

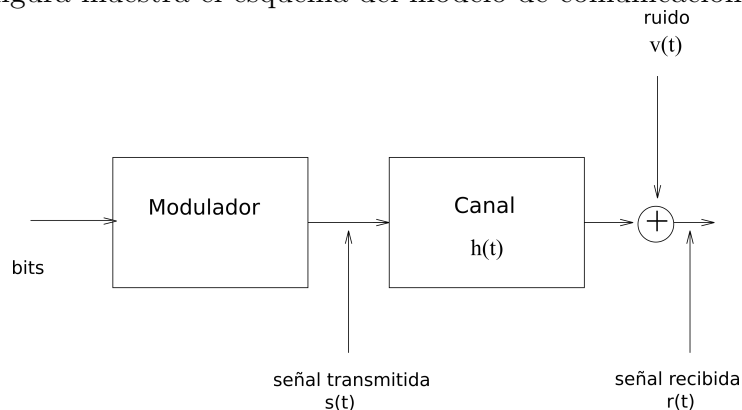
Gestión de Infraestructuras

Módulo I - Práctica 1

Simulación de un sistema de transmisión digital

La práctica consistirá en la simulación de un sistema de comunicación muy sencillo que realiza la transmisión de bits empleando pulsos positivos o negativos. Además, se simulará el efecto del ruido que introducen los canales para ello se representará la señal recibida y se calculará la probabilidad de error tanto real como teórica.

La siguiente figura muestra el esquema del modelo de comunicación.



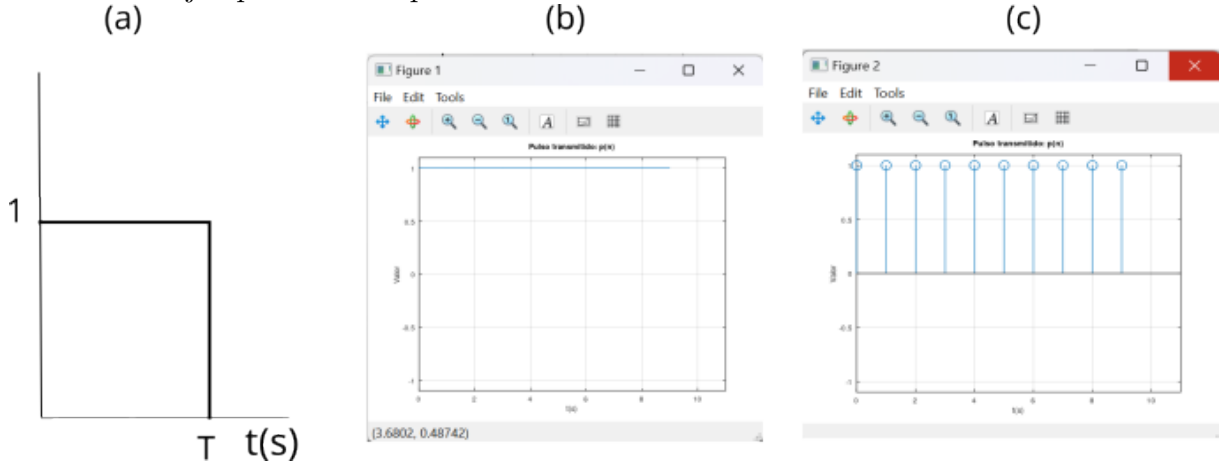
En cada sesión de prácticas se desarrollará una parte de este sistema, por lo que es necesario **terminar cada parte antes de la siguiente sesión de prácticas**.

El programa `practicatransmision.m` contiene el esquema de toda la práctica que usted tiene que ir modificando en cada sesión. Por lo tanto, es importante que lo vea con calma antes de comenzar. Este código está listo para ser ejecutado en Octave o Matlab.

La evaluación se realizará en horas de teoría de la semana del 14 de octubre. La prueba se realizará en papel y consistirá en modificar el código, ejecutarlo con unos parámetros concretos, representar gráficamente alguna de las señales que genera el código y contestar a algunas cuestiones sobre los resultados obtenidos. En el caso de la representación gráfica, se pedirá que **ajuste** los ejes a unos valores concretos. En el caso de la probabilidad de error, se pedirá que **redondee** el valor resultante.

