



Manual de Usuario

Interfaz Gráfica de Usuario (GUI)

Version 1.0.0

Ing. José Abisaid Martínez Mata
*Departamento de Robotica y Manufactura Avanzada
Laboratorio de Evaluaciones No Destructivas y
Sensado Inteligente (LabENDSI)
CINVESTAV, Saltillo, México
abisaid.martinez@cinvestav.mx*

January 18, 2026

Contents

1	Introducción	3
2	Requisitos previos	3
2.1	Requisitos de Hardware	5
2.2	Requisitos de Software	7
3	Instalación y Configuración	9
3.1	Instalación de MATLAB	9
3.2	Instalación de Toolboxes	10
3.3	Instalación de Drivers y Librerías	10
3.4	Configuración de controladores Velmex	10
3.5	Verificación del entorno	10
4	Preparación del sistema	11
4.1	Preparación del sistema experimental	11
4.2	Preparación del osciloscopio	13
5	Descripción de la interfaz grafica	15
5.1	Componentes de la Interfaz Gráfica de Usuario (GUI)	15
6	Uso basico	22
6.1	Primeros pasos	22
6.1.1	Experimentos para reconstrucción mediante Tomografía Ultrasonica Computarizada (USCT)	23
6.1.2	Adquisición individual de señales	25
7	Funciones avanzadas	26
8	Ilustraciones y capturas	26
8.1	Secciones clave de la GUI	27
8.2	Ejemplos de resultados	27
9	Ejemplos para su uso	28
10	Glosario de terminos	29
11	Contacto y soporte	29
11.1	Tipos de soporte disponibles	29
11.2	Canales de comunicación	30
11.3	Recomendaciones antes de solicitar soporte	30
11.4	Política de soporte	30

LabENDSI	José Abisaíd Martínez Mata	Cinvestav
12 Errores comunes		31
12.1 No se muestra GUI		31
12.2 No hay movimiento en el robot de inmersión		31
12.3 No hay interacción con el osciloscopio		32
12.4 No es posible visualizar las proyecciones		33
12.5 No se puede visualizar la reconstrucción experimental		34
12.6 No se pueden guardar las imágenes de reconstrucciones		34
12.7 La reconstrucción no es adecuada		35

List of Code Listings

1	Verificación de detección de dispositivos visa	9
2	Comando de conversión de GUIDE a App Designer.	9

1 Introducción

El presente documento tiene como finalidad describir los requerimientos de hardware y software necesarios para la correcta instalación y operación del entorno desarrollado, así como guiar al usuario en su uso básico, configuración y posibles soluciones ante incidencias comunes.

Adicionalmente, se incluyen recomendaciones y lineamientos que permitirán realizar futuras modificaciones o actualizaciones del sistema de forma segura y eficiente.

La creación de este entorno gráfico surge de la necesidad de optimizar la interacción hombre-máquina (HMI, por sus siglas en inglés) dentro de los procesos de adquisición de señales ultrasónicas y movimiento experimental, ofreciendo una herramienta que integre control automático y manual sin la necesidad de ejecutar código de manera directa.

De esta forma, el sistema proporciona una plataforma ergonómica, intuitiva y accesible, centrada en la operatividad práctica del proceso, permitiendo al usuario enfocarse en la obtención y análisis de datos de forma sencilla, segura y reproducible.

La interfaz gráfica fue desarrollada en MATLAB GUIDE, integrando módulos de procesamiento de señales, reconstrucción tomográfica por ultrasonido (USCT) e implementación de modelos de inteligencia artificial basados en arquitecturas ResNet50 y Faster R-CNN.

Estas herramientas permiten la adquisición y procesamiento de datos experimentales, además de la evaluación automática de resultados mediante técnicas de aprendizaje profundo (deep learning), ampliando el alcance del sistema hacia tareas de clasificación y detección de anomalías en las imágenes reconstruidas.

En conjunto, este manual busca servir como una guía integral que facilite el uso del sistema a nuevos usuarios, estandarice los procedimientos de instalación y operación, y establezca una base técnica sólida para la expansión futura del entorno hacia nuevas funciones o configuraciones experimentales.

2 Requisitos previos

Antes de proceder con la instalación y ejecución de la interfaz gráfica, es indispensable verificar que el sistema cuente con los recursos de hardware, software y configuración necesarios para garantizar un funcionamiento correcto y estable del entorno de trabajo.

La interfaz fue desarrollada en MATLAB® GUIDE (Graphical User Interface Development Environment, por sus siglas en inglés), empleando un conjunto de módulos que integran procesamiento de señales ultrasónicas, reconstruc-

ción tomográfica y análisis basado en redes neuronales convolucionales (CNNs).

Debido a la naturaleza de estas tareas (que implican el manejo intensivo de datos, cálculos matriciales y entrenamiento de modelos de inteligencia artificial), se requiere un entorno computacional con características suficientes para sostener las operaciones en un tiempo razonable. Asimismo, el sistema está diseñado para comunicarse con equipos de adquisición de datos y control, tales como controladores de movimiento de la marca Velmex® y osciloscopios de la marca Tektronix®, mediante protocolos VISA y SCPI. Por ello, es fundamental contar con los controladores oficiales TekVisa y los drivers USB/RS232 de Velmex correctamente instalados antes de la ejecución.

En cuanto al entorno de desarrollo, se recomienda utilizar MATLAB R2024a, versión en la cual fue diseñada y validada la interfaz. Esta versión conserva la compatibilidad total con GUIDE y con las herramientas necesarias para el procesamiento de señales, la adquisición de datos y la implementación de algoritmos de aprendizaje profundo.

Aunque versiones anteriores (desde R2016a) o la más reciente (R2024b) mantienen soporte parcial, a partir de R2025a, GUIDE fue descontinuado, por lo que no es posible ejecutar directamente los archivos .fig asociados a la interfaz sin realizar una migración a App Designer.

Asimismo, para la ejecución de los modelos de inteligencia artificial basados en ResNet50 y Faster R-CNN, se recomienda disponer de una GPU compatible con CUDA, la cual acelera significativamente las etapas de inferencia y entrenamiento. En equipos sin GPU, MATLAB ejecutará los modelos en modo CPU, lo que puede aumentar considerablemente el tiempo de cómputo.

Finalmente, se sugiere contar con las siguientes toolboxes instaladas:

- **Signal Processing Toolbox**, para el análisis de señales ultrasónicas.
- **Image Processing Toolbox**, para el manejo de imágenes reconstruidas.
- **Instrument Control Toolbox**, para la comunicación con dispositivos externos.
- **Deep Learning Toolbox y Computer Vision Toolbox**, para la implementación de modelos basados en CNNs.
- **Parallel Computing Toolbox**, para la comunicación e implementación de modelos entrenados mediante la GPU en Matlab.

Estas herramientas permiten ejecutar sin restricciones las diferentes rutinas de procesamiento, reconstrucción y clasificación integradas en la GUI.

2.1 Requisitos de Hardware

La tabla 1 presenta las especificaciones mínimas y recomendadas de hardware, que deben verificarse antes de la instalación y uso del sistema.

Table 1: Requisitos de Hardware mínimos

Componente	Especificación mínima	Especificación recomendada	Descripción
Procesador (CPU)	Intel Core i5 (4 núcleos) o equivalente	Intel Core i7 / AMD Ryzen 7 o superior	Requerido para procesamiento en tiempo real y control de dispositivos externos
Memoria RAM	8 GB	16 GB o más	Mejora la estabilidad del sistema durante la adquisición y análisis de señales
Almacenamiento	10 GB libres	20 GB o más (SSD preferido)	Espacio suficiente para MATLAB toolboxes y almacenamiento de datos experimentales
Controladores de movimiento	Velmex (modelo XSlide / BiSlide, según el montaje experimental)	—	Controladores motorizados empleados para el desplazamiento preciso del sistema
GPU (opcional)	NVIDIA con soporte CUDA (4 GB VRAM)	NVIDIA RTX 3060 o superior (8 GB VRAM)	Recomendado para acelerar el entrenamiento y validación de modelos CNN
Osciloscopio	Tektronix (modelos compatibles con TekVisa o SCPI)	Tektronix TDS/DPO/MSO	Para adquisición y monitoreo de señales, integrados mediante comunicación VISA
Monitor	Resolución mínima 1366×768	1920×1080 (Full HD)	Permite una visualización óptima de la interfaz gráfica y las gráficas generadas

2.2 Requisitos de Software

La tabla 2 presenta las especificaciones mínimas y recomendadas de software, que deben verificarse antes de la instalación y uso del sistema.

Table 2: Requisitos de Software mínimos

Elemento	Versión/Detalle	Descripción
Sistema Operativo	Windows 10/11 (64 bits)	Recomendado por su compatibilidad con MATLAB, TekVisa y controladores Velmex
Entorno MATLAB	MATLAB R2024a (versión recomendada)	Versión utilizada en el desarrollo de la GUI. Mantiene compatibilidad total con GUIDE
Compatibilidad extendida	R2016a – R2024b	Versiones que aún soportan GUIDE y permiten ejecución directa de archivos .fig y .m
Versiones no compatibles	R2025a o posteriores	GUIDE fue eliminado. En estas versiones, el proyecto debe migrarse con <i>guideToAppDesigner</i>
Toolbox requeridos	<ul style="list-style-type: none"> • MATLAB Base • Signal Processing Toolbox • Instrument Control Toolbox • Image Processing Toolbox • Deep Learning Toolbox • Computer Vision Toolbox • Parallel Computing Toolbox 	Necesarios para comunicación serial, procesamiento visual, manejo de dispositivos y uso de GPU
Dependencias externas	<ul style="list-style-type: none"> • TekVisa (para comunicación con osciloscopios Tektronix) • Controladores Velmex (drivers USB/RS232 según modelo) 	Deben instalarse previamente para permitir la comunicación y ejecución eficiente de las tareas de reconstrucción y entrenamiento

Como observaciones adicionales a considerarse, se recomienda lo siguiente

- Ejecutar Matlab con permisos de administrador al usar controladores externos o puertos VISA
- En equipos sin GPU, los modelos basados en CNN se ejecutaran en modo CPU, con tiempos de computo mas altos
- Se recomienda verificar la detección de dispositivos mediante los comandos:

```
instrhwinfo('visa')
gpuDevice
```

Listing 1: Verificación de detección de dispositivos visa

- En caso de usar versiones posteriores a R2025a, se recomienda migrar la GUI a App Designer con

```
guideToAppDesigner('BP_interface.fig')
```

Listing 2: Comando de conversión de GUIDE a App Designer.

