

# Filtres adaptatius

José Antonio Morán Moreno  
Joan Claudi Socoró Carrié

PID\_00175651



*Els textos i imatges publicats en aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement-NoComercial-SenseObraDerivada (BY-NC-ND) v.3.0 Espanya de Creative Commons. Podeu copiar-los, distribuir-los i transmetre'ls públicament sempre que en citeu l'autor i la font (FUOC. Fundació per a la Universitat Oberta de Catalunya), no en feu un ús comercial i no en feu obra derivada. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.ca>*

# Índex

<b>1. Introducció al filtratge adaptatiu.....</b>	<b>5</b>
<b>2. Estructura general d'un filtre adaptatiu.....</b>	<b>7</b>
2.1. Identificació de sistemes .....	8
2.2. Equalització de canal .....	8
2.3. Predicció lineal .....	9
2.4. Cancel·lació de soroll .....	10
<b>3. El filtre LMS.....</b>	<b>11</b>
3.1. Estructura del mètode LMS .....	12
3.2. El mètode del gradient descendent .....	12
3.3. El mètode LMS .....	19
<b>4. Gram-Schmidt.....</b>	<b>22</b>
4.1. Anàlisi de la solució òptima per a $N = 2$ senyals .....	23
4.2. Anàlisi de la solució òptima per a $N = 3$ senyals .....	24
4.3. Solució general òptima .....	25
4.4. Solució adaptativa .....	26
4.4.1. Algorisme del gradient descendent i estudi de la seva convergència .....	26
4.4.2. Algorisme LMS .....	27
<b>5. RLS.....</b>	<b>31</b>
5.1. Solució òptima .....	31
5.2. Solució adaptativa .....	34
<b>Activitats.....</b>	<b>41</b>



## 1. Introducció al filtratge adaptatiu

En aquest mòdul ens introduïrem en el fascinant món del filtratge adaptatiu. La vida és dinàmica, i la capacitat de l'home per a adaptar-se marca els principis de la supervivència. En les aplicacions de processament de senyal els escenaris en què treballen aquests sistemes són també escenaris dinàmics que experimenten canvis al llarg del temps. No n'hi ha prou, doncs, de desenvolupar algorismes òptims des del punt de vista freqüencial o estadístic, sinó que resulta fonamental aconseguir que aquests sistemes puguin anar adaptant el seu comportament a les necessitats específiques del sistema per a cada moment.

El filtratge adaptatiu determina la teoria dels sistemes de filtratge que són capaços d'adaptar les seves condicions als canvis de les característiques de l'entorn. Tal com s'ha vist en el filtratge de Wiener, la solució òptima del filtre implica l'estimació d'uns paràmetres estadístics del sistema. L'estimació d'aquests paràmetres de correlació es pot fer de dues maneres:

### Vegeu també

Hem vist el filtratge de Wiener en l'apartat 5 del mòdul "Filtratge lineal òptim" d'aquesta assignatura.

- **Processament per blocs:** en aquest cas concret es determina una finestra de treball de  $N$  mostres en què es procedeix a fer l'estimació dels paràmetres i a resoldre la solució òptima del sistema per a aquest bloc. Una vegada finalitzat, si volem que el sistema tingui capacitat d'adaptació, hem de procedir a l'anàlisi del bloc de dades següent i a tornar a fer el càlcul de la solució òptima. Aquests mètodes són apropiats quan el nombre de dades que es processen és finit i es compleix una certa estacionarietat en les característiques dels senyals durant la durada del bloc.
- **Procediments de filtratge adaptatiu:** els filtres adaptatius són solucions apropiades quan no es disposa de bloc de dades per a analitzar i el sistema ha d'anar treballant en temps real, i especialment quan ens trobem en entorns poc estacionaris. Si les característiques dels senyals varien en el temps, l'única possibilitat que ens queda és recórrer a mètodes que disposin d'una capacitat d'adaptació i seguiment de l'entorn.

El nombre d'aplicacions en què l'ús del filtratge adaptatiu pot estar justificat és molt extens, sobretot quan es treballa en escenaris reals en què les característiques varien o evolucionen. Seguidament es mostren alguns exemples d'escenaris en què el filtratge adaptatiu és la solució òptima:

- **Cancel·lació de soroll en aplicacions de veu:** tal com es va comentar en el mòdul "Filtratge lineal òptim", els senyals de veu no són estacionaris. El patró estadístic depèn en gran mesura del tipus de fonemes de la locució, de manera que un sistema amb capacitat per a adaptar-se al patró de veu serà una solució excel·lent en aquest tipus d'aplicacions.

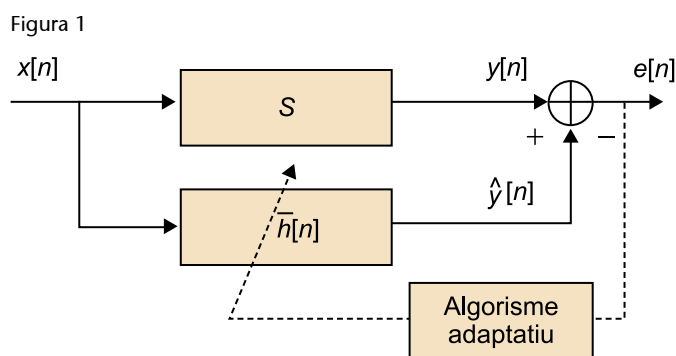
- **Cancel·lació d'interferències en antenes:** els entorns de comunicació avançats, com per exemple la telefonia mòbil, representen un repte per al dissenyador, pel fet que són molt variables i difícils de modelitzar. L'escenari de transmissió multicamí variant en el temps representa un repte en el disseny dels receptors. Els filtres adaptatius són solucions excel·lents per a reduir les interferències i equalitzar el canal en aquest tipus d'entorns.
- **Equalització adaptativa:** fins i tot en el cas dels canals més simples, és ben sabut per l'enginyer de telecomunicació que el senyal rebut experimenta distorsions a causa del canal de comunicació. Quan s'intenta aprofitar al màxim la capacitat del canal i arribar als límits de comunicació marcats per Shannon és necessari recórrer a sistemes avançats d'equalització de canal. Aquest tipus de sistemes permetran en bona mesura corregir l'efecte distorsionador del canal de comunicacions i reconstruir de la manera més òptima possible el senyal enviat.
- **Aplicacions biomèdiques:** el camp de la biomedicina ha experimentat grans avanços en els últims anys. L'anàlisi dels senyals biomèdics és un camp de gran interès per al diagnòstic i tractament de moltes malalties. El processament òptim és la base dels algorismes sobre els quals els equips biomèdics poden aïllar senyals i millorar-ne la qualitat per a obtenir resultats òptims en les proves.
- **Eliminació d'interferències en aplicacions d'imatge:** el processament òptim no sempre es fa sobre variacions en el domini temporal, sinó que pot ser molt útil també en aplicacions en el domini de l'espai, com per exemple en el cas d'imatges digitals. La imatge distorsionada presa sobre un objectiu en moviment es pot millorar en fer una estimació de l'efecte del moviment sobre la presa i equalitzar-lo per a obtenir-ne una imatge més nítida.

Com s'ha pogut observar, l'àmbit del filtratge adaptatiu i les seves aplicacions és un camp divers i útil en diferents disciplines. El processament de senyals ha entrat tan de ple en les aplicacions quotidianes que, a les nostres mans, quan manegem un telèfon intel·ligent, tenim un excel·lent exemple de proves d'aplicacions de processament de senyal adaptatives que s'utilitzen en un àmbit cada vegada més ampli.

## 2. Estructura general d'un filtre adaptatiu

En aquest apartat veurem a grans trets alguns dels escenaris en què s'apliquen els filtres adaptatius. Serà important interioritzar aquests escenaris per a facilitar la comprensió posterior dels conceptes involucrats en el desenvolupament d'aquest tipus de sistemes.

De manera genèrica podem dir que un filtre adaptatiu es pot representar per un esquema com el de la figura 1:



Tenim un sistema  $S$  que respon d'una determinada manera a un senyal d'entrada  $x[n]$ , que produeix una sortida  $y[n]$ . L'objectiu del filtre adaptatiu en aquest cas seria fer una estimació d'una resposta impulsional lineal que s'aproximi de manera òptima al comportament del sistema.

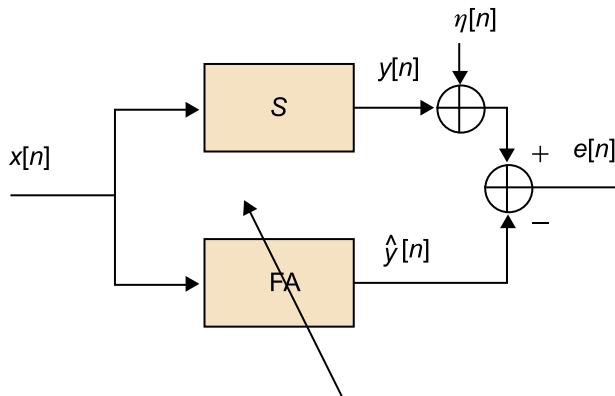
El sistema estarà d'entrada caracteritzat amb uns valors determinats abans d'iniciar-se el funcionament. En absència de més informació, es podria inicialitzar el sistema amb una resposta impulsional en forma de  $d[n]$  o amb tots els seus coeficients a zero. A mesura que van entrant mostres al sistema, es procedeix a avaluar el senyal d'error  $e[n]$ , i aquesta informació, convenientment utilitzada, serà la que alimentarà l'algorisme adaptatiu que, mostra per mostra, farà una modificació dels coeficients de la resposta impulsional fins a aconseguir establir-se en la solució òptima. Si l'escenari és estacionari, una vegada arribats a la solució òptima, el filtre tendirà a mantenir els seus coeficients constants. En canvi, en el cas d'escenaris variants en el temps, el filtre anirà modificant els seus coeficients per a seguir les variacions del sistema.

Seguidament veurem algunes aplicacions amb variacions en l'estructura del filtre per a diferents camps d'aplicació.

## 2.1. Identificació de sistemes

El primer esquema (figura 2) correspon a l'exemple base explicat anteriorment. Aquesta estructura serà útil quan tinguem la necessitat de modelitzar qualsevol tipus de sistema. Si imaginem, per exemple, que el sistema  $S$  és un motor, aquest esquema serviria per a proporcionar-nos un model lineal de comportament. Aquest model es podria utilitzar posteriorment per a obtenir un controlador òptim per a la velocitat de rotació del motor.

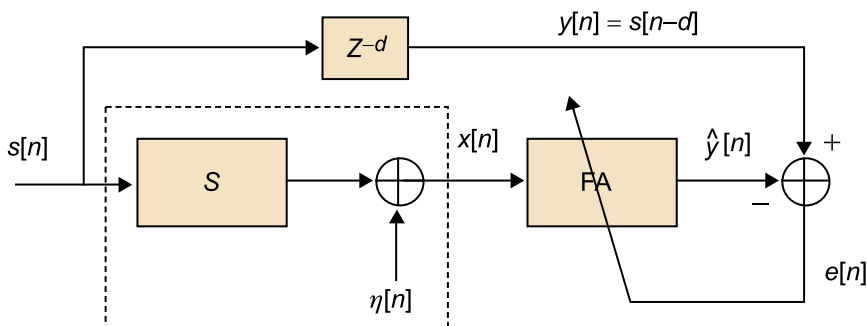
Figura 2



## 2.2. Equalització de canal

L'esquema que es mostra a la figura 3 correspon a una aplicació d'equalització de canal. Tot sistema de transmissió de dades busca tenir el canal òptim en la seva transmissió. El procés d'equalització de canal pretén desfer en la mesura del possible els efectes distorsionants del canal en la transmissió. El canal es pot modelitzar de manera simple com un sistema  $S$  que distorsiona el senyal d'entrada, més una addició d'un soroll interferent, tal com s'observa en la figura 3. El filtre adaptatiu (FA) intentarà desfer els efectes distorsionadors per a aconseguir obtenir a la sortida un senyal igual al que hi havia a l'entrada.

Figura 3





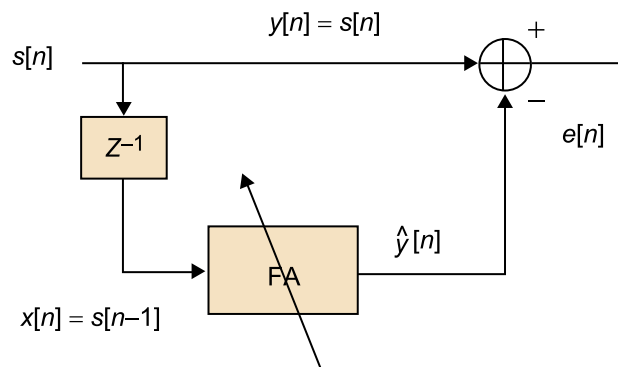
En aquest exemple concret és important remarcar la necessitat del retard del senyal de referència del filtre, que correspon al senyal d'entrada amb un retard de  $d$  mostres. Aquest retard és necessari per a compensar el retard de transmissió del canal, ja que en cas de no introduir-lo, al filtre adaptatiu li seria impossible reconstruir el senyal d'entrada, ja que hauria de ser un sistema no causal.

L'altra qüestió important d'aquest esquema és veure que per a fer l'entrenament és important que s'utilitzi una seqüència de dades coneguda pel receptor. En els sistemes d'equalització adaptativa, el protocol inclou una etapa d'entrenament (*training*) i una altra de seguiment (*tracking*). En la fase d'entrenament (o entrenament ràpid) el sistema utilitza una seqüència coneguda que li permet obtenir una aproximació inicial de qualitat del canal. Una vegada finalitzada aquesta fase, es procedeix a una etapa de seguiment, en què el sistema simplement segueix les petites variacions que sorgeixen a partir d'aquest moment.

### 2.3. Predicció lineal

Un sistema de predicció lineal és d'utilitat en un gran nombre d'aplicacions. La codificació de veu, l'evolució dels paràmetres financers o meteorològics, els compressors de dades i altres, són camps d'aplicació dels sistemes de predicció lineal. Per a l'home sempre ha estat important anticipar-se als esdeveniments futurs, i els sistemes de predicció són sistemes que permeten predir el valor futur d'una sèrie temporal.

Figura 4

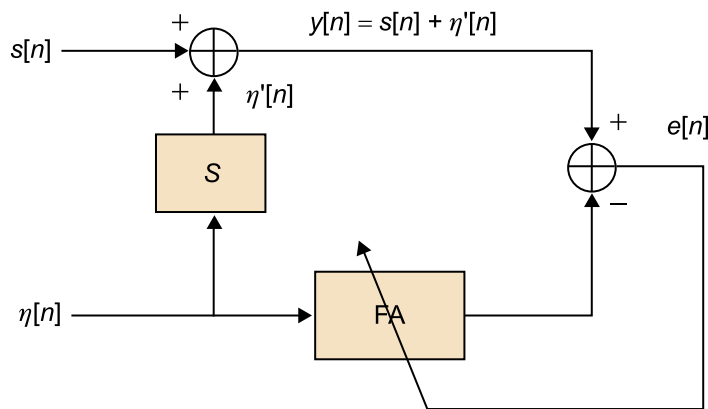


En l'esquema del gràfic (figura 4) s'observa com l'FA s'alimenta amb una versió retardada del senyal objectiu i s'entrena per a obtenir el senyal objectiu com a referència. Si l'entrenament és correcte, el sistema haurà estat capaç de determinar els coeficients que minimitzen l'error de predicció amb les dades d'entrenament, de tal manera que aquests coeficients es podran utilitzar per a fer estimacions de predicció en el futur.

