Universidad de Buenos Aires
Facultad de Ingeniería

Señales y Sistemas (66.74)

Procesamiento de señales de un arreglo de micrófonos

Alumno:

Rolando, Marcos Daniel 102323 mrolando@fi.uba.ar

26 de julio de 2021

Índice

Introducción

El objetivo del siguiente trabajo práctico es utilizar distintos conceptos adquiridos a lo largo de la materia sobre el procesamiento y análisis de señales para lograr identificar la posición de una fuente emisora de sonido. Contamos con cinco micrófonos dispuestos secuencialmente entre sí a una distancia de 5 cm en una habitación de 3x4 metros. Los micrófonos se encuentran ubicados a partir de los 2 metros de largo y 1 metro de alto respecto al origen tal y como se indica en la siguiente figura.

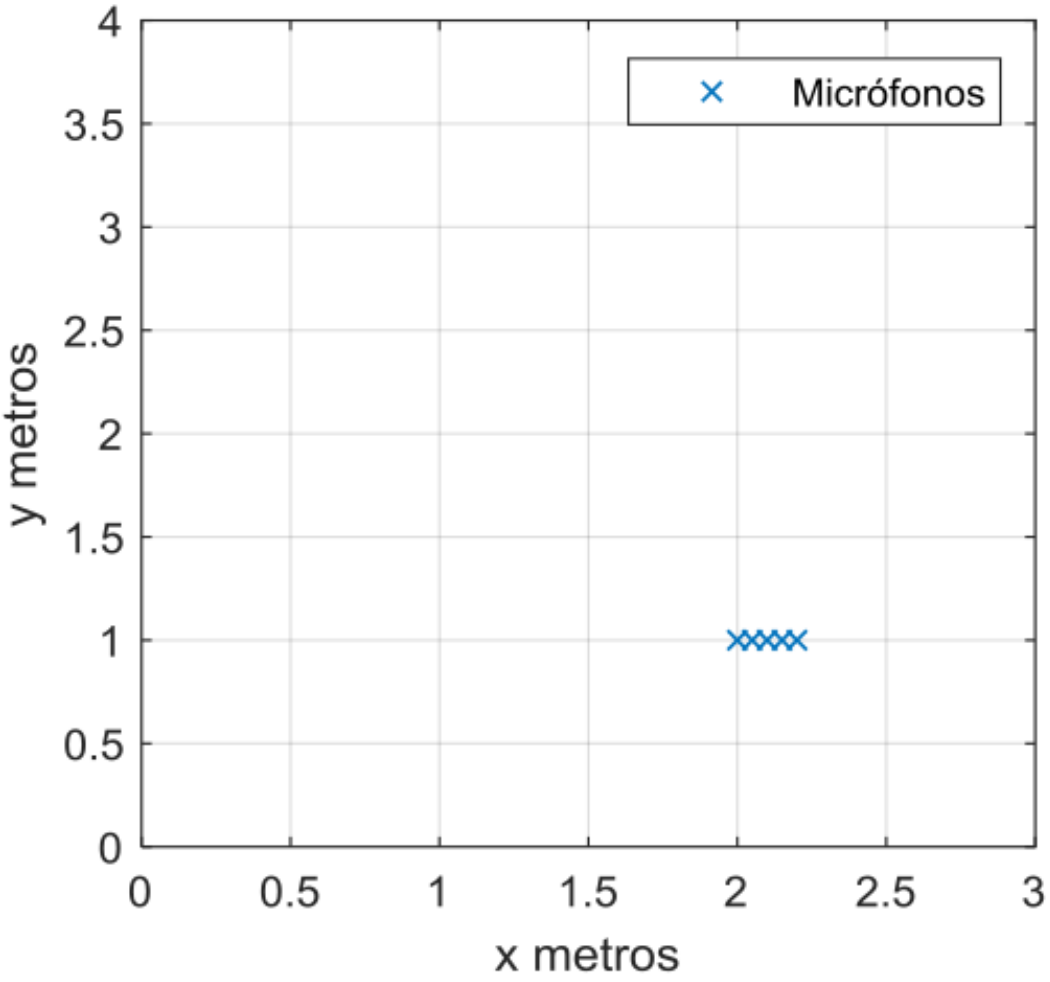

Figura 1: Disposición de los micrófonos en la habitación

Se deberá entonces calcular los retardos de la señal captada por los distintos micrófonos y utilizar estos valores para estimar, mediante aproximaciones del frente de onda sonoro y geometría, la posición de la fuente en la habitación.

Ejercicio 1

En el archivo audios1.mat se guardaron las señales registradas por un arreglo lineal de 5 micrófonos separados 5 cm ubicados en una habitación de 3x4 metros, según el esquema de la Figura 4. Se utilizó una frecuencia de muestreo de 48 kHz. Graficar en forma superpuesta las señales registradas por todos los micrófonos. Etiquetar ejes y sumar leyenda. Realizar una segunda figura de un segmento donde se puedan identificar los desplazamientos de las señales. Analizar los espectros de las señales mediante transformada de Fourier y espectrogramas.

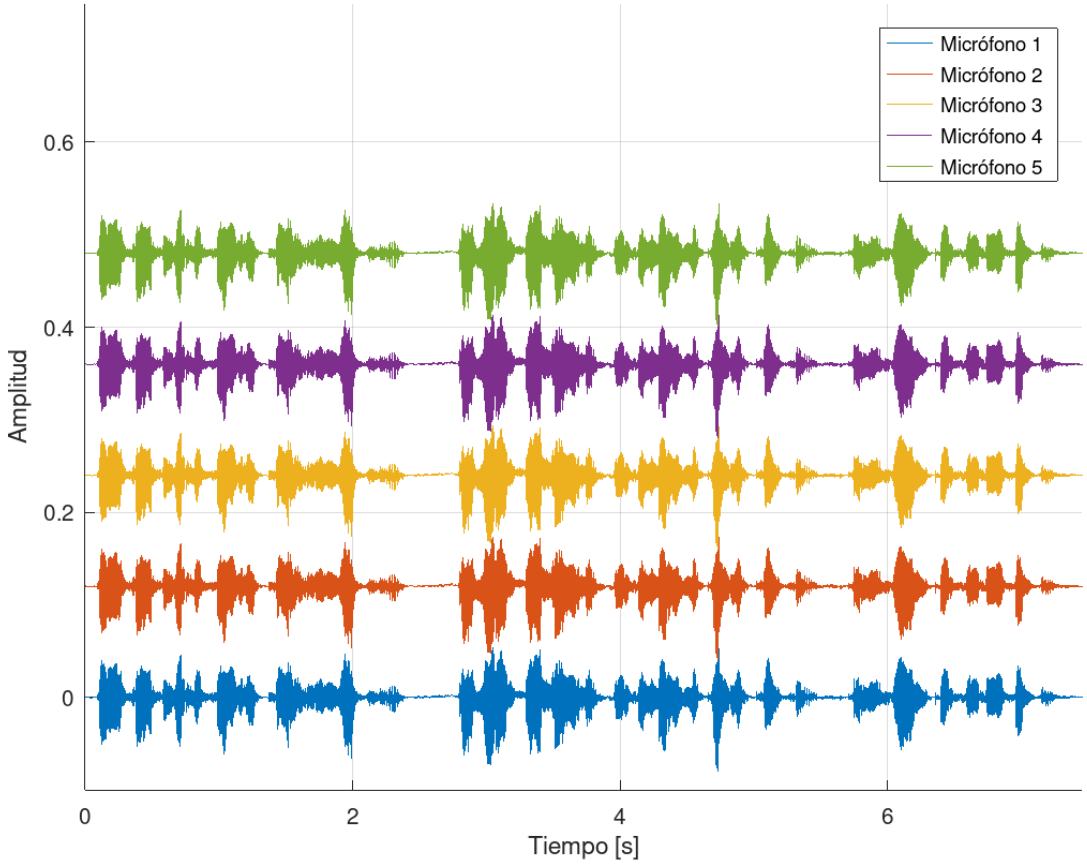

Figura 2: Señales registradas por los micrófonos

Para el gráfico de la **Figura 2** se agregó una continua a cada señal de amplitud (0.12 * (i-1)) donde **i** denota al micrófono i-ésimo, de forma tal que puedan distinguirse individualmente cada una. Podemos observar que a priori no se aprecian diferencias ni de retardo ni de amplitud entre ellas en la escala utilizada, por lo que a continuación nos restringiremos a únicamente un intervalo de las señales con el objetivo de que podamos identificar posibles diferencias.

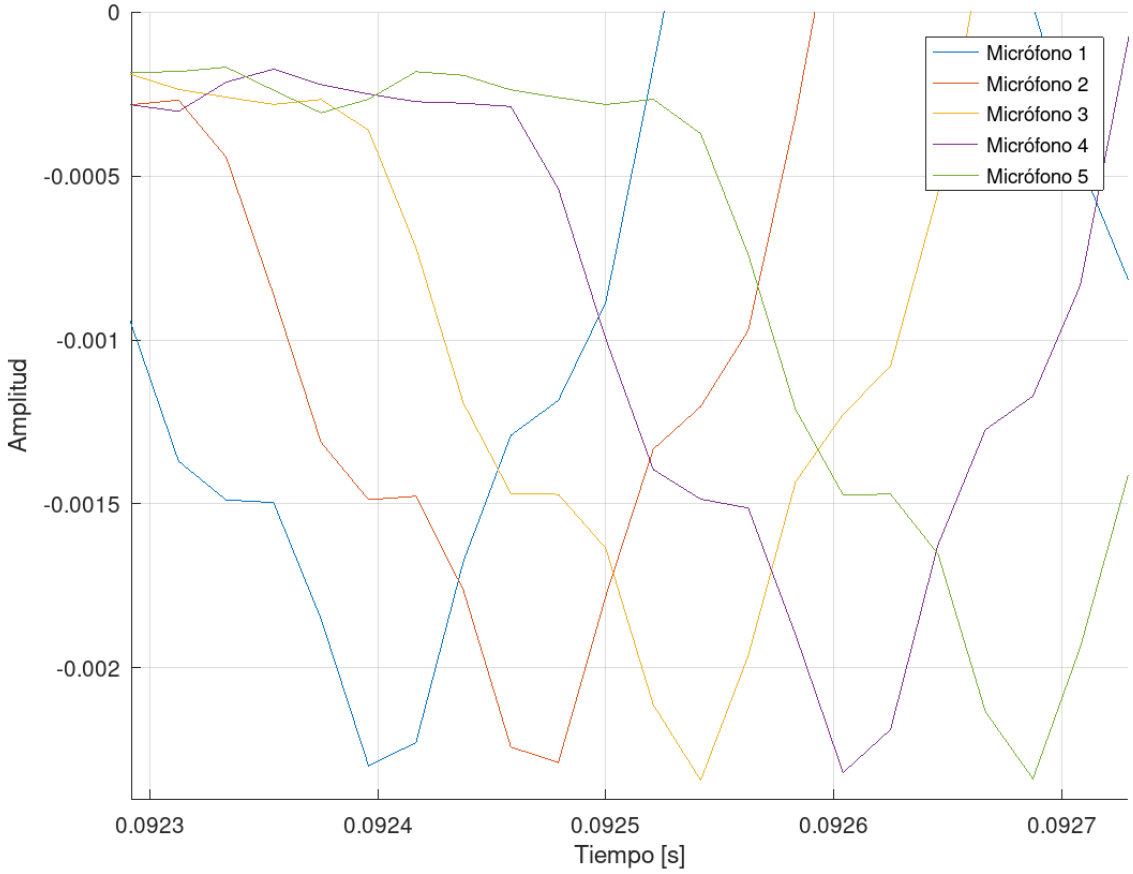

Figura 3: Señales registradas por los micrófonos entre los 92,3 **ms** y 92,7 **ms**

Enfocándonos en el intervalo de tiempo que abarca desde los 92,3 **ms** hasta los 92,7 **ms** resulta posible establecer diferencias entre las distintas señales. Si bien la amplitud es la misma se observa un claro retardo entre ellas, que además es aproximadamente el mismo entre dos señales consecutivas a simple vista.

Analicemos ahora el espectro de las señales. Nuevamente agregaremos un offset a cada espectro de las señales de amplitud (200 * (i-1)) donde **i** es el i-ésimo micrófono de manera que podamos apreciar cada gráfico sin superposición entre ellos.

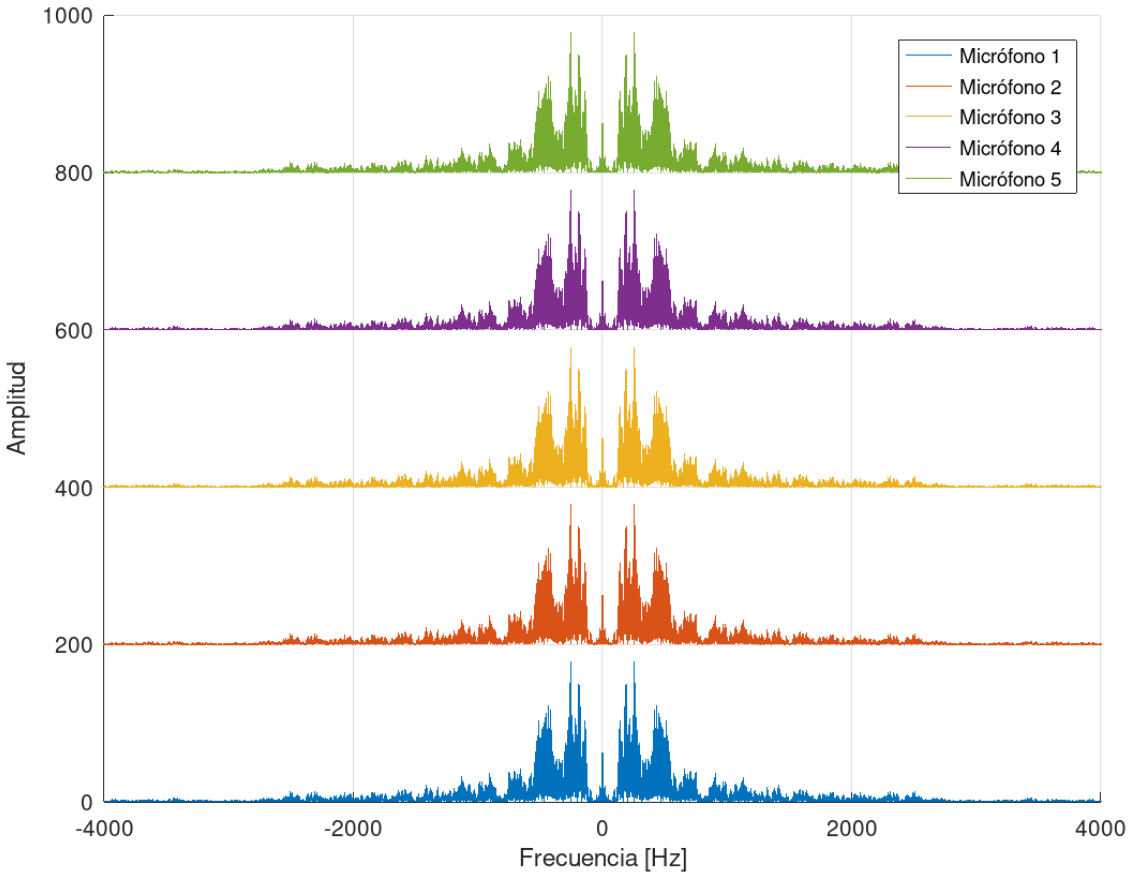

Figura 4: Magnitud del espectro de los micrófonos entre -4000 Hz y 4000 Hz

Nuevamente observamos gráficos que a simple vista son indistinguibles entre sí. Dado que habíamos establecido previamente que la única diferencia apreciable entre las distintas señales se debía a un retardo entre ellas es de esperar entonces que el análisis de las magnitudes del espectro resulte similar. Podemos observar que las señales contienen prácticamente la totalidad de sus frecuencias alrededor de los 4 KHz, por lo que en principio una estimación del ancho de banda de 4 KHz sería una buena aproximación.

Por último analicemos el espectrograma de las señales. Si somos coherentes con el análisis previo resulta evidente el hecho de que debido a que el retardo tiene un cota superior de 100 us (observable en la **Figura 3**) y dado que la frecuencia de muestreo **Fs** de las señales es de 48 KHz (20 us por muestra aproximadamente) estaríamos entonces teniendo un retardo entre señales de como máximo 5 muestras. Dado que 5 muestras es evidentemente muy poco para extraer información en frecuencias apreciable en el espectrograma nos limitaremos entonces a mostrar el espectrograma de uno sólo de los micrófonos (específicamente del primero, aunque es irrelevante). Se aclara que la escala del espectrograma va desde violeta hasta amarillo, siendo el amarillo el asociado a las frecuencias de mayor potencia de la señal y el violeta a las de menor.

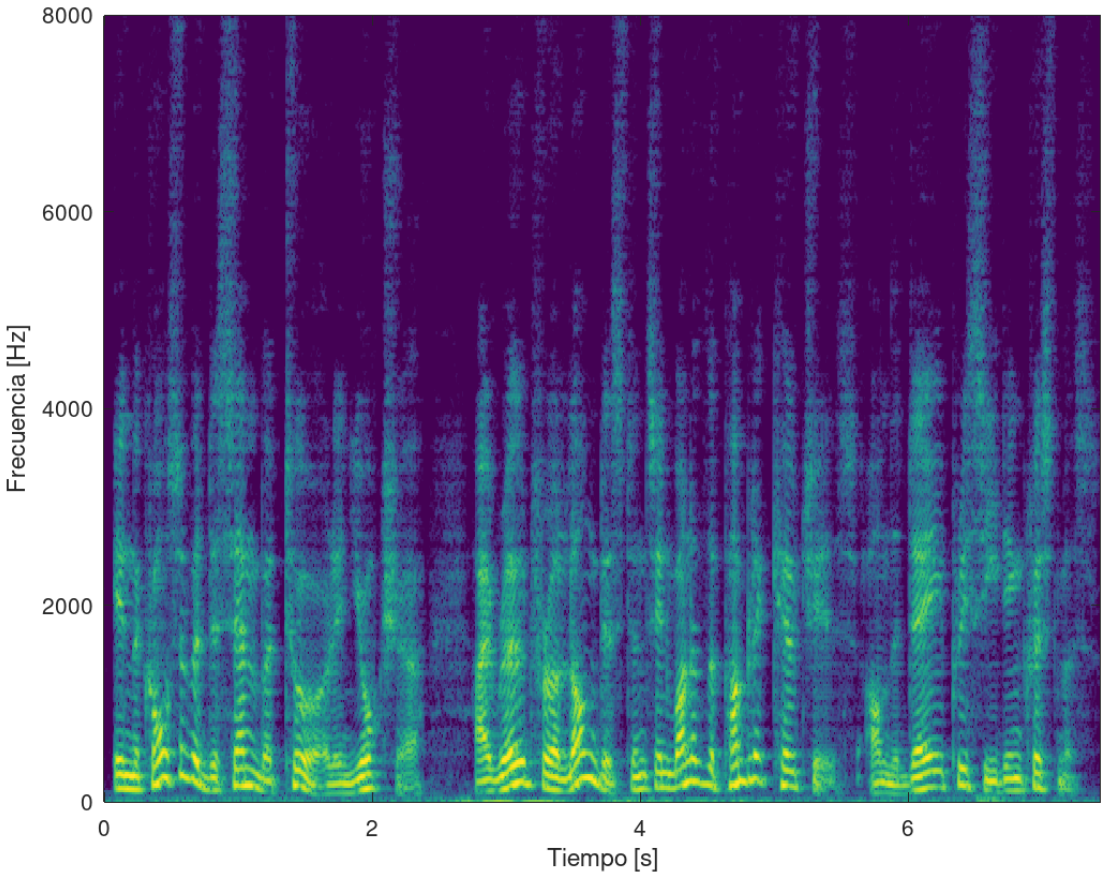

Figura 5: Espectrograma del Micrófono 1 entre los 0 Hz y 8000 Hz

Observando el espectrograma vemos que, al igual que lo que vimos en la espectro de las señales de la **Figura 4**, las frecuencias de nuestra señal se concentran desde los 0 hasta los 4/5 KHz. Más allá de ese rango vemos que casi todas las frecuencias están en la escala violeta, es decir, tienen un potencia despreciable. Podemos observar también unas curiosas rectas verticales que marcan frecuencias donde la potencia no es tan despreciable más allá de los 5 KHz, aunque probablemente se deban simplemente a ruido inherente de la señal. Las frecuencias que contienen más potencia se encuentran claramente por debajo de los 2000 Hz y pueden observarse lo que parecieran ser armónicos. Esto último tiene sentido al escuchar el audio que representa esta señal, el cual resulta ser un hombre hablando.

Ejercicio 2

Utilizando la figura del ejercicio anterior, estimar los retardos de llegada a todos los micrófonos. Suponer una posible posición de la fuente de sonido con respecto a los micrófonos a partir de los valores estimados de los retardos.

Debido a que establecimos que el espectrograma no nos provee información respecto al retardo nos basaremos entonces en el gráfico de la **Figura 3**, de manera que estimaremos a groso modo el retardo entre cada señal observando su comportamiento en el tiempo. En principio resulta intuitivo intentar medir los retardos con los picos inferiores, sin embargo se da el caso de que el primer par de micrófonos presentan dos picos. Dado que los tres micrófonos restantes presentan uno solo podríamos caer en un error de una muestra sencillamente por la elección de alguno de los dos picos iniciales. Una opción "más segura" para tener menos error podría ser entonces medir el retardo en base a los puntos con amplitud **-0.0015**, el cual es el punto inmediatamente anterior a una caída en amplitud de las señales y que es claramente reconocible en todas. Vemos que la escala avanza de a 100 us y que la diferencia entre las señales aparenta encontrarse en aproximadamente la mitad de la escala. Dado que la diferencia entre cada muestra es de unos 20 us (debido a la frecuencia de muestreo utilizada) entonces estaríamos observando un retardo de unas 2 o 3 muestras, es decir, unos 40 us a 60 us a simple vista. Midiendo más en detalle en el gráfico de Octave resulta más cerca de 60 us por lo que tomaremos este valor como medición entre las señales.

Respecto a la posición de la fuente, es inmediato que con esta resolución de tiempo entre muestras resultará imposible obtener una posición. Debido a la resolución que tenemos estamos obteniendo que los retardos entre cualquier par de micrófonos consecutivos es de unos 60 us. Esto implicaría que el frente de onda del sonido demora lo mismo en llegar de un micrófono al siguiente, lo que sumado al hecho de que los micrófonos se encuentran equiespaciados entre sí resultaría en un frente de onda plano. Que el frente de onda de una fuente de sonido sea plano surge de considerar que dicha fuente se encuentra a una distancia infinita de los micrófonos que captan la señal, por lo que establecer la posición de la fuente exacta resulta imposible. Lo que sí podremos estimar, sin embargo, será la dirección de la que proviene el sonido emitido por la fuente. Si hacemos uso de las ecuaciones provistas por el enunciado obtenemos los siguientes resultados.

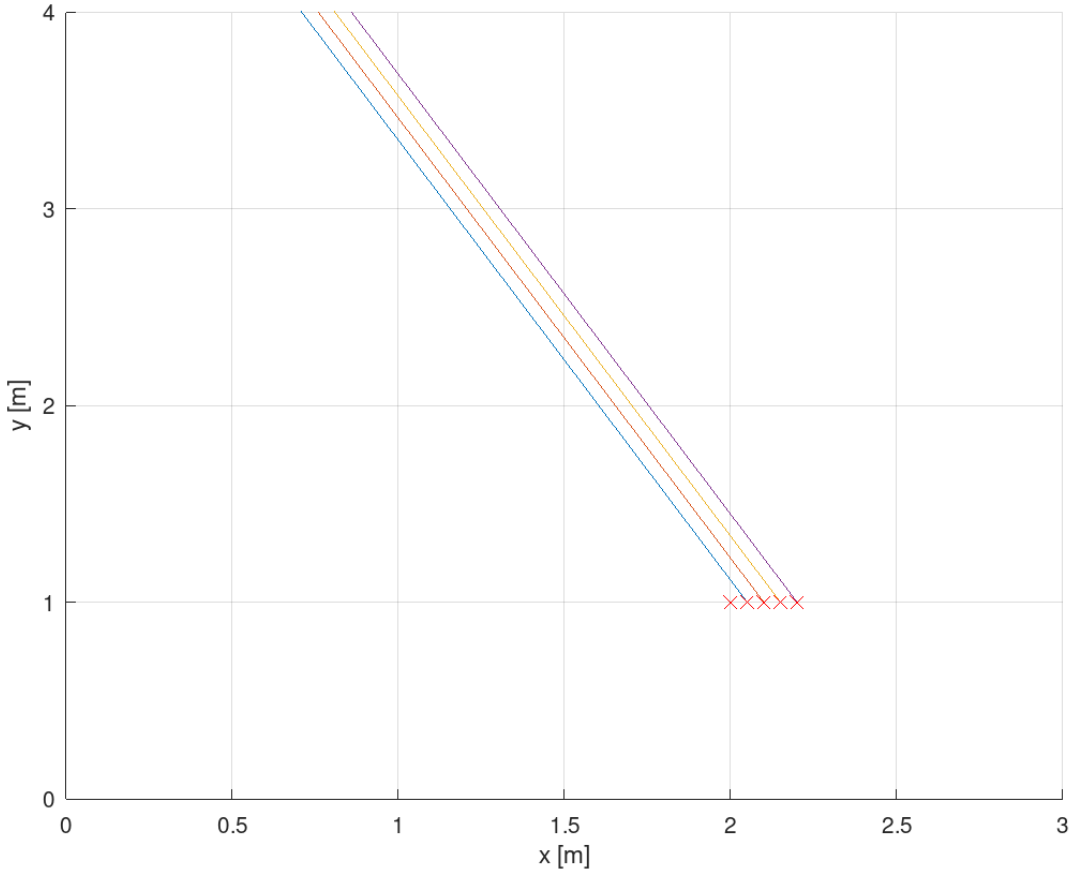

Figura 6: Disposición de los micrófonos y dirección de la que proviene el sonido para cada uno

Como era de esperar obtuvimos rectas paralelas en la dirección de la que proviene el sonido de la fuente. Debido a como se estima estas rectas se da el caso que el primer micrófono no tendrá una estimación. Los resultados obtenidos son lógicos en cuanto a que indicarían que la fuente se encuentra más cercana al micrófono 1 y luego se encuentra más alejada de los demás micrófonos secuencialmente, lo que es coherente con los retardos que habíamos observado en la **Figura 3**. El hecho de que establezcamos que la fuente proviene desde el lado superior del plano o el inferior es totalmente irrelevante ya que las distancias serían equivalentes, por lo que ambas propuestas serían válidas. La elección del plano superior se debe simplemente a utilizar el resultado del algoritmo tal cual nos viene dado y que la distribución del tamaño de la habitación en la que supuestamente fueron grabadas las señales nos da más rango en el plano superior.

Ejercicio 3

Calcule las correlaciones entre cada par de señales de micrófonos consecutivos en el dominio temporal y mediante la IDFT de GCC-PHAT. Obtenga los retardos a partir de ambos métodos.

Para el cálculo del retardo entre las señales en el dominio temporal debemos calcular la correlación

$$r_{xy} = \sum_{n=-\infty}^{\infty} x[n]y[n-l]$$

donde **x[n]** es el micrófono i-ésimo y **y[n]** es el micrófono i-ésimo + 1 (es decir, el micrófono consecutivo). El valor de **l** que maximice este cálculo será entonces el retardo en muestras estimado. Calculando para cada par de micrófonos consecutivos obtenemos entonces los siguientes retardos (estos retardos se miden respecto al micrófono anterior).

- **Micrófono 2**: 62,5 us (respecto al Micrófono 1)
- **Micrófono 3**: 62,5 us (respecto al Micrófono 2)
- **Micrófono 4**: 62,5 us (respecto al Micrófono 3)
- **Micrófono 5**: 83,3 us (respecto al Micrófono 4)

Vemos que nuestra estimación inicial con el gráfico de la **Figura 3** resultó bastante acertada. Excepto por el retardo entre el cuarto y quinto micrófono, todos los demás retardos dieron aproximadamente el valor que habíamos estimado de 60 us. En rigor, realmente estimó lo mismo en muestras (3 muestras) que nosotros con la salvedad de que nosotros habíamos redondeado el tiempo entre muestras a 20 us para hacer más amenas las cuentas.

Procederemos ahora a calcular los retardos mediante el segundo método (GCC-PHAT), obteniendo en este caso los siguientes retardos.

- **Micrófono 2**: 62,5 us (respecto al Micrófono 1)
- **Micrófono 3**: 62,5 us (respecto al Micrófono 2)
- **Micrófono 4**: 62,5 us (respecto al Micrófono 3)
- **Micrófono 5**: 62,5 us (respecto al Micrófono 4)

En un principio la diferencia en el último retardo podría sorprender. Mediante este segundo método resulta que estamos obteniendo unos 20 us de diferencia en la última medición, y en términos relativos estaríamos observando una diferencia del 33 % entre esas mediciones. Esto puede entenderse rápidamente si recordamos que debido a nuestra frecuencia de muestreo **Fs** de 48 KHz cada muestra se realiza cada unos 20 us. Como el retardo observado se encuentra alrededor de 3 muestras entonces tan solo una muestra de diferencia en una medición implicaría una diferencia porcentual muy importante. Se entiende entonces que este método estimó que el retardo entre el cuarto y quinto micrófono es de 3 muestras, lo que resultaría en una sola muestra de diferencia si comparamos con el resultado de ese retardo del anterior método.

Ejercicio 4

Estimar los retardos de llegada y graficar su evolución temporal utilizando el algoritmo de ventaneo propuesto. Realizar los histogramas de los retardos calculados para cada micrófono. Obtener un valor representativo de retardo para cada señal. Estimar y graficar la posición de la fuente.

Para el algoritmo se utilizó una ventana de 20.000 muestras con un overlapping de 10.000 muestras. Se aclara que si para algunos ventaneos se obtienen retardos inversos (es decir que el micrófono **i**-ésimo está atrasado respecto al **i**-ésimo + 1) o el retardo es nulo entonces dichos valores no se toman en cuenta entre los resultados ya que son claramente absurdos. La recta roja en los gráficos de los retardos obtenidos denota el valor medio.