3 Instalación y Configuración

Esta sección proporciona los recursos necesarios para realizar la instalación del entorno de trabajo y los controladores asociados al sistema. Se incluyen enlaces a tutoriales y guías oficiales que explican el procedimiento de manera clara y actualizada.

3.1 Instalación de MATLAB

Para realizar la instalación de MATLAB, se recomienda consultar los siguientes recursos que detallan el proceso paso a paso. Estos tutoriales han sido verificados por el autor de este manual.

- Instalación de MATLAB (versión general)¹
- Instalación de MATLAB 2024a²
- Guía paso a paso de instalación de MATLAB³

¹<https://youtu.be/n5T94c8e4w0?si=MvgMZ1ve5f00l9U->

²<https://youtu.be/7G0s9whng08?si=GzI3DAguGlqYJF1c>

³<https://share.google/STW1LGY1zXDI91A5U>

3.2 Instalación de Toolboxes

Para añadir los *Toolboxes* necesarios descritos en la Sección 2, puede seguir el procedimiento explicado en el siguiente recurso. El proceso es idéntico al mostrado, variando únicamente los nombres de los *Toolboxes* a instalar.

- ¿Cómo añadir un Toolbox en MATLAB?⁴

3.3 Instalación de Drivers y Librerías

Para habilitar la comunicación con dispositivos de National Instruments y Tektronix, es necesario instalar los controladores correspondientes. Descargue los instaladores oficiales y siga las instrucciones que aparecen durante el proceso.

- Instalación del Driver Tektronix (TekVISA)⁵
- Instalación de Drivers NI-VISA⁶

3.4 Configuración de controladores Velmex

Para establecer la comunicación con los controladores de la marca Velmex y garantizar su correcta interacción con el entorno de MATLAB, se recomienda revisar el conjunto de comandos disponibles. Estos comandos permiten controlar los ejes de movimiento, realizar calibraciones y ejecutar desplazamientos de manera precisa.

Para una referencia detallada de los comandos compatibles, consulte el siguiente documento:

- Comandos Velmex⁷

3.5 Verificación del entorno

Una vez completadas las instalaciones anteriores, se recomienda ejecutar el script `verificar_instalacion.m` incluido en este proyecto (Vease repositorio de GitHub) para confirmar que:

- MATLAB reconoce los *Toolboxes* requeridos.

⁴<https://www.youtube.com/watch?v=c1Js5FsDp5s>

⁵<https://www.tek.com/en/support/software/driver/tekvisa-connectivity-software-v404>

⁶<https://www.ni.com/es/support/downloads/drivers/download.ni-visa.html#575764>

⁷<https://share.google/VfNmh16kag1d955Rv>

- Los dispositivos externos son detectados correctamente.
- La comunicación con el sistema robótico es funcional.

4 Preparación del sistema

Antes de ejecutar la interfaz gráfica, es necesario preparar los componentes físicos y verificar las conexiones entre los dispositivos que conforman el sistema de adquisición y control. Esta etapa garantiza que, al iniciar la GUI, los controladores y el osciloscopio se encuentren correctamente identificados y operativos.

4.1 Preparación del sistema experimental

En la Figura 1 se muestra la estructura física del robot, junto con sus 5 grados de libertad (5-DOF). Este robot es capaz de realizar movimientos de traslación en los ejes X, Y y Z en ambas direcciones, además de contar con un movimiento rotacional alrededor del eje Z (θ_Z). Adicionalmente, incluye un movimiento angular denominado θ (como se aprecia en la Figura 1), que proporciona mayor versatilidad para orientarse y adaptarse a las necesidades del análisis ultrasónico. De esta manera, los motores M1, M3 y M4 son responsables de controlar la posición del sensor en el espacio, permitiendo los movimientos de traslación en los ejes X, Y y Z. Por otro lado, el motor M2 se encarga de la rotación en el eje Z (θ_Z), ajustando la trayectoria lineal que se seguirá durante el análisis. Finalmente, el motor M5 regula el ángulo del transductor (θ), orientándolo de forma normal a la superficie del material. Este sistema cuenta con cinco motores responsables de efectuar los movimientos sobre los ejes antes mencionados, los cuales están conectados a controladores encargados de trasladar o rotar según corresponda. La Tabla 3 muestra los controladores encargados de mover los ejes del sistema de inmersión anteriormente expuestos. Para poder realizar la comunicación

Table 3: Relación controladores-ejes.

Controlador 1	M1-Motor eje Z	M2-Motor Rotación Z
Controlador 2	M3-Motor eje X	M4-Motor eje Y
Controlador 3	M5-Motor Pendulo	-

con los controladores, sigase la siguiente serie de pasos a ejecutar

- Conecte el controlador *Velmex* al puerto serial o USB correspondiente del equipo de cómputo, como recomendacion, consiga un multipuerto usb, debido a la cantidad de controladores.

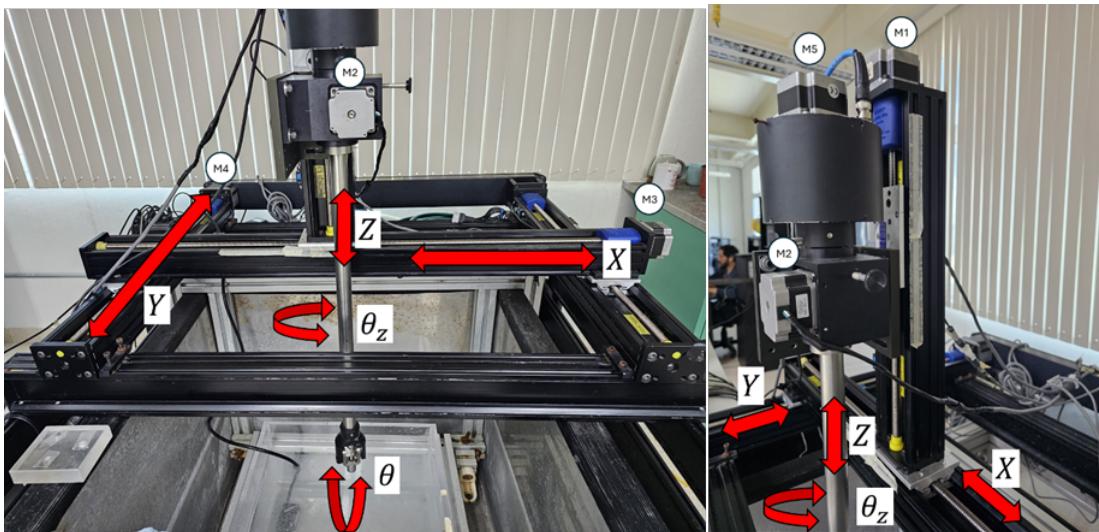


Figure 1: Estructura del robot de inmersión.

- Verifique en el *Administrador de dispositivos* (Windows) el número de puerto asignado (por ejemplo, COM3, como se aprecia en la Figura 2).

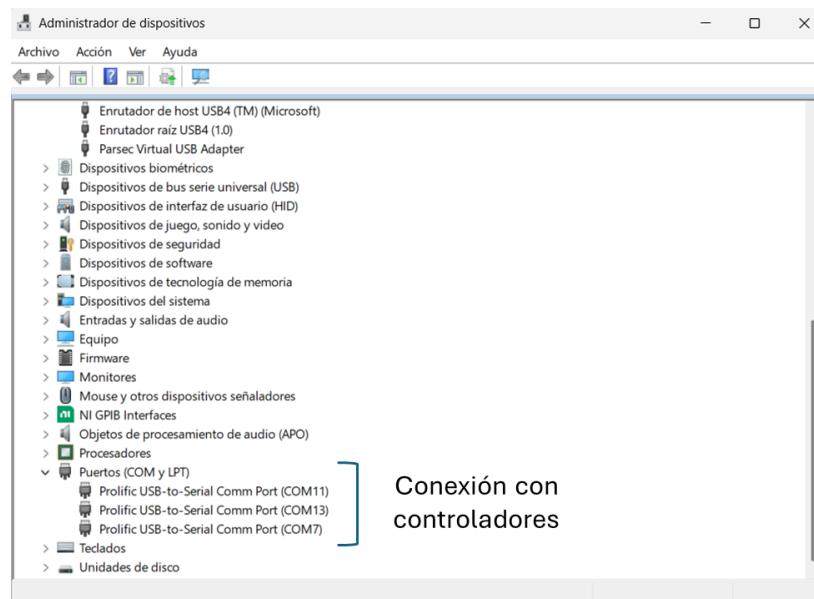


Figure 2: Captura de pantalla que muestra la comunicación de los puertos COM con el dispositivo, listo para su manipulación

- Asegúrese de que el controlador esté encendido, libre de errores de comunicación y correctamente configurado para recibir comandos desde MATLAB.

