

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIEROS ELECTRÓNICOS**

**TEMA:
COMPARACIÓN DE DOS ALGORITMOS DE GENERACIÓN DE HACES
ADAPTATIVOS PARA MIMO MASIVO CON MATLAB**

**AUTORES:
JORGE LUIS CHÁVEZ GARCÍA
DOUGLAS SEBASTIAN UNAPANTA ALBÁN**

**TUTORA:
VERÓNICA EMMA SORIA MALDONADO**

Quito, agosto del 2021

CESIÓN DE DERECHOS DE AUTOR

Nosotros Jorge Luis Chávez García y Douglas Sebastian Unapanta Albán, con documentos de identificación N° 1722270475 y N° 1725574782 respectivamente, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: COMPARACIÓN DE DOS ALGORITMOS DE GENERACIÓN DE HACES ADAPTATIVOS PARA MIMO MASIVO CON MATLAB, mismo que ha sido desarrollado para optar por el título de: Ingenieros Electrónicos, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



Jorge Luis Chávez García

CI: 1722270475



Douglas Sebastian Unapanta Albán

CI: 1725574782

Quito, agosto del 2021

DECLARATORIA DE COAUTORIA DEL DOCENTE TUTORA

Yo declaro que bajo mi dirección y asesoría fue desarrollado el artículo académico, COMPARACIÓN DE DOS ALGORITMOS DE GENERACIÓN DE HACES ADAPTATIVOS PARA MIMO MASIVO CON MATLAB, realizado por Jorge Luis Chávez García y Douglas Sebastian Unapanta Albán, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, agosto del 2021



Verónica Emma Soria Maldonado

CI: 1715185961

Comparación de dos algoritmos de generación de haces adaptativos para MIMO masivo con Matlab.

Jorge Luis Chávez García
Departamento de Ingeniería
Electrónica y
Telecomunicaciones
Quito, Ecuador
jchavezg3@est.ups.edu.ec

Douglas Sebastian Unapanta Albán
Departamento de Ingeniería
Electrónica y
Telecomunicaciones
Quito, Ecuador
dunapanta@est.ups.edu.ec

Verónica Soria Maldonado
Departamento de Ingeniería
Electrónica y
Telecomunicaciones
Quito, Ecuador
vsoria@ups.edu.ec

Resumen—Este artículo da a conocer la comparación entre los algoritmos de generación de haces adaptativos LMS y RLS para MIMO masivo con Matlab. Para determinar el algoritmo más eficiente en una comunicación inalámbrica se evalúan los parámetros, patrón de radiación, curva de aprendizaje y bit error rate (BER) en un sistema de 100 antenas en el transmisor y 1 en el receptor con modulación 16 QAM. Dando como resultado que el algoritmo LMS es dependiente de su tamaño de paso μ y necesita de un elevado número de iteraciones para converger, mientras tanto, el algoritmo RLS es de rápida convergencia, pero requiere de mayor costo computacional, en definitiva, el algoritmo RLS es más idóneo que el algoritmo LMS.

Palabras clave—MIMO masivo, LMS, RLS, BER, MSE, ULA

Abstrac— This paper reports the comparison between LMS and RLS adaptive beamforming algorithms for massive MIMO with Matlab. To determine the most efficient algorithm in a wireless communication, the parameters, radiation pattern, learning curve and bit error rate (BER) are evaluated in a system of 100 antennas in the transmitter and 1 in the receiver with 16 QAM modulation. As a result, the LMS algorithm is dependent on its step size μ and needs a high number of iterations to converge, while the RLS algorithm is fast converging, but requires higher computational cost, in short, the RLS algorithm outperforms the LMS algorithm.

I. INTRODUCCIÓN

Las redes de comunicaciones inalámbricas han tenido una evolución importante hasta la actualidad, esto ha ocasionado una gran demanda por parte de los usuarios para que sus dispositivos electrónicos se conecten a la Internet. Además, con el crecimiento de las redes inalámbricas y con la necesidad del uso de bandas de frecuencias con un amplio ancho de banda, se ha potencializado en gran medida la tecnología de quinta generación (5G) a través del uso del Internet de las cosas o también denominado IoT. Este crecimiento y sobre todo la gran demanda de usuarios para el uso de redes inalámbricas han provocado la implementación de tecnologías que aumenten la velocidad de transmisión de la información a bajas latencias y reduzcan las interferencias, es decir, existe interés en las comunicaciones inalámbricas que trabajen a través de ondas milimétricas y permitan la implementación de una gran cantidad de antenas.[1][2] Este artículo realiza una comparación entre los algoritmos de generación de haces adaptativos LMS y RLS con la finalidad de determinar el algoritmo más idóneo en una comunicación inalámbrica, para ello, se plantean 3 escenarios conformados por un arreglo de antenas (100) en el transmisor y un receptor

de antena única con 4 señales interferentes que ponen a prueba a la eficiencia de los algoritmos. El artículo se compone de seis secciones, la sección dos describe a los algoritmos adaptativos, la sección tres describe el arreglo de sensores, la sección cuatro el modelo y sistema simulado, la sección cinco muestra el análisis y resultados obtenidos de la simulación en matlab, la sección seis da a conocer las conclusiones de la investigación, y finalmente la sección siete muestra las referencias bibliográficas utilizadas.

A. MIMO masivo

El sistema MIMO masivo, por sus siglas Multiple-Input Multiple-Output, que representa el avance de la tecnología MIMO utilizada en la actualidad [3], esta tecnología es también conocida como Large-Scale Antenna Systems, Very Large MIMO, Hyper MIMO y Full-Dimension MIMO, estos sistemas utilizan arrays de una gran cantidad de antenas (decenas a centenas), las cuales pueden ayudar a enfocar la energía en una región pequeña del espacio para proporcionar una mejor eficiencia espectral y rendimiento. El servicio simultaneo proporcionado a los terminales es limitado por la incapacidad de adquirir la información del canal para un número ilimitado de terminales [4], los cuales utilizan los mismos recursos de frecuencia y tiempo. Está diseñado para funcionar con Duplexación por División de Tiempo (TDD, Time-Division Duplex), junto con pilotos en el enlace ascendente los cuales permiten realizar la estimación del estado del canal (CSI) para obtener información del mismo, teniendo en cuenta la interferencia entre celdas, la sobrecarga y los errores. [1]

La implementación de tecnología MIMO masivo presenta una serie de problemas y desafíos, entre los que se pueden citar, contaminación piloto, estimación de canal, precodificación, detección de señales codificadas, generación de haces adaptativos entre otros.

B. Beamforming

El beamforming es una técnica que tiene como principal característica dirigir la señal de varias antenas inteligentes que conforman la tecnología MIMO masivo hacia el receptor, esto mediante la reducción de la cantidad de energía que se genera en los lóbulos laterales en el patrón de radiación del transmisor.[5]

En [5] se menciona que se tienen dos tipos de formadores de haz para beamforming:

- Formadores de haz fijo, utilizan pesos y fases invariantes para adoptar señales sin tomar en cuenta las características de las señales que se reciben.
- Formadores de haz adaptable, permiten enfocar el lóbulo principal de radiación hacia la orientación del receptor, reduciendo interferencias en la transmisión.

