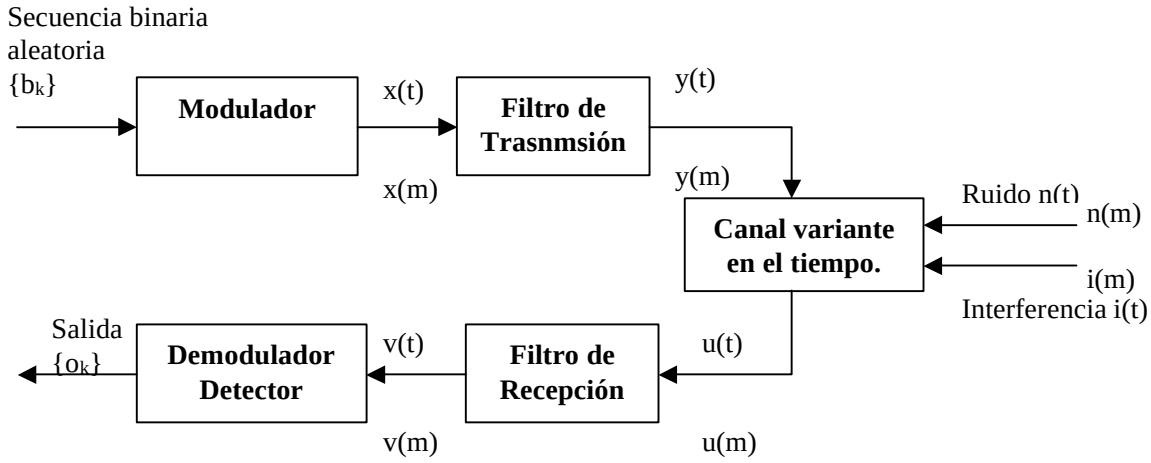
2.3. GENERACION DE NUMERO ALEATORIOS Y PSEUDOALEATORIOS.

2.3.1. Introducción.

La simulación por computador de sistemas de comunicaciones incluye la generación de números aleatorios (o secuencias) con la distribución probabilidad requerida. Estos números o secuencias aleatorios son procesados a través del modelo de bloques funcionales que describen el modelo correspondiente, hasta generar la salida. De esta forma podemos evaluar la probabilidad de error comparando la secuencia a la salida modificada por el modelo que caracteriza la simulación y la secuencia a la entrada del modelo. Un modelo típico en un sistema de comunicaciones es:



En este modelo se aprecian la necesidad de disponer de tres secuencias aleatorias que son b_k, n(m) y i(m), las cuales describen respectivamente a la secuencia de información, al ruido externo el cual esta descrito por las condiciones del entorno y las interferencias que le llegan al sistema de señales no deseadas, pero que guardan parecido con la señal de interés.

Este apartado describe de forma somera la generación de secuencias aleatorias y pseudoaleatorias utilizadas en el modelado del canal de comunicaciones. Las secuencias aleatorias generadas por computador se caracterizan por ser resultado de la aplicación de un algoritmo con un número inicial llamado semilla que proporciona la secuencia aleatoria o mejor dicho pseudoaleatoria ya que la elección de la semilla determina de forma unívoca la secuencia resultante. Por lo tanto podemos decir que el único parámetro que en realidad puede llegar a ser aleatorio puede ser la semilla (‘seed’). A pesar de esto se suele dar la oportunidad, en los programas utilizados en simulaciones, de utilizar la misma semilla determinándose de forma explícita por el usuario para poder comparar resultados cuando de forma sucesiva efectuemos la simulación. A pesar de estas falta de aleatoriedad en el sentido estricto, las secuencias generadas por computador tienen propiedades similares a las que tendría una secuencia aleatoria pura.

Las simulaciones que involucran la generación y el procesado de secuencias aleatorias reciben el nombre de simulaciones de Monte Carlo.

La generación de números aleatorios en computadores comienza con la generación de una secuencia aleatoria que esta distribuida de forma uniforme en el intervalo [0,1]. Si queremos obtener cualquier otra distribución de probabilidad lo único que tenemos que

hacer es aplicar la transformación correspondiente a la secuencia generada de forma uniforme.

