

# A hardware implementation of Compressive Sensing Theory

**Alexis D. Velázquez, José Luis Paredes, Francisco Viloria**

*Escuela de Ingeniería Eléctrica, Universidad de Los Andes. Mérida 5101, Venezuela.  
Tel-Fax: (0274) 240 2903*

## Abstract

In this paper, the new theory of compressive sensing (CS) that unifies signal sensing and compression into a single task is implemented on a Digital Signal Processing (DSP) board. An iterative algorithm for signal reconstruction known as Matching Pursuit is implemented on the DSP and used to the reconstruction of real signals from a reduced set of random projections. Two kinds of validation procedures are used to test the reconstruction algorithm implemented. More precisely, sparse signals synthesized on the DSP and sparse signals generated by a special-purpose generator are used to experimentally test the compressive sensing theory verifying in this way its potential. It is shown that the CS theory is able to recover the most significant values of the underlying signal, while yielding negligible differences between the original signals and the reconstructed ones.

**Keywords:** Compressive Sensing, sparse signals, signal reconstruction, Matching Pursuit.

## Implementación a nivel de hardware de la teoría de *Compressive Sensing*

### Resumen

En este artículo, la nueva teoría de *Compressive Sensing* es implementada en una tarjeta de adquisición. Esta teoría permite realizar en forma simultánea los procesos de muestreo y compresión de señales poco densas. El algoritmo iterativo de reconstrucción *Matching Pursuit* es implementado y usado para la reconstrucción de señales reales a partir de un conjunto reducido de proyecciones aleatorias. El algoritmo implementado es sometido a dos diferentes tipos de validación: 1) la reconstrucción de señales sintetizadas en la tarjeta y 2) la reconstrucción de señales pocas densas producidas por un generador de propósitos especiales. Se corrobora experimentalmente el potencial de esta teoría al comparar las señales reconstruidas con respecto a las señales originales observándose prácticamente no diferencia apreciable.

**Palabras clave:** *Compressive Sensing*, señales poco densas, reconstrucción de señales, *Matching Pursuit*.

### 1. Introducción

*Compressive Sensing* (CS) es una teoría recientemente introducida por Candes [1] y Donoho [2], la cual intenta unificar el proceso de muestreo y compresión de señales pocas densas en forma simultánea. En principio, la teoría de CS establece que una señal poco densa puede re-

cuperarse, con alta probabilidad, a partir de un conjunto reducido de proyecciones lineales aleatorias usando algoritmos de reconstrucción no lineal. La señal puede ser poco densa en cualquier dominio (tiempo, frecuencia, wavelet, etc.) y el número de proyecciones aleatorias usadas para recuperar la señal, en general, es mucho menor que el número de muestras, permitiendo así re-

ducir la frecuencia de muestreo y como consecuencia disminuir los recursos de conversión analógico-digital, los recursos de almacenamiento y/o los recursos de transmisión.

Innumerables trabajos han sido reportados recientemente donde se muestra el potencial de uso de esta teoría (Ver el compendio de trabajos disponibles en [www.dsp.ece.rice.edu/cs/](http://www.dsp.ece.rice.edu/cs/)), incluyendo la implementación a nivel de dispositivo de adquisición de señales bidimensionales [3]. Ante tales prometedores resultados surge la necesidad de realizar una implementación a nivel de hardware (tarjeta de desarrollo) de la teoría de *Compressive Sensing* (CS) con lo cual se permita corroborar experimentalmente y a nivel de dispositivos de hardware tal teoría, así como también estudiar las limitaciones y los aspectos inherentes a un proceso de implementación. En este trabajo se implementa la teoría de *Compressive Sensing* en la tarjeta de desarrollo TMS320C30. Inicialmente, se hacen pruebas del algoritmo de reconstrucción no lineal implementado en la tarjeta así como del proceso de proyección usando señales sintetizadas generadas en la misma tarjeta. Seguidamente, usando un generador programable se generan señales poco densas en el dominio de la frecuencia (señales cuadradas, triangulares, o combinaciones lineales de funciones senos/cosenos) las cuales se adquieren a través de la tarjeta de desarrollo, se proyectan y se ejecuta el algoritmo de reconstrucción en la tarjeta de desarrollo. A fin de validar el proceso de reconstrucción, se compara la señal analógica producida por el generador programable con la obtenida por el algoritmo de reconstrucción, ambas mostradas en un osciloscopio digital.

## 2. Revisión de la teoría de *Compressive Sensing*

Sea  $X \in \mathfrak{R}^N$  el vector cuyas componentes representan la versión discretizada de una señal analógica de interés y  $\Psi = [\psi_1, \psi_2, \dots, \psi_Z]$  la matriz cuyas columnas definidas en  $\mathfrak{R}^N$  conforman un diccionario de funciones básicas, ortogonales o no, que son usadas para representar señales como combinaciones lineales de ellas. A los vectores  $\psi_i$ , para  $i=1, 2, \dots, Z$ , se les denominan átomos del diccionario y pueden ser funciones bases senos y/o cosenos, funciones ortogonales gene-

radas a partir de una función generatriz tipo wavelet, o funciones no ortogonales generadas a partir de una función generatriz cualquiera [4].

