

Laboratorio Digital Interactivo

By:

Mariangela Mezoa

Laboratorio Digital Interactivo

By:

Mariangela Mezoa

Translated By:

Mariangela Mezoa

Online:

< <http://cnx.org/content/col11274/1.1/> >

C O N N E X I O N S

Rice University, Houston, Texas

This selection and arrangement of content as a collection is copyrighted by Mariangela Mezoa. It is licensed under the Creative Commons Attribution 3.0 license (<http://creativecommons.org/licenses/by/3.0/>).

Collection structure revised: February 9, 2011

PDF generated: April 22, 2011

For copyright and attribution information for the modules contained in this collection, see p. 111.

Table of Contents

1	1. Introducción a los Sistemas de Comunicaciones Digitales	1
2	2. Introducción a LabVIEW y teoría básica de PCM (Pulse Code Modulation)	5
3	3. Ortogonalización Gram-Schmidt y teoría básica de las Constelaciones	13
4	4. Códigos de Línea	19
5	5. Modulaciones Binarias: Teoría y simulación en LabVIEW	31
6	6. Modulaciones M-arias	41
7	7. Introducción del ruido en los sistemas de Comunicaciones Digitales: Detección Óptima y Probabilidad de Error	55
8	8. Detección no Coherente para modulaciones OOK y FSK	65
9	9. Interferencia Intersimbólica (ISI)	73
10	10. Codificación de Canal: Código Hamming y Código Convolutivo	85
	Glossary	109
	Index	110
	Attributions	111

Chapter 1

1. Introducción a los Sistemas de Comunicaciones Digitales¹

1.1 INTRODUCCIÓN A LOS SISTEMAS DE COMUNICACIONES DIGITALES

González C. Y. Venuska

Mezoa R. Mariangela

Resumen

Este módulo sirve como preámbulo a las comunicaciones, específicamente en el ámbito digital. Se explicará a través de un diagrama de bloques teórico el procedimiento necesario para transmitir exitosamente un mensaje, considerando factores como la codificación del mensaje, codificación del canal y la inclusión del ruido en el sistema.

La comunicación, como se sabe, es la transferencia de información desde un punto considerado como emisor hasta otro punto considerado como receptor. Dicha transferencia se hace a través de un canal, y se estima que el proceso sea seguro y confiable, de modo tal que la información desde la fuente sea recibida en su totalidad.

Ahora bien, es el **sistema de Comunicación** el conjunto de subsistemas y mecanismos que permite el enlace entre el emisor y el receptor. Ya sea entre humanos o entre máquinas, se deben cumplir ciertos conceptos para asegurar la transmisión y recepción óptima del mensaje, pero, en general, se toman en cuenta tres factores principales que son:

- Un **transmisor**, que toma la información y la convierte en una señal.
- Un **Medio de Transmisión** o Canal Físico que 'lleva' la señal.
- Un **Receptor**, que toma la señal del canal y la convierte de vuelta a información útil.

Es importante resaltar que para las Telecomunicaciones existen dos tipos de señales de comunicación: **Analógicas** y **Digitales**, siendo esta última con la que nos enfocaremos. La señal digital se reconoce porque la información es codificada por medio de un set finito de valores discretos (1's y 0's), mientras que en el sistema Analógico se representa con una señal continua en el tiempo.

A continuación se podrá observar un diagrama de bloques que define paso a paso el procedimiento de envío de información:

¹This content is available online at <<http://cnx.org/content/m35707/1.1/>>.

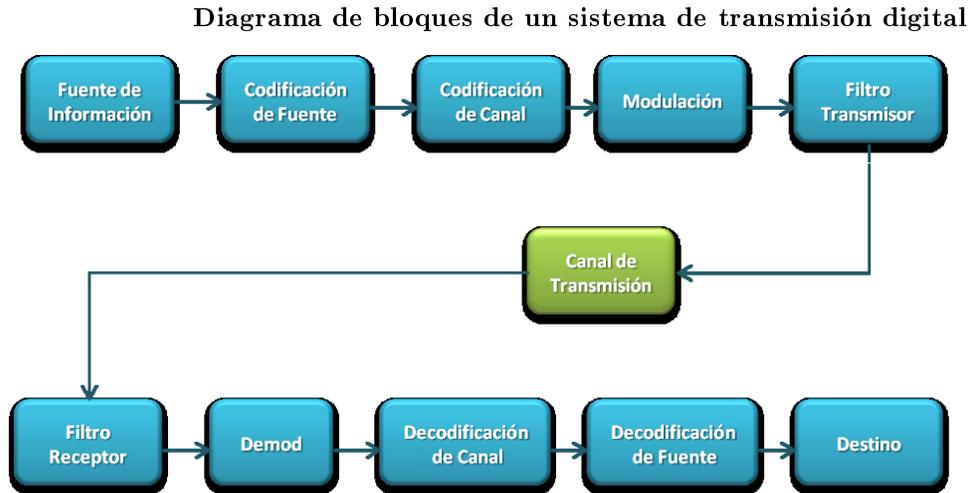


Figure 1.1

1.1.1 Fuente de Información:

Corresponde al procedimiento de conversión del mensaje a transmitir en una señal que se adecue al sistema a usar; dado que es un sistema digital, la fuente genera una señal de tipo digital (binaria). Es común encontrarnos con información a transmitir en formato analógico, por lo que será necesario usar un conversor Analógico/Digital que haga el proceso de muestreo y cuantificación.

1.1.2 Codificación de Fuente:

Encargada de eliminar parte de la redundancia de información ofrecida por el bloque de la fuente. Esto implica una compresión directa del mensaje.

1.1.3 Codificación de Canal:

Capaz de detectar y corregir los errores que se producen en los datos durante el proceso de la transmisión. Dichos errores pueden darse por la existencia de ruido en canal, por lo que este bloque es capaz de introducir redundancia en la cadena de datos, haciendo posible la reconstrucción de la misma lo más aproximado a la secuencia original.

1.1.4 Modulación:

Modular consiste, en el ámbito de las comunicaciones, en la variación de una o más propiedades de una forma de onda periódica de alta frecuencia (conocida como **señal portadora**) con respecto a una **señal moduladora**, que es la que se desea transmitir. Con esto es posible transportar el mensaje dentro de otra señal que puede ser transmitida físicamente a través de un canal pasabanda.

Para la transmisión digital estamos hablando de una señal portadora analógica **Sinusoidal** que es modulada de acuerdo con una cadena de bits. Ahora, dependiendo del parámetro que se decida variar, se generan distintos tipos de modulación. Los fundamentales son (Trabajan con una cadena de 0 y 1's):

- PSK (*Phase-shift Keying*)
- FSK (*Frequency-shift Keying*)
- ASK (*Amplitude-shift Keying*)

Pero, si hablamos de la modulación m-aria o multinivel, se tendrán entonces:

- QPSK (*Quadrature Phase-shift Keying*)
- MPSK
- QAM (*Quadrature Amplitude Modulation*)

1.1.5 Filtro Transmisor:

Al trabajar con un sistema de comunicaciones real, se debe considerar una restricción fundamental: El **Ancho de Banda**. Evidentemente, es muy común que estos sistemas limiten el ancho de banda del canal de transmisión, por lo que es necesario implementar un filtro que se encargue de limitar la señal modulada y que también adecue la potencia de transmisión de la misma.

1.1.6 Canal de Transmisión:

Es el medio de transporte de la señal modulada. En la vida real, el canal de transmisión es un gran contribuyente a la introducción de ruido e interferencias, lo que genera errores inmediatos en el mensaje. Esto se debe a la atenuación que condiciona la potencia recibida (es difícil aumentar la potencia de transmisión sin disminuir la distancia entre la fuente y el destino) y el ancho de banda limitado. Los siguientes módulos profundizarán mejor esta situación por medio de la inclusión de ruido AWGN en el sistema y una posible Interferencia Inter-Simbólica (ISI).

Luego de definir el Canal de Transmisión, deberá hacerse el proceso inverso para que el mensaje ya demodulado, decodificado y convertido de Digital a Analógico llegue al destino.

En los próximos módulos de este curso se desarrollará con más profundidad cada bloque perteneciente al sistema de Comunicaciones Digitales, a través de Laboratorios Digitales elaborados en la herramienta de simulación **LabVIEW**, permitiéndole al usuario enriquecer y reforzar los conocimientos teóricos como prácticos de este sistema.

Chapter 2

2. Introducción a LabVIEW y teoría básica de PCM (Pulse Code Modulation)¹

2.1 INTRODUCCIÓN AL SOFTWARE LABVIEW Y TEORÍA BÁSICA DE PCM (Pulse Code Modulation)

González C. Y. Venuska

Mezoa R. Mariangela

Resumen

A través del presente módulo se pretende dar una introducción muy general al software de simulación LabVIEW de National Instruments, englobando su funcionalidad para el área de las Comunicaciones Digitales. Adicionalmente, se explicará detalladamente la teoría referente a la modulación por código de pulso (PCM), en conjunto con una simulación en LabVIEW que respalde el contenido teórico.

2.1.1 ¿Qué es LabVIEW?

LabVIEW es una herramienta de programación destinada para desarrollar sistemas de pruebas, diseño y control a través de un lenguaje llamado **Lenguaje G**, siendo la G el símbolo que representa el ámbito Gráfico. Pudiéramos decir que estructurar un sistema en este software se aproxima a elaborar un diagrama de flujo, dado que sus herramientas (representadas por íconos gráficos) son intuitivas y secuenciales, permitiéndole hasta al usuario menos experimentado programar proyectos relativamente complejos.

Su ventaja más fuerte, como ya se mencionó, es que la programación es a través de **gráficos** o **bloques**, simplificando así el desarrollo de programas por líneas de código. En adición a lo anterior, su capacidad de desarrollar prácticamente cualquier tipo de programa lo hace compatible con miles de software.

2.1.2 ¿Cómo funciona LabVIEW?

Los programas elaborados en LabVIEW se conocen como **VI**s o *Virtual Instruments* (Instrumentos Virtuales) y están constituidos por dos paneles:

- **Panel Frontal (Front Panel):** Interfaz gráfica con el usuario. Es en esta sección donde el usuario se encargará de introducir datos para luego representar las salidas proporcionadas por el programa.

¹This content is available online at <<http://cnx.org/content/m35710/1.3/>>.

Aquí, es posible la introducción de botones, gráficos e indicadores que varíen los parámetros necesarios para el funcionamiento correcto del mismo.

Panel Frontal

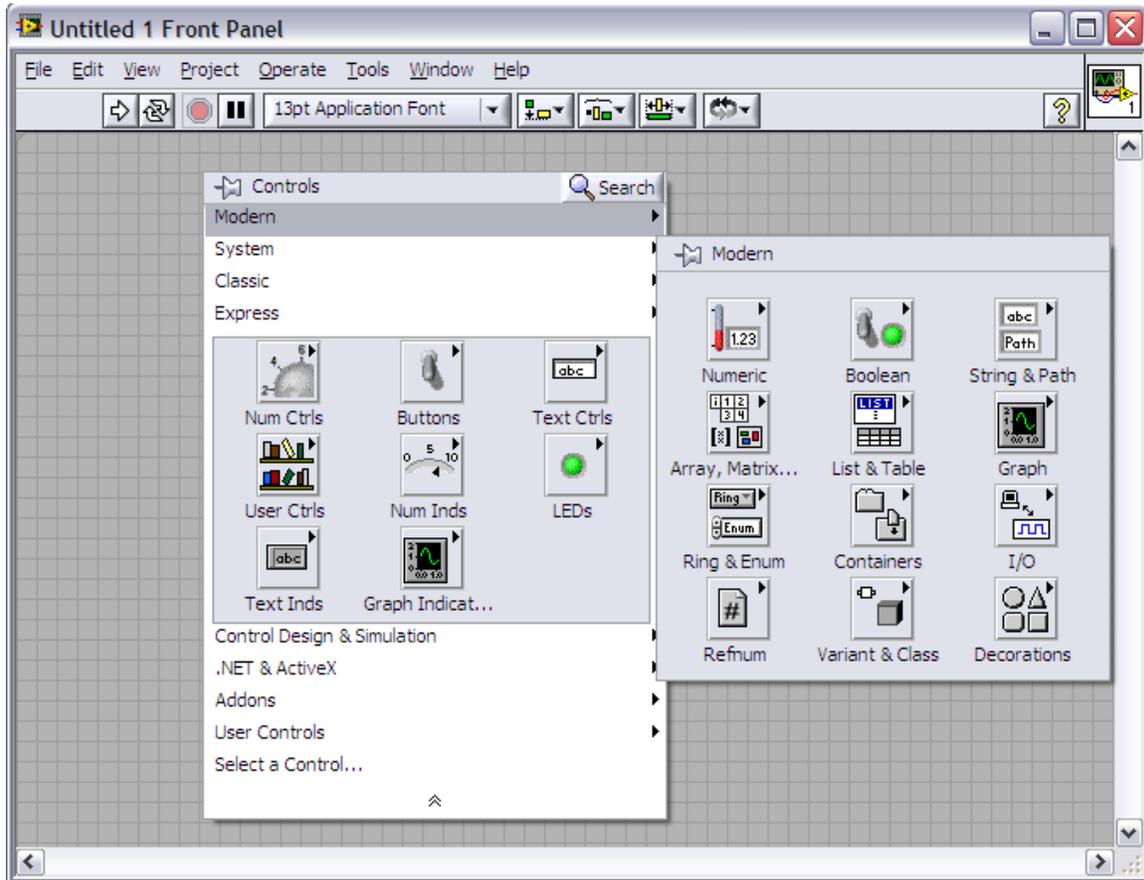


Figure 2.1: Menú desplegado de controles a implementar por el usuario. Como se puede observar en la imagen, se pueden definir desde controles numéricos hasta decoraciones de la interfaz del programa.

- **Diagrama de Bloques (*Block Diagram*):** Es el panel en donde se construye el código fuente del VI. Es el programa como tal, en donde se colocan bloques con cierta funcionalidad que conectados estratégicamente (de acuerdo con las necesidades del usuario) realizarán una determinada función. Cada Control que se coloque en el panel frontal tendrá en el Diagrama de Bloques un terminal que podrá interconectarse con la estructura hecha en el programa, generando así la interacción de información de entrada y salida del VI.

Diagrama de Bloques

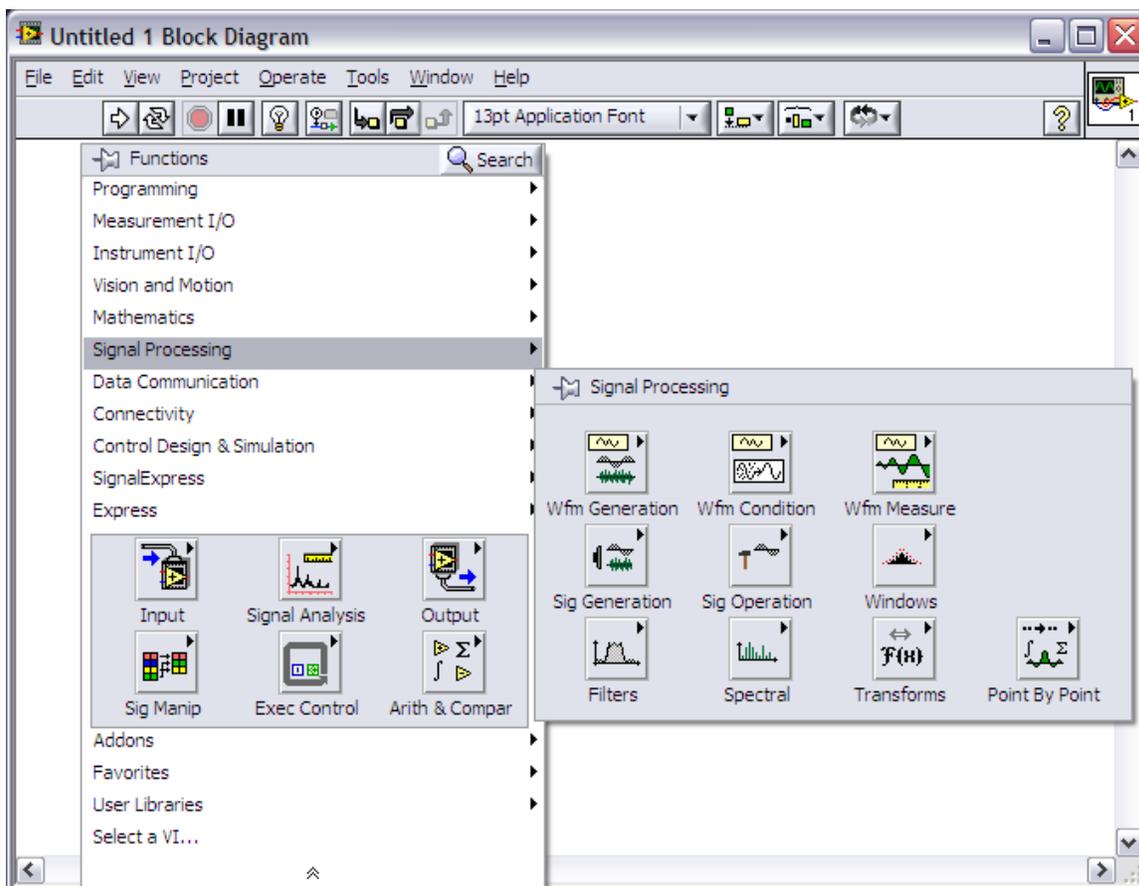


Figure 2.2: Menú desplegado para el Procesamiento de Señales. De acuerdo con el programa que se desee hacer, habrá una gran variedad de funciones que se adapten a la necesidad del usuario.

Para los módulos de este curso, se tomará en cuenta una librería en particular: El **Modulation Toolkit**. Con esta herramienta es posible desarrollar una gran cantidad de programas relacionados con la comunicación digital, dado que posee un listado de bloques que representan ciertos pasos en el procesamiento de una señal. En la imagen siguiente se puede apreciar que engloba 3 aspectos principales:

- Analog
- Digital
- VIs

Modulation Toolkit

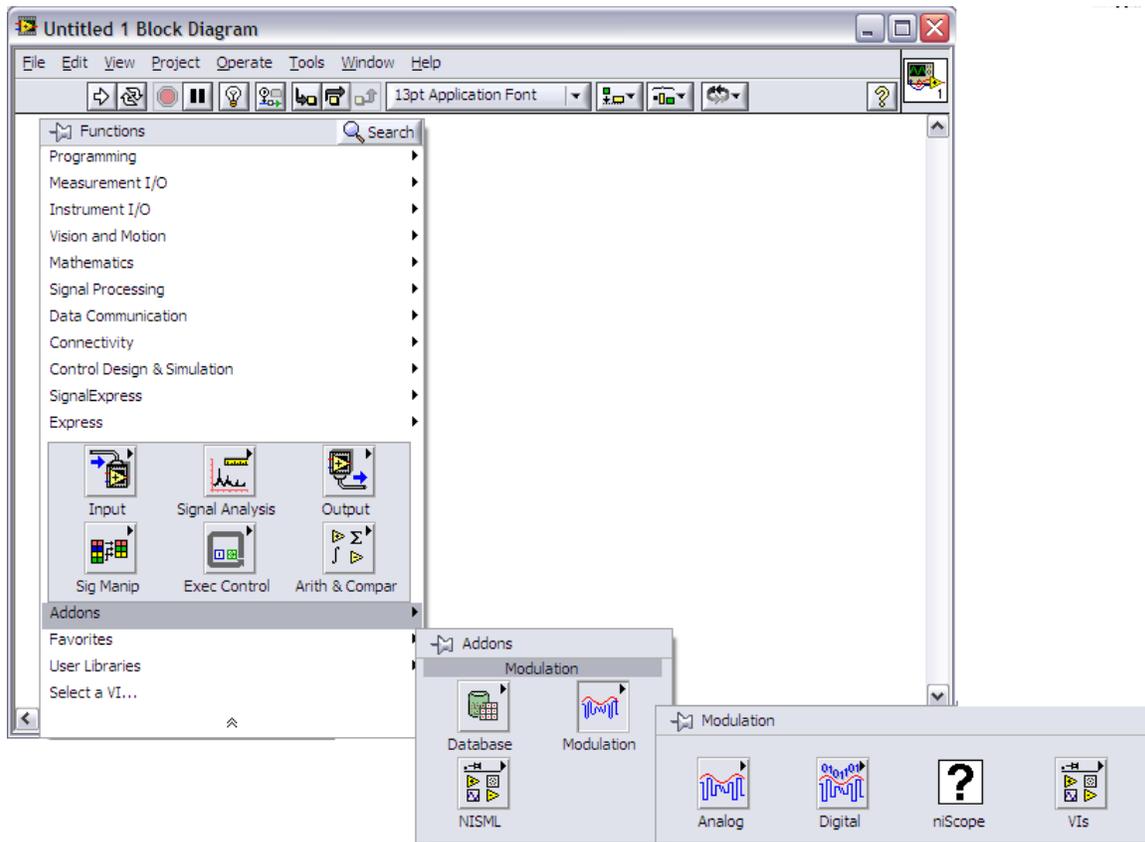


Figure 2.3

En el primero se ofrecen una serie de bloques que se encargan de los procesos de modulación y demodulación de señales analógicas:

Modulation Toolkit – Sección Analógica

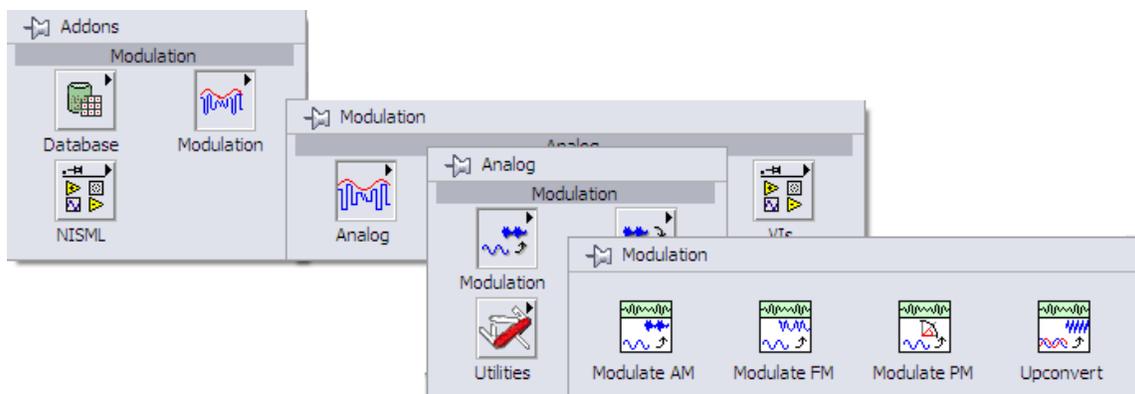


Figure 2.4

Por último, el bloque VIs permite la visualización de determinados gráficos como, por ejemplo, el Diagrama de Constelación:

Modulation Toolkit – Sección VIs

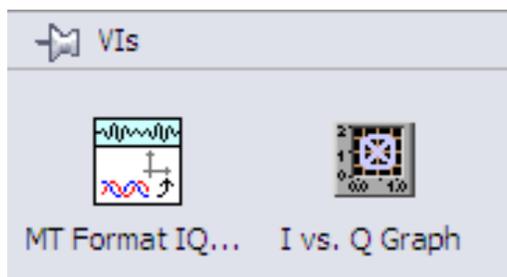


Figure 2.5

Otra de las facilidades de LabVIEW es que tiene una amplia librería de sub-VIs previamente elaborados (apartando el ya explicado Modulation Toolkit) que pueden agregarse a los programas sin necesidad de crearlos desde cero.

Para más información acerca del software y de su uso, consultar los siguientes módulos:

NI LabVIEW Getting Started FAQ²

LabVIEW Environment³

Introduction to the LabVIEW Modulation Toolkit⁴

²"NI LabVIEW Getting Started FAQ" <<http://cnx.org/content/m15428/latest/>>

³"LabVIEW Environment" <<http://cnx.org/content/m12193/latest/>>

⁴"Introduction to the LabVIEW Modulation Toolkit" <<http://cnx.org/content/m18715/latest/>>

Primeros pasos con LabVIEW⁵

Para complementar un poco el funcionamiento general de LabVIEW, el módulo contiene la simulación de un sistema PCM que, tomando una muestra de X segundos de voz muestreada a Y KHz, permite codificar una señal de forma digital (bits o símbolos). Sin embargo, es importante refrescar los conocimientos de este tipo de Modulación.

2.1.3 PCM (PULSE CODE MODULATION)

PCM es una técnica de modulación por codificación digital que es usada para la transmisión digital de información. Más que una modulación, es una forma de codificación de fuente puesto que toma una señal analógica para luego representarla de manera digital.

En esta representación, la magnitud de la señal analógica es muestreada a intervalos uniformes, siendo cada muestra cuantificada al valor más cercano de un rango de niveles digitales.

Esta codificación tiene dos propiedades que determinan la fidelidad de la señal digital con la señal original:

- **La tasa de muestreo**, que es el número de muestras que se toman por segundo;
- **Nivel de Cuantificación**: Rango de valores digitales que puede tomar una muestra.

Para codificar correctamente deben seguirse estos pasos:

2.1.4 1. Muestreo y Retención de la señal analógica

Teniendo una señal en formato analógico que varía en el tiempo, se toma una muestra periódica de la misma, para luego convertir el conjunto de muestras en una serie de pulsos que se puedan transformar con más facilidad a un código PCM binario o m-ario.

El método más común de muestreo y retención se conoce como muestreo de parte plana o muestreo tope plano, donde las amplitudes de los pulsos de muestreo toman el valor de la forma de onda analógica en el punto inicial del intervalo de muestreo y lo conservan durante una porción (*Duty Cycle*) del tiempo de muestreo.

En el muestreo natural, en cambio, los pulsos de muestreo toman los valores de la forma de onda analógica, durante la porción de *Duty Cycle* elegida por el usuario.

Muestreo Natural:

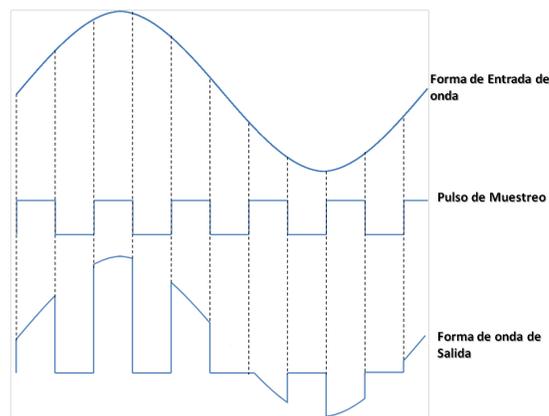


Figure 2.6

⁵ Primeros pasos con LabVIEW <<http://cnx.org/content/col10592/latest/>>

Muestreo Tope Plano:

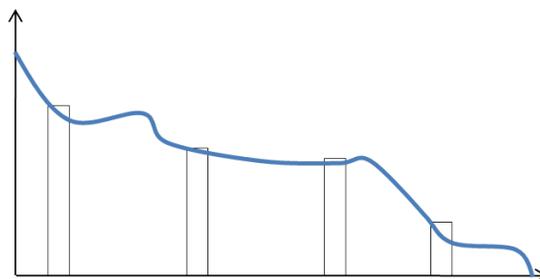


Figure 2.7

2.1.5 2. Cuantificación Digital

En este punto se toma el nivel de voltaje de cada una de las muestras del primer paso y se les atribuye un valor discreto de amplitud, de acuerdo al formato de aproximación dentro de un rango de valores digitales previamente establecido. Cuando se pretende obtener una señal binaria, el número de niveles de cuantificación toma el valor de alguna potencia de 2.