Las técnicas de beamforming en la tecnología MIMO masivo presentan las siguientes ventajas: mayor eficiencia energética, mayor eficiencia espectral, mayor seguridad del sistema y aplicabilidad para bandas de ondas milimétricas.[6]

II. ALGORITMOS ADAPTATIVOS

Para realizar el cálculo de los posibles pesos \mathbf{w} en un filtro adaptativo es necesario basarse en un criterio de optimización que permite minimizar o maximizar una función de costo (CF) definida $J(\mathbf{w})$. Además, si la naturaleza de la señal de entrada es no determinista (estocástica), el CF es una función de alguna estadística de la señal de error. En estos casos, es habitual considerar el valor esperado o valor promedio de conjunto o valor medio, indicado por el operador $E\{\cdot\}$ del cuadrado de la señal de error $e[n]$. Tal cantidad se la conoce como error cuadrático medio (MSE) y se define como.[7]

$$J(\mathbf{w}) = E\{|e[n]|^2\} \quad (1)$$

A. Algoritmo Least-Mean-Square (LMS)

LMS es un algoritmo adaptativo que necesita una señal de señal deseada para minimizar el error entre la salida del conformador del haz y la señal deseada, cabe mencionar que este algoritmo tiene una formulación sencilla pero matemáticamente es complejo de analizar. En comparación con otros algoritmos adaptativos, requiere un mayor número de iteraciones para llegar a la convergencia.[8][9]

$$y[n] = \mathbf{w}_n^H \mathbf{x} \quad (2)$$

Al vector de pesos se le realizan correcciones en la dirección negativa del vector gradiente para la minimización de $J(\mathbf{w})$, tendiendo hacia el error cuadrático medio, la función que describe la actualización de los pesos es la siguiente:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{1}{2} \mu [-\nabla \hat{J}(\mathbf{w}_{n-1})] \quad (3)$$

donde \mathbf{w}_n es el peso actualizado, \mathbf{w}_{n-1} es el peso anterior, μ es el tamaño de paso y controla las características de convergencia de LMS, y denotado como el CF $\hat{J}_n(\mathbf{w}) = |e[n]|^2$, como error instantáneo, donde el error se define como: [10]

$$e[n] = d[n] - y[n] \quad (4)$$

El vector gradiente $\nabla \hat{J}_{n-1}$ de (3) evaluado en el paso n simplemente se convierte en:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu e[n] \mathbf{x}_n \quad (5)$$

B. Algoritmo Normalized Least-Mean-Square (NLMS)

NLMS es un algoritmo de gradiente estocástico que se deriva del algoritmo LMS con la diferencia que permite actualizar su valor de paso de adaptación (μ) normalizándolo de acuerdo con los valores que toma el vector de entrada, es decir, este algoritmo tiene un comportamiento robusto y estable cuando se tratan de entradas o señales no estacionarias. [11]

La principal característica de NLMS es tener una tasa de aprendizaje variable de acuerdo con la siguiente ecuación: [7]

$$\mu(n) = \frac{\mu}{\delta + \|\mathbf{x}_n\|_2^2} \quad (6)$$

donde δ es el parámetro de regularización y asegura la computabilidad y \mathbf{x}_n es la señal de entrada.

Finalmente, la ecuación de actualización de paso es: [7]

$$\mathbf{W}_n = \mathbf{W}_{n-1} + \mu \frac{e^*[n] \mathbf{x}_n}{\delta + \mathbf{x}_n^H \mathbf{x}_n} \quad (7)$$

C. Algoritmo Recursive Least Square (RLS)

Es un algoritmo cuyo objetivo es encontrar los coeficientes del filtro recursivamente de modo que la señal deseada coincida lo más cercano posible con la señal de salida, es eficiente para entornos que varían en el tiempo. Este algoritmo utiliza el método de mínimos cuadrados recursivos con la finalidad de ajustar el vector de pesos. Utiliza el vector de Gancia Kalman para la ecuación de actualización de peso.[12]

Las ecuaciones más importantes del algoritmo RLS se muestran a continuación:

$$y[n] = \mathbf{w}_n^T \mathbf{x} \quad (8)$$

Donde $n = 1, 2, \dots, N$, $y[n]$ representa la salida del filtro, \mathbf{w}_n son los valores de peso que se genera por el algoritmo, \mathbf{x} es el vector de las señales de entrada.

$$e[n] = d[n] - y[n] \quad (9)$$

$e[n]$, es la señal de error producida, $d[n]$, es la señal deseada.

$$\mathbf{k}_n = \frac{\lambda^{-1} \mathbf{P}_{n-1} \mathbf{x}_n}{1 + \lambda^{-1} \mathbf{x}_n^H \mathbf{P}_{n-1} \mathbf{x}_n} \quad (10)$$

$$\mathbf{P}_n = \lambda^{-1} \mathbf{P}_{n-1} - \lambda^{-1} \mathbf{k}_n \mathbf{x}_n^H \mathbf{P}_{n-1} \quad (11)$$

\mathbf{k}_n , es la ganancia de Kalman, $\mathbf{P}_n = \mathbf{R}_{xx}^{-1}$, es la inversa de la matriz de correlación, λ es una constante positiva tal que $0 \leq \lambda \leq 1$ el cual es llamado factor de olvido, y tiene en cuenta la memoria del algoritmo, se denomina memoria creciente, para $\lambda = 1$, mientras que para $0 \ll \lambda \leq 1$, el algoritmo se denomina RLS ponderado exponencialmente (EWRLS). [13] El factor de olvido permite disminuir el error de convergencia entre la señal de referencia y el resultado de

la identificación, es decir, dicho factor descarta información que ya no es necesaria.[14]

Donde el factor de pesos queda definido por:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mathbf{k}_n[d^*[n] - \mathbf{w}_{n-1}^H \mathbf{x}_n] \quad (12)$$

III. ARREGLO DE SENSORES

El procesamiento de señales de una matriz de sensores homogénea y distribuida espacialmente se conoce como procesamiento de matriz (AP), de su nombre en inglés Array Processing.

El propósito del AP es utilizar un conjunto de sensores adecuadamente distribuidos los cuales realizan un muestreo espacio temporal de una onda viajera de un campo electromagnético con la finalidad de extraer información en presencia de ruido e interferencia que existen en el medio de propagación. [7]

El modelo de la señal debido a la propagación espacial es el resultante de la ecuación de onda que se puede escribir como función de espacio tiempo $s(t, r)$, en donde se representa la cantidad de forma de onda en un área de interés, donde r representa el vector posición, $r = [x \ y \ z]^T$, para ondas planas en coordenadas cartesianas se puede escribir la ecuación de la onda como: [7]

$$s(t, \mathbf{r}) = S_0 e^{j(\omega t - \mathbf{r}^T \mathbf{k})} \quad (13)$$

donde S_0 representa la amplitud, variable en el tiempo $S_0 \rightarrow S(t)$, ω representa la frecuencia angular ($\omega = 2\pi f$, $f = \frac{1}{T}$), el vector \mathbf{k} representa el número de onda y se define como $\mathbf{k} = k \cdot [\sin \theta \cos \phi \ \sin \theta \sin \phi \ \cos \theta]^T$, con $k = |\mathbf{k}|$ y $\mathbf{k} = k \cdot \mathbf{d}(\theta)$, donde $\mathbf{d}(\theta)$, es la dirección de llegada (DOA). [7]

La señal en la proximidad de sensor ubicado en una posición \mathbf{r}_m se puede escribir:

$$s_m(t, \mathbf{r}) = S_0 e^{j(\omega t - \mathbf{r}_m^T \mathbf{k})} = S_0 e^{j\omega t} e^{-j\frac{\omega}{c} \mathbf{r}_m^T \mathbf{d}(\theta)} \quad (14)$$

A partir del supuesto de onda plana, el vector de número de onda \mathbf{k} no depende de la posición del sensor para lo cual los receptores se irradian con la misma señal, pero retardada.

$$s_m(t, \mathbf{r}) = s(t) e^{-j\mathbf{r}_m^T \mathbf{k}} \quad (15)$$

La ecuación se puede representar omitiendo la dependencia temporal. En este caso, la propagación puede ser representada por el número complejo definido como $\bar{S} = e^{-j\mathbf{r}_m^T \mathbf{k}}$, llamado modelo fasorial de propagación.

El steering se define como el fasor de la ecuación de la onda $\bar{a}_m(\mathbf{r}, \mathbf{k}) = e^{-j\mathbf{r}_m^T \mathbf{k}}$, esta variable contiene toda la información geométrica sobre la onda que irradia el m -ésimo sensor. [7]

Para una sola fuente de radiación, para P sensores, se define un vector de dirección también llamado vector director.

$$\bar{\mathbf{a}}(\omega, \theta) \in \mathbb{C}^{P \times 1} = [e^{-j\mathbf{r}_1^T \mathbf{k}} \ e^{-j\mathbf{r}_2^T \mathbf{k}} \ \dots \ e^{-j\mathbf{r}_P^T \mathbf{k}}]^T \quad (16)$$

Entonces para cada fuente de radiación se tiene un vector director asociado con su ángulo de arribo en cada sensor.

$$\mathbf{A}(\omega, \theta) \in \mathbb{R}^{P \times N_s} = \begin{bmatrix} a_{1,1}^T & a_{1,2}^T & \dots & a_{1,N_s}^T \\ a_{2,1}^T & a_{2,2}^T & \dots & a_{2,N_s}^T \\ \vdots & \vdots & \ddots & \vdots \\ a_{P,1}^T & a_{P,2}^T & \dots & a_{P,N_s}^T \end{bmatrix}_{P \times N_s}$$

IV. MODELO Y SISTEMA SIMULADO

A. Modelo del sistema

El objetivo principal de este artículo consiste en comparar los algoritmos de generación de haces adaptativos LMS y RLS para determinar el algoritmo más idóneo en una comunicación inalámbrica en un sistema MIMO masivo. Para lograr el objetivo se realizó simulaciones en el software Matlab para medir parámetros de eficiencia como, ángulo de apuntamiento hacia el receptor, curva de aprendizaje (MSE) para determinar el número de iteraciones necesarias para encontrar los pesos óptimos de cada sensor del array para la conformación del patrón de radiación. Finalmente, transmitir información por medio del array, por un canal AWGN y determinar el BER en el receptor.

