



Universidad de Valladolid

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN**

TRABAJO DE FIN DE GRADO

**Caracterización Biométrica de un
Array Acústico 2D**

Autor: Pablo Chantre Galindo

Tutor: Alberto Izquierdo Fuente

Índice

1.- Introducción	4
2.- Objetivos	5
3.- Diseño del sistema de captura datos basado en la plataforma myRIO	6
3.1.- Distribución de funcionalidades del sistema de captura y procesado	7
3.2.- Procesos asignados a la FPGA	8
3.2.1.- Introducción al array de microfonos	
3.2.2.- Reloj de FPGA	9
3.2.3.- Sistema de captura en FPGA	9
3.2.4.- Sistema de emisión en FPGA	10
3.2.5.- Sistema de carga de señal en la FPGA	12
3.3.- Procesos asignados al procesador del dispositivo myRIO	13
3.4.- Procesado asignado al PC	15
3.4.1.- Apuntamiento del array y cálculo de desfases	16
3.4.2.- Caracterización del Haz	17
3.4.3.- Calculo de los desfases	20
3.4.4.- Sistema de barrido tridimensional. Resolución	22
3.4.5.- Diezmado de la señal	23

3.4.6.- Calculo de la DFT mediante FFT	24
4.- Diseño de la señal multifrecuencial	26
4.1.- Número de frecuencias	26
4.2.- Elección de frecuencias	27
4.3.- Duración de la señal	28
4.4.- Composición de la señal mediante la suma de tonos	28
4.5.- Generador de señal acústica multifrecuencial optimizada	31
5.- Calibración del sistema	34
5.1.- Compensación de la respuesta frecuencial de cada micrófono	34
5.2.- Compensación de la respuesta frecuencial del altavoz	35
5.3.- Compensación de la respuesta conjunta de altavoz y array	36
6.- Representación de la imagen tridimensional multifrecuencial	37
7.- Conclusiones	39
Anexo	

1.- Introducción

Este trabajo de fin de grado pretende exponer el proceso de desarrollo del software empleado en la captura de datos mediante un array acústico que permita su posterior análisis y utilización en un sistema de identificación biométrica.

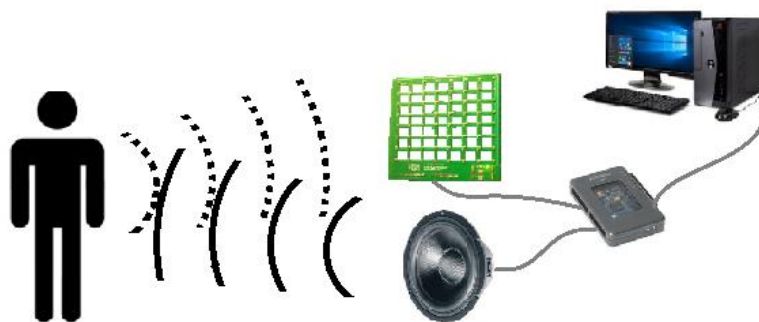
Como punto de partida se tomó un diseño previo en el que se trabaja con un array unidimensional en el que se realizaban capturas utilizando una secuencia de tonos puros de diferentes frecuencias. (Izquierdo-Fuente, Val, Jiménez, & Villacorta, 2011).

En ambos sistemas los datos recogidos se obtienen del análisis de la reflexión de un sonido concreto en el objeto que se desea caracterizar.

El hardware utilizado se compone de un único altavoz como sistema de emisión y de un array bidimensional de 64 micrófonos como sistema de captura.

El procesamiento de datos se realiza de forma distribuida mediante un dispositivo NI myRIO, conectado con los sistemas de emisión y recepción, y un PC, en el que se almacenan los datos obtenidos.

Este hardware permite obtener una representación tridimensional de un objeto. Si se obtiene la representación de una persona, dicha representación contiene datos biométricos suficientes para diferenciar entre sujetos, pudiendo así formar parte de un sistema de identificación biométrica.



2.- Objetivos

El objetivo principal es obtener imágenes tridimensionales con suficiente resolución para permitir su utilización en un sistema de identificación biométrico.

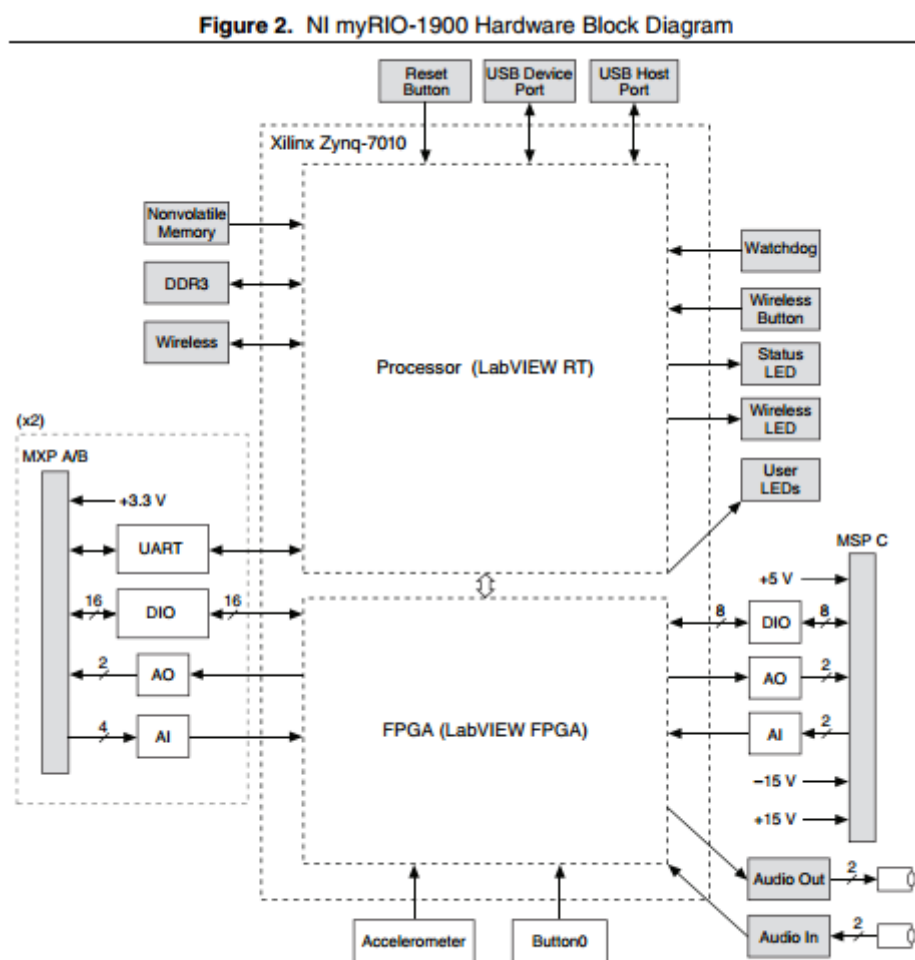
Para obtener dichos datos se plantea la caracterización del sujeto a identificar mediante la utilización de múltiples frecuencias de forma simultánea.

Los objetivos específicos son:

- a) Determinar las frecuencias utilizadas teniendo en cuenta tanto parámetros intrínsecos al sistema de captura como los parámetros derivados del método de procesado de datos que se elija, agrupándolas en una señal multifrecuencial viable.
- b) Diseñar un sistema de captura y procesado basado en la plataforma myRIO de National Instruments y utilizando el lenguaje de programación Labview.
- c) Diseñar e implementar un algoritmo de conformación electrónica basado en FFT que permita obtener imágenes acústicas.
- d) Realizar pruebas de captura de imágenes de un conjunto de personas para validar el sistema de captura y procesado espacial multifrecuencial.

3.- Diseño del sistema de captura datos basado en la plataforma myRIO

Como elemento central del sistema se emplea un dispositivo myRIO. Este dispositivo es capaz de gestionar una serie de entradas y salidas tanto digitales como analógicas, además de realizar un procesamiento a tiempo real de las señales recibidas.



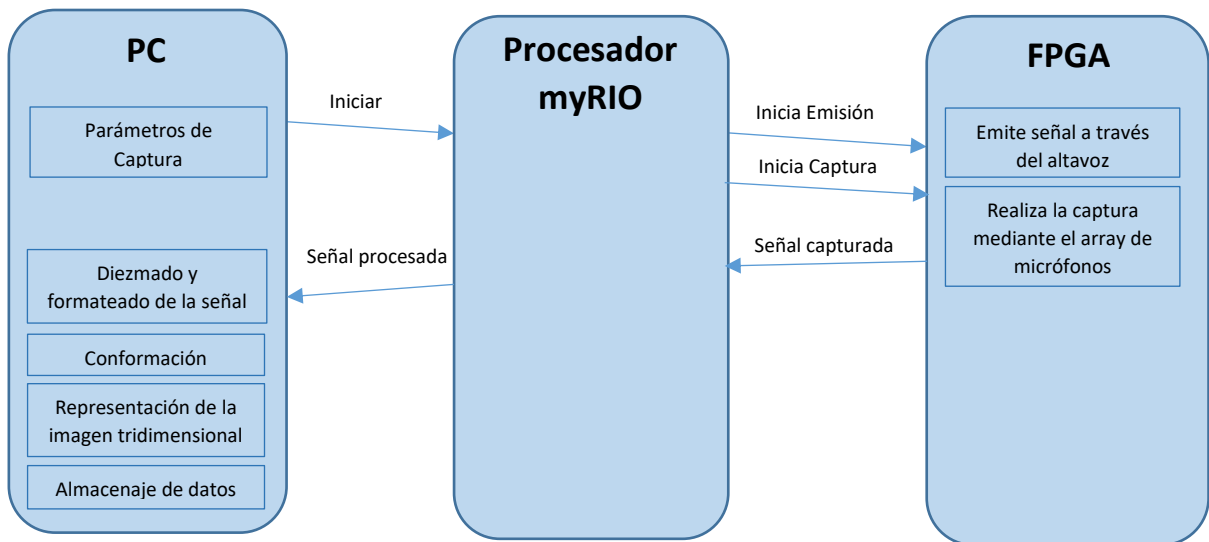
Como se puede apreciar en este diagrama, el dispositivo está compuesto por dos secciones claramente diferenciadas. Por una parte, las entradas y salidas están controladas por una FPGA (Field Programmable Gate Array).

Por otra parte, el dispositivo cuenta con un procesador que, además de soportar el software Labview ejecutando los programas de procesamiento a tiempo real que se introduzcan, se encarga de gestionar una pequeña memoria no volátil (en la que se pueden almacenar dichos programas para poder ser utilizado como una unidad de procesamiento independiente) y de establecer una conexión inalámbrica con una máquina externa.

3.1.- Distribución de funcionalidades del sistema de captura y procesamiento

Con el propósito de aprovechar todo lo posible las ventajas que ofrece el dispositivo myRIO, se plantea un sistema de procesamiento distribuido con la siguiente repartición de tareas:

- Tanto el altavoz como el array de micrófonos conectan con la FPGA a través de los puertos de entrada y salida del dispositivo myRIO, por tanto, la FPGA se encargará de proporcionar la señal para transmitir y de obtener los datos de cada uno de los micrófonos.
- El procesador del dispositivo myRIO controla las acciones de la FPGA, iniciando el proceso de emisión y de captura. Establece y gestiona la conexión con el PC.
- El procesamiento final que permite obtener las imágenes tridimensionales buscadas se realiza en un PC. En esa máquina se realiza el procesamiento de la señal y se visualizan las imágenes obtenidas y se almacenan los datos para su posterior uso en el sistema de identificación biométrica.



3.2.- Procesos asignados a la FPGA

Para que la FPGA realice un proceso concreto, todo el circuito debe ser reprogramado para tal propósito. Esto lleva el tiempo suficiente como para que no se pueda reprogramar durante la ejecución, por lo que todos los subprocesos que se realicen en la FPGA deben estar incluidos en un mismo esquema para que tras un único proceso de compilación y reprogramación queden disponibles para su uso simultáneo.

Este esquema único se compone de cuatro subprocesos. El primer sistema proporciona una señal de reloj para ser utilizada por el array de sensores y por los demás bloques. El segundo sistema se encarga de obtener los datos procedentes del array de micrófonos, el tercero se encarga de realizar el envío de la señal hacia el altavoz, y el cuarto tiene la única función es cargar la señal que se va a enviar o sustituirla por otra.

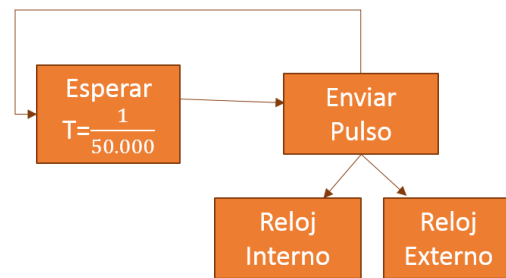
3.2.1 Introducción al array de micrófonos

El elemento hardware empleado para la captura es un array cuadrado de 8x8 micrófonos de tipo MEMS (*MicroElectroMechanical Systems*). Estos micrófonos producen una señal digital cuya codificación hace que cada muestra de la señal ocupe un solo bit. Esta codificación binaria se basa en emitir un 1 si la señal recibida aumenta su intensidad y un 0 si disminuye.

Para su correcto funcionamiento, requieren una señal de reloj externa de 1MHz, con la que los micrófonos emiten 1 bit por ciclo de reloj.

3.2.2.- Reloj de FPGA

La FPGA tiene una señal de reloj interna a 40 MHz (A un periodo de este reloj interno se le denomina *1 tick*, y se utiliza como unidad de medida para los tiempos de cada subproceso de la FPGA). Esta frecuencia es demasiado alta para utilizarla en este sistema.



*Implementación en la Figura 1 del Anexo

Por este motivo se añade este bloque, que genera una señal de reloj de frecuencia 1MHz.

Esta señal de reloj se envía a través de una línea del conector C hacia el array de micrófonos. Además se utilizará en los demás bloques de este esquema con lo que se logra que el circuito integrado del array y el circuito programado de la FPGA se sincronicen.

Este bucle se mantiene en funcionamiento en todo momento desde que se inicia la FPGA.

3.2.3.- Sistema de captura en FPGA

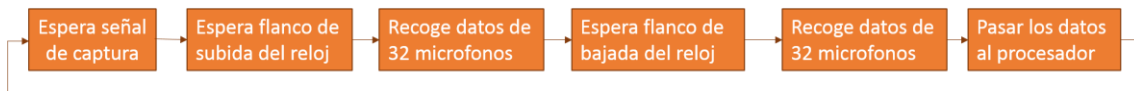
El array de micrófonos se conecta al dispositivo *myRIO* mediante dos puertos digitales, A y B, cada uno con 16 líneas de entrada o salida (en este caso actuarán de entradas al sistema). Entre los dos conectores se dispone de 32 líneas de entrada digitales.

Dado que se tienen 64 micrófonos, es necesario multiplexar dos señales para cada línea de entrada digital.

La multiplexación se realiza tomando los datos de la mitad de los micrófonos en el flanco de subida de la señal de reloj, y los datos de la otra mitad en el flanco de bajada.

Una vez se dispone de una muestra de cada uno de los 64 canales, se agrupan en una palabra de 64 bit, y se escriben como un elemento en una pila FIFO. Esta pila es el mecanismo de intercambio de información entre la FPGA y el procesador del *myRIO*

De esta pila, el procesador extrae los datos para el procesado a tiempo real.



*Implementación en la Figura 2 del Anexo

El primer bucle de espera que se observa en este diagrama logra que este bloque se mantenga disponible y a la espera de recibir la orden de iniciar la captura.

