

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**PROCESAMIENTO DIGITAL Y ANÁLISIS DE MEDICIONES
ULTRASÓNICAS A TRAVÉS DE MATLAB**

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

MARCOS CRISANTO MORALES SOTO

ASESORADO POR EL ING. GUILLERMO ANTONIO PUENTE ROMERO

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN ELECTRÓNICA

GUATEMALA, ABRIL DE 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Luis Pedro Ortiz de León
VOCAL V	Agr. José Alfredo Ortiz Herincx
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Romeo Neftalí López
EXAMINADOR	Ing. Julio César Solares Peñate
EXAMINADOR	Ing. José Aníbal Silva
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

PROCESAMIENTO DIGITAL Y ANÁLISIS DE MEDICIONES ULTRASÓNICAS A TRAVÉS DE MATLAB

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha agosto de 2009.



Marcos Crisanto Morales Soto.

Guatemala, 20 de enero de 2011.

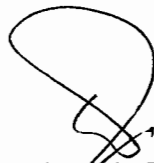
Ing. Carlos Eduardo Guzmán Salazar
Coordinador de Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Ingeniero Guzmán:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado: **“PROCESAMIENTO DIGITAL Y ANÁLISIS DE MEDICIONES ULTRASÓNICAS A TRAVÉS DE MATLAB”**, desarrollado por el estudiante **Marcos Crisanto Morales Soto con carné No. 2004-12914**, ya que considero que cumple con los requisitos establecidos, por lo que el autor y mi persona somos responsables del contenido y conclusiones del mismo.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,



Ing. Guillermo Antonio Puente Romero
ASESOR
Colegiado 5898

Guillermo A. Puente R.
INGENIERO ELECTRONICO
COL. # 5898



Ref. EIME 12. 2011
Guatemala, 4 de FEBRERO 2011.

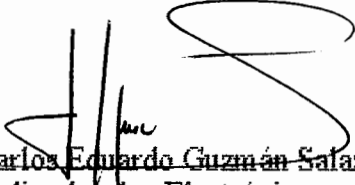
Señor Director
Ing. Guillermo Antonio Puente Romero
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado:
"PROCESAMIENTO DIGITAL Y ANÁLISIS DE MEDICIONES
ULTRASÓNICAS A TRAVÉS DE MATLAB", del estudiante,
Marcos Crisanto Morales Soto, que cumple con los requisitos
establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
D Y ENSEÑAD A TODOS


Ing. Carlos Eduardo Guzmán Salazar
Coordinador de Electrónica

CEGS/sro





REF. EIME 14. 2011.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; MARCOS CRISANTO MORALES SOTO titulado: "PROCESAMIENTO DIGITAL Y ANÁLISIS DE MEDICIONES ULTRASÓNICAS A TRAVÉS DE MATLAB" procede a la autorización del mismo.

Ing. Guillermo Antonio Puente Romero

GUATEMALA, 15 DE FEBRERO 2011.





DTG. 118.2011

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al trabajo de graduación titulado: **PROCESAMIENTO DIGITAL Y ANÁLISIS DE MEDICIONES ULTRASÓNICAS A TRAVÉS DE MATLAB**, presentado por el estudiante universitario **Marcos Crisanto Morales Soto**, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Murphy Olimpo Paiz Recinos
Decano

Guatemala, 27 de abril de 2011.



/gdech

ACTO QUE DEDICO A:

Mi padre

Quien con cada ingenio o invento, por sencillo que pareciera, hizo que emprendiera este camino.

Mis sobrinos

Michelle, Gerardo, Fabio y Alejandro, para que siempre sigan adelante.

AGRADECIMIENTOS A:

Familia Ramírez García

Karina, Fredy, Michelle y Alejandro, por dejarme formar parte de su familia.

Mi novia

Sol María Girón, por su apoyo imprescindible y su permanente confianza en mí.

Mis padres y hermano

Crisanto[†], Rosa y Cristian, por acompañarme y cuidarme desde niño y cultivar en mí el amor a Dios y al prójimo.

Familia Pérez Quintana

Por su apoyo y consejos, siendo una segunda familia para mí.

Familia Girón Cordón

Por todo el apoyo durante la carrera, principalmente el brindado durante los proyectos.

Tíos y Abuelos

Especialmente a: María Morales[†], Carlota Oliva[†], Isidoro Morales[†] y Ana Morales

Mis primos

Especialmente a: Edgar Roberto Soto, Susana, Alejandra y Luis Alberto Pérez, Jorge y Julio Muralles

Amigos y compañeros de Estudios

Especialmente a: Sol Girón, Rony Azurdia, German Chew, Juan Carlos González, Daniel Hernández Chang, Christian Aguirre, Jorge Espino, Manuel Ríos, Kevin Hernández, Carlos Archila, Manuel Lepe, Werner y Carlos Arreola, Alexis Bardales, Víctor Barrios y Fernando Vallejo

Amigos y compañeros del departamento de matemática

Especialmente a: Oscar Cerna, Alan y Juan Blanco, Armando Estumer, Erick Agustín, Marcos Sánchez, Daniel Cheley, Clyda González, Inga. Vera Marroquín, Inga. Ericka Cano, Ing. Arturo Samayoa, Ing. José Saquimux, Ing. Luis Carlos Bolaños, Ing. Francisco García, Ing. Carlos Garrido e Ing. Douglas Román

Ing. Guillermo Puente

Por ser mi amigo, catedrático y asesor de trabajo de graduación.

Universidad de San Carlos de Guatemala

Por permitirme formarme como profesional.

Y por último, pero no menos importante, DIOS

Por ponerme en el camino a todas las personas mencionadas y ayudarme a concluir esta meta.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
LISTA DE SÍMBOLOS	IX
GLOSARIO	XIII
RESUMEN	XXI
OBJETIVOS	XXIII
INTRODUCCIÓN	XXV
1. PRINCIPIOS ACÚSTICOS, ULTRASONIDOS Y APLICACIONES	1
1.1. Movimiento armónico simple	1
1.2. Energía del movimiento armónico simple	7
1.3. Tipos de ondas en el movimiento ondulatorio	8
1.3.1. Intensidad del movimiento ondulatorio	15
1.3.2. Ecuación diferencial del movimiento ondulatorio	16
1.4. Acústica física	17
1.4.1. Propagación de ondas sonoras	18
1.4.2. Impedancia Acústica	24
1.4.3. Magnitudes energéticas	25
1.4.4. Espectros de frecuencia	29
1.4.5. Medición del campo acústico	31
1.4.6. Disminución de niveles con la distancia	33
1.4.7. Reflexión y transmisión de ondas acústicas	34
1.5. Ultrasonido	38
1.6. Aplicaciones del ultrasonido	38
1.6.1. Aplicaciones médicas	39
1.6.2. Aplicaciones industriales	41

1.6.3.	Otras aplicaciones	42
2.	TRANSDUCTORES ULTRASÓNICOS Y TÉCNICAS DE MEDICIÓN	43
2.1.	Generación de ultrasonido	43
2.1.1.	Efecto piezoeléctrico	44
2.1.2.	Emisión y recepción de ultrasonido	46
2.2.	Transductores ultrasónicos	47
2.2.1.	Radiación del campo ultrasónico	48
2.2.2.	Tipos de transductores	52
2.3.	Técnicas de medición	55
2.3.1.	Técnica de transmisión	55
2.3.2.	Técnica de pulso eco	58
2.4.	Modos de presentación	61
2.4.1.	Representación A	62
2.4.2.	Representación B	62
2.4.3.	Representación C	63
3.	INTERFACES SENSOR-COMPUTADORA	65
3.1.	Descripción del sensor	65
3.2.	Comunicación serial	69
3.2.1.	El estándar <i>RS-232</i>	70
3.2.2.	Circuito integrado <i>MAX232</i>	71
3.2.3.	<i>Bus</i> serial universal	74
3.2.4.	Convertidor <i>USB</i> a <i>RS-232</i>	76
3.3.	Controlador del sensor	78
3.3.1.	Microcontrolador <i>PIC16F877A</i>	78
3.3.2.	Funciones del <i>BASIC Compiler</i> para comunicación Serial	79
3.4.	Comunicación serial utilizando <i>MATLAB</i>	81

4.	PROCESAMIENTO DIGITAL DE SEÑALES A TRAVÉS DE <i>MATLAB</i>	87
4.1.	Matrices e imágenes en <i>MATLAB</i>	87
4.1.1.	Conceptos básicos	88
4.1.2.	Lectura y escritura de imágenes	90
4.1.3.	Acceso a los elementos en imágenes	96
4.2.	Procesamiento de imágenes	100
4.2.1.	Cambio de tipo de imagen y formatos de color	100
4.2.2.	Detección de bordes	104
4.2.3.	Operaciones morfológicas	106
4.3.	Transformada de Radon	121
4.3.1.	Transformada de Radon en <i>MATLAB</i>	127
4.3.2.	Transformada inversa de Radon en <i>MATLAB</i>	131
4.4.	<i>Ultrasonic Signal Processing Toolbox</i>	134
4.4.1.	Obtención e instalación del paquete	135
4.4.2.	Descripción general	136
4.4.3.	Formatos aceptados	137
4.4.4.	Opciones de procesamiento de señales	137
4.5.	Resumen de Comandos y Funciones	138
4.5.1.	Funciones básicas y de uso general	138
4.5.2.	Funciones para el trabajo y procesamiento de Imágenes	139
4.5.3.	Funciones para el trabajo con la transformada de Radon	141
4.5.4.	Funciones impropias de <i>MATLAB</i>	141
5.	MONTAJES EXPERIMENTALES	143
5.1.	Descripción general	143
5.1.1.	Descripción para la obtención de mediciones	144

5.1.2.	Diagrama de bloques	144
5.2.	Diagrama del circuito eléctrico	145
5.3.	Programación del microcontrolador	147
5.4.	Programación en <i>MATLAB</i>	149
5.4.1.	Medidor de distancia	150
5.4.2.	Mapa de contorno	154
CONCLUSIONES		169
RECOMENDACIONES		173
BIBLIOGRAFÍA		175

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Representación de los parámetros del movimiento armónico simple	4
2.	Gráficas de desplazamiento, velocidad y aceleración	5
3.	Propagación a Velocidad v	11
4.	Vector de onda en 3 dimensiones	14
5.	Volumen alcanzado por una onda plana	15
6.	Onda de desplazamientos	18
7.	Intensidad de onda esférica	28
8.	Espectros a) onda pura, b) onda con dos armónicos, c) ruido	30
9.	Transmisión y reflexión de ondas acústicas	35
10.	Ecografía obstétrica a) 2D, b) 3D	40
11.	Representación de cuarzo con cortes x e y	45
12.	Diagrama de bloques de un generador ultrasónico	46
13.	Diagrama de bloques de un receptor ultrasónico	47
14.	Respuesta de un cristal a) débilmente, b) altamente atenuado	48
15.	Campo radiado por un transductor ultrasónico	49
16.	a) Intensidades de presión en el campo sónico, b) Amplitud de presión en función de la distancia al transductor	50
17.	Propagación de haces a diferentes frecuencias	51
18.	Transductor normal y sus partes	53
19.	Representación para técnica de transmisión	57
20.	Representación para técnica pulso-eco	60

21.	Técnica de incidencia angular	61
22.	Representación tipo B	63
23.	Representación tipo C	63
24.	Reconstrucción de imágenes por ultrasonido de inmersión	64
25.	Sensor <i>Devantech SRF04</i>	66
26.	Parte posterior del <i>SRF04</i> y agujeros de conexión	66
27.	Pulsos de sincronización del sensor	68
28.	Patrón de haz del <i>SRF04</i>	68
29.	Trama de comunicación asíncrona	69
30.	Conector <i>DB-9</i> y sus pines	70
31.	<i>MAX232</i> a) por <i>MAXIM</i> , b) por <i>TI</i>	73
32.	Conexión del <i>MAX232</i>	74
33.	a) cable <i>USB</i> , b) descripción de los pines	75
34.	Convertidor <i>TRENDNET TU-S9</i>	76
35.	Ventana de administrador de dispositivos	77
36.	Esquema <i>PIC16F877A DIP</i>	79
37.	Modificación de los parámetros del puerto serie	83
38.	Forma de envío de datos por el puerto serie	84
39.	Imagen escala de grises e indicación del editor de <i>colormap</i>	89
40.	Posibles escalas de la función <i>colormap</i>	89
41.	Detalles de imagen <i>RGB</i>	90
42.	Mostrando imagen con a) <i>imshow</i> , b) <i>imtool</i>	94
43.	Intensidad de grises de la imagen	94
44.	imágenes y sus distribuciones de intensidad	95
45.	Recorte en imagen	96
46.	Imagen a color	98
47.	Planos de imagen <i>RGB</i>	99
48.	Variaciones de intensidad utilizando <i>improfile</i>	100

49.	Imagen <i>RGB</i> , escala de grises y <i>HSV</i>	102
50.	Conversión de imágenes a blanco y negro	103
51.	Segmentación por umbral	104
52.	Detección de bordes	105
53.	Imágenes obtenidas por a) erosión, b) dilatación	108
54.	Criterios de conectividad	109
55.	a) imagen a procesar, b) imagen binaria	110
56.	Imagen erosionada utilizando <i>imobjct</i>	113
57.	Imagen dilatada utilizando <i>imobjct</i>	115
58.	Selección de objetos en una imagen	115
59.	Objetos resaltados a) erosión, b) dilatación	120
60.	Elementos en la transformada de Radon	121
61.	Representación del uso de la transformada de Radon	123
62.	Transformada para un ángulo de a) 19° y b) 64°	124
63.	Representación gráfica de la transformada de Radon	125
64.	Proyección de ángulos paralelos a determinado ángulo	127
65.	Imagen creada para aplicar la transformada de Radon	128
66.	Transformada de Radón en ángulos específicos	129
67.	Representación gráfica para la transformada completa de Radon	130
68.	Transformada de Radon con intensidades en la imagen original para a) 0° y 45°, b) todos los ángulos	131
69.	Conjunto de imágenes reconstruidas con <i>iradon</i>	134
70.	Interfaz gráfica de usuario del <i>USPT</i>	136
71.	Disposición del sensor para a) sonar, b) creación de mapa de entorno	144
72.	Diagrama de bloques, comunicación sensor computadora	145
73.	Diagrama del circuito eléctrico	146
74.	Imágenes de la interfaz	147

75.	Dispersión y tendencia de las mediciones	154
76.	Esquemas de la región a escanear	155
77.	Imágenes reales de la región escaneada	156
78.	Reconstrucción de la región escaneada	158
79.	Reconstrucción en 3D de la región escaneada	159
80.	Imagen escalada posterior al filtrado	160
81.	Imagen en 3D posterior al filtrado	161
82.	Objetos encontrados en la imagen escaneada	164

TABLAS

I.	Impedancia característica para distintos medios	24
II.	Niveles de voltaje <i>RS-232</i>	71
III.	Conversión de voltajes del <i>MAX232</i>	72
IV.	Formatos de imágenes permitidos en <i>MATLAB</i>	91
V.	Tipos de imágenes según el formato de sus píxeles	92
VI.	Funciones para modificar formatos de imágenes	101
VII.	Propiedades de la función <i>regionprops</i>	117
VIII.	Opciones de interpolación en la retroproyección	132
IX.	Opciones de filtrado	132
X.	Resultados de las primeras mediciones	150
XI.	Resultados de las segundas mediciones	153

LISTA DE SÍMBOLOS

Símbolo	Significado
a	Aceleración
α	Ángulo de desfase alfa
θ	Ángulo de desfase theta
BPS	<i>Bits</i> por segundo
USB	<i>Bus</i> serial universal
Δ	Cambio
l_0	Campo cercano de radiación
PIC	Circuito integrado programable
R_E	Coefficiente de reflexión
T_E	Coefficiente de transmisión
PC	Computadora personal
K	Constante elástica
dB	Decibelio
ρ	Densidad
ρ_0	Densidad del aire en reposo
x	Desplazamiento en el eje de las ordenadas
x_m	Desplazamiento máximo en el eje de las ordenadas
DIP	Encapsulado en línea dual
E_c	Energía cinética
E	Energía mecánica
U	Energía potencial
NDT	Ensayo no destructivo

COM	Estándar recomendado 232
RS-232	Estándar recomendado 232
TIFF	Formato agregado de imagen
GIF	Formato de intercambio gráfico
f	Frecuencia
f_0	Frecuencia fundamental
PNG	Gráficos de red portátil
JPEG	Grupos expertos de fonográficos comunes
HZ	Hertzio, hercio o <i>hertz</i>
Z	Impedancia acústica
I	Intensidad
MATLAB[®]	Laboratorio de matrices, marca registrada
π	Letra griega pi, representa la relación entre la longitud de una circunferencia y su diámetro
Σ	Letra griega Sigma, representa una sumatoria
TTL	Lógica transistor a transistor
λ	Longitud de onda
y	Magnitud del movimiento ondulatorio en el eje de las ordenadas
BMP	Mapa de <i>bits</i>
m	Masa
B	Módulo de compresibilidad
MAS	Movimiento armónico simple
k	Número de onda
$\vec{\nabla}$	Operador vectorial nabla
USPT	Paquete de procesamiento de señales ultrasónicas
T	Período
W	Potencia acústica

p	Presión acústica
P_0	Presión atmosférica
Pa	Presión atmosférica
.	Producto punto
ω	Pulsación o frecuencia angular
SRF04	Sensor ultrasónico utilizado
PS	Sistema personal
DB	Subminiatura D
ATDD	Subministro de fármacos a través de acústica dirigida
TI	<i>Texas instrument</i>
TU-S9	Tipo de convertidor <i>USB</i> a <i>RS-232</i>
HSB	Tonalidad de saturación de brillo
HSV	Tonalidad de saturación de valor
RGB	Tonos rojo, verde y azul
HIFU	Ultrasonido de alta intensidad
HUS	Ultrasonido focalizado
\vec{r}	Vector posición
\vec{u}	Vector unitario
v	Velocidad
c	Velocidad de propagación acústica
V	Volumen

GLOSARIO

Acústica	Rama de la física que estudia los infrasonidos, sonidos y ultrasonidos, es decir ondas mecánicas que se propagan en la materia.
Bit	Dígito del sistema binario de numeración.
Coefficiente de reflexión R_E	Describe la intensidad de una onda reflejada respecto a la onda incidente, que ocurre debido a un cambio de impedancia en el medio de propagación.
Coefficiente de transmisión R_T	Describe la intensidad de una onda transmitida respecto a la onda incidente, que ocurre debido a un cambio de impedancia en el medio de propagación.
Comunicación serial	Comunicación en la cual los datos de información viajan a través de <i>bits</i> y solamente un <i>bit</i> a la vez en el mismo medio.
Constante elástica K	Designa la propiedad mecánica de cierto material a sufrir deformaciones bajo fuerzas exteriores y la capacidad del mismo para obtener la forma original al eliminar dicha fuerza externa.

Decibelio	Llamado también decibel, es una unidad logarítmica empleada en acústica y telecomunicaciones para expresar la relación entre dos magnitudes, acústicas o eléctricas, o entre la magnitud que se estudia y una magnitud de referencia.
Ecografía	Procedimiento que emplea los ecos de una emisión de ultrasonidos dirigida sobre un cuerpo u objeto, como fuente de datos para formar una imagen de los órganos o masas internas con fines de diagnóstico.
Ecolocalización	También conocida como ecolocación y biosonar se asemeja al funcionamiento de un sonar activo; el animal emite un sonido que rebota al encontrar un obstáculo y analiza el eco recibido.
Efecto piezoeléctrico	Fenómeno presentado por determinados cristales que al ser sometidos a tensiones mecánicas adquieren una polarización eléctrica en su masa, apareciendo una diferencia de potencial y cargas eléctricas en su superficie.
Energía	Capacidad de realizar algún trabajo.
Energía cinética	Trabajo necesario para acelerar un cuerpo desde el reposo hasta la velocidad que posee éste.
Energía mecánica	Suma de energía cinética y potencial.

Energía potencial	Capacidad de un cuerpo para realizar trabajo en función de su posición.
Ensayo no destructivo	Prueba practicada a un material sin alterar de forma permanente sus propiedades dimensionales, físicas, mecánicas y/o químicas.
Espectro audible	Gama de frecuencias que pueden ser percibidas por el oído humano, comúnmente entre 20Hz a 20KHz.
Espectro de frecuencia	Medida de la distribución de la amplitud de una señal para cada frecuencia.
Frecuencia	Número de oscilaciones completas por unidad de tiempo, es el inverso del período.
Frente de ondas	Es el lugar geométrico de los puntos del espacio que en todo instante se encuentran en el mismo estado de perturbación.
Filtrado de dominio espacial	Proceso que relaciona, para todos y cada uno de los puntos de una imagen, un conjunto de píxeles próximos al píxel objetivo, con la finalidad de obtener una información útil, dependiente del tipo de filtro aplicado, que permita actuar sobre el píxel en que se está llevando a cabo el filtrado para obtener mejoras sobre la imagen y/o datos procesados.

Impedancia acústica	Cociente entre la presión acústica de un punto y la velocidad que posee una partícula en el momento en que pasa por dicho punto en determinado medio.
Indexar	Acción de registrar ordenadamente información para elaborar su índice, con la finalidad de obtener resultados de forma rápida al momento de realizar una búsqueda.
Infrasonido	Onda acústica o sonora cuya frecuencia está por debajo del espectro audible del oído humano.
Inteligencia artificial	Rama de las Ciencias de la Computación dedicada al desarrollo de agentes racionales no vivos.
Intensidad acústica	Vector que cuantifica la magnitud que transporta una onda acústica.
Interfaz	Conexión entre dos ordenadores o máquinas de cualquier tipo dando una comunicación entre ambas.
Movimiento armónico	Movimiento consecuente de la superposición de dos o más movimientos armónicos simples.
Movimiento armónico simple	Movimiento periódico descrito en función del tiempo por una sola función armónica.

Microcontrolador	Circuito integrado que posee las tres unidades funcionales de una computadora –recolección, procesamiento y transmisión de datos.
Onda	Propagación de la perturbación de una o más propiedades de un medio.
Operaciones morfológicas	Trabajo de imágenes binarias para la detección o reconocimiento de formas dentro de las mismas.
Período	Intervalo de tiempo mínimo entre dos posiciones idénticas de una partícula con movimiento oscilatorio.
Potencia acústica	Energía por unidad de tiempo que una fuente transmite al medio en forma de ondas acústicas.
Presión acústica	Presión entre las partículas alcanzadas por un movimiento ondulatorio con respecto a la presión de las mismas cuando se encuentran en reposo.
Propagación	Conjunto de fenómenos que conducen a las ondas desde un transmisor.
Pulsación o frecuencia angular	Ángulo girado por unidad de tiempo.
Rayl	Unidad de la impedancia acústica en el sistema inglés.

Rayos X	Designa a una radiación electromagnética, invisible, capaz de atravesar cuerpos opacos y de imprimir las películas fotográficas.
Retroproyección filtrada	Algoritmo utilizado para calcular la transformada inversa de Radon.
Segmentación por umbral	Técnica utilizada para convertir una imagen a imagen binaria, especificando el nivel de decisión para asignar la magnitud binaria en cada pixel.
Senograma	Denominación dada a la transformada de Radon en el contexto de tomografías y es el conjunto de datos almacenados de una medición tomográfica.
Sensor	Dispositivo capaz de transformar magnitudes físicas o químicas en variables eléctricas.
Sonar	Técnica que usa la propagación acústica para detectar otros objetos y la distancia entre estos.
Sonido	Fenómeno que involucra la propagación en forma de ondas elásticas –audibles o no–, que esté generando el movimiento vibratorio de un cuerpo.
Sonido audible	Oscilaciones de la presión del aire, que son convertidas en ondas mecánicas en el oído humano y percibidas por el cerebro.

Telémetro	Dispositivo capaz de medir distancias en forma remota.
Transductor	Un transductor es un dispositivo capaz de transformar o convertir un determinado tipo de energía de entrada, en otra de diferente a la salida.
Transformada de Fourier	Básicamente es el espectro de frecuencias de una función.
Transformada de Radon	Transformación lineal de una función espacial sobre un conjunto de rectas.
Tomografía	Método utilizado para la creación de imágenes por medio de datos almacenados de mediciones en secciones transversales de una región espacial.
Ultrasonido	Onda acústica o sonora cuya frecuencia está por encima del espectro audible del oído humano.
Visión artificial	También conocida como visión por computador, su propósito es crear un algoritmo para que un computador analice las características en una imagen.

RESUMEN

El ultrasonido es una onda acústica de alta frecuencia inaudible para los humanos. Normalmente está sobre los 20 KHz y posee las propiedades de cualquier onda mecánica, como la reflexión y transmisión bajo cambios en la impedancia acústica del medio de propagación. Esto hace posible trabajar con técnicas de transmisión y pulso-eco, dando origen a múltiples aplicaciones en diversas áreas.

Es posible utilizar diferentes tipos de transductores ultrasónicos, diseñados para aplicaciones específicas e incorporados en sensores, que a su vez proporcionan variables eléctricas necesarias para cada diseño. Con estos transductores se facilita la comunicación con un computador, a través de una interfaz. En este caso la interfaz está constituida por un microcontrolador *PIC16F877A*, que permite la transmisión serial de los datos y sirve como dispositivo de control para el sensor.

Desde el punto de vista del computador, la recolección, procesamiento y análisis de datos provenientes del sensor se realiza con ayuda de *MATLAB*. *MATLAB* fue escogido por su facilidad y robustez en el trabajo con imágenes de forma matricial y la sencillez con que se configura la transmisión serial de datos. Se sintetiza al final un medidor de distancias y un sistema de visión artificial a través de mediciones ultrasónicas.

OBJETIVOS

General

Estructurar algoritmos en *MATLAB* capaces de indexar, procesar, analizar y representar visualmente mediciones ultrasónicas, para la creación de un sistema de visión por computador, con posible aplicación en áreas de inteligencia artificial y robótica.

Específicos

1. Describir el comportamiento del ultrasonido desde el punto de vista acústico, por medio de modelos físicos y matemáticos, identificando algunas de las aplicaciones del mismo en diversas áreas.
2. Describir las técnicas y modos de representación existentes para mediciones ultrasónicas.
3. Identificar el principio de funcionamiento y clasificación de los transductores ultrasónicos y el campo radiado por estos utilizando diferentes frecuencias.
4. Crear una interfaz entre un sensor ultrasónico y un computador capaz de indexar las mediciones en matrices con ayuda de *MATLAB*.
5. Describir algoritmos creados en *MATLAB* para el procesamiento de imágenes y mediciones ultrasónicas almacenadas en matrices.

6. Elaborar un sistema de visión artificial básico a través de *MATLAB*, capaz de analizar un mapa de entorno creado a partir de mediciones ultrasónicas.

INTRODUCCIÓN

Hace poco más de cien años, el hombre se interesó en el comportamiento de ciertos insectos, que al chocar periódicamente algunas partes de su cuerpo entre sí emitían algún tipo de sonido. El problema radicó al observar que cuando dicho movimiento alcanzaba cierta rapidez, el sonido desaparecía por completo para el observador. Inevitablemente surgió el interés de averiguar si la onda sonora emitida por el insecto desaparecía o llegaba a un estado fuera del rango de percepción humana. La dificultad era demostrar cuál de las dos condiciones anteriores ocurría.

Fue hasta inicios del siglo XX en que se estableció la electrónica como ciencia y con ésta el inicio de los instrumentos electrónicos que proporcionaron a los investigadores las herramientas necesarias para hacer las ondas sonoras “visibles”. Pudiendo determinar que los movimientos periódicos a frecuencias mayores a unos 20,000 ciclos por segundo, se hacían inaudibles para los humanos, agrupándolos y denominando al conjunto como ultrasonidos. Es importante resaltar, que el hecho de no ser audibles por el hombre, no le resta ni una propiedad de cualquier onda acústica, por lo tanto no dejan de ser sonidos, no para las personas, pero si para diferentes tipos de insectos, murciélagos o delfines, que lo usan para comunicarse entre sí o para localizarse en entornos no visibles.

Desde el siglo pasado, a través de los avances en la electrónica y otras investigaciones relacionadas, se han realizado instrumentos ultrasónicos para finalidades que van desde el control de plagas, sistemas de sonar y ensayos no destructivos de materiales, hasta las ecografías que muestran imágenes complejas de órganos y masas internas al cuerpo humano. Ésta última es una de las aplicaciones más importantes, ya que es un método que contrasta con la mayoría de sistemas de visión médica por ser no invasivo, accesible y seguro.

La definición formal de ultrasonido, hasta el momento, hace referencia a cualquier onda acústica o sonora, cuya frecuencia se encuentre sobre el espectro audible humano, estandarizado en un rango de frecuencias entre 20Hz y 20KHz, aunque puede variar dependiendo de la persona y se modifica con la edad.

En el primer capítulo del presente trabajo, dirigido principalmente a personas con conocimientos básicos de física y cálculo matemático, se presentan las características y propiedades del ultrasonido desde el punto de vista acústico. El objetivo es comprender más a fondo las cualidades del ultrasonido que hacen posibles muchas aplicaciones, detallándose al final del capítulo algunas de éstas.

En el capítulo dos, se describen los fenómenos –como el efecto piezoeléctrico– que hacen posible la construcción de transductores para la transmisión y recepción de ultrasonidos. También se detallan las dos técnicas que sirven de base a cualquier medición ultrasónica: la técnica de transmisión y la técnica de pulso eco. Se detallan también los tres modos de representación de estas técnicas.

En el tercer capítulo se describe la interfaz encargada de la comunicación entre el computador y el transductor, las características del sensor *Devantech SRF04*, los fundamentos de la comunicación serial, la descripción del microcontrolador *PIC16F877A* y los fragmentos de código básicos para la recepción de datos en la computadora.

En síntesis, los primeros tres capítulos están enfocados al estudio del ultrasonido, sus aplicaciones y la comunicación del sensor ultrasónico con el computador. El otro tema central del presente trabajo es el procesamiento digital de señales e imágenes, el cual ha sido posible en los últimos años, debido al avance tecnológico y construcción de computadoras que incluyen procesadores capaces de realizar tareas complejas en tiempo real, o muy cercano a éste y que permiten crear algoritmos para la reconstrucción y análisis de imágenes de manera eficiente.

El capítulo cuatro, contiene una recopilación y descripción de algoritmos diseñados en *MATLAB*, versión R2009a, para el procesamiento y análisis de imágenes. *MATLAB* fue elegido por su eficiente trabajo con matrices y su capacidad para aprovechar el procesador interno de la computadora al utilizar paquetes de procesamiento digital de imágenes y señales, así como de adquisición de datos. Es por ello que *MATLAB* se convierte en el entorno de trabajo idóneo para analizar y visualizar mediciones ultrasónicas indexadas en matrices.

En el último capítulo se describen dos montajes experimentales. El primero consiste en un medidor de distancias que posteriormente se utiliza para el segundo montaje, con el cual se concluye el objetivo principal del presente trabajo. Éste último montaje hace uso de los algoritmos estructurados en el cuarto capítulo para procesar y analizar una imagen constituida por un conjunto mayor a mil mediciones ultrasónicas, no automatizadas, de una región. Se crea un mapa de contorno y se construye un sistema de visión por computador básico que es capaz de proporcionar la cantidad de objetos detectados en la región escaneada, la distancia de éstos al centroide y el área aproximada de los mismos.

Se espera que la información descrita sea de utilidad en investigaciones posteriores, buscando enlazar las diferentes áreas de aplicación del ultrasonido con proyectos de la carrera de electrónica a fin de potenciar la construcción de instrumentos de medición por ultrasonido.

1. PRINCIPIOS ACÚSTICOS, ULTRASONIDOS Y APLICACIONES

El ultrasonido es una onda acústica de alta frecuencia inaudible para el oído humano. Posee muchas aplicaciones en diversos campos, principalmente en sistemas de diagnóstico y tratamiento médico, en la industria para mantenimiento preventivo y ensayos no destructivos de materiales. Actualmente ha tomado importancia en sistemas de visión artificial aplicados a la robótica e inteligencia artificial.

En este primer capítulo se estudian las bases de la acústica física, los diferentes tipos de ondas, propagación de éstas y su comportamiento en diversos medios, ayudando a comprender el funcionamiento de los sistemas basados en ultrasonido.

1.1. Movimiento armónico simple

Se dice que un punto, o sistema de puntos, describe un movimiento periódico cuando sus parámetros cinemáticos y dinámicos toman valores idénticos a intervalos regulares de tiempo denominados períodos. Como ejemplo de esto, se considera el movimiento que describe un péndulo, se puede observar un cambio regular de su valor de estado -posición de una partícula de la masa- y el período de alcance de una condición instantánea.

En los movimientos periódicos de tipo oscilatorio o vibratorio, la partícula tiene una posición de equilibrio y cuando se desplaza de la misma, surge una fuerza restauradora que le hace experimentar un movimiento de vaivén respecto a la citada posición de equilibrio. El movimiento vibratorio más sencillo, es el armónico simple, en el cual la fuerza que actúa sobre la partícula es proporcional, en cada instante, a la distancia que le separa del origen o posición de equilibrio.

Por otra parte, según el teorema de Fourier, todo movimiento periódico puede considerarse como la suma de movimientos armónicos elementales. De ahí la importancia de estudiar el movimiento armónico simple –MAS–. Este es el movimiento vibratorio más elemental .y se define como aquel que posee un punto o partícula, cuya distancia a la posición de equilibrio se describe con la ecuación:

$$x(t) = x_m \text{sen}(\omega t + \theta_0) \quad (1.1)$$

El parámetro x indica la distancia de un punto a la posición de equilibrio para un tiempo t y se denomina desplazamiento o elongación del movimiento. Al desplazamiento máximo que alcanza la partícula x_m se le denomina amplitud.

A la cantidad $(\omega t + \theta_0)$ se le denomina fase del movimiento y al valor θ_0 , corresponde al instante inicial, o sea, la fase inicial para $t = 0$. Si la fase inicial se considera $\alpha_0 = \theta_0 - \pi/2$, la ecuación (1.1) se puede escribir

$$x(t) = x_m \cos\left(\omega t + \theta_0 - \frac{\pi}{2}\right) = x_m \text{sen}(\omega t + \alpha_0) \quad (1.2)$$

Se denomina período T , al intervalo de tiempo mínimo entre dos posiciones idénticas de la partícula. Para obtener la velocidad de la partícula se calcula la primera derivada de la ecuación de desplazamiento respecto al tiempo, ecuación 1.1

$$v(t) = x'(t) = x_m \omega \cos(\omega t + \theta_0) \quad (1.3)$$

Es posible denotar con Δt al tiempo necesario para que una partícula cumpla con las condiciones de posición $x(t) = x(t + \Delta t)$ y velocidad $v(t) = v(t + \Delta t)$, dando lugar a:

$$\omega(t + \Delta t) + \theta_0 - (\omega t + \theta_0) = 2n\pi$$

Por lo que

$$\Delta t = \frac{2n\pi}{\omega}$$

Este tiempo es mínimo cuando $n = 1$, y se puede definir el período como el tiempo que tarda una partícula en realizar una oscilación completa:

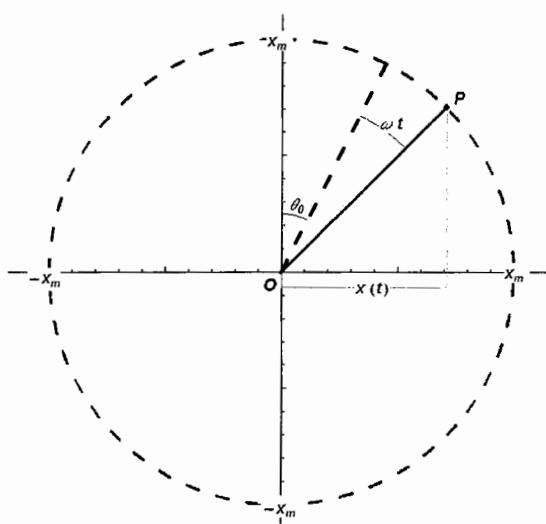
$$T = \frac{2\pi}{\omega} \quad (1.4)$$

La frecuencia f es el número de oscilaciones completas por unidad de tiempo, esto es, la inversa del período. La unidad de frecuencia, en el sistema internacional, es el Hertzio ($1\text{Hz}=1/\text{s}$).

$$f = \frac{1}{T} = \frac{\omega}{2\pi} \quad (1.5)$$

Se realiza otra explicación de todos los parámetros mencionados, representado al movimiento armónico simple de una partícula, como la proyección sobre el diámetro de una circunferencia con radio x_m del movimiento circular uniforme de una partícula y con velocidad angular constante ω , como se muestra en la figura 1. Es decir que elegida como fase inicial θ_0 , la distancia al centro O de la proyección P sobre el diámetro, coincide con la definición de movimiento armónico, ecuación 1.1.

Figura 1. **Representación de los parámetros del movimiento armónico simple**

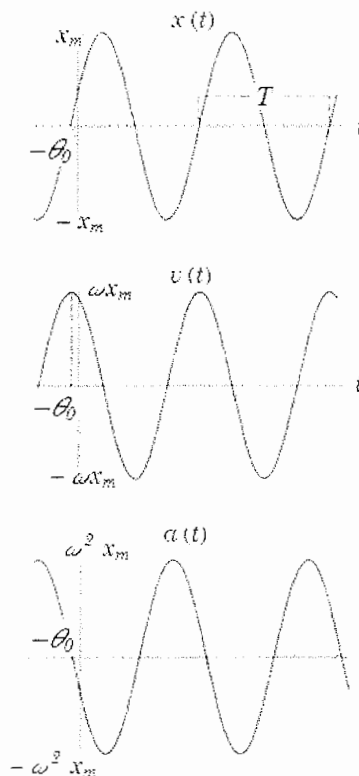


El período es el tiempo que tarda el punto P en dar una vuelta completa (por convenio en sentido anti horario como positivo) y ω es la velocidad angular de giro, denominada Pulsación o frecuencia angular. Para obtener la aceleración de la partícula en movimiento se deriva con respecto al tiempo la ecuación de velocidad (1.3).

$$a(t) = x''(t) = -x_m \omega^2 \text{sen}(\omega t + \theta_0) = -\omega^2 x(t) \quad (1.6)$$

Se puede observar que la aceleración es proporcional y opuesta al desplazamiento de la partícula. En la figura 2 se muestran las gráficas del desplazamiento, velocidad y aceleración para un movimiento armónico simple.