En la figura 1 se observa el escenario basado en MIMO masivo, el cual tiene una estructura de celda única (SC), es decir, mediante una estación base (BS) conformada con un arreglo de antenas $M = 100$ sirve a un usuario de antena única.

- La modulación QAM que se usa para la transmisión de información es de índice 16.
- Cuatro señales interferentes randómicas, y una señal senoidal deseada.
- Canal: ruido blanco aditivo con distribución gaussiana AWGN con media cero y sin correlación con la señal de entrada, $\mathbf{v}_n \sim N(0, \sigma_v^2)$.

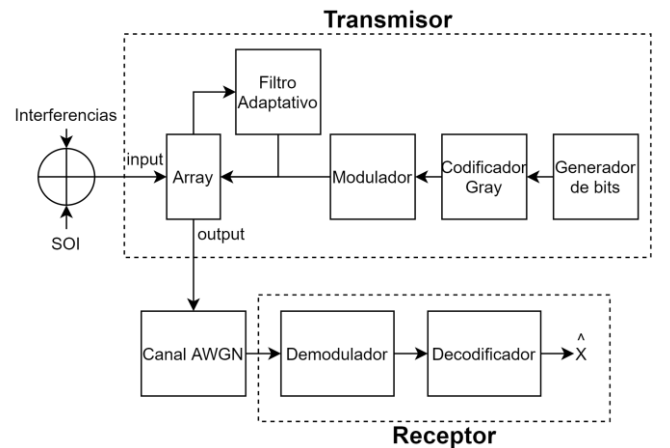


Fig. 1 Diagrama del sistema

En el array de sensores se utiliza un arreglo linealmente uniforme denominado ULA, en donde es habitual considerar la distribución de los sensores a lo largo de la dirección z, cada elemento del sensor está separado uno del otro a una distancia constante, teniendo en cuenta un sensor de referencia inicial en la posición $\mathbf{r}_1 = [0 \ 0 \ 0]^T$, entonces se tiene que las posiciones para los m sensores es $\mathbf{r}_m = [0 \ 0 \ (m-1)d]^T$,

donde $m = 1, \dots, P$, entonces el vector director para una sola fuente es:

$$a_m(\omega, \theta) = e^{-j\left(\frac{\omega}{c}\right)(m-1)d \cos \theta} \quad (17)$$

donde ω , representa la velocidad angular de la onda, c es la constante de velocidad de la luz, d es la distancia entre los sensores, m representa número de sensor, θ ángulo de arribo de la onda.

Para lo cual en el modelo de tiempo discreto se tiene que

$$x[n] = \mathbf{a}^T s[n] + \mathbf{n}[n] \quad (18)$$

donde s es una señal de cualquier forma, \mathbf{n} es el ruido.

Para el bloque de filtro adaptativo de la figura 1, se tienen los algoritmos LMS y RLS.

Dado que para el algoritmo LMS la convergencia depende del parámetro μ , se recomienda optar por valores dentro del rango que se muestra a continuación.

$$0 < \mu < \frac{2}{M * Pt} \quad (19)$$

donde M representa el número total de sensores, y Pt es la potencia promedio de la señal de datos. [15]

La velocidad de convergencia está determinada por el modo más lento de la matriz $\mathbf{R} \approx \mathbf{x}_n \mathbf{x}_n^H$, definida como:

$$\tau_{LMS} \cong \frac{1}{\mu \lambda_{min}} \quad (20)$$

Si el tamaño de μ es demasiado pequeño, la convergencia es lenta y si el tamaño del μ es demasiado grande, el algoritmo LMS sobrepasará los pesos óptimos de interés.

Para el bloque del filtro adaptativo de la figura 1 con algoritmo RLS, se consideró utilizar la regularización de Tikhonov el cual impone una restricción de amortiguamiento al conjunto de posibles soluciones y evita el mal acondicionamiento de la matriz de autocorrelación \mathbf{R}_{xx} . Adicional, se consideró utilizar el algoritmo de solución δ o también llamado variante de Levenberg-Marquardt, debido a que la matriz de autocorrelación \mathbf{R}_{xx} puede estar mal acondicionada y su inversión pueda causar inestabilidad numérica. El método consiste en añadir a la matriz $\mathbf{X}\mathbf{X}^H$ una matriz diagonal $\delta \mathbf{I}$, donde δ representa una amplitud mínima constante, el cual hace que la matriz siempre sea invertible $\delta > 0$, generalmente este valor es seleccionado inversamente proporcional a la relación de señal a ruido (SNR), entonces $\delta = 1/SNR$. [7]

Al iniciar el algoritmo RLS se utiliza el parámetro de regulación δ , como:

$$\mathbf{P}_n = \frac{1}{\delta} \mathbf{I} \quad (21)$$

evitando el mal acondicionamiento de \mathbf{R}_{xx} , cuando $\mathbf{P}_n = \mathbf{R}_{xx}^{-1}$, \mathbf{I} es la matriz identidad ($M \times M$), donde M es igual al número de sensores en el array.

A diferencia de LMS, la velocidad de convergencia de RLS no depende de los valores propios de la matriz de correlación \mathbf{R}_{xx} , y su convergencia es controlada por el factor de olvido λ

$$\tau_{RLS} = \frac{1}{\ln \lambda} \quad (22)$$

En la figura 2, se muestra el diagrama de flujo del funcionamiento del programa simulado en Matlab. Se tiene una señal deseada $d[n]$ de tamaño de N muestras que sumada con otras señales interferentes $s_i[n]$ arriban en cada sensor m del array con un ángulo específico. Las señales son muestreadas por los sensores e ingresan al filtro adaptativo LMS y RLS, para calcular los pesos w de los sensores con cada momento de la señal, al terminar el filtrado adaptativo, se generan bits aleatorios, los cuales se codifican, modulan y se envían por un canal AWGN, hacia el receptor, finalmente la información se demodula y decodifica, para obtener los datos enviados por el transmisor.

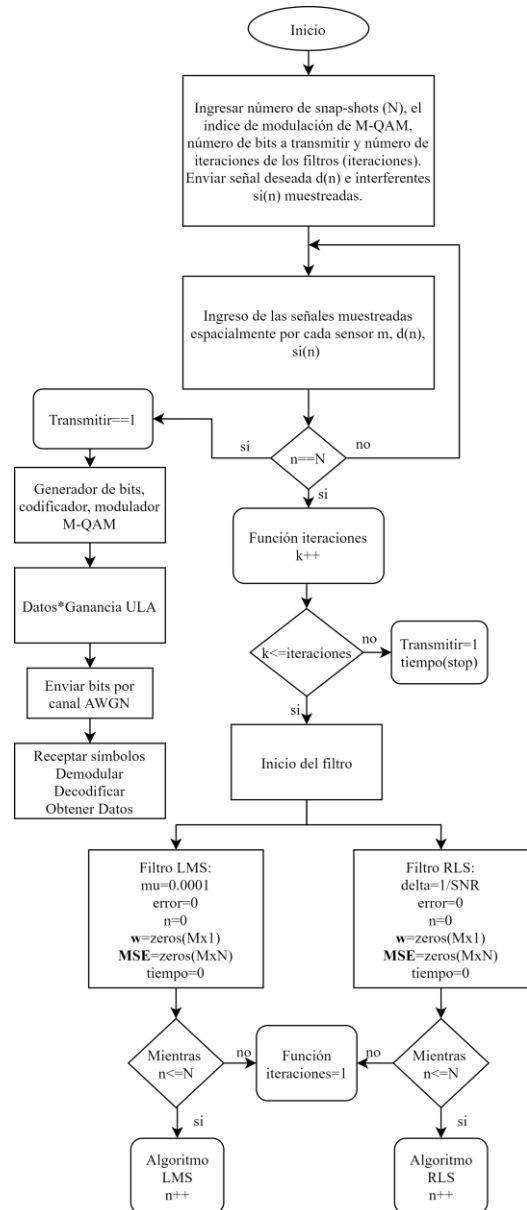


Fig. 2 Diagrama de flujo del programa

V. ANÁLISIS Y RESULTADOS

Los resultados obtenidos de la simulación se basan en 3 escenarios que permiten evidenciar el comportamiento de los algoritmos LMS y RLS. Cabe mencionar que los parámetros para considerar en cada escenario corresponden al ángulo de llegada, señales interferentes, índice de modulación y la cantidad de bits a transmitir.

