



# ESTIMADOR DE CANAL BASADO EN SENSADO COMPRESIVO Y LDPC PARA OFDM USANDO SDR

## COMPRESSIVE SENSING BASED CHANNEL ESTIMATOR AND LDPC THEORY FOR OFDM USING SDR

Anthony Yanza-Verdugo<sup>1</sup>, Christian Pucha-Cabrera<sup>1</sup> y Juan Inga-Ortega<sup>2,\*</sup>

### Resumen

Este trabajo propone la aplicación de un estimador de canal basado en sensado compresivo (CS, del inglés *Compressive Sensing*) sobre un sistema que usa multiplexación por división de frecuencias ortogonales (OFDM, del inglés *Orthogonal Frequency Division Multiplexing*) usando dispositivos de radio definido por *software* (SDR, del inglés *Software Defined Radio*). La aplicación de la teoría de CS se da a través del uso de algoritmos de reconstrucción dispersa como *Orthogonal Matching Pursuit* (OMP) y *Compressive Sampling Matching Pursuit* (CoSaMP) con el fin de aprovechar la naturaleza dispersa de las subportadoras piloto usadas en OFDM optimizando el ancho de banda del sistema. Además, para mejorar el rendimiento de estos algoritmos, se utiliza el concepto de la matriz de comprobación de paridad dispersa que se implementa en el despliegue de códigos de comprobación de paridad de baja densidad (LDPC, del inglés *Low Density Parity Check*) para obtener una matriz de sensado que mejore la propiedad de restricción isométrica (RIP, del inglés *Isometric Restriction Property*) perteneciente al paradigma de CS. El documento muestra el modelo implementado en los equipos SDR analizando la tasa de error de bit y la cantidad de símbolos piloto usados.

**Palabras clave:** estimación de canal, LDPC, OFDM, SDR, sensado compresivo.

### Abstract

This work proposes the application of a channel estimator based on Compressive Sensing (CS) over a system that uses Orthogonal Frequency Division Multiplexing (OFDM) using Software Defined Radio (SDR) devices. The application of the CS theory is given through the use of sparse reconstruction algorithms such as Orthogonal Matching Pursuit (OMP) and Compressive Sampling Matching Pursuit (CoSaMP) in order to take advantage of the sparse nature of the pilot subcarriers used in OFDM, optimizing the bandwidth of system. In addition, to improve the performance of these algorithms, the sparse parity checking matrix concept is used, which is implemented in the deployment of low density parity check codes (LDPC) to obtain a sensing matrix that improves the isometric restriction property (RIP) belonging to the CS paradigm. The document shows the model implemented in the SDR equipment and analyze the bit error rate and the number of pilot symbols used.

**Keywords:** Channel Estimation, LDPC, OFDM, SDR, Compressive Sensing.

<sup>1,\*</sup>Carrera de Ingeniería Electrónica / GITEL, Universidad Politécnica Salesiana – Ecuador.

<sup>1</sup><http://orcid.org/0000-0002-1710-3052>, <sup>1</sup><http://orcid.org/0000-0002-4734-7218>

<sup>2</sup>Carrera de Telecomunicaciones / GITEL, Universidad Politécnica Salesiana – Ecuador.

Autor para correspondencia ✉: jinga@ups.edu.ec <sup>1</sup><http://orcid.org/0000-0003-2580-9677>

Recibido: 31-10-2019, aprobado tras revisión: 09-12-2019

Forma sugerida de citación: Yanza-Verdugo, A.; Pucha-Cabrera, C. e Inga-Ortega, J. (2020). «Estimador de canal basado en sensado compresivo y LDPC para OFDM usando SDR». INGENIUS. N.º 23, (enero-junio). pp. 74-85. DOI: <https://doi.org/10.17163/ings.n23.2020.07>.

## 1. Introducción

La técnica de transmisión más implementada en la actualidad en redes inalámbricas es multiplexación por división de frecuencias ortogonales (OFDM, del inglés *Orthogonal Frequency Division Multiplexing*) debido a sus ventajas en la transmisión de datos de alta velocidad a través de canales selectivos en frecuencia. Así, el uso de OFDM ha tenido alto rendimiento debido a su alta eficiencia en el uso del espectro radio eléctrico y su robustez al retardo por multirayos [1, 2]. Además, a través del uso de un prefijo cíclico se reduce la interferencia entre símbolos (ISI, del inglés *Intersymbol Interference*) y la interferencia entre portadoras (ICI, del inglés *Inter Carrier Interference*) permitiendo reemplazar el uso de complejos ecualizadores en el dominio del tiempo por un simple ecualizador en el dominio de la frecuencia [3]. En este sentido, para canales de radio selectivos en frecuencia y que varían en el tiempo en sistemas de comunicaciones móviles de banda ancha, es necesario aplicar estimación de canal confiable para implementar detección coherente [1].

Hay tres tipos de estimadores de canal: estimación ciega, estimación semiciega y modulación asistida por pilotos (PSAM, del inglés *Pilot Symbol Assisted Modulation*). PSAM usa símbolos piloto que son conocidos tanto por el receptor como por el transmisor en diferentes posiciones para obtener una aproximación del canal a cambio de perder eficiencia espectral. Por su parte, la estimación ciega está enfocada en procesos estadísticos con alto nivel matemático presentando mejoras con respecto a la eficiencia espectral, aunque no es muy utilizado en la actualidad debido a su alta complejidad, velocidad lenta para converger y bajo rendimiento [4]. La estimación semiciega usa una combinación de los dos anteriores para llegar a un convenio entre complejidad y eficiencia; para ello se utilizan tanto secuencias de entrenamiento como modelos estadísticos de canal [5].

De acuerdo con lo antes mencionado, este trabajo propone el uso de PSAM para la estimación de canal, donde la ganancia del canal y la distorsión de fase se obtienen a partir de la señal recibida en las posiciones de los símbolos piloto [1]. Estos símbolos piloto también son usados para mejorar la sincronización de tiempo y frecuencia en el sistema de comunicación. Sin embargo, para incrementar el rendimiento, es necesario del uso de un mayor porcentaje de símbolos piloto e implica que ciertas subportadoras no transmitan información [2].

Además, este trabajo con el objeto de mejorar la utilización del ancho de banda reduce el número de subportadoras piloto a través del uso de la teoría de sensado compresivo (CS, del inglés *Compressive Sensing*) en la estimación de canal. Esto se puede aplicar ya que es posible estimar el canal aprovechando la teoría de CS, al considerar que el número de símbolos piloto

es disperso en relación con el número total de símbolos que conforman un mensaje OFDM obteniendo una versión comprimida del canal correspondiente a los efectos del mismo sobre las subportadoras piloto [6].

También, se optimiza el proceso en la implementación sobre equipos de radio definido por *software* (SDR, del inglés *Software Defined Radio*) a través del uso de la matriz de comprobación de paridad de baja densidad (LDPC, del inglés *Low Density Parity Check*) combinado con el paradigma de CS.

