

Práctico 4

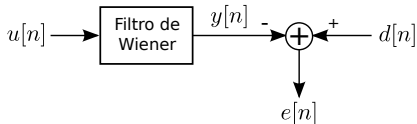
Filtros adaptivos: descenso por gradiente y algoritmo LMS

ESTIMACIÓN y PREDICCIÓN en SERIES TEMPORALES

Departamento de Procesamiento de Señales
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería

Curso 2021

Filtros lineales óptimos: filtro de Wiener



- ▶ $d[n]$ señal deseada
- ▶ $u[n]$ señal observable conjuntamente estacionaria con $d[n]$

- ▶ **Objetivo:** Diseñar un filtro discreto cuya salida $y[n]$ provea un estimador de una señal deseada $d[n]$ a partir de una señal de entrada correlacionada $u[n]$.
- ▶ El criterio de optimización es la minimización del error cuadrático medio de la estimación, definido como

$$J(\mathbf{w}) = E(|e[n]|^2), \quad \text{con} \quad e[n] = d[n] - y[n].$$

- ▶ En el caso en que el filtro Wiener es FIR con M coeficientes, la dependencia del error cuadrático medio con los coeficientes del filtro es

$$J(\mathbf{w}) = \sigma_d^2 - \mathbf{p}^H \mathbf{w} - \mathbf{w}^H \mathbf{p} + \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (1)$$

Filtro de Wiener

- ▶ Los coeficientes del filtro óptimo que minimizan el error cuadrático medio cumplen que

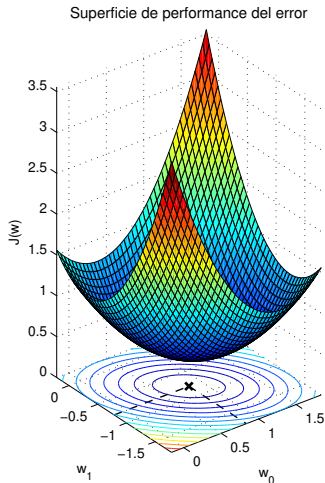
$$\nabla_{\mathbf{w}} J(\mathbf{w}) = 0,$$

- ▶ lo que conduce al sistema $M \times M$ denominado **ecuaciones de Wiener-Hopf**

$$\mathbf{R}\mathbf{w}_o = \mathbf{p} \quad \Leftrightarrow \quad \mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$$

- ▶ Con el filtro funcionando en condiciones óptimas, el error mínimo es

$$\begin{aligned} J(\mathbf{w}) &= J_{min} \\ &= \sigma_d^2 - \mathbf{w}_o^H \mathbf{R} \mathbf{w}_o \\ &= \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1} \mathbf{p} \end{aligned}$$

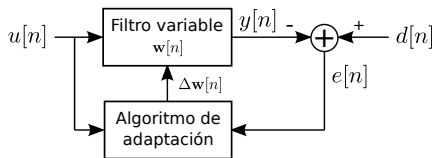


Filtro de Wiener

Observaciones

- ▶ Se necesita conocer la función de autocorrelación de la entrada \mathbf{R} y la correlación cruzada entre la entrada y la señal deseada \mathbf{p} .
- ▶ Se necesita invertir \mathbf{R} , que es Topelitz y simétrica.
 - ▶ Puede ser costoso computacionalmente si M es grande.
 - ▶ Mala estimación si la matriz está mal condicionada.
- ▶ **Hipótesis:** El filtro de Wiener solo puede aplicarse en condiciones de estacionaridad:
 - ▶ $u[n]$ estacionario en sentido amplio (*WSS*).
 - ▶ $u[n]$ y $d[n]$ tienen que ser conjuntamente estacionarios.

Filtros adaptivos



- ▶ $d[n]$ señal deseada
- ▶ $u[n]$ señal observable conjuntamente estacionaria con $d[n]$

- ▶ Los coeficientes del filtro se actualizan con el requerimiento de **reducir el error** cuadrático medio $J[n]$ en cada iteración,

$$\mathbf{w}[n + 1] = \mathbf{w}[n] + \Delta \mathbf{w}[n].$$

- ▶ El **algoritmo de adaptación** define como se realiza la corrección de los coeficientes en cada paso, es decir, define $\Delta \mathbf{w}[n]$.
 - ▶ Para eso, usa la señal de error $e[n]$, ya que permite al filtro medir su desempeño y determinar como deben ser actualizados los coeficientes.
- ▶ Teniendo en cuenta que los coeficientes son variables, la salida del filtro es

$$y[n] = \sum_{k=0}^{M-1} w_k[n] u[n - k] = \mathbf{w}^T[n] \mathbf{u}[n]$$

Filtros adaptivos

Curva de aprendizaje

- ▶ La curva de aprendizaje de un filtro adaptivo es la evolución del error cuadrático medio $J[n]$ con el número de iteración.
- ▶ Indica cuan rápido el filtro adaptivo aprende la solución de las ecuaciones de Wiener-Hopf.

Propiedades deseables de un filtro adaptivo

- ▶ En condiciones estacionarias, el filtro debe producir una sucesión de correcciones $\Delta \mathbf{w}[n]$ de forma de que $\mathbf{w}[n]$ converja al filtro de Wiener,

$$\lim_{n \rightarrow \infty} \mathbf{w}[n] = \mathbf{R}^{-1} \mathbf{p}.$$

- ▶ No debería ser necesario conocer $r_u[k]$ y $p[k]$ para calcular $\Delta \mathbf{w}[n]$. La estimación de la estadística de las señales debería ser parte del algoritmo adaptivo.
- ▶ En el caso no estacionario, el filtro adaptivo debería ser capaz de adaptarse a los cambios en la estadística de las señales y seguir a la solución a medida que evoluciona en el tiempo.

Descenso por gradiente

- ▶ En el método de descenso por gradiente, la regla de adaptación es

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \frac{1}{2}\mu(-\nabla J[n])$$

- ▶ $\mu > 0$: tamaño del paso de adaptación
- ▶ Aplicando el gradiente a $J[n]$ (ecuación 1) se tiene que

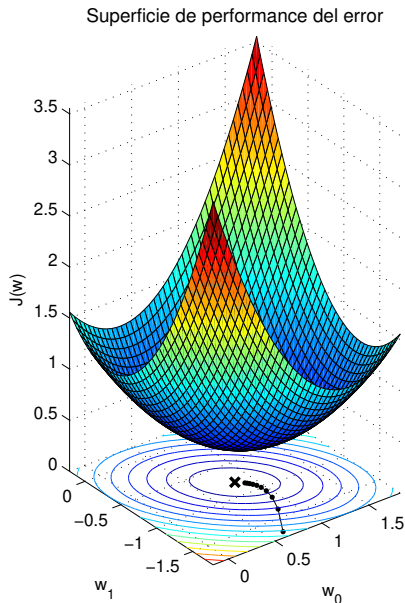
$$\nabla J[n] = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}[n]$$

- ▶ La regla de adaptación queda

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

- ▶ El algoritmo comienza con alguna estimación inicial de \mathbf{w} , por ejemplo,

$$\mathbf{w}[0] = \mathbf{0}.$$



Descenso por gradiente

Estabilidad del algoritmo de descenso por gradiente

- ▶ La evolución del i -ésimo coeficiente del filtro está dada por

$$w_i[n] = w_{io} + \sum_{k=0}^{M-1} a_{ik}(1 - \mu\lambda_k)^n \quad i = 0, 1, \dots, M - 1$$

- ▶ w_{io} es el valor óptimo del i -ésimo coeficiente del filtro.
- ▶ λ_k son los valores propios de \mathbf{R} .
- ▶ a_{ik} son constantes que dependen de las condiciones iniciales y los valores y vectores propios de \mathbf{R} .
- ▶ Cada exponencial en la sumatoria se llama modo natural del filtro.

