



7 Algoritmo Least Mean Square

- Se discutirá la teoría relacionada con el algoritmo denominado **least-mean-square** (LMS) por sus autores, Widrow y Hopf (1960).
- Este algoritmo es un miembro importante de la familia de **algoritmos de gradiente estocástico**.
- El término gradiente estocástico distingue al algoritmo LMS del método de Steepest Descent que utiliza un gradiente determinístico el el cálculo recursivo del filtro de Wiener para entradas estocásticas.
- Una característica significativa del algoritmo LMS es su **simplicidad**. No requiere mediciones de las funciones de correlación asociadas ni requiere inversión de una matriz.
- Además, su simplicidad lo ha colocado como un estándar frente a otros algoritmos de filtrado adaptivo.
- Se discutirán aspectos relacionados con el análisis de convergencia del algoritmo LMS, comenzando el estudio con una revisión de su estructura y operación.

7.1 Preliminares

- El algoritmo LMS es un algoritmo de filtrado adaptivo lineal que está formado por dos procesos básicos:
 1. **Un proceso de filtrado:** que involucra (a) calcular la salida del filtro transversal producida por la entrada y (b) generación de un error de estimación comparando la salida con una respuesta deseada.
 2. **Un proceso adaptivo:** que involucra el ajuste automático de los coeficientes del filtro de acuerdo con el error de estimación.
- La combinación de estos dos procesos trabajando juntos constituye un **lazo de realimentación** alrededor del algoritmo LMS, como ilustrado en el diagrama de bloques de la figura 27(a). Se tiene, primero, un filtro transversal alrededor del cual se construye el algoritmo LMS. Segundo, se dispone de un mecanismo para efectuar el proceso de control adaptivo sobre los coeficientes del filtro.
- Algunos detalles del filtro transversal se muestran en la figura 27(b). Las muestras de la entrada $u(n)$, $u(n - 1)$, \dots , $u(n - M + 1)$ forman los elementos del vector de entradas $\mathbf{u}(n)$ ($M \times 1$), que genera espacio multidimensional denotado por \mathcal{U}_n . En forma similar, los coeficientes $\hat{w}_0(n)$, $\hat{w}_1(n)$, \dots , $\hat{w}_{M-1}(n)$ forman el vector de coeficientes $\hat{\mathbf{w}}(n)$. El valor calculado para $\hat{\mathbf{w}}(n)$ usando el algoritmo LMS representa una estimación cuyo valor esperado se aproxima a la solución de Wiener \mathbf{w}_o (para un contexto estacionario en sentido amplio) cuando el número de iteraciones n tiende a infinito.

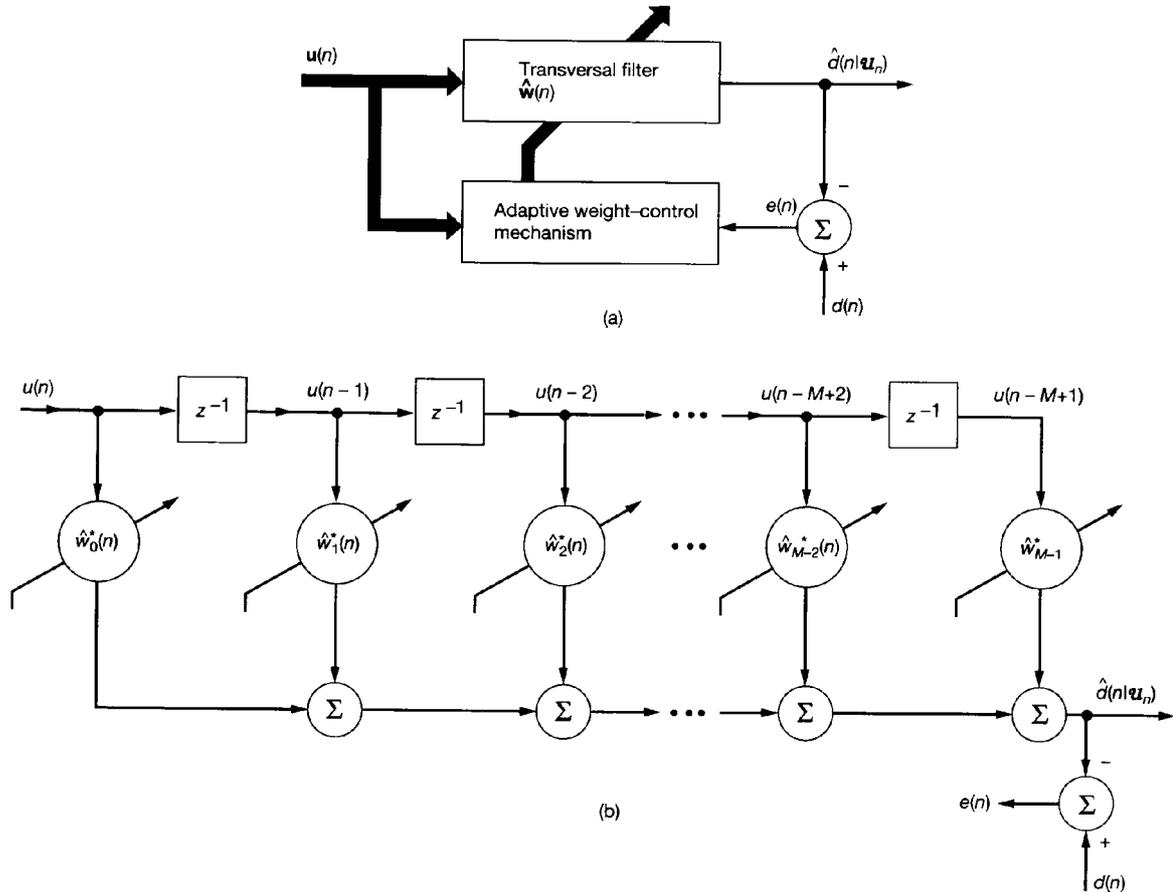


Figura 27: a) diagrama en bloques del filtro adaptivo transversal, b) detalle del filtro adaptivo c) detalle del mecanismo de control de los coeficientes.

- Durante el proceso de filtrado la respuesta deseada $d(n)$ se obtiene a partir de $\mathbf{u}(n)$. Dada esta entrada el filtro transversal produce una salida $\hat{d}(n|\mathbf{u}_n)$ usada como una estimación de la respuesta deseada $d(n)$. De esta forma es posible definir el error de estimación $e(n)$, como la diferencia entre la respuesta deseada y la salida del filtro, como es indicado en la figura 27(b). El error de estimación $e(n)$ y $\mathbf{u}(n)$ se aplican al mecanismo de control y de esta forma se cierra el lazo de realimentación alrededor de los coeficientes.