Se dice que  $X$  es poco densa si ésta puede representarse como una combinación lineal de  $K$  componentes del diccionario  $\Psi$ , es decir:

$$X = \sum_{j=1}^K \theta_{n_j} \psi_{n_j} = \Psi \Theta, \quad (1)$$

donde  $\Theta = [\theta_1, \theta_2, \dots, \theta_z]^T$  es un vector con  $K$  componentes distintas de cero; el valor de  $\theta_i$  representa la contribución de la  $i$ -ésima componente del diccionario en la conformación de la señal  $X$  a lo largo de todo su dominio de definición.

Sea  $\Phi \in \mathfrak{R}^{M \times N}$  la matriz de proyección aleatoria, donde cada componente de dicha matriz,  $\phi_{i,j}$ , proviene de una distribución normal de media cero y varianza  $1/N$  o de una distribución de Bernoulli. Adicionalmente, se considera que las filas de la matriz de proyección  $\Phi$  sean incoherentes con las columnas de  $\Psi$ , es decir, ningún vector columna de  $\Psi$  puede ser una combinación lineal de unas pocas filas de  $\Phi$ . El conjunto de mediciones aleatorias, denotado por  $Y$ , se obtienen al proyectar la señal original  $X$  en la base creada por la matriz de proyección  $\Phi$ .

$$Y = \Phi X. \quad (2)$$

Como se puede observar en la Ec. (2), el vector  $M$ -dimensional  $Y$  contiene la proyección de la señal de interés sobre una base aleatoria. En general,  $M \ll N$ , por lo que se tiene entonces un proceso de compresión, dado que la información representada por el vector  $N$ -dimensional  $X$  ha sido compactada en sólo  $M$  componentes.

La teoría de CS establece que existe un factor  $c > 1$ , tal que con sólo  $M = f(cK)$  medidas incoherentes son necesarias para reconstruir, con una alta probabilidad, la señal  $X$  a partir del conjunto de proyecciones aleatorias  $Y$ , resolviendo para ello el problema de optimización [1, 2]:

$$\min \|\Theta\|_0 \quad \text{sujeito a } Y = V\Theta \quad (3)$$

donde  $\|\cdot\|_0$  representa la norma  $l_0$ , definida como el número de componentes del vector  $\Theta$  distintas de cero y  $V = \Phi\Psi$  es el diccionario holográfico dado por la proyección de cada átomo del diccionario,

donde la señal original es poco densa, en el espacio expandido por la matriz aleatoria de proyección.

La minimización de la norma  $l_0$  es un problema de optimización no-convexo de elevada complejidad, por lo que envuelve dificultades computacionales en su solución [2, 5]. A fin de superar tal inconveniente, el problema de minimización dado por la Ec. (3) se puede reformular mediante la solución de un problema no-cuadrático convexo conocido como *Basis Pursuit* [6], cuya solución rinde el vector  $\Theta$  de baja densidad y como consecuencia la señal original  $X$  es recuperable a partir de sus proyecciones aleatorias. Específicamente, el problema de optimización (3) usando *Basis Pursuit* se reduce a:

$$\min \|\Theta\|_1 \quad \text{sujeto a } Y = V\Theta \quad (4)$$

donde  $\|\cdot\|_1$  denota la norma  $l_1$  definida como:

$$\|\Theta\|_1 = \sum_{i=1}^Z |\theta_i|.$$

Obsérvese que en la formulación de CS sólo se requiere como información *a priori* el dominio en el cual la señal es poco densa. Obsérvese además que las proyecciones obtenidas son completamente independientes de la estructura de la señal.

Ahora bien, resolver el problema de optimización *Basis Pursuit* dado por Ec. (4) sigue siendo computacionalmente costoso y no apropiado para aplicaciones en tiempo real, por lo que se han desarrollado un conjunto de algoritmos iterativos, rápidos y eficientes, que a costa de un pequeño incremento en el número de proyecciones, resuelven el problema de optimización definido en (4). Entre estos algoritmos de reconstrucción se encuentran el *Matching Pursuit* [7], *Orthogonal Matching Pursuit* [8], *Regularized Orthogonal Matching Pursuit* [9], entre otros. En este trabajo se realiza el proceso de reconstrucción usando el algoritmo *Matching Pursuit*, el cual presenta un menor costo computacional en comparación a los demás algoritmos [8].

### Algoritmo de reconstrucción *Matching Pursuit* (MP)

El algoritmo de reconstrucción MP es un algoritmo iterativo computacionalmente sencillo que recupera la señal original siguiendo el procedimiento descrito a continuación. Inicialmente se determina cuál de los átomos del diccionario holográfico es el que aporta la mayor contribución a la definición de la señal proyectada, se encuentra un estimado de dicha contribución y se remueve el correspondiente átomo de la conformación de la señal proyectada, definiéndose así una señal residuo. Seguidamente, se busca de nuevo cuál de los átomos del diccionario holográfico está presente con mayor fuerza en la señal residuo y se le sustrae del mismo. Este procedimiento es iterativamente repetido hasta que el residuo contenga información irrelevante de la señal original. La Figura 1 muestra el algoritmo MP.