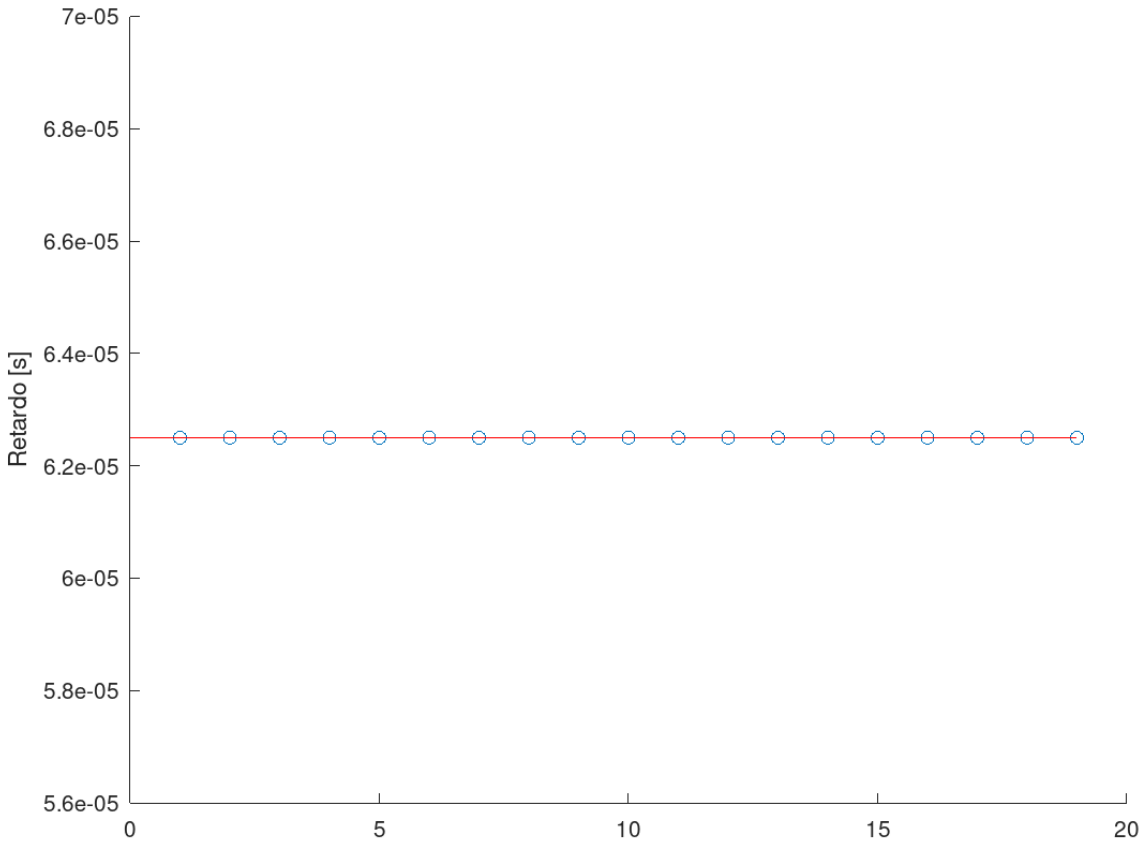

Figura 7: Retardos obtenidos para el micrófono 2 respecto al micrófono 1

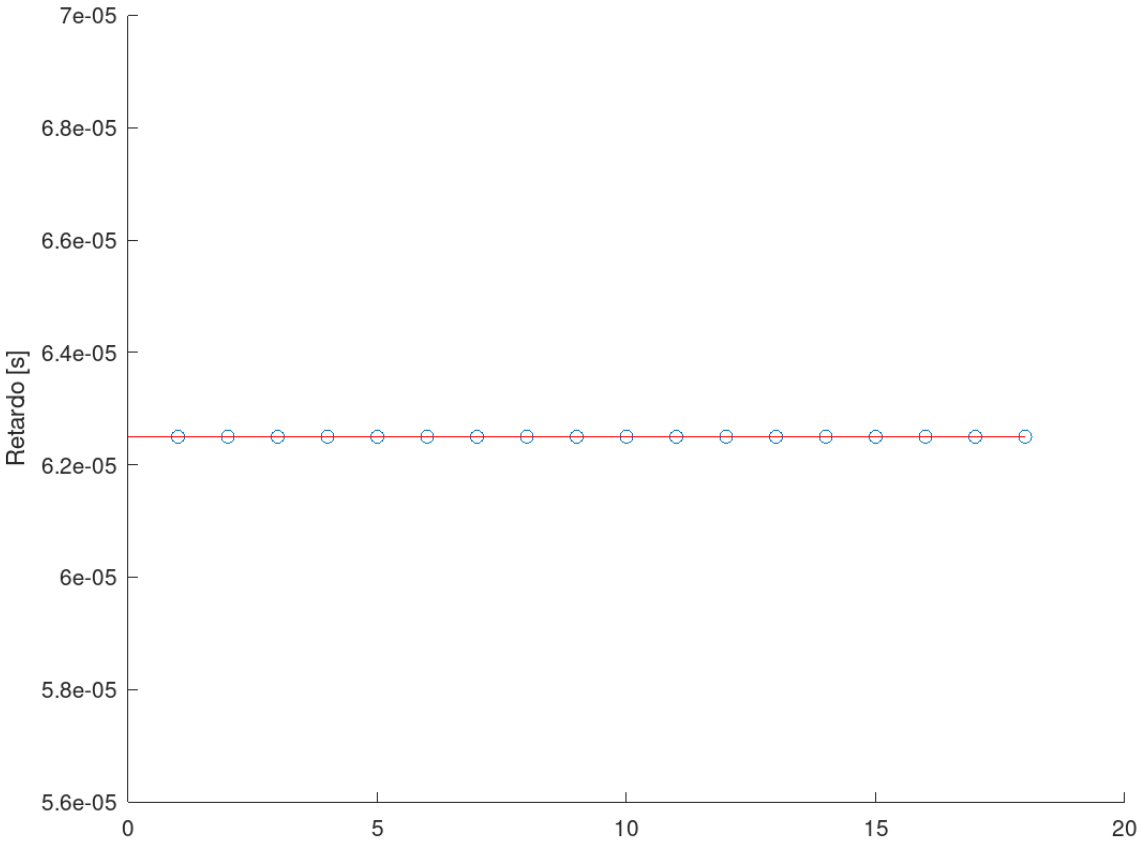

Figura 8: Retardos obtenidos para el micrófono 3 respecto al micrófono 2

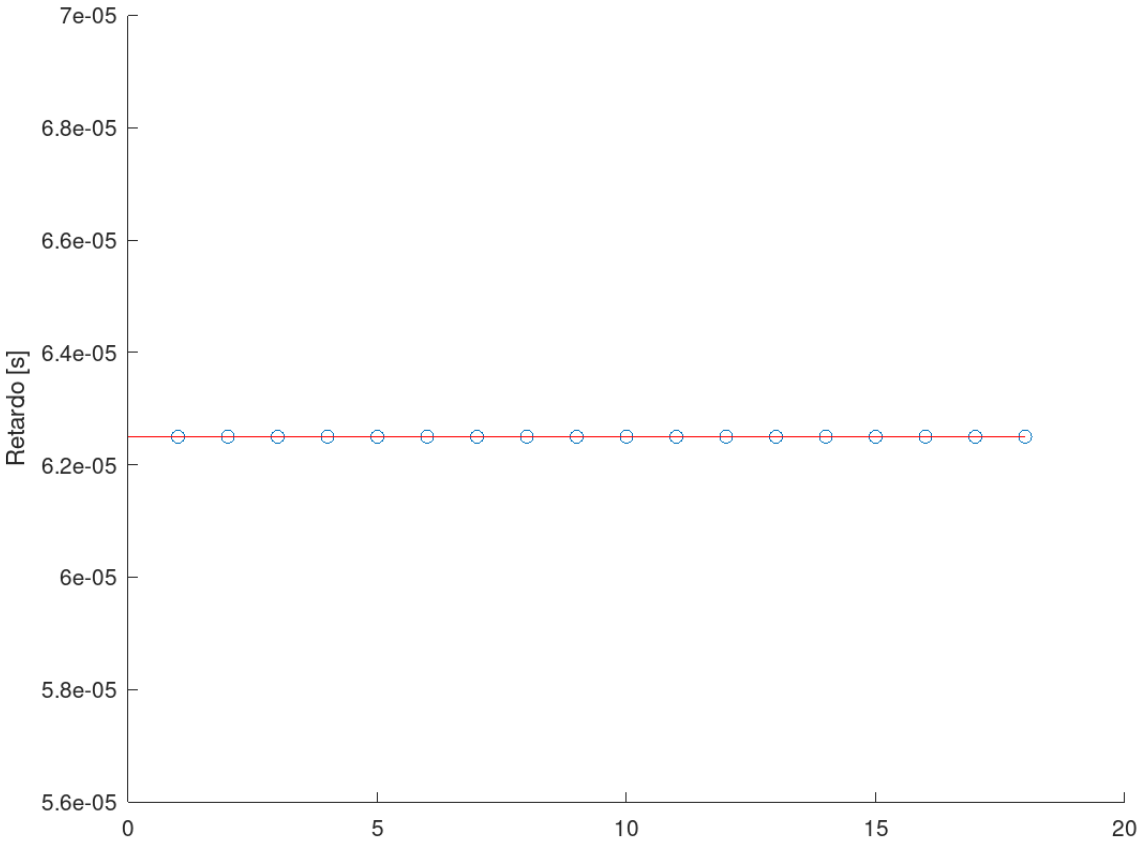

Figura 9: Retardos obtenidos para el micrófono 4 respecto al micrófono 3

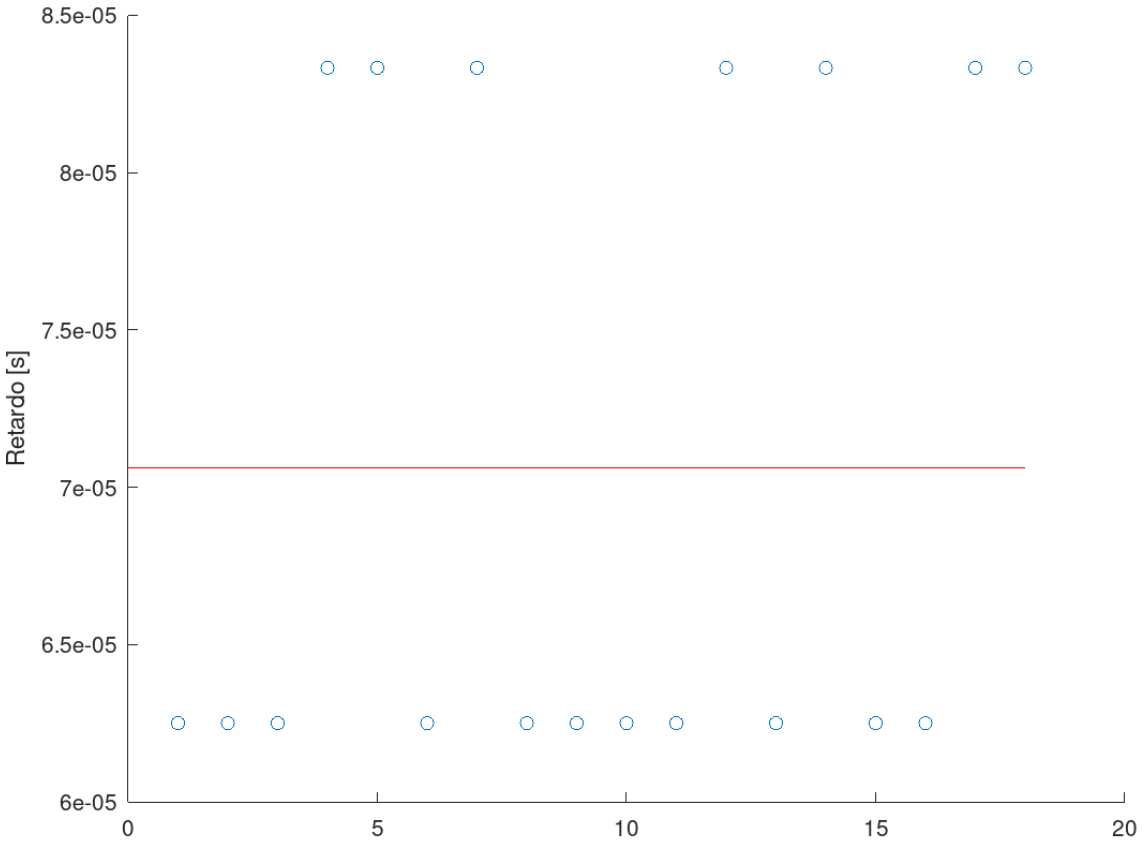

Figura 10: Retardos obtenidos para el micrófono 5 respecto al micrófono 4

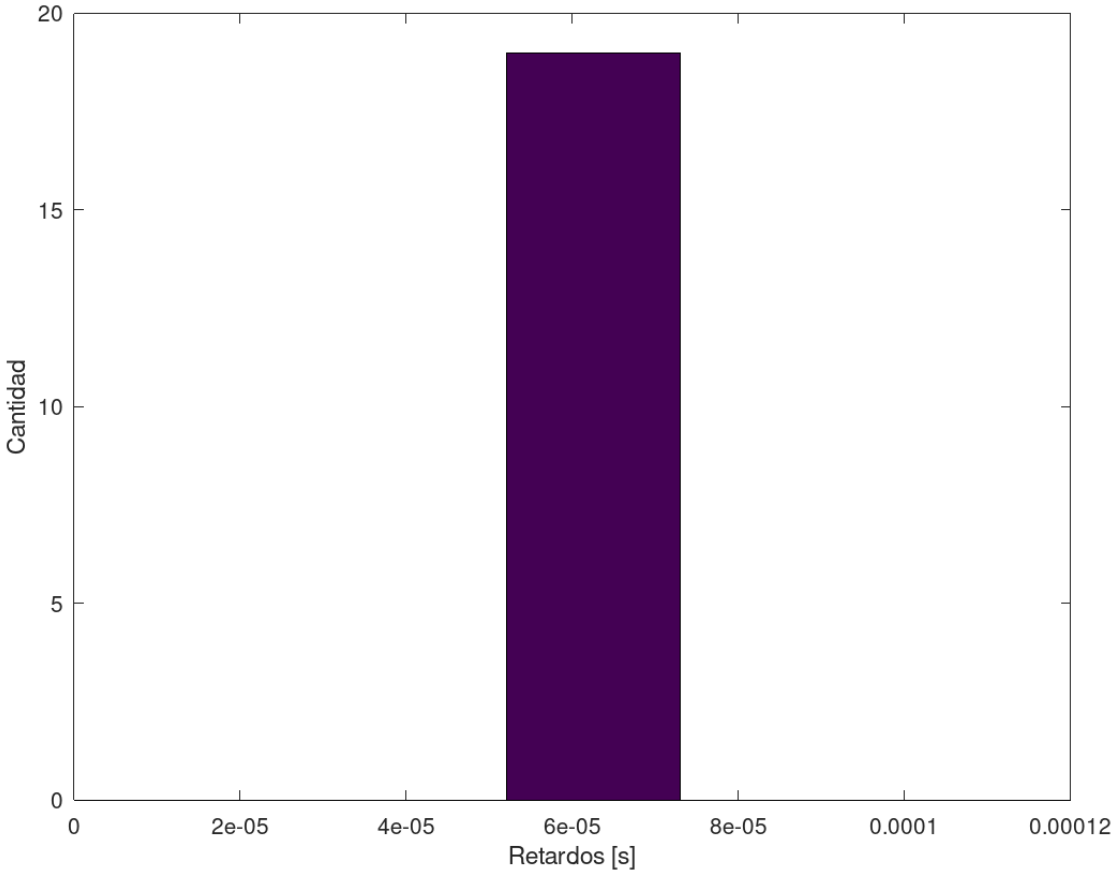

Figura 11: Histograma de los retardos del micrófono 2 respecto al micrófono 1

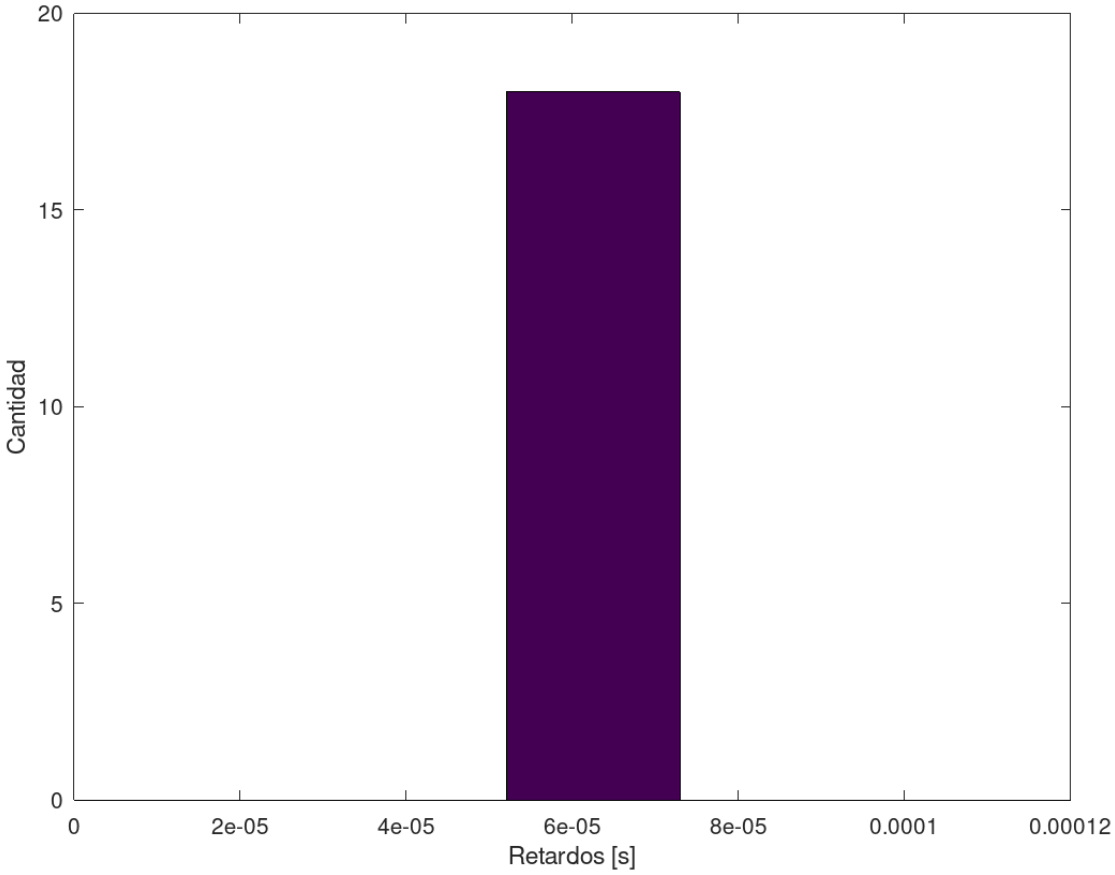

Figura 12: Histograma de los retardos del micrófono 3 respecto al micrófono 2

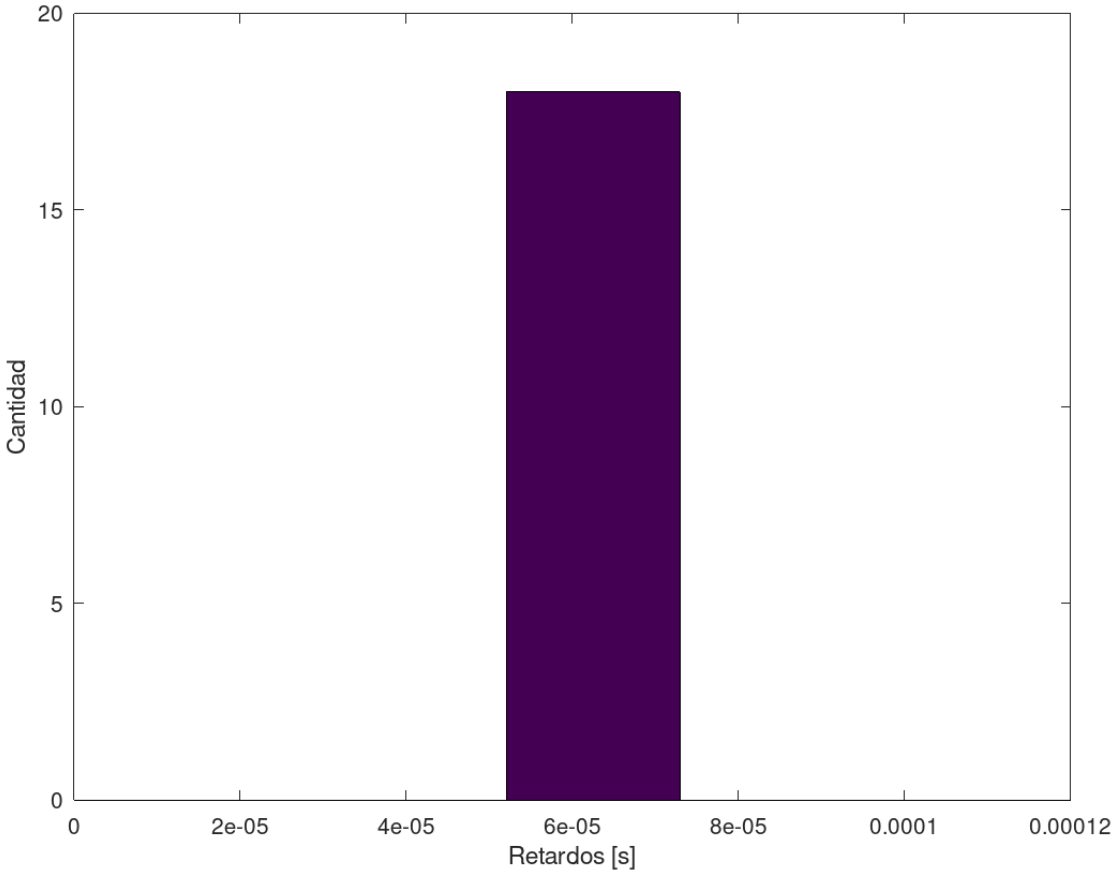

Figura 13: Histograma de los retardos del micrófono 4 respecto al micrófono 3

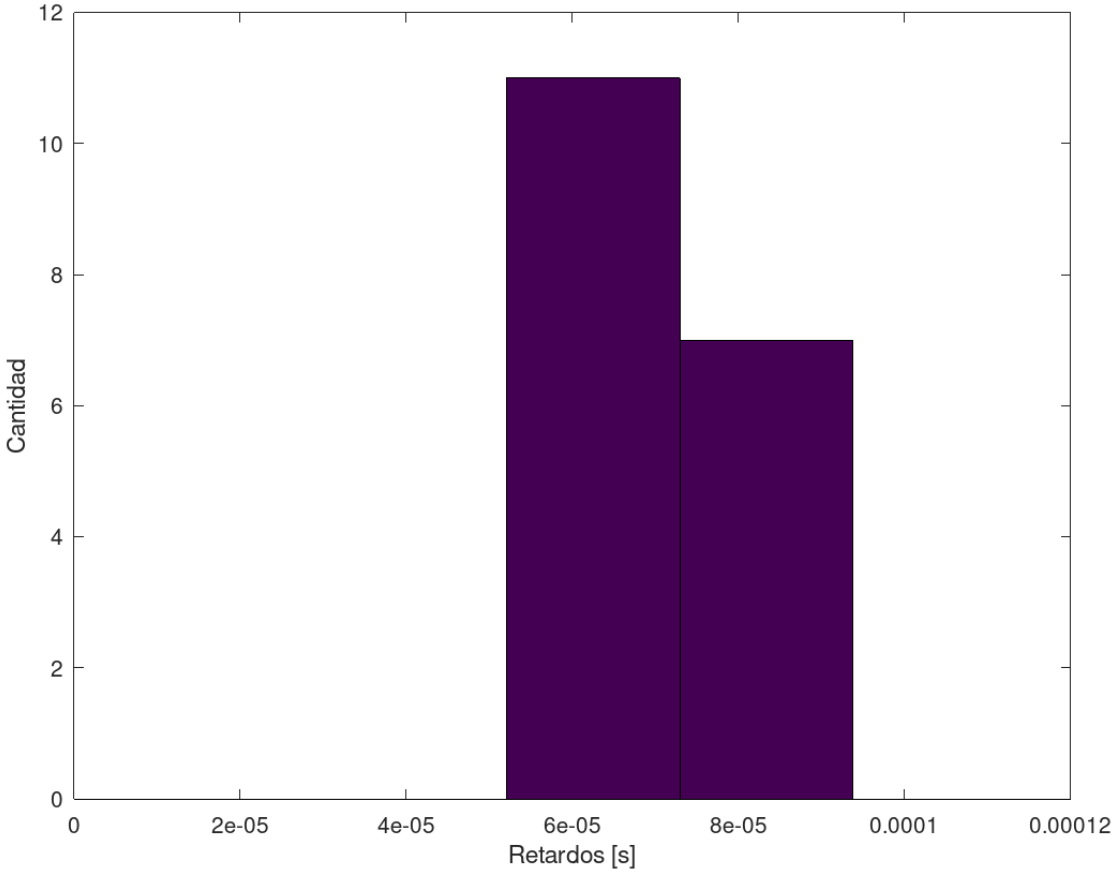

Figura 14: Histograma de los retardos del micrófono 5 respecto al micrófono 4

Los valores medio de los retardos obtenidos fueron:

- **Micrófono 2**: 62,5 us (respecto al Micrófono 1)
- **Micrófono 3**: 62,5 us (respecto al Micrófono 2)
- **Micrófono 4**: 62,5 us (respecto al Micrófono 3)
- **Micrófono 5**: 70,6 us (respecto al Micrófono 4)

Nuevamente conseguimos valores en el rango de los 60 us. El tamaño de la ventana no es realmente importante ya que la resolución que tenemos en el muestreo no es lo suficientemente alta como para poder apreciar diferencias sutiles independientemente del ventaneo realizado. Un detalle importante a notar es que vemos que el retardo entre el micrófono 4 y 5 es de 70 us. Esto es lógico si observamos que proviene de algunas mediciones en el ventaneo donde se obtuvo que el retardo era de 4 muestras, lo que daría un retardo temporal de 80 us. Al promediar todos los retardos obtenidos esto elevaría el retardo medio hacia los 70 us. El motivo por el que este es el único retardo para el que estimó 4 muestras en algunas mediciones en lugar de 3 para todas (como fue el caso en los demás micrófonos) se explica fácilmente por la cercanía en el retardo real a un valor cercano a los 70 us. Debido a que el retardo incrementa entre los distintos micrófonos (ya que el frente de onda del sonido proveniente de la fuente **no** es plano) resultará entonces que el quinto micrófono presentará el mayor de los retardos respecto al micrófono anterior. Si este retardo se encuentra en los 70 us aproximadamente entonces es posible que alguna medición termine por estimar que el retardo es de 4 muestras, es decir, 80

us. Esto explicaría los valores obtenidos tanto aquí como en el **Ejercicio 3**, donde obtuvimos 83,3 us para el último micrófono con el método de la correlación cruzada.

Graficando las direcciones del sonido respecto a cada micrófono observamos lo siguiente.

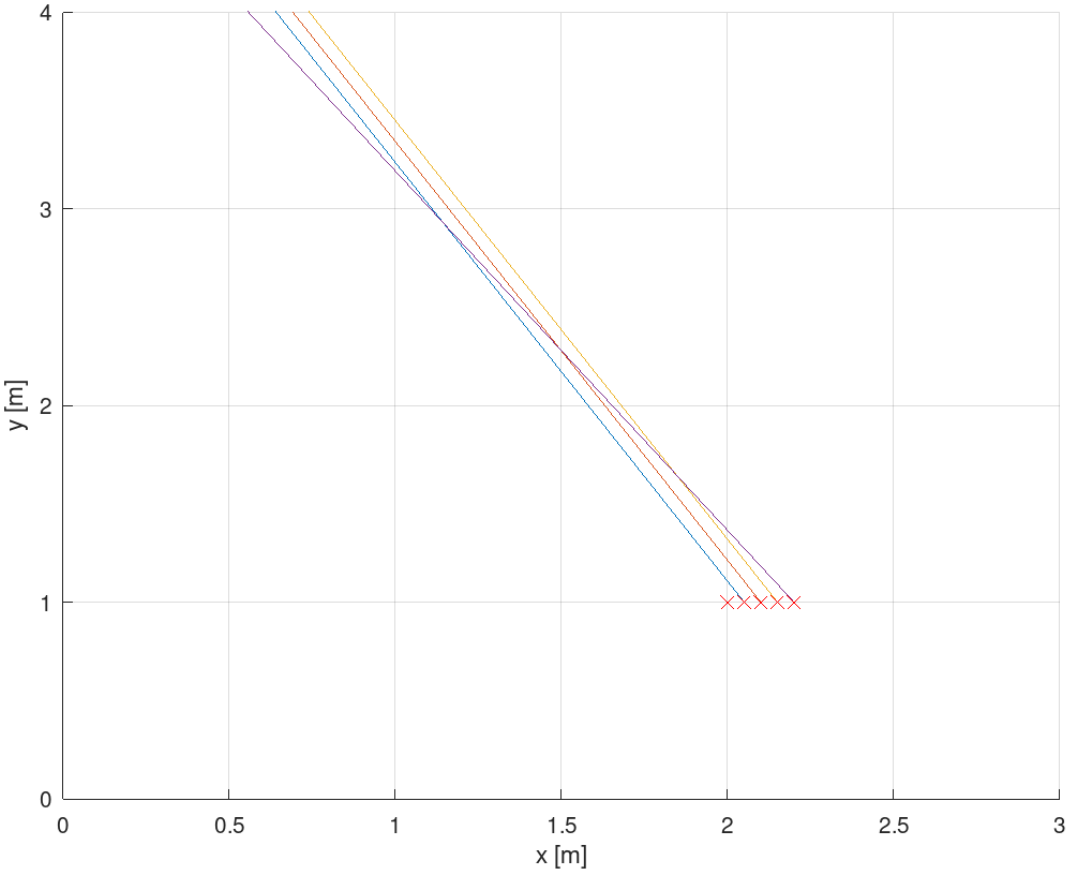

Figura 15: Disposición de los micrófonos y dirección de la que proviene el sonido para cada uno

Al igual que con los anteriores métodos resultará imposible tener una estimación de la posición de la fuente sin caer en altas cantidades de error. Este necesariamente será el caso siempre que sigamos manejándonos con una resolución de 20 us por muestra. Es necesario sobremuestrear ya sea las señales originales o la IDFT(Gph[k]) para obtener una precisión aceptable. De todas formas, si ignoramos el hecho de que las tres primeras rectas son paralelas y utilizamos la cuarta para estimar una posición de la fuente podríamos estimar (con muchísimo error, claro) la siguiente posición.

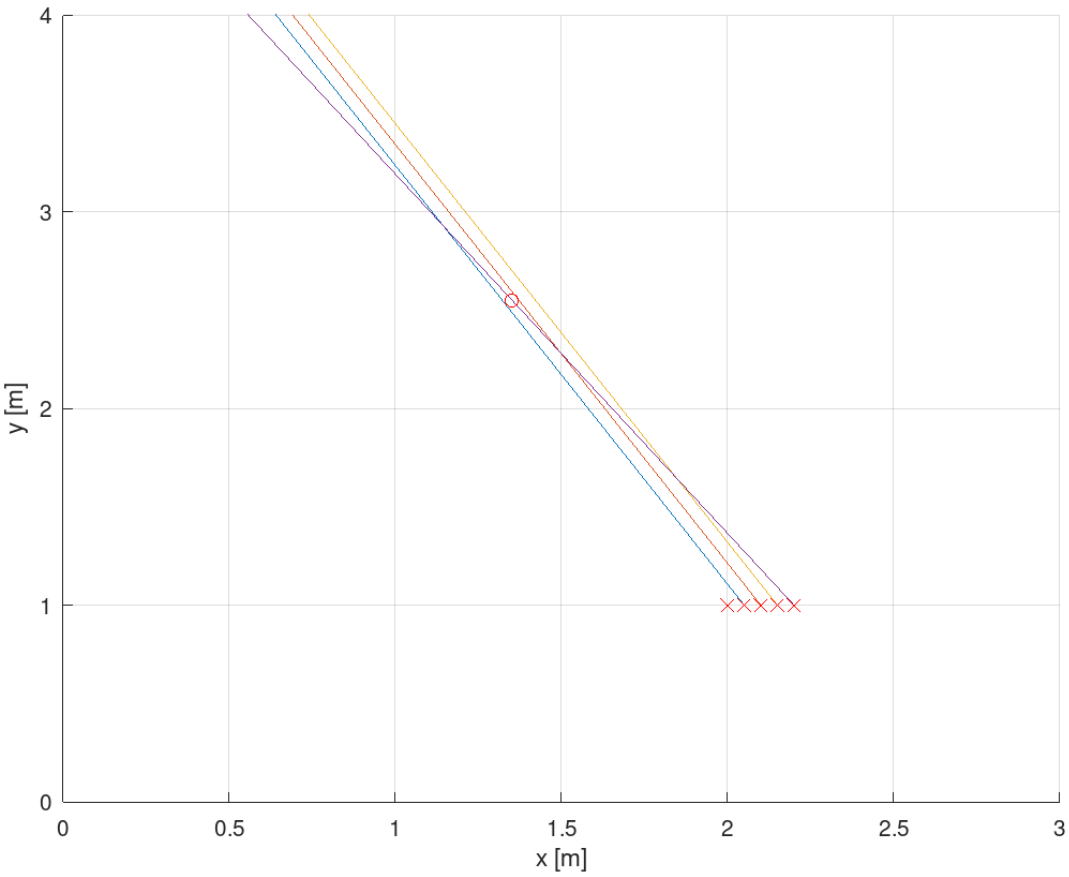

Figura 16: Disposición de los micrófonos y de la fuente de sonido

Para estimar la posición marcada en el gráfico se promediaron las intersecciones entre las distintas rectas. La posición de la fuente estimada es aproximadamente a 1,35 metros en **x** y 2,55 metros en **y** respecto del origen.

Ejercicio 5

Agregar, en forma independiente, un ruido blanco con una SNR de 20 dB a las señales capturadas por los micrófonos. Escuchar ambas versiones y comparar sus espectros. Repetir los ejercicios 1 a 4 con las nuevas señales.

En primer lugar, grafiquemos nuevamente las señales en el tiempo. Esperaríamos ver un gráfico muy similar al de la **Figura 2** sólo que más distorsionado debido al ruido blanco agregado. Esta distorsión será especialmente notable en aquellos intervalos de las señales donde originalmente teníamos silencio, dado que en esas secciones veremos que lo que para el gráfico de la señales sin ruido se veía como una línea horizontal fina resultará más gruesa en el gráfico de las señales contaminadas con ruido blanco (observar, por ejemplo, el intervalo entre los 2 y 2,5 segundos).

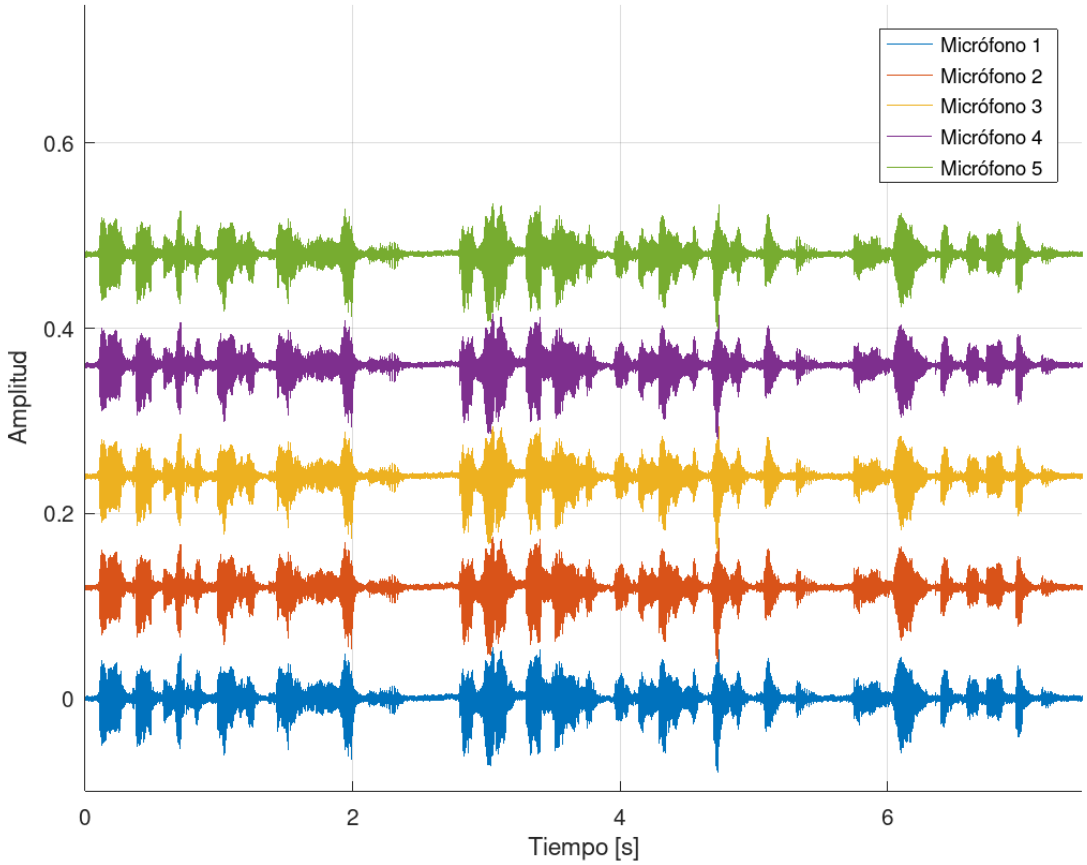

Figura 17: Señales registradas por los micrófonos con ruido agregado

Si observamos ahora el mismo intervalo que utilizamos antes para estimar visualmente los retardos veremos que el ruido provocó que la señales pierdan todo tipo de parecido en ese intervalo.