## 2.4. Cancel·lació de soroll

En aquesta aplicació (figura 5), l'objectiu és eliminar la informació que contamina el senyal desitjat  $s[n]$ . L'exemple típic d'una aplicació d'aquest tipus pot ser la de la conversa en un helicòpter, en què el senyal interferent del motor té una intensitat elevada que distorsiona en gran manera la veu que es vol escoltar. El soroll del motor es transforma dins de l'habitacle, i se suma al senyal desitjat. El sistema d'entrenament utilitza el soroll del motor com a senyal d'entrada, i pretén obtenir  $y[n]$  a la sortida. El senyal  $y[n]$  està compost per dos components, la veu del pilot i el component de soroll. Atès que el filtre només podrà establir una predicció estadística sobre el component de soroll, el senyal d'error després de la convergència del sistema serà la veu del pilot,  $s[n]$ .

Figura 5



En aquest cas també s'il·lustra un altre dels aspectes importants del filtratge òptim. Que el filtre sigui òptim no implica que l'error sigui molt petit. En aquest cas, el senyal d'error és el senyal desitjat i té més potència que el soroll, però el disseny del sistema ha permès cancel·lar únicament la part de senyal correlacionada estadísticament amb el soroll, i ha deixat la veu del pilot gairebé intacta, en no haver-hi correlació estadística entre la veu del pilot i el soroll.

### Vegeu també

Vegeu el mòdul "Filtratge lineal òptim" d'aquesta assignatura.

### 3. El filtre LMS

En el camp del filtratge adaptatiu, el filtre LMS (*least mean square*) representa una de les estructures de referència en l'anàlisi i el comportament dels filtres adaptatius. Hi ha tres paràmetres que controlen el comportament d'un filtre adaptatiu, i que són els principals components que ha de tenir en compte el dissenyador del sistema en el moment d'escollir un tipus de filtratge o un altre:

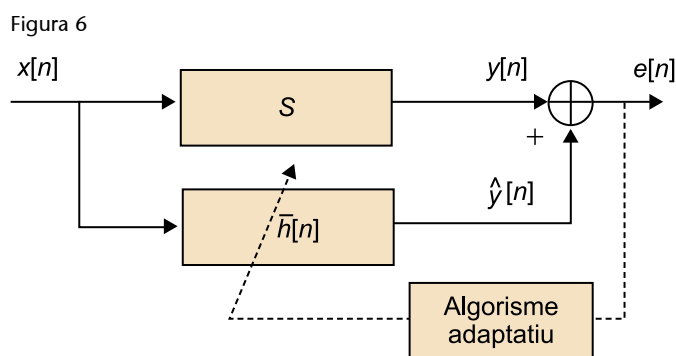
- **Velocitat de convergència del filtre:** la velocitat de convergència determina la capacitat amb la qual un filtre adaptatiu és capaç d'aproximar-se a la solució òptima del sistema. Com veurem, les diferents estructures de filtres adaptatius presenten diferents velocitats de convergència.
- **Cost computacional del sistema:** els algorismes de càlcul de filtres adaptatius presenten un cost computacional diferent segons el tipus d'estructura amb la qual es treballa. Aquest factor és fonamental en les aplicacions de temps real, ja que un sistema molt eficient però amb un cost computacional molt elevat pot no resultar la solució òptima en una aplicació concreta. El dissenyador ha d'avaluar la relació entre les prestacions del sistema i el cost de la implementació, ja que un cost computacional més elevat implicarà l'ús d'un maquinari de més cost, que encarrirà el producte final. El dissenyador ha d'avaluar aquests aspectes per a seleccionar la solució més apropiada.
- **Estabilitat de l'algorisme:** una vegada l'algorisme convergeix a la solució òptima del filtre, ha de ser capaç de mantenir-se en aquesta solució i amb la màxima robustesa possible davant canvis en els senyals. Les variacions dels senyals d'entrada i el fet de treballar amb precisió finita en aplicacions de maquinari fa que hi hagi algorismes que es puguin tornar inestables simplement per la falta de precisió numèrica d'un processador. Evidentment, aquest factor té un impacte cada vegada més petit a causa de la potència de les estructures de maquinari actuals, però en aplicacions de baix cost pot ser un factor important que calgui tenir en compte.

El filtre LMS, per les seves característiques, és una solució que s'adapta a un gran nombre d'aplicacions reals. D'una banda, presenta un cost computacional força reduït, i l'algorisme presenta una bona estabilitat si es mantenen les condicions de disseny apropiades. L'únic inconvenient que es pot trobar a aquest algorisme és la velocitat de convergència, que, segons les característiques dels senyals, pot alentir el sistema en algunes aplicacions.

### 3.1. Estructura del mètode LMS

El filtre LMS és la versió més senzilla de filtratge adaptatiu que podem trobar a la bibliografia. Es tracta d'una solució iterativa del filtre òptim de Wiener pel mètode del gradient descendent (*steepest descent*) que permet desenvolupar un algorisme adaptatiu de baixa complexitat i gran utilitat en aplicacions de processament, i és, possiblement, una de les estructures més utilitzades en aplicacions reals.

Per a la presentació de l'algorisme recordarem l'esquema bàsic de filtratge adaptatiu i la funció de cost per a minimitzar en el cas del filtratge lineal òptim o filtre de Wiener.



La idea del filtratge lineal òptim és aconseguir determinar el valor dels paràmetres òptims del filtre que minimitzen la funció de cost definida segons:

$$J = E[(y[n] - \hat{y}[n])^2] = E[e^2[n]]$$

### 3.2. El mètode del gradient descendent

La funció de cost definida en aplicacions de filtratge lineal òptim està definida segons l'error quadràtic mitjà. Convé conèixer amb una mica més de detall les característiques d'aquesta funció d'error, ja que això ens aportarà informació rellevant amb vista al desenvolupament de l'algorisme LMS, i també per a comprendre les possibilitats d'acceleració d'aquest, que portaran a estructures que s'exploraran en els subapartats següents.

Per al desenvolupament d'aquestes propietats convé repassar algunes operacions vectorials que ens permetran justificar matemàticament la funció d'error i comprendre els aspectes més importants d'aquest algorisme. Tanmateix, no es pretén abusar del desenvolupament matemàtic, sinó simplement aprendre aquells aspectes que realment siguin importants per a comprendre'l. Amb la finalitat de simplificar l'operatòria, i atès que això no resta rellevància als conceptes clau, farem el desenvolupament utilitzant el cas de senyals reals, fet que ajudarà a simplificar l'anàlisi.

Operant sobre la funció de cost obtenim que la funció d'error correspon a l'equació següent:

$$\begin{aligned} J &= E[(y[n] - \hat{y}[n])^2] = E\left[(y[n] - \vec{h}^T \vec{x}) (y[n] - \vec{h}^T \vec{x})^T\right] \\ &= E[y^2[n]] - \vec{h}^T E[\vec{x}y[n]] - E[y[n]\vec{x}^T] \vec{h} + \vec{h}^T E[\vec{x}\vec{x}^T] \vec{h} \end{aligned}$$

Els termes d'aquesta equació es poden relacionar amb uns estadístics que ja coneixeu, com són el vector de correlació creuada i la matriu d'autocorrelació:

$$J = \sigma_y^2 - 2\vec{h}^T \vec{r}_{yx} + \vec{h}^T \bar{R}_{xx} \vec{h}$$

Sabem que la solució òptima d'aquesta equació correspon a:

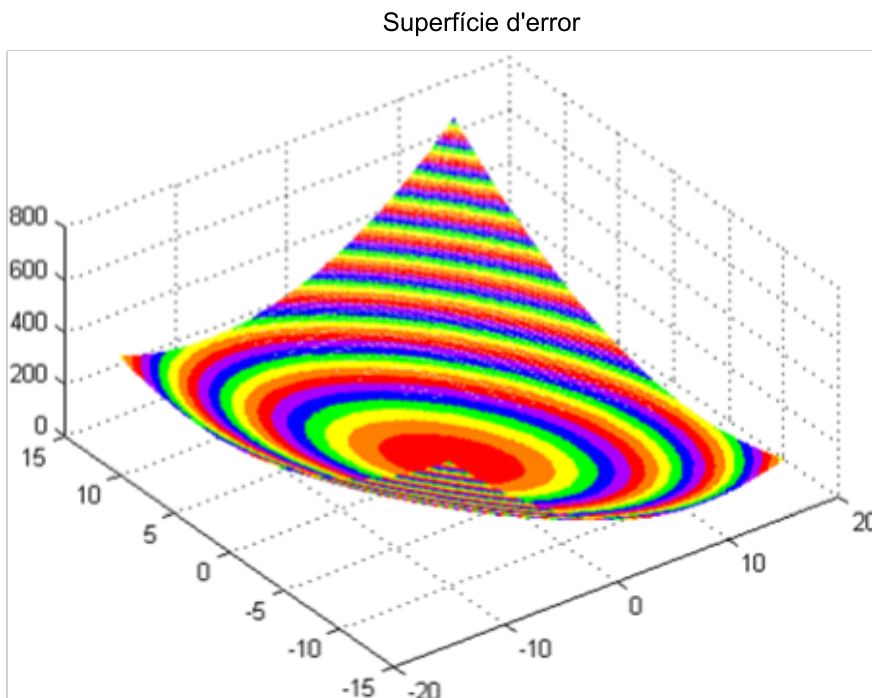
$$\vec{h}_{opt} = \bar{R}_{xx}^{-1} \vec{r}_{yx}$$

Podem descompondre la funció d'error  $J$  de manera que sigui més fàcil interpretar-ne la forma. Com bé sabem, es tracta d'una funció quadràtica, de manera que esperem que, a mesura que ens separem de la solució òptima, l'error augmentarà de manera quadràtica, però hi haurà direccions en què l'increment serà superior i altres en què serà més suau.

#### Vegeu també

Vegeu els diversos estadístics que es tracten en el mòdul "Filtratge lineal òptim" d'aquesta assignatura.

Figura 7



La representació bidimensional de la funció d'error per a un cas concret ajudarà a interpretar la morfologia de la funció i el seu impacte sobre les prestacions dels filtres dissenyats, ja que l'adaptació de la solució dependrà d'un recorregut al llarg d'aquesta funció fins a trobar el punt mínim.

En la figura 7 observem la forma de la funció d'error per a un cas concret bidimensional. La manipulació matemàtica ens permet reescriure la funció de cost segons:

$$J = J_{MIN} + (\vec{h} - \vec{h}_{opt})^T \vec{R}_{xx} (\vec{h} - \vec{h}_{opt})$$

$$J_{MIN} = \sigma_y^2 - \vec{r}_{yx}^T \vec{R}_{xx}^{-1} \vec{r}_{yx}$$

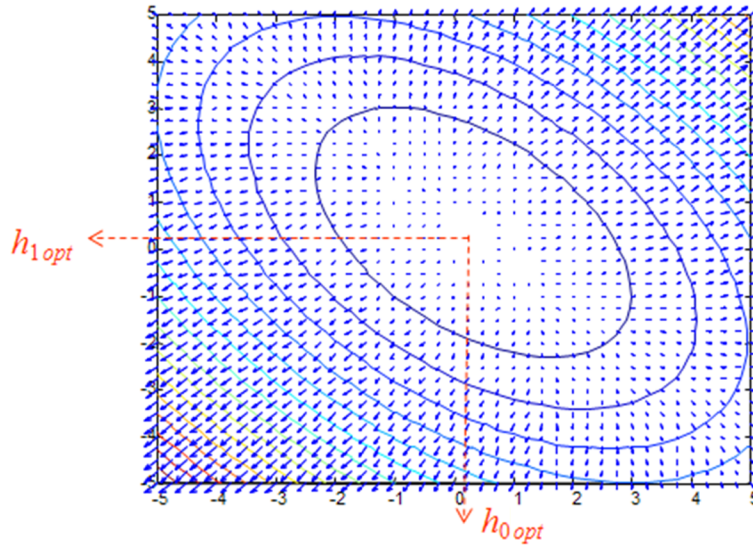
La interpretació d'aquesta equació ens indica que quan la resposta és l'òptima tenim un mínim de la funció de valor  $J_{MIN}$ , i que a mesura que ens separem del valor òptim, el creixement és quadràtic però ponderat per la matriu de correlació. Sabem que les matrius presenten guanyos diferents en direccions diferents, cosa que corrobora que hi ha direccions amb un creixement més gran que altres.

La matriu de correlació és una matriu simètrica, i per propietats de matrius sabem que aquestes són diagonalitzables, és a dir, es poden descompondre en direccions pròpies amb els seus autovalors respectius, de la manera següent:

$$J = J_{MIN} + (\vec{h} - \vec{h}_{opt})^T \vec{Q} \vec{\Lambda} \vec{Q}^T (\vec{h} - \vec{h}_{opt})$$

Sabem que la descomposició en autovectors i autovalors és com obtenir les empremtes dactilars de la matriu, atès que aquestes direccions són rellevants quant al comportament de la transformació. Les direccions pròpies de la matriu de correlació són justament els eixos de les el·lipses que es poden observar en la funció de cost  $J$ . La direcció que presenta l'autovalor més gran serà la que presenta un creixement més gran, mentre que la que presenta l'autovalor més petit correspon a la de menys pendent.

Figura 8



Observem en la figura 8 com el vector gradient (que indica la direcció de màxim creixement de la corba) és zero en el valor òptim, i va indicant la direcció de creixement, que és ortogonal a les corbes de nivell. S'observa que el vector gradient presenta més mòdul en el semieix menor de l'el·lipse, cosa que indica un creixement més gran de la funció de cost, i el mòdul és més petit en la direcció del semieix major, fet que indica justament el contrari.

Tenint en compte la representació de la funció d'error i els vectors gradient, aplicant senzillament el sentit comú veiem que per a trobar el mínim de la funció n'hi hauria prou de desplaçar-nos per la corba en sentit contrari al vector gradient i fer aproximacions successives fins que el gradient es fes zero. Aquesta idea és la que dóna origen al mètode d'optimització del gradient invers o *steepest descent*:

$$J = \sigma_y^2 - 2\vec{h}^T \vec{r}_{yx} + \vec{h}^T \vec{R}_{xx} \vec{h}$$

$$\nabla_{\vec{h}} J = \frac{\partial J}{\partial \vec{h}} = -2(\vec{r}_{yx} - \vec{R}_{xx} \vec{h})$$

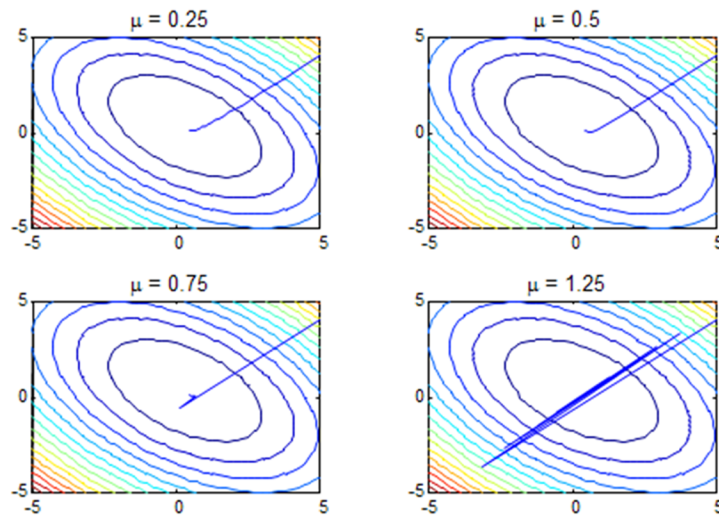
Una vegada conegut el valor del gradient de la corba, l'equació del filtre adaptatiu podria ser:

$$\vec{h}[n+1] = \vec{h}[n] - \mu \nabla_{\vec{h}} J = \vec{h}[n] + \mu(\vec{r}_{yx} - \vec{R}_{xx} \vec{h}[n])$$

En l'equació d'adaptació observem com el mètode del gradient invers modifica els coeficients del filtre en la direcció del gradient invers multiplicant el gradient per un factor d'actualització conegut com *pas d'adaptació*  $\mu$  o *step-size*.

No hi ha millor manera d'entendre l'efecte del pas de quantificació que observant el comportament de l'algorisme per a diferents valors del coeficient  $\mu$ .

