

MODULACIÓN QPSK EN EL USRP.

Áreas técnicas: telecomunicaciones, modulación digital, radio reconfigurable.

Burnes R. Susana, Espinoza Hernández Jesús, Rodríguez J. Pedro

Facultad de Ciencias Químicas e Ingeniería
Universidad Autónoma de Baja California,
Calzada Tecnológico 14418, Mesa de Otay, Tijuana, B. C., CP 22390
(664) 6821033-5800
correo-e: sburnes@uabc.edu.mx, chuyespinoza@gmail.com, rodriguezp@uabc.edu.mx

Ortega Corral César
Tecnologías de la Información y Comunicación
Universidad Tecnológica de Tijuana
Km. 10 Carretera Libre Tijuana-Tecate, Fracc. El Refugio.
Quintas Campestre. Tijuana, B.C., C.P. 22650
Tel +(664) 969 9700 ext 4789 FAX +(664) 969 9700 ext 4806
correo-e: cesar.ortega@utijuana.edu.mx

1. RESUMEN.

En este trabajo se analiza y desarrolla la modulación QPSK en un USRP en base al radio reconfigurable por software. Como el desarrollo de una herramienta educacional para la enseñanza de los principios de la comunicación de señales digitales con modulación multinivel. La caracterización se lleva a cabo utilizando el ambiente Linux y el software *GNURadio*. Como resultados preliminares se generan las gráficas BER, la gráfica de constelación de fases para el análisis de la transmisión, así como su espectro y el diagrama de ojo para la visualización de la calidad de la señal recibida.

2. INTRODUCCIÓN

La modulación QPSK es derivada de la modulación PSK (*Phase Shift Keying*). La modulación PSK utiliza una cantidad finita de cambios de fase derivados de una lista de patrones de bits denominada *alfabeto*, determinado por:

$$M = 2^b \quad (1.1)$$

donde M es el total de elementos del conjunto y b es la cantidad de bits asignados a cada elemento del conjunto. El tamaño del alfabeto lo determina la aplicación e implica consideraciones en el espacio entre símbolo en el plano complejo y la probabilidad de error de bit, y que un alfabeto muy grande aumenta la probabilidad de errores pero a su vez aumenta la eficiencia espectral de la transmisión.

La modulación PSK tiene la siguiente expresión general:

$$s(t) = \sqrt{\frac{2E}{T}} \cos[\omega_0 t + \phi_i(t)] \quad (1.2)$$

donde el término $\phi_i(t)$ representa a un conjunto de cambios discretos de fase en la señal. Estos valores son típicamente dados por la siguiente expresión:

$$\phi_i(t) = \frac{2\pi i}{M}; \quad i = 1, \dots, M \quad (1.3)$$

La forma más simple de PSK es la conocida BPSK (B de *Binary*, $b = 1$) correspondiente a un bit por cada símbolo. La modulación QPSK (Q del inglés *Quadrature*, o cuadratura) es una variación de PSK que utiliza cuatro puntos en el plano complejo y por lo tanto, se considera una señal bidimensional ya que también tiene proyección en el eje vertical. Este esquema agrupa los bits en pares por cada símbolo, haciendo esta modulación espectralmente más eficiente que BPSK, ya que se puede transmitir más información utilizando el mismo ancho de banda. La constelación de este esquema de modulación se muestra en la figura 1.

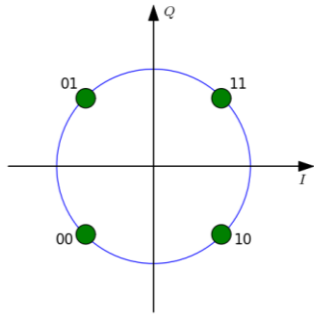


Figura 1 Diagrama de constelación de la modulación QPSK.

3. EL SISTEMA DE RADIO PERIFÉRICO RECONFIGURABLE POR SOFTWARE

El sistema USRP (*Universal Software Radio Peripheral* por sus siglas en inglés) es un dispositivo que permite diseñar radios mediante programación. Y es solamente una parte de la estructura completa de lo que se conoce como un SDR (*Software Defined Radio* por sus siglas en inglés). Su función primaria es actuar como la sección de banda base y frecuencia intermedia de un sistema de comunicaciones especificado en [5].

La figura 2 muestra los componentes que forman un sistema de comunicaciones implementado con el USRP. Todo el procesamiento en banda base es llevado a cabo en una PC x86. Esta PC contiene el programa Python que realiza todo el procesamiento digital con la señal o señales que se desean trabajar. Esto se lleva a cabo utilizando las herramientas proporcionadas por *GNURadio*.