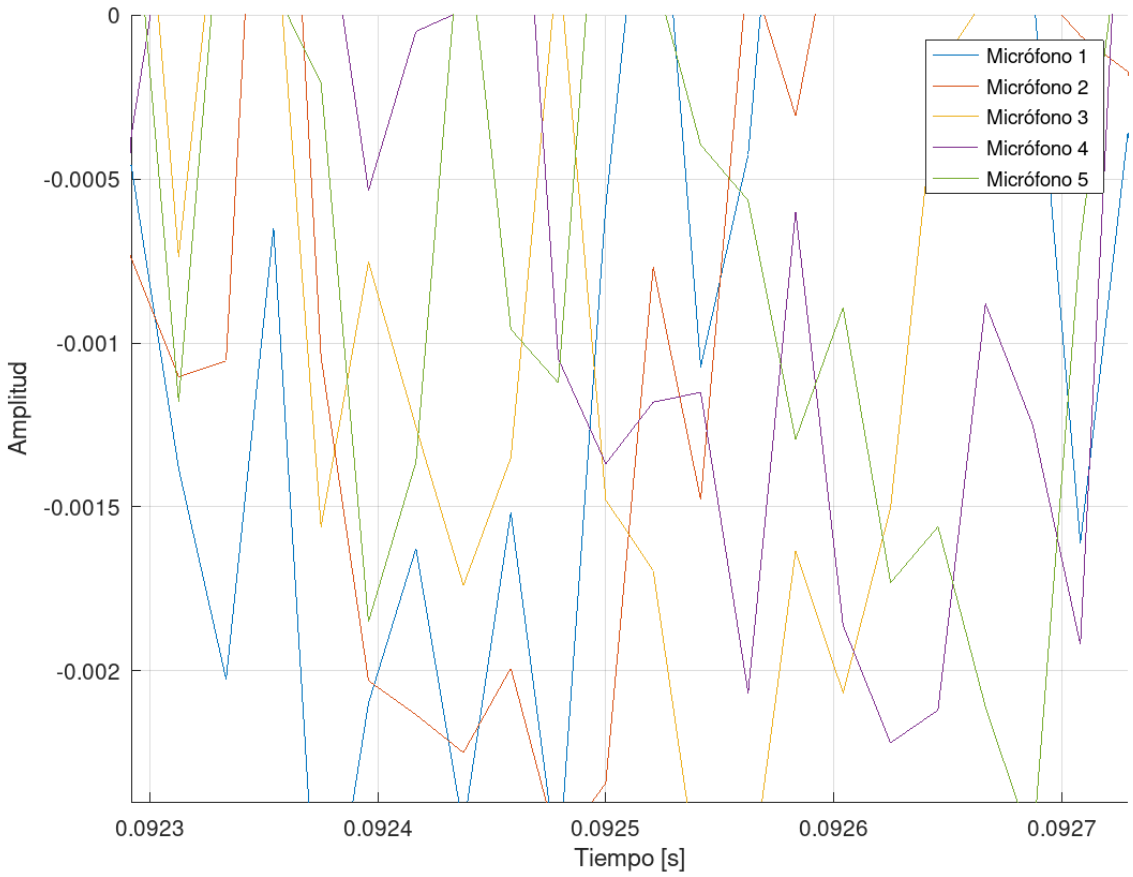

Figura 18: Señales con ruido registradas por los micrófonos entre los 92,3 **ms** y 92,7 **ms**

Si bien es posible seleccionar un intervalo considerablemente más grande para intentar estimar un retardo entre las señales realmente no resultaría en resultados ni remotamente confiables, y el ruido complicaría bastante cualquier análisis visual del retardo, por lo que estimaremos los retardos utilizando únicamente los métodos algorítmicos previos.

Observemos ahora los espectros de la señales y el espectrograma de la señal del micrófono 1 (nuevamente, no tendría sentido realizar el espectrograma de todos los micrófonos ya que serían prácticamente iguales).

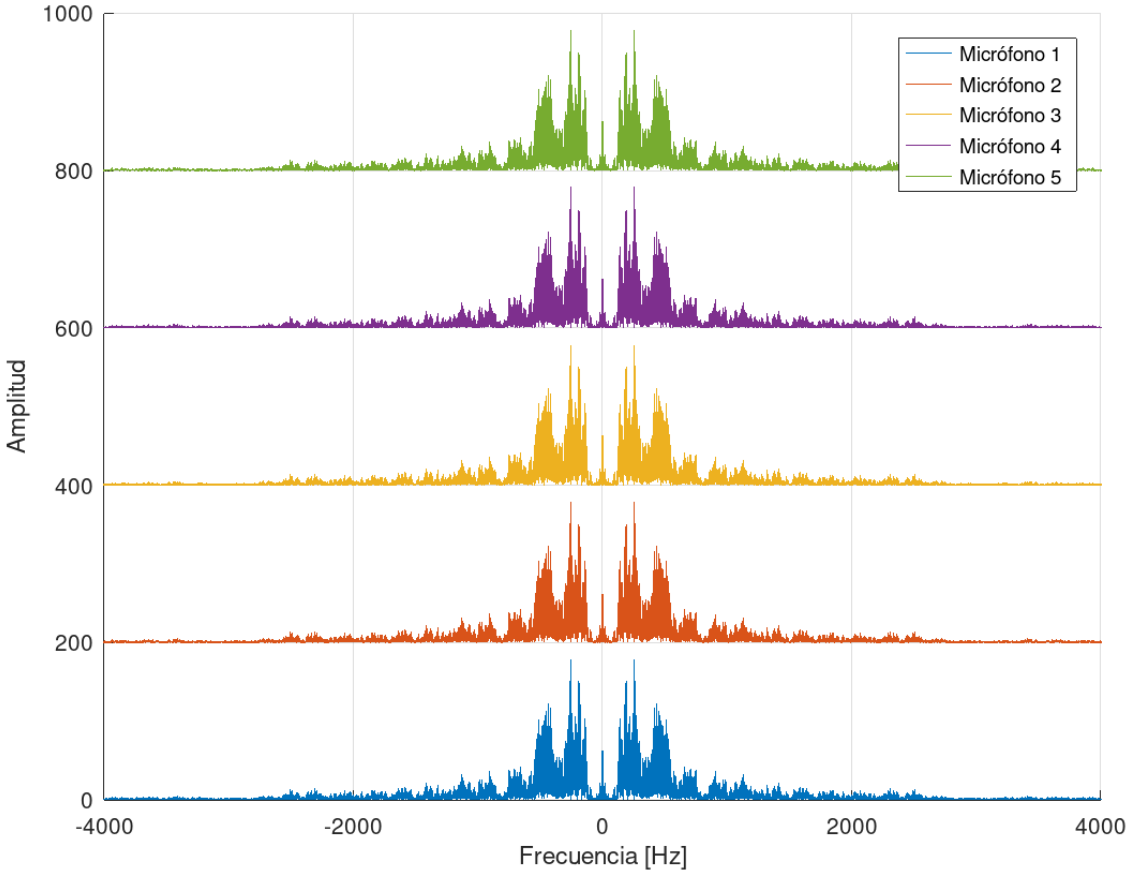

Figura 19: Magnitud del espectro de los micrófonos con ruido entre -4000 Hz y 4000 Hz

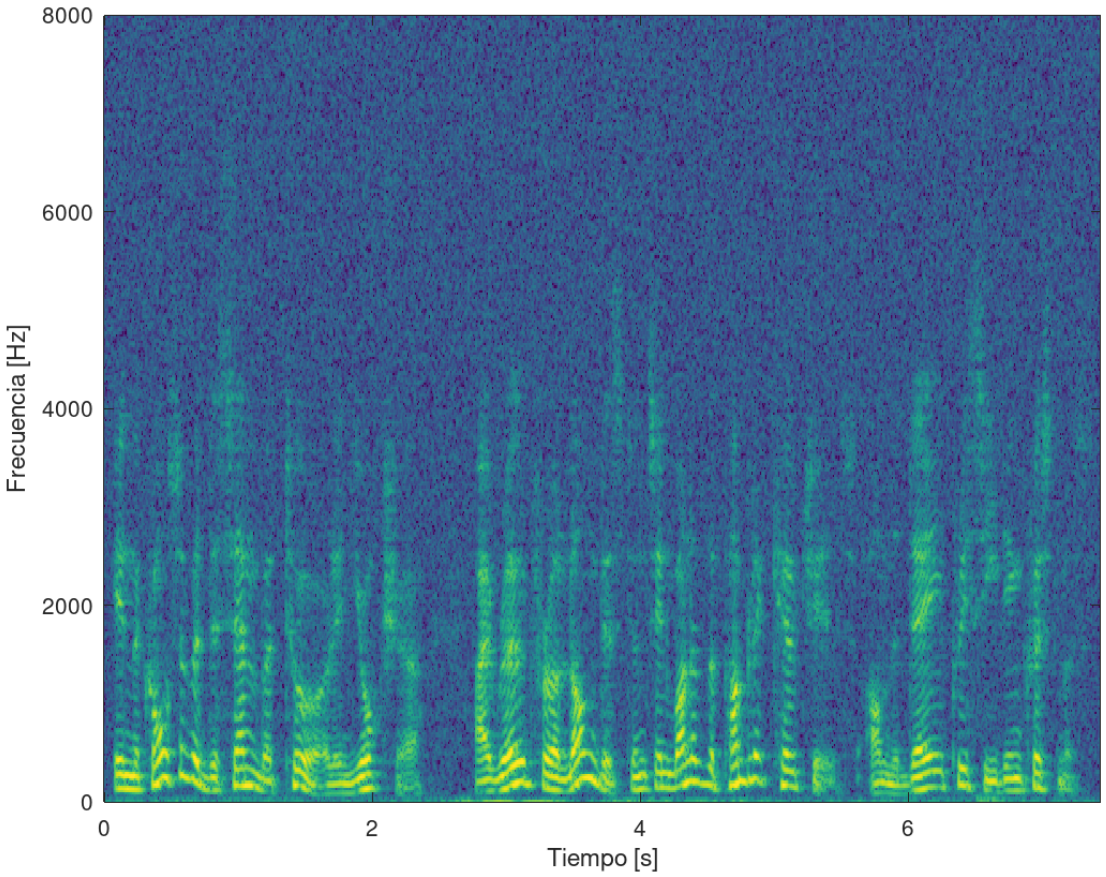

Figura 20: Espectrograma del Micrófono 1 con ruido entre los 0 Hz y 8000 Hz

En este caso el gráfico de la magnitud del espectro de las señales no nos resulta útil a la hora de identificar diferencias con las señales originales que no tenían el ruido introducido. En cambio sí que podemos notar diferencias entre los espectrogramas, donde ahora vemos que las frecuencias que antes tenían potencia despreciable (tonalidad violeta) ahora han adquirido todas una potencia uniforme notablemente inferior aún que las frecuencias principales de nuestra señal (las amarillas) pero mucho mayor que antes de agregar el ruido (evidenciado por la tonalidad azulada del nuevo espectrograma). Esto es lógico ya que el ruido blanco se distribuye uniformemente entre todas las frecuencias.

Los siguientes fueron los resultados de los retardos obtenidos mediante el método de correlación cruzada.

- **Micrófono 2**: 62,5 us (respecto al Micrófono 1)
- **Micrófono 3**: 62,5 us (respecto al Micrófono 2)
- **Micrófono 4**: 62,5 us (respecto al Micrófono 3)
- **Micrófono 5**: 83,3 us (respecto al Micrófono 4)

A pesar del ruido agregado obtuvimos exactamente los mismos resultados para este método. Veamos ahora los resultados que obtenemos con el método de GCC-PHAT sin ventaneos.

- **Micrófono 2**: 62,5 us (respecto al Micrófono 1)

- **Micrófono 3**: 62,5 us (respecto al Micrófono 2)
- **Micrófono 4**: 62,5 us (respecto al Micrófono 3)
- **Micrófono 5**: 83,3 us (respecto al Micrófono 4)

Aquí observamos la primera diferencia respecto a los resultados que habíamos obtenido para las señales originales, obteniendo un retardo de 83,3 us para el micrófono 5 respecto al micrófono 4 cuando previamente todos los retardos nos habían dado 62,5 us. Es, sin embargo, simplemente una diferencia de una muestra, por lo que a gran escala todavía no vimos cambios notorios.

Por otro lado, los resultados obtenidos mediante el ventaneo de las señales con ruido utilizando una ventana de 20.000 muestras con un overlapping de 10.000 muestras (mismos valores que utilizamos en el **Ejercicio 4** para mantener consistencia) fueron los siguientes.

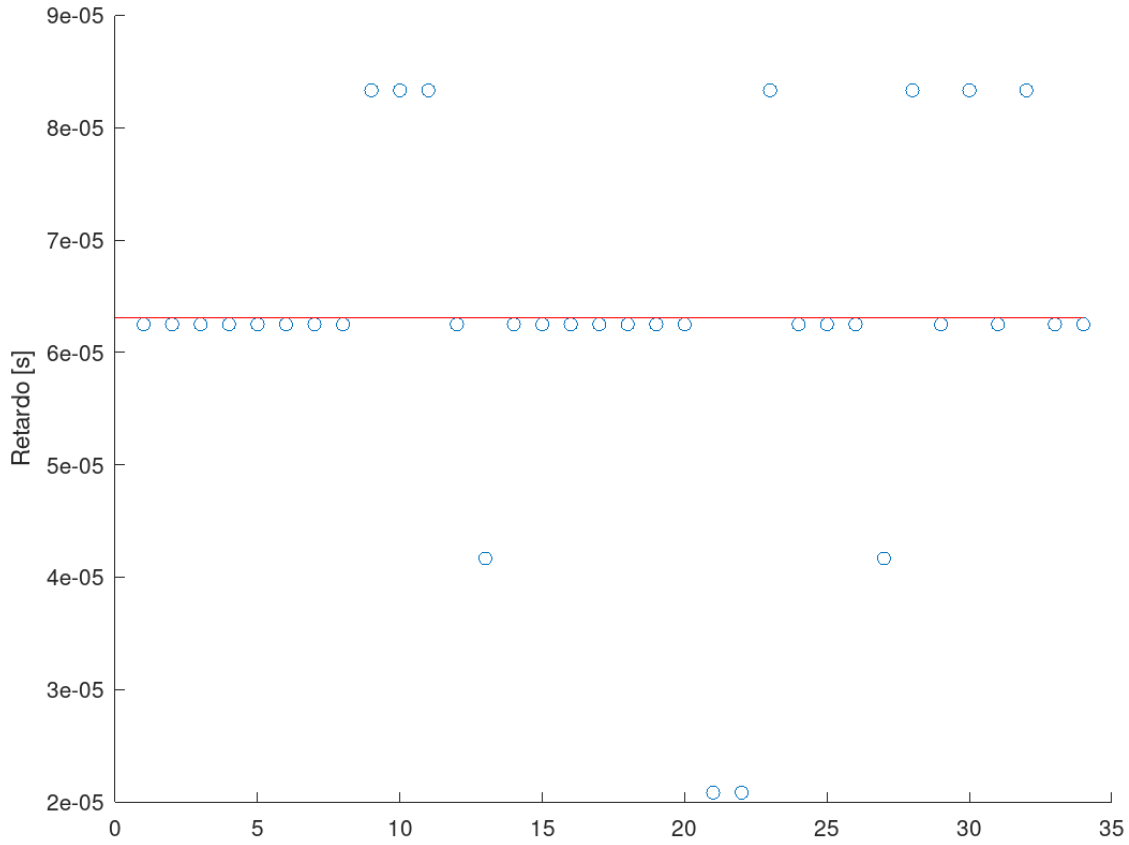

Figura 21: Retardos obtenidos para el micrófono 2 respecto al micrófono 1

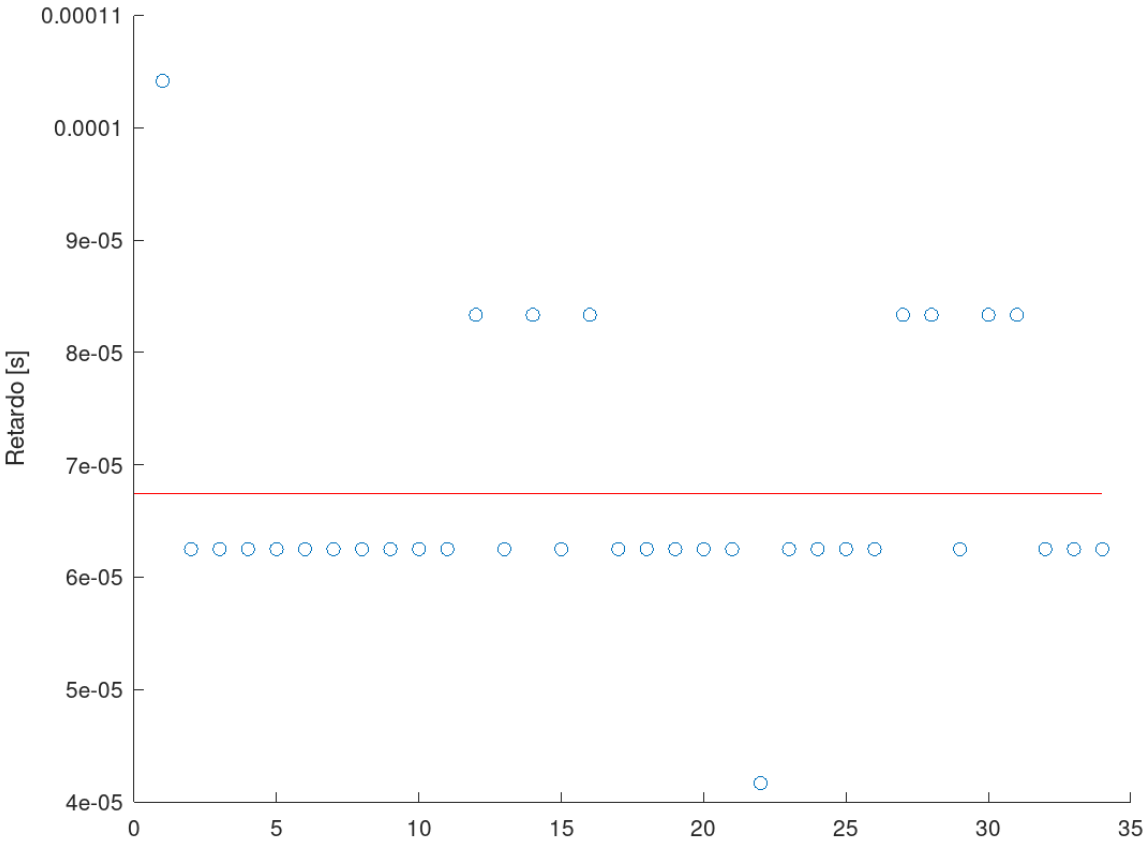

Figura 22: Retardos obtenidos para el micrófono 3 respecto al micrófono 2

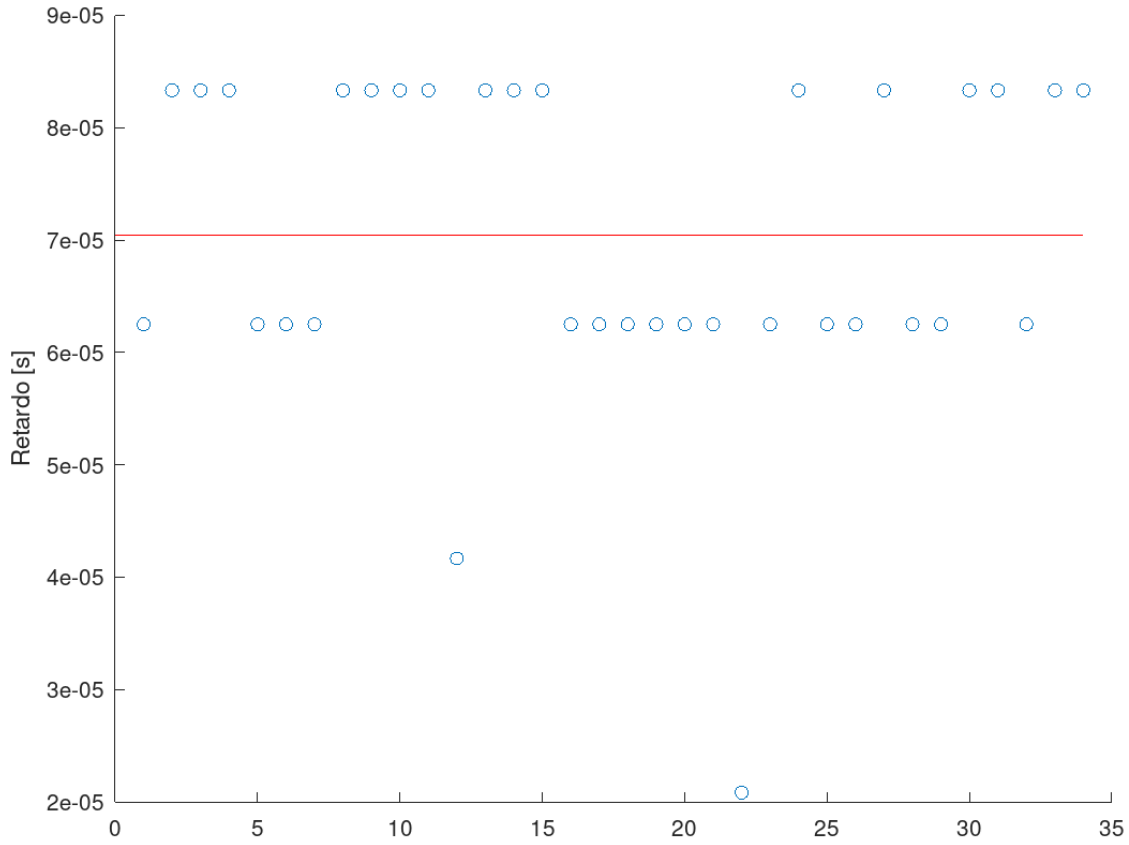

Figura 23: Retardos obtenidos para el micrófono 4 respecto al micrófono 3

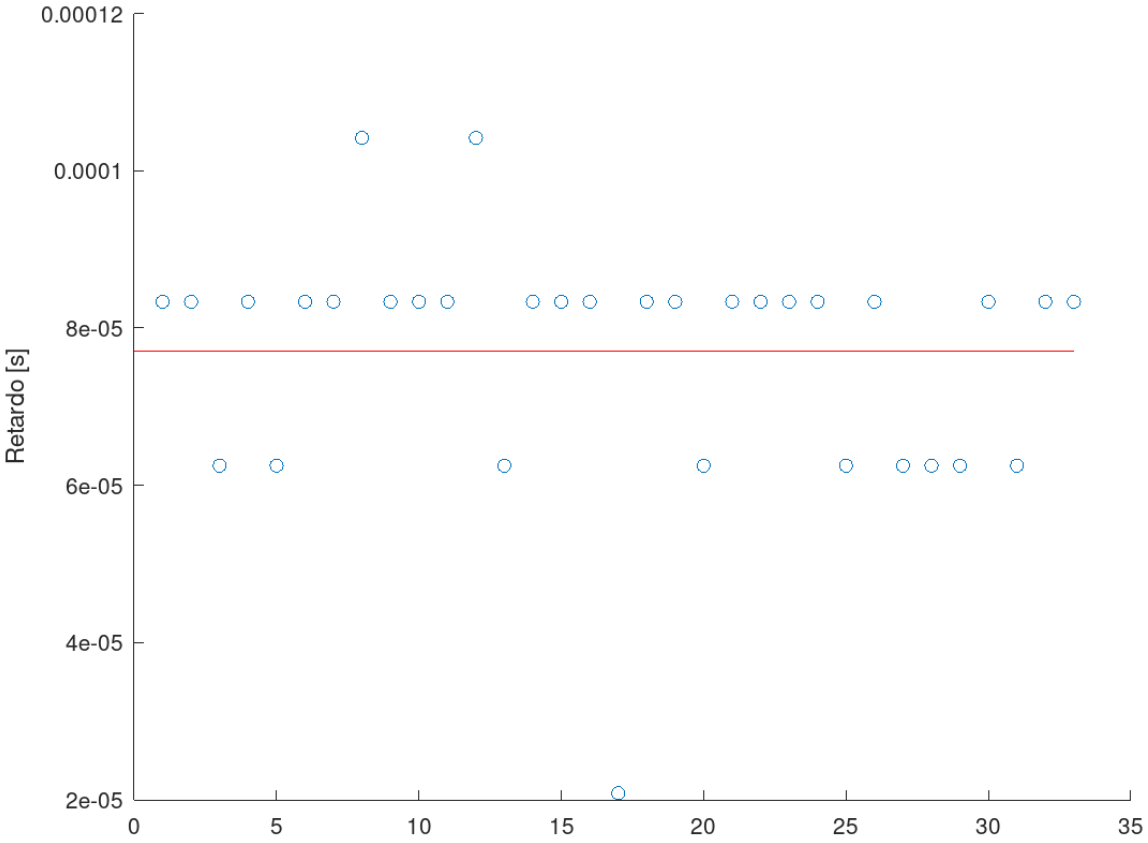

Figura 24: Retardos obtenidos para el micrófono 5 respecto al micrófono 4

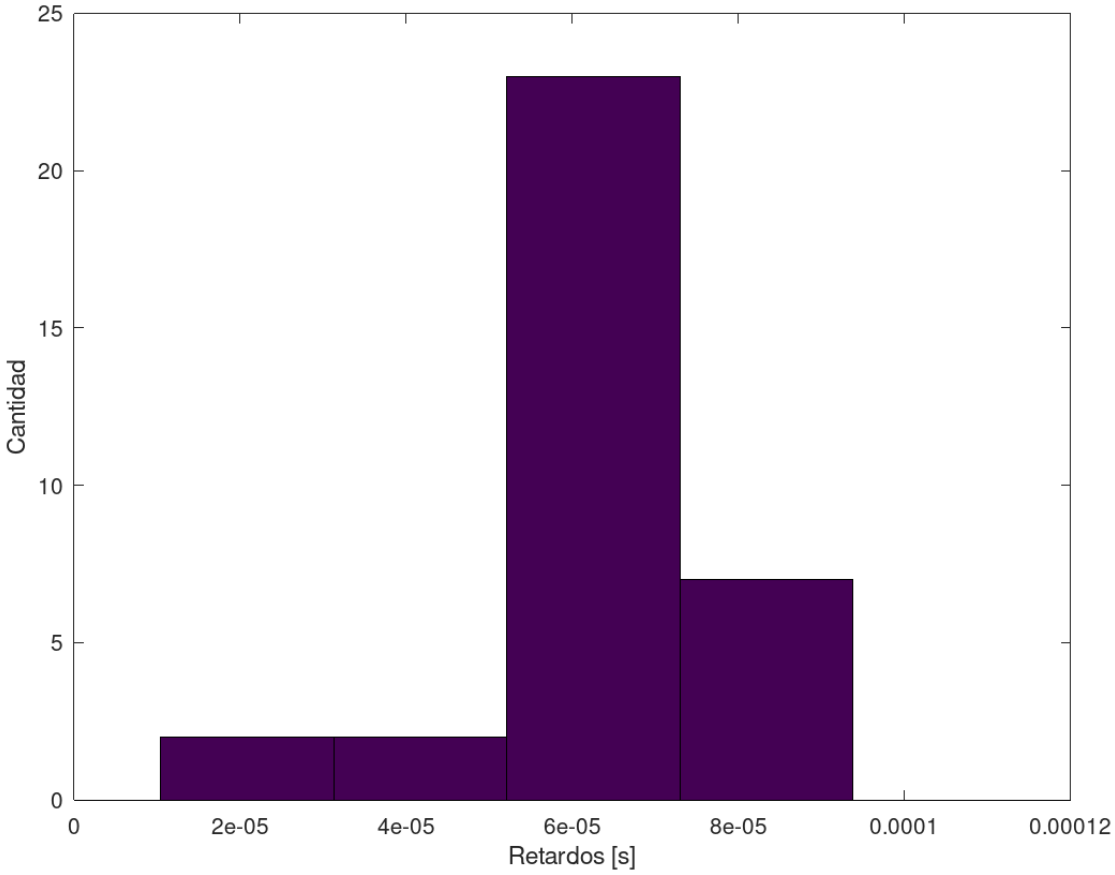

Figura 25: Histograma de los retardos del micrófono 2 respecto al micrófono 1

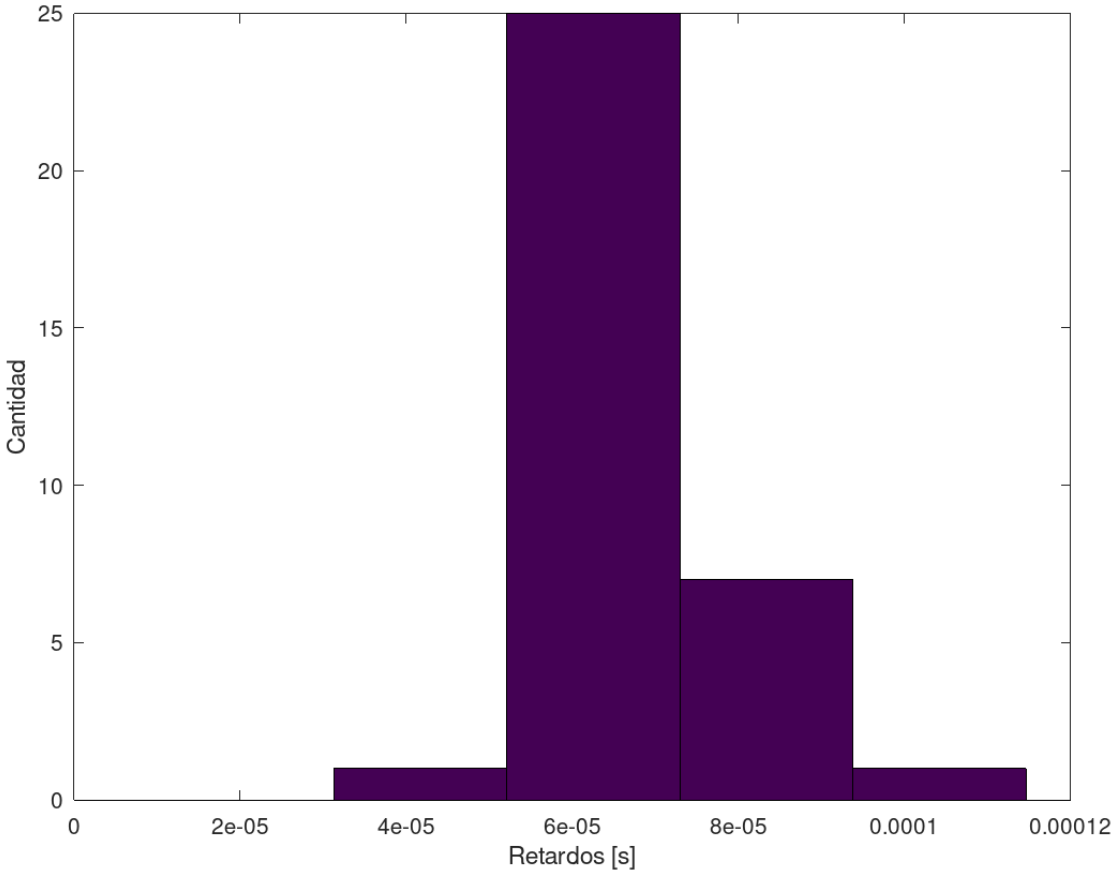

Figura 26: Histograma de los retardos del micrófono 3 respecto al micrófono 2

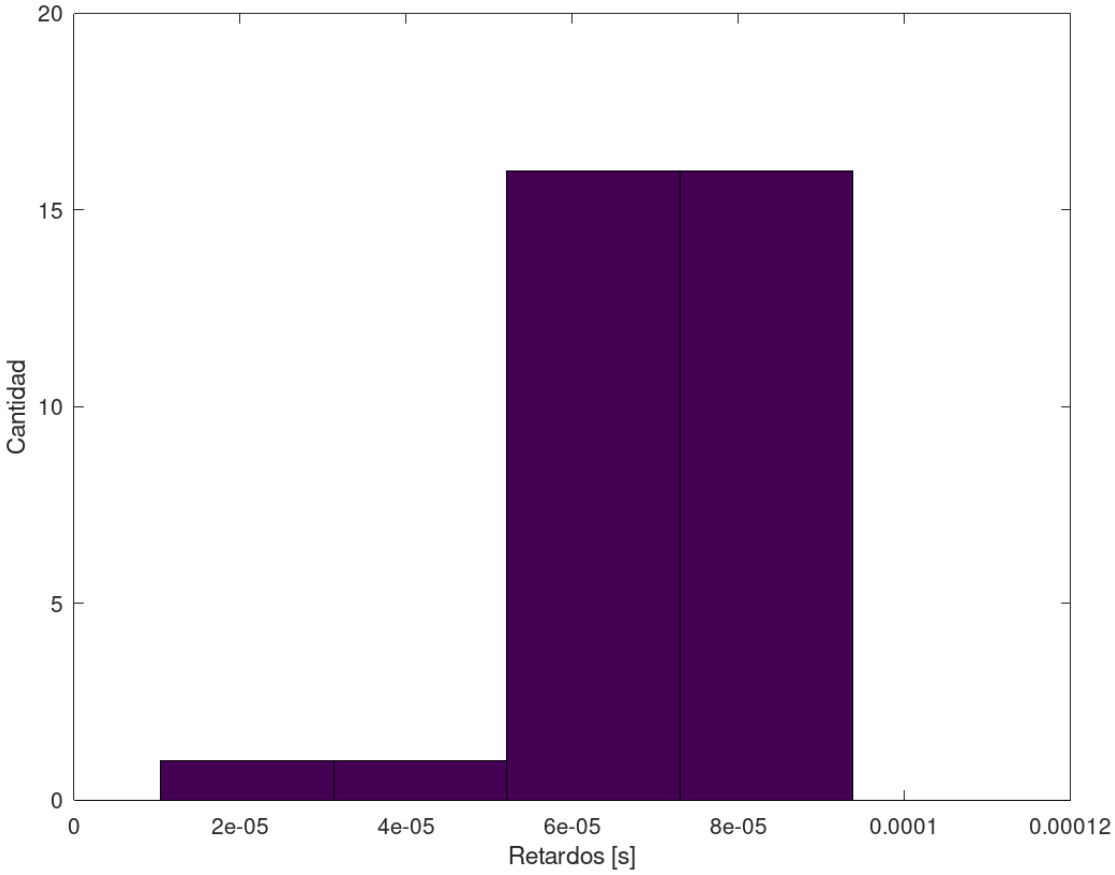

Figura 27: Histograma de los retardos del micrófono 4 respecto al micrófono 3

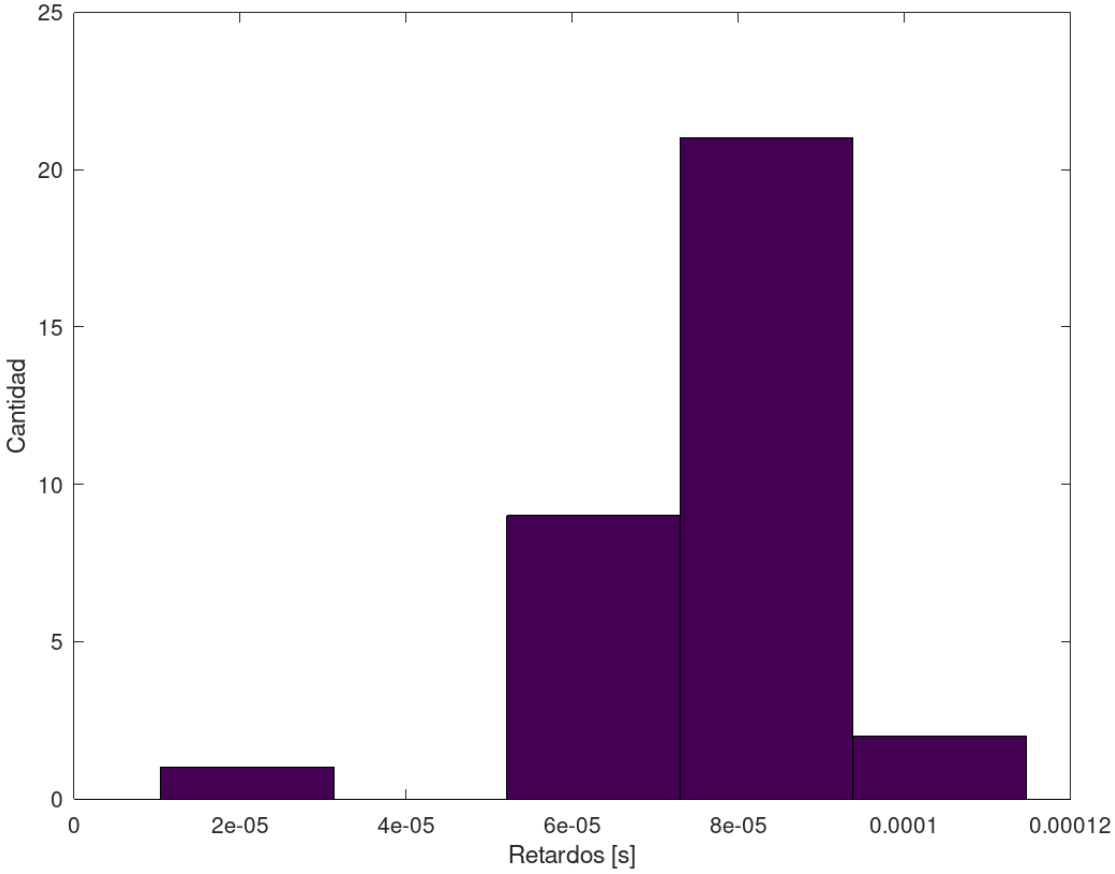

Figura 28: Histograma de los retardos del micrófono 5 respecto al micrófono 4

Los retardos medios obtenidos fueron:

- **Micrófono 2**: 63,1 us (respecto al Micrófono 1)
- **Micrófono 3**: 67,4 us (respecto al Micrófono 2)
- **Micrófono 4**: 70,5 us (respecto al Micrófono 3)
- **Micrófono 5**: 77,0 us (respecto al Micrófono 4)

Aquí sí que obtuvimos resultados diferentes a los obtenidos previamente. Vemos que los retardos van incrementando entre cada par de micrófonos consecutivos tal y como uno esperaría. Sin embargo es importante recalcar que no podemos interpretar estos resultados como mediciones precisas. El hecho de que sigamos teniendo una resolución de únicamente unos 20 us por muestra sigue resultando en una diferencia porcentual muy grande entre mediciones y no es un problema que el ruido mitigue. Que las mediciones con el ruido separen más los retardos se debe a la distorsión introducida a la señales que ocasionará que sea más común que distintas mediciones difieran en 1 muestra (o más) en el retardo calculado. La proporción de los retardos obtenidos establecerá entonces el retardo medio obtenido (lógicamente) y viendo que los retardos medios se encuentran entre los 60 us y 80 us entonces la proporción será principalmente entre estos valores. Dado que el ruido es de naturaleza aleatoria entonces cómo afecte a las señales y cómo se vea esto reflejado en los retardos también lo será.

