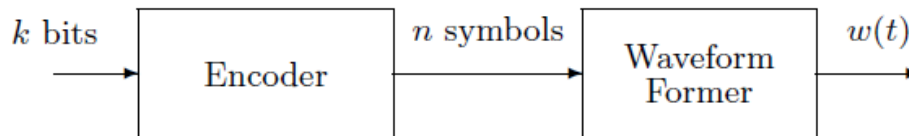


Sistemas de comunicación

De manera primitiva, se modela un transmisor:



Donde el Encoder es el traductor de bits a símbolos que protege la señal del ruido. Como es posible enviar k bits, podremos tener 2^k secuencias o mensajes distintos posibles como entrada. Expresa la entrada en una salida codificada por n símbolos. Si el alfabeto tiene una longitud de m símbolos, habrán m^n secuencias distintas de tamaño n que pueden ser formados con símbolos del mismo alfabeto. Esto implica que:

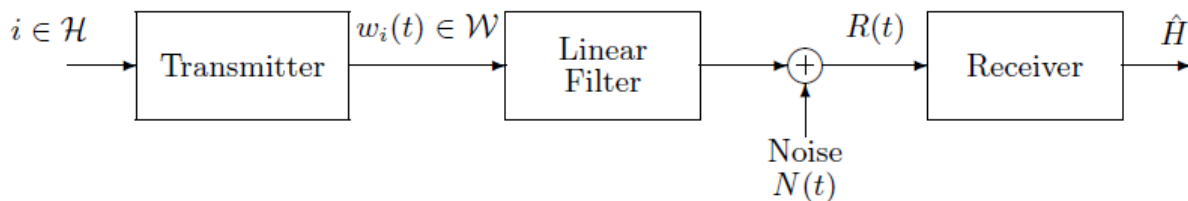
$$2^k \leq m^n$$

Debemos tener suficientes secuencias de símbolos o “palabras” que cubran las 2^k posibilidades de mensajes entrantes.

Una vez obtenidos los símbolos, estos entran al conformador de onda que realiza una conversión digital-analógica. Cada codificación en símbolos se mapea con formas de onda.

Haciendo foco en los aspectos de sistemas de comunicaciones digitales punto a punto:

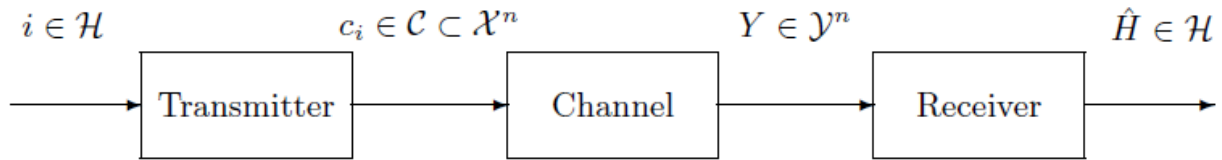
- Los sistemas son representados por bloques que tendrán detalles electrónicos.
- Que sean digitales significa que el mensaje podrá ser un conjunto finito de posibilidades.
- Limitamos a comunicaciones punto a punto ya que constituyen el elemento fundamental de los sistemas de comunicaciones.



La fuente (no representada en la figura), elige un mensaje i que pertenece a un conjunto de mensajes que puede tomar la variable aleatoria $H = \{0, 1, \dots, m - 1\}$. En realidad, el mensaje se representa como una secuencia de bits. El transmisor mapea un mensaje i en una señal $w_i(t) \in W$ donde W es el conjunto de formas de onda. Tanto W como H tienen la misma cardinalidad. El canal filtra la señal y agrega ruido gaussiano $N(t)$. El ruido también resulta ser una variable aleatoria. La tarea del receptor es estimar el mensaje basado en la salida del canal $R(t)$. Entonces \hat{H} es la estimación o hipótesis del mensaje recibido según receptor. El ruido del canal es gaussiano ya que representa la contribución de varias fuentes de ruido.

Diseño de receptor para comunicaciones en tiempo discreto

Para entender la aleatoriedad del ruido introducido por el canal, se comienza a estudiar canales con salidas discretas antes de considerarlas como formas de onda.



En la figura tenemos:

- Una **fente** que produce mensajes a transmitir. Elige un mensaje de un conjunto de posibles mensajes. Se modela como una variable aleatoria H . Se asume que todos los mensajes tienen la misma probabilidad pero en general permitimos que el mensaje i ocurra con una probabilidad $P_H(i)$.
- Un **canal** que se describe generalmente con su alfabeto de entrada X (conjunto de señales que son físicamente compatibles con la entrada del canal), el alfabeto de salida Y y la descripción estadística de la salida dada la entrada. El alfabeto de salida es un subconjunto de los reales. Para cada letra del alfabeto de entrada que entra al canal, la salida le asigna un valor según la variable aleatoria de salida, que depende de la entrada.

- Si Y es una variable aleatoria discreta, describimos el canal con su distribución de probabilidad de Y dada X :

$$P_{Y|X}(\cdot | x)$$

- Si Y es una variable aleatoria continua, describimos la función de densidad de probabilidad (pdf) de Y dada X :

$$f_{Y|X}(\cdot | x)$$

La señal que se envía puede ser alterada por el canal donde atenúa en ciertas frecuencias en ciertos lugares y filtra en otros. Un ejemplo del canal desfasando la señal, genera retardos y hasta elimina componentes en ciertos lugares (el sonido no llega a los dispositivos al mismo tiempo).

- Un **transmisor** encargado de mapear desde el conjunto de mensajes a un conjunto de señales $C = \{c_0, c_1, \dots, c_{m-1}\}$. La existencia de transmisión está dada en el hecho de que la señal que representa el mensaje no es suficiente para excitar el canal.
- Un **receptor**, cuya tarea es estimar la hipótesis H de la asignación de la secuencia de salida del canal. Se representa la estimación realizada por el receptor con el símbolo \hat{i} . El diseño del receptor siempre tendrá como fin minimizar la probabilidad del error, representada por P_e y define la probabilidad de que \hat{H} difiera de H .

Prueba de hipótesis

Refiere al problema de estimar la salida de esta variable aleatoria \hat{H} que toma valores de un alfabeto finito $H = \{0, 1, \dots, m - 1\}$, basado en la salida de una variable aleatoria Y llamada observación.

En comunicaciones, la hipótesis H es el mensaje a transmitir y la observación Y es la salida del canal o secuencia de salidas del canal. El receptor estima la comprensión de H basado en la comprensión de Y . Se asume que, para todo $i \in H$, el diseñador del sistema conoce $P_H(i)$ y $f_{Y|H}(\cdot|i)$ (asumiendo que Y es una variable aleatoria continua, estas son la probabilidad de que ocurra i dentro del conjunto de eventos H y la función de densidad de probabilidad de que ocurra Y dada la ocurrencia de H , respectivamente).

La interpretación del receptor está denotada como \hat{i} y la variable aleatoria correspondiente como \hat{H} , la cual también pertenece al alfabeto de la hipótesis original H . Generalmente es imposible asegurar que $H = \hat{H}$.

El objetivo es crear un receptor cuya estrategia de decisión maximice la probabilidad de que la hipótesis estimada sea igual a la transmitida:

$$P_c = \Pr\{\hat{H} = H\}$$

Es equivalente a minimizar a la probabilidad de error: $P_e = \Pr\{\hat{H} \neq H\} = 1 - P_c$

Ejemplo 2.4 - Problema de comunicación de un bit de información a través de fibra óptica

El bit transmitido es modelado por una variable aleatoria $H \in \{0, 1\}$, cuya probabilidad de ocurrencia para ambos eventos es $P_H(0) = P_H(1) = 1/2$ (equiprobables).

