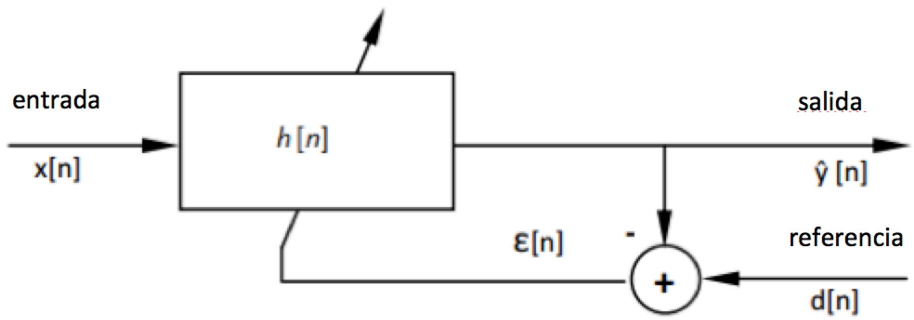
Enunciados

1) Buscar información por Internet e intentar resumir de forma concisa como son los diferentes tipos de estimadores que se listan a continuación:

- Estimador MAP (maximum a posteriori)
- Estimador ML (maximum likelihood)
- Estimador MS (mean square)
- Estimador LMS (lineal mean square)

Comentar en qué se basa cada uno de ellos y cuáles son las diferencias entre los distintos sistemas.

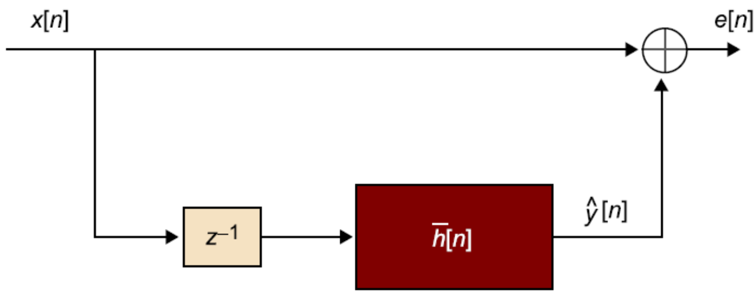
2) Sea el siguiente esquema:



Si consideramos que el filtro $h[n]$ es de dos coeficientes, determinar:

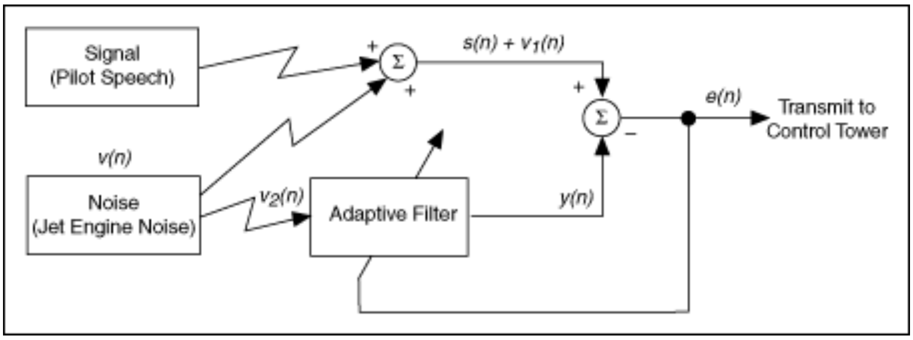
a) La expresión teórica de $E[|\varepsilon[n]|^2]$
b) Derivando el resultado anterior respecto de los valores de $h[n]$ e igualando la expresión a cero (es decir, buscando el mínimo de la función anterior), determinar la expresión teórica para el filtro $h[n]$ óptimo.
c) Razonar de qué tipo de filtro se trata.
d) Sustituyendo el resultado de (b) en la expresión de (a), determinar la expresión del error mínimo MSE.

3) Sea el sistema de la figura, que corresponde a un predictor lineal:



Determinar las ecuaciones teóricas de $E[|e[n]|^2]$, el valor óptimo para el filtro $h[n]$ y el valor mínimo del error $e[n]$ para este filtro óptimo. Particularizar el resultado general para un caso de filtro $h[n]$ de 2 coeficientes.

4) Sea el sistema de la figura, que corresponde a un típico sistema de cancelación de ruido:



Observamos que la señal original $s[n]$ está contaminada por un ruido $v_1[n]$ que es una versión filtrada del ruido $v[n]$. A la vez, tenemos un filtro adaptativo que tiene a su entrada otra versión de ruido $v_2[n]$ y genera una salida $y[n]$ que nos sirve para intentar recuperar la señal original (sin ruido).

Para simplificar el diseño, supondremos que $v_2[n] = v[n]$.

Se pide:

a) Explique cómo cree que funciona el sistema, cuál es el objetivo del filtro adaptativo, cómo trabaja, qué genera a su salida $y[n]$ y por lo tanto qué representa la señal $e[n]$.
b) Suponiendo que los ruidos $v[n]$ y $v_1[n]$ están relacionados por la ecuación $v_1[n] = v[n] * h_1[n]$, donde $h_1[n] = 0.5^n u[n]$ es la repuesta impulsional de un filtro causal, calcular y dibujar las primeras 10 muestras de $v_1[n]$ si $v[n] = cos\left(\frac{\pi}{2}n\right)u[n]$.
c) Sabiendo que el ruido tiene media nula, es blanco y de potencia σ_v^2, y es independiente de la señal $s[n]$, determine las expresiones de las matrices de correlación $\bar{\bar{R}}_{vv}$ y $\bar{\bar{R}}_{sv}$.
d) Calcular la solución óptima del filtro adaptativo, con respuesta impulsional $h[n]$ limitada a 10 muestras, para minimizar el efecto del ruido sobre la señal.
e) Explica de manera intuitiva cuál sería la estructura más adecuada para el filtro $h_1[n]$ para poder eliminar totalmente el ruido en $e[n]$

