



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELÉCTRICA Y
TELECOMUNICACIONES**

**TRABAJO DE TITULACIÓN, PREVIO A LA
OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y TELECOMUNICACIONES**

**Evaluación de desempeño para un canal no lineal en un
medio de comunicación inalámbrico**

AUTOR: Quijano Toaquiza, Rodrigo Fabian

**DIRECTOR: ING. Bernal Oñate, Carlos Paúl
SANGOLQUÍ**

2018

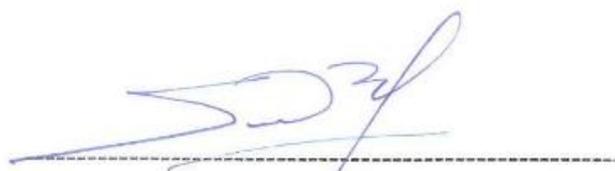


DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que el trabajo de titulación, “EVALUACIÓN DE DESEMPEÑO PARA UN CANAL NO LINEAL EN UN MEDIO DE COMUNICACIÓN INALÁMBRICO” realizado por el señor RODRIGO FABIAN QUIJANO TOAQUIZA, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor RODRIGO FABIAN QUIJANO TOAQUIZA para que lo sustente públicamente.

Sangolquí, 08 de enero del 2018



ING. CARLOS PAÚL BERNAL OÑATE
DIRECTOR



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORÍA DE RESPONSABILIDAD

Yo, RODRIGO FABIAN QUIJANO TOAQUIZA, con cédula de identidad N° 1722651815, declaro que este trabajo de titulación "EVALUACIÓN DE DESEMPEÑO PARA UN CANAL NO LINEAL EN UN MEDIO DE COMUNICACIÓN INALÁMBRICO" ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 08 de enero del 2018



RODRIGO FABIAN QUIJANO TOAQUIZA

C.C. 1722651815



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, RODRIGO FABIAN QUIJANO TOAQUIZA, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación "EVALUACIÓN DE DESEMPEÑO PARA UN CANAL NO LINEAL EN UN MEDIO DE COMUNICACIÓN INALÁMBRICO" cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 08 de enero del 2018



RODRIGO FABIAN QUIJANO TOAQUIZA
C.C. 1722651815

DEDICATORIA

Este trabajo lo dedico a la memoria de mi abuelita Esperanza Cabazos, en memoria de todos sus consejos y cuidados, es doloroso saber que no estará físicamente presente ese día tan especial pero sé que ella jamás morirá mientras este en mi corazón y mi pensamiento.

A ella le debo mucho de lo que soy y por enseñarme que la vida puede ser muy difícil pero eso lo aprendí sin ella.

Dedico también a mi hija Melanie Quijano ya que ella fue el motivo más grande que tuve para superarme y dejar muchas etapas atrás, te amo con todo mi corazón y espero que cuando seas más grandecita y puedas leer sepas que tu papi no tuvo una vida perfecta pero sin embargo termino lo que comenzó y este peso de superarme te lo dejo para toda tu vida, espero siempre estar a tu lado y sino espero estar en una dedicatoria como hoy está mi mamá

Mis padres sobre todo por jamás dejarme solo pese a muchos errores que uno comete a pasar del tiempo y por enseñarme a sobresalir y llevar una vida honesta.

Les amo mucho padres y siempre serán los mejores me enseñaron mucho de la vida y siempre estaré orgulloso de ustedes, gracias por todos los regaños que fueron necesarios por todas sus decisiones que buenas o no tan buenas me hicieron un hombre de responsabilidades y que jamás se rinde ante nada.

AGRADECIMIENTO

Agradezco de manera especial a mi amigo que hoy por hoy es mi familia, gracias Inge Paúl Bernal realmente nada hubiera podido hacer sin tus consejos y enseñanzas, estuviste cuando falleció mi mamá y jamás dejaste que me vaya al abandono solo puede decirte que cuando necesites ahí estaré siempre para ti, sin más mi querido amigo espero siempre estés aunque me hables pero sé que es por mi bien.

No puedo olvidar también a todos mis amigos que sin ellos tal vez no lo hubiese logrado, amigos míos Christian, Vlady, Tony, Julio, Andrés, y a todos quienes me dieron una mano cuando necesite solo puedo decirles que gracias y que ahí estaré para cuando necesiten también.

Mis profesores, jamás olvidare cada día de enseñanza pero sobre todo cada día que compartimos como amigos, los libros están escritos y se pueden leer cuando sea, pero las verdaderas amistades no son tan comunes de encontrar, les agradezco por todo.

Mi familia en general muchas gracias por el apoyo y la confianza, en especial a mi madrina Clemita que es como mi madre y sus hijos Paul y Julia les quiero mucho y solo puede decir por fin lo logre.

En mi mundo laboral hoy por hoy agradezco a todos mis jefes, compañeros y sobre todo amigos por ayudarme a crecer profesionalmente, pienso que aprendí muchísimo y me siento muy preparado para poder competir como profesional.

Esta última parte va para todos las personas que han estado en el camino junto a mí, se les quiso mucho y en verdad gracias por cada granito de arena les deseo éxitos en sus vidas.

ÍNDICE DE CONTENIDOS

RESUMEN.....	x
ABSTRACT	xi
CAPÍTULO 1	1
1. INTRODUCCIÓN	1
1.1. Antecedentes.	1
1.2. Justificación e Importancia.	2
1.3. Alcance del Proyecto.....	3
1.4. Objetivos	3
CAPÍTULO 2	5
2. MARCO TEÓRICO	5
2.1. Redes inalámbricas.....	5
2.1. Codificación de canal	7
2.2. Distorsión no lineal en diferentes estándares inalámbricos	11
2.3. Volterra	11
2.4. Análisis de banda base complejo para sistemas no lineales con memoria.....	17
2.5. Modelo de Volterra en frecuencia simple no lineal	18
2.6. Algoritmo de Volterra LMS	20
2.7. Algoritmo de Volterra NLMS	22
2.8. Algoritmo de Volterra RLS.....	23
2.9. Sistemas no lineales	25
CAPÍTULO 3	26
3. MATERIALES Y METODOS	26
3.1. Proceso de transmisión.....	26
CAPÍTULO 4	33

4. PRUEBAS Y RESULTADOS	33
4.1. Diagrama de bloques	33
4.2. Demodulación	34
4.3. Resultados del BER	39
4.4. BER de Volterra LMS.....	39
4.5. BER de Volterra RLS.....	40
CAPÍTULO 5	41
5. CONCLUSIONES Y RECOMENDACIONES.....	41
5.1. Conclusiones	41
5.2. Recomendaciones.....	42
5.3. Trabajos Futuros	42
Bibliografía	43

ÍNDICE DE TABLAS

Tabla 1. Características de los estándares IEEE 802.11	6
Tabla 2. Algoritmo LMS.....	21
Tabla 3. Algoritmo NLMS	22
Tabla 4. Variación de parámetros	30
Tabla 5. Variación de Coeficiente.....	31
Tabla 6. Variación de λ	32
Tabla 7. Variación de SNR de V-NLMS	39
Tabla 8. Variación de SNR de V-LMS	39
Tabla 9. Variación de SNR de V-RLS	40

ÍNDICE DE FIGURAS

Figura 1. Diagrama de convolución	8
Figura 2. Sistema de Volterra orden 3 con 2 retrasos	14
Figura 3. Descomposición del filtro de Volterra en bloques	15
Figura 4. (a) Señal original; (b) Zoom señal original intervalo (0-100) muestras. ...	26
Figura 5. (a) Señal del convolucional; (b) Zoom señal convolucional intervalo (0-100) muestras.....	27
Figura 6. Señal modulada 64-QAM	28
Figura 7. (a) Señal después del canal no lineal Rapp; (b) Zoom señal después del canal no lineal Rapp intervalo (0-100) muestras.....	29
Figura 8. (a) Curva de V-NLMS; (b) Zoom curva de V-NLMS intervalo (0-300) muestras.....	30
Figura 9. (a) Curva de V-LMS; (b) Zoom curva de V-LMS intervalo (0-300) muestras.....	31
Figura 10. (a) Curva de V-RLS; (b) Zoom curva de V-RLS intervalo (0-300) muestras.....	32
Figura 11. Diagrama de bloques de la señal transmitida	33
Figura 12. Diagrama de bloques del proceso de recepción.....	34
Figura 13. Señal Rx con V-NLMS (SNR=10 dB, BER= 9,05 %).....	34
Figura 14. Señal Rx con V-NLMS (SNR=20 dB, BER= 0,49 %).....	35
Figura 15. Señal Rx con V-NLMS (SNR=30 dB, BER= 0.075785 %).....	35
Figura 16. Señal Rx con V-LMS (SNR=10 dB, BER= 8,75 %).....	36
Figura 17. Señal Rx con V-LMS (SNR=20 dB, BER= 0,5 %).....	36
Figura 18. Señal Rx con V-LMS (SNR=30 dB, BER= 0.072734%).....	37
Figura 19. Señal Rx con V-RLS (SNR=10 dB, BER= 7.87%).....	37
Figura 20. Señal Rx con V-RLS (SNR=20 dB, BER= 0.000066121%).....	38
Figura 21 Señal Rx con V-RLS (SNR=30 dB, BER= 0%).....	38

RESUMEN

La utilización frecuente de las series de Volterra en numerosos campos de la ingeniería, suma especial importancia a los estudios sobre este tema, por su vital aporte a soluciones de transmisiones inalámbricas. El principal inconveniente de esta herramienta es su complejidad paramétrica, debida al elevado número de coeficientes en sus núcleos. En este proyecto se realizó una medición de rendimiento de canal, para evaluar el desempeño de una transmisión inalámbrica que aplica como base a las series antes mencionadas. Como primer paso, se generó una señal con datos binarios aleatorios, que fue transmitida utilizando el estándar IEEE 802.11, basándose en una investigación previa donde se definieron los parámetros de transmisión: OFDM y un codificador convolucional. Estos datos fueron modulados y enviados por un canal Rapp, para generar el efecto de distorsión no lineal y adicionar el ruido. Adicionalmente, se elaboró un código en Matlab[®], tomando como modelo a un canal no lineal, en el que se pudo utilizar las series complejas de Volterra para extraer las características de su polinomio y generar un compensador que funcione con cualquier señal aleatoria a transmitir. Posteriormente, tras el estudio y análisis del funcionamiento de este algoritmo en los sistemas de comunicaciones inalámbricas, se realizó la corrección de la distorsión no lineal, misma que generó componentes de frecuencias nuevas, que no formaban parte de la señal original transmitida. Finalmente, se efectuó la demodulación de la señal recibida para realizar un análisis de errores cuadráticos medios (MSE) y un análisis gráfico de dBm, con el fin de medir el desempeño del canal ante las soluciones propuestas, como fueron los algoritmos LMS, NLMS y RLS, de esta manera se pudo decidir cuál de los tres brinda un mejor valor a la tasa de error por bit (BER).

Palabras claves:

- ERROR CUADRATICO MEDIO
- SERIES COMPLEJAS DE VOLTERRA
- RLS, NLMS, LMS
- CANAL NO LINEAL
- BER

ABSTRACT

The frequent use of Volterra series in various fields of engineering, adds special importance to studies on this subject, for its vital contribution to wireless transmission solutions. The main drawback of this tool is its parametric complexity, due to the high number of coefficients in its core. In this project, it was made a measurement of channel performance to evaluate the yield of a wireless transmission that applies the previously mentioned series. As a first step, a signal with random binary data was generated, which was transmitted using the IEEE 802.11 standard, based on a previous investigation where the transmission parameters were defined: OFDM and a convolutional encoder. These data was modulated and sent through a Rapp channel to generate the non-linear distortion effect and to add the noise. Additionally, a Matlab[®] code was programmed, taking a non-linear channel as a model, in which the Volterra series can be used to extract the characteristics of its polynomial and generate a compensator that works with any random transmitted signal. Subsequently, after the study and analysis of the operation of this algorithm in the wireless communications systems, it was made the correction of the non-linear distortion, which generated new components that were not part of the original transmitted signal. Finally, the signal was demodulated to perform an analysis of mean square errors (MSE) and a graphical analysis of dBm, in order to measure the performance of the channel with the proposed solutions, such as the LMS, NLMS and RLS algorithms. Through this way, it was possible to decide which of the three solutions offers a better value for the error bit rate (BER).

Key words:

- MEAN SQUARE ERROR
- VOLTERRA COMPLEX SERIES
- RLS, NLMS, LMS
- NON LINEAR CHANNEL
- BER

CAPÍTULO 1

1. INTRODUCCIÓN

1.1. Antecedentes.

Los modelos o series de Volterra, son aquellos modelos polinucleares no recursivos que pueden ser utilizados para representar sistemas no lineales con memoria que se desvanece. Estos modelos han sido objeto de numerosas aplicaciones en diferentes campos, como por ejemplo la anulación del eco y del ruido, la linealización de altavoces, el modelado, la igualdad y la pre-distorsión de canales de comunicación lineales, el modelado de sistemas fisiológicos, o también el comando de sistemas no lineales (Crespo, 2015).