Como se puede observar en la Figura 1, en cada iteración se determina el grado de similitud entre la señal residuo (proyección aleatoria para la primera iteración) y cada átomo del diccionario usando el producto escalar como medida de similitud entre señales. Aquel átomo del diccionario holográfico que más se parezca a la señal residuo tendrá asociado el máximo producto escalar. Encontrado dicho átomo, se sustrae del residuo eliminándose así su contribución. Seguidamente, se actualiza el estimado del vector  $\Theta$  que en la medida en que progresa el algoritmo iterativo contendrá el estimado de la contribución de cada átomo en la definición de la señal. Finalmente, el estimado de la señal original se obtiene mediante la superposición de los distintos átomos encontrados en el proceso iterativo ponderados por su correspondiente contribución, es decir:  $X' = \Psi\Theta'$ .

Observe además que el algoritmo debería iterar tantas veces como átomos conformen a la señal<sup>1</sup>, sin embargo es posible que un mismo átomo sea seleccionado varias veces como el átomo que contribuye mayormente a la definición del residuo, en cuyo caso el número de iteraciones

1 Si la señal está conformada por la superposición de  $K$  componentes del diccionario, según Ec. (1) la señal es poco densa en dicho diccionario y se requeriría sólo  $K$  iteraciones para encontrar dichas componentes.

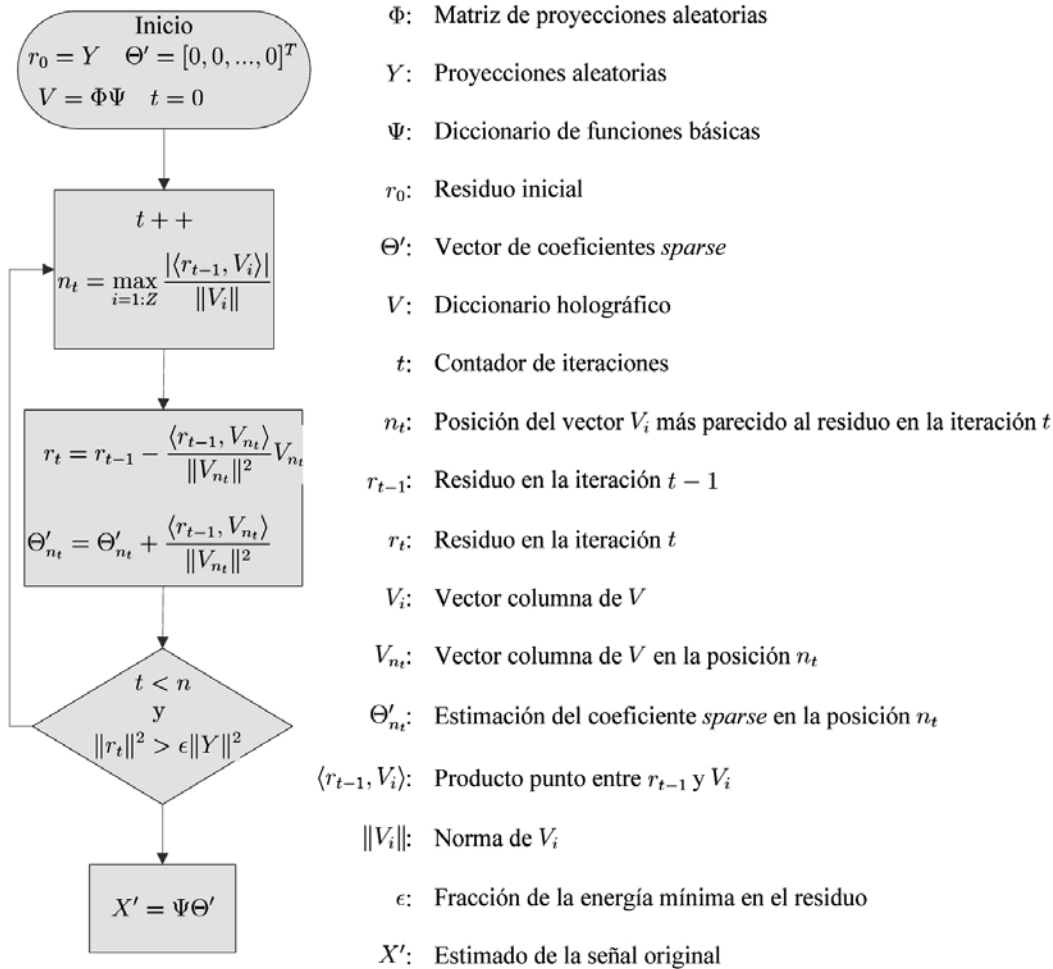


Figura 1. Algoritmo de reconstrucción MP.

crece, aumentando el tiempo y el costo computacional para reconstruir la señal dada. Como se mostró en [8], el costo computacional de este algoritmo está determinado principalmente por el paso en que se busca el átomo del diccionario holográfico que aporta la mayor contribución a la definición de la señal proyectada, siendo el costo computacional de este paso de orden  $\mathcal{O}(K \times M \times N)$ , donde  $K$  es el nivel de densidad de la señal original,  $M$  el número de proyecciones, y  $N$  la dimensión de la señal original.

