IEE352 - Procesamiento Digital de Señales

Clase 13: Filtros Adaptivos 2

Dr. Marco A. Milla
Sección Electricidad y Electrónica (SEE)
Pontificia Universidad Católica del Perú (PUCP)

email: milla.ma@pucp.edu.pe

Variaciones del Algoritmo LMS

Variaciones Algoritmo LMS

Leaky LMS

- Cuando $d[n]$ y $x[n]$ son procesos WSS, la convergencia del algoritmo LMS ($\bar{w}^n \to \bar{w}_{\text{opt}}$) está relacionada con los autovalores de la matriz de autocorrelación $\mathbf{R}_x$.
- Si uno de los autovalores $\lambda_k = 0$, entonces el algoritmo LMS perdería estabilidad (convergencia) cuando $n \to \infty$.
- **Solución:** Un modelo amortiguado (algoritmo leaky LMS):

$$\bar{w}^{n+1} = (1 - \mu\gamma)\bar{w}^n + \mu e[n]\bar{X}[n]$$

  donde $0 < \gamma \leq 1$ es el coeficiente de fuga que permite que los autovalores sean $\lambda_k + \gamma$. De esta forma aseguramos estabilidad del algoritmo (relacionado con el método de regularización para problemas inversos).

- **Desventaja:** Cuando se llega a la estabilidad en la solución, hay un diferencial entre el filtro estimado y el valor óptimo:

$$\bar{w}^n \nrightarrow \bar{w}_{\text{opt}}.$$

Variaciones Algoritmo LMS

LMS por bloques

- Algoritmos LMS con complejidad reducida: Algunas aplicaciones, por ejemplo, comunicación digital de alta velocidad, requieren algoritmos computacionalmente eficientes.
  - LMS clásico: Los coeficientes $\bar{w}^n$ se actualizan en cada iteración.
  - LMS por bloques: Los coeficientes $\bar{w}^{kL}$ se actualizan cada cierto número de muestras.
- **Algoritmo LMS por bloques:** Similar al algoritmo LMS solo que los coeficientes del filtro son actualizados una vez en cada bloque de L muestras $\bar{w}^n = \bar{w}^{kL}$,

$$\bar{w}^{(k+1)L} = \bar{w}^{kL} + \mu \frac{1}{L} \sum_{l=0}^{L-1} e[kL+l]\, \bar{X}[kL+l] \,.$$

Variaciones Algoritmo LMS

Sign LMS

- **Algoritmos sign LMS:** Consiste en utilizar el operador signo $\text{sgn}[]$ en el error $e[n]$, en la data $\bar{X}[n]$ o en ambos, con el objetivo de disminuir el número de multiplicaciones.
  - **Signo Error:** $\bar{w}^{n+1} = \bar{w}^{n} + \mu\, \text{sgn}\{e[n]\}\bar{X}[n]$
  - **Signo Data:** $\bar{w}^{n+1} = \bar{w}^{n} + \mu\, e[n]\, \text{sgn}\{\bar{X}[n]\}$
  - **Signo-Signo:** $\bar{w}^{n+1} = \bar{w}^{n} + \mu\, \text{sgn}\{e[n]\}\, \text{sgn}\{\bar{X}[n]\}$

donde:

$$\text{sgn}(y) = \begin{cases} 1, & y > 0, \\ 0, & y = 0, \\ -1, & y < 0. \end{cases}$$

- **Desventaja:** Aumenta el error en la gradiente
  - **Algoritmo signo error:** modifica la magnitud de la gradiente, pero no la dirección.
  - **Algoritmo signo data:** altera la dirección (puede divergir).
  - **Algoritmo signo-signo:** tiene ambos problemas y su convergencia es más lenta.

Variaciones Algoritmo LMS

Algoritmos de paso variable

- Se selecciona el tamaño de paso μ con el objetivo de lograr un balance entre la razón de convergencia y la cantidad excedente de error.
    - Cuando $\bar{w}^n$ está lejos de la solución óptima $\bar{w}_{\text{opt}}$, el paso μ debería ser grande con el fin de movernos rápidamente hacia la solución deseada.
    - Cuando $\bar{w}^n$ es estable, coeficientes casi constantes, el tamaño de paso μ debería disminuir para reducir el exceso de error.
- **Algoritmo:** $$w_k[n+1] = w_k[n] + \mu_k[n]e[n]x[n-k], \qquad k = 0,1,\ldots M-1,$$

  donde $\mu_k[n]$ es el k-ésimo valor del paso en el tiempo n.
- **Regla de cambio:** Basada en el signo de la gradiente $g_k[n] = \text{sgn}\{e[n]x[n-k]\}$
    - Si tiene el mismo signo en m_1 muestras sucesivas, entonces $\mu_k[n+1] = c_1\mu_k[n] \quad (\uparrow \mu)$
    - Si cambia de signo en m_2 muestras sucesivas, entonces $\mu_k[n+1] = c_2\mu_k[n] \quad (\downarrow \mu)$

  donde $\mu_{\min} < \mu_k[n] < \mu_{\max}$, y $0 < c_2 < 1 < c_1$ son constantes de decremento e incremento.