La serie compleja Volterra (CVS) según (Crespo, 2015) , es el modelo más usado para los sistemas no lineales. Este modelo es útil para filtrajes adaptativos debido a que su formulación es básica, y los filtros adaptativos fácilmente se pueden ampliar en base a este modelo además es una de las representaciones más populares para un modelo de sistemas de comunicación inalámbrica moderno.

Por lo tanto, en este trabajo se utilizara el software Matlab[®] para la simulación del algoritmo donde primero se introducirá un paquete de datos el cual será modulado para ser introducido en un canal no lineal el que será generado por código utilizando características propias para posteriormente utilizar las Series complejas de Volterra con los diferentes algoritmos que son RLS (Recursive-Least-Squares), LMS (Least-Mean-Square algorithm), NLMS (Normalized least mean square), para corregir la distorsión no lineal. Para finalmente evaluar el desempeño del canal utilizando la tasa de error de bit (BER) y también utilizando el error cuadrático medio (MSE), para así poder realizar la comparación entre los algoritmos y ver su comportamiento.

1.2. Justificación e Importancia.

Las comunicaciones inalámbricas han venido evolucionando al paso de los años permitiendo tener una comunicación clara y estable para los usuarios que hacen uso de esta. En los últimos años las comunicaciones inalámbricas son las más utilizadas por su funcionalidad ya que permiten movilidad y portabilidad sin tener un medio cableado.

Sin embargo, ahora han crecido tanto que al utilizar al mismo tiempo un gran número de usuarios llegan a saturarse por lo que se han desarrollado técnicas para combatir este problema. En comunicaciones por satélites, amplificadores de alta potencia a bordo de satélites y en estaciones terrestres trabajan en condiciones de saturación, las cuales introduce distorsión no lineal en las señales transmitidas. En redes de comunicaciones inalámbricas usando ondas de radio y fibras ópticas, convertidores electro-ópticos, fuentes no lineales. Los modelos de Volterra en banda base son muy útiles para el modelado de canales de comunicaciones no lineales.

El principal inconveniente de estos modelos es su complejidad paramétrica la cual se incrementa rápidamente con el orden de la no linealidad de los sistemas y la memoria de los núcleos. Según (Schetzen, 1980) la CVS es un modelo para el comportamiento no lineal similar a la serie de Taylor, pero se diferencia en su capacidad para captar los efectos de la "memoria". En la serie compleja de Volterra la salida del sistema no lineal depende de la entrada al sistema en todos los otros momentos. Esto proporciona la capacidad de capturar el efecto "memoria" de dispositivos como condensadores e inductores.

En la ingeniería eléctrica la serie compleja de Volterra se utiliza para modelar una intermodulación con distorsión en muchos dispositivos, incluyendo amplificadores de potencia y mezclador de frecuencias. Su principal ventaja es su generalidad ya que puede representar una amplia gama de sistemas. Por lo tanto, es a veces considerado un modelo no paramétrico. El procedimiento habitual para la extracción de los coeficientes de dichos modelos consiste en la adquisición de muestras en el dominio del tiempo de las formas de onda a la entrada y a la salida del sistema.

1.3. Alcance del Proyecto

Comparar la aplicación en el canal no lineal de las Series complejas de Volterra (CVS) tanto en BER como en error cuadrático medio (MSE) para una identificación de parámetros y una fácil demostración de la solución ante el canal.

Realizar una simulación de un canal no lineal con la ayuda de Matlab® para poder introducir la señal original para posteriormente podrá medir el desempeño al aplicar las CVS y los algoritmos RLS, LMS, NLMS.

Demostrar mediante simulación que las series complejas de Volterra (CVS) es una solución ante problemas generados en la transmisión de sistemas digitales que usan amplificadores de potencia y conversores análogo-digital los cuales serán simulados en Matlab®.

Investigar y utilizar todos los parámetros para poder hacer una simulación de transmisión del estándar IEEE 802.11 y ser mejorada con las series de Volterra y los algoritmos.

1.4. Objetivos

1.4.1. General

Analizar el desempeño de un canal de comunicación inalámbrica utilizando el modelo no lineal de Volterra.

1.4.2. Específicos

- Identificar los parámetros del modelo de series complejas de Volterra (CVS) y su forma de extraer características mediante su polinomio.
- Generar una señal para que sea transmitida por un canal no lineal simulado por parámetros tomados de un canal Rapp.

- Realizar una comparación entre la utilización de las series complejas de Volterra (CVS) con cada uno de los algoritmos como RLS, LMS y NMLS.
- Evaluar el desempeño del cañal mediante el software Matlab® en el cual se introducirá la señal y posteriormente será demodulada y analizada con la ayuda de la tasa de error de bit, BER.
- Variar parámetros para disminuir la cantidad de bits erróneos para que sea transmitida con los parámetros del estándar IEEE 802.11.

CAPÍTULO 2

2. MARCO TEÓRICO

2.1. Redes inalámbricas

Las redes inalámbricas permiten la interconexión entre dos o más puntos, nodos o estaciones, por medio de ondas electromagnéticas que viajan a través del espacio llevando información de un lugar a otro. Para lograr el intercambio de información existen diferentes mecanismos de comunicación o protocolos que establecen reglas que permiten el flujo confiable de información entre nodos. Por ejemplo, el conjunto de protocolos TCP/IP utilizado en redes de computadoras como Internet, permite que cualquier computadora que los use pueda comunicarse con otra que se encuentre conectada a la misma red. Los estándares son una serie de normas que definen la forma en que se deben realizar ciertos procesos para garantizar la calidad y seguridad de su funcionamiento, sin importar el tipo de dispositivo o las diferencias en su construcción.

Los estándares facilitan además la interoperabilidad entre componentes, aunque estos tengan características diferentes. Existen diferentes organismos internacionales que originan estándares. En el área de telecomunicaciones se encuentran, por ejemplo, el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por su sigla en inglés) y la Unión Internacional de Telecomunicaciones (UIT) (Alcocer, 2016).

La gran mayoría de las redes inalámbricas comunitarias utilizan tecnología Wi-Fi, esta tecnología está definida en la familia de estándares inalámbricos 802.11 del IEEE, siendo las más representativas las enmiendas 802.11a, 802.11b, 802.11g, 802.11e y la 802.11n.

Esta familia de estándares trabaja en las bandas de radiofrecuencia de 2.4GHz y 5GHz asignadas por la UIT para redes inalámbricas como uso secundario. En esta banda trabajan también los hornos de microondas domésticos y muchos otros dispositivos industriales que no requieren de licencia por parte de los entes reguladores del espectro en cada país; de aquí se deduce que el uso primario de estas frecuencias es para aplicaciones distintas a las de comunicación. Sin embargo, estas frecuencias

también han sido utilizadas para diferentes servicios de comunicaciones como teléfonos inalámbricos, intercomunicadores, transmisión de video y audio a corta distancia, etc. Por este motivo se debe tomar en cuenta los otros usuarios de esta porción del espectro en un espacio físico determinado y establecer límites en los niveles de potencia máximos utilizables, para evitar interferencias con otras redes. La enmienda 802.11g usa la banda de frecuencias de 2.4GHz mientras que la 802.11a usa la banda de frecuencia de 5GHz. Esto hace que las redes que utilizan una u otra tecnología no sean interoperables a pesar de que utilizan las mismas técnicas de modulación y por ende la misma tasa máxima de transmisión de 54Mbps. La enmienda 802.11b tiene una tasa de transferencia de 11 Mbps y opera también en la banda de 2.4GHz por lo cual es compatible con la 802.11g, que hace posible la interoperabilidad de dispositivos y redes que los utilicen, a continuación, se puede observar en la Tabla 1 algunas características de los estándares de IEEE 802.11.

Tabla 1.
Características de los estándares IEEE 802.11

Standard	Banda de Frecuencia	Ancho de banda	modulación	Tasa máxima	Rango alcance
802.11	2.4 GHz	20 MHz	BPSK a 256-QAM	2 Mbps	20 m
b	2.4 GHz	20 MHz	CCK, DSSS	11 Mbps	35 m
a	5 GHz	20 MHz	OFDM	54 Mbps	35 m
g	2.4 GHz	20 MHz	DSSS,OFDM	54 Mbps	70 m
n	2.4 y 5 GHz	20 MHz	OFDM	600 Mbps	70 m
ac	5 GHz	20,40 MHz	OFDM	6.93 Gbps	35 m

Fuente: (Alcocer, 2016)

2.1. Codificación de canal

Cuando transmitimos información, uno de los objetivos principales es el de minimizar la cantidad de errores que pudieran producirse en el proceso. Esta transmisión depende del factor Señal a Ruido (S/N), entre otros parámetros.

Si optimizando estas variables se necesita aún mejorar la calidad de la transmisión, entonces se deben buscar ciertos métodos que aseguren y mejoren la habilidad. Es a partir de aquí que surge el concepto de la codificación para control de errores según (Mezoa, 2011).

La codificación de canal para el control de errores se encarga, básicamente, de la adición de bits extra al mensaje a transmitir. Ellos no poseen información como tal, pero hacen posible la detección y corrección de errores en el bloque de recepción del mensaje.

En un sistema de comunicaciones, detectar un error es más sencillo que corregirlo. Si existen irregularidades, el receptor puede pedir una retransmisión del mensaje que contiene el error (ARQ: *Automatic Repeat Request*). Sin embargo, cuando el sistema no implementa esta técnica por no ser práctico o simplemente porque no es posible, debe aplicar redundancia en el código a través del método FEC (*Forward Error Correction*). El método más sencillo de redundancia en el código consiste en repetir n veces el símbolo de mensaje según (Mezoa, 2011).

2.1.1. Códigos Convolutivos

A diferencia de la codificación por bloque, los códigos convolutivos trabajan bit por bit. Su estructura pudiera generalizarse como se muestra en la Figura 1:

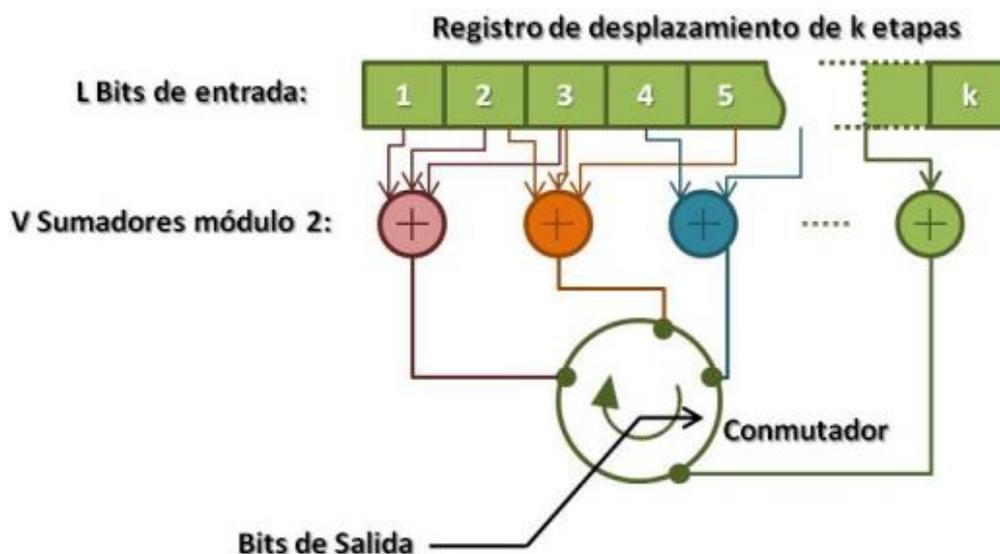


Figura 1. Diagrama de convolución

Fuente: (Mezoa, 2011)

Esta estructura genera V símbolos a la salida por cada símbolo de entrada. Su tasa de símbolos de salida será entonces $R_c = \frac{1}{V}$.

Los códigos convolucionales se utilizan para proteger la información añadiendo redundancia a la misma, de manera que las palabras del código tengan la distancia mínima necesaria.

Sin embargo, a diferencia de los códigos bloque, las palabras de un código convolucional se generan no sólo a partir de los dígitos de información actuales sino también con la información anterior en el tiempo. Es decir, un codificador convolucional es un sistema con memoria y en consecuencia, lleva asociada una cadena de Markov aunque ésta no es visible en la salida pero sí la condiciona. (Mezoa, 2011)

2.1.1.1. Codificación

Existen varios métodos de codificación de códigos convolucionales, aunque una de las más usuales es la basada en registros de desplazamiento conectados con sumadores

base 2 en los que se realiza la codificación. Por cada bit que entre en el codificador se obtienen n bits.

2.1.1.2. Decodificación

La decodificación de un código convolucional consiste en escoger la secuencia más probable entre todas las posibles. Existen diversos algoritmos que permiten la decodificación de este tipo de códigos y la decodificación óptima se consigue mediante el algoritmo de Viterbi.

2.1.2. Algoritmo Viterbi

Se considera un algoritmo de máxima similitud (*Maximum likelihood*) y altamente paralelizable, y consiste en determinar el camino más óptimo expresado como la elección del *codeword* (palabra código) con la métrica de máxima certeza o la elección del *codeword* con la métrica de mínima distancia, este algoritmo toma ventaja de la estructura espacial del diagrama de trellis¹.