Sesión 1: Modulación

El modulador transforma cada bit en una forma de onda. Es importante que tenga en cuenta que un sistema de comunicación real utilizaría pulsos de tiempo continuo, pero con un lenguaje de programación (como Octave o Matlab) solo podemos generar pulsos de tiempo discreto. Por lo tanto, estamos aproximando los pulsos de tiempo continuo con una señal de tiempo discreto. Por ejemplo, un sistema de transmisión utilizaría un pulso continuo de la forma $p(t) = u(t) - u(t - T)$ siendo T el periodo de símbolo. En la práctica, lo que se está utilizando es un pulso discreto $p(n) = u(n) - u(n - N)$. La siguiente figura muestra un ejemplo de esta aproximación.



Observe que las figuras (b) y (c) corresponden al mismo pulso en tiempo discreto, lo único que cambia es la función que se ha empleado para representarla. En la figura (b) se ha utilizado plot y en (c) se ha utilizado stem.

En esta práctica se emplearán pulsos de duración finita N , pero muchos sistemas reales utilizan pulsos de mayor duración. La forma de estos pulsos puede ser muy variada: pulso rectangular, pulso de Manchester, pulso coseno alzado, pulso de Nyquist, etc.

Al tratarse de una señal en tiempo discreto, la energía de pulsos de duración N se calcula con la siguiente expresión

$$E_p = \sum_{n=0}^{N-1} p(n)^2 \quad (1)$$

En un sistema PAM binario (2-PAM), el proceso de modulación asigna un pulso positivo a 0 y un pulso negativo a 1. Esto quiere decir que cada bit se transmite durante N segundos (duración del pulso) y, por ello, **la energía de bit coincide con la energía del pulso**. Además, este parámetro determina la velocidad de bit y de símbolo, ya que en un sistema 2-PAM la velocidad de bit es la misma que la velocidad de símbolo que, a su vez, es la inversa del periodo de símbolo.

La señal de salida del modulador se puede escribir como sigue

$$s(t) = \sum_{k=0}^{L-1} A_k p(t - kN) \quad (2)$$

donde N es la duración del pulso y A_k es un número que, para el sistema binario, toma valores ± 1 .

En esta sesión de prácticas, debe realizar los siguiente:

1. Entienda el código que se ha proporcionado y ejecútelo. Observe que el código genera un pulso rectangular con duración N (comienza en 0 y termina en $N - 1$) y amplitud 1.

2. Añada las líneas de código que permita generar los siguientes pulsos:

- $p(n) = u(n) - u(n - N/2)$. En este caso, el pulso sigue teniendo duración N , pero solamente toma valores 1 desde 0 hasta $N/2 - 1$.
- $p(n) = u(n) - 2u(n - N/2) + u(n - N)$. Este pulso, conocido como pulso de Manchester, toma valor 1 de 0 a $N/2 - 1$ y valor -1 desde $N/2$ hasta $N - 1$.

CONSEJO: Utilice if para que pueda elegir el pulso que desea utilizar. Esto le será muy útil para hacer pruebas con distintos tipo de pulso.