Figura 9



La figura 9 mostra com, a mesura que l'algorisme incrementa el valor del paràmetre, la rapidesa de convergència del sistema augmenta, i són necessàries menys iteracions per a assolir el mínim del sistema. No obstant això, en l'última figura s'observa que quan el valor és massa elevat, se salta d'un costat de la superfície a la següent, i no costa imaginar que si el valor continués creixent el sistema es tornaria inestable i no aconseguiria trobar el mínim de la corba.

Amb això podem observar que el mètode del gradient requereix una bona sintonització del paràmetre d'adaptació a fi d'aconseguir que el sistema funcioni de la manera més ràpida possible sense perdre l'estabilitat en la resposta.

### Velocitat de convergència del mètode del gradient

Com hem dit, un dels aspectes importants dels algorismes adaptatius és la velocitat de convergència, és a dir, la capacitat que té l'algorisme d'arribar de manera ràpida a la solució òptima. Algunes operacions matemàtiques sobre l'equació d'actualització dels coeficients del filtre permeten reescriure l'equació de manera que ens permeti interpretar la velocitat de convergència:

$$\begin{aligned}\vec{h}[n+1] &= \vec{h}[n] + \mu(\vec{r}_{yx} - \hat{R}_{xx}\vec{h}[n]) = (\bar{I} - \mu\hat{R}_{xx})\vec{h}[n] + \mu\vec{r}_{yx} \\ \vec{h}[n+1] &= (\bar{I} - \mu\hat{R}_{xx})\vec{h}[n] + \mu\hat{R}_{xx}\vec{h}_{opt} + \vec{h}_{opt} - \vec{h}_{opt} \\ (\vec{h}[n+1] - \vec{h}_{opt}) &= (\bar{I} - \mu\hat{R}_{xx})(\vec{h}[n] - \vec{h}_{opt})\end{aligned}$$

Finalment, si resollem aquesta equació de manera iterativa des de la iniciació del sistema, ens quedaria de la manera següent:

$$\vec{h}[n] = \vec{h}_{opt} + (\bar{I} - \mu\hat{R}_{xx})^n (\vec{h}[0] - \vec{h}_{opt})$$



S'observa que en la iteració inicial, és a dir, quan  $n = 0$ , estaríem en les condicions inicials, i quan  $n$  tendeix a infinit s'hauria d'anul·lar el segon terme de l'equació perquè l'algorisme convergís cap a  $h_{opt}$ . Veiem que l'equació presenta una potència d'una matriu, i perquè es compleixi l'estabilitat de l'algorisme, aquesta potència en el límit ha de tendir a zero:

$$\lim_{n \rightarrow \infty} (\bar{I} - \mu \bar{R}_{xx})^n = \bar{0} \quad \Leftrightarrow \quad |\bar{I} - \mu \bar{R}_{xx}| < 1$$

Com que estem treballant amb l'error quadràtic mitjà, podem utilitzar la norma euclidiana matricial, que determina que la norma de la matriu correspon al mòdul del seu autovalor màxim. Considerant els diferents autovalors de la matriu, el cas més crític quant a condicions d'estabilitat el marcarà l'autovalor màxim quan es compleixi la condició següent:  $1 - \mu \lambda_{max} = -1$ , que marcaria el valor màxim de  $\mu$  per a garantir l'estabilitat en el valor següent:

$$\mu \leq \frac{2}{\lambda_{max}}$$

Evidentment, aquest valor estaria en el límit de la inestabilitat, de manera que no seria un valor apropiat per a condicions de disseny. En aquest cas, seria molt millor que escollíssim un valor més prudent, com per exemple:

$$\mu \leq \frac{\beta}{\lambda_{max}}$$

en què  $\beta$  és un paràmetre de valor entre 0 i 1.

Sota aquesta premissa de disseny podem fer una aproximació a la velocitat de convergència del sistema i als paràmetres de les funcions d'entrada que influeixen en aquest aspecte. Per les condicions del disseny sota consideració, es pot garantir que:

$$(1 - \mu \lambda_k)^n \leq (1 - \mu \lambda_{min})^n$$

De tal manera que la velocitat de convergència del sistema es pot aproximar segons el teorema de Taylor a una exponencial decreixent del tipus:

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

Ara sí que estem en condicions de fer una avaluació dels aspectes del sistema que influeixen en la velocitat de convergència de l'algorisme.

Tal com havíem vist anteriorment, la constant  $\mu$  és determinant en la velocitat, i l'haurem d'adaptar per a aconseguir màximes prestacions; però hi ha un altre aspecte important, que és la dispersió d'autovalors. Una vegada determinat el valor del paràmetre  $\beta$ , la dispersió en els autovalors de la matriu de correlació serà un factor que provocarà un alentiment del comportament de l'algorisme.

### **Com afectarà la informació del senyal d'entrada a la dispersió d'autovalors?**

Aquesta pregunta serà clau per a poder determinar quin tipus de senyals provocaran velocitats de convergència ràpides i quin tipus de senyals provocaran velocitats de convergència lentes. En el cas de senyals incorrelacionats, la matriu de correlació serà la identitat, i podem deduir fàcilment que en aquest cas tots els autovalors tindran el mateix valor. Si ho relacionem amb les corbes de nivell en dues dimensions, les corbes de nivell serien circumferències i la superfície tindria el mateix creixement en qualsevol de les seves direccions.

A mesura que els senyals d'entrada comencen a estar més correlacionats, les corbes de nivell tendeixen a formes el·líptiques cada vegada més aplanades. Veurem llavors que hi haurà una direcció amb un pendent molt més pronunciat (la de més autovalor), i una altra que serà molt més suau. El disseny de l'algorisme l'haurem de fer per a garantir la convergència en la direcció de màxim pendent, i per aquest motiu el valor de  $\mu$  està limitat per l'autovalor màxim, però la velocitat de convergència més lenta la tindrem quan vinguem per la direcció de mínim pendent.

Vist això, podem intuir que quan els senyals estan fortament correlacionats, l'algorisme tendirà a presentar una velocitat de convergència molt més lenta. Això adquireix sentit també si ho mirem en el domini temporal, atès que quan un senyal està fortament correlacionat, cada mostra nova aporta molt poca informació que no estigui continguda en les mostres anteriors, de manera que a l'algorisme li costarà més temps determinar en quina direcció ha de modificar els coeficients. En canvi, quan el senyal d'entrada està incorrelacionat, cada mostra nova aporta informació que no estava continguda en les mostres anteriors. En haver-hi aquesta aportació d'informació nova de manera molt més clara, l'algorisme podrà evolucionar ràpidament cap al mínim de la funció.

Més endavant veurem que el fet de treballar amb versions incorrelacionades dels senyals d'entrada és la clau per al desenvolupament d'algorismes adaptatius de gran velocitat de convergència. La descorrelació aclareix la manera com es mostra la informació i impedeix la confusió generada amb vectors altament correlacionats, en què resulta complex extreure la informació nova aportada per cada mostra.

### 3.3. El mètode LMS

Una vegada analitzada l'estructura bàsica del mètode LMS i també la conceptualització del mètode del gradient, arriba el moment de dissenyar l'algorisme adaptatiu del mètode LMS que permetrà aplicar-lo a senyals reals.

És important remarcar que el mètode del gradient descendent és útil per a comprendre els conceptes de l'LMS, però que no s'utilitza en la pràctica, ja que si ja coneixem els valors de correlació creuada i la matriu d'autocorrelació, podem resoldre directament el sistema i obtenir la solució òptima de manera directa. La gràcia d'un mètode adaptatiu és que les modificacions dels coeficients del filtre es facin mitjançant operacions senzilles dels valors d'entrada i sense la necessitat d'estimar vectors i matrius de correlació.

El càlcul del vector gradient s'ha fet en subapartats anteriors, de manera que no en repetirem l'operatòria, i sabem que correspon a l'equació següent:

$$\nabla_{\vec{h}} J = \frac{\partial J}{\partial \vec{h}} = -2E[\vec{x}[n]e[n]] = -2E[y[n]\vec{x}[n]] + 2E[\vec{x}[n]\vec{x}[n]^T] \vec{h}[n]$$

en què el primer terme de l'equació correspon al vector de correlació creuada entre el senyal objectiu i les mostres d'entrada, i el segon terme, a la matriu d'autocorrelació.

El mètode LMS consisteix en una idea molt senzilla, que no és més que aproximar el càlcul del vector gradient mitjançant un estimador que faci servir només les mostres actuals:

$$\frac{\partial J}{\partial \vec{h}} = -2E[\vec{x}[n]e[n]] \approx -2\vec{x}[n]e[n]$$

Podem pensar que aquesta aproximació és molt pobre per a obtenir un algorisme amb bons resultats, però es pot demostrar matemàticament que el comportament estadístic de l'algorisme és exactament el mateix que el de l'*steepest descent* o gradient descendent:

$$\begin{aligned} E[\vec{h}[n+1]] &= E[\vec{h}[n] + \mu e[n]\vec{x}[n]] \\ &= E[\vec{h}[n]] + \mu E[e[n]\vec{x}[n]] \end{aligned}$$

La principal diferència és que l'evolució de l'algorisme es farà de manera més sorollosa, és a dir, amb més variància que la que tindriem en el mètode del gradient, però que finalment tots dos algorismes ens portarien a la mateixa solució. La variància de la solució òptima es pot estimar, i proporciona el resultat següent:

$$\lim_{n \rightarrow \infty} E \left[ \|\vec{h}[n] - \vec{h}_{opt}\|_2^2 \right] = \mu J_{\min}$$

No entrarem en els detalls del càlcul, però és important que ens quedem amb la idea, òbvia d'altra banda, que com més gran sigui el valor de la constant d'adaptació, més variància presentarà la solució del sistema, és a dir, més costarà al sistema establir-se en la solució correcta sense experimentar petites variacions entorn d'aquesta.

### Coefficient d'adaptació variable

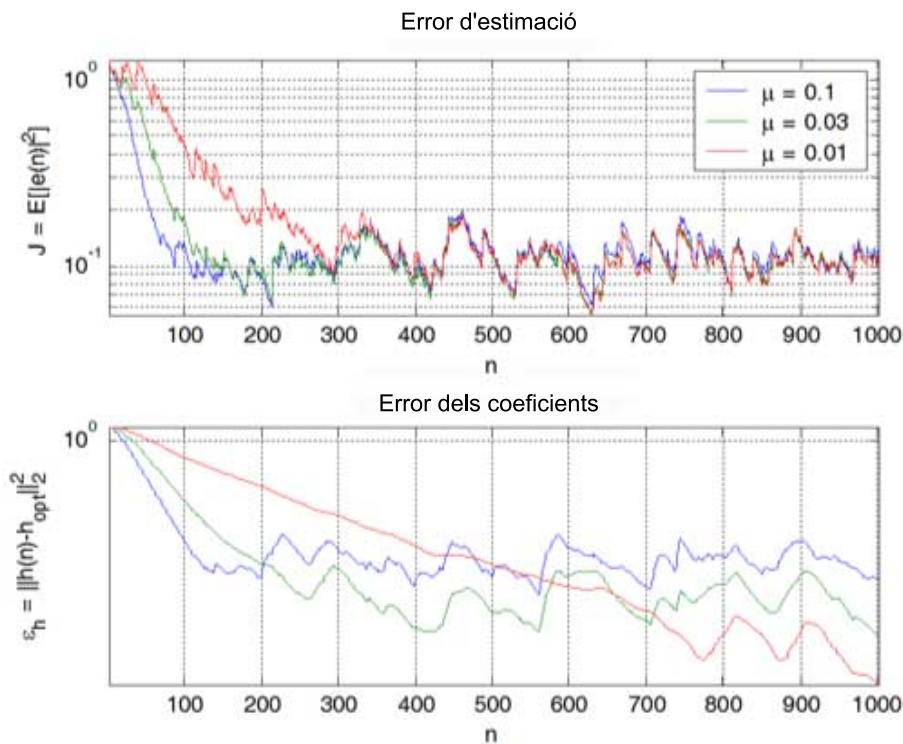
Com hem observat, en les condicions inicials de l'algorisme ens interessaria un coeficient d'adaptació prou gran per a aconseguir una ràpida aproximació cap al mínim de la funció d'error, però en canvi, quan ens trobem en valors propers al mínim de la funció, ens interessarà una constant d'adaptació variable que minimitzi la variància de l'estimació.

Amb això en ment, podríem establir l'estratègia següent:

- $\mu$  elevada en la fase inicial o fase d'entrenament.
- $\mu$  reduïda en la fase de seguiment.

L'exemple següent il·lustra el comportament d'un filtre LMS en funció de tres valors diferents de la constant d'adaptació  $\mu$ .

Figura 10



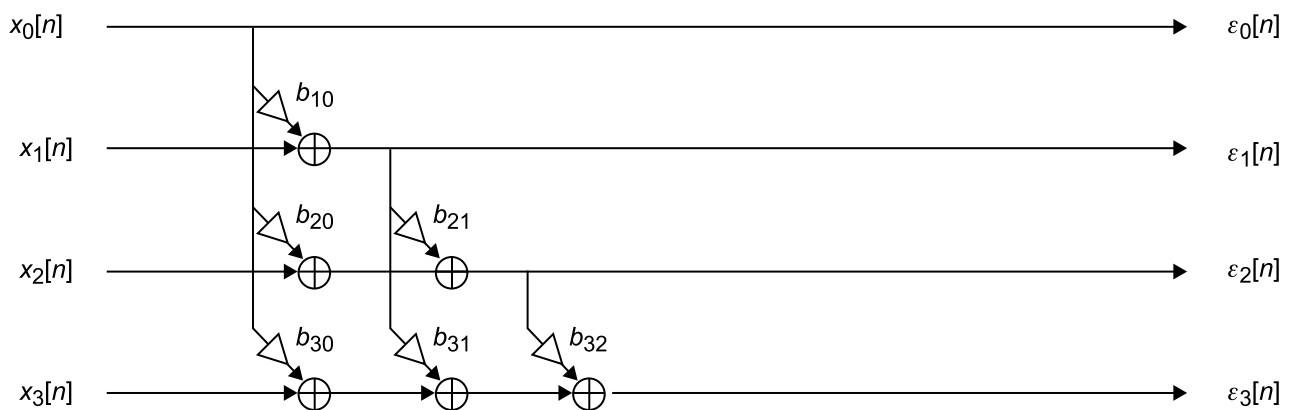
En la figura 10 podem observar com un valor petit de la constant d'adaptació  $\mu$  produeix una convergència més lenta en l'algorisme, tal com podem veure en la gràfica vermella, però la variància de l'error dels coeficients es fa més petita quan assoleix la convergència. Per la seva banda, quan el valor de la constant d'adaptació es fa més gran, com en la gràfica blava, la convergència es dona amb més rapidesa i s'arriben abans als valors òptims dels coeficients del filtre, però la variància de l'estimació és superior. La gràfica il·lustra el que hem comentat respecte al coeficient d'adaptació variable. En els primers instants de temps ens interessa un valor de convergència elevat per a apropar-nos ràpidament a la zona del mínim de la funció, per a reduir després el coeficient i disposar d'un sistema amb capacitat per a adaptar-se als canvis d'entorn però que presenti una variància més baixa en l'estimació dels coeficients del filtre, que proporcioni un resultat amb unes prestacions més estables i menys sensibles a aquest efecte del soroll.

## 4. Gram-Schmidt

Les estructures de Gram-Schmidt tenen com a propietat principal el fet d'aconseguir generar un conjunt de  $N$  senyals incorrelacionats, a partir de  $N$  senyals d'entrada. Aquesta propietat permet accelerar la velocitat de convergència de les tècniques adaptatives, atès que si es disposa de senyals incorrelacionats, les direccions d'avenç són molt més precises que les que es poden aconseguir amb senyals que tenen certes dependències lineals.

A la figura 11 es mostra l'esquema d'un processador de Gram-Schmidt de dimensió  $N = 4$ .