La cuantificación puede ser:

- **Uniforme:** La distancia entre cada nivel de cuantificación siempre es igual.
- **No Uniforme:** Se le asigna un número mayor de niveles a aquellas zonas de voltaje de la señal original más frecuente, y menos niveles en aquellas zonas de voltajes menos frecuentes.

Un detalle que es importante resaltar es que la señal resultante luego de la cuantificación es evidentemente distinta a la señal analógica original. Esta diferencia se conoce como Error o Ruido de Cuantificación, que se modela en la parte de recepción del sistema de Comunicación como un ruido que se le añade a la señal.

2.1.6 3. Codificación Digital

El último paso para este sistema es el de asignarle a los niveles discretos y finitos del paso 2 una traducción al sistema binario, quedando como señal resultante un tren de impulsos digitales (bits). Por ejemplo, si se precisaron 8 niveles de cuantificación, entonces se necesitarán tres bits para la codificación.

Para concluir este módulo, a continuación se muestra una simulación que se encarga de tomar una señal de voz para pasarla por los tres pasos explicados y observar así la señal codificada digitalmente, en conjunto con un video que le permitirá al usuario entender el grupo de bloques usados en el diagrama para construir el sistema PCM, de forma tal que el mismo pueda ir adaptándose al entorno del software LabVIEW.

Para descargar el archivo de audio necesario para la modulación, se debe hacer click en el siguiente enlace: This media object is an audio file. Please view or download it at

<m_b_part_2_2.wav>

El código fuente para la simulación de PCM puede descargarse aquí: [MEDIA OBJECT]⁶
[MEDIA OBJECT]⁷

⁶This media object is a downloadable file. Please view or download it at <pcmULTIMATE.vi>

⁷This media object is a video file. Please view or download it at

Chapter 3

3. Ortogonalización Gram-Schmidt y teoría básica de las Constelaciones¹

ORTOGONALIZACIÓN GRAM-SCHMIDT Y TEORÍA BÁSICA DE LAS CONSTELACIONES

González C. Y. Venuska

Mezoa R. Mariangela

Resumen

Este módulo contiene la teoría correspondiente al método de Ortogonalización Gram-Schmidt aplicado para el proceso de la comunicación digital. Se explicarán los pasos necesarios para generar las bases ortogonales dados ciertos parámetros de la señal. Finalmente, teniendo las bases calculadas, se explicará el procedimiento para hallar la constelación correspondiente.

En matemáticas, el concepto de **Ortogonalidad** está referido al de **Perpendicularidad**. Se dice que dos vectores pertenecientes a cierto espacio vectorial (V) son ortogonales si se cumple la condición de que el producto escalar de ellos **da cero**, es decir:

$$\begin{aligned} \text{Sean } & \quad : \\ x & \in V \\ y & \in V \\ \\ \text{Si } & \quad : \tag{3.1} \\ \langle x, y \rangle & = x \cdot y = 0 \\ \\ \text{Entonces } & \quad : \\ x & \perp y \end{aligned}$$

A partir de un conjunto de vectores linealmente independientes se puede construir un nuevo conjunto de vectores **ortonormales** (Que cumplan con las condiciones de ortogonalidad y norma vectorial). Esto se conoce como el método de **Ortogonalización Gram-Schmidt (G-S)**. Pero, ¿cómo aplicamos este concepto para un sistema de comunicación digital?

¹This content is available online at <http://cnx.org/content/m35711/1.1/>.

3.1 Ortogonalización Gram-Schmidt

Supongamos que se tiene una señal $S_i(t)$ que representa a un símbolo m_i . Se estima que esta señal pase por el receptor que está encargado de obtener cada símbolo de la misma. Sin embargo, es evidente que al pasar por el canal, la señal se contaminará debido a la existencia de ruido en el sistema. En una condición ideal, el resultado sería el siguiente:



Figure 3.1

Al introducir ruido (AWGN) en el sistema, quedaría como sigue:

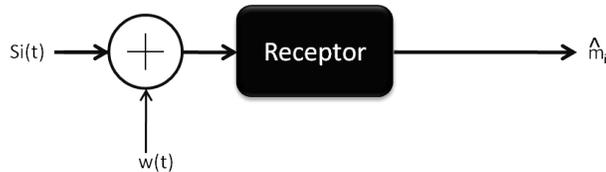


Figure 3.2: Sistema de recepción con introducción de ruido AWGN.

La segunda situación ocasiona que a la salida del receptor no se obtiene el símbolo m_i como tal, más bien se obtiene un **estimado** del símbolo original.

Es en este punto en donde entra el concepto de ortogonalización G-S: La señal $S_i(t)$ puede expresarse en función de un conjunto finito de bases (o vectores) ortonormales (\mathbf{U}), de forma tal que cada forma de onda estaría relacionada con un coeficiente que llamaremos s (Una señal de energía). Matemáticamente tendríamos esto:

$$S_i(t) = \sum_{j=1}^n s_{ij} \cdot U_j(t) \quad (3.2)$$

Es decir, a cada símbolo m_i se le asocia una forma de onda s . Si desarrollamos la fórmula anterior, **para todos los símbolos posibles**, tendríamos un sistema de ecuaciones como sigue:

$$\begin{aligned} s_1(t) &= s_{11} \cdot U_1(t) + s_{12} \cdot U_2(t) + s_{13} \cdot U_3(t) + \dots + s_{1n} \cdot U_n(t) \\ s_2(t) &= s_{21} \cdot U_1(t) + s_{22} \cdot U_2(t) + s_{23} \cdot U_3(t) + \dots + s_{2n} \cdot U_n(t) \\ s_3(t) &= s_{31} \cdot U_1(t) + s_{32} \cdot U_2(t) + s_{33} \cdot U_3(t) + \dots + s_{3n} \cdot U_n(t) \\ &\vdots \\ s_m(t) &= s_{m1} \cdot U_1(t) + s_{m2} \cdot U_2(t) + s_{m3} \cdot U_3(t) + \dots + s_{mn} \cdot U_n(t) \end{aligned} \quad (3.3)$$

El objetivo en el segundo sistema mostrado en la **Figura 1** es el de obtener el estimado que más se aproxime al valor real. Esto se hace minimizando la energía de la señal de error entre el símbolo original y el estimado:

$$s_j = \int_0^T s(t) \cdot U_j(t) dt \quad (3.4)$$

$$j = 1, 2, 3, \dots, N$$

Si lo vemos desde la perspectiva vectorial, el procedimiento será entonces el de obtener una representación de la señal en función de dos vectores en el plano. El estimado del vector original sería entonces la proyección de éste sobre el plano:

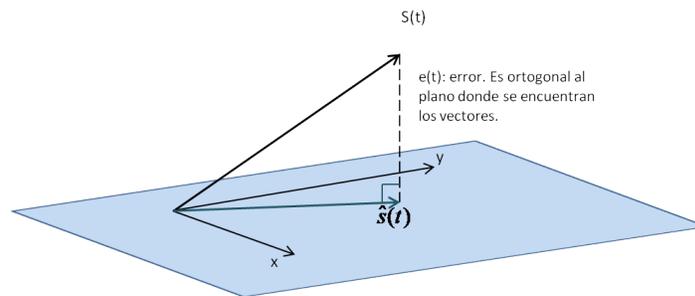


Figure 3.3: Ejemplo aplicado a vectores. $\hat{s}(t)$ es el estimado de cada forma de onda original $s(t)$ y $e(t)$ sería la introducción de ruido de AWGN en el sistema.

Habiendo explicado la síntesis teórica de la ortogonalización, ¿Cómo podemos hallar las bases necesarias para representar las señales de nuestro sistema? Para ello deben seguirse estos pasos:

Supongamos que se da un conjunto de señales de energía $s_i(t)$ que se quieren representar por medio de bases U_j en un intervalo de tiempo $[0, T]$:

$$S_i(t) = \sum_{j=1}^n s_{ij} \cdot U_j(t) \quad (3.5)$$

Las bases deben cumplir con el principio de ortonormalidad mencionado al principio:

$$\int_0^T U_j(t) \cdot U_k(t) dt = \begin{cases} 1 & \rightarrow j = k \\ 0 & \rightarrow j \neq k \end{cases} \quad (3.6)$$

Entonces:

3.2 Paso 1: Se fija $s_{ij} = 0$ exceptuando el primer valor: s_{i1} :

$$s_i(t) = s_{i1} U_1(t) + s_{i2} U_2(t) + s_{i3} U_3(t) + \dots + s_{in} U_n(t)$$

Figure 3.4

Elevamos toda la ecuación al cuadrado y la integramos en el intervalo $[0, T]$:

$$\int_0^T [s_1(t)]^2 dt = \int_0^T s_{11}^2 \cdot U_1^2(t) dt = \int_0^T s_{11}^2 \cdot U_1(t) \cdot U_1(t) dt \quad (3.7)$$

Por el principio de ortonormalidad:

$$\int_0^T [s_1(t)]^2 dt = s_{11}^2 \int_0^T U_1(t) U_1(t) dt$$

Figure 3.5

Quedando $s_1(t)$ sólo en función de s_{11} , por lo que ya se puede despejar:

$$\sqrt{\int_0^t [s_1(t)]^2 dt} = s_{11} \quad (3.8)$$

Finalmente:

$$U_1(t) = \frac{s_1(t)}{s_{11}} \quad (3.9)$$

Con esto obtenemos la primera base para representar nuestra señal. Para calcular $U_2(t)$, debemos restarle a $s_2(t)$ su proyección sobre $U_1(t)$; esto cumpliría con la condición de que la base sea ortogonal.

3.3 Paso 2: Se fija $S_{ij}=0$ exceptuando los valores de s_{21} y s_{22} :

$$s_2(t) = s_{21} \cdot U_1(t) + s_{22} \cdot U_2(t) + s_{23} \cdot U_3(t) + \dots + s_{2n} \cdot U_n(t)$$

Figure 3.6

Ecuación (a)

Multiplicamos la ecuación por $U_1(t)$ y la integramos en el intervalo $[0, T]$:

$$\int_0^T s_2(t) \cdot U_1(t) dt = \int_0^T s_{21} \cdot U_1(t) \cdot U_1(t) dt + \int_0^T s_{22} \cdot U_2(t) \cdot U_1(t) dt \quad (3.10)$$

Quedando entonces:

$$\int_0^T s_2(t) \cdot U_1(t) dt = s_{21} \quad (3.11)$$

La ecuación (a) podemos reordenarla así:

$$s_2(t) - s_{21} \cdot U_1(t) = s_{22} U_2(t) \quad (3.12)$$

Al igual que para el paso 1, elevamos toda la ecuación al cuadrado y la integramos en el intervalo $[0, T]$, quedando como sigue:

$$\int_0^T (s_2(t) - s_{21} U_1(t))^2 dt = \int_0^T s_{22}^2 U_2(t) \cdot U_2(t) dt \quad (3.13)$$

Usando nuevamente el principio de ortonormalidad, nos queda S_{22} en función de la señal S_2 , el coeficiente S_{21} y la base U_1 :

$$s_{22} = \sqrt{\int_0^T (s_2(t) - s_{21} U_1(t))^2 dt} \quad (3.14)$$

Finalmente, con la ecuación (a):

$$\begin{aligned} s_2(t) &= s_{21} U_1(t) + s_{22} U_2(t) \\ \Rightarrow U_2(t) &= \frac{[s_2(t) - s_{21} U_1(t)]}{s_{22}} \end{aligned} \quad (3.15)$$

Se buscarán cuantas bases sean necesarias hasta el punto en el que $U_n=0$. Se pudiera resumir este proceso de la siguiente forma:

$$\begin{aligned} u_1(t) &= \frac{s_1(t)}{\|s_1(t)\|} \\ u_2(t) &= \frac{s_2(t) - \langle s_2(t), u_1(t) \rangle u_1(t)}{\|s_2(t) - \langle s_2(t), u_1(t) \rangle u_1(t)\|} \\ u_n(t) &= \frac{s_n(t) - \sum_{m=1}^{n-1} \langle s_n(t), u_m(t) \rangle u_m(t)}{\|s_n(t) - \sum_{m=1}^{n-1} \langle s_n(t), u_m(t) \rangle u_m(t)\|} \end{aligned}$$

Figure 3.7

Donde:

$$\| X_1 \| = \sqrt{E_1} = \sqrt{\int_{-\infty}^{+\infty} X_1^2(t) dt}$$

y

$$\langle x(t), y(t) \rangle = \int x(t) y(t) dt$$
(3.16)

Es importante resaltar que si el proceso de ortogonalización se inicia con una señal diferente a la señal $s_1(t)$, se obtendría un conjunto distinto de bases ortonormales pero igualmente representativa.

3.4 Constelación

Es la representación gráfica de cada señal $s_i(t)$ en función de las bases U_i . Más adelante observaremos que los diagramas de constelación también sirven para representar los esquemas de modulación digital en el plano complejo. Cada punto perteneciente a la constelación corresponde a un símbolo de modulación.

Aquí consideraremos como ‘ejes’ las bases calculadas a partir de la Ortogonalización, es decir, U_j . El procedimiento es sencillo: sólo se debe representar con un punto a la(s) forma(s) de onda s_i sobre el eje de la base. Por ejemplo: Supongamos que se tienen dos señales, que identifican una determinada codificación o modulación, y que pueden representarse con una sola base de acuerdo a las siguientes ecuaciones:

$$\begin{aligned} S_1 &= V\sqrt{Tb} \cdot U_1 \\ S_2 &= -V\sqrt{Tb} \cdot U_1 \end{aligned}$$
(3.17)

Como sólo se necesita una base para representar estas formas de onda, entonces se tendrá un ‘eje’ que es U_1 :

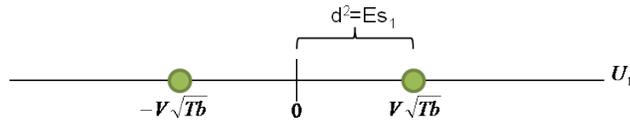


Figure 3.8: Ejemplo de constelación.

A partir de la constelación se puede obtener un parámetro fundamental que es la **Energía**. Si elevamos al cuadrado la distancia que existe entre el origen y un punto de la constelación obtendríamos la energía de la primera forma de onda S_1 :

$$E_{S_1} = V^2 T b$$

Para calcular la Energía de S_2 se hace exactamente el mismo procedimiento.

En la simulación de este módulo se podrá calcular el número de bases necesarias de acuerdo a los coeficientes s_i dados. A partir de ellas también se podrá observar la constelación correspondiente. Para descargar el código fuente, se debe hacer click en el siguiente enlace: [MEDIA OBJECT]²

[MEDIA OBJECT]³

²This media object is a downloadable file. Please view or download it at <gramschmidtULTIMATE1.vi>

³This media object is a video file. Please view or download it at <gs_constelacion.mpg>

Chapter 4

4. Códigos de Línea¹

4.1 CÓDIGOS DE LÍNEA

González C. Y. Venuska

Mezoa R. Mariangela

Resumen

En este módulo se desarrollarán teóricamente los códigos de línea usados comúnmente en un sistema de Comunicaciones Digitales: NRZ, RZ y Manchester. Se explicarán sus parámetros más importantes como lo son la señal en frecuencia y en tiempo, el ancho de banda, la potencia de la señal y las constelaciones correspondientes a cada código. Finalmente, se hará una demostración en LabVIEW que permita reforzar la información teórica del tema.

De acuerdo a la estructura de un sistema de Comunicaciones Digitales, al tener una señal discretizada en el tiempo y en la amplitud, estaría representada por símbolos cuya tasa de transmisión es medida en baudios. Ahora bien, cuando se quiere transmitir la señal en bandabase binario o m-ario a través del canal debe asignarse una forma de onda a cada uno de los símbolos. Esto se hace gracias al método conocido como codificación de línea.

4.1.1 ¿Qué es la codificación de Línea?

Esta surge de la necesidad de representar una señal en formato digital a través de diversos medios de transmisión. Para esto se le asignan formas de onda arbitrarias a cada bit o símbolo que representa la señal, generando cambios inmediatos en los parámetros más importantes de la señal como lo son la Potencia de Transmisión, el Ancho de Banda requerido por el canal, nivel DC, entre otros.

Antes de definir los códigos de línea más usados en los sistemas de Comunicaciones, repasaremos algunos conceptos fundamentales que facilitarán el entendimiento del tema:

Definition 4.1: Ancho de Banda (BW):

En el ámbito de las comunicaciones, el ancho de banda representa el rango (en Hertz, Hz) de frecuencias en donde se concentra la mayor parte de la potencia de la señal que se está usando. A partir de este punto se podrá calcular el ancho de banda en las simulaciones usando la transformada de Fourier, cuando se expresa la señal original en el dominio de la frecuencia. Matemáticamente se pudiera definir así:

$$BW = \Delta f = f_{cs} - f_{ci} \quad (4.1)$$

Siendo f_{cs} y f_{ci} las frecuencias de corte Superior e Inferior, respectivamente.

¹This content is available online at <<http://cnx.org/content/m35716/1.4/>>.

Definition 4.2: Tiempo de bit (T_b):

Está definido como el tiempo empleado para la representación y transmisión de un bit en el sistema.

Definition 4.3: Intervalo de impulso (τ):

Es la duración mínima del impulso usado para representar la información.

Definition 4.4: Velocidad de Transmisión (V_t):

Se define como el número máximo de cambios por segundo que experimenta la señal o el número de impulsos que pueden transmitirse en un segundo. Matemáticamente viene expresado como

$$V_t = \frac{1}{\tau} \text{ baudios} \quad (4.2)$$

Definition 4.5: Tasa de Bits (R):

Número de bits transmitidos por segundo. Su unidad son los bps (bits per second) y se representa matemáticamente como:

$$R = \frac{1}{T_b} \text{ bps} \quad (4.3)$$

A continuación se definen diversos códigos de línea binarios y se deduce su Densidad Espectral de Potencia asumiendo equiprobabilidad.

4.2 1. NRZ (Non Return to Zero)

Están caracterizados por mantener constante el valor de la señal de línea durante todo el intervalo T_b . Para efectos de asimilar la información proporcionada por la simulación al final del módulo, se considerarán dos tipos de codificación de este tipo:

4.2.1 1.a. NRZp (No Retorno a Cero-Polar)

Al símbolo "1" se le asigna un valor alto de señal (V) y al símbolo "0" se le asigna el valor opuesto, es decir, $-V$.

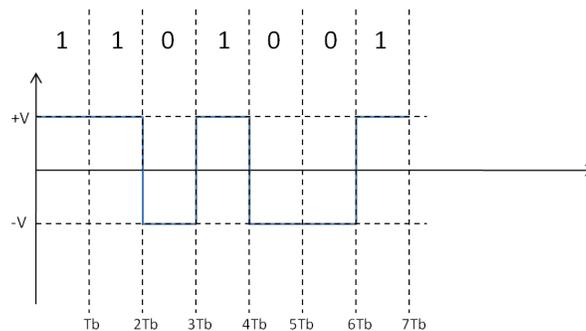


Figure 4.1: Representación de bits usando el código NRZp

Para hallar su **DEP**, se debe representar la señal como el resultado de la **convolución** de un tren de impulsos aleatorio y un pulso de duración T_b y Amplitud $+V$.

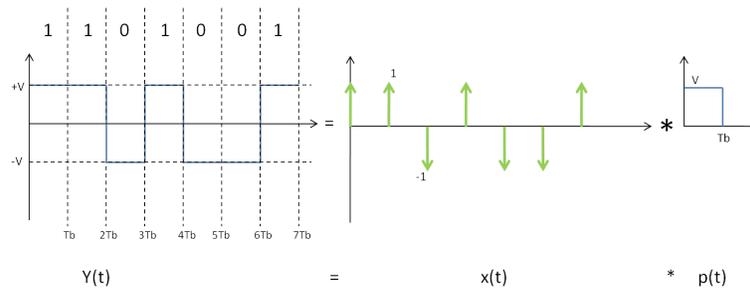


Figure 4.2

Por lo que:

$$Y(t) = x(t) * p(t) \quad (4.4)$$

$$G_y = G_x(f) |P(f)|^2$$

Dado que $p(t)$ es una función **determinística**, el reto estará en calcular $G_x(f)$. Para esto se calculará primero la autocorrelación $R_x(\tau)$:

$$G_x(f) = F\{R_x(\tau)\}$$

$$R_x(\tau) = \sum \frac{R_n}{T_b} \delta(\tau + nT_b) \quad (4.5)$$

$$R_n = \lim_{T \rightarrow \infty} \frac{tb}{T} \sum_k A_k \cdot A_{(n+k)}$$

NOTE: Estas fórmulas aplicarán para todos los códigos de línea, siendo el factor variante los valores de A_k , $A_{(n+k)}$ y T_b

Los posibles valores de A_k y $A_{(n+k)}$ son $+1$ y -1 , por lo que:

$$R_x(\tau) = \frac{1}{T_b} \delta(\tau) \quad (4.6)$$

Y al sacar la transformada de Fourier correspondiente, nos queda que:

$$G_x(f) = \frac{1}{T_b}$$

$$|P(f)|^2 = V^2 \cdot T_b^2 \cdot \text{Sinc}^2(ftb) \quad (4.7)$$

$$G_y(f) = V^2 \cdot T_b \cdot \text{Sinc}^2(ftb)$$

Su **DEP** será entonces:

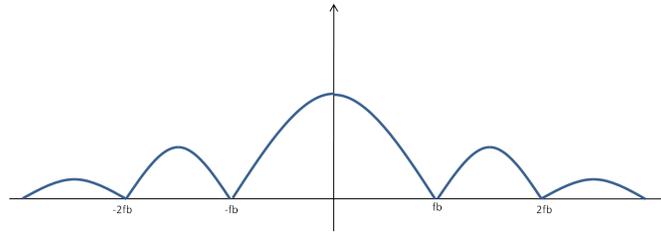


Figure 4.3: El BW correspondiente es fb .

Finalmente, se tiene que se necesita una sola base $U_1(t)$ que es un pulso de duración Tb y altura raíz de $1/Tb$

$$\begin{aligned} S_1(t) &= V\sqrt{Tb} \cdot U_1(t) \\ S_2(t) &= -V\sqrt{Tb} \cdot U_1(t) \end{aligned} \quad (4.8)$$

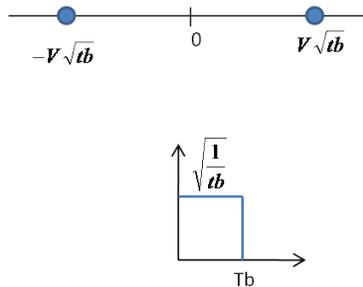


Figure 4.4: Constelación para NRZp

4.2.2 1.b. NRZu (No Retorno a Cero-Unipolar)

Al símbolo “1” se le asigna un valor alto de señal (V) y al símbolo “0” se le asigna el valor cero, es decir, $0V$.

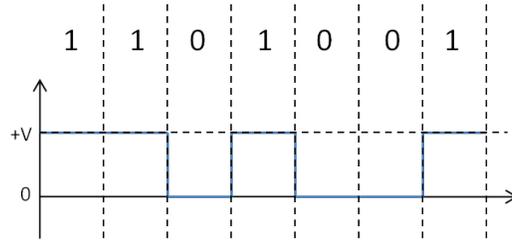


Figure 4.5: Representación de bits usando el código NRZu

Para obtener la DEP se descompone la señal:

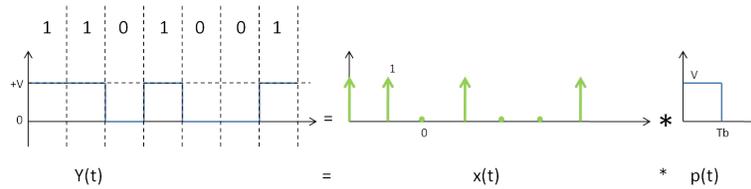


Figure 4.6

Ahora bien, siguiendo el mismo proceso que para el punto 1.a, los posibles valores de A_k y $A_{(n+k)}$ son $+1$ y 0 , por lo que:

$$R_x(\tau) = \frac{1}{4T_b} \delta(\tau) + \sum \frac{1}{4T_b} \delta(\tau + nT_b) \tag{4.9}$$

Y al sacar la transformada de Fourier correspondiente, nos queda que:

$$\begin{aligned} G_x(f) &= \frac{1}{4T_b} + \sum \frac{1}{4T_b^2} \delta(f + nfb) \\ |P(f)|^2 &= V^2 \cdot T_b^2 \cdot \text{Sinc}^2(ftb) \end{aligned} \tag{4.10}$$

$$G_y(f) = \frac{V^2}{4} \cdot T_b \cdot \text{Sinc}^2(ftb) + \frac{V^2}{4} \delta(f)$$

Su **DEP** será:

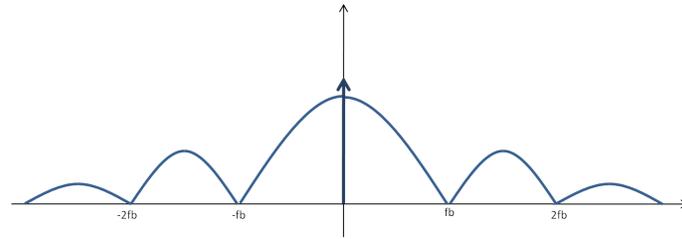


Figure 4.7: El BW correspondiente también es fb .

