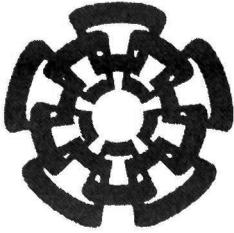




xx(131346.1)



# CINVESTAV

Centro de Investigación y de Estudios Avanzados del I.P.N.  
Unidad Guadalajara

---

## **Algoritmos para Optimización de Extracción y Mapeo de Información Geofísica Mediante Procesamiento de Datos de Percepción Remota**

**CINVESTAV  
IPN  
ADQUISICION  
DE LIBROS**

Tesis que presenta:

**Roberto Carrasco Álvarez**

para obtener el grado de:

**Maestro en Ciencias**

en la especialidad de:

**Ingeniería Eléctrica**

Director de Tesis

**Dr. Yuriy Shkvarko Sosnoff**

**CINVESTAV I.P.N.  
SECCION DE INFORMACION  
Y DOCUMENTACION**

Guadalajara, Jalisco, Septiembre de 2006.

CLASIF.: KIGJ.GB .C37 2006
ADQUIS.: S91-410
FECHA: 15-V-2007
PROCED.: Don. 2007
\$

18' 130732-1001

# **Algoritmos para Optimización de Extracción y Mapeo de Información Geofísica Mediante Procesamiento de Datos de Percepción Remota**

**Tesis de Maestría en Ciencias  
Ingeniería Eléctrica**

Por:

**Roberto Carrasco Álvarez**  
Ingeniero en Electrónica

Instituto Tecnológico de Morelia 1999-2004

Becario de Conacyt, expediente no. 191357

Director de Tesis  
**Dr. Yuriy Shkvarko Sosnoff**

CINVESTAV del IPN Unidad Guadalajara, Septiembre de 2006.

# RESUMEN

En el siguiente trabajo se presenta una serie de algoritmos computacionales desarrollados para la reconstrucción de señales y imágenes de percepción remota, las cuales han sido contaminadas con ruido gaussiano y suavizadas con una función de dispersión (PSF) conocida, que cuentan con alguna característica en particular que puede ser impuesta como una restricción durante el desarrollo del algoritmo como información a priori.

Los algoritmos están basados en el concepto del algoritmo Weigh Constrained Least Square (WCLS) el cual tiene sus bases teóricas en el algoritmo propuesto por Tikhonov, La modificación realizada es la de imponer restricciones lineales, las cuales se pueden ver como información a priori de las señales a reconstruir, este tipo de algoritmos permiten la reconstrucción con alta resolución y superresolución lo cual es una ventaja cuando se trabaja con imágenes de percepción remota ya que permite ver con mas detalle las imágenes obtenidas.

Una de las características de los algoritmos mencionados anteriormente, es que presentan la ventaja de que pueden ser calibrados ajustando ciertos grados de libertad para obtener una mejor reconstrucción, así que una de las metas de este trabajo de tesis es encontrar el algoritmo optimo para poder sintonizar estos parámetros y así obtener la mejor reconstrucción posible, lo cual no es una tarea fácil y es un área en la que todavía se sigue investigando al respecto, en este trabajo se presentará la implementación del método nuevo de la curva L para encontrar estos parámetros y una forma para su implementación de manera iterativa, además se presentarán los resultados obtenidos con este método y los obtenidos de forma iterativa.

# AGRADECIMIENTOS

Ante todo agradezco a mis padres por brindarme su apoyo durante todos mis años de estudio y así poder obtener la preparación necesaria para poder lograr mis metas.

Además quiero agradecer a todas las personas cercanas a mí por darme ánimo para seguir adelante, y estimularme cuando creía que no podía continuar.

De igual manera agradezco a mi asesor el Dr. Yuriy Shkvarko por orientarme a través del desarrollo de esta tesis y aportarme sus conocimientos.

Y finalmente agradezco mucho a CONACYT por el apoyo brindado ya que sin su valiosa ayuda esto no habría sido posible.

# TABLA DE CONTENIDOS

<b>RESUMEN</b>	<b>3</b>
<b>AGRADECIMIENTOS</b>	<b>4</b>
<b>TABLA DE CONTENIDOS</b>	<b>5</b>
<b>ÍNDICE DE FIGURAS</b>	<b>7</b>
<b>1. INTRODUCCIÓN</b>	<b>9</b>
1.1 Antecedentes	9
1.2 Procesamiento de imágenes	10
1.3 Percepción Remota	15
1.4 Modelo del Problema	21
1.5 Objetivos	24
1.6 Organización de la tesis	25
1.7 Conclusiones	25
<b>2 Capítulo II</b>	<b>27</b>
2.1 Filtro de Wiener	27
2.2 Método Bayesiano	33
2.3 Conclusiones	39
<b>3 Capítulo III</b>	<b>40</b>
3.1 Planteamiento del problema	40
3.2 Regularización de señales	42
3.3 Regularización de Tikhonov	44
3.4 Evaluación del desempeño	46
3.5 Implementación computacional	46
3.6 Conclusiones	48
<b>4 Capítulo IV</b>	<b>49</b>
4.1 Planteamiento del Problema y Análisis de Información a Priori	49
4.2 Método de Lagrange	50
4.3 Desarrollo Matemático	52
4.4 Evaluación del desempeño	55
4.5 Implementación Computacional	56
4.6 Implementación para el Procesamiento de Imágenes	57
4.7 Conclusiones	59
<b>5 Capítulo V</b>	<b>60</b>
5.1 Introducción al método de la Curva L	60

5.2	Planteamiento Matemático de la Curva L	61
5.3	Solución numérica el método de la Curva L	66
5.4	Esquema iterativo para la solución de la curva L	69
5.5	Implementación con redes neuronales	72
5.6	Conclusiones	79
<b>Resultados</b>		<b>81</b>
<b>Conclusiones</b>		<b>126</b>
<b>Bibliografía</b>		<b>128</b>
<b>Apéndice A</b>		<b>130</b>

# ÍNDICE DE FIGURAS

Figura 1.1: Demostración del efecto de aliasing en una señal de una dimensión, la línea gruesa es la señal reconstruida con las muestras de la señal original.....	10
Figura 1.2: Generación de imágenes digitales. ....	11
Figura 1.3: Efecto de Moire, a) Imagen original con dos patrones periódicos, b) Cada cuarta muestra de la imagen original, c) Cada quinta muestra de la imagen original.....	11
Figura 1.4: Digitalización de una imagen (a) Imagen analógica, (b) Imagen digitalizada, (c) Cuantización de cada píxel .....	12
Figura 1.5: Resolución de la imagen digitalizada a diferentes frecuencias de muestreo.....	13
Figura 1.6: Operaciones de realzado de imágenes.....	13
Figura 1.7: Proceso inherente para el procesamiento y análisis de imágenes.....	14
Figura 1.8: Formas de obtención de imágenes de percepción remota .....	16
Figura 1.9: Antena de RADAR.....	17
Figura 1.10: Tipos existentes de sensores de percepción remota y sus características.....	18
Figura 1.11: Variables biofísicas capaces de medirse a través de percepción remota.....	20
Figura 1.12: Fenomenología del problema a tratar .....	21
Figura 1.13: Modelos matemáticos para algunos fenómenos en la percepción de imágenes. ....	22
Figura 2.1: Respuesta del operador obtenido y de la función de sistema inversa en el dominio de la frecuencia .....	32
Figura 2.2: Esquema computacional del filtro de Wiener .....	32
Figura 2.3: Esquema computacional del método de riesgo mínimo de Bayes. ....	38
Figura 3.1: Formación de la matriz toeplitz de laPSF.....	41
Figura 3.2: Esquema computacional del método WCLS.....	47
Figura 5.1: Curva L típica para un parámetro de regularización, el círculo rojo corresponde al punto con mayor curvatura. ....	63
Figura 5.2: Curvatura Gaussiana de la curva L de la figura 5.1 .....	64
Figura 5.3: Gráfica de la distancia cuadrática entre la señal original y la reconstruida, el círculo rojo corresponde al valor con mayor curvatura de la gráfica 5.1 .....	64

Figura 5.4: Gráfica de la función de distancia entre el punto $(a,b)$ y cualquier punto de la curva $L$ .....	67
Figura 5.5: Esquema de una neurona. ....	73
Figura 5.6: Red neuronal propuesta por Wang para resolver un problema de optimización no lineal con restricciones. ....	77
Figura 5.7: Red neuronal implementada para minimizar la función de distancia.....	78

# INTRODUCCIÓN

## 1.1 Antecedentes

La estimación y reconstrucción de señales es aplicable en todas las áreas del conocimiento en las cuales se deben de tratar con datos, los cuales han sido degradados debido al proceso que se utilizó durante su obtención, mediante una transformación lineal y la adición de ruido, aplicaciones interesantes son en el campo de la medicina (tomografías, electrocardiogramas, radiografías, etc.), en comunicaciones como en telefonía celular y transmisión de datos, en ciencias de la tierra como oceanografía, meteorología, geología; en nuestro caso nos enfocamos más en este último ramo, al utilizar imágenes de percepción remota de la tierra; para la obtención de información tal como, el nivel de deforestación, cambios en los mantos acuíferos, cambios de las zonas urbanas, etc; lo cual permite tomar mejores decisiones acerca de las acciones que se deben de llevar a cabo para remediar algún problema.

Tomemos por ejemplo el problema de la deforestación, mediante una imagen de percepción remota se puede ver el cambio de dimensiones de un área boscosa en particular y evaluar si las medidas implementadas han sido efectivas para evitar la deforestación y cuales áreas hay que atender con más atención. Otras aplicaciones son mencionadas en [11], [12] y en [13] en las cuales mediante redes neuronales entrenadas y algoritmos basados en modelos estadísticos, se puede determinar a partir de una imagen de percepción remota de una ciudad, que elementos de la imagen son calles, cuales edificios, áreas verdes, etc.

Por todo lo anterior es importante el desarrollo de nuevos algoritmos para poder llevar a cabo de manera más eficiente y rápida la extracción de datos de interés a partir de imágenes de percepción remota, ya que estos procesos requieren el cálculo de muchas operaciones lo cual hace que a veces estos algoritmos sean prohibitivos de realizarse.

## 1.2 Procesamiento de imágenes

El término de procesamiento digital de imágenes generalmente se refiere al procesamiento de señales, generalmente de dos dimensiones (aunque puede extenderse la teoría para señales n-dimensionales) por medio de una computadora digital. Como se menciona en [3] algunas aplicaciones del procesamiento digital de señales pueden ser clasificadas como:

- Representación y modelado de imágenes
- Realzado de imágenes
- Restauración de imágenes
- Análisis de imágenes
- Reconstrucción de imágenes

La representación y modelado de imágenes se refiere a la manera en la cual representamos la imagen a procesar, ya que trabajamos con computadoras digitales, las imágenes deben ser muestreadas y cuantizadas, lo cual se conoce como digitalización.

Rigiéndonos por lo establecido en el teorema de Nyquist [3] en el cual se establece que la frecuencia de muestreo debe ser mucho mayor que la frecuencia máxima de la señal a muestrear para así poder preservar la mayor cantidad de información de la imagen a procesar y evitar el efecto de aliasing el cual ocurre cuando muestreamos una señal a una frecuencia menor de la mayor frecuencia de la señal a muestrear y la señal reconstruida es un múltiplo en frecuencia de la señal original.

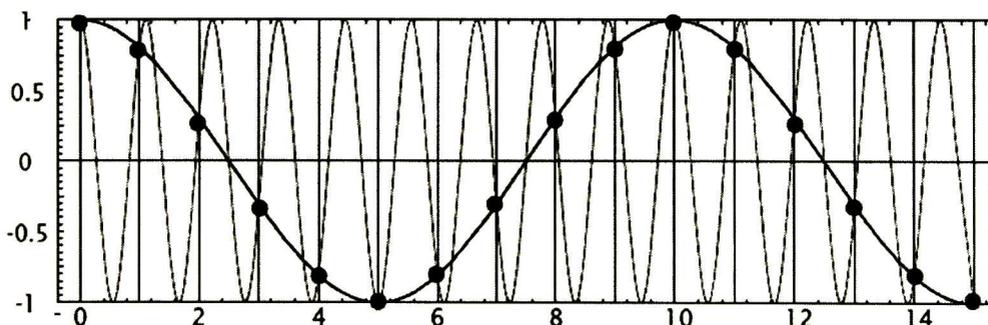


Figura 1.1: Demostración del efecto de aliasing en una señal de una dimensión, la línea gruesa es la señal reconstruida con las muestras de la señal original

Para el caso de imágenes podemos suponer que es un arreglo de señales de una dimensión, es decir las variaciones en los niveles de gris de la imagen, forman patrones que pueden ser expresados como señales.

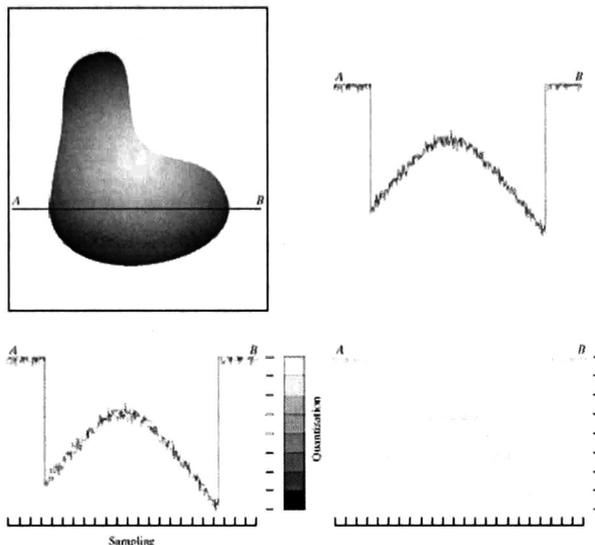


Figura 1.2: Generación de imágenes digitales.

El efecto de aliasing en las imágenes se puede observar en el efecto de moire, como se observa en la figura 1.3, se puede observar que al no elegir una frecuencia de muestreo adecuada la imagen reconstruida difiere mucho de la original.

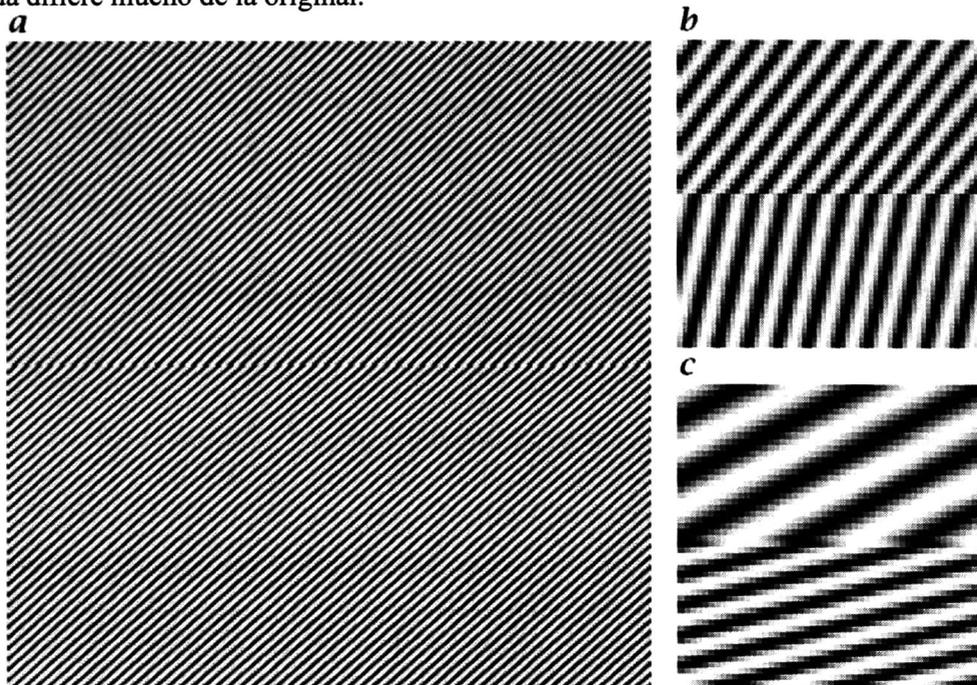


Figura 1.3: Efecto de Moire, a) Imagen original con dos patrones periódicos, b) Cada cuarta muestra de la imagen original, c) Cada quinta muestra de la imagen original

La cuantización es el proceso de convertir una imagen analógica muestreada a un número finito de valores, es decir cada muestra podrá ser representada con un valor predefinido, al realizar el proceso de digitalización la imagen se puede representar como una matriz en la cual cada elemento o píxel representa el nivel de luminiscencia de cada muestra de la imagen original muestreada.

En la figura 1.4 se puede observar el proceso de digitalización de una imagen analógica y en la figura 1.5 se puede observar el efecto que se tiene al muestrear la imagen analógica a diferentes frecuencias de muestreo, se puede observar que entre mayor sea la frecuencia de muestreo, la pérdida de información es menor en la imagen digitalizada lo cual hace que se obtenga una representación fiel de la imagen analógica original es decir una mejor resolución, aunque esto conlleva a que se tienen más datos a procesar, por lo tanto se debe encontrar un equilibrio entre la resolución de la imagen que vamos a procesar así como del gasto computacional que se requerirá para poder llevar a cabo el procesamiento.

Una explicación más a fondo del proceso de muestreo y cuantización puede ser encontrado en [3] y en [4].

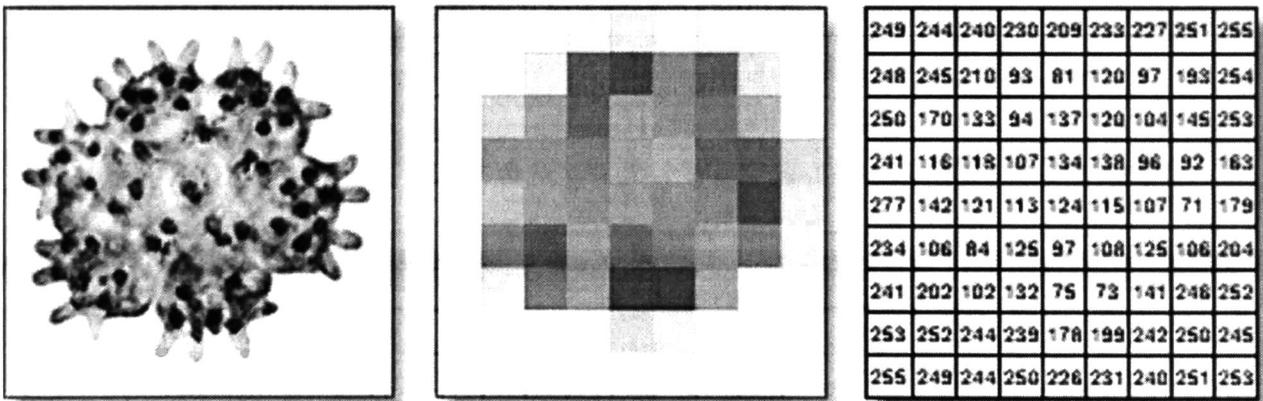


Figura 1.4: Digitalización de una imagen (a) Imagen analógica, (b) Imagen digitalizada, (c) Cuantización de cada píxel

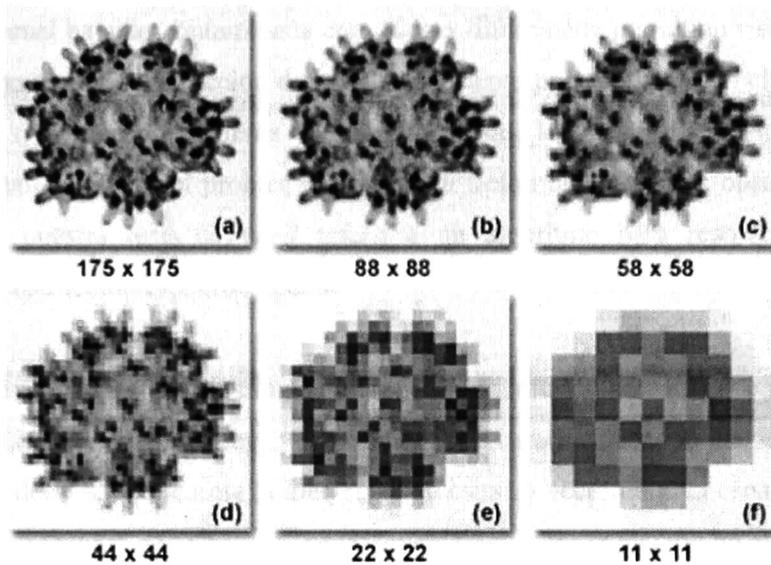


Figura 1.5: Resolución de la imagen digitalizada a diferentes frecuencias de muestreo

El realzado de imágenes se refiere al hecho de destacar ciertas características de las imágenes tales como lados, límites o contraste, para que su procesamiento pueda llevarse a cabo de una manera más sencilla, el realzado no aumenta la información inherente contenida en la imagen, algunas técnicas de realzado son manipulación del contraste, reducción de ruido, extracción de lados, filtrado, interpolación y magnificación (zoom), estas técnicas son enlistadas en la figura 1.6.

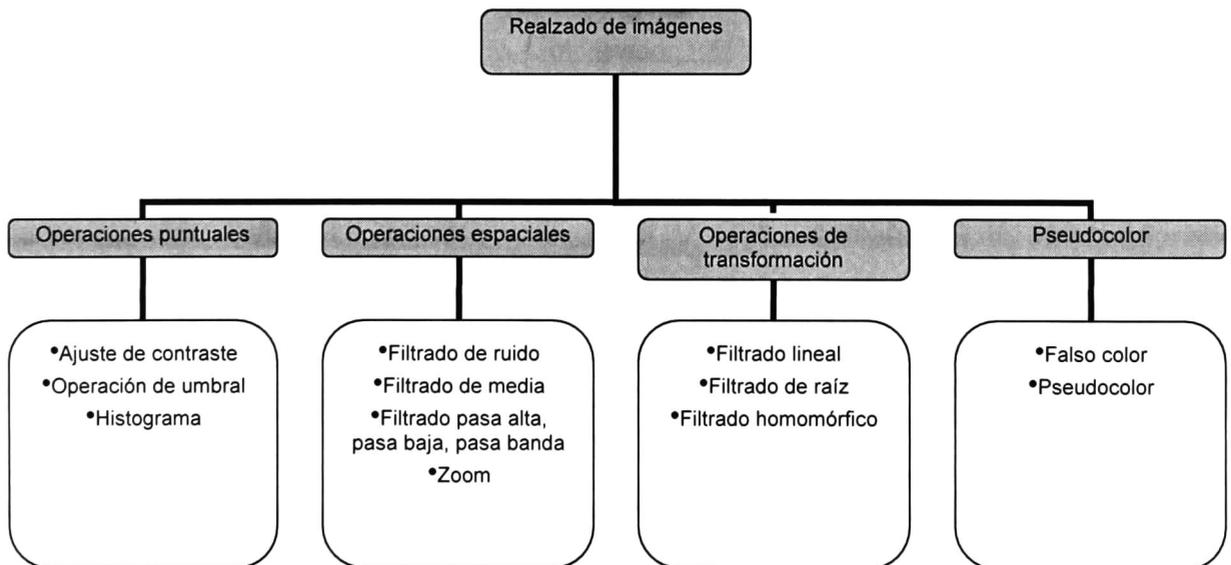


Figura 1.6: Operaciones de realzado de imágenes

La restauración de imágenes se refiere a poder obtener una imagen aproximadamente igual a partir de una imagen, la cual ha sido contaminada con ruido y difuminada por algún sistema, básicamente es la misma definición para reconstrucción de imágenes excepto por el espacio en el que se encuentra los datos observados, siendo en el problema de restauración que los datos observados están en el mismo espacio que la señal original y el problema de reconstrucción que los datos observados están en otro espacio vectorial, nuestra tesis se va a referir a un algoritmo para resolver los problemas de restauración y reconstrucción específicamente.

El análisis de imágenes es el campo del procesamiento de imágenes en el cual se extraen características de las señales e imágenes y a partir de estas características se pueden tener conclusiones acerca de estas, es decir se transforma la imagen de su espacio vectorial a un espacio de menor tamaño para una mejor comprensión de la imagen, generalmente esta transformación no es lineal y son difíciles de estimar ya que depende de las características que se desean obtener las cuales pueden ser muy subjetivas, en esta etapa es en la que se puede clasificar el grado de deforestación de un bosque, el crecimiento de una ciudad, o en una imagen determinar que componentes son calles, parques, etc.

Esta parte puede ser implementada con redes neuronales, lógica difusa, etc. Así se puede ver que para llevar a cabo el procesamiento de una imagen se deben llevar a cabo en secuencia las etapas antes mencionadas tal y como se muestra en la figura 1.7.

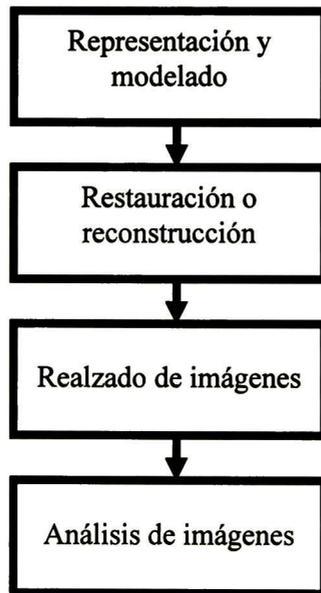


Figura 1.7: Proceso inherente para el procesamiento y análisis de imágenes

## 1.3 Percepción Remota

El objetivo de la percepción remota, es la de descubrir y analizar los fenómenos que nos rodean, y el conocimiento obtenido, utilizarlo para proteger el medio ambiente y para mejorar la calidad de vida del ser humano. Algunos de los fenómenos que se pueden observar y hacer mediciones son:

- Mundo físico (ej. La atmósfera, agua, tierra, rocas)
- Seres vivos (ej. Flora, fauna, ser humano)
- Procesos de actividades humanas (ej. Cantidad de desperdicios, deforestación, crecimiento urbano)

Para poder llevar a cabo lo anterior se puede hacer uso del sensado remoto, el cual consiste en medir o coleccionar información de alguna propiedad de un objeto o fenómeno usando un dispositivo que no esta físicamente o íntimamente en contacto con el objeto o fenómeno bajo estudio, de forma remota.

Para llevar acabo mediciones de fenómenos ocurridos en la tierra se utilizan, aeronaves suborbitales, satélites y vehículos aéreos no tripulados provistos con dispositivos, tales como cámaras y sensores, los cuales detectan la radiación electromagnética reflejada, emitida, o dispersada del objeto o área geográfica; y lo cual es utilizado como un sustituto de la propiedad bajo investigación.

La información obtenida puede ser almacenada en formato analógico o formato digital y analizada usando técnicas de procesamiento analógicas (observación) o técnicas de procesamiento digitales.

Durante el desarrollo de esta tesis las técnicas utilizadas son digitales ya que se hace uso de computadoras para poderlas llevar a cabo.

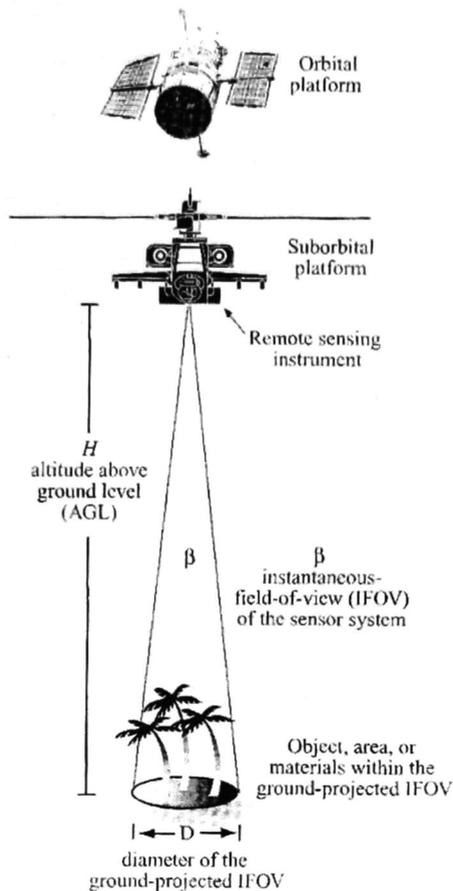


Figura 1.8: Formas de obtención de imágenes de percepción remota

Los sensores utilizados en el sensado remoto pueden ser clasificados en dos categorías:

### 1. Sensores pasivos

- Solo detectan la radiación electromagnética emitida o reflejada por el objeto o fenómeno bajo estudio, algunos ejemplos de este tipo de sensado son:
  - cámaras
  - scanners multispectrales
  - arreglos de sensores multispectrales.

### 2. Sensores activos

- Son aquellos que emiten una radiación electromagnética y detectan esta radiación después de haberse reflejado en el objeto o fenómeno bajo estudio, este tipo de sensado se considera invasivo ya que puede afectar dicho fenómeno, algunos ejemplos de este tipo de sensado es:
  - LIDAR – La señal emitida es un láser.

- SONAR – La señal emitida es sonido.
- RADAR – La señal emitida son microondas.

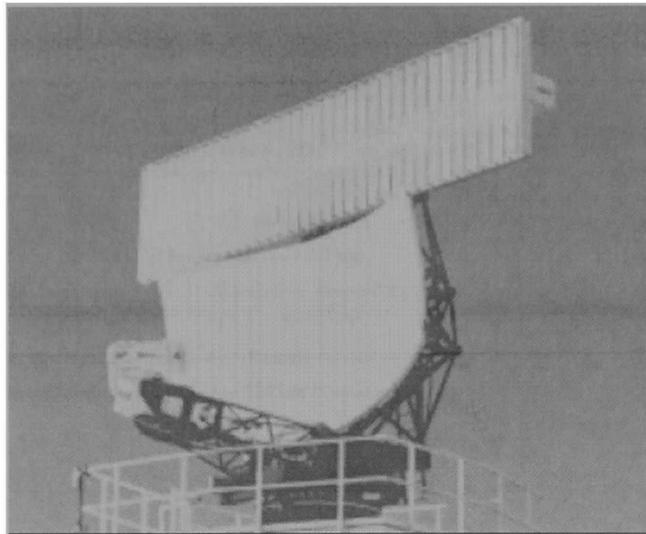


Figura 1.9: Antena de RADAR.

A su vez, los sistemas de sensado remoto pueden ser clasificados dependiendo de su resolución espectral y espacial. La resolución espectral es el un número y tamaño de intervalos de longitud de onda en el espectro electromagnético (referidos como bandas o canales) a los cuales el instrumento de sensado es sensible.

Se dice que el sensor es multispectral si es capaz de detectar múltiples bandas del espectro electromagnético tal como el sensor LANDSAT. Es hiperespectral si el sistema de sensado es capaz de recibir información de cientos de bandas del espectro electromagnético como el sistema (AVIRIS) el cual tiene 224 en la región de los 400 a los 2500 nm. Y es ultraspectral cuando el sistema de sensado es capaz de detectar varios cientos de bandas en el espectro electromagnético.

La resolución espacial es la medida de la separación angular o lineal mas pequeña entre dos objetos que el sensor puede sensar esta medida esta expresada como  $a \times a$  unidades de métricas, lo cual significa que el objeto más pequeño que puede ser detectado mide  $a$  unidades métricas.

En la figura 1.10 se presenta una lista de los sistemas de sensado remoto existentes, el espectro que son capaces de sensar y su resolución:

Remote Sensing Systems	Resolution									
	Spectral							Spatial (m)	Temporal (days)	
	Blue	Green	Red	Near-Infrared	Middle-Infrared (SWIR)	Thermal Infrared	Micro-wave			
<b>Suborbital Sensors</b>										
Panchromatic film (black & white)		0.5 — 0.7 $\mu\text{m}$						Variable	Variable	
Color film	0.4 — 0.7 $\mu\text{m}$							Variable	Variable	
Color-infrared film		0.5 — 0.9 $\mu\text{m}$						Variable	Variable	
Digital Frame Cameras (CCD)	1	1	1	1	—	—	—	0.25 – 5	Variable	
ATLAS - Airborne Terrestrial Applications Sensor	0.45 — 8 bands		— 2.35 $\mu\text{m}$		6	—	—	2.5 – 25	Variable	
AVIRIS - Airborne Visible Infrared Imaging Spectrometer	0.41 — 224 bands		— 2.5 $\mu\text{m}$					2.5 or 20	Variable	
Intermap Star-3i X-band radar							1	Variable	Variable	
<b>Satellite Sensors</b>										
NOAA-9 AVHRR LAC	—	—	1	1	—	3	—	1100	14.5/day	
NOAA- K, L, M	—	—	1	1	2	2	—	1100	14.5/day	
Landsat Multispectral Scanner (MSS)	—	1	1	2	—	—	—	79	16 – 18	
Landsat 4 and 5 Thematic Mappers (TM)	1	1	1	1	2	1	—	30 and 120	16	
Landsat 7 Enhanced TM (ETM <sup>+</sup> ) — Multispectral	1	1	1	1	2	1	—	30 and 60	16	
— Panchromatic	—	0.52 — 0.9 $\mu\text{m}$		—	—	—	—	15	16	
SPOT 4 HRV — Multispectral	—	1	1	1	—	—	—	20	Pointable	
— Panchromatic	—	0.51 — 0.73 $\mu\text{m}$		—	—	—	—	10	Pointable	
GOES Series (East and West)	—	0.52 — 0.72 $\mu\text{m}$		—	4	—	—	700	0.5/hr	
European Remote Sensing Satellite (ERS-1 and 2)	VV polarization C-band (5.3 GHz)							1	26 – 28	—
Canadian RADARSAT (several modes)	HH polarization C-band (5.3 GHz)							1	9 – 100	1 – 6 days
Shuttle Imaging Radar (SIR-C)	—	—	—	—	—	—	3	30	Variable	
Sea-Viewing Wide Field-of-View Sensor (SeaWiFS)	3	2	1	2	—	—	—	1130	1	
MODIS - Moderate Resolution Imaging Spectrometer	0.405 — 36 bands		— 14.385 $\mu\text{m}$		—	—	—	250, 500, 1000	1 – 2	
ASTER - Advanced Spaceborne Thermal Emission and Reflection Radiometer	0.52 — 3 bands		— 0.86 $\mu\text{m}$					15	5	
	1.6 – 6 bands		— 2.43 $\mu\text{m}$					30	16	
	8.12 – 5 bands		— 11.6 $\mu\text{m}$					90	16	
MISR - Multiangle Imaging SpectroRadiometer	Nine CCD cameras in four bands (440, 550, 670, 860 nm)							275 and 1100	1 – 2	
NASA Topex/Poseidon — TOPEX radar altimeter	(18, 21, 37 GHz)							315,000	10	
— POSEIDON single-frequency radiometer	(13.65 GHz)									
Space Imaging IKONOS — Multispectral	1	1	1	1			—	4	Pointable	
— Panchromatic	0.45 — 0.9 $\mu\text{m}$						—	1		
Digital Globe QuickBird — Multispectral	1	1	1	1			—	2.44	Pointable	
— Panchromatic	0.45 — 0.9 $\mu\text{m}$						—	0.61		

Figura 1.10: Tipos existentes de sensores de percepción remota y sus características

Las posibles características o información que pueden ser extraídas de los objetos o fenómenos sensados de manera remota se clasifican en dos grandes ramas:

- Variables biofísicas
- Variables híbridas

Las primeras son las que pueden ser medidas directamente de un sistema de percepción remota, esto significa que la medición provee suficiente información biológica o física sin hacer uso de otros datos, un ejemplo de esto es el sensado de la temperatura de una roca midiendo la energía irradiada en su superficie utilizando un sensor termal infrarrojo, similarmente es posible medir la cantidad de vapor de agua en la atmósfera, o la posición de un objeto por medio de una fotografía aérea estereoscópica.

Las variables híbridas son aquellas que pueden ser medidas al analizar más de una variable biofísica, por ejemplo se puede detectar stress de vegetación al medir características de absorción de la clorofila de las plantas, temperatura y contenido de humedad. En la figura 1.11 se presenta una relación de las variables biofísicas e híbridas que pueden ser sensadas y los respectivos sistemas de sensado de percepción remota para llevar a cabo esa tarea.

<b>Biophysical Variable</b>	<b>Potential Remote Sensing System</b>
<b>Snow and Sea Ice</b> - Extent and characteristics	- Color and CIR aerial photography, AVHRR, GOES, Landsat (TM, ETM <sup>+</sup> ), SPOT, RADARSAT, SeaWiFS, IKONOS, QuickBird, ASTER, MODIS
<b>Volcanic Effects</b> - Temperature, gases	- ASTER, MISR, Hyperion, MODIS, airborne hyperspectral systems
<b>BRDF (bidirectional reflectance distribution function)</b>	- MISR, MODIS, CERES
<b>Selected Hybrid Variables</b>	<b>Potential Remote Sensing System</b>
<b>Land Use</b> - Commercial, residential, transportation, utilities, etc. - Cadastral (property) - Tax mapping	- Very high spatial resolution panchromatic, color and /or CIR stereoscopic aerial photography, high spatial resolution satellite imagery (<1 m: IKONOS, QuickBird, OrbView-3), SPOT (2.5 m), LIDAR, high spatial resolution hyperspectral systems (e.g., AVIRIS, HyMap, CASI)
<b>Land Cover</b> - Agriculture, forest, urban, etc.	- Color and CIR aerial photography, Landsat (MSS, TM, ETM <sup>+</sup> ), SPOT, ASTER, AVHRR, RADARSAT, IKONOS, QuickBird, OrbView-3, LIDAR, IPSAR, SeaWiFS, MODIS, MISR, hyperspectral systems (e.g., AVIRIS, HyMap, CASI)
<b>Vegetation</b> - stress	- Color and CIR aerial photography, Landsat (TM, ETM <sup>+</sup> ), IKONOS, Quick-Bird, OrbView-3, AVHRR, SeaWiFS, MISR, MODIS, ASTER, airborne hyperspectral systems (e.g., AVIRIS, HyMap, CASI)

Biophysical Variable	Potential Remote Sensing System
<b>x, y, z Geodetic Control</b> <b>x, y Location from Orthorectified Imagery</b>	- Global Positioning Systems (GPS) - Analog and digital stereoscopic aerial photography, Space Imaging IKONOS, DigitalGlobe QuickBird, Orbimage OrbView-3, French SPOT HRV, Landsat (Thematic Mapper, Enhanced Thematic Mapper Plus), Indian IRS-1CD, European ERS-1 and 2 microwave, Canadian RADARSAT, LIDAR
<b>z Elevation</b> - Digital Elevation Model (DEM)  - Digital Bathymetric Model (DBM)	- GPS, stereoscopic aerial photography, LIDAR, SPOT, RADARSAT, IKONOS, QuickBird, OrbView-3, Shuttle Radar Topography Mission (SRTM), Interferometric Synthetic Aperture Radar (IFSAR) - SONAR, bathymetric LIDAR, stereoscopic aerial photography
<b>Vegetation</b> - Pigments (e.g., chlorophyll <i>a</i> and <i>b</i> )  - Canopy structure and height - Biomass derived from vegetation indices - Leaf area index (LAI) - Absorbed photosynthetically active radiation (APAR) - Evapotranspiration	- Color aerial photography, Landsat ETM <sup>+</sup> , IKONOS, QuickBird, OrbView-3, Orbimage SeaWiFS, Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER), Moderate Resolution Imaging Spectrometer (MODIS), airborne hyperspectral systems (e.g., AVIRIS, HyMap, CASI) - Large-scale stereoscopic aerial photography, LIDAR, RADARSAT, IFSAR - Color-infrared (CIR) aerial photography, Landsat (TM, ETM <sup>+</sup> ), IKONOS, QuickBird, OrbView-3, Advanced Very High Resolution Radiometer (AVHRR), Multiangle Imaging Spectroradiometer (MISR), airborne hyperspectral systems (e.g., AVIRIS, HyMap, CASI)
<b>Surface Temperature (land, water, atmosphere)</b>	- ASTER, AVHRR, Geostationary Operational Environmental Satellite (GOES), Hyperion, MODIS, SeaWiFS, airborne thermal infrared sensors
<b>Soil and Rocks</b> - Moisture  - Mineral composition - Taxonomy  - Hydrothermal alteration	- ASTER, passive microwave (SSM/I), RADARSAT, MISR, ALMAZ, Landsat (TM, ETM <sup>+</sup> ), ERS-1 and 2, Intermap Star 3i - ASTER, MODIS, hyperspectral systems (e.g., AVIRIS, HyMap, CASI) - High-resolution color and CIR aerial photography, airborne hyperspectral systems (e.g., AVIRIS, HyMap, CASI) - Landsat (TM, ETM <sup>+</sup> ), ASTER, MODIS, airborne hyperspectral systems (e.g., AVIRIS, HyMap, CASI)
<b>Surface Roughness</b>	- Aerial photography, ALMAZ, ERS-1 and 2, RADARSAT, Intermap Star 3i, IKONOS, QuickBird, ASTER, Envisat ASAR
<b>Atmosphere</b> - Aerosols (e.g., optical depth) - Clouds (e.g., fraction, optical thickness) - Precipitation  - Water vapor (precipitable water) - Ozone	- MISR, GOES, AVHRR, MODIS, CERES, MOPITT - GOES, AVHRR, MODIS, MISR, CERES, MOPITT, UARS - Tropical Rainfall Measurement Mission (TRMM), GOES, AVHRR, passive microwave (SSM/I) - GOES, MODIS - MODIS
<b>Water</b> - Color - Surface hydrology - Suspended minerals - Chlorophyll/gelbstoffe - Dissolved organic matter	- Color and CIR aerial photography, Landsat (TM, ETM <sup>+</sup> ), SPOT, IKONOS, QuickBird, OrbView-3, ASTER, SeaWiFS, MODIS, airborne hyperspectral systems (e.g., AVIRIS, HyMap, CASI), AVHRR, GOES, bathymetric LIDAR, MISR, CERES, Hyperion, TOPEX/POSEIDON, Envisat-1 MERIS

Figura 1.11: Variables biofísicas capaces de medirse a través de percepción remota

Algunas de las ventajas de utilizar sensado remoto es que puede ser no intrusivo si es que se utilizan sensores pasivos que solo detectan la energía reflejada o emitida por el fenómeno de interés, además puede proveer información biofísica fundamental tal como localización, biomasa, temperatura,

etc; otra ventaja es que puede cubrir un área geográfica grande, lo cual hace que sea más fácil modelar numerosos procesos naturales.

## 1.4 Modelo del Problema

El problema al que nos enfrentamos durante la restauración o reconstrucción de imágenes se muestra de manera gráfica en la figura 1.12

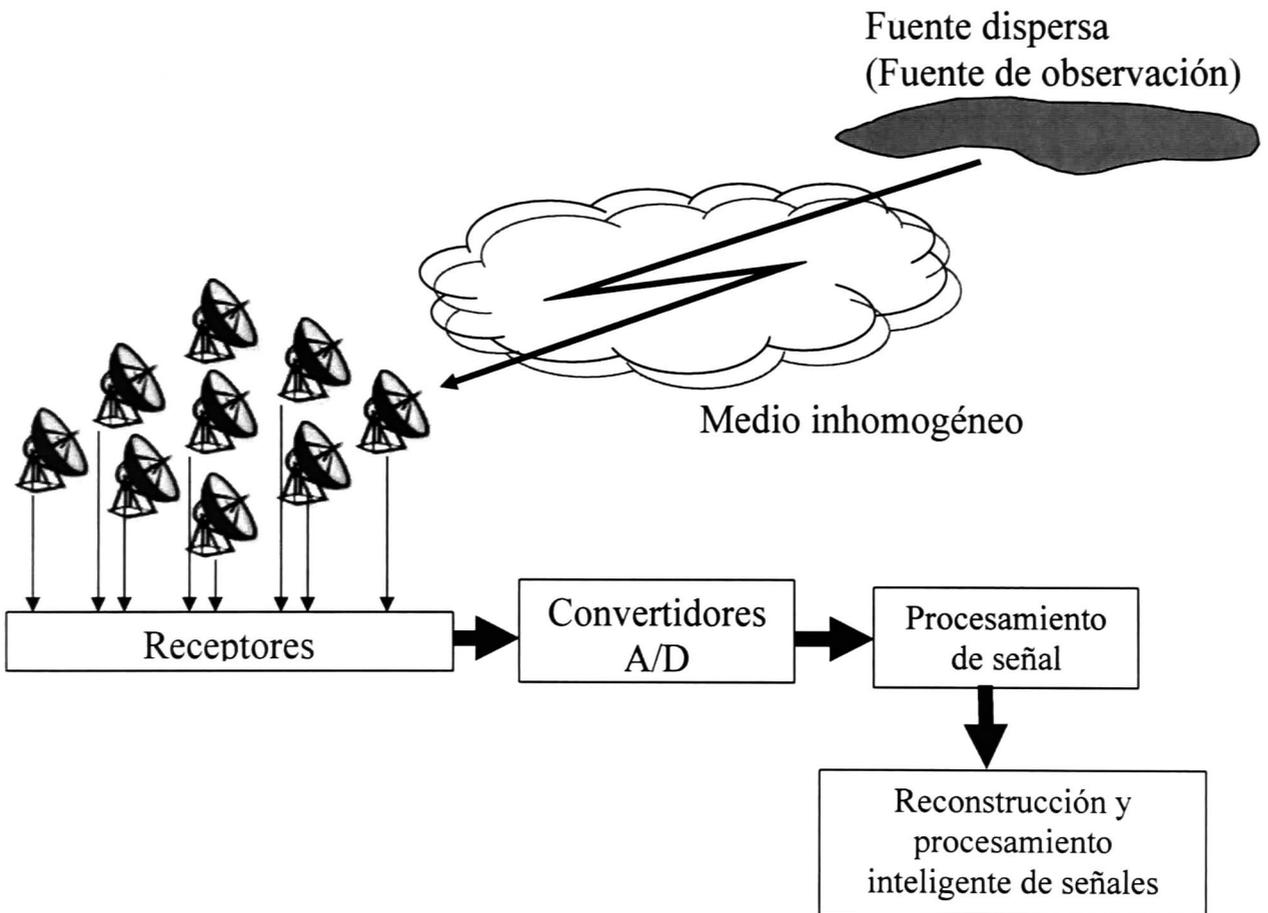


Figura 1.12: Fenomenología del problema a tratar

En el proceso representado en la figura se trata de observar la señal electromagnética radiada de una fuente dispersa mediante un arreglo de sensores, la señal que provee la fuente dispersa es transformada

o difuminada por el medio inhomogéneo, en el caso de imágenes de percepción remota este medio inhomogéneo es el todo aquello que se interpone entre los sensores y la imagen a obtener como por ejemplo nubes, corrientes de aire, lluvia, contaminación, partículas suspendidas en el ambiente, la óptica de los lentes de los sensores, en la figura 1.13 se muestra una relación de algunos modelos matemáticos para procesos que afectan la percepción de imágenes, se puede observar que están definidos en dos dimensiones, una mayor explicación de estas funciones de transferencia es mostrada en [4], aunque generalmente estas funciones de transferencia son no lineales y son muy difíciles de modelar matemáticamente, aunque existen técnicas tal y como se menciona en [3].

Una vez que en los sensores se tiene la imagen original transformada por el medio inhomogéneo, esta señal es contaminada con ruido gaussiano aditivo debido al ruido térmico introducido por los sensores. La función de los receptores es la de acondicionar la señal recibida para poderla restaurar. Para poder llevar acabo la restauración o reconstrucción debemos convertir la imagen o señal obtenida que es analógica a un formato digital, una vez que se tiene esta señal digital somos capaces de implementar algoritmos computacionales para implementar la restauración.

Type of system	Impulse response $h(x, y)$	Frequency response $H(\xi_1, \xi_2)$
Diffraction limited, coherent (with rectangular aperture)	$ab \operatorname{sinc}(ax) \operatorname{sinc}(by)$	$\operatorname{rect}\left(\frac{\xi_1}{a}, \frac{\xi_2}{b}\right)$
Diffraction limited, incoherent (with rectangular aperture)	$\operatorname{sinc}^2(ax) \operatorname{sinc}^2(by)$	$\operatorname{tri}\left(\frac{\xi_1}{a}, \frac{\xi_2}{b}\right)$
Horizontal motion	$\frac{1}{\alpha_0} \operatorname{rect}\left(\frac{x}{\alpha_0} - \frac{1}{2}\right) \delta(y)$	$e^{-j\pi\xi_1\alpha_0} \operatorname{sinc}(\xi_1\alpha_0)$
Atmospheric turbulence	$\exp\{-\pi\alpha^2(x^2 + y^2)\}$	$\frac{1}{\alpha^2} \exp\left[-\frac{\pi(\xi_1^2 + \xi_2^2)}{\alpha^2}\right]$
Rectangular scanning aperture	$\operatorname{rect}\left(\frac{x}{\alpha}, \frac{y}{\beta}\right)$	$\alpha\beta \operatorname{sinc}(\alpha\xi_1) \operatorname{sinc}(\beta\xi_2)$
CCD interactions	$\sum_{k,l=-1}^1 \alpha_{k,l} \delta(x - k\Delta, y - l\Delta)$	$\sum_{k,l=-1}^1 \alpha_{k,l} e^{-j2\pi\Delta(\xi_1k + \xi_2l)}$

Figura 1.13: Modelos matemáticos para algunos fenómenos en la percepción de imágenes.

El fenómeno anterior puede ser modelado matemáticamente mediante la ecuación de observación, que en forma operacional es:

$$u = Sv + n \quad v \in V_N \quad u \in U_L \quad S: V_N \rightarrow U_L \quad (1.1)$$

donde  $u$  es la señal o imagen observada,  $S$  es la función del sistema que nos difumina la señal original,  $v$  es la señal o imagen original y  $n$  es ruido gaussiano aditivo.

