

## SISTEMA DIDÁCTICO REMOTO PARA EL APRENDIZAJE DE COMUNICACIONES DIGITALES

### REMOTE TRAINING SYSTEM FOR DIGITAL COMMUNICATIONS LEARNING

Johannes Allen Sánchez, Fabio G. Guerrero  
Universidad del Valle, Cali (Colombia)

#### Resumen

En este artículo se presenta el diseño, desarrollo e implementación de un laboratorio remoto para el aprendizaje de comunicaciones digitales usando una plataforma de *hardware* reconfigurable y un *software* de instrumentación y procesamiento digital de señales. El laboratorio incluye inicialmente cuatro prácticas (espectro de un tren de pulsos, ruido blanco, multicanalización por división de tiempo y muestreo de una señal) pero la flexibilidad del diseño permite agregar fácilmente módulos adicionales. El laboratorio remoto fue integrado en el campus virtual de la Universidad del Valle. Se realizaron pruebas de desempeño para evaluar los resultados en términos de interactividad y confiabilidad de los datos.

**Palabras claves:** laboratorio remoto, comunicaciones digitales, *hardware* reprogramable.

#### Abstract

This paper presents the design, development and implementation of a remote laboratory for learning digital communication systems using a reconfigurable hardware platform and an instrumentation and digital signal processing software. The laboratory includes initially four practices (spectrum of a pulse train, white noise, time division multiplexing and, sampling) but the flexibility of the design allows easily adding additional modules. The remote laboratory was integrated into the virtual campus of Universidad del Valle. Performance tests were conducted to evaluate the results in terms of interactivity and data reliability.

**Keywords:** remote laboratory, digital communications, reprogrammable hardware.

#### Introducción

Los laboratorios de experimentación remotos en combinación con las nuevas tecnologías de la

información y las comunicaciones (TIC) juegan hoy en día un papel importante en la educación en ingeniería y más aún en el área de la electrónica (Gravier, 2008). Este tipo de tecnologías permite

complementar el aprendizaje teórico con la posibilidad de llevar el conocimiento a cualquier sitio con acceso a internet.

Hoy en día existen equipos de laboratorio tanto físicos como de simulación diseñados para introducir y demostrar principios de sistemas de comunicación digital (Bencomo, 2004). Por una parte, los laboratorios de naturaleza física, además de presentar limitantes debido a la rápida obsolescencia y a los espacios y horarios para el desarrollo de las prácticas, requieren en términos de *hardware* una amplia utilización de electrónica discreta para el ajuste y funcionamiento de las prácticas. Por otra, los laboratorios de naturaleza simulada están basados en modelos computacionales preestablecidos que recrean el comportamiento de un fenómeno físico y, aunque poseen grandes características en cuanto a disponibilidad, no generan una experiencia satisfactoria en los usuarios debido a la sensación de estar trabajando con una plataforma física inexistente.

Abu-aisheh (2010) presenta un sistema remoto de aprendizaje NI-ELVIS y la tarjeta Emona DATEX de uso comercial, con el cual los estudiantes pueden llevar a cabo experimentos en el área de comunicación análoga y digital a través de internet. Este conjunto de herramientas es bastante práctico y sofisticado. Sin embargo, su costo no permite una fácil adquisición cuando existen limitaciones de presupuesto.

En otro trabajo (Reyes, 2008), se detalla el desarrollo de un sistema para el aprendizaje de comunicaciones basado en una FPGA, que utiliza interfaces adicionales como puertos PS2 y VGA para su funcionamiento. A pesar de

usar muy bien los recursos *hardware* disponibles, este sistema únicamente cuenta con accesibilidad presencial.

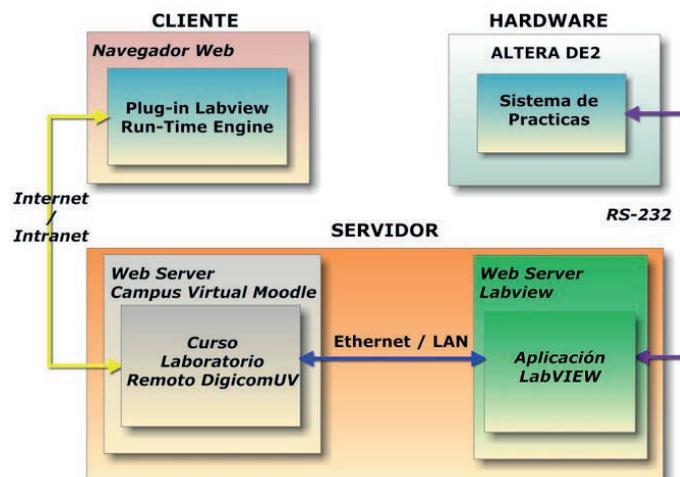
Las dificultades o falencias de los trabajos mencionados anteriormente, tales como costo y accesibilidad al laboratorio, motivaron la realización del presente trabajo, cuyo objetivo principal es diseñar e implementar un sistema didáctico para el aprendizaje de comunicaciones digitales que les permita a los interesados acceder de forma remota, vía web, a prácticas implementadas en un sistema *hardware*. Para una discusión en detalle del material presentado en este artículo se puede consultar Sánchez (2013).

El artículo está organizado como se explica a continuación. En la sección introductoria se presentan el estado del arte y la motivación para el desarrollo del proyecto. En la metodología se expone el procedimiento de diseño e implementación del *hardware* y el *software*, explicando la elaboración de los módulos *hardware* de prácticas y el método de acceso web. Luego se dan a conocer los resultados a las pruebas realizadas al sistema. En la sección de discusión se explican y analizan los resultados obtenidos. Finalmente, en las conclusiones se sintetizan las principales observaciones del trabajo.

## Metodología

La estructura del sistema está basada en el modelo convencional de comunicación cliente-servidor, donde el lado cliente es un usuario autorizado que desee acceder a través de internet al sistema de prácticas conectado en el lado servidor (figura 1).

Figura 1. Estructura del sistema DigicomUV



El cliente, mediante un navegador y vía internet, accede a una interfaz gráfica de usuario (GUI) provista por el servidor y realiza solicitudes a través de dicha GUI para recibir respuesta del sistema de prácticas. Entonces, el servidor procesa las solicitudes y las reenvía directo al sistema de prácticas. Una vez éste recibe las tramas de solicitud, modifica los parámetros solicitados por el usuario y envía la respuesta al servidor que, finalmente, recibe los datos de respuesta del sistema de prácticas, los procesa y con estos actualiza la GUI. La interfaz gráfica se envía al usuario a través de internet para que la visualice.

