

Diseño e Implementación de un Sistema de Comunicación de Voz en Tiempo Real Utilizando Tecnología SDR en dispositivos USRP E110.

M. D. Torres, C. S. Vaca, C. D. Altamirano and C. P. Bernal
Universidad de las Fuerzas Armadas - ESPE
Electrónica y Telecomunicaciones
Sangolquí - Ecuador

Abstract—The SDR devices along with GNU Radio Software tool have taken to a new level in designing radio communications systems. This technology uses blocks-processing software that substitutes everything that is far implemented at the physical layer hardware. By designing a software radio system, this paper presents the implementation of a real time voice Tx/Rx using an embedded GMSK modulation in USRP E110 devices, the system performance was evaluated from an cualitative and cuantitative viewpoint using Mean Opinion Score (MOS) and Modified Bark Spectral Distortion (MBSD) algorithm, another tests were also performed.

Palabras Clave: USRP E110, GNU Radio, Modulación GMSK.

I. INTRODUCCIÓN

La tecnología de Radio Definido por Software SDR constituye una clase de radios reconfigurables/reprogramables cuyas características de la capa física pueden ser significativamente modificadas mediante cambios en software [1], SDR define varias características de radio en banda base y posee cierto nivel de control en *software* en las operaciones de *Front-End*.

El *hardware USRP (Universal Software Radio Peripheral)* está constituido por una tarjeta madre *Motherboard*[2] que puede ser conectada con una variedad de tarjetas hijas *Daughterboards*[3] las cuales proveen varias interfaces, desde filtros análogos hasta conversiones complejas para operaciones de modulación y demodulación en varias bandas de frecuencia.

Sus bloques funcionales pueden ser cambiados en tiempo real y sus parámetros de operación pueden ser ajustados bien por un operador humano o por un proceso automatizado.

II. CONSIDERACIONES BÁSICAS

Los *USRP* a ser usados son los dispositivos *USRP E110* cuyo resumen de características técnicas son mostradas en la Tabla I.

	Características USRP E110
Procesador	ARM Cortex-A8 8MS/s
DSP C64	Para operaciones embebidas
FPGA	Xilinx Spartan 3A-DSP 3400
ADC	Dual 64MS/s
DAC	Dual 128MS/s
F. Operación	DC - 6GHz

Chart I

CARACTERÍSTICAS USRP E110

Un factor a tomar en cuenta es el ancho de banda que se requiera. Esta capacidad depende del *USRP* y que la *daughterboard* esté acoplada con la resolución de los datos que son transmitidos por la *motherboard*.

Por lo que se hizo uso de tarjetas *daughterboard RFX1800 1.5-2.1 GHz Rx/Tx* [4], las cuales operan en la banda 1900 MHz, con una salida de potencia de 100 mW, figura de ruido de 8dB y ancho de banda de 40MHz.

El diseño del sistema de comunicaciones presenta varios obstáculos a nivel de *hardware* que limitarán ciertos parámetros en el momento de la implementación. Al ser un equipo de radio embebido todas las operaciones de procesamiento se realizan dentro del mismo, por lo que se requieren tasas de muestreo y transmisión distintas a las que se colocaría por ejemplo en el *USRP N210*, el cual trabaja con una *PC-host* que realiza las operaciones de procesamiento digital de la señal.

Las altas tasas de muestreo y procesamiento se realizan en la FPGA integrada dentro del dispositivo, mientras que los procesos con baja tasa de muestreo ocurren en la tarjeta micro-SD que contiene una distribución de *Linux Angstrom* sobre la cual se encuentra instalada la plataforma de diseño.

Un *SDR* puede ser desarrollado sobre varias plataformas *software* como: *GNU Radio*, *Simulink*, *LabView* y demás aplicaciones de terceros. Particularmente *GNU Radio* permite la creación de bloques de procesamiento para la implementación de sistemas *SDR*. Mediante esta herramienta es posible la transmisión y recepción de ráfagas de datos digitales, los cuales son muestreados por el *hardware USRP* [5].

