# Filtros Adaptativos (parte 1)

## Procesamiento de señales

.UBA fiuba
FACULTAD DE INGENIERÍA

Ing. Débora P. Copa ● Dra. Cecilia G. Galarza
IIBM LPSC

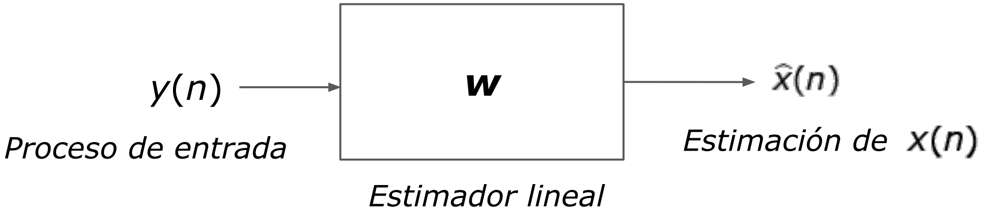
# Repaso de filtro de Wiener FIR

# Ejercicio: filtro de Wiener

# Ejercicio 1

**Sean:**

- *x(n)* e *y(n)* dos procesos conjuntamente ESA
- ***w*** un estimador lineal FIR




**Problema de optimización:**

*Encontrar los coeficientes **w**=*$\{w_0, w_1, \ldots, w_{M-1}\}$ *que minimizan la potencia del error:*

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

donde $e(n) = x(n) - \hat{x}(n)$ es el error de estimación

# Ejercicio: filtro de Wiener

Estimador del filtro de Wiener FIR de largo M

$$\hat{x}(n) = \sum_{i=0}^{M-1} w_i^* \, y(n-i)$$

$R_{xy}(i) = E[x(n)y(n\text{-}i)^*]$ Función de cross-correlation entre y,x

$R_{yx}(\text{-}i) = E[y(n\text{-}i)x(n)^*] = R_{xy}(i)*$ Función de cross-correlation entre x,y

$R_y(i) = E[y(n)y(n\text{-}i)^*]$ Función de auto-correlation de y

# Ejercicio: filtro de Wiener

Estimador del filtro de Wiener FIR de largo M

$$\hat{x}(n) = \sum_{i=0}^{M-1} w_i^* \, y(n-i)$$

Expresión vectorial

$$\hat{x}(n) = \underbrace{\begin{bmatrix} w_0^* & w_1^* & \cdots & w_{M-1}^* \end{bmatrix}}_{\mathbf{w}^H} \underbrace{\begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(n-M+1) \end{bmatrix}}_{\mathbf{y}(n)}$$

# Ejercicio: filtro de Wiener

Proceso estimado (notación vectorial)

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

Error de estimación

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

Potencia del error

$$J(\mathbf{w}) = E[|e(n)|^2] = E[\left(x(n) - \mathbf{w}^H \mathbf{y}(n)\right) \left(x(n)^* - \mathbf{y}(n)^H \mathbf{w}\right)] =$$

$$= \sigma_x^2 - \mathbf{w}^H \mathbf{R}_{yx} - \mathbf{R}_{yx}^H \mathbf{w} + \mathbf{w}^H \mathbf{R}_y \mathbf{w}$$

# Ejercicio: filtro de Wiener

Cálculos auxiliares

$$E[\mathbf{y}(n)x(n)^*] =$$

$$E\left\{\begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(n-M+1) \end{bmatrix} x(n)^*\right\} = \begin{bmatrix} E[y(n)x(n)^*] \\ E[y(n-1)x(n)^*] \\ \vdots \\ E[y(n-M+1)x(n)^*] \end{bmatrix} = \underbrace{\begin{bmatrix} R_{yx}(0) \\ R_{yx}(-1) \\ \vdots \\ R_{yx}(-M+1) \end{bmatrix}}_{\mathbf{R}_{yx} \quad (\mathrm{Mx1})}$$

# Ejercicio: filtro de Wiener

Cálculos auxiliares

$$E[x(n)\mathbf{y}(n)^H] =$$

$$E\left\{x(n)\left[y(n)^* \quad y(n-1)^* \quad \ldots \quad y(n-M+1)^*\right]\right\} =$$

$$\left[E[x(n)y(n)^*] \quad E[x(n)y(n-1)^*] \quad \ldots \quad E[x(n)y(n-M+1)^*]\right] =$$

$$= \underbrace{\left[R_{xy}(0) \quad R_{xy}(1) \quad \ldots \quad R_{xy}(M-1)\right]}_{\mathbf{R}_{xy} \quad (1\mathrm{x}M)}$$

# Ejercicio: filtro de Wiener

Cálculos auxiliares

$$E[\mathbf{y}(n)\mathbf{y}(n)^H] = \underbrace{E\left\{\begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(n-M+1) \end{bmatrix} \begin{bmatrix} y(n)^* & y(n-1)^* & \dots & y(n-M+1)^* \end{bmatrix}\right\}}_{\mathbf{R}_y \quad (\text{MxM})}$$

# Ejercicio: filtro de Wiener

Potencia del error $J(\mathbf{w}) = \sigma_x^2 - \mathbf{w}^H\mathbf{R}_{yx} - \mathbf{R}_{yx}^H\mathbf{w} + \mathbf{w}^H\mathbf{R}_y\mathbf{w}$

$\mathbf{R}_{xy} = \mathbf{R}_{yx}^H$

Buscamos el $\mathbf{w}_o$ (óptimo) que minimiza $J(\mathbf{w})$:

$J(\mathbf{w}_o) = Jmin \Rightarrow \nabla J(\mathbf{w}_o) = 0$ (1)

Gradiente de $J$: $\nabla J(\mathbf{w}) = 2\mathbf{R}_y\mathbf{w} - 2\mathbf{R}_{yx}$ (2)

Juntando (1) y (2):

$$2\mathbf{R}_y\mathbf{w}_o - 2\mathbf{R}_{yx} = 0$$

$$\mathbf{R}_y\mathbf{w}_o = \mathbf{R}_{yx} \Rightarrow$$

Coeficientes óptimos del filtro de Wiener FIR

$$\boxed{\mathbf{w}_o = \mathbf{R}_y^{-1}\mathbf{R}_{yx}}$$

# Ejercicio: filtro de Wiener

$$J(\mathbf{w}) = \sigma_x^2 - \mathbf{w}^H\mathbf{R}_{yx} - \mathbf{R}_{yx}^H\mathbf{w} + \mathbf{w}^H\mathbf{R}_y\mathbf{w}$$

$$\mathbf{w}_o = \mathbf{R}_y^{-1}\mathbf{R}_{yx}$$

$$\mathbf{R}_{yx} = \mathbf{R}_y\mathbf{w}_o$$

$$J_{min} = \sigma_x^2 - \mathbf{R}_{yx}^H\mathbf{R}_y^{-1}\mathbf{R}_{yx}$$

$$J_{min} = \sigma_x^2 - \mathbf{w}_o^H\mathbf{R}_y\mathbf{w}_o$$

**Mínima potencia del error**
(expresiones alternativas)

# Ejercicio: filtro de Wiener

Resumen

# Ejercicio: filtro de Wiener

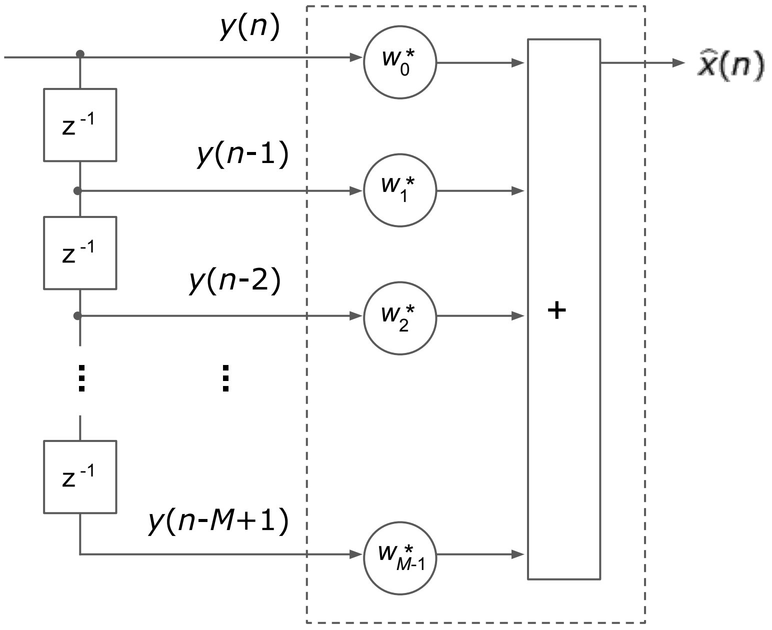
**Señal estimada**

$$\hat{x}(n) = \sum_{i=0}^{M-1} w_i^* \, y(n-i)$$

**Forma vectorial**

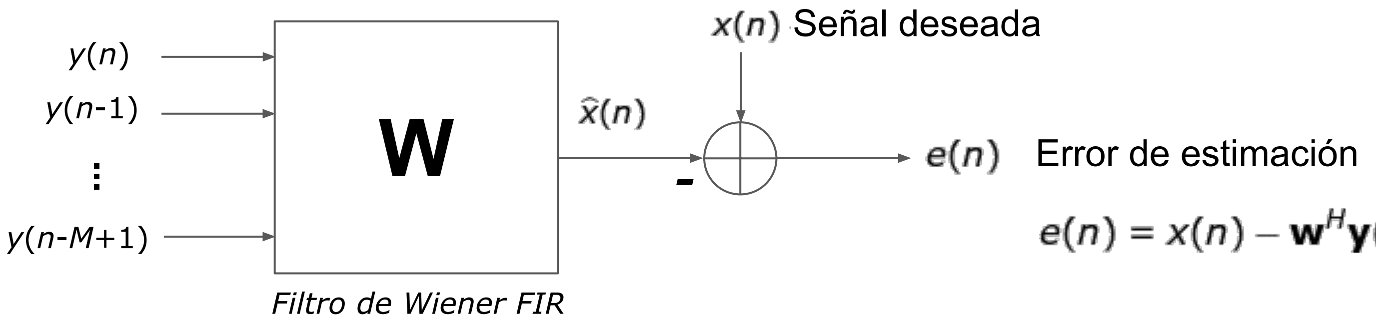
$$\hat{x}(n) = \mathbf{w}^H \mathbf{y}(n)$$

$$\mathbf{y}(n) = \begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(n-M+1) \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{bmatrix}$$