### 3. Implementación en la tarjeta de desarrollo de la teoría de *Compressive Sensing*

Tal como se describió en la introducción, la implementación a nivel de hardware de la teoría

de *Compressive Sensing* requiere implantar tanto el proceso de reconstrucción (algoritmo iterativo MP) a nivel de la tarjeta de desarrollo, como el proceso de proyección que implica adquisición de la señal y su posterior proyección. El primer aspecto requiere implementar en la tarjeta de desarrollo el diccionario en el cual la señal en estudio es poco densa, presentándose aquí una primera limitación en cuanto a las componentes del diccionario (número de átomos) que en este caso está restringido por la capacidad de la memoria de la tarjeta de desarrollo, la cual en nuestro caso es de 256 k-palabra de 32 bit. La segunda consideración es el número de muestras que pueden manejarse en la tarjeta, limitado por la velocidad de muestreo de la tarjeta de adquisición (20 kHz para la tarjeta usada). Adicionalmente, a fin de reducir el número de iteraciones del algoritmo de

reconstrucción y facilitar el proceso de comparación entre la señal adquirida y los átomos del diccionario, deben implementarse rutinas que permitan sincronizar la señal adquirida con el diccionario.

### 3.1. Descripción del sistema de desarrollo

En esta sección se describe brevemente las características del sistema de desarrollo TMS320C30 tales como la velocidad del procesador, capacidad de memoria, puertos de entrada y salida, velocidad de muestreo, consumo de energía y lenguaje de programación soportado por la tarjeta de desarrollo, incluyendo las librerías disponibles para realizar las operaciones matemáticas necesarias en el sistema de desarrollo (Ver referencia [10] para mayores detalles).

**Procesador:** El procesador de la tarjeta TMS320C30 contiene unidades aritméticas de tipo entero y punto flotante, procesa palabras de  $2048 \times 32$  bit on-chip RAM, palabras de  $4096 \times 32$  bit on-chip ROM, posee una unidad de control de interfaces en serie y en paralelo y velocidad del reloj de 33,3 MHz, para un rendimiento de 16,7 millones de instrucciones procesadas por segundo.

**Memoria:** La tarjeta incluye dos áreas de memoria denominadas por el fabricante como áreas A y B. El área A se divide en tres bloques de 64 k-palabra de 32 bit cada uno para un total de 192 k-palabra de 32 bit. El área B, es un solo bloque de 64 k-palabra de 32 bit para un total de memoria disponible de 256 k-palabra de 32 bit para almacenamiento temporal de los datos.

**Interfaz analógica:** La tarjeta de desarrollo incluye cuatro canales analógicos, dos de conversión analógica-digital de 16 bits y dos para la conversión digital-analógica, soportando velocidades de muestreo de hasta los 200 kHz. Los canales de entrada incluyen circuitos de muestreo y retención, ambos protegidos a través de un filtro paso bajo del tipo Sallen-Key de cuarto orden. La velocidad de muestreo es de 20 kHz limitada por el ancho de banda de los filtros de protección.

**Programación:** SPOX es el software que incluye las librerías necesarias para la ejecución de instrucciones en el procesador de señales TMS320C30, sin embargo, es posible ejecutar di-

chas instrucciones sin el uso de las librerías utilizando programación de bajo nivel (código *assembler*). SPOX presenta un software de alto nivel para el hardware del DSP que no se requiere de una experiencia avanzada para su programación, ya que se puede programar usando líneas de código del C estándar. La librería que maneja el DSP se pueden agrupar en dos grandes módulos: Un **Módulo básico** el cual incluye el conjunto de funciones para la gestión de memoria y el manejo de buffer de entrada-salida, y un **Módulo matemático** el cual incluye las librerías que permiten realizar operaciones aritméticas, lógicas e incluso de procesamiento de señales tales como convolución, auto-correlación, transformada de Fourier, operación de filtrado vía enventanado, entre otras. Además, el módulo matemático permite realizar operaciones básicas vectoriales y matriciales como producto escalar entre vectores, suma y multiplicación de matrices, transposición, y asignación de matrices. Adicionalmente, posee funciones para la generación de números pseudo aleatorios que siguen una distribución gaussiana y una distribución uniforme entre 0 y 1.