En general *GNU Radio* trabaja en tres niveles: *GRC (GNU Radio Companion)* [6] es el nivel en donde se encuentran las herramientas gráficas de diseño, *Python* [7] que compone los bloques y aplicaciones que corresponden a las etapas de procesamiento de la señal, y finalmente el nivel de *C++* que contiene las funciones de bajo nivel.

GRC provee una ayuda gráfica para determinar el tipo de dato que ingresa y sale de cada bloque de procesamiento [8], como puede verse en la Figura 1.

Dadas las características de una plataforma SDR y sus componentes, existen limitantes para cada plataforma y varios parámetros de diseño pueden afectar el desempeño del modelo resultante. El presente proyecto busca lograr un procesamiento

Types	
Color Mapping	Complex Float 64
	Complex Float 32
	Complex Integer 64
	Complex Integer 32
	Complex Integer 16
	Complex Integer 8
	Float 64
	Float 32
	Integer 64
	Integer 32
	Integer 16
	Integer 8
	Message Queue
	Async Message
	Wildcard

Figure 1. Tipos de dato en GNU Radio

en tiempo real de voz. El conocer la arquitectura de la plataforma SDR también es importante para determinar que tipo de operaciones deben realizarse para comunicar un dispositivo con otro.

Las consideraciones a tomar en cuenta cuando se desarrollan sistemas de comunicaciones basadas en plataformas SDR incluyen lo siguiente [4]:

- Integración entre la capa física y el *software* implementado en un procesador embebido requiere una óptima interdependencia entre todas las capas cuando se precisa procesamiento en tiempo real.
- Asegurar el suficiente AB para la transmisión.
- Muchos sistemas de comunicación digital poseen una arquitectura centralizada para controlar las operaciones sobre todo el sistema.
- Realizar experimentos controlados en distintos ambientes (multiobstáculos, multicaminos, etc.) para analizar la confiabilidad de un sistema SDR en particular.
- Reconfiguración y rápida adaptación por medio de un flujo en el diseño de software para la descripción de algoritmos y protocolos.

En resumen, en el diseño de un sistema de comunicaciones con tecnología SDR tanto los elementos fijos de *hardware* como los bloques de procesamiento configurables de *software* deben trabajar en concordancia para lograr que el diseño funcione adecuadamente.

Un sistema digital de comunicaciones contiene una serie de bloques de procesamiento como los de la Figura 2 indispensables para garantizar que la información pueda ser transmitida y recibida de manera óptima. El presente proyecto se enfoca en la transmisión y recepción de señales de voz utilizando *GMSK*.

La voz humana posee la mayoría de su información en bajas frecuencias, comprendidas desde los 50 Hz que es la frecuencia fundamental (*pitch*) hasta los 3500 Hz [9], el espectro audible para un aparato auditivo joven y sano va desde los 20Hz hasta 20kHz [10].

Es conveniente utilizar una tasa de muestreo lo mas grande posible para asegurar la claridad y comprensión de la voz, y el cumplimiento del criterio de *Nyquist* [11], pero no tan

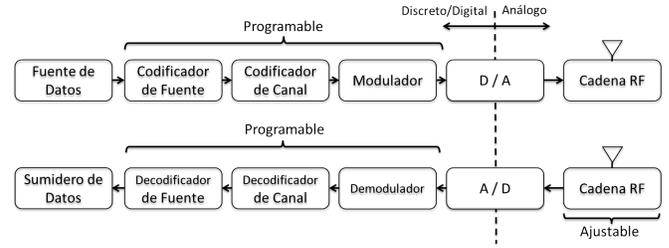


Figure 2. Esquema Básico de un Sistema de Comunicaciones Digital

grande como para comprometer la capacidad de procesamiento del sistema embebido del dispositivo *USRP E110*. El presente diseño utiliza 8kHz como tasa de muestreo.

III. DISEÑO DEL SISTEMA

GMSK es el acrónimo de *Gaussian Minimum Shift Keying*. Es un tipo de modulación binaria simple derivada de *MSK Minimum Shift Keying* con una desviación máxima igual a la mitad de la tasa de bits. *GMSK* es entonces un esquema de modulación de fase continua. En *GMSK* los bits a ser enviados son transformados a su vez en pulsos gaussianos (señales de forma acampanada) [12] mediante un filtro gaussiano [13].