Figura 11. Processador de Gram-Schmidt de dimensió  $N = 4$



Com es pot observar en la figura 11, el conjunt de senyals de sortida,  $\varepsilon_i[n]$ , es genera a partir de senzilles combinacions lineals dels senyals d'entrada,  $x_i[n]$ . L'objectiu de l'estructura és aconseguir eliminar les dependències lineals entre les variables d'entrada utilitzant els valors apropiats de les constants utilitzades en el procés de combinació,  $b_{ij}$ .

Aquest procés es fa de manera acumulativa, la qual cosa s'aconsegueix seguint un procés iteratiu: es comença per eliminar la dependència lineal entre les dues primeres variables d'entrada,  $x_0[n]$  i  $x_1[n]$ , i s'obtenen les sortides  $\varepsilon_0[n]$  i  $\varepsilon_1[n]$ . A continuació, s'elimina la dependència lineal entre aquests dos senyals ja descorrelacionats i el tercer senyal d'entrada,  $x_2[n]$ , i s'obté un tercer senyal dins del que ja denominem **conjunt de senyals descorrelacionats**:  $\{\varepsilon_0[n], \varepsilon_1[n], \varepsilon_2[n]\}$ . Es continua eliminant la dependència lineal entre aquest conjunt de senyals descorrelacionats i el tercer senyal d'entrada,  $x_3[n]$ , i així successivament.

#### 4.1. Anàlisi de la solució òptima per a $N = 2$ senyals

Comencem fent l'anàlisi per a la segona variable de sortida, per a veure el **criteri de disseny dels coeficients**  $b_{ij}$  del processador de Gram-Schmidt. Aquest criteri es basa senzillament en la minimització de la potència dels senyals de sortida; és a dir, per al cas del senyal de sortida  $\varepsilon_1[n]$ , de:

$$E[\varepsilon_1[n]^2] = E[x_1[n] + b_{10}x_0[n]^2]$$

Si volem minimitzar la potència del senyal  $\varepsilon_1[n]$ , procedirem de la manera següent, suposant senyals reals:

$$E[\varepsilon_1[n]^2] \Big|_{MIN} \Leftrightarrow \frac{\partial E[\varepsilon_1[n]^2]}{\partial b_{10}} = 0$$

Atès que estem aplicant un criteri basat en la minimització d'un error quadràtic sobre un operador que és lineal, podem arribar a la solució òptima resolent la derivada anterior:

$$\begin{aligned} \frac{\partial E[\varepsilon_1[n]^2]}{\partial b_{10}} &= \frac{\partial E[x_1[n] + b_{10}x_0[n]^2]}{\partial b_{10}} = 2E[x_0[n](x_1[n] + b_{10}x_0[n])] = \\ &= 2E[x_0[n]x_1[n]] + 2b_{10}E[x_0[n]x_0[n]] = 0 \end{aligned}$$

Amb la qual cosa la solució final és la següent:

$$b_{10} = -\frac{E[x_0[n]x_1[n]]}{E[x_0[n]x_0[n]]} = -\frac{E[\varepsilon_0[n]x_1[n]]}{E[\varepsilon_0[n]\varepsilon_0[n]]}$$

Com podem observar, el coeficient  $b_{10}$  és un coeficient de correlació creuada entre els senyals  $x_0[n]$  i  $x_1[n]$ , convenientment normalitzat segons la potència del senyal  $x_0[n]$  i canviat de signe. És a dir, que per a estimar la part de senyal de  $x_1[n]$ , que és independent, o més aviat dit, no està correlacionada amb el senyal  $x_0[n]$ , hem de sotstreure de  $x_1[n]$  la proporció de  $x_0[n]$  justa que està relacionada amb aquesta relació de dependència lineal entre tots dos senyals, i normalitzada respecte de la potència de  $x_0[n]$  per a independitzar-la de la seva energia.

#### Principi d'ortogonalitat

La solució òptima està directament relacionada amb el que es denomina **principi d'ortogonalitat**, de manera que minimitzar la potència del senyal de sortida  $\varepsilon_1[n]$  és equivalent a aconseguir que aquest senyal sigui ortogonal al senyal anterior  $\varepsilon_0[n]$ , que és justament el propòsit del processador de Gram-Schmidt. Això es pot veure justament en el desenvolupament de la solució òptima anterior. Si reescrivim el gradient respecte del coeficient  $b_{10}$  de la manera següent:

$$\frac{\partial E[\varepsilon_1[n]^2]}{\partial b_{10}} = 2E[x_0[n](x_1[n] + b_{10}x_0[n])] = 2E[x_0[n]\varepsilon_1[n]] = 0$$

$$\Rightarrow E[\varepsilon_0[n]\varepsilon_1[n]] = 0 \quad \equiv \quad \varepsilon_0[n] \perp \varepsilon_1[n]$$

És a dir, que  $\varepsilon_0[n]$  i  $\varepsilon_1[n]$  són senyals incorrelacionats, o el que és el mateix, són linealment independents.

#### 4.2. Anàlisi de la solució òptima per a $N = 3$ senyals

Atès que es tracta d'un procés incremental, l'anàlisi per a  $N = 3$  senyals inclou la solució per a les  $N = 2$  primeres variables. Analitzem a continuació el cas del tercer senyal de sortida. Com s'ha comentat en el subapartat anterior, el criteri es basa en la minimització de la potència del senyal de sortida. En aquest cas, i recordant l'esquema de la figura 11:

$$E[\varepsilon_2[n]^2] = E[x_2[n] + b_{20}\varepsilon_0[n] + b_{21}\varepsilon_1[n]^2]$$

Per a minimitzar la funció anterior, primer hem de derivar la funció respecte a cadascun dels coeficients de la combinació lineal que genera el tercer senyal de sortida:

$$\begin{aligned} \frac{\partial E[\varepsilon_2[n]^2]}{\partial b_{20}} &= \frac{\partial E[(x_2[n] + b_{20}\varepsilon_0[n] + b_{21}\varepsilon_1[n])^2]}{\partial b_{20}} = \\ &= 2E[\varepsilon_0[n](x_2[n] + b_{20}\varepsilon_0[n] + b_{21}\varepsilon_1[n])] = \\ &= 2E[\varepsilon_0[n]x_2[n]] + 2b_{20}E[\varepsilon_0[n]\varepsilon_0[n]] + 2b_{21}\underbrace{E[\varepsilon_0[n]\varepsilon_1[n]]}_{=0} = \\ &= 2E[\varepsilon_0[n]x_2[n]] + 2b_{20}E[\varepsilon_0[n]\varepsilon_0[n]] \end{aligned}$$

$$\begin{aligned} \frac{\partial E[\varepsilon_2[n]^2]}{\partial b_{21}} &= \frac{\partial E[(x_2[n] + b_{20}\varepsilon_0[n] + b_{21}\varepsilon_1[n])^2]}{\partial b_{21}} = \\ &= 2E[\varepsilon_1[n](x_2[n] + b_{20}\varepsilon_0[n] + b_{21}\varepsilon_1[n])] = \\ &= 2E[\varepsilon_1[n]x_2[n]] + 2b_{20}\underbrace{E[\varepsilon_1[n]\varepsilon_0[n]]}_{=0} + 2b_{21}E[\varepsilon_1[n]\varepsilon_1[n]] = \\ &= 2E[\varepsilon_1[n]x_2[n]] + 2b_{21}E[\varepsilon_1[n]\varepsilon_1[n]] \end{aligned}$$

en què s'ha aplicat el fet que els senyals de sortida  $\varepsilon_0[n]$  i  $\varepsilon_1[n]$  són senyals descorrelacionats, la qual cosa comporta una simplificació evident de tots dos gradients.

Amb això, la solució del procés de minimització és la següent:

$$E[\varepsilon_2[n]^2]_{MIN} \quad \Leftrightarrow \quad \frac{\partial E[\varepsilon_2[n]^2]}{\partial b_{20}} = \frac{\partial E[\varepsilon_2[n]^2]}{\partial b_{21}} = 0$$



$$b_{20} = -\frac{E[\varepsilon_0[n]x_2[n]]}{E[\varepsilon_0[n]\varepsilon_0[n]]} \quad b_{21} = -\frac{E[\varepsilon_1[n]x_2[n]]}{E[\varepsilon_1[n]\varepsilon_1[n]]}$$

Comprovem l'ortogonalitat del senyal  $\varepsilon_2[n]$  amb els dos senyals anteriors  $\varepsilon_0[n]$  i  $\varepsilon_1[n]$ , començant pel primer:

$$\begin{aligned} E[\varepsilon_2[n]\varepsilon_0[n]] &= E[(x_2[n] + b_{20}\varepsilon_0[n] + b_{21}\varepsilon_1[n])\varepsilon_0[n]] = \\ &= E[x_2[n]\varepsilon_0[n]] + b_{20}E[\varepsilon_0[n]\varepsilon_0[n]] + b_{21}\underbrace{E[\varepsilon_0[n]\varepsilon_1[n]]}_{=0} = \\ &= E[x_2[n]\varepsilon_0[n]] + \left(-\frac{E[\varepsilon_0[n]x_2[n]]}{E[\varepsilon_0[n]\varepsilon_0[n]]}\right)E[\varepsilon_0[n]\varepsilon_0[n]] = 0 \end{aligned}$$

És fàcil fer el mateix procediment per a verificar que, efectivament, tant  $\varepsilon_0[n]$  com  $\varepsilon_1[n]$  són senyals ortogonals a  $\varepsilon_2[n]$ , a més de ser-ho entre si (ja verificat en el subapartat anterior).

### 4.3. Solució general òptima

Com es pot observar, si es prossegueix amb l'anàlisi de més variables de sortida (per exemple, per a  $N = 4$ ), seguint exactament el mateix desenvolupament per al senyal de sortida  $i$ -èsim:

$$\begin{aligned} \frac{\partial E[|\varepsilon_i[n]|^2]}{\partial b_{ik}} &= \frac{\partial E[(x_i[n] + \sum_{j=0}^{i-1} b_{ij}\varepsilon_j[n])^2]}{\partial b_{ik}} = 2E[\varepsilon_k[n]\varepsilon_i[n]] = \\ &= 2E[\varepsilon_k[n](x_i[n] + \sum_{j=0}^{i-1} b_{ij}\varepsilon_j[n])] \\ &= 2E[\varepsilon_k[n]x_i[n]] + 2\sum_{j=0}^{i-1} b_{ij}\underbrace{E[\varepsilon_k[n]\varepsilon_j[n]]}_{=0, j \neq k} = \\ &= 2E[\varepsilon_k[n]x_i[n]] + 2b_{ik}E[\varepsilon_k[n]\varepsilon_k[n]] = 0 \end{aligned}$$

Amb això, les equacions de disseny òptim per als coeficients de l'estructura donaran lloc a la solució general següent:

$$b_{ik}^{opt} = -\frac{E[\varepsilon_k[n]x_i[n]]}{E[\varepsilon_k[n]\varepsilon_k[n]]} = -\frac{E[\varepsilon_k[n]x_i[n]]}{E[\varepsilon_k^2[n]]}$$

$$\varepsilon_i[n] = x_i[n] + \sum_{j=0}^{i-1} b_{ij}\varepsilon_j[n]$$

De la solució òptima es dedueix que en general es complirà la **condició d'ortogonalitat** per a totes les variables de sortida, escollides de dues en dues:

$$E[\varepsilon_k[n]\varepsilon_l[n]] = \frac{1}{2} \frac{\partial E[\varepsilon_i^2[n]]}{\partial b_{ik}} = 0 \quad \forall k \neq i$$

#### 4.4. Solució adaptativa

##### 4.4.1. Algorisme del gradient descendent i estudi de la seva convergència

A continuació es deriva la solució adaptativa basada en l'algorisme del gradient descendent per a l'estructura de Gram-Schmidt, i després es deriva la condició de convergència que s'ha de complir per al paràmetre d'adaptació  $\mu$  de l'algorisme.

En primer lloc, si apliquem aquest algorisme per a l'adaptació del coeficient genèric:

$$b_{ik}[n] = b_{ik}[n-1] - \mu \frac{\partial E[\varepsilon_i^2[n]]}{\partial b_{ik}} = b_{ik}[n-1] - \mu E[\varepsilon_k[n]\varepsilon_l[n]] \quad (1)$$

en què s'ha omès el valor constant 2 que multiplica el valor del gradient, per a poder ser absorbit en el factor mateix de convergència  $\mu$ .

Per a assegurar la convergència envers la solució òptima, primer hem d'analitzar amb més deteniment quins valors del paràmetre d'adaptació  $\mu$  permetrà aquesta convergència. Per a això, primer s'ha d'expressar l'equació d'adaptació de la manera següent:

$$\begin{aligned} b_{ik}[n] &= b_{ik}[n-1] - \mu E[\varepsilon_k[n]\varepsilon_l[n]] = \\ &= b_{ik}[n-1] - \mu(E[\varepsilon_k[n]x_l[n]] + b_{ik}[n-1]E[\varepsilon_k[n]\varepsilon_l[n]]) = \\ &= b_{ik}[n-1](1 - \mu E[\varepsilon_k^2[n]]) - \mu E[\varepsilon_k[n]x_l[n]] \end{aligned}$$

Restant a banda i banda de la igualtat la solució òptima per al coeficient  $b_{ik}$ :

$$\begin{aligned} b_{ik}[n] - b_{ik|opt} &= b_{ik}[n-1](1 - \mu E[\varepsilon_k^2[n]]) - \mu E[\varepsilon_k[n]x_l[n]] - b_{ik|opt} = \\ &= b_{ik}[n-1](1 - \mu E[\varepsilon_k^2[n]]) - \mu E[\varepsilon_k[n]x_l[n]] - \left(-\frac{E[\varepsilon_k[n]x_l[n]]}{E[\varepsilon_k^2[n]]}\right) = \\ &= b_{ik}[n-1](1 - \mu E[\varepsilon_k^2[n]]) - \left(-\frac{E[\varepsilon_k[n]x_l[n]]}{E[\varepsilon_k^2[n]]}\right)(1 - \mu E[\varepsilon_k^2[n]]) = \\ &= (b_{ik}[n-1] - b_{ik|opt})(1 - \mu E[\varepsilon_k^2[n]]) \end{aligned}$$

Partint d'una solució inicial  $b_{ik}[0]$  (per a  $n=0$ ), podem expressar, a partir de l'equació recursiva anterior, la solució per a un instant qualsevol en funció de la solució inicial, com segueix:

$$b_{ik}[n] - b_{ik|opt} = (b_{ik}[0] - b_{ik|opt}) \left(1 - \mu E[\varepsilon_k^2[n]]\right)^n$$

És a dir, la solució després de la  $n$ -èsima iteració depèn exponencialment del factor de convergència  $\mu$  i també de l'energia de la sortida  $E[\varepsilon_k^2[n]]$ . Com es pot observar en l'equació anterior, només si  $|1 - \mu E[\varepsilon_k^2[n]]| < 1$ , l'algorisme convergirà cap a la solució òptima, atès que la potencia  $n$ -èsima d'aquest factor tendirà cap a zero:

$$\lim_{n \rightarrow +\infty} (b_{ik}[n] - b_{ik|opt}) = \lim_{n \rightarrow +\infty} (b_{ik}[0] - b_{ik|opt}) \underbrace{\left(1 - \mu E[\varepsilon_k^2[n]]\right)^n}_{-1 < r < +1} = 0$$

$$\Rightarrow \lim_{n \rightarrow +\infty} b_{ik}[n] = b_{ik|opt}$$

D'això es dedueix que els valors del paràmetre d'adaptació  $\mu$  que garantiran la convergència de l'algorisme del gradient envers la solució òptima seran els següents:

$$-1 < 1 - \mu E[\varepsilon_k^2[n]] < +1 \quad \Rightarrow \quad 0 < \mu < \frac{2}{E[\varepsilon_k^2[n]]} \quad (2)$$

Una conclusió molt interessant per remarcar en aquest punt és que el processador de Gram-Schmidt adaptatiu permet assegurar, gràcies a les seves propietats de descorrelació entre les variables que s'obtenen, una convergència eficient envers la solució òptima si es disposa d'una estimació de l'energia de les variables de sortida del processador. Com ja es va veure en la solució LMS per a un estimador del tipus FIR, com més gran és el valor del paràmetre  $\mu$  més gran és la velocitat de convergència, sempre que no se superi el llindar que portaria a la inestabilitat de l'algorisme. En aquest cas, aquest llindar queda definit de manera inversament proporcional a l'energia de sortida, com es pot veure en l'equació (2).

