

5. FILTRE ADAPTIVE BAZATE PE MINIMIZAREA ERORII MEDII PATRATICE

5. FILTRE ADAPTIVE BAZATE PE MINIMIZAREA ERORII MEDII PATRATICE

- funcția cost este eroarea medie pătratică, $J(n)$
- ecuațiile Wiener-Hopf nu oferă o soluție practică
 - complexitate aritmetică ridicată
 - necesită funcțiile de autocorelație
 - se dorește un algoritm recursiv

Minimizarea unei funcții reale de variabilă complexă

Teoremă

O funcție $f(\mathbf{z}, \mathbf{z}^*): \mathbb{C}^N \rightarrow \mathbb{R}$ are direcția de variație maximă dată de gradientul complex $\nabla_{\mathbf{z}^*}\{f(\mathbf{z}, \mathbf{z}^*)\}$.

$$\nabla_{\mathbf{z}} J = \left[\frac{\partial J}{\partial z_0}, \frac{\partial J}{\partial z_1}, \dots, \frac{\partial J}{\partial z_{N-1}} \right]^T$$

$$\frac{\partial J}{\partial z_i} = \frac{1}{2} \left(\frac{\partial J}{\partial a_i} - j \frac{\partial J}{\partial b_i} \right), \quad \frac{\partial J}{\partial z_i^*} = \frac{1}{2} \left(\frac{\partial J}{\partial a_i} + j \frac{\partial J}{\partial b_i} \right), \quad z_i = a_i + b_i$$

$$\nabla_{\mathbf{z}}\{\mathbf{a}^H \mathbf{z}\} = \mathbf{a}^*$$

$$\nabla_{\mathbf{z}^*}\{\mathbf{a}^H \mathbf{z}\} = \mathbf{0}$$

$$\nabla_{\mathbf{z}}\{\mathbf{z}^H \mathbf{a}\} = \mathbf{0}$$

$$\nabla_{\mathbf{z}^*}\{\mathbf{z}^H \mathbf{a}\} = \mathbf{a}$$

$$\nabla_{\mathbf{z}}\{\mathbf{z}^H \mathbf{A} \mathbf{z}\} = (\mathbf{A} \mathbf{z})^*$$

$$\nabla_{\mathbf{z}^*}\{\mathbf{z}^H \mathbf{A} \mathbf{z}\} = (\mathbf{A} \mathbf{z})$$

5.1 Metoda pantei descendente maxime (Steepest Descent - SD)

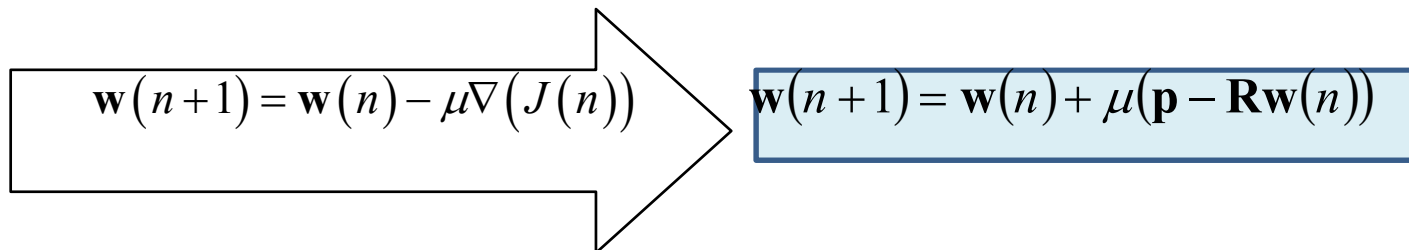
1. - Se pornește de la o valoare inițială a coeficienților $\mathbf{w}(0)$ (eventual $\mathbf{w}(0) = \mathbf{0}$).
2. - Se evaluează direcția pantei maxime de creștere în jurul acestui punct pe suprafața $J(\mathbf{w})$. Aceasta este exprimată prin gradientul complex $\nabla_{\mathbf{w}^*} (J(\mathbf{w}))$.
3. - Se reactualizează coeficienții printr-o deplasare pe direcția pantei descendente maxime. Aceasta este echivalent cu o deplasare pe direcția opusă gradientului cu un pas μ , $\mu \in \mathbf{R}_+$:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla (J(n))$$

4. - Se reia procedeul din punctul 2.

5.1 Metoda pantei descendente maxime (Steepest **D**escent - SD)

$$\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{N-1}(n)]$$
$$\nabla_{\mathbf{w}^*} (J(n)) = -\mathbf{p} + \mathbf{R}\mathbf{w}(n)$$



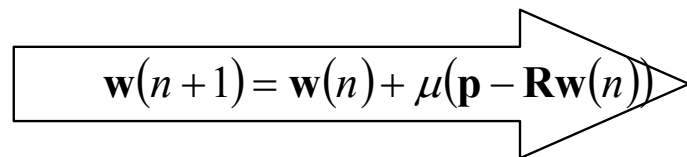
The diagram illustrates the derivation of the weight update equation for the steepest descent method. A large arrow points from the general update rule to the specific implementation.

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla (J(n))$$
$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu (\mathbf{p} - \mathbf{R}\mathbf{w}(n))$$

5.1 Metoda pantei descendente maxime (Steepest Descent - SD) *Analiza stabilității algoritmului*

Vom introduce *vectorul eroare a coeficienților*,

$$\mathbf{c}(n) = \mathbf{w}(n) - \mathbf{w}_o, \quad \mathbf{R}\mathbf{w}_o = \mathbf{p}$$


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

$$\mathbf{c}(n+1) = (\mathbf{I} - \mu\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H)\mathbf{c}(n)$$

5.1 Metoda pantei descendente maxime (Steepest Descent - SD)

Analiza stabilității algoritmului

$$\mathbf{c}(n+1) = (\mathbf{I} - \mu \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H) \mathbf{c}(n)$$

$$\mathbf{Q}^H \mathbf{c}(n+1) = (\mathbf{I} - \mu \mathbf{\Lambda}) \mathbf{Q}^H \mathbf{c}(n)$$

Vom introduce *vectorul eroare rotit*, $\mathbf{v}(n)$,

$$\mathbf{v}(n) = \mathbf{Q}^H \mathbf{c}(n) = \mathbf{Q}^H (\mathbf{w}(n) - \mathbf{w}_o)$$

$$\mathbf{Q}^H \mathbf{c}(n+1) = (\mathbf{I} - \mu \mathbf{\Lambda}) \mathbf{Q}^H \mathbf{c}(n) \quad \mathbf{v}(n+1) = (\mathbf{I} - \mu \mathbf{\Lambda}) \mathbf{v}(n)$$

5.1 Metoda pantei descendente maxime (Steepest Descent - SD)

Analiza stabilității algoritmului

Condiții inițiale:

$$\begin{aligned} \mathbf{v}(0) &= \mathbf{Q}^H (\mathbf{w}(0) - \mathbf{w}_o) \\ v_k(n+1) &= (1 - \mu\lambda_k)v_k(n), \quad k = 1, 2, \dots, N \\ v_k(n) &= (1 - \mu\lambda_k)^n v_k(0) \end{aligned}$$

Stabilitatea schemei, deci convergența algoritmului impune ca $v_k(n)$ să tindă la 0 când $n \rightarrow \infty$. Pentru aceasta este necesar și suficient ca:

$$|1 - \mu\lambda_k| < 1 \quad \text{sau} \quad 0 < \mu < \frac{2}{\lambda_k}, \quad k = 1, 2, \dots, N$$

