Bibliografía

Norman Abramson, "Teoría de la Información y Codificación", Paraninfo, 1986.

Enrique Mandado, "Sistemas Electrónicos Digitales", Marcombo, 1987.

J.-P. Meinadier, "Estructura y Funcionamiento de las Computadores Digitales", AC, 1986.

Roc&C'2010_ Comité de Comunicaciones de IEEE México , 2010

TEORIA DE LA INFORMACIÓN y DE LA CODIFICACION

1.- Generalidades

Hemos visto distintos temas relacionados con los sistemas de comunicaciones y se los vio en términos de señales, tanto deseadas como indeseadas, diferentes modelos de señales, su efecto en redes y diversos sistemas de modulación como medio de transmisión.

Asimismo, se llegó a la conclusión que los sistemas de comunicaciones están limitados por su potencia disponible, la relación señal a ruido, la necesidad de acotar su ancho de banda.

Como sabemos, el comportamiento de los diversos sistemas no es igual, algunos son para ciertas aplicaciones mejores que otros y es necesario, pensar en el desarrollo de nuevos sistemas, teniendo una visión más general y conceptual del proceso de las comunicaciones, que permita una perspectiva mejor. Para ello se deben determinar cuales son las características de dichos sistemas, ya sea tecnológicamente, como también su comportamiento en el mundo real. En esencia podemos decir que a eso se refiere la teoría de la información y los sistemas de comunicaciones.

Hemos definido a la “Comunicación” como el proceso en el cual la información es transferida desde un punto (fuente de información), a través de un medio de transmisión o canal, aun destinatario o usuario.

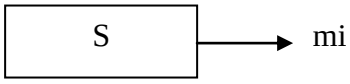
La teoría de la información, utilizando modelos matemáticos resuelve tres conceptos básicos:

1. La medición de la cantidad de información en la fuente de información.
2. La capacidad del canal de comunicación para transferir información.
3. La codificación, como medio de utilizar el canal en toda su capacidad.

2.-Fuente de información discreta

Es conveniente considerar en primer término una fuente de información discreta, para ello recordemos que una señal analógica se puede convertir en discreta, mediante la aplicación del teorema del muestreo y su cuantificación en niveles predeterminados, y a partir de esto tratarla como un caso particular.

Como fuente de información discreta, se puede suponer una caja negra "S", que produce símbolos o mensajes "mi", que luego se transformarán en señales.



Es como un bolillero, que tiene bolillas con una determinada probabilidad de salida, por ejemplo, si fuese el alfabeto, contiene como símbolos las 28 letras del alfabeto español, o si fuese una señal digital contiene los dos estados de una fuente binaria.

El modelo básico de una fuente de información discreta de “mq” símbolos es:

$$S = \begin{bmatrix} m_1, m_2, m_3, \ldots\ldots\ldots m_q \\ p_1, p_2, p_3, \ldots\ldots\ldots\ldots p_q \end{bmatrix}$$

Cada símbolo aparece asociado a una determinada probabilidad. Por supuesto, la sumatoria de todas las probabilidades de aparición de los “mi” símbolos de esta fuente discreta “S” es:

$$\sum_{i=1}^{q} p_i = 1$$

Desde este punto de vista general, podemos asumir que la función de un sistema de comunicaciones es transferir desde la fuente de información, al receptor, una secuencia de símbolos que son seleccionados entre un número finito de símbolos predeterminados.
Es decir, en determinados intervalos de tiempo estos símbolos son transmitidos. Los mismos son conocidos por el receptor, aunque éste no conoce a priori cual será, en cada momento, el seleccionado para ser transmitido.

La pregunta en el lado receptor no es ¿qué símbolos? Sino ¿cuál o cuáles de esos símbolos? y generalmente, no todos los símbolos tienen la misma probabilidad de ser seleccionados y aparecer para ser transmitido.

De la misma manera, cuando una señal continua es cuantificada, el receptor conoce todos los niveles de cuantificación posibles, pero necesita determinar, “cual” de esos niveles son los que se han transmitido. Por consiguiente, se conocen todas las posibles señales (o símbolos) que se podrían transmitir, e irán apareciendo de acuerdo a una determinada probabilidad y el problema del receptor es saber cual de ellas es la que se ha transmitido.

Desde este punto de vista, podemos considerar a la fuente de señales continuas, como un caso particular de la fuente de información discreta.

3.- Medida de la Información. Diferencia entre el Bit y el Binit

En los comienzos de la electrónica, las señales eran prácticamente consideradas como hechos naturales, poseían anchos de banda determinados, requerían determinada fidelidad en la reproducción, y estos requerimientos se satisfacían por medios bastante directos. La aparición de la modulación de frecuencia de banda ancha significó un sacudón para los ingenieros. Durante y después de la segunda guerra mundial, varios hombres de ciencia comenzaron a investigar a fondo la naturaleza de las señales, tratando de contestar la pregunta en esencia, ¿que enviamos a través de los sistemas de comunicación?
¿Como podemos caracterizar y medir lo que enviamos?, ¿de que modo podemos establecer comparaciones cuantitativas válidas, entre los diversos tipos de señales?

En 1948, Norbert Wiener publicó un libro "Cybernetics", en el cual se explayaba sobre la teoría de la comunicación, en el mismo año, Claude Shannon, otro matemático, publicó un articulo: 'A mathematical theory of comunication'. (Teoría matemática de la comunicación).

Shannon encontró una manera de caracterizar las señales mediante una magnitud que denominó “cantidad de información” o simplemente “información”.

En la teoría de la comunicación, la palabra "información", o “cantidad de información” es una expresión técnica y con valor cuantitativo: **es aquello que es producido por la fuente de información discreta para ser transferido al usuario**. Esto implica, que previo a la transmisión, si bien la información no esta disponible en el destino, si se conocen todos los posibles símbolos a transmitir, pues provienen de una fuente discreta que está definida.

Como se ve no tiene que ver con el conocimiento o comprensión de algo, como sugiere su uso común.

En ese sentido el concepto de información está asociado fuertemente al de incertidumbre.

Veamos un ejemplo intuitivo, en el que se mezclan los dos significados de la palabra información, que nos ayudará a determinar su concepto matemático.
Un hombre quiere viajar a Córdoba, y para determinar que ropa debe llevar llama a la oficina meteorológica, de la que se supone recibirá alguno de los siguientes pronósticos:

1. El sol va a salir.
2. Lloverá.
3. Habrá un huracán y nevará.

La cantidad de información, brindada por cada uno de esos “mensajes o simbolos”, es bastante diferente entre si, lo mismo que la probabilidad de que alguno de los eventos sea el que se produzca.

1.- El primero prácticamente no contiene información válida para el usuario, pues es seguro que el sol saldrá, como sale todos los días. Esta información no tiene sentido que sea transferida al usuario

2.- Lloverá le adelanta una cierta información, que el viajero previamente no poseía, pues en Córdoba, la lluvia no es cosa de todos los días, aunque hay cierta probabilidad de que pueda ocurrir.

3.- En cambio habrá un huracán y nevará, contiene un .gran valor informativo, pues este es un evento raro, no común, poco probable

Se nota, que los mensajes han sido ordenados según una “probabilidad decreciente y con crecimiento de información”. Cuanto menos probable o mas incierto es el acontecimiento, mayor es la información que se transfiere. Es evidente que “habrá un huracán y nevará" puede determinar incluso que la persona no viaje o necesite ropa especial.

Entonces, si una fuente, puede producir varios mensajes o símbolos, cada uno, con una probabilidad determinada, por ejemplo, si uno de esos mensajes es “A”, y la probabilidad que sea necesaria su transmisión P(A), se puede considerar, que la información asociada con A es:

$$I_A = f(P_A)$$

Entonces debemos determinar $f(P_A)$

Para ello, se pueden utilizar ciertos elementos intuitivos de razonamiento:

a.- La medida de información debe ser un número real y positivo (no hay razón en contra de ello).
b.- Si: $P_A = 1$

Entonces, no hay incertidumbre, hay certeza (tal como la salida diaria del sol), y en realidad no hay información, o mejor dicho no es necesario enviarla y ni siquiera producirla, es decir $I_A = 0$

c.- Si el mensaje “A” es menos probable que el “B”, o sea:

$$P_A < P_B$$

Entonces $I_A > I_B$

Matemáticamente estas condiciones podemos expresarlas como:

$$f(P_A) > 0, donde..0 \le P_A \le 1$$
$$f(P_A) > f(P_B), para..P_A < P_B$$

$$Lim P_{A \to 1} f(P_A) = 0$$

Distintas funciones cumplen estas condiciones. La decisión final, surge de considerar la transmisión de mensajes estadísticamente independientes (es decir, la ocurrencia de uno no depende del anterior).

Cuando se recibe un mensaje “A” se reciben I_A unidades de información. Si un segundo mensaje “B” es recibido, y es independiente del primero, la información total recibida es la suma de la información “$I_A + I_B$”
Supongamos ahora, que se recibe un mensaje $C = AB$. Si “A” y “B” son estadísticamente independientes, será:

$$P_C = P_A P_B$$
$$I_C = f(P_A P_B)$$

La información recibida *Ic* será:

$$I_C = I_A + I_B = f(P_A) + f(P_B)$$

Hay solo una ecuación que satisface todas estas condiciones, y es la función logaritmo, luego:

$$f(P_A) = -k \log_B P_A$$

Donde *k*, es una constante positiva y *B* la base de los logaritmos. Haciendo *k= 1*, podemos llegar a la siguiente definición de información:

$$I_A = -\log_B P_A = \log_B \frac{1}{P_A}$$

Como $0 \leq P_A \leq 1$, la cantidad de información, es siempre positiva.

Especificar la base *b*, de los logaritmos es equivalente a definir la unidad de información o de cantidad de información.
Es usual tomar como *b = 2*. La unidad correspondiente de información en este caso es el bit, como contracción de binary digit.

Es decir:

$$I_A = \log_B \frac{1}{P_A} bits$$

Desde el punto de vista técnico - práctico, el código Morse, es una muestra de como la medida de la información está asociada a la probabilidad de aparición de un símbolo. En dicho código, construido en base a una combinación de puntos y rayas, se asigna, para aumentar la velocidad de transmisión, la duración más breve (un solo punto) a la letra mas frecuente en el idioma inglés, la "e". Es decir, se le "asigna menor cantidad de información a la letra que tiene mayor probabilidad de aparición".

Es preciso aclarar, que desde un punto de vista teórico general, no se debe confundir la unidad de información, **el bit**, con el pulso que lo materializa, que recibe en la teoría de la información el nombre de **binit**.

El binit, o sea el pulso, transportará o no 1 bit de información, dependiendo de las probabilidades.

Supongamos una fuente binaria de información, esto es una fuente constituida por dos símbolos, el “0” y el “1”, con probabilidades 1/4 y 3/4 respectivamente.
En este caso, cada símbolo puede ser representado por la ausencia o presencia de un pulso, es decir, por un binit.

El binit “0” transporta: $\log_2 \frac{1}{1/4} = \log_2 4 = 2 bits$

El binit “1” transporta: $\log_2 \frac{1}{3/4} = \log_2 \frac{4}{3} = 0.42 bits$

Si las probabilidades de aparición del “0” y el “1” fueran iguales, como ocurre en la computación o en la transmisión digital, la información de los binits que representan al “0” y al “1” es: $\boxed{\log_2 \frac{1}{1/2} = \log_2 2 = 1bit}$

En estos casos no hay confusión posible, y se acepta la misma denominación para el pulso y la cantidad de información.

Ejemplos de medición de información:

a.- Información asociada a la caída de una moneda. Es evidente que los dos sucesos que son:

A: ……cara
B: ……cruz

Tendrán igual probabilidad $P = P_A = P_B = P(e) = 1/2$

Donde: $I(e) = \log_2 \frac{1}{P(e)} bits = \log_2 \frac{1}{1/2} = \log_2 2 = 1bit$

Es decir, al preguntar si salió cara y al contestar si o no, es decir, 1 o 0, estoy dando un bit de información. El bit es, por lo tanto, la información de un suceso cuya $P(e) = 1/2$

b.- Información asociada a la aparición de una letra, entre 32 equiprobables posibles. En el caso del teclado de una teletipo, con teóricamente 32 símbolos, cuando sale una letra, por ejemplo “x”. ¿Qué información se está dando?

Si cada probabilidad es $P = P_A = P_B = P_{32} = 1/32$; la información asociada en esta fuente para cada uno de los “mi” símbolos es:

$$\boxed{I_{32} = \log_2 \frac{1}{1/32} bits = \log_2 32 = 5bit}$$

Es decir, **cada una de las letras contiene 5 bits de información**, y puede representarse por cinco dígitos binarios.

c.- Si una fuente produce cuatro símbolos: A, B, C, D, con probabilidades 1/2, 1/4, 1/8, y 1/8 respectivamente. Como vemos, la suma de las probabilidades debe ser igual a 1.
Se quiere hallar la información por un mensaje integrado por los cuatro símbolos.

Símbolo	P	$I = \log_2(1/P).bits$
A	1/2	1
B	1/4	2
C	1/8	3
D	1/8	3

Si: x = B A C A y la aparición de cada símbolo es estadísticamente independiente:

$$P_x = P_B P_A P_C P_A = \frac{1}{4x2x8x2} = P_{128} = 1/128$$

$$I_{128} = \log_2 \frac{1}{1/128} bits = \log_2 128 = 7 bits$$

d.- Información de una foto en blanco y negro. Para resolver este problema, se debe tomar en cuenta que en este tipo de foto hay un grano mínimo, es decir, hay un mínimo elemento cuya superficie puede considerarse con una tonalidad uniforme.
Este grano mínimo, permitirá dividir a toda la fotografía en cuadraditos cuya superficie sea del orden del tamaño de ese grano mínimo. Supongamos que en cada línea haya 500 elementos y en cada columna también 500.

Es decir. 500 filas y 500 columnas, luego el número de elementos será: 500.500=250.000 elementos.

Cada uno de esos elementos, tendrá un determinado valor de coloración o tono de gris. Si se adopta un modelo en que se considera que se pueden distinguir 8 niveles de gris y se supone que todos los elementos son independientes entre sí (que si bien no es cierto, simplifica el problema), la fuente de información será una cuyos símbolos son los 8 niveles de grises.

Entonces cada vez que "aparece" un gris se tendrá:

$$\boxed{I = \log_2 \frac{1}{1/8} bits = \log_2 8 = 3 bits}$$

y la información asociada a la imagen será:

$$\boxed{I(imagen) = 3 bits.x.250000 elementos = 750000.bits}$$

4.- Entropía de una fuente de información discreta

La información hasta aquí, ha sido definida en términos de los mensajes individuales o de cada uno de los símbolos que la fuente puede producir. Esto no es, sin embargo, una descripción práctica de la fuente de información en un sistema de comunicaciones. Un sistema no es diseñado para cada mensaje en particular, sino para todos los mensajes posibles.

Se puede utilizar como ejemplo un concepto ya conocido. La potencia instantánea de una fuente de energía, es una función del tiempo, fluctuando continuamente. En estos casos, se comprende que para la mayoría de los casos es mas práctico hablar de potencia media, y no dar una tabla con todos los valores posibles.