Graficando las direcciones de las señales detectadas por los micrófonos vemos lo siguiente.

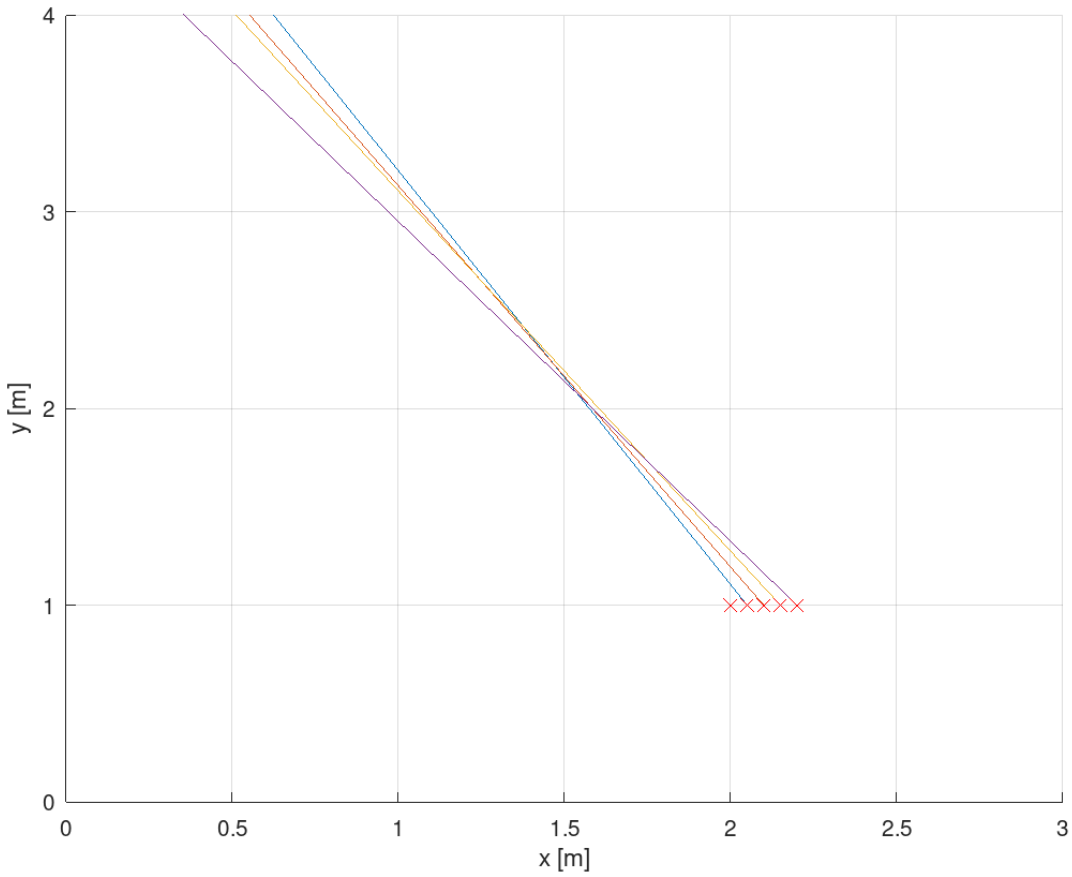

Figura 29: Disposición de los micrófonos y dirección de la que proviene el sonido para cada uno

Como era de esperar con los retardos obtenidos se da el caso que las cuatro rectas se intersectan cercanamente entre sí. Nuevamente hay que aclarar que estos resultados no deben ilusionarnos ya que son producto de alteraciones aleatorias provenientes del ruido. Aún así podríamos en este gráfico estimar (una vez más, con mucho error) una posición de la fuente tal y como se muestra en el siguiente esquema.

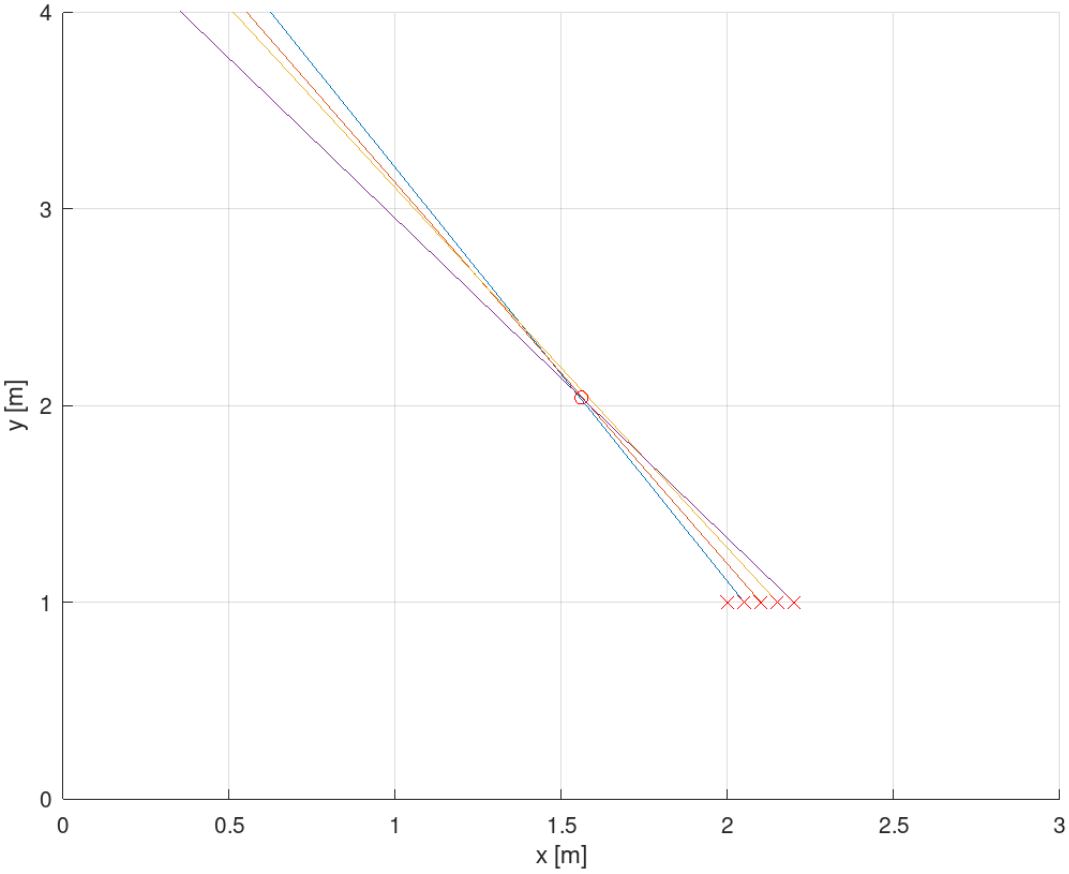

Figura 30: Disposición de los micrófonos y de la fuente de sonido

La posición de la fuente estimada es aproximadamente a 1,56 metros en **x** y 2,04 metros en **y** respecto del origen.

Ejercicio 6

Modificar el algoritmo de estimación de los retardos realizando un sobremuestreo de las señales IDFT(GPH[k]). Repetir los ejercicios 2 y 4.

El sobremuestreo pedido nos permitirá determinar con una mejor precisión los retardos entre los distintos micrófonos. Si originalmente teníamos una precisión de aproximadamente 20 us en el retardo entonces haciendo un sobremuestreo con un factor de **L** (es decir, agregando **L** - 1 ceros entre las muestras e interpolando mediante un filtro pasabajos adecuado) estaremos teniendo una precisión de $\frac{20}{L}$ us. Un factor de **L** razonable sería 10, de manera tal de pasar a tener una precisión de unos 2 us por muestra. En cuanto a la elección del tipo de filtro pasabajos que utilizaremos para interpolar es evidente que necesitamos un filtro tipo **FIR**. Siendo que la idea de sobremuestrear a la señal IDFT(GPH[k]) es conseguir una mejor precisión en su retardo estaríamos equivalentemente dependiendo de la precisión en la fase de su espectro. Aplicar un filtro tipo **IIR** resultaría en una distorsión no lineal del espectro de la señal lo cual afectaría el retardo que estaríamos obteniendo de forma incorregible. Aplicar un filtro **FIR** en cambio nos permitiría corregir el desfase introducido por el filtro adecuadamente mediante un simple desplazamiento en el tiempo y conseguir de esta manera exactamente la misma señal IDFT(GPH[k]) original pero con **L** - 1 muestras nuevas entre cada muestra original, de manera que ganamos resolución a la hora de estimar el retardo.

Calculando entonces los nuevos retardos medios habiendo realizado un sobremuestreo con un factor de **L** = 10 y utilizando el mismo ventaneo que en ejercicios anteriores (ventana de 20.000 muestras y un overlapping de 10.000 muestras) obtenemos los siguientes resultados.

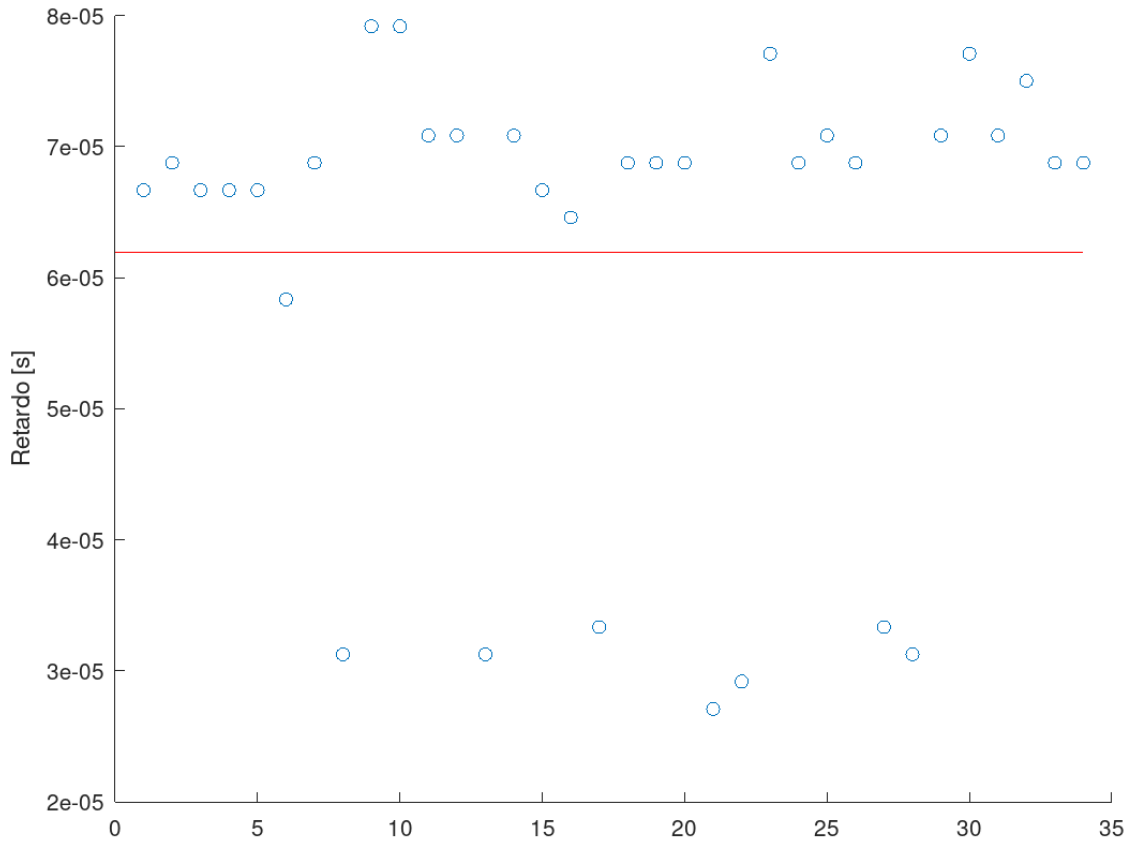

Figura 31: Retardos obtenidos para el micrófono 2 respecto al micrófono 1

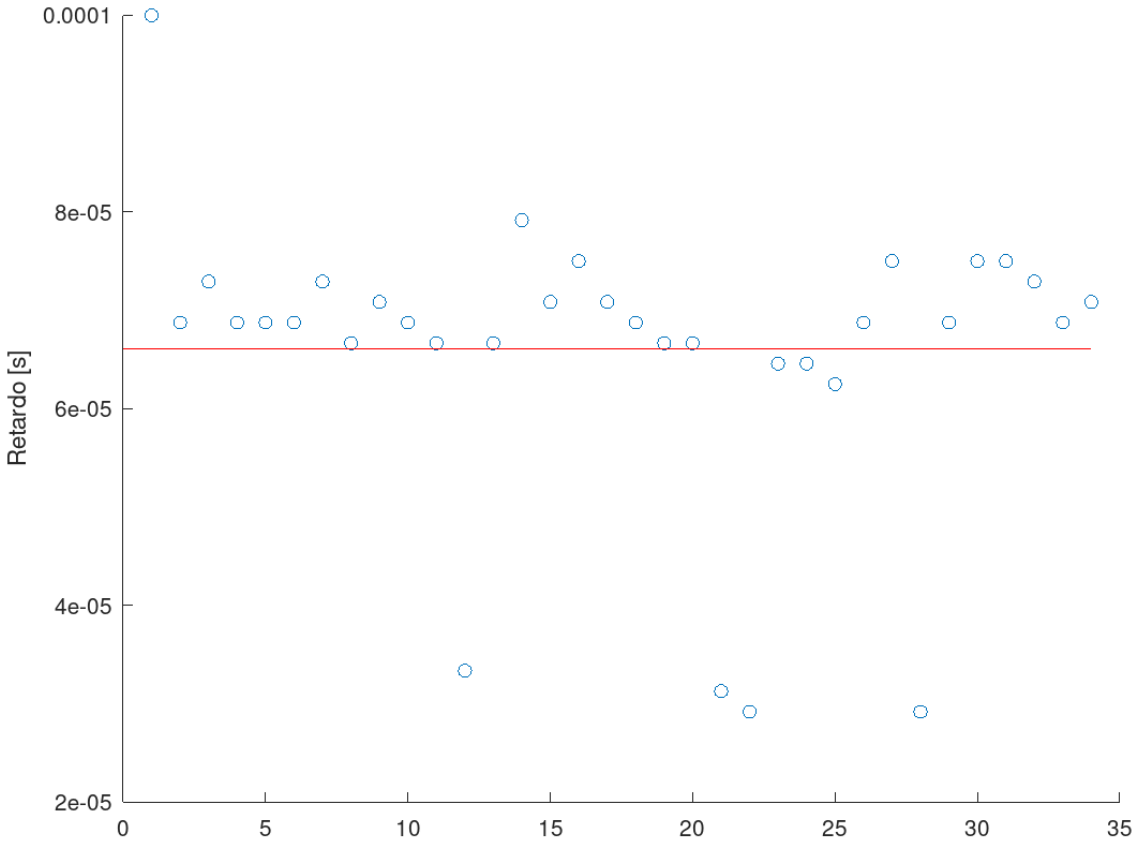

Figura 32: Retardos obtenidos para el micrófono 3 respecto al micrófono 2

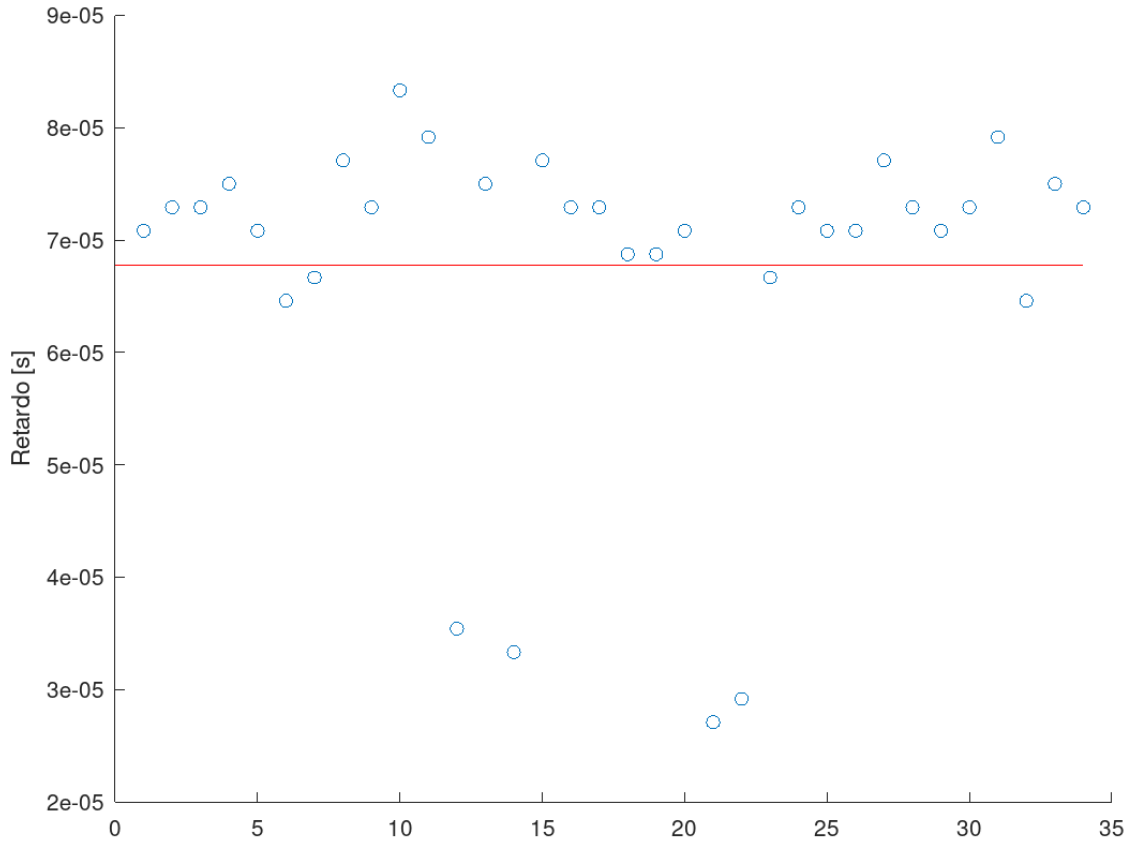

Figura 33: Retardos obtenidos para el micrófono 4 respecto al micrófono 3

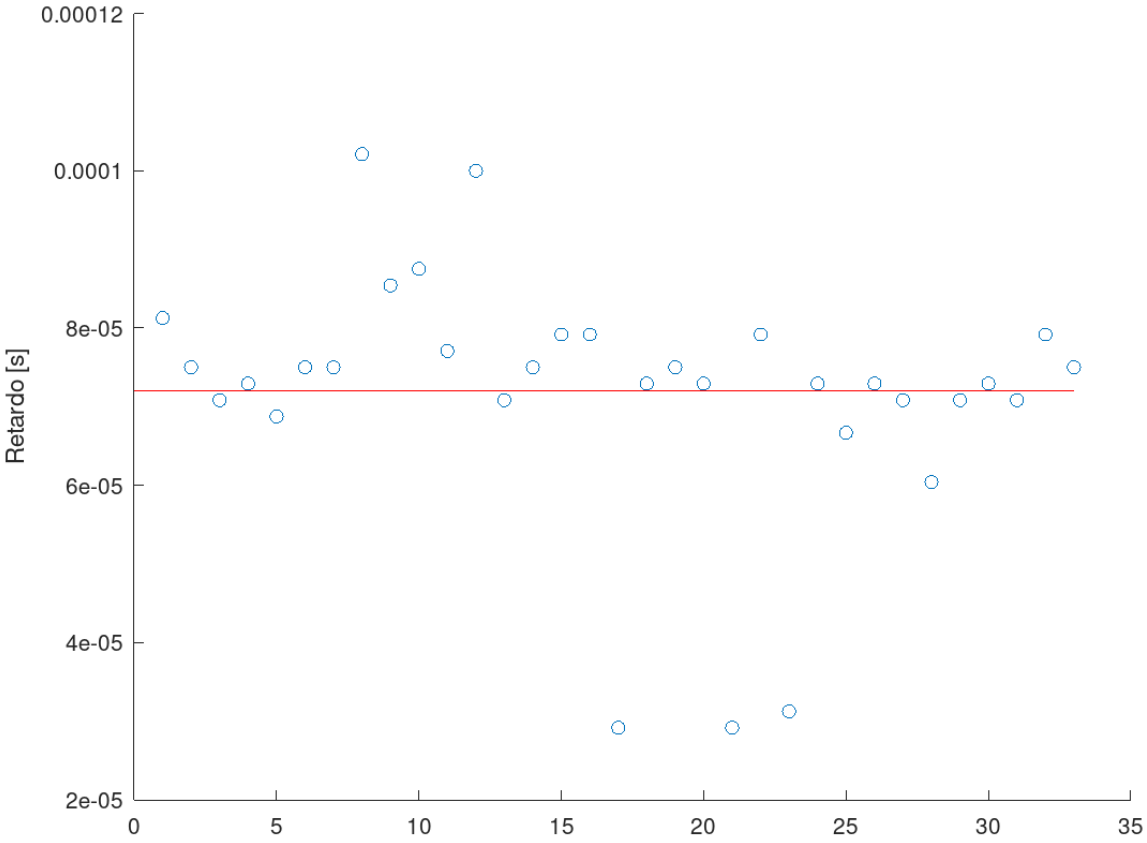

Figura 34: Retardos obtenidos para el micrófono 5 respecto al micrófono 4

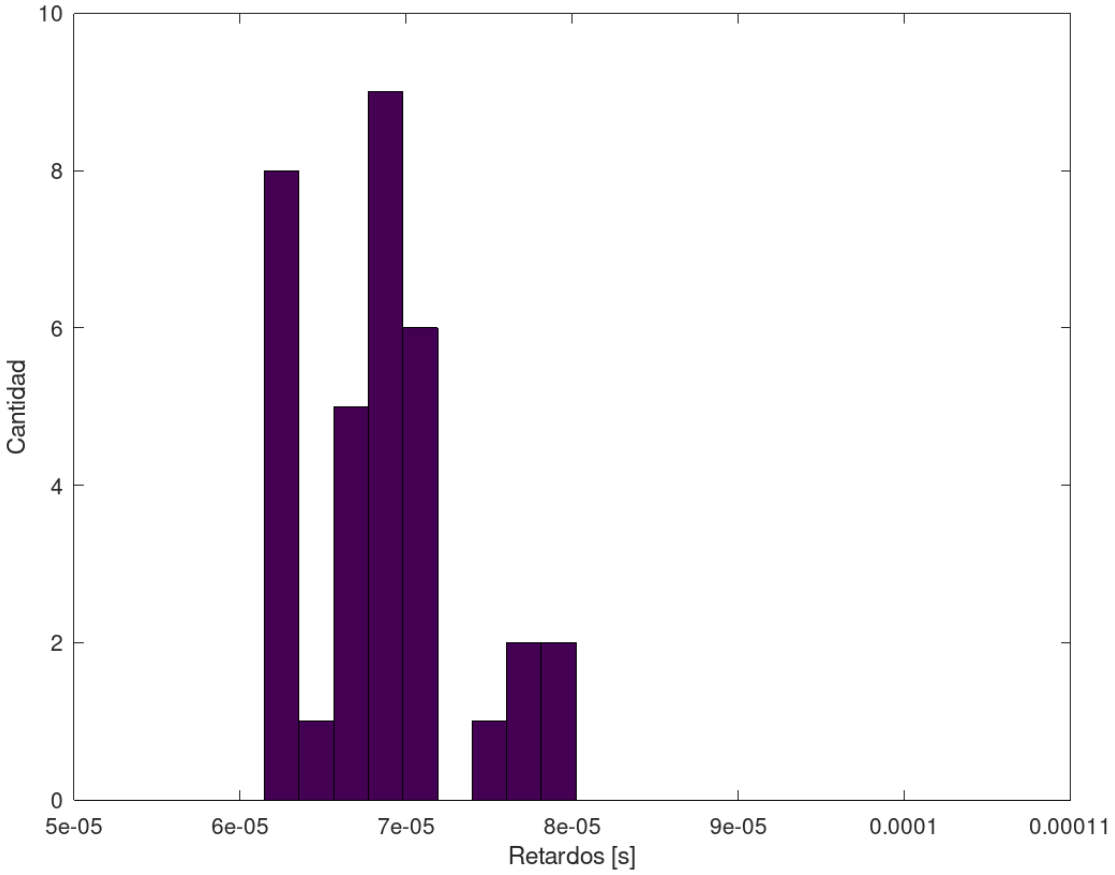

Figura 35: Histograma de los retardos del micrófono 2 respecto al micrófono 1

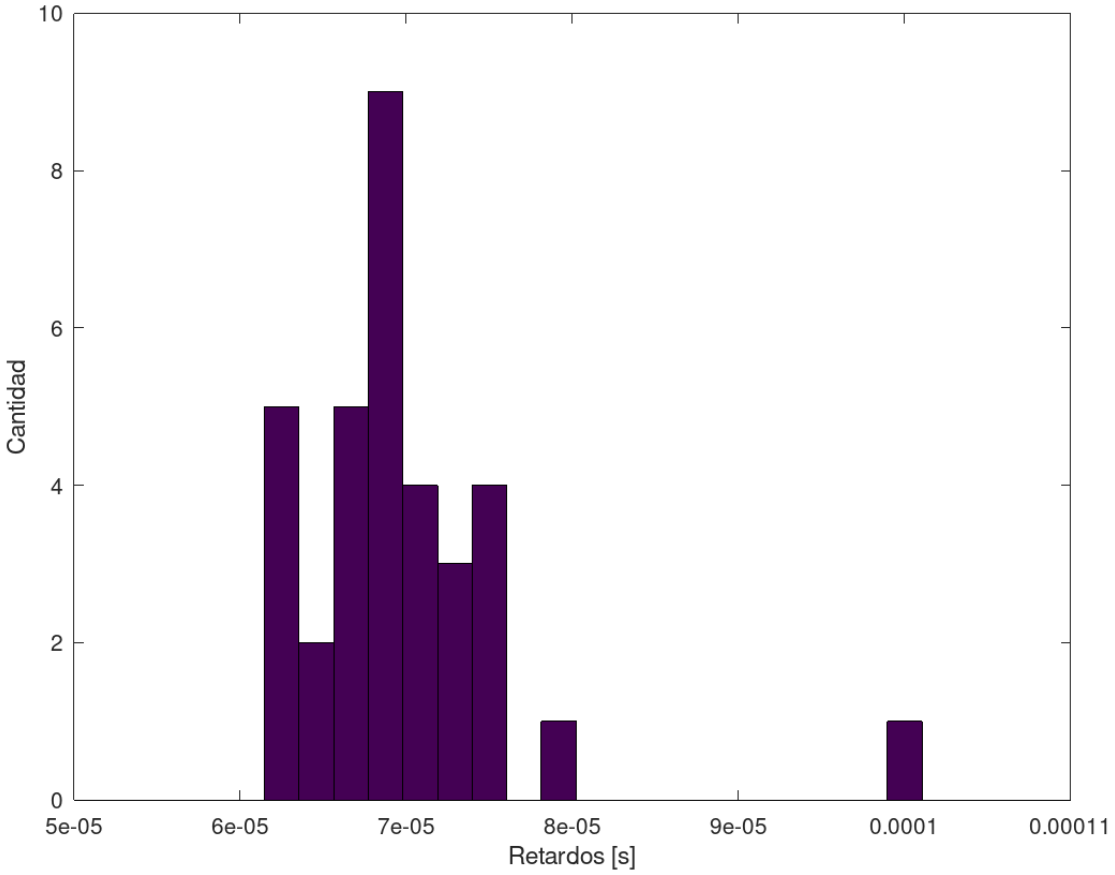

Figura 36: Histograma de los retardos del micrófono 3 respecto al micrófono 2

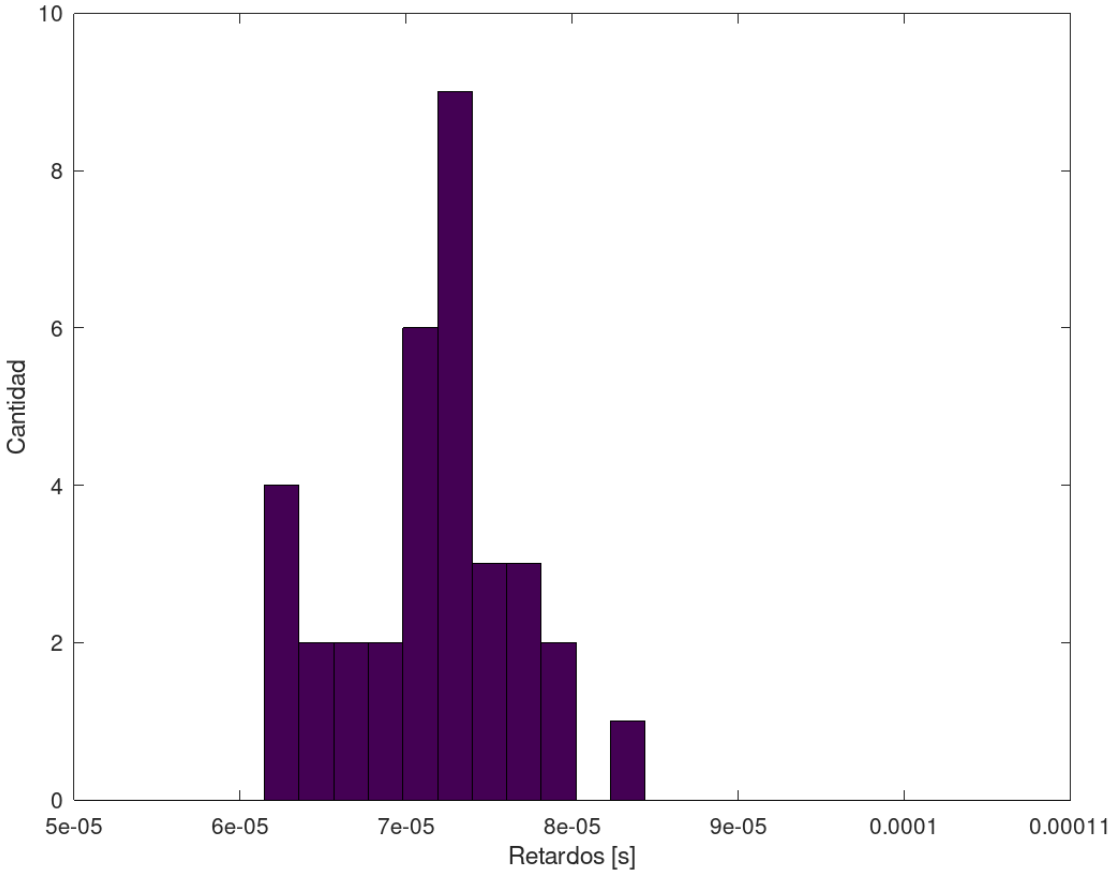

Figura 37: Histograma de los retardos del micrófono 4 respecto al micrófono 3

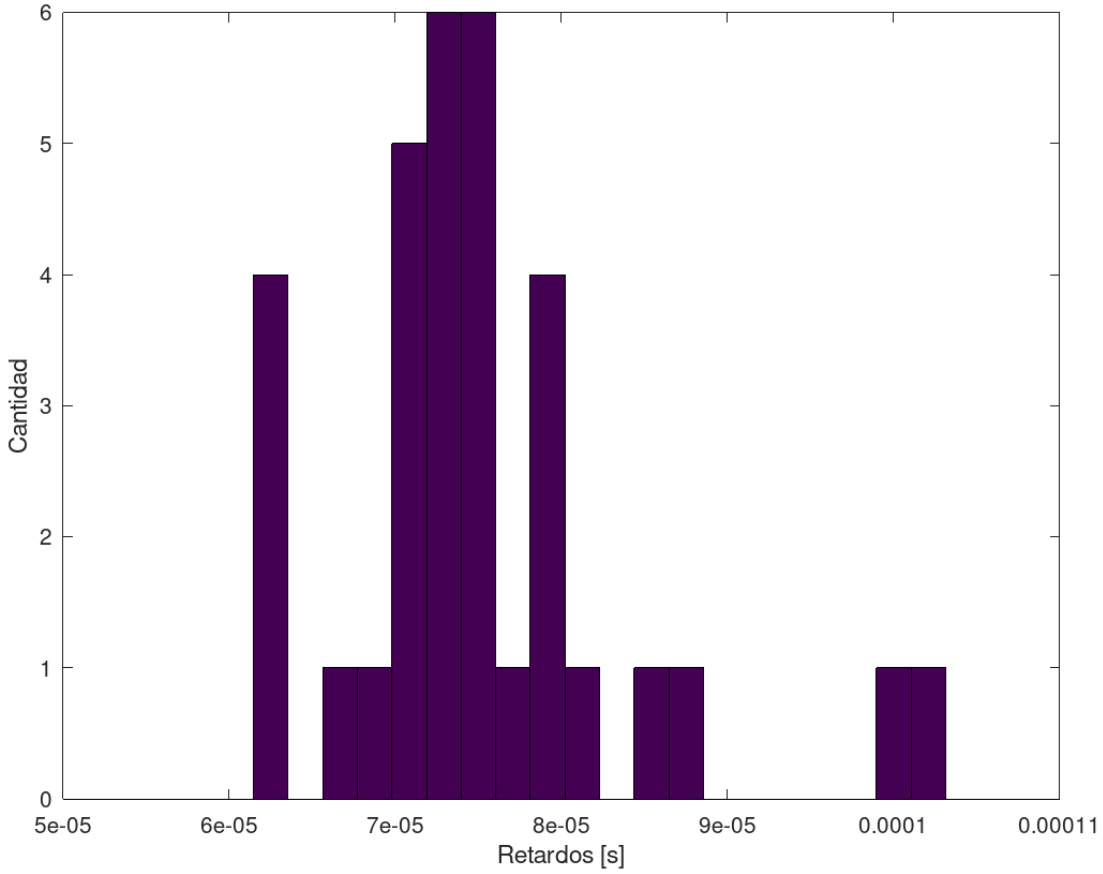

Figura 38: Histograma de los retardos del micrófono 5 respecto al micrófono 4

Los retardos medios obtenidos fueron:

- **Micrófono 2**: 61,9 us (respecto al Micrófono 1)
- **Micrófono 3**: 66,2 us (respecto al Micrófono 2)
- **Micrófono 4**: 67,8 us (respecto al Micrófono 3)
- **Micrófono 5**: 72,0 us (respecto al Micrófono 4)

De todas formas, aún habiendo ganado 10 veces más precisión mediante nuestro sobremuestreo, no podemos confiarnos de estos resultados. Al seguir teniendo el ruido introducido en las señales cualquier resultado que estemos obteniendo seguirá teniendo un considerable error. Será necesario filtrar este ruido para recién entonces poder aprovechar el sobremuestreo y conseguir estimaciones de los retardos más confiables. Aún así, si graficamos las rectas correspondientes y estimamos una posición de la fuente en base a ellas obtendríamos lo siguiente.