Si $H = 1$, se enciende el LED y la luz viaja a través de la fibra óptica hacia un fotodetector. La salida del fotodetector es el número de fotones detectados. Lo describimos como $Y \in \mathbb{N}$.

El problema recae en decidir cuando $H = 0$ (LED apagado) o $H = 1$ (LED encendido) según la cantidad de fotones receptados Y por el fotodetector, ya que, aun cuando el LED está apagado, el fotodetector recibe fotones debidos a la luz ambiental.

Suponiendo que Y es una [distribución de Poisson](#) con intensidad λ la cual depende de si el LED está encendido o apagado.

$$\begin{aligned} \text{when } H = 0, \quad Y &\sim P_{Y|H}(y|0) = \frac{\lambda_0^y}{y!} e^{-\lambda_0}, \\ \text{when } H = 1, \quad Y &\sim P_{Y|H}(y|1) = \frac{\lambda_1^y}{y!} e^{-\lambda_1}, \end{aligned}$$

donde $0 \leq \lambda_0 < \lambda_1$. Cuando $H = 0$, la observación Y es una distribución de Poisson con intensidad λ_0 y cuando $H = 1$, la observación Y utiliza una intensidad λ_1 .

Conociendo $P_H(i)$ y $f_{Y|H}(\cdot|i)$, se aplica la [regla de Bayes](#), obteniendo:

$$P_{H|Y}(i|y) = \frac{P_H(i)f_{Y|H}(y|i)}{f_Y(y)},$$

donde:

- **P(H)**: probabilidad **a priori** de que se haya enviado H.
- $f_{Y|H}(y|i)$: función de densidad de probabilidad de recibir Y si se envió H (modelo del canal).
- $f_Y(y)$: función de densidad de probabilidad total de recibir Y, sumando sobre todas las posibles hipótesis.

$P_{H|Y}(i|y)$ es la probabilidad a posteriori de obtener H dado Y. Es la probabilidad de que la hipótesis \hat{H} sea cierta *después* de haber observado Y.

Métodos de maximizar la probabilidad de que la hipótesis estimada sea igual a la transmitida

Ambas buscan decidir cuál símbolo fue transmitido, basándose en la observación recibida Y, pero se diferencian en cuánto conocimiento previo utilizan.

Maximum a posteriori decision rule - MAP

Con el objetivo de maximizar la probabilidad de ser correcto, la regla de decisión óptima es:

$$\hat{H}(y) = \arg \max_{i \in \mathcal{H}} P_{H|Y}(i|y) = \arg \max_{i \in \mathcal{H}} P_H(i) f_{Y|H}(y|i)$$

donde $\arg \max_i g(i)$ representa uno de los argumentos i para los cuales la función $g(i)$ alcanza su máximo. Dado un dato observado y , elegimos la hipótesis i que tenga la mayor probabilidad condicional $P(H = i | Y = y)$.

Si hay un empate entre dos hipótesis (es decir, dos valores de i tienen la misma probabilidad máxima), se puede elegir cualquiera de ellas sin afectar la probabilidad de acierto.

La probabilidad total de que nuestra decisión sea correcta (denotada como P_c) se obtiene promediando las probabilidades de acierto para cada posible valor de Y:

$$P_c = \mathbb{E}[P_{H|Y}(\hat{H}(Y)|Y)] = \int_y P_{H|Y}(\hat{H}(y)|y) f_Y(y) dy.$$

donde $E[\cdot]$ representa la esperanza matemática. Esto significa que la regla MAP maximiza la probabilidad global de tomar una decisión correcta al elegir siempre la hipótesis más probable para cada observación $Y = y$.

Maximum likelihood decision rule - ML

Si todas las hipótesis tienen la misma probabilidad previa $P(H = i)$, la regla MAP se reduce a la regla de máxima verosimilitud. Como el argumento que maximiza $P_{H|Y}(i|y)$ también maximiza $f_{Y|H}(y|i)$, entonces la regla de decisión resulta:

$$\hat{H}(y) = \arg \max_{i \in \mathcal{H}} f_{Y|H}(y|i)$$

Tengo mas error en la estimación si no se cumple la condición de equiprobabilidad, por lo que conviene utilizar MAP.

La regla ML se define incluso cuando no se conoce P_H , por lo que es la opción preferida en ausencia de este conocimiento. Las reglas de decisión MAP y ML son equivalentes sólo en el caso de que P_H sea uniforme.

Para ambos métodos es necesario conocer la estadística del canal.

Formas de maximizar la probabilidad de que la hipótesis estimada sea igual a la transmitida:

ML → criterio de elección → que la probabilidad de envío sean iguales (equiprobable)

ML solo mira cuál hipótesis H explica mejor lo que se recibió, sin importar cuán probable era antes de observar Y .

MAP → criterio de elección → la probabilidad de envío pueden no ser iguales

MAP tiene en cuenta también cuán probable era esa hipótesis antes de ver los datos.

Caso binario

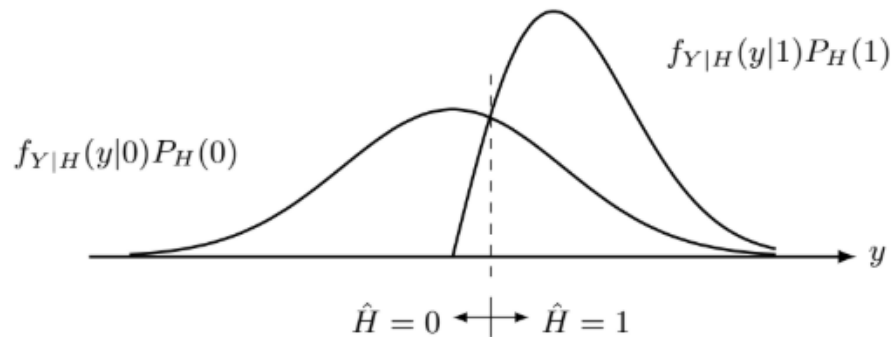
Cuando debemos tomar una decisión entre dos posibles valores (0, 1), la prueba MAP puede escribirse de la siguiente manera:

$$\frac{f_{Y|H}(y|1)P_H(1)}{f_Y(y)} \underset{\hat{H}=0}{\overset{\hat{H}=1}{\geq}} \frac{f_{Y|H}(y|0)P_H(0)}{f_Y(y)}.$$

La notación significa que la prueba MAP elige por $\hat{H} = 1$ cuando la expresión de la izquierda sea mayor o igual a la de la derecha y elige por $\hat{H} = 0$ en el caso contrario. El denominador es irrelevante ya que $f_Y(y)$ es una constante positiva y no afecta en la decisión. También se puede expresar:

$$\begin{array}{c}
 \hat{H} = 1 \\
 f_{Y|H}(y|1)P_H(1) \begin{array}{c} \geq \\ < \end{array} f_{Y|H}(y|0)P_H(0). \\
 \hat{H} = 0
 \end{array}$$

Asumiendo que $y \in R$, se dibujan las funciones de densidad de probabilidad para cada caso, las cuales resultan ayudaran para calcular la probabilidad de error:



La figura presenta:

- En el eje horizontal: el espacio de posibles valores de y (la señal recibida).
- Dos funciones de densidad:
 - $f_{Y|X}(y|0)$: curva de probabilidad de recibir y si se envió 0
 - $f_{Y|X}(y|1)$: curva de probabilidad de recibir y si se envió 1

Según el método de decisión:

ML: Solo compara las curvas de verosimilitud. La decisión se toma en el punto donde se cruzan ambas curvas.

MAP: Pondera cada función con la probabilidad a priori $P(x)$. Si las probabilidades a priori son diferentes, el punto de decisión se mueve.