De la misma manera, una fuente produce en cada instante un determinado valor de información, y la misma va variando en forma aleatoria. Resulta conveniente describir la fuente en términos de información media, y para ciertos cálculos completarla con los valores máximos y mínimos.

Esta información media es conocida con el nombre de **"entropía de la fuente de información"**. **El nombre de entropía y el símbolo utilizado para designarla H(s),** crea cierta confusión con el término llamado y designado de la misma forma en la mecánica estadística y en la termodinámica y si bien se puede hacer cierta analogía, hay que tener en cuenta que son cosas completamente diferentes, llamadas con el mismo nombre.

Expresión matemática de la entropía.

Sea "mi" el número de símbolos estadísticamente independientes, que puede producir una fuente.

Cuando el símbolo "mi" es transmitido, **su información** vale: $$I.i = \log \frac{1}{Pi} bits$$

Donde ***Pi*** es la **probabilidad del símbolo** genérico "***mi***".

En un mensaje largo con ***N>>1*** símbolos, el símbolo "***mi***" aparecerá ***NPi*** veces. Lo mismo ocurrirá con todos los otros, donde N será multiplicada por la respectiva probabilidad.

La información total del mensaje, considerando todos los símbolos será:

$$N.P_1 I_1 + N.P_2 I_2 + ... + N.P_m I_m = \sum_{j=1}^{m} N.P_j I_j bits$$

Puesto que han aparecido N símbolos, la información media por símbolo será:

$$\frac{1}{N} \sum_{j=1}^{m} N.P_j I_j = \sum_{j=1}^{m} .P_j I_j$$

Se define, en consecuencia, la entropía de una fuente discreta como:

$$H(s) = \sum_{j=1}^{m} .P_j I_j = \sum_{j=1}^{m} .P_j \log \frac{1}{P_j} bits / símbolo$$

Como se observa se trata de un promedio estadístico.

¿Cuál es el significado de este concepto?

En una fuente, los distintos símbolos que se producen están caracterizados de acuerdo a una determinada probabilidad, y aunque no se puede conocer exactamente, cual es el símbolo que aparecerá en un determinado instante, y por lo tanto no se conoce su valor de información, en promedio, se puede esperar que la fuente produzca:

H(s) bits de información por símbolo, o **H(s) N** bits, en un mensaje de N símbolos.

Si todos los símbolos de la fuente fueran equiprobables, es fácil ver que $H(s) = \log m$ y se pude demostrar que en ese caso la entropía es máxima, es decir: $0 \le H(s) \le \log m$

Veamos la variación de ***H(s)*** para el caso de una fuente binaria (***m = 2***}.

Las probabilidades serán

"*p*" para uno de los símbolos y "$q = 1 - p$" para el otro, pues $q + p = 1$

Entonces:

$$H(s) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{(1-p)} bits / símbolo$$

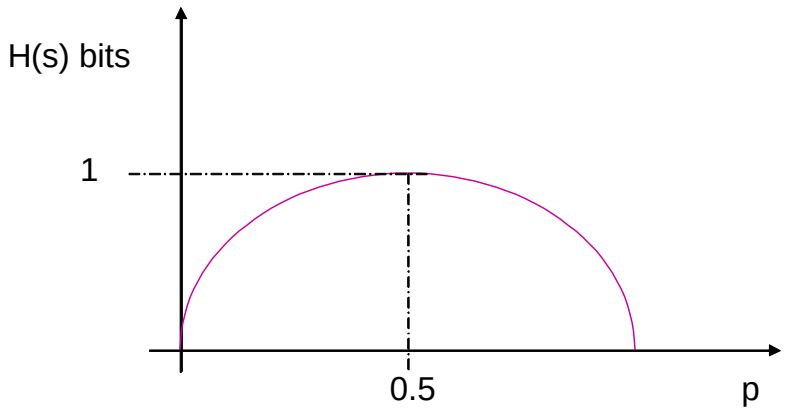
Supongamos, que $p = 0$ y $q = 1$, tendremos certeza que no hay información, porque

$$H(s) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{(1-p)} bits / símbolo = 0$$

Si ahora hacemos que $p = 1/2$ y $q = 1/2$ la entropía será máxima, de acuerdo a lo ya visto.

$$H(s) = \frac{1}{2} \log_2 \frac{1}{\frac{1}{2}} + (1 - \frac{1}{2}) \log_2 \frac{1}{(1 - \frac{1}{2})} = \frac{1}{2} \log_2 2 + \frac{1}{2} x \log_2 2 = 1 bit / simbolo$$

Si graficamos esta función, que se llama función entropía y que se da en el caso particular de la fuente binaria, tendremos:



El máximo será para *p = 0,5* y en este caso: *H(s} = 1 bit/ símbolo*

5.- Extensión de una fuente discreta

Dada una determinada fuente, definida en la forma

$$S = \begin{bmatrix} m_1, m_2, m_3, ...m_q \\ p_1, p_2, p_3, ...p_q \end{bmatrix}$$

Esto se define como enésima extensión de esa fuente S, que llamaremos S^n, a una nueva fuente en la cual, cada uno de sus símbolos, son cada uno de todas los posibles combinaciones de los símbolos de primera fuente tomadas de a "n" y cuyas probabilidades resultan del producto de las probabilidades originales

Por ejemplo, la segunda extensión de la fuente anterior seria

$$S = \begin{bmatrix} m_1m_1...m_1m_2...m_1m_3..m_2m_1...m_2m_2..m_2m_3....m_3m_1..m_3m_2...m_3m_3 \\ p_1p_1...p_1p_2...p_1p_3....p_2p_1...p_2p_2....p_2p_3....p_3p_1....p_3p_2...p_3p_3 \end{bmatrix}$$

El número de símbolos de la nueva fuente, si la fuente original tiene "q" símbolos y son tomados de a "n" es: q^n , que en este caso particular resulta igual a 9 símbolos.

Con el razonamiento anterior, se puede demostrar que la entropía de la fuente extendida es:

$$H(S^n) = nH(s)$$

Siendo *H(S)* la entropía de la fuente original.

Supongamos "una fuente binaria de datos, con muy alta probabilidad de que los símbolos sean equiprobables, es decir:

$$S = \begin{bmatrix} 1.....0 \\ 1/2...1/2 \end{bmatrix}$$

La tercera extensión es, con los dígitos agrupados en tribits

$$S^3 = \begin{bmatrix} 111..110...100..101...001..000....010..011 \\ 1/8...1/8...1/8...1/8..1/8...1/8...1/8....1/8 \end{bmatrix}$$

y la entropía $H(S^3) = 3H(s) = \log_2 8 = 3 bits / símbolo$.

6.- Fuentes con memoria (o de Markov). Aplicación al idioma inglés

En lo que ha visto hasta ahora, se ha establecido explícita e implícitamente, que la aparición de cada símbolo para ser transmitido, es independiente del que lo precede, es decir, la aparición de cada símbolo es un suceso estadísticamente independiente. En estos casos, se considera que la fuente de Información es una fuente sin memoria, pues la producción de un símbolo no está influido por los anteriores.

Pero la mayoría de las fuentes de información reales, no tienen esta característica, es decir, la aparición de un símbolo depende de los símbolos anteriores. A este tipo de fuentes se las llama con memoria o Fuentes de Markov. Se les asigna un orden, que es el número de símbolos precedentes de los cuales depende la aparición del nuevo símbolo.

Una fuente de Markov de primer orden, es una fuente en la cual, la aparición de un símbolo depende de cual haya sido el último símbolo producido, en las de segundo orden la dependencia es respecto de los dos últimos símbolos y así sucesivamente. ...
En una fuente sin memoria S = (m_1, m_2, ….mq), se tomaban en cuenta las probabilidades de aparición de cada uno de los símbolos Pj, es decir, las probabilidades a priori. En las fuentes con memoria hay que hacer intervenir las probabilidades condicionales.

Por ejemplo, en una fuente con memoria de primer orden, se debe definir la probabilidad

$$Pu = P(m_j / m_i)$$

Es decir, la probabilidad que salga el símbolo j dado el i, mientras que en una de tercer orden.

P (mi / $m_{i1,}$ m_{i2}, m_{i3}) probabilidad de aparición de “mi” si está precedido por $m_{i1,}$ m_{i2}, m_{i3}

Un idioma es un buen ejemplo de fuente del tipo de Markov, está muy bien estudiado para el idioma inglés. Así, en inglés escrito, la probabilidad a priori de la aparición de la letra U es del 0,02, pero si la letra previa fue la O, la probabilidad condicional de aparición de una U dada O es P(U/O)=1 (sucede lo mismo en el idioma español), mientras que en contraste la P(U/W)= 0,001. En este caso se debe hablar de la entropía condicional y para hallarla hay que considerar la historia de la fuente.

La información producida por un símbolo es:

$$Iji = \log \frac{1}{P(mj / mi)}$$

Y en promedio, la información media de una fuente de Markov, o entropía es:

$$Hc = \sum_i \sum_j PiP(mj / mi) \log \frac{1}{(mj / mi)}$$

Esto es fácil de ver intuitivamente, pues con las probabilidades condicionales disminuye la incertidumbre.
.

Una fuente que produce símbolos con una dependencia de los anteriores, si se la trata como sin memoria, es una fuente redundante, significando esto, que son generados símbolos que no son realmente necesarios para la transmisión de la información. En efecto, ¿es realmente necesario transmitir una U después de una O?

Se puede definir a la redundancia como: 1 -Hc / H(s)

Donde Hc y H(s) tienen los significados ya vistos, es decir, “Hc” la entropía de la fuente real con memoria y “H(s)” la entropía de la fuente, como si todos los símbolos fueran estadísticamente independientes.

La fuente sin memoria tiene la ventaja de que requiere de un Hw y un Sw mas sencillo, pero al tener mayor entropía requerirá de mayor ancho de banda como consecuencia de la mayor tasa de información.

Las fuentes con memoria, si bien son de una realización más compleja, tienen la ventaja de requerir una menor cantidad de información.

A continuación, para ilustración, se presentarán resultados del estudio de la estructura del lenguaje o idioma inglés, que es uno de los más estudiados, como fuente de Markov. Sobre la base de utilizar 26 letras y un espacio, es decir 27 símbolos, se obtienen los siguientes modelos:

1. Todos los símbolos son equiprobables, es decir, cada símbolo no depende del anterior, fuente sin memoria con m= 27, luego:

$H = \log_2 27 = 4{,}75$ bits / letra Siendo este es el máximo valor de entropía.

2. Fuente sin memoria, pero considerando la aparición de las letras con sus probabilidades reales, es decir:

P (espacio) = 0,1858 P(A) = 0,0642 …P(B) =0,012 ……P(Z)=0,0005

La entropía en este caso resulta:

H(s)= 4,03 bits / letra.

3. Cada letra depende solamente de la última transmitida, y además se consideran las probabilidades reales de que ello ocurra:

Hc = 3,32 bits / letra.

4. .Cada letra depende de las dos anteriores, con sus probabilidades reales:

Hc = 3, 10 bits / letra.

5. Tomando hasta 8 letras y probabilidades reales: Hc = 2bits / letra.

Considerando este valor de la entropía se ve que si se transmite un texto en inglés, considerando todas las letras como equiprobables, la redundancia es:

(1-2/4, 75) = 50 %

Es decir, que en una secuencia suficientemente larga, la mitad de las letras son innecesarias, y que el mensaje podía ser reconstruido sin ellas.

6. El caso límite, es una fuente de orden “n”, y considerando las probabilidades reales:

Hc = 1 bit / letra.

Entonces desde el punto de vista de la eficiencia (al tratar de transmitir la mayor cantidad de información en menor tiempo), la redundancia es innecesaria y no deseable. Pero por otro lado, la redundancia es un medio de resolver ambigüedades provocadas por errores en la recepción. Cuanto más se considere el efecto Markov, más eficiente es la transmisión, y ello es la base de los métodos de compresión de la información, que permiten reducir ancho de banda. Por otro lado, si se quiere detectar errores en la transmisión, se agrega algo de redundancia.

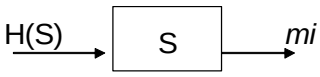
Un ejemplo conocido, es el de agregar bits de paridad para determinar si la recepción es correcta. Como se ve, si bien la solución final es de compromiso, el desarrollo de la microelectrónica actual permite llegar a relaciones de comprensión de la información altos.

En el caso límite solo se necesitaría un bit/letra, con lo que se puede transmitir más rápido, pero la complejidad técnica es mucho mayor, pues no solo debe haber una memoria de un orden muy elevado, sino un sistema y algoritmos de codificación en función de las probabilidades condicionales.

En los teletipos se utilizó el primer modelo, es decir, fuente equiprobable. El modelo número dos es prácticamente el código Morse, en donde las letras que aparecen más comúnmente tienen una longitud de codificación menor.

7.- Información media por unidad de tiempo o Tasa de Información:

Hasta ahora se han considerado las fuentes de información como productoras de símbolos pero sin tomar en cuenta el tiempo. En la realidad se producen símbolos sucesivos en el tiempo, a una determinada velocidad.
Es decir, considerando una fuente **S**, que produce “*mi”* símbolos



Esta fuente tiene una determinada entropía H(s) y produce un flujo de información. Se define la información media por unidad de tiempo o tasa de información “R” como:

$$R = mi..H(s) = bits / seg$$

8.- Codificación

Mediante la codificación, lo que se hace es adaptar la estructura de los símbolos a las características del canal. Si se tuviera que transmitir 27 letras y el canal solo acepta niveles que representan ceros y unos, lo que se hace es asignar grupos de ceros y unos a cada una de las letras. La forma en la que se hace la codificación es importante para adaptar la forma en que esa información pasará por el canal, para que dicha información fluya con la mayor eficiencia, es decir, que se obtenga la mayor información por unidad de tiempo.

Supongamos una fuente de "q" símbolos: S=(m1, m2,….,mq) y un alfabeto del código: X=(x_1. x_2, …..,x_r), de r letras.

Se llama código a la convención que hace corresponder a cada secuencia de los símbolos de la fuente con otra secuencia de caracteres o letras del alfabeto del código.

Ejemplo: Sean S=(A, B, C, D) y X=(0, 1),

Una codificación posible será:

A …00, B…01, C…. 10, D….. 11,

en la cual se ve que a cada símbolo de la fuente le corresponde una secuencias de los caracteres del alfabeto del código.

Se define como palabra del código al conjunto de letras del alfabeto código Xi = ($x_{1.....}$ x_n)

En el ejemplo anterior una de las palabras código es por ejemplo: 00

a.- Clasificación de los códigos

A través de la siguiente clasificación se verá una serie de características generales de todos los códigos.

a.1 Código Singular.

Sea por ejemplo, la fuente $S = S_1, S_2, S_3, S_4$ y un alfabeto binario X (0,1).

Si se define: $S_1.....0, S_2.....11, S_3.....00, S_4.....11$

A cada símbolo le corresponde una secuencia fija del alfabeto del código, aunque se puede observar que a una palabra de código 11, le corresponden dos símbolos de la fuente. Este es un código singular y seguramente habrá problemas para decodificar pues al aparecer 11 no se sabrá si es S_2 o S_4.

a.2 Código No Singular.