#### 4.4.2. Algorisme LMS

Igual que en la resta de tècniques adaptatives, el processador de Gram-Schmidt es pot adaptar a una versió LMS que no requereixi l'estimació de valors mitjans (o esperats) per a l'entrenament. De fet, la versió LMS parteix de la solució basada en l'algorisme del gradient descendent, estudiada en l'apartat anterior, i l'adapta per a poder usar el valor de les variables instantànies de què es disposa a cada moment.

En primer lloc, vegem com s'obté el valor de la variable d'adaptació. Seguint el desenvolupament de l'apartat anterior vam veure que el valor de  $\mu$  ha d'estar afitat segons el valor de l'energia de la sortida per a cada branca del processador. D'altra banda, recordem que valors elevats de  $\mu$  permetran obtenir convergències més ràpides que valors menors, sempre que no se superi el límit

#### Vegeu també

Hem vist la solució LMS per a un estimador del tipus FIR en l'apartat 3 d'aquest mòdul didàctic.

que porta a la inestabilitat del mètode, o el que és el mateix, a no convergir envers la solució òptima. Amb tot això, i recordant que l'objectiu del mètode LMS és evitar l'ús de valors esperats en les equacions d'adaptació, comencem per establir el mètode d'obtenció de l'energia de la sortida  $\varepsilon_k[n]$ , per mitjà d'un senzill mitjanament o filtratge passabaix amb un filtre IIR d'ordre 1, també denominat filtre de suavització o *smoothing*:

$$E_k[n] = \lambda E_k[n-1] + (1-\lambda)\varepsilon_k^2[n] \quad (3)$$

en què  $\lambda$  és el denominat **paràmetre de suavització**, definit en l'interval  $0 < \lambda < 1$ .

### Anàlisi del filtre de suavització

Si analitzem la funció de transferència d'aquest filtre,  $H_{lp}(z)$ , que té com a entrada el quadrat de la sortida,  $w_k[n] = \varepsilon_k^2[n]$ , i com a sortida la variable d'energia  $E_k[n]$ :

$$H_{lp}(z) = \frac{W_k(z)}{E_k(z)} = \frac{1-\lambda}{1-\lambda z^{-1}}$$

En continuem analitzant la resposta freqüencial, i concretament el guany o la resposta en mòdul:

$$H_{lp}(e^{j\omega}) = \frac{W_k(e^{j\omega})}{E_k(e^{j\omega})} = \frac{(1-\lambda)}{1-\lambda e^{-j\omega}}$$

$$|H_{lp}(e^{j\omega})| = \left| \frac{1-\lambda}{1-\lambda e^{-j\omega}} \right| = \frac{1-\lambda}{\sqrt{(1-\lambda \cos(\omega))^2 + \lambda^2 \sin^2(\omega)}} = \frac{1-\lambda}{\sqrt{1+\lambda^2 - 2\lambda \cos(\omega)}}$$

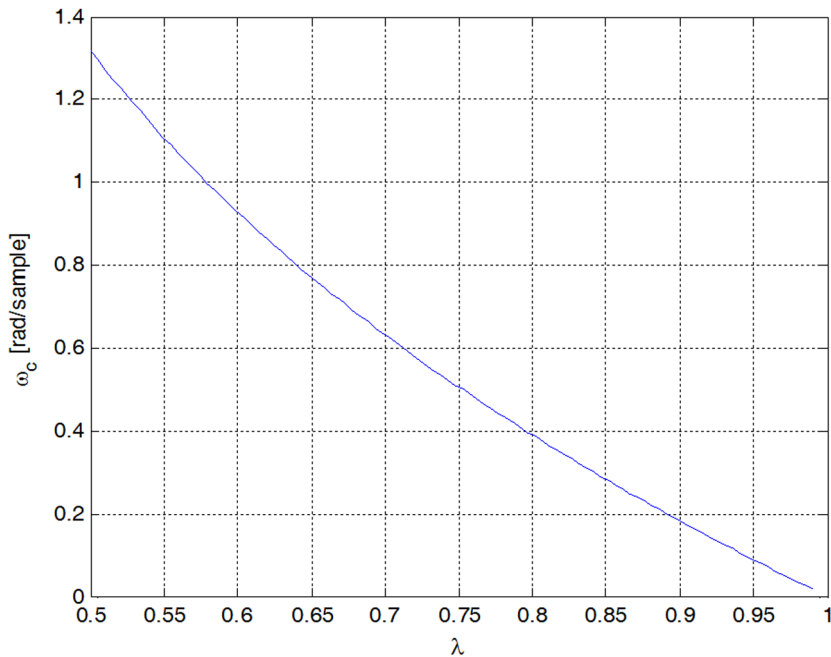
Podem observar com aquest filtre té guany unitari per a la pulsació  $\omega = 0$ , i que a mesura que  $\omega$  creix fins al valor de la freqüència de Nyquist ( $\omega = \pi$ ), el mòdul anirà decreixent de manera progressiva fins al seu valor mínim. Per a trobar-ne la freqüència de tall,  $\omega_c$ , en què el mòdul s'ha reduït fins a la meitat del guany màxim, igulem aquest guany en l'expressió del mòdul de la funció de transferència:

$$\begin{aligned} \left| H_{lp}(e^{j\omega}) \right|_{\omega=\omega_c} = \frac{1}{2} &= \frac{1-\lambda}{\sqrt{1+\lambda^2 - 2\lambda \cos(\omega_c)}} \Rightarrow \cos(\omega_c) = \frac{1+\lambda^2 - 4(1-\lambda)^2}{2\lambda} \\ &\Rightarrow \omega_c = \arccos\left(\frac{-3+8\lambda-3\lambda^2}{2\lambda}\right) \end{aligned}$$

En la figura 12 es pot observar la dependència entre la freqüència de tall  $\omega_c$  del filtre passabaix i el paràmetre de suavització  $\lambda$ . Queda clar que valors del paràmetre de suavització més propers a 1 provoquen un mitjanament del senyal

més intens, és a dir, es filtra amb un filtre passabaix amb freqüència de tall menor. En canvi, per a valors més petits de  $\lambda$ , el mitjanament és més subtil, o el filtre deixa de seleccionar freqüències baixes per a deixar passar freqüències mitjanes o altes. Així, el paràmetre de control del filtre  $\lambda$  permet modificar fàcilment la manera com volem mitjanar o suavitzar el senyal donat.

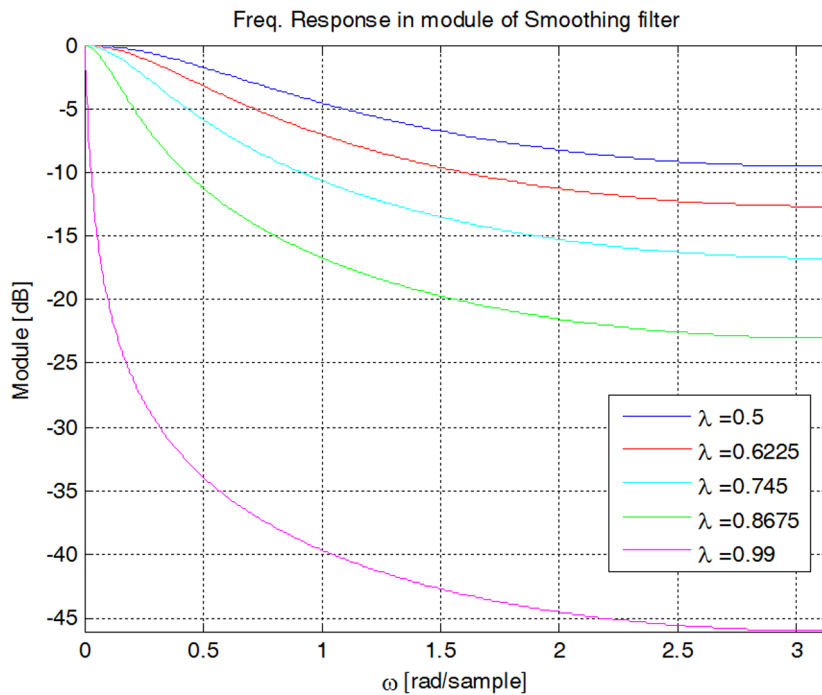
Figura 12. Relació entre el paràmetre de suavització  $\lambda$  i la freqüència de tall  $\omega_c$  del filtre de suavització



En la figura 13 es representa el mòdul del filtre de suavització per a diferents valors del paràmetre de suavització  $\lambda$ . Com es pot observar, el guany del filtre en tots els casos és de 0 dB (1 en valors lineals de guany), mentre que la caiguda del filtre és més abrupta com més proper a 1 és el valor del paràmetre de suavització. Aquest paràmetre no s'ha de fixar en el valor d'1 si es vol evitar que el filtre es faci inestable per a freqüències d'entrada iguals a 0; en aquest cas, la resposta del filtre seria creixent en el temps, de manera que l'estimació de l'energia augmentaria de manera indefinida.

Tornant a l'ús d'aquest tipus de filtre per al còmput de l'energia del senyal, si reprenem l'equació temporal (3) del filtre aplicat sobre el quadrat del senyal, podem veure que en el fons estem produint un mitjanament del quadrat, és a dir, estem d'alguna manera acumulant el senyal. Aquesta operació serveix com a substituta de l'operador valor esperat que apareixia en l'equació (2), de manera que podem establir un mètode per a ajustar la constant d'adaptació  $\mu$  de l'algorisme, de manera que s'asseguri la convergència del mètode i s'acceleri alhora la velocitat de convergència.

Figura 13. Mòdul de la resposta freqüencial del filtre de suavització per a cinc valors del paràmetre de suavització  $\lambda$



Passem ara a veure l'equació d'adaptació dels coeficients de l'estructura de Gram-Schmidt, segons la perspectiva de l'LMS. Si reprenem l'equació (1) d'adaptació de la versió de l'algorisme del gradient, la versió LMS d'aquesta equació és la següent:

$$b_{ik}[n] = b_{ik}[n-1] - \mu E[\varepsilon_k[n] \varepsilon_i[n]] \approx b_{ik}[n-1] - \mu \varepsilon_k[n] \varepsilon_i[n]$$

Com veiem, se substitueix l'operador valor esperat per una senzilla estimació instantània del senyal implicat, de manera que es delega el procés de mitjanament a l'estratègia mateixa d'iteració del mètode. Atès que el mètode consisteix a actualitzar la versió anterior de cada coeficient  $b_{ik}[n-1]$  afegint un nou terme  $-\mu \varepsilon_k[n] \varepsilon_i[n]$ , aquest procés acumulatiu ja representa en si un procés de mitjanament, que acabarà per determinar una direcció correcta envers la convergència del mètode.

## 5. RLS

El mètode RLS (de l'anglès, *recursive least squares*, 'mínims quadrats recursius') fa l'adaptació dels coeficients del filtre FIR d'ordre  $N$  ( $N + 1$  coeficients) mitjançant la minimització de la suma d'un error quadràtic acumulat segons una funció de ponderació exponencial decreixent cap al passat.

A diferència del mètode LMS, l'algorisme RLS parteix del plantejament d'una **funció de cost determinista**, és a dir, en la qual no intervé cap valor esperat. Aquesta funció es defineix com un error quadràtic mitjà ponderat amb un factor de memòria que permet que el filtre s'adapti millor al tipus de variacions temporals del sistema que s'intenta compensar o estimar. A mesura que es reben les mostres del senyal incident, es calcula la solució òptima del filtre de manera recursiva, i això dona lloc al mètode RLS.

L'algorisme RLS presenta **gran velocitat de convergència** respecte del mètode LMS, fins i tot quan la matriu de correlació del senyal d'entrada té autovalors molt dispersos, i presenta excel·lents característiques en entorns variants amb el temps. L'objectiu del mètode RLS és trobar els coeficients òptims de tal manera que el senyal de sortida del filtre sigui el més semblant possible al de referència segons el criteri dels mínims quadrats ponderats segons un **factor d'oblit** denominat  $\lambda$ .

### 5.1. Solució òptima

Definirem el vector  $\vec{x}[n]$  com un vector columna de  $\mathbb{R}^{N+1}$  que conté la informació del filtre en l'instant  $n$ , és a dir:

$$\vec{x}[n] = \begin{bmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-N] \end{bmatrix}$$

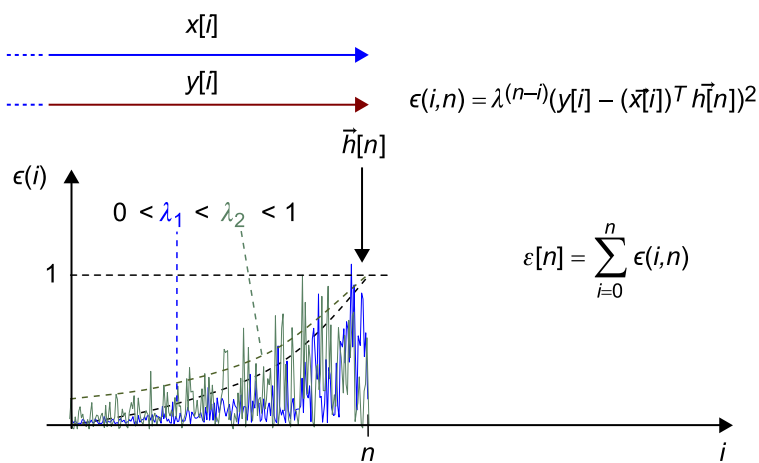
Els coeficients del filtre es calcularan minimitzant l'expressió següent d'un error determinista:

$$\epsilon[n] = \sum_{i=0}^n \lambda^{(n-i)} (y[i] - (\vec{x}[i]^T \vec{h}[n]))^2 \quad (4)$$

en què  $\vec{h}[n]$  és el vector de coeficients del filtre en l'instant  $n$ , i  $\lambda$  és el factor d'oblit, amb valors entre  $0 < \lambda < 1$ .

$\epsilon[n]$  és el denominat **error instantani del filtre** (en l'instant  $n$ ), i es tracta d'una norma definida positiva, que no pot donar lloc a valors negatius. A més, és una funció quadràtica dels coeficients del filtre,  $h_k[n]$ , en què  $k$  és l'índex de coeficient i  $n$  l'instant temporal en el qual es mesura, la qual cosa permet abordar la solució mitjançant sistemes d'equacions lineals. Aquesta nomenclatura, a diferència de la usada en la funció de cost del mètode LMS, deixa explícit que els coeficients del filtre varien temporalment, pel fet de ser l'algorisme RLS un mètode adaptatiu. Així doncs, tal com es llegeix de l'equació (4), l'error instantani del filtre depèn del valor dels coeficients en cada instant, però s'obté a partir d'una suma quadràtica de termes que està ponderada per un terme que decreix amb el temps passat. Aquest terme de ponderació depèn del **factor d'oblit**  $\lambda$ .

Figura 14. Representació gràfica del còmput de l'error quadràtic mitjà ponderat de la funció de cost RLS



Com es pot observar en la figura 14, l'error instantani,  $\epsilon(i, n)$ , en instants anteriors o iguals a l'actual  $i \leq n$ , es calcula amb la resposta impulsional actual del filtre,  $\vec{h}[n]$ , i l'entrada i la sortida en el temps  $i$ . D'aquesta manera, la resposta del filtre FIR  $\vec{h}[n]$  tindrà en compte un error acumulat però ponderat segons un factor exponencialment decreixent en el temps. Aquesta ponderació està marcada pel terme  $\lambda^{(n-i)}$ , que pren el valor 1 en  $i = n$ , i pren valors menors a mesura que  $i$  es fa més petit que  $n$ , o el que és el mateix, es consideren temps més passats a l'actual. Com es pot veure també en la mateixa figura, amb valors del factor d'oblit més propers a la unitat, es tindrà en compte l'error acumulat durant més temps passat, mentre que per a valors més propers a zero, l'error acumulat recollirà únicament el passat més recent.