### Desarrollo del hardware

Para el hardware del sistema de prácticas se utiliza la tarjeta de educación y desarrollo DE2 de Altera

con FPGA Cyclone II EP2C35F672C6. Éste es un kit de bajo costo, altamente reconfigurable, que también aventaja los sistemas de desarrollo similares en cuanto a desempeño para procesamiento digital de señales (Altera, 2005).

El diseño del hardware se realiza bajo una estructura modular, es decir, cada módulo de práctica cuenta con los componentes necesarios para funcionar independientemente de los demás. Adicionalmente, en el diseño se tiene muy en cuenta la parametrización que las prácticas deben tener, ya que es sumamente importante que el usuario pueda modificar variables de cada una de éstas. La implementación del hardware se lleva a cabo mediante el software Quartus II V9.1 utilizando bloques esquemáticos BDF y lenguaje VHDL. La figura 2 muestra el esquema del hardware implementado.

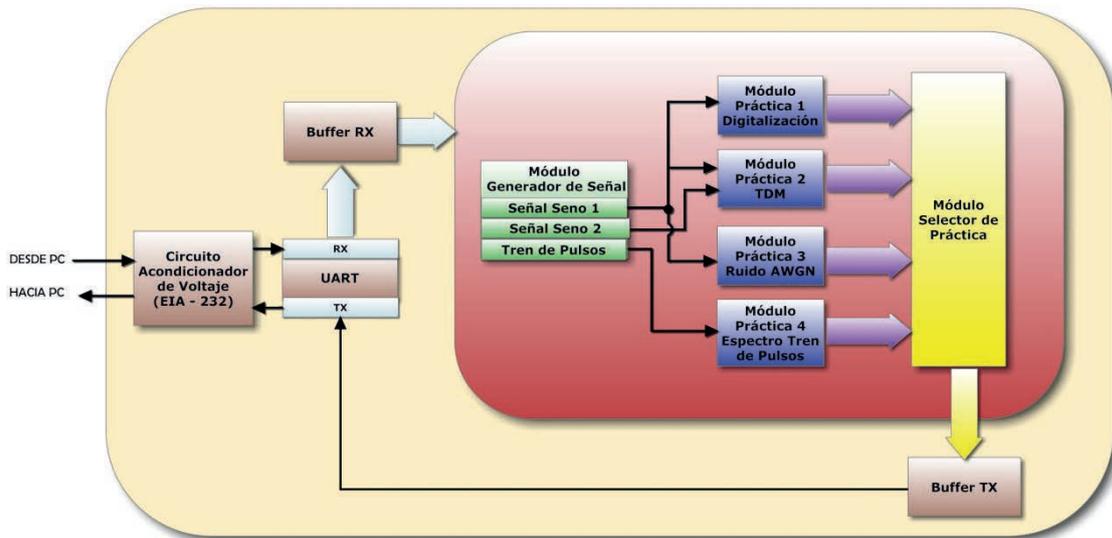


Figura 2. Arquitectura hardware del sistema de prácticas

### Módulo generador de señales

Este módulo se encarga de emular el funcionamiento de un generador de señales, con lo cual logra eliminar la necesidad de un dispositivo externo a la plataforma de desarrollo DE2. La salida de este módulo sirve como señal de entrada a los módulos de prácticas y se diseña para generar tres tipos de señales independientes, dos ondas seno y un tren de pulsos.

Para generar las señales sinusoidales, mediante un software matemático se adquieren los valores de una función seno de amplitud  $\pm 127$  para tres frecuencias distintas descritas por la ecuación

1, y cuya duración en el tiempo discreto es de 512 muestras, cada una de bits de tamaño. Estos valores se almacenan consecutivamente en ROM y se reproducen cada ciclo de reloj al ir aumentando en uno el valor de direccionamiento de lectura hacia esta ROM.

$$F_{\text{Seno}}(P) = \frac{F_{\text{reloj}}}{512 * 2^P} = \frac{50\text{Mhz}}{512 * 2^P} \quad (1)$$

Las frecuencias de las señales sinusoidales obtenidas para  $P = 0, 1, 2$  respectivamente son, 97.656kHz, 48.828 kHz y 24.414 kHz.

La señal tren de pulsos se realiza mediante un circuito encargado de conmutar su salida entre  $-127$  y  $+127$  cada múltiplo  $2^n$  del periodo de la señal de reloj del sistema  $T_{CLK}$ , donde  $n(1:8)$  es una señal decodificada del parámetro de entrada de 3 bits ajustado por el usuario; de esta forma se obtiene una frecuencia variable la cual es descrita por la ecuación 2.

$$F_{Pulso} = \frac{F_{CLK}}{2^n} = \frac{50MHz}{2^n} \quad (2)$$

### Practica 1. Digitalización

Ya que la señal de entrada es la emulación digital de una señal analógica, para lograr el proceso de muestreo sobre ésta se diseña un circuito de diezmo o *downsampler*, para tomar  $2^{n+1}$  muestras equidistantes de la señal original, donde  $n$  es la señal de entrada que configura el muestreo, que va de 0 a 7. Así, la

nueva frecuencia de muestreo queda determinada por la ecuación 3.

$$f_{Samp2} = \frac{f_{Samp1}}{D} = \frac{50MHz}{2^{n+1}} \quad (3)$$

Un circuito cuantificador toma los datos provenientes del circuito *downsampler* y realiza una comparación basada en unos rangos establecidos. Se introduce una entrada de 3 bits  $N[2..0]$  al circuito para modificar la cantidad de niveles de cuantificación  $L$  requeridos. Siendo el valor decimal de la entrada  $N[2..0]$ , se define mediante la expresión 4.

$$L = 2^c + 1 \quad (4)$$

Las datos de salida del módulo de práctica 1 (figura 3) así como los del resto de prácticas se conectan a los módulos de Selección y Comunicación para su posterior procesamiento.