En este caso todas las palabras del código son distintas, por ejemplo:

$S_1.....0, S_2.....11, S_3.....00, S_4.....01.$

a.3 No Unívocamente decodificables

Supongamos recibir el siguiente mensaje: 0011, tomando como base el código anterior. No hay forma de decidir si los símbolos emitidos son S_3S_2 o $S_1S_1S_2$.

Es decir, que en este caso son singulares la segunda y la tercera extensión de la fuente.

a.4 Códigos Unívocamente decodificables.

Supongamos tener: $S_1.....0, S_2.....01, S_3.....011, S_4.....0111.$

Este código es obviamente no singular y además es no singular para cualquier extensión de la fuente. Es un código unívocamente decodificable, pues cada vez que aparece un O comienza una nueva palabra. Pero tiene un inconveniente desde el punto de vista de la velocidad de decodificación.

Para visualizar ese inconveniente, veamos los siguientes ejemplos:

a)	b)
SI 0	SI 0
S2 01	S2 10
S3 011	S3 110
S4 0111	S4 1110

Ambos son códigos unívocamente decodificables. En la decodificación de a) y b) hay una diferencia de velocidad, cuando se recibe por ejemplo 01 del código a) , se debe esperar un tiempo pues no se sabe si se terminó la palabra o aparecerá un 1 (que lo transformará en 011). Por lo tanto, se necesita que aparezca el cero correspondiente a la próxima palabra para saber que terminó la anterior. En cambio en b) al aparecer un cero se sabe que la palabra en cuestión terminó de ser transmitida. El código a) es no instantáneo, en cambio el b} es instantáneo. Diremos que el código es instantáneo cuando ninguna palabra es sufijo de otra.

b.- Longitud Media de un Código. Código compacto

Si la codificación resulta

SI 0
S2 10
S3 110

S4 111
En todos los ejemplos vistos cada palabra Xj tiene asociada una longitud "li", que corresponde a la cantidad de letras o caracteres que la componen. En el ejemplo anterior:

Longitud de S1 =1
Longitud de S2 =2
Longitud de S3 =3
Longitud de S4 =3

Estas son las longitudes correspondientes a cada símbolo. Cuando se transmiten muchos símbolos en sucesión es importante, dada la sucesión ya codificada, saber cuantos caracteres o letras se utilizan en promedio.

La forma de calcular dicha longitud media es mediante la expresión:

$$L = \sum_{i=1}^{q} P(Si)Ii$$

Que se obtiene por un razonamiento similar al que se utilizó para obtener la expresión de la entropía. Aquel código que además de ser instantáneo, tiene la menor longitud media, se lo llama compacto. Este tipo de código se obtiene por medio del método de Huffman. La conclusión de este tema es que al construir códigos, desde el punto de vista de la eficiencia (velocidad de transmisión), interesan los códigos compactos e instantáneos.

9.- Capacidad de Canal

Para un estudio general de un sistema de comunicaciones es útil pensar los terminales del sistema como perfectos (libres de ruido, distorsión, sin limitaciones de ancho de banda, etc.) y adjudicar todas las limitaciones al proceso que tiene lugar entre el transmisor y el receptor, es decir al canal.

El **canal de comunicaciones** en la teoría de la información es una abstracción matemática, que representa todo el proceso de la transmisión más los fenómenos que tienden a restringir dicha transmisión y que ocurren no solo en el medio de transmisión, sino también en las diversas partes constitutivas del sistema. El hecho de la existencia de limitaciones físicas fundamentales (ruido y ancho de banda) a la transmisión, lleva al concepto de capacidad del canal.

Así como la **velocidad de entropía** (información media por unidad de tiempo) mide la cantidad de información que una fuente produce en un tiempo dado, **la capacidad es una medida de la cantidad de información confiable que el canal puede transferir al destinatario por unidad de tiempo.**

La capacidad del canal se simboliza con C y su unidad es bits / seg.

El **teorema fundamental de la teoría de la información** puede ser enunciado en términos de R y C como:

"Dado un canal de capacidad C y una fuente que produce una información media por unidad de tiempo R, si $R \leq C$, existirá una técnica de codificación tal, que la

salida de la fuente pueda ser transmitida por el canal, con una frecuencia arbitrariamente pequeña de error a pesar de la presencia de ruido, si $R > C$, no es posible transmitir sin error"

A través de un sencillo ejemplo, se podrá ver como funciona un aspecto de esto:

Supongamos una fuente con cuatro símbolos, con sus probabilidades

a1 = 1/2; a2 = 1/4; a3 = 1/8; a4 = 1/8

que son emitidos a una velocidad de 100 símbolos por segundo.

La información media será:

$$H(s) = 1/2\log_2 2 + 1/4\log_2 4 + 2x1/8\log_2 8 = 1.75 bits/símbolo$$

y la "*tasa de información*" será

$$R = m_i xH(s) = 100 símbolos/seg x 1.75 bits/símbolo = 175 bits/seg$$

Supongamos que el canal tiene también una capacidad: *C = 175 bits/ segundo*

Luego como: $R \leq C$

Es posible efectuar la transmisión sin errores. Siendo cuatro símbolos la primera codificación que se nos ocurre sería:

a1 = 01; a2 = 10; a3 = 11; a4 = 10

Este código tiene una longitud media $Li = 2 bits/símb$ y como se emiten $mi = 100 símb/seg$

La capacidad del canal debería ser $C = mixLi = 100 símb/seg x 2 bits/simb = 200 bits/seg$

Lo que excede la capacidad establecida.

Para solucionar el problema, se debe encontrar un código más eficiente, **un código compacto**

Aplicando el *método de Huffman* se obtiene la siguiente codificación:

a1 = 0; a2 = 10; a3 = 110; a4 = 111

En este código la longitud media será:

$$Li = \sum_{i=1}^{q} P(Si) xli = (\frac{1}{2} x1 + \frac{1}{4} x2 + 2x\frac{1}{8} x3) = 1.75 bits/símbolo$$

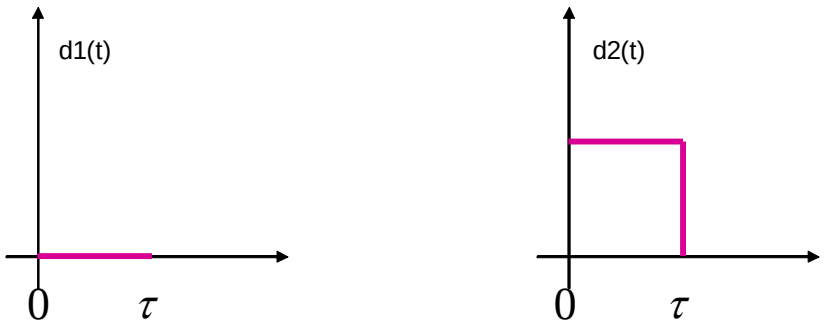
Con lo cual $C = mixLi = 100simb/segx1.75bits/símb = 175bits/seg$

Entonces, ahora la velocidad de emisión de símbolos, concuerda, con la capacidad del canal.

Capacidad del canal binario sin ruido

El canal binario que es aquel que transmite información tomando dos estados eléctricos, tensión o corriente, es un caso particular del canal discreto. Si la relación señal a ruido es grande y la probabilidad de error pequeña, el canal puede considerarse como sin ruido o ideal y cualquier secuencia de símbolos transmitido se interpreta correctamente en la recepción.

Si los símbolos son:



El canal queda definido por la duración τ de cada pulso. Al cambiarlo, cambio el canal.
Si los símbolos son equiprobables se transmite la máxima cantidad de información, y en ese caso se tiene:

$$R\max = \frac{1}{\tau} bits/seg$$

Como no hay probabilidad de error, este valor se puede tomar como capacidad del canal.

Si en la fuente los símbolos binarios se agrupan de a “m”, es decir, si se tienen 2^m niveles o símbolos, y se asume que todos son equiprobables y no hay probabilidad de error la tasa de información y la capacidad en este caso sería:

$$R\max = C = \frac{1}{\tau}\log_2 mxbits/seg$$

Capacidad del canal binario con ruido

En presencia de ruido, la capacidad del canal discreto disminuye, debido a errores en la transmisión, y no coincidirá con Rmáx .En este caso el tipo mas sencillo que se puede estudiar, pero también el mas importante desde el punto de vista práctico, es el conocido como canal binario simétrico **(BSC)** .Un canal de este tipo introduce aleatoriamente

errores en los bits, cambiando “unos” por “ceros” y “ceros” por “unos” con igual probabilidad.

El canal binario simétrico sólo necesita de un parámetro para ser caracterizado, pues la probabilidad de error es la misma tanto para los “unos” como para los “ceros”. Se utilizará como parámetro a “**p”**, la probabilidad de error por bit enviado.

Llamaremos Tx al mensaje enviado por la fuente y Rx al mensaje recibido por el receptor. Si en primera aproximación imaginamos que el mensaje consta de un solo bit, podemos establecer el siguiente diagrama :

$$p(x_1) = \alpha \text{.....} x_1 = 0 \text{.....} 1 - p \text{.....} y_1$$

$$p(x_2) = \alpha \text{.....} x_2 = 1 \text{.....} 1 - p \text{.....} y_2$$

y las siguientes probabilidades condicionadas:

$$p(0/0) = p(1/1) = 1 - p$$

$$p(0/1) = p(1/0) = p$$

Que se lee:

La probabilidad de recibir un bit, habiendo enviado ese bit es uno menos la probabilidad del error.

La probabilidad de recibir un bit, habiendo enviado el otro, es la probabilidad de error.

A partir de las probabilidades condicionadas, puede definirse una entropía del error o la equivocación H(e), que será igual a:

$$H(e) = px\log_2 1/p - (1-p)x\log_2\left[1/(1-p)\right]$$

La capacidad del canal, si se sigue considerando un bit será

$$C = 1 - H(e) = 1 + px\log_2 p + (1-p)x\log_2(1-p)$$

Como la fuente emitirá símbolos con una tasa de información ***R***, la capacidad del canal binario simétrico, será:

$$C = R\left[1 + px\log_2 p + (1-p)x\log_2(1-p)\right]$$

Se puede ver que para *p =0,5*, es decir cuando la equivocación es máxima, la *C =0* y no se
puede transmitir información.

Para *p = 0,001* la capacidad es: *C = 0.989 R*

Lo que indica que se deberá proceder a algún cambio para recibir sin error.

Capacidad del canal analógico con ruido

Para hallar la capacidad del canal, lo que se hace es transformar la señal analógica en una discreta. lo que se logra a través de la cuantificacíón y el muestreo. Para ello se considera:

S/N: Relación señal – ruido

A la salida del canal la potencia media de la señal es **S** y la del ruido es **N**, siendo entonces la potencia total recibida = S+N.

Los niveles de tensión en el receptor deberán estar como mínimo espaciados en $\sqrt{N}$ voltios, que es la tensión eficaz de ruido, para que puedan ser identificados, pues si hay un espaciamiento menor, el ruido va a tapar esta diferencia de niveles y no se los puede identificar.

Si la máxima tensión de salida RMS es $\sqrt{S+N}$, y los niveles están espaciados en $\sqrt{N}$, el número máximo de estados cuánticos del canal será:

$$M = \frac{\sqrt{S+N}}{\sqrt{N}} =$$

La información, asimilando el número de estados cuánticos a símbolos de una fuente de información, y asumiendo que todos los estados serán equiprobables, será:

$$I = \log_2 M = \log_2 \sqrt{1+\frac{S}{N}} = \frac{1}{2}\log_2(1+\frac{S}{N})$$

De acuerdo al teorema del muestreo, para reproducir la señal analógica se deben enviar por lo menos *2 fmax,* muestras en la unidad de tiempo, por lo cual, se requerirá una capacidad tal que:

$$C = \frac{2f\ \max}{2}\log_2(1+\frac{S}{N})$$

Ya que $\frac{1}{2}\log_2(1+\frac{S}{N})$, es la información en cada muestra y *fmax* = *B*, resulta

$\boxed{C = B\log_2(1+\frac{S}{N})}$ Ley de Shannon-Hartley

Si $B \Rightarrow \infty$, parecería que $C \Rightarrow \infty$, lo que no es posible. Rescribiendo “N” (que es el ruido) en la fórmula anterior

$$C = B\log_2(1+\frac{S}{\eta B})$$

Aquí $\eta = densidad..espectral..de..ruido$, y $S = potencia..de..señal$ aparece; una indeterminación. Resolviéndola se llega al siguiente límite teórico:

$$C_{B\Rightarrow\infty} = 1.44\frac{S}{\eta}$$

La Ley de Hartley -Shannon establece, dentro del límite descripto, como se puede "jugar" con el ancho de banda y la relación S/N para establecer la capacidad del canal.

Se puede deducir, de acuerdo a esa ley, la ley ideal para combinar el ancho de banda con la relación S/N. Considérese para ello una señal de banda base modulante de frecuencia máxima "*fmax*" en Hz.

Dicha señal modula idealmente a una portadora, que, como consecuencia de la modulación tiene un ancho de banda B. Esta señal modulada se aplica a la entrada de un demodulador ideal con una relación Si/Ni

La salida del demodulador es una señal caracterizada por *fmax* y una *So/No*

Como en el demodulador ideal la información de entrada y la de salida deben ser iguales.

$$I = B\log_2(1+\frac{Si}{Ni}) \qquad I = f\max x\log_2(1+\frac{So}{No})$$

En consecuencia

$$I = B\log_2(1+\frac{Si}{Ni}) = f\max x\log_2(1+\frac{So}{No})$$

$$(1+\frac{Si}{Ni})^{\frac{B}{f\max}} = (1+\frac{So}{No})$$ si despreciamos el "1", tenemos $$\boxed{\frac{So}{No} = \frac{Si}{Ni}^{\frac{B}{f\max}}}$$

Como se ve, en el sistema ideal, dentro de los límites vistos anteriormente, la *So/No* mejora con el aumento de la relación *B/fmax*, que figura en el exponente.

10.-Detección y Corrección de Errores

El objetivo principal en el diseño de todo sistema de comunicaciones es transmitir información tan rápido y exactamente como sea posible, dentro de las restricciones:

- Que imponen la potencia
- El ancho de banda

- Los medios económicos disponibles

Una vez terminado el diseño puede ser que alguno de los objetivos fijados no se cumplan, aún utilizando el mejor método de modulación y el más sofisticado de los demoduladores, en tales circunstancias se deben diseñar métodos de detección y posiblemente de corrección de los errores producidos en la transmisión de la información.

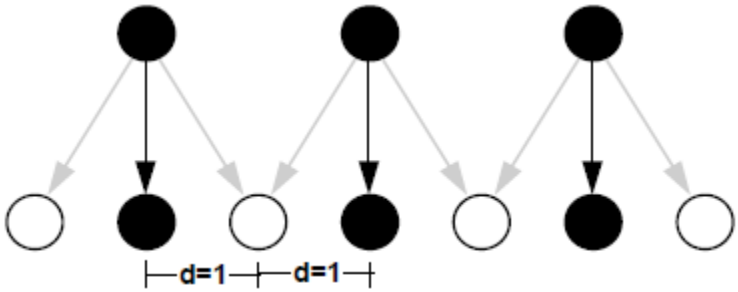
a.- Códigos de detección de error simple

Trataremos de estudiar códigos cuya principal característica es que permiten detectar si se ha producido o no error en la transmisión de la palabra de código. Aunque en caso de producirse, no son capaces de su corrección inmediata, su detección evita el uso de información incorrecta. El error se puede solucionar repitiendo la transmisión hasta que ésta desaparezca.