Si suponemos que la función de sistema  $S$  es invariante en el tiempo podemos expresar la ecuación de observación en forma integral tal y como se muestra en la ecuación 1.2.

$$u(y) = s(y) + n(y) = \int_x S(y-x)v(x)dx + n(y) \quad (1.2)$$

Para poder trabajar con computadoras digitales es necesario digitalizar la ecuación de observación, la representación en forma discreta de esta ecuación, puede representarse en forma vectorial como un arreglo de mediciones tal y como se muestra en la ecuación 1.3.

$$\mathbf{u} = \mathbf{S}\mathbf{v} + \mathbf{n} \quad (1.3)$$

donde  $\mathbf{u}$  es el vector de datos observados,  $\mathbf{S}$  es la matriz de transformación de  $\mathbf{v}$  conocida también como *point spread function* (PSF),  $\mathbf{v}$  es el vector de datos originales y  $\mathbf{n}$  es el vector de ruido gaussiano aditivo.

El problema que se plantea ahora es cómo llevar a cabo la restauración; la manera más sencilla que se nos viene a la mente es la de aplicar la función inversa de  $S$  a  $u$  y así obtener  $v$  despreciando el ruido  $n$  que se añadió durante la medición, en forma matemática:

$$v = S^{-1}u$$

y en forma discreta:

$$\mathbf{v} = \mathbf{S}^{-1}\mathbf{u}$$

Pero al realizar lo anterior se presentan varios problemas, por ejemplo pequeñas variaciones del ruido hacen que  $\mathbf{u}$  varíe y pueden hacer que la señal estimada cambie drásticamente, otro problema que

puede surgir es de que  $S$  sea singular es decir no tenga inversa, cuando se presentan estos problemas se dice que el problema esta mal planteado y se deben proponer otras formas de regularización para que el problema este bien planteado.

## 1.5 Objetivos

El objetivo principal planteado es el de encontrar algoritmos de reconstrucción eficientes en los cuales se puedan imponer restricciones lineales como información a priori.

Como objetivos secundarios se plantearon los siguientes:

- Realizar una comparación del algoritmo desarrollado con algún método de regularización existente.
- Encontrar una manera efectiva para estimar los parámetros de regularización óptimos del algoritmo de reconstrucción desarrollado, con los cuales se pueda obtener la mejor reconstrucción, para lograr esto se planteara el desarrollo de un algoritmo basado en las proposiciones hechas en [1], [17], sobre la hipersuperficie  $L$ , lo cual se explicará en capítulos posteriores.
- Tratar de obtener un algoritmo computacionalmente eficiente del método de la curva  $L$ , es decir tratar de realizar el menor número de operaciones posibles y tratar de que éstas se procesen lo más rápido posible.
- Por último presentar un análisis cuantitativo y cualitativo de las reconstrucciones obtenidas con el algoritmo computacional.

## 1.6 Organización de la tesis

En el capítulo II se presentará un panorama de los métodos existentes de reconstrucción y estimación, además de la manera en la cual trabajan y una breve explicación matemática de ellos.

En el capítulo III se presentará un introducción del método de regularización de Tikhonov, en el cual esta basado nuestro algoritmo computacional, así como los conceptos preliminares para poder desarrollar el algoritmo planteado.

En el capítulo IV se presentará el desarrollo matemático del algoritmo implementado así como un análisis matemático de su desempeño, además se presentará un esquema computacional para su implementación.

En el capítulo V se presentará una introducción del método de la curva e hipersuperficie L para obtener los parámetros de regularización óptimos, además se mostrará un método numérico para obtener estos valores a partir de la curva L y la integración del algoritmo desarrollado con el método de la curva L para obtener los parámetros de regularización y los resultados obtenidos.

Por último se presentarán conclusiones y comentarios acerca del trabajo.

## 1.7 Conclusiones

En este capítulo se dio una breve introducción acerca de procesamiento de imágenes, la cual es un área muy extensa que nos permite representar, restaurar, reconstruir y analizar, imágenes y señales, esto es posible gracias a la implementación de algoritmos computacionales, así la elaboración de algoritmos eficientes y rápidos conducirá a un mejor procesamiento de imágenes. Por otra parte se explicaron los conceptos básicos de percepción remota, se observó que es de gran importancia ya que con ella se puede analizar el comportamiento de fenómenos naturales y sociales en la tierra sin la necesidad de dispositivos que estén en contacto físico con el fenómeno a estudiar, y por ello es una herramienta muy útil cuando se desea realizar estudios de grandes porciones de territorio, así la combinación percepción

remota y procesamiento de imágenes proveen herramientas muy poderosas para el estudio de fenómenos en la tierra.

Por último se presento el fenómeno al que nos enfrentamos cuando queremos reconstruir una imagen, se vio que la señal observada se encuentra degradada por ruido gaussiano y difuminada por un PSF que modela de forma matemática la degradación de la señal por el medio ambiente y todos los dispositivos utilizados para observar la señal, por eso es de suma importancia conocer esta PSF, para el caso de los algoritmos desarrollados en esta tesis suponemos que conocemos la PSF.

# Capítulo II

## Panorama de los métodos existentes

Existen varios métodos de regularización para poder plantear el problema mal planteado en uno bien planteado, estos métodos trabajan en el mismo dominio de la imagen a restaurar o en otros dominios como el de la frecuencia, además algunos plantean el problema de forma determinística y otros lo hacen de manera estadística. En estos métodos se trata de minimizar la distancia que existe entre la señal original y la señal estimada lo cual es planteado como un funcional el cual debe de ser minimizado. Algunos de estos métodos son mencionados en [3] y [2]:

- Filtro de Wiener
- Interpolación con splines
- Filtros recursivos (Kalman)
- Métodos Bayesianos
- Mínimos cuadrados (Tikhonov)

A continuación se hará una mención del filtrado de Wiener y el método Bayesiano; para el caso de interpolación con splines y filtro de Kalman refiérase a [3] el método de mínimos cuadrados será explicado ampliamente en el siguiente capítulo.

### 2.1 Filtro de Wiener

En el caso de este método de regularización se trabajará en el dominio de la frecuencia para poder resolver el problema mal planteado de la estimación de señales. Para ello se hará uso de la transformada de Fourier discreta o de su algoritmo para implementarse de manera eficiente que es la FFT, una vez obtenido el espectro de la señal de datos observados se implementará el filtrado óptimo de Wiener y se transformará la señal resultante al dominio del tiempo mediante la transformada inversa de Fourier IDFT.

Para el caso de este método consideramos que el sistema que transforma la señal de información es tal que  $H:V(x) \rightarrow V(x)$  es decir transforma la señal el mismo espacio vectorial, y esta operación sobre  $v(x)$  esta definida en forma integral como:

$$s(y) = \int_x h(y-x)v(x)dx \quad (2.1)$$

la cual es la convolución entre la señal  $v(x)$  y el sistema  $h(x)$

Primero plantearemos el modelo matemático:

Consideremos la ecuación de observación en tiempo continuo:

$$u(y) = s(y) + n(y) = \int_x h(y-x)v(x)dx + n(y) \quad (2.2)$$

En forma de vectores la ecuación anterior tiene la forma:

$$\mathbf{u} = \mathbf{H}\mathbf{v} + \mathbf{n} \quad (2.3)$$

donde la matriz de sistema  $\mathbf{H}$  es de forma Toeplitz, esto significa que cada elemento de  $\mathbf{u}$  puede ser expresado en forma de una convolución discreta:

$$u(k) = \sum_n H(k-n)v(n) + n(k) \quad k, n = 1, \dots, N \quad (2.4)$$

Una forma de representar estas ecuaciones de observación de forma compacta es de la siguiente forma:

$$\text{Forma continua} \quad u(x) = h(x) \otimes v(x) + n(x) \quad (2.5)$$

$$\text{Forma discreta} \quad u(k) = h(k) \otimes v(k) + n(k) \quad (2.6)$$

Usando la transformada de Fourier la ecuación de observación se puede expresar como:

$$U(f) = H(f)V(f) + N(f) \quad (2.7)$$

donde:

$$U(f) \Leftrightarrow u(x) \quad H(f) \Leftrightarrow h(x) \quad V(f) \Leftrightarrow v(x) \quad N(f) \Leftrightarrow n(x) \quad (2.8)$$

son las pares de transformada.

Así el problema es encontrar un filtro óptimo  $W(f)$  en el dominio de la frecuencia dada la ecuación de observación, que al ser aplicada a los datos medidos produzca un estimado óptimo:

$$\hat{V}(f) = W(f)U(f) \Leftrightarrow \hat{v}(x) = w(x) \otimes u(x) \quad (2.9)$$

Como información a priori consideramos que la señal  $v(x)$  y el ruido  $n(x)$  son procesos aleatorios estacionarios independientes con medias cero funciones de correlación  $R_v(x)$   $R_n(x)$  respectivamente y espectros de densidad de potencia:

$$\begin{aligned} P_v(f) &\Leftrightarrow R_v(x) \\ P_n(f) &\Leftrightarrow R_n(x) \end{aligned} \quad (2.10)$$

La estrategia de solución es encontrar un operador lineal  $W(f)$  que produzca la restauración de la señal de una manera óptima siendo en el sentido LMS, para esto se definirá el criterio de media cuadrada en el dominio de la frecuencia utilizando el teorema de Parseval como:

$$r(W) = \left\langle \|\hat{v}(x) - v(x)\|^2 \right\rangle = \left\langle \|W(f)U(f) - V(f)\|^2 \right\rangle \quad (2.11)$$

Así el filtro óptimo será el argumento mínimo de  $r(W)$ :

$$W(f) = \arg \min_w r(W) \quad (2.12)$$

Expresando la función de media cuadrada como productos internos en el espacio de funciones quedaría expresada como:

$$\begin{aligned}
 r(W) &= \langle \|W(f)U(f) - V(f)\|_{L_2}^2 \rangle = \langle \| [W(f)H(f) - 1]V(f) - W(f)N(f) \|_{L_2}^2 \rangle = \\
 &= \langle \int \{ [W(f)H(f) - 1]V(f) - W(f)N(f) \} \{ [W(f)H(f) - 1]V(f) + W(f)N(f) \}^* df \rangle = \\
 &= \int |W(f)H(f) - 1|^2 P_v(f) df - \int |W(f)|^2 P_n(f) df
 \end{aligned} \tag{2.13}$$

la cual esta compuesta de dos tipos de errores:

el error sistemático:

$$r_s(W) = \int |W(f)H(f) - 1|^2 P_v(f) df \tag{2.14}$$

el error de ruido:

$$r_n(W) = \int |W(f)|^2 P_n(f) df \tag{2.15}$$

el cual este último representa la energía del ruido a la salida del operador de estimación  $W(f)$ .

Para obtener el algoritmo derivamos la función de media cuadrada con respecto a  $W(f)$  y lo igualamos a cero teniendo así la ecuación de variación:

$$\frac{\partial r(W)}{\partial W(f)} = 2H^*(f) \{ [W(f)H(f) - 1]P_v(f) + W(f)P_n(f) \} = 0$$

Reordenando y haciendo manipulaciones matemáticas esta ecuación nos conduce a la solución óptima deseada:

$$W(f) = \frac{H^*(f)}{|H(f)|^2 + \frac{P_n(f)}{P_v(f)}} = \frac{H^*(f)}{|H(f)|^2 + \frac{1}{\mu(f)}} \tag{2.16}$$

donde

$$\mu(f) = \frac{P_v(f)}{P_n(f)}$$

es la relación señal a ruido de los espectros de potencia.

Este operador se puede describir como:

$$\begin{aligned} W(f) &= \frac{H^*(f)}{|H(f)|^2 + \frac{P_n(f)}{P_v(f)}} = \frac{1}{H(f)} \frac{H^*(f) H(f)}{|H(f)|^2 + \frac{1}{\mu(f)}} = \frac{1}{H(f)} \frac{|H(f)|^2}{|H(f)|^2 + \frac{1}{\mu(f)}} = \\ &= H_{inv}(f) R_{\pi}(f) \end{aligned} \quad (2.17)$$

donde:

$$H_{inv}(f) = \frac{1}{H(f)}$$

es el filtro inverso y:

$$R_{\pi}(f) = \frac{|H(f)|^2}{|H(f)|^2 + \frac{1}{\mu(f)}}$$

es el filtro óptimo de regularización.

Para la evaluación del desempeño evaluamos el operador obtenido en la función de media cuadrada, es decir en la de error sistemático y error de ruido.

$$r_s(W) = \int |R_{\pi}(f) - 1|^2 P_v(f) df$$

$$r_n(W) = \int |R_{\pi}(f)|^2 |H(f)|^2 P_n(f) df$$

$$\begin{aligned} r(W) &= r_s(W) + r_n(W) = \int |R_{\pi}(f) - 1|^2 P_v(f) df + \int |R_{\pi}(f)|^2 |H(f)|^2 P_n(f) df = \\ &= \int \frac{P_v(f) \left(\frac{1}{\mu(f)}\right)^2}{(|H(f)|^2 + \frac{1}{\mu(f)})^2} df + \int \frac{P_n(f) |H(f)|^2}{(|H(f)|^2 + \frac{1}{\mu(f)})^2} df = \int \frac{P_n(f)}{|H(f)|^2 + \frac{1}{\mu(f)}} df \end{aligned}$$

(2.18)

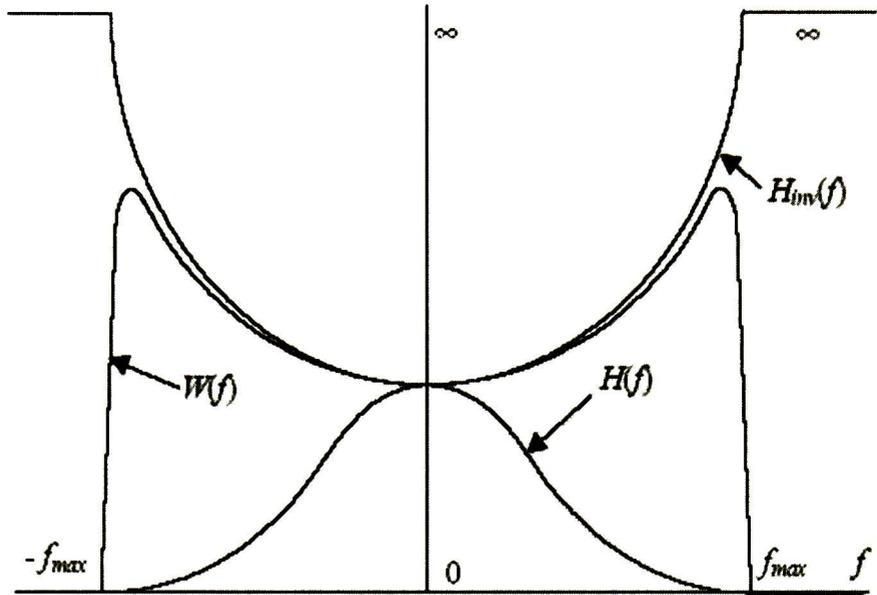


Figura 2.1: Respuesta del operador obtenido y de la función de sistema inversa en el dominio de la frecuencia

Un esquema de la implementación computacional es presentado a continuación:

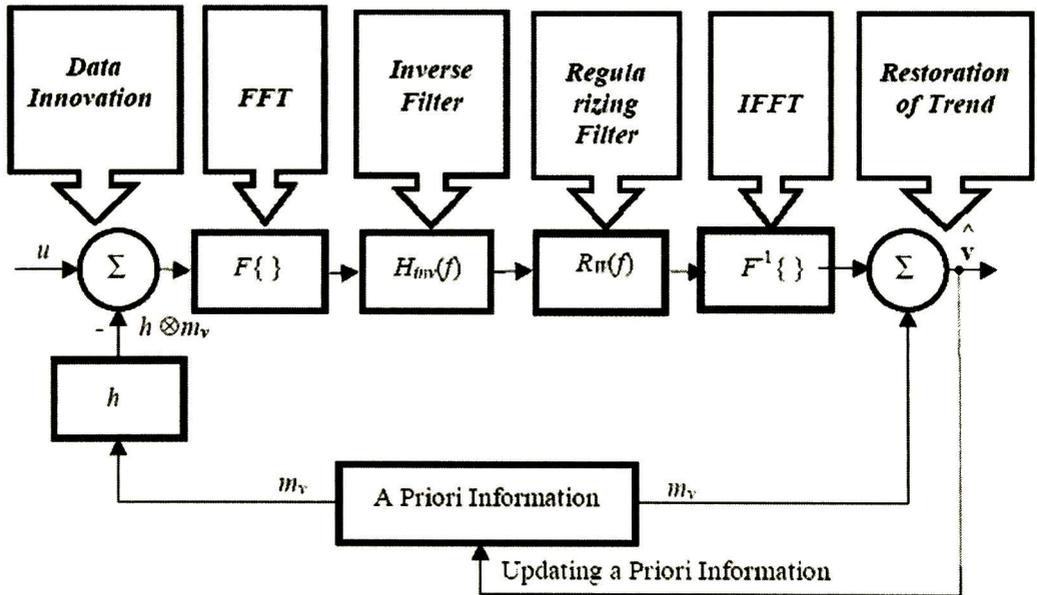


Figura 2.2: Esquema computacional del filtro de Wiener

Se puede observar que consta de varias etapas que son:

Innovación de la señal de datos

Transformación rápida de Fourier (FFT) de los datos innovados

Filtrado inverso utilizando  $H_{inv}(f)$

Regularización del filtro inverso usando el filtro de regularización

Transformación inversa de Fourier (IFFT)

- Restauración de la tendencia

## 2.2 Método Bayesiano

En el método de riesgo mínimo de Bayes, se considera información a priori estadística, tal como el primer y segundo momento de la señal de información y del ruido; y se desea encontrar un operador lineal óptimo tal que al aplicarse los datos observados nos entregue la mejor estimación.

Primero planteamos el modelo matemático del problema:

Consideremos la ecuación de observación:

$$\mathbf{u} = \mathbf{S}\mathbf{v} + \mathbf{n} \quad (2.19)$$

donde el vector  $\mathbf{v}$  es el vector de señal de entrada y  $\mathbf{S}$  es la matriz del sistema,  $\mathbf{n}$  es el vector de ruido aleatorio en las observaciones del vector  $\mathbf{u}$ .

Dado el modelo anterior se desea encontrar un operador  $\mathcal{W} : U \rightarrow V$  que opere sobre  $\mathbf{u}$  y produzca el estimado  $\hat{\mathbf{v}} = \mathcal{W}\mathbf{u}$

Como información a priori consideramos que la media del vector  $\mathbf{v}$  es:

$$\mathbf{m}_{\mathbf{v}} = \langle \mathbf{v} \rangle \quad (2.20)$$

y su matriz de covarianza es:

$$\mathbf{R}_v = \langle (\mathbf{v} - \mathbf{m}_v)(\mathbf{v} - \mathbf{m}_v)^+ \rangle \quad (2.21)$$

además se tiene que la media del vector de ruido aleatorio es cero esto es:

$$\mathbf{m}_n = \langle \mathbf{n} \rangle = \mathbf{0} \quad (2.22)$$

y su matriz de correlación es:

$$\mathbf{R}_n = \langle \mathbf{n} \mathbf{n}^+ \rangle \quad (2.23)$$

La estrategia de solución es encontrar un operador lineal  $\mathbf{W}$  que produzca la restauración de la señal de una manera óptima siendo definido del riesgo mínimo de Bayes.

Para simplificar el trabajo se supondrá que  $\mathbf{m}_v = \mathbf{0}$  para así obtener el operador lineal  $\hat{\mathbf{v}} = \mathbf{W}\mathbf{u}$  donde  $\mathbf{W}$  es la matriz solución, en el caso de que la media sea diferente de cero se encontrará el operador suponiendo que la media es cero y luego se realizará la siguiente modificación:

$$\hat{\mathbf{v}} = \mathbf{m}_v + \mathbf{W}(\mathbf{u} - \mathbf{S}\mathbf{m}_v) \quad (2.24)$$

Primero se introducirá una función de pérdida  $L(\mathbf{v}, \hat{\mathbf{v}}) = L(\mathbf{v}, \mathbf{W}\mathbf{u})$  la cual es la discrepancia entre la señal de información y su estimado y la cual esta definida como:

$$L(\mathbf{v}, \mathbf{W}\mathbf{u}) = \|\mathbf{W}\mathbf{u} - \mathbf{v}\|^2 \quad (2.25)$$

Además se define una función de riesgo la cual es:

$$r(\mathbf{W}) = \langle L(\mathbf{v}, \mathbf{W}\mathbf{u}) \rangle = \langle \|\mathbf{W}\mathbf{u} - \mathbf{v}\|^2 \rangle \quad (2.26)$$

Así nuestra estrategia será encontrar el operador  $\mathbf{W}$  que haga que la función de riesgo sea mínima es decir:

$$\mathbf{W} = \arg \min_{\mathbf{W}} r(\mathbf{W}) = \arg \min_{\mathbf{W}} \langle \|\mathbf{W}\mathbf{u} - \mathbf{v}\|^2 \rangle \quad (2.27)$$

Expresando la función de riesgo como productos internos tenemos que:

$$\begin{aligned} r(\mathbf{W}) &= \langle \|\mathbf{W}\mathbf{u} - \mathbf{v}\|^2 \rangle = \langle \|\mathbf{W}(\mathbf{S}\mathbf{v} + \mathbf{n}) - \mathbf{v}\|^2 \rangle = \\ &= \langle [(\mathbf{W}(\mathbf{S}\mathbf{v} + \mathbf{n}) - \mathbf{v}), (\mathbf{W}(\mathbf{S}\mathbf{v} + \mathbf{n}) - \mathbf{v})] \rangle = \\ &= \langle \mathbf{v}^{\sim} (\mathbf{W}\mathbf{S} - \mathbf{I})^{\dagger} (\mathbf{W}\mathbf{S} - \mathbf{I}) \mathbf{v} \rangle + \langle \mathbf{n}^{\sim} \mathbf{W}^{\sim} \mathbf{W} \mathbf{n} \rangle . \end{aligned} \quad (2.28)$$

Haciendo uso de las siguientes propiedades:

$$\mathbf{x}^{\sim} \mathbf{A} \mathbf{x} = \text{tr} \{ \mathbf{A} \mathbf{x} \mathbf{x}^{\dagger} \}$$

$$\text{tr} \{ \mathbf{ABC} \} = \text{tr} \{ \mathbf{BCA} \} = \text{tr} \{ \mathbf{CAB} \}$$

Podemos expresar la función de riesgo como:

$$\begin{aligned} r(\mathbf{W}) &= \text{tr} \{ (\mathbf{W}\mathbf{S} - \mathbf{I})^{\dagger} (\mathbf{W}\mathbf{S} - \mathbf{I}) \langle \mathbf{v}\mathbf{v}^{\dagger} \rangle \} + \{ \mathbf{W}^{\sim} \mathbf{W} \langle \mathbf{n}\mathbf{n}^{\sim} \rangle \} \\ &= \text{tr} \{ (\mathbf{W}\mathbf{S} - \mathbf{I}) \mathbf{R}_v (\mathbf{W}\mathbf{S} - \mathbf{I})^{\dagger} \} + \text{tr} \{ \mathbf{W} \mathbf{R}_n \mathbf{W}^{\sim} \} \end{aligned} \quad (2.29)$$

la cual esta compuesta de dos tipos de errores:

el error sistemático:

$$r_s(\mathbf{W}) = \text{tr} \{ (\mathbf{W}\mathbf{S} - \mathbf{I}) \mathbf{R}_v (\mathbf{W}\mathbf{S} - \mathbf{I})^{\dagger} \}$$

el error de ruido:

$$r_n(\mathbf{W}) = \text{tr} \{ \mathbf{W} \mathbf{R}_n \mathbf{W}^{\sim} \}$$

el cual este último representa la energía del ruido a la salida del operador de estimación  $\mathbf{W}$ .

Para obtener el algoritmo derivamos la función de riesgo con respecto a  $\mathbf{W}$  y lo igualamos a cero para hacer esto hacemos uso de la fórmula:

$$\frac{\partial \text{tr}\{\mathbf{W}\mathbf{C}\mathbf{W}^+\}}{\partial \mathbf{W}} = 2 \mathbf{W} \mathbf{C}$$

Esto conduce a:

$$\frac{\partial r(\mathbf{W})}{\partial \mathbf{W}} = 2 (\mathbf{W}\mathbf{S} - \mathbf{I})\mathbf{R}_v\mathbf{S}^+ + 2 \mathbf{W}\mathbf{R}_n = 0$$

Reordenando y haciendo manipulaciones matemáticas esta ecuación nos lleva a la primera forma del operador óptimo en el sentido de BMR el cual es:

$$\mathbf{W} = \mathbf{W}^{(1)} = \mathbf{R}_v\mathbf{S}^-\mathbf{R}_u^{-1} = \mathbf{R}_v\mathbf{S}^+(\mathbf{S}\mathbf{R}_v\mathbf{S}^+ + \mathbf{R}_n)^{-1} \quad (2.31)$$

Siendo así la primera solución:

$$\hat{\mathbf{v}} = \mathbf{W}^{(1)}\mathbf{u} = \mathbf{R}_v\mathbf{S}^+\mathbf{R}_u^{-1}\mathbf{u} = \mathbf{R}_v\mathbf{S}^+(\mathbf{S}\mathbf{R}_v\mathbf{S}^+ + \mathbf{R}_n)^{-1}\mathbf{u} \quad (2.32)$$

Haciendo uso de la fórmula de Frobenius podemos expresar  $\mathbf{R}_u^{-1}$  como:

$$\mathbf{R}_u^{-1} = (\mathbf{S}\mathbf{R}_v\mathbf{S}^- + \mathbf{R}_n)^{-1} = \mathbf{R}_n^{-1} - \mathbf{R}_n^{-1}\mathbf{S}\mathbf{K}\mathbf{S}^+\mathbf{R}_n^{-1} \quad (2.33)$$

donde:

$$\mathbf{K} = (\mathbf{R}_v^{-1} + \Psi)^{-1}$$

$$\Psi = \mathbf{S}^-\mathbf{R}_n^{-1}\mathbf{S}$$

Haciendo la sustitución en  $\mathbf{W}$  y acomodando tenemos el operador óptimo de la forma 2 como:

$$\begin{aligned}\mathbf{W} &= \mathbf{R}_v^{-1}(\mathbf{R}_n^{-1} - \mathbf{R}_n^{-1}\mathbf{S}\mathbf{K}\mathbf{S}^{-1}\mathbf{R}_n^{-1}) = \mathbf{R}_v^{-1}(\mathbf{I} - \mathbf{\Psi}\mathbf{K})\mathbf{S}^{-1}\mathbf{R}_n^{-1} = \\ &= \mathbf{R}_v^{-1}(\mathbf{K}^{-1} - \mathbf{\Psi})\mathbf{K}\mathbf{S}^{-1}\mathbf{R}_n^{-1} = \mathbf{K}\mathbf{S}^{-1}\mathbf{R}_n^{-1}\end{aligned}\quad (2.34)$$

Así la señal de información estimada será:

$$\begin{aligned}\hat{\mathbf{v}} &= \mathbf{W}\mathbf{u} = \mathbf{W}^{(2)}\mathbf{u} = (\mathbf{S}^{-1}\mathbf{R}_n^{-1}\mathbf{S} - \mathbf{R}_v^{-1})^{-1}\mathbf{S}^{-1}\mathbf{R}_n^{-1}\mathbf{u} \\ \hat{\mathbf{v}} &= \mathbf{m}_v + \mathbf{W}(\mathbf{u} - \mathbf{S}\mathbf{m}_v) = \mathbf{m}_v + (\mathbf{S}^{-1}\mathbf{R}_n^{-1}\mathbf{S} - \mathbf{R}_v^{-1})^{-1}\mathbf{S}^{-1}\mathbf{R}_n^{-1}(\mathbf{u} - \mathbf{S}\mathbf{m}_v)\end{aligned}\quad (2.35)$$

Para la evaluación del desempeño evaluamos el operador obtenido en la función de riesgo, es decir en la de error sistemático y error de ruido y así obtenemos:

$$\begin{aligned}r_s(\mathbf{W}) &= \text{tr}\{(\mathbf{W}\mathbf{S} - \mathbf{I})\mathbf{R}_v(\mathbf{W}\mathbf{S} - \mathbf{I})^{-1}\} = \text{tr}\{(\mathbf{K}\mathbf{\Psi} - \mathbf{I})\mathbf{R}_v(\mathbf{K}\mathbf{\Psi} - \mathbf{I})^{-1}\} = \\ &= \text{tr}\{\mathbf{K}(\mathbf{\Psi} - \mathbf{K}^{-1})\mathbf{R}_v(\mathbf{\Psi} - \mathbf{K}^{-1})\mathbf{K}\} = \text{tr}\{\mathbf{K}\mathbf{R}_v^{-1}\mathbf{R}_v\mathbf{R}_v^{-1}\mathbf{K}\} = \text{tr}\{\mathbf{K}\mathbf{R}_v^{-1}\mathbf{K}\}\end{aligned}$$

$$r_n(\mathbf{W}) = \text{tr}\{\mathbf{W}\mathbf{R}_n\mathbf{W}^{-1}\} = \text{tr}\{\mathbf{K}\mathbf{S}^{-1}\mathbf{R}_n^{-1}\mathbf{R}_n\mathbf{R}_n^{-1}\mathbf{S}\mathbf{K}\} = \text{tr}\{\mathbf{K}\mathbf{\Psi}\mathbf{K}\}$$

$$r(\mathbf{W}) = r_s(\mathbf{W}) - r_n(\mathbf{W}) = \text{tr}\{\mathbf{K}\mathbf{R}_v^{-1}\mathbf{K} + \mathbf{K}\mathbf{\Psi}\mathbf{K}\} = \text{tr}\{\mathbf{K}(\mathbf{R}_v^{-1} - \mathbf{\Psi})\mathbf{K}\} = \text{tr}\{\mathbf{K}\}\quad (2.36)$$

Un esquema de la implementación computacional es presentado a continuación:

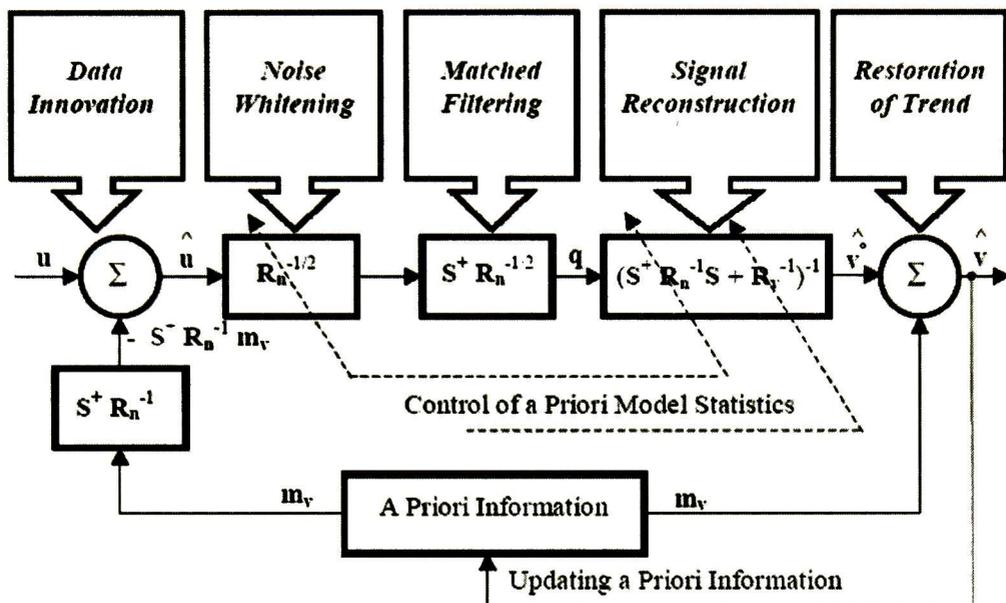


Figura 2.3: Esquema computacional del método de riesgo mínimo de Bayes.

Se puede observar que el operador  $W$  consta de tres fases las cuales son:

$$W = W_3 W_2 W_1 \tag{2.37}$$

donde:

$$\begin{aligned} W_1 &= R_n^{-1/2} \\ W_2 &= S^+ R_n^{-1/2} \\ W_3 &= K = (S^+ R_n^{-1} S - R_v^{-1})^{-1} \end{aligned} \tag{2.38}$$

La primera corresponde a la etapa de blanqueado del ruido, la segunda a la etapa de filtro óptimo y la tercera de la reconstrucción de la señal.

## 2.3 Conclusiones

En este capítulo se dio un panorama de los algoritmos de reconstrucción existentes, el primero que se mencionó fue el filtro de Wiener el cual hace uso de la transformada de Fourier y su transformada inversa para llevar a cabo la reconstrucción de la señal, esto puede ser una desventaja de este método ya que aunque existen algoritmos eficientes para la implementación de la transformada de Fourier tales como la FFT y la IFFT, esto implica realizar más operaciones.

El segundo algoritmo que se presentó fue el algoritmo de riesgo mínimo de Bayes (BMR) el cual consiste en formular todo el problema a partir de información estadística, y encontrar un operador solución que sea óptimo en sentido estadístico, la desventaja de este método es que se debe conocer la matriz de covarianza de la señal original, lo cual en algunos casos no es posible.

# Capítulo III

## Regularización de Tikhonov

### 3.1 Planteamiento del problema

Como se menciona en la introducción, el problema que se enfrenta, es el de restaurar o reconstruir una imagen o señal a partir de una señal observada la cual es una versión contaminada y difuminada de la señal original. Lo anterior se puede plantear en forma matemática con la siguiente ecuación de observación:

$$u = Sv + n \quad (3.1)$$

donde  $n$  es ruido aditivo gaussiano con media cero,  $H$  es la función de transferencia del medio en el que se propaga la señal original (PSF), y  $v$  es la señal original.

Al discretizar las señales para poder llevar a cabo su procesamiento la ecuación 3.1 puede ser expresada en forma vectorial como:

$$\mathbf{u} = \mathbf{S}\mathbf{v} + \mathbf{n} \quad (3.2)$$

donde la matriz  $\mathbf{S}$  es de forma Toeplitz, la cual generalmente esta mal planteada, la formación de esta matriz es explicada en [2],[3].

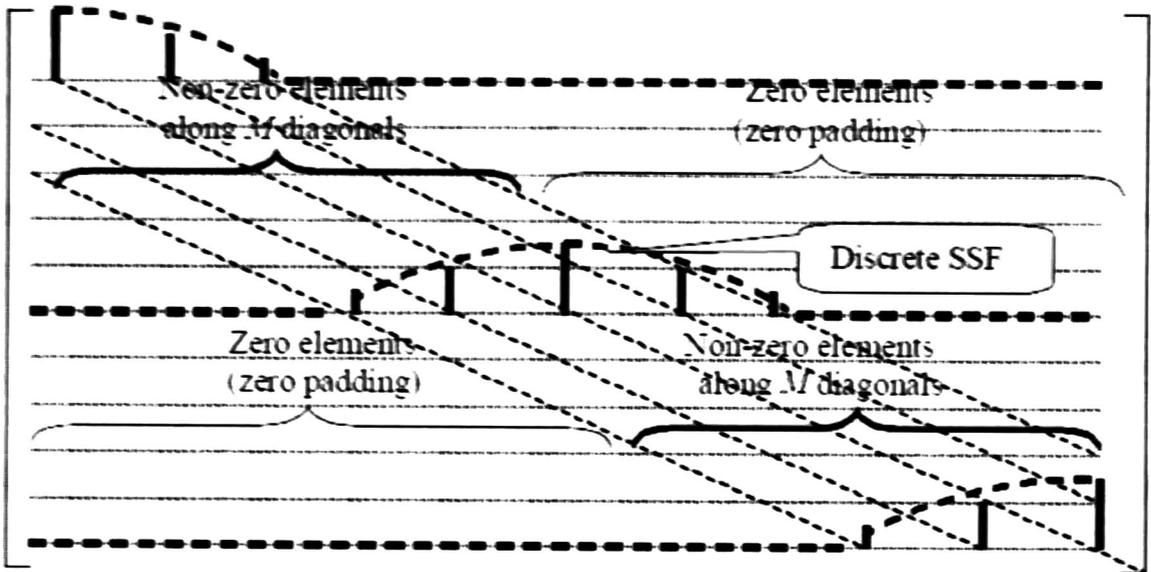


Figura 3.1: Formación de la matriz toeplitz de laPSF.

Como suponemos que el ruido es aditivo gaussiano, descorrelacionado y con media cero, podemos observar que las estadísticas del vector de ruido  $\mathbf{n}$  son:

$$\begin{aligned} \mathbf{m}_n &= \langle \mathbf{n} \rangle = \mathbf{0} \\ \mathbf{R}_n &= \langle (\mathbf{n} - \mathbf{m}_n)(\mathbf{n} - \mathbf{m}_n)^T \rangle = N_0 \mathbf{I} \end{aligned} \quad (3.3)$$

donde  $\mathbf{m}_n$  es el vector de media y  $\mathbf{R}_n$  es la matriz de covarianza la cual es una matriz identidad multiplicada por la varianza de cada muestra del ruido, esta forma de matriz es debido a que no existe correlación entre las diferentes muestras del vector de ruido.

El vector de la señal original  $\mathbf{v}$  tiene como estadísticas:

$$\begin{aligned} \mathbf{m}_v &= \langle \mathbf{v} \rangle \\ \mathbf{R}_v &= \langle (\mathbf{v} - \mathbf{m}_v)(\mathbf{v} - \mathbf{m}_v)^T \rangle \end{aligned} \quad (3.4)$$

Y el vector de la señal observada tiene como estadísticas:

$$\mathbf{m}_s = \langle \mathbf{Sv} + \mathbf{n} \rangle = \langle \mathbf{Sv} \rangle + \langle \mathbf{n} \rangle = \mathbf{S}\mathbf{m}_v \quad (3.5)$$

$$\begin{aligned} \mathbf{R}_u &= \langle (\mathbf{u} - \mathbf{m}_u)(\mathbf{u} - \mathbf{m}_u)^* \rangle = \langle (\mathbf{Sv} + \mathbf{n} - \mathbf{Sm}_v)(\mathbf{Sv} + \mathbf{n} - \mathbf{Sm}_v)^* \rangle = \langle (\mathbf{S}(\mathbf{v} - \mathbf{m}_v) + \mathbf{n})(\mathbf{S}(\mathbf{v} - \mathbf{m}_v) + \mathbf{n})^* \rangle = \\ &= \mathbf{S} \langle (\mathbf{v} - \mathbf{m}_v)(\mathbf{v} - \mathbf{m}_v)^* \rangle \mathbf{S}^* + \langle \mathbf{nn}^* \rangle = \mathbf{SR}_v \mathbf{S}^* + \mathbf{R}_n \end{aligned} \quad (3.6)$$

El problema a solucionar es encontrar un operador  $W$  tal que opere sobre  $\mathbf{u}$  y nos entregue una señal estimada  $\hat{\mathbf{v}}$  en un sentido óptimo:

$$\hat{\mathbf{v}} = W\mathbf{u} \quad (3.7)$$

## 3.2 Regularización de señales

Una manera para obtener una aproximación de la señal original es minimizar la distancia:

$$j(\mathbf{v}) = \|\mathbf{u} - \mathbf{Sv}\|_2^2 \quad (3.8)$$

Para obtener el mínimo obtenemos el gradiente del funcional anterior y lo igualamos a cero:

$$\frac{\partial j(\mathbf{v})}{\partial \mathbf{v}} = \frac{\partial}{\partial \mathbf{v}} \left( (\mathbf{u} - \mathbf{Sv})^T (\mathbf{u} - \mathbf{Sv}) \right) = \frac{\partial}{\partial \mathbf{v}} \left( \mathbf{u}^T \mathbf{u} - \mathbf{u}^T \mathbf{Sv} - \mathbf{v}^T \mathbf{S}^T \mathbf{u} + \mathbf{v}^T \mathbf{S}^T \mathbf{Sv} \right) = -2\mathbf{S}^T \mathbf{u} + 2\mathbf{S}^T \mathbf{Sv} = -2\mathbf{S}^T (\mathbf{u} - \mathbf{Sv}) = \mathbf{0}$$

$$\hat{\mathbf{v}} = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T \mathbf{u} \quad (3.9)$$

donde  $T$  denota matriz transpuesta si es que suponemos valores reales y matriz hermitiana si suponemos valores complejos. El resultado obtenido corresponde a la pseudo inversa de  $\mathbf{S}$  por el vector de observación  $\mathbf{u}$ .

Supongamos que conocemos la descomposición de valores singulares (SVD) [1], [14], [15], de  $\mathbf{S}$ ,

$$\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (3.10)$$

donde  $\mathbf{U}$ ,  $\mathbf{V}$  son ortogonales y  $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_n)$ ,  $\sigma_1 \geq \dots \geq \sigma_n \geq 0$  son los valores singulares de  $\mathbf{S}$ ; para  $\mathbf{S} \in R^{m \times n}$   $m \geq n$ ,  $\mathbf{U}$  es cuadrada de dimensión  $m \times m$ ,  $\mathbf{V}$  es una matriz cuadrada de dimensión  $n \times n$ , y  $\mathbf{\Sigma}$  es una matriz rectangular de tamaño  $m \times n$ . Así el operador obtenido anteriormente puede expresarse en términos de la descomposición singular como:

$$\begin{aligned}
\hat{\mathbf{v}} &= (\mathbf{V}\Sigma^T\mathbf{U}^T\mathbf{U}\Sigma\mathbf{V}^T)^{-1}(\mathbf{U}\Sigma\mathbf{V}^T)^T\mathbf{u} = \\
\mathbf{V}^{-T}(\Sigma^T\Sigma)^{-1}\mathbf{V}^{-1}\mathbf{V}\Sigma^T\mathbf{U}^T &= \mathbf{V}\Sigma^f\mathbf{U}^T\mathbf{u} = \\
\sum_{i=1, \sigma_i \neq 0}^n \frac{1}{\sigma_i} (\mathbf{u}_i^T \mathbf{g}_{ver} + \mathbf{u}_i^T \mathbf{n}) \mathbf{v}_i &
\end{aligned} \tag{3.11}$$

donde  $\mathbf{g}_{ver} = \mathbf{S}\mathbf{v}$ ,  $\mathbf{v}_i$ ,  $\mathbf{u}_i$  son las  $i$ -ésimas columnas de las matrices  $\mathbf{V}$  y  $\mathbf{U}$  respectivamente y:

$$\Sigma^f = \text{diag}(\sigma_i^f), \quad \sigma_i^f = \begin{cases} \frac{1}{\sigma_i} & \text{si } \sigma_i \neq 0 \\ 0 & \text{cualquier otro caso} \end{cases} \tag{3.12}$$

Si asumimos que  $\mathbf{g}_{ver}$  satisface la condición discreta de Picard [1]; el cual indica que los coeficientes generalizados de Fourier  $|\mathbf{u}_i^T \mathbf{g}_{ver}|$  de  $\mathbf{g}_{ver}$  decaen en promedio mas rápido que los valores singulares  $\sigma_i$ . Al asumir que el ruido es blanco, los coeficientes generalizados del ruido  $|\mathbf{u}_i^T \mathbf{n}|$  son constantes para toda  $i$  con valor medio:

$$\langle |\mathbf{u}_i^T \mathbf{n}| \rangle = \frac{\sigma_n}{n^{1/2}}, \quad i = 1, \dots, m \tag{3.13}$$

donde  $\sigma_n$  es la desviación estándar del ruido

Como consecuencia de esto, se puede observar que en la ecuación 3.11 la parte que va a dominar más es la que contiene al ruido dado que la otra se decremanta más rápido que los valores singulares y, por lo tanto, tiende a ser más pequeña esa parte con respecto a la otra; además, hay que tener en cuenta que las matrices mal condicionadas se caracterizan por tener valores singulares muy pequeños, por lo tanto, la señal reconstruida va a ser dominada por el término  $|\mathbf{u}_i^T \mathbf{n}|$  lo cual nos conlleva a que una pequeña variación del ruido produce una gran variación de la reconstrucción.

### 3.3 Regularización de Tikhonov

Para solucionar el problema planteado en el apartado anterior, en donde el ruido domina la señal reconstruida, se propone un método para regularizar el problema y hacerlo así bien planteado. Esto se logra con la regularización de Tikhonov, en el cual se propone el siguiente funcional a minimizar

$$J(\mathbf{v}) = J_1(\mathbf{v}) + J_2(\mathbf{v}) \quad (3.14)$$

donde:

$$J_1(\mathbf{v}) = \|\mathbf{u} - \mathbf{Sv}\|_U^2 = [(\mathbf{u} - \mathbf{Sv}), (\mathbf{u} - \mathbf{Sv})]_U = (\mathbf{u} - \mathbf{Sv})^+ \mathbf{M}_U (\mathbf{u} - \mathbf{Sv})$$

$$J_2(\mathbf{v}) = \|\mathbf{v} - \mathbf{m}_v\|_V^2 = [(\mathbf{v} - \mathbf{m}_v), (\mathbf{v} - \mathbf{m}_v)]_V = (\mathbf{v} - \mathbf{m}_v)^+ \mathbf{M}_V(\boldsymbol{\alpha})(\mathbf{v} - \mathbf{m}_v)$$

Es decir, se trata de minimizar la distancia cuadrada entre la señal observada y la señal reconstruida transformada por la PSF en el espacio inducido por  $U$ , el cual se propone como una matriz  $\mathbf{M}_U$ , y además se trata de minimizar la distancia entre la señal reconstruida y la media de la señal original en el espacio inducido por  $V$  lo cual se representa por la matriz  $\mathbf{M}_V(\boldsymbol{\alpha})$  la cual es definida como:

$$\mathbf{M}_V(\boldsymbol{\alpha}) = \alpha_1 \mathbf{R}_1^T \mathbf{R}_1 + \dots + \alpha_i \mathbf{R}_i^T \mathbf{R}_i \quad (3.15)$$

donde  $\alpha_i$  son los parámetros de regularización y  $\mathbf{R}_i$  son los operadores de regularización, usualmente estos operadores de regularización son escogidos para que sean la matriz identidad o una aproximación discreta de un operador de derivación, aunque existen aplicaciones en donde estos operadores son adecuados para obtener las componentes en wavelets como se menciona en [1].

Las formas típicas de la aproximación discreta para la primera y segunda derivada son mostradas a continuación:

$$\mathbf{D}^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots \\ -1 & 1 & 0 & 0 & \dots \\ 0 & -1 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -1 & 1 \end{bmatrix}$$

y:

$$\mathbf{D}^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ & & & \ddots & \ddots \\ & & & & 1 & -2 & 1 \end{bmatrix}$$

Para minimizar el funcional  $J(\mathbf{v})$ , obtenemos su gradiente e igualamos a cero, y una vez hecho esto despejamos el vector de señal estimada.

$$\begin{aligned} \frac{\partial J(\mathbf{v})}{\partial \mathbf{v}} &= \frac{\partial}{\partial \mathbf{v}} \left( (\mathbf{u} - \mathbf{S}\mathbf{v})^+ \mathbf{M}_U (\mathbf{u} - \mathbf{S}\mathbf{v}) + (\mathbf{v} - \mathbf{m}_v)^+ \mathbf{M}_v (\alpha) (\mathbf{v} - \mathbf{m}_v) \right) = \\ &= \frac{\partial}{\partial \mathbf{v}} \left( \mathbf{u}^+ \mathbf{M}_U \mathbf{u} - \mathbf{u}^+ \mathbf{M}_U \mathbf{S}\mathbf{v} - \mathbf{v}^+ \mathbf{S}^+ \mathbf{M}_U \mathbf{u} + \mathbf{v}^+ \mathbf{S}^+ \mathbf{M}_U \mathbf{S}\mathbf{v} + \mathbf{v}^+ \mathbf{M}_v (\alpha) \mathbf{v} - \mathbf{v}^+ \mathbf{M}_v (\alpha) \mathbf{m}_v - \mathbf{m}_v^+ \mathbf{M}_v (\alpha) \mathbf{v} + \mathbf{m}_v^+ \mathbf{M}_v (\alpha) \mathbf{m}_v \right) = \\ &= (2\mathbf{S}^+ \mathbf{M}_U (\mathbf{u} - \mathbf{S}\mathbf{v}) + 2\mathbf{M}_v (\alpha) (\mathbf{v} - \mathbf{m}_v)) = \mathbf{0} \end{aligned}$$

Así la señal estimada es:

$$\hat{\mathbf{v}} = \mathbf{m}_v + (\mathbf{S}^+ \mathbf{M}_U \mathbf{S} + \mathbf{M}_v (\alpha))^{-1} \mathbf{S}^+ \mathbf{M}_U (\mathbf{u} - \mathbf{S}\mathbf{m}_v) = \mathbf{m}_v + \mathbf{W}(\mathbf{u} - \mathbf{S}\mathbf{m}_v) \quad (3.16)$$

Donde:

$$\mathbf{W} = (\mathbf{S}^+ \mathbf{M}_U \mathbf{S} + \mathbf{M}_v (\alpha))^{-1} \mathbf{S}^+ \mathbf{M}_U \quad (3.17)$$

### 3.4 Evaluación del desempeño

En este apartado se evaluará el desempeño del algoritmo en un sentido estadístico, es decir, se obtendrá la media y la matriz de covarianza de la señal estimada, para eso hacemos el siguiente análisis:

$$\langle \hat{\mathbf{v}} \rangle = \langle \mathbf{m}_v + \mathbf{W}(\mathbf{u} - \mathbf{S}\mathbf{m}_v) \rangle = \mathbf{m}_v + \mathbf{W}\langle \mathbf{u} \rangle - \mathbf{W}\mathbf{S}\mathbf{m}_v = \mathbf{m}_v + \mathbf{W}\mathbf{S}\mathbf{m}_v - \mathbf{W}\mathbf{S}\mathbf{m}_v = \mathbf{m}_v \quad (3.18)$$

Haciendo un análisis de la covarianza de la señal estimada tenemos:

$$\mathbf{R}_{\hat{\mathbf{v}}} = \langle (\hat{\mathbf{v}} - \mathbf{m}_{\hat{\mathbf{v}}})(\hat{\mathbf{v}} - \mathbf{m}_{\hat{\mathbf{v}}})^T \rangle = \langle (\mathbf{W}(\mathbf{u} - \mathbf{m}_u))(\mathbf{W}(\mathbf{u} - \mathbf{m}_u))^T \rangle = \mathbf{W}\langle (\mathbf{u} - \mathbf{m}_u)(\mathbf{u} - \mathbf{m}_u)^T \rangle \mathbf{W}^T = \mathbf{W}\mathbf{R}_u \mathbf{W}^T \quad (3.19)$$

donde  $\mathbf{R}_u$  es la covarianza de la señal observada y fue especificada en apartados anteriores.