Muchas de las veces la duración de un pulso gaussiano supera a la de un bit, y esto origina interferencia intersimbólica (*ISI*). El grado de la *ISI* es determinado por el producto del ancho de banda del filtro gaussiano (*B*) y la duración de un bit (*T*). Este valor se conoce como *BT* [14] y viene dado por la expresión:

$$BT = \frac{f_{-3dB}}{BitRate} \quad (1)$$

donde f_{-3dB} es la frecuencia de corte del filtro gaussiano a -3dB y *BitRate* es la Tasa o velocidad de Bits. *BT* tiene un valor de 0.3 para *GSM (Global System for Mobile)* y 0.5 para *DECT (Digital Enhanced Cordless Telecommunications)* [15].

La probabilidad de error P_e para *GMSK* viene dado por:

$$P_e = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{\alpha E_b}{2N_0}} \right) \quad (2)$$

Donde α es un valor constante que depende del valor *BT* de filtro gaussiano, cuando *BT*=0.3, α es 0.9 [16].

La sección de Tx inicia con un muestreador de audio *Audio Source* que utiliza un módulo de 8 bits y una frecuencia de 8 kHz para la digitalización de la señal de voz. Un filtro pasabajos permite discriminar frecuencias indeseadas a fin de reducir la cantidad de muestras enviadas y lograr una mejor calidad de voz.

El bloque *G721 Audio Encoder*, es un algoritmo de compresión de voz *ADPCM (Adaptive Differential Pulse Code Modulation)* de 32kbps cuya tasa de muestreo es de 8kHz y tasa de compresión es 2:1 [17]. Este codificador de voz permite reducir en gran medida la carga computacional puesto que las muestras a ser moduladas corresponderán únicamente a la frecuencia fundamental de la onda de voz, mismo proceso que será revertido en la sección Rx.

El bloque *Packet Encoder* recibe el flujo entrante de datos y lo transforma en segmentos de bits. El número de bits en cada segmento debe ser ajustado para que su valor coincida con el del modulador. El modulador utilizado es *GMSK*, que corresponde a una modulación *MSK* básica. Este bloque codifica un bit en un símbolo, significa que cada bit en la muestra entrante de ocho bits deberá ser enviada hacia la salida un bit a la vez, esto cambia la frecuencia de muestreo desde la entrada hacia la salida del bloque.

Finalmente para comunicar tanto el Tx y Rx se encuentran los bloques *UHD: USRP Sink* y *UHD: USRP Source*. En este bloque se necesita especificar la tasa de muestreo y la frecuencia de la portadora. Para un muestreo compatible con el diseño previo se utiliza la ecuación 3.

$$R_{S_{USRP}} = R_{S_a} \times D_F \times c \times d \quad (3)$$

$$SR = 8kHz \times 8 \frac{\text{bits}}{\text{muestra}} \times 2 \frac{\text{muestras}}{\text{símbolo}} * 1 \frac{\text{símbolo}}{\text{bit}} \quad (4)$$

donde los parámetros:

R_{S_a} : Tasa de muestreo del bloque Audio Source.

D_F : Se usa un flujo de datos de 8 bits.

c : Muestras por símbolo del bloque modulador *GMSK*.

d : Inverso del mapeo de bits por símbolo del bloque modulador *GMSK*.

La $R_{S_{USRP}}$ calculada es 128 kHz, y para mejorar el desempeño se ha optado por una tasa de muestreo de 256 kHz. La Figura 3. ilustra los bloques de procesamiento que se han detallado en los párrafos anteriores.

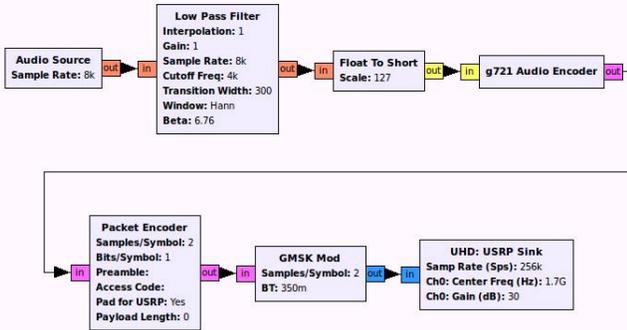


Figure 3. Diagrama de Bloques del Transmisor

El Rx constituye el proceso inverso al Tx. Inicia con un bloque *Costas Loop* para la recuperación de fase de la portadora [18]. Este bloque es imprescindible en las comunicaciones inalámbricas dado que incrementa la sensibilidad del receptor cuando se utilizan modulaciones del tipo *GMSK* [19].