Si en un código binario de longitud constante, se utilizaran todas las combinaciones posibles de sus "n" dígitos binarios (2^n), resultaría imposible detectar si se ha producido error, ya que una combinación del código se transformaría en otra que también pertenece a él o sea no se deben utilizar todas las combinaciones posibles y así se detecta el error.

Ejemplo: si transmitimos dígitos decimales para lo que usamos el código BCD natural. En éste no se emplean más que 10 de las 16 combinaciones posibles con 4 dígitos binarios. Supongamos que se emite la combinación 0011 y tras producirse error en un dígito binario durante la transmisión se recibe la 1011; como no pertenece al código utilizado es inmediato deducir que nunca ha podido ser transmitida, por lo que el error será detectado.

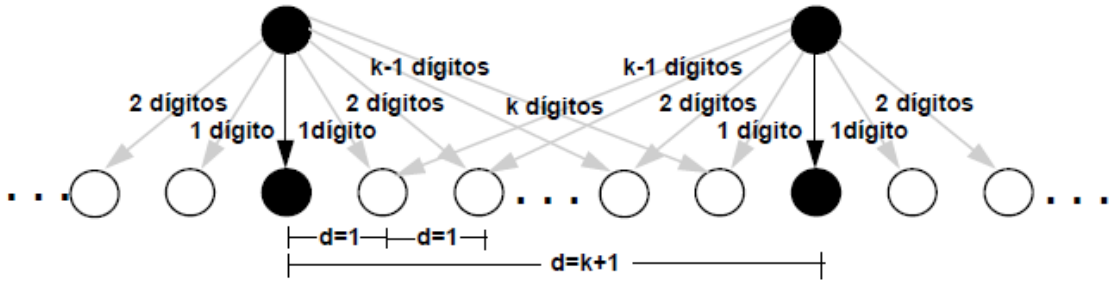
Lo explicitado es necesario pero no suficiente, pues si hubiésemos transmitido el dígito 0011 y recibimos el 0010 no sabríamos si hay error y lo tomaríamos como válido, lo que nos dice que toda combinación por error no debe pertenecer al código



Los círculos sombreados representan combinaciones pertenecientes al código y los blancos combinaciones no pertenecientes al código.
Las flechas negras indican transmisiones libres de error, mientras que las grises señalan las transmisiones con error en un dígito.

Los códigos que cumplen esta condición son aquellos cuya distancia mínima es mayor o igual a 2. Relacionando ambas afirmaciones se puede concluir que la condición necesaria y suficiente para que un código sea detector de error en un dígito binario es que su distancia mínima sea, como mínimo, dos.

La condición necesaria y suficiente para que un código permita detectar el error producido por la variación de un número de dígitos igual o menor que k durante la transmisión de la palabra de código, es que su distancia mínima sea, al menos, “k+1”.



Se puede ver como el código detector de k errores lo será también de un número de errores inferior a k.

b.- Códigos de paridad

Una combinación binaria tiene paridad par, si el número de unos de esa combinación es par y tiene paridad impar si su número de unos es impar.

Ejemplo

Dígito decimal	Código de paridad	
	Código BCD natural	Dígito de paridad
0	0000	0
1	0001	1
2	0010	1
3	0011	0
4	0100	1
5	0101	0
6	0110	0
7	0111	1
8	1000	1
9	1001	0

Existen dos proposiciones

a) Un código de paridad par, posee una distancia mínima estrictamente mayor que uno.

Razonamos por reducción al absurdo. Supóngase que existen dos combinaciones:

bk bk-1 ... b1 b0 y ak ak-1 ... a1 a0, con una distancia entre ellas de uno.

Esto implica que existe un bit, por ejemplo:

el j-ésimo, que hace que: $bi = ai \ \forall \ i \in [0, k] / i \neq j$

$$\overline{b_j} = a_j$$

De aquí se infiere que si la primera combinación posee m 1´s, la segunda tiene m+1 ó m-1. En cualquier caso si m es par m+1 ó m-1 son números impares, con los cual la segunda combinación no pertenecería al código. Este razonamiento lleva a un absurdo, con lo que queda demostrada la proposición.

b) En un código de paridad par, existen siempre dos combinaciones que distan dos unidades.

Como el código de partida es de distancia mínima uno, existirán al menos dos subcombinaciones

bk-1 ... b1 b0 y ak-1 ... a1 a0, en las que solamente un índice j hace que:
$bi = ai \ \forall \ i \in [0, k] / i \neq j$

$$\overline{b_j} = a_j$$

Si la primera combinación posee m 1´s, la segunda tendrá m+1 ó m-1.
Supóngase que m es par. Entonces los respectivos bits de paridad serán bk=0 y ak=1.

Si m fuera impar los bits de paridad serían bk=1 y ak=0.

En cualquier caso se cumple que
$$\overline{b_k} = a_k$$

Por lo tanto, la distancia de estas dos combinaciones resultante es dos

$\overline{b_k} = a_k$ y $\overline{b_j} = a_j$

Supongamos que el criterio de paridad del código usado es par. Entonces si se recibe:

10001 Su paridad es par ⇒ será correcta.
10000 Su paridad es impar ⇒ será incorrecta.
01110 Su paridad es impar ⇒ será incorrecta.
01100 Su paridad es par ⇒ será correcta.

c.- Códigos de peso constante

Se denomina peso de una combinación binaria, al número de unos que posee. Entonces, los códigos de peso constante serán aquellos cuyas combinaciones tienen siempre la misma cantidad de unos.

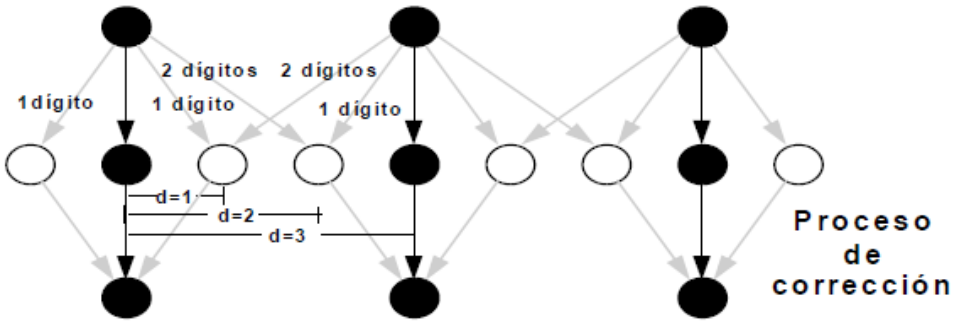
Dígito decimal	Código 2 entre 5	Código biquinario *peso* 5 0 4 3 2 1 0
0	0 1 1 0 0	0 1 0 0 0 0 1
1	1 1 0 0 0	0 1 0 0 0 1 0
2	1 0 1 0 0	0 1 0 0 1 0 0
3	1 0 0 1 0	0 1 0 1 0 0 0
4	0 1 0 1 0	0 1 1 0 0 0 0
5	0 0 1 1 0	1 0 0 0 0 0 1
6	1 0 0 0 1	1 0 0 0 0 1 0
7	0 1 0 0 1	1 0 0 0 1 0 0
8	0 0 1 0 1	1 0 0 1 0 0 0
9	0 0 0 1 1	1 0 1 0 0 0 0

Se puede comprobar que estos códigos poseen una distancia mínima de dos. El error se detecta cuando la combinación recibida tenga un número de unos distinto al peso del código usado.

d.- Códigos de corrección de error

Estos códigos permiten, además de detectar el error, corregir el código recibido sin necesidad de repetir la transmisión. De forma general, se puede decir que la técnica empleada para la corrección consiste en identificar como combinación correcta, a la perteneciente al código que sea más cercana a la errónea recibida, o sea, aquella cuya distancia de la combinación incorrecta sea menor.

Las propiedades de estos códigos que son capaces de corregir el error cometido al variar un dígito binario:



Conviene fijarse como un código corrector de orden 1 sólo corrige correctamente si hay un error en la transmisión. Esta afirmación es extrapolable a códigos correctores de cualquier orden.
En la figura anterior se puede observar esquemáticamente el proceso de corrección. Donde podemos decir que la condición necesaria y suficiente para que un código sea corrector de orden uno, o sea que corrija correctamente errores producidos al variar un dígito en la transmisión, es que su distancia mínima sea tres.

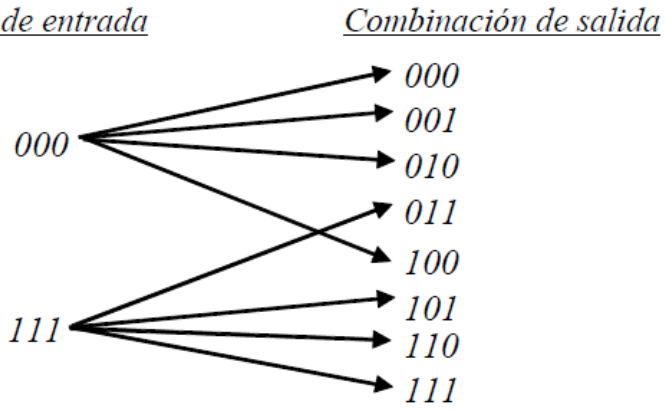
Si se desea transmitir dos mensajes A y B y la distancia mínima del código usado es menos que tres, es imposible una corrección no ambigua:

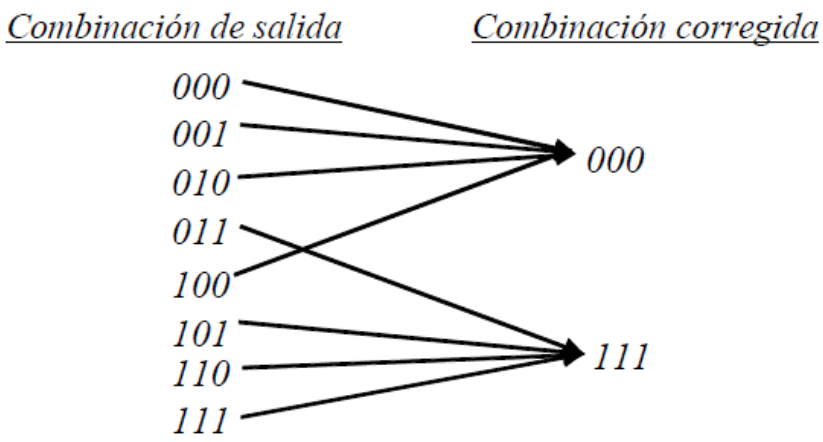
código I: A.....00
B.....11

Supongamos que usamos este código y se recibe la combinación 01, evidentemente ha existido un error, y suponiendo que ha sido en un solo dígito es imposible discernir si la combinación enviada ha sido la 00 o la 11 ya que ambas distan una unidad de la recibida.

Creemos ahora un código de distancia mínima tres:

código II: A.....000
B.....111

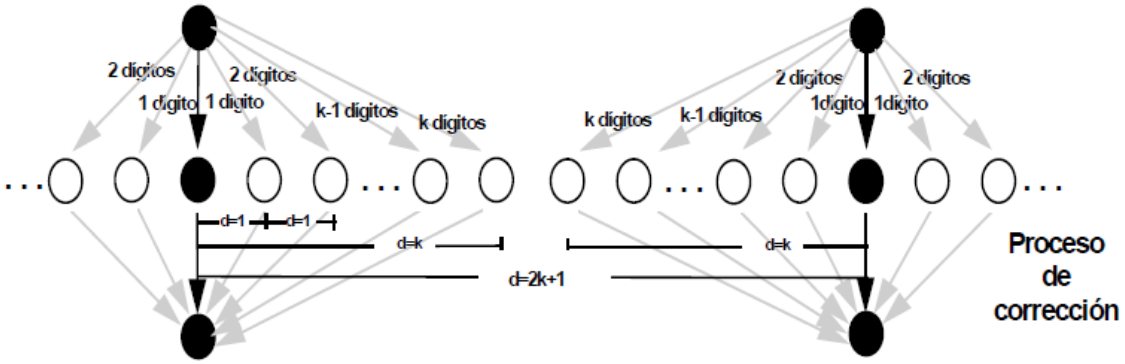




Aunque la restricción es que no existe más de un error en la transmisión de cada combinación, es aplicable en numerosos canales reales, es conveniente generalizar el proceso de corrección a un orden k

Si durante la transmisión de una combinación varían un número de dígitos igual o inferior a k, se recibirá una combinación no perteneciente al código; para que este error pueda ser corregido, se debe cumplir que la combinación emitida sea la única perteneciente al código cuya distancia de la recibida sea igual o inferior a k. Generalizando el razonamiento a cualquier combinación transmitida se puede concluir que la condición necesaria y suficiente para que un código sea corrector de orden k es que su distancia mínima sea de 2k+1; sólo en este caso el proceso de corrección será no ambiguo.

En la siguiente figura se puede observar una esquematización gráfica del proceso de corrección de orden k que puede ayudar a la comprensión de lo expuesto.



e.- Códigos de Hamming

Con este nombre se conoce a un conjunto de códigos correctores en k dígitos binarios, supongamos en principio el de orden uno: k=1.

Para trabajar con este tipo de códigos podemos distinguir dos operaciones:

a) Construcción, que se realizará en el emisor.
b) Interpretación, que se realizará en el receptor.

a)Construcción. Se parte de un código de “n” dígitos de distancia mínima uno. Estos “n” dígitos son conocidos dentro del código de Hamming como "dígitos de datos". A continuación se le añaden p (c_{p-1}, ..., c_2, c_1, c_0) dígitos denominados de control o paridad.

Así pues, el nuevo código tendrá una longitud de palabra de L=n+p.

La numeración de los dígitos es la habitual (de derecha a izquierda) pero comenzando por uno: d_{n+p} d_{n+p-1} ... d_2 d_1.

Cada uno de estos p dígitos que añadimos al código original va a afectar a unas determinadas posiciones de la nueva palabra de código de n+p dígitos, de forma que tomaran el valor adecuado para que se cumpla el criterio de paridad (par o impar) preestablecido en las subcombinaciones afectadas por cada uno.

En la construcción del código los p dígitos añadidos actúan como dígitos de paridad.

b) Interpretación. Recibida una combinación de un código de Hamming hay que comprobar si es correcta, y de no ser así habrá que detectar el dígito que varió en la transmisión.

Ahora los p dígitos añadidos actúan como dígitos de control y con ellos formamos una palabra binaria. Cada uno de los dígitos de esta palabra toma el valor 0 ó 1 dependiendo de si el número de unos de las posiciones de la palabra de código por el afectadas cumplen o no el criterio de paridad establecido. Interpretando la combinación resultante en binario natural, tendremos dos posibilidades:

- Que se corresponda con el 0. Entonces quiere decir que la transmisión ha sido correcta.
- Que se corresponda a un número distinto del 0. Entonces en la transmisión ha variado el dígito situado en la posición indicada por ese número.

Quedan varias cuestiones por resolver:

I.- Cómo calcular p.
II.- A que posiciones afecta cada uno de los p dígitos de control o paridad.
III.- Dónde se colocan estos dígitos dentro de la palabra de código.