Com es pot apreciar en l'equació d'error RLS (4), l'error acumulat es comença a comptabilitzar en  $i = 0$ , instant que se suposa, per conveni, l'instant inicial en el qual es disposa d'informació tant dels senyals d'entrada  $x[n]$  com del senyal desitjat  $y[n]$ . Derivant i igualant a zero l'expressió anterior, s'arriba al resultat següent:



$$\vec{h}[n] = \mathbf{R}_{xx}^{-1}[n] \vec{r}_{yx}[n] = \left( \underbrace{\sum_{i=0}^n \lambda^{(n-i)} \vec{x}[i] (\vec{x}[i])^T}_{\mathbf{R}_{xx}[n]} \right)^{-1} \underbrace{\left( \sum_{i=0}^n \lambda^{(n-i)} y[i] \vec{x}[i] \right)}_{\vec{r}_{yx}[n]} \quad (5)$$

en què  $\mathbf{R}_{xx}[n]$  és una matriu d'autocorrelació del senyal d'entrada  $x[n]$  i  $\vec{r}_{yx}[n]$  és un vector de correlació creuada entre el senyal desitjat  $y[n]$  i el senyal d'entrada  $x[n]$ . A diferència de la solució del filtre FIR òptim, en aquest cas tots dos termes no són estocàstics, ja que s'expressen a partir d'una equació determinista, o el que és el mateix, com una mitjana acumulada de valors passats i presents.

#### Vegeu també

Hem vist la solució del filtre FIR òptim en l'apartat 3.2. d'aquest mòdul didàctic.

#### Exercici

Deduïu la solució òptima del filtre RLS, és a dir, aquella solució del vector de coeficients  $\vec{h}[n]$  que minimitza l'error quadràtic  $\epsilon[n]$  (equació (5)).

#### Solució

Farem el tractament matemàtic per a derivar la solució òptima del filtre FIR-RLS sobre la base del principi d'ortogonalitat aplicat al producte escalar següent:

$$\langle a[i], b[i] \rangle = \sum_{i=0}^n \lambda^{(n-i)} a[i] b[i]$$

Podem reescriure l'error per minimitzar com una norma o producte escalar:

$$\epsilon[n] = \sum_{i=0}^n \lambda^{(n-i)} (y[i] - (\vec{x}[i])^T \vec{h}[n])^2 = \sum_{i=0}^n \lambda^{(n-i)} \epsilon_2[i] = \langle \epsilon[i], \epsilon[i] \rangle$$

en què:

$$\epsilon[i] = y[i] - (\vec{x}[i])^T \vec{h}[n]$$

Atès que l'error instantani  $\epsilon[i]$  és l'error entre el senyal desitjat  $y[i]$  i una combinació lineal, segons els coeficients  $h_k[n]$  del vector  $\vec{h}[n]$ , dels senyals  $x[i-k]$  per a  $0 \leq k \leq N$ , el valor mínim de la norma d'aquest error s'aconseguirà, segons el **teorema de la projecció ortogonal**, quan l'error  $\epsilon[i]$  aconseguit sigui ortogonal als senyals  $x[i-k]$  per a  $0 \leq k \leq N$ , és a dir:

$$\langle \epsilon[i], x[i-k] \rangle = 0 \quad 0 \leq k \leq N$$

Desenvolupem, doncs, aquest resultat:

$$\begin{aligned} \langle \epsilon[i], x[i-k] \rangle &= \sum_{i=0}^n \lambda^{(n-i)} x[i-k] \epsilon[i] = \\ &= \sum_{i=0}^n \lambda^{(n-i)} x[i-k] (y[i] - (\vec{x}[i])^T \vec{h}[n]) = \\ &= \sum_{i=0}^n \lambda^{(n-i)} y[i] x[i-k] - \sum_{i=0}^n \lambda^{(n-i)} x[i-k] (\vec{x}[i])^T \vec{h}[n] = \\ &= \sum_{i=0}^n \lambda^{(n-i)} y[i] x[i-k] - \left( \sum_{i=0}^n \lambda^{(n-i)} x[i-k] (\vec{x}[i])^T \right) \vec{h}[n] = 0 \end{aligned}$$

Si iterem l'equació anterior per als diferents valors de  $k$  dins del marge establert  $k \in [0, N]$ , podem reescriure les  $N + 1$  equacions de manera compacta amb una única equació matricial, que és la següent:

$$\sum_{i=0}^n \lambda^{(n-i)} y[i] \vec{x}[i] - \left( \sum_{i=0}^n \lambda^{(n-i)} \vec{x}[i] (\vec{x}[i])^T \right) \vec{h}[n] = 0$$

en què, com es pot observar, s'han substituït els valors del senyal d'entrada pel vector de senyal  $\vec{x}[i]$ .

Si multipliquem l'equació anterior per la matriu inversa de la matriu que multiplica per l'esquerra el vector de coeficients del filtre, s'arriba a la solució òptima del filtre RLS (equació (5)).

L'equació (5) s'ha de llegir de la manera següent: el primer terme o sumatori és un vector, ja que es tracta d'una suma de factors escalars que ponderen valors d'un senyal ( $y[i]$ ) per vectors de senyal d'entrada ( $\vec{x}[i]$ ). En canvi, el segon terme és una matriu, formada per una suma de matrius, i cadascuna d'aquestes matrius és el resultat del producte de dos vectors de senyal, un vector columna  $\vec{x}[i]$  per un vector fila  $(\vec{x}[i])^T$ , i el factor associat a la ponderació exponencial decreixent, que és funció del factor d'oblit.

Una observació important per remarcar a aquestes alçades de l'explicació és que la matriu d'autocorrelació del senyal d'entrada,  $\mathbf{R}_{xx}[n]$ , és de dimensions  $(N + 1) \times (N + 1)$ , per la qual cosa **la seva inversió en cada temps de mostra requereix un cost computacional d'ordre  $O((N + 1)^3)$** , per exemple, usant un algorisme com el d'eliminació de Gauss-Jordan. Aquesta complexitat computacional pot ser excessiva, per la qual cosa en el subapartat següent s'analitza la solució adaptativa del mètode RLS, que permet reduir l'ordre de complexitat de càlcul de la solució òptima.

## 5.2. Solució adaptativa

Per a arribar a la solució adaptativa del filtre RLS és necessari estudiar primer la identitat de Woodbury, també anomenada **lema d'inversió de matrius**. Aquest teorema ens permet simplificar la inversió de la matriu d'autocorrelació per a arribar a una versió adaptativa, o incremental, que repercutirà en un algorisme de complexitat computacional molt menor.

### Lema d'inversió de matrius

Siguin les matrius  $\mathbf{A} \in \mathbb{C}^{N \times N}$  i  $\mathbf{R} \in \mathbb{C}^{M \times M}$  no singulars<sup>1</sup>, i  $\mathbf{X} \in \mathbb{C}^{N \times M}$  i  $\mathbf{Y} \in \mathbb{C}^{M \times N}$ . Sigui la matriu  $\mathbf{B} \in \mathbb{C}^{N \times N}$  definida per l'equació següent:

$$\mathbf{B} = \mathbf{A} + \mathbf{XRY}$$

Llavors,  $\mathbf{B}$  és també una matriu no singular, i la seva inversa es pot expressar com:

<sup>(1)</sup>Una matriu (quadrada) no singular és una matriu invertible, ja que es compleix que el seu determinant és diferent a 0, o que el seu rang (o nombre de vectors fila o columna linealment independents) és igual a la seva dimensió.

$$\mathbf{B}^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{X} (\mathbf{R}^{-1} + \mathbf{Y} \mathbf{A}^{-1} \mathbf{X})^{-1} \mathbf{Y} \mathbf{A}^{-1}$$

Aquest teorema es pot demostrar de manera directa verificant que la matriu  $\mathbf{B}$  multiplicada per la seva inversa dóna com a resultat la matriu identitat, aspecte que us proposem com a exercici.

### Solució adaptativa per a la matriu inversa de correlació

El lema d'inversió de matrius permet aconseguir un càlcul recursiu de la matriu d'autocorrelació  $\mathbf{R}_{xx}^{-1}[n]$ . Primer convé desglossar la matriu com una suma de dos termes, un d'associat al passat, i un altre al present:

$$\mathbf{R}_{xx}[n] = \sum_{i=0}^n \lambda^{(n-i)} \vec{x}[i] (\vec{x}[i])^T = \underbrace{\lambda \sum_{i=0}^{n-1} \lambda^{(n-1-i)} \vec{x}[i] (\vec{x}[i])^T}_{\text{Passat}} + \underbrace{\vec{x}[n] (\vec{x}[n])^T}_{\text{Present}}$$

Com es pot observar, el terme associat al passat és una versió anterior d'aquesta matriu, ja que:

$$\mathbf{R}_{xx}[n] = \lambda \mathbf{R}_{xx}[n-1] + \vec{x}[n] \times 1 \times (\vec{x}[n])^T$$

en què 1 és una matriu identitat de  $\mathbb{C}^{1 \times 1}$ , que coincideix amb les dimensions de nombre de columnes de  $\vec{x}[n]$  i de files de  $(\vec{x}[n])^T$ , o sigui, l'escalar 1.

Si observem detingudament l'equació anterior, podem identificar els termes associats a la matriu  $\mathbf{B}$  del lema d'inversió de matrius, concretament:

$$\mathbf{A} = \lambda \mathbf{R}_{xx}[n-1] \quad \mathbf{X} = \vec{x}[n] \quad \mathbf{R} = 1 \quad \mathbf{Y} = (\vec{x}[n])^T$$

Amb això, podem expressar la matriu inversa de la matriu d'autocorrelació de manera recursiva aplicant el lema d'inversió:

$$\begin{aligned} \mathbf{R}_{xx}^{-1}[n] &= \left(\frac{1}{\lambda}\right) \mathbf{R}_{xx}^{-1}[n-1] - \left(\frac{1}{\lambda}\right) \mathbf{R}_{xx}^{-1}[n-1] \vec{x}[n] \times \\ &\times \left(1^{-1} + (\vec{x}[n])^T \left(\frac{1}{\lambda}\right) \mathbf{R}_{xx}^{-1}[n-1] \vec{x}[n]\right)^{-1} (\vec{x}[n])^T \left(\frac{1}{\lambda}\right) \mathbf{R}_{xx}^{-1}[n-1] = \\ &= \left(\frac{1}{\lambda}\right) \mathbf{R}_{xx}^{-1}[n-1] - \left(\frac{1}{\lambda}\right) \mathbf{R}_{xx}^{-1}[n-1] \frac{\vec{x}[n] (\vec{x}[n])^T \mathbf{R}_{xx}^{-1}[n-1]}{\lambda + (\vec{x}[n])^T \mathbf{R}_{xx}^{-1}[n-1] \vec{x}[n]} = \\ &= \left(\frac{1}{\lambda}\right) (\mathbf{R}_{xx}^{-1}[n-1] - \mathbf{K}[n] (\vec{x}[n])^T \mathbf{R}_{xx}^{-1}[n-1]) \end{aligned}$$

$$\mathbf{K}[n] = \frac{\mathbf{R}_{xx}^{-1}[n-1] \vec{x}[n]}{\lambda + (\vec{x}[n])^T \mathbf{R}_{xx}^{-1}[n-1] \vec{x}[n]}$$

en què el vector columna  $\mathbf{K}[n] \in \mathbb{C}^{1 \times (N+1)}$  és denominat **guany de Kalman**.

Fixeu-vos com, en aquest cas, en ser el terme entre parèntesis associat a  $(\mathbf{R}^{-1} + \mathbf{Y}\mathbf{A}^{-1}\mathbf{X})$  de dimensions  $1 \times 1$ , la seva inversió és directament el mateix terme com a quocient que multiplica la resta. Observeu que aquesta equació permet reduir la complexitat computacional de la inversió, que originalment era d'ordre  $O((N+1)^3)$  a un altre menor  $O((N+1)^2)$  associat als productes matrixials que apareixen, sempre que es disposi de la matriu de correlació inversa de la iteració anterior,  $\mathbf{R}_{xx}^{-1}[n-1]$ .

### Teorema

El guany de Kalman es pot expressar com la matriu inversa de correlació multiplicada pel vector de senyal:

$$\mathbf{K}[n] = \mathbf{R}_{xx}^{-1}[n] \bar{\mathbf{x}}[n]$$

### Demostració

Partint de l'equació adaptativa anterior, en la qual la matriu inversa de correlació s'expressa de manera recursiva a partir del seu valor anterior, desenvolupem i obtenim:

$$\begin{aligned} \mathbf{R}_{xx}^{-1}[n] \bar{\mathbf{x}}[n] &= \left(\frac{1}{\lambda}\right) (\mathbf{R}_{xx}^{-1}[n-1] - \mathbf{K}[n] \bar{\mathbf{x}}[n]^T \mathbf{R}_{xx}^{-1}[n-1]) \bar{\mathbf{x}}[n] = \\ &= \left(\frac{1}{\lambda}\right) (\mathbf{R}_{xx}^{-1}[n-1] \bar{\mathbf{x}}[n] - \mathbf{K}[n] \bar{\mathbf{x}}[n]^T \mathbf{R}_{xx}^{-1}[n-1] \bar{\mathbf{x}}[n]) = \\ &= \left(\frac{1}{\lambda}\right) (\mathbf{R}_{xx}^{-1}[n-1] \bar{\mathbf{x}}[n] - (\mathbf{R}_{xx}^{-1}[n-1] \bar{\mathbf{x}}[n] - \lambda \mathbf{K}[n])) = \mathbf{K}[n] \end{aligned}$$

en què, en la tercera igualtat s'ha substituït el terme  $\mathbf{K}[n] \bar{\mathbf{x}}[n]^T \mathbf{R}_{xx}^{-1}[n-1] \bar{\mathbf{x}}[n]$  pel valor que s'obté aïllant de l'equació que defineix el guany de Kalman.