Convergencia

En el caso estacionario, el algoritmo de descenso por gradiente **converge a la solución de las ecuaciones de Wiener-Hopf** si el tamaño de paso cumple que [los modos naturales se extinguen]

$$0 < \mu < \frac{2}{\lambda_{\max}}, \quad \text{con } \lambda_{\max} \text{ el valor propio máximo de } \mathbf{R}.$$

Descenso por gradiente

Velocidad de convergencia

- ▶ Se define la constante de tiempo τ_k como la cantidad de iteraciones para que el k -ésimo modo caiga $1/e$ de su valor inicial,

$$(1 - \mu\lambda_k)^{\tau_k} = \frac{1}{e} \quad \Rightarrow \quad \tau_k = -\frac{1}{\ln(1 - \mu\lambda_k)}$$

- ▶ En el caso en que μ es pequeño ($\mu\lambda_k \ll 1$), $\tau_k \approx \frac{1}{\mu\lambda_k}$
- ▶ Definiendo la constante de tiempo global τ como la cantidad de iteraciones para que el modo de decaimiento mas lento caiga $1/e$ de su valor inicial,

$$\tau = \max \tau_k \approx \frac{1}{\mu\lambda_{\min}}$$

- ▶ Definiendo α como el paso de adaptación normalizado, $\mu = \alpha \frac{2}{\lambda_{\max}}$, donde $0 < \alpha < 1$, se llega a que

$$\tau \approx \frac{1}{2\alpha} \frac{\lambda_{\max}}{\lambda_{\min}} = \frac{1}{2\alpha} \chi \quad \text{con } \chi \text{ el número de condición de } \mathbf{R}$$

Descenso por gradiente

Algoritmo (descenso por gradiente, máxima pendiente)

$$\begin{cases} \mathbf{w}[0] & = \mathbf{0} \\ \mathbf{w}[n+1] & = \mathbf{w}[n] + \mu (\mathbf{p} - \mathbf{R}\mathbf{w}[n]) \end{cases}$$

Observaciones

- ▶ Hay que conocer \mathbf{R} y \mathbf{p} , igual que en cálculo del filtro de Wiener.
- ▶ Hipótesis:
 - ▶ $u[n]$ estacionario en sentido amplio
 - ▶ $u[n]$ y $d[n]$ conjuntamente estacionarios
- ▶ No se necesita invertir \mathbf{R} .
- ▶ Si el algoritmo converge, lo hace al filtro óptimo (filtro de Wiener).
- ▶ **Condición de convergencia:** El paso de adaptación tiene que cumplir

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

- ▶ La **velocidad de convergencia** es proporcional al número de condición de la matriz de autocorrelación de la entrada $\chi(\mathbf{R})$.

Algoritmo LMS

- ▶ En el método de descenso por gradiente, la regla de adaptación es

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \frac{1}{2}\mu (-\nabla J[n]) \quad \text{con} \quad \nabla J[n] = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}[n]$$

- ▶ En el algoritmo LMS, el gradiente es estimado a partir de las muestras,

$$\hat{\mathbf{w}}[n+1] = \hat{\mathbf{w}}[n] + \frac{1}{2}\mu \left(-\hat{\nabla} J[n]\right) \quad \text{con} \quad \hat{\nabla} J[n] = -2\hat{\mathbf{p}} + 2\hat{\mathbf{R}}\hat{\mathbf{w}}[n]$$

Valores verdaderos

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

$$\mathbf{p}[n] = E(\mathbf{u}[n]d^*[n])$$

Estimadores instantáneos

$$\hat{\mathbf{R}}[n] = \mathbf{u}[n]\mathbf{u}^H[n]$$

$$\hat{\mathbf{p}}[n] = \mathbf{u}[n]d^*[n]$$

- ▶ Usando los estimadores instantáneos, la estimación del gradiente es

$$\hat{\nabla} J[n] = -2\hat{\mathbf{p}} + 2\hat{\mathbf{R}}\hat{\mathbf{w}}[n]$$

$$= -2\mathbf{u}[n]d^*[n] + 2\mathbf{u}[n]\mathbf{u}^H[n]\hat{\mathbf{w}}[n]$$

$$= -2\mathbf{u}[n] (d^*[n] - \mathbf{u}^H[n]\hat{\mathbf{w}}[n])$$

$$= -2\mathbf{u}[n] (d^*[n] - y^*[n])$$

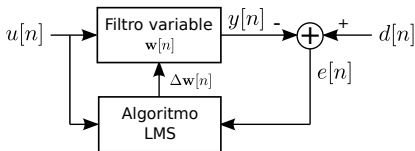
$$\hat{\nabla} J[n] = -2\mathbf{u}[n]e^*[n]$$

Algoritmo LMS

- ▶ La ecuación en recurrencia del algoritmo LMS queda

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

Algoritmo LMS



- ▶ Paso 0: Se establecen condiciones iniciales: $\hat{\mathbf{w}}[0] = \mathbf{0}$.
- ▶ Paso n : Se tiene la estimación de $\hat{\mathbf{w}}[n]$
 1. Se calcula la salida del filtro,

$$y[n] = \hat{\mathbf{w}}^H[n] \mathbf{u}[n]$$

2. Se calcula el error de estimación,

$$e[n] = d[n] - y[n]$$

3. Se adaptan los coeficientes del filtro,

$$\hat{\mathbf{w}}[n + 1] = \hat{\mathbf{w}}[n] + \mu \mathbf{u}[n] e^*[n]$$

Propiedades del algoritmo LMS

Hipótesis de independencia

Los datos $\mathbf{u}[n]$ y los coeficientes del filtro $\hat{\mathbf{w}}[n]$ son estadísticamente independientes.

- ▶ Como $\hat{\mathbf{w}}[n]$ depende de $\mathbf{u}[n-1]$, $\mathbf{u}[n-2]$, \dots , la hipótesis no es cierta.
- ▶ El uso de la hipótesis conduce a propiedades de convergencia que coinciden con los resultados experimentales.

Convergencia

Si los procesos son conjuntamente estacionarios y se cumple la hipótesis de independencia, el algoritmo LMS converge en media si se cumple que

$$0 < \mu < \frac{2}{\lambda_{\max}}, \quad \text{con } \lambda_{\max} \text{ el valor propio máximo de } \mathbf{R}.$$

- ▶ Convergencia en media: $\lim_{n \rightarrow \infty} E(\hat{\mathbf{w}}[n]) = \mathbf{R}^{-1}\mathbf{p}$
- ▶ La cota no establece restricciones en la varianza de $\hat{\mathbf{w}}[n]$.
- ▶ Para establecer la cota hay que conocer \mathbf{R} . En la práctica puede usarse la cota mas restrictiva

$$0 < \mu < \frac{2}{ME(|u[n]|^2)}$$

Propiedades del algoritmo LMS

Error por exceso

- ▶ Con el filtro de Wiener, el error cuadrático medio de estimación al usar los coeficientes óptimos es $J(\mathbf{w}_o) = J_{min}$.
- ▶ Las estimaciones sucesivas de los coeficientes $\hat{\mathbf{w}}[n]$ en el algoritmo LMS sufren de **ruido de gradiente** debido al uso de estimadores instantáneos.
- ▶ Como resultado, en funcionamiento en régimen los coeficientes permanecen fluctuando en torno a los coeficientes óptimos \mathbf{w}_o .
- ▶ El error en régimen con el algoritmo LMS será mayor que el error producido por el filtro óptimo. La diferencia es el **error por exceso**,

$$J_{ex}[n] = J[n] - J_{min} \quad \text{con} \quad J[n] = E(|e[n]|^2).$$

- ▶ En condiciones de convergencia, el error cuadrático medio $J[n]$ converge a la constante

$$J[\infty] = J_{min} + J_{ex}[\infty] = J_{min} \frac{1}{1 - \mu \sum_{k=0}^{M-1} \frac{\lambda_k}{2 - \mu \lambda_k}}$$

Propiedades del algoritmo LMS

Curva de aprendizaje

- ▶ En el algoritmo de descenso por gradiente, dado $\mathbf{w}[0]$, la curva de aprendizaje queda completamente determinada,

$$J[n] = \sigma_d^2 - \mathbf{p}^H \mathbf{w}[n] - \mathbf{w}^H[n] \mathbf{p} + \mathbf{w}^H[n] \mathbf{R} \mathbf{w}[n].$$

- ▶ En el algoritmo LMS, la curva de aprendizaje no es determinística porque la estimación del gradiente se realiza empleando valores muestrales de la entrada y la señal deseada,

$$J[n] = |e^2[n]|.$$

- ▶ La curva de aprendizaje es ruidosa. Una estimación adecuada se logra promediando las curvas de aprendizaje en muchos experimentos usando los mismos parámetros.

