

An Application of 8-APSK Modulation for the Uplink using SVD-SCMA

S. Vidal-Beltrán, J. L. López-Bonilla, F. Martínez-Piñón

Abstract— In recent years SCMA has been proposed as a strong candidate to perform as access technique for fifth-generation mobile communications systems; usually for the uplink, bit pairs are mapped directly to code words and MPA is used in the receiver. In this work, we proposed to encode three bits to increase the capacity of data transmission in the uplink using singular value decomposition of the initial mother constellation to generate the codewords. In the receiver, the detection and decoding of data is more complex than its predecessors with two bits, since the symbols in the IQ constellation are closer to each other, so a receiver based on neural networks is proposed to perform these tasks. The experimental performance of this proposal was evaluated by implementing a software defined radio platform. The results show that, even under unfavorable SNR conditions, it is possible to use 8-APSK as a modulation technique with low error rate in the detection and decoding of SCMA signals.

Keywords—5G, 8APSK, Adam, Neural networks, SCMA, Supervised Learning.

I. INTRODUCCIÓN.

SCMA(Sparse Code Multiple Access) es una técnica de Acceso múltiple que está bajo proceso de evaluación para ser implementada en los sistemas de comunicaciones móviles de quinta generación (5G) [1]-[3]. SCMA codifica directamente los datos de usuario en palabras de código (codewords) multidimensionales usando libros de código (codebooks) de baja densidad para reducir la complejidad en la detección y decodificación de datos. Con este método a cada usuario se le asigna un libro de código y los datos dentro de este son empleados para mapear el flujo de bits directamente a un vector de baja densidad (sparse vector) llamado palabra de código [2], [4], [5]. Libros de código multidimensionales son usados para complementar la modulación QAM (Quadrature Amplitude Modulation) [6]-[8] y el ensanchamiento depende de este factor. Cada vector contiene el valor cero en las mismas dos dimensiones de todos sus elementos y la posición de los ceros está determinada aleatoriamente para prevenir colisiones entre los usuarios. De acuerdo con [9]-[12], para detectar y decodificar las palabras de código SCMA se usan principalmente dos técnicas, por un lado, MPA (Message Passing Algorithm) el cual se basa en un complejo algoritmo de probabilidad conjunta y, por otro lado, se emplea el método de mínima distancia Euclidiana, que,

S. Vidal-Beltrán, Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica – Zacatenco, CDMX, México (e-mail: svidalb@ipn.mx)

F. Martínez-Piñón, Instituto Politécnico Nacional, Centro de Investigación e Innovación Tecnológica- Azcapotzalco, CDMX, México

J. L. López-Bonilla, Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica – Zacatenco, CDMX, México

aunque es un algoritmo más sencillo de implementar, requiere de más tiempo de máquina para hacer los cálculos.

En [2], [4], [5], [6], [7] se describen diferentes métodos para construir los libros de código SCMA, sin embargo, en todos ellos siempre se codifican dos bits en 4 palabras de código ($M = 4$); en este trabajo se propone la codificación de 3 bits utilizando descomposición en valores singulares (SVD - Singular Value Decomposition) [13] de la constelación madre; los resultados son comparados con el método tradicional de rotación de fase [2], [4], [5]. La codificación de tres bits, no ha sido abordada aún, ya que la complejidad para detectar y decodificar las señales aumenta considerablemente.

Este trabajo está organizado de la siguiente forma, en la sección II, se presenta la arquitectura general del escenario de pruebas usado, así como el software y hardware considerados. En la sección III, se describe el diseño y construcción de palabras de código SVD-SCMA empleando modulación 8-APSK para transmitir 3 bits simultáneamente; la sección IV está dedicada a exhibir las características de la detección y decodificación en el receptor y se describen los conceptos fundamentales de los algoritmos de optimización basado en gradiente descendente. La sección V, muestra los resultados de este trabajo, primeramente, se describen los resultados de la fase de entrenamiento de la red neuronal y posteriormente los resultados experimentales del escenario de pruebas usado; finalmente en la sección VI, se describen las conclusiones y se discute acerca de las consideraciones finales de este trabajo, así como posibles extensiones del mismo.

II. ARQUITECTURA GENERAL DEL ESCENARIO DE PRUEBAS.

La arquitectura general usada para el escenario de pruebas se presenta en la Fig. 1, donde se observa en la parte izquierda que los datos de los usuarios se dividen en tripletas de bits, los cuales son codificados mediante SCMA, para posteriormente ser asignados a los recursos de radio y transmitidos por canal inalámbrico hacia la estación base, la cual detecta y decodifica las señales SCMA mediante aprendizaje supervisado.

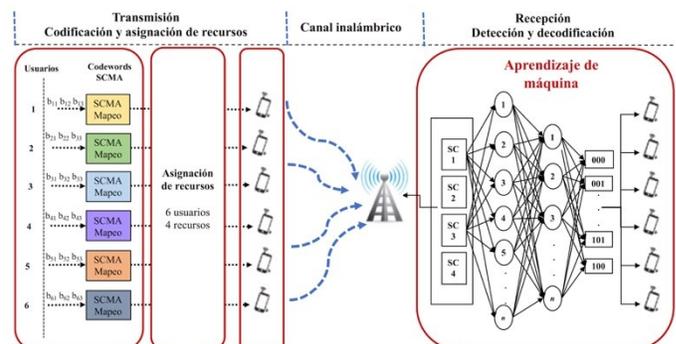


Fig. 1. Arquitectura general de transmisión-recepción.

El escenario de pruebas mostrado en la Fig. 1, fue implementado usando un radio definido por software (SDR - Software Defined Radio). Un SDR es considerado un sistema de radiocomunicación donde la mayor parte de los componentes (como mezcladores, filtros, amplificadores, moduladores/demoduladores y detectores entre otros e incluso sistemas completos tales como transmisores, receptores, transceptores, osciloscopios, analizadores de espectros o analizadores vectoriales de redes) se implementan en software en lugar de en hardware, normalmente utilizando una computadora (PC). Un SDR está compuesto de software y hardware, logrando modificar su configuración dinámicamente a nivel de software para habilitar comunicaciones entre una amplia variedad de normas de comunicaciones y protocolos, proporcionando una solución eficiente y de bajo costo donde, los servicios pueden ser mejorados a través de actualizaciones de software y reconfiguraciones de hardware [14].