Se puede observar que la media de la señal estimada es igual a la media de la señal original así, podemos decir que el algoritmo entrega una señal estimada sin bias de la señal original.

### 3.5 Implementación computacional

El algoritmo obtenido anteriormente se puede implementar de manera computacional mediante una secuencia de etapas a desarrollar, las cuales se explicarán a continuación:

En una primera etapa se realiza la innovación de la información, es decir en esta etapa el valor a priori que tenemos del valor medio de la señal observada es sustraído de la señal observada, obteniendo así un vector innovado:

$$\mathbf{u}^e = \mathbf{u} - \mathbf{S}\mathbf{m}_v \quad (3.20)$$

En una segunda etapa se hace una estimación robusta de la señal estimada:

$$\mathbf{q} = \mathbf{S}^+ \mathbf{M}_T \mathbf{u}^e \quad (3.21)$$

En una tercera etapa se realiza la reconstrucción de la señal:

$$\hat{\mathbf{v}}^\circ = \mathbf{A}_\alpha^{-1} \mathbf{q} = (\mathbf{S}^+ \mathbf{M}_v \mathbf{S} + \mathbf{M}_v(\alpha))^{-1} \mathbf{q} \quad (3.22)$$

Y por último se realiza la restauración de la tendencia:

$$\hat{\mathbf{v}} = \mathbf{m}_v + \hat{\mathbf{v}}^\circ \quad (3.23)$$

Se debe de notar que si el vector de parámetros de regularización  $\alpha$  es igual a cero,  $\mathbf{M}_v$  es igual a cero y el estimador se reduce al caso generalizado de pseudo inversa de Moore-Penrose:

$$\hat{\mathbf{v}} = \mathbf{m}_v + (\mathbf{S}^+ \mathbf{M}_v \mathbf{S})^{-1} \mathbf{q} = \mathbf{m}_v + \mathbf{S}^\# \mathbf{u} \quad (3.24)$$

El esquema de esta implementación computacional es presentado a continuación:

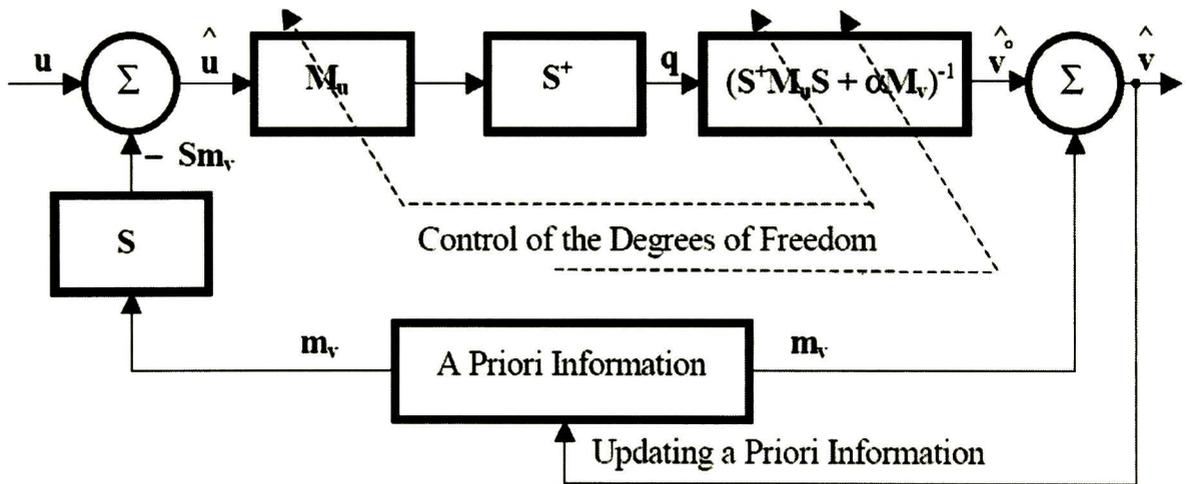


Figura 3.2: Esquema computacional del método WCLS

## 3.6 Conclusiones

En este capítulo se presentaron los algoritmos de regularización basados en la regularización de Tikhonov, se pudo observar que el mal planteamiento del problema nos conduce a malas reconstrucciones, esto es debido principalmente a dos causas, la primera es que la matriz de sistema o PSF es singular es decir, no tiene inversa y el segundo problema es que la señal observada esta contaminada con ruido, lo cual cuando se quiere reconstruir la señal el ruido introduce errores en la estimación.

Por eso es necesario añadir a la estrategia de solución un término de regularización, el cual nos va a solucionar estos problemas, este término de regularización es dependiente de parámetros los cuales podemos ajustar para obtener una mejor reconstrucción los cuales son conocidos como parámetros de regularización y son los grados de libertad del algoritmo, al ajustar estos parámetros de regularización en valores pequeños, la señal reconstruida va a presentar una gran presencia del ruido pero una gran resolución, en cambio si ajustamos estos parámetros en valores grandes la presencia del ruido disminuye pero la resolución también, por lo tanto la adecuada selección de estos parámetros es muy importante ya constituyen nuestro compromiso entre resolución y presencia de ruido en la señal reconstruida.

# Capítulo IV

## Método AWCLS

### 4.1 Planteamiento del Problema y Análisis de Información a Priori

En este capítulo se planteará el desarrollo del algoritmo implementado y la forma en que fue obtenido. Primero consideraremos el modelo matemático utilizado en este método, el cual es similar al modelo matemático utilizado en el método de Tikhonov pero sujeto a una ecuación de restricciones, así la ecuación de observación sería:

$$u = Sv + n \quad v \in V_N \quad u \in U_L \quad S: V_N \rightarrow U_L \quad (4.1)$$

la cual en forma vectorial sería expresada como:

$$\mathbf{u} = \mathbf{Sv} + \mathbf{n} \quad (4.2)$$

la cual esta restringida por la ecuación de restricciones:

$$Cv = b \quad v \in V_N \quad (4.3)$$

cuya forma en vector discreto es:

$$\mathbf{Cv} = \mathbf{b} \quad (4.4)$$

Donde  $\mathbf{C}$  es una matriz de valores reales, y la cual es un operador lineal que produce el mapeo,  $C: V_N \rightarrow CV_N$ , del espacio de señal original  $V_N$  al subespacio  $CV_N = V_P$ . El operador de restricciones

esta dado por la matriz de restricciones  $\mathbf{C}$  de dimensión  $P$  por  $N$ , donde  $N$  es la dimensión del vector de la señal  $\mathbf{v}$ , y  $P$  es la dimensión del vector de restricciones  $\mathbf{b}$ , siempre debe de cumplirse que  $P < N$ .

Ejemplos de la matriz y vectores de restricciones son:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 \end{bmatrix} \quad \text{con el vector de restricciones} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

La matriz anterior sirve como un proyector sobre el subespacio de los valores de frontera deseados en el vector de señal, donde  $v_1 = b_1$  es el valor de frontera en el primer dato de señal y  $v_N = b_2$  es el valor deseado para el último valor del vector de señal. Así, se puede plantear cualquier operador de restricción y establecer valores que suponemos están presentes en la señal que vamos a estimar, en otras palabras, imponemos restricciones de información que conocemos de manera a priori acerca de la señal, tales como valores de frontera, o valores en la misma señal.

Dada la ecuación de observación y la ecuación de restricciones tenemos que encontrar un operador solución  $W : U \rightarrow V$ , que opere sobre la señal observada  $\mathbf{u}$  el cual produzca un estimado óptimo:

$$\hat{\mathbf{v}} = W\mathbf{u} \tag{4.5}$$

siendo óptimo en el sentido de la teoría de optimización condicional bajo restricciones impuestas, lo cual hace uso de la teoría de optimización de Lagrange la cual se explicará a continuación.

## 4.2 Método de Lagrange

El método de Multiplicadores de Lagrange es utilizado ampliamente en problemas de optimización en donde la función a optimizar esta restringida por otra función, es decir dadas las funciones  $f : U \subset \mathbf{R}^n \rightarrow \mathbf{R}$  y  $g : U \subset \mathbf{R}^n \rightarrow \mathbf{R}$  con valores reales de clase  $C^1(U)$ , con  $\mathbf{x}_0 \in U$   $g(\mathbf{x}_0) = 0$  y  $S = \{\mathbf{x} \in \mathbf{R}^n : g(\mathbf{x}) = 0\}$ . Suponiendo que  $\nabla g(\mathbf{x}_0) \neq 0$ . Si  $f$  esta restringida a  $S$  tiene un máximo o mínimo relativo en  $\mathbf{x}_0 \in S$ , entonces existe un número real  $\lambda$  tal que:

$$\nabla f(\mathbf{x}_0) = \lambda \nabla g(\mathbf{x}_0) \quad (4.6)$$

El método de multiplicadores de Lagrange nos permite encontrar los puntos  $\mathbf{x} \in \mathbf{R}^n$  que optimizan (encontrar máximos o mínimos) una función dada  $f(\mathbf{x})$ , sujeta a la restricción  $g(\mathbf{x}) = 0$ .

En el caso general cuando se tienen  $k$  restricciones,  $g_1(\mathbf{x}) = 0, g_2(\mathbf{x}) = 0, \dots, g_k(\mathbf{x}) = 0$  donde  $\mathbf{x} = \{x_i, i = 1, \dots, n\}$  el método usa la fórmula:

$$\nabla f(\mathbf{x}) = \lambda_1 \nabla g_1(\mathbf{x}) + \lambda_2 \nabla g_2(\mathbf{x}) + \dots + \lambda_k \nabla g_k(\mathbf{x}) \quad (4.7)$$

donde  $\lambda = \{\lambda_i, i = 1, \dots, k\}$  son los multiplicadores de Lagrange. Así para resolver un problema de optimización sujeto a  $k$  restricciones construimos el funcional Lagrangiano:

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \sum_{i=1}^k \lambda_i g_i(\mathbf{x}) = f(\mathbf{x}) - [\lambda, \mathbf{g}(\mathbf{x})] \quad (4.8)$$

el cual debemos minimizar respecto a  $\mathbf{x}$  y  $\lambda = \{\lambda_i, i = 1, \dots, k\}$

$$\min_{\mathbf{x}, \lambda} \{L(\mathbf{x}, \lambda)\}$$

es decir encontrar los gradientes:

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial x_j} = 0 \quad \text{para } j = 1, \dots, n$$

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial \lambda_i} = 0 \quad \text{para } i = 1, \dots, k$$

y resolver el sistema de ecuaciones  $n \times k$  anterior para  $\mathbf{x}$  y  $\lambda = \{\lambda_i, i = 1, \dots, k\}$

### 4.3 Desarrollo Matemático

Para llevar a cabo el desarrollo del algoritmo implementado primero definimos el criterio a optimizar, tal criterio es la función objetivo  $J(\mathbf{v})$ , tal función esta definida como:

$$J(\mathbf{v}) = J_1(\mathbf{v}) + J_2(\mathbf{v}, \alpha) \quad (4.9)$$

donde:

$$J_1(\mathbf{v}) = d^2_{\mathbf{M}_U}(\mathbf{u}, \mathbf{Sv}) = \|\mathbf{u} - \mathbf{Sv}\|_U^2 = (\mathbf{u} - \mathbf{Sv})^T \mathbf{M}_U (\mathbf{u} - \mathbf{Sv})$$

y

$$J_2(\mathbf{v}) = \|\mathbf{v} - \mathbf{m}_v\|_v^2 = (\mathbf{v} - \mathbf{m}_v)^T \mathbf{M}_v(\alpha) (\mathbf{v} - \mathbf{m}_v)$$

Donde  $\mathbf{M}_v(\alpha)$  es definida como en el algoritmo de Tikhonov y depende de los parámetros de regularización  $\alpha$ .

El primer criterio es la distancia entre el vector de datos medidos y la componente de señal  $\mathbf{Sv}$  en el espacio de señal  $U$  impuesto por la matriz  $\mathbf{M}_U$ , el segundo criterio es conocido como criterio de normalización y es la distancia entre el vector de datos  $\mathbf{v}$  y su vector de valores medios.

El funcional anterior esta restringido por la ecuación de restricciones:

$$\mathbf{Cv} = \mathbf{b} \quad (4.10)$$

Así el planteamiento del problema esta dado por:

$$\min_{\mathbf{v}} J(\mathbf{v}) \quad \text{sujeto a } \mathbf{Cv} = \mathbf{b} \quad (4.11)$$

Para poder realizar lo anterior construimos un funcional de Lagrange para volver el problema con restricciones a un problema sin restricciones, este funcional esta dado por:

$$L(\mathbf{v}, \boldsymbol{\lambda}) = \frac{1}{2} J(\mathbf{v}) + [\boldsymbol{\lambda}, (\mathbf{C}\mathbf{v} - \mathbf{b})] \quad (4.12)$$

donde  $\boldsymbol{\lambda}$  es el vector de multiplicadores de Lagrange. Así el problema de optimización se convierte en:

$$\min_{\mathbf{v}, \boldsymbol{\lambda}} L(\mathbf{v}, \boldsymbol{\lambda}) \quad (4.13)$$

Así la solución deseada esta dada por:

$$\hat{\mathbf{x}} = (\hat{\mathbf{v}}^T, \hat{\boldsymbol{\lambda}}^T) = \min_{\mathbf{v}, \boldsymbol{\lambda}} \{L(\mathbf{v}, \boldsymbol{\lambda})\} = \min_{\mathbf{v}, \boldsymbol{\lambda}} \left\{ \frac{1}{2} J(\mathbf{v}) + [\boldsymbol{\lambda}, (\mathbf{C}\mathbf{v} - \mathbf{b})] \right\} \quad (4.14)$$

Para lograr lo anterior obtenemos el gradiente del funcional de Lagrange con respecto al vector de datos  $\mathbf{v}$  y respecto al vector  $\boldsymbol{\lambda}$  de multiplicadores de Lagrange e igualamos estos a  $\mathbf{0}$ .

$$\mathbf{g}_{\mathbf{v}} = \frac{\partial L(\mathbf{v}, \boldsymbol{\lambda})}{\partial \mathbf{v}} = \mathbf{0}$$

$$\mathbf{g}_{\boldsymbol{\lambda}} = \frac{\partial L(\mathbf{v}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathbf{0}$$

Utilizando reglas de diferenciación de funcionales respecto a vectores tenemos que estos gradientes son:

$$\mathbf{g}_{\mathbf{v}} = \frac{\partial L(\mathbf{v}, \boldsymbol{\lambda})}{\partial \mathbf{v}} = -\mathbf{S}^T \mathbf{M}_{\mathbf{u}} (\mathbf{u} - \mathbf{S}\mathbf{v}) + \mathbf{M}_{\mathbf{v}} (\boldsymbol{\alpha})(\mathbf{v} - \mathbf{m}_{\mathbf{v}}) + \mathbf{C}^T \boldsymbol{\lambda} = \mathbf{0}$$

$$\mathbf{g}_{\boldsymbol{\lambda}} = \frac{\partial L(\mathbf{v}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathbf{C}\mathbf{v} - \mathbf{b} = \mathbf{0} \quad (4.15)$$

Reordenando la ecuación 4.15 obtenemos:

$$(\mathbf{S}^T \mathbf{M}_U \mathbf{S} + \mathbf{M}_V(\boldsymbol{\alpha})) \mathbf{v} = \mathbf{S}^T \mathbf{M}_U \mathbf{u} + \mathbf{M}_V(\boldsymbol{\alpha}) \mathbf{m}_v - \mathbf{C}^T \boldsymbol{\lambda} \quad (4.16)$$

Sumando y restando el término  $\mathbf{S}^T \mathbf{M}_U \mathbf{S} \mathbf{m}_v$  obtenemos la expresión:

$$\begin{aligned} (\mathbf{S}^T \mathbf{M}_U \mathbf{S} + \mathbf{M}_V(\boldsymbol{\alpha})) \mathbf{v} &= \mathbf{S}^T \mathbf{M}_U \mathbf{u} - \mathbf{S}^T \mathbf{M}_U \mathbf{S} \mathbf{m}_v + \mathbf{S}^T \mathbf{M}_U \mathbf{S} \mathbf{m}_v + \mathbf{M}_V(\boldsymbol{\alpha}) \mathbf{m}_v - \mathbf{C}^T \boldsymbol{\lambda} = \\ &= (\mathbf{S}^T \mathbf{M}_U \mathbf{S} + \mathbf{M}_V(\boldsymbol{\alpha})) \mathbf{m}_v + \mathbf{S}^T \mathbf{M}_U (\mathbf{u} - \mathbf{S} \mathbf{m}_v) - \mathbf{C}^T \boldsymbol{\lambda} \end{aligned} \quad (4.17)$$

Despejando  $\mathbf{v}$  de la ecuación anterior obtenemos:

$$\hat{\mathbf{v}} = \mathbf{m}_v + (\mathbf{S}^T \mathbf{M}_U \mathbf{S} + \mathbf{M}_V(\boldsymbol{\alpha}))^{-1} \mathbf{S}^T \mathbf{M}_U (\mathbf{u} - \mathbf{S} \mathbf{m}_v) - (\mathbf{S}^T \mathbf{M}_U \mathbf{S} + \mathbf{M}_V(\boldsymbol{\alpha}))^{-1} \mathbf{C}^T \boldsymbol{\lambda} \quad (4.18)$$

Si definimos los siguientes términos:

$$\begin{aligned} \mathbf{K}_\alpha &= \mathbf{A}_\alpha^{-1} = (\mathbf{S}^T \mathbf{M}_U \mathbf{S} + \mathbf{M}_V(\boldsymbol{\alpha}))^{-1} \\ \mathbf{q} &= \mathbf{S}^T \mathbf{M}_U (\mathbf{u} - \mathbf{S} \mathbf{m}_v) \end{aligned} \quad (4.19)$$

La ecuación 4.18 queda como:

$$\hat{\mathbf{v}} = \mathbf{m}_v + \mathbf{K}_\alpha \mathbf{q} - \mathbf{K}_\alpha \mathbf{C}^T \boldsymbol{\lambda} \quad (4.20)$$

Se puede observar de la ecuación 4.20 que la señal estimada es dependiente del vector de multiplicadores de Lagrange para evitar esto sustituimos  $\hat{\mathbf{v}}$  en la ecuación 4.10 y despejamos el vector de multiplicadores  $\boldsymbol{\lambda}$

$$\mathbf{b} - \mathbf{C} \mathbf{v} = \mathbf{b} - \mathbf{C} (\mathbf{m}_v + \mathbf{K}_\alpha \mathbf{q} - \mathbf{K}_\alpha \mathbf{C}^T \boldsymbol{\lambda}) = \mathbf{0} \quad (4.20)$$

$$\hat{\boldsymbol{\lambda}} = (\mathbf{C} \mathbf{K}_\alpha \mathbf{C}^T)^{-1} (\mathbf{C} \mathbf{K}_\alpha \mathbf{q} - \mathbf{b}) + (\mathbf{C} \mathbf{K}_\alpha \mathbf{C}^T)^{-1} \mathbf{C} \mathbf{m}_v = \mathbf{D}_\alpha \mathbf{P}_{\alpha b} + \mathbf{D}_\alpha \mathbf{C} \mathbf{m}_v \quad (4.21)$$

$$\begin{aligned} \mathbf{D}_\alpha &= (\mathbf{C} \mathbf{K}_\alpha \mathbf{C}^T)^{-1} \\ \mathbf{P}_{\alpha b} &= (\mathbf{C} \mathbf{K}_\alpha \mathbf{q} - \mathbf{b}) \end{aligned} \quad (4.22)$$

Sustituyendo el vector de multiplicadores de Lagrange estimado en la ecuación 4.20 obtenemos:

$$\begin{aligned}\hat{\mathbf{v}} &= \mathbf{m}_v + \mathbf{K}_\alpha \mathbf{q} - \mathbf{K}_\alpha \mathbf{C}^T \mathbf{D}_\alpha \mathbf{P}_{\alpha b} - \mathbf{K}_\alpha \mathbf{C}^T \mathbf{D}_\alpha \mathbf{C} \mathbf{m}_v \\ \hat{\mathbf{v}} &= \mathbf{K}_\alpha \mathbf{q} - \mathbf{K}_\alpha \mathbf{C}^T \mathbf{D}_\alpha \mathbf{P}_{\alpha b}\end{aligned}\quad (4.23)$$

Esta es la expresión de la señal estimada la cual ya no depende del vector de multiplicadores de Lagrange.

## 4.4 Evaluación del desempeño

Para obtener la evaluación del desempeño obtenemos las estadísticas de la señal estimada las cuales están dadas por:

$$\langle \hat{\mathbf{v}} \rangle = \langle \mathbf{K}_\alpha \mathbf{q} - \mathbf{K}_\alpha \mathbf{C}^T \mathbf{D}_\alpha \mathbf{P}_{\alpha b} \rangle = \mathbf{K}_\alpha \langle \mathbf{q} \rangle - \mathbf{K}_\alpha \mathbf{C}^T \mathbf{D}_\alpha \langle \mathbf{P}_{\alpha b} \rangle = \mathbf{K}_\alpha \mathbf{S}^T \mathbf{M}_U (\langle \mathbf{u} \rangle - \mathbf{S} \langle \mathbf{m}_v \rangle) - \mathbf{K}_\alpha \mathbf{C}^T \mathbf{D}_\alpha (\mathbf{C} \mathbf{K}_\alpha \mathbf{S}^T \mathbf{M}_U (\langle \mathbf{u} \rangle - \mathbf{S} \langle \mathbf{m}_v \rangle) - \langle \mathbf{b} \rangle) \quad (4.24)$$

siendo:

$$\langle \mathbf{u} \rangle = \mathbf{S} \mathbf{m}_v \quad (4.25)$$

El valor medio de la señal estimada esta dado por:

$$\langle \hat{\mathbf{v}} \rangle = \mathbf{K}_\alpha \mathbf{C}^T \mathbf{D}_\alpha \mathbf{b} \quad (4.26)$$

La varianza de la señal estimada esta dada por:

$$\mathbf{R}_{\hat{\mathbf{v}}} = \langle (\hat{\mathbf{v}} - \mathbf{m}_{\hat{\mathbf{v}}}) (\hat{\mathbf{v}} - \mathbf{m}_{\hat{\mathbf{v}}})^T \rangle \quad (4.27)$$

donde el termino  $\hat{\mathbf{v}} - \mathbf{m}_{\hat{\mathbf{v}}}$  esta dado por:

$$\hat{\mathbf{v}} - \mathbf{m}_{\hat{\mathbf{v}}} = \mathbf{K}_\alpha \mathbf{q} - \mathbf{K}_\alpha \mathbf{C}^T \mathbf{D}_\alpha \mathbf{C} \mathbf{K}_\alpha \mathbf{q} + \mathbf{K}_\alpha \mathbf{C}^T \mathbf{D}_\alpha \mathbf{b} - \mathbf{K}_\alpha \mathbf{C}^T \mathbf{D}_\alpha \mathbf{b} = \mathbf{K}_\alpha \mathbf{q} - \mathbf{K}_\alpha \mathbf{q} = \mathbf{0} \quad (4.28)$$

Así la varianza de la señal estimada es:

$$\mathbf{R}_{\hat{\mathbf{v}}} = \langle (\mathbf{0})(\mathbf{0})^T \rangle = \mathbf{0} \quad (4.29)$$

## 4.5 Implementación Computacional

El esquema computacional utilizado para implementar el algoritmo desarrollado es el que se muestra a continuación:

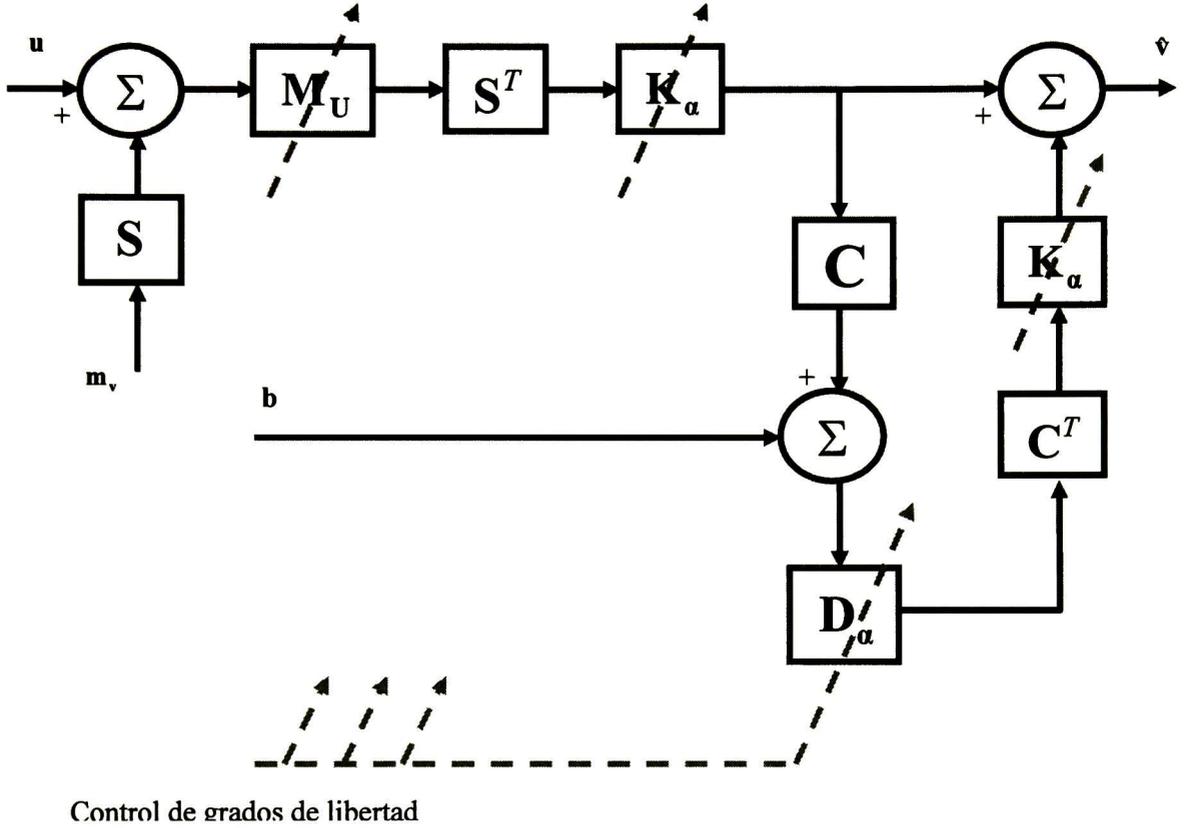


Figura 4.1: Esquema computacional del metodo WCLS con restricciones lineales.

En la primera etapa se lleva acabo una estimación robusta:

$$\mathbf{q} = \mathbf{S}^T \mathbf{M}_U (\mathbf{u} - \mathbf{S} \mathbf{m}_v) \quad (4.30)$$

En la segunda etapa del algoritmo se lleva acabo la reconstrucción de la señal sin imponer aun las restricciones:

$$\hat{\mathbf{v}}_{unc} = \mathbf{K}_a \mathbf{q} = (\mathbf{S}^T \mathbf{M}_U \mathbf{S} + \mathbf{M}_v)^{-1} \mathbf{q} \quad (4.31)$$

En la ultima etapa se lleva acabo la corrección de la señal restaurada sujeta a las restricciones impuestas:

$$\hat{\mathbf{v}} = \hat{\mathbf{v}}_{unc} - \mathbf{K}_a \mathbf{C}^T \mathbf{D}_a \mathbf{P}_{ab}$$

## 4.6 Implementación para el Procesamiento de Imágenes

Todos los algoritmos presentados hasta el momento son para estimar señales, en las cuales la señal observada está en forma de vector, es decir, es de dimensión  $n \times 1$ , para poder trabajar con imágenes de dimensión  $n \times m$  muchos autores proponen ordenar la imagen en forma lexicográfica [1],[3]; lo cual consiste en concatenar los  $m$  vectores renglón de los que se compone la imagen uno tras otro y tratar la imagen como un vector de dimensión  $(m \times n) \times 1$ .

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1m} \\ a_{21} & a_{22} & & & \vdots \\ a_{31} & & \cdot & & \vdots \\ \vdots & & & \cdot & \vdots \\ a_{n1} & \dots & \dots & \dots & a_{nm} \end{bmatrix} \Rightarrow \begin{bmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{1m} \\ a_{21} \\ a_{22} \\ \vdots \\ \vdots \\ a_{nm} \end{bmatrix}$$

La matriz de sistema se vuelve una matriz de dimensión  $(m \times n) \times (m \times n)$ , si suponemos que la PSF solo influye en el eje  $y$ , la matriz de sistema es una matriz diagonal a bloques donde cada bloque es la PSF que afecta el correspondiente renglón:

$$\mathbf{S}_{lex} = \begin{bmatrix} \mathbf{S} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{S} & & & \vdots \\ \mathbf{0} & & \mathbf{S} & & \vdots \\ \vdots & & & \cdot & \vdots \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{S} \end{bmatrix}$$

donde  $\mathbf{S}$  es una matriz de dimensión  $n \times n$  que difumina el  $k$  renglón de la imagen original, la matriz  $\mathbf{0}$  es de dimensión  $n \times n$  y esta compuesta de ceros,  $\mathbf{0} = [0_{ij}]$ .

Si la PSF afectara en ambos ejes de la imagen, la matriz  $\mathbf{S}_{lex}$  dejaría de ser diagonal a bloques y las matrices  $\mathbf{0}$  se convertirían en matrices que describirían la interacción entre los diferentes pixeles.

El problema que tenemos al trabajar con el ordenamiento lexicográfico es que trabajamos con matrices de sistema demasiado grandes (dimensión  $(mxn) \times (mxn)$ ), lo cual implica una gran cantidad de operaciones para llevar a cabo la multiplicación de esta matriz con el vector ordenado lexicográficamente, además de un gran gasto computacional para calcular su inversa, además de que ocupa una gran cantidad de memoria, como ejemplo una imagen de tamaño  $256 \times 256$  píxeles ocupa una matriz de sistema de tamaño  $65536 \times 65536$  lo cual implica tener memoria disponible en la computadora mayor a 4294967296 localidades, lo cual puede ser en ocasiones prohibitivo.

Si suponemos que la PSF solo afecta en el eje  $y$ , es decir nuestra matriz de sistema  $S_{lex}$  es diagonal a bloques, entonces podemos llevar a cabo el procesamiento de la imagen tomando renglón por renglón y reconstruirlo individualmente cada uno, la matriz de sistema que se utilizaría sería  $S$ . Haciendo estas consideraciones se reduce el tamaño de la matriz de sistema con la que se trabajaría, suponiendo el mismo ejemplo de una imagen de  $256 \times 256$  píxeles la matriz de sistema  $S$  a utilizar sería de dimensión  $256 \times 256$  lo cual reduce también la cantidad de operaciones que se llevarán a cabo.

Esta suposición de que la PSF solo afecta en el eje  $y$ , es valida porque muchas PSF reales son separables, es decir se puede expresar como una multiplicación de funciones la parte que afecta el eje  $y$  por la parte que afecta el eje  $x$ .

De aquí en adelante los algoritmos que se presentarán supondrán que la PSF solo afecta el eje  $y$ , pero se puede generalizar para PSF que afecten ambos ejes, realizando primero la reconstrucción en un sentido y luego llevar a cabo la reconstrucción en el otro sentido.



Figura 4.2: Estructura de reconstrucción en ambos ejes de una imagen.

## 4.7 Conclusiones

En este capítulo se presentó la modificación realizada al algoritmo WCLS al imponerle restricciones lineales, estas restricciones lineales pueden ser definidas como valores de frontera de la señal o imagen a reconstruir, o como valores predefinidos dentro de la imagen, estas restricciones deben ser conocidas y se imponen como información a priori de la señal a reconstruir, el resultado de imponer estas restricciones en el algoritmo de reconstrucción es el mejoramiento de la señal reconstruida en comparación con el algoritmo WCLS, además se observó al evaluar su desempeño que de manera teórica la varianza de la señal estimada es cero, lo cual indica que en forma estadística la señal reconstruida va a ser igual a la señal original.

También se comentó la manera en la cual se pueden procesar imágenes, ya que la teoría presentada hasta este punto implicaba que tratábamos con señales, se explicó que existe que la imagen puede ordenarse de manera lexicográfica y tratarse como una señal, pero esto implicaba realizar una mayor cantidad de operaciones y además de que se necesita una mayor cantidad de memoria dentro de la computadora en la cual se esté llevando acabo el procesamiento, así se propuso la manera de tratar cada renglón de la imagen como una señal y realizar la reconstrucción de cada renglón por separado, esto es posible si suponemos que la PSF solo afecta en un solo sentido de la imagen; al realizar este proceso se necesita de menos operaciones y menor cantidad de memoria utilizada.

# Capítulo V

## Curva L

### 5.1 Introducción al método de la Curva L

Como se describió en los capítulos anteriores, las reconstrucciones obtenidas con los algoritmos presentados dependen del ajuste de parámetros de regularización los cuales constituyen los grados de libertad de los algoritmos. En específico el algoritmo que se planteo, los parámetros de regularización vienen implícitos en la matriz  $\mathbf{M}_v(\alpha)$ , recordando que esta matriz es la suma ponderada de varias matrices las cuales pueden ser por ejemplo la aproximación numérica de la primera y segunda derivada, otros autores proponen otro tipos de matrices [1].

La selección de estos parámetros de regularización no es una tarea fácil y continua siendo un problema abierto, existiendo diferentes métodos para obtenerlos tales como el método de discrepancia, método generalizado de validación cruzada (GCV) [18], método del límite mínimo, el método de la curva L [16], [17], [19], e incluso un método intuitivo el cual es similar a ajustar la lente de una cámara para obtener la imagen enfocada correctamente la cual se basa en la experiencia del usuario.

Muchos de estos métodos han sido desarrollados para encontrar un solo parámetro de regularización y su implementación para un número mayor de parámetros de regularización no es sencilla. Los requerimientos que buscamos al seleccionar el método a utilizar son de que no sea dependiente en ningún sentido de la norma de la señal original,  $\|\mathbf{v}\|$ , o la varianza del ruido  $N_0$ , esto lo conseguimos con los métodos GCV y el método de la curva L.

El método GCV, tiene muchas propiedades deseables, la solución estimada converge a la solución real como la norma del error tiende a cero o el número de muestras del vector de datos tiende a infinito.

Este método encuentra los mejores parámetros al minimizar la función GCV la cual es:

$$G(\lambda) = \frac{\|\mathbf{u} - \mathbf{S}\hat{\mathbf{v}}(\lambda)\|_2^2}{\text{traza}(\mathbf{I} - \mathbf{S}\mathbf{W}(\lambda))} \quad (5.1)$$

donde  $\mathbf{W}$  es la matriz que mapea los datos observados al vector estimado i.e.  $\hat{\mathbf{v}} = \mathbf{W}(\lambda)\mathbf{u}$ .

La desventaja de este método es que el cálculo del denominador de la función GCV es muy complejo lo cual hace que el método tenga mucho gasto computacional. Por esta razón se escogió el método de la curva L cuya generalización para más de un parámetro de regularización es la hipersuperficie L.

Este método ha sido ampliamente estudiado en [1], [16], [17], [19] y se han desarrollado métodos más eficientes a partir del método de la curva L para encontrar los parámetros de regularización óptimos.

Hasta el momento se conocen pocas propiedades teóricas del método de la curva L, pero su simplicidad ha conducido a que se haya estudiado ampliamente en una variedad de diferentes aplicaciones, sin embargo se ha encontrado que falla bajo ciertas circunstancias como se explica en [20], donde se muestra que si la varianza del ruido tiende a cero el algoritmo tiende a producir un parámetro de regularización que es lo suficientemente pequeño pero suaviza la reconstrucción obtenida; otras consideraciones son presentadas en [1].

Todas estas condiciones en las cuales tiende a fallar el algoritmo son muy específicas y no son muy probables de que se presenten en un problema práctico.

## 5.2 Planteamiento Matemático de la Curva L

Primero describiremos en que consiste el método de la curva L y luego se explicará cómo extrapolarlo para más de un parámetro de regularización lo cual genera una hipersuperficie.

La curva L es una gráfica de la distancia cuadrada de la señal observada con la señal reconstruida afectada por la matriz de sistema, contra la norma cuadrada de la señal estimada en el espacio inducido por la matriz de regularización  $\mathbf{M}_v$  para diferentes valores del parámetro de regularización.

Matemáticamente esto se expresa de la siguiente manera:

Sea la gráfica de la curva L en la cual cada punto de esta gráfica esta definido por el par coordinado:

$$L(\alpha) = (\Phi(z(\alpha)), \Phi(x(\alpha))) \quad (5.2)$$

donde:

$$x(\alpha) = \|\hat{v}(\alpha)\|_v^2 = \|\mathbf{R}\hat{v}(\alpha)\|^2 = \frac{\hat{v}(\alpha)^T \mathbf{M}_v \hat{v}(\alpha)}{\alpha} \quad (5.3)$$

$$z(\alpha) = \|\mathbf{u} - \mathbf{S}\hat{v}(\alpha)\|_2^2 \quad (5.4)$$

Recordando que la matriz  $\mathbf{M}_v$  para un solo parámetro de regularización esta conformada por:

$$\mathbf{M}_v = \alpha \mathbf{R}^T \mathbf{R} \quad (5.5)$$

y  $\hat{v}(\alpha)$  es la señal reconstruida al proponer el parámetro de regularización; por último la función  $\Phi(\cdot)$  se aplica a las normas para poder observar mejor el comportamiento de la grafica para un amplio rango de valores de regularización, esta función puede ser cualquier logaritmo, pero en el caso específico de esta tesis se trabajará con el logaritmo base 10.

Una vez encontrada la grafica de la curva L, se debe encontrar la función de curvatura gaussiana de la curva L la cual para una curva en dos dimensiones esta definida por:

$$k(\alpha) = \frac{\tilde{z}' \tilde{x}'' - \tilde{x}' \tilde{z}''}{\left( (\tilde{x}')^2 + (\tilde{z}')^2 \right)^{3/2}} \quad (5.6)$$

donde  $k$  es la curvatura respecto al parámetro de regularización,  $(\cdot)'$  indica que es diferenciación respecto al parámetro de regularización  $\alpha$  y  $(\cdot)_{10}$  indica logaritmo base 10.

Una vez encontrada la función de curvatura, el punto en el que sea máximo positivo esta función es el que corresponde al parámetro de regularización óptimo. Así los pasos a seguir para poder encontrar el parámetro de regularización óptimo son los siguientes:

- 1.- Llevar a cabo la reconstrucción de la señal con diferentes valores de  $\alpha$ .
- 2.- Calcular los correspondientes valores de  $x(\alpha)$  y  $z(\alpha)$ .
- 3.- Construir la curva L.
- 4.- Calcular su curvatura gaussiana al aplicar la fórmula 5.6
- 5.- Encontrar el máximo de la curvatura.
- 6.- Observar a que parámetro de regularización corresponde el máximo.

A continuación se presentara un ejemplo del uso del método de la curva L para la obtención del parámetro óptimo:

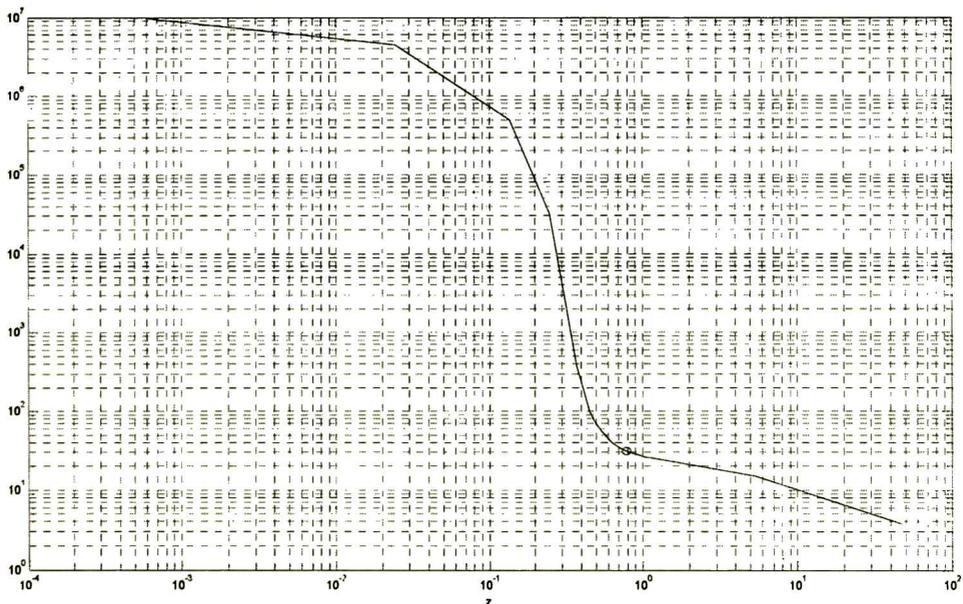


Figura 5.1: Curva L típica para un parámetro de regularización, el círculo rojo corresponde al punto con mayor curvatura.

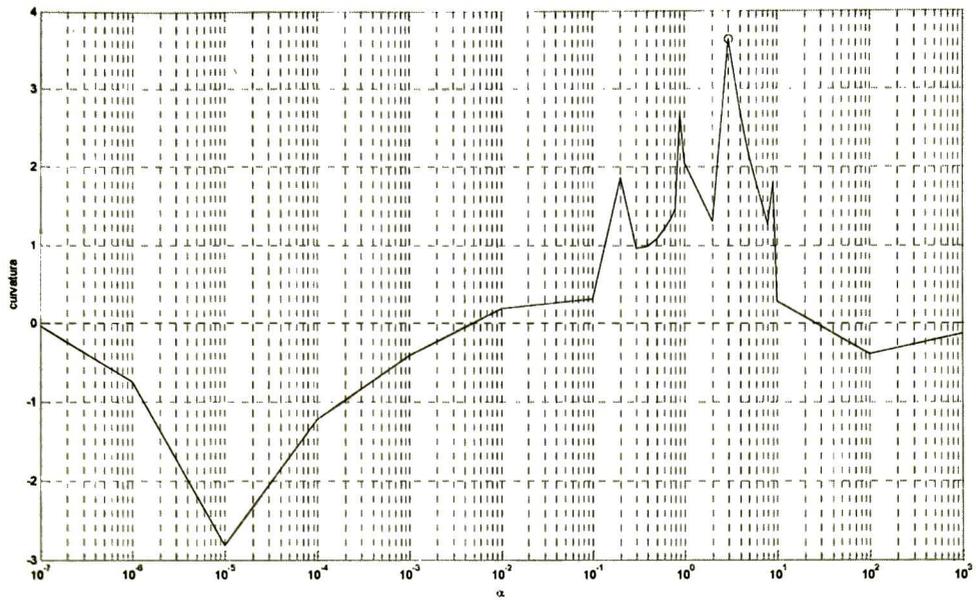


Figura 5.2: Curvatura Gaussiana de la curva L de la figura 5.1

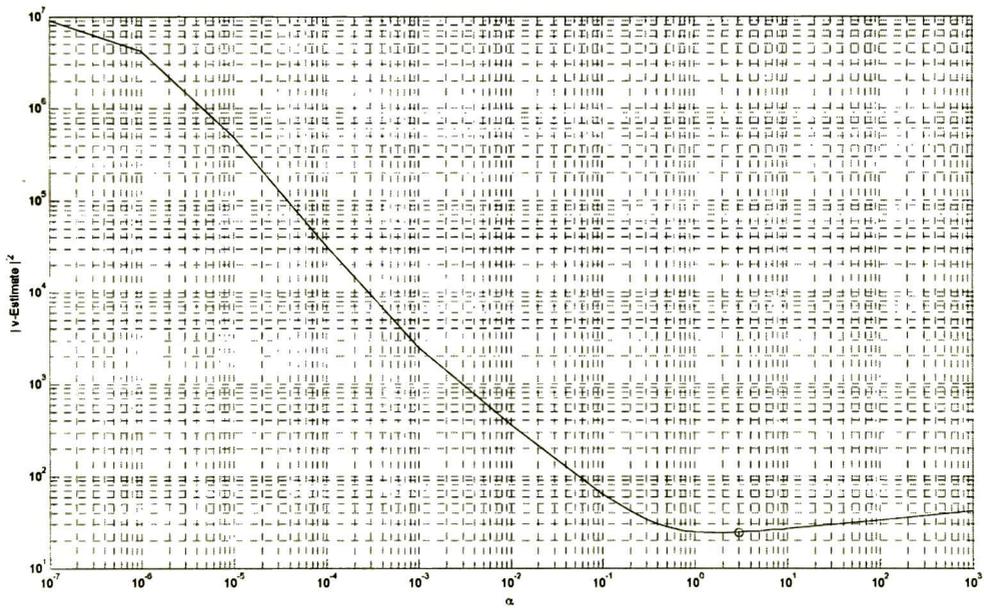


Figura 5.3: Gráfica de la distancia cuadrática entre la señal original y la reconstruida, el círculo rojo corresponde al valor con mayor curvatura de la gráfica 5.1

El ejemplo anterior se realizó para una señal observada de dimensión  $n \times 1$ , la cual está contaminada con ruido gaussiano con varianza de .01 y un PSF exponencial con un ancho de 7 muestras, el método de reconstrucción que fue el de Tikhonov en donde la matriz  $\mathbf{M}_v$  es igual a la matriz identidad. En las gráficas el círculo rojo señala la posición donde se alcanza el valor máximo en la gráfica de curvatura.

Se puede observar que esta posición concuerda con el punto de mayor curvatura en la curva L que es conocido como esquina y además es muy cercano al lugar donde se obtiene el error mínimo en la gráfica de error, esto significa que este procedimiento nos da una muy buena estimación del parámetro de regularización óptimo.

La extensión de la curva L a más de un parámetro de regularización es conocida como hipersuperficie L, en el que cada punto de esta superficie se plantea matemáticamente como:

$$L(\boldsymbol{\alpha}) = (\Phi(z(\boldsymbol{\alpha})), \Phi(x_1(\boldsymbol{\alpha})), \dots, \Phi(x_n(\boldsymbol{\alpha}))) \quad (5.7)$$

para  $N$  parámetros de regularización, donde:

$$x_n(\boldsymbol{\alpha}) = \|\mathbf{R}_n \hat{\mathbf{v}}(\boldsymbol{\alpha})\|^2 = \hat{\mathbf{v}}(\boldsymbol{\alpha})^T \mathbf{R}_n^T \mathbf{R}_n \hat{\mathbf{v}}(\boldsymbol{\alpha}) \quad (5.8)$$

$$z(\boldsymbol{\alpha}) = \|\mathbf{u} - \mathbf{S}\hat{\mathbf{v}}(\boldsymbol{\alpha})\|_2^2 \quad (5.9)$$

Siendo la matriz de regularización definida como:

$$\mathbf{M}_v = \alpha_1 \mathbf{R}_1^T \mathbf{R}_1 + \dots + \alpha_N \mathbf{R}_N^T \mathbf{R}_N \quad (5.10)$$

La curvatura gaussiana para este tipo de superficies con  $N$  parámetros de regularización se obtiene con la fórmula:

$$k(\boldsymbol{\alpha}) = \frac{(-1)^N}{\omega^{N+1}} |\mathbf{P}| \quad (5.11)$$

donde:

$$\omega^2 = 1 + \sum_{i=1}^N \left( \frac{\partial \bar{z}}{\partial \bar{x}_i} \right)^2$$

$$P_{i,j} = \frac{\partial^2 \bar{z}}{\partial \bar{x}_i \partial \bar{x}_j}$$

$$\bar{z} = \log(z)$$

$$\bar{x}_i = \log(x_i)$$

Una explicación más amplia al respecto de la curvatura gaussiana para hipersuperficies es encontrada en [1], donde también se explica detalladamente cómo poder encontrar esta función y el gasto computacional que se tiene al calcularla, cabe mencionar que el cálculo de los gradientes en la ecuación 5.11 es una tarea que requiere mucho gasto computacional por lo cual varios autores propusieron varios métodos iterativos para encontrar más rápido el punto con mayor curvatura en la hipersuperficie sin la necesidad de realizar tantas operaciones, a continuación se presentará una implementación iterativa para poder llevar a cabo esta tarea.

### 5.3 Solución numérica el método de la Curva L

Como se mencionó antes, el cálculo del gradiente es una tarea que requiere muchos cálculos además de que se tienen que calcular muchos puntos de la curva L para poder obtener un resultado aceptable, por lo cual varios autores propusieron métodos para hacer más eficiente el método de la curva L uno de esos métodos es el que se propone en [1] donde en vez de maximizar la función de curvatura se tratará de minimizar un función de distancia desde un punto conocido, otro método es el propuesto en [20] el cual se basa en computar lo que llaman listón L el cual contiene la curva L en su interior. El listón L puede ser computado sin mucho gasto computacional con la bi-diagonalización parcial de Lancos de la matriz de sistema PSF.

Otros métodos consideran calcular la curvatura de la curva L por medio de ajustar una spline cúbica suavizada a los puntos de la curva L.

El método con el se trabajó es el que se mencionó al principio y en el cual el autor propuso una función de distancia la cual debe de ser minimizada. Ésta es la distancia desde un punto conocido a

cualquier punto de la curva L y es mínima esta distancia a la esquina de la curva L, tal y como se muestra en la figura 5.4.