Ejercicios resueltos

5) Sean los dos esquemas presentados en la figura 1 y en la figura 2. Considera que todas las señales y filtros que tenemos son de coeficientes reales. Se pide:

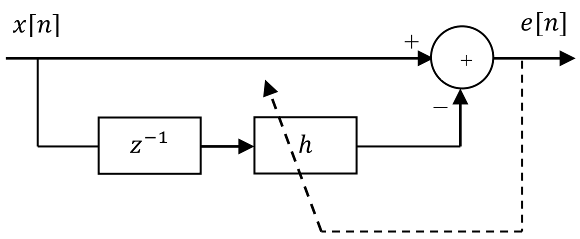


Figura 1

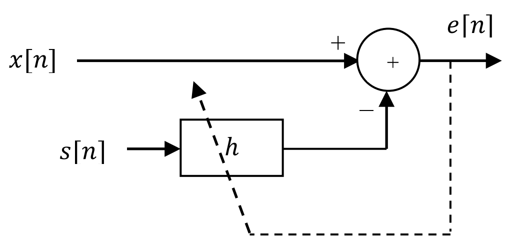


Figura 2

a. Expresar, para los dos casos, como queda la función de coste $J = E[|e[n]|^2]$

Para el caso 1: la señal de entrada al filtro FIR h es $x[n-1]$, mientras que la señal de referencia (señal deseada) es $x[n]$. Dado que nos dicen que tanto la señal como el filtro son reales, tendremos:

$$J = E[|e[n]|^2] = E[|x[n] - x[n-1] * h[n]|^2] =$$
$$= E\left[\left(x[n] - \overrightarrow{h^T}\vec{x}[n-1]\right)\left(x[n] - \overrightarrow{h^T}\vec{x}[n-1]\right)\right]$$

donde $\vec{h} = \begin{bmatrix} h_0 \\ \dots \\ h_N \end{bmatrix}$ es vector de coeficientes del filtro h y $\vec{x}[n-1] = \begin{bmatrix} x[n-1] \\ \dots \\ x[n-N-1] \end{bmatrix}$ es vector de muestras de la señal de entrada de este filtro. De momento no podemos saber qué ordren N de filtro tendremos.

Para el caso 2: ahora la señal de entrada al filtro es $s[n]$ mientras que la señal de referencia continua siendo $x[n]$. De manera similar al caso anterior, tendremos:

$$J = E[|e[n]|^2] = E[|x[n] - s[n] * h[n]|^2] =$$
$$= E\left[\left(x[n] - \overrightarrow{h^T}\vec{s}[n]\right)\left(x[n] - \overrightarrow{h^T}\vec{s}[n]\right)\right]$$

donden $\vec{h} = \begin{bmatrix} h_0 \\ \dots \\ h_N \end{bmatrix}$ es el vector de coeficientes del filtro h y $\vec{s}[n] = \begin{bmatrix} s[n] \\ \dots \\ x[n-N] \end{bmatrix}$. De moment no podem saber qué ordre N de filtro tendremos.

b. Deducir, a partir de esta función, la solución óptima para determinar los coeficientes del filtro *h*

En los dos casos, la solución óptima vendrá dada por la derivada de la función de coste, respecto de los coeficientes del filtro, igualada a cero: $\frac{\partial J}{\partial h} = 0$

De forma general, y siguiendo el desarrollo que tenemos en los apuntes, obtenemos la solución como:

$$h_{opt} = R^{-1}\vec{r}$$

Para el caso 1: $R = E\left[\vec{x}[n-1]\vec{x}^T[n-1]\right] = R_{xx}$
$\vec{r} = E[x[n]\vec{x}[n-1]]$

Para el caso 2: $R = E\left[\vec{s}[n]\vec{s}^T[n]\right] = R_{ss}$
$\vec{r} = E[x[n]\vec{s}[n]]$

c. Explicar qué crees que están realizando cada uno de los dos sistemas.

El primer sistema es un predictor lineal, en el que se intenta obtener una predicción del valor de $x[n]$ a partir de muestras anteriores de x. Si la predicción es buena, la señal error e tenderá a cero.

El segundo sistema está intentando identificar el sistema que genera la señal $x[n]$ a partir de $s[n]$. Nótese que si realmente $x[n]$ proviene (se ha generado) a partir de la señal s filtrada, entonces el sistema intenta identificar este filtro mediante h. Si la identificación se hace de forma correcta, la señal error e tenderá a cero.

d. Determinar los valores óptimos del filtro h para el caso de la figura 1 si sabemos que $r_x[m] = \delta[m+2] - 4\delta[m+1] + 6\delta[m] - 4\delta[m-1] + \delta[m-2]$ y que el filtro h tiene 4 coeficientes.