El USRP utiliza el puerto USB para toda comunicación con la PC. El tipo de conexión es USB 2.0 de alta velocidad [10] y es llevado a cabo por el controlador Cypress FX 2.

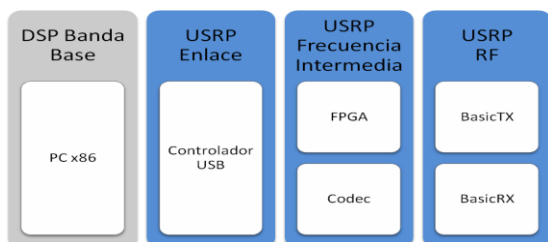


Figura 2 Diagrama a bloques de un sistema de comunicaciones basado en el USRP

El FPGA es un Altera Cyclone EP1C12 y es el encargado de realizar operaciones de alta velocidad, así como también reducir la tasa de datos en una más adecuada para que la señal pueda ser transferida por el puerto USB. Para la etapa de RX, la configuración básica contiene dos DDC (*Digital Down Converters*) implementados con filtros Peine Integradores en Cascada (*Cascaded Integrator Comb o CIC*) de 4 etapas y filtros de media banda (*Half-band o HB*) de orden 30.

Para la etapa de TX (transmisión) el proceso es el inverso. El FPGA contiene filtros CIC interpoladores que se encargan de interpolar la señal, elevarla a la frecuencia intermedia y después enviarla a los DACs.

El ADC y el DAC son implementados por el chip AD9862 de Analog Devices. El ADC muestrea la señal a 64MB/s dando así un ancho de banda de 32MB/s por criterio de Nyquist. Los datos son enviados al FPGA en paquetes de 16 bits en formato complejo (I y Q). Esto da un ancho de banda efectivo de 8Mhz (32Mhz/4bytes) entre el USRP y la PC. El DAC tiene una velocidad de muestro de 128MS/s con una resolución de 14 bits.

El sistema contiene 4 conectores para instalar diversas tarjetas auxiliares que trabajan con diversos rangos de frecuencia. Para este trabajo se utilizaron las tarjetas LFTX y LFRX. Estas permiten trabajar con un rango de frecuencia de 0-30Mhz.

4. DESARROLLO

Utilizando los programas *benchmark_tx.py* y *benchmark_rx.py* de *GNURadio*, se realiza una transmisión utilizando la modulación QPSK. La programación consiste en diseñar una gráfica que representa el flujo de información. La gráfica también representa la cadena de RF la cual forma parte del radio definido por software. La estructura del transmisor QPSK se muestra en la figura 3:

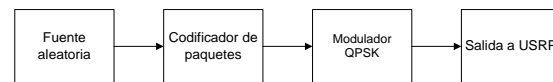


Figura 3. Diagrama a bloques del transmisor QPSK.

La fuente de información es un generador de bits aleatorios. Estos entran a un codificador de paquetes, el cual genera los paquetes con la siguiente estructura:

<código de acceso> <longitud> <información>
<crc32>

El código de acceso es una serie de unos y ceros que se utilizan para identificar el paquete. Este puede ser generado automáticamente o el usuario puede especificar uno. La longitud es la longitud total del paquete. El crc32 es un código generado a partir de la información. Este código es único y se utiliza para identificar errores en la información. Una vez que el paquete es generado entonces es procesado por el modulador QPSK. La estructura del modulador se muestra en la figura 4:

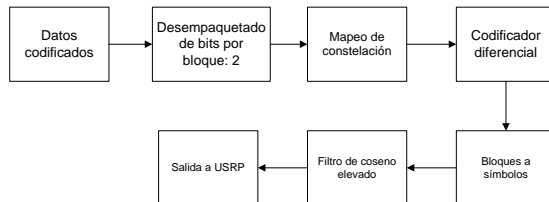


Figura 4. Modulador QPSK en GNURadio

Los bits de cada byte que va entrando al sistema son descompuestos en grupos de dos. Luego son enviados a un codificador grey de acuerdo al tamaño M (en este caso 4). Antes de ser clasificados en su constelación, los bits son codificados utilizando un codificador diferencial, en efecto haciendo esto un sistema de modulación DQPSK. Por último, los símbolos son introducidos a un filtro de coseno elevado para poder reducir la cantidad de ancho de banda necesaria para poder transmitirlos. El filtro está típicamente configurado para utilizar un factor de exceso de ancho de banda β de 0.50 aunque el usuario puede utilizar el valor que sea (entre 0 y 1) dependiendo de los requerimientos de su sistema.