Para hallar el camino más óptimo compara en cada tiempo t_i la señal recibida o *codewords* con todos los caminos del trellis que entran a cada estado en el tiempo t_i y elimina los caminos que no son posibles candidatos para la escogencia de la máxima certeza; la comparación de la señal recibida frente a la establecida del trellis se hace por medio del cálculo de la distancia Hamming (Número de posiciones en las cuales difieren dos *codewords*) para decisiones duras y para decisiones suaves la distancia euclidiana la cual se define a continuación. Los vectores de los *codewords* para decisiones suaves se representan por números reales, entonces la diferencia entre estos *codewords* se determina por la diferencia entre los elementos de cada *codeword* al cuadrado.

Los vectores de los *codewords* para decisiones suaves se representan por números reales, entonces la diferencia entre estos *codewords* se determina por la diferencia

¹ Es un diagrama en forma de red, cada línea horizontal se corresponde con uno de los estados del codificador. Cada línea vertical se correspondería con uno de los niveles del árbol del código.

entre los elementos de cada *codeword* al cuadrado. *Codewords* $X = [X_0, X_1, X_2, X_3, \dots, X_L]$, $Y = [Y_0, Y_1, Y_2, Y_3, \dots, Y_L]$ entonces la distancia euclidiana está dada por:

$$d_{xy} = \sqrt{(X_0 - Y_0)^2 + (X_1 - Y_1)^2 + \dots + (X_L - Y_L)^2} \quad (2.1)$$

A la distancia Hamming o euclidiana se le llama métrica de rama (λ).

Para dos caminos que entran al mismo estado en el tiempo t_i se escoge el de mejor métrica de distancia acumulada, la cual es calculada por medio de la métrica del estado en el tiempo t_{i-1} más la métrica de rama (λ), la métrica acumulada o de estado se expresa como:

$$\mu_{\mu} = \mu_{\mu-1} + \lambda \quad (2.2)$$

Después de tener las dos métricas de distancia acumulada del camino j y k , se escoge la métrica de menor distancia o peso para el caso de decisión dura $\mu_{ti} = \min[\mu_{jti}, \mu_{kti}]$, y para el caso de decisión suave se escoge la métrica de mayor peso $\mu_{ti} = \max[\mu_{jti}, \mu_{kti}]$; al camino elegido se le llama camino sobreviviente $\mu_{ti} m_k$ y a la métrica del camino ganador se le llama métrica de estado en el tiempo t_i y esto se hace para todos los estados, el decodificador continua así avanzando por el trellis eliminando y escogiendo el camino más probable.

Para determinar la probabilidad de error para un canal binario simétrico (BSC) y un canal AWGN (Aditivo Ruido Gaussiano Blanco) trabajando con el algoritmo de Viterbi se debe hacer dos suposiciones, la primera es que todos los posibles *codewords* son igualmente probables y la segunda es que la probabilidad de error debe ser independiente de la secuencia transmitida, de la segunda suposición se deriva que se debe trabajar con canales sin memoria, en los cuales el ruido afecta a los símbolos de manera independiente con respecto a los símbolos anteriores (Andrew, 1998).

2.2. Distorsión no lineal en diferentes estándares inalámbricos

El rendimiento para los sistemas de comunicaciones inalámbricos es usualmente caracterizado por el desempeño el cual se mide por el BER (la tasa de error de bit), que el parámetro fundamental para determinar la calidad de una señal modulada. La degradación en un sistema es debido a un número de factores como son ruido del receptor, ruido de fase, distorsión no lineal causada por los transmisores y receptores no lineales, canales no permitidos, interferencia, espurios y algunos otros factores.

La degradación del sistema puede ser referido a la degradación de la relación señal-ruido (SNR) del receptor, cual es relacionado al BER debido a su relación de dependencia al tipo de modulación, acceso múltiple, etc. SNR es una medida, como la relación de potencia de la señal útil a la potencia de Ruido Aditivo Gaussiano Blanco (AWGN).

2.3. Volterra

2.3.1. Series de Volterra

Considerando $x(n)$ y $y(n)$ como las señales de entrada y de salida, respectivamente, de un sistema no lineal, causal e discreto en el tiempo. Se puede expresar la Serie de Volterra usando $x(n)$ y dada por (Schetzen, 1980)

$$y(n) = h_0 + \sum_{m_1=0}^{\infty} h_1(m_1)x(n - m_1) \quad (2.3)$$

Desarrollando la sumatoria tenemos:

$$\begin{aligned} & \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} h_2(m_1, m_2)x(n - m_1)x(n - m_2) + \\ & + \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} \dots \dots \sum_{m_p=0}^{\infty} h_p(m_1, m_2, \dots, m_p)x(n - m_1)x(n - m_2) \dots \dots x(n \\ & \quad \quad \quad - m_p) + \dots \dots \end{aligned} \quad (2.4)$$

Donde, $h_p(m_1, m_2, \dots, m_p)$ es conocida como p-enésima orden de núcleo del sistema de Volterra (Schetzen, 1980). Asumiendo que los términos de la expresión enésima son simétricos, nos queda que $h_p(m_1, m_2, \dots, m_p)$, estos términos no sufren alteraciones ante perturbaciones con los índices de m_1, m_2, \dots, m_p .

Se observa que la serie infinita presentada en la ecuación 2.4 no es adecuada para utilizarla en filtraje adaptativo, se debe trabajar con sus extensiones truncadas. Realizando un truncamiento y asumiendo que $h_0 = 0$, a la ecuación 2.5 dada por:

$$\begin{aligned}
 y(n) &= h_0 + \sum_{m_1=0}^{N-1} h_1(m_1)x(n - m_1) \\
 &+ \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} h_2(m_1, m_2)x(n - m_1)x(n - m_2) \\
 &+ \dots + \sum_{m_1=0}^{N-1} \dots \sum_{m_p=0}^{N-1} h_p(m_1, m_2, \dots, m_p)x(n \\
 &- m_1) \dots x(n - m_p),
 \end{aligned} \tag{2.5}$$

Donde N-1 representa la cantidad de elementos de retardo considerada en la serie de Volterra truncada. En la Figura 2 representa un diagrama de la serie de Volterra truncada de tercer orden igual a 3 con 2 elementos de retardo. En este caso es posible modelar un núcleo de Volterra de tercer orden que permita expresar una señal de entrada de un filtro adaptativo para el siguiente vector (Diniz, 2002).

$$\mathbf{x}(n) = \begin{bmatrix}
 x(n) \\
 x(n-1) \\
 \vdots \\
 \vdots \\
 x(n-N+1) \\
 x^2(n) \\
 x(n)x(n-1) \\
 \vdots \\
 \vdots \\
 x(n)x(n-N+1) \\
 x^2(n-1) \\
 \vdots \\
 \vdots \\
 x^2(n-N+1) \\
 x^2(n) \\
 x(n)x^2(n-1) \\
 \vdots \\
 \vdots \\
 x(n)x^2(n-N+1) \\
 x^2(n)x(n-1) \\
 \vdots \\
 \vdots \\
 x^2(n)x(n-N+1) \\
 \vdots \\
 \vdots \\
 x^2(n-N+1) \\
 x(n)x(n-1)x(n-2) \\
 \vdots \\
 \vdots \\
 x(n-N-1)x(n-N)x(n-N+1)
 \end{bmatrix} \quad (2.6)$$

En la ecuación 2.6 y la Figura 2 es posible la comprensión de cómo se forma un núcleo de Volterra. Esta estructura desde una vector de entrada $\mathbf{x}(n)$, es la principal diferencia en relación a los filtros convencionales lineales, en de las series se utiliza solamente las muestras de la señal de entrada pasadas, y utilizar los productos cruzados entre muestras actuales y pasadas.

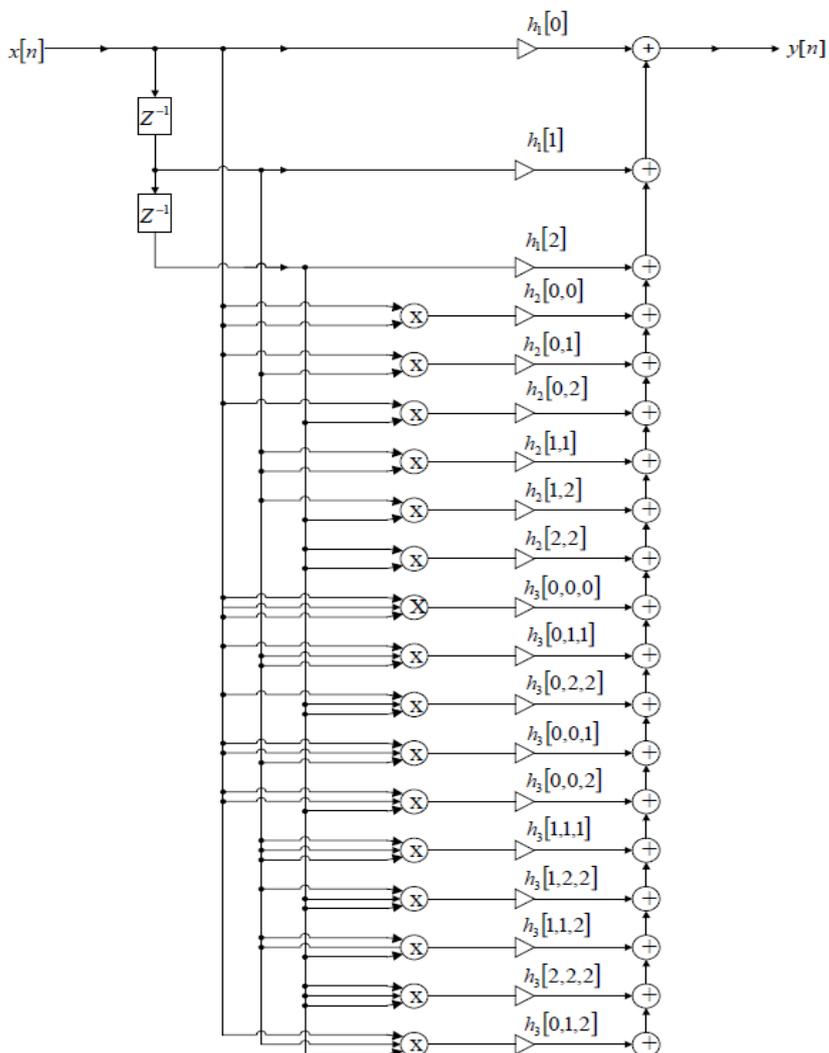


Figura 2. Sistema de Volterra orden 3 con 2 retrasos

Es preciso señalar que a medida que se aumenta el número de retardos o de orden de grado del polinomio generara un producto cruzado y se torna más difícil la elaboración del núcleo, generando una estructura de grandes dimensiones que se puede inviabilizar una solución del problema.

2.3.2. Filtro de Volterra

Un filtro de Volterra puede ser descompuesto en una estructura paralela de bloques, como se ilustra en la Figura 3, donde cada uno de los bloques corresponde a un filtro de orden p . Esa descomposición es bastante ventajosa desde el punto de vista práctico,

pues en algunas aplicaciones ciertos bloques pueden ser tratados de manera diferente o incluso ser ignorados.

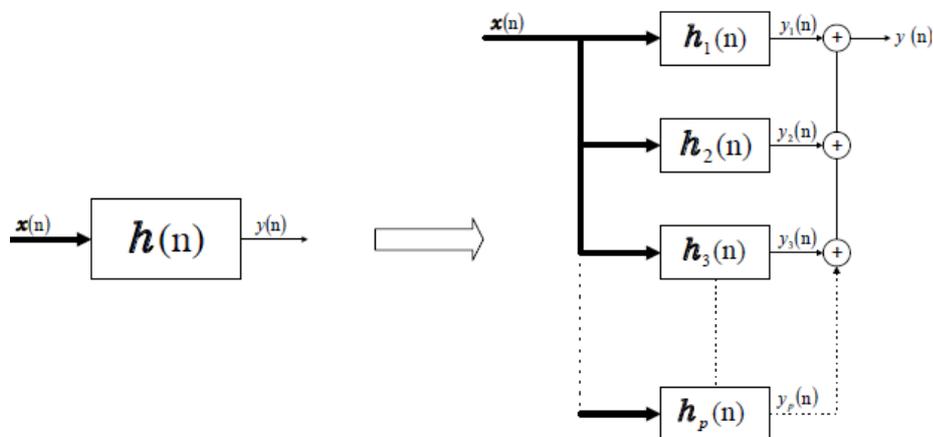


Figura 3. Descomposición del filtro de Volterra en bloques

La salida de uno de los bloques es un filtro de Volterra y puede ser escrito de forma vertical. Este enfoque es conocido para un bloque de primer orden (Oppenheim, 1996) que resulta ser:

$$Y_1(n) = \mathbf{h}_1^T \mathbf{x}_1(n) \quad (2.7)$$

Donde $\mathbf{x}_1(n)$ representa un vector de entrada de primer orden definido como $[x(n) \ x(n-1) \ \dots \ x(n-N+1)]^T$, \mathbf{h}_1 es un vector con los coeficientes. Para los demás bloques, es posible obtener representaciones analógicas representadas en la ecuación 2.7, considerándose una operación vectorial no lineal. Por ejemplo, para un bloque de segundo orden se puede escribir el vector de entrada del coeficiente como:

$$\mathbf{x}_2(n) = \begin{bmatrix} x^2(n) \\ x(n)x(n-1) \\ \vdots \\ x(n)x(n-N+1) \\ x^2(n-1) \\ \vdots \\ x^2(n-N+1) \end{bmatrix} \quad (2.8)$$

$$\mathbf{h}_2(n) = \begin{bmatrix} h_2(0,0) \\ h_2(0,1) \\ \vdots \\ h_2(0,N-1) \\ h_2(1,1) \\ \vdots \\ h_2(N-1,N-1) \end{bmatrix} \quad (2.9)$$