A partir de aquí, el artículo se ordena de la siguiente forma: en la sección 2 se discute el modelado matemático de OFDM, CS y la aplicación de matrices LDPC con CS. En la sección 3 se describe la implementación en los dispositivos SDR usando las matrices LDPC con CS. La sección 4 presenta los resultados obtenidos, el análisis del ancho de banda y el análisis de la tasa de error de bit (BER, del inglés *Bit Error Rate*). Finalmente, en la sección 5 se dan las conclusiones y recomendaciones para trabajos futuros.

## 2. Modelado matemático

La implementación se desarrolla a partir del modelo matemático de OFDM, donde un símbolo OFDM es representado por  $S(q) \in \mathbb{R}^Q$ , así un conjunto  $Q$  de símbolos de información es transmitido por el símbolo  $j$  de OFDM, cada uno con una subportadora  $q$  es  $S_j(q)$ , por tanto  $[S_j(0), \dots, S_j(q), \dots, S_j(Q-1)]^T$  representa el vector de símbolos de información transmitidos en el  $j$ -ésimo símbolo OFDM [2, 7].

Cada símbolo OFDM usa el flujo serial de símbolos de información convirtiéndolos en  $Q$  flujos paralelos para luego insertar los símbolos piloto necesarios en la estimación del canal según el esquema de transmisión propuesto. Además, se adicionan subportadoras en cero para establecer bandas de guarda entre cada símbolo OFDM y así evitar la interferencia de canales adyacentes. La cantidad de subportadoras de un símbolo OFDM se evidencia en (1).

$$Q = I + P + Z \quad (1)$$

Donde  $I$  corresponde al número de subportadoras con información,  $P$  es el número de subportadoras piloto y  $Z$  es el número de subportadoras en cero para un símbolo OFDM. De acuerdo con esto, la Ecuación (2) describe el formato de un símbolo OFDM correspondiente al dominio de la frecuencia, luego se aplica la transformada inversa de Fourier (IDFT, del inglés, *Inverse Discrete Fourier Transform*) para obtener el símbolo en el dominio del tiempo tal como se describe en la Ecuación (3). La distribución del relleno de ceros se detalla en [8].

$$S_z(q) \triangleq \begin{cases} S(q), & \frac{Q-(I+P)}{2} \leq q \leq \frac{Q-(I+P)}{2} - 1 \\ 0, & \text{cualquier otro caso} \end{cases} \quad (2)$$

$$S_z = F^H S_z \quad (3)$$

La reducción de ISI e ICI se da por el uso del llamado prefijo cíclico (CP) que consiste en una extensión cíclica del símbolo OFDM. El CP es tan largo como el retardo de propagación esperado y los efectos de ambos se encuentran en [9]. Por otra parte, la señal recibida para OFDM puede ser descrito como en la Ecuación (4).

$$y_j = H_j s_j + z_j \quad (4)$$

El vector de símbolos de información recibidos es  $y_j = [y_j(0), \dots, y_j(Q-1)]^T$  en el  $j$ -ésimo símbolo OFDM,  $z_j$  es el ruido gaussiano y  $H_j$  denota el valor del canal para el  $j$ -ésimo símbolo OFDM, que es obtenido mediante el bloque de estimación de canal [7].

El receptor usa el algoritmo de Van de Beek que aprovecha el prefijo cíclico para establecer el inicio y el fin de un símbolo OFDM. Este algoritmo también permitirá corregir los efectos del canal en posibles desplazamientos de fase y evitar una pérdida de ortogonalidad en las subportadoras [10].

A continuación, se retira el prefijo cíclico y se aplica la transformada discreta de Fourier (DFT, del inglés, *Discrete Fourier Transform*) para recuperar los símbolos de información. Previo a la recuperación de los símbolos de información se estima el canal luego de remover el relleno de ceros y extraer los símbolos piloto. Los datos ecualizados se obtienen a través de la operación indicada en la Ecuación (5) [11]. En donde  $\hat{H}_e(q)$  es el canal estimado. La Tabla 1 presenta una lista de todas las variables usadas a lo largo de este trabajo.

$$S_e = \frac{Y(q)}{\hat{H}_e(q)} \quad q = 0, 1, \dots, Q-1 \quad (5)$$

## 2.1. Sensado compresivo

Desde otra perspectiva, el paradigma de CS considera una señal «dispersa»  $x$  que posee solo  $K$  elementos diferentes de cero con  $K \ll N$  y donde  $x \in \mathbb{R}^N$ . Entonces, a través de una matriz  $\Phi$  de  $M \times N$  con  $N > M$ , llamada matriz de sensado, se busca concentrar la información más relevante de  $x$  a través del uso de otro vector  $y \in \mathbb{R}^M$ . Si esto no se cumple, se reduce la capacidad de reconstruir  $x$  a partir de  $y$ ; a esta restricción se la conoce como condición de espacio nulo (NSP, del inglés *Null Space Property*) [11, 12].

No es fácil encontrar señales consideradas «dispersas», no obstante, es posible encontrar una aproximación a este concepto que se da cuando la señal  $x$  puede ser «comprimible» en alguna base vectorial diferente a la original [11–13]. Por tanto,  $x = \Psi\alpha$ , donde  $\Psi$  (también llamada matriz diccionario de  $x$ ) corresponde a la base vectorial en la que se proyecta  $x$ , y  $\alpha$  corresponde

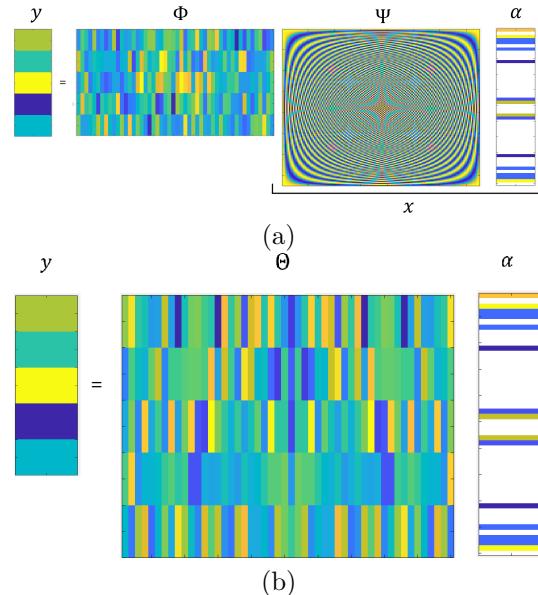
a los átomos de  $x$  dentro del dominio de  $\Psi$ . Así se tiene que la señal original  $x$  puede ser representada a través de la combinación lineal que propone en (6) [14]. El diccionario por ejemplo puede ser de Fourier, DCT o de Wavelet [11, 15, 16].

$$x = \Psi\alpha = \sum_{i=1}^N \alpha_i \psi_i \quad (6)$$

La compresión de la señal  $x$  en la señal  $y$  se puede obtener a través de (7) donde  $\Theta = \Phi\Psi$ . La Figura 1(a) y la Figura 1(b) exponen de forma gráfica el concepto de CS [13, 17].

$$y = \Phi x = \Phi\Psi\alpha = \Theta\alpha \quad (7)$$

La literatura recomienda que  $\Phi$  sea aleatoria para cumplir la propiedad de restricción isométrica (RIP, del inglés *Isometric Restriction Property*) y así  $x$  pueda ser reconstruida a partir de  $y$  [16, 18]. Las matrices de sensado en general se pueden obtener de un proceso aleatorio con distribución normal, consiguiendo un buen rendimiento durante la reconstrucción de la señal, pese a que, estas distribuciones pueden llegar a demandar muchos recursos [15, 16].