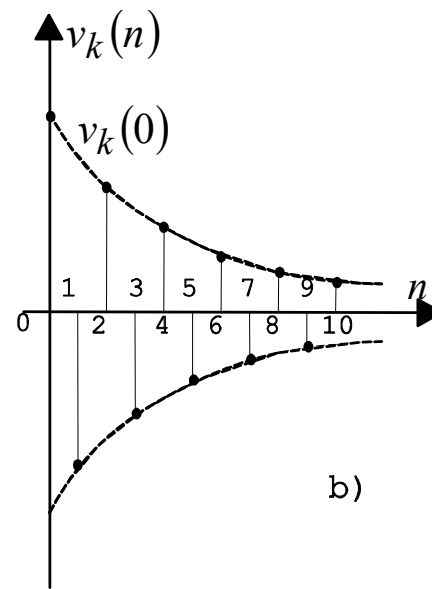
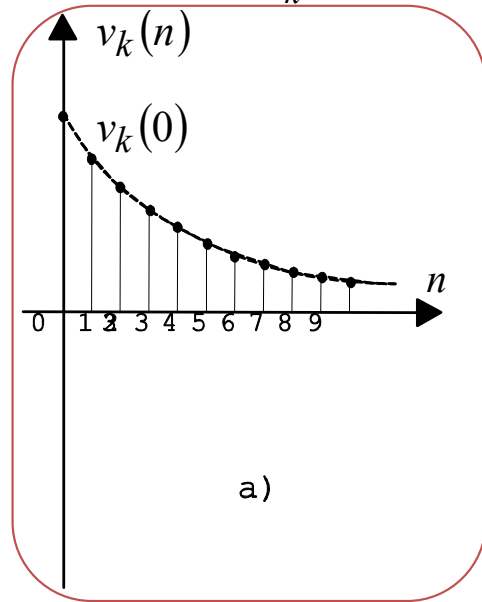
$$0 < \mu < \frac{2}{\lambda_{\max}}$$

5.1 Metoda pantei descendente maxime (Steepest Descent - SD)

Analiza convergenței

$\lambda_k \in \mathbf{R}_+$ $\Rightarrow v_k(n)$ va descrește către 0 fără oscilații. Distingem cazurile:

a) $0 < \mu\lambda_k < 1$ sau $0 < \mu < \frac{1}{\lambda_k} \Rightarrow v_k(n)$ descreșc uniform



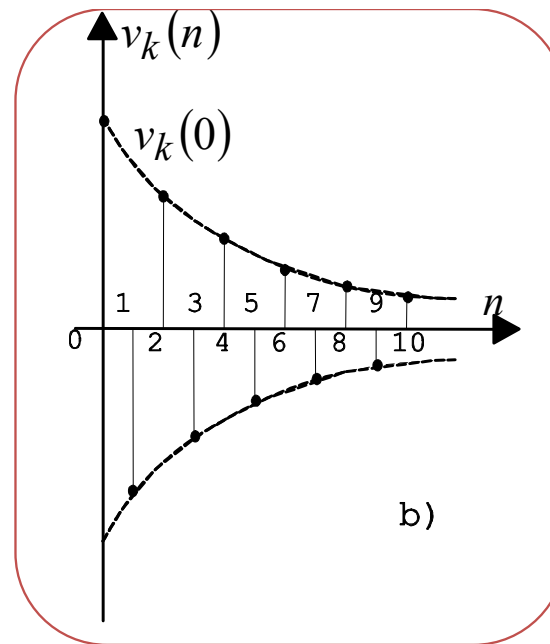
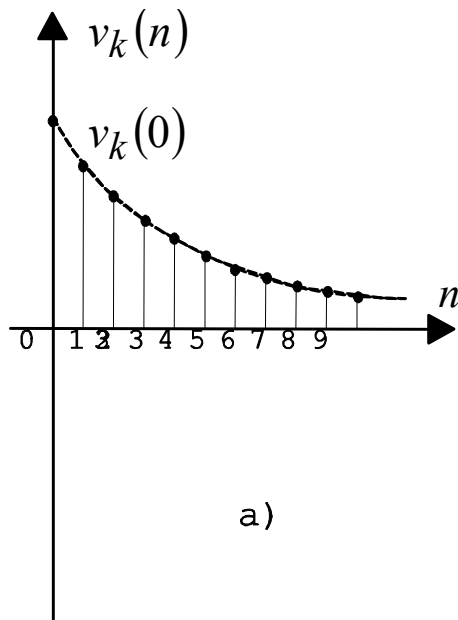
$$v_k(n) = e^{-n/\tau_k} v_k(0), \quad \tau_k = -\frac{1}{\ln(1 - \mu\lambda_k)}$$

5.1 Metoda pantei descendente maxime (Steepest Descent - SD)

Analiza convergenței

b) $-1 < 1 - \mu\lambda_k < 0$ sau $\frac{1}{\lambda_k} < \mu < \frac{2}{\lambda_k}$ - șir alternat

$$v_k(n) = e^{-n/\tau_k} (-1)^n v_k(0), \quad \tau_k = -\frac{1}{\ln(\mu\lambda_k - 1)}$$



5.1 Metoda pantei descendente maxime (Steepest Descent - SD)

Analiza convergenței

$$c) 1 - \mu\lambda_k = 0 \Rightarrow v_k(n) = 0, \quad n \geq 1$$

$$v_k(n) = e^{-n/\tau_k} s_k^n v_k(0),$$

$$\tau_k = -\frac{1}{\ln|1 - \mu\lambda_k|}, \quad s_k = \text{sgn}(1 - \mu\lambda_k)$$

Coeficienții $w_k(n)$ se pot exprima sub forma

$$\mathbf{w}(n) = \mathbf{w}_o + \mathbf{Q}\mathbf{v}(n) = \mathbf{w}_o + \sum_{k=1}^N \mathbf{q}_k v_k(n)$$

$$w_i(n) = w_{oi} + \sum_{k=1}^N \left(q_{ik} v_k(0) (1 - \mu\lambda_k)^n \right) = w_{oi} + \sum_{k=1}^N \left(q_{ik} v_k(0) s_k^n e^{-n/\tau_k} \right)$$

5.1 Metoda pantei descendente maxime (Steepest Descent - SD)

Analiza convergenței

$$\mathbf{w}(n) = \mathbf{w}_o + \mathbf{Q}\mathbf{v}(n) = \mathbf{w}_o + \sum_{k=1}^N \mathbf{q}_k v_k(n)$$

$$w_i(n) = w_{oi} + \sum_{k=1}^N \left(q_{ik} v_k(0) (1 - \mu \lambda_k)^n \right) = w_{oi} + \sum_{k=1}^N \left(q_{ik} v_k(0) s_k^n e^{-n/\tau_k} \right)$$

- Convergența coeficienților are loc după o sumă ponderată de exponențiale.
- Se obține viteza de convergență maximă atunci când $q_{ik} v_k[0]$ sunt nuli, pentru toți k , exceptând valoarea corespunzătoare lui λ_{Max} .
- Dacă nu sunt precizate condițiile inițiale, în cazul cel mai defavorabil, viteza de convergență este determinată de λ_{min} .

5.1 Metoda pantei descendente maxime (Steepest Descent - SD)

Analiza convergenței

$$\mu = r \frac{1}{\lambda_{\max}}, \quad 0 < r < 1$$

$$w_i(n) = w_{oi} + \sum_{k=1}^N \left(q_{ik} v_k(0) \left(1 - r \frac{\lambda_k}{\lambda_{\max}} \right)^n \right)$$

Termenul cel mai lent descrescător este acela care conține factorul

$$\left(1 - r \frac{\lambda_{\min}}{\lambda_{\max}} \right)^n$$

R rău condiționată (raport $\lambda_{\max}/\lambda_{\min}$ mare) \Rightarrow convergență lentă.