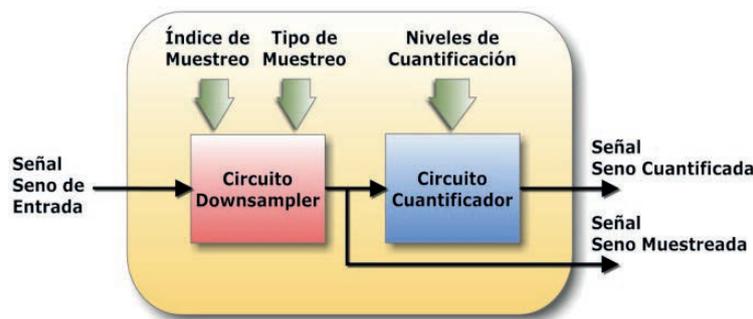


Figura 3. Módulo de Práctica 1. Digitalización (muestreo-cuantificación)

### Practica 2. TDM analógico

Se diseña un circuito para la multiplexación por división en el tiempo TDM (figura 4) de dos señales analógicas que provienen del módulo generador de señales. Éstas se conectan a la entrada de un multiplexor 2 a 1, el cual tiene como señal de control la salida de un arreglo de *Flip-Flops* tipo  $D$  que conmutan entre 0 y 1 a la frecuencia del reloj del sistema. Para este circuito TDM, el periodo de muestreo de 50MHz se divide en las dos señales multiplexadas, de esta forma cada señal se muestrea a una frecuencia igual a 25MHz:

$$f_{Samp1=samp2} = \frac{f_{Samp}}{2} = \frac{50MHz}{2} \quad (5)$$

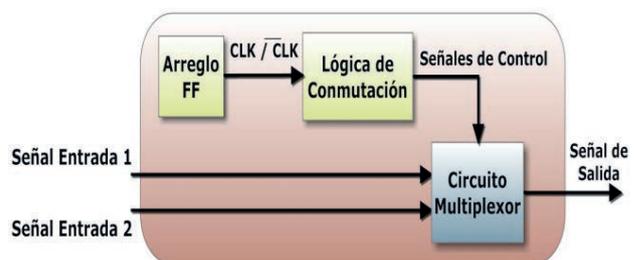


Figura 4. Circuito TDM de módulo de práctica 2

### Practica 3. Ruido blanco

Esta práctica, cuya arquitectura se muestra en la figura 5, se basa en el trabajo propuesto en Boutillon (2008), que presenta un generador de ruido blanco gaussiano para la emulación de un canal de comunicaciones.

Con el método de Box-Muller se genera una muestra aleatoria  $n$  de distribución gaussiana  $N(0,1)$  (Media  $\bar{x}=0$ , Desviación estándar  $\sigma=1$ ) a partir de dos variables aleatorias  $x_1$  y  $x_2$  distribuidas uniformemente en el rango  $[0,1]$ . Los valores de las funciones en ROM  $f(s)$  y  $g(s')$  de 9 y 8 bits, respectivamente, son leídos mediante acceso aleatorio a memorias por los LFSR y dirigidos a un multiplicador que entrega  $n$  muestras gaussianas de 10 bits, divididos así:  $A = 4bits$  para el valor entero y  $B = 6bits$  para la parte fraccional. Para reducir la variación de la distribución de las muestras, se acumulan cuatro variables Box-Muller cada periodo de reloj  $T_{CLK}$  mediante el teorema del límite central, para generar una sola muestra gaussiana  $N$  cada periodo  $T_{CLK}/4 = 50MHz/4 = 12.5MHz$ .

Se implementa el bloque SNR que toma únicamente el valor entero  $A$  de las muestras gaussianas  $N$  y según el valor de una entrada de 3 bits SNR  $[2..0]$  desplaza el punto decimal a la derecha o izquierda para multiplicar o dividir el valor máximo base  $(\pm 7)$  de la señal de ruido. De esta forma los valores máximos de amplitud de la señal de ruido blanco son:

$$A_{\text{ruido}} [7..0] = (0, 7, 15, 31, 63, 127)_{10}$$

La última etapa del módulo se encarga de sumar la señal de ruido a la señal analógica de entrada para emular el comportamiento de una señal analógica afectada por un ruido de amplitud variable.

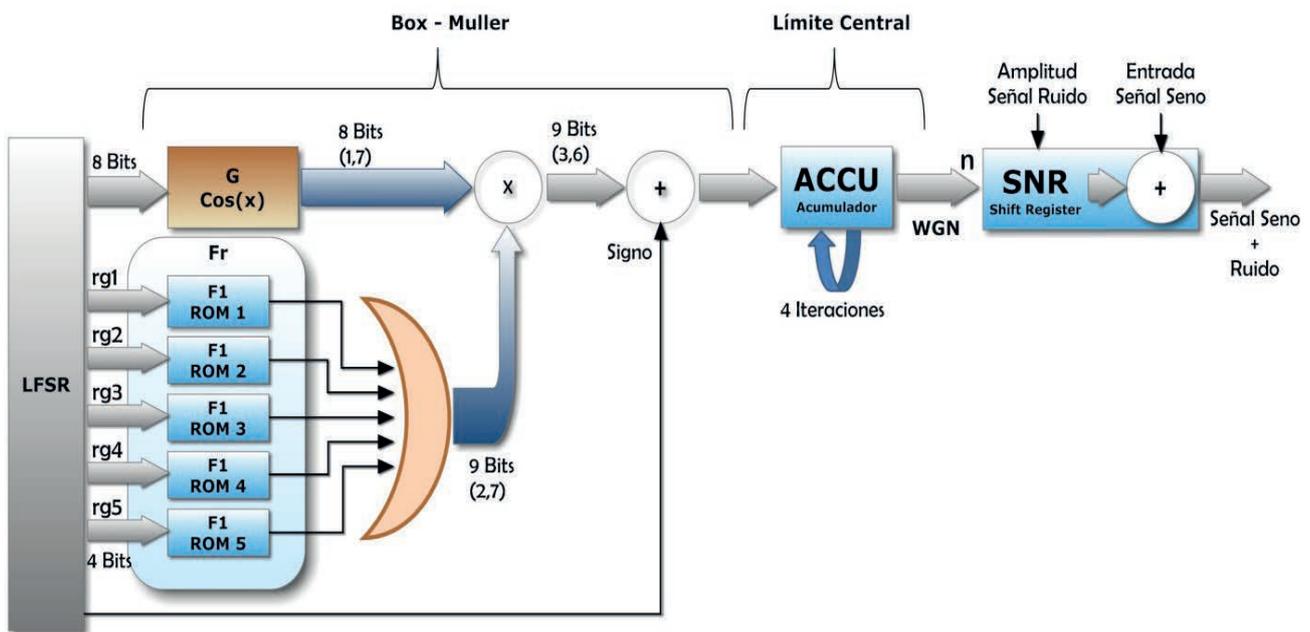


Figura 5. Módulo de práctica 3. Ruido blanco

#### Práctica 4. Espectro de un tren de pulsos

El módulo de la práctica 4 se compone de tres bloques principales: DFT, FSM\_DFT y Cordic. Este módulo se encarga de tomar las muestras de un tren de pulsos con frecuencia variable y entregar su respectivo espectro en frecuencia en forma de magnitud y fase. La figura 6 muestra la arquitectura *hardware* de la práctica.