Variaciones Algoritmo LMS

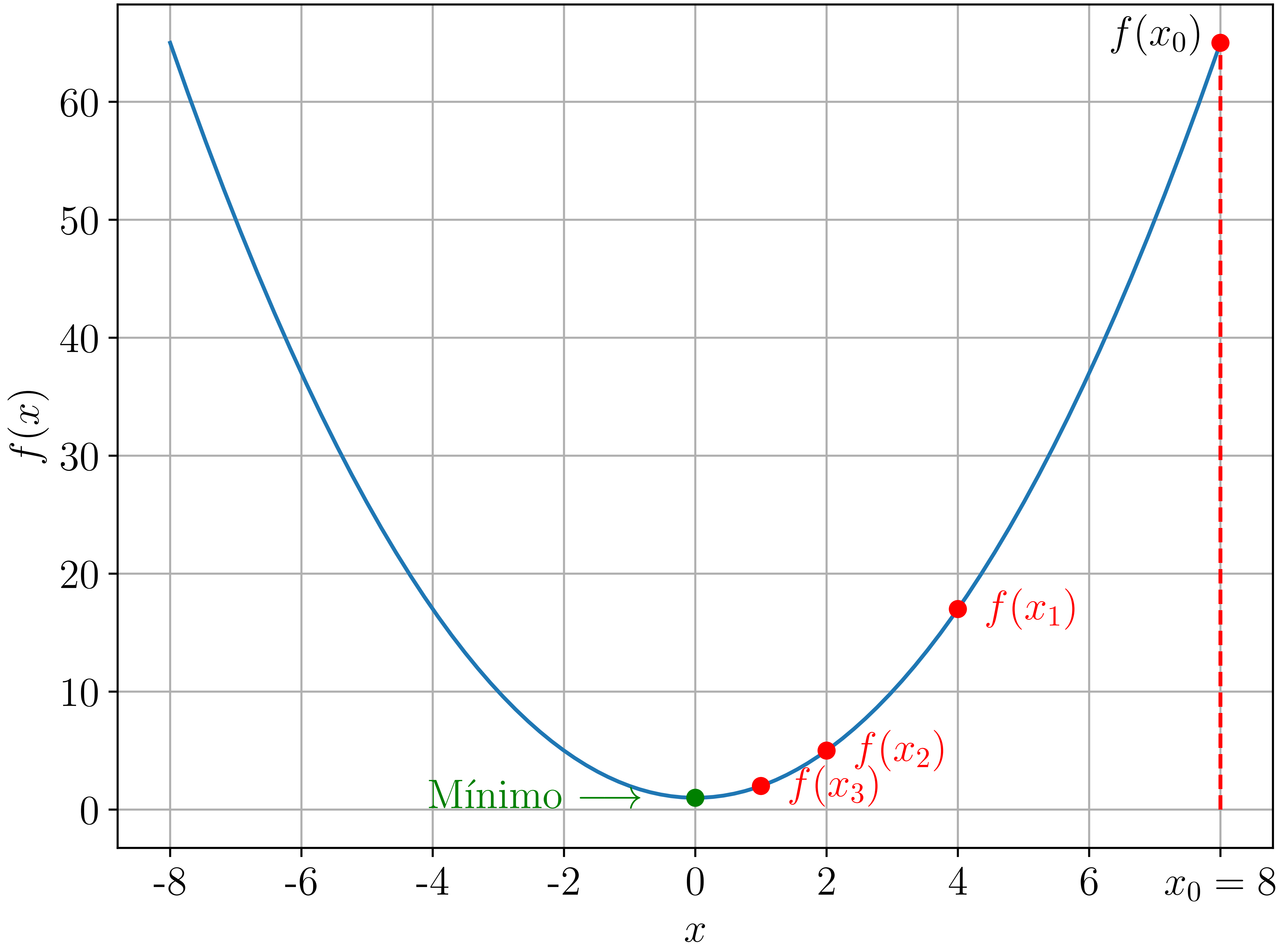
Algoritmos de paso variable - Ejercicio

Dada la función $f(x) = x^2 + 1$, encontrar el punto x que minimiza $f(x)$.

Algoritmo: $x_{k+1} = x_k - \mu \nabla F(x_k)$

Gradiente: $\nabla f(x) = \dfrac{\partial f}{\partial x} = 2x$ (dirección)

Inicializando: $x_0 = 8$ y $\mu = 0.25$



Algoritmo Recursivo de Mínimos Cuadrados (RLS)

Algoritmo Recursivo de Mínimos Cuadrados (RLS)

- En los algoritmos previos el objetivo era minimizar el error cuadrático medio (MSE)