Resultando en una relación de entrada y salida dada por:

$$\mathbf{y}_2(n) = \mathbf{h}_2^T \mathbf{x}_2(n) \quad (2.10)$$

Una operación similar puede ser realizada para los otros bloques no lineales con filtro de Volterra, resultando la siguiente relación de entrada y salida para un bloque de un orden p cualquiera.

$$\mathbf{y}_p(n) = \mathbf{h}_p^T \mathbf{x}_p(n) \quad (2.11)$$

Con respecto a la complejidad computacional requerida, el filtro de Volterra presenta un serio problema. El número de coeficientes crece exponencialmente con el aumento de tamaño de memoria. Donde $D_{pb}(N)$ es el número de coeficientes de cada bloque de orden P_b de un filtro de Volterra con tamaño de memoria N , (Tan, 2001):

$$D_{pb}(N) = \frac{(N + P_b - 1)}{(N - 1)! P_b!} \quad (2.12)$$

Considerando $D_v(N, p)$ como un número total de coeficientes de un filtro de Volterra de orden p y tamaño de memoria N , un número total de coeficientes está dada por la sumatoria de los coeficientes de cada uno de los bloques así:

$$D_v(N, p) = \sum_{P_b=1}^p D_{pb}(N) \quad (2.13)$$

Lo que resulta que:

$$D_v(N, p) = \frac{(N + p)!}{N! p!} - 1 \quad (2.14)$$

2.3.3. Modelo de Volterra en frecuencia-simple

La gran dificultad en la construcción del modelo de Volterra y la extracción de sus parámetros está en la medición de los datos que son multidimensionales en tiempo y en frecuencia. Los modelos de Volterra en frecuencia simple representan una versión simplificada del sistema general. Estas configuraciones simplifican la estimación de parámetros y reduce la complejidad computacional puesto que todo cálculo espectral requiere involucrar solo una variable de frecuencia simple en lugar de varias (Silva, Moulthrop, & Muha, 2001).

$$\begin{aligned}
 h_n(\lambda_1, \dots, \lambda_n) &= a_n h_n(\lambda_1) h_n(\lambda_2) \dots \dots \dots h_n(\lambda_n) \\
 h_n(\lambda_1, \dots, \lambda_n) &= a_n \prod_{i=1}^n h_n(\lambda_i)
 \end{aligned}
 \tag{2.15}$$

2.4. Análisis de banda base complejo para sistemas no lineales con memoria

2.4.1. Series complejas de Volterra

Las series de Volterra representa la salida del sistema como una convolución de potencias de la entrada con el núcleo de Volterra. Se puede analizar considerando una entrada que consiste en una portadora de modulación simple de frecuencia central ξ y aplicada a un amplificador no lineal. Esta señal de entrada puede escrita en términos de su envolvente compleja como:

$$\begin{aligned}
 \mathbf{x}(t) &= \text{Re}\{\tilde{\mathbf{x}}(t)e^{j2\pi\xi t}\} \\
 \mathbf{x}(t) &= \frac{1}{2}\tilde{\mathbf{x}}(t)e^{j2\pi\xi t} + \frac{1}{2}\tilde{\mathbf{x}}^*(t)e^{-j2\pi\xi t}
 \end{aligned}
 \tag{2.16}$$

La salida del modelo de Volterra puede ser expresada de la siguiente manera:

$$\begin{aligned}
 y(n) &= \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_n(\lambda_1, \dots, \lambda_n) x \prod_{i=1}^n \frac{1}{2} (\tilde{\mathbf{x}}(t - \lambda_i) e^{j2\pi\xi(t-\lambda_i)} \\
 &\quad + \tilde{\mathbf{x}}^*(t - \lambda_i) e^{-j2\pi\xi(t-\lambda_i)}) d\lambda_i
 \end{aligned}
 \tag{2.17}$$

El componente de $y_n(t)$ es la frecuencia fundamental central puede ser expresada de la siguiente manera:

$$\begin{aligned} \widetilde{y}_n(t) = \frac{1}{2^{\frac{n+1}{2}}} \binom{n}{\frac{n-1}{2}} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \tilde{h}_n(\tau_1, \dots, \lambda_n) \prod_{i=1}^{\frac{n+1}{2}} \tilde{x}(t \\ - \tau_i) \prod_{i=\frac{n+3}{2}}^n \tilde{x}^*(t - \lambda_i) d\lambda_i \dots d\lambda_n \end{aligned} \quad (2.18)$$

Donde $\frac{1}{2^{\frac{n+1}{2}}} \binom{n}{\frac{n-1}{2}} \tilde{h}_n(\tau_1, \dots, \lambda_n)$ es un equivalente a un filtro pasa bajos del núcleo de Volterra $h_n(\lambda_1, \dots, \lambda_n)$. Por lo tanto, la salida del modelo de Volterra en la envolvente compleja se podría escribir como:

$$\begin{aligned} \widetilde{y}_n(t) = \sum_{n=0}^N \frac{1}{2^{\frac{n+1}{2}}} \binom{n}{\frac{n-1}{2}} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \tilde{h}_n(\tau_1, \dots, \lambda_n) \prod_{i=1}^{\frac{n+1}{2}} \tilde{x}(t \\ - \tau_i) \prod_{i=\frac{n+3}{2}}^n \tilde{x}^*(t - \lambda_i) d\lambda_i \dots d\lambda_n \end{aligned} \quad (2.19)$$

2.5. Modelo de Volterra en frecuencia simple no lineal

Para el modelo de Volterra de filtro no lineal en frecuencia simple, se usa el núcleo en relación con la ecuación (2.15). Resulta que la salida $y_n(t)$ puede ser escrito como

$$\begin{aligned} y_n(t) = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} a_n \prod_{i=1}^n h_n(\lambda_i) x(t - \lambda_i) d\lambda_i \\ y_n(t) = a_n u_n^n(t) \end{aligned} \quad (2.20)$$

donde

$$u_n(t) = \int_{-\infty}^{\infty} h_n(\lambda) x(t - \lambda) d\lambda \quad (2.21)$$

representa la entrada filtrada. Por lo tanto, la salida n -ésima consiste en la potencia de forma de onda en las entradas filtradas para un filtro lineal de n -ésima salida. Aplicando el mismo análisis no lineal sin memoria, la envolvente compleja de la señal de salida de la frecuencia ξ puede ser escrita como:

$$\tilde{y}_n(t) = \sum_{n=1}^N \frac{1}{2^{n-1}} \left(\frac{n}{n+1} \right) a_n |\tilde{u}_n(t)|^{n-1} \tilde{u}_n(t) \quad (2.22)$$

donde

$$\tilde{u}_n(t) = \int_{-\infty}^{\infty} \tilde{h}_n(\lambda) \tilde{x}(t - \lambda) d\lambda \quad (2.23)$$

y $\tilde{h}_n(t)$ es la banda base equivalente a un filtro de n -ésima salida con respecto a la frecuencia ξ .

Para un modelo de filtro no lineal, el núcleo se relaciona con las salidas activas del sistema que se pueden escribir como

$$y_n(t) = a_n \int_{-\infty}^{\infty} h_n(\lambda) x^n(t - \lambda) d\lambda \quad (2.24)$$

Usando la ecuación (2.19), la envolvente compleja de salida de n -ésimo orden puede ser escrita como

$$\tilde{y}_n(t) = \frac{1}{2^{n-1}} \left(\frac{n}{n+1} \right) a_n \int_{-\infty}^{\infty} \tilde{h}_n(\lambda) |\tilde{x}_n(t - \lambda)|^{n-1} \tilde{x}(t - \lambda) \quad (2.25)$$

Donde \tilde{h}_n la banda base equivalente de un filtro $h_n(t)$ con respecto a la frecuencia ξ . Por lo tanto, la envolvente compleja de la forma de onda de salida $y(t)$ es:

$$\tilde{y}(t) = \sum_{n=1}^N \tilde{y}_n(t) \quad (2.26)$$

2.6. Algoritmo de Volterra LMS

Este algoritmo es ampliamente utilizado, el cual se conoce con el nombre de algoritmo de Cuadrados Medios Mínimos (LMS, por sus siglas en inglés) este algoritmo fue creado por Widrow y Hoff en el año de 1960. El algoritmo LMS pertenece a la familia de algoritmos de gradiente estocástico. El término “gradiente estocástico” pretende distinguir el algoritmo LMS del método de máxima pendiente que usa un gradiente determinista en un cálculo recursivo de los filtros Wiener para entradas estocásticas. Una característica muy importante y a la vez atractiva del algoritmo LMS es su simplicidad, no requiere medición de funciones de correlación, tampoco necesita de inversión de matrices. (Chicas, 2004).

El algoritmo LMS es un algoritmo de gradiente estocástico en el que se hacen iteraciones de cada valor de los coeficientes de un filtro transversal en la dirección del gradiente con la magnitud al cuadrado de la señal de error con respecto al valor del coeficiente. Así que, el algoritmo LMS está estrechamente relacionado con el concepto de aproximación estocástica desarrollada por (Monro & Robbins, 1951) que se encuentran en estadística para resolver ciertos problemas de estimación de parámetros secuenciales. La primera diferencia entre ellos, es que el algoritmo LMS utiliza un parámetro de tamaño de paso fijo para controlar la corrección aplicada a cada valor de los coeficientes de una iteración a la próxima, mientras que en el método de aproximación estocástica el parámetro del tamaño de paso es inversamente proporcional al tiempo o la potencia de n (n es la variable independiente de tiempo discreto).

El algoritmo LMS está basado en el algoritmo *steepest-descent* cuya adaptación es un vector de coeficientes dada por (Haykin, 1996):

$$\mathbf{w}(n + 1) = \mathbf{w}(n) - \mu \nabla \mathbf{w} \xi(n), \quad (2.27)$$

Donde μ es denominado *step-size* con un tamaño de paso de adaptación cuyo valor debe ser escogido de forma adecuada para garantizar la convergencia del algoritmo (Bernal, 2005).

En la practica el gradiente $\nabla \mathbf{w}\xi(n)$, no se puede obtener con exactitud, teniendo en cuenta que \mathbf{p} y \mathbf{R}_{xx} no son conocidas. Una manera de encontrar este problema es usar estimativas.

Para la implementación del algoritmo LMS, se emplea los siguientes estimadores despolarizados de \mathbf{p} y \mathbf{R}_{xx}

$$\hat{\mathbf{p}} = \mathbf{x}(n)\mathbf{d}(n) \quad (2.28)$$

$$\hat{\mathbf{R}}_{xx} = \mathbf{x}(n)\mathbf{x}^T(n) \quad (2.29)$$

Tabla 2.
Algoritmo LMS

V-LMS	
Inicializar:	$\mathbf{x}(0) = \mathbf{w}(0) = [0 \ 0 \ \dots \ 0]^T$
Para $k \geq 0$	$e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$
	$\mathbf{w}(k+1) = \mathbf{w}(k) + 2 \begin{bmatrix} \mu_1 & \dots & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & \dots & \mu_1 & 0 & \dots & 0 \\ 0 & \dots & 0 & \mu_2 & \dots & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & \dots & 0 & 0 & \dots & \mu_2 \end{bmatrix} e(k)\mathbf{x}(k)$
}	

Para estos estimadores, podemos observar una estimativa despolarizada de gradiente como:

$$\begin{aligned} \hat{\nabla} \mathbf{w}\xi(n) &= -2\mathbf{x}(n)d(n) + 2\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n) \\ &= -2\mathbf{x}(n)[d(n) - \mathbf{x}^T(n)\mathbf{w}(n)] \\ &= -2e(n)\mathbf{x}(n). \end{aligned} \quad (2.30)$$

Substituyendo la estimativa da la Ecuación 2.30 en Ecuación. 2.27, obtenemos la siguiente relación para la actualización del vector coeficientes del algoritmo LMS.

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n) \quad (2.31)$$

2.7. Algoritmo de Volterra NLMS

Uno de los principales inconvenientes del algoritmo LMS es la baja velocidad de convergencia. Para tratar de resolver este problema, se ha propuesto muchos algoritmos de filtraje adaptativo basados en LMS convencional. Uno de esos algoritmos es el LMS normalizado, comúnmente denotado por NLMS, que emplea una señal normalizada de entrada.

Con el fin de mejorar la tasa de convergencia del algoritmo LMS, se adopta en el algoritmo NLMS un factor de convergencia variable. Un valor de μ_n escogido tal que el error a posteriori (después de la actualización de dos coeficientes) será igual a cero, tal que $\frac{\partial \Delta e^2(n)}{\partial \mu_n} = 0$.