La estructura del receptor es un poco más compleja que la del transmisor ya que es necesario extraer la frecuencia y el reloj de la señal transmitida para poder sincronizarse con el reloj del transmisor. Si esto no se logra entonces no será posible recuperar la información que se transmitió. El diagrama a bloques del receptor se muestra en la figura 5:

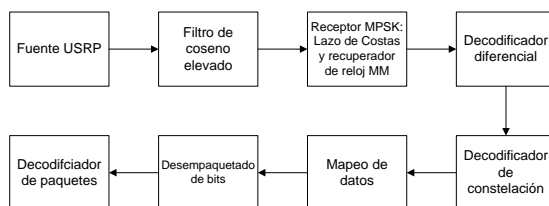


Figura 5. Estructura del demodulador DQPSK

La información primero pasa por un filtro acoplado de coseno elevado configurado igual que el del transmisor. Después entra a la etapa de recuperación de la frecuencia y del reloj. La frecuencia es recuperada utilizando un PLL llamado lazo de costas y luego el reloj es extraído de la señal utilizando el sincronizador rastreador de error en tiempo discreto de Muller y Mueller. La señal luego es decodificada con un decodificador diferencial y después los símbolos son decodificados. El resultado es una serie de bits que representan una aproximación a la secuencia original transmitida. Posteriormente los paquetes son procesados utilizando un correlador de códigos de acceso, un verificador de crc32 y un bloque “framer” que se encarga de extraer la información.

5. RESULTADOS PRELIMINARES.

Analizando los datos arrojados por el programa utilizando las herramientas que *GNURadio* proporciona, podemos observar el rendimiento de los bloques anteriores a través del diagrama de constelación. Transmitiendo 50kB de datos aleatorios a una tasa de bits de 100kb/s y un muestreo de 2 muestras por símbolo se obtuvieron los siguientes resultados:

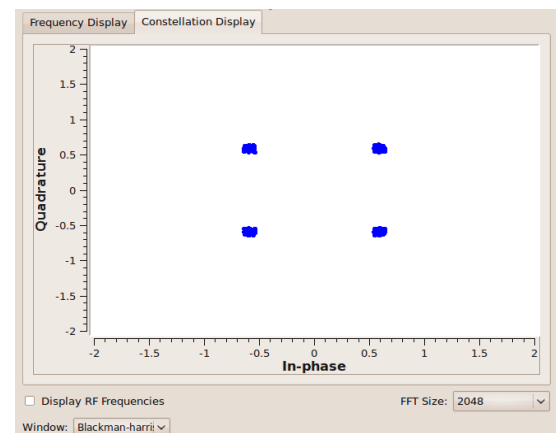


Figura 6. Constelación obtenida después de la recuperación de la señal

Utilizando las herramientas de visualización de GNURadio, se puede observar la constelación recibida. La figura 6 muestra que el canal introduce una cantidad mínima de ruido y se pueden decodificar correctamente todos los símbolos recibidos.

En la figura 7 se observa el espectro de la señal centrado en 0Hz con un ancho de banda de 100kHz, confirmando la sintonización correcta del

receptor. La razón de por qué la señal está centrada en 0Hz es porque una vez entrando a la PC los datos se encuentran en banda base debido a que la portadora es eliminada cuando la señal pasa por el convertidor digital hacia abajo que se encuentra en el USRP.

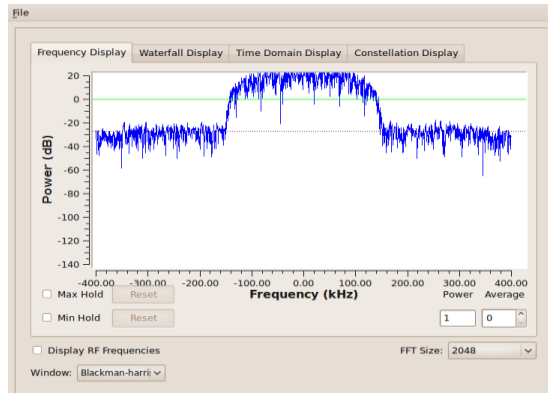


Figura 7. Espectro de la señal recibida.

La calidad de recepción se puede visualizar utilizando el diagrama ojo como se muestra en la figura 8. Los efectos de variar el parámetro β del filtro RRC se muestran en las figuras 8 y 9.