5.1 Metoda pantei descendente maxime (Steepest Descent - SD)

Curba de învățare

$$J(n) = E \left\{ e(n)^2 \right\}$$

$$e(n) = d(n) - \mathbf{w}^H(n) \mathbf{x}(n) = d(n) - \mathbf{w}_o^H(n) \mathbf{x}(n) - \mathbf{c}^H(n) \mathbf{x}(n) = e_o(n) - \mathbf{c}^H(n) \mathbf{x}(n)$$

$$J(n) = E \left\{ \left(e_o(n) - \mathbf{c}^H(n) \mathbf{x}(n) \right) \left(e_o^*(n) - \mathbf{x}^H(n) \mathbf{c}(n) \right) \right\} = E \left\{ e_o(n) e_o^*(n) \right\} - \\ - E \left\{ e_o(n) \mathbf{x}^H(n) \mathbf{c}(n) \right\} - E \left\{ \mathbf{c}^H(n) \mathbf{x}(n) e_o^*(n) \right\} + E \left\{ \mathbf{c}^H(n) \mathbf{x}(n) \mathbf{x}^H(n) \mathbf{c}(n) \right\}$$

$\mathbf{c}[n]$ este determinist, e_o satisface principiul ortogonalității \Rightarrow

$$J(n) = J_{\min} + \mathbf{c}^H(n) \mathbf{R} \mathbf{c}(n)$$

5.1 Metoda pantei descendente maxime (Steepest Descent - SD)

Curba de învățare

$$J(n) = J_{\min} + \mathbf{c}^H(n) \mathbf{R} \mathbf{c}(n)$$

sau

$$J(n) = J_{\min} + \mathbf{c}^H(n) \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H \mathbf{c}(n) = J_{\min} + \mathbf{v}^H(n) \mathbf{\Lambda} \mathbf{v}(n)$$

sau scalar:

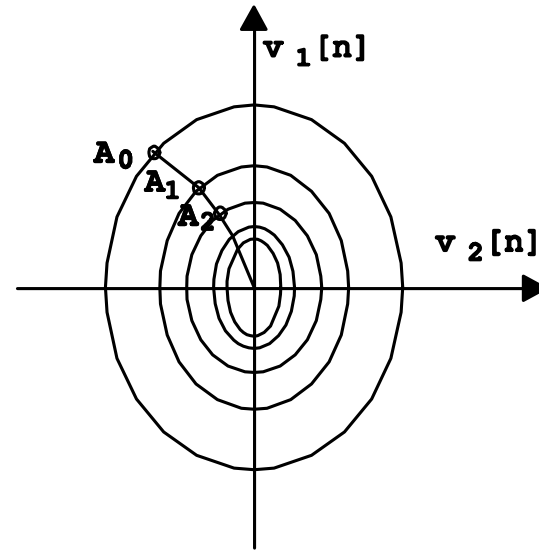
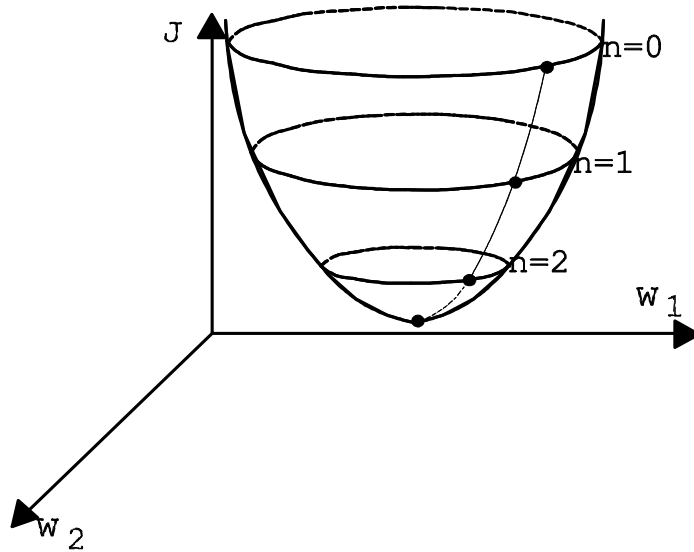
$$\begin{aligned} J(n) &= J_{\min} + \sum_{k=1}^N \lambda_k |v_k(n)|^2 = J_{\min} + \sum_{k=1}^N \lambda_k (1 - \mu \lambda_k)^{2n} |v_k(0)|^2 = \\ &= J_{\min} + \sum_{k=1}^N \lambda_k e^{-2n/\tau_k} |v_k(0)|^2 \end{aligned}$$

Curba obținută reprezentând $J(n)$ este *curba de învățare* a algoritmului. Indiferent de condițiile inițiale $v_k(0)$, eroarea medie pătratică tinde către J_{\min} dacă este îndeplinită condiția de stabilitate.

5.1 Metoda pantei descendente maxime (Steepest Descent - SD)

Curba de învățare

$$N = 2 \Rightarrow J(n) - J_{\min} = \lambda_1 v_1^2(n) + \lambda_2 v_2^2(n)$$



Intersecțiile cu plane $J = \text{const}$ sunt elipse cu semiaxele

$$a = \left(\frac{J(n) - J_{\min}}{\lambda_1} \right)^{1/2}, \quad b = \left(\frac{J(n) - J_{\min}}{\lambda_2} \right)^{1/2}$$

5.2 Algoritmul gradientului stohastic (LMS)

În cazul metodei SD ajustarea coeficienților se face pe baza gradientului erorii medii pătratice:

$$\nabla(J(n)) = -\mathbf{p} + \mathbf{R}\mathbf{w}(n); \quad \mathbf{R} = E\{\mathbf{x}(n)\mathbf{x}^H(n)\}; \quad \mathbf{p} = E\{\mathbf{x}(n)d^*(n)\}$$

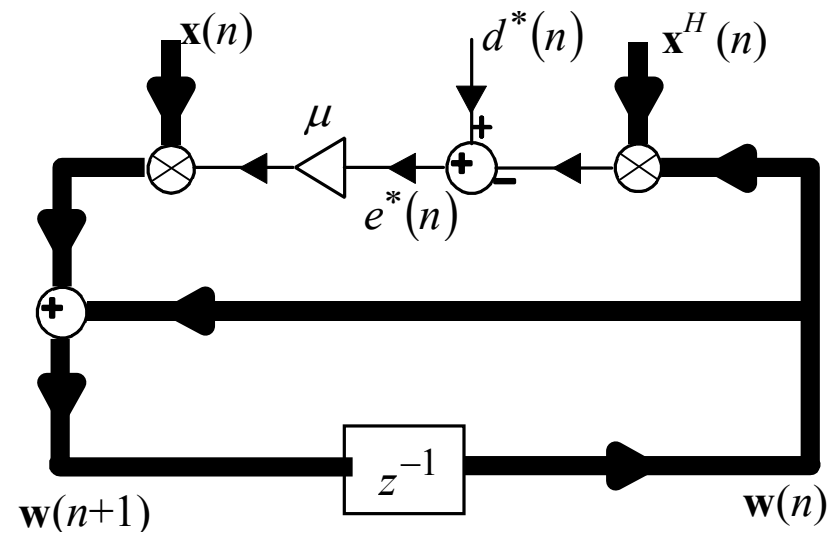
Mediile statistice în general nu sunt însă cunoscute. Se recurge la o estimare a gradientului utilizând niște valori estimate pentru cele două matrice renunțând la operațiile de mediere statistică:

$$\hat{\mathbf{R}}(n) = \mathbf{x}(n)\mathbf{x}^H(n); \quad \hat{\mathbf{p}}(n) = \mathbf{x}(n)d^*(n)$$
$$\hat{\nabla}(J(n)) = -\mathbf{x}(n)d^*(n) + \mathbf{x}(n)\mathbf{x}^H(n)\mathbf{w}(n) = -\mathbf{x}(n)e^*(n)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{x}(n)\left(d^*(n) - \mathbf{x}^H(n)\mathbf{w}(n)\right)$$
$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{x}(n)e^*(n)$$