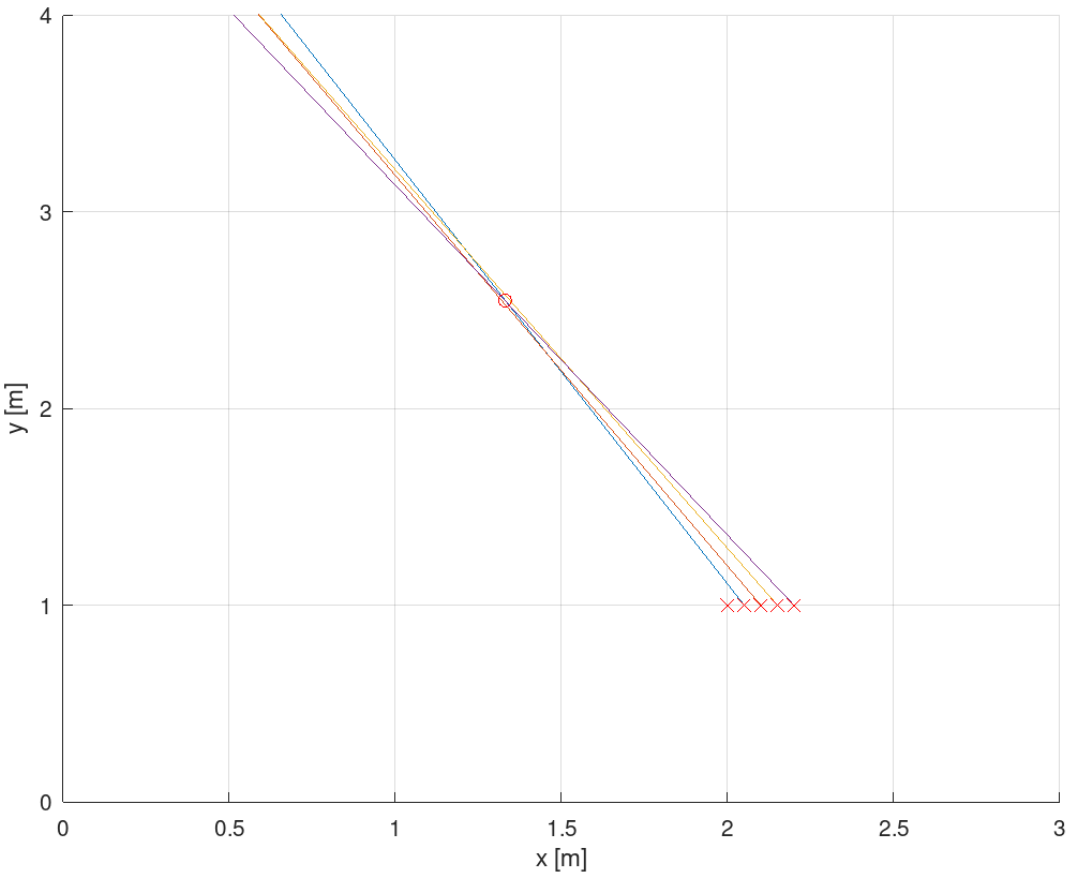

Figura 39: Disposición de los micrófonos y de la fuente de sonido

La posición de la fuente estimada es aproximadamente a 1,33 metros en **x** y 2,55 metros en **y** respecto del origen.

Ejercicio 7

Utilizando los espectros obtenidos en el ejercicio 5, diseñar un filtro pasabanda con el objetivo de reducir el ruido y por lo tanto mejorar la estimación de los retardos. Graficar la respuesta en frecuencia y diagrama de polos y ceros del filtro obtenido. Repetir los ejercicios 2 y 4.

Veamos nuevamente el espectrograma del **Ejercicio 5** para analizar el ancho de banda de la señal que nos interesa.

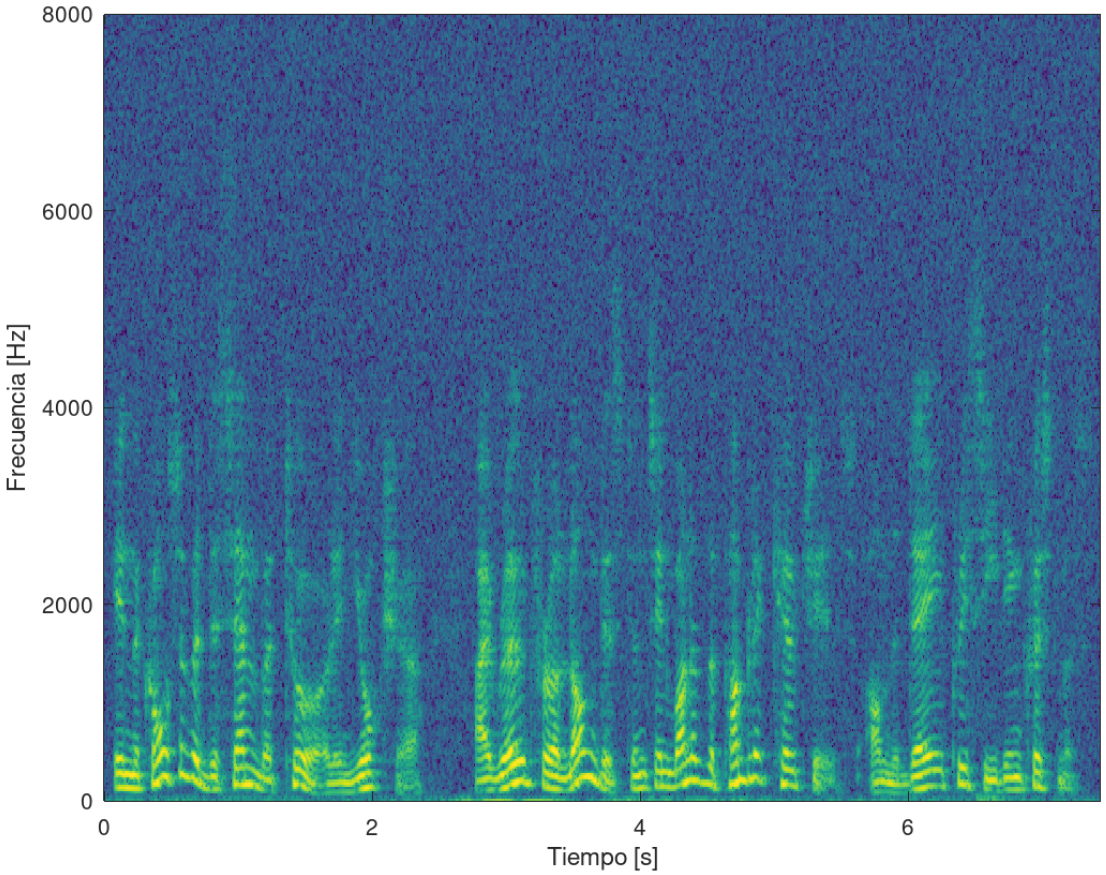

Figura 40: Espectrograma del Micrófono 1 (con ruido) entre los 0 Hz y 6500 Hz

Observando detenidamente el mismo podemos estimar un ancho de banda de la señal de aproximadamente entre los 4 KHz y 5 KHz. Si bien cortar a los 4 KHz sería válido, todavía podemos distinguir algunas frecuencias cercanas a los 5 KHz por lo que seremos un poco generosos por si acaso y filtraremos hasta los 5 KHz. El filtro en cuestión será entonces un filtro pasabajos con frecuencia de corte $\mathbf{Fc} = 5$ KHz debido a que como las frecuencias más bajas se encuentran alrededor de los 100 Hz resulta indiferente a un pasabanda que filtre frecuencias por debajo de esa (el ruido entre los 0 y 100 Hz resulta despreciable). El filtro utilizado es de tipo **FIR** con 150 coeficientes, si bien un filtro **IIR** sería igualmente válido debido a que si bien el desfase introducido no sería lineal este afectaría de igual forma a todas las señales, por lo que el retardo entre ellas no se vería afectado.

A continuación se presenta el diagrama de polos y ceros del filtro.

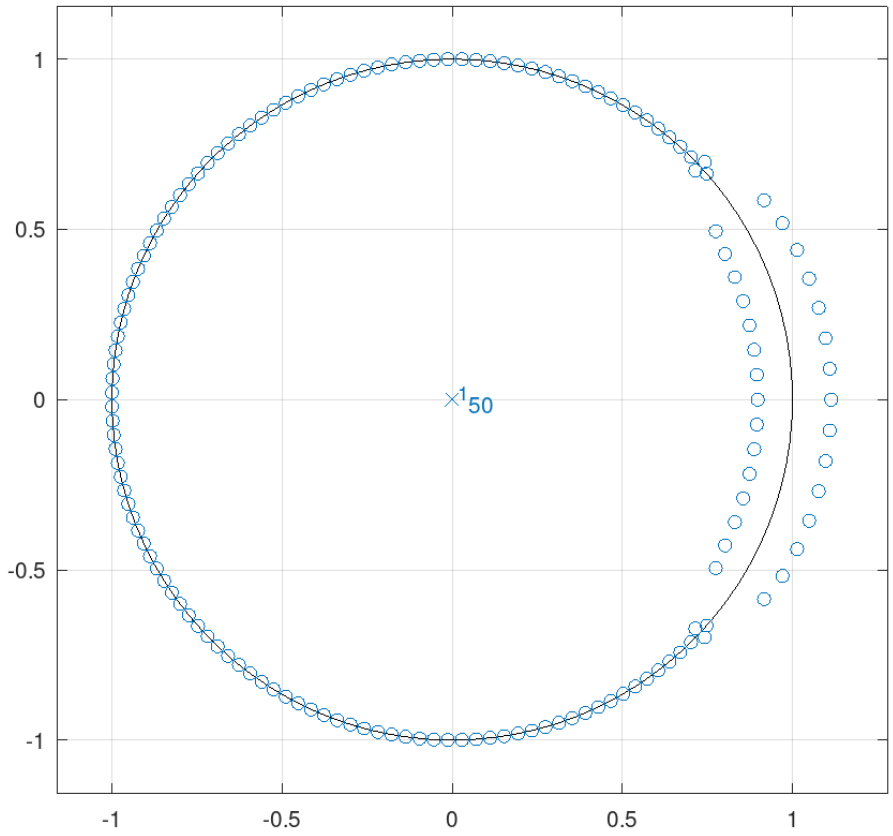

Figura 41: Diagrama de polos y ceros del filtro

Nótese que como utilizamos un filtro **FIR** todos los polos se encuentran en el cero, y dado que nuestro filtro tiene 150 coeficientes vemos que el orden del polo es 150, tal y como era de esperar. Vemos en el diagrama que tenemos la mayoría de los ceros sobre la circunferencia de módulo 1. Estos ceros son los que corresponden a las frecuencias que estamos filtrando en el filtro, que habíamos dicho que filtraba a partir de los 5 KHz hasta los 24 KHz (la mitad de la frecuencia de muestreo). Si entendemos los 24 KHz como π entonces $\frac{5KHz}{24KHz} * \pi \approx 0,2 * \pi$. Vemos que entonces en el gráfico tenemos justamente los ceros sobre la circunferencia a partir de aproximadamente un cuarto de pi, lo cual coincide entonces con lo esperado. Los ceros que se encuentran fuera de la circunferencia son aquellos asociados con las frecuencias que no estamos filtrando, es decir, aquellas entre 0 Hz y 5 KHz. La simetría presente entre los ceros no es casualidad sino que es característico de los filtros de fase lineal, en los cuales sucede que $H(z) = \pm z^{-N} \cdot H(\frac{1}{z})$ donde si z_0 es un cero del filtro H(z) entonces la igualdad establece que $\frac{1}{z_0}$ también lo es. Justamente esa propiedad es la que genera esa paridad entre los ceros observada (por ejemplo el cero en $z = j$ tiene un cero recíproco en $z = -j$).

La respuesta en frecuencia del filtro es la siguiente.

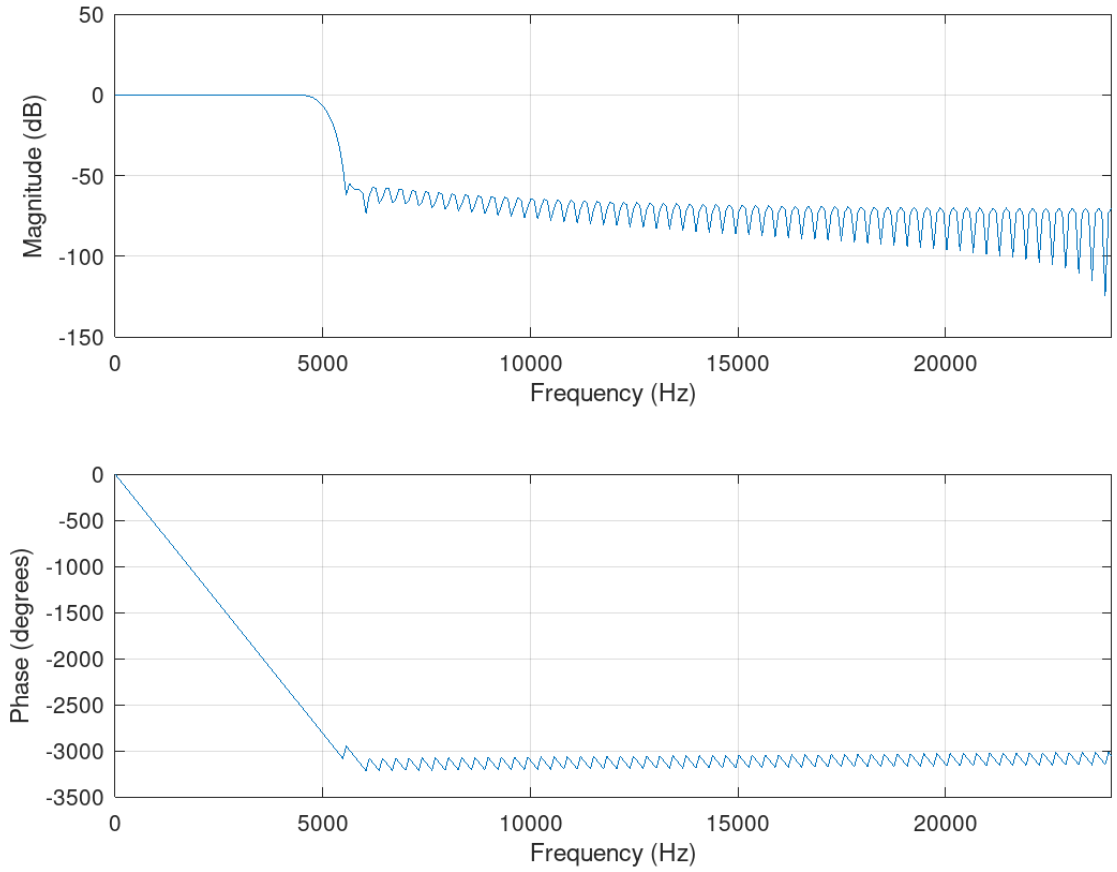

Figura 42: Respuesta en frecuencia del filtro

Vemos que la respuesta es coherente con el filtro diseñado. Aquellas frecuencias que se encuentran por debajo de los 5 KHz no presentan ganancia (se mantienen igual a la salida que a la entrada en potencia) mientras que de allí en adelante se ven totalmente atenuadas (una ganancia de -70 dB prácticamente elimina por completo esa frecuencia). Por el lado de la fase del filtro vemos que en el rango en el que dejamos pasar las frecuencias se tiene un desfase lineal (ventaja del tipo de filtro utilizado), el cual será fácilmente corregible en el tiempo mediante un simple desplazamiento de la señal. Finalmente, observemos el espectrograma de la señal del Micrófono 1 una vez que fue filtrada por nuestro filtro para corroborar que funcione correctamente.

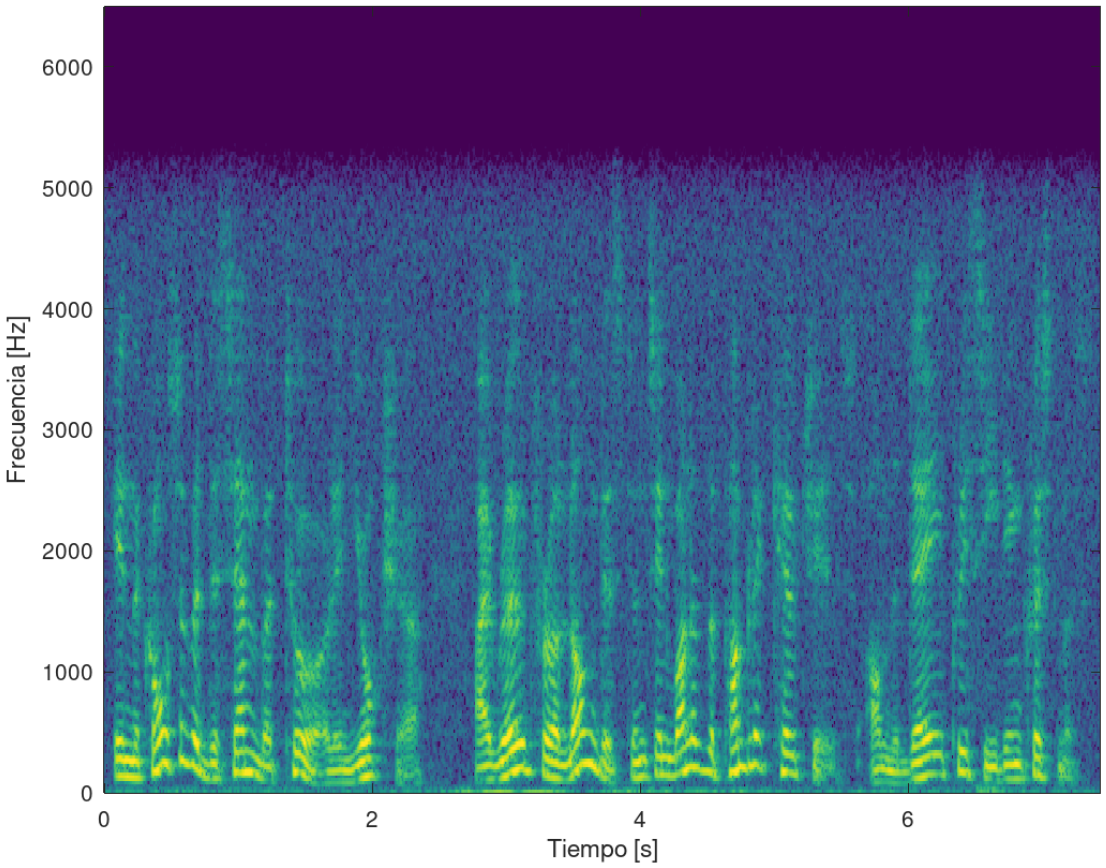

Figura 43: Espectrograma del Micrófono 1 (con ruido filtrado) entre los 0 Hz y 6500 Hz

Vemos que efectivamente se filtraron aquellas frecuencias por encima de los 5 KHz. El espectrograma ahora muestra que esas frecuencias tienen una potencia despreciable mientras que las frecuencias que dejamos pasar se mantienen igual que antes de haber filtrado la señal. En rigor vemos que algunas frecuencias cercanas a la frecuencia de corte no se encuentran tan atenuadas. Esto podría corregirse con un filtro de mayor orden todavía pero realmente no resultaría práctico incrementar tanto el orden para filtrar unas pocas frecuencias que no introducen diferencias apreciables en los cálculos.

A continuación se presentan los resultados de los retardos obtenidos con un ventaneo de 20.000 muestras y un overlapping de 10.0000 muestras.