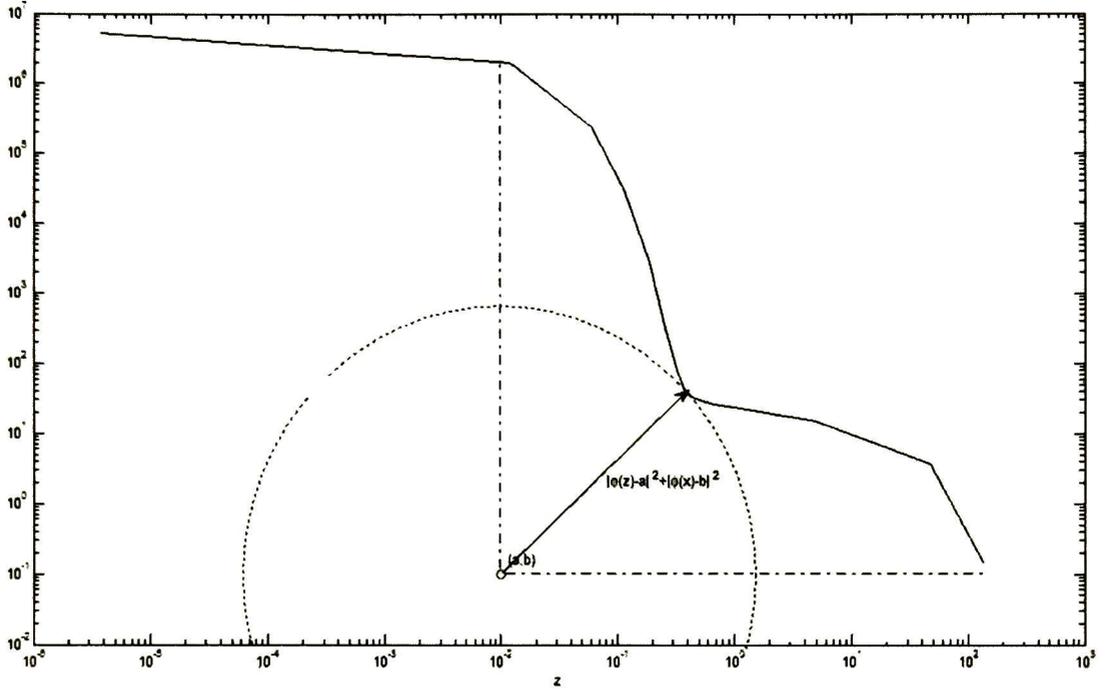


Figura 5.4: Gráfica de la función de distancia entre el punto (a,b) y cualquier punto de la curva L.

El punto conocido es dado como información a priori al algoritmo y éste se encuentra matemáticamente a partir de los valores singulares de la matriz de sistema, tal y como se menciona en [1], las coordenadas de este punto concuerdan con los valores en que la curva empieza a hacerse horizontal es decir  $z$  es constante, y la otra coordenada cuando la curva se vuelve vertical es decir, cuando  $x$  se vuelve constante, para poder encontrar este punto matemáticamente es necesario obtener los valores singulares máximo y mínimo de la matriz de sistema, una vez obtenidos las coordenadas de este punto son:

$$o = (a, b) \tag{5.12}$$

$$a = \log_{10} \left( \left\| \mathbf{u} - \mathbf{S}\hat{\mathbf{v}}(\alpha_a) \right\|_2^2 \right) \tag{5.13}$$

$$b = \log_{10} \left( \left\| \mathbf{R}\hat{\mathbf{v}}(\alpha_b) \right\|_2^2 \right) \tag{5.14}$$

donde:

$$\alpha_a = \sigma_{\min}^2$$

$$\alpha_b = \sigma_{\max}^2$$

y  $\sigma_{\min}$   $\sigma_{\max}$  son los valores singulares mínimo y máximo de la matriz de sistema respectivamente.

Para más de un parámetro de regularización este punto esta dado por las coordenadas:

$$o = (a, b_1, b_2, \dots, b_N)$$

$$a = \log_{10} \left( \|\mathbf{u} - \mathbf{S}\hat{\mathbf{v}}(\alpha_a)\|_2^2 \right)$$

$$b_n = \log_{10} \left( \|\mathbf{R}_n \hat{\mathbf{v}}(\alpha_{b_n})\|^2 \right) \text{ para } 1 < n < N$$

$$\alpha_a = (\sigma_{\min}^2, \sigma_{\min}^2, \dots, \sigma_{\min}^2)$$

$$\alpha_{b_1} = (\sigma_{\max}^2, 0, \dots, 0)$$

$$\alpha_{b_2} = (0, \sigma_{\max}^2, 0, \dots, 0)$$

$$\alpha_{b_N} = (0, 0, \dots, \sigma_{\max}^2)$$

Las características de este punto son detalladas en [1] aquí mencionaremos algunas de estas características.

La primera de estas características es que este punto, que es dado como información a priori, se encuentra debajo de la curva en una región en donde el punto más cercano a la curva L, es la esquina la cual es la que buscamos o bien un punto muy cercano a este punto, por lo tanto la distancia mínima es la que esta dada desde este punto a un punto muy cercano a la esquina de la curva L.

Al saber ésto se puede construir una nueva función para poder encontrar la esquina de la curva L, esta función es la distancia desde el punto dado como información a priori y la curva L y nuestro objetivo será minimizar esta distancia para poder encontrar la esquina de la curva L y por consiguiente los parámetros de regularización óptimos.

La distancia cuadrada entre el punto dado a priori y cualquier punto de la curva esta dada por:

$$d(\alpha) = \|\log_{10}(z(\alpha)) - a\|^2 + \sum_{i=1}^N \|\log_{10}(x_i(\alpha) - b_i)\|^2 \quad (5.15)$$

Para poder minimizar la ecuación anterior podemos obtener el gradiente respecto al vector de parámetros de regularización  $\alpha$  e igualar a cero y una vez obtenido esto resolver de manera numérica ya que se trata de un sistema de ecuaciones no lineales tal y como se explica en [1], o la otra manera es minimizar esta función directamente por un método numérico como el de gradiente descendente, gradiente conjugado o cualquier otro.

En el desarrollo de esta tesis se tomará esta opción y se explicará como se puede llevar a cabo la minimización de la función de distancia por medio del método de gradiente descendente y una implementación con redes neuronales.

## 5.4 Esquema iterativo para la solución de la curva L

A continuación se dará una breve explicación al método de gradiente descendente como método de minimización de funciones de manera numérica.

Un problema de minimización es encontrar los valores extremos de esta función, es decir los valores donde la derivada de esta función es cero. Para poder llevar a cabo esto existen varios métodos tales como:

método de Nelder-Mead

método de gradiente descendente

método Newton-Raphson

El método de Nelder-Mead como se menciona en [6] y [23], es un método simple para encontrar mínimos locales de una función de varias variables. Para funciones de dos variables, se construye un triángulo, y el método es un patrón de búsqueda que compara los valores de la función en los tres

vértices del triángulo. El peor vértice, donde  $f(x,y)$  es mayor, es rechazado y reemplazado con un nuevo vértice, un nuevo triángulo es formado y la búsqueda continúa.

El proceso genera una secuencia de triángulos, para los cuales los valores de la función en cada vértice es cada vez más pequeño. El tamaño de los triángulos se reduce cada vez y las coordenadas del punto mínimo son encontradas.

El método del gradiente descendente, hace uso del cálculo del gradiente de la función para poder encontrar el mínimo de ésta. Sea una función  $f(\mathbf{x})$  de  $N$  variables donde  $\mathbf{x} = (x_1, x_2, \dots, x_N)$  el gradiente esta definido como:

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = (f_1, f_2, \dots, f_N) \quad (5.16)$$

donde las derivadas parciales  $f_k = \frac{\partial f(\mathbf{x})}{\partial x_k}$  son evaluadas en  $\mathbf{x}$ , el vector gradiente apunta localmente en

la dirección del mayor incremento de  $f(\mathbf{x})$ , por lo tanto  $-\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$  apunta localmente en dirección del

mayor decremento. Una aproximación numérica de las derivadas parciales de las cuales se compone el vector gradiente es evaluando la función en  $\mathbf{x}$  y en un pequeño incremento del elemento con respecto al cual se esta diferenciando es decir:

$$f_k = \frac{\partial f(\mathbf{x})}{\partial x_k} \approx \frac{f(\mathbf{x} + \Delta \mathbf{x}_k) - f(\mathbf{x})}{\Delta x} \quad (5.17)$$

donde:

$$\Delta \mathbf{x}_k = (x_1, x_2, \dots, x_k + \Delta x, \dots, x_N)$$

Si este incremento es lo suficientemente pequeño entonces la aproximación numérica será casi igual al valor real, en el desarrollo de esta tesis se utilizo un incremento de  $10^{-8}$

Los pasos que se deben llevar a cabo para la implementación del método de gradiente descendente son:

- 1.- Evaluar el vector gradiente en el punto  $\mathbf{p}_k$   $\mathbf{g} = \frac{\partial f(\mathbf{p}_k)}{\partial \mathbf{x}}$
- 2.- Computar el vector de dirección  $\mathbf{s} = -\frac{\mathbf{g}}{\|\mathbf{g}\|}$
- 3.- Realizar la minimización del parámetro  $t$  en la función  $\phi(t) = f(\mathbf{p}_k + t\mathbf{s})$  en el intervalo  $[0, b]$  donde  $b$  es grande. Esto producirá un valor  $t = h_{\min}$  que es el mínimo local para  $\phi(t)$ . La relación  $\phi(h_{\min}) = f(\mathbf{p}_k + h_{\min}\mathbf{s})$  muestra que es un mínimo local para  $f(\mathbf{x})$  a través de la línea de búsqueda  $\mathbf{x} = \mathbf{p}_k + t\mathbf{s}$ .
- 4.- Construir el siguiente punto  $\mathbf{p}_{k+1} = \mathbf{p}_k + h_{\min}\mathbf{s}$ .
- 5.- Evaluar la regla de paro del algoritmo, la cual es, si se cumple que los valores de  $f(\mathbf{p}_k)$  y  $f(\mathbf{p}_{k+1})$  son suficientemente cercanos y  $\|\mathbf{p}_{k+1} - \mathbf{p}_k\| \leq \varepsilon$ .
- 6.- Repetir los pasos anteriores si no se cumple el punto 5

El algoritmo anterior produce una secuencia de puntos  $\{\mathbf{p}_k\}$  los cuales tienen la propiedad de que  $f(\mathbf{p}_0) > f(\mathbf{p}_1) > \dots > f(\mathbf{p}_k) > \dots$ . Si  $\lim_{k \rightarrow \infty} \mathbf{p}_k = \mathbf{p}$ , entonces  $f(\mathbf{p})$  ser un mínimo local de  $f(\mathbf{x})$ . Este método tiene la ventaja de que es de implementación sencilla y no se deben realizar tantas operaciones para poder obtener el resultado deseado, la desventaja es de que puede tardar un poco en converger además de que para poder obtener un mejor resultado se debe inicializar el algoritmo en un punto  $\mathbf{p}_k$  que sabemos es cercano al resultado deseado.

El método Newton-Raphson se basa en el cálculo del vector gradiente y la inversa de la matriz jacobiana la cual por definición es construida con las segundas derivadas respecto a cada variable de la función que se desea minimizar.

Sea la matriz jacobiana definida como:

$$[\mathbf{J}]_{,m} = \frac{\partial f(\mathbf{x})}{\partial x_n \partial x_m} \quad (5.18)$$

Una vez definida la matriz jacobiana el valor siguiente se calcula mediante:

$$\mathbf{p}_{k+1} = -\mathbf{J}(\mathbf{p}_k)^{-1} \mathbf{g}(\mathbf{p}_k) \quad (5.19)$$

teniendo como regla de paro la misma que en el algoritmo anterior.

Este método tiene la ventaja de que converge más rápido que el método de gradiente descendente ya que hace uso de la matriz jacobiana, pero con la desventaja de que se deben realizar más operaciones para poder calcular las segundas derivadas lo cual lo hace menos conveniente para el cálculo que deseamos realizar ya que aumenta considerablemente el número de operaciones a realizar al aumentar el número de parámetros de regularización.

## 5.5 Implementación con redes neuronales

Una red neuronal es un conjunto de elementos simples interconectados cuya funcionalidad esta basada en la de una neurona biológica, la forma más simple es referida a la unidad de umbral lógico propuesta por McCulloch y Pitts. Una neurona biológica esta compuesta por unidades de entrada llamadas synapses a las cuales llegan las señales bioeléctricas provenientes de otras neuronas o terminales nerviosas, estas señales son afectadas por un peso, es decir, son amplificadas o atenuadas, este efecto es el conocido como efecto post sinóptico; luego todas estas señales son sumadas lo cual se conoce como activación y esta activación es comparada con un umbral la cual es asociada con un umbral post sináptico si excede este umbral genera una señal conocida como potencial de acción el cual viaja a la salida de la neurona conocida como axión.

Una red neuronal es un procesador paralelo masivo, que tiene una propensidad natural para guardar conocimiento experimental y hacerlo disponible para su uso. Esto reensambla el cerebro en dos aspectos:

El conocimiento es adquirido por la red a través de un proceso de aprendizaje.

Las conexiones interneuronales mantienen el conocimiento como pesos sinápticos

El proceso de aprendizaje tiene la función de modificar los pesos sinápticos de tal manera que se cumpla el objetivo deseado.

De manera matemática el comportamiento de una red neuronal se puede plantear de la siguiente manera:

$$y_m = \sum_{k=1}^K w_{mk} x_k \tag{5.20}$$

$$z_m = \varphi(y_m - \theta_m)$$

donde:

$x_k$  para  $k=1, \dots, K$  son las entradas de la neurona  $m$

$w_{mk}$  para  $k=1, \dots, K$  son los pesos que se aplican a las entradas (pesos sinápticos)

$y_m$  es la salida de la suma de las entradas ponderadas

$\theta_m$  es el umbral

$\varphi(\cdot)$  es la función de activación

$z_m$  es la salida de la neurona

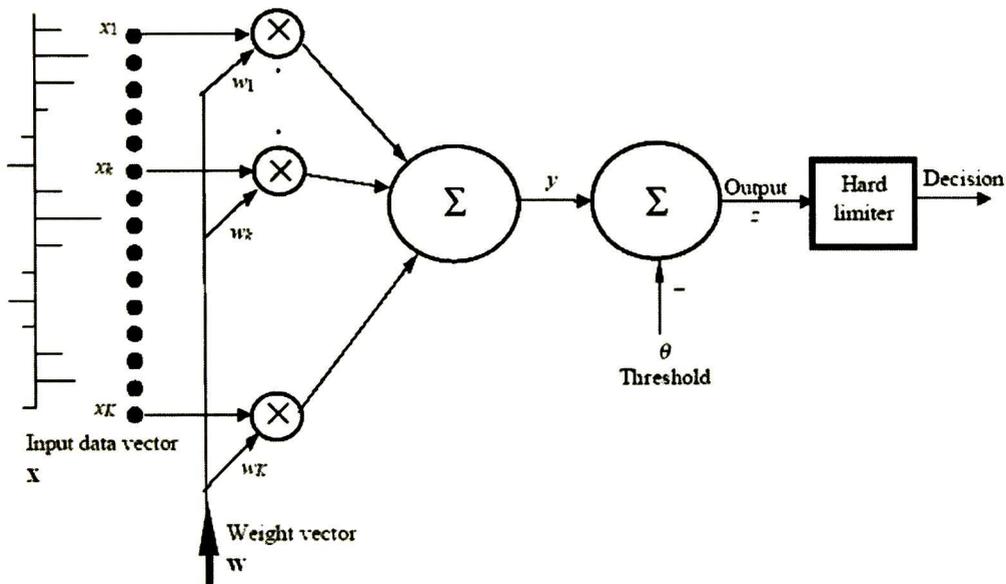


Figura 5.5: Esquema de una neurona.

de las ecuaciones anteriores se pueden observar cosas importantes, la primera es que la operación de la suma de las señales ponderadas corresponde a un producto interno además la resta de la suma de las señales ponderadas con el umbral puede ser incluida dentro de este producto interno si hacemos los siguientes reacomodos:

si

$$Y_m = y_m - \theta_m \quad (5.21)$$

entonces:

$$Y_m = \sum_{k=0}^K W_{mk} X_k \quad (5.22)$$

$$z_m = \varphi(Y_m) \quad (5.23)$$

donde el vector de dimensión  $k+1$  de entrada es  $\mathbf{X} = \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix}$

y el vector  $k+1$  de pesos es  $\mathbf{W}_m = \begin{pmatrix} -\theta_m \\ \mathbf{w}_m \end{pmatrix}$

Respecto a las funciones de activación existen varias, las cuales se utilizan dependiendo de la clase de clasificación que queremos realizar, la primera es la de tipo de umbral la cual queda expresada matemáticamente como:

$$\varphi(y) = \begin{cases} 1 & \text{si } y \geq 0 \\ 0 & \text{si } y < 0 \end{cases} \quad (5.24)$$

Alternativamente esta función se puede expresar para que exista una simetría con respecto al origen de tal manera que:

$$\varphi(y) = \begin{cases} 1 & \text{si } y \geq 0 \\ -1 & \text{si } y < 0 \end{cases} \quad (5.25)$$

Otra función utilizada es la sigmoid cuya expresión matemática es:

$$\varphi(v) = \frac{1}{1 + \exp(-\alpha v)} \quad (5.26)$$

la cual es una función continua y cuyo parámetro  $\alpha$  controla la pendiente de esta función, su contradominio es  $[0,1]$  al igual que la función de umbrales puede buscar que ésta sea simétrica, esto se logra aproximando la función sigmoid a un tangente hiperbólico como se muestra a continuación:

$$\varphi(v) = \tanh(v/2) = \frac{1 - \exp(-v)}{1 + \exp(-v)} \quad (5.27)$$

Una de las posibles aplicaciones de las redes neuronales recurrentes, es la optimización de funciones, lo cual es lo que se busca para poder encontrar los parámetros de regularización óptimos, esto presenta la ventaja de que sea mas efectivo en aplicaciones de tiempo real, además de que es posible implementarlos de manera física en circuitos electrónicos de aplicación específico en donde el problema de optimización es tratado de manera paralela y distribuida.

La primera propuesta que se hizo para resolver problemas de optimización fue hecha por Tank-Hopfield en 1986, aunque esta propuesta tiene la deficiencia de que el punto de equilibrio puede no ser una solución exacta al problema original, ha inspirado a varios investigadores para desarrollar algoritmos de optimización. Unos de estos algoritmos son propuesto en [21], [22], donde se pretende

optimizar funciones no lineales convexas sujetas a restricciones no lineales, lo cual se ajusta a lo que buscamos, ya que la función de distancia que deseamos minimizar es no lineal y convexa, el planteamiento matemático propuesto en este algoritmo es el siguiente:

$$\text{Minimizar } f(\mathbf{x}) \text{ sujeta a } \mathbf{c}(\mathbf{x}) \leq \mathbf{0}, \mathbf{x} \geq \mathbf{0} \quad (5.28)$$

donde  $f(\mathbf{x})$  puede ser no lineal pero debe ser convexa,  $\mathbf{x} \in R^n$   $f: R^n \rightarrow R$  y  $\mathbf{c}(\mathbf{x}) = [c_1(\mathbf{x}), \dots, c_m(\mathbf{x})]^T$

Si se propone el funcional de Lagrange:

$$L(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + \mathbf{y}^T \mathbf{c}(\mathbf{x}) \quad (5.29)$$

La solución propuesta es:

$$\begin{cases} (\mathbf{x} - \alpha(\nabla f(\mathbf{x}) + \nabla \mathbf{c}(\mathbf{x})\mathbf{y}))^+ - \mathbf{x} = \mathbf{0} \\ (\mathbf{y} + \alpha \mathbf{c}(\mathbf{x}))^+ - \mathbf{y} = \mathbf{0} \end{cases} \quad (5.30)$$

donde:

$\nabla f(\mathbf{x})$  es el gradiente de  $f(\mathbf{x})$ ,

$$\nabla \mathbf{c}(\mathbf{x}) = (\nabla c_1(\mathbf{x}), \dots, \nabla c_m(\mathbf{x}))$$

$$(\mathbf{x})^+ = ([x_1]^+, \dots, [x_n]^+)^T$$

$$(\mathbf{y})^+ = ([y_1]^+, \dots, [y_n]^+)^T$$

$$[x_i]^+ = \max\{0, x_i\}, [y_i]^+ = \max\{0, y_i\}$$

y  $\alpha \geq 0$  es constante

La implementación como una red neuronal es la siguiente:

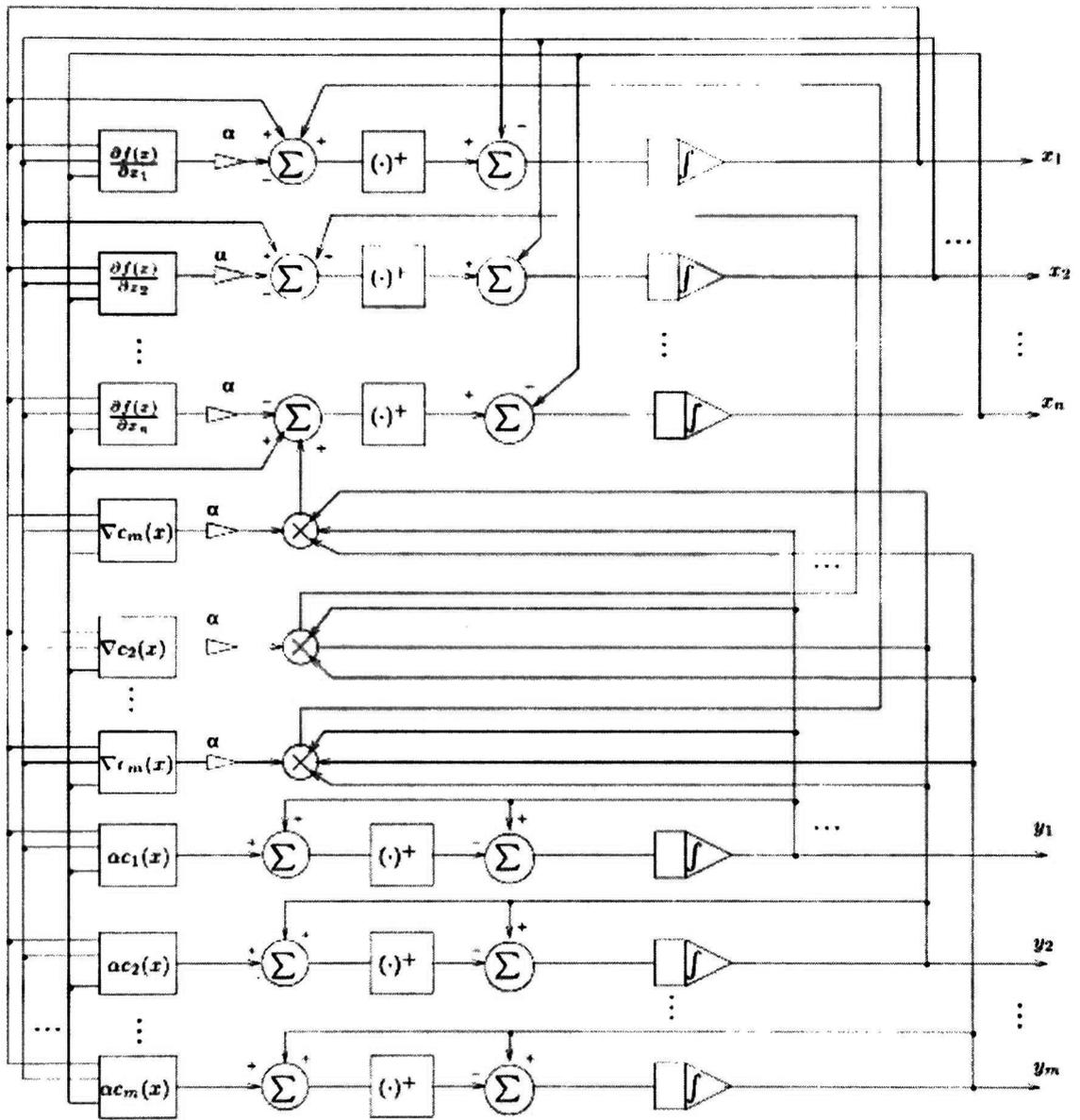


Figura 5.6: Red neuronal propuesta por Wang para resolver un problema de optimización no lineal con restricciones.

cuya ecuación dinámica es:

$$\frac{d}{dt} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{cases} (\mathbf{x} - \alpha(\nabla f(\mathbf{x}) + \nabla c(\mathbf{x})\mathbf{y}))^+ - \mathbf{x} \\ (\mathbf{y} + \alpha c(\mathbf{x}))^+ - \mathbf{y} \end{cases} \quad (5.31)$$

Para realizar la implementación discreta de la red neuronal anterior nos basamos en su ecuación dinámica y la convertimos en una ecuación de diferencias tal y como se muestra en la ecuación 5.32.

$$\mathbf{x}(k+1) - \mathbf{x}(k) = (\mathbf{x}(k) - \alpha(\nabla f(\mathbf{x}(k)) + \nabla \mathbf{c}(\mathbf{x}(k))\mathbf{y}(k)))^+ - \mathbf{x}(k)$$

$$\mathbf{y}(k+1) - \mathbf{y}(k) = (\mathbf{y}(k) + \alpha \mathbf{c}(\mathbf{x}(k)))^+ - \mathbf{y}(k)$$

$$\mathbf{x}(k+1) = (\mathbf{x}(k) - \alpha(\nabla f(\mathbf{x}(k)) + \nabla \mathbf{c}(\mathbf{x}(k))\mathbf{y}(k)))^+$$

$$\mathbf{y}(k+1) = (\mathbf{y}(k) + \alpha \mathbf{c}(\mathbf{x}(k)))^+$$

(5.32)

La ecuación 5.32 describe el comportamiento de la red neuronal en tiempo discreto y puede ser implementada como se muestra en el diagrama 5.7:

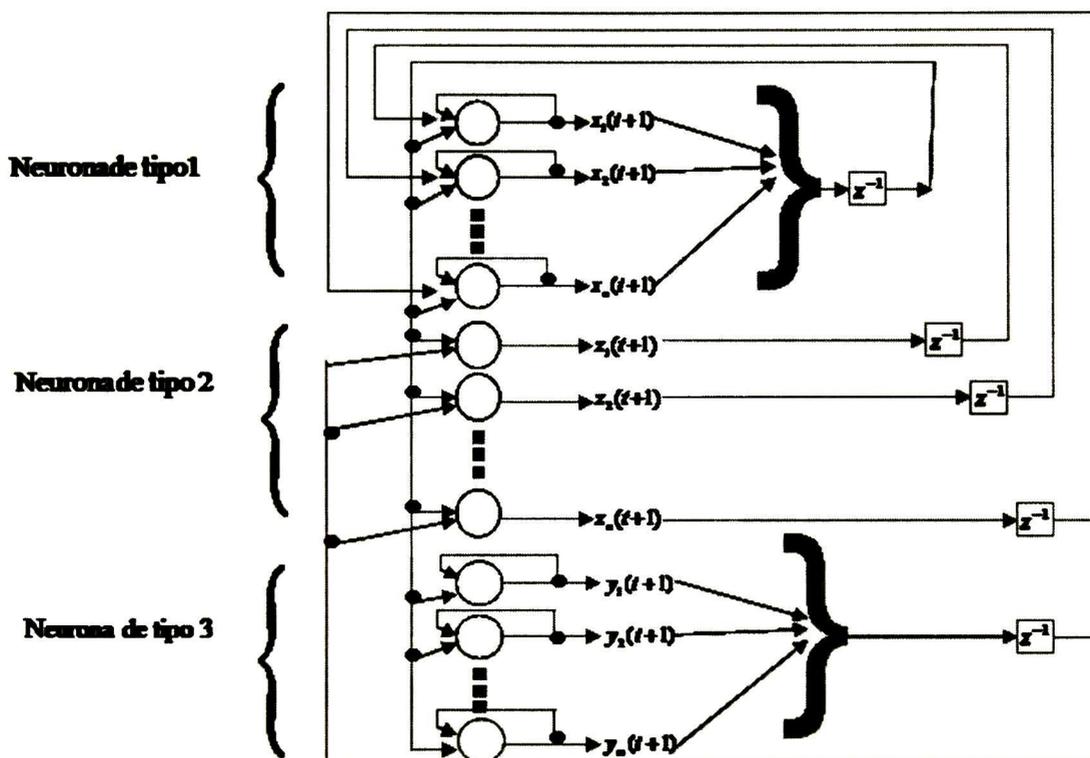
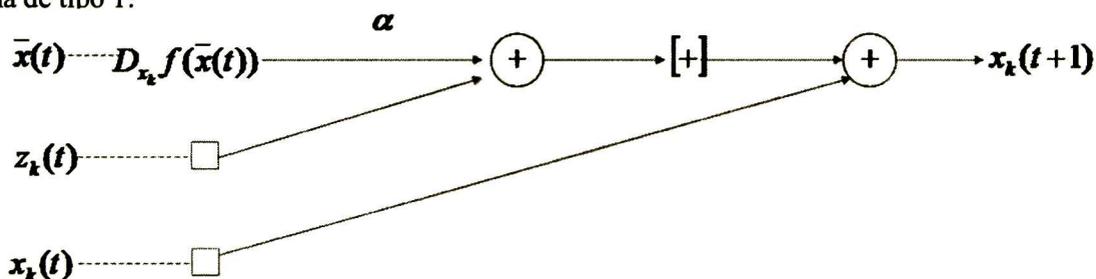


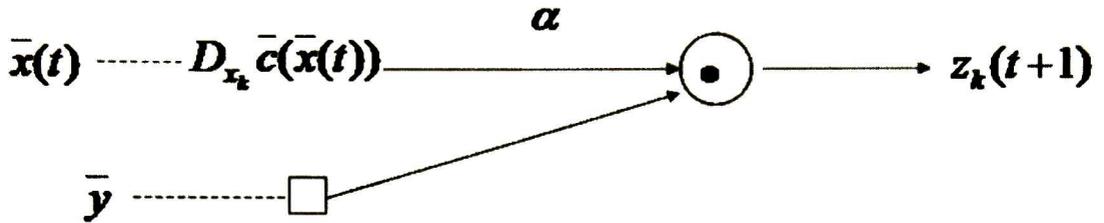
Figura 5.7: Red neuronal implementada para minimizar la función de distancia.

Donde

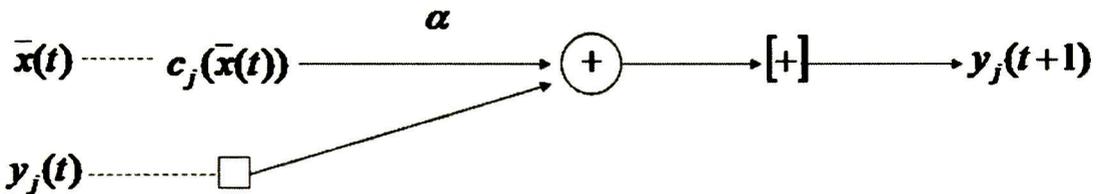
Neurona de tipo 1:



Neurona de tipo 2:



Neurona de tipo 3:



## 5.6 Conclusiones

En este capítulo se explicó un algoritmo para encontrar los parámetros de regularización que proporcionan la mejor reconstrucción, este método se basa en graficar la distancia cuadrada de la señal reconstruida afectada por la matriz del sistema y la señal observada, contra la norma cuadrada de la señal estimada, para diferentes parámetros de regularización; una vez realizado este proceso el objetivo es encontrar el punto con mayor curvatura el cual corresponde a los parámetros óptimos.

Este algoritmo conocido como curva L, presenta varias ventajas tales como simplicidad y de que es posible implementarse para encontrar más de un parámetro de regularización, una desventaja es de que se deben realizar muchas reconstrucciones para poder encontrar una cantidad suficiente de puntos de la curva L y así poder estimar el punto con mayor curvatura. Para eso se mostró una manera para poder encontrar este punto con mayor curvatura con una cantidad menor de operaciones, la cual consiste en

proponer una función de distancia entre un punto conocido y cualquier punto de la curva; y tratar de minimizar esta función de distancia, lo cual puede ser realizado de manera iterativa.

Los métodos propuestos para llevar a cabo esta minimización fueron el método de gradiente descendente el cual es ampliamente conocido y un método con redes neuronales, el cual presenta la ventaja de que puede ser implementado con operaciones muy sencillas y además de que su diseño permite que sea implementado de forma paralela lo cual reduce el tiempo de ejecución de este.

# Resultados

Las siguientes simulaciones fueron realizadas con señales de dimensión de 256 píxeles, fueron contaminadas con diferentes niveles de ruido (5dB, 15dB, 30dB) y con las PSF exponencial, Sinc y rectangular con diferentes anchos ( $\Delta = 25, 45, 65$  píxeles), fueron reconstruidas con el método AWCLS y con el método WCLS, el desempeño de estos algoritmos fue evaluado con la fórmula de IOSNR la cual esta dada por:

$$IOSNR = 10 \log_{10} \left( \frac{\sum_{k=1}^K (u_k - v_k)^2}{\sum_{k=1}^K (\hat{v}_k - v_k)^2} \right) = 10 \log_{10} \left( \frac{[(\mathbf{u} - \mathbf{v}), (\mathbf{u} - \mathbf{v})]}{[(\hat{\mathbf{v}} - \mathbf{v}), (\hat{\mathbf{v}} - \mathbf{v})]} \right)$$

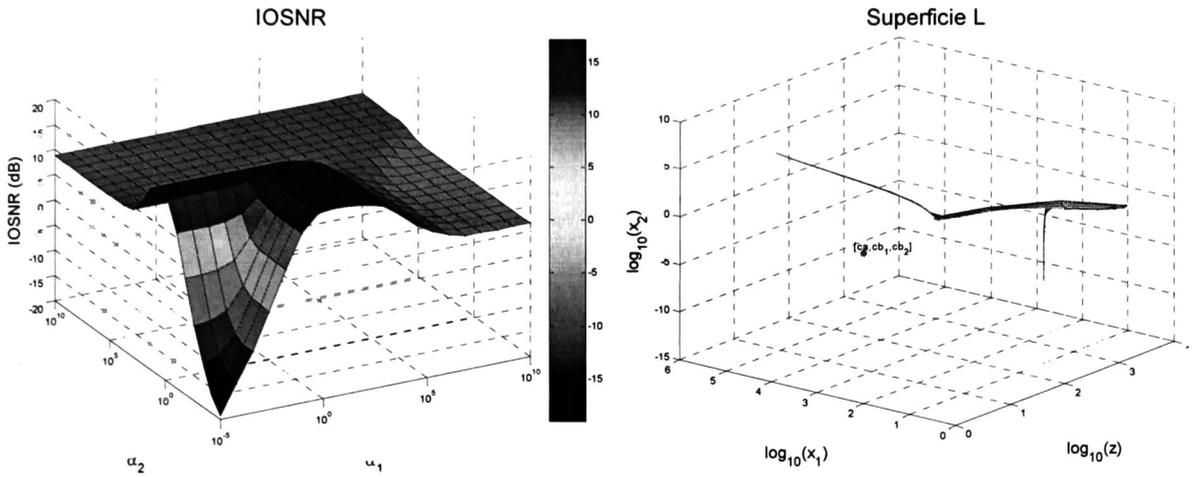
Donde:

$[\mathbf{a}, \mathbf{b}]$  indica producto entre  $\mathbf{a}$  y  $\mathbf{b}$ .

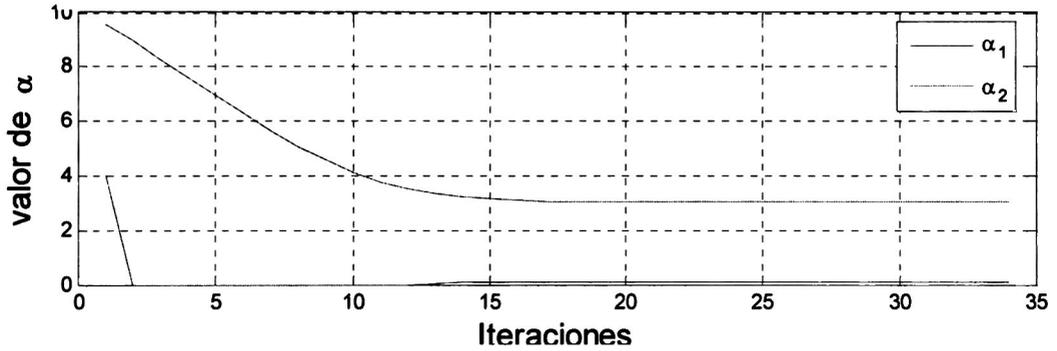
También se muestra la grafica de IOSNR para diferentes valores y el valor obtenido por la red neuronal, ademas de el punto encontrado en la hipersuperficie L. Por último se muestran graficas del comportamiento de los parámetros de regularización encontrados con la red neuronal y con el método de gradiente descendente.

El último grupo de simulaciones se hacen con imágenes de 256x256 píxeles y se reconstruyen utilizando el método AWCLS y WCLS con la implementación con red neuronal para encontrar los parámetros de regularización.

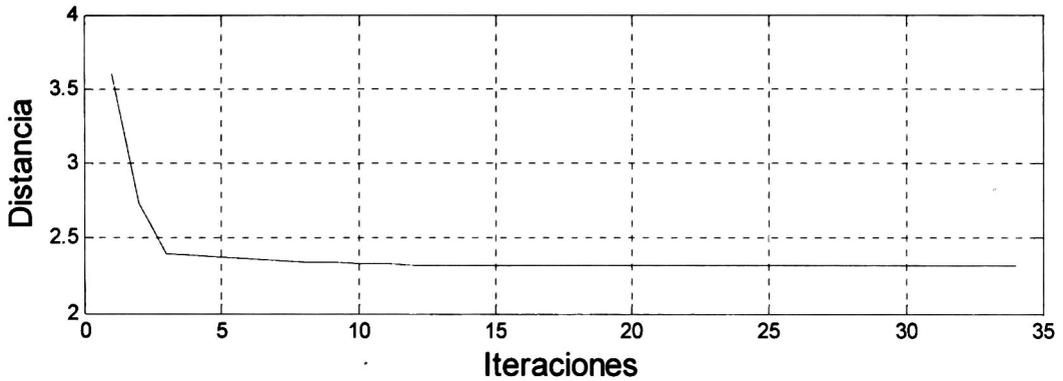
Simulación hecha con función exponencial SNR=5dB y  $\Delta=25$ .



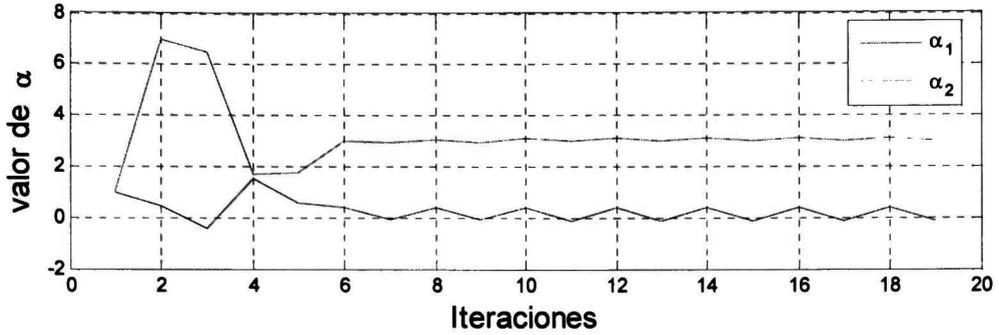
### Convergencia de los parámetros de regularización Red neuronal



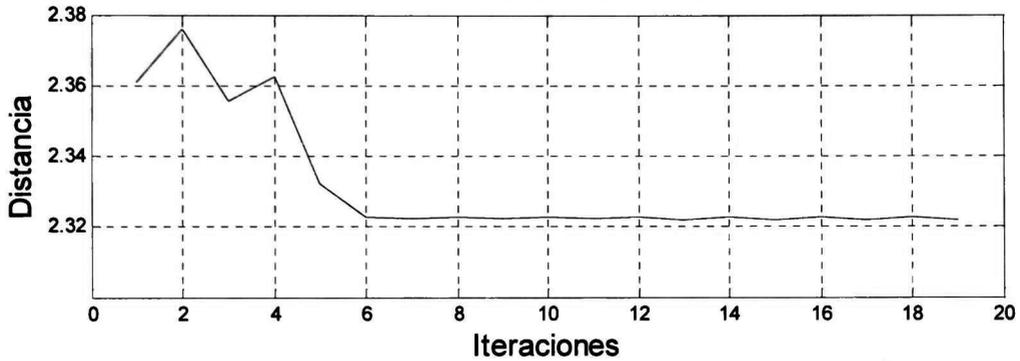
### Función de distancia



## Convergencia de los parámetros de regularización Gradiente descendente



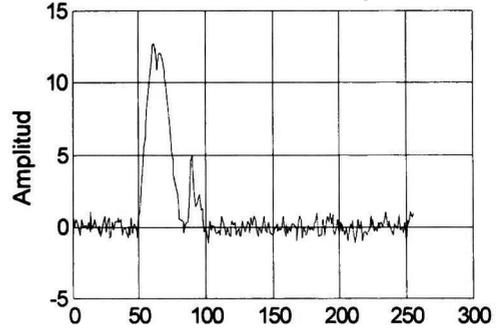
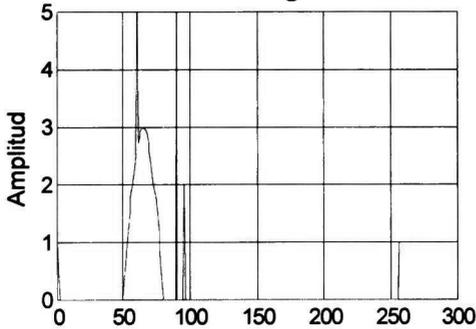
## Función de distancia



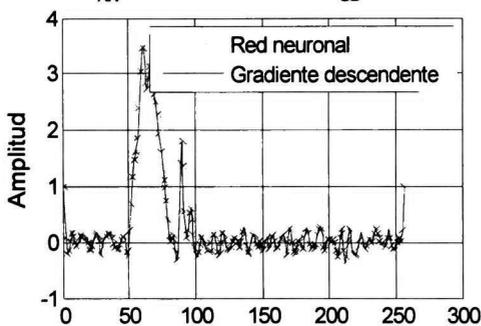
### Señal contaminada

SNR = 5dB Función exp  $\Delta = 25$

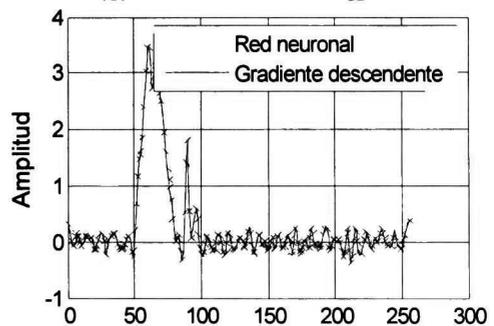
### Señal original



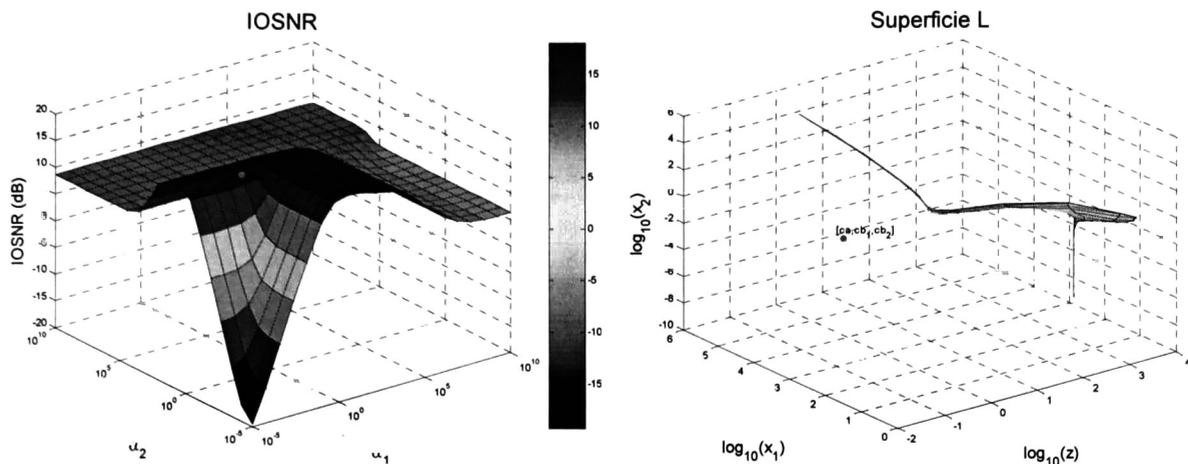
Reconstrucción con AWCLS  
IOSNR<sub>RN</sub> = 17.4406dB IOSNR<sub>GD</sub> = 17.4433dB



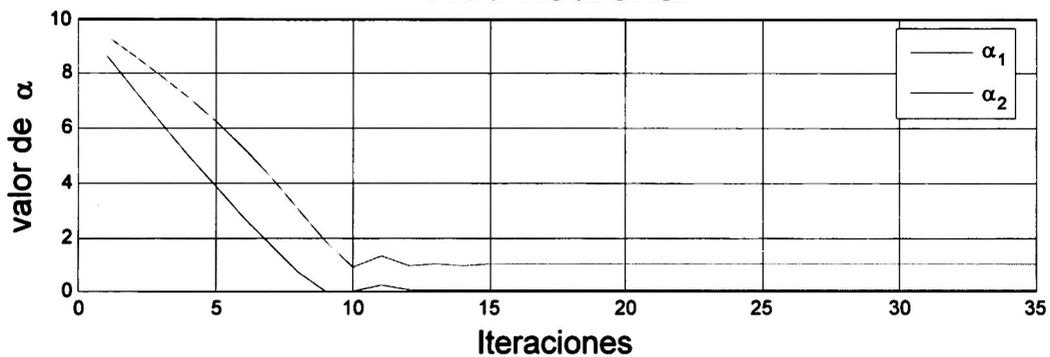
Reconstrucción con WCLS  
IOSNR<sub>RN</sub> = 17.2841dB IOSNR<sub>GD</sub> = 17.2824dB



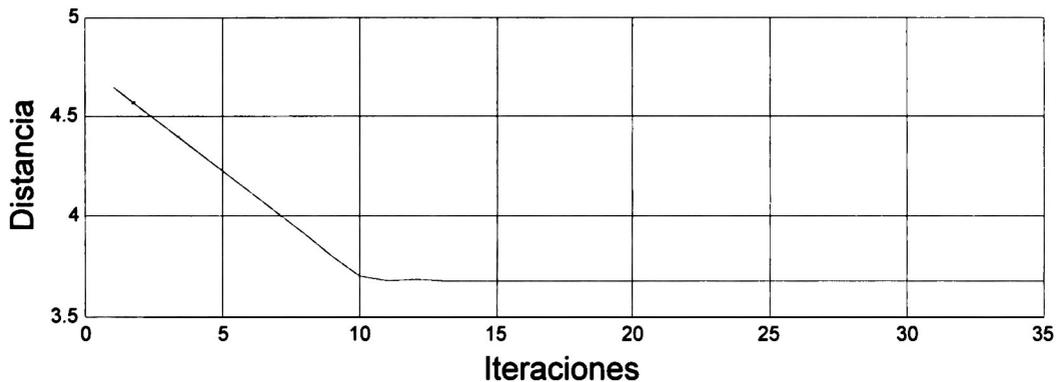
Simulación hecha con función exponencial SNR=15dB y  $\Delta=45$ .