### 3.2. Montaje experimental

El algoritmo de reconstrucción MP descrito en la Figura 1 se implementó en la tarjeta de desarrollo TMS320C30, optimizándose la implementación mediante el uso de las librerías que dispone la tarjeta. Se realizó una serie de pruebas que validaron el funcionamiento del algoritmo así como también la teoría de *Compressive Sensing* en forma experimental.

A fin de validar la implementación desarrollada, se realizan dos tipos de pruebas. Inicialmente se generan señales sintetizadas en la tarjeta de desarrollo para su posterior proyección aleatoria y reconstrucción mediante el algoritmo MP, ambos implementados en la tarjeta de desarrollo, tal como se muestra en la Figura 2(a).

Se aprovecha completamente el potencial de la tarjeta mostrándose tanto la señal sintetizada vía software como la reconstruida usando el algoritmo MP. Ambas señales se envían a los canales analógicos de salida del DSP y se visualizan usando un osciloscopio digital de doble canal y amplio ancho de banda. Adicionalmente, se guardan en archivos señales de interés para su posterior manipulación vía software de alto nivel (Matlab®).

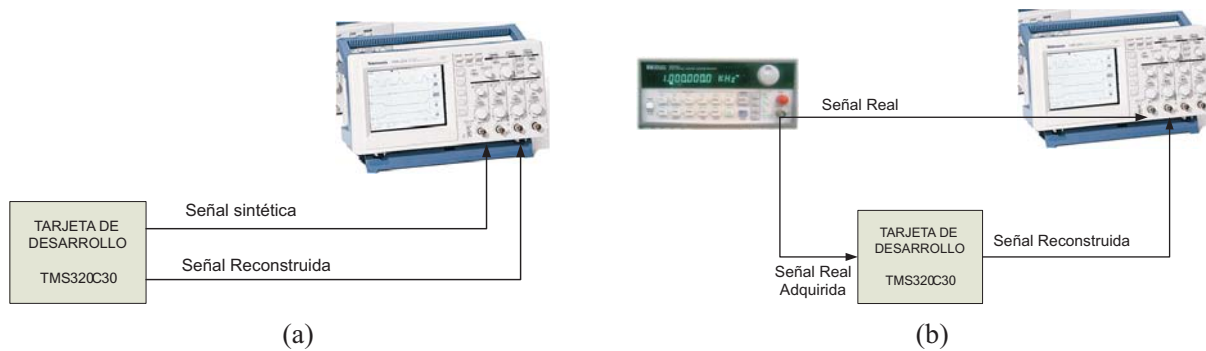


Figura 2. Montaje experimental.

Como un segundo grupo de pruebas, se adquieren señales reales poco densas en el dominio de la frecuencia, producidas por un generador programable. Se muestrean dicha señal a través de uno de los canales de entrada de la tarjeta de desarrollo, tal como se muestra en la Figura 2(b). Seguidamente, se proyecta aleatoriamente dicha señal y se reconstruye usando el algoritmo MP implementado en el DSP. Se valida el procedimiento de reconstrucción mediante la observación de la señal real generada y la señal reconstruida ambas visualizadas en un osciloscopio digital.

## 4. Resultados y análisis

### 4.1. Reconstrucción de señal sintetizada poco densa en el dominio de la frecuencia

Sea la señal

$$X(n) = 1,2 \operatorname{sen}\left(\frac{2\pi 5n}{10000}\right) - 1,1 \operatorname{sen}\left(\frac{2\pi 15n}{10000}\right) - 0,6 \operatorname{sen}\left(\frac{2\pi 45n}{10000}\right)$$

para  $n = 0, 1, \dots, 800$  sintetizada en la tarjeta de desarrollo. Dicha señal contiene tres componentes de frecuencia, por lo tanto es una señal poco densa en el dominio de la frecuencia. Considere la construcción del diccionario de Fourier dado por las funciones parametrizadas:

$$\begin{aligned} \psi(n, f, 0) &= C(f) \cdot \cos\left(\frac{2\pi f n}{N}\right) \\ \psi(n, f, 1) &= C(f) \cdot \operatorname{sen}\left(\frac{2\pi f n}{N}\right), \end{aligned} \quad (5)$$

donde  $n = 0, 1, 2, \dots, N$ ,  $C(f) = \frac{1}{\sqrt{N}}$  para  $f=0$ ,  $\sqrt{\frac{2}{N}}$  para otros valores de  $f$ , siendo  $N$  el tamaño de la señal sintetizada. Obsérvese que los átomos del diccionario, columnas de la matriz  $\Psi$ , se generan al definir valores discretos para el parámetro  $f$ . En nuestra aplicación  $f = 0, 5, 10, 15, \dots, F_0$ , siendo  $F_0$  la frecuencia máxima contenida en el diccionario. Para nuestro caso esta frecuencia máxima fue fijada en 100 Hz. Así, el diccionario está conformado por 42 átomos usándose 33.6 k-palabra de 32 bit de la memoria del dispositivo. La cantidad de átomos en el diccionario puede ser mucho mayor si se realiza la implementación en una tarjeta con mayor capacidad de memoria que la empleada en este trabajo. A modo de prueba los 42 átomos disponibles son suficientes para corroborar la teoría *Compressive Sensing* y el algoritmo *Matching Pursuit (MP)*, y demostrar que sí es posible recuperar una señal poco densa a partir de un conjunto muy reducido de proyecciones aleatorias.