Un bloque FSM\_DFT se implementa a partir de una máquina de estados para controlar las señales de la interfaz *Avalon Streaming*, que a su vez hace posible la transferencia de datos hacia el motor FFT sin detener el cálculo de la transformada.

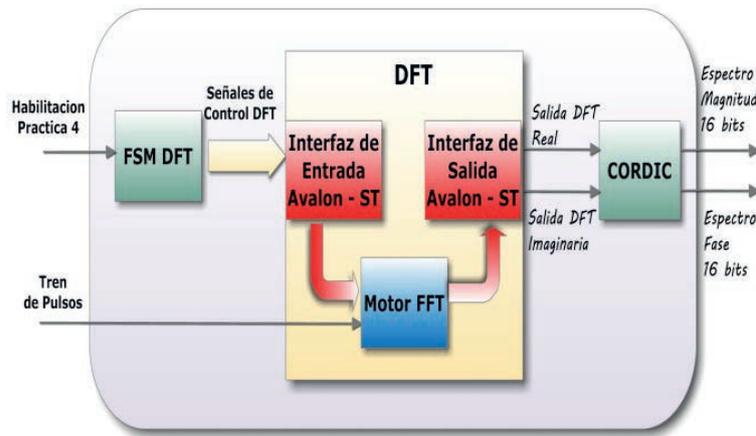


Figura 6. Arquitectura *hardware* del módulo de práctica 4

El bloque DFT se implementa a partir de la mega-función FFT V9.1 de Quartus, cuya arquitectura (Altera, 2011) ejecuta un algoritmo FFT Radix-2<sup>2</sup> con decimación en frecuencia y se configura para aceptar como entrada un vector de 512 muestras complejas de 8 bits (*Sink\_Real*[7..0], *Sink\_Imag*[7..0]) en formato complemento A2. Estos datos se procesan en orden natural FIFO por un motor FFT con representación de datos en punto fijo. El FFT entrega un vector de salida de 512 datos con ancho de palabra de 20 bits (*Source\_Real*[19..0], *Source\_Imag*[19..0]).

Para obtener el espectro en frecuencia de la señal del tren de pulsos en forma de magnitud y fase, la representación de la salida del bloque DFT se transforma de rectangular a polar. Esto se realiza mediante el bloque Cordic, que se desarrolla basado en el trabajo de Herveille (2001).

El bloque Cordic toma dos palabras con signo de 16 bits  $X_i, Y_i$  (*Source\_Real*[19..4], *Source\_Imag*[19..4]), las cuales son las coordenadas rectangulares de un punto en el espacio bidimensional, y ejecuta un algoritmo para calcular funciones trigonométricas a partir de 16 sucesivas rotaciones angulares ( $n=0: 1:15$ ) desde un vector ángulo inicial hasta un vector ángulo deseado, entregando un par de salidas de 16 bits equivalente a las coordenadas polares con amplitud y ángulo  $A[15..0], F[15..0]$ . La compensación a la reducción de amplitud se realiza posteriormente mediante *software*.

Finalmente se convierten los valores del ángulo de radianes a grados teniendo en cuenta el tamaño de palabra de salida de 16 bits:

$$Fase_{Grados^\circ} = (Fase_{Rads} * 360) / 2^{16} \quad (7)$$

### **Módulo de comunicación**

El módulo de comunicación permite la transmisión y recepción de información hacia y desde el sistema de prácticas. Está compuesto a su vez por tres módulos principales: el *UART*, el de transmisión *Buffer TX* y el de recepción *Buffer RX*.

Como mecanismo de comunicación entre el sistema de prácticas y el servidor se utiliza una interfaz serial RS-232 y se definen los siguientes parámetros de comunicación, Velocidad 115200B/s, 8 bits de datos, 1 bit de parada, sin paridad ni control de flujo. Con base en estos parámetros se diseña el módulo *UART*, que permite la conversión de una entrada en paralelo a serial para ser transmitida y de serial a paralelo para ser recibida. La velocidad de comunicación se obtiene mediante un bloque divisor de frecuencia que realiza un número de cuentas necesarios para generar el tiempo de bit de transmisión y recepción ( $T_{bit} = 1 \text{ seg} / 115200 \text{ bits} = 8.68 \mu\text{seg}$ ).

En cuanto al módulo *Buffer TX*, éste se encarga de controlar el envío de la información proveniente del módulo de prácticas hacia el *UART*, que realiza dos funciones principales: escribe los resultados de las prácticas en bancos de memoria RAM de forma paralela a la velocidad del reloj del sistema, 50MHz, y simultáneamente realiza la lectura secuencial de los datos almacenados en estas RAM a la velocidad de transmisión de la interfaz serial 11.52 KB/s.

Por otra parte, el módulo Buffer RX, mediante una máquina de estados FSM\_Recepción analiza y procesa las tramas de configuración enviadas por el usuario al sistema de prácticas. Una vez el módulo Buffer RX verifica que la trama entrante contenga los elementos de cabecera y cola válidos, los valores de configuración se sostienen en registros de lectura como entrada a los módulos de prácticas hasta que arribe otra trama de configuración válida.

## Desarrollo del software

La interfaz gráfica de usuario GUI se lleva a cabo a partir de un programa desarrollado en LabVIEW V11.0 de National Instruments, el cual procesa las solicitudes por parte del usuario y las envía al sistema de prácticas a través de la interfaz serial RS-232 y espera por la respuesta a dicha solicitud para graficarla. Esta GUI se basa en un panel frontal en el que se ejecutan dos buclas principales intercomunicadas, la comunicación y el llamado a subpanel.