La Figura 3 muestra de manera física los controladores de la marca Velmex, estructurados para la conexión con el dispositivo



Figure 3: a) Controladores físicos que realizan la comunicación con el sistema de inmersión y b) Confirmación de la comunicación con el robot de inmersión, cuando se acciona un movimiento de la GUI, el controlador enciende el LED superior, indicando que hay comunicación con el sistema.

4.2 Preparación del osciloscopio

Para preparar la interacción del osciloscopio, puede seguir esta secuencia de pasos.

Conecte el osciloscopio mediante cable USB o a través de una interfaz VISA (LAN o GPIB, según el modelo disponible). Para confirmar la interacción entre el osciloscopio y la computadora, acceda a la aplicación *Instrument Control* desde el entorno de *MATLAB*.

Una vez abierta la aplicación, seleccione el menú correspondiente a **VISA**. En la parte inferior de la ventana, presione el botón **Scan** para iniciar la búsqueda de dispositivos conectados. Al finalizar el escaneo, se mostrará el identificador (ID) del osciloscopio detectado, tanto por VISA como por Tektronix, si se instalaron de manera adecuada ambas librerías, confirmando una comunicación exitosa entre el equipo y el entorno de trabajo.

La Figura 4 muestra los pasos principales para acceder al sistema y verificar la comunicación con el osciloscopio.

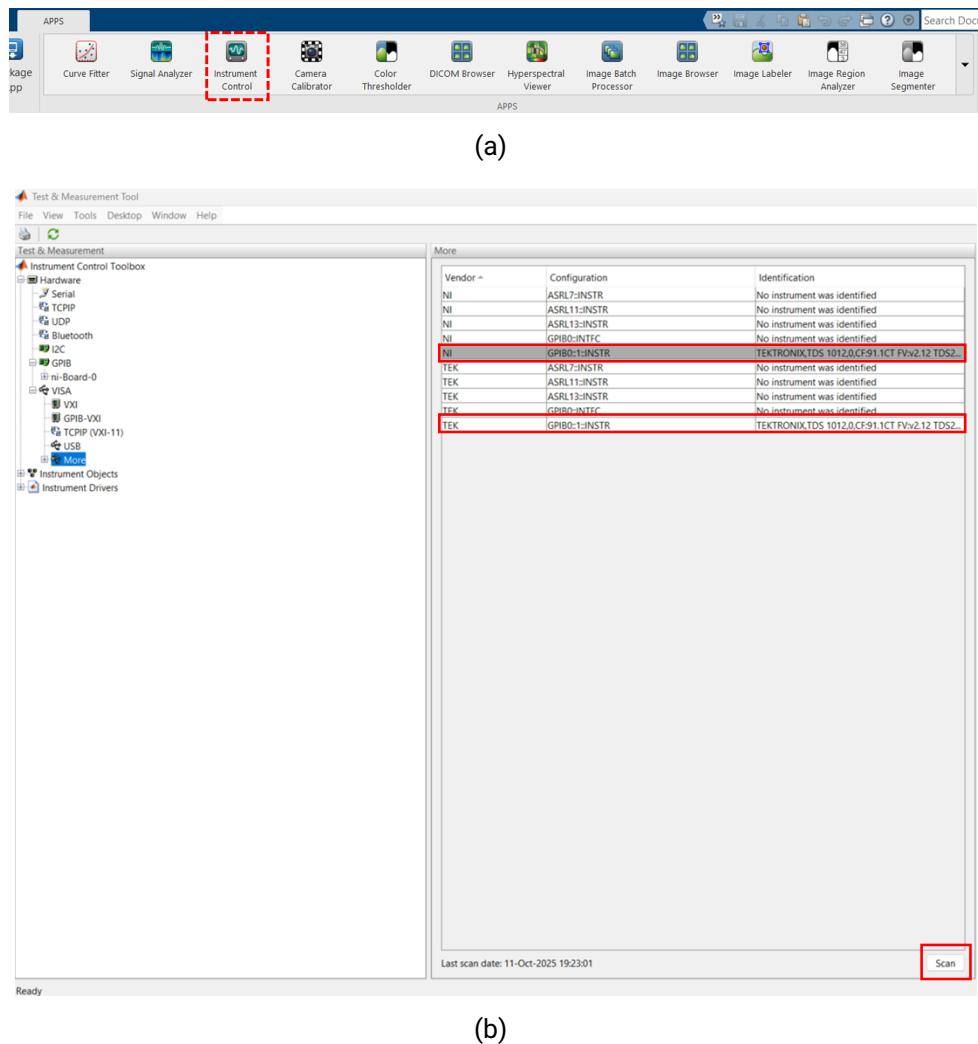


Figure 4: Pasos para verificar la comunicación del osciloscopio. a) Selección de la app Instrument Control y b) Confirmación de visualización del sistema

Una vez conectado, abra *MATLAB* y ejecute el comando:

```
instrhwinfo('visa')
```

o bien:

```
visadevlist
```

sobre la consola de comandos para confirmar que el dispositivo ha sido detectado correctamente.

Verifique que el osciloscopio esté configurado con los parámetros adecuados de comunicación (dirección IP o identificador GPIB, según corresponda).

Finalmente para verificar que todas las funciones auxiliares se encuentren en

la misma carpeta o sean añadidas al path de Matlab puede usarse mediante
`addpath(genpath('guide'))`

5 Descripción de la interfaz grafica

Esta sección detalla los principales componentes que conforman la interfaz de usuario desarrollada en MATLAB GUIDE, describiendo su función, disposición y forma de interacción. El propósito es que el usuario pueda identificar cada elemento visual y comprender su papel dentro del flujo general de operación.

5.1 Componentes de la Interfaz Gráfica de Usuario (GUI)

La interfaz está estructurada en múltiples secciones, cada una de las cuales corresponde a una función específica dentro del flujo experimental. La Figura 5 muestra la sección principal de la interfaz, donde se visualizan en tiempo real las señales ultrasónicas adquiridas por el sistema. Esta sección es fundamental para validar visualmente la calidad de las señales durante la adquisición.

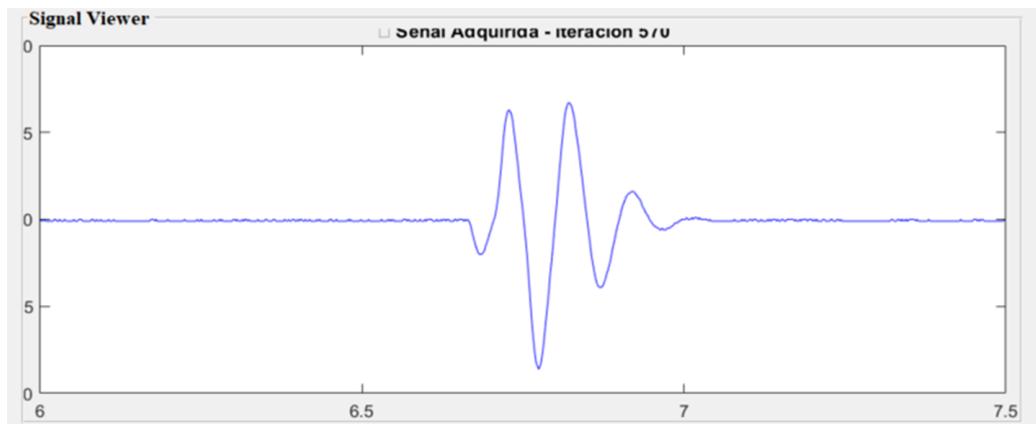


Figure 5: Visualización de señales ultrasónicas obtenidas por el sistema experimental.

En la Figura 6 se ilustran tres componentes clave del control del robot experimental. En la Figura 6a) se observa el módulo de control manual del sistema, el cual permite realizar ajustes finos sobre la posición del transductor. Como se mencionó en la sección anterior, el sistema experimental está conformado por un robot de cinco grados de libertad, regulado por motores a pasos.

En la Figura 6b) se presenta la rutina de adquisición automática, encargada de ejecutar una secuencia predeterminada a lo largo de un vector lineal de adquisición. En esta sección se definen los parámetros de interés, como el número de pasos, la distancia de desplazamiento y la ventana temporal para la visualización detallada de la señal ultrasónica. Además, se incluyen botones de control que permiten detener el sistema de forma inmediata (en caso de un movimiento no deseado), regresar a la posición inicial (evitando la reconfiguración manual) o capturar una señal en el punto actual del desplazamiento (para analizar el comportamiento instantáneo de la señal). Finalmente, la Figura 6c) muestra los puertos y direcciones COM asignados a los controladores y componentes electrónicos del sistema, necesarios para establecer la comunicación entre el entorno MATLAB y el hardware experimental.