La Figura 3(a) muestra de arriba hacia abajo la señal sintetizada, la señal reconstrucción y el error como la diferencia entre las dos señales anteriores, respectivamente. La Figura 3(b) muestra las proyecciones de la señal y el error de reconstrucción en modo expandido. Observe con particularidad el orden de magnitud de la señal de error. Estas dos últimas figuras se generaron a partir de los registros guardados en archivos y posteriormente leídos usando Matlab®.

### 4.2. Reconstrucción de señales reales adquiridas por el sistema de desarrollo

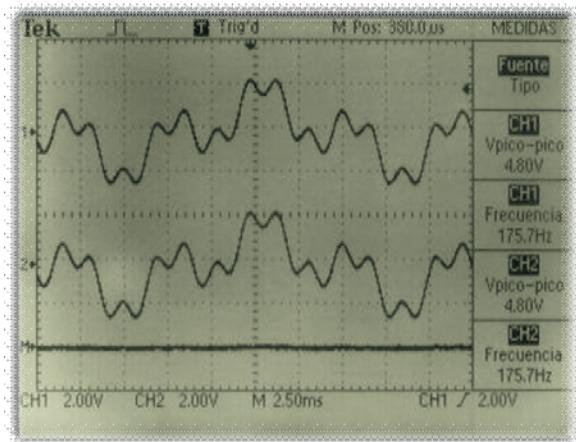
Las señales usadas para explorar el potencial de la teoría de *Compressive Sensing* son ge-

neradas desde el generador programable HP33120A el cual se programa adecuadamente a fin de producir señales que cumpla la propiedad de poca densidad. En particular, es de interés la generación de señales poco densas en el dominio de la frecuencia como por ejemplo una señal conformada por la suma de varias señales sinusoidales a frecuencias distintas y señales periódicas cuadradas o triangulares, las cuales como es bien sabido puede descomponerse mediante el uso de serie de Fourier como la sumas de sinusoides [11] y por tanto resultan señales poco densas en el dominio de la frecuencia.

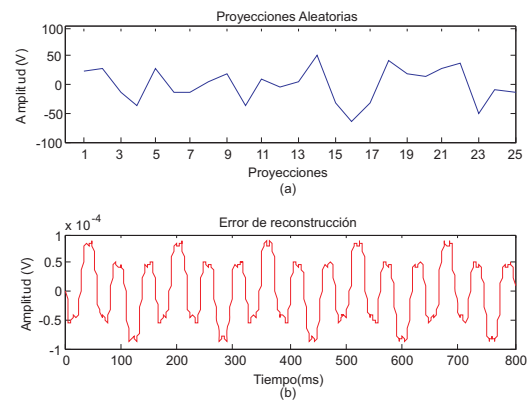
La Figura 4(a) muestra la señal original proveniente del generador, su reconstrucción y el

error que se comete al recuperar la señal a partir del conjunto de proyecciones. La señal original es adquirida por la tarjeta, proyectada usando la base aleatoria y reconstruida a partir de tan sólo 10 proyecciones aleatorias. La Figura 4(b) muestra las proyecciones y el error ampliado que se comete en la reconstrucción.

Observe en la Figura 4(a) la limitación de la frecuencia de trabajo del convertido digital analógico donde se observa el efecto de cuantización a 16 bits y el valor constante de la señal reconstruida entre sucesivos instantes de muestreo. Obsérvese además, en la Figura 4(b), que la señal mostrada no sólo refleja el error de reconstrucción sino además el error típico de cuantización pro-

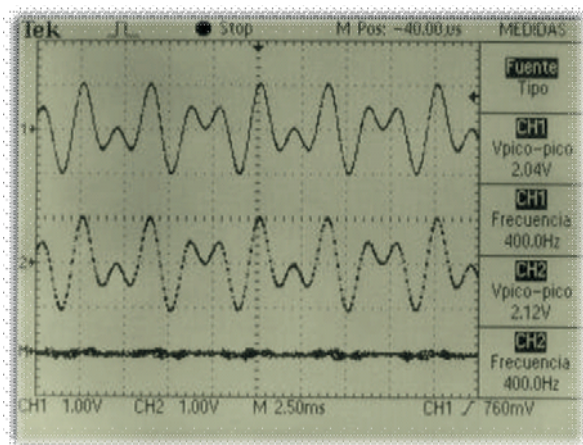


(a)

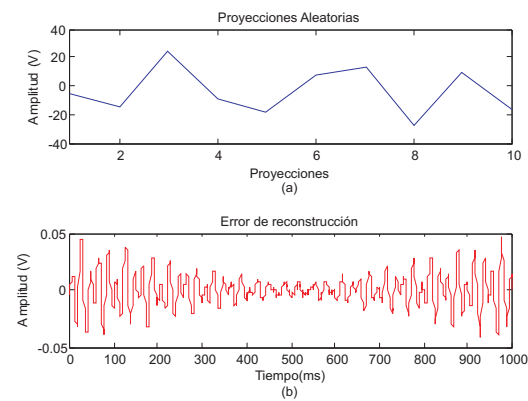


(b)

Figura 3. Reconstrucción de señal sintética poco densa en el dominio de la frecuencia a partir de 25 proyecciones aleatorias.