2.3.2. Generación de secuencias con distribución uniforme.

Existen múltiples formas de generar números aleatorios con distribución uniforme, pero la forma que generalmente se suele utilizar es aquella que utiliza formulas recursivas que son computacionalmente eficientes. De esta forma se persigue conseguir un compromiso entre la exactitud de la secuencia generada, y la velocidad de computación. Las formulas utilizadas reciben el nombre de generadores de números aleatorios (RNG) y deben caracterizarse por:

- Ser computacionalmente eficientes.
- Tener un periodo largo, a ser posible mayor que la longitud de la simulación. Este periodo es característico de la formula que se utilice y produce que la secuencia aleatoria empiece a repetirse a partir de un determinado punto de la secuencia.
- Tener buenas propiedades de distribución y temporales.
- Producir buenas secuencias parciales ya que la simulación puede solo utilizar secuencias de longitud corta comparada con el periodo de la secuencia aleatoria y necesitamos la secuencia corta mantenga las propiedades que caracterizan a la secuencia aleatoria completa.

Unos RNG's que cumple estas condiciones y que generan secuencias de números aleatorios distribuidos de forma uniforme son los generadores lineales congruenciales (LCG) o métodos de potencia residual los cuales se caracterizan por la formula recursiva:

$$X_{j+1} = (aX_j + c) \bmod(M)$$

donde:

- M es el modulo, M>0, M>>1. M suele ser un número primo o la potencia entera de un número primo. La ecuación producirá números aleatorios entre 1 y M. 0 no esta permitido ya que en la siguiente iteración al ser X_j=0 pararía la secuencia estancada en 0.
- a es el multiplicador, y es escogido de forma cuidadosa entre 0 y M.
- c es el incremento, $0 \leq c < M$.
- X_0 es el valor inicial o semilla y puede ser tomado entre 0 y M

Si M es suficientemente grande y a es escogido de forma apropiada entonces la secuencia de números parecerá prácticamente aleatoria. Para generar números aleatorios en computadores con palabras de 32 bits podemos utilizar un RNG tal como:

$$X_{n+1} = (69069 X_n + 1) \bmod(2^{32})$$

La secuencia uniformemente distribuida se obtendría a partir de la siguiente operación:

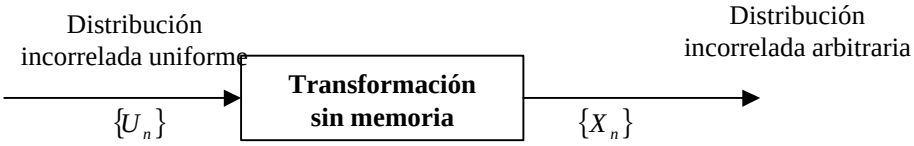
$$U_i = X_i / M$$

La secuencia será periódica y tendrá un periodo de 2^{32}.

Existen algoritmos que están optimizados para determinadas longitudes de palabra y operaciones aritméticas en los computadores. Uno de estos es el algoritmo de Wichmann-Hill. Este algoritmo ha sido ampliamente probado y produce una secuencia con un periodo de aproximadamente $7 \cdot 10^{12}$.

2.3.3. Métodos para la generación de secuencias de números aleatorios decorrelados con distribuciones arbitrarias.

Los métodos para generar números aleatorios de una distribución arbitraria están basados en la transformación u otro uso de la secuencia generada uniforme a partir del RNG.