**Figura 1.** Principio de operación de sensado compresivo: (a) proceso de medición en CS con una matriz de sensado aleatoria  $\Phi$  y matriz de medición  $\Psi$ . (b) proceso de medición con  $\Theta = \Phi\Psi$ .

Debido a que  $N > M$  y que  $N \gg K$ , el proceso de reconstrucción de  $x$  se convierte en un problema de optimización expuesto en la Ecuación (8) donde se busca obtener los átomos  $\alpha$  que minimicen el error de  $\|y - \Theta\alpha\|_0$ .

**Tabla 1.** Lista de variables

Variables	Descripción
$Q$	Total de subportadoras en el símbolo OFDM.
$S_j(q)$	Subportadora $q$ en el $j$ -ésimo símbolo OFDM.
$I$	Total de subportadoras con información.
$P$	Total de subportadoras piloto.
$Z$	Total de subportadoras en cero.
$S_z(q)$	Símbolo OFDM con zero-padding en la subportadora $q$ .
$F$	Matriz de transformada de Fourier.
$s_z$	Símbolo OFDM con relleno de ceros en el tiempo.
$y_j$	Vector de símbolos recibidos en el $j$ -ésimo símbolo OFDM en el tiempo.
$H_j$	Vector de canal en el $j$ -ésimo símbolo OFDM.
$s_j$	$j$ -ésimo símbolo OFDM transmitido en el tiempo.
$Z_j$	Ruido gaussiano complejo.
$Y(q)$	Símbolo OFDM en frecuencia, removido el prefijo cíclico, el relleno de ceros y sin pilotos.
$S_e$	Símbolos equalizados en la frecuencia.
$H_e(q)$	Canal estimado en la frecuencia.
$x$	Vector de datos de longitud $N$ .
$y$	Vector comprimido de longitud $M$ .
$\Phi$	Matriz de compresión o sensado, Phi.
$N$	Longitud total del vector de datos a comprimir.
$K$	Total de elementos diferentes de cero.
$M$	Longitud total del vector de datos comprimido.
$\Psi$	Matriz de medición o diccionario, Psi.
$\alpha$	Vector de coeficientes dispersos, alpha.
$\Theta$	Matriz de sensado en su forma completa, Theta.
$W_c$	Cantidad de "1"s lógicos por columna en $H_{LDPC}$
$W_r$	Cantidad de "1"s lógicos por fila en $H_{LDPC}$
$H_{LDPC}$	Matriz de paridad de LDPC.
$A$	Matriz de $D \times E$ elementos.
$B$	Matriz de $D \times D$ elementos.
$D$	Cantidad de bits de paridad en la codificación.
$E$	Cantidad de bits de información en la codificación.
$G$	Total de bits de información + paridad ( <i>codeword</i> ).
$H_i$	Matriz de paridad de LDPC irregular lado izquierdo.
$SC$	Matriz de paridad de LDPC escalera lado derecho.
$T_r$	Matriz de paridad de LDPC triángulo lado derecho.
$\hat{H}_p$	Canal de pilotos.
$Y_p$	Símbolos pilotos recibidos.
$S_p$	Símbolos pilotos conocidos.
$N_f$	Separación frecuencial entre pilotos.
$\Phi_{LDPC}$	Matriz de sensado construido con base LDPC.

$$\alpha = \arg \min \|\alpha\|_0 \text{ s.t. } \Phi \Psi \alpha \quad (8)$$

A partir de  $\alpha$  se puede reconstruir  $x$  usando la Ecuación (6). Sin embargo, el problema de optimización no tiene solución ya que es del tipo NP Hard [11–13]. En este aspecto, los algoritmos de reconstrucción dispersa buscan relajar el problema de optimización para obtener una solución pseudoóptima. Además, pueden mejorar su rendimiento si se usa una matriz de sensado adecuada que permita la preservación de la información y garantice la reconstrucción de la señal original de manera única [15, 16, 19–22].

Entre los algoritmos de reconstrucción dispersa más populares usados en la aplicación de CS para la estimación de canal, destaca *Orthogonal Matching Pursuit* (OMP) perteneciente al tipo de algoritmos basados en búsqueda codiciosa. Así, OMP se basa en aproximaciones sucesivas de los

coeficientes  $\alpha$ , identificando de manera iterativa el soporte de la señal hasta alcanzar un criterio de convergencia [19, 23]. El algoritmo que lo describe se indica en la Tabla 2.

**Tabla 2.** Algoritmo OMP**Algoritmo 1:** Algoritmo OMP (Orthogonal Matching Pursuit)

---

Paso 1 **Entrada:**  $Y = \{y_1, \dots, y_M\}$ ; Entrada Comprimida  
 $\Theta_{N,N}$ ; Matriz de sensado completa  
 $k$ ; Número de repeticiones

Paso 2 **Salida:**  $\alpha_N$

Paso 3 **Inicialización:**  $res_M = Y$ ;  
 $indx = \phi$ ;

Paso 4 **for**  $iter = 1$  to  $k$ :  
 $\lambda = \arg \max |\Theta^T \times res_M|$ ;  
 $indx = indx \cup \lambda_M$ ;  
 $\alpha_N = \phi$ ;  
 $\alpha_N(indx) = \text{pinv}(\Theta(:, indx)) \times Y$ ;  
 $res = Y - \Theta \times \alpha_N$ ;  
**endfor**

Paso 5 **Retornar:**  $\alpha_N$

---

Otro algoritmo de reconstrucción dispersa empleado en este caso de aplicación de CS es *Compressive Sampling Matching Pursuit* (CoSaMP) descrito en [24]. El algoritmo aproxima la señal objetivo de forma iterativa y en cada iteración la aproximación actual obtiene un residual actualizando las muestras para que reflejen el residuo actual. Estas muestras se utilizan para identificar los componentes grandes para estimar la aproximación utilizando mínimos cuadrados. Este proceso se repite hasta que se haya encontrado la energía recuperable en la señal. El algoritmo que lo describe se indica en la Tabla 3 [25].