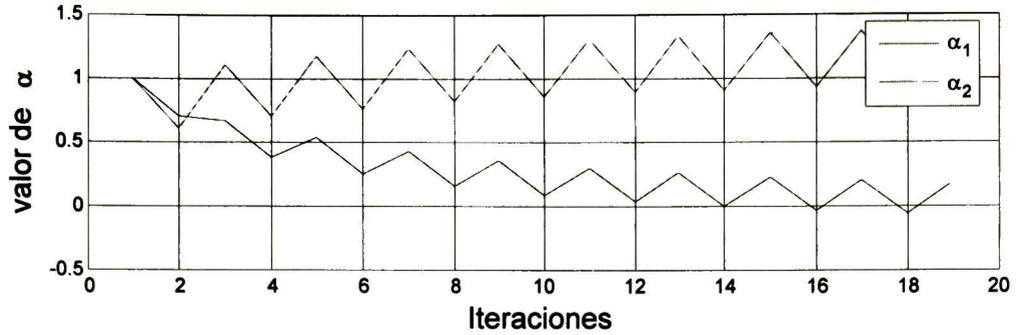
## Convergencia de los parámetros de regularización Red neuronal



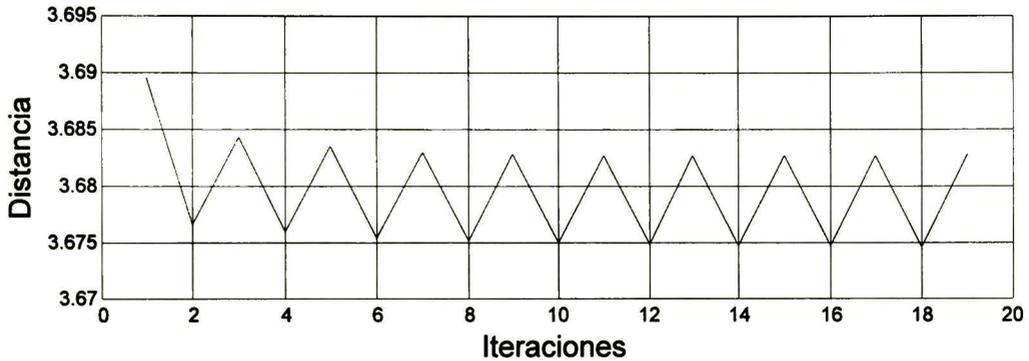
## Función de distancia



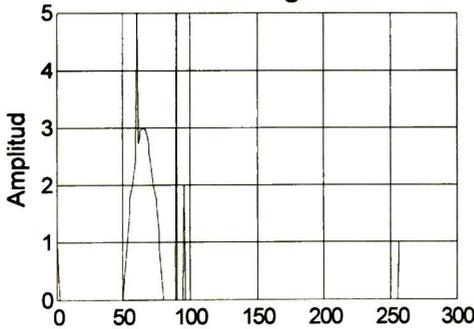
# Convergencia de los parámetros de regularización Gradiente descendente



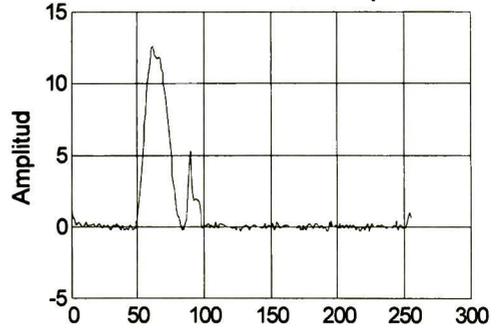
## Función de distancia



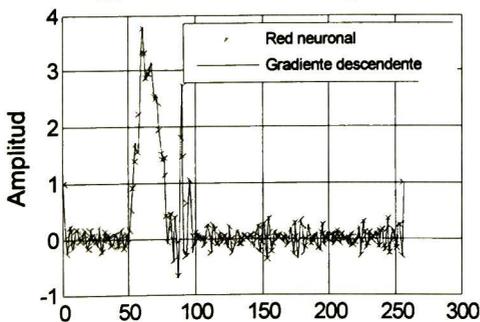
### Señal original



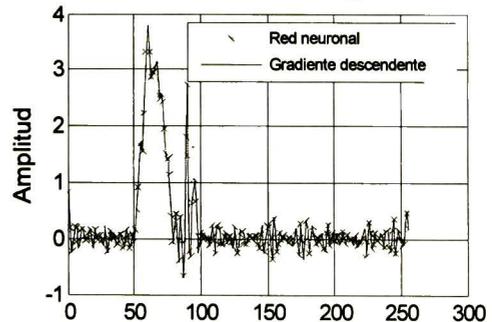
### Señal contaminada SNR = 15dB Función exp $\Delta = 45$



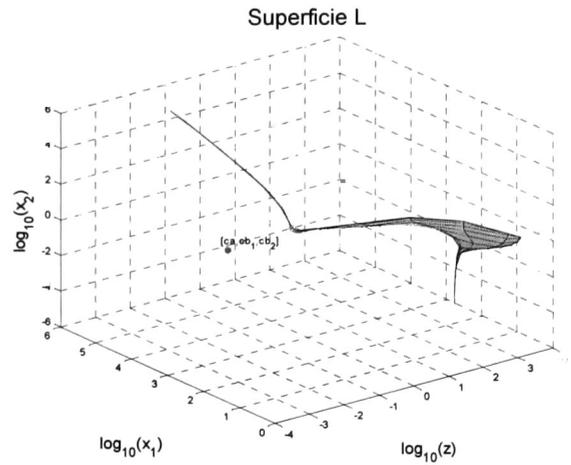
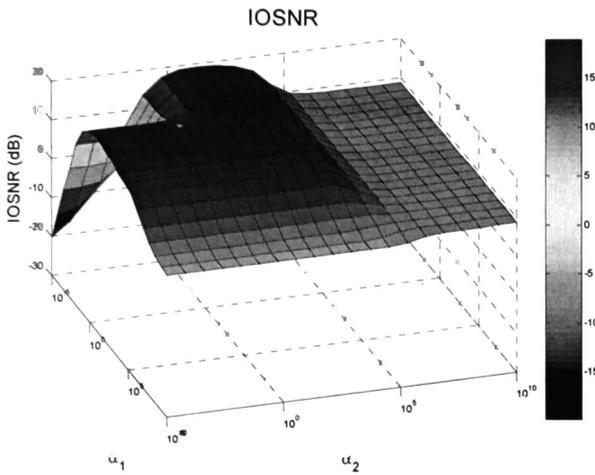
### Reconstrucción con AWCLS IOSNR<sub>RN</sub> = 18.0284dB IOSNR<sub>GD</sub> = 17.9637dB



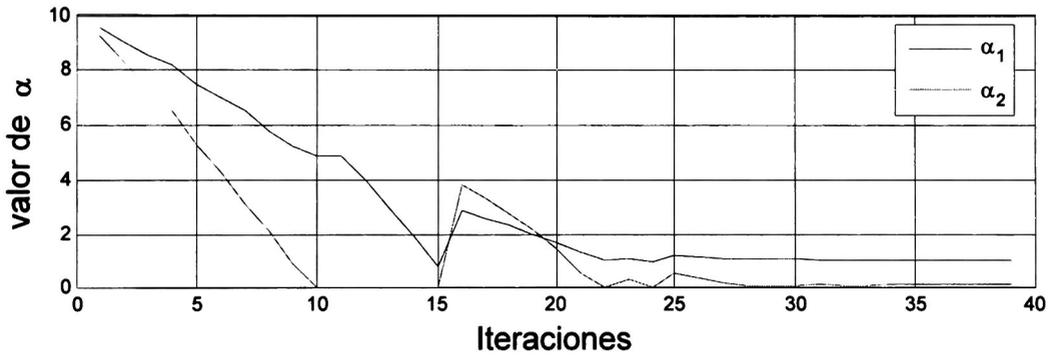
### Reconstrucción con WCLS IOSNR<sub>RN</sub> = 17.8617dB IOSNR<sub>GD</sub> = 17.7954dB



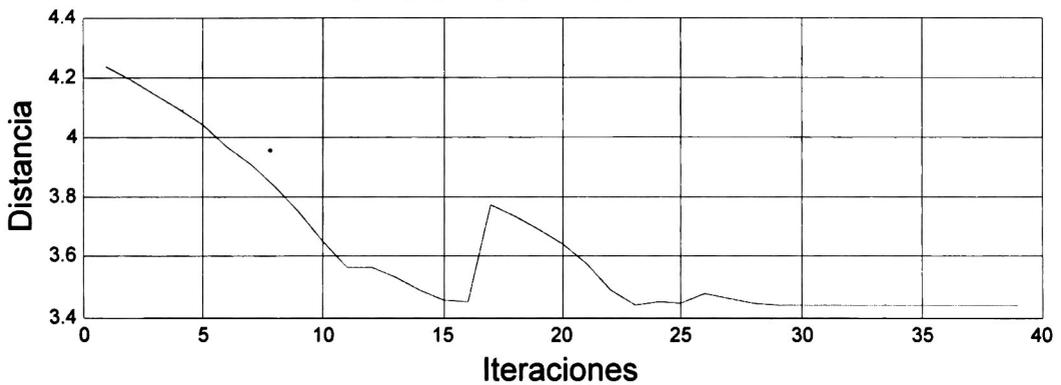
Simulación hecha con función exponencial SNR=30dB y  $\Delta=65$ .



## Convergencia de los parámetros de regularización Red neuronal

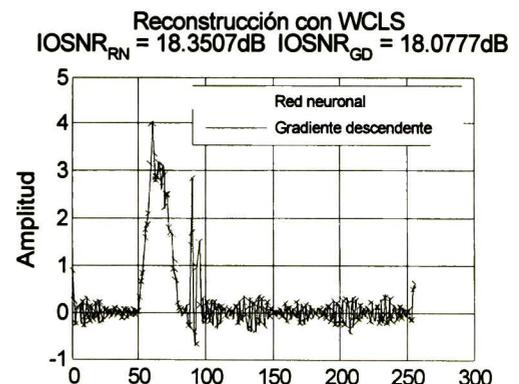
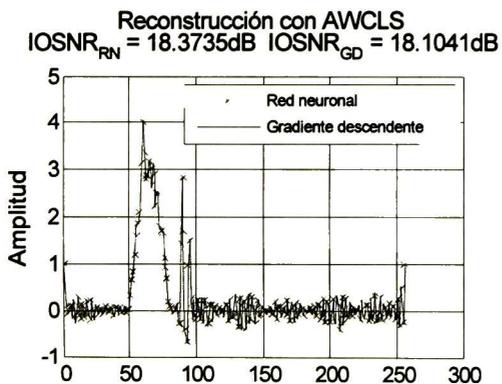
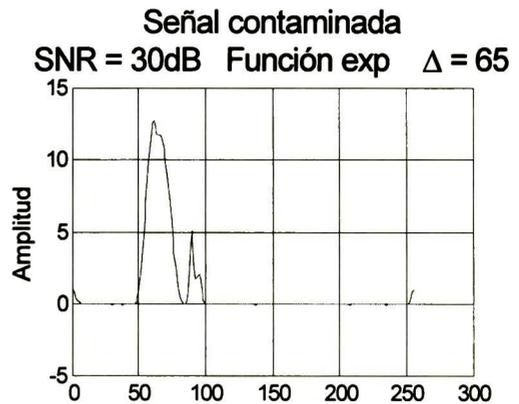
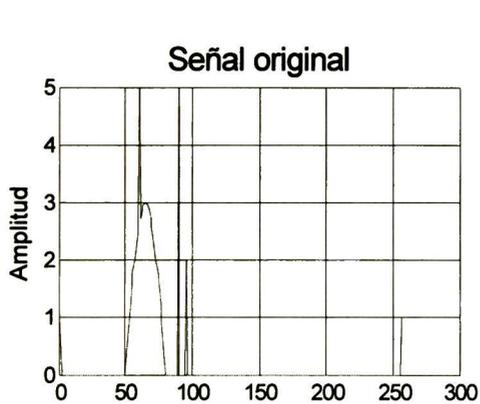
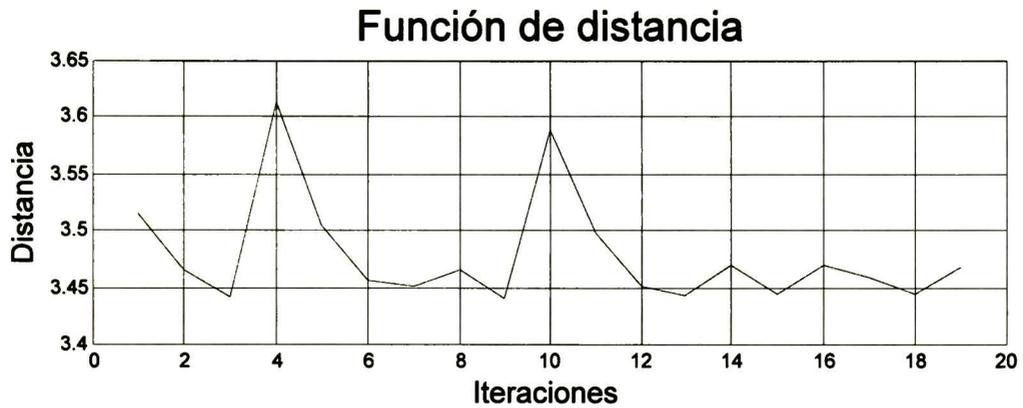
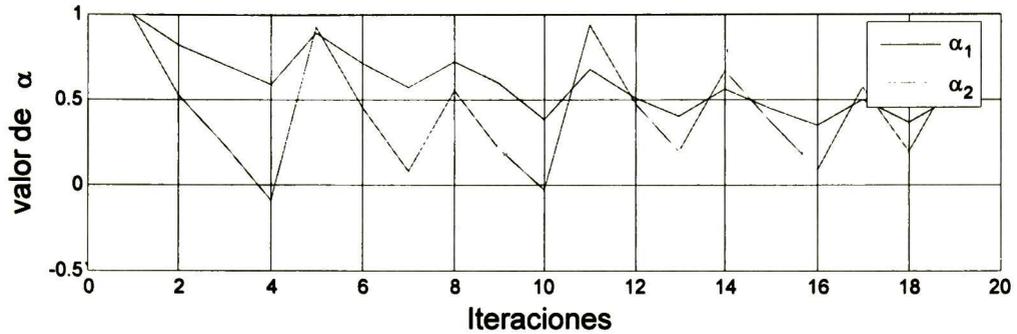


## Función de distancia

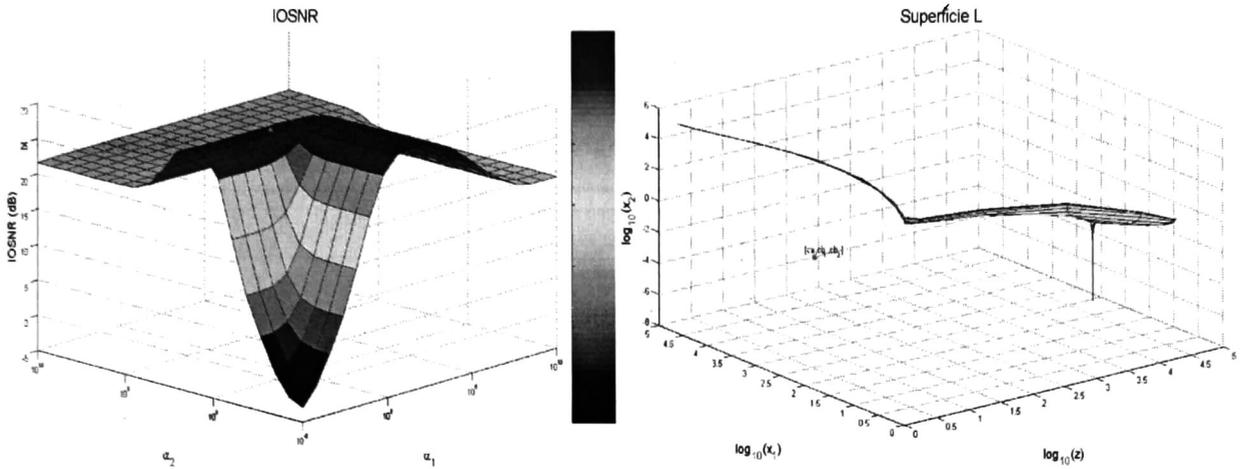


# Convergencia de los parámetros de regularización

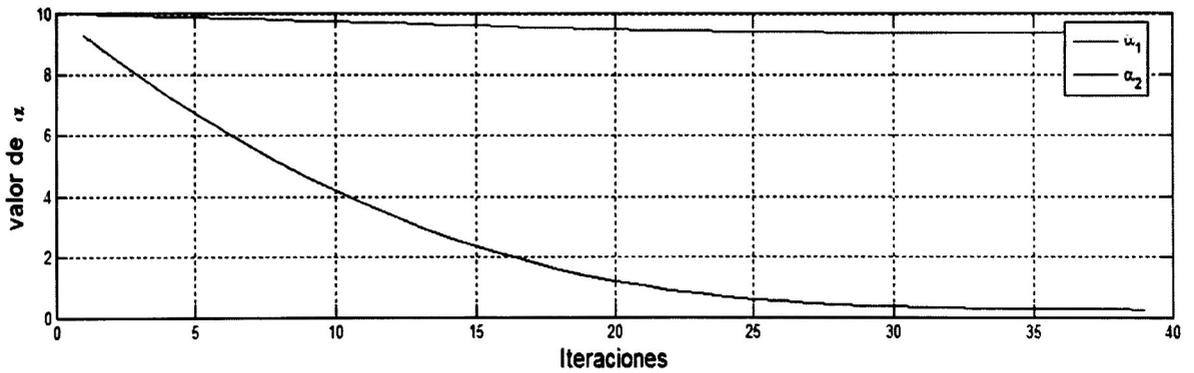
## Gradiente descendente



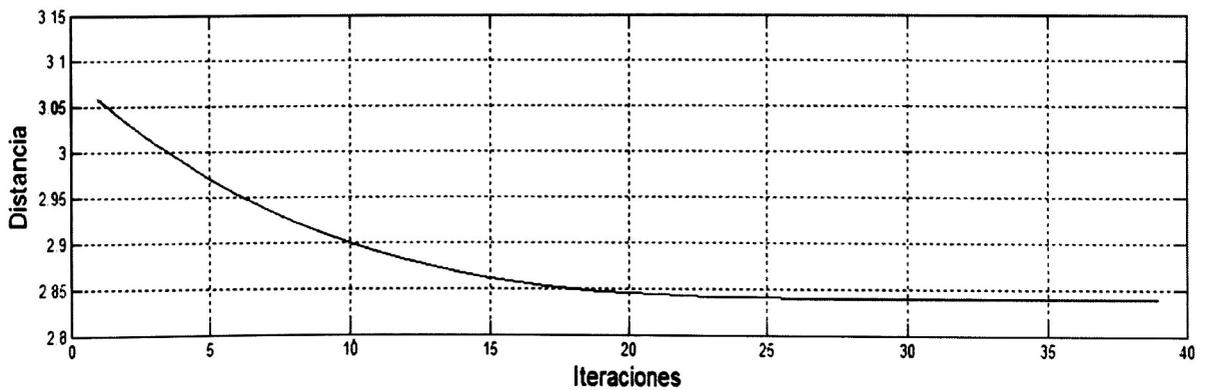
Simulación hecha con función Sinc SNR=5dB y  $\Delta=25$ .



### Convergencia de los parámetros de regularización Red neuronal

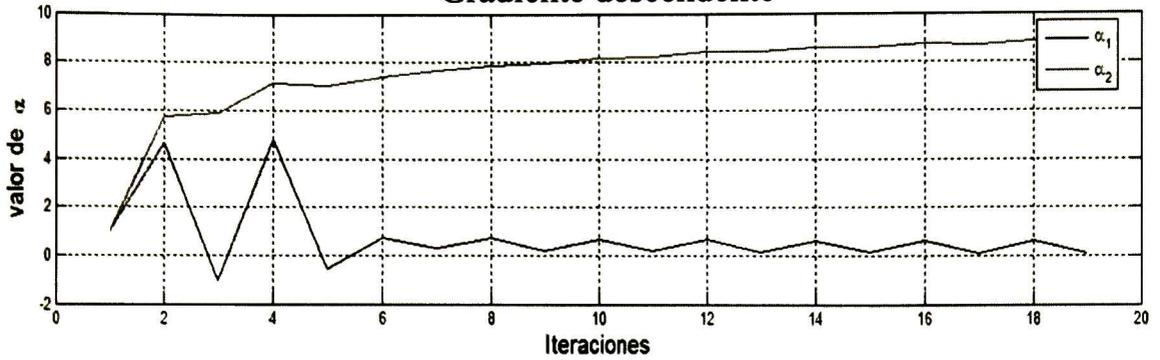


### Función de distancia

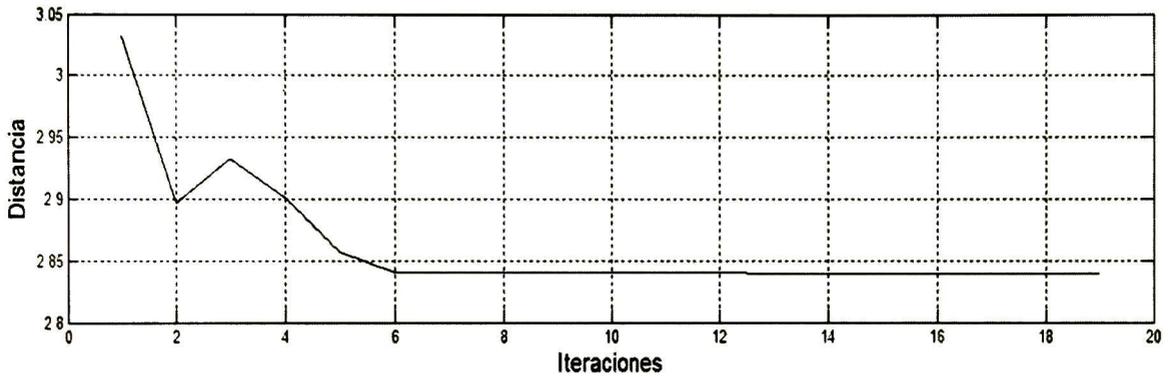


# Convergencia de los parámetros de regularización

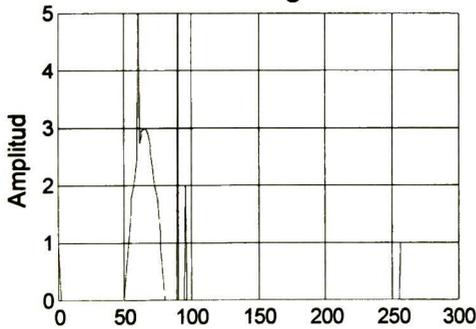
## Gradiente descendente



## Función de distancia

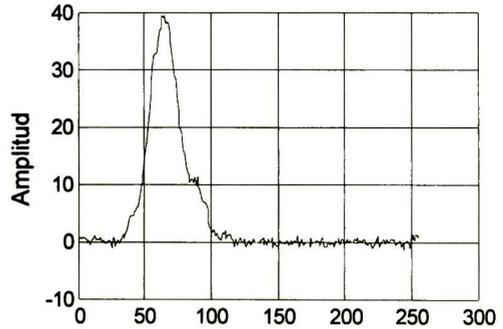


## Señal original



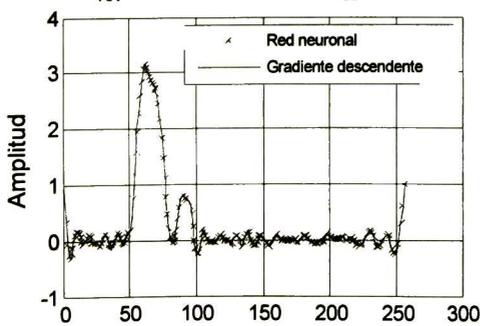
## Señal contaminada

SNR = 5dB Función sinc  $\Delta = 25$



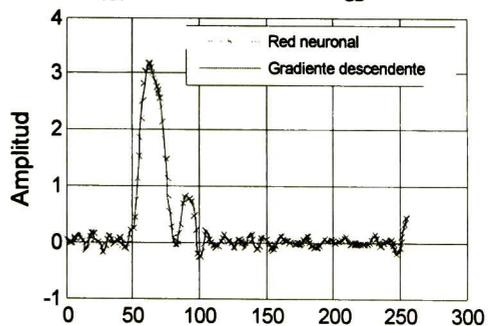
## Reconstrucción con AWCLS

IOSNR<sub>RN</sub> = 29.3332dB IOSNR<sub>GD</sub> = 29.3509dB

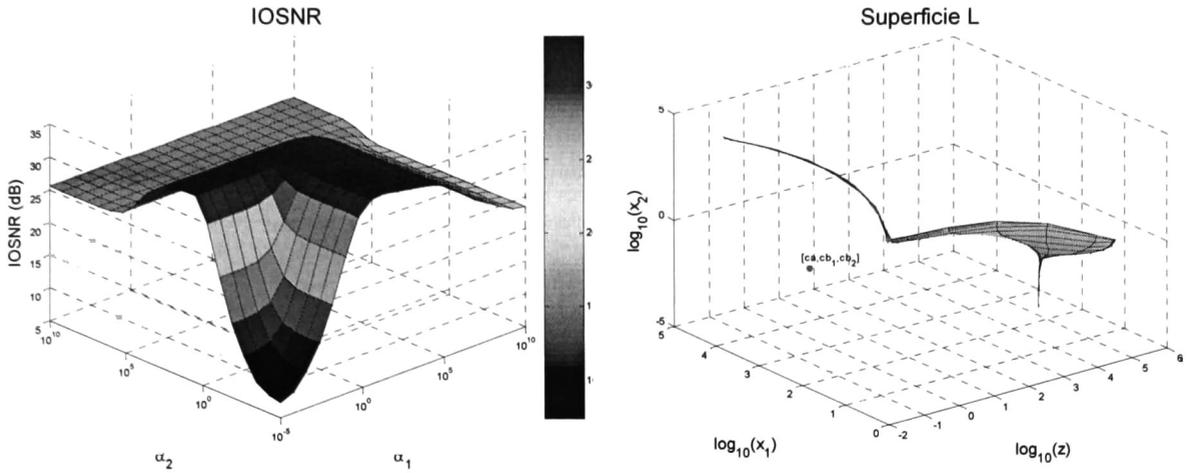


## Reconstrucción con WCLS

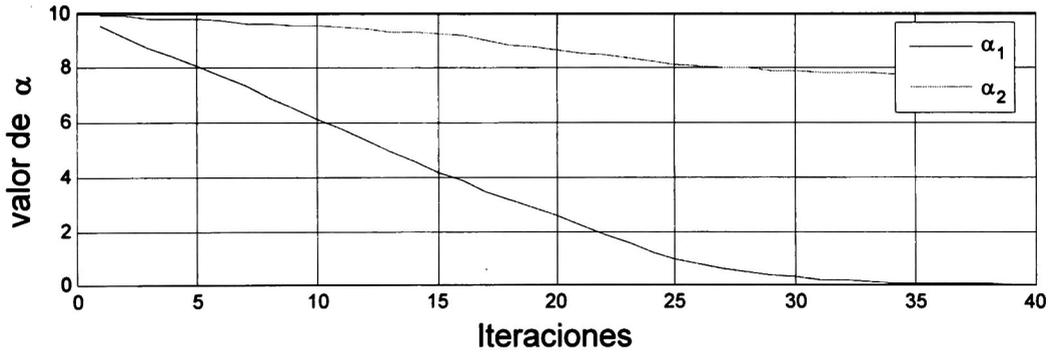
IOSNR<sub>RN</sub> = 29.2426dB IOSNR<sub>GD</sub> = 29.2503dB



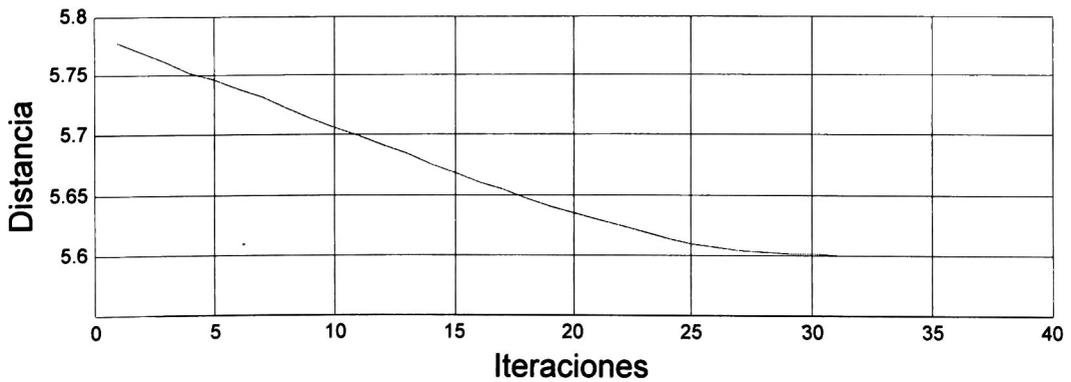
Simulación hecha con función Sinc SNR=15dB y  $\Delta=45$



## Convergencia de los parámetros de regularización Red neuronal

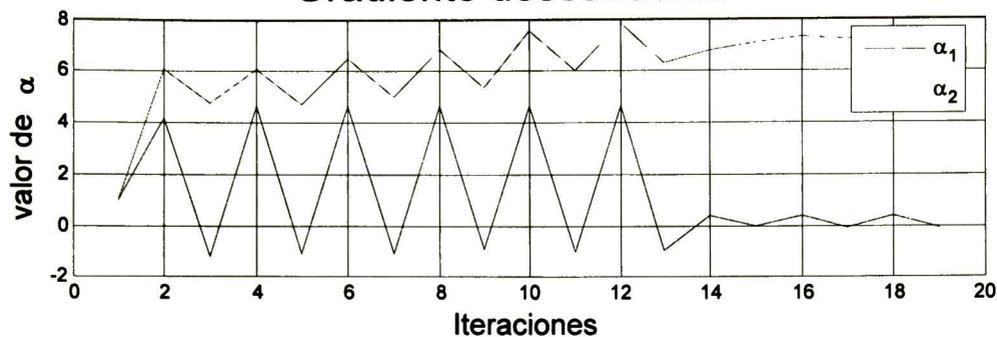


## Función de distancia

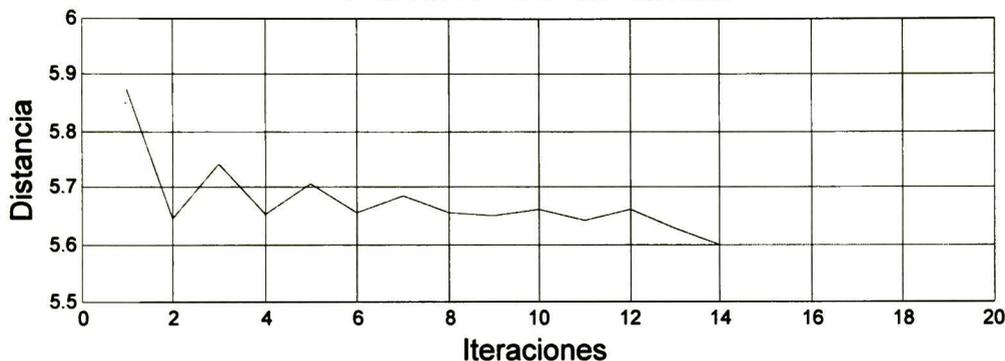


# Convergencia de los parámetros de regularización

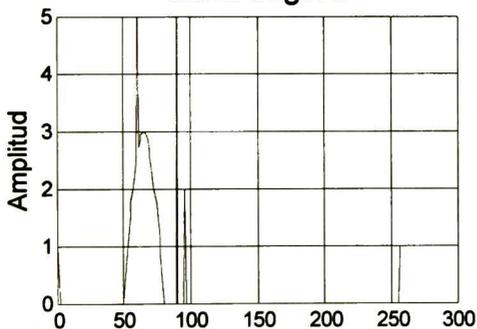
## Gradiente descendente



## Función de distancia

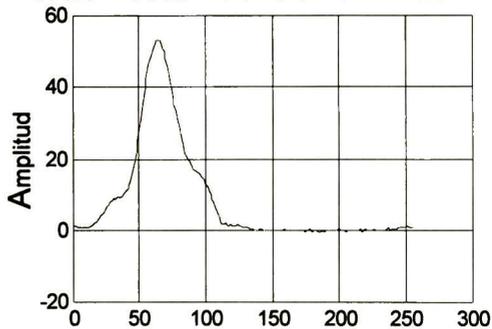


## Señal original



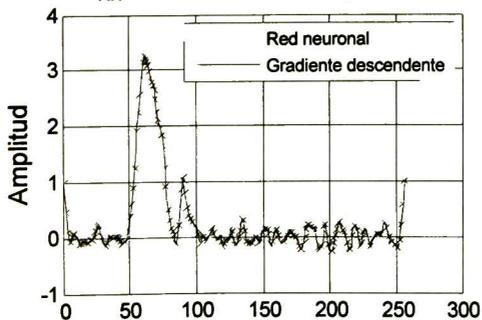
## Señal contaminada

SNR = 15dB Función sinc  $\Delta = 45$



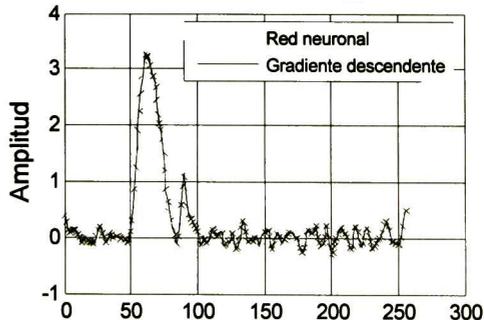
## Reconstrucción con AWCLS

IOSNR<sub>RN</sub> = 33.4161dB IOSNR<sub>GD</sub> = 33.4485dB

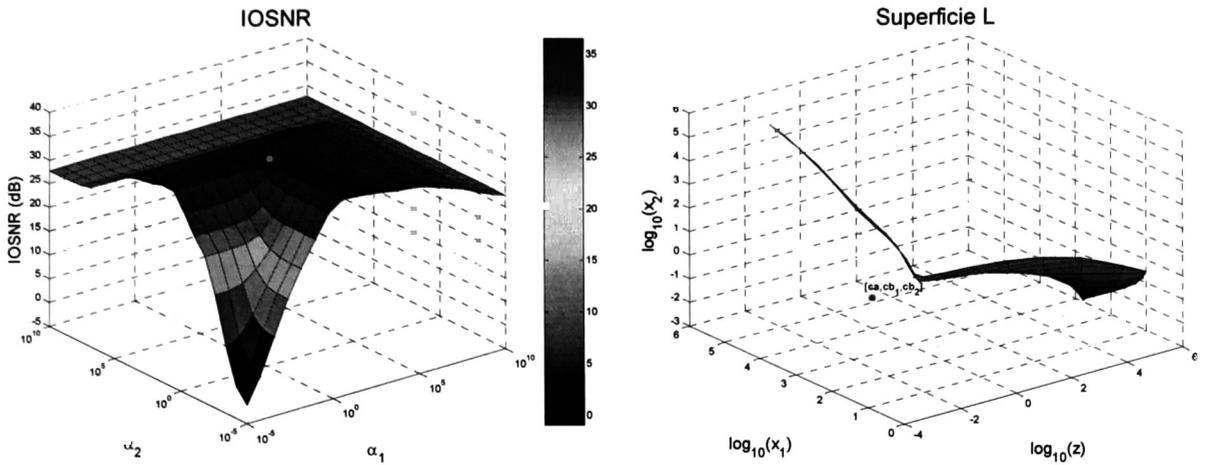


## Reconstrucción con WCLS

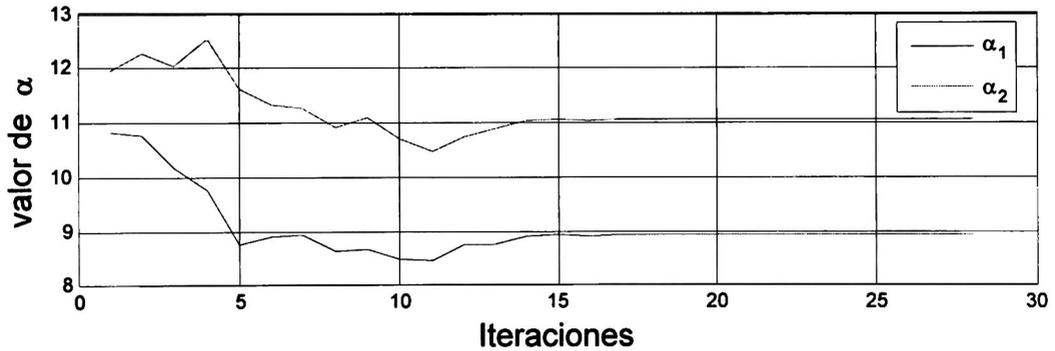
IOSNR<sub>RN</sub> = 33.4137dB IOSNR<sub>GD</sub> = 33.4482dB



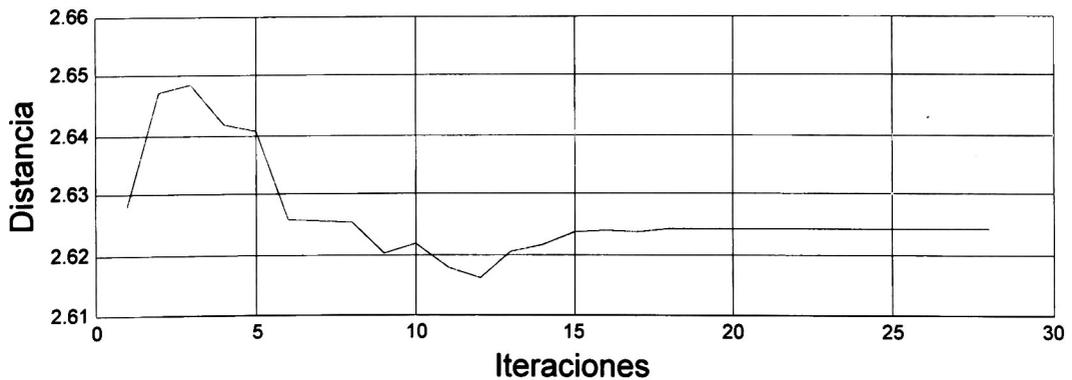
Simulación hecha con función Sinc SNR=30db y  $\Delta=65$



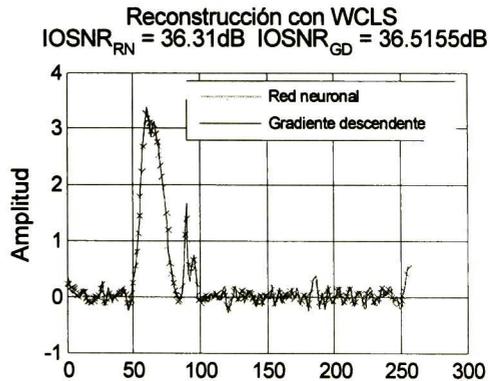
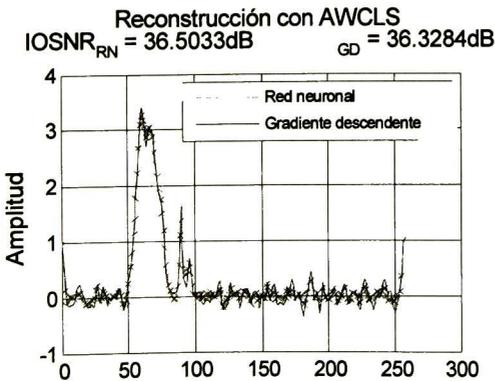
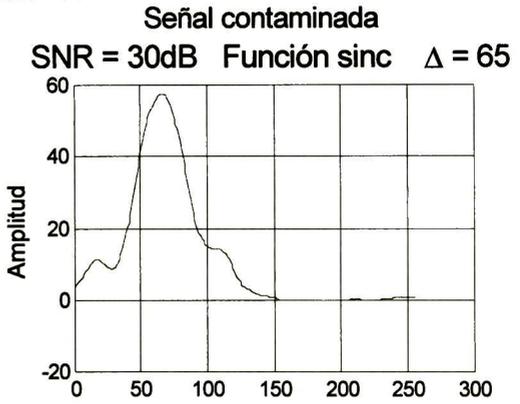
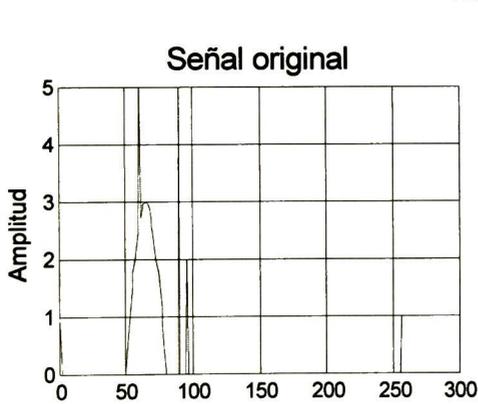
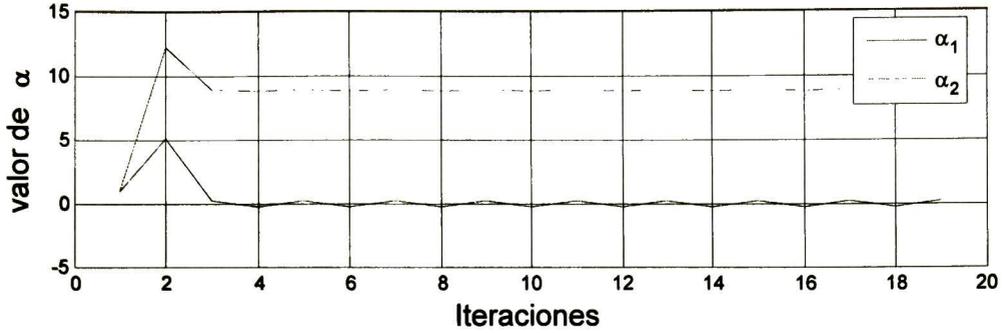
## Convergencia de los parámetros de regularización Red neuronal



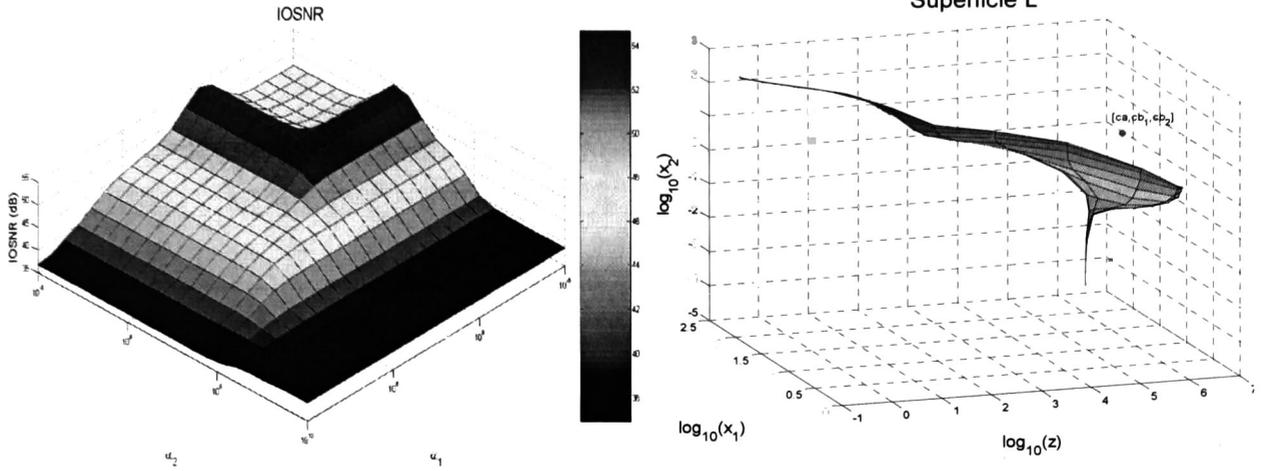
## Función de distancia



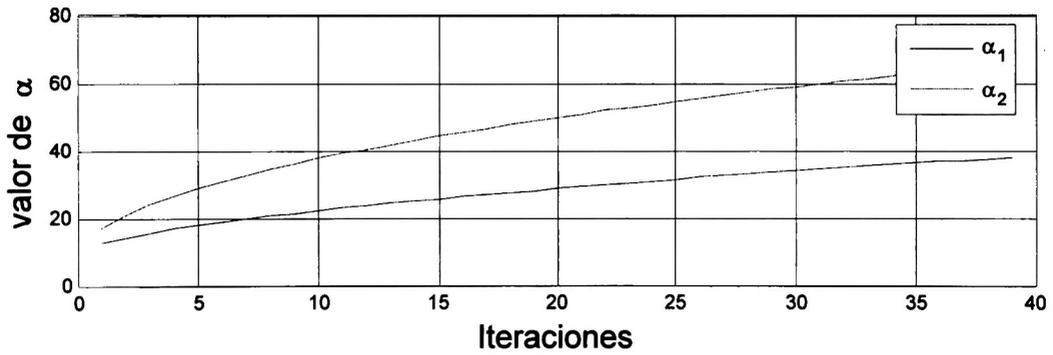
# Convergencia de los parámetros de regularización Gradiente descendente



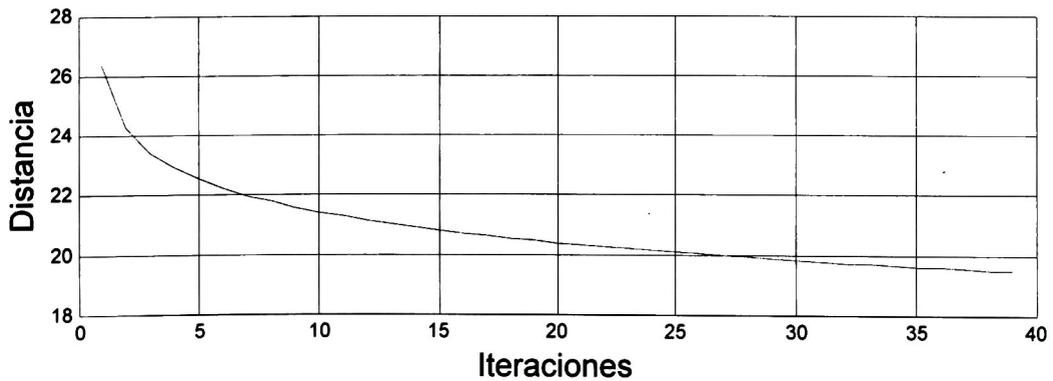
Simulación hecha con función rectangular SNR=5db y  $\Delta=25$



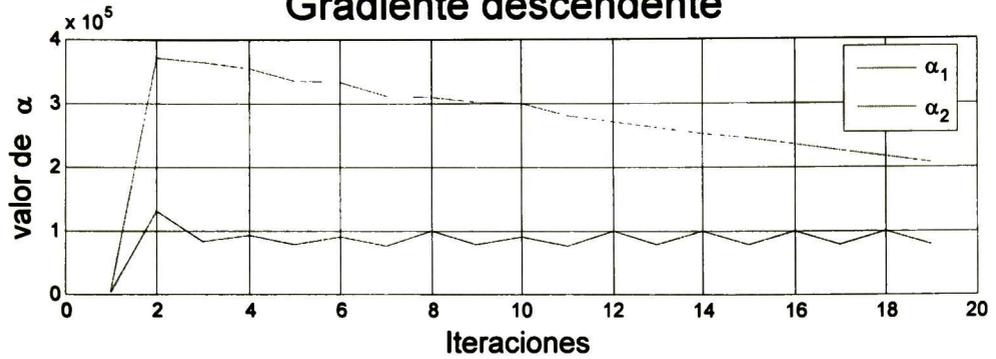
## Convergencia de los parámetros de regularización Red neuronal



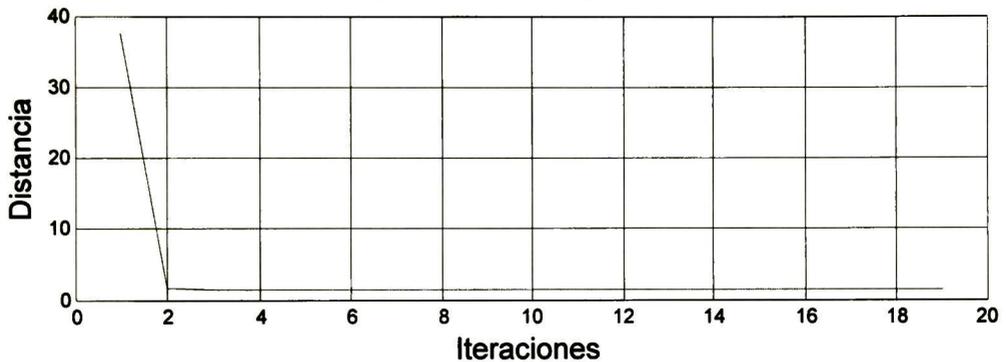
## Función de distancia



# Convergencia de los parámetros de regularización Gradiente descendente

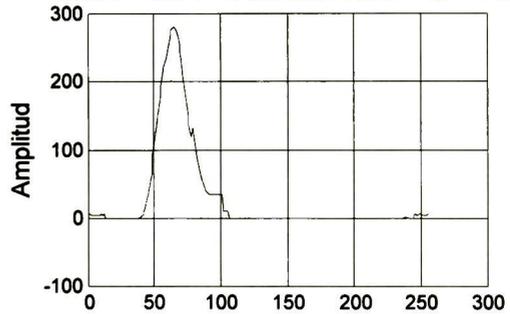


## Función de distancia

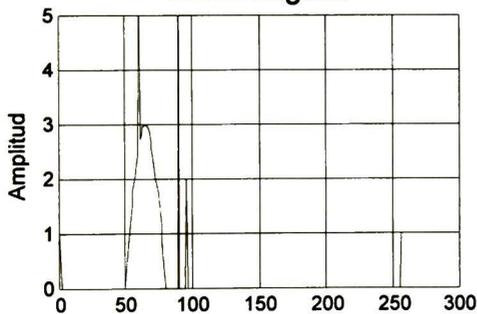


### Señal contaminada

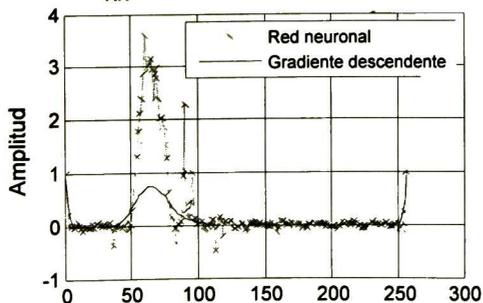
SNR = 5dB Función rect  $\Delta = 25$



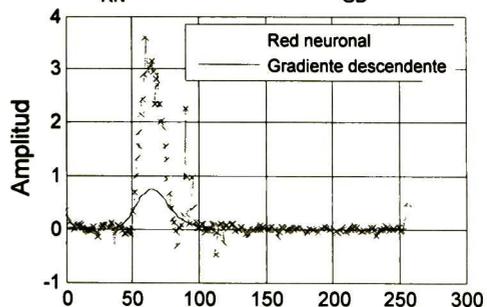
### Señal original



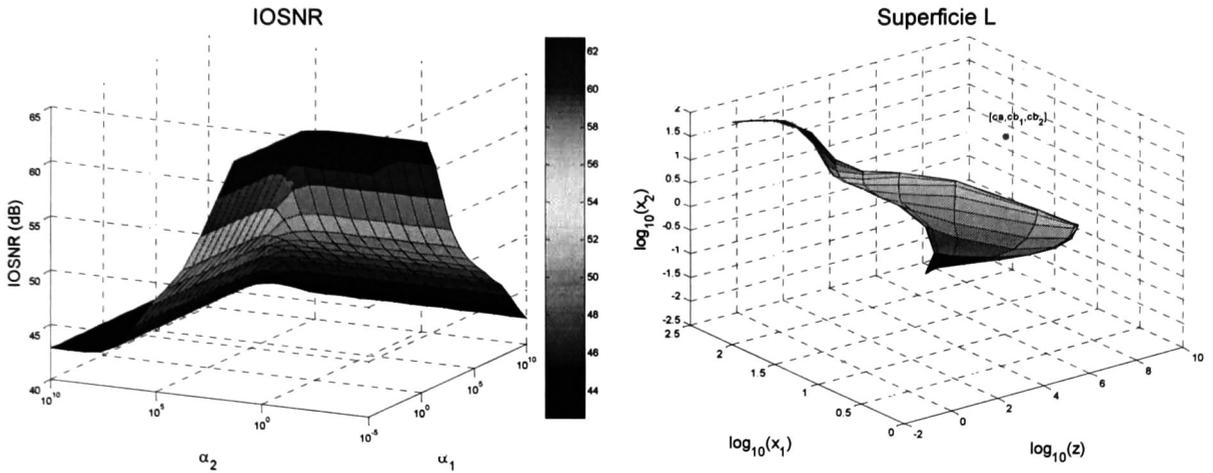
Reconstrucción con AWCLS  
IOSNR<sub>RN</sub> = 49.7714dB IOSNR<sub>GD</sub> = 40.912dB