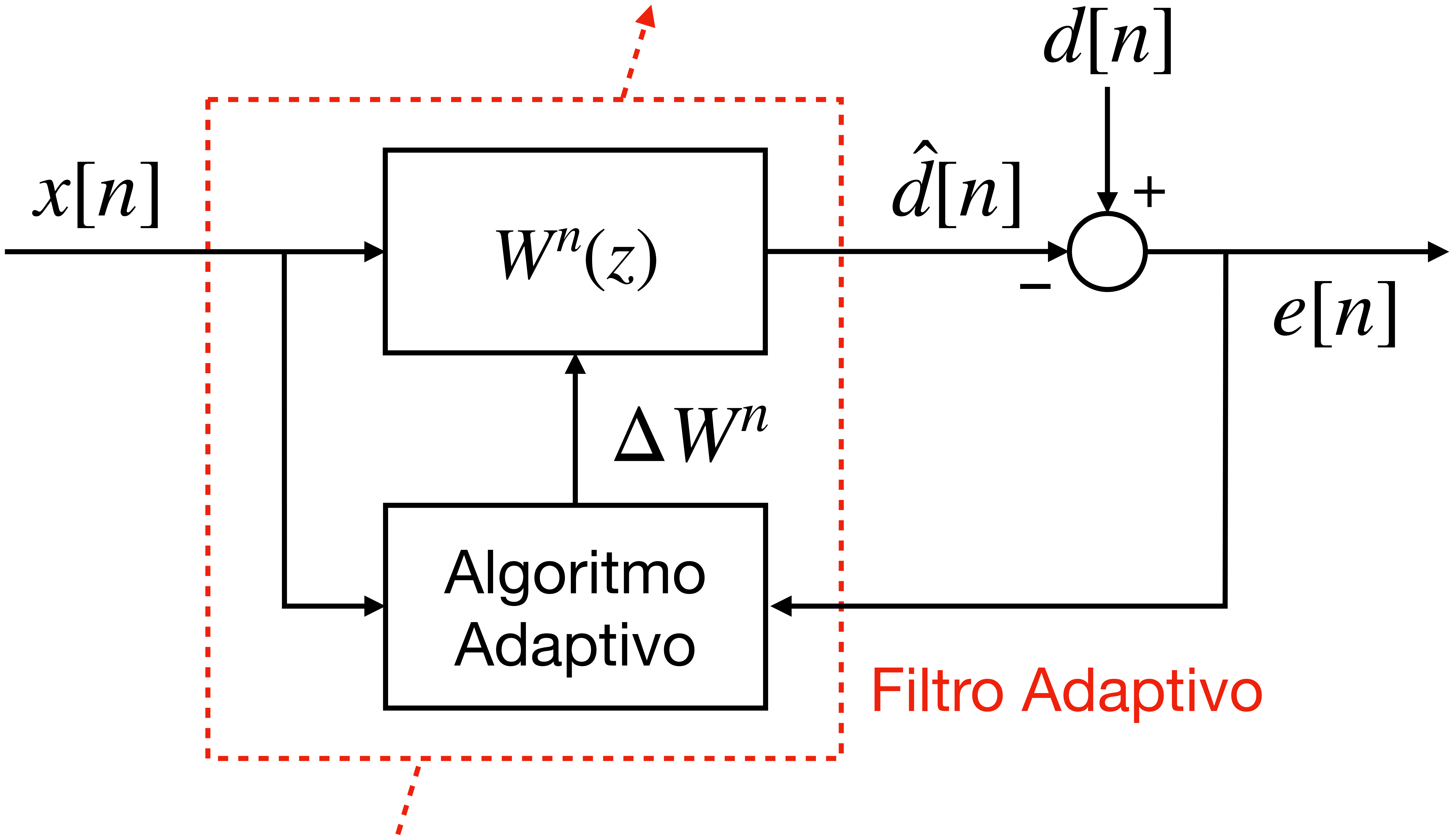
$$\xi[n] = E\{\,|e[n]|^2\,\}\ .$$

- **Dificultad:** Requiere que los valores estadísticos sean conocidos.
  - Wiener: Autocorrelación $r_x = E\{x[n]x[n-k]\}$ y correlación cruzada $r_{dx} = E\{d[n]x[n-k]\}$
  - LMS: $\hat{E}\{e[n]x[n-k]\} = e[n]x[n-k]$
- El Algoritmo LMS es adecuado en algunas aplicaciones, en otras la gradiente estimada no proporciona una tasa de convergencia suficientemente rápida, o un error suficientemente pequeño.
- **Alternativa:** Considerar una medida de error que no incluya esperanza y que pueda ser computada directamente de los datos temporales

$$\varepsilon[n] = \sum_{j=0}^{n} |e[j]|^2\ .$$

Algoritmo Recursivo de Mínimos Cuadrados (RLS)

Esquema general de Filtro Adaptivo



Algoritmo Recursivo de Mínimos Cuadrados (RLS)

RLS con ponderación exponencial

Se desea diseñar un filtro FIR adaptivo con coeficientes

$$\bar{w}^n = \left[w_0[n], w_1[n], \ldots, w_{M-1}[n]\right]^T,$$

que minimice, en cada tiempo n, el error de mínimos cuadrados ponderados

$$\varepsilon[n] = \sum_{j=0}^{n} \lambda^{n-j} |e[j]|^2$$

donde $0 < \lambda \leq 1$ es un factor de ponderación exponencial (factor de olvido) y

$$e[j] = d[j] - \hat{d}[j] = d[j] - (\bar{w}^n)^T \bar{X}[j] .$$

Algoritmo Recursivo de Mínimos Cuadrados (RLS)