5.2 Algoritmul gradientului stohastic (LMS)

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}(n) (d^*(n) - \mathbf{x}^H(n) \mathbf{w}(n))$$
$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}(n) e^*(n)$$



5.2 Algoritmul gradientului stohastic (LMS)

```

$$\mathbf{w}(0) = \mathbf{0}$$

$$\text{for } n = 0, 1, 2, \dots$$

$$y(n) = \mathbf{w}^H(n)\mathbf{x}(n)$$

$$e(n) = d(n) - y(n)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{x}(n)e^*(n)$$

$$\text{end}$$

```

5.2 Algoritmul gradientului stohastic (LMS)

Observații

- Complexitate aritmetică: $2N+1$ înmulțiri și $2N$ adunări pentru fiecare iterație
- Având în vedere criteriul de optimizare se întâlnește în limba engleză sub denumirea *Least Mean Square* (LMS).
- Fiind calculat de fiecare dată utilizând setul de eșantioane $\mathbf{x}[n]$, ce au un caracter aleator, fără a efectua o mediere, gradientul estimat va avea de asemenea un caracter aleator.
- Înmulțirile cu $\mathbf{x}[n]$ din schema algoritmului dau acestuia un caracter neliniar

5.2 Algoritmul gradientului stohastic (LMS)

Analiza convergenței

Probleme de convergență

- Tinde valoarea medie a vectorului $\mathbf{w}(n)$ către \mathbf{w}_0 atunci când $n \rightarrow \infty$? În caz afirmativ, se zice că algoritmul este *convergent în medie*.
- Tinde $J(n)$ către o valoare finită atunci când $n \rightarrow \infty$? In caz afirmativ se spune că algoritmul este *convergent în medie pătratică*.

5.2 Algoritmul gradientului stohastic (LMS)

Ipoteze de independență:

- $x(1), x(2), \dots, x(n)$ sunt statistic independenți.
- $\mathbf{x}(n)$ și $d(n)$ sunt statistic independente față de $d(1), \dots, d(n-1)$.
- vectorul de intrare $\mathbf{x}(n)$ și $d(n)$ formează împreună un set de variabile aleatoare gaussiene.

Datorită lipsei medierii în calculul gradientului apare un *zgomot de gradient*

$$\begin{aligned}\hat{\nabla}(J(n)) &= \nabla(J(n)) + \mathbf{N}(n) \\ \mathbf{N}(n) &= \hat{\nabla}(J(n)) - \mathbf{E}\{\hat{\nabla}(J(n))\} = -(\mathbf{x}(n)e^*(n) - \mathbf{E}\{\mathbf{x}(n)e^*(n)\}) \\ \mathbf{E}\{\mathbf{N}(n)\} &= \mathbf{0}\end{aligned}$$

5.2 Algoritmul gradientului stohastic (LMS)

Pe de altă parte, relația de reactualizare a coeficienților devine

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu(\nabla J(n) + \mathbf{N}(n)) = \mathbf{w}(n) + \mu(\mathbf{p} - \mathbf{R}\mathbf{w}(n)) - \mu\mathbf{N}(n)$$

$$\mathbf{w}_o + \mathbf{c}(n+1) = \mathbf{w}_o + \mathbf{c}(n) + \mu(\mathbf{p} - \mathbf{R}\mathbf{w}_o - \mathbf{R}\mathbf{c}(n)) - \mu\mathbf{N}(n)$$

$$\mathbf{c}(n+1) = (\mathbf{I} - \mu\mathbf{R})\mathbf{c}(n) - \mu\mathbf{N}(n)$$

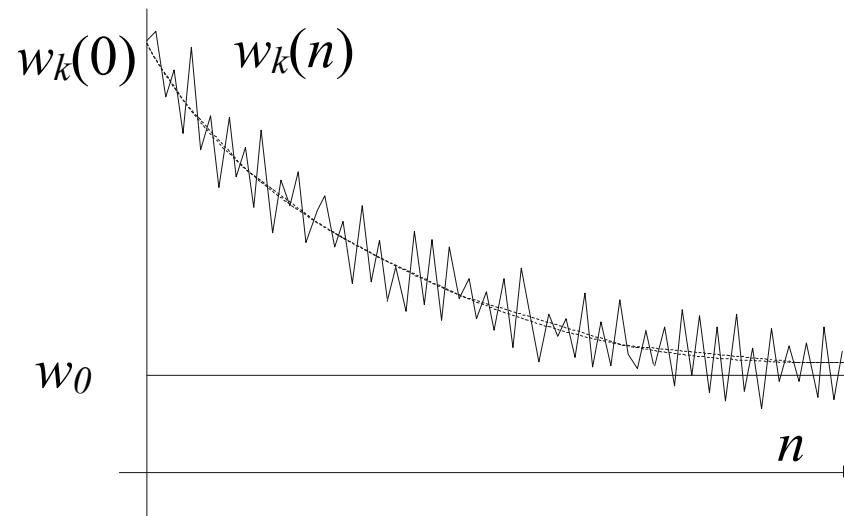
$$\mathbf{v}(n+1) = (\mathbf{I} - \mu\mathbf{\Lambda})\mathbf{v}(n) - \mu\mathbf{Q}^H\mathbf{N}(n)$$

$$E\{\mathbf{v}(n+1)\} = (\mathbf{I} - \mu\mathbf{\Lambda})E\{\mathbf{v}(n)\}$$

5.2 Algoritmul gradientului stohastic (LMS)

Având în vedere analogia formală dintre această ecuație și cea corespunzătoare în cazul algoritmului SD, concluziile trase acolo pentru vectorul eroare a coeficienților se pot transpune aici pentru media acestui vector. Algoritmul LMS este *convergent în medie* dacă:

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



5.2 Algoritmul gradientului stohastic (LMS)

Analiza convergenței în medie pătratică

$$\begin{aligned} J(n) &= J_{\min} + E\{\mathbf{c}^H(n)\mathbf{x}(n)\mathbf{x}^H(n)\mathbf{c}(n)\} \\ E\{\mathbf{c}^H(n)\mathbf{x}(n)\mathbf{x}^H(n)\mathbf{c}(n)\} &= E\{\text{tr}\{\mathbf{c}^H(n)\mathbf{x}(n)\mathbf{x}^H(n)\mathbf{c}(n)\}\} = E\{\text{tr}\{\mathbf{x}(n)\mathbf{x}^H(n)\mathbf{c}(n)\mathbf{c}^H(n)\}\} = \\ &= \text{tr}\{E\{\mathbf{x}(n)\mathbf{x}^H(n)\}E\{\mathbf{c}(n)\mathbf{c}^H(n)\}\} = \text{tr}\{\mathbf{R}\mathbf{C}(n)\} \\ \mathbf{C}(n) &= E\{\mathbf{c}(n)\mathbf{c}^H(n)\} \\ J(n) &= J_{\min} + \text{tr}\{\mathbf{R}\mathbf{C}(n)\} = J_{\min} + J_{ex}(n) \end{aligned}$$

5.2 Algoritmul gradientului stohastic (LMS)

$$J(n) = J_{\min} + \text{tr}\{\mathbf{R}\mathbf{C}(n)\} = J_{\min} + J_{ex}(n)$$

unde $J_{ex}(n)$ reprezintă o *eroare în exces*.

$$J(n) = J_{\min} + \text{tr}\{\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H\mathbf{C}(n)\} = J_{\min} + \text{tr}\{\mathbf{\Lambda}\mathbf{Q}^H\mathbf{C}(n)\mathbf{Q}\} = J_{\min} + \text{tr}\{\mathbf{\Lambda}\mathbf{K}(n)\}$$

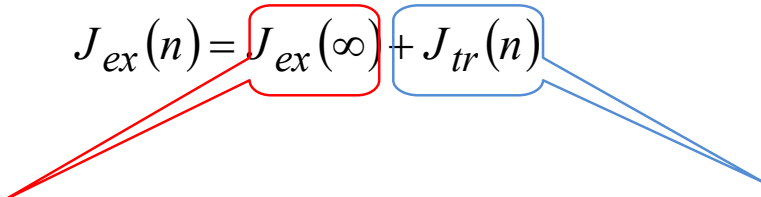
unde

$$\mathbf{K}(n) = \text{E}\{\mathbf{v}(n)\mathbf{v}^H(n)\} = \mathbf{Q}^H\mathbf{C}(n)\mathbf{Q}$$

$$J_{ex}(n) = \sum_{i=1}^N \lambda_i k_{ii}(n)$$

În raport cu filtrul optimal, apare deci o eroare medie pătratică suplimentară, sau în exces, notată cu J_{ex} , ce poate fi pusă pe seama zgomotului de gradient. Pentru evaluarea sa sunt necesari termenii de pe diagonala principală a matricei $\mathbf{K}(n)$.