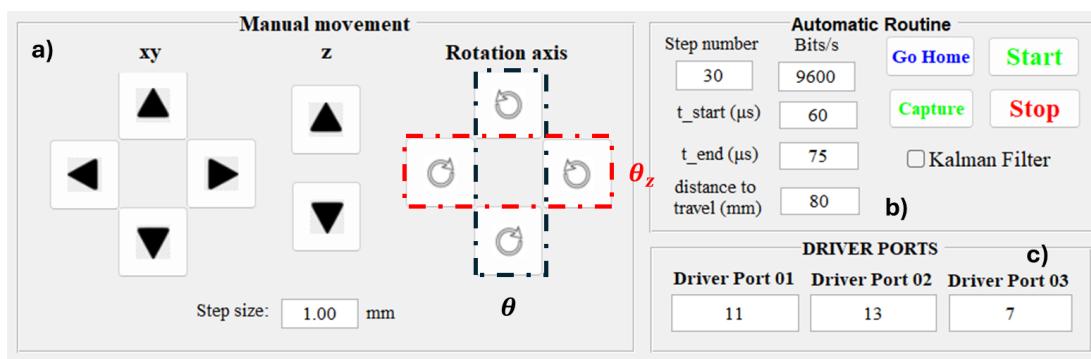


Figure 6: a) Sección de movimiento manual del sistema experimental. b) Rutina automática de adquisición de señales. c) Puertos y direcciones COM asignados para la conexión con los controladores del sistema experimental.

La Figura 6a) muestra de manera ilustrativa que los dos primeros conjuntos de controles corresponden a los movimientos lineales sobre los ejes principales, ejecutando el desplazamiento indicado en función de la Figura 1. En cuanto a los movimientos rotacionales, los botones dispuestos verticalmente controlan la rotación alrededor del eje Z (θ_Z), mientras que los horizontales controlan la rotación del efecto final (transductor) (θ).

Profundizando en la Figura 6b), se muestran los parámetros previos a la adquisición de señales y las rutinas automatizadas:

- **Step number:** número de pasos deseados para la adquisición de señales a lo largo del vector definido.
- **Bits:** tasa de transmisión (*baud rate*) establecida para la comunicación serial de los puertos COM del sistema de movimiento.

- t_{start} y t_{end} : definen la ventana temporal de visualización de la señal, expresada en microsegundos. Es importante establecer esta ventana antes de iniciar la adquisición.
- **Distance to travel**: distancia total que recorrerá el robot durante la adquisición, ajustable según las necesidades experimentales.
- **Start**: inicia la rutina de adquisición automatizada.
- **Kalman Filter**: permite aplicar un filtrado a la señal para reducir el ruido y mejorar su visualización.
- **Go Home**: retorna el sistema a su posición inicial.
- **Stop**: detiene de inmediato cualquier movimiento del robot.
- **Capture**: almacena individualmente la señal adquirida en la posición actual del sistema.

Finalmente, la Figura 6c) presenta los puertos seriales y sus respectivas direcciones COM, empleados para establecer la comunicación con los controladores del sistema robótico y demás componentes electrónicos. La Figura 7 muestra la consola de mensajes, la cual notifica al usuario sobre el estado del sistema, indicando si la adquisición fue completada correctamente, si se detectaron errores de comunicación o si se iniciaron procesos de movimiento o captura de datos.

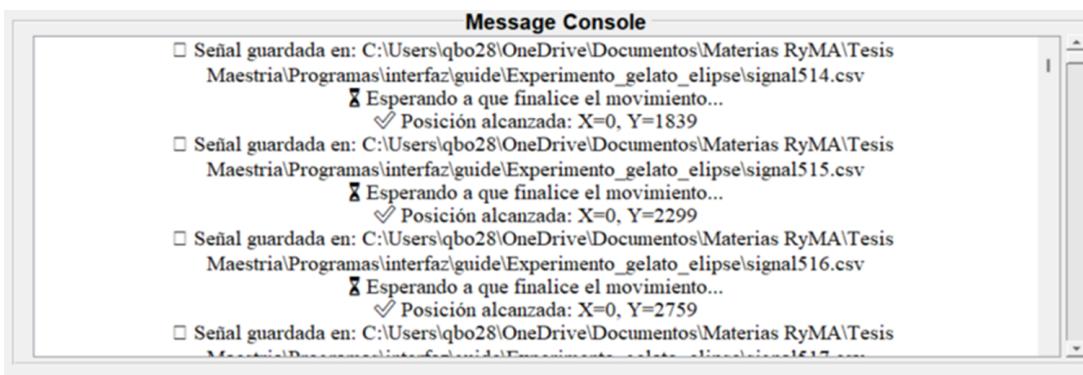


Figure 7: Consola de mensajes, indicando las instrucciones de las acciones ejecutadas sobre el sistema.

La Figura 8 describe la sección de configuración de almacenamiento y adquisición. En (a), se encuentra la asignación de carpetas para el almacenamiento de señales, la identificación del osciloscopio conectado, y la selección del archivo de configuración correspondiente. En (b), se muestra la interfaz de adquisición de parámetros, donde se definen el número de señales, proyecciones y configuraciones de reconstrucción.

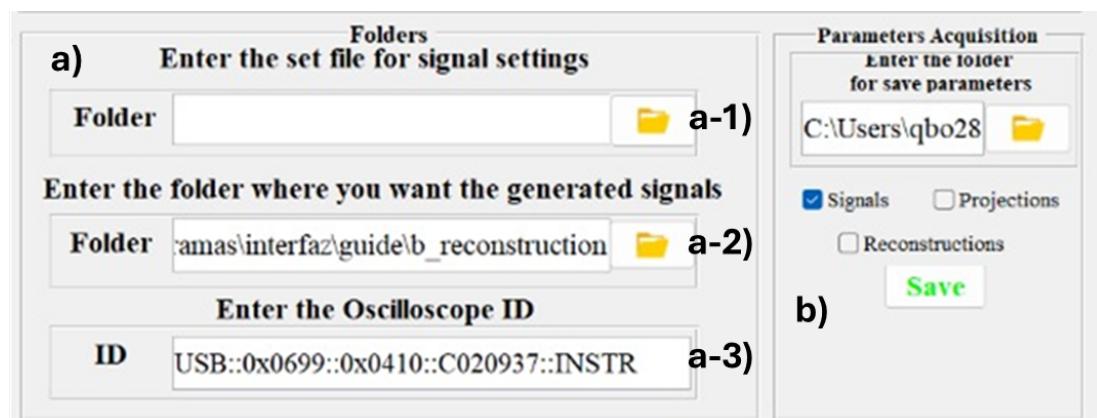


Figure 8: a) Sección de carpetas para el almacenamiento de los archivos adquiridos, asignación del ID del osciloscopio y carga del archivo de configuración. b) Sección de adquisición de parámetros, donde se guardan las señales, proyecciones y reconstrucciones generadas.

La Figura 8a) muestra tres regiones principales destinadas a la gestión de directorios y configuración del sistema:

- **(a-1)** Corresponde al campo para seleccionar un archivo con extensión .set, el cual contiene la configuración deseada del osciloscopio para la adquisición de señales. Este archivo es opcional; si no se especifica, el sistema utilizará la configuración actual del osciloscopio al momento de la adquisición.
- **(a-2)** Indica el directorio donde se almacenarán las señales adquiridas. Es recomendable verificar que la ruta seleccionada tenga permisos de escritura y suficiente espacio disponible.
- **(a-3)** Es el campo destinado a la inserción del *ID* correspondiente al osciloscopio. Para conocer este identificador, consulte los pasos descritos en la Figura 4.

Por su parte, la Figura 8b) muestra la sección donde se especifica el directorio de almacenamiento para las imágenes重建 and las variables de interés, tales como las señales, las proyecciones o las reconstrucciones, en formato .mat. En la Figura 9 se muestra la sección encargada de visualizar las proyecciones recolectadas en cada paso angular del experimento. Esta visualización permite monitorear la cobertura de datos obtenida y verificar su consistencia antes de proceder con el proceso de reconstrucción ultrasónica.

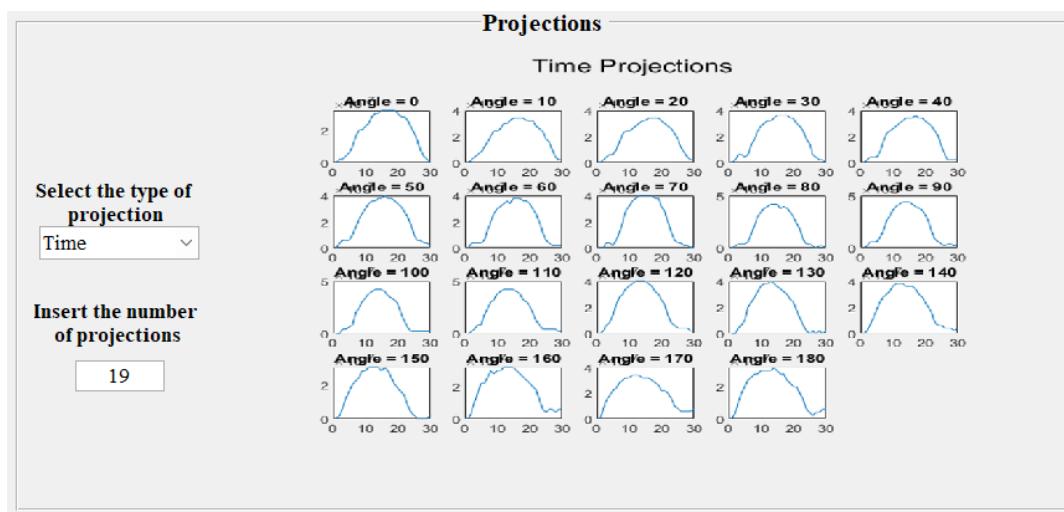


Figure 9: Visualización de proyecciones obtenidas para la reconstrucción de imagen.

Para adquirir las proyecciones correspondientes a los diferentes parámetros de las señales ultrasónicas, basta con seleccionar una de las opciones disponibles y esperar a que se visualice el resultado en el sistema. Una vez mostrada la proyección seleccionada, pueden previsualizarse las demás opciones según el tipo de parámetro deseado.

