

# APLICACIÓN DE UN FILTRO ADAPTATIVO PARA MINIMIZAR EL RUIDO EN UNA IMAGEN



**Jhon Fredy Ortega Muñoz**  
**Carlos Eduardo Rivera Valencia**

*UNIVERSIDAD DEL CAUCA*  
**FACULTAD DE INGENIERÍA ELECTRÓNICA Y  
TELECOMUNICACIONES**  
**DEPARTAMENTO DE TELECOMUNICACIONES**  
**GRUPO DE INVESTIGACIÓN Y DESARROLLO NUEVAS  
TECNOLOGÍAS EN TELECOMUNICACIONES-GNTT**  
**POPAYAN**  
**2015**

# **APLICACIÓN DE UN FILTRO ADAPTATIVO PARA MINIMIZAR EL RUIDO EN UNA IMAGEN**



Trabajo de grado presentado como requisito para obtener el título de ingeniero el Electrónica y Telecomunicaciones

**Jhon Fredy Ortega Muñoz  
Carlos Eduardo Rivera Valencia**

Director  
PhD. Pablo Emilio Jojoa Gómez

*UNIVERSIDAD DEL CAUCA*  
**FACULTAD DE INGENIERÍA ELECTRÓNICA Y  
TELECOMUNICACIONES  
DEPARTAMENTO DE TELECOMUNICACIONES  
GRUPO DE INVESTIGACIÓN Y DESARROLLO NUEVAS  
TECNOLOGÍAS EN TELECOMUNICACIONES-GNTT  
POPAYAN  
2015**

# TABLA DE CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO 1 .....	3
PROCESAMIENTO DIGITAL DE IMÁGENES .....	3
1.1. INTRODUCCIÓN .....	3
1.2. IMAGEN DIGITAL .....	3
1.2.1. IMÁGENES BINARIAS .....	4
1.2.2. IMÁGENES A ESCALA DE GRISES .....	5
1.2.3. IMÁGENES RGB .....	6
1.2.4. TAMAÑO DE LA IMAGEN DIGITAL.....	7
1.2.5. RESOLUCIÓN DE LA IMAGEN DIGITAL .....	7
1.3. RUIDO EN IMÁGENES DIGITALES .....	8
1.3.1. RUIDO GAUSSIANO .....	8
1.4. FILTROS ESPACIALES .....	9
1.4.1. FILTRO DE LA MEDIA .....	11
1.4.2. FILTRO DE LA MEDIANA.....	13
CAPÍTULO 2 .....	15
FILTRAJE ADAPTATIVO.....	15
2.1. INTRODUCCIÓN.....	15
2.2. FILTROS ADAPTATIVOS.....	16
2.2.1. IDENTIFICACIÓN.....	18
2.2.2. MODELADO INVERSO .....	19
2.2.3. PREDICCIÓN .....	19
2.2.4. CANCELACIÓN DE INTERFERENCIAS .....	20
2.3. ALGORITMO LMS.....	22
2.4. ALGORITMO NLMS.....	24
2.5. ALGORITMO AR <sub>y</sub> .....	25
CAPÍTULO 3 .....	27
MODELADO, SIMULACIÓN E IMPLEMENTACIÓN .....	27
3.1. INTRODUCCIÓN.....	27

3.2. METODOLOGÍA DE SIMULACIÓN.....	27
3.2.1. DEFINICIÓN DEL SISTEMA: .....	28
3.2.2. ANÁLISIS DEL SISTEMA: .....	29
3.2.3. FORMULACIÓN DEL MODELO:.....	29
3.2.4. SELECCIÓN DEL LENGUAJE:.....	29
3.2.5. DISEÑO DEL SISTEMA:.....	29
3.2.6. SIMULACIÓN DEL SISTEMA: .....	36
3.2.7. EXPERIMENTACIÓN .....	37
CAPÍTULO 4 .....	39
EVALUACIÓN Y ANALISIS DE RESULTADOS .....	39
4.1. INTRODUCCIÓN.....	39
4.2. ANÁLISIS DE RESULTADOS.....	40
4.2.1. COMPARACIÓN ENTRE LECTURA VECTORIAL COMO COLUMNA, FILA Y LECTURA VECTORIAL COMO MATRIZ .....	40
4.2.2. CONFIGURACIÓN DE LOS PARÁMETROS ÓPTIMOS DEL FILTRO ADAPTATIVO CON EL ALGORITMO LMS .....	44
4.2.3. CONFIGURACIÓN DE LOS PARÁMETROS ÓPTIMOS DEL FILTRO ADAPTATIVO CON EL ALGORITMO NLMS .....	45
4.2.4. CONFIGURACIÓN DE LOS PARÁMETROS ÓPTIMOS DEL FILTRO ADAPTATIVO CON EL ALGORITMO AR <sub>γ</sub> .....	46
4.2.5. COMPARACIÓN DE LOS ALGORITMOS LMS, NLMS, AR <sub>γ</sub> .....	50
CAPÍTULO 5 .....	59
CONCLUSIONES Y TRABAJOS FUTUROS.....	59
5.1. INTRODUCCIÓN.....	59
5.2. CONCLUSIONES.....	59
5.3. TRABAJOS FUTUROS .....	60
REFERENCIAS BIBLIOGRÁFICAS .....	61
APÉNDICES.....	63

## LISTA DE FIGURAS

Figura 1. 1. Comparación entre formato mapa de bits (izquierda) y vectorial (derecha) [2].	4
Figura 1. 2. Ejemplo de una imagen binaria [3].	5
Figura 1. 3. Ejemplo de una imagen a escala de grises [3].	5
Figura 1. 4. Separación de canales RGB [6].	6
Figura 1. 5. Ejemplo de una imagen RGB [3].	6
Figura 1. 6. Imagen digital con dimensiones de 20 x 20 píxeles [8].	7
Figura 1. 7. Resolución de una imagen 10 x 10 y 20 x 20 por pulgada c/u [8].	7
Figura 1. 8. Imagen original (izquierda), imagen con ruido gaussiano (derecha).	8
Figura 1. 9. Imagen original (izquierda), varianza $s=0.025$ (centro), varianza $s=0.05$ (derecha) [9].	9
Figura 1. 10. Esquema general de la matriz de una máscara.	10
Figura 1. 11. Valor de los píxeles.	10
Figura 1. 12. Máscara 3x3.	11
Figura 1. 13. Resultado de la operación anterior.	11
Figura 1. 14. Máscara de convolución del filtro de media 3x3.	11
Figura 1. 15. Matriz 4x4 que se procesará.	12
Figura 1. 16. Área de la matriz que se quiere procesar.	12
Figura 1. 17. Resultado de la operación anterior.	13
Figura 1. 18. Resultado final del filtro de media.	13
Figura 1. 19. Área de la matriz que se quiere procesar.	14
Figura 1. 20. Resultado de la primera área procesada.	14
Figura 1. 21. Resultado final de la matriz tratada con el filtro de mediana.	14
Figura 2. 1. Estructura directa de un filtro adaptativo.	17
Figura 2. 2. Diagrama de bloques del sistema de identificación.	18
Figura 2. 3. Diagrama de bloques del sistema de modelado inverso.	19
Figura 2. 4. Diagrama de bloques del sistema de predicción.	20
Figura 2. 5. Diagrama de bloques del sistema de cancelación de interferencias.	20
Figura 2. 6. Diagrama del filtro transversal.	23
Figura 2. 7. Mecanismo de control de los coeficientes.	24
Figura 3. 1. Adaptación de la metodología usada en “simulación de sistemas de telecomunicaciones”.	28
Figura 3. 2. Sistema simulado.	30
Figura 3. 3. Cancelador de interferencias.	30
Figura 3. 4. Proceso para lectura de una imagen en escala de grises.	31
Figura 3. 5. Imagen .TIFF a escala de grises, 512 x 512.	32
Figura 3. 6. Imagen afectada por el ruido gaussiano.	33
Figura 3. 7. Arreglo matricial leído como columna.	34
Figura 3. 8. Arreglo matricial leído como fila.	34

Figura 3. 9. Arreglo matricial 3x3.....	35
Figura 3. 10. Diagrama de flujo del proceso de simulación.....	37
Figura 4. 1. Casos de estudio.....	40
Figura 4. 2. Error en la salida del filtro y el algoritmo AR $\gamma$ con la imagen leída a) como columna, b) como fila y c) como matriz 3x3.....	41
Figura 4. 3. Error en la salida del filtro y el algoritmo LMS con la imagen leída a) como columna, b) como fila y c) como matriz 3x3.....	42
Figura 4. 4. Error en la salida del filtro y el algoritmo NLMS con la imagen leída a) como columna, b) como fila y c) como matriz 3x3.....	43
Figura 4. 5. Grafica de error cuadrático en un tres dimensiones para diferentes valores de $m_1$ y $\gamma$ con $\alpha = 0.01$ .....	48
Figura 4. 6. Grafica de error cuadrático en dos dimensiones para diferentes valores de $m_1$ y $\gamma$ con $\alpha = 0.01$ . ....	48
Figura 4. 7. Graficas del error con la tendencia para a) algoritmo LMS, b) algoritmo AR $\gamma$ y c) algoritmo NLMS.....	52
Figura 4. 8. Graficas de tendencia para los tres algoritmos utilizados. ....	53
Figura 4. 9. Imagen filtrada para un $MSE = 8,8 \times 10 - 5$ con a) algoritmo LMS, b) algoritmo NLMS con un y c) algoritmo AR $\gamma$ . ....	55
Figura 4. 10. Imagen filtrada para un $MSE = 8,8 \times 10 - 5$ usando una parte de la imagen en la etapa de aprendizaje para los algoritmos a) LMS, b) NLMS y c) AR $\gamma$ .....	57
Figura 4. 11. Imágenes filtradas para los algoritmos a) LMS, b) NLMS y c) AR $\gamma$ afectada por ruido gaussiano en un sector.....	58

## LISTA DE TABLAS

Tabla 4. 1. Valores de la pérdida de información en las imágenes filtradas con los diferentes métodos de lectura. ....	44
Tabla 4. 2. Valores del parámetro $\mu$ para el algoritmo LMS.....	45
Tabla 4. 3. Valores del parámetro $\mu$ para el algoritmo LMS.....	46
Tabla 4. 4. Valores con los que se trabajó en las simulaciones. ....	47
Tabla 4. 5 Relación entre los parámetros $\alpha$ y $m_1$ del algoritmo AR $\gamma$ .....	49
Tabla 4. 6. Valores óptimos de los parámetros del algoritmo AR $\gamma$ . ....	50
Tabla 4. 7. Error cuadrático en un número de iteraciones fijas. ....	53
Tabla 4. 8. Número de iteraciones para llegar a un error cuadrático.....	54

## LISTA DE ACRÓNIMOS

<b>AAC</b>	Algoritmo Acelerador Complejo.
<b>AD:</b>	Analógico-Digital.
<b>AR<math>\gamma</math>:</b>	Algoritmo Acelerador Regresivo versión $\gamma$ .
<b>FIET:</b>	Facultad de Ingeniería Electrónica y Telecomunicaciones.
<b>GNTT:</b>	Grupo de investigación y desarrollo Nuevas Tecnologías en Telecomunicaciones.
<b>LMS:</b>	Least Mean Square, algoritmo de mínimos cuadrados medios.
<b>MSE:</b>	Mean Squared Error, Error Cuadrático Medio.
<b>ND:</b>	Número Digital.
<b>NLMS:</b>	Normalized Least Mean Square, algoritmo de mínimos cuadrados medios normalizados.
<b>px:</b>	Píxel.
<b>PPP:</b>	Píxeles Por Pulgada.
<b>RGB:</b>	Red, Green, Blue; Rojo, Verde, Azul.

# INTRODUCCIÓN

La imagen digital<sup>1</sup> representa información visual que está asociada a escenas ambientales en tiempo real, por ello el procesamiento digital de imágenes en la actualidad es usado en diversos campos del conocimiento, como astronomía, geografía, geología, biología y medicina; en este último se obtienen múltiples beneficios centrados en la salud y el bienestar humano.

Uno de los principales problemas que afectan a las imágenes en el momento de procesarlas es el ruido que puede presentarse de diversas maneras, como: el ruido gaussiano<sup>2</sup>. Existen filtros que ayudan a reducir el efecto que el ruido produce en la calidad de la imagen, como por ejemplo el filtro<sup>3</sup> de media<sup>4</sup> que se apoya en la vecindad del pixel<sup>5</sup> seleccionado buscando un promedio entre ellos para reducir el ruido, filtro de media ponderada<sup>6</sup> que funciona igual al filtro de media pero éste utiliza una matriz<sup>7</sup> de filtrado con sus elementos diferentes de 1, debido a ello el resultado será más parecido a la imagen original y el filtro de mediana que selecciona cada pixel de la imagen y lo reemplaza por la mediana de los pixeles vecinos, así el resultado del filtrado será una imagen con valores de pixel presentes en la imagen original mas no un promedio entre el pixel seleccionado y su vecindad, pero el limitante que tienen este tipo de filtros (media) es que utilizan una función lineal que trata de igual manera a todos los pixeles de la imagen, por lo cual se opta por utilizar un filtro adaptativo<sup>8</sup>, que se puede configurar por medio de algoritmos, algunos como el Algoritmo de Mínimos Cuadrados Medios (LMS, *Least Mean Square*), el Algoritmo de Mínimos Cuadrados Medios Normalizado (NLMS, *Normalized Least Mean Square*) y el Algoritmo Acelerador Versión  $\gamma$  (AR $\gamma$ ); con estos algoritmos el filtro adaptativo ofrece un tratamiento especial a cada pixel de la imagen en su procesamiento, con lo que se trata de llevar a cabo un mayor filtrado en zonas donde solo se presume que hay ruido y filtrar menos en aquellas zonas próximas a los bordes.

Cada vez se usan más las imágenes con una mayor exigencia en su calidad, por eso se hace necesario el uso de filtros de mayor complejidad que permitan reducir

---

<sup>1</sup> La imagen digital, es una representación bidimensional de una imagen a partir de una matriz numérica.

<sup>2</sup> Ruido gaussiano, es aquel que produce pequeñas variaciones de intensidad en todos los pixeles de la imagen, suele ser originado por componentes electrónicos.

<sup>3</sup> Circuito eléctrico o software de computadora que se aplica a un conjunto de datos ruidosos para extraer una cantidad de información previamente definida como la respuesta deseada o señal de interés.

<sup>4</sup> Filtro que toma cada pixel de la imagen y se reemplaza por la media de los pixeles vecinos.

<sup>5</sup> Es la menor unidad homogénea en color que forma parte de una imagen digital.

<sup>6</sup> Filtro de Media ponderada, los elementos de la matriz de filtrado no son todos 1 sino que se da más peso a uno de ellos.

<sup>7</sup> Representación ordenada de filas y columnas.

<sup>8</sup> Filtro adaptativo, es un dispositivo que intenta modelar la relación entre señales en tiempo real de manera iterativa.

los problemas generados por interferencias no deseadas tanto en la captura<sup>9</sup> como en su procesamiento, una de las soluciones que se está implementando para ello es el uso de filtros adaptativos que presentan ventajas con respecto a los filtros de coeficientes fijos ya que los primeros, por medio de algoritmos de adaptación permiten procesar señales desconocidas y variables.

Teniendo en cuenta lo anterior se hace necesario el estudio de algoritmos de adaptación como por ejemplo el LMS, NLMS, ARy, que permitan hacer uso del filtro adaptativo para disminuir el efecto que producen las señales no deseadas en el procesamiento digital de imágenes. En este contexto, se presenta en este documento el diseño y la implementación de un filtro adaptativo para procesar imágenes digitales y reducir el ruido presente en ellas.

A continuación se describe el contenido de los cinco capítulos que conforman el texto.

**Capítulo I:** Presenta el estudio de las imágenes digitales explicando su formato, los tipos que existen y su estructura, también el estudio del ruido Gaussiano que afecta a las imágenes, además, algunos filtros espaciales (media, mediana.).

**Capítulo II:** Explica conceptos sobre el filtraje adaptativo, además de las aplicaciones, factores y algunos algoritmos (LMS, NLMS, ARy) que se usan en los filtros para procesar las señales.

**Capítulo III:** Muestra el modelado, simulación e implementación del filtro adaptativo para el procesamiento sobre la imagen.

**Capítulo IV:** En este capítulo se explica el análisis de resultados obtenidos a través de la simulación.

**Capítulo V:** Expone las conclusiones del trabajo realizado, seguido de recomendaciones y trabajos futuros que pueden ser desarrollados a partir del realizado.

---

<sup>9</sup> Es una imagen digital tomada por un dispositivo electrónico por ejemplo cámaras digitales, teléfonos inteligentes.

# CAPÍTULO 1

## PROCESAMIENTO DIGITAL DE IMÁGENES

### 1.1. INTRODUCCIÓN [1]

El procesamiento digital de imágenes son un conjunto de técnicas que se aplican a las imágenes digitales con el fin de mejorar la calidad y/o facilitar la búsqueda de información en su estructura, dadas las exigencias hace su aparición en la historia de la computación tardíamente porque la tecnología que existía en ese entonces no era la adecuada para poder procesarla, por ello se necesitó primero desarrollar el hardware y software que permitiera soportar dichos procedimientos, después de esto se dio paso a los algoritmos y técnicas de optimización que permiten el procesamiento digital de imágenes, dichas funciones son muy sofisticadas y elaboradas, ahora bien, este proceso se hace con el fin de mitigar el efecto que el ruido produce en las imágenes. Con el pasar del tiempo se iban haciendo descubrimientos que permitieron contrarrestar el efecto del ruido en imágenes de manera más simple, uno de ellos fue el uso de los filtros espaciales que pueden ser lineales o no lineales, cabe destacar que los filtros lineales tratan de igual manera a todos los píxeles de la imagen, lo cual es una limitante porque en algunos casos no se necesita tanto filtrado en todas las zonas de la imagen por ejemplo en los bordes de ésta, evidenciando esta necesidad se desarrollaron los algoritmos que permiten que un filtro se adecue al sistema y se modifiquen los pesos o coeficientes del filtro para tener un resultado más parecido al original, el proceso de filtrado puede llevarse a cabo sobre los dominios de la frecuencia y/o espacio, en este trabajo de grado se trabajará en el dominio del espacio, realizando las operaciones de filtrado directamente sobre los píxeles.