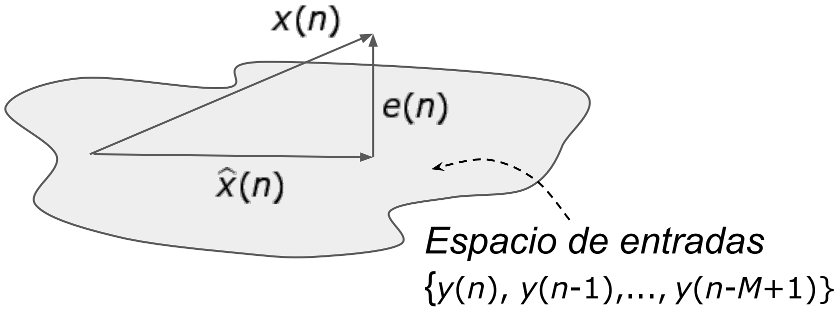

*Filtro de Wiener FIR*

# Ejercicio: filtro de Wiener




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

## Interpretación geométrica




## FIltro de Wiener óptimo

$$\mathbf{w}_o = \mathbf{R}_y^{-1} \mathbf{R}_{yx}$$

## Principio de ortogonalidad

$$E[y(n-i)e(n)^*)]=0$$

# Repaso
# Steepest descent

# Steepest descent

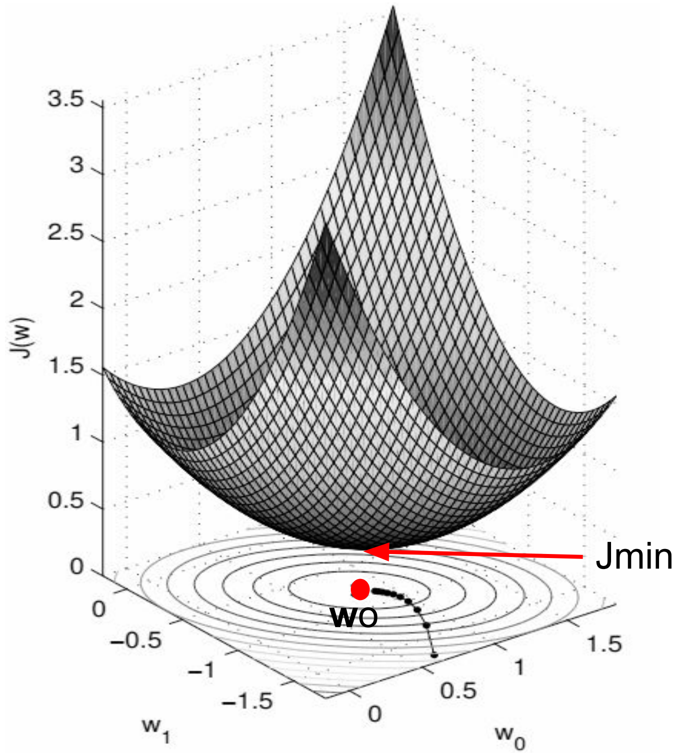
*Supone conocimiento de* $\boldsymbol{R_y}$ *y* $\boldsymbol{R_{yx}}$

$$\mathbf{w}_o = \mathbf{R}_y^{-1}\mathbf{R}_{yx}$$

*Función costo teórica  (potencia del error)*

$$J(\mathbf{w}) = \sigma_x^2 - \mathbf{w}^H\mathbf{R}_{yx} - \mathbf{R}_{yx}^H\mathbf{w} + \mathbf{w}^H\mathbf{R}_y\mathbf{w}$$

***J(w) es una superficie cuadrática convexa. Tiene un mínimo global.***



# Steepest descent

*Supone conocimiento de* $\boldsymbol{R_y}$ *y* $\boldsymbol{R_{yx}}$

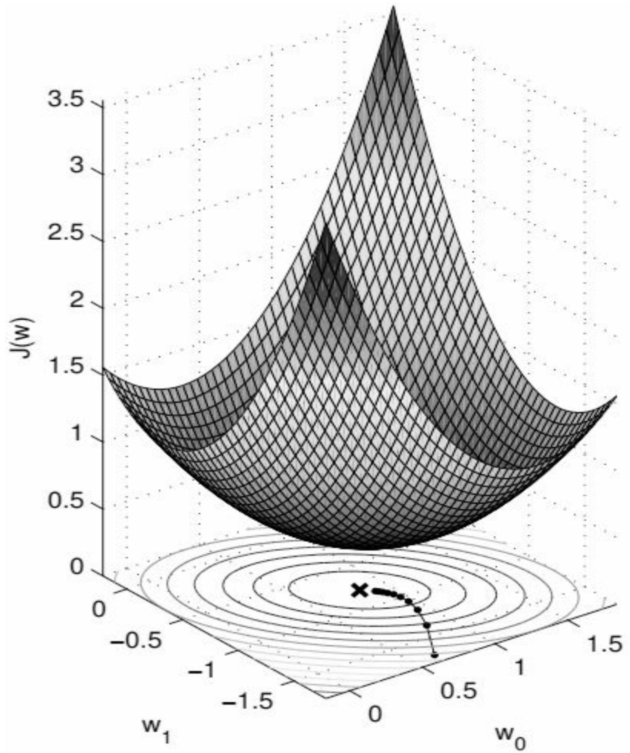
$$\mathbf{w}_{n+1} = \mathbf{w}_n - \tfrac{1}{2}\mu \nabla J(\mathbf{w}_n) \quad ; \quad \mathbf{w}_{inicial}$$

*Gradiente determinístico*

$$\nabla J(\mathbf{w}_n) = -2\mathbf{R}_{yx} + 2\mathbf{R}_y\,\mathbf{w}_n$$

*Curva de aprendizaje*

$$\widehat{J}(n) = \frac{1}{m}\sum_{k=0}^{m-1} |e(n)|^2 = \frac{1}{m}\sum_{k=0}^{m-1} |x(n) - \widehat{x}(n)|^2$$



# Ejercicio 1
LMS

# Ejercicio 1

Sea un proceso $y(n) \in$ ESA la entrada de un filtro de coeficientes $\{w_0, w_1, ..., w_{M-1}\}$ y $x(n)$ una señal deseada conjuntamente estacionaria con $y(n)$. Se busca estimar los coeficientes del filtro según el criterio MMSE.

Dado que la función costo $J(\mathbf{w})$ es una función cuadrática, escalar y continuamente diferenciable, los coeficientes del filtro se pueden obtener recursivamente mediante el método *steepest descent* de acuerdo a $\mathbf{w}_{n+1} = \mathbf{w}_n - \frac{1}{2}\mu\nabla J(\mathbf{w}_n)$, donde $\mu$ es un parámetro que permite ajustar el paso del algoritmo y $\nabla J(\mathbf{w}_n)$ es el gradiente de la función costo $J = E[\,|e(n)|^2]$. Conocidas la matriz de autocorrelación $\mathbf{R}_y$ de $y(n)$ y el vector $\mathbf{R}_{yx}$ de correlación cruzada entre $y(n)$ y $x(n)$, puede demostrarse que el gradiente resulta:

$$\nabla J(\mathbf{w}_n) = -2\mathbf{R}_{yx} + 2\mathbf{R}_y\mathbf{w}_n \qquad (1)$$

# Ejercicio 1

(a) Para implementar de forma práctica el algoritmo de Steepest-descent, uno de los métodos propuestos es el algoritmo adaptativo LMS (Least Mean Square), el cual supone las siguientes aproximaciones: $\widehat{\mathbf{R}}_y(n) = \mathbf{y}(n)\mathbf{y}^H(n)$ y $\widehat{\mathbf{R}}_{yx}(n) = \mathbf{y}(n)x^*(n)$. En base a esta aproximación y al resultado de la Ecuación 1, demuestre que la ecuación recursiva para la estimación de los coeficientes del filtro LMS se puede expresar como:

$$\widehat{\mathbf{w}}_{n+1} = \widehat{\mathbf{w}}_n + \mu\,\mathbf{y}(n)e^*(n),$$

# Ejercicio 1

a) $$\mathbf{w}_{n+1} = \mathbf{w}_n - \frac{1}{2}\mu \nabla J(\mathbf{w}_n)$$ *Steepest descent*

$$\nabla J(\mathbf{w}_n) = -2(\mathbf{R}_{yx} - \mathbf{R}_y \mathbf{w}_n)$$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu(\mathbf{R}_{yx} - \mathbf{R}_y \mathbf{w}_n)$$

*Aproximación (LMS)*
$$\begin{cases} \widehat{\mathbf{R}}_y(n) = \mathbf{y}(n)\mathbf{y}(n)^H \\ \widehat{\mathbf{R}}_{yx}(n) = \mathbf{y}(n)x(n)^* \end{cases}$$

$$\widehat{\mathbf{w}}_{n+1} = \widehat{\mathbf{w}}_n + \mu(\mathbf{y}(n)x(n)^* - \mathbf{y}(n)\mathbf{y}(n)^H \widehat{\mathbf{w}}_n)$$

$$\widehat{\mathbf{w}}_{n+1} = \widehat{\mathbf{w}}_n + \mu \mathbf{y}(n) \underbrace{\left(x(n)^* - \mathbf{y}(n)^H \widehat{\mathbf{w}}_n\right)}_{e(n)^*}$$

$$\boxed{\widehat{\mathbf{w}}_{n+1} = \widehat{\mathbf{w}}_n + \mu \mathbf{y}(n) e(n)^*}$$

# Ejercicio 1:

a) ***Algoritmo LMS***

- Definir condiciones iniciales: **w**(0)
- Paso 0: estimar **w**(1) a partir de **w**(0)
- Paso n: estimar **w**(n+1) a partir de **w**(n):

1. Se calcula la salida del filtro, $\widehat{x}(n) = \mathbf{w}_n^H \mathbf{y}(n)$
2. Se calcula el error de estimación, $e(n) = x(n) - \widehat{x}(n)$
3. Se adaptan los coeficientes del filtro, $\widehat{\mathbf{w}}_{n+1} = \widehat{\mathbf{w}}_n + \mu \mathbf{y}(n) e(n)^*$

# Ejercicio 1

(b) Se requieren estimar los coeficientes $\mathbf{w}$ de un filtro LMS cuya entrada es una secuencia $y(n)$, de modo tal que su salida se ajuste a una señal deseada definida como $x(n) = s(n) + v(n)$, donde $v(n)$ es ruido blanco gaussiando de media nula y varianza $\sigma_v^2 = 0{,}1$. Para generar las señales $y(n)$ y $s(n)$, considere que éstas pueden obtenerse como:

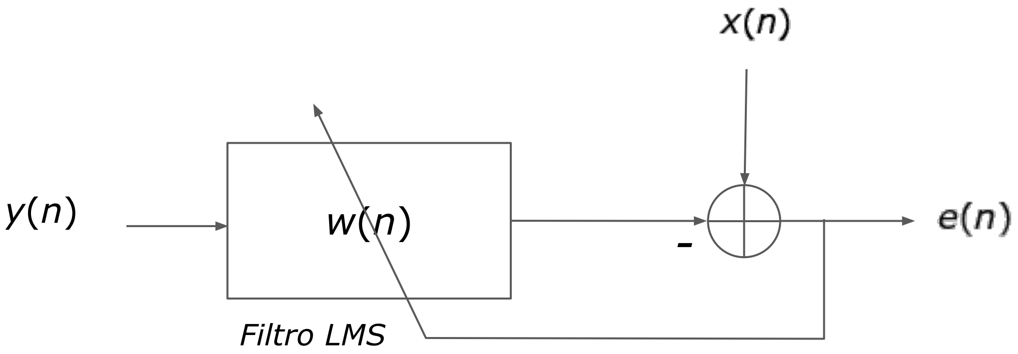
$$y(n) = f(n) + 0{,}5\, f(n-1) + 0{,}25\, f(n-2)$$

$$s(n) = f(n) + 1{,}2 f(n-1) + 0{,}6\, f(n-2) + 0{,}3\, f(n-3)$$

donde $f(n) \sim N(0,1)$ es de largo $L = 2000$.

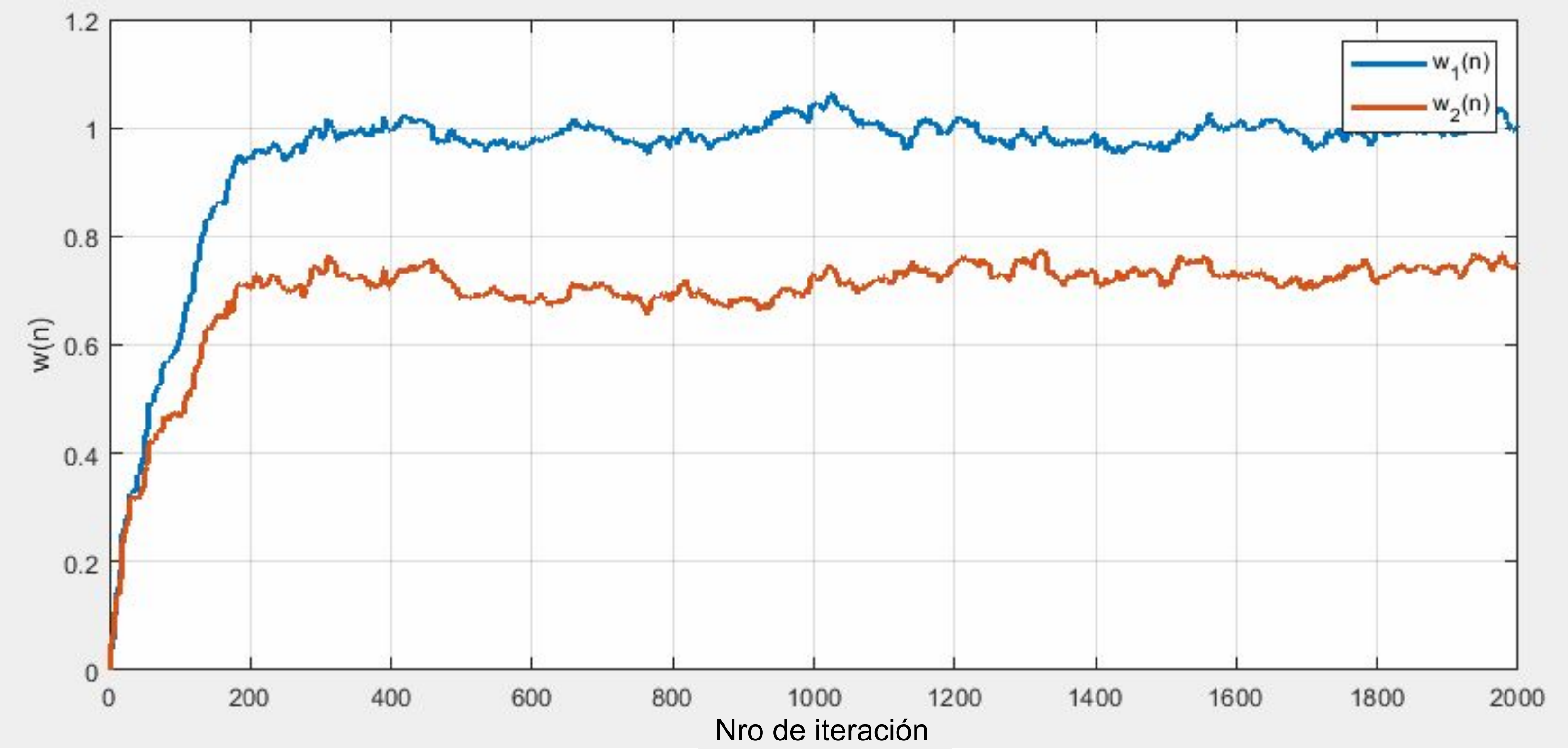
Implemente el filtro LMS y obtenga los coeficientes $\widehat{\mathbf{w}}$ para un largo $M = 2$ y $\mu = 0{,}01$. Grafique los coeficientes $\widehat{\mathbf{w}}_n$ y la curva de aprendizaje $\widehat{J}(n) = \frac{1}{m}\sum_{k=0}^{m-1} |x(n) - \widehat{x}(n)|^2$ (con $m \geq 200$ realizaciones y en escala logarítmica) en función de las iteraciones.

# Ejercicio 1



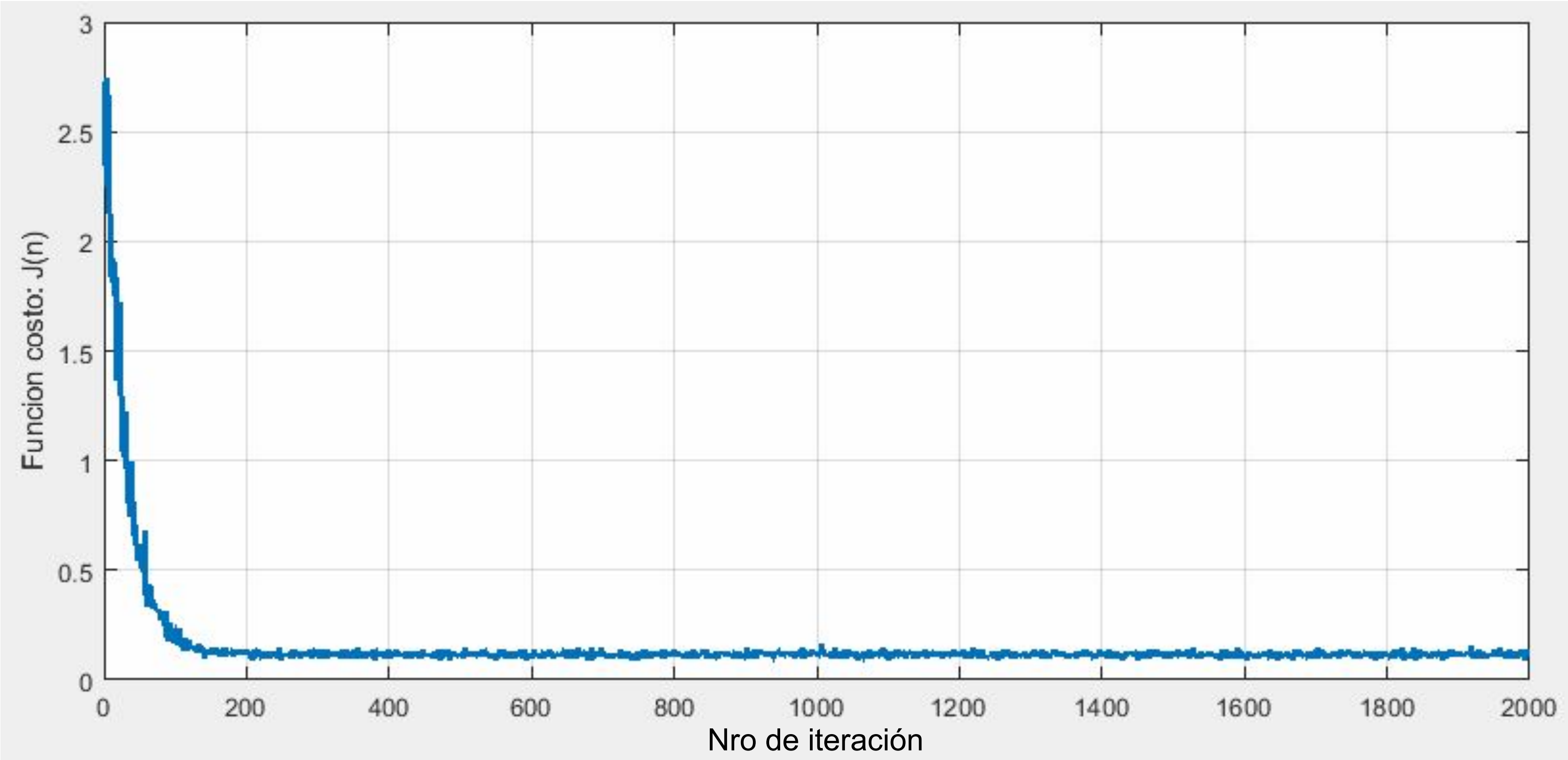
# Ejercicio 1

b)



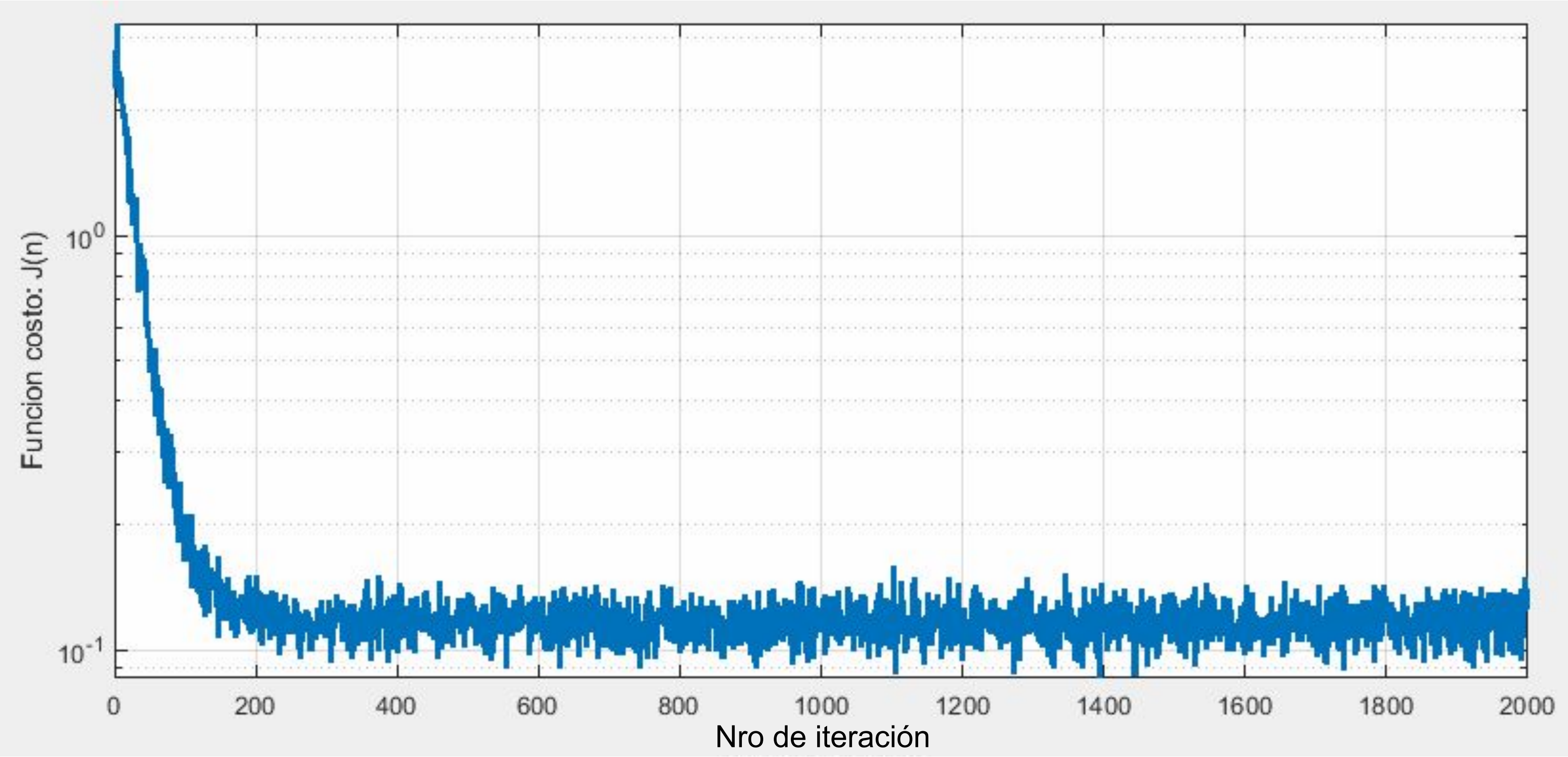
# Ejercicio 1

b)



# Ejercicio 1

b)



# Ejercicio 1

(c) Para una única realización, grafique en un plano $(w_0, w_1)$ las curvas de nivel de $J(w_0, w_1)$ (para generarla va a requerir estimar $\mathbf{R}_y$ y $\mathbf{R}_{yx}$) y la trayectoria de los pesos (resultantes de las iteraciones) sobre el plano. Considere para esto las siguientes condiciones iniciales:

i) $\mathbf{w}_0 = [-3{,}19 \;\; 4{,}47]^T$

ii) $\mathbf{w}_0 = [4{,}84 \;\; 4{,}52]^T$

iii) $\mathbf{w}_0 = [1{,}65 \;\; 8{,}99]^T$

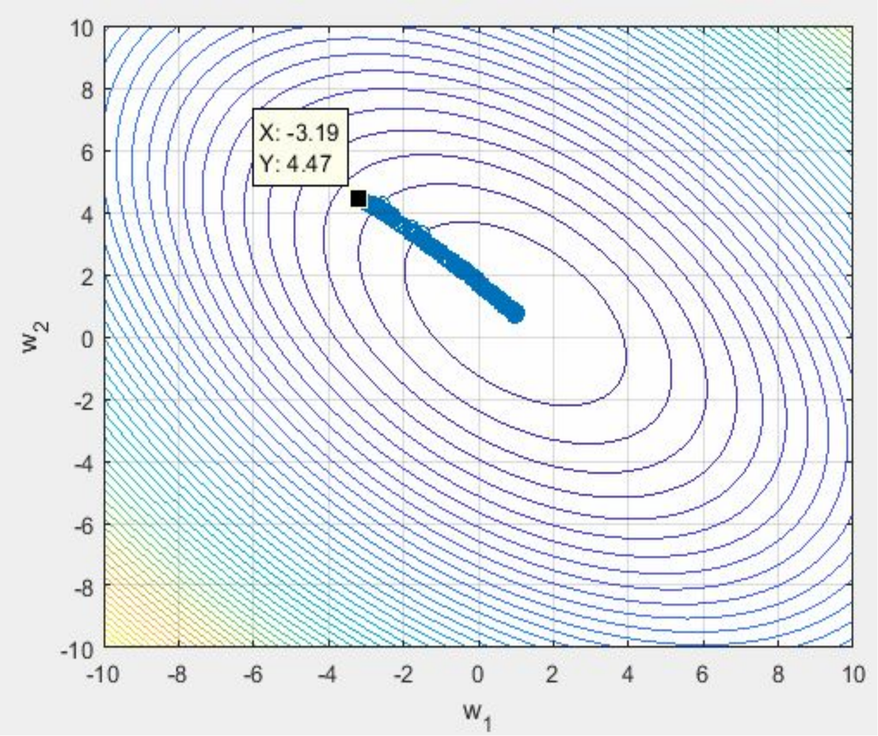
(d) Repita el punto anterior pero para $\mu = 0,12$.

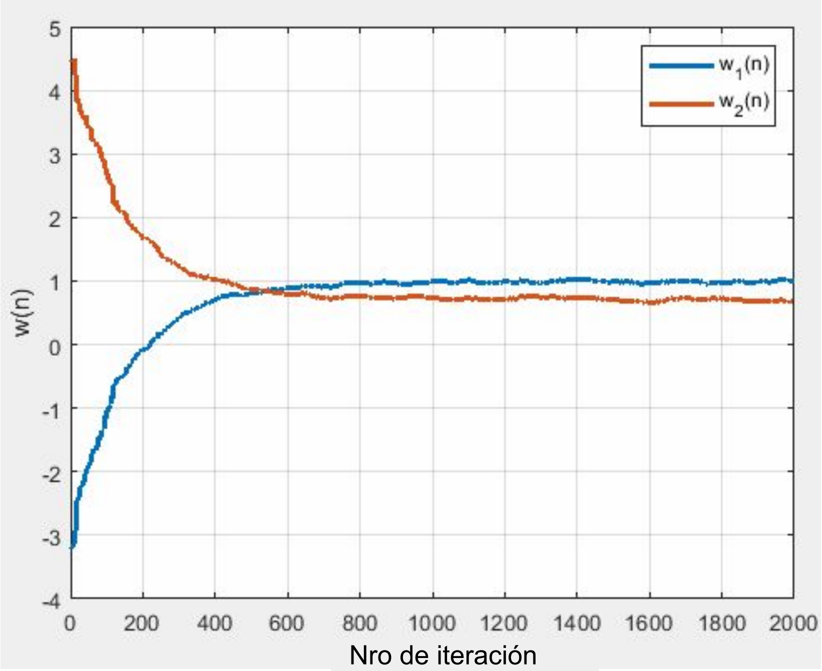
Recordar:

$$J(\mathbf{w}) = \sigma_x^2 - \mathbf{w}^H \mathbf{R}_{yx} - \mathbf{R}_{yx}^H \mathbf{w} + \mathbf{w}^H \mathbf{R}_y \mathbf{w}$$

# Ejercicio 1

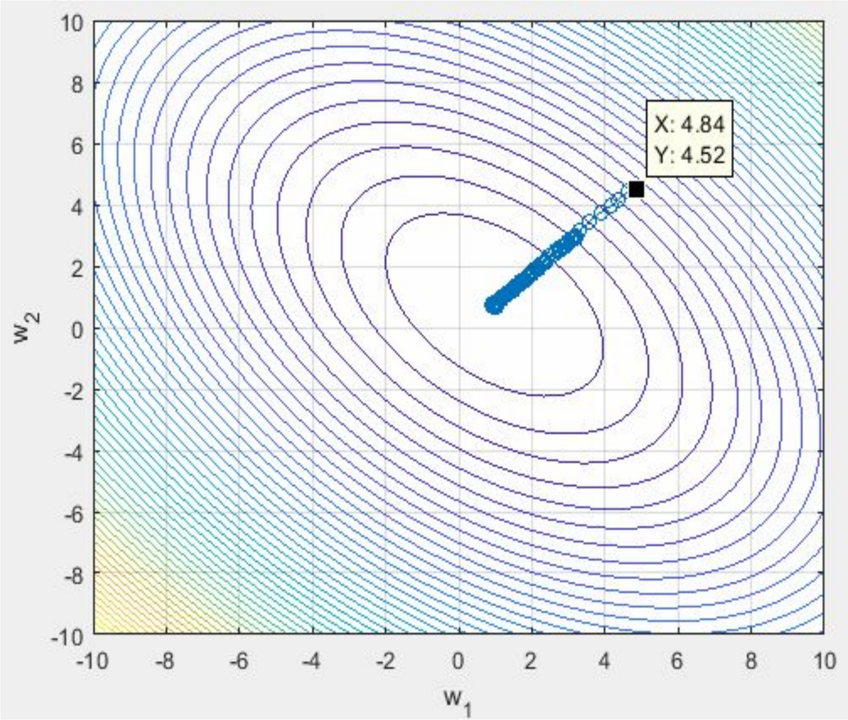
c)

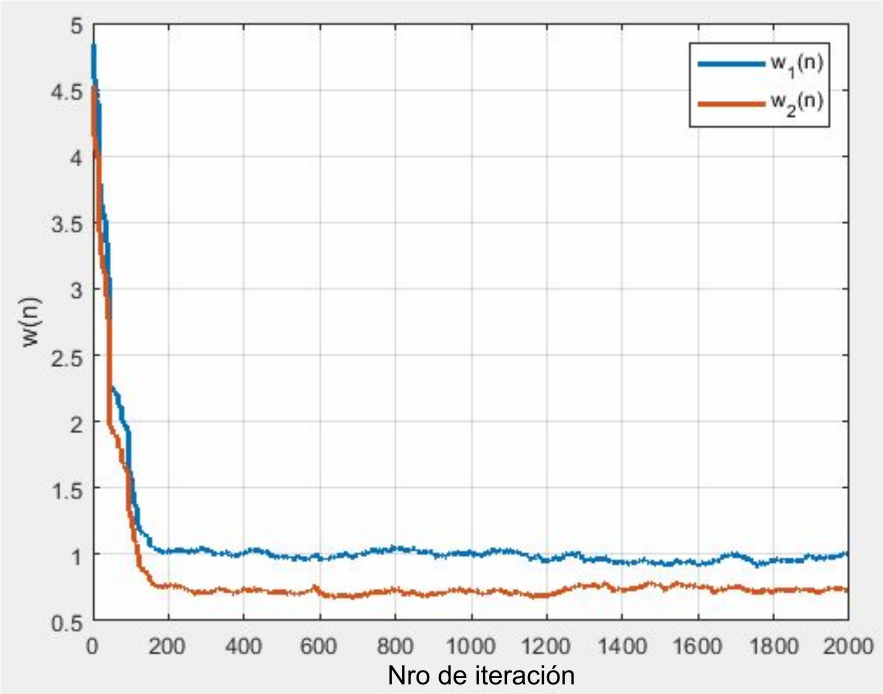


# Ejercicio 1

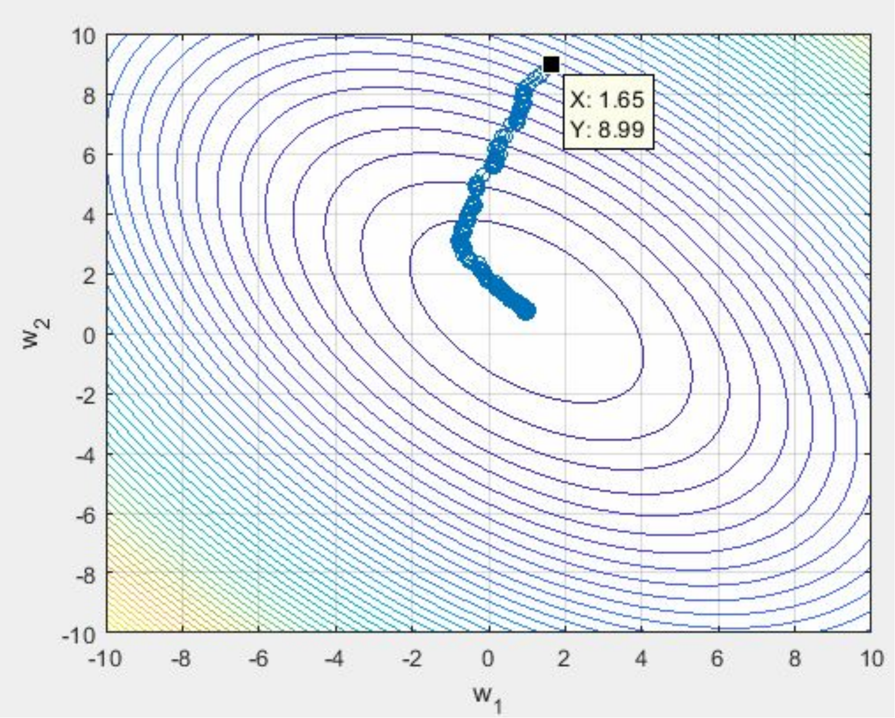
c)

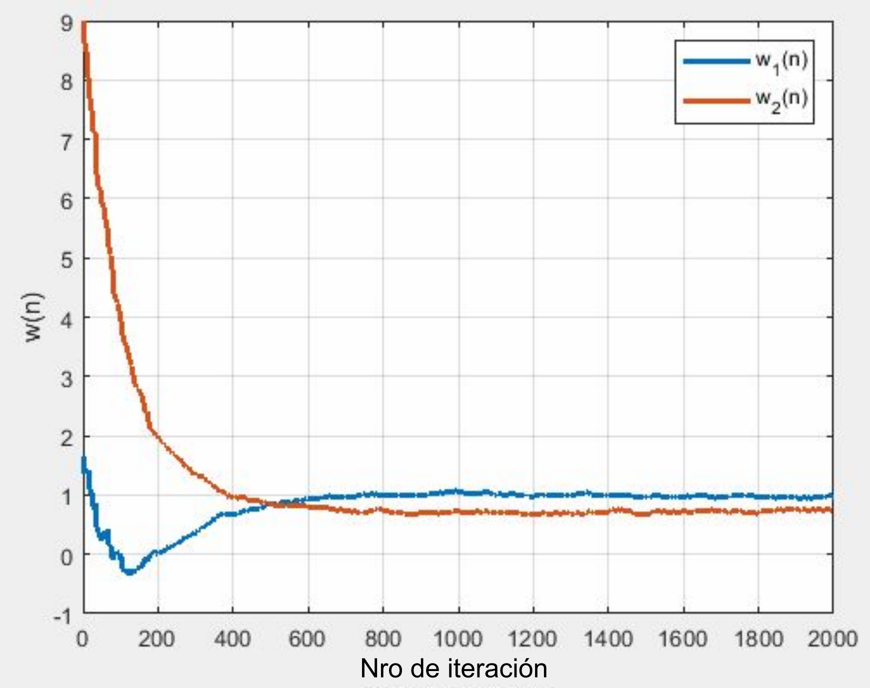


# Ejercicio 1

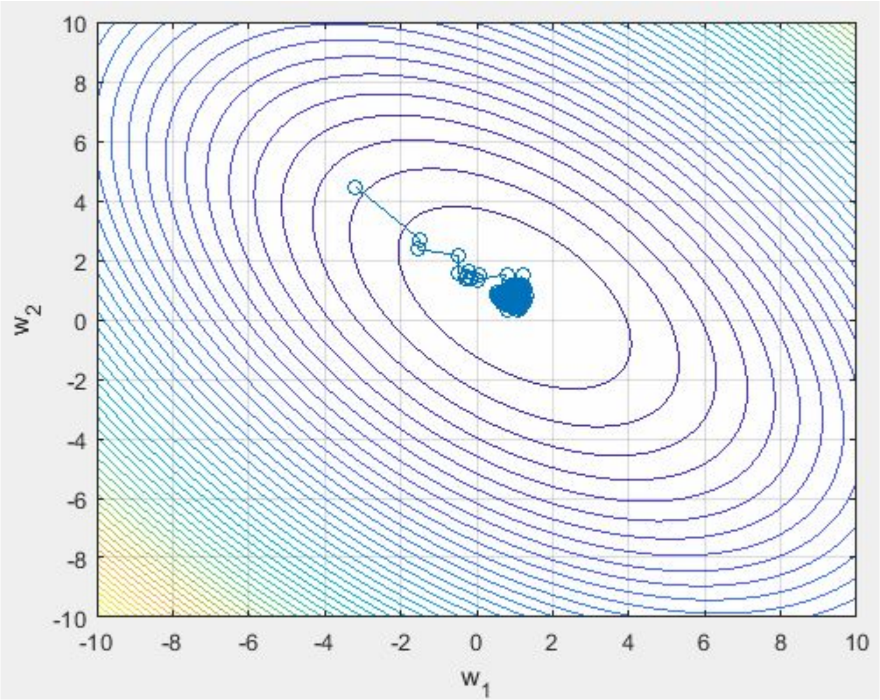
c)

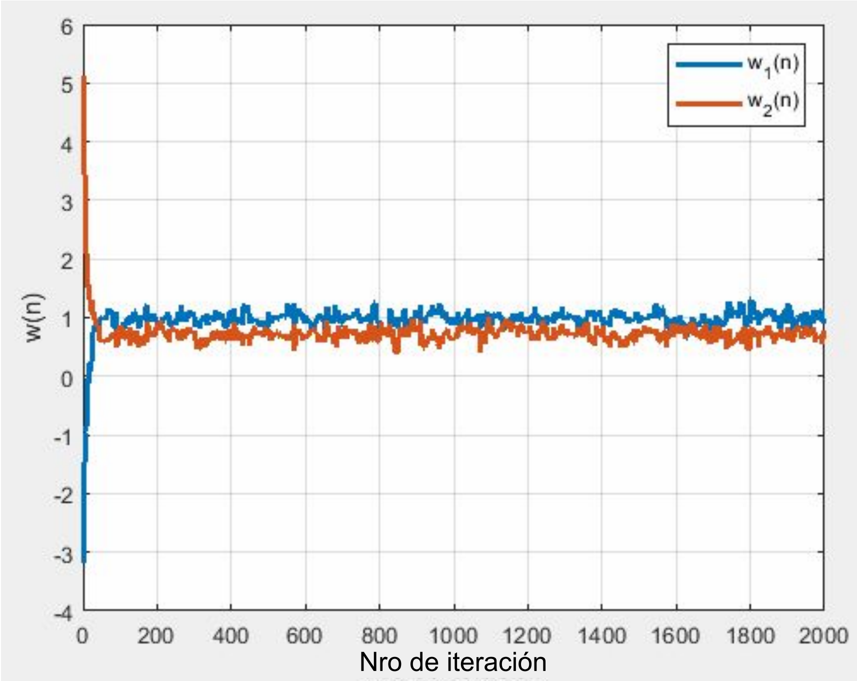


# Ejercicio 1

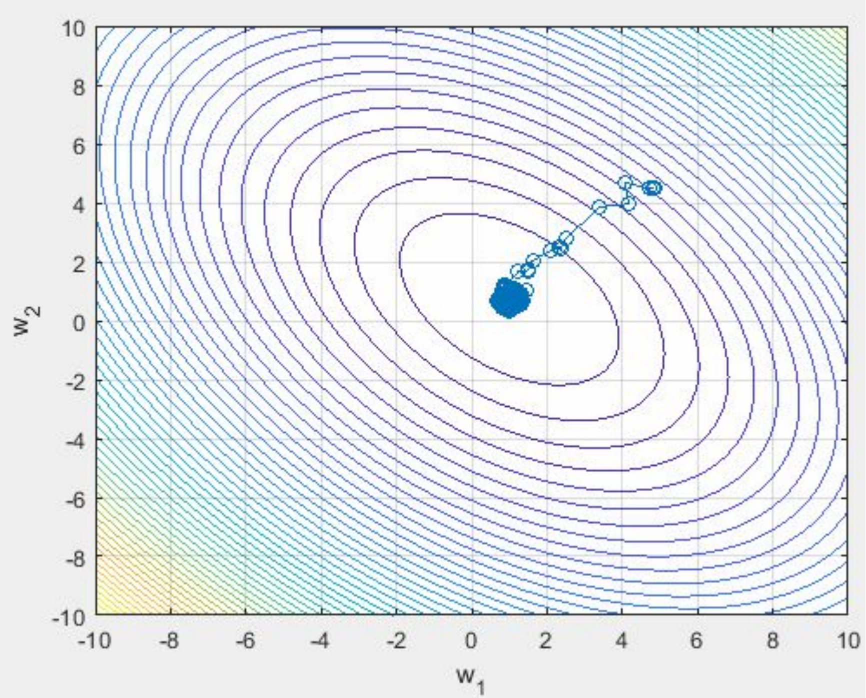
d)

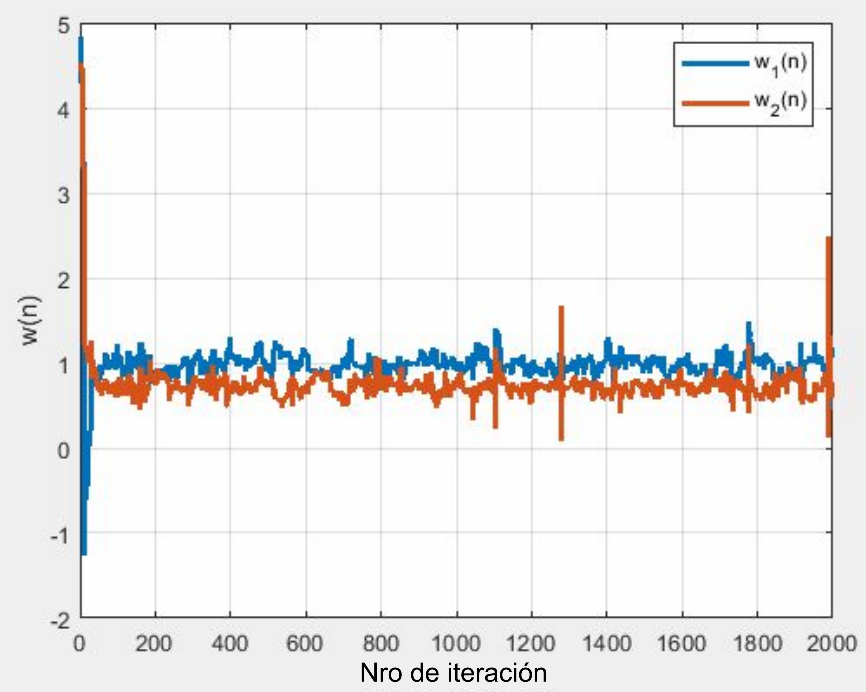


# Ejercicio 1

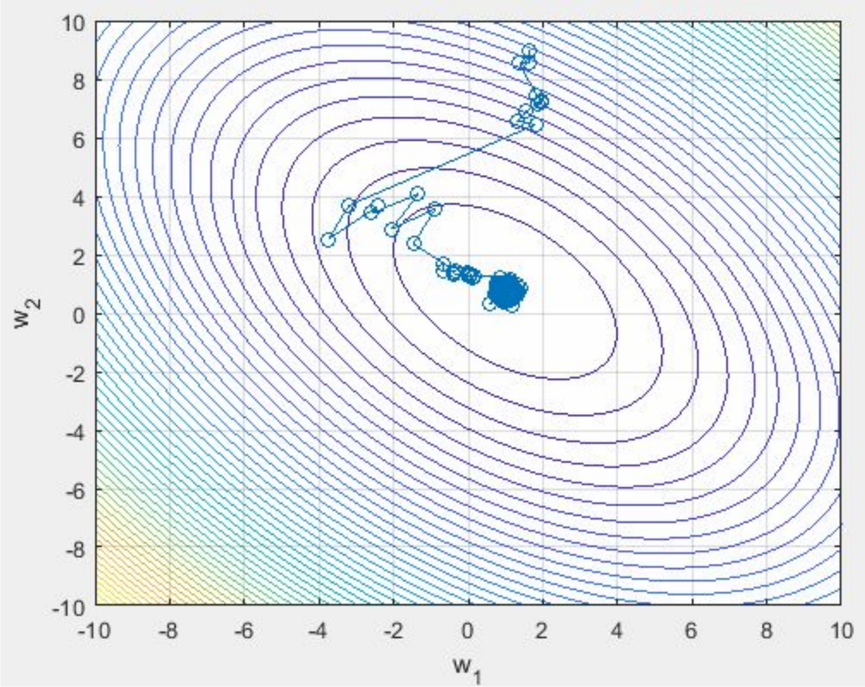
d)

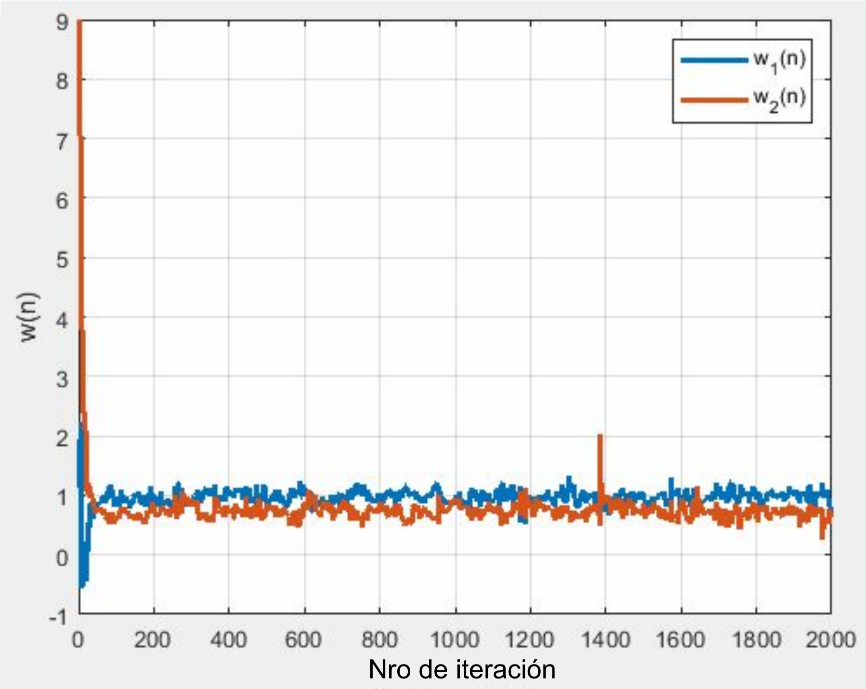


# Ejercicio 1

d)



# Ejercicio 2
# Cancelación de interferencia

# Ejercicio 2

Una señal de audio $s(t)$ es contaminada por una interferencia $g(t)$ de banda angosta de frecuencia conocida $f_0 = 1{,}5$ kHz, aunque se desconoce su amplitud y fase. La señal recibida se puede expresar como $x(t) = s(t) + g(t)$. Suponga que $s(t)$ es digitalizada a una frecuencia de $fs = 44100$ Hz y que la interferencia en el dominio discreto puede ser modelada como $g(n) = A\cos(\omega_i n + \theta)$. Se requiere utilizar un filtro adaptativo LMS para cancelar esta interferencia y obtener una aproximación $\widehat{s}(n)$ de la señal de audio $s(n)$.

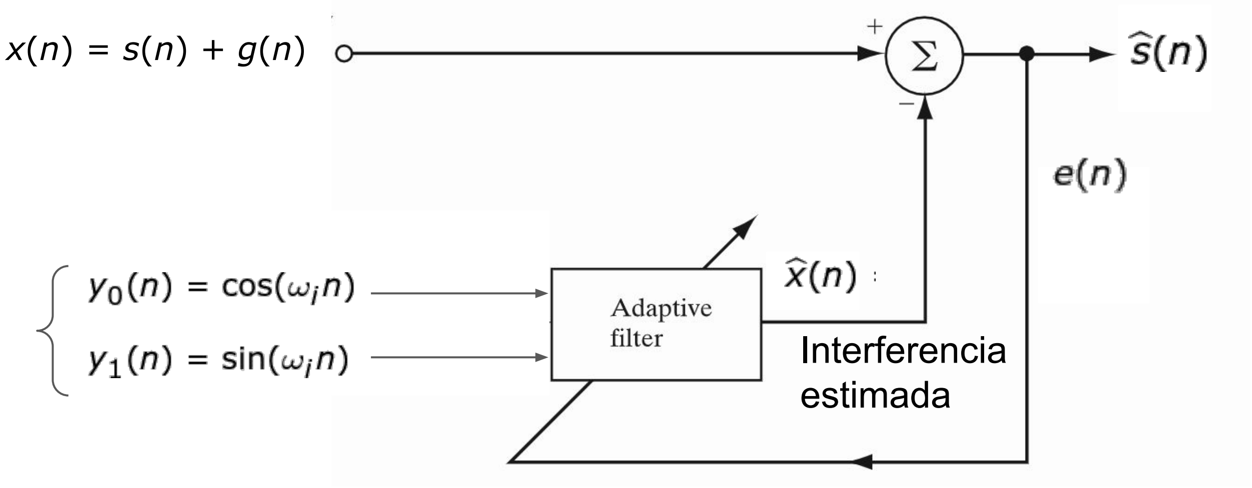
# Ejercicio 2

(a) Considerando un largo de filtro $M = 2$ y tomando como entradas al vector $\mathbf{y}(n) = [\cos(\omega_0 n) \;\; \sin(\omega_0 n)]^T$, haga un diagrama en bloques proponiendo una solución mediante el filtrado con el algoritmo LMS. ¿Qué significado tiene el error óptimo $e_o(n)$ en este problema? Tenga en cuenta que la interferencia también se puede expresar como

$$g(n) = A\cos(\omega_0 n + \theta) = B\cos(\omega_0 n) + C\sin(\omega_0 n)$$

# Ejercicio 2

a) *Cancelador de interferencias*



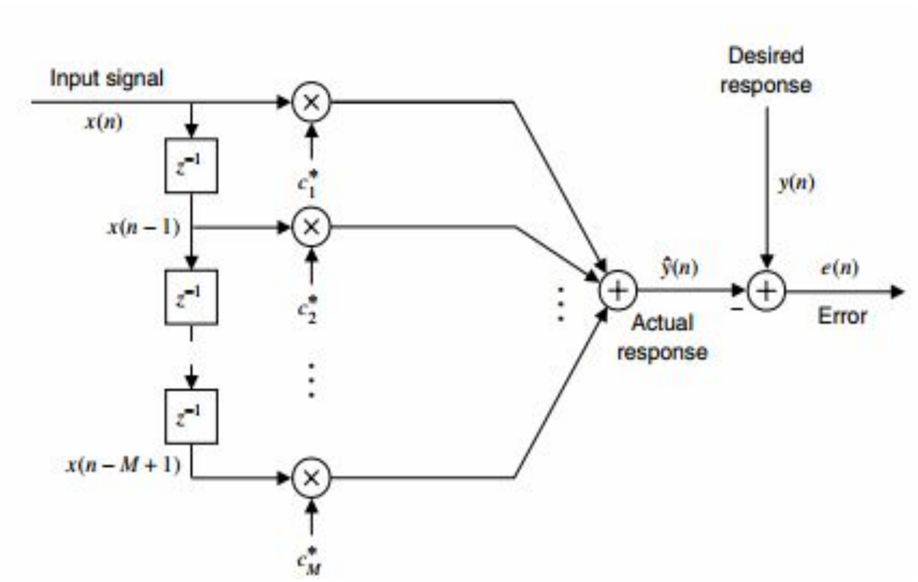

Entradas

$$\mathbf{y}(n) = \begin{bmatrix} \cos(\omega_i n) \\ \sin(\omega_i n) \end{bmatrix} \qquad \begin{cases} y_0(n) = \cos(\omega_i n) \\ y_1(n) = \sin(\omega_i n) \end{cases}$$

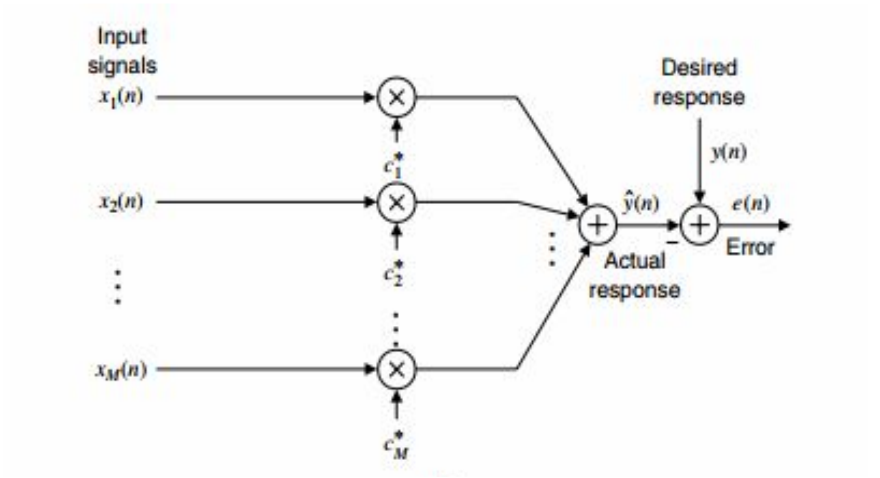
Modelo de la interferencia $\quad g(n) = A\cos(\omega_i n + \theta) = B\cos(\omega_i n) + C\sin(\omega_i n)$

# Ejercicio 2

Entradas de una ventana para un único proceso




Entradas de procesos en paralelo



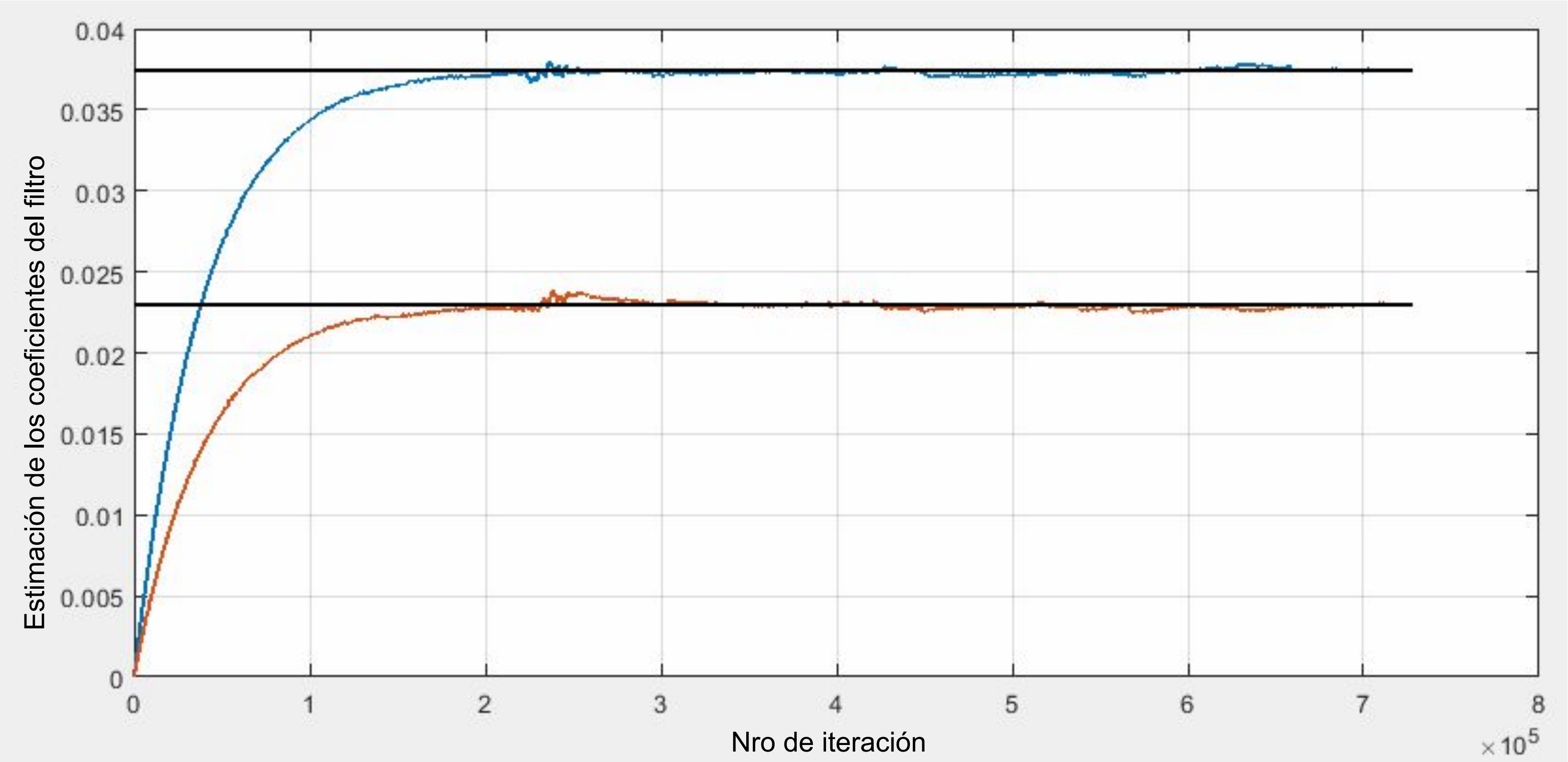
# Ejercicio 2

(b) Genere la señal $x(n)$ utilizando alguna de las pistas del TP1 para $s(n)$. Para generar la interferencia $g(n)$ considere que los parámetros desconocidos son V.A. $A \sim N(0{,}01\,;\,0{,}001)$ y $\theta \sim U(-\pi\,;\,\pi)$ (tenga en cuenta que éstos parámetros son V.A. pero toman un valor fijo en cada realización).

# Ejercicio 2

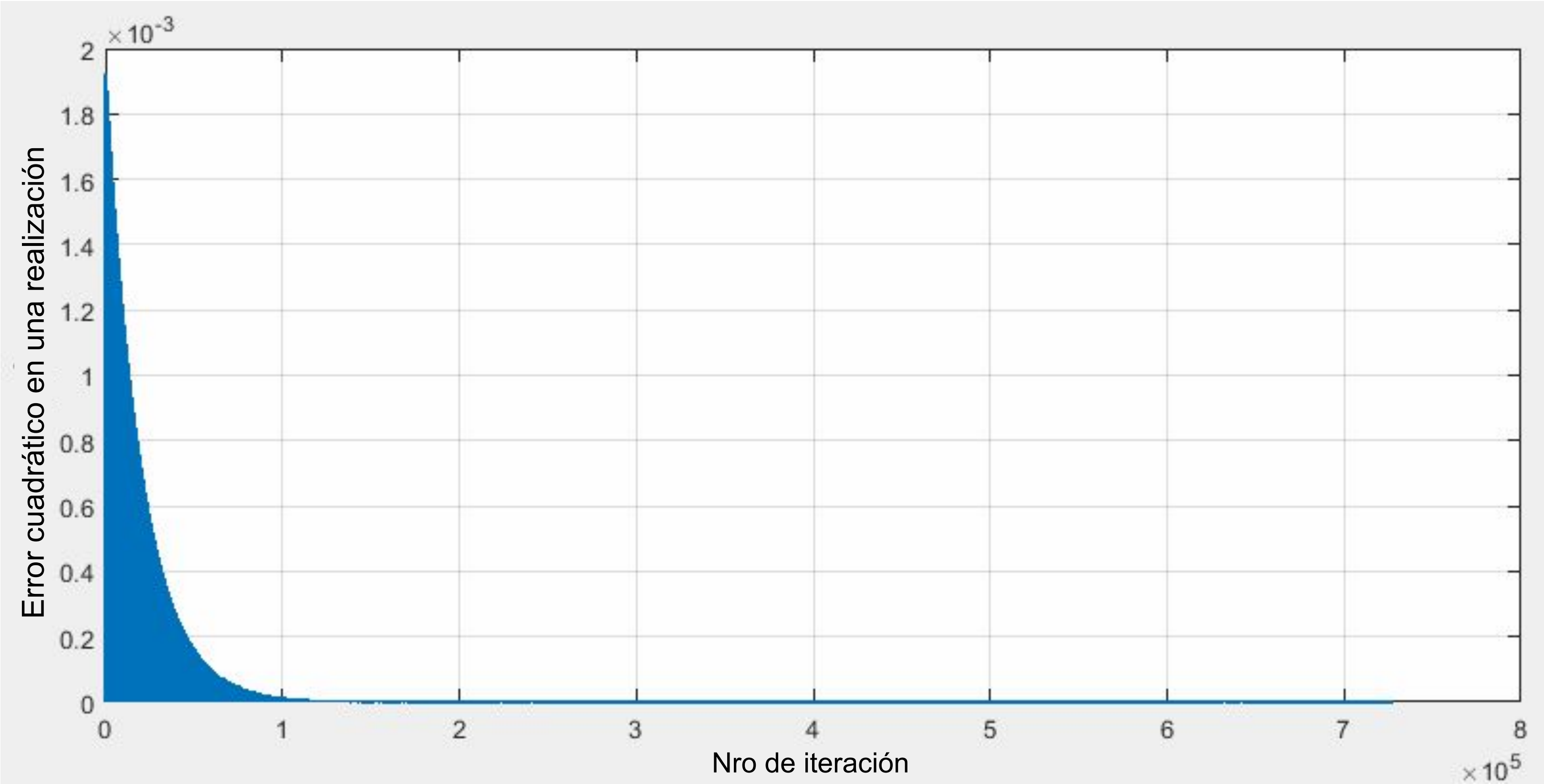
(c) Implemente un algoritmo LMS para resolver los coeficientes óptimos que permitan mitigar la interferencia. Ejecute el algoritmo para $\mu = 5 \times 10^{-5}$ y grafique en un mismo plot los $M$ coeficientes del filtro en función de las iteraciones. Grafique también la diferencia cuadrática entre la señal de interferencia y la salida del filtro LMS $|g(n) - \widehat{x}(n)|^2$.

# Ejercicio 2



0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
Estimación de los coeficientes del filtro
0
1
2
3
4
5
6
7
8
Nro de iteración
$\times 10^5$

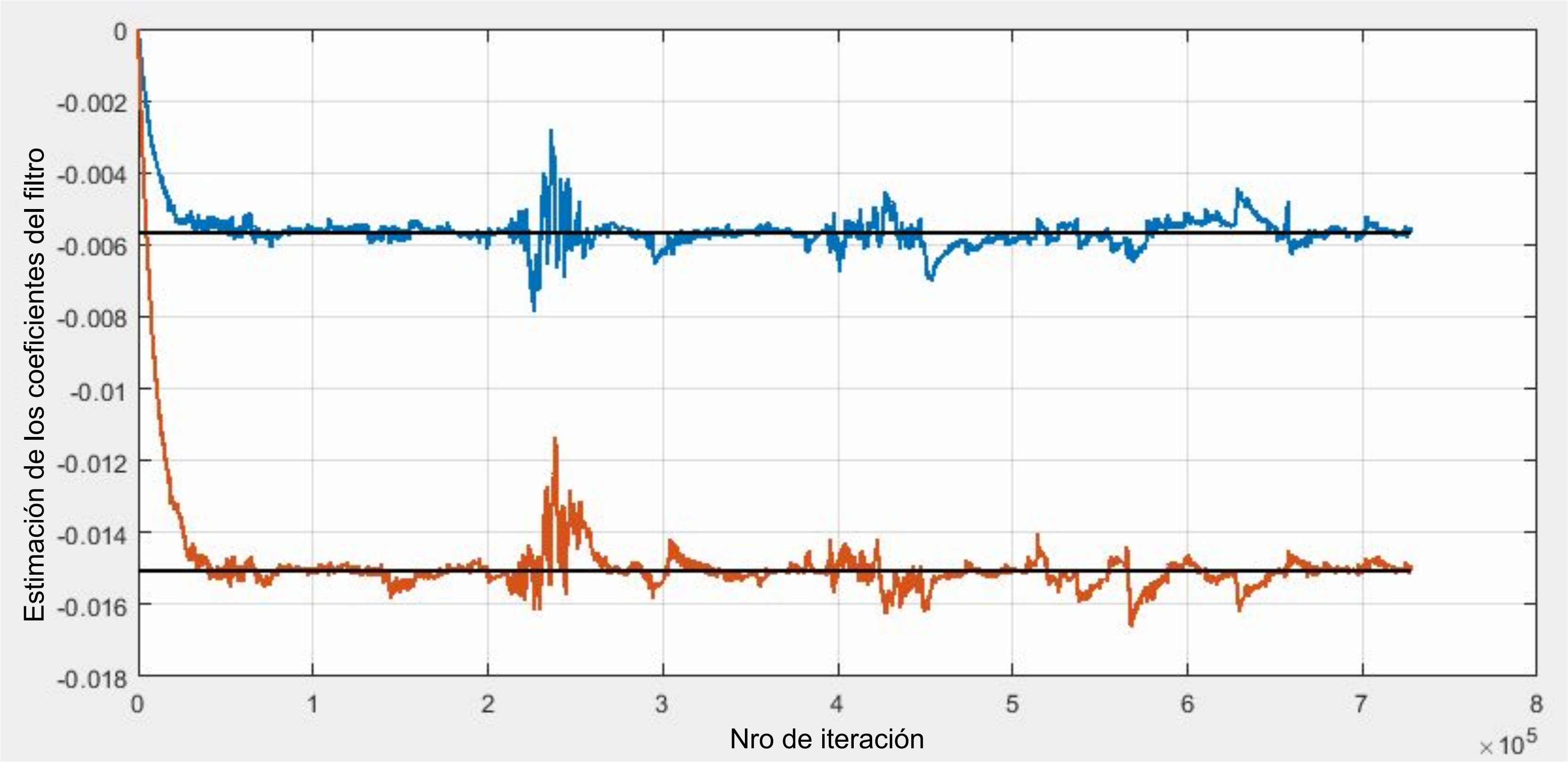
# Ejercicio 2


$\times 10^{-3}$
Error cuadrático en una realización
Nro de iteración
$\times 10^{5}$
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
1
2
3
4
5
6
7
8