Algoritmo NLMS

- ▶ En el algoritmo LMS, la adaptación de los coeficientes está dada por

$$\hat{\mathbf{w}}[n+1] = \hat{\mathbf{w}}[n] + \mu \mathbf{u}[n] e^*[n] \quad \text{con } 0 < \mu < 2/\lambda_{max}$$

- ▶ Aparecen dos dificultades:
 - ▶ Si $\mathbf{u}[n]$ es grande, se produce **amplificación del ruido de gradiente**.
 - ▶ No se conoce \mathbf{R} y por lo tanto tampoco λ_{max} .
- ▶ La condición de convergencia puede sustituirse por la condición mas restrictiva (convergencia en media cuadrática)

$$0 < \mu < \frac{2}{ME(|u[n]|^2)}$$

- ▶ $E(|u[n]|^2)$ puede estimarse a partir de $\mathbf{u}[n]$,

$$E(\widehat{|u[n]|^2}) = \frac{1}{M} \sum_{k=0}^{M-1} |u[n-k]|^2 = \frac{1}{M} \mathbf{u}^H[n] \mathbf{u}[n] = \frac{1}{M} \|\mathbf{u}[n]\|^2.$$

- ▶ lo que conduce a la restricción $0 < \mu < \frac{2}{\|\mathbf{u}[n]\|^2}$.

Algoritmo NLMS

- ▶ Teniendo en cuenta esta restricción, el paso de adaptación del algoritmo LMS puede ser sustituido por el paso variable

$$\mu[n] = \frac{\beta}{\|\mathbf{u}[n]\|^2} \quad \text{con} \quad 0 < \beta < 2.$$

- ▶ Esto conduce al algoritmo NLMS,

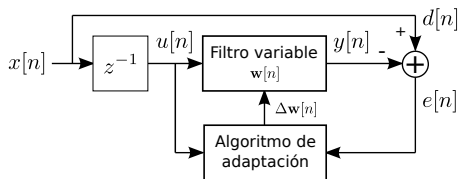
$$\hat{\mathbf{w}}[n+1] = \hat{\mathbf{w}}[n] + \beta \frac{\mathbf{u}[n]}{\|\mathbf{u}[n]\|^2} e^*[n]$$

- ▶ Ocurre un problema similar al de amplificación del ruido de gradiente si $\|\mathbf{u}[n]\|$ es pequeño. Para evitarlo se aplica la siguiente modificación,

$$\hat{\mathbf{w}}[n+1] = \hat{\mathbf{w}}[n] + \beta \frac{\mathbf{u}[n]}{\epsilon + \|\mathbf{u}[n]\|^2} e^*[n],$$

con ϵ algún número real positivo pequeño.

Ejemplo: predicción lineal [Hayes, 1996]



- ▶ La señal deseada $d[n]$ es cierta señal $x[n]$ que se quiere predecir.
- ▶ La entrada al filtro adaptivo es la señal retardada 1 muestra, $u[n] = x[n - 1]$.

Ejemplo

- ▶ La señal a predecir es un proceso AR(2):

$$x[n] = -a_1x[n - 1] - a_2x[n - 2] + v[n] \quad \text{con } v[n] \text{ de potencia } \sigma_v^2$$

- ▶ El filtro adaptivo a diseñar es de primer orden ($M = 2$).

Filtro de Wiener

- ▶ Para encontrar el filtro óptimo, hay que resolver las ecuaciones de Wiener-Hopf de tamaño 2×2 .

$$\mathbf{R} \mathbf{w}_o = \mathbf{p}, \quad \begin{bmatrix} r_u[0] & r_u[1] \\ r_u[1] & r_u[0] \end{bmatrix} \begin{bmatrix} w[0] \\ w[1] \end{bmatrix} = \begin{bmatrix} p[0] \\ p[-1] \end{bmatrix}$$

Ejemplo: predicción lineal

- ▶ Como $d[n] = x[n]$ y $u[n] = x[n - 1]$, la correlación cruzada es

$$\begin{aligned}p[-k] &= E\{u[n - k]d[n]\} \\ &= E\{x[n - 1 - k]x[n]\} \\ &= r_x[k + 1] = r_u[k + 1]\end{aligned}$$

- ▶ Las ecuaciones de Wiener-Hopf quedan en este caso

$$\begin{bmatrix} r_u[0] & r_u[1] \\ r_u[1] & r_u[0] \end{bmatrix} \begin{bmatrix} w[0] \\ w[1] \end{bmatrix} = \begin{bmatrix} r_u[1] \\ r_u[2] \end{bmatrix}$$

- ▶ Teniendo en cuenta que $x[n]$ (y por lo tanto $u[n]$) es un proceso AR(2), usando las ecuaciones de Yule-Walker se tiene que

$$r_u[0] = \frac{1 + a_2}{(1 - a_2)[(1 + a_2)^2 - a_1^2]} \sigma_u^2 = \sigma_u^2$$

$$r_u[1] = \frac{-a_1}{1 + a_2} \sigma_u^2$$

$$r_u[2] = \left(\frac{a_1^2}{1 + a_2} - a_2 \right) \sigma_u^2$$

Ejemplo: predicción lineal

- ▶ Las ecuaciones de Wiener-Hopf quedan

$$\sigma_u^2 \begin{bmatrix} 1 & \frac{-a_1}{1+a_2} \\ \frac{-a_1}{1+a_2} & 1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \sigma_u^2 \begin{bmatrix} \frac{-a_1}{1+a_2} \\ \frac{a_1^2}{1+a_2} - a_2 \end{bmatrix}$$

- ▶ Resolviendo el sistema se llega a que los coeficientes del predictor son,

$$\mathbf{w}_o = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} -a_1 \\ -a_2 \end{bmatrix} \quad y[n] = -a_1 x[n-1] - a_2 x[n-2]$$

- ▶ El error de estimación $e[n]$ del filtro óptimo es

$$e[n] = d[n] - y[n] = x[n] - y[n] = v[n],$$

por lo que el error cuadrático medio mínimo es

$$J_{min} = E(|e^2[n]|) = \sigma_v^2$$

Ejemplo: predicción lineal

Predictor lineal adaptivo con descenso por gradiente

- ▶ La regla de adaptación en el algoritmo de descenso por gradiente es

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

- ▶ La matriz de autocorrelación de la entrada y la correlación cruzada entre la entrada y la señal deseada son,

$$\mathbf{R} = \sigma_u^2 \begin{bmatrix} 1 & \frac{-a_1}{1+a_2} \\ \frac{-a_1}{1+a_2} & 1 \end{bmatrix} \quad \mathbf{p} = \sigma_u^2 \begin{bmatrix} \frac{-a_1}{1+a_2} \\ \frac{a_1^2}{1+a_2} - a_2 \end{bmatrix}$$

- ▶ Hay que decidir el valor del paso de adaptación μ que cumpla las condiciones de convergencia,

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

- ▶ Para eso, hay que calcular los valores propios de la matriz \mathbf{R} .

Ejemplo: predicción lineal

- ▶ En este ejemplo, el cálculo de los valores propios puede hacerse analíticamente,

$$\lambda_{1,2} = \frac{\pm a_1 + a_2 + 1}{(1 - a_2) [(1 + a_2)^2 - a_1^2]}.$$

1. Estudio de la convergencia con el número de condición de \mathbf{R}

- ▶ Se consideran tres casos con distintos parámetros del proceso AR(2) a predecir,

a_1	a_2	λ_{max}	λ_{min}	$\chi(\mathbf{R})$	$\mu_{max} = 2/\lambda_{max}$
0	0.8	2.7778	2.7778	1	0.72
-1.2	0.8	8.3333	1.6667	5	0.24
-1.75	0.8	100	1.4085	71	0.02

- ▶ Se corre el algoritmo de descenso por gradiente para los tres casos usando los mismos parámetros,
 - ▶ $\mathbf{w}[0] = \mathbf{0}$
 - ▶ $\mu = 0.05 \mu_{3max} = 0.001$
 - ▶ 1500 iteraciones [los puntos están graficados cada 20 iteraciones]

Ejemplo: predicción lineal

Parámetros de los modelos AR(2) a predecir

Lugar del plano a_1 - a_2 para condición de estacionaridad asintótica

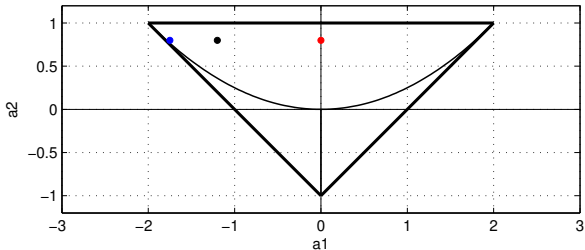
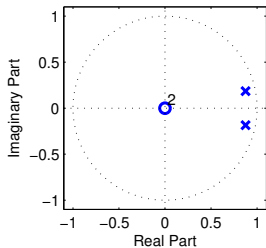
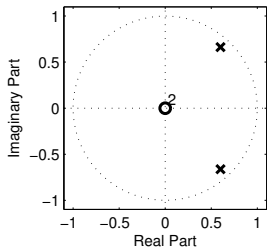
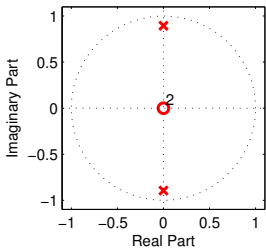