A. Escenario 1

Este escenario pretende demostrar la velocidad de convergencia del algoritmo RLS frente al algoritmo LMS, mediante los parámetros que se muestran la tabla 1.

SNR	15
Ángulo de señales interferentes	10°, 70°, 20°, 0°
Ángulo Deseado (Rx)	-70°
Muestras	300
Índice de modulación (M-QAM)	16
Bits Transmitidos	10000

Tabla. 1. Parámetros simulados – Escenario 1

En las figuras 3 y 4, se muestran los resultados del ángulo de arribo de la señal deseada después de incidir en el array de sensores, y pasar por el filtro adaptativo con los algoritmos RLS y LMS. Se observa que el algoritmo RLS presenta un mejor valor de ganancia en su amplitud de 0.996268 apuntando a -69.8883° , a diferencia del algoritmo LMS el cual obtiene una ganancia para el array de 0.597063 en un ángulo de -69.8883° , es decir, existe una diferencia notable en las amplitudes de ganancia, pero ambos algoritmos generan un ángulo de apuntamiento que se aproxima al azimuth deseado, esta diferencia en la ganancia es debido a que el algoritmo LMS no converge, por lo tanto, no obtiene los pesos óptimos o ideales que proporcionan mayor ganancia del array en el ángulo de arribo de la señal deseada.

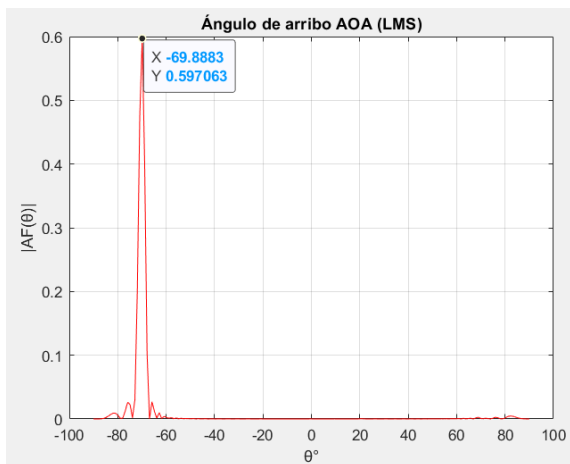


Fig. 3 Ángulo de arribo con algoritmo LMS – Escenario 1

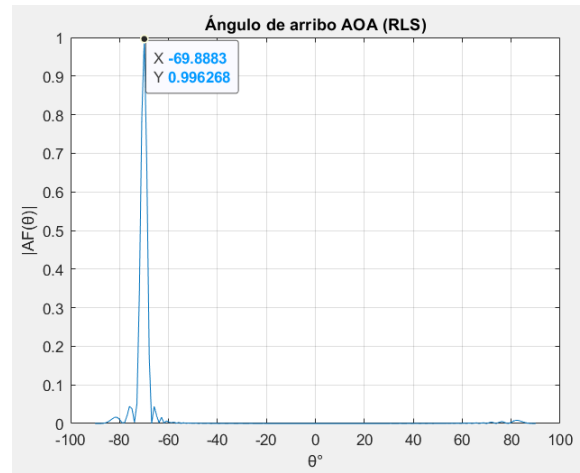


Fig. 4 Ángulo de arribo con algoritmo RLS – Escenario 1

En las figuras 5 y 6, se obtiene el patrón de radiación del array, el cual apunta a la dirección deseada con su ganancia expuesta en las figuras 3 y 4 respectivamente para cada algoritmo.

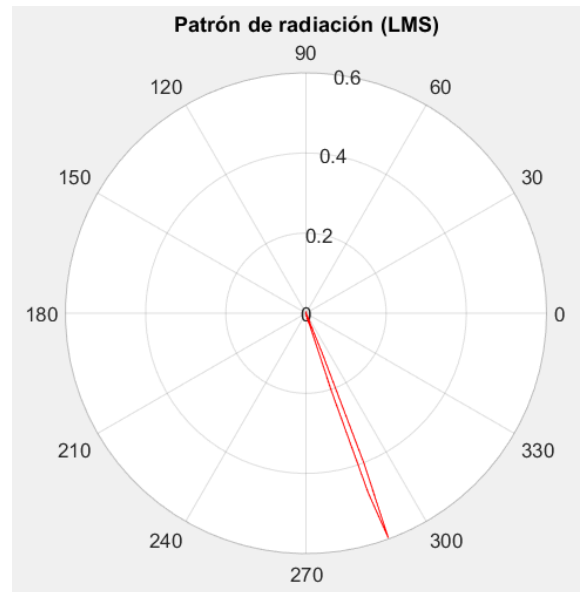


Fig. 5 Patrón de radiación con algoritmo LMS – Escenario 1

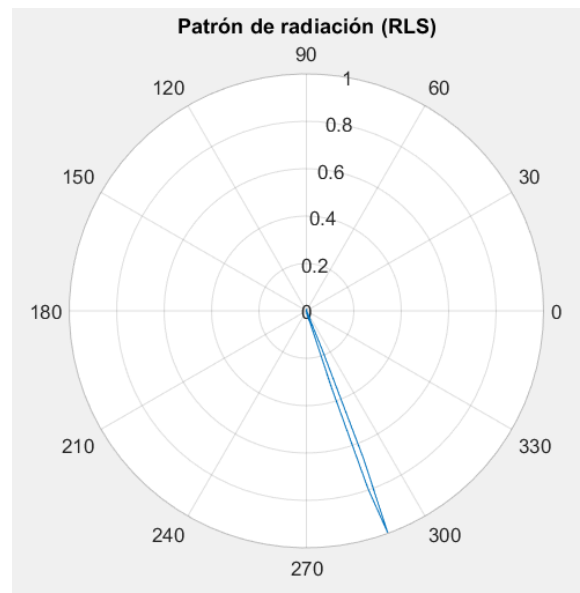


Fig. 6 Patrón de radiación con algoritmo RLS – Escenario 1

La figura 7 muestra el resultado del MSE para cada algoritmo simulado. Se puede observar que el algoritmo RLS requiere de un aproximado de 300 muestras (iteraciones del algoritmo) para llegar a la convergencia y formar un patrón de radiación con los pesos óptimos para cada sensor. El algoritmo LMS hace uso de las 300 iteraciones, pero no llega a converger, cabe mencionar que se hace uso de $\mu = 0.0001$, el mismo que se encuentra dentro del rango de la ecuación 19 y asegura que el algoritmo converja, en consecuencia, se evidencia que es necesario mayor número de iteraciones lo cual es una característica del algoritmo LMS.

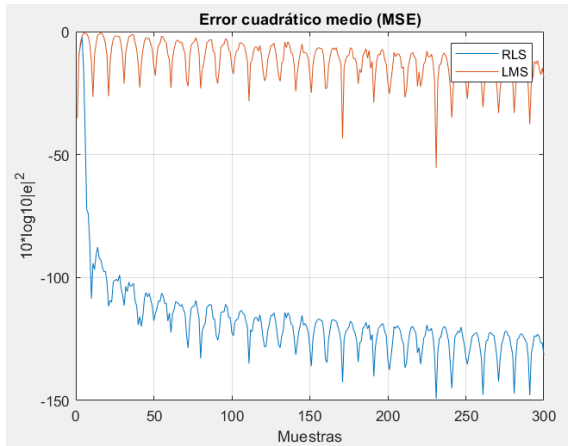


Fig. 7 Error cuadrático medio para los algoritmos LMS (rojo) y RLS (azul) – Escenario 1

En la figura 8, se muestran los tiempos que se demoraron cada algoritmo en filtrar las 300 muestras de la señal, dando como resultado un tiempo mayor para RLS (0.2742 segundos) y menor para LMS (0.1619 segundos).