Una vez iniciada la captura, todos los pasos de la secuencia se realizan en un solo ciclo del reloj de 1 MHz. Esto es posible porque las funciones que componen cada paso, incluidas las de lectura de los puertos de entrada y salida que requieren varios ciclos, se realizan guiadas por el reloj de 40 MHz.

Dado que las muestras se escriben una vez por cada iteración del bucle, en la pila se almacena una señal de 64 canales muestreada a 1 MHz.

3.2.4.- Sistema de emisión en FPGA

El sistema de emisión está diseñado para cumplir con varias funcionalidades que pueden requerirse según las circunstancias de hardware empleado o para realizar un proceso de calibración del sistema.

La función principal de este bloque es el de emitir a través de un canal analógico de salida del dispositivo myRIO la señal multifrecuencial diseñada específicamente para este sistema, y por lo tanto no puede aprovechar los generadores de señal que el software *Labview* tiene integradas.

La señal multifrecuencial la proporciona el operador del sistema a través del PC, no obstante, debe estar almacenada en la FPGA antes de iniciarse el proceso de emisión.

Esto se debe a que el proceso de lectura de una variable externa a la FPGA lleva demasiado tiempo como para poder proporcionar la señal muestra a muestra con una frecuencia de muestreo suficiente.

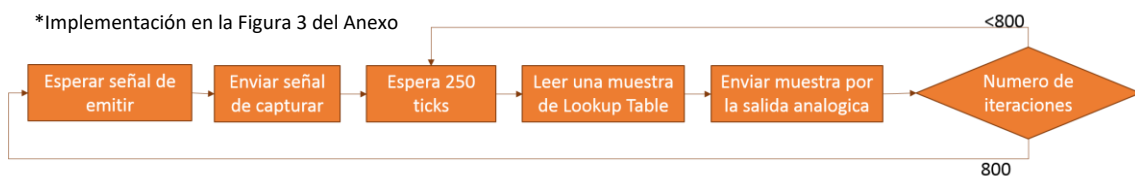
Para solucionar esto, una parte de la FPGA se programa para actuar como una memoria interna de la cual se pueden extraer los datos en un solo ciclo de reloj (del reloj de 40

MHz). Este espacio de memoria se configura como una tabla (*LookUp Table*, LUT) en la que se almacena cada muestra de la señal que se va a emitir.

Utilizando este método, la FPGA es capaz de emitir como máximo una muestra de la señal cada 224 *ticks* (periodos de su reloj de 40MHz), lo que supone una frecuencia de muestreo máxima de aproximadamente 178,57 KHz, debido a las limitaciones que impone el conversor D/A a través del que se accede al canal de salida analógico.

La precisión en la frecuencia de muestreo es esencial para que este sistema funcione correctamente, por lo que es conveniente utilizar una frecuencia de muestreo que pueda ser reproducida con facilidad a la hora de generar la señal.

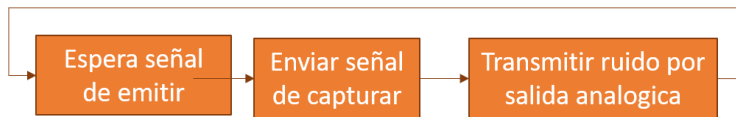
La frecuencia de muestreo elegida es **160 KHz**, que corresponde con leer de la *LookUp Table* y enviar por un canal analógico una muestra cada 250 *ticks*. Si tenemos una tabla con **800 muestras**, la señal producida tendrá una duración de **5 ms**, lo que, como se verá más adelante, es apropiado para los demás parámetros del sistema.



En el diseño anterior se puede observar como se ha duplicado el sistema de emisión, creando una segunda *LookUp Table* de las mismas características que la principal, y a la que se accede simultáneamente. Las muestras obtenidas de esta segunda tabla se emiten a través de un segundo canal analógico.

Esta duplicidad permite, si fuese necesario por la arquitectura hardware elegida, emitir la misma señal o señales complementarias por dos altavoces independientes.

Este bloque contiene una funcionalidad alternativa que permite enviar ruido a las dos salidas analógicas de forma simultánea. Mediante un control se permite alternar entre el envío de señal y el de ruido.



*Implementación en la Figura 4 del Anexo

Para producir el ruido se emplea un Generador de Ruido Blanco Gaussiano incluido en *Labview*. La capacidad para emitir este tipo de ruido es útil en procesos de testeado del hardware. A diferencia del caso de emisión de señales almacenadas, dado que el ruido se genera en la propia FPGA, la duración del mismo no está determinada a priori; está determinada por un control externo.

Sería posible añadir con facilidad un tercer módulo alternativo que permitiese la emisión de tonos puros generados en la misma FPGA, pero nada impide cargar una señal que contenga un tono puro en una de las *Lookup Tables*, por lo que se estarían duplicando funcionalidades.

3.2.5.- Sistema de carga de señal en la FPGA

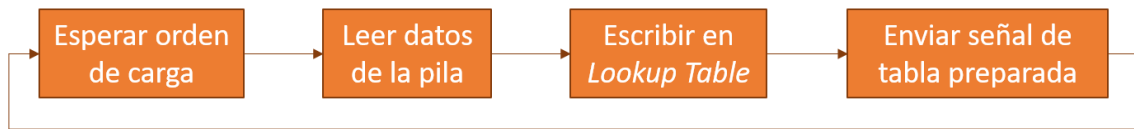
Como se ha explicado, la señal multifrecuencial tiene que estar almacenada en la FPGA para poder ser emitida a la frecuencia de muestreo requerida.

El diseño y posterior generación de esa señal se lleva a cabo en el PC, por lo que tiene que ser transmitida a la FPGA en un procedimiento previo a la primera captura.

Con la finalidad de disponer de los datos de la señal en la FPGA, pero para que estos puedan ser modificados en ejecución, se emplea el recurso de las *Lookup Tables*, que son memorias dentro de la FPGA en las que se puede almacenar y extraer datos mediante procedimientos de lectura escritura rápidos.

En las *Lookup Tables* los datos de la señal perduran hasta que se desconecta el dispositivo. Cuando el dispositivo se reinicia las tablas recuperan un valor predefinido en la compilación.

Es conveniente, por tanto, determinar que señal se grabara en la compilación ya que esta se comportara como una señal por defecto. Para preparar otra señal para su emisión se hace uso del siguiente bloque.



*Implementación en la Figura 5 del Anexo

En este diseño se realiza la carga de dos señales de forma simultánea en dos *Lookup Tables*. En el proceso se emplean dos pilas de datos que almacenan de forma temporal la señal. Dichas pilas reciben los datos del procesador del dispositivo *myRIO*, realizando el proceso inverso al que se sigue en la captura, con la ventaja de que en este caso no es necesario imponer una limitación a la velocidad con la que se carga y descarga la pila, por lo que todo el proceso de carga de datos se realiza de forma casi instantánea.

Al igual que los bloques de captura y emisión, este también tiene un bucle que se mantiene a la espera de una señal que inicie el proceso. La particularidad en este caso viene dada por la incompatibilidad con el bloque de emisión, por lo que es necesaria la implementación de un semáforo que impida la ejecución del bloque de emisión hasta que no se haya completado la carga de las nuevas *Lookup Tables*.

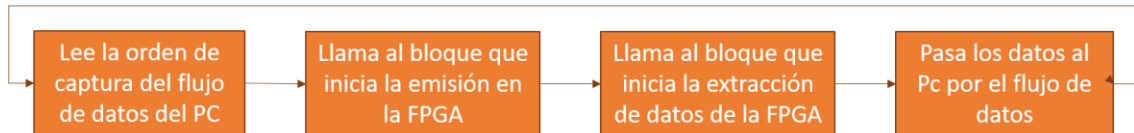
3.3.- Procesos asignados al procesador del dispositivo *myRIO*

La función principal del procesador del dispositivo *myRIO* es actuar de intermediario entre el PC y la FPGA.

Por una parte, la conexión con el PC se realiza creando un flujo de datos a través del cual se transmiten los órdenes y parámetros de captura desde el PC hacia la FPGA y se envían los datos capturados hacia el PC.

Este flujo de datos se establece al iniciar el software en ambos extremos y se mantiene conectado mientras ambos extremos se mantengan activos.

En cambio, las interacciones con la FPGA se realizan mediante sesiones. Cada una de las sesiones se establece con un propósito concreto y se cierra cuando este se ha llevado a cabo. Durante estas sesiones el procesador puede pasar parámetros hacia la FPGA, incluyendo las órdenes de iniciar emisión y captura. El intercambio de datos se realiza mediante las pilas FIFO anteriormente mencionadas.



*Implementación en la Figura 6 del Anexo

Mediante el diagrama de bloques anterior se realiza la recepción de un paquete de parámetros procedente del PC relativos al proceso de captura, así como la orden de iniciar dicho proceso.

En primer lugar, se lanza el subprocesso encargado de dar comienzo a la emisión de la señal almacenada en la *Lookup Table* de la FPGA (Implementación en la Figura 7 del Anexo).

Como parte del proceso de emisión en la FPGA comienza también al proceso de captura, que comienza a escribir los datos capturados en la pila FIFO.

Esta pila tiene una capacidad limitada, por lo que es esencial que el proceso de extracción de datos de la pila comience inmediatamente después de iniciarse el de emisión (Implementación en la Figura 8 del Anexo). De otro modo la pila llegaría a un estado de desbordamiento y se perderían los datos.

Mediante este bloque se leen los datos de la FIFO, extrayéndolos en el mismo orden en que entraron. Los datos extraídos de la pila se transmiten hacia el PC a través del flujo de datos establecido.

Cuando todos los datos de la pila se han pasado al flujo de datos este sistema vuelve a un estado de espera, preparado para recibir otro paquete del PC que dé comienzo a una nueva captura.

3.4.- Procesos asignados al PC

La parte del sistema de captura que se ejecuta en el PC, además de servir de punto de acceso para que el operador interactúe con el sistema, realiza el procesamiento de los datos obtenidos.

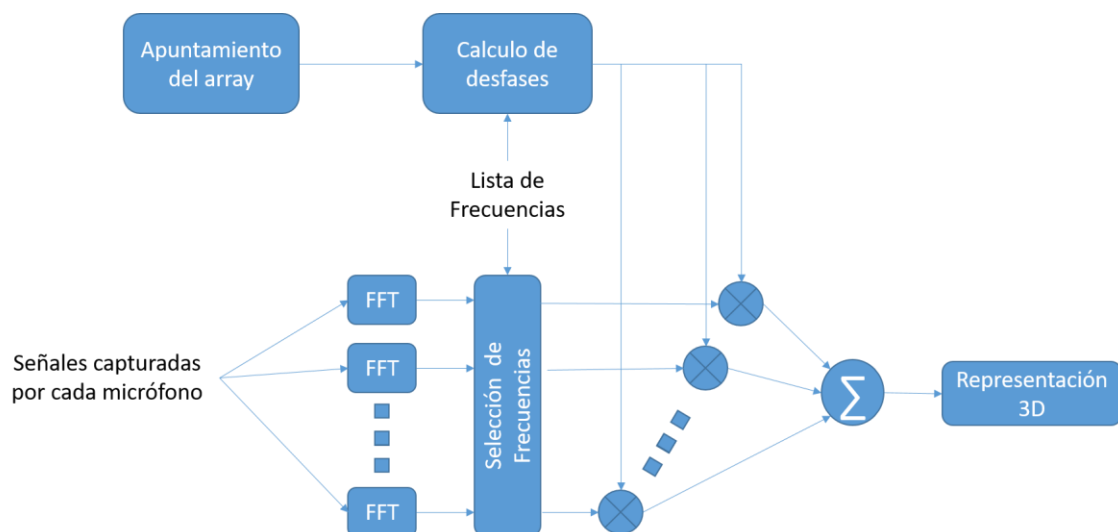
Este procesamiento tiene como objetivo interpretar la información que contienen los datos y representarla como una imagen tridimensional multifrecuencial del objeto que se quiere caracterizar.

Para obtener la información necesaria relativa a cada punto del espacio que permita realizar una representación tridimensional es necesario realizar un proceso de conformación de haz (*beamforming*) mediante el cual se concentra el array hacia la posición concreta que se quiere caracterizar.

La conformación de haz se puede realizar mediante dos métodos diferentes: Introducir retardos o introducir desfases para cada una de las señales recibidas. Ambos procesos son equivalentes. Mientras que, en el método de los retardos, se trabaja con las señales en el dominio temporal, en el método de los desfases es necesario realizar una Transformada de Fourier en Tiempo Discreto (DFT) para introducir los desfases en el dominio frecuencial.

En este sistema se utilizará el método de los desfases, implementando la DFT mediante el algoritmo FFT proporcionado por el software *Labview*.

Este procesamiento se compone de múltiples etapas:



Como se puede observar, hay ciertas etapas en las que no interviene la señal capturada. En estas etapas se emplearán los parámetros utilizados para realizar la captura, como las frecuencias empleadas y la posición del objeto que se desea caracterizar.

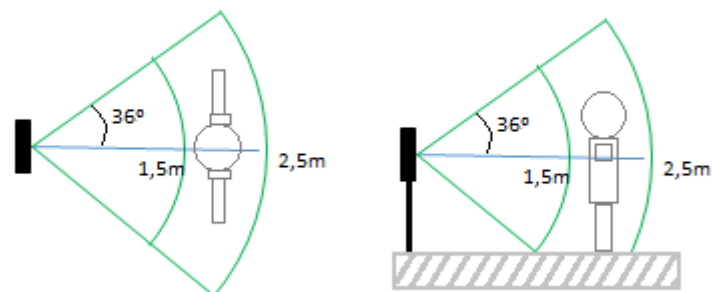
3.4.1.- Apuntamiento del array y cálculo de desfases

El primer paso para realizar el procesado es identificar un volumen en el que se encuentre contenido el objeto y caracterizarlo en un sistema de coordenadas concreto que permita identificar una serie de puntos en los que muestrear.

El sistema de coordenadas más común cuando se trabaja con array es el esférico, ya que se adecua bien a las formas descritas por los diagramas de radiación. Sin embargo, en este caso utilizar el sistema de coordenadas esférico complica sustancialmente los cálculos de forma innecesaria.

Teniendo en cuenta que el objetivo del sistema es la caracterización biométrica, un volumen razonable para contener a una persona con los brazos extendidos tendrá unas medidas aproximadas de 2x2x1 metros. Situando el array a 1 metro de altura tendremos un sistema equivalente para las direcciones vertical y horizontal.

Por este motivo el sistema de coordenadas elegido medirá las distancias angulares tomando como origen la dirección perpendicular al array, midiendo primero la variación de acimut, después la elevación y finalmente la distancia. El volumen elegido, será el contenido entre dos secciones de esfera de radios 1,5 y 2,5 metros cuyo centro es el centro del array. Y las distancias angulares respecto a la perpendicular serán de $\pm 36^\circ$.



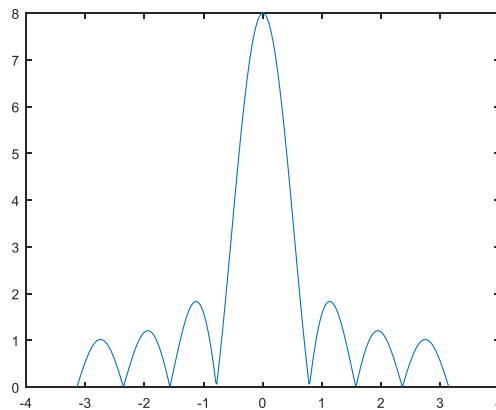
3.4.2.- Caracterización del Haz

Para describir la dirección en la que apunta un array se utiliza el concepto de “Factor de Agrupación” o “Factor de Array”, que es una representación de la ganancia del array en función de la dirección desde el centro del mismo.