Es una visualización fundamental para entender cómo las decisiones óptimas cambian dependiendo de si se tiene información previa o no.

Otra alternativa de prueba se obtiene dividiendo ambos lados de la inecuación con la cantidad no negativa $f_{Y|H}P_H(1)$. Obtenemos la *prueba binaria para MAP*:

$$\Lambda(y) = \frac{f_{Y|H}(y|1)}{f_{Y|H}(y|0)} \underset{\hat{H}=0}{\overset{\hat{H}=1}{\geq}} \frac{P_H(0)}{P_H(1)} = \eta.$$

A la izquierda tenemos la **razón de verosimilitud** denotada como $\Lambda(y)$ mientras que a la derecha tenemos el **umbral** η .

- La **razón de verosimilitud** es una medida que compara cuán probable es la observación recibida y bajo dos hipótesis distintas.
- El **umbral** es el valor límite contra el que se compara la razón de verosimilitud para tomar una decisión.

Se cumplirá que:

- Si $\Lambda(y) \geq \eta \rightarrow$ decido $H = 1$
- Si $\Lambda(y) < \eta \rightarrow$ decido $H = 0$

Cuando los eventos son equiprobables, el umbral es la unidad y la prueba MAP se convierte en la prueba ML binaria:

$$f_{Y|H}(y|1) \underset{\hat{H}=0}{\overset{\hat{H}=1}{\geq}} f_{Y|H}(y|0).$$

Función de decisión

La función

$$\hat{H} : \mathcal{Y} \rightarrow \mathcal{H} = \{0, \dots, m-1\}$$

es conocida como función de decisión o función de decodificación. Una manera de describir la función de decisión es mediante las regiones de decisión

$$\mathcal{R}_i = \{y \in \mathcal{Y} : \hat{H}(y) = i\}, i \in \mathcal{H}.$$

Por lo tanto, \mathcal{R}_i es el conjunto de $y \in \mathcal{Y}$ para los cuales $\hat{H}(y) = i$. Las regiones de decisión son zonas en el espacio de las observaciones (por ejemplo, los posibles valores de la señal recibida y) donde el receptor decide cuál símbolo fue enviado.

Probabilidad de error

Para calcular la probabilidad de error, es conveniente calcular la probabilidad de error para cada hipótesis y luego calcular el promedio. En el caso binario, cuando $H = 0$, la decisión es incorrecta si $Y \in \mathcal{R}_1$ (Si en realidad se envió un 0 pero el valor recibido y cae en la región \mathcal{R}_1 , que es donde el receptor decide que fue un 1, entonces se ha cometido un error de decisión) o

lo que es igual a decir que $\Lambda(y) \geq \eta$. De esta forma, si definimos $P_e(i)$ como la probabilidad de error cuando $H = i$:

$$P_e(0) = Pr\{Y \in \mathcal{R}_1 | H = 0\} = \int_{\mathcal{R}_1} f_{Y|H}(y|0) dy$$

lo que es equivalente a decir

$$P_e(0) = Pr\{\Lambda(Y) \geq \eta | H = 0\}.$$

En el gráfico, donde las curvas se cruzan y se solapan es donde **hay ambigüedad**, y por tanto **posibilidad de error**. Si no se superpusieran, no habría errores posibles (decisión perfecta).

Caso m-ario

Tomando ahora un problema con m hipótesis ($H = \{0, 1, \dots, m - 1\}$), reescribimos la regla de decisión MAP:

$$\begin{aligned} \hat{H}_{MAP}(y) &= \arg \max_{i \in \mathcal{H}} P_{H|Y}(i|y) \\ &= \arg \max_{i \in \mathcal{H}} \frac{f_{Y|H}(y|i) P_H(i)}{f_Y(y)} \\ &= \arg \max_{i \in \mathcal{H}} f_{Y|H}(y|i) P_H(i), \end{aligned}$$

donde $f_{Y|H}(\cdot|i)$ es la función de densidad de probabilidad de la observación Y cuando la hipótesis es i y $P_H(i)$ es la probabilidad de la i-ésima hipótesis.

Si hay más de un i que alcanza el máximo en el lado derecho de una (y por lo tanto todas) de las expresiones anteriores, entonces podemos decidir por cualquier i de ese tipo sin afectar la probabilidad de error. Si queremos que la regla de decisión sea unívoca, podemos acordar, por ejemplo, que en caso de empates, elegimos el i más grande que alcanza el máximo.

Cuando todas las hipótesis tienen la misma probabilidad, usamos la regla ML:

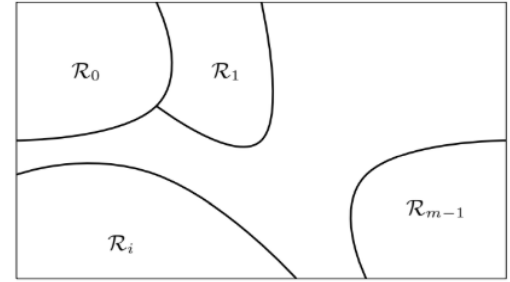
$$\hat{H}_{ML}(y) = \arg \max_{i \in \mathcal{H}} f_{Y|H}(y|i).$$

Siempre asumimos $f_{Y|H}$ está dada como parte de la formulación del problema o que se puede deducir a partir de la configuración. En comunicaciones generalmente conocimos el transmisor y el canal:

- El transmisor es la función de mapeo de H a $\mathcal{C} \subset X^n$
- El canal está descrito por la función de densidad de probabilidad $f_{Y|X}(y|x)$ conocida para todo $x \in X^n$ y todo $y \in Y^n$.

A partir de estos dos, obtenemos inmediatamente $f_{Y|H}(y|i) = f_{Y|X}(y|c_i)$ donde c_i es la señal asignada a i .

Es importante notar que la función de decisión asigna un $i \in H$ a cada $y \in R^n$ y puede ser descrita por las regiones de decisión R_i , $i \in H$, donde R_i consiste en aquellos y para los cuales $\hat{H}(y) = i$. Es conveniente pensar en R^n cómo dividido por regiones de codificación.



Utilizando las regiones de decodificación para expresar la probabilidad de error P_e o también la probabilidad de decidir correctamente P_c y condicionando en $H = i$ tenemos:

$$P_e(i) = 1 - P_c(i) = 1 - \int_{R_i} f_{Y|H}(y|i) dy.$$

Función Q

Permite calcular la probabilidad desde un valor x hasta el infinito, teniendo en cuenta una distribución gaussiana. Sea Z es una variable aleatoria normalmente distribuida con media cero y varianza unitaria: $Z \sim N(0, 1)$, entonces

$$Pr\{Z \geq x\} = Q(x)$$

donde

$$Q(x) := \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{\xi^2}{2}} d\xi,$$

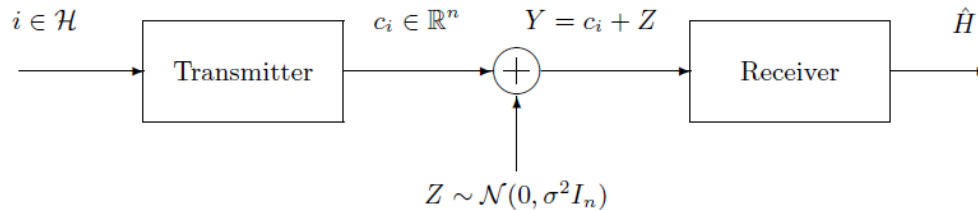
Para distribuciones con media m y varianza σ^2 : $Z \sim N(m, \sigma^2)$, la probabilidad $Pr\{Z \geq x\}$ también puede ser escrita usando la función Q . De hecho, el evento $\{Z \geq x\}$ es equivalente a $\{\frac{Z-m}{\sigma} \geq \frac{x-m}{\sigma}\}$. Pero, como $\frac{Z-m}{\sigma} \sim N(0, 1)$ entonces $Pr\{Z \geq x\} = Q(\frac{x-m}{\sigma})$.