En este trabajo se utilizó HackRF One como SDR capaz de transmitir o recibir señales de radio desde 1 MHz hasta 6 GHz, el cual fue diseñado para facilitar el desarrollo y prueba de tecnologías de comunicación por radio, HackRF One es una plataforma de hardware libre que puede ser usada como un periférico vía USB, o programada para operar de forma autónoma. Las características principales del SDR se listan a continuación:

- Operación en frecuencia desde 1 MHz hasta 6 GHz.
- Transceptor con capacidad de operación half-duplex.
- 20 millones de muestras por segundo.
- Compatible con los principales programas para SDR (SDR#, GNU Radio).
- Configuración vía software de los amplificadores de ganancia, con 3 etapas dedicadas para recepción y 2 etapas para transmisión.
- Configuración vía software los filtros de señal en banda base, con un máximo de ancho de banda de señal de 28 MHz, y con una caída de 3 dB hasta 30 MHz.
- Potencia suministrada al puerto de la antena, de hasta 50 mA a 3.3 V.
- Conector de antena SMA hembra.
- Interfaz USB 2.0.

GNU Radio es un Entorno de Desarrollo Integrado (IDE. - Integrated Development Environment) de código abierto creado con fines educativos y de investigación científica, para la simulación y/o el desarrollo de aplicaciones de procesamiento de señales. Las aplicaciones, programas o nuevos bloques de procesamiento en este IDE se desarrollan en Python y C++. Cuenta con un entorno gráfico denominado GNU Radio Companion similar a Simulink en Matlab [14] donde, mediante bloques gráficos de procesamiento, que incluyen filtros, ecualizadores, moduladores, demoduladores, entre otros. También permite la creación de nuevos bloques escribiendo el código en los lenguajes de programación mencionados. Para ejecutar GNU Radio se utilizó Ubuntu 18.04 como sistema operativo. La tabla 1, muestra el resumen de los elementos usados en la puesta en operación del escenario de pruebas usado en este trabajo.

TABLA I
ELEMENTOS USADOS EN EL ESCENARIO DE PRUEBAS

Elemento	Descripción
Laptop 1	Ubuntu desktop 18.04, 4GB de memoria RAM, GNU radio 3.7.
Laptop 2	Ubuntu desktop 18.04, 4GB de memoria RAM, GNU radio 3.7.
HackRF One 1	1 MHz hasta 6 GHz
HackRF One 2	1 MHz hasta 6 GHz
Python, Numpy y SciPy	Lenguaje de programación en su versión 2.7

La Fig. 2, muestra los SDR usados como transmisor y receptor:



Fig. 2. Escenario experimental con SDR.

Como se muestra en la Fig. 2 cada HackRF One se conectó vía USB a una laptop, una de las cuales hizo la función del dispositivo móvil y la otra la función de la estación base. Dado que la programación se realizó en Python, se seleccionó GNU Radio como plataforma para programar el SDR; de esta forma se facilita la integración de las pruebas, las antenas fueron colocadas con una separación de 30 centímetros entre ellas, las variaciones en la SNR se realizaron por software.

III. TRANSMISIÓN DE PALABRAS DE CÓDIGO SVD-SCMA USANDO MODULACIÓN 8-APSK.

De acuerdo con la Fig. 1, para este trabajo los datos de usuario se codificaron en 3 bits empleando una constelación 8-APSK [15] (Amplitude and phase-shift keying), la cual se define por (1):

$$X = \begin{cases} r_1 e^{i(\varphi_1 + \frac{2\pi}{n_1}k)} & | k = 0, 1, \dots, n_1 - 1 \\ r_2 e^{i(\varphi_2 + \frac{2\pi}{n_2}k)} & | k = 0, 1, \dots, n_2 - 1 \\ \vdots \\ r_N e^{i(\varphi_N + \frac{2\pi}{n_N}k)} & | k = 0, 1, \dots, n_N - 1 \end{cases} \quad (1)$$

donde N es el número de anillos concéntricos ($N = 2$), n el número de puntos por anillo ($n_1 = n_2 = 4$), r es el radio del círculo, ($r_1 = 1.41, r_2 = 4.24$), y φ es el ángulo de desplazamiento ($\varphi_1 = \varphi_2 = \frac{\pi}{4} = 45^\circ$), lo cual produce una constelación de 8 símbolos ($M = 8$). Una vez definida la constelación inicial, se procede a crear los libros de código acorde a [2], [4], [5], el primer vector (S_1) de la constelación madre (M_c) está dado por (2):

$$\begin{aligned} S_{11} &= 1 + 1i, S_{12} = -1 + 1i, S_{13} = -1 - 1i, S_{14} = 1 - 1i \\ S_{15} &= 3 + 3i, S_{16} = -3 + 3i, S_{17} = -3 - 3i, S_{18} = 3 - 3i \end{aligned} \quad (2)$$

La segunda dimensión S_2 es el resultado de multiplicar $U_N S_1$, donde $U_N = \text{diag}(1e^{i\theta_{l-1}}) \in \mathbb{C}^{N \times M}$ y $\theta_{l-1} = \frac{(l-1)\pi}{MN}$, $l = 1, \dots, N$, por lo que M_C está dada por (3):

$$M_C = (S_1, S_2)^T = \begin{bmatrix} S_{11} S_{12} \dots S_{18} \\ S_{21} S_{22} \dots S_{28} \end{bmatrix} \quad (3)$$

Por otro lado, la descomposición en valores singulares ha sido usada en procesamiento de señales, en procesamiento de imágenes, y en el campo de las comunicaciones móviles ha sido empleada para la pre-codificación de señales [16] en sistemas MIMO-OFDM (Multiple-input Multiple-output - Orthogonal Frequency Division Multiplexing) [17] y en la estimación de señales [17]. En términos generales, la descomposición en valores singulares SVD es la factorización de una matriz $n \times m$ en tres factores [13], [18], como se muestra en (4):

$$A_{n \times m} = U \Sigma V^* \quad (4)$$

Considerando una matriz rectangular $A_{n \times m}$ con $n \neq m$, U es una matriz unitaria $n \times p$, V^* es una matriz unitaria de dimensiones $m \times p$, Σ es una matriz diagonal de $p \times p$ con valores no-negativos ordenados de forma descendente, y p es el rango de la matriz $A_{n \times m}$. Las columnas de U son conocidas como los vectores singulares izquierdos, mientras que las columnas de V^* se llaman vectores singulares derechos. Tanto U como V^* son matrices unitarias, y las columnas de cada matriz forman un conjunto ortonormal de eigenvectores.