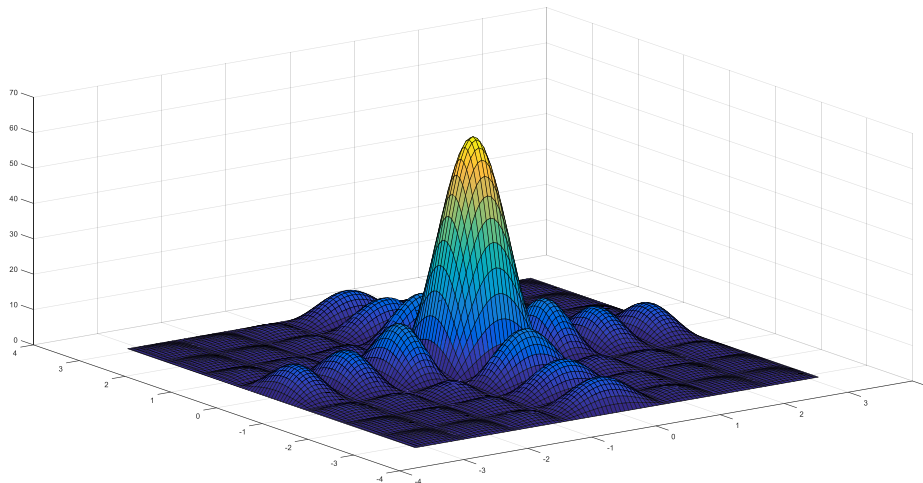
Al tratarse de un array plano uniforme (todos los elementos del array tienen la misma ganancia, o lo que es lo mismo, contribuyen en igual medida a la señal final) su Factor de Agrupación se puede descomponer en el producto de los factores de agrupación en sus dos direcciones.

$$FA = FA_x \cdot FA_y = \frac{\sin\left(8 \frac{\psi_x}{2}\right)}{\sin\left(\frac{\psi_x}{2}\right)} \cdot \frac{\sin\left(8 \frac{\psi_y}{2}\right)}{\sin\left(\frac{\psi_y}{2}\right)}$$

Además, al tratarse de un array cuadrado, ambos factores serán equivalentes. Ambos se pueden calcular como si se tratase de un array lineal uniforme de 8 elementos, cuyo factor de agrupación tiene la siguiente forma:



Al combinar ambos factores de agrupación se obtiene el siguiente diagrama, en el que se puede observar la ganancia en el eje perpendicular al array es de 64, muy superior a la del resto del diagrama:



Este factor de agrupación está calculado en función de Ψ (ángulo eléctrico), cuya expresión incluye la frecuencia de la onda que se va a emplear y la distancia entre los elementos del array. Aparece, además, la variable con la que se modifica la dirección de apuntamiento del array: el desfase progresivo α , que refleja la diferencia de fase de la señal al alcanzar cada elemento del array, producida por la diferencia en la longitud del camino que la señal debe recorrer para alcanzar a cada elemento.

$$\Psi_x = \frac{2\pi}{\lambda} \cdot d \cdot \sin(\theta_x) + \alpha_x$$

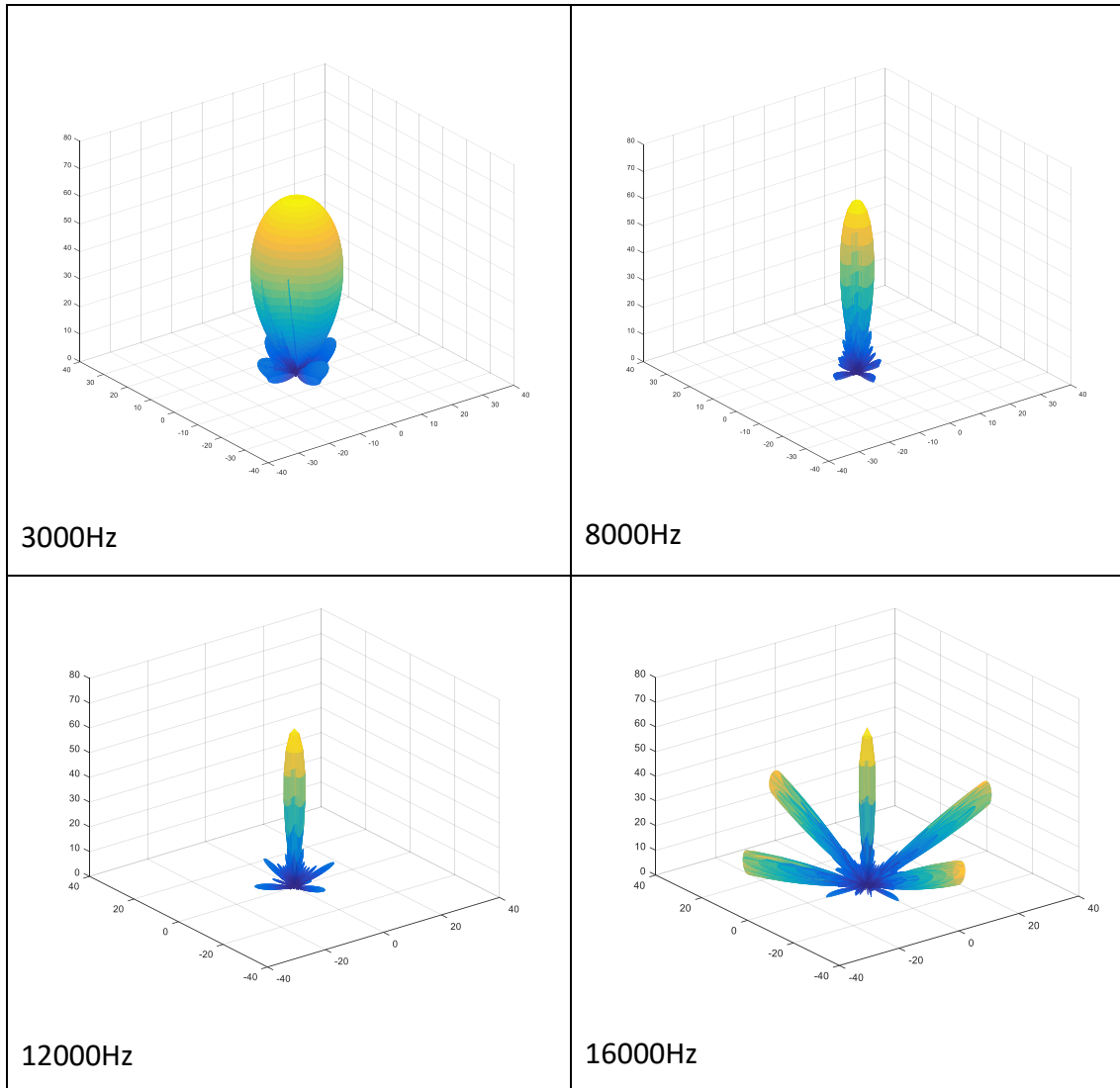
$$\Psi_y = \frac{2\pi}{\lambda} \cdot d \cdot \sin(\theta_y) \cdot \cos(\theta_x) + \alpha_y$$

Dado el amplio rango de frecuencias contenidas en la señal multifrecuencial, el comportamiento del array varía notablemente de unas frecuencias a otras.

En la siguiente figura, sin incluir ningún desfase, es decir, manteniendo la dirección de apuntamiento del array perpendicular, se puede observar claramente cómo se estrecha el haz a medida que se incrementa la frecuencia.

Esto implica que para frecuencias más bajas la precisión es menor. Sin embargo, cuando se incrementa demasiado la frecuencia empiezan a aparecer *Grating Lobes*.

Dado que la separación entre los micrófonos del array es de 2.5 cm, se puede calcular que, para un desfase nulo, los *Grating Lobes* aparecen a partir de los **13720Hz**, aunque empieza a aparecer cierta contaminación a frecuencias ligeramente inferiores.



La aparición de los *grating lobes* afecta más al resultado de este proceso cuanto más alta es la frecuencia y cuanto más pronunciado es el ángulo del apuntamiento respecto a la perpendicular del array, ya que los lóbulos no deseados se apuntan hacia la zona del objetivo, y por tanto se produce una imagen múltiple.

Para garantizar que los *grating lobes* no apuntasen al volumen que contiene al objetivo para ninguno de los ángulos de apuntamiento necesarios para caracterizarlo, la frecuencia no debería superar los 11500 Hz.

No obstante, debido a la mayor resolución que proporciona trabajar a altas frecuencias debido al estrechamiento del haz, se consideraran frecuencias superiores, luego hay que tener en cuenta que se producirá este fenómeno indeseado.

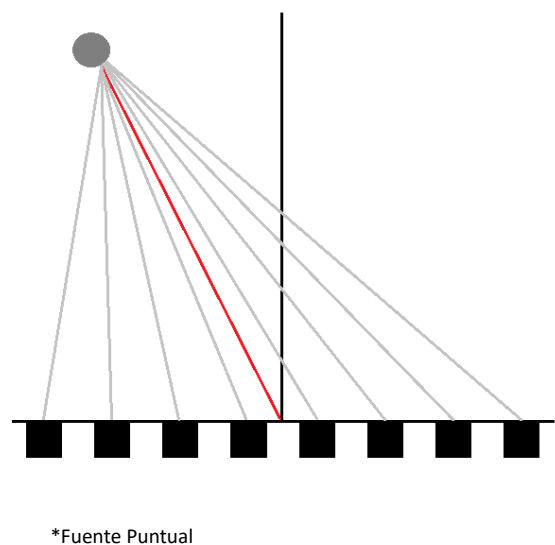
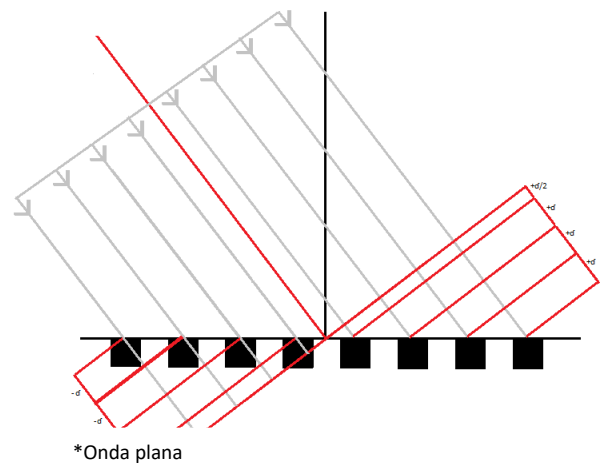
3.4.3.- Calculo de los desfases

Mediante el método anterior se podrían calcular unos desfases basados en la dirección de apuntamiento. Los desfases calculados mediante este método tendrían su base en la suposición de que la onda que alcanza el array es plana. Esto es razonable cuando el punto en el que se pretende concentrar la sensibilidad del array es lejano.

Empleando la suposición de onda plana, basta con introducir un desfase progresivo, es decir, cada a elemento se le introduce un desfase d multiplicado por su distancia al centro del array.

Teniendo en cuenta que este sistema pretende realizar un apuntamiento tridimensional, se ha considerado una mejor opción realizar el apuntamiento basado en la posición de cada punto que se desea muestrear, y no en función de una dirección.

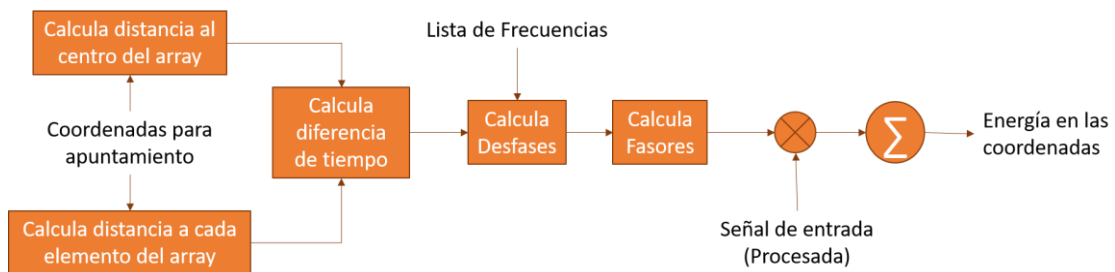
Al suponer una fuente puntual, en vez de una onda plana, el cálculo de los desfases se basa en calcular la distancia entre cada elemento del array (d) y la posición de apuntamiento y compararla con la distancia al centro del array (D). El desfase para cada elemento se obtendrá calculando la diferencia en el tiempo que tarda la señal en recorrer ambos caminos y multiplicándola por la frecuencia.



Esto implica que los desfases dependerán de la frecuencia de trabajo y de la velocidad del sonido, además de la posición del elemento en el array bidimensional, por lo que obtendremos una matriz (**M**) de desfases (δ) tridimensional de $8 \times 8 \times N$, siendo N el número de frecuencias utilizadas.

$$M_{\delta}(a, b, i) = \frac{2\pi}{343} \cdot (d_{a,b} - D) \cdot f_i$$

La metodología a seguir para implementar este procedimiento consiste en realizar un bloque al que se le proporcionen como parámetros de entrada las coordenadas de un punto y la FFT de la señal (cuyo cálculo se detallara más adelante).



*Implementación en la Figura 9 del Anexo

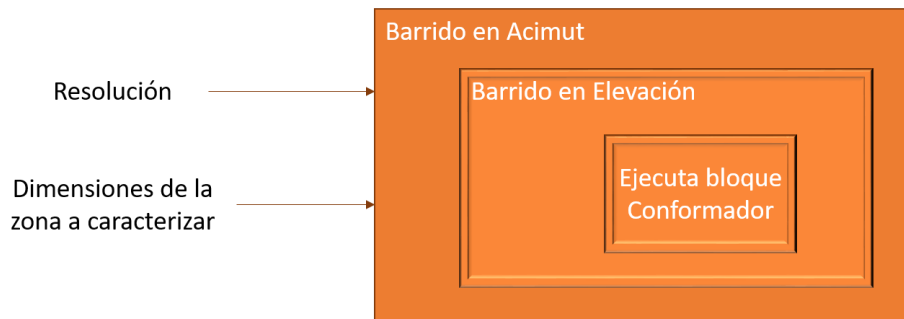
Este bloque se compone de dos partes. En primer lugar, se calculan las distancias a cada elemento del array de forma iterativa. La segunda parte se trata de un bucle que itera utilizando las frecuencias como índice.

En este bucle se emplea la señal en frecuencia, cuya obtención se explicará a continuación. Se multiplica cada canal por los fasores calculados para el elemento del array correspondiente en función de los desfases: $e^{i \cdot d \cdot \delta}$. Posteriormente se suman todos los canales desfasados.

De todo este proceso se obtiene un valor concreto para cada frecuencia para el punto concreto que se está analizando, que corresponde con la energía de señal reflejada proveniente de ese punto en cada una de las frecuencias.

3.4.4.- Sistema de barrido tridimensional. Resolución.

El paso siguiente es utilizar el bloque anterior de forma iterativa, proporcionando secuencialmente una serie superficies concéntricas al centro del array, y recorriendo cada una de dichas superficies variando acimut y elevación en incrementos controlados. Realizar este barrido implica realizar una serie de bucles anidados: Un bucle que barre en distancia, que contendrá otro que barre en acimut, que a su vez contiene otro que barre en elevación. Este último contiene el bloque de cálculo de desfases, que como se explicó anteriormente itera en frecuencia. En total esto implica un barrido en 4 dimensiones.