(a)



(b)

Figura 4. Reconstrucción de señal real poco densa en el dominio de la frecuencia a partir de 10 proyecciones aleatorias.

pio de un proceso de conversión analógico-digital-analógico.

Finalmente, considere una señal  $X(t)$  triangular anti-simétrica y periódica con valor medio cero y frecuencia fundamental 100 Hz generada por el Generador de Señales. Como es conocido dicha señal puede ser aproximada por la superposición de señales sinusoidales con frecuencia múltiplos de la frecuencia fundamental. Específicamente,  $X(t) \approx \sum_{m=1}^K b_m \text{sen}(2\pi 100mt)$ , donde  $b_m$

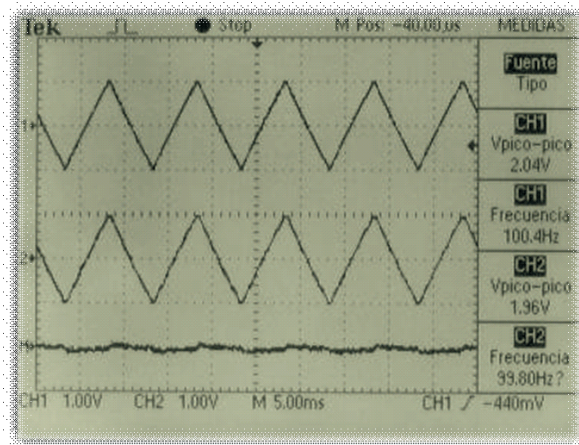
son los coeficientes de Fourier que para el caso de la señal triangular es dado por  $b_m = \frac{8V_{\max}(-1)^{(m-1)/2}}{(m^2\pi^2)}$  [11]. Para un valor de

$K > 10$ , dicha aproximación es bastante buena, por lo que se puede considerar que  $X(t)$  es poco densa en el dominio de la frecuencia y por lo tanto la teoría de *Compressive Sensing* puede ser aplicada.

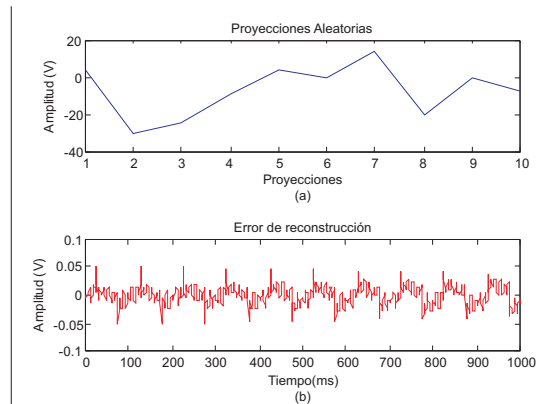
En la Figura 5 se muestra los correspondientes resultados. Como se puede observar el error cometido en el proceso de reconstrucción es elevado en los picos de la señal triangular debido a las discontinuidades existentes en estos puntos que implica altas frecuencia que no se encuentran definidos como parte del diccionario. Para este caso el diccionario contiene un conjunto finito de 21 componentes de frecuencia donde se encuentran contenidos los 5 primeros armónicos diferentes de cero de la señal original.

Al igual que en las pruebas anteriores la cantidad de armónicos en el diccionario está limitado a la capacidad de almacenamiento de la tarjeta. Observe que la señal es poco densa en el dominio de Fourier y de manera aproximada se puede reconstruir una señal triangular ó cuadrada a partir de sus 5 primeros armónicos.

La Tabla 1 muestra los coeficientes del vector  $\Theta'$  que arroja el algoritmo *Matching Pursuit*



(a)



(b)

Figura 5. Reconstrucción de señal triangular a partir de 50 proyecciones aleatorias.

Tabla 1  
Coeficientes de la señal triangular real para 4 realizaciones de MP

Realización de la Matriz de proyección	$i$ -ésimo coeficiente de reconstrucción ( $\Theta_i$ )								
	1	2	3	4	5	6	7	8	9
1	0,876	0	-0,051	0	0,046	0	-0,016	0	0
2	0,887	0	-0,107	0	0,037	0	-0,016	0	0,009
3	0,883	0	-0,090	0	0,030	0	-0,019	0	0
4	0,883	0	-0,097	0	0,030	0	-0,015	0	0,012
Coeficientes de Fourier ( $b_m$ )	0,811	0	-0,090	0	0,032	0	-0,017	0	0,010



para la reconstrucción de la señal triangular para diferentes realizaciones de la matriz de proyección. Se muestran los coeficientes asociados a los primeros 10 armónicos, todos los demás son cero o prácticamente despreciables. Como es de esperarse estos coeficientes se aproximan a los 10 primeros coeficientes dados por la Serie de Fourier,  $b_m = \frac{8V_{\max}(-1)^{(m-1)/2}}{m^2\pi^2}$  para  $m$  impar, en consecuencia la señal reconstruida se aproxima a la señal triangular.