Las opciones disponibles son las siguientes:

- **Time:** Proyecciones generadas a partir del tiempo de arribo de las señales, aplicando un espejado (inversión) sobre la proyección inicial. Este tipo de proyección suele generar reconstrucciones con intensidades predominantes en tonos oscuros.
- **Time Standard:** Proyecciones obtenidas sin aplicar inversión temporal. Su uso puede producir reconstrucciones con predominancia de intensidades negras, dependiendo del contraste de la señal adquirida.
- **Amplitude:** Proyecciones calculadas a partir de la amplitud máxima del valor absoluto de la señal, útiles para resaltar regiones de alta respuesta acústica.
- **Energy:** Proyecciones basadas en la energía total de la señal, proporcionando una representación global de la potencia acústica registrada.

En la región inferior de la interfaz se encuentra un control que permite definir el número de proyecciones a utilizar en la reconstrucción. Es importante considerar que un aumento en el número de proyecciones implica una mayor cantidad de adquisiciones experimentales, mientras que una reducción en este valor disminuye el número total de señales registradas, mejorando o

empeorando la reconstrucción final.

Finalmente, en la Figura 10, se presentan dos fases críticas del experimento. En (a), se muestra la imagen del objeto a reconstruir junto con el ángulo de rotación actual, una vez que el sistema ha adquirido el número de señales configurado. En (b), se representa la región de reconstrucción final obtenida después de procesar los datos con la técnica de BP, concluyendo así el ciclo de adquisición y reconstrucción.

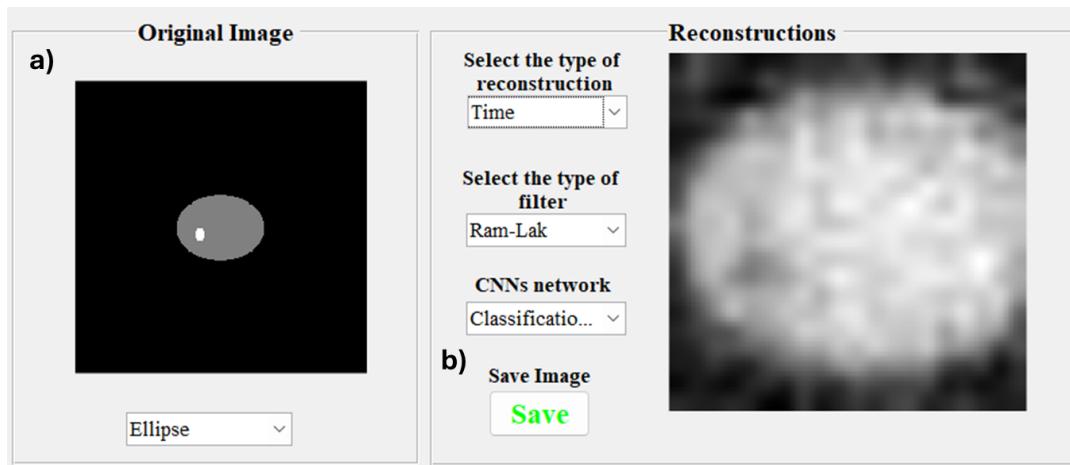


Figure 10: a) Imagen original a reconstruir. Cuando el sistema adquiere el número de señales indicado y se prepara la adquisición, se genera una rotación sobre esta figura, indicando el ángulo en el que se tomarán las señales. b) Región de reconstrucción. Una vez finalizado el experimento, se procesa la información adquirida mediante la técnica de reconstrucción para obtener los resultados experimentales. Posteriormente, pueden implementarse algoritmos de clasificación o detección.

Profundizando en las opciones de esta sección, en la Figura 10a) se muestran las siguientes alternativas para seleccionar la figura original a reconstruir:

- **Ellipse**
- **a character**
- **b character**
- **e character**

Estas opciones representan las figuras de referencia utilizadas como base para la reconstrucción. De esta manera, el usuario puede comparar visualmente la imagen reconstruida con la figura original y evaluar la precisión del proceso. En la Figura 10b), se presentan los parámetros que intervienen en la obtención de la reconstrucción:

- **Tipo de Filtro:** Antes de realizar la reconstrucción, es necesario seleccionar el tipo de filtro que se aplicará al sinograma. Las opciones disponibles son:
 1. **None:** No se aplica ningún filtro.
 2. **Ram-Lak:** Filtro clásico de reconstrucción, ideal para mantener la alta resolución.
 3. **Shepp-Logan:** Atenúa las altas frecuencias reduciendo el ruido.
 4. **Cosine:** Aplica una ponderación suave sobre las frecuencias.
 5. **Hamming:** Minimiza los efectos del truncamiento en el dominio de la frecuencia.
 6. **Hann:** Reduce el ruido de alta frecuencia mediante una ventana de atenuación progresiva.

Para obtener información más detallada sobre el efecto de estos filtros, se recomienda consultar la tesis titulada “**Desarrollo de un sistema de tomografía ultrasónica usando aprendizaje máquina**”, donde se presentan comparativas de reconstrucciones y el análisis de la atenuación de las altas frecuencias provocada por el ruido del sistema. Una vez seleccionado el filtro, se puede elegir el tipo de reconstrucción deseada. Las opciones son análogas a las empleadas en el caso de las proyecciones, con las mismas descripciones y efectos en la calidad del resultado.

Finalmente, el tercer selector corresponde a la red neuronal basada en CNN utilizada para la clasificación o detección de anomalías en las reconstrucciones. Las opciones disponibles son:

- **Classification Ellipses:** Clasifica las reconstrucciones de elipses con y sin anomalías, mostrando en la consola de mensajes la clase asignada.
- **Detection Ellipses:** Detecta y señala las posibles regiones candidatas a ser anomalías sobre la reconstrucción.
- **Classification Letters:** Clasifica las letras *a*, *b* y *e*, con y sin anomalías, mostrando en la consola la clase correspondiente.
- **Detection Letters:** Identifica las posibles regiones anómalas dentro de las reconstrucciones de caracteres.

Esta interfaz integral proporciona un control completo sobre el experimento y garantiza la trazabilidad y organización de los datos obtenidos, facilitando tanto el manejo del sistema como el análisis posterior de los resultados experimentales.

6 Uso basico

6.1 Primeros pasos

Para ejecutar correctamente la interfaz gráfica (GUI), es necesario ubicarse en la carpeta que contiene todos los archivos asociados con el proyecto, incluyendo las funciones auxiliares, las imágenes y los scripts necesarios para su operación.

La Figura 11 muestra la estructura inicial que debe visualizarse en el entorno de trabajo de MATLAB antes de ejecutar el sistema.

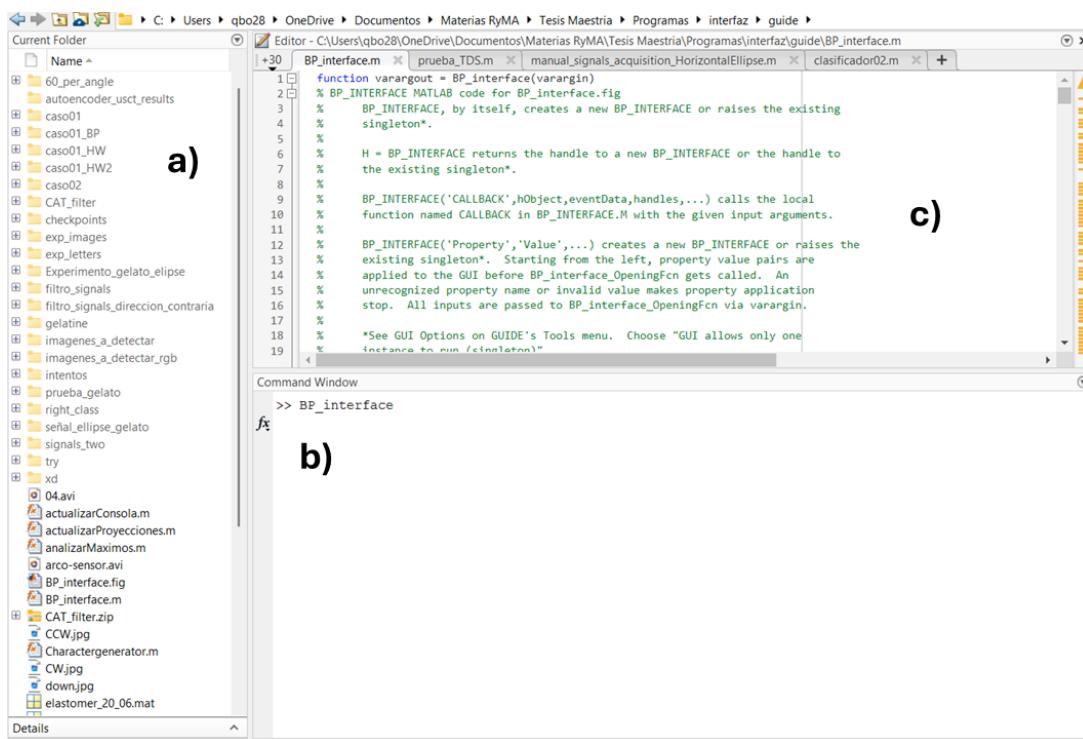


Figure 11: Estructura inicial del entorno de trabajo en MATLAB. a) Carpeta donde se encuentra el archivo principal de la GUI. b) Línea de comando para ejecutar la interfaz, escribiendo el nombre del archivo de la figura. c) Código fuente asociado a la GUI, no es necesario ejecutarlo directamente salvo en caso de comportamiento inesperado.