Tabla 3.
Algoritmo NLMS

Volterra NLMS	
Inicializar:	$\mathbf{x}(0) = \mathbf{w}(0) = \mathbf{0}$
Elegir μ_N en un intervalo $0 < \mu_N \leq 2$	
$\gamma =$ constante pequeña	
para $n = 1, 2, \dots$	
{	
$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n)$	
$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu_n}{\gamma + \mathbf{x}^T(n)\mathbf{x}(n)}$	$\begin{bmatrix} \mu_1 & \dots & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & \dots & \mu_1 & 0 & \dots & 0 \\ 0 & \dots & 0 & \mu_2 & \dots & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & \dots & 0 & 0 & \dots & \mu_2 \end{bmatrix} e(n)\mathbf{x}(n)$
}	

Resolviendo esta ecuación diferencial para μ_n se llega a que:

$$\mu_n = \frac{1}{2\mathbf{x}^T(n)\mathbf{x}(n)} \quad (2.32)$$

Considerando la Ecuación 2.28, el uso de μ_n en lugar de μ empleándose en la ecuación así obteniendo la Ecuación 2.29, se llega a la ecuación adoptada por el algoritmo LMS normalizado, que está dada por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{e(n)\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)} \quad (2.33)$$

Usando el factor de convergencia rígida μ_N , introducido en la fórmula de actualización como controles de ajuste, e introduciendo un parámetro γ al fin de evitar posibles divisiones por cero, la ecuación de actualización de los coeficientes puede ser expresada por:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu_N}{\gamma + \mathbf{x}^T(n)\mathbf{x}(n)} e(n)\mathbf{x}(n) \quad (2.34)$$

El conjunto de ecuaciones del algoritmo NLMS esté representado en la Tabla 3 (Diniz, 2002). A pesar de μ_N se puede asumir valores mayores de uno y menos de dos, en la práctica, se utiliza $0 < \mu_N \leq 2$.

2.8. Algoritmo de Volterra RLS

Cuando el medio es altamente dinámico se requiere de algoritmos que se adapten rápidamente a los cambios. Para estos casos el algoritmo LMS no nos brinda un buen desempeño. Con esos propósitos se crearon algoritmos de rápida respuesta, tal como el algoritmo RLS. Una manera de aumentar la velocidad de convergencia del algoritmo adaptativo es calcular directamente el vector de pesos a través de la ecuación de Wiener.

El algoritmo RLS realiza una búsqueda iterativa por el vector de coeficientes óptimos, directamente a través de la siguiente ecuación:

$$\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1} r_{dx} \quad (2.35)$$

Las aproximaciones estocásticas de las matrices \mathbf{R}_{xx} y r_{dx} son actualizadas recursivamente en cada iteración de la siguiente manera:

$$\mathbf{R}_{xx}(k) = \varphi \mathbf{R}_{xx}(k-1) + \mathbf{x}(k) \mathbf{x}^H(k) \quad (2.36)$$

$$\mathbf{r}_{dx}(k) = \varphi \mathbf{r}_{dx}(k-1) + d^*(k) \mathbf{x}(k) \quad (2.37)$$

Para obtener la ecuación de Wiener, se necesita calcular la inversa de $\mathbf{R}_{xx}(k)$. Para esto se utiliza la matriz de inversión de Lemma.

$$\mathbf{R}_{xx}^{-1}(k) = \varphi^{-1} \left[\mathbf{R}_{xx}^{-1}(k-1) - \frac{\mathbf{R}_{xx}^{-1}(k-1) \mathbf{x}(k) \mathbf{x}^H(k) \mathbf{R}_{xx}^{-1}(k-1)}{\varphi + \mathbf{x}^H(k) \mathbf{R}_{xx}^{-1}(k-1) \mathbf{x}(k)} \right] \quad (2.38)$$

Para la obtención del vector de ponderaciones óptimas se reemplazan las Ecuaciones (2.37) y (2.36) en (2.35), con lo que se obtiene la forma genérica del algoritmo RLS.

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \frac{\varphi^{-1} \mathbf{R}_{xx}^{-1}(k-1)}{1 + \varphi^{-1} \mathbf{x}^H(k) \mathbf{R}_{xx}^{-1}(k-1) \mathbf{x}(k)} \mathbf{x}(k) e^*(k) \quad (2.39)$$

$$\text{Haciendo } \varphi_{RLS} = \frac{\varphi^{-1} \mathbf{R}_{xx}^{-1}(k-1)}{1 + \varphi^{-1} \mathbf{x}^H(k) \mathbf{R}_{xx}^{-1}(k-1) \mathbf{x}(k)} \quad (2.40)$$

Se obtiene una forma más resumida de la ecuación (2.36). Ésta se presenta a continuación.

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \varphi_{RLS} \mathbf{x}(k) e^*(k) \quad (2.41)$$

A diferencia del algoritmo anterior, φ_{RLS} es variable. Éste se va adaptando frente a los cambios en el medio. De esta forma se obtiene un algoritmo RLS de rápida

convergencia, pero con mayor complejidad, por la mayor carga de procesamiento de datos (Espinosa & Villaroel, 2007).

2.9. Sistemas no lineales

El principal objetivo es analizar el comportamiento no lineal y la aplicabilidad de varios modelos a un sistema dado. En primer lugar, se presentan y analizan modelos no lineales sin memoria de banda.

A continuación, se presentan modelos analíticos no lineales con memoria basados en un modelo de la serie de Volterra y sus variantes. Los casos especiales del modelo de Volterra se presentan definiendo su relación del núcleo. Finalmente, se presentan modelos empíricos que se basan en ajustar las características medidas a los modelos preestablecidos. Estos modelos se usan generalmente para generar características no lineales en simulaciones por computadora de sistemas no lineales (Gharaibeh, 2012).

2.9.1. Modelo Rapp

El modelo Rapp considerado solamente la conversión AM-AM y es definido por un factor de suavidad y de salida niveles de parámetros de saturación. La conversión AM-AM es calculada acorde a la ecuación 2.42.

$$F_{AM-AM}(u) = \frac{G_u}{\left(1 + \left(\frac{u}{V_{sat}}\right)^{2p}\right)^{\frac{1}{2p}}} \quad (2.42)$$

Donde u es la magnitud de la señal escalada, p es el factor de suavidad y V_{sat} el nivel de saturación de salida. Los parámetros límites del nivel de saturación de salida es el nivel de salida de la señal, mientras que los parámetros del factor de suavidad controlan las características de ganancia-amplitud.

CAPÍTULO 3

3. MATERIALES Y METODOS

Se generó una señal binaria aleatoria la cual fue introducida al convolucional para posteriormente ser modulada con 64QAM y aplicada OFDM. Esta señal fue la que paso por un canal no lineal “Rapp” para ser utilizada en el algoritmo creado que tuvo como objetivo fundamental la extracción de características del polinomio de Volterra y estos coeficientes se pudo diseñar un filtro compensador utilizando los algoritmos RLS, LMS y NLMS, después se procedió a realizar el proceso inverso para simular y realizar los análisis comparativos del caso.

3.1. Proceso de transmisión.

3.1.1. Creación de señal aleatoria

Para la primera etapa se generó una señal binaria aleatoria conforme a las características que posee el estándar IEEE 802,11 y el uso del OFDM (Multiplexacion por división de frecuencia ortogonal), así como sus demás características no menos importantes. En la Figura 4 como se puede observar a continuación esta la señal generada es decir una binaria aleatoria.

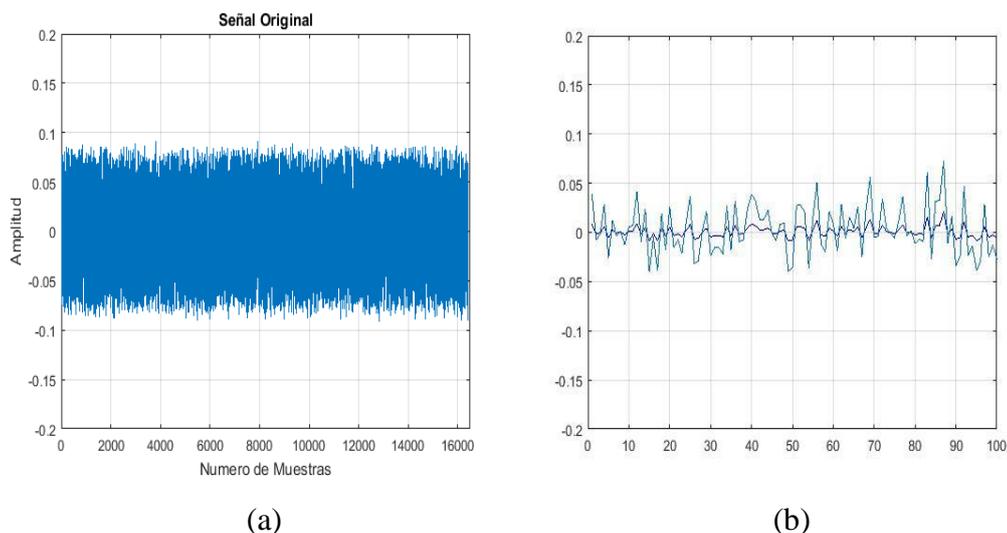


Figura 4. (a) Señal original; (b) Zoom señal original intervalo (0-100) muestras.

3.1.2. Código convolucional

Se utilizó códigos convolucionales ya que son de canal, y sirven para poder mejorar el porcentaje de la tasa de error, en el programa la señal generada fue introducida con una tasa convolucional $r_c = \frac{2}{3}$ la cual hace referencia a # de señales de entrada con el # de señales de salida y con una matriz generadora G_c la cual ayudó a generar una nueva señal, se puede observar la señal en la Figura 5 después del convolucional que a simple vista es la misma pero se realizó una ampliación para observar mejor que están ahí los 1's y 0's.

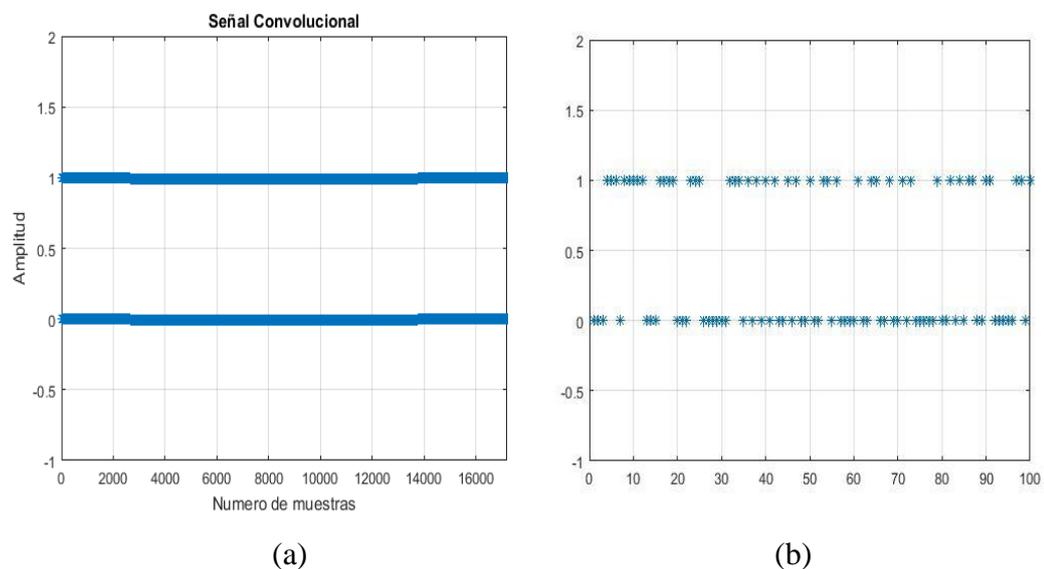


Figura 5. (a) Señal del convolucional; (b) Zoom señal convolucional intervalo (0-100) muestras.

3.1.3. Modulación y demodulación 64-QAM

Se aplicó la modulación 64-QAM para poder aprovechar el ancho de banda y que se pueda transmitir más datos en menor tiempo dicha modulación es utilizada para transmisión en HD (high definition), y para usos posteriores es necesario ya que se utilizó el estándar IEEE 802.11, por otra parte se realizó su proceso de demodulación para posteriormente entrar al código de Viterbi.

Se puede observar en la Figura 6 como está formada la constelación 64-QAM.