Figura 2. **Gráficas de desplazamiento, velocidad y aceleración**



Utilizando la expresión de aceleración, ecuación 1.6, se puede obtener una expresión para la fuerza F que experimenta la partícula en determinado momento.

$$F(t) = -m\omega^2 x_m \text{sen}(\omega t + \theta_0) = -m\omega^2 * x(t)$$

Se dice que $K = m\omega^2$ y se denomina constante elástica y se relaciona con el período T mediante:

$$T = \frac{2\pi}{\omega} = 2\pi \sqrt{\frac{m}{K}}$$

Teniendo como resultado:

$$F(x) = -K * x(t) \tag{1.7}$$

La fuerza siempre trata de llevar a la partícula a su posición de equilibrio. Cuando la partícula alcanza la posición de equilibrio, la fuerza restauradora ha disminuido hasta cero, pero debido a la velocidad adquirida, ésta rebasa la posición de equilibrio y continúa moviéndose. Un instante después de pasar la posición de equilibrio, la fuerza entra de nuevo en acción, aunque la velocidad disminuye cuando aumenta con la distancia hasta que se detiene, siendo en este instante la fuerza que lo atrae nuevamente hacia el origen es máxima. En utilizar la ecuación fundamental de la dinámica $F = ma(t) = -Kx(t)$ se obtiene

$$\frac{d^2}{dt^2}x(t) + \omega^2x(t) = 0$$

Ecuación diferencial de segundo orden que posee solución del tipo $x(t) = x_m \text{sen}(\omega t + \theta_0)$, donde ω y θ_0 son constantes. Con lo que se confirma la definición dada con la ecuación 1.1. A ω^2 se le conoce como constante armónica.

1.2. Energía del movimiento armónico simple

Para conocer la energía cinética de una partícula con movimiento armónico simple, se sustituye la velocidad, ecuación 1.3, en la expresión general de energía cinética:

$$E_c(t) = \frac{1}{2}mv(t)^2 = \frac{1}{2}mx_m^2\omega^2\cos^2(\omega t + \theta_0) \quad (1.8)$$

La energía potencial desarrollada por la fuerza conservativa $F = -Kx$ será

$$-dU = \vec{F} \cdot \vec{dr} = -Kx dx$$

$$U(t) = \int_0^x Kudu = \frac{K}{2}x^2(t) = \frac{K}{2}x_m^2\sin^2(\omega t + \theta_0)$$

Tomando la constante elástica $K = m\omega^2$, resulta:

$$U(t) = \frac{1}{2}m\omega^2x_m^2\sin^2(\omega t + \theta_0) \quad (1.9)$$

La energía total en cualquier instante t , es la suma de energía cinética y potencial.

$$E(t) = \frac{1}{2}mx_m^2\omega^2[\cos^2(\omega t + \theta_0) + \sin^2(\omega t + \theta_0)]$$

$$E(t) = \frac{mx_m^2\omega^2}{2} = E \quad (1.10)$$

Como se puede ver en la expresión anterior, la energía total es la misma para cualquier instante.

1.3. Tipos de ondas en el movimiento ondulatorio

El campo acústico es el medio por el cual se propagan variaciones de magnitudes físicas como presión, densidad o desplazamiento de partículas de aire que describen un movimiento ondulatorio u oscilatorio.

Como ejemplo de movimiento ondulatorio se considera la superficie de un estanque con agua. Sobre el agua se dispone una fila de corchos puntuales equidistantes. Si en un punto de la superficie se provoca una perturbación, ésta se propaga en todas direcciones, con particular interés de estudio, en la línea de corchos. Se puede intuir que la perturbación se propaga a una cierta velocidad alcanzando todos los corchos. También se observa que los corchos realizan movimientos transversales a la dirección de propagación, sin desplazarse en dirección de la misma, describiendo un movimiento periódico con características similares al armónico simple descritas en el apartado anterior.

Se puede concluir que el lugar geométrico de los corchos, mientras transcurre el tiempo, constituye una onda o movimiento ondulatorio. La magnitud física estudiada en este ejemplo es el desplazamiento respecto a la horizontal que varía en dirección normal a la de la propagación.

Se considera ahora el movimiento de propagación de una perturbación en el aire, considerado como fluido perfecto e inicialmente en equilibrio. Se supone que la perturbación, que origina el movimiento, está provocada por el movimiento vibratorio de un plano vertical, las partículas de aire próximas al plano que vibra, serán desplazadas por el movimiento al ser empujadas. Las partículas realizan movimientos vibratorios en la dirección de propagación de la perturbación, a medida que son alcanzadas por la misma.

Al lugar geométrico de los puntos del espacio que en todo instante se encuentran en el mismo estado de perturbación, se denomina frente de ondas. En los ejemplos descritos, los frentes de ondas son circunferencias concéntricas con el centro de vibración el punto donde se origina la perturbación. En ejemplos que se plantearon anteriormente los dos tipos de onda necesitan un medio material para poder propagarse, ya que lo hacen por contacto y choques entre las partículas que vibran y las inmediatamente próximas a ellas.

Según los ejemplos descritos y por tanto, según la dirección de la variación de la magnitud física que se propaga, las ondas mecánicas se dividen en ondas longitudinales y transversales.

- **Ondas longitudinales:** cuando el movimiento de las partículas es paralelo a la dirección de propagación, como es el caso de las partículas de aire.
- **Ondas transversales:** cuando el movimiento de las partículas es perpendicular a la dirección de propagación, éstas descritas en el ejemplo de la superficie del líquido.

En los sólidos pueden propagarse ambos tipos de onda, pero en los fluidos perfectos no pueden propagarse ondas transversales, dado a que no aceptan tensiones tangenciales. Se hace sencillo representar físicamente el perfil de onda, para las ondas transversales, pues las ordenadas del perfil son los desplazamientos en cada momento de cada partícula. Mientras que para las ondas longitudinales el perfil de onda no posee existencia física, para obtenerlos en posición habría que abatirlos sobre la dirección de propagación, que es la dirección donde se produce el movimiento.

Por otra parte, las ondas mecánicas también pueden clasificarse dependiendo de la forma de su frente de onda. Cuando los frentes de ondas son planos paralelos entre sí y perpendiculares a la dirección de propagación, se denominan ondas planas. Las ondas cuyos frentes de ondas son superficies esféricas concéntricas, son llamadas ondas esféricas y las ondas cilíndricas con frentes de ondas superficies cilíndricas coaxiales.

Es importante notar que en las ondas mecánicas las partículas del medio entran en vibración, oscilando alrededor de su posición de equilibrio, pero éstas no se desplazan. Por lo que se afirma que en este tipo de movimiento se transmite energía y no materia.

Teniendo los conceptos necesarios, se modela matemáticamente la representación del movimiento ondulatorio. Si la perturbación es periódica, ésta se propaga a los demás puntos del medio, entrando en vibración sin importar que tanto después y más alejados se encuentren del origen de vibración. Para simplificar la deducción matemática se considera que la propagación ocurre en un medio ideal, o sea que no presenta fuerzas de amortiguamiento.

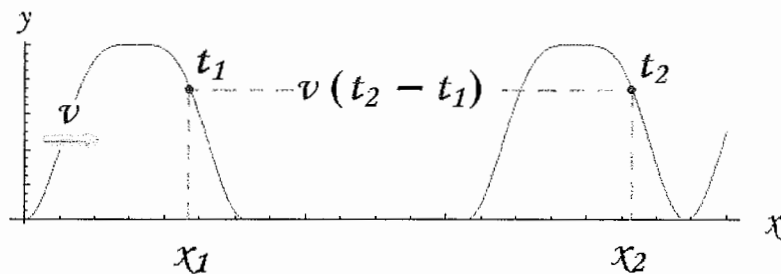
El valor de la magnitud física de interés y para cualquier punto en el eje de las abscisas y en un instante determinado, es función de la distancia al origen x y del tiempo t . Se escribe entonces.

$$y = y(x, t) \quad (1.11)$$

Se puede relacionar los parámetros x y t , con la velocidad v , velocidad de propagación de la onda, ver figura 3.

$$x_2 - x_1 = vt_2 - vt_1$$

Figura 3. Propagación a Velocidad v



Siendo $x_2 - x_1$ la distancia recorrida por la propagación entre los instantes t_1 y t_2 . Se ordena esta última expresión y se obtiene:

$$x_2 - vt_2 = x_1 - vt_1 \quad (1.12)$$

Esta relación cumple para cualquier par de puntos. Por lo que se afirma que la ecuación (1.11) será de la forma:

$$y = f(x - vt) \quad (1.13)$$

La expresión anterior representa a una propagación con dirección hacia la derecha del origen, y se puede modificar para representar a una onda propagándose hacia la izquierda $y = f(x + vt)$.

Se observa que el movimiento ondulatorio posee periodicidad y según el teorema de Fourier, cualquier función periódica puede expresarse como la combinación lineal de funciones senoidales, se enfoca el estudio a la forma elemental componente:

$$y = y_m \text{sen}k(x - vt) \quad (1.14)$$

Se denomina longitud de onda, λ , a la menor distancia, medida en el eje de propagación, entre dos puntos consecutivos que tienen el mismo estado de vibración, (magnitud y velocidad). Se escribe que para un instante t :

$$y(x_1, t) = y(x_2, t); \quad \frac{d}{dx} y(x_1, t) = \frac{d}{dx} y(x_2, t)$$

Utilizando la ecuación (1.14).

$$y_m \text{sen}k(x_1 - vt) = y_m \text{sen}k(x_2 - vt)$$

$$y_m k \text{cos}k(x_1 - vt) = y_m k \text{cos}k(x_2 - vt)$$

Implicando que:

$$k(x_1 - vt) - k(x_2 - vt) = 2n\pi$$

$$x_1 - x_2 = \frac{2n\pi}{k}$$

La menor distancia se produce para $n = 1$, obteniendo la longitud de onda:

$$\lambda = \frac{2\pi}{k} \quad (1.15)$$

Al igual que en el apartado anterior, el número $k = 2\pi/\lambda$ representa el número de longitudes de onda que se dan en la distancia 2π . Se define período T del movimiento ondulatorio, como el tiempo en el que una partícula realiza una vibración completa. Puesto que dicha vibración es armónica simple de pulsación $\omega = kv$, el período es:

$$T = \frac{2\pi}{kv} = \frac{\lambda}{v} \quad (1.16)$$

La frecuencia dada por:

$$f = \frac{1}{T} = \frac{v}{\lambda} \quad (1.17)$$

Con lo anterior (1.14) se puede escribir:

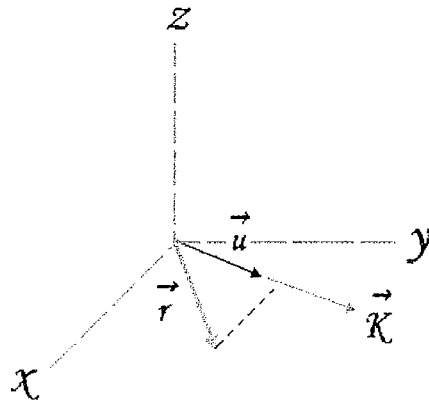
$$y = y_m \text{sen} 2\pi \left(\frac{x}{\lambda} - \frac{t}{T} \right)$$

Por lo que se dice que el movimiento ondulatorio presenta doble periodicidad, una en el tiempo, dada por el período T y otra en el espacio, dada por λ . Estado relacionadas por $v = \lambda/T$.

Hasta el momento se ha tomado la dirección de propagación como el eje x , pero es conveniente definirla independiente de los ejes coordenados de referencia. Se tomara \vec{u} como la dirección de propagación de una onda plana. En las expresiones (1.13) y (1.14) x representa la distancia al centro de perturbación de un punto cualquiera P .

La distancia al origen del punto P , medida en la dirección de propagación, sería la proyección de su vector posición \vec{r} , sobre la dirección de propagación definida por su vector unitario \vec{u} . Con lo que la expresión (1.13) quedara $f(\vec{u} \cdot \vec{r} - vt)$.

Figura 4. **Vector de onda en 3 dimensiones**



Se define el vector $\vec{k} = k\vec{u}$, llamado vector de onda o de propagación, lleva el sentido y dirección de la propagación y módulo k , como se muestra en la figura 4. Este vector permite expresar la ecuación de onda como sigue:

$$y_m \text{sen}(\vec{k} \cdot \vec{r} - \omega t) = y_m \text{sen}(k_x x + k_y y + k_z z - \omega t)$$

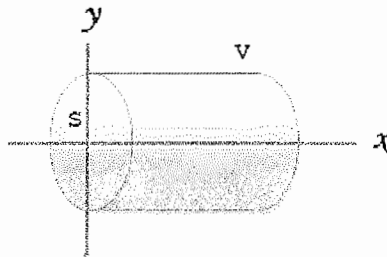
En donde $k_x^2 + k_y^2 + k_z^2 = \frac{4\pi^2}{\lambda^2} = \frac{\omega^2}{v^2}$, contiene las tres coordenadas pero es una onda unidireccional.

Existen otras ondas que se propagan en más direcciones, como lo son las ondas esféricas y cilíndricas.

1.3.1. Intensidad del movimiento ondulatorio

Se entiende como la intensidad del movimiento ondulatorio a la energía que se transmite por unidad de tiempo.

Figura 5. **Volumen alcanzado por una onda plana**



En la figura 5 se considera una onda armónica en un volumen V con sección transversal S , normal a la dirección de propagación. La energía transmitida por unidad de tiempo a través de la sección S es la suma de las que poseen las partículas del volumen de la figura, es decir:

$$\frac{E}{t} = nE_i = n\left(\frac{m\omega^2 y_m^2}{2}\right)$$

Donde E_i es la energía de cada partícula de masa m , al estar realizando movimiento armónico simple. Como $n \cdot m$ es la masa total de las partículas que ocupan el cilindro. Se supone que el cilindro posee densidad homogénea ρ .

$$n * m = vS\rho$$

Por lo que

$$\frac{E}{t} = \frac{\omega^2 y_m^2 v S \rho}{2}$$

Al dividir en la sección transversal se obtiene que:

$$I = \frac{\omega^2 y_m^2 V \rho}{2} = 2v(\pi f y_m)^2 \quad (1.18)$$

1.3.2. Ecuación diferencial del movimiento ondulatorio

La ecuación diferencial que describe el movimiento ondulatorio con velocidad de propagación v

$$\frac{\partial^2}{\partial t^2} y = v^2 \frac{\partial^2}{\partial x^2} y \quad (1.19)$$

La solución de esta ecuación es de la forma $y = f_1(x - vt) + f_2(x + vt)$ y se interpreta como la superposición de dos movimientos ondulatorios que se propagan en la misma dirección y con la misma velocidad pero en sentidos opuestos. Para una onda que se propague en un solo sentido, aparece solamente una de las dos funciones.

1.4. Acústica física

Como ya se ha visto, una perturbación puntual en el aire se propaga en el mismo en forma de pulso u onda viajera y cuando llega al oído produce la sensación de sonido. Si dicha perturbación es periódica, su perturbación es un movimiento ondulatorio longitudinal, que se supondrá propagándose en el eje x , y se considera el medio, por el momento, sin amortiguamiento.

Una onda sinusoidal con frecuencia y amplitud conocida, es la onda sonora más sencilla que existe, siendo ésta un sonido puro que raras veces se encuentra en la práctica. Cualquier perturbación periódica puede descomponerse en suma de sonidos puros. Se estudia por tanto la onda sonora sinusoidal, por ser componente elemental de cualquier otra onda periódica.

En un instante anterior al inicio del movimiento, las partículas de aire se encuentran en reposo y homogéneamente distribuidas. El movimiento perturba la atmósfera que le rodea, esto se traduce en contracciones y dilataciones de volúmenes de aire. Pudiendo asegurar que

- Se modifica la densidad del aire, que en reposo es ρ_0 .
- Se modifica la presión, que en reposo era la atmosférica P_0 , en el sistema ingles $P_0 = 10^5 Pa$.

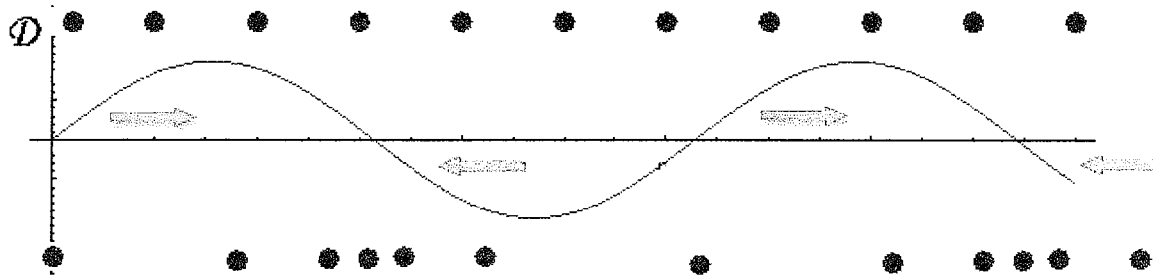
Se denomina presión acústica a la diferencia $p = P - P_0$ y es la presión instantánea resultante en un punto. Análogamente, $\rho - \rho_0$ se denomina densidad dinámica.

Se toma como base la descripción del campo acústico de los desplazamientos producidos en forma de movimiento armónico simple, se expresa como D a los desplazamientos de las partículas

$$D = D_m \text{sen}k(x - vt) = D_m \text{sen}\omega \left(\frac{x}{v} - t \right) \quad (1.20)$$

En la figura 6, se muestra el desplazamiento de las partículas para determinado instante t .

Figura 6. Onda de desplazamientos



1.4.1. Propagación de ondas sonoras

Se considera un elemento diferencial de volumen $dV = dx dy dz$, referido a un sistema ortonormal, en un medio deformable, homogéneo e isótropo. Se parte de la ecuación de la dinámica aplicada a este elemento, suponiendo que la fuerza exterior es la debida a las presiones que el resto del medio ejerce sobre el elemento considerado, así sobre el eje x :

$$\sum dF_x = \frac{\partial}{\partial t} (dm v_x)$$

$$\left(-\frac{\partial}{\partial x} p dx\right) dydz = \frac{\partial}{\partial t} (\rho dx dy dz D'_x)$$

Donde D'_x es la velocidad del desplazamiento que sufren las partículas en la dirección del eje x . Simplificando la última expresión

$$-\frac{\partial}{\partial x} p = \frac{\partial}{\partial t} \rho D'_x$$

Se puede generalizar la última expresión para las otras componentes espaciales, escribiéndola en forma vectorial:

$$-\vec{\nabla} p = \frac{\partial}{\partial t} (\rho \vec{D}') \quad (1.21)$$

Por otra parte, este elemento diferencial debe cumplir la ecuación de continuidad o conservación de la masa, es decir, que la disminución de la masa que sufre dV por unidad de tiempo, debe ser igual a la que fluye por su superficie externa S por unidad de tiempo.

$$-\frac{\partial}{\partial t} \iiint_V (\rho dx dy dz) = \iint_S (\rho \vec{D}') \cdot d\vec{S} \quad (1.22)$$

Aplicando divergencia al segundo término

$$-\frac{\partial}{\partial t} p = \nabla \cdot (\rho \vec{D}') \quad (1.23)$$

Al sustituir esta expresión en la divergencia de (1.21):

$$-\nabla \cdot (\nabla p) = \nabla \cdot \frac{\partial}{\partial t} (\rho \vec{D}') = \frac{\partial}{\partial t} \nabla \cdot (\rho \vec{D}')$$

$$-\nabla^2 p = -\frac{\partial^2}{\partial t^2} \rho$$

$$\frac{\partial^2}{\partial x^2} p + \frac{\partial^2}{\partial y^2} p + \frac{\partial^2}{\partial z^2} p = \frac{\partial^2}{\partial t^2} \rho$$

Esta última expresión, representa la ecuación general de la onda de presiones en función de la densidad del medio. Para facilitar la obtención de los parámetros que intervienen en el campo acústico, se estudia una onda unidireccional propagándose en el eje x . Se parte entonces de la última expresión particularizada para una dimensión:

$$\frac{\partial^2}{\partial x^2} p = \frac{\partial^2}{\partial t^2} \rho \quad (1.24)$$

Se considera un elemento cilíndrico de fluido con sección transversal circular y con área unitaria. Se supondrá, en una propagación, que todas las partículas de una sección transversal tienen el mismo desplazamiento. Se consideran dos secciones A y B separadas una distancia dx , antes de iniciarse el movimiento. Una vez alcanzadas las secciones por la perturbación distarán $dx + dD$. En consecuencia, la masa de fluido que ocupaba un volumen inicial $dV_0 = dx$, pasa a ocupar un volumen $dV = dx + dD$, cumpliéndose que:

$$\rho_0 dx = \rho(dx + dD)$$

Donde ρ es la densidad del fluido perturbado y vale $\rho_0 + \Delta\rho$, al trabajar y despreciar infinitésimos de primer orden, resulta:

$$\Delta\rho = -\rho_0 \frac{\partial}{\partial x} D \quad (1.25)$$

Es posible encontrar una función $P(\rho)$ por medio del desarrollo de la serie de Maclaurin y despreciando infinitésimos de segundo orden en adelante, siendo ésta: $P(\rho) = P_0 + \Delta\rho \left[\frac{dP}{d\rho} \right]_0$

Se puede utilizar el módulo de compresibilidad del medio, $B = -\frac{dP}{dV/V}$ y se sabe que al permanecer constante la masa, la deformación volumétrica unitaria toma el mismo valor, resultando:

$$\frac{dV}{V} = -\frac{d\rho}{\rho}$$

Por tanto

$$\frac{dP}{d\rho} = \frac{B}{\rho}$$

$$\left[\frac{dP}{d\rho} \right]_0 = \frac{B}{\rho_0}$$

Se puede obtener entonces una relación entre presiones y densidades

$$p = P - P_0 = \Delta\rho \frac{B}{\rho_0} \quad (1.26)$$

Sustituyendo en (1.25) el valor $\Delta\rho$ obtenido en (1.26), se obtiene:

$$p = -B \frac{\partial D}{\partial x} \quad (1.27)$$

Esta última relaciona presiones con desplazamientos. Por otra parte, el movimiento del fluido se realiza por la acción de la fuerza resultante $-\left(\frac{\partial}{\partial x} p\right) dx$, aplicando la ecuación de la dinámica

$$\begin{aligned} -\left(\frac{\partial}{\partial x} p\right) dx &= m \frac{\partial^2}{\partial t^2} D \\ -\left(\frac{\partial}{\partial x} p\right) dx &= \rho_0 dx \frac{\partial^2}{\partial t^2} D \\ \frac{\partial}{\partial x} p &= -\rho_0 \frac{\partial^2}{\partial t^2} D \end{aligned} \quad (1.28)$$

Se calcula la segunda derivada, respecto a x de (1.26) y sustituyendo en (1.24), obteniendo la ecuación de propagación de la onda de densidades:

$$\frac{\partial}{\partial x} p \frac{\partial^2}{\partial x^2} \rho = \frac{\rho_0}{B} \frac{\partial^2}{\partial t^2} \rho = -\rho_0 \frac{\partial^2}{\partial t^2} D \quad (1.29)$$

Realizando procedimientos similares, se obtienen las ecuaciones de propagación de presiones y de desplazamientos:

$$\frac{\partial^2}{\partial x^2} p = \frac{\rho_0}{B} \frac{\partial^2}{\partial t^2} p \quad (1.30a)$$

$$\frac{\partial^2}{\partial x^2} D = \frac{\rho_0}{B} \frac{\partial^2}{\partial t^2} D \quad (1.30b)$$

Teniendo en cuenta la ecuación de diferencial (1.19) vista en el apartado anterior, se pone de manifiesto que las tres ondas se propagan con la misma velocidad $c = \sqrt{B/\rho_0}$, denominada velocidad de propagación del sonido o de las ondas sonoras en un medio.

Según (1.20) se puede expresar la onda de velocidad como:

$$\frac{\partial}{\partial t} D = D' = -D_m \omega \cos \omega \left(\frac{x}{c} - t \right) \quad (1.31)$$

La onda de presiones, según la expresión (1.27) es $p = -B \frac{\partial}{\partial x} D$, por lo que

$$p = -B D_m \frac{\omega}{c} \cos \omega \left(\frac{x}{c} - t \right) = \frac{B}{c} D'$$

Teniendo en cuenta que $c = \sqrt{B/\rho_0}$, entonces

$$p = -c \rho_0 D_m \omega \cos \omega \left(\frac{x}{c} - t \right)$$

La onda de densidades según (1.25)

$$\Delta \rho = -\rho_0 \frac{\omega}{c} D_m \cos \omega \left(\frac{x}{c} - t \right) \frac{\partial}{\partial t} D = D' = -D_m \omega \cos \omega \left(\frac{x}{c} - t \right) \quad (1.32)$$

1.4.2. Impedancia Acústica

Se define como impedancia acústica en un punto, al cociente entre la presión acústica en ese punto y la velocidad que la partícula posee en ese instante, $z = p/D'$. Dicha velocidad ocasionada únicamente por la presión acústica. Para una onda plana armónica y progresiva, se consideran las ondas de presión y velocidad, obteniendo que:

$$z = \rho_0 c \quad (1.33)$$

Al producto en la ecuación (1.33) se le conoce como impedancia característica o resistencia del medio, cuya unidad en el sistema inglés es el *Rayl*.

Si se habla de una plana armónica regresiva $z = -\rho_0 c$, es decir, ésta es igual y opuesta a la impedancia característica. En la tabla I, se muestran valores de la impedancia característica para distintos medios, a temperatura ambiente.

Tabla I. **Impedancia característica para distintos medios**

Medio:	c [m/s]	ρ [kg/m³]	ρc Rayl
Aire	340	1.2	408
Agua	1480	1.0	148*10 ⁴
Hormigón	3160	2.3	7*10 ⁶
Ladrillo	3000	1.8	5*10 ⁶
Madera	700	-----	0.4*10 ⁶
Acero	5900	7.8	46*10 ⁶

Fuente: Sancho, Javier y otros, *Acústica Arquitectónica y Urbanística*, p. 79

Si se trata de una onda esférica armónica progresiva, la razón entre la presión acústica y la velocidad de las partículas o impedancia específica, no permanece constante, depende de la distancia r y de la frecuencia angular ω . Es posible encontrar una expresión para la impedancia característica del medio, suponiendo distancias grandes comparadas con la longitud de onda, siendo esta:

$$\frac{p}{D_r'} = \frac{\rho_0 c}{1 - \frac{\tan(kr - \omega t)}{kr}} \quad (1.34)$$

En este caso, cada elemento de la superficie del frente de onda se considera como un elemento de onda plana.

1.4.3. Magnitudes energéticas

Como se ha visto las magnitudes acústicas varían de forma continua con el tiempo y la posición, como consecuencia no se puede tener una medida inmediata de su magnitud, por lo que es necesario promediarla en un determinado período o tiempo de medida. El valor medio de una magnitud A en un tiempo T de promediado, que en acústica se encuentra entre 0.1 a 1 segundo, se expresa:

$$\langle A \rangle = \frac{1}{T} \int_0^T A(t) dt \quad (1.35)$$

Debido a que la magnitud puede ser positiva o negativa, para no obtener subestimaciones del valor, se calcula su valor eficaz

$$A_{ef} = \sqrt{\langle A^2 \rangle} = \sqrt{\frac{1}{T} \int_0^T A^2(t) dt} \quad (1.36)$$

Siempre que no se indique lo contrario, cuando se hable de magnitud acústica variable, se entenderá como valor medio o eficaz según corresponda. Para la presión eficaz se obtiene:

$$p_{ef} = \sqrt{\langle p^2 \rangle} = \sqrt{\frac{1}{T} \int_0^T p^2(t) dt} \quad (1.37)$$

Una onda acústica transporta energía y la magnitud física que la cuantifica en forma vectorial, es la intensidad de la onda acústica. El módulo de dicho vector para cierto punto, como la energía que atraviesa la unidad de área normal a la dirección de propagación, en la unidad de tiempo

$$|\vec{I}| = \frac{dE}{dS \cdot dt}$$

Siendo la unidad en el sistema internacional es $\frac{J}{m^2 \cdot s} = \frac{W}{m^2}$. Para el caso de ondas planas existe una relación sencilla entre el módulo de la intensidad y la presión eficaz.

La potencia media desarrollada por las partículas que se mueven a velocidad \vec{D}' bajo la acción de una fuerza \vec{F} en un tiempo T , es

$$\left\langle \frac{dE}{dt} \right\rangle = \frac{1}{T} \int_0^T \vec{F} \cdot \vec{D}' dt$$

Considerando un área S perpendicular a la dirección de propagación y teniendo en cuenta que la fuerza debida a un campo de presiones $\vec{F} = p\vec{S}$ se obtiene que:

$$\left\langle \frac{dE}{dt} \right\rangle = \frac{1}{T} \int_0^T p\vec{S} \cdot \vec{D}' dt = \frac{S}{T} \int_0^T pD' dt = \frac{S}{T} \int_0^T \frac{p^2}{\rho_0 c} dt$$

$$\left\langle \frac{dE}{dt} \right\rangle = \frac{S}{\rho_0 c} p_{ef}^2$$

$$\langle |\vec{I}| \rangle = \frac{p_{ef}^2}{\rho_0 c} \quad (1.38)$$

La potencia acústica de una fuente, se define como la energía por unidad de tiempo que la fuente transmite al medio en forma de ondas acústicas.

$$W = \frac{E}{t}$$

Para fuentes esféricas, que a cierta distancia, la mayoría se consideran como tal, se posee una relación entre la potencia de la fuente y el módulo de la intensidad. Suponiendo que la energía se distribuye por igual en todas direcciones, puesto que el frente de ondas es mayor, el módulo de la intensidad debe ser menor, por lo que

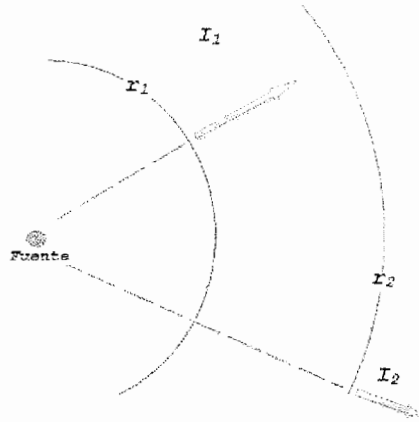
$$W = \frac{E}{t} = |\vec{I}_1|(4\pi r_1^2) = |\vec{I}_2|(4\pi r_2^2)$$

con

$$\frac{|\vec{I}_1|}{|\vec{I}_2|} = \frac{r_2^2}{r_1^2}$$

En la figura 7 se representa la situación anterior

Figura 7. **Intensidad de onda esférica**



También se puede expresar:

$$|\vec{I}| = \frac{W}{4\pi r^2}$$

Las expresiones anteriores muestran, que de no existir degradación de energía acústica (medio no disipativo), la intensidad de una onda plana permanece constante, mientras que para ondas esféricas disminuye con el cuadrado de la distancia. Pero en realidad la propagación del sonido lleva siempre implícita una disipación de energía bajo la forma de desprendimiento de calor y así, la presión queda reducida en el transcurso de la propagación de la onda por un factor exponencial de la forma

$$p(x) = p_0 e^{-\alpha_a(x-x_0)} \quad (1.39)$$

Para una onda plana, y

$$p(r) = \frac{p_1}{r} e^{-\alpha_a(r-r_0)} \quad (1.40)$$

Para una onda esférica. Donde α_a representa la constante de atenuación del medio. Esta atenuación depende de la frecuencia de los sonidos, la temperatura y del grado de humedad del aire en que se propaga. Los sonidos de bajas frecuencias se atenúan menos que los de altas frecuencias, por lo que cuanto más alejado se encuentre un oyente de una fuente, más grave le parece el sonido percibido.

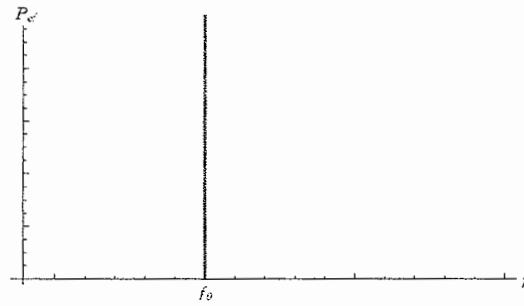
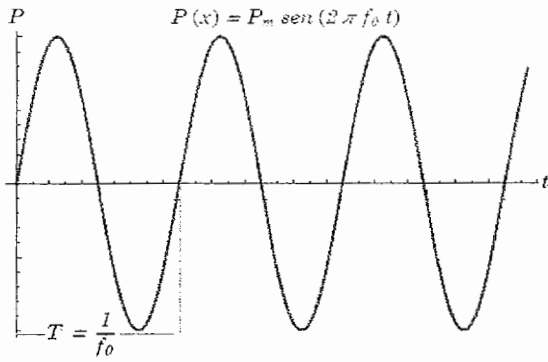
1.4.4. Espectros de frecuencia

El espectro de una señal variante con el tiempo, es la gráfica que relaciona amplitudes o presiones eficaces, en función de todas las componentes frecuenciales del mismo. Más adelante se utilizará el análisis de Fourier para adentrar mejor en el tema.

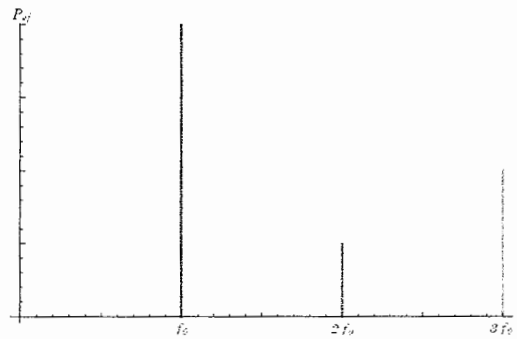
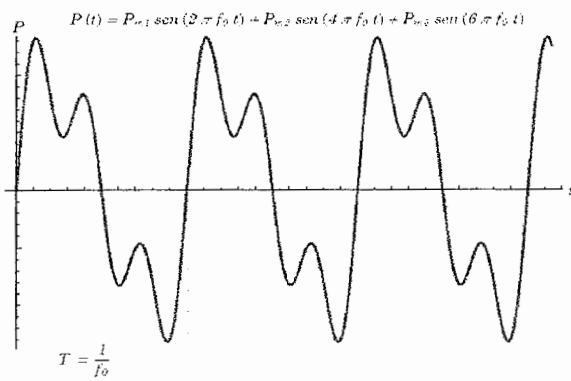
En la figura 8 se muestran tres casos de diagramas espectrales. En el primer caso, figura 8a, se muestra una onda pura, por tener ésta una sola frecuencia, el diagrama se reduce a un solo impulso en dicha frecuencia. En el segundo caso, figura 8b, se observa una señal que posee una frecuencia central y dos múltiplos de esa frecuencia central, denominados armónicos. Para el último caso, figura 8c, se muestra una variación aleatoria de presión, esta variación está constituida por una cantidad muy grande de ondas puras y normalmente se le denomina ruido. Esto da lugar a una normalización de la anchura en todo el rango de frecuencias.

Figura 8. Espectros de a) onda pura, b) onda con dos armónicos, c) ruido

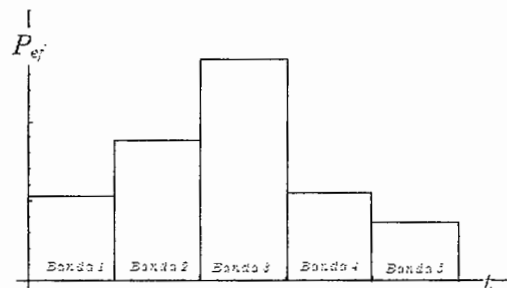
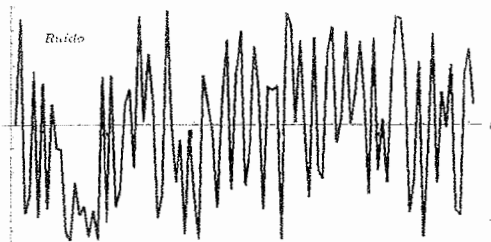
a)



b)



c)



1.4.5. Medición del campo acústico

El mecanismo auditivo humano no responde linealmente a cambios de presión sonora. Siendo la presión mínima correspondiente al umbral de audición humano al orden de $2 * 10^{-5} Pa$ y la máxima de $10^3 Pa$, rango amplio de presiones de difícil manejo. La ley de Weber-Fechner establece que la magnitud percibida es proporcional al logaritmo del estímulo que lo provoca, es decir:

$$L = K * \log(p) \quad (1.41)$$

Los niveles extremos para esta escala son $L_{inf} = K \log(2 * 10^{-5}) = -4.7K$ y $L_{sup} = K \log(10^3) = 3K$. Para obtener una escala que sea positiva, que no se precisen decimales y que el nivel obtenido sea adimensional, se toma como presión de referencia $p_0 = 2 * 10^{-5} Pa$, que es el umbral de audición para sonidos de 1,000Hz en el aire y la constante K igual a 20. Se define así el nivel de presión en decibelios (dB).

$$LP = 20 \log\left(\frac{p}{p_0}\right) = 10 \log\left(\frac{p}{p_0}\right)^2 \quad (1.42)$$

Siendo por tanto los límites $LP_{inf} = 0dB$ y $LP_{sup} = 174dB$. Aunque se pueden percibir sonidos de hasta 174dB, a partir de 120dB se produce una sensación dolorosa, por lo que este valor se ha fijado como el máximo tolerable para el oído humano.

