}(k)\mathbf{p}(k)$, energia erorii este minimizată

$$E_{\min}(k) = E_d(k) - \mathbf{h}^T(k)\mathbf{p}(k)$$

Soluția recursivă a ecuației normale LS

- Exprimăm $\mathbf{R}(k)$ și $\mathbf{p}(k)$ direct din definiții

$$\mathbf{R}(k) = w\mathbf{R}(k-1) + \mathbf{x}(k)\mathbf{x}^T(k); \quad \mathbf{p}(k) = w\mathbf{p}(k-1) + d(k)\mathbf{x}(k)$$

- Dorim să aflăm o expresie pentru $\mathbf{R}^{-1}(k)$ în funcție de $\mathbf{R}^{-1}(k-1)$ astfel încât ecuația normală să poată fi rezolvată la fiecare pas, fără a trebui să inversăm $\mathbf{R}(k)$ de fiecare dată

- Lema de inversare a matricei: Presupunem că $\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}\mathbf{C}^T$. Atunci $\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^T\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^T\mathbf{B}$

- Substituții în lema de inversare a matricei:

$$\mathbf{A} = \mathbf{R}(k), \quad \mathbf{B}^{-1} = w\mathbf{R}(k-1), \quad \mathbf{C} = \mathbf{x}(k), \quad \mathbf{D} = \mathbf{I}$$

$$\mathbf{R}^{-1}(k) = w^{-1}\mathbf{R}^{-1}(k-1) - \frac{w^{-1}\mathbf{R}^{-1}(k-1)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)}{w + \mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)} \tag{9}$$

- Ecuația (9) ne arată cum inversa lui $\mathbf{R}$ la momentul k poate fi calculată din inversa lui $\mathbf{R}$ la momentul $k-1$

- Avem nevoie de o valoare inițială $\mathbf{R}^{-1}(0)$ de la care să pornim ecuația recursivă; se alege $\mathbf{R}^{-1}(0) = \sigma^{-1}\mathbf{I}$

- Introducem vectorul câștig de adaptare $\mathbf{g}(k)$

$$\mathbf{g}(k) = \frac{\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}{w + \mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}$$

- (9) $\Leftrightarrow \mathbf{R}^{-1}(k) = w^{-1}\mathbf{R}^{-1}(k-1) - w^{-1}\mathbf{g}(k)\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)$

$$\mathbf{g}(k) = w^{-1}[\mathbf{R}^{-1}(k-1) - \mathbf{g}(k)\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)]\mathbf{x}(k)$$

$$\mathbf{g}(k) = \mathbf{R}^{-1}(k)\mathbf{x}(k) \tag{10}$$

Actualizarea ponderilor vectorului

- Dorim recurența pentru calculul lui $\mathbf{h}(k)$ în funcție de $\mathbf{h}(k-1)$

$$\mathbf{h}(k) = \mathbf{R}^{-1}(k)\mathbf{p}(k); \qquad \mathbf{p}(k) = w\mathbf{p}(k-1) + d(k)\mathbf{x}(k)$$

$$\mathbf{h}(k) = w\mathbf{R}^{-1}(k)\mathbf{p}(k-1) -$$
$$\mathbf{g}(k)\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{p}(k-1) + \mathbf{R}^{-1}(k)\mathbf{x}(k)d(k)$$

$$\mathbf{h}(k) = \mathbf{h}(k-1) + \mathbf{g}(k)[d(k) - \mathbf{h}^T(k-1)\mathbf{x}(k)]$$

- $\alpha(k) \doteq d(k) - \mathbf{h}^T(k-1)\mathbf{x}(k) =$ eroare de estimare *a priori*
  Ne dă eroarea dintre răspunsul dorit și ieșirea filtrului transversal înainte ca ponderea să fie actualizată de la $\mathbf{h}(k-1)$ la $\mathbf{h}(k)$

$$\alpha(k) = d(k) - \mathbf{h}^T(k-1)\mathbf{x}(k) \Rightarrow \mathbf{h}(k) = \mathbf{h}(k-1) + \mathbf{g}(k)\alpha(k)$$