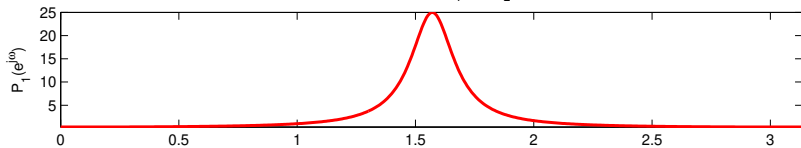
Diagrama de polos y ceros del filtro generador



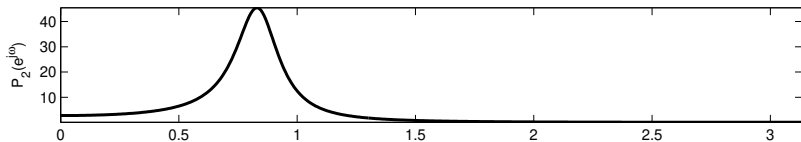
Ejemplo: predicción lineal

Densidad espectral de potencia de los modelos AR(2) a predecir

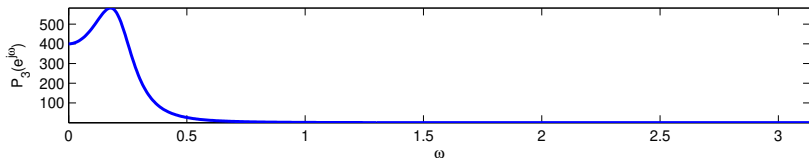
Proceso AR(2) con $a_1 = 0$, $a_2 = 0.8$



Proceso AR(2) con $a_1 = -1.2$, $a_2 = 0.8$



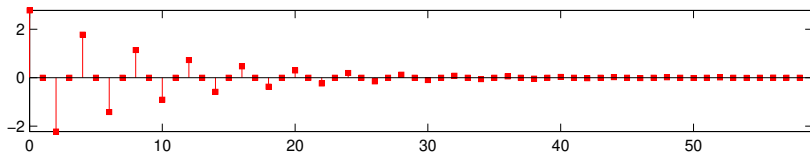
Proceso AR(2) con $a_1 = -1.75$, $a_2 = 0.8$



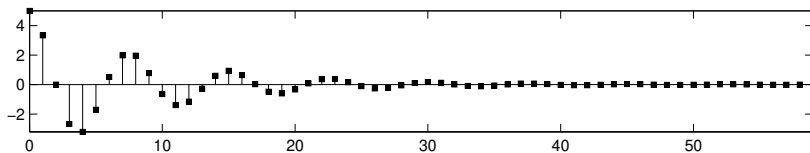
Ejemplo: predicción lineal

Función de autocorrelación de los modelos AR(2) a predecir

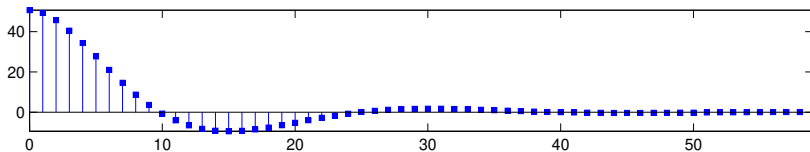
Proceso AR(2) con $a_1 = 0$, $a_2 = 0.8$



Proceso AR(2) con $a_1 = -1.2$, $a_2 = 0.8$



Proceso AR(2) con $a_1 = -1.75$, $a_2 = 0.8$

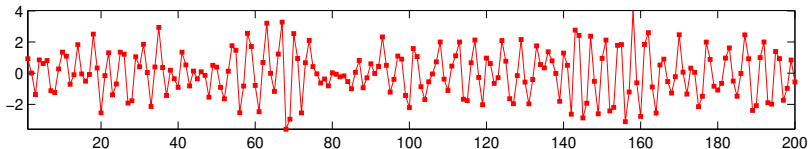


Retardo (muestras)

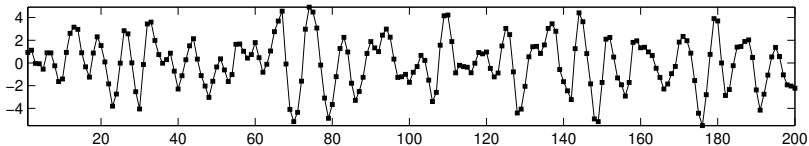
Ejemplo: predicción lineal

Realizaciones de los modelos AR(2) a predecir

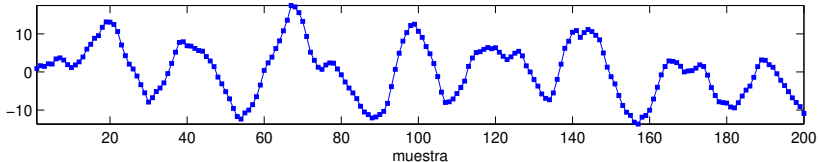
Proceso AR(2) con $a_1 = 0$, $a_2 = 0.8$



Proceso AR(2) con $a_1 = -1.2$, $a_2 = 0.8$



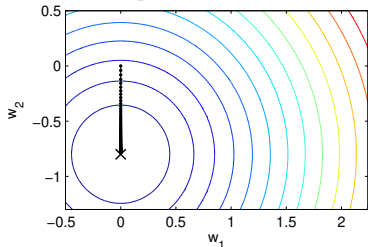
Proceso AR(2) con $a_1 = -1.75$, $a_2 = 0.8$



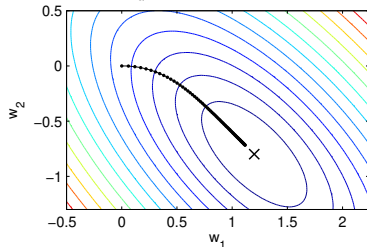
Ejemplo: predicción lineal

Curvas de nivel de la superficie de performance del error y evolución de los coeficientes

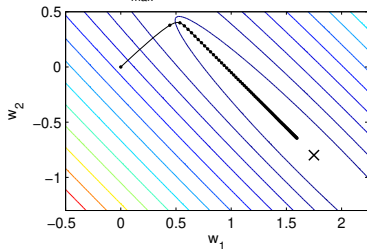
$\chi(R)=1$, $\mu_{\max}=0.72$, $\mu=0.001$, 1500 iteraciones



$\chi(R)=5$, $\mu_{\max}=0.24$, $\mu=0.001$, 1500 iteraciones

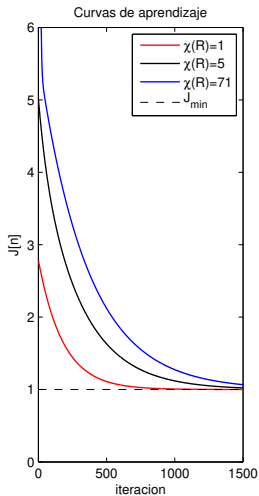
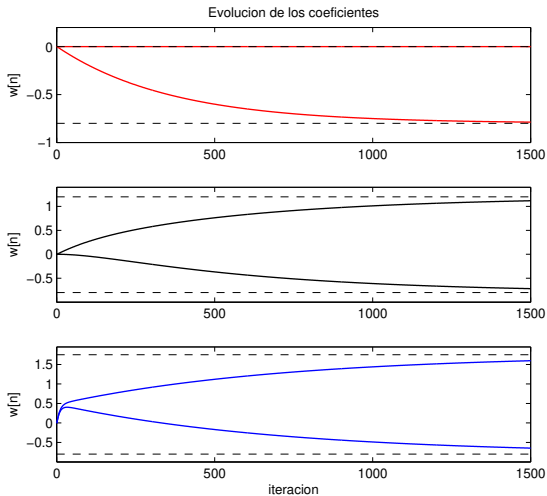