Con el bloque de comunicación se establecen los parámetros de la interfaz serial RS-232 del equipo servidor y se realiza el proceso de transmisión y recepción entre éste y el sistema de prácticas. En la transmisión se toman los valores seleccionados por el usuario para cada interfaz de práctica y con estos se ensambla una trama de configuración que se envía al sistema de prácticas a través de la interfaz serial. Las respuestas del sistema de prácticas se almacenan en el *buffer* FIFO de entrada del chip UART 16550 del equipo servidor, se configura el programa para que continuamente se lean 4 bytes desde este *buffer* y se grafiquen en paneles de forma de onda.

En cuanto a la publicación Web de la interfaz, se aprovecha que LabVIEW cuenta con un servidor web que permite acceder a los instrumentos virtuales a través de internet. Se configura este servidor web para acceder a la aplicación LabVIEW desarrollada a través del puerto 80 y se habilita la opción de paneles frontales remotos. Una vez realizado esto mediante la herramienta de publicación web de LabVIEW se embebe la aplicación en un documento HTML el cual se aloja en el servidor web de prácticas.

A través de un recurso web de la plataforma Moodle 2.2 para gestión de cursos del campus virtual de la Universidad del Valle, se embebe el documento HTML

anteriormente mencionado; de esta manera, el servidor del campus virtual redirecciona las peticiones HTML al servidor de prácticas a través del puerto 80. Esto se logra debido a que los dos servidores citados están conectados en la misma red local. De esta forma, el cliente puede visualizar y controlar remotamente la interfaz de prácticas a través de un navegador web únicamente ingresando al sitio web del curso y teniendo instalado el plug-in gratuito LabVIEW RTE en su computador. La figura 7 presenta el esquema de la arquitectura de red que hace posible la interacción del usuario con el sistema de prácticas.

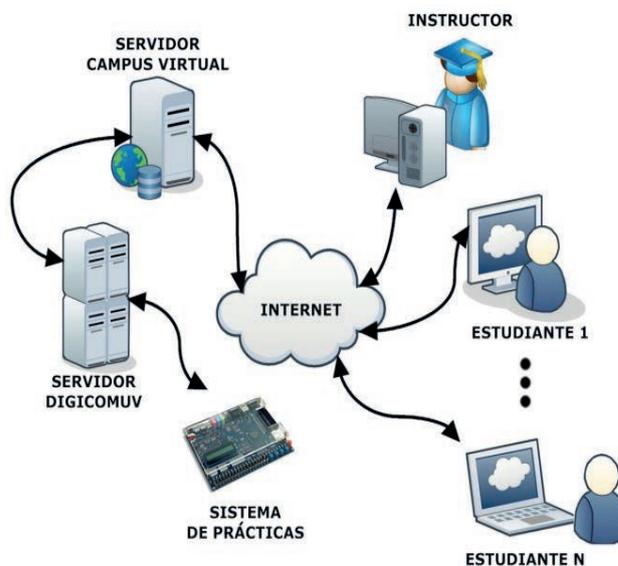


Figura 7. Configuración cliente-servidor-sistema de prácticas

## Resultados

Para probar el desempeño del *hardware* de las prácticas se utilizó DSPBuilder, de modo tal que se llevó a cabo la descripción *hardware* VHDL de los módulos implementados a Matlab, con lo cual se buscó comparar con la misma herramienta los resultados del *hardware* desarrollado con su modelo idealizado equivalente de Matlab.

### Resultados de la práctica 1

Se tomaron los resultados en amplitud entre el modelo implementado y el modelo Matlab para cada una de los bloques que comprenden la práctica 1, y haciendo uso de la ecuación 9 se obtuvo el error promedio relativo para las 512 muestras de la señal de respuesta (tabla 1).

$$\varepsilon = \frac{\text{Amplitud de muestra Matlab} - \text{Amplitud de muestra implementada}}{\text{Amplitud de muestra Matlab}} \quad (9)$$

Tabla 1. Error relativo entre modelo implementado y modelo idealizado Matlab

Módulo Generador			% $\bar{\varepsilon}$ Frec 24kHz	% $\bar{\varepsilon}$ Frec 48kHz	% $\bar{\varepsilon}$ Frec 96kHz
Módulo de práctica 1	Circuito muestreador	$T_{samp} = \frac{T}{2^{m+1}} ; m = 0,1, \dots, 7$	0.00	0.00	0.00
	Circuito cuantificador	Número de Niveles $N = 2^n$ $n = 1,2, \dots, 7$	0.39	0.78	1.56

### Resultados de la práctica 2

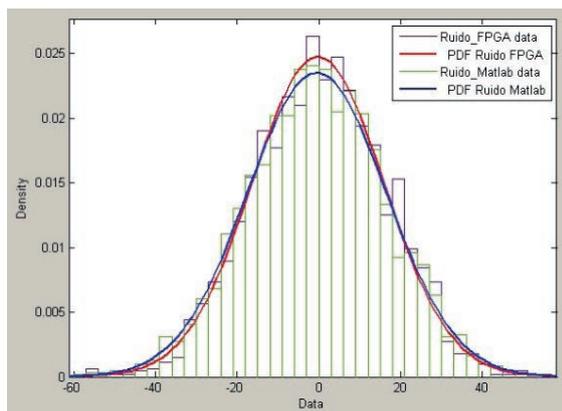
En el módulo de práctica 2 se realizaron mediciones para las 6 posibles configuraciones en frecuencia. Los resultados fueron comparados con Matlab y se consignaron en la tabla 2.

Tabla 2. Resultados del módulo de práctica 2 – Implementado vs. Matlab

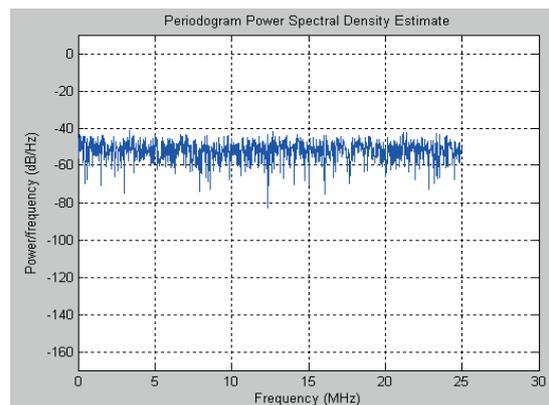
Señal 1	Señal 2	% $\varepsilon_{Amp}$	Frecuencia obtenida		% $\varepsilon_{Frec}$	
			$f_{señal1}$	$f_{señal2}$	$f_{señal1}$	$f_{señal2}$
24 kHz		0.00	24.03 kHz		1.25	
48 kHz			48.83 kHz		1.73	
96 kHz			97.66 kHz		1.73	
24 kHz	48 kHz		23.65 kHz	48.83 kHz	1.45	1.73
24 kHz	96 kHz		24.03 kHz	97.66 kHz	1.25	1.73
48 kHz	96 kHz		48.45 kHz	97.66 kHz	0.93	1.73

### Resultados de la práctica 3

Para la práctica 3 se midió la función de densidad probabilística PDF para 512 muestras (Fig. 8.a), y la densidad espectral de potencia PSD con ventana Hamming para 2048 puntos (Fig. 8.b)



(a)



(b)

Figura 8. (a) PDF Modelo implementado vs. MATLAB, (b) PSD WGN implementado

La tabla 3 resume los resultados obtenidos en la medición de la distribución para diferentes configuraciones del módulo entre el modelo del ruido implementado y el modelo en Matlab.