A la matriz definida en (3) se le aplica la descomposición en valores singulares (SVD), tal como se aprecia en (5):

$$M_{SVD} = SVD [M_C] = U_{n \times p} \Sigma_{p \times p} V_{m \times p}^* \quad (5)$$

Las matrices resultantes U y V^* son ortonormales lo cual resulta en una distancia Euclidiana óptima entre los puntos resultantes, lo cual es fundamental para conformar la constelación final de SCMA. Como se mencionó anteriormente, V^* es una matriz de 2×8 similar a M_C , con la diferencia que V^* es un conjunto de eigenvectores ortonormales. A diferencia de lo expuesto en [2], [4], [5], en este trabajo los libros de código para todos los usuarios se obtienen a partir de V^* , en lugar de M_C . Más detalles respecto a la descomposición de valores singulares puede ser consultada en [13], [19]. Para este trabajo se consideran 4 subportadoras de radio ($K = 4$), la cantidad de usuarios en este sistema está determinada por (6):

$$J = C_N^K \quad (6)$$

Por lo que se producen seis diferentes libros de código $J = 6$. SCMA a diferencia de OFDMA permite sobrecarga, con lo cual más de un usuario es transportado en el mismo recurso de radio, la cantidad de usuarios concurrentes (d_f) se determina por (7):

$$d_f = \left(\frac{K-1}{N-1} \right) = \frac{JN}{K} \quad (7)$$

y el factor de sobrecarga λ , está dado por (8):

$$\lambda = \frac{J}{K} = \frac{d_f}{N} \quad (8)$$

Los ángulos de rotación para los distintos usuarios se obtienen a partir de d_f & e_u , los cuales representan el número de usuarios que se transportan por un recurso de radio (en este caso 3), por lo que el ángulo de rotación φ_u para los distintos usuarios se define por (9):

$$\varphi_u = (u-1) \frac{2\pi}{M d_f} + e_u \frac{2\pi}{M}, \forall u = 1, \dots, d_f \quad (9)$$

El valor de φ_u se asigna a las posiciones no-cero del factor de gráfica [20]- [21], siguiendo la regla de Latin Squares. El factor de gráfica define cuáles portadoras (filas) son empleadas por cada usuario (columnas). Considerando $M = 8$ & $N = 2$, el factor de gráfica está dado por (10):

$$F = \begin{bmatrix} \varphi_1 & \varphi_2 & 0 & \varphi_3 & 0 & 0 \\ 0 & \varphi_3 & \varphi_1 & 0 & 0 & \varphi_2 \\ \varphi_2 & 0 & \varphi_3 & 0 & \varphi_1 & 0 \\ 0 & 0 & 0 & \varphi_1 & \varphi_2 & \varphi_3 \end{bmatrix} \quad (10)$$

De (10) se observa que los datos del usuario 1 (columna 1) se transportan por las subportadoras 1 y 3 (filas 1 y 3). Considerando (10) y haciendo que f_j sea la j -ésima columna de F sin elementos cero. Se define el operador Δ_j para el j -ésimo usuario tal como se indica en (11) y (12):

$$\Delta_j = \text{diag}(f_j) = f_j, |j = 1, 2, \dots, 6. \quad (11)$$

$$\begin{aligned} \Delta_1 &= \begin{bmatrix} \varphi_1 & 0 \\ 0 & \varphi_2 \end{bmatrix} & \Delta_2 &= \begin{bmatrix} \varphi_2 & 0 \\ 0 & \varphi_3 \end{bmatrix} & \Delta_3 &= \begin{bmatrix} \varphi_1 & 0 \\ 0 & \varphi_3 \end{bmatrix} \\ \Delta_4 &= \begin{bmatrix} \varphi_3 & 0 \\ 0 & \varphi_1 \end{bmatrix} & \Delta_5 &= \begin{bmatrix} \varphi_1 & 0 \\ 0 & \varphi_2 \end{bmatrix} & \Delta_6 &= \begin{bmatrix} \varphi_2 & 0 \\ 0 & \varphi_3 \end{bmatrix} \end{aligned} \quad (12)$$

De forma similar, con base en (10), la matriz de dispersión para cada usuario V_j la cual contiene $K - N$ filas con valor cero, está dada por (13):

$$\begin{aligned} V_1 &= \begin{bmatrix} 10 \\ 00 \\ 01 \\ 00 \end{bmatrix} & V_2 &= \begin{bmatrix} 10 \\ 01 \\ 00 \\ 00 \end{bmatrix} & V_3 &= \begin{bmatrix} 00 \\ 10 \\ 01 \\ 00 \end{bmatrix} \\ V_4 &= \begin{bmatrix} 10 \\ 00 \\ 00 \\ 01 \end{bmatrix} & V_5 &= \begin{bmatrix} 00 \\ 00 \\ 10 \\ 01 \end{bmatrix} & V_6 &= \begin{bmatrix} 00 \\ 10 \\ 00 \\ 01 \end{bmatrix} \end{aligned} \quad (13)$$

Basado en (10), (12) y (13), los libros de código para los diferentes usuarios se obtienen a partir de (14):

$$X_j = V_j \Delta_j V^*, |j = 1, 2, \dots, J \quad (14)$$

donde V_j es la matriz de dispersión, Δ_j es el operador de rotación y V^* es el resultado de aplicar SVD a la matriz madre

M_c . X_j es una matriz 4×8 que representa los 6 libros de código, donde las líneas definen las subportadoras utilizadas y las columnas las 8 palabras posibles; 2 filas son cero de acuerdo con V_j . Considerando (10), la SC1 (SubCarrier 1) transporta datos de los usuarios 1, 2 y 4; cada palabra de tres bits se representa hasta por 64 posibles combinaciones, ya que cada palabra de tres bits del usuario 1 debe ser sumado con los 8 posibles valores del usuario 2 y los 8 posibles valores del usuario 4, resultando que las 8 palabras de 3 bits pueden ser representados hasta por 512 combinaciones. La Fig. 3, muestra los libros de código para el Usuario 1 y 2 usando SVD, mientras que la Fig. 4 representa los libros de código empleando el método de rotación de fase.

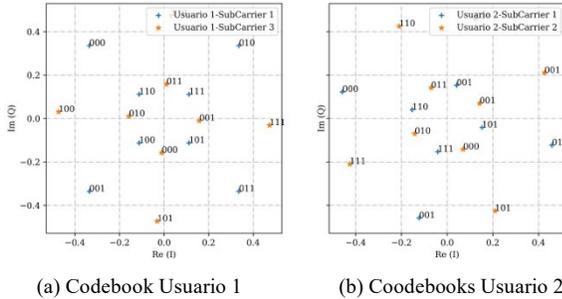


Fig. 3. Codebooks para el Usuario 1 y 2 usando SVD.