**Tabla 3.** Algoritmo CoSaMP**Algoritmo 2:** Algoritmo CoSaMP (Compressive Sampling Matching Pursuit)

---

Paso 1 **Entrada:**  $x \in R^N$ ; Entrada Comprimida  
 $\Theta \in R^{m \times N}$ ; con  $N > m$ ; Matriz de sensado completa  
 $k$ ; Número de repeticiones

Paso 2 **Salida:**  $S_{est}$

Paso 3 **Inicialización:**  $u = \Theta x$ , Vector de medición;  
 $\Omega = \{1, 2, \dots, N\}$  Índice de columnas de  $\Theta$ ;

Paso 4 **for**  $iter = 1$  to  $k$ :  
 $\Omega_k \subset \Omega \rightarrow \Theta_{\Omega_k}$ ;  
 $\Omega_{k+1} = J_s^*$  donde  $J_s^* \subset \Omega$ ;  
 $J = \text{Supp}\{T_1(|\Theta \times r_k|; \tau_{k,1})\}$  con  $|J| \leq 2s$ ;  
donde:  $T_1$  es una función de umbral acorde a:  $\tau_{k,1} \geq 0$ ;  
 $r_k$  : residual;  
con  $\Omega_k \cup J$  :  $\|u - \Theta I_{\Omega_k \cup J} b\|_2$  minimizado;  
donde:  $I_{\Omega_k \cup J}$  es una matriz diagonal  $N \times N$ ;  
 $i \in \Omega_k \cup J$ ;  
 $J_s^* = \text{Supp}\{T_2(|b|; \tau_{k,2})\}$ ;  
donde:  $T_2$  es una función de umbral;  
 $\tau_{k,2} \geq 0$  máximo de  $s$  elementos de  $b$  que son retenidos;  
 $S_{est} = b(J_s^*)$ ;  
 $r_{k+1} = u - \Theta I_{\Omega_{k+1}} b$ ;  
**endfor**

Paso 5 **Retornar:**  $S_{est}$

---

Estos algoritmos son aptos para ser usados en aplicaciones o reconstrucciones donde su funcionamiento se dé en tiempo real debido a su bajo coste computacional.

Debido a que la matriz de sensado debe satisfacer la propiedad RIP, se considera el uso de las matrices de comprobación de paridad usadas en el diseño de codificadores de canal en los códigos LDPC ya que cumplen el criterio de RIP y son deterministas, por ello, consumen menos recursos mostrándose como una buena opción para la matriz de medición en CS [20–22]. Esto permitirá no usar un patrón de pilotos del tipo peine o bloque como es comúnmente utilizado en sistemas OFDM con PSAM. Conforme a esto, se busca usar un patrón de pilotos pseudoaleatorio disperso empleando una matriz de comprobación de paridad LDPC [5].

## 2.2. Matriz de comprobación de paridad LDPC

La matriz de comprobación de paridad  $H_{LDPC}$  define las relaciones entre los distintos símbolos de codificación (símbolos fuente y símbolos de paridad). Existen dos tipos, las matrices regulares que tienen un número fijo  $W_c$  de «1»s lógicos por columnas y un número fijo  $W_r$  de «1»s lógicos por filas. El segundo es el caso de las matrices irregulares cuyos valores de «1»s lógicos por filas es  $W_r$  y por columnas  $W_c$  con  $W_r \neq W_c$ . La matriz está formada por elementos con valores «0» y «1», y es dispersa ya que la mayoría de elementos son nulos [26]. Se describe esta matriz en la Ecuación (9).

$$H_{LDPC} = [A|B] \quad (9)$$

La matriz A tiene dimensiones  $D \times E$  y la matriz B es de  $D \times D$ . Donde E corresponde a la cantidad de bits de información, D es la cantidad de bits de paridad en la codificación LDPC y  $G = E + D$  correspondería a la longitud del codeword. Las matrices LDPC regulares se dividen en Gallager caracterizada por una estructura de bandas horizontales [27]. También, en Mackay-Neal caracterizada por una construcción aleatoria [28]. Este tipo de matrices no garantiza una independencia entre columnas necesaria por RIP en CS, por lo cual no se usarán en esta implementación [13].

Las matrices irregulares se dividen en escalera y triángulo (*staircase* y *triangle*) [29]. Son matrices que contienen dos submatrices y cada una de estas un método diferente de creación [29, 30]. La submatriz izquierda  $H_i$  de tamaño  $D \times E$  se construye de la siguiente manera:

- Usar un vector en el que conste una lista aleatoria de posibles posiciones de «1»s en las filas. Este vector se utiliza para garantizar una distribución homogénea.

- Se coloca en cada columna  $W_c$  elementos no cero, escogiendo las posiciones del vector de posibles localizaciones. Si ya no se cumple la condición de máximo  $W_r$  elementos no cero por fila se escoge otra posición aleatoria disponible.
- Se añade elementos no cero en las filas que tengan menos de dos elementos para evitar problemas en la decodificación.

La submatriz derecha hace la diferencia en la formación de una matriz LDPC irregular ya que se puede formar en escalera o de forma triangular y en cualquier caso es una matriz de dimensión  $D \times D$ . En el caso de la matriz  $S_C$ , es una matriz identidad a la que posteriormente se coloca los elementos no cero en la parte inferior diagonal. La Ecuación (10) muestra una matriz  $H_{LDPC}$  construida mediante este método.

$$(H_i|SC) = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 & : & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & : & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & : & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & : & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & : & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (10)$$

La submatriz derecha triángulo  $Tr$  se compone de una matriz identidad como base de dimensión  $D \times D$ , posteriormente se coloca una matriz triangular inferior dispersa. La Ecuación (11) muestra una matriz  $H_{LDPC}$  construida mediante este método.

$$(H_i|Tr) = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 & : & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & : & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & : & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & : & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & : & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (11)$$

## 3. Implementación de CS y LDPC para la estimación de canal

Este trabajo se implementa en equipos periféricos universales de radio por *software* (USRP, del inglés *Universal Software Radio Peripheral*). El diagrama de bloques del sistema implementando se puede apreciar en la Figura 2, donde se presenta cada etapa del sistema de comunicación.

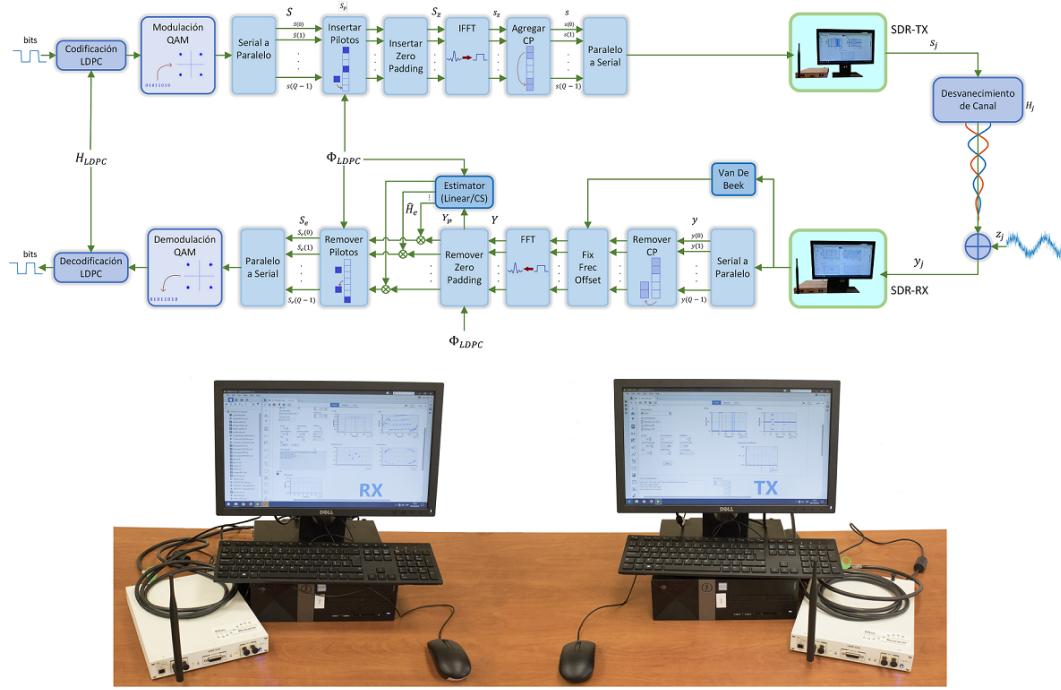


Figura 2. Modelo del sistema OFDM implementado.

