

Taller de simulación: Modulación digital

Introducción

Los esquemas de modulación digital modernos permiten comunicaciones eficaces y eficientes, tanto banda-base como paso-banda. Técnicas como modulación de amplitud de pulso (PAM, por sus siglas en inglés) y de amplitud en cuadratura (QAM, por sus siglas en inglés) son de gran relevancia para la mayoría de sistemas de comunicaciones actuales.

La simulación de las señales involucradas en esquemas de modulación digital paso-banda es relativamente complicada, pero mediante el uso del esquema banda-base equivalente se puede obtener resultados similares de forma más sencilla. Al utilizar este esquema, el canal debe ser también modificado a una versión banda-base equivalente.

Descripción

Este taller busca desarrollar destrezas de software para trabajar con modulación digital y ejemplificar algunos conceptos relacionados con el estudio de las señales involucradas en este tipo de modulación. Así, el taller consiste en utilizar herramientas para estudiar esquemas de modulación PAM y QAM en banda-base con canales ruidosos y, opcionalmente, filtros que permiten reducir la interferencia inter-simbólica.

El taller está diseñado para utilizar *Matlab* como herramienta principal, pero es posible desarrollarlo con herramientas libres como librerías de procesamiento digital de señales y comunicaciones para *C*, *C++* o *Python*.

Consideraciones generales

- El taller debe realizarse en grupos de 2 personas.
- Se promueven las consultas al profesor y el asistente; sin embargo, deben ser puntuales y claras.
- Los resultados del taller deben ser presentados en un reporte breve *compuesto* en formato PDF.
- Todo el código fuente utilizado para desarrollar el taller (los archivos `.m`, en el caso de Matlab) debe adjuntarse al reporte.
- Se castigará severamente cualquier intento de copia en el código o los reportes.

1. Secuencia de bits de entrada

Considere una secuencia de bits b proveniente de una imagen .png, .jpg ó .bmp (de un tamaño no mayor a 640x480 pixeles).

Trabajo de simulación

1. Obtenga una matriz de pixeles img que represente la imagen a color.¹
2. Convierta img a escala de grises y visualícela.²
3. Convierta img a una matriz de bits y reacomódelo en una sola secuencia b .³

2. Modulación 4-PAM

Suponga que la secuencia de bits de entrada b_n se modula según 4-PAM. Considere que se utiliza un tiempo de símbolo $T_{sim} = 2$ ms y un pulso rectangular $g(t) = \text{rect}\left(\frac{t}{T_{sim}}\right)$.

Trabajo previo

1. Determine la cantidad de bits/símbolo k y la cantidad de símbolos posibles M .
2. Si se utiliza una distancia entre símbolos de 2, determine los símbolos posibles a_i .
3. Obtenga una asignación de bits a símbolos según *gray*.
4. Esboce la señal modulada $x(t)$ para los primeros símbolos de a_n .
5. Esboce el espectro de magnitud de $x(t)$.

Trabajo de simulación


1. Realice la modulación banda-base de los bits b según 4-PAM con asignación *gray*, para obtener un vector de símbolos banda-base a .⁴
2. Observe los primeros símbolos de a y verifique que corresponden a los primeros bits de b .
3. Obtenga el vector de la señal modulada x correspondiente a los símbolos a y su FFT X .⁵
4. Grafique x en el tiempo y su espectro de magnitud (en la frecuencia) para los primeros símbolos de a .⁶
5. Comente sobre los resultados obtenidos.


3. Modulación 16-QAM (en banda-base)


Suponga que la secuencia de bits de entrada se modula según 16-QAM. El equivalente banda-base de la señal modulada $s(t)$ son las señales $x_I(t)$ y $x_Q(t)$, que se pueden considerar como el resultado de una modulación PAM en cuadratura. Considere que para estas señales se utiliza un tiempo de símbolo $T_{sim} = 2$ ms y un pulso rectangular $g(t) = \text{rect}\left(\frac{t}{T_{sim}}\right)$.


Trabajo previo


1. Determine la cantidad de bits/símbolo k y la cantidad de símbolos posibles M .
2. Si se utiliza una distancia entre símbolos de 2, dibuje el diagrama de constelación.

¹  Considere la función `imread()`.

²  Considere las funciones `rgb2gray()` y `imshow()`.

³  Considere las funciones `dec2bin()` y `reshape()`.