Además se visualizan los valores de bit error rate para la señal transmitida por el array. Con el filtro RLS, se obtuvieron 46 bits erróneos de 10000, y para el array con filtro LMS 414 bits erróneos, es decir, al no converger el algoritmo LMS se obtiene un valor de BER más alto en comparación de RLS.

```

Command Window
Número de snap-shots del piloto... 300
Ingrese el índice de modulación: 16
Ingrese el número de bits: 10000

tiempo_RLS =
    0.2742

tiempo_LMS =
    0.1619

RLS: El BER es 4.60e-03, basado en 46 errores.
LMS: El BER es 4.14e-02, basado en 414 errores.
fx >>
    
```

Fig. 8 Resultados de tiempos de trabajo para los algoritmos LMS/RLS y BER – Escenario 1

símbolos transmitidos mediante el array con el algoritmo RLS se encuentran cercanos a sus respectivos puntos de constelación, a diferencia del algoritmo LMS donde se visualizan símbolos distantes de dichos puntos referenciales, corroborando los valores de BER de la figura 8.

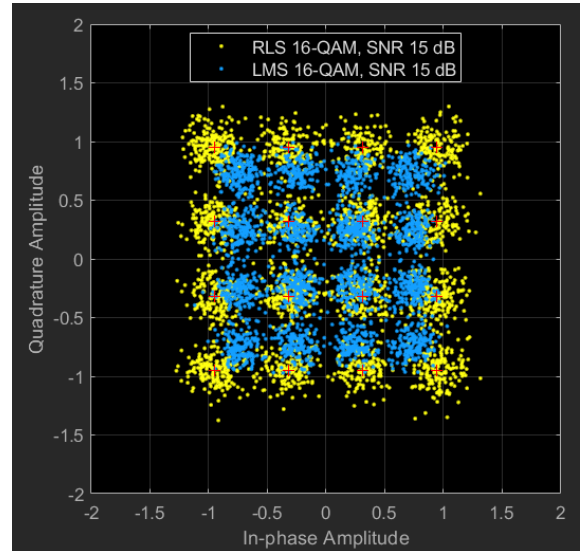


Fig. 9 Diagrama de constelaciones para los algoritmos LMS (azul) y RLS (amarillo) – Escenario 1

En la figura 10, se muestra la gráfica BER vs SNR. Se observa que para distintos valores de SNR se obtiene un valor de BER correspondiente. Tomando como referencia un $BER = 10^{-3}$ que es el mínimo recomendado en comunicaciones inalámbricas, se determina que el algoritmo RLS requiere un $SNR = 16.4 \text{ dB}$, mientras que, el algoritmo LMS requiere un $SNR = 23.4 \text{ dB}$, esto es debido a que el algoritmo RLS presenta mayor ganancia en el ángulo de arriba en comparación con algoritmo LMS como se observó en las figuras 3 y 4.

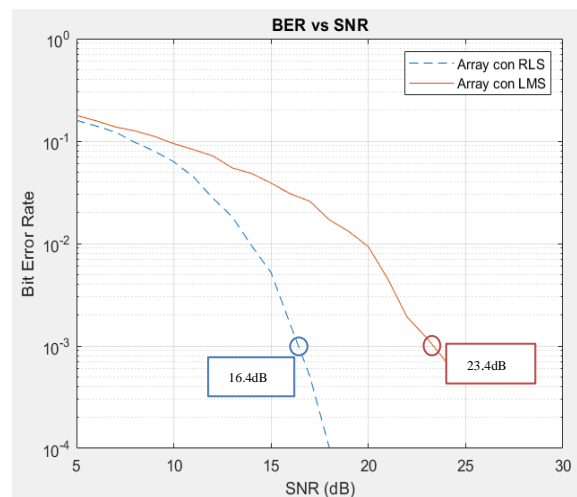


Fig. 10 Gráfica BER vs SNR para los algoritmos LMS (rojo) y RLS (azul) – Escenario 1

En la figura 9, se aprecia el diagrama de constelación de los símbolos transmitidos y recibos. Para el caso de los

B. Escenario 2

Este escenario se demuestra la lenta convergencia del algoritmo LMS frente a RLS, con los siguientes parámetros, mediante los parámetros que se muestran la tabla 2.

SNR	15
Ángulo de señales interferentes	10°, 70°, 20°, 0°
Ángulo Deseado (Rx)	-70°
Muestras	10000
Índice de modulación (Q-AM)	16
Bits Transmitidos	10000

Tabla. 2. Parámetros simulados – Escenario 2

En las figuras 11 y 12, se muestran los resultados del ángulo de arribo de la señal deseada después de incidir en el array de sensores, y pasar por el filtro adaptativo con los algoritmos RLS y LMS. Se observa que tanto el algoritmo LMS como RLS presentan valor de ganancia en su amplitud igual a 0.996269 apuntando a -69.8883° , es decir, ambos algoritmos generan un ángulo de apuntamiento que se aproxima al azimuth deseado con ganancia casi unitaria.

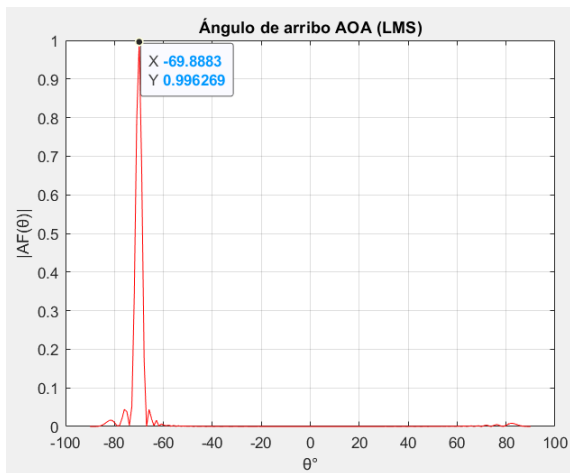


Fig. 11 Ángulo de arribo con algoritmo LMS – Escenario 1

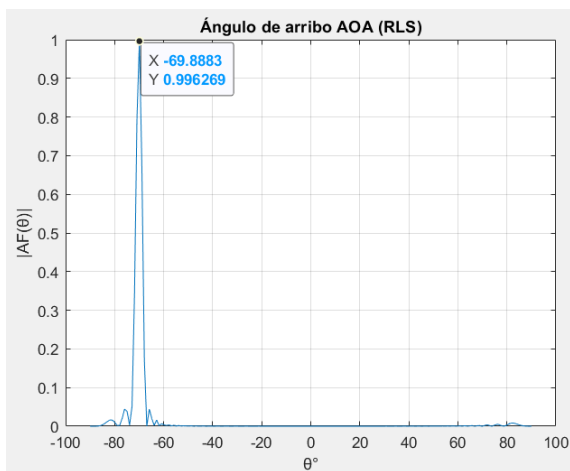


Fig. 12 Ángulo de arribo con algoritmo RLS – Escenario 1

En las figuras 13 y 14, se obtiene el patrón de radiación del array, el cual apunta a la dirección deseada con su ganancia expuesta en las figuras 11 y 12 respectivamente para cada algoritmo.