### 1.2. IMAGEN DIGITAL [1] [2]

La imagen digital es una representación bidimensional de una escena de la realidad a partir de una matriz numérica, la imagen consiste en una colección ordenada de valores y está formada por un conjunto de puntos definidos llamados píxeles que se establecen en filas y columnas, entre mayor sea el número filas y columnas mayor será el tamaño y el detalle de la imagen; pueden tener un formato **vectorial** que son imágenes cuyos gráficos son formados a base de curvas y líneas a través de elementos geométricos definidos como vectores o por **mapa de bits** que son imágenes compuestas por una serie de puntos (píxeles) que

contienen información acerca del color y pueden ser capturadas por dispositivos de conversión Analógica a digital (AD, *Analogic to Digit*) como escáneres y/o cámaras digitales, para este trabajo de grado se utilizarán las imágenes con formato de mapa de bits, en la figura 1.1 se hace una comparación entre los formatos de las imágenes.



**Figura 1. 1.** Comparación entre formato mapa de bits (izquierda) y vectorial (derecha) [2].

En la actualidad existen tres tipos de imágenes en mapa de bits, pueden ser imágenes binarias, imágenes a escala de grises o imágenes Rojo, Verde y Azul (RGB). A continuación se explican las características de los tipos de imágenes ya mencionados.

#### 1.2.1. IMÁGENES BINARIAS [3]

Son representaciones de una imagen digital que presenta únicamente dos valores en cada píxel de su estructura, puede tomar el valor de 0, lo cual significa que tomara el color negro o puede tomar el valor de 1, lo cual indica que tomará el color blanco, en la figura 1.2 se muestra un ejemplo de una imagen binaria.

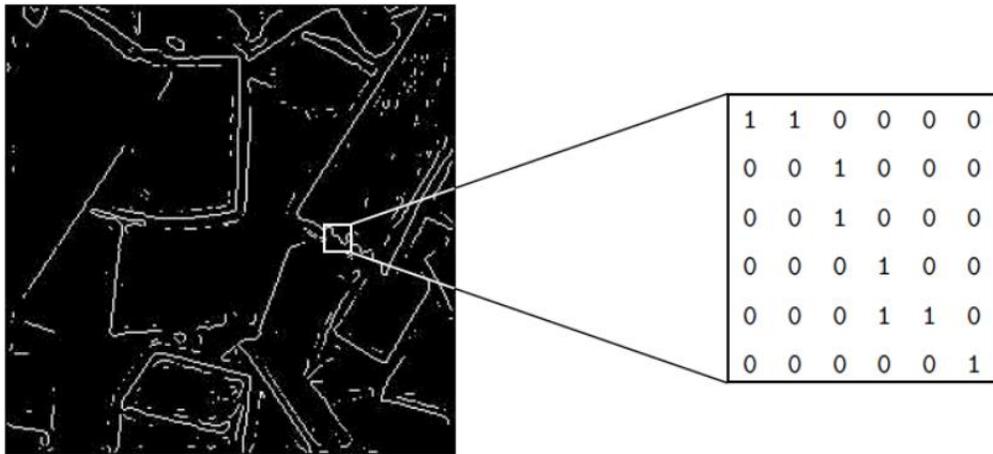


Figura 1. 2. Ejemplo de una imagen binaria [3].

### 1.2.2. IMÁGENES A ESCALA DE GRISES [3] [4]

Son representaciones de imágenes que emplean una escala en la que cada píxel posee un valor equivalente a una graduación de gris o bien cada píxel se dibuja usando un valor numérico individual que representa su luminancia, esta escala se extiende entre el color negro que posee un valor de 0 y el color blanco que posee un valor de 255, esto indica que cada píxel puede ser representado por 1 byte (8 bits) y puede tomar  $2^8$  valores diferentes como se muestra en la figura 1.3.



Figura 1. 3. Ejemplo de una imagen a escala de grises [3].

### 1.2.3. IMÁGENES RGB [3] [5]

Son representaciones de imágenes que utilizan tres canales para constituir una imagen a color, estos canales son los mostrados en la figura 1.4, cada píxel de cada campo puede tomar  $2^8$  valores que se extienden desde 0 a 255, es decir la imagen RGB puede tomar  $2^{24}$  valores diferentes, dicha imagen está compuesta por tres imágenes, una por cada canal, esto para una imagen con píxeles de intensidad de brillo convencional, para una imagen de alta resolución se trabaja con 48 bits, es decir cada canal tiene  $2^{16}$  valores diferentes para representar el color de la imagen, en la figura 1.5 se muestra un ejemplo de una imagen RGB.



Figura 1. 4. Separación de canales RGB [6].

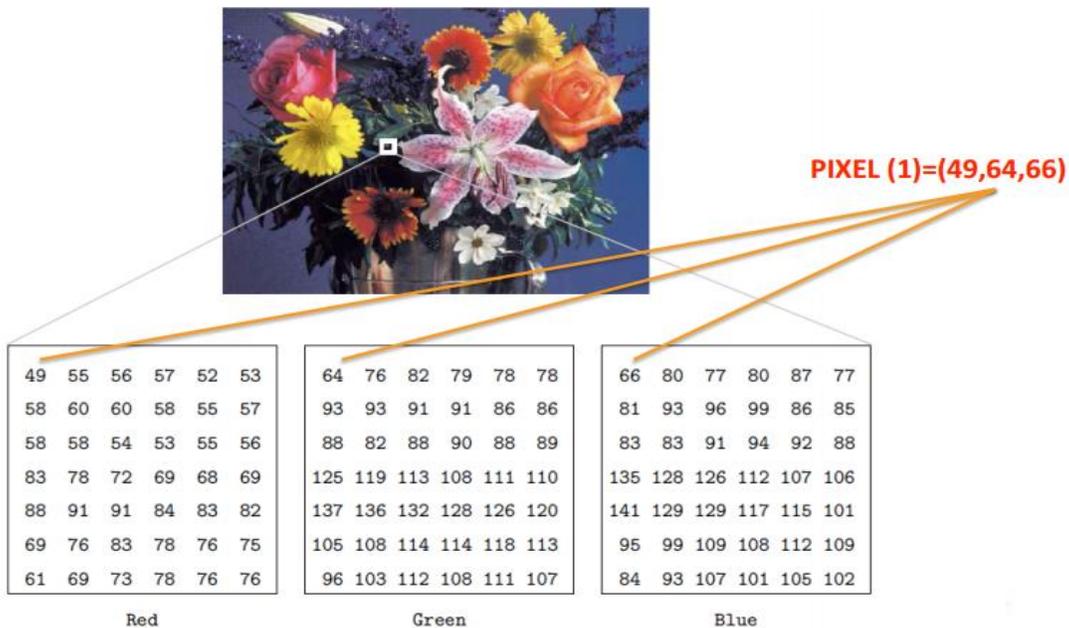


Figura 1. 5. Ejemplo de una imagen RGB [3].

#### 1.2.4. TAMAÑO DE LA IMAGEN DIGITAL [6] [7]

El tamaño de la imagen se define con las dimensiones de la matriz o cuadrícula y está dada en píxeles, su representación en la figura 1.6; este tamaño da las dimensiones de la imagen (ancho x alto).

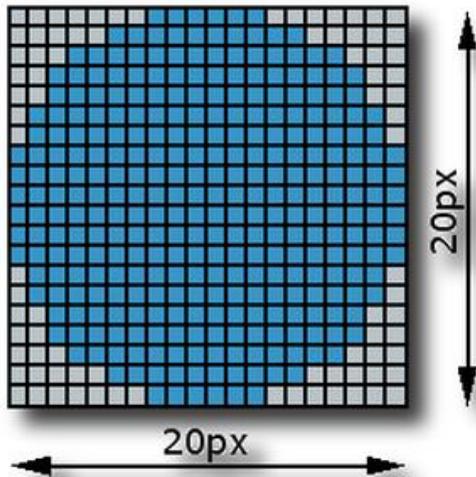


Figura 1. 6. Imagen digital con dimensiones de 20 x 20 píxeles [8].

#### 1.2.5. RESOLUCIÓN DE LA IMAGEN DIGITAL [6] [7]

La resolución de la imagen es definida como el producto entre el ancho y el alto de una imagen digital dado en píxeles o también como la medida de cantidad de píxeles por unidad de longitud, frecuentemente Píxeles Por Pulgada (PPP), figura 1.7, con lo que se observa que con mayor resolución en la imagen se presenta mayor detalle en la misma.

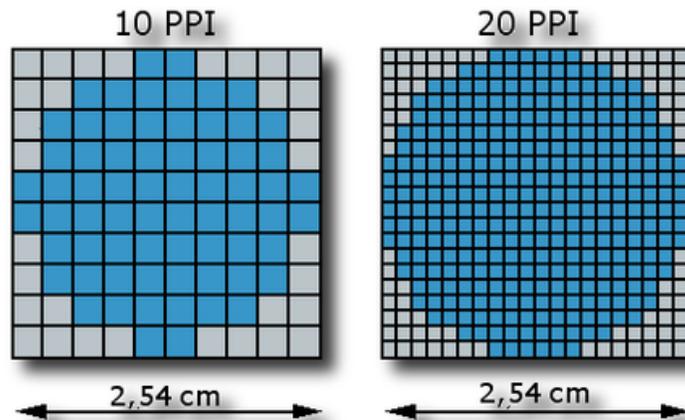


Figura 1. 7. Resolución de una imagen 10 x 10 y 20 x 20 por pulgada c/u [8].

### 1.3. RUIDO EN IMÁGENES DIGITALES [8] [9]

El ruido en una imagen aparece de manera imprevista y es muy frecuente, este fenómeno se describe como la variación aleatoria del brillo o el color en ellas, en el caso específico de las imágenes digitales se puede definir como píxeles erróneos que se mezclan con los píxeles aceptables que componen la imagen disminuyendo así su calidad, el ruido puede introducirse en una imagen de diversas maneras, todo depende de cómo se obtenga la imagen, por ejemplo si la imagen se adquiere en formato digital directamente, la señal puede ser dañada por medio del ruido presente en los dispositivos electrónicos producido por el receptor en las primeras etapas de amplificación, porque en estas etapas el valor de la señal recibida es bajo y cualquier ruido producido posee un valor notable frente a la señal recibida (ruido interno), a continuación se explica cómo afecta el ruido gaussiano a las imágenes digitales, debido este tipo de ruido es una función ideal que se considera una buena entrada a un sistema adaptativo por su semejanza con muchos casos reales.

#### 1.3.1. RUIDO GAUSSIANO [8] [9]

El ruido gaussiano es un tipo de ruido que suele ser producido por componentes electrónicos como por ejemplo sensores, digitalizadores, etc. Produce pequeñas variaciones en la imagen y como es el resultado del uso de la energía, este ruido afecta a todos los píxeles por ende toda la estructura de la imagen será afectada como se observa en la figura 1.8.



**Figura 1. 8.** Imagen original (izquierda), imagen con ruido gaussiano (derecha).

La relación señal a ruido se estima en una imagen por medio de la varianza; la figura 1.9 muestra dicha influencia.

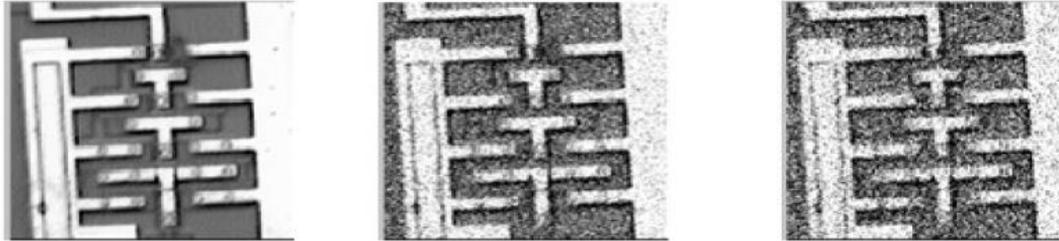


Figura 1. 9. Imagen original (izquierda), varianza  $s=0.025$  (centro), varianza  $s=0.05$  (derecha) [9].

#### 1.4. FILTROS ESPACIALES [10]

El procesamiento por grupo de píxeles, opera sobre un grupo de píxeles de entrada que rodean a un píxel central. Por lo tanto, la vecindad<sup>10</sup> del píxel central posee información importante sobre la tendencia del brillo del área en la que se está procesando, el uso y análisis de dichas tendencias presentadas en una imagen son las que motivan las operaciones de filtrado espacial, los detalles que se marcan en ellas son muy acentuados y otros no tanto, los detalles se comprenden de transiciones de brillo que van del oscuro al claro y vuelta al oscuro. Una imagen puede filtrarse para acentuar o eliminar una banda de frecuencias espaciales altas o bajas, este tipo de operaciones de procesamiento digital de imágenes se conocen como filtrado espacial o filtros en el dominio del espacio.

Para procesar este tipo de elementos se recurre al uso de máscaras espaciales que son matrices de coeficientes de tamaño  $m \times n$  que constituyen los coeficientes del filtro con los cuales se recorre toda la imagen centrando las operaciones sobre los píxeles que se encuentran en las áreas donde los píxeles de la imagen coinciden con los píxeles de la máscara.

Los filtros espaciales responden a la ecuación 1.1.

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t), \quad (1.1)$$

donde

$f(x + s, y + t)$ , son los valores de los píxeles seleccionados.

$w(s, t)$ , son los coeficientes que se aplicarán al bloque (máscara).

Siendo la matriz del bloque  $st$ .

---

<sup>10</sup> Grupo de píxeles que se encuentran alrededor de un píxel central.

$s$  (filas):  $2a + 1$

$t$  (columnas):  $2b + 1$

La figura 1.10 muestra el esquema general de la matriz de una máscara usada por los filtros espaciales para procesar imágenes; la expresión  $g(x,y)$  depende de los coeficientes  $w(s,t)$  contenidos en la máscara para dar un resultado final, las variables  $s$  y  $t$  son las filas y columnas de la máscara utilizada que rodeará al pixel seleccionado.

$(-a,-b)$	$(-a, -b+1)$	$(-a,-b+2)$	...	$(-a,b-2)$	$(-a,b-1)$	$(-a,b)$
$(-a+1,-b)$	$(-a+1, -b+1)$	$(-a+1,-b+2)$	...	$(-a+1,b-2)$	$(-a+1,b-1)$	$(-a+1,b)$
$(-a+2,-b)$	$(-a+2, -b+1)$	$(-a+2, -b+2)$	...	$(-a+2,b-2)$	$(-a+2,b-1)$	$(-a+2,b)$
...	...	...	...	...	...	...
$(a-2,-b)$	$(a-2,-b+1)$	$(a-2, -b+2)$	...	$(a-2,b-2)$	$(a-2,b-1)$	$(a-2,b)$
$(a-1,-b)$	$(a-1,-b+1)$	$(a-1, -b+2)$	...	$(a-1,b-2)$	$(a-1,b-1)$	$(a-1,b)$
$(a,-b)$	$(a,-b+1)$	$(a,-b+2)$	...	$(a,b-2)$	$(a,b-1)$	$(a,b)$

Figura 1. 10. Esquema general de la matriz de una máscara.

Para realizar operaciones basadas en vecindades se deben tener en cuenta las siguientes consideraciones:

- Se selecciona el píxel que se va a procesar.
- Se selecciona el entorno de dicho píxel.
- Se aplica una función que depende del valor de los píxeles del entorno seleccionado.
- Se reemplaza el píxel de la imagen de salida equivalente al píxel de la imagen de entrada, por el valor devuelto por la función.

Ejemplo: En el siguiente ejemplo se muestra un tipo de operación entre los píxeles de una imagen y una máscara de tamaño 3x3.

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

Figura 1. 11. Valor de los píxeles.

1	2	1
2	4	2
1	2	1

Figura 1. 12. Máscara 3x3.

$$g(2,4) = 1(1) + 8(2) + 15(1) + 7(2) + 14(4) + 16(2) + 13(1) + 20(2) + 22(1)$$

$$g(2,4) = 209.$$

Con lo que se tiene como resultado final:

X	X	X
X	209	X
X	X	X

Figura 1. 13. Resultado de la operación anterior.

#### 1.4.1. FILTRO DE LA MEDIA [10]

Es un filtro pasa bajo que tiene por objetivo suavizar los contrastes espaciales presentes en una imagen, un filtro espacial de paso bajo tiene como función dejar pasar o mantener intactas los componentes de frecuencias bajas en una imagen y por otra parte atenuar los componentes de frecuencias altas, este filtro responde a la ecuación 1.2 mostrada en la ecuación 1.2.

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b \frac{1}{mn} (f(x + s, y + t)) \quad (1.2)$$

Con  $m = n$ .

Uno de los elementos más utilizados en la filtración de componentes de frecuencia baja es el filtro de media que utiliza una máscara de convolución con dimensiones 3x3 con sus nueve coeficientes iguales como se muestra en la figura 1.11:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Figura 1. 14. Máscara de convolución del filtro de media 3x3.

Esta máscara produce el promedio entre los valores de los píxeles que se están tratando, la suma de sus coeficientes es 1 y todos son valores positivos, las características anteriores son válidas para los filtros pasa bajos, ahora, si se desea operar sobre una región donde el valor del brillo de los píxeles tienen el mismo valor, es decir, un área de baja frecuencia espacial, dará como resultado el mismo valor de brillo, dicho de otra manera el resultado de una región cuyo brillo es constante será el mismo, en cambio si se aplica la máscara en la regiones de altas frecuencias espaciales, donde los cambios de brillos de los píxeles cambian rápidamente de blanco al negro y viceversa, el resultado será un valor medio de grises entre los blancos y negros, con este proceso se obtiene una imagen con valores de grises que van variando lentamente.

Ejemplo: En este ejemplo se explica el mecanismo que usa un filtro de media para procesar una matriz 4x4 por medio de la máscara de coeficientes 1/9 de tamaño 3x3.

20	23	30	31
22	21	29	30
23	24	32	33
29	31	34	37

Figura 1. 15. Matriz 4x4 que se procesará.

- Se tomará un área de igual tamaño que la máscara para realizar el procedimiento de filtrado y se realiza la sumatoria de los productos entre el valor de los píxeles y los coeficientes de la máscara.

20	23	30	31
22	21	29	30
23	24	32	33
29	31	34	37

Figura 1. 16. Área de la matriz que se quiere procesar.