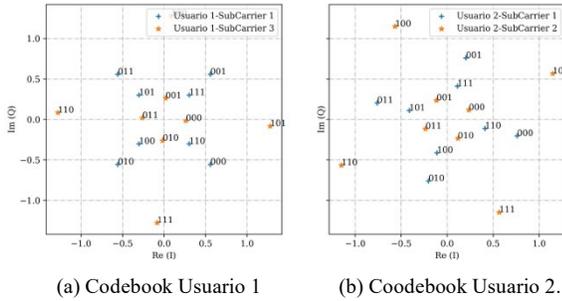


Fig. 4. Codebooks para el Usuario 1 y 2 usando rotación de fase.

Los trabajos mostrados en [2], [4], [5], [6], [7], [8] codifican 2 bits de usuario en palabras de código SCMA, y la distancia entre puntos adyacentes en la constelación es fundamental para la correcta detección y decodificación de las palabras de código. La distancia mínima entre puntos empleando SCMA, LDS, MD-SCMA, SVD-SCMA [2], [4], [5], aparece en la Tabla 2:

TABLA II
DISTANCIA MÍNIMA
ENTRE PUNTOS

	d_{min}
SCMA	2.44
LDS	$\sqrt{2}$
MD-SCMA	2
SVD-SCMA	2.99

En este trabajo, a diferencia de sus predecesores, cada palabra de código SCMA representa 3 bits, produciendo una constelación IQ más compleja, la Fig. 5 muestra la separación entre puntos de la constelación usando SVD y rotación de fase.

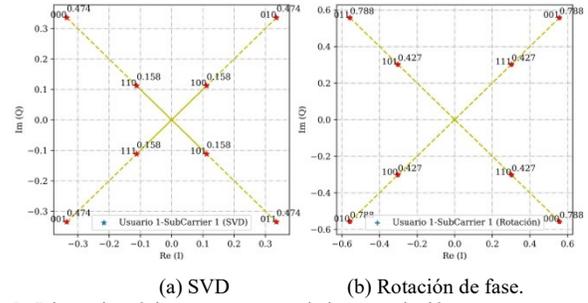


Fig. 5. Distancia mínima entre puntos de la constelación.

De acuerdo con la figura 5a, la distancia mínima entre puntos adyacentes de la constelación SVD es $\frac{0.474}{0.158} = 3$, mientras que de la Fig. 5b se observa que la distancia mínima en la constelación por rotación de fase es $\frac{0.788}{0.427} = 1.84$. Como se mencionó anteriormente al aplicar SVD a la constelación madre, las matrices resultantes son ortogonales con vectores ortonormales, lo cual produce constelaciones con mejor distancia entre puntos adyacentes. El valor singular de la matriz M_c afecta directamente la distancia entre puntos adyacentes de la constelación. En este caso el valor singular es 3, por lo que la distancia entre puntos adyacentes también es 3. La relación entre puntos adyacentes usando SVD-SCMA permanece constante tanto si se codifican dos bits como tres bits en cada palabra de código (ver Tabla 2), mientras que si se usa rotación de fase la relación disminuye de 2 a 1.84. lo cual haría que se presenten más errores en la decodificación de datos. En este trabajo, los parámetros principales para la construcción de los libros y palabras de código SCMA son:

- Cantidad de palabras: 8 ($M = 8$: 000, 001, 011, 010, 110, 111, 101, 100)
- Número de usuarios: 6 ($J = 6$)
- Cantidad de Subportadoras disponibles: 4 ($K = 4$).
- Factor de sobrecarga: 1.5 ($\lambda = 1.5$)
- Usuarios por subportadora: 3 ($d_f = 3$)
- Dimensiones de código: 2 ($N = 2$)

Los recursos de radio (subportadoras), se asignan estáticamente acorde al factor de gráfica dado en (7). Una vez que las palabras de código son seleccionadas, se envían a la antena para la transmisión inalámbrica. Ruido aleatorio se añade a la señal, la magnitud del ruido depende de la SNR (Signal-to-Noise Ratio) [22]. Para el enlace ascendente los efectos de un canal afectado por desvanecimiento Rayleigh se simula con SNR desde -20 dB hasta +18 dB.

IV. RECEPCIÓN Y DECODIFICACIÓN DE PALABRAS DE CÓDIGO SCMA.

En el receptor, la detección y decodificación de las palabras de código SCMA dependen en mayor parte del diseño de los libros de código y del ruido del canal. En [2], [4], [5], [6], [7], [8] se usa MPA y mínima distancia Euclidiana para la detección y decodificación. MPA emplea el algoritmo *sum-product* para calcular la probabilidad marginal de diferentes palabras de código. Se utilizan métodos de probabilidad

Bayesiana para calcular la probabilidad de la palabra de código dada una probabilidad inicial dentro del libro de código [9], [10], [11]. Dado que no se tiene conocimiento previo de los datos a transmitir, se asume que cada palabra de código seleccionada tiene la misma probabilidad. Es posible establecer la probabilidad inicial de un usuario escogiendo una palabra de código igual a 1 dividido por la cardinalidad M del conjunto de palabras de código en el libro de código de cada usuario. Para este el caso donde $M = 4$, todas las palabras de código tienen una probabilidad inicial de $1/4$.

Por otro lado, en el método de mínima distancia Euclidiana, el transmisor y el receptor acuerdan la misma constelación de referencia para modular y demodular la información del usuario [12]. El primer paso en la detección IQ es calcular la distancia Euclidiana entre dos vectores dados (el arreglo de referencia y los símbolos recibidos con ruido). Para cada dato recibido se calcula la distancia mínima a cada punto de la constelación de referencia, con lo cual se decodifican los datos. Este proceso, aunque es más sencillo de implementar que MPA, resulta en una gran cantidad de tiempo de máquina, dado que cada dato recibido es comparado con 512 valores posibles de la constelación de referencia.

Para este trabajo, el escenario de pruebas se ejecutó sobre Python, utilizando Numpy y SciPy como bibliotecas de código abierto para programar los distintos algoritmos. El aprendizaje de máquina ha sido ampliamente usado en ingeniería [23]-[26] y en [27] se demuestra la aplicabilidad del aprendizaje supervisado basado en redes neuronales para la detección y decodificación de señales SCMA afectado por un canal AWGN, por lo que en éste trabajo, se implementa un receptor de este tipo.