$\chi(R)=71$, $\mu_{\max}=0.02$, $\mu=0.001$, 1500 iteraciones



Ejemplo: predicción lineal

Evolución de los coeficientes y curvas de aprendizaje



Ejemplo: predicción lineal

2. Estudio de la convergencia con el paso de adaptación

- ▶ Se considera ahora solo el segundo proceso AR(2),

a_1	a_2	λ_{max}	λ_{min}	$\chi(\mathbf{R})$	$\mu_{max} = 2/\lambda_{max}$
-1.2	0.8	8.3333	1.6667	5	0.24

- ▶ Se corre el algoritmo de descenso por gradiente para distintos pasos de adaptación,

μ_1	μ_2	μ_3	μ_4
$0.01\mu_{max}$	$0.2\mu_{max}$	$0.9\mu_{max}$	μ_{max}

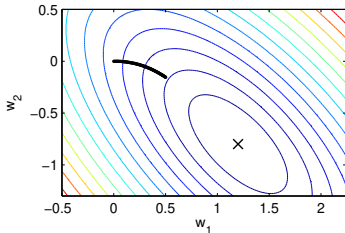
con los parámetros,

- ▶ $\mathbf{w}[0] = \mathbf{0}$
- ▶ 100 iteraciones

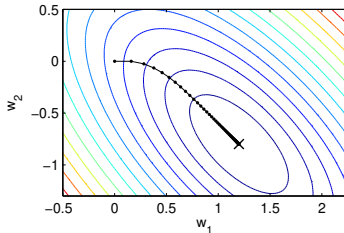
Ejemplo: predicción lineal

Curvas de nivel de la superficie de performance del error y evolución de los coeficientes

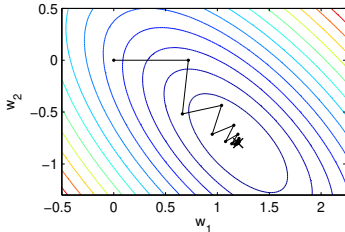
$\chi(R)=5$, $\mu_{\max}=0.24$, $\mu=0.01\mu_{\max}$, 100 iteraciones



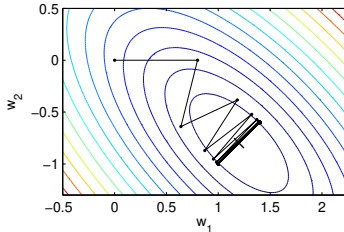
$\chi(R)=5$, $\mu_{\max}=0.24$, $\mu=0.2\mu_{\max}$, 100 iteraciones



$\chi(R)=5$, $\mu_{\max}=0.24$, $\mu=0.9\mu_{\max}$, 100 iteraciones

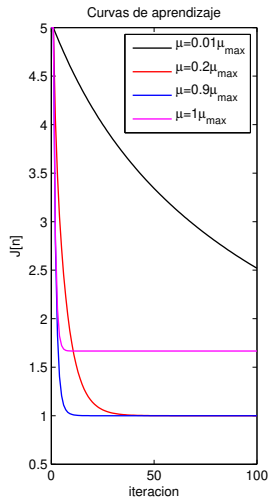
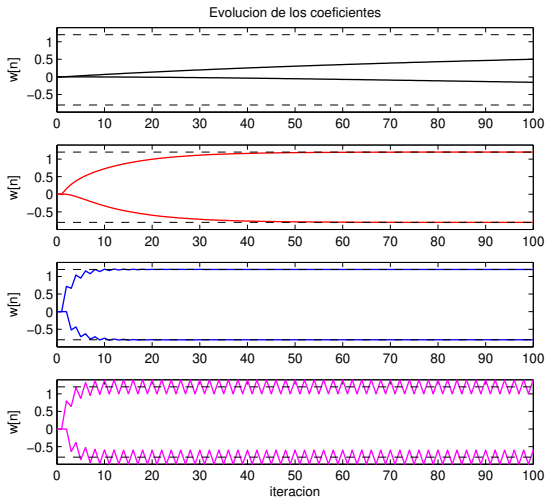


$\chi(R)=5$, $\mu_{\max}=0.24$, $\mu=\mu_{\max}$, 100 iteraciones



Ejemplo: predicción lineal

Evolución de los coeficientes y curvas de aprendizaje



Ejemplo: predicción lineal

Predictor lineal adaptivo con el algoritmo LMS

- ▶ Se emplea ahora como predictor un filtro adaptivo entrenado con el algoritmo LMS.
- ▶ Para entrenar el filtro no es necesario conocer \mathbf{R} ni \mathbf{p} .
- ▶ El procedimiento para realizar las simulaciones es el siguiente:
 - ▶ Se generan muestras del proceso AR(2) $x[n]$ a predecir.
 - ▶ La señal deseada es $d[n] = x[n]$.
 - ▶ La entrada al filtro adaptivo es $u[n] = x[n - 1]$

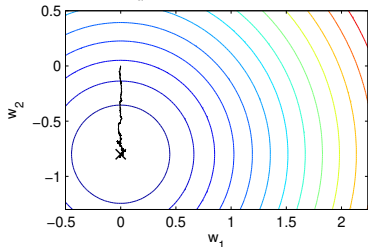
1. Estudio de la convergencia con el número de condición de \mathbf{R}

- ▶ Se consideran los tres mismos casos de antes, donde se varía el número de condición de la matriz de autocorrelación del proceso de entrada.
- ▶ Se corre el algoritmo LMS usando los mismos parámetros,
 - ▶ $\mathbf{w}[0] = \mathbf{0}$
 - ▶ $\mu = 0.05\mu_{3max} = 0.001$
 - ▶ 1500 iteraciones

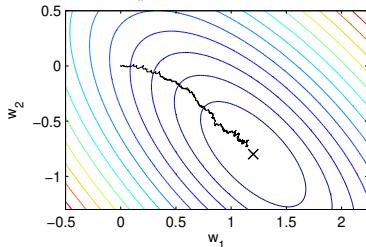
Ejemplo: predicción lineal

Curvas de nivel de la superficie de performance del error y evolución de los coeficientes

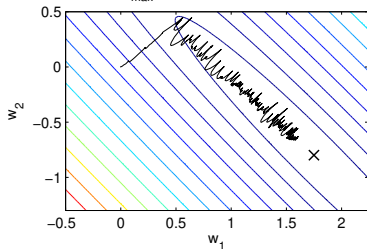
$\chi(R)=1$, $\mu_{\max}=0.72$, $\mu=0.001$, 1500 iteraciones



$\chi(R)=5$, $\mu_{\max}=0.24$, $\mu=0.001$, 1500 iteraciones

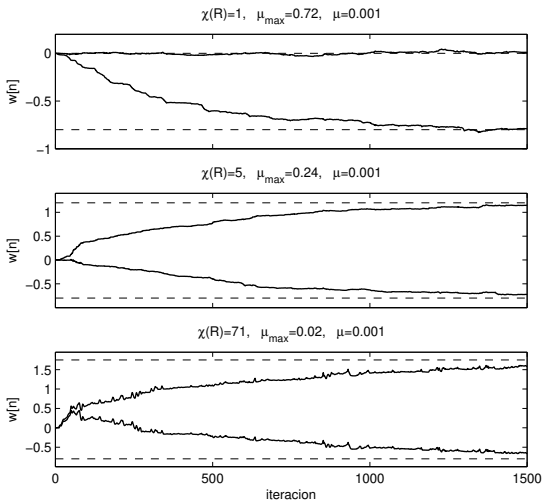


$\chi(R)=71$, $\mu_{\max}=0.02$, $\mu=0.001$, 1500 iteraciones

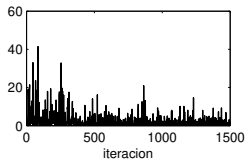
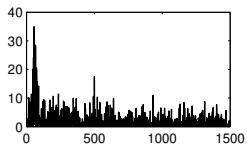
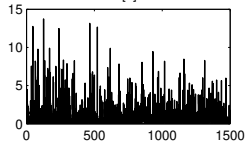