Como se observa en la Figura 11, para garantizar una ejecución adecuada de la interfaz, MATLAB debe estar posicionado en el directorio que contenga todos los archivos que estructuran el sistema.

Una vez dentro de la carpeta correspondiente, la GUI puede iniciarse escribiendo en la línea de comandos el nombre del archivo de la figura (sin extensión). La Figura 12 mostrará la visualización inicial del entorno una vez que el sistema se haya cargado correctamente.



Figure 12: Vista general de la interfaz gráfica de usuario. Se destacan las secciones principales: control manual, adquisición automática, parámetros de adquisición y visualización de proyecciones y reconstrucciones.

La GUI se compone de los aspectos detallados en la sección 5, para poder realizar la rutina automática del sistema. Dependiendo del tipo de rutina sera la configuración inicial para que el sistema este listo para la adquisición de experimentos

6.1.1 Experimentos para reconstrucción mediante Tomografía Ultrasonica Computarizada (USCT)

Para la adquisición de experimentos orientados a tejidos blandos, es importante considerar la configuración física del sistema. La estructura recomendada para la correcta disposición del conjunto emisor–objeto–receptor se muestra en la Figura 13.

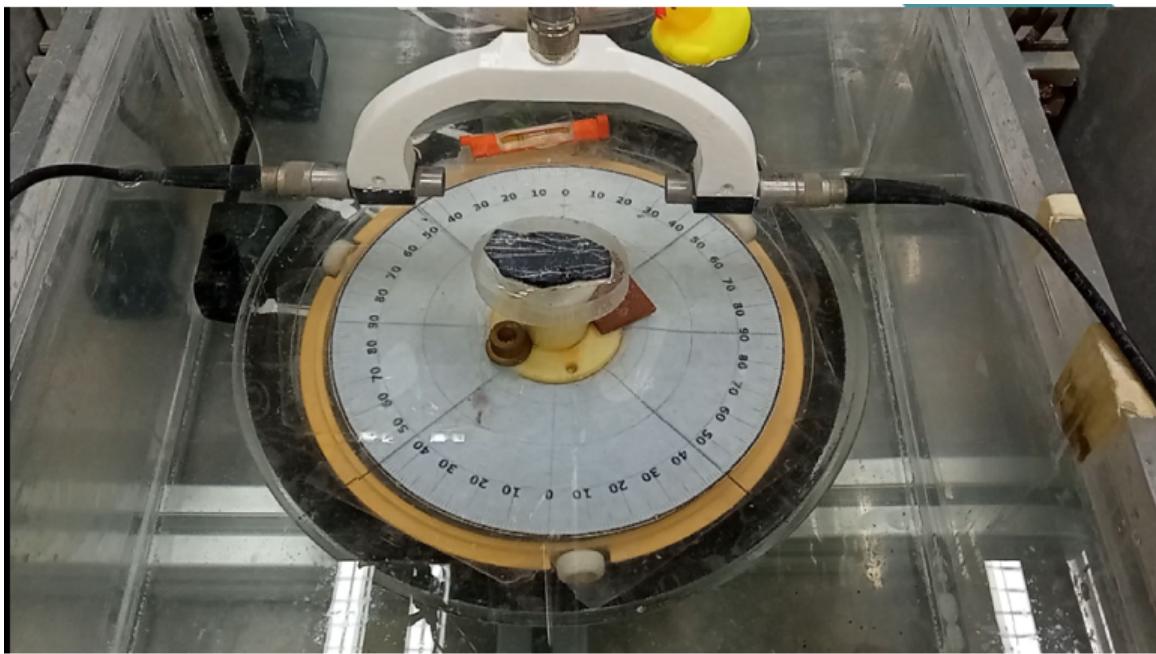


Figure 13: Configuración recomendada para la adquisición de señales experimentales en USCT.

Si se requiere información más detallada sobre los parámetros específicos de adquisición, se recomienda consultar la tesis titulada **Desarrollo de un sistema de tomografía ultrasónica usando aprendizaje máquina**, donde se describen todas las variables y consideraciones experimentales relevantes en la sección de metodología.

Como complemento a lo expuesto en dicho trabajo, es fundamental asegurar la correcta alineación del sistema emisor–objeto–receptor, de manera que el objeto sea detectado dentro del rango de adquisición. Para ello, el operador debe posicionar manualmente el sistema y verificar que el recorrido cubra completamente el objeto bajo estudio sobre el plano xy, como en el eje z. En caso contrario, se deberá ajustar la distancia de desplazamiento o los límites de exploración antes de iniciar el experimento.

La secuencia de pasos para la adquisición y reconstrucción de señales en este sistema es la siguiente:

1. Seleccione la carpeta donde se almacenarán las señales de estudio.
2. Asigne el ID correspondiente al osciloscopio que se utilizará.
3. Seleccione el archivo .set si desea cargar los parámetros de configuración del osciloscopio (opcional).
4. Posicione adecuadamente el sistema emisor–objeto–receptor para la adquisición de señales sobre el objeto de interés.

5. Indique la cantidad de pasos a adquirir y la región de interés temporal para la captura de señales.
6. Presione el botón **Start** y confirme la acción seleccionando “Sí” en el cuadro de diálogo que aparecerá.
7. Una vez concluido el experimento, rote el sistema experimental para repetir la adquisición las veces que sea necesario.
8. Tras completar el número deseado de adquisiciones, seleccione una opción dentro de la ventana **Projections** y espere hasta que se visualicen los resultados.
9. Luego, diríjase a la sección de reconstrucción y elija el tipo de filtro que se aplicará antes de generar la imagen reconstruida.
10. Una vez seleccionado el filtro, elija el método de reconstrucción que desee visualizar y espere a que se muestre el resultado.
11. Si se desea aplicar clasificación o detección de anomalías sobre la reconstrucción obtenida, seleccione la opción correspondiente según el caso de interés.
12. Para guardar la reconstrucción, seleccione una carpeta de destino y presione el botón **Save** dentro de la sección de reconstrucción. (*Nota: se recomienda guardar la imagen antes de aplicar las técnicas de detección*).

6.1.2 Adquisición individual de señales

Si se desea adquirir señales de manera individual sobre un material o sistema, no es necesario contar con una configuración experimental completa como en el caso anterior. Para ello, los pasos a seguir son los siguientes:

1. Seleccione la carpeta donde desea guardar las señales de interés.
2. Asigne el ID correspondiente al osciloscopio que se utilizará.
3. Mueva manualmente el sistema hasta posicionarse sobre la región de interés donde desea adquirir la señal.
4. Indique la región de interés temporal para la captura de señales.
5. Presione el botón **Capture**.
6. Repita los pasos 3 y 4 tantas veces como sea necesario para obtener las señales requeridas para su análisis en posprocesamiento.

7 Funciones avanzadas

Todas las funciones externas al sistema se encuentran documentadas en el repositorio de GitHub: [Cinvestav-LabENDSI](#)⁸. El objetivo de esta sección es mantener el documento más limpio y enfocado en el uso de la interfaz, evitando la saturación con fragmentos de código.

A continuación, se describen brevemente los principales algoritmos implementados en la GUI:

- **actualizarConsola**: Elimina mensajes cuando se supera un número determinado de líneas en la consola, evitando la sobrecarga visual.
- **logMessage**: Muestra mensajes en la consola de la interfaz, permitiendo monitorear en tiempo real las acciones y estados del sistema.
- **leerCSVdesdeCarpeta**: Procesa los archivos CSV almacenados en carpetas, facilitando la carga de datos cuando no se adquieren señales durante el uso de la GUI.
- **kalman_filtrar**: Aplica el filtro de Kalman a las señales ultrasónicas, mejorando su calidad y reduciendo el ruido.
- **SistemaElipses**: Genera una elipse sintética utilizada como figura de referencia en la versión inicial del sistema, para comparar con resultados experimentales.
- **analizarMaximos**: Extrae los parámetros de amplitud, tiempo de arribo y energía de las señales adquiridas durante los experimentos.
- **prepararDatos**: Establece la conexión entre los datos almacenados en carpetas y el procesamiento de las señales, preparando la información para la visualización y reconstrucción.
- **actualizarProyecciones**: Muestra de manera visual las proyecciones del sistema durante la etapa de procesamiento o reconstrucción.

8 Ilustraciones y capturas

Esta sección presenta imágenes descriptivas de la interfaz gráfica, mostrando ejemplos de los resultados obtenidos. El objetivo es ofrecer un apoyo visual para complementar las explicaciones de las secciones anteriores y facilitar la comprensión del flujo de trabajo.

⁸<https://github.com/AbisaídMartinez/Cinvestav-LabENDSI>

8.1 Secciones clave de la GUI

Para mayor claridad, a continuación se presentan imágenes enfocadas en secciones específicas de la GUI:

- **Control manual del robot:** Permite mover el sistema experimental de forma manual y capturar señales en cualquier posición.
- **Adquisición automática:** Ejecuta secuencias predeterminadas para adquirir señales sobre vectores lineales y ángulos definidos.
- **Parámetros de adquisición:** Sección donde se definen los pasos, ventana de interés, número de proyecciones y carpeta de almacenamiento.
- **Visualización de proyecciones:** Permite inspeccionar las proyecciones obtenidas antes de la reconstrucción.
- **Reconstrucción de imagen y clasificación:** Presenta los resultados procesados y, opcionalmente, permite aplicar algoritmos de detección de anomalías mediante redes neuronales.