*Implementación en la Figura 10 del Anexo

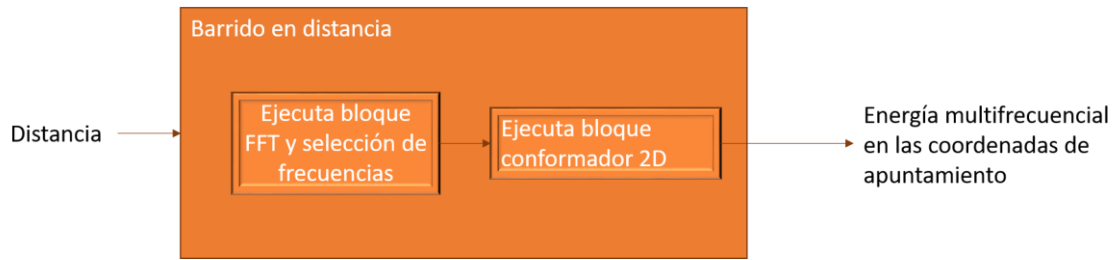
A falta de concretar la resolución, es decir, el número de puntos en los que se tomaran valores en cada una de las dimensiones, es evidente que la carga computacional es muy elevada.

Es evidente, por tanto, que no se debe incrementar la resolución de forma innecesaria o ineficiente.

En teoría, la resolución puede ser tan alta como se desee, no obstante, existe un límite determinado por el ancho del haz a partir del cual una mayor resolución no aporta más datos. Esta resolución máxima efectiva depende de la frecuencia utilizada.

En el bloque anterior solo se realiza el barrido en las dos dimensiones del array. El barrido en distancia se realiza en otro bloque externo que realiza una iteración de este para cada corte en distancia que se desee realizar.

Esto es necesario, ya que el dato de la distancia es necesario para realizar el procesamiento de la señal previo a la FFT y el filtrado.



*Implementación en la Figura 11 del Anexo

Esto implica que, como se puede ver en el diseño del bloque, es necesario realizar el filtrado de la señal para cada distancia. Por este motivo, la resolución en distancia incrementa mucho más la carga computacional que la resolución en las otras dos dimensiones, por lo que el número de cortes en distancia debe ser muy inferior a la resolución de las otras dimensiones.

La señal una vez realizado el filtrado se pasa al bloque que realiza el barrido en las otras dos dimensiones.

3.4.5.- Diezmado de la señal

La señal obtenida del dispositivo myRIO tiene una frecuencia de muestreo de 1 MHz. Se ha determinado que la frecuencia de trabajo máxima estará en torno a los 20 KHz, con lo cual, según el teorema de Nyquist, se dispone de un número de muestras muy superior al necesario, lo que implica una carga computacional innecesaria.

Además, hay que tener presente que se va a realizar un filtrado en frecuencia mediante FFT en el cual, como se detallará a continuación, una elevada frecuencia de muestreo en el tiempo implica una resolución frecuencial menor.

Por este motivo previamente a cualquier otro procesamiento de la señal, se realiza un diezmado con un factor 20, logrando una señal de **50KHz** con la que aún se tiene un cierto margen para que no se produzca aliasing al trabajar con las frecuencias más altas del rango establecido.

Al trabajar con las señales almacenadas en vectores, la implementación del diezmado consiste en crear un nuevo vector para cada una de las 64 señales en el que se almacena una de cada 20 muestras de las señales originales.

Este diezmado no se ha realizado en un bloque individual, sino que se implementa directamente en el bloque principal que se ejecuta en el PC (Figura 0 del Anexo).

3.4.6.- Cálculo de la DFT mediante FFT

El diseño del utilizado se basa en un procesamiento mediante FFT (Fast Fourier Transform), que es un algoritmo para el cálculo rápido de la Transformada de Fourier Discreta (DFT, por sus siglas en inglés).

La Transformada de Fourier Discreta es equivalente a realizar un muestreo de la Transformada de Fourier en Tiempo Discreto. Es decir, obtiene una señal en frecuencia discreta a partir de una señal en tiempo discreto.

El software Labview proporciona un bloque que realiza una FFT tomando como parámetros de entrada una señal muestreada y el número de puntos que se evaluarán en frecuencia.

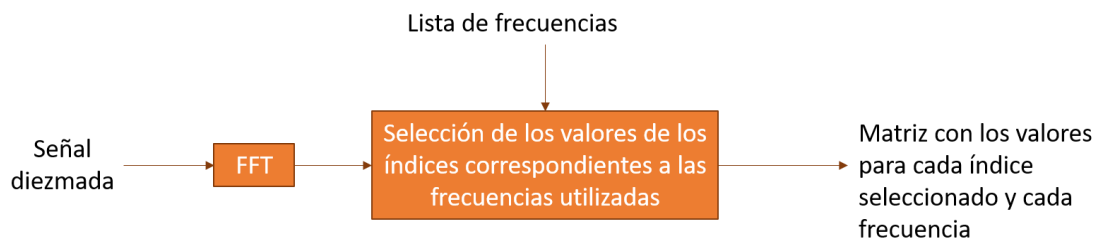
La separación entre los puntos de la FFT es de la forma

$$\Delta f = \frac{\text{Frecuencia de muestreo de la señal}}{\text{Numero de puntos de la FFT}}$$

Aquí interviene el diezmado que se realizó anteriormente, ya que, si la frecuencia de muestreo de la señal de entrada a la FFT es muy alta, o bien se incrementa el número de puntos de la FFT, con lo que crece enormemente la carga computacional, o bien se trabaja con una resolución frecuencial muy baja, lo que limita el número de frecuencias que se pueden utilizar.

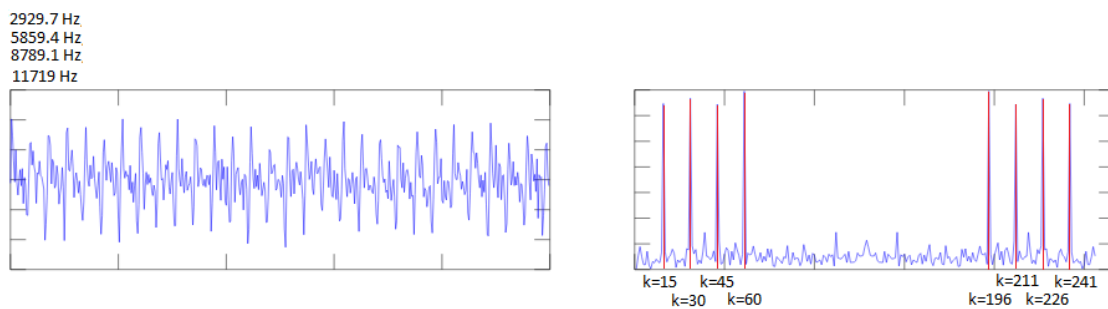
Tomando la frecuencia de muestreo de 50 KHz obtenida tras el diezmado, se ha elegido realizar una FFT de 256 puntos, alcanzando un compromiso entre la carga computacional y la resolución frecuencial. Esto permite una distancia entre frecuencias de **195.3125 Hz**.

La clave del funcionamiento de este sistema es que las frecuencias empleadas para la caracterización coincidan exactamente con puntos de la FFT.



*Implementación en la Figura 12 del Anexo

Dado que la señal que se envía se compone exclusivamente de tonos puros, al muestrear exactamente en la frecuencia de esos puntos, es posible elegir los índices k exactos a los que se encuentra la frecuencia de interés, eliminando gran parte de la influencia del ruido.



*Ejemplo de selección de índices k de la FFT para una señal de 4 tonos.

A la salida de la FFT se tienen vectores de 256 puntos por cada canal. Conociendo la lista de frecuencias de trabajo, se calcula el índice del punto que corresponde a cada frecuencia. Tras esto, se extraen los valores correspondientes a los índices de cada una de las frecuencias de interés para cada micrófono.

Esta matriz compacta, que contiene información sobre la energía captada en las frecuencias utilizadas para cada micrófono, es la que se utiliza como señal de entrada en el bloque del conformador visto anteriormente.

4.- Diseño de la señal multifrecuencial

En el proceso de elaboración de la señal acústica que se emplea en este sistema se deben tener en cuenta las restricciones impuestas por tanto por el hardware de que se dispone como por el procesado descrito:

- Limitación en la potencia de la señal: Tanto por el límite del conjunto amplificador-altavoz como por los límites de seguridad por tratarse de un sistema biométrico.
- Frecuencias utilizadas: Es necesario determinar que frecuencias coincidirán con los puntos muestreados al realizar el procesado mediante FFT.
- Duración de la señal: Esta limitada por el tiempo de ida y vuelta del sonido entre el array y el blanco.

4.1.- Número de frecuencias

Para empezar a componer la señal a utilizar lo primero que es necesario determinar cuántas frecuencias utilizar. La cantidad de datos obtenida es mayor cuantas más frecuencias se utilicen.

Sin embargo, dado que la potencia de señal se dividirá entre las diferentes frecuencias utilizadas, cuantas más frecuencias se utilicen se tendrá una peor relación señal a ruido para cada frecuencia lo que disminuirá la calidad de los datos.

Por este motivo, aun planteando un sistema inicial con un número elevado de frecuencias, el software de captura se diseña con la capacidad de modificar las frecuencias utilizadas con facilidad. De este modo se podrá aumentar el número de frecuencias si se logra una relación señal a ruido suficiente, o disminuirlo en el caso contrario.

La posibilidad de modificar las frecuencias utilizadas abre la puerta a eliminar una frecuencia concreta en caso de que se compruebe experimentalmente que no está aportando información, aumentando así la energía de que se dispone para las demás.

Manteniendo el concepto de número variable de frecuencias, se tomarán 32 frecuencias como el estándar de frecuencias utilizadas.

4.2.- Elección de frecuencias

Las frecuencias que se utilicen en el sistema de emisión deben estar adaptadas a las condiciones del sistema de recogida de datos para que este resulte eficiente.

Tal y como se detalla en el apartado dedicado al procesado, la señal de 64 canales proveniente del array de micrófonos, se diezma y queda muestreada a 50KHz. Como parte esencial del procesado se realiza una FFT de 256 puntos.

Esto supone que a partir de este procesado se trabajara con una señal discreta en frecuencia, y, por tanto, no se utilizara la información que la señal inicial pudiera contener entre las muestras.

El rango de frecuencias que es posible utilizar viene dado por el teorema de muestreo de Nyquist. Teniendo una frecuencia de muestreo de 50KHz, la frecuencia más alta que el sistema de recepción será capaz de procesar sin sufrir aliasing será de 25KHz.

Además, para no desperdiciar energía, las frecuencias utilizadas se harán coincidir exactamente con las frecuencias a las que se muestrea la FFT, que en adelante podrán ser identificadas con un índice entero k desde 0 a 255. Las frecuencias que corresponden a cada k se pueden obtener con la siguiente formula:

$$f_k = k \frac{50}{256} = 195.3125 k$$

Esto implica que cualquier frecuencia múltiplo de 195.3125 que sea menor de 25KHz será elegible para ser utilizada en este sistema.

Un ejemplo de 32 frecuencias, que de hecho es el utilizado por defecto en los experimentos aquí reflejados, serían las frecuencias calculadas de la forma

$$f_n = 1757.8125 + n 585.9375$$

Dichas frecuencias están equiespaciadas entre **1757.8125 Hz** y **19921.875 Hz**.

4.3.- Duración de la señal

La duración de la señal emitida está limitada para evitar que se superponga con la reflejada cuando esta alcance el array de micrófonos.

Consideremos una señal de duración T_s segundos. Si tenemos tanto el array de micrófonos como el altavoz situados a una distancia v de el blanco, la duración máxima de la señal vendrá dada por:

$$T_s < 2 \frac{d}{v}$$

Donde v es la velocidad del sonido, que, en condiciones estándar, tendrá un valor aproximado de 343 m/s.

Situando el blanco entre 1,5 m de distancia del conjunto del array y altavoz, tendremos una duración máxima de señal:

$$T_s < 8.746 \text{ ms}$$

4.4.- Composición de la señal mediante la suma de tonos

La señal utilizada será una suma de tonos simples de la misma amplitud en las frecuencias elegidas. La suma de tonos de diferentes frecuencias produce señales de amplitud diferente en función de los desfases entre los tonos.

Dado que la señal final tiene una duración determinada y una amplitud máxima limitada, la energía máxima de la misma también es limitada.

Ya que la señal contiene múltiples tonos de la misma amplitud, la energía de la señal esta equidistribuida entre esos tonos. Si queremos realizar un análisis en un número relativamente alto de frecuencias, la energía disponible en cada una de ellas será muy baja, por lo que optimizar la composición de la señal será crítico para mejorar la relación entre la amplitud de los tonos utilizados y la amplitud de la señal suma.

Teniendo determinadas las frecuencias que se van a incluir, todas con la misma amplitud, solo queda un parámetro sobre el que realizar la optimización de la señal: la fase de cada tono.

Esto supone una función de optimización no lineal en N variables siendo N el número de frecuencias utilizadas:

$$\min_{\varphi} \left[\sum_N \sin(2\pi f_i + \varphi_i) \right]$$

La solución de esta función de minimización proporcionaría un vector φ de desfases que producirían la señal de menor amplitud posible para la combinación de las frecuencias que se han elegido.

La función propuesta no tiene una solución analítica que pueda ser evaluada en las frecuencias elegidas, por lo tanto, la única opción posible es recurrir a algún procedimiento de cálculo numérico.

Dado que el número de frecuencias propuesto es de 32 frecuencias, se plantearía un problema con una muy elevada carga computacional.

El método elegido para resolver dicho problema es realizar la optimización por un método de Montecarlo.

Inicialmente se ha optado por implementar este método mediante un programa en Matlab.

Utilizando un fichero con las frecuencias deseadas, se realizan experimentos utilizando vectores de desfase aleatorios y midiendo el factor de reducción en la amplitud de la señal final que se obtiene. Si el resultado obtenido es mejor que los obtenidos anteriormente, se almacena el vector de desfases, en caso contrario se descarta.

```

1 - load Frecuencias.mat
2
3 - Nex=1000;
4
5 - Tmax=1/min(f);
6
7 - Tmin=1/max(f);
8
9 - lf=length(f);
10 - t=0:Tmin/40:40*Tmax;
11
12 - for k=1:lf
13 -     C(k, :) =cos(2*pi*f(k)*t);
14 -     S(k, :) =sin(2*pi*f(k)*t);
15 - end
16
17
18 - minimo=lf;
19 - Amin=zeros(lf, 1);
20 - tic
21 - for K=1:Nex
22 -     A=2*(rand(lf,1)-0.5);
23 -     fun=max(A'*C+(1-(A'.^2)).^(1/2))*S);
24 -     minimo=fun*(minimo>fun)+minimo*(minimo<fun);
25 -     Amin=A*(minimo>fun)+Amin*(minimo<fun);
26 - end
27 - toc
28 - minimo
29 - desfases=acos(A);
30
31 - save Desfases.mat desfases

```

*Implementación en Matlab del optimizador

Realizando este proceso iterativo, alcanzado un número suficiente de experimentos, la reducción en la amplitud de la señal es sustancial. Si bien no se puede garantizar que el resultado obtenido sea optimo, el factor de reducción obtenido mediante este sistema supone una mejora sustancial en el aprovechamiento de la energía del sistema.

El proceso, no obstante, se ha comprobado ineficiente debido al tiempo que se requiere para realizar el procesado. Incluso con máquinas muy potentes, alcanzar un resultado razonablemente bueno lleva muchas horas, lo que dificultaría enormemente cambiar las frecuencias de trabajo del sistema, ya que para cada nuevo grupo de frecuencias sería necesario realizar el proceso de optimización.