Finalmente cabe mencionar que se hicieron experimentos con señales poco densas en el dominio del tiempo y en el dominio wavelet corroborándose experimentalmente la aplicación de la teoría, por razones de espacio no son incluidos en el presente artículo. Ver referencia [12] para mayores detalles.

## 5. Conclusiones

En este trabajo se comprobó de manera práctica el potencial que tiene la teoría de CS en la reconstrucción de señales poco densas a partir de un conjunto limitado de proyecciones aleatorias. Se implementó el algoritmo MP en una tarjeta de desarrollo, se adquirió las señales reales poco densas en la frecuencia y se usó el algoritmo MP para su correspondiente reconstrucción. Se mostró que, con pocas iteraciones, el algoritmo MP detecta las componentes sinusoidales que conforman la señal de interés. El potencial de uso de una implementación a nivel de hardware como la desarrollada en este trabajo emerge en el escenario donde las proyecciones aleatorias son transmitidas a través de un canal de comunicación y el receptor requiere reconstruir la señal para su posterior manipulación. Al realizar tal operación a nivel de hardware (tarjeta de desarrollo) evita el uso de recursos de software y acelera el proceso de reconstrucción.

## Agradecimiento

Los autores desean expresar su gratitud al Fondo Nacional de Ciencia, Tecnología e Innovación (FONACIT) bajo el Proyecto G-2005000342 y al Consejo de Desarrollo Científico, Humanístico y Tecnológico de la Universidad de Los Andes (CDCHT-ULA) bajo el proyecto I-1109-08-02-F por el soporte económico dado en el desarrollo de

presente trabajo. Igualmente, agradece al Laboratorio de Investigación y Desarrollo en Automatización e Instrumentación (LABIDAI) por ceder sus equipos y espacios para el desarrollo de este trabajo.

## Referencias bibliográficas

1. Candès, E., Romberg, J., y Tao, T. Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information. *IEEE Transactions On Information Theory*, Vol. 52 N° 2, (2006) 489-509.
2. Donoho, D. Compressed Sensing. *IEEE Transactions On Information Theory*, Vol. 52, N° 4, (2006) 1289-1306.
3. Duarte Marco, Davenport Mark, Takhar Dharmpal, Laska N. Jasson, Sun Ting, Kelly F. Kevin, y Baraniuk Richard G. Single-Pixel Imaging via Compressive Sampling. *IEEE Signal Processing Magazine*. Vol. 25, N° 2, (2008) 83-91.
4. Paredes J.L., Arce, G y Wang Z. Ultra-Wideband Compressed Sensing: Channel Estimation. *IEEE Journal of Selected Topics in Signal Processing*, Vol. 1,(3) (2007) pp. 383-395.
5. Mancera, L., y Portilla, J.  $L_0$ -Norm-Based Sparse Representation Through Alternate Projections. *Proc. of the IEEE ICIP 2006*, pp. 2089-92, Atlanta, GA, USA.
6. Chen S. S., D. L. Donoho, y M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, Vol. 20, N° 1, (1998) pp. 33-61.
7. Mallat, S. G., y Zhang, Z. Matching Pursuit With Time-Frequency Dictionaries. *IEEE Transactions On Signal Processing*, Vol. 41 (12), (1993) 3397-3415.
8. Tropp, J. A., y Gilbert, A. C. Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit. *IEEE Transactions On Information Theory*, Vol. 53, N° 12, (2007) 4655-4666.
9. Needell D. y Vershynin R.. Uniform Uncertainty Principle and Signal Recovery via Regularized Orthogonal Matching Pursuit. *Found. Comput. Math.*, Vol. 9, N° 3, pp. 317-334.

10. LSI - Loughborough Sound Images Ltd. (1989). SPOX Application Programming Reference Manual, TMS320C30 PC System Board Technical Reference Manual. TMS320C30 PC System Board User Guide. Spectron Micro System.
11. Proakis, J. G., y Manolakis, D. G. Tratamiento Digital de Señales. 3ra Edición, Madrid, España: Prentice Hall. 1998.
12. Velazquez Alexis, Paredes José L. y Vilorio Francisco. Implementación a Nivel de Hardware de la Teoría de Compressive Sensing. Proyecto de Grado. Escuela de Ing. Eléctrica. Universidad de Los Andes. Mérida. Jun. 2008.

Recibido el 10 de Enero de 2010

En forma revisada el 26 de Septiembre de 2011