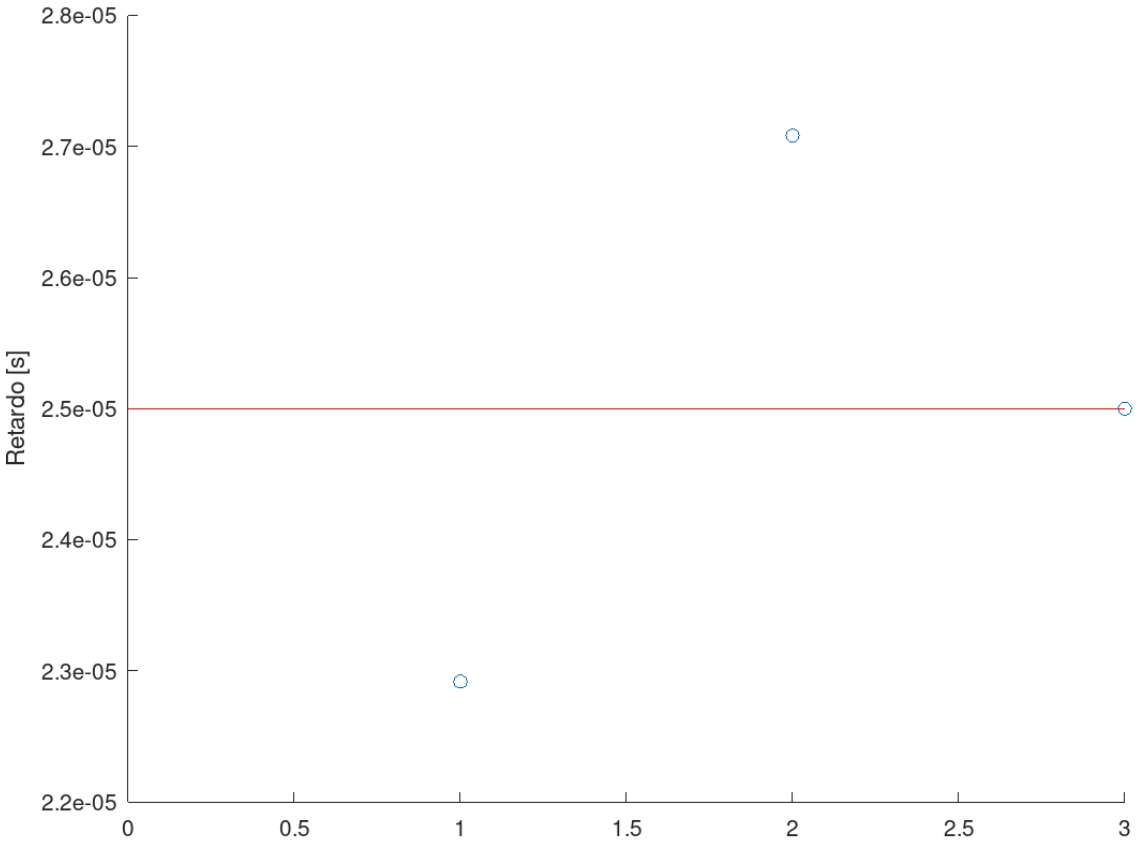

Figura 44: Retardos obtenidos para el micrófono 2 respecto al micrófono 1

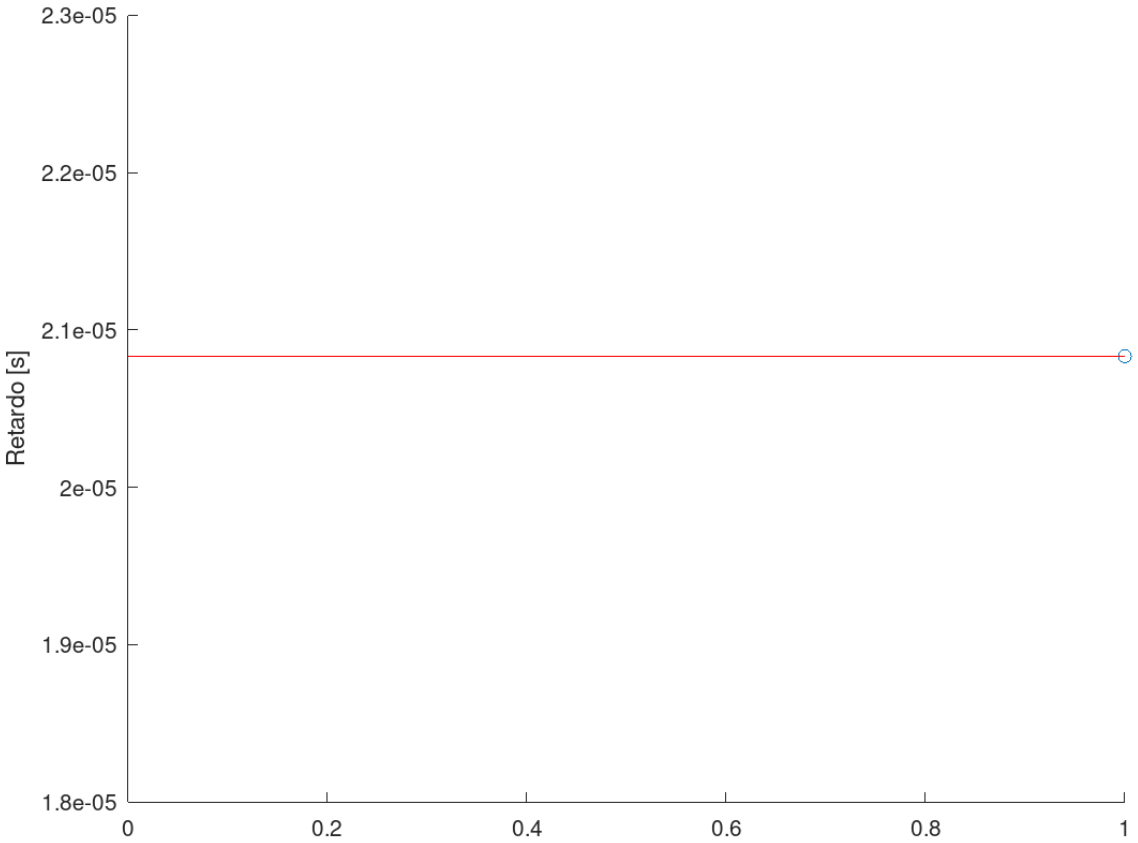

Figura 45: Retardos obtenidos para el micrófono 3 respecto al micrófono 2

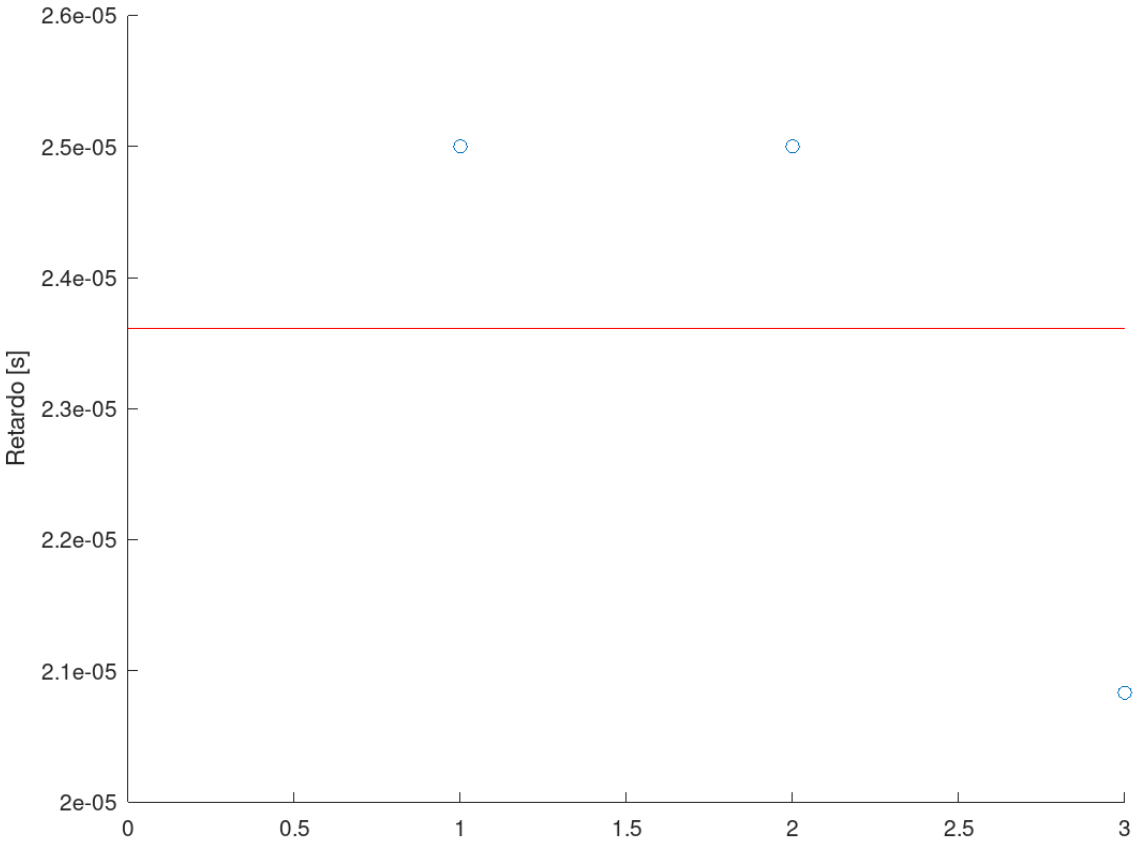

Figura 46: Retardos obtenidos para el micrófono 4 respecto al micrófono 3

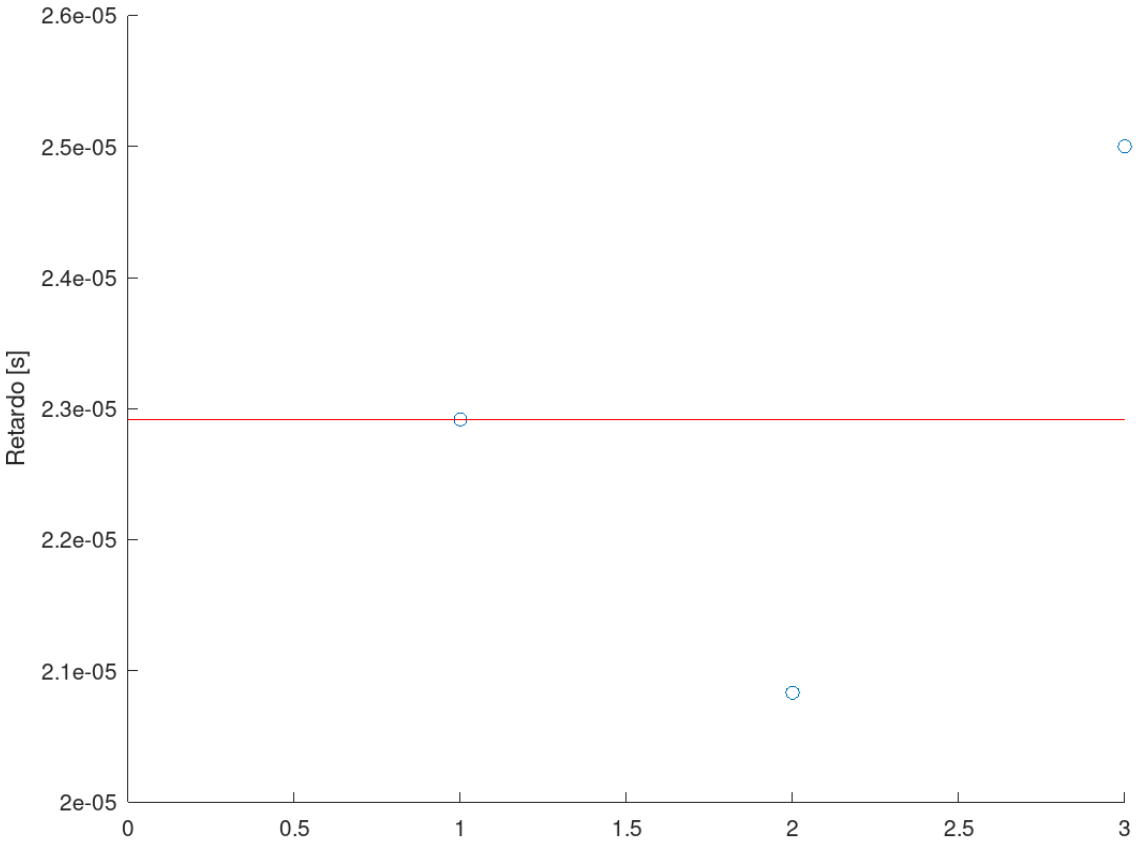

Figura 47: Retardos obtenidos para el micrófono 5 respecto al micrófono 4

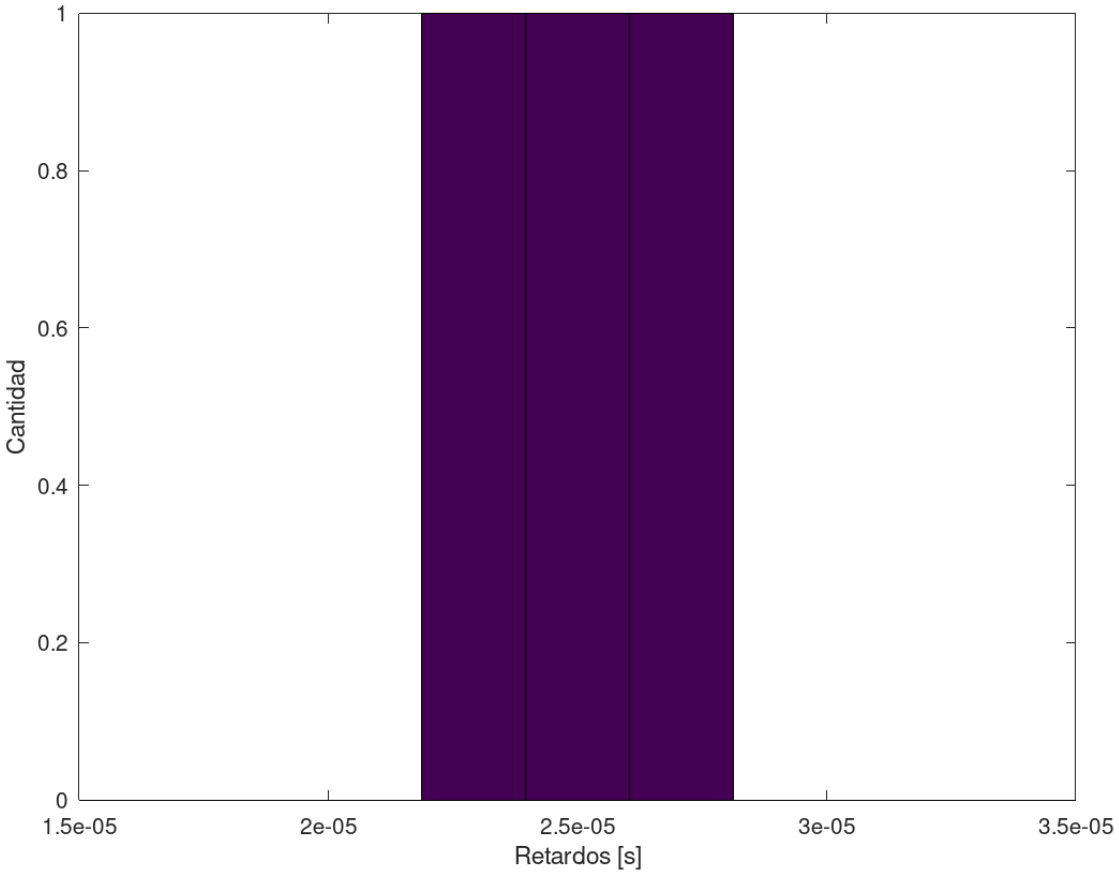

Figura 48: Histograma de los retardos del micrófono 2 respecto al micrófono 1

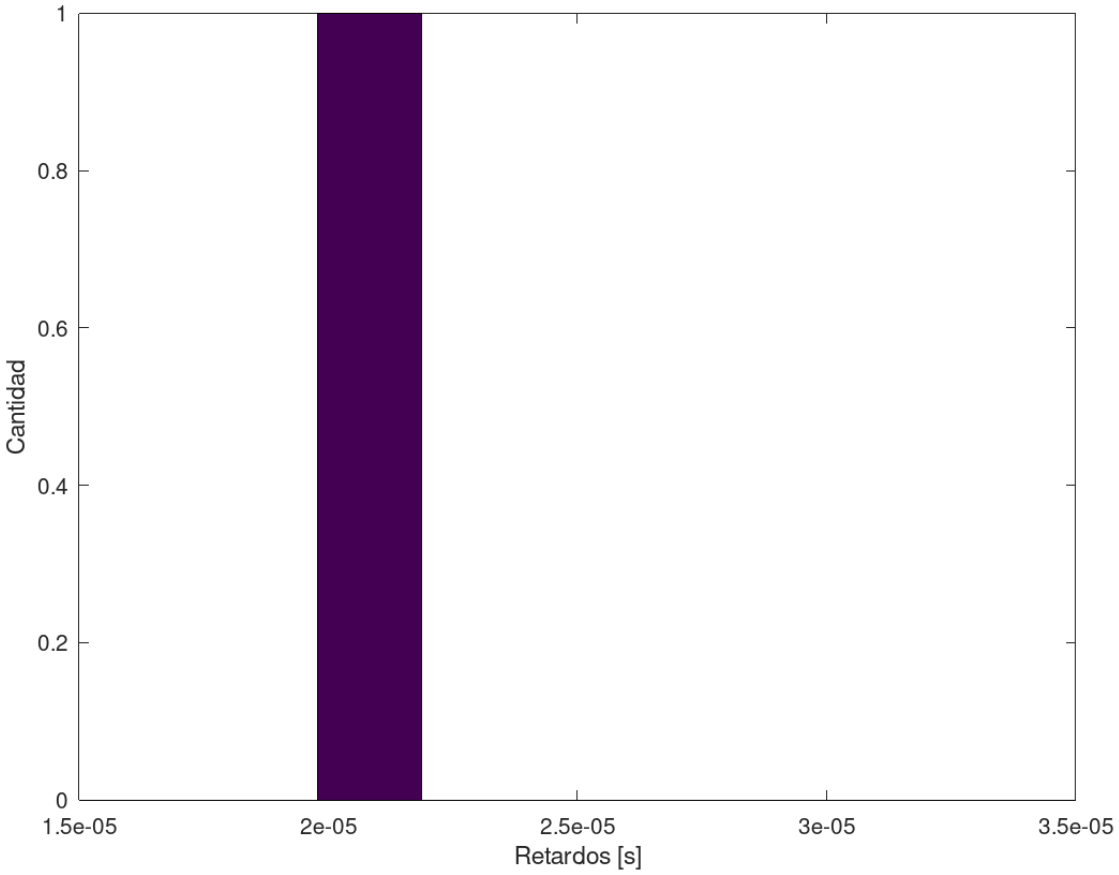

Figura 49: Histograma de los retardos del micrófono 3 respecto al micrófono 2

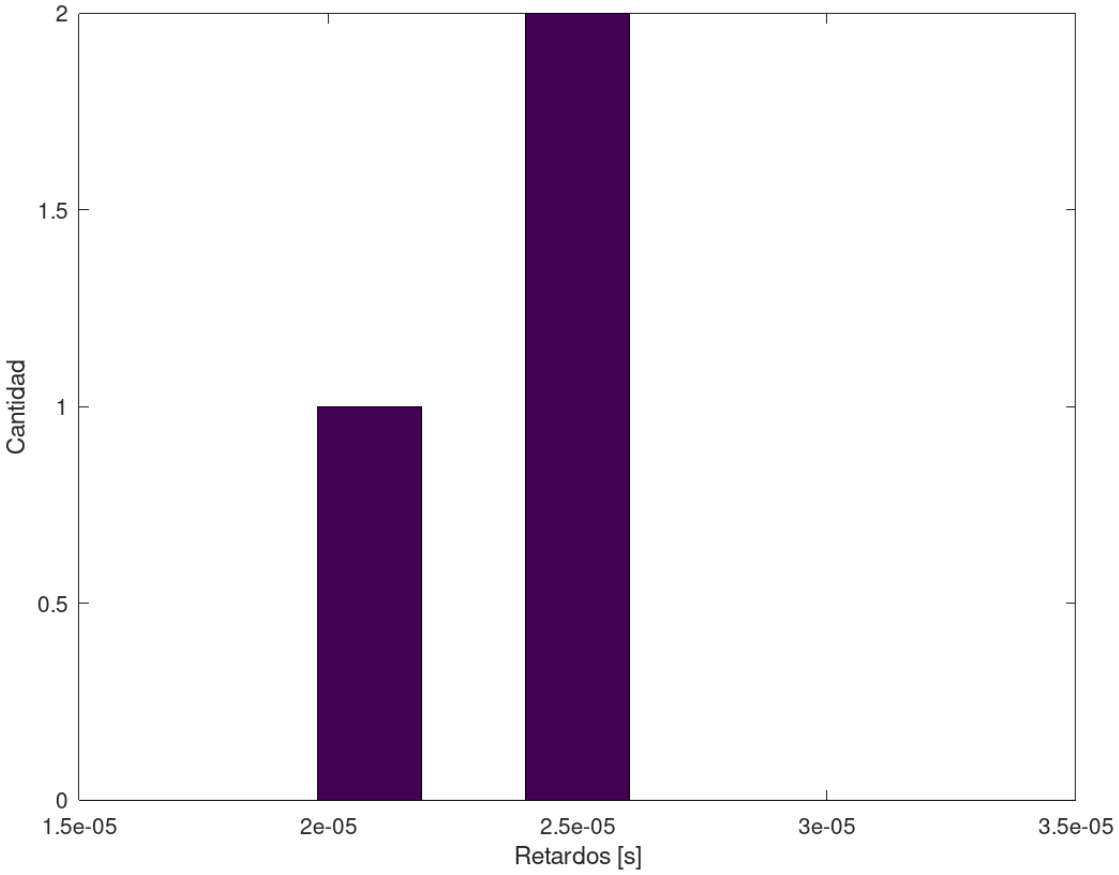

Figura 50: Histograma de los retardos del micrófono 4 respecto al micrófono 3

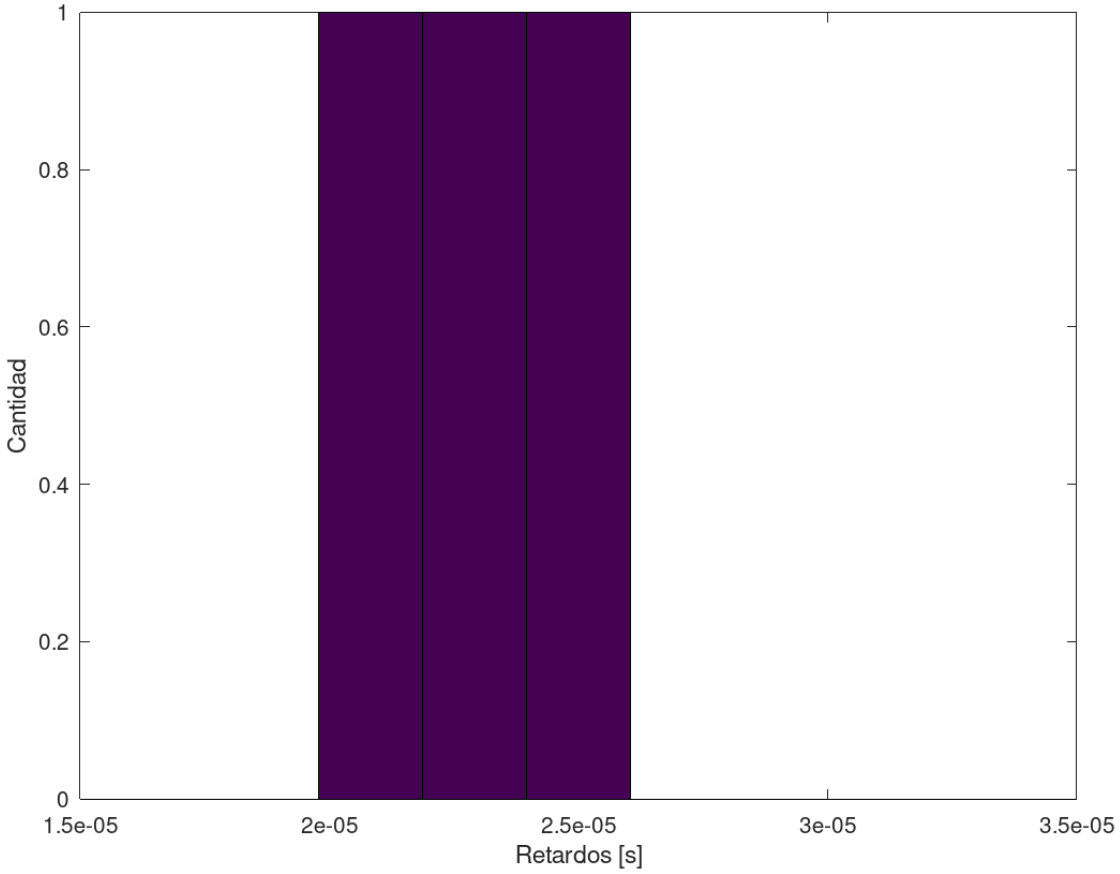

Figura 51: Histograma de los retardos del micrófono 5 respecto al micrófono 4

Sorprendentemente vemos que estamos obteniendo resultados extremadamente distintos a todos los anteriores. En primer lugar estamos obteniendo retardos cercanos a los 20 us cuando habíamos observado en todas las anteriores mediciones que los retardos se encontraban entre los 60 y 80 us. En segundo lugar vemos que son extremadamente pocos los retardos válidos con los que nos estamos quedando. En general nos quedamos con únicamente 3 retardos, pero entre el micrófono 3 y el micrófono 2 tan sólo nos quedamos con uno. Recordemos que los retardos que consideramos válidos en el ventaneo son aquellos en los que obtenemos que existe un retardo desde el micrófono **i**-ésimo + 1 con respecto al micrófono **i**-ésimo. Si algún ventaneo nos retorna un retardo opuesto o que no hay retardo entre ambas señales entonces lo descartamos porque es una medición absurda. En este caso casi todas las mediciones retornan que no hay retardo entre las señales por lo que no las estamos considerando, pero esto nos deja únicamente con unas muy pocas muestras a considerar. Lógicamente realizar el promedio con esos retardos nulos nos terminará llevando el retardo entre señales a valores muy cercanos a 0 (del orden de los 2 us) lo cual es incluso más ilógico. Estos resultados se repiten independientemente del tamaño de la ventana y el overlapping, donde para algunos ventaneos incluso obtenemos que todos los retardos son 0 us. Totalmente absurdo.

Para estos resultados en cuestión, los valores medios de los retardos obtenidos fueron los siguientes.

- **Micrófono 2**: 25,0 us (respecto al Micrófono 1)
- **Micrófono 3**: 20,8 us (respecto al Micrófono 2)
- **Micrófono 4**: 23,6 us (respecto al Micrófono 3)

- **Micrófono 5**: 22,9 us (respecto al Micrófono 4)

Vemos que los retardos obtenidos no siguen tampoco una secuencia coherente por lo que ya podemos prever que la posición de la fuente tampoco será realmente estimable. Aún así, con los resultados claramente erróneos, lo mejor que podemos estimar es la siguiente posición.

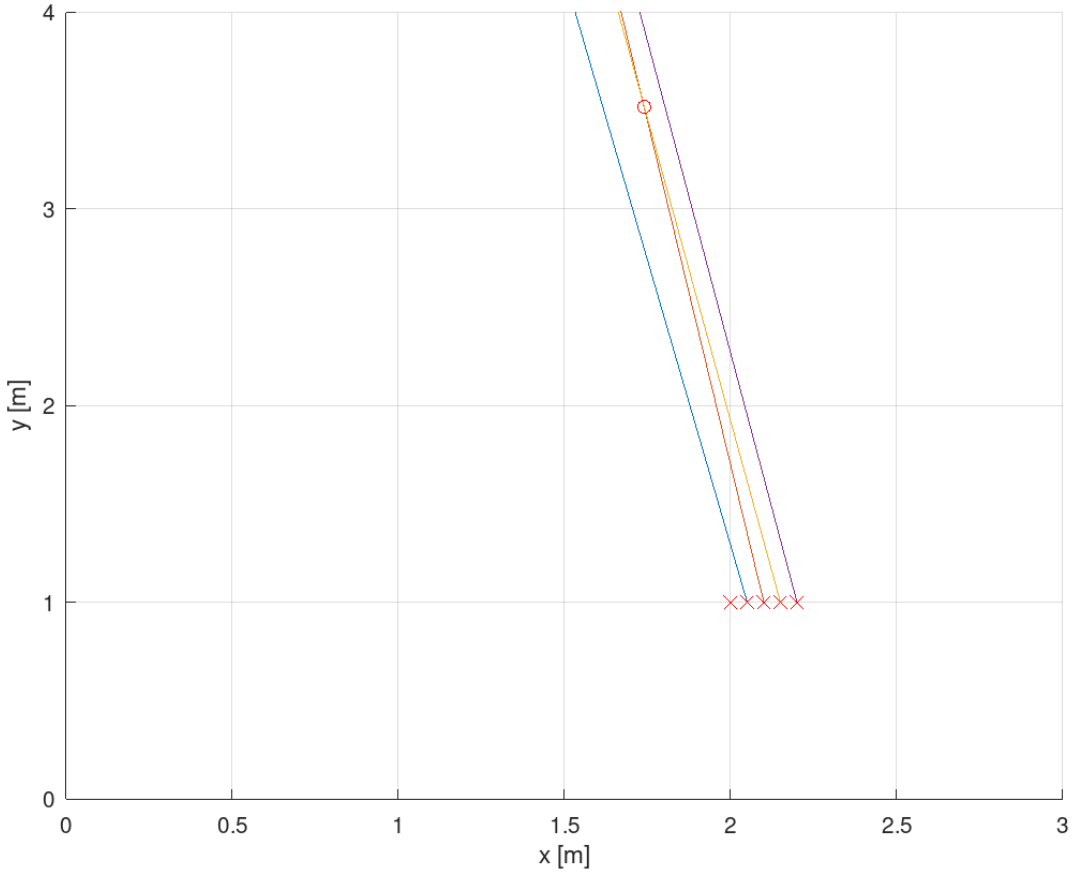

Figura 52: Disposición de los micrófonos y de la fuente de sonido

La posición de la fuente estimada es aproximadamente a 1,74 metros en **x** y 3,52 metros en **y** respecto del origen.

Si bien no fue posible confirmar con seguridad la causa del error, debido a que el filtro no aparenta tener ningún problema se descartó el mismo como posible causante. Lo curioso sin embargo es que el método de correlación cruzada utilizado por ejemplo en el **Ejercicio 3** sigue arrojando los mismo resultados luego de aplicarle el filtro. Debido a esto se investigó más detalladamente el método de GCC-PHAT y se descubrió que lo que aparenta generar este error proviene de la normalización de la Gph al dividirse por el módulo del producto de las DFT de las señales. Aparentemente este algoritmo es inestable para valores muy pequeños de las DFT, dado que al tratarse de números tan chicos una pequeña variación en los mismos producto de redondeos puede producir muchísimo error en los valores obtenidos. Esto resultó evidente cuando se sobremuestreo las señales originales sin ruido y se intentó estimar los retardos tanto con el método de correlación cruzada como con el de GCC-PHAT. El primer método estimó los retardos prácticamente a la perfección mientras que el segundo daba resultados incoherentes como los vistos en este ejercicio. Al eliminar la normalización del método de GCC-PHAT (no realizar la división en Gph) se obtuvieron exactamente los mismos resultados que en la

correlación cruzada (lo esperable ya que al no normalizar lo que tendríamos es la transformada de la correlación cruzada).

Los siguientes posts mencionan exactamente el mismo problema y hacen referencia al problema de estabilidad del algoritmo de GCC-PHAT al implementarse de la forma que está enunciado en el trabajo práctico.

- https://dsp.stackexchange.com/questions/31956/gcc-phat-generalized-cross-correlation-matlab
- https://www.dsprelated.com/thread/2149/generalized-cross-correlation-method-not-producing-desired-results
- https://stackoverflow.com/questions/42403722/gcc-phat-on-matlab-still-challenging

Si se agrega un *epsilon* de aproximadamente $\epsilon \approx 5 * 10^{-4}$ al divisor los resultados mejoran considerablemente, obteniéndose valores en el rango de los 60 us. A pesar de esto dichos valores tampoco son ideales (a veces no varían entre micrófonos o dependen mucho del *epsilon*) por lo que no se modificó el algoritmo provisto por la cátedra.

Con el objetivo de conseguir resultados coherentes en los ejercicios siguientes se optó por realizar un sobremuestreo con un factor de $\mathbf{L} = 10$ sobre las señales luego de haber filtrado el ruido con el filtro propuesto en este ejercicio para finalmente aplicar el método de correlación cruzada (de la misma forma que se había hecho en el **Ejercicio 3**) y obtener buenos resultados. Dado que filtramos el ruido en aquellas frecuencias que no componen a nuestra señal de interés y que además realizamos un sobremuestreo que nos da una resolución de unos 2 us por muestra, los siguientes resultados obtenidos son los mejores.

- **Micrófono 2**: 66,7 us (respecto al Micrófono 1)
- **Micrófono 3**: 68,8 us (respecto al Micrófono 2)
- **Micrófono 4**: 70,8 us (respecto al Micrófono 3)
- **Micrófono 5**: 72,9 us (respecto al Micrófono 4)

La posición de la fuente estimada es la siguiente.

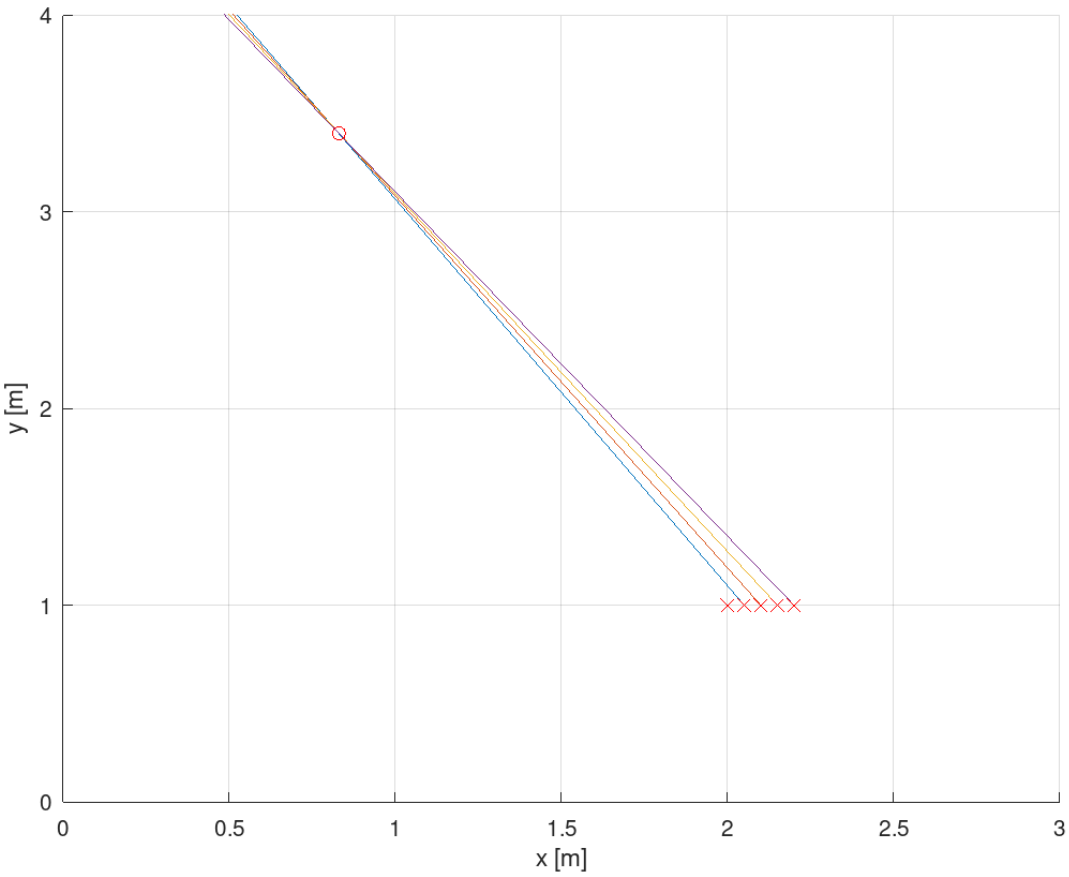

Figura 53: Disposición de los micrófonos y de la fuente de sonido

La posición de la fuente estimada es aproximadamente a 0,83 metros en **x** y 3,40 metros en **y** respecto del origen.

Ejercicio 8

Alinear las señales utilizando los retardos estimados en el ejercicio anterior y obtener una nueva señal promediada. Escuchar ambas versiones. Tener en cuenta que es necesario sobremuestrear las señales con la misma tasa del ejercicio 6 de modo de obtener una mejor alineación.

Al comparar ambas versiones se aprecia una notable diferencia en la intensidad del ruido, escuchándose más atenuado en la señal promediada que en aquella que tan solo tenía aplicado el filtro pasabanda. Estos resultados son esperables ya que al promediar todas las señales estamos reduciendo el ruido debido a que este ruido es blanco. Este tipo de ruido se distribuye uniformemente en todas las frecuencias por lo que al promediarlo se "cancelaría". En la práctica lo que sucede es que si bien no se anula en su totalidad sí disminuye considerablemente. Se adjuntan en este trabajo los audios "audio_promediado.wav" y "audio_no_promediado.wav" para comparar.

Ejercicio 9

Graficar los espectrogramas de la señal del micrófono 1, antes y después del filtrado del ejercicio 8. Identificar los efectos del filtrado.

Los siguientes son los espectrogramas del micrófono 1 con ruido sin filtrar, con ruido pero filtrado con el filtro pasabanda, y de la señal obtenida al promediar los micrófonos filtrados.

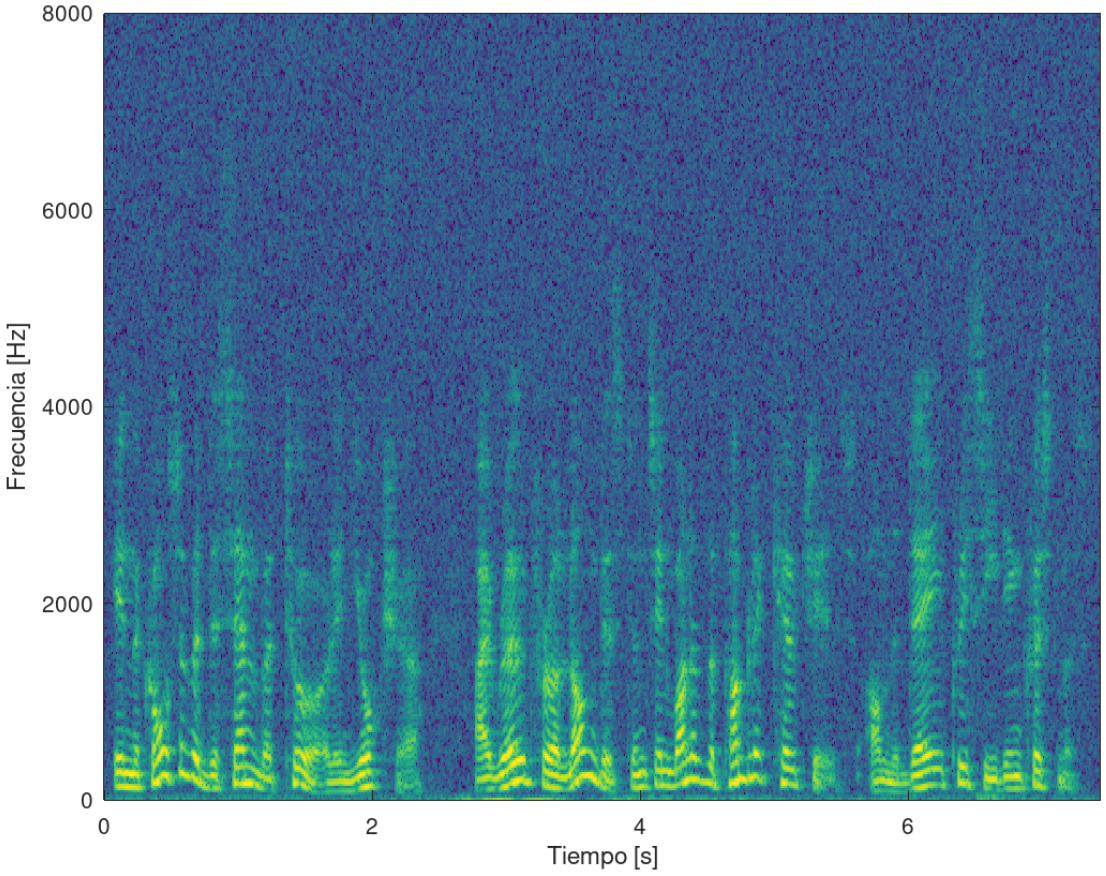

Figura 54: Espectrograma del Micrófono 1 con ruido

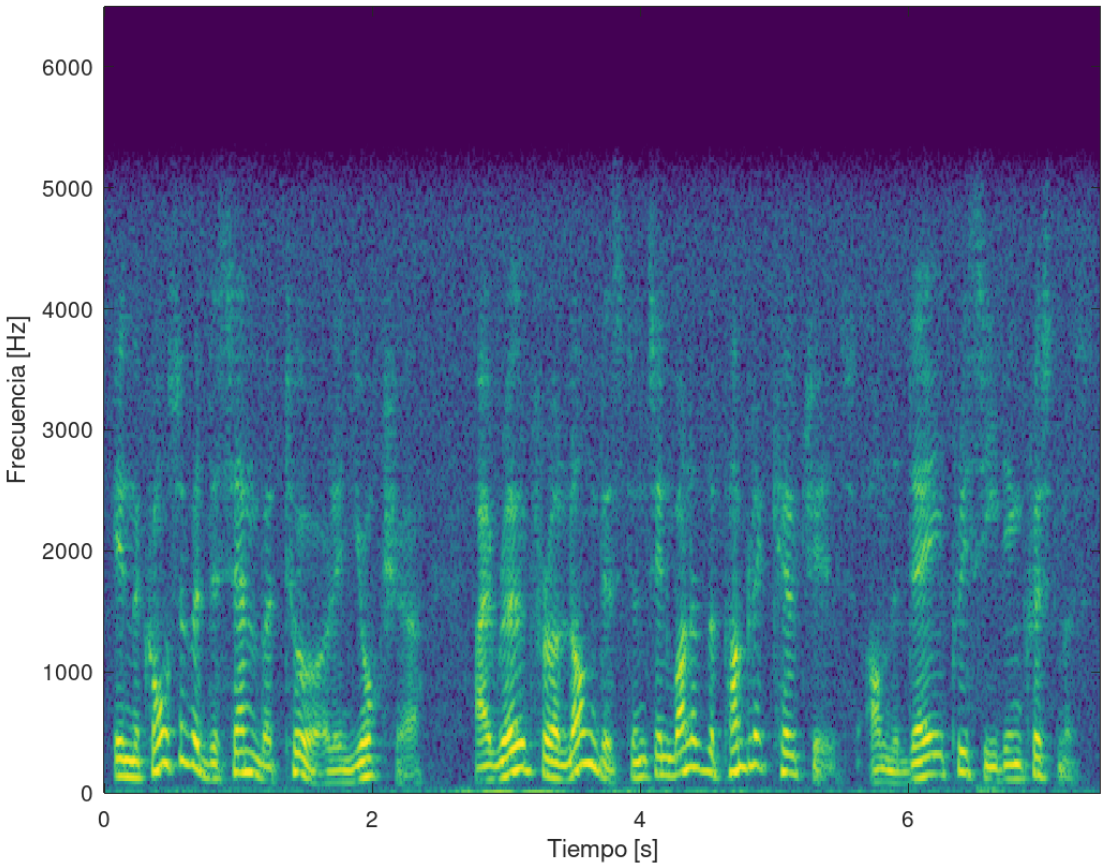

Figura 55: Espectrograma del Micrófono 1 con ruido y filtrado con el pasabanda

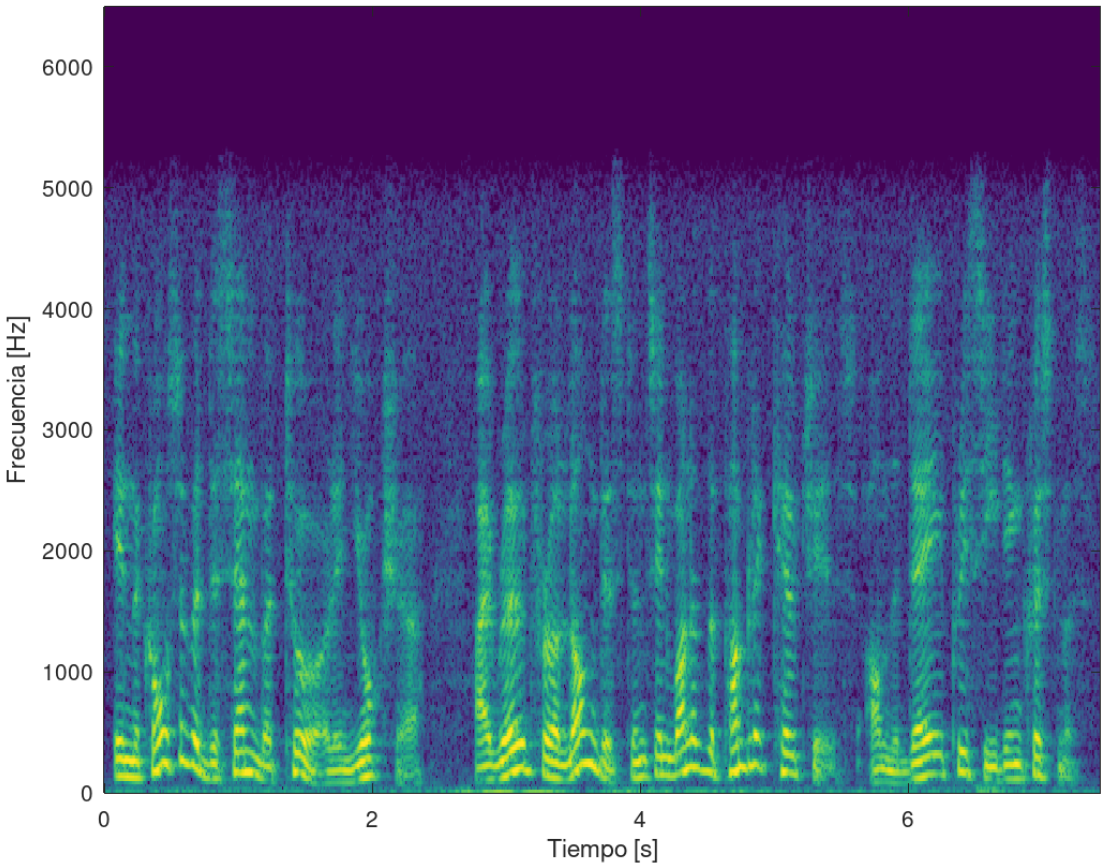

Figura 56: Espectrograma del promedio de las señales alineadas filtradas

Vemos que el primero de los espectrogramas, el cual corresponde a la señal con ruido blanco de 20 dB de SNR, si bien se siguen distinguiendo perfectamente las frecuencias principales de la señal en sí todas las demás frecuencias tienen más potencia ya que su tonalidad es azulada (la escala es desde violeta a amarillo, siendo violeta la mínima potencia). El espectrograma de la señal una vez filtrada por el pasabanda del **Ejercicio 7** muestra que las frecuencias por encima de 5 KHz ya prácticamente no tienen potencia mientras que por debajo de 5 KHz el gráfico es el mismo, lo cual es lógico dado que el filtro lo habíamos diseñado para conseguir esos resultados. Por último al realizar la comparación con el espectrograma obtenido al promediar las señales filtradas y alineadas vemos una disminución en el azulado de las frecuencias por debajo de los 5 KHz, indicando una disminución en el ruido de la señal. Esto se debe a lo explicado en el **Ejercicio 8** que sucede con el ruido blanco al promediarlo.