- Se toma el área seleccionada y se realiza el procedimiento para el píxel que se encuentra en la posición  $(x,y) = (2,2)$ .

$$g(2,2) = \sum_{s=1}^3 \sum_{t=1}^3 \frac{1}{3 \times 3} (f(2,2) \frac{1}{9})$$

$$g(2,2) = 20(1/9) + 23(1/9) + 30(1/9) + 22(1/9) + 21(1/9) + 29(1/9) + 23(1/9) + 24(1/9) + 32(1/9),$$

$$g(2,2) = \mathbf{24,8}.$$

Quedando por lo tanto:

X	x	x	X
X	24,8	X	x
X	X	X	x
x	X	x	X

Figura 1. 17. Resultado de la operación anterior.

Así mismo para el resto de los píxeles de la matriz:

$$g(2,3) = 23(1/9) + 30(1/9) + 31(1/9) + 21(1/9) + 29(1/9) + 30(1/9) + 24(1/9) + 32(1/9) + 33(1/9) = \mathbf{28,1}$$

$$g(3,2) = 22(1/9) + 21(1/9) + 29(1/9) + 23(1/9) + 24(1/9) + 32(1/9) + 29(1/9) + 31(1/9) + 34(1/9) = \mathbf{27,2}$$

$$g(3,3) = 21(1/9) + 29(1/9) + 30(1/9) + 24(1/9) + 32(1/9) + 33(1/9) + 31(1/9) + 34(1/9) + 37(1/9) = \mathbf{30,1}$$

X	x	x	X
X	24,8	28,1	x
X	27,2	30,1	x
x	X	x	X

Figura 1. 18. Resultado final del filtro de media.

#### 1.4.2. FILTRO DE LA MEDIANA [10]

Existe otro tipo de filtro usado para tratar el ruido en una imagen pero se diferencia del anterior debido a que no es función lineal del valor del brillo de los píxeles pues no se calculan como una suma lineal de los elementos multiplicados por los coeficientes fijos de una máscara, a este tipo de filtros se les conoce como filtros espaciales no lineales, éste tipo de filtros tienen técnicas que operan sobre un núcleo de píxeles que rodean a un píxel central de entrada, pero la diferencia radica que en vez de utilizar un promedio de los elementos presentes en la matriz de valor y máscara, combinan el valor del brillo del grupo de píxeles. Para el caso

del filtro de mediana, su principal ventaja radica en que el valor final del píxel que se presenta en su salida es un valor presente real en la imagen y no un promedio.

Ejemplo: a continuación se muestra el mecanismo usado por el filtro de mediana para reducir el efecto que el ruido produce en una imagen, para ello se analizan campos de tamaño  $3 \times 3$  en una matriz de tamaño  $4 \times 4$ .

- Los que se debe hacer es ordenar los números presentes en el área que se desea analizar, después tomar el valor de la mediana y reemplazarlo por ese valor de píxel.

20	23	30	31
22	21	29	30
23	24	32	33
29	31	34	37

Figura 1. 19. Área de la matriz que se quiere procesar.

- Mediana de  $\{20, 21, 22, 23, \mathbf{23}, 24, 29, 30, 32\} = 23$ .

X	x	x	X
X	23	X	x
X	X	X	x
x	X	x	X

Figura 1. 20. Resultado de la primera área procesada.

Así mismo para el resto de la matriz:

- Mediana de  $\{21, 23, 24, 29, \mathbf{30}, 30, 31, 32, 33\} = 30$ .
- Mediana de  $\{21, 22, 23, 24, \mathbf{29}, 29, 31, 32, 34\} = 29$ .
- Mediana de  $\{21, 24, 29, 30, \mathbf{31}, 32, 33, 34, 37\} = 31$ .

La figura 1.19, muestra el resultado final de la matriz que se trató, como se ve, hay campos que no se pueden llenar, porque se requieren el número total de elementos y cuando se tratan los píxeles que se encuentran en los bordes, dicha área sobrepasa los límites establecidos, el inconveniente con este tipo de filtros es que en las zonas próximas al borde se pierde información dependiendo del área que se desee analizar.

X	x	x	X
X	23	30	x
X	29	31	x
x	X	x	X

Figura 1. 21. Resultado final de la matriz tratada con el filtro de mediana.

# CAPÍTULO 2

## FILTRAJE ADAPTATIVO

### 2.1. INTRODUCCIÓN

En el capítulo anterior se habló de la imagen, su estructura y los tipos que hay, además se explicó el problema que se tiene en el momento de procesarla, que consiste en información no deseada que puede agregarse, y por último se mencionó una manera de tratar la información presente en la imagen, conocida como filtraje espacial, con esto se expusieron dos maneras de llevar a cabo este procedimiento para filtrar los valores de brillo de los píxeles que tienen estos elementos, los filtros lineales y no lineales, pero existen ciertas limitantes con estos dos filtros, una es que el filtro de media trata de manera lineal a todos los componentes de la imagen, es decir las trata de igual manera, y con el filtro de mediana se tiene el inconveniente de organizar los valores de los píxeles de menor a mayor y seleccionar la mediana de este grupo de valores, además de estos inconvenientes, existe otro que afecta a los dos filtros y es la detección de bordes, lo que quiere decir que la información que se encuentra próxima a los bordes se pierde en mayor o menor cantidad, dependiendo del tamaño de la máscara, evidenciando las limitantes que existen con este tipo de filtros se decidió estudiar el filtraje adaptativo para implementarlo en las imágenes, en este capítulo se realizará el estudio acerca de los filtros adaptativos, su estructura, tipo y algunos de los algoritmos que se usan en este tipo de filtros para llevar a cabo el filtraje.

Para procesar una señal y reducir el ruido que está presente en ella se puede usar un filtro para realizar tres tareas básicas:

- **Filtrado:** Consiste en extraer la información deseada de una señal en un tiempo  $t$  y usarla al mismo tiempo para mostrarla en la salida sin ruido.
- **Suavizado:** Es un procedimiento parecido al proceso de filtrado, pero la diferencia radica en que a veces la información deseada no se encuentra disponible y por esto se utiliza información extraída en otro tiempo  $t$ , por ello se produce un retardo en la salida.

- **Predicción:** La predicción es un tratamiento preventivo que se le hace a la señal que contiene la señal de interés, el objetivo de esto es obtener la cantidad de información que será enviada en algún momento  $t + \tau$ , con  $\tau > 0$ .

Cuando se tienen parámetros estadísticos se busca una solución aproximada para el problema de filtrado lineal, el objetivo es diseñar un filtro capaz de tomar los datos ruidosos de entrada y minimizar el efecto del ruido a la salida del filtro para minimizar el valor cuadrático medio de la señal de error que se define como la diferencia entre la señal primaria<sup>11</sup> y la salida actual del filtro; cuando hay entradas estacionarias, la solución más adecuada son los filtros de Wiener<sup>12</sup> que son óptimos para reducir el error cuadrático medio.

Cuando existen señales dinámicas no estacionarias, es necesaria la adaptación del filtro a los cambios bruscos de la señal de entrada, por otra parte el filtro de Wiener solo se aplica a señales estacionarias cuyas propiedades estadísticas son conocidas, por esta razón se diseña un filtro adaptativo con el propósito de no depender de esta hipótesis, dado que el filtro adaptativo por medio de los algoritmos adaptativos, como por ejemplo *Least Mean Square* (LMS), *Normalized Least Mean Square* (NLMS), Algoritmo Acelerador versión  $\gamma$  (AR $\gamma$ ), permiten actualizar sus coeficientes con el fin de minimizar el error cuadrático medio y obtener una respuesta que se asimile bastante a la solución del filtro Wiener. La velocidad con que convergen estos algoritmos depende mucho de la implementación que se realice además del tipo de señales con las que se estén trabajando.

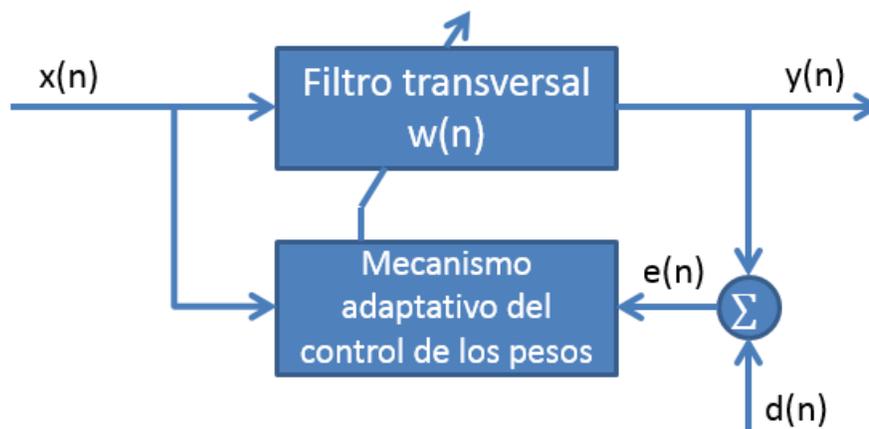
## 2.2. FILTROS ADAPTATIVOS [11] [12]

Un filtro adaptativo se define como un filtro que intenta modelar la relación entre dos señales de manera iterativa en tiempo real, a diferencia de los filtros digitales convencionales cuyos coeficientes son invariantes en el tiempo, los filtros adaptativos poseen la estructura que se muestra en la figura 2.1.

---

<sup>11</sup> Dependiendo de la configuración con la que se trabaje puede ser señal original + ruido, señal deseada, señal deseada retardada.

<sup>12</sup> Los filtros de Wiener son los mejores filtros lineales de mínimos cuadrados, que pueden ser usados para predicción, estimación, interpolación, filtrado de señal y ruido, etc.



**Figura 2. 1.** Estructura directa de un filtro adaptativo.

Al algoritmo adaptativo llegan dos señales; una es la señal de entrada  $x(n)$  y la otra es la señal de error  $e(n)$ , por medio de esta última el algoritmo puede actualizar los coeficientes del mismo debido a que esta señal proviene de la diferencia entre la señal deseada  $d(n)$  y la señal de salida del filtro  $y(n)$ , el objetivo del algoritmo adaptativo, en el caso de una configuración de identificación de sistema es que esta diferencia sea o tienda a cero; para obtener la salida  $y(n)$  se multiplican la señal de entrada por los coeficientes del filtro denotados por  $w(n)$ , como se describen en las ecuaciones 2.1 y 2.2.

$$y(n) = w(n)x(n), \quad (2.1)$$

$$e(n) = d(n) - y(n), \quad (2.2)$$

donde los filtros adaptativos pueden aplicarse de manera satisfactoria a ambientes desconocidos ya que gracias a su habilidad de rastrear las variaciones estadísticas en el tiempo y ofrecer tratamiento especial a las zonas que lo requieran, hace de ellos dispositivos eficientes para el tratamiento y control de señales aleatorias en tiempo real, en la figura 2.1 se ha mostrado la aplicación más simple que puede tener un filtro adaptativo, pero existen otro tipo de aplicaciones que se relacionan con la mencionada debido a que operan con un vector de entrada y una respuesta de salida que son usadas para ajustar el peso o valor de los coeficientes presentes en el filtro, la diferencia entre estas aplicaciones radica en la manera en que es extraída la información de la respuesta deseada, por ello se muestran a continuación cuatro clases básicas de aplicaciones de filtros adaptativos y para ello se toman las siguientes notaciones:

$x$  = Entrada aplicada al algoritmo adaptativo<sup>13</sup>.

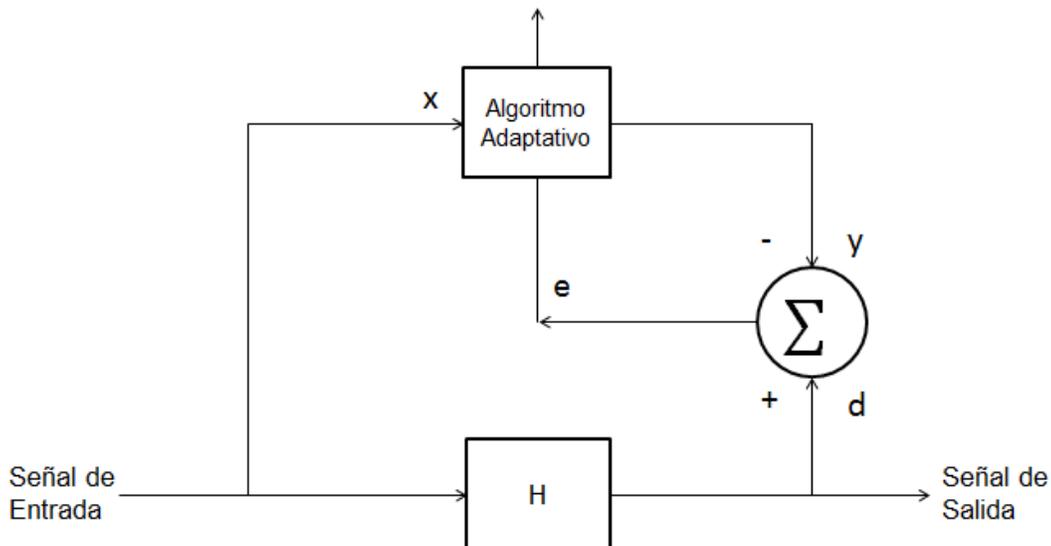
$y$  = Salida del algoritmo adaptativo.

$d$  = Señal deseada o señal de referencia.

$e = d - y =$  Señal de error.

### 2.2.1. IDENTIFICACIÓN [12] [13]

La identificación de un sistema desconocido consiste en especificar el modelo desconocido en términos de los datos disponibles experimentalmente, es decir, señales de respuestas deseadas de entrada-salida y una señal de error que ayuda a mejorar en cada ciclo los parámetros desconocidos del sistema, con esto la identificación adaptativa pretende llevar a cabo un procedimiento particular que utiliza las mediciones recibidas para aprender más sobre el modelo y con ello actualizar e incorporar la información que se recibe en el sistema que se está tratando, en la figura 2.2 se muestra el diagrama de bloques de la aplicación mencionada.



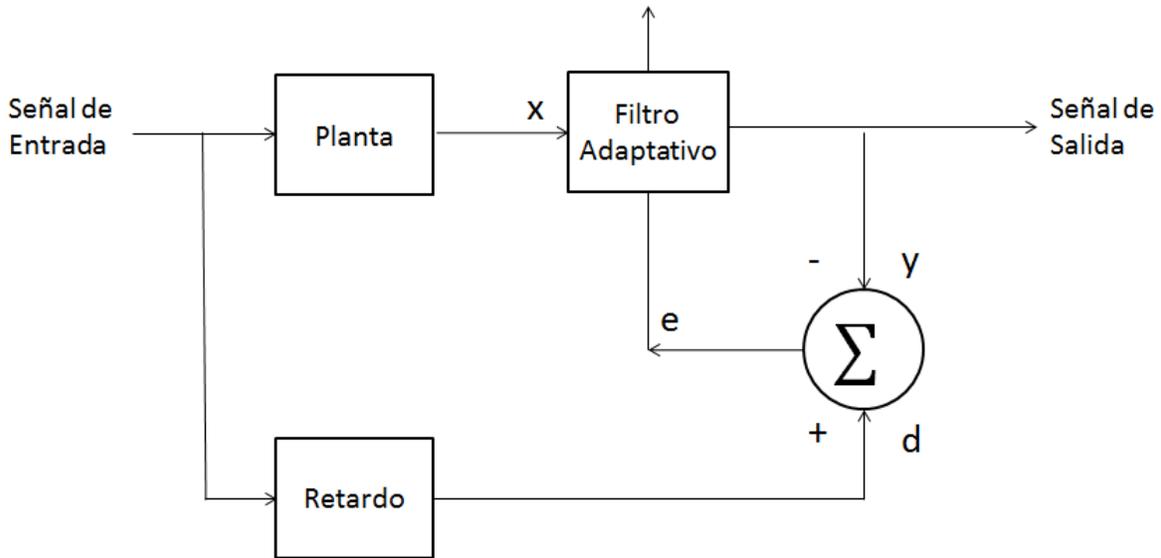
**Figura 2. 2.** Diagrama de bloques del sistema de identificación.

---

<sup>13</sup> Otros autores utilizan U en vez de X, en este caso se usará la letra x para denotar la señal de entrada.

### 2.2.2. MODELADO INVERSO [12]

La función del filtro adaptativo en esta aplicación es proporcionar un modelo inverso que represente el que mejor se adapte a una planta<sup>14</sup> desconocida, en la figura 2.3 se muestra la configuración que tiene esta aplicación.



**Figura 2. 3.** Diagrama de bloques del sistema de modelado inverso.

### 2.2.3. PREDICCIÓN [12]

La predicción lineal adaptativa es otra aplicación de los filtros adaptativos, la función del filtro en esta aplicación es proporcionar la mejor predicción hasta donde sea posible del valor presente de la señal aleatoria, en la figura 2.4 se muestra la configuración que tiene esta aplicación.

---

<sup>14</sup> Planta, se le llama así al sistema que va a ser controlado, éste puede ser ruidoso que está sujeto a perturbaciones y además de un tipo desconocido.

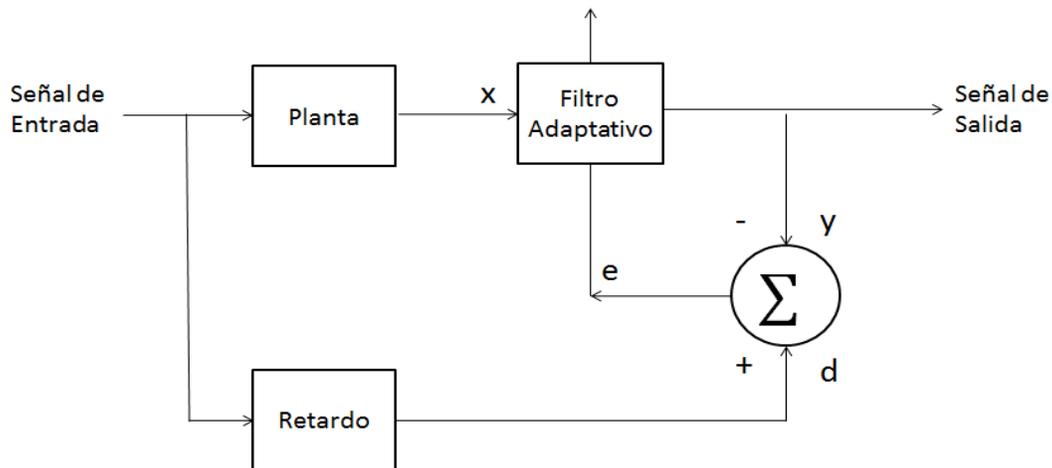


Figura 2. 4. Diagrama de bloques del sistema de predicción.