Intentando reducir este tiempo de computación se decide realizar este procesado mediante empleando el software Labview.

Al utilizar este software se obtienen factores de reducción de la amplitud semejantes a los obtenidos mediante el código anterior en una fracción del tiempo, reduciendo las horas que suponía realizar el proceso de obtener una nueva señal a pocos minutos.

Teniendo en cuenta las ventajas que supone este tipo de procesado se ha diseñado e implementado un software que, tomando un vector de frecuencias como entrada, produce una señal lista para su utilización en este u otros sistemas de requisitos semejantes.

4.5.- Generador de señal acústica multifrecuencial optimizada

Generar una señal que pueda ser empleada en el contexto que tratamos requiere cumplir con algunos requisitos adicionales, además de la minimización de la amplitud que se está tratando como requisito principal.

En primer lugar, hay que considerar el principio y el final del fragmento de señal que se va a transmitir. Cuando se envía un pulso de un tono simple se hace con desfase nulo y por tanto la señal comienza en un valor nulo. En este caso, al enviar un fragmento de una señal compleja el primer valor que tome la señal podría ser muy grande en valor absoluto.

Si esto sucede, lo que en el tiempo se percibirían como flancos de subida o bajada, en frecuencia supondrían una contaminación de amplio espectro que modificarían los valores de energía en cada una de las frecuencias elegidas para el análisis de forma aleatoria, perjudicando así los datos obtenidos.

Se debe, por tanto, garantizar que la transición del principio y final del pulso sean tan suaves como sea posible. Si bien se puede considerar un leve margen, lo ideal es forzar que los valores inicial y final del pulso enviado sean nulos.

En el diseño del programa generador de señal se tiene en cuenta este requisito, y se satisface seleccionando un fragmento de señal apropiado. Como se calculó anteriormente, la duración máxima del pulso a enviar es inferior a los 10 ms, por lo que,

si generamos una señal de aproximadamente un segundo, existe una alta probabilidad de encontrar un fragmento de la misma que cumpla este requisito.

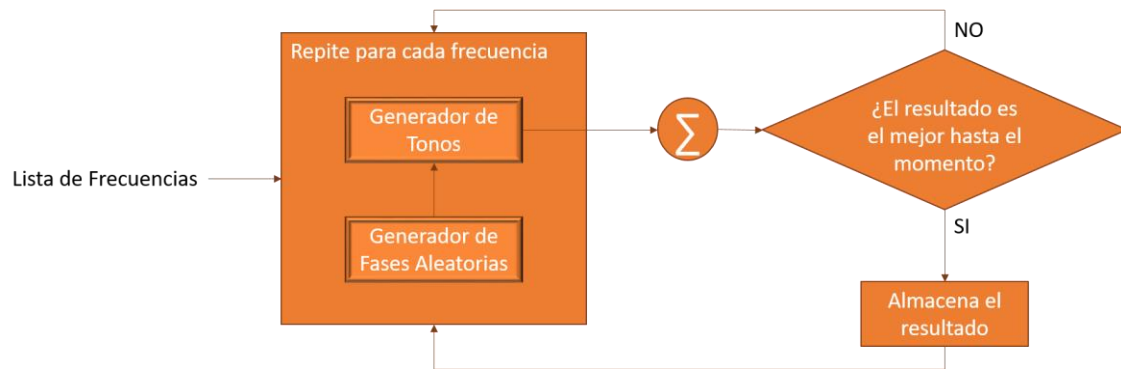
Mediante un simple algoritmo de búsqueda sobre la señal optimizada con el que se busquen dos muestras nulas (o inferiores a un valor umbral que se considere suficientemente bajo para considerar la contaminación frecuencial despreciable) separadas entre sí la duración del pulso deseado, se obtiene un fragmento que cumpla esta característica.

Por otra parte, se debe tener en cuenta que el fragmento de señal que se va a transmitir tendrá una duración muy inferior a su periodo, por lo tanto, nada garantiza que su media sea nula (como es lo normal en cualquier señal acústica). Esto, desde el punto de vista del sistema del amplificador y el altavoz, es equivalente a introducir una señal acústica corriente de media nula a la vez que un nivel de continua. Dichos sistemas pueden verse seriamente dañados si reciben un nivel de continua, por lo que la eliminación de esa componente continua de la señal es esencial para preservar la integridad del sistema.

En el programa que se expone a continuación se incluye una funcionalidad adicional que permite generar dos señales independientes dado un grupo de frecuencias, agrupando en una de las señales las frecuencias bajas y en otra las altas. Este procedimiento se ve motivado por dos razones diferentes:

Por una parte, se ha comprobado de forma experimental que el factor de minimización de la amplitud que se logra es mayor cuanto menor sea el rango de frecuencias a optimizar. Esto es lógico ya que la optimización se basa en hacer coincidir las crestas positivas de unos tonos con las negativas de otros, con lo que se tendrán mejores resultados cuanto más próximas estén las frecuencias entre sí.

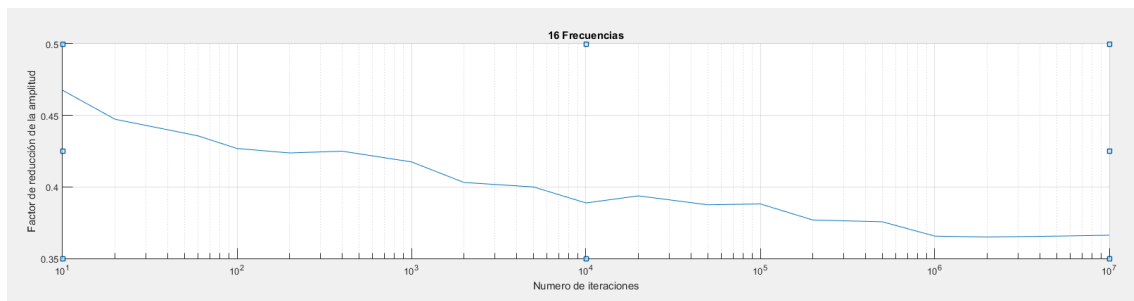
Por otra parte, estas dos señales podrían ser enviadas a través de canales distintos para ser emitidas a través de diferentes vías de un altavoz, cada una de las cuales presenta mejor respuesta ante un rango frecuencial determinado, o incluso a través de diferentes altavoces.



*Implementación en la Figura 13 del Anexo

Tratándose de una optimización basada en un método de Montecarlo, cuantos más experimentos se realicen más se aproximará el vector de desfases obtenido al vector de desfases ideal que optimice al máximo la señal.

Utilizando este software en la optimización una señal de 16 frecuencias (uniformemente repartidas entre los 3kHz y los 20 kHz), obtenemos los siguientes resultados en función del número de experimentos realizados:



Aunque, como se ha dicho, no es posible afirmar que se ha alcanzado un resultado óptimo, se puede observar como el factor de reducción tiende asintóticamente hacia un valor concreto. Se observa además como por encima de un millón de experimentos la mejora, de haberla, es inapreciable, y por tanto no merece la pena incrementar el tiempo de procesado por encima de ese punto. Esta cifra varía en función del número de frecuencias.

5.- Calibración del sistema

En este sistema se incluyen funcionalidades que permiten la calibración del mismo. La necesidad de calibrar el sistema radica en que la respuesta frecuencial del altavoz empleado y de los micrófonos del array no es plana, y varía ligeramente de un micrófono a otro.

La respuesta frecuencial del altavoz es compensable durante la emisión, si se logra caracterizar adecuadamente. Para esto es necesario un micrófono de referencia con respuesta frecuencial plana.

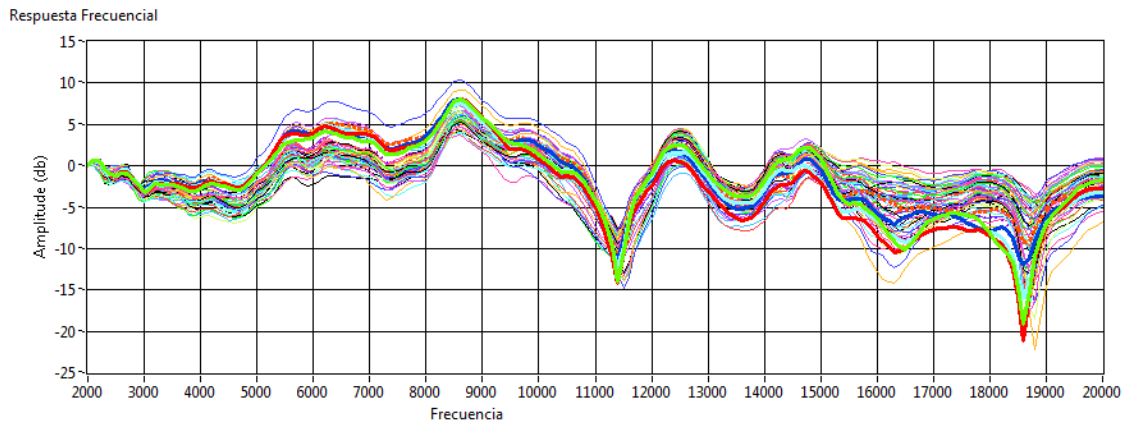
También es posible realizar una caracterización conjunta de la respuesta frecuencial del sistema que permita obtener una matriz de coeficientes que normalice la energía recibida para cada micrófono a cada una de las frecuencias utilizadas.

Para realizar este procedimiento de calibración se sitúa el altavoz en la posición en la que se encontrará el objeto a caracterizar, con lo que la señal recibida por el array no será una señal reflejada sino la onda directa desde el altavoz.

Con esta configuración se envía un tono puro de frecuencia variable barriendo todo el rango frecuencial que se desea caracterizar. Tras esto, se plantean múltiples opciones para realizar la calibración.

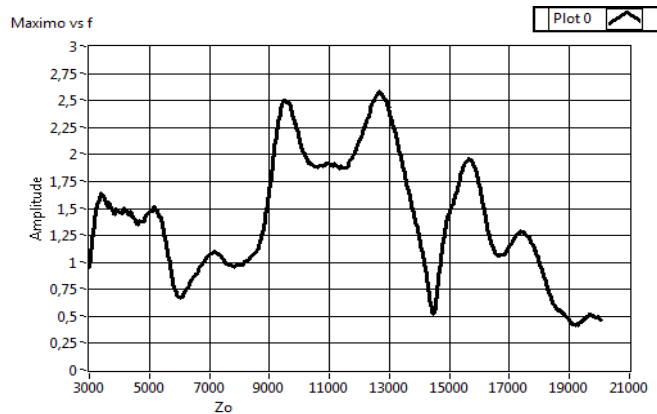
5.1.- Compensación de la respuesta frecuencial de cada micrófono

Mediante el sistema de captura se obtiene la energía que cada micrófono ha recibido para cada frecuencia. Tras repetir el experimento y promediar las respuestas obtenidas para eliminar dentro de lo posible la influencia del ruido, se obtiene la respuesta frecuencial conjunta del cada micrófono con el altavoz.



Con esta información se elabora una matriz de coeficientes que normaliza las respuestas frecuenciales, haciendo que cada micrófono reciba la misma intensidad de señal para todas las frecuencias, y la misma que los demás micrófonos. Dicha matriz de normalización se introduce tras realizar el filtrado de la señal en recepción.

5.2.- Compensación de la respuesta frecuencial del altavoz



Si se opta por realizar una calibración exclusivamente del altavoz, se requiere caracterizar el altavoz de forma independiente de los micrófonos mediante un micrófono de referencia.

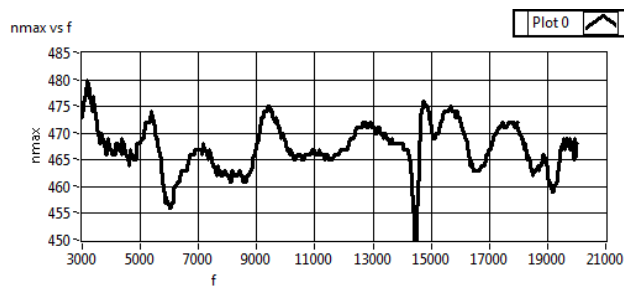
Una vez obtenida la respuesta frecuencial del altavoz se genera un vector de coeficientes que puede ser aplicado en el momento de generar la señal multitono, modificando la contribución de cada frecuencia que la compone.

El segundo procedimiento tiene la ventaja de que la señal acústica que se genera por el altavoz se corresponde mejor con la que se pretendía enviar. Además, dado que el

procedimiento se realiza durante la generación de la señal multitono, no conlleva ningún coste computacional con cada captura.

El primer procedimiento permite compensar los fallos de cada uno de los micrófonos por separado, pero conlleva un coste computacional para cada captura.

5.3.- Compensación de la respuesta conjunta de altavoz y array



Es posible realizar un procedimiento intermedio, en el que se caracterizan de forma conjunta el altavoz y los micrófonos, promediando la respuesta de todos los micrófonos

del array.

Este procedimiento genera un vector de coeficientes que puede ser aplicado en emisión. Dado que la respuesta frecuencial de cada micrófono es semejante a la de los demás, este procedimiento logra compensar la respuesta de cada uno de los micrófonos, pero no completamente.

6.- Representación de la imagen tridimensional multifrecuencial

El último paso en desarrollo de este sistema es el diseño e implementación de un método efectivo de representación de las imágenes tridimensionales multifrecuenciales obtenidas mediante el sistema de captura.

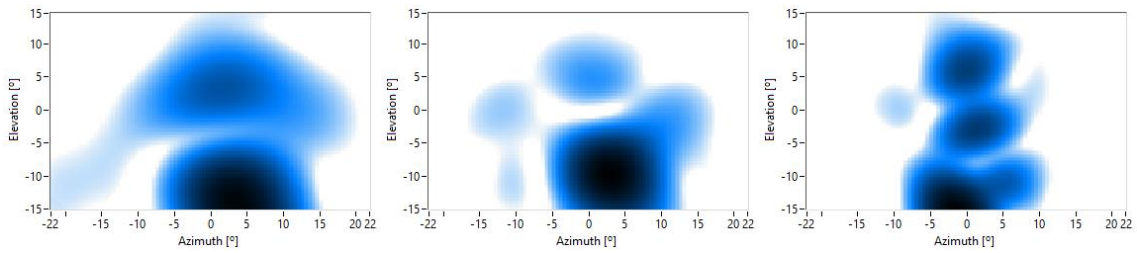
Hay que tener presente que el objetivo principal de este módulo es proporcionar al operador información suficiente que le permita controlar que el proceso de captura se está realizando correctamente.

Este módulo tiene la particularidad de seguir en ejecución una vez iniciado, hasta que el usuario desee iniciar una nueva captura o detenga el programa por completo. Esto es debido al método empleado para que la gráfica mostrada se actualice adecuadamente cuando el usuario selecciona otra frecuencia u otro corte.

El sistema de representación elegido para realizar las capturas es una gráfica en dos dimensiones, en el que se fijan una frecuencia y una distancia mediante dos controles, representando las dos dimensiones restantes a las que llamaremos plano XY. (Implementación en la Figura 14 del Anexo)

De este modo, el operador del sistema puede modificar la frecuencia y la distancia que está observando.

Este módulo tiene la particularidad de seguir en ejecución una vez iniciado, hasta que el usuario desee iniciar una nueva captura o detenga el programa por completo. Esto es debido al método empleado para que la gráfica mostrada se actualice adecuadamente cuando el usuario selecciona otra frecuencia u otro corte. Este tipo de visualización es muy útil para comprobar si el objeto a caracterizar está siendo captado por el sistema o es necesario realizar algún reajuste. Además, se puede apreciar con claridad cuando aparecen *Grating Lobes*.