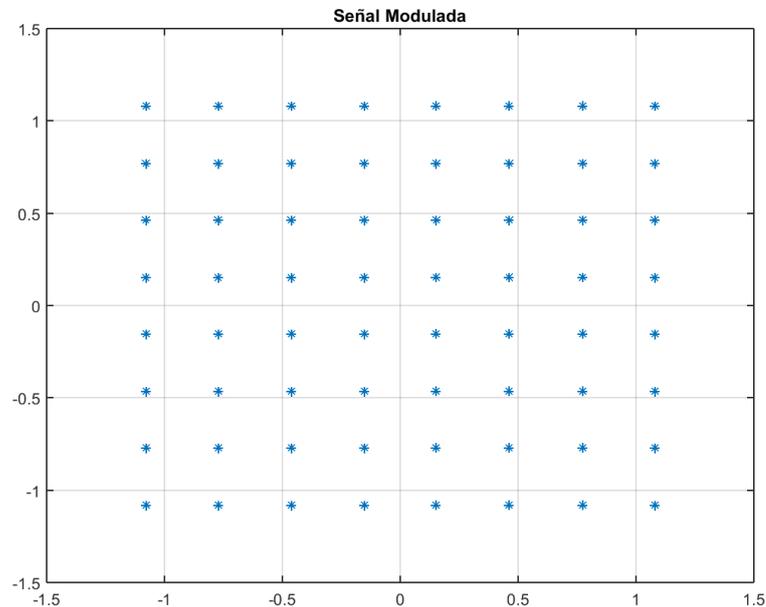


Figura 6. Señal modulada 64-QAM

3.1.4. OFDM

La modulación por división ortogonal de frecuencia fue aplicada ya que por lo general se utiliza tras pasar por el codificador convolucional y al usar en conjunto estas dos la modulación ayuda a modular en un conjunto de portadoras de diferente frecuencia pero que sean ortogonales entre sí. Todo esto tuvo como objetivo corregir la interferencia co-canal.

3.1.5. Canal no lineal Rapp

Se utilizó el modelo Rapp ya que estos son diseñados para modelar amplificadores de potencia de estado sólido, este modelo tiene una distorsión no lineal en amplitud y por este mismo motivo fue utilizado para esta simulación de canal, se introdujo la señal modulada la cual dio como resultado una señal nueva llamada la que fue guardada para ser usada posteriormente, a continuación, en la Figura 7 se puede observar la distorsión que produjo la cual es evidentemente.

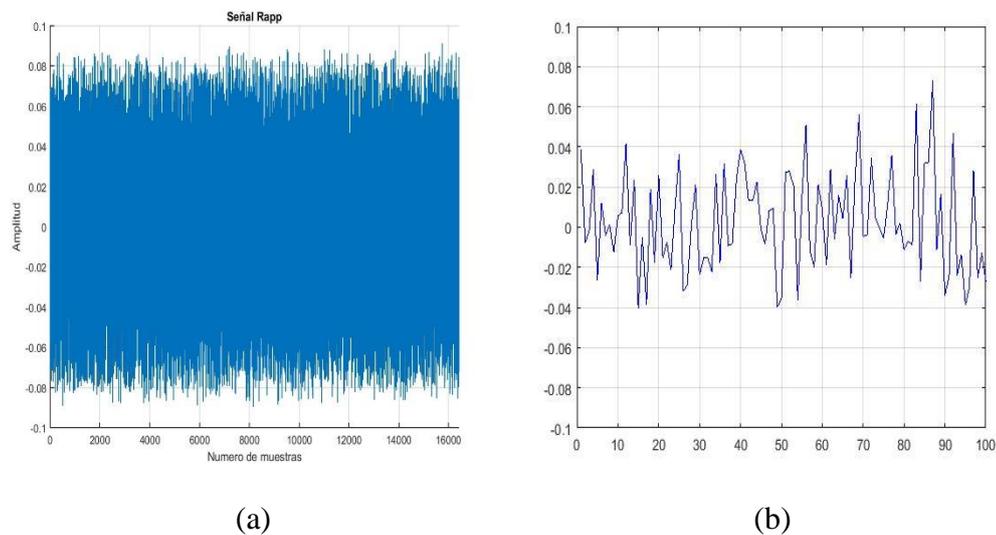


Figura 7. (a) Señal después del canal no lineal Rapp; (b) Zoom señal después del canal no lineal Rapp intervalo (0-100) muestras.

3.1.6. Filtro de Volterra con los algoritmos NLMS, LMS y RLS

Para esta etapa se cargó la señal guardada que paso por el canal no lineal y la señal original, para realizar el filtro de Volterra con los 3 algoritmos se usó varias combinaciones con varios niveles y varios retrasos hasta poder obtener el que mejor se adapte para una buena convergencia.

Se extrajo los coeficientes característicos de la señal y se obtuvo la media de las últimas características donde la señal converge por completo, esta fue guardada para utilizarla posteriormente en el compensador. A continuación, se muestra los resultados de convergencia de cada algoritmo.

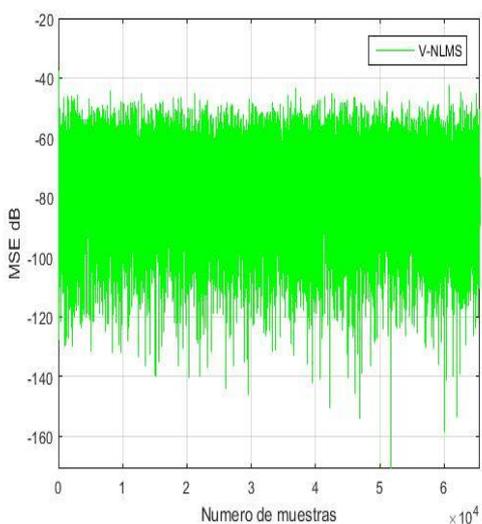
Los modelos no lineales analíticos se basan en el análisis de la serie de Volterra y la teoría desarrollada por Wiener (Diniz, 2002) para modelar un sistema no lineal con memoria. En un análisis general de la serie de Volterra, un modelo no lineal generalmente consiste en filtros lineales o no lineales con anchos de banda finita que determinan la selectividad de frecuencia del sistema.

3.1.6.1. Volterra NLMS

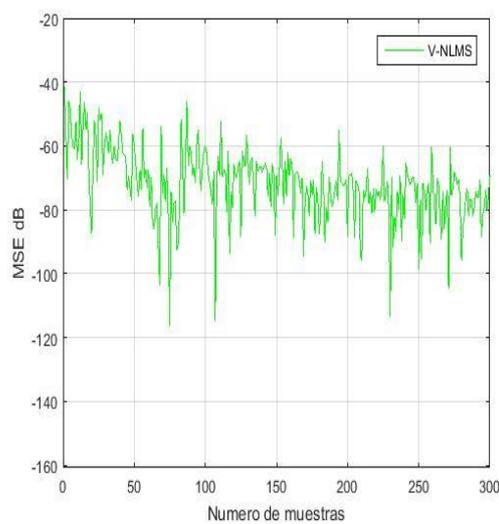
Se realizó varias mediciones para el algoritmo normalizado, se tomó en cuenta los coeficientes para cada nivel, para poder observar el comportamiento del filtro se graficó como se puede ver en la Figura 8, por lo tanto después de realizar varias simulaciones se obtuvieron los siguientes resultados mostrados en la Tabla 4.

Tabla 4.
Variación de parámetros

V-NLMS	
μ	MSE
0.999	57.6890
0.99999	-57.6884
0.79999	-57.8178
0.991	-56.3937
0.01999	-58.4439
0.00199	-57.0867



(a)



(b)

Figura 8. (a) Curva de V-NLMS; (b) Zoom curva de V-NLMS intervalo (0-300) muestras.

3.1.6.2. Volterra LMS

En este algoritmo de Volterra se maneja un coeficiente para cada nivel como se trabajó con orden 3 debido a un mejor desempeño, los mejores coeficientes son $\mu_1 = 0.00098$, $\mu_2 = 0.0087$, $\mu_3 = 0.0099$, por esta razón se trabajó con estos valores para obtener la mejor convergencia y MSE, como se puede observar en la figura 9 y en la

Tabla 5.
Variación de Coeficiente

V-LMS	
μ	MSE
$\mu_1 = 0.99, \mu_2 = 0.98, \mu_3 = 0.97$	-56.1507
$\mu_1 = 0.79, \mu_2 = 0.78, \mu_3 = 0.77$	-56.2163
$\mu_1 = 0.079, \mu_2 = 0.078, \mu_3 = 0.077$	-58.1293
$\mu_1 = 0.069, \mu_2 = 0.068, \mu_3 = 0.067$	-58.4250
$\mu_1 = 0.9, \mu_2 = 0.8, \mu_3 = 0.7$	-57.2078
$\mu_1 = 0.1, \mu_2 = 0.2, \mu_3 = 0.3$	-57.8030

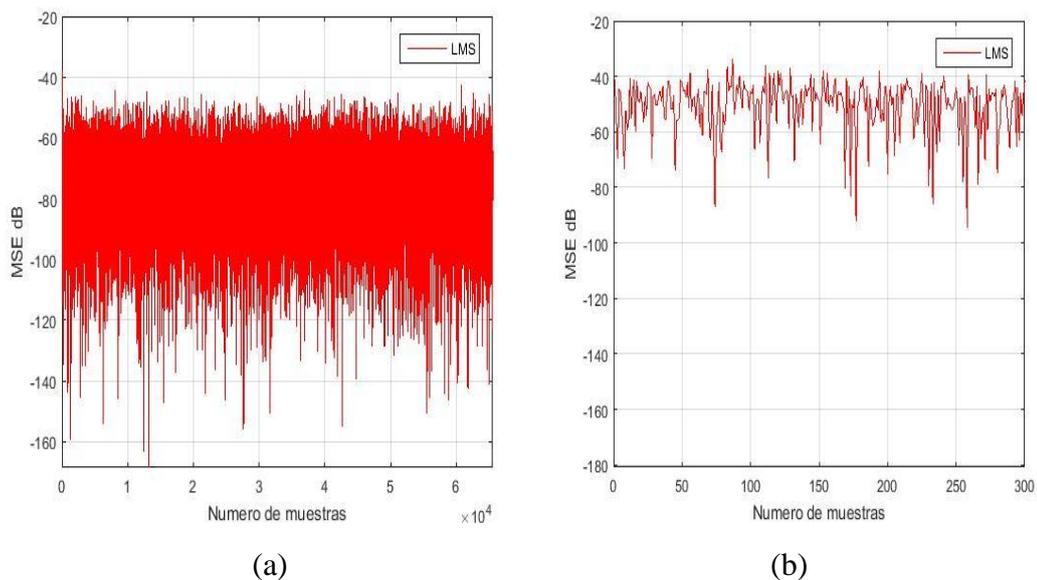


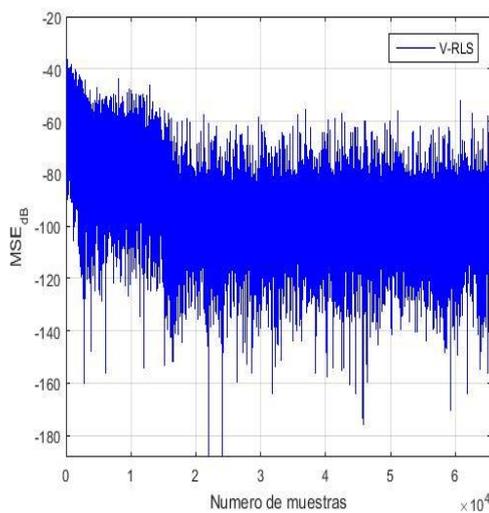
Figura 9. (a) Curva de V-LMS; (b) Zoom curva de V-LMS intervalo (0-300) muestras.

3.1.6.3. Volterra RLS

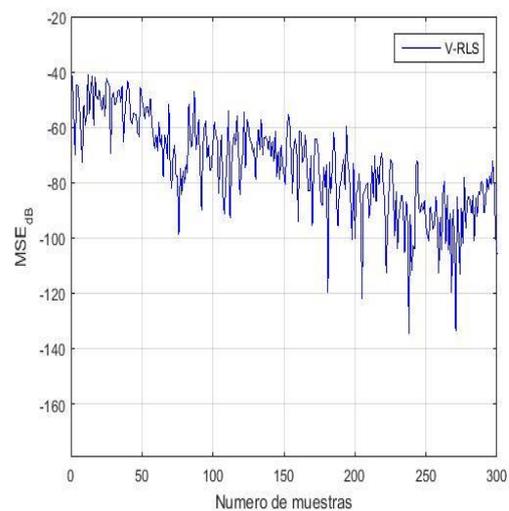
Después de observar el comportamiento de este algoritmo se podría decir que es el mejor hasta este momento por su rapidez de convergencia, por lo tanto, se escogió el mejor coeficiente que es $\lambda=0.9989$ que se demuestra en la Tabla 6, con este valor se trabajó hasta el final de las comparaciones, a continuación, se puede observar el comportamiento en la Figura 10.

Tabla 6.
Variación de λ

V-RLS	
λ	MSE
0.999881	-66.6151
0.99881	-56.4812
0.999911	-57.8515
0.9989	-63.3950
0.9999981	-67.0488
0.9999989	-67.0518



(a)



(b)

Figura 10. (a) Curva de V-RLS; (b) Zoom curva de V-RLS intervalo (0-300) muestras.

CAPÍTULO 4

4. PRUEBAS Y RESULTADOS

Es este capítulo se mostrara la demodulación y se podrá observar en las Figuras 13, 14 y 15 las constelaciones generadas y las que fueron recibidas así como también los resultados en tablas tabuladas para poder saber que algoritmo de Volterra fue el más eficiente utilizando el MSE y el BER para poder saber por reacción de convergencia y por porcentaje de pérdida de información. A continuación se presenta los siguientes resultados para posteriormente mostrar los análisis y conclusiones correspondientes de este trabajo realizado.