RLS con ponderación exponencial

Para minimizar el error cuadrático ponderado, se calcula la derivada de la siguiente forma

$$\frac{\partial \varepsilon[n]}{\partial w_k[n]} = \sum_{j=0}^{n} \lambda^{n-j} e[j] \frac{\partial e[j]}{\partial w_k[n]} = -\sum_{j=0}^{n} \lambda^{n-j} e[j] x[j-k] = 0 \,,$$

utilizando la definición de $e[j]$ tenemos que

$$\sum_{j=0}^{n} \lambda^{n-j} \left\{ d[j] - \sum_{l=0}^{M-1} w_l[n] x[j-l] \right\} x[j-k] = 0$$

$$\sum_{l=0}^{M-1} w_l[n] \left[\sum_{j=0}^{n} \lambda^{n-j} x[j-l] x[j-k] \right] = \sum_{j=0}^{n} \lambda^{n-j} d[j] x[j-k]$$

entonces

$$\mathbf{R}_x[n]\, \bar{w}^n = \bar{r}_{dx}[n] \,.$$

Algoritmo Recursivo de Mínimos Cuadrados (RLS)

RLS con ponderación exponencial

Autocorrelación determinística con ponderación exponencial

$$\mathbf{R}_x[n] = \sum_{j=0}^{n} \lambda^{n-j} \bar{X}[j] \bar{X}^T[j] \,.$$

Correlación cruzada deterministíca con ponderación exponencial

$$\bar{r}_{dx}[n] = \sum_{j=0}^{n} \lambda^{n-j} d[j] \bar{X}[j] \,.$$

donde $\bar{X}[j]$ es un vector columna de la forma

$$\bar{X}[j] = \left[x[j], x[j-1], \ldots, x[j-M+1]\right]^T .$$

Algoritmo Recursivo de Mínimos Cuadrados (RLS)

Derivando ecuación recursiva

Punto de partida $\bar{w}^n = \mathbf{R}_x[n]^{-1}\,\bar{r}_{dx}[n]$, el objetivo es derivar una ecuación recursiva de la forma

$$\bar{w}^{n+1} = \bar{w}^n + \Delta\bar{w}^n\,.$$

Se expresa la autocorrelación de forma recursiva

$$\begin{aligned}\mathbf{R}_x[n] &= \sum_{j=0}^{n} \lambda^{n-j}\bar{X}[j]\bar{X}^T[j] = \sum_{j=0}^{n-1} \lambda^{n-j}\bar{X}[j]\bar{X}^T[j] + \lambda^0\bar{X}[n]\bar{X}^T[n] \\ &= \lambda\mathbf{R}_x[n-1] + \bar{X}[n]\bar{X}^T[n]\,.\end{aligned}$$

Fórmula Sherman-Morrison: Es una técnica utilizada para invertir matrices de la forma

$$(\mathbf{A} + \bar{u}\bar{v}^H)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\bar{u}\bar{v}^H\mathbf{A}^{-1}}{1 + \bar{v}^H\mathbf{A}^{-1}\bar{u}}$$

donde $\bar{u}\bar{v}^H$ es el producto externo de los vectores $\bar{u}$ y $\bar{v}$.

Algoritmo Recursivo de Mínimos Cuadrados (RLS)

Derivando ecuación recursiva

Utilizando la fórmula Sherman-Morrison para invertir la matriz de autocorrelación $\mathbf{R}_x[n]$ tenemos

$$\begin{aligned}\mathbf{R}_x^{-1}[n] &= \left(\lambda \mathbf{R}_x[n-1] + \bar{X}[n]\bar{X}^T[n]\right)^{-1} \\ &= \lambda^{-1}\mathbf{R}_x^{-1}[n-1] - \frac{\lambda^{-2}\mathbf{R}_x^{-1}[n-1]\bar{X}[n]\bar{X}^T[n]\mathbf{R}_x^{-1}[n-1]}{1+\lambda^{-1}\bar{X}^T[n]\mathbf{R}_x^{-1}[n-1]\bar{X}[n]}.\end{aligned}$$

Aplicando un cambio de variable $\mathbf{P}[n] = \mathbf{R}_x^{-1}[n]$, se reescribe la inversa de la matriz de autocorrelación de la siguiente forma

$$\mathbf{P}[n] = \lambda^{-1}\left[\mathbf{P}[n-1] - \bar{G}[n]\,\bar{X}^T[n]\,\mathbf{P}[n-1]\right]$$

donde

$$\bar{G}[n] = \frac{\lambda^{-1}\,\mathbf{P}[n-1]\,\bar{X}[n]}{1+\lambda^{-1}\,\bar{X}^T[n]\,\mathbf{P}[n-1]\,\bar{X}[n]}.$$

Algoritmo Recursivo de Mínimos Cuadrados (RLS)