I.- Dada la forma de calcular la posición errónea, con p dígitos binarios se tiene que poder detectar el error en todas y cada una de la n+p posiciones de la palabra de código. Como la combinación formada por los p dígitos de control se interpreta en binario natural, se debe cumplir que:

$$2^p - 1 \geq n + p$$

Donde $2^p - 1$ es el mayor número que se puede representar en binario natural con p dígitos.

II.- Construyamos todas las combinaciones posibles con p dígitos de control, e interpretemos cada una en binario natural:

c_{p-1} ... c_2 c_1 c_0	Posición
0 ... 0 0 0	0
0 ... 0 0 1	1
0 ... 0 1 0	2
0 ... 0 1 1	3
0 ... 1 0 0	4
0 ... 1 0 1	5
.....	

Cada dígito de control ha de afectar a aquellas posiciones en las que sea capaz de detectar error, o sea, va a afectar a las posiciones de la tabla anterior para las que ese dígito valga 1

Dígito	Posiciones
C_0	1, 3, 5, 7, 9, 11, 13, ...
C_1	2, 3, 6, 7, 10, 11, 14, 15, ...
C_2	4, 5, 6, 7, 12, 13, 14, 15, ...
C_p	2^p, 2^{p+1}, 2^{p+2}, ...

III.- Han de colocarse en aquellas posiciones en las que no se vean afectados por otro dígito de control, así no existirán ambigüedades a la hora de otorgarles valor en la creación del código. Estas posiciones han de ser entonces:

Dígito	Posición	Combinación en binario natural
C_0	2^0	0 ... 001
C_1	2^1	0 ... 010
C_2	2^2	0 ... 100
...		

Veamos un ejemplo de creación e interpretación de un código de Hamming de paridad par: para ello tomemos el código binario natural de 4 bits.

a) Creación. Dividimos el proceso en una secuencia lógica de pasos:

1.- Cálculo del número de dígitos de control necesarios:

$2^p - 1 \geq n+p$

Con n=4

$\Rightarrow 2^p \geq 4+1+p \Rightarrow 2^p \geq 5+p$ o sea p=3

La palabra de código tendrá, entonces, una longitud L=4+3=7 dígitos: d7d6d5...d1. Para identificar los dígitos de control les denominamos c2, c1 y c0.

2.- Hallamos las posiciones de la palabra de código afectadas por cada dígito de control.

c2	c1	c0	Posición
0	0	1	 1
0	1	0	 2
0	1	1	 3
1	0	0	 4
1	0	1	 5
1	1	0	 6
1	1	1	 7

Los controles de paridad se efectúan sobre las siguientes subcombinaciones:

c0 .- d1, d3, d5, d7.

c1 .- d2, d3, d6, d7.

c2 .- d4, d5, d6, d7.

3.- Cada dígito de control estará situado en las siguientes posiciones de la palabra de código:

co .- d1 c1 .- d2 c2 .- d4

4.- Construcción del código de Hamming:

				c_2		c_1	c_0					c_2		c_1	c_0
	d_7	d_6	d_5	d_4	d_3	d_2	d_1		d_7	d_6	d_5	d_4	d_3	d_2	d_1
0	0	0	0	**0**	0	**0**	**0**	8	1	0	0	**1**	0	**1**	**1**
1	0	0	0	**0**	1	**1**	**1**	9	1	0	0	**1**	1	**0**	**0**
2	0	0	1	**1**	0	**0**	**1**	10	1	0	1	**0**	0	**1**	**0**
3	0	0	1	**1**	1	**1**	**0**	11	1	0	1	**0**	1	**0**	**1**
4	0	1	0	**1**	0	**1**	**0**	12	1	1	0	**0**	0	**0**	**1**
5	0	1	0	**1**	1	**0**	**1**	13	1	1	0	**0**	1	**1**	**0**
6	0	1	1	**0**	0	**1**	**1**	14	1	1	1	**1**	0	**0**	**0**
7	0	1	1	**0**	1	**0**	**0**	15	1	1	1	**1**	1	**1**	**1**

Analicemos todos los casos posibles de error en un bit.

1.- Alteración de un bit de datos:

				C_2		C_1	C_0
	d7	d6	d5	d4	d3	d2	d1
Combinación transmitida	0	1	1	0	1	0	0
Combinación recibida	1	1	1	0	1	0	0

- Control de paridad de c0 → d7 d5 d3 d1 ; 3 unos → impar ⇒ co = 1.
- Control de paridad de c1 → d7 d6 d3 d2 ; 3 unos → impar ⇒ c1 = 1.
- Control de paridad de c2 → d7 d6 d5 d4 ; 3 unos → impar ⇒ c2 = 1.

El bit erróneo es: c2 c1 c0 = 1 1 1 = o sea 7.

Otro ejemplo

				C_2		C_1	C_0
	d7	d6	d5	d4	d3	d2	d1
Combinación transmitida	0	1	1	0	1	0	0
Combinación recibida	0	0	1	0	1	0	0

- Control de paridad de c0 → d7 d5 d3 d1 ; 2 unos → par ⇒ co = 0.
- Control de paridad de c1 → d7 d6 d3 d2 ; 1 uno → impar ⇒ c1 = 1.
- Control de paridad de c2 → d7 d6 d5 d4 ; 1 uno → impar ⇒ c2 = 1.

El bit erróneo es: c2 c1 c0 = 1 1 0 = o sea 6.

2.- Alteración de un bit de control

	d_7	d_6	d_5	d_4	d_3	d_2	d_1
Combinación transmitida	0	1	1	0	1	0	0
Combinación recibida	0	1	1	1	1	0	0

- Control de paridad de c0 → d7 d5 d3 d1 ; 2 unos → par ⇒ co = 0.
- Control de paridad de c1 → d7 d6 d3 d2 ; 2 unos → par ⇒ c1 = 0.
- Control de paridad de c2 → d7 d6 d5 d4 ; 3 unos → impar ⇒ c2 = 1.

El bit erróneo es: c2 c1 c0 = 1 0 0 = 4

3.- No hay alteración:

	d6	d5	d4	d3	d2	d1
d7						
Combinación transmitida	0	1	1	0	1	0	0
Combinación recibida	0	1	1	0	1	0	0

- Control de paridad de c0 → d7 d5 d3 d1 ; 2 unos → par ⇒ co = 0.
- Control de paridad de c1 → d7 d6 d3 d2 ; 2 unos → par ⇒ c1 = 0.
- Control de paridad de c2 → d7 d6 d5 d4 ; 2 unos → par ⇒ c2 = 0.

Como c2 c1 c0 = 0 0 0 = 0, entonces no existe error en la combinación recibida

11.- Código Reed Solomon

a.- Generalidades

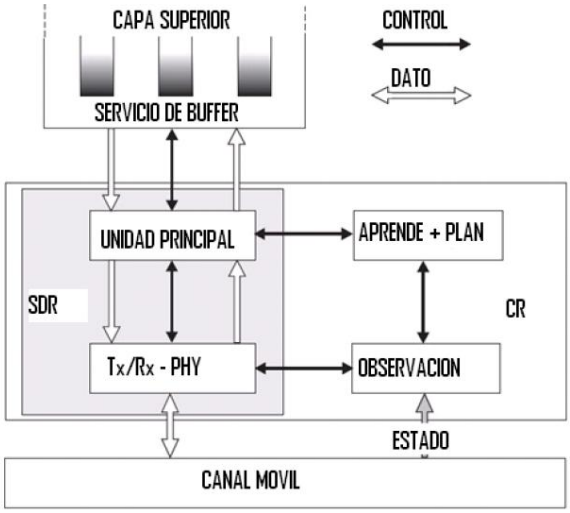
Es fundamental estudiar un código como el Reed Solomon debido a su importancia en la transmisión de televisión digital, (DVB_ Digital Video Broadcast), efectuar un código modificable y reprogramable mediante equipos de radio desarrollados por programa o "Radio Software" o sigla SDR (Software Defined Radio) producen facilidades de su uso y adaptación a las necesidades del usuario.

La metodología de funcionamiento del codificador Reed Solomon, hace cada vez más competitiva y cada vez se adoptan más estrategias a fin de garantizar el éxito de los transmisores y receptores de comunicaciones móviles, lo que ha permitido conocer de manera más detallada su funcionamiento.

Como sabemos los equipos transmisores y receptores de radiocomunicaciones están constituidos por una gran cantidad de componentes electrónicos, los cuales forman circuitos sintonizados, circuitos de frecuencia intermedia, detectores, amplificadores de baja frecuencia, etc.; es decir están constituidos por "hardware", mientras que el Radio definido por Software está conformado por componentes que funcionan por medio de programas en un ordenador, es decir, están constituidos por "Software"

En la época de los 90' se dio paso a la introducción de chips "Procesadores Digitales de Señal" o DSP en los equipos modernos de radio, permitiendo realizar filtros pasabanda y supresores de ruido mediante técnicas digitales siendo éstos dispositivos muy eficaces e incluso hasta mejores que circuitos analógicos.

Tanto los equipos de radio realizados enteramente con componentes electrónicos con los informáticos son del tipo "Radio Hardware" mientras que a partir de la década de los 2000, se comenzó a utilizar equipos de radio aficionados con el concepto "Radio Software" con las siglas SDR (Software Defined Radio) utilizando un mínimo hardware y funciones definidas por software.
Los receptores y transmisores de radio o sea los transceptores, que conforman los SDR son capaces de ser programados y reprogramados con diferentes protocolos y formas de onda a través de la carga dinámica de los mismos.



Estructura General de un Radio Definido por Software

En la estructura general de un radio definido por software, se localizan las funciones que se realizarán mediante la ejecución de programas en un procesador adecuado, los algoritmos del canal móvil, producen señales de radio compatibles con los estándares que son soportados por SDR y los algoritmos en el receptor son desarrollados para recuperar la información enviada desde el transmisor.

De acuerdo al área de operación del SDR,

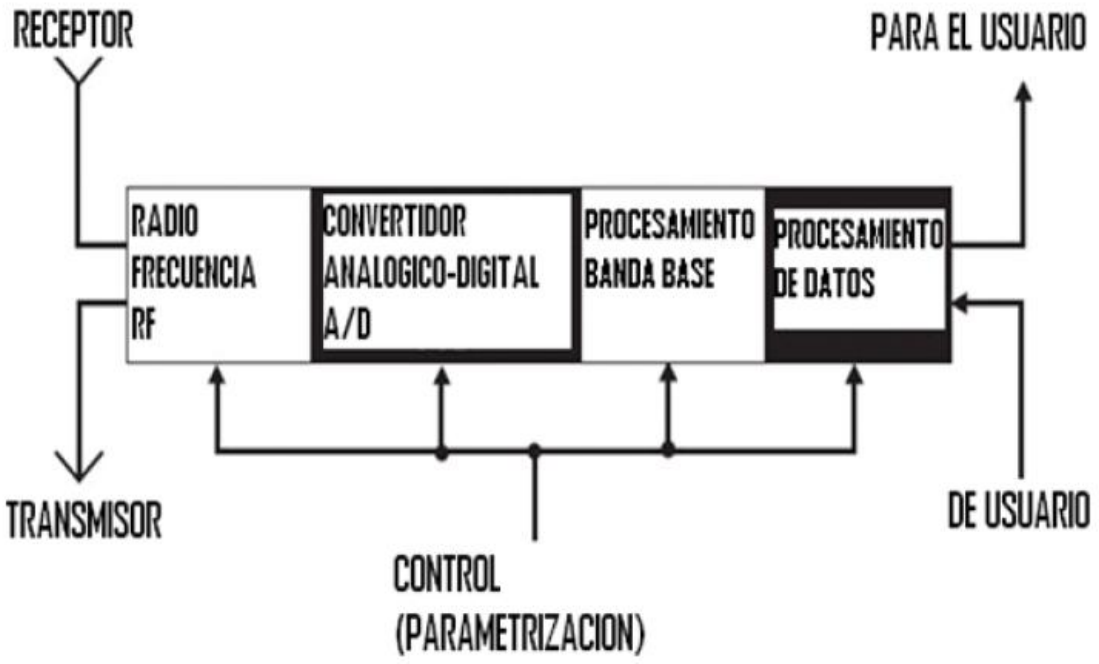
- Un sistema *Multibanda:* soporta más de una banda de frecuencias dedicadas, usado para estándares wireless (como GSM 900, GSM 1800, GSM 1900).

- Un sistema *Multiestándar:* soporta más de una interface aérea. Los sistemas *Multiestándar* pueden trabajar dentro de una familia estándar (como UTRA-FDD, UTRA-TDD para UMTS) a través de diferentes redes (como DECT, GSM, UMTS, WLAN).

- Un sistema *Multiservicio:* provee diferentes servicios (Datos, Telefonía, Video).

- Un sistema *Multicanal:* soporta dos o más transmisiones independientes y recibiendo canales simultáneamente.

El Radio Definido por Software tiene algunas características que lo hacen único en comparación con otros tipos de radios. Una de ellas es que tiene la capacidad de ser transformado a través del uso de software o de lógica re definible, a menudo esto se hace con Procesadores Digitales de Señales (DSPs) o con FPGAs (Field Programmable Gate Arrays) de propósito general.



Esquema de Transceptor Definido por Software

La opción que proporciona la digitalización de señales en la antena, es el uso de un *Analogue Front End (AFE)* flexible, capaz de interpretar un amplio rango de frecuencias y bandas para que con los convertidores de datos puedan procesarlas adecuadamente.

El punto clave es que los SDRs tienen la habilidad de ir más allá de la simple tecnología *single-channel transceiver (Transceptor de Canal Único)* y *single-mode transceiver (Trnsceptor de Modo Único)* con la habilidad para cambiar la función arbitrariamente, ya que el ancho de banda del canal, tasa y modulación son determinados a través de software.

b.- Aplicaciones en DBV _ Digital Video Broadcast

El Digital Video Broadcast es una organización que promueve los estándares aceptados internacionalmente de televisión digital, en especial para HDTV y televisión satelital, así como transmisiones de datos vía satélite. Por lo tanto, permite ejecutar estrategias de inclusión social, y reducir la brecha digital, ofreciendo soluciones adaptables a cada país

Estándares DVB	Codificación de la fuente	Codificador Reed-Solomon	Entrelazado	Codificador Convolucional	Filtrado	Modulación
DVB-S	MPEG-2	(204,188)	12 bytes	½	Nyquist	QPSK
DVB-C	MPEG-2	(204,188)	12 bytes	½	Nyquist	QAM
DVB-T	MPEG-2	(204,188)	12 bytes	½	Nyquist	OFDM

Esquema y características de los diferentes elementos de DVB

Una de las primeras decisiones del DVB fue utilizar el MPEG-2 como estándar de compresión de video y audio, además de definir las técnicas de modulación y métodos de codificación para la corrección de errores, que permitan la transmisión vía satélite, cable y terrestre (DVB_S, DVB_C y DVB_T), a su vez el codificador Reed Solomon adoptado fue el RS(204, 188)

c.- Código BHC

Para poder entender mejor el funcionamiento de un codificador Reed Solomon, veamos el código BHC, dicho código fue inventado en los años 1959/60 por Bose, Chaudhuri y Hocquengem, de ahí su sigla; es un subconjunto de códigos cíclicos, tipo binario y no binario, similar al Reed Solomon, siendo el código más conveniente para errores independientes.