8.2 Ejemplos de resultados

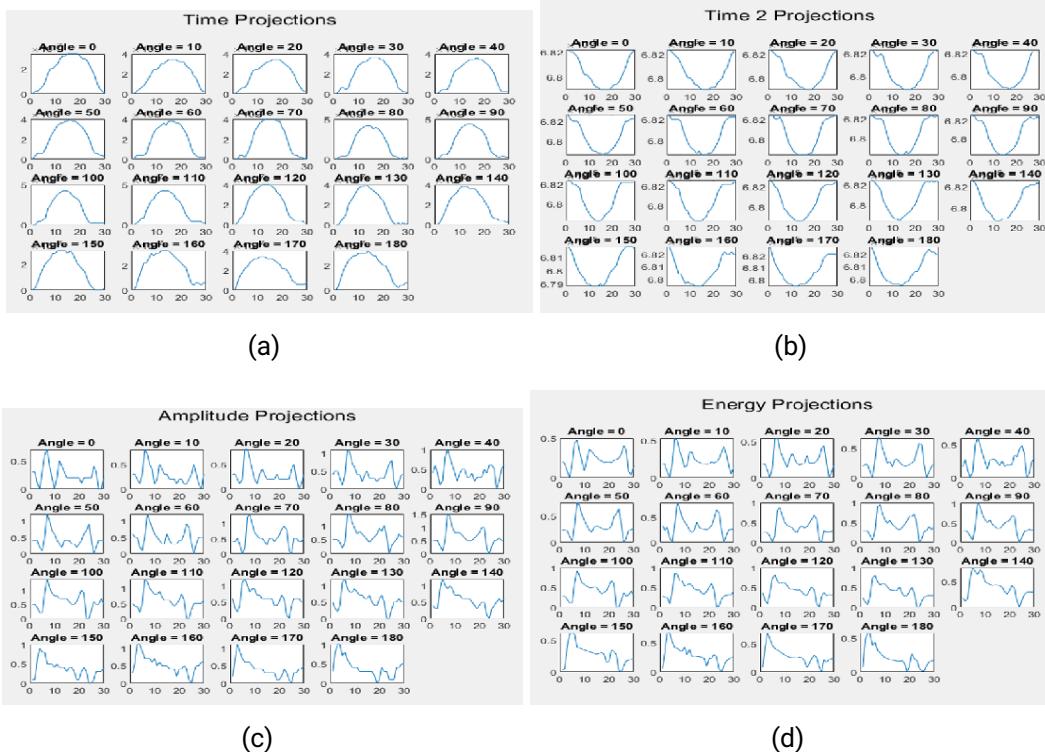


Figure 14: Ejemplo de visualización de proyecciones de un objeto experimental. Se observan las proyecciones en diferentes ángulos.

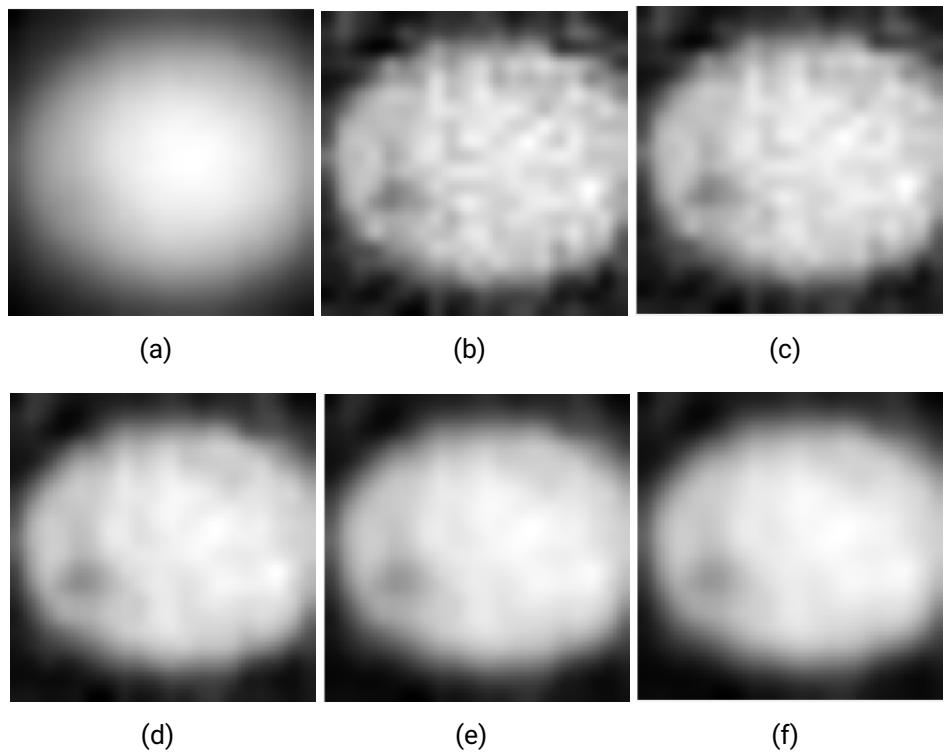


Figure 15: Ejemplo de visualización de reconstrucción de un objeto experimental. Se observan las imágenes reconstruidas utilizando filtros seleccionados en la GUI.

Nota: Todas las imágenes presentadas en esta sección se corresponden con los ejemplos de la ejecución de la GUI, pero los resultados pueden variar dependiendo de los parámetros seleccionados y del objeto experimental.

9 Ejemplos para su uso

Esta sección presenta un ejemplo práctico del uso de la interfaz HMI desarrollada para el sistema de Tomografía Computarizada por Ultrasonido (USCT). En dicho caso, se muestra la interacción entre los módulos de adquisición, control de movimiento y reconstrucción automática.

Para visualizar el video demostrativo de la aplicación en funcionamiento, acceda al siguiente enlace: **HMI Interface - Ejecución de USCT⁹**.

Este ejemplo corrobora la implementación completa del proceso de reconstrucción automatizada mediante USCT, permitiendo la extracción de parámet-

⁹https://www.linkedin.com/posts/jos%C3%A9-abisaid-mart%C3%ADnez-mata-ba60a6263_roboticsengineering-humanrobotinteraction-activity-735438961229269nW7FPJkKAWrgDKx4UfL8SZg

ros físicos a partir de las señales adquiridas. Además, se observa el flujo de trabajo integrado desde la interfaz gráfica hasta la visualización final de las reconstrucciones, validando la comunicación entre el sistema de control y los equipos de medición.

10 Glosario de términos

BP	Backprojection
CNNs	Redes Neuronales Convolucionales
DL	Deep Learning / Aprendizaje profundo
FBP	Filtered Backprojection
GPIB	General Purpose Interface Bus
GUIDE	Graphical User Interface Development Environment
HMI	Interfaz Hombre-Máquina
ResNet	Residual Network
SCPI	Standard Commands for Programmable Instruments
USCT	Tomografía Ultrasonica Computarizada
VISA	Virtual Instrument Software Architecture

11 Contacto y soporte

Esta sección proporciona la información necesaria para contactar al desarrollador o al equipo responsable del mantenimiento del sistema, así como las recomendaciones para la resolución de problemas comunes y canales disponibles para soporte técnico y operativo.

11.1 Tipos de soporte disponibles

Para apoyo al usuario, se contemplan los siguientes tipos de soporte:

- **Soporte Técnico:** Atención a problemas relacionados con la instalación del entorno *MATLAB*, la conexión con los controladores *Velmx* o la comunicación con el osciloscopio *Tektronix*.
- **Soporte Operativo:** Resolución de dudas sobre el uso de la interfaz, la configuración de parámetros, la adquisición de datos o los procesos de reconstrucción ultrasónica.
- **Actualizaciones y mejoras:** Recepción de solicitudes para incorporar nuevas funciones, reportar fallos, realizar sugerencias o garantizar la

compatibilidad con futuras versiones de *MATLAB*.

11.2 Canales de comunicación

En caso de requerir asistencia correspondiente a los dos primeros tipos de soporte, favor de contactar por los siguientes medios:

- Correo: abisaid.martinez@cinvestav.mx
- Linkedin: www.linkedin.com/in/josÃl-abisaid-martÃnez-mata-ba60a6263

Las solicitudes serán atendidas en un periodo estimado de **24 a 120 horas hábiles**, dependiendo de la complejidad del problema reportado.

En caso de que la solicitud corresponda al tercer tipo de soporte (actualizaciones o mejoras), favor de dirigirse al repositorio oficial del proyecto Cinvestav-LabENDSI¹⁰.

Las actualizaciones estarán disponibles conforme a la planeación y disponibilidad del desarrollador principal.

11.3 Recomendaciones antes de solicitar soporte

Antes de contactar al equipo técnico, se recomienda:

- Verificar que MATLAB esté correctamente licenciado y actualizado.
- Confirmar la instalación de los controladores VISA y los drivers Velmex correspondientes.
- Revisar las conexiones físicas (USB, Serial o GPIB) entre la PC, los controladores y el osciloscopio.
- Consultar la sección “Solución de problemas comunes” del presente documento.
- Enviar capturas de pantalla o registros (logMessage) del error si persiste.

11.4 Política de soporte

El soporte está orientado a entornos académicos y de investigación, no a uso comercial. Las solicitudes deben incluir una breve descripción del entorno de trabajo (versión de MATLAB, modelos de hardware Velmex y Tektronix, y sistema operativo).

¹⁰<https://github.com/AbisaidMartinez/Cinvestav-LabENDSI>

12 Errores comunes

En un sistema experimental compuesto por múltiples componentes, pueden surgir fallos durante la adquisición de datos. Es importante identificar rápidamente el componente que genera el mal funcionamiento para corregirlo y continuar con los experimentos de manera eficiente.