La Fig. 6, muestra la arquitectura general del receptor propuesto:

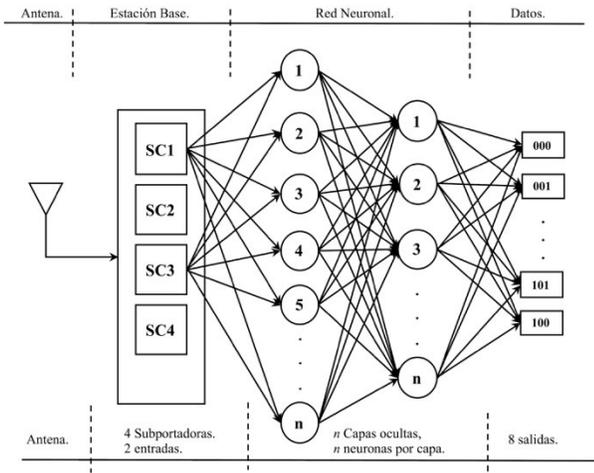


Fig. 6. Diagrama general de la red neuronal.

Considerando (7), para decodificar los datos del usuario 1, la estación base conoce el libro de código correspondiente y busca la información en las subportadoras 1 y 3 las cuales son la entrada a la red neuronal, la cual está compuesta por n número de capas densas, cada una con n número de neuronas, 8 salidas representan los datos a decodificar (000, 001, 011,

010, 110, 111, 101, 100). A continuación se describen los parámetros usados, para la implementación del receptor basado en redes neuronales.

A. Configuración de la Red Neuronal.

Como se muestra en [27], para detectar y decodificar las palabras de código SCMA solo requiere tres capas ocultas con 64, 32 y 26 neuronas, respectivamente. En contraste, dado que los puntos de la constelación 8APSK están más cercanos, se requiere de más capas densas y más neuronas por capa. En este trabajo se utilizaron 6 capas densas con 512, 512, 256, 128, 64 y 32 neuronas, respectivamente, para producir 8 salidas, correspondientes a los 3 bits a codificar.

B. Algoritmos de Optimización Basados en Gradiente Descendente.

Para problemas de aprendizaje de máquina y aprendizaje profundo se ha demostrado la efectividad de usar algoritmos de optimización basados en Gradiente Descendente (GD). GD es un método que minimiza o maximiza una función objetivo $J(\theta)$, la cual es una función matemática que depende de una serie de parámetros de aprendizaje que son usados para calcular los valores objetivo (Y), dado un conjunto de datos de entrada (X). Las redes neuronales basan su operación en una actualización constante de los parámetros de aprendizaje (weights and bias), lo cual produce una minimización de la función de pérdida y aumenta la precisión de las predicciones en datos nuevos. Existen diferentes variantes de GD los siguientes tres son empleados ampliamente:

- *Batch Gradient Descent (BGD)*.

Batch gradient Descent calcula el gradiente de la función objetivo (3) con respecto a θ , para el conjunto total de datos de entrenamiento.

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta). \quad (15)$$

Esta variante de GD calcula el gradiente para todo el conjunto de datos de entrenamiento para obtener una actualización, lo cual lo hace un algoritmo que consume más tiempo de cómputo, por lo que no es útil para aprendizaje en tiempo real.

- *Stochastic Gradient Descent (SGD)*.

Para cada dato de entrenamiento $x^{(i)}$ con una salida $y^{(i)}$, se obtiene una actualización de los parámetros de aprendizaje, resultando en (16):

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)}) \quad (16)$$

BGD para producir una actualización en los parámetros de aprendizaje debe hacer cálculos sobre grandes volúmenes de datos de entrenamiento aun cuando no haya cambios significativos en los datos, SGD hace actualizaciones continuas, lo cual lo hace útil para aprendizaje en tiempo real.

- *Adam*

Adaptive Moment Estimation (Adam) combina las características de sus antecesores [28]. Este método calcula

tasas de aprendizaje individuales adaptativas para el primer momento (mean) y el segundo momento (varianza) del gradiente (m_t, v_t), como se expresa en (17) y (18):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (17)$$

$$v_t = \beta_2 m_{t-1} + (1 - \beta_2) g_t^2 \quad (18)$$

donde β_1, β_2 (tasa de decaimiento exponencial) son cercanos a 1.

La regla de actualización para Adam, está dada por (19):

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \quad (19)$$

Considerando que:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (20)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (21)$$

donde $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$.

En 2015, D. P. Kingma y J. Lei-Ba, fueron los primeros en proponer Adam como algoritmo de optimización [28], obteniendo que el método resuelve eficientemente problemas de aprendizaje profundo; en comparación con otros métodos estocásticos Adam presenta mejor desempeño cuando se usa en redes neuronales de múltiples capas, asimismo, presenta un buen desempeño para un amplio conjunto de problemas de optimización no-convexos. En [27] Adam presentan un mejor desempeño que SGD para la detección y decodificación de palabras de código SCMA razón por la cual se eligió Adam como algoritmo de optimización para realizar este trabajo, con los siguientes parámetros de configuración: tamaño de batch de 50, intervalo de validación de 0.08, $\beta_1 = 0.9, \beta_2 = 0.999$ y una tasa de aprendizaje $l_r = 0.002$.

C. Datos de Entrenamiento de la Red Neuronal.

La detección y decodificación de palabras de código SCMA, se aborda como un problema de clasificación, por lo que los datos de entrenamiento son fundamentales para la correcta decodificación. El conjunto de datos de entrenamiento se generó usando (22):

$$X_t = M^{d_f} * \text{Rango de dB} \quad (22)$$

donde X_t , representa el conjunto inicial de datos de entrenamiento, considerando $M = 8, d_f = 3$, la red neuronal se entrenó en un rango de -20 dB a +18 dB, por lo que el conjunto de datos de entrenamiento es 19,534 valores. Un *epoch* es una iteración completa del algoritmo de optimización sobre el total del conjunto de datos de entrenamiento. Si un algoritmo de optimización logra valores altos de precisión en un número pequeño de *epochs*, indica que tiene una más rápida tasa de aprendizaje que aquellos que requieren un mayor número de *epochs*. En este trabajo se consideraron 90 iteraciones para la etapa de aprendizaje de la red neuronal.

D. Predicciones en Datos Nuevos.

Una vez que la red neuronal ha sido entrenada, para probar el funcionamiento y rendimiento, se procedió a la fase experimental usando radios definidos por software, generando 1 millón de tripletas de bits aleatorios, afectados por diferentes SNR (desde -20 dB hasta +18 dB), y sobre esos datos el sistema realiza la detección y decodificación de palabras de código SVD-SCMA.