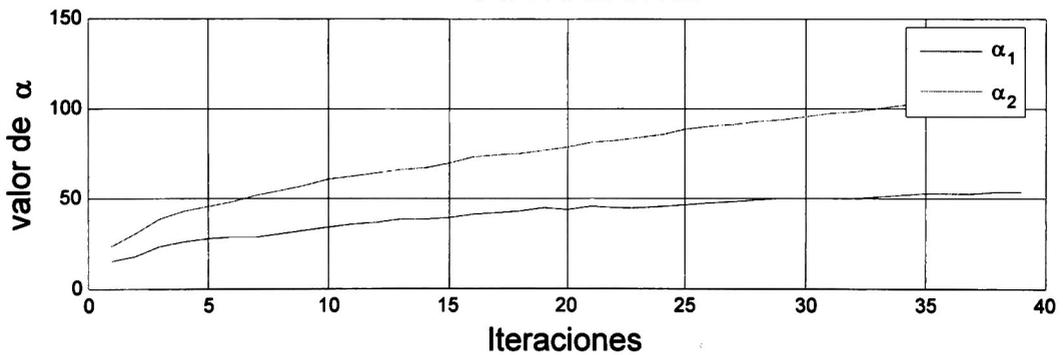
Reconstrucción con WCLS  
IOSNR<sub>RN</sub> = 49.5515dB IOSNR<sub>GD</sub> = 40.8599dB



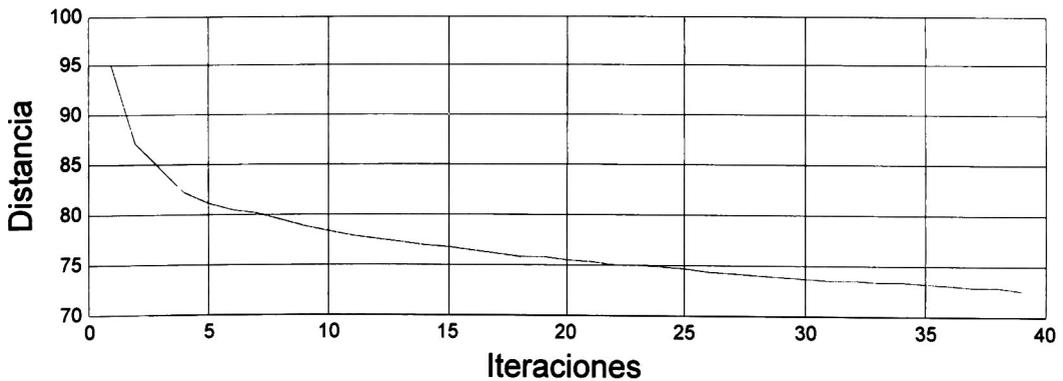
Simulación hecha con función rectangular SNR=15db y  $\Delta=45$



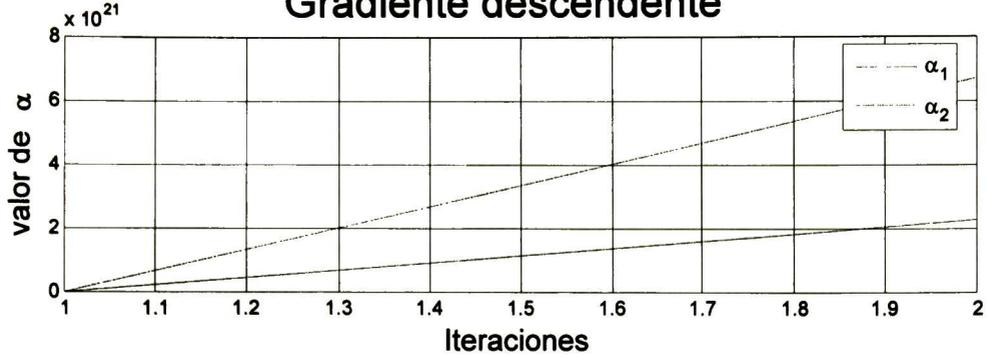
## Convergencia de los parámetros de regularización Red neuronal



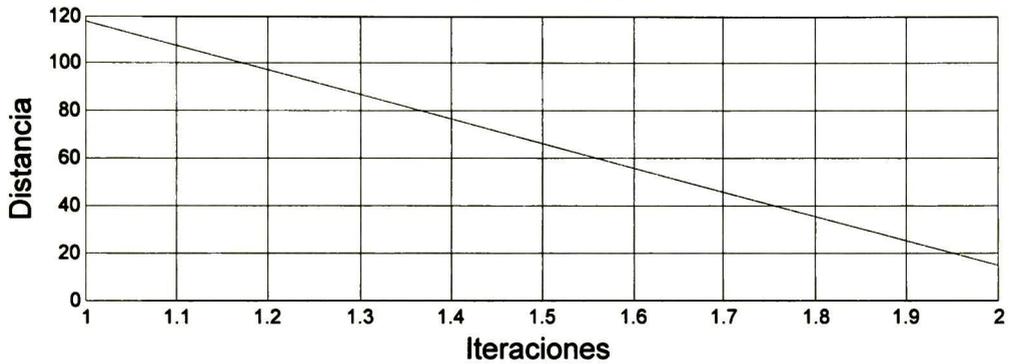
## Función de distancia



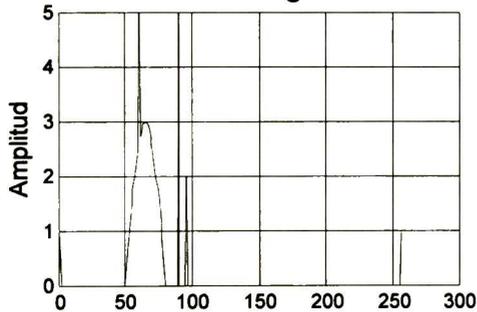
# Convergencia de los parámetros de regularización Gradiente descendente



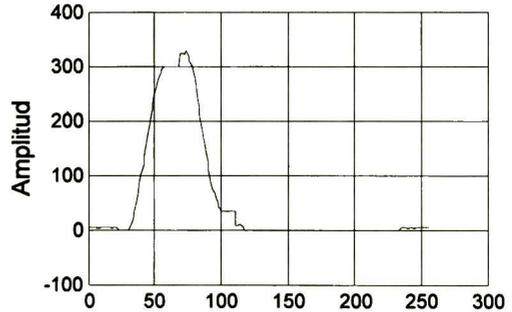
## Función de distancia



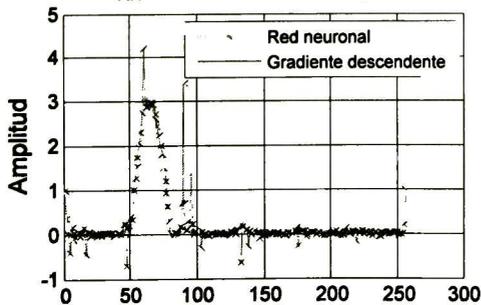
### Señal original



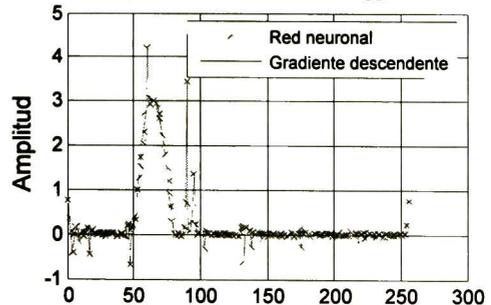
### Señal contaminada SNR = 15dB Función rect $\Delta = 45$



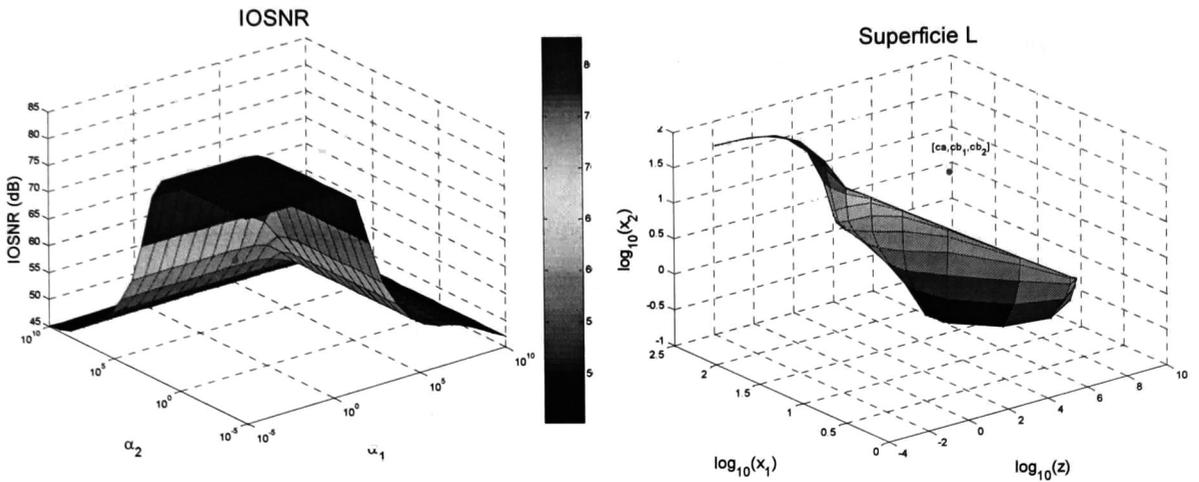
### Reconstrucción con AWCLS IOSNR<sub>RN</sub> = 56.8012dB IOSNR<sub>GD</sub> = NaN dB



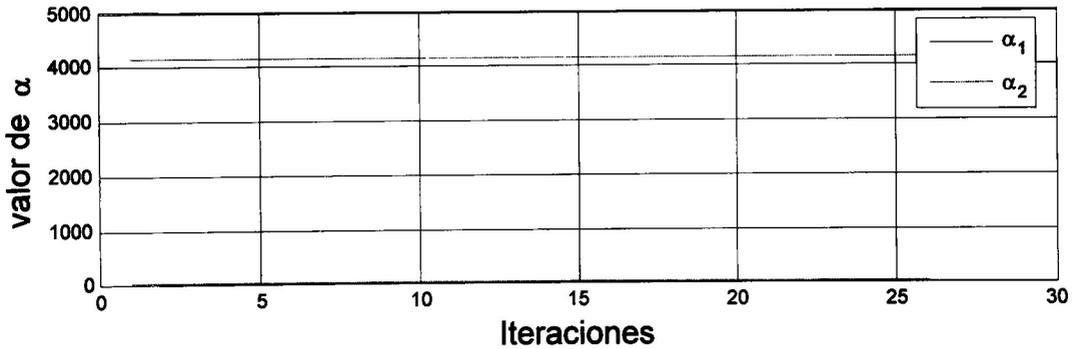
### Reconstrucción con WCLS IOSNR<sub>RN</sub> = 56.7944dB IOSNR<sub>GD</sub> = NaN dB



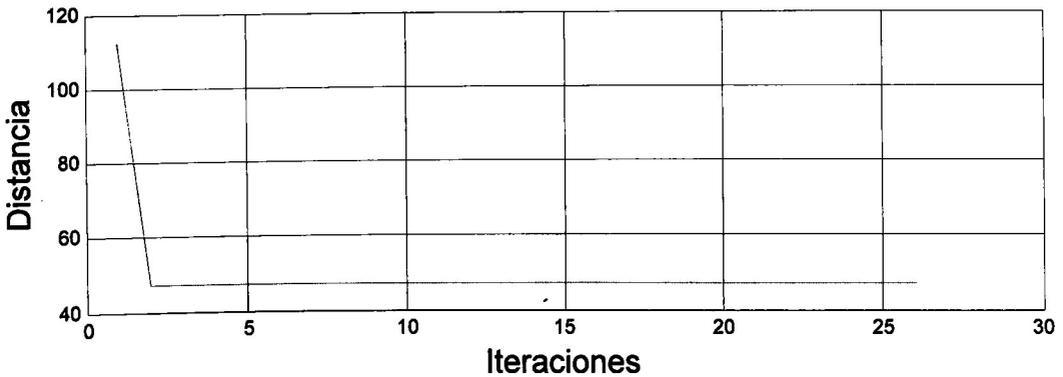
Simulación hecha con función rectangular SNR=30db y  $\Delta=65$



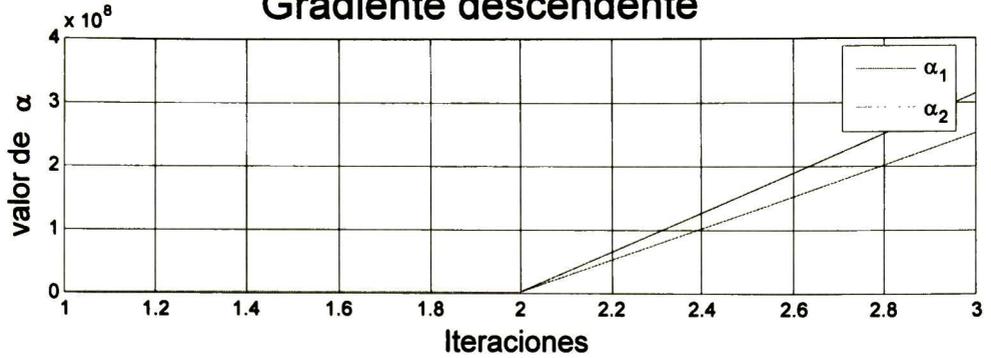
### Convergencia de los parámetros de regularización Red neuronal



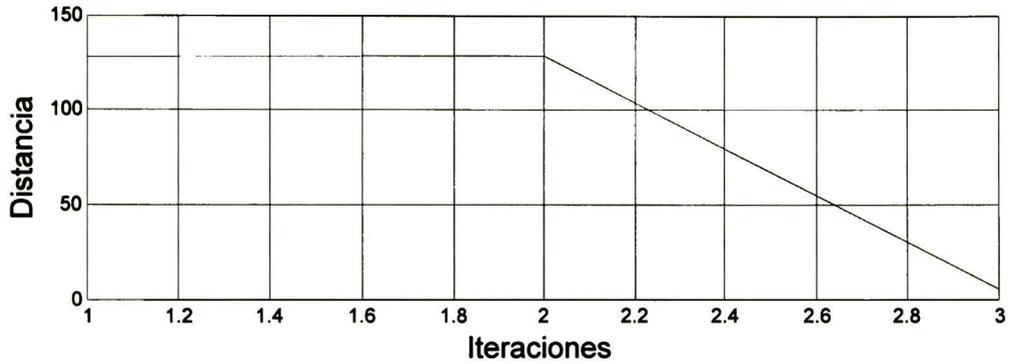
### Función de distancia



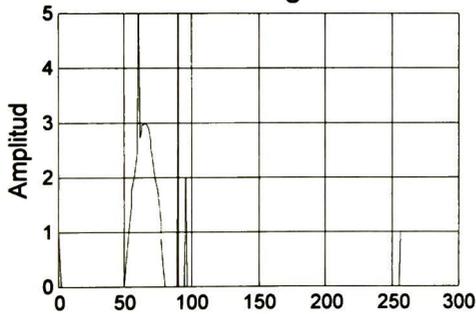
# Convergencia de los parámetros de regularización Gradiente descendente



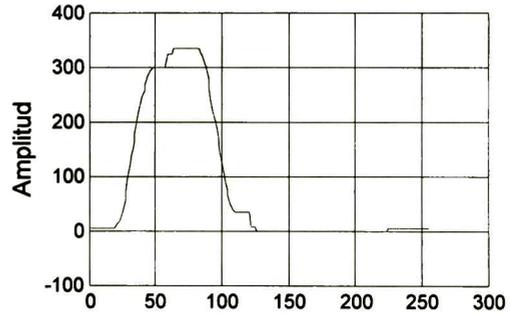
## Función de distancia



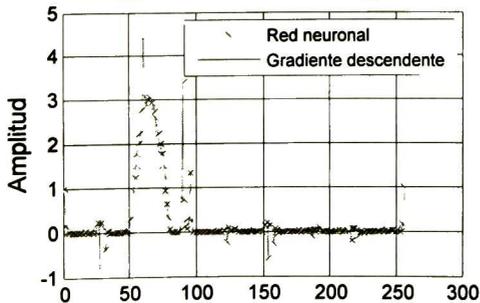
### Señal original



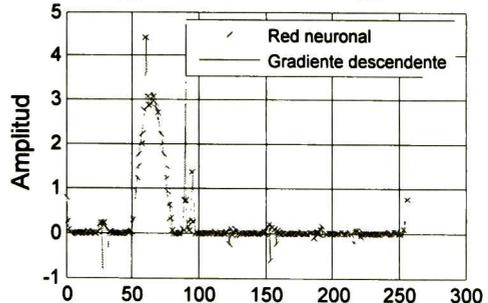
### Señal contaminada SNR = 30dB Función rect $\Delta = 65$



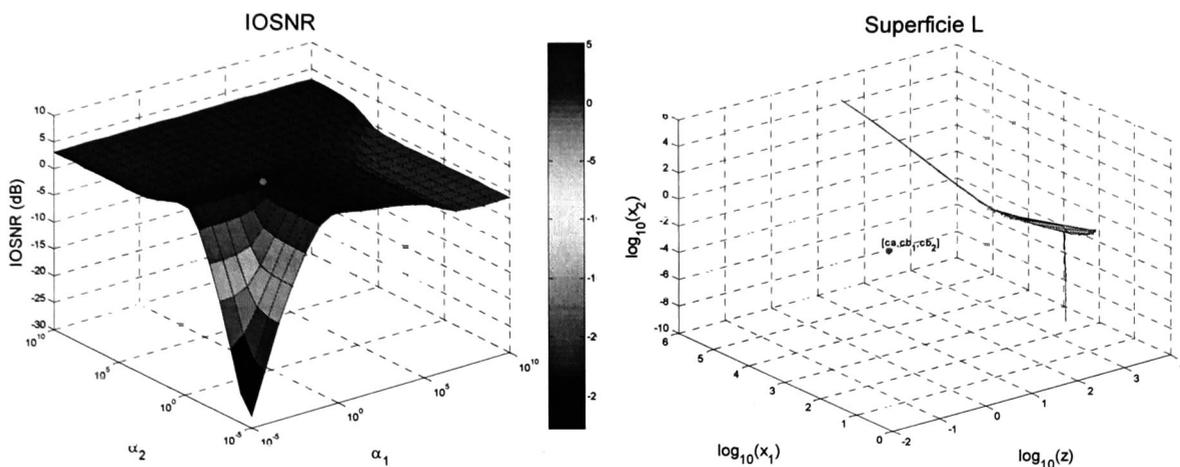
### Reconstrucción con AWCLS IOSNR<sub>RN</sub> = 59.3267dB IOSNR<sub>GD</sub> = NaNdB



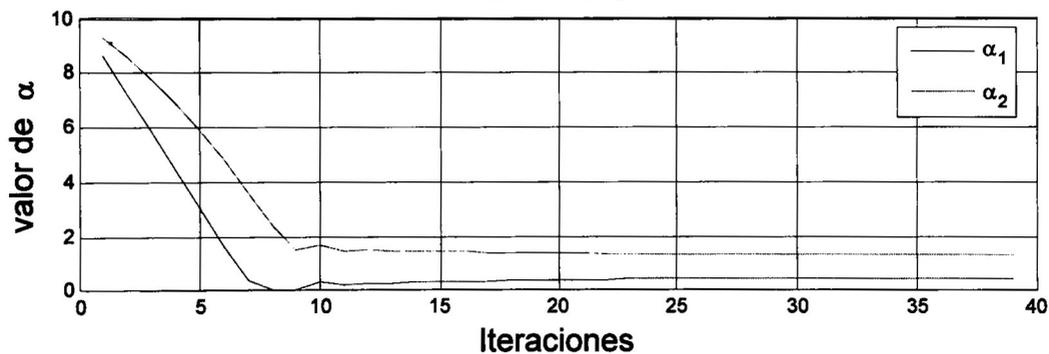
### Reconstrucción con WCLS IOSNR<sub>RN</sub> = 59.1784dB IOSNR<sub>GD</sub> = NaNdB



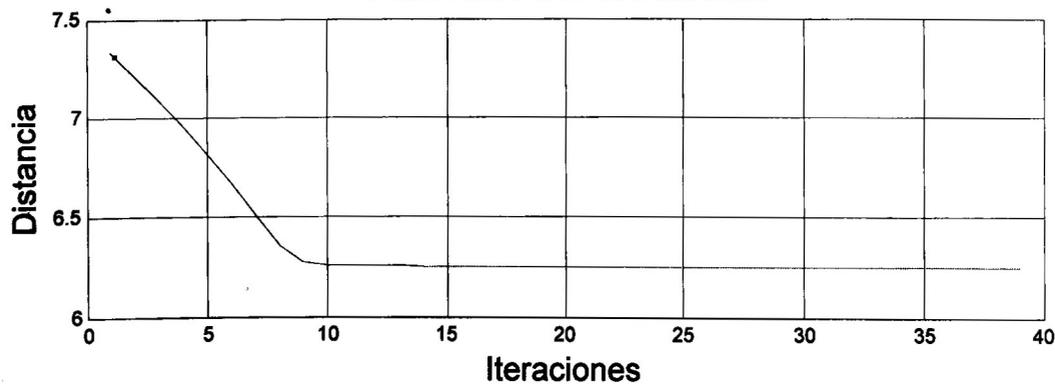
Simulación hecha con función exponencial SNR=5db y  $\Delta=25$



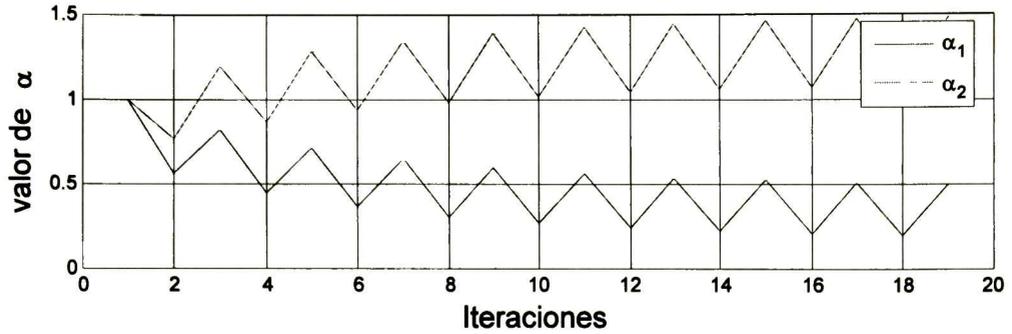
## Convergencia de los parámetros de regularización Red neuronal



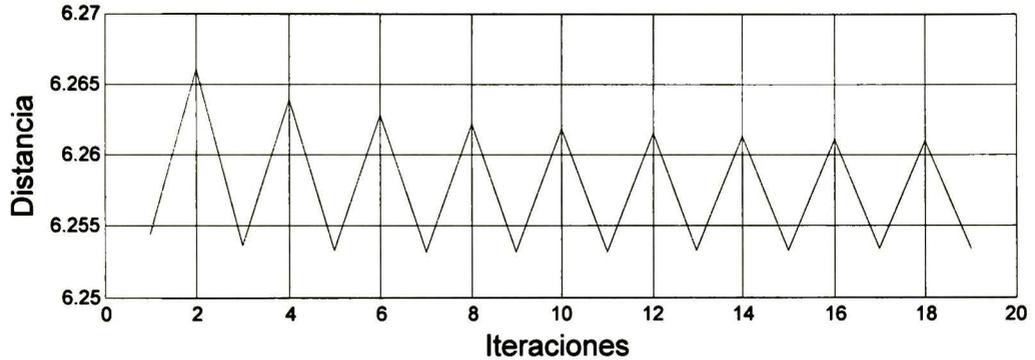
## Función de distancia



# Convergencia de los parámetros de regularización Gradiente descendente



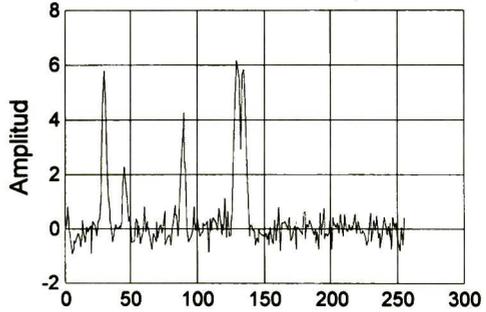
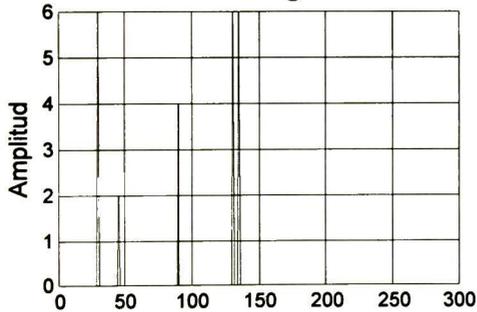
## Función de distancia



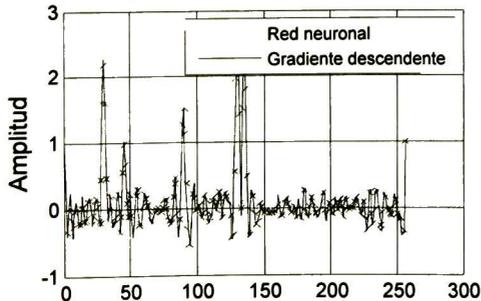
Señal contaminada

SNR = 5dB Función exp  $\Delta = 25$

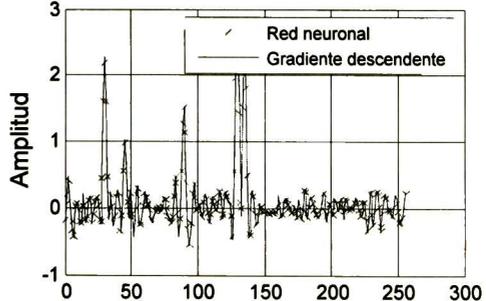
Señal original



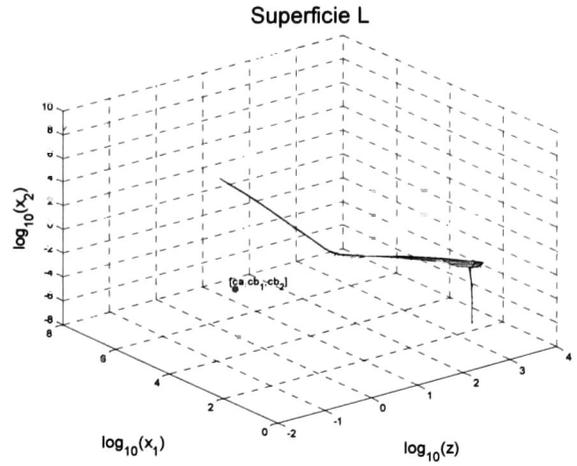
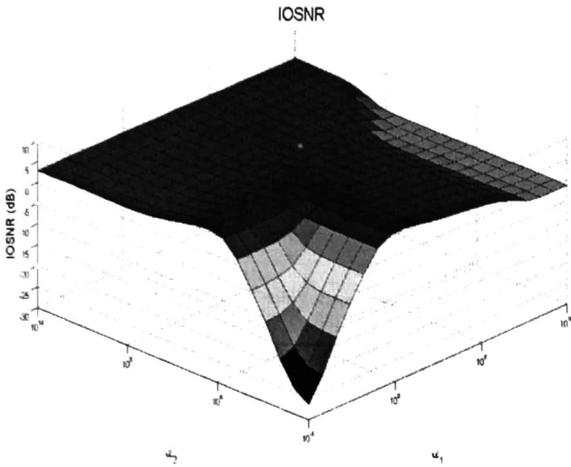
Reconstrucción con AWCLS  
IOSNR<sub>RN</sub> = 5.3639dB IOSNR<sub>GD</sub> = 5.4109dB



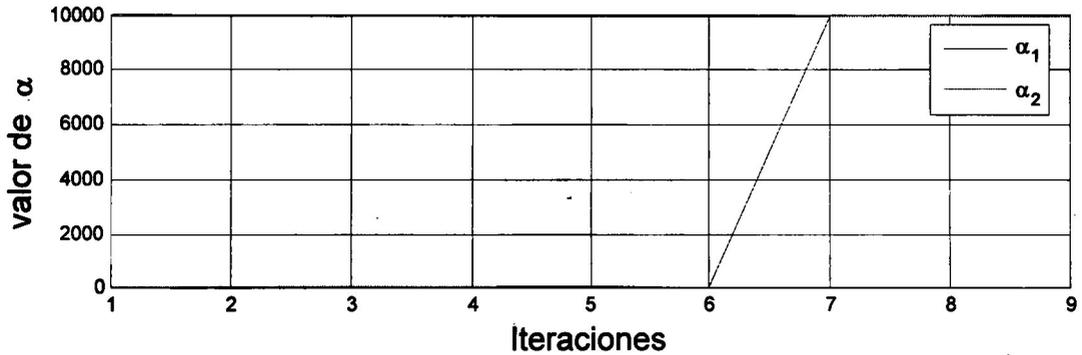
Reconstrucción con WCLS  
IOSNR<sub>RN</sub> = 5.4759dB IOSNR<sub>GD</sub> = 5.5263dB



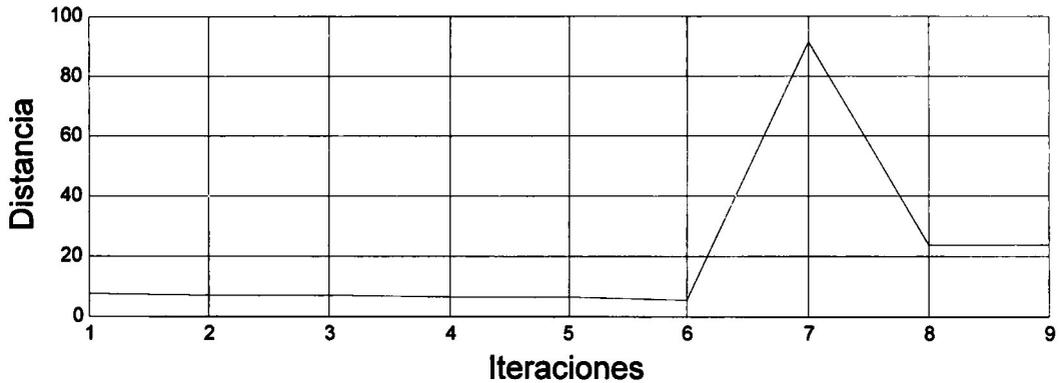
Simulación hecha con función exponencial SNR=15db y  $\Delta=45$



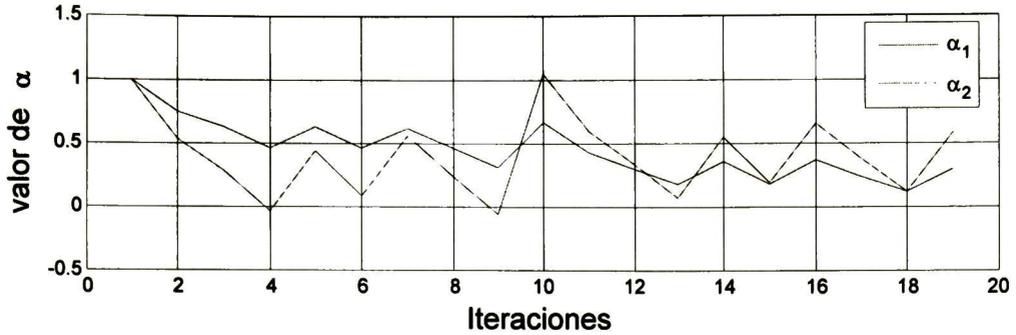
## Convergencia de los parámetros de regularización Red neuronal



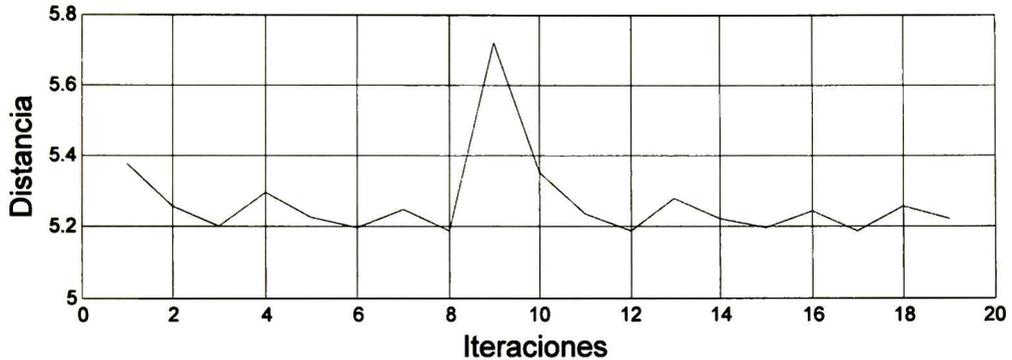
## Función de distancia



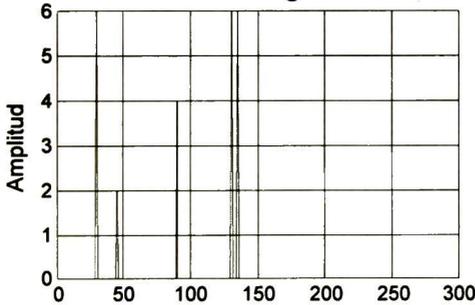
# Convergencia de los parámetros de regularización Gradiente descendente



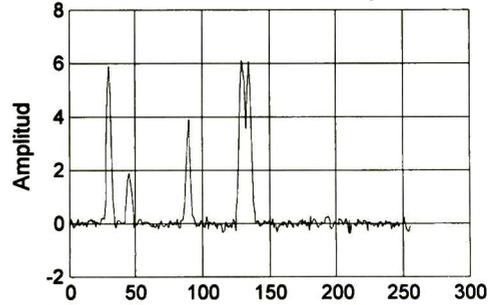
## Función de distancia



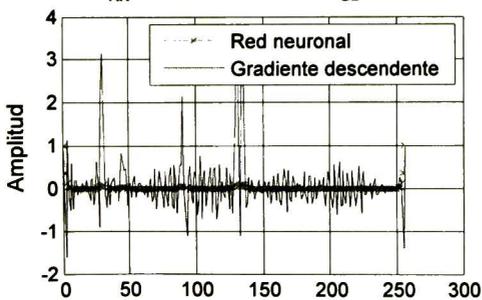
### Señal original



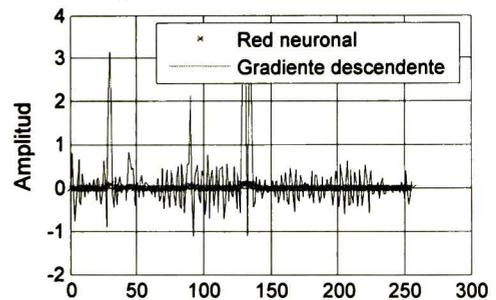
### Señal contaminada SNR = 15dB Función exp $\Delta = 45$



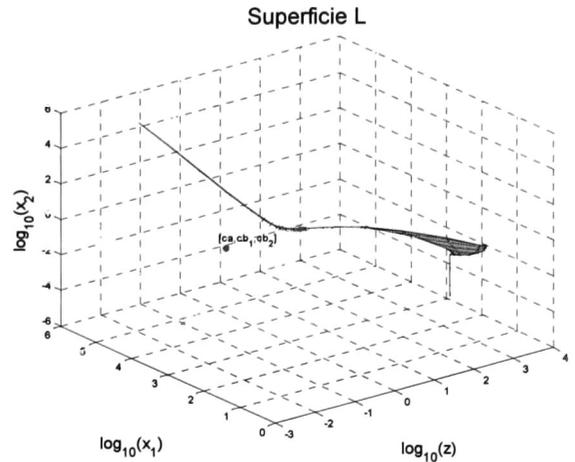
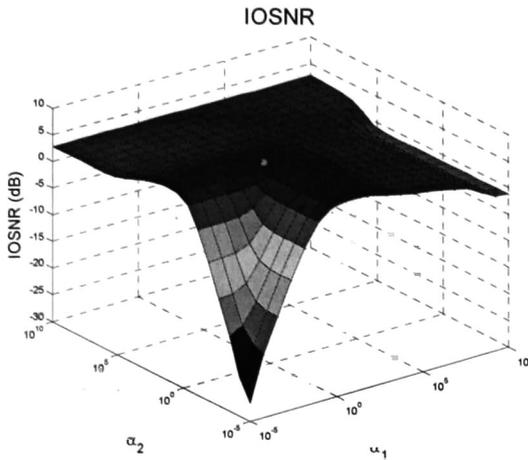
### Reconstrucción con AWCLS IOSNR<sub>RN</sub> = 2.9703dB IOSNR<sub>GD</sub> = 4.6444dB



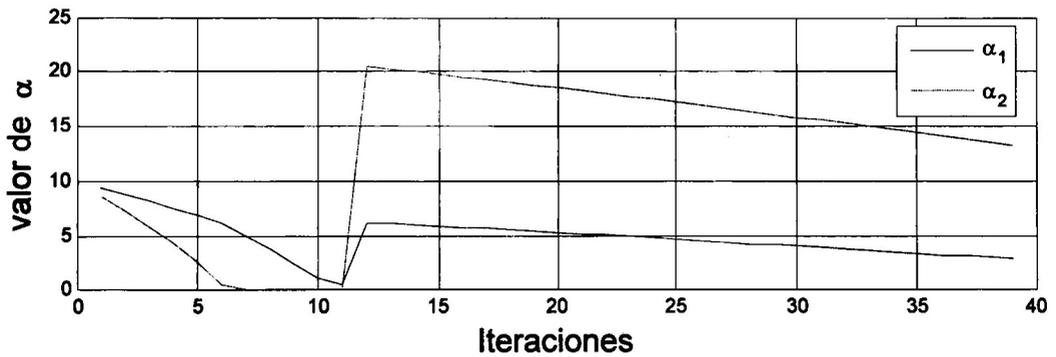
### Reconstrucción con WCLS IOSNR<sub>RN</sub> = 3.0504dB IOSNR<sub>GD</sub> = 4.9606dB



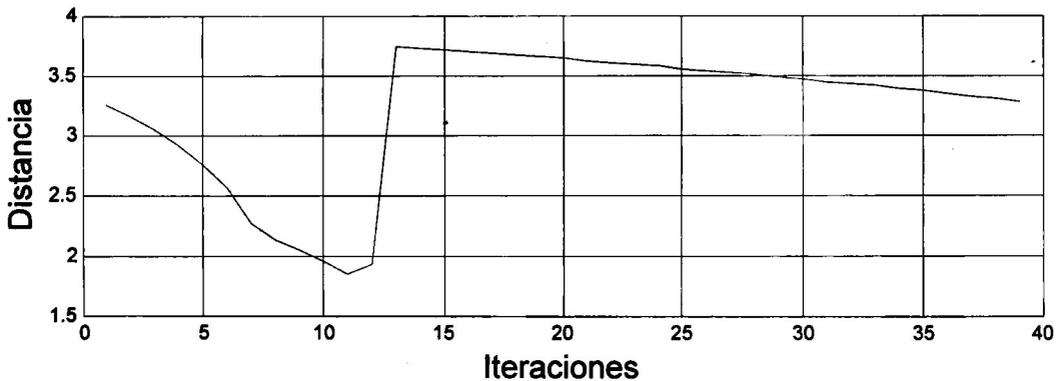
Simulación hecha con función exponencial SNR=30db y  $\Delta=65$



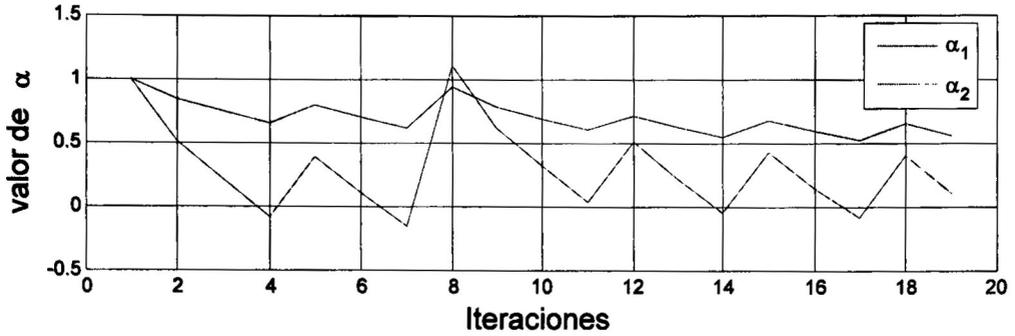
## Convergencia de los parámetros de regularización Red neuronal



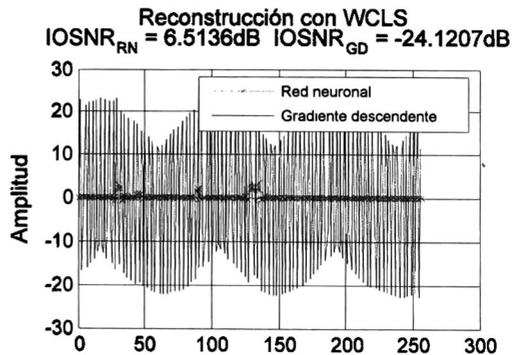
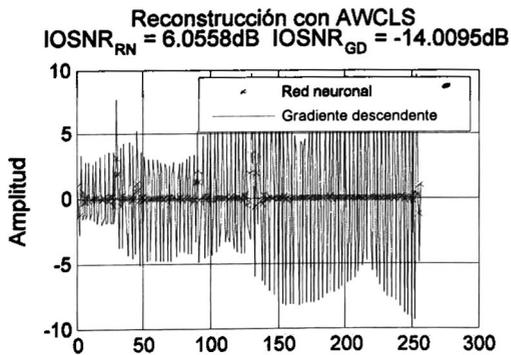
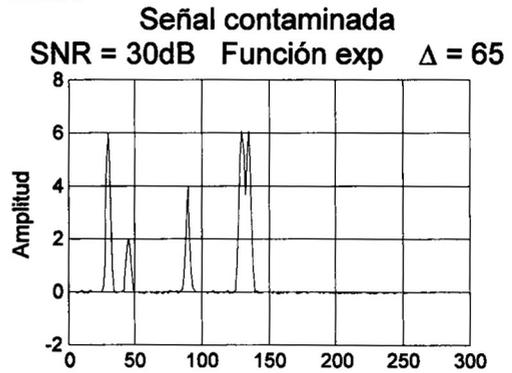
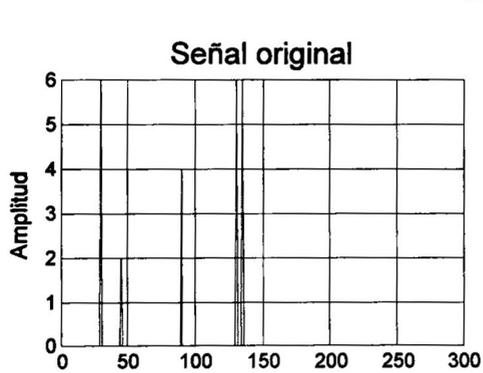
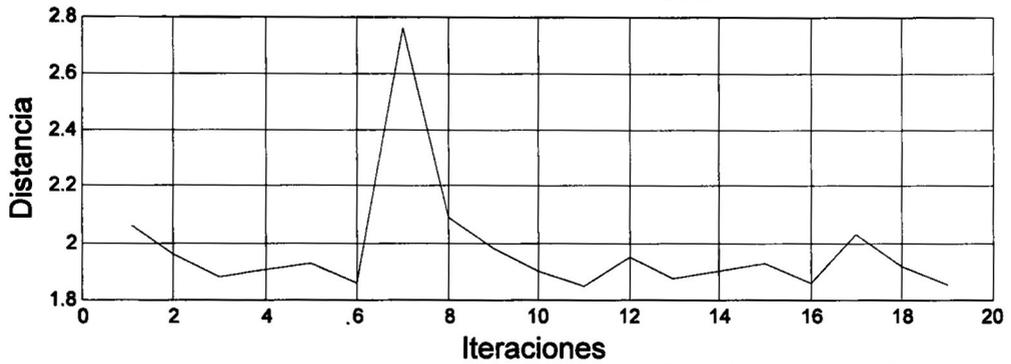
## Función de distancia



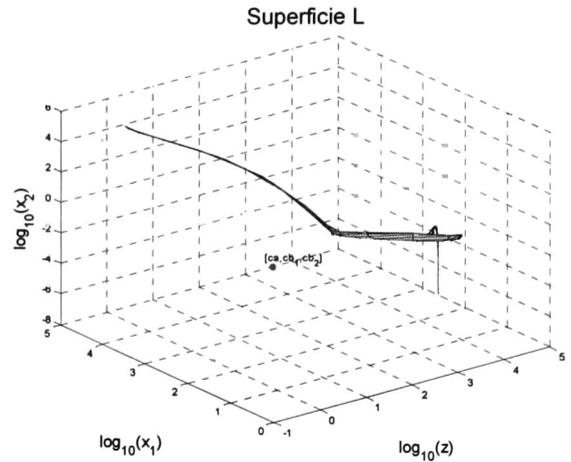
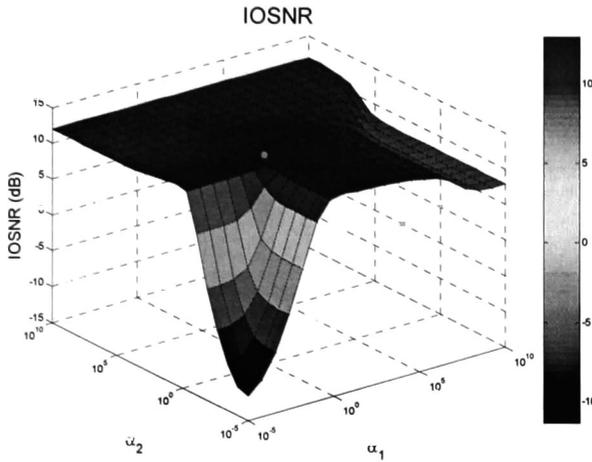
# Convergencia de los parámetros de regularización Gradiente descendente



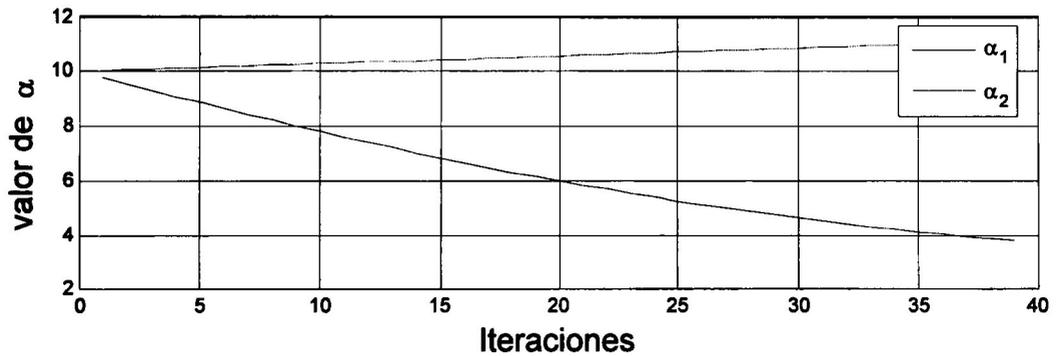
## Función de distancia



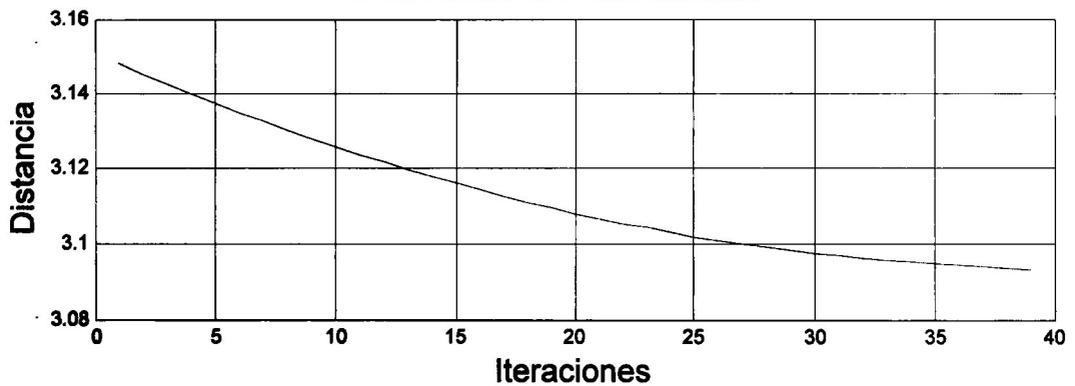
Simulación hecha con función sinc SNR=5db y  $\Delta=25$



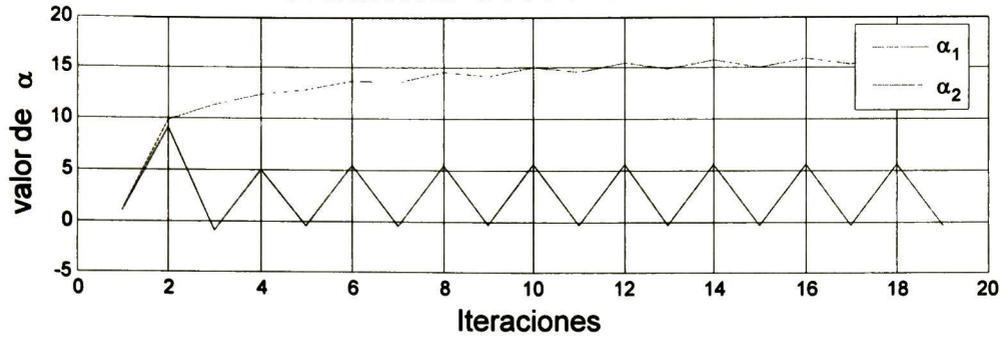
## Convergencia de los parámetros de regularización Red neuronal