Dado que tenemos la forma de r_x estamos en disposición de calcular los valores de la matriz R_{xx} y del vector $\vec{r}$:

$$R_{xx} = \begin{bmatrix} r_x[0] & r_x[1] & r_x[2] & r_x[3] \\ r_x[-1] & r_x[0] & r_x[1] & r_x[2] \\ r_x[-2] & r_x[-1] & r_x[0] & r_x[1] \\ r_x[-3] & r_x[-2] & r_x[-1] & r_x[0] \end{bmatrix} = \begin{bmatrix} 6 & -4 & 1 & 0 \\ -4 & 6 & -4 & 1 \\ 1 & -4 & 6 & -4 \\ 0 & 1 & -4 & 6 \end{bmatrix}$$

$$\vec{r} = \begin{bmatrix} r_x[1] \\ r_x[2] \\ r_x[3] \\ r_x[4] \end{bmatrix} = \begin{bmatrix} -4 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Por lo tanto:

$$h_{opt} = R^{-1}{}_{xx}\,\vec{r} = \begin{bmatrix} 0.4762 & 0.5714 & 0.4286 & 0.1905 \\ 0.5714 & 1.0857 & 0.9143 & 0.4286 \\ 0.4286 & 0.9143 & 1.0857 & 0.5714 \\ 0.1905 & 0.4286 & 0.5714 & 0.4762 \end{bmatrix} \begin{bmatrix} -4 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1.3333 \\ -1.2000 \\ -0.8000 \\ -0.3333 \end{bmatrix}$$

e. ¿Qué pasaría si la función $r_x[m]$ fuese nula? Qué valor se obtendría en $e[n]$ de la figura 1? Razonar claramente la respuesta según lo que habéis explicado en el apartado (c).

Si $r_x[m]$ fuera siempre cero significaría que la señal $x[n]$ tiene autocorrelación nula, es decir, que sus valores son aleatorios y por tanto no podemos predecir ningún valor utilizando muestras pasadas. En este caso, como el predictor lineal no funcionaría, tendríamos los coeficientes del filtro h nulos y $e[n]$ sería directamente $x[n]$.

f. Determinar los valores óptimos del filtro h para el caso de la figura 2 si sabemos que s[n] es un proceso estocástico ergódico blanco, de potencia 1, que $r_{xs}[m] = \delta[m] - 2\delta[m-1] + \delta[m-2]$, y que el filtro h tiene 4 coeficientes.

Dado que nos dan la forma de r_{xs} estamos en disposición de calcular los valores de la matriz R_{ss} y del vector $\vec{r}$:

$$R_{ss} = I_4$$

$$\vec{r} = \begin{bmatrix} r_{xs}[0] \\ r_{xs}[1] \\ r_{xs}[2] \\ r_{xs}[3] \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 1 \\ 0 \end{bmatrix}$$

Por lo tanto:

$$h_{opt} = R^{-1}{}_{xx}\,\vec{r} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 1 \\ 0 \end{bmatrix}$$

g) Cómo interpretas este resultado para los valores de h de la figura 2? Razonar claramente la respuesta según lo que habéis explicado en el apartado (c).

Como hemos dicho que se trataba de un sistema para identificar la relación entre $s[n]$ y $x[n]$, lo que hemos obtenido es el filtro $h[n]$ que permite generar $x[n]$ a partir del proceso $s[n]$ que es estocástico ergódico blanco, de potencia 1. Esto quiere decir que $x[n] = h[n] * s[n]$.

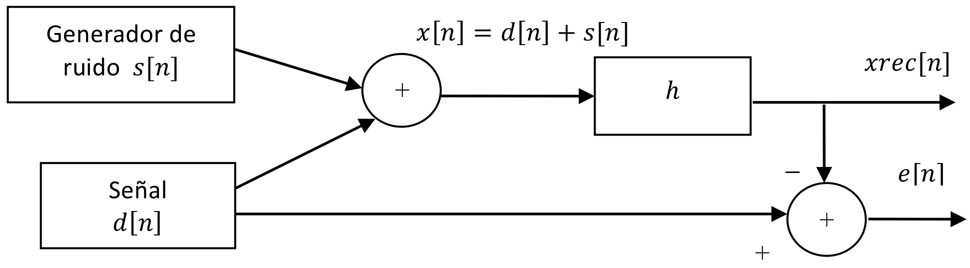
6) Sea el código Matlab siguiente, que implementa la solución del filtro de Wiener.

```
clear all
close all
clc
q = input('orden del filtro: ');
N = 100;
w = 0.45;
d = sin(w*(1:N));
llavor = RandStream('mcg16807', 'Seed',0);
RandStream.setGlobalStream(llavor); % para forzar el
% generador pseudoaleatorio a que genere siempre las
% mismas muestras
n = randn(1,N);
x = d+n;
r = xcorr(x);
R = toeplitz(r(N:N+q-1));
ptemp = xcorr(x,d);
p = ptemp(N:N+q-1)';
h = R\p;
```

Se pide:

a) Mirar el código y comentar qué se está realizando en las diferentes partes importantes. A partir de la identificación de estas partes, dibujar un diagrama de bloques en el cuál se vean las diferentes variables del código y por lo tanto se pueda entender qué hace el programa.

Observando el código, deducimos que el sistema que estamos implementando es como el de la figura, donde por simplicidad hemos llamado $s[n]$ al ruido que en el código aparece con el nombre de variable n:



b) Completar el código con las instrucciones que falten para recuperar la señal sin ruido, que llamaremos $xrec$.

Si el sistema es capaz de compensar el ruido $s[n]$, lo que obtendremos en la salida será una señal $xrec[n]$ que se corresponderá con una estimación de la señal original sin ruido: $xrec[n] \approx d[n]$.