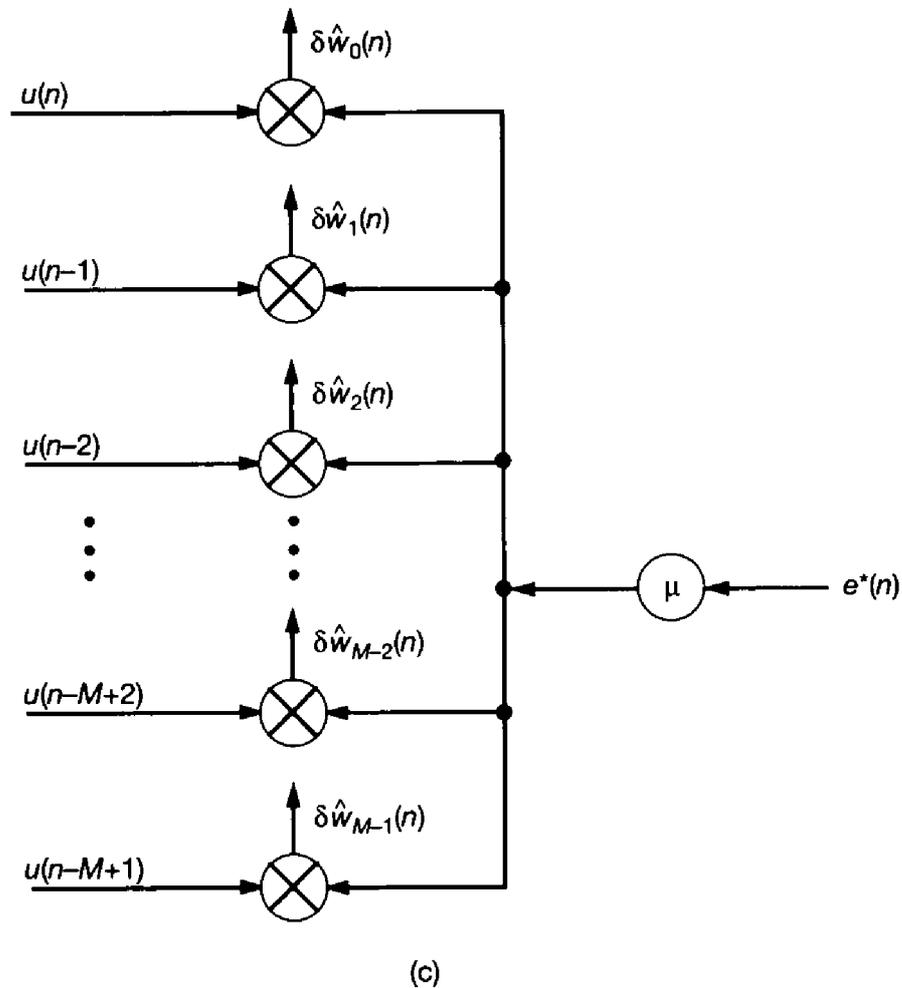


Figura 28: a) diagrama en bloques del filtro adaptivo transversal, b) detalle del filtro adaptivo c) detalle del mecanismo de control de los coeficientes.

- La figura 28(c) ilustra detalles del mecanismo de control de los coeficientes. Específicamente, una versión escalar del producto interno del error de estimación $e(n)$ y las entradas $u(n - k)$, para $k = 0, 1, 2, \dots, M - 2, M - 1$. El resultado obtenido define la corrección $\delta \hat{w}_k(n)$ aplicado al coeficiente $\hat{w}_k(n)$ en la iteración $n + 1$. El factor de escala utilizado en este cálculo es denotado por μ , y se denomina frecuentemente **factor de convergencia**.



- Comparando el mecanismo de control de la figura 28(c) para el algoritmo LMS con el del método Steepest Descent es posible concluir que el primero utiliza el producto $u(n-k)e^*(k)$ como una estimación del elemento k del vector gradiente $\nabla J(n)$ que caracteriza al método Steepest Descent. En consecuencia, el cálculo recursivo de cada coeficiente por medio del algoritmo LMS sufre de un **ruido de gradiente**.
- En este estudio se supone que $\mathbf{u}(n)$ y $d(n)$ se obtienen de un contexto estacionario en sentido amplio conjuntamente. Para ese contexto, sabemos que el método Steepest Descent calcula un vector de coeficientes $\mathbf{w}(n)$ que se mueve en la superficie de desempeño sobre una trayectoria determinística que termina en la solución de Wiener \mathbf{w}_o . El algoritmo LMS, por otro lado, se comporta en forma diferente por la presencia del ruido de gradiente. En lugar de terminar en la solución de Wiener, $\hat{\mathbf{w}}(n)$ ejecuta un movimiento aleatorio alrededor del punto de mínimo de la superficie de desempeño.
- Anteriormente señalamos que el algoritmo LMS involucra una operación de realimentación, la cual introduce el aspecto relacionado de **estabilidad**. En este contexto, un criterio significativo es requerir que

$$J(n) \rightarrow J(\infty), \quad \text{cuando } n \rightarrow \infty$$

donde $J(n)$ es el error medio cuadrático producido por el algoritmo LMS en el instante n , y su valor final $J(\infty)$ es una constante. Un algoritmo que satisface este requerimiento se dice **convergente en sentido medio cuadrático**. Para que el algoritmo LMS satisfaga este criterio, el factor de convergencia μ

debe satisfacer ciertas condiciones relacionadas con la estructura de autovalores de la matriz correlación de la entrada.

- La diferencia entre el valor final $J(\infty)$ y el valor mínimo J_{min} alcanzado por la solución de Wiener se denomina **exceso de error medio cuadrático** $J_{ex}(\infty)$. Esta diferencia representa el precio pagado por usar un mecanismo adaptivo para controlar los coeficientes del algoritmo LMS en lugar de un mecanismo determinístico. El cociente entre $J_{ex}(\infty)$ y J_{min} se denomina **desajuste** \mathcal{M} , que representa una medida de cuán lejos la solución de estado estacionario obtenida con el algoritmo LMS está de la solución de Wiener. Es importante tener en cuenta, sin embargo, que el desajuste \mathcal{M} puede controlarse. En particular, el lazo de realimentación que actúa sobre los coeficientes se comporta como un **filtro pasabajos**, cuya constante de tiempo promedio es inversamente al **factor de convergencia** μ . De aquí que, **asignando un valor pequeño a μ , se logra que el proceso adaptivo progrese lentamente y los efectos del ruido de gradiente sobre los coeficientes se suavizan. Esto a su vez conduce a una reducción del desajuste.**

7.2 Ajuste de los coeficientes

- Las mediciones exactas del vector gradiente no están en general disponibles y debe estimarse.
- Para una estimación de $\nabla J(n)$, la estrategia obvia es sustituir estimaciones de \mathbf{R} y \mathbf{p} en $\nabla J(n) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n)$.
- La **elección más simple** de los estimadores para \mathbf{R} y \mathbf{p} es utilizar **valores instantáneos**, dados por

$$\hat{\mathbf{R}}(n) = \mathbf{u}(n)\mathbf{u}^H(n) \quad \hat{\mathbf{p}}(n) = \mathbf{u}(n)d^*(n)$$

- En consecuencia, la estimación instantánea del vector gradiente será

$$\hat{\nabla} J(n) = -2\mathbf{u}(n)d^*(n) + 2\mathbf{u}(n)\mathbf{u}^H(n)\hat{\mathbf{w}}(n)$$

- Sustituyendo esta ecuación en el algoritmo Steepest Descent se obtiene

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)[d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{w}}(n)] \quad (75)$$

lo que puede escribirse en base a tres relaciones básicas:

1. **Salida del filtro:** $y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n)$.
2. **Error de estimación:** $e(n) = d(n) - y(n)$.
3. **Adaptación de los coeficientes:** $\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)e^*(n)$.

- El algoritmo LMS requiere $2M + 1$ multiplicaciones complejas y $2M$ sumas complejas por iteración, donde M es el número de coeficientes del filtro adaptivo. En otras palabras, la complejidad computacional del algoritmo LMS es $O(M)$.

7.3 Ejemplos

Ejemplo 1: Modelo canónico del LMS complejo

- El algoritmo LMS descrito anteriormente es **complejo** en el sentido que la entrada y la salida así como los coeficientes son complejos. Definimos

$$\text{Vector de entrada } \mathbf{u}(n) = \mathbf{u}_I(n) + j\mathbf{u}_Q(n)$$

$$\text{Respuesta deseada } d(n) = d_I(n) + jd_Q(n)$$

$$\text{Vector de coeficientes } \hat{\mathbf{w}}(n) = \hat{\mathbf{w}}_I(n) + j\hat{\mathbf{w}}_Q(n)$$

$$\text{Salida del filtro } y(n) = y_I(n) + jy_Q(n)$$

$$\text{Error de estimación } e(n) = e_I(n) + je_Q(n)$$

- Usando las definiciones del algoritmo se tiene que

$$y_I(n) = \hat{\mathbf{w}}_I^T(n)\mathbf{u}_I(n) - \hat{\mathbf{w}}_Q^T(n)\mathbf{u}_Q(n)$$

$$y_Q(n) = \hat{\mathbf{w}}_I^T(n)\mathbf{u}_Q(n) - \hat{\mathbf{w}}_Q^T(n)\mathbf{u}_I(n)$$

$$e_I(n) = d_I(n) - y_I(n)$$

$$e_Q(n) = d_Q(n) - y_Q(n)$$

$$\hat{\mathbf{w}}_I(n+1) = \hat{\mathbf{w}}_I(n) + \mu[e_I(n)\mathbf{u}_I(n) - e_Q(n)\mathbf{u}_Q(n)]$$

$$\hat{\mathbf{w}}_Q(n+1) = \hat{\mathbf{w}}_Q(n) + \mu[e_I(n)\mathbf{u}_Q(n) + e_Q(n)\mathbf{u}_I(n)]$$

- Este modelo del LMS complejo es equivalente a un conjunto de cuatro LMS reales con **acoplamiento cruzado** entre ellos.