E. Parámetros de Evaluación.

Primeramente, se presentan los resultados del proceso de aprendizaje de la red neuronal tanto para las palabras de código generadas vía SVD, como para las generadas por el método de rotación de fase; por lo que se analizan los siguientes parámetros:

i. Precisión en el aprendizaje sobre datos de entrenamiento.

Valores cercanos a 100% indican que la red neuronal aprende en su totalidad derivado de los datos con los cuales fue entrenada.

ii. Función de pérdida sobre datos de entrenamiento.

El grado de aprendizaje de una red neuronal está dado por la función de pérdida. Es un método mediante el cual se evalúa qué tan bien un algoritmo modela los datos de entrenamiento. Si las predicciones se alejan mucho de los resultados, la función de pérdida tiene valores grandes, un valor cercano a cero indica un mejor desempeño de las predicciones.

En segundo término, se muestran los resultados experimentales del desempeño de la red neuronal haciendo predicciones en datos nuevos, considerando los siguientes parámetros:

iii. Precisión del algoritmo de optimización.

Las redes neuronales y aprendizaje de máquina usan algoritmos de optimización que iteran sobre los datos introducidos para encontrar los mejores resultados. Valores cercanos a 100% indican un alto grado de precisión mientras que valores cercanos a cero indican un bajo desempeño del optimizador.

iv. Cantidad de errores en las predicciones.

Para comprobar el desempeño experimental del receptor se presentan los resultados la cantidad de errores en las predicciones cuando la SNR cambia entre -20 dB y +18 dB. El escenario de pruebas usado es el descrito en la sección II.

Con base en los parámetros anteriores, en la siguiente sección se muestran los resultados de este trabajo.

V. RESULTADOS.

Los resultados se organizaron en dos secciones, la concerniente al desempeño de la fase de entrenamiento y posteriormente los resultados experimentales basados en las predicciones de la red neuronal sobre datos nuevos.

A. Resultados de la Fase de Entrenamiento de la Red Neuronal.

La red neuronal implementada en el receptor se entrenó con un grupo de 19,534 datos, un conjunto de validación de 10% y

usando Adam como algoritmo de optimización con 90 iteraciones para la etapa de entrenamiento. La función de activación de las capas ocultas de la red neuronal fue Relu, a excepción de la última capa donde se utilizó la función de activación SOFTMAX. La Fig. 7 muestra la precisión de los algoritmos de optimización donde la red neuronal con SVD-SCMA alcanzó una precisión de 95.71%, mientras que SCMA mediante rotación de fase presentó 93.53%.

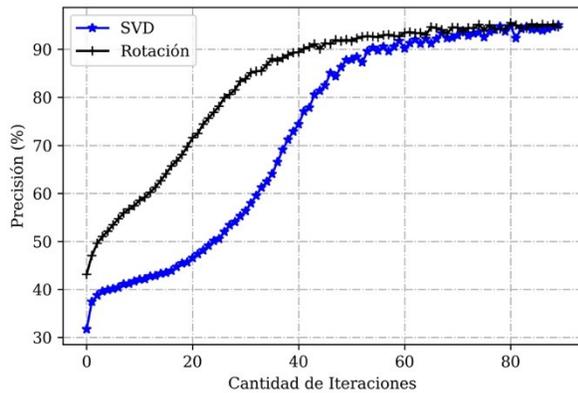


Fig. 7. Precisión de la red neuronal.

De la Fig. 7 se observa que, el método de rotación de fase tiene una mejor precisión durante el periodo inicial de la fase de entrenamiento, y SVD-SCMA tiene una mejora menos rápida; sin embargo, a partir de la iteración 68 ambos métodos presentan una precisión similar al 93%, para la iteración 90 SVD logra su mejor desempeño con 95.71% mientras que el máximo de rotación de fase es de 93.53%. A partir de la iteración 80 ya no se presenta una mejora significativa en el aprendizaje de ambos algoritmos de optimización.

El segundo aspecto que se evaluó fue la función de pérdida, también conocida como función objetivo (objective function); la Fig. 8, muestra el comparativo entre los dos métodos presentados en este trabajo; usando el método de rotación de fase se obtuvo una función de pérdida de 0.1825 mientras que SVD-SCMA logró 0.2082.

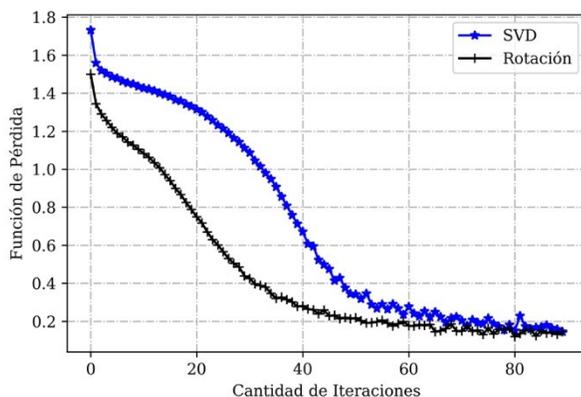


Fig. 8. Función de pérdida para SCMA.

Se observa de la Fig. 8, que el método de rotación de fase alcanza niveles bajos con menos iteraciones que SVD-SCMA, sin embargo, a partir de la iteración 76 ambos algoritmos de

optimización presentan un desempeño semejante, y después de la iteración 80 los algoritmos ya no mejoran su desempeño significativamente.

B. Resultados Experimentales.

Una vez que la red neuronal ha sido entrenada y se han obtenido sus parámetros de rendimiento (precisión y función de pérdida), se procede a generar datos nuevos y basado en el aprendizaje realizado, a hacer predicciones. Como se mencionó anteriormente, para cada SNR entre -20 dB a +18 dB, se generaron 1 millón de tripletas de bits y la red neuronal basado en su aprendizaje realiza las predicciones usando el escenario de pruebas descrito en la sección II. La Fig. 9, muestra la cantidad de errores para la sección de fase y SVD-SCMA en función de la relación señal a ruido:

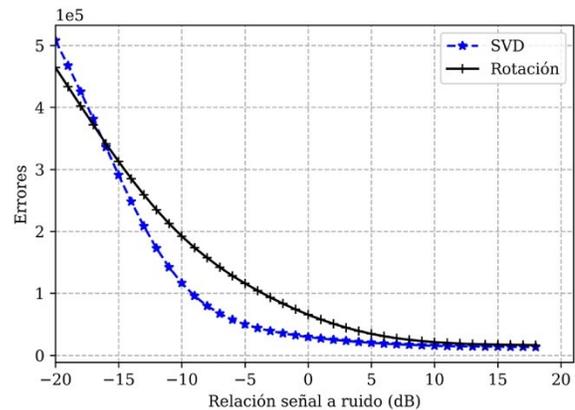


Fig. 9. Cantidad de errores en función de la SNR.