Ejemplo: predicción lineal

Evolución de los coeficientes y error cuadrático medio de estimación



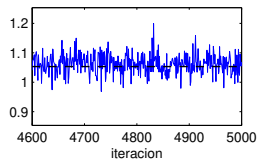
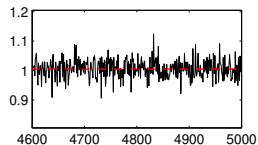
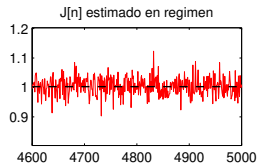
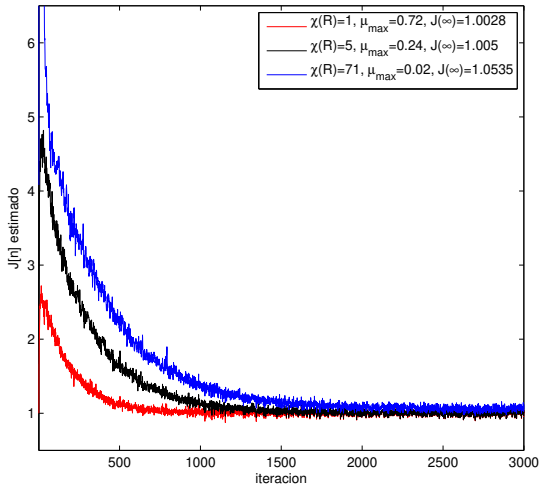
Error cuadrático $e^2[n]$ en una realización



Ejemplo: predicción lineal

Curvas de aprendizaje

Curvas de aprendizaje. Promedio de $e^2[n]$ en 2000 experimentos con $\mu = 0.001$



Ejemplo: predicción lineal

2. Estudio de la convergencia con el paso de adaptación

- ▶ Se considera ahora solo el segundo proceso AR(2),

a_1	a_2	λ_{max}	λ_{min}	$\chi(\mathbf{R})$	$\mu_{max} = 2/\lambda_{max}$
-1.2	0.8	8.3333	1.6667	5	0.24

- ▶ Se corre el algoritmo de descenso por gradiente para distintos pasos de adaptación,

μ_1	μ_2	μ_3	μ_4
$0.01\mu_{max}$	$0.3\mu_{max}$	$0.1\mu_{max}$	$0.2\mu_{max}$

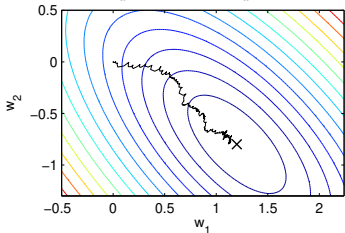
con los parámetros,

- ▶ $\mathbf{w}[0] = \mathbf{0}$
- ▶ 500 iteraciones

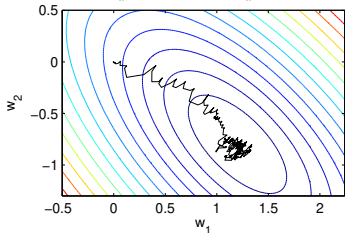
Ejemplo: predicción lineal

Curvas de nivel de la superficie de performance del error y evolución de los coeficientes

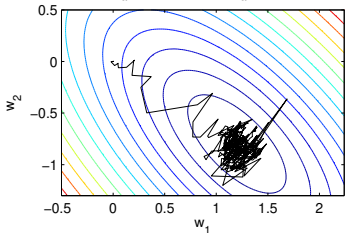
$\chi(R)=5$, $\mu_{\max}=0.72$, $\mu=0.01\mu_{\max}$, 500 iteraciones



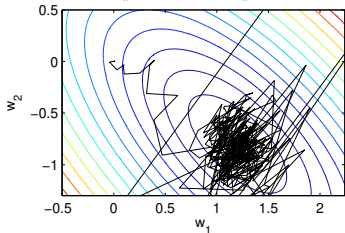
$\chi(R)=5$, $\mu_{\max}=0.72$, $\mu=0.03\mu_{\max}$, 500 iteraciones



$\chi(R)=5$, $\mu_{\max}=0.72$, $\mu=0.1\mu_{\max}$, 500 iteraciones



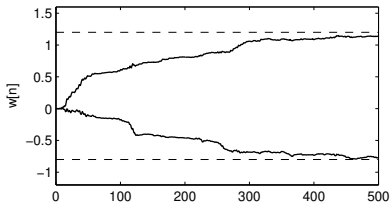
$\chi(R)=5$, $\mu_{\max}=0.72$, $\mu=0.2\mu_{\max}$, 500 iteraciones



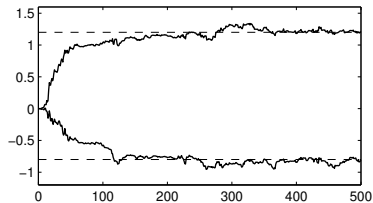
Ejemplo: predicción lineal

Evolución de los coeficientes

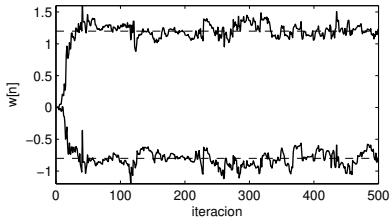
$$\mu = 0.01 \mu_{\max} = 0.0024$$



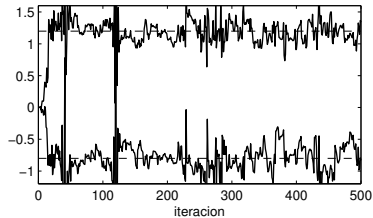
$$\mu = 0.03 \mu_{\max} = 0.0072$$



$$\mu = 0.1 \mu_{\max} = 0.024$$



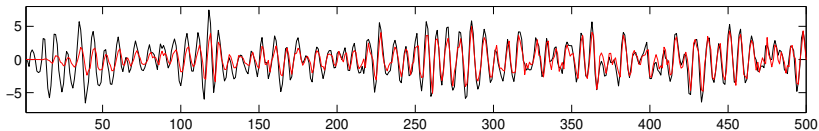
$$\mu = 0.2 \mu_{\max} = 0.048$$



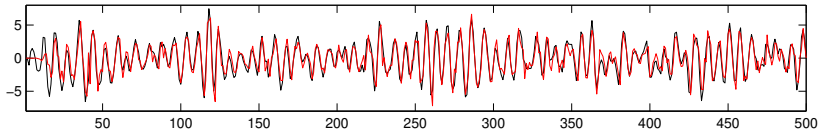
Ejemplo: predicción lineal

Resultado de la predicción del proceso

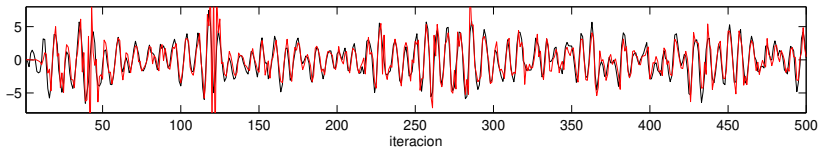
$$\mu = 0.03 \mu_{\max} = 0.0072$$



$$\mu = 0.1 \mu_{\max} = 0.024$$

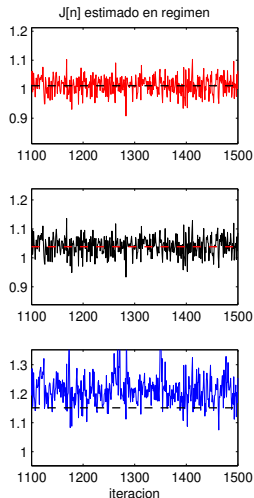
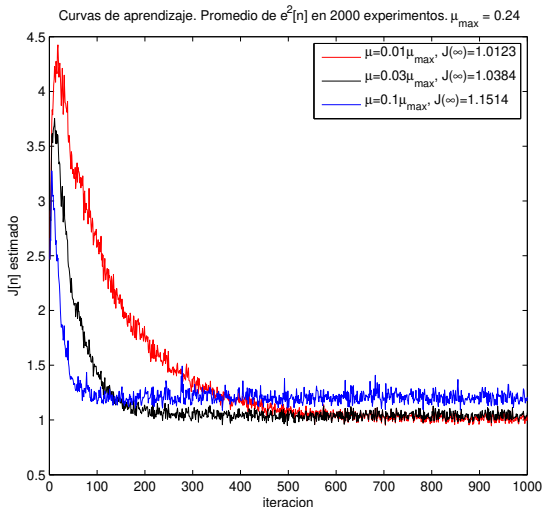


$$\mu = 0.2 \mu_{\max} = 0.048$$



Ejemplo: predicción lineal

Curvas de aprendizaje



Ejemplo: predicción lineal

- ▶ Considerando el caso de procesos con \mathbf{R} de distinto número de condición, para que la comparación de J_{min} tenga sentido, la potencia σ_u^2 de los procesos debe ser igual.
- ▶ La potencia del proceso AR(2) es,

$$r_u[0] = \frac{1 + a_2}{(1 - a_2) [(1 + a_2)^2 - a_1^2]} \sigma_v^2 = \sigma_u^2.$$