Ejemplo 2: Deconvolución adaptativa para el procesamiento de datos sísmicos

- La idea de **deconvolución predictiva** es útil para el procesamiento de sismogramas de reflexión. Discutiremos la utilización del algoritmo LMS para la implementación adaptativa de esa idea. Se supondrá que los datos sísmicos son estacionarios durante la ventana utilizada para generar la operación de deconvolución.
- El conjunto de datos reales procesados por el sismograma es $u(n)$, $n = 1, 2, \dots, N$, donde N es la longitud de la muestra. El método de deconvolución adaptativa con base en el algoritmo LMS es el siguiente
 - Un operador M -dimensional $\hat{\mathbf{w}}(n)$ se utiliza para generar una predicción de la traza a partir de los datos:

$$\hat{\mathbf{u}}(n + \Delta) = \hat{\mathbf{w}}^T(n)\mathbf{u}(n)$$

donde $\hat{\mathbf{w}}(n) = [\hat{w}_0(n), \hat{w}_1(n), \dots, \hat{w}_{M-1}(n)]^T$, $\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-M+1)]^T$ y $\Delta \geq 1$ es el **perfil de predicción**.

- La traza deconvolucionada $y(n)$ se define como la diferencia entre la entrada y la traza predicha,

$$y(n) = u(n) - \hat{u}(n)$$

- Se adapta el operador $\hat{\mathbf{w}}(n)$,

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu[u(n+\Delta) - \hat{u}(n+\Delta)]\mathbf{u}(n)$$

Ejemplo 3: Medición de frecuencia instantánea

- Estudiaremos el uso del algoritmo LMS como base para estimar el contenido de frecuencia de una señal **de banda estrecha**.
- Con ese objetivo se discutirá la relación entre tres ideas básicas: un modelo AR para describir un proceso estocástico, un predictor para analizar el proceso y el algoritmo LMS para estimar los parámetros AR.

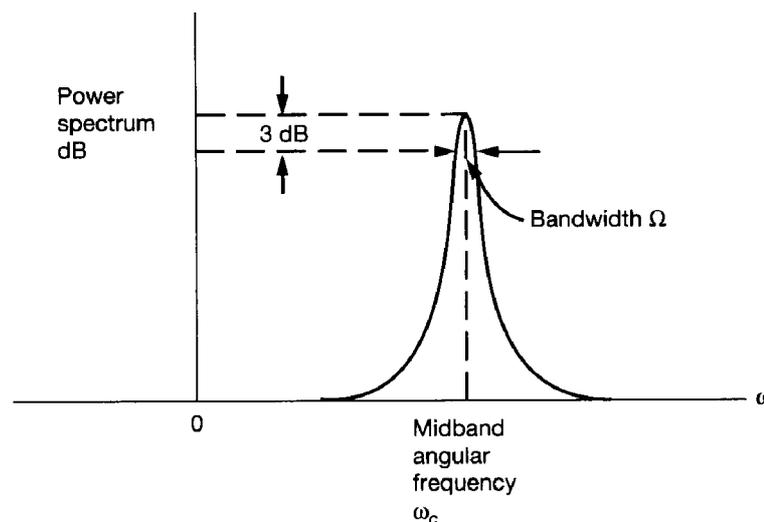


Figura 29: Definición de una señal de banda estrecha en función de su espectro.

- Definimos una señal de banda estrecha como aquella cuyo ancho de banda Ω es pequeño comparado con la frecuencia central ω_c . Una señal modulada en frecuencia (FM) es un ejemplo de señal de banda estrecha. La **frecuencia instantánea** (definida como la derivada de la fase con respecto al tiempo) de una señal FM varía linealmente con la señal moduladora.
- Considerando un proceso de banda estrecha $u(n)$ generado por un modelo AR **variante en el tiempo** de orden M , dado por

$$u(n) = - \sum_{k=1}^M a_k(n)u(n-k) + \nu(n)$$

donde $a_k(n)$ son los parámetros variantes en el tiempo del modelo y $\nu(n)$ es un proceso de ruido blanco con media cero y varianza $\sigma_\nu^2(n)$. El espectro de potencia AR (variante en el tiempo) del proceso $u(n)$ está dado por

$$S_{AR}(\omega; n) = \frac{\sigma_\nu^2(n)}{|1 + \sum_{k=1}^M a_k(n)e^{-j\omega k}|^2}, \quad -\pi < \omega \leq \pi$$

- Notar que un proceso AR cuyos polos está próximos al círculo unitario del plano z tienen la característica de un proceso de banda estrecha.
- Para estimar los parámetros del modelo se utiliza un filtro transversal como un predictor lineal de orden M , con coeficientes $\hat{w}_k(n)$, $k = 1, 2, \dots, M$. Los coeficientes se adaptan continuamente con la señal de entrada $u(n)$. En particular, utilizamos el algoritmo LMS como mostrado por

$$\hat{w}_k(n+1) = \hat{w}_k(n) + \mu u(n-k)f_M(n), \quad k = 1, 2, \dots, M$$

donde $f_M(n)$ es el **error de predicción** dado por

$$f_M(n) = u(n) - \sum_{k=1}^M \hat{w}_k(n)u(n-k)$$

- Los coeficientes del predictor adaptivo están relacionados con los parámetros del modelo AR por

$$-\hat{w}_k(n) = \text{estimación de } a_k(n) \text{ en el instante } n, \quad \text{para } k = 1, 2, \dots,$$

- Además, la potencia promedio del error de predicción $f_M(n)$ provee una estimación de la varianza del ruido $\sigma_\nu^2(n)$.
- El objetivo es localizar la frecuencia de una señal de banda estrecha. En consecuencia, en lo siguiente se ignorará la estimación de $\sigma_\nu^2(n)$. Específicamente, utilizaremos los coeficientes del predictor adaptivo para definir un **función frecuencia variante en el tiempo** definida por

$$F(\omega; n) = \frac{1}{|1 + \sum_{k=1}^M a_k(n)e^{-j\omega k}|^2}$$

- La diferencia esencial entre $F(\omega; n)$ y $S_{AR}(\omega; n)$ reside en sus factores de escala del numerador.
- Es posible utilizar $F(\omega; n)$ para medir la frecuencia instantánea de una señal de FM $u(n)$, previendo que las siguientes suposiciones sean válidas
 - El predictor adaptivo ha estado en operación por tiempo suficiente para asegurar que cualquier transitorio causado por la inicialización de los coeficientes ha finalizado.
 - El factor de convergencia μ se eligió correctamente para que el predictor adaptivo funcione adecuadamente, o sea, el error de predicción $f_M(n)$ es pequeño para todo n .
 - La señal moduladora es esencialmente constante sobre el rango de muestras del predictor adaptivo, el cual se extiende desde el instante $(N - M)$ hasta $(n - 1)$.

Ejemplo 5: Detección de sinusoidales en ruido

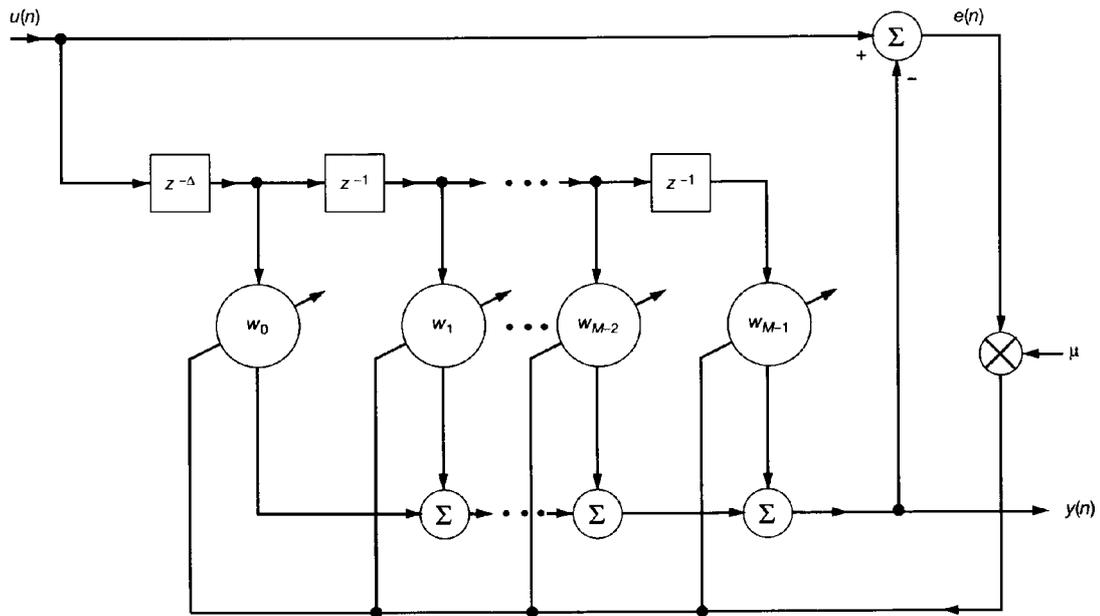


Figura 30: Mejorador de línea espectral adaptivo.