5.2 Algoritmul gradientului stohastic (LMS)

$$J_{ex}(n) = J_{ex}(\infty) + J_{tr}(n)$$


Comp. staționară

Comp. tranzitorie

$$J_{ex}(\infty) = J_{\min} \frac{\sum_{i=1}^N \frac{\mu \lambda_i}{2 - \mu \lambda_i}}{1 - \sum_{i=1}^N \frac{\mu \lambda_i}{2 - \mu \lambda_i}}$$

5.2 Algoritmul gradientului stohastic (LMS)

Se definește *dezadaptarea* (misadjustment) prin

$$M = \frac{J_{ex}(\infty)}{J_{\min}} = \frac{\sum_{i=1}^N \frac{\mu\lambda_i}{2 - \mu\lambda_i}}{1 - \sum_{i=1}^N \frac{\mu\lambda_i}{2 - \mu\lambda_i}}$$

Pentru $\mu \ll \frac{2}{\lambda_{\max}}$

$$M \cong \frac{\mu}{2} \sum_{i=1}^N \lambda_i = \frac{\mu N}{2} r(0) = \frac{\mu N}{2} \lambda_{\text{med}}, \quad \text{unde} \quad \lambda_{\text{med}} = \frac{1}{N} \sum_{i=1}^N \lambda_i$$

5.2 Algoritmul gradientului stohastic (LMS)

Are în general o tendință de scădere cu micșorarea pasului μ , de creștere cu ordinul filtrului N , și e proporțional cu puterea medie a semnalului de intrare.

Componenta tranzitorie $J_{tr}(n) \rightarrow 0$ când $n \rightarrow \infty$ dacă

$$0 < \mu < \frac{2}{\lambda_{\max}} \quad \cap \quad \sum_{i=1}^N \frac{\mu \lambda_i}{2 - \mu \lambda_i} < 1$$

Dacă $0 < \mu \ll 1$ ambele condiții sunt îndeplinite dacă

$$0 < \mu < \frac{2}{\sum_{j=1}^N \lambda_j}$$

5.2 Algoritmul gradientului stohastic (LMS)

$$\sum_{j=1}^N \lambda_j = Nr(0), \quad r(0) = E\{x^*(n-k)x(n-k)\} = P_x, \quad k = 0, 1, \dots, N-1$$

P_x reprezintă **puterea secvenței** $x(n)$, deci o formă simplificată a condiției de convergență este:

$$0 < \mu < \frac{2}{NP_x}; \quad M = \frac{\mu N}{2} P_x$$

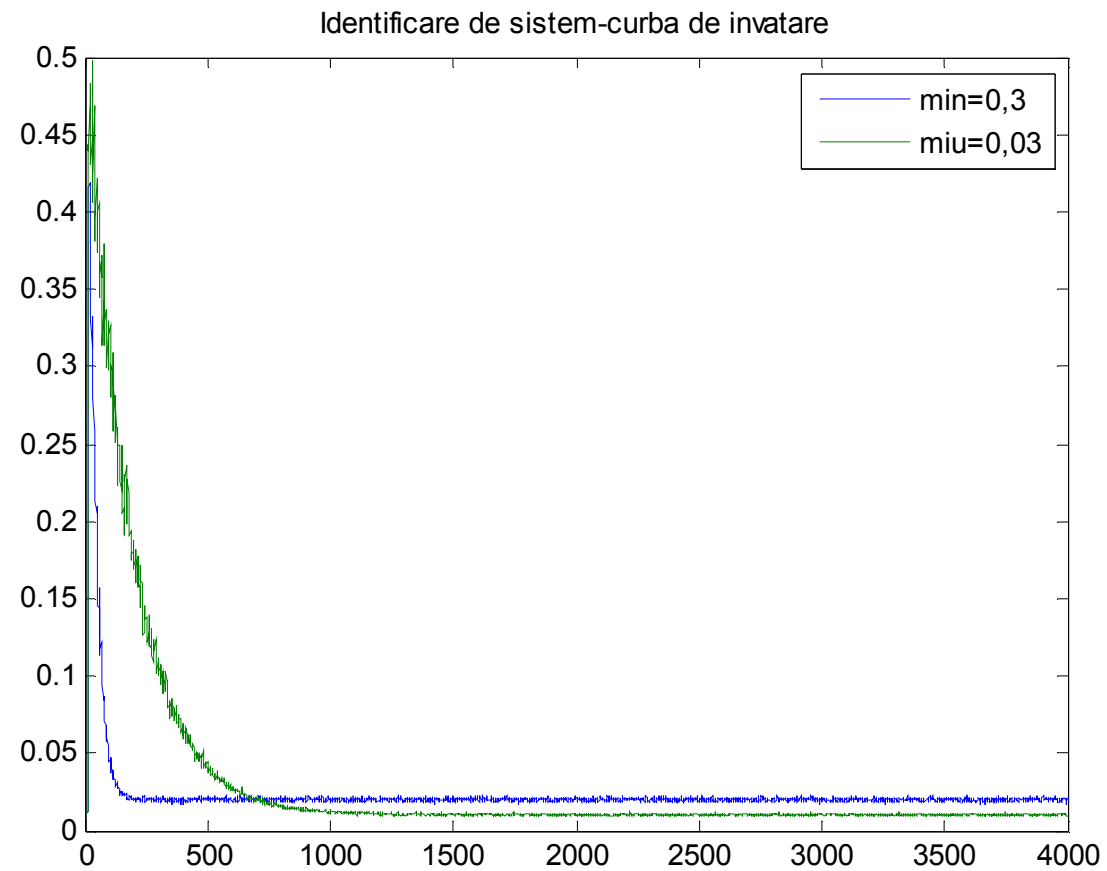
Se poate introduce o constantă de timp medie:

$$\tau_{\text{med}} = \frac{1}{2\mu\lambda_{\text{med}}}$$

care caracterizează viteza de scădere a părții tranzitorii a erorii.

Se constată că dacă μ este mic, constanta de timp e mare, conducând la o adaptare lentă, dar dezadaptarea este mică.

Exemplu – efectul valorii pasului μ



5.3. Metoda gradientului stohastic normalizat (Metoda normalizată a minimizării erorii medii pătratice - NLMS)

Poate fi privită ca o problemă de optimizare cu constrângeri. Ne propunem să determinăm noile valori $\mathbf{w}(n+1)$ ale coeficienților astfel încât să se minimizeze norma euclidiană a variației:

$$\delta\mathbf{w}(n+1) = \mathbf{w}(n+1) - \mathbf{w}(n)$$

cu condiția ca:

$$\mathbf{w}^H(n+1)\mathbf{x}(n) = d(n) \quad (1)$$

deci noii coeficienți să aibă acele valori care, cu un tact mai înainte, ar fi anulat eroarea.