El bloque *Costas Loop* presenta dos parámetros: el *loop_bw* que establece el ancho de banda del filtro de segundo orden y está en el rango de $\pi/200$ a $2\pi/100$ [20]. El orden es 2 para *GMSK* [21].

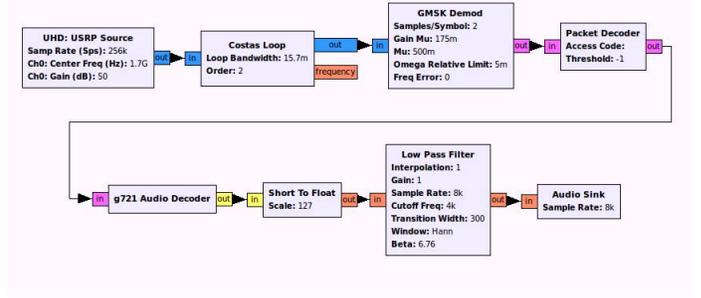


Figure 4. Diagrama de Bloques del Receptor

Si la amplitud de la señal recibida es demasiado débil puede agregarse un bloque *Multiply Const* a continuación del filtro pasabajos para elevar la ganancia de la señal Rx, sin embargo este proceso implica que factores como el ruido producido por el medio de propagación y el ruido inducido por los conectores en los puertos de audio también será incrementada en el mismo porcentaje que la señal de voz debido a que este tipo de ruido también se encuentra en el rango que el bloque del filtro no discrimina.

Es conveniente modificar los valores correspondientes a los archivos de configuración de audio contenidos en: `/usr/etc/gnuradio/conf.d` para lograr mayor fidelidad en los dispositivos de audio. Para el desarrollo del presente proyecto se ha elegido la configuración del *driver* de audio *ALSA* (*Advanced Linux Sound Architecture*), y se ha manipulado los valores correspondientes al tiempo de período *period_time* y al número de período *n_time* para lograr un cambio en el tiempo de *buffering* total. Los valores *period_time*=0.10 y *nperiods*=0.001 fueron establecidos.

IV. PRUEBAS Y RESULTADOS

El sistema fue probado en un ambiente de laboratorio siendo capaz de mantener la comunicación a través de una distancia de alrededor de 6 metros con una división de cristal y madera como obstáculo. Se ha utilizado un micrófono activo en Tx y parlantes estándar de PC en la sección Rx.

GNU Radio que ofrece varios bloques de interfaz que permiten la visualización de la relación señal-ruido, localizados en el módulo `gnuradio.qtgui` y deben ser importados dentro de *Python* con el comando `from gnuradio.qtgui import qtgui`.

La Figura 5 muestra la salida del bloque modulador *GMSK*, observándose la señal modulada en el dominio del tiempo con sus respectivos cambios de fase característicos de la modulación. La Figura 6 muestra la constelación generada en el modulador *GMSK* donde se observa la rotación de símbolos.

La Figura 7 corresponde a mediciones de potencia Rx con un analizador de espectros externo al entorno GNU Radio. El valor de la potencia medida a una distancia de 6 m es de -40 dB aproximadamente con un ancho de banda de alrededor de 121 kHz. Esta relación señal-ruido continuará decreciendo progresivamente en tanto los equipos se encuentren más alejados