Y su base y constelación:

$$S_1(t) = V\sqrt{tb}U_1(t) \quad (4.11)$$

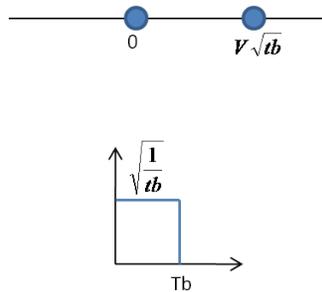


Figure 4.8: Constelación para NRZu

4.3 2. RZ (Return to Zero)

Los códigos de Retorno a Cero se caracterizan por mantener el valor de la señal constante durante el primer semi-intervalo de T_b , para luego pasar a otro nivel en el segundo semi-intervalo. Dependiendo del tipo de código, esta segunda parte puede tener un nivel de $-V$ o un nivel $0V$. Los tipos de código RZ pueden ser:

4.3.1 2.a. RZp (Retorno a Cero-Polar)

Para el símbolo “1” tendrá dos valores: en el primer semi-intervalo $[0, T_b/2]$ tendrá un nivel $+V$ y para el segundo semi-intervalo $[T_b/2, T_b]$ retornará a $0V$. Ahora, para el símbolo “2” será: en el primer semi-intervalo tendrá un nivel opuesto ($-V$) y para el segundo semi-intervalo también retornará a $0V$. Gráficamente:

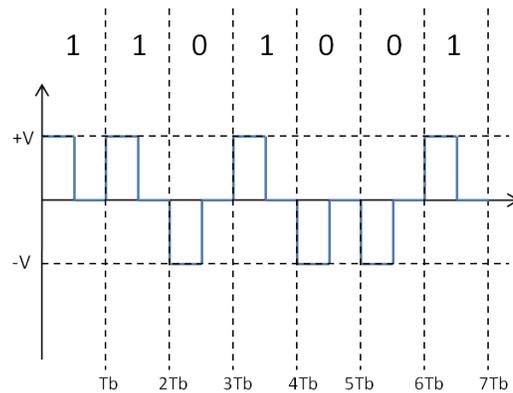


Figure 4.9: Representación de bits usando el código RZp

Si descomponemos la señal:

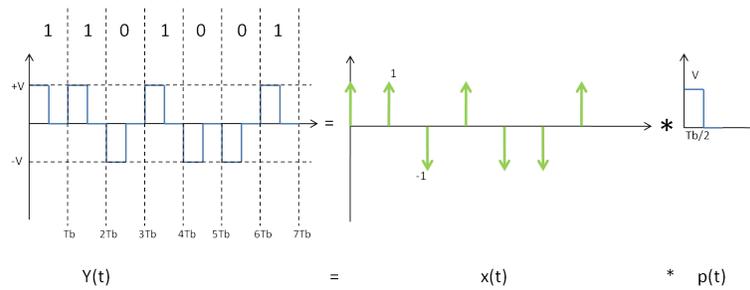


Figure 4.10

Ahora bien, los posibles valores de A_k y $A_{(n+k)}$ son $+1$ y -1 , por lo que:

$$R_x(\tau) = \frac{1}{T_b} \delta(\tau) \quad (4.12)$$

Y al sacar la transformada de Fourier correspondiente, nos queda que:

$$G_x(f) = \frac{1}{T_b}$$

$$|P(f)|^2 = \frac{V^2 \cdot T_b^2}{4} \cdot \text{Sinc}^2\left(f \frac{T_b}{2}\right)$$

$$G_y(f) = \frac{V^2}{4} \cdot T_b \cdot \text{Sinc}^2\left(f \frac{T_b}{2}\right)$$

Su **DEP** es como sigue:

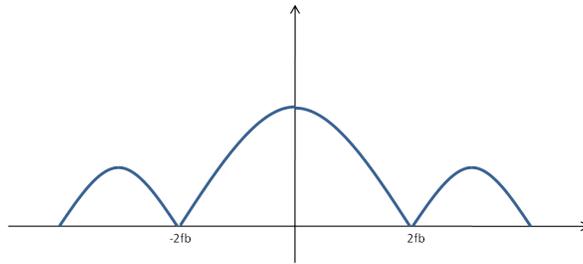


Figure 4.11: El BW correspondiente es $2fb$.

4.3.2 2.b. RZu (Retorno a Cero-Unipolar)

Para el símbolo “1” tendrá dos valores: en el primer semi-intervalo $[0, T_b/2]$ tendrá un nivel $+V$ y para el segundo semi-intervalo $[T_b/2, T_b]$ retornará a $0V$. Ahora, para el símbolo “2” se mantendrá en $0V$ por todo el intervalo T_b . Gráficamente:

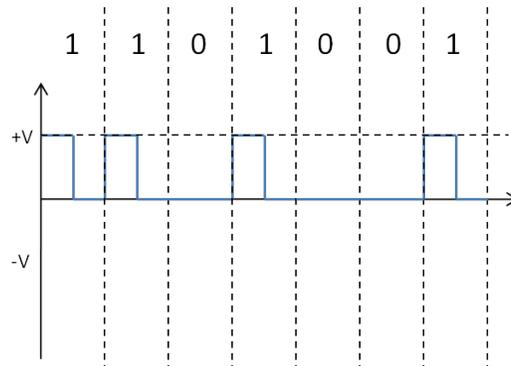


Figure 4.12: Representación de bits usando el código RZu

La descomposición de la señal para obtener la DEP es como sigue:

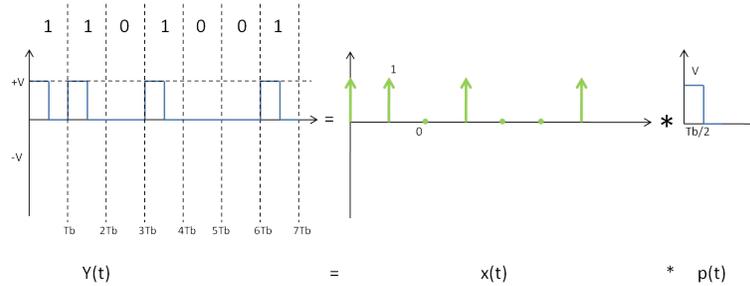


Figure 4.13

La autocorrelación es entonces:

$$R_x(\tau) = \frac{1}{4T_b} \delta(\tau) + \sum_{\forall n} \frac{1}{4T_b} \delta(\tau + nT_b) \quad (4.13)$$

Por lo que la transformada de Fourier de la misma y la Función Gy(f) nos queda así:

$$\begin{aligned} G_x(f) &= \frac{1}{4T_b} + \sum_{\forall n} \frac{1}{4T_b} \delta(f + nfb) \\ |P(f)|^2 &= \frac{V^2 \cdot T_b^2}{4} \cdot \text{Sinc}^2\left(f \frac{T_b}{2}\right) \\ G_y(f) &= \frac{V^2 T_b}{16} \text{Sinc}^2\left(f \frac{T_b}{2}\right) + \sum_{\forall n} \frac{V^2}{16} \text{Sinc}^2\left(f \frac{T_b}{2}\right) \delta(f + nfb) \end{aligned} \quad (4.14)$$

En el dominio de la frecuencia, Gy puede expresarse como:

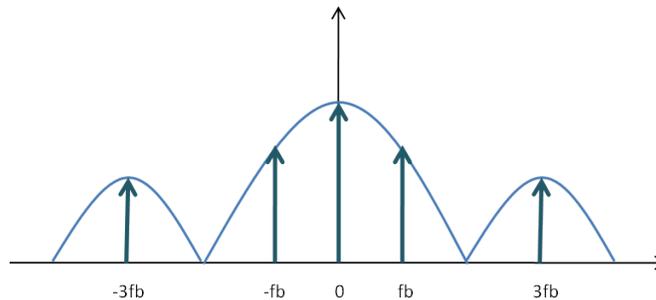


Figure 4.14: El BW correspondiente es 2fb.

4.3.3 3. Manchester

Al igual que con los códigos RZ, el código Manchester se caracteriza por tener una transición de valor en $T_b/2$ durante el intervalo $[0, T_b]$. El “1” se representa por cambio de $+V$ a $-V$ y el “0” hace el proceso opuesto.

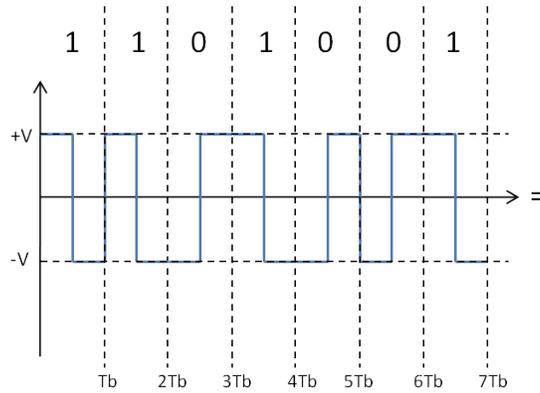


Figure 4.15: Representación de bits usando el código Manchester

Descomponemos la señal:

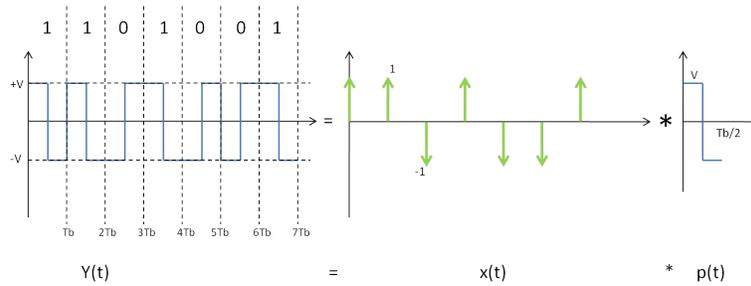


Figure 4.16

Y, haciendo el mismo procedimiento que para los puntos anteriores, llegamos a:

$$G_y(f) = V^2 T_b \cdot \text{Sinc}^2\left(f \frac{T_b}{2}\right) \cdot \text{Sen}^2\left(f \frac{T_b}{2}\right) \quad (4.15)$$

Con su expresión gráfica en el dominio de la frecuencia:

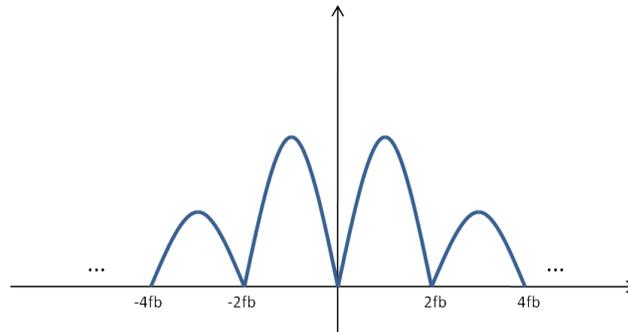


Figure 4.17: El BW correspondiente es $2fb$

En la simulación de este módulo se podrán observar los diferentes códigos de línea explicados previamente, pudiendo observar con más exactitud las gráficas en los dominios de tiempo y frecuencia, el cálculo de la potencia de la señal, observar la constelación de cada código (en base a la teoría explicada en el módulo 3 (Chapter 3)) y el diagrama de Ojo correspondiente.

4.3.4 VI de simulación

El VI principal y sus Sub-VIs correspondientes pueden descargarse en el siguiente enlace. [MEDIA OBJECT]²

4.3.5 Video de demostración

[MEDIA OBJECT]³

²This media object is a downloadable file. Please view or download it at <codigolinea.zip>

³This media object is a video file. Please view or download it at <codlinea.mpg>

Chapter 5

5. Modulaciones Binarias: Teoría y simulación en LabVIEW¹

5.1 MODULACIÓN BINARIA: TEORÍA Y SIMULACIÓN EN LABVIEW

González C. Y. Venuska

Mezoa R. Mariangela

Resumen

En este módulo se podrán analizar con profundidad los tipos de modulación binaria más importantes: OOK, PSK, FSK, MSK y GMSK para los sistemas de comunicación digital. Se desarrollarán sus parámetros más relevantes como lo son la potencia, el Ancho de Banda, su constelación y su representación en los dominios de la frecuencia y el tiempo.

Como ya mencionamos al principio de este curso, el término modular implica modificar ciertos parámetros de una onda portadora en función de otra señal llamada onda moduladora, que es la que contiene información, para que sea transmitida por el canal de nuestro sistema de Comunicación Digital.

De acuerdo con la naturaleza de la señal portadora, los métodos de modulación –a modo general- se pueden clasificar de dos tipos:

- Modulación por onda continua.
- Modulación por pulsos.

En este módulo trabajaremos con la Modulación por Onda Continua, dado que se da una onda portadora sinusoidal a la que se le puede modificar su amplitud, su frecuencia o su fase en función de la señal moduladora; como estamos trabajando con un sistema digital, el mensaje que se desea transmitir viene en formato binario.

Entre los métodos de modulación binaria más importantes tenemos:

- Modulación Digital de Amplitud/Modulación por cambio de amplitudes. (ASK)
- Modulación por Desplazamiento de Frecuencia (FSK)
- Modulación por Cambio de Fase (PSK)
- Modulación por Desplazamiento Mínimo de Frecuencia (MSK)
- Modulación por Desplazamiento Mínimo de Frecuencia Gaussiano (GMSK)

¹This content is available online at <<http://cnx.org/content/m35717/1.1/>>.

5.1.1 1. Amplitude Shift Keying (ASK)

La técnica de modulación digital más simple es la modulación digital de amplitud, ya que el parámetro que se varía de la portadora es la amplitud. Como el mensaje es binario, entonces la amplitud de la senoide varía entre dos valores posibles. Un caso especial de esta modulación es cuando uno de los dos valores posibles es cero, por lo que la modulación toma el nombre de OOK (On-Off Keying): Aquí la portadora se considera encendida (Valor de Amplitud V para el bit 1) o apagada (supresión de la portadora para el bit 0). Se tiene entonces:

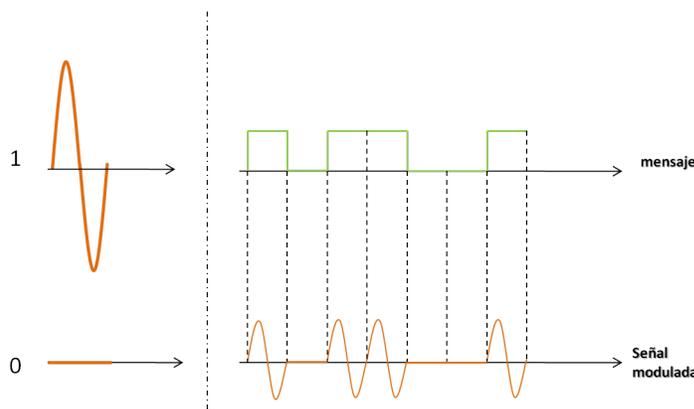


Figure 5.1: Ejemplo de modulación OOK.

Una ecuación que define este tipo de modulación digital es:

$$X_{\text{ask}}(t) = (1 + b(t)) \cdot \left(\frac{V}{2} \text{Cos}(w_c t) \right) \quad (5.1)$$

Donde:

$X_{\text{ask}}(t)$ = Señal modulada.

$b(t)$ = señal binaria moduladora (volts). Puede tomar dos valores: $+1$ cuando el bit enviado es **1**, y -1 cuando el bit enviado es **0**.

V = Amplitud de la señal portadora (volts)

w_c = Frecuencia de la señal portadora (radianes por segundo)

Entonces tenemos que si $b(t) = 1$:

$$X_{\text{ask}}(t) = V \text{Cos}(w_c t) \quad (5.2)$$

Pero, si $b(t) = -1$:

$$X_{\text{ask}}(t) = 0$$

Para hallar el espectro de la señal modulada OOK es necesario hacerle la transformada de Fourier a la función de autocorrelación.

$$X_{\text{ask}}(t) \rightarrow G_{\text{ask}}(f) \quad (5.3)$$

$$X_{\text{ask}}(t) \rightarrow \frac{V^2}{16} [\delta(f - f_c) + \delta(f + f_c) + t_b \text{Sinc}^2((f + f_c) t_b) + t_b \text{Sinc}^2((f - f_c) t_b)]$$

Al graficar, tenemos que:

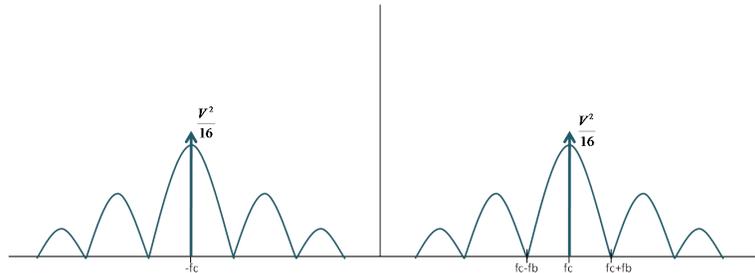


Figure 5.2: OOK en el dominio de la frecuencia

La potencia puede obtenerse integrando la Densidad Espectral de Potencia o a partir de la señal en tiempo. Para equiprobabilidad la potencia resultaría $V^2/4$

El Ancho de banda puede determinarse a partir del espectro:

$$BW = fc + fb - (fc - fb) = fc - fc - 2fb = 2fb \quad (5.4)$$

Finalmente, se puede representar la señal modulada en OOK a partir de la constelación. Se define entonces la base para representar la señal:

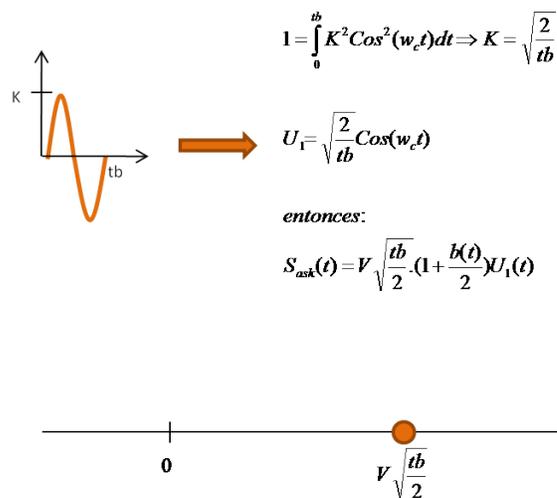


Figure 5.3: Constelación de OOK

5.1.2 2. Frequency-Shift Keying (FSK)

Aquí, el parámetro que se varía de la portadora es la frecuencia.

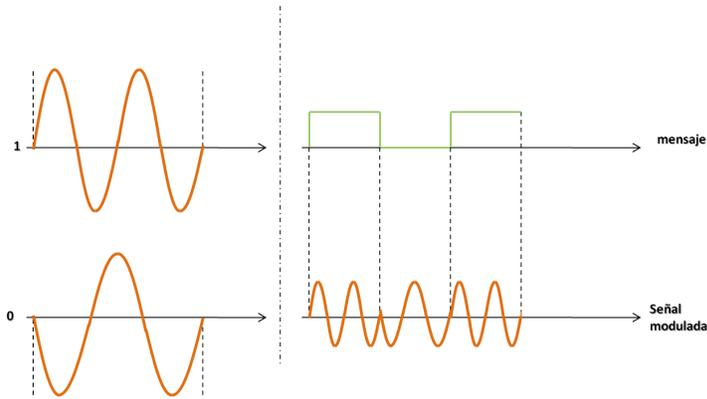


Figure 5.4: Ejemplo de modulación FSK.

Si se transmite el bit 1, la señal modulada será una senoide de frecuencia f_A . Si se transmite el bit 0, la señal FSK tendrá frecuencia f_B .

$$\begin{aligned} f_A &= \frac{(\omega_c + \Omega)}{2\pi} \\ f_B &= \frac{(\omega_c - \Omega)}{2\pi} \end{aligned} \quad (5.5)$$

La frecuencia de la señal portadora es:

$$f_c = \frac{(f_A + f_B)}{2} \quad (5.6)$$

Ahora, la ecuación que define esta modulación es:

$$X_{fsk}(t) = V \cdot \text{Cos}(2\pi (f_c + b(t) \Delta f) t) \quad (5.7)$$

Donde:

$X_{fsk}(t)$ = Señal modulada.

$b(t)$ = señal binaria moduladora (volts). Puede tomar dos valores: $+1$ cuando el bit enviado es **1**, y -1 cuando el bit enviado es **0**.

V = Amplitud de la señal portadora (volts)

f_c = Frecuencia central de la señal portadora (Hz)

Δf = Desviación máxima de frecuencia alrededor de la portadora. $f_A = f_c + \Delta f$; $f_B = f_c - \Delta f$

Entonces tenemos que si $b(t) = 1$:

$$X_{fsk}(t) = V \text{Cos}(2\pi (f_c + \Delta f) t) \quad (5.8)$$

Pero, si $b(t) = -1$:

$$X_{fsk}(t) = V \text{Cos}(2\pi (f_c - \Delta f) t) \quad (5.9)$$

La Densidad Espectral de Potencia de la señal FSK puede obtenerse modelándola como la suma de dos señales OOK; en este caso

$$G_{\text{fsk}}(f) = \frac{V^2}{16} [\delta(f - f_A) + \delta(f + f_A) + \delta(f - f_B) + \delta(f + f_B)] + \frac{V^2}{16} [\text{tb.Sinc}^2((f - f_A) \text{tb}) + \text{tb.Sinc}^2((f + f_A) \text{tb}) + \text{tb.Sinc}^2((f - f_B) \text{tb}) + \text{tb.Sinc}^2((f + f_B) \text{tb})] \quad (5.10)$$

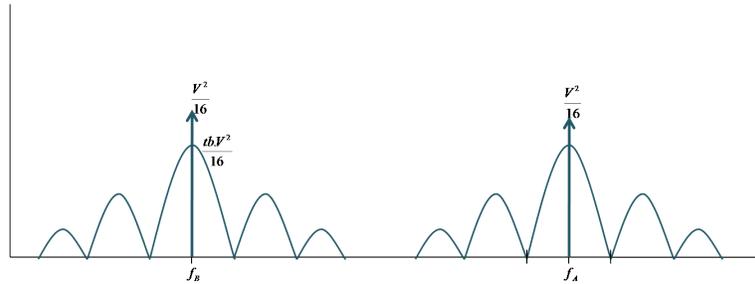


Figure 5.5: CPFSK en el dominio de la frecuencia

El ancho de banda total de la señal es:

$$\text{BW} = (f_A - f_B) + 2f_b \quad (5.11)$$

Y su constelación:

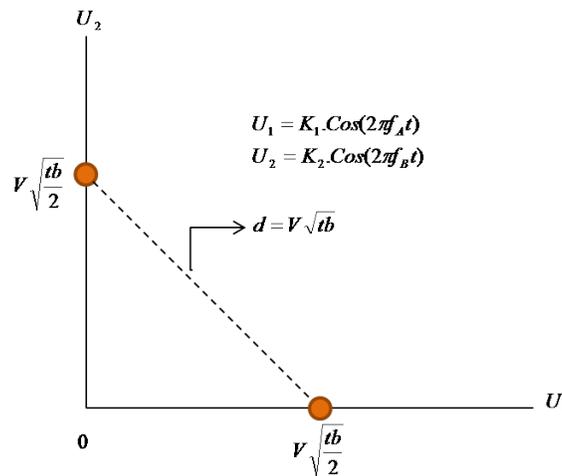


Figure 5.6: Constelación de CPFSK

5.1.3 3. Phase-Shift Keying (PSK)

En esta modulación, el parámetro de la señal sinusoidal portadora que se varía es la fase. Es una forma de modulación digital angular en donde la amplitud de la portadora se mantiene constante. Teóricamente

se aproxima al concepto de modulación en fase convencional (PM), con la diferencia que aquí la señal de entrada es digital binaria, por lo que se tienen dos fases de salida para una sola frecuencia portadora: Una fase de salida representa el bit 1 y otra representa el bit 0. Cuando la señal binaria de entrada cambia de estado, la fase de la portadora de salida varía entre dos ángulos que están desfasados 180° (0 y π). En este caso la modulación se le llama PRK (Phase Reverse Keying)

Se tiene entonces la siguiente situación:

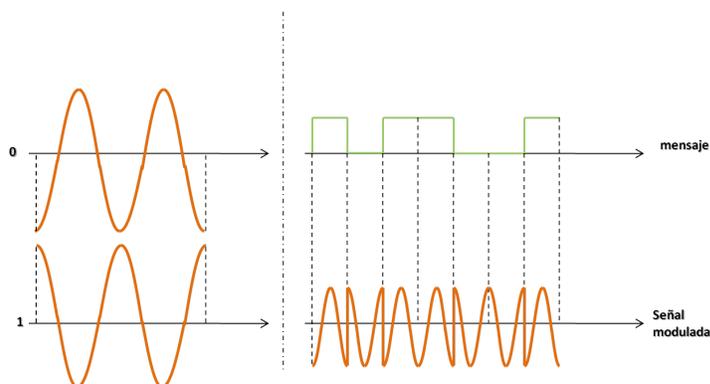


Figure 5.7: Ejemplo de modulación PRK.

La ecuación correspondiente para este tipo de modulación es:

$$X_{\text{prk}}(t) = V \cdot b(t) \cdot \text{Cos}(w_c t) \quad (5.12)$$

Donde:

$X_{\text{prk}}(t)$ = Señal modulada.

$b(t)$ = señal binaria moduladora (volts). Puede tomar dos valores: $+1$ cuando el bit enviado es **1**, y -1 cuando el bit enviado es **0**.

V = Amplitud de la señal portadora (volts)

Wc = Frecuencia de la señal portadora (radianes por segundo)

Si

$$b(t) = 1 \Rightarrow X_{\text{prk}}(t) = V \cdot \text{Cos}(w_c t) \quad (5.13)$$

$$b(t) = -1 \Rightarrow X_{\text{prk}}(t) = -V \cdot \text{Cos}(w_c t)$$

La Densidad Espectral de potencia resulta ser:

$$X_{\text{prk}}(t) \rightarrow G_{\text{prk}}(f) \quad (5.14)$$

$$G_{\text{prk}}(f) = \frac{V^2}{4} [t_b \text{Sinc}^2((f + f_c) t_b) + t_b \text{Sinc}^2((f - f_c) t_b)]$$

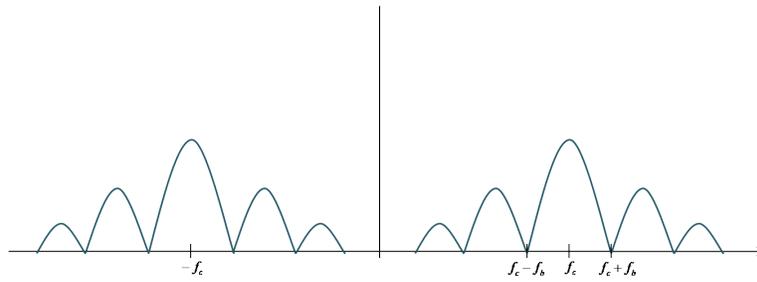


Figure 5.8: PRK en el dominio de la frecuencia

Como se observa, el ancho de banda práctico será $BW=2f_b$. Finalmente su constelación es:

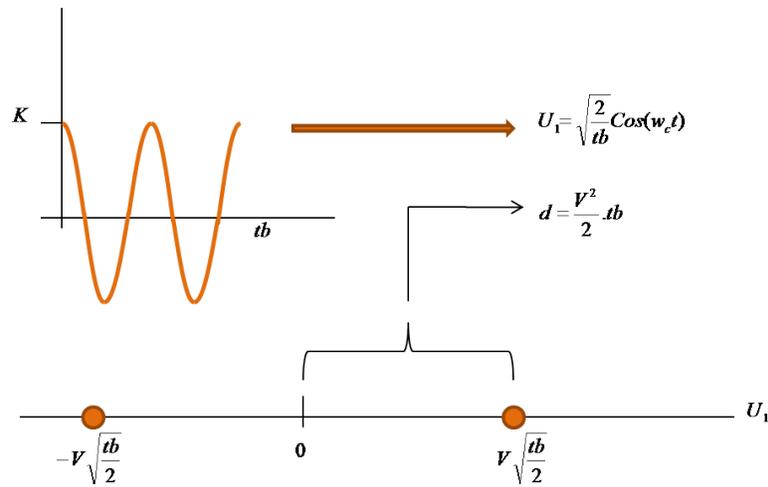


Figure 5.9: Constelación de PRK

5.1.4 4. Minimum-Shift Keying (MSK)

En la modulación digital FSK, cuando se tiene una señal binaria de entrada que cambia de **0** lógico a un **1** lógico (y viceversa), la frecuencia de la señal modulada en la salida se desplaza entre dos frecuencias: una **frecuencia de marca** (o de 1 lógico, f_m) y una **frecuencia de espacio** (o de 0 lógico, f_s). Ellas se encuentran separadas de la frecuencia de la portadora por la desviación máxima de frecuencia Δf , de modo que

$$f_c \pm \Delta f \tag{5.15}$$

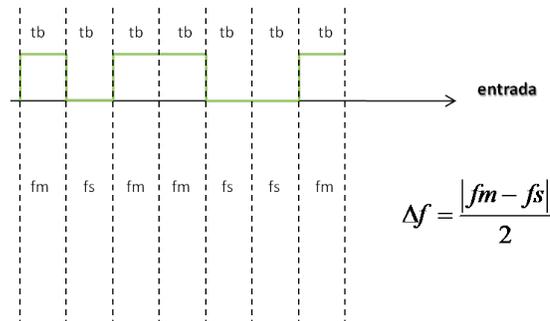


Figure 5.10

La modulación por desplazamiento mínimo se da cuando la diferencia entre la frecuencia de espacio y la frecuencia de marca es igual a $f_b/2$. Entonces, con una señal binaria de entrada, la señal modulada en MSK quedaría así

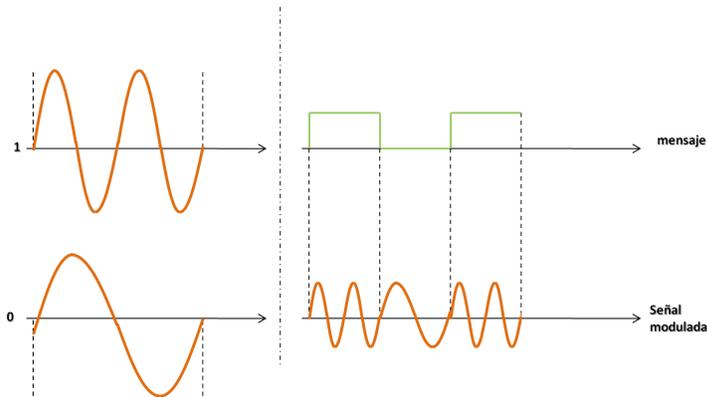


Figure 5.11: Ejemplo de modulación MSK.