Tabla 3. Resultados de PDF para el módulo de práctica 3

PDF CÁLCULO DE ERROR DISTRIBUCIÓN NORMAL						
Amplitud de Ruido	Implementado		Matlab		Aproximación a la distribución (%)	
	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu_e$	$\sigma_e$
x1	-0.489	1.046	-0.027	0.898	(-) 46.2	16.48
x2	-0.478	2.037	-0.062	1.995	(-) 41.6	2.10
x4	-0.444	4.045	-0.124	3.991	(-) 32.0	1.35
x8	-0.376	8.072	-0.279	8.980	(-) 9.7	10.11
x16	-0.252	16.109	-0.528	16.962	(+) 27.6	5.02

**Resultados de la práctica 4**

En la figura 9 se muestra el resultado al comparar la respuesta del *hardware* implementado para la práctica 4 y su equivalente Matlab con un tren de

pulsos, Amplitud 0 - 127, frecuencia 781,25 kHz – 64 muestras de 20nseg, ciclo útil 0,5, tamaño de transformada 512 puntos.

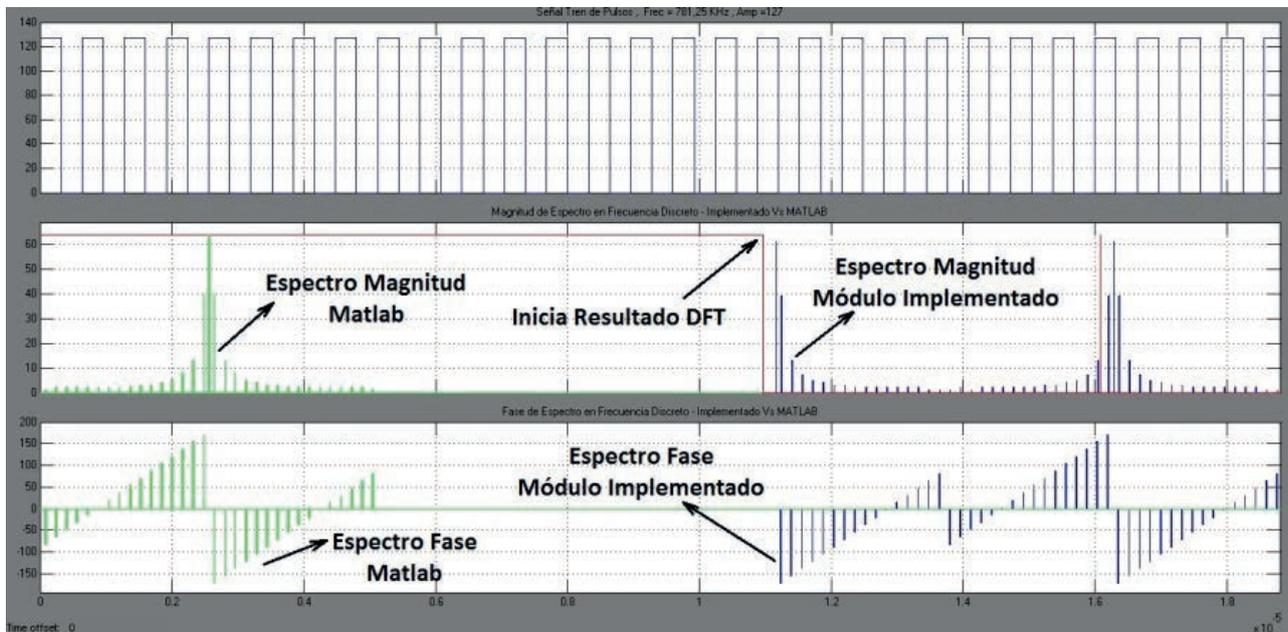


Figura 9. Espectro discreto de amplitud y fase de un tren de pulsos. Modelo implementado vs. Matlab

**Latencia del sistema**

Considerando que la interactividad entre el usuario y el sistema de prácticas es uno de los aspectos más importantes, se realizaron pruebas para la estimación

del tiempo de respuesta del laboratorio remoto con el fin de identificar la mayor latencia del sistema en general.

Para estimar el tiempo promedio de carga de la interfaz gráfica en el navegador del cliente se realizaron 10

capturas de tráfico con el programa de uso libre Wireshark en su versión 1.8.3 con un computador portátil, Windows 7, 2GB de memoria RAM, navegador Google Chrome, conectado a una red wi-fi de la Universidad del Valle con dirección IP dinámica. Gracias a estas capturas se logró calcular un tiempo promedio  $\bar{t}_{CS} = 4.7042 \text{ seg}$ , equivalente al tiempo de respuesta por parte del servidor ante una solicitud realizada por el cliente. De igual forma, con estos resultados se logró estimar el tiempo promedio de la actualización de los datos en la GUI ante solicitudes del servidor al sistema de prácticas igual a  $\bar{t}_{GUI} = 9.964 \text{ seg}$ .

## Discusión

### *Análisis de desempeño del hardware implementado*

Se obtuvo una muy buena respuesta del módulo generador de señales, ya que el error promedio  $\bar{\epsilon}$  obtenido para éste no supera el 1.55%. Este mínimo error se debe al redondeo de los valores seno ingresados en ROM a ser reproducidos, el cual podría ser disminuido aún más al aproximar los valores almacenados en ROM a los ideales de la función seno.