#### 2.2.4. CANCELACIÓN DE INTERFERENCIAS [14] [15]

En este último tipo de aplicación, el filtro adaptativo es utilizado para cancelar una interferencia desconocida que esta sumada a la señal con la información de interés, el principio usado en la cancelación de interferencias adaptativo se muestra en la figura 2.5.

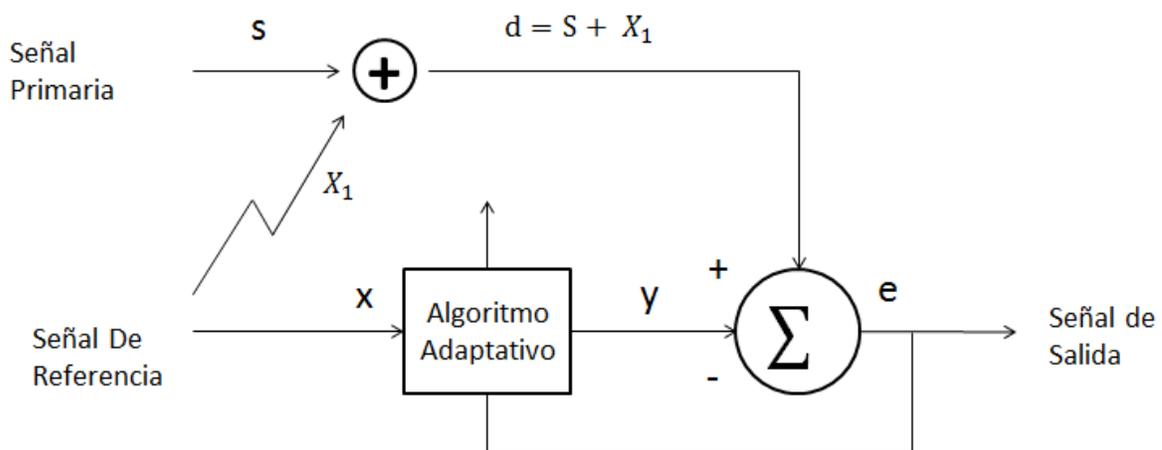


Figura 2. 5. Diagrama de bloques del sistema de cancelación de interferencias.

En todas las aplicaciones de este tipo existen dos señales, una señal primaria que contiene información útil afectada con interferencia correlacionada con la señal de referencia, el objetivo de esta aplicación es recuperar la señal primaria reduciendo al mínimo la interferencia, para ello se utilizan dos sensores independientes, el

primer sensor se usa para obtener la señal primaria  $d[n]$  cuya expresión está dada por la ecuación 2.3.

$$d[n] = s[n] + x_1[n], \quad (2.3)$$

donde  $s[n]$  es la señal generada por la fuente que se transmite a través del canal entre la fuente y el sensor primario, además de la señal generada por la fuente, el sensor primario también capta otra señal de interferencia  $x_1[n]$  que se suma a la señal original; el segundo sensor referido como señal de referencia recibe una señal de interferencia  $x[n]$  que se correlaciona con la señal  $x_1[n]$  pero no correlacionada con la señal  $s[n]$ . La salida del filtro  $y[n]$  es producida por el filtraje que realiza el algoritmo adaptativo a la interferencia de la señal de referencia  $x[n]$ , con esto los coeficientes del filtro se ajustan de manera tal, que sea lo más aproximado a la interferencia del sensor primario y como resultado aparece la señal de error especificada en la ecuación 2.4.

$$e[n] = d[n] - y[n], \quad (2.4)$$

que ayuda a llevar un control óptimo sobre los coeficientes del filtro minimizando cada vez más la señal de error, como resultado de este procedimiento se tiene la señal original con menor interferencia. Si se quiere cancelar múltiples interferencias no deseadas en señales contaminadas, simplemente se añade una rama de referencia extra por cada señal de interferencia, se sigue el mismo procedimiento y al final se suman todas las interferencias para restarla posteriormente de la señal dañada para tener una señal con mayor similitud a la original.

Para el diseño de un filtro adaptativo se utiliza un proceso de dos fases que son estimación y actualización, en el cual el algoritmo estima los parámetros estadísticos de la señal y luego actualiza los resultados obtenidos en la fórmula no recursiva para calcular el error respectivamente y con esto, empezar el proceso nuevamente; el algoritmo que se utilice debe converger a una solución donde la señal de error sea pequeña y estable, además debe ofrecer la capacidad de rastreo para actualizar los parámetros de un filtro adaptativo en cada iteración, así que, los parámetros se vuelven datos dependientes de la entrada y del error, con lo que se muestra que un filtro adaptativo es un dispositivo no lineal, ya que no obedece el principio de superposición, a pesar de esta propiedad, estos dispositivos son clasificados como lineales o no lineales [16], para decir que un filtro adaptativo es lineal, la estimación de una cantidad de interés calculada adaptativamente se hace mediante combinaciones lineales del conjunto de muestras disponibles en la entrada del mismo.

Para saber que algoritmo debe utilizarse se deben tener en cuenta los siguientes factores:

- **Velocidad de convergencia**, Se define como el número de iteraciones necesarias para alcanzar una solución óptima donde la señal de error sea la más pequeña posible.
- **Pérdida de información**, Se define como la similitud que se tiene entre la señal de salida del filtro y la señal original, cuando la similitud entre ellas es mayor, hay menor pérdida de información, por el contrario, cuando la similitud es menor hay mayor pérdida de información. Para hacer una estimación numérica de este factor se usa el Error Cuadrático Medio (MSE, *Mean Squared Error*), que compara las señales elemento a elemento, donde la diferencia de estos valores es la información perdida entre ambas, se describe mediante la ecuación 2.5.

$$MSE = \frac{1}{n} \sum_{i=1}^n (S_{filtrada} - S_{original})^2 \quad (2.5)$$

Donde,  $n$  es el número de muestras,  $S_{filtrada}$  es la señal filtrada y  $S_{original}$  la señal original.

### 2.3. ALGORITMO LMS [14] [17]

El algoritmo LMS es usado en filtros adaptativos para encontrar los coeficientes del filtro que permiten hallar el valor esperado mínimo del cuadrado de la señal de error, definida como la diferencia entre la señal deseada y la señal producida a la salida del filtro, hace parte de los algoritmos de gradiente estocástico, es decir, el filtro se adapta con base al error únicamente en el instante actual, su importancia radica en que es un algoritmo muy simple y no requiere medidas de las funciones de correlación, como tampoco inversión de la matriz de correlación; éste algoritmo responde a la ecuación 2.6.

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e[n] \mathbf{X}_n, \quad (2.6)$$

donde

$\mathbf{w}(n)$  → Vector de coeficientes del filtro.

$\mu$  → Factor de convergencia.

$e[n]$  → Señal de error.

$X_n \rightarrow$  Señal de entrada.

Cabe destacar que en éste algoritmo se observan dos procesos básicos:

- **Un proceso de filtrado**, éste primero incluye:
  - (I) Calcular la salida del filtro transversal producida por la entrada.
  - (II) Generar una estimación de error comparando la salida con una respuesta deseada.
- **Un proceso adaptativo**, por medio del cual se ajustan automáticamente los coeficientes del filtro de acuerdo a la estimación del error.

Cuando se unen los dos procesos que se han mencionado, se construye un lazo de realimentación que trabaja alrededor del algoritmo LMS.

Se tiene primero, un filtro transversal alrededor del cual se construye el algoritmo LMS, el cual desempeña el trabajo de filtrado, además, se dispone de un mecanismo para efectuar el proceso de control adaptativo sobre los valores de los coeficientes del filtro transversal.

En la figura 2.6 se muestra la estructura del filtro transversal.

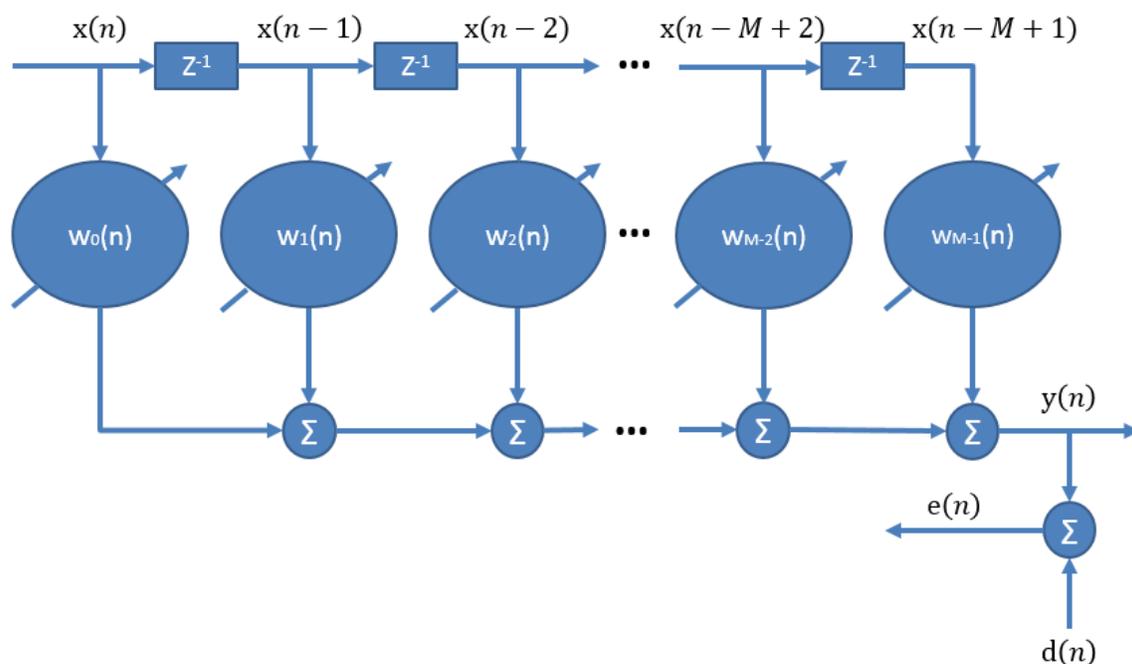


Figura 2. 6. Diagrama del filtro transversal.

Las muestras de la entrada  $x(n), x(n - 1), \dots, x(n - M + 1)$  forman los elementos del vector de entradas, en forma similar los coeficientes  $w_0(n), w_1(n), \dots, w_{M-1}(n)$  forman el vector de coeficientes  $\mathbf{w}(n)$ ,

La figura 2.7 detalla el mecanismo de control de los coeficientes, se tiene un valor escalar del producto del error de estimación  $e(n)$  y las entradas  $x(n - k)$ , para  $k = 0, 1, 2, \dots, M - 2, M - 1$ . Con este resultado se define la corrección  $\delta w_k(n)$  aplicado al coeficiente  $w_k(n)$  en la iteración  $n + 1$ . El factor escalar que se usa en este cálculo se denota por  $\mu$ , y se denomina con frecuencia **factor de convergencia**.

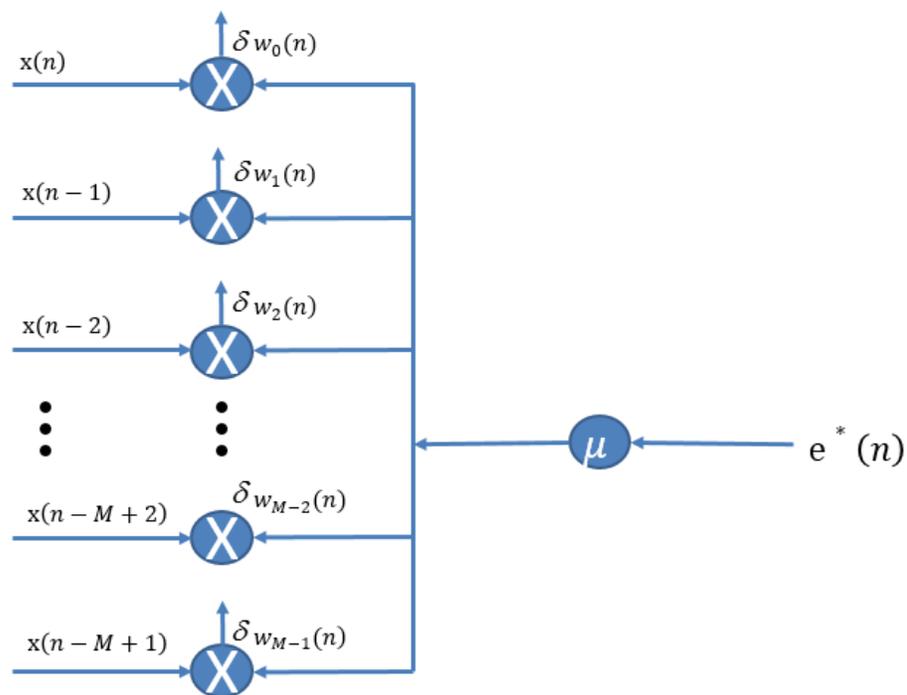


Figura 2. 7. Mecanismo de control de los coeficientes.

#### 2.4. ALGORITMO NLMS [11] [18]

En el algoritmo LMS, la actualización de los coeficientes depende del vector que se tenga en la entrada  $x(n)$ , cuando se tienen valores muy grandes de ruido se presenta un problema llamado amplificación del ruido de gradiente, para dar una solución a este problema se hace uso del algoritmo NLMS, donde la actualización de los coeficientes está normalizada con respecto a la norma cuadrada del vector de entrada, la ecuación 2.7 describe la relación de este algoritmo.

$$w(n+1) = w(n) + 2\beta e(n)X_n, \quad (2.7)$$

donde,

$w(n)$  → Vector de coeficientes del filtro.

$\beta$  → Estabilidad y velocidad de convergencia.

$e[n]$  → Señal de error.

$X_n$  → Señal de entrada.

El factor  $\beta$  de la ecuación del algoritmo NLMS está dado por la ecuación 2.8.

$$\beta = \frac{\alpha}{X^T(n)X(n)}. \quad (2.8)$$

En este caso el factor  $\alpha$  es una constante que puede ser elegido independientemente de las características de la señal de entrada.

## 2.5. ALGORITMO AR $\gamma$ [19] [20]

El algoritmo AR $\gamma$  es un algoritmo adaptativo cuya principal característica es la gran relación que hay entre la velocidad de convergencia y la variación del error de estimación. Las ecuaciones que describen el algoritmo AR $\gamma$  son las 2.9, 2.10, 2.11 y 2.11.

$$e[n] = x^T[n]w[n-1] - d[n] \quad (2.9)$$

$$g[n] = \frac{e[n] + \gamma x^T[n]q[n-1]}{1 + \alpha \gamma x^T[n]m_1 x[n]} \quad (2.10)$$

$$q[n] = \frac{\gamma}{\alpha + \gamma} q[n-1] - \alpha g[n] m_1 x[n] \quad (2.11)$$

$$w[n] = w[n-1] + \alpha q[n] \quad (2.12)$$

Donde  $d[n]$ , ecuación 2.13, corresponde a la señal deseada, obtenida de la siguiente forma, para un sistema de identificación:

$$d[n] = x^T[n]w_0[n] + r[n] \quad (2.13)$$

Donde  $x[n]$ : Vector de la señal de entrada.

$\mathbf{w}[n]$ : Vector de coeficientes del filtro adaptativo.

$d[n]$ : Escalar que corresponde a la señal deseada en el instante  $n$ .

$e[n]$ : Escalar que corresponde al error de medida en el instante  $n$ .

$g[n]$ : Escalar auxiliar en el instante  $n$ .

$\mathbf{q}[n]$ : Vector auxiliar.

$\mathbf{w}_0[n]$ : Vector de coeficientes del sistema a identificar.

$\alpha, \gamma, m_1$ : Parámetros de ajuste fijo.

# CAPÍTULO 3

## MODELADO, SIMULACIÓN E IMPLEMENTACIÓN

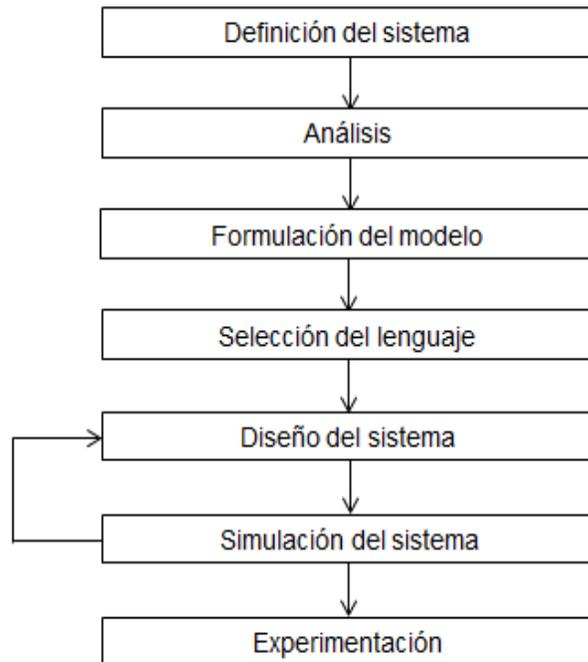
### 3.1. INTRODUCCIÓN

En este capítulo se muestra la metodología empleada en este trabajo de grado. Con el fin de cumplir con los objetivos propuestos se usa una adaptación del modelo descrito por Astaiza. E [21]; se diseñan filtros adaptativos con tres algoritmos diferentes (LMS, NLMS, AR $\gamma$ ), los cuales se comparan mediante la **velocidad de convergencia** mencionada en la sección 2.2 del capítulo anterior y además a esto se agregara la **pérdida de información** que sufre la imagen al pasar por los filtros con sus respectivos algoritmos, medida en términos del error cuadrático medio.

Primero se define un sistema con los requisitos que necesita, luego se hace un análisis de los requisitos mencionados para buscar un modelo que se adapte a las exigencias expuestas, posteriormente se selecciona el lenguaje adecuado para codificar el modelo y con esto realizar su validación y experimentación.