Como se observa, la señal modulada es muy parecida a la señal FSK, por lo que se le considera un caso especial de la última (Dado que la transición de bits no presenta discontinuidad). Las fórmulas matemáticas y la ecuación de MSK en el dominio de la frecuencia coinciden con las de FSK pero gráficamente la Densidad Espectral de Frecuencia luce diferente, presentando un menor ancho de banda ya que los dos Sinc al cuadrado se superponen.

5.1.5 5. Gaussian Minimum-Shift Keying (GMSK)

En las comunicaciones digitales, GMSK también representa un esquema de modulación por desplazamiento de frecuencia con fase continua. Es muy parecido a la modulación MSK, con la diferencia de que la señal

digital binaria de entrada pasa primero por un filtro Gaussiano antes de ser modulada. Esto provoca una reducción evidente en los lóbulos laterales del espectro de la señal modulada y en la interferencia entre portadoras de señales en canales de frecuencia adyacentes, con una consecuente reducción del ancho de banda comparada incluso con MSK

5.2 Simulaciones en LabVIEW

Para entender mejor el funcionamiento de estas modulaciones digitales se muestran a continuación cinco VIs correspondientes a cada punto explicado previamente. Para descargarlos se debe seguir los siguientes enlaces: [MEDIA OBJECT]²

5.3 Demostración:

[MEDIA OBJECT]³

²This media object is a downloadable file. Please view or download it at <binarias.zip>

³This media object is a video file. Please view or download it at <binarias.mpg>

Chapter 6

6. Modulaciones M-arias¹

6.1 MODULACIONES M-ARIAS

González C. Y. Venuska

Mezoa R. Mariangela

Resumen

El contenido de este módulo engloba las características más relevantes de las modulaciones m-arias, además de la simulación de cada modulación en LabVIEW.

Hasta el módulo anterior (Modulaciones Binarias²) habíamos considerado las técnicas de modulación cuando sólo se empleaba un bit a la vez del mensaje de entrada. Aquí tomaremos en consideración los diferentes tipos de modulación cuando se toman símbolos para producir cambios sobre la señal portadora.

El término **M-ario** deriva de la palabra **binario**. La letra **M** representa la cantidad de condiciones posibles (símbolos) para una determinada cantidad de variables binarias consideradas. En los casos anteriores de modulación (OOK, PRK, FSK, etc) habíamos trabajado con bits individuales (1 y 0), por lo que se podría decir que estos métodos son M-arios en donde $M = 2$.

Si los símbolos provienen de combinaciones de bits, se puede usar la siguiente ecuación para relacionar el número de bits con el número de símbolos:

$$N = \log_2 M$$

Figure 6.1

En donde:

N = Cantidad de bits codificados

M = Cantidad de condiciones posibles de salida con N bits.

Entonces, si por ejemplo tomamos la modulación digital binaria FSK tendríamos dos posibles variantes: “1” lógico o un “0” lógico, así que $M = 2$:

¹This content is available online at <<http://cnx.org/content/m35715/1.1/>>.

²"Modulaciones Binarias" <<http://cnx.org/content/m35235/latest/>>

$$\begin{aligned}
 N &= \log_2 2 \\
 2^N &= 2 \\
 N &= 1
 \end{aligned}$$

Figure 6.2

Si se toman dos bits codificados juntos entonces $M=2^2=4$. En general:

N	M
1	2
2	4
3	8
4	16
5	32

Table 6.1

El ancho de banda de cada señal modulada M-aria estará relacionado con la tasa de símbolos fs y con la tasa de bits fb de la siguiente forma:

$$BW \leftrightarrow fs = \frac{fb}{\log_2 M} = \frac{fb}{N}$$

Figure 6.3

Donde fs es la tasa de símbolos (bauds) y fb es tasa de bits de entrada(bps).

6.1.1 1. QPSK (Quaternary Phase Shift Keying)

La modulación por desplazamiento cuaternario de fase es un tipo de modulación M-aria en donde el factor M es igual a 4. Aquí se tendrán entonces cuatro posibles fases de salida para una sola frecuencia de portadora y cuatro condiciones distintas de entrada. Como con un sólo bit no se pueden obtener cuatro condiciones distintas (sólo dos, 1 y 0), tomamos combinaciones de dos bits, de modo que:

- 00
- 01
- 10
- 11

Cada par de bits (**dbits**) genera una fase posible de salida. Tenemos entonces la siguiente ecuación que define este tipo de modulación:

$$X_{\text{qpsk}}(t) = \sqrt{Ps}b_p(t)\text{Cos}(w_c t) + \sqrt{Ps}b_i(t)\text{Sen}(w_c t)$$

Figure 6.4

En donde:

$$\begin{aligned}
 X_{\text{qpsk}}(t) &= \text{Señal modulada.} \\
 \sqrt{Ps} &= \sqrt{\frac{V^2}{2}} \\
 b_p(t) &= \text{Bit par} \\
 w_c &= \text{Frecuencia de la portadora en Radianes.} \\
 b_i(t) &= \text{Bit impar}
 \end{aligned}$$

Figure 6.5

Los parámetros **bp** y **bi** siguen el mismo lineamiento que para las modulaciones binarias: para un **1** lógico tienen valor de **+1** y para un **0** lógico tienen valor de **-1**.

Si $b_p = +1$ y $b_i = +1$:

$$X_{\text{QPSK}}(t) = \sqrt{P_s(t)} \cdot \text{Cos}(w_c t) + \sqrt{P_s(t)} \cdot \text{Sen}(w_c t)$$

Si $b_p = +1$ y $b_i = -1$:

$$X_{\text{QPSK}}(t) = \sqrt{P_s(t)} \cdot \text{Cos}(w_c t) - \sqrt{P_s(t)} \cdot \text{Sen}(w_c t)$$

Si $b_p = -1$ y $b_i = +1$:

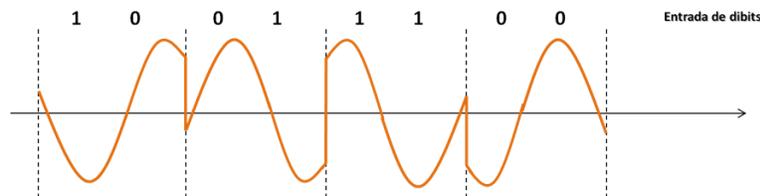
$$X_{\text{QPSK}}(t) = -\sqrt{P_s(t)} \cdot \text{Cos}(w_c t) + \sqrt{P_s(t)} \cdot \text{Sen}(w_c t)$$

Si $b_p = -1$ y $b_i = -1$:

$$X_{\text{QPSK}}(t) = -\sqrt{P_s(t)} \cdot \text{Cos}(w_c t) - \sqrt{P_s(t)} \cdot \text{Sen}(w_c t)$$

Figure 6.6

Para una de las cuatro fases posibles de salida, la portadora tiene exactamente la misma amplitud, lo que evidencia el tipo de modulación con amplitud constante.



Entrada Binaria	Fase de Salida QPSK
00	$3\pi/4$
01	$-3\pi/4$
10	$\pi/4$
11	$-\pi/4$

Figure 6.7: Fase de salida para QPSK.

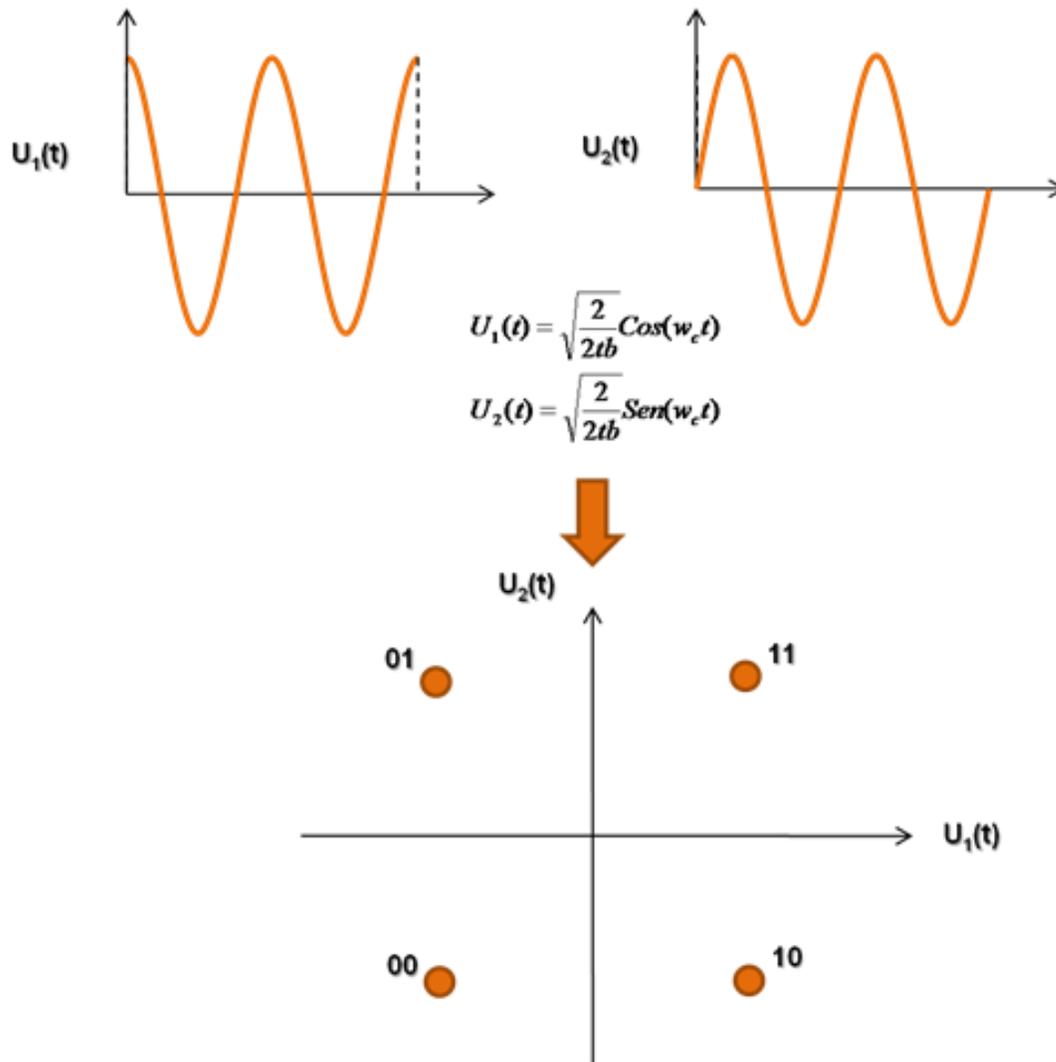


Figure 6.8: Bases y constelación para QPSK La separación angular entre dos puntos adyacentes es de 90° .

La ecuación de la señal modulada en función de las bases ortonormales quedará así:

$$X_{qpsk}(t) = \sqrt{E_1} b_p(t) U_1(t) + \sqrt{E_1} b_i(t) U_2(t)$$

Figure 6.9

Podemos hallar la ecuación en el dominio de la frecuencia usando la Transformada de Fourier de la autocorrelación, resultando:

$$G_{qpsk}(f) = E_1 (\text{Sinc}^2(2tb(f - fc)) + \text{Sinc}^2(2tb(f + fc)))$$

$$BW = \frac{1}{tb}$$

Figure 6.10

6.1.2 2. MPSK (Multiple PSK)

En el caso anterior se hacía la combinación de dos bits para generar cuatro salidas de fase. El término MPSK aplica para cuando se desee combinar bits de N en N. Pudiéramos llegar entonces a una ecuación general que aplique para cuantas combinaciones de bits se necesite:

$$X_{mpsks}(t) = V \cos(\omega_c t + \phi_k)$$

donde:

$$\phi_k = (2a_k(t) + 1) \frac{\pi}{M}$$

$$a_k = 0, 1, \dots, M - 1$$

Figure 6.11

Esto produce $M=2^N$ Símbolos. Las bases quedarían como sigue:

$$U_1(t) = \sqrt{\frac{2}{Ntb}} \cos(\omega_c t)$$

$$U_2(t) = \sqrt{\frac{2}{Ntb}} \sin(\omega_c t)$$

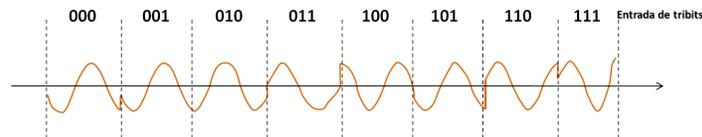
Figure 6.12

Y la Densidad Espectral de Potencia es :

$$G_{mpsk}(f) = \frac{1}{2} Ntb \cdot \frac{V^2}{2} (\text{Sinc}^2(Ntb(f - f_c)) + \text{Sinc}^2(Ntb(f + f_c)))$$

Figure 6.13

Por ejemplo, si queremos realizar la modulación 8-PSK:



Entrada Binaria	Fase de Salida QPSK
000	-112.5°
001	-157.5°
010	-67.5°
011	-22.5°
100	+112.5°
101	+157.5°
110	+67.5°
111	+22.5°

Figure 6.14: Fase de salida para 8-PSK

Sus bases y Constelación serían:

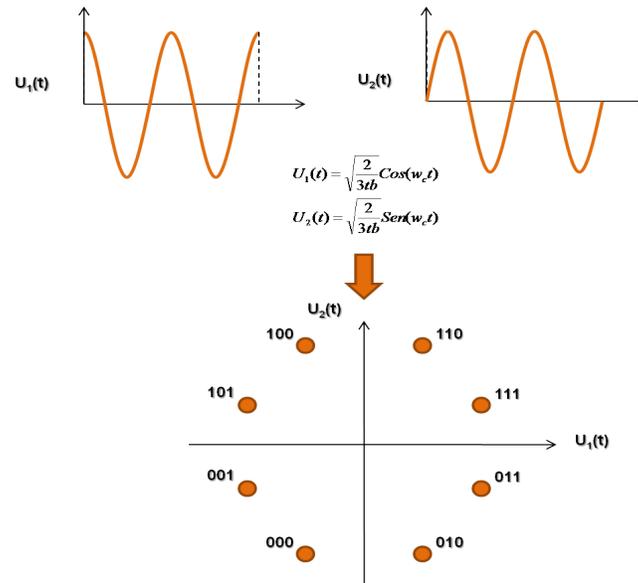


Figure 6.15

Y para la modulación 16-PSK:

Entrada Binaria	Fase de Salida QPSK	Entrada Binaria	Fase de Salida QPSK
0000	11.25°	1000	191.25°
0001	33.75°	1001	213.75°
0010	56.25°	1010	236.25°
0011	78.75°	1011	258.75°
0100	101.25°	1100	281.25°
0101	123.75°	1101	303.75°
0110	146.25°	1110	326.25°
0111	168.75°	1111	348.75°

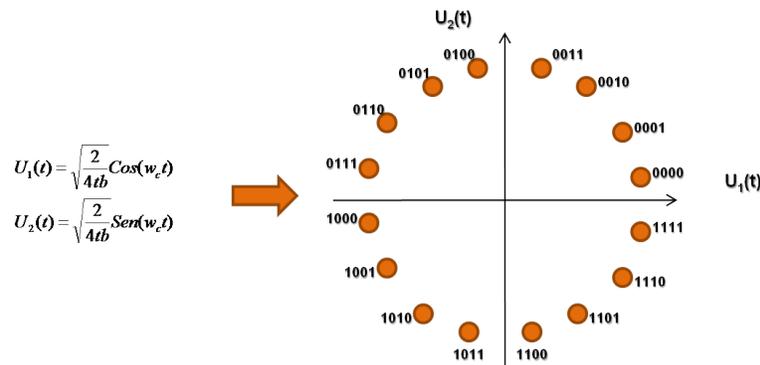


Figure 6.16

6.1.3 3. QAM (Quadrature Amplitude Modulation)

En este tipo de modulación se varían los parámetros de amplitud y fase de la señal portadora transmitida. Básicamente, consiste en modular en amplitud dos señales portadoras que trabajan con la misma frecuencia pero se encuentran desfasadas 90° entre sí lo que produce, también, cambios en la fase. En general, su ecuación viene dada por:

$$X_{QAM}(t) = a_r \cos(w_c t) + b_r \sin(w_c t)$$

Figure 6.17

En donde a_n y b_n son valores discretos correspondientes al total de los N bits codificados.

Supongamos que tomamos el valor $M=8 \Rightarrow 8\text{-QAM}$

Los datos que llegan antes del modulador se dividen en combinaciones de tres bits (tribits). Se definen entonces dos bases ortonormales para representar la señal:

$$U_1(t) = \sqrt{\frac{2}{Ntb}} \text{Sen}(w_c t)$$

$$U_2(t) = \sqrt{\frac{2}{Ntb}} \text{Cos}(w_c t)$$

$$N = 3$$

Figure 6.18

Por lo que la ecuación de QAM en el tiempo quedaría así;

$$X_{QAM}(t) = k1.aU_1(t) + k2.aU_2(t)$$

Figure 6.19

Siendo a una constante y $k1$ y $k2$ señales Non-Return to Zero (NRZ) multinivel con tiempo de duración $N \cdot tb$. Su constelación es como sigue:

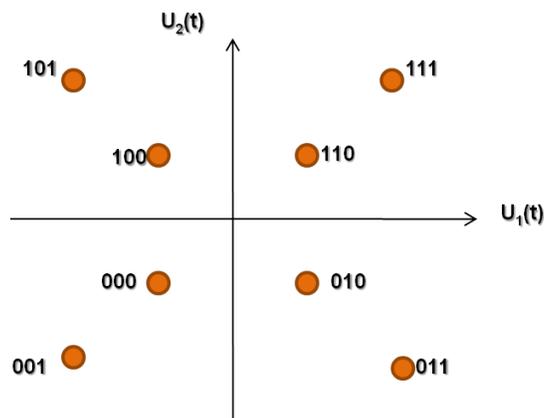


Figure 6.20: Constelación para 8-QAM.

Las formas de onda correspondientes a las combinaciones de bits quedarían así:

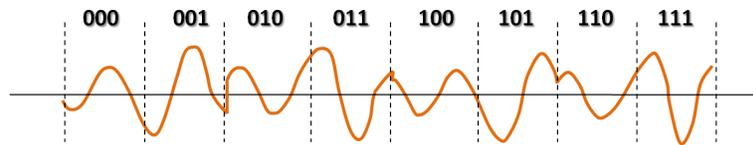


Figure 6.21

Y la ecuación de 8-QAM en el dominio de la frecuencia:

$$G_{QAM}(f) = \frac{1}{2} 3tb \cdot \frac{V^2}{2} (\text{Sinc}^2(3tb(f - fc)) + \text{Sinc}^2(3tb(f + fc)))$$

Figure 6.22

Si ahora tomamos el valor $M=16 \Rightarrow 16\text{-QAM}$

$$U_1(t) = \sqrt{\frac{2}{Ntb}} \text{Sen}(w_c t)$$

$$U_2(t) = \sqrt{\frac{2}{Ntb}} \text{Cos}(w_c t)$$

$$N = 4$$

Figure 6.23

Su constelación:

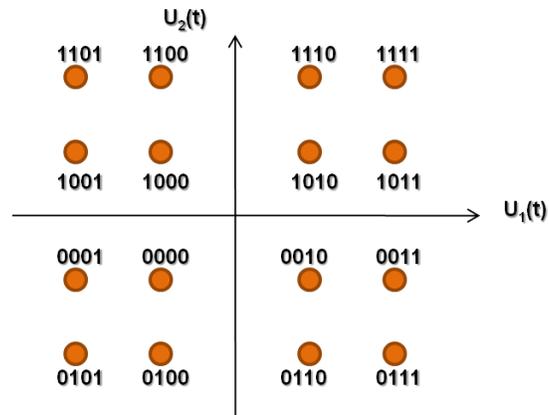


Figure 6.24: Constelación para 16-QAM

Entrada Binaria	Salida 16-QAM	
0000	0.311V	-135°
0001	0.850V	-165°
0010	0.311V	-45°
0011	0.850V	-15°
0100	0.850V	-105°
0101	1.161V	-135°
0110	0.850V	-75°
0111	1.161V	-45°
1000	0.311V	135°
1001	0.850V	165°
1010	0.311V	45°
1011	0.850V	15°
1100	0.850V	105°
1101	1.161V	135°
1110	0.850V	75°
1111	1.161V	45°

Figure 6.25

Y la Densidad Espectral de Potencia sería:

$$G_{16-QAM}(f) = \frac{1}{2} 4tb \cdot \frac{V^2}{2} (\text{Sinc}^2(4tb(f - fc)) + \text{Sinc}^2(4tb(f + fc)))$$

Figure 6.26

6.1.4 Simulación en LabVIEW

para observar los tipos de modulación previamente explicados, se debe descargar el código fuente del VI m-arias.vi a través del siguiente enlace: [MEDIA OBJECT]³

6.1.5 Demostración

[MEDIA OBJECT]⁴

³This media object is a downloadable file. Please view or download it at <marias.zip>

⁴This media object is a video file. Please view or download it at <m-arias.mpg>

Chapter 7

7. Introducción del ruido en los sistemas de Comunicaciones Digitales: Detección Óptima y Probabilidad de Error¹

7.1 INTRODUCCIÓN DEL RUIDO EN LOS SISTEMAS DE COMUNICACIONES DIGITALES: DETECCIÓN ÓPTIMA Y PROBABILIDAD DE ERROR

González C. Y. Venuska

Mezoa R. Mariangela

Resumen

El contenido de este módulo abarca la introducción del ruido en los sistemas de comunicación digital, específicamente el ruido AWGN (*Additive White Gaussian Noise*). Se observarán los cambios que produce en las señales codificadas y moduladas en formato tanto binario como m-ario y se usará el procedimiento de Detección Coherente (Óptima) para la recepción de las mismas.

En un sistema digital (y en cualquier sistema de comunicación en general) se desea que el mensaje a transmitir llegue completo y sin problemas hasta el destino. Sin embargo, debido a ciertos factores como:

- Limitación del ancho de banda del canal;
- Medio de transmisión que se use;
- Interferencias producidas por el hombre o por la naturaleza del medio, entre otros.

Existe la posibilidad de que el mensaje llegue errado o incompleto (o incluso que el mismo no llegue a su destino). De los factores mencionados nos encontramos con uno que siempre está presente en el sistema, independientemente de si hay una señal de entrada o no: este se llama **Ruido No Correlacionado**.

De este tipo de ruido resalta uno que es el que tomaremos en cuenta para este módulo: **Ruido AWGN (Blanco Gaussiano Aditivo)**.

Definition 7.1: AWGN

Corresponde a un modelo en el que al canal de comunicaciones se le suma ruido blanco en banda base, con una densidad espectral constante ($[U+F068] / 2$) y una amplitud de distribución Gaussiana.

Antes de observar los cambios que genera el ruido AWGN sobre el sistema, debemos resaltar el procedimiento de detección óptima o coherente:

¹This content is available online at <http://cnx.org/content/m36883/1.4/>.

7.1.1 Detección Coherente

Este tipo de detección se obtiene cuando en el receptor se usa un filtro óptimo cuya respuesta impulsiva estará relacionada con las formas de onda transmitidas codificadas o moduladas.

Supongamos que se tiene una señal de entrada (modulada) $\mathbf{p}(\mathbf{t})$, y que al pasar por el filtro óptimo, se obtiene la señal de salida $\mathbf{y}(\mathbf{t})$. A la entrada también se le suma el ruido $\mathbf{n}(\mathbf{t})$, que al pasar por el filtro se llamará $\mathbf{n}_{out}(\mathbf{t})$.



Figure 7.1

El objetivo de esta detección es la de maximizar la relación entre la señal modificada por el filtro $\mathbf{y}(\mathbf{t}_0)$ en un tiempo de muestreo específico t_0 y el voltaje *rms* del ruido a la salida (σ):

$$\frac{|y(t_0)|^2}{\sigma^2} = \frac{|\int_{-\infty}^{\infty} P(f) \cdot H_R(f) \cdot e^{j\omega t_0} df|^2}{\int_{-\infty}^{\infty} G_n(f) |H_R(f)|^2 df} \quad (7.1)$$

Esta ecuación puede simplificarse si usamos el concepto de la *desigualdad de Schwartz*:

$$\begin{aligned} |\int_{-\infty}^{\infty} V(f) \cdot W(f) df|^2 &\leq \int_{-\infty}^{\infty} |V(f)|^2 df \cdot \int_{-\infty}^{\infty} |W(f)|^2 df \\ \text{Si: } V(f) &= k \cdot W(f) \end{aligned} \quad (7.2)$$

$$\int_{-\infty}^{\infty} |W(f)|^2 df = \frac{|\int_{-\infty}^{\infty} V(f) \cdot W(f) df|^2}{\int_{-\infty}^{\infty} |V(f)|^2 df}$$

Pudiéramos entonces asignar los siguientes valores en base a la función maximizada:

$$\begin{aligned} V(f) &= \sqrt{G_n(f)} H_R(f) \\ V(f) \cdot W^*(f) &= \sqrt{G_n(f)} H_R(f) \cdot W^*(f) = P(f) \cdot H_R(f) \cdot e^{j\omega t_0} \\ W(f) &= \frac{P^*(f) e^{-j\omega t_0}}{\sqrt{G_n(f)}} \end{aligned} \quad (7.3)$$

Para cumplir con la igualdad:

$$V(f) = kW(f) = k \cdot \frac{P^*(f) e^{-j\omega t_0}}{\sqrt{G_n(f)}} \quad (7.4)$$

Finalmente, tenemos que la ecuación del filtro óptimo es:

$$H_r(f) = k \cdot \frac{P^*(f)e^{-j\omega t_0}}{Gn(f)}$$

$$\frac{|y(t_0)|^2}{\sigma^2} = \int_{-\infty}^{\infty} |W(f)|^2 df = \int_{-\infty}^{\infty} \frac{|P(f)|^2}{Gn(f)} df$$

Figure 7.2

Cuando el ruido que se introduce en el sistema es AWGN, el filtro toma el nombre de **filtro adaptado** ya que su respuesta impulsiva tiene la forma de $p(t)$ (pulso transmitido). En este caso:

$$\frac{|y(t_0)|^2}{\sigma^2} = \frac{2}{\eta} \int_{-\infty}^{\infty} |P(f)|^2 df = \frac{2}{\eta} E$$

Donde :

$$E_x = \int_{-\infty}^{\infty} |X|^2(f) df = S \cdot ts$$

S = Potencia

ts = Tsímbolo

(7.5)

$$\Rightarrow \frac{|y(t_0)|^2}{\sigma^2} = \frac{2}{\eta} S \cdot ts = \frac{S}{\eta B} = \left(\frac{S}{N} \right)$$

7.1.2 Probabilidad de Error:

La probabilidad de error es la expectativa de que cierto sistema contenga una tasa de errores. Por ejemplo, si se obtiene una Probabilidad de Error de 10^{-4} , quiere decir que por cada 10.000 transmitidos se tiene, en promedio, 1 bit errado.