Sus parámetros son:

- Longitud del bloque código:

  $n = 2^m - 1$, donde m>=3.

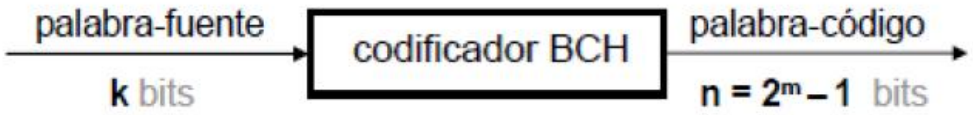
- Dimensión del código:

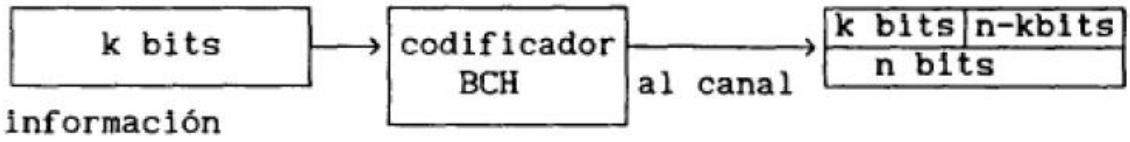
  $k = n$ – *grado del polinomio generador (gx)*; el cual depende del número de errores que pueda corregir el código *t*.

La siguiente tabla muestra la familia de códigos BCH de longitud 15

n	k	t
15	11	1
15	7	2
15	5	3

La inserción de estos códigos es la siguiente, reciben un paquete de información de longitud “k” y se procesa convirtiéndolo en uno de longitud “n” donde n>k,





Para tamaños de unos pocos cientos de bits o menos, los códigos *BCH* son de los mejores para un mismo tamaño de bloque e índice de código (relación entre el bloque de entrada y el bloque de salida). Algunos códigos comunes expresados en la forma (*n, k, t*) de *BCH* son: (7, 4, 1), (15, 11, 1), (15, 7, 2), (15, 5, 3), (31, 26, 1), (31, 21, 2), (31, 16, 3), (31, 11, 5) y (31,6, 7)

d.- Códigos Cíclicos

Los códigos cíclicos son una subclase de los códigos de bloque lineales, los cuales tienen esquemas de decodificación eficientes, es decir con algoritmos relativamente simples. Se dice que un código es cíclico cuando cualquier desplazamiento en lazo cerrado de una palabra-código da como resultado otra palabra-código existente dentro del conjunto empleado para codificar los posibles mensajes.

Existen una gran variedad de códigos cíclicos. Por ejemplo, el Código de Redundancia Cíclica empleado en comunicaciones de datos y el código Golay que es un código binario como el Hamming. Además, están los códigos como el Bose-Chaudhuri-Hocquenqhem y el Reed Solomon. Dada la versatilidad de parámetros de estos dos últimos son los que se seleccionaron para ser analizados en esta sección y en las siguientes.

d.1.- Polinomio de palabra código

La representación matemática de la operación de los códigos cíclicos está basada en el uso de polinomios. Los elementos de una palabra-código de tamaño *n* pueden ser los coeficientes de un polinomio de grado *n-1*. Por ejemplo, la palabra-código con elementos $x_0, x_1, \ldots, x_{n-1}$ puede ser representada en forma de polinomio como:

$$X(D) = x_0 + x_1 D, \ldots, + x_{n-1} D^{n-1}$$

D: es una variable real arbitraria

d.2.- Polinomio generador

Un código cíclico (*n, k*) es especificado por un conjunto de polinomios de palabra-código de grado *n*-1 o menos, el cual contiene un polinomio de grado mínimo *n-k* como un factor. Este factor especial, denotado por *g(D)* es seleccionado como el Polinomio Generador del código.

d.3.- Polinomio para codificación de un código cíclico

Multiplicar el polinomio del mensaje *m(D)* por D^{n-k}

$$D^{n-k}\, m(D) = m_0 D^{n-k} + m_1 D^{n-k+1} + \ldots + m_{k-1} D^{n-1}$$

Dividir $D^{n-k}\, m(D)$ por el polinomio generador *g(D)*, obteniendo el residuo *b(D)*.

$$\frac{D^{n-k} m(D)}{g(D)} = a(D) + \frac{b(D)}{g(D)}$$

Agregar *b(D)* a $D^{n-k}\, m(D)$ para obtener el polinomio de la palabra-código *x(D)*.

$$x(D) = b(D) + D^{n-k}\, m(D)$$

d.4.- Cálculo de síndrome de Detección de Errores

Considerando que la palabra-código recibida con error sea:

$$Y(D) = y_0 + y_1 D, \ldots, + y_{n-1} D^{n-1}$$

El polinomio del síndrome se obtiene con el residuo de la división del polinomio de la palabra código entre el polinomio generador $g(D)$.

$$\frac{Y(D)}{g(D)} = q(D) + \frac{s(D)}{g(D)}$$

Dónde: q es el cociente y s es el síndrome.

d.5.- Detección y Corrección de Errores

En comunicaciones prácticamente todas las señales digitales producidas en la actualidad llevan asociados el proceso de detección o corrección de errores. Este proceso se ocupa de la detección mediante los métodos CRC y BIP y corrección de errores mediante FEC a bloques y convolucional.

Para detectar que hubo un error, al enviarse un marco se guarda en una tabla cuándo se envió y se le asocia un tiempo para recibir su confirmación. Si no se recibe la confirmación por parte del receptor, se re-envía el marco. El problema que puede surgir es que si se perdió la confirmación, el receptor puede tener marcos duplicados, lo cual se soluciona al asignar un número de secuencia a cada marco, para descartar los duplicados y re-enviar su confirmación.

Otra forma de detectar un error (que ya no fue la pérdida del marco, sino la corrupción de su contenido), es insertar un código de chequeo, y para esta labor se utilizan códigos basados en el concepto de "distancia de Hamming".

La distancia de Hamming para un código cualquiera se define como el número de bits diferentes al hacer un XOR entre todos sus símbolos.

Si los símbolos de un código difieren al menos en 2X+1 bits, al variar X bits (dañar X bits) obtengo un nuevo símbolo que se parecerá más en un bit a un código válido que a otro código válido y por lo tanto puede decir que el símbolo dañado en realidad es el más parecido realizando así su corrección.

Para el diseño estándar de protocolos, se han especificado algunas cadenas de chequeo, bien conocidas como CRC-12, CRC-16 y CRC-CCITT con CRC=12,16 bits y CCITT=16 bits respectivamente. Estas cadenas se interpretan con polinomios de la siguiente manera.

$$\text{CRC-12} = 1100000001111 = X^{12} + X^{11} + X^3 + X^2 + X + 1.$$

$$\text{CRC-16} = 11000000000000101 = X^{16} + X^{15} + X^2 + 1$$

$$\text{CRC-CCITT} = 10001000000100001 = X^{16} + X^{12} + X^5 + 1$$

Se observa que los bits con “1” representan la potencia del polinomio, cada polinomio se conoce con el nombre de “genenerador polinomial “ y las CRC (código de redundancia cíclica)

e.- FEC (Forward Error Correction_Corrección de errores)

A menudo en las listas de canales se ven números como en este ejemplo:

- TVE 27,500 7/8.
- Canal Tal 11,500 3/4.

Los números al final después del SR en forma de quebrados son el FEC y casi todos los receptores del mercado lo detectan de forma automática, pero no era así en los primeros receptores de los 90 cuando comenzábamos a trabajar en esta industria.

FEC es un término llamado **Forward Error Correction** que es aplicable solo a transmisiones digitales y que es una "Repetición" de ese dato para asegurar que la misma

llegue sin perdidas al receptor o usuario sin que pierda la señal ni su calidad. Para hacerlo más sencillo, es una transmisión "Repetitiva".

Ese FEC indica cuantos bytes se usan para una señal y cuantas correcciones de errores se usan en la misma. Por ejemplo, un FEC de 1/2 significa que 1 byte de cada 2, se usa para control de errores y corregir esos errores; cuando un FEC de 7/8 por ejemplo, significa que 7 de cada 8 se usan para corregir esos errores.

En el mundo de la transmisión digital, un FEC de 1/2 da la posibilidad de una transmisión casi perfecta y sin fallas de recepción porque cada byte de la señal, es controlado por otro byte que la corrige. Pero cuando un proveedor usa 7/8 de FEC por ejemplo, significa que no pierde ancho de banda contra el costo de entregar la señal al recipiente o receptor.

Aclaración: cuando más bajo el nivel de FEC, mejor receptor se necesita

f.- Definición del codificador

Un codificador es un circuito combinacional con dos veces más entradas que salidas, cuya misión es presentar en la salida el código binario correspondiente a la entrada activada.

Existen dos tipos fundamentales de codificadores:

- Los primeros solo admiten una entrada activada, codificando en la salida el valor binario de la misma y cero cuando no existe ninguna activa.
- En los segundos puede haber más de una entrada activada, existiendo prioridad en aquella cuyo valor decimal es más alto

Los códigos Reed Solomon, manejan algoritmos que pueden ser implementados en software o hardware

En vista de la creciente tendencia hacia el uso de dispositivos de lógica reconfigurable a alta escala de integración y de los beneficios que esta tecnología ofrece a los diseñadores de sistemas digitales, mediante el empleo de un lenguaje de descripción de hardware como VHDL, que permite configurar sistemas digitales según las especificaciones demandadas por los usuarios, ajustar cambios en la programación y optimizar los diseños tratándolos en forma modular, se plantea el diseño de estos módulos de codificación bajo esta tecnología.

h.- Bases del código Reed Solomon

En el estudio de los codificadores de canal, el Reed Solomon, permite observar que su probabilidad frente al error por ruido está cercana al límite de Shannon y presenta mayor eficiencia sobre otros códigos correctores de error, su implementación tecnológica se debió a un algoritmo estudiado por Berlekamp

Modelo de Shannon para la corrección de errores

El código Reed-Solomon es un código corrector de errores basado en bloques en donde el codificador procesa un bloque de símbolos de datos, a los que agrega redundancia para producir un bloque de símbolos codificados.

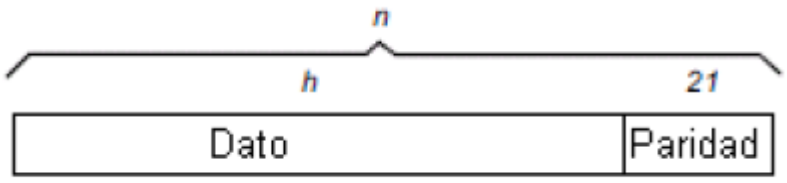
En la actualidad, los códigos Reed-Solomon se utilizan para corregir errores en varios sistemas incluyendo los dispositivos de almacenamiento –cintas, discos compactos, DVD, códigos de barras, etc.–, comunicaciones inalámbricas o móviles –telefonía celular, enlaces de microondas, etc.–, comunicaciones satelitales, televisión Digital/ DVB, módem de alta velocidad como ADSL, x DSL.

i.- Propiedades del código Reed Solomon

El código Reed-Solomon es un subconjunto de los códigos BCH (Bose Chaudhuri Hocquenqhem), códigos cíclicos que presentan entre sus parámetros (*n,k,t*) una relación entre los símbolos de datos (*k*), del código total (*n*) y del número máximo de errores por ser corregidos (*t*), y son de bloques lineales. Un código Reed-Solomon se especifica como RS(n,k) con símbolos de s bits. Lo anterior significa que el codificador toma k símbolos de los s bits y añade símbolos de paridad para hacer una palabra de código de n símbolos.

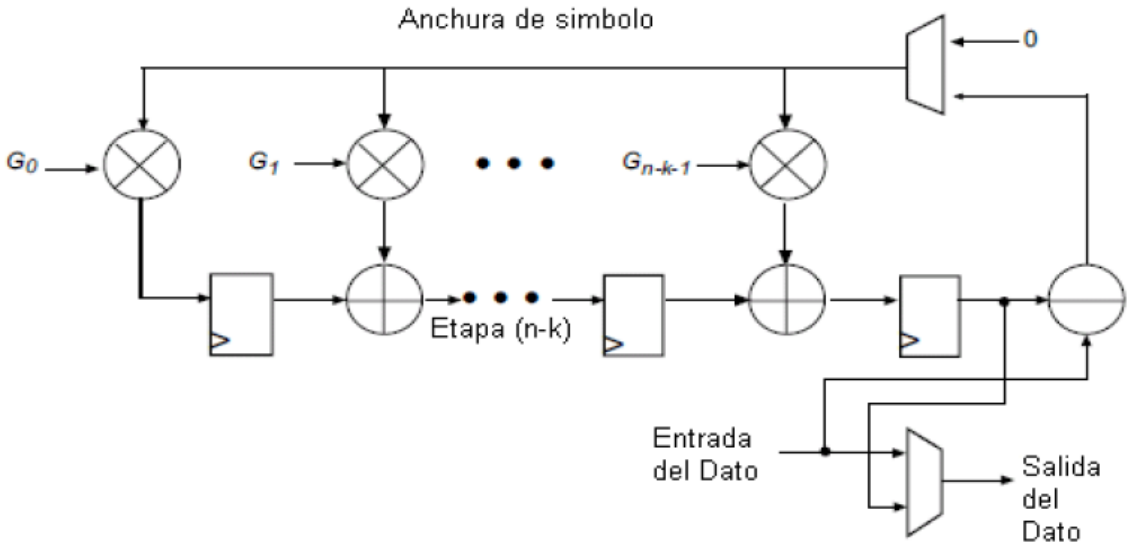
Existen n-k símbolos de paridad de s bits cada uno. Un decodificador puede corregir hasta t símbolos que contienen errores en una palabra de código, donde 2t = (n-k).

La siguiente muestra de código se conoce como un código sistemático puesto que los datos se dejan inalterados y los símbolos de paridad se anexan



Palabra de código Reed Solomon

Para codificar una trama con esta estructura se debe procesar a través de un circuito digital que opere bajo los fundamentos de campo finitos de Galois



Arquitectura genérica de un codificador Reed Solomon

j.- Campos de galois aplicados a la codificación Reed Solomon

Los códigos Reed-Solomon se basan en un área especializada de la matemática llamada campos de Galois o campos finitos. Un campo finito tiene la propiedad de que las operaciones aritméticas sobre elementos del campo siempre tienen un resultado en el campo. Un codificador o decodificador Reed-Solomon debe ser capaz de realizar estas operaciones aritméticas.

k.- Generador polinomial de Campos de Galois

Una palabra de código Reed-Solomon es generada usando un polinomio especial. Todas las palabras de código válidas son divisibles exactamente por el polinomio generador representado por la siguiente ecuación.

$$g(x) = (x - \alpha^i)(x - \alpha^{i+1})(x - \alpha^{i+2t})$$

La palabra de código se genera de c(x)=g(x)*i(x), donde g(x) es el polinomio generador, i(x) es el bloque de información, c(x) es una palabra de código válida y alfa se conoce como un elemento primitivo del campo.

El primer paso corresponde a la definición del campo de Galois para la codificación, el cual estará definido en función de la longitud del símbolo - entiéndase m, bits/símbolo -, permitiendo así conocer el polinomio irreducible del campo GF(2m), tal que para m=3 bits, corresponde a p(x)= x3+x+1, 1 de manera que las operaciones que produzcan resultados que se rebasan, serán reajustados por el polinomio irreducible, como lo es el caso α3 = α + 1, dando así un elemento perteneciente al campo.