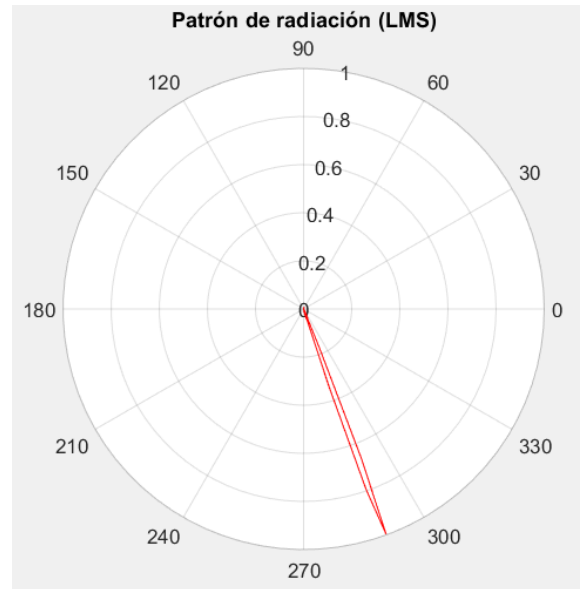


Fig. 13 Patrón de radiación con algoritmo LMS – Escenario 2

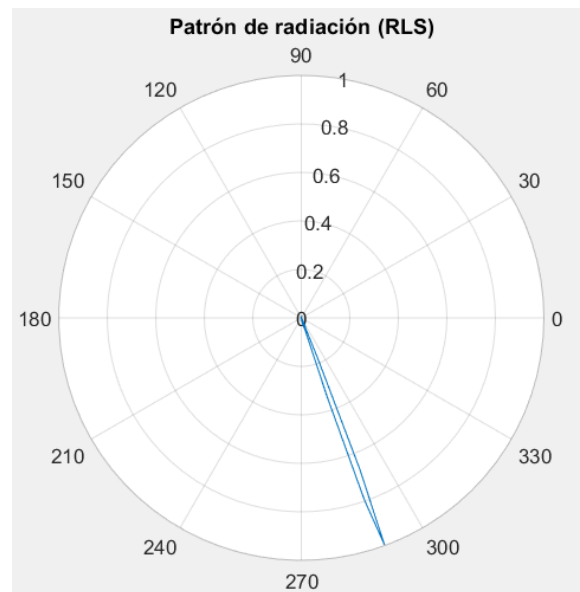


Fig. 14 Patrón de radiación con algoritmo RLS – Escenario 1

La figura 15 muestra el resultado del MSE para cada algoritmo simulado. Se puede observar que el algoritmo RLS requiere de un aproximado de 300 iteraciones con un valor promedio de $J(\mathbf{w}) = -150 \text{ dB}$ para llegar a su estado estacionario y oscilar entre los pesos óptimos encontrados, esto debido a que en el escenario 1 el algoritmo RLS converge y obtiene un ángulo de arribo con ganancia en la amplitud igual a 0.996268 que es muy cercano al que se visualiza en la figura 12, cabe mencionar que la única diferencia de los parámetros evaluados entre el escenario 1 y 2 es el número de iteraciones. El algoritmo LMS necesita aproximadamente 8000 iteraciones para llegar a la convergencia y formar un patrón de radiación con un tamaño de paso $\mu = 0.0001$, con el cual se define su velocidad y pasos de adaptación del algoritmo.

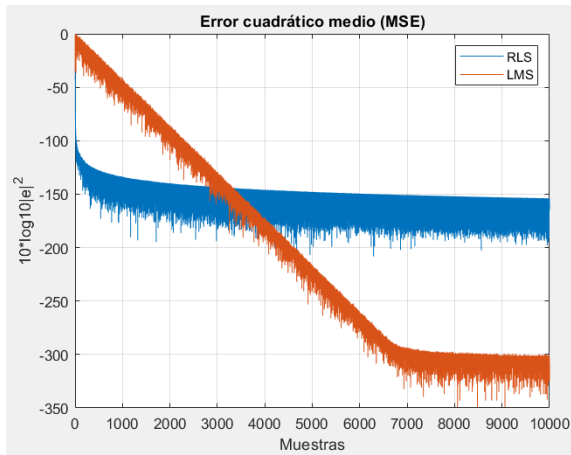


Fig. 15 Error cuadrático medio para los algoritmos LMS (rojo) y RLS (azul) – Escenario 2

En la figura 16, se muestran los tiempos que se demoraron cada algoritmo en filtrar las 10000 muestras de la señal, dando como resultado un tiempo mayor para el RLS (6.8730 segundos) y menor para el LMS (0.3687 segundos).

Además se visualizan los valores de bit error rate para la señal transmitida por el array. Con el filtro RLS, se obtuvieron 37 bits erróneos de 10000, y para el array con filtro LMS 30 bits erróneos, es decir, ambos algoritmos obtienen un BER similar con baja cantidad de bits erróneos

```

Command Window
Número de snap-shots del piloto... 10000
Ingrese el índice de modulación: 16
Ingrese el número de bits: 10000

tiempo_RLS =
    6.8730

tiempo_LMS =
    0.3687

RLS: El BER es 3.70e-03, basado en 37 errores.
LMS: El BER es 3.00e-03, basado en 30 errores.
fx >> |

```

Fig. 16 Resultados de tiempos de trabajo para los algoritmos LMS/RLS y BER – Escenario 2

En la figura 17, se aprecia el diagrama de constelación de los símbolos transmitidos y recibidos. Tanto para los símbolos transmitidos mediante el array con el algoritmo LMS y RLS se encuentran cercanos a sus respectivos puntos de constelación, corroborando los valores de BER de la figura 16.

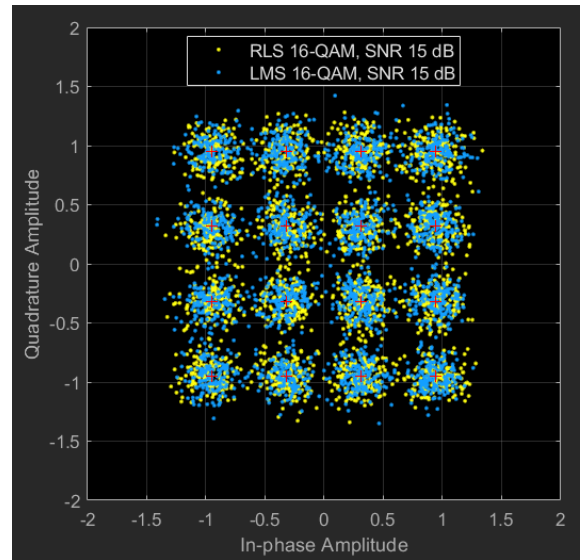


Fig. 17 Diagrama de constelaciones para los algoritmos LMS (azul) y RLS (amarillo) – Escenario 2

En la figura 18, se muestra la gráfica BER vs SNR. Se observa que para distintos valores de SNR se obtiene un valor de BER correspondiente. Tomando como referencia un $BER = 10^{-3}$ que es el mínimo recomendado en comunicaciones inalámbricas, se determina que el algoritmo RLS requiere un $SNR = 16.8 \text{ dB}$, mientras que, el algoritmo LMS requiere un $SNR = 16.24 \text{ dB}$, es decir, ambos algoritmos presentan valores de SNR similares, esto es debido a que el algoritmo LMS hizo uso de un mayor número de iteraciones para converger y obtener mayor ganancia en el ángulo de arriba como se observó en la figura 11, esta es una diferencia clara en comparación al escenario 1.

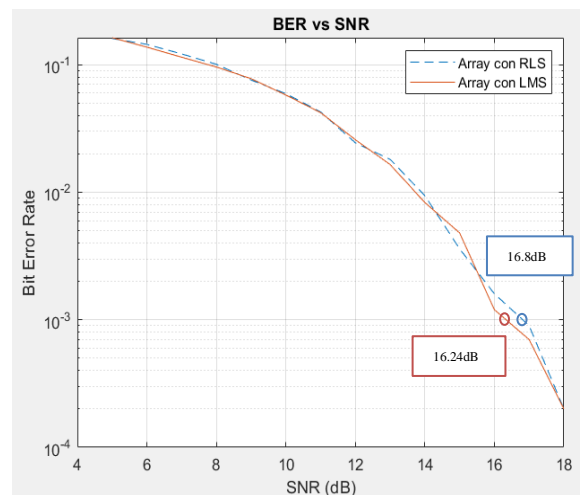


Fig. 18 Gráfica BER vs SNR para los algoritmos LMS (rojo) y RLS (azul) – Escenario 2

C. Escenario 3

En este escenario se pone a prueba a los algoritmos LMS y RLS, con ángulos de arriba de señales interferentes más cercanas a la señal deseada, los siguientes parámetros se simularon para el entorno en la tabla 3:

SNR	15
Ángulo de señales interferentes	-20°, 40°, 55°, 35°
Ángulo Deseado (Rx)	60°
Muestras	700
Índice de modulación (Q-AM)	16
Bits Transmitidos	10000

Tabla. 3. Parámetros simulados – Escenario 3

En las figuras 19 y 20, se muestran los resultados del ángulo de arribo de la señal deseada después de incidir en el array de sensores, y pasar por el filtro adaptativo con los algoritmos RLS y LMS. Se observa que el algoritmo RLS presenta un mejor valor de ganancia en su amplitud de 0.985831 apuntando a 59.8324° sin embargo, su ganancia disminuye 0.010437 en comparación al escenario 1 como resultado de atenuar las interferencias cercanas al ángulo deseado, por otro lado el algoritmo LMS obtiene una ganancia para el array de 0.922993 en un ángulo de 59.8324°, con este resultado se observa que al igual que en los escenarios anteriores el algoritmo LMS continúa dependiendo de un número de iteraciones elevadas, debido a que no se llega a encontrar los pesos óptimos para el array de sensores.