Las características de los equipos usados, son los USRP Ettus X310 equivalentes al NI-USRP 2940R con las siguientes especificaciones:

- Ancho de banda de hasta 40 MHz por cada canal (2 canales).
- La imagen cargada en el FPGA permite conexión Ethernet de 1 Gbps para transmitir 25 MS/s Full Duplex.
- La arquitectura UHD usada es la disponible para «LABVIEW Communication Design Suite».
- Arquitectura de reloj flexible con frecuencia de muestreo configurable.

El ancho de banda del sistema está en función del índice IQ usado. El índice IQ determina el ancho de banda disponible [31].

Se debe indicar que con el objeto de trabajar con un sistema de comunicaciones más apegado a la realidad, se ha usado un codificador de canal LDPC considerando lo trabajado en [32, 33]. La selección de las matrices  $H_{LDPC}$  de esta implementación se estableció por validación a través del análisis de BER. La selección de la matriz  $H_{LDPC}$  aplicó simulaciones variando la dispersidad de «1»s de la matriz considerando un canal AWGN con desvanecimiento de Rayleigh. Estas matrices tienen un tamaño de  $G=40$  y  $E=20$ , debido a su eficiencia y velocidad de acuerdo con la Figura 3. Por esto, la implementación del codificador de canal usa las matrices de verificación de paridad de tipo escalera.

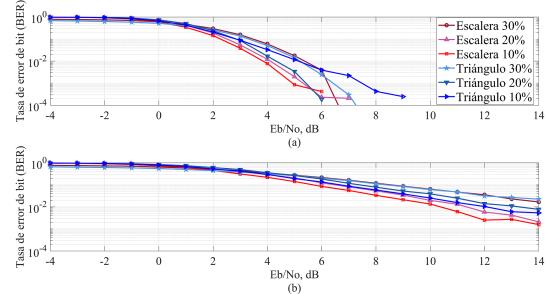


Figura 3. Análisis BER para matrices LDPC irregulares con diferentes porcentajes de dispersión de «1»s: (a) en un canal AWGN. (b) en un canal con desvanecimiento de Rayleigh.

### 3.1. Estimación de canal basado en CS

En la estimación lineal, el proceso mejora cuando la cantidad de pilotos aumenta a expensas de una reducción del ancho de banda disponible para transmitir información. No obstante, el uso de CS para estimar el canal reduce el número de subportadoras piloto en el proceso es posible ya que solo hay  $K$  subportadoras usadas como pilotos. Así, en la Ecuación (12),  $\hat{H}_p$  es la versión comprimida del canal. Entonces, el vector comprimido y corresponde a  $\hat{H}_p$  dividiendo los pilotos recibidos  $Y_p$  para los enviados  $S_p$ . Por tanto, la matriz de sensado  $\Phi$  es una matriz de tamaño  $P \times N$  en donde cada  $N_f$  elemento es distinto de 1 y corresponde a la posición de cada símbolo piloto, como se observa en la Ecuación (13) [6]. En una distribución tipo peine  $N_f$  es constante.

$$\hat{H}_p = \frac{Y_p}{S_p} \quad (12)$$

$$\Phi = \begin{bmatrix} & \underbrace{N_f} \\ 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}_{P,N} \quad (13)$$

Para aplicar CS, se consideró un diccionario  $\Psi$  a partir de la DFT para aprovechar el proceso inherente a la IDFT y DFT que se da en OFMD. Los algoritmos de reconstrucción dispersa usados son OMP y CoSaMP, usan como parámetros de entrada los datos comprimidos  $\hat{H}_p$ , la matriz  $\Theta = \Phi\Psi^H$  y el total de iteraciones. El número máximo de iteraciones debe estar en relación con el número  $K$  de elementos no cero de la señal a construir, en este caso, el número de portadoras piloto [13]. De esta manera, el canal estimado  $\hat{H}_e$ , se obtiene de la forma indicada en la Ecuación (14).

$$\hat{H}_e = 0_N$$

$$\hat{H}_e = \hat{H}_e + \Psi_{N,i}^H \times \alpha_i \quad \text{con } i = 1, 2, \dots, N \quad (14)$$

### 3.2. Estimación de canal con CS y LDPC

Como se indicó en la sección 2, la matriz de sensado  $\Phi$  debe cumplir el criterio de RIP para que pueda recuperar adecuadamente la señal dispersa, no obstante, la matriz usada en la Ecuación (13) cumple este criterio en los límites de la restricción ya que no es aleatoria. Por lo tanto, se desarrolla una matriz de posición de pilotos  $\Phi_{LDPC}$  que mejore este requisito. De forma similar a la Ecuación (13), cada elemento no cero corresponde a una posición de cada subportadora piloto en la Ecuación (15) [6].

$$\Phi_{LDPC} = \begin{bmatrix} & \underbrace{N_f} \\ 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}_{P,N} \quad (15)$$

La Ecuación (15) mantiene la distancia entre las subportadoras piloto  $N_f$  como variable y pseudoaleatoriedad debido al uso de los algoritmos de LDPC con el que se construye la matriz  $H_{LDPC}$ . Para el diseño de  $H_{LDPC}$  la cantidad de «1»s por fila es igual a 1

tomando en cuenta la cantidad de subportadoras piloto. Además, el porcentaje total de elementos no cero debe ser igual al número de pilotos  $P$ .

A partir de la obtención de esta matriz  $\Phi_{LDPC}$  el proceso continúa con la estimación del canal utilizando los algoritmos de reconstrucción dispersa OMP y CoSaMP considerando la distribución de los pilotos obtenidos en  $\Phi_{LDPC}$ .

### 3.3. Escenarios implementados

Este trabajo aplicó diferentes escenarios manteniendo un enfoque para sistemas Long Term Evolution (LTE). Para esto, se tomó en cuenta las características de LTE tales como el número de subportadoras, símbolos de referencia, subportadoras nulas y la longitud del CP indicadas en [31, 34]. Se usó una separación de 7.5 KHz entre subportadoras debido a que con el uso prolongado el rendimiento decae al usar 15 KHz. Entonces el ancho de banda implementado fue de 1.92 MHz. Es necesario aclarar que no se usó un número mayor de subportadoras como si lo permite LTE debido a que el procesamiento del código fuente se dio desde el computador y no sobre el FPGA del equipo.