Las bases teóricas que sustentan este codificador están dadas por el polinomio en su forma general.

$$g(x) = \prod_{i=0}^{n-k-1} \left(x - \alpha^{hx(Comienza\ generador+i)}\right)$$

Al expandir el polinomio se obtiene la ecuación siguiente.

$$g(x) = G_{n-k-1}x^{n-k-1} + G_{n-k-2}x^{n-k-2} + \cdots + G_1 x + G_0$$

Dónde:

N: longitud de la palabra codificada (en símbolos).

K: longitud del mensaje codificado (en símbolos).

M: longitud del símbolo (bits) [15].

El segundo paso corresponde a definir el polinomio generador donde se obtiene:

$$G(x) = \alpha^2x^3 + \alpha^5x^2 + \alpha^5x + \alpha6$$

Multiplicación en el campo finito de Galois para el polinomio generador $P(x)=x^3+x+1$

		000	001	010	011	100	101	110	111
	x	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
000	0	0	0	0	0	0	0	0	0
001	1	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
010	x	0	x	x^2	x^2+x	$x+1$	1	x^2+x+1	x^2+1
011	$x+1$	0	$x+1$	x^2+x	x^2+1	x^2+x+1	x^2	1	x
100	x^2	0	x^2	$x+1$	x^2+x+1	x^2+x	x	x^2+1	1
101	x^2+1	0	x^2+1	1	x^2	x	x^2+x+1	$x+1$	x^2+x
110	x^2+x	0	x^2+x	x^2+x+1	1	x^2+1	$x+1$	x	x^2
111	x^2+x+1	0	x^2+x+1	x^2+1	x	1	x^2+x	x^2	$x+1$

Esto se utiliza para generar el campo de Galois para el código. Entra como un número decimal donde los bits del archivo binario equivalente corresponden a los coeficientes del polinomio. Por ejemplo:

x8 + x4 + x3 + x2 +1 => 100011101 => 285

Un valor de cero causa al polinomio por defecto el Ancho del Símbolo dado para ser seleccionado. Si el campo polinomial no es primitivo, la siguiente tabla muestra el campo polinomial por defecto.

Ancho de Símbolo	Polinomio	Representación del arreglo binario	Rango de bits a codificar (k)	Rango de bits codificados (n)
3	x^3+x+1	[1011]	2-5	4-7
4	x^4+x+1	[10011]	2-13	4-15
5	x^5+x^2+1	[100101]	2-29	4-31
6	x^6+x+1	[1000011]	2-61	4-63
7	x^7+x^3+1	[10001001]	2-125	4-127
8	$x^8+x^4+x^3+x^2+1$	[100011101]	2-253	4-255
9	x^9+x^4+1	[1000010001]	2-509	4-511
10	$x^{10}+x^3+1$	[10000001001]	2-1021	4-1023
11	$x^{11}+x^2+1$	[100000000101]	2-2045	4-2047
12	$x^{12}+x^6+x^4+x+1$	[1000001010011]	2-4093	4-4095

Polinomios y rango de bits

I.- Diseño de un codificador Reed Solomon

Para simular el diseño del codificador de Reed-Solomon en el ISE 10.1 de Xilinx se necesita System Generator, el cual proporcionará todas las herramientas para la simulación del mismo.

Los bloques a utilizan son los siguientes:

Bloque Generador Binario de Bernoulli: El bloque Generador binario de Bernoulli genera números binarios aleatorios usando una distribución de Bernoulli.

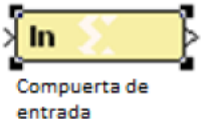
La distribución de Bernoulli con parámetro P produce 0 (cero) con probabilidad p y 1 (uno) con una probabilidad de 1-p. La distribución de Bernoulli tiene valor medio 1-P y varianza P(1-p). La probabilidad de un parámetro cero especifica p, y puede ser cualquier número real entre cero y uno.

Los atributos de la señal de salida pueden ser un cuadro basado en una matriz, una muestra basada en filas o columnas, o una muestra que se basa en matriz unidimensional. Estos atributos son controlados por el marco base de resultados, las muestras por cuadro, y el vector de interpretación de parámetros 1-D.

Generador Binario de Bernoulli

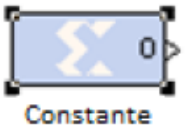
Una vez teniendo el generador del mensaje se requiere una compuerta de entrada para introducir nuestros datos al codificador Reed-Solomon.

Compuerta de entrada: compuerta de entrada de Xilinx; son las entradas para los bloques de Xilinx que es parte del diseño de Simulink. Este bloque convierte los tipos de datos a enteros, dobles o de punto fijo dentro del System Generator. Cada bloque define un nivel superior de entrada introducido en el diseño generado por System Generator.



Estos dos elementos están conectados en serie hacia el codificador Reed-Solomon en el pin de entrada DATA_IN. Además de esta entrada, se requiere obligatoriamente, las entradas del bypass y start.

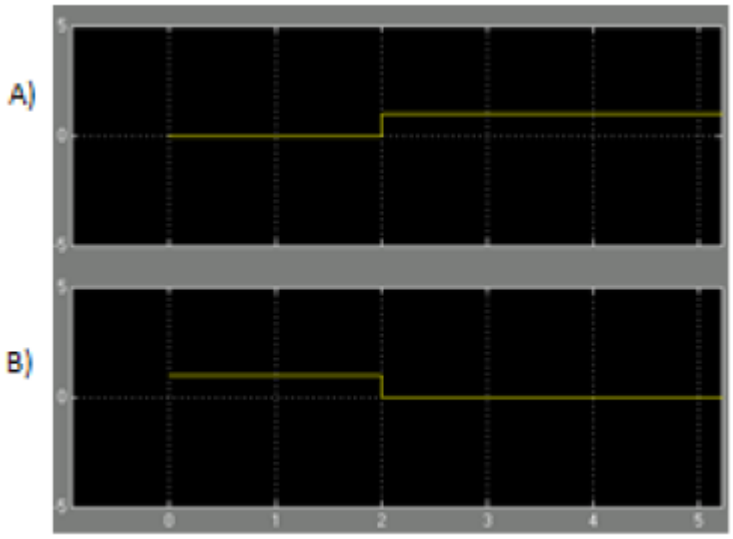
En el Bypass se conecta una constante de 0 (cero), ya que no se pretende que el mensaje pase sin ser afectado. Esta constante puede ser tomada simplemente del simulador de Matlab pero seria necesario conectar otra compuerta; para su simplificación tomamos una constante de Xilinx para conectar directamente con el codificador



Constante 0 (cero) de Xilinx

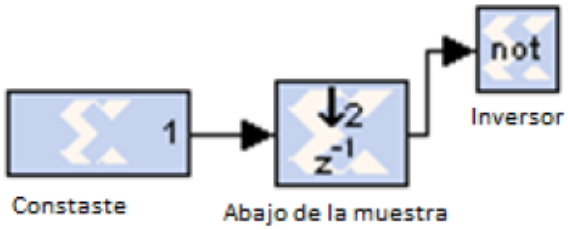
El Xilinx es una herramienta que puede ser instalada en el Matlab (se utiliza para programar VHDL y utilizarlos en FPGA). En el caso de pin de entrada Start, se conecta un tren de pulsos para su activación y sincronización con el mensaje

Se puede generar este pulso con una conexión en serie de una constante en alto, un muestreo en bajo y un inversor. La función del muestreo en bajo es que retarde la constante, en un rango de 0 a 2 sobre el eje de la frecuencia; aun no se genera el pulso pero si se le aplica un inversor tendremos lo siguiente.



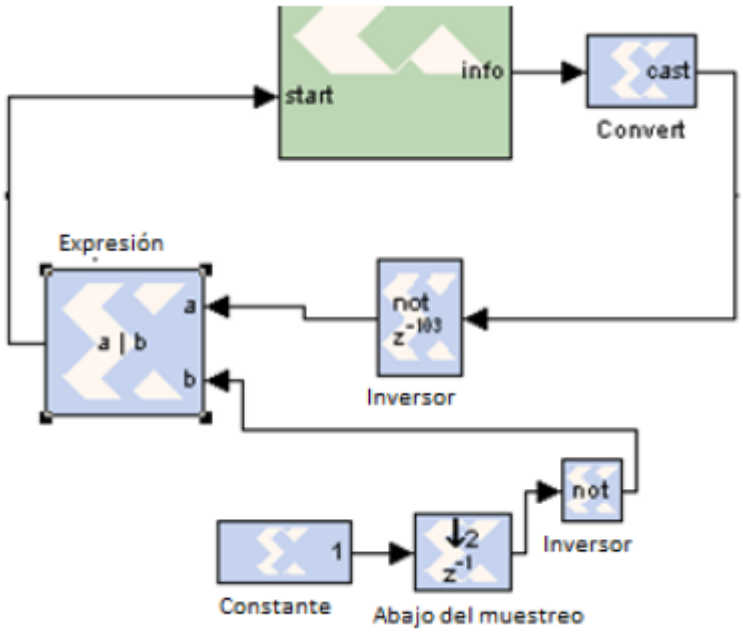
A) Constante con muestreo en nivel bajo
B) Pulso realizado por la constante

Estas conexiones quedan de la siguiente forma en el System Generator:



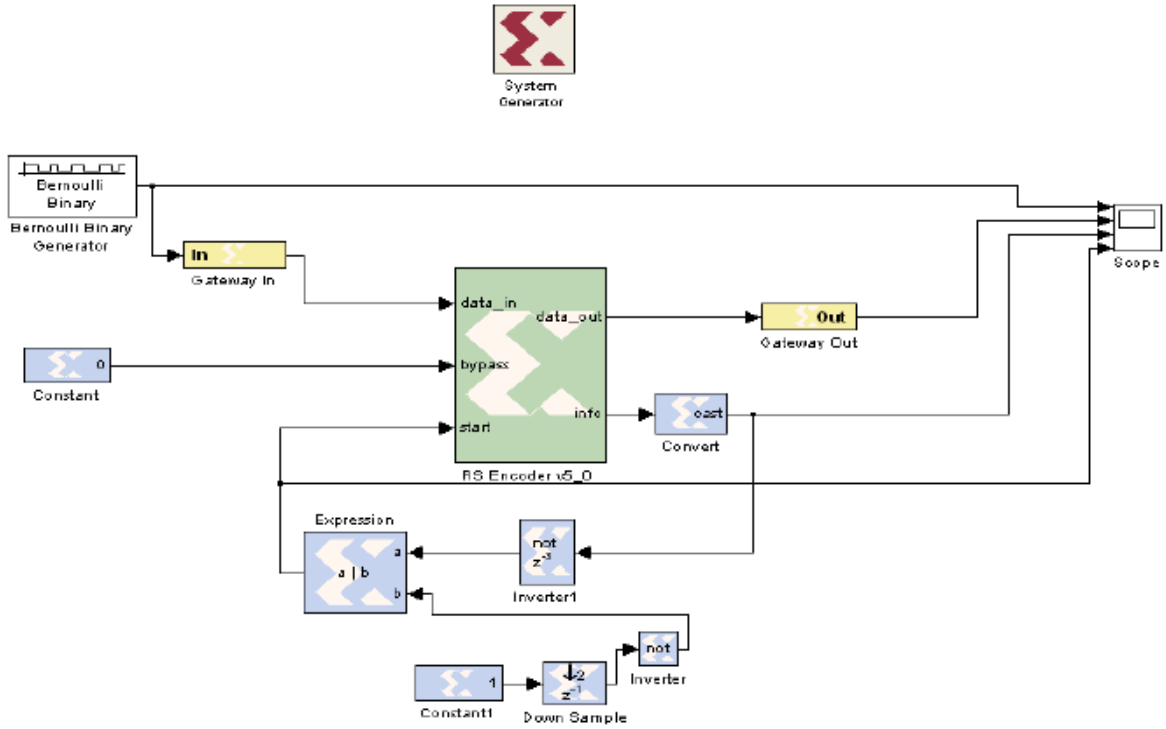
El tren de pulsos se realiza con una retroalimentación del pin de salida INFO del codificador Reed-Solomon; pero debido a su funcionamiento no se puede hacer una conexión directa, ya que debe de llevar un convertidor de tipo de variable booleano (falso, verdadero) para así conectar un inversor que genere los pulsos.

Estos pulsos controlaran toda la entrada del mensaje para poder crear los bloques codificados. El pulso de inicio y el tren de pulsos se unen con una expresión lógica OR bit a bit, resultando de la siguiente manera:

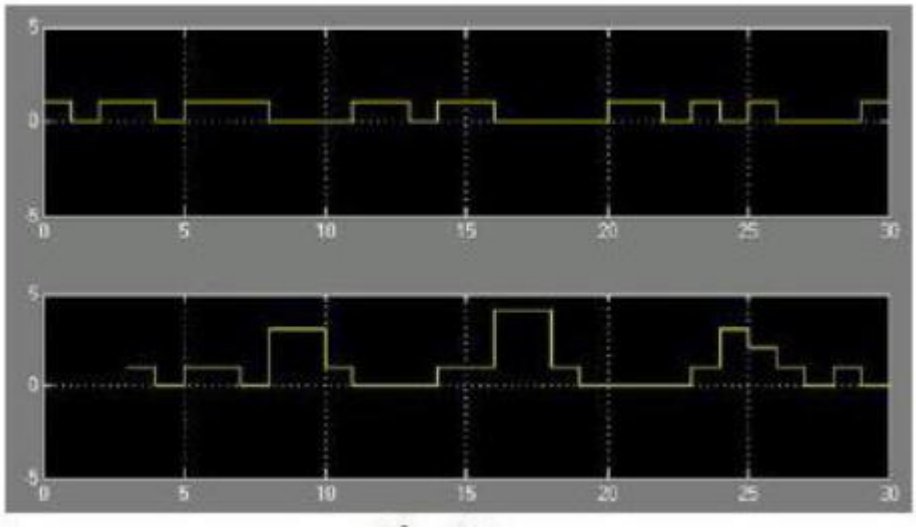


Generador de tren de Pulsos en la entrada Start

La latencia (retraso) en el inversor se ejecuta para tomar la medida de bloques de símbolos, así mismo llevar el control de los bloques en la salida (DATA_OUT).



Bloques de un sistema generador del codificador Reed Solomon



Resultado de la simulación de un código RS (7,3)

12.-Espectro Expandido y su aplicación en CDMA

a.- Introducción

La multiplexación por desviación de código (CDMA), permite la transmisión simultánea de diferentes fuentes de información utilizando “todas”, la misma portadora y compartiendo el mismo ancho de banda, que es bastante mayor que el de cada banda base. Cada una de esas fuentes de información pueden diferenciarse en la recepción pues cada una de ellas tienen "impreso" un código particular.

CDMA se define como una forma de multiplexación de señales basada en el uso del llamado espectro ensanchado o esparcido (en inglés spread spectrum, SS)

“SS” es el medio de transmisión, en el cual la señal ocupa un ancho de banda (BW) superior al mínimo necesario para enviar la información; la banda base es ensanchada por medio de un código independiente de los datos, este es conocido por el receptor, quien correlaciona la señal recibida con el código en correcta fase para recuperar los datos enviados.