Vom constitui *funcția cost* reală:

$$J(n) = \|\delta\mathbf{w}(n+1)\|^2 + \text{Re}\left\{\lambda\left[\mathbf{w}^H(n+1)\mathbf{x}(n) - d(n)\right]\right\}$$

care își atinge minimul odată cu $\|\delta\mathbf{w}(n+1)\|$ dacă este îndeplinită condiția (1)

**5.3. Metoda gradientului stohastic normalizat
(Metoda normalizată a minimizării erorii medii pătratice -
NLMS)**

$$\begin{aligned}
 J(n) &= \left(\mathbf{w}^H(n+1) - \mathbf{w}^H(n) \right) \left(\mathbf{w}(n+1) - \mathbf{w}(n) \right) + \\
 &+ \frac{1}{2} \left[\lambda \left(\mathbf{w}^H(n+1) \mathbf{x}(n) - d(n) \right) + \lambda^* \left(\mathbf{x}^H(n) \mathbf{w}(n+1) - d^*(n) \right) \right] = \\
 &= \mathbf{w}^H(n+1) \mathbf{w}(n+1) - \mathbf{w}^H(n+1) \mathbf{w}(n) - \mathbf{w}^H(n) \mathbf{w}(n+1) + \mathbf{w}^H(n) \mathbf{w}(n) + \\
 &+ \frac{1}{2} \left(\lambda \mathbf{w}^H(n+1) \mathbf{x}(n) + \lambda^* \mathbf{x}^H(n) \mathbf{w}(n+1) \right) - \frac{1}{2} \left(\lambda d(n) + \lambda^* d^*(n) \right)
 \end{aligned}$$

5.3. Metoda gradientului stohastic normalizat (Metoda normalizată a minimizării erorii medii pătratice - NLMS)

Pentru a găsi vectorul $\mathbf{w}(n+1)$ ce minimizează această expresie vom aplica metoda gradientului complex.

$$\begin{aligned}\nabla_{\mathbf{w}^*(n+1)} \left(\mathbf{w}^H(n+1) \mathbf{w}(n+1) \right) &= \mathbf{w}(n+1) \\ \nabla_{\mathbf{w}^*(n+1)} \left(\mathbf{w}^H(n+1) \mathbf{w}(n) + \mathbf{w}^H(n) \mathbf{w}(n+1) \right) &= \mathbf{w}(n) \\ \nabla_{\mathbf{w}^*(n+1)} \left(\lambda \mathbf{w}^H(n+1) \mathbf{x}(n) + \lambda^* \mathbf{x}^H(n) \mathbf{w}(n+1) \right) &= \lambda \mathbf{x}(n)\end{aligned}$$

Rezultă:

$$\begin{aligned}\nabla_{\mathbf{w}^*(n+1)} &= \mathbf{w}(n+1) - \mathbf{w}(n) + \frac{1}{2} \lambda \mathbf{x}(n) \\ \nabla_{\mathbf{w}^*(n+1)} = \mathbf{0} &\Rightarrow \mathbf{w}(n+1) - \mathbf{w}(n) = -\frac{1}{2} \lambda \mathbf{x}(n)\end{aligned}$$

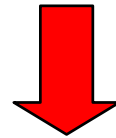
5.3. Metoda gradientului stohastic normalizat (Metoda normalizată a minimizării erorii medii pătratice - NLMS)

λ se obține punând condiția

$$\mathbf{w}^H(n+1)\mathbf{x}(n) = d(n)$$

Pentru aceasta se înmulțește la stânga cu $\mathbf{x}^H(n)$ relația:

$$\mathbf{w}(n+1) - \mathbf{w}(n) = -\frac{1}{2}\lambda\mathbf{x}(n)$$



$$\mathbf{x}^H(n)\mathbf{w}(n+1) - \mathbf{x}^H(n)\mathbf{w}(n) = -\frac{1}{2}\lambda\mathbf{x}^H(n)\mathbf{x}(n) = -\frac{1}{2}\lambda\|\mathbf{x}(n)\|^2$$

$$\lambda = -\frac{2}{\|\mathbf{x}(n)\|^2} \left(d^*(n) - \mathbf{x}^H(n)\mathbf{w}(n) \right) = -\frac{2}{\|\mathbf{x}(n)\|^2} e^*(n)$$

5.3. Metoda gradientului stohastic normalizat (Metoda normalizată a minimizării erorii medii pătratice - NLMS)

Noii coeficienți se vor calcula deci cu formula:

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

Se obișnuiește să se introducă o scalare a pasului cu o constantă $\bar{\mu}$, deci:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\bar{\mu}}{\|\mathbf{x}(n)\|^2} \mathbf{x}(n) e^*(n)$$

Poate fi echivalat cu algoritmul gradientului stohastic pentru:

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

în care *pasul este variabil*.

Convergența în medie pătratică este asigurată dacă:

$$0 < \bar{\mu} < 2$$

5.3. Metoda gradientului stohastic normalizat (Metoda normalizată a minimizării erorii medii pătratice - NLMS)

Evită prin normare *amplificarea zgomotului gradientului*, în expresia coeficienților $\mathbf{w}(n+1)$.

Apar în plus un număr de N înmulțiri și $N-1$ adunări la calculul fiecărui coeficient, ca urmare a necesității evaluării lui $\|\mathbf{x}(n)\|^2$. Eventual

$$\|\mathbf{x}(n)\|^2 = \sum_{k=0}^{N-1} |x(n-k)|^2 = \|\mathbf{x}(n-1)\|^2 + |x(n)|^2 - |x(n-N)|^2$$

Frecvent

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\bar{\mu}}{a + \|\mathbf{x}(n)\|^2} \mathbf{x}(n)e^*(n); \quad a > 0$$

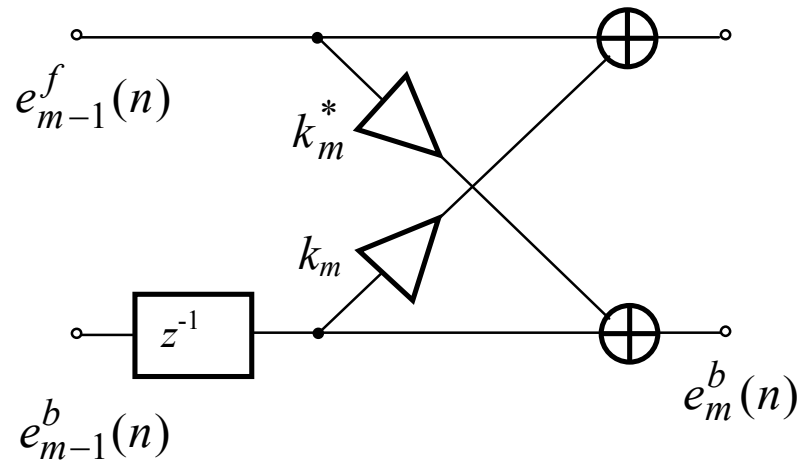
unde a este o constantă mică, adăugată pentru a evita împărțirea prin zero.

5.4 Metoda gradientului pentru structuri latice (GAL)

Ne punem, pentru început, problema proiectării unui *predictor latice adaptiv*.

Vor trebui deci optimizați coeficienții k_m iar funcția cost poate fi definită pornind de la eroarea de predicție directă, inversă, sau compusă.

Să considerăm o celulă m a filtrului și să determinăm k_m așa încât să fie minimizată eroarea medie pătratică de predicție, în forma compusă,



$$J_m(n) = E \left\{ \left| e_m^f(n) \right|^2 + \left| e_m^b(n) \right|^2 \right\}$$

5.4 Metoda gradientului pentru structuri latice (GAL)

Abordarea SD.