El primer escenario implementado no usa el codificador de canal y tiene las siguientes características:

- IQ sample rate: 1.92 MS/s
- Frecuencia de portadora: 1.99 GHz
- Modulación: 4 QAM
- Iteraciones del algoritmo OMP: 5
- Longitud del mensaje transmitido (tren de bits): 1600
- Número de muestras en el receptor:  $3 \times$  total de datos transmitidos
- Total de subportadoras OFDM: 256
- Total de subportadoras con datos + pilotos: 150
- Ganancia del transmisor: 15.5 - 21 dB
- Sensibilidad del receptor: 0 dB

Para obtener los resultados del comportamiento de los estimadores de canal para los diferentes algoritmos de reconstrucción dispersa, OMP y CoSaMP, así como del estimador lineal, se usaron dos SDR por separado ubicados a una distancia considerable en un ambiente cerrado (indoor) en el laboratorio de pruebas como se observa en la Figura 2. La distancia se consideró como constante y se modificaron las potencias de los equipos para obtener las variaciones de la relación señal a ruido (SNR, del inglés *Signal to Noise Ratio*) y someter los diferentes estimadores de canal al análisis de BER.

Para el segundo escenario las condiciones anteriores se repiten agregando el codificador de canal LDPC usando las siguientes configuraciones:

- Iteraciones del algoritmo de propagación para la decodificación LDPC: 100
- Matriz  $H_{LDPC}$  con G=40 y E=20
- Tipo de matriz LDPC en el codificador: escalera con dispersidad de «1»s entre el 5 al 10 %.

El escenario final de análisis se implementa con distancia variable entre transmisor y receptor, también en un ambiente cerrado para evaluar un canal en condiciones más rigurosas. La distancia de separación entre los SDR consideró la longitud, distribución física del laboratorio con distancias entre 4 y 10 metros considerando una potencia constante que no sature el canal.

La variación de distancia permitió recrear canales con gran desvanecimiento, generando un aumento del efecto de multirayos para evaluar el comportamiento de los estimadores. La frecuencia de transmisión es de 2.4 Ghz, por lo que coexiste con la red de wifi del laboratorio, haciendo que el canal varíe constantemente.

En este escenario se trabajó en la presencia de estudiantes utilizando un codificador LDPC manteniendo la siguiente configuración:

- IQ sample rate: 1.92 MS/s
- Frecuencia de portadora: 2.4 GHz
- Modulación: 4 QAM
- Iteraciones del algoritmo OMP: 5
- Iteraciones del algoritmo de propagación para la decodificación LDPC: 100
- Matriz  $H_{LDPC}$  con G=40 y E=20
- Longitud del mensaje transmitido (tren de bits): 1240
- Número de muestras en el receptor:  $2 \times$  total de datos transmitidos
- Número Total de subportadoras OFDM: 256
- Número Total de subportadoras con datos + subportadoras piloto: 150
- Tipo de matrices LDPC en el codificador: escalera con dispersidad de 1 s entre el 5 al 10 %, 10 al 20 % y 20 al 30 %
- Ganancia del transmisor: 20 dB
- Sensibilidad del receptor: 20 dB

## 4. Análisis de resultados

### 4.1. Escenarios 1 y 2 - Análisis de BER

En la implementación del algoritmo OMP, el número de interacciones debe ser elegido de forma que  $K$  corresponda al número de elementos diferentes de cero de la señal que se busca recuperar. Para definir este valor, se desarrollaron múltiples pruebas buscando minimizar el error en la estimación de canal obteniendo que el valor adecuado para que este algoritmo converja en la implementación es de  $K = 5$  correspondiente al número mínimo de subportadoras usado.

La Figura 4 muestra el comportamiento de BER con 5 y 25 subportadoras piloto para los estimadores basados en OMP, CoSaMP y el estimador lineal para el primer escenario.

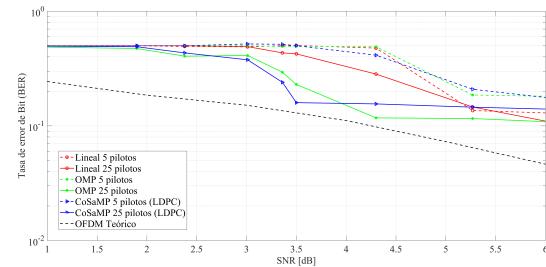
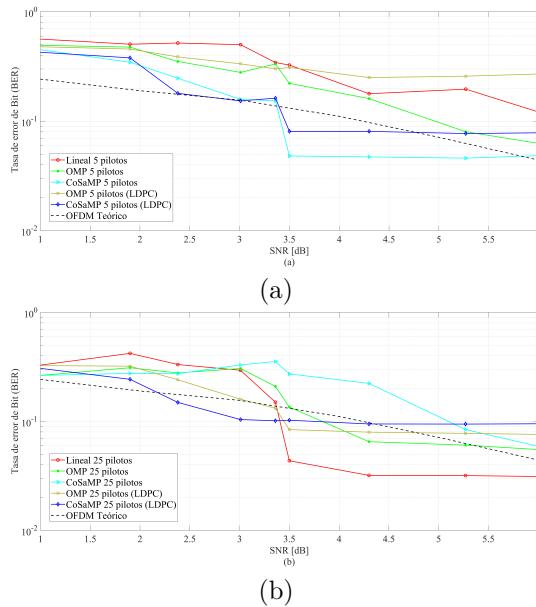


Figura 4. Análisis de BER sin codificador de canal.

En la Figura 4 se puede observar que a bajos niveles de SNR el estimador lineal presenta un rendimiento menor con respecto al uso de los estimadores basados en el paradigma de CS empleados. Con 25 subportadoras piloto, los estimadores de canal alcanzan una reducción considerable en la probabilidad de pérdida de bits. En estas pruebas los algoritmos de CoSaMP y OMP trabajaron con la matriz  $\Phi_{LDPC}$  y en ausencia de esta (distribución de subportadoras piloto de tipo peine).

La Figura 5 muestra el comportamiento de BER para 5 estimadores de canal usando las configuraciones del segundo escenario. Entonces, la Figura 5(a) de acuerdo con lo esperado por el uso de un codificador de canal, puede observarse una clara mejora respecto del caso anterior. El estimador lineal tiene los peores resultados ante un SNR bajo. Por otra parte, los estimadores CoSaMP tienen un rendimiento similar usando o no la matriz  $\Phi_{LDPC}$ . No obstante, la versión con matriz  $\Phi_{LDPC}$  presenta un mejor rendimiento. También, a medida que mejora el SNR, el estimador CoSaMP sin matriz  $\Phi_{LDPC}$  tiene una ligera mejora, pero continuando ambos con la misma tendencia en su comportamiento. Los estimadores OMP por su parte, presentan un menor rendimiento contrastados con CoSaMP, teniendo en cuenta que la versión con la matriz  $\Phi_{LDPC}$  es la que peores resultados ofrece en particular para este algoritmo.