De la Fig. 9 se observa que SVD-SCMA tiene un mejor desempeño ante condiciones de ruido adversas (entre -15 y +11 dB de SNR), para lograr 2 % de errores SVD-SCMA lo logra con +5 dB y rotación de fase requiere +11 dB. Para SNR cercanas a +20 dB ambos métodos presentan un rendimiento semejante.

En la Fig. 10 se muestra la precisión en las predicciones para SVD-SCMA y rotación de fase. Para conseguir una precisión de 98 % SVD-SCMA requiere una SNR de +5 dB, mientras que rotación de fase necesita una SNR de +11 dB.

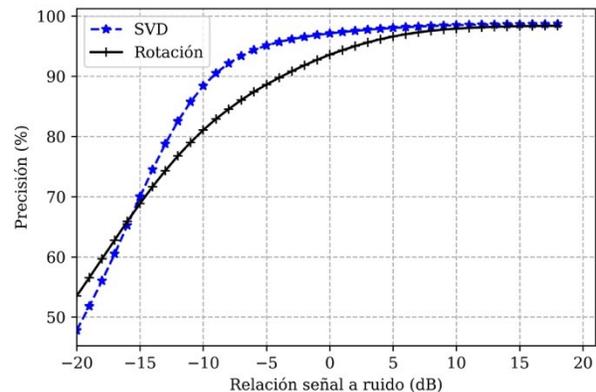


Fig. 10. Precisión de las predicciones.

De la Fig. 10, se confirma que SVD-SCMA tiene un mejor desempeño en condiciones adversas de SNR (entre -15 dB a 11 dB).

VI. CONCLUSIONES.

En este trabajo se propone y evalúa emplear una constelación de 8-APSK para codificar los datos de usuario en palabras de código SCMA (usando SVD y rotación de fase) en el enlace ascendente de un sistema de comunicación móviles de quinta generación, para verificar el desempeño de esta propuesta se implementó un escenario experimental usando radios definidos por software empelando HackRF One como sistema embebido para programar las funciones de transmisión y recepción de datos. Las principales contribuciones de este trabajo se resumen en:

- i. Los resultados experimentales demuestran que es posible emplear 8-APSK conservando un bajo número de errores en la detección y decodificación de señales SCMA, aún en presencia de SNR desfavorables.
- ii. El método de la descomposición de valores singulares para generar palabras de código SCMA, presentó mejor desempeño (precisión y cantidad de errores) que el método tradicionalmente utilizado de rotación de fase.
- iii. Se propuso un detector basado en redes neuronales con 6 capas densas y Adam como algoritmo para entrenar la red neuronal. En la detección, no es posible comparar con MPA, ya que éste fue diseñado para detectar y decodificar solo dos bits de información.
- iv. El receptor basado en redes neuronales, con SVD-SCMA para generar palabras de código, presentó mejor desempeño en condiciones de alto ruido (entre -15 dB y +11 dB), lo cual lo hace mejor opción que su contraparte usando rotación de fase.
- v. Para condiciones, de SNR menores a -20 dB, la tendencia de las curvas indica que rotación de fase, podría tener un mejor desempeño que SVD, lo cual podría hacer a rotación de fase un mejor candidato en situaciones muy desfavorables del canal inalámbrico.

Adam es un algoritmo de optimización ampliamente empleado para entrenar redes neuronales, en [28] apareció la primera propuesta de este algoritmo. En [29]-[30] se establece que para Adam no se ha realizado una prueba de la convergencia para encontrar soluciones óptimas, en [29] se presenta una prueba de la convergencia de Adam cuando se usa el modo batch completo, de esta forma se garantiza la convergencia local aún para problemas no-convexos. Posteriormente, en [30] se expone que una de las razones por la cual los algoritmos de optimización estocásticos (entre ellos Adam, SGD, etc) no alcanzan una solución óptima es debido al promedio del movimiento exponencial del gradiente usado, produciendo una solución no-óptima. Para resolver este problema, en [30] se propone dotar a tales algoritmos de "memoria a largo plazo" de los gradientes pasados, y se proponen nuevas variantes de Adam que no solo solucionan

los problemas de convergencia, sino que a menudo también conducen a un rendimiento mejorado.

A diferencia de lo mostrado en [29], en este trabajo, se usa un tamaño de batch de 50, esto podría ser la razón por la cual Adam no logra un 100 % de aprendizaje ni una función de pérdida de 0, usar Adam con batch completo, podría aumentar la precisión en el entrenamiento y por consecuencia en la precisión de las predicciones, sin embargo, esto implica mayor tiempo de máquina ya que todos los datos de entrenamiento deben ser procesados en cada iteración. Otra extensión de este trabajo es considerar, como se describe en [30], modificar Adam con una memoria de largo plazo de los gradientes usados para actualizar los parámetros de la red neuronal adaptativamente y de esta forma aumentar el desempeño del algoritmo.

Finalmente, con base en los resultados obtenidos en este trabajo se demuestra que es posible emplear constelaciones más complejas para codificar los datos de los usuarios en el enlace ascendente usando SVD-SCMA para generar las palabras de código y utilizando un receptor basado en redes neuronales. El rendimiento de las predicciones podría mejorar modificando Adam con una memoria de largo plazo de los gradientes usados para actualizar los parámetros de la red neuronal adaptativamente.

AGRADECIMIENTOS.

Los autores agradecen al Instituto Politécnico Nacional, el soporte recibido para llevar a cabo este trabajo. Esta investigación se desarrolló bajo el proyecto de investigación No. 20201028, financiado por el Instituto Politécnico Nacional.

REFERENCIAS.