### 3.2. METODOLOGÍA DE SIMULACIÓN [21]

La metodología usada en este trabajo de grado es la siguiente:



**Figura 3. 1.** Adaptación de la metodología usada en “simulación de sistemas de telecomunicaciones”.

### 3.2.1. DEFINICIÓN DEL SISTEMA:

Uno de los principales problemas que afecta a la imagen en el momento de procesarla es el ruido que se puede presentar de diversas maneras, tales como: ruido impulsivo, ruido sal-pimienta, ruido blanco, ruido granular, etc. Existen filtros que ayudan a reducir el efecto que el ruido produce en la calidad de la imagen, pero el limitante que tienen este tipo de filtros es que utiliza una función lineal que trata de igual manera a todos los píxeles de la imagen, por lo cual se opta por utilizar un filtro adaptativo, que se puede configurar por medio de un algoritmo, gracias al cual el filtro ofrece un tratamiento especial a cada píxel de la imagen en su procesamiento, dadas estas exigencias este trabajo en su término debe cumplir con:

- Identificación y análisis de las características del filtro adaptativo que será implementado en el procesamiento digital de una imagen.
- Definición de una configuración para la implementación del filtro adaptativo que se va a utilizar para la minimización de ruido en una imagen.
- implementación del filtro adaptativo y evaluación de su desempeño mediante simulación.

### 3.2.2. ANÁLISIS DEL SISTEMA:

En el sistema con el que se trabajará se tienen variables que se pueden controlar para alterar el funcionamiento del filtro adaptativo dependiendo del algoritmo que se esté usando, las variables similares en los filtros con diferentes algoritmos son la señal de entrada que consta de una imagen en arreglo vectorial sumado a ella ruido que es de tipo gaussiano, la señal de referencia que es una fuente de ruido gaussiano que tiene correlación con el ruido sumado a la señal de entrada, la señal de error que se usa para actualizar el peso de los coeficientes, el número de coeficientes que junto con el factor de convergencia pueden hacer que el filtro converja con mayor o menor velocidad para que la salida del filtro sea lo más fiel posible a la imagen que se usó en la entrada y las variables diferentes en cada algoritmo son el factor de convergencia  $\mu$  en el algoritmo LMS, la velocidad de convergencia  $\beta$  en el algoritmo NLMS y los parámetros de ajuste fijo  $\alpha$ ,  $\gamma$  y  $m_1$  en el algoritmo ARy.

### 3.2.3. FORMULACIÓN DEL MODELO:

Para cumplir con las exigencias que se plantearon se hace necesario utilizar la aplicación del cancelador de interferencias (sección 2.2.4) de un filtro adaptativo con los algoritmos LMS, NLMS y ARy, la cual tiene una señal de referencia que sirve como modelo para adaptar los coeficientes del filtro y con ello procesar la imagen para disminuir el ruido presente en ellas.

### 3.2.4. SELECCIÓN DEL LENGUAJE:

Para llevar a cabo el desarrollo de los objetivos propuestos se hace necesario conseguir un lenguaje de programación capaz de manipular matrices, simular filtros, implementar algoritmos, procesar imágenes digitalmente; dadas estas exigencias el lenguaje de programación más apropiado para cumplir con los objetivos es el lenguaje MATLAB, debido a que se cuenta con licencia para usarlo.

### 3.2.5. DISEÑO DEL SISTEMA:

El sistema que se utiliza en las simulaciones es el mostrado en la figura 3.2, se muestra un esquema general y luego se explica con más detalle que significa cada bloque.

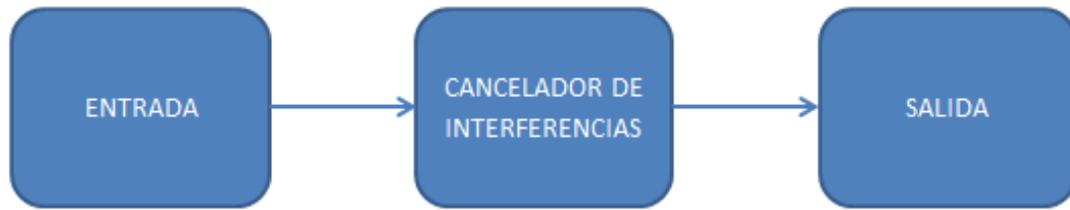


Figura 3. 2. Sistema simulado.

Primero se tiene una imagen que está representada por una matriz de valores numéricos, antes de pasarla de un arreglo matricial a un arreglo vectorial para que pase por el filtro ya sea como fila o columna, se agrega ruido y se envía en una señal denominada  $d[n]$ , la cual se suma con otra señal proveniente del filtro adaptativo para restarle este valor, esta señal se denomina  $y[n]$  y con esto los coeficientes del filtro se van adaptando con la ayuda de la señal de error que resulta de la operación entre las señales  $y[n]$  y  $d[n]$  con el fin de filtrar los valores presentes en la imagen que no se desean, cuando se quita el mayor número de irregularidades en la imagen se procede a recuperar la imagen y se hace el proceso inverso, es decir, se toma la imagen con arreglo vectorial y se vuelve a su estructura original como matriz la cual se muestra en la señal de salida.

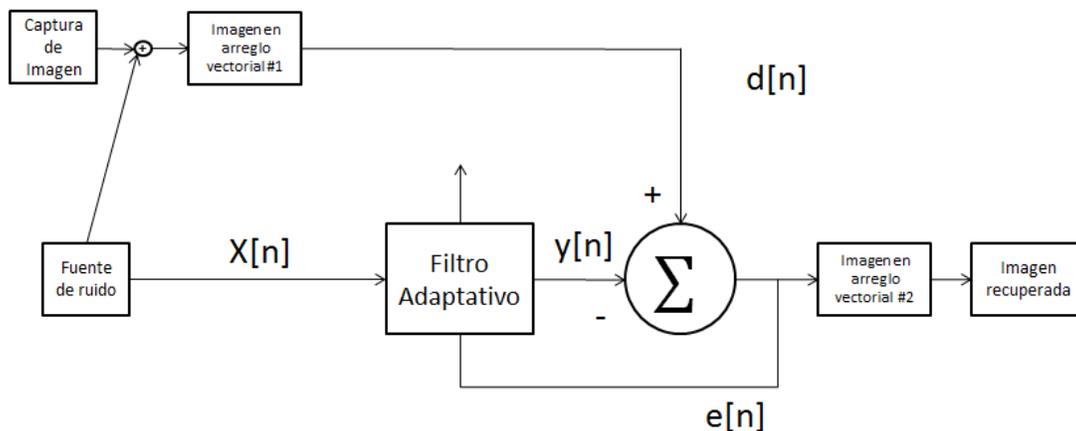
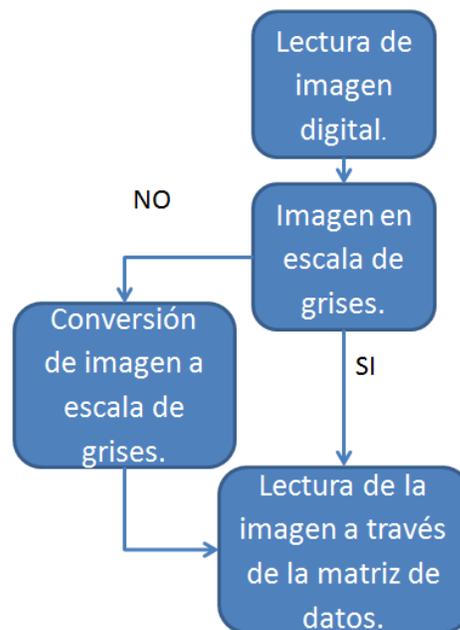


Figura 3. 3. Cancelador de interferencias.

### 3.2.5.1. Captura de imagen

La imagen que se tomó en este caso está compuesta por una matriz de tamaño  $512 \times 512$ , la categoría con la que se trabaja es la de mapa de bits y de tipo escala de grises y formato Tagged Image File Format (.TIFF), el proceso que usa

MATLAB para leer la imagen a partir de una matriz de datos se muestra en la figura 3.4:



**Figura 3. 4.** Proceso para lectura de una imagen en escala de grises.

- **Lectura de imagen digital:** Para reconocer un archivo de imagen en MATLAB se utiliza la función “imread” que lee una imagen de un archivo gráfico, para esto es necesario tener instalado el toolbox de procesamiento de imágenes de MATLAB que permite trabajar con gráficos, la sintaxis utilizada para usar la función imread es:

$A = \text{imread}(\text{'nombre del archivo.fmt'})$ , donde

**A:** Matriz de datos que se formará a partir de la imagen.

**fmt:** Representa el formato de la imagen que se usará, algunos de los formatos soportados por esta función son: .TIFF, .GIF, .JGP, .JPEG, etc.

- **Imagen a escala de grises:** Como se mencionó en la sección 1.2.2, una imagen a escala de grises es un tipo de imagen compuesto por un solo plano y los valores que pueden tomar van de 0 – 255, estos valores representan los colores negro y blanco respectivamente.
- **Conversión a escala de grises:** Si la imagen con la que se está trabajando es de color y compuesta por los tres planos RGB, es posible utilizar un comando que permite convertir la imagen de color a escala de grises, el comando es “rgb2gray” y la sintaxis es la siguiente:

$I = \text{rgb2gray}(\text{'nombre del archivo.fmt'})$

- **Lectura de la imagen a través de la matriz de datos:** Teniendo lista la imagen en escala de grises, por último se usa la función “imshow” para mostrar la imagen resultante de una matriz de datos y su sintaxis es:

$X = \text{imshow}(I)$ , donde  $I$  es la imagen convertida a escala de grises.

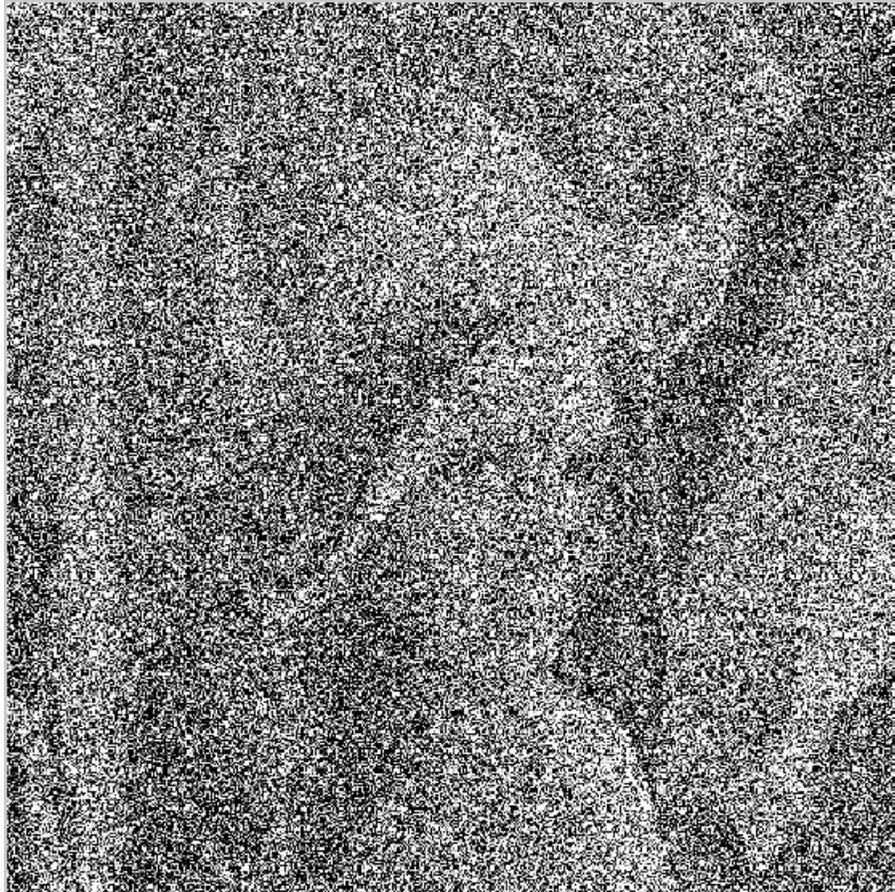


**Figura 3. 5.** Imagen .TIFF a escala de grises, 512 x 512.

#### 3.2.5.2. Fuente de ruido

La señal de referencia que se utiliza es una fuente de ruido correlacionada con el ruido sumado a la señal de entrada, en este caso es una imagen con los parámetros que se mencionaron en el literal anterior, este ruido afecta a los píxeles presentes en la imagen, los ruidos que afectan a la imagen son variados, tales como: ruido de luminancia, ruido de cromancia, de disparo, de cuantificación,

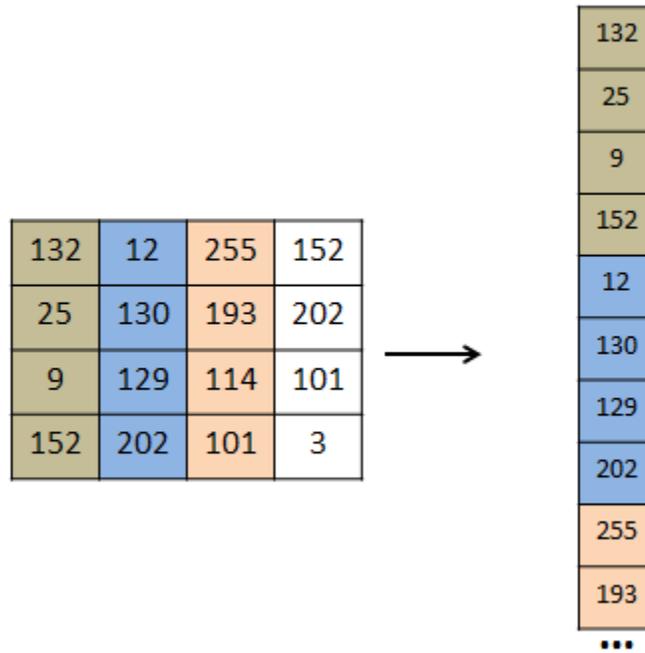
entre otros, pero el escogido para realizar el desarrollo de este trabajo es el ruido gaussiano con  $N(0, 1)$  que afecta a todos los pixeles presentes en la imagen.



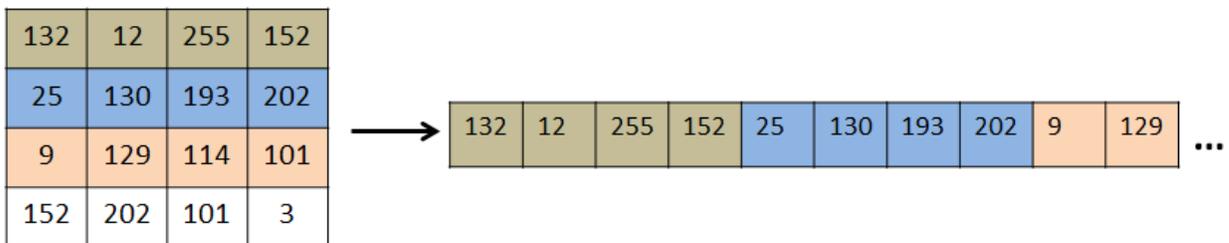
**Figura 3. 6.** Imagen afectada por el ruido gaussiano.

### 3.2.5.3. Imagen en arreglo vectorial #1

Con lo que se explicó en los literales 3.2.5.1. y 3.2.5.2. se obtiene la señal primaria que es la suma entre la imagen y la fuente de ruido, luego de esta operación se procede a pasar la imagen de un arreglo matricial a un arreglo vectorial, dado que el filtro adaptativo lee señales en una sola dimensión, en las figuras 3.7, 3.8 y 3.9 se muestran las formas que se ajustan a los arreglos vectoriales.



**Figura 3. 7.** Arreglo matricial leído como columna.



**Figura 3. 8.** Arreglo matricial leído como fila.

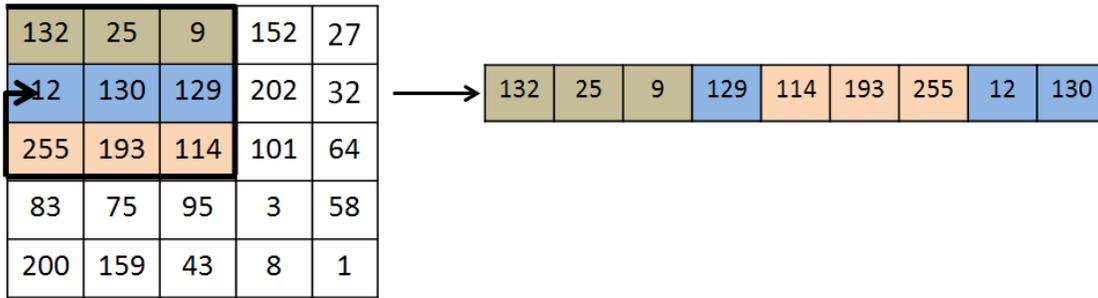


Figura 3. 9. Arreglo matricial 3x3.

El arreglo matricial  $3 \times 3$ , consiste en tener un pixel central y recorrer su vecindad, teniendo al pixel central como el ultimo en ser leído por el filtro; dado que el filtro solo puede procesar señales en una sola dimensión, se decide convertir el arreglo matricial  $3 \times 3$  en un vector fila de tamaño  $1 \times 9$  para cada lectura, cuando se realiza la lectura en los bordes de la matriz el arreglo  $3 \times 3$  se completa con ceros (0); este proceso se repetira hasta que todos los pixeles de la imagen hayan sido procesados.