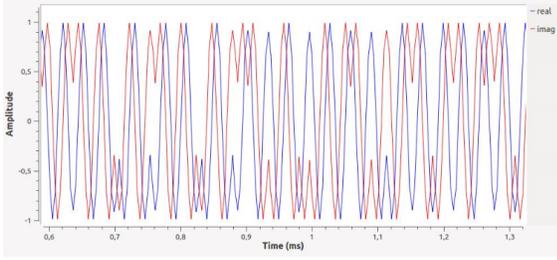


Figure 5. Transmisión en el dominio del tiempo

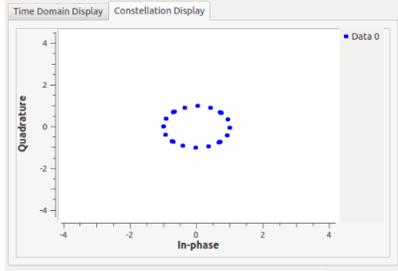


Figure 6. Constelación GMSK

hasta a cierta distancia llegar a igualar al ruido, sin embargo el ancho de banda de la portadora se mantendrá estático.

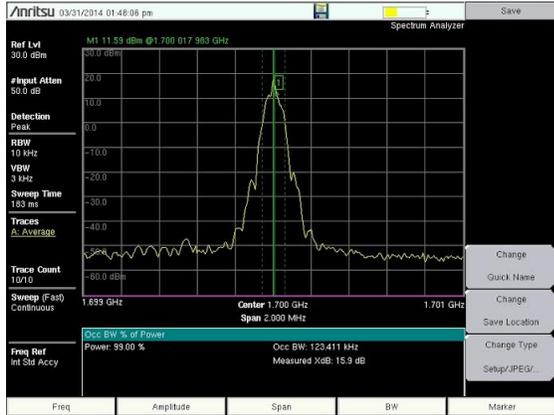


Figure 7. Potencia de Recepción

GMSK posee una eficiencia espectral ν menor a 0.7 bps/Hz, que viene dado por la expresión [22] [16]:

$$\nu = \frac{R_b}{W} = \frac{\log_2(A)}{1 + \alpha}, \quad (5)$$

En donde A representa el alfabeto de la constelación y α corresponde al factor asociado al BT. La eficiencia espectral obtenida para el sistema diseñado es 0.5263 bps/Hz, lo cual contrasta con el valor teórico. El ancho de banda GMSK en banda base W se halla a partir de:

$$W = \frac{R_b}{\nu} = \frac{32000bps}{0.5263} = 60.80kHz \quad (6)$$

Donde R_b es la tasa de transmisión o tasa de bits (bps) entrante al bloque de modulación (salida del bloque G.721

audio encoder). Recordar que en banda base W corresponde a la mitad del ancho de banda en pasa banda, que es valor observado en el analizador de espectros. Por tanto, el ancho de banda total para este diseño es:

$$B = 2W = 2 \times 60.80kHz = 121.6kHz \quad (7)$$

Para calcular el retraso se realizó mediciones en función del número de muestras. Existen un total de 8000 muestras por segundo. Cada muestra tiene una duración D de:

$$D = \frac{1}{f_{muestreo}} = \frac{1}{8000} = 0.125ms \quad (8)$$

Si cada muestra dura 0.125 ms y existen 330 muestras de retraso entre Tx y Rx (medidas en Matlab), el retardo total será el producto de ambos: 41.25 ms.

Para el análisis de la calidad de voz se usó el algoritmo *MBS* (*Modified Bark Spectral Distortion*) [23] como una medida de desempeño que estima la distorsión del habla teniendo en cuenta el umbral de enmascaramiento de ruido, con el fin de incluir sólo distorsiones en su cálculo, para ello se requiere que ambas muestras se encuentren sincronizadas, por lo que se hizo uso de funciones de correlación para determinar el punto de mayor sincronía entre las señales Tx y Rx obteniéndose los resultados mostrados en la Figura 8.

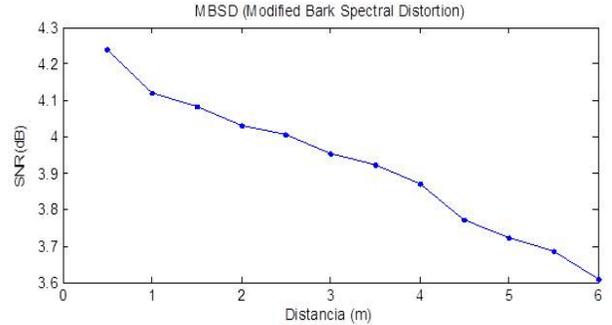


Figure 8. Resultados MBS a varias distancias.