Conform metodei gradientului, va trebui luat $k_m(n+1)$,

$$k_m(n+1) = k_m(n) - \mu_m \nabla_m (J_m(n))$$

Va trebui deci evaluat gradientul complex în raport cu k_m^* :

$$\begin{aligned} \nabla_m (J_m(n)) = & \text{E} \left\{ \nabla_m \left(e_m^{f*}(n) \right) e_m^f(n) + e_m^{f*}(n) \nabla_m \left(e_m^f(n) \right) \right\} + \\ & + \text{E} \left\{ \nabla_m \left(e_m^{b*}(n) \right) e_m^b(n) + e_m^{b*}(n) \nabla_m \left(e_m^b(n) \right) \right\} \end{aligned}$$

Vom folosind relațiile de recurență:

$$\begin{aligned} e_m^f(n) &= e_{m-1}^f(n) + k_m e_{m-1}^b(n-1) \\ e_m^b(n) &= k_m^* e_{m-1}^f(n) + e_{m-1}^b(n-1) \\ k_m &= k_m^r + j k_m^i \end{aligned} \tag{2}$$

5.4 Metoda gradientului pentru structuri latice (GAL)

$$\nabla_m (e_m^{f*}(n)) = \nabla_{k_m^*} (e_{m-1}^{f*}(n) + k_m^* e_{m-1}^{b*}(n-1)) = e_{m-1}^{b*}(n-1)$$

$$\nabla_m (e_m^f(n)) = \nabla_{k_m^*} (e_{m-1}^f(n) + k_m e_{m-1}^b(n-1)) = 0$$

$$\nabla_m (e_m^b(n)) = \nabla_{k_m^*} (k_m^* e_{m-1}^f(n) + e_{m-1}^b(n-1)) = e_{m-1}^f(n)$$

$$\nabla_m (e_m^{b*}(n)) = \nabla_{k_m^*} (k_m e_{m-1}^{f*}(n) + e_{m-1}^{b*}(n-1)) = 0$$

$$\nabla_m (J(n)) = \mathbb{E} \{ e_m^f(n) e_{m-1}^{b*}(n-1) + e_m^{b*}(n) e_{m-1}^f(n) \}$$

5.4 Metoda gradientului pentru structuri latice (GAL)

Rezultă deci:

$$k_m(n+1) = k_m(n) - \mu_m \mathbb{E} \left\{ e_m^f(n) e_{m-1}^{b*}(n-1) + e_m^{b*}(n) e_{m-1}^f(n) \right\}$$

sau aplicând (2)

$$k_m(n+1) = k_m(n) \left[1 - \mu_m \mathbb{E} \left\{ |e_{m-1}^b(n-1)|^2 + |e_{m-1}^f(n)|^2 \right\} \right] - 2\mu_m \mathbb{E} \left\{ e_{m-1}^f(n) e_{m-1}^{b*}(n-1) \right\}$$

Condiția de stabilitate este:

$$\left| 1 - \mu_m \mathbb{E} \left\{ |e_{m-1}^b(n-1)|^2 + |e_{m-1}^f(n)|^2 \right\} \right| < 1$$

$$0 < \mu_m < \frac{2}{\mathbb{E} \left\{ |e_{m-1}^b(n-1)|^2 + |e_{m-1}^f(n)|^2 \right\}}$$

5.4 Metoda gradientului pentru structuri latice (GAL)

Abordarea LMS

Cum însă mediile statistice nu sunt în general cunoscute, se preferă de obicei utilizarea *gradientului stochastic*, renunțând la operațiile de mediere:

$$k_m(n+1) = k_m(n) - \mu_m E \left\{ e_m^f(n) e_{m-1}^{b^*}(n-1) + e_m^{b^*}(n) e_{m-1}^f(n) \right\}$$

$$k_m(n+1) = k_m(n) - \mu_m \left(e_m^f(n) e_{m-1}^{b^*}(n-1) + e_m^{b^*}(n) e_{m-1}^f(n) \right)$$

5.4 Metoda gradientului pentru structuri latice (GAL)

Abordarea NLMS

Se poate utiliza un algoritm normalizat, normând incrementul la energia erorii de predicție compusă, corespunzătoare intrărilor celulei m :

$$\mu_m = \mu_m(n) = \frac{\bar{\mu}}{W_{m-1}(n)}$$
$$W_{m-1}(n) = E \left\{ \left| e_{m-1}^f(n) \right|^2 + \left| e_{m-1}^b(n-1) \right|^2 \right\}$$

5.4 Metoda gradientului pentru structuri latice (GAL)

Aproximativ

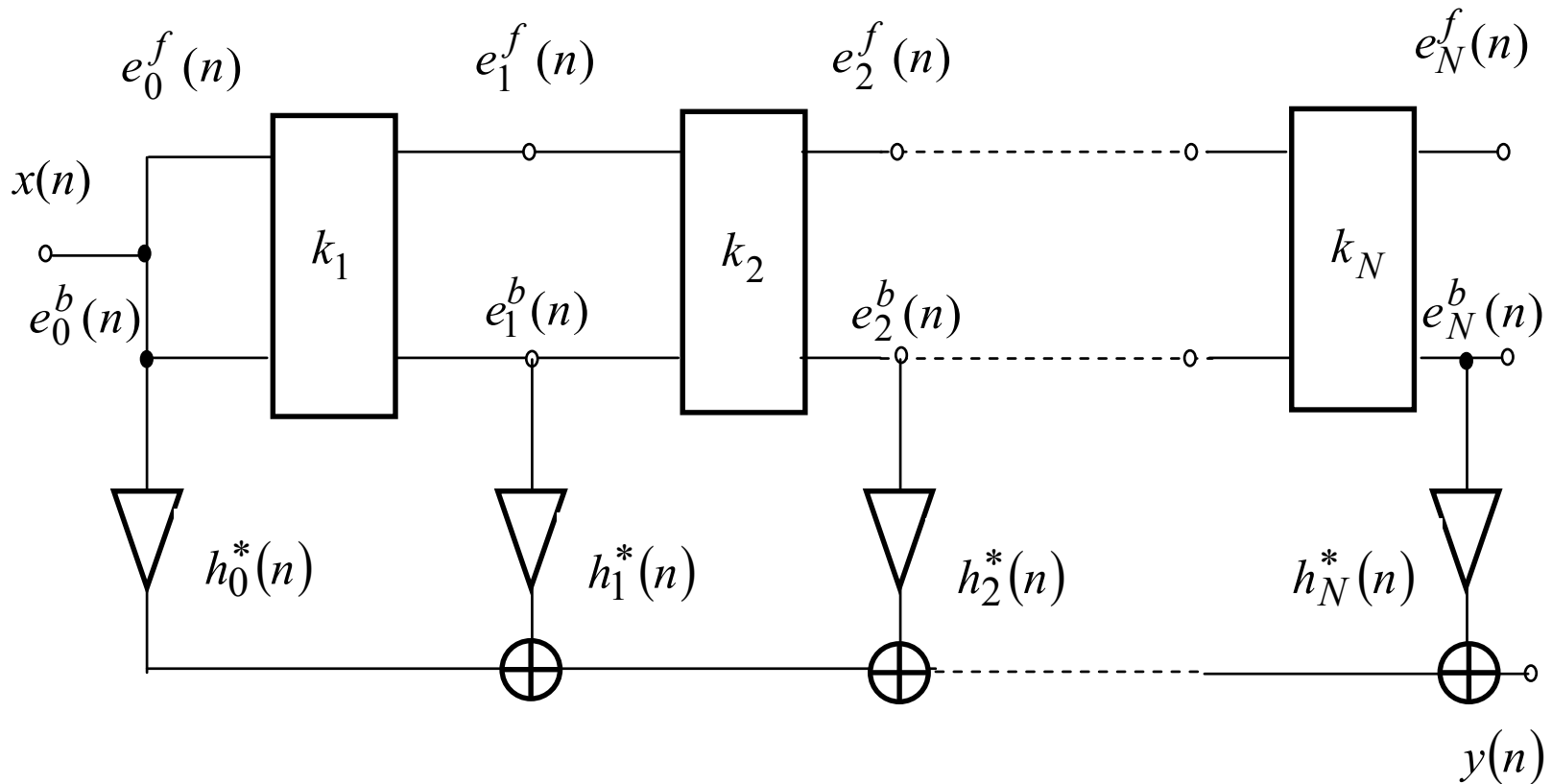
$$\begin{aligned} W_{m-1}(n) &\approx \frac{1}{n} \sum_{i=1}^n \left[|e_{m-1}^f(i)|^2 + |e_{m-1}^b(i-1)|^2 \right] = \\ &= \frac{1}{n} \left((n-1)W_{m-1}(n-1) + |e_{m-1}^f(n)|^2 + |e_{m-1}^b(n-1)|^2 \right) \end{aligned}$$