*Ejemplo de captura realizada a tres frecuencias (de más baja a más alta), a distancia constante

Durante la realización de capturas, se ha demostrado muy práctico tener una gráfica adicional, en la que se muestra en que corte se encuentra máximo del plano XY para la frecuencia seleccionada.

Esta representación es útil para que el operador observe la estimación de la distancia al blanco que se está realizando en el sistema en función de la frecuencia. De esta forma puede verificar que dicha estimación es consistente al coincidir la realizada para las diferentes frecuencias, y por otra parte puede reajustar la distancia de apuntamiento si fuese necesario.

Una posible futura implementación es un sistema de representación tridimensional en el que se alterne entre frecuencias. Para realizar dicha implementación es necesario definir una superficie de nivel a un cierto umbral para que no se incluyan datos del espacio vacío que impidan ver el sólido que se quiere representar.

En este caso no se ha elegido este sistema de representación porque, además de suponer una carga computacional mayor, los datos de los que hay que prescindir para que dicha representación sea viable aportan información esencial al operador, como la aparición de *grating lobes*, la presencia de fuentes de ruido o la presencia de otro objeto que produzca reflexiones indeseadas.

7.- Conclusiones

Durante el diseño e implementación del software para realizar capturas tridimensionales multifrecuenciales para su uso en un sistema de caracterización biométrica se han extraído las siguientes conclusiones:

- Utilizando el rango sónico de frecuencias se han obtenido los resultados esperados.
 - Si se emplease un rango subsónico se perdería la resolución necesaria para la biometría.
 - Para emplear un rango ultrasónico es necesario solucionar la aparición de *grating lobes*, adecuando las dimensiones del array.
- La combinación de la plataforma myRIO para gestionar entradas y salidas, analógicas y digitales, junto a un PC como elemento de computación y almacenamiento cumple con los objetivos previstos, permitiendo una velocidad suficiente en el flujo de datos, además de proporcionar cierta flexibilidad al hardware, permitiendo operar el dispositivo myRIO desde diferentes PC conectados de forma inalámbrica.
- La carga computacional derivada del procesado mediante FFT depende del número de sensores, mientras que la carga computacional de un procesado mediante retardos depende de en cuantas coordenadas se quiera realizar el apuntamiento.
 - Si se va a realizar un barrido del espacio con una resolución muy alta es más eficiente el procesado mediante FFT. Una resolución elevada solo es eficiente si se utilizan frecuencias altas para que el ancho del haz sea más estrecho.
 - El procesado mediante FFT permite una mayor precisión de apuntamiento ya que evita la discretización del espacio que supone el método de los retardos.
- El sistema utilizado para el diseño y la generación de la señal multifrecuencial ha demostrado ser efectivo en la optimización y su implementación en Labview es eficiente. Es viable utilizar este sistema de forma independiente para optimizar una combinación de frecuencias en otros ámbitos.

Anexo

En este anexo se incluyen las implementaciones, todas ellas funcionales, de los bloques funcionales explicados anteriormente:

Figura 0: Bloque principal del software de captura, ejecutado en el PC.

Figura 1: Bloque parte de la FPGA que implementa un reloj local y para el array.

Figura 2: Bloque parte de la FPGA que realiza la obtención de datos del array.

Figura 3: Bloque parte de la FPGA que realiza la emisión de la señal guardada.

Figura 4: Bloque parte de la FPGA que realiza la emisión de ruido blanco.

Figura 5: Bloque parte de la FPGA que se realiza la carga de datos en la *Lookup Table*.

Figura 6: Bloque principal del myRIO. Llama a los bloques de captura y envía.

Figura 7: Bloque encargado de dar la orden de enviar a la FPGA.

Figura 8: Bloque encargado de leer los datos de la FPGA.

Figura 9: Bloque del PC encargado de realizar la conformación.

Figura 10: Bloque del PC que realiza el apuntamiento en acimut y elevación.

Figura 11: Bloque del PC que realiza el apuntamiento en distancia.

Figura 12: Bloque que realiza el procesado FFT y la selección de frecuencias.

Figura 13: Optimizador de señal

Figura 14: Bloque que se encarga de generar la imagen acústica.