El código final (en rojo lo que hemos añadido) serà:

```
clear all
close all
clc
q = input('orden del filtro: ');
N = 100;
w = 0.45;
d = sin(w*(1:N));
llavor = RandStream('mcg16807', 'Seed',0);
```

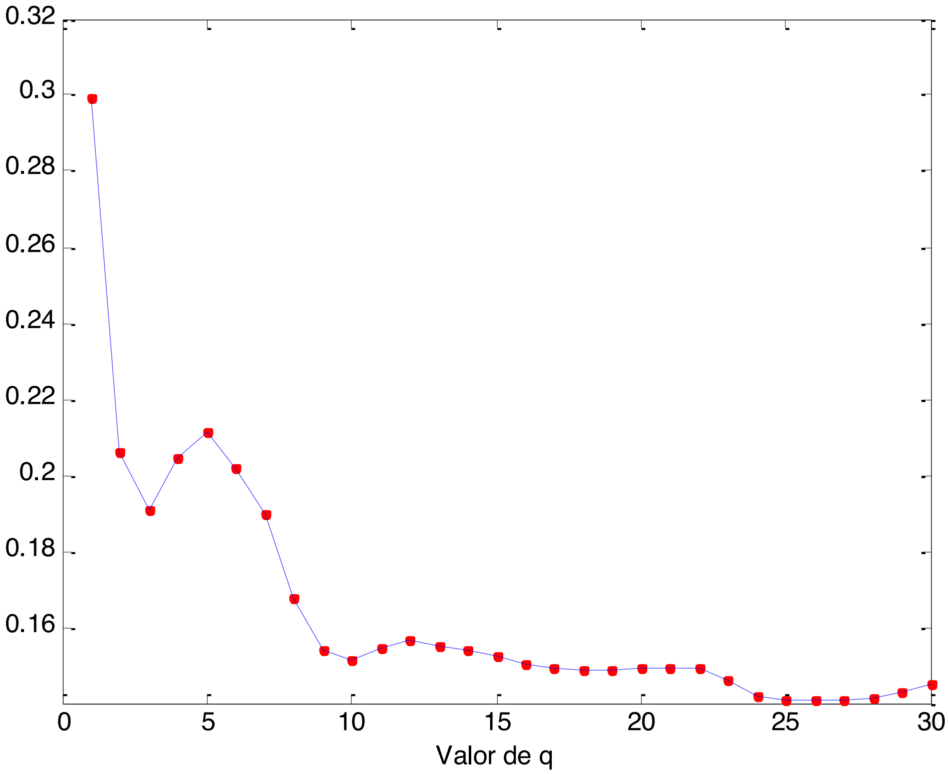
```
RandStream.setGlobalStream(llavor); % para forzar el
% generador pseudoaleatorio a que genere siempre las
% mismas muestras
n = randn(1,N);
x = d+n;
r = xcorr(x);
R = toeplitz(r(N:N+q-1));
ptemp = xcorr(x,d);
p = ptemp(N:N+q-1)';
h = R\p;
xrec = filter(h,1,x);           % estimación de la señal original d
error = sum(abs(xrec-d).^2)/N;  % cálculo del error según apartado (c)
```

c) Generar una taba de la evolución del error MSE que se obtiene cuando cambia el valor del orden del filtro, desde $q = 1$ hasta $q = 30$:[1]

q	$MSE = \frac{1}{N}\sum_{i=1}^{N}\lvert e\rvert^2$
1	0.2994
2	0.2062
3	0.1908
4	0.2046
5	0.2114
6	0.2018
7	0.1895
8	0.1677
9	0.1540
10	0.1514
11	0.1547
12	0.1566
13	0.1551
14	0.1542
15	0.1522
16	0.1502
17	0.1494
18	0.1489
19	0.1489
20	0.1492
21	0.1494
22	0.1490
23	0.1462
24	0.1418
25	0.1410
26	0.1410
27	0.1410
28	0.1416
29	0.1431
30	0.1449

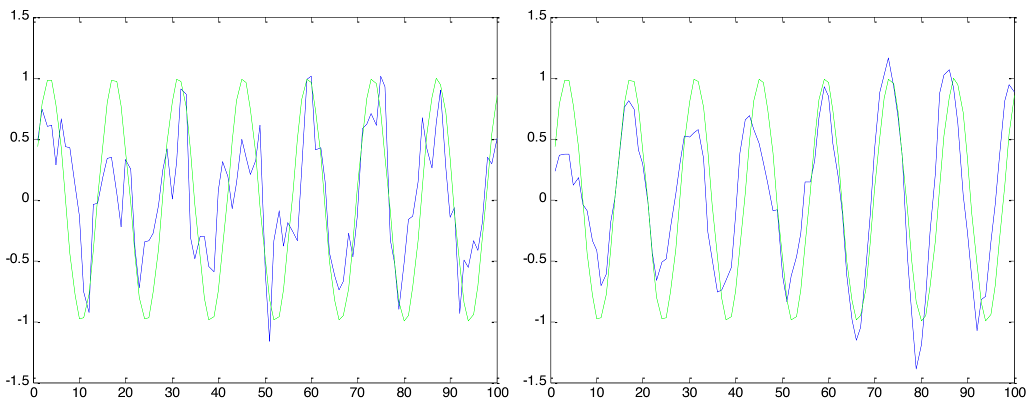
[1] Estrictamente hablando, el orden del filtro es $q - 1$ ya que q es el número de coeficientes del filtro FIR

Si visualizamos la evolución del error MSE tendremos:



d) Visualizar la señal $xrec$ y la señal que teóricamente queríamos obtener, para un valor de de $q = 2$ y para un valor de $q = 20$.

Observamos la señal que recuperamos (en azul) y la señal original sin ruido (en verde), a la izquierda para $q = 2$ y a la derecha para $q = 20$:



e) Comentar y relacionar las dos figuras obtenidas en el apartado anterior. Qué efecto tiene q en la forma de la señal recuperada?