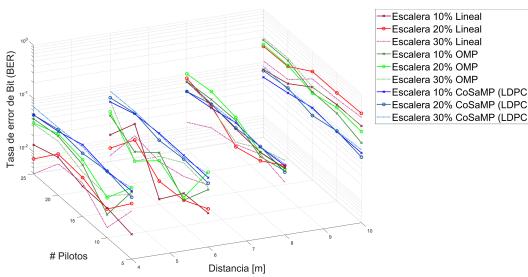
La Figura 5(b) muestra resultados del rendimiento de los estimadores frente al BER usando 25 subportadoras piloto para cada uno. Se puede observar que el estimador lineal presenta mayor probabilidad de error de bit para un bajo SNR, mostrando el peor comportamiento de entre todos los estimadores analizados. Sin embargo, cuando el SNR supera a *priori* el valor de 3.5 dB su rendimiento aumenta, obteniendo la menor probabilidad de error de bit frente a los otros estimadores.



**Figura 5.** Análisis de BER con codificador de canal LDPC: (a) BER con 5 pilotos (b) BER con 25 pilotos.

#### 4.2. Escenario 3 – Análisis para distancias variables

Las Figuras 6 y 7 corresponden al comportamiento del estimador lineal y de los estimadores basados en CS con y sin la matriz  $\Phi_{LDPC}$ .

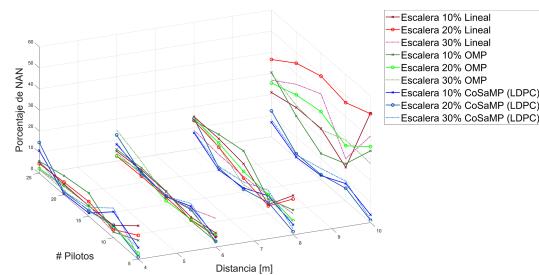


**Figura 6.** Análisis de BER para diferentes distancias.

La Figura 6 muestra que el estimador lineal tiene un mejor comportamiento a muy corta distancia, lo que corresponde a los resultados anteriores en el análisis de BER. No obstante, a medida que la distancia aumenta, este estimador de canal reduce su rendimiento

en forma drástica. El estimador CoSaMP con  $\Phi_{LDPC}$  es mejor en contraste con lo que sucede con el estimador que usa OMP. Se debe indicar que esta figura toma en cuenta los resultados ante el uso de 5 a 25 subportadoras piloto.

Debido a que el decodificador de canal usa como entrada las respuestas de los estimadores de canal, en este escenario se analizó el porcentaje de valores que de dicho codificador no pudo resolver. De esta manera, la Figura 7 muestra el porcentaje de valores nulos (NaN) en contraste con los cambios de distancia para cada estimador de canal.

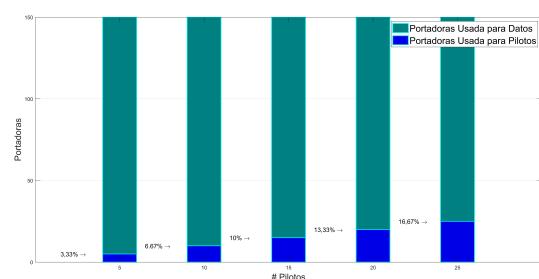


**Figura 7.** Análisis de la convergencia del decodificador de canal según la respuesta de los estimadores a diversas distancias.

La Figura 7 también corrobora los resultados anteriores resaltando la misma tendencia en los estimadores. El estimador lineal trabaja mejor a distancias cortas y los estimadores CS tienen mejor rendimiento que el lineal frente a canales ruidosos, comprobándose esto al aumentar la distancia.

#### 4.3. Análisis del ancho de banda

La Figura 8 muestra la relación que existe entre el ancho de banda usado por las subportadoras piloto y el ancho de banda para el envío de información en un símbolo OFDM. De este modo, usar 5 subportadoras piloto en lugar de 25, corresponde a una mejora de eficiencia espectral del 13.33 %. Esta mejora espectral corresponde al uso de los estimadores de canal basados en CS como se ve en la Figura 8.



**Figura 8.** Ancho de banda ocupado a diferente número de pilotos.

## 5. Conclusiones

De acuerdo con los resultados obtenidos, los estimadores CoSaMP presentan un mejor rendimiento para bajos niveles de SNR con el uso de la matriz  $\Phi_{LDPC}$ . Los estimadores OMP muestran un comportamiento similar al CoSaMP aunque con menor rendimiento. Así, con los resultados obtenidos se deduce que, un estimador lineal trabaja el problema de la estimación de canal con muy baja complejidad computacional. No obstante, cuando el canal es muy ruidoso sobre todo debido a multitrayectos demanda mayor cantidad de subportadoras piloto para tratar de mantener su rendimiento, lo que significa reducción del ancho de banda efectivo. En contraste, los estimadores que basados en CS usados en este trabajo, además de mantener baja complejidad computacional, para canales ruidosos muestran un rendimiento adecuado, permitiendo la transferencia de datos en canales ruidosos, considerando que se evaluó en un ambiente indoor.

El uso del paradigma de CS mejora mientras las propiedades de RIP y de espacio nulo puedan asegurarse, en este caso, el canal se puede considerar como disperso porque el número de subportadoras piloto es disperso.

Además, implementar una matriz de sensado diseñada a partir de la conceptualización de las matrices LDPC ha permitido mejorar el comportamiento de los estimadores basados en CS, la mejora es más clara para el estimador que usa OMP. El estimador CoSaMP con  $\Phi_{LDPC}$  muestra ser el mejor estimador frente a esta problemática de trabajo en conjunto con el codificador de canal usado. Entre los trabajos a futuro es implementar la estimación de canal y procesamientos previos en el FPGA del dispositivo para permitir que la comunicación entre el equipo y la fuente de datos pueda usar una mayor tasa de transmisión de datos.