Figura 0: Bloque Principal del PC

MAIN

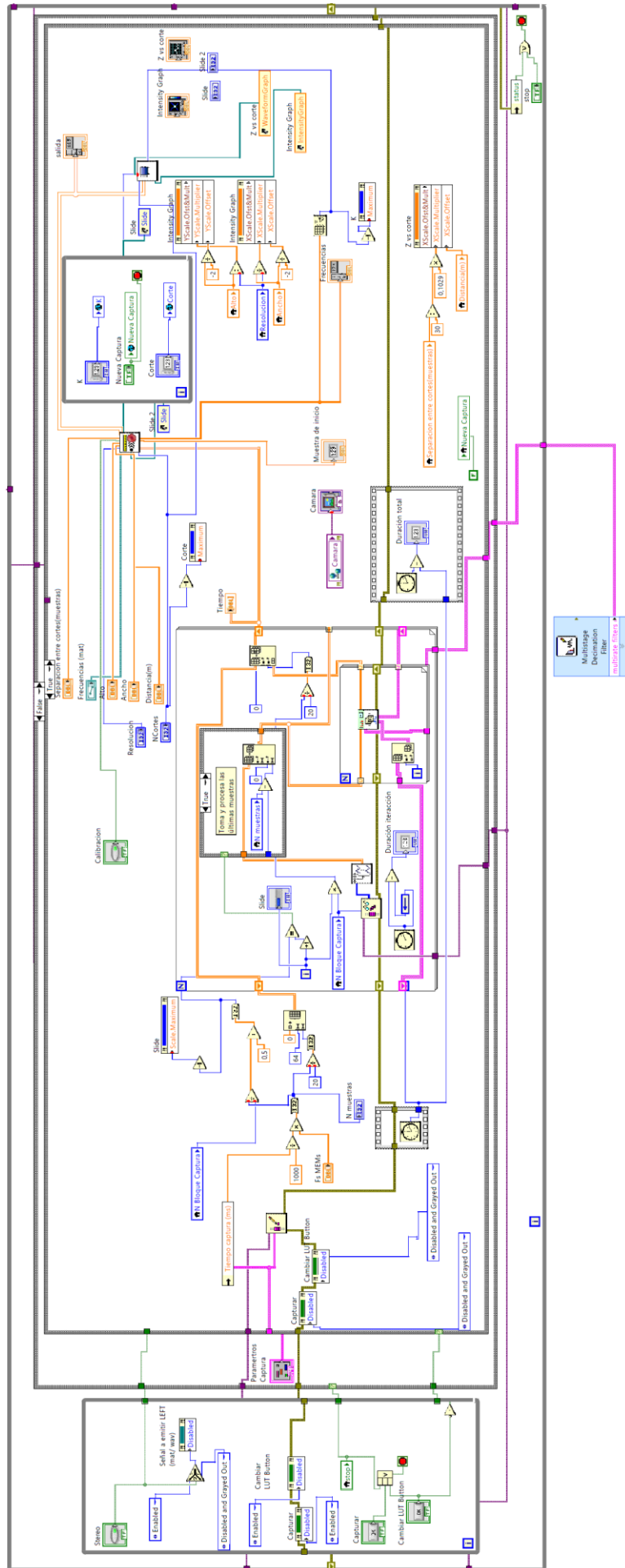


Figura 1: Reloj en FPGA

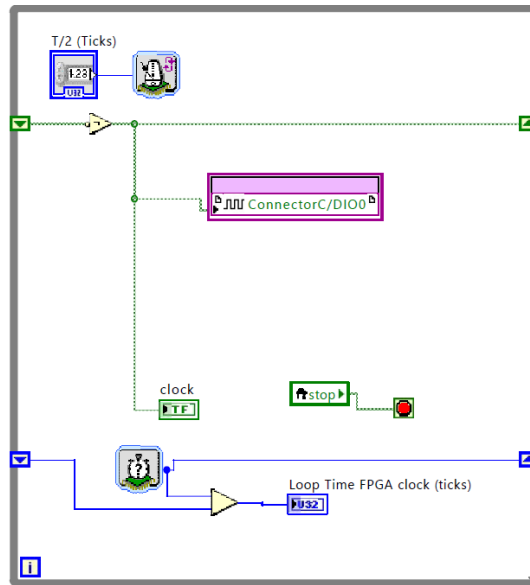


Figura 2: Captura en FPGA

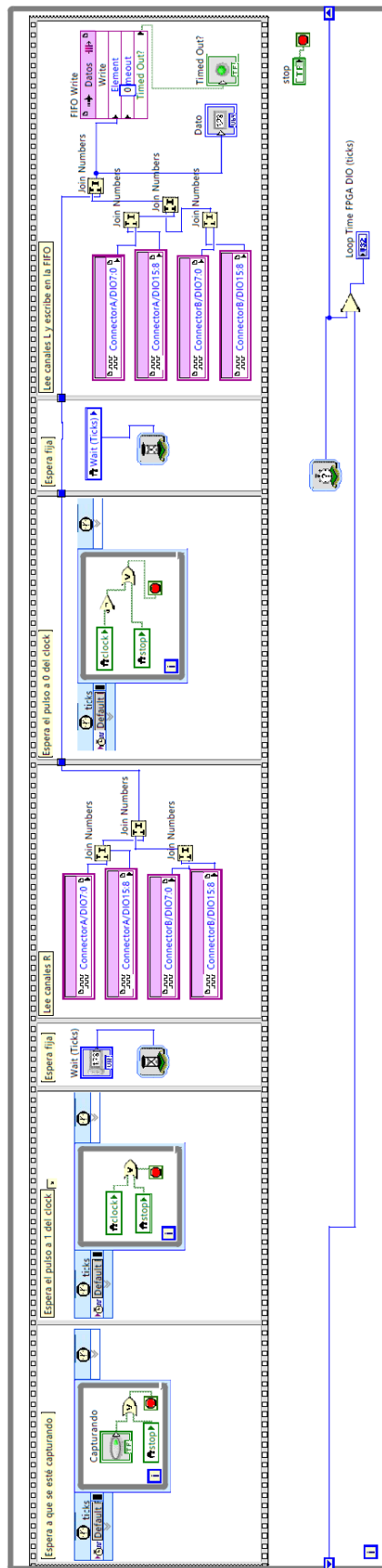


Figura 3: Emitir Señal en FPGA

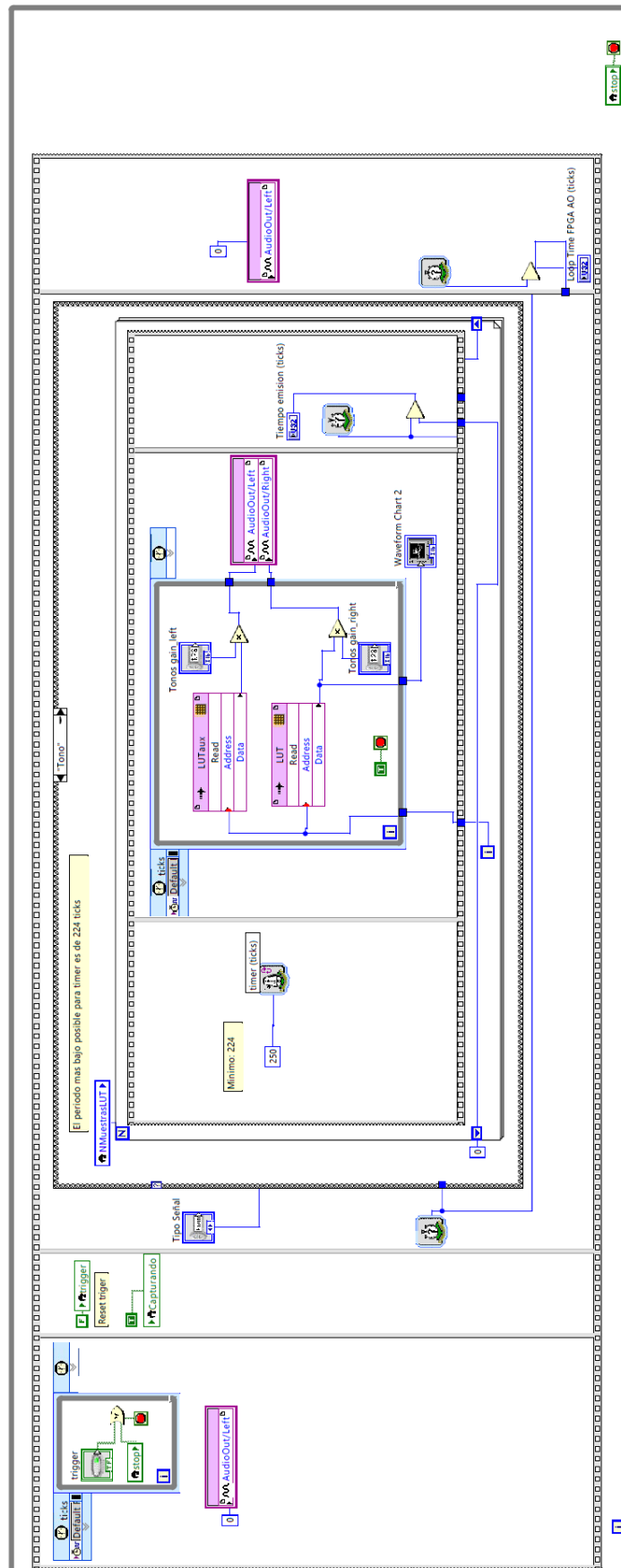


Figura 4: Emitir Ruido en FPGA

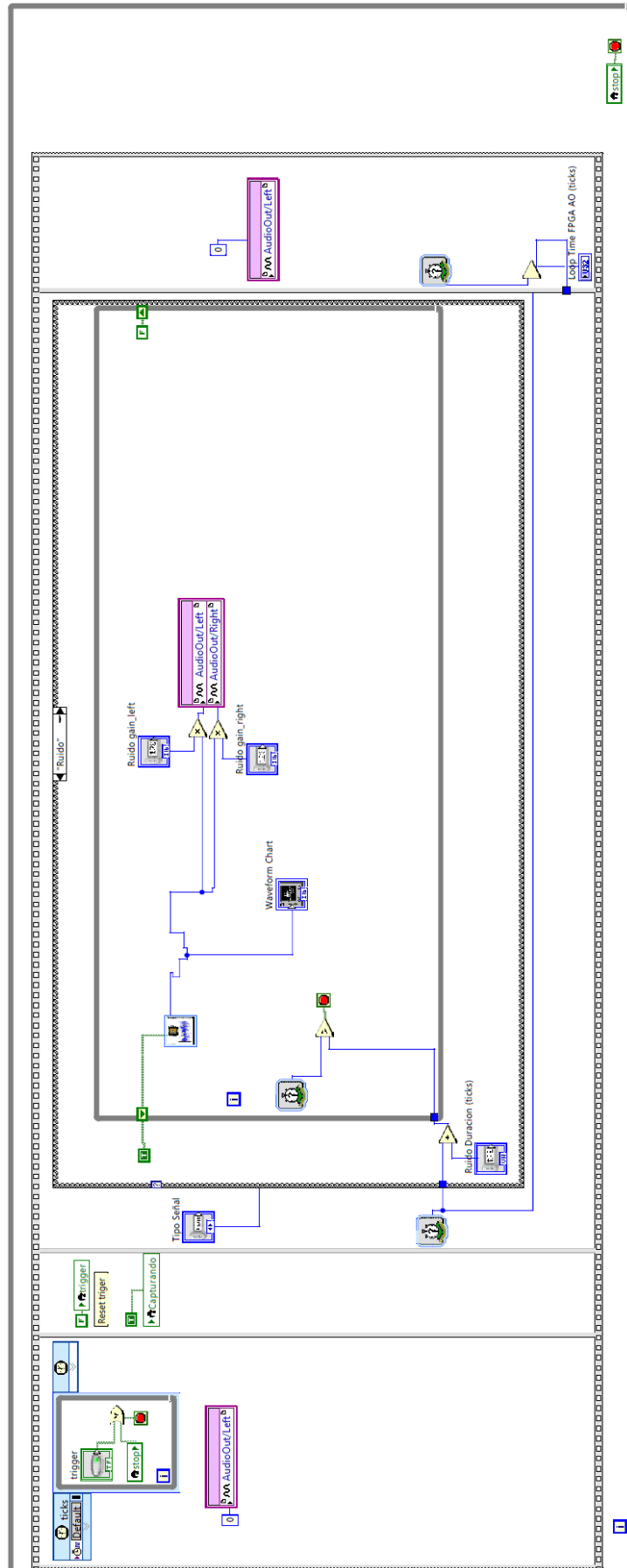


Figura 5: Bloque de Carga de Señal en FPGA

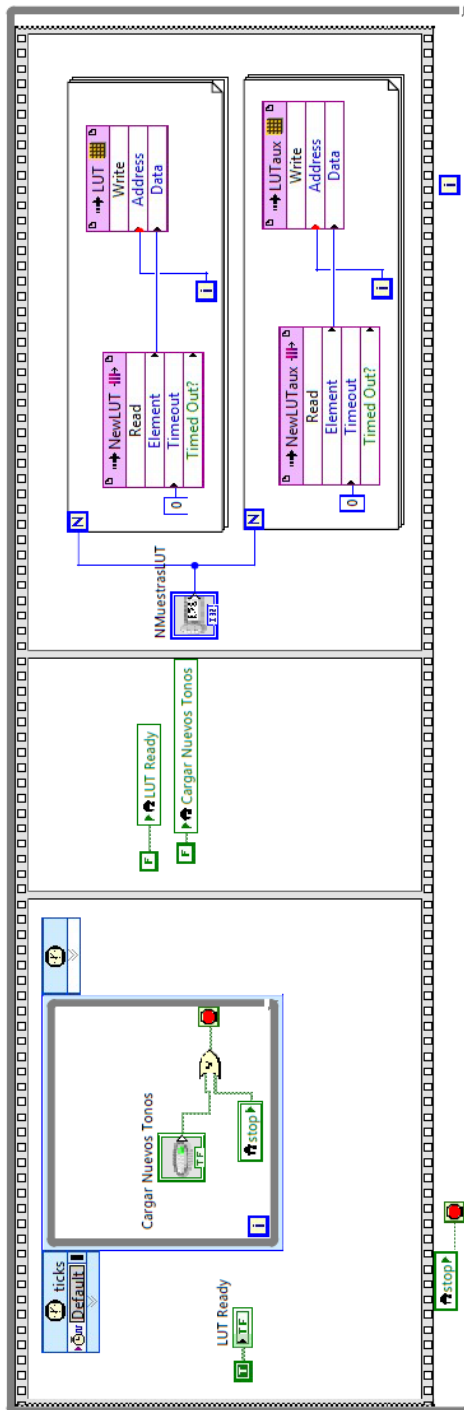


Figura 6: Bloque Captura y Envía en procesador myRIO

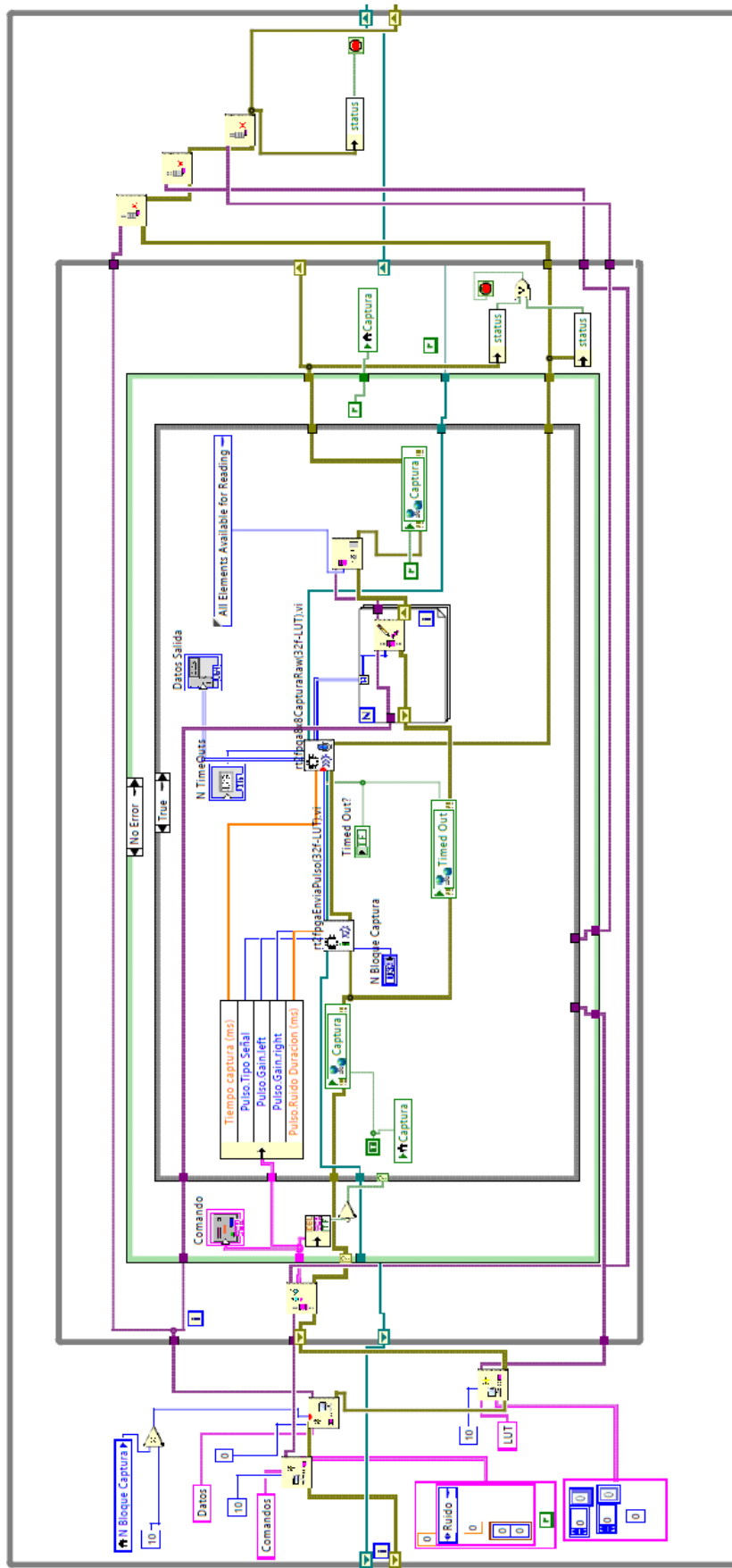


Figura 7: Bloque Envía en myRIO

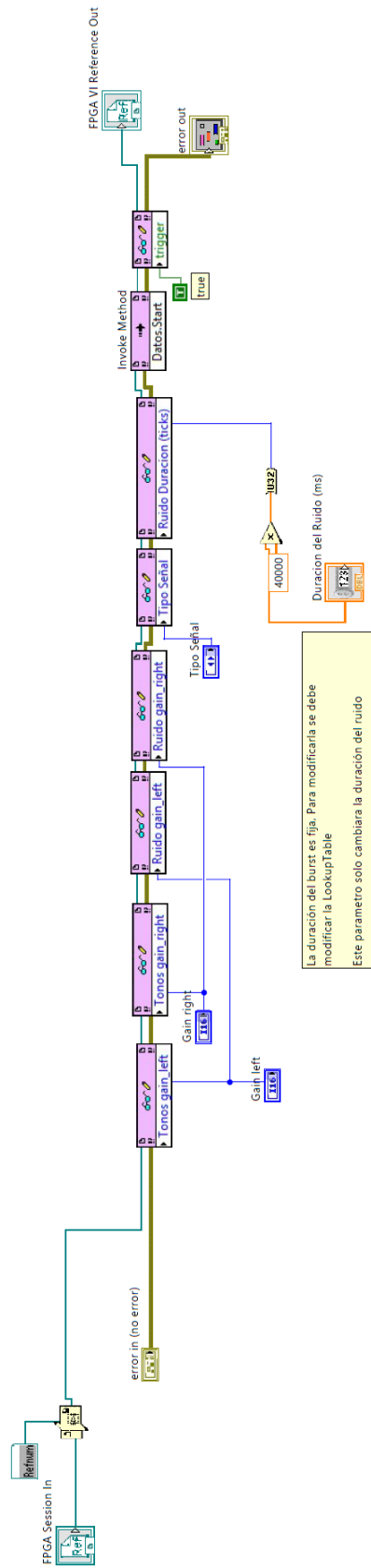
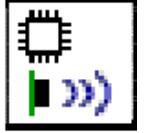


Figura 8: Bloque encargado de leer los datos de la FPGA en myRIO

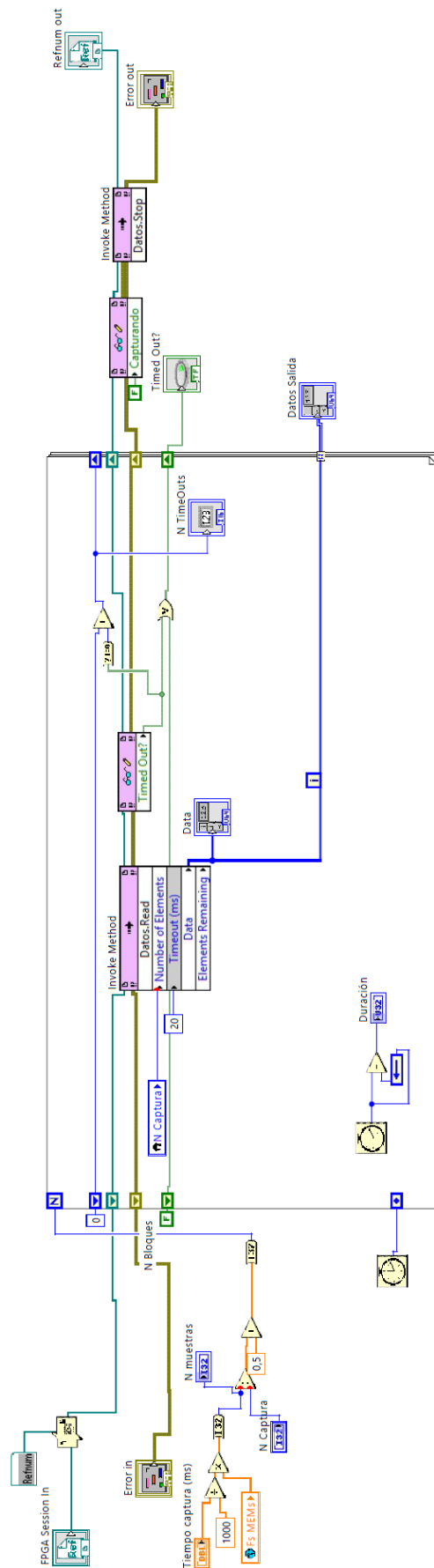


Figura 9: Bloque Conformador 3D en PC

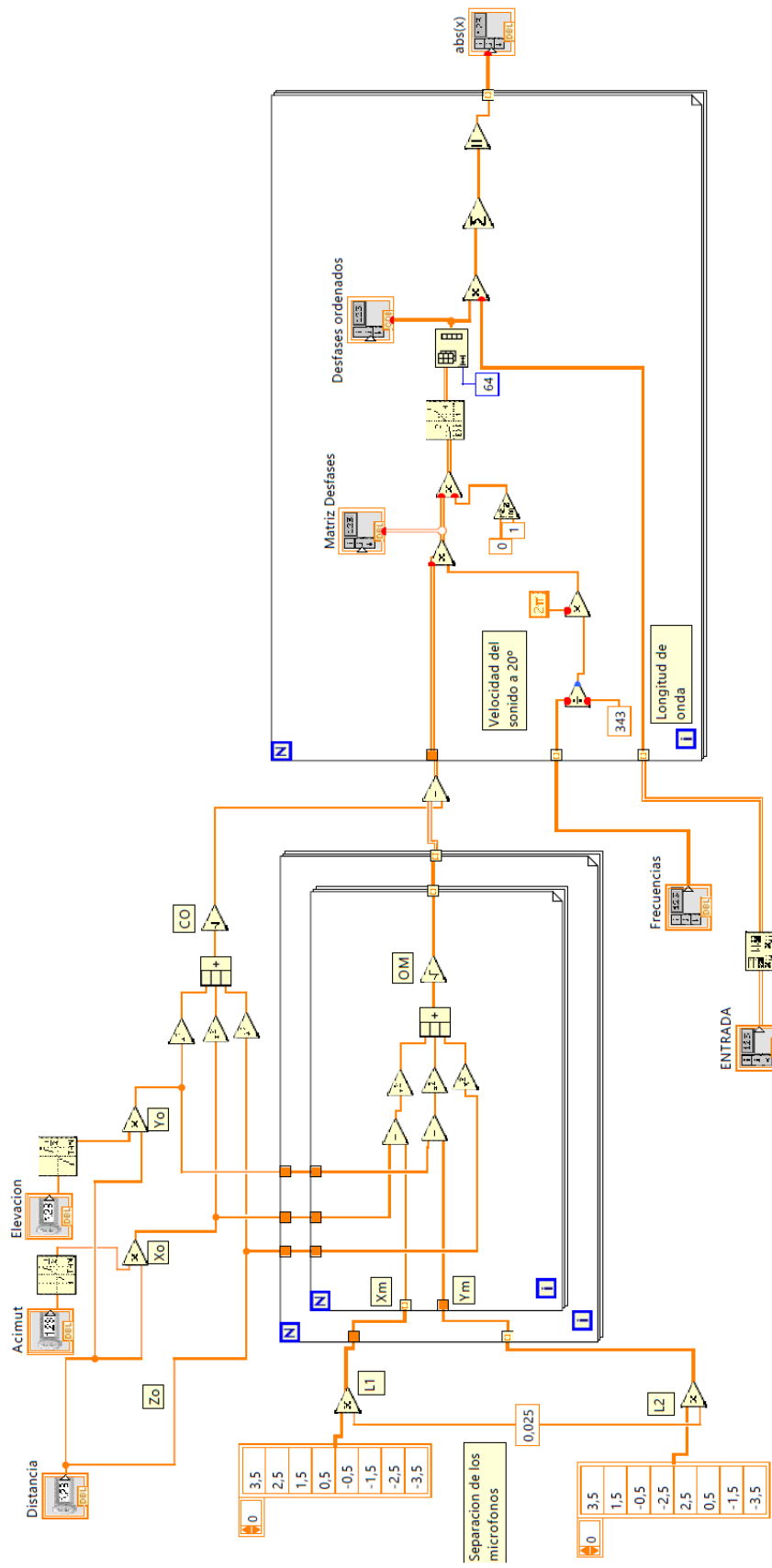
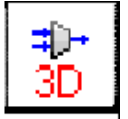