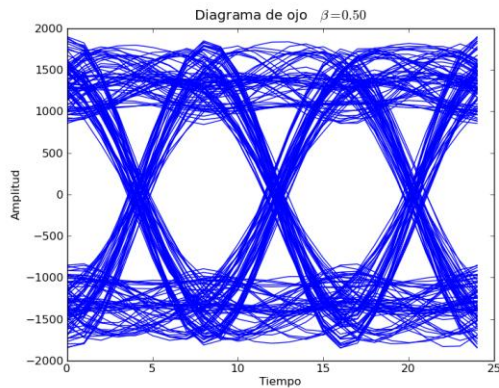


Figura 8. Diagrama de ojo con $\beta = 0.5$

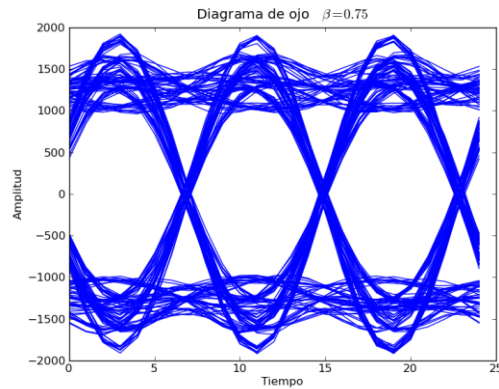
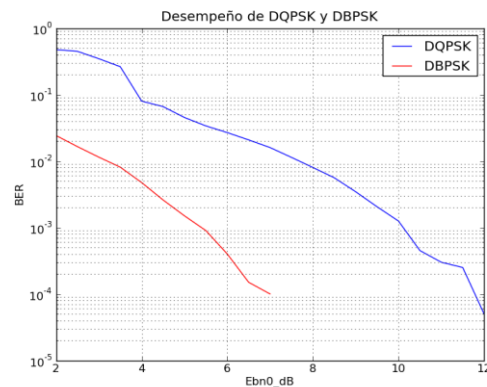


Figura 9. Diagrama de ojo con $\beta = 0.75$

Debido a que el canal utilizado introduce una mínima cantidad de ruido, se puede observar que la apertura del ojo es buena y permite un muestreo del centro óptimo para cada símbolo.

El rendimiento del algoritmo del demodulador DQPSK fue evaluado utilizando un canal AWGN (*Additive White Gaussian Noise*) simulado. En la figura 10 se muestra una comparación contra BPSK utilizando también su algoritmo ofrecido por *GNURadio*. Se puede observar como DQPSK requiere más energía de bits para lograr una menor probabilidad de error que DBPSK, ya que DQPSK trabaja con el doble de bits por símbolo que DBPSK.



6. CONCLUSIONES

Se pudo observar que *GNURadio* tiene la capacidad para estudiar más a fondo las modulaciones digitales, así como sus aplicaciones. Una desventaja que se observó fue que existe muy poca documentación concreta para poder dominar todas sus herramientas. Se espera que el usuario tenga altos conocimientos de programación en Python y sepa manejar las herramientas o

comandos de Linux. A pesar de esto, su potencial para el estudio y desarrollo de aplicaciones en el área de telecomunicaciones es considerable gracias a su flexibilidad de poder desarrollar bloques y diagramas que puedan hacer el trabajo de todo un sistema de comunicaciones con el mínimo requerimiento de hardware.

7. TRABAJO FUTURO

Con la familiarización del funcionamiento de *GNURadio* y el USRP, se espera trabajar en una aplicación como la transmisión de voz a través de un canal inalámbrico utilizando las diversas modulaciones digitales y analizar su rendimiento. También se espera trabajar en el análisis de una aplicación *GNURadio* bajo un ambiente en tiempo real y comparar su desempeño con un sistema operativo normal.

8. BIBLIOGRAFIA

[1] Abendroth, S. (2003). *A Software Defined Radio Front-End Implementation*. *IEEE DSPEnabled Radio Conference*, (págs. 1-4). Livingston, Scotland.

[2] Collins, R. JTRS Cluster 1. *Software-defined radio*. Rockwell Collins Communications Sector, Cedar Rapids, Iowa.

[3] Donadio, M. P. (18 de Julio de 2000). *CIC Filter Introduction*.

[4] Ettus Research LLC. (20 de Abril de 2009). *Ettus Research LLC*. Recuperado el 18 de Mayo de 2009, de Ettus Research LLC: <http://www.ettus.com/>

[5] GNU Radio. (18 de Mayo de 2009). *Gnu Radio*. Obtenido de Gnu Radio - The Gnu Software Radio: <http://gnuradio.org/trac/wiki>

[6] Texas Instruments. (Noviembre de 2008). *Block Diagram (SBD) - Software Defined Radio* - TI.com. Recuperado el 18 de Mayo de 2009, de Texas Instruments: <http://focus.ti.com/docs/solution/folders/print/357.html>

[7] Mitola, J. (1992). *The Software Radio*. IEEE National Telesystems Conference.

[8] Santamarina, J. C. (2005) , *Discrete Signals and Inverse Problems* (págs. 3-4). Wiley.

[9] Sklar, B. (2001). *Digital Communications Fundamentals and Applications*. Prentice Hall. California, EUA.

[10] Universal Implementers Forum, Inc. (27 de Abril de 2000). *Universal Serial Bus 2.0 Specification*.