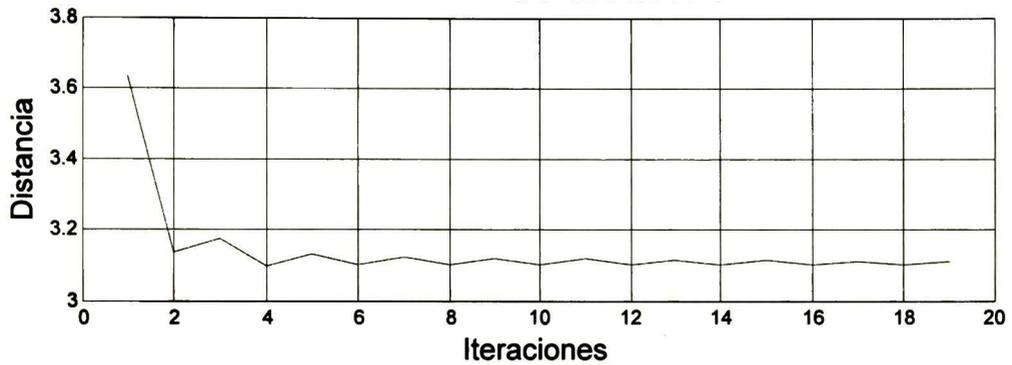
## Función de distancia



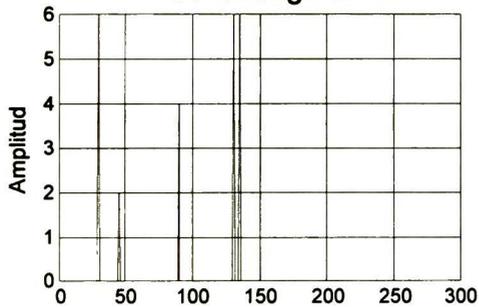
# Convergencia de los parámetros de regularización Gradiente descendente



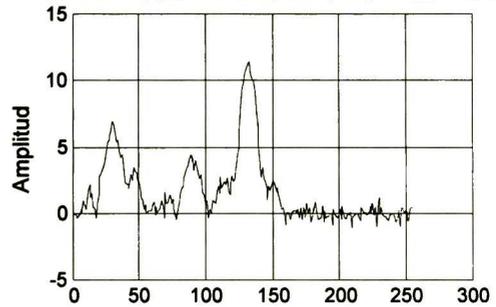
## Función de distancia



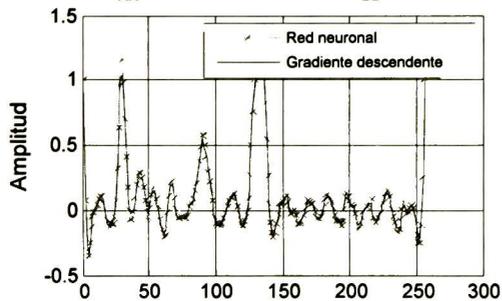
### Señal original



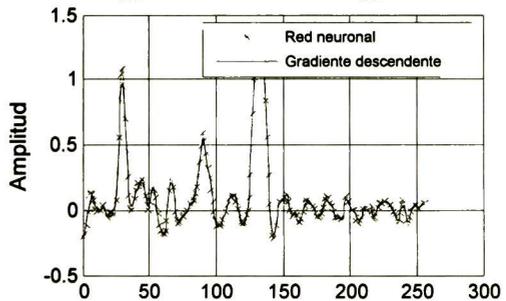
### Señal contaminada SNR = 5dB Función sinc $\Delta = 25$



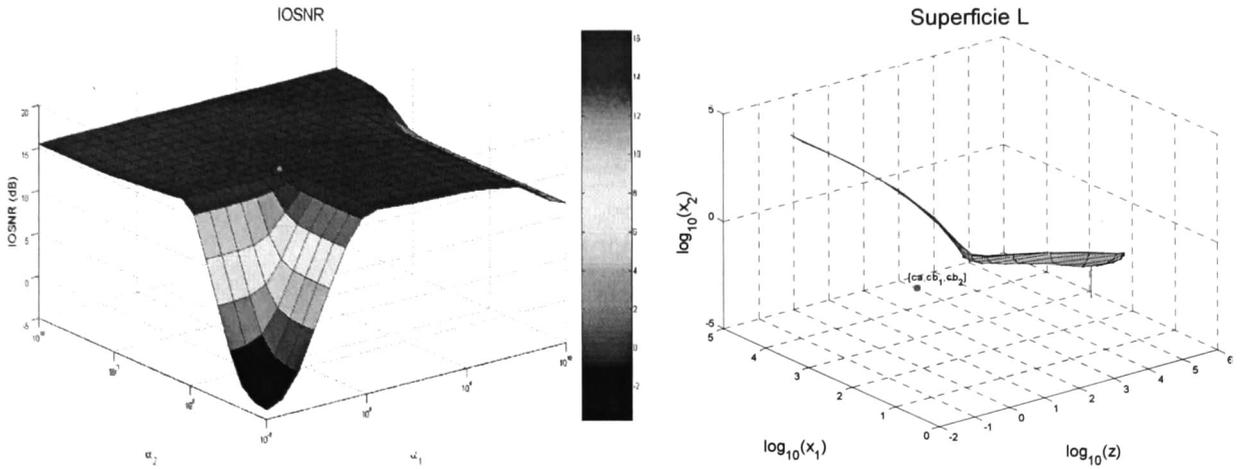
### Reconstrucción con AWCLS IOSNR<sub>RN</sub> = 12.9111dB IOSNR<sub>GD</sub> = 12.8619dB



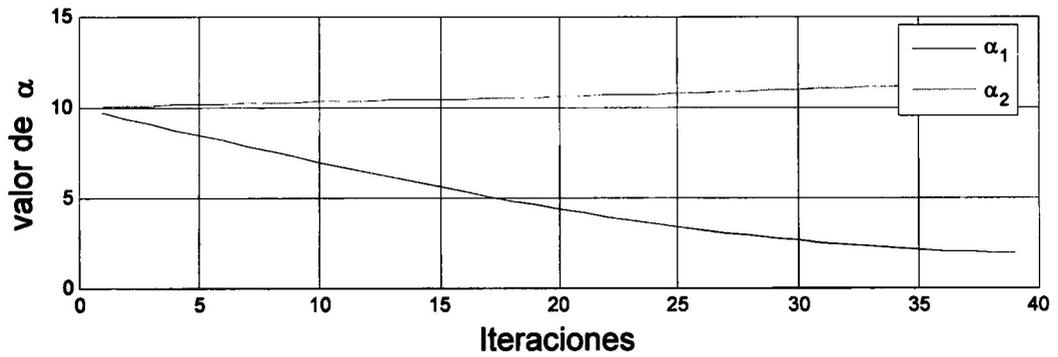
### Reconstrucción con WCLS IOSNR<sub>RN</sub> = 13.0188dB IOSNR<sub>GD</sub> = 12.9562dB



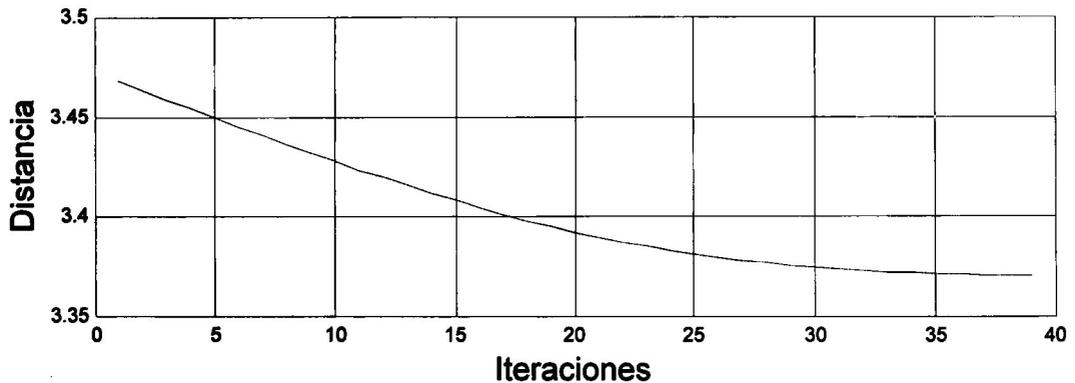
Simulación hecha con función sinc SNR=15db y  $\Delta=45$



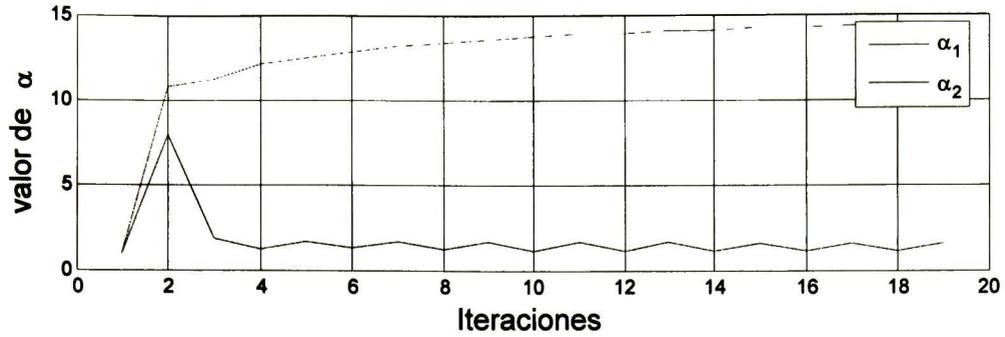
## Convergencia de los parámetros de regularización Red neuronal



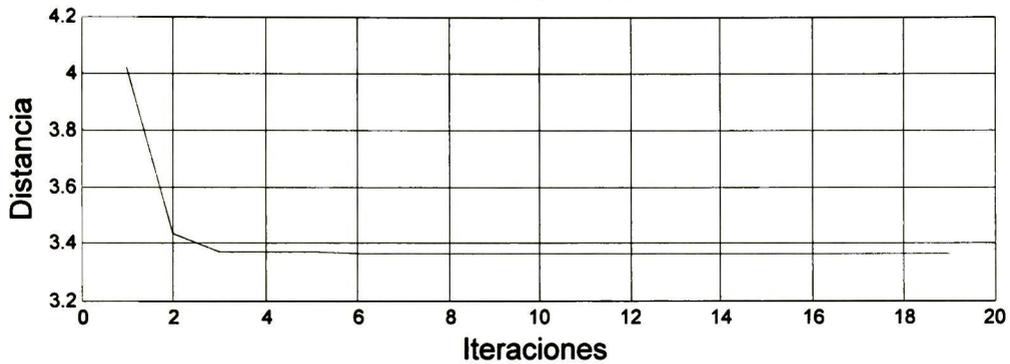
## Función de distancia



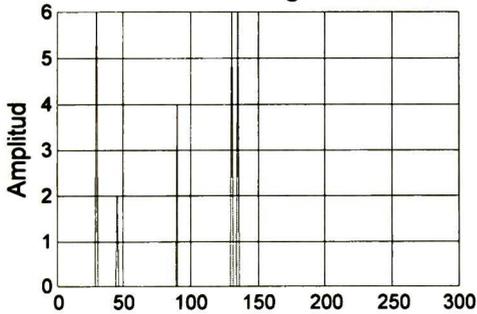
# Convergencia de los parámetros de regularización Gradiente descendente



## Función de distancia

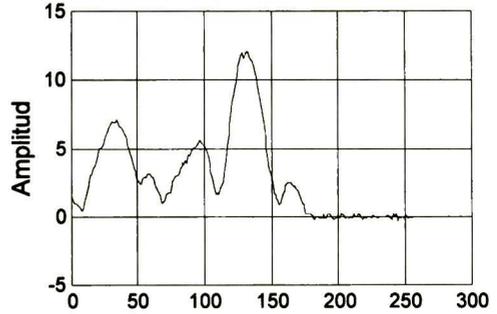


### Señal original

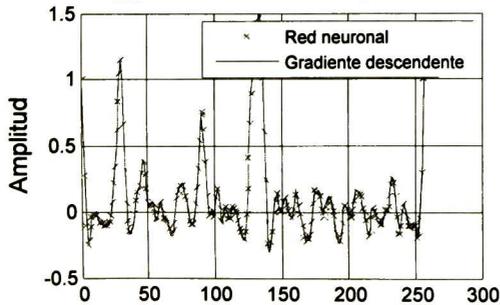


### Señal contaminada

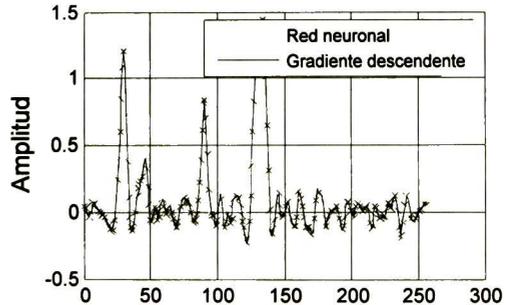
SNR = 15dB Función sinc  $\Delta = 45$



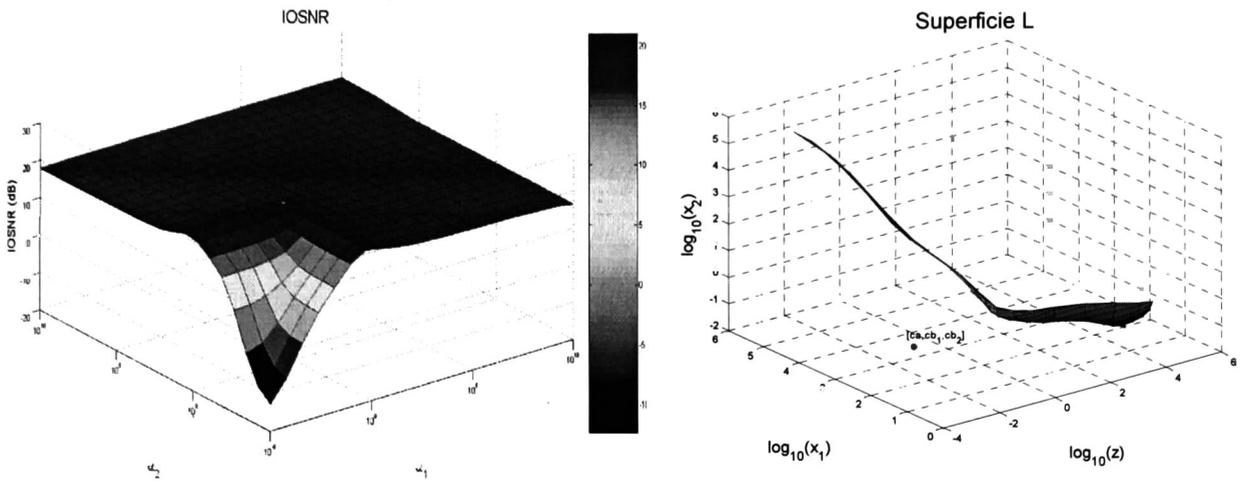
Reconstrucción con AWCLS  
IOSNR<sub>RN</sub> = 16.4145dB IOSNR<sub>GD</sub> = 16.3797dB



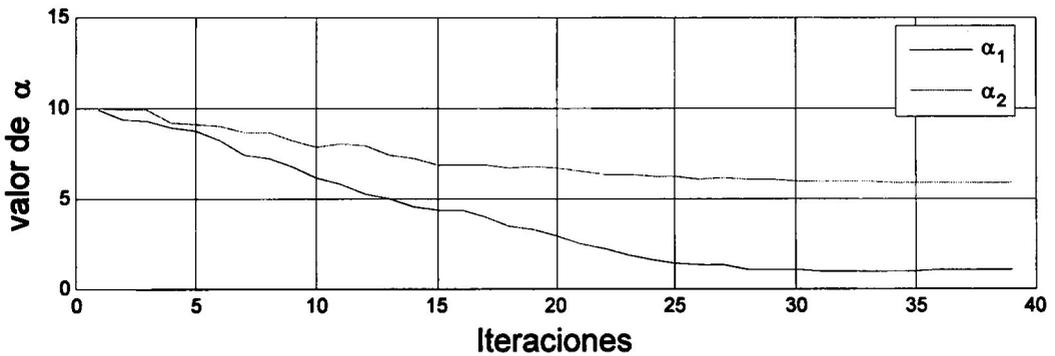
Reconstrucción con WCLS  
IOSNR<sub>RN</sub> = 16.5806dB IOSNR<sub>GD</sub> = 16.5647dB



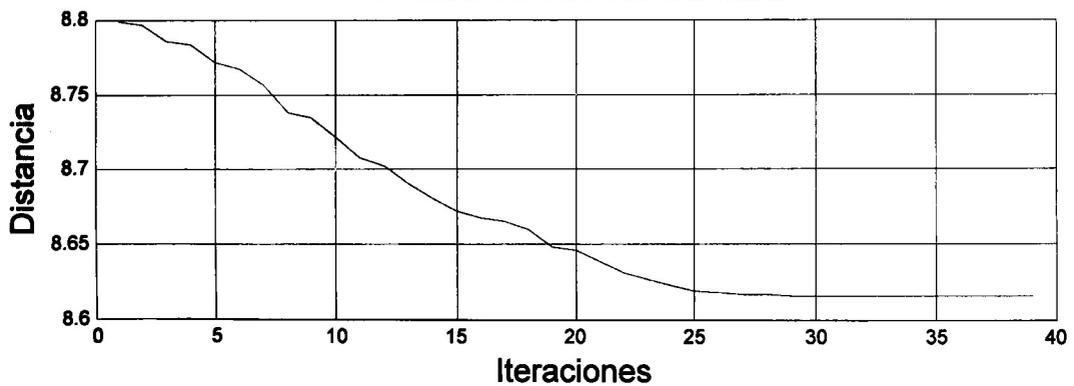
Simulación hecha con función sinc SNR=30db y  $\Delta=65$



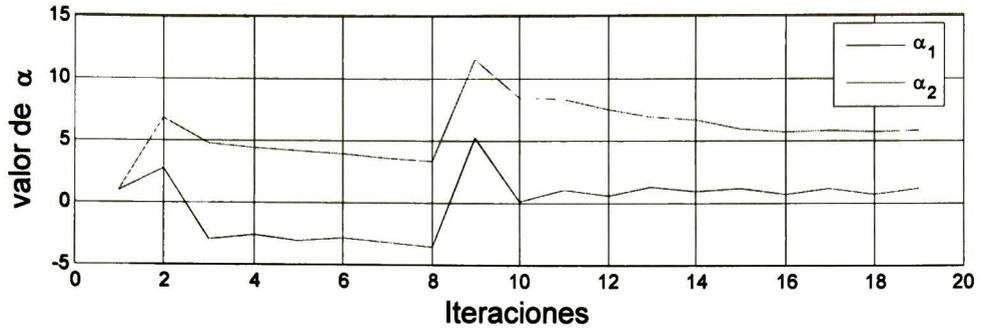
## Convergencia de los parámetros de regularización Red neuronal



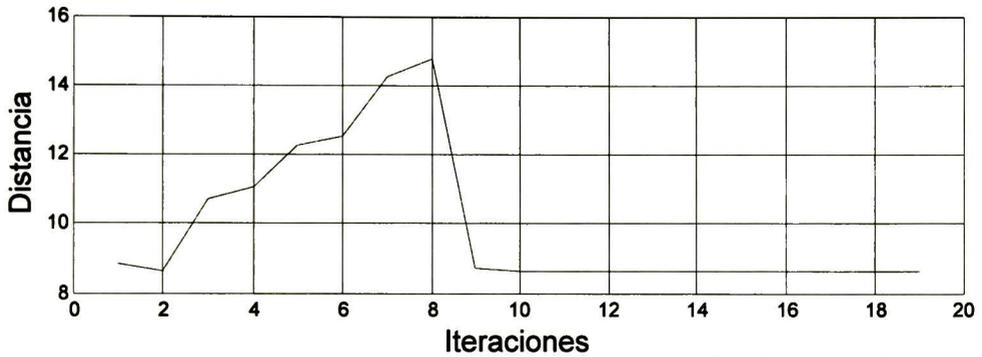
## Función de distancia



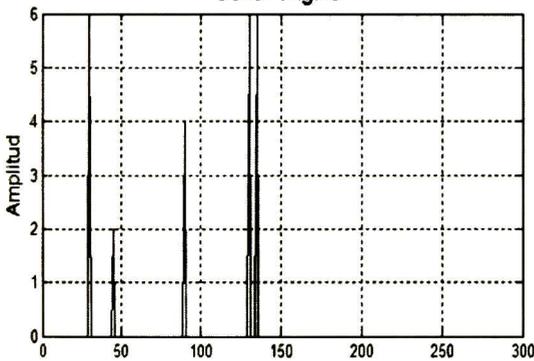
# Convergencia de los parámetros de regularización Gradiente descendente



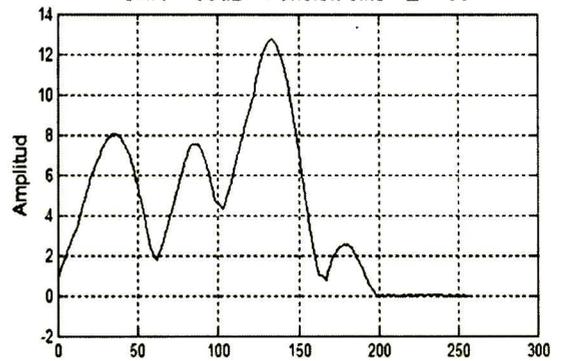
## Función de distancia



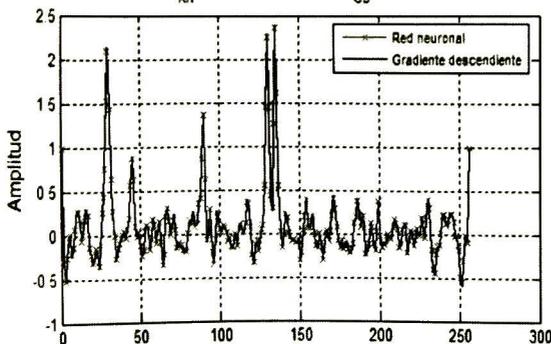
Señal original



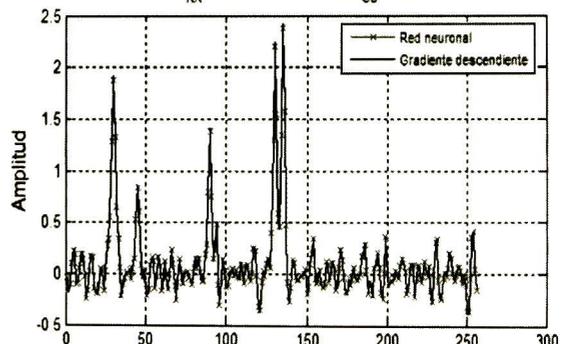
Señal contaminada  
SNR = 30dB Función sinc  $\Delta = 65$



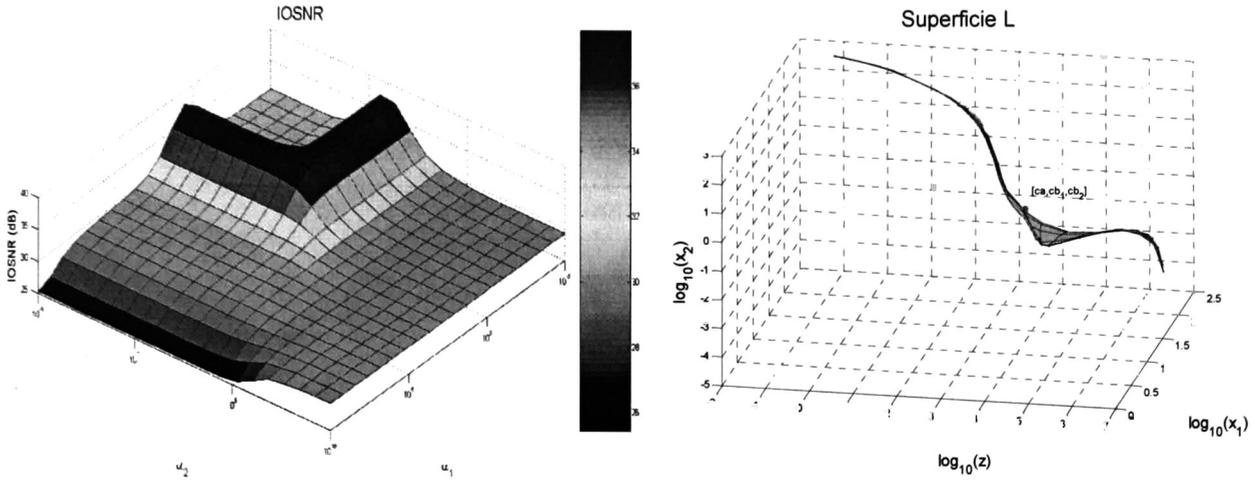
Reconstrucción con AWCLS  
 $\text{IOSNR}_{\text{RN}} = 20.0176\text{dB}$   $\text{IOSNR}_{\text{GD}} = 20.0174\text{dB}$



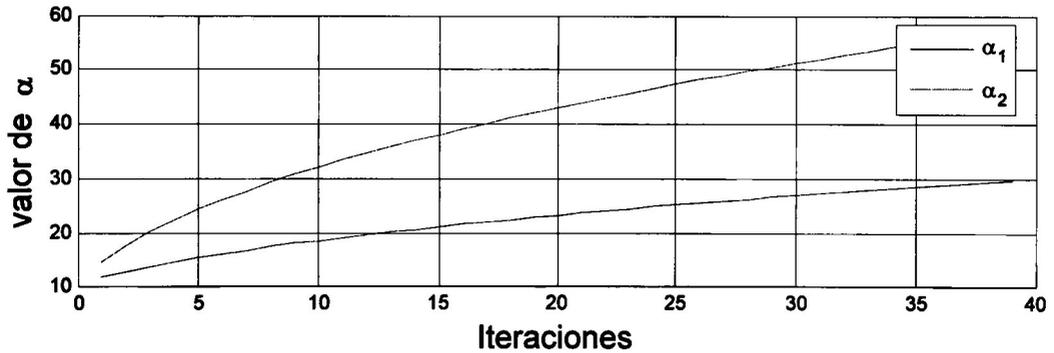
Reconstrucción con WCLS  
 $\text{IOSNR}_{\text{RN}} = 20.2624\text{dB}$   $\text{IOSNR}_{\text{GD}} = 20.2985\text{dB}$



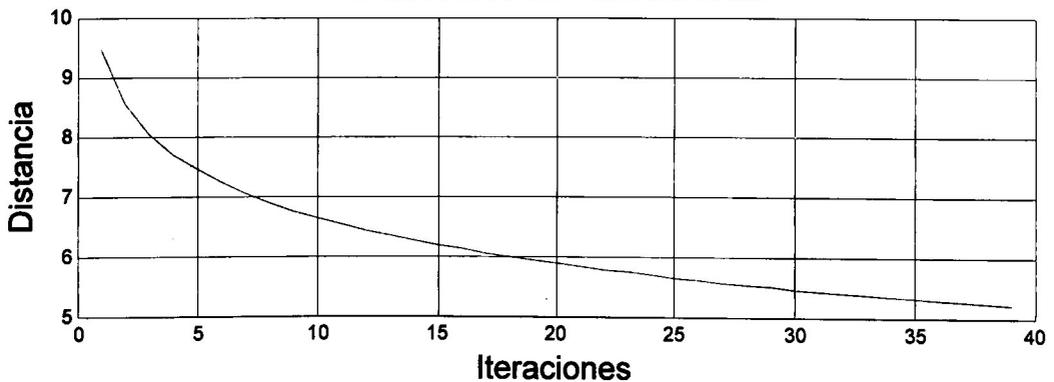
Simulación hecha con función rect SNR=5db y  $\Delta=25$



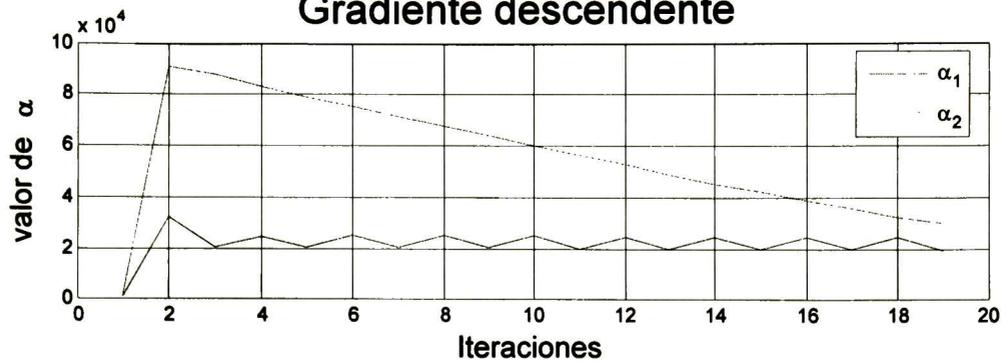
## Convergencia de los parámetros de regularización Red neuronal



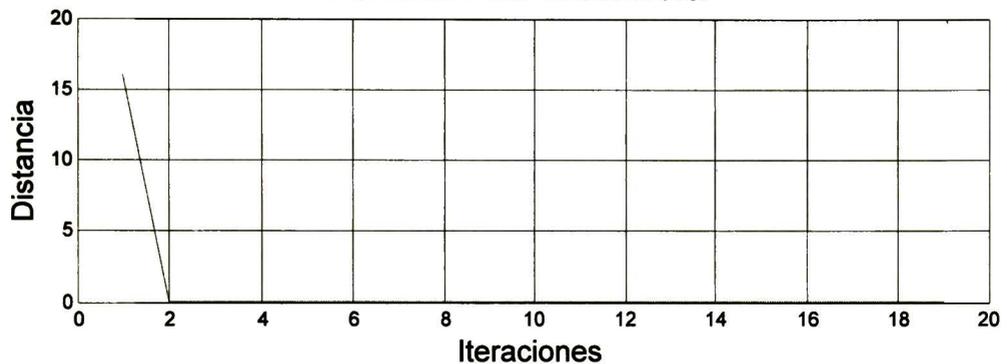
## Función de distancia



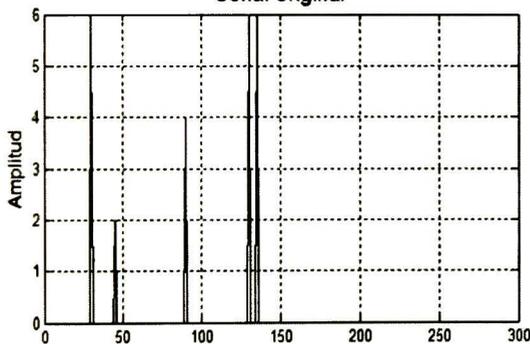
# Convergencia de los parámetros de regularización Gradiente descendente



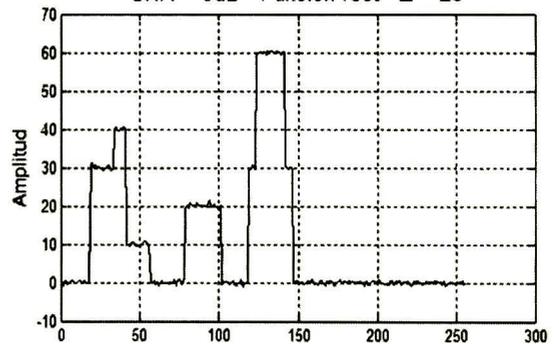
## Función de distancia



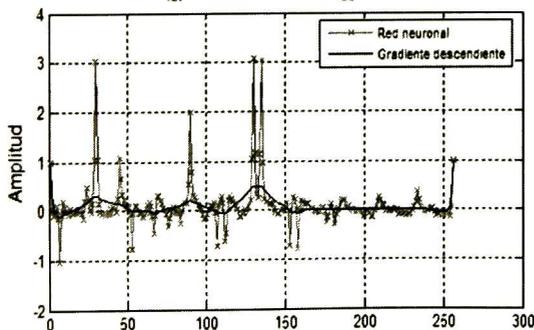
### Señal original



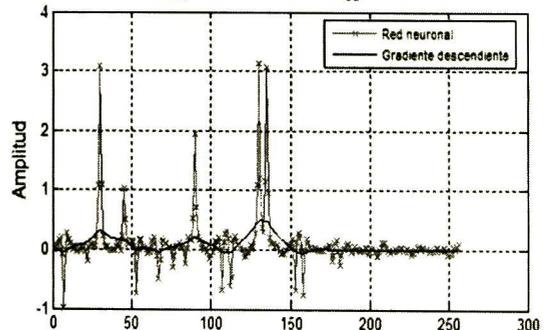
### Señal contaminada SNR = 5dB Funcion rect $\Delta = 25$



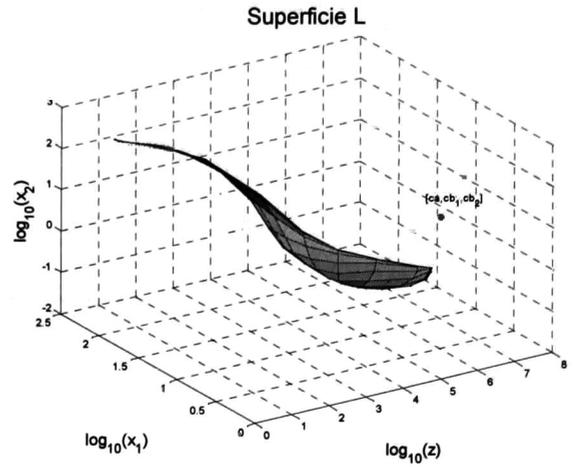
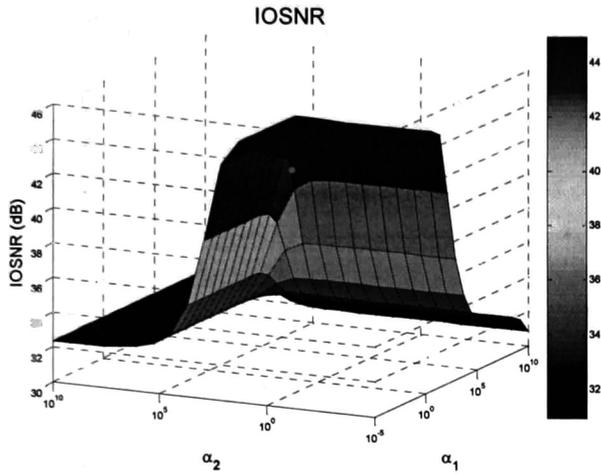
### Reconstrucción con AWCLS IOSNR<sub>RH</sub> = 33.4804dB IOSNR<sub>GD</sub> = 29.6684dB



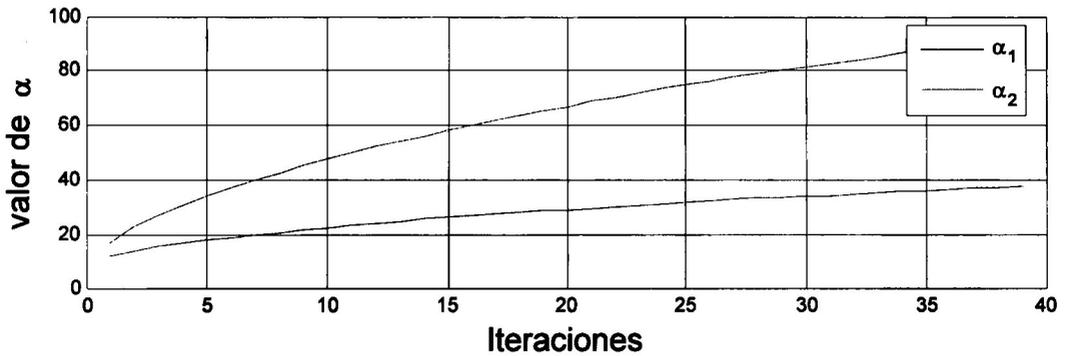
### Reconstrucción con WCLS IOSNR<sub>RH</sub> = 33.8234dB IOSNR<sub>GD</sub> = 29.7525dB



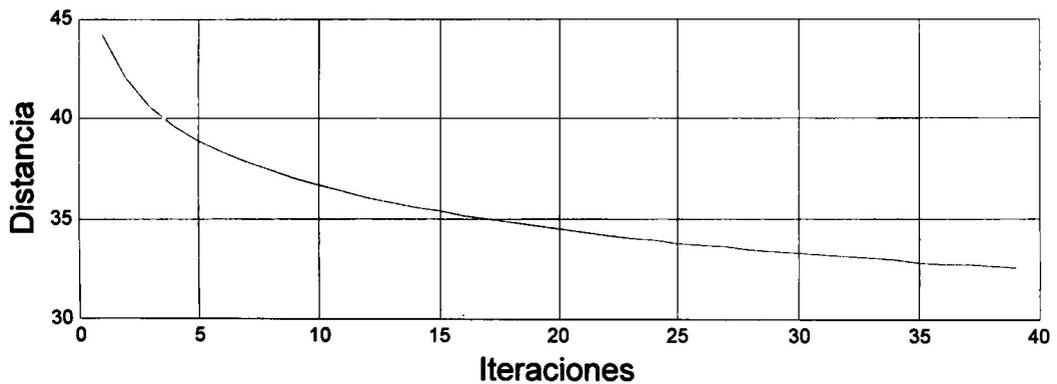
Simulación hecha con función rect SNR=15db y  $\Delta=45$



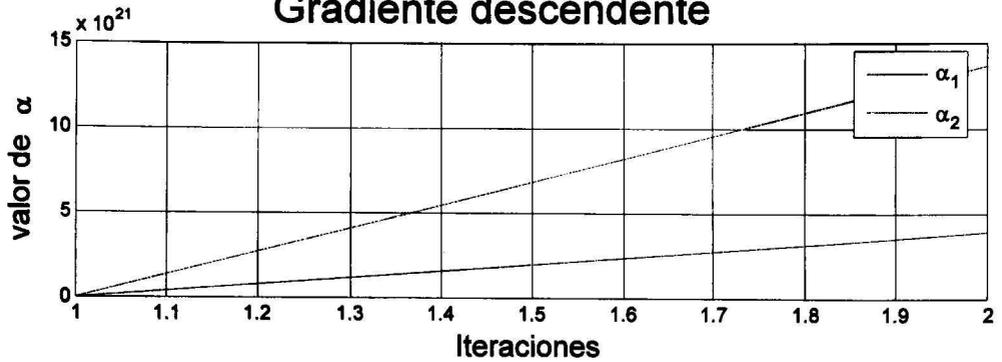
## Convergencia de los parámetros de regularización Red neuronal



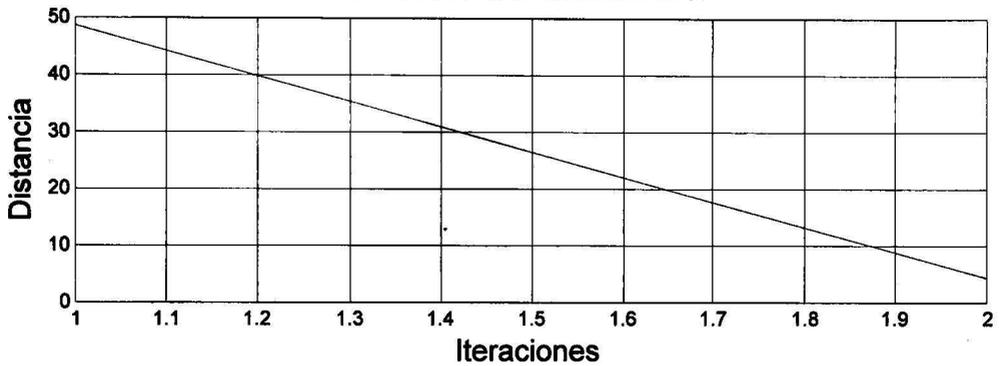
## Función de distancia



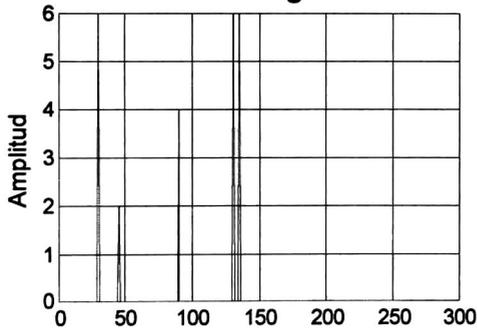
# Convergencia de los parámetros de regularización Gradiente descendente



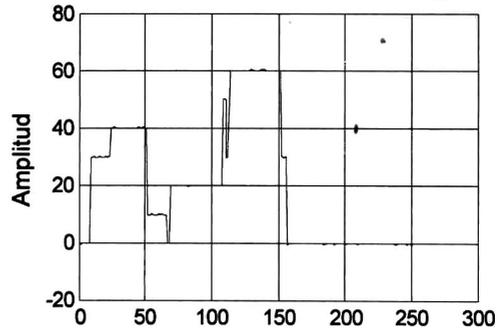
## Función de distancia



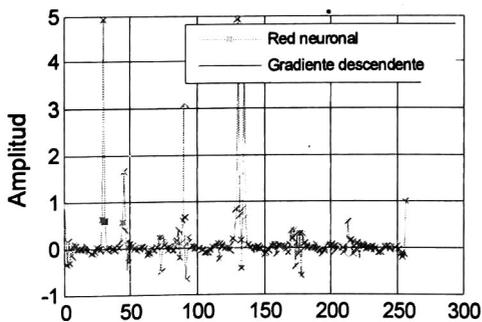
## Señal original



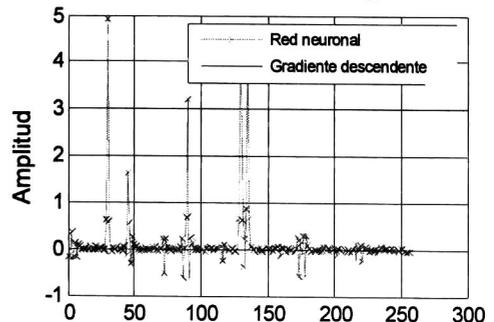
## Señal contaminada SNR = 15dB Función rect $\Delta = 45$



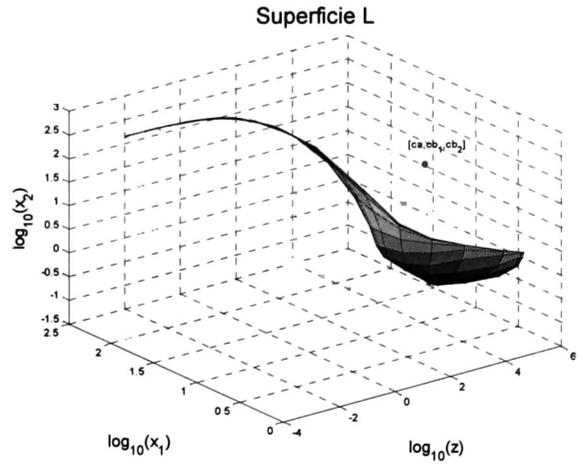
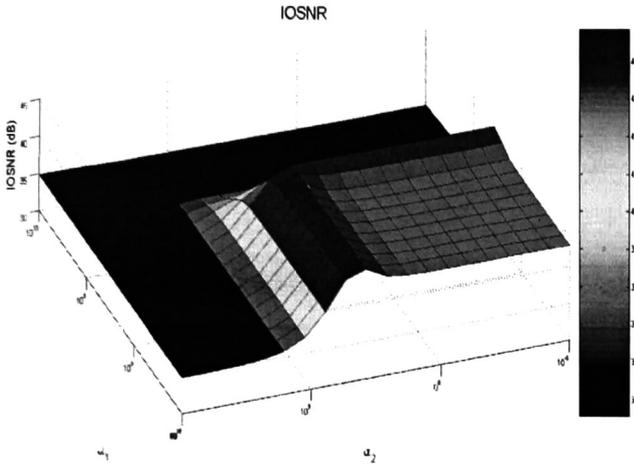
## Reconstrucción con AWCLS IOSNR<sub>RN</sub> = 41.5326dB IOSNR<sub>GD</sub> = NaN dB



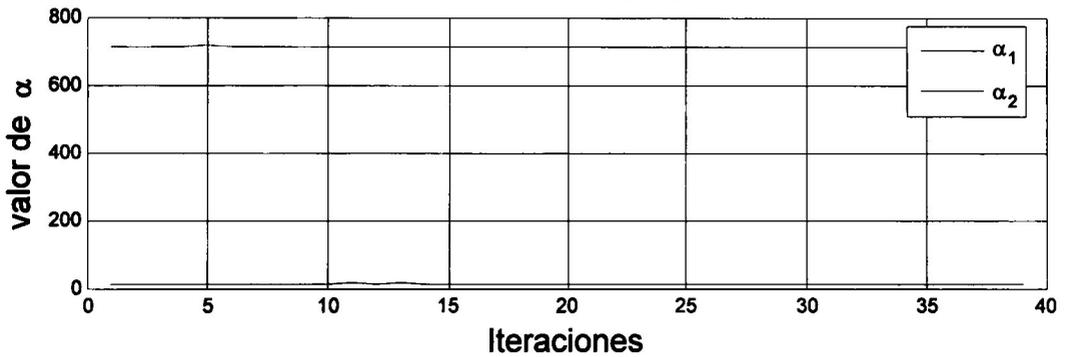
## Reconstrucción con WCLS IOSNR<sub>RN</sub> = 42.3226dB IOSNR<sub>GD</sub> = NaN dB



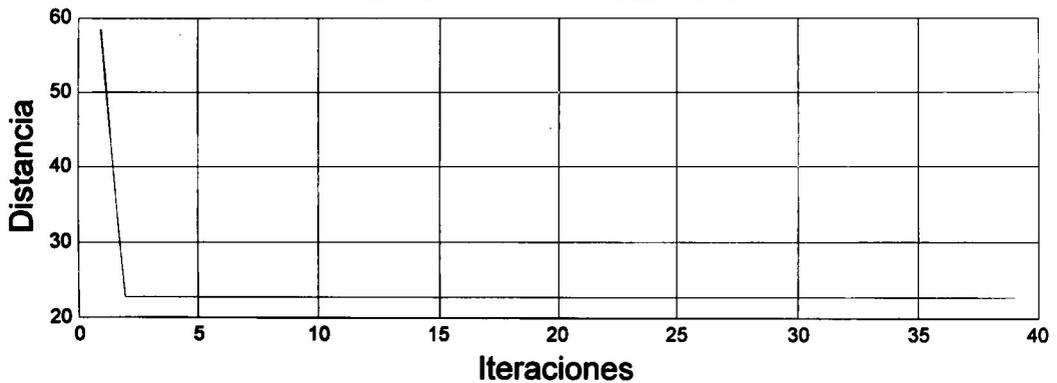
Simulación hecha con función rect SNR=30db y  $\Delta=65$



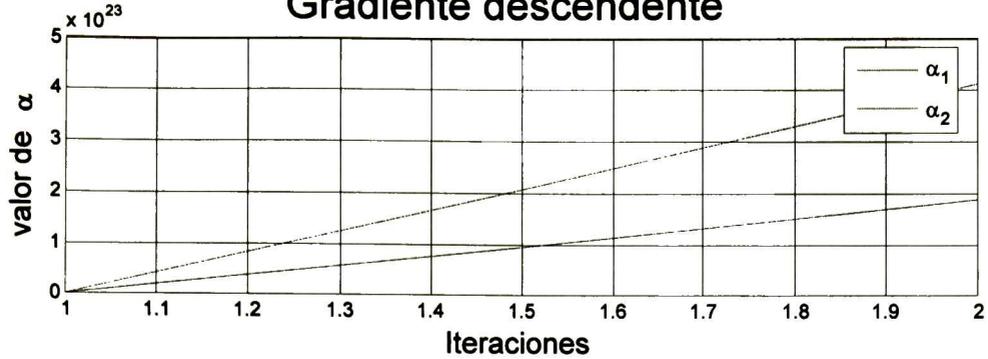
## Convergencia de los parámetros de regularización Red neuronal



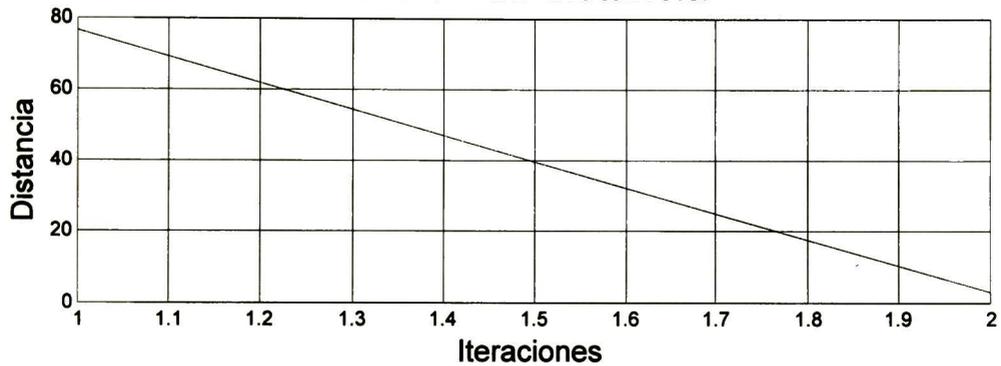
## Función de distancia



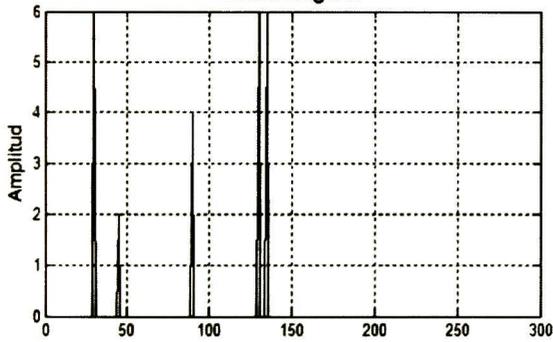
# Convergencia de los parámetros de regularización Gradiente descendente