⁴  Considere la función `pammod()`. Observe que para esta función los símbolos de entrada deben estar en formato decimal, por lo que las funciones `reshape()` y `bin2dec()` pueden ser útiles.

⁵  Considere la función `rectpulse()`.

⁶ Considere que, según lo indicado, la frecuencia de muestreo es $F_s = \frac{N_{xs}}{T_{sim}}$ y la cantidad total de muestras es N_{xs} veces la cantidad total de símbolos.

3. Obtenga una asignación de bits a símbolos según *gray*.
4. Esboce las señales moduladas en banda-base $x_I(t)$ y $x_Q(t)$ para los primeros símbolos de a_n .
5. Esboce el espectro de magnitud de $x_I(t)$ y $x_Q(t)$.

Trabajo de simulación

1. Realice la modulación banda-base de los bits b según 16-QAM con asignación *gray*, para obtener un vector de símbolos banda-base a .⁷
2. Observe los primeros símbolos de a y verifique que corresponden a los primeros bits de b .⁸
3. Grafique el diagrama de constelación.
4. Obtenga el vector de la señal modulada x correspondiente a los símbolos a .^{9,10}
5. Grafique x para los primeros símbolos de a y su espectro de magnitud.
6. Comente sobre los resultados obtenidos.

4. Canal AWGN (banda-base)

Considere un canal con ruido blanco aditivo gaussiano (AWGN, por sus siglas en inglés) $n(t)$ en el que $x^*(t) = x(t) + n(t)$.

Trabajo previo

1. Esboce $x^*(t)$ y la magnitud de su espectro $X^*(f)$.
2. Comente sobre cómo cambian ambos con la relación señal-a-ruido (SNR, por sus siglas en inglés) en el receptor.

Trabajo de simulación

1. Agregue ruido blanco aditivo gaussiano a la señal banda-base x , generando la señal recibida x_R .¹¹
2. Grafique x_R para los primeros símbolos de a y su espectro de magnitud. Compare el resultado con las gráficas de x y X .

5. Demodulación 16-QAM banda-base

Trabajo previo

1. Explique cómo se obtiene y_n^* a partir de $x^*(t)$ con un receptor de *integración y descarte*.
2. Muestre los *límites de decisión* para un esquema de modulación 16-QAM.

Trabajo de simulación

1. Realice la recepción banda-base de la señal recibida x_R para obtener un vector de muestras y_R .¹²
2. Observe las primeras muestras de y_R y compárelas con los primeros símbolos de a .
3. Realice la decisión sobre las muestras y_R y asigne los símbolos recibidos a_R a bits según la asignación

⁷ Considerare la función `qammod()`. Observe que para esta función los símbolos de entrada deben estar en formato decimal.

⁸ Observe que Matlab utiliza una notación compleja para los símbolos en cuadratura, es decir, a_{nI} se representa como `real(a(n))` y a_{nQ} como `imag(a(n))`.

⁹ Considerare la función `rectpulse()`.

¹⁰ Observe que Matlab utiliza una notación compleja para las señales en cuadratura, es decir, $x_I(t)$ se representa como `real(x)` y x_Q como `imag(x)`.

¹¹ Considerare la función `awgn()` con la opción `'measured'`, utilizando un SNR (en decibels) `SNRdB` apropiado para apreciar el efecto del ruido pero que no afecte significativamente el proceso de demodulación.

¹² Considerare la función `intdump()`.

gray utilizada en el modulador, para obtener una secuencia de bits recibidos *bR*.¹³

4. Compare los bits recibidos *bR* con los bits de entrada *b* y comente sobre los resultados obtenidos.

6. Secuencia de bits de salida

Trabajo de simulación

1. Reacomode la secuencia de bits recibidos *bR* a una matriz de valores decimales *imgR* con las mismas dimensiones de la matriz de pixeles *img*.¹⁴
2. Convierta *imgR* en una imagen y visualícela.¹⁵
3. Compare las imágenes *img* y *imgR* y comente sobre el desempeño del esquema de modulación. Intente utilizar otros valores de SNR para observar su efecto en la imagen recibida.

7. Medidas de desempeño

Trabajo previo

1. Investigue sobre cómo estimar el ancho de banda *ocupado* (en el que se encuentra el 99 % de la potencia de la señal) para un pulso rectangular.
2. Estime el ancho de banda *ocupado* por la señal modulada $x(t)$, B_x .
3. Investigue sobre la relación entre $\frac{E_b}{N_0}$ y *SNR*.
4. Obtenga la curva de razón de errores de bits (BER, por sus siglas en inglés) contra $\frac{E_b}{N_0}$ para 16-QAM.