Ahora bien, debemos considerar la situación en la que se tiene una señal que se desea pasar por el proceso de detección óptima para obtener el mensaje original en el destino. Para eso tomamos como condición inicial el cálculo de la probabilidad de error por bit de una señal polar en bandabase.

Para un sistema binario, supongamos que a la salida del receptor se tienen dos niveles de amplitud: $0.5A$ y $-0.5A$ más el ruido blanco. La probabilidad de error quedaría así:

$$Pe = \frac{1}{2}P(0.5A + n < \text{Umbral}) + \frac{1}{2}P(-0.5A + n > \text{Umbral}) \quad (7.6)$$

Como los valores de salida ($0.5A$ y $-0.5A$) son equiprobables, es decir, tienen la misma probabilidad de ocurrencia, entonces el valor del umbral es CERO:

$$\text{Umbral} = \frac{1}{2}(0.5A + (-0.5A)) = 0 \quad (7.7)$$

Por lo que la probabilidad de error quedaría:

$$Pe = \frac{1}{2}P(0.5A + n < 0) + \frac{1}{2}P(-0.5A + n > 0) = P(n > 0.5A) \quad \text{Ecuación (a)} \quad (7.8)$$

En general, para un sistema bandabase polar con $y(t)$ de salida, tendríamos que la probabilidad de error es:

$$Pe = P(n > y(t_o)) \quad (7.9)$$

Siendo t_o el momento donde se toma la decisión.

Partiendo de la ecuación (a) podemos obtener las probabilidades de error correspondientes a cada modulación binaria en la cual los pulsos para el "0" ($p_0(t)$) y para el "1" ($p_1(t)$) toman formas diferentes. Entonces la probabilidad de error puede calcularse en función de los pulsos a la salida $p_{0o}(t_o)$ y $p_{1o}(t_o)$:

$$Pe = \frac{1}{2} (P_{0o}(t_o) + n > \text{Umbral}) + \frac{1}{2} (P_{1o}(t_o) + n < \text{Umbral})$$

$y :$

$$\text{Umbral} = \frac{1}{2} (P_{0o}(t_o) + P_{1o}(t_o))$$

Entonces :

$$(7.10)$$

$$Pe = \frac{1}{2} (P_{0o}(t_o) + n > \frac{1}{2} (P_{0o}(t_o) + P_{1o}(t_o))) + \frac{1}{2} (P_{1o}(t_o) + n < \frac{1}{2} (P_{0o}(t_o) + P_{1o}(t_o)))$$

$$\Rightarrow Pe = (n > \frac{1}{2} (P_{1o}(t_o) - P_{0o}(t_o)))$$

Haciendo uso de la fórmula para el filtro óptimo podemos hallar la probabilidad de error. El valor de $p(t)$ lo sustituimos por $\frac{1}{2}(p_1(t)-p_0(t))$ (Que serán los dos posibles pulsos de entrada correspondientes al "1" lógico y "0" lógico) y cuando se consiga la representación en frecuencia se sustituirá por su equivalente en frecuencia

$$H_R(f) = k \cdot \frac{P^*(f)e^{-j\omega t_o}}{Gn(f)}$$

$$\frac{|y(t_o)|^2}{\sigma^2} = \int_{-\infty}^{\infty} |W(f)|^2 df = \int_{-\infty}^{\infty} \frac{|P(f)|^2}{Gn(f)} df$$


$$H_R(f) = k \cdot \frac{[P_1(f) - P_0(f)]^*(f)e^{-j\omega t_o}}{2 \cdot Gn(f)}$$

$$\frac{|y(t_o)|^2}{\sigma^2} = \int_{-\infty}^{\infty} \frac{|P_1(f) - P_0(f)|^2}{4 \cdot Gn(f)} df$$

Figure 7.3

$$H_R(f) \Rightarrow h_R(t)$$

$$h_R(t) = k \cdot \frac{[p_1(t_o-t) - p_0(t_o-t)]}{\eta} \quad (7.11)$$

Entonces :

$$\frac{y^2(t_o)}{\sigma^2} |_{\text{máx}} = \int_{-\infty}^{\infty} \frac{|p_1(t) - p_0(t)|^2}{2\eta} dt = \int_{-\infty}^{\infty} \frac{p_{12}(t)}{2\eta} dt + \int_{-\infty}^{\infty} \frac{p_{02}(t)}{2\eta} dt - 2 \int_{-\infty}^{\infty} \frac{p_1(t) \cdot p_0(t)}{2\eta} dt \quad (b)$$

Se define el coeficiente de correlación entre los pulsos como λ y:

$$\lambda = \frac{1}{\sqrt{E_1 E_0}} \int_{-\infty}^{\infty} p_0(t) p_1(t) dt$$

$$E_1 = \int_{-\infty}^{\infty} p_{12}(t) dt$$

$$E_0 = \int_{-\infty}^{\infty} p_{02}(t) dt \quad (7.12)$$

Por lo que pudiéramos ver la ecuación (b) expresada en función de las definiciones anteriores:

$$\begin{aligned} \frac{y^2(t_0)}{\sigma^2} \Big|_{\text{máx}} &= \frac{1}{2\eta} (E_1 + E_0 - 2\lambda\sqrt{E_1 E_0}) \\ \Rightarrow \text{Pe} &= Q\sqrt{\frac{y^2(t_0)}{\sigma^2}} = Q\left(\sqrt{\frac{1}{2\eta} (E_1 + E_0 - 2\lambda\sqrt{E_1 E_0})}\right) \\ \text{Para } E_1 &= E_0 = E \\ \frac{y^2(t_0)}{\sigma^2} &= \frac{E}{\eta} (1 - \lambda) \quad \text{Pe} = Q\left(\sqrt{\frac{E}{\eta} (1 - \lambda)}\right) \\ \text{Para } \lambda &= 0; \quad p_1 \wedge p_0 : \text{ortogonales} \\ \frac{y^2(t_0)}{\sigma^2} &= \frac{E}{\eta} \quad \text{Pe} = Q\left(\sqrt{\frac{E}{\eta}}\right) \\ \text{Para } \lambda &= -1 \\ \frac{y^2(t_0)}{\sigma^2} &= \frac{2E}{\eta} \quad ; \quad \text{Pe} = Q\left(\sqrt{\frac{2E}{\eta}}\right) \end{aligned} \tag{7.13}$$

7.2 1. DETECCIÓN COHERENTE PARA MODULACIONES BINARIAS

7.2.1 1.1 Detección Coherente para la modulación ASK

Para detectar de forma coherente la señal ASK se necesita un receptor como sigue:

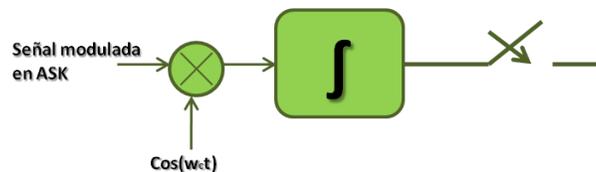


Figure 7.4: Receptor para ASK

Queda claro que la integración y el cierre del interruptor ocurrirá cada t_b .

Dado que trabajamos con un caso especial de ASK: **OOK (On-Off Keying)**, los pulsos $p_1(t)$ y $p_0(t)$ son:

$$\begin{aligned} p_1(t) &= V \prod \left[\frac{t - \frac{tb}{2}}{tb} \right] \text{Cos}(w_c t) \\ p_0(t) &= 0 \end{aligned} \tag{7.14}$$

Entonces:

$$P_{e_{OOK}} = Q\left(\sqrt{\frac{E_p}{\eta}}\right) \quad (7.15)$$

7.2.2 1.2 Detección Coherente para la modulación PSK

El diagrama de bloques para un receptor coherente PRK es:

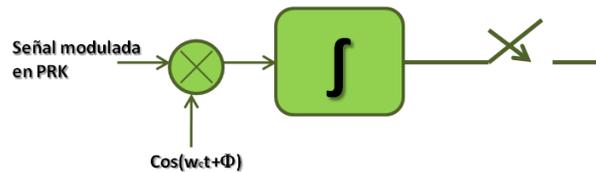


Figure 7.5: Receptor para PRK

Dado que $p_1(t)$ y $p_0(t)$ son señales antípodas (opuestas entre sí), el factor λ será igual a -1, por lo que:

$$P_{e_{PRK}} = Q\left(\sqrt{\frac{2E_p}{\eta}}\right) \quad (7.16)$$

7.2.3 1.3 Detección Coherente para la modulación FSK

Como en la modulación FSK (binaria) se trabaja con dos frecuencias distintas, se estima un proceso de detección para cada frecuencia:

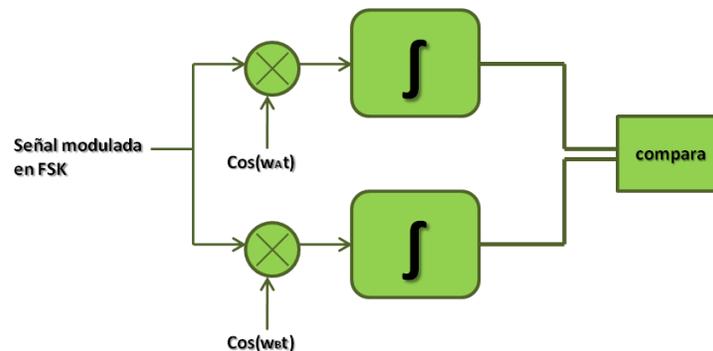


Figure 7.6: Receptor para FSK

Los pulsos $p_1(t)$ y $p_0(t)$ Se diferencian por la frecuencia a la que trabajan, es decir:

$$\begin{aligned} p_1(t) &= V\cos((w_c + \Omega)t) \\ p_0(t) &= V\cos((w_c - \Omega)t) \end{aligned} \quad (7.17)$$

Ahora bien, que si ocurre que $\lambda=0$ (es decir, si las frecuencias son tales que los pulsos son ortogonales) y los factores E_1 y E_0 son iguales, la probabilidad de error para FSK será la misma que para OOK. Es decir:

$$P_{e\text{FSK}} = Q\left(\sqrt{\frac{E_p}{\eta}}\right) \quad (7.18)$$

A continuación se muestra una gráfica que compara los valores de P_e vs E/η . A través de ella se puede determinar la fortaleza de cada modulación ante la inducción del ruido en el sistema.

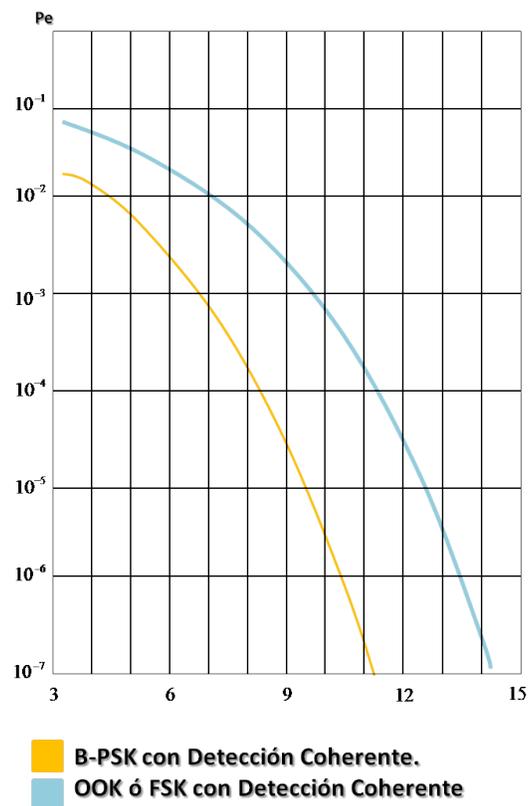


Figure 7.7: Curvas de Probabilidad de Error para modulaciones Binarias.

7.3 2. PROBABILIDAD DE ERROR PARA MODULACIONES M-ARIAS

A continuación se muestran las gráficas de P_e vs E_b / N_0 para diversas modulaciones m-arias utilizando, en todos los casos, un receptor óptimo:

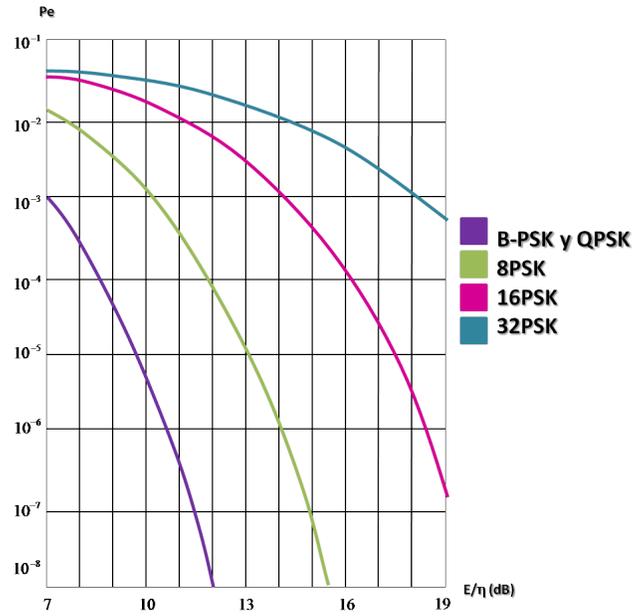


Figure 7.8: Curvas de Probabilidad de Error para modulaciones M-aria PSK.

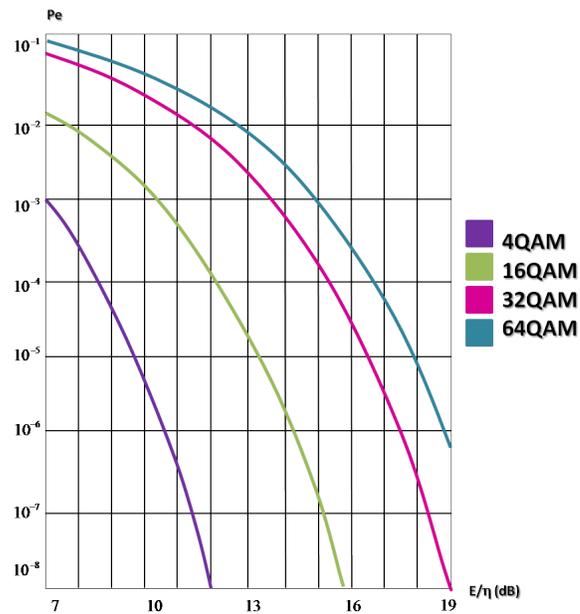


Figure 7.9: Curvas de Probabilidad de Error para modulaciones M-aria QAM.

7.4 Simulaciones en LabVIEW

Para reforzar los conocimientos adquiridos en este módulo, se presentan a continuación una serie de VIs referentes a la detección óptima binaria y M-aria. Es importante resaltar que para la ejecución de cada uno es necesario tener instalado en el ordenador el software MATLAB, debido a que varios procedimientos fueron realizados haciendo uso del módulo de mathscript.

VI de Detección Coherente Binaria

This media object is a downloadable file. Please view or download it at <ASK Coherente.rar>

This media object is a downloadable file. Please view or download it at <FSK Coherente.rar>

This media object is a downloadable file. Please view or download it at <PSK Coherente.rar>

VI de Detección Coherente M-aria

This media object is a downloadable file. Please view or download it at <QAMdeteccionULTIMATE.vi>

This media object is a downloadable file. Please view or download it at <QPSKdeteccionULTIMATE.vi>

Chapter 8

8. Detección no Coherente para modulaciones OOK y FSK¹

8.1 DETECCIÓN NO COHERENTE PARA MODULACIONES OOK Y FSK

González C. Y. Venuska

Mezoa R. Mariangela

Resumen

En este módulo se explica el método de recepción no coherente de las señales OOK (On-Off Keying) y FSK (Frequency Shift Keying), su esquema general y la forma de calcular la Probabilidad de Error para cada sistema.

En el módulo anterior² mencionábamos el método de detección coherente para obtener el mensaje original en el destino de nuestro sistema de comunicaciones. En él, por ejemplo para OOK o para PSK, se multiplicaba la señal modulada por un tono de la misma frecuencia que la señal portadora, pasando luego por un sistema de comparación con un umbral, obteniendo así el mensaje original. Sin embargo, este tipo de detección es muy costosa e innecesaria si la señal es lo suficientemente fuerte para una fiabilidad adecuada. Es por esto (y también para evitar los problemas de sincronización) que se implementan los sistemas de **detección no coherente**.

Antes de observar la detección no coherente para las modulaciones OOK y FSK, se analizará la envolvente de una señal sinusoidal con ruido en Banda Base.

Supongamos que se tiene una sinusoides (portadora) contaminada con ruido pasabanda gaussiano $n(t)$ de varianza σ^2 y media nula, entonces:

$$n(t) = n_i(t) \cos(w_c t) - n_q(t) \sin(w_c t)$$

$$A \cos(w_c t) + n(t) = R(t) \cos(w_c t + \Phi(t)) \quad (a) \tag{8.1}$$

Donde :

$$R = \sqrt{(A + n_i)^2 + n_q^2} \quad (b) \quad ; \quad \Phi = \arctg \frac{n_q}{A + n_i} \quad (c)$$

Como se sabe, las componentes de ruido n_i , n_q son variables aleatorias independientes que tienen la misma

¹This content is available online at <<http://cnx.org/content/m36884/1.1/>>.

²<http://cnx.org/content/m36883/latest/>

distribución que $n(t)$. Nuestro objetivo será buscar la función de densidad de Probabilidad (**fdp**) de R ya que esto es lo que ofrece a la salida un detector de envolvente.

Si $\mathbf{A}=\mathbf{0}$, el valor de R se convierte en una envolvente que llamaremos A_n con distribución de Rayleigh:

$$P_{A_n}(A_n) = \frac{A_n}{\sigma^2} e^{-\frac{A_n^2}{2\sigma^2}}; \quad \text{Si: } A_n > 0 \quad (8.2)$$

Hacemos un paréntesis aquí para recordar el modelo de Rayleigh:

Distribución Rayleigh:

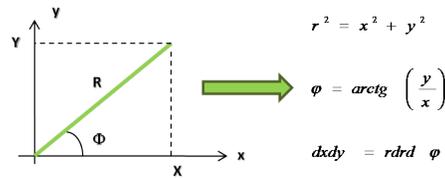
El modelo de Rayleigh describe una variable aleatoria resultante de dos variables aleatorias Gaussianas:

Si X y Y son dos variables aleatorias independientes con media nula y varianza σ^2 , entonces la variable definida como R tiene **distribución de Rayleigh**:

$$R = \sqrt{X^2 + Y^2}$$

Este modelo aplica para cualquier conversión de rectangular a polar, siempre y cuando las coordenadas rectangulares sean idénticas pero gaussianas independientes con media nula. Para hallar su **fdp** se introduce un ángulo aleatorio Φ y se comienza con la relación de fdp conjunta:

$$P_{R\Phi}(r, \varphi) |drd\varphi| = P_{XY}(x, y) |dxdy|$$



Entonces:

$$P_{XY}(x, y) = p_X(x)p_Y(y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

$$P_{R\Phi}(r, \varphi) = \frac{r}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}}$$

Figure 8.1

Si asignamos entonces un valor arbitrario para \mathbf{A} , hacemos la conversión de rectangular a polar para obtener la **fdp** conjunta de \mathbf{R} y Φ :

$$P_{R\Phi}(R, \Phi) = \frac{R}{2\pi\sigma^2} e^{-\frac{R^2 - 2AR\cos\Phi + A^2}{2\sigma^2}} \quad \text{Para: } A \geq 0; |\Phi| \leq \pi \quad (8.3)$$

En la ecuación observamos que \mathbf{R} y Φ no son estadísticamente independientes, por lo que se deben hallar las **distribuciones marginales**, es decir, la **fdp** de R debe calcularse integrando la **fdp** conjunta en el rango de Φ):

$$P_R(R) = \frac{R}{2\pi\sigma^2} e^{-\frac{R^2 + A^2}{2\sigma^2}} \int_{-\pi}^{\pi} e^{\frac{AR\cos\Phi}{\sigma^2}} d\Phi \quad (8.4)$$

Aquí se identifica un elemento relacionado con la **función de Bessel** modificada de primera clase y orden 0:

$$I_0(v) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{v \cos \phi} d\phi$$

Donde

$$I_0(v) \approx \begin{cases} e^{\frac{v^2}{4}} & v \ll 1 \\ \frac{e^v}{\sqrt{2\pi v}} & v \gg 1 \end{cases} \quad (8.5)$$

Por lo que:

$$P_R(R) = \frac{R}{\sigma^2} e^{-\frac{(R^2+A^2)}{2\sigma^2}} I_0\left(\frac{AR}{\sigma^2}\right) \quad R \geq 0 \quad (8.6)$$

Esta última ecuación se conoce como **Distribución Rice** (*Rician Distribution*).

Si $A \gg \sigma$, El valor de R tendrá una menor proporción del ruido, por lo que:

$$R = A \sqrt{1 + \frac{2n_i}{A} + \frac{(n_i^2 + n_q^2)}{A^2}} \approx A + n_i \quad (8.7)$$

Esto implica que la envolvente R será gaussiana ya que n_i lo es.

Para cuando $A \gg \sigma$, la ecuación puede simplificarse como:

$$P_R(R) = \sqrt{\frac{R}{2\pi A \sigma^2}} e^{-\frac{(R^2-A^2)}{2\sigma^2}} \quad A \geq \sigma \quad (8.8)$$

Finalmente, se confirma que la fdp es una curva Gaussiana con varianza σ^2 centrada en $R \approx A$.

En definitiva se tienen 3 posibles distribuciones para la envolvente: Rayleigh cuando $A=0$, Rice cuando A es comparable con la desviación estándar del ruido y Gaussiana cuando $A \gg \sigma$

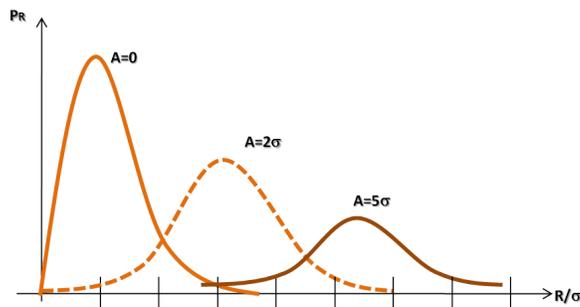


Figure 8.2: fdp de la envolvente R con ruido.

8.1.1 1. Detección No Coherente - OOK

El receptor no coherente para una señal modulada OOK puede expresarse como sigue:

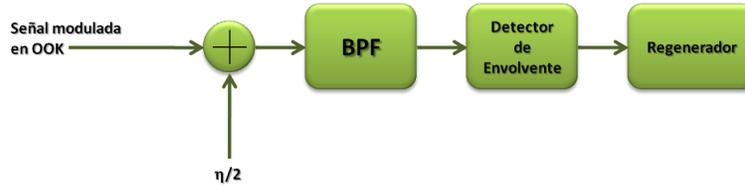


Figure 8.3

Como se observa, la señal modulada OOK será detectada por un filtro pasabanda, seguido por un detector de envoltura. Si se transmite un 0 lógico, el ruido tendrá distribución Rayleigh. Pero, si se transmite un 1 lógico, el ruido puede tener distribución Rice o Gaussiana.

La probabilidad de error será:

$$P_e = P() P(T_1) + P() P(T_0) = P(A + n_1 < \text{Umbral}) P(T_1) + P(n_2 > \text{Umbral}) P(T_0) \quad (8.9)$$

La probabilidad de error tal que se transmite un 1 se calculará usando la distribución Rice. La probabilidad de error tal que se transmite un 0 será calculada usando la distribución de Rayleigh.

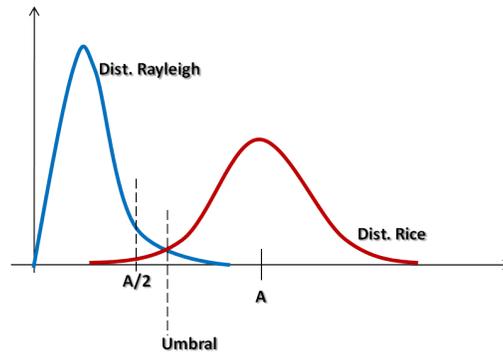


Figure 8.4

Al igualar las áreas obtenemos el Umbral:

$$\begin{aligned} \text{Umbral} &= \frac{A}{2} \sqrt{1 + \frac{8\sigma}{A}} \\ E_1 &= \frac{A^2 t_b}{2} \\ E_0 &= 0 \end{aligned} \tag{8.10}$$

Entonces :

$$\left(\frac{S}{N}\right) = \frac{E_1}{2\eta} = \frac{E_p}{\eta} = \left(\frac{A}{\sigma}\right)^2$$

Si se tiene bajo ruido ($E_p / [U+F068] \gg 1$):

$$P(e/T_0) = \int_0^{\infty} P_R(\text{Rayleigh}) dR = e^{-\frac{A^2}{8\sigma^2}} = e^{-\frac{E_p}{2\eta}} \tag{8.11}$$

$$P(e/T_1) = \int_0^{\infty} P_R(\text{Gauss}) dR \approx Q\left(\sqrt{\frac{E}{\eta}}\right) \approx \frac{1}{\sqrt{2\pi}\sqrt{\frac{E}{\eta}}} e^{-\frac{E_p}{2\eta}}$$

Finalmente:

$$P_{e\text{OOK}} = \frac{1}{2} e^{-\frac{E_p}{2\eta}} \tag{8.12}$$

A continuación se comparan las curvas de probabilidad de error cuando se usa detector coherente y no coherente para OOK:

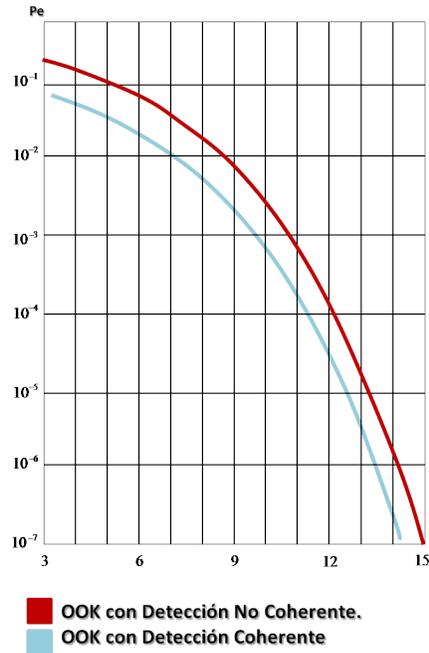


Figure 8.5

Como se observa, OOK con detección coherente presenta **menor** valor de Probabilidad de error para un determinado valor $E/[U+F068]$. Para que OOK con detección no coherente tenga la misma probabilidad de error que con detección coherente es necesario que se transmita más potencia.