- După ce vectorul ponderilor este actualizat, eroarea de estimare *a posteriori* poate fi calculată prin $e(k) = d(k) - \mathbf{h}(k)\mathbf{x}(k)$

Actualizarea energiei erorii

- Dorim recursia pentru calculul lui $E(k)$ în funcție de $E(k-1)$. Reamintim

$$E(k) = \sum_{n=1}^{k} w^{k-n} e^2(n); \quad E_d(k) = \sum_{n=1}^{k} w^{k-n} d^2(n)$$

$$E(k) = E_d(k) - \mathbf{p}^T(k)\mathbf{h}(k)$$
$$E_d(k) = wE_d(k-1) + d^2(k)$$
$$\mathbf{p}(k) = w\mathbf{p}(k-1) + d(k)\mathbf{x}(k)$$
$$\mathbf{h}(k) = \mathbf{h}(k-1) + \mathbf{g}(k)\alpha(k)$$

$\alpha(k) =$ eroare de estimare *a priori*; $\mathbf{g}(k) =$ vectorul câștig de adaptare

$$E(k) = wE_d(k-1) + d^2(k) - w\mathbf{p}^T(k-1)\mathbf{h}(k-1) - w\mathbf{p}^T(k-1)\mathbf{g}(k)\alpha(k)$$
$$-d(k)\mathbf{x}^T(k)\mathbf{h}(k-1) - d(k)\mathbf{x}^T(k)\mathbf{g}(k)\alpha(k)$$

- Din
  $E_d(k-1) - \mathbf{p}^T(k-1)\mathbf{h}(k-1) = E(k-1);$
  $d(k) - \mathbf{x}^T(k)\mathbf{h}(k-1) = \alpha(k) \Rightarrow$

$$E(k) = wE(k-1) + [d(k) - \mathbf{p}(k)\mathbf{g}(k)]\alpha(k)$$

- Din
  $\mathbf{R}(k)\mathbf{h}(k) = \mathbf{p}(k);$
  $\mathbf{g}(k) = \mathbf{R}^{-1}(k)\mathbf{x}(k) \Rightarrow \mathbf{p}^T(k)\mathbf{g}(k) = \mathbf{h}^T(k)\mathbf{x}(k)$

$$E(k) = wE(k-1) + [d(k) - \mathbf{h}^T(k)\mathbf{x}(k)]\alpha(k)$$

- $e(k) = d(k) - \mathbf{h}^T(k)\mathbf{x}(k) =$ eroarea de estimare *a posteriori*

$$E(k) = wE(k-1) + \alpha(k)e(k)$$

Algoritmul RLS. Performaţe RLS. Comparaţie cu LMS

- Algoritmul RLS poate fi descris astfel

$k = 0$: $\mathbf{R}^{-1}(0) = \sigma^{-1}\mathbf{I}$

$\forall k \geq 1$: Fiind date $\mathbf{x}(\mathbf{k})$, $d(k)$ se calculează

$$\mathbf{g}(k) = \frac{\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}{w + \mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}$$

$$\alpha(k) = d(k) - \mathbf{h}^T(k-1)\mathbf{x}(k)$$

$$\mathbf{h}(k) = \mathbf{h}(k-1) + \mathbf{g}(k)\alpha(k)$$

$$\mathbf{R}^{-1}(k) = w^{-1}[\mathbf{R}^{-1}(k-1) - \mathbf{g}(k)\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)]$$

$$e(k) = d(k) - \mathbf{h}^T(k)\mathbf{x}(k)$$

$$E(k) = wE(k-1) + \alpha(k)e(k)$$

- Pentru fiecare k, exprimăm secvența eroare de estimare *a priori*

$$\begin{bmatrix} \alpha(1) \\ \alpha(2) \\ \vdots \\ \alpha(N) \\ \vdots \\ \alpha(k) \end{bmatrix} = \begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \\ \vdots \\ d(k) \end{bmatrix} - \begin{bmatrix} x(1) & 0 & 0 & \dots & 0 \\ x(2) & x(1) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x(N) & x(N-1) & \dots & \dots & x(1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x(k) & x(k-1) & \dots & \dots & x(k-N+1) \end{bmatrix} \times$$

$$\times [h_0(k)\ h_1(k)\ \dots\ h_{N-1}(k)]^T$$

- La fiecare k, ponderile $h_i(k)$ sunt calculate pentru minimizarea sumei pătratelor erorilor
  Dacă $k \leq N$, erorile pot fi identic zero, sistemul de ecuații fiind este subdeterminat. Începând cu $k = N + 1$ sistemul de ecuații devine supradeterminat și procedura LS poate începe