Derivando ecuación recursiva

Expandiendo tenemos que

$$\bar{G}[n] + \lambda^{-1}\,\bar{G}[n]\,\bar{X}^T[n]\,\mathbf{P}[n-1]\,\bar{X}[n] = \lambda^{-1}\,\mathbf{P}[n-1]\,\bar{X}[n]$$

$$\bar{G}[n] = \underbrace{\lambda^{-1}\left[\mathbf{P}[n-1] - \bar{G}[n]\,\bar{X}^T[n]\,\mathbf{P}[n-1]\right]}_{\mathbf{P}[n]}\,\bar{X}[n]$$

Volviendo al punto de partida

$$\bar{w}^n = \mathbf{P}[n]\,\bar{r}_{dx}[n] \quad \Longrightarrow \quad \bar{w}^{n+1} = \bar{w}^n + \Delta\bar{w}^n,$$

tenemos que

$$\bar{r}_{dx}[n] = \sum_{j=0}^{n} \lambda^{n-j} d[j]\bar{X}[j] = \sum_{j=0}^{n-1} \lambda^{n-j} d[j]\bar{X}[j] + \lambda^0 d[n]\bar{X}[n]$$

$$= \lambda\bar{r}_{dx}[n-1] + d[n]\bar{X}[n]$$

luego

$$\bar{w}^n = \mathbf{P}[n]\left[\lambda\bar{r}_{dx}[n-1] + d[n]\bar{X}[n]\right].$$

Algoritmo Recursivo de Mínimos Cuadrados (RLS)

Derivando ecuación recursiva

Expandiendo $\bar{w}^n = \mathbf{P}[n]\left[\lambda \bar{r}_{dx}[n-1] + d[n]\bar{X}[n]\right]$ tenemos que

$$\bar{w}^n = \lambda \mathbf{P}[n]\bar{r}_{dx}[n-1] + d[n]\mathbf{P}[n]\bar{X}[n]$$

$$\bar{w}^n = \left[\mathbf{P}[n-1] - \bar{G}[n]\bar{X}^T[n]\mathbf{P}[n-1]\right]\bar{r}_{dx}[n-1] + d[n]\bar{G}[n]$$

Recordar que $\bar{w}_n = \mathbf{P}[n]\bar{r}_{dx}[n]$, entonces

$$\bar{w}^n = \underbrace{\mathbf{P}[n-1]\bar{r}_{dx}[n-1]}_{\bar{w}^{n-1}} - \bar{G}[n]\bar{X}^T[n]\underbrace{\mathbf{P}[n-1]\bar{r}_{dx}[n-1]}_{\bar{w}^{n-1}} + d[n]\bar{G}[n]$$

Finalmente:

$$\bar{w}^n = \bar{w}^{n-1} + \bar{G}[n]\left[d[n] - \bar{X}^T[n]\bar{w}^{n-1}\right].$$

Algoritmo Recursivo de Mínimos Cuadrados (RLS)

Implementación

Algoritmo 2: Algoritmo RLS con ponderación exponencial para un filtro adaptivo FIR de longitud M.

Parámetros : M = Longitud del filtro

λ = Factor de ponderación exponencial ($0 < \lambda \leq 1$)

δ = Valor inicial de $\mathbf{P}[0]$

Inicialización : $\bar{w}^0 = [0,0,\ldots,0]^T$

$\mathbf{P}[0] = \delta\mathbf{I}$

$x[k] = d[k] = 0, \quad k < 0,$ luego $\bar{X}[0] = [x[0],0,\ldots,0]^T$

Computar : For $n = 1,2,\ldots$

(a) $\bar{Z}[n] = \mathbf{P}[n-1]\bar{X}[n]$

(b) $\alpha[n] = d[n] - \hat{d}[n] = d[n] - (\bar{w}^{n-1})^T\bar{X}[n]$

(c) $\bar{G}[n] = \dfrac{1}{\lambda + \bar{X}^T[n]\bar{Z}[n]}\bar{Z}[n]$

(d) $\bar{w}^n = \bar{w}^{n-1} + \alpha[n]G[n]$

(e) $\mathbf{P}[n] = \lambda^{-1}\left[\mathbf{P}[n-1] - \bar{G}[n]\bar{Z}^H[n]\right]$

Algoritmo Recursivo de Mínimos Cuadrados (RLS)

Relación entre $\mathbf{P}[n-1] \rightarrow \bar{G}[n] \rightarrow \Delta\bar{w}^n$

$$\Delta\bar{w}^1 \iff P[0] = \left(0.5^0\,\mathbf{R}_x[0]\right)^{-1}$$

$$\Delta\bar{w}^2 \iff P[1] = \left(0.5^0\,\mathbf{R}_x[1] + 0.5^1\,\mathbf{R}_x[0]\right)^{-1}$$

$$\Delta\bar{w}^3 \iff P[2] = \left(0.5^0\,\mathbf{R}_x[2] + 0.5^1\,\mathbf{R}_x[1] + 0.5^2\,\mathbf{R}_x[0]\right)^{-1}$$