8.1.2 2. Detección No Coherente – FSK

El sistema de detección no coherente para una señal FSK es:

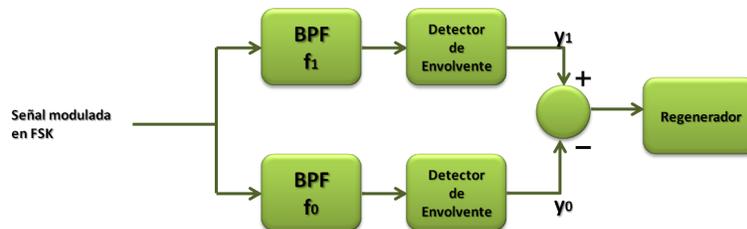


Figure 8.6

La señal de entrada FSK entra de forma simultánea a los dos filtros Pasabanda. Cada filtro pasará la frecuencia asignada para un 1 lógico o un 0 lógico a su detector de envolvente correspondiente. Cuando llegue la señal modulada correspondiente al "0" y_1 será pequeño y y_0 será grande; la resta será negativa. Para un "1" la situación será opuesta.

La probabilidad de error tal que se transmite un 1 se calculará usando la distribución Rice. La probabilidad de error tal que se transmite un 0 será calculada usando la distribución de Rayleigh.

La regeneración de la señal está basada en la diferencia de envolventes $\mathbf{Y1-Y0} = \mathbf{y1(t)-y0(t)}$. Dada la simetría del receptor, no importa el valor de A, La $P(e/T_1)$ será igual a:

$$P() = P(Y_1 - Y_0 < 0 | H_1) = P(Y_0 > Y_1 | H_1)$$

$$\begin{aligned} \text{Como } P() = \bar{P}() = P_e \\ P_e = \int_0^{\infty} P_{Y_1}(Y_1 | H_1) \left[\int_{Y_1}^{\infty} P_{Y_0}(Y_0 | H_1) dy_0 \right] dy_1 \end{aligned} \quad (8.13)$$

$$P_e = \int_0^{\infty} \frac{y_1}{\sigma^2} e^{-I_0\left(\frac{Ay_1}{\sigma^2}\right)} dy_1 \quad (d)$$

Si hacemos los cambios de variable:

$$\begin{aligned} \lambda &= \sqrt{2}y_1 \\ \alpha &= \frac{A}{\sqrt{2}} \end{aligned} \quad (8.14)$$

Y los sustituimos en la ecuación (d), tendremos que el integrando de la Probabilidad de Error total de FSK es exactamente la misma función correspondiente a la fdp de Distribución Rice. Por lo tanto:

$$P_e = \frac{1}{2} e^{-\frac{\lambda}{\sigma^2}} \int_0^{\infty} \frac{\lambda}{\sigma^2} e^{-I_0\left(\frac{\alpha\lambda}{\sigma^2}\right)} d\lambda \quad (8.15)$$

$$P_{e\text{FSK}} = \frac{1}{2} e^{-\frac{E_p}{2\eta}} = \frac{1}{2} e^{-\frac{E_p}{2\eta}}$$

Se observa que las Probabilidades de Error para FSK y OOK no coherentes son prácticamente iguales. Sólo se diferenciarán para cuando $E_p/[U+F068]$ sea un valor muy pequeño.

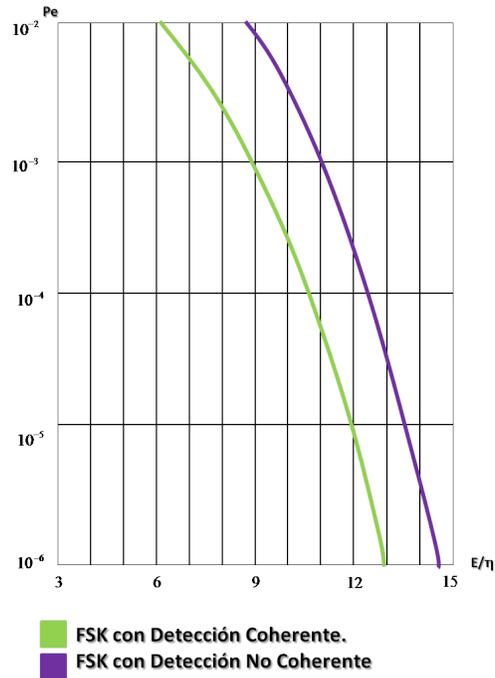


Figure 8.7

Entre los dos sistemas, quien presenta menor probabilidad de error es FSK con Detección Coherente, lo que la hace más fuerte contra el ruido.

8.1.3 Simulaciones en LabVIEW

Los VIs correspondientes a la detección No Coherente pueden descargarse en los siguientes enlaces:

VI de Detección No Coherente OOK

[MEDIA OBJECT]³

VI de Detección No Coherente FSK

[MEDIA OBJECT]⁴

³This media object is a downloadable file. Please view or download it at <ASK no Coherente.rar>

⁴This media object is a downloadable file. Please view or download it at <FSK no Coherente.rar>

Chapter 9

9. Interferencia Intersimbólica (ISI)¹

9.1 INTERFERENCIA INTERSIMBÓLICA (ISI)

González C. Y. Venuska

Mezoa R. Mariangela

Resumen

En este módulo se introducirá el concepto de la interferencia intersimbólica dentro de un sistema de comunicación digital. Se considerarán los dos criterios de Nyquist y se explicarán los pasos para poder calcular la fdp de la ISI y su probabilidad de Error.

Cuando deseamos transmitir un mensaje, el objetivo primordial es que éste sea recibido de manera íntegra. Sin embargo, debido a ciertas limitaciones en el sistema, se dan los casos en el que el mensaje llega completamente distorsionado. Pongamos el siguiente ejemplo: Supongamos que se tiene una señal a la entrada de un filtro Pasabajos:

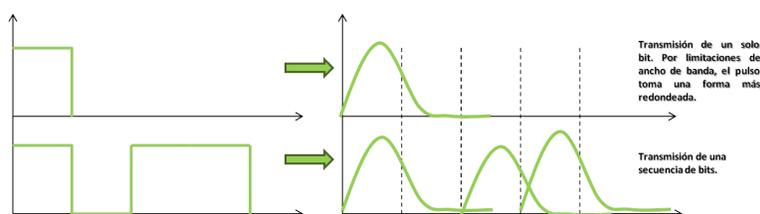


Figure 9.1: Respuestas al impulso.

El caso ideal sería un filtro que no introduce ningún tipo de distorsión de fase o amplitud a la señal, por lo que la señal a la salida del filtro deberá ser igual a la señal de entrada. Pero, en el caso real, el filtro es imperfecto, por lo que la respuesta de salida se aproximará a la parte derecha de la imagen mostrada. Se observa que los pulsos de salida se ‘chorrean’ en el tiempo, interfiriendo con los pulsos siguientes. Dicho de otra manera, los extremos de los pulsos se solapan, interfiriendo con el lóbulo principal del pulso. Esto es lo que se conoce como **Interferencia Intersimbólica** (*intersymbol interference, ISI*): los pulsos rectangulares no mantendrán su forma siempre que el ancho de banda para el proceso de transmisión sea finito. Mientras más pequeño sea el ancho de banda, los pulsos se dispersarán, interfiriendo con el siguiente pulso transmitido.

¹This content is available online at <<http://cnx.org/content/m36885/1.1/>>.

Esta interferencia puede atribuirse a cuatro causas principales: problemas de sincronismo, distorsión de amplitud o de fase, o limitación del ancho de banda del canal

La *FDP* de la ISI se calculará mediante el ejemplo de transmisión polar banda base siguiente: Cuando un pulso pasa por la cadena conformada por Tx-Canal-Rx, se verá afectado por retardos o limitaciones del canal:

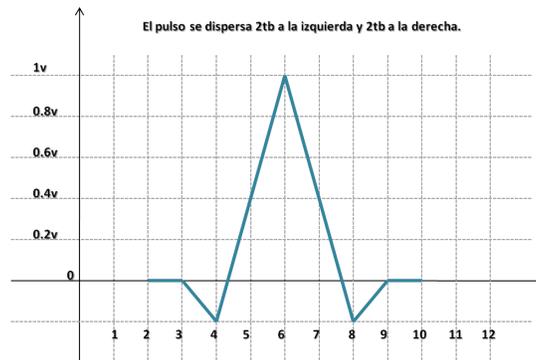


Figure 9.2

Si transmitimos una secuencia de pulsos:

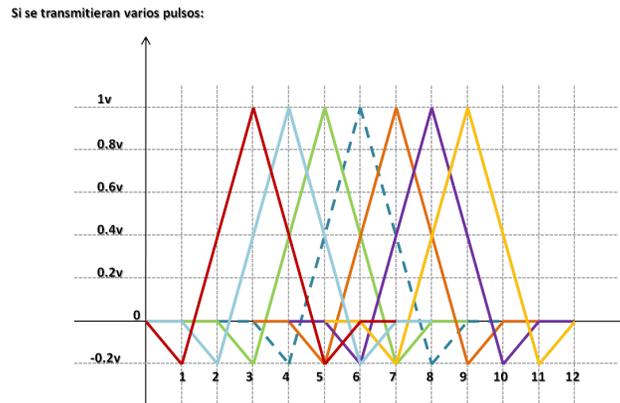


Figure 9.3

Observe que en este ejemplo $t_b=1$ seg. El pulso punteado (centrado en $t=6$ seg) será el que aislaremos. Tal como lo muestra la figura, se ve afectado por sus pulsos vecinos, por lo que se deberá calcular el valor total de ISI en el período donde el pulso esté dispersado (es decir, de $6\text{seg}-2t_b$ hasta $6\text{seg}+2t_b$).

Se deben tomar los voltajes **de todos los pulsos** que coincidan con el centro del pulso aislado ($t=6$ seg). A primera vista se observa que los pulsos amarillo y rojo tienen voltaje 0 para 6 seg., por lo tanto no lo

interfieren

9.1.1 Paso 1:

Primero tomamos el pulso que se encuentra a $-2tb$ a la izquierda (azul claro). Para $t=6\text{seg.}$, toma el valor de $-0.2V$. Pero si el pulso fuese el de polaridad opuesta, el valor sería de $0.2V$. Luego tomamos el pulso que está a $-tb$ a la izquierda (verde). Para $t=6\text{seg.}$ toma un valor de $0.4V$. Si fuese el de polaridad opuesta, tendría un valor de $-0.4V$.

9.1.2 Paso 2:

Repetimos el procedimiento del paso 1, pero tomando en consideración los pulsos que estén a $+tb$ y $+2tb$ del aislado (naranja y morado, respectivamente).

9.1.3 Paso 3:

Elaboramos una tabla que contenga todas las combinaciones de ISI posibles. Dado que el pulso aislado se ve afectado por cuatro vecinos (dos a su izquierda y dos a su derecha), cada uno con dos valores posibles, el total de combinaciones será de dieciséis (16):

-2tb	-tb	tb	2tb	TOTAL ISI	Probabilidad de ocurrencia
-0.2V	0.4V	0.4V	-0.2V	0.4	1/16
-0.2V	0.4V	0.4V	0.2V	0.8	1/16
-0.2V	0.4V	-0.4V	-0.2V	-0.4	1/16
-0.2V	0.4V	-0.4V	0.2V	0	1/16
-0.2V	-0.4V	0.4V	-0.2V	-0.4	1/16
-0.2V	-0.4V	0.4V	0.2V	0	1/16
-0.2V	-0.4V	-0.4V	-0.2V	-1.2	1/16
-0.2V	-0.4V	-0.4V	0.2V	-0.8	1/16
0.2V	0.4V	0.4V	-0.2V	0.8	1/16
0.2V	0.4V	0.4V	0.2V	1.2	1/16
0.2V	0.4V	-0.4V	-0.2V	0	1/16
0.2V	0.4V	-0.4V	0.2V	0.4	1/16
0.2V	-0.4V	0.4V	-0.2V	0	1/16
0.2V	-0.4V	0.4V	0.2V	0.4	1/16
0.2V	-0.4V	-0.4V	-0.2V	-0.8	1/16
0.2V	-0.4V	-0.4V	0.2V	-0.4	1/16

Table 9.1

Finalmente:

Total ISI	Probabilidad
0	4/16
0.4	3/16
-0.4	3/16
0.8	2/16
-0.8	2/16
1.2	1/16
-1.2	1/16

Table 9.2

Es a partir de esta tabla que se puede graficar la fdp:

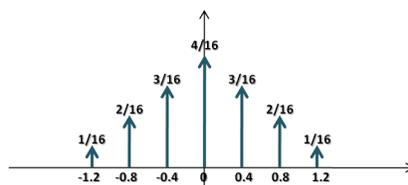


Figure 9.4

Para el cálculo de la probabilidad de error se deben considerar las combinaciones de la primera tabla que pudieran provocar la decisión incorrecta. El pulso punteado toma como máximos valores 1 (para el “1”) y -1 (para el “0”). Cuando a estos valores se les suma la ISI podrá ocurrir un error. Por ejemplo si se transmitió un “1” (1 volt.) y la ISI vale -1.2, al sumar $1-1.2=-0.2$. Como el umbral está en cero se tomará la decisión de que lo transmitido fue un “0” y se comete un error. Si se analizan todos los casos posibles, existen dos posibilidades de error por interferencia: que se transmita un “1” y la ISI sea de -1.2V o que se transmita un “0” y la ISI dé 1.2V. Por lo tanto:

$$P_e = \frac{1}{2}P() + \frac{1}{2}P() = 2 \left(\frac{1}{2} \cdot \frac{1}{16} \right) = \frac{1}{16} \quad (9.1)$$

Aún cuando la interferencia Intersimbólica es crítica al momento de recibir una señal, pudiéramos disminuir (o eliminar) los errores de decisión a través de ciertos métodos. Como se sabe, una señal codificada $y(t)$ puede representarse como la convolución de una secuencia de impulsos aleatorios $x(t)$ con un pulso conocido $p(t)$ (determinístico). Este pulso se puede variar de forma tal que se controle interferencia: Para ello presentaremos los siguientes criterios:

9.1.4 Primer Criterio de Nyquist

El pulso determinístico $p(t)$ debe cumplir con los siguientes parámetros:

$$p(t) = \begin{cases} 1 & ; \quad t = 0 \\ 0 & ; \quad t = \pm nt_b \end{cases} \quad (9.2)$$

$$BW = \frac{1}{2t_b}$$

Una posibilidad es que $p(t)$ sea un Sinc lo que arrojaría un pulso rectangular en frecuencia como se muestra en la figura:

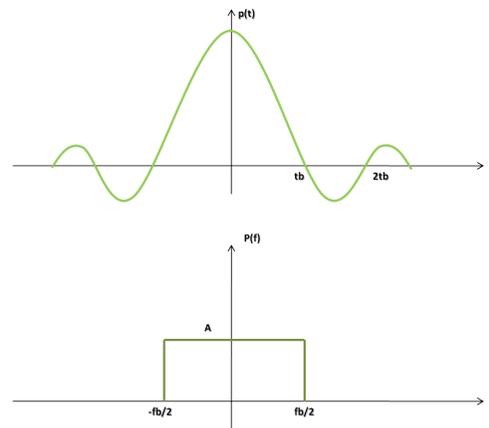


Figure 9.5: Pulso determinístico en los dominios de tiempo y frecuencia.

Sin embargo, esto aplicaría para un caso ideal. Para el caso real, pudiéramos considerar un ancho de banda ligeramente mayor. Para determinar que tipo de pulsos, como éste, pueden ser usados para evitar la ISI se plantea el problema en el dominio del tiempo y luego se llega a una condición en el dominio de la frecuencia. En el receptor lo que se hace es muestrear cada t_b (sería como convolucionar $p(t)$ con una sumatoria infinita de impulsos); lo que se desea es que al muestrear cada pulso y sus vecinos, solo quede el valor del pulso en el instante de muestreo de interés. Por ejemplo suponga que estamos tomando el valor en $t=0$

$$p(t) \sum \delta(t - kt_b) = p(0)\delta(t) = \delta(t)$$

$$P(f) * \frac{1}{t_b} \sum \delta(f - nf_b) = \frac{1}{t_b} \sum P(f - nf_b) = 1$$

Figure 9.6

Esto implica que al sumar todas las repeticiones de $P(f)$ cada f_b , estas deben sumar una constante

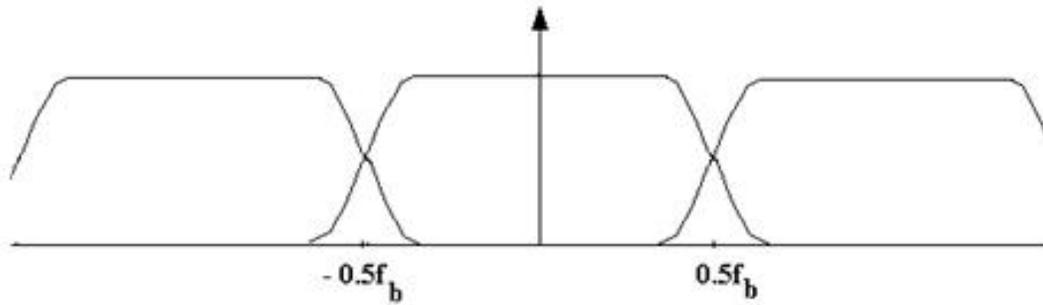


Figure 9.7

El pulso resultante tiene simetría vestigial

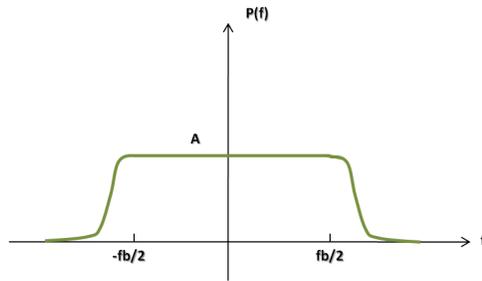


Figure 9.8

Este filtro puede representarse como la suma de:

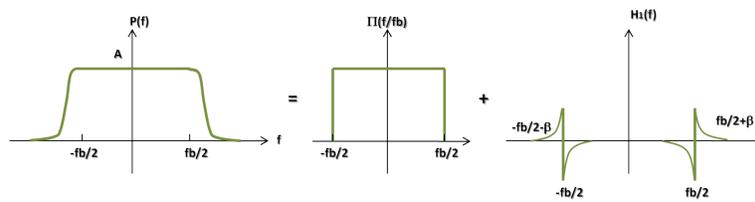


Figure 9.9

Matemáticamente:

$$\begin{aligned}
 P(f) &= \Pi\left(\frac{f}{f_b}\right) + H_1(f) \\
 p(t) &= \frac{1}{t_b} \text{Sinc}\left(\frac{t}{t_b}\right) + h_1(t)
 \end{aligned}
 \tag{9.3}$$

$H_1(f)$ es simétrica y par.

$$h_1(t) = \int_{-\infty}^{\infty} H_1(f) e^{j\omega t} df = 2 \int_0^{\infty} H_1(f) \text{Cos}(\omega t) df$$

$$h_1(t) = 2 \int_{-\beta}^{\beta} H_1(f) \text{Cos}(\omega t) df + 2 \int_0^{+\beta} H_1(f) \text{Cos}(\omega t) df$$

Paralaprimeraintegral : $f = -x$

c.d.v : {

Paralasegundaintegral : $f = +x$

Entonces :

$$h_1(t) = 2 \int_0^{\beta} H_1\left(\frac{fb}{2} - x\right) \text{Cos}\left(2\pi\left(\frac{fb}{2} - x\right)t\right) dx + 2 \int_0^{\beta} H_1\left(\frac{fb}{2} + x\right) \text{Cos}\left(2\pi\left(\frac{fb}{2} + x\right)t\right) dx \quad (9.4)$$

$$\text{Porsimetría : } H_1\left(\frac{fb}{2} + x\right) = -H_1\left(\frac{fb}{2} - x\right)$$

$$h_1(t) = 2 \int_0^{\beta} H_1\left(\frac{fb}{2} + x\right) [\text{Cos}\left(2\pi t\left(\frac{fb}{2} + x\right)\right) - \text{Cos}\left(2\pi t\left(\frac{fb}{2} - x\right)\right)] dx$$

$$h_1(t) = -4 \int_0^{\beta} H_1\left(\frac{fb}{2} + x\right) [\text{Sen}(\pi f_b t) \cdot \text{Sen}(2\pi x t)] dx$$

$$h_1(t) = -4 \text{Sen}(\pi f_b t) \int_0^{\beta} H_1\left(\frac{fb}{2} + x\right) \text{Sen}(2\pi x t) dx$$

Para cada **ntb** el término que se encuentra fuera de la integral se anulará. De esta forma, se evita la interferencia.

A partir de este criterio podemos implementar el filtro de simetría vestigial de tipo **Coseno Alzado**: Este se caracteriza porque puede reducir la ISI. La parte no nula del espectro es un coseno que, en su forma más simple, está alzado (es decir, se encuentra por encima del eje de frecuencia):

$$|P(f)| \begin{cases} 1 & \text{para } |f| \leq \frac{f_b}{2} - \beta \\ \frac{1}{2} \left[1 - \text{Cos} \frac{\pi \left(|f| - \frac{f_b}{2} + \beta \right)}{2\beta} \right] & \text{para } \frac{f_b}{2} - \beta < |f| < \frac{f_b}{2} + \beta \\ 0 & \text{para } |f| \geq \frac{f_b}{2} + \beta \end{cases}$$

Figure 9.10

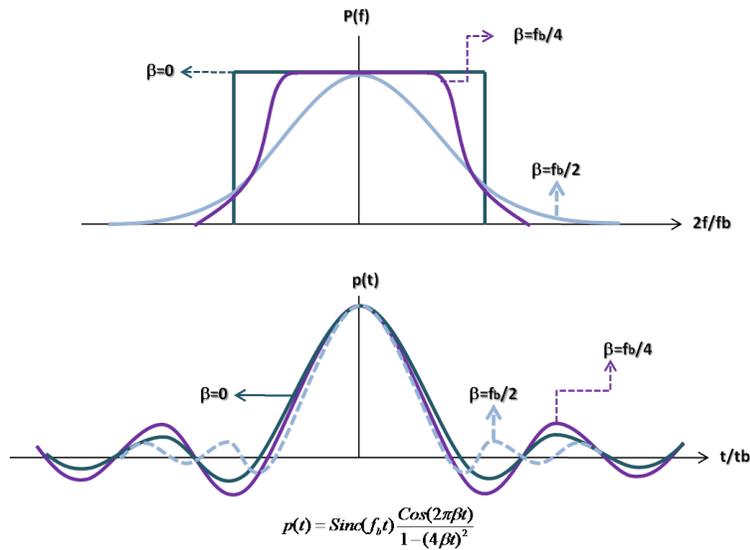


Figure 9.11: Representación del pulso en los dominios de frecuencia y tiempo.

9.1.5 Segundo Criterio de Nyquist

En este criterio se busca no sólo eliminar la interferencia, también se presenta como objetivo el disminuir el ancho de banda. Esto se hace definiendo, en el transmisor, una interacción conocida entre pulsos vecinos. El sacrificio, en este caso, es un mayor consumo de potencia.

Entonces, en vez de transmitir a_k (secuencia original), se enviará $y_k = a_k + a_{k-1}$. De esta forma se pueden enviar dos bits haciendo uso del mismo ancho de banda. Supongamos el siguiente ejemplo:

Secuencia original de bits: **01010011**

Secuencia original	0	1	0	1	0	0	1	1
ak	-1	1	-1	1	-1	-1	1	1
yk		0	0	0	0	-2	0	2

Table 9.3

El filtro que se coloca en el transmisor pudiera modelarse como:

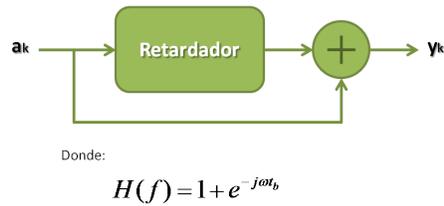


Figure 9.12

Teniendo considerada la condición de un sistema con interferencia, ahora se debe tomar en cuenta cuando se introduce ruido AWGN al canal. Supongamos que a la entrada de un sistema de comunicaciones se tiene una secuencia aleatoria, con código de línea NRZ y duración t_b . La Densidad Espectral de Potencia sería:

$$G(f) = \frac{|P(f)|^2}{t_b}; \quad (9.5)$$

$P(f) \rightarrow$ Transformada de Fourier de la señal de entrada.

Asumiendo un sistema como sigue:

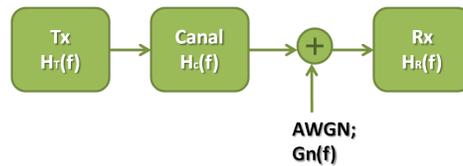


Figure 9.13

La salida del sistema sería una sucesión de pulsos $y(t)$, asociada a un pulso de salida $p_R(t)$ y a los de entrada:

$$A_k | P_R(f) | = | P(f) | \cdot | H_T(f) | \cdot | H_c(f) | \cdot | H_R(f) | \rightarrow \text{Ecuación (1)}$$

Si la potencia de transmisiones :

$$S_T = \int_{-\infty}^{\infty} \frac{|P(f)|^2 |H_T(f)|^2}{t_b} df$$

Pudiéramos expresar la función de la ecuación (1) :

$$t_b \cdot S_T = A_k^2 \int_{-\infty}^{\infty} \frac{|P_R(f)|^2}{|H_c(f)| \cdot |H_R(f)|} df$$

Ahora bien, como nuestro objetivo es maximizar la relación señal a ruido, despejamos el valor de A_k

(Amplitud del pulso) y definimos σ^2 (que debe ser minimizado):

$$A_{k^2} = \frac{tb.S_T}{\int_{-\infty}^{\infty} \frac{|P_R(f)|^2}{|H_C(f)| \cdot |H_R(f)|} df}$$

$$\sigma^2 = \int_{-\infty}^{\infty} G_n(f) |H_R(f)|^2 df$$

Por lo que :

(9.6)

$$\frac{A_{k^2}}{\sigma^2} = \frac{tb.S_T}{\int_{-\infty}^{\infty} G_n(f) |H_R(f)|^2 df \cdot \int_{-\infty}^{\infty} \frac{|P_R(f)|^2}{|H_C(f)| \cdot |H_R(f)|} df} \rightarrow \text{Minimizar}$$

A través de la igualdad de Schwartz podemos cumplir el objetivo:

$$\left| \int_{-\infty}^{\infty} V(f) \cdot W(f) df \right| \leq \int_{-\infty}^{\infty} |V(f)|^2 df \cdot \int_{-\infty}^{\infty} |W(f)|^2 df$$

Que será igual cuando $V(f) = k \cdot W(f)$. Si :

$$W(f) = \frac{|P_R(f)|}{|H_C(f)| \cdot |H_R(f)|} y V(f) = |H_R(f)| \sqrt{G_n(f)}$$

Entonces :

$$|H_R(f)| \sqrt{G_n(f)} = k \left[\frac{|P_R(f)|}{|H_C(f)| \cdot |H_R(f)|} \right]$$

$$|H_R(f)|^2 = k \left[\frac{|P_R(f)|}{|H_C(f)| \cdot \sqrt{G_n(f)}} \right]$$

Finalmente, con la ecuación (1) tenemos que:

$$|H_T(f)|^2 = \frac{A_{k^2} |P_R(f)| \sqrt{G_n(f)}}{k |H_C(f)| |P_R(f)|^2} \quad (9.7)$$

9.1.6 Simulaciones en LabVIEW

El VI correspondiente a la teoría de este módulo puede descargarse a través del siguiente enlace: [MEDIA OBJECT]²

²This media object is a downloadable file. Please view or download it at <ISI.rar>

Chapter 10

10. Codificación de Canal: Código Hamming y Código Convolutacional¹

10.1 CODIFICACIÓN DE CANAL: CÓDIGO HAMMING Y CÓDIGO CONVOLUCIONAL

González C. Y. Venuska

Mezoa R. Mariangela

Resumen

Este módulo contiene información acerca del bloque de codificación de canal de un sistema de comunicaciones digitales, específicamente la codificación por bloque (Código Hamming) y la codificación convolutacional para la detección y corrección de errores.