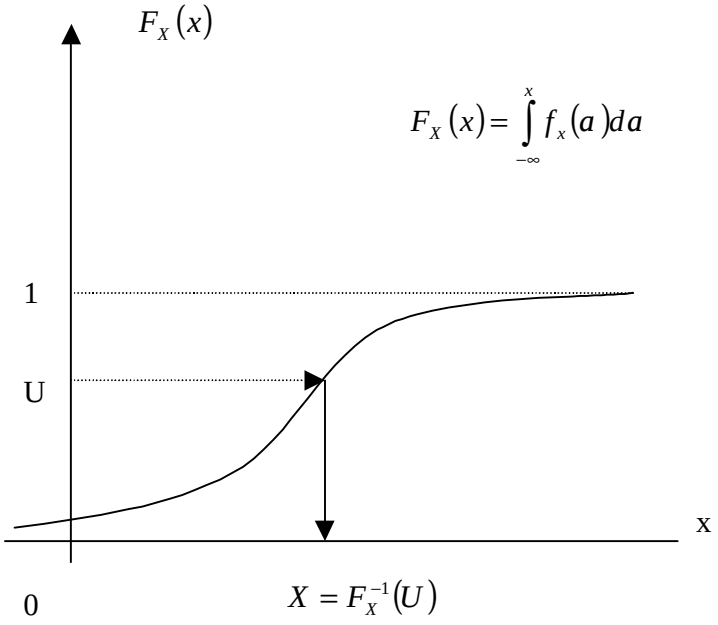


Los métodos más generales para la obtención de números con una distribución determinada son:

- Método de la transformación inversa.
- Método de aceptación-rechazo.

2.3.3.1.Método de la transformación inversa.

Aplicando una simple transformación a una variable aleatoria uniforme U, podemos generar una variable Z con una distribución de probabilidad $f_Z(z)$. Si Z es una variable aleatoria continua con una distribución cumulativa? $F_Z(z)$, entonces $Y= F_Z(z)\sim U$ y por lo tanto $Z=F_Z^{-1}(U)$ transformara la variable aleatoria uniforme a Z.



El siguiente proceso genera una Z arbitraria de la cual conocemos su $F_Z^{-1}(\cdot)$:

1. Generar U mediante los métodos de generación de secuencias aleatorias con distribución uniforme.
2. Obtener la salida $Z=F_Z^{-1}(U)$ para cada valor de la secuencia uniforme.

Este método actúa de forma correcta si $F_Z(z)$ y su inversa están expresadas de forma estricta. Si no es así podemos utilizar algoritmos empíricos para implementar el método de la transformada.

En este caso si Z es una variable aleatoria continua, entonces la distribución es primero cuantizada.

Si $p_1,p_2,...,p_N1$ son las probabilidades de n celdas distribución cuantizada uniforme, podemos aplicar el siguiente algoritmo:

1. Generar U mediante los métodos de generación de secuencias aleatorias con distribución uniforme.
2. Obtener $F_i = \sum_{j=1}^{i} p_j$, i=0,1,2,...,N con $F_0=0$.
3. Encontrar el valor más pequeño de i que satisfaga
   $F_{i-1} < U \le F_{i,}$ i=1,2,...,N
4. La salida es $Z = z_{i-1} + (U - F_{i-1}) / C_i$

El ultimo paso es utilizado para interpolar el valor de Z en el intervalo $[z_{i-1},z_i]$.

Cuando Z es una variable discreta aleatoria con valores $z_1,z_2,...,z_N$, $P(Z=z_i)=p_i$ y $F_i = \sum_{j=1}^{i} p_j,$, entonces los siguientes algoritmos pueden ser usados para generar las muestras de Z.

- Cuando existe un número finito de valores de Z.

0. Poner k=1.
1. Generar U, uniforme en [0,1].
2. Si $U \le F_k$, la salida es $Z=z_k$ y volvemos al paso 0. Si no:
3. Incrementar k=k+1 y volver al paso 2.

- Número incontable de valores de Z:

0. Poner $C=p_1$, B=C y k=1.
1. Generar U, uniforme en [0,1].
2. Si $U \le B$ la salida $Z=z_k$ y volvemos al paso 0.
   Si no:
3. Poner k=k+1.
4. Poner $C=A_{k+1}C$; ($A_{k+1=}$ $p_{k+1/}$ p_k) y B=B+C y volvemos al paso 0.

2.3.3.2.Método de aceptación-rechazo.