- [1] L. Dai, B. Wang, Y. Yuan, S. Han, and Z. Wang, "Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 74-81, Sept. 2015.
- [2] J. Bao, Z. Ma, Z. Ding, G. K. Karagiannidis and Z. Zhu, "On the design of multiuser codebooks for uplink SCMA systems". *IEEE Communications Letters*, vol. 20, no. 10, pp. 1920-1923, Oct. 2016.
- [3] E. A. Degollado-Rea, S. Vidal-Beltrán, and J. L. López-Bonilla, "Simplified Equation for Received Power Calculation of the LTE Signal in 2.13 GHz," *Applied Mathematics & Information Sciences*, vol. 12, no. 6, pp. 1177-1182, 2018.
- [4] L. Yu, X. Lei, P. Fan, and D. Chen, "An optimized design of SCMA codebook based on star-QAM signaling constellations," in 2015 International Conference on Wireless Communications Signal Processing (WCSP), Nanjing, China, Oct 2015.
- [5] J. L. López-Bonilla, S. Vidal-Beltrán, I. Santiago-Rivera, and F. Martínez-Piñón, "Construction of SCMA CodeBooks using the phase rotation method," in *IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. Ixtapa, México, 2018.
- [6] S. Liu, J. Wang, J. Bao and C. Liu, "Optimized SCMA Codebook Design by QAM Constellation Segmentation with Maximized MED," *IEEE Access*, vol. 6, pp. 63232-63242, 2018.
- [7] C. Yan, G. Kang and N. Zhang, "A Dimension Distance-Based SCMA Codebook Design," *IEEE Access*, Vol. 5, pp. 5471-5479, 2017.
- [8] S. Lou, C. Gong, Q. Gao and Z. Xu, "SCMA with Low Complexity Symmetric Codebook Design for Visible Light Communication," in *IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, 2018.
- [9] H. Mu, Z. Ma, M. Alhaji, P. Fan and D. Chen, "A Fixed Low Complexity Message Pass Algorithm Detector for Up-Link SCMA

- System," *IEEE Wireless Communications Letters*, vol. 4, no. 6, pp 585-588, 2015.
- [10] L. Yang, Y. Liu, Y. Siu, "Low complexity message passing algorithm for SCMA system", *IEEE Communications Letters*, vol. 20, pp. 2466-2469, 2016.
- [11] A. Bayesteh, H. Nikopour, M. Taherzadeh, H. Baligh, and J. Ma. "Low complexity techniques for SCMA detection". In 2015 IEEE Globecom Workshops (GC Wkshps), pp. 1-6, San Diego, CA, USA, Dec 2015.
- [12] S. Vidal-Beltrán, J. L. López-Bonilla, and F. Martínez-Piñón, "The Euclidean distance-based detection method applied on SCMA Codewords" *International Journal of Latest Research in Science and Technology*, vol. 7, no. 3, pp. 14-18, May-June 2018.
- [13] G. W. Stewart, "On the early history of the SVD", *SIAM Review*, vol. 35, pp. 551-566, 1993.
- [14] A. Huseyin. *Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems*. Springer, 2007.
- [15] M. Baldi, F. Chiaraluce, A. de Angelis, R. Marchesani, S. Schillaci, "A comparison between APSK and QAM in wireless tactical scenarios for land mobile systems", *EURASIP Journal on Wireless Communications and Networking*, Oct. 2012. <https://doi.org/10.1186/1687-1499-2012-317>.
- [16] H. Zamiri-Jafarian and G. Gulak, "Adaptive channel SVD estimation for MIMO-OFDM systems", in *IEEE 61st Vehicular Technology Conference*, Stockholm, Sweden, 2005.
- [17] W. Chun-Hun, L. Chin-Yi and T. Pei-Yun, "Design of an SVD engine for 8 x 8 MIMO precoding systems", in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore, MD, USA, 2017.
- [18] M. Moghaddari and H. Zamiri-Jafarian, "Algorithms of channel SVD estimation in MIMO-OFDM systems: Evaluation and comparison", in *International Symposium on Telecommunications*, Tehran, Iran, 2008.
- [19] V. Gaftoi, J. López-Bonilla, G. Ovando, "Singular value decomposition and Lanczos potential", *Current Topics in Quantum Field Theory Research*, ed. O. Kovras, Nova Science Pub., New York, 2007.
- [20] R. C. Minnick, B. Elspas, R. A. Short, "Symmetric Latin Squares". *IEEE Transactions on Electronic Computers*. vol. EC-12, no. 2, pp. 130-131, Apr. 1963.
- [21] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498-519, Feb. 2001.
- [22] A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.
- [23] O. Simeone, "A Very Brief Introduction to Machine Learning with Applications to Communication Systems", *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648-664, Dec. 2018.
- [24] A. S. Brandao, D. C. Jorge, "Artificial Neural Networks Applied to Image Steganography", *IEEE Latin America Transactions*, vol. 14, no. 3, pp. 1361-1366, 2016.
- [25] J. Wu, I. Bisio, C. Gniady, E. Hossain, M. Valla, H. Li, "Context-aware networking and communications: Part 1", *IEEE Communications Magazine*, vol. 52, no. 6, June 2014.
- [26] R. Atat, L. Liu, J. Wu, G. Li, C. Ye, Y. Yang, "Big Data Meet Cyber-Physical Systems: A Panoramic Survey", *IEEE Access*, vol. 6, pp. 73603-73636, November 2018.
- [27] S. Vidal-Beltrán, J. L. López-Bonilla, and F. Martínez-Piñón, "Supervised learning applied to the decoding of SCMA codewords", *IEEE Latin America Transactions*, vol. 17, no. 11, pp. 1843-1848, 2019.
- [28] D. P. Kingma, J. Lei-Ba, "Adam: A method for stochastic optimization," in the 2nd International Conference on Learning Representations, San Diego, USA, 2015.
- [29] S. Bock, M. Wei, "A Proof of Local Convergence for the Adam Optimizer", in *International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, 2019.
- [30] S. Reddi, S. Kale and S. Kumar, "On the convergence of Adam and beyond", *International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018.



Sergio Vidal-Beltrán profesor e Investigador Científico en la Maestría en Ciencias en Ingeniería de Telecomunicaciones del Instituto Politécnico Nacional, México. Especialista en Interconectividad de redes y en sistemas de comunicaciones móviles.



Fernando Martínez-Piñón profesor e Investigador científico del Centro de Investigación e Innovación Tecnológica (CIITEC) del Instituto Politécnico Nacional, México. Especialista en Fotónica (Fibras Ópticas para Telecomunicaciones y Sensores), Dispositivos láser, Dispositivos electro-ópticos y Radiocomunicaciones.



José Luis López-Bonilla profesor e Investigador científico de la Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME) del Instituto Politécnico Nacional, México. Especialista en Métodos Matemáticos Aplicados a la Ingeniería y Física Teórica.