Se define el nivel de intensidad sonora como $LI = 10 \log \frac{|\vec{I}|}{I_0}$, el factor 10 y el factor de la intensidad de referencia I_0 a manera que, para una onda libre y en el aire, coincidan los niveles de presión e intensidad. Por lo que:

$$LI = 10 \log \frac{p^2 / \rho_0 c}{p_0^2 / (\rho_0 c)_{ref}} = 20 \log \frac{p}{p_0} + 10 \log \frac{\rho_0 c}{(\rho_0 c)_{ref}} = LP + 10 \log \frac{\rho_0 c}{(\rho_0 c)_{ref}}$$

Escogiendo como medio de referencial el aire con una impedancia característica $(\rho_0 c)_{ref} = 400 \text{ Rayls}$, $I_0 = p_0 / (\rho_0 c)_{ref}$. por lo que $I_0 = (2 * 10^{-12})^2 / 400 = 10^{-12} \text{ W/m}^2$. Se obtiene por definición

$$LI = 10 \log \frac{|\vec{I}|}{10^{-12}} \quad (1.43)$$

Manteniendo la relación entre LP y LI , válida para ondas libres y utilizando el mismo procedimiento se define el nivel para la potencia W como $10 \log (W / 10^{-12})$ y el nivel de densidad energía acústica $L\rho_E$ igual a $10 \log (\rho_E / 10^{-12})$. Por lo que se puede obtener una relación entre la intensidad sonora y la potencia

$$LI = 10 \log \frac{|\vec{I}|}{10^{-12}} = 10 \log \frac{W / 4\pi r^2}{10^{-12}}$$

$$LI = 10 \log \frac{W}{10^{-12}} - 10 \log 4\pi r^2 \quad (1.44)$$

1.4.6. Disminución de niveles con la distancia

Como ya se ha visto, de no existir degradación de la energía acústica, la onda esférica disminuye su intensidad con el cuadrado de la distancia, siendo la relación entre intensidades y distancias $\frac{|\vec{I}_1|}{|\vec{I}_2|} = \frac{r_2^2}{r_1^2}$, expresando en niveles de intensidad se obtiene:

$$10 \log \frac{|\vec{I}_1|/I_0}{|\vec{I}_2|/I_0} = LI_1 - LI_2 = 10 \log \frac{r_2^2}{r_1^2}$$
$$\Delta LI = 20 \log \frac{r_2}{r_1} \quad (1.45)$$

De modo que para una distancia $r_2 = 2r_1$, existe una pérdida de 6dB en la intensidad, y cada vez que se dobla la distancia a la fuente se pierden 6dB por efecto de la divergencia esférica de las ondas. Si se relaciona ahora la intensidad con la potencia de la fuente se tiene en el medio de referencia

$$|\vec{I}| = \frac{W}{4\pi r^2} \Rightarrow \frac{|\vec{I}|}{10^{-12}} = \frac{W}{4\pi r^2 * 10^{-12}}$$

Para expresar en niveles

$$10 \log \frac{|\vec{I}|}{10^{-12}} = LI = LP \cong 10 \log \frac{W}{10^{-12}} - 20 \log r - 11$$

Si se trata de medios disipativos, la pérdida de nivel debida a la atenuación del medio con la distancia se obtiene utilizando la parte exponencial de la ecuación 1.40

$$\Delta LP_{dis} = 20 \log e^{-\alpha_a(r_1-r_2)} = 8.7\alpha_a(r_2 - r_1)$$

La cantidad $8.7\alpha_a$ representa el número de dB perdidos por metro recorrido, a causa de pérdidas energéticas del medio. La pérdida total de potencia será:

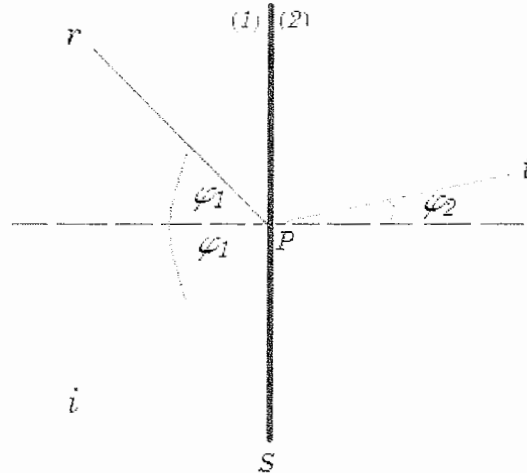
$$\Delta LI = \Delta LP = 20 \log \frac{r_2}{r_1} + 8.7\alpha_a(r_2 - r_1) \quad (1.46)$$

1.4.7. Reflexión y transmisión de ondas acústicas

Cuando una onda encuentra en su propagación una superficie de separación entre dos medios distintos se originan dos nuevas ondas, una reflejada en el primer medio y una onda transmitida en el segundo medio. Esto se ejemplifica en la figura 9. Siendo S la superficie de separación de los medios con impedancias características z_1 y z_2 respectivamente. Según la ley de Snell, para cualquier tipo de onda, rigen los fenómenos de reflexión y transmisión desde el punto de vista geométrico, deduciendo que el ángulo de incidencia y de reflexión son iguales y el de incidencia y el de refracción guardan la relación:

$$\frac{\text{sen}\varphi_1}{c_1} = \frac{\text{sen}\varphi_2}{c_2}$$

Figura 9. Transmisión y reflexión de ondas acústicas



Los puntos de la superficie S y en particular P , deben cumplir con las condiciones de contorno por ser puntos frontera entre ambos medios. Se expresa la continuidad de las presiones, la presión que el medio 1 ejerce sobre P tendrá que ser igual a la que ejerce el medio 2, de esta manera $p_i + p_r = p_t$ que también puede expresarse en función de las velocidades de las partículas de cada onda, teniendo en cuenta que las ondas incidente y transmitida son progresivas y por tanto en ellas $p = zD'$, y la onda reflejada es regresiva $p = -zD'$, por lo que se puede expresar:

$$z_1 D'_i - z_1 D'_r = z_2 D'_t$$

$$z_1 (D'_i - D'_r) = z_2 D'_t \quad (1.47)$$

Para las condiciones de contorno, también debe cumplirse la continuidad de la velocidad, de la partícula P , normal a la superficie de incidencia, resulta que:

$$\cos\varphi_1 D'_i - \cos\varphi_1 D'_r = \cos\varphi_2 D'_t \quad (1.48)$$

Se incluye $R_{D'} = \frac{D_{r'}}{D_i'}$ y $T_{D'} = \frac{D_t'}{D_i'}$ en las dos ecuaciones anteriores, obteniendo que

$$z_1(1 - R_{D'}) = z_2 T_{D'}$$

$$(1 - R_{D'})\cos\varphi_1 = T_{D'}\cos\varphi_2$$

Sistema resuelto para $R_{D'}$ y $T_{D'}$, da

$$R_{D'} = \frac{z_1 \cos\varphi_2 - z_2 \cos\varphi_1}{z_1 \cos\varphi_2 + z_2 \cos\varphi_1}$$

$$T_{D'} = \frac{2z_1 \cos\varphi_1}{z_1 \cos\varphi_2 + z_2 \cos\varphi_1}$$

Los resultados anteriores son los valores de reflexión y transmisión respectivamente dan un índice de la magnitud correspondiente a cada fenómeno. En el caso particular de incidencia normal $\varphi_1 = 0$ y por tanto $\varphi_2 = 0$ se obtiene:

$$R_{D'} = \frac{z_1 - z_2}{z_1 + z_2} \quad \text{y} \quad T_{D'} = \frac{2z_1}{z_1 + z_2}$$

Con lo que se puede concluir que:

- A mayor diferencia entre impedancias de ambos medios, mayor reflexión
- Si ambas impedancias son iguales, no existe reflexión y la transmisión es total

Ahora se analiza el balance entre energías, $E_i = E_r + E_t$, si se divide entre E_i , se obtiene:

$$1 = \frac{E_r}{E_i} + \frac{E_t}{E_i} = R_E + T_E$$

Donde R_E y T_E representan los coeficientes de reflexión y transmisión, respectivamente y representan las fracciones de energía reflejada y transmitida del medio 1 al 2. Para cuantificar este fenómeno en función de ángulos de incidencia y de transmisión, se establece el balance de energía por unidad de tiempo y superficie, de modo que:

$$I_i \cos \varphi_1 = I_r \cos \varphi_1 + I_t \cos \varphi_2 \quad (1.49)$$

Los coeficientes toman los valores $R_E = I_r/I_i$ y $T_E = I_t \cos \varphi_2 / I_i \cos \varphi_1$, que de acuerdo con $\rho_E = dE/dv$ y que $D'_m = D_m$, se obtienen los coeficientes en funciones de los desplazamientos de las ondas

$$R_E = \frac{\frac{1}{2} \rho_1 c_1 \omega^2 D_{mr}^2}{\frac{1}{2} \rho_1 c_1 \omega^2 D_{mi}^2} = \left(\frac{D'_{mr}}{D'_{mi}} \right)^2 \quad (1.50)$$

$$T_E = \frac{\frac{1}{2} \rho_1 c_1 \omega^2 D_{mt}^2 \cos \varphi_2}{\frac{1}{2} \rho_1 c_1 \omega^2 D_{mi}^2 \cos \varphi_1} = \frac{z_2 \cos \varphi_2}{z_1 \cos \varphi_1} \left(\frac{D'_{mt}}{D'_{mi}} \right)^2 \quad (1.51)$$

En particular para una incidencia normal se obtiene que:

$$R_{En} = \left(\frac{z_1 - z_2}{z_1 + z_2} \right)^2 \quad T_{En} = \frac{4z_1 z_2}{(z_1 + z_2)^2}$$

1.5. Ultrasonido

El mecanismo auditivo humano se compone de un órgano de toma de datos -oído externo y medio-, un órgano de conversión analógica a digital -oído interno- y un sistema de memorias -cerebro-. Para que la onda acústica se transforme en sensación de sonido, es necesario que la frecuencia de las variaciones de presión esté comprendida en una determinada banda y que la amplitud de éstas sea superior a cierto valor para cada frecuencia.

El mecanismo humano de audición es capaz de percibir señales acústicas comprendidas entre 20Hz y los 20KHz, aproximadamente, aunque estos límites varían según la persona. Las ondas acústicas que se encuentran por debajo de los 20Hz se denominan infrasonidos y las ondas acústicas que se encuentran sobre los 20KHz se denominan ultrasonidos. La razón por la que el sistema auditivo humano no percibe los ultrasonidos, se debe al oído medio que funciona como filtro pasa bajo.

Los ultrasonidos son utilizados por algunos animales, como los murciélagos, ballenas y delfines, en forma de radar para su orientación, esto se conoce como ecolocalización. Lo que indica que algunos animales poseen un límite superior de frecuencia al de los humanos.

1.6. Aplicaciones del ultrasonido

Tanto los infrasonidos y ultrasonidos cumplen con todos los parámetros establecidos en las secciones anteriores, esto porque son ondas acústicas. A pesar que ambas ondas no son escuchadas por los humanos, el ultrasonido posee ciertas ventajas sobre el infrasonido, que ha hecho posibles diversas aplicaciones. Posiblemente la ventaja más notoria, es que posee un ancho de

banda mucho mayor que el infrasonido. La segunda es, que según la expresión (1.10) a mayor frecuencia mayor energía y en consecuencia mayor potencia, sin necesitar aumentar demasiado la amplitud de la perturbación del medio en el que se propaga la onda.

También se puede mencionar que a mayor frecuencia, la longitud de onda λ es menor, por lo que se puede utilizar en distancias reducidas, comparadas con las longitudes de onda grandes de los infrasonidos por poseer bajas frecuencias.

Una técnica muy común en diversas aplicaciones, consiste en ingresar una onda ultrasónica en un medio y observar la reflexión de la misma, esto se convierte en una muy buena fuente de detalles de la estructura interna de dicho medio, la aplicación más conocida de esta técnica es la ecografía, utilizada para producir imágenes del feto dentro del útero de la madre. A continuación se especifica más esta y otras aplicaciones en diversas áreas.

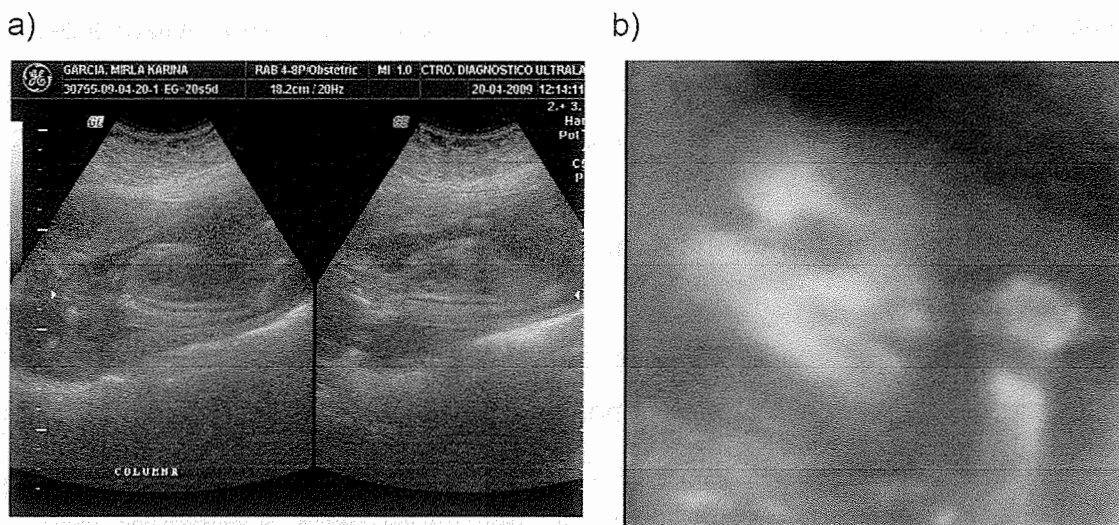
1.6.1. Aplicaciones médicas

La ecografía médica se ha utilizado por más de 50 años y es una de las herramientas de diagnóstico médico más utilizada en la actualidad. La ecografía es un método basado en ultrasonido para crear imágenes de músculos, tendones y muchos órganos internos, la estructura éstos y cualquier lesión patológica en tiempo real.

La ecografía obstétrica es de uso común durante el embarazo. Es un método más barato y portátil comparado con la resonancia magnética y la tomografía computarizada, la ecografía no conlleva riesgos conocidos para el paciente, esto dado a que no utiliza radiaciones ionizantes, que impone riesgos como cáncer y rupturas cromosómicas en el paciente. Sin embargo posee dos posibles efectos debido a la energía del ultrasonido, puede aumentar la respuesta de inflamación y calentar los tejidos blandos.

En la figura 10a se muestra una ecografía obstétrica en dos dimensiones y en la figura 10b se muestra una ecografía 3D, se ha dado el término ecografías 4D (3D más tiempo).

Figura 10. **Ecografía obstétrica a) 2D, b) 3D**



Fuente: centro de diagnóstico ultrasónico, Guatemala. V730_090420_1318.

Existe un proceso denominado ultrasonido de alta intensidad (*HIFU*) utilizado para el tratamiento de tumores, también se le conoce como cirugía de ultrasonido focalizado (*FUS*). Para este procedimiento se utilizan frecuencias entre 250KHz a 2,000KHz, que son menores a las utilizadas en diagnóstico médico, pero son aplicadas por mayor tiempo y a mayor intensidad promedio, a menudo son guiadas por imágenes de resonancia magnética.

Existe un procedimiento llamado suministro de fármacos a través de acústica dirigida (*ATDD*), que consiste en la unión de quimioterapia con ultrasonido de alta frecuencia, entre 1MHz a 100MHz y rangos de intensidades entre 0 a 20v/cm². La energía acústica se concentra en la matriz del tejido de interés para hacerla más permeable a las drogas terapéuticas.

La ecografía también se utiliza en dermatología, para establecer existencia y dimensiones de tumores cutáneos, de esa manera delimitar límites quirúrgicos. El ultrasonido pulsado de baja intensidad se utiliza para el tratamiento y limpieza dental y para la regeneración del tejido óseo.

1.6.2. Aplicaciones industriales

Para determinar fallas en materiales como plástico, metal y compuestos aeroespaciales, se utilizan ensayos no destructivos, que analizan éstos sin causarles daños, también se utilizan para medir el grosor de los objetos. Las frecuencias para estas pruebas se encuentran entre los 2MHz y 10MHz.

También se pueden inspeccionar materiales menos densos como la madera y el concreto utilizando frecuencias más bajas, entre los 50KHz y 500KHz.

En inspección eléctrica, el ultrasonido se utiliza para detección de efecto corona, descarga parcial y arcos eléctricos. También se utiliza para localizar fugas de líquidos en tuberías o pérdidas de presión o vacío en líneas presurizadas. Para detección de rodamientos mecánicos dañados en bandas transportadoras.

1.6.3. Otras aplicaciones

Un uso común del ultrasonido es el sonar, éste consiste en emitir un pulso ultrasónico en determinada dirección y de encontrar un objeto en dicha dirección, parte de la onda es reflejada de vuelta hacia el emisor. Midiendo el tiempo que tarda en regresar la reflexión se obtiene información de que tan lejos se encuentra el objeto, de esta forma el sonar se puede utilizar como sensor de movimiento. También emitiendo en determinado sitio, y en muchas direcciones los pulsos ultrasónicos, se puede construir un mapeo del sitio en cuestión, esta técnica es la ecolocalización. Últimamente esta técnica se ha empleado para la construcción de sistemas de visión artificial aplicados a robótica y en sistemas de inspección de procesos automatizados.

2. TRANSDUCTORES ULTRASÓNICOS Y TÉCNICAS DE MEDICIÓN

En el presente capítulo se describen algunos métodos utilizados en la generación de ondas ultrasónicas, comprendiendo dispositivos mecánicos, el efecto magnetoestrictivo y el efecto piezoeléctrico, haciendo énfasis en este último por la actual variedad de sensores ultrasónicos de este tipo y la fácil obtención de los mismos. Del mismo modo, se proporcionan clasificaciones de los sensores ultrasónicos dependiendo de su funcionamiento y el comportamiento del campo radiado por éstos.

Además, se detallan diversas técnicas de medición y las posibles maneras de representación de mediciones de ultrasonido, necesarias para las pruebas descritas en capítulos posteriores.

2.1. Generación de ultrasonido

Existen muchos métodos para la generación de ondas ultrasónicas, en principio, se aplican los mismos utilizados para la generación de ondas audibles con procedimientos mecánicos, consistentes en la oscilación de dispositivos a frecuencias correspondientes a las ultrasónicas. El inconveniente de estos métodos, es que los dispositivos poseen límites de oscilación, inferiores a los necesarios para muchas pruebas, como los ensayos no destructivos y de diagnóstico médico.

Otro método consiste en utilizar el efecto magnetostrictivo utilizando la propiedad de algunos materiales ferromagnéticos, como el níquel y el acero, de contraerse o expandirse por efecto de campo magnético. Este método también se puede utilizar de manera inversa, cambiando la compresión o expansión del material produciendo una variación de campo magnético. El efecto magnetostrictivo se aprovecha para la producción de ultrasonidos altas potencias a frecuencias relativamente bajas. Un método de mayor importancia por su aprovechamiento en diversas áreas es el efecto piezoeléctrico descrito a continuación.

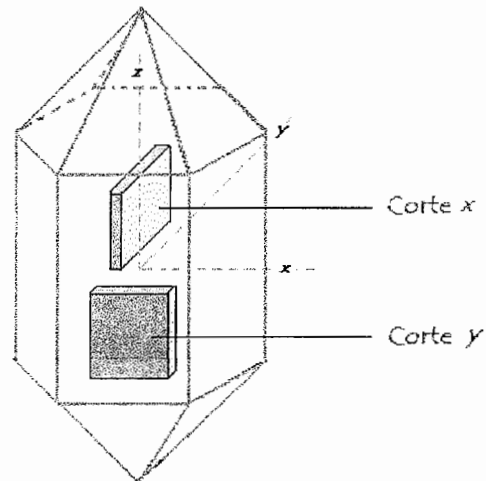
2.1.1. Efecto piezoeléctrico

El efecto piezoeléctrico es un fenómeno presente en algunos cristales que al aplicarles variaciones de presión mecánica, adquieren una polarización eléctrica, dando lugar a una diferencia de potencial y de cargas eléctricas en su superficie. Éste al igual que el efecto magnetostrictivo, es reversible, cuando se aplica una carga eléctrica en la superficie del cristal, ésta se contrae o se expande dependiendo del signo de la carga. Por lo que una aplicación eléctrica alterna aplicada en la superficie del cristal, da origen a oscilaciones mecánicas de presión, cambios periódicos de expansiones y contracciones.

El efecto no es propio de todos los cristales y también se da solo bajo ciertas condiciones. El más utilizado es el cuarzo, éste posee una forma determinada, siempre recurrente descrita por los ejes cristalográficos, designados como x , y y z al igual que un sistema de coordenadas cartesianas, como el mostrado en la figura 11.

El efecto deseado se consigue únicamente si la lámina a utilizar es cortada del cristal perpendicularmente al eje x o y . En el primer caso, la deformación mecánica poseerá la misma dirección que el campo eléctrico y en el segundo caso perpendicularmente a éste, denominando a las diferentes oscilaciones de corte x o de corte y . Los cuarzos de corte x generan ondas longitudinales, mientras que los otros generan ondas transversales. En la mayoría de casos se utiliza el de corte x .

Figura 11. **Representación de cuarzo con cortes x e y**



A pesar que existen otros cristales útiles para la piezoelectricidad, como la Turmalina, el cuarzo fue el primer material utilizado, por presentar características muy estables en frecuencia. Los transductores de cuarzo presentan la desventaja de generar poca energía en las ondas acústicas, por lo que es suplantado por otros materiales sintéticos.

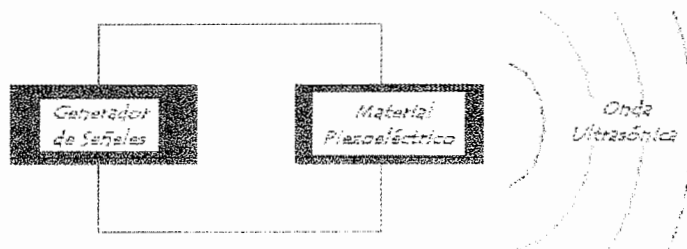
Las cerámicas sintéticas son más eficientes para producir energía, la desventaja de éstas radica en su tendencia al desgaste, de cualquier manera estos últimos compuestos han ido abarcando un mayor campo en la creación de sensores ultrasónicos. Dichas cerámicas adquieren su piezoelectricidad por polarización, esto es un enfriamiento desde una alta temperatura característica del material con exposición a altos voltajes eléctricos.

El efecto piezoeléctrico puede perderse en estos materiales por despolarización cuando se excede la temperatura de Curie, temperatura por encima de la cual un cuerpo ferromagnético pierde su magnetismo. Entre los cerámicos sintéticos se pueden mencionar al titanio de bario, el sulfato de litio, el zirconato de plomo titanio y el metanobiato de plomo.

2.1.2. Emisión y recepción de ultrasonido

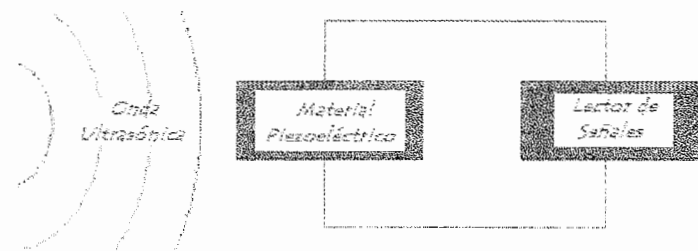
El principio de generar ultrasonido aprovecha el efecto piezoeléctrico inverso. El material piezoeléctrico, que se encuentra en el medio en donde viajará la onda acústica, recibe una señal eléctrica que lo hace oscilar a la frecuencia ultrasónica deseada. En la figura 12 se muestra el diagrama de bloques de este principio.

Figura 12. Diagrama de bloques de un generador ultrasónico



De manera similar se pueden captar las ondas ultrasónicas, cuando existen oscilaciones en un medio físico en donde se encuentre un material piezoeléctrico, éste genera una señal eléctrica proporcional al fenómeno percibido como se muestra en la figura 13.

Figura 13. **Diagrama de bloques de un receptor ultrasónico**



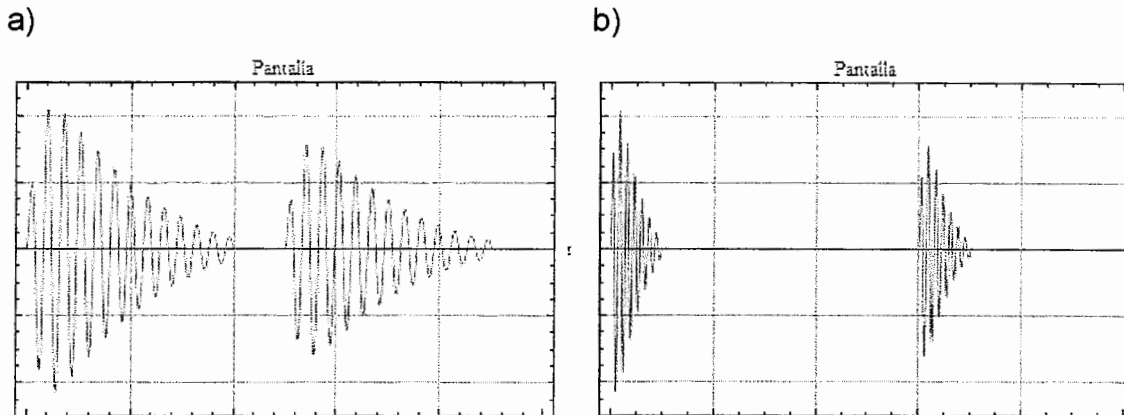
2.2. Transductores ultrasónicos

La mayoría de transductores o palpadores ultrasónicos operan bajo el efecto piezoeléctrico. Cuando el material piezoeléctrico es excitado por una señal eléctrica alterna y de corta duración, la respuesta de la oscilación del cristal decae lentamente dependiendo de su propia frecuencia resonante. Dicha frecuencia fundamental f_0 se relaciona con la velocidad v a la que se propaga el ultrasonido en el material y el grosor del cristal d :

$$f_0 = \frac{v}{2d} \quad (2.1)$$

La figura 14a representa una medición ultrasónica utilizando un cristal débilmente atenuado y en la figura 14b, una medición con un cristal altamente atenuado. Se utilizan cristales altamente atenuados para dar lugar a mayores intervalos de tiempo entre mediciones y evitar traslape entre éstas

Figura 14. **Respuesta de un cristal a) débilmente atenuado, b) altamente atenuado**

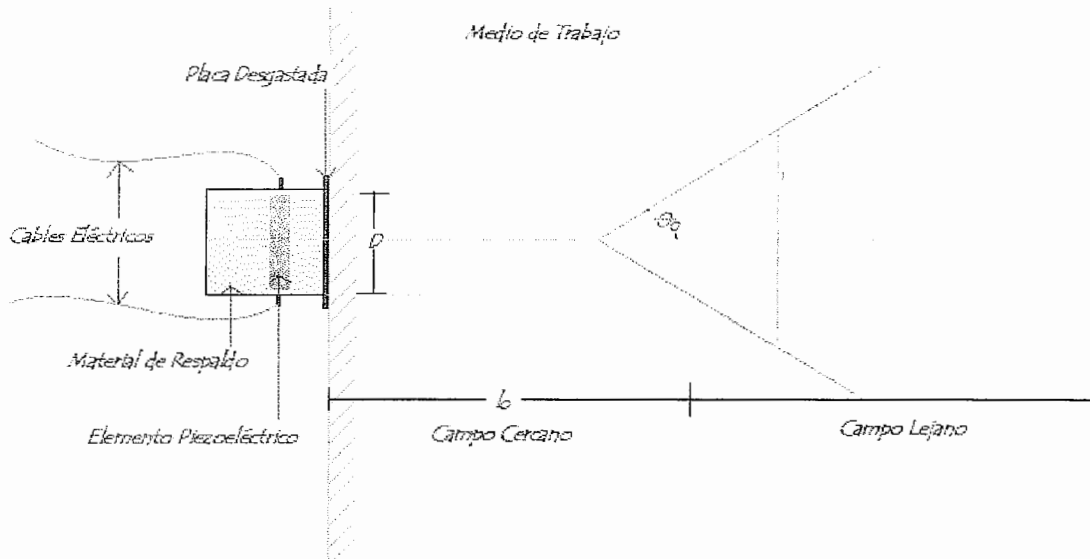


El término resolución hace referencia a la capacidad del transductor de indicar dos discontinuidades muy próximas entre sí. Con transductores débilmente atenuados se obtienen pulsos anchos en un grado de resolución menor, causando que la potencia y la sensibilidad sean altas. Con transductores altamente atenuados se obtiene buena resolución pero la potencia y sensibilidad son decrecientes. La resolución del transductor también depende en gran parte del material de construcción. Actualmente se encuentran transductores construidos específicamente para trabajar con altas potencias o para alta resolución.

2.2.1 Radiación del campo ultrasónico

Al área de influencia de las vibraciones causadas por un transductor se conoce como campo acústico, que en este caso consiste en un campo ultrasónico. En la figura 15 se muestran las partes de un transductor básico y el campo de radiación ideal producido por éste.

Figura 15. **Campo radiado por un transductor ultrasónico**



Se denomina campo cercano al área de rayos que se propagan paralelos al eje de propagación y campo lejano al área de rayos que se abren cónicamente. La longitud l_0 para un transductor circular se puede calcular por medio de:

$$l_0 = \frac{D^2 f}{4v} \quad (2.2)$$

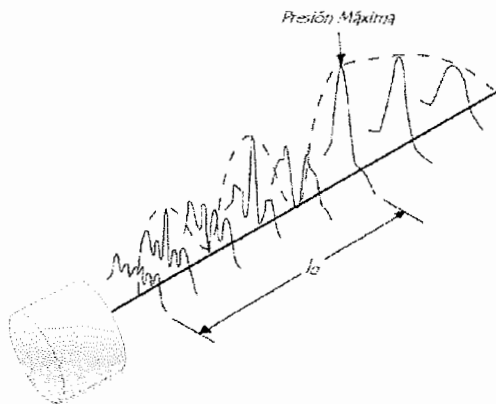
En donde D es el diámetro efectivo del cristal, f es la frecuencia ultrasónica y v la velocidad de la onda a la que se propaga en el medio. Cuando no se conoce D se calcula la longitud del campo cercano experimentalmente.

En la figura 16a se muestra que la presión sónica en el campo cercano, sufre muchas variaciones. Para l_0 se obtiene la mayor presión sónica, y disminuye conforme la distancia de propagación aumenta, en esta región el haz ultrasónico se dispersa cónicamente. En la figura 16b se muestra una respuesta típica de la presión dependiendo de la distancia en el eje de propagación.

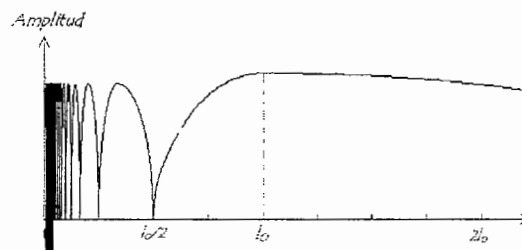
En el momento de realizar inspecciones, se trabaja con el campo lejano o zona de Fraunhofer. Como se puede ver en la figura 16, después de sobrepasar l_0 ocurre una difracción ultrasónica los haces laterales desviándose respecto al eje de propagación, formando un ángulo θ , conocido como ángulo de divergencia del haz, al ángulo que se forma entre los haces laterales, se conoce como ángulo de propagación del haz, y equivale a el doble del ángulo de divergencia.

Figura 16. **a) Intensidades de presión en el campo sónico, b) Amplitud de presión en función de la distancia al transductor**

a)

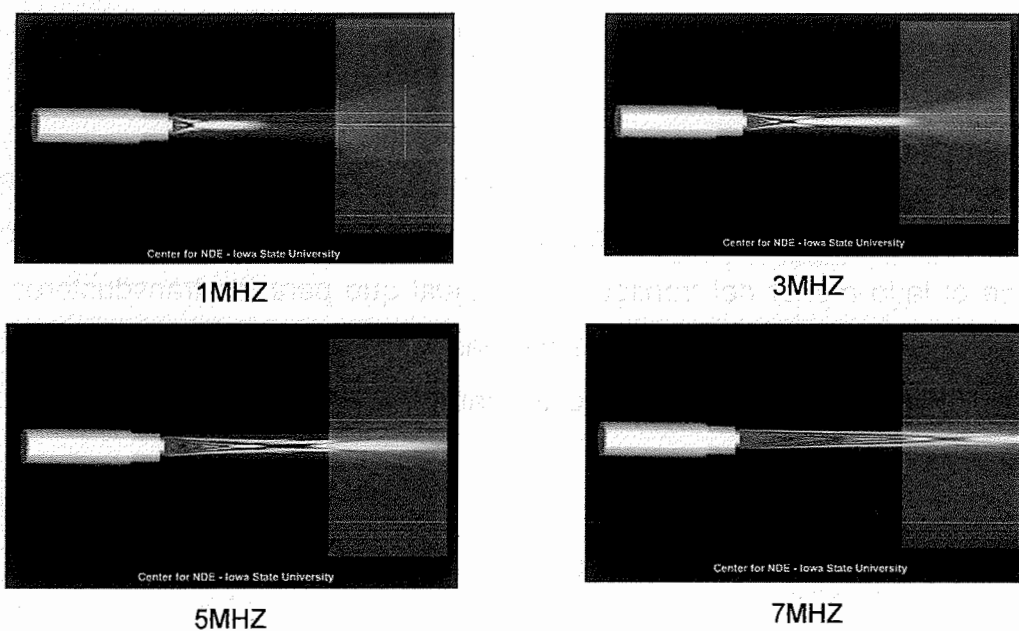


b)



En la figura 17, se muestra una secuencia de haces a diferentes frecuencias para un transductor de 0.75 pulgadas de diámetro. Se observa que el ángulo de divergencia es mayor para frecuencias bajas. De la misma manera, si el diámetro del transductor aumenta, el ángulo de propagación disminuye.

Figura 17. Propagación de haces a diferentes frecuencias



Fuente: *Center for NDE, IOWA State University*, Disponible en Web: <<http://www.ndt-ed.org/EducationResources/CommunityCollege/Ultrasonics/EquipmentTrans/beamspread.htm>

> noviembre 2009

Es posible encontrar una relación entre el ángulo de divergencia y los parámetros del transductor y la onda ultrasónica. Dicho ángulo es una medida desde el eje de propagación, hasta un punto en la dispersión en donde la presión decae a la mitad (a -6dB) de la presión existente en el eje de propagación.

$$\text{sen}\left(\frac{\theta}{2}\right) = \frac{0.514v}{Df} \quad (2.3)$$

Para los transductores rectangulares, con un lado menor a y lado mayor b , se puede calcular la longitud del campo cercano por medio de:

$$l_0 = \frac{a^2 + b^2}{\pi \lambda} \left(1 - \frac{2a}{b}\right) = \frac{f(a^2 + b^2)}{\pi v} \left(1 - \frac{2a}{b}\right) \quad (2.4)$$

Para este caso, la sección transversal del campo no es circular, y cada lado del rectángulo produce un ángulo de divergencia, siendo mayor el que produce el lado menor del transductor. Al igual que para los transductores de sección circular, es posible determinar las dimensiones de un transductor rectangular, en base a una longitud de campo cercano deseada en pruebas experimentales.

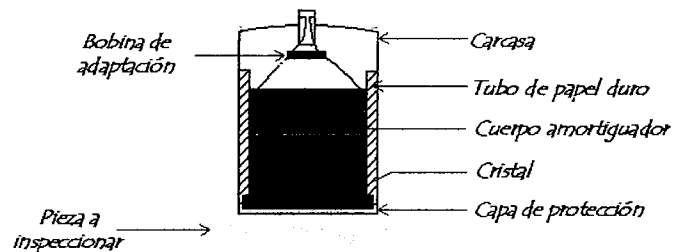
2.2.2. Tipos de transductores

Los transductores ultrasónicos son manufacturados para diversas aplicaciones, es posible clasificarlos en dos grupos, transductores de contacto y transductores de inmersión. El primer grupo incluye a transductores usados en contactos directos sobre la superficie de la región a inspeccionar, estos se encuentran protegidos por una carcasa para resistir el deslizamiento sobre diversos materiales. En el momento de utilizar estos, es necesario utilizar agua, grasas, aceites o materiales comerciales para el acoplamiento, esto con el fin de eliminar el espacio de aire entre transductor y el material a inspeccionar.

Los transductores de inmersión están diseñados para operar en entornos líquidos y todas sus conexiones son impermeabilizadas. Por lo regular poseen una capa de acoplamiento de impedancia que les ayuda a obtener mayor energía acústica. Algunos cuentan con lentes planos, esféricos o cilíndricos de enfoque, pudiendo mejorar la sensibilidad y resolución debido a la concentración de energía en un área menor. A continuación se describen las configuraciones más importantes en el uso de los transductores de contacto:

- **De incidencia normal:** su nombre se deriva de la dirección en que las ondas ultrasónicas viajan en la pieza analizada, esta dirección es perpendicular a la superficie de incidencia. El cristal se supone, por el momento de corte x , generando ondas longitudinales y posee las caras metalizadas para la recepción de pulsos eléctricos. Una de estas caras se pega al cuerpo amortiguador, la otra cara puede estar en contacto directo con el material de ensayo o cubierta con una capa de material plástico o cerámico. Este transductor se muestra en la figura 18

Figura 18. Transductor normal y sus partes



Las características del tamaño del transductor dependen de las propiedades de amortiguamiento o atenuación. Los cables de conexión pasan por una pequeña bobina de acople eléctrica entre el emisor de pulsos y el material piezoeléctrico. El cristal, el elemento amortiguador y

la bobina en conjunto se denominan inserto o *nob* y van dentro de una carcasa metálica conectada a tierra

- **De doble cristal:** es una variante del transductor de incidencia, combinando dos sistemas completos dentro de una sola carcasa. Un sistema se utiliza como emisor y el otro como receptor, estos se encuentran separados por una barrera acústica evitando la transmisión entre ellos dentro de la carcasa. Una ventaja de este transductor, es que puede trabajar dentro del campo cercano de propagación. Utilizando cristales adecuados para cada función, por ejemplo titanio de bario como emisor y sulfato litio como receptor
- **De incidencia normal corte y:** la construcción de los transductores incidentes de corte *y* es la misma que para los de corte *x*. La diferencia entre estos es la orientación cristalográfica y por tanto la orientación de las ondas. Estos generan ondas cortantes, razón por lo que es imposible utilizar líquidos como acople. Dicho acople se puede llevar a cabo con materiales rígidos, por presión firme del transductor sobre la superficie utilizando una goma protectora por debajo del cristal, o por acople sobre la pieza con un agregado de cera de abejas
- **De incidencia angular:** consisten básicamente en un cristal montado sobre una cuña de plexiglás, típicamente usados para transmitir una onda cortante en el medio a inspeccionar. Se pueden obtener en diferentes configuraciones de ángulos o en versiones ajustables, para decidir el ángulo de difracción y reflexión de las ondas. Estos transductores son utilizado para inspeccionar metales como el acero y pueden generar ondas superficiales para detectar desperfectos en la superficie de los materiales

2.3. Técnicas de medición

Existen dos técnicas de medición, técnica de transmisión y técnica de pulso-eco. El fundamento teórico de estas técnicas consiste analizar el comportamiento de las ondas viajeras incidentes en un material con impedancia acústica diferente a la del primer material, tal como se describe en el capítulo 1, sección 1.5. El resultado es una onda refractada en el primer material y otra onda transmitida en el segundo.