Propiedades:

1. Si $Z \sim N(0, 1)$, $F_Z(z) := Pr\{Z \leq z\} = 1 - Q(z)$.
2. $Q(0) = 1/2$, $Q(-\infty) = 1$, $Q(\infty) = 0$.
3. $Q(-x) + Q(x) = 1$
4. $\frac{1}{\sqrt{2\pi}} e^{-\frac{\alpha^2}{2}} \left(\frac{\alpha^2}{1+\alpha^2} \right) < Q(\alpha) < \frac{1}{\sqrt{2\pi}\alpha} e^{-\frac{\alpha^2}{2}}$, para $\alpha > 0$.
5. Alternativa de Q para $x \geq 0$: $Q(x) = \frac{1}{\pi} \int_0^{\frac{\pi}{2}} e^{-x^2/(2\sin^2(\theta))} d\theta$.
6. $Q(\alpha) \leq \frac{1}{2} e^{-\frac{\alpha^2}{2}}$, para $\alpha \geq 0$.

Diseño de receptor para un canal AWGN en tiempo discreto

En la figura, el transmisor mapea $H = i$ a una n -tupla de señales $c_i \in H$. El canal agrega un vector de ruido aleatorio que tiene media cero y componentes gaussianas independientes e idénticamente distribuidas con varianza σ^2 : $Z \sim N(0, \sigma^2 I_n)$.



- Que tenga media cero implica que el ruido no tiene ningún sesgo hacia valores positivos o negativos, su promedio es cero.
- La matriz de covarianza ($\sigma^2 I_n$) significa que cada componente de ruido tienen varianza σ^2 y no hay correlación entre ellos. La matriz de identidad I_n es diagonal, lo que en matrices de covarianza significa ausencia de correlación. En una distribución gaussiana multivariada, la ausencia de correlación implica independencia estadística sólo si la distribución es gaussiana.

La observación resulta $Y = c_i + Z$.

Caso binario

Suponiendo que los mensajes $H \in \{0, 1\}$ son equiprobables y asumiendo que en el mapeo se convierten:

- $H = 0$ en $c_0 \in R \rightarrow Y \sim N(c_0, \sigma^2)$
- $H = 1$ en $c_1 \in R \rightarrow Y \sim N(c_1, \sigma^2)$

Las variables observadas toman una distribución gaussiana donde la media sería el valor que utiliza el transmisor para mapear la hipótesis correspondiente.

Podemos expresar la estadística de la salida para cada hipótesis como:

$$f_{Y|H}(y|0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y - c_0)^2}{2\sigma^2} \right\}$$

$$f_{Y|H}(y|1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y - c_1)^2}{2\sigma^2} \right\}.$$

Calculando la razón de verosimilitud y el umbral:

$$\Lambda(y) = \frac{f_{Y|H}(y|1)}{f_{Y|H}(y|0)} = \exp \left\{ -\frac{(y - c_1)^2 - (y - c_0)^2}{2\sigma^2} \right\} = \exp \left\{ y \frac{c_1 - c_0}{\sigma^2} + \frac{c_0^2 - c_1^2}{2\sigma^2} \right\}.$$

$$\eta = \frac{P_H(0)}{P_H(1)}.$$

Para comparar estos parámetros, se utilizará la función logaritmo natural a ambos lados:

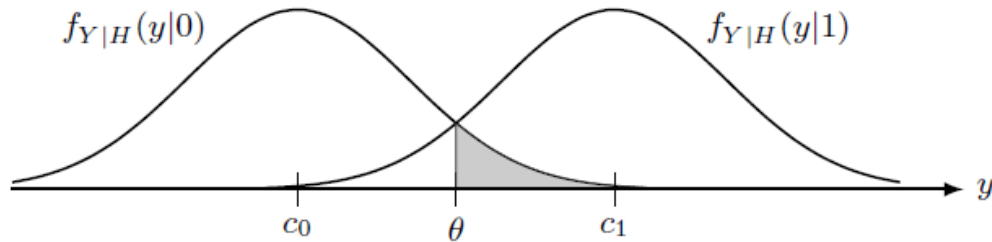
$$\begin{array}{ccc} \hat{H} = 1 & & \\ y \frac{c_1 - c_0}{\sigma^2} + \frac{c_0^2 - c_1^2}{2\sigma^2} & \geq & \ln \eta. \\ \hat{H} = 0 & & \end{array}$$

Se utiliza la razón de verosimilitud logarítmica para que la expresión se reduzca a su exponente.

Suponiendo que $c_1 > c_0$, entonces podemos dividir ambos lados por $\frac{(c_1 - c_0)}{\sigma^2}$. No cambia el resultado de la comparación. Moviendo las constantes a la derecha se obtiene:

$$\hat{H}_{\text{MAP}}(y) = \begin{cases} 1, & \text{si } y \geq \theta \\ 0, & \text{si no} \end{cases} \quad \text{donde} \quad \theta = \frac{\sigma^2}{c_1 - c_0} \ln \eta + \frac{c_0 + c_1}{2}.$$

Si $P_H(0) = P_H(1)$, entonces $\ln \eta = 0$ y el umbral θ se convierte en el punto medio $\frac{(c_1 + c_0)}{\sigma^2}$.



En un receptor digital, el nivel de decisión óptimo es el valor límite que se usa para decidir qué símbolo se transmitió (por ejemplo, un 0 o un 1), basándose en la señal que se recibió, que suele estar afectada por ruido.

El área sombreada representa la probabilidad de error condicionada por $H = 0$, la cual se calcula:

$$P_e(0) = \Pr\{Y > \theta | H = 0\} = \int_{\theta}^{\infty} f_{Y|H}(y|0) dy$$

Esta es la probabilidad de que una variable aleatoria gaussiana con media c_0 y varianza σ^2 excede el umbral θ .

La probabilidad de error se calcula utilizando la función Q. De esta manera podemos calcular:

$$P_e(0) = Q\left(\frac{\theta - c_0}{\sigma}\right) \text{ y } P_e(1) = Q\left(\frac{c_1 - \theta}{\sigma}\right)$$

entonces

$$P_e = P_H(0)Q\left(\frac{\theta - c_0}{\sigma}\right) + P_H(1)Q\left(\frac{c_1 - \theta}{\sigma}\right)$$

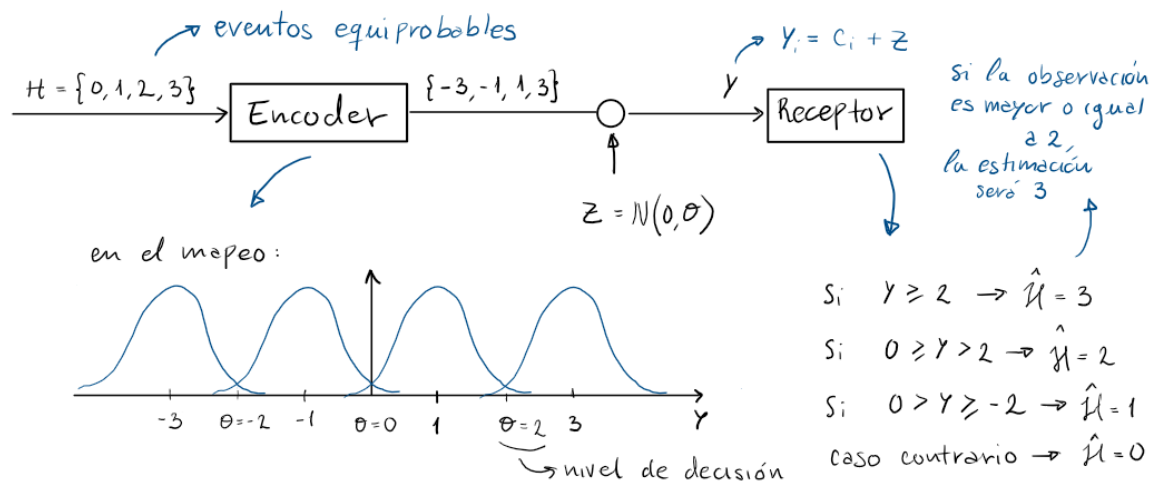
En el caso binario con hipótesis equiprobables:

$$P_H(0) = P_H(1) = 1/2. \text{ entonces } \frac{\theta - c_0}{\sigma} = \frac{c_1 - \theta}{\sigma} = \frac{c_1 - c_0}{2\sigma} = \frac{d}{2\sigma}$$

donde d es la distancia entre c_0 y c_1 . La probabilidad de error total es:

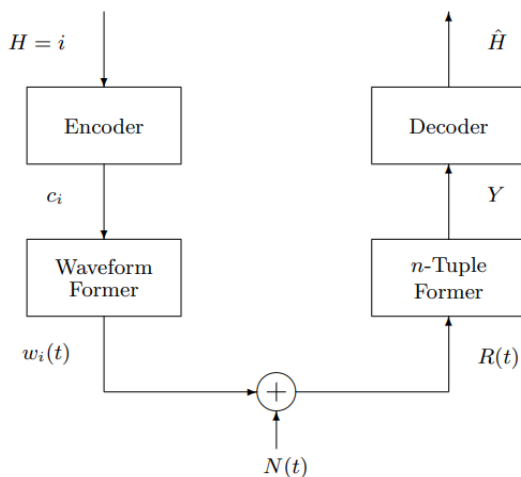
$$P_e = Q\left(\frac{d}{2\sigma}\right)$$

Caso m-ario:



Diseño de receptor para canales AWGN de tiempo continuo

El diseño del receptor para el canal AWGN de tiempo discreto se basó en ideas geométricas dentro de un espacio de producto interno. Lo mismo se replica para el caso en tiempo continuo.



Se descompone el transmisor y el receptor en distintos bloques:

- Transmisor: consiste en un encoder que mapea el mensaje $i \in H$ en una n-tupla c_i seguido de un formador de ondas que mapea c_i en una forma de onda $w_i(t)$.

- Receptor: consiste en un formador de n-tuplas que toma la salida del canal y produce una n-tupla Y .

Entre el formador de ondas y el formador de n-tuplas tenemos el canal AWGN de tiempo discreto.

Ruido Blanco Gaussiano

Se asume que las observaciones realizadas por el receptor son arrojadas por algún filtro lineal e invariante en el tiempo. Si $N(t)$ representa el ruido introducido por el canal, el receptor ve una versión filtrada $Z(t)$ de $N(t)$.

Se modela $Z(t)$ como un proceso estocástico y, como uno, se describe por estadísticas $Z(t_1), Z(t_2), \dots, Z(t_k)$ para cualquier entero positivo k y para cualquier colección finita de tiempos de muestra.

Si la respuesta al impulso es $h(t)$, la teoría de sistemas lineales sugiere que

$$Z(t_i) = \int N(\alpha)h(t_i - \alpha)d\alpha,$$

$Z(t_i)$ es la variable aleatoria que modela la salida en el tiempo t_i de un filtro lineal con respuesta al impulso invariante en el tiempo $h(t)$, alimentado con ruido blanco gaussiano $N(t)$.

La función $h(t_i - \alpha)$ se representa como $g_i(\alpha)$.

Definición:

$N(t)$ es el ruido blanco gaussiano de densidad espectral de potencia $\frac{N_0}{2}$ si, para cualquier colección finita de funciones reales $g_1(\alpha), \dots, g_k(\alpha)$,

$$Z_i = \int N(\alpha)g_i(\alpha)d\alpha, \quad i = 1, 2, \dots, k$$

es una colección de variables aleatorias conjuntamente gaussianas con media cero y covarianza

$$\text{cov}(Z_i, Z_j) = \mathbb{E}[Z_i Z_j^*] = \frac{N_0}{2} \int g_i(t)g_j^*(t)dt = \frac{N_0}{2} \langle g_i, g_j \rangle.$$

Esta definición nos dice cómo debe comportarse un proceso aleatorio $N(t)$ para que se lo pueda considerar ruido blanco gaussiano con densidad espectral de potencia $\frac{N_0}{2}$.

$N(t)$ es el ruido gaussiano con densidad espectral de potencia $N_0/2$ si para un conjunto finito de funciones realmente valuadas $g_1(\alpha), \dots, g_k(\alpha)$, $Z_i = \dots$ es un conjunto de variables aleatorias conjuntamente gaussianas de media cero y covarianza $\text{cov}(Z_i, Z_j) = \dots$ es decir que si $g_1(t), \dots, g_k(t)$ es un conjunto de funciones ortonormales realmente valuadas, $Z = (Z_1, \dots, Z_k)^T$ es un vector aleatorio con distribución gaussiana de media cero y componentes independientes e igualmente distribuidas de varianza $N_0/2$.

Ruido blanco: significa que el ruido es independiente del tiempo, su espectro de potencia es plano (constante en todas las frecuencias) y su valor en un instante t es independiente del valor en otro instante t' .

Gaussiano: cualquier combinación lineal de muestras del ruido sigue una distribución normal.

Cada Z_i es una variable aleatoria obtenida al proyectar $N(\alpha)$ sobre la función $g_i(\alpha)$, como si se aplicara un filtro a $N(t)$. Es una especie de producto interno entre el ruido y cada función g_i .

El ruido blanco gaussiano es tal que, cuando lo “filtras” a través de funciones arbitrarias g_i , obtenés variables aleatorias gaussianas con media cero y una covarianza proporcional al producto interno entre esas funciones.

Ejemplo 3.6: Considerando dos filtros pasa banda con respuesta de frecuencia no superpuestas pero idénticas, según la relación de Parseval, las respuestas de impulso correspondientes son ortogonales entre sí. Si alimentamos los dos filtros con ruido blanco gaussiano y mostramos su salida, obtenemos dos variables aleatorias gaussianas independientes e idénticamente distribuidas (iid). Extendiendo el experimento a n filtros con respuestas de frecuencia no superpuestas, obtendremos n variables aleatorias que son iid, por lo tanto, de varianza idéntica.

Estadísticas suficientes y observables

Queremos detectar qué mensaje H se transmitió, observando una señal recibida $R(t)$, que es la suma de la señal enviada $\omega_i(t)$ y ruido blanco gaussiano $N(t)$. El problema es que $R(t)$ no se puede observar directamente ni manejar fácilmente, así que buscamos representarlo mediante un conjunto finito de mediciones que contenga toda la información útil.

La estadística suficiente es una función del observable que conserva toda la información necesaria para tomar una decisión óptima sobre la hipótesis (por ejemplo, qué mensaje se transmitió). Si tienes una estadística suficiente, no necesitas más datos.

Es a partir de Y que se intenta inferir la información original enviada. Lo que podemos observar a través de experimentos son k -tuplas $V = (V_1, \dots, V_k)^T$ tales que

$$V_i = \int_{-\infty}^{\infty} R(\alpha) g_i^*(\alpha) d\alpha, \quad i = 1, 2, \dots, k,$$

donde k es un entero positivo arbitrario y $g_1(t), \dots, g_k(t)$ son formas de onda arbitrarias de energía finita. Aunque el operador de conjugación compleja $g_i^*(\alpha)$ resulta redundante en el caso de señales reales, se vuelve necesario cuando se trabaja con representaciones en banda base de respuestas al impulso pasabanda, ya que estas representaciones son típicamente complejas.