Supongamos que queremos generar x de acuerdo con la función de densidad $f_Z(z)$. En ese caso consideramos el siguiente algoritmo para generar X.

1. Generar U_1, distribución uniforme en [0,a].
2. Generar U_2, distribución uniforme en [0,b].
3. Si $U_2<f_X(U_1)$, entonces la salida $X= U_1$; si no rechazar U_1 y volver al paso 1.

2.3.3.3.Generación de variables aleatorias gausianas.

El método mas simple para la generación de números aleatorios con una distribución gausiana estándar es a través del uso del teorema central del limite de acuerdo con la formula:

$$Y = \sum_{K=1}^{12} U(k) - 6.0$$

donde U(k), K=1,2,...,12 es un conjunto de variables independientes y uniformemente distribuidas en el intervalo [0,1]. Según el teorema central del limite el sumatorio debería extenderse entre –inf y inf. El número 12 es normal usarlo ya que

representa un compromiso entre velocidad y exactitud pero no existe razón para limitar k a 12.

Si X e U tienen una distribución N(0,1) en ese caso:

- $R = \sqrt{X^2 + Y^2}$ tiene una distribución Rayleigh
- $q = tan^{-1}\left(\frac{Y}{X}\right)$ tiene una distribución uniforme en [0,2pi].

Este hecho puede ser usado para generar dos muestras de variables gausianas transformando una pareja de variables Rayleigh y uniforme. De igual forma nos va a resultar útil porque podríamos obtener una distribución Rayleigh muy útil en simulación de canales de comunicación a partir de distribuciones gaussianas. La relación anteriormente expresada es:

$$X = R\cos(q) = \left[-2\ln(U_1)\right]^{1/2}\cos(2pU_2)$$
$$Y = Rsin(q) = \left[-2\ln(U_1)\right]^{1/2}sin(2pU_2)$$

que da como resultado dos variables aleatorias gausianas independientes con media 0 y varianza unidad. Este algoritmo es conocido como método de Box-Müller.

2.3.3.4.Generación de secuencias aleatorias independientes.

- Ruido blanco gausiano.

El ruido blanco gausiano tiene una densidad espectral de potencia constante para todas las frecuencias

$$S_{NN}(f) = \frac{N_0}{2} \quad \text{para} -\infty < f < \infty$$

Sin embargo los sistemas reales tienen ancho de bandas finitos, B, y la tasa de muestre en la simulación f_s es escogida tal que sea más grande que 2B.

Si usamos ruido blanco gausiano limitado en banda con densidad espectral de potencia constante sobre el ancho de banda de la simulación $-f_s/2$ a $f_s/2$,

$$S_{NN}(f) = \frac{N_0}{2} \quad \text{para} -f_s/2 < f < f_s/2$$

entonces la respuesta del sistema es la misma poniendo $S_{N_sN_s}(f)$ en lugar de $S_{NN}(f)$.

- Secuencia binaria aleatoria.

Una secuencia binaria aleatoria $\{b_k\}$, b_k=0 o 1, puede ser generada a partir de una secuencia uniforme $\{U_k\}$ por

$$b_k = \begin{cases} 1 & siU_k > p_1 \\ 0 & siU_k \le p_0 \end{cases}$$

donde p_0=P[b_k=0].

Las muestras de una forma de onda binaria pueden ser generada por

$$X(kN+m) = (2b_k - 1)p(mT_s), \quad k = \ldots,-3,-2,-1,0,1,2,3,\ldots$$
$$m = 1,2,\ldots,N$$

donde p(mT_s) son los valores muestreados de la forma de onda digital y la tasa de muestreo es N veces la tasa de bit.

- Secuencias binarias pseudoaleatorias.

Una secuencia binaria aleatoria consiste en una secuencia aleatoria de 0's y 1's que cada ocurre con una probabilidad de ½ Un pseudoruido (PN) o una secuencia pseudoaleatoria es una secuencia binaria periódica con una función de autocorrelación que se asemeja a la función de autocorrelación de una secuencia binaria aleatoria. La secuencia pseudoaleatoria es generada usando un registro de desplazamiento con realimentacion. Aunque determinista , la secuencia pseudoaleatoria tiene características similares a las de las secuencias binarias aleatorias.