En los procedimientos de ensayos ultrasónicos, la onda incidente consiste en una onda ultrasónica y el cambio de impedancia puede estar comprendido por diversos factores, entre los cuales se pueden encontrar un defecto dentro de un material a prueba de ensayo no destructivo o el cambio de impedancia entre líquido amniótico y un feto en el procedimiento de ecografías obstétricas. Con el análisis de las ondas ultrasónicas transmitidas y reflejadas, se inician técnicas de medición del ultrasonido.

2.3.1. Técnica de transmisión

Esta técnica consiste en el análisis de la onda transmitida en el medio de interés. Se utiliza principalmente en ensayos no destructivos en materiales. En un extremo de la pieza a analizar se coloca un emisor ultrasónico y al extremo opuesto, el receptor. En caso de un defecto en el material (cambio en impedancia acústica), la intensidad sonora del receptor disminuye en relación a la del emisor debido a una reflexión parcial en el medio de propagación o se hace cero en el caso de una reflexión total en el medio.

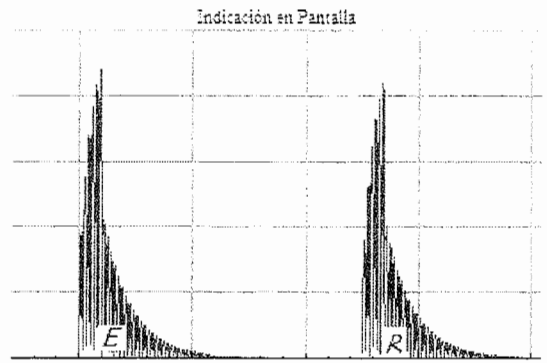
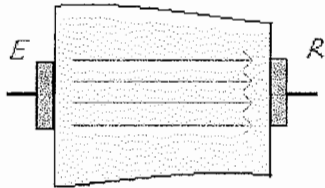
Sin importar la forma de la señal transmitida, el receptor debería captar la señal emitida, ya que se encuentran aislados con el emisor. Un inconveniente de esta técnica, es que resulta imposible detectar la profundidad a la que se encuentra el defecto en la pieza de ensayo.

En el recorrido de la onda se presentan dos zonas o etapas de acoplamiento, la zona de transición entre el emisor y la pieza y otra entre la pieza y receptor, estas zonas influyen sobre la intensidad en el receptor. Es necesario que el emisor y receptor se encuentren alineados exactamente, para que no existan mediciones incorrectas.

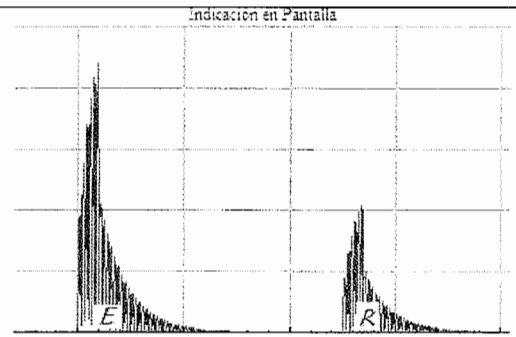
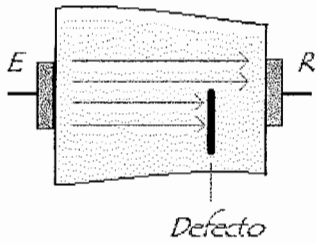
En la figura 19 se observan diferentes situaciones al utilizar esta técnica. Las señales de emisor y receptor se muestran en una misma pantalla y separadas por el tiempo necesario para que la onda atravesase la pieza de ensayo.

Figura 19. Representación para técnica de transmisión

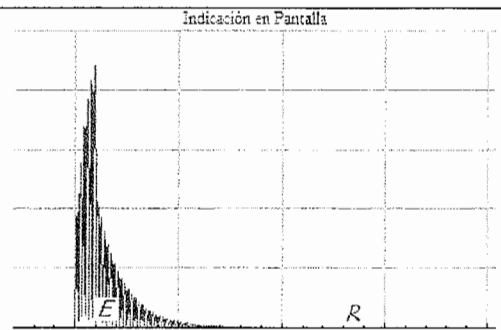
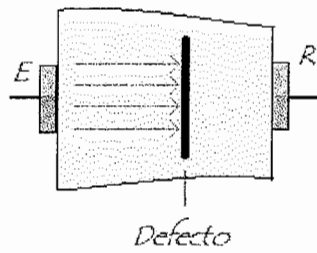
Caso 1



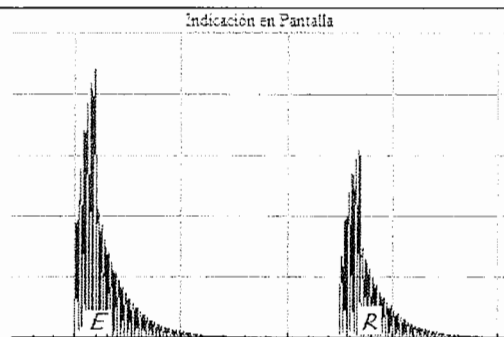
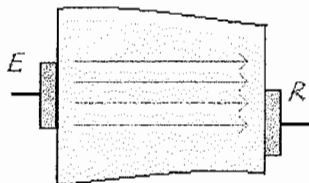
Caso 2



Caso 3



Caso 4



2.3.2. Técnica de pulso eco

También se conoce como técnica de pulsos o impulsos reflejados, utilizando la porción de onda reflejada para calcular la distancia hasta un cambio de impedancia acústica. Esta técnica es utilizada en diversas áreas para ensayos no destructivos, en mediciones de distancias, sistemas de localización, creación de imágenes obstétricas, estimación de profundidad de océanos, entre otros.

En este procedimiento el cristal funciona como emisor y receptor o es posible utilizar un transductor de doble cristal. Un tren de pulsos eléctricos de muy corta duración genera una análoga onda ultrasónica en el transductor, un instante después el mismo oscilador está listo para funcionar como receptor. La onda penetra el material hasta que como resultado de una superficie límite, tiene lugar a una recepción total o parcial. En la mayoría de aplicaciones el tiempo t que tarda en ir y regresar la onda desde y hasta el transductor es del orden de microsegundos o menor.

Sí se supone que el cambio de impedancia se encuentra a una distancia d del transductor y la onda viaja a una velocidad v en el material, se puede obtener la relación:

$$2d = vt \quad (2.5)$$

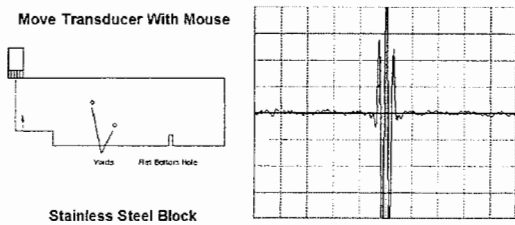
En la figura 20 se representa una simulación de los ecos en un osciloscopio, para pruebas realizadas en el estudio de una pieza de acero inoxidable, en el centro de evaluación no destructiva de la Universidad Estatal de Iowa. La señal fue trabajada con un programa computacional y se utilizó un transductor de 5MHz con un diámetro de 0.25 pulgadas.

Para el primer caso, se observa en el osciloscopio la onda reflejada desde el extremo opuesto de la pieza. En el caso 2, se observa que la reflexión ocurre en un tiempo más prolongado que en el caso anterior, debido a que la pieza es más ancha en esa región, por lo que la señal en el osciloscopio se ve corrida hacia la derecha, indicando mayor tiempo de ir y vuelta.

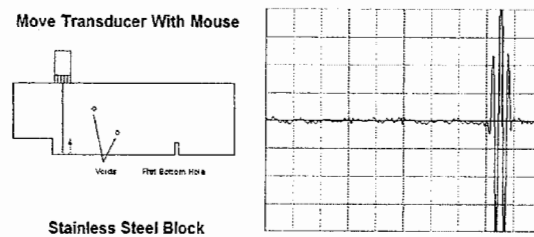
En el caso 3, la señal que sale del transductor encuentra un pequeño vacío en el viaje, reflejando una pequeña parte de la onda, esto se observa en el lado izquierdo del osciloscopio, la mayor intensidad de la onda, continúa hasta el extremo opuesto de la pieza, mostrando en el osciloscopio la reflexión que éste recibe. En el último caso, se observa un defecto cercano al extremo de la pieza, por lo que en el osciloscopio se observa una señal muy cercana a la recibida desde el extremo de la pieza y de mayor intensidad que en el caso anterior, debido a que éste refleja una mayor energía que el pequeño vacío.

Figura 20. Representación para técnica pulso-eco

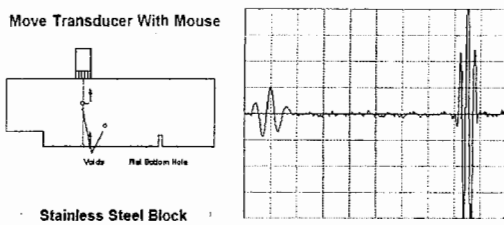
Caso 1:



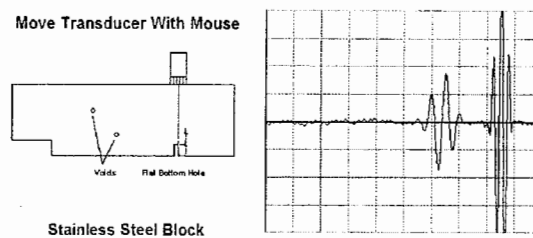
Caso 2:



Caso 3:



Caso 4:

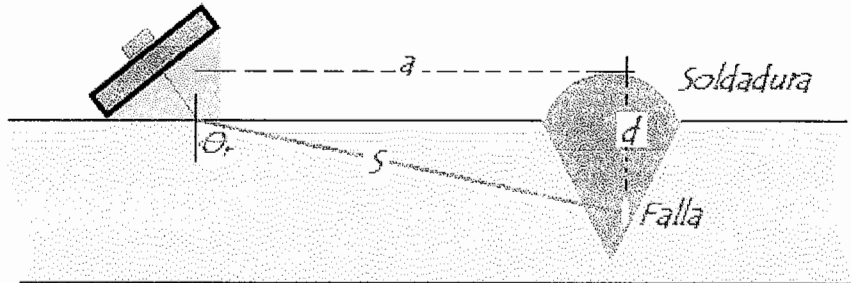


Fuente: *Center for NDE, IOWA State University*. Disponible en Web <http://www.ndt-ed.org/EducationResources/CommunityCollege/Ultrasonics/MeasurementTech/applet_4_1/applet_4_1.htm> Noviembre 2009

La onda que retorna al transductor no se convierte totalmente en energía eléctrica, parte de esta energía pasa a ser una nueva onda reflejada que atraviesa por segunda vez el medio y así sucesivamente. Se debe tener en cuenta que cualquier cambio de impedancia dentro del medio, produce ecos múltiples de menor energía.

En casos donde no es posible lograr una incidencia normal a la superficie, por ejemplo en regiones de metales soldados, es necesario utilizar el transductor de incidencia angular para analizar posibles fallas en las soldaduras, como se muestra en la figura 21.

Figura 21. Técnica de incidencia angular



Si S representa la distancia recorrida por la onda y es posible conocerla, se pueden obtener las distancias a y d para la localización de la falla con las siguientes ecuaciones:

$$a = S * \text{sen}\theta_r \quad (2.7a)$$

$$y \quad (2.7b)$$

$$d = S * \text{Cos}\theta_r$$

2.4. Modos de presentación

Los datos ultrasónicos obtenidos de las mediciones, pueden ser mostrados de diferentes formas. Las tres principales formas son la representación tipo A, B y C. Cada forma o modo de representación provee diferentes maneras de visualizar y evaluar la región que se inspecciona. Actualmente los sistemas computacionales permiten mostrar las tres representaciones simultáneamente.

2.4.1. Representación A

Este método consiste en representar la magnitud de la onda reflejada al transductor en base del tiempo. Esta representación es la utilizada para representar los resultados en la Figura 20. Normalmente los equipos utilizados para la representación rectifican y filtran la señal, mostrando solamente el valor de la magnitud en pantalla.

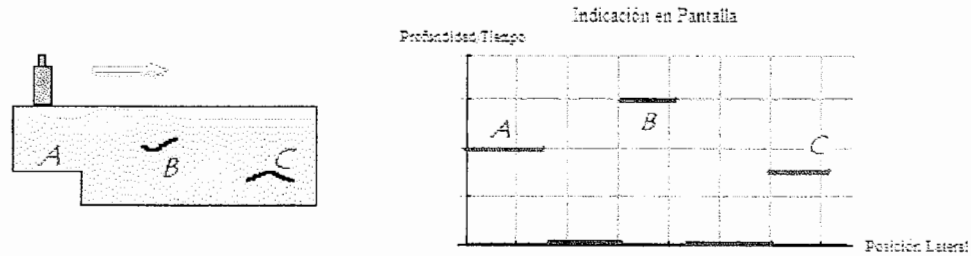
La representación tipo A, se discrimina de izquierda a derecha en pantalla y se acostumbra a mostrar la amplitud de la señal original del transductor y obtener un nivel de referencia del eco proveniente de un reflector conocido.

2.4.2. Representación B

En la pantalla se muestra una sección transversal de la región inspeccionada. La imagen es retenida el tiempo necesario para evaluar la pieza, obteniendo un registro permanente de la región transversal.

En el eje vertical se muestra el tiempo en que se obtiene la reflexión, haciendo posible determinar la profundidad donde se localiza el cambio de impedancia. En el eje horizontal se coloca la distancia lateral de inspección, esto se ve en la figura 22.

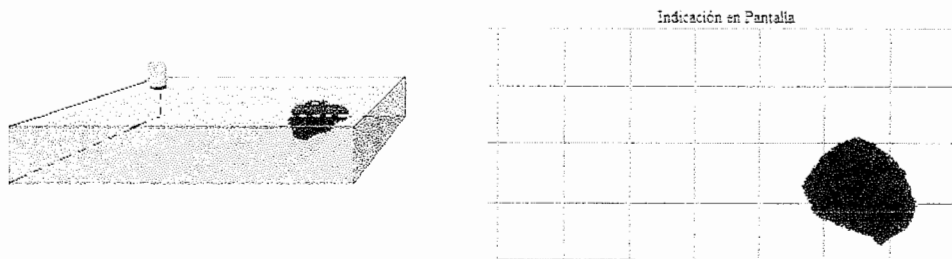
Figura 22. **Representación tipo B**



2.4.3. **Representación C**

Ésta muestra una representación de planta para la región inspeccionada, muestra la forma y localización del cambio de impedancia, pero no muestra la profundidad de ésta, figura 23. Normalmente se utiliza un sistema de adquisición de datos automatizado.

Figura 23. **Representación tipo C**



La combinación de las diferentes representaciones hace posible la reconstrucción de imágenes en tres dimensiones. Para mostrar el resultado de este tipo de representación se muestran dos imágenes de monedas obtenidas en un sistema ultrasónico utilizando un transductor de alta resolución.

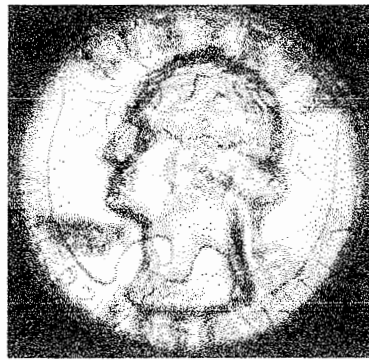
En la figura 24a se arreglan los datos para obtener los ecos solamente de la superficie frontal de la moneda. En la figura 24b se obtienen los datos de los pulsos en la parte posterior de la moneda, se puede observar que el sonido que atraviesa la moneda afecta la reconstrucción de la imagen.

Figura 24. **Reconstrucción de imágenes por ultrasonido de inmersión**

a)



b)



Fuente: *Center for NDE, IOWA State University*. Disponible en Web <<http://www.ndt-ed.org/EducationResources/CommunityCollege/Ultrasonics/EquipmentTrans/DataPres.htm>>

noviembre 2009

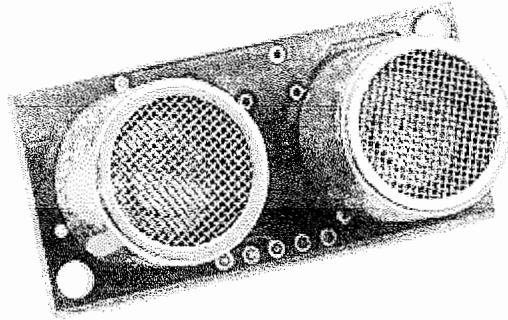
3. INTERFACES SENSOR-COMPUTADORA

En el presente capítulo, se muestran las especificaciones técnicas del sensor a utilizar en los montajes experimentales y se describe la conexión de éste con la computadora, repasando los tópicos de comunicación serial, la comunicación serial utilizando un microcontrolador *PIC16F877A* y los comandos necesarios en *MATLAB* para comunicarse con dicho microcontrolador.

3.1. Descripción del sensor

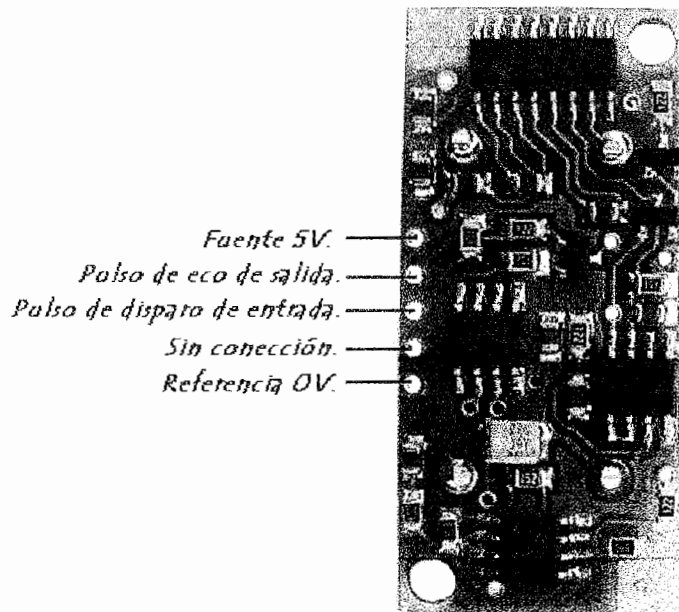
Para la técnica de pulso-eco descrita en el capítulo 2, se utiliza el sensor ultrasónico *SRF04 #28015* manufacturado por *Devantech*, figura 25, los pasos realizados en las pruebas se detallan más adelante en el capítulo 5. Este sensor constituye un *telémetro* (dispositivo capaz de medir distancias en forma remota) con rango entre 3cm hasta 3m. Es recomendable utilizar microcontroladores como el *BASIC Stamps 2* para el control del mismo, pero para los objetivos deseados, en las pruebas se conecta al microcontrolador *PIC16F877A* como interfaz con la computadora. En la computadora *MATLAB* se utiliza para la indicación de señales de control hacia el microcontrolador, también se encarga del almacenaje y procesamiento de las mediciones realizadas por el sensor.

Figura 25. **Sensor *Devantech SRF04***



El *Devantech SRF04* posee cuatro agujeros para señales de entrada y salida, señales de control, referencia y alimentación. En la figura 26 se muestra la parte posterior del sensor y se especifica la posición de cada uno de los agujeros.

Figura 26. **Parte posterior del *SRF04* y agujeros de conexión**

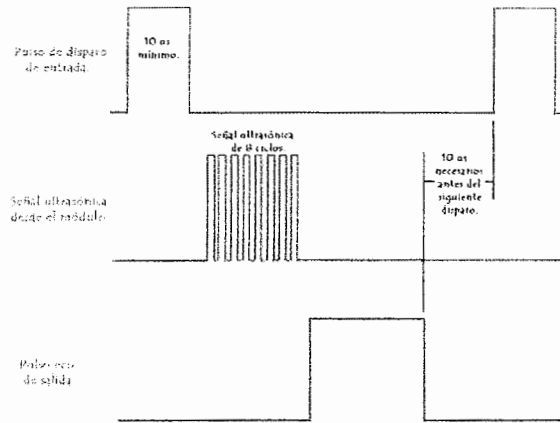


La salida que proporciona el dispositivo es un pulso de ancho variable, el ancho del pulso proporciona información suficiente de la distancia a la que se encuentra el objeto con el que colisiona el haz ultrasónico emitido por el sensor. A continuación se listan las características y especificaciones del mismo:

- Voltaje: 5V
- Corriente: típica 30mA y máxima de 50mA
- Frecuencia: 40KHz
- Rangos: máximo 3m y mínimo de 3cm
- Sensibilidad: capaz de detectar un elemento circular de 3cm de diámetro a distancias mayores de 2m
- Disparo de entrada: 10 microsegundos como mínimo en nivel de pulso *TTL*
- Pulso de eco: Pulso *TTL* positivo con ancho proporcional al rango de medición
- Dimensiones del sensor: 43mm x 20mm x 17mm

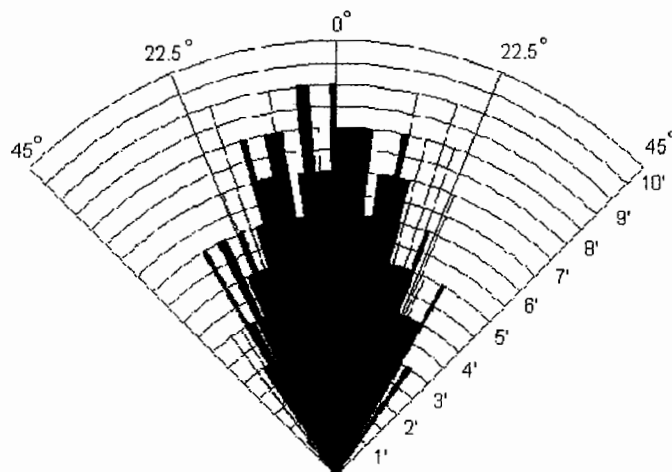
Como se mencionó, el sensor detecta objetos emitiendo una pequeña ráfaga ultrasónica y percibiendo una reflexión de la misma. La señal ultrasónica es enviada a 40Khz y viaja por el aire a velocidad de 340m/s (ver sección 1.5.2 del capítulo 1), de encontrar un objeto, en los rangos especificados del dispositivo, la señal golpea con éste y se refleja un eco hacia el sensor. Midiendo el tiempo de retorno de la señal puede determinarse la distancia hacia el objeto. En la figura 27 se muestra el diagrama de los pulsos para la sincronía del *SRF04*.

Figura 27. Pulsos de sincronización del sensor



El ancho del pulso de eco varía desde $100\mu s$ para una distancia de 3cm y unos $18ms$ para 3m de distancia al objetivo. En la figura 28 se muestra el patrón del haz para el sensor, las distancias radiales se encuentran en pulgadas.

Figura 28. Patrón de haz del SRF04



Fuente: Acrome Robotics. Disponible en Web

<<http://www.acrome.com/robotics/parts/R93-SRF04.html>>. Marzo 2010

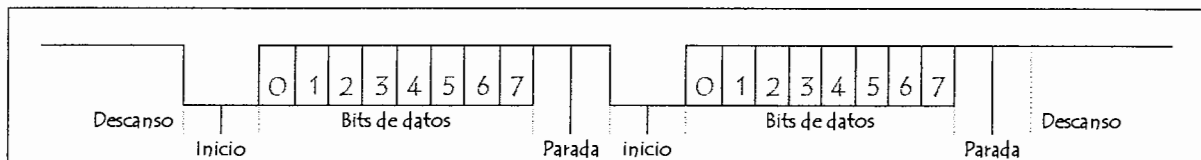
3.2. Comunicación serial

La comunicación serial es utilizada comúnmente por computadores y periféricos para comunicarse con dispositivos terminales. Los datos de información viajan a través de *bits* y solamente un bit a la vez, a diferencia de la comunicación en paralelo en donde viajan simultáneamente. La comunicación serial es utilizada en recorridos largos o cuando la sincronización hace impráctica la comunicación en paralelo.

Un puerto serie, es la interfaz física de comunicación en serie a través del cual se transfieren los bits. Existen varias interfaces de comunicación serial, como *Ethernet*, *FireWire* y *USB*, pero el término de "puerto serie" normalmente hace referencia a al hardware que cumple con el estándar *RS-232*, el cual se estudia posteriormente.

El puerto serie asíncrono utiliza el protocolo de comunicación asíncrono, el cual establece que antes del *byte*, carácter o palabra codificada, se debe enviar una señal inicial. La señal de inicio sirve para preparar al terminal receptor y luego registrar una trama de datos, la señal de detención sirve para que el receptor tome un pequeño tiempo antes de prepararse para tomar un nuevo *bit*. Una trama de comunicación asíncrona se muestra en la figura 29

Figura 29. Trama de comunicación asíncrona

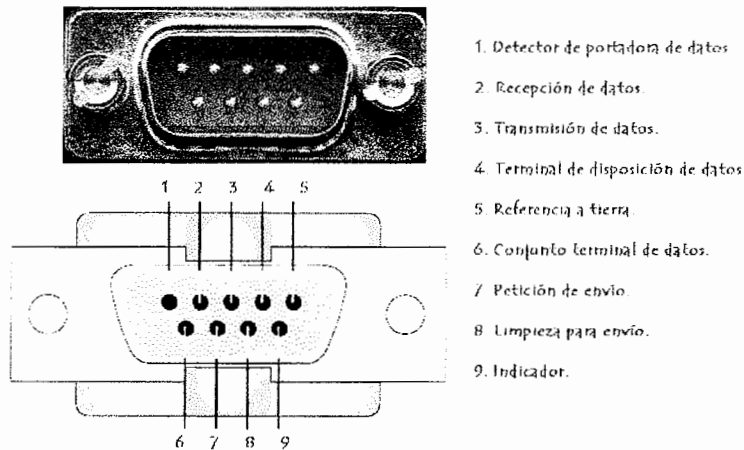


Pueden existir tres tipos de comunicación serial, *simplex*; el emisor y el receptor están bien definidos y no pueden realizar el trabajo opuesto, la comunicación es unidireccional. El segundo tipo es llamado *Duplex, semi-duplex o half-duplex*; ambos extremos del sistema pueden funcionar como emisor o receptor, los datos se desplazan en ambos sentidos pero no de manera simultánea. El tercer tipo se conoce como *full-duplex*; es similar al caso anterior, pero la comunicación puede fluir en ambos sentidos al mismo tiempo.

3.2.1. El estándar RS-232

De sus siglas en ingles, *Recommended Standar 232*, también llamado simplemente COM. Es una interfaz que designa las normas para la comunicación entre un equipo terminal de datos y un equipo de comunicación de datos. El conector físico del estándar es un *DB-25* (del inglés *D-Subminiature* de 25 pines), aunque es común encontrar la versión de 9 pines (*DB-9*). En la figura 30 se muestra un conector *DB-9* (macho) y la descripción de sus pines. Se hace énfasis en este conector porque es el que se utiliza para la comunicación con el sensor ultrasónico

Figura 30. Conector *DB-9* y sus pines



La interfaz *RS-232* está diseñada para distancias relativamente cortas, hasta 15 m según la norma y para velocidades de transmisión bajas, no mayores a 20Kbits/s. La interfaz puede funcionar en modo síncrono o asíncrono y en tipos de canal, *simplex*, *half-duplex* o *full-duplex*. Las convenciones de voltaje utilizadas son las siguientes:

Tabla II. **Niveles de voltaje *RS-232***

Voltaje.	Señal.	Nivel lógico
de -3V a -15V	marca	1
de +3V a +15V	espacio	0
de -3V a 3 V	Indeterminado	Indeterminado

Los niveles de voltajes del *RS-232* no son compatibles con los niveles de voltaje que de la tecnología *TTL* (del inglés *transistor-transistor logic*), que consiste en la construcción de circuitos electrónicos digitales. Cuando se desea comunicar a través de la interfaz *RS-232* y uno de estos circuitos, es necesario utilizar dispositivos especiales, uno de ellos es el *MAX232* descrito en la siguiente sección.

3.2.2. Circuito integrado *MAX232*

Es un circuito integrado que convierte señales provenientes de una interfaz *RS-232* a señales soportables por la tecnología *TTL*, compatibles con circuitos lógicos digitales y también hace el acoplamiento de señales en el sentido contrario, cuando un dispositivo *TTL* envía datos hacia la interfaz *RS-232*.

Se pueden obtener las versiones *Multichannel RS-232 Drivers/Receiver* de *MAXIM Dallas Semiconductor* y el *MAX232 DUAL EIA-232 Driver/Receiver* de *TEXAS INSTRUMENT*. A continuación se muestra la conversión que realiza el *MAX232*.

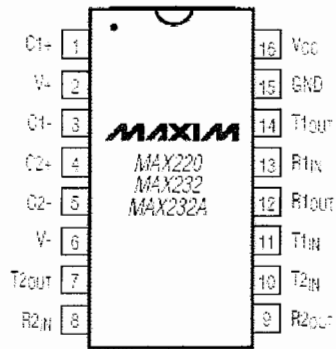
Tabla III. **Conversión de voltajes del MAX232**

Linea del RS-232 y nivel lógico	Voltaje RS-232	Voltaje TTL
Transmisión de datos (Rx/Tx)/ 0 lógico	+3V to +15V	0V
Transmisión de datos (Rx/Tx) /1 Lógico	-3V to -15V	5V
Señal de control/ 0 Lógico	-3V to -15V	5V
Señal de control/1 Lógico	+3V to +15V	0V

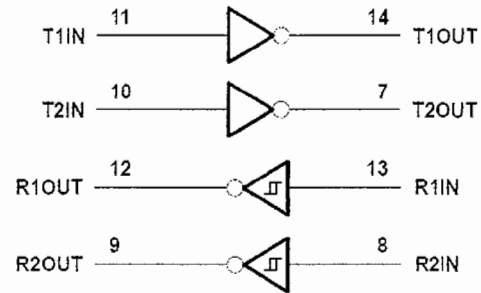
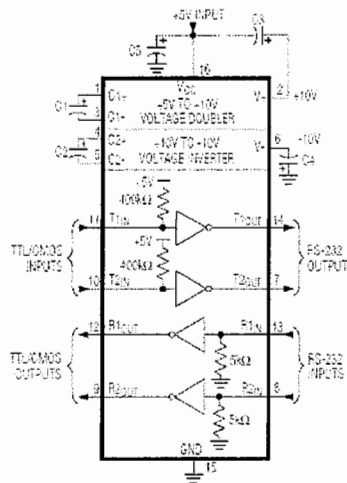
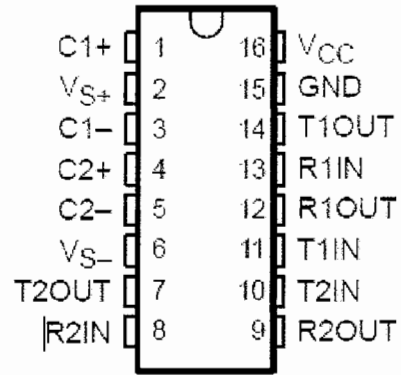
En la figura 31 se muestran las vistas superiores de los integrados proporcionados por *MAXIM* (a) y por *Texas Instrument* (b) y los diagramas internos equivalentes de cada uno de estos.

Figura 31. **MAX232 a) por MAXIM, b) por TI**

a)



b)

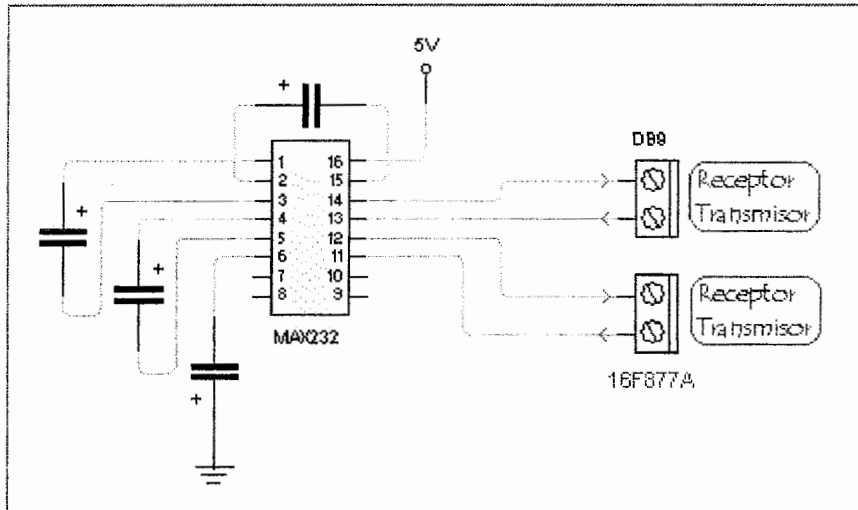


Fuente: a) *MAXIM Dallas Semiconductor*. Disponible en Web <<http://datasheets.maxim-ic.com/en/ds/MAX220-MAX249.pdf>>, b) *Texas Instrument*. Disponible en Web: <<http://www.datasheetcatalog.org/datasheet/texasinstruments/max232.pdf>>. Marzo

2010

A continuación se muestra un diagrama básico de la conexión del *MAX232* con un *DB-9* para el envío y recepción de datos con la computadora y el microcontrolador. Esta conexión será necesaria entre el *RS-232* y el microcontrolador *16F877A* para acoplar las señales que estos manejan.

Figura 32. Conexión del MAX232



3.2.3. Bus serial universal

Abreviado por sus siglas en ingles *USB (universal serial bus)*, es una especificación que establece la comunicación entre dispositivos y un ordenador central, normalmente una *PC*. Este *bus* está destinado a remplazar a los puertos serie y paralelo, debido a la facilidad de conexión *plug and play*, permitiendo a los dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciar el mismo. Entre los dispositivos que el *USB* puede conectar con el ordenador central se encuentran: ratones, teclados, impresoras, cámaras digitales, teléfonos móviles, reproductores multimedia, tarjetas de sonido, sistemas de adquisición de datos y componentes de red.

El *USB* ha remplazado a los ratones y teclados *PS/2*, dando lugar a las placas bases modernas sin dicho puerto. En la mayoría de dichas placas se observa la carencia del puerto serie, siendo el predominante el puerto *USB* para transmisión de datos entre dispositivos y ordenador.

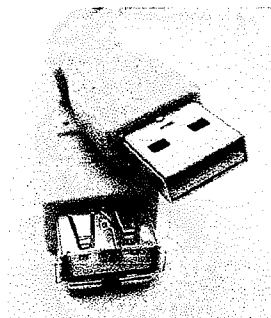
Es posible clasificar los dispositivos *USB* dependiendo de la velocidad de transmisión de datos:

- Baja velocidad (*USB 1*), transfiere datos hasta 1.5Mbps. utilizado principalmente en dispositivos de interfaz humana, como teclados y ratones
- Velocidad completa (*USB 1.1*), transferencias hasta de 12Mbps, fue la mayor hasta que surgió la versión 2. Muchos dispositivos aún trabajan bajo esta especificación
- Alta velocidad (*USB 2*), transferencias hasta de 480Mbps. Presente en la mayoría de ordenadores actuales
- Súper alta velocidad (*USB 3*), velocidades de hasta 4.8Gbps

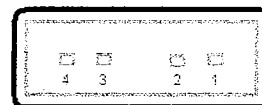
Las señales *USB* viajan a través de un cable de par trenzado de aproximadamente 90Ω . Utilizan señalización diferencial full dúplex. Los niveles de transmisión varían entre 0V y 0.3V para niveles bajos y de 2.8V a 3.6V para niveles altos. La conexión *USB* solamente admite dispositivos de bajo consumo, 100mA por cada puerto. En la figura 33 se muestra un cable con sus terminales macho y hembra y un esquema con la descripción de los pines

Figura 33. a) cable *USB*, b) descripción de los pines

a)



b)



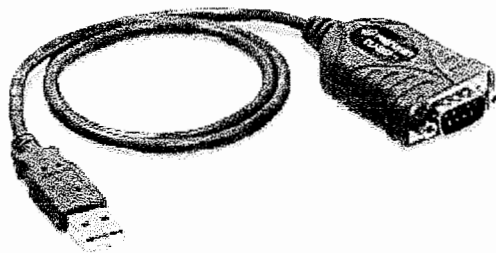
- | | |
|------------|---------|
| 1. 5V | Rojo. |
| 2. Datos - | Blanco. |
| 3. Datos + | Verde. |
| 4. Tierra | Negro. |

3.2.4. Convertidor *USB* a *RS-232*

Como se mencionó anteriormente, muchas de las computadoras modernas no poseen el puerto serie, debido que para la mayoría de aplicaciones se ha vuelto obsoleto. Sin embargo, en muchas transmisiones electrónicas el protocolo *RS-232* es más conveniente para la comunicación de datos entre algún dispositivo y la computadora. Es posible utilizar un convertidor *USB* a serial, los cuales se encuentran disponibles en el mercado integrados en un solo cable o adaptador. Con estos convertidores se puede emular un puerto serial a partir del puerto *USB* de la computadora y de esta manera hacer uso del protocolo *RS-232*.

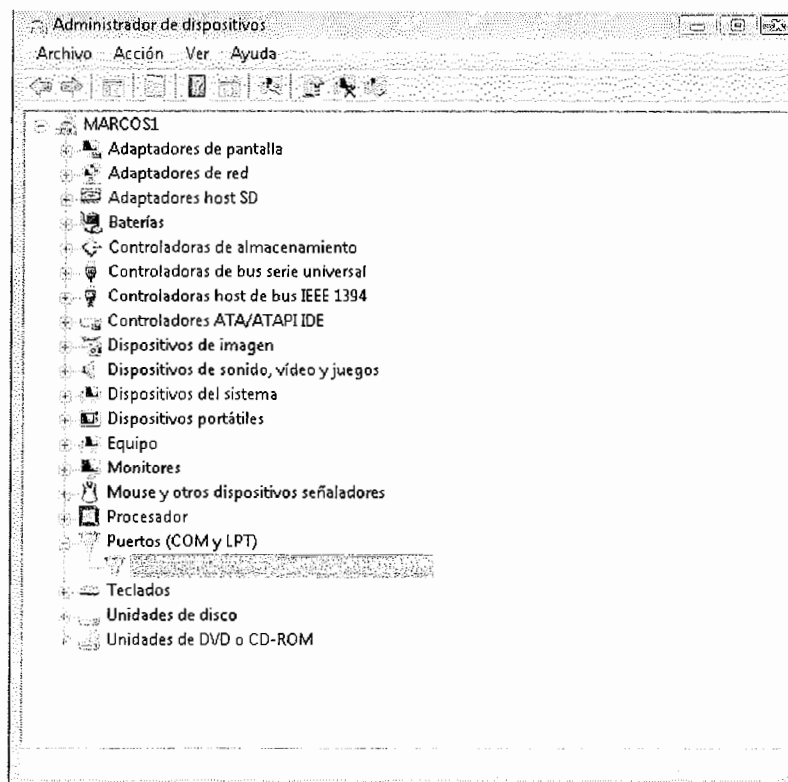
En las pruebas a realizar se utilizará el convertidor *TRENDNET TU-S9*. Posee una terminal tipo *USB* 1.1 macho y la otra terminal tipo *RS-232* con conector *DB-9* macho. El convertidor es compatible con sistemas operativos *Windows* 98 o posterior y *Mac* 10.x. La gama de datos es de 50Kbps con un consumo máximo de 500mA. El convertidor se muestra en la figura 34.