## Solució adaptativa del mètode RLS

Arribats al punt d'expressar la matriu d'autocorrelació de manera recursiva a partir de la seva versió temporal anterior, podem arribar a la solució adaptativa del filtre RLS de manera senzilla. En primer lloc, expressem el vector de correlació creuada de manera adaptativa, tal com es va fer amb la matriu d'autocorrelació, desglossant-lo en els termes associats al passat i el terme associat al present, això és:

$$\begin{aligned} \vec{r}_{yx}[n] &= \sum_{i=0}^n \lambda^{(n-i)} y[i] \bar{\mathbf{x}}[i] = \sum_{i=0}^{n-1} \lambda^{(n-i)} y[i] \bar{\mathbf{x}}[i] + y[n] \bar{\mathbf{x}}[n] = \\ &= \lambda \underbrace{\sum_{i=0}^{n-1} \lambda^{(n-1-i)} y[i] \bar{\mathbf{x}}[i]}_{\vec{r}_{yx}[n-1]} + y[n] \bar{\mathbf{x}}[n] = \lambda \vec{r}_{yx}[n-1] + y[n] \bar{\mathbf{x}}[n] \end{aligned}$$

A continuació tornem a expressar la solució òptima del filtre RLS i comencem per substituir el vector de correlació creuada per la seva versió adaptativa:

$$\vec{h}[n] = \mathbf{R}_{xx}^{-1}[n] \vec{r}_{yx}[n] = \mathbf{R}_{xx}^{-1}[n] (\lambda \vec{r}_{yx}[n-1] + y[n] \vec{x}[n]) = \lambda \mathbf{R}_{xx}^{-1}[n] \vec{r}_{yx}[n-1] + \mathbf{R}_{xx}^{-1}[n] y[n] \vec{x}[n]$$

A continuació, substituïm la matriu inversa de correlació per la seva versió adaptativa en el primer terme de l'equació anterior:

$$\begin{aligned} \vec{h}[n] &= \lambda \left( \frac{1}{\lambda} \right) (\mathbf{R}_{xx}^{-1}[n-1] - \mathbf{K}[n] (\vec{x}[n])^T \mathbf{R}_{xx}^{-1}[n-1]) \vec{r}_{yx}[n-1] + \mathbf{R}_{xx}^{-1}[n] y[n] \vec{x}[n] = \\ &= \mathbf{R}_{xx}^{-1}[n-1] \vec{r}_{yx}[n-1] - \mathbf{K}[n] (\vec{x}[n])^T \mathbf{R}_{xx}^{-1}[n-1] \vec{r}_{yx}[n-1] + \mathbf{R}_{xx}^{-1}[n] \vec{x}[n] y[n] \end{aligned}$$

En l'equació anterior podem identificar dos productes que es corresponen amb la definició de la solució òptima del filtre RLS per a l'instant de temps  $n-1$ , és a dir,  $\vec{h}[n-1] = \mathbf{R}_{xx}^{-1}[n-1] \vec{r}_{yx}[n-1]$ , per la qual cosa:

$$\begin{aligned} \vec{h}[n] &= \vec{h}[n-1] - \mathbf{K}[n] (\vec{x}[n])^T \vec{h}[n-1] + \frac{\mathbf{R}_{xx}^{-1}[n] \vec{x}[n] y[n]}{\mathbf{K}[n]} = \\ &= \vec{h}[n-1] + \mathbf{K}[n] (y[n] - (\vec{x}[n])^T \vec{h}[n-1]) = \vec{h}[n-1] + \mathbf{K}[n] e[n] \end{aligned}$$

En aquest desenvolupament s'ha utilitzat el teorema anterior per a arribar a la solució final.

Com es pot observar en l'equació d'adaptació del filtre RLS, la resposta impulsionals s'adapta a partir de l'error de sortida, que té en compte la resposta del filtre en el temps anterior,  $e[n] = y[n] - (\vec{x}[n])^T \vec{h}[n-1]$ , i el guany de Kalman. Arribats a aquest punt, convé fer algunes substitucions, tenint el compte l'equació que defineix el guany de Kalman, per a veure certes similituds amb l'equació d'adaptació del filtre LMS.

$$\vec{h}[n] = \vec{h}[n-1] + \mu[n] \mathbf{R}_{xx}^{-1}[n-1] \vec{x}[n] e[n]$$

en què  $\mu[n] = \frac{1}{\lambda + (\vec{x}[n])^T \mathbf{R}_{xx}^{-1}[n-1] \vec{x}[n]}$  és una constant d'adaptació variable.

### Interpretació

A continuació, deduïda l'expressió iterativa per a l'adaptació del filtre FIR segons el criteri dels mínims quadrats recursius, podem fer la interpretació dels termes que hi apareixen, contrastant-los amb els que apareixen en l'equació d'adaptació del filtre adaptatiu que fa ús del criteri LMS.

#### Vegeu també

Hem vist l'equació d'adaptació del filtre LMS en l'apartat 3 d'aquest mòdul didàctic.

Per començar, veiem que si comparem aquestes dues versions, criden l'atenció dos aspectes molt concrets: l'existència d'un paràmetre d'adaptació  $\mu$  adaptatiu per a la solució RLS, en lloc de ser un paràmetre constant, com ho és per a la versió LMS; i l'existència d'una matriu de descorrelació de les dades d'entrada,  $\mathbf{R}_{xx}^{-1}[n-1]$ , per a la versió RLS, que no existeix per al filtre LMS.

A continuació, podem concretar què ens aporten totes dues diferències:

- **Matriu de descorrelació del senyal d'entrada:** aquesta matriu multiplica per l'esquerra el vector de mostres de senyal d'entrada,  $\vec{x}[n]$ ; la seva missió és justament aconseguir descorrelacionar les dades d'entrada i obtenir, per tant, un vector  $\vec{w}[n] = \mathbf{R}_{xx}^{-1}[n-1]\vec{x}[n]$  de  $N+1$  variables descorrelacionades entre si. Aquest procés permet que la convergència de la solució RLS s'enfoqui de manera més precisa en el sentit del mínim error quadràtic mitjà definit al principi de tot, i evita així les convergències més lentes típiques de la solució LMS, especialment quan el vector de senyal d'entrada està format per variables molt correlacionades entre si.

Recordem, a més, que la versió adaptativa del filtre RLS s'ha deduït sense fer cap aproximació, a diferència de l'algorisme LMS, el qual sí que és una versió aproximada de l'algorisme del gradient. Per això, el filtre RLS aconseguix en cada iteració minimitzar l'error definit de manera instantània i sense fer ús de valors esperats.

- **Paràmetre d'adaptació  $\mu$  intel·ligent:** a diferència de la versió LMS, el filtre RLS adaptatiu disposa d'un paràmetre d'aprenentatge intel·ligent, que creix o decreix en funció de la similitud de les dades noves proporcionades pel que fa a les dades processades fins al moment. Quan les dades que processa el filtre, que estan incloses dins de les posicions del vector de senyal  $\vec{x}[n]$ , són dades similars a les obtingudes fins al moment, el terme  $(\vec{x}[n])^T \mathbf{R}_{xx}^{-1}[n-1]\vec{x}[n]$  donarà lloc a un valor elevat, que farà que el paràmetre d'adaptació tingui un valor petit, i això permetrà aproximar-se millor a un error cada vegada menor. Això succeirà, per exemple, durant els períodes de temps en què el sistema que se segueix (en el cas de l'aplicació del filtratge adaptatiu, el problema de l'estimació de sistemes lineals variants en el temps) no tingui variacions temporals significatives, i s'aconseguirà una bona aproximació a l'error mínim. Ara bé, quan el sistema varia de manera abrupta és necessari augmentar el valor del paràmetre d'adaptació per a aconseguir seguir millor el sistema, i sacrificar capacitat d'aproximar-se de manera precisa a l'error mínim.

Això últim significa que el terme  $(\vec{x}[n])^T \mathbf{R}_{xx}^{-1}[n-1]\vec{x}[n]$  donarà valors petits, ja que les últimes mostres de senyal observades pel filtre estaran poc correlacionades amb les observades fins llavors, per la qual cosa el paràmetre  $\mu$  donarà valors més grans i s'aconsegueix així la capacitat de seguir grans canvis en els coeficients del sistema.

## Resum de l'algorisme RLS adaptatiu

A continuació es resumeixen les equacions del filtre RLS adaptatiu, en què la matriu inversa de correlació del senyal d'entrada,  $\mathbf{R}_{xx}^{-1}$ , ha estat reanomenada com a  $\mathbf{P}$ :

### 1) Inicialització

$$\mathbf{P}[-1] = \delta \mathbf{I} \quad (\delta = 1/\hat{P}_x)$$

$$\vec{h}[-1] = \vec{x}[-1] = [0, 0, \dots, 0]^T$$

2) Per a  $n \geq 0$ :

$$\mathbf{K}[n] = \frac{\mathbf{P}[n-1]\vec{x}[n]}{\lambda + (\vec{x}[n])^T \mathbf{P}[n-1]\vec{x}[n]}$$

$$\mu[n] = \frac{1}{\lambda + (\vec{x}[n])^T \mathbf{P}[n-1]\vec{x}[n]}$$

$$\vec{h}[n] = \vec{h}[n-1] + \mu[n]\mathbf{P}[n-1]\vec{x}[n]e[n]$$

$$\mathbf{P}[n] = \left(\frac{1}{\lambda}\right) \left(\mathbf{P}[n-1] - \mathbf{K}[n](\vec{x}[n])^T \mathbf{P}[n-1]\right)$$

La inicialització del filtre RLS implica la posada a condicions inicials zero, tant per al vector de mostres de senyal com per al vector associat a la resposta impulsional del filtre FIR. A més, la matriu inversa de correlació es pot inicialitzar a una matriu diagonal multiplicada pel factor d'inicialització  $\delta$ , que es pot correspondre amb l'invers de l'estimació de la potència de senyal d'entrada ( $1/\hat{P}_x$ ). Ara bé, aquest factor tampoc no és crític, i generalment hi ha un bon rang de valors que es pot usar com a valor d'inicialització.

Quant a les equacions de funcionament del filtre RLS, podem veure en primer lloc el càlcul del vector guany de Kalman,  $\mathbf{K}[n]$ , les equacions de càlcul del paràmetre d'adaptació  $\mu[n]$  i també el vector de coeficients del filtre adaptatiu  $\vec{h}[n]$ , i finalment l'equació recursiva associada al càlcul de la matriu inversa de correlació del senyal d'entrada,  $\mathbf{P}[n]$ , que s'usarà en la iteració següent del filtre.





## Activitats

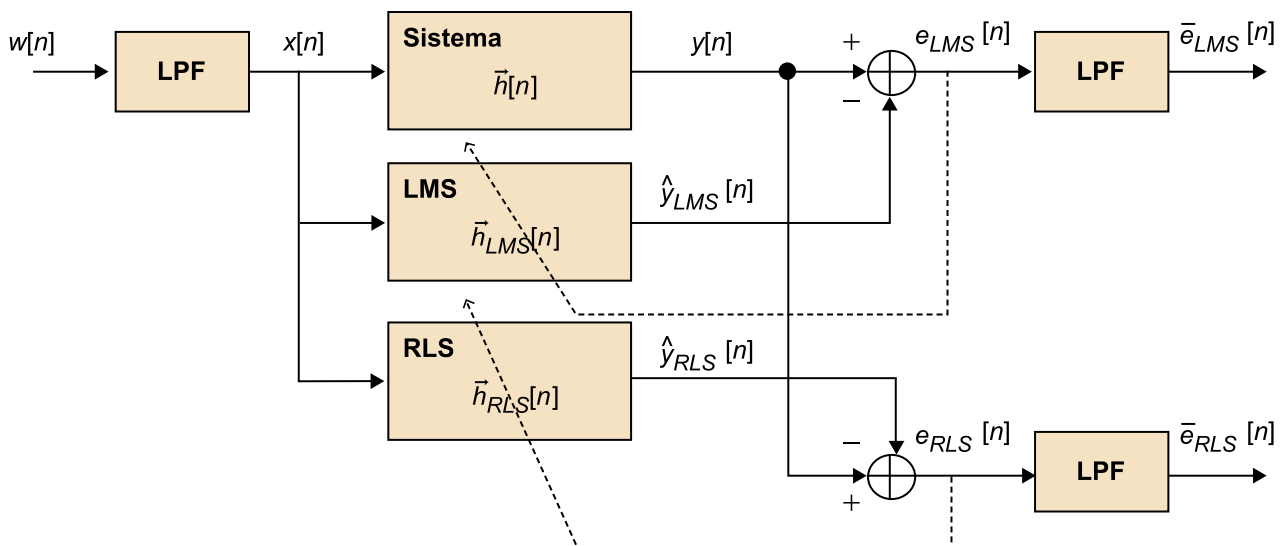
Programeu amb el Matlab els algorismes LMS i RLS aplicats a un problema d'identificació de sistemes lineals i variants en el temps (de tipus FIR). Estudieu per mitjà de la simulació la velocitat de convergència de tots dos algorismes i la seva dependència respecte de les característiques de correlació del senyal d'entrada. Per a això, cal comparar tant l'error de sortida  $e[n] = y[n] - \hat{y}[n]$  de tots dos algorismes, com també l'error d'estimació de la resposta impulsional del sistema, definit com:

$$e_h[n] = \|\vec{h}[n] - \hat{h}[n]\|_2^2$$

en què  $\vec{h}[n]$  és el vector de la resposta impulsional del sistema, i  $\hat{h}[n]$  el vector de la resposta impulsional de l'estimador, tots dos sistemes de tipus FIR.

L'esquema és el que es dibuixa en la figura 15. Com es pot observar, el senyal d'entrada al sistema,  $x[n]$ , es generarà a partir d'un senyal de soroll blanc i gaussià de variància 1 i mitjana 0 ( $w[n]$ ) filtrat per mitjà d'un filtre passabaix (LPF) de suavització com el que hem analitzat en el subapartat 4.4.2. Aquest filtre permetrà introduir certa correlació en el senyal d'entrada al sistema, i també en els filtres adaptatius que fan la tasca d'estimació o identificació, atès que aquests estan situats en paral·lel juntament amb el sistema i tenen el mateix senyal d'entrada. El factor  $\lambda_x$  del filtre LPF del senyal d'entrada permetrà generar un senyal descorrelacionat (per a  $\lambda_x = 0$  tenim un filtre passatot) o bé un senyal molt correlacionat ( $\lambda_x \rightarrow 1$ , tenim un filtre passabaix amb freqüència de tall molt petita).

Figura 15. Diagrama de blocs de l'exercici de simulació



Els errors de sortida dels filtres adaptatius també cal mitjanar-los amb filtres de suavització, amb la finalitat de poder destriar més bé les diferències entre el comportament de tots dos filtres.

De la mateixa manera, el sistema és un sistema FIR (la longitud en nombre de coeficients cal fixar-la igual al nombre de coeficients dels estimadors LMS i RLS) que varia amb el temps.

La variabilitat del sistema s'ha d'implementar a partir d'un altre filtre de suavització, però ha de treballar amb tot el vector de mostres, que s'alimentarà a partir d'un vector aleatori amb soroll blanc gaussià. Per a generar les mostres d'entrada al filtre, cada mostra del vector ha d'estar totalment descorrelacionada de les altres. L'equació d'adaptació del model de sistema variant és la següent:

$$\vec{h}[n] = \lambda_s \vec{h}[n-1] + (1 - \lambda_s) \vec{q}[n]$$

en què  $\vec{q}[n]$  és un vector amb mostres de soroll blanc gaussià (descorrelacionades entre si) de variància 1 i mitjana 0.

D'aquesta manera, la constant  $\lambda_s$  permetrà controlar la velocitat de variació del sistema, i aquesta serà elevada si  $\lambda_s \rightarrow 0$  (en aquest cas,  $\vec{h}[n] = \vec{q}[n]$ , i cada component del vector  $\vec{h}[n]$  serà totalment independent dels seus valors anteriors), mentre que el sistema variarà més

lentament si  $\lambda_s \rightarrow 1$  (el soroll blanc gaussià filtrat amb un filtre passabaix de freqüència de tall baixa tindrà una evolució més lenta).

La funció ha de ser un *script* en el qual es defineixin les variables de simulació següents:

- Paràmetres generals de simulació
  - Nombre de coeficients del sistema ( $N$ )
  - Nombre de mostres de senyal per simular ( $M$ )
  - Factor de correlació del senyal d'entrada ( $\text{Lambda}_x$ )
  - Nombre de mostres entre visualitzacions de l'error ( $L$ )
  - Factor de mitjanament per a l'error dels filtres ( $\text{Lambda}_E$ )
- Paràmetres del sistema variant amb el temps
  - Factor de memòria del sistema ( $\text{Lambda}_S$ )
- Paràmetres de l'algorisme LMS
  - Constant d'adaptació ( $\mu_{LMS}$ )
- Paràmetres del filtre RLS
  - Constant d'inicialització de la matriu  $P$  ( $\text{delta}_{RLS}$ )
  - Factor de memòria ( $\text{Lambda}_{RLS}$ )

El paràmetre  $L$  permetrà modificar el nombre de mostres entre visualitzacions dels errors, tant de sortida com d'estimació. La funció *script* ha de mostrar els resultats mitjançant una figura en la qual cal dibuixar:

- L'autocorrelació dels senyals  $w[n]$  i  $x[n]$ , amb un retard de  $\pm 20$  mostres.
- Els errors de sortida de tots dos filtres mitjanats amb el filtre de suavització ( $\bar{e}_{LMS}[n]$  i  $\bar{e}_{RLS}[n]$ ).
- Els errors d'estimació de tots dos filtres mitjanats amb el filtre de suavització ( $\bar{e}_h^{LMS}[n]$  i  $\bar{e}_h^{RLS}[n]$ ).
- La resposta impulsional original i l'estimada per a tots dos filtres ( $\vec{h}[n]$ ,  $\vec{h}_{LMS}[n]$ ,  $\vec{h}_{RLS}[n]$ ).