4.1. Diagrama de bloques

Todo el trabajo de simulación se basa en los siguientes estados los cuales se pueden apreciar en la Figura 11, donde está el proceso de transmisión y recepción de la señal aleatoria generada.



Figura 11. Diagrama de bloques de la señal transmitida

Después del proceso de transmisión se tomó la señal y se realizó el proceso inverso utilizando los diferentes filtros de Volterra como NLMS, LMS y RLS, para la creación del filtro correspondiente al extraer las características y crear su compensador. A continuación, se puede observar en la Figura 12 el proceso de recepción de la señal.



Figura 12. Diagrama de bloques del proceso de recepción

Así es como se simuló todo el escenario de transmisión y recepción utilizando las características del estándar 802.11, a continuación se detalla el proceso de demodulación y filtraje no lineal.

4.2. Demodulación

A continuación se mostraron las gráficas obtenidas de la constelación 64-QAM de la señal recibida, se evaluó cada algoritmo para poder verificar la demodulación en errores de bits expresados en porcentaje como las gráficas que muestran los resultados para cada SNR obtenidos de la mejor MSE mostrados en la Tabla 4, 5 y 6.

4.2.1. Volterra NLMS

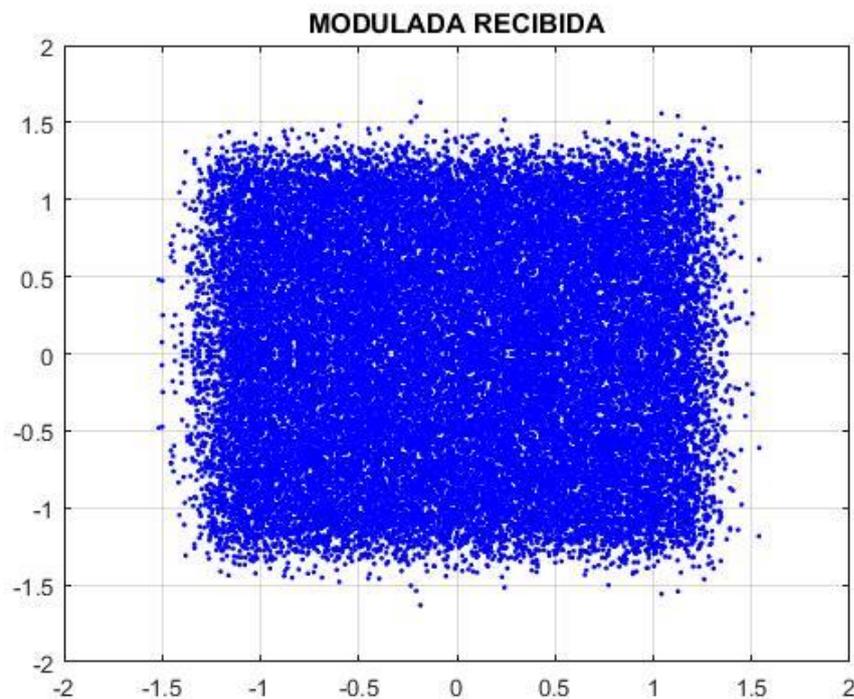


Figura 13. Señal Rx con V-NLMS (SNR=10 dB, BER= 9,05 %)

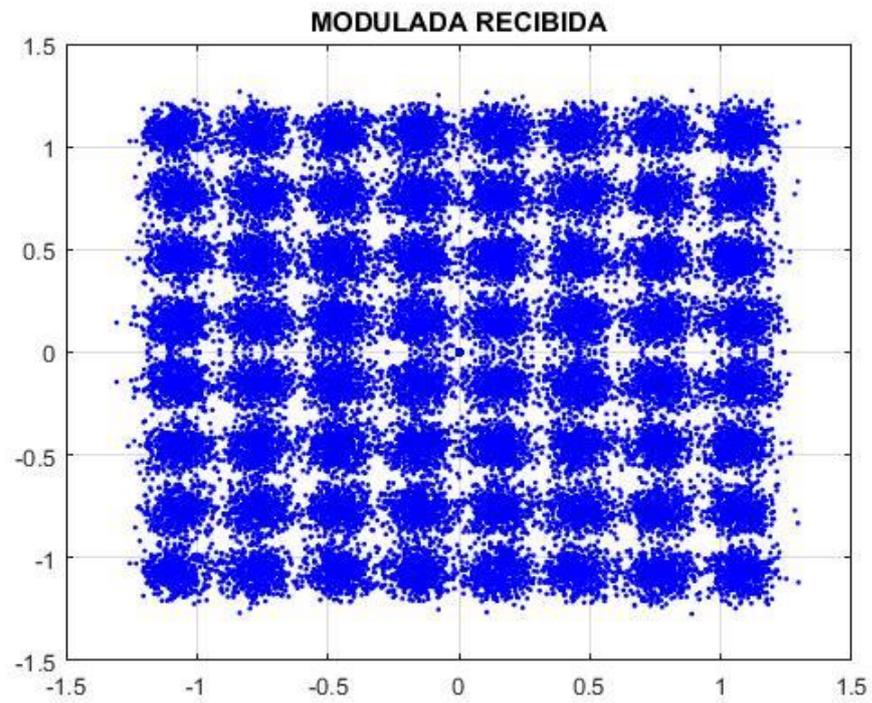


Figura 14. Señal Rx con V-NLMS (SNR=20 dB, BER= 0,49 %)

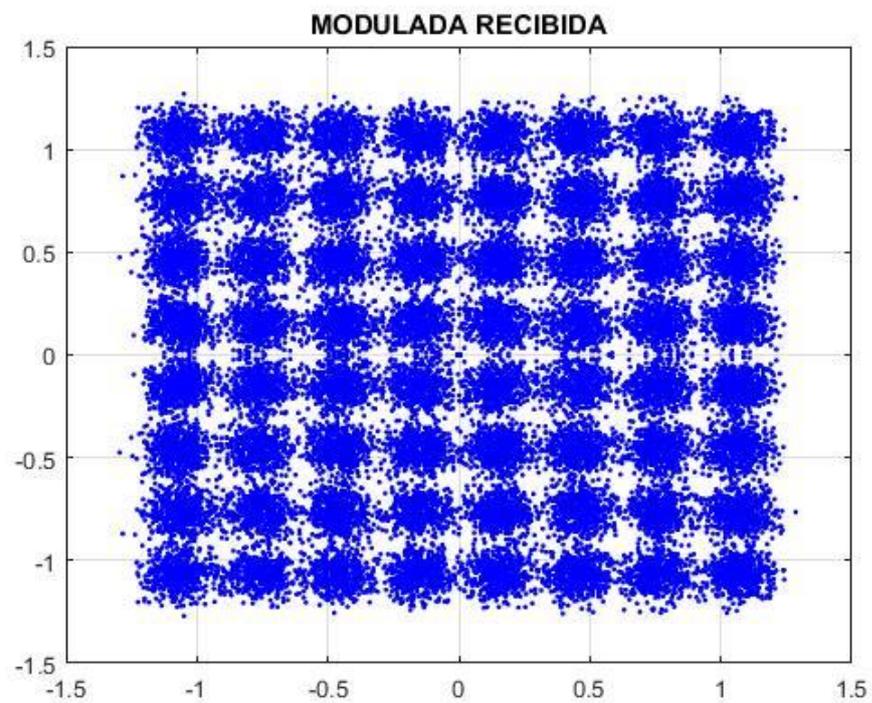


Figura 15. Señal Rx con V-NLMS (SNR=30 dB, BER= 0.075785 %)

4.2.2. Volterra LMS

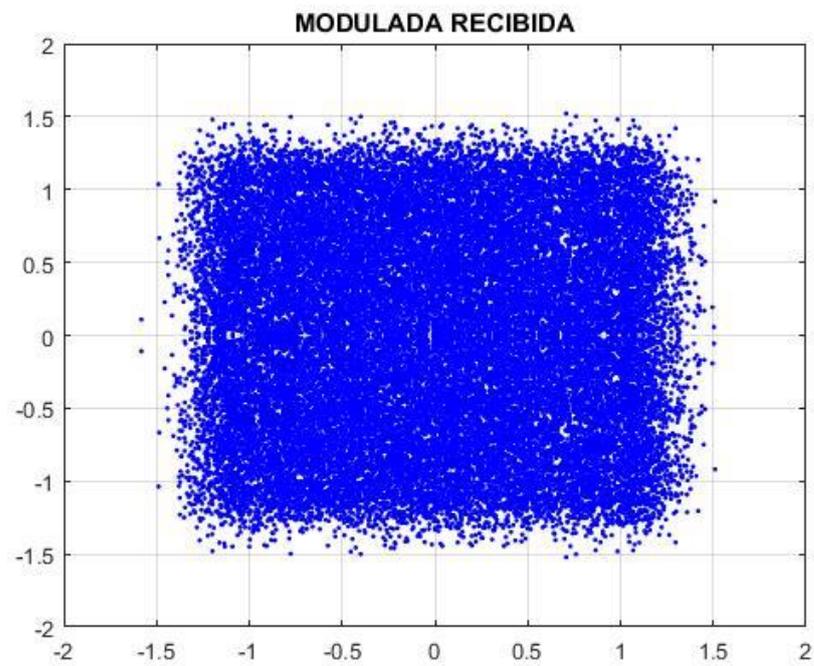


Figura 16. Señal Rx con V-LMS (SNR=10 dB, BER= 8,75 %)

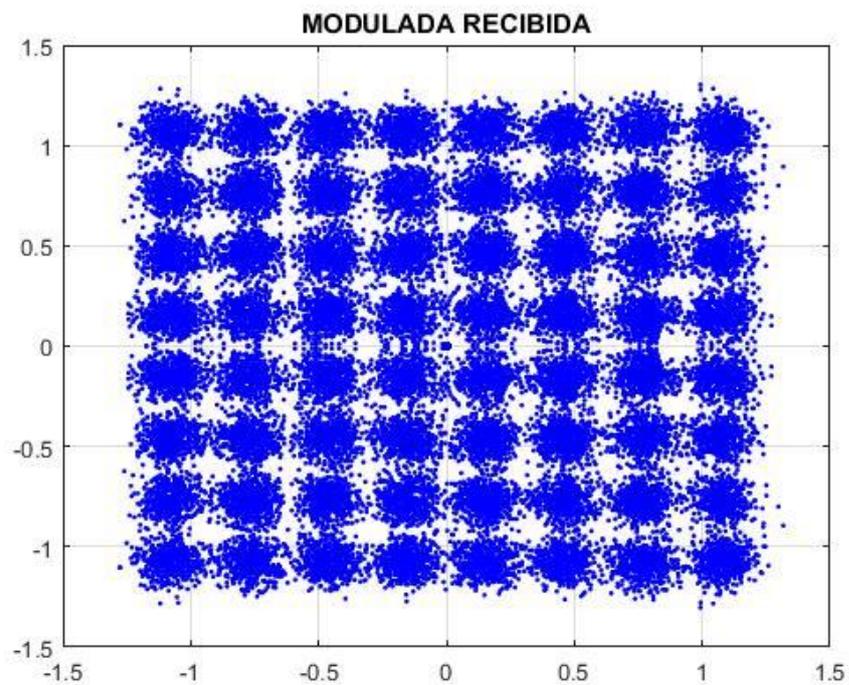


Figura 17. Señal Rx con V-LMS (SNR=20 dB, BER= 0,5 %)

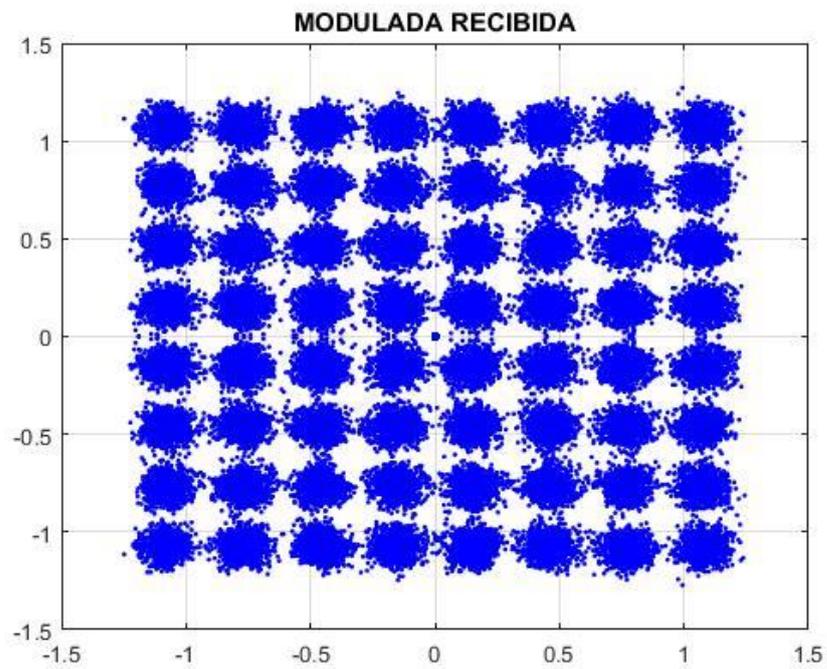


Figura 18. Señal Rx con V-LMS (SNR=30 dB, BER= 0.072734%)