Cuando transmitimos información, uno de los objetivos principales es el de minimizar la cantidad de errores que pudieran producirse en el proceso. Esta transmisión depende del factor Señal a Ruido (S/N), potencia y velocidad de transmisión. Si optimizando estas variables se necesita aún mejorar la calidad de la transmisión, entonces se deben buscar ciertos métodos que aseguren y mejoren la fiabilidad. Es a partir de aquí que surge el concepto de la codificación para control de errores.

La **codificación de canal** para el control de errores se encarga, básicamente, de la adición de dígitos extra al mensaje a transmitir. Ellos no poseen información como tal, pero hacen posible la detección y corrección de errores en el bloque de recepción del mensaje.

10.1.1 Detección y Corrección de Errores:

En un sistema de comunicaciones, detectar un error es más sencillo que corregirlo. Si existen irregularidades, el receptor puede pedir una retransmisión del mensaje que contiene el error (**ARQ**: *Automatic Repeat Request*). Sin embargo, cuando el sistema no implementa esta técnica por no ser práctico o simplemente porque no es posible, debe aplicar redundancia en el código a través del método **FEC** (*Forward Error Correction*).

El método más sencillo de redundancia en el código consiste en repetir **n** veces el símbolo de mensaje. Cuando los símbolos son 1 y 0, cualquier error de transmisión en una palabra código recibida alterará el patrón de repetición cambiando un 1 a 0 (o viceversa). Si los errores de transmisión ocurren de forma aleatoria (e independiente) con probabilidad **P_e**, entonces se pudiera definir la probabilidad de que ocurran **i** errores en una palabra código de **n** bits como:

¹This content is available online at <<http://cnx.org/content/m36906/1.1/>>.

$$P(i, n) = \binom{n}{i} Pe^i (1 - Pe)^{n-i} \approx \binom{n}{i} Pe^i ; \text{ Siempre que:}$$

$$Pe \ll 1 \quad ; \quad \binom{n}{i} = \frac{n!}{i!(n-i)!}$$

Pe : probabilidad de error sin codificar

Figure 10.1

Por ejemplo, si se considera un código de repetición triple (3, 1) (1 bit de mensaje, dos bits de repetición, palabra código de tres bits), las palabras código serían 000 y 111. Cualquier mensaje recibido que no coincida con estas palabras código evidencian la presencia de errores. Pero, si los tres bits están errados (Se transmite 000 pero se recibe 111) entonces será imposible detectar el error:

$$P_{we} = P(3,3) = Pe^3$$

Figure 10.2

Si se quiere corregir el error, asumimos que al menos dos de los tres bits son correctos. Entonces, 001 se decodifica como 000 y 101 se decodifica como 111. Esto corrige palabras con un solo error, pero para dos o tres errores la probabilidad de que una palabra esté errada resulta como:

$$P_{we} = P(2,3) + P(3,3) = 3Pe^2 - 2Pe^3$$

Figure 10.3

10.1.2 Distancia Hamming

Una palabra código de n bits puede ser representarse como un vector en un espacio de n dimensiones. Por ejemplo, el código 010 puede representarse como $X=(0\ 1\ 0)$. Tomemos el caso anterior de código de repetición (3, 1):

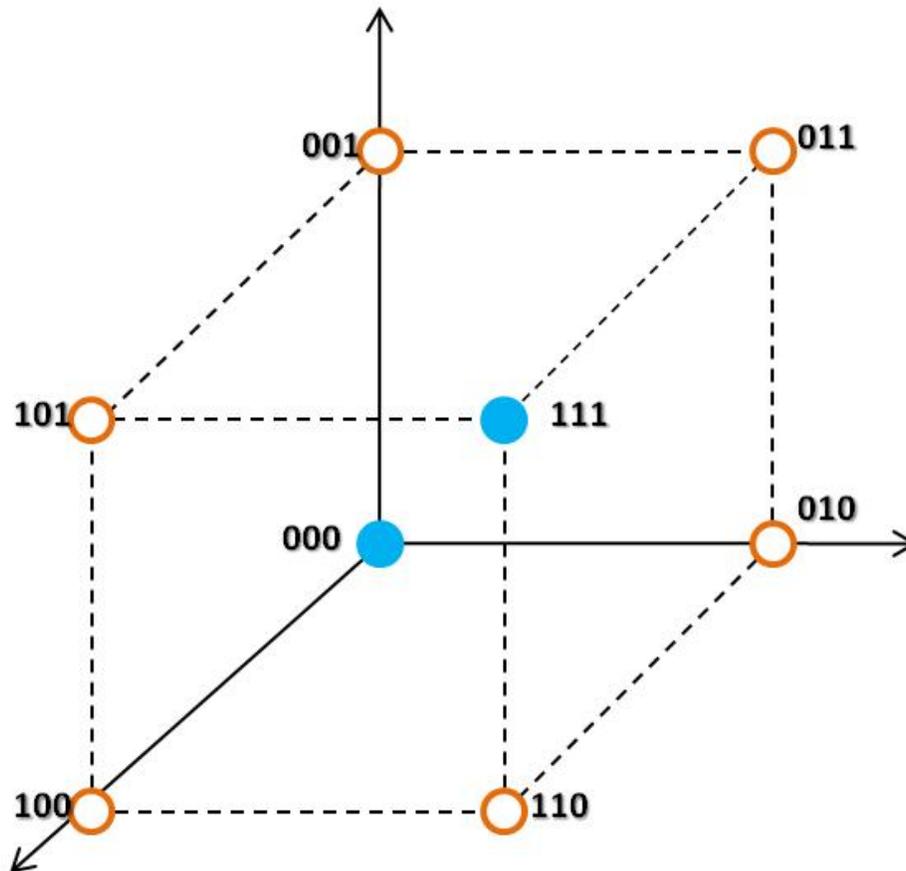


Figure 10.4: Vectores que representan palabras código de tres bits.

En total se grafican las 8 posibles combinaciones de palabras código y los puntos azules representan el código de repetición. La distancia que separa dos puntos cualesquiera se reconoce como la distancia Hamming, que está relacionada con el poder de control de errores del código (Fortaleza del código) y se define como el número de elementos diferentes entre dos puntos. Por ejemplo:

Si consideramos los vectores:

$$X = (0 \ 1 \ 1)$$

$$Y = (1 \ 1 \ 0)$$

Entonces:

Distancia Hamming = $d(X, Y) = 2$, Porque el primer y tercer elemento son distintos.

Figure 10.5

La distancia mínima de un código en particular es la distancia Hamming **más pequeña** entre dos vectores de código **válidos**. De esta forma, la detección de errores es posible siempre que se cumpla que el número de errores de transmisión en una palabra código sea menor a d_{\min} , por lo que la palabra errada no es un vector válido. Si es mayor o igual, la palabra errada puede corresponder a un vector válido y el error no podría ser detectado.

La capacidad de corrección y detección de un código se define como:

$$\text{Corrección: } t \leq \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$

$$\text{Detección: } \ell \leq d_{\min} - 1$$

Figure 10.6

Para el ejemplo desarrollado, $d_{\min} = 3$, por lo que el código de repetición es capaz de corregir hasta 1 error por palabra y detectar hasta 2 errores. Evidentemente, la fortaleza del código depende del número de bits que se le agregan al mensaje original. La distancia mínima de una codificación por bloque (repetición) se define como:

$$d_{min} \leq n - k + 1$$

Donde:

$n =$ Longitud de la palabra código.

$k =$ Longitud del Mensaje original.

Figure 10.7

10.1.3 Codificación FEC

Un sistema FEC puede expresarse gráficamente como:

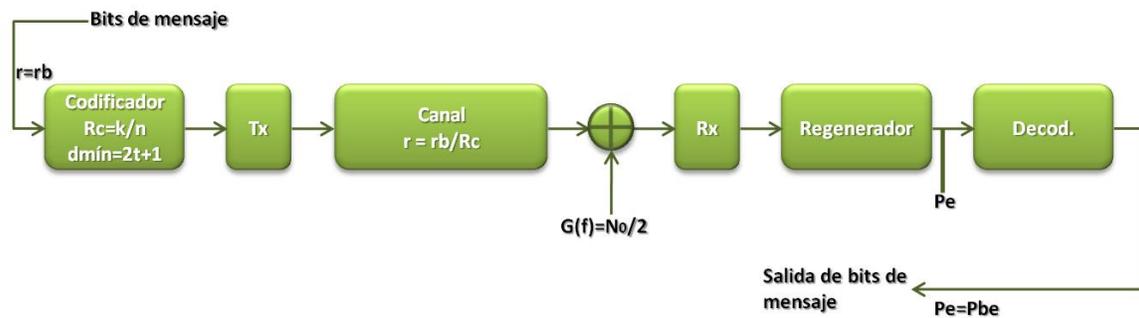


Figure 10.8

Los bits de entrada llegan con una tasa de r_b . El codificador toma bloques de k bits del mensaje y construye un código de bloques (n, k) con tasa $R_c = k/n < 1$. La tasa de bits del canal será $r = r_b / R_c > r_b$. La probabilidad de error tomará un valor de $P_e \ll 1$, que evidentemente dependerá de la energía de la señal y la densidad de potencia del ruido en el receptor ($[U+F068]_0$). Si E_b representa la energía promedio por bit de mensaje, entonces la energía promedio por bit de código es $R_c \cdot E_b$. Entonces:

La probabilidad de error por PALABRA es:

$$P_{we} = \sum_{i=t+1}^n P(i, n) \approx \binom{n}{t+1} P_e^{t+1}$$

La probabilidad de error por bit de mensaje de salida es:

$$P_{be} = \frac{t+1}{n} P_{we} \approx \binom{n-1}{t} P_e^{t+1}$$

La probabilidad de error SIN codificar es:

$$P = Q\left(\sqrt{\frac{2Eb}{\eta}}\right)$$

Figure 10.9

10.2 Codificación por Bloques Lineal

Como ya se mencionó antes, un **código por bloques**(n, k) consiste en vectores de n bits, con cada vector correspondiente a un bloque único de k bits de mensaje. Dado que existen 2^k bloques de mensaje y 2^n vectores de n bits, la meta está en seleccionar los 2^k vectores de código que cumplan con la condición de que la **distancia mínima Hamming** sea lo más grande posible.

Supongamos un vector de código arbitrario tal que:

$$X = (x_1, x_2, x_3, \dots, x_n)$$

Donde los elementos son números binarios.

Figure 10.10

Se cumple que esta codificación por bloques es lineal si al sumar, en módulo 2, dos palabras código se genera una tercera palabra código válida, incluyendo el código de puros ceros. Esta suma se define como:

$$X + Z = (x_1 \oplus z_1 \quad x_2 \oplus z_2 \quad x_3 \oplus z_3 \quad \dots \quad x_n \oplus z_n)$$

recordamos :

$$a_1 \oplus a_2 \begin{cases} 0 & \text{si } a_1 = a_2 \\ 1 & \text{si } a_1 \neq a_2 \end{cases}$$

Figure 10.11

Ahora bien, una codificación por bloques sistemática consiste en vectores cuyos primeros (o últimos) k elementos son idénticos a los bits de mensaje, tomando el resto de los elementos ($n-k$) como bits de chequeo. Por lo tanto, un vector de código se expresa como:

$$X = (m_1 \quad m_2 \quad m_3 \quad | \quad c_1 \quad c_2 \quad c_3)$$

$$X = (M \mid C)$$

M : Vector de mensaje de k bits.

C : Vector de chequeo de q bits. $q = n - k$

Figure 10.12

El vector de código X puede obtenerse a través de una multiplicación de matrices:

$$X = MG$$

G : matriz generadora con la siguiente estructura:

$$G = [I_k \mid P]$$

Donde:

I_k = Matriz identidad de $k \times k$

$$P = \text{Matriz de paridad} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1q} \\ p_{21} & p_{22} & \cdots & p_{2q} \\ \vdots & \vdots & & \vdots \\ p_{k1} & p_{k2} & & p_{kq} \end{bmatrix}$$

Figure 10.13

La matriz identidad en G reproduce el vector de mensaje para los primeros k elementos de X , mientras que la sub-matriz P genera el vector C a través de $C=MP$. Al igual que con la suma de vectores explicada inicialmente, esta multiplicación de matrices sigue el procedimiento de la suma módulo 2.

Para la recepción, tomaremos el vector Y . Cualquier error de transmisión resultará en $Y \neq X$. Uno de los métodos más fáciles es haciendo uso de una **matriz de chequeo de paridad H** derivada de la sub-matriz P . Se define como:

$$H^T = \begin{bmatrix} P \\ I_q \end{bmatrix}$$

Donde:

P : Matriz de Paridad

I_q = Matriz identidad de $q \times q$

$$y$$

$$X H^T = 0$$

Figure 10.14

Si esta última condición no se cumple, el código recibido contiene errores. La detección de errores puede basarse a partir del vector síndrome S:

$$S = Y H^T$$

Figure 10.15

Si todos los elementos de S son iguales a CERO, entonces hay dos posibilidades:

- Y es igual a X o Y;
- Y es igual a otro vector código y el error de transmisión no se detecta.

De igual forma, los errores se indican con la presencia de elementos distintos a cero en S.

Ahora bien, si el vector X se corrompe con un vector de error E, el vector Y resultará en:

$$Y = X + E$$

Figure 10.16

Por lo que S será:

$$S = (X + E)H^T = XH^T + EH^T = EH^T$$

Figure 10.17

Esto evidencia que el Síndrome dependerá completamente del patrón de error, no del vector transmitido. Al existir errores en la codificación, S indicará en dónde se encuentra el mismo.

Si un código debe corregir hasta t errores por palabra, entonces q y n deben cumplir con la desigualdad o relación de Hamming, definida como:

Desigualdad de HAMMING :

$$2^q \geq \sum_0^t \binom{n}{i}$$

Figure 10.18

10.2.1 Código Hamming

Es un código de bloques lineal (n,k) con 3 ó más bits de chequeo/redundancia (q) que cumple con estas ecuaciones:

$$n = 2^q - 1$$

$$k = 2^q - q - 1 = n - q$$

Figure 10.19

Dado que su distancia mínima siempre es 3 (sin importar el valor de q), este tipo de código es capaz de corregir hasta un error y detectar dos, por lo que:

$$d_{\min} = 3$$

$$R_c = \frac{k}{n} = 1 - \frac{q}{2^q - 1}$$

Figure 10.20

Supongamos el caso Hamming (7,4) ($n=7$, $q=3$, $k=4$). Para construir dicho código, tomamos la submatriz P de la matriz generadora G y la llenamos con todas las palabras de q bits que tengan dos o más 1's, sin ningún orden en particular. Como son $k=4$ bits de mensaje original, entonces la matriz G tendrá cuatro filas. Entonces:

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array} \right]$$

Figure 10.21

Dado un bloque de bits de mensaje $M = (m_1 m_2 m_3 m_4)$, los bits de chequeo se calculan sustituyendo los elementos de la sub-matriz P en la ecuación $C=MP$, por lo que:

$$\begin{aligned} c_1 &= m_1 \oplus 0 \oplus m_3 \oplus m_4 \\ c_2 &= m_1 \oplus m_2 \oplus m_3 \oplus 0 \\ c_3 &= 0 \oplus m_2 \oplus m_3 \oplus m_4 \end{aligned}$$

Figure 10.22

Para este ejemplo se pudiera construir un codificador Hamming de la siguiente forma:

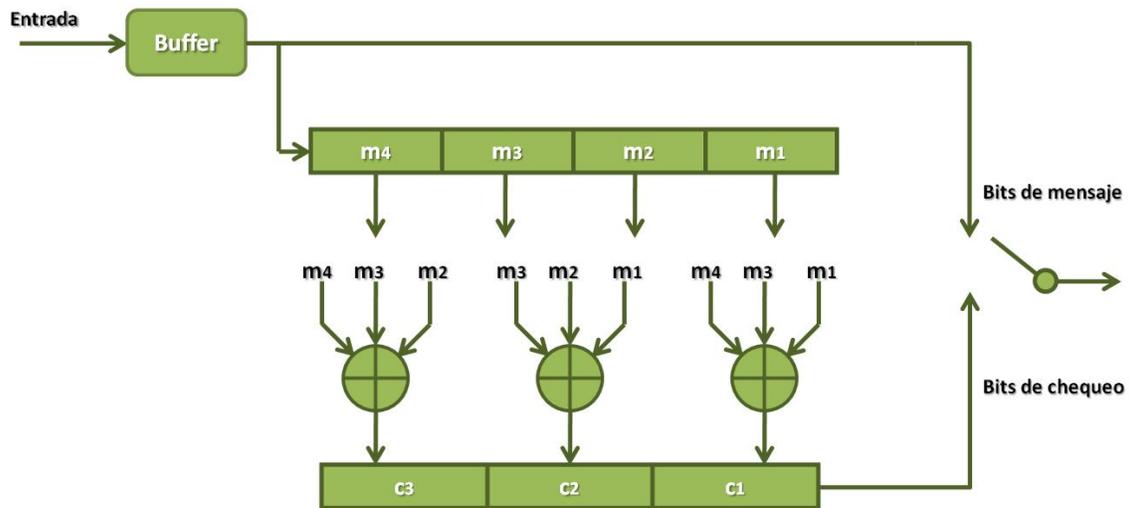


Figure 10.23

Las palabras código correspondientes a este código Hamming (7,4) se observan en la tabla (Todas generadas a partir de $C=MP$):

Mensaje				Bits Redundantes		
0	0	0	0	0	0	0
0	0	0	1	1	0	1
0	0	1	0	1	1	1
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	1	1	0
0	1	1	0	1	0	0
0	1	1	1	0	0	1
1	0	0	0	1	1	0
1	0	0	1	0	1	1
1	0	1	0	0	0	1
1	0	1	1	1	0	0
1	1	0	0	1	0	1
1	1	0	1	0	0	0
1	1	1	0	0	1	0
1	1	1	1	1	1	1

Table 10.1

10.2.2 Códigos Convolucionales

A diferencia de la codificación por bloque, los códigos convolucionales trabajan bit por bit. Su estructura pudiera generalizarse de la siguiente forma:

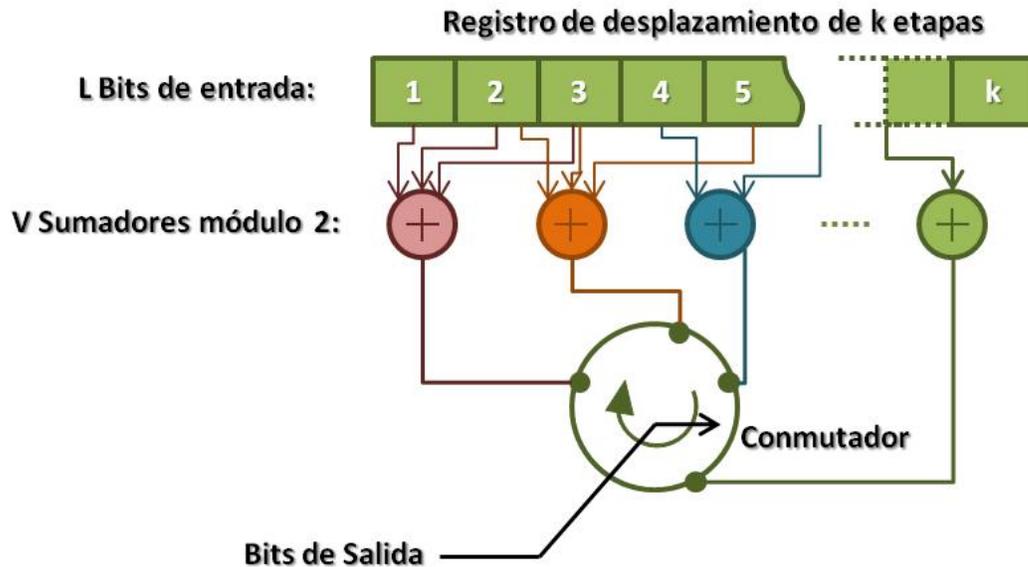


Figure 10.24

Esta estructura genera V símbolos a la salida por cada símbolo de entrada. Su tasa de símbolos de salida será entonces $R_c = 1/V$. Existen 4 métodos de representación de los códigos convolucionales:

- Por Conexión.
- Diagrama de estados.
- Árbol de código.
- Diagrama Trellis

10.2.3 Representación por Conexión:

Supongamos que se tiene el siguiente codificador convolucional:

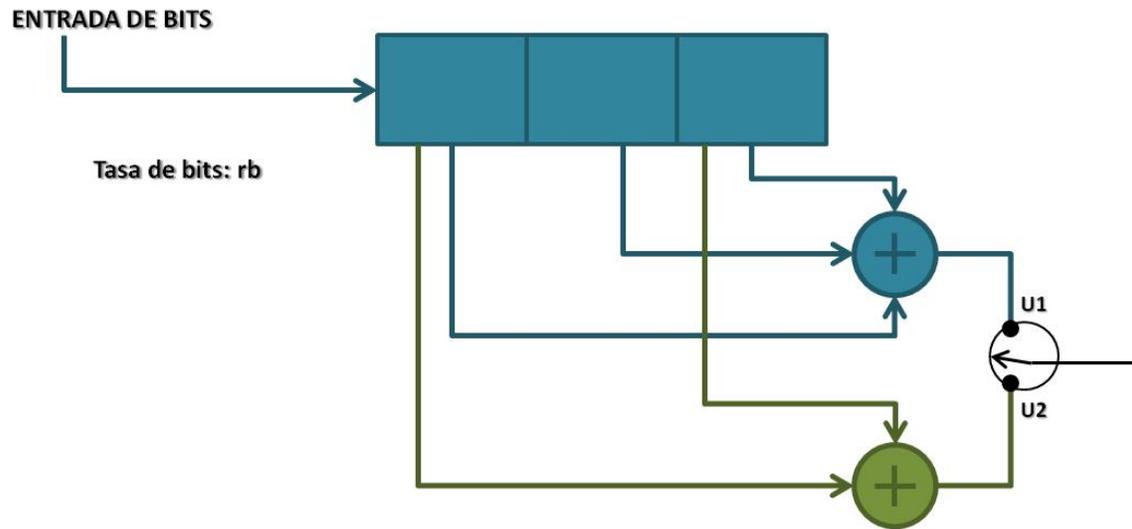


Figure 10.25

Para que el código funcione correctamente, antes de que llegue el primer bit del mensaje el registro de desplazamiento está 'limpio' (Sólo contiene 0's). Esto implica que la primera salida será $U1=0$ y $U2=0$. Supongamos que el mensaje de entrada es **101**. Entonces, el codificador hará los siguientes pasos:

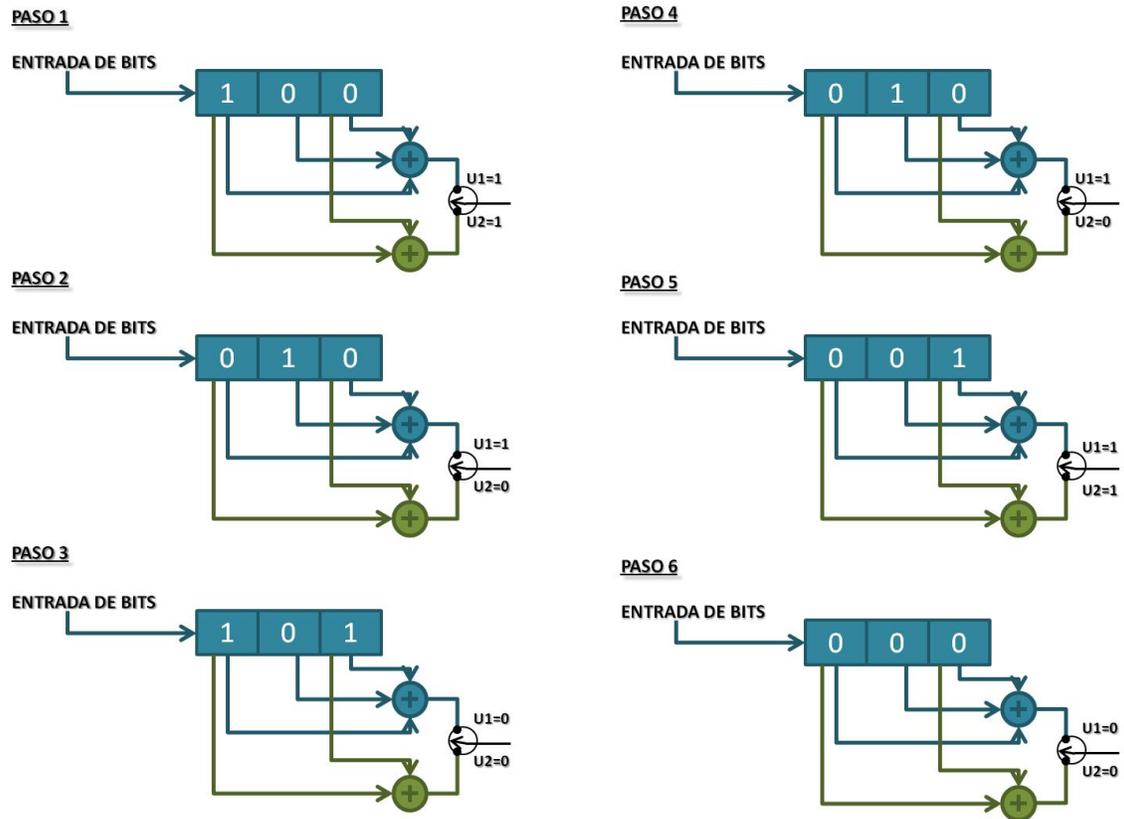


Figure 10.26: Diagrama de conexión con salidas U1 y U2 cuando se tiene como mensaje de entrada la secuencia 101. Se asume que inicialmente los registros están vacíos.