#### 3.2.5.4. Filtro adaptativo

Cuando se tiene lista la señal primaria (Imagen en arreglo vectorial + ruido gaussiano) y la señal de referencia, se procede a pasar la señal primaria por el filtro adaptativo que puede operar bajo condiciones cambiantes y reajustarse continuamente para minimizar la señal de error con el fin de filtrar el ruido que la imagen posee en su estructura, esto se logra comparándola con la señal de referencia ya que el ruido que entra en la señal de referencia esta correlacionado con el ruido que se suma a la señal primaria y así ir reduciendo los valores no deseados en la imagen, para realizar este proceso el filtro adaptativo utiliza un algoritmo para ajustar los coeficientes con los que opera, los algoritmos que se usaron en este trabajo son **LMS**, **NLMS** y **AR $\gamma$** , los parámetros  $\mu$ ,  $\beta$ ,  $\alpha$ ,  $\gamma$ ,  $m_1$  se variaron experimentalmente hasta encontrar valores con los cuales el filtro presenta un buen funcionamiento; las ecuaciones que actualizan los coeficientes son las 3.1, 3.2, 3.3.

$$\text{LMS} \rightarrow w(n+1) = w(n) + \mu e[n]X_n \quad (3.1)$$

$$\text{NLMS} \rightarrow w(n+1) = w(n) + 2\beta e(n)X_n \quad (3.2)$$

$$\text{AR}\gamma \rightarrow w(n) = w(n-1) + \alpha q(n) \quad (3.3)$$

### 3.2.5.5. Imagen en arreglo vectorial #2

Luego de que el filtro adaptativo haya procesado la imagen y reducido los valores indeseados en ella, en la salida del filtro al igual que en su entrada la señal es un arreglo vectorial que regresará nuevamente a ser un arreglo matricial, es decir, se hace el proceso inverso que a la entrada del filtro.

### 3.2.5.6. Imagen recuperada

Por último se tiene la imagen transformada nuevamente a un arreglo matricial y restaurado, producto de todo el proceso que se viene mencionando y así tener la señal de salida con reducción de los valores que no se desean en ella, terminado todo este proceso se muestra la imagen resultante.

## 3.2.6. SIMULACIÓN DEL SISTEMA:

Para simular el sistema se siguieron los siguientes pasos:

- ✓ Selección del algoritmo.
- ✓ Configuración de los parametros del algoritmo.
- ✓ Se definen parámetros de ajuste del filtro.
- ✓ Ejecución de la simulación.
- ✓ Recolectar la información, dar un orden y analizar los resultados.

En la figura 3.9 se observa el diagrama de flujo que muestra la secuencia para simular el sistema mostrado en la figura 3.3, los resultados que se obtienen serán analizados en el siguiente capítulo.



Figura 3. 10. Diagrama de flujo del proceso de simulación.

### 3.2.7. EXPERIMENTACIÓN

En esta etapa se realiza el estudio del filtro adaptativo con los algoritmos seleccionados, con el fin de encontrar los parámetros óptimos que los tres algoritmos utilizados (LMS, NLMS, AR $\gamma$ ) necesitan para realizar el proceso de filtrado, junto con la configuración de cancelador de interferencias, para obtener el mejor resultado.



# CAPÍTULO 4

## EVALUACIÓN Y ANÁLISIS DE RESULTADOS

### 4.1. INTRODUCCIÓN

En este capítulo se hace la evaluación y respectivo análisis a los resultados que se obtuvieron en las simulaciones hechas con el filtro adaptativo configurado como cancelador de interferencias, las cuales se hicieron con la herramienta MATLAB R2012a y para su análisis se hace uso del error cuadrático medio que permite medir la cantidad de información que se pierde cuando una imagen con ruido es procesada por un filtro, cuanto más bajo sea el valor que se obtenga con el MSE mejor será la calidad que presenta, esto es, la imagen en la salida del filtro tendrá menos pérdida de información y mayor similitud con la imagen original, para medir la eficiencia del filtro adaptativo y los algoritmos se procede a analizar los siguientes escenarios en donde se varían los parámetros que contiene cada uno, los casos de estudio que se tomaron en cuenta para el desarrollo de este trabajo de grado se muestran a continuación:

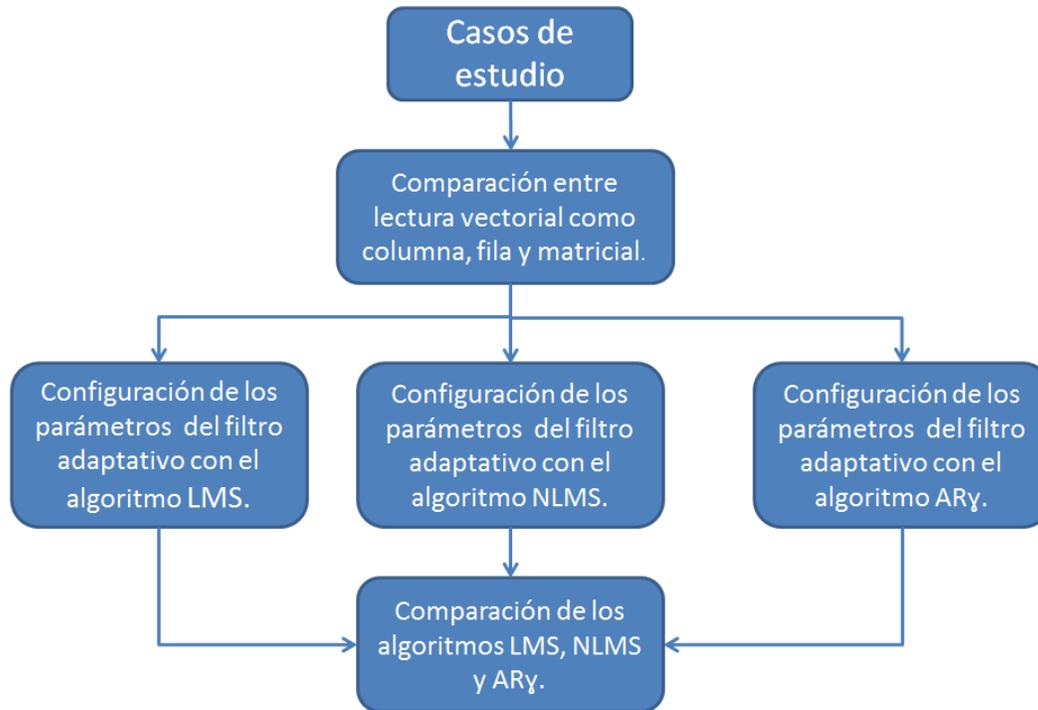
Caso de estudio 1: Comparación entre lectura vectorial como columna, fila y lectura vectorial como matriz.

Caso de estudio 2: Configuración de los parámetros óptimos del filtro adaptativo con el algoritmo LMS.

Caso de estudio 3: Configuración de los parámetros óptimos del filtro adaptativo con el algoritmo NLMS.

Caso de estudio 4: Configuración de los parámetros óptimos del filtro adaptativo con el algoritmo ARy.

Caso de estudio 5: Comparación de los algoritmos LMS, NLMS y ARy.



**Figura 4. 1.** Casos de estudio.

## 4.2. ANÁLISIS DE RESULTADOS

En esta sección se comparan los resultados obtenidos; en cada uno de los casos se ejecutaron 20 simulaciones y se tuvo en cuenta la media como dato de tabulación para las gráficas realizadas en MATLAB.

### 4.2.1. COMPARACIÓN ENTRE LECTURA VECTORIAL COMO COLUMNA, FILA Y LECTURA VECTORIAL COMO MATRIZ

Se realizaron pruebas con la forma de lectura de la imagen, primero se realizó el proceso de filtrado haciendo uso de los algoritmos ARy, LMS y NLMS con la imagen leída como una columna, como fila y en forma matricial, los resultados que arrojaron estas pruebas son expuestos en las siguientes figuras.

- En la figura 4.2 se muestran el resultado del promedio de las simulaciones hechas con el filtro adaptativo y el algoritmo AR $\gamma$ , este experimento se realiza bajo las siguientes condiciones:

Número de iteraciones: Tamaño de la imagen.

Número de coeficientes: 2.

$$\alpha = 0.01$$

$$\gamma = 0.01$$

$$m_1 = 1$$



a)



b)



c)

**Figura 4. 2.** Error en la salida del filtro y el algoritmo AR $\gamma$  con la imagen leída a) como columna, b) como fila y c) como matriz 3x3.

- En la figura 4.3 se muestra el resultado del promedio de las simulaciones hechas con el filtro adaptativo y el algoritmo LMS, este experimento se realiza bajo las siguientes condiciones:

Número de iteraciones: Tamaño de la imagen.

Número de coeficientes: 2.

$$\mu = 0.00008$$



a)



b)



c)

**Figura 4. 3.** Error en la salida del filtro y el algoritmo LMS con la imagen leída a) como columna, b) como fila y c) como matriz 3x3.

- En la figura 4.4 se muestra el resultado del promedio de las simulaciones hechas con el filtro adaptativo y el algoritmo NLMS, este experimento se realiza bajo las siguientes condiciones:

Número de iteraciones: Tamaño de la imagen.

Número de coeficientes: 2.

$\beta = 0.0001$



a)



b)



c)

**Figura 4. 4.** Error en la salida del filtro y el algoritmo NLMS con la imagen leída a) como columna, b) como fila y c) como matriz 3x3.

La tabla 4.1 muestra los valores de MSE de las figuras 4.2, 4.3 y 4.4 de las imágenes recuperadas comparadas con la imagen original.

**Tabla 4. 1.** Valores de la pérdida de información en las imágenes filtradas con los diferentes métodos de lectura.

ALGORITMO	METODO DE LECTURA	MSE IMAGEN ORIGINAL	MSE IMAGEN TRANSPUESTA
<b>LMS</b>	Columna	7,162E-04	105,45E-04
	Fila	105,01E-04	6,74E-04
	Matricial	117E-04	117,57E-04
<b>NLMS</b>	Columna	1,2142E-04	101,12E-04
	Fila	100,72E-04	1,34E-04
	Matricial	964E-04	96,09E-04
<b>AR<sub>γ</sub></b>	Columna	1,6433E-04	101,25E-04
	Fila	100,86E-04	1,47E-04
	Matricial	49E-04	48,52E-04

La tabla 4.1, permite observar que el método por el cual la imagen seleccionada pierde menos información en su proceso de filtrado, es la recuperada mediante el método de lectura por columnas, es decir, columna por columna; ahora, al transponer la imagen, ésta pierde menos información en el proceso de filtrado usando el método de filas. Con esto se observa que el método de lectura depende de las características de la imagen a filtrar, porque la variación de la frecuencia en forma vertical es diferente con la horizontal.

En las siguientes secciones se busca obtener los valores óptimos para los parámetros de cada algoritmo, el criterio de selección que se usa para la escogencia es el de menor valor en el MSE, en este caso se mide después de que en la imagen se reducen al mínimo las interferencias presentes en su estructura.

#### 4.2.2. CONFIGURACIÓN DE LOS PARÁMETROS ÓPTIMOS DEL FILTRO ADAPTATIVO CON EL ALGORITMO LMS

Para encontrar los valores con los que el filtro funciona de manera óptima, se varía el parámetro del algoritmo LMS y se escoge el valor del parámetro con el cual se consigue el menor MSE, en la tabla 4.2 se muestran algunos de los valores que se obtuvieron por medio de simulación, este experimento se realizó bajo las siguientes condiciones:

Número de iteraciones: Tamaño de la imagen.

Número de coeficientes: 2.

**Tabla 4. 2.** Valores del parámetro  $\mu$  para el algoritmo LMS.

$\mu$	MSE
0,00001	0,36394757
0,00005	0,00872468
0,0001	0,00015157
<b>0,0002</b>	<b>3,2032E-05</b>
0,0003	3,8813E-05
0,0004	4,4123E-05
0,0005	5,24E-05
0,0006	5,658E-05
0,0007	7,3418E-05
0,0008	7,1408E-05
0,0009	7,94E-05
0,001	8,3994E-05
0,005	0,00035787
0,01	0,00070123
0,05	0,00345571
0,1	0,00695273

En la tabla 4.2 se observa que para un factor de convergencia  $\mu = 0,0002$  se obtiene el mínimo error cuadrático medio ( $MSE = 3,2032E - 05$ ), lo que significa que para este valor de  $\mu$  la imagen recuperada presenta menor pérdida de información con respecto a la imagen original.

#### 4.2.3. CONFIGURACIÓN DE LOS PARÁMETROS ÓPTIMOS DEL FILTRO ADAPTATIVO CON EL ALGORITMO NLMS

Para encontrar los valores con los que el filtro funciona de manera óptima, se varía el parámetro del algoritmo NLMS y se escoge el valor del parámetro con el cual se consigue el menor MSE, en la tabla 4.2 se muestran algunos de los valores que se obtuvieron por medio de simulación, este experimento se realizó bajo las siguientes condiciones:

Número de iteraciones: Tamaño de la imagen.

Número de coeficientes: 2.

**Tabla 4. 3.** Valores del parámetro  $\mu$  para el algoritmo LMS.

$\beta$	MSE
0,00001	0,13673307
0,00002	0,02054567
0,00003	0,00335002
0,00004	0,00067139
0,00005	0,00018297
0,00006	0,00010558
<b>0,00007</b>	<b>8,9188E-05</b>
0,00008	0,00010283
0,00009	0,00010944
0,0001	0,00011485
0,0002	0,0002173
0,0003	0,0003285
0,0004	0,00043804
0,0005	0,00057672
0,001	0,00108892
0,005	0,00540265
0,01	0,01077597
0,05	0,05374374
0,1	0,10874669

En la tabla 4.3 se observa que para un factor de convergencia  $\beta = 0,00007$  se obtiene el mínimo error cuadrático medio ( $MSE = 8,9188E - 05$ ), lo que significa que para este valor de  $\beta$  la imagen recuperada presenta menor pérdida de información con respecto a la imagen original.

#### 4.2.4. CONFIGURACIÓN DE LOS PARÁMETROS ÓPTIMOS DEL FILTRO ADAPTATIVO CON EL ALGORITMO AR $\gamma$

Para encontrar los valores con los que el filtro funciona de manera óptima, se realizan simulaciones con diferentes combinaciones con los valores mostrados en la tabla 4.3 entre los tres parámetros que tiene el algoritmo AR $\gamma$ , al parámetro que mayor número de muestras se le dió fue el parámetro

$m_1$ , debido a que el algoritmo presenta más sensibilidad cuando se hace la variación de este parámetro en el proceso de filtrado y por el contrario en el parámetro  $\alpha$  se toman saltos más grandes en los valores de simulación porque es menos sensible en el mismo proceso; en la tabla 4.4 se muestran los valores de los parámetros cuya configuración tiene el menor valor del MSE, en esta tabla se muestra un resumen de todos los valores arrojados por las simulaciones en los cuales el filtro adaptativo con el algoritmo AR $\gamma$  funciona de manera eficiente.

$\alpha$	$\gamma$	$m_1$	
0,01	0,001	0,001	0,13
0,05	0,002	0,002	0,14
0,1	0,003	0,003	0,15
0,5	0,004	0,004	0,16
1	0,005	0,005	0,17
1,5	0,006	0,006	0,18
3	0,007	0,007	0,19
5	0,008	0,008	0,2
10	0,009	0,009	0,21
	0,01	0,01	0,3
	0,02	0,011	0,4
	0,03	0,012	0,5
	0,04	0,013	0,6
	0,05	0,014	0,7
	0,06	0,015	0,8
	0,07	0,016	0,9
	0,08	0,017	1
	0,09	0,018	1,1
	0,1	0,019	1,2
	0,2	0,02	1,3
	0,3	0,03	1,4
	0,4	0,04	1,5
	0,5	0,05	1,6
	0,6	0,06	1,7
	0,7	0,07	1,8
	0,8	0,08	1,9
	0,9	0,09	2
	1	0,1	2,1
	2	0,11	3
	3	0,12	4

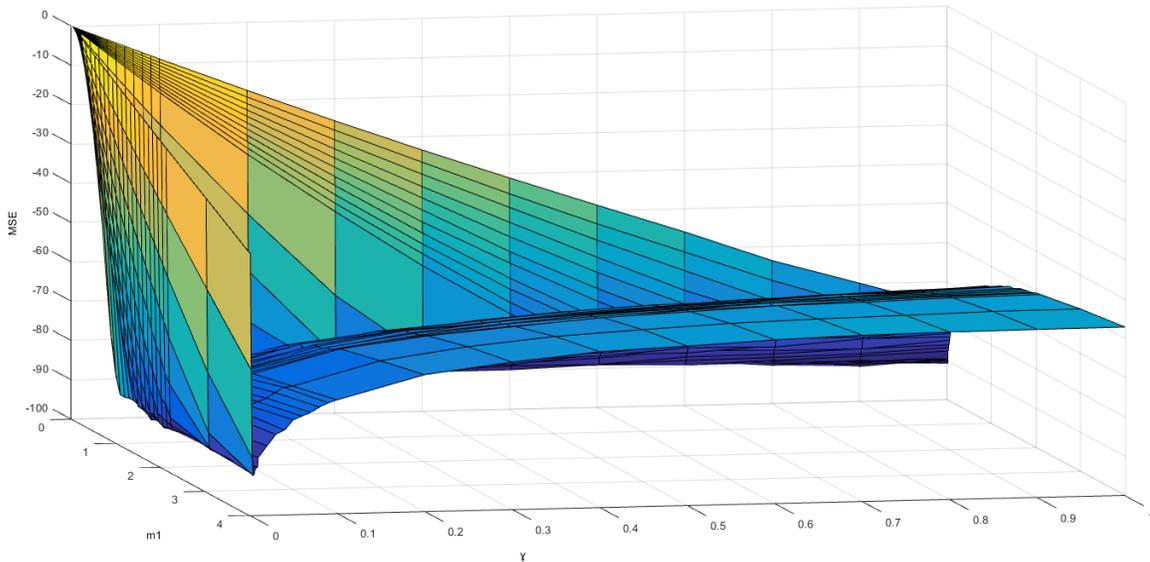
Tabla 4. 4. Valores con los que se trabajó en las simulaciones.

Las figuras 4.5 y 4.6, indican el comportamiento del MSE para el algoritmo AR $\gamma$  manteniendo el parámetro  $\alpha$  fijo con un valor de  $\alpha = 0,01$  y variando los parámetros  $\gamma$  y  $m_1$  visto desde dos maneras. Se puede observar que entre más oscura sea la tonalidad de azul, menor será el valor de MSE que arroja el algoritmo. Finalmente en la tabla 4.6 se describe la relación entre los parámetros  $\alpha\gamma m_1$ ; se tomaron cinco muestras por cada caso, en los cuales se mantenía fijo  $\alpha$  y se varían los parámetros  $\gamma$  y  $m_1$ .