- El **mejorador de línea espectral adaptivo** (adaptive line enhancer, ALE), es un dispositivo utilizado para detectar una señal periódica en ruido de banda ancha. De esta figura es posible concluir que el ALE es un cancelador de ruido adaptivo en el que su señal de referencia, en lugar de obtenerse separadamente, consiste en muestras retrasadas de la señal de entrada primaria.
- El retardo Δ se denomina **paso de predicción** o **retardo de decorrelación**, y se mide en unidades del período de muestreo.
- Consideraremos que

$$u(n) = A \sin(\omega_0 n + \phi_0) + \nu(n)$$

donde ϕ_0 es un desplazamiento de fase arbitrario y el ruido de medición $\nu(n)$ se supone de media cero y varianza σ_ν^2 . El ALE actúa como un detector en base a los siguientes acciones:

- El paso de predicción Δ tiene un valor suficientemente grande para **remover** la correlación entre el ruido $\nu(n)$ en la entrada y el ruido $\nu(n - \Delta)$ en la referencia, mientras se introduce un desplazamiento de fase igual a $\omega_0\Delta$ entre componentes sinusoidales de esas dos entradas.
- Los coeficientes del filtro transversal se ajustan con el algoritmo LMS para minimizar el valor medio cuadrático de la señal de error y en consecuencia compensar el desplazamiento de fase $\omega_0\Delta$.
- El resultado neto de esas acciones es producir una salida $y(n)$ formada por una sinusoidal con ruido de media cero. En particular, cuando ω_0 es varios múltiplos mayor que π/M de cero ó π , es posible mostrar que

$$y(n) = aA \sin(\omega_0 n + \phi) + \nu_{out}(n)$$

donde ϕ es un desplazamiento de fase, y ν_{out} es el ruido de salida. El factor de escala a está definido por

$$a = \frac{(M/2)SNR}{1 + (M/2)SNR}$$

donde M es la longitud del filtro transversal y SNR es la relación señal a ruido a la entrada del ALE: $SNR = A^2/(2\sigma_\nu^2)$. De acuerdo a la ecuación de la salida del ALE, este actúa como un **filtro autosintonizado** cuya respuesta en frecuencia exhibe un pico en ω_0 .

- Es posible mostrar también que la densidad espectral de potencia de la salida $y(n)$ del ALE puede expresarse como

$$S(\omega) = \frac{\pi A^2}{2} (a^2 + \mu \sigma_v^2 M) [\delta(\omega - \omega_0) + \delta(\omega + \omega_0)] + \mu \sigma_v^4 M + \frac{a^2 \sigma_v^2}{M^2} \left[\frac{1 - \cos M(\omega - \omega_0)}{1 - \cos(\omega - \omega_0)} + \frac{1 + \cos M(\omega - \omega_0)}{1 + \cos(\omega - \omega_0)} \right], \quad -7$$

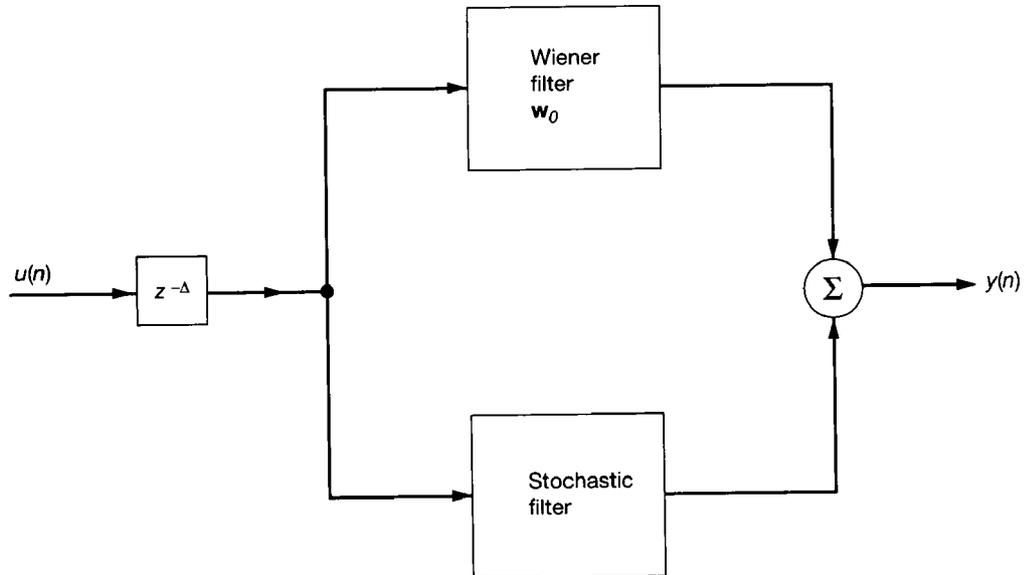


Figura 31: Modelo del mejorador de línea espectral adaptivo.

- En un ambiente ESA, el valor medio de $\hat{\mathbf{w}}(n)$ obtenido por el algoritmo LMS converge a la solución de Wiener \mathbf{w}_o . El modelo de estado estacionario, representado en la figura 31, está compuesto por la solución de Wiener \mathbf{w}_o actuando en paralelo con una componente lentamente variable, de media cero, $\epsilon(n)$ debida al ruido de gradiente.
- Teniendo en cuenta que la entrada al ALE tiene dos componentes (sinusoidal y ruido), es posible distinguir cuatro componentes del espectro de potencia:
 - Una componente sinusoidal de frecuencia ω_0 y potencia promedio $\pi a^2 A^2 / 2$, como resultado del procesamiento de la sinu-

soide de entrada por el filtro de Wiener representado por w_o .

- Una componente sinusoidal de frecuencia ω_0 y potencia promedio $\pi \mu A^2 \sigma_\nu^2 M / 2$, debida al filtro estocástico representado por $\epsilon(n)$ actuando sobre la entrada sinusoidal.
 - Una componente de ruido de banda ancha de varianza $\mu \sigma_\nu^4 M$, debida a la acción del filtro estocástico sobre el ruido $\nu(n)$.
 - Una componente de ruido de banda estrecha filtrado centrado en ω_0 , debida al procesamiento del ruido $\nu(n)$ por el filtro de Wiener.
- De esta forma el espectro de potencia en la salida del ALE consiste de una sinusoidal centrada sobre la base de ruido de banda estrecha filtrado, la combinación del cual forma parte del ruido de banda ancha.
 - Más importante, cuando existe una SNR adecuada en la entrada, la salida del ALE es, en promedio, aproximadamente igual a la componente sinusoidal presente en la entrada, determinando en consecuencia un dispositivo simple para la detección.