La secuencia de salida será:

U1	U2
1	1
1	0
0	0
1	0
1	1

Table 10.2

10.2.4 Representación por Diagrama de Estados:

El estado del codificador convolucional se define como los contenidos de los registros de desplazamiento de las $k-1$ etapas más a la derecha del codificador. Para el ejemplo actual:

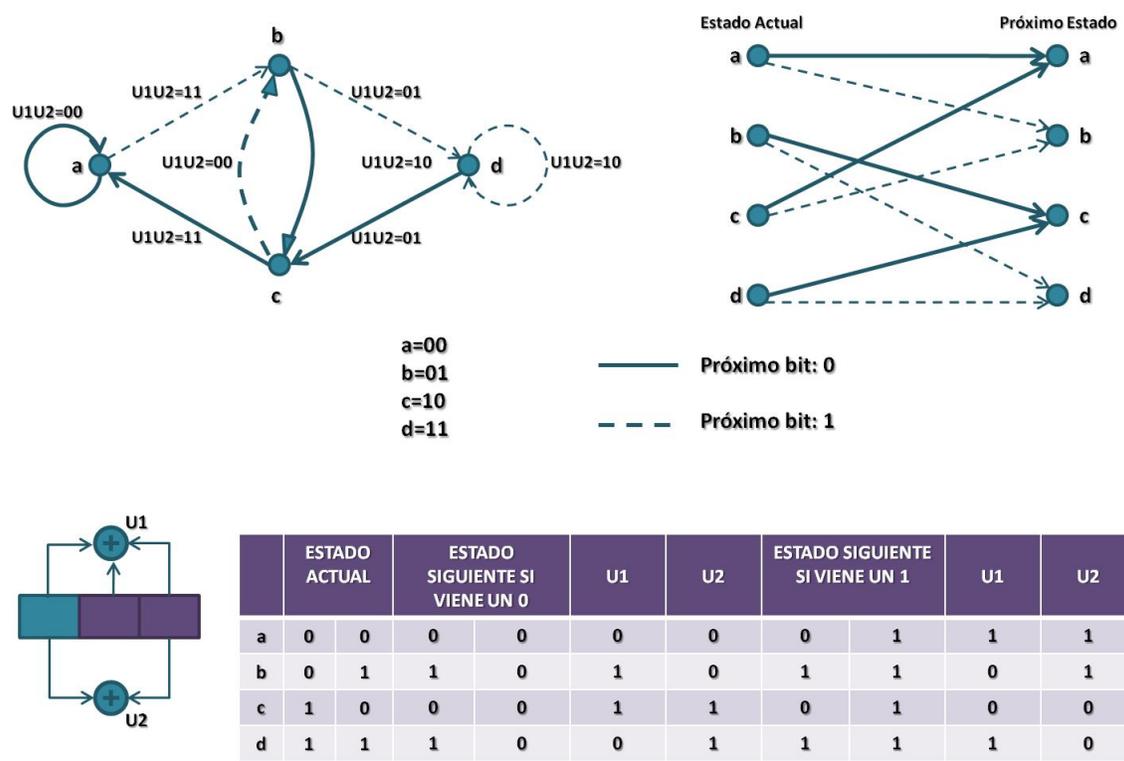


Figure 10.27: Diagrama de estados.

10.2.5 Representación por Árbol de código:

La raíz representa el estado inicial (U1U2=00). Si el próximo bit es un 0, se toma la ramificación superior. Dado que la longitud del registro es igual a 3 (L=3), entonces el árbol será:

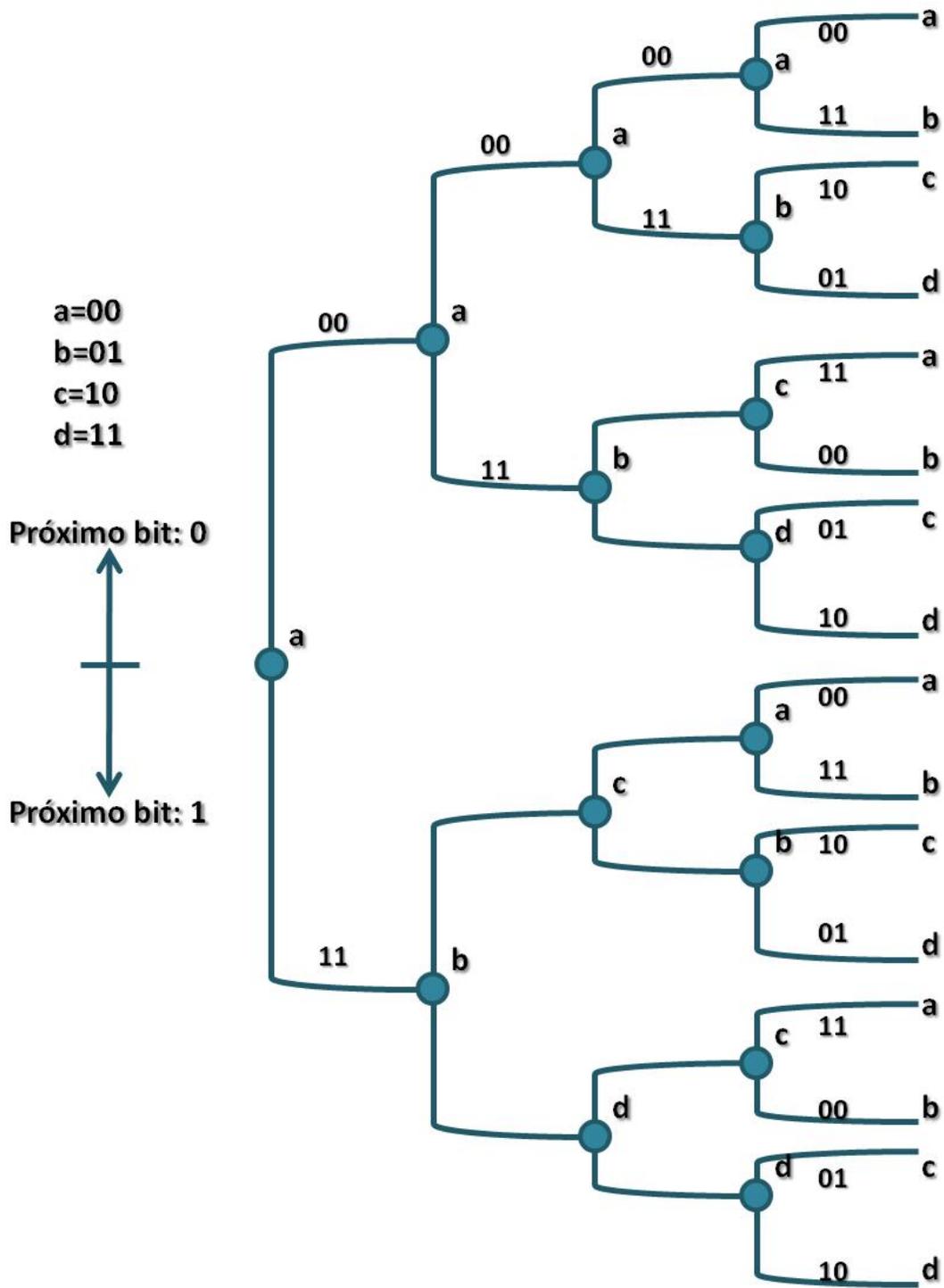


Figure 10.28

10.2.6 Diagrama Trellis

A través de él se puede observar la evolución del codificador por cada T que pase. El diagrama deberá tener 2^{k-1} nodos que representen los posibles estados (a,b,c y d). Entonces:

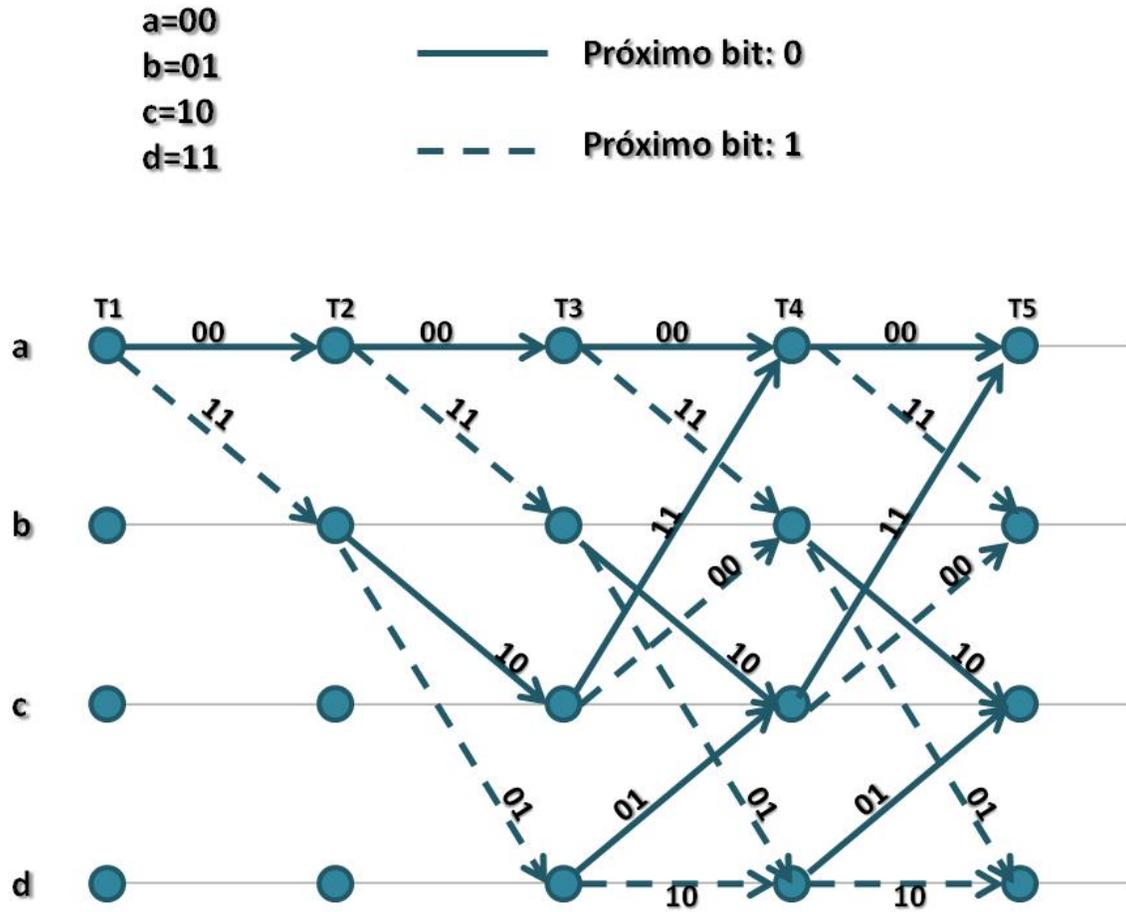


Figure 10.29

Para la decodificación implementamos la **máxima verosimilitud** (*maximum-likelihood*), que busca maximizar $P(Z/U_m)$, siendo U_m las posibles secuencias recibidas.

L bits de mensaje = 2^L secuencias

$$P\left(\frac{Z}{U^m}\right) = \prod_i P(Z_i / U_i^m) = \prod_{i=1}^{\infty} \prod_{j=1}^n P(Z_{ji} / U_{ji}^m)$$

Donde:

Z_i = i -ésima rama de la secuencia Z .

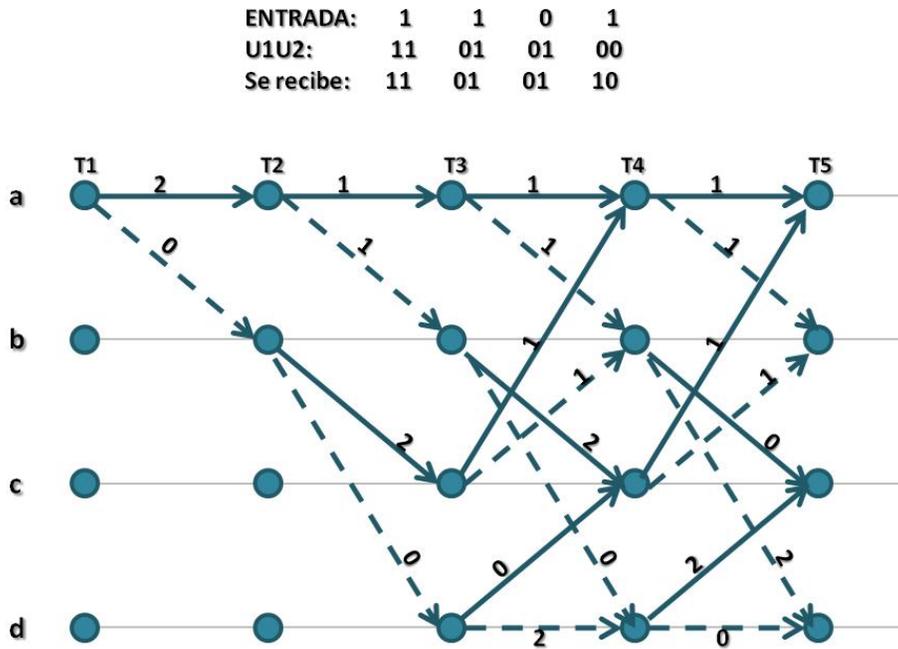
U_i^m = i -ésima rama de una secuencia U^m determinada.

Z_{ji} = j -ésimo símbolo de Z_i .

U_{ji}^m = j -ésimo símbolo de U_i^m .

Figure 10.30

Esto puede verse a través del **Algoritmo de Viterbi**: Se basa en el Diagrama de Trellis. Asigna a cada rama una **métrica** que es igual a la distancia Hamming correspondiente a la rama recibida y todas las ramas de Trellis que intervienen en ese instante de tiempo. Supongamos el siguiente ejemplo:



Se coloca la métrica entre lo recibido y la salida U1U2 del diagrama Trellis. Al encontrarnos con dos trayectorias que convergen, se considera la de métrica menor.

Figure 10.31

Como se observa, se debe esperar hasta el instante T4 (que es cuando empiezan a cerrarse los lazos) para tomar una decisión sobre el camino más corto. Allí, se tendrán dos caminos para llegar a ese instante:

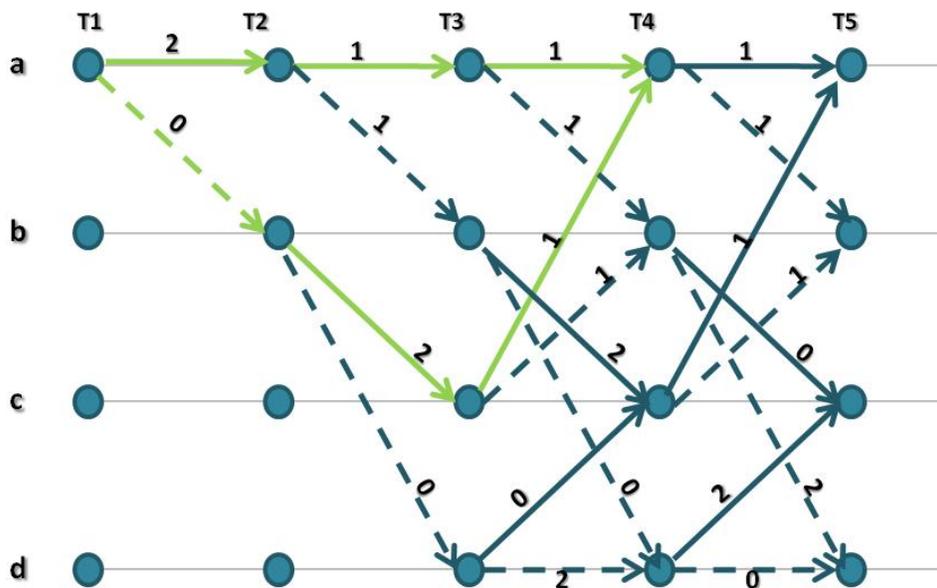


Figure 10.32

Partiendo del estado a, en el estado 4 convergen dos trayectorias dos métricas: la primera con un total de 4 (la de arriba) y la segunda con un total de 3 (la que va por debajo). Luego se analizan situaciones similares en T4. Por ejemplo, al estado b en T4 se llega por dos trayectorias: la de arriba con métrica 4 y la de abajo con métrica 3. Para T4 pero al estado c también se llega por dos caminos: el de arriba con métrica 5 y el de abajo con métrica 0. Para T4 pero al estado d los dos caminos tienen métricas 3 y 2 respectivamente. Al eliminar todas las trayectorias de métrica mayor, en este ejemplo se logra ya despejar el bit transmitido entre la transición T1-T2 y decidir que es un “1”.

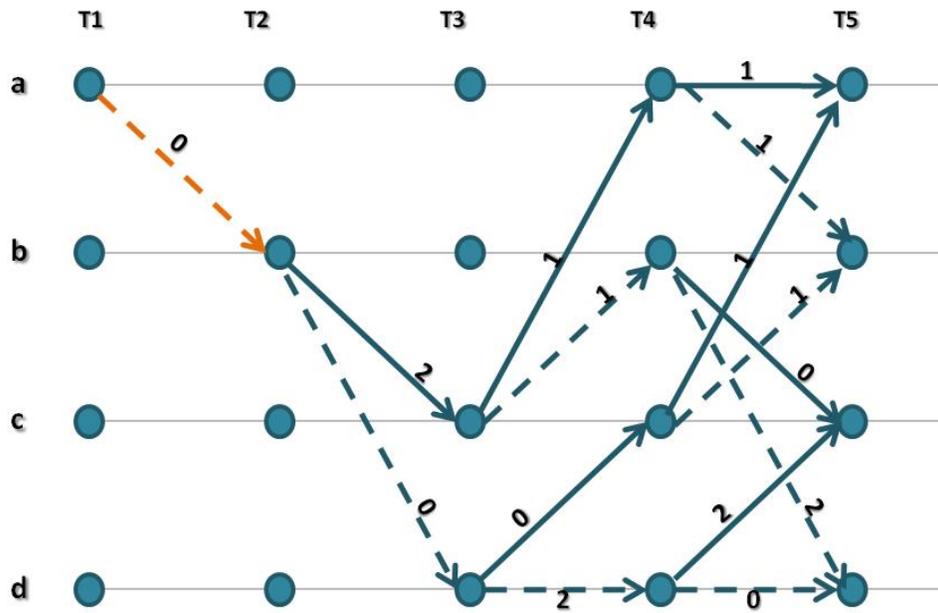


Figure 10.33

Se repite el mismo procedimiento para el nodo T5 y luego para los demás nodos hasta que se vayan obteniendo cada uno de los bits transmitidos.

Se descartan entonces todos los caminos no viables. Este proceso debe repetirse hasta que se logre obtener la secuencia original.

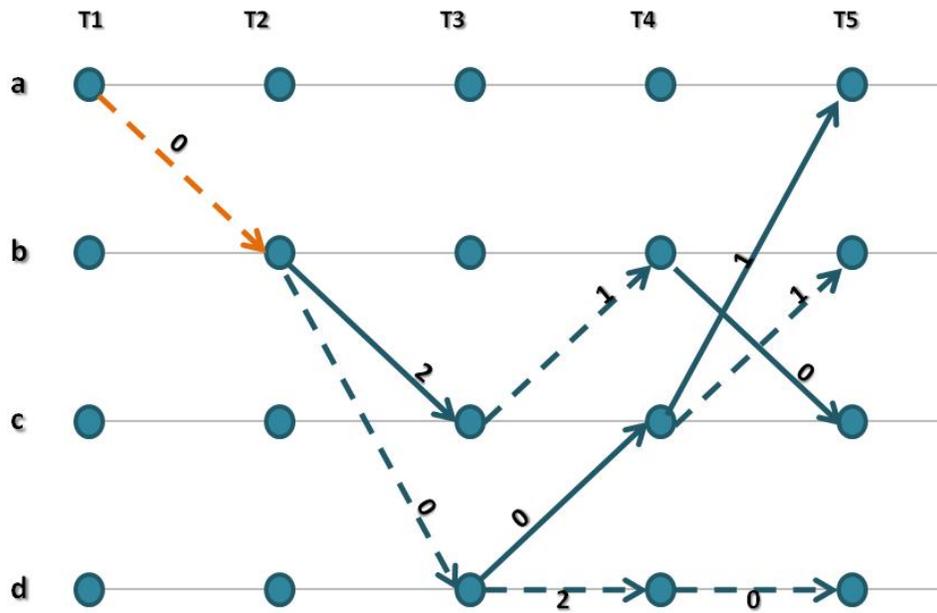


Figure 10.34

10.2.7 Simulación en LabVIEW

Las pruebas para las codificaciones Hamming(7,4), Hamming(15,11) y Convolutcional pueden realizarse descargando el VI del siguiente enlace: [MEDIA OBJECT]²

²This media object is a downloadable file. Please view or download it at <canal.rar>

Glossary

A Ancho de Banda (BW):

En el ámbito de las comunicaciones, el ancho de banda representa el rango (en Hertz, Hz) de frecuencias en donde se concentra la mayor parte de la potencia de la señal que se está usando. A partir de este punto se podrá calcular el ancho de banda en las simulaciones usando la transformada de Fourier, cuando se expresa la señal original en el dominio de la frecuencia. Matemáticamente se pudiera definir así:

AWGN

Corresponde a un modelo en el que al canal de comunicaciones se le suma ruido blanco en banda base, con una densidad espectral constante ($[U+F068]/2$) y una amplitud de distribución Gaussiana.

I Intervalo de impulso (τ):

Es la duración mínima del impulso usado para representar la información.

T Tasa de Bits (R):

Número de bits transmitidos por segundo. Su unidad son los bps (bits per second) y se representa matemáticamente como:

Tiempo de bit (Tb):

Está definido como el tiempo empleado para la representación y transmisión de un bit en el sistema.

V Velocidad de Transmisión (Vt):

Se define como el número máximo de cambios por segundo que experimenta la señal o el número de impulsos que pueden transmitirse en un segundo. Matemáticamente viene expresado como

Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

- A** Analógicas, 1
Ancho de Banda (BW):, 19
AWGN, 55
- B** binario, 41
bloques, 5
- C** código por bloques, 90
- D** de todos los pulsos, 74
Detección Coherente, LabVIEW, mathscript,
Probabilidad de error, ruido AWGN, § 7(55)
Detección no coherente, ruido AWGN,
LabVIEW, probabilidad de error, § 8(65)
Digitales, 1
- G** Gram-Schmidt (G-S)., 13
gráficos, 5
- I** Interferencia Intersimbólica, 73
Intervalo de impulso (τ):, 20
- L** LabVIEW, Comunicación Digital, § 1(1)
LabVIEW, Coseno alzado, raíz de coseno
alzado, filtros adaptados, probabilidad de
error, mathscript, § 9(73)
LabVIEW, Hamming, Convolutacional,
Probabilidad de Error, § 10(85)
LabVIEW, modulación, Modulation Toolkit,
§ 5(31)
LabVIEW, NRZ, RZ, Manchester,
constelación, § 4(19)
LabVIEW, ortonormalidad, ortogonalización,
constelación, § 3(13)
LabVIEW, PCM, Analógico y Digital, § 2(5)
LabVIEW, QAM, QPSK, modulación,
modulation toolkit., § 6(41)
- O** Ortogonalización, 13
ortonormales, 13
- R** Resumen, 1, 5, 13, 19, 31, 41, 55, 65, 73, 85
- T** Tasa de Bits (R):, 20
Tiempo de bit (Tb):, 20
- V** Velocidad de Transmisión (Vt):, 20

Attributions

Collection: *Laboratorio Digital Interactivo*

Edited by: Mariangela Mezoa

URL: <http://cnx.org/content/col11274/1.1/>

License: <http://creativecommons.org/licenses/by/3.0/>

Module: "Introducción a los Sistemas de Comunicaciones Digitales"

Used here as: "1. Introducción a los Sistemas de Comunicaciones Digitales"

By: Mariangela Mezoa

URL: <http://cnx.org/content/m35707/1.1/>

Pages: 1-3

Copyright: Mariangela Mezoa

License: <http://creativecommons.org/licenses/by/3.0/>

Module: "2. Introducción a LabVIEW y teoría básica de PCM (Pulse Code Modulation)"

By: Mariangela Mezoa

URL: <http://cnx.org/content/m35710/1.3/>

Pages: 5-11

Copyright: Mariangela Mezoa

License: <http://creativecommons.org/licenses/by/3.0/>

Module: "3. Ortogonalización Gram-Schmidt y teoría básica de las Constelaciones"

By: Mariangela Mezoa

URL: <http://cnx.org/content/m35711/1.1/>

Pages: 13-18

Copyright: Mariangela Mezoa

License: <http://creativecommons.org/licenses/by/3.0/>

Module: "4. Códigos de Línea"

By: Mariangela Mezoa

URL: <http://cnx.org/content/m35716/1.4/>

Pages: 19-29

Copyright: Mariangela Mezoa

License: <http://creativecommons.org/licenses/by/3.0/>

Module: "5. Modulaciones Binarias: Teoría y simulación en LabVIEW"

By: Mariangela Mezoa

URL: <http://cnx.org/content/m35717/1.1/>

Pages: 31-39

Copyright: Mariangela Mezoa

License: <http://creativecommons.org/licenses/by/3.0/>

Module: "6. Modulaciones M-arias"

By: Mariangela Mezoa

URL: <http://cnx.org/content/m35715/1.1/>

Pages: 41-53

Copyright: Mariangela Mezoa

License: <http://creativecommons.org/licenses/by/3.0/>

Module: "7. Introducción del ruido en los sistemas de Comunicaciones Digitales: Detección Óptima y Probabilidad de Error"

By: Mariangela Mezoa

URL: <http://cnx.org/content/m36883/1.4/>

Pages: 55-63

Copyright: Mariangela Mezoa

License: <http://creativecommons.org/licenses/by/3.0/>

Module: "8. Detección no Coherente para modulaciones OOK y FSK"

By: Mariangela Mezoa

URL: <http://cnx.org/content/m36884/1.1/>

Pages: 65-72

Copyright: Mariangela Mezoa

License: <http://creativecommons.org/licenses/by/3.0/>

Module: "9. Interferencia Intersimbólica (ISI)"

By: Mariangela Mezoa

URL: <http://cnx.org/content/m36885/1.1/>

Pages: 73-83

Copyright: Mariangela Mezoa

License: <http://creativecommons.org/licenses/by/3.0/>

Module: "10. Codificación de Canal: Código Hamming y Código Convolutivo"

By: Mariangela Mezoa

URL: <http://cnx.org/content/m36906/1.1/>

Pages: 85-108

Copyright: Mariangela Mezoa

License: <http://creativecommons.org/licenses/by/3.0/>

Laboratorio Digital Interactivo

En esta colección se explicarán los tópicos más importantes de un sistema de Comunicaciones Digitales. Para realizar pruebas en cada VI publicado es necesario tener instalado en el ordenador: - LabVIEW versión 8.5 o mayor. - Matlab compatible para los módulos de mathscript incluidos en ciertos VIs.

About Connexions

Since 1999, Connexions has been pioneering a global system where anyone can create course materials and make them fully accessible and easily reusable free of charge. We are a Web-based authoring, teaching and learning environment open to anyone interested in education, including students, teachers, professors and lifelong learners. We connect ideas and facilitate educational communities.

Connexions's modular, interactive courses are in use worldwide by universities, community colleges, K-12 schools, distance learners, and lifelong learners. Connexions materials are in many languages, including English, Spanish, Chinese, Japanese, Italian, Vietnamese, French, Portuguese, and Thai. Connexions is part of an exciting new information distribution system that allows for **Print on Demand Books**. Connexions has partnered with innovative on-demand publisher QOOP to accelerate the delivery of printed course materials and textbooks into classrooms worldwide at lower prices than traditional academic publishers.