Figura 34. Convertidor *TRENDNET TU-S9*



Después de instalar el controlador y conectar el *TU-S9* a la computadora, se puede verificar en la ventana de dispositivos de *Windows*, accesible desde el panel de control/sistema, si el dispositivo es reconocido. En este caso se observa *Prolific USB-to-Serial Comm Port (COM9)*, figura 35. El número de COM se puede cambiar haciendo doble click sobre este, permitiendo también ver las propiedades del mismo, que se vuelve útil para el posterior uso de del puerto desde el software en donde se programará.

Figura 35. **Ventana de administrador de dispositivos**



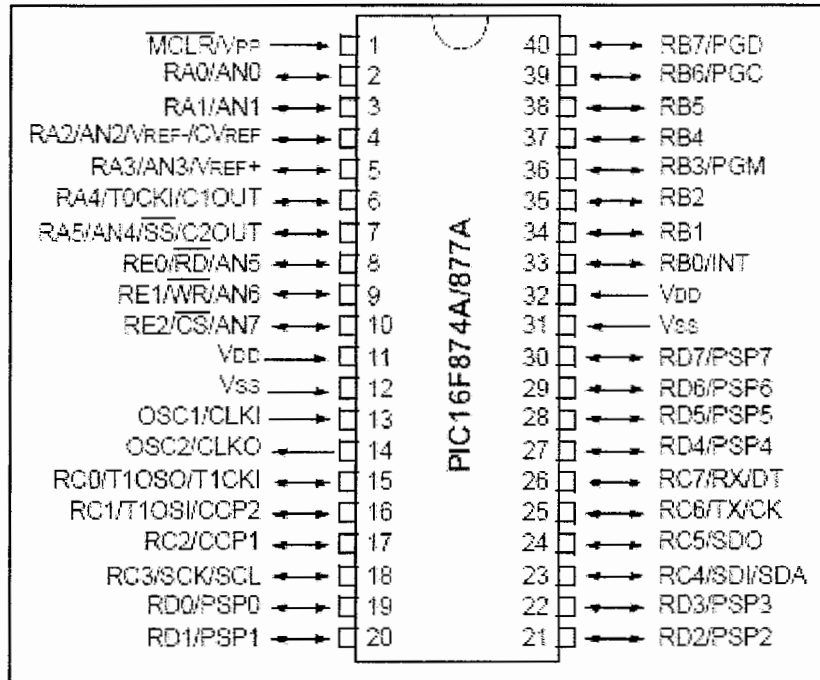
3.3. Controlador del sensor

Una vez conocidas las señales que debe recibir el sensor, es necesario contar con el dispositivo que envíe estas señales, también debe recibir la respuesta del sensor y decidir qué hacer con la información. Un microcontrolador cumple con estas características y se ha escogido el *PIC16F877A* para que realice dicho trabajo, que también cuenta con un módulo de comunicación serial para la transmisión y recepción de datos con la computadora.

3.3.1. Microcontrolador *PIC16F877A*

Pertenece a la subfamilia de *PIC16F87XA* de microcontroladores *PIC* (*Peripheral Interface Controller*) de 8 bits, fabricados por *Microchip Technology Inc.* Cuentan con memoria de programa tipo *EEPROM flash*, lo que permite programarlos fácilmente, lo cual es favorable comparado con el método de programación de los microprocesadores. La letra "A" al final de los modelos indica que cuentan con módulos de comparación analógica. Se clasifican de 8 bits debido a la longitud de datos que manejan sus instrucciones, correspondiente al tamaño del *bus* de datos y el de los registros del *CPU*. En la figura 36 se muestra el *PIC16F877A DIP* (*Dual In-line Package*).

Figura 36. Esquema **PIC16F877A DIP**



Fuente: Microchip, *PIC16F87XA data sheet*, p. 3

El microcontrolador se utiliza para enviar señales hacia el pin de control del sensor, basándose en las señales que este necesita según lo mostrado en la figura 26. Además el microcontrolador se utiliza para recibir las señales en algún puerto con las que responda el sensor, para procesar las mismas y enviar la información hacia la computadora.

3.3.2. Funciones del **BASIC Compiler** para comunicación serial

Para la programación *del PIC16F877A* es posible utilizar diferentes tipos de compiladores, en la realización del trabajo se utilizará el *PIC simulator IDE* que cuenta con *BASIC Compiler*.

El *BASIC Compiler* proporciona las funciones *SerIn* y *SerOut* para llevar a cabo la comunicación serial en *PIC*'s que soporten esta. El primer argumento de estas funciones debe ser el pin por donde se envía o recibe la información, siendo el pin 25 (*Tx*) para el envío de datos y el pin 26 (*Rx*) para la recepción en el caso del *PIC16F877A* como se muestra en la figura 13. El segundo argumento es la tasa de baudios (300, 600, 1200, 2400, 4800, 9600 o 19200 baudios), hay que tener en cuenta colocar un reloj de frecuencias altas si se utiliza una tasa de baudios alta para evitar errores de entramado en la comunicación. El tercer argumento constituye el nombre de la variable que se envía o en la información, dependiendo de la función utilizada. El siguiente fragmento de código muestra la utilización de las funciones antes descritas

Fragmento de código 1. Uso de las funciones *SerIn* y *SerOut*

```
Dim i As Byte
loop:
  SerIn PORTC.7, 9600, i
  SerOut PORTC.6, 9600, "Number: ", #i, CrLf
Goto loop
```

El simbolo **#** que se coloca anterior a la variable, se hace para que el dato enviado por el puerto se encuentre en representación decimal. La función *CrLf* se utiliza para cambiar de línea (*carriage return*) y cada vez que se envíe un dato por el puerto se escriba en una nueva línea en su lugar de destino.

3.4. Comunicación serial utilizando *MATLAB*

En esta sección se describen las funciones necesarias para la escritura y lectura del puerto serial desde *MATLAB* 7.8.0 (R2009a). La utilización de estas funciones, dan punto final a la comunicación entre el sensor y la computadora, para que la información que el sensor proporciona sea almacenada y procesada por *MATLAB* con técnicas descritas en capítulo 4.

Para poder utilizar el puerto serial es necesario crear un objeto utilizando la función *serial('port')*, como se muestra en el fragmento de código mostrado a continuación.

Fragmento de código 2. Función serial de *MATLAB*

```
>> a=serial('COM9')

Serial Port Object : Serial-COM9

Communication Settings
Port:          COM9
BaudRate:     9600
Terminator:   'LF'

Communication State
Status:       closed
RecordStatus: off

Read/Write State
TransferStatus: idle
BytesAvailable: 0
```

```
ValuesReceived: 0
ValuesSent: 0
```

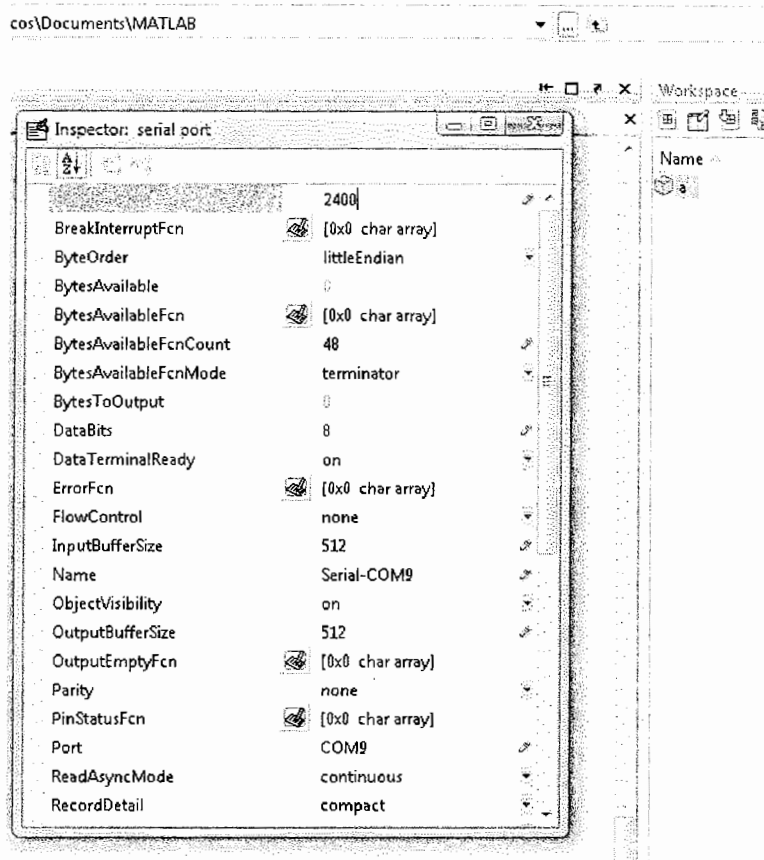
Como se observa, se creó un objeto con el nombre *a*, que ahora posee los parámetros del *COM9*, de las cuales se despliegan algunos después de utilizar la función. Es posible cambiar estos parámetros utilizando la función *set(a,'nombre del parámetro',valor)*. Por ejemplo, si se desea cambiar el *BaudRate* a 2400 y cambiar otros parámetros o asegurar el valor de estos, por ejemplo que se utilicen 8 bits de datos, que no exista bit de paridad y un solo bit de parada en el protocolo RS-232 se escribe

Fragmento de código 3. **Modificación de los parámetros del puerto**

```
set(a,'BaudRate',2400);
set(a,'DataBits',8);
set(a,'Parity','none');
set(a,'StopBits',1);
```

También es posible hacerlo de otra forma, una vez creado el objeto, en el *Workspace* de *MATLAB* aparecerá el objeto con el nombre *a*, para el cual es posible realizar algunos cambios en sus parámetros, ver figura 37.

Figura 37. **Modificación de los parámetros del puerto serie**



Se puede observar el estado del puerto como *closed*, esto porque aún no se ha abierto el puerto, para esto se utiliza la función *fopen(a)*, con lo cual queda listo el puerto para enviar o recibir datos. Para enviar datos en el puerto se utiliza:

Fragmento de código 4. Envío de datos por el puerto serial

```
>> fopen(a)
>> fprintf(a,'%s','10') % envía el 10 como un string
```

Se escribe el %s para indicar que se enviará un *string* (cadena de caracteres). Existen diferentes formatos para el envío de datos con la función *fprintf*, en la figura 38 se muestran las maneras posibles.

Figura 38. **Forma de envío de datos por el puerto serie**

Specifier	Description
%c	Single character
%d	Decimal notation (signed)
%e	Exponential notation (using a lowercase e as in 3.1415e+00)
%E	Exponential notation (using an uppercase E as in 3.1415E+00)
%f	Fixed-point notation
%g	The more compact of %e or %f, as defined in [2]. Insignificant zeros do not print.
%G	Same as %g, but using an uppercase E
%i	Decimal notation (signed)
%o	Octal notation (unsigned)
%s	String of characters
%u	Decimal notation (unsigned)
%x	Hexadecimal notation (using lowercase letters a-f)
%X	Hexadecimal notation (using uppercase letters A-F)

Fuente: *MATLAB Help, Title: fprintf:: Functions(MATLAB®)*

Para leer datos del puerto se utiliza la función $[A,c]=fscanf(a,formato,n)$. Los n datos serán almacenados en la variable A y la variable c indica cuantos datos se recibieron con éxito. Para poder manipular aritméticamente estos datos es necesario cambiarlos a números, para esto se utiliza la función $x=str2num(A)$.

Es importante cerrar el puerto después de utilizar este, y recomendable borrar la variable que fue creada, como se muestra en el siguiente fragmento de código

Fragmento de código 5. **Cerrando el puerto serial**

```
>> fclose(a)
```

```
>> clear a
```


4. PROCESAMIENTO DIGITAL DE SEÑALES A TRAVÉS DE *MATLAB*

En este capítulo se presentan las técnicas de procesamiento de señales¹ que se consideran más necesarias para los montajes experimentales del capítulo próximo. Estas técnicas, en forma de ejemplificación, se aplican a imágenes con ayuda de *MATLAB*.

En la última sección se encuentran agrupadas las principales funciones de *MATLAB* utilizadas acá y una breve descripción del funcionamiento de las mismas.

4.1. Matrices e imágenes en *MATLAB*

En este primer apartado se muestran las funciones básicas de *MATLAB* para la manipulación de imágenes en forma matricial.

¹El término señales, en este contexto, hace referencia a las mediciones provenientes del controlador del sensor y almacenadas en la computadora.

4.1.1. Conceptos básicos

Utilizando *MATLAB* es posible representar una imagen en una escala de colores por medio de una matriz de $m \times n$ elementos, donde m representa el número de filas de la matriz e indica también el número de píxeles de alto que posee la imagen y n representa el número de columnas indicando también el número de píxeles de ancho de la imagen. En el fragmento de código 4.1 se crea una matriz aleatoria de 5×5 y se representa como una imagen en escala de grises.

Fragmento de código 4.1. Generación de imagen en escala de grises de una matriz aleatoria

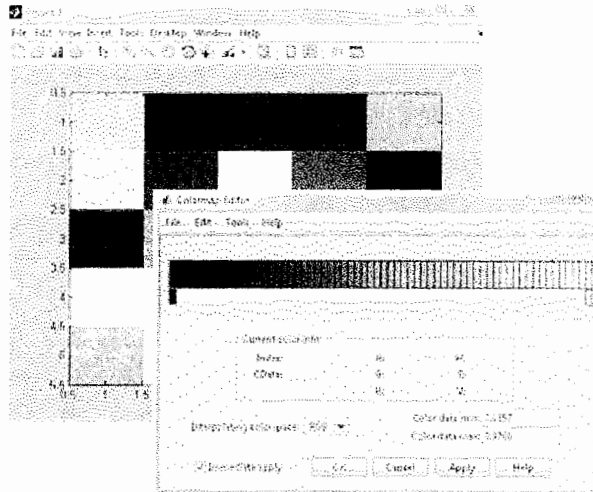
```
>> I=rand(5,5)
I =
    0.8147    0.0975    0.1576    0.1419    0.6557
    0.9058    0.2785    0.9706    0.4218    0.0357
    0.1270    0.5469    0.9572    0.9157    0.8491
    0.9134    0.9575    0.4854    0.7922    0.9340
    0.6324    0.9649    0.8003    0.9595    0.6787

>> imagesc(I); colormap(gray)
```

La función *rand(m,n)* se utiliza para crear una matriz aleatoria de m filas por n columnas. La función *imagesc* (imagen escalada) junto a *colormap(gray)*, se utilizan para mostrar la matriz como una imagen en escala de grises, asignando el color negro (0) al menor valor existente en la matriz y el color blanco (255) al mayor valor existente en la matriz, la intensidad de gris asignada a los datos restantes es proporcional a la diferencia existente entre estos y los datos extremos.

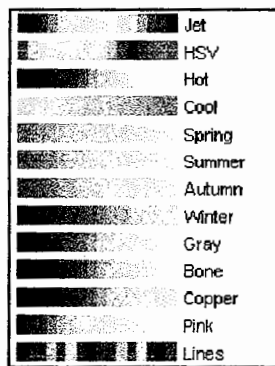
En la figura 39 se muestra la gráfica que representa la matriz I y el editor de colormap que muestra los valores máximos y mínimos presentes en ésta.

Figura 39. **Imagen escala de grises e indicación del editor de *colormap***



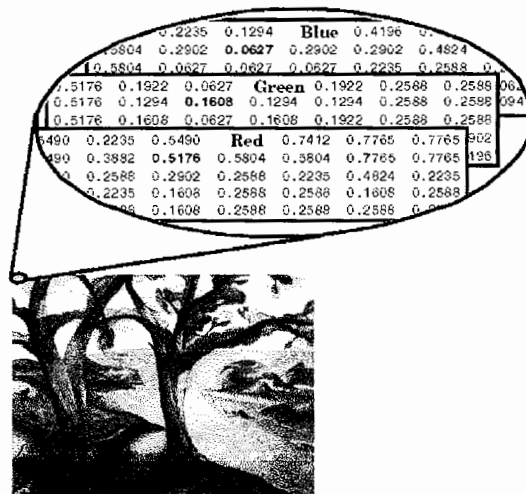
A pesar que la función más utilizada para mostrar imágenes es *imshow*, en este caso se substituyó por *imagesc*, por mostrar en un tamaño mayor al real las imágenes, a pesar de los pocos elementos que comprenden la matriz I. La función *colormap* posee diversas escalas de colores para aplicar a matrices. En la figura 40 se muestran las diferentes escalas que proporciona dicha función

Figura 40. **Posibles escalas de la función *colormap***



También es posible trabajar con imágenes en color *RGB* (*Red Green Blue*) que es la opción por defecto que utiliza *MATLAB*. Este tipo de imágenes se representan por una matriz $n \times m \times c$, las variables n y m representan las dimensiones de la imagen, al igual que para las imágenes escaladas descritas anteriormente. La variable c indica el plano *RGB* que puede ser 1 para el rojo, 2 para el verde y 3 para el azul. La figura 41 muestra detalles de lo descrito anteriormente.

Figura 41. **Detalles de imagen *RGB***



Fuente: *The Mathworks. Image Processing ToolBox User's Guide*, p. 74

4.1.2. Lectura y escritura de imágenes

Es posible leer imágenes archivadas hacia el escritorio de trabajo de *MATLAB*, los formatos soportados se muestran en la tabla IV.

Tabla IV. **Formatos de imágenes permitidos en MATLAB**

Formato:	Extensión:
<i>TIFF (Tagged Image File Format)</i>	<i>.tiff</i>
<i>JPEG (Joint Phonographics Experts Groups)</i>	<i>.jpg</i>
<i>GIF (Graphics Interchange Format)</i>	<i>.gif</i>
<i>BMP (BitMaP)</i>	<i>.bmp</i>
<i>PNG (Portable Network Graphics)</i>	<i>.png</i>
<i>XWD (X Windows Command)</i>	<i>.xwd</i>

Para obtener información de una imagen se utiliza el comando *imfinfo*, en este caso se desea conocer la información de una imagen contenida en la carpeta de trabajo actual de MATLAB, con nombre imagen1 y con formato JPEG. Se utiliza el siguiente fragmento de código.

Fragmento de código 4.2. **Obtener información de una imagen en archivo**

```
>> imfinfo('imagen1.jpg')

ans =

Filename: 'imagen1.jpg'
FileModDate: '30-abr-2010 17:51:25'
FileSize: 30603
Format: 'jpg'
FormatVersion: ''
Width: 362
Height: 400
BitDepth: 8
ColorType: 'grayscale'
FormatSignature: ''
NumberOfSamples: 1
CodingMethod: 'Huffman'
```

CodingProcess: 'Sequential'

Comment: {}

Por lo regular los elementos que constituyen una imagen en *MATLAB* tienen un formato entero llamado *uint8*, permitiendo poseer valores enteros entre 0 y 255, sin poder soportar valores fuera del rango o decimales entre el rango. En ocasiones es necesario realizar operaciones en donde se trabaja con decimales, para las cuales deja de ser útil usar el formato *uint8*. *MATLAB* permite cambiar el formato a *double* el cual permite operaciones con punto flotante, para regresar posteriormente al formato *uint8* y mostrar la imagen. A continuación se muestran los diferentes tipos de imágenes dependiendo de los elementos que conformen esta, tabla V.

Tabla V. **Tipos de imágenes según el formato de sus pixeles**

Tipo:	Descripción:
<i>Double</i>	De doble precisión, números en punto flotante que varían aproximadamente entre -10308 a 10308 y poseen 8 bytes por elemento.
<i>uint8</i>	Enteros entre 0 a 255 con un byte por elemento.
<i>uint16</i>	Enteros entre 0 a 65535 con dos bytes por elemento.
<i>uint32</i>	Enteros entre 0 a $2^{32}-1$ con cuatro bytes por elemento.
<i>Int8</i>	Enteros entre -128 a 127.
<i>Int16</i>	Enteros entre -32768 a 32767.
<i>Int32</i>	Enteros entre $-(2^{32})/2$ a $(2^{32})/2-1$.
<i>Logical</i>	Valores 0 ó 1 con un bit por elemento.

Para leer una imagen se utiliza la función *imread* y su argumento depende de la ubicación de la imagen. Sí la imagen se encuentra en la carpeta actual de trabajo, solamente se escribe el nombre con su respectiva extensión, de no ser así, es necesario anteponer la dirección completa de la ubicación de la imagen.

Esta función se muestra en el fragmento de código 4.3, en donde se utiliza la función *whos* para obtener información de la variable a la cual se asignó la imagen.

Fragmento de código 4.3. Lectura de imagen

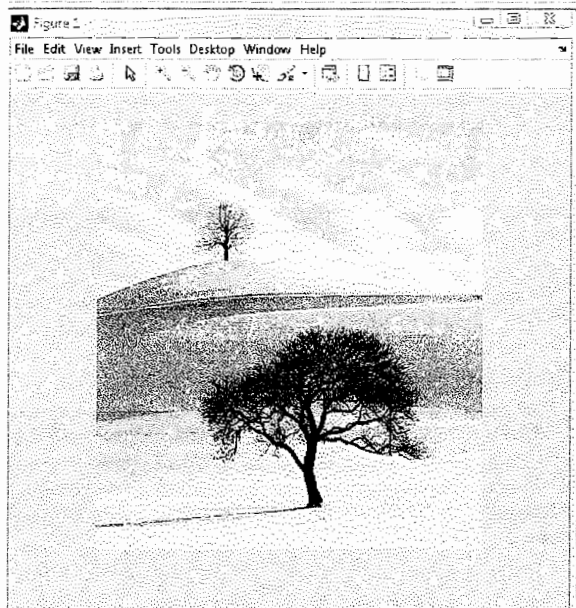
```
>> I1=imread('imagen1.jpg');
>> whos
```

Name	Size	Bytes	Class	Attributes
I1	400x362	144800	uint8	

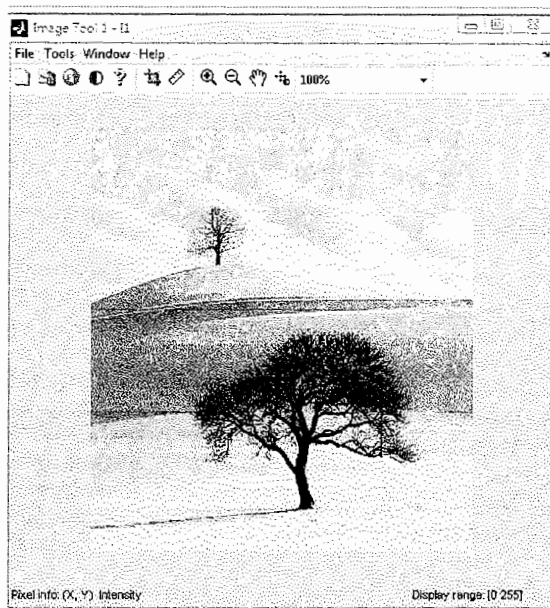
Existen distintas maneras de mostrar una imagen, la más común es utilizando la función *imshow*, con la cual se muestra el Toolbox fundamental para mostrar imágenes. Otra forma es utilizando la función *imshow*, la cual muestra un *Toolbox* con un entorno integrado y completo para el procesamiento de la imagen mostrada. En la figura 42 se muestran las opciones descritas anteriormente, la primera con la función `imshow(I1)` y la segunda con `imshow(I1)`.

Figura 42. Mostrando imagen con a) *imshow*, b) *imshow*

a)

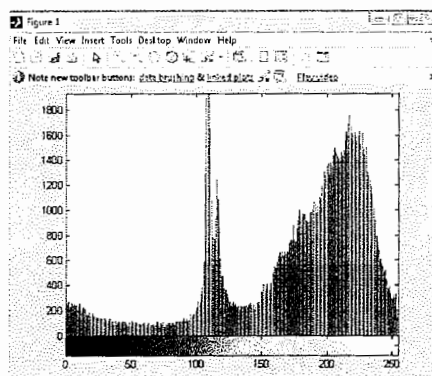


b)



La imagen se encuentra almacenada en una variable de *MATLAB*, es posible procesar ésta. Como ejemplo de esto, se observará la distribución de las intensidades de la imagen, se utiliza la función *imhist(I1)* para que el *Image Processing Toolbox* de *MATLAB* muestre de manera gráfica la distribución de intensidades de grises dentro de un rango de 255 posibilidades como lo muestra la figura 43.

Figura 43. Intensidad de grises de la imagen



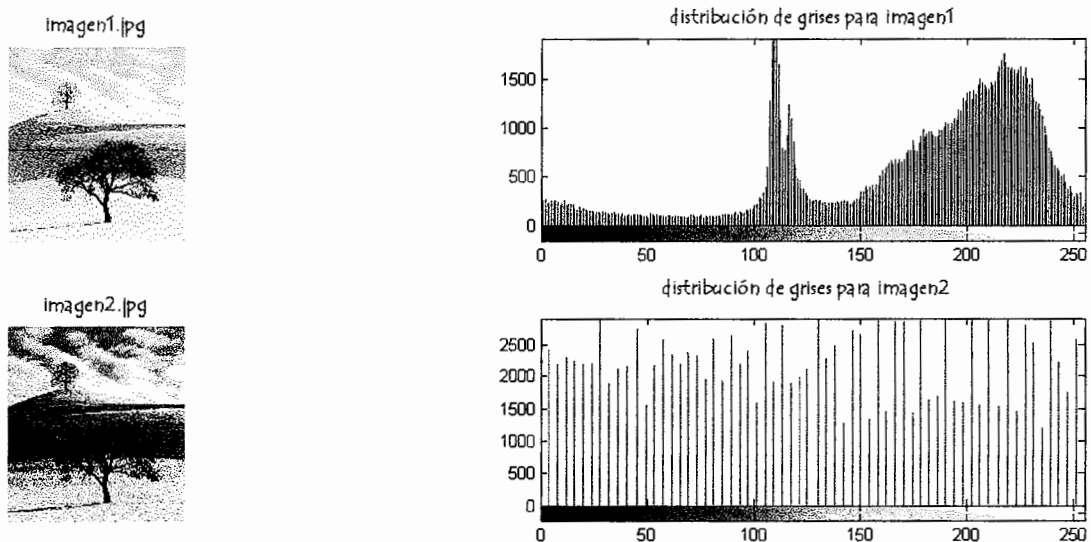
Se puede escribir el comando `I2=histeq(I1)` para ecualizar la imagen y obtener un menor contraste en la misma. Se puede observar la distribución de intensidades de la variable `I2` y posteriormente guardar la nueva imagen utilizando la función `imwrite`, como se muestra en el fragmento de código 4.5.

Fragmento de código 4.4. **Modificando el contraste de una imagen**

```
>> I2=histeq(I1);  
>> imwrite(I2,'imagen2.jpg') % es posible utilizar  
                             % otra extensión si se desea
```

A continuación se muestran las imágenes en la figura 44, la original `imagen1.jpg` y la modificada `imagen2.jpg` con sus respectivos histogramas de distribución de grises.

Figura 44. **Imágenes y sus distribuciones de intensidad**

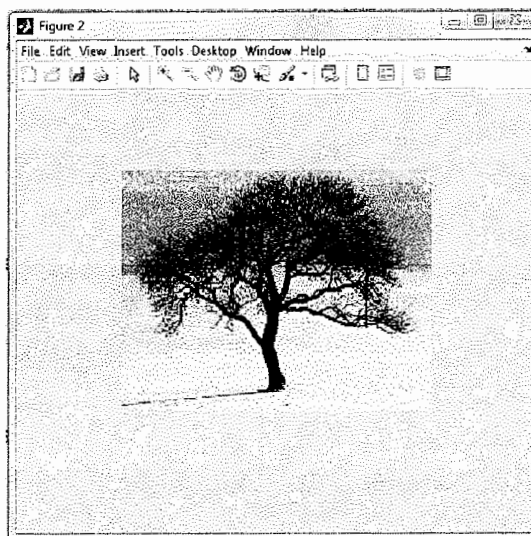


4.1.3. Acceso a los elementos en imágenes

Las imágenes en escala de grises poseen solo un tipo de elementos, los píxeles, que representan la intensidad de gris en determinada posición. En *MATLAB* es sencillo acceder a los píxeles de las imágenes, indexando el elemento dentro de la matriz que representa al píxel de interés. Para conocer el píxel en la fila 200 y columna 300 de la imagen1.jpg almacenada en la matriz I1, se escribe `I1(200,300)` con lo que *MATLAB* responde con un valor de 131. También es posible modificar la intensidad de color de un píxel, por ejemplo para el píxel anterior, para volverlo color negro solamente se escribe `I1(200,300)=0` y se escribe `I1(200,300)=255` para convertirlo en blanco.

También es posible recortar solamente una parte de la imagen, de desear seleccionar un fragmento dentro de la imagen1 contenido entre las filas 190 hasta la 375 y columnas 85 hasta la 325, se escribe `Ir=I1(190:376,85,326)`. Al mostrar la nueva imagen almacenada, ver figura 45.

Figura 45. Recorte en imagen



Otra alternativa es utilizar la función *imcrop(l1,area)* en donde *area* representa el rectángulo que seleccionará el corte en la imagen, bajo los siguientes parámetros: *area*=[columna inicio, fila inicio, base rectángulo, altura rectángulo]. Para realizar la misma selección que se hizo anteriormente se deben seguir los pasos del fragmento de código 4.5, obteniendo los mismos resultados.

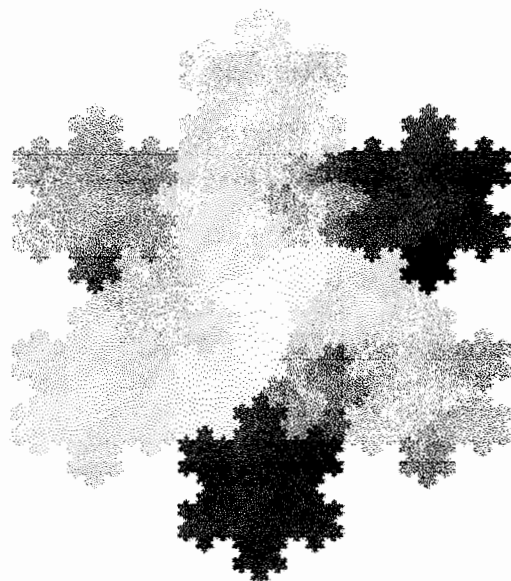
Fragmento de código 4.5. Recorte en imagen

```
area=[85,190,326-85,376-190];  
lr2=imcrop(l1,area);  
>> figure; imshow(lr2)  
>> imwrite(lr2,'im1rec.jpg')
```

Para hacer una imagen más pequeña se usa la función *imresize(l1,porcentaje)*. En donde el parámetro porcentaje representa el tamaño, con respecto a la imagen original de la nueva imagen.

Las imágenes que se analizaron anteriormente eran en escala de grises, las cuales poseen solamente un plano, una matriz $m \times n$. No es el mismo caso para imágenes a color, éstas poseen diferentes planos, como las imágenes RGB descritas en la primera sección del presente capítulo. Para ejemplificar, se posee la imagen3 almacenada en la carpeta actual de trabajo de *MATLAB* y se muestra en la figura 46.

Figura 46. **Imagen a color**



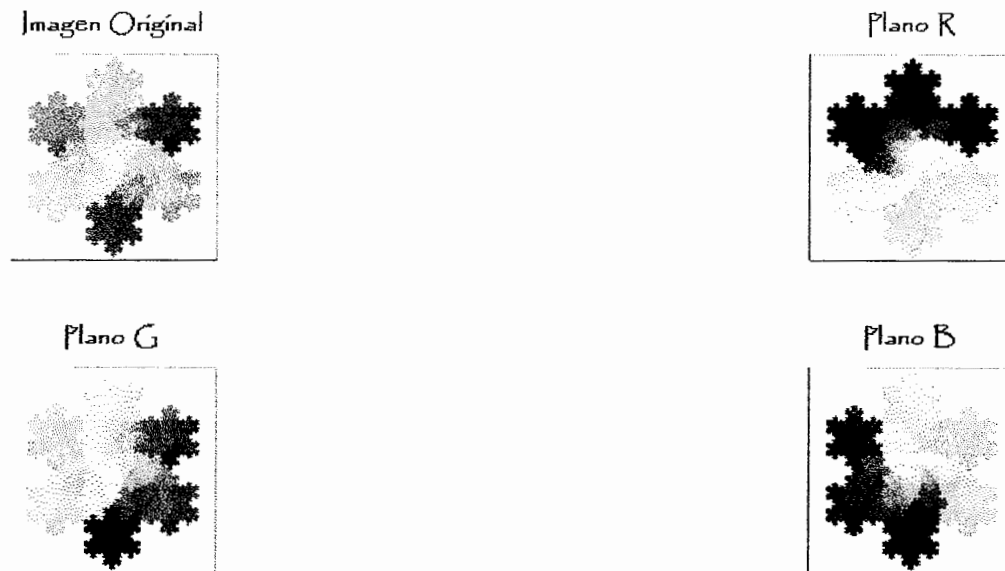
Se separa y visualiza cada plano que componen a la imagen3.jpg con el siguiente fragmento de código

Fragmento de código 4.6. **Separación y visualización de los planos *RGB***

```
>> I3=imread('imagen3.jpg');  
>> IR=I3(:, :, 1);  
>> IG=I3(:, :, 2);  
>> IB=I3(:, :, 3);  
>> subplot(2,2,1); imshow(I3)  
>> subplot(2,2,2); imshow(IR)  
>> subplot(2,2,3); imshow(IG)  
>> subplot(2,2,4); imshow(IB)
```

MATLAB muestra la figura 47, en la cual los títulos se insertaron desde las herramientas proporcionadas por el *imshow Toolbox*.

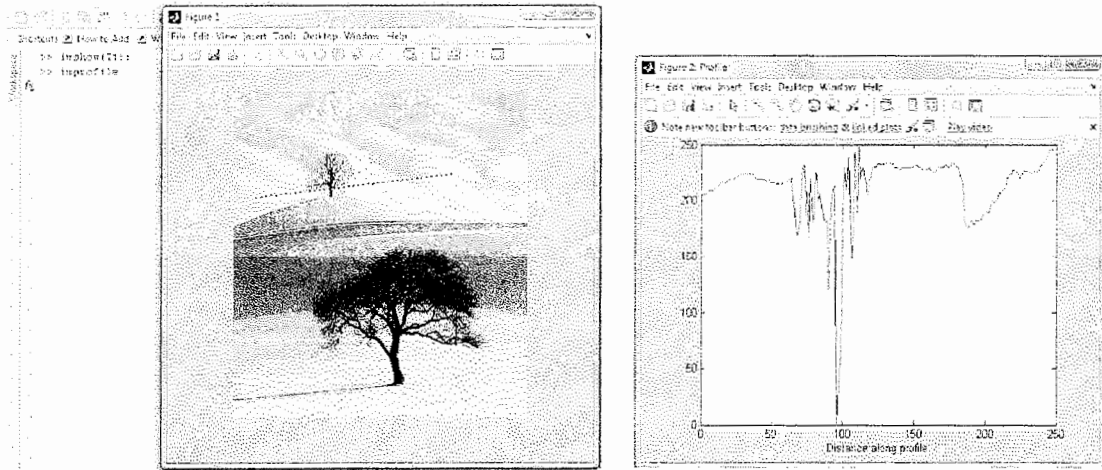
Figura 47. Planos de imagen RGB



Otra forma de conocer el valor de un pixel es utilizando la función *impixel*, posterior a tener mostrada la imagen a analizar, se escribe en la ventana de comandos `valor=impixel` y se seleccionan diferentes puntos de la imagen haciendo click en estos, una vez seleccionados los puntos de interés se presiona la tecla enter. En la variable `valor` se almacenan las diferentes intensidades para cada punto seleccionado, si es una imagen en escala de grises solamente existe un valor por pixel y si es otro tipo de imagen se muestra un valor para cada plano de la misma.

La función *improfile* por su parte, nos permite observar gráficamente las variaciones en la intensidad de color en una imagen. Para utilizarla se muestra la imagen previamente, se escribe *improfile* en la ventana de comandos y se hace *click* en el punto de inicio y en el punto final de la trayectoria en la cual se desea medir las variaciones de intensidad, posteriormente presionar *enter* y se mostrará una gráfica en la cual se muestran las variaciones vs la longitud de la trayectoria.

Figura 48. Variaciones de intensidad utilizando *improfile*



De trabajar con una imagen que posee más de un plano, se mostrará una gráfica del perfil por cada plano.

4.2. Procesamiento de imágenes

Teniendo en cuenta los conceptos básicos de manipulación de imágenes, se presentan funciones para el procesamiento de las mismas, que incluyen cambio del tipo de imagen, detección de bordes, detección de objetos y otras operaciones morfológicas.

4.2.1. Cambio de tipo de imagen y formatos de color

En la sección anterior se describieron diferentes tipos de imágenes dependiendo de los elementos que las conformen, también se mencionaron las imágenes en escala de grises e imágenes en colores como las *RGB*. Ahora se explica cómo transformar diferentes formatos de imágenes y el cambio entre los formatos de colores para poder realizar operaciones específicas en éstas.

En la tabla VI se muestran algunas funciones necesarias para el cambio de tipo de imagen según sus elementos y las entradas que éstas aceptan.