7.4 Estabilidad y desempeño del algoritmo LMS

- En el estudio de las propiedades conviene trabajar con el vector de error en los coeficientes antes que con el vector de coeficientes

$$\boldsymbol{\epsilon}(n) = \hat{\boldsymbol{w}}(n) - \boldsymbol{w}_o$$

En base al algoritmo LMS es posible obtener

$$\boldsymbol{\epsilon}(n+1) = [\boldsymbol{I} - \mu \boldsymbol{u}(n) \boldsymbol{u}^H(n)] \boldsymbol{\epsilon}(n) + \mu \boldsymbol{u}(n) e_o^*(n) \quad (76)$$

donde $e_o(n)$ es el **error de estimación producido con la solución óptima de Wiener**, dado por

$$e_o(n) = d(n) - \boldsymbol{w}_o^H \boldsymbol{u}(n)$$

Método de promediación directa

- La ecuación 76 es una **ecuación a diferencias estocástica** en el vector de error de los coeficientes $\boldsymbol{\epsilon}$ con la siguiente característica
 - Una matriz de sistema $[\boldsymbol{I} - \mu \boldsymbol{u}(n) \boldsymbol{u}^H(n)]$, aproximadamente igual a \boldsymbol{I} para todo n , previendo que μ sea suficientemente pequeño.
- De acuerdo con este método, la solución a esa ecuación a diferencias estocástica, operando bajo la suposición de un μ pequeño, es próxima a la solución de otra ecuación a diferencias estocástica cuya matriz de sistema es

$$E[\mathbf{I} - \mu \mathbf{u}(n) \mathbf{u}^H(n)] = \mathbf{I} - \mu \mathbf{R}$$

Específicamente es posible reemplazar (76) con

$$\boldsymbol{\epsilon}(n+1) = (\mathbf{I} - \mu \mathbf{R}) \boldsymbol{\epsilon}(n) + \mu \mathbf{u}(n) e_o^*(n) \quad (77)$$

- Este método es una aproximación heurística razonable, dado que está basado en la idea que, para μ pequeño, la aleatoriedad de $\boldsymbol{\epsilon}(n)$ tiende a desaparecer.

Teoría de Independencia

- Es posible realizar el análisis estadístico del algoritmo LMS bajo las **suposiciones de independencia** consistentes en los siguientes cuatro puntos
 1. Los vectores $\mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(n)$ constituyen una secuencia de vectores estadísticamente independientes.
 2. En el instante n , $\mathbf{u}(n)$ es estadísticamente independiente de las muestras pasadas de la respuesta deseada $d(1), d(2), \dots, d(n-1)$.
 3. En el instante n , la respuesta deseada $d(n)$ depende de $\mathbf{u}(n)$, pero es estadísticamente independiente de las muestras previas de la respuesta deseada.
 4. $\mathbf{u}(n)$ y la respuesta deseada $d(n)$ son variables aleatorias **con distribución Gaussiana conjuntamente**, para todo n .
- El análisis estadístico del algoritmo LMS basado en estas suposiciones se denomina **teoría de la independencia**.

- De la ecuación (75) observamos que $\hat{\mathbf{w}}(n+1)$ en el instante $(n+1)$ depende solamente de las siguientes entradas
 1. Los vectores de entrada de muestras pasadas: $\mathbf{u}(n)$, $\mathbf{u}(n-1)$, \dots , $\mathbf{u}(1)$.
 2. La respuesta deseada de muestras pasadas: $d(n)$, $d(n-1)$, \dots , $d(1)$.
 3. El valor inicial del vector de coeficientes $\hat{\mathbf{w}}(0)$.
- En base a los puntos 1 y 2 de las suposiciones de independencia, hallamos que $\hat{\mathbf{w}}(n+1)$, y en consecuencia $\epsilon(n+1)$, es independiente de $\mathbf{u}(n+1)$ y $d(n+1)$. Esta observación es muy útil y será utilizada repetidamente en lo subsiguiente.
- La suposición de independencia puede justificarse fácilmente en ciertas aplicaciones tal como en beamforming adaptivo, donde es posible que los vectores de entrada sucesivos recibidos por el arreglo de sensores sean independientes unos de otros.
- Sin embargo, en aplicaciones de comunicaciones (predicción, ecualización y cancelamiento de eco), la secuencia de vectores de entrada que dirige al vector de coeficientes hacia la solución óptimo da Wiener no es estadísticamente independiente. Esta dependencia aparece debido a la **propiedad de desplazamiento** de los datos de entrada. Específicamente, el vector de entrada en el instante n es

$$\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-M+1)]^T$$

- En el instante $n+1$ tome el nuevo valor

$$\mathbf{u}(n+1) = [u(n+1), u(n), \dots, u(n-M)]^T$$

- De esta forma, con la llegada de cada nueva muestra $u(n + 1)$, la muestra más antigua $u(n - M + 1)$ se descarta de $\mathbf{u}(n)$, y las muestras remanentes $u(n)$, $u(n - 1)$, \dots , $u(n - M + 2)$ son desplazadas hacia atrás en el tiempo en una unidad para hacer lugar a la nueva muestra. .
- La teoría de la independencia **ignora** esta dependencia estadística entre vectores de entrada sucesivos en ciertos puntos del desarrollo. Por ejemplo, para evaluar la esperanza del término $\mathbf{u}(n)\mathbf{u}^H(n)\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)$ se supone que el vector de error en los coeficientes y $\mathbf{u}(n)$ son estadísticamente independientes y de esta forma

$$E[\mathbf{u}(n)\mathbf{u}^H(n)\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)] = E[\mathbf{u}(n)\mathbf{u}^H(n)] E[\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)]$$

- Como contraste, cuando es necesario evaluar $E[\mathbf{u}(n)\mathbf{u}^H(n)]$, la estructura de correlación en $\mathbf{u}(n)$ se preserva, así como la de $E[\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)]$.

Critério de convergencia

- En base a (77) es posible establecer las condiciones necesarias para la **convergencia de la media**, o sea:

$$E[\boldsymbol{\epsilon}(n)] \rightarrow \mathbf{0}, \quad \text{para } n \rightarrow \infty$$

o equivalentemente

$$E[\hat{\mathbf{w}}(n)] \rightarrow \mathbf{w}_o, \quad \text{para } n \rightarrow \infty$$

- Ese criterio de convergencia tiene poco valor práctico, dado que una secuencia de variables aleatorias de media cero arbitraria converge en este sentido.
- Otro criterio más fuerte es la **convergencia en media**, descrita por

$$E [||\epsilon(n)||] \rightarrow 0, \quad \text{para } n \rightarrow \infty$$

Sin embargo, una prueba de convergencia en media es bastante tediosa, básicamente porque $||\epsilon(n)||$ es singular en el origen.

- Para evitar estos problemas, es posible considerar la **convergencia en media cuadrática**. Específicamente, decimos que el algoritmo LMS es convergente en media cuadrática si

$$\mathcal{D}(n) = E [||\epsilon(n)||] \rightarrow \text{constante}, \quad \text{para } n \rightarrow \infty$$

donde la cantidad escalar $\mathcal{D}(n)$ se denomina **desviación del error cuadrático**.

- Otro camino para describir la convergencia del algoritmo LMS en media cuadrática es requerir que

$$J(n) = E [|e(n)|^2] \rightarrow \text{constante}, \quad \text{para } n \rightarrow \infty \quad (78)$$

donde $e(n)$ es el error de estimación y $J(n)$ es el error medio cuadrático.

- Más adelante será posible mostrar que

$$\lambda_{min} \mathcal{D}(n) \leq J_{ex}(n) \leq \lambda_{max} \mathcal{D}(n), \quad \text{para todo } n$$

donde $J_{ex}(n)$ es el exceso de error medio cuadrático, o sea, la diferencia entre $J(n)$ y el error medio cuadrático mínimo J_{min} producido por el filtro óptimo de Wiener. A la luz de esta desigualdad es posible establecer que la disminución de $J_{ex}(n)$ y $\mathcal{D}(n)$ cuando aumenta n es matemáticamente equivalente.

- En consecuencia, es suficiente en nuestro análisis colocar nuestra atención en el criterio de convergencia descrito por (78).
- Con este criterio de convergencia para el análisis del algoritmo LMS procederemos como sigue
 1. Utilizaremos la ecuación a diferencias estocástica (77) para obtener una ecuación recursiva para calcular la matriz correlación de $\epsilon(n)$.
 2. Luego se obtendrá una expresión para el exceso de error medio cuadrático $J_{ex}(n)$.