El conjunto $W = \{\omega_0(t), \dots, \omega_{m-1}(t)\}$ contiene las posibles señales transmitidas. Estas señales viven en un espacio vectorial V , un subconjunto del espacio L^2 (espacio de señales cuadráticamente integrables). Este espacio tiene dimensiones finitas (dimensión n).

Sea $\psi_1(t), \dots, \psi_n(t)$ una base ortonormal arbitraria para V , podemos expresar cualquier señal $\omega_i(t) \in W$ como combinación lineal:

$$w_i(t) = \sum_{k=1}^n c_{ik} \psi_k(t)$$

El receptor proyecta la señal recibida $R(t)$ sobre la base $\psi_n(t)$:

$$Y_i = \int R(\alpha) \psi_i^*(\alpha) d\alpha$$

Como $R(t) = \omega_i(t) + N(t)$, también se proyecta el ruido:

$$Y_k = c_{ik} + Z_k \text{ donde } Z_k = \langle N(t), \psi_k(t) \rangle$$

La proyección del ruido blanco gaussiano sobre la base $\psi_n(t)$ genera un vector gaussiano.

Considerando ahora una base ortonormal del espacio ortogonal a V llamado V^\perp con dimensión \tilde{n} . Sea U el espacio de producto interno generado por $V \cup \{g_1(t), \dots, g_{\tilde{n}}(t)\}$ y sea

$\{\psi_1(t), \dots, \psi_n(t), \phi_1(t), \dots, \phi_{\tilde{n}}(t)\}$ una base ortonormal para U obtenida mediante la extensión de la base ortonormal $\{\psi_1(t), \dots, \psi_n(t)\}$ para V . Se proyecta $R(t)$ sobre esta base también, obteniendo un nuevo vector U , que contiene solo ruido:

$$U_i = \int R(\alpha) \phi_i^*(\alpha) d\alpha, \quad i = 1, \dots, \tilde{n}.$$

Podemos recuperar V a partir de Y y U , debido a que, a partir de las proyecciones sobre una base, podemos obtener la proyección sobre cualquier forma de onda en el espacio generado por la base:

$$\begin{aligned}
 V_i &= \int_{-\infty}^{\infty} R(\alpha) g_i^*(\alpha) d\alpha \\
 &= \int_{-\infty}^{\infty} R(\alpha) \left[\sum_{j=1}^n \xi_{i,j} \psi_j(\alpha) + \sum_{j=1}^{\tilde{n}} \xi_{i,j+n} \phi_j(\alpha) \right]^* d\alpha \\
 &= \sum_{j=1}^n \xi_{i,j}^* Y_j + \sum_{j=1}^{\tilde{n}} \xi_{i,j+n}^* U_j,
 \end{aligned}$$

donde los ξ son el conjunto único de coeficientes en la expansión ortogonal de $g_i(t)$ con respecto a la base $\{\psi_1(t), \dots, \psi_n(t), \phi_1(t), \dots, \phi_{\tilde{n}}(t)\}$.

Cuando $H = i$:

$$Y_j = \int R(\alpha) \psi_j^*(\alpha) d\alpha = \int (w_i(\alpha) + N(\alpha)) \psi_j^*(\alpha) d\alpha = c_{i,j} + Z_{|V,j},$$

donde $c_{i,j}$ es la componente j-ésima de la n-tupla de coeficientes c_i , que representa la forma de onda $\omega_i(t)$ con respecto a la base ortonormal elegida y $Z_{|V,j}$ es una variable aleatoria gaussiana de media cero y varianza $\frac{N_0}{2}$.

De manera similar:

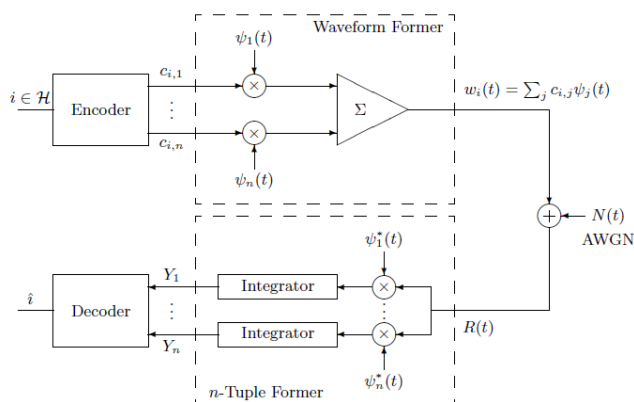
$$U_j = \int R(\alpha) \phi_j^*(\alpha) d\alpha = \int (w_i(\alpha) + N(\alpha)) \phi_j^*(\alpha) d\alpha = \int N(\alpha) \phi_j^*(\alpha) d\alpha = Z_{\perp V,j},$$

Como la señal $\omega_i(t)$ está en V , su proyección sobre V^\perp es cero. Por lo tanto, para tomar la mejor decisión sobre el mensaje transmitido H , solo hace falta observar Y , que es la proyección de la señal recibida sobre el espacio de las señales transmitidas. Cualquier otra parte de la señal contiene solo ruido y es irrelevante para la detección.

$Z_{|V}$ y $Z_{\perp V}$ son independientes entre sí y de H . La independencia ayuda a descartar muestras que no nos hacen falta (vector U). Si tengo componentes de ruido correlacionados (existe dependencia), se puede reducir una para reducir la otra.

Arquitectura del transmisor y receptor

Un receptor MAP para el canal AWGN de formas de onda puede estructurarse:



En el frente del receptor se calcula $Y \in R^n$ a partir de $R(t)$ en un bloque llamado formador de n-tuplas. Realiza una gran reducción de datos desde la salida del canal $R(t)$ hasta la estadística

suficiente Y . El problema de prueba de hipótesis basado en la observación Y es

$$H = i : \quad Y = c_i + Z,$$

donde $Z \sim N(0, \frac{N_0}{2} I_n)$ es independiente de H .

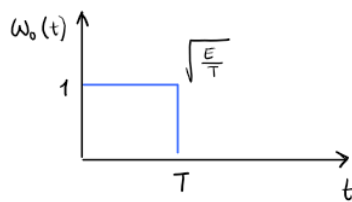
También podemos descomponer el transmisor en un módulo que produce símbolos c_i , llamado *encoder* y un módulo que produce señales $\omega_i(t)$ llamado *formador de formas de onda*. La descomposición del transmisor y receptor es consistente con la filosofía basada en el modelo OSI en el sentido que el encoder y el decoder están diseñados como si hablaran entre sí directamente mediante el canal AWGN en tiempo discreto. En la realidad, el canal visto por el par encoder/decoder es el resultado del trabajo proporcionado por el formador de formas de onda y el formador de n-tuplas.

Ejemplo 3.7

Las señales $W = \{\omega_0(t), \omega_1(t)\}$ comparten el mismo codebook $C = \{c_0, c_1\}$ donde $C_0 = (\sqrt{E}, 0)^T$ y $C_1 = (0, \sqrt{E})^T$

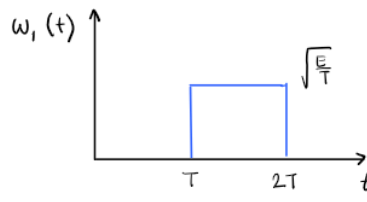
Se verifica que $\langle \omega_i, \omega_j \rangle = E$ si $i = j$ y es igual a 0 en el caso contrario.