3. Añada el código para generar un pulso de duración N distinto a los anteriores.

4. Añada las líneas de código para calcular la energía del pulso utilizando la expresión (1).

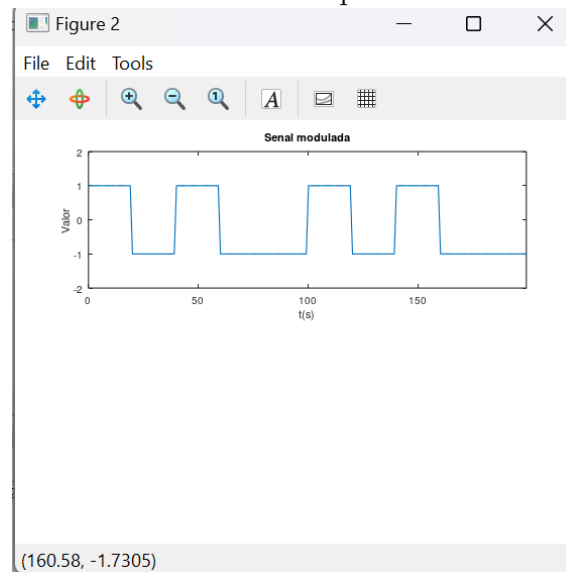
5. Añada las líneas de código que permite modular una secuencia de L bits: la señal a transmitir se creará asignando un pulso con $A_k = 1$ a un bit 0 y un pulso con $A_k = -1$ a un bit 1.

CONSEJO: Para obtener la señal correspondiente a los L bits, puede utilizar un bucle y emplear la concatenación de vectores: si tenemos dos vectores a y b , la concatenación se realiza como $[a \ b]$.

6. Represente el pulso y la señal modulada. Ajuste para que en la gráfica se vea la señal para los 10 primeros bits y el eje de las ordenadas vaya de -2 a 2.

CONSEJO: Para ajustar la gráfica e identificarla correctamente, le resultará útil utilizar los comandos axis, xlabel, xtext, title, ... vistos en el tutorial.

La siguiente gráfica muestra el resultado que debe obtener.



PREGUNTA: ¿Cuántos símbolos se han representado en esa figura?

PREGUNTA: ¿A qué secuencia de bits transmitidos corresponde esa figura?

7. Ejecute el código para los todos los pulsos. En cada ejecución, compare el valor de E_p con el visto en la clase de teoría e interprete las gráficas.

PREGUNTA: ¿Qué pulso tiene mayor energía?

PREGUNTA: Todos los pulsos son de energía finita, ¿cuál es su potencia media?

Sesión 2: Transmisión por canales con ruido

Consideraremos que la señal modulada es transmitida a través de un canal con ruido, de forma que la señal recibida tiene la siguiente expresión

$$r(t) = s(t) + v(t) \quad (3)$$

donde $v(t)$ es la componente de ruido que introduce el canal. En particular, se suele asumir la existencia de ruido aditivo con distribución gaussiana de media nula y potencia $N_0/2$.

La relación entre la energía de bit y la del ruido E_b/N_0 es un parámetro que determina la calidad de la transmisión. Como se vio en la Sesión 1, en un sistema 2-PAM, la energía de bit E_b coincide con la energía del pulso E_p , así que, para el mismo tipo de pulso, el factor que determina la calidad de la transmisión es la potencia del ruido N_0 .

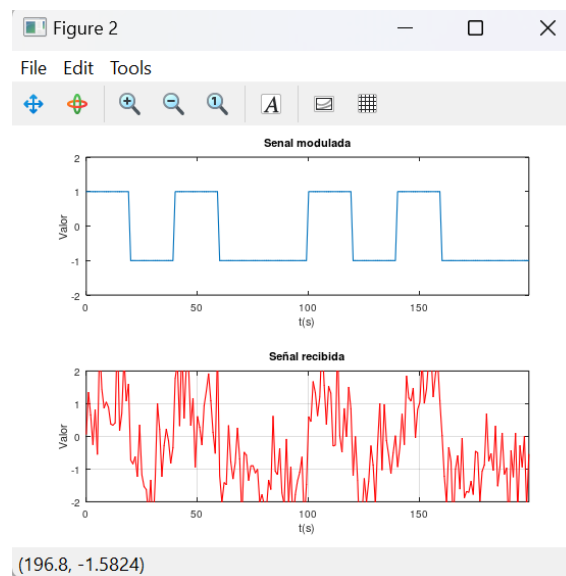
En esta sesión de prácticas, debe actualizar el código como sigue:

1. Incluya el parámetro $E_b N_0$ (dB) en su código.
2. Añada las siguientes líneas para generar el ruido.

```
%Cambio a unidades naturales y calculo de No
EbNo=10^(EbNodB/10);
No=Eb/EbNo;
ruido=sqrt(No/2)*randn(1,N*L);
```

PREGUNTA: ¿Cuál es la unidad de la N_0 ?

