

Proyecto Individual
Estructuras de Datos abstractos y Algoritmos

Marco Antonio Montero Chavarría Carné: A94000

8 de octubre de 2015

Objetivo General

Crear un sistema de comunicación que tome en cuenta tipos de ruido.

Objetivos específicos

- Comprender la teoría relacionada a funciones de distribución probabilísticas.
- Comprender el uso de templates en C++.
- Crear un código capaz de generar suficientes puntos de acorde a una función de distribución elegida.
- Gráficar estas distribuciones
- Obtener un sistema de comunicación con un receptor, emisor y un canal.

1. Justificación

En investigación no siempre las variables con las que se debe trabajar son determinísticas, muchas veces se debe trabajar con variables y procesos aleatorios, también conocidos como procesos estocásticos. Por ende se debe buscar una manera óptima de modelar el comportamiento de estas variables, para poder aplicar herramientas matemáticas conocidas. Una manera eficiente es utilizando distribuciones probabilísticas para poder obtener un modelo matemático del comportamiento que se quiere entender, normalmente se deben realizar múltiples iteraciones y hacer comparaciones para ver cual de las distribuciones se asemeja más al comportamiento deseado. Ejemplo de esto es el viento que no es posible modelarlo con modelos lineales, es decir se ocupa algo más, un modelo no lineal, ya que su movimiento es aleatorio. Por ende una aplicación como esta permitirá al lector una mejor comprensión de modelados probabilísticos.

2. Marco Teórico

En la teoría de la probabilidad y estadística, una función de distribución de probabilidad describe la probabilidad de que una variable aleatoria real X (mayúscula) que esta sujeta a una ley de distribución de probabilidad específica se encuentre en la zona de valores menores o iguales a x (minúscula). Es decir que a cada suceso definido sobre la variable aleatoria se le asigna la probabilidad de que dicho suceso ocurra.

En lenguaje matématico esto se expresa de la siguiente forma:

$$F(x) = P(X \leq x) \quad (1)$$

Existen dos grandes bloques de distribuciones, uno esta definido para variables continuas y otro para variables discretas.

Una distribución para variable discreta es de la forma:

$$F(x) = P(X \leq x) = \sum_{k \rightarrow \infty}^x f(k) \quad (2)$$

Una distribución para una variable continua tiene la siguiente forma:

$$F(x) = P(X \leq x) = \int_{-\infty}^x f(t) dt \quad (3)$$

Para esta investigación se tomaron en cuenta 3 funciones de densidad de probabilidad continuas.[Pee00].

2.1. Funcion de distribución gaussiana

La función de distribución gaussiana o normal es la más utilizada, también es conocida por su forma de campana, fue definida por Moivre y utilizada en muchos de sus estudios por Gauss. Posee la siguiente forma:

$$F(x) = \frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt \quad (4)$$

Con media μ y varianza σ^2 . La media es valor medio de un fenómeno aleatorio, es la cantidad media que se espera de múltiples iteraciones como resultado de un experimento. La varianza es la desviación de la variable aleatorio con respecto a su media.[Pee00]

2.2. Función de distribución rayleigh

$$F(x) = 1 - e^{-\frac{x^2}{2\sigma^2}} \quad (5)$$

Con media $E[X] = \sigma \sqrt{\frac{\pi}{2}}$

Y varianza $E[X]^2 = \frac{4-\pi}{2} \sigma^2$

Esta función se utiliza para modelar las variaciones en la velocidad del tiempo. [H.C70]

2.3. Función de distribución uniforme

Se define como :

$$F(x) = \frac{x - a}{b - a} \quad (6)$$

Esta función es 0 para $x < a$ y 1 para $b \leq x$. La media $E[X] = \frac{a+b}{2}$ y varianza $E[X]^2 = \frac{(b-a)^2}{12}$. Se utiliza más que todo en la teoría para realizar ejemplos y comprender conceptos.[Pee00]

3. Desarrollo

C++ posee templates de la biblioteca stl para generar números aleatorios según una distribución deseada. Para este proyecto se decidió no usarlos y usando el algoritmo de box-muller se logró pasar de una distribución uniforme a una distribución gaussiana y a una rayleigh.

El método box-muller con variación polar permite usar dos variables $U1, U2$ uniformemente distribuidas entre $[-1,1]$ independientes, para crear dos variables aleatorias distribuidas de forma gaussiana, en forma estándar con $\mu = 0$ y $\sigma = 1$. [MB64]

Según el método: $Z1 = U1 * \sqrt{\frac{-2 * \ln w}{w}}$ y $Z2 = U2 * \sqrt{\frac{-2 * \ln w}{w}}$ donde $w = U1 * U1 + U2 * U2$ a partir de esto se puede crear X como variable aleatoria distribuida normalmente con un σ y un μ que se pueden variar. [MB64]

$$X = \sigma * Z + \mu$$

Y Z es igual a $Z1$ y $Z2$ según el método se debe alternar entre las dos para generar la campana de gauss completa.

Además una variación de método box muller llamada método inverso permite generar una variable distribuida rayleigh tomando un $U1$ uniformemente distribuido entre $[0,1]$ para generar X , donde X es igual al radio del círculo que genera el método original de box muller para crear la distribución gaussiana.

Esta define $R = \sigma * \sqrt{-2 * \ln U}$ por lo tanto $X = R$ y obtengo la variable aleatoria distribuida rayleigh. [MB64]

El lector se debe preguntar como se logra crear las distribuciones uniformes. Peculiarmente el método *rand()* de la biblioteca de c++, genera distribuciones uniformes entre un rango $[a,b]$ por lo tanto solo se requiere definirlo entre el rango deseado para crear $U1$ $U2$ y U para el método de rayleigh.

Una vez implementados los métodos para generar variables aleatorios distribuidos rayleigh o gaussiana, se debe crear un sistema de comunicación con un receptor, un emisor y un canal o medio. Para este proyecto se eligió como medio el aire por lo tanto este posee ruido gaussiano y además para hacer el proyecto más interesantes se le añadió otro tipo de ruido el rayleigh que no es precisamente una implementación real es más solo una prueba.

El canal se modeló con el template de pila que tiene la biblioteca stl y los métodos de enviar y recibir apilan y desapilan el mensaje enviado. El canal modifica lo que esta en esa pila según alguna de las dos funciones de distribución.

Por último se añadió un pequeño código para dibujar la función de distribución definida para el canal, está es un simple histograma tipo bucket que generará un * por cada vez que uno de los enteros que el vector bin tenga en un índice.

4. Conclusión

El lector probablemente pensará en la utilidad del proyecto. Básicamente en el tiempo pre era de las comunicación inalámbricas, modelar el ruido en el aire era esencial para crear métodos de modulación y codificación eficientes. Un proyecto como este les permitiría hacer un análisis básicos y comprobar si sus protocolos funcionan o funcionaban de manera correcta y si la salida en el receptor es la salida esperada.

Por la parte de c++ el proyecto ayudo a mejorar el conocimiento del lenguaje y el uso de paradigmas de programación, además el uso de templates y preparar para un proyecto más denso como la creación de una librería.

Referencias

- [H.C70] H.Crammer. *Random Variables and Probability Distributions*. 1970.
- [MB64] G. Marsaglia and T. A. Bray. *A Convenient Method for Generating Normal Variables*. 1964.
- [Pee00] Peyton Z. Peebles. *Principios de Probabilidad: Variables y Señales Aleatorias-4edición*. 2000.