Analizando cada punto de la definición tenemos:

a) La señal ocupa un ancho de banda (BW) superior al mínimo necesario para transmitir la
información. Se define un coeficiente de expansión a la relación entre el ancho de banda expandido utilizado para la transmisión y el ancho de banda de la señal a transmitir:

$$Ce = Be / f \max$$

Este coeficiente es mucho mayor que 1.

La ampliación del ancho de banda trae aparejado otra ventaja, según el principio enunciado por Shannon, en la Teoría de la Información. En un canal ruidoso se pude intercambiar ancho de banda por potencia de señal, manteniendo constante la capacidad del canal.

$$C = B\log_2(1+\frac{S}{N})$$

Donde:

“C” es la capacidad del canal ruidoso.

“B” es el ancho de banda de la señal en el canal

“S/N” es la relación señal a ruido en la recepción.

De esta forma se puede regular la potencia de los emisores para que no se interfieran en forma significativa, disminuyendo la potencia y ampliando el ancho de banda en una relación acorde para mantener constante la capacidad del canal. La disminución de la potencia y el incremento del ancho de banda hace que la transmisión tenga una densidad espectral de potencia baja, que se confunde con el ruido del canal, ya que la relación señal a ruido tiene a menudo una relación señal a ruido en cualquier intervalo de frecuencias de la transmisión menor que la unidad.

b) La banda base es ensanchada por medio de una señal, código independiente de los datos, cuya composición varía en forma pseudoaleatoria. Debido a que la señal así obtenida es similar al ruido para las que no trabajan con el mismo código se la conoce también con su denominación en inglés de pseudo noise “PN”

.

Otras técnicas de modulación, como la “FM” y el “PCM” también ensanchan en forma importante el ancho de banda, respecto de la señal de banda base, sin embargo no alcanzan todas las ventajas del CDMA debido a que dicha expansión no ha sido realizada mediante la multiplicación con una señal código. Esta manera de expansión es fundamental en el CDMA.

c). En el receptor la señal de datos original es recuperada "correlacionando" la señal recibida con una réplica sincronizada y en fase de la señal de código utilizada para el ensanchamiento de la banda base.

Esta tecnología se ha utilizado inicialmente para propósitos militares. Al final de la Segunda Guerra Mundial, ya que se había avanzado en el estudio de esta técnica y eran conocidas sus ventajas con respecto a la capacidad para rechazar interferencias, ya sean intencionales o no. Durante los años subsiguientes se efectuaron estudios sobre posibles aplicaciones destacándose algunas como:

1. Supresión de interferencia "antijamming" o antibloqueo, ya que la posibilidad de interceptación se reduce a una pequeña parte de la información transmitida, que se encuentra en una banda muy superior a la de la banda base.

2. Bajos niveles de potencia de transmisión, que unido a una banda ensanchada da una densidad espectral de potencia baja, una relación señal ruido baja, lo que aumenta la privacidad.
3. Poder compartir la misma frecuencia de portadora con otros usuarios, sin interferirse unos con otros, conformando sistemas de acceso múltiple ("CDMA: Code Division Multiple Access)
4. Alta resolución para determinar posicionamiento.

Las aplicaciones cubren una amplia variedad de servicios, entre los cuales podemos mencionar:

Telefonía celular
Comunicaciones satelitales
Sistemas inalámbricos para extensión de redes LAN
Comunicaciones punto a punto para transmisión de voz, datos, fax. etc.

Las formas más difundidas de transmisión por **"SS"** son las denominadas:

1. Secuencia Directa o en inglés Direct Secuence Spread Spectrum (DSSS)
2. Salto en Frecuencia o Frecuency Hopping Spread Spectrum (FHSS)

b.- Transmisión en Secuencia Directa (DSSS)

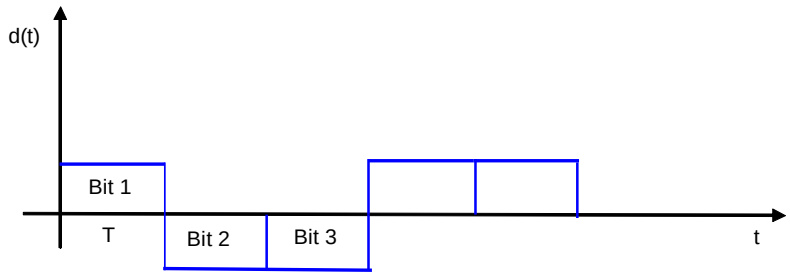
Podemos destacar dos características:

1) "La señal es ensanchada por un código independiente de los datos".

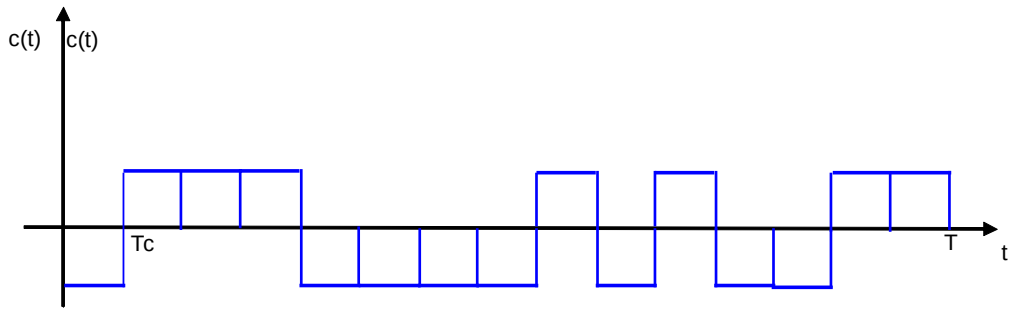
Esto se realiza multiplicando cada "bit" de la banda base por un código pseudoaleatorio, cuyos dígitos para diferenciarlos de los datos, se los llama chips. Si se usa un código de 15 chips, como el que se ejemplifica abajo, cada bit del mensaje será representado en la señal **"SS"** por 15 chips, incrementándose el ancho de banda 15 veces. Cuanto mayor sea la cantidad de chips, mayor será la protección contra posibles ataques ala privacidad de la comunicación.

1	0	0	0	1	1	1	1	0	1	0	1	1	0	0

Si la señal de datos fuese



Donde T es la duración de cada bit y la señal del código pseudoaleatorio es c(t)



La duración de cada chip es Tc, siendo Tc = T/N.

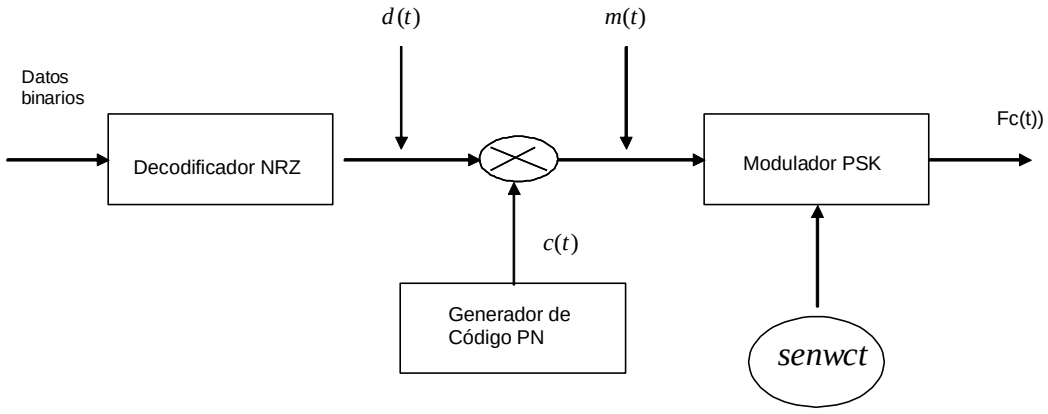
Donde N es longitud del código en chips, que en este caso particular es 15, y T es la duración del código, que debe coincidir con la de cada bit.

Si multiplicamos d(t) por c(t), veríamos el ensanche de cada bit, que contendrá 15 chips y esto incrementará el ancho de banda

2) El código es independiente de los datos y además debe contar con propiedades de aleatoriedad para disminuir la interferencia entre ellos y mantener la comunicación con un alto nivel de privacidad.

Sin embargo para recibir el mensaje se necesita un alto nivel de sincronización del receptor con el transmisor, donde en el receptor se debe conocer la réplica del código utilizado.

El diagrama en bloques de un sistema de espectro ensanchado modulado por secuencia directa (DSSS) es el siguiente:



La entrada binaria que tiene una tasa de información R = 1/T bits por segundo es multiplicada por los chips pseudoaleatorios en un circuito del tipo OR exclusiva dando lugar a una señal cuya expresión temporal es:

$$m(t) = c(t).d(t)$$

Posteriormente esta señal es modulada en un modulador BPSK dando lugar a la señal

$$fc(t) = Ac(t)d(t)sen(wct + \vartheta(t))$$

Donde $0 = \vartheta(t)...ó..\vartheta(t) = .\pi$ dependiendo de la polaridad que tenga el producto de la señal binaria por la pseudoaleatoria en cada instante t.

c.- Detección de la señal transmitida en secuencia directa

La señal de datos para el usuario es d(t), cuyos pulsos rectangulares de amplitud ± 1, tienen una duración T. El usuario tiene asignado una señal de código Ci (t) la cual consiste de una secuencia periódica de pulsos de duración Tc y amplitud ± 1, que es una réplica de la transmitida.

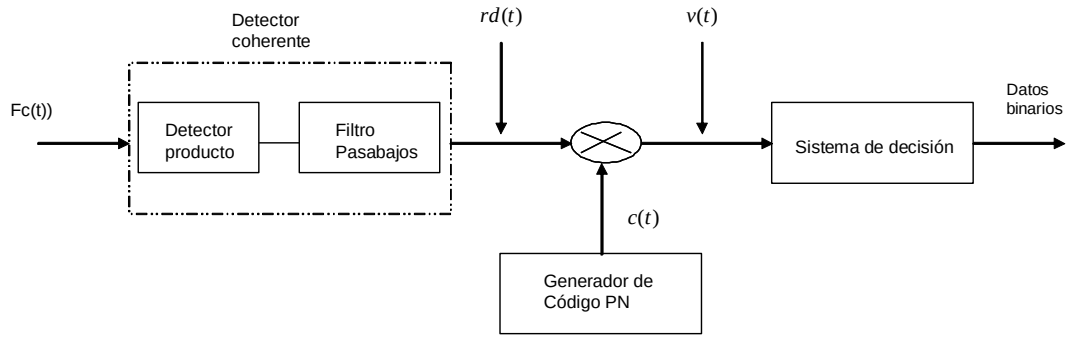
En el receptor se recibirá:

$$r(t) = Ac(t)d(t)sen(wct + \vartheta(t)) + n(t) + \sum_{j=1} Acj(t - tj)dj(t - tj)sen(wct + \phi j) + i(t)$$

Donde n(t) es el ruido térmico, i (t) es una posible interferencia y los componentes j son transmisiones en la misma frecuencia fc , al mismo tiempo pero con distinto código.

Como se observa se comparte el medio temporal y espectral, la división entre comunicaciones es el código y este interfiere al resto de acuerdo a su característica de correlación. Cuanto más distintos son los códigos entre sí, y más largos, menos se interfieren, menos ruido se producen, más señales pueden compartir el medio dado una cierta relación señal a ruido propia del receptor.

Un diagrama en bloques puede ser:



En el receptor se realiza la detección de la señal BPSK, que da lugar a la siguiente señal

$$rd(t) = c(t)d(t) + n(t) + \sum_{j=1} Aci(t - t1)dj(t - t1)sen(wct + \phi 1) + i(t)$$

y la correlación discreta con el código convenido, dando lugar a una señal en la que solo se considera a la señal interferente, pues se supone más importante que el ruido del canal.

$$v(t) = c(t)rd(t)$$
$$v(t) = c^2(t)d(t) + c(t)i(t) + c(t)\sum_{j=1} Acj(t - tj)dj(t - tj)sen(wct + \phi j)$$

Como se vió c(t) tiene valores que oscilan entre +1 y -1. Esta oscilación es suprimida al efectuar la correlación digital consigo mismo, (básicamente en otro circuito OR exclusivo), es decir $c^2(t) = 1$ para todo t.

Por el contrario los otros términos de la salida son muy pequeños, pues al ser multiplicados por el código se produce el mismo efecto que en el transmisor, expanden su banda de frecuencia y reducen su densidad espectral de potencia. Esta reducción es inversamente proporcional a la longitud del código. Por lo tanto, cuanto más largo sea el código mejor se podrá discriminar, siempre y cuando el transmisor esté bien sincronizado con el receptor.

Luego del producto de correlación la señal d(t) vuelve a ser de relativamente banda angosta, mientras que los otros términos son de banda ancha expandida, por lo que mediante un filtro pasabajos, que funciona como integrador y un circuito de decisión se puede recuperar la señal de banda base, tal como se ilustra en el diagrama bloque.

d.- Transmisión por Salto en Frecuencia o Frecuency Hopping Spread Spectrum (FHSS).

En este modo de modulación, la portadora, en vez de operar en una frecuencia fija, cambia muchas veces por segundo de acuerdo a una secuencia de canales preprogramada, que se conoce como secuencia de ruido pseudoaleatoria (PN) .El receptor, programado con la misma secuencia y sincronizado con el transmisor, sigue los saltos de la transmisión a los efectos de decodificar el mensaje. Como la portadora pasa sólo muy pocos milisegundos en cada canal, cualquier interferencia a esa frecuencia resulta de corta duración.

Se presentan a continuación los diagramas bloques del transmisor y el receptor de uno de los métodos de FHSS, para este caso, se utilizó para la modulación digital FSK

A su vez, como en el sistema de DSSS, la etapa de sincronización del transmisor y el receptor, no está representada

Diagrama en Bloques del Transmisor

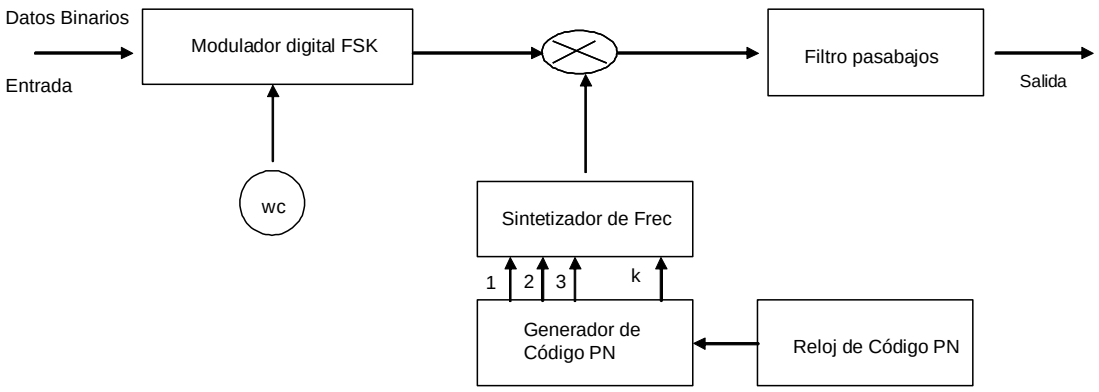


Diagrama en Bloques del Receptor