Modulación de posición de pulso rectangular



$$w_0 = \frac{\sqrt{E}}{T} \mathbb{1}\{t \in [0, T]\}$$

función que simboliza un pulso rectangular



$$w_1 = \frac{\sqrt{E}}{T} \mathbb{1}\{t \in [T, 2T]\}$$

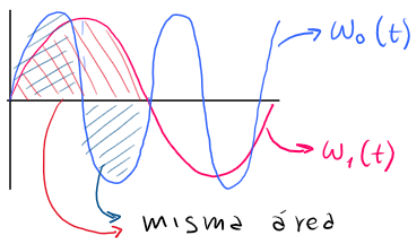
vale 1 en el intervalo $[0, T]$

$$\langle w_0, w_1 \rangle = \int w_0 \cdot w_1 dt = 0$$

Al no haber solapamiento, la integral nos dará 0

las señales son ortogonales

Modulación por desplazamiento de frecuencia



Por simetría, son ortogonales

$$w_0(t) = \sqrt{\frac{2E}{T}} \sin\left(\frac{\pi k t}{T}\right) \mathbb{1}\{t \in [0, T]\}$$

$$w_1(t) = \sqrt{\frac{2E}{T}} \sin\left(\frac{\pi \ell t}{T}\right) \mathbb{1}\{t \in [0, T]\}$$

donde $k > \ell$

Generalización y estructuras alternativas de receptor

Repasando:

- La fuente de información produce un mensaje $H = i$ con una cierta probabilidad $P_H(i)$ donde i pertenece al conjunto de mensajes H . Cuando se transmite el mensaje $H = i$, la salida del canal es $R(t) = \omega_i(t) + N(t)$ donde $\omega_i(t)$ es una señal de una constelación de señales conocida y de energía finita: $W = \{\omega_0(t), \omega_1(t), \dots, \omega_{m-1}(t)\}$ y $N(t)$ es ruido blanco Gaussiano.
- Se asume naturalmente que el espacio vectorial V generado por las señales W forma un espacio de producto interno. Esto implica que la única señal de energía cero es la señal nula, lo cual es realista en sistemas físicos. Para representar V , se construye una base ortonormal: $\{\psi_1(t), \psi_2(t), \dots, \psi_n(t)\}$.
- En el receptor obtenemos una estadística suficiente proyectando la señal recibida $R(t)$ sobre cada elemento de esta base ortonormal, calculando los productos internos:

$$Y_i = \langle R, \psi_i \rangle \text{ para } i = 1, \dots, n$$

Agrupando estos resultados:

$$Y_i = (Y_1, Y_2, \dots, Y_n)^T$$

Ahora supongamos un problema de prueba de hipótesis donde la variable observada tiene una distribución condicional exponencial:

$$f_{Y|H}(y|i) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(-\frac{\|y - c_i\|^2}{2\sigma^2}\right)$$

donde c_i es el vector de coeficientes asociado a la señal $\omega_i(t)$ y $\sigma^2 = \frac{N_0}{2}$ siendo N_0 la densidad espectral de potencia del ruido.

El receptor MAP decide que se transmitió el mensaje \hat{i} que maximiza el producto:

$$P_H(i) f_{Y|H}(y|i)$$

Como la función de densidad es una exponencial, podemos tomar su logaritmo natural para simplificar el criterio de decisión. Esto da lugar a tres reglas equivalentes para el receptor MAP:

1. Elegir el mensaje j que minimiza

$$\|y - c_j\|^2 - N_0 \ln P_H(j).$$

2. Elegir el mensaje j que maximiza

$$\langle y, c_j \rangle - \frac{\|c_j\|^2}{2} + \frac{N_0}{2} \ln P_H(j)$$

3. Elegir el mensaje j que maximiza

$$\int r(t) w_j^*(t) dt - \frac{\|w_j\|^2}{2} + \frac{N_0}{2} \ln P_H(j)$$

Estos enfoques son equivalentes, sólo difieren en la forma matemática.

- El primero se basa en la distancia entre vectores (mínima distancia ajustada).
- El segundo usa productos internos (máxima correlación ajustada)
- El tercero opera directamente sobre las señales continuas recibidas (integración de correlación).

Estas reglas permiten diseñar receptores óptimos incluso cuando la distribución de los mensajes no es uniforme (es decir, algunos mensajes pueden ser más probables que otros).

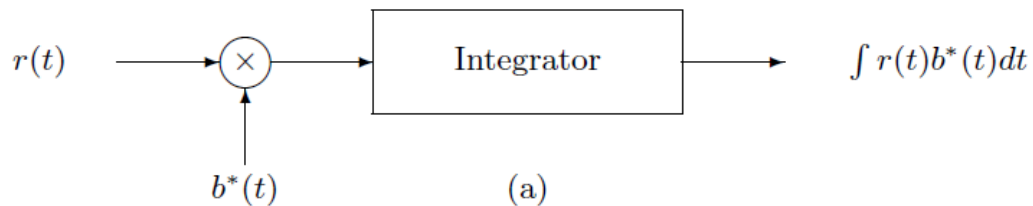
Las reglas MAP 2 y 3 requieren realizar una operación del siguiente tipo:

$$\int r(t)b^*(t)dt,$$

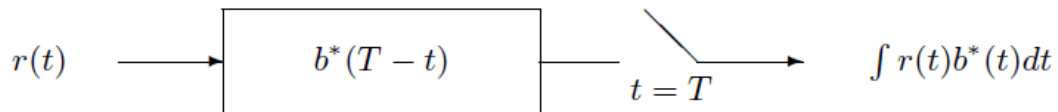
donde $b(t)$ es alguna función $\omega_j(t)$ o $\psi_j(t)$. En la regla 2, esta operación se traduce en calcular productos internos entre el vector de observación y y los vectores de constelación c_j . La regla 3 se trata directamente de integrar la señal recibida $r(t)$ contra la señal de referencia $\omega_j^*(t)$.

Hay dos formas de implementar esta operación:

1. Correlador: multiplica la señal recibida $r(t)$ por la conjugada de la función $b(t)$ y luego integra.



2. Filtro adaptado (Matched filter): está diseñado especialmente para maximizar la respuesta a una señal particular que esperamos recibir, en presencia de ruido utiliza un filtro cuya respuesta al impulso es $h(t) = b^*(T - t)$. La salida del filtro se toma en $t = T$ para obtener el mismo resultado que el correlador.



Invertimos la señal en el tiempo (reflejando respecto a T), la conjugamos (si es compleja) y obtenemos la forma del filtro. Luego la salida del filtro se muestra en el instante $t = T$ para decidir cuál señal fue enviada.

La salida del filtro adaptado es

$$y(t) = \int r(\alpha)h(t - \alpha)d\alpha = \int r(\alpha)b^*(T + \alpha - t)d\alpha,$$

Códigos

Gram-Schmidt

```
import numpy as np

def gram_schmidt(vectors):
    orthonormal_basis = []
    for v in vectors:
        # Quitar proyecciones sobre los vectores ya ortonormalizados
        for u in orthonormal_basis:
            v = v - np.dot(v, u) * u
        norm = np.linalg.norm(v)
        if norm > 1e-10: # evitar división por cero (vector nulo)
            orthonormal_basis.append(v / norm)
    return orthonormal_basis

# Vectores originales
alpha_1 = np.array([1, 0, 1, 1], dtype=float)
alpha_2 = np.array([2, 1, 0, 1], dtype=float)
alpha_3 = np.array([1, 0, 1, -2], dtype=float)
alpha_4 = np.array([2, 0, 2, -1], dtype=float)

# Lista de vectores
vectors = [alpha_1, alpha_2, alpha_3, alpha_4]

# Ortonormalización
basis = gram_schmidt(vectors)

# Mostrar resultados
print("Base ortonormal:")
for i, vec in enumerate(basis, 1):
    print(f"u_{i} =", np.round(vec, 4))
```