Por último se muestra el espectrograma de la señal del Micrófono 1 original, aquel mostrado en el **Ejercicio 1**, de forma de poder compararlo con el filtrado obtenido de la señal con ruido de la **Figura 56**.

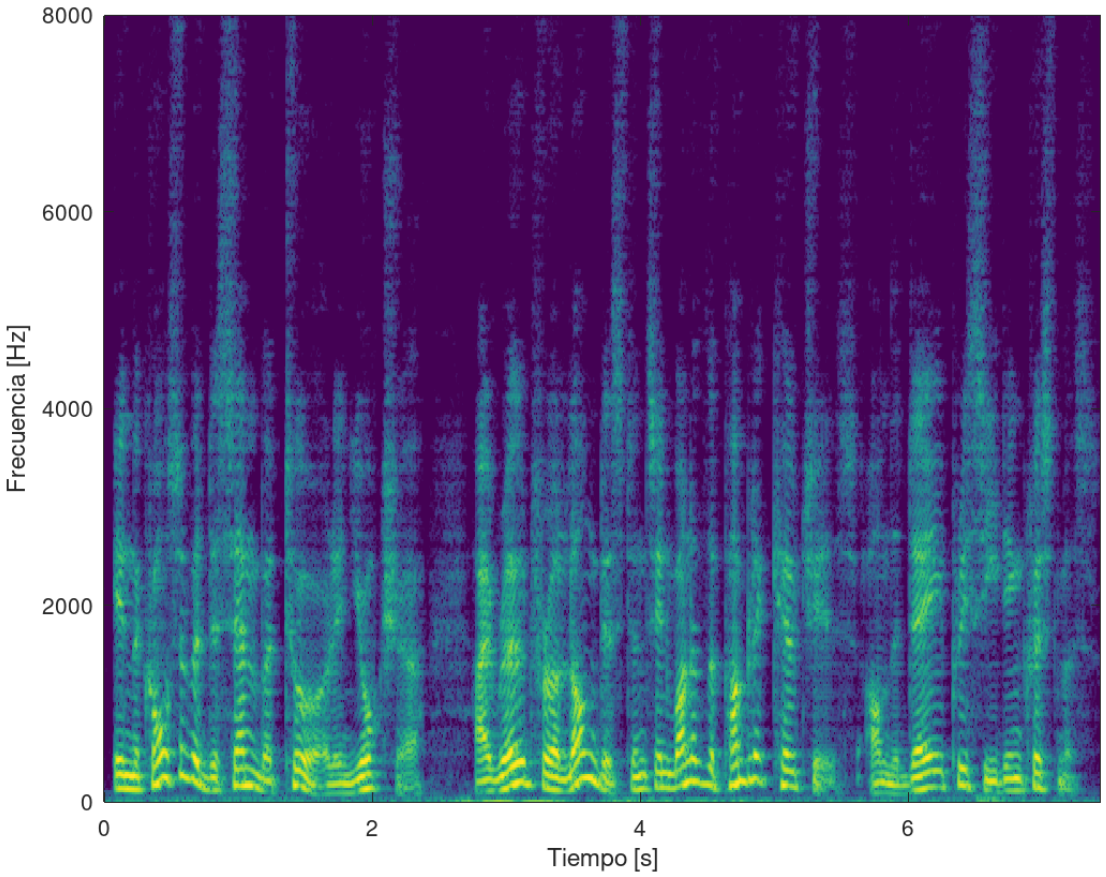

Figura 57: Espectrograma del Micrófono 1 sin ruido entre los 0 Hz y 8000 Hz

Se observa que si bien las frecuencias que tienen más potencia son claramente distinguibles en ambos espectrogramas, sigue notándose fácilmente el efecto del ruido en la **Figura 56**. Esto es esperable ya que al sólo estar promediando 5 señales contaminadas con ruido blanco no se logra disminuir mucho el mismo. Si se desease disminuirlo más todavía se podrían generar muchas más señales con ruido blanco y promediarlas de forma tal que este disminuya y se pueda conseguir una mejor aproximación a la señal original, pero esto excede a lo requerido por el enunciado.

Ejercicio 11

Escuche la señal audio2.wav donde se simula el comportamiento del oído humano para localizar fuentes de sonido. Determine preceptivamente de qué lado provienen los distintos segmentos de la señal. Utilice el algoritmo de estimación de retardos para confirmar las posiciones.

El audio en cuestión contiene 5 repeticiones de un mismo audio donde en cada repetición se puede percibir la dirección de la que proviene el mismo. A oído los 5 audios parecen provenir de una dirección levemente frontal, como si proviniesen de adelante. Además se pueden reconocer las siguientes direcciones (en el orden en el que se escuchan los audios): **izquierda**, **derecha**, **derecha**, **derecha**, **izquierda**.

Para este ejercicio se entiende que los micrófonos vendrían a representar el oído humano. Se utilizó una distancia entre "micrófonos" entonces de 20 cm de forma tal de aproximar la distancia promedio entre los oídos de una persona. Separando el audio en 5 partes donde cada una contiene una de las repeticiones que se mencionaron y procesando cada una con el algoritmo GCC-PHAT del **Ejercicio 4** (con un sobremuestreo con factor de $\mathbf{L} = 10$ en la IDFT(Gph[k])) se obtuvieron los siguientes retardos.

- **Primer audio**: 259 us de retardo del Micrófono 2 respecto al Micrófono 1.
- **Segundo audio**: 252 us de retardo del Micrófono 1 respecto al Micrófono 2.
- **Tercer audio**: 252 us de retardo del Micrófono 1 respecto al Micrófono 2.
- **Cuarto audio**: 252 us de retardo del Micrófono 1 respecto al Micrófono 2.
- **Quinto audio**: 259 us de retardo del Micrófono 2 respecto al Micrófono 1.

Al solo tener dos micrófonos tan solo podemos captar la dirección de la fuente por lo que no podremos estimar una posición exacta, pero esto no es un problema ya que únicamente nos interesa comparar con las direcciones que identificamos auditivamente. Observamos entonces las siguientes direcciones de la fuente de sonido.

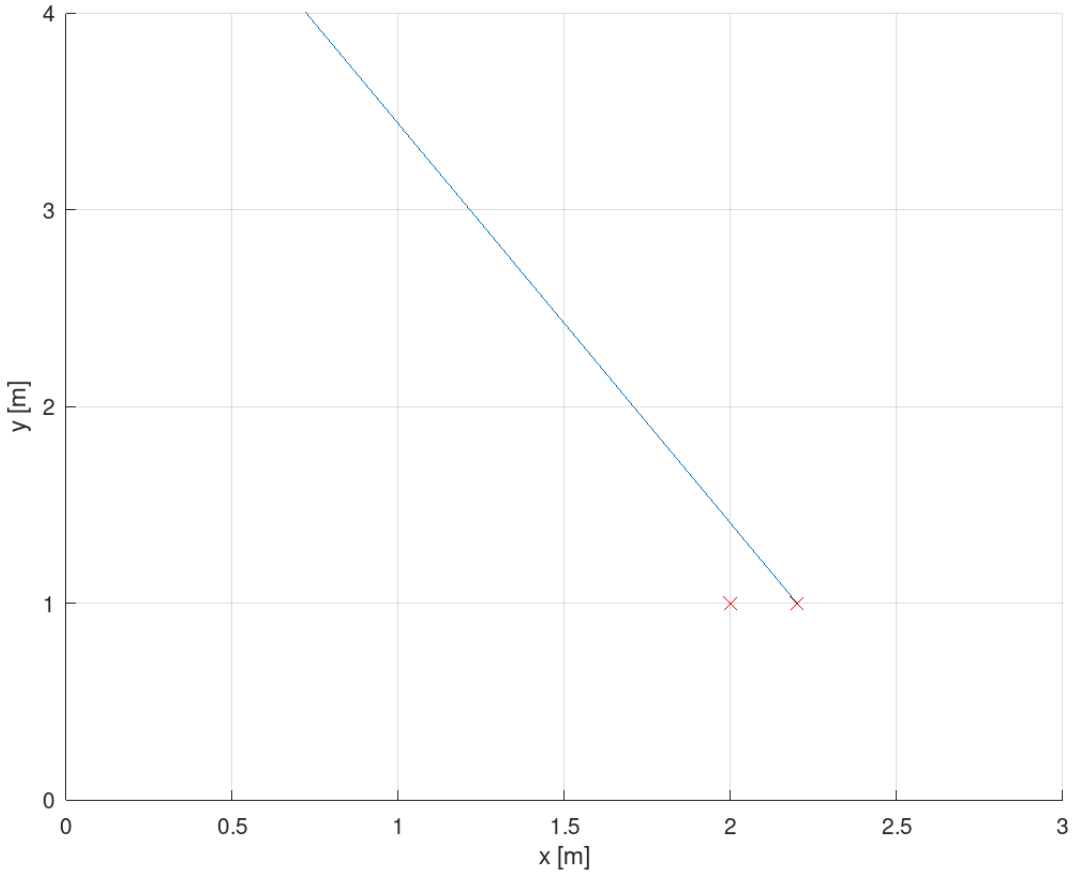

Figura 58: Disposición de los micrófonos y dirección de la fuente de sonido en el primer audio

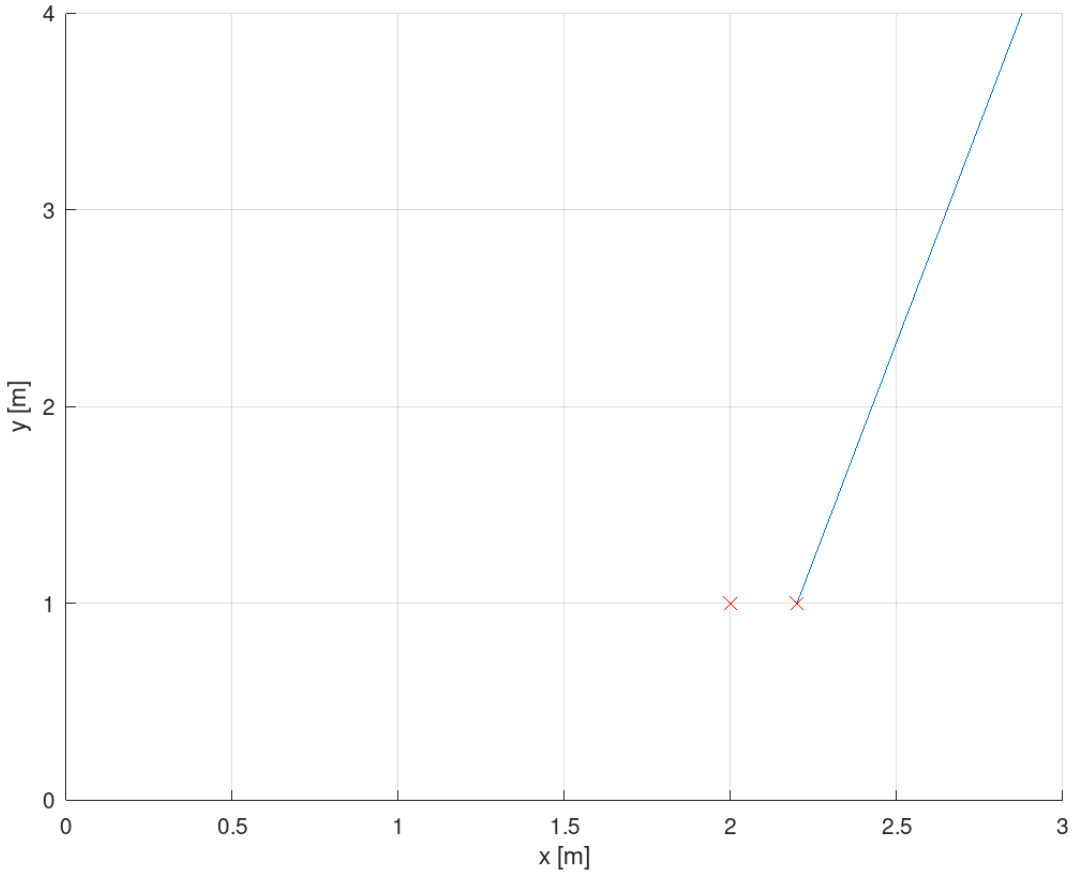

Figura 59: Disposición de los micrófonos y dirección de la fuente de sonido en el segundo audio

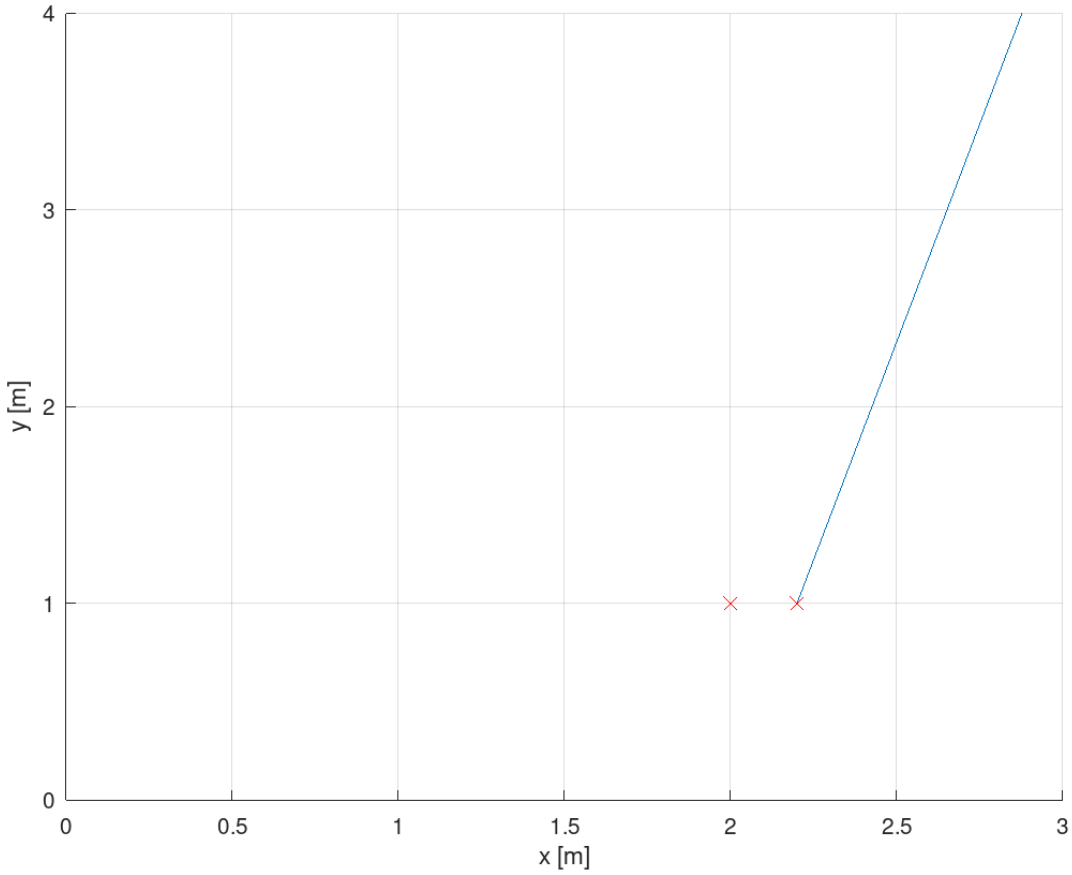

Figura 60: Disposición de los micrófonos y dirección de la fuente de sonido en el tercer audio

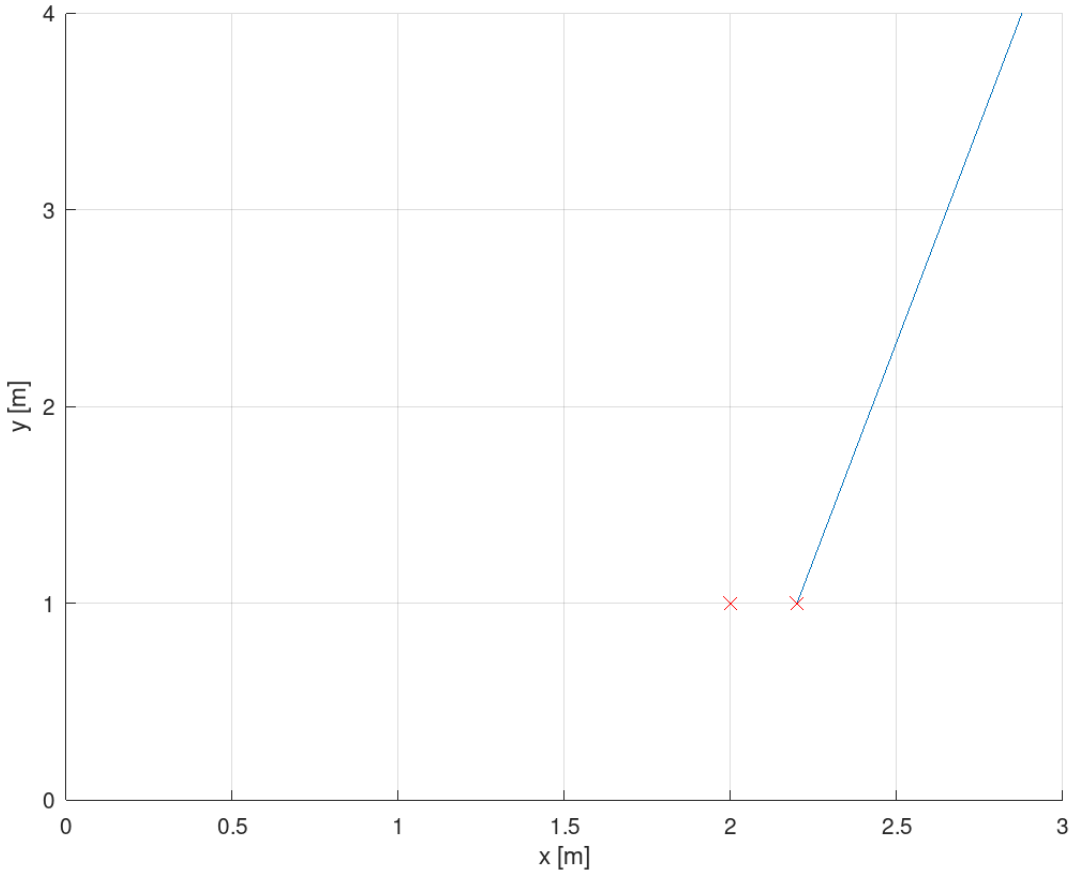

Figura 61: Disposición de los micrófonos y dirección de la fuente de sonido en el cuarto audio

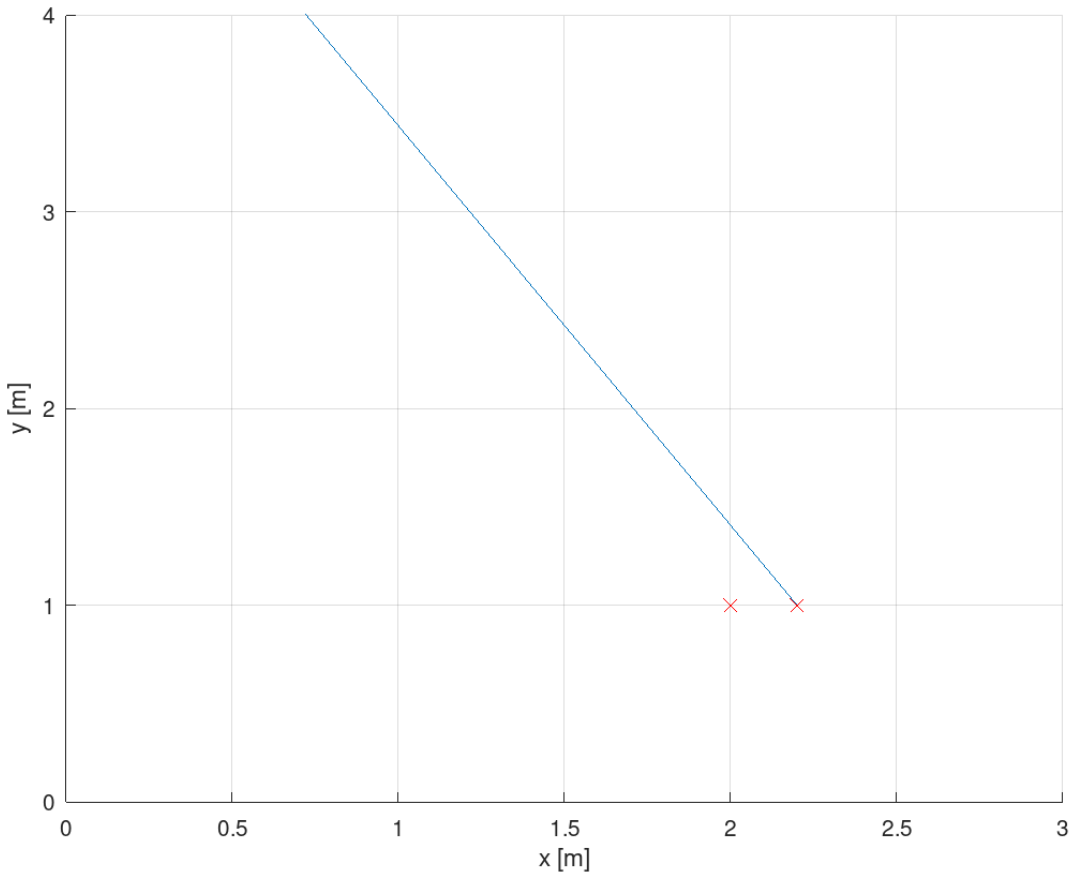

Figura 62: Disposición de los micrófonos y dirección de la fuente de sonido en el quinto audio

Podemos ver que efectivamente hemos acertado con las direcciones que estimamos a oído en un comienzo, ya que obtuvimos las mismas direcciones mediante el análisis algorítmico de las señales en todos los casos.

Ejercicio 12

Escuche la señal audio3.wav donde se simula una fuente sonora en movimiento. ¿Que percibe que está sucediendo? Utilice el algoritmo de estimación de retardos para analizar la evolución temporal de los mismos.

Al escuchar el audio3 se percibe en un comienzo que el audio proviene de la izquierda. A medida que la grabación avanza se empieza a notar como el origen del audio se va desplazando hacia la derecha paulatinamente, por lo que estamos tratando con una fuente en movimiento. A diferencia del **Ejercicio 11** en este caso el audio solo se escucha una vez por lo que el desplazamiento es relativamente rápido (el audio apenas dura unos 5 segundos) pero igualmente es notorio. Utilizando el algoritmo de estimación de los retardos esperaríamos entonces ver como inicialmente el segundo micrófono (recordemos que son nuestros oídos en la práctica) capta el sonido con un retardo respecto al primero y donde a medida que avanza el tiempo se van invirtiendo los roles, donde entonces eventualmente será el primer "micrófono" el que tenga un retardo en el audio captado respecto al segundo. Sabemos que esto es así ya que si nuestro cerebro percibe una señal sonora con un leve retardo en (por ejemplo) el oído derecho respecto al izquierdo, entonces interpreta que ese sonido proviene desde la izquierda (y viceversa).

Para procesar la señal se optó por separarla en cinco partes y procesar el retardo en cada una. Se realizó un sobremuestreo con factor $\mathbf{L} = 10$ en la IDFT(Gph[k]) al igual que en el **Ejercicio 11**. En este caso se utilizó una ventana de 10.0000 muestras y un overlapping de 5.000 muestras. Los resultados obtenidos fueron los siguientes.

- **Primera parte**: 199 us de retardo del Micrófono 2 respecto al Micrófono 1.
- **Segunda parte**: 61,1 us de retardo del Micrófono 2 respecto al Micrófono 1.
- **Tercera parte**: 463 ns de retardo del Micrófono 2 respecto al Micrófono 1.
- **Cuarta parte**: 92,1 us de retardo del Micrófono 1 respecto al Micrófono 2.
- **Quinta parte**: 263 us de retardo del Micrófono 1 respecto al Micrófono 2.

Vemos que el retardo es prácticamente nulo en la tercera parte y se invierte a partir de la cuarta, lo cual coincide con lo que esperábamos ver. A continuación se muestran las direcciones de la fuente correspondiente a cada parte del audio procesada.

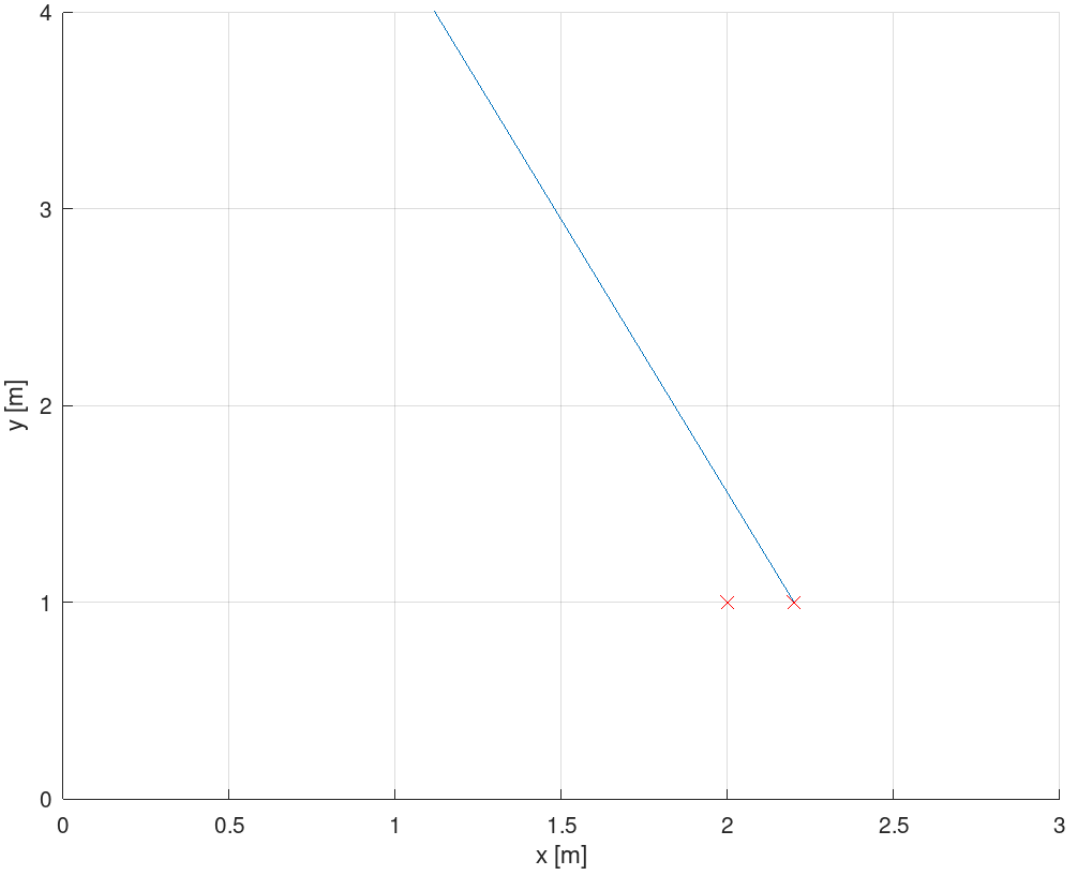

Figura 63: Disposición de los micrófonos y dirección de la fuente de sonido en la primera parte del audio

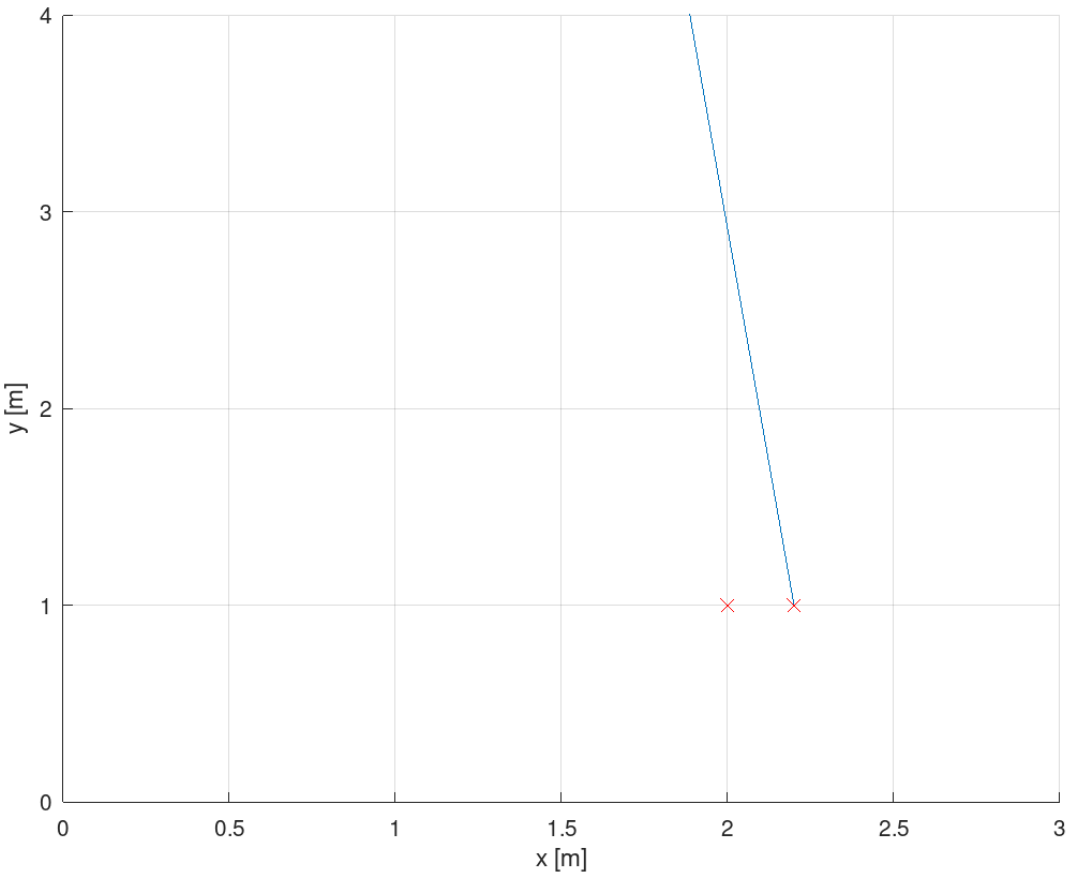

Figura 64: Disposición de los micrófonos y dirección de la fuente de sonido en la segunda parte del audio

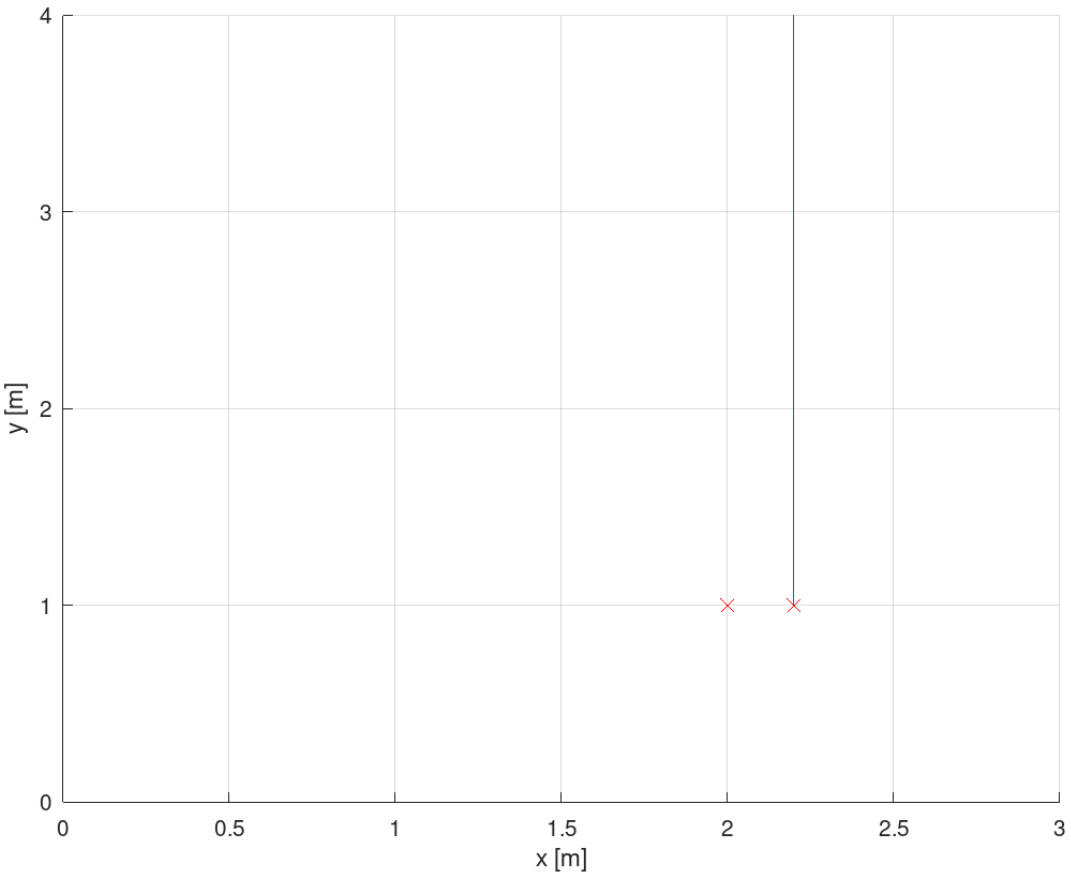

Figura 65: Disposición de los micrófonos y dirección de la fuente de sonido en la tercera parte del audio