Uneori se introduce un parametru $0 < \beta < 1$, care permite alocarea unor ponderi diferite pentru eșantioanele precedente și cele actuale:

$$W_{m-1}(n) = \beta W_{m-1}(n-1) + (1 - \beta) \left(|e_{m-1}^f(n)|^2 + |e_{m-1}^b(n-1)|^2 \right)$$

5.5 Algoritm LMS pentru structuri latice-scară

Algoritm GAL permitea obținerea unui predictor. Pentru a obține un Filtru adaptiv se adaugă o structură în scară.



5.5 Algoritmul LMS pentru structuri latice-scară

Abordarea SD

Punem condiția ca ieșirea să estimeze cât mai corect un semnal dorit $d(n)$.

$$\mathbf{h}(n) = (h_0(n), h_1(n), \dots, h_N(n))^T$$

$$\mathbf{e}_N^b(n) = (e_o^b(n), e_1^b(n), \dots, e_N^b(n))^T$$

$$e(n) = d(n) - y(n) = d(n) - \mathbf{h}^H(n) \mathbf{e}_N^b(n)$$

Coeficienții k_i se calculează cu algoritmul GAL, iar h_i din condiția minimizării funcției cost

$$J(n) = E\{|e(n)|^2\}$$

5.5 Algoritmul LMS pentru structuri latice-scară

Conform ecuației Wiener-Hopf, coeficienții optimi sunt dați de

$$\mathbf{R}_e \mathbf{h}_o = \mathbf{r}_{ed}$$

unde

$$\mathbf{R}_e = E\left\{\mathbf{e}_N^b(n)\mathbf{e}_N^{bH}(n)\right\} \quad \mathbf{r}_{ed} = E\left\{\mathbf{e}_N^b(n)d^*(n)\right\}$$

În cazul coeficienților k_i optimi, ieșirile laticeii sunt ortogonale,

$$E\left\{e_r^b(n)e_l^{b*}(n)\right\} = \begin{cases} 0, & r \neq l \\ E\left(\left|e_r^b(n)\right|^2\right) = J_r^b, & r = l \end{cases}$$

$$\mathbf{R}_e = \text{diag}\left\{J_0^b, J_1^b, \dots, J_N^b\right\}$$

$$h_{o,k} = \frac{1}{J_k^b} E\left\{e_k^b(n)d^*(n)\right\}$$

5.5 Algoritmul LMS pentru structuri latice-scară

Pentru un algoritm recursiv, recurgem la metoda gradientului. În varianta SD:

$$\begin{aligned}\mathbf{h}(n+1) &= \mathbf{h}(n) - \mu \nabla J_N(n) \\ \nabla(J_N(n)) &= -\mathbf{r}_{ed} + \mathbf{R}_e \mathbf{h}(n) = \\ &= -E\{\mathbf{e}_N^b(n)d^*(n)\} + E\{\mathbf{e}_N^b(n)\mathbf{e}_N^{bH}(n)\}\mathbf{h}(n)\end{aligned}$$

Abordarea LMS

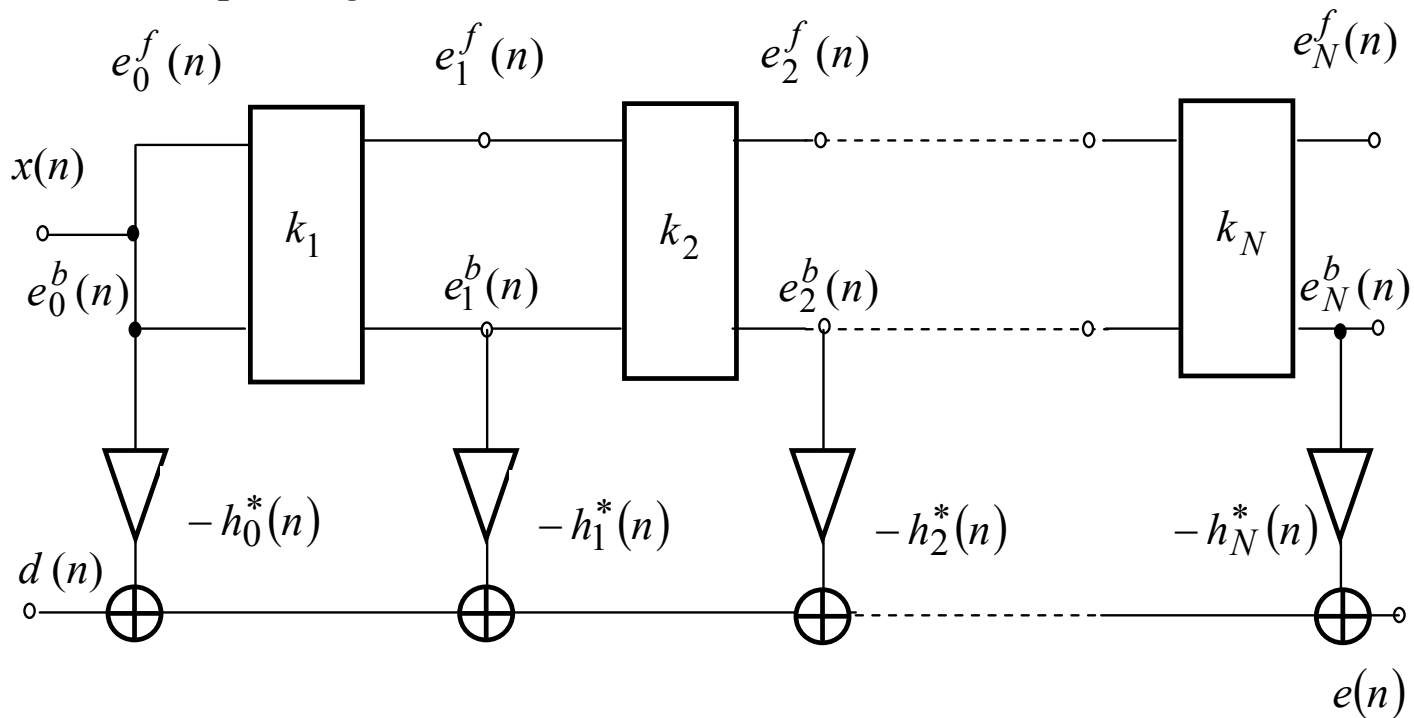
Cum valorile medii nu sunt în general cunoscute, aplicăm metoda gradientului stohastic (LMS),

$$\hat{\nabla}(J_N(n)) = -\mathbf{e}_N^b(n)d^*(n) + \mathbf{e}_N^b(n)\mathbf{e}_N^{bH}(n)\mathbf{h}(n) = -\mathbf{e}_N^b(n)e^*(n)$$

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu e^*(n)\mathbf{e}_N^b(n)$$

5.5 Algoritmul LMS pentru structuri latice-scară

Variantă pentru generarea erorii de estimare.



Avantaj: convergență mai rapidă, datorită decorelării datelor, de către structura latice.

Dezavantaj: complexitate aritmetică mai ridicată