Al analizar las respuestas temporales de las prácticas 1 y 2, se pudo establecer que los resultados de los valores en amplitud son muy próximos a sus respectivos modelos equivalentes idealizados de Matlab. Esto se debe a la exactitud en los instantes de las muestras ejecutados por los circuitos que conforman los módulos. En cuanto al bloque de cuantificación, se observa en la tabla 1 un error promedio  $\bar{\epsilon}$  de 0,39% que, aún muy bajo, se debe al intervalo de decisión en los cruces por cero de la señal por cuantificar.

Los resultados espectrales obtenidos para la práctica 2 (tabla 2) permitieron ver las frecuencias contenidas en la señal TDM de respuesta del módulo. Gracias a estos resultados se puede establecer que para cada una de las configuraciones del módulo el error obtenido entre la frecuencia esperada y obtenida no superó el 1,73%. Este error podría ser más bajo si el análisis

espectral de la señal TDM fuera calculado con más de los 32768 tomados.

Con la implementación de la práctica 3 se pudo ver el efecto del canal de ruido blanco gaussiano que afecta la amplitud de la señal sinusoidal. Con los resultados de la PDF (figura 8.a) se logró establecer que aunque la media  $\mu$  alcanzada para las distintas amplitudes del ruido es bastante próxima a cero, con este módulo se sacrificó un poco la medida ideal de la distribución normal para obtener una relación señal a ruido SNR variable que el usuario pueda ver con más claridad. La distribución gaussiana puede mejorarse notablemente si se considera la parte fraccionaria en la amplitud de ruido. Al analizar la PSD del ruido obtenido (figura 8.b), se pudo establecer que con un cálculo del espectrograma superior a 2048 puntos se comienza a evidenciar el espectro plano del ruido WGN.

En relación con la práctica 4, en los resultados que se muestran en la figura 9 se logra apreciar que el motor FFT del bloque DFT calcula y entrega el primer resultado después de 1024 muestras (línea roja) y se obtiene un retardo  $T_{FFT\_Delay} = 20nseg * 1024 = 20,48 \mu seg$  después de la respuesta del modelo equivalente de Matlab. Adicional a esto, el bloque Cordic tarda 16 muestras más para la transformación de coordenadas rectangulares a polares, con un retardo total en la primera muestra del espectro de magnitud y fase de aproximadamente  $20,80 \mu seg$ . Aunque este retardo es apreciable en términos computacionales, no lo es para el usuario final, ya que la lectura en la interfaz para el primer dato se logra en el tiempo

$$\frac{1}{(Vel_{TX-RX})_{Byte}} = 1/11520 B/seg = 86,8 \mu seg.$$

Igualmente, a partir de los resultados en amplitud y fase observados para el módulo *hardware* de la práctica 4 se logra determinar que los valores de frecuencia obtenidos coinciden fielmente con los resultados del modelo Matlab. Se verifica también el teorema de Nyquist al observar la máxima frecuencia permitida ( $25MHz$ ) en la muestra  $N = 256$ .

### **Análisis de recursos hardware utilizados en la FPGA**

La cantidad de recursos utilizados sobre la FPGA no supera el 20% en elementos lógicos combinacionales y registrados, mientras que el consumo de elementos DSP es del 47%, del cual el 45,71% se debe a los multiplicadores para cada etapa *butterfly* en el cálculo de la transformada FFT. Esto permite pensar que la FPGA, luego de la implementación, aún cuenta con suficiente espacio para agregar más *hardware* o adicionar complementos *hardware* a los módulos de las prácticas existentes.

### **Análisis de resultados del software implementado y de la latencia del sistema**

El tiempo promedio que tarda en cargar la GUI del sistema en el navegador web es aceptable ya que el usuario únicamente nota este retardo cuando solicita por primera vez la GUI; posterior a esto, la página web  $\overline{t_{CS}} = 4.7042 \text{ seg}$  es continuamente actualizada dependiendo del tipo de conexión a internet que tenga el usuario. Para las pruebas realizadas esta actualización ocurre entre 0,4 y 6mseg.

De igual forma se registró un tiempo promedio  $\overline{t_{GUI}} = 9,964$  que equivale al tiempo que tarda en actualizarse la GUI con los resultados del sistema *hardware* ante la modificación de parámetros de las prácticas desde la GUI en el servidor. Esta latencia se debe principalmente al tamaño del *buffer* de lectura, el cual se escogió superior a la tasa de arribo de bits por el enlace serial

$$\text{Buffersize} = 5 * \frac{11520B}{\text{seg}} = 57600\text{Bytes}$$

con el fin de evitar errores por desbordamiento del *buffer* de lectura. Pero debido a este tamaño del *buffer* y a que sólo se leen simultáneamente 4 bytes que corresponden a las señales por graficar, la lectura de los nuevos datos tarda en actualizarse en los paneles de las prácticas. Esta latencia puede disminuir aumentando la velocidad de lectura del

*buffer* y colocando los valores leídos en colas de datos.

### **Análisis del entorno web y la interfaz gráfica de usuario.**

Como se muestra en la figura 10, a través de una ventana emergente se embebe la GUI que simultáneamente se ejecuta en el servidor de prácticas. De esta forma los servicios ofrecidos al usuario son separados. Mientras el servidor del campus provee la información del curso y el acceso hacia la interfaz gráfica, el servidor de prácticas provee la experimentación con el sistema de prácticas a través de la GUI. Este entorno web, además de contener el recurso de acceso al sistema de prácticas, cuenta con materiales como la guía de prácticas para el usuario y el manual de funcionamiento del sistema.

El uso del método de subpaneles para el desarrollo del programa de la GUI fue apropiado, ya que fácilmente se pueden agregar en un futuro los instrumentos virtuales correspondientes a otras prácticas sin necesidad de modificar el programa de la interfaz principal. En cuanto a los paneles gráficos de la GUI para las prácticas, se diseñaron primordialmente para que las señales se desplazaran en el tiempo y no permanecieran estáticas, así los usuarios reconocerán el laboratorio como un sistema remoto y no como un sistema virtual simulado.

A través de las múltiples pruebas de navegación realizadas en la interfaz gráfica, variando los parámetros de las prácticas y observando las señales de respuesta presentadas en los gráficos, se determina que los resultados visualizados en la GUI corresponden fielmente a los resultados del sistema de prácticas en la tarjeta de desarrollo. fielmente a los resultados del sistema de prácticas en la tarjeta de desarrollo.