Matriz correlación del error en los coeficientes

- Por definición, la matriz correlación del vector de error en los coeficientes es

$$\mathbf{K}(n) = E[\epsilon(n)\epsilon^H(n)]$$

de forma que, aplicando esta definición a la ecuación (77) e invocando luego la suposición de independencia tenemos

$$\mathbf{K}(n+1) = (\mathbf{I} - \mu\mathbf{R})\mathbf{K}(n)(\mathbf{I} - \mu\mathbf{R}) + \mu^2 J_{min}\mathbf{R} \quad (79)$$

- Esta ecuación puede justificarse como sigue

- El término $(\mathbf{I} - \mu \mathbf{R}) \mathbf{K}(n)(\mathbf{I} - \mu \mathbf{R})$ es el resultado de evaluar la esperanza de $(\mathbf{I} - \mu \mathbf{R}) \boldsymbol{\epsilon}(n)$ consigo mismo.
- La esperanza del producto cruzado $\mu e_o(n)(\mathbf{I} - \mu \mathbf{R}) \boldsymbol{\epsilon}(n) \mathbf{u}^H(n)$ es cero en virtud de la independencia entre $\boldsymbol{\epsilon}(n)$ y $\mathbf{u}(n)$.
- El término $\mu^2 J_{min} \mathbf{R}$ se obtiene aplicando el teorema de factorización Gaussiana al producto $\mu^2 e_o^*(n) \mathbf{u}(n) \mathbf{U}^H(n) e_o(n)$.
- El término $\mu^2 J_{min} \mathbf{R}$ evita que $\mathbf{K}(n) = \mathbf{0}$ sea una solución de esta ecuación. En particular, $\boldsymbol{\epsilon}(n)$ solo se aproxima a cero, pero ejecuta pequeñas fluctuaciones alrededor de cero. Esta formalidad confirma lo ilustrado en el ejemplo 5.
- Dado que \mathbf{R} es positiva definida y μ es pequeño es posible concluir que el primer término de (79) es positivo definido siempre que $\mathbf{K}(n)$ sea positiva definida. Claramente el último término de esta expresión es siempre positivo definido. En consecuencia $\mathbf{K}(n+1)$ es positiva definida siempre que $\mathbf{K}(n)$ lo sea. La prueba por inducción se completa notando que $\mathbf{k}(0)$ es positiva definida, ya que

$$\mathbf{K}(0) = \boldsymbol{\epsilon}(0) \boldsymbol{\epsilon}^H(0) = [\hat{\mathbf{w}}(0) - \mathbf{w}_o][\hat{\mathbf{w}}^H(0) - \mathbf{w}_o^H]$$

Exceso de error medio cuadrático

- Para determinar el comportamiento en el transitorio del error medio cuadrático del algoritmo LMS, procederemos como sigue. El error producido por el algoritmo LMS es

$$\begin{aligned} e(n) &= d(n) - \hat{\mathbf{w}}^H(n) \mathbf{u}(n) \\ &= d(n) - \mathbf{w}_o^H \mathbf{u}(n) - \boldsymbol{\epsilon}^H(n) \mathbf{u}(n) \\ &= e_o(n) - \boldsymbol{\epsilon}^H(n) \mathbf{u}(n) \end{aligned} \quad (80)$$

- Luego, usando (80) para obtener $J(n)$ e invocando la suposición de independencia se obtiene

$$\begin{aligned}
 J(n) &= E[|e(n)|^2] \\
 &= E[(e_o(n) - \boldsymbol{\epsilon}^H(n)\mathbf{u}(n))(e_o^*(n) - \mathbf{U}^H(n)\boldsymbol{\epsilon}(n))] \\
 &= J_{min} + E[\boldsymbol{\epsilon}^H(n)\mathbf{u}(n)\mathbf{u}^H(n)\boldsymbol{\epsilon}(n)] \quad (81)
 \end{aligned}$$

- La esperanza de esta última ecuación está asociada con una variable aleatoria escalar representada por un producto de vectores triple, y como la traza de un escalar es el escalar mismo se tiene que

$$\begin{aligned}
 E[\boldsymbol{\epsilon}^H(n)\mathbf{u}(n)\mathbf{u}^H(n)\boldsymbol{\epsilon}(n)] &= E[\text{tr}\{\boldsymbol{\epsilon}^H(n)\mathbf{u}(n)\mathbf{u}^H(n)\boldsymbol{\epsilon}(n)\}] \\
 &= E[\text{tr}\{\mathbf{u}(n)\mathbf{u}^H(n)\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)\}] \\
 &= \text{tr}\{E[\mathbf{u}(n)\mathbf{u}^H(n)\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)]\}
 \end{aligned}$$

- Invocando la suposición de independencia nuevamente es posible reducir esta expresión a

$$\begin{aligned}
 E[\boldsymbol{\epsilon}^H(n)\mathbf{u}(n)\mathbf{u}^H(n)\boldsymbol{\epsilon}(n)] &= \text{tr}\{E[\mathbf{u}(n)\mathbf{u}^H(n)] E[\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)]\} \\
 &= \text{tr}\{\mathbf{R}\mathbf{K}(n)\}
 \end{aligned}$$

- De esta forma:

$$J(n) = J_{min} + \text{tr}\{\mathbf{R}\mathbf{K}(n)\} \quad (82)$$

- Esta ecuación indica que para todo n , el valor medio cuadrático del error de estimación en el algoritmo LMS consiste en dos

componentes: el error medio cuadrático mínimo J_{min} y una componente dependiente del comportamiento transitorio de la matriz correlación asociada al error en los coeficientes $\mathbf{K}(n)$. Dado que esta última componente es positiva definida para todo n , **el algoritmo LMS siempre produce un error medio cuadrático $J(n)$ en exceso del error medio cuadrático mínimo.**

- Definimos formalmente el **exceso de error medio cuadrático** como

$$J_{ex}(n) = J(n) - J_{min} = tr[\mathbf{R}\mathbf{K}(n)]$$

- Para un análisis más informativo, introducimos un cambio de coordenadas sobre la matriz \mathbf{R} , tal que como usualmente $\mathbf{Q}^H \mathbf{R} \mathbf{Q} = \mathbf{\Lambda}$, tal que en particular

$$\mathbf{Q}^H \mathbf{K}(n) \mathbf{Q} = \mathbf{X}(n)$$

donde, en general $\mathbf{X}(n)$ no es una matriz triangular. Entonces

$$\begin{aligned} J_{ex}(n) &= tr[\mathbf{R}\mathbf{K}(n)] \\ &= tr[\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H\mathbf{Q}\mathbf{X}(n)\mathbf{Q}^H] \\ &= tr[\mathbf{Q}\mathbf{\Lambda}\mathbf{X}(n)\mathbf{Q}^H] \\ &= tr[\mathbf{Q}^H\mathbf{Q}\mathbf{\Lambda}\mathbf{X}(n)] \\ &= tr[\mathbf{\Lambda}\mathbf{X}(n)] \end{aligned}$$

- Dado que $\mathbf{\Lambda}$ es una matriz diagonal es posible escribir

$$J_{ex}(n) = \sum_{i=1}^M \lambda_i x_i \quad (83)$$

donde $x_i(n)$, $i = 1, 2, \dots, M$ son los elementos de la diagonal de $\mathbf{X}(n)$ y λ_i son los autovalores de \mathbf{R} .

- Usando la transformación de coordenadas anterior es posible reescribir (79) como

$$\mathbf{X}(n+1) = (\mathbf{I} - \mu\mathbf{\Lambda})\mathbf{X}(n)(\mathbf{I} - \mu\mathbf{\Lambda}) + \mu^2 J_{min}\mathbf{\Lambda} \quad (84)$$

- Notar que como $J_{ex}(n)$ depende solo de los elementos $x_i(n)$ asociados a la diagonal de $\mathbf{X}(n)$, entonces en esta última ecuación entera

$$x_i(n+1) = (1 - \mu\lambda_i)^2 x_i(n) + \mu^2 J_{min}\lambda_i, \quad i = 1, 2, \dots, M \quad (85)$$

- Definiendo $\mathbf{x}(n) = [x_1(n), x_2(n), \dots, x_M(n)]^T$ y $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_M]^T$, entonces

$$\mathbf{x}(n+1) = \mathbf{B}\mathbf{x}(n) + \mu^2 J_{min}\boldsymbol{\lambda} \quad (86)$$

donde \mathbf{B} es una matriz ($M \times M$) con elementos dados por

$$b_{ij} = \begin{cases} (1 - \mu\lambda_i)^2, & i = j \\ \mu^2\lambda_i\lambda_j, & i \neq j \end{cases}$$

de forma que es relativamente simple verificar que \mathbf{B} es real, positiva y simétrica.