Como es de esperarse, los resultados de *MBS* indican que el desempeño del sistema disminuirá en tanto la distancia entre dispositivos *USRP* se incremente, hasta eventualmente en algún punto determinado el mensaje no sea claro o se haya perdido la comunicación.

También se realizaron encuestas de *Mean Opinion Score* MOS para valorar el desempeño del sistema por parte del usuario y tests de inteligibilidad segmental e inteligibilidad en contexto [24] para determinar con mayor precisión las composiciones silábicas en las que el sistema presenta falencias, las frases utilizadas para los tests de inteligibilidad en contexto hicieron uso de combinaciones vocálicas de baja potencia, como por ejemplo: “No des patadas a los rosales” y “Me diste la paga de este mes”.

Los resultados del MOS califican al sistema con 3.9/5, lo cual es aceptable cuando se hace uso de un algoritmo de compresión de voz G.721, el cual posee una valoración de 3.85/5 según los estándares [25]. La Figura 9 muestra la

valoración porcentual obtenida de un banco de 30 personas encuestadas.

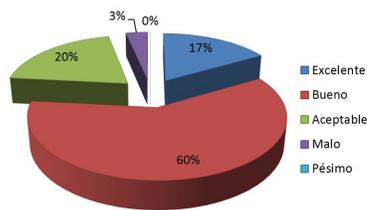


Figure 9. Resultados Mean Opinion Score porcentual

Los resultados de los test de inteligibilidad segmental arrojaron dificultades en la comprensión de composiciones silábicas con letras de baja energía como ‘S’, ‘F’, ‘V’ etc. Sin embargo no se presentaron problemas por parte de los individuos encuestados para comprender estas composiciones silábicas cuando se encontraban en el contexto de una oración, corroborando la calificación de bueno por parte de las encuestas MOS.

V. CONCLUSIONES

El algoritmo de compresión de voz usado reduce significativamente el costo computacional que implica la transmisión de las muestras de voz ya que utiliza una tasa de compresión 2:1, enviando solamente muestras correspondientes a la frecuencia fundamental de la voz para ser reconstruidas posteriormente en Rx.

La eficiencia espectral y el ancho de banda obtenidos contrastan respectivamente con los valores teóricos y experimentales para una modulación GMSK.

Los resultados de *MBSD* indican que el desempeño del sistema disminuirá en tanto la distancia entre dispositivos *USRP* se incrementa, hasta eventualmente en algún punto determinado el mensaje no sea claro o se haya perdido la comunicación.

El retardo obtenido entre Tx y Rx es muy bajo como para que el oído humano lo detecte, pero si da muestras de la carga computacional que se induce en los *USRP*

Dentro de un ambiente de laboratorio el sistema obtuvo una calificación de “bueno” en encuestas MOS. Se presentaron ciertos problemas en cuanto a la inteligibilidad de combinaciones vocálicas de baja energía, sin embargo el contexto general de la comunicación fue perfectamente entendido.

Se recomienda colocar una etapa de filtrado de la señal en el Rx, ya que los armónicos introducidos generan ruido sobre la información transmitida.

Para incrementar el alcance en ambientes de campo abierto se recomienda incluir un amplificador para la señal enviada o reducir la frecuencia de la portadora.

Puede agregarse un bloque de ganancia automática *AGC2*, contenido en la categoría de control de nivel que se encargará de normalizar el volumen de voz sin importar el nivel de la entrada del transmisor.

El sistema de transmisión de voz presentado en este documento no incluye un bloque de codificación de canal principalmente por razones de capacidad de procesamiento del equipo, el cual no soporta algoritmos de redundancia y corrección de errores debido a que el costo computacional asociado a estos procesos es bastante alto.

Como trabajos futuros están: el diseño de un bloque de codificación de canal acorde a las capacidades de procesamiento del *USRP* y la optimización a nivel de líneas de código del sistema diseñado.