Tabla VI. Funciones para modificar formatos de imágenes

Función:	Conversión a:	Posibles entradas:
<i>im2uint8</i>	<i>uint8</i>	<i>logical, uint16, uint8 y double</i>
<i>im2uint16</i>	<i>uint16</i>	<i>logical, uint16, uint8 y double</i>
<i>im2double</i>	<i>Double</i>	<i>uint8, uint16 y double</i>
<i>im2bw</i>	<i>Logical</i>	<i>uint8, uint16 y double</i>

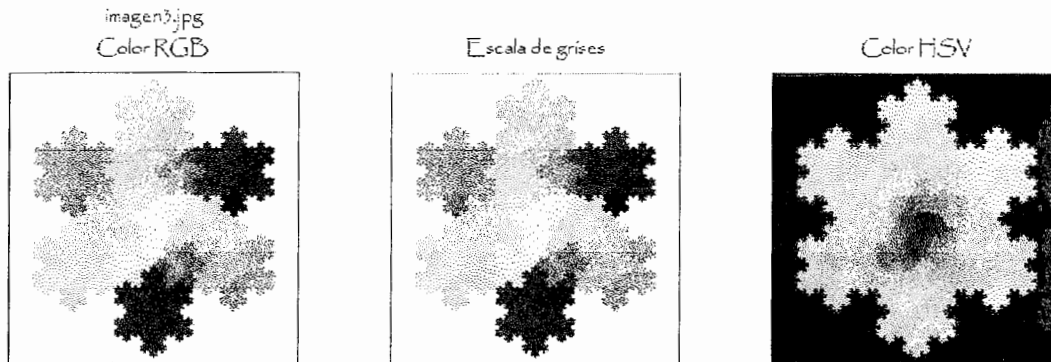
Existen funciones que realizan el cambio del tipo de color, si se posee una imagen *RGB* como la imagen3.jpg es posible convertirla a una escala de grises con la función *rgb2gray*. También se puede transformar de color *RGB* a formato *HSV* (*Hue Saturarion Value*), con la función *rgb2hsv*. El formato *HSV* es una transformación no lineal del espacio *RGB*, a veces llamado también *HSB* (*Hue Saturation Brightness*), para el proceso inverso se utiliza la función *hsv2rgb*. En el fragmento de código 4.7 se utilizan las funciones descritas para realizar cambios en los colores de la imagne3.jpg

Fragmento de código 4.7. Cambiando tipos de color

```
>> I3=imread('imagen3.jpg');
>> Igris=rgb2gray(I3);
>> Ihsv=rgb2hsv(I3);
>> subplot(1,3,1); imshow(I3)
>> subplot(1,3,2); imshow(Igris)
>> subplot(1,3,3); imshow(Ihsv)
```

La figura 49 muestra la imagen original y los resultados de cambiar los tipos de color.

Figura 49. **Imagen RGB, escala de grises y HSV**



Otra opción posible, es cambiar una imagen, ya sea a color o en escala de grises, a blanco y negro o imagen lógica, para esto se utiliza la función *im2bw*. Para cambiar una imagen a color, primero se transforma a escala de grises y posteriormente a imagen lógica.

Se cambia la imagen1.jpg a blanco y negro, la forma de hacer esto es $lbw=im2bw(I1,n)$ donde n debe ser un número entre 0 y 1. Este número representa el valor crítico (porcentaje del rango de la imagen) que asignará un 0 (negro) a valores menores a ese porcentaje n , o un 1 (blanco) a los valores sobre el porcentaje n . De no especificar el valor n , *MATLAB* trabaja por defecto con 0.5. El fragmento de código 4.8 modifica la imagen1.jpg a imágenes en blanco y negro con diferentes valores de n .

Fragmento de código 4.8. **Conversión a imágenes binarias**

```
>> I1=imread('imagen1.jpg');  
>> lbw1=im2bw(I1);  
>> lbw2=im2bw(I1,0.9);  
>> lbw3=im2bw(I1,0.1);  
>> subplot(2,2,1); imshow(I1)  
>> subplot(2,2,2); imshow(lbw1)
```

```
>> subplot(2,2,3); imshow(lbw2)
>> subplot(2,2,4); imshow(lbw3)
```

Figura 50. **Conversión de imágenes a blanco y negro**



La técnica descrita anteriormente se conoce como segmentación por umbral. Esta técnica es posible realizarla utilizando otros comandos, si se escribe `lbw=l1>=75`, donde `l1` posee almacenada `imagen1.jpg` (tipo `uint8`), en `lbw` quedan asignados en forma de 0 (negro) los valores entre 0 y 74 y 1 (blanco) los valores entre 75 y 255. Es posible obtener el negativo de la función anterior utilizando el comando `lbw=l1<75`. En la figura 51 se muestran los resultados de aplicar los comandos anteriores.

Figura 51. Segmentación por umbral



4.2.2. Detección de bordes

En ocasiones es necesario extraer los bordes de una imagen, para reconocer objetos o para determinar límites entre regiones adyacentes. La función utilizada con dicho propósito es *edge* y la cual permite diferentes algoritmos en la detección de bordes. A continuación se muestra el uso de dicha función en la imagen1.jpg utilizando los algoritmos *Canny*, *Sobel* y *Roberts*.

Fragmento de código 4.9. Detección de bordes

```
>> I=imread('im1rec.jpg');  
>> I1=edge(I,'canny');  
>> I2=edge(I,'sobel');  
>> I3=edge(I,'roberts');  
>> subplot(2,2,1); imshow(I)  
>> subplot(2,2,2); imshow(I1)  
>> subplot(2,2,3); imshow(I2)
```

```
>> subplot(2,2,4); imshow(I3)
```

Es importante aclarar que la imagen a la cual se extraerán los bordes debe ser en escala de grises. En la figura 52 se muestran las detecciones de bordes realizadas con el anterior código.

Figura 52. **Detección de bordes**



4.2.3. Operaciones morfológicas

Las operaciones morfológicas se basan en el trabajo de imágenes binarias para la detección o reconocimiento de formas. Estas operaciones dan como resultado otra imagen binaria cuyos píxeles representan la relación del píxel correspondiente con los vecinos en la imagen original. Las principales operaciones morfológicas son la erosión (*imerode* en *MATLAB*) y dilatación (*imdilate* en *MATLAB*). Éstas agregan o remueven píxeles de las fronteras de los objetos, ayudadas con una matriz definida previamente que determinará la manera de agregar o remover píxeles en la imagen original.

El comando para llevar a cabo estos procedimientos es *imerode(imagengrises, w)* o *imdilate(imagengrises,w)* donde *w* representa la matriz que determinará la manera de realizar la erosión o dilatación. En el fragmento de código 4.10 (guardado como un archivo *EyD.m*) se aplican las funciones a la imagen1.jpg, *w* se determina de dos maneras distintas, la primera es una matriz definida manualmente, la otra es una matriz determinada con la función *strel* y seleccionando un diamante dentro de una matriz de 5x5.

Fragmento de código 4.10. Erosión y dilatación en imagen (*EyD.m*)

```
%% Lectura de imagen
I1=imread('imagen1.jpg');

%% cambio a imagen binaria
Ibw=im2bw(I1, .15);

%% Creación de matrices
w1=zeros(3); % matriz 1
w1(2,2)=1
w2=strel('diamond',2) % matriz 2:

%% Erosión y dilatación
I1e=imerode(Ibw,w1);
I2e=imerode(Ibw,w2);

I1d=imdilate(Ibw,w1);
```



```

I2d=imdilate(Ibw,w2);

% Gráficas:
figure;
subplot(1,2,1); imshow(I1e)
subplot(1,2,2); imshow(I2e)

figure;
subplot(1,2,1); imshow(I1d)
subplot(1,2,2); imshow(I2d)

```

Al ejecutar el archivo se generan las siguientes matrices:

Fragmento de código 4.11. Matrices utilizadas en erosión y dilatación

```

w1 =
    0    0    0
    0    1    0
    0    0    0

w2 =
Flat STREL object containing 13 neighbors.
Neighborhood:
    0    0    1    0    0
    0    1    1    1    0
    1    1    1    1    1
    0    1    1    1    0
    0    0    1    0    0

```

A continuación se muestran las imágenes modificadas anteriormente, figura 53.

Figura 53. **Imágenes obtenidas por a) erosión y b) dilatación**

a)

Erosión con matriz w_1



Erosión con matriz w_2



b)

Dilatación con matriz w_1



Dilatación con matriz w_2



Es posible contar la cantidad de elementos que posee una imagen, para la cual se utiliza la función *bwlabel*, esta función debe contener una imagen binaria en la entrada y un tipo de conectividad, la cual indica que tan unidos se encuentren los pixeles con sus pixeles vecinos en la imagen original.

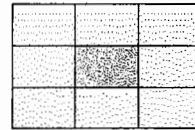
La conectividad puede tomar dos valores 4 u 8 como se muestra en la figura 54.

Figura 54. **Criterios de conectividad**

Conectividad 4:



Conectividad 8:

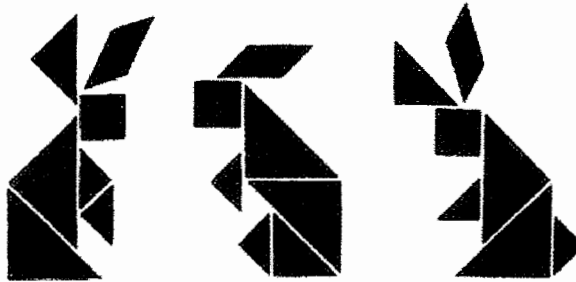


En conectividad 4 se dice que el píxel del centro pertenece a un objeto, si los cuatro píxeles grises alrededor son unos. En conectividad 8 se dice que el píxel del centro pertenece a un objeto si los píxeles grises alrededor son todos unos. La función *bwlabel*, realiza un etiquetado de los objetos pertenecientes a la imagen binaria original, ayudando a averiguar el número de elementos que se encuentran agrupados bajo uno de los criterios de conectividad.

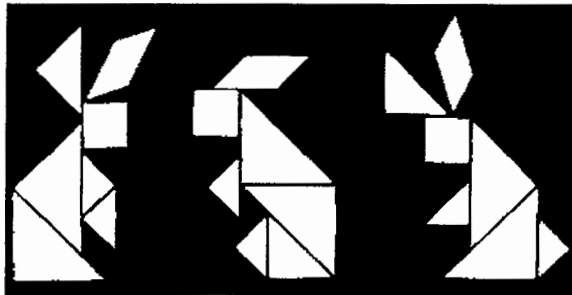
Se trabaja con la imagen en el siguiente fragmento de código para averiguar cuántos objetos de *tangram* (juego que consiste en formar siluetas de figuras con las siete piezas dadas) contiene la imagen4.jpg almacenada en la carpeta de trabajo actual, ésta se muestra en la figura 55a y su imagen binaria con una segmentación de umbral de 128 figura 55b.

Figura 55. a) imagen a procesar, b) imagen binaria

a)



b)



Fragmento de código 4.12. **Detección de objetos**

```
>> I4=imread('imagen4.jpg');  
lbw=I4<128;  
imshow(lbw)  
lobjetos=bwlabel(lbw,4);  
max(max(lobjetos))
```

```
ans =
```

```
21
```

Obteniendo un resultado de 21 objetos que conforman la imagen4.jpg. La aplicación de las funciones descritas hasta el momento varían mucho dependiendo del tipo de imágenes a procesar y del objetivo que se tenga al procesar la imagen. Una forma de ver esto es en el ejemplo descrito anteriormente, en el cual se quería contar las piezas de *tangram* que posee la imagen, para contar el número de imágenes que formen las piezas de *tangram* (en este caso tres conejos).

A continuación se muestra una función creada para poder elegir la imagen, el umbral de segmentación, la conectividad entre objetos y el tipo de detección en las regiones (dilatación y erosión). El comando de la función creada (*imobjet.m*) es `[w n llab]=imobjet(umbral,conect,a)`, donde el *umbral* es el utilizado para transformar la imagen a blanco y negro, *conect* es utilizado por la función `bwlabel` y para dilatación se escoge `a=1` y para erosión `a=2`. En las variable *w* se almacena la matriz utilizada en el proceso de dilatación o erosión, en este caso es una matriz en forma de *X* con tamaño 5×5 y en la variable *n* se almacena el número de objetos encontrados en la imagen analizada

Fragmento de código 4.13. Función para detección de objetos

```

% .....
% : DETECCIÓN DE OBJETOS :
% : LLAMADA: [w n llab]=imobjet(umbral,conect,a) :
% : umbral: umbral de segmentación (común entre 0 y 255):
% : conect: tipo de conexión entre objetos (4 u 8) :
% : a: dilatación (a=1) erosión (a=2) :
% : .....:

function[w n llab]=imobjet(umbral,conect,a)

%lectura y conversión de colores:

I=input('Leer imagen (escala de gris): '); %lectura de imagen
I=I<umbral; %conversión a binario

% Trabajo con dilatación o erosión

```

```

w=eye(5); %creación de matriz
w(5,1)=1; w(4,2)=1; w(2,4)=1; w(1,5)=1;

if a==1
    Id=imdilate(I,w); % dilatación
    figure(1); imshow(Id) % imagen con dilatación aplicada
    title('Imagen dilatada');
elseif a==2
    Id=imerode(I,w);
    figure(1); imshow(Id) % imagen con dilatación aplicada
    title('Imagen erosionada');
end

% etiquetando objetos

Ilab=bwlabel(Id,conect);
n=max(max(Ilab)); % contando el máximo número
                % de objetos etiquetados

```

A continuación se muestra el fragmento de código 4.14, necesario para obtener ayuda de la función y hacer uso de ésta con un umbral de segmento de 128, conectividad de 4 y haciendo uso de la erosión (a=2) para procesar la imagen4.jpg

Fragmento de código 4.14. Utilizando la función imobjet con a=2

```

>> help imobjet
.....
: DETECCIÓN DE OBJETOS :
: LLAMADA: [w n Ilab]=imobjet(umbral,conect,a) :
: umbral: umbral de segmentación (común entre 0 y 255) :
: conect: tipo de conexión entre objetos (4 u 8) :
: a: dilatación (a=1) erosión (a=2) :
:.....:

>> [w n Ilab]=imobjet(128,8,2)
Leer imagen (escala de gris): imread('imagen4.jpg')

```

```

w =

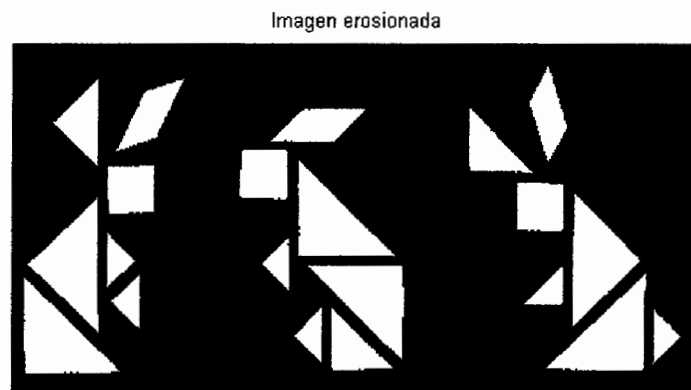
  1  0  0  0  1
  0  1  0  1  0
  0  0  1  0  0
  0  1  0  1  0
  1  0  0  0  1

n =
  21

```

La función da como resultado $n=21$, lo que concuerda con el resultado obtenido en el procedimiento anterior. En este caso, como se trabajó con erosión, los elementos de tangram fueron separados entre ellos, permitiendo distinguir cada uno con mayor facilidad. Este procedimiento es útil cuando se quiere contar objetos en una imagen que se encuentran muy estrechos entre sí. La imagen proporcionada por la función se muestra en la figura 56

Figura 56. **Imagen erosionada utilizando *imobjct***



Ahora se utiliza la *imobjct* para contar la cantidad de objetos formados por las piezas de *tangram* en conjunto (conejos). Para esto se escribe el comando `[w n llab]=imobject(128,8,1)`, con lo que se escoge una dilatación. Como se indica en el fragmento de código 4.15.

Fragmento de código 4.15. Utilizando imobjct con a=1

```
>> [w n llab]=imobject(128,8,1)

Leer imagen (escala de gris): imread('imagen4.jpg')

w =

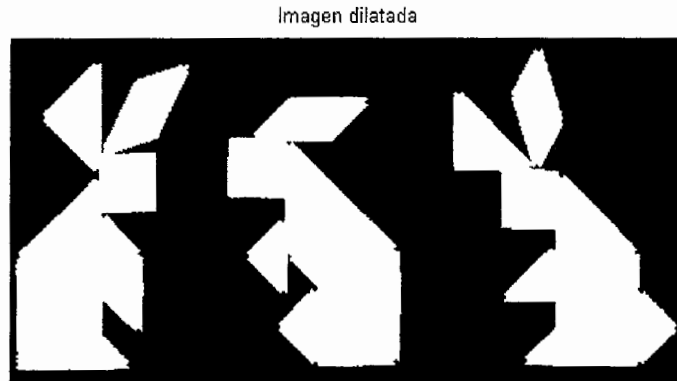
     1     0     0     0     1
     0     1     0     1     0
     0     0     1     0     0
     0     1     0     1     0
     1     0     0     0     1

n =

     3
```

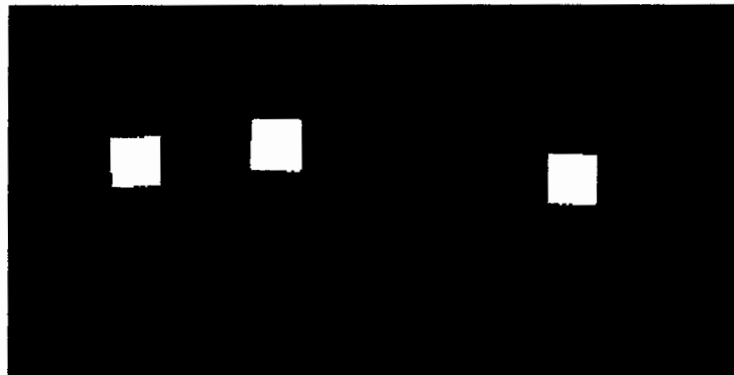
Obteniendo como resultado solamente 3 elementos, esto debido al procesamiento que se realizó en la imagen. Este procedimiento resulta útil cuando una imagen ha sido deteriorada en algunas regiones interiores a objetos que la componen. En la figura 57 se muestra la imagen dilatada.

Figura 57. **Imagen dilatada utilizando *imobjct***



Es posible hacer uso de la función *bwselect* para seleccionar objetos dentro de una imagen que se encuentre mostrada en *MATLAB*. Se escribe en la ventana de comando la función y posteriormente con el puntero se seleccionan los objetos que se desean mostrar aparte en una imagen. Por ejemplo, teniendo mostrada la imagen erosionada (Figura 4.18), es posible seleccionar los cuadros que posea dicha imagen, primero escribiendo la función *bwselect*, seleccionando los cuadrados (haciendo *click* dentro de ellos) y luego presionando *enter*. Al efectuar los pasos anteriores se posee la siguiente imagen mostrada en la figura 58

Figura 58. **Selección de objetos en una imagen**



Cuando se poseen etiquetados los objetos de una imagen es posible conocer algunas propiedades de cada uno, esto con la ayuda de la función *regionprops* (conocida como *imfeature* en versiones anteriores de *MATLAB*). Si se desean conocer los centroides de los 21 objetos etiquetados en la imagen erosionada de *tangram* se escribe `centro=regionprop(Ilab,'centroid')`. En la variable `centro` quedan almacenadas los centroides de todos los objetos etiquetados en forma de estructura. A continuación se muestra el fragmento de código 4.16 de la utilización de la función.

Fragmento de código 4.16. **Utilización de la función *regionprops***

```
>> [w n Ilab]=imobjet(128,8,2);
Leer imagen (escala de gris): imread('imagen4.jpg')
>> centro=regionprops(Ilab,'centroid')

centro =

21x1 struct array with fields:
    Centroid

>> centro(1)

ans =

    Centroid: [29.6533 190.8397]

>> centro(15)

ans =

    Centroid: [306.1843 68.7420]
```

En el ejemplo también se muestra como obtener los centroides individuales de cada objeto (objeto 1 y 15) en la imagen. A continuación se muestran todas las posibles propiedades que se pueden obtener con la función *regionprops* y se almacenan en la estructura *prop*, tabla VII.

Tabla VII. **Propiedades de la función regionprops**

'Area'	'PixelIdxList'	'Image'
'EulerNumber'	'ConvexArea'	'SubarrayIdx'
'Orientation'	'FilledArea'	'Eccentricity'
'BoundingBox'	'PixelList'	'MajorAxisLength'
'Extent'	'ConvexHull'	'EquivDiameter'
'Perimeter'	'FilledImage'	'MinorAxisLength'
'Centroid'	'Solidity'	
'Extrema'	'ConvexImage'	

Es posible seleccionar todas las propiedades de los objetos escribiendo *regionprops*(I_{lab}, 'all') como se muestra a continuación.

Fragmento de código 4.16. **Seleccionando propiedades de objetos**

```
>> prop=regionprops(Ilab, 'all');
>> prop(1)

ans =

    Area: 1921
   Centroid: [29.6533 190.8397]
 BoundingBox: [8.5000 150.5000 62 61]
 SubarrayIdx: {[1x61 double] [1x62 double]}
MajorAxisLength: 71.5763
MinorAxisLength: 40.9216
 Eccentricity: 0.8204
```

```
Orientation: -43.8475
ConvexHull: [11x2 double]
ConvexImage: [61x62 logical]
ConvexArea: 1971
    Image: [61x62 logical]
FilledImage: [61x62 logical]
FilledArea: 1921
EulerNumber: 1
    Extrema: [8x2 double]
EquivDiameter: 49.4560
    Solidity: 0.9746
    Extent: 0.5079
PixelIdxList: [1921x1 double]
PixelList: [1921x2 double]
Perimeter: 211.5807
```

```
>> prop(15)
```

```
ans =
```

```
    Area: 841
    Centroid: [306.1843 68.7420]
    BoundingBox: [292.5000 42.5000 41 41]
    SubarrayIdx: {1x2 cell}
MajorAxisLength: 48.2597
MinorAxisLength: 26.9865
Eccentricity: 0.8290
Orientation: -45.7943
    ConvexHull: [11x2 double]
    ConvexImage: [41x41 logical]
    ConvexArea: 872
```

```

Image: [41x41 logical]
FilledImage: [41x41 logical]
FilledArea: 841
EulerNumber: 1
Extrema: [8x2 double]
EquivDiameter: 32.7230
Solidity: 0.9644
Extent: 0.5003
PixelIdxList: [841x1 double]
PixelList: [841x2 double]
Perimeter: 137.3970

```

Se agrega el fragmento de código mostrado a continuación a la función *imobjet.m* (fragmento de código 4.17), con el fin de que encierre en rectángulos los *BoundingBox* de cada elemento y muestre los centroides de estos, mostrándolos en la imagen original y en la imagen etiquetada.

Fragmento de código 4.17. Dibujar rectángulos en objetos

```

%% Propiedades de los objetos
prop=regionprops(I1ab,'all');

%% Agregar centroides y rectángulos

hold(imgca,'on')
centro = cat(1, prop.Centroid);
plot(imgca,centro(:,1), centro(:,2), 'r*') %agregar centroides
for i=1:size(prop,1) %agregar rectángulos

rectangle('Position',prop(i).BoundingBox,'EdgeColor','b','LineWidth',2)
end
hold(imgca,'off')

%% Agregar en imagen original

figure(2);
imshow(I1)

```

```

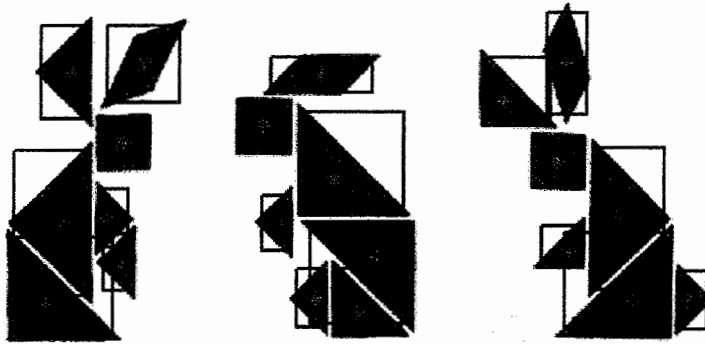
hold(imgca,'on')
centro = cat(1, prop.Centroid);
plot(imgca,centro(:,1), centro(:,2), 'r*') %agregar centroides
for i=1:size(prop,1) %agregar rectángulos
    rectangle('Position',prop(i).BoundingBox,'EdgeColor','b','LineWidth',2)
end
hold(imgca,'off')

```

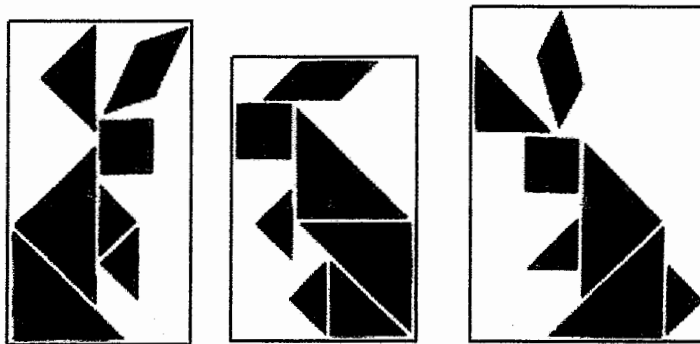
Al aplicar a la imagen4.jpg se obtienen las siguientes imágenes mostradas en la figura 59a para erosión y 59b para dilatación.

Figura 59. **Objetos resaltados a) erosión, b) dilatación**

a)



b)



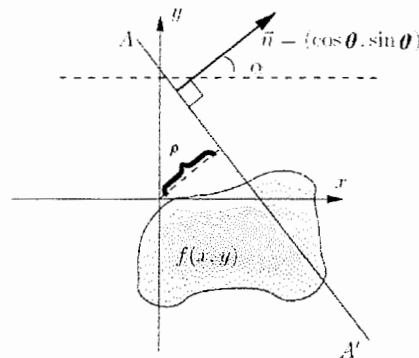
4.3. Transformada de Radon

La transformada de Radon es una transformación lineal de una función sobre un conjunto de rectas. Es muy utilizada en tomografía, método para creación de imágenes por medio de datos almacenados de mediciones en secciones transversales de un objeto. La tomografía se utiliza ampliamente en medicina, arqueología, geología, biología, oceanografía, ciencia de los materiales, entre otras ciencias. Pueden realizarse utilizando diferentes técnicas, tales como rayos X, rayos gamma, ultrasonido, aniquilación de partículas o resonancia magnética.

Si $f(x, y)$ representa una densidad desconocida en una determinada región, entonces la transformada de Radon representa los datos almacenados a la salida de un escaneo tomográfico. De esta manera, utilizando la transformada inversa es posible reconstruir una imagen de la densidad $f(x, y)$ si se conocen los datos almacenados de una medición.

Si una recta se representa como $x \cos \theta + y \sin \theta = \rho$, siendo ρ la menor distancia de la línea hasta el origen y θ es el ángulo que forma la normal de la recta con el eje x , como se ve en la figura 60.

Figura 60. Elementos en la transformada de Radon



La definición formal de la transformada de Radon para un espacio bidimensional se escribe como

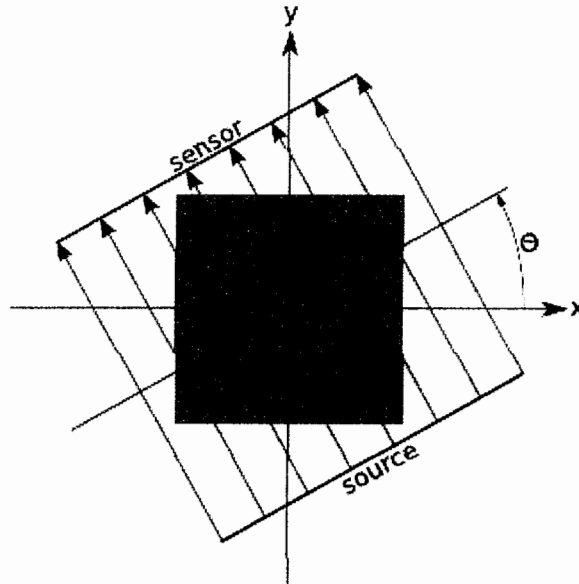
$$\mathcal{R}\{f(x, y)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy = R(\theta, \rho) \quad (4.1)$$

Se observa que se transforma de un dominio de rectas (x, y) en coordenadas cartesianas en variables (ρ, θ) en coordenadas polares. La fórmula de la transformada en un espacio bidimensional y tridimensional fue propuesta por Johann Radon en 1917, quien también planteó la transformada inversa. Si se habla de un espacio n-dimensional la transformada es la integral de una función en hiperplanos y se le denomina transformada de rayos x.

La transformada de Radon suele llamarse Senograma en el contexto de tomografías, debido que la transformada de una función delta es un seno, por lo que la representación gráfica de la transformada de un conjunto de pequeños objetos se asemeja a un conjunto de senos con diferentes amplitudes y fases.

A continuación se describe el uso común que se le da a la transformada de Radon. En la figura 61 se representa la región que es inspeccionada con alguna de las técnicas antes mencionadas de tomografía.

Figura 61. **Representación del uso de la transformada de Radon**



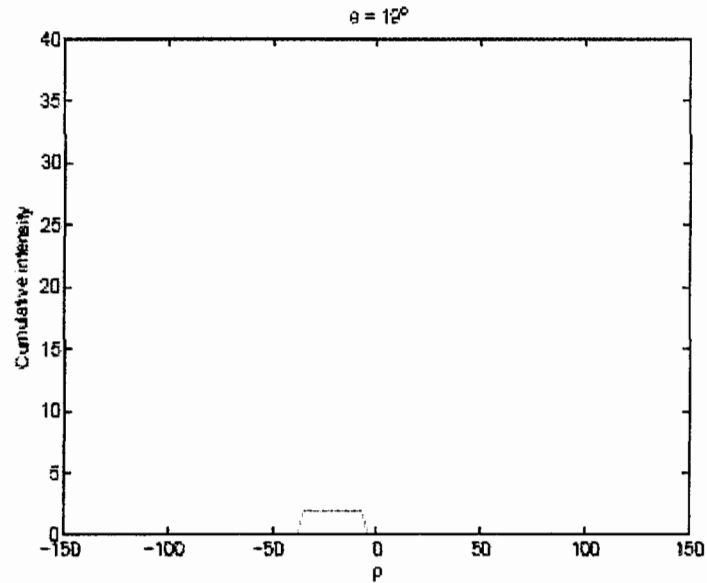
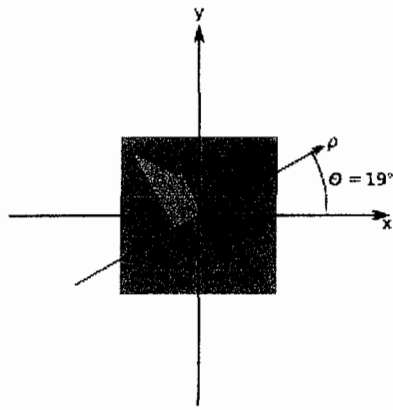
Fuente: Carsten Hoiland, *The Radon transform*, p. 5

La fuente y el sensor van rotando respecto al centro del objeto inspeccionado, para cada ángulo la densidad de rayos es almacenada en el sensor. El ángulo normalmente comprende desde los 0° hasta los 180° , de esta manera, para cada ángulo y cada recta inspeccionada se obtienen intensidades diferentes en el sensor, que se van acumulando con los resultados anteriores comprendiendo al final la transformada $R(\rho, \theta)$.

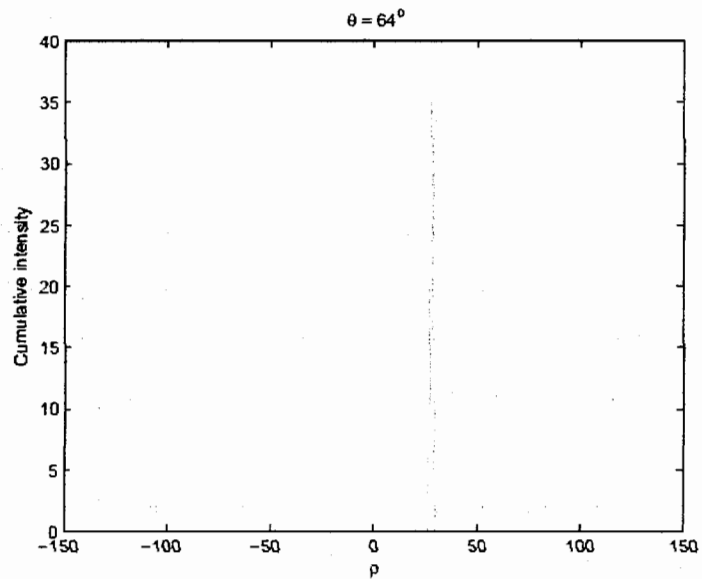
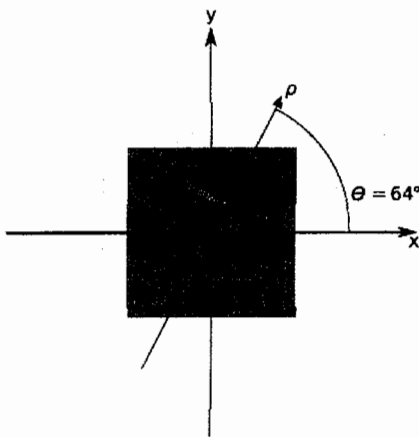
En la figura 62 se muestran mediciones para 19° y para 64° , cada una con su respectiva transformada.

Figura 62. Transformada para un ángulo de a) 19° y b) 64°

a)



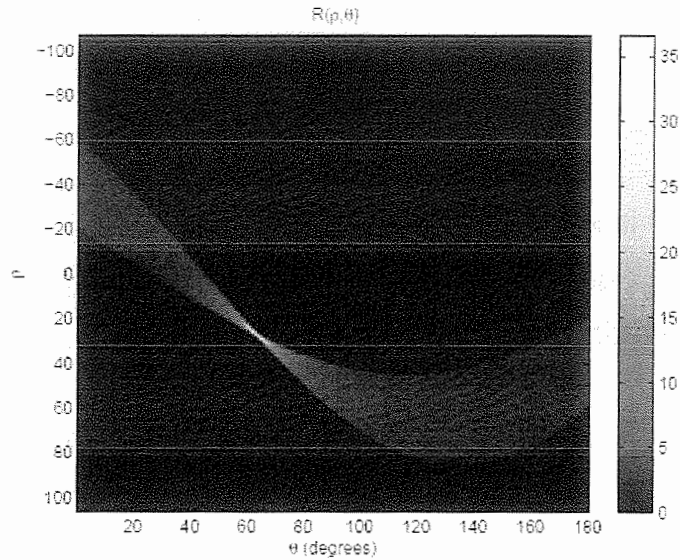
b)



Fuente: Carsten Hoilund, *The Radon transform*. p. 6 y 7

En la figura 63, se muestra la transformada completa de Radon, la región en tono blanco de la imagen muestra la distancia y el ángulo para los cuales la intensidad es mayor.

Figura 63. Representación gráfica de la transformada de Radon



Fuente: Carsten Hoiland, *The Radon transform*. p. 8

La transformada de Radon se encuentra relacionada con la transformada de Fourier bidimensional de variable $\mathbf{x} = (x, y)$ definida por

$$\hat{f}(\mathbf{w}) = \int_{-\infty}^{\infty} f(\mathbf{x}) e^{-2\pi i \mathbf{x} \cdot \mathbf{w}} dx dy \quad 4.2$$

Por conveniencia se deja constante la variable θ en la ecuación 4.1, y se cambia la nomenclatura de la transformada por una función únicamente de variable ρ , de esta manera $R(\rho, \theta) = \widehat{R_{\theta}(\rho)}$ y es posible escribir el teorema de secciones de Fourier:

$$\widehat{R_{\theta}(\rho)} = \hat{f}(\rho \mathbf{n}(\theta)) \quad 4.3$$

Con $\mathbf{n}(\theta) = (\cos \theta, \sin \theta)$.

En este punto es posible conocer la transformada inversa de Radon para los espacios en los cuales sea posible, esto utilizando la fórmula inversa de Fourier. Sin embargo, la igualdad 4.3 no es particularmente aplicada a un método numérico en el que se utilice Fourier.

Existe un algoritmo rápido para la transformada inversa discreta de Radon bidimensional, este algoritmo se conoce como retroproyección filtrada. Primero se considera el operador adjunto de R

$$R^*[g](\mathbf{x}) = \int_0^{2\pi} g(\theta, \mathbf{n}(\theta) \cdot \mathbf{x}) d\theta \quad 4.4$$

R recibe el nombre de retroproyector puesto que toma las proyecciones sobre rectas y las esparce o retro-proyecta para producir una imagen. El siguiente paso es definir un filtro rampa h de una variable como

$$\widehat{H[\hat{h}]}(\omega) = |\omega| \hat{h} \quad 4.5$$

Si se aplica el teorema de secciones de Fourier (ecuación 4.3) y se cambia la variable de integración, se observa que para f y $g(\rho, \theta) = R[f(x, y)]$

$$f(x, y) = \frac{1}{4} R^* H[g(\rho, \theta)] \quad 4.6$$

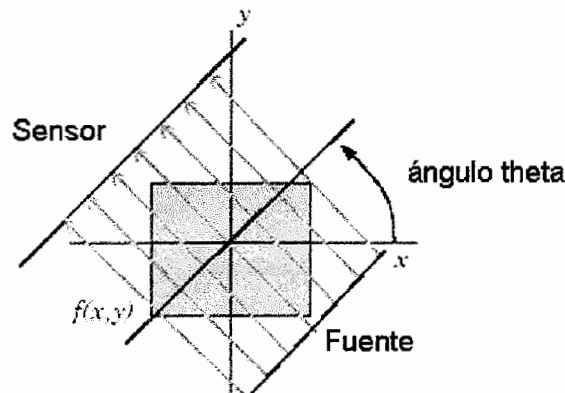
Se puede concluir que la imagen representada con f puede ser recuperada del Sinograma g al aplicar un filtro $H[g(\rho, \theta)]$ y retroproyectando. Es posible implementar el filtrado con técnicas de procesamiento de imágenes y el retroproyectado es la acumulación pixeles en la imagen que se desea reconstruir. Estas técnicas se describen en la sección 4.3.2.