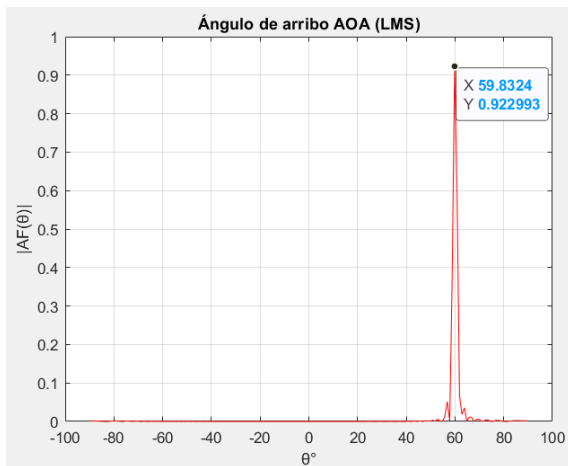


Fig. 19 Ángulo de arribo con algoritmo LMS – Escenario 3

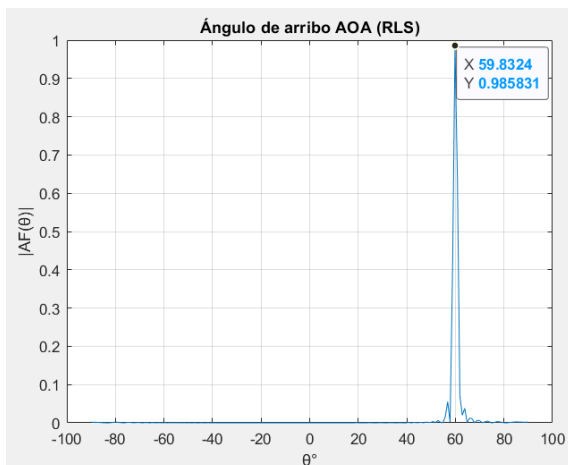


Fig. 20 Ángulo de arribo con algoritmo RLS – Escenario 3

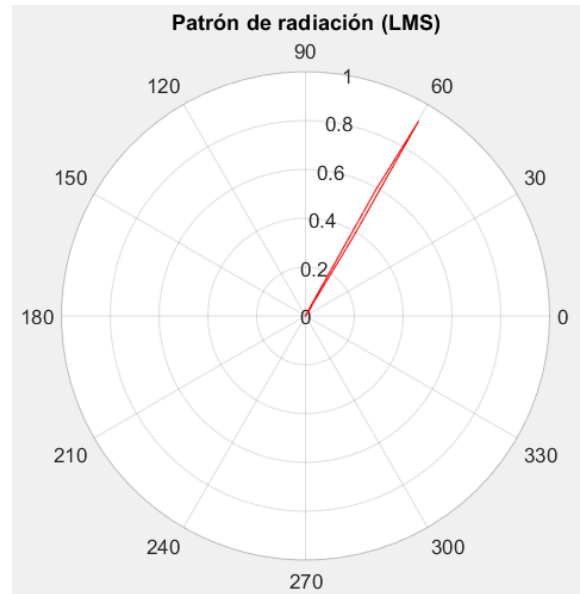


Fig. 21 Patrón de radiación con algoritmo LMS – Escenario 3

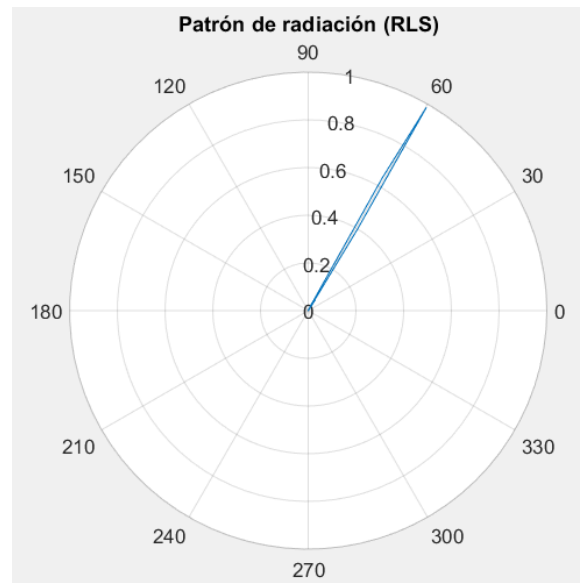


Fig. 22 Patrón de radiación con algoritmo LMS – Escenario 3

La figura 23 muestra el resultado del MSE para cada algoritmo simulado. Se puede observar que el algoritmo RLS sigue encontrando sus pesos óptimos para el array de antenas y es mucho más rápido que el algoritmo LMS al pesar de que existen interferencias más cercanas.

En las figuras 21 y 22, se obtiene el patrón de radiación del array, el cual apunta a la dirección deseada (60°), a diferencia de los escenarios 1 y 2, en los cuales se tiene un azimuth de -70°.

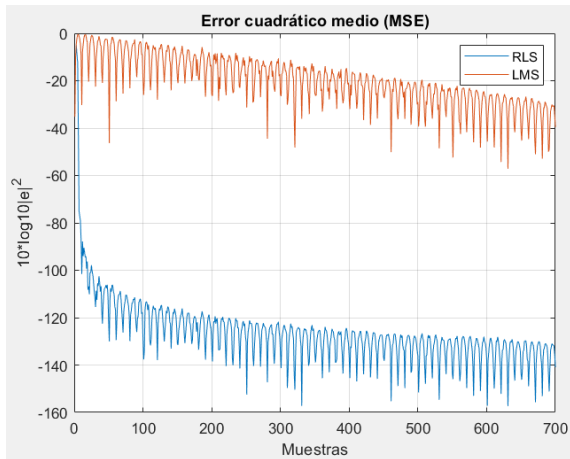


Fig. 23 Error cuadrático medio para los algoritmos LMS (rojo) y RLS (azul) – Escenario 3

En la figura 24, se muestran los tiempos que se demoró cada algoritmo en filtrar las 700 muestras de la señal, dando como resultado un tiempo mayor para el RLS al igual que los dos escenarios anteriores (0.588 segundos) y menor para el LMS (0.1728 segundos).

Además se visualizan los valores de BER para la señal transmitida por el array demostrando que para 700 iteraciones el algoritmo RLS tiene 8 errores menos que el algoritmo LMS.

```

Command Window
Número de snap-shots del piloto... 700
Ingrese el índice de modulación: 16
Ingrese el número de bits: 10000

tiempo_RLS =
    0.5880

tiempo_LMS =
    0.1728

RLS: El BER es 4.50e-03, basado en 45 errores.
LMS: El BER es 5.30e-03, basado en 53 errores.
fx >>

```

Fig. 24 Resultados de tiempos de trabajo para los algoritmos LMS/RLS y BER – Escenario 3

En la figura 25, se observa el diagrama de constelación para los símbolos transmitidos mediante el array de antenas para los algoritmos LMS y RLS respectivamente, a través de un canal AWGN con $SNR = 15$. Se observa que los símbolos se encuentran cercanos al punto de constelación, corroborando los valores de BER de la figura 24.