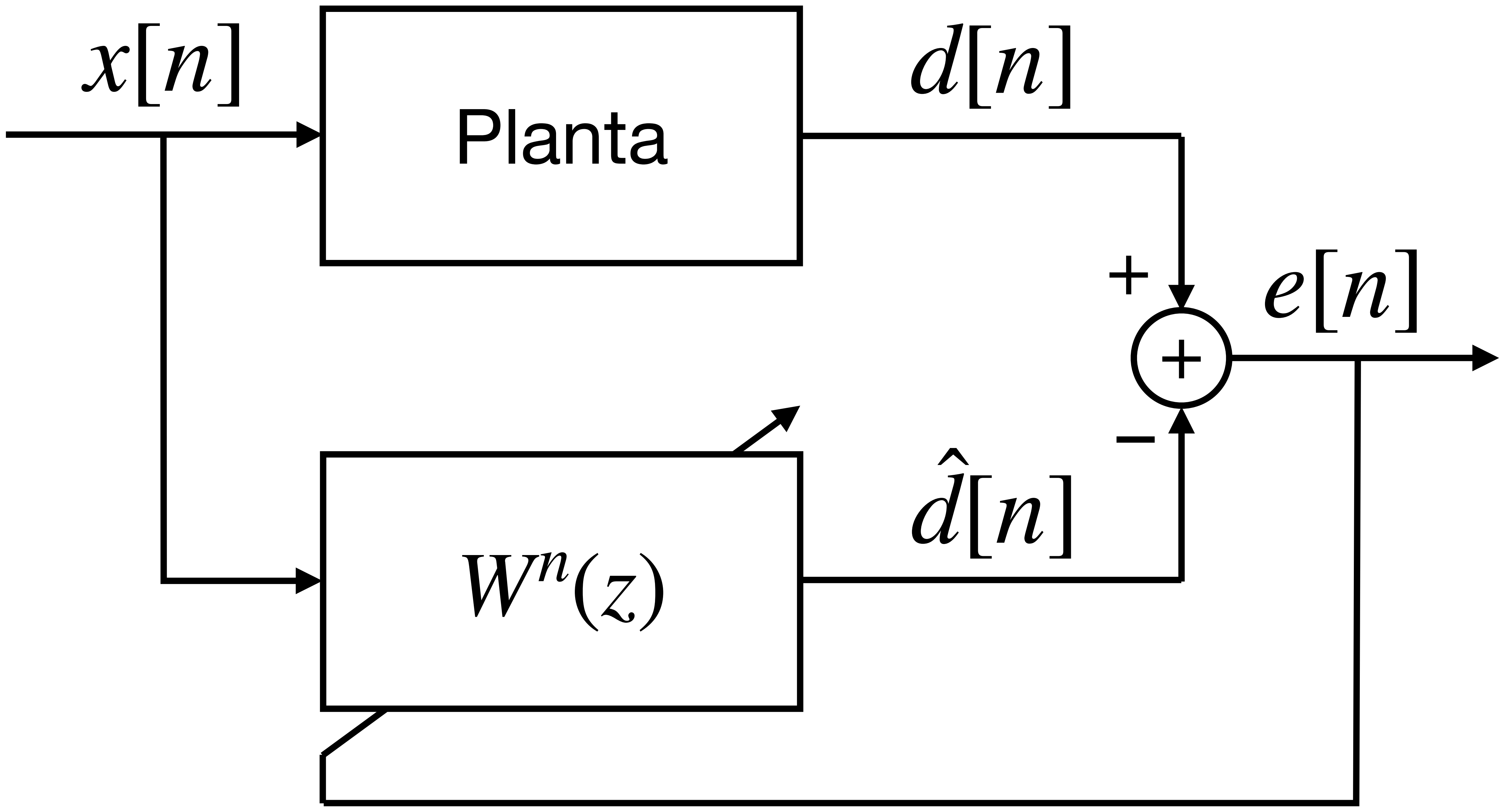
$$\Delta\bar{w}^4 \iff P[3] = \left(0.5^0\,\mathbf{R}_x[3] + 0.5^1\,\mathbf{R}_x[2] + 0.5^2\,\mathbf{R}_x[1] + 0.5^3\,\mathbf{R}_x[0]\right)^{-1}$$