Esta sección describe algunos de los errores más frecuentes, cómo se manifiestan en la interfaz y recomendaciones para solucionarlos.

12.1 No se muestra GUI

La Figura 16 muestra el caso del error cuando no se puede abrir la GUI en Matlab

```
>> BP_interface
BP_interface is not found in the current folder or on the MATLAB path, but exists in:
    C:\Users\qbo28\OneDrive\Documentos\Materias RyMA\Tesis Maestria\Programas\interfaz\guide

Change the MATLAB current folder or add its folder to the MATLAB path.
```

Figure 16: Error: no se muestra la GUI al intentar ejecutarla en Matlab.

Para solucionar este problema, es necesario dirigirse a la carpeta que contiene toda la información para ejecutar adecuadamente la GUI.

12.2 No hay movimiento en el robot de inmersión

La Figura 17 muestra un ejemplo de error que ocurre cuando no se genera el movimiento especificado desde la GUI.

```
Error using serialport (line 121)
Unable to connect to the serialport device at port 'COM7'. Verify that a device is connected to the port, the
port is not in use, and all serialport input arguments and parameter values are supported by the device.
See related documentation for troubleshooting steps.

Error in BP\_interface>rightxy\_bottom\_Callback (line 182)
s1 = serialport(puerto_xy, baudrate); % Puerto para el controlador

Error in gui\_mainfcn (line 95)
    feval(varargin{:});

Error in BP\_interface (line 42)
    gui_mainfcn(gui_State, varargin{:});

Error in matlab.graphics.internal.figfile.FigFile/read>@\(hObject,eventdata\)BP\_interface\('rightxy\_bottom\_Callback'\)

Error while evaluating UIControl Callback.
```

Figure 17: Error: incapacidad de conectarse con el sistema robótico para ejecutar el movimiento deseado.

Para solucionar este problema, se recomienda verificar los siguientes puntos:

- Confirme que las conexiones de los controladores estén correctamente identificadas y que los puertos COM asignados coincidan con los que aparecen en la interfaz. De no ser así, realice la corrección correspondiente.
- Si el LED superior mostrado en la Figura 3b) comienza a parpadear, indica que la conexión con el controlador está fallando. En este caso, reinicie el controlador apagándolo durante 10-15 segundos y enciéndolo nuevamente. Cuando se reinicie correctamente, el LED superior permanecerá apagado hasta que se genere nuevamente la interacción de movimiento, momento en el cual se encenderá indicando la regularización del sistema.
- Si hay una correcta comunicación pero se presiona de manera continua el botón de movimiento, se está saturando el buffer del controlador, se recomienda aumentar la velocidad de adquisición o esperar de 1-2 segundos tras presionar la instrucción correspondiente.
- Verifique que la velocidad de comunicación sea la adecuada. Ajuste el parámetro *Bits/s* para asegurar que los controladores puedan comunicarse correctamente con la computadora.

12.3 No hay interacción con el osciloscopio

La Figura 18 muestra un ejemplo de error que ocurre cuando no se pueden adquirir señales desde la GUI.

```
Warning: instrreset will be removed in a future release. For serialport, tcpclient, tcpserver, udpport, visadev, aardvark, and ni845x objects, use delete(serialportfind), delete(tcpclientfind), delete(tcpserverfind), delete(udpportfind), delete(visadevfind), delete(aardvarkfind), and delete(ni845xfind) instead.  
Warning: visa will be removed in a future release. Use visadev instead.  
If you are using visa with icdevice, continue using visa in this MATLAB release.  
Error using icinterface/fopen (line 83)  
Unsuccessful open: Could not open VISA object. Use INSTRHWINFO for a list of available configurations.  
  
Error in BP\_interface>Capture\_signal\_Callback (line 1520)  
fopen(visaObj);  
  
Error in qui\_mainfcn (line 95)  
feval(varargin{:});  
  
Error in BP\_interface (line 42)  
gui_mainfcn(gui_State, varargin{:});  
  
Error in matlab.graphics.internal.figfile.FigFile/read>@\(hObject,eventdata\)BP\_interface\('Capture\_signal\_Callback'\)  
Error while evaluating UIControl Callback.  
--
```

Figure 18: Error: incapacidad de adquirir señales con el osciloscopio.

Para solucionar este problema, se recomienda verificar los siguientes puntos:

- Confirme que la conexión con el osciloscopio esté establecida, siguiendo los pasos indicados en la Figura 4. Si el sistema continúa sin detectar el osciloscopio, intente cambiar de puerto, lo que puede ayudar a restablecer la comunicación.
- Asegúrese de que el ID del dispositivo sea el correcto; de no ser así, realice la corrección correspondiente.
- Verifique que se haya seleccionado una carpeta para guardar los archivos, ya que sin esta selección no habrá un destino para almacenar las señales adquiridas.
- Algunos osciloscopios (de los modelos aceptados en esta interfaz DPO, TDS y MSO) pueden no soportar ciertas funciones o velocidades de procesamiento. Como mejora futura, se recomienda optimizar estos procesos para asegurar una adquisición eficiente de señales con el sistema.

12.4 No es posible visualizar las proyecciones

La Figura 19 muestra un caso donde no es posible visualizar ni procesar las señales para la obtención de las proyecciones del sistema.

```
Unrecognized field name "folderSelected".  
  
Error in prepararDatos (line 3)  
    carpeta = handles.folderSelected;  
  
Error in BP_interface>projection_menu_Callback (line 1296)  
    handles = prepararDatos(hObject, handles, numElementos);  
  
Error in gui_mainfcn (line 95)  
    feval(varargin{:});  
  
Error in BP_interface (line 42)  
    gui_mainfcn(gui_State, varargin{:});  
  
Error in matlab.graphics.internal.figfile.FigFile/read>@(hObject,eventdata)BP_interface('projection_menu_Callback'  
Error while evaluating UIControl Callback.
```

Figure 19: Error: incapacidad de visualizar las proyecciones adquiridas.

Para resolver este problema, se recomienda inspeccionar los siguientes puntos:

- Verifique que la cantidad de proyecciones seleccionadas coincida con la cantidad de experimentos realizados. Por ejemplo, si se adquirieron

90 señales y cada proyección usa 30 señales, asegúrese de ajustar la cantidad de proyecciones para cubrir todas las señales.

- En casos de posprocesamiento, asegúrese de indicar correctamente la carpeta desde donde se tomarán los datos; de lo contrario, el sistema no podrá acceder a ellos.
- Una vez seleccionada una opción, espere hasta que la visualización se complete antes de seleccionar otra. Seleccionar múltiples opciones simultáneamente puede saturar el sistema y generar errores de visualización.

12.5 No se puede visualizar la reconstrucción experimental

La Figura 20 presenta el ejemplo cuando no es posible visualizar la reconstrucción experimental tras la adquisición de las señales ultrasonicas

```
Unrecognized field name "datosGuardados".  
  
Error in BP_interface>menu_reconstruction_Callback (line 1224)  
datos = handles.datosGuardados;  
  
Error in gui_mainfcn (line 95)  
feval(varargin{:});  
  
Error in BP_interface (line 42)  
gui_mainfcn(gui_State, varargin{:});  
  
Error in matlab.graphics.internal.figfile.FigFile/read>@(hObject,eventdata)BP_interface('menu_reconstruction_Callback')  
Error while evaluating UIControl Callback.
```

Figure 20: Error: no es posible visualizar la reconstrucción experimental.

Para resolver este problema, basta con indicar primero cual filtro se desea implementar y posteriormente se puede seleccionar el filtro de interés

12.6 No se pueden guardar las imágenes de reconstrucciones

La Figura 21 muestra un ejemplo de error que ocurre cuando no es posible guardar la reconstrucción obtenida tras el proceso automatizado.

```
Error in matlab.graphics.internal.figfile.FigFile/read>@(hObject,eventdata)BP_interface('Capture_signal_Callback')
Error while evaluating UIControl Callback.

Unrecognized field name "folderSelectedsave".

Error in BP_interface>Save_Image_reconstruction_Callback (line 2084)
    carpeta = handles.folderSelectedsave;

Error in gui_mainfcn (line 95)
    feval(varargin{:});

Error in BP_interface (line 42)
    gui_mainfcn(gui_State, varargin{:});

Error in matlab.graphics.internal.figfile.FigFile/read>@(hObject,eventdata)BP_interface('Save_Image_reconstructi
Error while evaluating UIControl Callback.
```

Figure 21: Error: incapacidad de almacenar la reconstrucción obtenida experimentalmente.

Para resolver este problema, se recomienda:

- Asegúrese de seleccionar correctamente la carpeta de destino donde se desea guardar la reconstrucción, dentro de la sección **Parameters acquisition**.
- Si el error persiste, verifique que las variables relacionadas con el almacenamiento de los archivos estén correctamente configuradas y que la ruta de la carpeta exista y sea accesible desde MATLAB.

Esta solución aplica de la misma manera si se desea guardar las variables mostradas sobre la sección de **Parameters acquisition**, correspondientes a las señales, proyecciones y reconstrucciones en formato .mat

12.7 La reconstrucción no es adecuada

Si la reconstrucción obtenida no cumple con la calidad esperada bajo los parámetros seleccionados, se recomienda considerar las siguientes acciones:

- Ajustar las condiciones iniciales del sistema experimental, asegurando una correcta alineación emisor-objeto-receptor.
- Verificar y, si es necesario, modificar la distancia de adquisición de señales para cubrir todo el objeto de interés.
- Aumentar la densidad de puntos de adquisición y la cantidad de proyecciones, ya que esto contribuye a una mejora significativa en la calidad de la imagen reconstruida.