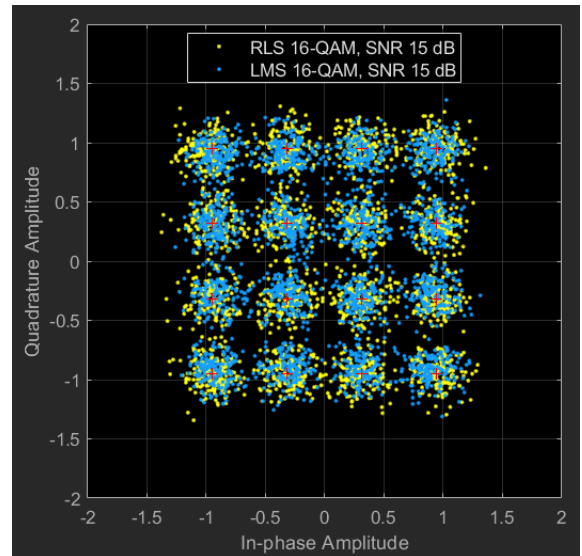


Fig. 25 Diagrama de constelaciones para los algoritmos LMS (azul) y RLS (amarillo) – Escenario 3

En la figura 26, se muestra la gráfica BER vs SNR. Se observa que para distintos valores de SNR se obtiene un valor de BER correspondiente. Tomando como referencia un $BER = 10^{-3}$ que es el mínimo recomendado en comunicaciones inalámbricas, se determina que el algoritmo RLS requiere un $SNR = 16.32$ dB, mientras que, el algoritmo LMS requiere un $SNR = 16.8$ dB. Al igual que en los escenarios 1 y 2, el algoritmo RLS presentó valores de SNR alrededor de los 16 dB, es decir, este algoritmo obtiene menor cantidad de bits erróneos en ambientes más ruidosos en comparación al algoritmo LMS, mediante una rápida convergencia.

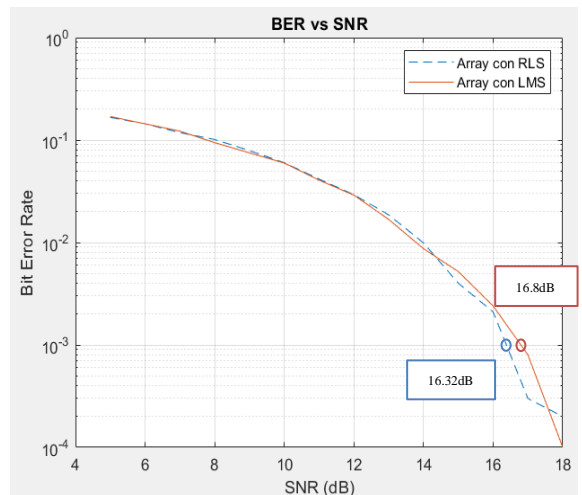


Fig. 26 Gráfica BER vs SNR para los algoritmos LMS (rojo) y RLS (azul) – Escenario 3

VI. CONCLUSIONES

En el presente artículo se realizó una comparación entre los algoritmos de generación de haces adaptativos LMS y RLS para MIMO masivo. Mediante la figura 15 se observó la curva de aprendizaje de cada uno de algoritmos y se muestra que el algoritmo LMS necesita aproximadamente 8000 iteraciones para converger, cabe mencionar que se hicieron uso de varios

valores de μ dentro del rango permitido en la ecuación 19 la misma que asegura la convergencia del algoritmo, siendo $\mu = 0.0001$ el que mejor ejemplifica el funcionamiento del algoritmo LMS para los escenarios 1, 2, 3 planteados, por lo tanto, el algoritmo LMS depende de su tasa de aprendizaje μ , sin embargo, no existe un valor específico para el tamaño de paso que optimice la velocidad de convergencia del algoritmo, es decir, si se tiene un valor del tamaño de paso demasiado pequeño, se obtiene una convergencia lenta, por lo contrario, si el tamaño de paso es demasiado grande se sobrepasarán los pesos óptimos de interés, por otro lado, el algoritmo RLS necesitó alrededor de 300 iteraciones para converger con un factor de olvido igual a 1 que considera todas las muestras de la matriz de datos, por consiguiente, para la conformación del haz del ángulo de arribo de la señal de interés de cada uno de los algoritmos se obtuvieron las figuras 11 y 12 con una ganancia en el array cercana a la unidad respectivamente.

Para terminar, de los resultados de BER expuestos en la figura 10 para el escenario 1 con una transmisión de 10000 bits se evidenció que el algoritmo RLS tuvo un mejor desempeño al obtener un $BER = 10^{-3}$ con una relación de señal a ruido $16.4dB$ y un patrón de radiación (figura 6) con ganancia cercana a la unidad, en comparación con el algoritmo LMS que obtuvo un $BER = 10^{-3}$ con $SNR = 23.4dB$ y patrón de radiación (figura 5) con ganancia aproximada de 0.6, además, de acuerdo a la figura 8 con un $SNR = 15dB$ se obtienen 368 bits erróneos más que el algoritmo RLS. En este mismo contexto para el escenario 2, mediante la figura 18, se determinó que el algoritmo RLS obtuvo un $BER = 10^{-3}$ con un valor de $SNR = 16.8dB$ y el algoritmo LMS un $BER = 10^{-3}$ con un valor de $SNR = 16.24dB$ dando como resultado una diferencia de 7 bit erróneos a favor del algoritmo LMS con un $SNR = 15dB$ como se muestra en la figura 16, es necesario resaltar que en este escenario se hizo uso de un mayor número de iteraciones para lograr que el algoritmo LMS converja. A partir de esto, se determina que el algoritmo más idóneo para una comunicación inalámbrica es el algoritmo RLS debido a su rápida convergencia y desempeño en la curva de BER.

VII. BIBLIOGRAFÍA

[1] G. L. Agredo Méndez, P. E. Jojoa Gómez, and V. Almenar Terré, "Sistemas MIMO con un Elevado Número de Antenas: Clave para la 5G Inalámbrica," *Entramado*, vol. 11, no. 2, pp. 1–12, 2015, doi: 10.18041/entramado.2015v11n2.22240.

[2] J. Leyva and B. David, "La Comunicación Inalámbrica a través de la Banda de los 60GHZ," *Rev. Científica Univ. y Soc.*, vol. 8, pp. 89–96, 2016.

[3] R. Chataut and R. Akl, "Massive MIMO Systems for 5G and Beyond Networks—Overview, Recent Trends, Challenges, and Future Research Direction," *Sensors (Switzerland)*, vol. 20, no. 10, pp. 1–35, 2020, doi: 10.3390/s20102753.

[4] I. Romero Ibáñez and M. Sánchez Fernández, "Ortogonalidad y Capacidad en Sistemas MIMO Masivo," pp. 1–80, 2014.

[5] N. Hassan and X. Fernando, "Massive MIMO

Wireless Networks: An Overview," *Electron.*, vol. 6, no. 3, pp. 1–29, 2017, doi: 10.3390/electronics6030063.

[6] E. Ali, M. Ismail, R. Nordin, and N. F. Abdulah, "Beamforming Techniques for Massive MIMO Systems in 5G: Overview, Classification, and Trends for Future Research," *Front. Inf. Technol. Electron. Eng.*, no. 6, pp. 1–20, 2017, doi: 10.1631/FITEE.1601817.

[7] A. Uncini, *Fundamentals of Adaptive Signal Processing: Estimation Theory*. 2015.

[8] J. Alvarez, M. Chuez, and P. Vargas, "Implementaciones en Matlab de los Algoritmos Adaptativos para los Sistemas de Antenas Inteligentes," pp. 1–8, 2011.

[9] S. Haykin, *Adaptive filter theory*. Simon Haykin, 5th ed. 2014.

[10] I. Stepanets, G. Fokin, and A. Müller, "Beamforming Techniques Performance Evaluation for 5G Massive MIMO Systems," in *CEUR Workshop Proceedings*, 2019, vol. 2348, pp. 1–13.

[11] P. A. Herrera Zapata and R. G. Rueda Álvarez, "Estudio Comparativo de los Algoritmos de Filtrado Adaptativo (LMS, NLMS, RLS) para la Eliminación del Ruido Audible Externo en Tiempo Real," escuela politécnica nacional, 2020.

[12] J. A. Thomas, P. R. Mini, and M. N. A. Kumar, "DOA Estimation and Adaptive Beamforming Using MATLAB and GUI," *2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput. ICECDS 2017*, pp. 1890–1896, 2018, doi: 10.1109/ICECDS.2017.8389778.

[13] H. Cuevas Esteban, "Propuesta de Aplicación de la Tecnología MIMO Masivo a las Comunicaciones Móviles 5G," pp. 1–66, 2017, [Online]. Available: <http://oa.upm.es/44669/>.

[14] K. A. Aguilar Cruz, "Filtro Digital para la Estimación de Parámetros Variantes Usando el Factor de Olvido Exponencial," 2016.

[15] M. Á. Catillo Marcelllo, "Tratamiento de la Señal de Voz Usando Filtros Adaptativos en Subbandas para Ambientes Altamente Ruidosos," 2014.