- Puede mostrarse que la solución de la ecuación a diferencias (86) está dada por

$$\mathbf{x}(n) = \sum_{i=1}^M c_i^n \mathbf{g}_i \mathbf{g}_i^T [\mathbf{x}(0) - \mathbf{x}(\infty)] + \mathbf{x}(\infty)$$

donde

- El coeficiente c_i es el i -ésimo autovalor de la matriz \mathbf{B} y \mathbf{g}_i es su autovector asociado, o sea: $\mathbf{G}^T \mathbf{B} \mathbf{G} = \mathbf{C}$, donde $\mathbf{C} = \text{diag}[c_1, c_2, \dots, c_M]$ y $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_M]$.
 - El vector $\mathbf{x}(0)$ es el valor inicial de $\mathbf{x}(n)$ y $\mathbf{x}(\infty)$ su valor final.
- El exceso de error medio cuadrático usando esta solución será entonces

$$\begin{aligned} J_{ex}(n) &= \boldsymbol{\lambda}^T \mathbf{x}(n) \\ &= \sum_{i=1}^M c_i^n \boldsymbol{\lambda}^T \mathbf{g}_i \mathbf{g}_i^T [\mathbf{x}(0) - \mathbf{x}(\infty)] + \boldsymbol{\lambda}^T \mathbf{x}(\infty) \\ &= \sum_{i=1}^M c_i^n \boldsymbol{\lambda}^T \mathbf{g}_i \mathbf{g}_i^T [\mathbf{x}(0) - \mathbf{x}(\infty)] + J_{ex}(\infty) \end{aligned} \quad (87)$$

tal que el primer término del segundo miembro define el comportamiento transitorio del error medio cuadrático y el segundo término representa el valor final del exceso de error medio cuadrático después que la adaptación se ha completado (o sea el valor de estado estacionario).

Comportamiento del MSE en el transitorio

- Usando la ecuación (87) es posible expresar la evolución temporal del error medio cuadrático para el algoritmo LMS como sigue

$$J(n) = \sum_{i=1}^M \gamma_i c_i^n + J_{min} + J_{ex}(\infty) \quad (88)$$

donde c_i es el i -ésimo autovalor de la matriz \mathbf{B} y γ_i está definido por

$$\gamma_i = \boldsymbol{\lambda}^T \mathbf{g}_i \mathbf{g}_i^T [\mathbf{x}(0) - \mathbf{x}(\infty)], \quad i = 1, 2, \dots, M$$

- La ecuación (88) provee la base para un mejor entendimiento de la operación del algoritmo LMS en un ambiente ESA, como se describirá en la forma de cuatro propiedades.

Propiedad 1 La componente transitoria del error medio cuadrático, $J(n)$, no exhibe oscilaciones.

Propiedad 2 La componente transitoria del error medio cuadrático $J(n)$ desaparece, o sea, el algoritmo LMS es convergente en media cuadrática si y solo si el factor de convergencia μ satisface la condición

$$0 < \mu < \frac{2}{\lambda_{max}} \quad (89)$$

donde λ_{max} es el mayor autovalor de la matriz correlación \mathbf{R} .

Parámetros: M = número de coeficientes
μ = factor de convergencia tal que
$0 < \mu < \frac{2}{\text{potencia promedio de } u(n)}$
Datos: $\mathbf{u}(n)$ vector de entradas ($M \times 1$).
$d(n)$ respuesta deseada en el instante n .
Inicialización
$\hat{\mathbf{w}}(0) = \mathbf{0}$
Para cada n , $n = 1, 2, \dots$, calcular
$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n)$
$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)e^*(n)$

Tabla 2: Algoritmo LMS

Propiedad 3 El valor final del exceso de error medio cuadrático es menor que el error medio cuadrático mínimo si el factor de convergencia μ satisface la condición

$$\sum_{i=1}^M \frac{2\lambda_i}{2 - \mu\lambda_i} < 1 \quad (90)$$

donde λ_i , $i = 1, 2, \dots, M$, son los autovalores de la matriz correlación \mathbf{R} .

Propiedad 4 El desajuste, definido como la relación del valor de estado estacionario $J_{ex}(\infty)$ y el error medio cuadrático mínimo J_{min} , tiene la forma

$$\mathcal{M} = \frac{J_{ex}(\infty)}{J_{min}} = \sum_{i=1}^M \frac{\mu\lambda_i}{2 - \mu\lambda_i} \quad (91)$$

que es menor que la unidad si el factor de convergencia μ satisface la condición (90).

Algunas reglas prácticas

- El análisis de estabilidad y desempeño en términos del error medio cuadrático del algoritmo LMS operando en un ambiente ESA se basó en: a) el método de promediación directa, suponiendo μ suficientemente pequeño y b) la suposición de independencia.
- A pesar de esas suposiciones, en la práctica la teoría discutida se verifica razonablemente para un rango de valores de μ amplio.
- La condición para que el algoritmo LMS sea convergente en media cuadrática requiere el conocimiento del mayor autovalor λ_{max} , que en una aplicación típica no está disponible. Para contornar esta dificultad práctica, puede tomarse la traza de \mathbf{R} como una estimación conservativa de λ_{max} en cuyo caso

$$0 < \mu < \frac{2}{tr[\mathbf{R}]} \quad (92)$$

- Además, como \mathbf{R} es Toeplitz, los elementos de la diagonal principal son todos iguales a $r(0)$. Como, a su vez, $r(0)$ es el valor medio cuadrático de la entrada se tiene que $tr[\mathbf{R}] = M r(0) = \sum_{k=0}^{M-1} E[|u(n-k)|^2]$.
- De esta forma (92) puede reescribirse como

$$0 < \mu < \frac{2}{\text{potencia promedio de } u(n)} \quad (93)$$

- Otra ecuación que puede simplificarse en términos prácticos es la de (91) correspondiente al desajuste \mathcal{M} Para evitar el conocimiento

de los autovalores de \mathbf{R} , es posible utilizar la suposición de un valor pequeño para μ comparado con λ_{max} , tal que

$$\mathcal{M} \cong \frac{\mu}{2} \sum_{i=1}^M \lambda_i = \frac{\mu}{2} \text{potencia promedio de } u(n) \quad (94)$$

- Suponiendo que la curva de aprendizaje promedio del algoritmo LMS se aproxima por una única exponencial con constante de tiempo $(\tau)_{mse,av}$, entonces haciendo el mismo razonamiento que para el método SD, definimos la siguiente **constante de tiempo promedio** para el algoritmo LMS

$$(\tau)_{mse,av} \cong \frac{1}{2\mu\lambda_{av}}$$

tal que usando estos valores promedio en (94) se obtiene

$$\mathcal{M} \cong \frac{\mu M \lambda_{av}}{2} = \frac{M}{4\tau_{mse,av}} \quad (95)$$

de lo que es posible hacer las siguientes observaciones

1. El desajuste \mathcal{M} aumenta linealmente con la longitud del filtro M para un valor fijo de $\tau_{mse,av}$.
2. El **tiempo de convergencia** del algoritmo LMS (o sea el que toma para eliminar todo el transitorio) es proporcional a la constante de tiempo promedio $\tau_{mse,av}$. Además \mathcal{M} es inversamente proporcional a este tiempo de convergencia.
3. El desajuste \mathcal{M} es directamente proporcional al factor de convergencia μ , mientras que la constante de tiempo promedio es inversamente proporcional a μ . De aquí que es necesario satisfacer el compromiso entre bajo desajuste y tiempo de convergencia corto a través del factor de convergencia μ .