3. Añada la línea de código para sumar el ruido a la señal modulada (señal generada en la Sesión 1). Llamaremos s_{rec} a esta señal.
4. Represente la señal con ruido en la misma figura que la señal modulada. La siguiente figura muestra la gráfica que debe obtener para un pulso rectangular y $E_b/N_0 = 10$ dB.



PREGUNTA: ¿Cuántos símbolos se han representado en esa figura?

PREGUNTA: ¿A qué secuencia de bits transmitidos corresponde esa figura?

5. Ejecute el código con otros valores de E_b/N_0 . Por ejemplo, $E_b/N_0 = 3$ dB, $E_b/N_0 = 5$ dB y $E_b/N_0 = 100$ dB. Es importante que se fije en el valor de la E_b/N_0 en dB y en unidades naturales, y en el valor de la potencia del ruido N_0 .

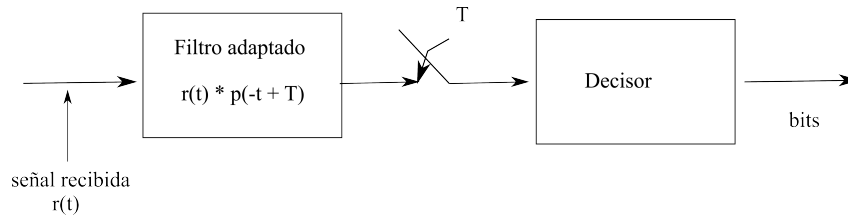
PREGUNTA: ¿Existe mayor o menor distorsión al aumentar la E_b/N_0 ?

PREGUNTA: ¿Qué sucede con la potencia del ruido al aumentar E_b/N_0 ?

6. Ejecute el código para todos los pulsos.
7. Ejecute el código cambiando los valores de N y L .

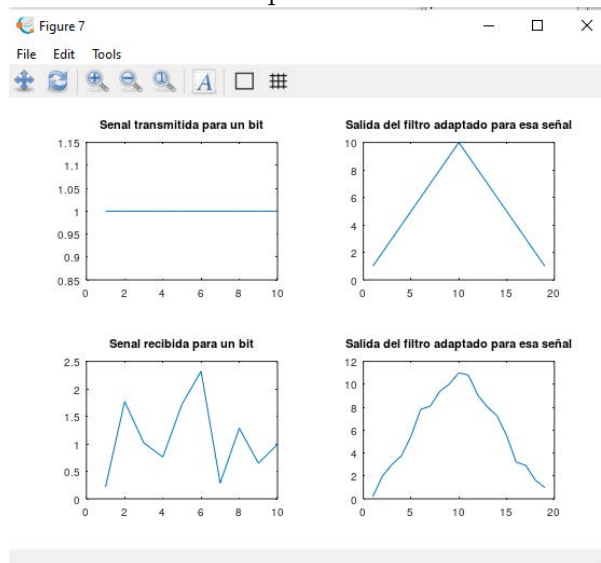
Sesión 3: Detección y probabilidad de error

En esta parte de la práctica se implementará el esquema que se muestra en la siguiente figura.



El filtro adaptado es un método muy simple que permite estimar los bits transmitidos a partir de la señal recibida. Se trata de un sistema que realiza la convolución entre cada segmento de tamaño N de la señal recibida y $p(-t + T)$. Posteriormente, se toma la muestra ubicada en la posición T y se compara con un umbral.

La siguiente figura muestra un ejemplo de cómo funciona. Recordemos que el sistema 2-PAM asigna un pulso con valor 1 si se transmite un bit 0 y un pulso con valor -1 si se transmite un bit 1. Por ejemplo, en la primera fila se puede ver la señal transmitida para un bit 0 y la señal a la salida del filtro adaptado en ausencia de ruido. Claramente, tomando el valor en T , podemos determinar que se ha transmitido un bit 0. En la parte inferior de la figura, se muestran esas señales pero para una $E_b/N_0 = 10$ dB. Se observa una gran distorsión en la señal recibida, pero el valor en T de la salida del filtro adaptado sigue siendo claro y permite determinar que se ha transmitido un bit 0.