## Referencias

- [1] J. Sterba and D. Kocur, "Pilot symbol aided channel estimation for ofdm system in frequency selective rayleigh fading channel," in *2009 19th International Conference Radioelektronika*, April 2009, pp. 77–80. [Online]. Available: <https://doi.org/10.1109/RADIOELEK.2009.5158729>
- [2] C. Tzi-Dar and T. Pei-Yun, *OFDM Baseband Receiver Design OFDM Baseband Receiver Design for Wireless Communications*, J. W. . S. A. P. Ltd, Ed., 2007. [Online]. Available: <http://doi.org/10.1002/9780470822500>
- [3] M. R. Raghavendra, S. Bhashyam, and K. Giridhar, "Exploiting hopping pilots for parametric channel estimation in ofdm systems," *IEEE Signal Processing Letters*, vol. 12, no. 11, pp. 737–740, Nov 2005.
- [4] N. N. Jimenez Castro Martinez, "Identificación ciega de canal disperso basado en algoritmos de sub-espacio," Master's thesis, 2016. [Online]. Available: <http://inaoe.repositorioinstitucional.mx/jspui/handle/1009/40>
- [5] M. Cordero Limón, "Técnicas de estimación de canal en la capa física wirelessman-ofdm de la norma ieee 802.16e," 2009. [Online]. Available: <https://bit.ly/2P9Q1co>
- [6] S. Zhang, J. Kang, Y. Song, and N. Wang, "An optimization for channel estimation based on compressed channel sensing," in *2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, Aug 2012, pp. 597–602. [Online]. Available: <https://doi.org/10.1109/SNPD.2012.128>
- [7] Y. Liao, G. Sun, X. Shen, S. Zhang, X. Yang, X. Zhang, H. Yao, and N. Zhang, "Bem-based channel estimation and interpolation methods for doubly-selective ofdm channel," in *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, Aug 2018, pp. 70–75. [Online]. Available: <https://doi.org/10.1109/SmartIoT.2018.00022>
- [8] Y. A. Al-Jawhar, K. N. Ramli, M. A. Taher, N. S. Mohd Shah, L. Audah, and M. S. Ahmed, "Zero-padding techniques in ofdm systems," *International Journal on Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 704–725, 2018. [Online]. Available: <https://bit.ly/2YAQGqc>
- [9] P. H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *IEEE Transactions on Communications*, vol. 42, no. 10, pp. 2908–2914, Oct 1994. [Online]. Available: <https://doi.org/10.1109/26.328961>
- [10] J. van de Beek, M. Sandell, and P. O. Börjesson, *ML estimation of timing and frequency offset in multicarrier systems*, div. of signal processing lulea university of technology s-971 87 lulea, sweden ed. Luleå tekniska universitet, 1996. [Online]. Available: <https://bit.ly/38qaO2W>
- [11] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006. [Online]. Available: <https://doi.org/10.1109/TIT.2006.871582>
- [12] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, July 2007. [Online]. Available: <https://doi.org/10.1109/MSP.2007.4286571>

- [13] R. Baraniuk, M. A. Davenport, M. F. Duarte, and C. Hegde, *An introduction to compressive sensing*, 2011. [Online]. Available: <https://bit.ly/38uCOCG>
- [14] Milliarde. (2016) Compressed sensing intro & tutorial w/ matlab. CODE PROJECT for those who code. [Online]. Available: <https://bit.ly/2YC5Ewa>
- [15] J. Inga-Ortega, E. Inga-Ortega, C. Gómez, and R. Hincapié, “Electrical load curve reconstruction required for demand response using compressed sensing techniques,” in *2017 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*, Sep. 2017, pp. 1–6. [Online]. Available: <https://doi.org/10.1109/ISGT-LA.2017.8126731>
- [16] M. Pinos and J. Inga, “Predicción de consumo eléctrico en la UPS de Cuenca usando P1P y censado comprimido,” 2018.
- [17] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008. [Online]. Available: <https://doi.org/10.1109/MSP.2007.914731>
- [18] E. J. Candés, “The restricted isometry property and its implications for compressed sensing,” *Comptes Rendus Mathématique*, vol. 346, no. 9, pp. 589–592, 2008. [Online]. Available: <https://doi.org/10.1016/j.crma.2008.03.014>
- [19] E.-G. Astaiza-Hoyos and H. F. Pablo Emilio Bermúdez-Orozco, “Compressive sensing: A methodological approach to an efficient signal processing,” *Dyna*, 2015. [Online]. Available: <https://bit.ly/2qLx8TI>
- [20] H. Yuan, H. Song, X. Sun, K. Guo, and Z. Ju, “Compressive sensing measurement matrix construction based on improved size compatible array ldpc code,” *IET Image Processing*, vol. 9, no. 11, pp. 993–1001, 2015. [Online]. Available: <https://doi.org/10.1049/iet-ipr.2015.0117>
- [21] A. G. Dimakis, R. Smarandache, and P. O. Vontobel, “Ldpc codes for compressed sensing,” *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3093–3114, May 2012. [Online]. Available: <https://doi.org/10.1109/TIT.2011.2181819>
- [22] S. Pawar and K. Ramchandran, “A hybrid dft-ldpc framework for fast, efficient and robust compressive sensing,” in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2012, pp. 1943–1950. [Online]. Available: <https://doi.org/10.1109/Allerton.2012.6483460>
- [23] H. Wang, W. Du, and Y. Bai, “Compressed sensing based channel estimation for OFDM transmission under 3 GPP channels,” *International Journal of Future Generation Communication and Networking*, vol. 9, no. 4, pp. 85–94, 2016. [Online]. Available: <http://dx.doi.org/10.14257/ijfgcn.2016.9.4.08>
- [24] D. Needell and J. A. Tropp, “Cosamp: Iterative signal recovery from incomplete and inaccurate samples,” *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009. [Online]. Available: <https://doi.org/10.1016/j.acha.2008.07.002>
- [25] B. L. Sturm. (2011) Algorithm power hour: Compressive sampling matching pursuit (cosamp). [Online]. Available: <https://bit.ly/2RS0DOX>
- [26] I. D. F. Lava, “Implementación y evaluación de la codificación ldpc para la transmisión de ficheros en entornos unidireccionales,” in *Computer Science*, 2013. [Online]. Available: <https://bit.ly/2YCauJQ>
- [27] R. Gallager, “Low-density parity-check codes,” *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, January 1962. [Online]. Available: <https://doi.org/10.1109/TIT.1962.1057683>
- [28] D. J. C. Mackay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003. [Online]. Available: <https://bit.ly/2E5yfQX>
- [29] G. G. Andrade Salinas, “Implementación de la codificación y decodificación del código LDPC (low density parity check) en MATLAB,” 2017. [Online]. Available: <https://bit.ly/34ahKxG>
- [30] V. Roca, C. Neumann, and D. Furodet. (2008) Low density parity check (ldpc) staircase and triangle forward error correction (fec) schemes. [Online]. Available: <https://bit.ly/2RH8PBq>
- [31] E. Seidel, “Overview lte phy: Part 1 - principles and numerology etc,” *NOMOR Research*, 2007. [Online]. Available: <https://bit.ly/35cxH7R>
- [32] R. Prieto, A. Abril, and A. Ortega, “Experimental alamouti-stbc using ldpc codes for mimo channels over sdr systems,” in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, April 2017, pp. 1–5. [Online]. Available: <https://doi.org/10.1109/CCECE.2017.7946842>
- [33] B. Peñafiel and A. Ortega, “A joint multilevel modulator and robust ldpc codes processing over optical systems by providing solutions for holistic

- 5g backhaul,” in *2017 IEEE 13th Malaysia International Conference on Communications (MICC)*, Nov 2017, pp. 197–202. [Online]. Available: <https://doi.org/10.1109/MICC.2017.8311758>
- [34] J. Zyren, “Overview of the 3gpp long term evolution physical layer,” NXP Semiconductors, Tech. Rep., 2007. [Online]. Available: <https://bit.ly/35gg8E0>