Figura 10: Bloque Apuntamiento 2D en PC

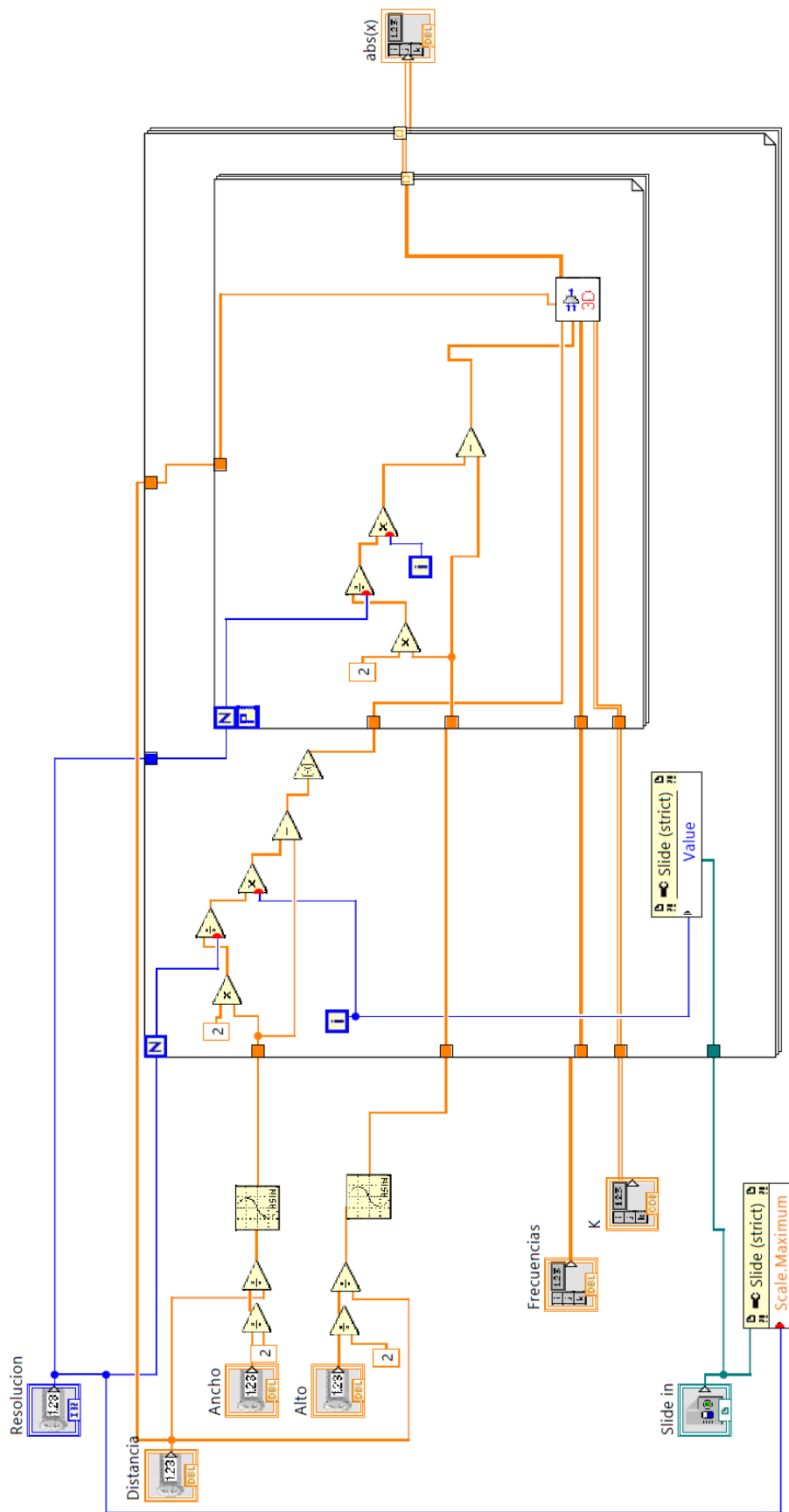


Figura 11: Bloque Apuntamiento Distancia en PC

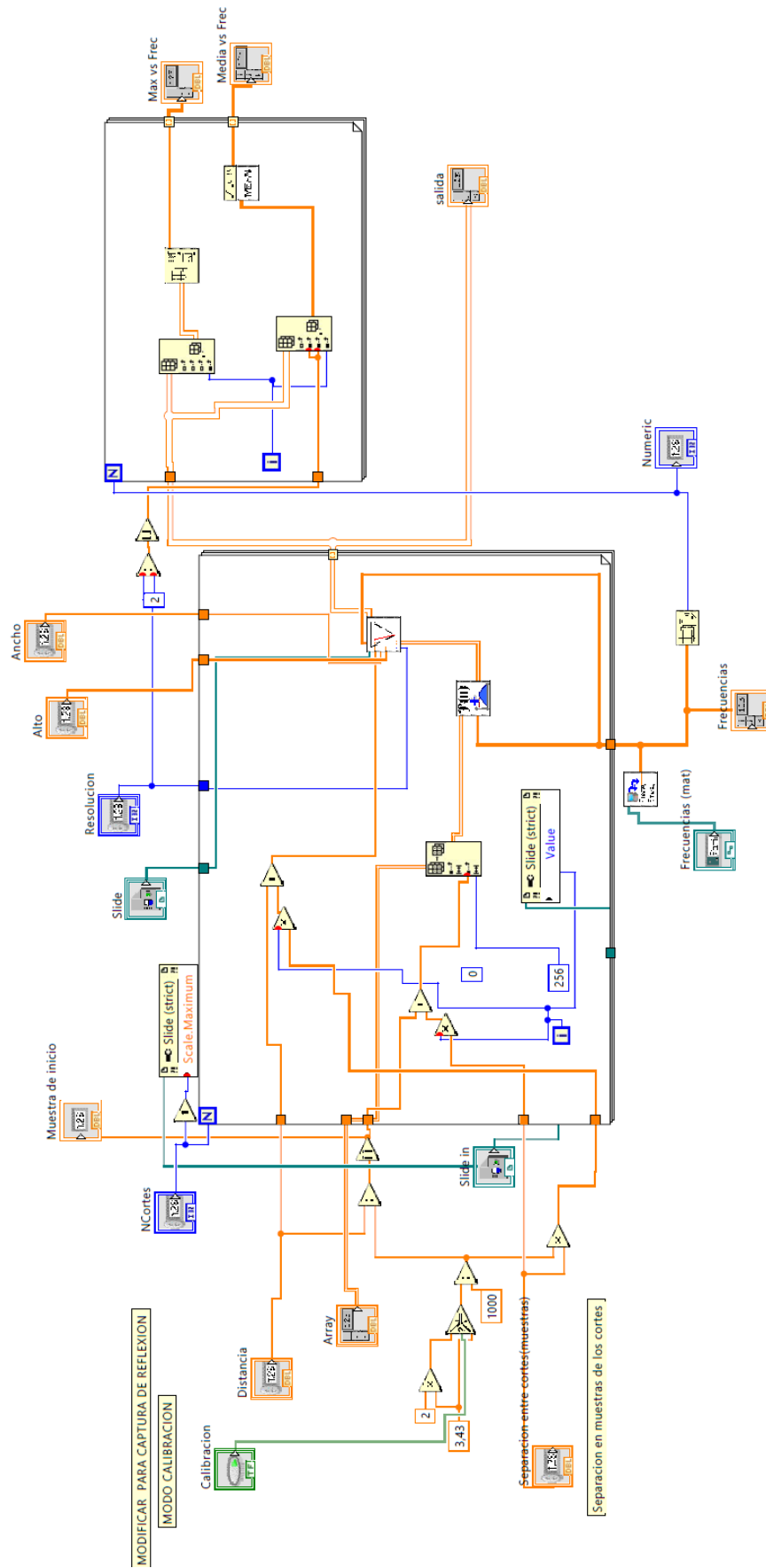


Figura 12: Bloque FFT y Selección de Frecuencias en PC

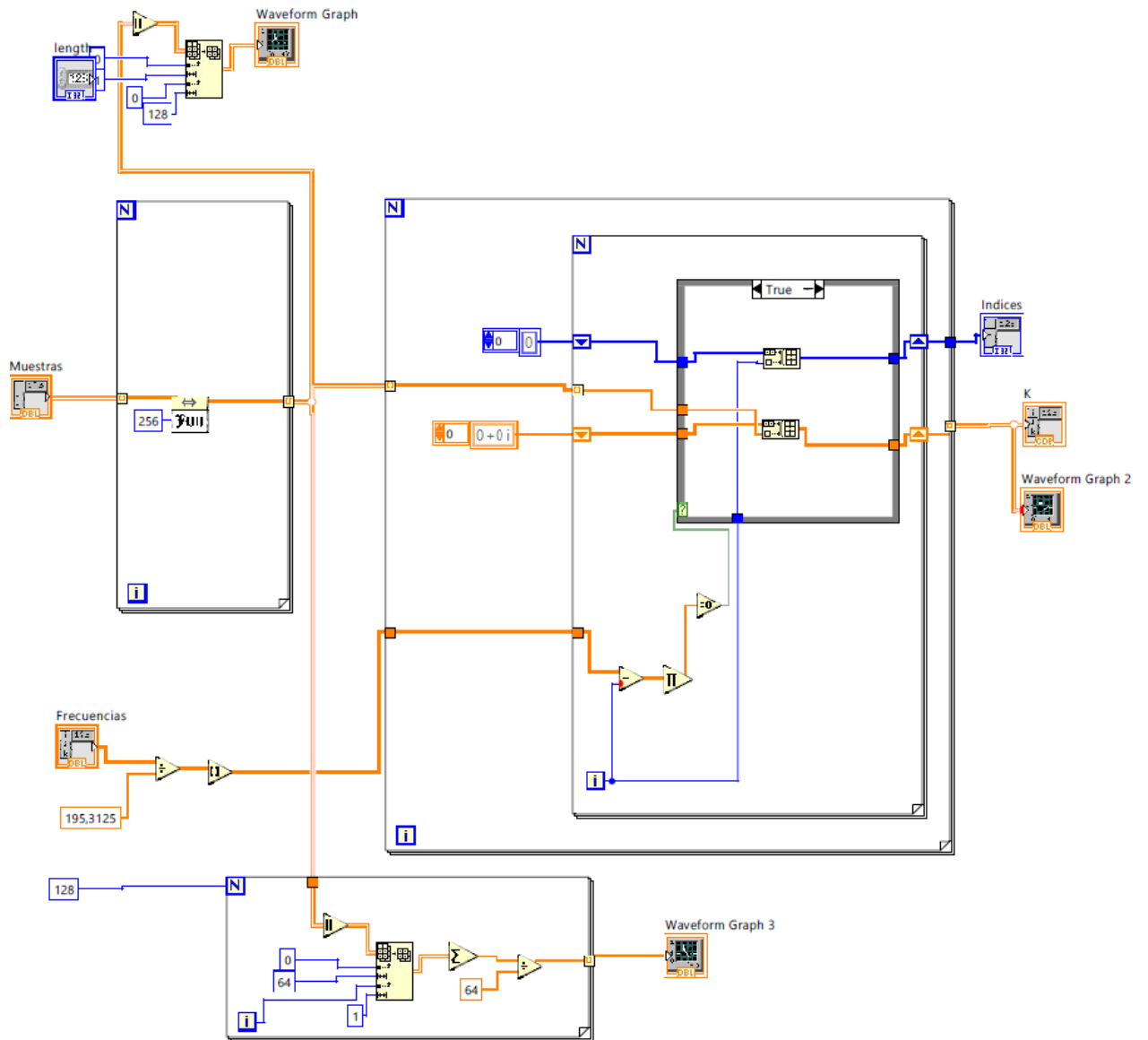


Figura 13: Optimizador de señal en PC

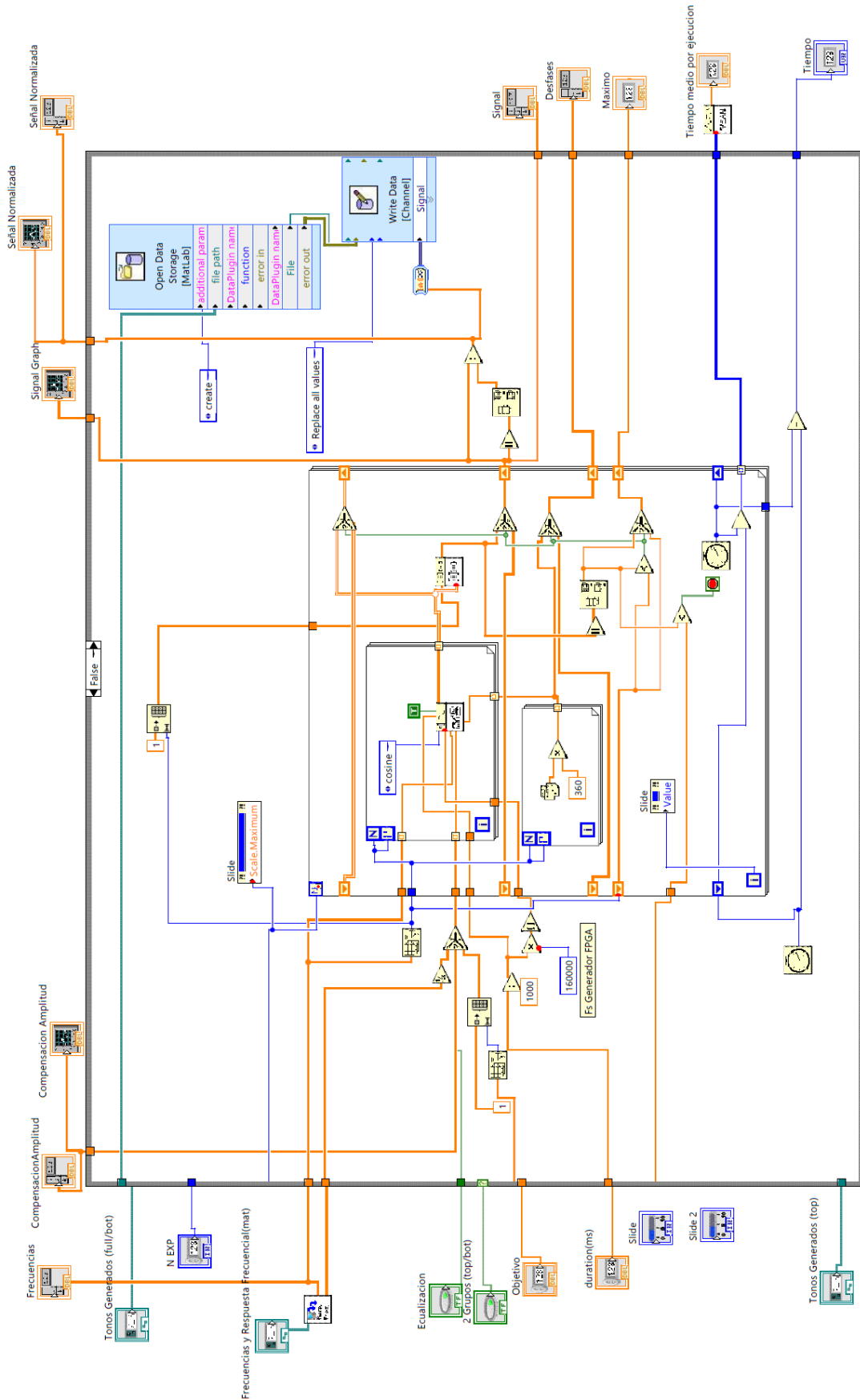


Figura 14: Bloque Imagen Acústica en PC