REFERENCES

- [1] C. Aldáz, “Emulador de un sistema de comunicaciones utilizando tecnología sdr.” Master’s thesis, Universidad Politécnica Nacional, 2009.
- [2] “Ushr motherboard.” [Online]. Available: gnuradio.org/redmine/projects
- [3] “Ushr daughterboard.” [Online]. Available: gnuradio.org/redmine/projects
- [4] W. Alexander, “Digital communications systems engineering with software defined radio.” [Online]. Available: <http://ecewp.ece.wpi.edu/wordpress/wireless/textbooks/sdrlabs/>
- [5] E. Blossom, “Gnu radio: Tools for exploring the radio frequency spectrum.” *Linux J.*, vol. 2004, no. 122, pp. 4–, Jun. 2004. [Online]. Available: <http://dl.acm.org/citation.cfm?id=993247.993251>
- [6] G. Radio, “Gnu radio companion.” [Online]. Available: gnuradio.org/redmine/projects/gnuradio/wiki/GNURadioCompanion
- [7] M. Andrés and G. Isabel, “Introducción a la programación con python,” *Universitat Jaume I*, 2003.
- [8] E. Research, “Selecting an usrp daughterboard.” [Online]. Available: www.ettus.com/content/files
- [9] A. EHU, “Fundamentos de telefonía.” [Online]. Available: <http://elastixtech.com/fundamentos-de-telefonía/transmision-de-la-voz/>
- [10] “La voz humana.” [Online]. Available: <http://www.ehu.es/acustica/espanol/musica/vohues>
- [11] J. Alvarado Reyes and C. Stern Forgach, “Un complemento al teorema de nyquist,” *Revista mexicana de física E*, vol. 56, no. 2, pp. 165–171, 2010.
- [12] H. Torres-Silva, “Propagación de ondas pulsadas en un chiroplasma magnetizado,” *Revista Mexicana de Física*, vol. 44, pp. 53–58, 1998.
- [13] A. Andrés, B. Pablo, and Q. Edwin, “Modulación gmsk para transmisión de información a través de líneas eléctricas.” *Scientia Et Technica*, 2010. [Online]. Available: <http://www.redalyc.org/articulo.oa?id=84920977017>
- [14] F. Kostedt and J. Kemerling, “Practical gmsk data transmission,” *Wireless Design and Development*, vol. 3, no. 1, pp. 21–25, 1995.
- [15] K. F and K. J, “Practical gmsk data transmission.” [Online]. Available: <http://pdfdata.datasheetsite.com/web/67149/MX429.pdf>
- [16] J. M. T. Nova and H. P. Penagos, “Estudio y comparación en eficiencia espectral y probabilidad de error de los esquemas de modulación gmsk y dbpsk,” *Ingeniería e Investigación*, vol. 28, no. 3, pp. 75–80, 2008.
- [17] I.-T. vocoders, “G.721 itu-t compatible vocoder datasheet.” [Online]. Available: <http://www.design-reuse.com/>
- [18] F. Jeff, “Features-featuring technologies: signal processing-practical costas loop design-designing a simple and inexpensive bpsk costas loop carrier recovery circuit.” *RF design*, vol. 25, no. 1, pp. 20–37, 2002.
- [19] S. Shah and V. Sinha, “Gmsk demodulator using costas loop for software-defined radio,” in *Advanced Computer Control, 2009. ICACC’09. International Conference on*. IEEE, 2009, pp. 757–761.
- [20] G. R. Manual and C. A. Reference, “Costas loop configuration.” [Online]. Available: gnuradio.org/doc/doxygen/classgr_1_1digital_1_1costas_loop_cc.html
- [21] G. R. U. Documentation, “Synchronization.” [Online]. Available: www.reynwar.net/gnuradio/sphinx/gr/sync_blk.html
- [22] M. Narayan, “Wireless communication technologies,” *Rutgers University*, 2002.
- [23] W. Yang and R. Yantorno, “Comparison of two objective speech quality measures: Mbsd and itu-t recommendation p. 861,” in *Multimedia Signal Processing, 1998 IEEE Second Workshop on*. IEEE, 1998, pp. 426–431.
- [24] F. Z. Marcos, “Tratamiento digital de voz e imagen y aplicación a la multimedia,” *Tratamiento de Imagen*, 2000.
- [25] “Tabla resumen de codecs.” [Online]. Available: www.voipforo.com/