Claramente se observa que el orden del filtro más grande nos permite una mejor estimación del ruido y por lo tanto una compensación de su efecto que nos genera al final una señal recuperado mucho más cercano al original. Es el mismo efecto que hemos visto en el apartado (c) con la dismunució del error al aumentar q.

Mirando con atención la figura del apartado (c) vemos que para q entre 25 y 27 el error es el más pequeño, y que aumenta posteriormente con q mayores. Este efecto nos estaría diciendo que órdenes superiores a los necesarios también generan problemas porque los coeficientes que "sobran" deberían ser nulos pero en la práctica no lo son, provocando por tanto que el error aumente.

7) Sea el esquema de la figura 1 donde el filtro FIR que aparece está formado por sólo dos coeficientes reales, w_0 y w_1. Sabemos que la varianza de la señal deseada y(n), σ_y^2, vale 0.9498. También conocemos las estadísticas R_x y r_{yx} ; están descritas a continuación:

$$\mathbf{R_x} = \begin{bmatrix} 1.1 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \qquad \text{y} \qquad \boldsymbol{r}_{y\mathbf{x}} = \begin{bmatrix} 0.5272 \\ -0.4458 \end{bmatrix}$$

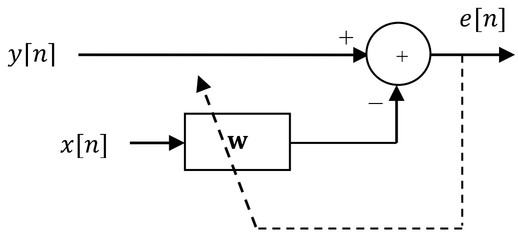


Figura 1

Se pide:

a. Desarrolla la función de coste $J(w_0, w_1)$ si sabemos que todas las señales implicadas son reales.

Sabemos que la función de coste toma la forma:

$$J(w_0, w_1) = E[e[n]^2] = E[(y[n] - \mathbf{w}^T\mathbf{x})(y[n] - \mathbf{w}^T\mathbf{x})]$$

En el nuestro caso:

$$\mathbf{x} = \begin{bmatrix} x[n] \\ x[n-1] \end{bmatrix} \text{ i } \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

Desarrollando la esperanza llegamos al resultado que aparece en los apuntes:

$$J(w_0, w_1) = E[e[n]^2] = \sigma_y^2 + \mathbf{w}^T\mathbf{R_x}\mathbf{w} - 2\mathbf{w}^T\mathbf{r}_{y\mathbf{x}}$$

Particularizando para los valores del enunciado:

$$J(w_0, w_1) = 0.9498 - 2\begin{bmatrix} w_0 & w_1 \end{bmatrix}\begin{bmatrix} 0.5272 \\ -0.4458 \end{bmatrix} + \begin{bmatrix} w_0 & w_1 \end{bmatrix}\begin{bmatrix} 1.1 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}\begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

y operando llegamos a desarrollar la función de coste:

$$J(w_0, w_1) = 0.9498 - 1.0544w_0 + 0.8916w_1 + 1.1(w_0^2 + w_1^2) + w_0w_1$$

b. Determina los coeficientes w_{0opt} y del filtro óptimo.

Sabemos que la solución óptima vendrá dada por la derivada de la función de coste, respecto de los coeficientes del filtro, igualada a cero: $\frac{\partial J(w_0, w_1)}{\partial \mathbf{w}} = \mathbf{0}$

De forma general, y siguiendo el desarrollo que encontramos en los apuntes, presentamos la solución como:

$$\mathbf{w}_{opt} = \mathbf{R_x}^{-1}\mathbf{r}_{y\mathbf{x}}$$

Particularizando los valores proporcionados en el enunciado tendremos:

$$\mathbf{w}_{opt} = \mathbf{R_x}^{-1}\mathbf{r}_{y\mathbf{x}} = \begin{bmatrix} 1.1458 & -0.5208 \\ -0.5208 & 1.1458 \end{bmatrix} \begin{bmatrix} 0.5272 \\ -0.4458 \end{bmatrix} = \begin{bmatrix} 0.8370 \\ -0.7857 \end{bmatrix}$$

$$w_{0opt} = 0.8370\ ,\ w_{1opt} = -0.7857$$

c. Determina el valor mínimo de la función de coste encontrada en el apartado a).

Podemos proceder de diferentes maneras. Una de ellas sería evaluar la función de coste en el punto $\mathbf{w}_{opt}$, o sea:

$$J(w_0, w_1)_{min} = J\left(w_{0opt}, w_{1opt}\right) =$$
$$0.9498 - 1.0544w_{0opt} + 0.8916w_{1opt} + 1.1\left(w_{0opt}^2 + w_{1opt}^2\right) + w_{0opt}w_{1opt}$$
$$= 0.1569$$

Otra posibilidad consiste en aplicar el resultado de los apuntes:

$$J(w_0, w_1)_{min} = \sigma_y^2 - \mathbf{w}_{opt}^T\mathbf{r}_{y\mathbf{x}} = 0.9498 - [0.8370 \quad -0.7857]\begin{bmatrix} 0.5272 \\ -0.4458 \end{bmatrix}$$
$$= 0.1569$$

8) La configuración representada a la figura 2 se utiliza para identificar la respuesta al impulso de un sistema S, a priori desconocido. La idea básica consiste en inyectar al sistema S una señal conocida, $x[n]$, y medir su salida y[n]. y[n] es la señal deseada y que queremos reproducir a la salida del filtro FIR, la entrada del cual es la misma señal x[n].
Para las pruebas que queremos hacer en este ejercicio vamos a asumir un sistema S con respuesta al impulso S=[2 -0.5 1.2].