Compareu els errors de sortida i d'estimació de tots dos sistemes per a un ordre del sistema  $N = 5$ ; escolliu un nombre total de mostres de  $M = 10.000$ , un nombre de mostres entre visualitzacions de l'error de  $L = 100$ , i un factor de mitjanament dels errors dels filtres de  $\text{Lambda}_E = 0,99$ . Feu l'estudi dels errors de sortida tant per a un sistema invariant en el temps ( $\text{Lambda}_S = 1$ ) com per a un sistema variant (amb  $\text{Lambda}_S = 0,999$ ), i també tant per a un senyal d'entrada descorrelacionat ( $\text{Lambda}_x = 0$ ) com per a un senyal d'entrada correlacionat ( $\text{Lambda}_x = 0,95$ ). En cada escenari, escolliu les constants d'adaptació dels filtres ( $\mu$  del filtre LMS, i  $\lambda$  del filtre RLS) que garanteixin una convergència ràpida i també l'estabilitat de l'algorisme.

### Nota

Per simplicitat, en aquest exercici pràctic es defineix  $N$  com el número de coeficients del filtre FIR, tot i que en l'estudi teòric s'ha definit el nombre de coeficients com  $N + 1$ .

## Solucionari

A continuació es mostra el codi de l'*script* Matlab que fa la simulació que es demana en l'exercici. Primer es mostra la part del codi en què es defineixen les constants o paràmetres de simulació, on es pot observar que se simula una situació en la qual el senyal d'entrada al sistema i als filtres està correlacionat ( $\text{Lambda}_x = 0,95$ ).

La inicialització inclou:

- La dels vectors de resposta impulsional, de  $N$  mostres, tant dels filtres com del sistema ( $h_{LMS}$ ,  $h_{RLS}$ ,  $h$ ); la dels sistemes és un vector columna de zeros, mentre que la del sistema és un vector columna aleatori (utilitzant la funció *randn*).
- La matriu  $P$  del filtre *RLS* (utilitzant la funció *eye*, que defineix una matriu identitat).
- Vectors de mostres per a la visualització dels diferents senyals de  $M$  o de  $N$  punts ( $nv$ ,  $nh$ ).
- Vector de  $N$  mostres consecutives del senyal d'entrada,  $xv$ , inicialitzat a zeros, que s'utilitzarà tant per a generar la sortida del sistema com la de tots dos filtres adaptatius.
- Vectors de  $M$  mostres, associats als errors de sortida i d'estimació de tots dos filtres adaptatius ( $e_{LMS}$ ,  $e_{RLS}$ ,  $eh_{LMS}$ ,  $eh_{RLS}$ ), inicialitzats a vectors amb zeros.

```

%% Paràmetres generals de la simulació
%-----
N = 5;                % Nombre de coeficients del sistema
M = 10000;           % Nombre de mostres per simular
Lambda_x = 0.95;    % Factor de correlació del senyal d'entrada
L = 100;             % Nombre de mostres entre visualitzacions de l'error
Lambda_E = 0.99;    % Factor de mitjanament per a l'error dels filtres
%% Sistema variant amb el temps
%-----
Lambda_S = 0.999;   % Factor de memòria del sistema

%% Paràmetres del filtre LMS
%-----
mu_LMS = 0.1;       % Constant d'adaptació

%% Paràmetres del filtre RLS
%-----
Lambda_RLS = 0.8;   % Factor de memòria
delta_RLS = 1;     % Constant d'inicialització de la matriu P

%% Inicialització
%-----
h_LMS = zeros(N,1);
h_RLS = zeros(N,1);
P = delta_RLS*eye(N);
h = randn(N,1);
nv = 1:M;
nh = 1:N;
xv = zeros(N,1);
e_LMS = zeros(M,1);
e_RLS = zeros(M,1);
eh_LMS = zeros(M,1);
eh_RLS = zeros(M,1);
minh = min(h);

%% Generació del senyal d'entrada
w = randn(M,1);

% Filtratge passabaix del senyal (correlació)
x = filter(1-Lambda_x, [1 -Lambda_x], w);

% Càlcul correlacions original i senyal filtrat
Rww = xcorr(w,20,'unbiased');
Rxx = xcorr(x,20,'unbiased');
maxh = max(x);

%% Visualització resultats
figure(1);clf
subplot(221);plot(-20:20,Rww/max(Rww),-20:20,Rxx/max(Rxx));axis
tight;grid;title('Autocorrelación');
legend('Original','Entrada x[n]')

```

La segona part del codi mostrada és la que fa el bucle general de la simulació. En la visualització dels resultats s'ha emprat la funció *semilogy*, que permet usar un eix d'abscisses logarímic, més apropiat per a mostrar els errors tant de sortida com d'estimació.

Per a cadascuna de les 4 situacions simulades (entrada correlacionada o incorrelacionada, i sistema invariant o variant) s'han escollit els valors de les constants d'adaptació ( $\mu_{LMS}$  i

$\lambda_{RLS}$  que permeten obtenir errors d'estimació menors. En el cas de l'algorisme RLS, la constant  $\lambda_{RLS}$  es defineix entre 0 i 1, mentre que en el cas de l'algorisme LMS es pot escollir una  $\mu_{LMS}$  més gran que 1 per a accelerar encara més la convergència. No obstant això, un valor massa elevat pot produir inestabilitat de l'algorisme, aspecte que cal evitar.

```

%% Bucle de simulació general
for n = 1:M
    % Sistema
    %-----

    xv = [x(n);xv(1:(N-1))];
    y = transpose(xv)*h;
    h = Lambda_S*h + (1-Lambda_S)*randn(N,1);

    % Filtre LMS
    %-----

    y_LMS = transpose(xv)*h_LMS;
    ei_LMS = (y - y_LMS);
    e_LMS(n+1) = e_LMS(n)*Lambda_E + (1-Lambda_E)*ei_LMS;
    h_LMS = h_LMS + mu_LMS*xv*ei_LMS;

    eih_LMS = norm(h - h_LMS)^2;
    eh_LMS(n+1) = eh_LMS(n)*Lambda_E + (1-Lambda_E)*eih_LMS;

    % Filtre RLS
    %-----

    mu_RLS = 1/(Lambda_RLS + transpose(xv)*P*xv);
    K = (P*xv)*mu_RLS;
    y_RLS = transpose(xv)*h_RLS;
    ei_RLS = (y - y_RLS);
    e_RLS(n+1) = e_RLS(n)*Lambda_E + (1-Lambda_E)*ei_RLS;
    h_RLS = h_RLS + mu_RLS*P*xv*ei_RLS;
    P = (1/Lambda_RLS)*(P - K*transpose(xv)*P);

    eih_RLS = norm(h - h_RLS)^2;
    eh_RLS(n+1) = eh_RLS(n)*Lambda_E + (1-Lambda_E)*eih_RLS;

    %% Visualització resultats
    if (max(h)>maxh)
        maxh = max(h);
    end
    if (min(h)<minh)
        minh = min(h);
    end
    if (mod(n,L) == 0)
        figure(1);
        subplot(222);semilogy(nv,abs(e_LMS),nv,abs(e_RLS));
        axis tight;grid;title('Error sortida')
        legend('LMS','RLS');
        subplot(223);plot(nh,h,nh,h_LMS,nh,h_RLS);
        axis tight;grid;title('Resposta impulsional')
        %axis([1 N minh*0.7 maxh*0.7])
        legend('Sistema','LMS','RLS');
        subplot(224);semilogy(nv,abs(eh_LMS),nv,abs(eh_RLS));
        axis tight;grid;title('Error estimació')
        legend('LMS','RLS');
    end
end
end

```

En la taula següent es mostren els valors dels paràmetres que permeten obtenir les figures de resultats que es mostren a continuació.

	$\lambda_x$	$\lambda_S$	$\mu_{LMS}$	$\lambda_{RLS}$
Sistema invariant i entrada incorrelacionada	0	1	0,2	0,9
Sistema invariant i entrada correlacionada	0,95	1	2	0,8
Sistema variant i entrada incorrelacionada	0	0,999	0,1	0,7
Sistema variant i entrada correlacionada	0,95	0,999	0,1	0,7

En tots els casos s'han escollit els factors d'adaptació ( $\mu_{LMS}$  i  $\lambda_{RLS}$ ) que aconseguen una convergència ràpida sense arribar a divergir. A continuació, en les figures 16, 17,

18 i 19 es mostren els resultats obtinguts per a cadascuna de les 4 simulacions fetes. Com es pot observar, en tots els casos l'algorisme RLS aconsegueix errors de sortida i d'estimació de la resposta impulsional menors, encara que aquesta diferència s'accentua especialment quan el senyal d'entrada està correlacionat. Recordem que l'algorisme RLS és una tècnica que millora la convergència en aplicar una descorrelació de les dades d'entrada de manera dinàmica, i que la seva constant d'adaptació intel·ligent permet adaptar-se millor a les condicions de variació d'un sistema dinàmic.

Figura 16. Resultats de la simulació amb un sistema invariant i entrada incorrelacionada

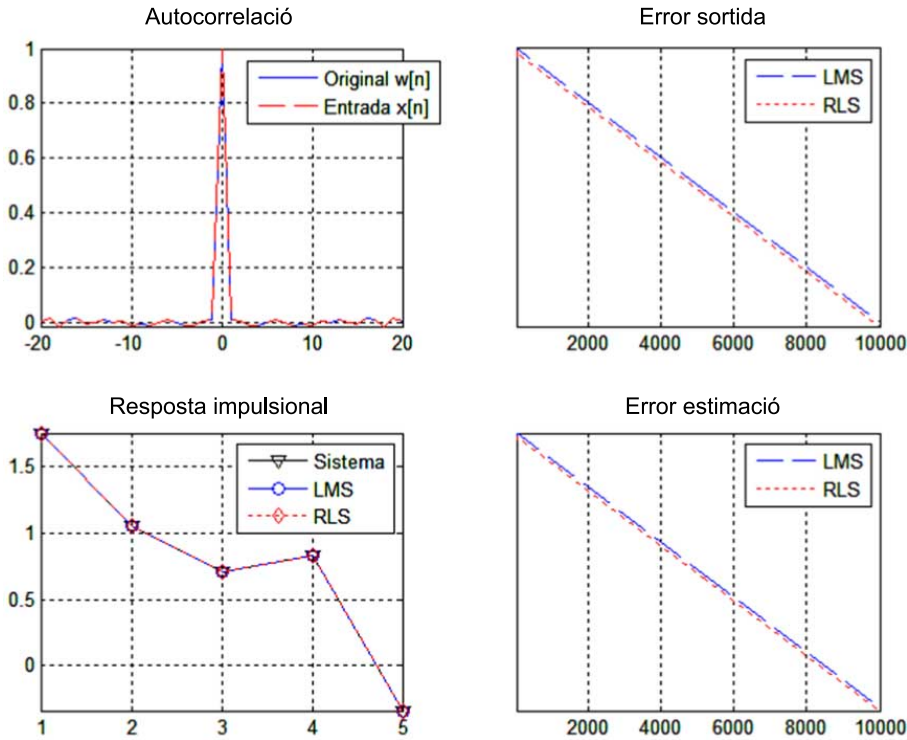


Figura 17. Resultats de la simulació amb un sistema invariant i entrada correlacionada

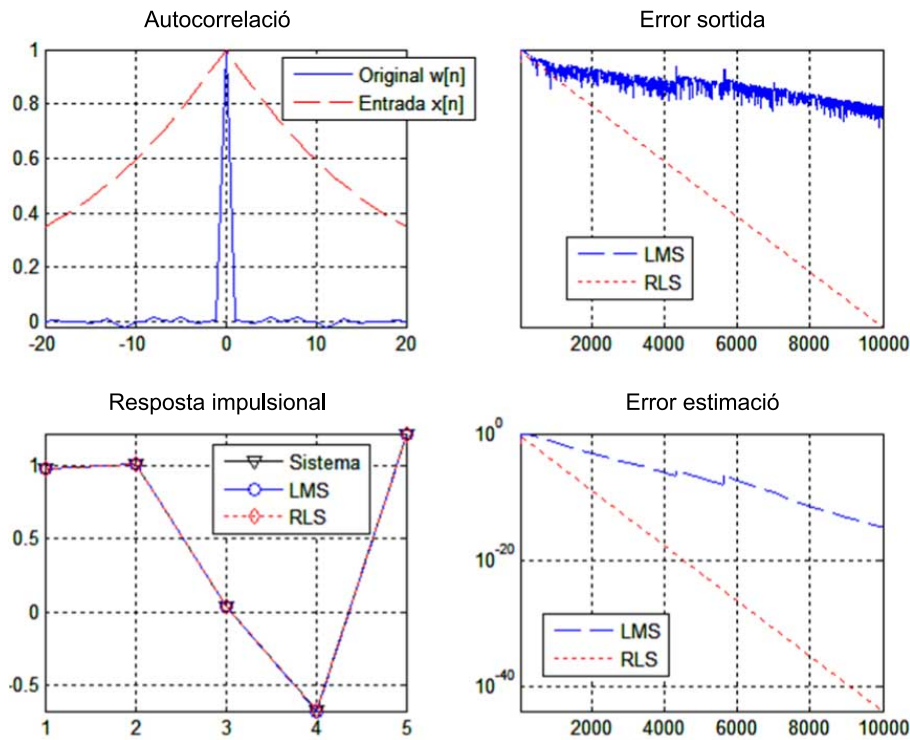


Figura 18. Resultats de la simulació amb un sistema variant i entrada incorrelacionada

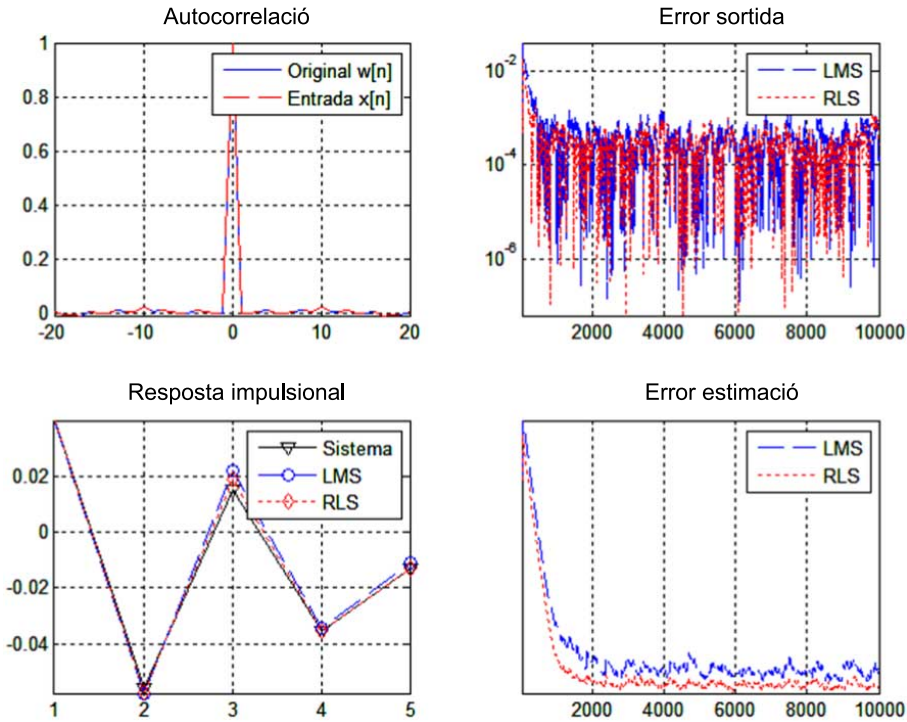


Figura 19. Resultats de la simulació amb un sistema variant i entrada correlacionada