Realice los siguientes cambios en su código:

1. Añada las siguientes líneas que implementa el filtro adaptado seguido de un detector que compara cada valor con 0.

```
pulsoinv = pulso(N:-1:1);
ind = 1;
for k = 1:N:L*N-1
    s_conv = conv(s_rec(k:k+N-1),pulsoinv);
    s_muest = s_conv(N);
    bits_rec(ind) = s_muest <= 0;
    ind = ind + 1;
```

```
end;
```

Observe que la línea $bits_rec(ind) = s_muest \leq 0$; es equivalente a la siguiente instrucción condicional:

```
if s_muest > 0
    bits_rec(ind) = 0;
else
    bits_rec(ind) = 1;
end;
```

2. Ahora comparando la variable $bits_recuperados$ con $bits$ (es decir, los bits originales), podemos calcular la probabilidad de error real del sistemas. Para ello, incluya el siguiente código:

```
pe_real = mean(bits_rec ~= bits);
```

3. Para un sistema 2-PAM, determine la probabilidad de error teórica añadiendo la siguiente línea:

```
pe_teo = erfc(sqrt(EbNo))/2;
```

En este caso, $EbNo$ está expresada en unidades naturales (no en dB).

Pruebas:

1. Simulación 1: Haga una simulación con los siguientes parámetros con un pulso rectangular, $N = 10$, $L = 1000$ y $EbNodB = 10$. Debería obtener estos resultados:

```
pe_real = 0
pe_teo = 3.8721e-06
```

La probabilidad de error teórica debe ser la misma. En el caso de la probabilidad de error real podría haber una pequeña diferencia. La probabilidad de error teórica y real no deben coincidir entre sí, pero deben tener el mismo orden de magnitud. Por ejemplo, un valor del orden de 10^{-6} significa que se comete 1 error cada 1000000 bits, que es muy bajo. En concreto, para $L = 1000$ bits, nuestro sistema no ha cometido ningún error.

Para un sistema que tenga una probabilidad de error teórica de orden 10^{-6} , la simulación tendría que usar un valor de L mayor o igual a 1000000.

IMPORTANTE: El valor de L tiene que ser alto para conseguir una buena precisión. Pero en las gráficas solamente debe visualizar un número pequeño de símbolos (por ejemplo, 5, 10 o 20) para distinguir claramente los símbolos transmitidos.

2. Simulación 2: Repita la simulación anterior con $Eb/N_0 = 5$ dB. Resultados:

```
pe_real = 5.0000e-03
pe_teo = 5.9539e-03
```


Un valor del orden de 10^{-3} significa que se comete 1 error cada 1000 bits. La probabilidad de error real indica que el sistema ha cometido 5 errores en la secuencia de $L = 1000$ bits, que es muy parecido al valor teórico.

Ejecute varias veces el programa. Fíjese que la probabilidad de error teórica es siempre la misma, pero la real puede que cambie.

PREGUNTA: ¿La probabilidad de error aumenta o disminuye al aumentar la E_b/N_0 ?

3. Simulación 3: Haga pruebas cambiando el periodo del pulso.

PREGUNTA: ¿La probabilidad de error depende del periodo del pulso? Fíjese en la teórica.

PREGUNTA: ¿Cuál es la velocidad de bit para esos pulsos? ¿Qué relación existe entre la velocidad de bit y la probabilidad de error?

4. Simulación 4: Haga pruebas cambiando el tipo de pulso.

PREGUNTA: ¿La probabilidad de error depende del tipo de pulso? Fíjese en la teórica.

PREGUNTA: ¿Cuál es la velocidad de bit para esos pulsos? ¿Qué relación existe entre la velocidad de bit y la probabilidad de error?