$$\vdots$$

Filtros Adaptivos - Casos

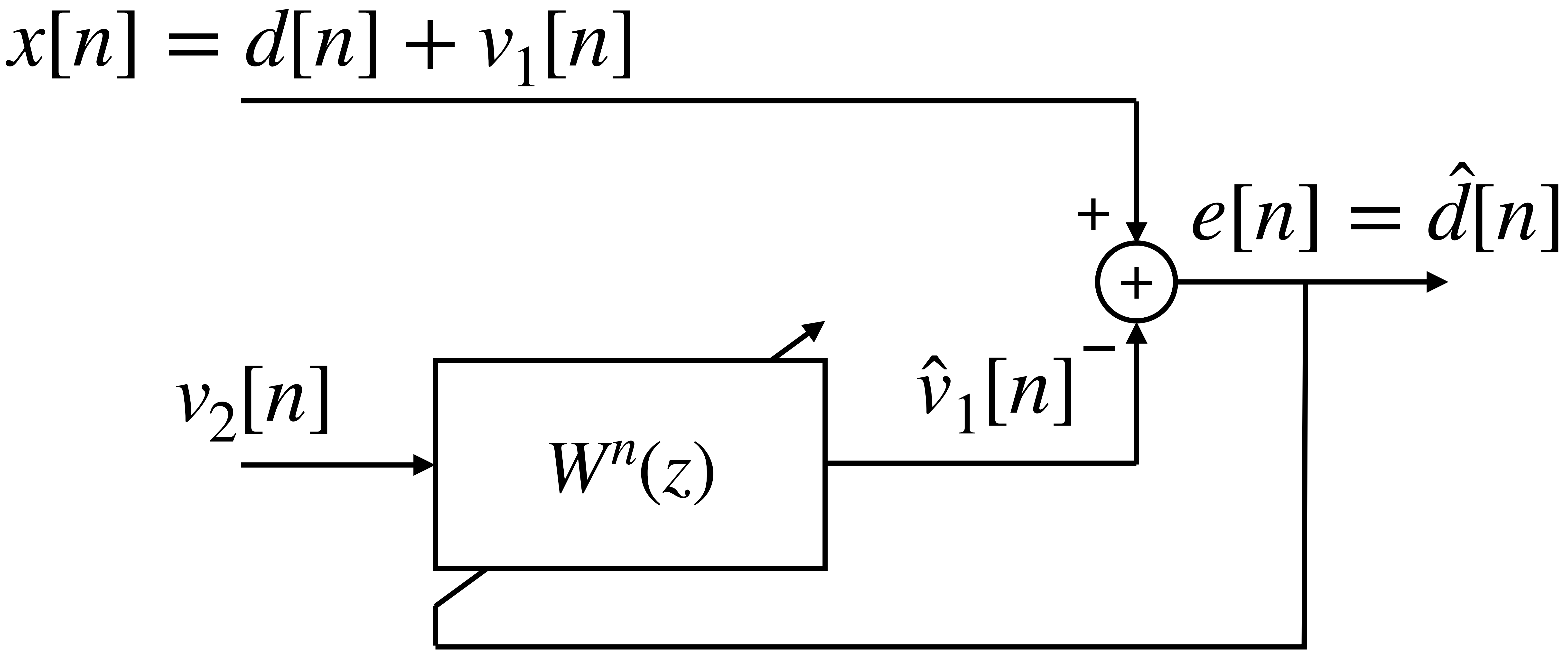
Identificador de sistema

Recordar: El algoritmo RLS estima filtros de la forma $W^n(z) = \sum_{k=0}^{M-1} w_k[n] z^{-k}$.