Función Q

```
from scipy.special import erf
import numpy as np

# Función Q
def Q_function(x):
    return 0.5 * (1 - erf(x / np.sqrt(2)))

#Definimos los parametros
m = 1 #media
v = 0 # varianza
x = (m-1)/v

q_value = Q_function(x)
print("Q({}) = {:.15f}".format(x, q_value))
```

Norma

```
import sympy as sp

# Definir variable simbólica
t = sp.Symbol('t')

# Definir la función simbólicamente
f = -2*t + 4

# Definir los límites de integración
a = 1
b = 2

# Calcular la norma
norma = sp.sqrt(sp.integrate(f**2, (t, a, b)))

# Mostrar el resultado
print(norma.evalf()) # Evalúa numéricamente
```

Pe en la detección con modulación 6PAM

```
import numpy as np
import math

def Q(x):
    return 0.5 * math.erfc(x / np.sqrt(2))

d = 1
umbral = d / 2

p5 = Q(umbral)
print(round(p5, 2))

p3 = 2 * Q(umbral)
print(round(p3, 2))
```

Sonda espacial rumbo a Mercurio - relación epsilon/No

```
import math
from scipy.stats import norm
import numpy as np
kb=1.381e-23 #Constante de Boltzmann
def BER(pt, lda, gt, gr, d, tn, rb):
    d = d * 1000
    rb = rb * 1000
    epsilon = ((1 / rb) * gt * gr * pt) / (((4 * 3.14 * d) / lda) ** 2) * kb * tn
    return epsilon
print(BER(14.9,0.13,575.44 ,1.38e6 ,1.6e6,13.5 ,117.6))#Cambiar datos no sean vergas
```

Cálculo tita

```
import numpy as np

#Definimos los parámetros
x1 = -0.5
x2 = 4.7
m = 0
v2 = 1
px1 = 0.333
px2 = 0.666

tita = ((x1+x2)/2) + (v2/(x2-x1))*np.log(px1/px2)
print(round(tita,2))
```

Conformador de onda - raíz de coseno realizado

```
import numpy as np

def rrcosfilter(t, beta, Ts, iT):
    return 1/np.sqrt(Ts) * np.sinc((t-iT)/Ts) * np.cos(np.pi*beta*(t-iT)/Ts) / (1 -
(2*beta*(t-iT)/Ts) ** 2)

def raiz_coseno_realizado(simbolos, Beta, T, tt):
    sum_sig = 0
    signal = []

    for i in range(len(simbolos)):
        signal.append(simbolos[i]*rrcosfilter(tt,Beta,T,i*T))
        sum_sig = sum_sig + signal[i]
        #plt.plot(tt,output[i])
    return sum_sig,signal
```

bit error rate para una modulación antipodal

```
import numpy as np
from scipy.stats import norm
kb=1.381e-23
def BER(pt,lda,gt,gr,d,tn,rb):
    arg=(1/(rb*1000)*(gt*gr*pt)/((4*3.14*d*1000/lda)**2*tn*kb))
    return(np.round(norm.sf(np.sqrt(2*arg)),4))
```

Cálculo de la esperanza $E\{x\}$

```
def function(n):
    return(2/(5**n))*n
sumatoria = sum(function(n) for n in range(101))
print(sumatoria)
```

F(0) para gaussiana media cero

```
import scipy.stats as stats

mu = 0
sigma = 1 # Asumimos una desviación estándar de 1 para una distribución normal
estándar

# Calcular la CDF en x = 0
F_0 = stats.norm.cdf(0, mu, sigma)

print(f"La función de distribución acumulativa en x = 0 es {F_0}")
```

Correlador (forma de onda triangular)

```
import numpy as np
def correlador(signal,T):
    t = np.arange(0,2+T,T)
    waveform = np.zeros(len(t))
    for i in range(len(t)):
        if t[i] <= 1:
            waveform[i] = t[i]
        else:
            waveform[i] = 2 - t[i]
    min_len = len(signal) if len(signal)<len(waveform) else len(waveform)
    corr = 0
    for i in range(min_len):
        corr += signal[i] * waveform[i]
    return corr*T
```

Calculo producto interno

```
import sympy as sp

# Defino la variable simbólica
t = sp.symbols('t')

# Defino las dos señales
beta2 = 2*t - 4
psi1 = -sp.sqrt(3)*t + 2*sp.sqrt(3)

# Calculo el producto interno
inner = sp.integrate(beta2 * psi1, (t, 1, 2))

# Muestro el resultado numérico
print("resultado =", float(inner.evalf()))
```

Conformador de onda (Single-Shot QAM)

```
def waveformer(c, T=1, fc=1, delta_t=0.1):
    def phi1(t):
        return np.sqrt(2/T) * np.cos(2 * np.pi * fc * t)

    def phi2(t):
        return np.sqrt(2/T) * np.sin(2 * np.pi * fc * t)

    c1, c2 = c

    num = int(T / delta_t) + 1
    t = np.linspace(0, T, num)

    w = c1 * phi1(t) + c2 * phi2(t)

    return w
```

Cálculo de la PE cuando se transmite una de las hipótesis binarias

```
import numpy as np
from scipy.stats import norm

# Varianza del ruido
sigma_squared = 0.01
sigma = np.sqrt(sigma_squared)

# Usamos la fórmula:  $P_e = Q(1 / \sigma)$ 
x = 1 / sigma
Pe = norm.sf(x)

print(f"Probabilidad de error (Pe): {Pe:.6f}")
```

Caja de cartas

```
# Probabilidades iniciales
P_C1 = 0.5
P_C2 = 0.5
P_B_C1 = 999/1000
P_B_C2 = 1/1000

# Aplicar el teorema de Bayes
P_C1_B = (P_B_C1 * P_C1) / ((P_B_C1 * P_C1) + (P_B_C2 * P_C2))

# Imprimir el resultado
print(f"La probabilidad de que la carta blanca seleccionada provenga de la Caja 1 es {P_C1_B}")
```


Detector de mentiras

```

import numpy as np

# Hypothesis source : TRUE (1) False (0)
def get_hypothesis(p_true,nb):
    p_lye = 1 - p_true
    out = np.random.choice(2,nb,p=[p_true,p_lye])
    return out

# Output of brain cell activity instrument
def get_brain_activity(hypothesis, alfa, beta):
    out = [np.random.exponential(1/alfa) if hypothesis[i]==1 else
np.random.exponential(1/beta) for i in range(len(hypothesis)) ]
    return out

# Lyier Detector
def detect(input, alpha, beta, p_true):
    p_lye = 1 - p_true
    theta = 1/(beta-alpha) * np.log( (p_lye*beta)/(p_true*alpha))
    out_hypothesis = [1 if input[i]>theta else 0 for i in range(len(input))]
    return [theta, out_hypothesis]

def simulador_pe(alpha, beta, p_truth, nb):
    hypothesis = get_hypothesis(p_truth,nb)
    observation = get_brain_activity(hypothesis, alpha, beta)
    [theta, hyp_detected] = detect(observation, alpha, beta, p_truth)
    error = [1 if hypothesis[i]!= hyp_detected[i] else 0 for i in range(nb)]
    return (np.sum(error)/nb)

```

MAP decoder para $H = \{-1, 1\}$

```

import numpy as np
def MAP_decoder(ys,p1,p2,sigma):
    eta=p2/p1
    theta=(sigma**2/2)*np.log(eta)
    if ys> theta:
        return 1
    else:
        return -1

```

ML decoder para $H = \{-3, -1, 1, 3\}$

```

def decoder(ys,p1,p2,sigma):
    if(ys<=-2):
        return -3
    elif(ys<=0):
        return -1
    elif(ys<=2):
        return 1
    else:
        return 3

```