- Analiza algoritmului RLS pentru estimarea parametrilor LS
- Presupunem că răspunsul dorit este $d(k) = \mathbf{h}_o^T \mathbf{x}(k)$
- $\mathbf{h}_o^T$ = vectorul ponderilor (exact); $e_o(k)$ = secvența eroare de măsurare, de medie zero, proces alb și necorelat cu măsurătorile $x(k)$
- Soluția RLS la problema de estimare este $\mathbf{h}(k) = \mathbf{R}^{-1}(k)\mathbf{p}(k)$

$$\mathbf{R}(k) = \sum_{n=1}^{k} w^{k-n}\mathbf{x}(n)\mathbf{x}^T(n); \quad \mathbf{p}(k) = \sum_{n=1}^{k} w^{k-n} d(n)\mathbf{x}(n)$$

$$h(k) = \mathbf{R}^{-1}(k)\left[\sum_{n=1}^{k} w^{k-n}\mathbf{x}(n)\mathbf{x}^T(n)\right]\mathbf{h}_o + \mathbf{R}^{-1}(k)\left[\sum_{n=1}^{k}\mathbf{x}(n)e_o(n)\right]$$

$$h(k) = \mathbf{h}_o + \mathbf{R}^{-1}(k)\left[\sum_{n=1}^{k}\mathbf{x}(n)e_o(n)\right]$$

- Aplicăm expectația, utilizând presupunerile despre secvența de eroare.

- $E[\mathbf{h}(k)] = \mathbf{h}_o \Rightarrow$ RLS produce un estimator nedeplasat al vectorului $\mathbf{h}_o$
  Nu utilizăm matricea exactă $\mathbf{R}(k)$, deoarece avem inițializarea $\mathbf{R}(0) = \mathbf{I}$.

- Se poate arăta că în acest caz avem un estimator asimptotic deplasat

$$E[\mathbf{h}(k)] = \mathbf{h}_o + \frac{1}{k}\mathbf{R}^{-1}\mathbf{h}_o$$

- La valori mari k, diferența nu este importantă

- Vectorul diferență $\Delta\mathbf{h}(k) = \mathbf{h}(k) - \mathbf{h}_o$

$$\Delta\mathbf{h}(k) = \mathbf{R}^{-1}(k)\left[\sum_{n=1}^{k}\mathbf{x}(n)e_o(n)\right]$$

- Covarianța $\Delta\mathbf{h}(k)$ este dată de $E[\Delta\mathbf{h}(k)(\Delta\mathbf{h}(k)^T)] =$

$$\mathbf{R}^{-1}(k)E\left[\sum_{n=1}^{k}\sum_{j=1}^{k} w^{k-n}w^{k-j}\mathbf{x}(n)\mathbf{x}^T(j)e_o(n)e_o(j)\right]\mathbf{R}^{-1}(k)$$

- $\{e_o(n)\}$ zgomot alb, cu putere σ^2

$$E[\Delta\mathbf{h}(k)(\Delta\mathbf{h}(k)^T)] = \sigma^2\mathbf{R}^{-1}(k)\left[\sum_{n=1}^{k} w^{2(k-n)}\mathbf{x}(n)\mathbf{x}^T(n)\right]\mathbf{R}^{-1}(k)$$

$$w = 1 \Rightarrow E[\Delta\mathbf{h}(k)(\Delta\mathbf{h}(k)^T)] = \sigma^2\mathbf{R}^{-1}(k)\mathbf{R}^{-1}(k)\mathbf{R}^{-1}(k) = \sigma^2\mathbf{R}^{-1}(k)$$