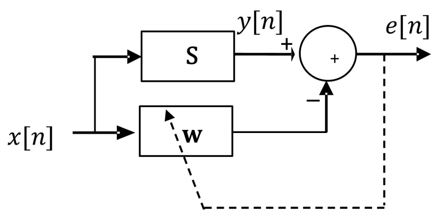


Figura 2

a. Genera 500 muestres de una señal x[n] a partir del código que se adjunta a continuación y calcula $\mathbf{R}_x$ y $\mathbf{r}_{yx}$ para la obtención de $\mathbf{w}_{opt}$ de longitudes 2,3 y 4 muestras utilizando la función *xcorr*. Proporciona los diferentes valores de $\mathbf{R}_x$ y $\mathbf{r}_{yx}$.

```
N=500;
seed = RandStream('mt19937ar','Seed',1)
RandStream.setGlobalStream(seed);
x = randn(1,N);
```

A parte de la señal x que ya viene determinada en el enunciado generamos y[n] y calcularemos la estadística necesaria con:

```
S=[2 -0.5 1.2];
y=filter(S,1,x);

r = xcorr(x);
Rx = toeplitz(r(N:N+q-1))

ptemp = xcorr(y,x);
ryx = ptemp(N:N+q-1)'
```

Donde `q` valdrá 2, 3 y 4.

Los resultados serán:

L=2

$$\mathbf{R_x} = \begin{bmatrix} 512.9 & 7.5 \\ 7.5 & 512.9 \end{bmatrix} \quad \mathbf{r}_{yx} = \begin{bmatrix} 1052.7 \\ -232.3 \end{bmatrix}$$

L=3

$$\mathbf{R_x} = \begin{bmatrix} 512.9 & 7.5 & 25.6 \\ 7.5 & 512.9 & 7.5 \\ 25.6 & 7.5 & 512.9 \end{bmatrix} \quad \mathbf{r}_{yx} = \begin{bmatrix} 1052.7 \\ -232.3 \\ 661.7 \end{bmatrix}$$

L=4

$$\mathbf{R_x} = \begin{bmatrix} 512.9 & 7.5 & 25.6 & -18.8 \\ 7.5 & 512.9 & 7.5 & 25.6 \\ 25.6 & 7.5 & 512.9 & 7.5 \\ -18.8 & 25.6 & 7.5 & 512.9 \end{bmatrix} \mathbf{r}_{yx} = \begin{bmatrix} 1052.7 \\ -232.3 \\ 661.7 \\ -41.9 \end{bmatrix}$$

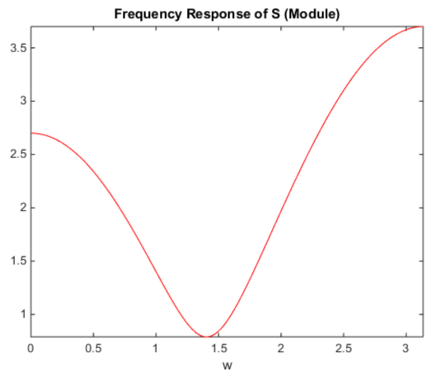
b. Porqué crees que se ha escogido una señal pseudoaleatoria como señal conocida a aplicar a la entrada?

Porque es una señal cuyas propiedades se parecen a las que tiene el ruido blanco y por tanto es una señal que tiene energía a todas las frecuencias del espectro y en consecuencia todas las frecuencias están representadas.

c. Representa el módulo de la respuesta en frecuencia del filtro de Wiener, **w**, para los casos de 2, 3 y 4 muestras y compárala con su respuesta real.

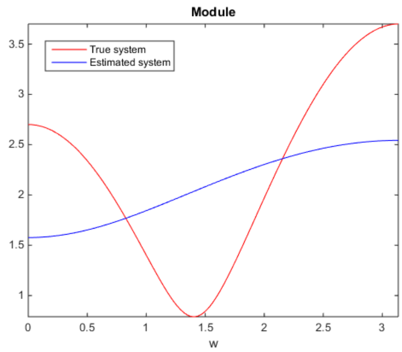
Calculamos y representamos el módulo de la respuesta en frecuencia de S:

```
S=[2 -0.5 1.2];
[H,w] = freqz(S,1);
plot(w,abs(H),'r'); title('Frequency Response of S Module)');xlabel('w');
axis tight
```

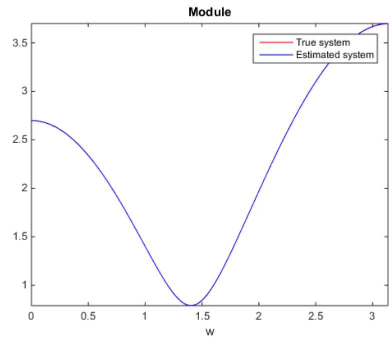


Utilizando $\mathbf{w}_{opt} = \mathbf{R_x}^{-1}\mathbf{r}_{yx}$ con los valores del apartado a) tendremos:

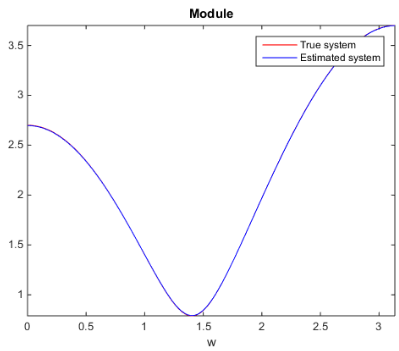
L=2



L=3



L=4



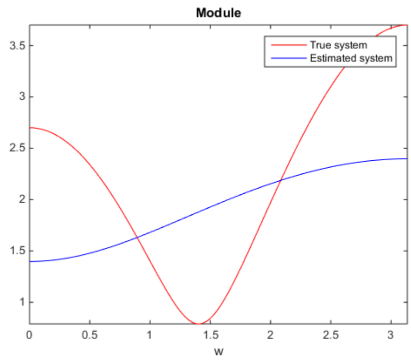
d. Comenta de forma breve y concisa los resultados del apartado anterior.