- ▶ Fijando σ_u^2 , hay que elegir σ_v^2 como

$$\sigma_v^2 = \frac{(1 - a_2) [(1 + a_2)^2 - a_1^2]}{1 + a_2} \sigma_u^2.$$

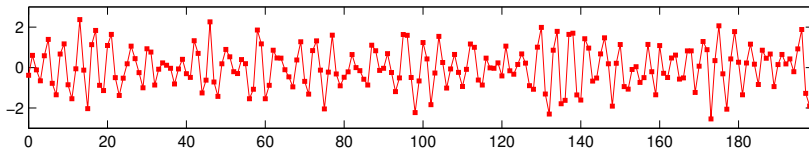
- ▶ En el ejemplo considerado, si $\sigma_u^2 = 1$, se tiene entonces

a_1	a_2	$\chi(\mathbf{R})$	$\mu_{max} = 2/\lambda_{max}$	σ_v^2
0	0.8	1	0.72	0.36
-1.2	0.8	5	0.24	0.2
-1.75	0.8	71	0.02	0.0197

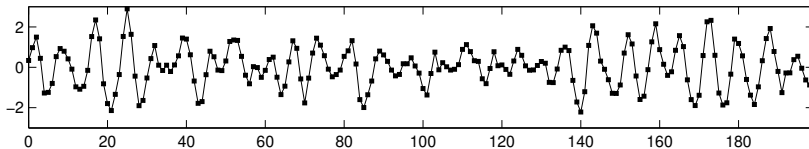
Ejemplo: predicción lineal

Realizaciones de los procesos AR(2) con igual potencia

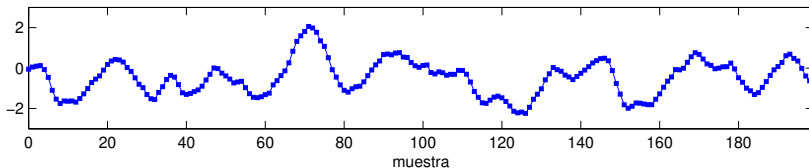
Proceso AR(2) con $a_1 = 0$, $a_2 = 0.8$



Proceso AR(2) con $a_1 = -1.2$, $a_2 = 0.8$

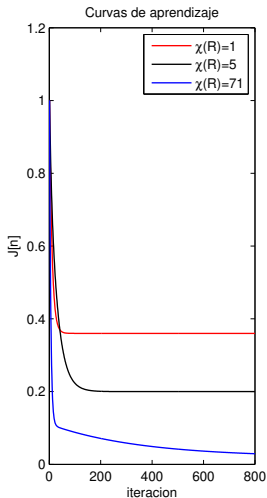
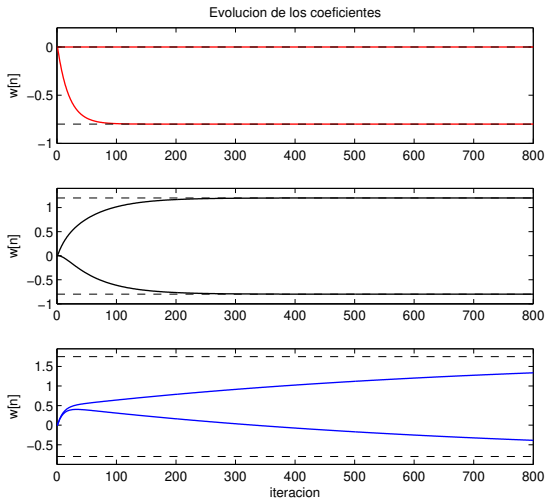


Proceso AR(2) con $a_1 = -1.75$, $a_2 = 0.8$

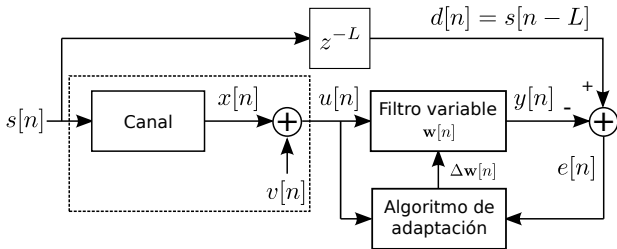


Ejemplo: predicción lineal

Evolución de los coeficientes y curvas de aprendizaje usando descenso por gradiente



Ejemplo: ecualización de canal [Haykin, 1995]



Fase de entrenamiento

- ▶ $s[n]$ ruido blanco de media nula y potencia σ_s^2 .
- ▶ $v[n]$ ruido blanco introducido por el canal de media nula y potencia σ_v^2 independiente de $s[n]$.

Ejemplo

- ▶ El canal se modela como el siguiente filtro FIR de tres coeficientes

$$h[n] = \begin{cases} \frac{1}{2} [1 + \cos(\frac{2\pi}{W}(n-2))] , & \text{si } n = 1, 2, 3 \\ 0, & \text{en otro caso} \end{cases}$$

W controla la cantidad de distorsión del canal.

- ▶ $s[n]$ es un proceso Bernoulli que toma el valor -1 o 1 con igual probabilidad.
- ▶ El filtro adaptivo tiene $M = 11$ coeficientes.

Ejemplo: ecualización de canal

Filtro de Wiener

1. Matriz de autocorrelación de la entrada

- ▶ La entrada al filtro es $u[n] = x[n] + v[n]$, con $x[n]$ y $v[n]$ procesos independientes. Por lo tanto,

$$r_u[k] = r_x[k] + r_v[k].$$

- ▶ $v[n]$ es ruido blanco de potencia σ_v^2 , así que $r_v[k] = \sigma_v^2 \delta[k]$.
- ▶ Teniendo en cuenta que $x[n]$ es un proceso filtrado, se cumple que,

$$r_x[k] = r_s[k] * h[k] * h[-k],$$

- ▶ $s[n]$ es un proceso blanco de potencia σ_s^2 , así que $r_s[k] = \sigma_s^2 \delta[k]$. Además $\sigma_s^2 = E(s^2[n]) = 1$. Entonces,

$$r_u[k] = h[k] * h[-k] + \sigma_v^2 \delta[k].$$

Ejemplo: ecualización de canal

- Como $h[n]$ tiene solo tres coeficientes no nulos, se tiene que,

$$r_u[k] = \begin{cases} h^2[1] + h^2[2] + h^2[3] + \sigma_v^2, & k = 0 \\ h[1]h[2] + h[2]h[3], & k = 1 \\ h[1]h[3], & k = 2 \\ 0, & k \geq 3 \end{cases}$$

$$\mathbf{R} = \begin{bmatrix} r_u[0] & r_u[1] & r_u[2] & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r_u[1] & r_u[0] & r_u[1] & r_u[2] & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r_u[2] & r_u[1] & r_u[0] & r_u[1] & r_u[2] & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_u[2] & r_u[1] & r_u[0] & r_u[1] & r_u[2] & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_u[2] & r_u[1] & r_u[0] & r_u[1] & r_u[2] & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_u[2] & r_u[1] & r_u[0] & r_u[1] & r_u[2] & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_u[2] & r_u[1] & r_u[0] & r_u[1] & r_u[2] & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_u[2] & r_u[1] & r_u[0] & r_u[1] & r_u[2] & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_u[2] & r_u[1] & r_u[0] & r_u[1] & r_u[2] \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_u[2] & r_u[1] & r_u[0] & r_u[1] \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_u[2] & r_u[1] & r_u[0] \end{bmatrix}$$