- Eroarea de estimare *a priori*

$$\alpha(k) = d(k) - \mathbf{h}^T(k-1)\mathbf{x}(k) = e_o(k) - \mathbf{x}(k)\Delta\mathbf{h}(k-1)$$

- $\{e_o(n)\}$ zgomot alb, cu putere σ^2

$$E[\alpha^2(k)] = \sigma^2 + \mathbf{x}^T(k)E[\Delta\mathbf{h}(k)(\Delta\mathbf{h}(k)^T)]\mathbf{x}(k) = \sigma^2 + \mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)$$

- Presupuneri suplimentare: dacă $\{x(n)\}$ este WSS, media în timp a matricei de autocorelație este aproximativ egală cu media pe ansamblu a autocorelației $\mathbf{R}(k) \approx k\mathbf{R}$; $\mathbf{R}^{-1}(k) \approx (1/k)\mathbf{R}^{-1}$

- tr(scalar) = scalar: $\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k) = \mathrm{tr}[\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)]$

$$\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA}) \Rightarrow \mathrm{tr}[\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)] = \mathrm{tr}[\mathbf{R}^{-1}(k-1)\mathbf{x}^T(k)\mathbf{x}(k)]$$

$$E[\alpha^2(k)] \approx \sigma^2 + \frac{\sigma^2}{k-1}\mathrm{tr}[\mathbf{R}^{-1}(k-1)\mathbf{x}^T(k)\mathbf{x}(k)]$$

- $E_R(k)$ = media reziduală medie = media pe ansamblu peste toate realizările $\{x(n)\}$

$$E_R(k) = \sigma^2 + \frac{\sigma^2}{k-1}\mathrm{tr}(\mathbf{I}) = \sigma^2\left[1 + \frac{N}{k-1}\right] \Rightarrow \lim_{k\to\infty} E_R(k) = \sigma^2$$

- RLS cu $w = 1$:

1. Eroarea minimă MSE pentru estimarea parametrilor $= \sigma^2$;
2. Dezadaptare = 0;
3. Convergența MSE independentă de împrăștierea valorilor proprii

$$w \neq 1 \Rightarrow \mathbf{R}(k) = \sum_{n=1}^{k} w^{k-n}\mathbf{x}(n)\mathbf{x}^(n) \approx \frac{1-w^k}{1-w}\mathbf{R}$$

$$E_R(k) = \sigma^2\left[1 + N\cdot\frac{1-w}{1+w}\cdot\frac{1+w^k}{1-w^k}\right]; \quad \lim_{k\to\infty} E_R(k) = \sigma^2\left[1 + N\cdot\frac{1-w}{1+w}\right]$$

- Dezadaptare (misadjustment) $M = \dfrac{E_R(\infty) - E_{\min}}{E_{\min}} = N \cdot \dfrac{1-w}{1+w}$

- $w =$ factor de uitare (forgetting factor); urmărește statisticile semnalelor

- Pentru $k \gg 1$, $\mathbf{h}(k) \to \mathbf{h}_o$, cu o constantă de timp $\approx \frac{1}{1-w}$

- Pentru $w \approx 1$ vom avea constantă mare de timp și dezadaptare mică

- Dacă statisticile semnalelor variază în timp, avem nevoie de o constantă de timp mică care să urmărească statisticile semnalului (**tracking**)
  $\Rightarrow w \searrow \Rightarrow$ Dezadaptarea crește cu $1 - w$

- Initializare RLS: $\delta \geq 0,01\sigma_x^2$, $x(-N) = \sqrt{w^{-N}, \delta}$
  $x(-N+1) = x(-N+2) = \ldots = x(0) = 0.$

Comparație RLS cu LMS

- Convergența la RLS $\approx$ un ordin de mărime mai rapidă decât LMS
- Spre deosebire de LMS, convergența la RLS nu depinde de împrăștierea valorilor proprii ale semnalului de la intrare
- RLS este superior LMS la convergență și adaptare
- La RLS efortul de calcul este mult mai mare ca la LMS
- Trebuie analizat compromisul dintre complexitatea de calcul și performanțele filtrării adaptive