Para obtener una estimación correcta la longitud de la respuesta impulsional del filtro de Wiener debe ser igual o mayor que la respuesta impulsional del filtro a estimar. Si se cumple esta condición la estimación es prácticamente perfecta.

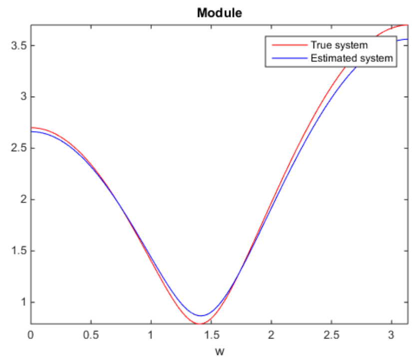
e. Utiliza el mismo código del apartado b cambiando N=500 per N=50 y vuelve a generar la señal x[n] de entrada y, a partir de esta vuelve a estimar $\mathbf{R}_x$ y $\mathbf{r}_{yx}$. En estas condiciones representa el módulo de la respuesta en frecuencia del filtro de Wiener, **w**, para los casos de 2, 3 y 4 muestras. Compara estos resultados con la respuesta real de S.

Procediendo de la misma manera que en el apartado c obtenemos las gráficas siguientes:

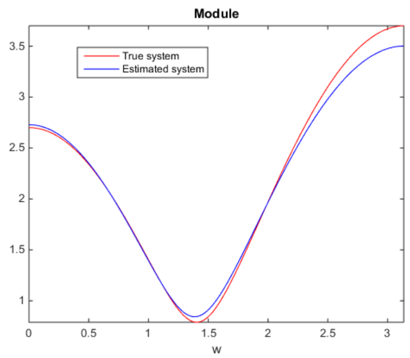
L=2



L=3



L=4



f. Comenta de forma breve y concisa los resultados del apartado anterior.

En este caso, a pesar de cumplir la condición de utilizar un filtro **w** de más muestras que las del filtro a estimar, se introduce un error debido a la pobre estimación de la estadística ($\mathbf{R}_x$ y $\mathbf{r}_{yx}$) ya que sólo disponemos de 50 muestras para su obtención.

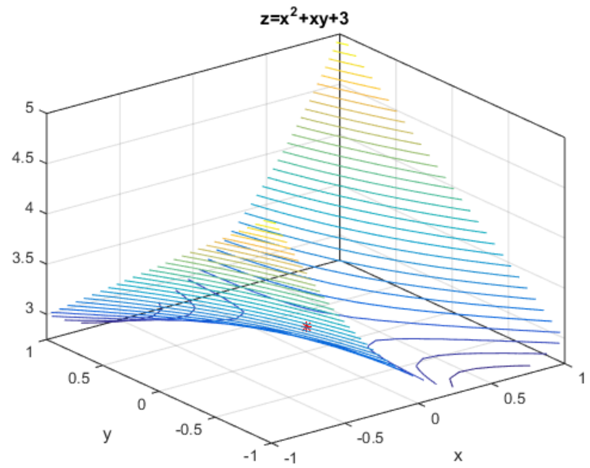
9) Necesitaremos realizar este problema a partir de los resultados del ejercicio anterior utilizando el programa MATLAB. Procederemos de forma guiada. Queremos construir una representación 3D de la función de coste calculada en el apartado a) del primer problema, de manera que los ejes x e y del plano se correspondan a los coeficientes del filtro w_0 y w_1 respectivamente y el eje z se corresponda con el valor que toma la función de coste $J(w_0,w_1)$.

a. Pera obtener la representación 3D utilizaremos principalmente las funciones *meshgrid, contour3, plot3* y *view* (a parte de otras funciones genéricas ya utilizadas como son *xlabel, ylabel, title, hold on* y *hold off*). Pasa eso, como paso previo realizad un *help* de cada una de les funciones anteriores y interpretad el código siguiente que proporciona una representación de la función $z=x^2+xy+3$ en la sección de plano $-1 \leq x \leq 1$ y $-1 \leq y \leq 1$ a incrementos de 0.1 tanto en x como en y. Simultáneamente aparece representado el punto (0,0,3) mediante un asterisco de color rojo. Ejecuta el código y proporciona la gráfica resultante.

```
close all
clear all
% Example
d=0.1;
[X,Y]=meshgrid(-1:d:1,-1:d:1);
Z=X.^2+X.*Y+3; % Z function

N=30;
figure(1)
contour3(X,Y,Z,N)
xlabel('x');ylabel('y');title('z=x^2+xy+3')
view(3) % Same point of view
hold on
plot3(0,0,3,'*r')
hold off
```

El resultado de ejecutar el código anterior proporciona la gráfica siguiente.



b. En el código anterior, qué significan las operaciones .^ y .* que aparecen en la construcción de la matriz Z? ¿Qué controla el paràmetre N? I el d? Quintos puntos de Z se han calculado?

El punto delante las operaciones ^ i * significa que la operación se realiza elemento a elemento dentro la matriz, en vez de ser una operación matricial. Ej: si A es una matriz, A^2 es el producto matricial equivalente a A*A, en cambio A.^2 és una nueva matriz los elementos de la cual son los antiguos elementos elevados al cuadrado.