7.5 El algoritmo LMS normalizado

- En el algoritmo LMS estándar, la corrección que se aplica a $\hat{\mathbf{w}}(n+1)$ es directamente a $\mathbf{u}(n)$. Para $u(n)$ grande, el algoritmo LMS experimenta un problema de amplificación del ruido de gradiente. Para contornar este problema es posible utilizar el algoritmo LMS normalizado. En particular, la corrección aplicada a $\hat{\mathbf{w}}(n+1)$ se normaliza con respecto a la norma Euclidiana al cuadrado de $\mathbf{u}(n)$ en la iteración n .
- Es posible formular el algoritmo LMS normalizado como una modificación natural del algoritmo LMS estándar. Alternativamente, es posible obtener el algoritmo LMS normalizado en forma directa, lo que se muestra a continuación.
- El algoritmo LMS normalizado puede verse como la solución a un problema de optimización con restricciones. Específicamente, el problema de interés puede definirse como sigue:

Dado $\mathbf{u}(n)$ y $d(n)$, determinar $\hat{\mathbf{w}}(n+1)$ que minimize la norma Euclidiana al cuadrado del cambio

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

de $\hat{\mathbf{w}}(n+1)$ con respecto a $\hat{\mathbf{w}}(n)$, sujeta a la restricción

$$\hat{\mathbf{w}}^H(n+1)\mathbf{u}(n) = d(n) \quad (96)$$

- Para resolver este problema de optimización con restricciones utilizaremos el método de multiplicadores de Lagrange. La norma al cuadrado del cambio $\delta \hat{\mathbf{w}}(n+1)$ puede expresarse como

$$\begin{aligned}
 \|\delta \hat{\mathbf{w}}(n+1)\|^2 &= \delta \hat{\mathbf{w}}^H(n+1) \delta \hat{\mathbf{w}}(n+1) \\
 &= [\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)]^H [\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)] \\
 &= \sum_{k=0}^{M-1} |\hat{w}_k(n+1) - \hat{w}_k(n)|^2
 \end{aligned}$$

- Definiendo los coeficientes $\hat{w}_k(n)$, $k = 0, 1, \dots, M-1$ en términos de sus partes real e imaginaria se obtiene

$$\hat{w}_k(n) = a_k(n) + jb_k(n), \quad k = 0, 1, \dots, M-1$$

- Entonces se tiene que

$$\|\delta \hat{\mathbf{w}}(n+1)\|^2 = \sum_{k=0}^{M-1} ([a_k(n+1) - a_k(n)]^2 + [b_k(n+1) - b_k(n)]^2)$$

- Expresando $u(n-k)$ y $d(n)$ en términos de sus partes real e imaginaria, es posible escribir la restricción (96) como

$$\begin{aligned}
 \sum_{k=0}^{M-1} (a_k(n+1)u_1(n-k) + b_k(n+1)u_2(n-k)) &= d_1(n) \\
 \sum_{k=0}^{M-1} (a_k(n+1)u_2(n-k) - b_k(n+1)u_1(n-k)) &= d_2(n)
 \end{aligned}$$

de forma que la función costo $J(n)$ para el problema de optimización puede formularse como

$$\begin{aligned}
 J(n) = & \sum_{k=0}^{M-1} ([a_k(n+1) - a_k(n)]^2 + [b_k(n+1) - b_k(n)]^2) \\
 & + \lambda_1 \left[d_1(n) - \sum_{k=0}^{M-1} (a_k(n+1)u_1(n-k) + b_k(n+1)u_2(n-k)) \right] \\
 & + \lambda_2 \left[d_2(n) - \sum_{k=0}^{M-1} (a_k(n+1)u_2(n-k) - b_k(n+1)u_1(n-k)) \right]
 \end{aligned}$$

donde λ_1 y λ_2 son los **multiplicadores de Lagrange**. Para hallar los valores óptimos de $a_k(n+1)$ y $b_k(n+1)$ diferenciamos $J(n)$ con respecto a esos parámetros e igualamos el resultado a cero. Entonces

$$\frac{\partial J(n)}{\partial a_k(n+1)} = 0$$

lo que produce el siguiente resultado

$$2[a_k(n+1) - a_k(n)] - \lambda_1 u_1(n-k) - \lambda_2 u_2(n-k) = 0$$

y similarmente a partir

$$\frac{\partial J(n)}{\partial b_k(n+1)} = 0$$

de donde se obtiene

$$2[b_k(n+1) - b_k(n)] - \lambda_1 u_2(n-k) - \lambda_2 u_1(n-k) = 0$$

- Luego, usando la definición de los coeficientes complejos $\hat{w}_k(n)$ es posible combinar estos últimos resultados reales como sigue

$$2[\hat{w}_k(n+1) - \hat{w}_k(n)] = \lambda^* u(n-k), \quad k = 0, 1, \dots, M-1 \quad (97)$$

donde $\lambda = \lambda_1 + j\lambda_2$ es un multiplicador de Lagrange complejo. Para resolver por la incógnita λ^* , multiplicamos ambos lados de (97) por $u^*(n-k)$ y luego sumamos sobre todos los enteros entre 0 y $M-1$, para obtener

$$\begin{aligned} \lambda^* &= \frac{2}{\sum_{k=0}^{M-1} |u(n-k)|^2} \left[\sum_{k=0}^{M-1} \hat{w}_k(n+1) u^*(n-k) - \sum_{k=0}^{M-1} \hat{w}_k(n) u^*(n-k) \right] \\ &= \frac{2}{\|\mathbf{u}(n)\|^2} [\hat{\mathbf{w}}^T(n+1) \mathbf{u}^*(n) - \hat{\mathbf{w}}^T(n) \mathbf{u}^*(n)] \end{aligned}$$

donde $\|\mathbf{u}(n)\|$ es la norma Euclidiana de $\mathbf{u}(n)$. Utilizando ahora la restricción compleja en la ecuación anterior es posible obtener

$$\begin{aligned} \lambda^* &= \frac{2}{\|\mathbf{u}(n)\|^2} [d^*(n) - \hat{\mathbf{w}}^T(n) \mathbf{u}^*(n)] \\ &= \frac{2}{\|\mathbf{u}(n)\|^2} e^*(n) \end{aligned}$$

donde se utilizó la definición del error de estimación $e(n)$. Finalmente, sustituyendo esta ecuación en (97) se obtiene

$$\begin{aligned} \delta \hat{w}_k(n+1) &= \hat{w}_k(n+1) - \hat{w}_k(n) \\ &= \frac{1}{\|\mathbf{u}(n)\|^2} u(n-k) e^*(n), \quad k = 0, 1, \dots, M-1 \\ &= \frac{1}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n) e^*(n) \end{aligned}$$

- Para controlar el cambio del vector de coeficientes de una iteración a la siguiente sin cambiar su dirección, introducimos un factor de escalamiento positivo real denotado por $\bar{\mu}$, tal que

$$\begin{aligned}\delta \hat{w}_k(n+1) &= \hat{w}_k(n+1) - \hat{w}_k(n) \\ &= \frac{\bar{\mu}}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n) e^*(n)\end{aligned}$$

o equivalentemente

$$\hat{w}_k(n+1) = \hat{w}_k(n) + \frac{\bar{\mu}}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n) e^*(n) \quad (98)$$

que determina la ecuación de adaptación para el algoritmo LMS normalizado.

- El aspecto importante a notar es que este algoritmo ajusta $\hat{\mathbf{w}}(n+1)$ en términos del **mínimo cambio** (en el sentido de la norma Euclidiana) con respecto a $\hat{\mathbf{w}}(n)$.
- Comparando la recursión de (98) con la del algoritmo LMS estándar puede realizarse las siguientes observaciones
 - La constante de adaptación $\bar{\mu}$ para el algoritmo LMS normalizado es **adimensional**, mientras que para el estándar tiene dimensiones de **potencia inversa**.
 - Haciendo

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

es posible interpretar al algoritmo normalizado como el algoritmo LMS estándar con **factor de convergencia variante en el tiempo**.

- El algoritmo LMS normalizado es **convergente en media cuadrática** si la constante de adaptación $\bar{\mu}$ satisface la siguiente condición

Parámetros: M = número de coeficientes
$\bar{\mu}$ = constante de adaptación tal que
$0 < \bar{\mu} \leq 2$
a = constante positiva
Datos: $\mathbf{u}(n)$ vector de entradas ($M \times 1$).
$d(n)$ respuesta deseada en el instante n .
Inicialización
$\hat{\mathbf{w}}(0) = \mathbf{0}$
Para cada n , $n = 1, 2, \dots$, calcular
$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n)$
$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\bar{\mu}}{a + \ \mathbf{u}(n)\ ^2} \mathbf{u}(n)e^*(n)$

Tabla 3: Algoritmo LMS normalizado

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

- Otro aspecto importante es que el algoritmo LMS normalizado exhibe una velocidad de convergencia que es potencialmente mayor que la del algoritmo LMS estándar tanto para entradas decorrelacionadas como correlacionadas.
- Aparte del interés en contornar el problema de amplificación de ruido de gradiente, el algoritmo normalizado introduce un problema particular. Específicamente, cuando $\|\mathbf{u}(n)\|$ es pequeña, pueden aparecer dificultades numéricas. Para evitar este problema es posible modificar levemente la ecuación (98) como sigue

$$\hat{w}_k(n+1) = \hat{w}_k(n) + \frac{\bar{\mu}}{a + \|\mathbf{u}(n)\|^2} \mathbf{u}(n)e^*(n) \quad (99)$$

donde $a > 0$ es una constante suficientemente pequeña.