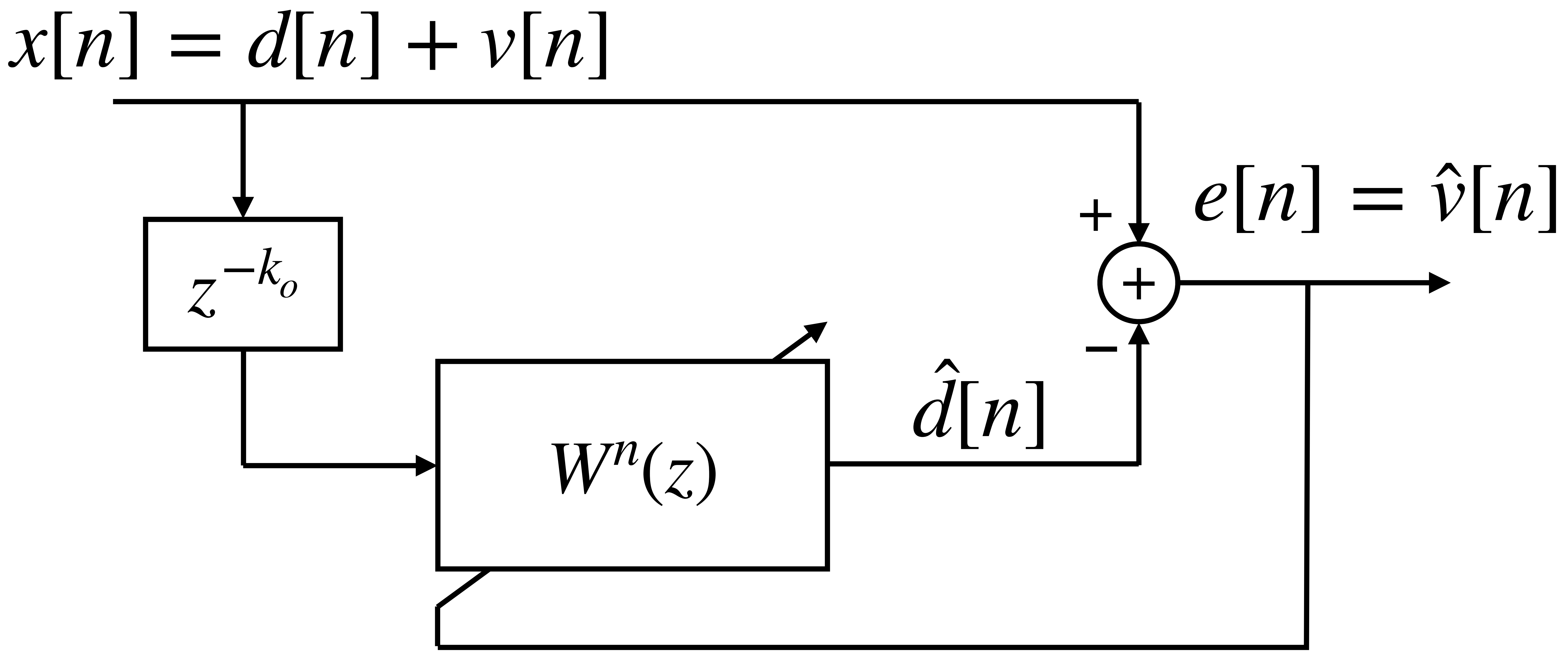
Filtros Adaptivos - Casos

Cancelador de ruido



Filtros Adaptivos - Casos

Esquema de un predictor lineal



Algoritmo RLS con Ponderación Exponencial

Observaciones

- En comparación a LMS, RLS involucra más cálculos (mayor complejidad computacional), sin embargo, RLS generalmente converge más rápido y su estabilidad es menos sensible a los autovalores.
- Cuando $\lambda = 1$, el algoritmo RLS con ponderación exponencial es conocido como algoritmo RLS con ventana creciente (los errores cuadráticos $|e[j]|^2$ son igualmente ponderados desde $j = 0$ hasta $j = n$).
- **Posible desventaja:** Toda la data afecta la estimación de los coeficientes.
- En el caso de un proceso no estacionario (las estadísticas cambian rápidamente en el tiempo n), sería necesario un valor adecuado de λ (factor pequeño) para poder seguir las variaciones del proceso.

Algoritmo RLS con ventana desplazable

- **Idea:** Con el objetivo de lidiar de manera más fácil con procesos no estacionarios una ventana finita permitiría olvidar datos luego de ciertas iteraciones.
- **RLS con ventana desplazable:** Algoritmo que minimiza la suma de errores cuadráticos sobre una ventana finita longitud L.

$$\varepsilon_L[n] = \sum_{j=n-L+1}^{n} |e[j]|^2 .$$

Algoritmo RLS con ventana desplazable

- El error cuadrático considerando una ventana de longitud L se puede expresar de la siguiente forma

$$\varepsilon_L[n] = \sum_{j=n-L+1}^{n} |e[j]|^2 \implies \varepsilon_L[n] = \sum_{j=0}^{n} |e[j]|^2 - \sum_{j=0}^{n-L} |e[j]|^2 .$$

- Por ejemplo considerando $L = 4$, la inversa de la matriz de correlación debería ser calculada solo usando las últimas cuatro muestras.

$$\tilde{\mathbf{P}}[5] = \left(\mathbf{R}_x[5] + \mathbf{R}_x[4] + \mathbf{R}_x[3] + \mathbf{R}_x[2] + \mathbf{R}_x[1] + \mathbf{R}_x[0]\right)^{-1}$$

$$\mathbf{P}[5] = \left(\mathbf{R}_x[5] + \mathbf{R}_x[4] + \mathbf{R}_x[3] + \mathbf{R}_x[2]\right)^{-1}$$

Algoritmo RLS con ventana desplazable

Algoritmo 3: Algoritmo RLS con ventana desplazable para un filtro FIR de longitud M.

Parámetros e Inicialización : Igual a algoritmo RLS

Computar : For $n = 0,1,2,\ldots$

(a) $\bar{G}[n] = \dfrac{1}{1 + \bar{X}^T[n]\mathbf{P}[n-1]\bar{X}[n]}\mathbf{P}[n-1]\bar{X}[n]$

(b) $\tilde{w}^n = \bar{w}^{n-1} + \bar{G}[n]\left[d[n] - (\bar{w}^{n-1})^T\bar{X}[n]\right]$

(c) $\tilde{\mathbf{P}}[n] = \mathbf{P}[n-1] - \bar{G}[n]\bar{X}^T[n]\mathbf{P}[n-1]$

(d) $\tilde{G}[n] = \dfrac{1}{1 - \bar{X}^T[n-L]\tilde{\mathbf{P}}[n]\bar{X}[n-L]}\tilde{\mathbf{P}}[n]\bar{X}[n-L]$

(e) $\bar{w}^n = \tilde{w}^n - \tilde{G}[n]\left[d[n-L] - (\tilde{w}^n)^T\bar{X}[n-L]\right]$

(f) $\mathbf{P}[n] = \tilde{\mathbf{P}}[n] + \tilde{G}[n]\bar{X}^T[n-L]\tilde{\mathbf{P}}[n]$

¡Muchas gracias!