Trabajo de simulación

1. Estime el ancho de banda *ocupado* por x , B_x .¹⁶
2. Estime el valor de $\frac{E_b}{N_0}$ como $E_b N_0 = \text{SNR} * T_{\text{sim}} * F_s / k$.¹⁷
3. Cuente la cantidad de bits diferentes entre *bR* y *b* y determine el BER resultante.
4. Repita el cálculo de BER para valores de $E_b N_0$ desde 0 dB hasta 20 dB en intervalos de 5 dB.
5. Obtenga la gráfica de *BER* contra $\frac{E_b}{N_0}$ para esta simulación de 16-QAM.

¹³ Considerare la función `qamdemod()`. Observe que para esta función los símbolos de salida están en formato decimal, por lo que las funciones `reshape()` y `bin2dec()` pueden ser útiles.

¹⁴ Considerare las funciones `reshape()` y `bin2dec()`.

¹⁵ Considerare las funciones `mat2gray()` y `imshow()`.

¹⁶ Considerare la función `obw()`.

¹⁷ Observe que el SNR está expresado en unidades lineales.

8. Reto: Pulso de coseno elevado

Suponga que la secuencia de bits de entrada se modula según 16-QAM. Considere que para estas señales se utiliza un tiempo de símbolo $T_{\text{sim}} = 2 \text{ ms}$ y un pulso $p(t)$ de coseno elevado.

Trabajo de simulación

1. Realice la modulación banda-base de los bits b según 16-QAM con asignación *gray*, para obtener un vector de símbolos banda-base a .
2. Escoja una cantidad de muestras para representar cada símbolo N_{xs} en la señal modulada. Se propone $N_{\text{xs}} = 8$.
3. Obtenga la respuesta al impulso de un filtro raíz de coseno elevado g con un factor de reducción $\beta = 0.25$ y que se extienda por el tiempo de 16 símbolos.¹⁸
4. Grafique la respuesta al impulso g obtenida y su espectro de magnitud.

Para generar un pulso $g(t)$ de raíz de coseno elevado, los símbolos a_n se representan como impulsos y se pasan por un filtro raíz de coseno elevado.

5. Aumente la frecuencia de muestreo de a agregando $N_{\text{xs}}-1$ ceros entre cada uno, para producir una señal de impulsos y .¹⁹
6. Realice la convolución de la señal de impulsos y con la respuesta al impulso del filtro g para obtener la señal filtrada x .²⁰
7. Para compensar por el retardo que introduce el filtrado, es necesario descartar las primeras y últimas muestras del resultado de la convolución. Según los valores propuestos, es necesario descartar las primeras y últimas 24 muestras.²¹
8. Grafique x para los primeros símbolos de a y su espectro de magnitud.

Considere un canal con ruido blanco aditivo gaussiano (AWGN, por sus siglas en inglés) $n(t)$ en el que $x^*(t) = x(t) + n(t)$.

9. Agregue ruido blanco aditivo gaussiano a la señal banda-base x , generando la señal recibida x_R .
10. Grafique x_R para los primeros símbolos de a y su espectro de magnitud.
11. Defina la respuesta al impulso de un filtro en el receptor h *acoplado* al del transmisor (es decir, $h=g$).
12. Filtre la señal recibida x_R con el filtro definido por h de la misma forma que en el transmisor para obtener una señal filtrada y_R .
13. Muestree la señal filtrada y_R cada N_{xs} muestras para obtener un vector de muestras y_{Rn} .²²
14. Realice la decisión sobre las muestras y_{Rn} y asigne los símbolos recibidos a_R a bits según la asignación *gray* utilizada en el modulador, para obtener una secuencia de bits recibidos b_R .
15. Compare los resultados obtenidos para el pulso de coseno elevado con los obtenidos para el pulso rectangular.



¹⁸ Considere la función `rcosdesign()`.

¹⁹ Considere la función `upsample()`.

²⁰ Considere la función `conv()`.

²¹ Considere la expresión $x = x(\text{ret}+1:\text{end}-\text{ret})$ con $\text{ret}=24$.

²² Considere la expresión $y_{Rn} = y_R(1:N_{\text{xs}}:N)$.