



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
(ICAI)

Grado en Ingeniería en Tecnologías de Telecomunicaciones

**DISEÑO DE FILTROS ADAPTATIVOS PARA
LA REDUCCIÓN DE ARTEFACTOS EN
SEÑALES EMG**

Autor

Antón Cobián Iregui

Dirigido por

Nombres de los Directores

Madrid

Mayo 2026

Índice general

0.1. Introduction	2
1. Pruebas	3
1.1. Codigos en latex	3
1.2. Tablas y links	5
1.3. Funciones/Señales	6



0.1. Introduction

Capítulo 1

Pruebas

En este capítulo se practicará con diferentes funcionalidades de \LaTeX que pueden resultar útiles a la hora de redactar la memoria de un TFG.

1.1. Codigos en latex

Esta primera sección probará a ilustrar en el documento diversos lenguajes de programación con sus correspondientes códigos de ejemplo.

Comenzando por matlab, se muestra un breve programa de prueba.

```
1 mu = 0.01;
2 N = length(x);
3 y = zeros(N,1);
4 e = zeros(N,1);
5 w = zeros(M,1);
6 % Pepe
7 for n = M:N
8     x_vec = x(n:-1:n-M+1);
9     y(n) = w' * x_vec;
10    e(n) = d(n) - y(n);
11    w = w + mu * x_vec * e(n);
12 end
```

Listing 1.1: Filtro LMS en MATLAB

A continuación, se muestra otro ejemplo representando un código de python, pero en esta ocasión empleando la librería minted.

```
def generar_pares(limite):
    pares = []
    for num in range(1, limite + 1):
        if num % 2 == 0:
            pares.append(num)
    return pares

def sumar_lista(lista):
    total = 0
    for numero in lista:
        total += numero
```

```
        return total
    #Soy pepe
    limite = 20
    lista_pares = generar_pares(limite)
    suma = sumar_lista(lista_pares)

    print(f"Numeros pares hasta {limite}: {lista_pares}")
    print(f"Suma de los numeros pares: {suma}")
```

Ahora por último un ejemplo de código con java.

```
public class HolaMundoConNota {

    public static void main(String[] args) {
        String nombre = "María";
        saludar(nombre);

        // Aquí simulamos una "nota al pie"
        System.out.println("\nNota:1 Este saludo es parte de un
        ↪ ejemplo en Java.");
    }

    public static void saludar(String nombre) {
        System.out.println("Hola, " + nombre + "!"1); // El
        ↪ superíndice 1 actúa como marcador
    }
}
```

1.2. Tablas y links

Primeros	Segundos	Terceros
1	2	3
4	5	6

Cuadro 1.1: Tabla con numeros del 1 al 6

La tabla que se encuentra encima se ha diseñado haciendo uso de un paquete ¹

Por otro lado, para rellenar el resto de la página, usaré lo que se denomina *lipsum* ²

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Por último, aquí va una referencia a recursos que voy a emplear en el trabajo fin de grado:

1. Signals and Systems [1]
2. ECG Signal Processing Using Deep Learning [2]

¹Los paquetes en *l*átex permiten personalizar de manera exhaustiva el documento.

²*Lipsum* es otro paquete de *l*átex que permite rellenar con texto en latín

1.3. Funciones/Señales

Esta sección del capítulo de pruebas consistirá en conocer diferentes maneras de representar señales en figuras y gráficas. Primero, comenzaremos por una gráfica como la de la función $f(x) = \delta(t - 2)$.

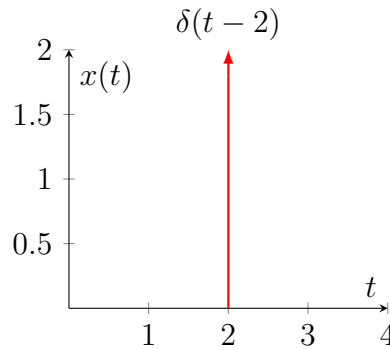


Figura 1.1: Caption

A continuación, se comentará por encima el concepto de "alyasing". A la hora de muestrear se debe emplear un f_s tal que $f_s \geq Bw$ para que se pueda recuperar la señal original. Con el fin de mostrar esta regla, se representará la señal $x(t) = \sin(100x) + \cos(40x)$ empleando 100 y 1000 puntos.

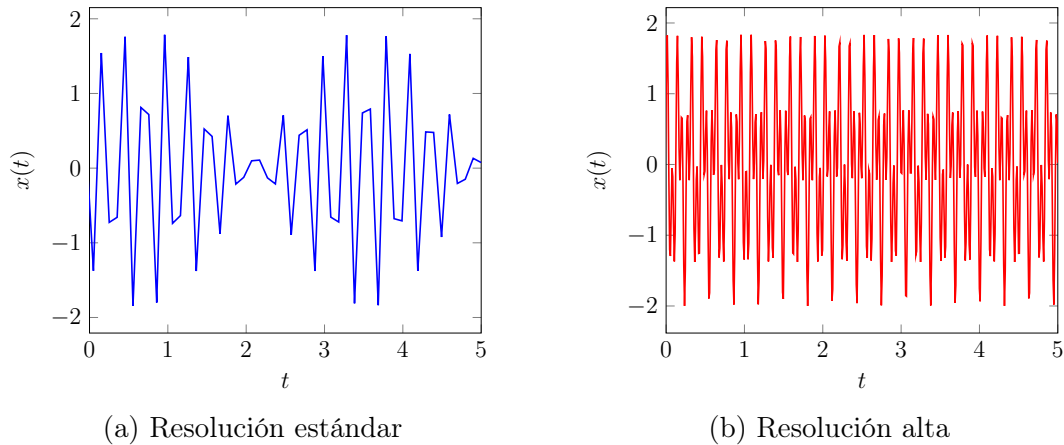


Figura 1.2: Comparación de una señal con diferente cantidad de muestras

Bibliografía

- [1] Alan V. Oppenheim y Alan S. Willsky. *Signals and Systems*. Pearson Education, 2009.
- [2] John Smith y Li Wang. «ECG Signal Processing Using Deep Learning». En: *IEEE Transactions on Biomedical Engineering* 67.5 (2020), págs. 1234-1243.