# Ejercicio 2

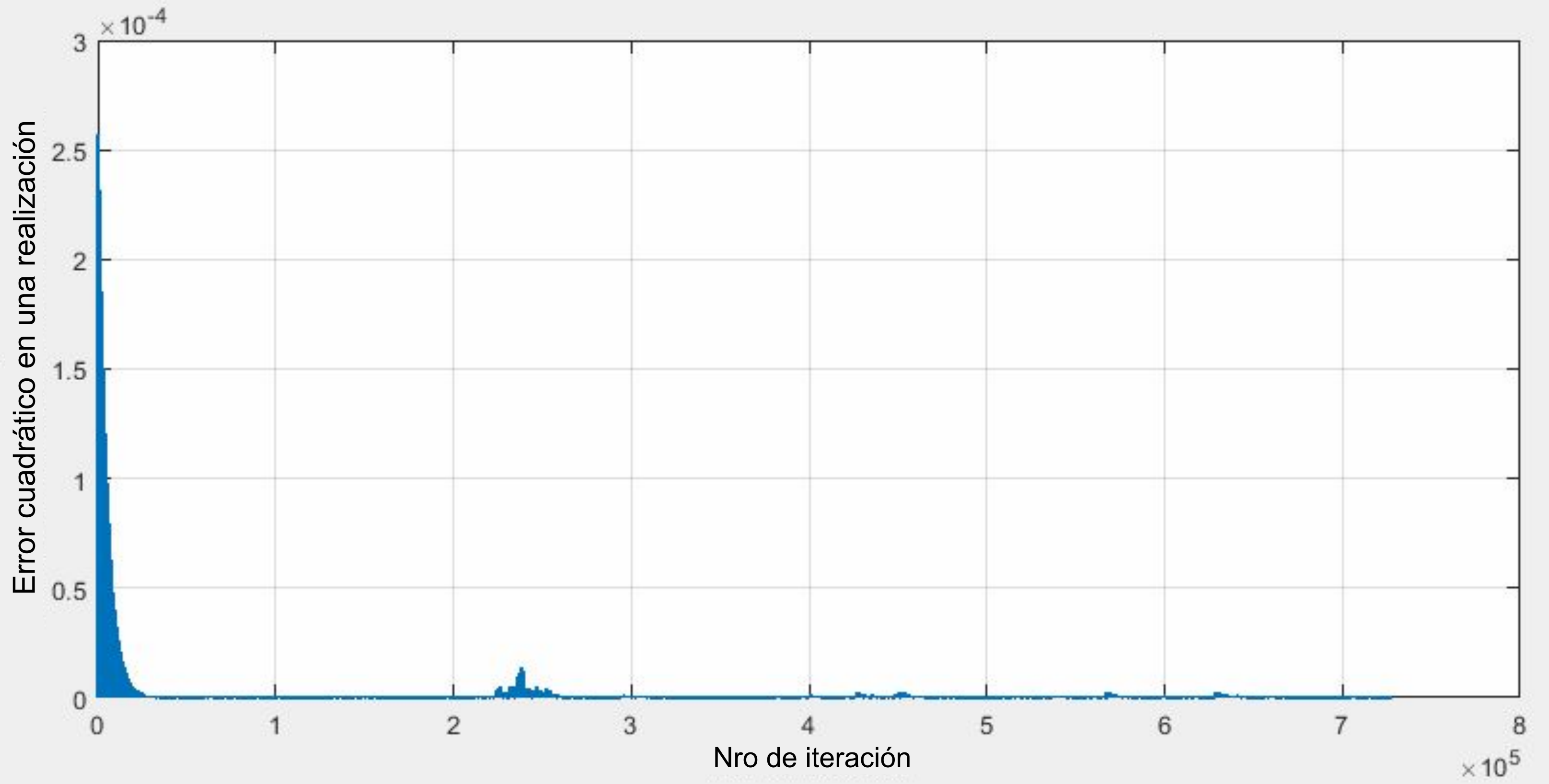
(d) Repita el punto anterior para $\mu = 2 \times 10^{-4}$. ¿Qué diferencias observa?

# Ejercicio 2



0
-0.002
-0.004
-0.006
-0.008
-0.01
-0.012
-0.014
-0.016
-0.018
Estimación de los coeficientes del filtro
0
1
2
3
4
5
6
7
8
Nro de iteración
×10^5

# Ejercicio 2

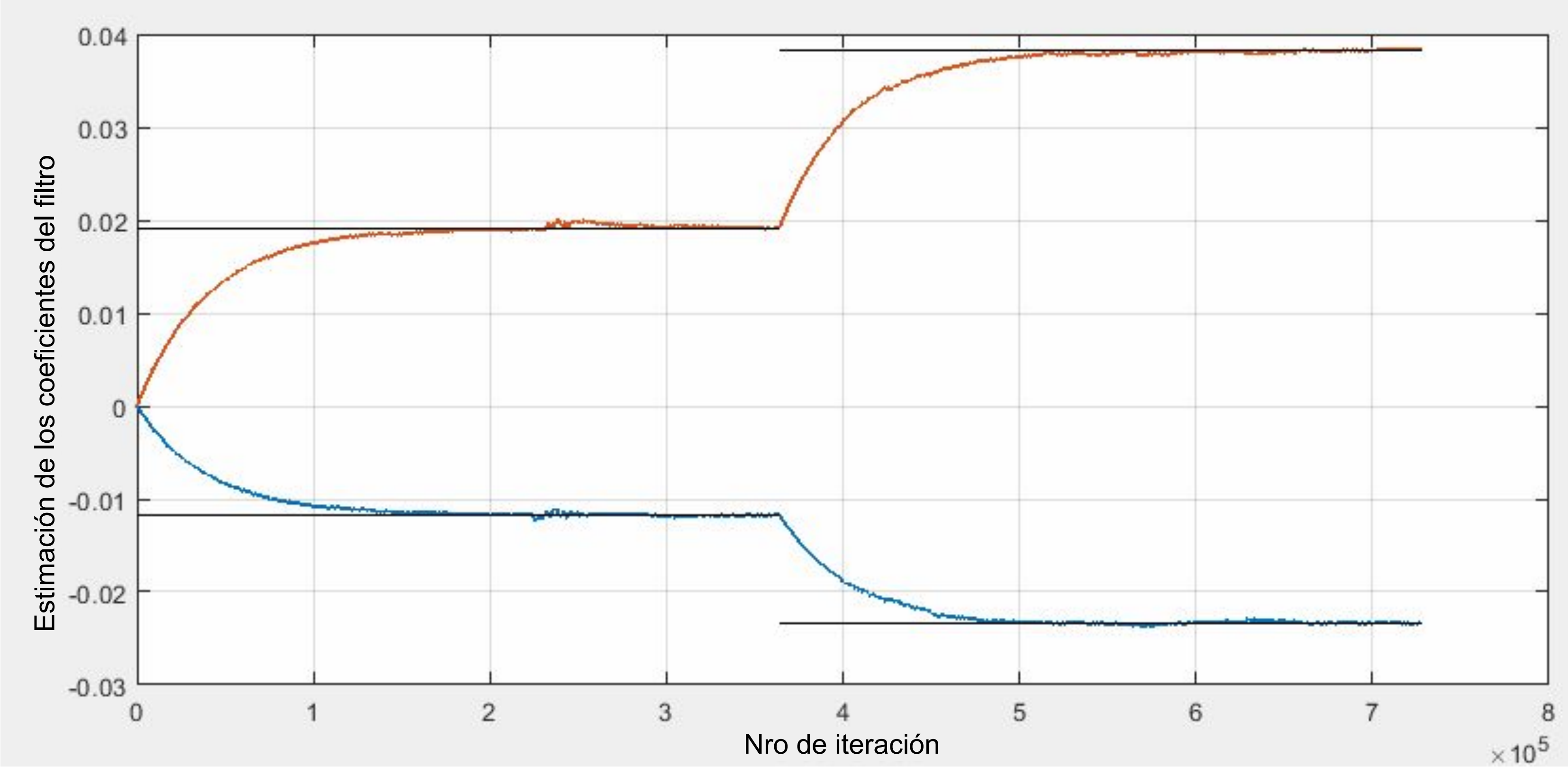


×10^-4
3
2.5
2
1.5
1
0.5
0
Error cuadrático en una realización
0
1
2
3
4
5
6
7
8
Nro de iteración
×10^5

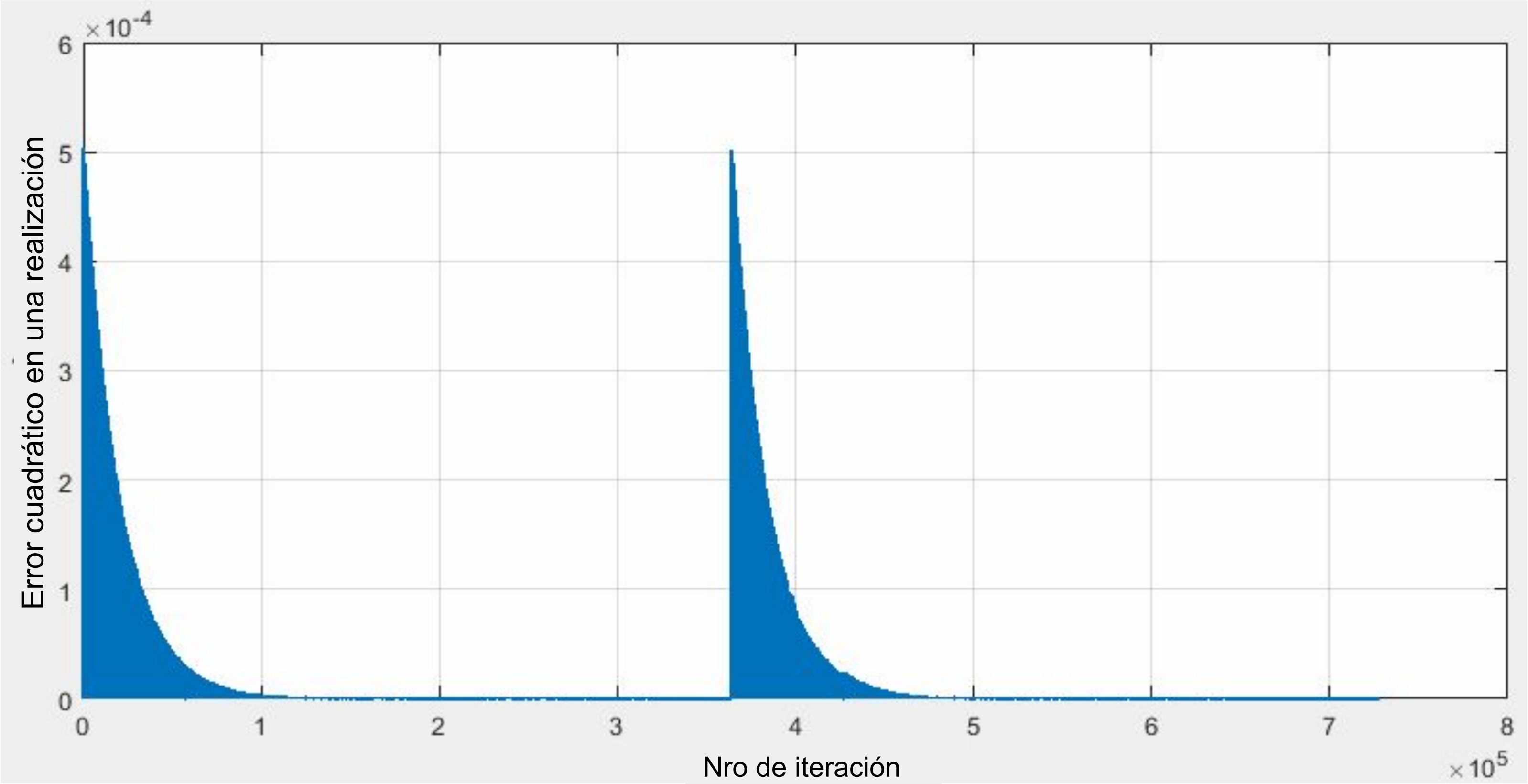
# Ejercicio 2

(e) Repita el punto (c) pero esta vez con una interferencia $g_2(n)$ que sufre un cambio abrupto tal que su amplitud se duplica en la mitad del intervalo temporal, es decir para $n > L/2$.

# Ejercicio 2


0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
Estimación de los coeficientes del filtro
0
1
2
3
4
5
6
7
8
Nro de iteración
×10^5

# Ejercicio 2



$\times 10^{-4}$
6
5
4
3
2
1
0
Error cuadrático en una realización
0
1
2
3
4
5
6
7
8
$\times 10^{5}$
Nro de iteración

# Ejercicio 2

(f) Reproduzca la señal de audio original contaminada con la interferencia $s(n)$ y luego la señal resultante con la cancelación de interferencia. Analice subjetivamente el desempeño del algoritmo para los distintos casos simulados.