4.3.1. Transformada de Radon en *MATLAB*

Una proyección de una función bidimensional $f(x,y)$, que normalmente representa una imagen, es un conjunto de integrales de líneas, en donde la función *radon* en *MATLAB* computa la integral de línea para múltiples fuentes a lo largo de direcciones paralelas o rayos en cierta dirección. Los rayos están espaciados un pixel uno de otro.

Con el fin de mostrar transformada de Radon, *MATLAB* toma múltiples proyecciones de rayos paralelos que atraviesan a la función $f(x,y)$ mientras la fuente de rayos rota alrededor del centro de la imagen. En la figura 64 se muestra una proyección única para determinado ángulo de rotación.

Figura 64. **Proyección de ángulos paralelos a determinado ángulo**

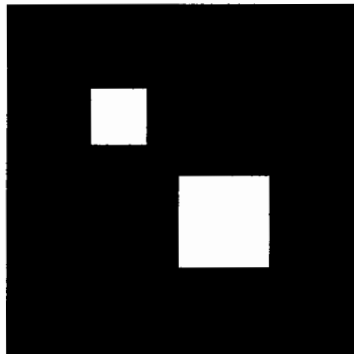


La sintaxis utilizada para obtener la transformada R a una imagen I a θ grados es $[R, xp]=radon(I, \theta)$. Si se considera θ como un escalar, R representa la transformada para ese ángulo en específico. En cambio, si θ es un vector R es una matriz, en la que cada columna es la transformada para cada ángulo que comprende θ . Si en el comando se omite la entrada θ , *MATLAB* toma por defecto un vector 0:179.

Por otra parte, en la variable `xp` se asignan las coordenadas radiales correspondientes a cada fila en `R`. Estas coordenadas son los valores a lo largo del eje x' orientado a θ grados en sentido de las agujas del reloj a partir del eje x . El origen de ambos ejes, es el centro de la imagen `I`, definido por `floor((size(I)+1)/2)`.

En la figura 65 se muestra una imagen creada en *MATLAB* a la cual se le calcula la transformada de Radon a 0° y a 45° según se indica en el fragmento de código 4.18.

Figura 65. **Imagen creada para aplicar la transformada de Radon**



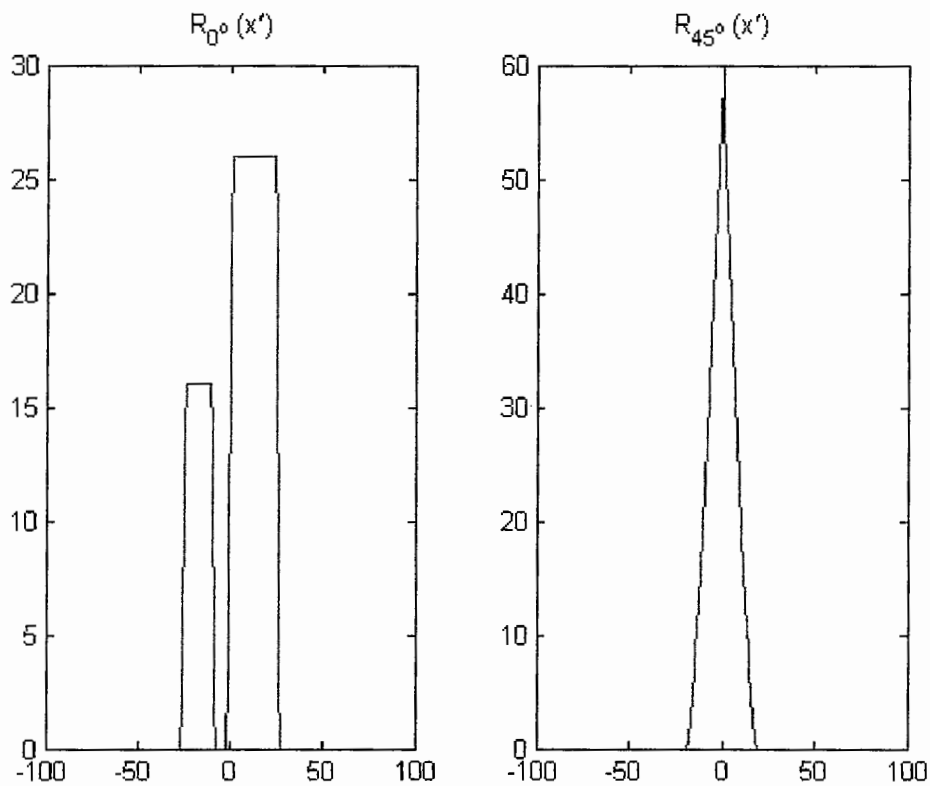
Fragmento de código 4.18. **Transformada de Radon en ángulos específicos**

```
>> I = zeros(100,100);  
>> I(25:40, 25:40) = 1;  
>> I(50:75, 50:75) = 1;  
>> imshow(I)  
>> [R1,xp1]=radon(I,0);  
>> [R2,xp2]=radon(I,45);  
>> figure;  
>> subplot(1,2,1);plot(xp1,R1);title('R_{0^o} (x\prime)')
```

```
>> subplot(1,2,2);plot(xp2,R2);title('R_{45^o} (x\prime)')
```

Las gráficas obtenidas de las transformadas para los ángulos mencionados se muestran en la figura 66

Figura 66. **Transformada de Radón en ángulos específicos**



Al considerar todos los ángulos de medición al aplicar el siguiente fragmento de código a la imagen I.

Fragmento de código 4.18. **Calculando transformada completa**

```
>> theta=[0:179];  
>> [R,xp]=radon(I,theta);  
>> figure; imagesc(theta,xp,R)  
>> title('R_{\theta} (X\prime)');
```

```

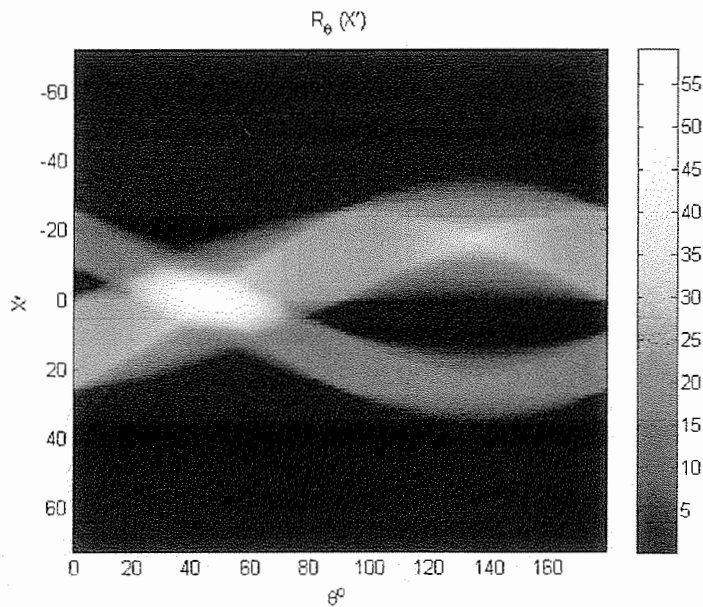
>> xlabel('\theta^o');
>> ylabel('X\prime');
>> set(gca,'XTick',0:20:179);
>> colormap(hot);
>> colorbar

```

Obteniendo la figura 67 mostrada a continuación.

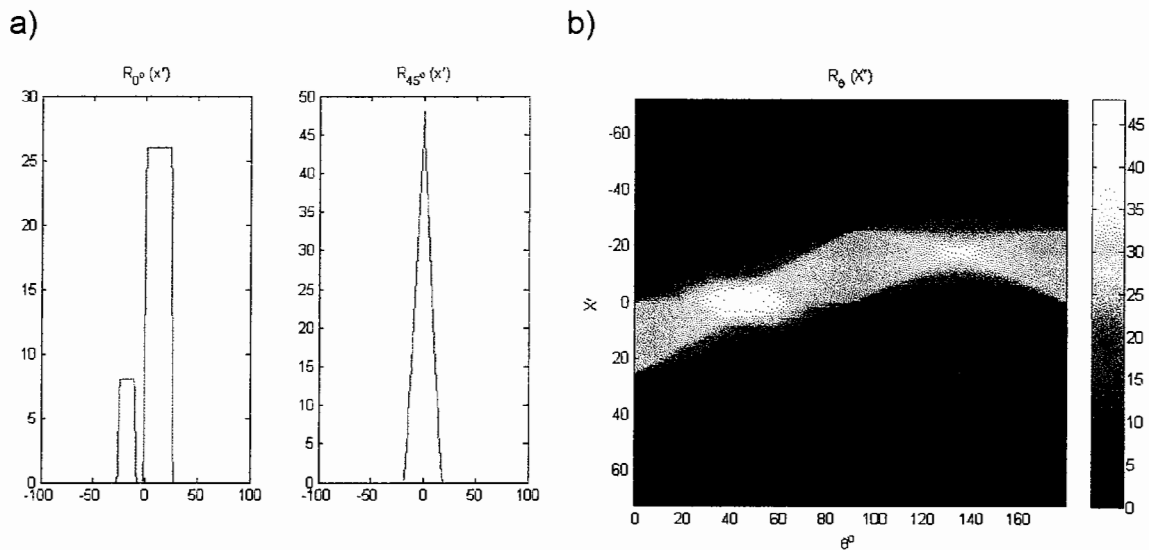
Figura 67. **Representación gráfica para la transformada completa de**

Radon



No es necesario que la imagen I sea lógica, puede ser en escala de grises. cambiando algunos valores $I(25:40, 25:40) = 0.5$, se obtienen los resultados que muestra la figura 68a para los ángulos especificados en el fragmento de código y la figura 68b para la transformada completa. Como se observa la transformada toma en cuenta la intensidad de color dentro de la imagen.

Figura 68. Transformada de Radon con intensidades en la imagen original para a) 0° y 45°, b) todos los ángulos



4.3.2. Transformada inversa de Radon en *MATLAB*

La función *iradon* se utiliza para construir las proyecciones angulares de una imagen I en una aproximación de ésta. La sintaxis del comando es $IR=iradon(R,theta)$. En la mayoría de aplicaciones no se posee la imagen original de las proyecciones, por lo que es muy utilizada en técnicas de reconstrucción de imágenes por tomografía.

La función utiliza un algoritmo de retroproyección filtrada para computar la transformada inversa de Radon. Este algoritmo forma una aproximación de la imagen I basándose en las proyecciones almacenadas una columna de la matriz R .

Es posible especificar todos los parámetros deseados en la transformada inversa bajo la sintaxis $IR=iradon(R, \theta, interp, filter, frequency_scaling, output_size)$. De no escribir todos los parámetros, *MATLAB* tomará por defecto los mismos para la construcción de I . Las posibles opciones de *interp* para la retroproyección se muestran en la tabla VIII.

Tabla VII. Opciones de interpolación en la retroproyección

Opción:	Descripción:
'nearest'	Interpolación del vecino más cercano.
'linear'	Interpolación lineal (por defecto).
'spline'	Interpolación de ranura.
'pchip'	Interpolación de preservación de forma cúbica.
'cubic'	Igual que 'pchip'
'v5cubic'	Interpolación cúbica de <i>MATLAB5</i> , la cual no extrapola y usa 'spline' si X no se encuentra igualmente espaciado.

El parámetro *filter* dentro del comando, especifica el filtrado en el dominio de la frecuencia. Las opciones se muestran en la tabla IX.

Tabla IX. Opciones de filtrado

Opción:	Descripción:
'Ram-Lak'	O filtro rampa (opción por defecto). La respuesta en frecuencia es $ f $ debido a que es sensible a la proyección del ruido. Se recomienda no utilizar este filtro.
'Shepp-Logan'	Multiplica un filtro <i>Ram-Lak</i> por una función <i>sinc</i>
'Cosine'	Multiplica un filtro <i>Ram-Lak</i> por una función <i>cosine</i>
'Hamming'	Multiplica un filtro <i>Ram-Lak</i> por una ventana de <i>Hamming</i> .
'Hann'	Multiplica un filtro <i>Ram-Lak</i> por una ventana de <i>Hann</i>
'None'	Sin filtrado. Se obtiene una retroproyección no filtrada.

El parámetro *frequency_scaling* es un escalar entre (0,1] que modifica al filtro en su eje de frecuencia. Por defecto se toma el 1. Si se especifica un número menor que 1, el filtro se comprime en el rango [0,*frequency_scaling*].

Sí el parámetro *out_size* especifica el número de filas y columnas que se mostraran de la imagen reconstruida (no hace de menor o mayor tamaño la imagen). Si no se especifica, se tomará por defecto el largo de las proyecciones $2*\text{floor}(\text{size}(R,1)/(2*\text{sqrt}(2)))$.

Con el fin de mostrar ejemplos del *iradon*, se utiliza la función *phantom* para crear una imagen de una cabeza ilusoria en dos dimensiones. La imagen obtenida es en escala de grises y contiene una elipse grande (que puede representar el cerebro en una tomografía) con muchas elipses en su interior (que representan las características en el cerebro). En el fragmento de código 4. 18 muestra diferentes opciones para la reconstrucción de una imagen *phantom*.

Fragmento de Código 4. 18. **Reconstrucción de una imagen**

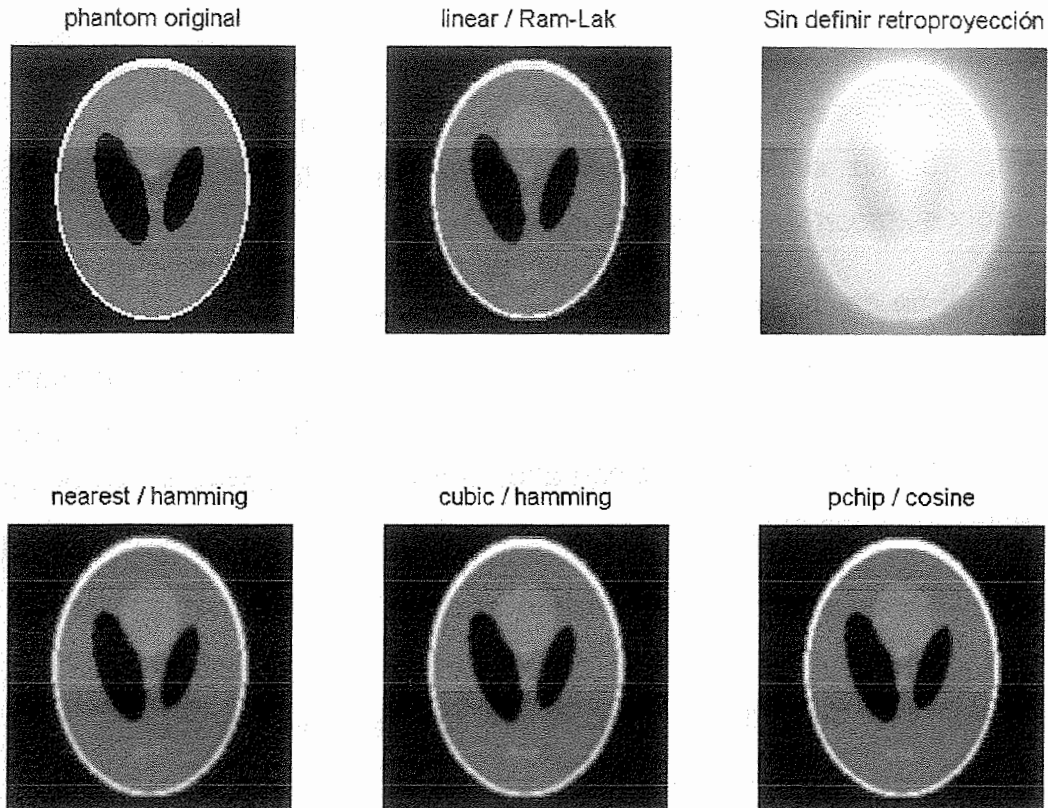
```
% creación y transformación de imagen.
P = phantom(128);
R = radon(P,0:179);

% conjunto de transformadas inversas
I1 = iradon(R,0:179);
I2 = iradon(R,0:179,'linear','none');
I3 = iradon(R,0:179,'nearest','hamming');
I4 = iradon(R,0:179,'cubic','hamming');
I5 = iradon(R,0:179,'pchip','cosine');

% graficas.
subplot(2,3,1), imshow(P), title('phantom original')
subplot(2,3,2), imshow(I1), title('linear / Ram-Lak')
subplot(2,3,3), imshow(I2,[]), title('Sin definir retroproyección')
subplot(2,3,4), imshow(I3), title('nearest / hamming')
subplot(2,3,5), imshow(I4), title('cubic / hamming')
subplot(2,3,6), imshow(I5), title('pchip / cosine')
```

En la figura 69 puede observarse el conjunto de imágenes reconstruidas:

Figura 69. Conjunto de imágenes reconstruidas con *iradon*



4.4. *Ultrasonic Signal Processing Toolbox*

El paquete de procesamiento de señales ultrasónicas –*USPT* por sus siglas en inglés– no es incluido en *MATLAB*, éste fue diseñado en el área de señales y sistemas del departamento de ciencias de la ingeniería, en la universidad de *UPSSALA*, Suecia por Lars Ericsson y Tomas *Olofsson*. El *software* puede ser utilizado únicamente con fines de investigación.

El objetivo principal del *USPT* es eliminar el ruido de las mediciones realizadas en ensayos no destructivos de materiales. Es operado desde una interfaz gráfica y fue diseñado para ser utilizado fácilmente por personas que trabajen en dichos ensayos.

A continuación se describe brevemente el funcionamiento del *USPT*. Si es de interés se sugiere revisar detenidamente el manual de usuario proporcionado en el sitio descrito adelante.

4.4.1. Obtención e instalación del paquete

El *USPT* es un software libre y se puede obtener fácilmente¹. En la sección de descarga del índice es posible obtener el archivo en formato *rar* o *zip*.

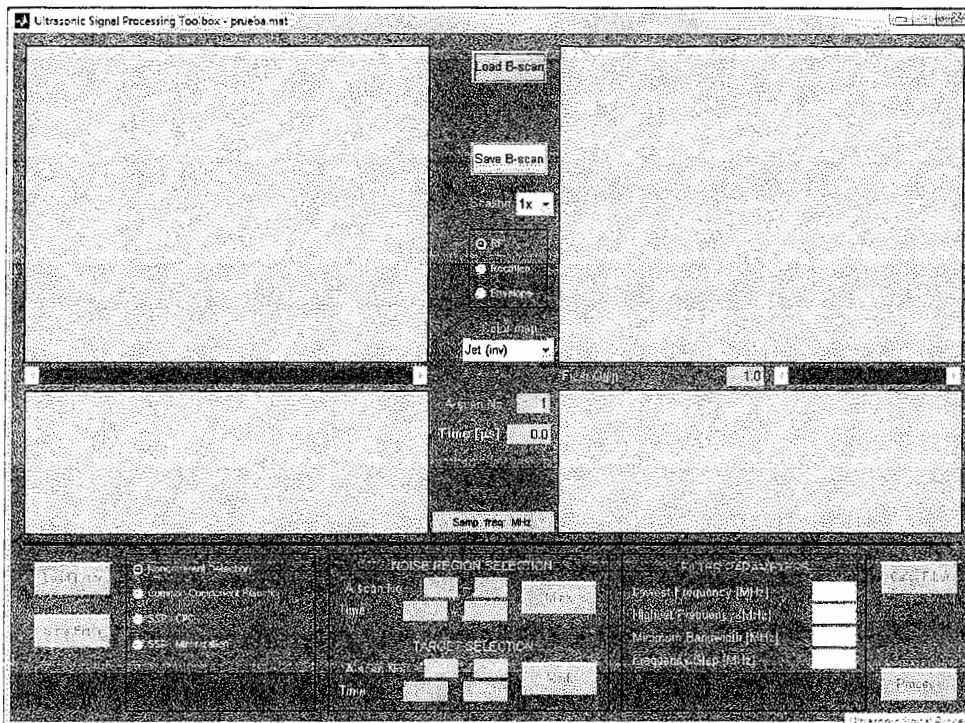
El sitio también proporciona información detallada de los autores y el manual de usuario del *software*. Al momento de descargarlo no es necesario realizar ninguna operación para la instalación más que descomprimir el archivo en la carpeta actual de trabajo de *MATLAB*, lo que proporciona todos los archivos necesarios para la utilización del *Toolbox*.

¹sitio de la universidad de *UPSSALA*, área de procesamiento digital de señales, *USPT*
<<http://www.signal.uu.se/Toolbox/USPT/>>.

4.4.2. Descripción general

Para correr el *USPT* se escribe en la ventana de comandos *ultra*. A continuación aparece la interfaz gráfica mostrada en la figura 70.

Figura 70. Interfaz gráfica de usuario del *USPT*



El *software* puede leer datos que representen escaneo tipo B (ver sección 2.4 capítulo 2), por lo que pueden cargarse datos que representen volúmenes en 3 dimensiones. Posteriormente a la lectura de los datos estos pueden ser observados (como escaneo tipo B y tipo A), procesados y posteriormente ser guardados con los cambios deseados.

4.4.3. Formatos aceptados

Los formatos que pueden ser leídos por el Toolbox son:

- Nativo de *MATLAB* (.mat)
- *Ultra Optec* (.scn)
- Equipo de ensayo no destructivo (.ndt)

Es posible cambiar o agregar nuevos formatos de lectura cambiando los archivos *loaddata.m* y *loadatb.m*. En la sección que inicia con *elseif strcmpi* (*extension,'xxx'*) en donde *xxx* debe ser reemplazado por la nueva extensión que se desea agregar.

4.4.4. Opciones de procesamiento de señales

El *software* cuenta con cuatro algoritmos para el procesamiento de las señales, los cuales se basan en la supresión de ruido "*grain*". Los dos primeros algoritmos son el detector no coherente y el rechazador de componente común, el tercero es el procesamiento de espectro dividido que es representado por un algoritmo de minimización de ruido tradicional y el último es una modificación del algoritmo de umbral de polaridad, denominado coincidencia de polaridad consecutiva.

4.5. Resumen de Comandos y Funciones

A continuación se muestra un resumen de los comandos y funciones de *MATLAB* utilizados en el presente capítulo. Estos se agrupan por función desempeñada dentro del capítulo y se da una breve descripción de los mismos.

4.5.1. Funciones básicas y de uso general

- **cat**: se utiliza para concatenar matrices.
- **end**: termina un código de bloque.
- **eye**: crea una matriz identidad.
- **figure**: despliega un objeto gráfico (ventana) en donde *MATLAB* muestra una salida gráfica.
- **floor**: aproxima valores decimales al entero menor más cercano.
- **for**: ejecuta una o más instrucciones de *MATLAB* en un ciclo.
- **hold**: determina si nuevos objetos gráficos son agregados o reemplazados en una gráfica.
- **if**: en base al cumplimiento de una condición lógica, realiza una o más instrucciones de *MATLAB*.
- **max**: proporciona el valor máximo existente en una matriz.
- **min**: proporciona el valor mínimo existente en una matriz.
- **plot**: traza gráficas continuas de vectores.
- **rand**: crea una matriz con valores pseudoaleatorios en base a una distribución estándar en el intervalo (0,1).
- **rectangle**: dibuja un rectángulo en especificado origen y dimensiones.
- **set**: establece las propiedades nombradas de objetos.
- **subplot**: divide un *figure* actual en paneles rectangulares.
- **title**: coloca texto en la parte superior de una gráfica.

- **whos:** despliega en orden alfabético todas las variables existentes en el escritorio de trabajo actual.
- **xlabel:** coloca texto en el eje x de un *figure*.
- **ylabel:** coloca texto en el eje y de un *figure*.
- **zeros:** crea una matriz con todos sus elementos igual a cero.

4.5.2. Funciones para el trabajo y procesamiento de imágenes

- **bwlabel:** etiqueta objetos relacionados entre sí, dentro de una imagen.
- **bwselect:** proporciona una imagen binaria que contiene objetos que se superponen entre sí, dentro de una imagen.
- **colorbar:** despliega la barra de color en una figura y cambia el tamaño de los ejes para colocarla.
- **colormap:** asigna especificada escala de colores a una imagen.
- **edge:** calcula los bordes detectados en una imagen lógica o en escala de grises.
- **histeq:** mejora el contraste de imágenes.
- **im2bw:** transforma una imagen en escala de grises a una imagen binaria.
- **im2double:** transforma una imagen de intensidad a una imagen de doble precisión (punto flotante)
- **im2uint16:** transforma una imagen de intensidad a formato *uint16*.
- **im2uint8:** transforma una imagen de intensidad a formato *uint8*.
- **image:** crea un objeto gráfico de imagen interpretando cada elemento de una matriz como índice en el mapa de colores de la figura actual o como *RGB* por defecto.
- **imagesc:** escala datos de una matriz en una imagen, cubriendo toda la gama del mapa de colores actual.

- ***imcrop***: se utiliza para realizar cortes en imágenes de tamaño especificado.
- ***imdilate***: dilata bordes de los objetos contenidos en imágenes binarias o en escala de gris.
- ***imerode***: erosiona bordes de los objetos contenidos en imágenes binarias o en escala de gris.
- ***imhist***: despliega el histograma de una imagen por encima de la escala de grises.
- ***imfinfo***: muestra la información de una imagen almacenada como archivo.
- ***impixel***: proporciona el valor en *RGB* de un pixel en una imagen.
- ***improfile***: calcula la intensidad de cambio de color a lo largo de una línea dentro de una imagen.
- ***imread***: lee una imagen almacenada en un archivo con nombre especificado.
- ***imresize***: proporciona una imagen con un tamaño menor a la imagen de entrada.
- ***imshow***: despliega una imagen en una ventana con funciones básicas para manipulación de la misma.
- ***imtool***: despliega una imagen en una ventana con opciones avanzadas para el procesamiento de imágenes.
- ***imwrite***: guarda matrices como imágenes en un archivo con nombre y formato especificado.
- ***regionprops***: proporciona las propiedades de objetos con cierta conexión entre sí dentro de una imagen binaria.
- ***phantom***: genera una imagen representativa de un cerebro.

4.5.3. Funciones para el trabajo con la transformada de Radon

- ***iradon***: reconstruye una imagen en base a datos de proyecciones en una matriz de dos dimensiones.
- ***radon***: se utiliza para calcular la transformada de Radon de una imagen escalada.

4.5.4. Funciones impropias de *MATLAB*

- ***imobjct***: función creada para procesar una imagen, proporcionando el número de objetos y encontrados en dicha imagen y los parámetros de éstos.
- ***ultra***: función para abrir el *Ultrasonic Processing Toolbox*, una vez se encuentre instalado.

5. MONTAJES EXPERIMENTALES

Este capítulo consiste en la descripción de dos montajes experimentales, ambos pretenden poner en práctica algunas consideraciones teóricas de los primeros dos capítulos y utilizar técnicas y procedimientos descritos en el segundo, tercer y cuarto capítulo.

5.1. Descripción general

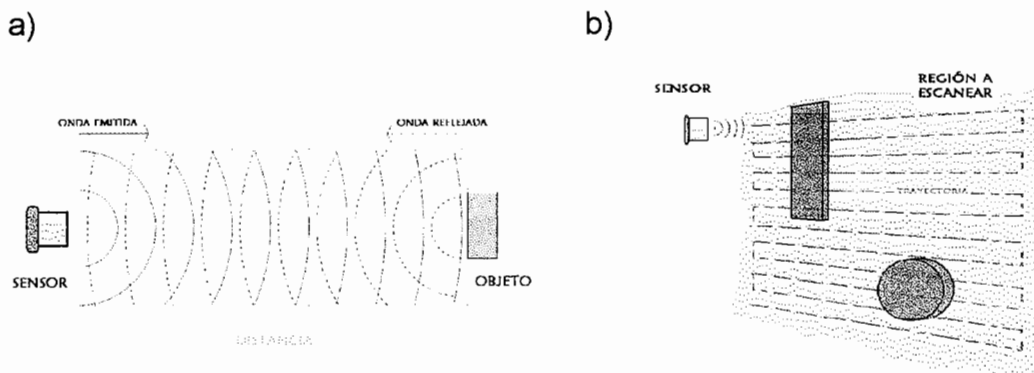
El primer montaje consiste en la utilización de la técnica pulso-eco, para medir distancias entre objetos y el sensor utilizado, procedimiento conocido comúnmente como sonar (ver sección 1.6.3 y 2.3.2). El segundo aprovecha los resultados del primer montaje para realizar un escaneo de una región y a través de la representación tipo B y C (ver sección 2.4.2 y 2.4.3). Posteriormente crear un mapa de contorno aproximado de la región trabajada y teniendo la información almacenada, el sistema sea capaz de analizar los resultados para conformar un sistema de visión artificial o visión por computador.

Ambos sistemas se ayudan de *MATLAB* para almacenar las mediciones y posteriormente procesar y analizar estas con técnicas descritas en el capítulo anterior.

5.1.1. Descripción para la obtención de mediciones

En la figura 71 se representan las maneras en las que se dispone el sensor para tomar mediciones según los procedimientos descritos anteriormente.

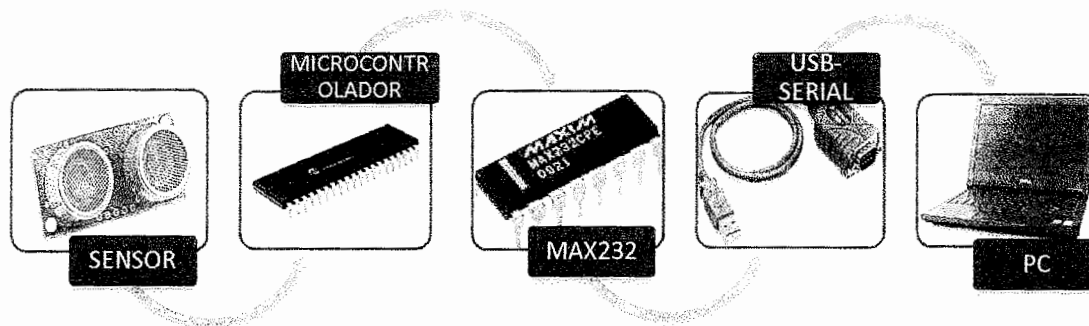
Figura 71. Disposición del sensor para a) sonar, b) creación de mapa de contorno



5.1.2. Diagrama de bloques

En el capítulo 3 se proporciona la información necesaria para el control del sensor y comunicación del mismo con la computadora – para los montajes descritos se utiliza el sensor ultrasónico *Devantech SRF04* (ver sección 3.1) –. En la figura 68 se muestra el diagrama de bloques general que se utiliza para llevar a cabo las mediciones descritas en la sección anterior.

Figura 72. Diagrama de bloques, comunicación sensor computadora

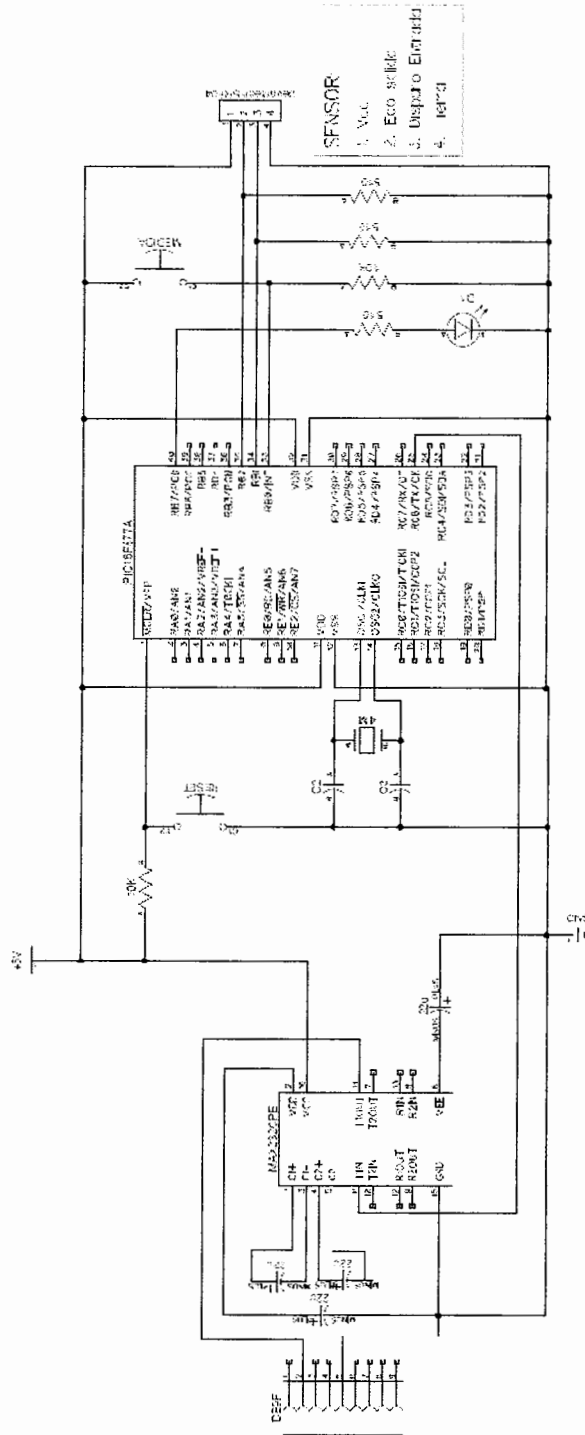


El diagrama puede utilizarse como base para comunicación entre diversos sensores ultrasónicos y la computadora. Sin embargo es posible que el control del sensor se lleve a cabo con un microcontrolador que permita la comunicación *USB* con la *PC*, como es el caso de la familia *PIC18FXXX*. De ser así, sería innecesaria la utilización del integrado *MAX232* y del convertidor *USB* serial. También es posible encontrar en el mercado, equipo de medición ultrasónica con transmisión de datos *USB*. Estas últimas dos consideraciones reducen o eliminan por completo la interfaz sensor-computadora pero implican un mayor precio de implementación del proyecto que se realice.

5.2. Diagrama del circuito eléctrico

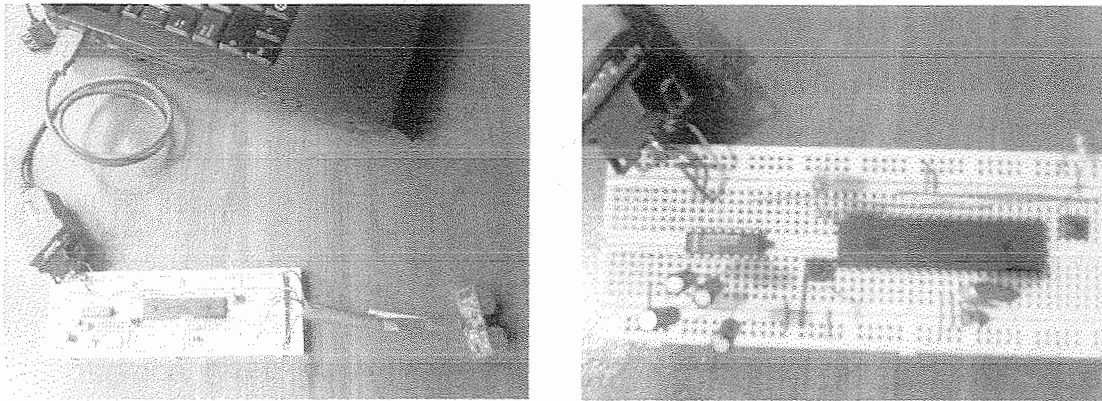
En la figura 69 se muestra el diagrama del circuito eléctrico descrito en diagrama de bloques anteriormente.

Figura 73. Diagrama del circuito eléctrico



La figura 74 muestra fotografías del circuito real utilizado como interfaz entre el *SRF04* y la computadora.

Figura 74. **Imágenes de la interfaz**



5.3. Programación del microcontrolador

El programa se realiza con el *Basic Compiler* del *Simulator Ide*, utilizando una frecuencia de reloj de 4MHz. El *PIC* posee el mismo programa para los dos montajes experimentales, teniendo modificaciones en la programación en *MATLAB* para cada uno de estos.

Para comprender mejor la estructura del programa se recomienda revisar la sección 3.1 y ver la sincronización de pulsos con la que trabaja el *SRF04*, figura 27. A continuación se describe el funcionamiento básico del programa:

- Cuando el pin 0 del puerto B recibe una señal de encendido (botón de *medida* figura 69) envía un pulso de 10 microsegundos a través del pin 1 del mismo puerto

- Dicho pulso se dirige a la entrada del sensor, el cual responde con un pulso variable dependiendo de la distancia a la que encuentre un objeto
- El pulso que envía el sensor es recibido en el pin 2 del puerto B y el programa da inicio a un ciclo que incrementará la variable *a*. Esta crece proporcionalmente a la duración que posee el pulso en el pin 2. En el momento en que el microcontrolador comienza a tomar la medida envía un pulso por el pin 7 del puerto B, que sirve para encender un led e indicar que se realiza una medida
- La variable *a* es enviada por el *PIC* a través de su pin designado para la comunicación serial
- El programa espera para realizar la siguiente medición

El fragmento de código 5.1 muestra el programa del microcontrolador, denominado Interfaz.bas

Fragmento de código 5.1. **Programa para el microcontrolador**

```
'Definición de variables:
Dim a As Word

'Definición de los puertos como entradas o salidas:
TRISB = %00000101

'Activación de transmisión serial a 9600 baudios
Hseropen 9600

inicio:

PORTB.7 = 0 'Condiciones iniciales
a = 0

If PORTB.0 = 1 Then 'Si se presiona el boton para tomar medición.
PORTB.1 = 1 'Envío de pulso de 10us para el sensor.
```

```

WaitUs 10
PORTB.1 = 0
ciclo:
    If PORTB.2 = 1 Then 'Espera respuesta del sensor.
        PORTB.7 = 1      'Enciendo led.
            While PORTB.2 = 1
                a = a + 1 'Utilizada como contador.
                WaitUs 1
            Wend
            Hserout #a, CrLf 'Se envía a hacia la PC
            a = 0
            WaitMs 500
        Else
            Goto ciclo
        Endif
    Endif
Goto inicio

```

Una vez compilado y ensamblado el programa, se crean varios archivos con el mismo nombre (nombre del archivo .bas), en este caso Interfaz.lst, Interfaz.asm e Interfaz.hex. Este último es el utilizado por la programadora de PIC's –conocida comúnmente como quemadora– para grabar el programa en el PIC.

5.4. Programación en *MATLAB*

Para comprender mejor la programación en *MATLAB* se recomienda revisar la sección 3.4, en donde se muestran las funciones principales utilizadas para leer el puerto serie de la computadora, también se puede ver la sección 3.2.4 para verificar la asignación y utilización del puerto *COM* a utilizar en el programa.