4.2.3. Volterra RLS

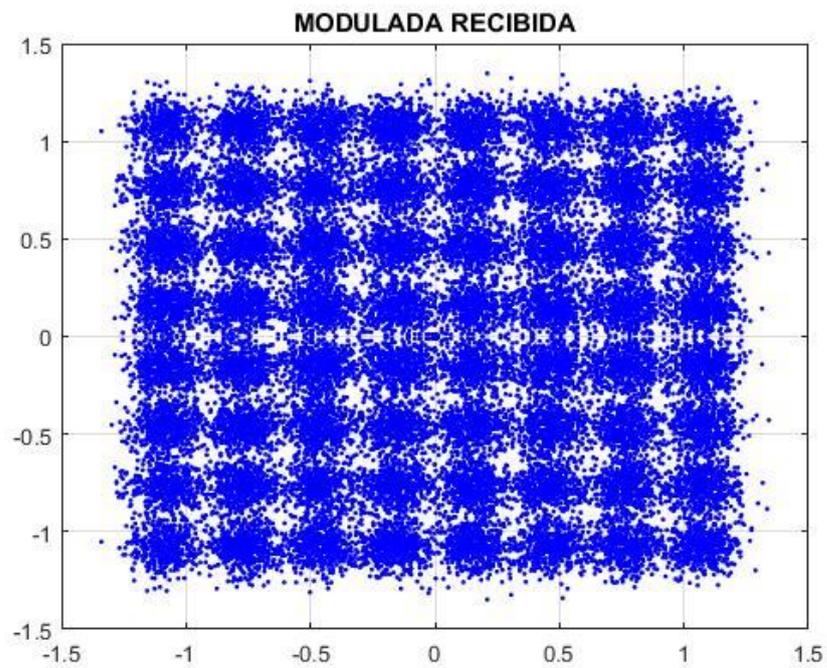


Figura 19. Señal Rx con V-RLS (SNR=10 dB, BER= 7.87%)

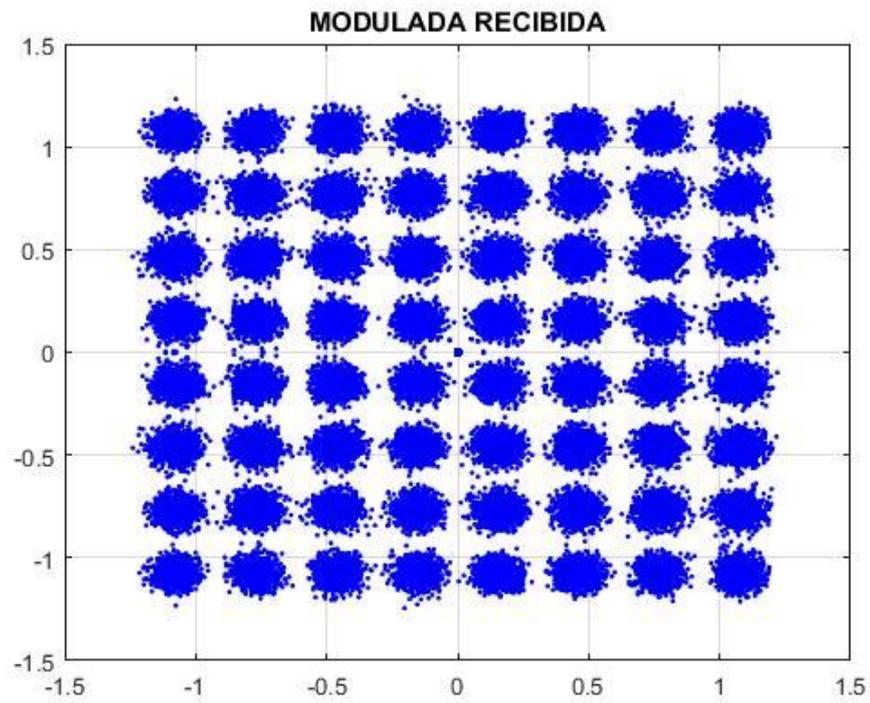


Figura 20. Señal Rx con V-RLS (SNR=20 dB, BER= 0.000066121%)

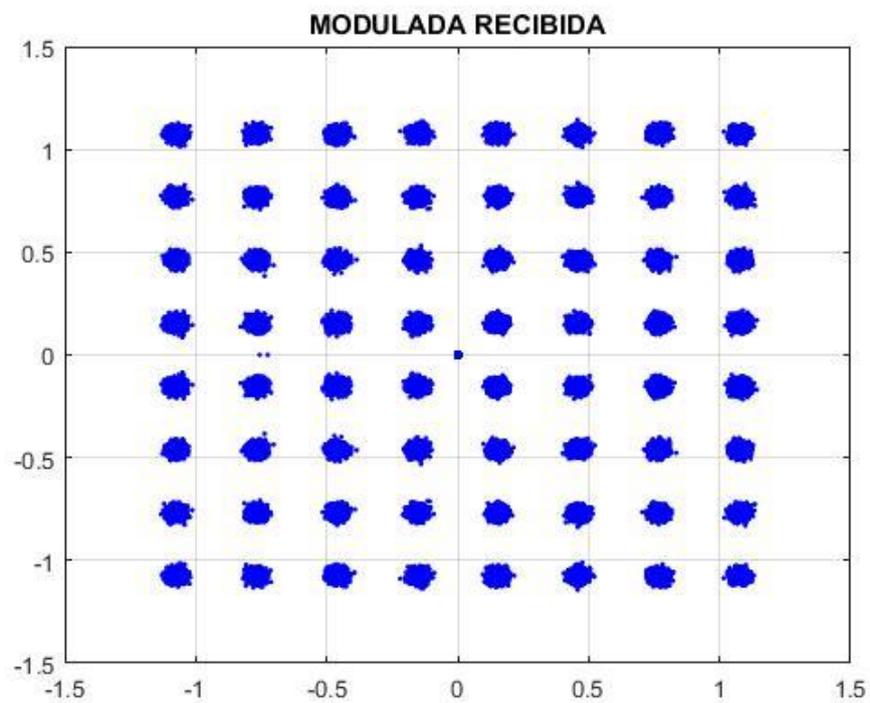


Figura 21 Señal Rx con V-RLS (SNR=30 dB, BER= 0%)

4.3. Resultados del BER

4.3.1. BER de Volterra NLMS

Para esta comparación se tomó 6 medidas de relación señal-ruido para observar el comportamiento del algoritmo, como se puede ver obviamente al aumentar el SNR el porcentaje de error disminuye.

Tabla 7.
Variación de SNR de V-NLMS

NLMS con $\delta=0.0001$ y coeficiente=0.01999	
SNR	BER
5	0.1775
10	0.0905
15	0.0260
20	0.0049
25	0.0013
30	7.5785e-04

4.4. BER de Volterra LMS

Como se puede observar en comparación con la Tabla 7, la Tabla 8 posee una mejor respuesta en convergencia como en BER por lo tanto el algoritmo V-RLS es más eficiente que el algoritmo V-NLMS.

Tabla 8.
Variación de SNR de V-LMS

LMS con $\mu_1 = 0.069$, $\mu_2 = 0.068$, $\mu_3 = 0.067$	
SNR	BER
5	0.1771
10	0.0875
15	0.0247
20	0.0050
25	0.0660
30	0.0619

4.5. BER de Volterra RLS

Sin duda frente a los resultados mostrados tanto en la Tabla 7 y 8, en la Tabla 9 se puede apreciar un porcentaje menor de errores en la transmisión. Por lo tanto el mejor algoritmo utilizado es el RLS.

Tabla 9. Variación de SNR de V-RLS

RLS con $\lambda=0.9999989$	
SNR	BER
5	0.1723
10	0.0787
15	0.0118
20	6.6121e-05
25	3.6621e-04
30	0

CAPÍTULO 5

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Después de realizar las comparaciones pertinentes se concluye que el algoritmo V-RLS tiene una mejor respuesta tanto en convergencia en relación al tiempo como en el BER, con una ventaja muy evidenciable tanto grafica como numéricamente.
- Se observó una mejor visualización de la constelación en el cambio de 10 a 15 dB en el SNR en los 3 algoritmos.
- Los algoritmos V-NLMS y V-LMS tiene una similitud por su tiempo de respuesta ante la convergencia y por lo tanto necesita mucho más muestras para la convergencia, debido a esto no es eficiente en comparación al V-RLS.
- El polinomio de tercer orden y cinco retrasos fue el que tuvo un mejor comportamiento para realizar el filtraje y su compensador para la recepción de la señal, el cambio de orden no influye más que en el tiempo de procesamiento.
- Por otro lado el número de retrasos mejora o empeora y para este estudio cinco retrasos fue el que mejor se comportó.
- Para el algoritmo V-RLS el coeficiente λ cambia mucho su reacción a la convergencia cambiando el 3er y 4to decimal pues si cambiamos los dos primeros el algoritmo nunca converge, para este estudio $\lambda=0.9999989$ es el valor que mejor se comportó.
- El algoritmo V-LMS es un algoritmo muy simple pero posee una baja convergencia por lo que serviría para un ambiente invariable debido a que si existe un cambio utiliza mucho costo computacional.
- Los coeficientes de los algoritmos al tomar variaciones en su tercer y cuarto digito provocan cambios bastante radicales por lo tanto los valores tomados fueron analizados para cada uno.

- Se logró demostrar que esa diferencia de 9 a 10 dB en el MSE del algoritmo V-RLS hace que sea el mejor ya que su pérdida de bits en porcentaje llega a 0 con un SNR de 30 dB.
- El algoritmo V-RLS al pasar entre los 15 a 20 dB del SNR sus valores porcentuales de pérdida tienden a 0, por lo tanto este algoritmo queda demostrado que es el mejor para la transmisión de datos.

5.2. Recomendaciones

- Para un ambiente de gran variación se recomienda utilizar el algoritmo V-RLS debido a su gran velocidad de respuesta frente a cambios bruscos en el ambiente de estudio.
- Para utilizar el codificador de canal se debe tomar en cuenta el uso de OFDM para poder realizar un estudio exacto, debido al cambio de longitud al utilizar un codificador de canal.
- Para el uso de los algoritmos V-NLMS y V-LMS van muy bien ante estudios invariantes es decir que no tengas cambios fuertes debido a su baja reacción antes cambios.

5.3. Trabajos Futuros

- Se desea implementar una grabación de voz para realizar todo este estudio en un ambiente real, debido a que esto es una simulación ideal donde no se toma en cuenta varios factores que posee una comunicación inalámbrica real.
- La transmisión de imágenes sería interesante ya que en este estudio se utilizó una modulación 64-QAM la cual nos ayudaría a alcanzar una alta definición, por lo tanto realizar un estudio con estos algoritmos y ver su comportamiento sería una buena contribución.
- Realizar una transmisión de datos más extensa con un equipo mejor de procesamiento ya que este estudio se realizó con una computadora con procesador i3.

Bibliografía

- Alcocer, C. (2016). Red Inalámbrica. (págs. 3-6). RF Essentials, Jean-Jacques Deslise. Technical Editor.
- Andrew, J. (1998). *CDMA Principles of Spread Spectrum Communication*. New York.
- Bernal, P. (2005). Transformada Wavelet aplicando filtros de Volterra XXII., (pág. 78). Rio de Janeiro.
- Chicas, W. L. (2004). Diseño de un filtro digital adaptativo como cancelador de ruido basado en el algoritmo LMS., (pág. 139). San Salvador.
- Crespo, C. (2015). A widely nonlinear approach to compensate impairments in I/Q modulators., (págs. 506-509). Paris.
- Diniz, P. (2002). Adaptive Filtering: Algorithms and Practical Implementations. *Kluwer Academic Publishers*. Boston.
- Espinosa, P., & Villaroel, C. (2007). Proposición y simulación de un algoritmo adaptativo para antenas inteligentes. *Ingeniare. Revista chilena de ingeniería*, 344-350.
- Gharaibeh, M. (2012). *Nonlinear distortion wireless system*. Jordan: WILEY.
- Haykin, S. (1996). Adaptive filter theory. *Englewood-Cliffs*. New Jersey.
- Mezoa, M. (2011). Codificación de Canal. *Codificación de canal: código hamming y código convolucional* (págs. 2-23). OpenStax-CNX.
- Monro, S., & Robbins, H. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, (págs. 400-407).
- Oppenheim, A. (1996). Signals and Systems. Mexico: Prentice-Hall.
- Schetzen, M. (1980). The Volterra and Wiener Theories of Nonlinear Systems. Estados Unidos.
- Silva, C., Moulthrop, A., & Muha, M. a. (2001). Application of polyspectral techniques to nonlinear modeling and compensation. *In IEEE MTT-S* (págs. 13-16). Int Microwave Symp.
- Tan, L. (2001). Adaptive Volterra filters for active control of nonlinear noise processes. *IEEE Transactions on Signal Processing*, (págs. 1667 - 1676). USA.
- The MathWorks, Inc. (s.f.). *MathWorks*. Obtenido de Texture Analysis: <https://www.mathworks.com/help/images/texture-analysis.html>