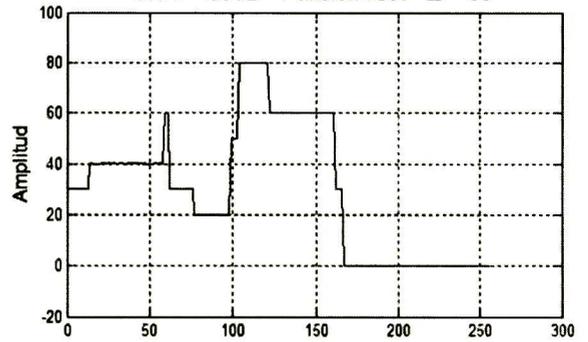
## Función de distancia



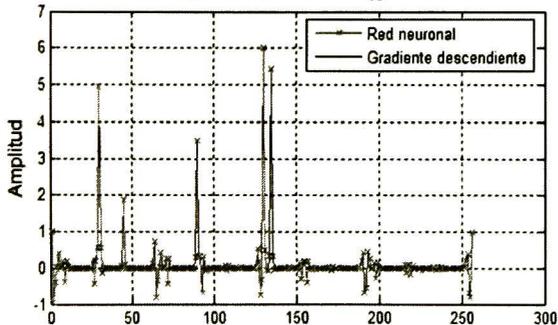
Senal original



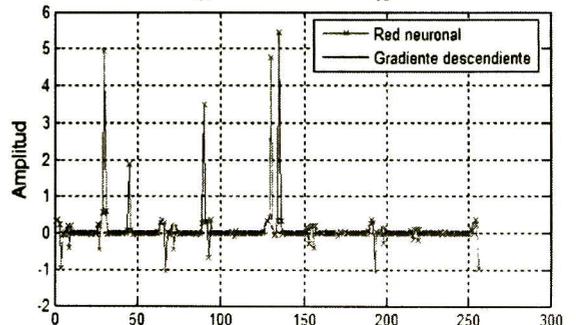
SNR = 30dB Funcion rect  $\Delta = 65$



Reconstrucción con AWCLS  
IOSNR<sub>RH</sub> = 44.6208dB IOSNR<sub>GD</sub> = NaNdB



Reconstrucción con WCLS  
IOSNR<sub>RH</sub> = 45.0319dB IOSNR<sub>GD</sub> = NaNdB

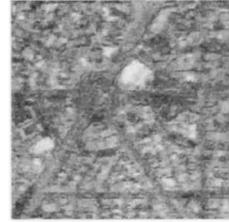


Simulación hecha con función exponencial  $SNR=5db$  y  $\Delta x=25$

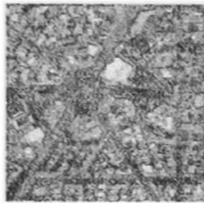
Imagen original



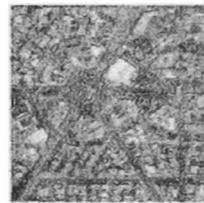
Imagen contaminada  
 $SNR = 5dB$  Funcion exp  $\Delta x = 25$



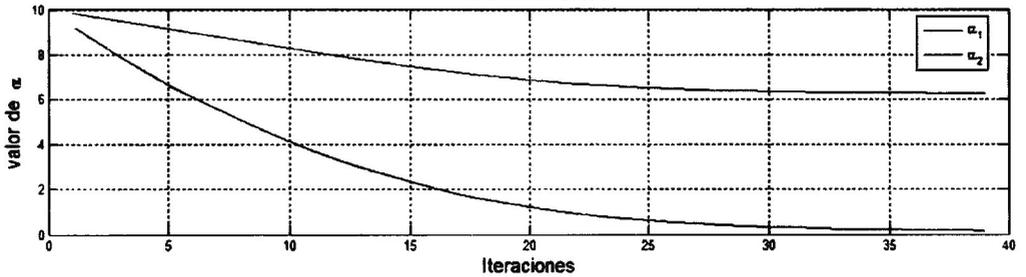
Reconstrucción con AWCLS y red neuronal  
 $IOSNR_{RN} = 22.0003dB$



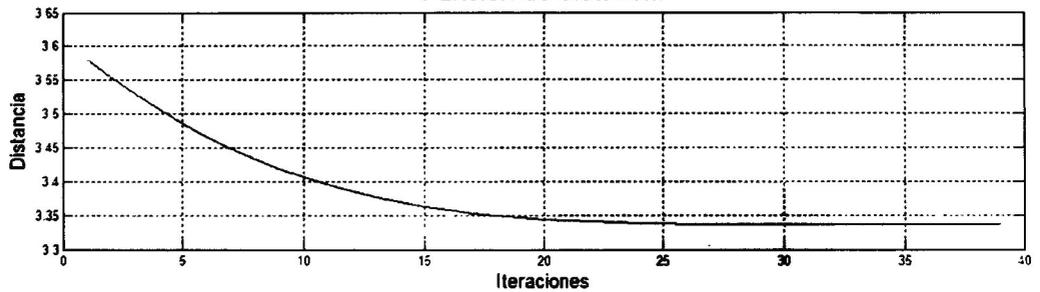
Reconstrucción con WCLS y red neuronal  
 $IOSNR_{RN} = 21.5651dB$



Convergencia de los parametros de regularizacion  
Red neuronal



Funcion de distancia

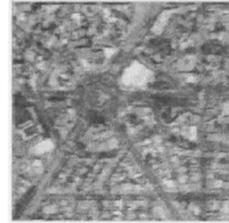


Simulación hecha con función exponencial SNR=15db y  $\Delta x=45$

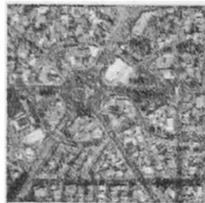
Imagen original



Imagen contaminada  
SNR = 15dB Funcion exp  $\Delta x = 45$



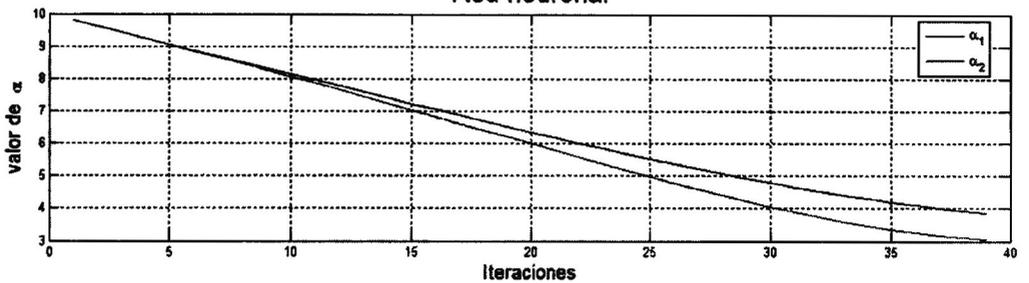
Reconstrucción con AWCLS y red neuronal  
 $IOSNR_{RN} = 23.7977\text{dB}$



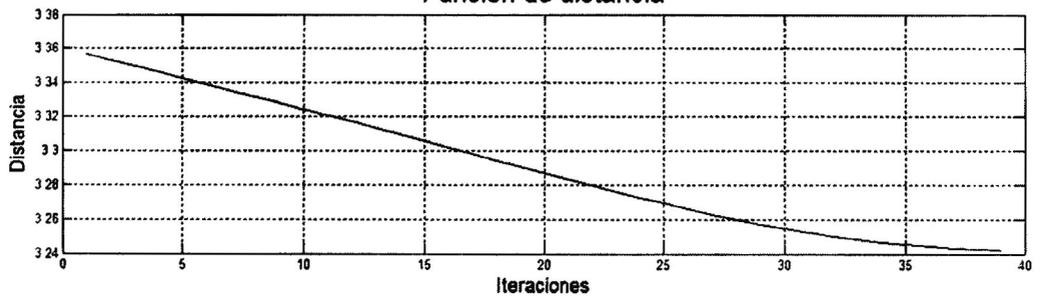
Reconstrucción con WCLS y red neuronal  
 $IOSNR_{RN} = 23.3159\text{dB}$



Convergencia de los parametros de regularizacion  
Red neuronal



Funcion de distancia



Simulación hecha con función exponencial  $SNR=30db$  y  $\Delta x=65$

Imagen original

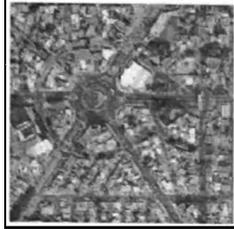
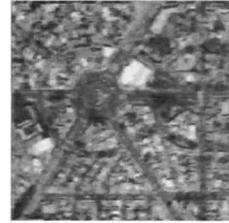
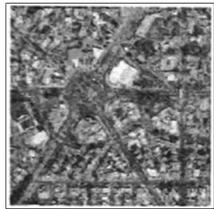


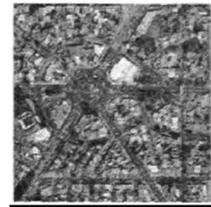
Imagen contaminada  
 $SNR = 30dB$  Funcion exp  $\Delta x = 65$



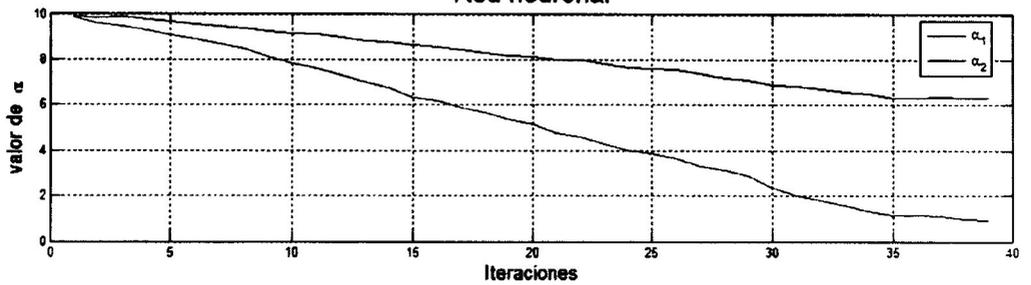
Reconstrucción con AWCLS y red neuronal  
 $IOSNR_{RN} = 24.8618dB$



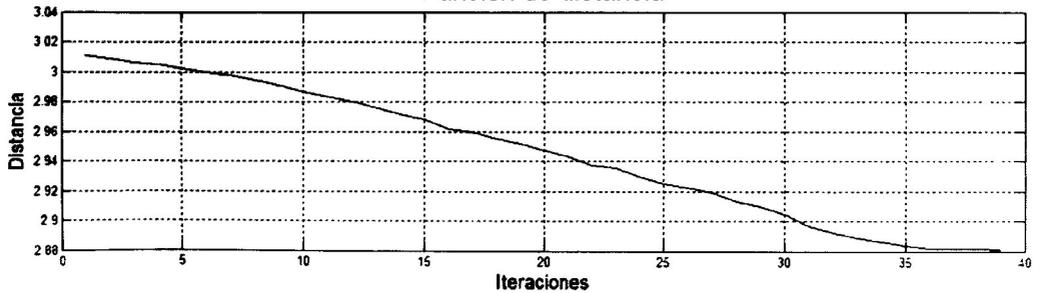
Reconstrucción con WCLS y red neuronal  
 $IOSNR_{RN} = 24.6522dB$



Convergencia de los parametros de regularizacion  
Red neuronal



Funcion de distancia

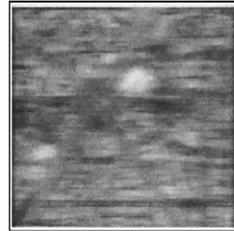


Simulación hecha con función sinc SNR=5db y  $\Delta x=25$

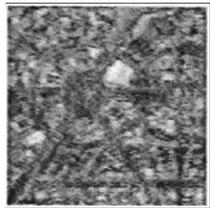
Imagen original



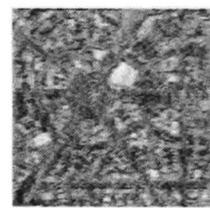
Imagen contaminada  
SNR = 5dB Funcion sinc  $\Delta x = 25$



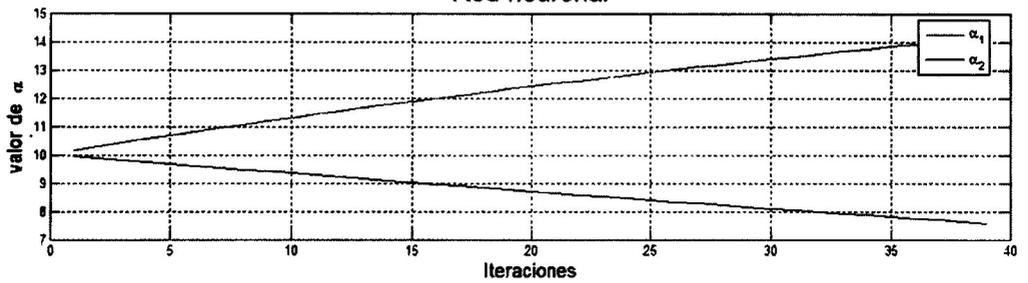
Reconstrucción con AWCLS y red neuronal  
 $\text{IOSNR}_{\text{RN}} = 37.131\text{dB}$



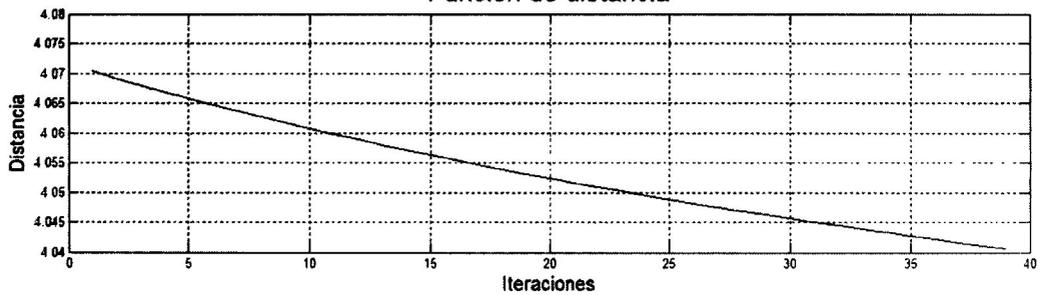
Reconstrucción con WCLS y red neuronal  
 $\text{IOSNR}_{\text{RN}} = 36.2347\text{dB}$



Convergencia de los parametros de regularizacion  
Red neuronal



Funcion de distancia

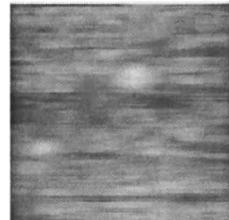


Simulación hecha con función sinc SNR=15db y  $\Delta x=45$

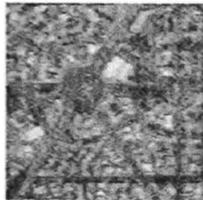
Imagen original



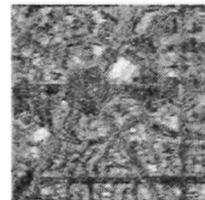
Imagen contaminada  
SNR = 15dB Funcion sinc  $\Delta x = 45$



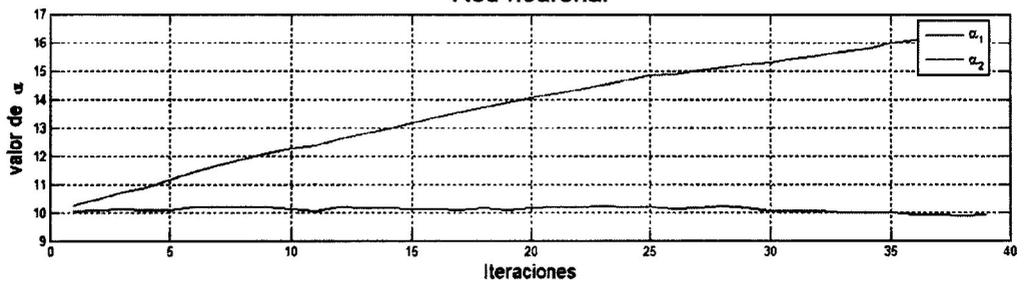
Reconstrucción con AWCLS y red neuronal  
 $\text{IOSNR}_{\text{RN}} = 42.6274\text{dB}$



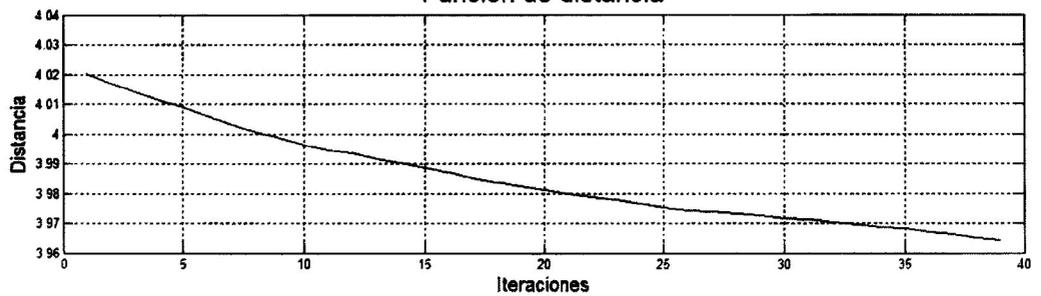
Reconstrucción con WCLS y red neuronal  
 $\text{IOSNR}_{\text{RN}} = 41.6995\text{dB}$



Convergencia de los parametros de regularizacion  
Red neuronal



Funcion de distancia

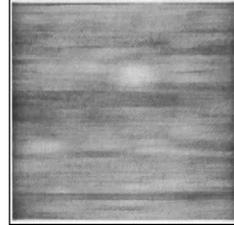


Simulación hecha con función sinc SNR=30db y  $\Delta x=65$

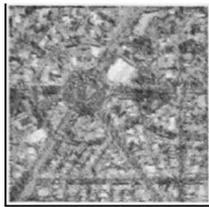
Imagen original



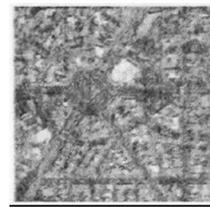
Imagen contaminada  
SNR = 30dB Función sinc  $\Delta x = 65$



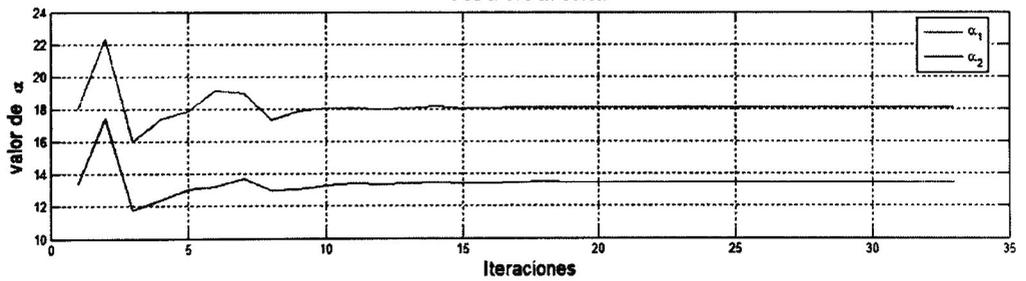
Reconstrucción con AWCLS y red neuronal  
 $IOSNR_{RN} = 47.3227\text{dB}$



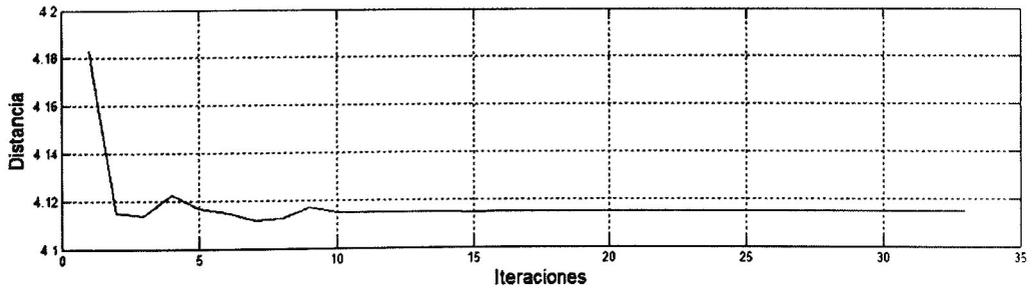
Reconstrucción con WCLS y red neuronal  
 $IOSNR_{RN} = 46.3943\text{dB}$



Convergencia de los parametros de regularizacion  
Red neuronal



Funcion de distancia



A continuación se presentan los resultados cuantitativos de las simulaciones anteriores, los cuales contienen los niveles de IOSNR obtenidos:

Tabla de IOSNR en dB para señal con restricciones lineales

			Red neuronal		Gradiente descendente	
función	SNR	$\Delta$	AWCLS	WCLS	AWCLS	WCLS
EXP	5	25	17.4406	17.2841	17.4433	17.2824
EXP	15	45	18.0284	17.8617	17.9637	17.7954
EXP	30	65	18.3735	18.3507	18.1041	18.0777
SINC	5	25	29.3332	29.2426	29.3509	29.2503
SINC	15	45	33.4161	33.4137	33.4485	33.4482
SINC	30	65	36.5033	36.3100	36.3284	36.5155
RECT	5	25	49.7714	49.5515	40.9120	40.8599
RECT	15	45	56.8012	56.7944		
RECT	30	65	59.3267	59.1784		

Tabla de IOSNR en dB para señal sin restricciones lineales

			Red neuronal		Gradiente descendente	
función	SNR	$\Delta$	AWCLS	WCLS	AWCLS	WCLS
EXP	5	25	5.3639	5.4759	5.4109	5.5263
EXP	15	45	2.9703	3.0504	4.6444	4.9606
EXP	30	65	6.0558	6.5136	-14.0095	-24.1207
SINC	5	25	12.9111	13.0188	12.8614	12.9562
SINC	15	45	16.4145	16.5806	16.3797	16.5647
SINC	30	65	20.0176	20.2624	20.0174	20.2985
RECT	5	25	33.4804	33.8234	29.6684	29.7525
RECT	15	45	41.5326	42.3226		
RECT	30	65	44.6208	45.0319		

Tabla de IOSNR en dB para imagen con restricciones lineales

			Red neuronal	
función	SNR	$\Delta$	AWCLS	WCLS
EXP	5	25	22.0003	21.5651
EXP	15	45	23.7977	23.3159
EXP	30	65	24.8618	24.6522
SINC	5	25	37.1310	36.2347
SINC	15	45	42.6274	41.6995
SINC	30	65	47.3227	46.3943
RECT	5	25	59.5695	58.5328
RECT	15	45	68.7106	68.5065
RECT	30	65	84.2530	75.9686

En casi todos los casos simulados cuando se cumple las restricciones lineales cuyos resultados se muestran en la primera tabla, se pudo observar que se tiene un IOSNR mas alto en la reconstrucción con el algoritmo AWCLS que con el algoritmo WCLS, independientemente del método numérico utilizado para encontrar los parámetros de regularización siendo mayor el IOSNR, cumpliéndose de que entre mayor sea el SNR mayor es el IOSNR esto es debido a que tenemos un pequeño porcentaje de ruido en la señal observada además se pudo observar que la mejores reconstrucciones se obtienen cuando la PSF es la función rectangular, aunque sus valores singulares de este tipo de matrices son muy pequeños haciéndolas difíciles de invertir. También se pudo observar que el algoritmo de la curva L implementado con redes neuronales es mucho mas eficiente que el implementado con el método de gradiente descendente, ya que el primero encontró los parámetros óptimos en menos iteraciones que el segundo además que para algunos casos el método de gradiente descendente fue incapaz de encontrar estos parámetros al divergir el método.

En la segunda tabla se pueden apreciar los resultados obtenidos al simular una señal que no cumple con las restricciones lineales, en este caso se observó que el método WCLS proporciona mejores reconstrucciones al tenerse valores de IOSNR mas altos en comparación que con el método AWCLS, esto es debido a que el método AWCLS trata de poner dichas restricciones en la señal estimada lo cual hace que el valor de IOSNR disminuya. En estas simulaciones también se cumplió que el método implementado con redes neuronales es más eficiente ya que encuentra los valores óptimos en menos iteraciones, además de que en ciertos casos (función rectangular) el método de gradiente descendente fue incapaz de encontrar los valores óptimos.

En la ultima tabla se presentan los resultados cuantitativos de las simulaciones realizadas con imágenes las cuales cumplían con las restricciones lineales impuestas, en este caso se pudo observar que el método AWCLS fue mejor que el método WCLS al obtenerse mejores valores de IOSNR, además se pudo observar que las mejores reconstrucciones se obtienen cuando se tiene una PSF rectangular o sinc, y como en los casos anteriores se pudo ver que el método con redes neuronales fue más eficiente que el método implementado con el gradiente descendente.

# Conclusiones

Las conclusiones que se obtuvieron durante el desarrollo de este trabajo de tesis pueden ser catalogadas en dos categorías, la primera son acerca del desempeño del método propuesto en comparación con métodos de regularización existentes en este caso con el método WCLS, en varios escenarios de señales observadas; la segunda categoría es acerca de la hipersuperficie L como método para encontrar los parámetros de regularización óptimos y su desempeño al ser implementado de manera iterativa con una red neuronal y con el método de gradiente descendente.

Podemos decir que el método estudiado (AWCLS) presenta un mejor desempeño al ofrecer un IOSNR más alto en comparación con el método WCLS, cuando se presenta una señal contaminada que cumple con las restricciones lineales que propusimos como información a priori durante el desarrollo del método AWCLS, esto es debido al lazo de retroalimentación que se encuentra en el esquema computacional del método AWCLS el cual mejora la reconstrucción. En caso de que la señal contaminada no presenta la restricción lineal impuesta, el método AWCLS intenta poner en la señal reconstruida las restricciones lineales lo cual hace de que el IOSNR sea menor que el obtenido en una reconstrucción con el método WCLS, por lo cual podemos dar como primera conclusión de que el método estudiado da mejores resultados cuando la señal observada cumple con las restricciones lineales, en algunos casos hasta 2dB por encima del IOSNR obtenido con el método WCLS; por lo tanto dada la señal a reconstruir, si conocemos cierta información la cual podemos expresar como restricciones lineales entonces el método AWCLS es una mejor opción.

Las conclusiones que se obtuvieron acerca de la evaluación del método de la hipersuperficie L con diferentes métodos numéricos son los siguientes:

Se pudo observar que la curva L es un método de fácil implementación, el cual permite que se modifique fácilmente para poder encontrar más de un parámetro de regularización, en comparación con otros métodos los cuales son diseñados para encontrar un solo parámetro.

Se pudo observar que se requiere menos gasto computacional para implementar un método iterativo para encontrar la esquina de la curva L, que el evaluar varios puntos de la hipersuperficie y tratar de

encontrar el punto con mayor curvatura Gaussiana, ya que las fórmulas de curvatura implican cálculo de varios gradientes y segundas derivadas las cuales a veces son difíciles de calcular.

Se observó también que las reconstrucciones obtenidas con los parámetros de regularización calculados a partir de la curva L ofrecen valores de IOSNR cercanos a los máximos posibles.

Otras características que se observaron es de que tanto la implementación con redes neuronales como con el método de gradiente descendente se obtiene el mínimo de la función de distancia mas o menos en el mismo número de iteración, con la diferencia de que el método de gradiente descendente lleva acabo mas operaciones dentro de cada ciclo que el método con redes neuronales, además de que tiende a oscilar el resultado que entrega, en caso contrario la implementación con red neuronal es más sencilla, se requieren de menos operaciones dentro de cada ciclo y el resultado que entrega es más estable, la única desventaja es de que depende de un parámetro de relajación para controlar la convergencia del método, pero en general podemos decir que la implementación con red neuronal es mejor que la de gradiente descendente.

Podemos decir que el método estudiado tiene un buen desempeño ya que entrega buenas reconstrucciones, su uso puede extenderse para cualquier señal o imagen que contenga información que pueda representar como una ecuación lineal. Se puede recomendar como trabajo futuro la implementación del método AWCLS en una red neuronal como una alternativa para mejorar el desempeño y posiblemente fusionar este algoritmo junto con el algoritmo de la curva L para la estimación de los parámetros óptimos.

Por último las aportaciones que se realizaron durante la elaboración de este trabajo fueron las de implementar el método de búsqueda de la esquina de la curva L mediante una red neuronal de manera iterativa, ya que en otros trabajos relacionados encuentran este punto mediante la obtención del gradiente, lo cual en muchas ocasiones es muy difícil de obtener.

# Bibliografía

- [1] Belge, Murat. *Multiscale and curvatura methods for the regularization of linear inverse problems*. Ph.d. tesis, Universidad de Northeastern, 1999.
- [2] Shkvarko, Yuriy. *Apuntes de procesamiento digital de señales*. CINVESTAV, Mexico, 2004.
- [3] Jain, Anil K. *Fundamentals of digital image processing*. Prentice Hall, Englewood Cliffs, NJ, 1989.
- [4] Gonzales, Rafael C. Woods, Richard E. *Digital image processing*. Prentice Hall, Upper Saddle River, NJ,
- [5] Jensen, John R., *Introductory digital image processing*. Prentice Hall, Upper Saddle River, NJ, 2005.
- [6] Mathews, John H., *Numerical methods for mathematics, science, and engineering*. Prentice Hall, Englewoos Cliffs, NJ, 1992.
- [7] Apostol, Tom M., *Calculus*. John Wiley, New York, 1969.
- [8] Principe, Jose C., *Neural and adaptive systems*. John Wiley and Sons, New York, 2000.
- [9] Golden, Richard M. *Mathematical methods for neural networks analysis and design*. MIT press, Cambridge, MA, 1996.
- [10] Haykin, Simon S. *Neural networks: a comprehensive foundation*. McMillan Publishing Co., Englewood cliffs, NJ, 1994.
- [11] Benediktsson, Jon A. *Classification and feature extraction for remote sensing images from urban areas based on morphological transformations*. IEEE Trans. Geoscience and remote sensing, vol 41, No. 9: 1940-1947, Septiembre 2003
- [12] Quartulli, M. *Information fusion for scene understanding from interferometric SAR data in urban environments*. IEEE Trans. Geoscience and remote sensing, vol 41, No. 9: 1976-1985, Septiembre 2003
- [13] Greenhill, Rarrel R. *Characterization of suburban areas for land use planning using landscape ecological indicators derived from Ikonos-2 multespectral imagery*. IEEE Trans. Geoscience and remote sensing, vol 41, No. 9: 2015-2021, Septiembre 2003
- [14] Bai, Zhaojun. *The CSD, GSVD, their applications and computations*. 1992.
- [15] De Moor, Bart. *Generalizations of the singular value and QR descomposition*

- [16] Kilmer, Misha E., *Choosing regularization parameters in iterative methods for ill-posed problems.*
- [17] Hansen, P.C., *The L-curve and its use in the numerical treatment of inverse problems.*
- [18] Golub, Gene H., *Generalized cross-validation for large scale problems.*
- [19] Engl, Heinz W., *Using the L-curve for determining optimal regularization parameters.* 1994.
- [20] Calvetti, D., *L-curve and curvature bounds for Tikhonov regularization.*
- [21] Liang, Xue-Bin. *A recurrent neural network for nonlinear optimization with a continuously differentiable objective function and bound constraints.*
- [22] Xia, Youshen. *A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints.*
- [23] Mathews, John H., *Numerical methods using Matlab.* Prentice Hall, Upper Saddle River, NJ, 2004.

# Apéndice A

## Código de MATLAB

%función para evaluar y graficar el desempeño de los algoritmos utilizados

```
snr=30; %nivel de ruido
func='rect'; %tipo de funcion
K=65; %ancho de la PSF
[a,ac,mv,d,N0,q,qS,S,Mu,Mv1,Mv2,I,C,b,ca,cb]=formaciondata(snr,func,K);

% [imarecons,wcls,imarecons_1wcls,ac,a,iosnr,ztt,xt1,xt2]=pruebametodo(a,ac,mv,d,N0,q,qS,S,Mu,Mv1,Mv2,I,C,b,ca,cb);
% bet=logspace(-5,10,20);
% figiosnr=figure();surf(bet,bet,iosnr);set(gca,'XScale','log','YScale','log');
% title('IOSNR','fontsize',20),zlabel('IOSNR (dB)','fontsize',15),xlabel('\alpha_1','fontsize',15),ylabel('\alpha_2','fontsize',15),colorbar();
% figsurf=figure();surf(log10(ztt),log10(xt1),log10(xt2));
% title('Superficie L','fontsize',20),ylabel('log_{10}(x_1)','fontsize',15),zlabel('log_{10}(x_2)','fontsize',15),xlabel('log_{10}(z)','fontsize',15),
% hold on,plot3(ca,cb(1),cb(2),'go','markerfacecolor','g');
% text(ca,cb(1)+.2,cb(2)+.2,['ca,cb_1,cb_2']);

%con red neuronal
[iosnr,iosnr_1,ztg,xtg1,xtg2,Estimate,Estimate_1,iter,parametrog,funparg,alef,a,ac]=metodoiter2m(a,ac,mv,d,N0,q,qS,S,Mu,Mv1,Mv2,I,C,b,ca,cb);

%con gradiente descendente
%[iosnrdes,iosnr_1des,Estimatedes,Estimate_1des,iterdes,ztt1des,xtt1des,xtt2des,parametroguarddes,funpargdes]=metodoiter(a,ac,mv,d,N0,q,qS,S,Mu,Mv1,Mv2,I,C,b,ca,cb);

%para imagenes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,2,1),imshow(a),title('Imagen original','fontsize',20)
beto=sum(sum(a));
subplot(2,2,2),imshow(ac*beto/sum(sum(ac))),title({'Imagen contaminada','[SNR = ',num2str(snr),'dB Funcion ',func,' \Deltax = ',num2str(K)]','fontsize',20)
subplot(2,2,3),imshow(Estimate*beto/sum(sum(Estimate)));title({'Reconstruccion con AWCLS y red neuronal','[IOSNR_{RN} = ',num2str(iosnr),'dB']','fontsize',20);
subplot(2,2,4),imshow(Estimate_1*beto/sum(sum(Estimate_1)));title({'Reconstruccion con WCLS y red neuronal','[IOSNR_{RN} = ',num2str(iosnr_1),'dB']','fontsize',20);
%subplot(3,2,5),imshow(Estimate*beto/sum(sum(Estimatedes)));title({'Reconstruccion con AWCLS y gradiente descendente','[IOSNR_{GD} = ',num2str(iosnrdes),'dB']','fontsize',20);
%subplot(3,2,6),imshow(Estimate_1*beto/sum(sum(Estimate_1des)));title({'Reconstruccion con WCLS y gradiente descendente','[IOSNR_{GD} = ',num2str(iosnr_1des),'dB']','fontsize',20);

figure,
subplot(2,1,1)
plot(parametrog);xlabel('Iteraciones','fontsize',15);ylabel('valor de \bf \alpha','fontsize',15),grid on,
title({'Convergencia de los parametros de regularizacion', 'Red neuronal'},'fontsize',20)
legend('\alpha_1','\alpha_2');
subplot(2,1,2)
plot(funparg);title('Funcion de distancia','fontsize',20);xlabel('Iteraciones','fontsize',15);ylabel('Distancia','fontsize',15),grid on;
% subplot(4,1,3)
% plot(parametroguarddes);xlabel('Iteraciones','fontsize',15);ylabel('valor de \bf \alpha','fontsize',15),grid on,
% title({'Convergencia de los parametros de regularizacion', 'Gradiente descendente'},'fontsize',20)
% legend('\alpha_1','\alpha_2');
% subplot(4,1,4)
% plot(funpargdes);title('Funcion de distancia','fontsize',20);xlabel('Iteraciones','fontsize',15);ylabel('Distancia','fontsize',15),grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
iosnr,
iosnr_1,
% iosnrdes,
% iosnr_1des,
```

```

% quitar comentarios si se desea evaluar se\u00f1ales
%
% figura=figure();subplot(2,1,1);plot(parametrog);xlabel('Iteraciones','fontsize',15);ylabel('valor de \alpha','fontsize',15),grid on,
% title('Convergencia de los parametros de regularizacion','Red neuronal'),'fontsize',20)
% legend('\alpha_1','\alpha_2');
% subplot(2,1,2);plot(funparg);title('Funcion de distancia','fontsize',20);xlabel('Iteraciones','fontsize',15);ylabel('Distancia','fontsize',15),grid on;
%
%
% figparades=figure();subplot(2,1,1);plot(parametroguarddes);xlabel('Iteraciones','fontsize',15);ylabel('valor de \alpha','fontsize',15),grid on,
% title('Convergencia de los parametros de regularizacion','Gradiente descendente'),'fontsize',20)
% legend('\alpha_1','\alpha_2');
% subplot(2,1,2);plot(funpargdes);title('Funcion de distancia','fontsize',20);xlabel('Iteraciones','fontsize',15);ylabel('Distancia','fontsize',15),grid on;
%
%
% figrecons=figure();
% subplot(2,2,1);plot(a);title('Se\u00f1al original','fontsize',15),ylabel('Amplitud','fontsize',12),grid on,
% subplot(2,2,2);plot(ac);title('Se\u00f1al contaminada','[SNR = \num2str(snr),dB Funcion \Delta = \num2str(K)]','fontsize',15),ylabel('Amplitud','fontsize',12),grid on,
% subplot(2,2,3);plot(Estimate,'g-x'),hold on,plot(Estimatedes,'b');title('Reconstruccion con AWCLS','[IOSNR_{RN} = \num2str(iosnr),dB','IOSNR_{GD} = \num2str(iosnrdes),dB]'),'fontsize',11),ylabel('Amplitud','fontsize',12),grid on,
% legend('Red neuronal','Gradiente descendente','fontsize',8);
% subplot(2,2,4);plot(Estimate_1,'g-x'),hold on,plot(Estimate_1des,'b');title('Reconstruccion con WCLS','[TOSNR_{RN} = \num2str(iosnr_1),dB','IOSNR_{GD} = \num2str(iosnr_1des),dB]'),'fontsize',11),ylabel('Amplitud','fontsize',12),grid on,
% legend('Red neuronal','Gradiente descendente','fontsize',8);
% figure(figsurf),hold on,plot3(log10(xtg),log10(xtg1),log10(xtg2),'gx');
% figure(figiosnr),hold on,plot3(parametrog(end,1),parametrog(end,2),iosnr,'go','markerfacecolor','g')
%
clear all;

```

```

function [a,ac,mv,d,N0,q,qS,S,Mu,Mv1,Mv2,I,C,b,ca,cb]=formaciondata(snr,func,K)
% funcion que genera y abre se\u00f1ales y entrega una imagen observada

```

```

%para imagen
rutaentrada='D:\simulaciones\gd133.bmp';
a=imread(rutaentrada);
a=im2double(a);

%para se\u00f1al
%a=zeros(1,256);
%1ra simulacion
% a(1)=1;a(256)=1; %condiciones de frontera
% a(50:80)=3*sin(pi/30)*(0:1:30));
% a(60)=5;a(90)=5;a(95)=2;
%segunda simulacion
%a(30)=6; a(45)=2; a(90)=4; a(130)=6; a(135)=6;
% a(1)=1;a(256)=1; %restricciones

```

```

d=size(a);
N=d(2);
%K=7;
%N0=1e-1;
%snr=5
tmv=1;
%func='exp';
%snr=10*log10((sum(sum(a.*a))/prod(d))/N0);
N0=((sum(sum(a.*a))/prod(d))/(10^(snr/10)));
%supone que la media es igual a cero
mv=zeros(N,1);

```

```

%se genera la matriz del sistema
S=matriz_sistema(func,N,K);

```

```

%matriz de restricciones y valores de frontera
C=zeros(2,N); %matriz de restriccion
C(1,1)=1;
C(2,N)=1;

```

```
b=[1;1]; %valores de frontera
```

```
%se genera las matrices de pesos  
[Mu,Mv1,Mv2,1]=wmatrices(N,N0,tmv);  
%parte del algoritmo de reconstruccion  
q=S'*Mu;  
qS=q*S;
```

```
for i=1:d(1)  
%formacion de la señal del sistema  
[u,v,Noise]=formacion(S,a(i,:),N0,N);  
ac(i,:)=u';  
end
```

```
singular=svd(S);  
mins=(min(singular))^2;  
maxs=(max(singular))^2;
```

```
%[imarecons_wcls,imarecons_1wcls,zt,xt]=recons_imagen(parametro,ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,zt,xt)
```

```
[Estimate,Estimate_1,zt,xt]=recons_imagen([mins,mins,mins],ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,0,[0,0,0]);  
ca=log10(zt);  
[Estimate,Estimate_1,zt,xt]=recons_imagen([maxs,0,0],ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,0,[0,0,0]);  
cb(1)=log10(xt(1));  
[Estimate,Estimate_1,zt,xt]=recons_imagen([0,maxs,0],ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,0,[0,0,0]);  
cb(2)=log10(xt(2));  
% [Estimate,Estimate_1,zt,xt]=recons_imagen([0,0,maxs],ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,0,[0,0,0]);  
% cb(3)=log10(xt(3));
```

---

```
function [imarecons_wcls,imarecons_1wcls,ac,a,iosnr,zt1,xt1,xt2]=pruebametodo(a,ac,mv,d,N0,q,qS,S,Mu,Mv1,Mv2,I,C,b,ca,cb)
```

```
alphaa=logspace(-5,10,20);  
for i=1:1:20  
for j=1:1:20
```

```
[imarecons_wcls,imarecons_1wcls,zt,xt]=recons_imagen([alphaa(i),alphaa(j),0],ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,0,[0,0,0]);
```

```
iosnr(i,j)=10*log10((sum(sum((ac-a).^2)))/(sum(sum((imarecons_wcls-a).^2))));  
zt(i,j)=zt;  
xt1(i,j)=xt(1);  
xt2(i,j)=xt(2);  
end  
end
```

---

```
%prueba del metodo iterativo para encontrar los valores optimos
```

```
%lee la imagen de entrada
```

```
function[iosnr,iosnr_1,ztg,xtg1,xtg2,Estimate,Estimate_1,iter,parametrog,funparg,alef,a,ac]=metodoiter2m(a,ac,mv,d,N0,q,qS,S,Mu,Mv1,Mv2,I,C,b,ca,cb)
```

```
zt=0;  
xt=[0,0,0];  
parametroa=zeros(1,3);  
parametro=[10,10];  
sumaant=[0,0];  
acum=[0,0];  
regla=1;  
iter=1;  
alef=5;  
funpara=1e6;  
%metodo
```

```
%iterativo  
while (regla > 1e-4)&(iter<40)
```

```
[funpar1,zt,xt,Estimate,Estimate_1]=funpar([parametro,0],ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,ca,cb);  
paraa=[parametro,0]+[1e-10,0,0];  
[funpard1,zt1,xt1,Estim,Estim_1]=funpar(paraa,ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,ca,cb);  
paraa=[parametro,0]+[0,1e-10,0];  
[funpard2,zt1,xt1,Estim,Estim_1]=funpar(paraa,ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,ca,cb);
```

```
funparg(iter)=funpar1;
```

```

ztg(iter)=zt;
xtg1(iter)=xt(1);
xtg2(iter)=xt(2);
if funpar1<funpara
    alef=alef;
else
    alef=alef/2;
end

funpara=funpar1;

difpar(1)=(funpard1-funpar1)/1e-10;
difpar(2)=(funpard2-funpar1)/1e-10;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%implementacion con red neuronal

parametro=((positivo(parametro-alef*(difpar))));
parametrog(iter,:)=parametro;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%calculo del iosnr
iosnr=10*log10((sum(sum((ac-a).^2)))/(sum(sum((Estimate-a).^2))));
iosnr_1=10*log10((sum(sum((ac-a).^2)))/(sum(sum((Estimate_1-a).^2))));
regla=abs((parametro*parametro')-(parametroa*parametroa));
parametroa=parametro;
iter=iter+1;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%estimacion de las funciones
function [dist,zt,xt,Estimate,Estimate_1]=funpar(parametro,ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,ca,cb)

[Estimate,Estimate_1,zt,xt]=recons_imagen(parametro,ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,0,[0,0,0]);
dist = (((abs(log10(zt)-ca))^2)+((abs(log10(xt(1))-cb(1)))^2)+((abs(log10(xt(2))-cb(2)))^2));

```

---

```

function [imarecons_wcls,imarecons_1wcls,zt,xt]=recons_imagen(parametro,ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,zt,xt)

```

```

%se anaden los parametros de regularizacion a las matrices de pesos
Mv = I*parametro(1) + parametro(2)*Mv1 + parametro(3)*Mv2;
%se generan las matrices necesarias para la reconstruccion
[Kalpha,Dalpha,W]=matics_est(S,Mu,Mv,C,q,qS);

for i=1:d(1)
[Estimate,Estimate_1]=Estimacion(ac(i,:),mv,S,C,b,q,Kalpha,Dalpha,W);
imarecons_wcls(i,:)=Estimate'; %metodo aumentado
imarecons_1wcls(i,:)=Estimate_1';

```

```

z1=(ac(i,:)- S*Estimate);
z=z1'*z1;
x1=(Estimate)*(Estimate); %el de la identidad
x2=Estimate'* Mv1 * Estimate; %el de Mv1
x3=Estimate'* Mv2 * Estimate; %el de Mv2
zt=zt+z;
xt(1)=xt(1)+x1;
xt(2)=xt(2)+x2;
xt(3)=xt(3)+x3;
end

```

---

```

%algoritmo de reconstruccion

```

```

function [Estimate,Estimate_1]=Estimacion(u,mv,S,C,b,q,Kalpha,Dalpha,W)
suavi=(u-S*mv);
palpha=C*Kalpha*q*suavi-b;
Estimate_1=mv+W*suavi;
Estimate = Estimate_1 - Kalpha*C'*Dalpha*palpha;

```

---

```
function [Kalpha,Dalpha,W]=matrices_est(S,Mu,Mv,C,q,qS)
%q=S'*Mu;
Kalpha=inv(qS+Mv);      %alpha se calculara desde Mv
Dalpha=inv(C*Kalpha*C');
W=(Kalpha)*q;
```

---

```
%funcion que genera la matrices de sistema
%func es el tipo de funcion 'sinc' 'rect' 'exp'
%N es la cantidad de
function S=matriz_sistema(func,N,K)
```

```
switch lower(func)
case 'sinc'
    S=fsinc(N,K);
case 'rect'
    S=frect(N,K);
case 'exp'
    S=fexp(N,K);
otherwise
    disp('funcion desconocida');
return
end
```

```
%----- Formation of the system matrix S -----
%----- for the system spread function: |sin(x)/x| -----
```

```
function S=fsinc(N,K)
a=zeros(N,1);
for i=1:2*floor(K/2),
a(i+1)=abs(sin(pi*i/floor(K/2))/(pi*i/floor(K/2)));
end;
a(1)=1;
%para hacer que la energia de la funcion sea 1
%a=a/sum(a);
S=toeplitz(a);
```

```
%----- for the system spread function exp-----
```

```
function S = fexp(N,K)
b=zeros(N,1);
for i=1:2*floor(K/2),
b(i+1)=exp(-i*i/5);
end
b(1)=1;
%hacer que la energia de la funcion sea 1
%b=b/sum(b);
S=toeplitz(b);
```

```
%----- for the system spread function rect-----
```

```
function S = frect(N,K)
b=zeros(N,1);
for i=1:floor(K/2),
    b(i)=5;
end
%para hacer energia igual a 1
%b=b/sum(b);
```

```
S=toeplitz(b);
%-----
```

---

```
%funcion que entrega la matriz de pesos Mv
%[Mu,Mv1,Mv2,1]=wmatrices(N,N0,bete,gammam,Alphe,tmv)
%N tamaño de la matriz
%N0 varianza del ruido
%tmv tipo de Mv 1 doble derivada 2 PID
function [Mu,Mv1,Mv2,1]=wmatrices(N,N0,tmv)
```

```

Mu=(1/N0)*eye(N);

b=zeros(N,1);
b(1)=2;
b(2)=-1;
Mv1=toeplitz(b);
Mv1(1,1)=1;
Mv1(N,N)=1;

I=eye(N);
%%parte anadida para realizar fórmula de Mv de la lectura 7 dsp1
%%parte para hacer la segunda derivada ya estan en la forma D*D
switch tmv
case 1
    b1=zeros(N,1);
    b1(1)=6;
    b1(2)=-4;
    b1(3)=1;
    Mv2=toeplitz(b1);
    Mv2(N,N)=1;
    Mv2(N,N-1)=-2;
    Mv2(N-1,N)=-2;
    Mv2(N-1,N-1)=5;
    Mv2=zeros(N);%solo para un caso de prueba y no tomar en cuenta un 3 parametro
%%partee para dar la parte integral que aumente a Mv2
case 2
    Mv2=triu(ones(N,N))*triu(ones(N,N));
otherwise
    disp('no existe Mv');
    return;
end
%Mv = I*Alphe + bete*Mv1 + gammam*Mv2;

function a=positivo(x)
for i=1:prod(size(x))
if x(i) < 0
    a(i)=0;
elseif x(i)>1e4
    a(i)=1e4;
else
    a(i)=x(i);
end
end

%%prueba del metodo iterativo para encontrar los valores optimos
%lee la imagen de entrada
function [iosnr,iosnr_1,Estimate,Estimate_1,iter,ztt1,xtt1,xtt2,parametroguard,funparg]=metodoiter(a,ac,mv,d,N0,q,qS,S,Mu,Mv1,Mv2,I,C,b,ca,cb)

xt=[0,0,0];
%parametroa=zeros(1,2);
parametro=[1e0,1e0];
parametroa=[0.0001,0.0001];
regla=1;
iter=1;

%metodo
%iterativo
while (regla > 1e-3) & iter<20
    [Estimate,Estimate_1,ztt,xtt]=recons_imagen([parametro,0],ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,0,[0,0,0]);
    ztt1(iter)=ztt;
    xtt1(iter)=xtt(1);
    xtt2(iter)=xtt(2);
    %para calcular los gradientes
    [Estimate,Estimate_1,ztd(1),xttd(1,:)] = recons_imagen([parametro,0]+[1e-8,0,0],ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,0,[0,0,0]);
    [Estimate,Estimate_1,ztd(2),xttd(2,:)] = recons_imagen([parametro,0]+[0,1e-8,0],ac,mv,S,C,b,q,qS,I,Mv1,Mv2,d,Mu,0,[0,0,0]);

    parametroa=parametro;
    parametroguard(iter,:)=parametroa;

for i=1:1

```



# **CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N. UNIDAD GUADALAJARA**

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

**Algoritmos para Optimización de Extracción y Mapeo de Información Geofísica Mediante Procesamiento de Datos de Percepción Remota**

del (la) C.

**Roberto CARRASCO ÁLVAREZ**

el día 07 de Septiembre de 2006.

Dr. Yuriy Shkvarko Sosnoff  
Investigador CINVESTAV 3C  
CINVESTAV Unidad Guadalajara

Dr. José Luis Leyva Montiel  
Investigador CINVESTAV 3B  
CINVESTAV Unidad Guadalajara

Dr. Deni Librado Torres Román  
Investigador CINVESTAV 3A  
CINVESTAV Unidad Guadalajara



CINVESTAV  
BIBLIOTECA CENTRAL



SSIT000008742