La secuencia binaria es desplazada a través del registro de desplazamiento en respuesta a los impulsos de reloj. El contenido del registro es combinado lógicamente para producir la entrada a la primera etapa del registro. El contenido inicial del registro y la lógica de realimentacion determinan el posterior contenido de registro. Este es llamado lineal si la lógica de realimentación consiste enteramente en sumadores de modulo 2. Un registro lineal m etapas genera una secuencia pseudoaleatoria de acuerdo con

$$S_n = c_{m-1}S_{n-1} \oplus c_{m-2}S_{n-2} \oplus \cdots \oplus c_1 S_{n-m+1} \oplus c_0 S_{n-m}$$

donde S_n es el valor de la secuencia en el instante t y los coeficientes c_i son valores binarios 0 o 1.

Ya que el número de distintos estados de un registro de m etapas es 2^m, la secuencia de estados y la secuencia de salida se convierte en periódica con un periodo máximo de $2^{m.}$

La secuencia tendrá $\{S_n\}$ una longitud máxima si y solo si $P_m(\cdot)$ es un polinomio primitivo.

En la siguiente tabla se listan los polinomios primitivos hasta grado 16.

2	[1,2]
3	[1,3]
4	[1,4]
5	[2,5][2,3,4,5][1,2,4,5]
6	[1,6][1,2,5,6][2,3,5,6]
7	[3,7][1,2,3,7][1,2,4,5,6,7][2,3,4,7]
	[1,2,3,4,5,7][2,4,6,7][1,7][1,3,6,7]
	[2,5,6,7]
8	[2,3,4,8][3,5,6,8][1,2,5,6,7,8]
	[1,3,5,8][2,5,6,8][1,5,6,8]
	[1,2,3,4,6,8][1,6,7,8]
9	[4,9][3,4,6,9][4,5,8,9]
	[1,4,8,9][2,3,5,9][1,2,4,5,6,9,]
	[5,6,8,9][1,3,4,6,7,9][2,7,8,9]
10	[3,10][2,3,8,10][3,4,5,6,7,8,9,10]
	[1,2,3,5,6,10][2,3,6,8,9,10][1,3,4,5,6,7,8,10]
11	[2,11][2,5,8,11][2,3,7,11]
	[2,3,5,11][1,3,8,9,10,11]
12	[1,4,6,12][1,2,5,7,8,9,11,12]
	[1,3,4,6,8,10,11,12][1,2,5,10,11,12]
	[2,3,9,12][1,2,4,6,11,12]
13	[1,3,4,13][4,5,7,9,10,13][1,4,7,8,11,13]
	[1,2,3,6,8,9,10,13][5,6,7,8,12,13][1,5,7,8,9,13]
14	[1,6,10,14][3,4,6,7,9,10,14]
	[4,5,6,7,8,9,12,14][1,6,8,14]
	[5,6,9,10,11,12,13,14][1,2,3,4,5,7,8,10,13,14]
15	[1,2,4,5,10,15][1,2,6,7,11,15][1,2,3,6,7,15]
	[1,15][1,5,10,15][1,3,12,15]
16	[1,3,12,16][1,3,6,7,11,12,13,16]
	[1,2,4,6,8,9,10,11,15,16][1,6,8,14]
	[5,6,9,10,11,12,13,14][1,2,3,5,6,7,10,15,16]

Tabla de polinomios primitivos para generadores de secuencias binarias pseudoaleatorias.

2.3.3.5.Generación de secuencias correladas.

Una secuencia correlada de variables aleatorias puede ser generada a partir de una secuencia decorrelada aplicando a esta ultima una transformación lineal apropiada.

En este caso tenemos secuencias correladas ya que han sido modificadas o caracterizadas según el patrón que define la transformación.

Vector de componentes incorreladas

$X = AZ + m$

Vector de componentes correladas