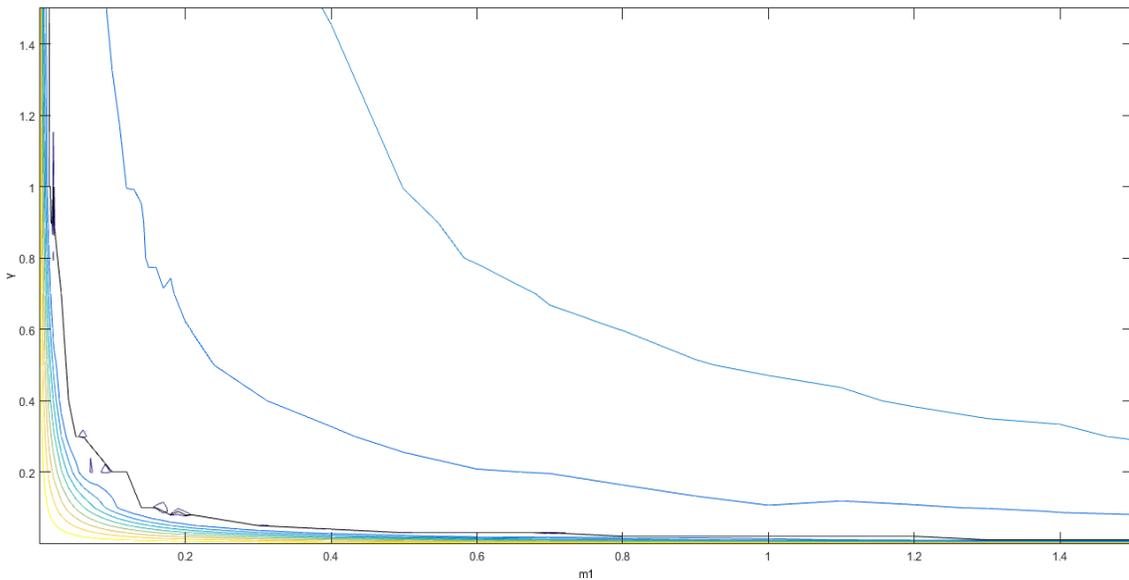
Este experimento se realiza bajo las siguientes condiciones:

Número de iteraciones: Tamaño de la imagen.

Número de coeficientes: 2.



**Figura 4. 5.** Grafica de error cuadrático en un tres dimensiones para diferentes valores de  $m_1$  y  $\gamma$  con  $\alpha = 0.01$ .



**Figura 4. 6.** Grafica de error cuadrático en dos dimensiones para diferentes valores de  $m_1$  y  $\gamma$  con  $\alpha = 0.01$ .

## Implementación de un filtro adaptativo para reducir el ruido en una imagen

$\alpha$	$\gamma$	$m_1$	$\alpha \gamma m_1$
0,01	1	0,015	0,00015
	0,4	0,04	0,00016
	0,1	0,16	0,00016
	0,02	0,9	0,00018
	0,01	1,8	0,00018
0,05	0,3	0,011	0,00016
	0,1	0,03	0,00015
	0,02	0,16	0,00016
	0,003	1,2	0,00018
	0,002	1,8	0,00018
0,1	0,5	0,004	0,00020
	0,2	0,008	0,00016
	0,04	0,04	0,00016
	0,09	0,019	0,00017
	0,005	0,4	0,00020
0,5	0,3	0,001	0,00015
	0,07	0,005	0,00017
	0,01	0,04	0,00020
	0,002	0,16	0,00016
	0,001	0,3	0,00015
1	0,1	0,002	0,00020
	0,04	0,004	0,00016
	0,009	0,017	0,00015
	0,003	0,06	0,00018
	0,001	0,16	0,00016
1,5	0,1	0,001	0,00015
	0,03	0,004	0,00018
	0,01	0,011	0,00016
	0,002	0,06	0,00018
	0,001	0,11	0,00016
3	0,06	0,001	0,00018
	0,03	0,002	0,00018
	0,01	0,005	0,00015
	0,002	0,03	0,00018
	0,001	0,06	0,00015
5	0,03	0,001	0,00015
	0,02	0,002	0,00020
	0,005	0,006	0,00015
	0,002	0,015	0,00015
	0,001	0,04	0,00020
10	0,02	0,001	0,00020
	0,004	0,004	0,00016
	0,003	0,006	0,00018
	0,002	0,009	0,00018
	0,001	0,017	0,00017
Media del producto de las combinaciones.			0,0001697

**Tabla 4. 5** Relación entre los parámetros  $\alpha$  y  $m_1$  del algoritmo AR $\gamma$ .

La tabla 4.6 proporciona los resultados para los cuales la imagen recuperada presenta menor pérdida de información usando el algoritmo AR $\gamma$ .

A	$\gamma$	$m_1$	MSE	$\alpha m_1$
0,01	0,9	0,018	2,85E-05	0,000162
0,05	0,06	0,05	2,88E-05	0,00015
<b>0,1</b>	<b>0,3</b>	<b>0,005</b>	<b>2,72E-05</b>	<b>0,00015</b>
0,5	0,04	0,009	2,99E-05	0,00018
1	0,02	0,01	2,79E-05	0,0002
1,5	0,001	0,11	2,87E-05	0,000165
3	0,009	0,006	2,94E-05	0,000162
5	0,008	0,004	2,99E-05	0,00016
10	0,002	0,009	2,98E-05	0,00018

Tabla 4. 6. Valores óptimos de los parámetros del algoritmo AR $\gamma$ .

La combinación que mejores resultados presentó, es la realizada con los valores de los parámetros de ajuste fijo con  $\alpha = 0,1$ ,  $\gamma = 0,3$  y  $m_1 = 0,005$  con un valor de  $MSE = 2,72 \times 10^{-5}$ .

Al hacer un análisis más profundo de las combinaciones presentadas en las tablas 4.5 y 4.6, se nota que para una configuración de cancelador de interferencias las muestras cumplen con una relación entre las tres que indica que el producto de los parámetros  $\alpha \times \gamma \times m_1 \approx 0,0001697$ , bajo las condiciones iniciales del experimento.

Las tablas 4.2, 4.3 y 4.6 muestran los resultados medidos en cuanto la pérdida de información por medio del MSE, la imagen que presenta menor pérdida de información es la recuperada con el algoritmo AR $\gamma$  con un valor de  $MSE = 2,72E-05$  y la imagen que presenta mayor pérdida de información es la recuperada con el algoritmo NLMS con un valor de  $MSE \approx 8,92E-05$ , si bien se observa que los algoritmos tienen resultados diferentes son diferencias muy pequeñas que rondan alrededor de potencias de  $10^{-5}$ , en cuanto a la velocidad de convergencia se realizará el análisis de los tres algoritmos en la siguiente sección.

#### 4.2.5. COMPARACIÓN DE LOS ALGORITMOS LMS, NLMS, AR $\gamma$

En las secciones anteriores se han encontrado los valores de los parámetros para los algoritmos utilizados, con los cuales, la imagen recuperada presenta el mínimo MSE, es decir, la menor pérdida de información. En esta sección, se compara la velocidad de convergencia, definiendo una condición de MSE mínimo, al cual los tres algoritmos pueden llegar, observando el comportamiento de los datos en la etapa de aprendizaje.

Las figuras 4.7 y 4.8, muestra el comportamiento de los resultados que se obtuvieron en la señal de error para cada algoritmo con su respectiva tendencia para un valor de  $MSE \approx 8,8 \times 10^{-5}$ .

Este experimento se realizó bajo las siguientes condiciones:

Número de iteraciones: 120000, se escoge este valor porque abarca el rango de iteraciones necesarias para que los algoritmos alcancen un MSE.

Número de coeficientes: 2.

$\mu = 0,001$ .

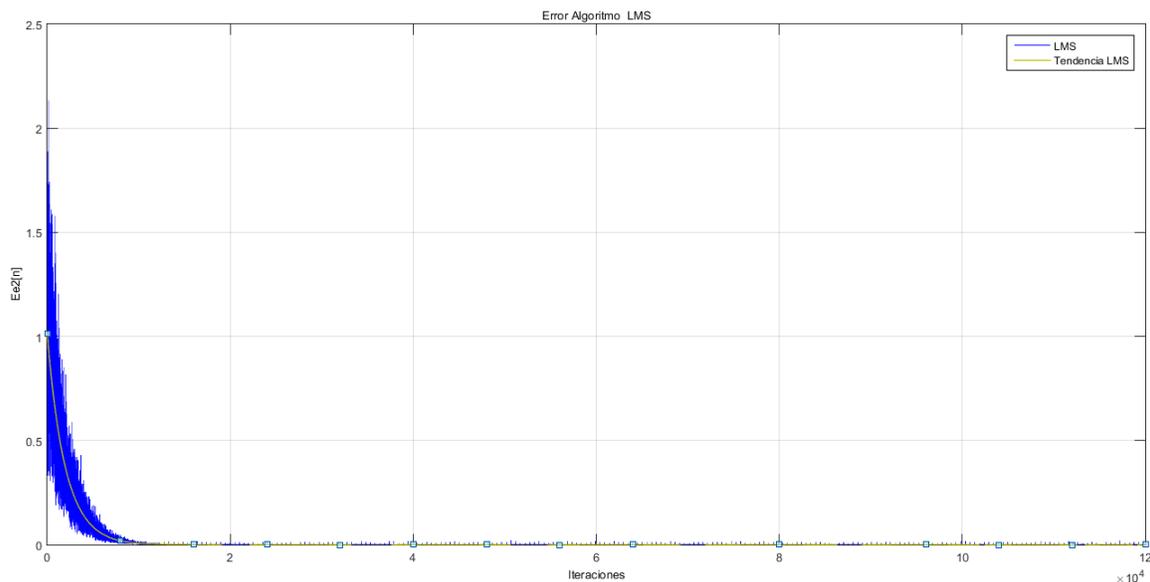
$\beta = 0,00007$ .

$\alpha = 0,1$ .

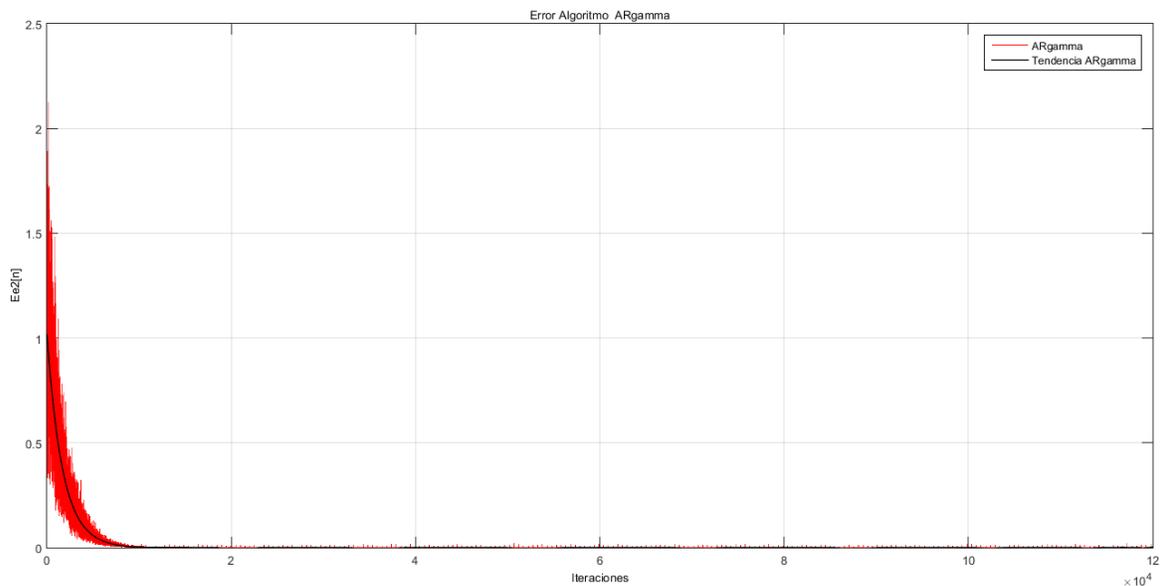
$\gamma = 0,9$ .

$m_1 = 0,013$ .

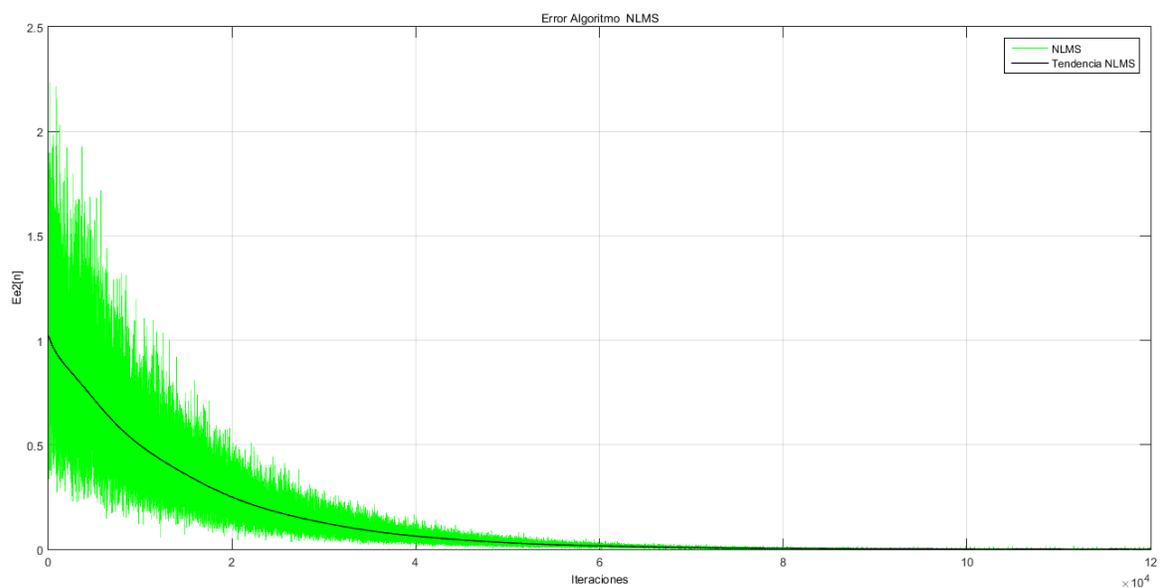
$MSE \approx 8,8 \times 10^{-5}$ .



a)



b)



c)

**Figura 4. 7.** Gráficas del error con la tendencia para a) algoritmo LMS, b) algoritmo  $AR\gamma$  y c) algoritmo NLMS.

La figura 4.8 muestra las tres líneas de tendencia superpuestas, se aprecia la velocidad de convergencia para los tres algoritmos utilizados.

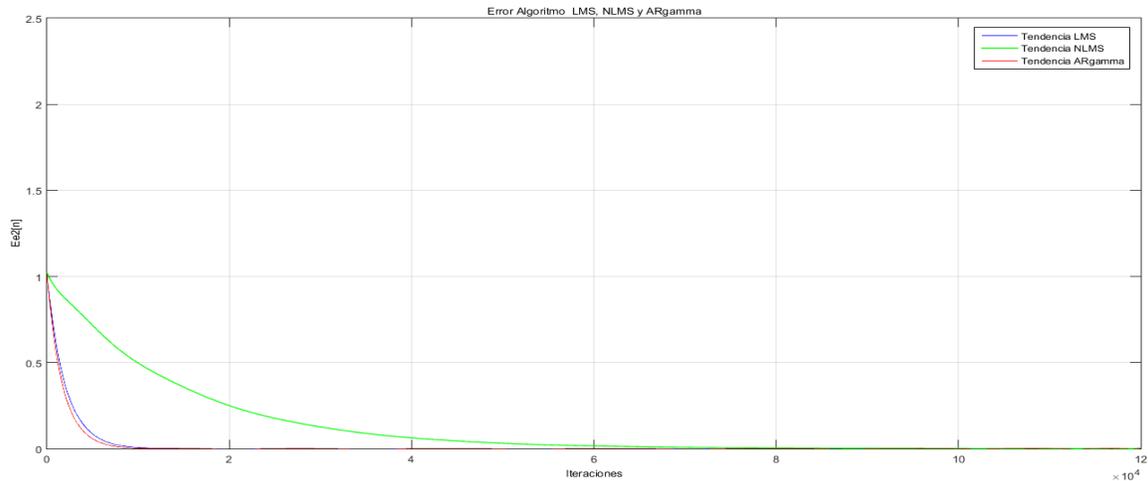


Figura 4. 8. Gráficas de tendencia para los tres algoritmos utilizados.

Los datos de la tabla 4.7, indican un valor de error cuadrático medido para un número determinado de iteraciones en los tres algoritmos utilizados durante su etapa de aprendizaje.

Número de iteraciones	Error Cuadrático LMS	Error Cuadrático NLMS	Error Cuadrático AR <sub>γ</sub>
500	0,780603565	0,977624936	0,749830422
1000	0,619523724	0,939399314	0,578901276
1500	0,488028679	0,903993605	0,441601294
2000	0,381489378	0,87099963	0,332422353
2500	0,295876812	0,840069648	0,246592567
3000	0,227698965	0,810909158	0,179997269
3500	0,173943005	0,783270375	0,129106663
4000	0,132022381	0,756946303	0,090909716
4500	0,099728512	0,731765382	0,062853881
5000	0,075186768	0,707586667	0,042790263
5500	0,056816444	0,684295491	0,028923861
6000	0,043294464	0,661799584	0,01976852
6500	0,033522548	0,640025614	0,014106281
7000	0,026597596	0,618916108	0,010950784
7500	0,02178505	0,598426735	0,009514445
8000	0,018495019	0,578523914	0,009179108
8500	0,016260943	0,559182719	0,009469904
9000	0,014720611	0,540385055	0,010032063
9500	0,013599331	0,522118092	0,010610434
10000	0,012695077	0,504372912	0,011031489

Tabla 4. 7. Error cuadrático en un número de iteraciones fijas.

En la figura 4.8, se nota que el error cuadrático para los algoritmos LMS y ARy decrece con mayor rapidez comparado con el algoritmo NLMS, para corroborar lo observado en la figura, se tomaron valores del error cuadrático en un número determinado de iteraciones, notando que para cada iteración, los valores del MSE medido en el algoritmo NLMS son mayores en comparación con los otros.

En la tabla 4.8, se indica el número las iteraciones que necesita cada algoritmo para que el error cuadrático alcance el valor de MSE definido en las condiciones iniciales de este experimento.