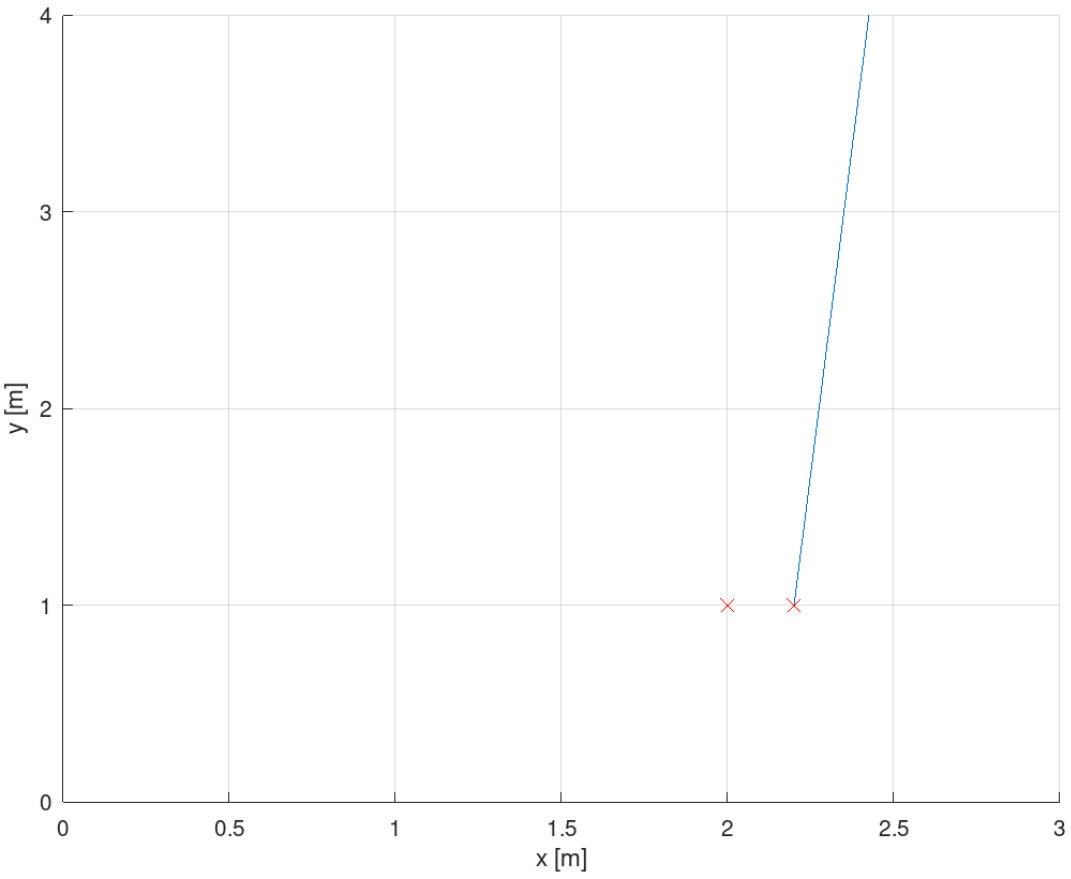

Figura 66: Disposición de los micrófonos y dirección de la fuente de sonido en la cuarta parte del audio

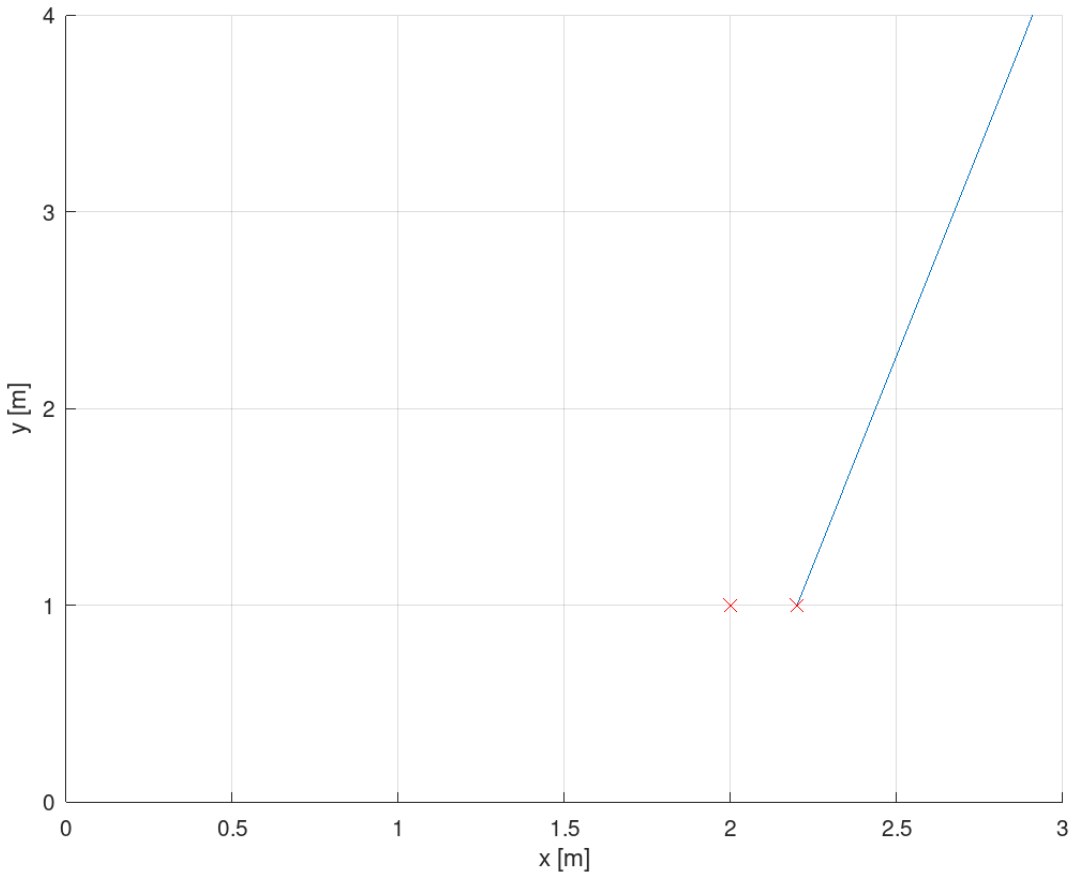

Figura 67: Disposición de los micrófonos y dirección de la fuente de sonido en la quinta parte del audio

En el siguiente gráfico se muestran todas las direcciones registradas secuencialmente.

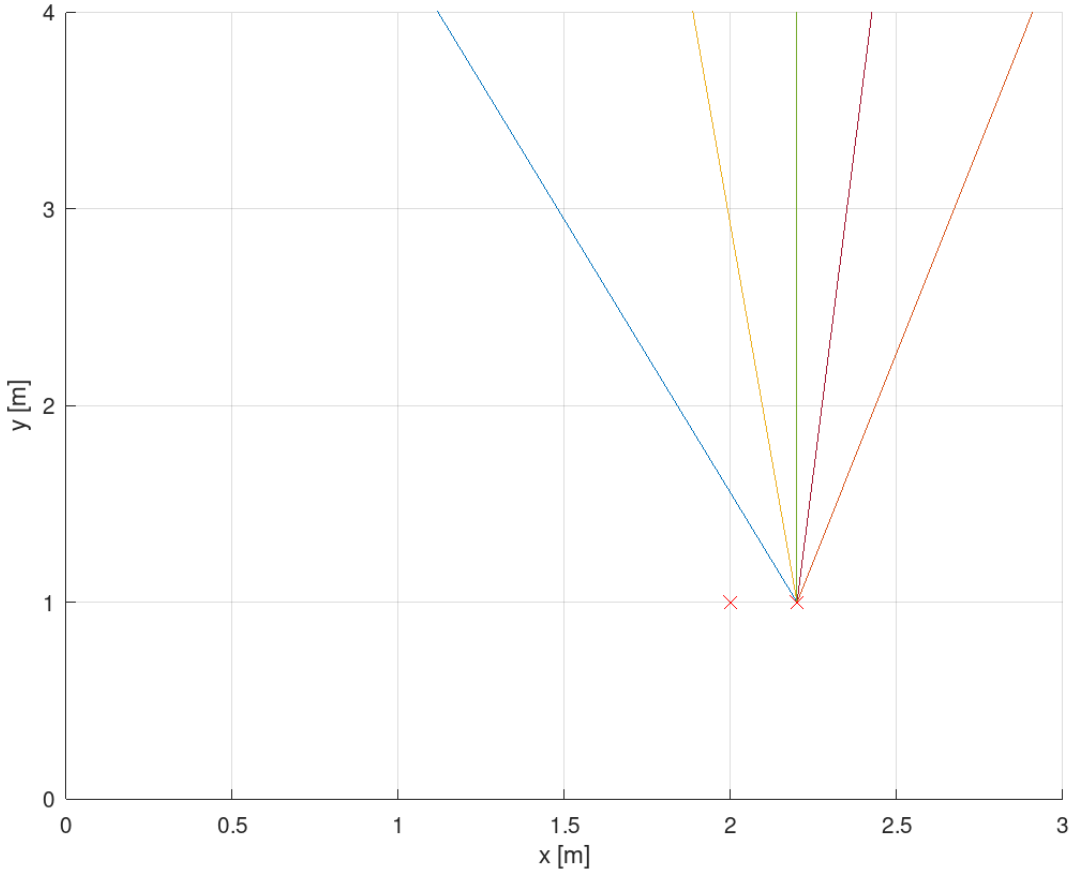

Figura 68: Disposición de los micrófonos y direcciones registradas de la fuente

Nuevamente conseguimos resultados que coinciden con lo percibido auditivamente. Vemos que la dirección inicial que calcula el algoritmo de la fuente es hacia la izquierda y luego va avanzando hacia la derecha, tal y como habíamos establecido en un principio.

Conclusión

A lo largo de este trabajo utilizamos distintos métodos para estimar los retardos entre los micrófonos así como también modificamos las señales mediante el agregado de ruido blanco pudiendo estudiar como afecta el mismo a los algoritmos de estimación y como se refleja el mismo en los espectros de las señales. Si bien en la teoría el método de GCC-PHAT es más robusto y menos susceptible a ambientes ruidosos observamos que en la práctica, al menos con la implementación propuesta para el método, no obtuvimos resultados coherentes al filtrar el ruido agregado a la señales. Si bien se estableció una posible causa en la estabilidad del algoritmo utilizado la realidad es que no se pudo encontrar con total seguridad el causante de esto. Por otro lado, el método de correlación cruzada sí que arrojó buenos resultados. Si bien conseguimos estimar una posible posición de la fuente con bastante exactitud mediante este método en el **Ejercicio 7** resulta una decepción que GCC-PHAT no haya funcionado en ese ejercicio. Cabe aclarar que en todos los demás ejercicios (incluso los optativos) este método funcionó correctamente.

Por último, vimos como al promediar las señales alineadas se obtuvo una disminución en el ruido de la señal final, lo sí concordó con lo que se esperaba en la teoría. Comprobamos empíricamente así una de las propiedades del ruido blanco.

A modo de conclusión se adjunta a continuación la última (y mejor) estimación que se pudo realizar de la fuente, tal y como se había mostrado den la **Figura 53**.

La posición de la fuente estimada es la siguiente.

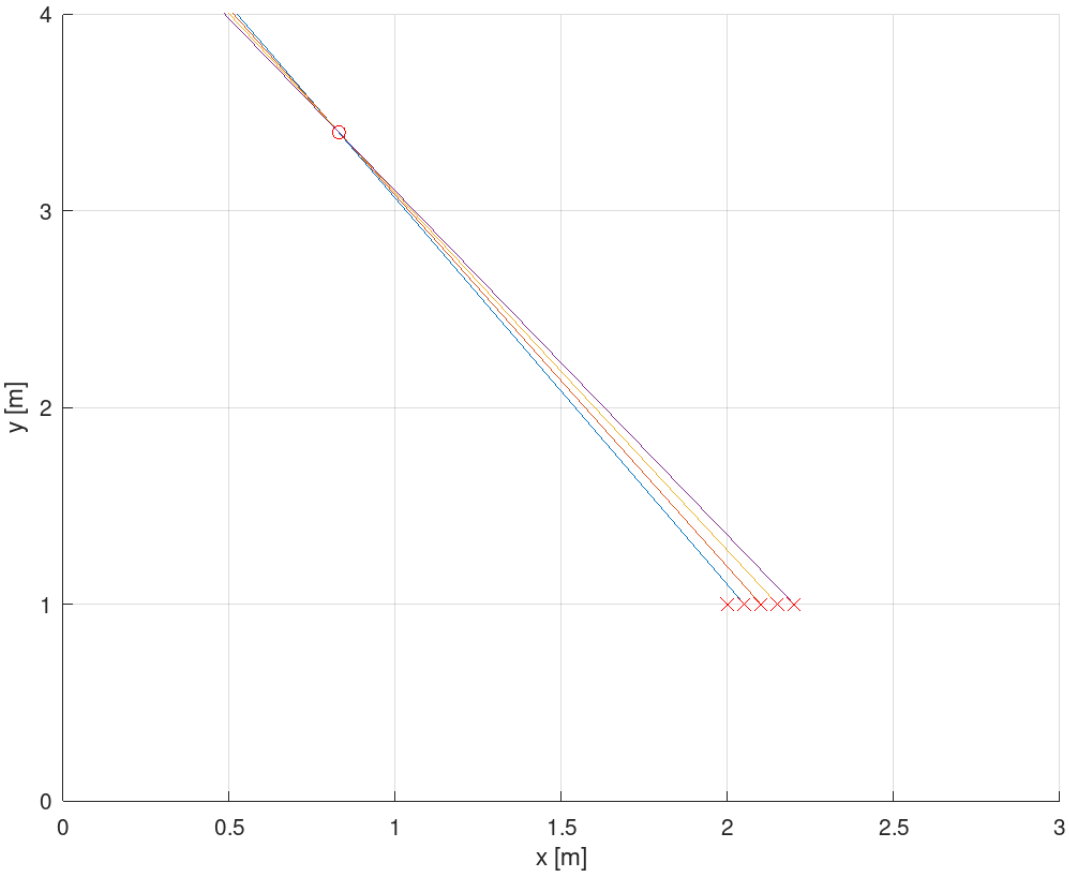

Figura 69: Disposición de los micrófonos y de la fuente de sonido

La fuente se encuentra a 0,83 metros en **x** y 3,40 metros en **y** respecto del origen.

```
pkg load signal
pkg load communications % Para la funcion awgn()
global audios Fs c d hist_figure_counter delay_time_counter

c = 340; % Velocidad del sonido en el aire en m/s
d = 0.05; % Distancia entre los microfonos en m
hist_figure_counter = 100; % Para crear distintas figuras para los histogramas
delay_time_counter = 50; % Para crear distintas figuras para los retardos en el tiempo

format short e % Para mostrar los resultados en notacion cientifica

function plot_lines(slope)
  hold
  grid on

  for i = (1:4)
    line([(2 + 0.05*i) 0], [1 (1 - slope(i)*(2 + 0.05*i))])
  endfor
  plot([2, 2.05, 2.1, 2.15, 2.2], [1, 1, 1, 1, 1], 'x', 'color', 'r')
  %plot([0.83], [3.40], 'o', 'color', 'r')
  xlim([0, 3])
  ylim([0, 4])
  xlabel("x [m]")
  ylabel("y [m]")
endfunction

% Specgram basado en la documentacion de Octave
function plot_specgram(audio, Fs, F_range)
  step = fix(5*Fs/1000);     # one spectral slice every 5 ms
  window = fix(40*Fs/1000);  # 40 ms data window
  fftn = 2^nextpow2(window); # next highest power of 2
  [S, f, t] = specgram(audio, fftn, Fs, window, window-step);
  S = abs(S); # magnitude
  S = S/max(S(:));           # normalize magnitude so that max is 0 dB.
  S = max(S, 10^(-40/10));   # clip below -40 dB.
  S = min(S, 10^(-3/10));    # clip above -3 dB.
  imagesc(t, f, log(S));    # display in log scale
  set(gca, "ydir", "normal"); # put the 'y' direction in the correct direction
  xlabel("Tiempo [s]")
  ylabel("Frecuencia [Hz]")
  ylim([0 F_range])
endfunction

%load audios1.mat % Cargo las seniales sin ruido
load audiosRuido1.mat % Cargo las seniales con ruido

close all % Para cerrar los graficos al correr de nuevo el programa

audios(384000:end,:) = []; % Elimino los ultimos 2 segundos aprox (del 8 al 10 en segundos) donde la senial es 0
audios(1:25000,:) = [];

% Ejercicio 5
%{
for i = (1:5)
  audios(:,i) = awgn(audios(:,i), 20, "measured", 1); % Agrego 20dB de ruido blanco a cada senial
endfor
%}

% Ejercicio 1

Fs = 48000; % 48 KHz

t = (0:rows(audios)-1)/Fs;

hf = figure(1);

hold
grid on

for i = (1:5)
  plot(t, audios(:,i) + (i - 1)*0.12)
endfor

xlim([0, 7.45])
ylim([-0.1, 0.75])

legend("Micrófono 1", "Micrófono 2",
       "Micrófono 3", "Micrófono 4"
       , "Micrófono 5")
xlabel("Tiempo [s]")
ylabel("Amplitud")

print(hf, "audios_con_ruido.png")

% Hacemos Zoom para apreciar el retardo

hf = figure(2);
hold
grid on

for i = (1:5)
  plot(t, audios(:, i))
endfor

xlim([4430/Fs, 4451/Fs])
ylim([-0.0024, 0])

legend("Micrófono 1", "Micrófono 2",
       "Micrófono 3", "Micrófono 4"
       , "Micrófono 5")
```

```
      , "Micrófono 5")
xlabel("Tiempo [s]")
ylabel("Amplitud")

print(hf, "audios_con_ruido_zoom.png")

hf = figure(3);
hold
grid on
for i = (1:5)
   fourier_audio = abs(fftshift(fft(audios(:,i))));
   plot((0:1:rows(fourier_audio)-1)*Fs/rows(fourier_audio) - Fs/2, fourier_audio + (i-1)*200)
endfor
xlim([-4000, 4000])
legend("Micrófono 1", "Micrófono 2",
       "Micrófono 3", "Micrófono 4"
       , "Micrófono 5")
xlabel("Frecuencia [Hz]")
ylabel("Amplitud")

print(hf, "transf_fourier_con_ruido.png")

hf = figure(4);
plot_specgram(audios(:,1), Fs, 8000)

print(hf, "specgram_con_ruido.png")

% Ejercicio 2

tau = [-60*10^(-6), -60*10^(-6), -60*10^(-6), -60*10^(-6)]; % Taus medidos visualmente con el grafico del Ejercicio 1
tita = acos(tau * c / d);
slope = tan(tita);

hf = figure(20);
plot_lines(slope);
print(hf, "ejercicio2_sin_ruido.png")

% Ejercicio 3
disp("Ejercicio 3")

for i = (1:4)
   [coeffs, lags] = xcorr(audios(:,i+1), audios(:,i), 20); % Hasta 20 de max_lag
   [~, max_index] = max(coeffs);
   k(i) = -lags(max_index); % Dado que xcorr usa en la definicion x[i + k] pero el enunciado usa x[i - k],
                            % debemos hacer un negado del indice para que sea equivalente a la del enunciado
endfor

disp("\nRetardos obtenidos mediante el método de correlación cruzada (en segundos):\n")
taus = k / Fs % Retardos entre los microfonos consecutivos

for i = (1:4)
   dft_curr_mic = fft(audios(:,i));
   dft_next_mic = fft(audios(:,i+1));
   R = dft_curr_mic .* conj(dft_next_mic);
   Gph = R ./ abs(R);
   gph = real(ifft(Gph));
   [~, max_index] = max(gph);
   if (max_index > rows(audios)/2)
      max_index = max_index - rows(audios);
   endif
   taus(i) = max_index - 1;
endfor

disp("\nRetardos obtenidos mediante el método de GCC-PHAT (en segundos):\n")
taus = taus / Fs % Paso de muestras a tiempo

% Ejercicio 4
disp("Ejercicio 4")

function slope = calculate_lines (audios, Fs, N, delta_n, upsample_gph)
   global c d hist_figure_counter delay_time_counter
   L = 1;

   if (upsample_gph)
      L = 10; % Factor de upsampling
      Fs = Fs * L; % Nueva frecuencia de muestreo
      b = fir1(150, 1/L, "low") * L; % Filtro pasabajos interpolador para el upsampling
      delay = mean(round(grpdelay(b))); % Retardo del filtro, como es lineal el filtro entonces el valor medio es el valor de cada elemento realmente y ademas es la mitad del orden del filtro
      for i = (1:20)
         hist_bins(i) = (i + 29) / Fs;
      endfor
   else
      for i = (1:5)
         hist_bins(i) = i / Fs;
      endfor
   endif

   tita = []; % Angulos
   slope = []; % Pendientes

   for i = (1:4)
      x = audios(:,i);
      y = audios(:,i+1);
      n0 = N / 2;
      j = 1;

      while ((n0 + N/2) < rows(audios))
         w_start = n0 - N/2 + 1;
         w_end = n0 + N/2;
         dft_x = fft(x(w_start:w_end));
```

```
        dft_y = fft(y(w_start:w_end));
        R = dft_x .* conj(dft_y);
        Gph = R ./ abs(R); % Un epsilon que evita la inestabilidad del algoritmo cuando R es muy cercano a 0
        gph = real(ifft(Gph));
        if (upsample_gph) % Realizar un upsample de gph para estimar mejor los retardos
            aux = upsample(gph, L); % Upsample de gph
            gph = zeros(rows(aux)+delay, 1); % Agrego delay cantidad de 0s de forma que no pierda la informacion original en el filtro al final de la senial
            gph(1:end-delay) = aux;
            gph = filter(b, 1 , gph); % Filtramos/Interpolamos
            gph = gph(delay+1:end); % Anulamos el desfase introducido por el filtro
        endif
        [~, m] = max(gph);
        if (m > N*L/2) % Si el indice es mayor a la mitad del ancho de la ventana
            m = m - N*L;
        endif
        curr_tau = (m-1) / Fs;
        if (m < 1) % No tiene sentido que me de retardo inverso o que no hay retardo asi que no lo considero
            tau_xy(j) = curr_tau;
            j = j + 1;
        endif
        n0 = n0 + delta_n; % Actualizo la ventana
    endwhile

    hf = figure(hist_figure_counter);
    h = gca();
    hist(abs(tau_xy), hist_bins) % Grafico el histograma de los retardos
    xlabel(h, "Retardos [s]")
    ylabel(h, "Cantidad")
    print(hf, strcat("hist_audios_con_ruido", num2str(hist_figure_counter), ".png"))
    hf = figure(delay_time_counter);
    hold
    h = gca();
    plot(abs(tau_xy), 'o')
    mean_tau = mean(tau_xy)
    line([0 length(tau_xy)], abs([mean_tau mean_tau]), 'color', 'r')
    ylabel(h, "Retardo [s]")
    print(hf, strcat("delay_audios_con_ruido", num2str(delay_time_counter), ".png"))
    delay_time_counter = delay_time_counter + 1;
    hist_figure_counter = hist_figure_counter + 1;
    tita(i) = acos(mean_tau * c / d);
    slope(i) = tan(tita(i));
    clear tau_xy
  endfor

endfunction

N = 20000;
delta_n = N/2;
disp("\nRetardos medios mediante GCC-PHAT (en segundos)\n")
slope = calculate_lines(audios, Fs, N, delta_n, false);

hf = figure(5);
plot_lines(slope);
print(hf, "rectas_ejercicio4_con_ruido_con_fuente.png")

%Ejercicio 6

disp("\nEjercicio 6")
disp("\nRetardos medios mediante GCC-PHAT (en segundos)\n")
slope = calculate_lines(audios, Fs, N, delta_n, true);

hf = figure(6);
plot_lines(slope)
print(hf, "rectas_ejercicio4_con_ruido_con_fuente_con_sobremuestreo.png")

% Ejercicio 7

disp("\nEjercicio 7")

b = fir1(150, 5000/(Fs / 2), "low"); % Disenio el filtro pasabanda que filtre desde los 80Hz hasta los 800Hz
delay = round(mean(grpdelay(b))); % Es la mitad del orden del filtro
audios_noise_filtered = [];
for i = (1:5)
  aux = audios(:,i);
  aux(end+1:end+delay) = 0;
  audios_noise_filtered(:,i) = filter(b, 1, aux);
endfor
audios_noise_filtered = audios_noise_filtered(delay+1:end, :);

hf = figure(7);
plot_specgram(audios_noise_filtered(:,1), Fs, 6500)
print(hf, "ejercicio7_specgram.png")

hf = figure(8);
hold
freqz(b, 1, length(b), Fs);
print(hf, "ejercicio7_respuesta_filtro.png")

hf = figure(9);
zplane(b, 1);
print(hf, "ejercicio7_polos_y_ceros.png")

disp("\nRetardos medios mediante GCC-PHAT (en segundos)\n")
slope = calculate_lines(audios_noise_filtered, Fs, N, delta_n, true);
hf = figure(10);
plot_lines(slope)
print(hf, "ejercicio7_posicion_fuente.png")

% Upsampling de las seniales originales
```

```
L = 10;
b = fir1(150, 1/L, "low") * L; % Filtro pasabajos interpolador para el upsampling
delay = round(mean(grpdelay(b)));

for i = (1:5)
  aux = upsample(audios_noise_filtered(:,i), L); % Upsample de gph
  gph = zeros(rows(aux)+delay, 1); % Agrego delay cantidad de 0s de forma que no pierda la informacion original en el filtro al final de la senial
  gph(1:end-delay) = aux;
  gph = filter(b, 1 , gph); % Filtramos/Interpolamos
  gph = gph(delay+1:end); % Anulamos el desfase introducido por el filtro
  audios_f(:,i) = gph;
endfor

for i = (1:4)
  [coeffs, lags] = xcorr(audios_f(:,i+1), audios_f(:,i), 100); % Paso
  [~, max_index] = max(coeffs);
  k(i) = -lags(max_index); % Dado que xcorr usa en la definicion x[i + k] pero el enunciado usa x[i - k],
                           % debemos hacer un negado del indice para que sea equivalente a la del enunciado
endfor

Fs = Fs * L;

disp("\nRetardos obtenidos mediante el método de correlación cruzada (en segundos):\n")
taus = k / Fs % Retardos entre los microfonos consecutivos

hf = figure(11);
tita = acos(taus * c / d);
slope = tan(tita);
plot_lines(slope)
print(hf, "ejercicio7_mejor_fuente.png")

 % Ejercicio 8


aux = zeros(rows(audios_f), 5);
aux(:,1) = audios_f(:,1);
delay = 0;
for i = (2:5)
  delay = delay + k(i-1);
  aux(1:end-abs(delay),i) = audios_f(1+abs(delay):end,i);
endfor
audios_f = aux;
audio_prom = audios_f(:,1);
for i = (2:5)
  audio_prom = audio_prom + audios_f(:,i);
endfor
audio_prom = audio_prom / 5;
audio_prom = audio_prom(1:L:end); % Decimacion

 % Ejercicio 9

hf = figure(12);
plot_specgram(audio_prom, 48000, 6500)
print(hf, "ejercicio9.png")


clear all % Clear all variables
```

```
pkg load signal

close all

global c d

c = 340; % Velocidad del sonido en el aire en m/s
d = 0.20; % Distancia entre los oidos en m

format short e

function plot_lines(slope)
  hold
  grid on

  if (slope < 0) % Asi las recta la dibujo en el intervalo de interes
    line([(2 + 0.20) 0], [1 (1 - slope*(2 + 0.20))])
  else
    line([(2 + 0.20) 4], [1 (1 + slope*(4 - 0.20))])
  endif
  plot([2, 2.20], [1, 1], 'x', 'color', 'r')
  xlim([0, 3])
  ylim([0, 4])
  xlabel("x [m]")
  ylabel("y [m]")
endfunction

function slope = calculate_lines (audios, Fs, N, delta_n, upsample_gph)
  global c d
  L = 1;

  if (upsample_gph)
    L = 10; % Factor de upsampling
    Fs = Fs * L; % Nueva frecuencia de muestreo
    b = fir1(150, 1/L, "low") * L; % Filtro pasabajos interpolador para el upsampling
    delay = mean(round(grpdelay(b))); % Retardo del filtro, como es lineal el filtro entonces el valor medio es el valor de cada elemento realmente y ademas es la mitad del orden del filtro
  endif

  tita = []; % Angulos
  slope = [];% Pendientes

  x = audios(:,1);
  y = audios(:,2);
  n0 = N / 2;
  j = 1;

  while ((n0 + N/2) < rows(audios))
    w_start = n0 - N/2 + 1;
    w_end = n0 + N/2;
    dft_x = fft(x(w_start:w_end));
    dft_y = fft(y(w_start:w_end));
    R = dft_x .* conj(dft_y);
    Gph = R ./ abs(R); % Un epsilon que evita la inestabilidad del algoritmo cuando R es muy cercano a 0
    gph = real(ifft(Gph));
    if (upsample_gph) % Realizar un upsample de gph para estimar mejor los retardos
      aux = upsample(gph, L); % Upsample de gph
      gph = zeros(rows(aux)+delay, 1); % Agrego delay cantidad de 0s de forma que no pierda la informacion original en el filtro al final de la senial
      gph(1:end-delay) = aux;
      gph = filter(b, 1 , gph); % Filtramos/Interpolamos
      gph = gph(delay+1:end); % Anulamos el desfase introducido por el filtro
    endif
    [~, m] = max(gph);
    if (m > N*L/2) % Si el indice es mayor a la mitad del ancho de la ventana
      m = m - N*L;
    endif

    if (m != 1)
      curr_tau = (m-1) / Fs;
      tau_xy(j) = (m-1) / Fs;
      j = j + 1;
    endif
    n0 = n0 + delta_n; % Actualizo la ventana
  endwhile

  mean_tau = mean(tau_xy)
  tita = acos(mean_tau * c / d);
  slope = tan(tita);
  clear tau_xy

endfunction

[audios, Fs] = audioread("audio2.wav");
cut_n = rows(audios) / 5;
N = 20000;
delta_n = N / 2;
disp("\nRetardos medios obtenidos mediante GCC-PHAT (en segundos)\n")
for i = (1:5)
  hf = figure(i);
  slope = calculate_lines(audios(1+cut_n*(i-1):cut_n*i, :), Fs, N, delta_n, true);
  plot_lines(slope)
  print(hf, strcat("ejercicio11", num2str(i), ".png"))
endfor

clear all
```

```
pkg load signal

close all

global c d

c = 340; % Velocidad del sonido en el aire en m/s
d = 0.20; % Distancia entre los oidos en m

format short e

function plot_lines(slope)
  hold
  grid on

  if (slope < 0) % Asi las recta la dibujo en el intervalo de interes
    line([(2 + 0.20) 0], [1 (1 - slope*(2 + 0.20))])
  else
    line([(2 + 0.20) 4], [1 (1 + slope*(4 - 0.20))])
  endif
  plot([2, 2.20], [1, 1], 'x', 'color', 'r')
  xlim([0, 3])
  ylim([0, 4])
  xlabel("x [m]")
  ylabel("y [m]")
endfunction

function slope = calculate_lines (audios, Fs, N, delta_n, upsample_gph)
  global c d
  L = 1;

  if (upsample_gph)
    L = 10; % Factor de upsampling
    Fs = Fs * L; % Nueva frecuencia de muestreo
    b = fir1(150, 1/L, "low") * L; % Filtro pasabajos interpolador para el upsampling
    delay = mean(round(grpdelay(b))); % Retardo del filtro, como es lineal el filtro entonces el valor medio es el valor de cada elemento realmente y ademas es la mitad del orden del filtro
  endif

  tita = []; % Angulos
  slope = [];% Pendientes

  x = audios(:,1);
  y = audios(:,2);
  n0 = N / 2;
  j = 1;

  while ((n0 + N/2) < rows(audios))
    w_start = n0 - N/2 + 1;
    w_end = n0 + N/2;
    dft_x = fft(x(w_start:w_end));
    dft_y = fft(y(w_start:w_end));
    R = dft_x .* conj(dft_y);
    Gph = R ./ abs(R); % Un epsilon que evita la inestabilidad del algoritmo cuando R es muy cercano a 0
    gph = real(ifft(Gph));
    if (upsample_gph) % Realizar un upsample de gph para estimar mejor los retardos
      aux = upsample(gph, L); % Upsample de gph
      gph = zeros(rows(aux)+delay, 1); % Agrego delay cantidad de 0s de forma que no pierda la informacion original en el filtro al final de la senial
      gph(1:end-delay) = aux;
      gph = filter(b, 1 , gph); % Filtramos/Interpolamos
      gph = gph(delay+1:end); % Anulamos el desfase introducido por el filtro
    endif
    [~, m] = max(gph);
    if (m > N*L/2) % Si el indice es mayor a la mitad del ancho de la ventana
      m = m - N*L;
    endif

    curr_tau = (m-1) / Fs;
    tau_xy(j) = (m-1) / Fs;
    j = j + 1;
    n0 = n0 + delta_n; % Actualizo la ventana
  endwhile

  mean_tau = mean(tau_xy)
  tita = acos(mean_tau * c / d);
  slope = tan(tita);
  clear tau_xy

endfunction

[audios, Fs] = audioread("audio3.wav");
cut_n = rows(audios) / 5;
N = 10000;
delta_n = N / 2;
disp("\nRetardos medios obtenidos mediante GCC-PHAT (en segundos)\n")
for i = (1:5)
  hf = figure(i);
  slope = calculate_lines(audios(1+cut_n*(i-1):cut_n*i, :), Fs, N, delta_n, true);
  plot_lines(slope)
  print(hf, strcat("ejercicio12", num2str(i), ".png"))
endfor

clear all
```