Ejemplo: ecualización de canal

2. Correlación cruzada entre la entrada y la señal deseada

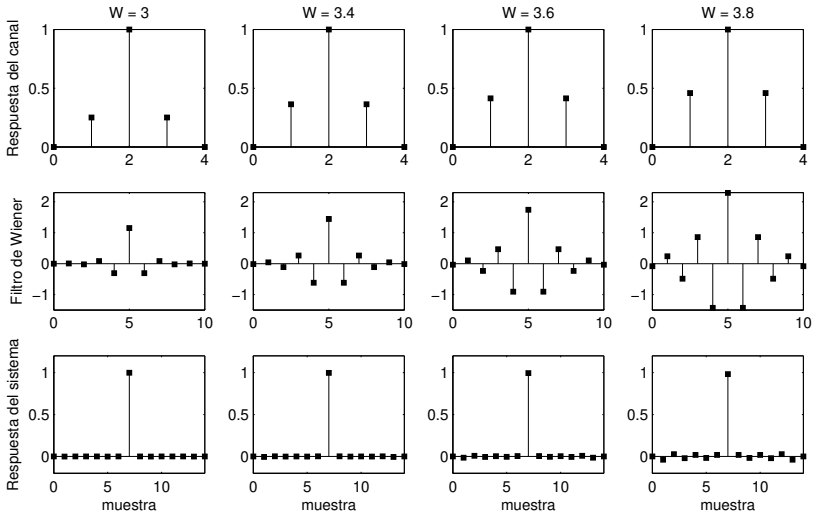
$$\begin{aligned} p[-k] &= E \{u[n-k]d[n]\} \\ &= E \{(x[n-k] + v[n-k])s[n-L]\} \\ &= E \{x[n-k]s[n-L]\} \\ &= E \{x[n]s[n+k-L]\} \\ &= E \left\{ \left(\sum_{l=-\infty}^{\infty} h[l]s[n-l] \right) s[n+k-L] \right\} \\ &= \sum_{l=-\infty}^{\infty} h[l]E \{s[n-l]s[n+k-L]\} \\ &= \sum_{l=-\infty}^{\infty} h[l]r_s[-l-k+L] \\ &= \sum_{l=-\infty}^{\infty} h[l]\delta[-l-k+L] \\ &= h[-k+L] \end{aligned}$$

Por ejemplo, si $L = 7$,

$$\mathbf{p} = \begin{bmatrix} h[7] \\ h[6] \\ h[5] \\ h[4] \\ h[3] \\ h[2] \\ h[1] \\ h[0] \\ h[-1] \\ h[-2] \\ h[-3] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ h[3] \\ h[2] \\ h[1] \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

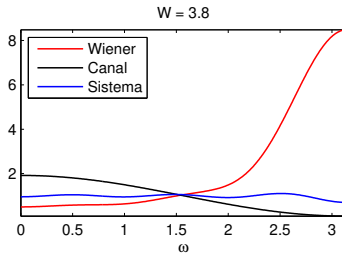
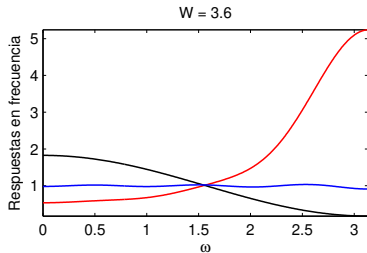
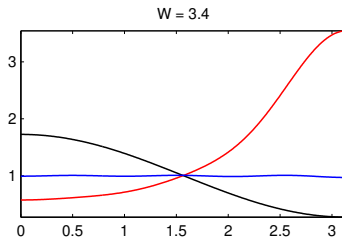
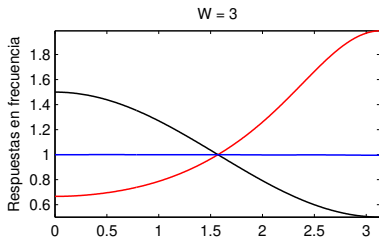
Ejemplo: ecualización de canal

Respuesta del canal al variar W , respuesta del filtro de Wiener y respuesta global del sistema



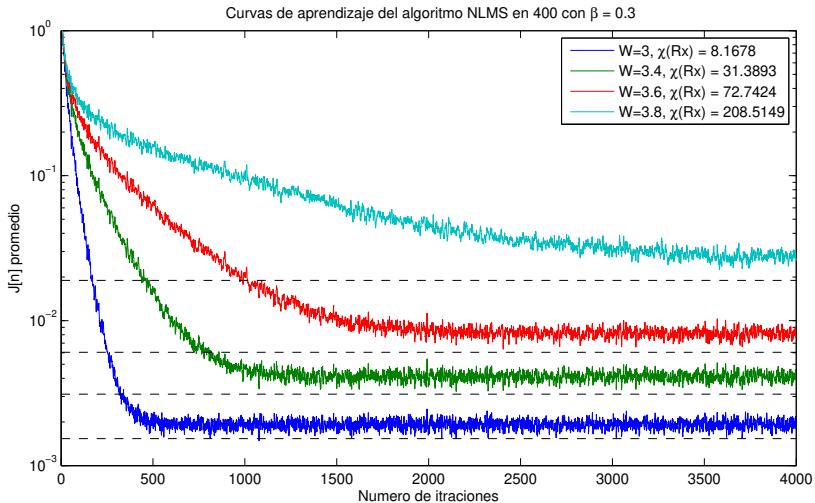
Ejemplo: ecualización de canal

Respuestas en frecuencia al variar W



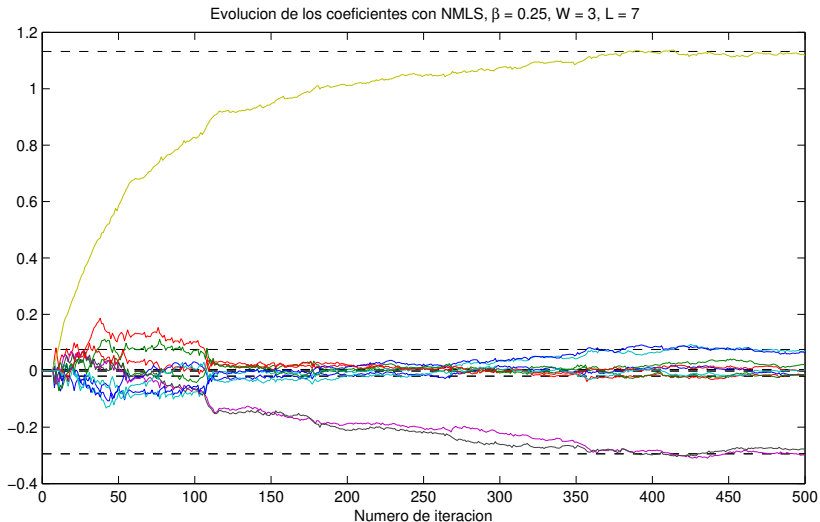
Ejemplo: ecualización de canal

Curvas de aprendizaje del algoritmo NMLS al variar W



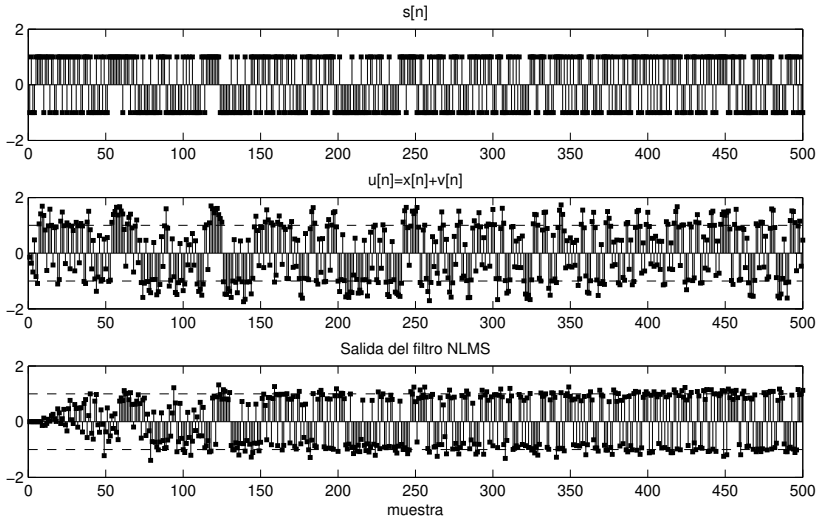
Ejemplo: ecualización de canal

Evolución de los coeficientes del filtro NLMS en un experimento.



Ejemplo: ecualización de canal

Realizaciones de los procesos en un experimento



Observación

- ▶ La respuesta del canal en este ejemplo es un filtro FIR simétrico.
- ▶ Puede demostrarse que el filtro FIR óptimo para contrarrestar la distorsión del canal debe ser simétrico.
- ▶ El sistema global (cascada del canal y el filtro adaptivo) tiene un retardo de grupo igual a la suma de los retardos de grupo de ambos filtros.
- ▶ En este ejemplo se tiene que,

$$\tau_{g,\text{canal}} = 2 \text{ muestras}$$

$$\tau_{g,\text{wiener}} = 5 \text{ muestras}$$

$$\tau_{g,\text{global}} = 7 \text{ muestras}$$

- ▶ Para obtener el filtro óptimo, el retardo de la señal deseada debe coincidir con el retardo de grupo global.