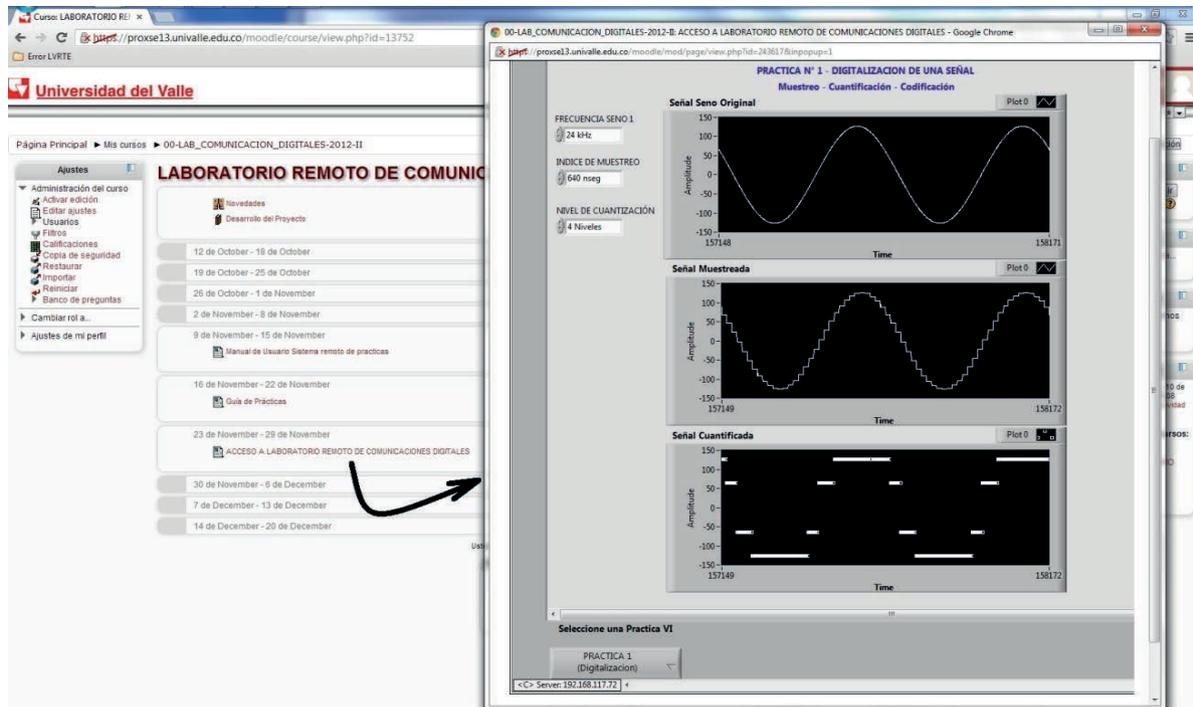


Figura 10. Visualización del entorno web del curso y de la interfaz gráfica de usuario

## Conclusiones

En este trabajo se ha presentado el diseño e implementación de un laboratorio didáctico remoto para el aprendizaje de conceptos de comunicación digital, partiendo de cuatro prácticas básicas. Sin embargo, el diseño del sistema es suficientemente flexible para agregar módulos adicionales y aumentar la funcionalidad del laboratorio.

Aunque el sistema es monousuario, algunas ventajas del que ya ha sido implementado como poca área ocupada en FPGA y el mecanismo de comunicación servidor-FPGA permiten pensar en agregar al sistema la característica de multiusuarios.

La velocidad de respuesta del sistema se sacrificó al aumentar las funcionalidades propias de los recursos gráficos de la GUI, con el fin de aproximar a los usuarios a un entorno remoto agradable y dinámico. A pesar de latencia registrada, la interactividad entre usuario y sistema es aceptable.

A diferencia de otros laboratorios remotos existentes que necesitan de equipos de medición y generación de señales para funcionar, éste integra todos los módulos *hardware* necesarios, lo que quiere decir que se ha desarrollado un sistema autodependiente.

El sistema de prácticas se basa en una tarjeta de muy bajo costo, lo que genera una solución efectiva para las instituciones educativas con recursos precarios que desean tener acceso a laboratorios de tipo presencial.

## Agradecimientos

Deseamos dar nuestro agradecimiento al grupo de Bionoelectrónica de la Universidad del Valle por su colaboración con el sistema ALTERA DE2; y a la Dirección de Nuevas Tecnologías y Educación Virtual de la Universidad del Valle (DINTEV) por permitirnos realizar las pruebas del sistema con sus equipos y en sus instalaciones.

## Referencias

---

- Abu-aisheh, A. A., Eppes, T., & Al-Zoubi, A. Y. (2010). Implementation of a Remote Analog and Digital Communications Lab for a Learning, *IEEE EDUCON Education Engineering, Session T1A*, 1425–1429.
- Altera. (2005). *FPGAs for High-Performance DSP Applications*. Ver 1.1.
- Altera. (2011). *FFT MegaCore Function User Guide*, 3-6.
- Bencomo, S. (2004). Control learning: present and future, *Annual Reviews in Control*, 28,115-36.
- Boutillon E., Danger Jean-Luc, Ghazel Adel, Laamari Hédi. (2008). Efficient FPGA implementation of gaussian noise generator for communication channel emulation. *7<sup>th</sup> IEEE International Conference on Electronicsm Circuits & Systemes*, Kaslik: Liban.
- Gravier, C., Fayolle, J., Bayard, B., Ates, M., & Lardon, J. (2008). State of the art about remote laboratories paradigms - foundations of ongoing mutations, *International Journal of Online Engineering*, 4(1), 1–9.
- Herveille Richard. (2001). *Cordic Core Specification*, Opencores.
- Reyes, R. S. J., Oppus, C. M., Monje, J. C. N., Patron, N. S., Guerrero, R. C., & Fajardo J. T. B. (2008). FPGA Implementation of a Telecommunications Trainer System, *International Journal of Circuits, Systems and Signal Processing*, 2(2),174-180.
- Sánchez J. A. (2013). *Diseño e implementación de un sistema didáctico remoto para el aprendizaje de comunicaciones digitales*. Trabajo de grado en Ingeniería Electrónica, Universidad del Valle, Santiago de Cali.

## Autores

---

### Johannes Allen Sánchez

Ingeniero Electrónico, Universidad del Valle, Cali, Colombia,  
johannes.sanchez@correounivalle.edu.co

### Fabio G. Guerrero

Ingeniero Electrónico y Telecomunicaciones M.Sc, Docente, Universidad del Valle, Cali, Colombia,  
fabio.guerrero@correounivalle.edu.co

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.