Error Cuadrático	Número de iteraciones LMS	Número de iteraciones NLMS	Número de iteraciones ARy
$8,8 \times 10^{-5}$	17106	114975	14732

**Tabla 4. 8.** Número de iteraciones para llegar a un error cuadrático.

El algoritmo que necesita un menor número de iteraciones para alcanzar el MSE objetivo de  $8,8 \times 10^{-5}$  es el ARy.

Al analizar la figura 4.8 y las tablas 4.7 y 4.8, se observa que el algoritmo ARy presenta la mayor velocidad de convergencia en comparación con los algoritmos LMS y NLMS, debido, a que éste llega primero a un valor de error cuadrático definido; además, al tomar muestras del error cuadrático en la etapa de aprendizaje de los tres algoritmos en un número determinado de iteraciones, es menor comparados con los otros dos.

La figura 4.9 muestra las imágenes recuperadas con el filtro adaptativo con los tres algoritmos diferentes, en la izquierda está el ruido gaussiano sumado a la estructura de las imágenes utilizadas para realizar el filtrado, las del centro muestran el resultado de las imágenes a la salida del filtro cuando está en la etapa de aprendizaje, es decir, cuando los coeficientes por medio de iteraciones se van adecuando a la señal de referencia; mientras esto ocurre el ruido todavía afecta a la señal de manera evidente como se aprecia en las imágenes del medio. Cuando se inicia esta etapa, los coeficientes se van adaptando por cada iteración, permitiendo que los efectos que el ruido produce disminuyan, y por último, las imágenes de la derecha, son las filtradas por segunda vez, es decir, se toman nuevamente las de la izquierda y se repite el proceso de filtrado para cada algoritmo, para llevar a cabo el segundo proceso de filtrado, los coeficientes deben terminar su etapa de aprendizaje para mostrar en la salida del filtro, la imagen con mayor similitud con respecto a la original (figura 3.5).

Este experimento se realizó bajo las siguientes condiciones:

Número de iteraciones: Tamaño de la imagen.

Número de coeficientes: 2.

$\mu = 0,0002.$   
 $\beta = 0,00007.$   
 $\alpha = 0,1.$   
 $\gamma = 0,3.$   
 $m_1 = 0,005.$



a)



b)



c)

**Figura 4. 9.** Imagen filtrada para un  $MSE = 8,8 \times 10^{-5}$  con a) algoritmo LMS, b) algoritmo NLMS con un y c) algoritmo AR $\gamma$ .

En las imágenes del centro de la figura 4.9, se observa que la etapa de aprendizaje no requiere que estas sean procesadas en su totalidad, por tal razón, se procesa una parte de estas para realizar la etapa de aprendizaje, al terminar esta etapa, se pasa la imagen nuevamente en su totalidad para cada algoritmo y observar su funcionamiento.

La figura 4.10, muestra la parte de la imagen que necesita cada algoritmo para llevar a cabo su etapa de aprendizaje.

Este experimento se realizó bajo las siguientes condiciones:

Número de iteraciones: Tamaño de la imagen.

Número de coeficientes: 2.

$$\mu = 0,0002.$$

$$\beta = 0,00007.$$

$$\alpha = 0,1.$$

$$\gamma = 0,3.$$

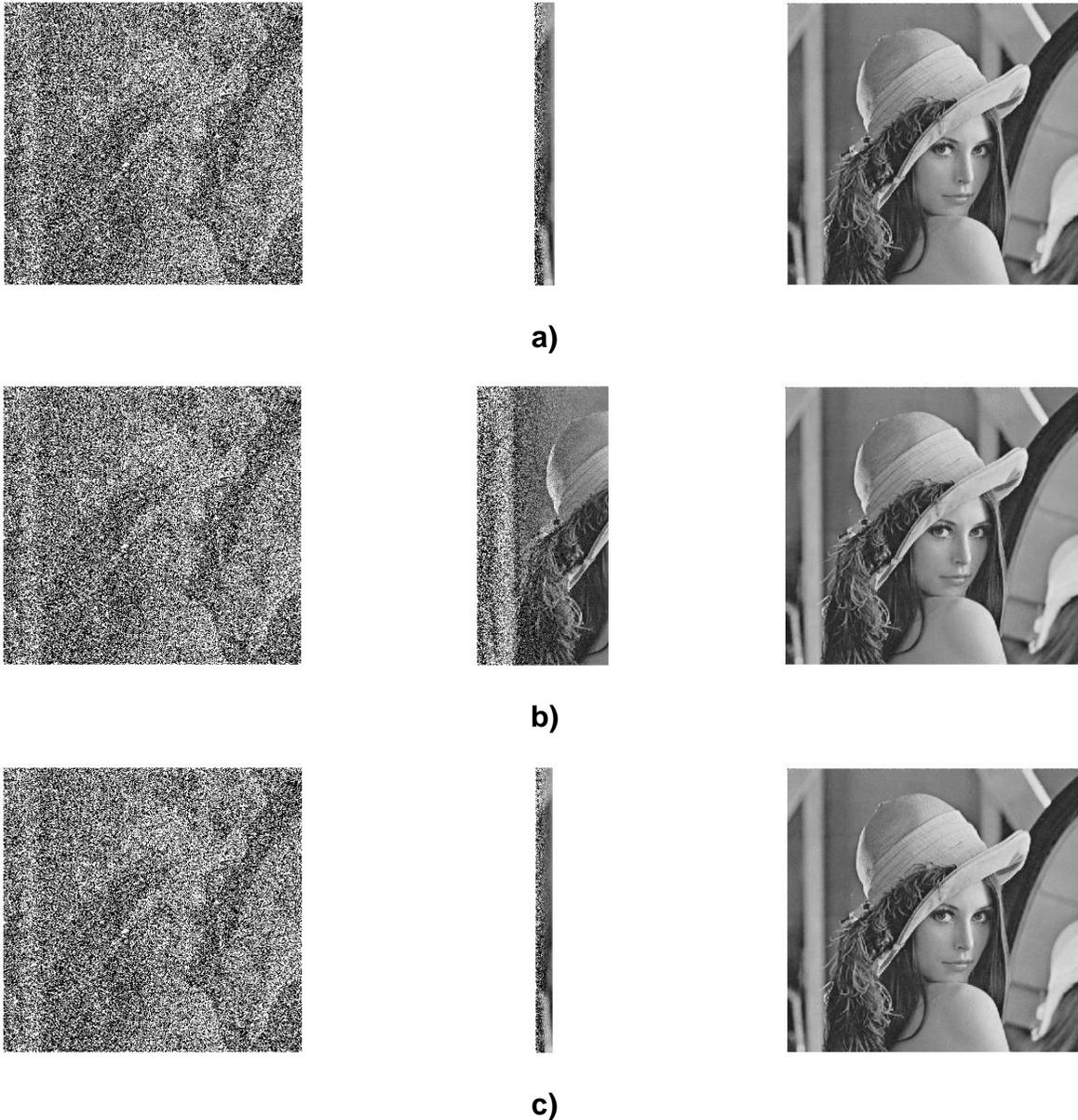
$$m_1 = 0,005.$$

Para conocer el número de columnas que requieren los algoritmos usados en la etapa de aprendizaje, se toma el número de iteraciones para alcanzar el MSE deseado (tabla 4.8) y se divide respecto al tamaño de cada columna (512).

$$\text{Algoritmo LMS: } \frac{17106}{512} \approx 34 \text{ columnas.}$$

$$\text{Algoritmo NLMS: } \frac{114975}{512} \approx 225 \text{ columnas.}$$

$$\text{Algoritmo AR}\gamma: \frac{17106}{512} \approx 29 \text{ columnas.}$$



**Figura 4. 10.** Imagen filtrada para un  $MSE = 8,8 \times 10^{-5}$  usando una parte de la imagen en la etapa de aprendizaje para los algoritmos **a)** LMS, **b)** NLMS y **c)** ARy.

Las imágenes de la derecha en la figura 4.10, presentan un filtraje de principio a fin, confirmando que no es necesario que en la etapa de aprendizaje se procese toda la imagen sino una parte de ella, mientras los coeficientes de los algoritmos usados terminan su etapa de aprendizaje.

La figura 4.11, presenta la imagen original con un sector de la misma contaminada con ruido gaussiano.



**Figura 4. 11.** Imágenes filtradas para los algoritmos a) LMS, b) NLMS y c) ARy afectada por ruido gaussiano en un sector.

Se aplicó el filtro adaptativo con los algoritmos utilizados a toda a la imagen, observando que solo actúa en el sector afectado por la interferencia.

# CAPÍTULO 5

## CONCLUSIONES Y TRABAJOS FUTUROS

### 5.1. INTRODUCCIÓN

En este capítulo se presentan las conclusiones, las recomendaciones y los trabajos futuros resultado de la realización de este trabajo de grado. Las conclusiones se basan en los objetivos propuestos, ejecución y análisis de las simulaciones. Las recomendaciones se dan acerca de la utilización de los algoritmos para filtrar el ruido que se presenta en una imagen y los trabajos futuros con respecto a la utilización de los mismos en diferentes estructuras.

### 5.2. CONCLUSIONES

- De acuerdo a los resultados arrojados en las simulaciones hechas, el algoritmo que presentó menor pérdida de información en el proceso de filtrado medido mediante el MSE, es el ARy con un valor de  $MSE = 2,72 \times 10^{-5}$ , seguido por el LMS con un valor de  $MSE = 3,2032 \times 10^{-5}$  y por último el NLMS con un valor de  $MSE = 8,9188 \times 10^{-5}$ .
- Mediante los datos obtenidos en simulaciones, el filtro adaptativo con el algoritmo ARy para procesamiento digital de imágenes presenta mayor velocidad de convergencia en comparación con los algoritmos LMS y NLMS, porque tomando un error cuadrático objetivo de  $8,8E-04$  el algoritmo ARy necesita de 17106 iteraciones para alcanzarlo, mientras los algoritmos NLMS y LMS necesitan 114975 y 14732 iteraciones respectivamente.

- Los parámetros del filtro adaptativo con el algoritmo ARy en procesamiento digital de imágenes funcionó de manera eficaz, bajo las condiciones de simulación de este trabajo de grado, cuando ellos cumplían con la relación  $\alpha \times \gamma \times m_1 \approx 0,0001697$ , esto se realizó mediante simulación, debido a que el proceso matemático para encontrar una relación más exacta esta fuera del alcance de este trabajo.

### 5.3. TRABAJOS FUTUROS

Mediante el desarrollo del presente trabajo de grado se identificaron nuevas investigaciones referentes a la implementación de un filtro adaptativo para reducir el ruido en imágenes.

- Implementar el filtro adaptativo para reducir el ruido presente en imágenes en movimiento (Video).

# REFERENCIAS BIBLIOGRÁFICAS

- [1] Rafael C. Gonzales and Richard E. Woods, *Digital Image Processing*. New Jersey, EEUU: Prentice-Hall, Inc., 2002.
- [2] Lorena Arranz. (2005, Octubre) Ministerio de Educación, Cultura y Deporte. [Online]. <http://recursostic.educacion.es/observatorio/web/es/software/software-general/293-lorena-arranz>
- [3] Leticia Flores Pulido. (2014, Febrero) wordpress. [Online]. [https://aicitel.files.wordpress.com/2011/08/clase3\\_pdi.pdf](https://aicitel.files.wordpress.com/2011/08/clase3_pdi.pdf)
- [4] Fotonostra. [Online]. <http://www.fotonostra.com/glosario/escalagrisescayscale.htm>
- [5] Fotonostra. FotoNostra: Fotografía y diseño gráfico digital. [Online]. <http://www.fotonostra.com/grafico/rgb.htm>
- [6] Maixu61. (2007, Enero) slideshare. [Online]. <http://es.slideshare.net/maixu61/caractersticas-de-la-imagen-digital>
- [7] Steffen Schirmer. Foto-Mosaik-Edda. [Online]. [http://www.fmedda.com/es/article/dpi\\_ppi](http://www.fmedda.com/es/article/dpi_ppi)
- [8] Juliana Villamonte. (2010, Septiembre) Slideshare. [Online]. <http://es.slideshare.net/Julianalsola/ruido-digital-en-la-fotografia>
- [9] Rodrigo Hernandez Navarrete, *Procesamiento digital de imagenes en ultrasonido*. Mexico, Mexico, 2010.
- [10] B Aldalur and M Santamaria, "Realce de imagenes: Filtraje espacial," *Revista de teledetección*, no. 17, pp. 31-42, Junio 2002.
- [11] Monson H. Hayes, *Statistical Digital Signal Proce*. Atlanta, Georgia: john wiley & sons, 1996.
- [12] S.C. Douglas, "Introduction to adaptive filters," in *Digital Signal Processing Handbook*. Salt Lake, EEUU: Vijay K. Madisetti and Douglas B. Williams, 1999, ch. 18.
- [13] Irina Dornean, Marina Topa, Botond Sandor Kirei, and Erwin Szopos, "System identification with least mean square adaptive algorithm," in *Interdisciplinary in engineering, Scientific international conference*, Mures, Romania, 2007, pp. IV-1-1, IV-1-4.

- [14] Gyanendra Singh, Kiran Savita, and Shivkumar Yadav, "DESIGN OF ADAPTIVE NOISE CANCELLER USING LMS ALGORITHM," *International Journal of Advanced Technology & Engineering Research*, vol. 3, pp. 85-89, Mayo 2013.
- [15] Jashvir Chhikara and Jagbir Singh, "Noise cancellation using adaptive lagorithms," *International Journal of Modern Engineering Research*, vol. 2, pp. 792-795, Mayo-Junio 2012.
- [16] Walter Leopoldo Zelaya Chicas, "Diseño de un filtro digital adaptativo como cancelador de ruido basado en el algoritmo LMS," Universidad del Salvador, El Salvador, Trabajo de Grado 2004.
- [17] Simon Haykin, "Least Mean Square," in *Communication Systems 3Rd Ed*, Wiley India Pvt. Limited, Ed., ch. 9, pp. 365-443.
- [18] Radhika Chinaboina, D.S.Ramkiran , and Habibulla Khan, "ADAPTIVE ALGORITHMS FOR ACOUSTIC ECHO CANCELLATION IN," *International Journal of Recent Research and Applied Studies*, vol. 7, pp. 38-42, Abril 2011.
- [19] Pablo Emilio Jojoa Gómez, UM ALGORITMO ACELERADOR DE PARAMETROS, 2003, Tesis Doctoral, Escuela Politécnica de la Universidad de Sao Paulo, Brasil.
- [20] Virginia Solarte Muñoz and Pablo Emilio Jojoa Gómez, "El Algoritmo Acelerador Regresivo versión  $\gamma$  (AR $\gamma$ ) y," *Universitaria en Telecomunicaciones Informática y Control*, vol. 1, no. 2, pp. 29-37, Noviembre 2012.
- [21] Evelio H. Astaiza, "MODELOS DE SIMULACIÓN," in *SIMULACIÓN DE SISTEMAS DE TELECOMUNICACIONES*. Popayan, Colombia: Padilla Bejarano, 2007, ch. 2, pp. 17-21.
- [22] Mike. (2014, Enero) Wikipedia. [Online].  
[http://commons.wikimedia.org/wiki/File:Barn\\_grand\\_tetons\\_rgb\\_separation.png](http://commons.wikimedia.org/wiki/File:Barn_grand_tetons_rgb_separation.png)

# APÉNDICES

## APÉNDICE A. ARCHIVOS DISEÑO ALGORITMOS ADAPTATIVOS

### A.1. ALGORITMO LMS

```
function [A,Y,E] = lms(x,d,mu,P,w)
%x - señal de entrada
%d - señal deseada o señal de referencia
%mu - factor de convergencia del algoritmo LMS
%P - Numero de coeficientes
%w - Valor inicial de coeficientes
N=length(x);
Y=zeros(1,N);
A=zeros(N+1,P);
E=Y; % reservamos para Y[] y E[]
A(P,:)=w;
rP=0:-1:-P+1;
for k=P:N,
X=x(k+rP); % ultimas P entradas x[k], x[k-1], ... x[k-P]
Y(k)=X*A(k,:)'; % salida del filtro: convolucion de x y h
E(k)=d(k)-Y(k); % error
A(k+1,:)=A(k,:)+mu*E(k)*conj(X); % actualización coeficientes del filtro
end
```

### A.2. ALGORITMO NLMS

```
function [A,Y,E] = nlms(x,d,beta,P,w)
%x - señal de entrada
%d - señal deseada o señal de referencia
%beta - factor de convergencia del algoritmo NLMS
%P - Numero de coeficientes
%w - Valor inicial de coeficientes
N=length(x);
A=zeros(N+1,P);
Y=zeros(1,N);
E=Y; % reservamos para Y[] y E[]
A(P,:)=w;
rP=0:-1:-P+1;
for k=P:N,
X=x(k+rP); % ultimas P entradas x[k], x[k-1], ... x[k-P]
Y(k)=X*A(k,:)'; % salida del filtro: convolucion de x y h
E(k)=d(k)-Y(k); % error
DEN=X*X' + 0.0001;
A(k+1,:)=A(k,:)+beta/DEN*E(k)*conj(X); % actualización coeficientes del
filtro
end
```

### A.3. ALGORITMO AR $\gamma$

```
function [wn,ea] = ARgama(x,d,al,ga,m1,M,w)
%x - señal de entrada
```

## Implementación de un filtro adaptativo para reducir el ruido en una imagen

---

```
%d - señal deseada o señal de referencia
%al - parámetro de ajuste fijo  $\alpha$  del algoritmo ARY
%ga - parámetro de ajuste fijo  $\gamma$  del algoritmo ARY
%m1 - parámetros de ajuste fijo  $m_1$  del algoritmo ARY
%P - Numero de coeficientes
%w - Valor inicial de coeficientes
N = 50000;
qn = zeros([M N]);
wn = zeros([M N]);
wn(:,M-1)=w;
rP=0:-1:-M+1;

for n=M:N
    X=x(n+rP); % ultimas P entradas x[k], x[k-1], ... x[k-P]
    xn=X';
    eal = xn'*wn(:,n-1)-d(n);
    g1 = eal+ga*(xn'*qn(:,n-1));
    g2 = 1+al*ga*m1*(xn'*xn);
    gn = g1/g2;
    qn(:,n) = ga*(qn(:,n-1)-al*m1*gn*xn)/(al+ga);
    wn(:,n) = wn(:,n-1)+al*qn(:,n);

    %almacenamiento de datos
    ea(n) = -eal;
end
```