Como se menciona en la sección anterior, las medidas provenientes del sensor utilizan la misma interfaz para ambos montajes, por lo que se modifica la programación en *MATLAB* para el análisis de las mismas.

5.4.1. Medidor de distancia

El fragmento de código 5.2 muestra las funciones utilizadas para tomar mediciones, estas escritas desde la ventana de comandos de *MATLAB*.

Fragmento de código 5.2: Comandos necesarios para tomar mediciones

```
>> A=serial('COM9')
>> fopen(A)
>> a=fscanf(A,'%d')
a =
    '1234'
```

Es importante que después de correr la función *fscanf* se presione el botón de medida en circuito del microcontrolador y realizar el procedimiento cuantas veces se desee hacer la medida. Otra manera es presionando el botón del circuito el número de mediciones a realizar (cuidando no sobrepasar la capacidad del puerto serial para retener datos, que es de 128bytes) y posteriormente correr la función *fscanf* igual número de veces

Para verificar el funcionamiento de todo el montaje se realiza una serie de cinco mediciones a distancias conocidas y se observa el resultado que recibe *MATLAB*. Los resultados se muestran en la tabla X

Tabla X. **Resultados de las primeras mediciones**

Distancia [cm]	VARIABLE a				
	medida 1	medida 2	medida 3	medida 4	medida 5
5	911	910	910	910	911
15	942	941	942	941	941
25	969	72	968	968	968
35	996	995	995	100	100

Continúa Tabla X

45	1023	127	127	128	1019
55	1045	1043	1042	1042	1042
65	182	1071	182	1070	1073
75	1102	212	1100	212	1097
85	238	238	237	1124	1129
95	264	264	1152	1155	264
105	1180	293	1184	1184	1184
115	1212	1212	326	322	322
125	350	1238	355	355	1240
135	376	1268	1268	1267	1267
145	1295	401	401	1292	1293
155	1322	1321	1319	1322	1322
165	1349	460	1345	1345	1345
175	1374	1373	1373	1372	1375
185	1402	1400	1399	513	512
195	1431	1431	1431	1429	1428
205	1457	1456	1456	1456	1456
215	1384	1483	1483	595	596
225	1512	1512	1512	628	1512
235	1541	1539	649	1539	1539
245	1567	1565	1564	1564	680
255	1597	707	1595	708	1597
265	1625	1622	735	1621	1621
275	762	762	1652	1648	1648
285	1677	1680	787	787	1678
295	1707	1708	1706	1707	814
305	1736	1732	846	844	1732
315	1762	1765	1763	1761	1762
325	1793	899	1789	1788	1789
335	1819	1818	923	923	1818

Con fondo gris se resaltan valores que parecen presentar error en la medición. Si se observa con cuidado, estos valores corresponden al valor supuesto como correcto menos correspondiente a la medida en 0cm, 890. Esto se debe a una onda reflejada por el cambio de impedancia en el momento en que el sensor envía el tren de pulsos ultrasónicos.

Para solucionar el problema se realiza una estructura condicional en *MATLAB*, se crea el archivo *Medida.m* y se muestra en el fragmento de código 5.3.

Fragmento de código 5.3. Estructura condicional para arreglo en error de medición del sensor

```
%% Tomar medición:
a=fscanf(A, '%d');

%% Estructura condicional:
if med <= 900           % 900 es tomado al azar, siempre que
sea                    % menor al valor en 0cm
    a = med+890
elseif med > 900
    a=med
end
```

En el fragmento de código 5.4 se utiliza *Medida.m* desde la ventana de comandos de *MATLAB*.

Fragmento de código 5.4. Utilización del arreglo

```
>> A=serial('COM9');
>> fopen(A)
>> Medida
med =
    929
a =
    929
>> Medida
med =
    505
a =
    1413
```

Se realiza una nueva corrida de mediciones ahora con espaciamentos de 20cm y se muestran los resultados en la tabla 5.2

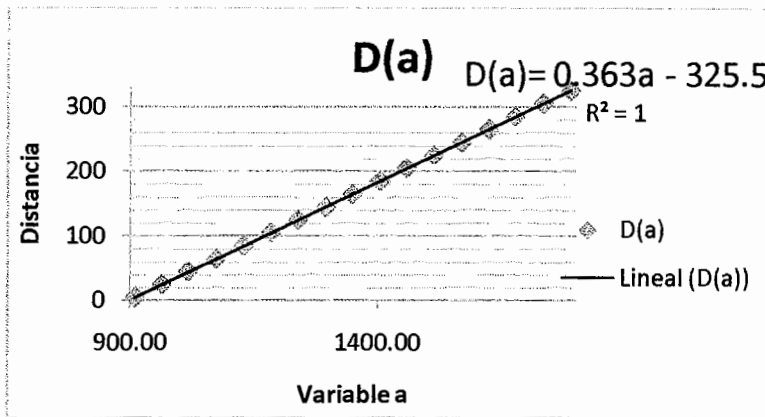
Tabla XI. **Resultados de las segundas mediciones**

Distancia [cm]	VARIABLE a					PROMEDIO	DESVIACIÓN
	med 1	med 2	med 3	med 4	med 5		
5	910	910	908	909	910	909.40	0.89
25	966	965	964	964	962	964.20	1.48
45	1017	1017	1017	1017	1017	1017.00	0.00
65	1074	1073	1072	1076	1072	1073.40	1.67
85	1127	1129	1131	1126	1131	1128.80	2.28
105	1182	1182	1183	1186	1186	1183.80	2.05
125	1241	1239	1237	1239	1236	1238.40	1.95
145	1299	1298	1296	1293	1292	1295.60	3.05
165	1348	1348	1348	1350	1347	1348.20	1.10
185	1405	1403	1403	1403	1406	1404.00	1.41
205	1457	1457	1457	1457	1458	1457.20	0.45
225	1513	1512	1513	1513	1513	1512.80	0.45
245	1570	1570	1569	1568	1568	1569.00	1.00
265	1622	1626	1622	1626	1626	1624.40	2.19
285	1678	1677	1677	1677	1678	1677.40	0.55
305	1736	1734	1733	1733	1734	1734.00	1.22
325	1790	1788	1790	1788	1788	1788.80	1.10

También se proporciona el promedio para cada distancia y la desviación estándar. Es posible realizar mediciones mayores de 3m, pero no es recomendable, ya que puede no ser efectiva la estructura condicional en Medida.m. La columna de promedio es utilizada para obtener un modelo matemático que relacione la distancia D en función del promedio de la variable a, de esta manera es posible estimar cualquier distancia. Se recomienda que las mediciones se encuentren en el rango de 3cm a 3m como lo establecen las especificaciones del sensor.

La figura 75 muestra la dispersión de datos y la tendencia de éstos. Como se espera, es lineal y proporciona la distancia en términos de la variable a $D(a)=0.363 a - 325.5$

Figura 75. **Dispersión y tendencia de las mediciones**



El modelo es agregado a Medida.m y mostrado en el fragmento de código 5.5

5.4.2. Mapa de contorno

Se realiza un mapa de contorno para una pequeña región que contiene figuras geométricas utilizando representaciones tipo B y C, esto con el fin de crear un sistema de visión artificial a través de *MATLAB*, capaz de analizar la región escaneada. El esquema de la región con sus respectivas medidas y las medidas de los objetos en la misma se muestra en la figura 72a. El esquema de la figura 72b detalla la distancia de los objetos hasta el punto más lejano de la región. La distancia elegida desde el sensor hasta el objeto más cercano es de 28cm. De la misma manera en la figura 77 se muestran imágenes reales de la región a escanear y el montaje para la toma de mediciones.

Figura 76. Esquemas de la región a escanear

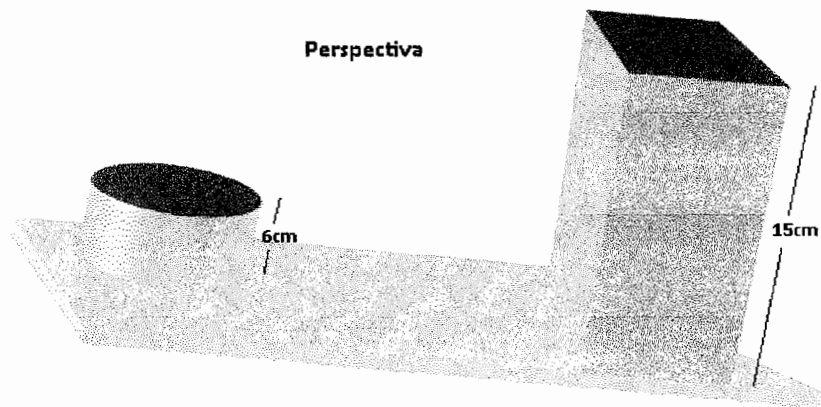
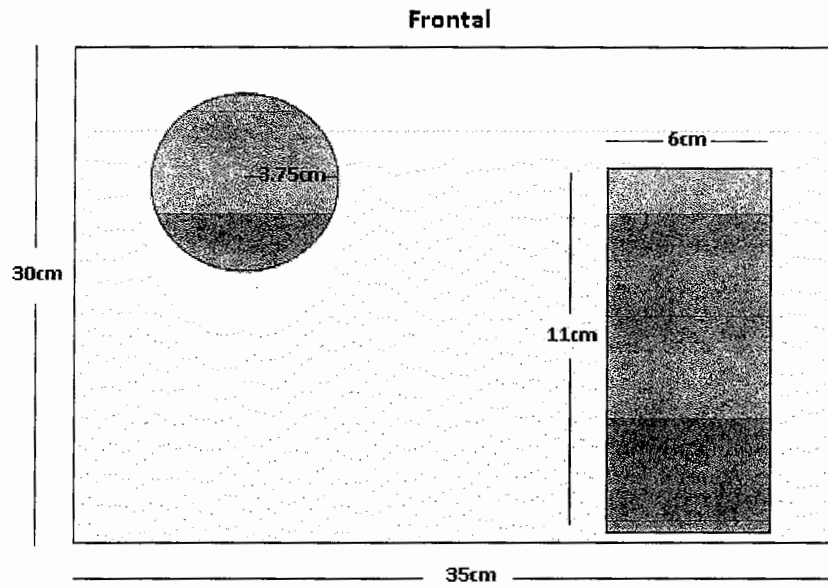
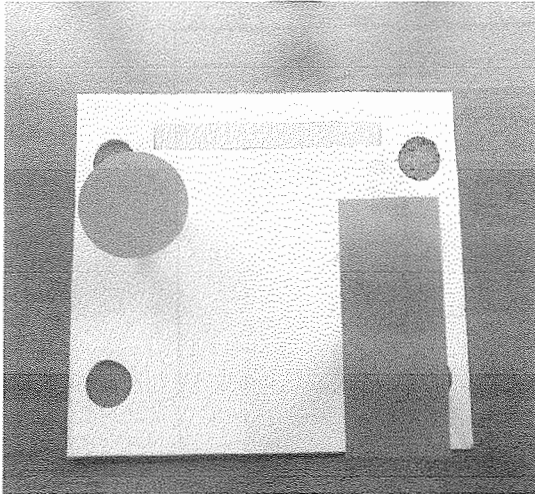
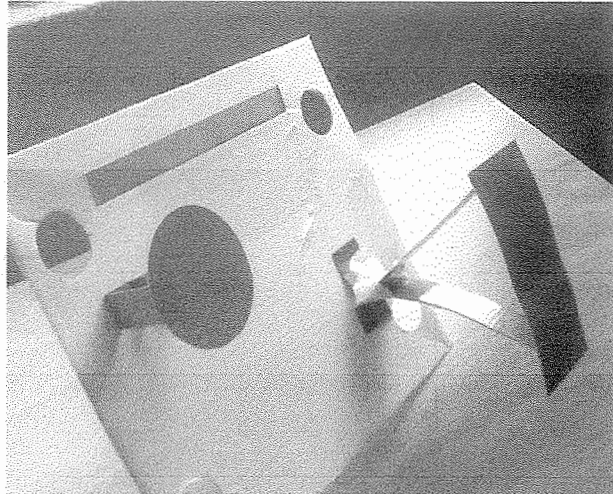


Figura 77. Imágenes reales de la región escaneada

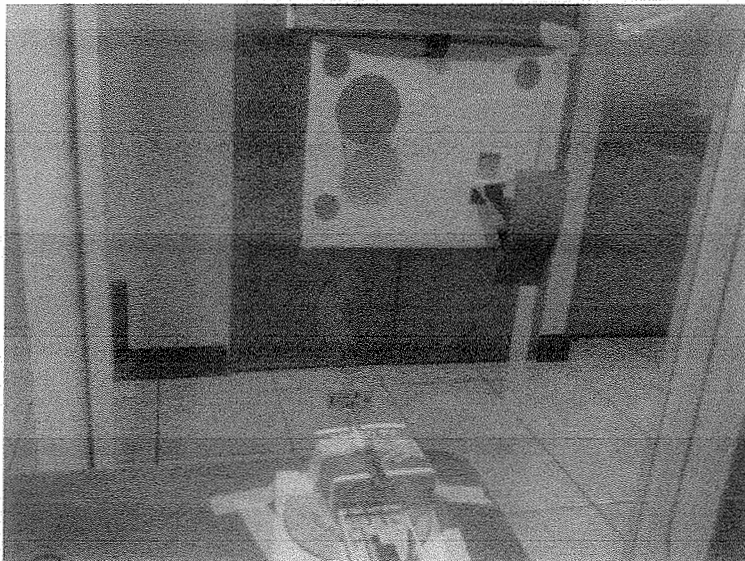
Frontal



Lateral



Montaje



Para hacer esta prueba se solicita que se ingresen las mediciones de alto y ancho que se realizaron –píxeles de alto y ancho de la imagen final correspondientemente. En base a esta información y la utilización de ciclos condicionales se ingresa cada dato de la medición en una matriz *I* de *MATLAB* que representa la imagen final. Las mediciones deben tomar en cuenta la trayectoria en el escaneo descrita figura 71b. El programa *Mapeo.m* se muestra en el fragmento de código 5.5

Fragmento de código 5.5. Almacenamiento de muestras del escaneo

```
function I=Mapeo();

clear I

%% Medidas de la región a escanear:
col=input('Ancho de la imagen: ');
fil=input('Alto de la imagen: ');

%% Indexando medidas en la matriz:
I=zeros(fil,col);

f=1;c=1;
while f*c < col*fil
    for f=1:fil
        if mod(f,2)==1
            for c=1:col
                a=fscanf(A,'%d');
                if a <= 900
                    a = a + 890;
                end
                I(f,c)=a;
                c=c+1;
            end
        elseif mod(f,2)==0
            for c=1:col
                a=fscanf(A,'%d');
                if a <= 900
                    a=a+890;
                end
                I(f,col+1-c)=a;
                c=c+1;
            end
        end
        f=f+1;
    end
end
```

```
% Modelo para estimar distancias:  
I=0.363*I-325.5;  
  
%% Mostrando imágenes:  
figure(1)  
imagesc(I)  
  
figure(2)  
imagesc(I)  
colormap gray  
  
%% Guardar imágenes:  
  
Igrises=mat2gray(I);  
imwrite(Igrises, 'MapaEntorno.jpg')
```

La imagen escalada se muestra en la figura 78, utilizando la representación tipo C.

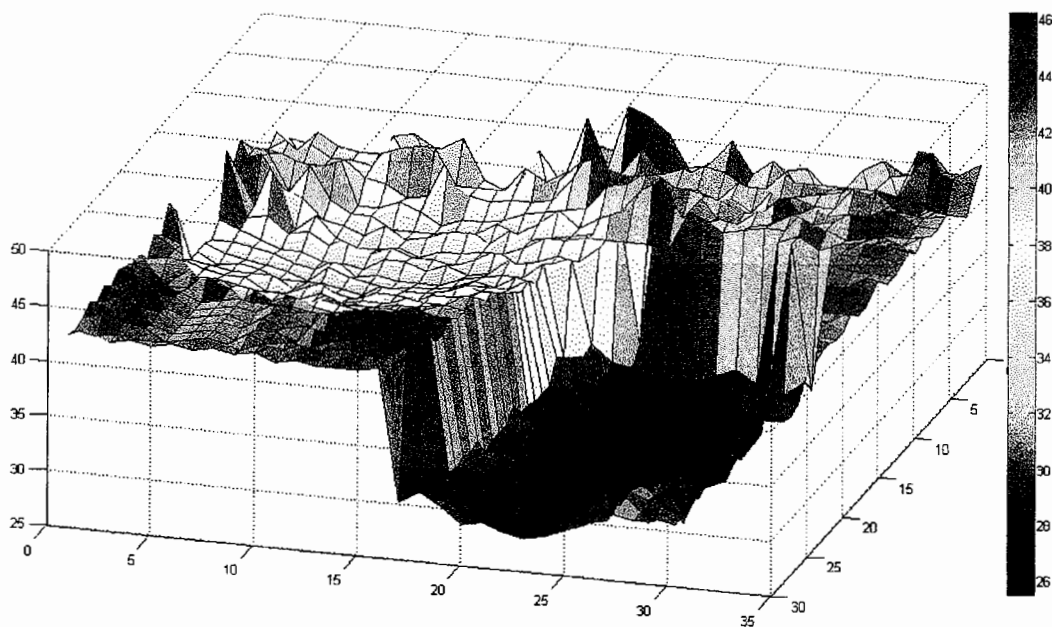
Figura 78. Reconstrucción de la región escaneada



Los tonos azules –más oscuros en escala de gris– son los objetos más cercanos y los tonos rojos –más claros en escala de gris– son los más alejados del sensor. Estos tonos se debe a la utilización del *Jet* como *colormap* cuando se utiliza la función *imagesc*, ver sección 4.1.

La superficie en tres dimensiones se muestra en la figura 79, que resulta de la combinación de la representación tipo C con la tipo B. siendo el eje vertical, la distancia del objeto en cm.

Figura 79. **Reconstrucción en 3D de la región escaneada**



Es posible observar que las mediciones varían abruptamente dependiendo de la posición del sensor. Para hacer más suaves estos cambios la imagen debe ser filtrada. Esto se muestra en el fragmento de código 5.6 *filtrar.m*.

Fragmento de código 5.6. Filtrado espacial del escaneo

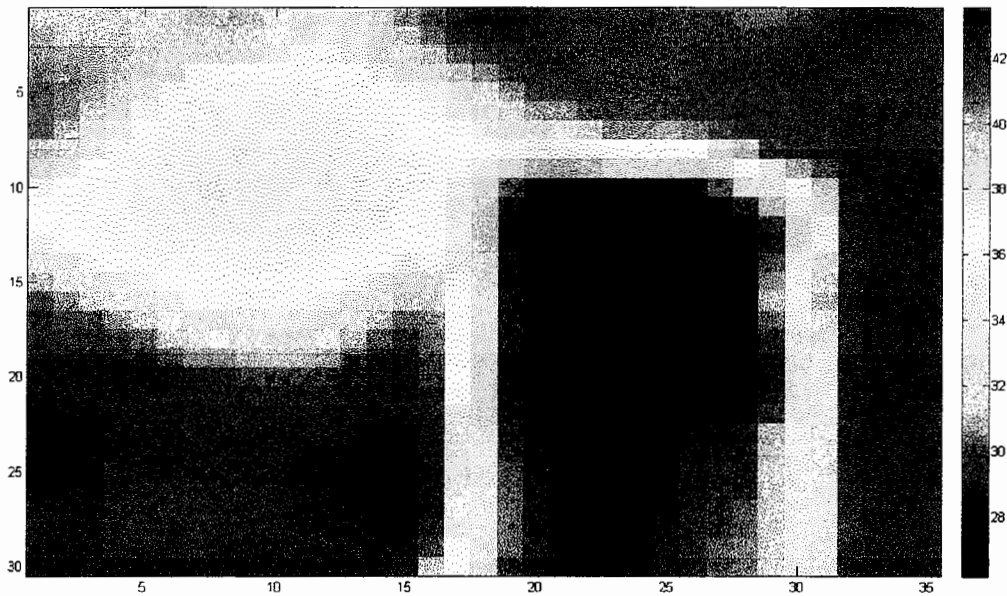
```
function J=filtrar(I0)

H=fspecial('disk',2);      % filtro: Obtención promedio entre
pixeles vecinos
J=imfilter(I0,H,'symmetric'); % aplicación del filtro

figure(1);imagesc(J)
figure(2); surf(J)
```

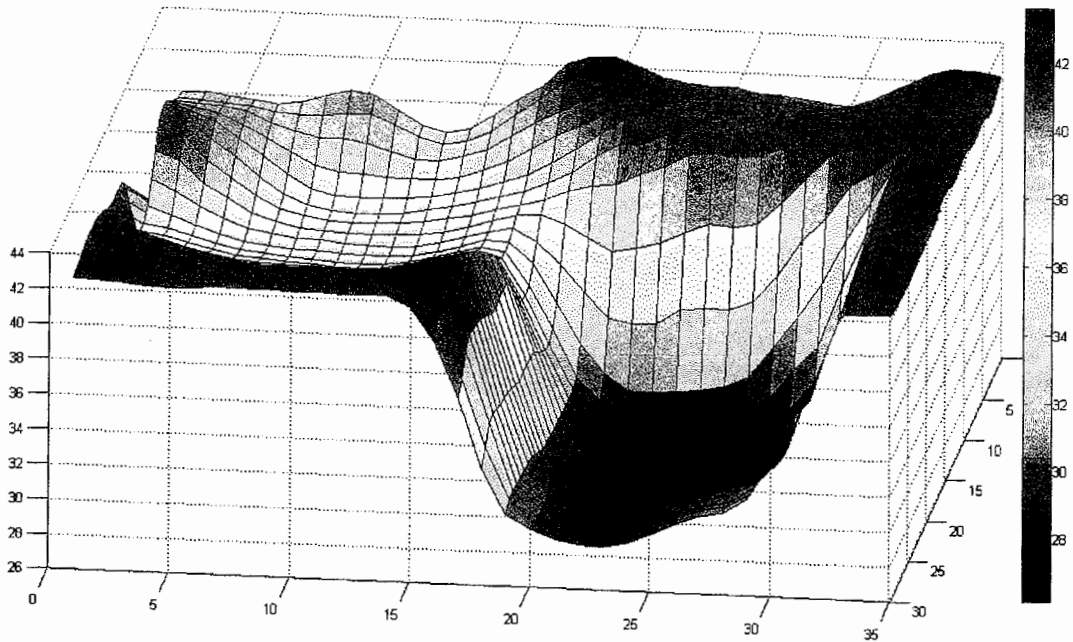
En la figura 80 se muestra la imagen escalada, en la que es posible observar más definidos los objetos encontrados en el escaneo.

Figura 80. Imagen escalada posterior al filtrado



En la figura 81 se muestra la imagen en 3D filtrada y se observa que disminuyen los cambios abruptos en las mediciones.

Figura 81. Imagen en 3D posterior al filtrado



Teniendo los datos almacenados en la matriz, es posible utilizar técnicas de procesamiento descritas en la sección 4.2 para detección de objetos a través de operaciones morfológicas en la imagen.

Se utilizan el fragmento de código 4.13 utilizado para contar objetos dentro de una imagen y proporcionar las propiedades de estos. Se definen los valores para detección de umbral y conectividad como constantes y se utiliza una matriz que erosione de forma vertical la imagen lógica. Esta matriz se elige dada la necesidad de contrarrestar el error por dispersión de haz horizontal que posee el sensor. Estas modificaciones se muestran en el fragmento de código 5.7

ProcesarMapeo.m

Fragmento de código 5.7. Procesamiento de la imagen para detectar y obtener propiedades de objetos escaneados

```
function[objetos prop]=ProcesarMapeo()
umbral=140;
conect=8;

%% lectura y conversión de colores:

I1=imread('MapaEntorno.jpg');
I=I1<umbral; %conversión a binario

%% filtrado de imagen:
H=fspecial('disk',2); % Obtención promedio entre pixeles vecinos
J=imfilter(I1,H,'symmetric');

I=imfill(I,'holes');

%% Trabajo con erosión

w=zeros(5); %creación de matriz
w(:,3)=1;

Ie=imerode(I,w); % erosión

%% etiquetando objetos

Ilab=bwlabel(Ie,conect);
objetos=max(max(Ilab)); % contando el máximo número
% de objetos etiquetados

%% Propiedades de los objetos

prop=regionprops(Ilab,'all');

% Agregar centroides y rectángulos a imagen original

figure(1);
imagesc(I1); colormap gray
hold(imgca,'on')
centro = cat(1, prop.Centroid);
plot(imgca,centro(:,1), centro(:,2), 'r*') %agregar centroides
for i=1:size(prop,1) %agregar rectángulos

rectangle('Position',prop(i).BoundingBox,'EdgeColor','b','LineWidth',2)
end
hold(imgca,'off')
```



```

%% Agregar en imagen original filtrada

figure(2);
imagesc(J); colormap gray
hold(imgca,'on')
centro = cat(1, prop.Centroid);
plot(imgca,centro(:,1), centro(:,2), 'r*') %agregar centroides
for i=1:size(prop,1) %agregar rectángulos

rectangle('Position',prop(i).BoundingBox,'EdgeColor','b','LineWidth',2)
end
hold(imgca,'off')

```

Se utiliza el fragmento anterior desde la línea de comandos de *MATLAB* como lo indica el fragmento de código 5.8.

Fragmento de código 5.8. **Utilizando *ProcesarMapeo.m***

```

>> [n p]=ProcesarMapeo()
n =

    2

p =

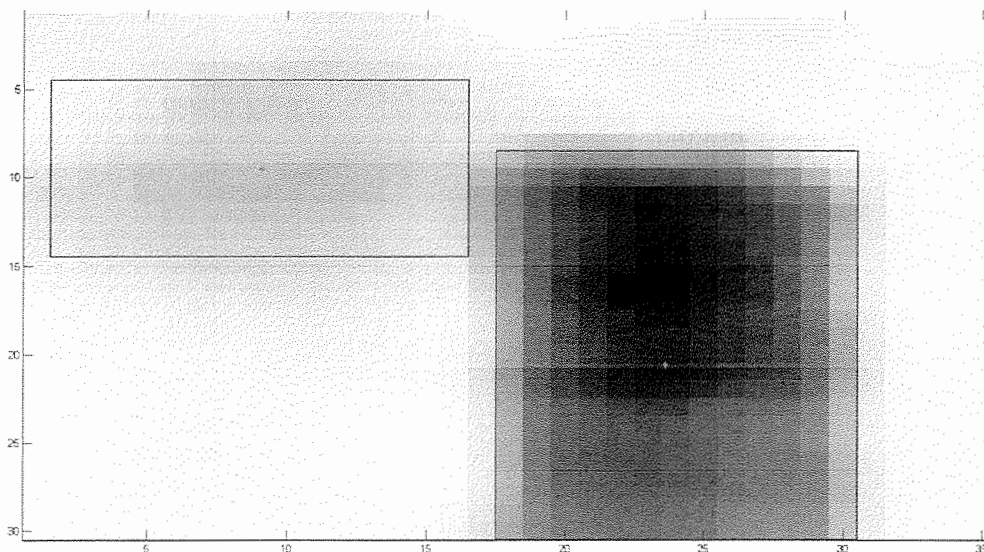
2x1 struct array with fields:
    Area
    Centroid
    BoundingBox
    SubarrayIdx
    MajorAxisLength
    MinorAxisLength
    Eccentricity
    Orientation
    ConvexHull
    ConvexImage

```

- ConvexArea
- Image
- FilledImage
- FilledArea
- EulerNumber
- Extrema
- EquivDiameter
- Solidity
- Extent
- PixelIdxList
- PixelList
- Perimeter

Es posible observar que la función proporciona el número n de objetos encontrados y las propiedades p de los mismos. Para este caso se encontraron dos objetos. En la figura 82 se muestra la imagen filtrada con los objetos encontrados en ella encerrados en rectángulos e indicando los centroides de estos.

Figura 82. **Objetos encontrados en la imagen escaneada**



Se calculan las áreas de los objetos encontrados con el fragmento de código 5.9.

Fragmento de código 5.9. Cálculo de áreas de los objetos

```
>> p.Area
ans =
    80

ans =
   247
```

Es importante tener en cuenta que las áreas calculadas se encuentran en pixeles cuadrados y para este caso cada pixel posee una longitud de 1cm (espaciamiento entre mediciones) y que el área encontrada es mucho mayor a las áreas reales, esto por los errores en la medición debido a la dispersión del haz ultrasónico en el sensor.

El área real del objeto correspondiente al círculo con radio de 3.75cm es de 44.18cm^2 aproximadamente y el área real del rectángulo con base de 6cm y altura de 11cm es de 66cm^2 . Se calcula un modelo lineal (similar al procedimiento para calcular el modelo de la sección anterior) que relaciona el área real aproximada A_r con el área medida en *MATLAB* A_m y un *factor* en términos de la distancia D a la que se encuentra el objeto.

Se utilizan los datos conocidos para calcular una regresión lineal entre la distancia D y el *factor*, teniendo como resultado; $\text{factor}=0.049D-1.157$, por último el área real debe ser el factor multiplicado por el área medida por *MATLAB*. Este modelo se utiliza en el fragmento de código 5.10 –Análisis.m– que proporciona información de los objetos encontrados en la región escaneada.

Fragmento de código 5.10. Análisis de los objetos detectados

```

function Tabla=Análisis(I0,n,p)

%% creación de matriz de ceros
Tabla=zeros(n,5);

%% llenado de matriz
for f=1:n
    Tabla(f,1)=f;           %n filas para n objetos encontrados
                           %enumeración de objetos

    C=p(f).Centroid;      %centroide
    Tabla(f,2)=C(1);
    Tabla(f,3)=C(2);

    C=round(C);          %encontrado distancias en la matriz
                           %de distancias

    d=I0(C(1),C(2));
    Tabla(f,4)=d;

    factor=0.049*d-1.157; %factor en términos de la distancia
    area=p(f).Area*factor; %aproximación al área real
    Tabla(f,5)=area;

    f=f+1;
end

%% Mostrando Tabla:
disp(' ')
disp('-----')
disp('          TABLA DE ANÁLISIS:          ')
disp(' ')
disp('          n          Centroide          Distancia          Área')
disp('          x          y          [cm]          Aprox. ')
disp(' ')
disp(' ')
disp(Tabla)
disp('-----')
disp(' ')

%% Guardar en excel:
xlswrite('medidas.xlsx',Tabla,'Análisis','B2');

```

Para utilizar la función anterior debe proporcionarse los valores de n y p encontrados en el fragmento de código 5.9 y la matriz de inicial de distancias $I0$. En el fragmento de código 5.11 se muestra la utilización desde la línea de comandos y el resultado obtenido. Además de guardar el resultado en un archivo de Excel.

Fragmento de código 5.11. **Obtener información de los objetos detectados**

```
>> Tabla=Analisis(I0,n,p);
```

TABLA DE ANÁLIS:

n	Centroide		Distancia [cm]	Área Aprox.
	x	y		
1.0000	9.0875	9.5625	35.6850	47.3252
2.0000	23.6275	20.6113	29.1510	67.0356

CONCLUSIONES

1. El ultrasonido pertenece a las ondas acústicas inaudibles para los humanos: sobre los 20KHz y se puede estudiar como cualquier onda sonora.
2. Las múltiples aplicaciones del ultrasonido se basan en las técnicas de medición denominadas de transmisión y pulso eco. Estas se fundamentan en la reflexión y transmisión de ondas debido a cambios en la impedancia acústica del medio de propagación.
3. Debido al avance en sistemas de generación, medición y representación electrónicos, ha sido posible la construcción de instrumentos para la aplicación del ultrasonido en áreas de diagnóstico y tratamiento médico, ciencias de los materiales y robótica, entre muchas otras.
4. Se han propuesto tres modos de representación –A, B y C– y combinaciones de los mismos para visualizar y analizar mediciones ultrasónicas.
5. Utilizar un microcontrolador resulta un método práctico y eficiente para realizar una interfaz entre el sensor ultrasónico y un computador, debido a la facilidad de programarlo y la posibilidad, en algunos modelos, para realizar comunicación serial o *USB*.

6. Para cada diseño o investigación a realizar, el transductor ultrasónico se selecciona atendiendo características de dispersión de haz, acople de impedancia, frecuencia de trabajo, capacidad de inmersión y tipo de incidencia.
7. El sensor *Devantech SRF04* presenta problemas para el acople de impedancia, produciendo errores en los cálculos de mediciones de distancias.
8. El *Devantech SRF04* posee alta dispersión de haz, lo cual produce errores de traslape durante las pruebas realizadas para la construcción de imágenes.
9. A través de *MATLAB* es posible indexar en matrices los datos enviados por el sensor hacia la computadora y posteriormente con ayuda del paquete de procesamiento digital de imágenes, estructurar algoritmos para visualizar, procesar y analizar mediciones ultrasónicas.
10. Aumentar la complejidad en la estructura del algoritmo en *MATLAB*, no disminuye las fuentes de error debidas al sensor, pero si ayuda a que éstas no afecten en gran medida los resultados de procesamiento y análisis de las mediciones.
11. Los algoritmos estructurados permiten ser modificados y extendidos para adaptarlos a diversas áreas, principalmente en aplicaciones que utilicen la técnica de pulso—eco.

12. *MATLAB* permite trabajar con la transformada de Radon y con la técnica de retroproyección filtrada para el cálculo de la transformada inversa de Radon. Esto permite aplicar métodos no invasivos en la reconstrucción de imágenes internas del cuerpo humano a través de senogramas obtenidos con mediciones ultrasónicas.

RECOMENDACIONES

1. Motivar la investigación y desarrollar proyectos dentro de la Facultad de Ingeniería, principalmente en la carrera de electrónica, ya que consta con el pensum de estudio suficiente para realizar diversos instrumentos ultrasónicos, así como el procesamiento y análisis de las mediciones logradas con los mismos.
2. Implementar un proyecto con un sistema de automatización, en el curso de robótica, que se adapte a las mediciones realizadas en la creación del mapa de entorno para visión artificial. Se pueden analizar e incluir sistemas de coordenadas cilíndricas y esféricas para las mediciones y no únicamente un escaneo rectangular como el presentado en el capítulo 5.
3. Establecer un curso, dentro del pensum de ingeniería electrónica, de “Señales y Sistemas en Tiempo Continuo y Discreto”. Este curso deberá ser previo a los cursos de comunicaciones. El curso de “Comunicaciones 4” debiera enfocarse específicamente al procesamiento digital de señales y procesamiento de imágenes, abarcando temas de filtraje en el dominio de la frecuencia, filtraje espacial, transformadas de Fourier, Hilbert, Radon y *Wavelet* y sus aplicaciones, así como el trabajo de cada tema en *MATLAB* y/o *Simulink*.

4. Crear manuales de laboratorio como trabajos de graduación para los diversos cursos de la carrera de electrónica, principalmente en cursos de comunicaciones, éstos asesorados por el catedrático de cada curso con el fin de motivar a los estudiantes a utilizar programas como MATLAB y/o Simulink.

5. Motivar a los estudiantes del curso Proyectos de Computación Aplicados a Ingeniería Electrónica a estructurar algoritmos eficientes en MATLAB, respetando el contenido de dicho curso y que puedan ser aprovechados en diversas áreas en la Facultad de Ingeniería.

BIBLIOGRAFÍA

1. CUEVAS JIMENEZ, Erik, ZALDIVAR NAVARRO, Daniel. "*Visión por computador usando MatLAB y el Toolbox de procesamiento digital de imágenes*". Informe inédito. Maestría en Ciencias en Ingeniería en Electrónica y Computación de la Universidad de Guadalajara, México. 2006. 34 p.
2. ECHEVERRÍA, Ricardo. "*Ultrasonido*". Informe inédito. Laboratorio de Ensayos No Destructivos, Facultad de Ingeniería, Universidad Nacional del Comahue, Argentina. 2002. 89 p.
3. HØILUND, Carsten. "*The Radon Transform*". Informe inédito. Universidad de Aalborg, Dinamarca. 2007. 20 p.
4. LARS, Ericsson. "*ULTRASONIC SIGNAL PROCESSING TOOLBOX User Manual v1.0*". Informe inédito. Laboratorio de Señales y Sistemas, Universidad de UPPSALA, Suecia. 2001. 12 p.
5. LORENTE GASSÓ, Jose. "*Transductores de ultrasonido*". Universidad Politécnica de Valencia, España. s.a. 15 p.
6. MICROCHIP TECHNOLOGY Incorporated. *Pic16f877 data sheet* [en línea]. Estados Unidos. 2003 [Consulta: marzo 2010]. Disponible en Web:
<<http://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf>>

7. NDT Education Resource Center. LARSON, Brian. *Introduction to Ultrasonic Testing* [en línea]. Universidad Estatal de Iowa: 2001. Revisión: Enero 2009. [Consulta: Noviembre 2009]. Disponible en Web:
<www.ndt-ed.org/EducationResources/CommunityCollege/Ultrasonics/cc_ut_index.htm>
8. OSHON SOFTWARE, Software Solutions. *PIC Simulator IDE* [en línea]. 2001. Revisión: 2010. [Consulta: Marzo 2010]. Disponible en Web:
<www.oshonsoft.com/pic.html>
9. PRATAP, Rudra. *Getting Started with MATLAB, A Quick Introduction for Scientists And Engineers*. Estados Unidos: Oxford University Pres. 2002. 245 p.
10. RANA, Mank, *et al.* "Automated tracking of muscle fascicle orientation in B-mode ultrasound images". *Journal of Biomechanics*. Canada. 2009, Vol. 42. p 2068-2073
11. RESTREPO G., Andrés D. *et al.* "Control y adquisición serial de señales ultrasónicas con MATLAB". *Ingeniería e Investigación*. Bogotá, Colombia. 2006, Vol. 26. p. 103-109
12. SANCHO VENDELL, Javier, LLOPIS REYNA, Ana y LLINARES GALIANA, Jaime. *Acústica arquitectónica y urbanística*. Universidad Politécnica de Valencia: Editorial Limusa, 2008. 368 p.

13. THE MATHWORKS. *Image Processing Toolbox™ 7 User's Guide* [en línea]. Estados Unidos. 1997. Revisión: marzo 2010 [Consulta: junio 2010]. Disponible en Web: <<http://www.mathworks.com/help/toolbox/images/bqj5b2j.html>>
14. WIKIPEDIA, The Free Encyclopedia. *Ultrasound*. 2003. Revisión: Octubre 2009. [Consulta: Noviembre 2009]. Disponible en Web: <en.wikipedia.org/wiki/Ultrasound>