N controla el número de curvas de nivel de la representación tridimensional.

d controla el espaciado entre puntos vecinos tanto de les abscisas como de les ordenadas.

Hemos calculado un número de 21x21= 441 puntos de la función Z.

c. Representa la función de coste calculada en el apartado a) del problema 1 de manera que los ejes x e y del plano se correspondan a los coeficientes del filtro w_0 y w_1 respectivamente y el eje z se corresponda al valor que toma la función de coste $J(w_0,w_1)$. El espaciado entre puntos vecinos, tanto en w_0 como en w_1 deberá ser de 0.01. El nombre de curvas de nivel tendrá que ser 30 y el área a representar la comprendida entre: $-2\leq w_0 \leq 3$ y $-3\leq w_1 \leq 1$. Además la perspectiva de las representación deberá fijarse a través de la instrucción *view(3)*.

La función de coste a implementar es:

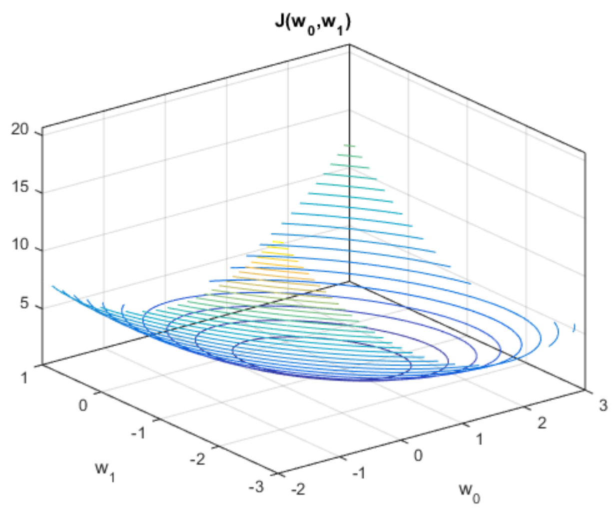
$$J(w_0,w_1) = 0.9498 - 1.0544w_0 + 0.8916w_1 + 1.1(w_0^2 + w_1^2) + w_0w_1$$

Siguiendo las especificaciones del enunciado generaremos el código siguiente:

```
[W0,W1]=meshgrid(-2:0.01:3,-3:0.01:1);
N=30; % Numero de seccions
% Apartat A
J=0.9498-1.0544*W0+0.8916*W1+1.1*(W0.^2+W1.^2)+W0.*W1;

figure(1)
contour3(W0,W1,J,N)
xlabel('w_0');ylabel('w_1');title('J(w_0,w_1)')
view(3)
hold on
```

El resultado lo representamos en la figura siguiente:

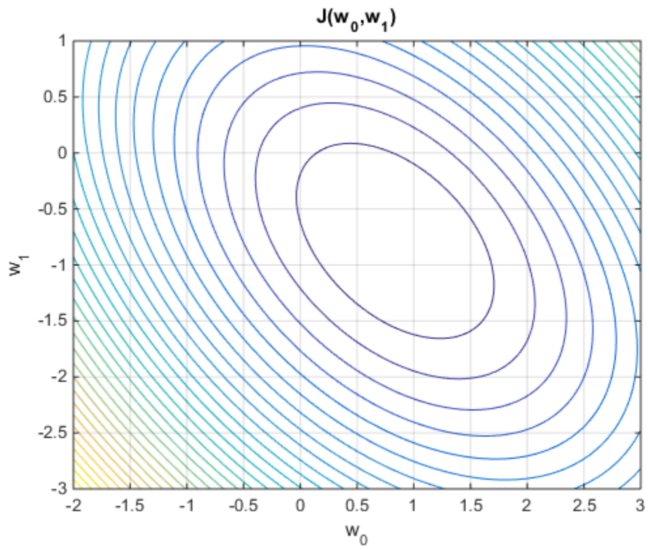


d. Con las mismas especificaciones anteriores repite la representación usando la perspectiva dada por la instrucción *view(2)*.

Cambiamos simplemente las últimas instrucciones

```
figure(2);
contour3(W0,W1,J,N)
xlabel('w_0');ylabel('w_1');title('J(w_0,w_1)')
view(2)
```

para obtener la representación:



e. A la vista de los resultados anteriores razona si los filtros de Wiener de dos coeficientes siempre presentan una solución y si esta es única o puede haber más de una. Este resultado es extensible para cualquier orden del filtre?

Vemos que la función de coste es una función cuadrática y como tal sólo presenta un mínimo que, además, es absoluto. La solución, que minimiza la función de coste encuentra sistemáticamente este mínimo. Para su cálculo sólo se necesita la estadística de segundo orden de los procesos. Este resultado se hace extensible a cualquier orden del filtro, aunque soluciones de mayor orden no son representables.

f. Con la ayuda de los resultados del problema 1, representa nuevamente las gráficas de los apartados c) y d) conjuntamente con el punto en color rojo correspondiente a la solución de Wiener.

Recopilando los resultados del problema 1 tenemos:

$$w_{0opt} = 0.8370$$
$$w_{1opt} = -0.7857$$
$$J(w_0, w_1)_{min} = 0.1569$$

Añadimos la porción de código:

```
w0opt=0.8370;
w1opt=-0.7857;
Jmin=0.1569;

figure(1)
plot3(w0opt,w1opt,Jmin,'*r')
```

```
hold off

figure(2)
plot3(w0opt,w1opt,Jmin,'*r')
hold off
```

y se obtiene: